

MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR



Fuzzy szabály-interpolációs módszerek és mintaadatok alapján történő automatikus rendszergenerálás

PhD értekezés

JOHANYÁK ZSOLT CSABA

OKLEVELES MÉRNÖK-INFORMATIKUS

OKLEVELES GÉPÉSZMÉRNÖK

HATVANY JÓZSEF INFORMATIKAI TUDOMÁNYOK DOKTORI ISKOLA

A DOKTORI ISKOLA VEZETŐJE:

DR. TÓTH TIBOR

TUDOMÁNYOS VEZETŐ:

Dr. Kovács Szilveszter.

Miskolc, 2007.

Nyilatkozat

Alulírott, Johanyák Zsolt Csaba kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Az értekezés bírálatai és a védésről készült jegyzőkönyv megtekinthető a Miskolci Egyetem Gépészmérnöki és Informatikai Karának Dékáni Hivatalában.

Miskolc, 2007. november 19.

Johanyák Zsolt Csaba

Köszönetnyilvánítás

Az értekezésben a Miskolci Egyetem Gépészmérnöki és Informatikai Karán, a Hatvany József Informatikai Tudományok Doktori Iskola keretein belül valamint a Kecskeméti Főiskola GAMF Karán, a Kalmár Sándor Informatikai Intézet keretein belül végzett kutatási munkám során elért eredményeket foglaltam össze.

Ezúton fejezem ki köszönetemet témavezetőmnek Dr. Kovács Szilveszternek, aki széles körű és kiemelkedő szakmai ismereteivel, gyakorlati tapasztalataival munkámat mindvégig támogatta, a dolgozat megírásához nélkülözhetetlen segítséget nyújtott.

Köszönetet mondok Prof. Dr. Tóth Tibornak, a doktori iskola vezetőjének, aki a kutatómunkámat támogatta és hasznos tanácsokkal látott el.

Köszönetet mondok Dr. Erdélyi Ferencnek, aki hasznos tanácsokkal segítette az értekezés elkészítését.

Köszönetet mondok Dr. Danyi Józsefnek a Kecskeméti Főiskola rektorának, Dr. Kodácsy Jánosnak a GAMF Kar dékánjának és Dr. Kovács Tamásnak a Kalmár Sándor Informatikai Intézet vezetőjének, akik lehetővé tették és támogatták, hogy a doktori képzésben részt vegyek, konferenciákon publikáljak, és kutatómunkát folytassak.

Köszönetet mondok munkatársaimnak, akik oktatási feladataim egy részének átvállalásával segítették kutatómunkámat.

Hálámat és köszönetemet fejezem ki családtagjaimnak biztatásukért és támogatásukért.

Köszönetet mondok mindazoknak, akik közvetlen vagy közvetett módon elősegítették, támogatták munkámat, ezáltal hozzájárultak ahhoz, hogy a dolgozat elkészüljön.

Tartalomjegyzék

Nyilatkozat	ii
Köszönetnyilvánítás	iii
Tartalomjegyzék	iv
1. Bevezetés	1
2. Tudományos előzmények	3
2.1. Az értekezésben felmerülő alapfogalmak és jelölésük ([H7][H30][H52][H59])	3
2.2. Rendezés és távolság	6
2.2.1. Az α -vágat alapú KH fuzzy távolság és a rá épülő rendezés	6
2.2.2. Az α -vágat alapú VKK fuzzy távolság és a rá épülő rendezés	7
2.2.3. A referencia pont alapú távolság és a rá épülő rendezés	8
2.3. Szabályok és szabálybázis	8
2.4. Következtetés ritka szabálybázist alkalmazó fuzzy rendszerekben	11
2.5. Eglépéses szabály-interpolációs módszerek	12
2.5.1. Lineáris szabály-interpoláció (KH módszer)	12
2.5.2. A KH interpoláció módosítása több szabály figyelembevételével	13
2.5.3. VKK módszer	13
2.5.4. Fuzzy jelleg megőrzésén alapuló szabály-interpoláció (GK módszer)	14
2.5.5. Relatív fuzzy jelleg megőrzésén alapuló szabály-interpoláció (KHG módszer) ..	14
2.5.6. Módosított α -vágat alapú szabály-interpoláció (MACI)	15
2.5.7. Javított többdimenziós módosított α -vágat alapú szabály-interpoláció (IMUL) ...	16
2.5.8. HCL szabály-interpoláció	17
2.5.9. Fuzzy interpoláció bizonytalan környezetben (FIVE)	17
2.6. A fuzzy szabály-interpoláció általánosított módszertanát követő eljárások	18
2.6.1. A fuzzy szabály-interpoláció általánosított módszertana (GM)	18
2.6.2. Testmetszés alapú interpoláció (SCM)	19
2.6.3. Hasonlóság megőrzési módszer (ST)	20
2.6.4. A lineáris revíziós elven alapuló módszerek	21
2.6.5. IGRV módszer	22
2.7. Összegzés	23
3. Kutatási célok	24
4. Fuzzy halmaz-interpoláció – javasolt módszerek	25
4.1. Nyelvi értékek eltolásának elve	26
4.2. Polár-vágaton alapuló halmaz-interpoláció (FEAT-p)	27
4.3. Legkisebb négyzetek elvén alapuló halmaz-interpoláció (FEAT-LS)	30
4.4. Halmaz-interpoláció bizonytalan környezetben	32
4.4.1. Bizonytalan környezet	32
4.4.2. Halmaz-interpoláció a bizonytalan környezetben (VESI)	34
5. Szabálmódosításon alapuló egyszabályos következtetés – javasolt módszerek	38
5.1. Polár-vágat alapú egyszabályos következtetés (SURE-p)	40
5.2. Legkisebb négyzetek elvén alapuló egyszabályos következtetés (SURE-LS)	42
5.3. Szabálmódosítás bizonytalan környezetben (REVE)	45
6. Új fuzzy szabály-interpolációs módszerek	49
6.1. A három szabály-interpolációs eljárás	49
6.1.1. Polár-vágat alapú szabály-interpoláció (FRIPOC)	49

6.1.2. Legkisebb négyzetek elvén alapuló szabály-interpoláció (LESFRI)	49
6.1.3. Kétlépéses fuzzy szabály-interpoláció bizonytalan környezetben (VEIN).....	49
6.2. A konzekvens fuzzy halmazok helyzetének meghatározása	50
7. FRI Matlab toolbox	52
7.1. Adatszerkezetek	52
7.2. Az eljárásgyűjtemény és a rá épülő programok	54
7.2.1. TestIt.....	54
7.2.2. FRIT	56
7.2.3. Sűgő.....	63
8. Fuzzy következtető rendszer automatikus generálása.....	65
8.1. Rendszergenerálás iteratív szabálybázis kiterjesztéssel.....	65
8.1.1. A szabálybázis kiterjesztés elve	66
8.1.2. Fuzzy halmazok paraméterezése és kötöttségek	69
8.1.2.1. Töréspontok.....	69
8.1.2.2. Relatív távolságok	71
8.1.2.3. Ruspini partíció megőrzése	72
8.1.2.4. Referencia pont	74
8.1.2.5. A paraméterezési mód kiválasztása.....	75
8.1.3. Paraméter-azonosítás és szabálybázis kiterjesztés	76
8.1.4. A lépésköz nagysága	78
8.1.5. Teljesítménymutató	78
8.1.6. Új szabály generálása	80
8.2. Fuzzy klaszterezésen alapuló automatikus rendszergenerálás	82
8.2.1. Egydimenziós FCM fuzzy klaszterezés	83
8.2.2. Az optimális klaszterszám meghatározása	86
8.2.3. Fuzzy halmazok és partíció klaszterekből.....	88
8.2.4. Vetítés az antecedens térbe	89
8.2.5. Paraméter-azonosítás.....	91
8.3. RuleMaker - Automatikus rendszergenerálást támogató Matlab eljárásgyűjtemény....	93
8.3.1. Bemeneti adatok	94
8.3.2. Nyers rendszer létrehozása a kimenet szélsőértékein alapuló módszerrel	95
8.3.3. Mentés	96
8.3.4. Paraméter-azonosítás és szabálybázis kiterjesztés	97
9. Szabály-interpolációs módszerekkel dolgozó speciális fuzzy következtető rendszerek létrehozása modellezési feladatok megoldására.....	102
9.1. Fuzzy rendszergenerálás szintetikus előállított adatok alapján	102
9.1.1. SISO rendszer.....	102
9.1.2. MISO rendszer	103
9.2. Anaerob kúpos szuszpendált ágy reaktor működésének fuzzy modellezése	104
9.3. Kőolajkutatás során végzett fúrások mérési eredményei közötti összefüggés fuzzy modellezése	109
10. Új tudományos eredmények tézisekben történő összefoglalása	112
11. Az elért eredmények hasznosíthatósága és további kutatási tervek.....	114
12. Összefoglalás.....	115
13. Summary	117
14. Irodalomjegyzék.....	119
14.1. Az értekezés témakörében megjelent saját tudományos közlemények.....	119

14.2. Hivatkozott irodalom.....	121
1. Melléklet. A szennyvíztisztítási feladathoz készített kétbemenetes fuzzy modellek identifikációjának eredményei	1

1. Bevezetés

A hétköznapi életben gyakran találkozunk olyan helyzettel, amikor egy fogalmat, egy tárgyat, egy értéket, stb. nem tudunk egyértelműen egyik vagy másik kategóriába besorolni, így vagy úgy minősíteni. Úgy érezzük, hogy többé-kevésbé ide is, oda is tartozik, ilyen is, olyan is. A klasszikus példa erre a személyek életkoruk alapján fiatal, középkorú és idős korcsoportba történő besorolása. Egy húszévesnél egyértelmű a helyzet, de egy 35 vagy 55 éves személy esetében már nem az.

A probléma oda vezethető vissza, hogy az emberi gondolkodásban nincs egyértelmű éles határvonal egyes fogalmak között. A kategóriák közötti éles határvonal meghúzása bár egyértelművé teszi a döntéshozatalt, de igen gyakran szembesülünk azzal az érzéssel, hogy nem vagyunk biztosak döntésünk helyességében. Egy dolgozat pontszámának értékelése során sokszor nincs lényegi különbség aközött, amit a határvonal alatti utolsó vagy a határvonal feletti első pontszámmal minősítenek.

Egy kategóriába történő besorolást gyakran követ egy következő lépés, ami valamilyen szabály alkalmazásával az értékelés tárgyát képező személy vagy bármely más objektum jövőjét hosszabb vagy rövidebb távon befolyásolja. Például korosztály alapján egy film megtekinthetőségének megállapítása, egy vizsgajegy vagy akár egy alkalmassági vizsga kimenetelének meghatározása.

A fenti, első látásra nehezen kezelhetőnek tűnő helyzetek, azaz a nyelvi fogalmakban rejlő bizonytalanság (pontatlanság) kezelésére teremtette meg L. A. Zadeh 1965-ben a fuzzy (homályos, életlen [H87]) fogalmat és a ráépülő logika alapjait. A fuzzy megközelítésen alapuló megoldások széleskörű elterjedését az iparban E. H. Mamdani és S. Assilian sikeres gőzgép szabályozási kísérletei [H54] alapozták meg.

Az elméleti háttér alapos kidolgozásával párhuzamosan és annak eredményeképpen a fuzzy logika széleskörű alkalmazást nyert hétköznapi életünk sok területén is (mosógép, vízmelegítő, porszívó, stb.), ahol több, esetenként homályosan megfogalmazott szempont által befolyásolt döntés meghozatala szükséges, vagy a vizsgált rendszer matematikai modelljének megalkotása és az optimális megoldás megtalálása túl bonyolult, illetve a modell pontatlansága miatt a kapott eredmény használhatatlan.

A fuzzy megközelítés segítségével irányíthatóvá válnak olyan folyamatok, amelyek a klasszikus elméletekkel csak igen körülményesen lennének kezelhetők. Ezek általában ember által jól kézben tarthatóak, azonban matematikai modelljük hiányos vagy túl komplikált, esetleg pontatlan volta miatt a klasszikus irányítástechnikai módszerek hatékonyan rájuk nem alkalmazhatók.

Értekezésem olyan fuzzy rendszerek következtetési módszereivel és automatikus generálásával foglalkozik, amelyek ritka szabálybázis esetében is képesek értelmezhető eredmények előállítására.

A témához kapcsolódó fontosabb fogalmak, jelölésmódok és szabály-interpolációs módszerek áttekintése után (2. fejezet) ismertetem kutatási célkitűzéseimet (3. fejezet).

Az elért tudományos eredmények részletes ismertetését három általam kidolgozott halmaz-interpolációs módszerrel kezdem (4. fejezet), majd három szabálmódosításon alapuló egyszabályos következtetési eljárásomat mutatom be (5. fejezet). Az előzőekben ismertetett eljárások segítségével három szabály-interpoláción alapuló fuzzy következtetési módszert dolgoztam ki, melyeket a 6. fejezetben tárgyalok. A fuzzy szabály-interpoláció témakörét a 7. fejezet zárja. Ebben mutatom be az általam Matlab környezetre írt keretrendszert, amelynek

célja a fuzzy szabály-interpoláción alapuló következtetési módszerek összevetésének, vizsgálatának és gyakorlati alkalmazásának támogatása.

Értekezésem második fő témaköre a fuzzy következtető rendszerek mintaadatok alapján történő automatikus generálása (8. fejezet). Itt két szabálybázis-kiterjesztésen alapuló és egy fuzzy klaszterezést alkalmazó módszert is javaslok. Ezt követően a 9. fejezetben mesterségesen előállított és valós mérési adatokon alapuló fuzzy modellezési feladatokat oldok meg.

Az értekezés utolsó részében tárgyalom az elért eredmények hasznosíthatóságát (10. fejezet) és elképzeléseimet a téma továbbviteléről (11. fejezet).

Az értekezés témakörében megjelent saját tudományos közlemények és a kutatómunka során fejlesztett szoftverek szabadon hozzáférhetőek és letölthetőek a [S32] címről.

2. Tudományos előzmények

A fejezet első részében áttekintem a fontosabb alapfogalmakat és jelölésüket, majd ismertetek néhány olyan további fogalmat és jellemzőt, amelyek nélkülözhetetlenek a fuzzy szabály-interpolációs módszerek áttekintéséhez és az értekezésben bemutatásra kerülő kutatási eredményeim ismertetéséhez.

2.1. Az értekezésben felmerülő alapfogalmak és jelölésük ([H7][H30][H52][H59])

A jelen szakasz felépítése a következő mintát követi: félkövéren kiemelem a fogalom nevét, majd rövid szöveges résszel és szükség esetén ábrákkal, egyenletekkel magyarázom azt.

Alaphalmaz. Jelölése: X vagy U

Az alaphalmaz (univerzum) egy olyan éles (nem fuzzy) halmaz, amelyen a fuzzy halmazok elemeit értelmezzük (értelmezési tartomány). Az értekezés további részében megjelenő alaphalmazok a valós számok halmazának részhalmazai lesznek ($X \subset \mathbb{R}$).

Fuzzy halmaz. Jelölése: a római ABC egy nagybetűje, pl. A

A hagyományos (crisp, éles) halmazfogalom kiterjesztése. Míg a crisp halmazok esetén egy alaphalmaz minden elemét egyértelműen a halmaz tagjaként vagy halmazon kívülként jelölünk meg, addig fuzzy halmazok esetén az alaphalmaz-elem halmazhoz tartozásának mértékét is definiálhatjuk (ld. tagsági függvény). A fuzzy halmazok fogalmát először Zadeh [H87] használta nyelvi kifejezésekben rejlő bizonytalanság leírására.

Nyelvi érték (címke)

Egy fuzzy halmazhoz rendelt név (címke), ami egy emberi nyelven megfogalmazott szó vagy esetleg szimbólum.

Nyelvi változó

Egy olyan speciális változó, ami nyelvi értékeket (címkeket) és általuk fuzzy halmazokat kaphat értékül.

Megfigyelés. Jelölése: A^*

Egy fuzzy következtető rendszer bemenő adatául szolgáló fuzzy halmaz egydimenziós bemenet esetén, vagy egy fuzzy halmazokat tartalmazó N elemű vektor N -dimenziós bemenet esetén.

Tagsági függvény. Jelölése: μ_A

A $\mu_A : X \rightarrow [0,1]$ függvény megadja, hogy az X alaphalmaz elemei milyen mértékben tartoznak az A fuzzy halmazhoz.

Fuzzy halmaz megadása

Diszkrét alaphalmaz esetén a fuzzy halmazt elemei és a hozzájuk tartozó tagsági értékek felsorolásával adjuk meg az alábbi jelölést alkalmazva

$$A = \mu_{A,1} / x_1 + \mu_{A,2} / x_2 + \dots + \mu_{A,n} / x_n = \sum_{i=1}^n \mu_{A,i} / x_i, x_i \in X, \quad (1)$$

ahol $X = \{x_1, \dots, x_n\}$ az alaphalmaz, a „/”, a „+” jeleknek nincs aritmetikai jelentésük, csak gyakorlati jelölések [H30]. Folytonos alaphalmaz esetén a fuzzy halmazt az alaphalmaz és a tagsági függvény segítségével adjuk meg az alábbi jelölést alkalmazva

$$A = \int_{x_{\min}}^{x_{\max}} \mu_A(x) / x, x \in X, \quad (2)$$

ahol $X = [x_{\min}, x_{\max}]$ az alaphalmaz, a diszkrét esethez hasonlóan a „ $\int$ ” és a „/” csak a jelölés segédeszközei.

α -vágat. Jelölése: $[A]_\alpha$

Egy A halmaz α -vágata egy olyan éles halmaz, amit következőképpen definiálunk:

$$[A]_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha; \alpha \in (0, 1]\}. \quad (3)$$

Mag. Egy A fuzzy halmaz magja: $[A]_1$

Tartó. Jelölése: $\text{supp}(A)$ vagy $[A]_{0+}$

Egy A fuzzy halmaz tartója

$$\text{supp}(A) = \{x \in X \mid \mu_A(x) > 0\}. \quad (4)$$

Kompakt fuzzy halmaz

Az A fuzzy halmaz akkor kompakt, ha a tartója korlátos [H26]. Ez gyakorlatilag a

$$\text{supp}(A) \subseteq X \quad (5)$$

teljesülését jelenti.

Normál (normalizált) fuzzy halmaz. Az A fuzzy halmaz akkor normál, ha $\exists x \in X$, amelyre $\mu_A(x) = 1$.

Szubnormális fuzzy halmaz. Az A fuzzy halmaz akkor szubnormális, ha $\forall x \in X$ esetén $\mu_A(x) < 1$.

Konvex fuzzy halmaz. Valamely X alaphalmazon értelmezett A fuzzy halmaz konvex, ha valamennyi α -vágata konvex halmaz:

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)] \quad \forall x_1, x_2 \in \mathbb{R}^2, \lambda \in [0, 1]. \quad (6)$$

Fuzzy jelle. Jelölése: $f_{A,L}, f_{A,U}$.

A Gedeon és Kóczy által a [H22] irodalomban bevezetett alsó és felső (bal- és jobb oldali) fuzzy jelleget a mag és tartó azonos oldali szélső pontjai közötti vízszintes irányú távolságként definiáljuk konvex és normalizált fuzzy (CNF – Convex Normal Fuzzy) halmazokon

$$f_{A,L} = \inf\{[A]_1\} - \inf\{[A]_{0+}\}, \quad (7)$$

$$f_{A,U} = \sup\{[A]_{0+}\} - \sup\{[A]_1\}, \quad (8)$$

ahol $f_{A,L}$ és $f_{A,U}$ az A fuzzy halmaz alsó illetve felső fuzzy jellege.

Egyelemű (singleton) fuzzy halmaz

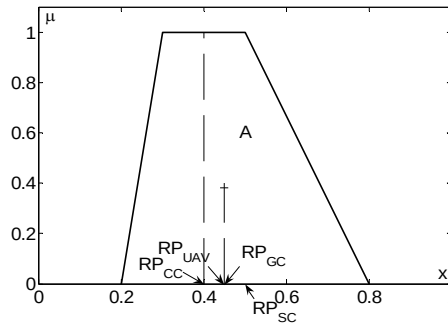
A továbbiakban egyeleműnek nevezem azt a fuzzy halmazt, amely csak egyetlen alaphalmaz elemet tartalmaz, aminek tagsági értéke 1. Grafikus képe alapján (függőleges egyenes) szingleton vagy szálla alakúnak is nevezik ezt a speciális halmaztípust. Az egyelemű fuzzy halmaz alsó és felső fuzzy jellege nulla ($f_{A,L} = 0$ és $f_{A,U} = 0$).

Egyértékű fuzzy halmaz

A továbbiakban egyértékűnek nevezem azt a fuzzy halmazt, amelynek eleme azonos tagsági értékű. Amennyiben ez a tagsági érték 1-el egyenlő, akkor egy hagyományos (crisp, éles) halmazról van szó. Az egyértékű fuzzy halmaz alsó és felső fuzzy jellege nulla ($f_{A,L} = 0$ és $f_{A,U} = 0$).

Referencia pont. Jelölése: $RP(A)$

A referencia pont egy kitüntetett elem a halmaz tartójáról, amit egyes fuzzy eljárásokban a halmaz helyzetének jellemzésére használnak. A referencia pont kijelölésénél több lehetőség közül is választhatunk. Így a feladatot elláthatja például a mag középpontjának (RP_{CC})[H4][H10][S12], az alakzat tömegközéppontjának (RP_{GC})[H25], a tartó középpontjának (RP_{SC}) [H10] megfelelő alaphalmaz elem vagy szakaszonként lineáris halmazalak esetén a töréspontok abszcisszáinak súlyozott (RP_{WAV}) vagy súlyozatlan (RP_{UAV}) átlaga. Az 1. ábra egy trapéz alakú fuzzy halmaz esetén mutatja be az egyes referencia pont típusokat.



1. ábra. Referencia pont típusok

A legtöbb szabály-interpolációs eljárás megköveteli, hogy a referencia pontban a tagsági függvény értéke érje el maximumát

$$\mu_A(RP(A)) = \max\{\mu_A(x) \mid x \in X\}. \quad (9)$$

Lefedettségi mértéke

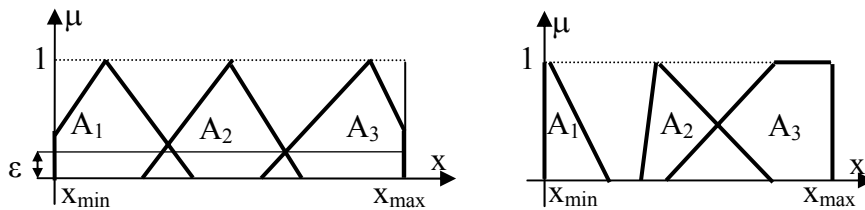
Feltételezzük, hogy az $X = [x_{\min}, x_{\max}]$ alaphalmazon véges számú fuzzy halmazzal ($A_j, j \in \{1, \dots, n\}$) rendelkezünk. Az alaphalmaz lefedettség mértékét (ε) a következőképpen definiáljuk

$$\arg \max_{\varepsilon} (\forall x \in X, \exists j \in \{1, \dots, n\} : \mu_{A_j}(x) \geq \varepsilon), \quad \varepsilon \in [0, 1], \quad (10)$$

ahol $\arg \max_{\varepsilon}(\cdot)$ azt az ε értéket jelenti, amelyre a zárójelben szereplő kifejezés felveszi maximális értékét.

Partíció

Hagyományosan (pl. [H52]) egy X alaphalmaz fuzzy partíciójának nevezzük azt az $A = \{A_1, \dots, A_n\}$ halmazcsaládot, amelynek halmazai biztosítják az X univerzum 0 -nál nagyobb mértékű lefedését ($\varepsilon > 0$ a 2. ábra bal oldali részén). A szabály- és halmaz-interpolációs feladatoknál ez a feltétel nem teljesül mindig, ezért a partíció fogalmának kiterjesztéseként **ritka partíciónak** nevezzük azt az esetet, amikor csak $\varepsilon = 0$ -ra teljesül a (10) kifejezés (a 2. ábra jobb oldali részén). A továbbiakban az egyszerűség kedvéért röviden **partícióként** hivatkozom az eredeti szigorúan értelmezett (fedő) partíció fogalomra és annak kiterjesztésére, a ritka partícióra is.



2. ábra. Szigorúan értelmezett (fedő) partíció és ritka partíció

2.2. Rendezés és távolság

A fuzzy szabály-interpolációs módszerek vizsgálata során háromféle megoldással találkozhatunk a fuzzy halmazok rendezése és távolságának értékelése vonatkozásában.

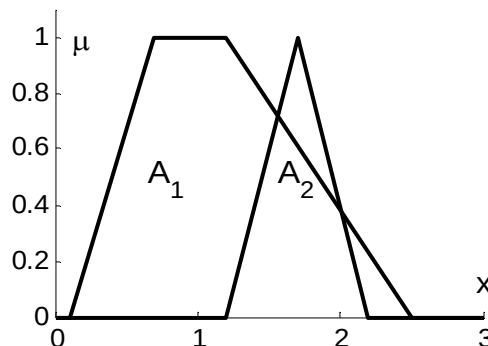
2.2.1. Az α -vágat alapú KH fuzzy távolság és a rá épülő rendezés

A Kóczy és Hirota [H44] által definiált első megközelítés α -vágatokat alkalmaz. Eszerint az A_k halmaz akkor előzi meg az A_l halmazt ($A_k \prec A_l$), ha $\forall \alpha \in (0, 1]$ és $k \neq l$ esetén teljesül az alábbi két egyenlőtlenség

$$\inf\{[A_k]_\alpha\} < \inf\{[A_l]_\alpha\}, \quad (11)$$

$$\sup\{[A_k]_\alpha\} < \sup\{[A_l]_\alpha\}, \quad (12)$$

ahol $\inf$ és $\sup$ az α -vágat alsó illetve felső végpontját jelöli. A megoldás hátránya, hogy csak részleges rendezést biztosít. Például a 3. ábra egy olyan esetet illusztrál, amikor csak néhány α -szint esetén írható fel a „ $<$ ” reláció. A módszer csak CNF halmazokra alkalmazható.

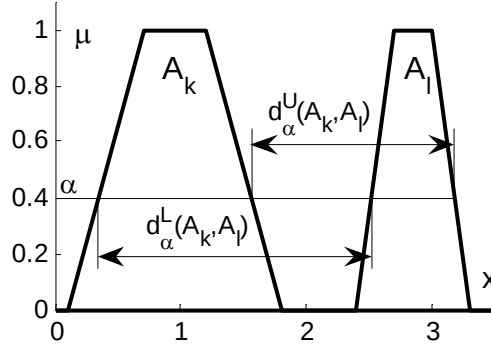


3. ábra. A Kóczy és Hirota által javasolt α -vágat alapú megközelítéssel nem állapítható meg a két fuzzy halmaz sorrendje

Kóczy és Hirota [H44] a halmazok távolságának értékelésére szintén α -vágat alapú megoldást javasolt. A **KH fuzzy távolság** egy alsó ($d_\alpha^L(A_k, A_l)$) és egy felső ($d_\alpha^U(A_k, A_l)$) eltérést (4. ábra) mér α -vágatonként az alábbiak szerint

$$d_\alpha^L(A_k, A_l) = |\inf\{[A_k]_\alpha\} - \inf\{[A_l]_\alpha\}|, \quad (13)$$

$$d_\alpha^U(A_k, A_l) = |\sup\{[A_k]_\alpha\} - \sup\{[A_l]_\alpha\}|. \quad (14)$$



4. ábra. Alsó és felső KH fuzzy távolság az α -szinten

2.2.2. Az α -vágat alapú VKK fuzzy távolság és a rá épülő rendezés

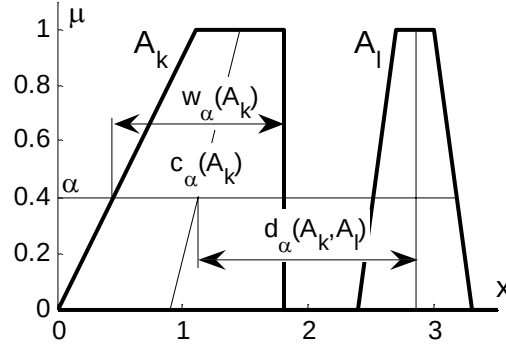
Vass, Kalmár és Kóczy [H75] szintén α -vágat alapú megoldást javasoltak a fuzzy halmazok távolságának mérésére. Az előzőektől eltérően az α -vágatokat nem végpontjaik segítségével írták le, hanem a vágat középpontjával (c_α) és szélességével (w_α) (10. ábra)

$$c_\alpha(A) = \frac{\inf\{[A]_\alpha\} + \sup\{[A]_\alpha\}}{2}, \quad (15)$$

$$w_\alpha(A) = \sup\{[A]_\alpha\} - \inf\{[A]_\alpha\}. \quad (16)$$

A VKK fuzzy távolság α -vágatonként a középpontok euklideszi távolságát méri

$$d_\alpha(A_k, A_l) = |c_\alpha(A_k) - c_\alpha(A_l)|. \quad (17)$$



5. ábra. VKK távolság

A VKK távolságon alapuló rendezésnél az A_k halmaz akkor előzi meg az A_l halmazt ($A_k \prec A_l$), ha $\forall \alpha \in (0, 1]$ és $k \neq l$ esetén teljesül az alábbi egyenlőtlenség

$$c_\alpha(A_k) < c_\alpha(A_l). \quad (18)$$

A megoldás előnye, hogy egyszerűsíti a számításokat, mivel α -vágatonként csak egy távolság mértékkel kell dolgozni, viszont még mindig nem alkalmas arra, hogy általános esetben egyetlen számmal kifejezhessük a halmazok távolságát. Bár bővítette az alkalmazási teret, de továbbra is csak részleges rendezést biztosít. Könnyen elképzelhető olyan eset, amikor bizonyos α -vágatoknál az egyik halmaz c -értéke kisebb, míg másoknál a másik halmaz c -értéke kisebb.

2.2.3. A referencia pont alapú távolság és a rá épülő rendezés

A Baranyi, Kóczy és Gedeon (pl. [H4]) által is javasolt harmadik megközelítés referencia pont segítségével jellemzi a fuzzy halmazok helyzetét. Így az A_k halmaz akkor előzi meg az A_l halmazt ($A_k \prec A_l$), ha $RP(A_k) < RP(A_l)$. Ebben az esetben a halmazok távolságát referencia pontjaik euklideszi távolsága határozza meg

$$d(A_k, A_l) = |RP(A_k) - RP(A_l)|. \quad (19)$$

Ez a megoldás kisebb számításigényű és szélesebb körben alkalmazható, mint az α -vágat alapú fuzzy távolságok.

2.3. Szabályok és szabálybázis

A fuzzy következtető rendszerek (a továbbiakban röviden fuzzy rendszerek) szabályalapú tudásábrázolást alkalmaznak. A szabályok feltétel (antecedens) és következmény (konzekvens) részében nyelvi változók és fuzzy halmazok (vagy azokat azonosító nyelvi értékek) szerepelnek az alábbi formában

$$R : HA \ a_1 = A_{1,j_1} \ \acute{E}S \dots \acute{E}S \ a_N = A_{N,j_N} \ AKKOR \ b_1 = B_{1,k_1} \ \acute{E}S \dots \acute{E}S \ b_M = B_{M,k_M}, \quad (20)$$

ahol a_i bemeneti nyelvi változó az i . antecedens dimenzióban, N az antecedens dimenziók száma, j_i a szabály i . dimenzióbeli antecedensének sorszáma a saját partíciójában, b_i a kimeneti nyelvi változó az i . konzekvens dimenzióban, k_i a szabály i . dimenzióbeli konzekvens nyelvi értékének sorszáma a saját partíciójában és M a konzekvens dimenziók száma.

Egy fuzzy rendszerben tárolt szabályok összességét szabálybázisnak nevezzük

$$RB = \{R_1, R_2, \dots, R_{N_R}\}, \quad (21)$$

ahol N_R a szabályok száma.

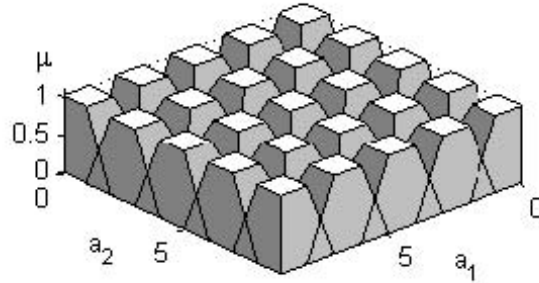
A szakirodalom döntő többségéhez hasonlóan a továbbiakban csak olyan fuzzy következtető rendszerekkel és szabályokkal foglalkozom, amelyek egy kimeneti dimenzióval rendelkeznek, ugyanis a többkimenetű rendszerek szabálybázisa információvesztés nélkül lebontható egykimenetű szabálybázisokra.

A fuzzy rendszerek működésének és az egyes következtetési eljárások alkalmazhatóságának egyik kulcskérdése, hogy a szabálybázisban tárolt szabályok antecedensei milyen mértékben fedik a bemeneti alaphalmazok által meghatározott antecedens (bemeneti) teret. Az antecedens tér lefedettségének mértékét (ε) az alábbi összefüggés határozza meg

$$\arg \max_{\varepsilon} \left(\min_{i=1}^N \left\{ \max_{j=1}^{n_i} \left\{ t(A_{i,j}, A_i^*) \right\} \right\} \right) \geq \varepsilon, \forall A_i^* \subset X_i, \quad \varepsilon \in [0, 1], \quad (22)$$

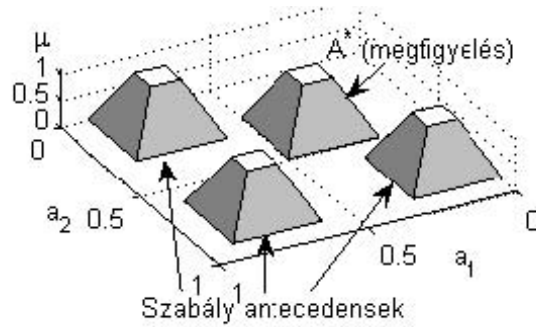
ahol X_i az antecedens tér (alaphalmaz) i -edik dimenziója, A_i^* a megfigyelést az i -edik antecedens dimenzióban leíró halmaz, A_{ij} az i -edik antecedens dimenzió j -edik nyelvi értéke, t tetszőleges t-norma, n_i az i -edik antecedens dimenzió nyelvi értékeinek száma, N az antecedens dimenziók száma és $\arg \max_{\varepsilon}(\cdot)$ azt az ε értéket jelenti, amelyre a zárójelben szereplő kifejezés felveszi maximális értékét.

Egy szabálybázist akkor nevezünk **fedőnek** vagy **sűrűnek**, ha a (22) összefüggés nullánál nagyobb ε értékre teljesül. A 6. ábra egy ilyen szabálybázis antecedens terét illusztrálja egy kétdimenziós bemenetű rendszer esetén.



6. ábra. Sűrű szabálybázis antecedens tere

Minden szabály antecedenst egy csonkagúla jelképez, amit az antecedens fuzzy halmazok határoznak meg. Itt bármilyen megfigyelés esetén létezik legalább egy olyan szabály, amelynek antecedense metszi vagy átfedi a megfigyelést. Egy szabálybázist akkor tekintünk ritkának vagy nem fedőnek, ha a (22) összefüggés csak $\varepsilon=0$ értékre teljesül, azaz a kétdimenziós példánál maradva az antecedens síkban üres területek találhatóak. A 7. ábra egy ilyen esetet szemléltet. Itt az A^* megfigyelés esetén a szabálybázis három szabálya közül egyik antecedense sem érintkezik a megfigyeléssel.



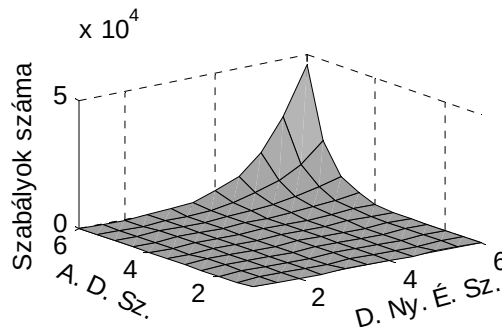
7. ábra. Ritka szabálybázis antecedens tere

A bemeneti nyelvi változók (antecedens dimenziók) és a dimenziókénti fuzzy halmazok számának növekedése a teljes lefedettség biztosításához szükséges szabályok mennyiségének és a szabálybázisra épülő fuzzy rendszer komplexitásának robbanásszerű növekedését eredményezi (8. ábra), ami tárolási és feldolgozási nehézségeket okozhat [H47][H52]. Általános esetben a szükséges szabályok számát

$$N_R = \prod_{i=1}^N n_i \quad (23)$$

összefüggés határozza meg, ahol n_i az i . antecedens dimenzió fuzzy halmazainak száma és N az antecedens dimenziók száma, továbbá feltételezzük, hogy minden partícióban teljesül a (22) feltétel $\varepsilon > 0$ -ra. A 8. ábra azt az esetet illusztrálja, amikor minden dimenzióban azonos a fuzzy halmazok száma. Erre az egyszerűsítésre az ábrázolhatóság érdekében volt szükség.

A komplexitás növekedés problémájának megoldását jelentheti olyan szabálybázis kialakítása, amely nem fedő (azaz ritka), de tartalmazza a feladat megoldásához szükséges releváns tudást [H40].



8. ábra. Szabályok számának alakulása az antecedens dimenziók számának (A. D. Sz.) és a dimenziókénti nyelvi értékek számának (D. Ny. É. Sz.) függvényében

Szándékosság mellett még további két ok vezethet ritka szabálybázis kialakulásához. Egyrészt előfordulhat, hogy a szabálybázis alapjául szolgáló ismeretanyag eleve hiányos, másrészt a rendszer fejlesztésének utolsó lépésében az ún. paraméter-azonosítás (hangolás) során változhat az antecedens fuzzy halmazok alakja és helyzete, ami lefedetlen területek kialakulását okozhatja [H11].

Számos szabály-interpoláción alapuló fuzzy következtetési eljárásnál találkozunk a **két közrefogó szabály** fogalmával. Két szabályt akkor tekintünk a megfigyelést közrefogónak, ha minden antecedens dimenzióban teljesül az, hogy az első szabály antecedens halmaza

megelőzi a megfigyelés halmazt és a második szabály antecedens halmaza követi a megfigyelés halmazt

$$A_{i,j_i}^1 \prec A_i^* \prec A_{i,k_i}^2, \forall i \in 1, \dots, N, \quad (24)$$

ahol N az antecedens dimenziók száma, A_{i,j_i}^1 az első szabály antecedens halmaza az i . dimenzióban és A_{i,k_i}^2 a második szabály antecedens halmaza az i . dimenzióban.

2.4. Következtetés ritka szabálybázist alkalmazó fuzzy rendszerekben

A hagyományos fuzzy következtetéssel (Zadeh [H88], Mamdani [H54], Larsen [H53], Takagi-Sugeno [H67], stb.) működő fuzzy rendszerek a szabály antecedensek és a megfigyelést leíró nyelvi értékek illeszkedése alapján a szabály konzekvensek súlyozott kombinációjaként határozzák meg a következményt, ami az adott módszernek megfelelően egyaránt lehet fuzzy halmaz vagy éles érték. Működésükből következően ezen eljárások alkalmazásának feltétele a szabálybázis fedő jellege (ld. 6. ábra), azaz bármely bemenő adat esetén léteznie kell legalább egy olyan szabálynak, amelynek antecedense $\varepsilon > 0$ mértékben fedi a megfigyelést a bemeneti tér minden dimenziójában. A fenti megkötés következtében indokolt lehet valamely fuzzy közelítő következtetési technika használta a hagyományos kompozíciós megközelítés helyett azon alkalmazásoknál, amelyek ritka szabálybázisra épülnek, és amelyektől elvárjuk, hogy bármely megfigyelésre kiértékelhető következtetést eredményezzenek.

A közelítő következtetés céljára kifejlesztett eljárások a legtöbb esetben valamely fuzzy szabály-interpolációs módszer segítségével határozzák meg a következményt.

Ezen eljárásokat két csoportba oszthatjuk aszerint, hogy közvetlenül állítják-e elő a következményt vagy először egy segéd szabályt interpolálnak, és annak felhasználásával számítják a következményt.

Az egylépéses módszerek családjának fontosabb képviselői a Kóczy és Hirota által kidolgozott lineáris szabály-interpoláció (KH módszer) [H40], az abnormális eredményt kizáró koordináta-transzformációra épülő MACI [H69] (Tikk és Baranyi), a szabály-interpoláció feladatát az ún. bizonytalan környezetbe helyező FIVE [H37] (Kovács és Kóczy), a relatív fuzzy jelleg megőrzésén alapuló CRF [H48] (Kóczy, Hirota és Gedeon), és a Wong, Gedeon és Tikk által javasolt IMUL [H80] valamint Dubois és Prade (DP) [H19] módszere.

A kétlépéses módszerek a Baranyi, Kóczy és Gedeon [H4] által publikált általánosított fuzzy szabály-interpolációs módszertant (GM) követik. Ezen eljárások családjába tartoznak a Baranyi és szerzőtársai által javasolt technikák [H4][H3][H5], a megfigyelés és a szabály antecedens közti hasonlóság megőrzésén alapuló ST [H86] (Yan, Mizumoto és Qiao) módszer, az általános reprezentatív értéken alapuló IGRV [H25] (Huang és Shen), valamint az általam kidolgozott LESFRI [S13], FRIPOC [S12] és VEIN [S26] módszerek. A következő két fejezetben röviden áttekintem az utolsó három kivételével az előzőekben felsorolt eljárásokat. Nem célom a számítás menetének pontos matematikai leírása, csupán az alap gondolatot és néhány fontos tulajdonságot szeretnék kiemelni egy áttekintő kép nyújtása érdekében.

Számos olyan publikációt találhatunk a szakirodalomban, amely áttekint, értékel fuzzy szabály-interpolációs módszereket. Így például Mizik, Baranyi, Korondi és Kóczy [H55], illetve Mizik, Szabó és Korondi [H56] a megfigyelés és a következmény fuzzy jellege közötti kapcsolatot vizsgálva értékelték a KH, a VKK és a MACI eljárásokat, valamint a GM egy

megvalósítását. Jenei [H26] axiomatikus megközelítést alkalmazva egy nyolc pontból álló követelmény rendszert definiált, és ezek tükrében vizsgálta a KH, a DP és a GM egy megvalósítását.

2.5. Egylépéses szabály-interpolációs módszerek

A fuzzy szabály-interpolációs módszerek első csoportjába sorolt eljárások a megfigyelés és két vagy több, ún. közrefogó szabály felhasználásával közvetlenül állítják elő az eredményt. Az áttekintést a témakör alapszabályával, a fuzzy szabály-interpolációs kutatást elindító KH eljárással nyitom, majd sorra veszem a „leszármazottakat”, azaz az eredeti lineáris szabály-interpoláció korlátait túllépő, továbbfejlesztett, kiegészített, fontosabb újabb típusokat. Utolsóként tárgyalom a FIVE módszert, amely teljesen új megközelítést alkalmazva az ún. bizonytalan környezetbe helyezte a szabály-interpoláció feladatát.

2.5.1. Lineáris szabály-interpoláció (KH módszer)

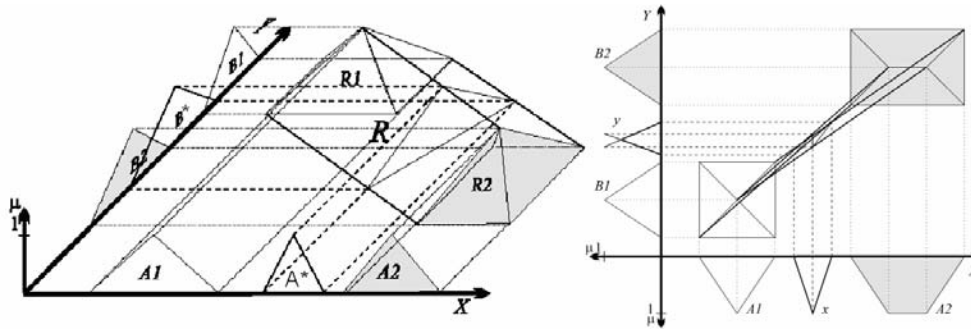
A Kóczy és Hirota által kidolgozott lineáris szabály-interpoláció (KH módszer) [H40] alap gondolata az, hogy a becsült következmény ugyanolyan arányban osztja fel a két közrefogó szabály konzekvenseinek távolságát, mint amilyen arányban a megfigyelés felosztja ugyanezen szabályok antecedens halmazainak távolságát (25). Ez a megoldás később a fuzzy szabály-interpoláció alapegyenlete (FERI-Fundamental Equation of the fuzzy Rule Interpolation) (pl. [H4]) elnevezést kapta. A módszer α -vágat alapú, a fent említett arányokat minden vágat esetén külön-külön számítja az alsó és felső távolságok segítségével

$$d_{\alpha}^i(A_1, A^*):d_{\alpha}^i(A^*, A_2)=d_{\alpha}^i(B_1, B^*):d_{\alpha}^i(B^*, B_2) \quad (25)$$

ahol A_1, A_2 a közrefogó szabályok antecedens halmazai, A^* a megfigyelés, B_1, B_2 a közrefogó szabályok konzekvens halmazai, B^* a becsült következmény. Az i helyén állhat L vagy U aszerint, hogy alsó (L) vagy felső (U) távolsággal számolunk-e. Az eljárás a klasszikus crisp Shepard interpoláció [H63] fuzzy halmazokra történő kiterjesztése [H73]. A pontos eredmény számításához elméletileg végtelen számú α -vágatra lenne szükség, azonban szakaszonként lineáris tagsági függvények esetén csak a töréspontoknak megfelelő szintekre számítják ki a következményt. Bár a módszer nem őrzi meg a szakaszonkénti linearitást, de a gyakorlati alkalmazásoknál az eltérés legtöbbször elhanyagolható [H49][H51][H69].

A KH interpoláció a következő feltételek teljesülése esetén alkalmazható: (1) a számítások során figyelembe vett halmazok CNF és kompakt jelleggel kell bírjanak, (2) minden partícióban számítható kell legyen a KH fuzzy távolság és az ezen alapuló rendezési reláció.

A módszer előnyös tulajdonsága szemléletessége és alacsony számításigénye. Részletes vizsgálata (pl. [H49][H51][H64]) arra a felismerésre vezetett, hogy egyes esetekben az eljárás érvénytelen fuzzy halmazt eredményezhet (ld. 9. ábra jobb oldali része).



9. ábra. Érvényes (bal oldal) és érvénytelen (jobb oldal) fuzzy halmazt eredményező KH interpoláció [H51]

A KH módszert egydimenziós antecedens alaphalmaz esetére dolgozták ki, azonban kiterjeszthető többdimenziós esetre is Minkowski (az esetek többségében euklideszi vagy Manhattan) távolságok alkalmazásával.

2.5.2. A KH interpoláció módosítása több szabály figyelembevételével

Publikálása óta a KH interpoláció számos olyan módosított változatát készítették el, amelyek nem csak a közrefogó két szabályt veszik figyelembe. Közös jellemzőjük, hogy a felhasznált szabályok számának növekedésével párhuzamosan javul az eljárás approximációs képessége.

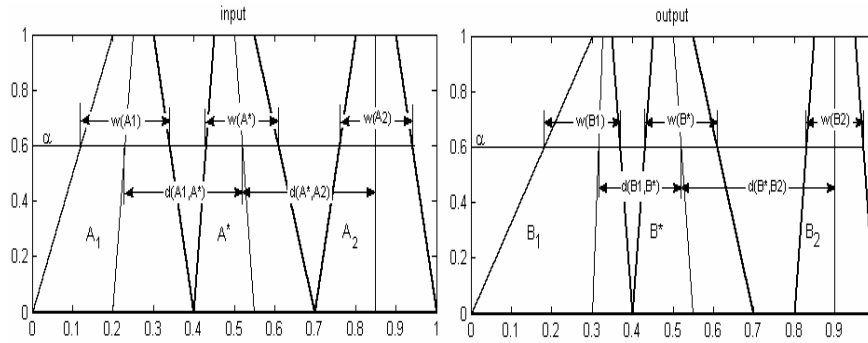
A Kóczy és Hirota [H45] irodalomban javasolt megoldása súlyozottan veszi figyelembe a szabályokat pl. a megfigyeléstől mért távolság reciprokának négyzetét alkalmazva súlytényezőként. Így a megfigyeléstől távolabb levő szabályok sokkal kisebb szerepet kapnak az eredmény számításában, mint a közvetlen környezetben elhelyezkedők [H45] [H73].

Tikk és szerzőtársai [H73] a Shepard interpoláció [H63] általánosításaként olyan képleteket javasoltak az α -vágatok végpontjainak számítására, amelyekben a távolság N . hatványát vették figyelembe, ahol N az antecedens dimenziók száma.

2.5.3. VKK módszer

A szerzők [H75] nevének (Vass, Kalmár és Kóczy) kezdőbetűi után VKK-nak elnevezett módszer kiküszöbölte a KH interpoláció egyik leggyakrabban előforduló problémáját, az érvénytelen következmény halmazokat azáltal, hogy új távolság mértéket vezetett be (ld. 2.2.2. szakasz), és az α -vágatok szélessége alapján végezte a számításokat. Bár eredetileg egybemenetű fuzzy rendszerekhez dolgozták ki, de a KH módszerhez hasonlóan kiterjeszthető többbemenetű rendszerekre is a Minkowski távolság bevezetésével.

A módszer csak kompakt CNF halmazok esetén alkalmazható. További hátránya, hogy nem képes kezelni azt az esetet, amikor egy fuzzy halmaz valamely α -vágatának szélessége nulla (pl. egyelemű halmaz). A KH szabály-interpolációhoz hasonlóan nem őrzi meg a szakaszonkénti linearitást, de az eltérés itt is viszonylag kis mértékű [H4].

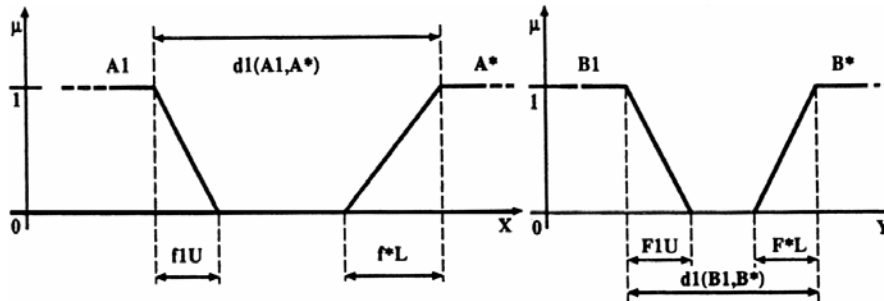


10. ábra. VKK módszer

2.5.4. Fuzzy jelleg megőrzésén alapuló szabály-interpoláció (GK módszer)

A Gedeon és Kóczy által publikált [H22] szabály-interpolációs eljárást trapéz alakú CNF halmazokra fejlesztették ki. A KH módszerhez hasonlóan a GK eljárás a következményt két közrefogó szabály segítségével becsüli. A GK módszer kidolgozása abból a feltételezésből indult ki, hogy számos alkalmazásnál az antecedens halmazok tartói jóval szélesebbek a megfigyelés tartójánál, és ezért a következmény egyes éleinek becsléséhez elegendő az azonos oldali (bal vagy jobb) megfigyelés él és az azzal szomszédos antecedens él figyelembevétele [H48].

Az eljárás a következmény alakjának és helyzetének számításához felhasználja a megfigyelés magja és a szomszédos antecedens halmazok magjai közötti távolságot (ld. $dI(A1, A^*)$ a 11. ábrán) és a halmazok fuzzy jellegét (ld. fIU, f^*L, FIU és F^*L a 11. ábrán).



11. ábra. Fuzzy jelleg és magtávolság [H48]

A következmény fuzzy jellegének (pl. F^*L) megállapítása során csak a megfigyelés azonos oldali fuzzy jellegét (f^*L) és a szomszédos antecedens fuzzy jellegét (fIU) veszik figyelembe. Az antecedens halmazok többi része nem játszik szerepet az eredmény meghatározásában. A módszer többdimenziós esetre is kiterjeszthető, ilyenkor a távolságokat euklideszi értelemben veszi figyelembe.

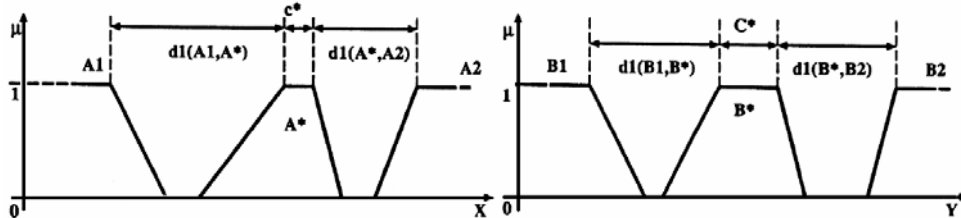
Bár a módszer nem α -vágat alapú és nincs közvetlen kapcsolatban a FERI-vel, de azzal összhangban van [H48].

2.5.5. Relatív fuzzy jelleg megőrzésén alapuló szabály-interpoláció (KHG módszer)

A KHG módszer (Kóczy, Hirota és Gedeon [H48]) az előző szakaszban ismertetett fuzzy jelleg megőrzésén alapuló szabály-interpoláció továbbfejlesztett változata. Az eljárás alkalmazhatóságát kibővítették általában a CNF halmazokra, így elődjével ellentétben crisp

halmazok kezelésére is képes. Továbbá a KHG módszer többbemenetű rendszerek esetében is alkalmazható.

A következmény helyzetét a FERI segítségével határozza meg az eljárás egy ún. különbözőségi mértéket bevezetve, ami a fuzzy jelleg és a két halmaz magtávolságának kapcsolatát jellemzi. A következmény magszélességét (C^*) (12. ábra) a konzekvens magtávolság ($d1(B1, B2)$) és az antecedens magtávolság ($d1(A1, A2)$) arányának és a megfigyelés magszélességének szorzatából számolja az eljárás.



12. ábra. Magtávolságok és magszélességek [H48]

A relatív fuzzy jelleg megőrzése azt jelenti, hogy a becsült következmény bal (jobb) oldali fuzzy jellegének és a vele szomszédos konzekvens fuzzy jellegnek az aránya megegyezik a megfigyelés bal (jobb) oldali fuzzy jellegének és a vele szomszédos antecedens fuzzy jellegnek az arányával.

2.5.6. Módosított α -vágat alapú szabály-interpoláció (MACI)

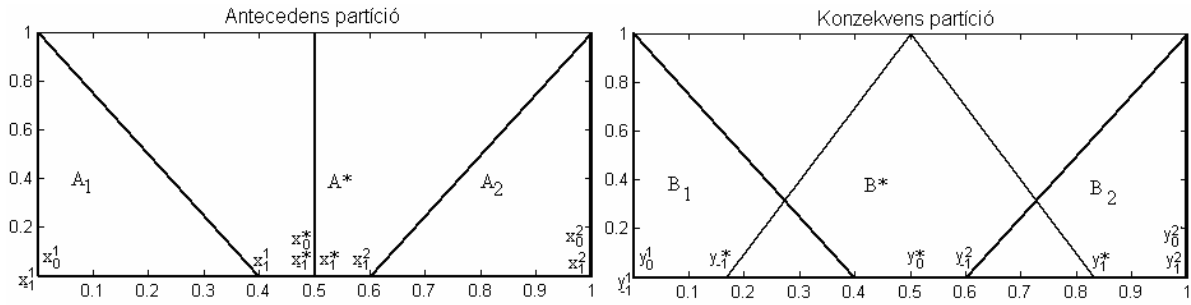
A Tikk és Baranyi által kidolgozott módosított α -vágat alapú szabály-interpoláció (MACI - Modified α -Cut based Interpolation) [H69] a Yam [H85] által publikált vektorreprezentációt alkalmazva egy olyan térbe transzformálja a számításokat, ahol kiküszöbölhető a fuzzy halmazként nem értelmezhető következmény.

A szemléletesség érdekében az eljárás illusztrálása az alábbiakban háromszög alakú fuzzy halmazokkal történik. A MACI szabály-interpoláció minden halmazt két vektor segítségével ír le, amelyek a referencia pont által meghatározott bal (alsó) és jobb (felső) él töréspontjainak abszcissza értékeit tartalmazzák. Sima tagsági függvények esetében egy sor α -vágatot készítenek, és ezek végpontjai kerülnek a vektorokba. Például a 13. ábra A_1 halmazára vonatkozóan ezek a következők

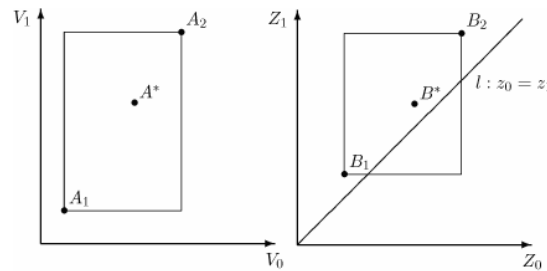
$$A_1^L = [x_{-1}^1, x_0^1] \quad , \quad (26)$$

$$A_1^R = [x_0^1, x_1^1] \quad . \quad (27)$$

A továbbiakban a számításokat a bal és jobb oldali élre egymástól függetlenül végzik. Háromszög alakú halmazokat alkalmazva mindkét esetben az éleket leíró vektorok egy síkbeli ponttal ábrázolhatóak (ld. 14. ábra). Ekkor a következmény érvényességének (fuzzy halmazként értelmezhető következmény) az a feltétele, hogy a B^* pont a B_1 és B_2 pontok által meghatározott téglalapon belülre essen, és az l vonal felett helyezkedjen el (ld. 14. ábra jobb oldali rész). Ezt az eljárást egy koordináta-transzformáció segítségével éri el. A transzformált térben a következményt a FERI segítségével számítják.



13. ábra. MACI-val előállított következmény egybementű rendszerénél

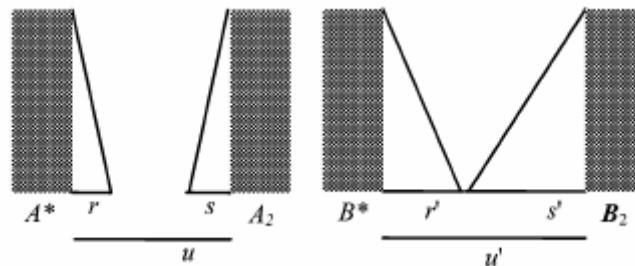


14. ábra. Az éleket leíró vektorok grafikus szemléltetése [H68]

A módszer legfontosabb pozitív tulajdonsága, hogy mindig érvényes fuzzy halmazt eredményez. Bár eredetileg egybemenetű rendszerekre fejlesztették ki, de kiterjeszthető többdimenziós bemenet esetére is [H69]. A módszer általános változata [H70] nem konvex fuzzy halmazok kezelésére is képes. Az eljárás csak akkor eredményez egyelemű fuzzy halmazt, ha a közrefogó szabályok konzekvenszei szintén egyeleműek [H55]. Bár a MACI interpoláció sem őrzi meg a szakaszonkénti linearitást, de az eltérés kisebb a KH módszernél tapasztaltnál [H69].

2.5.7. Javított többdimenziós módosított α -vágat alapú szabály-interpoláció (IMUL)

A Wong, Gedeon és Tikk [H80] által javasolt javított többdimenziós módosított α -vágat alapú szabály-interpoláció (IMUL - Improved MULTidimensional modified α -cut based interpolation) ötvözi a MACI és KHG módszerek előnyös tulajdonságait. A becült következmény magjának meghatározása a MACI-ból átvett karakterisztikus pontokat tároló vektorrepresentáció és koordináta-transzformáció segítségével történik. A következmény fuzzy jellegét (r' a 15. ábrán) a megfigyelés azonos oldali fuzzy jellege (r), valamint a szomszédos antecedens (s'/u') és konzekvens (s'/u') relatív fuzzy jellegek figyelembevételével számítják.



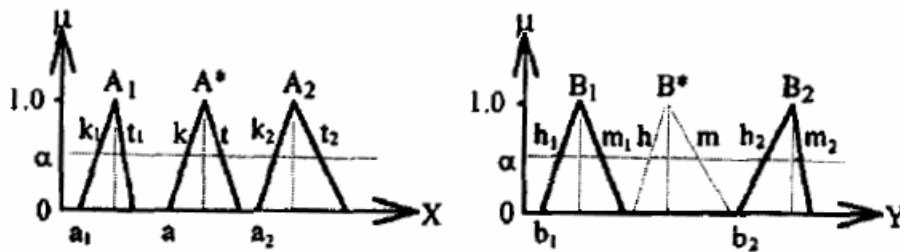
15. ábra. Az IMUL jelölésrendszere [H80]

Az eljárást eleve többdimenziós bemenet esetére dolgozták ki. További előnyös tulajdonsága, hogy mindig érvényes fuzzy halmazt eredményez. Hátrányként említhető, hogy nem őrzi meg a szakaszonkénti linearitást és nem alkalmazható tetszőleges halmazalakra.

2.5.8. HCL szabály-interpoláció

A Hsiao, Chan and Lee [H23] által kidolgozott szabály-interpoláció a KH eljárást ötvözi a halmazélek meredekségei közötti interpolációval. Az eljárás alapgondolata az antecedens élek meredekségei közötti kapcsolat megőrzése.

Első lépésként a tartó végpontjait a KH módszer segítségével határozzák meg. Ezt követően meghatározzák, hogy a két közrefogó szabály antecedens halmazai bal (jobb) élei meredekségeinek (k_1 és k_2 illetve t_1 és t_2 a 16. ábrán) milyen lineáris kombinációjaként állítható elő a megfigyelés bal (jobb) élének meredeksége (k és t). A becsült következmény bal (jobb) élének meredekségét (h és m) ugyanazon lineáris kombináció konzekvens meredekségekre (h_1 és h_2 illetve m_1 és m_2) történő alkalmazásával állítják elő úgy, hogy a kapott halmaz háromszög alakú és CNF jellegű legyen.



16. ábra. A HCL módszer jelölésrendszere [H23]

Az eljárás kis számításigényű és mindig érvényes fuzzy halmazt eredményez, azonban hátrányként említhető, hogy csak háromszög alakú CNF halmazok és csak egybemenetű rendszerek esetében alkalmazható.

2.5.9. Fuzzy interpoláció bizonytalan környezetben (FIVE)

A Kovács és Kóczy [H37] által kifejlesztett FIVE (Fuzzy Interpolation in the Vague Environment) módszer a fuzzy szabályok becslésének feladatát egy virtuális térbe, az ún. bizonytalan környezetbe [H28] helyezi át, aminek koncepciója az objektumok hasonlóságán illetve megkülönböztethetlenségén alapszik.

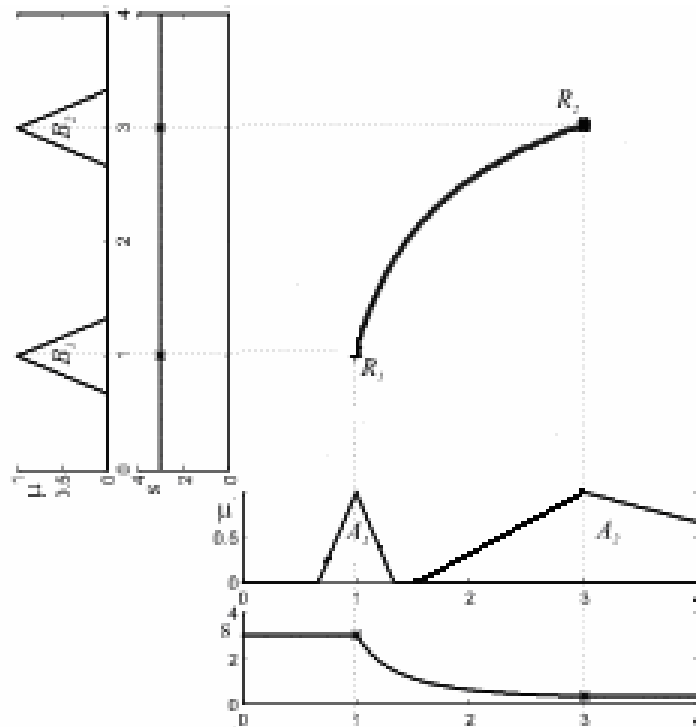
A bizonytalan környezetben két fuzzy halmaz hasonlóságát az adott környezetet leíró ún. skálafüggvénnyel súlyozott távolságuk jellemzi. A skálafüggvény a partíció fuzzy halmazai alakjának leírására szolgál. A bizonytalan környezet fogalmát a 4.4.1. szakasz tárgyalja részletesebben.

A módszer alkalmazásának az szabhat korlátot, hogy találunk-e úgy az antecedens, mint a konzekvens partíciókra egy-egy olyan közelítő univerzális skálafüggvényt, ami a teljes partíciót leírja olyan esetben is, amikor az nem Ruspini jellegű. A skálafüggvény választására háromszög és trapéz alakú halmazok esetén találunk megoldást a [H37] irodalomban.

Az antecedens és a konzekvens oldali univerzum bizonytalan környezetének meghatározását követően létrejön a szabálybázis saját bizonytalan környezete is. Ebben minden szabály egy-egy pontként ábrázolható. Amennyiben a megfigyelés egyelemű fuzzy halmaz, akkor bármely interpolációs vagy közelítő technika segítségével előállítható a keresett következmény, ami szintén egyelemű.

A feltétel és a következmény oldali bizonytalan környezetek előre elkészíthetők. Ez biztosítja a módszer gyorsaságát, mivel a rendszer működése közben csak a szabálybázist leíró pontok közötti interpolációt kell végrehajtani. Fuzzy megfigyelésnél az antecedens oldali és a bemenethez kapcsolódó bizonytalan környezetek összeolvasztása szükséges [H34].

A 17. ábra egydimenziós antecedens alaphalmaz és két szabály esetén ábrázolja a partíciókat, a skálafüggvényt és a két szabály közötti megfigyelések esetére érvényes interpolált pontok által meghatározott görbét egyelemű megfigyeléseket feltételezve.



17. ábra. Antecedens és konzekvens partíciók, skálafüggvények és következmény FIVE alkalmazása esetén

Az eljárás fontosabb előnyös tulajdonsága a gyorsaság és a többdimenziós antecedens tér kezelésének képessége. A FIVE hátránya, hogy a fuzzy megfigyelés kezelése csak bizonyos halmazalakok (háromszög és egyelemű) esetén megoldott, és a módszer nem őrzi meg a szakaszonkénti lineáris jelleget.

2.6. A fuzzy szabály-interpoláció általánosított módszertanát követő eljárások

A fuzzy szabály-interpolációs módszerek második csoportja két lépésben, a Baranyi, Kóczy és Gedeon [H4] által javasolt általánosított módszertan (GM) szerint állítja elő a következményt. A továbbiakban először áttekintem a GM koncepciót, majd néhány eljárást ismertetek, amelyek a módszertan két lépése során alkalmazhatóak. Végül röviden bemutatom az IGRV szabály-interpolációs módszert.

2.6.1. A fuzzy szabály-interpoláció általánosított módszertana (GM)

A Baranyi, Kóczy és Gedeon [H4] által kidolgozott általánosított fuzzy szabály-interpolációs módszertan (GM - Generalized Methodology of fuzzy rule interpolation) referencia pont alapú távolság mérést és rendezést alkalmaz (ld. 2.2.3. szakasz).

A GM moduláris felépítésének köszönhetően az általa megfogalmazott egyes részfeladatok megoldását különböző módszerekkel is megkísérélhetjük bizonyos konvencionális elemek (pl. referencia pont típusa) következetes alkalmazása mellett [H4]. E jellemzőt szem előtt tartva ismertetem a következőkben a módszertan fontosabb elemeit.

A GM két lépésben oldja meg a következmény előállításának feladatát. Az első lépésben egy új szabály interpolációjára kerül sor a megfigyeléssel azonos pozícióban, ami azt jelenti, hogy az új szabály minden antecedens nyelvi értékének referencia pontja egybeesik a neki megfelelő dimenzióbeli megfigyelést leíró fuzzy halmaz referencia pontjával. A módszertan első lépésében három feladatot kell megoldani. Ezek a következők.

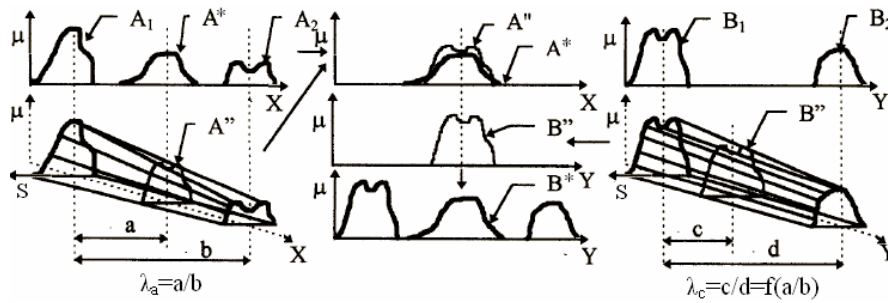
1. *Az interpolált szabály antecedens halmazai alakjának meghatározása.* A [H4] irodalomban például a testmetszés alapú módszert (SCM – Solid Cutting Method) [H3], a rögzített ponton alapuló eljárást (FPL – Fixed Point Law) [H18] és a rögzített érték elvén alapuló eljárást (FVL – Fixed Value Law) [H62] ajánlják e célra.
2. *A konzekvens halmazok helyzetének (referencia pontjainak) meghatározása.* Itt egy interpolációs/approximációs módszer szükséges, ami nem feltétlenül kell lineáris legyen. Például a fuzzy szabály-interpoláció alapegyenlete (FERI) [H45] is alkalmazható e célra. Egyes módszerek megkövetelik legalább két, a megfigyelést közrefogó szabály meglétét a számításokhoz, míg mások ennél több vagy az összes szabályt veszik figyelembe a pozíció meghatározása során.
3. *Az interpolált szabály konzekvens halmazai alakjának meghatározása.* A feladat hasonlóságából adódóan célszerű itt is az antecedens halmazalak számítására alkalmazott módszert használni.

A GM a második lépésben az újonnan létrehozott (interpolált) szabály segítségével állítja elő a következményt. Bár a megfigyelés és az új szabály antecedens halmazainak referencia pontjai azonosak minden bemeneti dimenzióban, a halmazalakok általános esetben eltérőek lehetnek. Ezért a következményt egy ún. egyszabályos, vagy más néven szabálmódosításon alapuló következtetési módszer segítségével állítjuk elő. Ezen módszerek alapgondolata az, hogy amilyen mértékben hasonlít a megfigyelés a szabály antecedensére ugyanolyan mértékben kell hasonlítson a következmény a szabály konzekvensére. Ez például az FPL [H18] és a szemantikai módosítási eljárások (SRM I és II, SRM – Semantic Revision Method) [H62] segítségével is megoldható.

Az értekezés első két tézisének alkotó halmaz-interpolációs és egyszabályos következtetési módszerek, valamint a segítségükkel létrehozott fuzzy szabály-interpolációs módszerek a GM koncepciót követik.

2.6.2. Testmetszés alapú interpoláció (SCM)

A Baranyi, Gedeon és Kóczy által kidolgozott testmetszés alapú interpoláció (SCM) - Solid Cutting Method) [H3] jól alkalmazható a GM első lépésében az antecedens és konzekvens halmazok alakjának meghatározására. Az eljárás alapgondolata az, hogy az A^* megfigyelést (ld. 18. ábra) közrefogó két antecedens halmaz (A_1 és A_2) referencia pontjainál egy-egy függőleges tengelyt határozzunk meg, majd ezek körül 90° -al elforgatjuk a két halmazt. Az ily módon előállított virtuális teret az S , X és μ ortogonális koordinátatengelyek határozzák meg. A két halmaz a $\mu x S$ síkkal párhuzamos síkokban fog elhelyezkedni (18. ábra).



18. ábra. Testmetszés alapú interpoláció [H3]

A következő lépésben egy felületet illesztünk a két halmaz körvonalára és tartójára, ami egy testet eredményez. Ezután a megfigyelés referencia pontjánál elmetszük a testet egy a μxS -el párhuzamos síkkal.

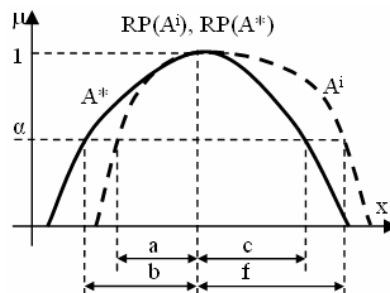
A metszetet visszaforgatva a μxX síkba megkapjuk a becsült szabály antecedensét. Az új szabály következményét analóg módon a két szomszédos konzekvens és a referencia pont ismeretében határozzuk meg. A módszer előnye, hogy bármilyen konvex halmazalak esetén alkalmazható, és mindig érvényes halmazalakot eredményez. Az SCM hátránya, hogy implementációja igen bonyolult és számításigényes akkor, amikor nem trapéz vagy trapézzal leírható (háromszög, singleton) alakúak az ismert fuzzy halmazok.

2.6.3. Hasonlóság megőrzési módszer (ST)

A hasonlóság megőrzési (ST - Similarity Transfer) [2] módszert Yan, Mizumoto és Qiao dolgozta ki. Az eljárás alapgondolata az, hogy az α -vágatokat a referencia pontnál két részre osztják, majd egy alsó és felső hasonlósági mértéket (b/a és f/c a 19. ábrán) számolnak a szabály antecedens (A_i) és a megfigyelés (A^*) között az azonos oldali α -vágatrészek arányaként.

A következményt α -vágatonként számítják. A végpontokat úgy határozzák meg, hogy a szabály konzekvenséhez mért alsó és felső hasonlósági értéke egyezzen meg a feltétel oldalon számolt értékekkel.

Az ST módszer a GM második lépésében alkalmazható a következmény kiszámítására. Az eljárás egyszerű, kis számításigényű és hatékonyan alkalmazható kis karakterisztikus pontszámmal leírható CNF halmazalakok esetén. Az eljárás gyenge pontjai a következők: nem képes egyelemű megfigyelések kezelésére, szubnormális és nem-konvex esetekben nem használható, és nem tartalmaz megoldást a valós alkalmazásokban legtöbbször előforduló többdimenziós antecedens alaphalmaz esetére.



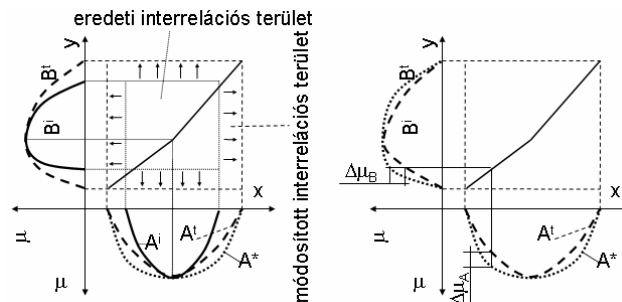
19. ábra. Az ST módszer jelölésrendszere

2.6.4. A lineáris revíziós elven alapuló módszerek

A Shen, Ding és Mukaidono által kidolgozott lineáris revíziós elven alapuló módszerek (pl. [H62]) jól alkalmazhatók a GM koncepción alapuló szabály-interpolációban. A jelen szakasz két ilyen eljárást (FPL és FVL) tekint át röviden.

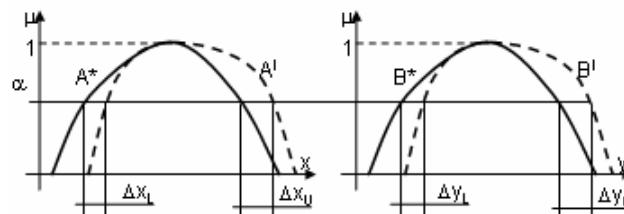
A **rögzített pont törvénye** (Fixed Point Law - FPL) [H62] egy interrelációnak nevezett függvénykapcsolat (Inter-Relation Function - IRF) segítségével egy egyértelmű leképezést definiál a szabály antecedens (A^i) és konzekvens (B^i) halmazának elemei között. Az így meghatározott téglalapot interrelációs területnek (InterRelation Area - IRA) nevezi. A továbbiakban, amennyiben ez szükséges, az IRA-t és az érintett két fuzzy halmazt arányosan módosítja ($A^i, B^i \rightarrow A^t, B^t$) úgy, hogy elérje az egybeesést az IRA megfelelő oldala és megfigyelés (A^*) tartója között (20. ábra). A továbbiakban az eljárás abból a feltételezésből indul ki, hogy az így kapott IRF azonos a megfigyelés és a következmény (B^*) közötti valós interrelációs kapcsolattal. Ezért a becült következmény pontjait úgy számolja ki, hogy az A^* minden kiválasztott halmazelemére kiszámolja a megfigyelés és az antecedens nyelvi értékhez való tartozás mértékét kifejező tagsági függvények különbségét, és ugyanezzel az értékkel módosítja a B^t konzekvens halmaz interreláció által meghatározott pontjának tagsági értékét (20. ábra).

Az eljárás előnyös tulajdonsága, hogy az IRF bevezetése egy jól hangolható technikát eredményez. Hátrányaként említhető az IRF használat és az IRA módosítás következtében megnövekedett számításigény, a bonyolultabb megvalósíthatóság és az a tény, hogy csak CNF halmazok esetén használható. Az FPL-t eredetileg egydimenziós esetre dolgozták ki, de a [H4] irodalomban találunk megoldási javaslatot a többdimenziós alkalmazásra is.



20. ábra. Interrelációs függvény és terület valamint a következmény számítása

A **rögzített érték törvénye** (Fixed Value Law - FVL) [H62] a tagsági függvény értékek (μ) mentén haladva határozza meg a következményt. Minden szükséges α szinten a megfigyelés (A^*) és az antecedens (A^i) halmazok azonos oldali vágat végpontjainak távolságából (Δx_L és Δx_U a 21. ábrán) kiindulva számítja ki, hogy milyen mértékben (Δy_L és Δy_U) szükséges eltolni a konzekvens vágat végpontjait a módosítás során.



21. ábra. Az FVL jelölésrendszere

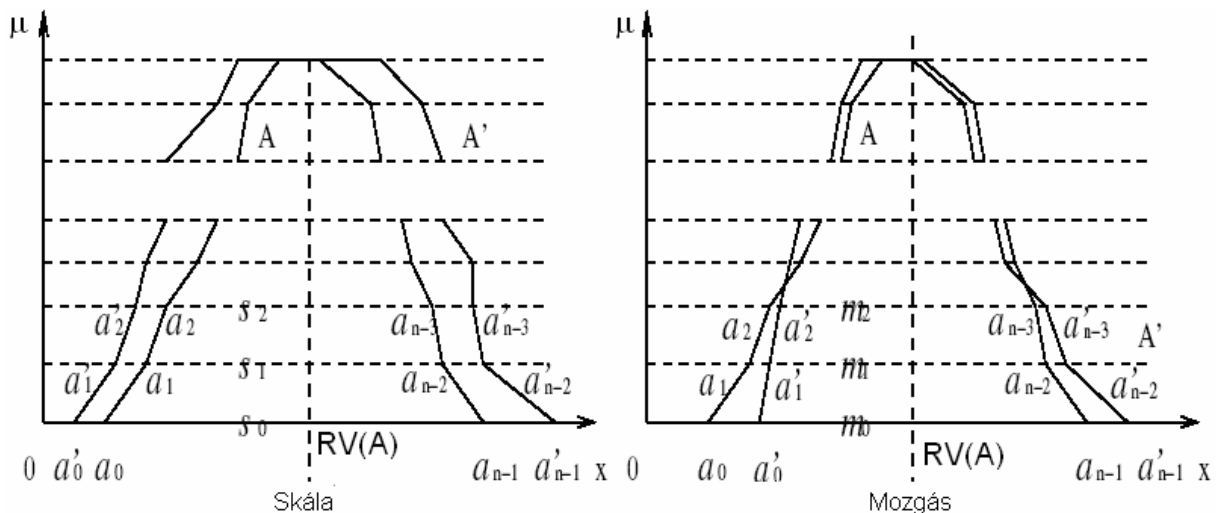
A módszer egyszerű, kis számításigényű és hatékonyan alkalmazható kis karakterisztikus pontszámmal leírható konvex és normál halmazalakok esetén. Hátrányos tulajdonsága az, hogy a kapott eltolás érték erősen függ attól, hogy a szabály antecedens és konzekvens milyen távolságban helyezkedik el saját alaphalmazának alsó és felső végpontjaitól. Könnyen előfordulhat olyan abnormális eredmény, amikor egy halmazelemhez több tagsági érték is tartozik. Az eljárás csak CNF halmazok esetén használható. Az FVL-t eredetileg egydimenziós esetre dolgozták ki, de a [H4] irodalomban találunk megoldási javaslatot a többdimenziós alkalmazásra is.

2.6.5. IGRV módszer

A Huang és Shen [H25] által kidolgozott IGRV (Interpolation with Generalized Representative Values) a GM által lefektetett kétlépéses megközelítést követi. Az eljárás először minden halmazhoz egy reprezentatív értéket (RV) számol, ami azonos a más módszerekben alkalmazott referencia ponttal. A halmazok közötti távolságot a továbbiakban az RV -k euklideszi távolságaként értékelik.

Az eljárás első lépésében az interpolált szabály antecedensét α -vágatonként határozzák meg úgy, hogy a következő két feltétel teljesüljön: (1) az antecedens halmaz reprezentatív értéke essen egybe a megfigyelés RV -jével, és (2) az antecedens halmaz α -vágatainak végpontjai olyan arányban osszák fel a szomszédos antecedens halmazok azonos (bal vagy jobb) oldali α -vágat végpontjai közötti távolságot, mint amilyen arányban a megfigyelés RV pontja felosztja ezen halmazok RV -i közötti távolságot. A szabály konzekvens halmazát ugyanezen elv alkalmazásával számítja az eljárás.

Az IGRV második lépésében a megfigyelés és az új szabály antecedens halmaza közötti hasonlóságot ún. skála- és mozgástranszformációkkal (22. ábra) jellemzik. A következményt úgy állítják elő, hogy ugyanezen transzformációkat hajtják végre a konzekvens halmazon, és alkalmazzák az érvényességet (fuzzy halmazként értelmezhető következmény) biztosító megkötéseket.



22. ábra. Skála- és mozgástranszformációk [H25]

A módszert eredetileg sokszög alakú (szakaszonként lineáris tagsági függvényű) fuzzy halmazokhoz dolgozták ki, de kellő számú α -vágat esetén elméletileg más halmazalakokra is kiterjeszthető, azonban más α -vágat alapú szubnormális halmazok kezelésére nem képes. Az IGRV képes kezelni a többdimenziós bemenetet is.

2.7. Összegzés

A jelenleg ismert ritka szabálybázist is kezelni képes fuzzy következtetési módszerek gyakorlati alkalmazhatósága általában korlátozott. Az ismert eljárások jellegzetesen gyenge pontjai az alábbiak.

- A legtöbb módszer csak egydimenziós bemenet esetére lett kidolgozva.
- Egyes módszerek csak bizonyos halmazalakok (tagsági függvény típusok) esetén használhatóak.
- Az extrapolációs képesség hiánya. A legtöbb módszer megköveteli legalább két olyan szabály meglétét, amelyek antecedensei minden dimenzióban közrefogják a megfigyelést.
- Különböző magasságú halmazok kezelésének hiánya a legtöbb módszernél.
- Az egyes következtetési eljárások leírása a fellelhető irodalomban gyakran nem eléggé részletes, csak bizonyos esetekre kidolgozott.
- Nem áll rendelkezésre hozzáférhető szoftveres implementáció, ami lehetővé tenné az egyes módszerek könnyű gyakorlati alkalmazását és összevetését.

A megfelelő következtetési módszer választása mellett a fuzzy rendszerek kialakításának jelentős szereppel bíró témaköre a mintaadatok alapján történő automatikus rendszergenerálás. A ritka szabálybázis adatokból történő előállítására szolgáló jelenleg ismert módszerek nehézkesek, bonyolultak, illetve gyakran csak igen hosszadalmas hangolási folyamat után eredményeznek olyan fuzzy rendszert, ami elfogadható pontossággal képes modellezni az összetartozó bemenő-kimenő adathalmazok által leírt jelenséget.

Összegzésként megállapítható, hogy a tématerület jelentős elméleti háttere és a publikációk nagy száma ellenére is szükséges lehet további kutatás, újabb szabály-interpolációs következtetési módszerek, illetve automatikus szabálybázis generálási eljárások kidolgozása.

3. Kutatási célok

A ritka szabálybázisra épülő fuzzy rendszerek gyakorlati megvalósításának kulcskérdése a megfelelő következtetési módszer választása, illetve a rendszer paramétereinek „tanító” mintaadathalmaz alapú meghatározása. Ezen követelmények figyelembevételével határoztam meg kutatási célkitűzéseimet, amelyek az alábbi pontokban foglalhatók össze.

1. Új fuzzy következtetési módszerek kidolgozása az eddigi eljárásokkal nem, vagy nehezen kezelhető esetekhez (pl. eltérő magasságú halmazok egy partíción belül vagy partícióra jellemző halmazalak megőrzése). A fuzzy szabály-interpoláció általánosított módszertanát [H4] követve az ebben a pontban meghatározott célkitűzést két részfeladatra bontottam. Ezek a következők:
 - a. Új halmaz-interpolációs módszerek kidolgozása az interpolált szabály antecedens és konzekvens halmazainak számításához.
 - b. Új szabálmódosításon alapuló egyszabályos következtetési módszerek kidolgozása a következmény számításához.
 - c. Számítógépes eljárások kidolgozása és implementálása az új módszerekhez.
2. Olyan módszerek kidolgozása, amelyek alkalmasak ritka szabálybázis létrehozására az összetartozó be- és kimenetet leíró mintaadatok alapján. Itt az alábbi két részfeladatot határoztam meg:
 - a. Ismert, klaszterezésen alapuló módszerek továbbfejlesztése, új eljárás kialakítása.
 - b. Új megközelítésen alapuló módszerek kidolgozása.
 - c. Számítógépes eljárások kidolgozása és implementálása az új módszerekhez.
3. Szoftverrendszer fejlesztése, amely lehetővé teszi a különböző fuzzy szabály-interpoláción alapuló következtetési módszerek egységes keretek között történő alkalmazását és összevetését.
4. A kidolgozott következtetési és modellgenerálási módszerek gyakorlati feladatokra történő alkalmazhatóságának példákon keresztüli igazolása.

4. Fuzzy halmaz-interpoláció – javasolt módszerek

A fuzzy halmaz-interpoláció célja egy új fuzzy halmaz előállítása egy alaphalmaz egy megadott pontjában (interpolációs pont) úgy, hogy az új halmaz referencia ($RP(A^i)$) pontja essen egybe az interpolációs ponttal (x^i). A halmaz-interpolációs módszerek tárgyalása során csak kompakt fuzzy halmazokkal foglalkozom, azaz olyan nyelvi értékekkel, amelyeknek korlátos a tartója [H26].

Feltételezzük, hogy az $X = [x_{\min}, x_{\max}]$ alaphalmazon véges számú fuzzy halmazzal ($A_j, j \in \{1, \dots, n\}$) rendelkezünk. Ezek a fuzzy halmazok nem feltétlenül biztosítják az alaphalmaz teljes lefedettségét, azaz ε értéke lehet 0 a lefedettség mértékét definiáló (10) kifejezésben.

A halmaz-interpoláció alapjául az a feltételezés szolgál, hogy a partíció nyelvi értékeinek alakjai nem véletlenszerűek, hanem valamilyen szabályszerűség jellemzi őket [H3][H26]. Az újonnan létrehozott fuzzy halmaznak illeszkednie kell ehhez a rendszerhez, azaz az x^i ($RP(A^i)$) interpolációs pont és egy ismert A_j halmaz $RP(A_j)$ referencia pontjának egyezése esetén az interpolációval előállított A^i fuzzy halmaz meg kell egyezzen az adott ismert A_j halmazzal

$$A^i = A_j \text{ ha } RP(A^i) = RP(A_j), \forall j \in \{1, \dots, n\}, \quad (28)$$

ahol n a partíció halmazainak száma. Az alábbiakban ismertetésre kerülő mindhárom halmaz-interpolációs eljárásnál a mag középpontjának megfelelő alaphalmaz elemet (RP_{CC} az 1. ábrán) fogom referencia pontként alkalmazni. Az egyszerűbb jelölés érdekében a referencia pont típust jelölő CC alsó indexet elhagyom.

Az új halmaz előállításának feladatát interpolációnak nevezzük, amennyiben az ismert nyelvi értékek referencia pontjai által meghatározott intervallumon belül dolgozunk ($RP(A_1) \leq RP(A^i) \leq RP(A_n)$), és extrapolációnak, amennyiben az adott intervallumon kívül kell létrehozni az új nyelvi értéket ($RP(A^i) < RP(A_1)$ vagy $RP(A^i) > RP(A_n)$). Bár az alábbiakban ismertetésre kerülő mindhárom módszer alkalmas extrapolációs feladatokra is, de a továbbiakban az egyszerűség kedvéért a szövegben csak interpolációként hivatkozom az eljárásokra.

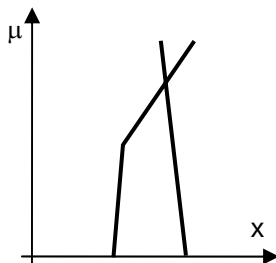
A halmaz-interpolációs módszerek kidolgozása során alapvetően két alkalmazási területet tartottam szem előtt. Ezek (1) az új fuzzy halmazok előállítása az interpolált szabály antecedens és konzekvens része számára az általánosított módszertan [H4] első lépésében, valamint (2) az új fuzzy halmaz előállításával történő szabálybázis generálás (RBE-DSI [S22], 8.1.6. fejezet). A módszerek közös jellemzője, hogy egyszerre egy dimenzióval foglalkoznak, a különböző dimenziókban a számítások egymástól függetlenül történnek, és ezért akár párhuzamosan is végrehajthatók, amennyiben ezt az implementáció és a platform lehetővé teszik.

Az általam kifejlesztett halmaz-interpolációs módszerek összehasonlítása és értékelése érdekében egy öt pontból álló követelményrendszert állítottam össze a szakirodalomban a fuzzy szabály-interpolációs eljárásokkal szemben támasztott elvárások (pl. [H4][H26][H69]), valamint saját ötletek alapján. Ezek a következők.

1. *Abnormális (ld. pl. 23. ábra) halmazalak elkerülése* (pl. [H4][H26][H69]). A halmaz-interpolációval előállított új nyelvi érték legyen mindig egy érvényes fuzzy halmaz. Ezt a követelményt α -vágatok segítségével lehet a legegyszerűbben leírni

$$\inf\{[A^i]_{\alpha_1}\} \leq \inf\{[A^i]_{\alpha_2}\} \leq \sup\{[A^i]_{\alpha_2}\} \leq \sup\{[A^i]_{\alpha_1}\} \quad \forall \alpha_1 < \alpha_2 \in [0,1] \quad , \quad (29)$$

ahol $\inf\{[V]_{\alpha}\}$ és $\sup\{[V]_{\alpha}\}$ a V fuzzy halmaz α -vágatának alsó illetve felső végpontja.



23. ábra. Egymást metsző élekkel rendelkező érvénytelen fuzzy halmaz

2. *Kompatibilitás a partícióval* (interpolációs képesség). Ez az elvárás azt fejezi ki, hogy amennyiben az x^i interpolációs pont megegyezik a partíció valamely ismert halmazának referencia pontjával, akkor az új halmaz azonos kell legyen az adott nyelvi értékkel (28).
3. *Alakzat megőrző képesség*. Ha a partíció minden halmaza azonos alakzat típusba tartozik, akkor elvárható, hogy az interpolált halmaz is őrizze meg ezt a tulajdonságot. A gyakorlatban ezt úgy kell értelmezni, hogy ha például az ismert halmazok szakaszonként lineárisak és a megfelelő töréspontok mindannyiuknál azonos α -szinteken helyezkednek el, akkor az interpolált halmaznak is illeszkednie kell ebbe a rendszerbe (pl. [H4]).
4. *Halmazalaktól független alkalmazhatóság*. Az eljárás minden érvényes fuzzy halmazalak esetén képes kell legyen új nyelvi érték interpolálására.
5. *Extrapolációs képesség*. Az eljárás olyan esetekben is alkalmazható kell legyen, amikor az x^i interpolációs pont az ismert nyelvi értékek referencia pontjai által meghatározott intervallumon kívülre esik ($x^i \notin [RP(A_1), RP(A_n)]$).

4.1. Nyelvi értékek eltolásának elve

Az általam kidolgozott első két halmaz-interpolációs eljárás (FEAT-p és FEAT-LS) a nyelvi értékek eltolásának elvét bevezetve és alkalmazva kínál megoldást az új nyelvi érték meghatározására az interpolációs pontban.

A nyelvi értékek eltolási elvének [S14] az az alapgondolata, hogy meghatározzuk az alaphalmaz összes ismert fuzzy halmazának referencia pontját, majd eltoljuk őket vízszintesen úgy, hogy referencia pontjuk essen egybe az interpolációs ponttal (24. ábra jobb oldal). Ezután az egymást átfedő alakzatok felhasználásával meghatározzuk az új fuzzy halmazt és nyelvi értéket. A számítások során nem csak az interpoláció helyét közrefogó két fuzzy halmazt, hanem az alaphalmazon értelmezett összes ismert halmazt figyelembe vesszük.

Így vélhetően jobb halmaz-approximáció érhető el, és a számítási igény sem növekszik jelentős mértékben. Továbbá kis halmazszámú partíciókat feltételezve (általában max. 11 [H1][S7]) általában egyszerűbb az összes halmazt felhasználni, mint kikeresni a két közrefogó szomszédosat. Mindemellett előnyt jelenthet, hogy a halmaz-interpolációs eljárás így egyben extrapolációra is alkalmazhatóvá válik.

A nyelvi értékek eltolásával dolgozó halmaz-interpolációs eljárások alkalmazásánál módosítanunk kell az alaphalmaz alsó és felső korlátainak ($x_{\min}$, $x_{\max}$) fogalmát. Hagyományosan ezek a korlátok amellet, hogy meghatározzák a be- illetve kimenet szélsőértékeit, az adott partícióban elhelyezkedő fuzzy halmazok tartóinak és ezáltal összes α -vágatának is határt szabnak

$$\inf \{ [A]_{\alpha} \} \geq x_{\min}, \forall \alpha \in [0,1], \quad (30)$$

$$\sup \{ [A]_{\alpha} \} \leq x_{\max}, \forall \alpha \in [0,1]. \quad (31)$$

A halmazalakok között szabályszerűséget feltételezve elvárható, hogy abban az esetben, ha a partíció minden halmaza a referencia pont mindkét oldalán tartalmaz alaphalmazbeli elemeket (pl. a halmazalak egy egyenlő oldalú háromszög vagy egy trapéz)

$$\inf \{ [A]_0 \} < RP(A) < \sup \{ [A]_0 \}, \quad (32)$$

akkor ez a tulajdonság azon halmaz esetében is teljesüljön, mely referencia pontja egybeesik az alaphalmaz valamelyik korlátjával. Ez a követelmény azonban csak akkor teljesíthető, ha megengedjük, hogy a halmazalak egy része túlnyúljon a korábbi korlátokon. Természetesen a referencia pont nem kerülhet az $[x_{\min}, x_{\max}]$ intervallumon kívülre. Ezen engedménynek csak a halmaz-interpoláció szempontjából van lényegi szerepe, ezért alkalmazása esetén el kell kerülni a nemkívánt mellékhatásokat például a defuzzifikáció során.

4.2. Polár-vágaton alapuló halmaz-interpoláció (FEAT-p)

A polár-vágaton alapuló halmaz-interpoláció [S16] (FEAT-p – Fuzzy sEt interpolAtion based on linguistic Term shifting and polar cuts) a nyelvi értékek eltolásával és az ún. polár-vágatok segítségével oldja meg a halmaz-interpoláció feladatát.

Az új nyelvi érték alakjának számításához bevezetjük egy fuzzy halmaz polár-vágatának fogalmát. Ennek érdekében egy poláris koordináta-rendszert illesztünk a halmazhoz (ld. 25. ábra) úgy, hogy a középpont essen egybe a nyelvi érték referencia pontjának megfelelő halmazelemmel. A polár-vágat

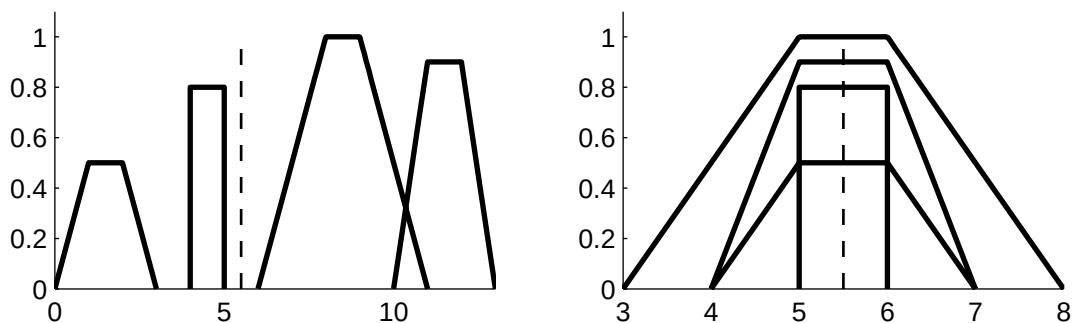
$$[A]_{\theta} = \{\rho, \theta\} \quad (33)$$

egy érték pár, ami a halmazalak egy pontját írja le annak poláris távolságával (ρ) és poláris szögével (θ) (25. ábra).

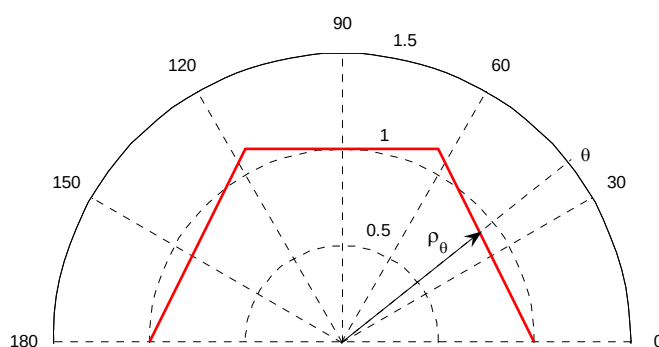
A FEAT-p halmaz-interpolációs módszer abból a feltevésből indul ki, hogy minden konvex fuzzy halmaz felbontható polár-vágatokra, és leírható polár-vágatok összességéként

$$A = \bigcup_{\theta \in [0, \pi]} [A]_{\theta}. \quad (34)$$

Az új interpolált fuzzy halmaz alakja az egyes polár-vágatokra végzett számítások uniójaként áll elő.



24. ábra. Eredeti fuzzy halmazok (nyelvi értékek) és az eltolás eredménye

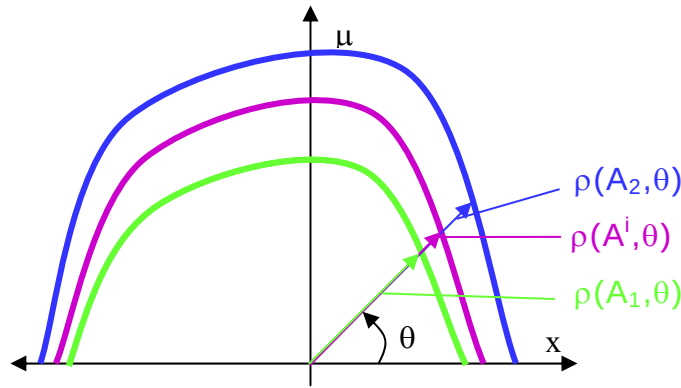


25. ábra. Polár-vágat

Az új fuzzy halmaz minden polár-vágatát úgy kapjuk meg, hogy a meglévő (egymásra tolt) halmazok azonos polár szög alatti polár távolságainak súlyozott átlagát képezzük (ld. 26. ábra)

$$\rho(A^i, \theta) = \begin{cases} \rho(A_j, \theta), & \text{ha } d(A^i, A_j) = 0, j \in [1, n] \\ \frac{\sum_{j=1}^n w_j \cdot \rho(A_j, \theta)}{\sum_{j=1}^n w_j}, & \text{ha } d(A^i, A_j) \neq 0, j \in [1, n] \end{cases} \quad (35)$$

ahol $d(X, Y) = |RP(X) - RP(Y)|$ a két fuzzy halmaz referencia pontjai között értelmezett euklideszi távolság, w_j a j -edik fuzzy halmaz súlytényezője (ld. (36)).

26. ábra. Egyásra tolt halmazok (A_1, A_2) és interpolált halmaz (A^i)

A számítások során alkalmazott poláris szintek gyűjteményét úgy kell meghatározni, hogy a három kulcsfontosságú szögérték mindig benne legyen. Ezek a 0 , a $\pi/2$ és a π szögek.

Természetes követelmény, hogy azok a halmazok, amelyek eredetileg az interpolációs pont közelében voltak nagyobb hatást gyakoroljanak az eredményre, mint azok, amelyek a partíció távolabbi részeiben helyezkedtek el. A súlytényező feladata ezen igény kielégítése. A súlytényező értékét az érintett halmaz és az interpolációs pont közötti távolság függvényében határozzuk meg. A legegyszerűbb súlytényező a távolság reciproka ($p=1$ és $\lambda=1$ a (36) képletben), de a szakirodalomban több távolságfüggő súlyfaktort is találhatunk. Ilyen például [H85] a

$$w_j = \frac{1}{\lambda \cdot d(A^i, A_j)^p}, \quad (36)$$

ahol λ egy pozitív konstans. A halmaz-interpolációnál értékét 1 -re választjuk, mivel az a számítások során úgyis kiesik. Kóczy és Hirota [H45] súlytényezőként a távolság reciprokának négyzetét ajánlja ($p=2$ és $\lambda=1$ az (36) képletben) az m -edik hatványon, ahol m az antecedens dimenziók száma. Yam és Kóczy [H85] a kiterjeszthetőségi függvény fogalmát bevezetve három súlytényező típust is javasol. Az első kettő a (36)-ból előállítható a $p=1$, $\lambda=1$ illetve a $p=2$, $\lambda=1$ paraméter értékekkel, míg a harmadik a

$$w_j = e^{-\lambda \cdot d(A_j^*, A_j)} \quad (37)$$

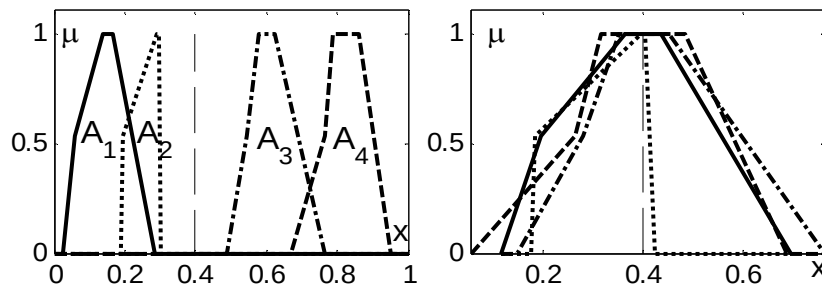
képlettel írható le, ahol λ egy pozitív állandó, ami meghatározza az effektív kiterjeszthetőségi távolságot. A súlyfüggvények paraméterei további hangolási lehetőséget biztosíthatnak a rájuk épülő halmaz-interpolációs eljárásoknak.

Az eljárás egyik lényeges előnyös tulajdonsága, hogy bármilyen érvényes halmazalak esetén alkalmazható, valamint nem szükséges, hogy az ismert fuzzy halmazok azonos alakúak és magasságúak legyenek, azaz egy vagy több halmaz szubnormális is lehet. Továbbá az eljárás egyaránt alkalmazható interpolációra és extrapolációra is. A módszer hátrányos tulajdonsága, hogy mivel nem őrzi meg a szakaszonkénti lineáris jelleget, így az eredmény meghatározásához nagyszámú polár-vágatra van szükség, ami növeli a számítási igényt. Összességében elmondható, hogy a módszer kielégíti az 1, 2, 4 és 5 elvárásokat.

4.3. Legkisebb négyzetek elvén alapuló halmaz-interpoláció (FEAT-LS)

A legkisebb négyzetek elvén alapuló halmaz-interpoláció [S1] (FEAT-LS – Fuzzy sEt interpolAtion based on the method of Least Squares) a 4.1. fejezetben bemutatott nyelvi értékek eltolásával, valamint a legkisebb négyzetes eltérésű halmazalak megkeresésével oldja meg a halmaz-interpoláció feladatát.

Gyakran előfordul az alkalmazásokban, hogy a partíció összes fuzzy halmaza ugyanazon alakzat-típushoz (pl. sokszög, trapéz, háromszög, singleton) tartozik, valamint szakaszonként lineáris tagsági függvények esetén a töréspontok (karakterisztikus pontok) azonos α -szinteken helyezkednek el minden halmaznál. Ezekben az esetekben a halmaz-interpolációtól elvárjuk, hogy megőrizze eme jellegzetességet.



27. ábra. Azonos alakzat típusba tartozó ismert fuzzy halmazok és az eltolás eredménye. Az interpoláció helyét függőleges szaggatott vonal jelöli.

A legkisebb négyzetek elvén alapuló halmaz-interpolációs eljárást kifejezetten a fent ismertetett körülményekhez fejlesztettem ki. A FEAT-p módszerhez hasonlóan alkalmazza a nyelvi értékek eltolásának elvét, valamint az összes ismert fuzzy halmazt felhasználja a számítások során. Így egyaránt alkalmazható interpolációs és extrapolációs feladatokra is.

Az eljárás első lépéseként az univerzumon definiált összes ismert fuzzy halmazt vízszintesen eltoljuk úgy, hogy referencia pontjuk essen egybe az interpolációs ponttal. A 27. ábra bal oldali része a négy ismert fuzzy halmazt mutatja be, míg a jobb oldali rész az eltolás eredményét ábrázolja. Az interpolációs pont az $x'=0.4$ -es értéknél helyezkedik el.

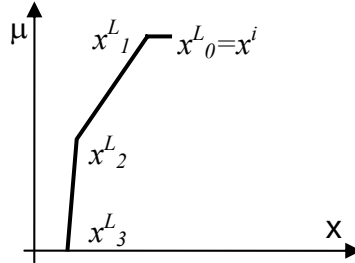
A számítások során az új halmaz alakját definiáló karakterisztikus pontokat α -vágatonként határozzuk meg úgy, hogy hozzájuk viszonyítva az ismert halmazok azonos α -szintű pontjai a lehető legkisebb négyzetes eltérést mutassanak.

A halmazalakot a referencia pontnál két részre (élre), egy bal oldalira és egy jobb oldalira bontjuk úgy, hogy mindkét él tartalmazza a referencia pontot és az alakzatot a saját oldalán definiáló töréspontokat. Ezután a két élre külön-külön, csak az adott oldal töréspontjainak megfelelő α -szinteken végezzük el a számításokat. Ez a megközelítés kis számításigényű annak következtében, hogy mindkét oldalon csak a feltétlenül szükséges α -szintszámmal kell dolgoznunk. Az így meghatározott pontok lesznek az új halmaz karakterisztikus pontjai. A tagsági függvény fennmaradó részét úgy kapjuk meg, hogy a karakterisztikus pontokat egyenesekkel kötjük össze. A számítás menetét és az alkalmazott jelölést a továbbiakban a bal oldali él esetére ismertetjük. Ennek alapján a jobb oldali él meghatározása könnyen levezethető.

A számítások során csak a karakterisztikus pontok abszcisszáinak meghatározása szükséges, az ordináta értékek azonosak az ismert halmazok megfelelő karakterisztikus pontjainak ordináta értékeivel. A módszer a legnagyobb α -szintű vágattól indul az $\alpha=0$ szint irányába. A

két él közös pontjaként megjelenő referencia pont abszcisszáját az interpoláció helye ($x_0^L = x^i$), míg tagsági értékét a partícióra jellemző legmagasabb α -szint határozza meg ($\mu_{A^i}(x_0^L) = \sup_{x \in X} \{\mu_{A^i}(x)\}$). Normális halmazok esetén ennek értéke 1. A referencia pontra $k=0$

index segítségével hivatkozunk, míg a többi pont sorszáma az α -szint szerinti csökkenő sorrendben haladva növekszik egyesével (ld. 28. ábra).



28. ábra. Bal oldali él és karakterisztikus pontok

A bal oldali él minden további karakterisztikus pontjának abszcisszáját úgy határozzuk meg, hogy minden α -szinten minimális legyen az ismert halmazok karakterisztikus pontjai és az interpolált halmaz karakterisztikus pontja között mért súlyozott négyzetes eltérések összege. A bal oldali él egy pontjára felírva a súlyozott négyzetes eltérések összegét az alábbi egyenletet kapjuk

$$Q_l^L = \sum_{j=1}^n w_j \cdot (x_{jl}^L - x_l^{Li})^2, \quad (38)$$

ahol Q_l^L a bal oldali él l . pontjához kapcsolódó négyzetösszeg, w_j a j -edik ismert fuzzy halmazhoz társított súlytényező, n az ismert fuzzy halmazok száma, x_{jl}^L a j -edik halmaz bal éle l -edik pontjának abszcisszája és x_l^{Li} az interpolált halmaz bal éle j -edik pontjának abszcisszája. Az egyenletben szereplő x értékek az eltolts halmazokra vonatkoznak.

Az x_l^{Li} értékekkel szemben két követelményt támasztunk, a Q_l^L négyzetösszeg minimális kell legyen, és teljesülnie kell a következő abnormalitás elkerülési feltételnek

$$x_l^L \leq x_{l-1}^L \quad l = 1, \dots, n_j^L - 1, \quad (39)$$

ahol n_j^L a bal oldali él karakterisztikus pontjainak száma (a referencia pontot is beleértve). Feltételezve, hogy a partíció minden nyelvi értékére teljesül a (39) megkötés, könnyen bizonyítható, hogy az a legkisebb súlyozott négyzetes eltéréssel számított pontokra is teljesül. Ennek megfelelően a (38) négyzetösszeg minimalizálásával az alábbi képlethez jutunk:

$$x_l^L = \begin{cases} x_{jl}^L & \text{ha } d(A_j, A^i) = 0, j = 1, \dots, n \\ \frac{\sum_{j=1}^n w_j \cdot x_{jl}^L}{\sum_{j=1}^n w_j} & \text{ha } d(A_j, A^i) \neq 0, j = 1, \dots, n. \end{cases} \quad (40)$$

Az új halmaz minden karakterisztikus pontjának abszcisszáját a figyelembe vett fuzzy halmazok megfelelő pontjai abszcisszáinak súlyozott átlagaként számítjuk. A módszer attól

válíkat interpolációvá, hogy abban az esetben, ha az interpolációs pont abszcisszája azonos valamelyik eredeti halmaz referencia pontjának abszcisszájával, akkor az előállított új halmaz is azonos lesz az adott halmazzal. Ennek biztosítása érdekében definiáltuk a (40) képletet kéttagúként, továbbá a w_j súlytényezőt úgy választjuk meg, hogy a nyelvi érték és az interpolációs pont közötti $d(A_j, A^i)$ távolság csökkenésével, a (40) képlet első tagjának határértéke tartson x_{ji}^L -hez. Az előző szakaszban ismertetett súlytényezők (pl. (36) és (37)) itt is alkalmazhatók.

A FEAT-LS eljárás legfontosabb előnyös tulajdonsága alacsony számításigénye és alakzat megőrző képessége. Bár alapvetően arra az esetre lett kidolgozva, amikor a partíció minden fuzzy halmaza azonos alakzat típusba tartozik, de mivel a módszer α -vágat alapú, így vegyes alakzatok esetén is képes értelmezhető eredmény előállítására. Az egyetlen megkötés az, hogy minden halmaz magassága azonos kell legyen. Összességében kijelenthető, hogy a módszer teljesíti az 1, 2, 3, 4 és 5 elvárásokat.

4.4. Halmaz-interpoláció bizonytalan környezetben

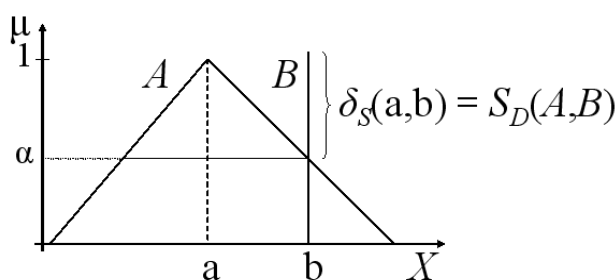
A fejezet első részében röviden áttekintem a bizonytalan környezet fogalmát és azt meghatározó skálafüggvény előállítási módozatait, majd ezt követően ismertetem az általam kidolgozott VESI [S2] eljárást.

4.4.1. Bizonytalan környezet

A Klawonn által javasolt [H28] bizonytalan környezet (VE - Vague Environment) fogalom egy univerzum elemeinek hasonlóságán illetve megkülönböztethetlenségén alapszik. A megközelítés alapgondolata az, hogy a bizonytalan környezetben két (x_1, x_2) értéket ε mértékben megkülönböztethetetlennek (hasonlónak) tekintünk, amennyiben távolságuk legfeljebb ε értékű ($\delta_s(x_1, x_2) \leq \varepsilon$). A távolságot nem euklideszi értelemben számítjuk, hanem a bizonytalan környezetet jellemző ún. skálafüggvény ($s(x)$) segítségével határozzuk meg

$$\delta_s(x_1, x_2) = \left| \int_{x_2}^{x_1} s(x) dx \right|. \quad (41)$$

Ez egy súlyozott távolság érték, ahol a súlytényezőt skála faktornak vagy skála függvénynek nevezzük.



29. ábra. Egy háromszög és egy egyelemű fuzzy halmaz közötti diszkonzisztencia mértéke [H34]

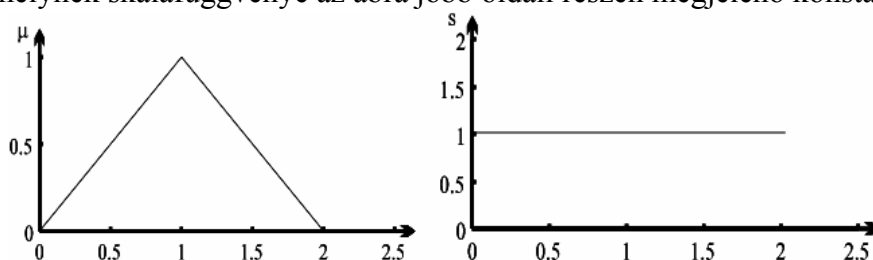
A fuzzy halmazok és a bizonytalan környezet közötti kapcsolatot a fuzzy halmazok közötti hasonlóság egyik értékszámaként is alkalmazott $S_D(A,B)$ diszkonzisztencia mérték [H74] és a skálafüggvénnyel súlyozott $\delta_s(a,b)$ távolság azonossága teremti meg (ld. 29. ábra) [H34]. A bizonytalan környezet sajátosságainak következtében csak egy egyszerű fuzzy halmaz és egy egyelemű fuzzy halmaz közötti távolság értelmezhető a segítségével. Ennek tudható be, hogy

a FIVE szabály-interpolációs eljárás első változata [H37] csak singleton alakú megfigyelésekkel dolgozott.

A bizonytalan környezet alkalmazásának kulcskérdése egy olyan általános skálafüggvény készítése, amely képes leírni a partíció összes nyelvi értékének alakját egyetlen függvény segítségével. A Klawonn [H28] által javasolt

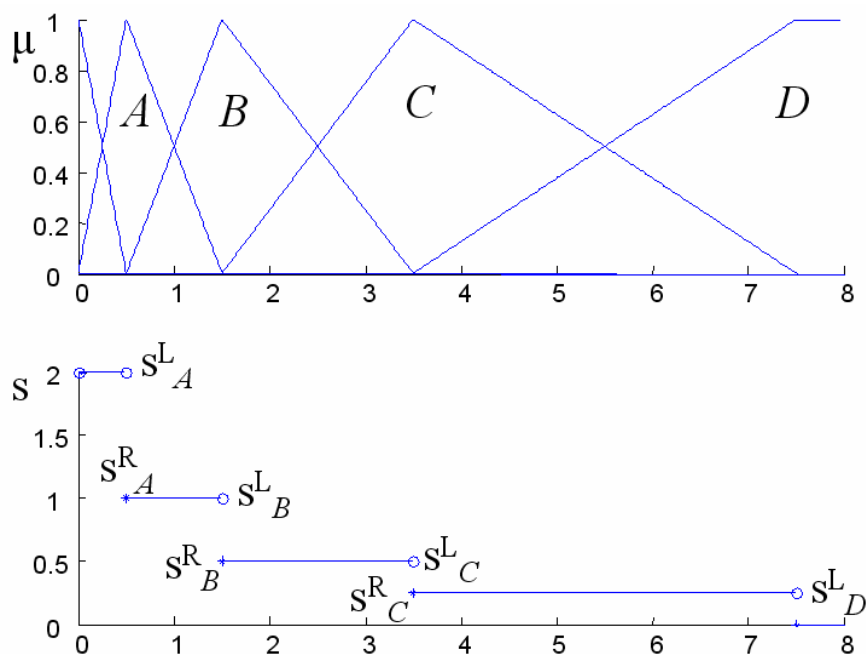
$$S(x) = |\mu'(x)| = \left| \frac{d\mu}{dx} \right| \quad (42)$$

skálafaktorokból (42) összeálló pontos skálafüggvény azonban csak egyes speciális partíciók esetén állítható elő. Ilyen például a 30. ábra bal oldali részén megjelenő legegyszerűbbnek tekinthető eset, ahol egy egyenlő szárú háromszögből álló egyhalmazos ritka partíciót láthatunk, amelynek skálafüggvénye az ábra jobb oldali részén megjelenő konstans érték.



30. ábra. Egyenlő szárú háromszög alakú fuzzy halmaz és skálafüggvénye [H35]

További alkalmazási lehetőséget jelentenek a Ruspini típusú partíciók (pl. 31. ábra), amelyeket leíró pontos skálafüggvény szintén könnyen előállítható Klawonn módszerével. Itt a skálafüggvény vízszintes szakaszok sorozatából épül fel. Minden szakasz egy vagy két él meredekségét írja le. Például az $S_A^R = S_B^L$ értékű rész az A halmaz jobb oldali és a B halmaz bal oldali éléről ad információt.



31. ábra. Ruspini partíció és skálafüggvénye [H34]

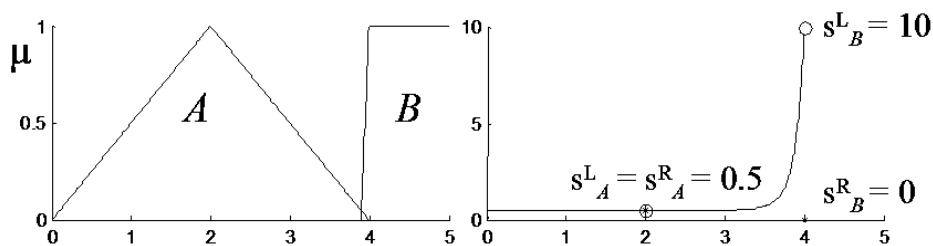
Általános esetben azonban a szomszédos halmazok szemközti éleinek dőlése abszolút értékben sem azonos, így a mindkét élt leíró pontos skálafaktor előállítása nem lehetséges. Ilyenkor közelítő skálafaktorok segítségével jellemezhetők a partíció azon részei, ahol az

egymás melletti fuzzy halmazok alakja jelentősen eltérő. A közelítő skálafüggvényeket a szemközti éleket külön-külön pontosan leíró skálafaktorok közötti interpoláció segítségével állítjuk elő. A 32. ábra bal oldali része egy kéthalmazos partíciót mutat be. A partíciót a bizonytalan környezetben jellemző skálafüggvény első (S_A^L) és harmadik (S_B^R) szakaszát a (42) képlet segítségével számítjuk, míg a második szakaszt, ami az A halmaz csúspontjától a B halmaz magjának bal széléig tart a Kovács és Kóczy [H36] által javasolt nemlineáris interpolációval határozzuk meg

$$S(x) = \begin{cases} \frac{DS_i}{(x_{i+1} - x_i + 1)^{k \cdot DS_i} - 1} \cdot \left(\frac{(x_{i+1} - x_i + 1)^{k \cdot DS_i}}{(x - x_i + 1)^{k \cdot DS_i}} - 1 \right) + s_{i+1}^L & \text{ha } S_i^R \geq S_{i+1}^L \\ \frac{DS_i}{(x_{i+1} - x_i + 1)^{k \cdot DS_i} - 1} \cdot \left(\frac{(x_{i+1} - x_i + 1)^{k \cdot DS_i}}{(x_{i+1} - x + 1)^{k \cdot DS_i}} - 1 \right) + s_i^R & \text{ha } S_i^R < S_{i+1}^L, \end{cases} \quad (43)$$

ahol i az aktuális szakasz kezdőpontjának sorszáma, $[x_i, x_{i+1}) \forall i \in [1, n-1]$ a partíció aktuális szakasza, k egy konstans, ami a szemközti élek meredeksége közti eltéréssel szembeni érzékenységet befolyásolja és DS_i a szemközti élek skálafaktorainak különbsége abszolút értékben

$$DS_i = |S_{i+1}^L - S_i^R|. \quad (44)$$



32. ábra. Eltérő szemközti élek és az őket leíró lineáris interpolációval előállított közelítő skálafüggvény [H36]

A bizonytalan környezetben minden A_j fuzzy halmazt a helyzetét megadó ún. prototípus pont

$(P(A_j) = \{RP(A_j), \mu_{A_j}(RP(A_j))\})$ és az alakját az előzőekben megismert módon a partíció

többi nyelvi értékével együttesen leíró skálafüggvény ($S(x)$) jellemzi. Pontos skálafüggvény esetén a bizonytalan környezet ismeretében a partíció fuzzy halmazai a prototípus pontból kiindulva a skálafüggvény integrálásával bármikor újra előállíthatóak (ld. (52) és (53)). Közelítő skálafüggvény esetén azonban a bizonytalan környezet generálása információvesztéssel jár, így az eredeti halmazalak csak becsülhető.

4.4.2. Halmaz-interpoláció a bizonytalan környezetben (VESI)

A bizonytalan környezet alapú halmaz interpoláció [S2] (VESI - Vague Environment based Set Interpolation) a skálafüggvény partíció leíró képességét felhasználva oldja meg a halmaz-interpoláció feladatát.

Más halmaz-interpolációs/approximációs technikákhoz hasonlóan a VESI is a partíció nyelvi értékei között fennálló kapcsolatra épül. Az eljárás első lépéseként az előző szakaszban megismert módon generáljuk a partíció bizonytalan környezetét. Ezt csak egyszer kell

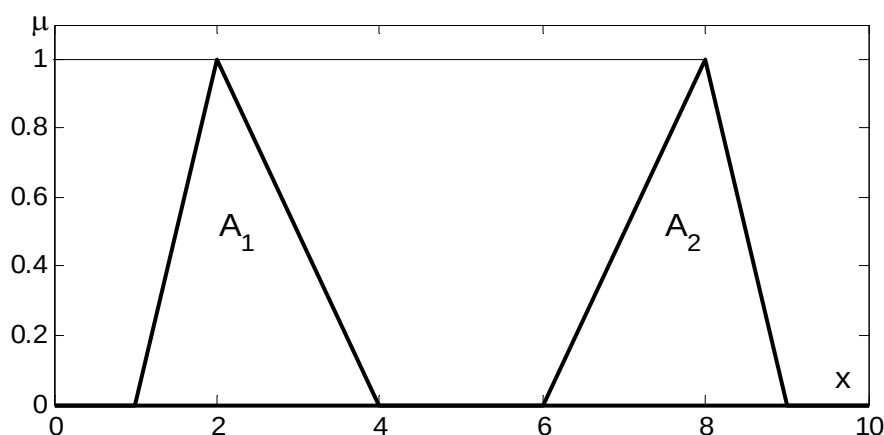
elvégezni, ugyanis a továbbiakban a halmaz-interpoláció során mindig az eredeti skálafüggvénnyel dolgozunk.

A VESI következő szakaszában meghatározzuk a prototípus pontot. Ehhez az interpoláció helyén ($x^i = RP(A^i)$) kívül még azt is ismernünk kell, hogy az adott alaphalmaz elem milyen tagsági értékkel kell bírjon. Amennyiben a partíció minden ismert halmazának prototípus pontja azonos magasságú ($\mu_{A_j}(RP(A_j)) = \mu_{A_1}(RP(A_1))$, $j = 1, \dots, n$), akkor az új halmaz is ezzel a tulajdonsággal fog rendelkezni, egyébként a Shepard interpoláció [H63] segítségével számítjuk az új nyelvi érték prototípus pontjának magasságát az ismert prototípus pontok magasságaiból.

Az interpolált fuzzy halmaz bal és jobb oldali élét külön-külön számítjuk a skálafüggvény segítségével feltételezve, hogy az új nyelvi érték megőrzi a partíció skálafüggvény által leírt alakzat jellegzetességét. A továbbiakban az eljárást egy numerikus példán keresztül ismertetem részletesen. Az egyszerűség és áttekinthetőség érdekében a kiinduló partíció (33) ritka, két CNF háromszög alakú fuzzy halmazt tartalmaz. Ezek definíciója a következő

$$A_1 = 0/1 + 1/2 + 0/4, \quad (45)$$

$$A_2 = 0/6 + 1/8 + 0/9. \quad (46)$$



33. ábra. Ritka partíció két nyelvi értékkel

A partíció alaphalmaza $X = [x_{\min}, x_{\max}] = [0, 10]$. A bizonytalan környezetet jellemző skálafüggvény a (47)...(50) skálafaktorokból áll össze az alábbiak szerint

$$S_{A_1}^L = \left| \frac{d\mu_{A_1^L}}{dx} \right| = \frac{1}{1} = 1, \quad (47)$$

$$S_{A_1}^R = \left| \frac{d\mu_{A_1^R}}{dx} \right| = \frac{1}{2} = 0.5, \quad (48)$$

$$S_{A_2}^L = \left| \frac{d\mu_{A_2^L}}{dx} \right| = \frac{1}{2} = 0.5, \quad (49)$$

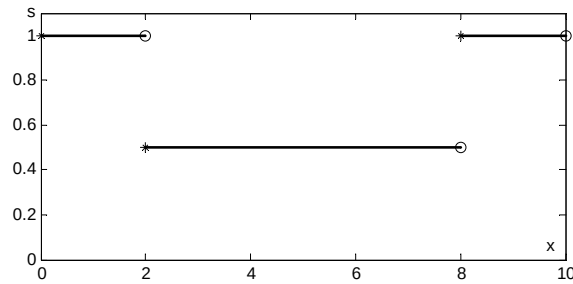
$$S_{A_2}^R = \left| \frac{d\mu_{A_2^R}}{dx} \right| = \frac{1}{1} = 1, \quad (50)$$

ahol S_V^Z a V fuzzy halmaz (A_1 vagy A_2) Z (L-bal vagy R-jobb) oldalának megfelelő skálafaktor.

A skálafüggvény a bizonytalan környezetet szakaszonként írja le állandó vagy nem állandó skálafaktorok segítségével. A szakaszok végpontjait az alaphalmaz alsó és felső korlátai, valamint a prototípus pontok határozzák meg. Minden egyes szakasz skálafaktorát csak az adott szakaszban jelen levő él(ek) befolyásolják [H34][H37]. Így például a $[4, 6)$ intervallumot, ami a partíció egy fuzzy halmazzal nem fedett része, ugyanaz az $S=0,5$ skálafaktor jellemzi, mint az öt közrefogó $[2, 4)$ és $[6, 8)$ intervallumokat. Hasonlóképpen a partíció bevezető és záró üres részeit leíró skálafüggvényt is a hozzájuk legközelebbi él(ek) skálafaktorai határozzák meg. Így a $[0, 1)$ és a $[9, 10)$ intervallumok skálafaktora $S=1$. Ez a tulajdonság biztosítja a bizonytalan környezet halmaz-interpolációs/extrapolációs képességét. A fentiek figyelembevételével a skálafüggvény az alábbi formát kapja

$$S(x) = 1(x) - \frac{1}{2} \cdot 1(x-2) + \frac{1}{2} \cdot 1(x-8), \quad (51)$$

ahol, $1(x)$ az egységugrás függvény.



34. ábra. A 33. ábrán bemutatott partíció skálafüggvénye

Tegyük fel, hogy az interpolációs pont $x^i = 5$ és a partíció minden nyelvi értéke normalizált. Ekkor az interpolált halmaz referencia pontja $RP(A^i) = 5$, míg alakját az alábbiak szerint számítjuk. A bal oldali élt az $S(x)|_{x \leq x^i}$ határozza meg, így az egy olyan egyenes lesz, amely a vízszintes tengelyt a 4-es értéknél metszi ($\mu_{A^i}(4) = 0$). Általános esetben az alábbi képlet írja le a bal él tagsági függvényét

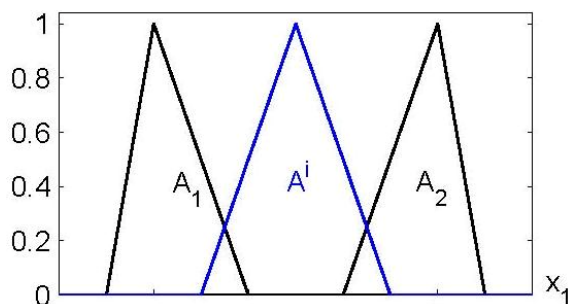
$$\mu_{A^i}^L(x) = \max \left\{ 0, 1 - \int_x^{x^i} S(x) dx \right\}, x \in [x_{\min}, x^i]. \quad (52)$$

A gyakorlatban a tagsági függvényt numerikus módszerrel pontonként számítjuk az x^i értéktől indulva $x_{\min}$ irányába haladva egészen addig, amíg az integrál értéke el nem éri az 1-et.

A jobb oldali élt az $S(x)|_{x \geq x^i}$ határozza meg, így az egy egyenes lesz, amely a vízszintes tengelyt a 7-es értéknél metszi ($\mu_{A^i}(7) = 0$). Általános esetben az alábbi képlet írja le a jobb él tagsági függvényét

$$\mu_{A^i}^R(x) = \max\left\{0, 1 - \int_{x^i}^x S(x)dx\right\}, x \in [x^i, x_{\max}]. \quad (53)$$

A bal élt a gyakorlatban a jobb élhez hasonlóan határozzuk meg. A számítások eredményét a 35. ábra szemlélteti.



35. ábra. A halmaz-interpoláció eredménye

Az eljárás értékeléseként elmondható, hogy a partícióval való kompatibilitás iránti igény csak abban az esetben teljesül, amikor a skálafüggvény minden szakasza pontos skálafaktorokból épül fel. Közelítő skálafaktorok alkalmazása esetén a módszer halmaz-approximációnak minősül. Ugyanezen okból kifolyólag az alakzat megőrzés sem teljesül minden esetben.

A VESI módszer minden érvényes halmazalak típusra alkalmazható, valamint az x^i pont az ismert nyelvi értékek által definiált intervallumon kívülre is eshet. A fentiek tükrében az eljárás feltétel nélkül teljesíti az 1, 4 és 5 követelményeket. Hiányosságai ellenére fuzzy szabály-interpolációs célokra történő alkalmazását vonzóvá teszi gyorsasága, tekintettel arra, hogy a bizonytalan környezetet csak egyszer kell előállítani még a rendszer működése előtt. A következtetési folyamat során csak a numerikus integrálási lépések végrehajtása szükséges. Emellett a közelítésből eredő esetleges pontatlanságok jól kezelhetők a rendszer megfelelő hangolásával.

5. Szabálmódosításon alapuló egyszabályos következtetés – javasolt módszerek

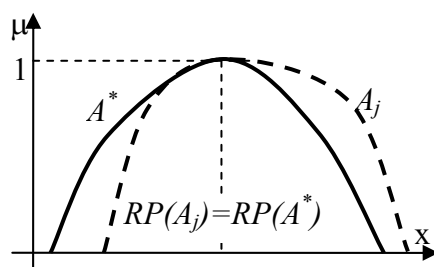
Egyszabályos fuzzy következtetési módszereknek nevezik azokat a technikákat, amelyek egyetlen fuzzy szabályból és az annak feltétel részével nem feltétlenül azonos megfigyelésből kiindulva állítják elő a következményt. Egydimenziós antecedens alaphalmaz esetén a 36. ábra segítségével szemléltethetjük a helyzetet. Ilyenkor a következmény előállítás az általános modus ponens (54) alkalmazásával történik.

Megfigyelés: x értéke A^*

Szabály: $\text{HA } x \text{ értéke } A \text{ AKKOR } y \text{ értéke } B$ (54)

Következmény: y értéke B^*

ahol x és y a bemeneti és kimeneti nyelvi változók, A és B az antecedens és a konzekvens alaphalmazbeli nyelvi értékek, A^* a megfigyelés és B^* az előállított következmény.



36. ábra. Azonos referencia ponttal rendelkező, de eltérő alakú antecedens (A_j) és megfigyelés (A^*) halmaz

A folyamat alapja valamilyen hasonlóságmérték meghatározása és az általános modus ponens olyan megvalósítása, hogy amilyen mértékben hasonlít a megfigyelés a szabály antecedens részére, ugyanolyan mértékben hasonlítson az előállított következmény a szabály konzekvens részére. A továbbiakban három olyan módszert ismertetek, amelyek egy bizonyos speciális feladatra alkalmazhatóak. Ez a kiemelt terület a fuzzy szabály-interpoláció egy részfeladata, mégpedig a becsült következmény meghatározása az általánosított módszertant [H4] követő szabály-interpolációs módszerek második lépésében. Itt egy fuzzy szabály-interpolációval előállított szabállyal dolgozunk, amelynek antecedens halmazához rendelt referencia pont azonos a megfigyelés referencia pontjával.

A szabálmódosításon alapuló következtetési módszerek tárgyalása során csak kompakt fuzzy halmazokkal foglalkozunk, azaz olyan nyelvi értékekkel, amelyeknek korlátos a tartója [H26]. A szakirodalomban fellelhető publikációk többségéhez hasonlóan a továbbiakban csak (55) típusú szabályok esetére korlátozom az általam kidolgozott módszerek ismertetését. Itt a szabályok antecedense lehet többdimenziós, a feltétel részben megnevezett nyelvi értékek között ÉS kapcsolat (konjunkció) áll fenn, továbbá a kimenet egydimenziós (MISO¹ rendszer). Ez utóbbi egyszerűsítést az teszi lehetővé, hogy a több kimenetű szabálybázisok

¹ Multiple Input Single Output

(SIMO² vagy MIMO³) információvesztés nélkül mindig felbonthatók egy kimenetűekre (SISO⁴ vagy MISO).

$$HA\ a_1 = A_{1j_1} \text{ ÉS } a_2 = A_{2j_2} \text{ ÉS } \dots \text{ ÉS } a_N = A_{Nj_N} \text{ AKKOR } b = B_k, \quad (55)$$

ahol a_i a bemeneti nyelvi változó az i . antecedens dimenzióban, N az antecedens dimenziók száma, j_i a szabály i -edik dimenzióbeli antecedensének sorszáma a saját partíciójában, b a kimeneti nyelvi változó, k a szabály konzekvens nyelvi értékének sorszáma a saját partíciójában.

Az alábbiakban ismertetésre kerülő mindhárom egyszabályos következtetési eljárásnál a mag középpontjának megfelelő alaphalmaz elem (RP_{CC} az 1. ábrán) lesz a referencia pont. Az egyszerűbb jelölés érdekében a referencia pont típust jelölő CC alsó indexet elhagyom.

Az általam kidolgozott egyszabályos következtetési módszerek összehasonlítása és értékelése érdekében a fuzzy szabály-interpolációs eljárásokkal szemben támasztott általános elvárások [S5][S24] alapján egy öt pontból álló követelményrendszert állítottam össze. Ezek a következők.

1. *Abnormális (ld. pl. 23. ábra) halmazalak elkerülése* (pl. [H4][H26][H69]). A halmaz-interpolációval előállított új nyelvi érték legyen mindig egy érvényes fuzzy halmaz. Ezt a követelményt α -vágatok segítségével lehet a legegyszerűbben leírni

$$\inf\{[B^*]_{\alpha_1}\} \leq \inf\{[B^*]_{\alpha_2}\} \leq \sup\{[B^*]_{\alpha_2}\} \leq \sup\{[B^*]_{\alpha_1}\} \quad \forall \alpha_1 < \alpha_2 \in [0,1] \quad , \quad (56)$$

ahol $\inf\{V_\alpha\}$ és $\sup\{V_\alpha\}$ a V fuzzy halmaz α -vágatának alsó illetve felső határa és B^* az eljárás által előállított következmény.

2. *Kompatibilitás a szabálybázissal* (a modus ponens érvényességének a feltétele). Ez az elvárás azt fejezi ki, hogy amennyiben a megfigyelés azonos egy szabály antecedens részével, akkor a következmény azonos kell legyen az adott szabály konzekvens részével.
3. *A becsült következmény fuzzy jellege*. Ezzel kapcsolatban két ellentétes megközelítéssel találkozhatunk a szakirodalomban. Az első (3.a) a megfigyelés sajátosságából eredezteti a becsült következmény fuzzy jellegét. Eszerint minél kevésbé bizonytalan a megfigyelés, annál kevésbé kell fuzzy jelleggel bírjon a következmény [H4][H26]. A második (3.b) megközelítés a becsült következmény fuzzy jellegét a szabálybázisból származtatja [H69][H82]. Eszerint egyértékű következmény csak akkor keletkezhet, ha a következtetés során figyelembe vett szabályok egyértékű következményűek.
4. *Alakzat megőrző képesség*. Ha a konzekvens partíció összes halmaza, valamint az interpolált szabály konzekvensé azonos alakzat típusba tartozik, akkor elvárható, hogy a becsült következmény is őrizze meg ezt a tulajdonságot. A gyakorlatban ezt úgy kell értelmezni, hogy ha például az ismert halmazok szakaszonként lineárisak, és a megfelelő töréspontok mindannyiuknál azonos α -szinteken helyezkednek el, akkor az interpolált halmaznak is illeszkednie kell ebbe a rendszerbe (pl. [H4]).
5. *Alkalmazhatóság többdimenziós antecedens univerzum esetében is*.

² Single Input Multiple Output

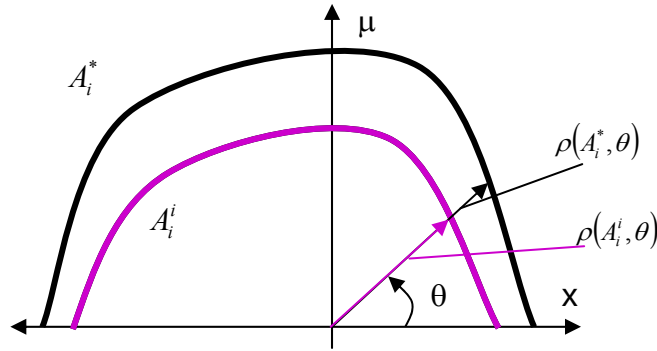
³ Multiple Input Multiple Output

⁴ Single Input Single Output

6. *Halmazalaktól független alkalmazhatóság.* Az eljárás minden érvényes fuzzy halmazalak esetén képes kell legyen becsült következmény előállítására.

5.1. Polár-vágat alapú egyszabályos következtetés (SURE-p)

A SURE-p [S12] (Single rUle REasoning based on polar cuts) eljárás polár-vágatok és súlyozott átlagszámítás segítségével oldja meg a szabálmódosításos következtetés feladatát. Kifejlesztésére a FRIPOC [S12] szabály-interpolációs módszer keretében a FEAT-p halmaz-interpolációs módszer kiegészítéseként került sor.



37. ábra. Megfigyelés (A_i^*) és antecedens (A_i^i) halmaz az i -edik bemeneti dimenzióban

A későbbi számítások jelentős mértékben egyszerűsíthetők, amennyiben a rendszer összes partícióját az egység intervallumra ($[0,1]$) normalizáljuk. A továbbiakban ezért normalizált partíciókat feltételezve ismertetjük az eljárást.

Első lépésként minden antecedens dimenzióban minden polár szinten meghatározzuk a szabály antecedens halmaz aktuális pontjának poláris távolsága és a megfigyelés aktuális pontjának poláris távolsága közötti eltérést (ld. 37. ábra)

$$r_i(\theta) = \rho(A_i^i, \theta) - \rho(A_i^*, \theta), \quad (57)$$

ahol i az aktuális antecedens dimenzió és θ az aktuális polár szög, A_i^i az interpolált szabály antecedens nyelvi értéke az i -edik dimenzióban és A_i^* az i -edik dimenzióbeli megfigyelés halmaz. Ezt követően meghatározzuk a polár-vágatonkénti átlagos eltérést ($\bar{r}(\theta)$) a dimenziókénti eltérések ($r_i(\theta)$) átlagaként:

$$\bar{r}(\theta) = \frac{\sum_{i=1}^N r_i(\theta)}{N}. \quad (58)$$

A SURE-p módszernél a szabálmódosítás alapjául szolgáló hasonlóság értékelést a polár-vágatonkénti eltérések segítségével valósítjuk meg. A módszer alapgondolata az antecedens oldalon mért átlagos eltérések megőrzése és alkalmazása a konzekvens oldalon. Ennek tükrében a következményt polár-vágatonként a (59) egyenlőség segítségével határozzuk meg

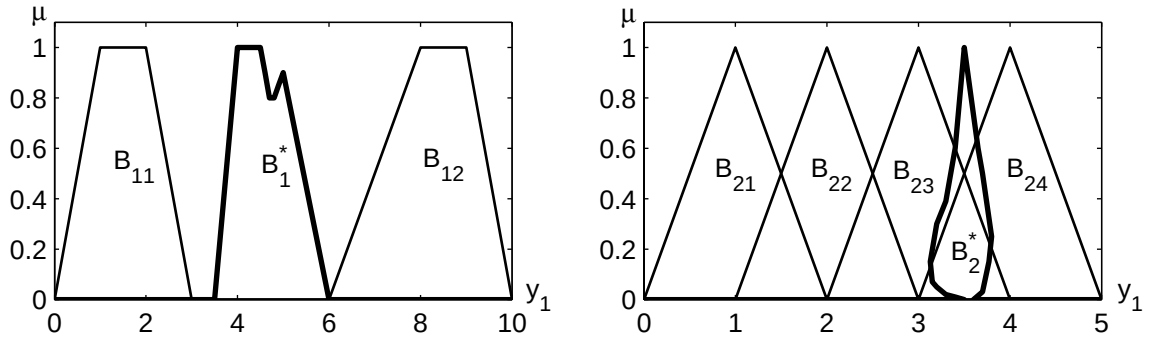
$$\rho(B^i, \theta) - \rho(B^*, \theta) = \bar{r}(\theta), \quad (59)$$

ahol B^i a szabály konzekvensé és B^* a becsült következmény. A fuzzy halmazok sajátosságaira való tekintettel azonban további megkötéseket is figyelembe kell vennünk.

Ilyen feltétel például az, hogy a tagsági függvény értéke nem haladhatja meg az 1-es értéket. Ennek figyelembevételével az alábbi képlethez jutunk

$$\rho(B^*, \theta) = \begin{cases} \min \left[\rho(B^i, \theta) - \bar{r}(\theta), \frac{1}{\sin(\theta)} \right] & \text{ha } \theta \in (0, \pi) \\ \rho(B^i, \theta) - \bar{r}(\theta) & \text{ha } \theta = 0 \text{ vagy } \theta = \pi. \end{cases} \quad (60)$$

A polár-vághatonkénti eltérő átlagos módosítások következtében a (60) képlet könnyen eredményezhet olyan fuzzy halmazt, ami nem teljesíti az érvényesség feltételeit. Példaként tekintsük meg a 38. ábrán látható két esetet. A bal oldali konzekvens partíció két (B_{11} és B_{12}) nyelvi értéket tartalmaz. A (60) képlettel előállított B_1^* következményt vastag vonallal jelöltük. Jól megfigyelhető, hogy a referencia ponttól jobbra egy gyors csökkenésű szakasz után a tagsági függvény értéke újból növekedésnek indul. A jobb oldalon bemutatott esetben a partíció négy nyelvi értéket tartalmaz. A (60) képlettel előállított B_2^* következményt vastag vonallal jelöltük. Itt a tagsági függvény mindkét oldalon a halmaz „alá” kanyarodik így egy alaphalmaz értékhez két tagsági érték is társulhat, ami érvénytelenné teszi az eredményt.



38. ábra. A (60) képlettel számolt érvénytelen következmény halmazok

Az érvényes halmazalak biztosítása érdekében az eljárás menetét a következőképpen pontosítjuk. A számításokat $\pi/2$ -es polár szögnél (a referencia pontnak megfelelő tagsági értéknél) kezdjük, és külön-külön határozzuk meg a bal- és jobb oldali él pontjait. A továbbiakban csak a jobb oldali él előállításának lépéseit ismertetjük, a bal oldali él előállítása hasonló módon történik.

Az első pontot a (60) képlet segítségével határozzuk meg. Trigonometriai negatív irányban haladva (csökkenő polár szögek) a második ponttól kezdődően (ld. 39. ábra) a halmazalak érvényessége iránti követelmény (ld. (56)) akkor és csak akkor teljesül, ha minden pont kielégíti az alábbi két feltételt.

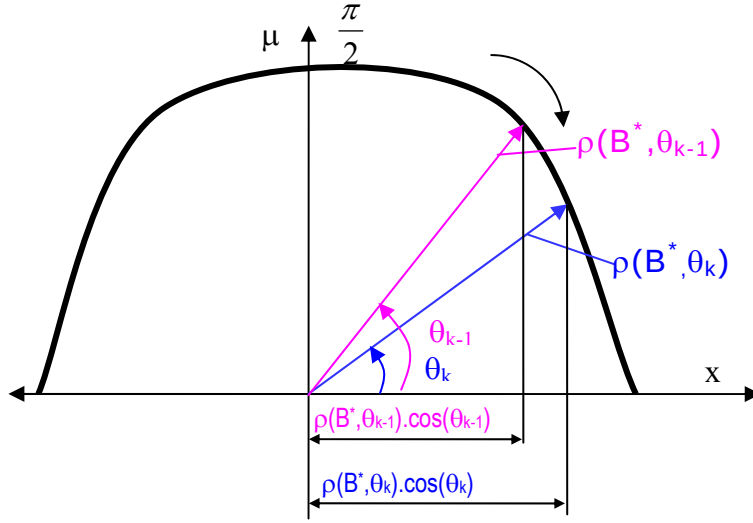
- Az aktuális pont vízszintes irányú távolsága a polár koordináta-rendszer középpontjától nem lehet kisebb, mint az öt megelőző pont ugyanilyen távolsága

$$\rho(B^*, \theta_k) \cdot \cos(\theta_k) \geq \rho(B^*, \theta_{k-1}) \cdot \cos(\theta_{k-1}), \quad (61)$$

ahol θ_k a k -adik pont polár szöge.

- Az aktuális pont függőleges irányú távolsága a polár koordináta-rendszer középpontjától (magasság) nem lehet nagyobb, mint az öt megelőző pont ugyanilyen távolsága

$$\rho(B^*, \theta_k) \cdot \sin(\theta_k) \leq \rho(B^*, \theta_{k-1}) \cdot \sin(\theta_{k-1}). \quad (62)$$



39. ábra. Egymást követő polár-vágatok

A fentiek figyelembevételével a második ponttól kezdődően a (60) képlettel kapott értékre alkalmazzuk először a (63), majd ezt követően a (64) korrekciót.

$$\rho(B^*, \theta_k) = \text{Max} \left[\rho(B^*, \theta_{k-1}) \cdot \frac{\cos(\theta_{k-1})}{\cos(\theta_k)}, \rho(B^*, \theta_k) \right] \quad (63)$$

$$\rho(B^*, \theta_k) = \text{Min} \left[\rho(B^*, \theta_{k-1}) \cdot \frac{\sin(\theta_{k-1})}{\sin(\theta_k)}, \rho(B^*, \theta_k) \right] \quad (64)$$

Az eljárás jelentős előnyös tulajdonsága, hogy bármilyen érvényes halmazalak esetén alkalmazható, valamint a számítások során felhasznált fuzzy halmazok nem szükséges, hogy azonos alakúak és magasságúak legyenek, azaz egy vagy több halmaz szubnormális is lehet. Amennyiben a megfigyelés azonos valamely szabály antecedensével, a SURE-p módszer nem módosítja a szabály konzekvensét, így teljesül a szabálybázissal való kompatibilitás iránti igény. Az eljárás egy- vagy többdimenziós antecedens univerzum esetén is egyaránt alkalmazható, azonban nem alkalmas a partícióra jellemző alakzattípus megőrzésére. A megfigyelés fuzzy jellegének csökkenése a következmény fuzzy jellegének csökkenését vonja maga után. Összefoglaló értékelésként elmondható, hogy a SURE-p kielégíti az 1, 2, 3.a, 5 és 6 elvárásokat.

5.2. Legkisebb négyzetek elvén alapuló egyszabályos következtetés (SURE-LS)

A legkisebb négyzetek elvén alapuló egyszabályos következtetés [S13] (SURE-LS – Single rUle REasoning based on the method of Least Squares) α -vágatonként haladva a konzekvens partícióra jellemző halmazalak típus megőrzésével oldja meg a szabálmódosításon alapuló következtetés feladatát. Kifejlesztésére a LESFRI [S13] szabály-interpolációs módszer keretében a FEAT-LS [S1] halmaz-interpolációs módszer kiegészítéseként került sor.

A SURE-p-hez hasonlóan a későbbi számítások jelentős mértékben egyszerűsíthetők, amennyiben a rendszer összes partícióját normalizáljuk. A továbbiakban ezért normalizált partíciókat feltételezve ismertetjük az eljárást.

Első lépésként összeállítjuk azon α -szintek halmazát, amelyek tartalmazzák az összes antecedens és konzekvens dimenzióban a töréspontok jellegzetes magasságait szakaszonként lineáris halmazalak esetén, illetve a karakterisztikus pontoknak megfelelő szinteket más nyelvi érték alakoknál.

$$\Lambda = \left(\bigcup_{i=1}^N \Lambda_i^a \right) \bigcup \Lambda^c, \quad (65)$$

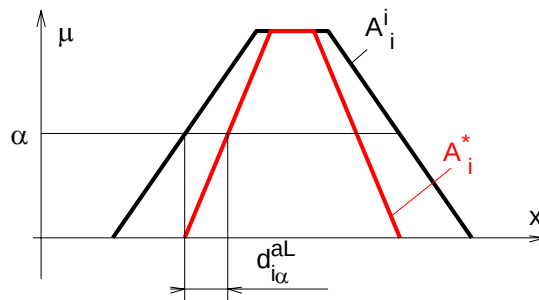
ahol Λ_i^a az i -edik antecedens dimenzió által meghatározott α -szintek halmaza, N az antecedens dimenziók száma, míg Λ^c a konzekvens dimenzió által meghatározott α -szintek halmaza.

A számítások további részét külön-külön végezzük el a referencia pont tagsági szintjénél szétválasztott bal és jobb oldali élre. Az alábbiakban csak a bal él meghatározását ismertetjük, a jobb élnél hasonló módon járunk el. Először minden antecedens dimenzióban minden α -szintre meghatározzuk az interpolált szabály antecedens α -vágatának alsó végpontja és a megfigyelés halmaz α -vágatának alsó végpontja közti eltérést

$$d_{i\alpha}^{al} = \inf \{ [A_i^i]_{\alpha} \} - \inf \{ [A_i^*]_{\alpha} \}, \quad (66)$$

ahol i az antecedens dimenzió sorszáma, $[A_i^i]_{\alpha}$ az i -edik dimenzióbeli interpolált szabály antecedens halmaz α -vágata, $A_{i\alpha}^*$ az i -edik dimenzióbeli megfigyelés halmaz α -vágata (ld. 40. ábra). Ezt követően minden α -szintre kiszámítjuk az átlagos eltérést

$$ad_{\alpha}^{al} = \frac{\sum_{i=1}^N d_{i\alpha}^{al}}{N}. \quad (67)$$



40. ábra. Megfigyelés (A_i^*) és antecedens (A_i^i) halmaz bal oldali eltérése az i -edik bemeneti dimenzióban az α szinten

A SURE-LS módszernél a szabálymódosítás alapjául szolgáló hasonlóság értékelést az α -vágatonkénti eltérések segítségével valósítjuk meg. A módszer alap gondolata az antecedens

oldalon mért átlagos eltérések megőrzése és alkalmazása a konzekvens oldalon. Ennek tükrében a következményt α -vágatonként a (68) egyenlőség segítségével határozzuk meg

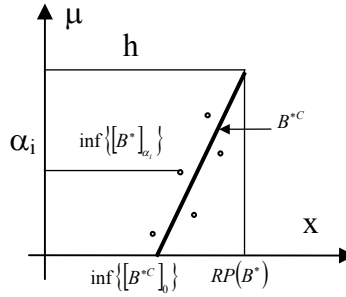
$$d_{\alpha}^{cl} = \inf \{ [B^i]_{\alpha} \} - \inf \{ [B^*]_{\alpha} \} = ad_{\alpha}^{al} \quad (68)$$

ahol $[B^i]_{\alpha}$ az interpolált szabály konzekvens halmazának α -vágata és $[B^*]_{\alpha}$ a következmény halmaz α -vágata.

Az érvényes halmazalak biztosítása érdekében nem engedhetjük meg, hogy az él (68) egyenlőség alapján számított pontjai a referencia ponttól jobbra helyezkedjenek el. Így az α -vágatok alsó végpontjait a következő képlettel nyerjük

$$\inf \{ [B^*]_{\alpha} \} = \text{Min} \{ \inf \{ [B^i]_{\alpha} \} - ad_{\alpha}^{al}, RP(B^*) \} \quad (69)$$

A SURE-LS eljárást kifejezetten arra az esetre fejlesztettem, amikor a konzekvens partíció minden nyelvi értéke azonos alakzat típusba tartozik. A (69) képlettel előállított halmazalak azonban általános esetben nem teljesíti ezt az elvárást, ezért következő lépésként a legkisebb négyzetek módszerének alkalmazásával megkeresem azt a halmazél alakot, ami illeszkedik a partíció sajátosságaihoz és emellett a lehető legkisebb vízszintes irányú négyzetes eltéréssel rendelkezik a (69)-vel számított pontoktól.



41. ábra. A háromszög bal éle

A fentiek következtében a további számítások menete az adott alakzat típustól függ. Például singleton típus (egyelemű fuzzy halmaz) esetén a végeredmény is egy singleton alakú nyelvi érték lesz, amely a következmény referencia pontja által meghatározott pozícióban helyezkedik el. Az alábbiakban a háromszög alakzat esetét ismertetjük. Ekkor a referencia pont tagsági értéke (a partíció jellegzetes halmaz magassági értéke) határozza meg a háromszög csúcsát (41. ábra). Amennyiben az élt leíró egyenes egyenletét

$$x(\alpha) = a \cdot \alpha + b, \alpha \in \Lambda \quad (70)$$

alakban írjuk fel, és alkalmazzuk a csúcspont ismeretéből adódó $b = RP(B^*) - a \cdot h$ behelyettesítést, ahol h a partícióra jellemző halmazalak magasság, akkor a négyzetes eltérések összege az alábbi

$$Q = \sum_{k=1}^{n_A} \left[\inf \{ [B^*]_{\alpha_k} \} - RP(B^*) + a \cdot (h - \alpha_k) \right]^2, \quad (71)$$

ahol n_Λ az alkalmazott α -vágatok száma. A négyzetösszeg az alábbi a érték esetén éri el minimumát

$$a = \frac{\sum_{k=1}^{n_\Lambda} (h - \alpha_k) \cdot (RP(B^*) - \inf\{[B^*]_{\alpha_k}\})}{\sum_{k=1}^{n_\Lambda} (h - \alpha_k)^2}. \quad (72)$$

Ezt felhasználva a bal oldali él másik végpontját a következő képlet szolgáltatja

$$\inf\{[B^{*C}]_0\} = RP(B^*) - h \frac{\sum_{k=1}^{n_\Lambda} (h - \alpha_k) \cdot (RP(B^*) - \inf\{[B^*]_{\alpha_k}\})}{\sum_{k=1}^{n_\Lambda} (h - \alpha_k)^2}, \quad (73)$$

ahol B^{*C} az eljárás által számított következmény halmaz.

A módszer értékeléseként elmondható, hogy mindig érvényes halmazalakot eredményez, és megőrzi a konzekvens partíció jellegzetes halmazalak típusát. Amennyiben a megfigyelés azonos valamely szabály antecedensével, az eljárás nem módosítja a szabály konzekvensét, így biztosított a szabálybázissal való kompatibilitás iránti igény kielégítése. Az eljárást alapvetően arra az esetre dolgoztam ki, amikor a konzekvens partíció minden nyelvi értéke azonos alakzat típusú. A SURE-LS az α -vágat alapú technikának köszönhetően vegyes alakzatoknál is alkalmazható, amennyiben egy kellően általános befoglaló alakzat típust tudunk definiálni és az összes fuzzy halmaz úgy antecedens, mint konzekvens oldalon azonos magasságú. A módszer egy- és többdimenziós antecedens tér esetén egyaránt használható. A következmény fuzzy halmaz egyértékű lesz, amennyiben a konzekvens partíció minden nyelvi értéke egyértékű. Egyéb esetekben a megfigyelés fuzzy jellegének csökkenése a következmény fuzzy jellegének csökkenését eredményezi. A SURE-LS eljárás teljesíti az 1, 2, 4, 5 és 6 elvárásokat.

5.3. Szabálymódosítás bizonytalan környezetben (REVE)

A bizonytalan környezet fogalmára épülő szabálymódosítási eljárás [S26] (REVE - Revision mEthod based on the Vague Environment) az antecedens partíciókat leíró skálafüggvények és a megfigyelés bizonytalan környezetét jellemző skálafüggvény eltérése alapján határozza meg a következményt. Kifejlesztésére a VEIN [S26] szabály-interpolációs módszer keretében a VESI [S2] halmaz-interpolációs módszer kiegészítéseként került sor.

A REVE módszer első lépéseként elkészítjük a rendszer antecedens és konzekvens partícióira jellemző skálafüggvényeket ($S_{A_i}(x)$ és $S_B(x)$). Amennyiben a REVE-t a VESI halmaz-interpolációs módszerrel együtt alkalmazzuk, akkor a korábban a VESI használata során létrehozott bizonytalan környezetekkel dolgozhatunk itt is. A fentiek mellett minden antecedens dimenzióban szükségünk van a megfigyelést leíró $S_{A_i^*}(x)$ skálafüggvényre.

Következő lépésként minden antecedens dimenzióban kiszámítjuk a megfigyelés skálafüggvényének és az aktuális partíció skálafüggvényének az arányát ($r_{A_i}(x)$). Ez az i -edik dimenzióban az alábbi

$$r_{A_i}(x) = \frac{S_{A_i^*}(x)}{S_{A_i}(x)}. \quad (74)$$

A REVE módszer alapgondolata a skálafüggvények arányának megőrzése szabálymódosítás során. Ennek megfelelően a következmény alakját meghatározó skálafüggvényt úgy definiáljuk, hogy a felhasznált szabály konzekvenssekre jellemző skálafüggvénnyel (azaz a konzekvens partíció skálafüggvényével) képzett aránya legyen azonos az előzőekben számolt antecedens oldali aránnyal. Más szóval a fuzzy halmazok közötti hasonlóságot a bizonytalan környezetek hasonlóságán keresztül értékeljük a skálafüggvények arányával. Többdimenziós antecedens terű (MIxx) rendszerek esetén az antecedens skálafüggvényarányok valamilyen aggregációja szükséges, azaz egyfajta átlagos arányszámot kell megállapítani. A feladatot a harmonikus átlag segítségével oldjuk meg

$$mr_A(x) = \frac{N}{\sum_{i=1}^N \frac{1}{r_{A_i}(x)}}, \quad (75)$$

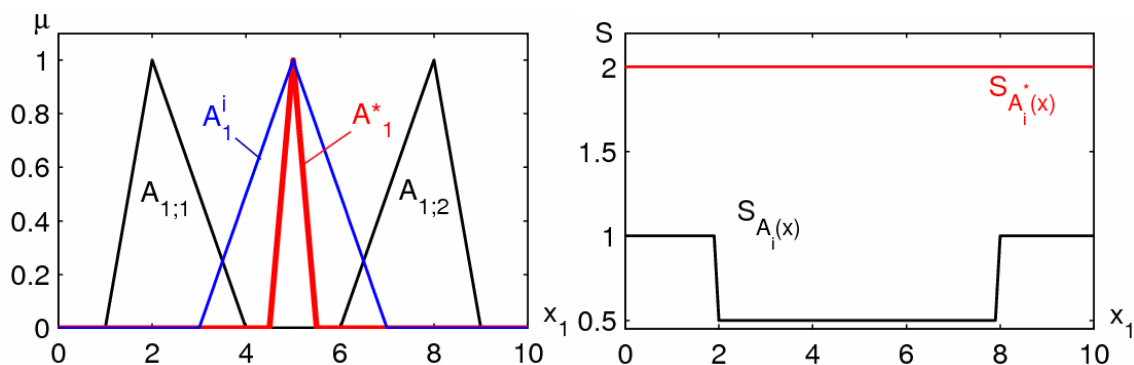
ahol N az antecedens dimenziók száma. Ennek ismeretében megfogalmazhatjuk az átlagos skálafüggvény arány megőrzés elvét, ami szerint a konzekvens oldali skálafüggvényarány ($r_B(x)$) azonos kell legyen az antecedens oldali átlagos skálafüggvényaránnal ($mr_A(x)$).

$$r_B(x) = \frac{S_{B^*}(x)}{S_B(x)} = mr_A(x), \quad (76)$$

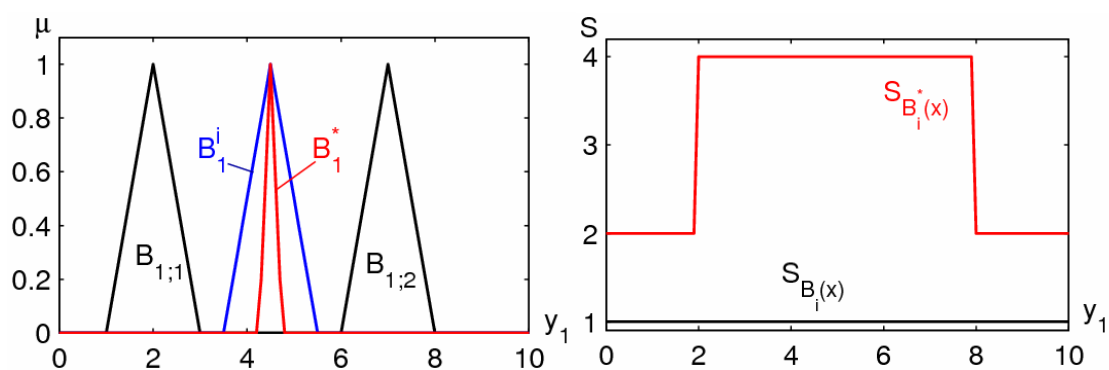
ahol $S_B(x)$ a konzekvens partícióra jellemző skálafüggvény. Ebből adódik a következményt leíró skálafüggvény

$$S_{B^*}(x) = S_B(x) \cdot mr_A(x). \quad (77)$$

A következmény skálafüggvényének ismeretében a VESI halmaz-interpolációs módszer tárgyalása során megismert módon (ld. (52) és (53)) könnyen számítható a következmény halmaz alakja, ami bizonytalan környezet sajátosságából adódóan mindig érvényes fuzzy halmaz lesz.



42. ábra. Eredeti szabály antecedens halmazok ($A_{1;1}$ és $A_{1;2}$), megfigyelés (A_1^*), a megfigyeléssel azonos pozícióban interpolált antecedens halmaz (A_1^i) (bal oldali ábra) valamint antecedens ($S_{A_i(x)}$) és megfigyelés ($S_{A_i^*(x)}$) skálafüggvények (jobb oldali ábra)



43. ábra. Eredeti szabály konzekvens halmazok ($B_{1;1}$ és $B_{1;2}$), megfigyelés (B_1^*), a megfigyeléssel azonos pozícióban interpolált konzekvens halmaz (B_1^i) (bal oldali ábra) valamint konzekvens ($S_{B_i(x)}$) és következmény ($S_{B_i^*(x)}$) skálafüggvények (jobb oldali ábra)

Az eljárás illusztrációjaként tekintünk a 42. és 43. ábrákon bemutatott SISO rendszert, amelynek antecedens partíciója azonos a 4.4.2. szakaszban példaként használt partícióval. A megfigyelés szintén a 4.4.2. szakaszbeli. A rendszerünk két szabállyal rendelkezik, valamint feltételezzük, hogy az interpolált szabály konzekvensét már előállítottuk. A következmény skálafüggvényének előállítása és integrálása után a 43. ábrán bemutatott B_1^* kapjuk.

A REVE szabálmódosításon alapuló következtetési eljárás egyik legfontosabb előnyös tulajdonsága az alacsony számítási bonyolultság és az ebből következő gyorsaság. További pozitív jellemzője, hogy egyaránt alkalmazható egy- és többdimenziós antecedens univerzumok esetén, valamint minden érvényes halmazalak kezelésére képes. A harmonikus átlag aggregációs operátorként történő alkalmazása egyértékű fuzzy halmaz következményt eredményez ($S_{B_i^*}(x) = \infty$), amennyiben a megfigyelés minden dimenzióban egyértékű halmaz ($S_{A_i}(x) = \infty, \forall i = 1, \dots, N$). A skálafüggvényarány megőrzési elvének köszönhetően a következmény fuzzy jellegének változása ugyancsak követi a megfigyelés fuzzy jellegének változását.

Az eljárás gyenge pontja a szabálybázissal való kompatibilitás kérdéséhez kapcsolódik. Bár a következmény skálafüggvénye minden esetben azonos a szabály konzekvensének skálafüggvényével, ha a megfigyelés skálafüggvénye is megegyezik a szabály antecedens

skálafüggvényével, de a skálafüggvények definíciója következtében a megkívánt kompatibilitás csak akkor teljesül, amikor minden partícióban pontos skálafüggvénnyel dolgozhatunk. Összefoglalásként megállapítható, hogy az eljárás kielégíti az 1, 3.a, 5 és 6 követelményeket.

6. Új fuzzy szabály-interpolációs módszerek

Bár az előző két fejezetben bemutatott fuzzy halmaz-interpolációs és egyszabályos következtetési módszerek bármilyen párosításban alkalmasak a saját kategóriájukon belül szükséges számítások elvégzésére, azonban egyértelműen meghatározható egy ideális pár illetve kiegészítő mindegyikük esetében. Ez annak köszönhető, hogy a szabálmódosítási eljárások nem a halmaz-interpolációs módszerektől teljesen függetlenül lettek kifejlesztve, hanem úgy, hogy megfelelő párosítás esetén a halmaz-interpoláció során keletkező adatok egy része az egyszabályos következtetés során is hasznosítható legyen.

A fentiek figyelembevételével ezen eljárások előnyös tulajdonságai a következő három szabály-interpolációs eljárás keretei között használhatók ki a leghatékonyabban. Mindhárom eljárás közös jellemzője, hogy az általánosított fuzzy szabály-interpolációs módszertant (GM) [H4] követik, és a Shepard interpoláció [H63] egy adaptált változatával számítják ki az új szabály konzekvens halmazának helyzetét. Ez utóbbi bemutatására a 6.2. szakaszban kerül sor.

6.1. A három szabály-interpolációs eljárás

6.1.1. Polár-vágat alapú szabály-interpoláció (FRIPOC)

A Polár-vágat alapú szabály-interpoláció [S12] (FRIPOC – Fuzzy Rule Interpolation based on Polar Cuts) nevének megfelelően az első lépésben a segéd szabály antecedens és konzekvens nyelvi értékeinek meghatározására polár-vágat alapú halmaz-interpolációt (FEAT-p) használ. Mivel ezen fuzzy halmazok számítása gyakorlatilag egymástól függetlenül történik, így implementáció esetén a programozási nyelv és a platform függvényében a számítások jelentősen gyorsíthatók az egyes halmaz-interpolációk párhuzamos végrehajtásával.

Az új szabály konzekvens halmazának referencia pontját a Shepard interpoláció [H63] egy adaptált változatával (ld. 6.2. szakasz) számítjuk. A következtetési folyamat második lépésében a szintén polár-vágat alapú SURE-p eljárás segítségével határozzuk meg a következményt.

6.1.2. Legkisebb négyzetek elvén alapuló szabály-interpoláció (LESFRI)

A legkisebb négyzetek elvén alapuló szabály-interpoláció [S13] (LESFRI - LEast Squares based Fuzzy Rule Interpolation) az első lépésben a segéd szabály antecedens és konzekvens nyelvi értékeinek meghatározására a legkisebb négyzetek elvén alapuló halmaz-interpolációt (FEAT-LS) alkalmazza. Mivel ezen fuzzy halmazok számítása gyakorlatilag egymástól függetlenül történik, így az implementáció esetén a programozási nyelv és a platform függvényében a számítások jelentősen gyorsíthatók az egyes halmaz-interpolációk párhuzamos végrehajtásával.

Az új szabály konzekvens halmazának referencia pontját a Shepard interpoláció [H63] egy adaptált változatával (ld. 6.2. szakasz) számítjuk, majd a következtetési folyamat második lépésében a szintén legkisebb négyzetek módszerén alapuló SURE-LS eljárás segítségével határozzuk meg a következményt.

6.1.3. Kétlépéses fuzzy szabály-interpoláció bizonytalan környezetben (VEIN)

A bizonytalan környezetben végrehajtott kétlépéses fuzzy szabály-interpoláció [S26] (VEIN - Vague Environment based two-step fuzzy rule INterpolation) az első lépésben a segéd szabály antecedens és konzekvens nyelvi értékeinek meghatározására bizonytalan környezet alapú

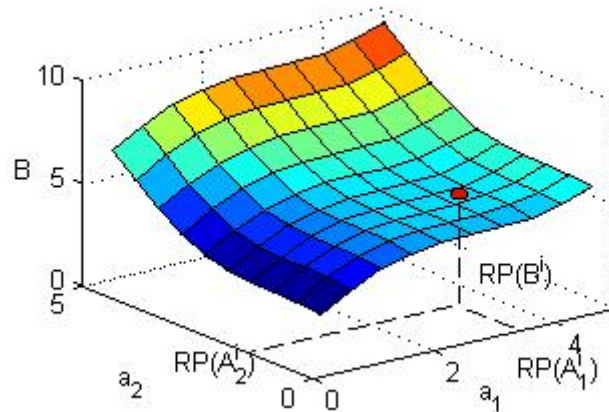
halmaz interpolációt (VESI) alkalmaz. Mivel ezen fuzzy halmazok számítása gyakorlatilag egymástól függetlenül történik, így az implementáció esetén a programozási nyelv és a platform függvényében a számítások jelentősen gyorsíthatók a halmaz-interpolációk párhuzamos végrehajtásával.

Az új szabály konzekvens halmazának referencia pontját a Shepard interpoláció [H63] egy adaptált változatával (ld. 6.2. szakasz) számítjuk. A következtetési folyamat második lépésében a szintén bizonytalan környezet fogalmára épülő szabálmódosítási eljárás (REVE) segítségével határozzuk meg a következményt.

6.2. A konzekvens fuzzy halmazok helyzetének meghatározása

A GM első lépésének második szakaszában meg kell határozni az interpolált szabály konzekvens nyelvi értékének pozícióját (referencia pont). Kiindulásként feltételezzük, hogy a szabálybázis minden eleme (szabálya) egy ponttal jellemezhető egy hiperfelületen, mely pontot a szabály antecedens halmazainak referencia pontjai és a konzekvens nyelvi érték referencia pontja definiálnak.

Illusztrációként tekintsük a 44. ábrát, amely egy két bemenettel és egy kimenettel rendelkező MISO rendszer szabálybázisához rendelt felületet mutatja be. Egy interpolált szabály konzekvens halmaza helyzetének meghatározása ebben az esetben azt jelenti, hogy a két antecedens halmaz helyzetének ($RP(A_1^i)$ és $RP(A_2^i)$) ismeretében egy pontot keresünk ezen a felületen. A kapott pont magassága határozza meg a konzekvens halmaz referencia pontját ($RP(B^i)$).



44. ábra. A szabályokban szereplő fuzzy halmazok referencia pontjai által meghatározott hiperfelület két bemenettel és egy kimenettel rendelkező MISO rendszerénél

Általános esetben ritka szabálybázist feltételezve a feladatot szabálytalanul elhelyezkedő pontok közötti N dimenziós, ahol N az antecedens dimenziók száma, interpolációként fogalmazhatjuk meg, amit a következő egyenlettel írhatunk le

$$RP(B^i) = f(RP(A_1^i), RP(A_2^i), \dots, RP(A_j^i), \dots, RP(A_N^i), RB), \quad (78)$$

ahol $RP(B^i)$ az interpolált szabály konzekvensének referencia pontja, f a tényleges felületet közelítő interpolációs függvény és RB a szabálybázis. A konzekvens halmaz helyzetének számítására a 2D adatok közötti interpoláció céljára kidolgozott Shepard interpoláció [H63] többdimenziós esetre kiterjesztett változatát [H73] adaptáltam az alábbiak szerint.

A keresett referencia pontot az ismert szabályok konzekvens referencia pontjainak súlyozott átlagaként számítjuk

$$RP(B^i) = \frac{\sum_{k=1}^{N_R} RP(B_k) \cdot w_k}{\sum_{k=1}^{N_R} w_k}, \quad (79)$$

ahol N_R a szabálybázis szabályainak száma, $RP(B_k)$ a k -edik szabály konzekvensének referencia pontja, w_k a k -edik szabályhoz rendelt súlytényező. A súlyozással azt kívánjuk kifejezni, hogy az interpolált szabályhoz közelebbi szabályok nagyobb hatást kell fejtsenek a referencia pont számításánál, mint azok, amelyek távolabb helyezkednek el. Itt a szabályok távolságán antecedenseik távolságát értjük. Minden szabály antecedens része jellemezhető egy ponttal az antecedens térben, amelyet a benne szereplő nyelvi értékek referencia pontjai definiálnak. A szabály antecedensek és ezáltal a szabályok távolságán ezen pontok közötti euklideszi távolságot értjük

$$d(RA_k, RA_l) = \sum_{i=1}^N (RP(A_{ik}) - RP(A_{il}))^2, \quad (80)$$

ahol RA_k a k -edik szabály antecedense, $RP(A_{ik})$ a k -edik szabály i -edik dimenzióbeli antecedens halmazának referencia pontja.

A távolság alapú súlyozás a nyelvi értékek eltolásának elvét alkalmazó halmaz-interpolációs eljárásokhoz hasonlóan többféleképpen is megoldható, például a távolság reciprokának négyzetével

$$w_k = \frac{1}{d(RA^i, RA_k)^2}, \quad (81)$$

ahol RA^i az interpolált szabály antecedense.

A számításigény csökkentése érdekében a 2D interpoláció esetében Shepard csak a tíz legközelebbi pont figyelembe vételét javasolta. Többdimenziós antecedens terű ritka szabálybázis esetén általában a szabálysám nem nagy, és az összes szabály figyelembe vétele nem növeli a számításigényt ahhoz képest, mintha először kikeresnénk a tíz legközelebbi szabályt, és csak azokat használnánk fel. Ezért az általam alkalmazott eljárás az összes szabály figyelembevételével dolgozik.

7. FRI Matlab toolbox

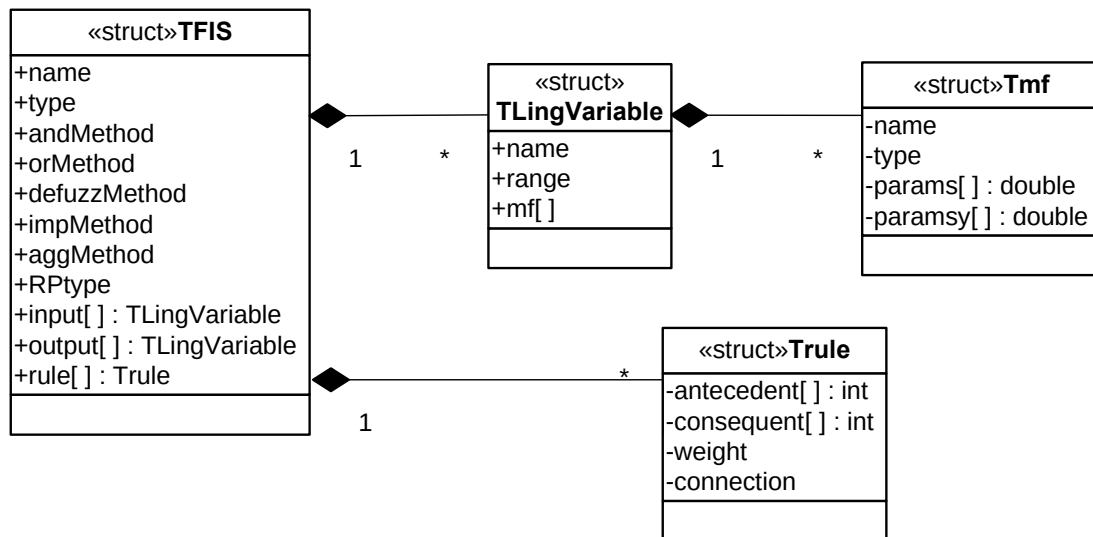
A fuzzy szabály-interpoláción alapuló következtetési módszerek összevetésének, vizsgálatának és gyakorlati alkalmazásának támogatására egy keretrendszer struktúrát terveztem és implementáltam.

A keretrendszer megvalósításához Matlab környezetet választottam, mivel a rendelkezésre álló ismeretek alapján ez a szoftver tűnt a legelterjedtebbnek a fuzzy rendszerekkel foglalkozó szakemberek körében. További szempont volt, hogy már létezik hozzá egy Fuzzy Logic ToolBox (FLTB a továbbiakban) nevű eljárásgyűjtemény, ami Mamdani és a Takagi-Sugeno féle következtetés alapú fuzzy rendszerek fejlesztését támogatja.

7.1. Adatszerkezetek

A keretrendszer kidolgozása és az egyes eljárások implementálása során a fuzzy rendszerek leírására egy olyan adatszerkezetet alakítottam ki, amely a Matlab szabványos Fuzzy Logic ToolBox FIS adatstruktúrájának kiterjesztése. Ezáltal a szabványos Matlab FIS modellek közvetlen alkalmazása mellett lehetőség nyílt szubnormális fuzzy halmazalakok kezelésére is.

Az eredeti FIS struktúra csak CNF halmazok kezelését tette lehetővé, illetve szakaszonként lineáris tagsági függvény típusoknál csak szingleton, háromszög és trapéz alakzatok leírására volt alkalmas, mivel csak a töréspontok abszcisszáit tárolta. Az általam kidolgozott adatszerkezet egyik fő újdonsága az, hogy a tagsági függvényeket leíró struktúrában (*Tmf* a 45. ábrán) a *paramsy* tömb adattag beépítésével lehetővé tettem a halmazalakok töréspontjai tagsági értékeinek a tárolását. Ezáltal az alkalmazható tagsági függvény alakok körét kibővítettem az általános sokszög típussal, valamint lehetővé tettem szubnormális halmazalakok kezelését. Az adatszerkezet további kiegészítése még a referencia pont típusának tárolása.



45. ábra. A fuzzy rendszert leíró alap adatszerkezet

A fuzzy rendszert leíró adatokat az FLTB egy szöveges állományban tárolja, ami könnyen olvasható és a felhasználó által a Matlabtól függetlenül is átirható. A töréspontok tagsági értékeire vonatkozó kiterjesztést ebbe az állományba is beépítettem (ld. 46. ábra bal oldal). Az adatok a sima szöveges állomány mellett XML formátumban is menthetők és olvashatók (ld. 46. ábra jobb oldal), ami a kényelmesebb kezelésük mellett segíti a más alkalmazásokba történő könnyű integrálhatóságukat.

```
[System]
Name='Sin_RBE_SI_LESFRI_01'
Type='Sparse'
Version=2.0
NumInputs=1
NumOutputs=1
NumRules=2
AndMethod=''
OrMethod=''
ImpMethod=''
AggMethod=''
DefuzzMethod='COG'
```

```
[Input1]
Name='x'
Range=[0 10]
NumMFs=2
MF1='A_{1;1}':'trapmf',[0.85 1.35 1.85 2.35]*[0 1 1 0]
MF2='A_{1;2}':'trapmf',[3.95 4.45 4.95 5.45]*[0 1 1 0]
```

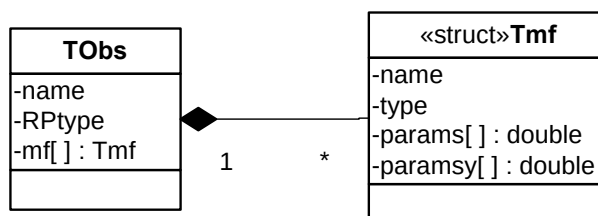
```
[Output1]
Name='y'
Range=[-1 1]
NumMFs=2
MF1='B_{1;1}':'trapmf',[-1.15 -1.05 -0.95 -0.85]*[0 1 1 0]
MF2='B_{1;2}':'trapmf',[0.85 0.95 1.05 1.15]*[0 1 1 0]
```

```
[Rules]
2, 1 (1) : 1
1, 2 (1) : 1
```

```
<?xml version="1.0" ?>
- <root xml_tb_version="3.1" idx="1" type="struct" size="1 1">
  <FileName idx="1" type="char" size="1 34">Sin_RBE_SI_LESFRI_01.xml</FileName>
  <name idx="1" type="char" size="1 20">Sin_RBE_SI_LESFRI_01</name>
  <type idx="1" type="char" size="1 6">Sparse</type>
  <andMethod idx="1" type="char" size="0 0" />
  <orMethod idx="1" type="char" size="0 0" />
  <defuzzMethod idx="1" type="char" size="1 3">COG</defuzzMethod>
  <impMethod idx="1" type="char" size="0 0" />
  <aggMethod idx="1" type="char" size="0 0" />
  <input idx="1" type="struct" size="1 1">
    <name idx="1" type="char" size="1 1">x</name>
    <range idx="1" type="double" size="1 2">0 10</range>
    <mf idx="1" type="struct" size="1 2">
      <name idx="1" type="char" size="1 7">A_{1;1}</name>
      <type idx="1" type="char" size="1 6">trapmf</type>
      <params idx="1" type="double" size="1 4">0.85 1.35 1.85 2.35</params>
      <paramsy idx="1" type="double" size="1 4">0 1 1 0</paramsy>
      <name idx="2" type="char" size="1 7">A_{1;2}</name>
      <type idx="2" type="char" size="1 6">trapmf</type>
      <params idx="2" type="double" size="1 4">3.95 4.45 4.95 5.45</params>
      <paramsy idx="2" type="double" size="1 4">0 1 1 0</paramsy>
    </mf>
  </input>
  <output idx="1" type="struct" size="1 1">
    <rule idx="1" type="struct" size="1 2">
      <antecedent idx="1" type="double" size="1 1">2</antecedent>
      <consequent idx="1" type="double" size="1 1">1</consequent>
      <weight idx="1" type="double" size="1 1">1</weight>
      <connection idx="1" type="double" size="1 1">1</connection>
      <antecedent idx="2" type="double" size="1 1">1</antecedent>
      <consequent idx="2" type="double" size="1 1">2</consequent>
      <weight idx="2" type="double" size="1 1">1</weight>
      <connection idx="2" type="double" size="1 1">1</connection>
    </rule>
  </output>
</root>
```

46. ábra. A rendszert leíró adatok tárolása sima szöveges állományban (bal oldal) és XML formátumban (jobb oldal)

Az FLTB további hiányossága, hogy bemeneti adatként (megfigyelésként) kizárólag egyelemű fuzzy (gyakorlatilag nem fuzzy) halmazokat fogad el. A keretrendszer fejlesztése során úgy a fuzzy, mint a nem fuzzy jellegű adatok bevitelét is lehetővé kívántam tenni, ezért egy olyan adatszerkezetet (ld. 47. ábra) és kapcsolódó szöveges állománybeli tárolási módot (ld. 48. ábra) dolgoztam ki, ami könnyű kezelhetősége mellett jól átláthatóan biztosítja a fuzzy megfigyelések kezelését. További szempont volt, hogy az általam kidolgozott adatszerkezet jól illeszkedik a már meglévő FLTB FIS struktúrához.



47. ábra. A megfigyelést leíró adatszerkezet

```
NumInputs=2
ObsName='Obs_2D_Trap_01'
[Observation]
OBS1='A^*_1':'trapmf',[0 0.1 0.2 0.3]*[0 1 1 0]
OBS2='A^*_2':'trapmf',[0.4 0.5 0.6 0.7]*[0 1 1 0]
```

```
<?xml version="1.0" ?>
- <root xml_tb_version="3.1" idx="1" type="struct" size="1 1">
  <FileName idx="1" type="char" size="0 0" />
  <mf idx="1" type="struct" size="1 2">
    <name idx="1" type="char" size="1 5">A^*_1</name>
    <type idx="1" type="char" size="1 6">trapmf</type>
    <params idx="1" type="double" size="1 4">0 0.1 0.2 0.3</params>
    <paramsy idx="1" type="double" size="1 4">0 1 1 0</paramsy>
    <name idx="2" type="char" size="1 5">A^*_2</name>
    <type idx="2" type="char" size="1 6">trapmf</type>
    <params idx="2" type="double" size="1 4">0.4 0.5 0.6 0.7</params>
    <paramsy idx="2" type="double" size="1 4">0 1 1 0</paramsy>
  </mf>
  <name idx="1" type="char" size="1 14">Obs_2D_Trap_01</name>
</root>
```

48. ábra. A megfigyelést leíró adatok tárolása sima szöveges állományban (bal oldal) és XML formátumban (jobb oldal)

A megfigyelések tárolására itt is kétfajta megoldást dolgoztam ki. A 48. ábra bal oldalán az egyszerű szöveges állományra láthatunk egy kétdimenziós példát, míg a jobb oldalon ugyanennek XML-es megvalósítása jelenik meg.

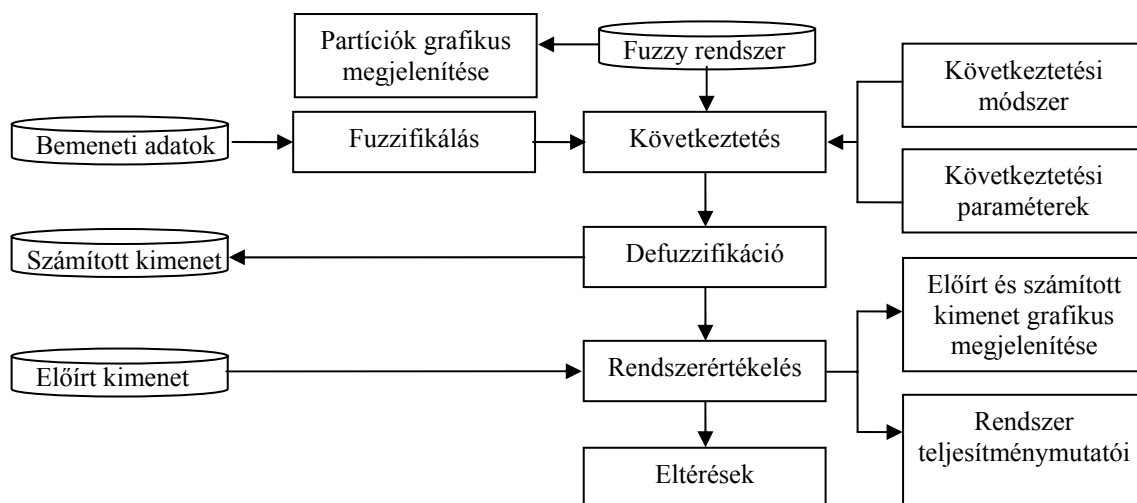
7.2. Az eljárásgyűjtemény és a rá épülő programok

A keretrendszerben megvalósított eljárásgyűjtemény jelenleg tizenegy szabály-interpoláción alapuló fuzzy következtetési módszer implementációját tartalmazza. Ezek a következők: KH [H40], stabilizált KH [H73], MACI [H69], IMUL [H80], CRF [H48], FIVE [H37], VKK [H75], GM (SCM+FERI+FPL) [H4], LESFRI [S13], FRIPOC [S12], VEIN [S26]. A FIVE-hoz szükséges számításokat végző függvényeket témavezetőm készítette, ezen kívül minden saját fejlesztés eredménye.

A fuzzy szabály-interpolációs módszereket implementáló, illetve a részfeladatokat megoldó függvények Matlabban közvetlenül meghívhatóak, így más alkalmazásba is beépíthetők. Mindemellett a szoftver által nyújtott szolgáltatások két grafikus felületű program segítségével is igénybe vehetők. Ezek a TestIt és a FRIT. Főbb jellemzőiket és használatukat a következő két szakaszban ismertetem.

7.2.1. TestIt

A TestIt program egy fuzzy rendszert tesztel éles (nem fuzzy) bemeneti és kimeneti értékek sorozatával. A menüből kiválasztott következtetési módszer segítségével minden bemeneti adatra kiszámolja a kimenetet és az előírt kimenettől való eltérést. Az eredmények alapján többfajta teljesítménymutatóval értékeli a rendszert, és méri a számítások időigényét, így az egyes módszerek az eltelt idő szempontjából is összehasonlíthatóak. A program grafikusan megjeleníti az egyes partíciókat, valamint a számított és az előírt kimeneti adatokat. A következtetési módszer paramétereit grafikus felületen állíthatóak. A program belső struktúráját a 49. ábra mutatja be.



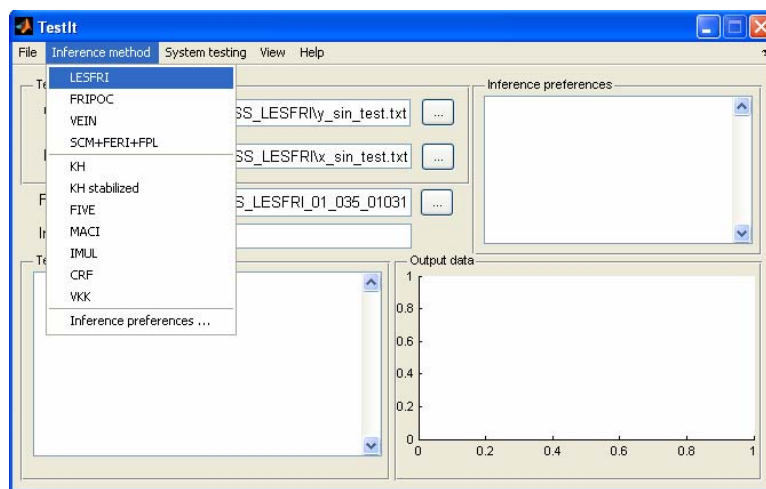
49. ábra. A tesztelő szoftver belső struktúrája

Tekintsük át a program működését egy példán keresztül. A *TestIt* paranccsal történő indítás után először beolvassuk az ellenőrzésre, illetve értékelésre használt adatokat. Ezeket két szöveges állomány formájában várja a program. Az első állomány a bemeneti adatokat tartalmazza (pl. *x_sin.txt* az 1. táblázatban), míg a második a kimeneti adatok (pl. *y_sin.txt* az 1. táblázatban) leírására szolgál. Az összetartozó, azaz egyetlen adatpontot leíró számok mindkét állományban azonos sorban találhatóak (ld. 1. táblázat). Többdimenziós bemenet esetén a bemeneti állomány mindegyik sorában több adat jelenik meg, amelyeket tabulátorjelek választanak el egymástól. Az állományneveket szabványos párbeszédablakban választhatjuk ki, amit menüből vagy az állománynév melletti nyomógombbal hívhatunk elő.

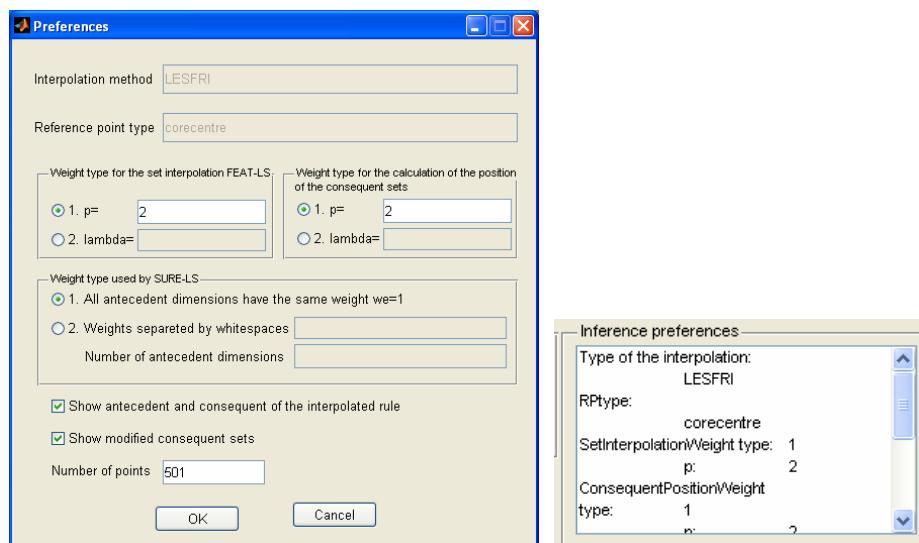
Az adatok beolvasását a rendszerleírás betöltése követi, majd az *Inference method* menüben kiválasztjuk a következtetési módszert (50. ábra). A módszer paramétereit egy párbeszédablak segítségével állítjuk be (51. ábra). A beállított értékek a főablak *Inference preferences* részében megtekinthetők. A fuzzy rendszer bemeneti és kimeneti partíciói a *View* menü segítségével jeleníthetők meg (52. ábra).

1. táblázat. Az első öt pontot leíró mintaadathalmaz

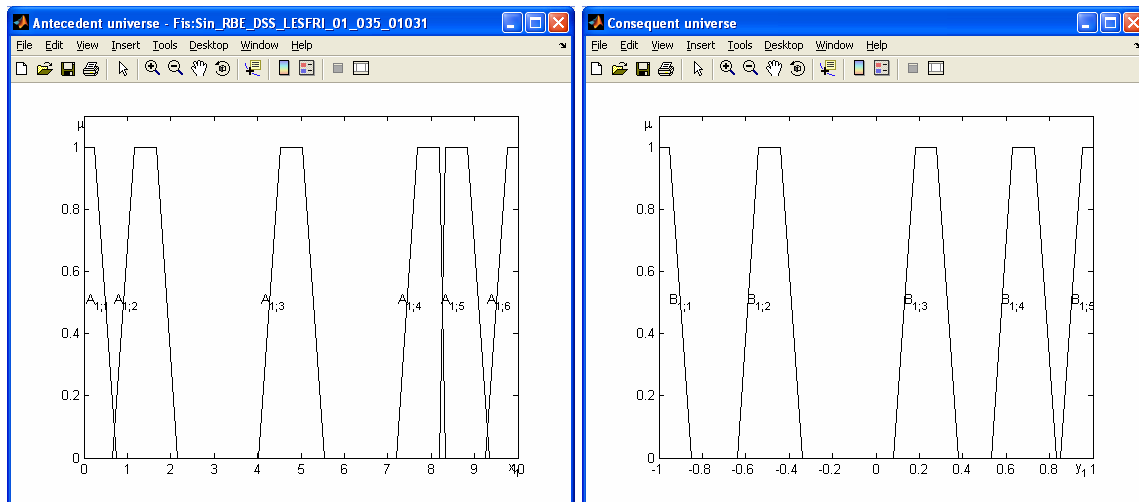
Adatpont sorszáma	x_sin.txt	y_sin.txt
1	0.0000000e+000	0.0000000e+000
2	1.0000000e-001	9.9833417e-002
3	2.0000000e-001	1.9866933e-001
4	3.0000000e-001	2.9552021e-001
5	4.0000000e-001	3.8941834e-001



50. ábra. Következtetési módszer kiválasztása a TestIt programban

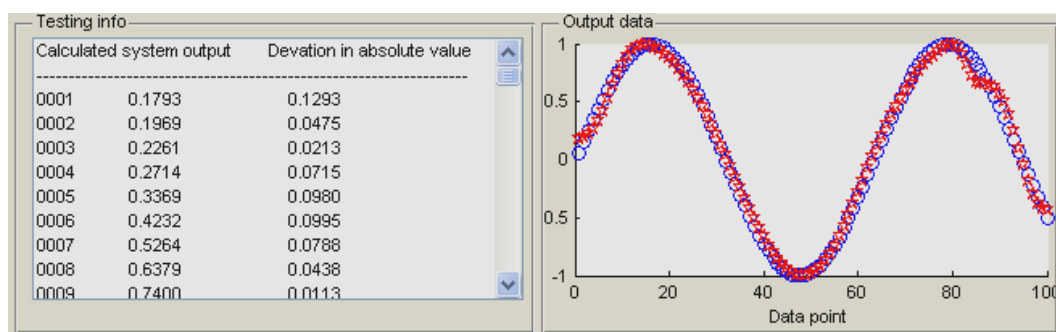


51. ábra. Következtetési módszer paramétereinek beállítása (bal oldal) és kijelzése (jobb oldal) a TestIt programban



52. ábra. Antecedens és konzekvens partíciók megjelenítése

A rendszer tesztelését a *System testing/Test system again given output data* menüponttal indíthatjuk. Ekkor a program a mintaadathalmaz összes bemeneti értékére kiszámolja a kimenetet, és összehasonlítja azt az előírt kimenettel. A *Testing info* ablakban szövegesen megkapjuk minden egyes adatpont esetén a számított kimenetet és az eltérést az előírt kimenettől abszolút értékben (53. ábra bal oldala), majd a teljesítménymutatók értékét. A program többfajta teljesítménymutatóval értékeli a rendszer jóságát, azaz az előírt és számított adatok közti eltérést. A teljesítménymutatók bemutatására a rendszergenerálási módszerekhez kapcsolódóan a 8.1.5. szakaszban kerül sor.

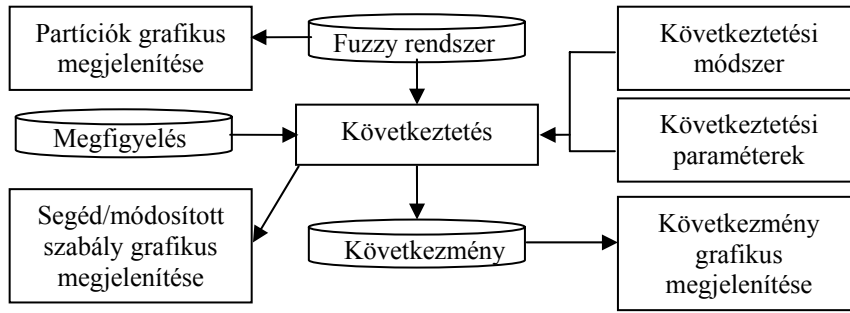


53. ábra. A fuzzy rendszer tesztelési eredményeinek szöveges és grafikus megjelenítése

A főablak jobb alsó részében (53. ábra jobb oldala) vizuálisan is összevethetjük a számított (piros csillag) és az előírt (kék kör) értékeket. A szoftver a rendszer kimenetét a COG defuzzifikációs módszer segítségével állítja elő. A számítási eredményeket (rendszer kimenete) a *File/Save calculated output data...* menüpont segítségével menthetjük el.

7.2.2. FRIT

A FRIT program fuzzy halmaz(ok) formájában megadott megfigyelésre különböző következtetési módszerekkel képes előállítani a következmény halmaz(oka)t. A be- és kimeneti partíciók grafikus megjelenítése mellett a kétlépéses módszerek által előállított segéd-, illetve módosított szabályok antecedens és konzekvens halmazai is megjeleníthetők. A TestIt programhoz hasonlóan az FLTB-vel kompatibilis sima szöveges formátum mellett XML alapú leírással is megadhatók, illetve menthetők a rendszert, a megfigyelést és az eredményt leíró adatok. A következtetés paramétereit grafikus felületen állíthatóak. A program belső struktúráját a 54. ábra mutatja be.



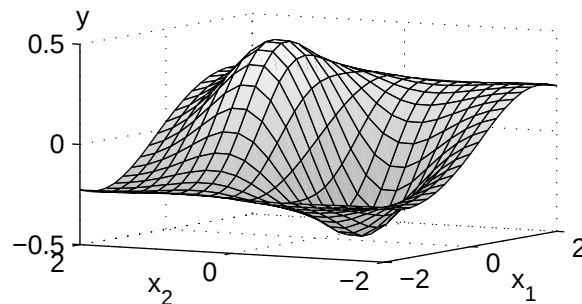
54. ábra. A FRIT szoftver belső struktúrája

A kidolgozott eljárásgyűjtemény, keretrendszer és adatstruktúra a [S27] és [S29] kiadványokban került publikálásra.

A szoftver használatának bemutatásához és az egyes fuzzy szabály-interpolációs módszerek által szolgáltatott eredmények eltérésének illusztrálásához egy fuzzy rendszert hoztam létre, majd két megfigyelés esetére kiszámoltam a következményt minden, a keretrendszerben implementált következtetési módszer segítségével. A fuzzy modellt az alábbi függvénnyel generált adatok alapján képeztem

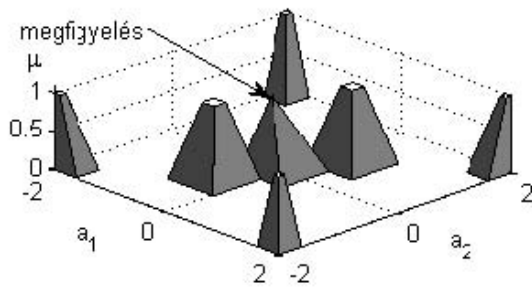
$$y = x_2 \cdot e^{-x_1^2 - x_2^2} + \sin(x_1)/4, \quad (82)$$

ahol $x_1 \in [-2, 2]$ és $x_2 \in [-2, 2]$. A bemenet és kimenet közötti kapcsolatot leíró felület a 55. ábrán látható.

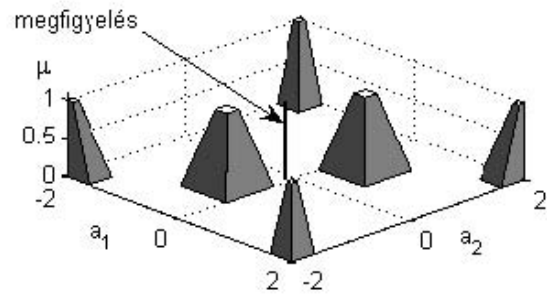


55. ábra. A rendszer által modellezett felület

A fuzzy modell csak trapéz alakú fuzzy halmazokat tartalmaz. A szabályok leírják a maximális és minimális kimenetet, valamint az x és y értelmezési tartományainak határértékeihez tartozó z értékeket. Ez utóbbiakat azért szükségesek, mert egyes szabály-interpolációs módszerek (pl. KH) megkívánják legalább két olyan szabály meglétét, amelynek feltétel részei közrefogják a megfigyelést minden antecedens dimenzióban. A kísérletek során először háromszög, majd ezt követően egyelemű (singleton) megfigyelést alkalmaztam. A 56. és 57. ábrák a szabálybázis antecedens terét és a megfigyeléseket mutatják be.



56. ábra. Szabály antecedensek és háromszög alakú megfigyelés



57. ábra. Szabály antecedensek és egyelemű megfigyelés

A továbbiakban az első megfigyelés és a FRIPOC módszer esetén végighaladunk a program használatának lépésein, majd röviden áttekintjük a többi esetben kapott eredményeket. Az alkalmazást a FRIT parancssal indítjuk, majd a *File/Load FIS...* menüpont segítségével betöltjük a fuzzy rendszer leírását. Ezt követően a *File/Load observation...* menüponttal kezdeményezzük a megfigyelést leíró állomány beolvasását. A két állománynév és az elérési utak megjelennek a főablakban (58. ábra)

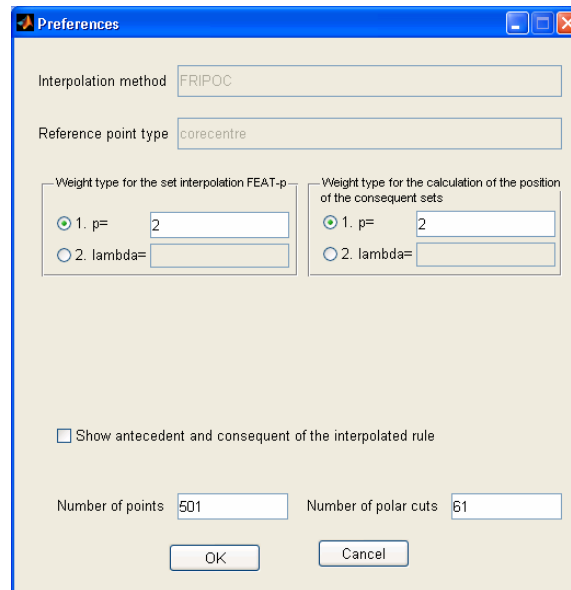
FIS data file	C:\doc\cikkek\Dolgozat\FRIT\FIS-Fuzzy rendszerek\system_1.fis
Obs data file	C:\doc\cikkek\Dolgozat\FRIT\Obs-Megfigyelesek\observation_1.obs

58. ábra. Rendszerleíró és megfigyelés adatállományok neveinek kijelzése

Következő lépésként a *Method selection/Fripoc* menüponttal kiválasztjuk az alkalmazni kívánt szabály-interpolációs módszert. Ekkor a főablak *Preferences* szerkesztőmezőjében megjelennek a kiválasztott eljárás alapértelmezett paraméterei (59. ábra), amelyeknek értékeit szükség esetén az *Interpolation/Preferences...* menüponttal hívható párbeszédablakban változtatunk meg (60. ábra). Itt a következtetés jellemzőin túl a segéd szabály halmazainak megjelenítését is engedélyezhetjük/tilthatjuk.

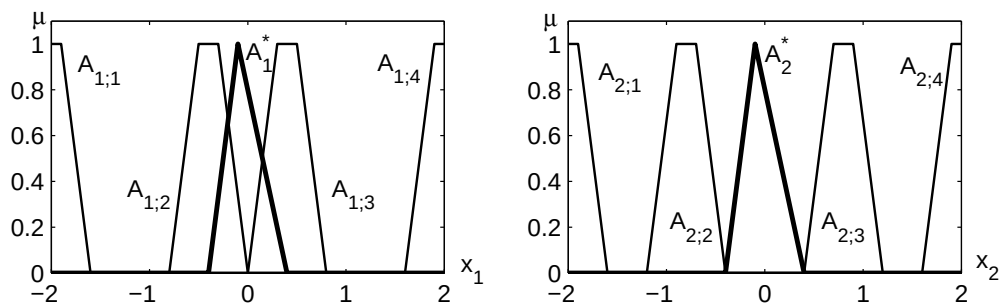
Interpolation method	FRIPOC
Preferences	
Type of the interpolation:	FRIPOC
Rtype:	corecentre
SetInterpolationWeight type:	1
p:	2
ConsequentPositionWeight type:	1
p:	2
ShowInterpolatedRule:	true
NumOfPoints:	501
NumOfpCuts:	61

59. ábra. Szabály-interpolációs eljárás paramétereinek kijelzése

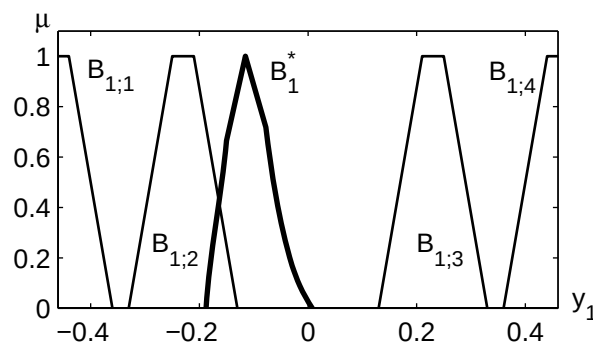


60. ábra. Szabály-interpolációs eljárás paramétereinek beállítása

A következtetést az *Interpolation/Interpolate* menüponttal indítjuk. Ekkor két új ablak jelenik meg. Az első az antecedens partíciókat és a megfigyelést ábrázolja, míg a másodikban a konzekvens partíciót és a következményt láthatjuk. A 61. és 62. ábrákon a két ablak tartalmát láthatjuk a feliratok elrendezése után.



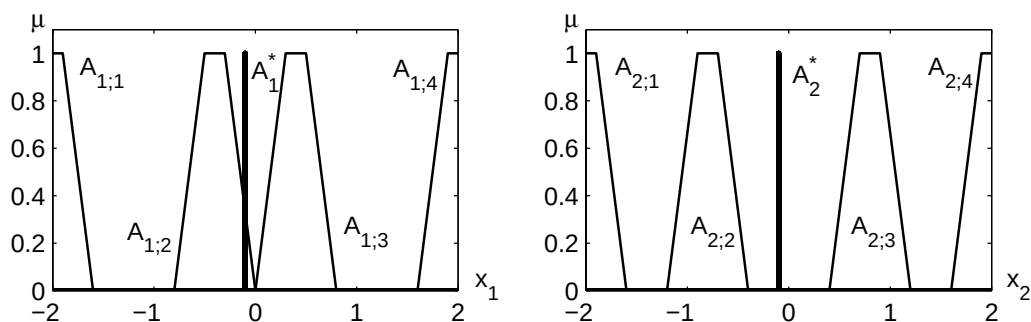
61. ábra. A mintafeladat antecedens partíciói és a háromszög alakú megfigyelés halmazok (vastag vonallal)



62. ábra. A konzekvens partíció és a következmény (vastag vonallal) háromszög alakú megfigyelések és FRIPOC alkalmazása esetén

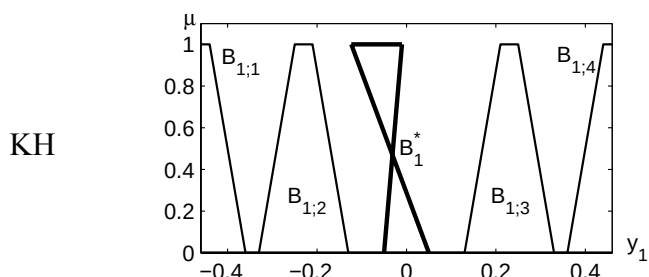
A kapott eredményt a *File/Save results...* menüpont segítségével menthetjük el. A következő példában röviden áttekintjük, hogy a mintául választott két megfigyelés mellett az egyes

szabály-interpolációs módszerek milyen eredményre vezettek. A 63. ábra bemutatja az antecedens partíciókba elhelyezett egyelemű megfigyeléseket.

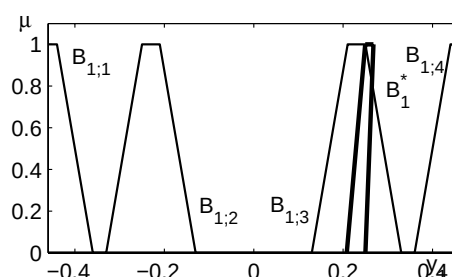


63. ábra. A mintául választott antecedens partíciók és egyelemű megfigyelés halmazok (vastag vonallal)

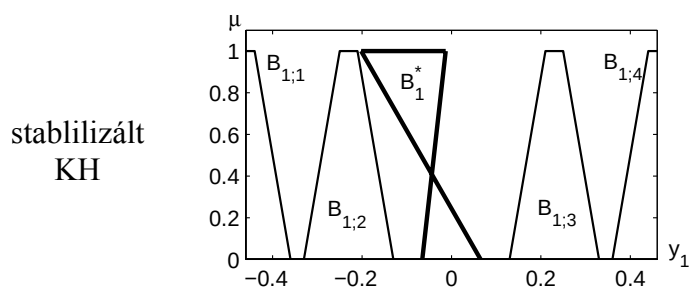
A kísérletek eredményei háromszög és egyelemű halmaz megfigyelések esetén



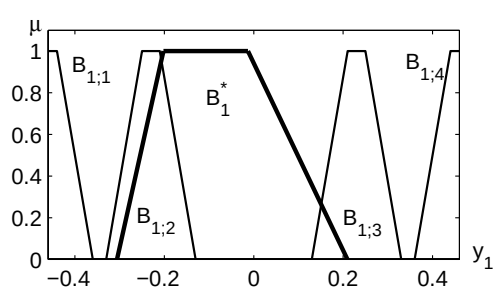
64. ábra. Konzekvens partíció és következmény (vastag vonallal) háromszög alakú megfigyelés esetén KH módszernél



65. ábra. Konzekvens partíció és következmény (vastag vonallal) egyelemű megfigyelés esetén KH módszernél

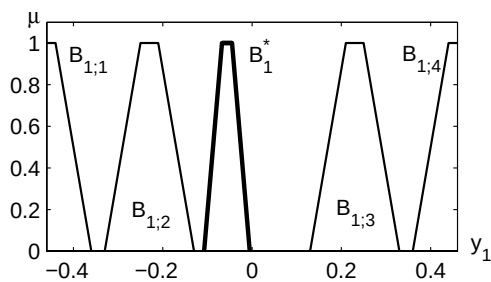


66. ábra. Konzekvens partíció és következmény (vastag vonallal) háromszög alakú megfigyelés esetén stab. KH módszernél

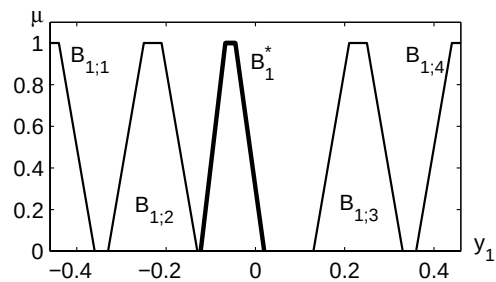


67. ábra. Konzekvens partíció és következmény (vastag vonallal) egyelemű megfigyelés esetén stab. KH módszernél

MACI

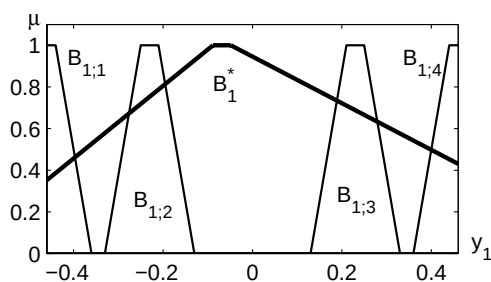


68. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
MACI módszernél

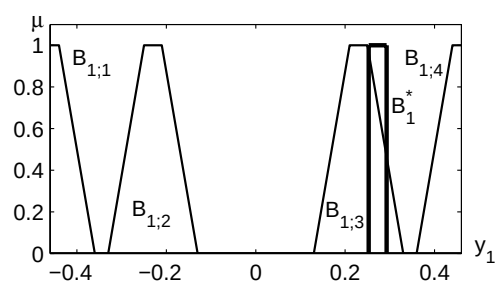


69. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén MACI
módszernél

IMUL

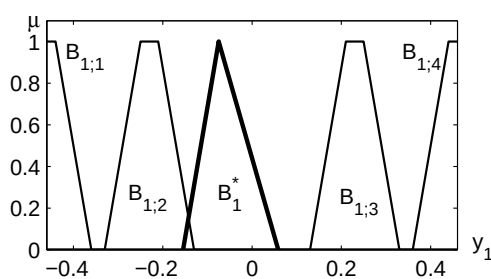


70. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
IMUL módszernél

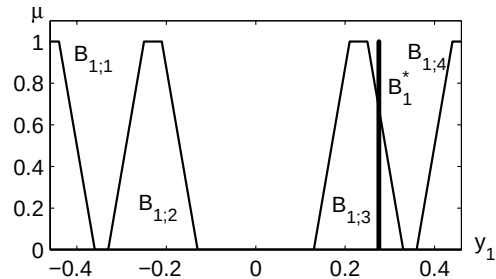


71. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén IMUL
módszernél

CRF

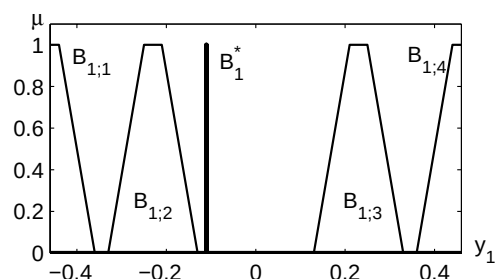
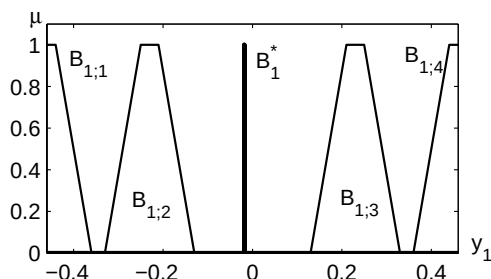


72. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
CRF módszernél



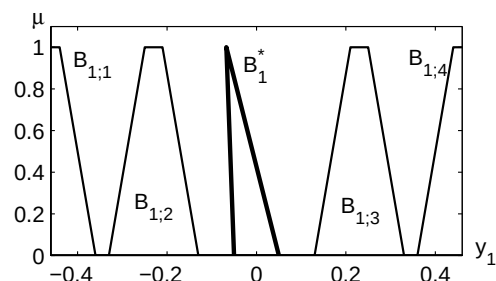
73. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén CRF
módszernél

FIVE

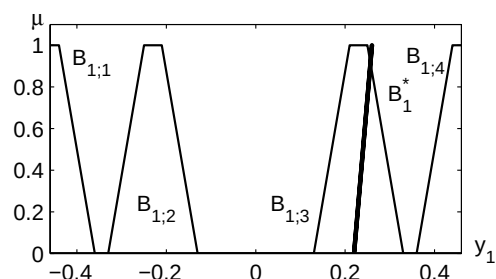


74. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
FIVE módszernél

VKK

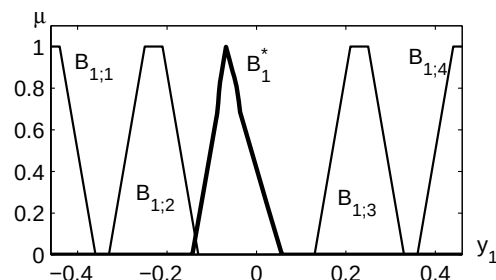


75. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén FIVE
módszernél

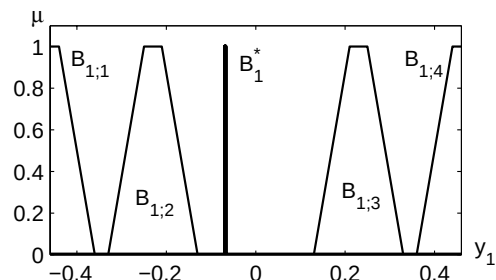


76. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
VKK módszernél

SCM+FPL

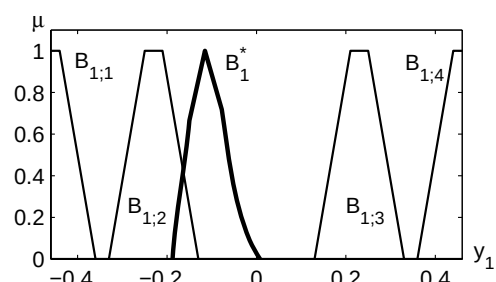


77. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén VKK
módszernél

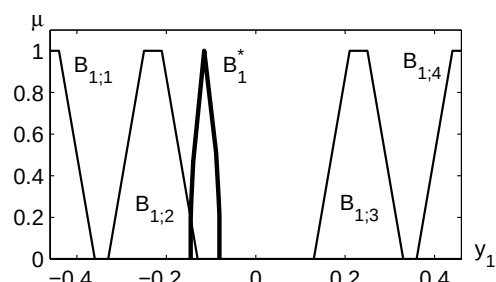


78. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
SCM+FPL módszernél

FRIPOC



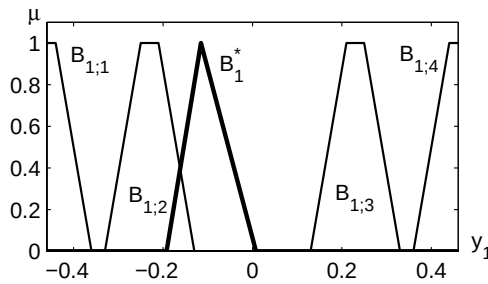
79. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén SCM+FPL
módszernél



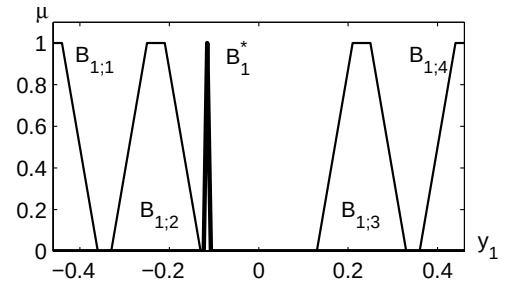
80. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
FRIPOC módszernél

81. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén FRIPOC
módszernél

LESFRI

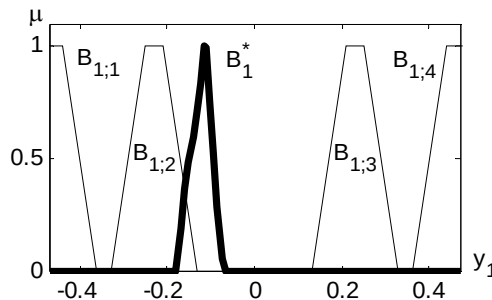


82. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
LESFRI módszernél

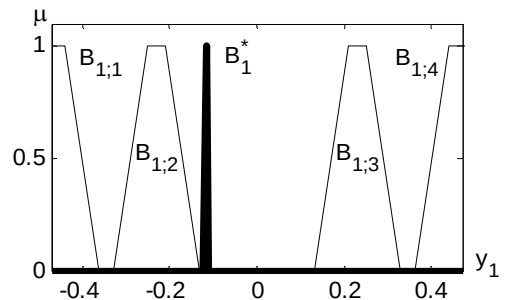


83. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén LESFRI
módszernél

VEIN



84. ábra. Konzekvens partíció és
következmény (vastag vonallal)
háromszög alakú megfigyelés esetén
VEIN módszernél



85. ábra. Konzekvens partíció és
következmény (vastag vonallal)
egyelemű megfigyelés esetén VEIN
módszernél

A 64-85. ábrák vastag vonallal jelölik az egyes szabály-interpolációs eljárásokkal számított következményeket a konzekvens partícióban. Az egyes módszerekkel kapott eltérő eredmények mellett a következő jellegzetességek figyelhetők meg:

- van olyan eset, amikor a KH (64. és 65. ábra), stabilizált KH (66. ábra) és a VKK (76. és 77. ábra) módszerek érvénytelen fuzzy halmazt eredményeznek;
- a FRIPOC (80. és 81. ábra) módszer nem őrzi meg a partícióra jellemző halmazalakot;
- az egyelemű megfigyelés esetén a CRF (73. ábra), a VKK (77. ábra) és az IMUL (71. ábra) módszereknél a következmény halmaz nem esik teljes mértékben a megfigyelést közrefogó szabályok konzekvenséi közé;
- a háromszög alakú megfigyelés esetén az IMUL (70. ábra) eljárással számolt következmény tartója túl széles lett, túllépte a partíció határait.

7.2.3. Súly

Az általam kidolgozott keretrendszerhez a FRIT programból és közvetlenül is elérhető részletes HTML súly tartozik. A szöveges leírás (ld. 86. ábra) mellett egy függőségi gráf (ld. 87. ábra) segítségével a függvények kapcsolatrendszere vizuálisan is nyomon követhető. Bármelyik függvény nevére kattintva megjelenik annak leírása.

Home > FRIT > LESFRI.m

LESFRI

PURPOSE [↑](#)

Fuzzy Rule Interpolation based on the method of least squares

SYNOPSIS [↑](#)

function out=LESFRI(OriginalObs,OriginalFis,params)

DESCRIPTION [↑](#)

```

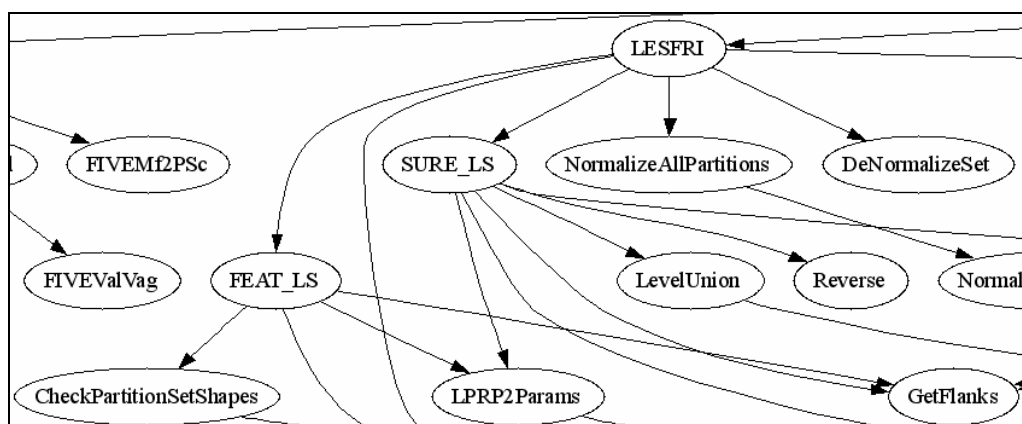
Fuzzy Rule Interpolation based on the method of least squares

function out=LESFRI(OriginalObs,OriginalFis,params)

Calculate the conclusion by the help of Fuzzy Rule Interpolation
based on the method of least squares
-----
Parameters
-----
OriginalFis - structure describing the fuzzy inference system.
              The partitions are normalized.
OriginalObs - structure describing the observation corresponding to the
              normalized partitions.
params - structure containing information about the parameters of the
         method. It should have the
         following fields:
         InterpolationType    LESFRI

```

86. ábra. Részlet a keretrendszer súgójának egy lapjából



87. ábra. Részlet a függvények kapcsolatrendszerét leíró függőségi gráfból

8. Fuzzy következtető rendszer automatikus generálása

Mint ahogyan az a 2.3. fejezetben már bővebben kifejtésre került, az antecedens dimenziók és a bennük megjelenő nyelvi értékek számának emelkedése az antecedens tér lefedéséhez szükséges szabálysorszám robbanásszerű növekedésével jár (ld. (23) és 8. ábra). A problémára megoldást szolgáltatathat a ritka (nem fedő) szabálybázisok és a szabály-interpoláción alapuló következtetési módszerek alkalmazása [H40][H41].

A fuzzy modell kialakítása során alacsony szabálysorszámú szabálybázist alapvetően két módon nyerhetünk. Az első megközelítés [H47] egy teljesen fedő ($\epsilon > 0$ az (22)-ben) szabálybázisból indul ki, majd a nem releváns szabályok elhagyásával vagy egymáshoz közeli (hasonló) szabályok összevonásával csökkenti a szabályok számát. Ez utóbbi csoportba tartoznak a Botzheim, Cabrita, Kóczy és Ruano [H8], Botzheim, Hámori és Kóczy [H9], valamint Kóczy, Botzheim és Gedeon [H39] által javasolt evolúciós algoritmusokon alapuló megoldások.

A második megközelítés közvetlenül olyan szabálybázist állít elő, ami nem fedi teljes mértékben az antecedens teret. Az utóbbi a csoportba tartozó módszerek általában az alábbi két megközelítés egyikével dolgoznak.

- Az ún. optimális fuzzy szabályok azonosítására törekszenek [H32][H33].
- Fuzzy klaszterezés segítségével hozzák létre a szabályokat [H12][H15][H29][H66][H72][H81].

A továbbiakban először az általam kidolgozott új (harmadik) megközelítést alkalmazó RBE elvet és a rá alapozott RBE-DSS valamint RBE-SI eljárásokat ismertetem, majd ezt követően bemutatom az általam módosított fuzzy klaszterezésen alapuló ACP eljárást. Ritka szabálybázissal rendelkező fuzzy rendszer mintaadatok alapján történő generálásának támogatására egy eljárásgyűjteményt (ToolBoxot) dolgoztam ki Matlab keretrendszerben, ami a klaszterezésen alapuló ACP módszert, valamint az iteratív szabálybázis kiterjesztés elvén alapuló két módszer (RBE-DSS és RBE-SI) [S22] megvalósítását is tartalmazza. A fejezet utolsó részében röviden áttekintem a szoftver felépítését, funkcionalitását és használatát.

8.1. Rendszergenerálás iteratív szabálybázis kiterjesztéssel

Az alábbiakban ismertetésre kerülő két fuzzy modell-identifikációs módszer alkalmazásával az összetartozó ki- és bemeneti értékeket leíró adathalmazból automatikusan generálható fuzzy rendszer, amelynek szabálybázisa az esetek többségében ritka lesz. A két módszer alapvetően az új nyelvi értékek alakjának meghatározási módjában tér el egymástól.

Az egyszabályos következtetési módszerek ismertetéséhez hasonlóan itt is csak SISO és MISO rendszerekkel foglalkozunk, a több kimenetes (xxMO) rendszerek szabálybázisát egykimenetű rendszerek szabálybázisai aggregációjának tekintjük. A szabálybázist generáló módszerek tárgyalása során csak kompakt fuzzy halmazokat alkalmazunk, azaz olyan nyelvi értékeket, amelyeknek korlátos a tartója [H26].

Bár a módszerek nem kötődnek egy adott halmazalak típushoz, de az egyszerűség kedvéért a továbbiakban a módszerek ismertetése során az alkalmazásokban gyakori trapéz alakú nyelvi értékekkel fogunk dolgozni. A választott halmazalak további előnye, hogy a háromszög és egyértékű alakok leírására is képesek. A továbbiakban adatsornak nevezzük azt az összetartozó $x^k = \{x_1^k, \dots, x_N^k, x_{N+1}^k\}$, $k = 1, \dots, M$ adathalmazt, amely minden antecedens dimenzióban tartalmaz egy bemeneti értéket ($x_1^k, \dots, x_N^k$), valamint tartalmazza az adott bemenetek esetén elvárt egyetlen kimeneti értéket (x_{N+1}^k). M -el jelöltem a mintaadathalmaz

adatsorainak számát, N -el az antecedens dimenziók számát, és az egyszerűbb jelölés érdekében a konzekvens dimenzióra $N+1$ -ként hivatkoznak.

8.1.1. A szabálybázis kiterjesztés elve

A szabálybázis kiterjesztés (RBE-Rule Base Extension) [S22] elve alatt azt értem, hogy a fuzzy rendszert két lépésben hozzuk létre az alábbiak szerint. A kiindulópont egy üres szabálybázis és egy tanító ponthalmaz, ami összetartozó be- és kimeneti értékek sorozatát tartalmazza. Az RBE első lépésében egy nyers szabálybázist hozunk létre, ami két szabályt tartalmaz, egyet a maximális és egyet a minimális kimenet leírására. A második lépésben egy iteratív hangolási folyamat keretében azonosítjuk a nyelvi értékek paramétereit, valamint új fuzzy halmazokat és szabályokat hozunk létre a rendszer teljesítményének javítása érdekében.

Az első lépést minden dimenzióban az alapértelmezett mag ($W_{C,i}$) és tartó ($W_{S,i}$) szélességek meghatározásával kezdjük. Értékeiket az adott dimenzióban érvényes értelmezési tartomány szélessége (terjedelem - DR_i) arányában határozzuk meg

$$W_{C,i} = C_{WC} \cdot DR_i, 1 \leq i \leq N+1, \quad (83)$$

$$W_{S,i} = C_{SC} \cdot DR_i, 1 \leq i \leq N+1, \quad (84)$$

ahol I -től N -ig terjedően az antecedens dimenziókat számoltuk, és az egyszerűbb jelölés érdekében a konzekvens dimenzióra $N+1$ -ként hivatkoztunk. A terjedelmet a korlátok ismeretében számítjuk a

$$DR_i = x_{i,\max} - x_{i,\min}, 1 \leq i \leq N+1 \quad (85)$$

képlettel.

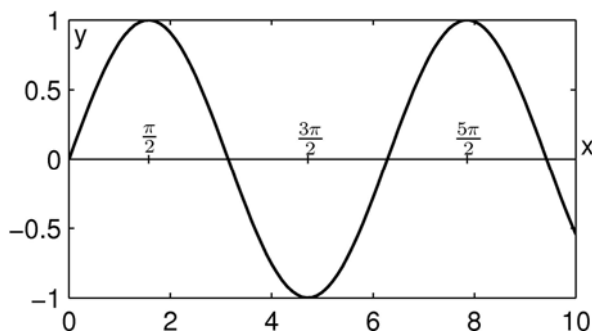
A tartományok alsó ($x_{i,\min}$) és felső ($x_{i,\max}$) korlátait előírt értékek vagy a rendelkezésre álló mintaadatok határozzák meg. Mint később látni fogjuk az RBE-SI eljárás esetében az alapértelmezett szélesség értékek csak az első két szabály előállításánál szükségesek.

A módszer következő lépéseként az első két szabály definiálásával egy kiinduló szabálybázist hozunk létre. A két szabály célja a minimális és a maximális kimeneti értékeket jellemző relációk leírása. Ezen két szabály megalkotásakor megkeressük a két kimeneti szélsőértéket, majd keresünk egy-egy rájuk illeszkedő adatsort. Amennyiben több adatsor is ugyanazt a kimeneti minimumot/maximumot tartalmazza, akkor azt választjuk, amelyik közelebb van az antecedens tér széleihez.

Tegyük fel példaként, hogy a (86) függvény által leírt görbét (88. ábra) modellező fuzzy rendszer előállítása a célunk. Ekkor két maximumunk van, éspedig a $\pi/2$ és az $5\pi/2$ értékeknél. Ezek közül az elsőt ($\pi/2$) választjuk ki, mert az van közelebb az x alsó korlátjához.

$$y = \sin x, \quad x \in [0, 10] \quad (86)$$

Ezt követően fuzzy halmazokat rendelünk az adatokhoz úgy, hogy a halmazok referencia pontjai essenek egybe a két adatsor értékeivel. Az alapértelmezett mag és tartó szélességeket alkalmazzuk mindegyik dimenzióban. A fuzzy halmazok definiálása során figyelembe vesszük a 8.1.2. szakaszban ismertetett megkötéseket. Ezek a szélesség értékek módosulását eredményezhetik.



88. ábra. Modellezett függvénykapcsolat

A fuzzy halmazok számának túlzott növekedésének elkerülése érdekében a hasonló (egymáshoz közel álló vagy azonos) nyelvi értékeket egyesítjük az antecedens dimenziókban. A halmaz egyesítéseket az alábbi két metaszabály alapján határozzuk meg.

1. *metaszabály*. Amennyiben két fuzzy halmaz referencia pontja egy megadott $d_{i,min}$ határértéknél közelebb kerül egymáshoz, akkor ezeket egyesítjük. A $d_{i,min}$ határértéket a partíció terjedelmének arányában adjuk meg

$$d_{i,min} = C_d \cdot DR_i, \quad (87)$$

ahol C_d dimenziótól független együttható. A kísérletek során általában $C_d = 0,01$ értékkel dolgoztam.

2. *metaszabály*. Amennyiben két fuzzy halmaz paramétereinek átlagos eltérése egy megadott $dp_{i,min}$ határértéknél kisebb, akkor ezeket egyesítjük. A $dp_{i,min}$ határértéket a partíció terjedelmének arányában adjuk meg

$$dp_{i,min} = C_{dp} \cdot DR_i, \quad (88)$$

ahol C_{dp} dimenziótól független együttható. A kísérletek során általában $C_{dp} = 0,005$ értékkel dolgoztam.

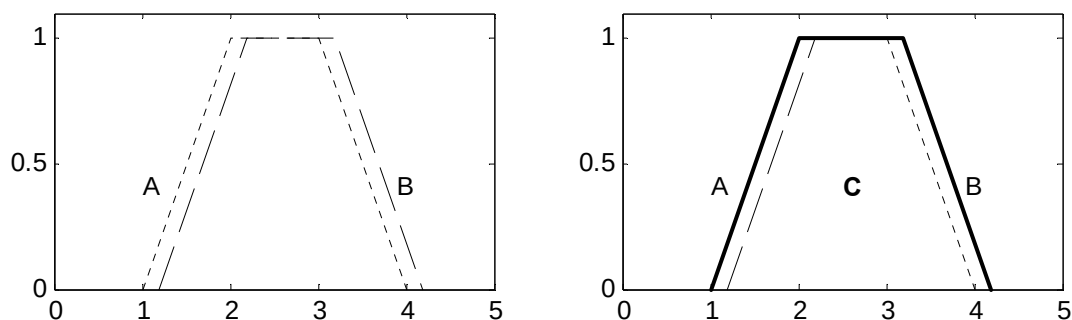
A fuzzy halmazok egyesítését a Kóczy [H38] által javasolt CNF unió műveletével valósítjuk meg, ami a két halmaz konvex burkaként határozza meg uniójukat. A jelen esetben, ahol minden nyelvi érték CNF és trapéz alakú, az uniójuk ugyancsak trapéz, melyet két α -vágatával (0 és 1 szinten) határozhatjuk meg a következőképpen

$$\inf \{ [C]_\alpha \} = \min \{ \inf \{ [A]_\alpha \}, \inf \{ [B]_\alpha \} \}, \alpha = 0, 1, \quad (89)$$

$$\sup \{ [C]_\alpha \} = \max \{ \sup \{ [A]_\alpha \}, \sup \{ [B]_\alpha \} \}, \alpha = 0, 1, \quad (90)$$

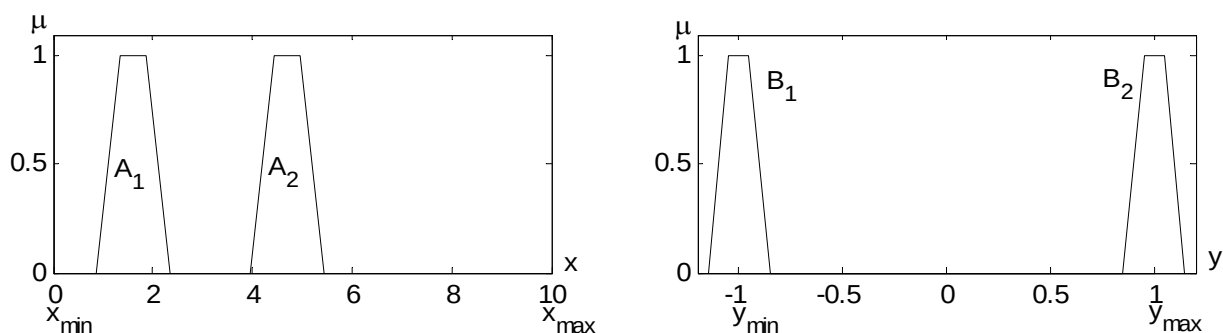
ahol A és B az eredeti két nyelvi érték, míg C az unióval előállított fuzzy halmaz. A 89. ábra két egymáshoz közel álló fuzzy halmaz CNF unióját szemlélteti.

Az eddig meghatározott négy empirikus előre meghatározott érték (két szélesség és két határérték) hangolási lehetőséget biztosít a rendszer számára. A 90. ábra bemutatja a fentiek alapján létrehozott antecedens és konzekvens ritka partíciókat a 88. ábrán látható görbét modellező fuzzy rendszer esetén.



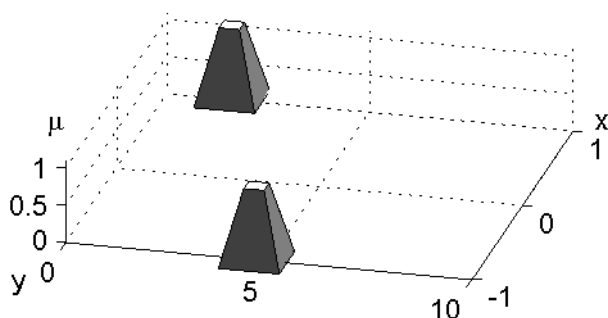
89. ábra. Két egymáshoz közel álló fuzzy halmaz (bal oldal) és CNF uniójuk (jobb oldal)

A rendszergenerálás során a $C_{WC} = 0,05$, $C_{WS} = 0,15$ és $C_d = 0,01$ értékekkel dolgoztam. A halmaz-interpolációs módszerek tárgyalása során a 4.1. szakaszban ismertetett megfontolásokat követve itt is megengedtük, hogy a konzekvens partíció két fuzzy halmazának egy része „kilógjon” az $[y_{\min}, y_{\max}]$ intervallumból.



90. ábra. Antecedens és konzekvens partíciók

A kezdő nyelvi értékekre alapozva létrehozuk az első két szabályt, amit a 91. ábra illusztrál. Mindkét szabálynak egy-egy csonka gúla felel meg, amit a szabályban szereplő nyelvi értékek definiálnak. Itt jól megfigyelhető, hogy az y tengely irányában a két konzekvens halmaz, illetve a nekik megfelelő gúlák egy része „kilóg” az $[y_{\min}, y_{\max}]$ intervallumból.



91. ábra. Az első két szabályt tartalmazó szabálybázis

Az RBE első lépését meghatározó algoritmus vázlatát a 92. ábra ismerteti. Az x a mintaadathalmaz bemenő adatait tartalmazó mátrix, amelynek minden sora egy adatpontnak és minden oszlopa egy dimenzióknak felel meg. Az y a mintaadathalmaz kimenő adatait tartalmazó oszlopvektor. Sorainak száma azonos az x mátrix sorainak számával. A FIS

azonosító a fuzzy rendszert leíró komplex adatstruktúrát jelöli. Részletes ismertetése a 7. fejezetben található.

FIS létrehozása: dimenzióként alsó és felső korlátok és általános adatok
 Minden dimenzióban: DR kiszámítása
 Maximális és minimális kimenetet leíró adatsorok megkeresése
 Minden dimenzióban: nyelvi értékek létrehozása a két adatsorhoz
 Minden dimenzióban: 1. és 2. metaszabály alkalmazása
 Minden dimenzióban: nyelvi értékek beillesztése a FIS-be
 Első két szabály létrehozása és beillesztése a FIS-be

92. ábra. Az RBE első lépésének algoritmusai

Az RBE második lépésének bemutatását a nyelvi értékek paraméterezésének és más kapcsolódó megfontolások tárgyalása után folytatjuk a 8.1.3. szakaszban.

8.1.2. Fuzzy halmazok paraméterezése és kötöttségek

A fejezet címében megjelenő paraméterezés szó arra utal, hogy abban az esetben, amikor egy fuzzy halmaz alakzattípusra jellemző tagsági függvény leírható analitikusan egy

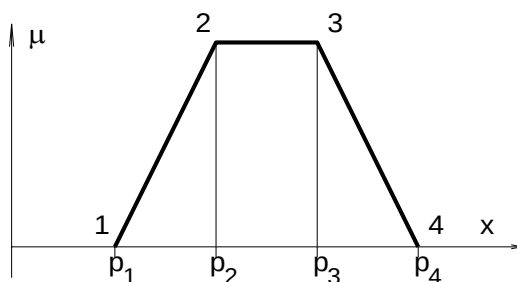
$$\mu_A(x) = f(p_1, \dots, p_n, x) \quad (91)$$

alakú függvény segítségével, akkor bizonyos konkrét értékek, azaz $(p_1, \dots, p_n)$ paraméterek ismeretében egy konkrét fuzzy halmaz állítható elő. A paraméterek száma (n) és értelmezése az adott alakzattípushoz kötődik.

Az alábbiakban áttekintünk néhány paraméterezési lehetőséget. Mivel a rendszergenerálás ismertetése során trapéz alakú CNF fuzzy halmazokkal dolgozunk, ezért elsősorban olyan módszereket választottam, amelyek a trapéz alakzathoz kötődnek. Egyedül az utolsó, a referencia pont alapú az alakzatfüggetlen közülük. A kapcsolódó megkötések szintén ismertetésre kerülnek. A továbbiakban a mag középpontjának megfelelő alaphalmaz elemet választom referencia pontnak

8.1.2.1. Töréspontok

Az első és talán legkézenfekvőbb paraméterezési megoldás (pl. [H66], [H72]) a töréspontok abszcisszáinak alkalmazása. Mivel csak trapéz alakú és CNF halmazokkal dolgozunk, azaz a töréspontok ordináta értékei ismertek, ezért a helyzet és az alak leírásához elegendő a négy csúcspont horizontális koordinátája. Az egyszerűség és az áttekinthetőség érdekében a jelen szakaszban a paraméterek indexében csak ott jelölöm a halmaz sorszámát (azonosítóját), ahol ez feltétlenül szükséges.



93. ábra. Töréspontok abszcissaértékei paraméterként

A csúcspontokat órajárás szerint sorszámozva (ld. 93. ábra) a következő megkötéseket kell figyelembe venni a hangolás során.

- A második csúcsponttól kezdődően egyetlen paraméter sem lehet kisebb, mint az őt megelőző paraméter.

$$p_k \geq p_{k-1}, 1 < k \leq 4 \quad (92)$$

- A referencia pont az alaphalmaz adott dimenziójában érvényes alsó és felső korlátok közzé kell essen.

$$x_{\min} \leq RP(A) = \frac{p_2 + p_3}{2} \leq x_{\max} \quad (93)$$

A halmaz-interpolációs módszerek tárgyalása során a 4.1. szakaszban ismertetett megfontolásokat követve itt is megengedett, hogy a konzekvens partíció két fuzzy halmazának egy része az $[x_{\min}, x_{\max}]$ intervallumon kívülre kerüljön.

- Olyan esetekben, amikor a fuzzy rendszer működése során nem alkalmazzuk a nyelvi értékek eltolásának elvét (ld. 4.1. szakasz), a korlátok szigorú értelmezése szükséges, ami az alábbi megkötésekkel fejezhető ki

$$p_1 \geq x_{\min} \text{ és } p_4 \leq x_{\max} . \quad (94)$$

- Az egyes nyelvi értékek és a rájuk épülő szabályok későbbi értelmezhetőségének elősegítése érdekében nem engedhetjük meg, hogy egy fuzzy halmaz tartója átfedje egy másik halmaz magját. A j -edik halmaz esetében ezt a következőképpen fejezhetjük ki

$$p_{j,1} \geq p_{j-1,3}, j = 2, \dots, n, \quad (95)$$

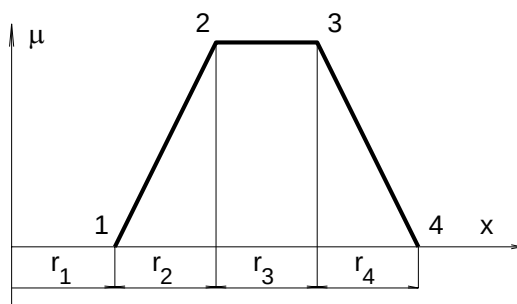
$$p_{j,4} \leq p_{j+1,2}, j = 1, \dots, n-1, \quad (96)$$

ahol n a nyelvi értékek száma a partícióban. Érvényes fuzzy halmazokat feltételezve a fenti megkötés természetesen azt is magában foglalja, hogy a halmaz magja nem fedheti át egy másik halmaz magját.

A (95) és (96) a 8.1.2.3. szakaszban ismertetett Ruspini partíciónál lényegesen gyengébb korlátozás, ugyanis itt lehetővé tesszük ritka partíciók létrejöttét is.

8.1.2.2. Relatív távolságok

Egy trapéz alakú fuzzy halmazt leírhatunk csúcspontjainak relatív távolságait felhasználva is. Ebben az esetben az első paraméter (r_1 a 94. ábrán), ami azonos a nagyalap bal szélső pontjának abszcisszájával, meghatározza a halmaz helyzetét. A fennmaradó három csúcspont helyzetét egy-egy olyan nemnegatív értékkel azonosítjuk be, ami megadja a csúcspont előző csúcspontához mért vízszintes irányú távolságát. Az egyszerűség és az áttekinthetőség érdekében a jelen szakaszban a paraméterek indexében csak ott jelölöm a halmaz sorszámát (azonosítóját), ahol ez feltétlenül szükséges.



94. ábra. Relatív koordináták alkalmazása paraméterként

A 8.1.2.1. szakaszban alkalmazott számozás- és jelölésmóddal dolgozva az első két paraméterezési mód közötti kapcsolatot a következőképpen fejezhetjük ki.

$$r_k = \begin{cases} p_k, & k = 1 \\ p_k - p_{k-1}, & 1 < k \leq 4 \end{cases} \quad (97)$$

Relatív távolságokat alkalmazó paraméterezés esetén a paraméter-azonosítási folyamat során érvényes megkötések az alábbiak.

- A másodiktól kezdődően minden paraméter nulla vagy pozitív értékű kell legyen.

$$r_k \geq 0, \quad 1 < k \leq 4 \quad (98)$$

- A referencia pont az alaphalmaz adott dimenziójában érvényes alsó és felső korlátok közé kell essen.

$$x_{\min} \leq RP(A) = r_1 + r_2 + \frac{r_3}{2} \leq x_{\max} \quad (99)$$

A halmaz-interpolációs módszerek tárgyalása során az 4.1. szakaszban ismertetett megfontolásokat követve itt is megengedett, hogy a konzekvens partíció két fuzzy halmazának egy része az $[x_{\min}, x_{\max}]$ intervallumon kívülre essen.

- Olyan esetekben, amikor a fuzzy rendszer működése során nem alkalmazzuk a nyelvi értékek eltolásának elvét (ld. 4.1. szakasz), a korlátok szigorú értelmezése szükséges, ami az alábbi megkötésekkel fejezhető ki

$$r_1 \geq x_{\min} \text{ és } r_1 + r_2 + r_3 + r_4 \leq x_{\max}. \quad (100)$$

- Az egyes nyelvi értékek és a rájuk épülő szabályok jobb értelmezhetőségének elősegítése érdekében nem engedhetjük meg, hogy egy fuzzy halmaz tartója átfedje egy másik halmaz magját. A j -edik halmaz esetében ezt a következőképpen fejezhetjük ki

$$r_{j,1} \geq r_{j-1,1} + r_{j-1,2} + r_{j-1,3}, j = 2, \dots, n, \quad (101)$$

$$r_{j,1} + r_{j,2} + r_{j,3} + r_{j,4} \leq r_{j+1,1} + r_{j+1,2}, j = 1, \dots, n-1, \quad (102)$$

ahol n a nyelvi értékek száma a partícióban. Érvényes fuzzy halmazokat feltételezve a fenti megkötés természetesen azt is magában foglalja, hogy a halmaz magja nem fedheti át egy másik halmaz magját.

A (95) és (96) a 8.1.2.3. szakaszban ismertetett Ruspini partíciónál lényegesen gyengébb korlátozás, ugyanis itt lehetővé tesszük ritka partíciók létrejöttét is.

8.1.2.3. Ruspini partíció megőrzése

Az automatikusan generált fuzzy halmazok egyik előnyös tulajdonsága az, hogy míg a neurális hálók fekete dobozként viselkednek, addig a fuzzy rendszerekből az ember számára is könnyen értelmezhető tudás nyerhető ki szabályok formájában. Ez a tulajdonság akkor használható ki a legjobban, ha a rendszer Ruspini partíciókból [H61] épül fel, azaz minden be- és kimeneti dimenzió minden pontjában a tagsági értékek összege 1,

$$\sum_{j=1}^n \mu_{A_j}(x) = 1, \forall x \in [x_{\min}, x_{\max}], \quad (103)$$

ahol $x_{\min}$ és $x_{\max}$ az alaphalmaz alsó illetve felső korlátja és n a fuzzy halmazok száma az adott dimenzióban.

Ruspini partíciók esetén nyelvi értékenként elegendő két paraméter a trapéz alakú fuzzy halmazok leírására. Ezek a mag két végpontjának az abszcisszáját tárolják

$$v_{j,1} = \begin{cases} x_{\min} & \text{ha } j = 1 \\ \inf\{[A_j]_1\} & \text{ha } j = 2, \dots, n, \end{cases} \quad (104)$$

$$v_{j,2} = \begin{cases} \sup\{[A_j]_1\} & \text{ha } j = 1, \dots, n-1 \\ x_{\max} & \text{ha } j = n. \end{cases} \quad (105)$$

A halmazok tartóinak végpontjait a szomszédos halmazok magjai alapján számíthatjuk

$$\inf\{[A_j]_0\} = \begin{cases} x_{\min} & \text{ha } j = 1 \\ v_{j-1,1} & \text{ha } j = 2, \dots, n, \end{cases} \quad (106)$$

$$\sup\{[A_j]_0\} = \begin{cases} v_{j+1,1} & \text{ha } j = 1, \dots, n-1 \\ x_{\max} & \text{ha } j = n. \end{cases} \quad (107)$$

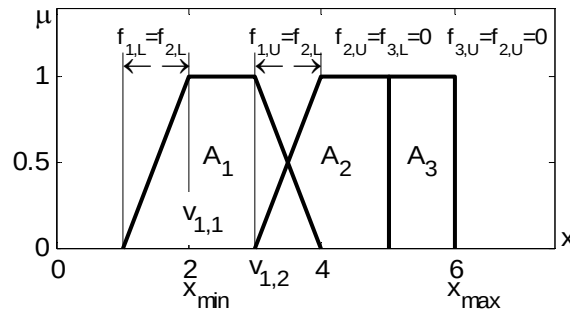
A nyelvi értékek eltolásának elvét alkalmazó szabály-interpolációs módszerekre épülő fuzzy rendszerek esetén a 4.1. szakaszban ismertetett megfontolások lapján megengedett, hogy a

partíció szélein elhelyezkedő fuzzy halmazok alakjának egy része túlnyúljon az alaphalmaz korlátokon. Ruspini partíciónál ezt az alsó és felső fuzzy jelleg felhasználásával oldjuk meg. Ezek értékét a (7) és (8) alapján itt az alábbi összefüggéssel számítjuk

$$f_{j,L} = \inf\{[A_j]_1\} - \inf\{[A_j]_0\}, \quad (108)$$

$$f_{j,U} = \sup\{[A_j]_0\} - \sup\{[A_j]_1\}, \quad (109)$$

ahol $f_{j,L}$ és $f_{j,U}$ a j -edik fuzzy halmaz alsó illetve felső fuzzy jellege.



95. ábra. Alsó és felső fuzzy jelleg

Az első halmaz bal oldali fuzzy jellegét azonosnak vesszük a második bal alsó fuzzy jellegével ($f_{1,L} = f_{2,L}$, ld. 95. ábra), míg az utolsó halmaz jobb oldali fuzzy jellegét azonosnak vesszük az utolsó előtti halmaz felső fuzzy jellegével ($f_{n,U} = f_{n-1,U}$, ld. 95. ábra). A partíció sajátosságából adódóan ezek az értékek egyben azonosak lesznek az érintett első vagy utolsó halmaz másik oldali fuzzy jellegével. Így a tartó végpontjaira a következő összefüggéseket kapjuk

$$\inf\{[A_j]_0\} = \begin{cases} x_{\min} - f_{1,L} & \text{ha } j = 1 \\ v_{j-1,1} & \text{ha } j = 2, \dots, n, \end{cases} \quad (110)$$

$$\sup\{[A_j]_0\} = \begin{cases} v_{j+1,1} & \text{ha } j = 1, \dots, n-1 \\ x_{\max} + f_{n,U} & \text{ha } j = n. \end{cases} \quad (111)$$

A paraméter-azonosítási folyamat során az új paraméter értékek meghatározásánál a következő megkötéseket vesszük figyelembe.

- Egy halmaz második paramétere nem lehet kisebb az első paraméternél

$$v_{j,2} \geq v_{j,1}, \quad \forall j = 1, \dots, n. \quad (112)$$

- Egy fuzzy halmaz magja még csak részben sem fedheti át egy másik halmaz magját, azonban két szomszédos halmaz magjai érintkezhetnek

$$v_{j,1} \geq v_{j-1,2}, \quad \forall j = 2, \dots, n, \quad (113)$$

$$v_{j,2} \leq v_{j+1,1}, \forall j = 1, \dots, n-1. \quad (114)$$

A Ruspini jelleget definiáló (103) tulajdonság megőrzése érdekében azt az alaphalmaz elemet, amelynél két szomszédos nyelvi érték magja érintkezik a bal oldali (alsó) fuzzy halmaz elemének tekintjük, tehát tagsági értéke a jobb oldali halmazban 0. A 95. ábrán bemutatott partíció esetében ez azt jelenti, hogy $\mu_{A_2}(5)=1$ és $\mu_{A_3}(5)=0$.

- Az első halmaz első ($v_{1,1}$) és az utolsó halmaz második ($v_{n,2}$) paramétere a hangolás során nem módosítható.

Megjegyzés: a Ruspini partíció megőrzését célzó paraméterezés nem teszi lehetővé ritka partíciók keletkezését és kezelését.

8.1.2.4. Referencia pont

Néhány gyakorlati feladatnál, ahol a partíciókon belül minden fuzzy halmaz azonos alakú és ez az alakzat szimmetrikus a referencia pontban elhelyezett függőleges tengelyre, előnyös lehet olyan hangolási algoritmus alkalmazása, ami csak a halmazok helyzetét módosítja, és alakjukat csak a megkötések kielégítése érdekében változtatja. Ilyenkor halmazonként elég egy, csak a helyzetet beazonosító paraméter, ami azonos a referencia ponttal. Emellett partíciónként a partícióra jellemző mag (W_C) és tartó (W_S) szélességeket is tárolnunk kell, ezek azonban konstans értékek, amelyek nem változnak a paraméter-azonosítási folyamat során. A fuzzy halmazt meghatározó két α -vágatot a referencia pont és a szélesség értékek ismeretében a következőképpen számítjuk.

$$\inf\{[A]_0\} = RP(A) - \frac{W_S}{2} \quad (115)$$

$$\sup\{[A]_0\} = RP(A) + \frac{W_S}{2} \quad (116)$$

$$\inf\{[A]_1\} = RP(A) - \frac{W_C}{2} \quad (117)$$

$$\sup\{[A]_1\} = RP(A) + \frac{W_C}{2} \quad (118)$$

A referencia pont alapú paraméterezésnél az alábbi megkötések érvényesek.

- A referencia pont az alaphalmaz adott dimenziójában érvényes alsó és felső korlátok közzé kell essen.

$$x_{\min} \leq RP(A) \leq x_{\max} \quad (119)$$

A halmaz-interpolációs módszerek tárgyalása során a 4.1. szakaszban ismertetett megfontolásokat követve itt is megengedtük, hogy a konzekvens partíció két fuzzy halmazának egy része az $[x_{\min}, x_{\max}]$ intervallumon kívülre essen.

- Olyan esetekben, amikor a fuzzy rendszer működése során nem alkalmazzuk a nyelvi értékek eltolásának elvét (ld. 4.1. szakasz), a korlátok szigorú értelmezése szükséges, ami az alábbi megkötésekkel fejezhető ki

$$RP(A) - \frac{W_s}{2} \geq x_{\min} \text{ és } RP(A) + \frac{W_s}{2} \leq x_{\max}, j = 1, \dots, n, \quad (120)$$

ahol n a nyelvi értékek száma a partícióban.

- Az egyes nyelvi értékek és a rájuk épülő szabályok jobb értelmezhetősége érdekében nem engedhetjük meg, hogy egy fuzzy halmaz tartója átfedje egy másik halmaz magját. A j -edik halmaz esetében ezt a következőképpen fejezhetjük ki

$$RP(A_j) - \frac{W_s}{2} \geq RP(A_{j-1}) + \frac{W_c}{2}, j = 2, \dots, n, \quad (121)$$

$$RP(A_j) + \frac{W_s}{2} \geq RP(A_{j+1}) - \frac{W_c}{2}, j = 1, \dots, n-1, \quad (122)$$

ahol n a nyelvi értékek száma a partícióban. Érvényes fuzzy halmazokat feltételezve a fenti megkötés természetesen azt is magában foglalja, hogy a halmaz magja nem fedheti át egy másik halmaz magját.

8.1.2.5. A paraméterezési mód kiválasztása

Az előző szakaszokban ismertetett paraméterezési módokat kísérletek során vizsgáltam különböző rendszergenerálási módszerek alkalmazása mellett (pl. [S25]). Az eredményeket a következőképpen foglalhatjuk össze.

A töréspont alapú paraméterezési mód egyszerű, könnyen értelmezhető, és lehetővé teszi, hogy egyszerre csak egy töréspont helyzetét módosítsuk az alakzatban. Ezáltal finomabb hangolás valósítható meg.

A relatív távolság alapú paraméterezés előnye, hogy a négyből három paraméter értéke csak pozitív vagy nulla lehet, ami megkönnyíti az ellenőrzést és az implementációt. A módszer hátránya, hogy az első három paraméter bármelyikének változtatása az alakzat több karakterisztikus pontjának a módosulását is eredményezi, ami esetleg megnehezítheti a paraméter-azonosítás során az egyes paraméterek optimális értékeinek megtalálását.

Számos kísérlet elvégzése alapján elmondható, hogy az első két paraméterezési mód alkalmazásával elért teljesítménymutatók minden esetben igen közel álltak egymáshoz. Körülbelül azonos számú esetben vezetett jobb eredményhez az első (töréspont alapú), mint a második (relatív távolság alapú) megközelítés.

A Ruspini partíció megőrzésére irányuló módszer kivétel nélkül minden esetben rosszabb teljesítménymutatót eredményezett előző két társánál. A megközelítés előnye, hogy a partíció leírására csak fele annyi paramétert használ, mint a másik két módszer, aminek következtében kisebb tárterület- és számításigényű implementációt készíthetünk. Hátrányának tekinthető, hogy nem teszi lehetővé ritka partíció létrehozását, és így a Ruspini tulajdonság (103) erőltetett megőrzése következtében a szükségesnél szélesebb nyelvi értékek keletkeznek. A probléma egy lehetséges, bár megvalósítását tekintve bonyolult megoldását jelentheti olyan ál-nyelvi értékek elhelyezése az egyes partíciókban, amelyek egyetlen szabályban sem szerepelnek. Ily módon ritka partíciót is szimulálhatunk. Ezen megoldás implementálhatóságát és eredményességét nem vizsgáltam.

A referencia pont alapú paraméterezés a legegyszerűbb úgy a hangolás implementációját, mint a megkötések érvényesítését tekintve. Előnyös tulajdonsága a nyelvi értékek könnyű elkülöníthetősége, ami biztosítja a szabályok ember számára is egyszerű értelmezhetőségét.

További előnye a ritka partíciók kezelésének képessége. Hátránya, hogy mivel halmazonként csak egy paraméter változtatható, a rendszer kevésbé hangolható.

A rendszergenerálás és paraméter-azonosítás céljára fejlesztett szoftver (RuleMaker) lehetővé teszi bármelyik paraméterezési módszer alkalmazását.

8.1.3. Paraméter-azonosítás és szabálybázis kiterjesztés

A paraméterek optimális értékeinek meghatározását egy heurisztikus „hegymászó” típusú algoritmus segítségével oldottam meg. A módszer fontos segédeszköze az ún. teljesítménymutató, ami lehetővé teszi a különböző paraméter értékek által definiált fuzzy modellek összehasonlítását és értékelését (bővebben ld. 8.1.5. szakasz). Az algoritmus célja az, hogy megtaláljuk azt a paramétersort, amelyik biztosítja az alkalmazott teljesítménymutató lehető legjobb értékét. Az egységesség érdekében a továbbiakban mindig „kisebb a jobb” típusú teljesítménymutatót használunk.

Az egyes partíciókban megjelenő fuzzy halmazok paramétereire vonatkozó és egymással szoros összefüggésben álló megkötések (ld. 8.1.2. szakasz) következtében azt a megoldást választottam, hogy az algoritmus minden iterációs menet során egyenként veszi sorra az összes partíció minden paraméterét. Az eljárás minden esetben két új értéket számít az aktuális paraméterhez úgy, hogy az eredeti értéket növeli, illetve csökkenti egy előre megadott lépésközzel (st_i), majd ezután alkalmazza a megkötéseket. A lépésköz kezdőértékének meghatározását és pillanatnyi értékének korlátait bővebben a 8.1.4. szakasz tárgyalja.

Az eljárás kiszámítja az így kapott két új értékre a fuzzy rendszer teljesítménymutatóját. Amennyiben az eredeti értékre vonatkozó mutató nem ismert, akkor arra is elvégzi a számításokat. A három lehetséges paraméter érték közül azt tartjuk meg, amelyik a legjobb teljesítménymutatót eredményezte. Az előzőekben vázolt technika egy hegymászó megközelítés, ahol a gradiens irányát a pillanatnyi állapot értéke és a szomszédos állapotok értékének viszonyával becsüljük.

Minden iterációs menet végén, amikor végighaladtunk egyszer a rendszer összes hangolni kívánt paraméterén, összehasonlítjuk az aktuális teljesítménymutatót (PI_k) az előző iterációs menet végén (PI_{k-1}) számított teljesítménymutatóval, illetve az első iteráció esetén a kiinduláskor (PI_0) mért teljesítménymutatóval

$$\Delta PI = PI_{k-1} - PI_k. \quad (123)$$

Amennyiben a teljesítménymutató értéke jobban javult egy felső határértéknél ($\Delta PI \geq \Delta PI_{\max}$, ld. 8.1.5.), akkor a lépésköz nagyságát meghatározó együttható értékét megduplázzuk a 8.1.4. szakaszban ismertetésre kerülő $C_{\min}$ és $C_{\max}$ korlátok figyelembevételével

$$C = \min\{C_{\max}, 2 \cdot C\}. \quad (124)$$

Amennyiben a teljesítménymutató javulása egy előre megadott alsó határérték ($\Delta PI \leq \Delta PI_{\min}$, ld. 8.1.5.) alá esik, akkor az együttható értékét felezzük a 8.1.4. szakaszban ismertetésre kerülő $C_{\min}$ és $C_{\max}$ korlátok figyelembevételével

$$C = \max\{C_{\min}, C/2\}. \quad (125)$$

Amennyiben az együtttható már korábban elérte alsó korlátját (C_{min}), akkor nem osztjuk 2-vel, hanem egy új szabályt generálunk. Ennek megvalósítását a 8.1.6. szakaszban tárgyaljuk. Az új szabály létrehozása után a C együttthatót újból kezdeti értékére állítjuk, majd folytatjuk az iterációt. A paraméter-azonosítási folyamat akkor áll le, ha elértük az előre megszabott maximális iterációs menetszámot vagy a teljesítménymutató elért egy előre meghatározott jósági értéket (PI_{tr}). Az algoritmus metanyelvű leírását a 96. ábra ismerteti.

```

i=1
PIi=Nyers/kiinduló rendszer kiértékelése
PImin=PIi
ISMÉTLÉS AMÍG i≤MaxIterNo ÉS PIi≥PItr
  Minden be- és kimeneti dimenzióban
    Lépésköz számítása
    Minden nyelvi értékre
      Minden paraméterre
        Mindkét változtatási irányra
          Új paraméter érték számítása
          Megkötések érvényesítése
          PIi=Nyers/kiinduló rendszer kiértékelése
          HA PIi < PImin AKKOR
            Paraméter érték tárolása
            PImin=PIi
          VégeHA
        VégeMindkét
      VégeMinden
    VégeMinden
  DPI=PIi-1-PIi
  HA DPI≥DPImax AKKOR
    Lépésköz együtttható növelése
  KÜLÖNBEN
    HA DPI≤DPImin AKKOR
      HA Lépésköz együtttható kicsi AKKOR
        Új szabály generálása
        Lépésköz együtttható kezdeti értékre állítása
      KÜLÖNBEN
        Lépésköz együtttható csökkentése
    VégeHA
  VégeHA
i=i+1
VégeISMÉTLÉS

```

96. ábra. Az RBE-DSS/SI módszerek paraméter-azonosítási algoritmus

A jelen szakaszban felvázolt eljárás a [H13] és [H72] irodalmakban bemutatott módszerek alap gondolatát viszi tovább. Azokhoz hasonlóan az optimális paraméter értékeket egymást követő iterációs lépések sorozatával keresi, egyszerre csak egy paramétert változtat, és minden paraméter esetén minden iterációban két új értéket számol. Az említett módszerektől eltérően az általam kidolgozott eljárás a konzekvens halmazok paramétereit is hangolja, a lépésköz meghatározására teljesen más megközelítést alkalmaz, többfajta teljesítménymutató

alkalmazását és több paraméterezési lehetőség közüli választást tesz lehetővé, valamint új szabályok generálásával egészíti ki a szabálybázist.

8.1.4. A lépésköz nagysága

A lépésköz nagysága nem állandó a folyamat során. Értékét az adott dimenzióban érvényes alaphalmazkorlátok által definiált intervallum szélességének (DR_i) függvényében az alábbiakban ismertetésre kerülő adaptív módszerrel határozzuk meg. Az st_i lépésköz az i -edik dimenzióban

$$st_i = C \cdot DR_i. \quad (126)$$

A C együttható változik a folyamat során. Pillanatnyi értéke a $[C_{\min}, C_{\max}]$ intervallumba kell eszen, ahol $C_{\max} = 1$ és

$$C_{\min} = \frac{10^{-dn}}{\min_{i \in [1, N+1]} (DR_i)}, \quad (127)$$

ahol dn a paraméterek leírásában (értékük tárolása során) alkalmazható maximális tizedesjegyszám. Ez egy gyakorlati implementációs korlát, és azt fejezi ki, hogy nincs értelme a paramétereket olyan kis mértékben módosítani, hogy az eredményül kapott rendszer a kerekítések következtében azonos legyen a paraméter eredeti értékét tartalmazó rendszerrel.

A C együttható kezdőértéke egy empirikus érték, ami befolyásolja a hangolási folyamat időigényét, és hatással van annak eredményességére. A gyakorlati számítások során $C = 0,2$ -es kezdőértéket alkalmaztam.

8.1.5. Teljesítménymutató

A paraméter-azonosítási folyamat kezdetén és annak során minden lehetséges új paraméter érték kiszámítása után értékeljük a fuzzy modellt. Míg a kezdeti értékelés csak egy viszonyítási alap megállapítását szolgálja, addig a későbbiekben már a paraméter változtatások hatására vagyunk kíváncsiak. A fuzzy rendszer értékelését egy olyan mintaadathalmaz segítségével végezzük, amelynél ismerjük a bemenetekhez elvárt kimeneti értékeket. Minden bemeneti adatsorra kiszámítjuk a rendszer által előállított kimenetet, és megvizsgáljuk annak eltérését az elvárt kimenettől.

Általában nem az egyedi eltérések érdekesek számunkra, hanem egy összefoglaló képet szeretnénk kapni, azaz egyetlen számra van szükségünk, ami lehetővé teszi egy paraméter két lehetséges értéke által fémjelzett két fuzzy modell összevetését, és a számunkra előnyösebb kiválasztását. Az egyedi eltérésekből tehát valamilyen aggregációs művelet segítségével egy teljesítménymutatót képezünk. Az alábbiakban néhány lehetséges teljesítménymutatót ismertetünk.

MSE. A szakirodalom (pl. [H66] és [H72]) egyik kedvelt teljesítménymutatója az átlagos négyzetes eltérés (MSE – Mean Square Error)

$$PI_{MSE} = \frac{\sum_{j=1}^M (y_j - \hat{y}_j)^2}{M}, \quad (128)$$

ahol M a mintaadathalmaz adatsorainak száma, y_j a j -edik adatsor elvárt kimenete és $\hat{y}_j$ a rendszer által előállított kimenet. Előnye viszonylag egyszerű számíthatóságában és

közismertségében rejlik. Értékének minősítését azonban megnehezíti, hogy négyzetes jellege miatt nem vehető össze könnyen a kimenet nagyságával és értékkészletének tartományával.

RMSE. Az átlagos négyzetes hiba négyzetgyöke vagy más néven empirikus szórás [H77] (RMSE – Root Mean Square Error) az MSE-vel szemben a rendszer kimenetével azonos mértékegységű mennyiség, ami megkönnyíti értelmezését. Képlete a következő

$$PI_{RMSE} = \sqrt{\frac{\sum_{j=1}^M (y_j - \hat{y}_j)^2}{M}}. \quad (129)$$

Az RMSE a gyökvonás bevezetése következtében nagyobb számításigényű az előzőekben ismertetett társánál.

RMSEP. A számításigény növelése árán tovább javíthatjuk a teljesítménymutató értelmezhetőségét, valamint könnyíthetjük a rendszer minősítését. Az átlagos négyzetes eltérés négyzetgyökének relatív értéke (RMSEP – Root Mean Square Error in Percentage) a kimeneti alaphalmaz terjedelméhez viszonyítva, annak százalékában jellemzi a rendszert. Képlete a következő

$$PI_{RMSEP} = \sqrt{\frac{\sum_{j=1}^M (y_j - \hat{y}_j)^2}{M}} \cdot DR_{N+1}^{-1} \cdot 100, \quad (130)$$

ahol DR_{N+1} a (85)-hez hasonlóan a kimeneti dimenzió terjedelmét jelöli, és N az antecedens dimenziók száma.

Korrelációs együttható. Az előzőekben ismertetett egymáshoz viszonylag közel álló teljesítménymutatóktól teljesen eltérő értékszámot javasolt Wong és Gedeon a [H79] irodalomban

$$PI_R = \frac{\sum_{j=1}^M (y_j - \bar{y}) \cdot (\hat{y}_j - \bar{\hat{y}})}{\sqrt{\sum_{j=1}^M (y_j - \bar{y})^2 \cdot \sum_{j=1}^M (\hat{y}_j - \bar{\hat{y}})^2}}, \quad (131)$$

ahol $\bar{y}$ a mintaadathalmaz elvárt kimeneti értékének átlaga és $\bar{\hat{y}}$ a rendszer által előállított kimenetek átlaga.

A (131) képlettel számított korrelációs együtthatót, ami a rendszer kimenete és az előírt kimenet közötti kapcsolatot jellemzi viszonylag, ritkán alkalmazzák fuzzy rendszerek értékelésénél. Ennek oka az, hogy a (128), (129) és (130) képleteknél jelentősen nagyobb a számításigénye, továbbá nemcsak olyankor eredményezi az optimális 1-es értéket, ha az y_i és $\hat{y}_i$ értékek azonosak (ami a rendszergenerálás célja), hanem olyankor is, ha lineáris függvénykapcsolat áll fenn közöttük. Hátrányai ellenére az automatikus rendszergenerálást és az ellenőrzést végző RuleMaker és TestIt szoftverek lehetővé teszik egy, az egységesítés érdekében módosított változatának használatát. Mivel a (128), (129) és (130) teljesítménymutatók mind „kisebb a jobb” típusúak, így célszerűnek tűnt a „nagyobb a jobb” típusú korrelációs együtthatóból is egy olyan teljesítménymutatót képezni, amely illeszkedik a többiekhez

$$PI_{RR} = 1 - \frac{\sum_{j=1}^M (y_j - \bar{y}) \cdot (\hat{y}_j - \bar{\hat{y}})}{\sum_{j=1}^M (y_j - \bar{y})^2 \cdot \sum_{j=1}^M (\hat{y}_j - \bar{\hat{y}})^2} . \quad (132)$$

A módosítás következtében a PI_{RR} értékei a $[0, 2]$ intervallumba esnek, ahol a fuzzy rendszerek értékelésénél a 0-s értéket tekintjük optimálisnak.

AD. Az abszolút eltérések átlaga viszonylag ritkán alkalmazott teljesítménymutató. Előnyös tulajdonsága a kis számításigény, míg hátránya, hogy a kapott eredmény nem annyira kézzelfogható, nem ad olyan jó képet, mint az empirikus szórás. Képlete a következő

$$PI_{AD} = \frac{\sum_{j=1}^M |y_j - \hat{y}_j|}{M} . \quad (133)$$

ADP. Az abszolút eltérések esetében is alkalmazhatjuk a kimeneti dimenzió terjedelméhez viszonyított százalékos értékelést

$$PI_{ADP} = \frac{\sum_{j=1}^M |y_j - \hat{y}_j|}{M \cdot DR_i} \cdot 100 . \quad (134)$$

A teljesítménymutató változásának minősítése során használt alsó és felső határértékek empirikus mennyiségek. Számértékük szorosan kapcsolódik a kiválasztott teljesítménymutató típushoz. Például az PI_{RMSE} esetében a kísérletek során az alábbi értékeket alkalmaztam:

$$\Delta PI_{\min} = 0,01 \cdot DR_{N+1} , \quad (135)$$

$$\Delta PI_{\max} = 0,10 \cdot DR_{N+1} . \quad (136)$$

8.1.6. Új szabály generálása

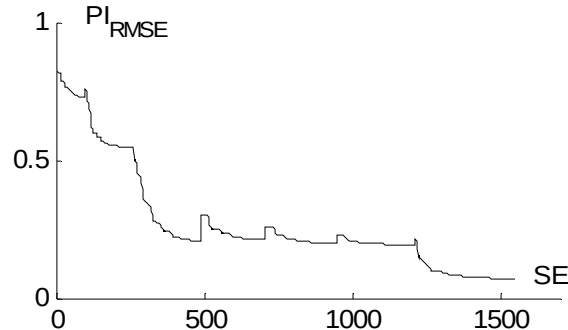
Az RBE megközelítésben az új szabályt úgy hozzuk létre, hogy az illeszkedjen a konzekvens alaphalmaz azon pontjához, ahol a legnagyobb az eltérés a mintaadatokat által előírt és a fuzzy modell által számított kimenet között. Ennek érdekében az érintett mintaadatsor kikeresése után új nyelvi értékeket hozunk létre úgy, hogy ezek referencia pontjai essenek egybe az adatsor azonos dimenzióbeli értékeivel. Az új fuzzy halmazok alakját a két javasolt módszer eltérően határozza meg.

Az **RBE-DSS** (Rule Base Extension based on Default Set Shapes - Szabálybázis kiterjesztés alapértelmezett halmazalak használatával) az első két szabály létrehozása során már alkalmazott, partícióra jellemző alapértelmezett halmazalakokat használja az új nyelvi értékek létrehozása során.

A partíciónkénti halmazszám túlzott növekedésének elkerülése érdekében két metasabályt alkalmazunk minden antecedens és konzekvens dimenzióban az esetleges nyelvi érték összevonások szükségességének elbírálására. Ezek azonosak a 8.1.1. szakaszban ismertetett két metasabállyal. Az RBE-DSS esetében az újonnan létrehozott fuzzy halmaz és az eredeti fuzzy halmazok viszonylatában vizsgáljuk az összevonhatóságot. A halmaz-úniókat itt is a

CNF unió segítségével hajtjuk végre. Szükség esetén ezt követően módosítjuk az újonnan létrehozott halmazok alakját a 8.1.2. szakaszban leírt megkötések kielégítése érdekében.

Az RBE-DSS előnye egyszerűségében és gyorsaságában rejlik, gyenge pontja viszont az, hogy a segítségével létrehozott új szabály beillesztése a szabálybázisba a teljesítménymutató ideiglenes romlásához vezethet.



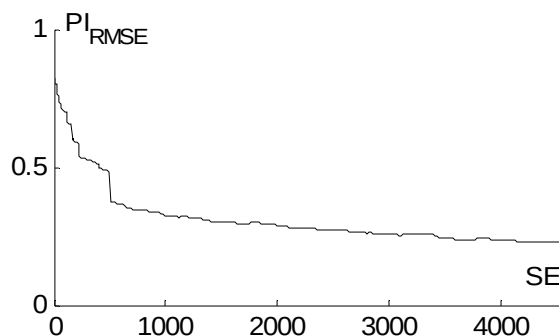
97. ábra. Teljesítménymutató alakulása a modellgenerálási folyamat során RBE-DSS esetén

A 97. ábrán nyomon követhetjük egy fuzzy modell generálása során a teljesítménymutatóként alkalmazott RMSE alakulását LESFRI szabály-interpolációs módszer alkalmazása esetén az RBE-DSS használatával. A vízszintes tengelyen a rendszerkiértékelések (teljesítménymutató számítások) száma szerepel. A görbén jól kivehetőek a lokális maximumok, amelyek az új szabályok létrehozásához kapcsolódnak.

Az **RBE-SI** (Rule Base Extension based on Set Interpolation - Szabálybázis kiterjesztés halmaz-interpoláció használatával) az új szabályok létrehozásakor az előző módszernél tapasztalt ideiglenes teljesítménymutató romlást azáltal kívánja elkerülni, hogy a szabályban megjelenő új nyelvi értékeket halmaz-interpolációval állítja elő. A további számítások megkönnyítése érdekében célszerű az alkalmazott halmaz-interpolációs eljárást a szabály-interpolációs módszerhez igazítani, azaz pl. FRIPOC esetén a FEAT-p, LESFRI esetén a FEAT-LS, míg VEIN esetén a VESI alkalmazása ajánlott.

Az előző módszerhez hasonlóan itt is alkalmazzuk az összes be- és kimeneti dimenzióban a korábban megismert két metasabályt annak érdekében, hogy megakadályozzuk a partíciónkénti halmazszám túlzott növekedését. Itt is csak az újonnan létrehozott fuzzy halmazok és az eredeti halmazok viszonylatában vizsgáljuk az összevonhatóságot. A halmaz-unió céljára a CNF uniót alkalmazzuk. Ezt követően az újonnan létrehozott halmazok alakját szükség esetén módosítjuk a 8.1.2. szakaszban leírt megkötések kielégítése érdekében.

A halmaz-interpoláción alapuló RBE-SI előnye, hogy alkalmazásának eredményeképpen az új szabály bevezetése várhatóan legfeljebb kis mértékben rontja a teljesítménymutató pillanatnyi értékét, hátránya viszont az, hogy megnöveli a végső rendszer generálásához szükséges számítások mennyiségét és a folyamat időigényét.



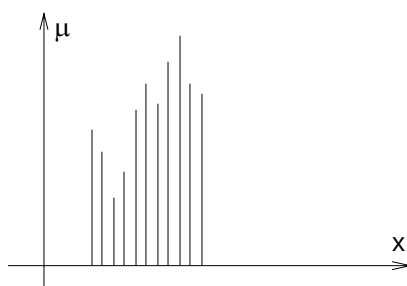
98. ábra. Teljesítménymutató alakulása a rendszergenerálási folyamat során RBE-SI esetén

A 98. ábrán nyomon követhetjük egy fuzzy modell generálása során a teljesítménymutatóként alkalmazott átlagos négyzetes eltérés négyzetgyökének (RMSE) alakulását LESFRI szabály-interpolációs módszer alkalmazása esetén az RBE-SI használatával. A nyers rendszer és a modellezett jelenség azonos azzal, amit a 97. ábrán nyomon követett rendszergenerálásnál használtam. A vízszintes tengelyen a rendszerkiértékelések (teljesítménymutató számítások) száma szerepel. Jól látható, hogy a görbe nem tartalmaz olyan erős kiemelkedést, mint amilyeneket az RBE-DSS használata esetén a 97. ábrán megfigyelhettünk. Itt az új szabályok bevezetése csak kismértékű „dombocskákat” eredményezett.

8.2. Fuzzy klaszterezésen alapuló automatikus rendszergenerálás

A fuzzy klaszterezésen alapuló ritka szabálybázist generáló módszereknek két irányzatot különböztethetünk meg. Míg az első a teljes adathalmazon hajt végre egy többdimenziós klaszterezést, addig a második a kimeneti adatok klaszterezéséből indul ki.

Az első megközelítés a klaszterezést követően minden klaszterből egy szabályt képez úgy, hogy a többdimenziós klaszterek vetületeit képzik az egyes antecedens és konzekvens dimenziókban. Ezt a megközelítést javasolja például Klawonn és Kruse [H29]. A megoldás szemléletes és logikus, hátránya azonban, hogy az adatpontok segítségével kapott klaszter vetületek alapján igen nehézkes a nyelvi értékek becslése [H29] (egyes esetekben nehezen értelmezhetők konvex fuzzy halmazként ld. 99. ábra).



99. ábra. Klaszter vetület

Az előző megoldástól eltérően Sugeno és Yakusawa [H66] a kimeneti adatok klaszterezésével és konzekvens nyelvi értékek előállításával kezdi a szabálybázis előállítását. A konzekvens klaszterek alapján többdimenziós klasztereket hoznak létre az antecedens térben, amelyeknek vetületei alapján képzik az antecedens nyelvi értékeket. Az eljárás előnye, hogy alacsony komplexitású fuzzy rendszerhez vezet, hátránya viszont, hogy számos részletkérdését a szerzők nem tisztázták, illetve csak felületesen kezelték azt az esetet, amikor egy konzekvens klaszter/fuzzy halmaz több szabályban is szerepel (különböző antecedensek azonos

konzekvenshez vezetnek). Tikk, Gedeon, Kóczy és Bíró [H72] a hiányzó részleteket kidolgozták, majd Wong, Kóczy, Gedeon, Chong és Tikk [H81] továbbfejlesztették azt az antecedens klaszterek összevonhatóságának vagy elkülönülésének alapos vizsgálata tekintetében. A továbbfejlesztett módszer gyenge pontja jelentősen megnövekedett számítási bonyolultsága és az, hogy a Klawonn és Kruse által jegyzett módszerhez hasonlóan itt is nehézkes a vetületek alapján történő halmazbecslés.

Később Chong, Gedeon és Tikk [H15] PRE néven egy olyan megoldást dolgozott ki, amely az előzőekhez hasonlóan a kimeneti adatok klaszterezésével indít, majd klaszterenként különválasztva az adatsorokat, az egyes antecedens dimenziókban külön-külön hajt végre klaszterezést, és ezen klaszterek kombinálásával közelíti a többdimenziós antecedens klasztereket. A módszer egyszerűbb és áttekinthetőbb elődeinél.

Az alábbiakban ismertetésre kerülő eljárást [S25] (ACP – Automatic fuzzy system generation based on fuzzy Clustering and Projection) a [H13], [H15], [H66], [H72] és [H81] irodalmakban közzétett eljárások egyes elemeinek és gondolatainak felhasználásával alakítottam ki alapvetően a PRE módszerre [H15] támaszkodva. Implementációját beépítettem a RuleMaker Matlab ToolBoxba.

Az egyszerű szabályos következtetési módszerek ismertetéséhez hasonlóan itt is csak egykimenetű (SISO és MISO) rendszerekkel foglalkozom. A szabálybázist generáló módszerek tárgyalása során csak kompakt fuzzy halmazokat alkalmazok. Elterjedt volta miatt az előzőekhez hasonlóan itt is trapéz alakú nyelvi értékekkel dolgozom.

Az ACP-vel történő fuzzy rendszerelőállítás során alapvetően két lépést különböztetünk meg. Az első lépésben fuzzy klaszterezés és a vetítési elv alkalmazásával egy nyers fuzzy rendszert állítunk elő, míg a második lépésben egy iteratív hangolási folyamat keretében azonosítjuk a nyelvi értékek paramétereit. Az első lépés algoritmusának fontosabb elemeit a 100. ábra ismerteti. A második lépést a 8.2.5. szakasz tárgyalja.

Optimális klaszterszám meghatározása a kimeneti adatokra
Kimeneti adatok 1D FCM klaszterezése
Konzekvens nyelvi értékek képzése
Vetítés az antecedens dimenziókba

100. ábra. Az ACP első lépésének algoritmus

8.2.1. Egydimenziós FCM fuzzy klaszterezés

A vetítési elvre támaszkodó (ld. 8.2.4. szakasz) automatikus szabályelőállítás során az FCM (FCM-Fuzzy C-Means clustering) klaszterezést használom a be- és kimeneti nyelvi értékek létrehozására. A feladathoz igazodva a jelen esetben csak az egydimenziós klaszterezés ($N=1$) szempontjából vizsgálom az FCM eljárást. Az adatpontok hasonlóságát euklideszi távolságukkal értékelem.

A Dunn [H20] által bevezetett, majd Bezdek [H6] által továbbfejlesztett fuzzy c-közép klaszterezés (FCM) az $X = \{x_1, \dots, x_M\} \subset R^N$ adathalmaz pontjait előre megadott számú (c) klaszterbe sorolja a pontok $V = \{v_1, \dots, v_c\} \subset R^N$ klaszterközéppontoktól mért távolságai alapján. Az algoritmus felügyelet nélküli, a klaszterközéppontokat egy célfüggvény minimumát keresve iterációk sorozatával határozza meg az eljárás.

A módszer fuzzy jellege abból adódik, hogy az éles k-közép módszertől eltérően lehetővé teszi az adatpontok számára, hogy egynél több klaszterhez tartozzanak. A klaszterhez tartozás fokát tagsági értékek segítségével fejezzük ki.

Az eljárás előnyös tulajdonsága viszonylagos egyszerűsége és könnyű implementálhatósága, azonban a szabálygenerálás során történő alkalmazását az alábbi gyakorlati problémák nehezítik.

- Nem képes helyesen kezelni azt az esetet, amikor egy vagy több adatpont azonos értékű.
- Nem képes helyesen kezelni azt az esetet, amikor egy vagy több adatpont azonos értékű egy klaszterközponttal.
- A klaszterszám kettőnél nagyobb kell legyen. Ritka szabálybázis esetén azonban elég gyakran előfordul, hogy egy kimeneti nyelvi értékhez csak egy bemeneti nyelvi érték tartozik valamelyik antecedens dimenzióban.

A fenti problémák megoldása érdekében módosítottam a klaszterközpontok és a tagsági mátrix számítását. A klaszterezési algoritmus így a 101. ábrán ismertetett lépésekből áll.

```

U=tagsági mátrix inicializálása
Üres klaszterek eltávolítása
i=1
ΔO=ΔOmin
ISMÉTLÉS AMÍG i≤MaxIterNo ÉS ΔO≥ΔOmin
    Klaszterközpontok számítása
    Egybeeső klaszterközpontok eltávolítása
    U=tagsági mátrix számítása
    O=Célfüggvény számítása
    HA i>1 AKKOR
        ΔO=Oi-Oi-1
    VégeHA
    i=i+1
VégeISMÉTLÉS

```

101. ábra. Módosított FCM algoritmus

Tekintsük át az egyes lépések során szükséges számításokat.

Tagsági mátrix inicializálása

A tagsági mátrixot kétféleképpen inicializálhatjuk. Az első megközelítés a számítások ismételhetőségére helyezi a hangsúlyt egy olyan $U = [u_{jk}]_{c \times M}$ mátrixot létrehozva, aminek minden eleme nulla

$$U = 0_{c,M}, \quad (137)$$

ahol M az adatpontok száma, és c a klaszterszám. A mátrix minden sora egy klasztert jelképez, és minden oszlopa egy adatpontnak felel meg. Ezt követően minden adatponthoz egy klasztert rendelünk 1-es tagsági értékkel az alábbiak szerint

$$u_{k \bmod c+1,k} = 1, \quad k = 1, \dots, M, \quad (138)$$

ahol k az adatpont sorszáma, u_{jk} a k -edik pont tagsági értéke az j -edik klaszterben.

A második megközelítés véletlenszámokat használ a kezdeti tagsági mátrix előállítására. Itt először az egységintervallumból választott véletlenszámokkal feltöltünk egy

$U = [u_{jk}]_{c \times M}$ mátrixot, majd oszloponként összegezzük a mátrix adatait, és minden

mátrixelemet elosztunk az oszlopában korábban számolt összeggel, azaz egy oszloponkénti normalizálást hajtunk végre. Ez azt az elvet követi, ami szerint az alaphalmaz egy pontjának különböző klaszterekben mért tagsági értékeinek összege 1 kell legyen

$$\sum_{j=1}^c u_{jk} = 1, k = 1, \dots, M. \quad (139)$$

Üres klaszterek eltávolítása

Üresnek tekintjük azokat a klasztereket, amelyekhez nullánál nagyobb tagsági értékkel egyetlen adatpont sem tartozik. Olyankor keletkezhetnek például, amikor az eljárás paramétereként megadott klaszterszám nagyobb az adatpontok számánál. Az üres klaszterek eltávolítása a tagsági mátrix megfelelő sorainak törlésével és a c klaszterszám csökkentésével jár.

Klaszterközéppontok számítása

A klaszterközéppontokat (v_j) az alábbiak szerint számítjuk [H6]

$$v_j = \frac{\sum_{k=1}^M u_{jk}^m \cdot x_k}{\sum_{k=1}^M u_{jk}^m}, m > 1, \quad (140)$$

ahol m (fuzzy kitevő) az eljárás paramétere és értéke általában 2.

Egybeeső klaszterközéppontok eltávolítása

A gyakorlatban időnként előfordul, hogy két vagy több klaszter középpontja azonos, vagy távolságuk kisebb egy előre definiált határértéknél. Egybeeső klaszterközéppontokat kapunk például akkor, amikor az adatpontok azonosak és egynél nagyobb az előírt klaszterszám. Ilyenkor az azonos pozícióban levő második, harmadik, stb. klasztert egybeolvasztjuk az eredetivel (elsővel). Ekkor a tagsági mátrixból eltávolítjuk a felesleges sorokat és a c értékét is csökkentjük.

Új tagsági értékek számítása

A tagsági értékek számítása során a következő három szabály szerint járunk el.

1. Ha csak egy klaszterünk van, akkor minden adatpont 1-es tagsági értékkel eleme a klaszternek.
2. Ha egy pont egybeesik a klaszter középpontjával, tagsági értéke az adott klaszterben 1 lesz. Ez a pont nem lesz tagja más klasztereknek, azaz tagsága a többi klaszterben 0.
3. Minden más esetben a tagsági értéket az eredeti képlettel [H6] számítjuk.

A szövegesen megfogalmazott fenti három szabályt az alábbi képlettel írhatjuk le [H6]

$$u_{jk} = \begin{cases} 1 & \text{ha } c = 1 \text{ vagy } x_k = v_j \\ 0 & \text{ha } x_k = v_l \text{ és } l \neq j \\ \left[\sum_{l=1}^c \left(\frac{|x_k - v_j|}{|x_k - v_l|} \right)^{\frac{2}{m-1}} \right]^{-1} & \text{egyébként.} \end{cases} \quad (141)$$

A célfüggvény számítása

A célfüggvény minősíti a klaszterezés mindenkor állapotát. Az iterációk során értékének csökkentésére törekszünk. Számítására az alábbi képletet alkalmazzuk [H6]

$$O = \sum_{k=1}^M \sum_{j=1}^c u_{jk}^m \cdot (x_k - v_j)^2. \quad (142)$$

8.2.2. Az optimális klaszterszám meghatározása

Az FCM klaszterezés a priori információként megköveteli a klaszterek számának (c) ismeretét. Különböző c paraméter értékek eltérő végső partíciókhoz vezetnek. A klaszterszám pontos ismeretének hiányában az ún. klaszter érvényességi (validitási) mutatók nyújthatnak segítséget az optimális c érték meghatározásában.

A szakirodalomban számos ilyen mutatóval találkozhatunk. Ezek három csoportba sorolhatók aszerint, hogy milyen adatokat vesznek figyelembe [H76].

1. Csak a tagsági értékkel dolgozó mutatók: pl. Bezdek [H6] által javasolt partíció együtttható (PC) és partíció entrópia (PE), Windham arány kitevője [H78], valamint Kim és szerzőtársai által javasolt érvényességi mutató [H27].
2. A tagsági értéket és az adathalmazt figyelembe vevő mutatók: pl. Fukuyama és Sugeno [H21] FS mutatója, Xie és Beni [H83] mutatója, Davies és Bouldin [H17] által javasolt mutató, valamint a Yang és Wu [H84] partíció szeparációs indexe (PS).
3. Egyéb megoldások: pl. Chong, Gedeon és Kóczy [H14] hibrid kétlépéses mutatója valamint a Rhee és Oh által kidolgozott relatív és normalizált indexek [H60].

A klaszter érvényességi mutatók nagy száma ellenére a Yang és Wu [H84], a Chong [H13], valamint a Wang és Zheng [H76] által végzett vizsgálatok rámutattak, hogy egyik módszer sem képes arra, hogy minden adathalmaz esetén hibamentesen határozza meg az optimális klaszterszámot.

Mindezek figyelembevételével a Chong [H13] által jelzett viszonylag jó eredmények és szemléletessége miatt Chong, Gedeon és Kóczy [H14] hibrid kétlépéses mutatóját választottam egy módosítással kiegészítve, ami lehetővé teszi, hogy a felhasználó felső korlátot szabjon a klaszterszámra az egyes adatsoroknál tapasztalt komplexitásrobbanás elkerülése érdekében.

A Chong, Gedeon és Kóczy [H14] hibrid kétlépéses megoldásának első lépésében az FS index [H21] (143) segítségével megbecsülik az optimális klaszterszámot, majd egy összevonási mutató számításával megvizsgálják az egymáshoz közeli klaszterek összevonhatóságát.

Az első lépésben az FS mutatót a következő képlettel számítjuk

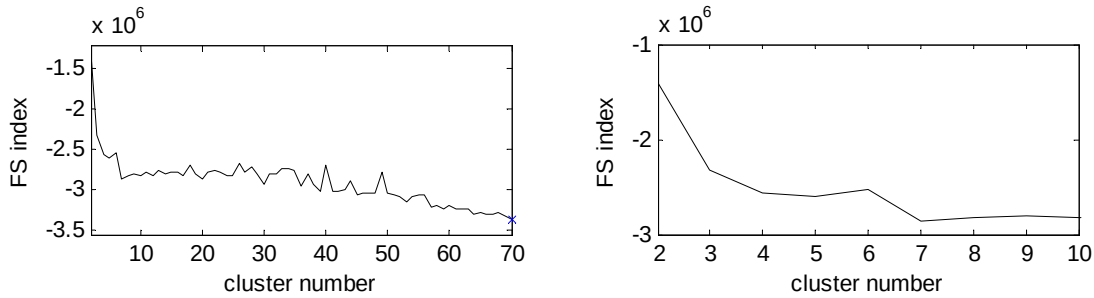
$$FS = \sum_{j=1}^c \sum_{k=1}^M (u_{jk})^m \cdot ((x_k - v_j)^2 - (v_j - \bar{v})^2), 2 \leq c \leq M \quad (143)$$

ahol M a mintaadatsorok száma, c a klaszterek száma, m a fuzzy kitevő, v_i az i -edik klaszter középpontja, u_{jk} a k -adik pont tagsága a j -edik klaszterben és

$$\bar{v} = \frac{\sum_{j=1}^c v_j}{c}. \quad (144)$$

Az FS mutató számítása során minden lehetséges klaszterszámmal elvégezzük a klaszterezést, és azt a klaszterszámot tekintjük optimálisnak, amely mellett a mutató értéke minimális.

Kísérletek során több különböző adathalmaznál is tapasztaltam, hogy a mutató több lokális minimum után csak egy igen nagy klaszterszám mellett éri el globális minimumát. Például a 102. ábra bal oldala a 9.2. fejezetben ismertetésre kerülő anaerob kúpos szuszpendált ágy reaktor működésének fuzzy modellezése során az ötödik kimeneti dimenzióban tapasztalt FS index alakulását mutatja a klaszterszám függvényében.



102. ábra. Az FS mutató alakulása a klaszterszám növekedésével klaszterszám korlátozás nélkül (bal oldal) és korlátozással (jobb oldal)

A magas klaszterszám (70 darab) nagy nyelvi érték- és szabálysámot eredményezne, ami a rendszer számításigényességének robbanásszerű növekedéséhez vezetne. Ennek megakadályozása érdekében két eszközzel élünk. Egyrészt egy felső korlátot adunk a klaszterszámmal, miáltal a folyamat számításigénye is jelentősen csökken. Ezt a korlátot a felhasználó adja meg, értékét a tapasztalatok szerint célszerű 7 és 15 között választani. A 102. ábra jobb oldalán láthatjuk, hogy egy 10-es maximum érték mellett az FS index 7 klaszternél érte el minimumát.

A hibrid módszer második lépéseként páronként megvizsgáljuk a klaszterközéppontok összevonhatóságát, alkalmazva a Chong, Gedeon és Kóczy [H14] által javasolt klaszter összevonási mutatót (145). Feltételezzük, hogy ha összevonnánk az i -edik és a k -adik klasztert, akkor az új klaszter középpontja a v_m (146) pontban lenne. Mindhárom középpontra (v_i, v_k, v_m) kiszámítjuk a P mutató [H14] értékét. Ha $P(v_m) < P(v_i)$ és $P(v_m) < P(v_k)$, akkor nem vonjuk össze a két klasztert, minden egyéb esetben végrehajtjuk az összevonást. Az alkalmazott képletek a következők

$$P(v_m) = \sum_{j=1}^M e^{-\left(\frac{v_m - x_j}{r_i}\right)^2}, \quad (145)$$

$$v_m = \frac{v_i + v_k}{2}, i \neq j \text{ és } i, j = 1, \dots, n, \quad (146)$$

$$ri = \frac{|v_i - v_k|}{2}, \quad (147)$$

ahol x_j a j -edik adat az aktuális dimenzióban, v_i és v_k a két összevonni kívánt klaszter középpontja, v_m az összevonás után keletkező klaszter középpontja, M az adatpontok száma és ri a hatósugár.

Egyes adathalmazoknál (pl. a 102. ábrán illusztrált esetben) a klaszter összevonási mutató nem képes hatékonyan ellátni feladatát, azaz gyakorlatilag nem eredményez klaszterszám csökkenést. Így a rendszer megvalósíthatósága szempontjából indokolt az előzőekben javasolt felső klaszterszám korlát bevezetése.

8.2.3. Fuzzy halmazok és partíció klaszterekből

Fuzzy klaszterek alapján többféle képpen is képezhetünk fuzzy halmazokat. Az egyes módszerek kötődnek az alkalmazott halmazalak típushoz. Az előző fejezetekhez hasonlóan a továbbiakban is trapéz alaktípussal dolgozunk. Ehhez kapcsolódóan kétféle megközelítést illetve módszert tanulmányoztam.

Az első a mintaadatok és a klaszter tagsági értékek alapján dolgozik. Ezt a megközelítést alkalmazza például a Tikk, Gedeon, Kóczy és Bíró [H72] által javasolt eljárás, ami két lépésben határozza meg a nyelvi értékeket. Az első lépésben egy konvex burkot képez az adatok és tagsági értékeik által meghatározott pontokra, majd ebből kiindulva becsli a trapéz paramétereit. Az eljárásnak három változata van, amelyek a tartó szélességében különböznek egymástól.

A második a klaszterközéppontokkal dolgozik. Ezt a megközelítést alkalmazza például a Chong [H13] által javasolt eljárás, ami egyszerűbb módon, egyetlen α -vágat alapján állít elő Ruspini [H61] partíciót.

Az egyszerűsége és a Ruspini partíció által biztosított későbbi könnyű értelmezhetőség miatt a második megoldást választottam. Az eljárás menete a következő. Először egy meghatározott szinten elkészítjük a középpontok által definiált klaszterek α -vágatait. Ezt követően az első és az utolsó nyelvi érték kivételével minden fuzzy halmaz magját ezen a vágatokkal képezzük. A vágási szint az eljárás paramétere, és a $[0,5, 1]$ intervallumba kell essen. Ajánlott értéke $\alpha = 0,85$ [H15].

Az első (bal szélső) nyelvi érték magjának jobb szélét és az utolsó (jobb szélső) nyelvi érték magjának bal szélét szintén a fentiek szerint számítjuk. Ezen fuzzy halmazok magjainak külső végpontjai a Ruspini partíciónál megszokott módon kötöttek az alaphalmaz alsó és felső korlátja, valamint az 1-es tagsági érték által

$$\inf\{[A_j]_1\} = \begin{cases} x_{\min} & \text{ha } j = 1 \\ \inf\{[C_j]_\alpha\} & \text{ha } j = 2, \dots, c, \end{cases} \quad (148)$$

$$\sup\{[A_j]_1\} = \begin{cases} x_{\max} & \text{ha } j = c \\ \sup\{[C_j]_\alpha\} & \text{ha } j = 1, \dots, c-1, \end{cases} \quad (149)$$

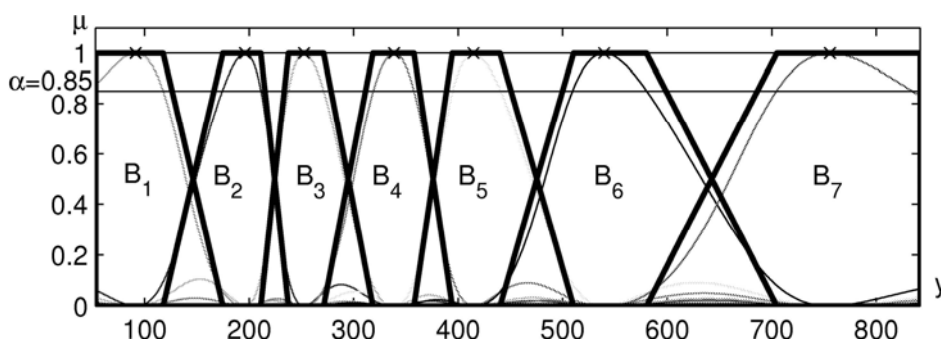
ahol c a klaszterek/fuzzy halmazok száma, C_j a j -edik klaszter A_j a j -edik fuzzy halmaz, $x_{\min}$ és $x_{\max}$ az alaphalmaz alsó és felső korlátja, α a becsléshez használt vágatszint.

A halmazok tartóját a szomszédos halmazok magjának figyelembevételével a Ruspini szabályosság (ld. (103)) megőrzésével határozzuk meg

$$\inf\{[A_j]_0\} = \begin{cases} x_{\min} & \text{ha } j=1 \\ \sup\{[A_{j-1}]_1\} & \text{ha } j=2, \dots, c, \end{cases} \quad (150)$$

$$\sup\{[A_j]_0\} = \begin{cases} x_{\max} & \text{ha } j=c \\ \inf\{[A_{j+1}]_1\} & \text{ha } j=1, \dots, c-1. \end{cases} \quad (151)$$

A 103. ábra szemlélteti a 102. ábra elkészítéséhez használt kimeneti adathalmaz alapján létrehozott konzekvens partíciót.



103. ábra. Ruspini partíció előállítás $\alpha=0,85$ vágaton alapuló magbecslés segítségével a kimeneti adathalmazt jellemző klaszterekből

8.2.4. Vetítés az antecedens térbe

A fuzzy rendszer szabályait és az antecedens nyelvi értékeket az ún. vetítés segítségével állítjuk elő. A vetítés első lépéseként minden konzekvens nyelvi értékhez összeállítunk egy X_j adathalmazt, ami az $X = \{x^1, \dots, x^M\} \subset R^{N+1}$ mintaadathalmaz egy részhalmaza az alábbiak szerint

$$X_j = \left\{ x \mid \mu_{B_j}(x_{N+1}) = \max_{l=1, \dots, c} (\mu_{B_l}(x_{N+1})) \right\} \subset X, \quad (152)$$

ahol M a mintaadathalmaz pontjainak száma (felső indexben jelöltük az adatsor sorszámát), N az antecedens dimenziók száma, c a konzekvens partíció nyelvi értékeinek száma, x egy adatsor a mintaadathalmazból és x_{N+1} az adatsor kimeneti dimenzióbeli értéke (alsó indexben szerepel a dimenzió sorszáma). A (152) azt fejezi ki, hogy az X_j halmazba azok az adatsorok kerülnek, amelyek kimeneti értéke a konzekvens partíció j -edik fuzzy halmazában rendelkezik maximális tagsággal.

Következő lépésként soravesszük az X_j halmazokat. Minden halmaz esetén minden antecedens dimenzióban végrehajtunk egy egydimenziós FCM klaszterezést a 8.2.1. szakaszban megismert módon. Ezáltal minden konzekvens nyelvi értékhez hozzárendelünk minden antecedens dimenzióban egy vagy több klasztert. A klasztereket egy $A_{i,j,k}$ alakú egyedi azonosítóval látjuk el, ahol i az antecedens dimenzió sorszáma, j a konzekvens fuzzy

halmaz sorszáma, amelyikhez kapcsolódóan a klasztert létrehoztuk és k a klaszter adott klaszterezésen belüli sorszáma. A továbbiakban már csak a klaszterek azonosítóival ($A_{i,j,k}$) és középpontjaikkal (a $v_{i,j,k}$ - indexelés azonos az azonosítóknál használttal) dolgozunk, a mintaadatpontokat tovább nem használjuk.

Bár még nincsenek antecedens nyelvi értékeink, de a klaszterközéppontok és az azonosítók ismeretében elkészítjük a szabálybázis első változatát, ahol a szabályok az alábbi mintát követik

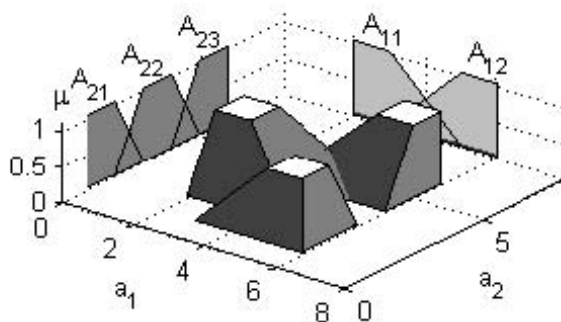
$$R_m : a_1 = A_{i,j,k_1} \text{ ÉS } \dots \text{ ÉS } a_N = A_{N,j,k_N} \text{ AKKOR } b = B_j \quad (153)$$

ahol m a szabály sorszáma és $a_1 \dots a_N$ az antecedens dimenziók nyelvi változói.

Ezt követően minden antecedens dimenzióban elkészítjük a klaszterközéppontok (és azonosítók) halmazát, növekvő sorrendbe rendezve a klaszterközéppontokat, és megvizsgáljuk ezek összevonhatóságát. Az összevonásokra az alábbi megfontolás következtében kerülhet sor.

Több bemenetű fuzzy rendszereknél egy antecedens nyelvi érték különböző konzekvenseket tartalmazó szabályok feltétel részében is szerepelhet. Például a 104. ábra egy két bemenetű és egy kimenetű rendszer ritka szabálybázisának antecedens terét illusztrálja, ahol az egyes szabályok antecedenseit csonka gúla jelképezik, és a partíciók Ruspini jellegűek. A kapcsolódó szabályok definícióját a (154), (155) és (156) ismerteti. Ebben az esetben az A_{12} halmaz két szabály feltétel részében is jelen van.

Mivel a vetítési elv következtében minden szabály minden antecedens nyelvi értékéhez külön klaszterközépponttal rendelkezünk, ezért könnyen előfordulhat, hogy két különböző szabályból eltérő azonosítójú, de valójában azonos klaszterközéppontot kapunk. Ilyen egybeesések esetén a megfelelő klaszterközéppontokat össze kell vonni.



104. ábra. Egy két bemenetű és egy kimenetű rendszer ritka szabálybázisának antecedens tere

$$R_1 : a_1 = A_{1,1} \text{ ÉS } a_2 = A_{2,2} \text{ AKKOR } b = B_1 \quad (154)$$

$$R_2 : a_1 = A_{1,2} \text{ ÉS } a_2 = A_{2,1} \text{ AKKOR } b = B_1 \quad (155)$$

$$R_3 : a_1 = A_{1,2} \text{ ÉS } a_2 = A_{2,3} \text{ AKKOR } b = B_2 \quad (156)$$

Numerikus módszerről lévén szó igen csekély annak az esélye, hogy a B_1 és a B_2 nyelvi értékek vetítéséből keletkező két érintett klaszter azonos középponttal fog bírni. Emiatt

nemcsak azokat a klasztereket vonjuk össze, amelyeknek középpontjai azonosak, hanem egy távolságkorlátot is definiálunk, és azokat a klasztereket vonjuk össze, amelyek középpontjainak távolsága kisebb a korlátnál. Összevonás után az új középpont a két eredeti középpont között félúton helyezkedik el. Az összevonást a szabálybázis korrekciója követi, ami a szabály antecedensekben érintett azonosítók cseréjét és a szabályok többszöri előfordulásának kiküszöbölését jelenti.

A távolságkorlátot az alaphalmaz adott dimenzióbeli szélességének (DR_i ld. (85)) függvényében határozzuk meg

$$dv_{\max} = C_U \cdot DR_i, \quad (157)$$

ahol a C_U együttható az eljárás paramétere.

A klaszter összevonásokat követően a 8.2.3. szakaszban ismertetett módon elvégezzük minden antecedens dimenzióban a fuzzy halmazok becslését. Az eredményül kapott összes partíció Ruspini típusú lesz, és az egybemenetű rendszereknél sűrű, míg az egynél nagyobb antecedens dimenziószámú esetekben várhatóan ritka szabálybázist kapunk. A vetítés algoritmusát a 105. ábra ismerteti.

```

MINDEN Konzekvens nyelvi értékre
    Rész adathalmaz képzése
    MINDEN antecedens dimenzióban
        Rész adathalmaz 1D FCM klaszterezése
    VégeMINDEN
    Szabályok képzése
VégeMINDEN
MINDEN antecedens dimenzióban
    Klaszterközéppont összevonhatóság vizsgálata és szükség
    esetén összevonás
VégeMINDEN
Antecedens fuzzy halmazok képzése
  
```

105. ábra. A vetítési algoritmus vázlata

8.2.5. Paraméter-azonosítás

Az ACP második lépése a modell paramétereinek azonosítása. A feladatot egy olyan iteratív algoritmussal oldottam meg, ami sokban hasonlít a 8.1.3. szakaszban ismertetett társához.

A fuzzy halmazok paraméterezési lehetőségei és a vonatkozó megkötések azonosak a 8.1.2. szakaszban ismertetettel. Az alkalmazható teljesítménymutatók megegyeznek a 8.1.5. szakaszban bemutatottakkal.

Minden iterációs menet során egyenként vesszük sorra az összes partíció minden egyes paraméterét. Az eljárás minden esetben két új értéket számít az aktuális paraméterhez úgy, hogy az eredeti értéket növeli, illetve csökkenti egy előre megadott lépésközzel (st_i), majd ezt követően alkalmazza a megkötéseket (ld. 8.1.4. szakasz).

Mindkét új érték esetében az algoritmus kiszámítja a fuzzy rendszer teljesítménymutatóját. Amennyiben az eredeti értékre vonatkozó mutató nem ismert, akkor arra is elvégzi a számításokat. A három lehetséges paraméter érték közül azt tartjuk meg, amelyik a legjobb teljesítménymutatót eredményezte. Az előzőekben vázolt technika egy hegymászó módszer,

ahol a gradiens irányát a pillanatnyi állapot értéke és a szomszédos állapotok értékének viszonyával becsüljük.

```

i=1
PIi=Nyers/kiinduló rendszert kiértékelése
PImin=PIi
ISMÉTLÉS AMÍG i ≤ MaxIterNo ÉS PIi ≥ PItr
    Minden be- és kimeneti dimenzióban
        Lépésköz számítása
        Minden nyelvi értékre
            Minden paraméterre
                Mindkét változtatási irányra
                    Új paraméter érték számítása
                    Megkötések érvényesítése
                    PIi=Nyers/kiinduló rendszer kiértékelése
                    HA PIi < PImin AKKOR
                        Paraméter érték tárolása
                        PImin=PIi
                    VégeHA
                VégeMindkét
            VégeMinden
        VégeMinden
    DPI=PIi-1-PIi
    HA DPI ≥ DPImax AKKOR
        Lépésköz együtttható növelése
    KÜLÖNBEN
        HA DPI ≤ DPImin AKKOR
            HA Lépésköz együtttható kicsi AKKOR
                HA Volt már véletlen együtttható AKKOR
                    VÉGE
                KÜLÖNBEN
                    Véletlen együtttható képzése
                VégeHA
            Lépésköz együtttható kezdeti értékre állítása
        KÜLÖNBEN
            Lépésköz együtttható csökkentése
        VégeHA
    VégeHA
    VégeHA
    i=i+1
VégeISMÉTLÉS

```

106. ábra. Az ACP paraméter-azonosítási algoritmus

Minden iterációs menet végén, amikor az algoritmus végighaladt egyszer a rendszer összes hangolni kívánt paraméterén, kiszámítjuk a teljesítménymutató változását (ΔPI ld. (123)), azaz az előző iterációs menet végén (PI_{k-1}), vagy az első iteráció esetén kiinduláskor (PI_0) mért teljesítménymutató és az aktuális teljesítménymutató (PI_k) különbségét (123).

Amennyiben a teljesítménymutató értéke jobban javult egy felső határértéknél ($\Delta PI \geq \Delta PI_{\max}$, ld. 8.1.5.), akkor a lépésköz nagyságát meghatározó együtttható értékét

megduplázzuk (124) az alsó és felső korlátok figyelembevételével (ld. 8.1.4.). Amennyiben a teljesítménymutató javulása egy előre megadott alsó határérték ($\Delta PI \leq \Delta PI_{\max}$, ld. 8.1.5.), alá esik, akkor az együttható értékét felezzük (125) a korlátok figyelembevételével (ld. 8.1.4.).

Amennyiben az együttható már korábban elérte alsó korlátját ($C_{\min}$), akkor nem osztjuk 2-vel, hanem véletlenszerűen egy új együtthatót képzünk a $[C_{\min}, C_{\max}]$ intervallumból. Ezután a C együtthatót újból kezdeti értékére állítjuk, majd folytatjuk az iterációt. A véletlenszerűen generált együttható célja az, hogy egy második esélyt adjunk az algoritmusnak, hogy az kikerüljön a teljesítménymutató lokális minimuma által jellemzett állapotból. A hangolás túlzott elhúzódnak elkerülése érdekében egy paraméter-azonosítási feladat során csak egy véletlen együtthatót engedélyezünk.

A paraméter-azonosítási folyamat akkor áll le, ha elértük az előre megszabott maximális iterációs menetszámot, vagy a teljesítménymutató elért egy előre meghatározott jósági értéket (PI_{tr}), vagy elértünk ahhoz a ponthoz, ahol a második véletlenszerű együttható előállítása következne. Az algoritmus leírását a 106. ábra ismerteti.

A jelen eljárás az RBE-DSS/SI paraméter-azonosítási algoritmusától alapvetően abban különbözik, hogy új szabályok generálása (a szabálybázis bővítése) helyett a lépésközt meghatározó együtthatónak véletlenszerű értéket adva próbálja kikököteni a teljesítménymutatót lokális minimumából. Ezáltal az algoritmus egy második esélyt kap az optimális paraméter értékek meghatározására.

8.3. RuleMaker - Automatikus rendszergenerálást támogató Matlab eljárásgyűjtemény

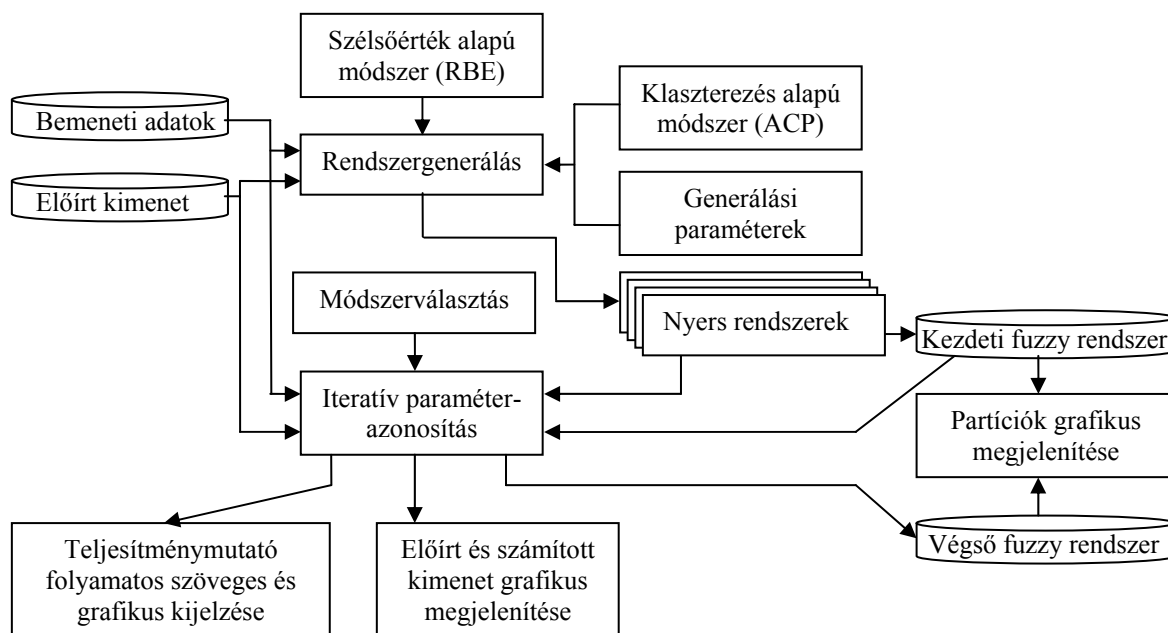
Az adatok alapján történő automatikus rendszergeneráláshoz egy ToolBox-ot fejlesztettem Matlab környezetben RuleMaker néven, amely szervesen kapcsolódik a szabály-interpolációs eljárásgyűjteményhez. A szoftver egyes függvényei külön-külön meghívhatók akár más alkalmazásból is, de készítettem egy grafikus felületű programot is, ami a teljes funkcionalitást kényelmesen elérhetővé teszi. A szoftver funkcionalitását alapvetően két témakörre bonthatjuk. Ezek a következők.

- Nyers fuzzy rendszer előállítása be- és kimeneti adatok alapján menüből választható módon klaszterezésen vagy a 8.1.1. szakaszban ismertetett szélsőértékeken alapuló módszer segítségével. Több kimenettel rendelkező adatsorok esetén minden kimeneti dimenzióhoz egy külön fuzzy rendszer keletkezik.
- Egy vagy több bemenettel és egy kimenettel rendelkező kiinduló (nyers) fuzzy rendszer iteratív, hegymászó típusú hangolása hatféleképpen választhatóan három különböző módszer (RBE-DSS, RBE-SI és ACP) és négy paraméterezési technika felhasználásával, és a választható három teljesítménymutató valamelyikének figyelembevételével.

A programban a paraméter-azonosítás során az összes olyan fuzzy következtetési módszer használható, amelynek implementációját tartalmazza a FRIT keretrendszer. A hangolási és következtetési paraméterek párbeszédpanelek segítségével állíthatók be. A szoftver képes a partíciók grafikus megjelenítésére, továbbá a hangolás folyamata során a numerikus kijelzés mellett grafikonon is követhető a teljesítménymutató alakulása. Az előírt és a számított kimeneti értékek is folyamatosan nyomon követhetők. A program belső struktúráját a 107. ábra mutatja be.

Ismerkedjünk meg egy egyszerű példán keresztül a szoftver működésével és kezelésével. Tegyük fel, hogy a 88. ábrán megjelenített görbe fuzzy szabályokkal történő leírása a cél az $X = [0, 10]$ intervallumban, és a fuzzy rendszert az RBE-DSS módszerrel kívánjuk előállítani.

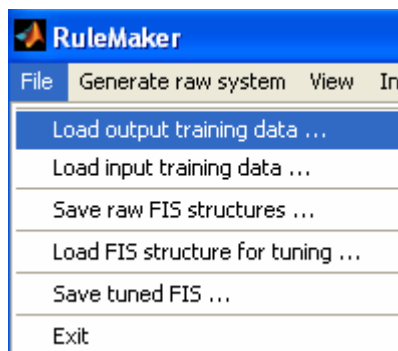
A többi módszer alkalmazása is hasonló módon történik, ezért bemutatásukra most nem kerül sor. A programot a *RuleMaker* paranccsal indítjuk.



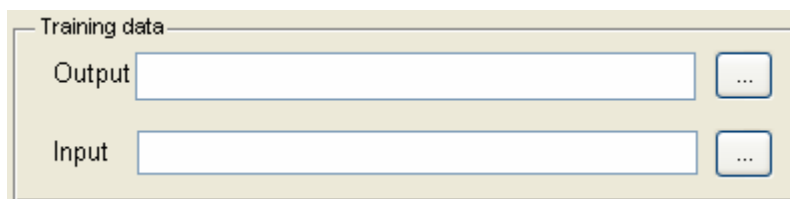
107. ábra. A rendszergeneráló program belső struktúrája

8.3.1. Bemeneti adatok

A rendszergenerálás és paraméter-azonosítás alapjául szolgáló mintaadatokat a TestIt alkalmazás bemutatásánál (ld. 7.2.1. fejezet) megismert módon szöveges állományok formájában várja a program.



108. ábra. File menü

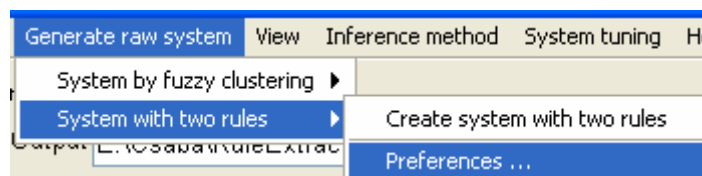


109. ábra. Mintaadatok beolvasását segítő Training data vezérlőcsoport

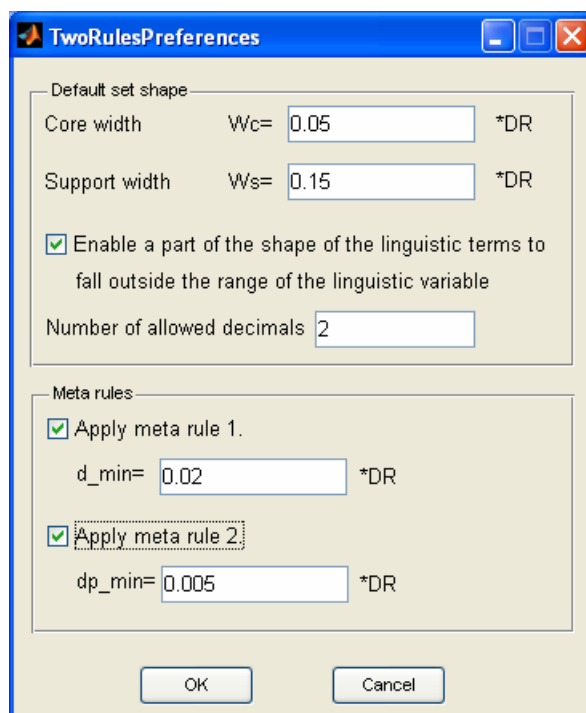
Az adatokat a *File* menü (108. ábra) *Load input/output training data ...* menüpontjai vagy a főablak *Training data* vezérlőcsoportjában megjelenő „...” feliratú nyomógombok (109. ábra) segítségével tölthetjük be. Mindkét esetben egy szabványos állomány kiválasztási/megnyitási párbeszédablak jelenik meg, amelyben kiválasztjuk a megfelelő adatfájlt.

8.3.2. Nyers rendszer létrehozása a kimenet szélsőértékein alapuló módszerrel

A kimenet szélsőértékein alapuló rendszergenerálás (ld. 8.1.1. szakasz) első lépéseként megadjuk az alapértelmezett mag és tartó szélességek ($W_{C,i}$ és $W_{S,i}$) együtthatóit, engedélyezzük a metasabályok alkalmazását, valamint beállítjuk a két metasabályban alkalmazott közelségi határértékek ($d_{\min}$ és $dp_{\min}$) C_d és C_{dp} együtthatóit. Az adatbevitelt szolgáló párbeszédablakot (ld. 111. ábra) a 110. ábrán bemutatott menüútvonalon érhetjük el.

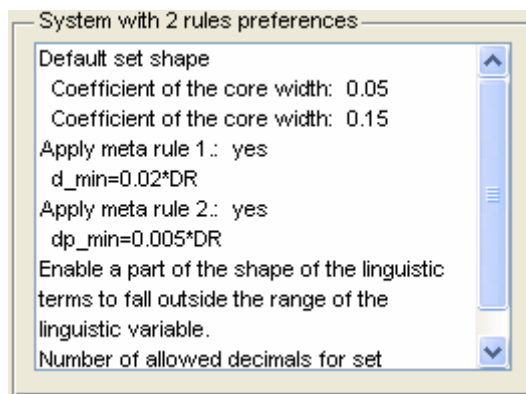


110. ábra. Beállítások



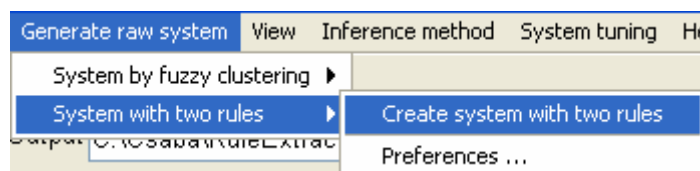
111. ábra. A nyers rendszer generálását befolyásoló paraméterek beállítása

A beállított alapértelmezések megjelennek egy csoportablakban (112. ábra).



112. ábra. Nyers rendszer létrehozása során alkalmazott beállítások

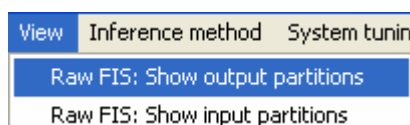
A rendszer generálását a *Create system with two rules* menüpont kiválasztása idézi elő (113. ábra). Több kimeneti dimenziós adathalmaz esetén a program minden kimeneti dimenzióhoz egy külön fuzzy rendszert állít elő.



113. ábra. Kétszabályos rendszer generálása menüpont

A program a mintaadatok alapján határozza meg az egyes dimenziókban érvényes alsó és felső korlátokat. A nyelvi változók nevei *input_x* bemeneti dimenzió és *output* kimeneti dimenzió esetén, ahol *x* a dimenzió sorszáma. A nyelvi értékek nevei $X_{\{U;V\}}$, ahol *X* helyett *A* áll bemeneti és *B* áll kimeneti dimenzió esetén, *U* a dimenzió sorszáma, *V* a nyelvi érték sorszáma a partícióján belül. A Matlab által használt Latex értelmezőnek köszönhetően a grafikus megjelenítésnél mindez $X_{U,V}$ alakban lesz látható.

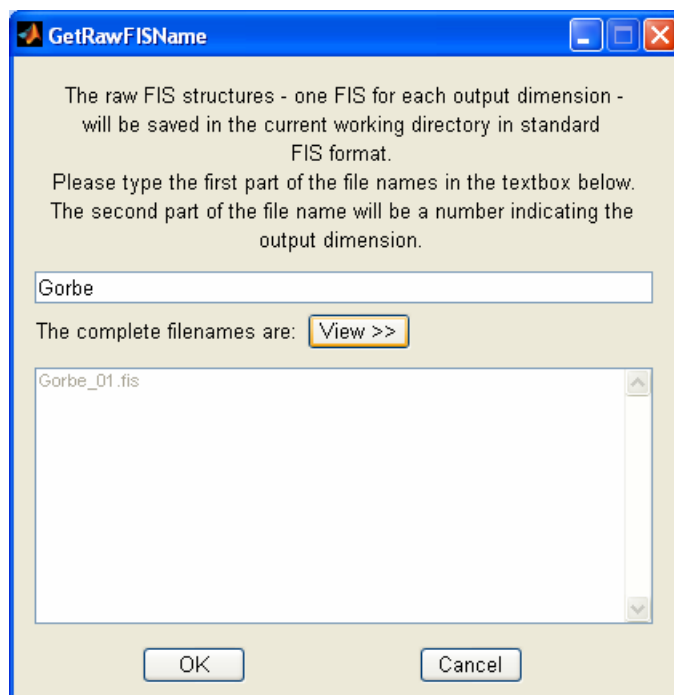
A nyers rendszerek antecedens és konzekvens partícióit a *View* menü segítségével tekinthetjük meg (ld. 114. ábra). Példánkban a 90. ábra két partíciója jelenik meg.



114. ábra. Nyers rendszer partícióinak megjelenítése

8.3.3. Mentés

A nyers rendszert a *File/Save raw FIS structures* menüponttal menthetjük háttértárolóra (ld. 110. ábra). A 115. ábrán látható párbeszédpanel segítségével állítjuk be az állomány nevét.

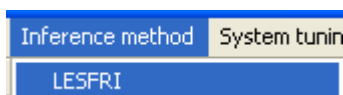


115. ábra. A fuzzy rendszereket leíró állományok elnevezése

Több kimeneti dimenziós adathalmaz esetén az egyes dimenziókhoz létrehozott fuzzy rendszereket úgy nevezi el a program, hogy az alapnév közös, és a név végén szereplő szám azonosítja be a kimeneti dimenziót. A rendszer mentése a korábban már megismert kiterjesztett FIS szöveges formátumban történik.

8.3.4. Paraméter-azonosítás és szabálybázis kiterjesztés

Egy korábban elkészített nyers fuzzy rendszer hangolását és szabálybázisának kiterjesztését a következő lépésekben hajtjuk végre. Amennyiben szükséges, akkor az előzőekben látott módon betöltjük a mintaadatokat tartalmazó állományokat és a hangolni kívánt nyers rendszert a menüpontok vagy vezérlőgombok használatával.



116. ábra. Szabály-interpolációs módszer választás

Többdimenziós kimenetű adatok esetén a kimeneti dimenzió megválasztása is szükséges. Ez a *System tuning/Select output dimension* menüpont alatt érhető el. Ezt követően az *Inference method* menü segítségével kiválasztjuk a használni kívánt fuzzy következtetési módszert, ami a jelen esetben a LESFRI lesz (116. ábra). A szabály-interpolációs eljárás paramétereit az *Inference method/Inference preferences ...* menüpont hatására megjelenő párbeszédablak segítségével adjuk meg (117. ábra).

The 'Preferences' dialog box contains the following settings:

- Interpolation method: LESFRI
- Reference point type: corecentre
- Weight type for the set interpolation FEAT-LS:
 - ☒ 1. $p=$ 2
 - ☐ 2. $\lambda=$
- Weight type for the calculation of the position of the consequent sets:
 - ☒ 1. $p=$ 2
 - ☐ 2. $\lambda=$
- Weight type used by SURE-LS:
 - ☒ 1. All antecedent dimensions have the same weight $w_e=1$
 - ☐ 2. Weights separated by whitespaces
 - Number of antecedent dimensions:
- ☒ Show antecedent and consequent of the interpolated rule
- ☒ Show modified consequent sets
- Number of points: 501
- Buttons: OK, Cancel

117. ábra. Következtetési paraméterek beállítása

The 'Inference preferences' dialog box displays the current values of the settings:

- Type of the interpolation: LESFRI
- Rtype: corecentre
- SetInterpolationWeight type: 1
- p : 2
- ConsequentPositionWeight type: 1
- p : 2

118. ábra. Következtetési paraméterek aktuális értékei

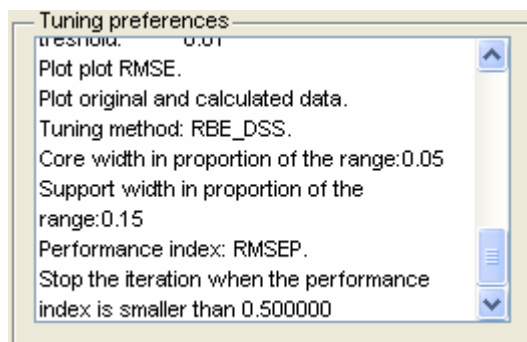
A beállított értékek egy csoportablakban (118. ábra) jelennek meg.

119. ábra. Hangolási folyamat beállításai

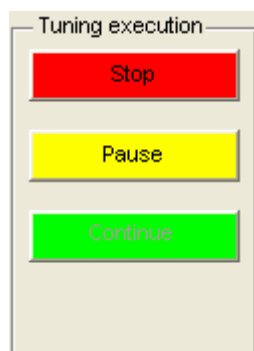
A következtetési jellemzők megadását követően beállítjuk/módosítjuk a paraméterazonosítási folyamat paramétereit (119. ábra). A beállított értékek ebben az esetben is kijelzésre kerülnek a főablak egy vezérlőjén (120). A szoftver jelenlegi változata az RMSE, RMSEP, RR, AD és ADP teljesítménymutatók használatát támogatja. Hangolási és paraméterezési módok viszonylatában a következő lehetőségek választhatók: RBE-DSS+töréspontok, RBE-DSS+referencia pont, RBE-SI+töréspontok, szabálybázis kiterjesztés nélküli hangolás+töréspontok, szabálybázis kiterjesztés nélküli hangolás+relatív távolságok, szabálybázis kiterjesztés nélküli hangolás+Ruspini partíció megőrzése. A következtetési és hangolási paraméterek között nem szerepel a defuzzifikáció módja. A program jelenleg csak a COG (Centre Of Gravity) módszer használatát teszi lehetővé.

A hangolást a *System tuning/System tuning* menüponttal indítjuk. Ezt követően a főablakon megjelenik a 121. ábrán látható nyomógomb csoport, ami lehetővé teszi a hangolás szüneteltetését, leállítását vagy újraindítását szüneteltetés után.

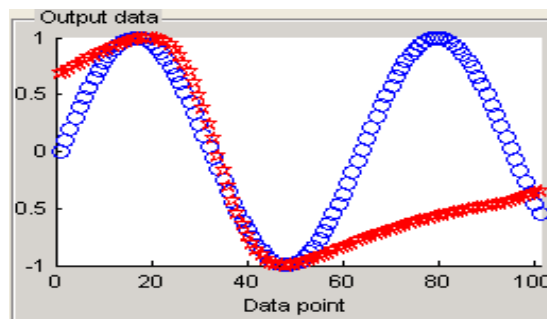
A rendszer által előállított és a mintaadatsorban megkövetelt kimenetet folyamatosan összevethetjük egy grafikon segítségével (ld. 122. ábra). Itt a vízszintes tengelyen az adatpontok sorszáma szerepel, mivel a fuzzy rendszer bemenete többdimenziós is lehet kék körök jelölik az elvárt és piros csillagok a rendszer által számított kimenetet.



120. ábra. Az érvényes beállítások kijelzése



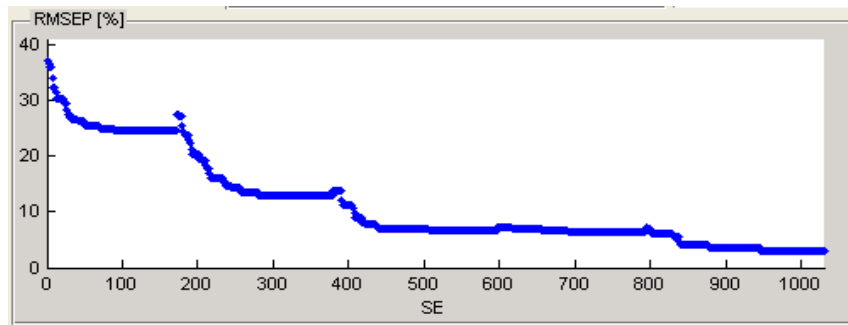
121. ábra. Leállítás, szüneteltetés és újraindítás vezérlése



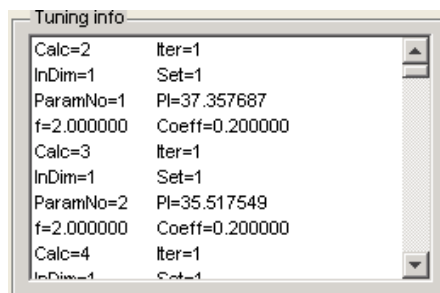
122. ábra. Elvárt és a rendszer által előállított kimeneti értékek a hangolás kezdetén

A hangolás pillanatnyi állapotáról szöveges és grafikus tájékoztatást kapunk. A grafikus kijelzés (123. ábra) a teljesítménymutató alakulását követi a rendszerkiértékelések számának (SE) függvényében. Mivel az egyes hangolási és paraméterezési módok eltérő rendszerkiértékelés számot eredményeznek egy iteráción belül, és a folyamat időigényét alapvetően a rendszerkiértékelések száma határozza meg, így az első látásra természetesnek tűnő iterációs szám helyett abszcisszaként a rendszerkiértékelések számát választottam.

A szöveges kijelzés (124. ábra) információt ad egyebek között a rendszerkiértékelések aktuális számáról, az iterációs számról, a módosítás alatt álló paraméterről és a teljesítménymutatóról.

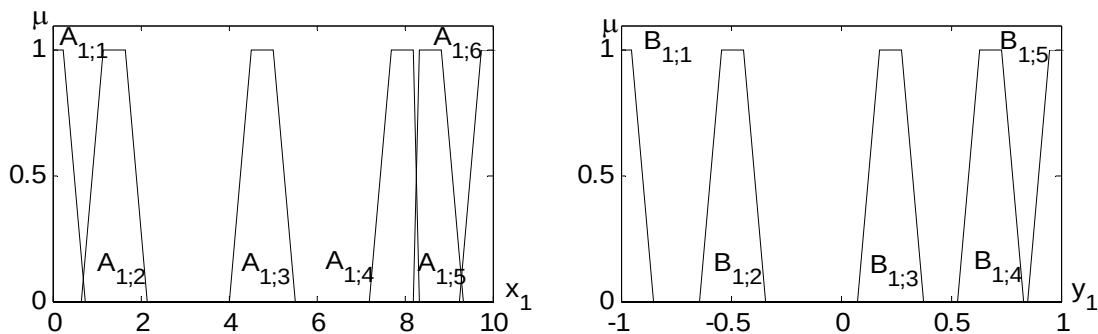


123. ábra. A teljesítménymutató alakulása a rendszer kiértékelések (SE) számának függvényében



124. ábra. A hangolás nyomonkövetését segítő vezérlő

A rendszergenerálás végeztével a File/Save tuned FIS menüponttal menthetjük háttértárolóra a kapott fuzzy rendszer leírását. A be- és kimeneti partíciókat megtekinthetjük a *View/Tuned FIS: Show input partitions* és *View/Tuned FIS: Show output partitions* menüpontok segítségével. Esetünkben az eredmény a 125. ábrán látható.



125. ábra. A hangolás eredményeképpen kapott antecedens és konzekvens partíciók

A rendszer szabálybázisát a 2. táblázat ismerteti. A táblázat mindkét részében az első oszlopban a szabály sorszáma, a második oszlopban az antecedens nyelvi érték sorszáma és a harmadik oszlopban a konzekvens nyelvi érték sorszáma szerepel.

2. táblázat. A rendszer szabálybázisa

Ssz.	Be	Ki	Ssz.	Be	Ki
1	1	3	4	4	5
2	2	5	5	5	4
3	3	1	6	6	2

9. Szabály-interpolációs módszerekkel dolgozó speciális fuzzy következtető rendszerek létrehozása modellezési feladatok megoldására

A fuzzy szabály-interpoláción alapuló LESFRI és FRIPOC következtetési módszerek, valamint az RBE-DSS, RBE-SI és ACP automatikus rendszergenerálási eljárások segítségével fuzzy modellezési feladatokat oldottam meg. Az első feladattípusnál (9.1. szakasz) szintetikus (függvények segítségével) előállított adatokból kiindulva készítettem SISO és MISO rendszereket. Ezt követően két gyakorlati feladatot oldottam meg, amelyeket a 9.2. és a 9.3. szakaszokban ismertetek.

9.1. Fuzzy rendszergenerálás szintetikus előállított adatok alapján

9.1.1. SISO rendszer

A kísérletek során elsőként olyan fuzzy rendszert kívántam előállítani, amely egy bemenettel és egy kimenettel rendelkezik (SISO). Az adatokat szintetikus képeztem a (86) függvény segítségével. A be- és kimenet közti kapcsolatot a 88. ábra szemlélteti. A mintaadathalmaz 101 adatpontból állt, amelyek a vízszintes tengely mentén egyenletesen szétosztva helyezkedtek el.

Külön-külön generáltam fuzzy rendszereket az RBE-DSS és RBE-SI eljárások, valamint a LESFRI és a FRIPOC következtetési módszerek segítségével, teljesítménymutatóként pedig az RMSE-t alkalmaztam.

Mindkét következtetési módszer mellett az RBE-DSS eredményezett jobb teljesítménymutatót (3. táblázat), és mindkét rendszergenerálási eljárás alkalmazásakor a LESFRI-hez készített fuzzy rendszer bizonyult jobbnak. A kapott eredmények a [S22] irodalomban lettek közzétéve.

3. táblázat. SISO rendszer teljesítménymutatói

	LESFRI	FRIPOC
RBE-DSS	0,0709	0,0915
RBE-SI	0,2252	0,3742

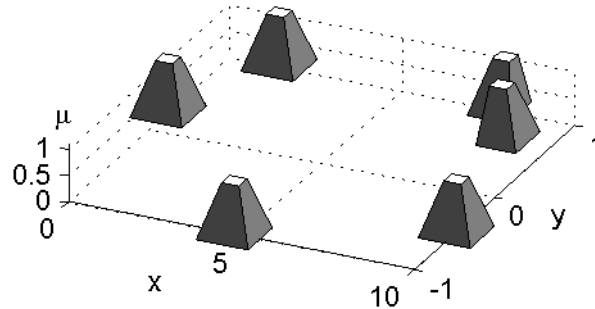
A hangolási algoritmus kismértékű módosítását követően sikerült jobb eredményeket biztosító fuzzy rendszereket generálni az RBE-DSS segítségével a LESFRI módszerhez. A kapott rendszert mindhárom szabály-interpoláció alapú következtetési módszerrel kipróbáltam, és a 4. táblázatban szereplő teljesítménymutatókat kaptam.

4. táblázat. SISO rendszer teljesítménymutatói

	LESFRI	FRIPOC	VEIN
Tanító adathalmaz	0,0582	0,0645	0,0774
Teszt adathalmaz	0,0559	0,0623	0,0762

Ebben az esetben teszt adathalmazt is alkalmaztam, ami 100 adatpontból állt. A kapott rendszer szabálybázisát a 126. ábra mutatja be. Jól megfigyelhető, hogy a szabályok

illeszkednek a görbe fordulópontjaihoz, Kosko [H32] által tárgyalt ún. optimális fuzzy szabályoknak megfelelően.



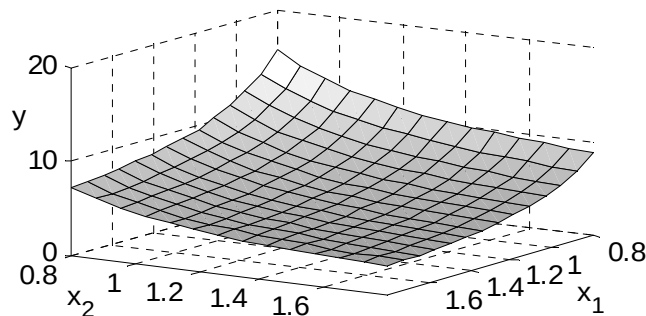
126. ábra. Az RBE-DSS rendszergenerálási módszerrel LESFRI következtetéshez készített SISO fuzzy rendszer szabálybázisa

9.1.2. MISO rendszer

A második kísérletsor keretében MISO fuzzy rendszer előállítása volt a cél. Az adatokat szintetikusán képeztem a

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2 \quad x \in [0.8, 1.8] \quad (158)$$

függvény segítségével. Itt egyebek között a [H72] irodalomban közzétett eredményekkel való összehasonlítás volt a cél. A be- és kimenet közti kapcsolatot a 127. ábra szemlélteti. A mintaadathalmaz 196 adatpontból állt, amelyek az antecedens térben egyenletesen szétosztva helyezkedtek el.



127. ábra. A második rendszer bemenete és kimenete közti kapcsolat

Külön-külön generáltam fuzzy rendszereket az RBE-DSS és RBE-SI eljárások segítségével a LESFRI és a FRIPOC következtetési módszerek segítségével, az RMSE-t alkalmaztam teljesítménymutatóként.

Mindkét következtetés módszer esetében az RBE-SI eredményezett jobb teljesítménymutatót (5. táblázat), és mindkét rendszergenerálási eljárás alkalmazásakor a LESFRI-hez készített fuzzy rendszer bizonyult jobbnak. A kapott eredmények a [S22] irodalomban lettek közzétéve.

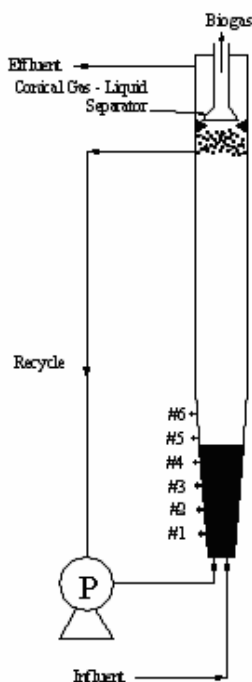
5. táblázat. MISO rendszer teljesítménymutatói

	LESFRI	FRIPOC
RBE-DSS	0,2701	0,4143
RBE-SI	0,1809	0,2143

Az RBE-SI LESFRI párossal elért legjobb eredmény (0,1809) kissé elmarad a [H72] irodalomban közzétett hatféle megközelítéssel kapott hat teljesítménymutató legjobbjától (0,1794), de jobb, mint az ott közzétett másik öt eredmény.

9.2. Anaerob kúpos szuszpendált ágy reaktor működésének fuzzy modellezése

A Sri Venkateswara College of Engineering indiai egyetemmel való együttműködés keretében a feladatom az volt, hogy az általuk készített anaerob kúpos szuszpendált ágy reaktoron (ATFBR – Anaerob Tapered Fluidized Bed Reactor) végrehajtott kísérletek eredményei alapján olyan fuzzy rendszereket hozzak létre, amelyek modellezik a négy bemenet és öt kimenet közötti kapcsolatot kimenetenként egy-egy külön fuzzy rendszer segítségével. Az ATFBR rendszer célja a börgyártás során keletkező ipari szennyvíz tisztítása, felépítését a 128. ábra ismerteti vázlatosan. A rendszer leírását és az elért eredményeket a [S28] irodalomban publikáltuk.



128. ábra. Az ATFBR rendszer sematikus bemutatása

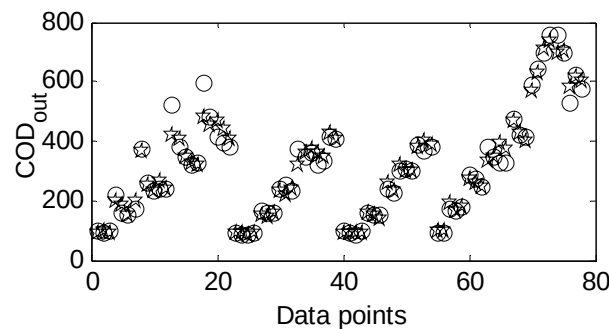
Az adathalmaz 78 adatsort tartalmazott. A négy bemenet jelentése a következő: átfolyási sebesség (flow rate), kémiai oxigén igény bemeneti oldalon (COD_{in} – Chemical Oxygen Demand In), pH és bemeneti biológiai oxigén igény (BOD_{in} – Biological Oxygen Demand In). A kimeneti adatok a következők: COD kimeneti oldalon (COD_{out}), biogáz, illózsírsav (VFA – Volatile Fatty Acid), lúgosság (Alkalinity), kimeneti biológiai oxigén igény (BOD_{out}). A be- és kimeneti alaphalmazok alsó és felső korlátai nem a mintaadathalmazból lettek megállapítva, hanem előírt értékek voltak.

A rendszerek generálására és hangolására először az ACP algoritmus első változatát és a FRIPOC szabály-interpolációs következtetési módszert alkalmaztam. Teljesítménymutatóként az RMSE-t és az RMSEP-t használtam. A 6. táblázat ismerteti a generált rendszerek teljesítménymutatóit.

6. táblázat. A generált rendszerek teljesítménymutatói

Kimenet	PI _{RMSE}	PI _{RMSEP} (%)
COD _{out}	30,1954	4,47
Biogas	0,8012	2,46
VFA	18,2828	7,75
Alkalinity	76,0786	9,64
BOD _{out}	88,4201	9,96

Mivel minden rendszer négy bemenettel rendelkezik, ezért az előállított és a kísérletek során mért kimenetek összehasonlítására olyan diagramokat készítettem, amelyeknek vízszintes tengelyén az adatsorok sorszáma szerepel. Az adatpontok ábrázolásánál (129., 131., 133., 135. és 137. ábrák) körök jelölik a mért értékeket, és csillagok jelölik a fuzzy rendszerek által számított értékeket.

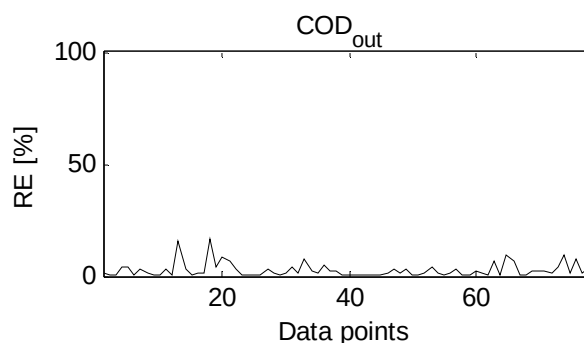


129. ábra. Mért (körök) és számított (csillagok) adatok az első kimeneti dimenzióban (COD_{out})

Minden fuzzy rendszer esetén megvizsgáltam a relatív hibát, amit a terjedelem százalékában kifejezett abszolút értékű eltérésekkel számoltam

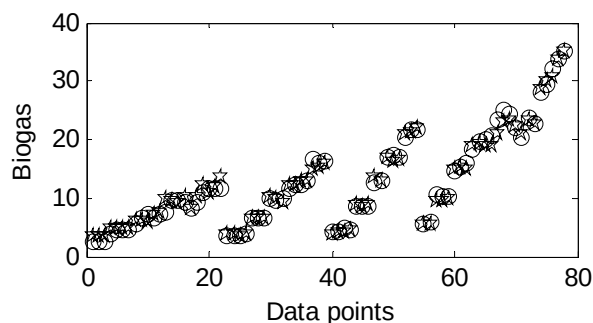
$$RE_i = \frac{|y_i - \hat{y}_i|}{DR_i} \cdot 100, i = 1, \dots, 5, \quad (159)$$

ahol i az aktuális kimeneti dimenzió, y_i a mért érték, $\hat{y}_i$ a fuzzy rendszer által számított érték és DR_i a terjedelem, amit a (85) képlettel határozunk meg. A relatív hibák alakulását (130., 132., 134., 136. és 138. ábrák) az adatpontokhoz hasonlóan úgy ábrázoltam, hogy az adatpontok sorszáma szerepel a vízszintes tengelyen.

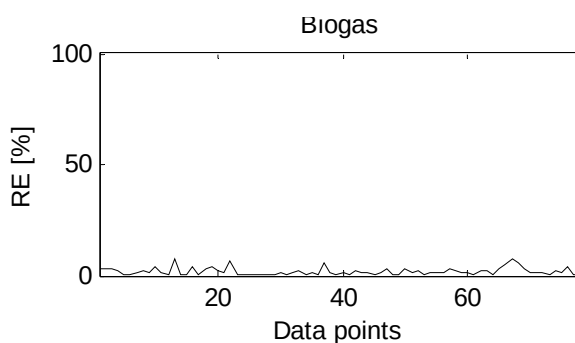


130. ábra. Relatív hiba az első kimeneti dimenzióban (COD_{out})

A COD_{out} esetében viszonylag jó eredményeket sikerült elérni (ld. 129. és 130. ábrák). A 130. ábrán jól látható, hogy csak néhány pont esetében szignifikáns az eltérés. A biogáz és a bemenet közötti kapcsolatot modellező rendszer (ld. 131. és 132. ábrák) hozta a legjobb eredményt, a körök és a csillagok jól illeszkednek egymásra.

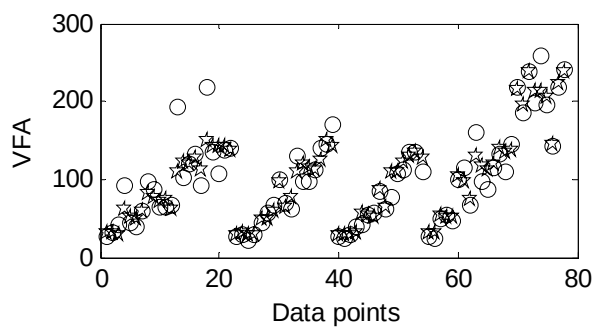


131. ábra. Mért (körök) és számított (csillagok) adatok a második kimeneti dimenzióban (biogáz)

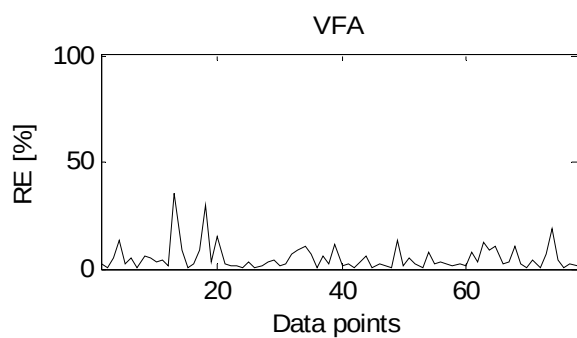


132. ábra. Relatív hiba a második kimeneti dimenzióban (biogáz)

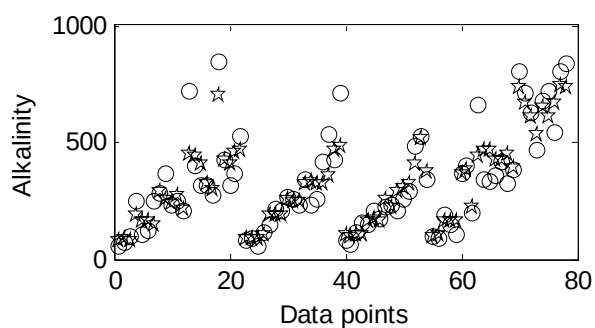
Az illózsírsavak és a lúgosság esetében (ld. 133., 134., illetve 135. és 136. ábrák) az eredmény közepesnek mondható. A relatív hiba diagramok csak három-négy csúcsot mutatnak mindkét esetben. Itt valószínűsíthető, hogy az eredmények javíthatók újabb szabályok bevitelével és a hangolási algoritmus fejlesztésével.



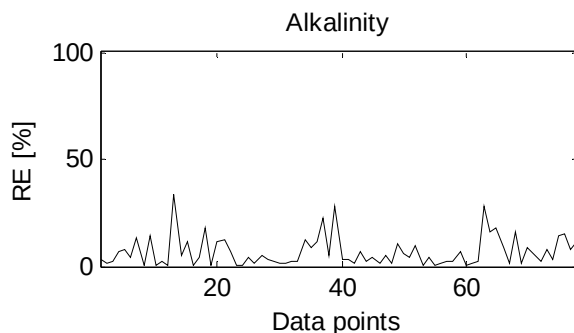
133. ábra. Mért (körök) és számított (csillagok) adatok a harmadik kimeneti dimenzióban (illózsírsav)



134. ábra. Relatív hiba a harmadik kimeneti dimenzióban (illózsírsav)

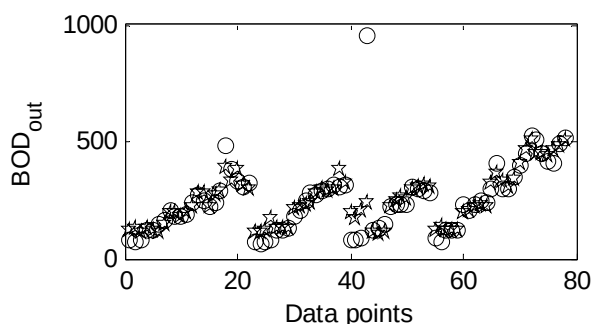


135. ábra. Mért (körök) és számított (csillagok) adatok a negyedik kimeneti dimenzióban (lúgosság)

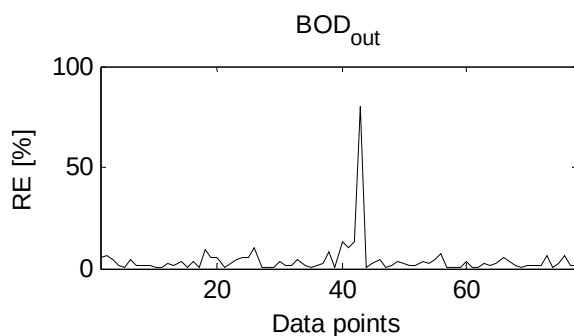


136. ábra. Relatív hiba a negyedik kimeneti dimenzióban (lúgosság)

Az utolsó kimeneti dimenzió esetében (ld. 137. és 138. ábrák) egyetlen adatpont erősen elkülönül az összes többitől. Az *RE* diagramon (138. ábra) is megjelenik, hogy míg az adatpontok fennmaradó részénél a fuzzy rendszer által számított adatok jól fedik a mért adatokat, itt egy erős eltérés tapasztalható. Valószínűsíthető, hogy egy mérési vagy beállítási hibával állunk szemben. Ezen feltevés helyességéről azonban nem állt módomban megbizonyosodni, mivel a kísérletek megismétlésére nem nyílt lehetőség.



137. ábra. Mért (körök) és számított (csillagok) adatok az ötödik kimeneti dimenzióban



138. ábra. Relatív hiba az ötödik kimeneti dimenzióban

A szennyvíztisztítás-modellezési feladat keretében további két fuzzy modellt generáltam RBE-DSS módszerrel a LESFRI típusú következtetéshez. Az eredmények eddig nem lettek publikálva, ezért azokat az 1. mellékletben ismertetem részletesebben.

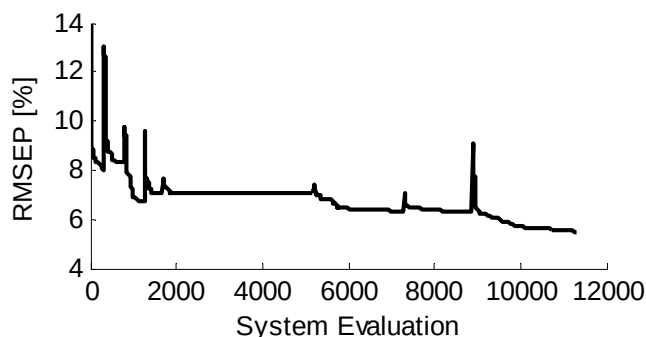
9.3. Kőolajkutatás során végzett fúrások mérési eredményei közötti összefüggés fuzzy modellezése

A Murdoch Egyetemen (Perth, Ausztrália) folytatott együttműködés keretében hozzájutottam kőolajkutatás során végzett fúrások mérési eredményeihez. Az irodalomban korábban publikált [H79] adatokkal történő összehasonlítás érdekében az RBE-DSS módszer segítségével először a LESFRI majd a FRIPOC szabály-interpolációs módszerhez fuzzy rendszert generáltam. Az eredményeket az [S10] irodalomban publikáltam.

A kőolajkutatási fúrások idő- és költségigényességéből adódóan a kitermelés gazdaságosságának előrejelzése nagy jelentőséggel bír. Az eredmények elemzése során az egyik legfontosabb feladat a petrofizikai tulajdonságok becslése a bemeneti adatok alapján. Ilyen tulajdonságok a porozitás, permeabilitás és agyagmenyiség [H16].

A modellezés során olyan kis szabályszerű MISO (3Be1Ki) fuzzy rendszer előállítás volt a feladat, amely a gamma sugárzás (GR), a mélységi indukciós ellenállás (deep induction resistivity) (ILD) és az akusztikus terjedési idő (DT), mint bemenő adatok és a porozitás (PHI), mint kimenő adat közötti kapcsolatot modellezi.

A tanító mintaadathalmaz 71 adatsort tartalmazott, a tesz adathalmaz 51 adatsorból állt. Az adathalmazokat előfeldolgozás után kaptam meg, az értékek minden dimenzióban a [0, 1] intervallumra voltak normalizálva. A rendszergenerálás során az RMSEP (130) teljesítménymutatóval dolgoztam. A 139. ábra ismerteti a teljesítménymutató alakulását a hangolási folyamat során a LESFRI típusú rendszer esetén. A korábbiakhoz hasonlóan itt is a rendszerkiértékelések száma szerepel a vízszintes tengelyen. A korábban publikált [H79] adatokkal történő összehasonlítás érdekében a kapott rendszerre kiszámoltam a korrelációs együtthatót, ami a mért és számított adatok kapcsolatára világít rá.



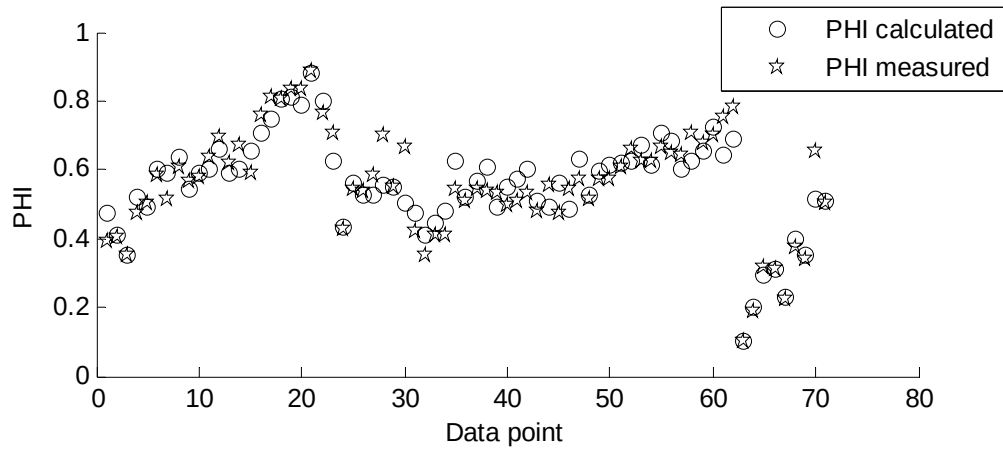
139. ábra. A teljesítménymutató alakulása a rendszergenerálás során a rendszerkiértékelések számának függvényében RBE-DSS és LESFRI alkalmazásánál

Az elért eredmények (7. táblázat) mindkét rendszerrel úgy a tanító, mint a teszt adathalmaz esetében jobbnak bizonyultak a [H79] irodalomban közzétettekénél.

7. táblázat. A hangolt rendszer teljesítménymutatói tanító és teszt adathalmaz esetén

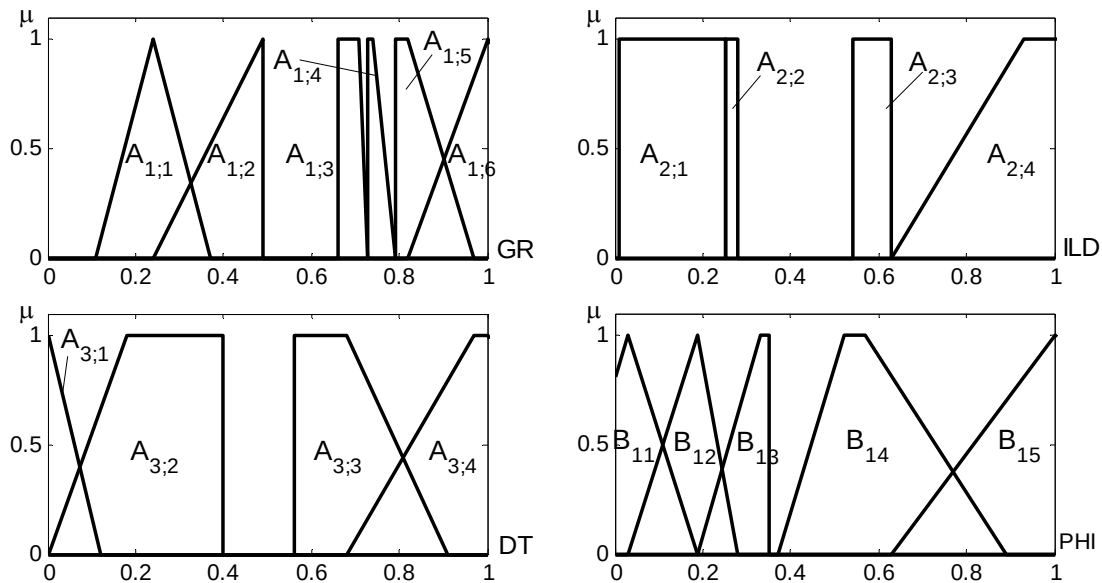
Alkalmazott módszer	Korrelációs együttható	
	Tanító adathalmaz	Teszt adathalmaz
[H79]	0,917	0,865
RBE-DSS + LESFRI	0,934	0,890
RBE-DSS + FRIPOC	0,966	0,891

Az RBE-DSS-el előállított rendszerek további előnye, hogy míg a [H79]-ben 63 szabályra volt szükség a 7. táblázatban szereplő eredményekhez, addig az RBE-DSS + LESFRI párossal kilenc szabállyal, és a RBE-DSS + FRIPOC párossal harminchét szabállyal értem el a táblázatban jelzett eredményeket.

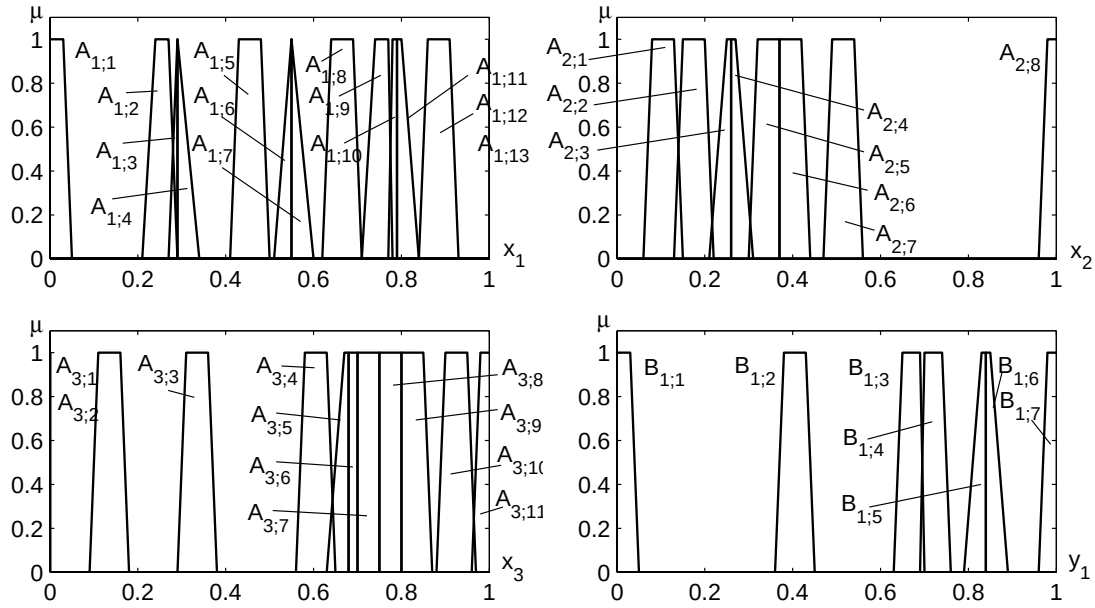


140. ábra. Mért (körök) és a generált rendszer által számított (csillagok) porozitás értékek a tanító adathalmaz esetében a LESFRI rendszer esetében

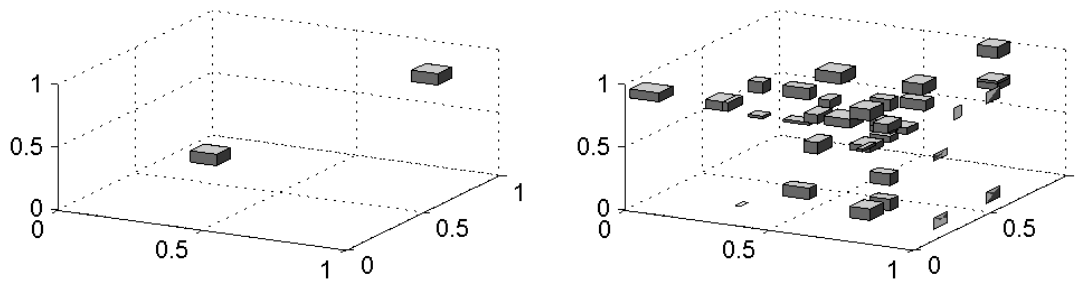
A tanító adathalmaz esetében a 140. ábra ad képet a mért és számított adatok közelségéről a LESFRI rendszer esetében. A három antecedens és egy konzekvens partíciót a 141. és 142. ábra mutatja be.



141. ábra. A GR, ILD, DT $\rightarrow$ PHI kapcsolatot modellező RBE-DSS + LESFRI-vel előállított fuzzy rendszer partíciói



142. ábra. A GR, ILD, DT $\rightarrow$ PHI kapcsolatot modellező RBE-DSS + FRIPOC-kal előállított fuzzy rendszer partíciói



143. ábra. Az RBE módszerrel előállított kezdeti szabálybázis (bal oldalon) és a FRIPOC típusú következtetéshez kialakított rendszer (jobb oldalon) szabálybázisának antecedens tere

A 143. ábra bal oldala illusztrálja az RBE-DSS módszer első lépésében előállított kétszabályos rendszer szabálybázisának antecedens terét. A jobb oldalon a modellidentifikáció után kapott szabálybázis antecedens terét láthatjuk. Három bemenetű rendszer lévén csak az egyes halmazok tartói ábrázolhatók, ezek definiálják a téglatestek éleit. Minden téglatest egy szabály antecedensét jelképezi. Jól megfigyelhető, hogy az eredményül kapott szabálybázis ritka.

10. Új tudományos eredmények tézisekben történő összefoglalása

Az alábbiakban ismertetem a kutatási eredményeimhez kapcsolódó téziseimet és felsorolom azokat a publikációimat, amelyekben az egyes tézisek tartalmi része közzé lett téve.

- 1. tézis.** *Új halmaz-interpolációs eljárásokat vezettem be a fuzzy szabályinterpoláció általánosított módszertanához és ritka szabálybázissal rendelkező fuzzy modellek identifikációjához.*
- 1.1. altézis.** *Igazoltam, hogy az általam kidolgozott, polár-vágaton (FEAT-p), legkisebb négyzetek elvén (FEAT-LS) és bizonytalan környezetben (VESI) alapuló halmaz-interpolációs eljárások alkalmazhatók a szabály-interpoláció általánosított módszertanában az új szabály antecedens és konzekvens halmazainak előállítására minden olyan esetben, amikor a partíció ismert halmazai konvexek és tartójuk korlátos.*
- 1.2. altézis.** *Igazoltam, hogy az általam kidolgozott halmaz-interpolációs eljárások alkalmazhatók a szabálybázis kiterjesztés elvén alapuló (RBE-DSS és RBE-SI) fuzzy modell-identifikációban új fuzzy halmazok előállítására, amikor a partíció ismert halmazai konvexek és tartójuk korlátos.*
- 1.3. altézis.** *Elkészítettem a FEAT-p, FEAT-LS és VESI halmaz-interpolációs módszereket megvalósító számítógépes eljárásokat és az őket implementáló szoftvert, amely a FEAT-p esetében sokszög és Gauss görbe alakú, a FEAT-LS és VESI esetében trapéz alakú tagsági függvények esetén alkalmas halmaz-interpolációs feladat megoldására.*

A téziscsoport tartalmi része publikálásra került az [S1], [S2], [S3], [S6], [S12], [S13], [S15], [S16], [S19], [S21] és [S26] kiadványokban.

- 2. tézis.** *Igazoltam, hogy az általam kidolgozott, polár-vágaton (SURE-p), legkisebb négyzetek elvén (SURE-LS) és bizonytalan környezetben (REVE) alapuló egyszabályos következtetési eljárások alkalmazhatók a szabály-interpoláció általánosított módszertanában a következmény számítására minden olyan esetben, amikor a partíció ismert halmazai konvexek és tartójuk korlátos.*
- 2.1. altézis.** *Elkészítettem a SURE-p, SURE-LS és REVE egyszabályos következtetési módszereket megvalósító számítógépes eljárásokat és az őket implementáló szoftvert, amely trapéz alakú tagsági függvények esetén alkalmas a következmény számítására.*

A téziscsoport tartalmi része publikálásra került az [S12], [S13], [S23] és [S26] kiadványokban.

- 3. tézis.** *Egy fuzzy szabály-interpolációs alkalmazási keretrendszert dolgoztam ki és implementáltam (FRI ToolBox), valamint megmutattam, hogy a keretrendszer alkalmas szabály-interpolációval dolgozó fuzzy következtető rendszerek megvalósítására, összehasonlítására és értékelésére.*
- 3.1. altézis.** *Egységes adatszerkezetet dolgoztam ki fuzzy rendszerek és megfigyelések számítógépes ábrázolására és tárolására egyszerű szöveges és XML alapú formátumban.*

- 3.2. altézis.** *Számítógépes eljárásokat dolgoztam ki és implementáltam tíz szabály-interpolációs módszerhez singleton, háromszög és trapéz alakú tagsági függvények esetére.*
- 3.3. altézis.** *Megmutattam, hogy az FRI ToolBox részeként elkészített TestIt és FRIT szoftverek alkalmasak éles és fuzzy bemenetű fuzzy rendszerek értékelésére és a különböző következtetési módszerek által adott eredmények összehasonlítására.*

A téziscsoport tartalmi része publikálásra került az [S27] és [S29] kiadványokban.

- 4. tézis.** *Új eljárásokat javasoltam ritka szabálybázisú fuzzy modellek mintaadatok alapján történő automatikus identifikációjára, és megmutattam, hogy segítségükkel az irodalomban közzétettnél kedvezőbb tulajdonságú fuzzy rendszer generálható.*
- 4.1. altézis.** *Kifejlesztettem a szabálybázis kiterjesztés elvét (RBE), és kidolgoztam két, azon alapuló fuzzy modell-identifikációs módszert (RBE-DSS és RBE-SI). Számítógépes eljárást és implementációt dolgoztam ki az RBE-DSS és RBE-SI módszerekhez trapéz alakú tagsági függvények esetére.*
- 4.2. altézis.** *Számítógépes eljárást és implementációt dolgoztam ki a fuzzy klaszterezésen alapuló (ACP) modell-identifikációs módszerhez trapéz alakú tagsági függvények esetére.*
- 4.3. altézis.** *Szintetikus úton (ismert függvénnyel) előállított mintaadathalmazból RBE-DSS és RBE-SI módszerekkel olyan fuzzy rendszereket generáltam, amelyek az első két tézis eljárásait alkalmazó szabály-interpolációs módszerek (FRIPOC, LESFRI és VEIN) segítségével állítják elő a következményt. Kimutattam, hogy az irodalomban közzétettnél kedvezőbb teljesítménymutató érhető el ezen eljárásokkal.*
- 4.4. altézis.** *Egy anaerob kúpos szuszpendált ágy reaktor működésének fuzzy modellezéséhez készített fuzzy rendszerek segítségével igazoltam az ACP és a FRIPOC módszerek alkalmazhatóságát.*
- 4.5. altézis.** *Kőolajkutatás során végzett fűrésok mérési eredményei közötti összefüggés modellezésére készítettem LESFRI és FRIPOC következtetéssel dolgozó fuzzy modelleket az RBE-DSS rendszergenerálási módszer segítségével. Kimutattam, hogy ezen eljárásokkal az irodalomban korábban közzétettnél kedvezőbb teljesítménymutatók érhetők el.*

A tézis tartalmi része publikálásra került az [S10], [S22], [S25], [S11] és [S28] kiadványokban.

11. Az elért eredmények hasznosíthatósága és további kutatási tervek

Mesterségesen előállított és mérésekből származó adatok alapján végzett futtatások azt mutatják, hogy az általam kifejlesztett automatikus fuzzy rendszergenerálási és –hangolási módszerek alkalmasak fuzzy rendszerek előállítására olyan esetekben, amikor a modellezett jelenségre vonatkozó ismeretek összefüggő be- és kimeneti adatok formájában állnak rendelkezésre.

A feladatot megoldó szoftver szabadon hozzáférhető és letölthető az általam üzemeltetett [S31] webhelyről Matlab eljárásgyűjtemény formájában, így könnyen beépíthető más alkalmazásokba, vagy meghívható más alkalmazásokból. A könnyen kezelhető grafikus felülettel rendelkező program önállóan, Matlabtól függetlenül telepíthető formában is rendelkezésre áll.

A saját fejlesztésű három halmaz-interpolációs eljárás alapvetően két területen alkalmazható. Az egyik feladattípus az RBE-SI fuzzy rendszergenerálási módszer használata esetén az iteratív szabálybázis bővítés során az új nyelvi értékek előállítása. A másik feladattípus a szabály-interpoláción alapuló fuzzy következtetés.

A saját fejlesztésű halmaz-interpolációs és egyszabályos következtetési módszerek felhasználásával kialakított három szabály-interpoláción alapuló fuzzy következtetési eljárás ritka szabálybázist alkalmazó fuzzy rendszer következtető motorjaként is alkalmazható.

A 3. tézis keretében kidolgozott eljárásgyűjtemény fuzzy szabály-interpolációs módszerek konkrét implementációját tartalmazza. A kifejlesztett keretrendszer, mint segédeszköz számos fuzzy szabály-interpolációs eljárás vizsgálatát és összevetését teszi lehetővé.

Az általam kialakított és üzemeltetett, a fuzzy szabály-interpoláció témakörére összpontosító webhely [S31] célja, hogy széles körű és folyamatosan frissülő információt nyújtson a fuzzy szabály-interpoláció témaköréhez kapcsolódó módszerekről és gyakorlati alkalmazásokról. Mindemellett számos a témával foglalkozó cikk és szoftver szabad hozzáférését biztosítja. A webhely támogatja az FRI témával kapcsolatos kutatást és az eredmények népszerűsítését. Mindemellett segédeszközként is hasznosítható a mesterséges intelligencia oktatás, illetve a témához kapcsolódó TDK, és diplomamunkák, szakdolgozatok készítése során.

Az értekezés elkészítésével nem tekintem lezártnak a fuzzy szabály-interpoláció témaköréhez kapcsolódó kutatási munkámat. A továbbiakban tervezett kutatási célkitűzéseim az alábbi pontokban foglalhatók össze.

- A keretrendszerben rendelkezésre álló fuzzy következtetési módszerek körének bővítése.
- Az egyes módszerek tulajdonságainak vizsgálata, összevetése.
- A mintaadatok alapján történő automatikus fuzzy rendszergenerálási módszerek finomítása és továbbfejlesztése, különös tekintettel a paraméter-azonosítási algoritmus hatékonyságára.
- Egy webszolgáltatás alapú osztott architektúrán alapuló fuzzy rendszergeneráló megoldás kifejlesztése.
- A gyakorlati alkalmazási lehetőségek bővítése és további alkalmazáspéldák kidolgozása.

12. Összefoglalás

Az értekezés elkészítéséhez kapcsolódó kutatómunkám során két fő témakörrel foglalkoztam. Ezek a fuzzy szabály-interpoláción alapuló következtetés és a mintaadatok alapján történő automatikus fuzzy modell-identifikáció.

A fuzzy szabály-interpoláción alapuló következtetési módszerek a hagyományos kompozíciós megközelítés korlátain túllépve olyan esetekben is képesek értelmezhető eredmény előállítására, amikor nem áll rendelkezésre alkalmazható szabály minden elképzelhető bemenethez.

Az ismert szabály-interpolációs módszerek tanulmányozását követően olyan eljárások kidolgozását tűztem ki célul, amelyek a fuzzy szabály-interpoláció általánosított módszertanát [H4] követik és a lehető legszélesebb alkalmazási területet biztosítják.

Első téziscsoportom a fuzzy szabály-interpoláció általánosított módszertanában az interpolált szabály antecedens és konzekvens halmazainak meghatározásával foglalkozik. Ebben a témakörben három új halmaz-interpolációs módszert (FEAT-p [S16], FEAT-LS [S15] és VESI [S2]) és a hozzájuk kapcsolódó számítógépes eljárásokat dolgoztam ki.

A halmaz-interpolációs eljárásokkal párhuzamosan, azokat kiegészítve szabálmódosításon alapuló egyszabályos következtetési módszereket (SURE-p [S12], SURE-LS [S13] és REVE [S23]) és hozzájuk kapcsolódó számítógépes eljárásokat dolgoztam ki. Ezen kutatómunka eredményét második téziscsoportomban foglaltam össze.

Az első két tézisben szereplő halmaz-interpolációs és egyszabályos következtetési eljárások felhasználásával három új szabály-interpolációs módszert dolgoztam ki FRIPPOC [S12], LESFRI [S13] és VEIN [S26] néven.

A különböző szabály-interpolációs fuzzy következtetési módszerek összehasonlításának, gyakorlati alkalmazásának és értékelésének nehézségeit felismerve olyan egységes adatszerkezetet, keretrendszert és implementációt dolgoztam ki, amely lehetővé teszi ezen eljárások egységes keretek közötti alkalmazását és összehasonlítását. A Matlab platformon fejlesztett FRI ToolBox nevet viselő szoftvercsomag a saját három mellett nyolc további szabály-interpolációs módszert implementál. A keretrendszerrel kapcsolatos eredményeket a harmadik téziscsoportban foglaltam össze.

Az új szabály-interpolációs módszerek és az FRI ToolBox kidolgozása után természetes lépésnek tűnt egy vagy több gyakorlati alkalmazási lehetőség keresése. Itt elsősorban olyan feladatokkal találkoztam, ahol a modellezni kívánt jelenség leírása összetartozó be- és kimeneti adatok formájában állt rendelkezésre. A feladat megoldásár elsőként egy olyan eljárást és rá épülő szoftvert fejlesztettem [S25] (ACP – Automatic fuzzy system generation based on fuzzy Clustering and Projection), ami a [H13], [H15], [H66], [H72] és [H81] irodalmakban közzétett eljárások egyes elemeinek és gondolatainak felhasználásával, főképpen a PRE módszerre [H15] támaszkodva készült. Az ACP eljárás fuzzy klaszterezésen alapszik, és hegymászó megközelítésű iteratív algoritmust alkalmaz paraméter-azonosításra.

Mivel a szabály-interpoláción alapuló fuzzy következtetés és a ritka szabálybázisok alkalmazásának egyik lehetséges indoka a rendszer komplexitás csökkentése iránti igény, ezért a továbbiakban olyan automatikus rendszergenerálási módszerek kidolgozását tűztem ki célul, amely a korábban alkalmazott ACP módszernél alacsonyabb szabályszámmal képes azonos vagy jobb teljesítménymutatóval rendelkező fuzzy rendszert előállítani. Ez az igény vezetett egy olyan megoldás gondolatához, amelyben a kezdeti nyers rendszer csak két, a minimális és a maximális kimenethez illeszkedő szabályt tartalmaz, majd egy iteratív

rendszerhangolási folyamat keretében a kívánt teljesítménymutató érték eléréséig fokozatosan újabb szabályokkal bővül.

A fenti gondolatmenet alapján kidolgozott két módszeremet (RBE-DSS és RBE-SI [S22]) valós adatokon is teszteltem, ahol jelentős szabálysám csökkenés mellett (pl. [S10]) sikerült jobb eredményeket elérni a szakirodalomban korábban publikáltaknál [H79]. Az automatikus fuzzy rendszergenerálás céljára kifejlesztett grafikus felületű szoftver kétféle nyers rendszer generálást és hatféle hangolást támogat. A szakirodalomban közzétett más módszerekkel elért eredményekkel való összehasonlíthatóság érdekében a felhasználó a rendszer értékelése során hétfajta teljesítménymutató közül választhat. Az automatikus modell-identifikációval és valós adatok alapján történő fuzzy modellgenerálással kapcsolatos kutatási eredményeimet a negyedik téziscsoportban foglaltam össze.

A további kutatási munkám során elsősorban az automatikus modell-identifikációs módszerek továbbfejlesztésére szeretnék koncentrálni. Egy automatikus paraméter-identifikációt lehetővé tevő osztott alkalmazás kidolgozása jelenleg a tervezés szakszában áll. Emellett nagyon fontosnak tartom a gyakorlati alkalmazások körének bővítését. Az FRI ToolBox továbbfejlesztése is terveim között szerepel.

13. Summary

During my research work related to the preparation of my dissertation I focused on two main topics. They are the fuzzy reasoning based on rule interpolation, and the automatic fuzzy model identification from sample data.

Fuzzy reasoning methods based on rule interpolation exceed the delimitations of the classical compositional approach, being able to produce interpretable conclusion even in cases when the rule base of the system does not contain rules for each possible observation.

My first research goal was the development of new fuzzy reasoning methods that are based on the generalized methodology of fuzzy rule interpolation [H4] and allow the widest possible application area.

The first thesis group deals with the determination of the antecedent and consequent fuzzy sets of the interpolated rule in the generalized methodology of fuzzy rule interpolation. I have developed three new set interpolation methods called FEAT-p [S16], FEAT-LS [S15] and VESI [S2] as well as their software implementation. The FEAT-p method introduces the concept of polar cuts and linguistic term shifting and calculates the shape of the new set taking into consideration all the known sets of a partition. The FEAT-LS method is also based on the concept of linguistic term shifting, but uses α -cuts and the method of least squares for the determination of the new fuzzy sets. VESI transfers the task of interpolation in the so called vague environment.

The second thesis group summarizes my research results related to the single rule reasoning methods applicable in the second step of the generalized methodology of fuzzy rule interpolation for the determination of the conclusion. I have developed three new single rule reasoning methods called SURE-p [S12], SURE-LS [S13] and REVE [S23] as well as their software implementation. The SURE-p method calculates the conclusion by its polar cuts taking into consideration the deviations between the corresponding antecedent and observation polar lengths. SURE-LS determines the necessary modification of the rule consequent by its α -cuts and applies the method of least squares. REVE solves the revision problem in the vague environment by applying the antecedent scaling function ratio on the consequent side as well.

Based on the techniques introduced in the first two thesis groups I have worked out three new fuzzy rule interpolation methods that are called FRIPOC [S12], LESFRI [S13] and VEIN [S26]. FRIPOC uses FEAT-p for set interpolation and SURE-p for single rule reasoning. LESFRI uses FEAT-LS for set interpolation and SURE-LS for single rule reasoning. VEIN uses VESI for set interpolation and REVE for single rule reasoning. All of the three methods calculate the position of the consequent sets by an adapted version of the Shepard interpolation [H63].

Recognizing the difficulties of the practical application, comparison and evaluation of the different rule interpolation based fuzzy reasoning methods, I developed a uniform data structure, framework and implementation, which makes possible the application and comparison of these methods within the same frames. My Fuzzy Rule Interpolation Matlab ToolBox implements eight fuzzy rule interpolation techniques beside my own three methods. The framework related research results are summarized in the third thesis group.

Next to the development of new fuzzy rule interpolation methods it seemed to be a natural pursuance the practical application of the methods using the FRI ToolBox. In order to reach this goal I focused on research and development of methods that automatically generate fuzzy models from sample numerical data. First I have developed a method called Automatic fuzzy

system generation based on fuzzy Clustering and Projection (ACP) [S25] by adapting and enhancing some concepts and part solutions from the techniques [H13], [H15], [H66], [H72] and [H81]. The ACP method is based on fuzzy clustering and adapts an iterative hill climbing parameter optimization approach.

An important motivation of the development of fuzzy rule interpolation techniques is the complexity reduction in fuzzy systems. Therefore after the development of the ACP method I have concentrated my efforts to the research and development of new model identification techniques that ensure the same or better system performance with a lower number of rules. This goal led me to the development of the concept of Rule Base Extension (RBE). Its basic idea is that one creates first two initial rules whose consequent parts match the minimal respective maximal output of the modeled phenomena, and next an iterative tuning algorithm is started. If the amelioration speed of the system performance decreases or even stops in course of the iterations a new rule is created. This incremental rule base extension approach ensures the low number of rules.

Based on the concept of RBE I have developed to methods called RBE-DSS and RBE-SI [S22] as well as their software implementation. RBE-DSS creates the antecedent and consequent sets of the new rules using default set shapes defined for each dimension separately. RBE-SI calculates the shapes of the new sets by set interpolation. Here the applied method should be chosen in accordance with the applied fuzzy reasoning method. I have developed a Matlab ToolBox called RuleMaker that contains the implementation of the RBE-DSS, RBE-SI and ACP methods, and makes possible the automatic model identification from sample data. It also has an easy to use graphical user interface. The methods were tested on synthetic and real world data with good results.

The fourth thesis summarizes my results related to automatic fuzzy model identification methods and fuzzy modeling in practice.

My further research work will concentrate on enhancement of fuzzy model identification methods. Recently I have started the design of a distributed application for this task. Besides, I set a high value on finding further practical application possibilities and the enhancement of the FRI ToolBox.

14. Irodalomjegyzék

14.1. Az értekezés témakörében megjelent saját tudományos közlemények

- [S1] Johanyák Zs. Cs.: Fuzzy halmaz-interpoláció legkisebb négyzetek módszerével, Gép 2006/10, pp. 51-57.
- [S2] Johanyák, Zs. Cs.: Vague Environment Based Set Interpolation, A GAMF Közleményei, Kecskemét, XXI. évfolyam (2006-2007), pp. 33-44.
- [S3] Johanyák, Zs. Cs., Alvarez Gil, R. P. and Kovács, Sz.: Extending the Polar Cut based set interpolation and Revision Methods to the case of Polygonal and Gaussian Shaped Fuzzy Sets, Annals of the Faculty of Engineering Hunedoara, Tome V, Fascicule 3, 2007, pp. 191-198.
- [S4] Johanyák, Zs. Cs., Kovács, Sz.: A brief survey and comparison on various interpolation based fuzzy reasoning methods, in Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest, Hungary, 2005, pp. 323-334.
- [S5] Johanyák, Zs. Cs. and Kovács, Sz.: A brief survey and comparison on various interpolation based fuzzy reasoning methods, in Acta Politechnica Hungarica, Journal of Applied Sciences at Budapest Tech Hungary, Vol. 3(1), 2006, pp. 91-105.
- [S6] Johanyák, Zs. Cs., Kovács, Sz.: A brief survey on fuzzy set interpolation methods, in Proceedings of "Doktoranduszok Fóruma", Miskolc, Hungary, 2006, pp. 72-77.
- [S7] Johanyák, Zs. Cs. and Kovács, Sz.: A fuzzy tagsági függvény megválasztásáról, A GAMF Közleményei, Kecskemét, XIX. évfolyam (2004), pp. 73-84.
- [S8] Johanyák, Zs. Cs. and Kovács, Sz.: Distance based similarity measures of fuzzy sets, in Proceedings of the 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI 2005), Herl'any, Slovakia, 2005, pp. 265-276.
- [S9] Johanyák Zs. Cs. and Kovács Sz.: Fuzzy következtetés sűrű és ritka szabálybázisok esetén, Magyar Tudomány Ünnepe, Bács-Kiskun Megyei Tudományos Fórum, Kecskemét, 2005, ISSN: 1586-846x, pp. 201-206.
- [S10] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy modeling of Petrophysical Properties Prediction Applying RBE-DSS and LESFRI, in Proceedings of the International Symposium on Logistics and Industrial Informatics (Lindi 2007), Wildau, Germany, 2007, pp.87-92.
- [S11] Johanyák, Zs. Cs., Kovács, Sz.: Fuzzy rendszer generálása szabálybázis bővítéssel, AGTEDU 2007, 2007 november 8, Kecskemét, ISSN: 1586-846x, pp. 241-246
- [S12] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy Rule Interpolation Based on Polar Cuts, in Computational Intelligence, Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 499-511.
- [S13] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy Rule Interpolation by the Least Squares Method, in Proceedings of the 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), Budapest, Hungary, 2006, pp. 495-506.

- [S14] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy set approximation based on linguistic term shifting, MicroCad 2006, Section N: Applied Information Engineering, Miskolc, Hungary, 2006, pp. 123-128.
- [S15] Johanyák, Zs. Cs., Kovács, Sz.: Fuzzy Set Approximation by Weighted Least Squares regression, Annals of the Faculty of Engineering Hunedoara 2006, Tome IV, Fascicule 1, pp. 27-34.
- [S16] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy set approximation using polar co-ordinates and linguistic term shifting, 4rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence (SAMI 2006), Herl'any, Slovakia, 2006, pp. 219-227.
- [S17] Johanyák, Zs. Cs. and Kovács, Sz.: Intepolation-based Fuzzy Reasoning - a comparison, MicroCAD 2005, International Scientific Conference, Section N: Applied Information Engineering, Miskolc, Hungary, 2005, pp. 189-194.
- [S18] Johanyák Zs. Cs. and Kovács Sz.: Következtetés fuzzy szabálmódosítással, Fiatal Műszakiak Tudományos Ülésszaka XI., International Scientific Conference, Kolozsvár, 2006, pp. 165-168.
- [S19] Johanyák Zs. Cs. and Kovács Sz.: Polár-vágat alapú fuzzy halmaz-interpoláció, Fiatal Műszakiak Tudományos Ülésszaka XI., International Scientific Conference, Kolozsvár, 2006, pp. 169-172.
- [S20] Johanyák, Zs. Cs. and Kovács, Sz.: Similarity measurement in interpolative fuzzy reasoning, in Proceedings of the 6th International Carpathian Control Conference (ICCC 2005), Lillafüred, Hungary, 2005, Vol. I., pp. 317-322.
- [S21] Johanyák, Zs. Cs. and Kovács, Sz.: Single Rule Reasoning Methods in Fuzzy Rule Interpolation, in Proceeding of the „Doktoranduszok Fóruma”, Miskolc, Hungary, 2005, pp. 75-80.
- [S22] Johanyák, Zs. Cs. and Kovács, Sz.: Sparse Fuzzy System Generation by Rule Base Extension, in Proceedings of the 11th IEEE International Conference of Intelligent Engineering Systems (IEEE INES 2007), Budapest, Hungary, 2007, pp. 99-104.
- [S23] Johanyák, Zs. Cs. and Kovács, Sz.: Survey on three single rule reasoning methods, A GAMF Közleményei, Kecskemét, XXI. évfolyam (2006-2007), pp. 75-86
- [S24] Johanyák, Zs. Cs. and Kovács, Sz.: Survey on various interpolation based fuzzy reasoning methods, in Production Systems and Information Engineering, 2006, Vol. 3, pp. 39-56.
- [S25] Johanyák, Zs. Cs. and Kovács, Sz.: The effect of different fuzzy partition parameterization strategies in gradient descent parameter identification, in Proceedings of the 4th International Symposium on Applied Computational Intelligence and Informatics (IEEE SACI 2007), Timisoara, Romania, 2007, pp. 141-146.
- [S26] Johanyák, Zs. Cs. and Kovács, Sz.: Vague Environment-based Two-step Fuzzy Rule Interpolation Method, in Proceedings of the 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007), Poprad, Slovakia, 2007, pp. 189-200.
- [S27] Johanyák Zs. Cs., Kovács Sz., Tikk D., K. W. Wong: Fuzzy szabály-interpolációt támogató eljárásgyűjtemény fejlesztése Matlab rendszerben, AGTEDU 2006, Kecskemét, 2006, ISSN: 1586-846x, pp. 177-182.

- [S28] Johanyák, Zs. Cs., Parthiban, R, and Sekaran, G.: Fuzzy Modeling for an Anaerobic Tapered Fluidized Bed Reactor, Scientific Bulletin of “Politehnica” University of Timisoara, Romania, Transactions on Automatic Control and Computer Science, 2007, Vol. 52(66), pp.67-72.
- [S29] Johanyák, Zs. Cs., Tikk, D., Kovács, Sz. and Wong, K. W: Fuzzy rule interpolation Matlab toolbox – FRI toolbox, in Proceedings of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), 2006, Vancouver, BC, Canada, pp. 1427-1433.
- [S30] Tikk, D., Johanyák, Zs. Cs., Kovács, Sz. and Wong, K. W.: Overview of Fuzzy Interpolation Techniques in Multidimensional Spaces, 8th International Conference on Fuzzy Set Theory and Applications (FSTA 2006), Liptovský Ján, Slovak Republic, 2006, pp. 104-105.
- [S31] <http://fri.gamf.hu>, Fuzzy szabály-interpolációval foglalkozó webhely, bibliográfiai és cikkgyűjtemény, az FRI ToolBox közzétételi helye.
- [S32] http://www.johanyak.hu/?q=PhD_Kutatas, Az értekezés témakörében megjelent saját tudományos közlemények és a kutatómunka során fejlesztett szoftverek webhelye.

14.2. Hivatkozott irodalom

- [H1] Ali, Y. M. and Zhang, L.: A methodology for fuzzy modeling of engineering systems, Fuzzy Sets and Systems, 118/2001. pp 181-197.
- [H2] Babuska, R.: Construction of fuzzy systems - interplay between precision and transparency, in Proceedings of the European Symposium on Intelligent Techniques (ESIT'00), Aachen, Germany, 2000, pp. 445-452.
- [H3] Baranyi, P., Kóczy, L. T.: A General and Specialised Solid Cutting Method for Fuzzy Rule Interpolation, in Journal BUSEFAL, URA-CNRS, Vol. 66., Toulouse, France, 1996, pp. 13-22.
- [H4] Baranyi, P., Kóczy, L. T. and Gedeon, T. D.: A Generalized Concept for Fuzzy Rule Interpolation, in IEEE Transaction on Fuzzy Systems, ISSN 1063-6706, Vol. 12, No. 6, 2004. pp 820-837.
- [H5] Baranyi, P., Mizik, S., Kóczy, L.T., Gedeon, T. and Nagy, I.: Fuzzy Rule Base Interpolation Based on Semantic Revision, in Proceedings of the IEEE International Conference on System Man and Cybernetics (IEEE SMC'98), San Diego, USA, 1998, pp.1306-1311.
- [H6] Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [H7] Borgulya, I.: Szakértői rendszerek, technikák és alkalmazások, ComputerBooks, Budapest, 1995.
- [H8] Botzheim, J., Cabrita, C., Kóczy, L. T. and Ruano, A. E.: Fuzzy rule extraction by bacterial memetic algorithms, in Proceedings of the 11th World Congress of International Fuzzy Systems Association, IFSA 2005, Beijing, China, July 2005, pp. 1563-1568.
- [H9] Botzheim, J., Hámori, B., and Kóczy, L.T.: Extracting trapezoidal membership functions of a fuzzy rule system by bacterial algorithm, 7th Fuzzy Days, Dortmund 2001, Springer-Verlag, pp. 218-227.

- [H10] Bouchon-Meunier, B., Marsala, C. and Rifqi, M.: Interpolative reasoning based on graduality, in Proceedings of the FUZZ-IEEE'2000, San Antonio, USA, 2000, pp. 483-487.
- [H11] Burkhardt, D.G. and Bonissone, P.P.: Automated fuzzy knowledge base generation and tuning, in Proceedings of the FUZZ-IEEE'92, San Diego, USA, 1992, pp. 179-196.
- [H12] Chiu, S. L.: Fuzzy Model Identification Based on Cluster Estimation, Journal of Intelligent and Fuzzy Systems, ISSN 1064-1246, Vol. 2, 1994, pp. 267-278.
- [H13] Chong, A.: Constructing Sparse and Hierarchical Fuzzy Rulebases, PhD Thesis, Murdoch University, Perth, W.A., 2004.
- [H14] Chong, A., Gedeon, T. D. and Kóczy, L. T.: A Hybrid Approach for Solving the Cluster Validity Problem, in Proceedings of the 14th International Conference on Digital Signal Processing, Santorini, Greece, 2002, pp. 1207-1210.
- [H15] Chong, A., Gedeon, T. D. and Kóczy, L. T.: Projection Based Method for Sparse Fuzzy System Generation, in Proceedings of the 2nd WSEAS International Conference on Scientific Computation and Soft Computing, Crete, Greece, 2002, pp. 321-325.
- [H16] Crain, E. R.: The Log Analysis Handbook Vol. 1: Quantitative Log Analysis Methods, Penn Well Publishing Company, 1986.
- [H17] Davies, D. L. and Bouldin, D. W.: A cluster separation measure, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 1, 1979, pp. 224-227.
- [H18] Ding, L, Shen, Z. and Mukaidono, M.: A new method for approximate reasoning, in Proceedings of the 19th International Symposium on Multiple-Valued Logic, Guangzhou, China, 1989, pp. 179-185.
- [H19] Dubois, D. and Prade, H.: Gradual inference rules in approximate reasoning, in Information Sciences, Vol. 61, 1992, pp. 103-122.
- [H20] Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, in Journal of Cybernetics, Vol. 3, 1973, pp. 32-57
- [H21] Fukuyama, Y. and Sugeno, M.: A new method of choosing the number of clusters for fuzzy c-means method, in Proceedings of the 5th Fuzzy Systems Symposium, 1989, pp. 247-250.
- [H22] Gedeon, T. D. and Kóczy, L. T.: Conservation of fuzziness in rule interpolation, in Proceedings of the Symposium on New Trends in Control of Large Scale Systems, Vol. 1, Herl'any, Slovakia, 1996, pp. 13-19.
- [H23] Hsiao, W.-H., Chen, S.-M., Lee, C.-H.: A new interpolative reasoning method in sparse rule-based systems, in Fuzzy Sets and Systems, Vol. 93, 1998, pp. 17-22.
- [H24] Huang, Z. H. and Shen, Q.: A new interpolative reasoning method based on centre of gravity, in Proceedings of the 12th International Conference on Fuzzy Systems, Vol. 1, 2003, pp. 25-30.
- [H25] Huang, Z. H. and Shen, Q.: Fuzzy interpolation with generalized representative values, in Proceedings of the UK Workshop on Computational Intelligence, 2004, pp. 161-171.

- [H26] Jenei, S.: Interpolation and Extrapolation of Fuzzy Quantities revisited - (I). An Axiomatic Approach, in *Soft Computing*, ISSN: 1432-7643, Vol. 5, 2001, pp. 179-193.
- [H27] Kim, Y, Kim, D.W., Lee, D. and Lee, K.H.: A cluster validation index for GK cluster analysis based on relative degree of sharing, in *Information Sciences*, Vol. 168, 2004, pp. 225–242.
- [H28] Klawonn, F.: Fuzzy Sets and Vague Environments, in *Fuzzy Sets and Systems*, Vol. 66, 1994, pp. 207-221.
- [H29] Klawonn, F. and Kruse, R.: Constructing a fuzzy controller from data, *Fuzzy Sets and Systems*, Vol. 85, Issue 2, 1997, pp. 177-193.
- [H30] Klir, G. J. and Folger, T. A.: *Fuzzy Sets , Uncertainty, and Information*, Prentice-Hall International Inc., Binghamton, 1988.
- [H31] Kondorosi K., László Z., Szirmay-Kalos L.: *Objektum-orientált szoftverfejlesztés!*, ComputerBooks, 2007, Budapest,
<http://www.hik.hu/tankonyvtar/site/books/b10110/index.html>
- [H32] Kosko, B. and Dickerson, J. A.: Function approximation with additive fuzzy systems, in *Theoretical Aspects of Fuzzy Control*, John Wiley & Sons, 1995.
- [H33] Kosko, B.: Optimal fuzzy rules cover extrema, *International Journal of Intelligent Systems*, Vol. 10, 1995, pp. 249 - 255.
- [H34] Kovács, Sz.: Extending the Fuzzy Rule Interpolation "FIVE" by Fuzzy Observation, *Theory and Applications*, Springer Berlin Heidelberg, 2006, pp. 485-497.
- [H35] Kovács, Sz.: Interpolation-based Fuzzy Reasoning as an Application Oriented Approach, *Acta Polytechnica Hungarica*, Vol. 2, No. 1, 2005, ISSN 1785-8860, pp. 93-107.
- [H36] Kovács, Sz. and Kóczy, L.T.: The use of the concept of vague environment in approximate fuzzy reasoning, *Fuzzy Set Theory and Applications*, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, vol.12., pp.169-181, Bratislava, Slovakia, (1997).
- [H37] Kovács, Sz. and Kóczy, L. T.: Application of an approximate fuzzy logic controller in an AGV steering system, path tracking and collision avoidance strategy, *Fuzzy Set Theory and Applications*, in *Tatra Mountains Mathematical Publications*, Mathematical Institute of Slovak Academy of Sciences, Vol. 16, Bratislava, Slovakia, 1999, pp. 456-467.
- [H38] Kóczy, L. T.: Fuzzy if ... then rule models and their transformation into one another, in *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 26, 1996, pp. 621-637.
- [H39] L. T. Kóczy, J. Botzheim, and T. D. Gedeon. Fuzzy models and interpolation. In M. Nikraves, J. Kacprzyk, and L. A. Zadeh, editors, *Forging New Frontiers: Fuzzy Pioneers I & II*, volume 217 of *Studies in Fuzziness and Soft Computing*. Springer, 2007.
- [H40] Kóczy, L. T. and Hirota, K.: Approximate reasoning by linear rule interpolation and general approximation, in *International Journal of Approximative Reasoning*, Vol. 9, 1993, pp. 197–225.

- [H41] Kóczy, L. T. and Hirota, K.: Interpolation and size reduction in fuzzy rule bases, in Technical Report TR 93-94/401, Life Chair of Fuzzy Theory, Tokyo Institute of Technology, Yokohama, 1993, 40 p
- [H42] Kóczy, L.T. and Hirota, K.: Interpolation in structured fuzzy rule bases, in Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, San Francisco, USA, 1993, pp. 402-405.
- [H43] Kóczy, L.T. and Hirota, K.: Modular rule bases in fuzzy control, in Proceedings of the FUZZ-IEEE 93, Aachen, Germany, 1993, pp. 606-610.
- [H44] Kóczy, L. T. and Hirota, K.: Ordering, distance and closeness of fuzzy sets, in Fuzzy Sets and Systems, Vol. 59, 1993, pp. 281-293.
- [H45] Kóczy, L. T. and Hirota, K.: Rule interpolation by α -level sets in fuzzy approximate reasoning, in Journal BUSEFAL, URA-CNRS, Vol. 46, Toulouse, France, 1991, pp. 115-123.
- [H46] Kóczy, L. T. and Hirota, K.: Rule interpolation in approximate reasoning based fuzzy control, in Proceedings of the 4th IFSA World Congress, Brussels, Belgium, 1991, pp. 89-92.
- [H47] Kóczy, L. T. and Hirota, K.: Size reduction by interpolation in fuzzy rule bases, in IEEE Transactions on System, Man and Cybernetics, Vol. 27, 1997, pp. 14-25.
- [H48] Kóczy, L.T., Hirota, K. and Gedeon, T. D.: Fuzzy rule interpolation by the conservation of relative fuzziness, in Journal of Advanced Computational Intelligence, Vol. 4/1, 2000, pp. 95-101.
- [H49] Kóczy, L. T. and Kovács, Sz.: Shape of the fuzzy conclusion generated by linear interpolation in trapezoidal fuzzy rule bases, in Proceedings of the 2nd European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, 1994, pp. 1666-1670.
- [H50] Kóczy, L.T. and Kovács, Sz.: Shape of the fuzzy conclusion generated by linear interpolation of trapezoidal if ... then rules, in Fuzzy Set Theory and its Applications, Tatra Mountains Mathematical Publications, Vol. 6, 1995, pp. 83-93.
- [H51] Kóczy, L. T., Kovács, Sz.: The convexity and piecewise linearity of the fuzzy conclusion generated by linear fuzzy rule interpolation, In J. BUSEFAL 60, Automne, URA-CNRS. Toulouse, France, Univ. Paul Sabatier, 1994, pp. 23-29.
- [H52] Kóczy, L. T., Tikk, D.: Fuzzy rendszerek, Typotex Kft., Budapest, 2000.[H53] Larsen, P. M.: Industrial application of fuzzy logic control, in International Journal of Man Machine Studies, Vol. 12(4), 1980, pp. 3-10.
- [H54] Mamdani, E. H. and Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller, in International Journal of Man Machine Studies, Vol. 7, 1975, pp. 1-13.
- [H55] Mizik, S., Baranyi, P., Korondi, P. and Kóczy, L.T.: Comparison of fuzzy interpolation techniques, 4th Meeting of the Euro Working Group on Fuzzy Sets and 2nd International Conference on Soft and Intelligent Computing (EUROFUSE-SIC'99), 1999, Budapest, pp.544-549.
- [H56] Mizik, S., Szabó, D. and Korondi, P.: Survey on fuzzy interpolation techniques, in Proceedings of the IEEE International Conference on Intelligent Engineering Systems, Poprad, Slovakia, 1999, pp. 587–592.

- [H57] Nawa, N.E. and Furuhashi, T.: Fuzzy System Parameters Discovery by Bacterial Evolutionary Algorithm, in IEEE Transactions on Fuzzy Systems, Vol. 7, 1999, pp. 608-616.
- [H58] Oliveira, J. V.: Toward neurolinguistic modeling: constraint for optimization of membership functions, in Fuzzy Sets and Systems, Vol. 106(3), 1999, pp. 357-380.
- [H59] Retter Gy.: Fuzzy, neurális, genetikus, kaotikus rendszerek (Bevezetés a „lágy számítás” módszereibe), Akadémiai Kiadó, Budapest, 2006.
- [H60] Rhee, N. S. and Oh, K. W.: A validity measure for fuzzy clustering and its use in selecting optimal number of clusters, in Proceedings of the IEEE International Conference on Fuzzy Systems, Barcelona, Spain, Vol. 2, 1996, pp. 1020–1025.
- [H61] Ruspini, E. H.: A new approach to clustering, in Information Control, Vol. 15(1), 1969, pp. 22–32.
- [H62] Shen, Z., Ding, L., Mukaidono, M.: Methods of revision principle, in Proceedings of the 5th IFSA World Congress, Seoul, Korea, 1993, pp. 246–249.
- [H63] Shepard, D.: A two dimensional interpolation function for irregularly spaced data, in Proceedings of the 23rd ACM International Conference, New York, USA, 1968, pp. 517-524.
- [H64] Shi, Y., Mizumoto, M., Wu, Z. Q.: Reasoning conditions on Kóczy's interpolative reasoning method in sparse fuzzy rule bases, in Fuzzy Sets and Systems, Vol. 75, 1995, pp. 63-71.
- [H65] Sugeno, M.: An introductory survey of fuzzy control, in Information Sciences, Vol. 36, 1985, pp. 59-83.
- [H66] Sugeno, M. and Yasukawa, T.: A fuzzy-logic-based approach to qualitative modeling, in IEEE Transactions on Fuzzy Systems, Vol. 1, 1993, pp. 7-31.
- [H67] Takagi, T. and Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control, in IEEE Transactions on System, Man and Cybernetics, Vol. 15, 1985, pp. 116-132.
- [H68] Tikk, D.: Investigation of fuzzy rule interpolation techniques and the universal approximation property of fuzzy controllers, Ph. D. dissertation, TU Budapest, Budapest, 1999.
- [H69] Tikk, D. and Baranyi, P.: Comprehensive analysis of a new fuzzy rule interpolation method, in IEEE Transactions on Fuzzy Systems, Vol. 8, 2000, pp. 281-296.
- [H70] Tikk, D., Baranyi, P., Gedeon, T. D. and Muresan, L.: Generalization of the Rule Interpolation method Resulting Always in Acceptable Conclusion, in Tatra Mountains Mathematical Publications, Mathematical Institute of Slovak Academy of Sciences, 2001, vol. 21, pp. 73-91.
- [H71] Tikk, D., Baranyi, P., Yam, Y. and Kóczy, L. T.: Stability of a new interpolation method, in Proceedings of the IEEE International Conference on System, Man, and Cybernetics (IEEE SMC'99), Tokyo, Japan, 1999, Vol. III, pp. 7-9.
- [H72] Tikk, D., Gedeon, T. D. Kóczy, L. T. and Bíró, G.: Implementation details of problems in Sugeno and Yasukawa's qualitative modeling, Research Working Paper RWP-IT-02-2001, School of Information Technology, Murdoch University, Perth, W.A., 2001.

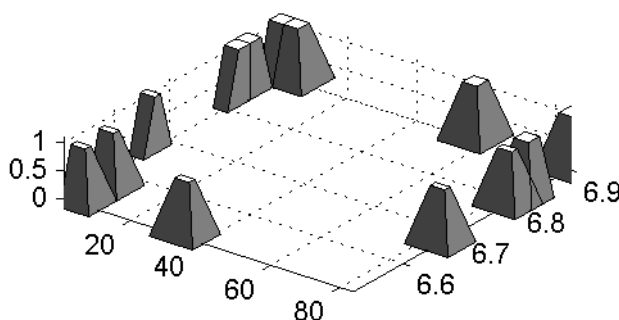
- [H73] Tikk, D., Joó, I., Kóczy, L. T., Várlaki, P., Moser, B. and Gedeon, T. D.: Stability of interpolative fuzzy KH-controllers, in *Fuzzy Sets and Systems*, Vol. 125(1), 2002, pp. 105-119.
- [H74] Turksen, I. B., Zhong, Z.: An approximate analogical reasoning scheme based on similarity measures and interval valued fuzzy sets, in *Fuzzy Sets and Systems*, Vol. 34, 1990, pp. 323–346.
- [H75] Vass, G., Kalmár, L. and Kóczy, L. T.: Extension of the fuzzy rule interpolation method, in *Proceedings of the International Conference on Fuzzy Sets Theory Applications (FSTA '92)*, Liptovsky Mikulas, Czechoslovakia, 1992, pp. 1-6.
- [H76] Wang, W. and Zhang, Y.: On fuzzy cluster validity indices, in *Fuzzy Sets and Systems*, 2007, Vol. 158, Issue 19, pp. 2095-2117.
- [H77] Weisstein, E. W.: Root-Mean-Square, From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Root-Mean-Square.html>
- [H78] Windham, M.P.: Cluster validity for fuzzy clustering algorithms, in *Fuzzy Sets and Systems*, 1981, Vol. 5, pp. 177–185.
- [H79] Wong, K. W. and Gedeon, T. D.: Petrophysical Properties Prediction Using Self-generating Fuzzy Rules Inference System with Modified Alpha-cut Based Fuzzy Interpolation, in *Proceedings of the Seventh International Conference of Neural Information Processing ICONIP 2000*, 2000, Taejon, Korea, pp. 1088 - 1092.
- [H80] Wong, K. W., Gedeon, T. D. and Tikk, D.: An improved multidimensional α -cut based fuzzy interpolation technique, In *Proc. Int. Conf. Artificial Intelligence in Science and Technology (AISAT'2000)*, Hobart, Australia, 2000, pp. 29–32.
- [H81] Wong, K. W., Kóczy, L. T., Gedeon, T. D., Chong, A. and Tikk, D. :Improvement of the Cluster Searching Algorithm in Sugeno and Yasukawa's Qualitative Modeling Approach, in *Lecture Notes in Computer Science*, Vol. 2206, 2001, pp. 536–549.
- [H82] Wong, K. W., Tikk, D., Gedeon, T. and Kóczy, L. T.: Fuzzy Rule Interpolation for Multidimensional Input Spaces With Applications: A Case Study, in *IEEE Transactions on Fuzzy Systems*, Vol. 13, No. 6, 2005, pp. 809-819.
- [H83] Xie, X. L. and Beni, G.: A validity measure for fuzzy clustering, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, 1991, 841-847.
- [H84] Yang, M. S. and Wu, K. L.: A New Validity Index For Fuzzy Clustering, in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, Vol. 1, 2001, pp. 89 - 92.
- [H85] Yam, Y. and Kóczy, L. T.: Representing membership functions as points in high dimensional spaces for fuzzy interpolation and extrapolation, in *IEEE Transactions on Fuzzy Systems*, Vol. 8., Issue 6, 2000, pp. 761-772
- [H86] Yan, S., Mizumoto, M. and Qiao, W. Z.: An Improvement to Kóczy and Hirota's Interpolative Reasoning in Sparse Fuzzy Rule Bases, in *International Journal of Approximate Reasoning*, Vol. 15, 1996, pp. 185-201.
- [H87] Zadeh, L. A.: Fuzzy Sets, in *Information and Control*, Vol. 8, 1965, pp. 338-353.
- [H88] Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes, in *IEEE Transactions on System, Man and Cybernetics*, Vol. 3, 1973, pp. 28-44.

1. Melléklet. A szennyvíztisztítási feladathoz készített kétbemenetes fuzzy modellek identifikációjának eredményei

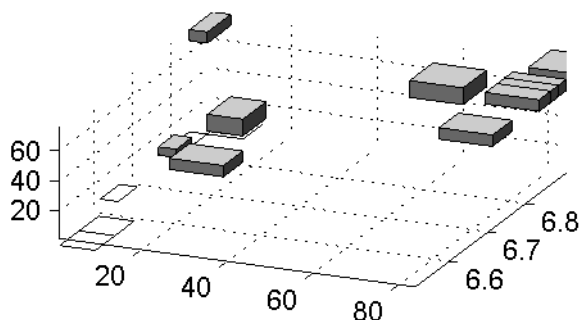
A Sri Venkateswara College of Engineering indiai egyetemmel való együttműködés keretében a feladatom az volt, hogy készítsek fuzzy modellt az általuk készített szennyvíztisztító berendezés működéséhez kapcsolódóan. Ebben az esetben a modell két bemeneti és két kimeneti dimenzióval rendelkezett.

A feladat megoldásaként a két kimeneti változóhoz külön-külön szabályrendszert azonosítottam az RBE-DSS módszer segítségével. Szabály-interpoláción alapuló fuzzy következtetési módszerként LESFRI-t alkalmaztam.

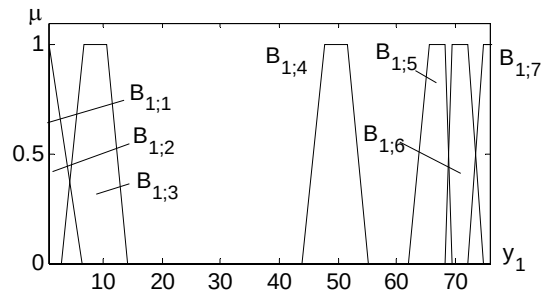
Az első rendszer a kimeneti OLR érték valamint a bemeneti OLR és pH értékek között teremtett kapcsolatot.



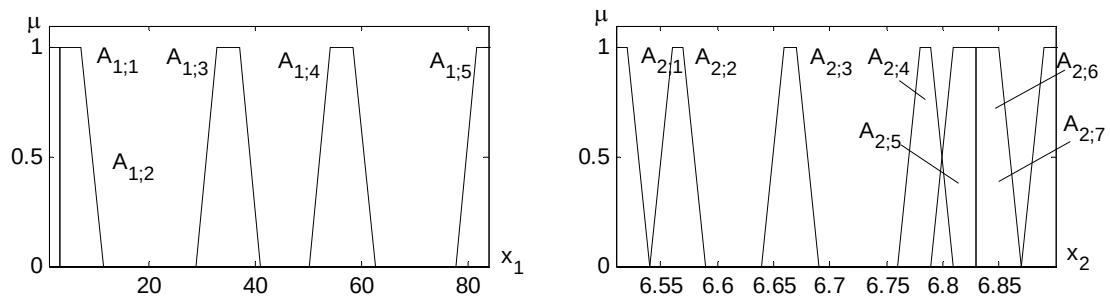
144. ábra. Az első szabálybázis antecedens tere.
Minden szabály antecedensét egy csonka gúla jelképez.



145. ábra. A rendszer szabálybázisa. Minden szabályt egy téglatest
jelképez, amelynek éleit az antecedens és konzekvens halmazok tartói definiálják.

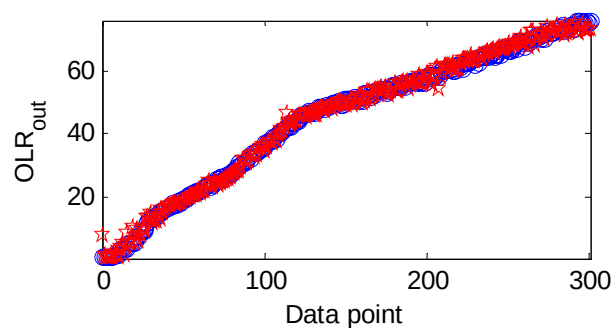


146. ábra. Az első rendszer konzekvens partíciója



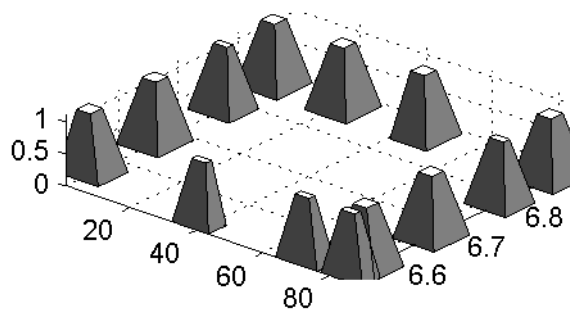
147. ábra. Antecedens partíciók OLR bemeneti (bal oldal) és pH bemeneti (jobb oldal) érték

A rendszert az RMSEP teljesítménymutató alapján hangoltam a $PI_{RMSEP}=1.7899\%$ értékkel.

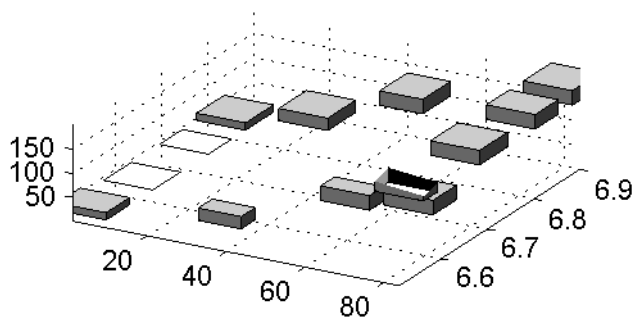


148. ábra. Mért (körök) és számított pontok illeszkedése

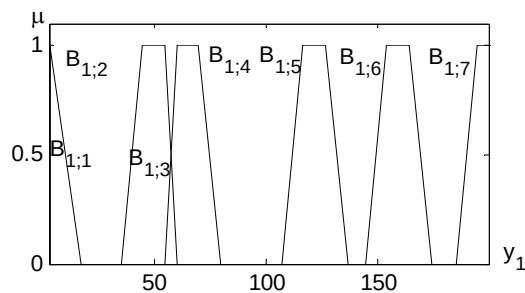
A második rendszer a kimeneti biogáz érték valamint a bemeneti OLR és pH értékek között teremtett kapcsolatot.



149. ábra. A második szabálybázis antecedens tere.
 Minden szabály antecedensét egy csonka gúla jelképez.

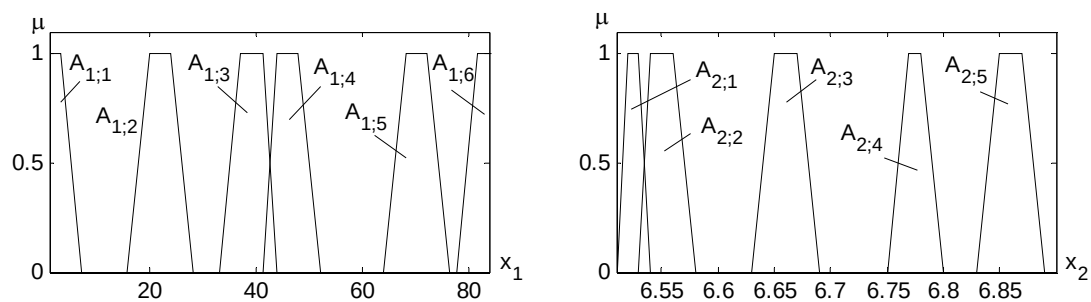


150. ábra. A második rendszer szabálybázisa. Minden szabályt egy téglatest
 jelképez, amelynek éleit az antecedens és konzekvens halmazok tartói definiálják.



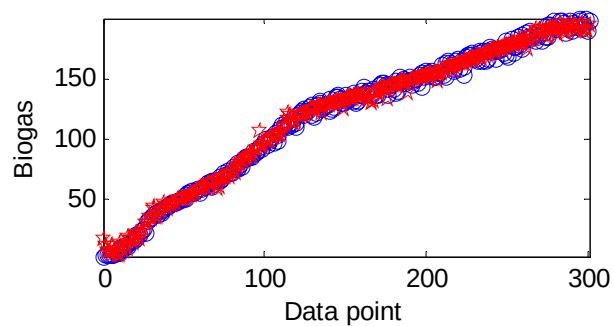
151. ábra. A második rendszer konzekvens partíciója

1. Melléklet. A szennyvíztisztítási feladathoz készített kétbemenetes fuzzy modellek
 identifikációjának eredményei



152. ábra. A második rendszer antecedens partíciói, OLR bemeneti (bal oldal) és pH bemeneti (jobb oldal) érték

A rendszert az RMSEP teljesítménymutató alapján hangoltam a $PI_{RMSEP}=2.0336\%$ értékkel.



153. ábra. Mért (körök) és számított pontok illeszkedése a második rendszer esetén