

Értékünk AZ Ember

Humánerőforrás-fejlesztési Operatív Program



Kóczy T. László
Tikk Domonkos
Botzheim János

Intelligens rendszerek

Készült a HEFOP 3.3.1-P-2004-09-0102/1.0 pályázat támogatásával

Szerzők: Kóczy T. László
Tikk Domonkos
Botzheim János

© Kóczy T. László, Tikk Domonkos, Botzheim János, 2007

Tartalomjegyzék

I. Elméleti alapok	7
1. Bevezetés	9
1.1. A kezdetek	9
1.2. Igények és motivációk	12
1.3. Fuzzy logika és közelítés	14
1.4. Fuzzy vagy hagyományos logika?	20
1.5. A fuzzy tudomány rövid története	23
1.6. E kötet tartalma	26
2. Alapfogalmak	29
2.1. A hagyományos halmazelmélet rövid áttekintése	29
2.2. Fuzzy halmazok alapvető típusai	32
2.3. Fuzzy halmazok jellemzői	38
3. Műveletek fuzzy halmazokon	43
3.1. Fuzzy komplementek	45
3.2. Fuzzy metszetek (t-normák)	51
3.3. Fuzzy uniók (t-konormák, s-normák)	55
3.4. Aggregációs operátorok	59
3.5. I-fuzzy struktúrák	62
4. Fuzzy relációk	66
4.1. Projekció és hengeres kiterjesztés	68
4.2. Bináris fuzzy relációk	71
4.3. Irányított gráfok	74
4.4. Hasonlóság, kompatibilitás, fuzzy rendezések	78
II. Fuzzy irányítási rendszerek és alkalmazásaik	87
5. A fuzzy irányítási rendszerek bevezetése	89

6. Tudásbázis-alapú szakértő rendszerek	97
6.1. Hagyományos irányítási rendszerek	97
6.2. Fuzzy szakértő rendszerek	102
7. Fuzzy irányítási rendszerek	105
7.1. A fuzzy irányítási rendszerek felépítése	105
7.2. A fuzzy irányítási rendszerek alkotóegységei	108
7.2.1. A szabálybázis szerkezete	108
7.2.2. A szabályok ábrázolása fuzzy relációkkal	110
7.2.3. Nyelvi változók és fuzzy halmazok szemantikája	112
7.2.4. Fuzzy partíciók és tulajdonságaik	114
7.3. Mamdani-féle fuzzy irányítási rendszerek	116
7.4. Defuzzifikációs módszerek	121
7.4.1. Súlypont módszer (COG)	122
7.4.2. Geometriai középpont módszer (COA)	123
7.4.3. Maximumok közepe módszer (MOM)	124
7.4.4. Középső maximum módszer (COM)	125
7.5. Függvény kimenetű fuzzy irányítási rendszerek	125
7.6. Fuzzy irányítási rendszerek explicit függvényei	128
7.6.1. Explicit függvények egyenlő szárú háromszög alakú szabályok esetén	128
7.6.2. Explicit függvények trapéz alakú szabályok esetén	131
7.6.3. Az explicit függvények jelentősége	133
7.7. Fuzzy irányítási rendszerek univerzális közelítő tulajdonsága	133
8. Fuzzy redukciós módszerek	137
8.1. Klasszikus fuzzy következtető algoritmusok komplexitása	137
8.1.1. Algoritmusok bonyolultsága	137
8.1.2. Klasszikus algoritmusok bonyolultsága	139
8.2. Csökkentési lehetőségek	140
8.3. Ritka szabálybázisok	141
8.4. Fuzzy szabályinterpoláció	143
8.4.1. A lineáris (KH)-szabályinterpolációs eljárás	144
8.4.2. A lineáris interpolációs eljárás elemzése	147
8.5. Interpolációs módszerek áttekintése	153
8.5.1. VKK-eljárás	153
8.5.2. Szabályinterpoláció testmetszéssel	154
8.5.3. További szabályinterpolációs módszerek	155
8.5.4. A MACI-módszer	156

8.5.5. A MACI-módszer vizsgálata	162
8.6. Hierarchikus szabálybázisok	165
9. Alkalmazások	170
9.1. Egy demonstrációs példa: a fordított inga szabályozása . . .	171
9.2. Vezetőnélküli targonca irányítása	175
9.2.1. A targonca modellje és irányítási stratégiája	176
9.2.2. Irányítás Mamdani-módszerrel	178
9.2.3. Irányítás szabályinterpolációs módszerrel	180
10. Evolúciós algoritmikus módszerek	184
10.1. Genetikus algoritmusok	184
10.1.1. Gyakran használt fogalmak	185
10.1.2. Az algoritmus	186
10.1.3. Az alkalmassági (fitness) függvény	186
10.1.4. Szelekció	187
10.1.5. Keresztezés	188
10.1.6. Mutáció	189
10.1.7. Visszahelyettesítés	189
10.1.8. Migráció	190
10.2. Genetikus programozás	190
10.2.1. Keresztezés	191
10.2.2. Mutáció	192
10.3. Bakteriális evolúciós algoritmusok	193
10.3.1. Bakteriális mutáció	194
10.3.2. Génátadás	195
10.3.3. Összehasonlítás, paraméterek	196
10.4. Egyéb módszerek	196
11. Neurális hálózatok	197
11.1. Többrétegű perceptron	197
11.2. Radiális bázisfüggvény hálózatok	200
11.3. B-spline neurális hálózatok	201
11.3.1. Normalizált bemeneti tér réteg	201
11.3.2. A bázisfüggvények rétege	203
11.3.3. A súlyvektor réteg	203
11.3.4. Almodulok	203
11.3.5. B-spline neurális hálózatok tervezése	204
11.4. Más típusú neurális hálózatok	205

11.5. Neurofuzzy irányítási rendszerek	205
11.6. Backpropagation eljárás	209
11.7. Levenberg-Marquardt algoritmus	210
12. Intelligens számítási modellek identifikációja	213
12.1. Fuzzy szabályoptimalizálás bakteriális evolúciós algoritmussal	213
12.2. Szabályredukciós operátorokkal kiegészített bakteriális evolúciós algoritmus alkalmazása	215
12.2.1. A javasolt algoritmus	216
12.2.2. A módszer tesztelése	222
12.3. A Levenberg-Marquardt algoritmus alkalmazása Mamdani-típusú fuzzy szabálybázis optimalizálására	223
12.3.1. A Jacobi-mátrix meghatározása	224
12.3.2. A módszer alkalmazása	226
12.4. Bakteriális memetikus algoritmus alkalmazása Mamdani-típusú fuzzy szabálybázis optimalizálására	235
12.4.1. A javasolt algoritmus	235
12.4.2. Az algoritmus alkalmazása	237
12.5. Bakteriális programozás alkalmazása B-spline neurális hálózatok identifikációjára	245
12.5.1. A javasolt módszer	245
12.5.2. A módszer alkalmazása	250
Irodalomjegyzék	263
Tárgymutató	282

I. rész

Elméleti alapok

1. fejezet

Bevezetés

1.1. A kezdetek

A következő paradoxont általában HENRI POINCARÉ (1854–1912) francia matematikus és filozófus nevéhez kötik, de annak gyökerei valószínűleg az ókori tudományosságig nyúlnak vissza.

Képzeljünk el egy kupac homokot. A pontos meghatározására vonatkozó „Mi ez?” kérdésre a nyilvánvaló válasz: ez egy homokkupac. Ha elveszünk egyetlen homokszemet a kupacból, annak hiányát nem lehet észrevenni, ezért a válaszuk ugyanaz marad. Ismételjük meg a műveletet még néhányszor. Az eredmény változatlan. Ha az itt leírt kísérletet valamilyen matematikai formalizmussal közelítjük, a következő meglehetősen abszurd egyenletet kapjuk:

$$\text{homokkupac} - 1 \text{ homokszem} = \text{homokkupac}$$

Ez az egyenlet csak akkor lehetne igaz, ha a homokszem a nullával volna egyenértékű. Ez azonban — bármily kicsi is egy homokszem — nem igaz. Sőt az is nyilvánvaló, hogy minden homokkupac véges számú homokszemből áll, tehát az előbbi műveletet véges sokszor megismételve a homokkupacot teljesen eltüntethetjük, azaz nullát kaphatunk. A matematikai teljes indukció módszerét alkalmazva a fentiekből következhetnék, hogy

$$\text{homokkupac} = 0.$$

E nyilvánvaló paradoxon feloldása abban rejlik, hogy a homokkupac fogalmát nem definiáltuk kellő pontossággal. Ahhoz, hogy a homokszemek egy véges halmazát kupacnak nevezzük, legalább egy bizonyos minimális számú homokszemnek kell együtt lennie, és ez a szám nem is kicsi. Ezenkívül természetesen a homokszemek kupacszerű elrendezése is fontos.

Egy egyetemi előadáson az alábbi pontosítást javasolta egy hallgató: „A homokkupac definíciója legyen az, hogy a homokszem halmaz elemszáma legalább négy és az elrendezés legyen tetraéderszerű.” Ezzel a precíz

matematikai definícióval a paradoxon feloldható, mivel az állítás így módosítható: ha egy homokkupac elemszáma legalább 5 és elveszünk belőle egy homokszemet, a fennmaradó rész még mindig homokkupac (feltéve, hogy az elrendezés kupacszerű marad). A probléma az, hogy a definíció egyáltalán nincs összhangban a „homokkupac” hétköznapi fogalmával. Senki sem nevezne egy 4 homokszemből álló kis tetraédert kupacnak, mondjuk egy tengerparti strandon!

A homokkupac-paradoxon megoldása természetesen a „homokkupac” definíciójában rejlik. Nem az a baj, hogy hiányzik a precíz definíció, hiszen az ilyen mindennapi életben használt fogalmak a legritkább esetben írhatók le egzakt matematikai kifejezésekkel; a gond sokkal inkább az, hogy a precíz fogalmakat használó matematikánk nem alkalmas az ilyen pontatlan meghatározások formális kezelésére. Felmerül a kérdés, hogy szükségszerű-e, hogy a matematika csak precíz definíciókat tudjon kezelni?

Nyilvánvaló, hogy vannak olyan homokszemegyüttesek, melyeket mindenki minden körülmények között homokkupacnak tekint, és persze vannak olyan homokszemegyüttesek, amelyeket soha senki. A kettő között vannak „a félig-meddig homokkupacok”. Az olyan homokszemegyüttesek, melyek valamennyire kielégítik a „homokkupacság” feltételeit, de nem teljes mértékben. A megoldás lényege tehát itt van: a homokkupac jellegzetességei fokozatosan tűnnek el, és így vannak olyan helyzetek, amikor az „ez egy homokkupac” állítás nem nevezhető igaznak, de ugyanakkor hamisnak sem, mert csak részben igaz. A *fuzzy logika* részben igaz állításokat is megengedő logika. Az európai, „nyugati” tudományosság a formális logikát már az ókortól kezdve az igaz és a hamis értékpár világába próbálta belekényszeríteni. Ez a gondolkodás már ARISZTOTELÉSnél (görög filozófus, i.e. 384–322) jól megfigyelhető. Az olyan logikai–filozófiai alapelvek, mint az ellentmondás törvénye, vagy a harmadik kizárása ARISZTOTELÉSZig nyúlnak vissza. Ennek értelmében nem lehet valami egyszerre A és $\bar{A}$ (nem A), illetve valamelyiknek a kettő közül igaznak kell lennie. A homokkupac-paradoxon azonban jól szemlélteti, hogy ezek az elvek nem mindig teljesülnek.

Az arisztotelészi logikát a XIX. században G. BOOLE (angol matematikus, 1815–1864) foglalta axiomatikus rendszerbe. A BOOLE-*algebrát*, azaz a kétértékű matematikai logika és a halmazalgebra struktúráját tovább általánosítják az olyan absztrakt algebrák, mint például a háló, melynek részletes vizsgálata G. BIRKHOFF (1911–1996) nevéhez kapcsolódik.

A kétértékű logika és halmazelmélet mellett azonban az ókortól kezdve fel-felmerült a többértékű logika formalizálásának igénye. Kézenfekvő-

nek tűnik például a háromértékű rendszer, amelyben az igaz és hamis értékek mellett megjelenik az eldönthetetlen, vagy eldöntetlen harmadik logikai igazság értéke. (A szokásos szimbólumok: igaz = 1, hamis = 0, eldönthetetlen = $1/2$.) A háromértékű logika sokféleképpen definiálható és számos lehetőség van a logikai alpműveletek általánosítására is, olyan módon, hogy a speciális kétértékű esetben az általánosítás visszaadja az eredeti BOOLE-algebrai struktúrát. Jó példa erre a negáció művelet, amelyet az összes ismert háromértékű logika a $\bar{a} = 1 - a$ függvénnyel definiál, melynek értékei: $\bar{1} = 0$, $\bar{0} = 1$, és $\bar{1/2} = 1/2$. Más alpműveletek, mint a $\wedge$ (metszet, vagy logikai ÉS), $\vee$ (unió, vagy logikai VAGY), $\rightarrow$ (implikáció), és a $\leftrightarrow$ (ekvivalencia) azonban a különböző háromértékű logikákban eltérhetnek egymástól, amint ez az 1.1. táblázatban látható, ahol feltüntettük a logika megalkotójának nevét is.

1.1. táblázat. Az alpműveletek értékei a legismertebb háromértékű logikákban [86]

ŁUKASIEWICZ						BOCHVAR				KLEENE				HEYTING				REICHENBACH			
a	b	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{1}{2}$	0	$\frac{1}{2}$	1	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1
$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$
1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A táblázatból megfigyelhetjük, hogy egyik háromértékű logika sem teljesíti a harmadik kizárása és az ellentmondás törvénye elvét, és a BOOLE-algebrák néhány további tautológiáját sem (vö. 2.1. táblázattal a 31. oldalon), sőt például a BOCHVAR-logika a kétértékű logika egyik alaptulajdonágát sem elégíti ki, ugyanis ez bármely műveletre $1/2$ eredményt ad, ha valamegyik operandus értéke $1/2$.

A háromértékű logika után már könnyű az n -értékű irányba történő általánosítás, mely szintén a XX. század terméke. A legjelentősebb eredmények itt ŁUKASIEWICZ (lengyel matematikus, 1878–1956) nevéhez fűződnek; az n értékű logika igazság értékeit általában $\frac{k}{n-1}$ jelöli, ahol $k = 0, \dots, n-1$.

A nyugaton csak lassan, a XX. századra kibontakozó többértékű logikai megközelítés elemi formában folyamatosan jelen van Keleten, szinte az összes nagy filozófiai irányzatban és vallásban, de különösen a taoizmusban, chan-, vagy zen-buddhizmusban, és az ezekből merítő gondolkodóknál.

A ma népszerű, és Nyugaton is közismert zen-paradoxonok megoldása gyakran a bennük használt fogalmak pontatlan definíciójában és az igazságtartalmuk viszonylagosságában rejlik.

Igen jól szemlélteti a két halmaz bizonytalan határvonalát, az igazság és hamisság egymásbaolvadását az ősi kínai yin-yang szimbólum. E szimbólum egy körön belül mutatja A és $\bar{A}$ egybeolvadását. A fehér és a fekete, a kemény és a lágy, a jó és a rossz, stb. nem éles egyenes határvonal mentén választja ketté az univerzumot jelentő kört, hanem hullámvonal mentén, mely mintegy az ellentétek részleges és fokozatos egymásbanyúlását szimbolizálja. Különösen jól szemlélteti a részleges átlapolást a fehér mezőben felbukkanó kis fekete, és a fekete mezőben felbukkanó kis fehér kör, melyek az ellentétes, komplementes szín (tulajdonság) részleges benyúlását jelentik a másik végletbe. A fentiekben bemutatott egyszerű példák rámutatnak, hogy az emberi gondolkodásban a kezdetek óta jelen van az igény a kétértékű, túlságosan merev logikától való eltérésre, a nem szélsőségekben való gondolkodásra. A következőkben megmutatjuk, hogy az ilyen formalizmusra való igény számos tudományos és alkalmazási területen is felmerül.

1.2. Igények és motivációk

Bármennyire is izgalmas kérdés a homokkupac-paradoxon formális feloldása, az ilyen és hasonló problémák aligha vezettek volna el a fuzzy halmazok és fuzzy logika megalkotásához. Régóta jelen van azonban az igény, hogy azokat a komplex funkciókat, amelyek megvalósítására a legtöbb ember könnyedén képes valamiképpen automatikussá tegyék. A mesterségesen létrehozott ember, a homonculus mondája egészen ősi, de régóta megfigyelhető a törekvés intelligens gépek megalkotására.

Ilyen ambíciók fűtötték a XVIII. századi igen érdekes személyiségű magyar polihisztort KEMPELEN FARKAST (1734–1804), aki beszélő gépről értekezett, és állítólagos sakkozógépet is szerkesztett (mely azonban minden bizonnyal csaláson alapult).

Az intelligens gép megalkotása felé az első komoly lépést NEUMANN JÁNOS (1903–1957) zseniális magyar származású tudós, a modern számítógép megteremtője tette, habár az univerzális számítógép önmagában

természetesen semmilyen intelligenciával nem rendelkezik.

A számítógépek, különösen a félvezető alapú elektronikus számítógépek megléte azonban nagyon erős hajtóerőt jelentett olyan modellek, algoritmusok megalkotására, amelyek az emberi intelligencia valamelyik elemét igyekeznek lemásolni. Az ilyen módszereket együttvéve *Mesterséges Intelligencia* (Artificial Intelligence) néven tárgyalja a szakirodalom.

Miközben az eszközök fejlődése önmagában is megtermékenyítőleg hatott e terület kutatására, mindig igen erős motivációt jelentettek az ún. lágy természettudományok (biológia, orvostudomány, stb.) és a társadalomtudományok (szociológia, közgazdaságtan, stb.), mivel itt eleve rosszul definiált fogalmak és rosszul modellezhető jelenségek képezik a kutatás tárgyát.

Érdekes példa SELYE JÁNOS (magyar származású kutatóorvos, 1907–1982) elemzése a lágy természettudomány kutatási módszertanáról az „Álomtól a felfedezéséig. Egy tudós vallomásai” [157] c. munkájában. Itt világosan leírja, hogy egy biológiai kísérlet eredménye önmagában nem elegendő valamilyen hipotézis bizonyításához vagy cáfolásához, hanem a sokszor újra meg újra megismételt kísérlet többé-kevésbé egymást erősítő eredményei kelljenek ahhoz, hogy a kísérletező tudós a hipotézist elegendő mértékben elfogadja. Figyeljünk fel arra, hogy itt lehetőség van a hipotézis részleges alátámasztására is, ahol korántsem arról van szó, hogy a hipotézis valamilyen valószínűséggel igaz, hanem sokkal inkább arról, hogy esetleg csak részben igaz.

A legerősebb motivációt mégiscsak azok a problémák jelentik, amelyek műszaki területen jelentkeznek. Megdöbbentő, hogy a felnőtt emberek többsége képes megtanulni autót vezetni, de mind a mai napig nem sikerült olyan gépet létrehozni, amely korlátozás nélkül, valóságos forgalmi körülmények között képes egy autó vezetésére. Az utóbbi években megismerhető eredmények, melyek közúti forgalomban résztvevő gépkocsik automatikus irányítására vonatkoznak (pl. CALPATH [65, 129]), csak igen speciális körülmények között, külön védett sávban, kizárólag automatikusan irányított konvojokban való közlekedés esetén érvényesek.

Természetesen hiszünk abban, hogy a teljesen automatikus autó a nem túl távoli jövőben megvalósítható lesz, mégis érdemes elgondolkodni azon, hogy mi a magyarázata e probléma bonyolultságának, a megvalósítás igen nagy nehézségének. A természetes forgalmi viszonyok között közlekedő autó vezetőjének szinte felülről nem korlátos számú különböző információelemet kell feldolgoznia. Ilyenek a közlekedésben résztvevő többi jármű helyzete, nézete, sebessége, iránya, sebességváltozása, stb.; a közlekedé-

si táblák, lámpák, útburkolati jelek, stb. értelmezése; a közelben mozgó emberek helyzete, kora, viselkedése; az útvonal és környezetének topológiája, környezeti tárgyak helyzete, stb.; a közlekedést akadályozó tényezők (úthibák, útjavítások, az úttestre került tárgyak, állatok, stb.). Mindezen tényezők figyelembevétele mellett a gépkocsinak valamilyen kiindulási pontról valamely célpontra kell eljutnia (a lehetőségek szerint minél gyorsabban, minél kisebb üzemanyag-fogyasztással és természetesen az összes korlátozó tényező mindenkor figyelembevételével). Ha jól meggondoljuk, minden egyes autót út egy igen bonyolult komplex optimalizálási feladat megoldását jelenti, melynek során a peremfeltételek nagy száma és időben változó volta nagyfokú adaptivitást és rugalmasságot igényel. Mai tudásunk szerint egy ilyen feladat megoldására csak az ember képes.

Ha valaha megpróbálnánk egy ilyen valóságos közlekedési feladatot számítógépen modellezni, hamarosan fel kellene ismernünk, hogy a probléma matematikai értelemben kezelhetetlen. Hogyan lehetséges akkor mégis, hogy a gyakorlatban az ilyen feladatok elég jól megoldhatók? A megoldás egyszerű: az autót vezető ember olyan mértékben leegyszerűsíti ezt az optimalizálási feladatot, hogy miközben csak közelítő optimumot keres, a feladat mégis kezelhetővé válik. Ennek ára természetesen az, hogy a minél gyorsabb eljutás, a minél kisebb üzemanyagfogyasztás célfüggvényei csak részben optimalizálhatók. A részben leggyorsabb, részben legtakarékosabb (és esetleg a közlekedési szabályokat csak részben megtartó) megoldások óhatatlanul eszünkbe juttatják az előző szakaszban említett részleges igazság kérdését. A legújabb kutatási eredmények azt mutatják, hogy a részleges igazságot megengedő fuzzy logika, és az ezzel rokon formális módszerek alkalmazása lényegesen közelebb visz az ilyen nagybonyolultságú problémák hatékony megoldásához. Igen meggyőző példa erre SUGENO tokiói professzor vezetésével az 1990-es évek eleje óta folyó pilóta nélküli helikopterrel végzett kísérletek sikere [172, 174], ahol éppen a fuzzy logika alkalmazása hozott áttörést.

Összefoglalva megállapíthatjuk, hogy a fuzzy halmazok és fuzzy logika megalkotásában a legdöntőbb motiváló erő kétség kívül a nagybonyolultságú műszaki feladatok megoldásának igénye volt [208].

1.3. Fuzzy logika és közelítés

Az 1950-es évektől kezdve a mesterséges intelligencia kutatása elsősorban a formális szimbolikus logika eszközeit használta. A szakértő rendszerek előszeretettel alkalmaztak ha-akkor típusú és a BOOLE-féle logika

implikációjára vonatkozó következtetési szabályokat. Az implikáció ($\rightarrow$) egyike a fontos BOOLE-algebrai kétváltozós műveleteknek — jelentése: A implikálja B -t, azaz ha A igaz, akkor B is igaz —, amelyet a legelterjedtebb NEM, ÉS, VAGY műveletrendszerben a következő módon lehet kifejezni: $(\bar{A} \vee B)$. A három legelterjedtebb következtetési szabály:

A modus ponens:

$$\frac{A \rightarrow B \quad A}{B}$$

A modus tollens:

$$\frac{A \rightarrow B \quad \bar{B}}{\bar{A}}$$

Végül a *hipotetikus szillogizmus*:

$$\frac{A \rightarrow B \quad B \rightarrow C}{A \rightarrow C}$$

A ha–akkor típusú szabályok implikációként is interpretálhatók. A

$$\text{ha } x = A \text{ akkor } y = B \quad (1.1)$$

szabály (tömören $A(x) \rightarrow B(y)$), egy lehetséges jelentése, hogy ha az x változó az A szimbolikus értéket veszi fel 1 igazságértékkel, akkor az y változó a B értéket veszi fel 1 igazságértékkel. Nézzünk egy egyszerű példát: Egy légkondicionáló berendezés 22°C hőmérsékletű levegőt fúj ki, ha a szoba hőmérséklete meghaladja a 25°C -ot. Itt x a szobahőmérséklet y a légkondicionáló által kifújt levegő hőmérséklete, A a 25°C -nál magasabb hőmérsékleti tartományt jelölő szimbólum, B pedig a kifújt levegő 22°C -os hőmérsékletét jelöli. Hasonló szabályokból felépíthető egy olyan szakértő rendszer, amely a példában szereplő légkondicionálót irányítja. Ha elemezzük az (1.1) szabályra vonatkozó példát, akkor felfigyelhetünk arra, hogy a B szimbólum jelentése túlságosan idealisztikus. Nem valószínű ugyanis, hogy a kifújt levegő hőmérsékletét olyan pontossággal be lehet állítani, hogy az a rendelkezésre álló mérési pontosságon belül megfeleljen a 22°C -nak. Módosítsuk tehát a B jelentését a következőképpen: $22\text{--}23^\circ\text{C}$ közötti hőmérséklet. Ha a példát gondolatban tovább folytatjuk, egy sereg

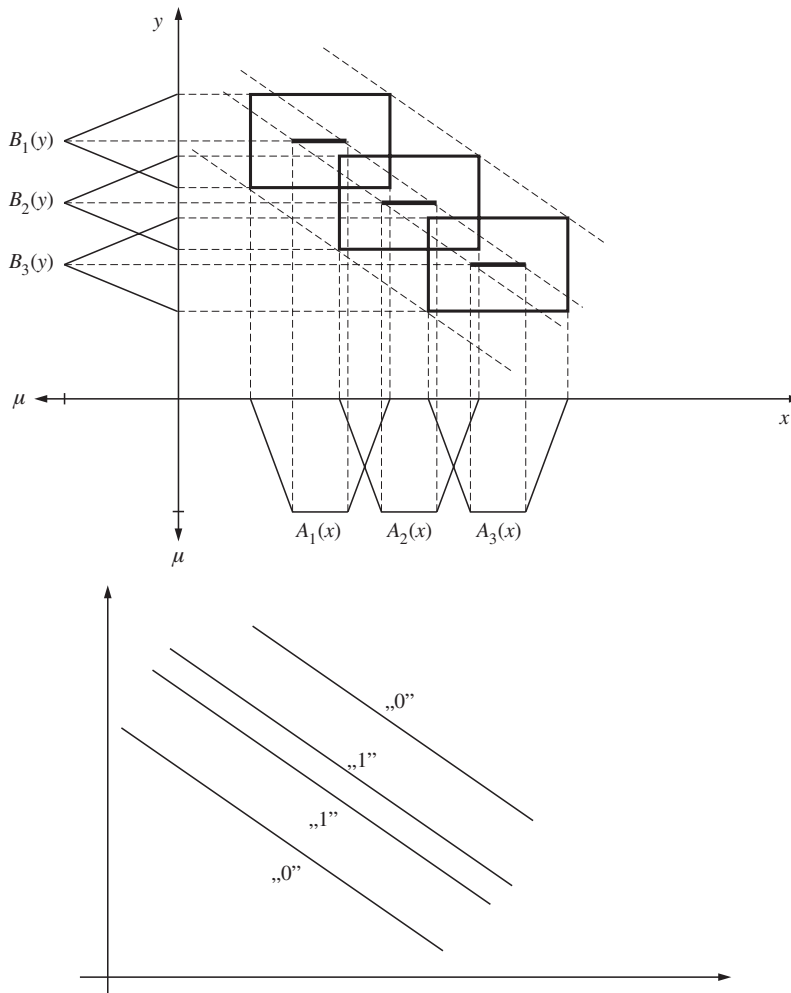
hasonló szabályt konstruálhatunk, melyek mindegyike a szoba hőmérsékletének egy tartományát adja meg kimenetként. Minél pontosabb irányítást akarunk elérni, annál több tartományra kell a szoba jöhető hőmérsékleti intervallumot felosztani. Ezek számával természetesen nő a szabályok száma, valamint arányosan növekszik a szakértő rendszer szabálybázisának mérete is.

Az elmondott példa végletesen leegyszerűsített, de közel áll a gyakorlathoz. Megfigyelhető, hogy a formálisan implikációként kezelt szabályok tulajdonképpen az x és y változók közötti valamilyen hozzárendelést írnak le, mely akár halmazértékű függvényként is felfogható. Az implikációs értelmezés ezért tűnik kedvezőnek, mert formálisan lehetővé teszi a logika következtetési szabályainak alkalmazását.

Ha azonban a szabálybázist függvényszerű érték-hozzárendelésként értelmezzük, akkor az 1.1. ábrán látható közelítő függvényszerű grafikon rajzolódik ki.

Ez nem más, mint egy közönséges $y = f(x)$ függvény közelítő ábrázolása. A közelítés annál pontosabb, minél rövidebbek az érintett intervallumok, melyek határértékben a függvény egy-egy pontjára zsugorodhatnak; ilyenkor a szabálysám természetesen minden határon túl nő. Az ilyen szabálybázison alapuló megközelítés gyenge pontja éppen a szabálysám nem korlátos növekedése. Elviekben kimondható ugyanis az az állítás, hogy egy szimbolikus logikán és ha-akkor szabályokon alapuló szakértő rendszer univerzális közelítő (ld. 7.7. szakasz), a modellben szereplő változók számával azonban a szabálybázis mérete exponenciális gyorsasággal nő. Tegyük fel ugyanis, hogy a bemenet valójában k változót tartalmaz: $x_1, \dots, x_k$, a bemeneti alaphalmaz tehát az $X = X_1 \times \dots \times X_k$ szorzathalmaz. Legyen továbbá T az a küszöbérték, mely az egyes bemeneti változók terében a megkülönböztetett értéktartományok, azaz különböző logikai szimbólumok számának felső korlátját jelzi. Ekkor a szabályhalmaz elemszámának felső korlátja T^k .

Minél finomabb a közelítés, annál nagyobb T értéke és természetesen egy kétszer finomabb felosztás a szabálybázis méretét nem kétszeresére, hanem 2^k -szorosára növeli meg. Ezzel rámutattunk a mesterséges intelligencia modellek legsúlyosabb dilemmájára: minél pontosabb a modell (minél jobb a közelítés), annál magasabb a számítási bonyolultság; minél rövidebb a futásidő, annál rosszabb a közelítés. Úgy tűnik, az ember intelligenciája alkalmas arra, hogy olyan optimális közelítést találjon, ahol a megoldás ideje (az agy „futásideje”) az adott probléma szempontjából még elfogadható (a következtetés valós időben megtörténik), ugyanakkor



1.1. ábra. a) R_1, R_2, R_3 szabálybázis által generált hozzárendelés;
 b) Ezen hozzárendelés („fuzzy függvény”) α -vágatai

a modell pontatlansága nem okoz olyan mértékű tévedést, ami a probléma megoldását megghiúsítaná. A közelítés pontosságának és a megoldási algoritmus matematikai értelemben vett kezelhetőségének ellentmondását a következő egyszerű példán illusztráljuk.

Képzeljünk el egy MI macskát, melynek az a feladata, hogy elfogjon egy egeret. A macska fejében egy szimbolikus szabálybázis van, mely az egér pozícióját, mozgási jellemzőit és minden egyéb szükséges információt figyelembevéve következtet arra, hogy a következő mintavételi pillanatban hol lesz az egér. A macska az egér mozgásterét úgy látja, mint egy raszter-

háló által felosztott síkidomot. A következtetés eredménye a raszterháló egy mezeje; ezen belül a macska a kimerítő keresés módszerével határozza meg az egér tényleges helyzetét. Ha a macska fejében finom modell van, azaz nagyszámú szabály, akkor a következtetés eredménye egy kis méretű rasztermező lesz, és ezért a mező azonosítása után a macska hamar meg fogja találni az egeret. A probléma ilyenkor onnan adódik, hogy a macska fejében lévő finom modell nagy szabályszámot feltételez és ezért a macska következtetési ideje megnő (ez persze visszahat arra is, hogy az egér pillanatnyi helyzete mégiscsak kisebb pontossággal adható meg, hiszen hosszabb idő alatt az egér nagyobb távolságot mozdulhat el). Ha ezzel szemben olyan megoldást választunk, ahol a macska következtetési ideje rövid, ez kis szabályszámot, következtetésképpen pontatlan modellt jelent, vagyis a macska hamar kikövetkezteti az egér új helyzetét jelentő rasztermezőt, de ez a rasztermező nagy kiterjedésű lesz és ezért a keresés második fázisa lesz hosszadalmas.

Vajon van-e optimális kompromisszum? Bebizonyítható, hogyha a macska gondolkodási ideje és a mezőn belüli keresés lépésszáma rögzített költségeket jelentenek, akkor a szabálybázis méretének optimuma számos konkrét modellfajta esetén egyértelműen meghatározható [110, 111]. Az optimum egyszerűbb esetekben analitikusan is, bonyolultabb modell típusoknál azonban csak numerikus technikával található meg, illetve előfordul, hogy az optimum létezésének bizonyítása nem konstruktív.

Analitikus módszerrel meghatározható az optimum tetszőleges bemeneti változószám esetén, ha például a modell egykimenetű és a megfigyelés pontos, azaz crisp halmaz. Most az egyszerűség kedvéért az egyváltozós esetet mutatjuk be. Tegyük fel, hogy a szabályok ekvidisztánsan helyezkednek el, és a tagsági függvények egyenlő szárú háromszögek (azaz legfeljebb 2 szabály tüzel egyszerre). A T_1 következtetési időt

$$T_1 = c_0 r + 2c_1$$

adja meg, ahol c_0 és c_1 alkalmas konstansok, r a szabályok száma. A T_2 keresési idő arányos a konzekvens halmazok tartójának hosszával, ami nyilván fordítottan arányos a szabályok számával:

$$T_2 = 2 \frac{c_2}{r-1},$$

ahol c_2 egy rasztermező keresésének költségtényezője. Az összesített keresési idő tehát

$$\tau = T_1 + T_2 = c_0 r + 2c_1 + 2 \frac{c_2}{r-1},$$

melynek a szabálysámra vonatkozó optimuma deriválással könnyen meghatározható.

Amennyiben pontatlan azaz, fuzzy halmaz a megfigyelés, akkor már egy változó esetén is csak numerikus eljárással adható meg az optimum, több változó esetén pedig csak egzisztenciális eredményt kapunk.

A fentiekben vázolt MI modelltípus gyengéje az volt, hogy a benne szereplő szimbólumok nem tartalmaznak semmilyen információt az eredeti állapottér struktúrájára nézve. A légkondicionáló példájánál maradvá osszuk fel a szobahőmérséklet teljes szóbjajöhető tartományát öt intervallumra (15 °C alatt, 15–20 °C, 20–23 °C, 23–26 °C, 26 °C felett), és jelöljük ezt az öt intervallumot öt különböző szimbólummal ($A_1, \dots, A_5$). Ekkor sem a szimbólumok jelölése, sem egyéb adat nem árulja el, hogy például az A_2 intervallum az A_1 és az A_4 között helyezkedik el, vagy hogy az A_3 közelebb esik az A_4 -hez, mint az A_2 . A szobahőmérsékletek tere ugyanis rendezett, és értelmezhető rajta egy a hőmérsékletek különbségével kifejezhető hőmérséklet-távolság. Összetettebb feladatoknál, ahol több változó szerepel, a rendezés nem tartható meg, de valamely részbenrendezés igen, s megfelelő normalizálás után a távolságfogalom is értelmezhető a többdimenziós állapottérben. A szimbolikus kétértékű logika alkalmazása a rendezés, vagy részbenrendezés és a távolság (metrika) meglétét nem tudja figyelembe venni. A klasszikus MI rendszerek alapvető sikertelenségének magyarázata az, hogy egy elfogadható pontosságú modell esetén a T értéknek már igen magasnak kell lennie. Ekkor azonban a T^k mennyiség értéke miatt gyakorlati problémák kezelésére a modell alkalmatlan.

Mikor L. A. ZADEH 1965-ben bevezette a fuzzy halmaz fogalmát [205], olyan eszközt teremtett, amely lehetővé tette T -nek csökkentését azáltal, hogy a szimbólumokhoz dimenzióként fuzzy tagsági függvény formájában további szubszimbolikus információt rendelt [208], amely a szimbólumok egymáshoz viszonyított helyzetét és távolságát is figyelembe veszi. A fuzzy logika és fuzzy halmazok fogalmainak bevezetése tehát az MI modellekben mind T^k , mind T lényeges csökkentését eredményezte, amint az e könyvben részletesen bemutatásra kerül. Úgy véljük, ez a modellalkotási módszer a természetes emberi gondolkodásnak is sajátja, hiszen az előbbi példában felsorolt öt jól definiált szimbólum helyett sokkal természetesebben hat a következő felosztás: nagyon hűvös, hűvös, kellemes, meleg, nagyon meleg. Ezek a szimbólumok már nem jól definiáltak, jelentésük részben átfed, de éppen emiatt ki is fejezi egymáshoz való viszonyukat. Még érdekesebb, hogy az előbbi légkondicionáló modell kisebb szabályszámmal is megvalósítható. Legyen ugyanis az előző modellben B_1

jelentése 25–26 °C-os levegő, B_2 jelentése 23–24 °C-os levegő, B_3 jelentése: nincs fűtés, B_4 jelentése 22–23 °C-os levegő, B_5 jelentése 20–21 °C-os levegő. A teljes modell szabálybázisa legyen:

$$R_i : \{ \text{Ha } x = A_i \text{ akkor } y = B_i \} \quad i = 1, \dots, 5.$$

Az új fuzzy modellnél elegendő a következő három szabályt használni:

$$\begin{array}{ll} \text{Ha } x = \text{hűvös} & \text{akkor } y = \text{meleg} \\ \text{Ha } x = \text{kellemes} & \text{akkor } y = \text{semmi} \\ \text{Ha } x = \text{meleg} & \text{akkor } y = \text{hűvös} \end{array}$$

A hűvös, meleg stb. szimbólumok megfelelő szubszimbolikus háttere esetén ugyanis a közbenső szabályok közelítő módon kiadódnak. A példa mélyebb megértéséhez szükséges ismereteket a későbbiekben fogjuk tárgyalni.

A fenti példa alapján kimondható a következő: a fuzzy halmazok és logika alkalmazása lehetővé teszi a természetes emberi intelligenciát jobb hatásfokkal másoló, ugyanolyan közelítési pontosság mellett alacsonyabb számítási bonyolultságú modellek, algoritmusok alkalmazását.

1.4. Fuzzy vagy hagyományos logikát követ-e a világ?

Az előzőekben több olyan példát láttunk, ahol a hagyományos (európai, kétértékű) logika alkalmatlannak tűnik a jelenség modellezésére, vagy kezelésére. A homokkupac fogalma nem kezelhető a BOOLE-féle logikával, mert nem határozható meg élesen, hogy hol van a határ a homokkupac és a nem homokkupac között. Az autóvezetés kérdésében több vonatkozásban is felmerül a „fuzzyság” igénye, hiszen a „lehető leggyorsabb”, „lehető legtakarékosabb” feltételek csak közelítő, körülbelüli értelemben vehetők figyelembe, a tényleges autóvezetés során a gáz- vagy fékpedál lenyomásának az erőssége csak hozzávetőlegesen adható meg. Folytatni lehetne tovább a példák sorát, ám ezeknek döntő többségében jól megfigyelhető közös elem a pontatlanságnak, vagy bizonytalanságnak ez a fajtája; a fuzzy típusú pontatlanság valami módon az emberi gondolkodáshoz, vagy emberi cselekvéshez kötődik. A „homokkupac” nem önmagában létező ideál (platóni értelemben), hanem olyan fogalom, amelyet valamilyen természetes emberi nyelven alkottak meg. Azt, hogy az ilyen fuzzy definíciók mennyire kötődnek valamilyen természetes nyelvhez, világosan mutatja

az alapszínek megnevezésének rendszere. Számos ősi nyelvben nincs jelen az egymáshoz eléggé közelálló kék és zöld színek megkülönböztetése, például a japán nyelvben ugyanazt a szót használják az ég színének és a közlekedési lámpa szabad jelzésének megnevezésére (aoki). E szó a modern japánban egyre inkább a kék szín megnevezésére szűkül le, míg a zöldet az új keletkezésű midori jelenti. Egyáltalán nem különböztetik meg a kékét és a zöldet egyes amerikai indián nyelvek sem. Bizonyos elméletek szerint az európai nyelvekben is csak a keresztes háborúk idején szilárdul meg a kék és a zöld megkülönböztetése. Ekkor ugyanis a heraldika tudományának kifejlődésével szükségessé vált a zárt páncélban felismerhetetlen lovagok címerpajzsai alapján történő megkülönböztetése és egyértelmű azonosítása, amelyeknél előfeltétel volt a címer rajz nélküli, egyértelmű szöveges leírhatósága. Mindenesetre érdekes, hogy a ma is használatos ófrancia eredetű heraldikai angolban a kék szín megnevezése azure, azaz égszín (a magyarországi középkori latinban szó szerint „*coelertini coloris*”, azaz égszínű kifejezés szerepel), míg a zöld kifejezése a francia-angolban „vert”, ami a viruló, zöldellő etimológiájára vezethető vissza. Az alapszínek határai tehát emberi megegyezésen alapulnak, amelyek különböző nyelvek esetén máshol húzódnak meg. (Természetesen nem az egyes színek hullámhosszáról van szó, hanem a szubjektív színérzetről.)

A színek kapcsán felvetett kérdéseket ZADEH granulációnak nevezte el, ami tulajdonképpen a diszjunkt elemekre történő particionálás általánosítása [212], hiszen az egyes „granulusok”, azaz megkülönböztetett fogalmak részben átlapolnak. Az egyes nyelvek, sőt az egyes beszélők granulációja eltérő lehet, például egy divattervező összehasonlíthatatlanul több színárnyalatot képes megkülönböztetni, sőt megnevezni, mint ugyanazon nyelvnek átlagos beszélője. Ez a granuláció azonban már meglehetősen diszjunkt és mesterséges, tudományos jellegű partíció.

Természetesen nem állítjuk azt, hogy mindaz, ami természetes emberi fogalmakkal kapcsolatos, az fuzzy. Bizonyos területek (egyes tudományok, a jogalkotás) megkövetelik a szigorúan nem fuzzy definíciót. A fuzzy és hagyományos logikán alapuló fogalmak megkülönböztetésére a következő példát szoktam a bevezető előadáson elmondani: Először azt kérdezem meg a hallgatóktól, hogy ki mennyire jó és magabiztos autóvezető. A határozottan feltett kérdésre, hogy ki tud nagyon jól autót vezetni, néhányan felteszik a kezüket, mások nem, van aki pedig bizonytalan mozdulatot tesz. Amikor azt kérem, hogy ki-ki olyan magasra emelje a kezét, amilyen jól tud vezetni, kialakul a hallgatói csoport alaphalmazán értelmezett „igen jó autóvezetők” fuzzy halmaza; az egészen magasra emelt kezűek teljesen

beletartoznak a halmazba, azaz őrájuk nézve egy mértékben igaz az az állítás, hogy „jó autóvezető”, egyesek egyáltalán nem emelik fel a kezüket, a többieknél pedig a kézfelemelés magassága hozzávetőlegesen kifejezi azt a 0 és 1 közötti mértéket, amennyire ők magukat jó autóvezetőnek érzik. A jó autóvezetők halmaza tehát tipikusan fuzzy halmaz.

Ezután azt a kérdést teszem fel, hogy kinek van vezetői jogosítványa. Erre csak egyértelmű kézfelemeléssel vagy kéz fel nem tevással lehet válaszolni. Itt ugyanis egyáltalán nem fuzzy, hanem hagyományos (crisp) halmazról van szó. Ha egy rendőr igazoltatja az autóvezetőt, az hiába mondja, hogy „már majdnem megvan a jogosítványom, mert holnap fogom megkapni”. Az illető a törvény szerint éppúgy engedély nélküli vezetőnek minősül mint az, aki még el sem kezdte a KRESZ-tanfolyamot. Ezzel szemben az a vezető, aki egy perccel korábban vette át a friss jogosítványt éppúgy teljes joggal vezetheti az autót, mint aki több évtizedes tapasztalattal rendelkezik. (Ez utóbbi nehezen indokolható, és ezért egyre több országban vezetik be a többfokozatú jogosítványt, amely csak több éves gyakorlat után válik teljes értékűvé.)

Újra fel kell tegyük a kérdést, fuzzy vagy hagyományos logikán alapul az objektív világ? Kézenfekvő lenne az a válasz, hogy a fuzzy jellegű bizonytalanságot az emberi intellektus teremtette. Vannak azonban olyan modern elméletek, amelyek szerint a kvantummechanika szintjén a világ tulajdonképpen fuzzy jellegű, és a korábban felállított statisztikus jellegű kvantummechanikai modellek a fuzzy valóságnak csupán pontatlan közelítését adták. E kérdés jelenleg még nem eldöntött.

Érdemes néhány mondatot szólni a fuzzy jellegű 0 és 1 közötti mérték és a valószínűségi mérték kapcsolatára, illetve különbözőségére. A fuzzy elméletet megjelenése idején sok matematikus támadta úgy érvelve, hogy az tulajdonképpen a valószínűségelmélet matematikai struktúrájának új reprezentációja, mely azonban matematikai értelemben nem tekinthető újnak. Ezt az érvelést nem cáfolják azok a megfontolások sem, melyek arra mutatnak rá, hogy a fuzzy bizonytalanság lényegét tekintve más, mint a valószínűségi bizonytalanság, hiszen nem áll mögötte valamilyen statisztikai háttér, ám ettől még tekinthető volna szubjektív valószínűségnek. A későbbi pontos matematikai vizsgálatok azonban megmutatták, hogy a fuzzy mérték axiomatikus tulajdonságai is különböznek a valószínűségi mértéktől, s a legfontosabb, hogy ez utóbbi additivitása helyét a fuzzy mértéket legpontosabban reprezentáló ún. lehetőségi mérték maximum axiómája veszi át. E kérdésre röviden ki fogunk térni a könyvben.

A valószínűségi és lehetőségi mértékek közös tulajdonságainak felis-

merése alapján egyébként létrejött egy sokkal általánosabb mértékelmélet, melyet ma a fuzzy mértékek elméletének neveznek.

1.5. A fuzzy tudomány rövid története

A fuzzy logika közvetlen előzménye ŁUKASIEWICZ [127, 128] többértékű logikája volt, amelyet később megszámlálhatatlan végtelen értékre is általánosítottak. A kontinuum végtelen értékkészletű fuzzy logika, illetve annak halmazelméleti aspektusa L. A. ZADEH berkeley-i professzor ötlete volt, aki már az 1960-as évek elején felvetette rendszerelméleti munkáiban a fuzzy halmazelmélet szükségességét. Az 1965-ben megjelent *Fuzzy Sets* c. tanulmánya [205] végre egyértelműen megfogalmazta a téma alapdefinícióit. ZADEH a rendszerelmélet, illetve az irányításelmélet oldaláról közelítette meg a kérdést, és a kezdetektől fogva világosan rámutatott, hogy az új elmélet jelentősége a nagy bonyolultságú rendszerek közelítő modellezésében rejlik.

A fuzzy halmazelméletet a tudományos közvélemény vegyes reakciókkal fogadta. Sokan a valószínűség-elmélet alternatív megfogalmazásának tekintették, s mint ilyet feleslegesnek ítélték. Ezt a nézetet csak a fuzzy mértékelmélet pontos kidolgozása után sikerült matematikai eszközökkel cáfolni. Egy másik irányzat az arisztotelészi logika tulajdonságait mintegy abszolútnak tekintve, a harmadik kizárása és az ellentmondás törvényének nem teljesülése miatt a fuzzy logikát eleve értelmetlennek minősítette, s ez a nézet meglepő módon egészen az 1990-es évek elejéig tartotta magát. Különösen motiváltak ezen álláspont támogatásában a szimbolikus logikán alapuló mesterséges intelligencia irányzat képviselői. Végül sokan kétségbevonták azt, hogy a téma gyakorlati feladatok megoldására valóban alkalmazható lesz. A különböző irányokból érkező negatív reakciók ellenére 1965-től kezdve exponenciális módon növekedtek a fuzzy témájú publikációk, mind elméleti kutatások terén, mind pedig alkalmazásorientált vizsgálatok eredményeiről beszámolva. ZADEH 1973-ban jelentette meg azt a döntő fontosságú tanulmányát [208], amelyben megmutatta, hogy hogyan lehetséges a ha-akkor típusú szimbolikus szabálybázisok és a fuzzy halmazok szubszimbolikus információját hatékonyan összekapcsolni, s egyben javasolt egy olyan módszert, a kompozíciós következtetési szabályt (CRI), mely alkalmas a fuzzy szabálybázisok, valamint fuzzy, vagy nem fuzzy megfigyelések kombinációjával fuzzy következtetés kiszámítására. A következő évben E. H. MAMDANI londoni professzor ezen módszert átalakította alacsonyabb számítási bonyolultságú, a gyakorlatban jól imple-

mentálható technikává (igaz a szóhajóhető szabálybázisok körét lényegesen leszűkítve), s e módszert igen eredményesen alkalmazta egy nagy bonyolultságú gőzgépes rendszer irányítására [130]. Az első sikeres alkalmazás nyomán hamarosan megszületett az első ipari alkalmazás is (egy dán cementmű irányítása) [76], melyet továbbiak követtek. A MAMDANI-eljárás irányítástechnikai alkalmazásai mellett tovább folyt a kutatás az igen bonyolult problémák megoldásának kérdéseiben. 1975-ben a VAMOS TIBOR által Budapesten szervezett magyar–amerikai *Alakfelismerési szemináriumon* megtartott előadásában ZADEH a lehetséges képfeldolgozási alkalmazásokra mutatott rá. Ezen a téren valóban komoly sikereket értek el főleg az 1980-as évek vége óta. Az említett szeminárium prominens előadói közül egyébként többen fejtettek ki a későbbiekben komoly fuzzy vonatkozású kutatást: például K. S. FU (adaptív rendszerek), A. ROSENFELD (fuzzy geometriai kérdések), R. DE MORI (beszédfelismerés).

Az első időszak lényeges alkalmazási sikereit mégis a CRI-, illetve MAMDANI-módszer jelentette. Az 1984-ben megalakult *Nemzetközi Fuzzy Rendszer Szövetsége (IFSA)* Tokióban, 1987-ben rendezett második világkongresszusán számos japán kutatóiskola mutatta be igen eredményes alkalmazási kísérleteit (elsősorban irányítási területeken, illetve számítógépes látás témájában), sőt a konferencia résztvevői megtekinthették a Sendai városában akkor már működő fuzzy irányítású (vezető nélküli) felsővasutat is. Ugyanakkor Japánban már szennyvíztisztító-rendszerek, alagútszellőzési rendszerek, stb. működtek fuzzy irányítással. 1987 után hamarosan beköszöntött a japán Fuzzy Aranykor. A Sony, Hitachi, Matsushita (Panasonic National), stb. háztartási gépeket és fogyasztói elektronikát gyártó cégek ugyanis sorra dobták piacra a fuzzy logikát alkalmazó energiatakarékos, kezelőbarát, nagyintelligenciájú termékeiket. A legtipikusabb ilyen gépek — melyek ma is igen elterjedtek —, a mosógép, porszívó, légkondicionáló, fürdőszobai víz hőmérséklet szabályozó, rizsfőző, villanyborotva, majd később fényképezőgép és videokamera. Ezek a mindennapi életben sűrűn használt tárgyak olyan népszerűvé tették Japánban a fuzzy logikát, hogy a televízióadók is rendszeresen szerepeltették programjaikban, és szinte minden általános iskolás japán gyerek megismerte e tudomány alapgondolatait.

1989-től a Japán Nemzetközi Kereskedelmi Minisztérium (MIT, mely Japánban komoly szerepet vállal a kutatás finanszírozásában) 50 japán magánvállalattal együtt létrehozta a Nemzetközi Fuzzy Technológiai Laboratórium Alapítványt, mely hat éven át finanszírozta a Yokohamában működő Life kutatólaboratóriumot és a Tokiói Műszaki Egyetemen 1990-ben felállított Fuzzy Elméleti Tanszéket. (Melynek évente változó tanszék-

vezető professzorait az egyetem külföldről hívta meg.) A Life projekt és a hozzákapcsolódó egyetemi kutatások legérdekesebb eredményei a fuzzy szabályalapú pénzügyi előrejelző rendszerek, a már említett vezetőnélküli helikopter, az együttműködő és kommunikáló robotegyüttesek, statikus és dinamikus képfelismerési technikák, stb. voltak. A Life laboratórium tudományos vezetését egyébként a Tokiói Műszaki Egyetem professzora TERANO T. látta el.

A japán sikerek mellett, és részben ezek hatására más távol-keleti országokban is megindult az ipari és háztartási elektronikai berendezésekben való alkalmazás, így Koreában, Tajvanon, stb. Igen érdekes alkalmazási területnek bizonyul a gépjárműtechnika is. Több japán autógyártó vállalat mellett a Life projektben résztvevő Volkswagen cég is megjelent például a fuzzy logikán alapuló automatikus adaptív sebességváltóval.

Érdekes módon az USA-ban, ahonnan az elmélet elindult hosszú ideig jóformán csak az űrkutatás és a haditechnika mutatott komolyabb érdeklődést a fuzzy logika iránt. Kevesek számára ismert, hogy a Sivatagi Vihar háborúban a Patriot rakéták éjszakai célpontazonosító rendszere fuzzy eljárásokon alapult, melyet J. KELLER professzor vezetésével a Missouri Egyetem fejlesztett ki.

Érdekes az a tény is, hogy miközben a gyakorlati alkalmazások súlypontja Európából és részben Észak-Amerikából Kelet-Ázsiába tevődött át, a legkomolyabb fuzzy matematika eredmények döntő többsége Európában született, s itt vannak ma is a leghíresebb fuzzy iskolák. Természetesen ez nem jelenti azt, hogy Európában nincsenek komoly alkalmazási eredmények. Példaként említhetjük a Németországban 1992 óta évente megrendezett Dortmundi Fuzzy Napokat, mely döntően alkalmazási eredményeket vezetett be. Ehhez kapcsolódott a Life mintájára, kisebb tartományi méretekben elindított Észak-Rajna-Westfáliai Fuzzy Iniciatíva, melynek keretében létrejött — elsősorban műszaki és döntéstámogatási alkalmazásokra — a ma már komoly nyereséggel működő Dortmundi Fuzzy Demonstrációs Centrum is, és igen komoly iskolája van az aacheni Észak-Rajna Westfáliai Egyetemen.

Sikeres alkalmazásoknak egy egészen más területe az orvosi biológia, ahol a gyakorlatban is léteznek már fuzzy elven működő, például az altatás vagy a dialízis irányítását végző, valamint diagnosztikai döntéstámogató rendszerek.

Fontos területet jelentenek a pénzügyi alkalmazások: biztosítási kockázatfelmérésben, portfólióválasztásban, illetve pénzügyi előrejelző-rendszerekben alkalmaznak fuzzy technikát.

A sikeres alkalmazási területek sorát még folytathatnánk, ehelyett azonban arra utalunk röviden, hogy a fuzzy logikát közvetve más szubszimbolikus mesterséges intelligens módszerek is megjelentek, elsősorban a mesterséges neurális hálózatok, az evolúciós programok, genetikus algoritmusok, kaotikus rendszerek, stb., mely területek gyakran kombinálódnak is és együttesen a lágy számítástudomány (Soft Computing) megnevezés alatt ismertek.

Ha ma valaki beszél egy japán áruház háztartási gépek osztályára általában legalább három, négy különféle „neurofuzzy” felíratú hibrid rizsfőző, mosógép stb. közül válogathat, azaz mára a fuzzy és rokon modellek alkalmazása mindennapivá vált.

TERANO professzor az 1990-es évek elején négy fázisba osztotta a fuzzy elmélet alkalmazásait. Az első három: az egyszerű fuzzy tudásbázisú rendszerek (például irányítási rendszerek), a bonyolult fuzzy tudásbázisú rendszerek (például nem műszaki szakértő rendszerek), a fuzzy kommunikációt alkalmazó rendszerek (például intelligens kooperatív robotegyüttesek), melyek mindegyike ma számos területen megvalósult, alkalmazásra került, vagy az alkalmazás küszöbén áll. A negyedik fázis a komplex integrált intelligencia, mely ma még „a jövő története”, vagy ha úgy tetszik inkább a sci-fi témakörébe tartozik.

1.6. E kötet tartalma

Az Olvasó az első magyar fuzzy tankönyvet tartja a kezében, melynek anyaga többé-kevésbé követi az 1992 óta KÓCZY T. LÁSZLÓ és tanítványai által a Budapesti Műszaki Egyetemen tartott Fuzzy Rendszerek I. és II. választható, illetve doktori tárgyak tematikáját.

Egy kissé bővebb, de kevésbé egységes tárgyalásmódú, angol nyelvű változata 1996-ban az Európai Unió MODIFY TEMPUS projekt keretében készült el, amelyet ma mintegy 15 európai egyetemen használnak valamilyen formában [93]. Az említett előadónak a Budapesti Műszaki Egyetemen, valamint a koreai Pohangi Műszaki Egyetemen, a Tokiói Műszaki Egyetemen, a Linzi J. Kepler Egyetemen, és az olaszországi Trento Tudományos Egyetemen különböző érdeklődésű hallgatóknak tartott előadások tapasztalatait felhasználva ez a könyv a fuzzy elmélet alapjait tárgyalja olyan mélységig, hogy elegendő ismeretet nyújtson azok számára, akik a gyakorlati alkalmazások iránt érdeklődnek. Ezt követően a bonyolult rendszerek modellezésével és irányításával kapcsolatos algoritmikus kérdések kerülnek részletesebben tárgyalásra, amely magában foglalja a

Budapesti Műszaki Egyetemen e téren az utóbbi tíz évben elért fontosabb eredményeket is.

A könyv két részből áll. Az első rész a fuzzy logikai alapismereteket és a szükséges matematikai háttérrel foglalja össze. A jelen bevezetést követően a 2. fejezet a fuzzy halmazokkal kapcsolatos alapvető fogalmakat és definíciókat tárgyalja, valamint rövid összefoglalást ad a hagyományos kétértékű logika alapfogalmairól és műveleteiről, amire a könyv későbbi fejezeteiben a megfelelő fuzzy műveletekkel való összehasonlítások alkalmával többször is támaszkodunk. A fejezetben a fuzzy halmazok általánosítási lehetőségeit is bemutatjuk.

A 3. fejezet a fuzzy halmazokon értelmezett alpműveleteket (negáció, metszet, egyesítés), és ezek axiómáit és tulajdonságait tárgyalja. Rövid áttekintést nyújt a fuzzy aggregációs műveletekről, majd részletesebben foglalkozik a standard műveletek DE MORGAN-algebrájának egyik lehetséges alternatívájával, az algebrai operátorokon alapuló I-fuzzy struktúrákkal.

A 4. fejezet a fuzzy relációkat ismerteti. Először röviden áttekintést ad a reláció fogalmáról és a bináris relációkról, majd a fuzzy bináris relációkat és tulajdonságaikat ismerteti. A 4.4. szakaszban a hagyományos és fuzzy relációk osztályozása, valamint ezen relációtípusok összehasonlítása található (ekvivalencia, hasonlósági és rendezési relációk).

A második rész a fuzzy tudományág legfontosabb gyakorlati alkalmazásainak, a fuzzy irányítási rendszereknek elméleti háttérét ismerteti és néhány egyszerű példával szemlélteti azokat. Bemutatja továbbá a lágy számítástudomány másik két fő területét is.

Az 5. fejezetben áttekintést adunk a fuzzy irányítási rendszerekről, melyeket a következő fejezetekben részletesen tárgyalunk.

A 6. fejezet a tudásalapú szakértői rendszerek témakörét elemzi az irányítás lehetőségeinek és megvalósításainak szempontjából.

A 7. fejezet tárgyalja a fuzzy információ, tudásbázis és nyelvi változók reprezentálásának módját, amely alapján egy fuzzy irányítási rendszer felépül. Ezután bemutatásra kerül a leggyakrabban használt fuzzy irányítók modellje, alkalmazási lehetőségük, modell-leíró képességük és korlátjaik. A könyv kitér a fuzzy modellek explicit függvényeinek, és az univerzális approximációs tulajdonságának tárgyalására is.

A 8. fejezet az előzőekben tárgyalt algoritmusok bonyolultságát vizsgálja. Mivel ez exponenciálisan nő az alkalmazott szabályok számával, ezért nagy rendszerek esetén szükség van a korábbi módszerek módosítására, és olyan alternatív lehetőségek kidolgozására, melyek csökkentik a bonyolultságot. A fejezet gerincét egy ilyen eljárás, a fuzzy szabályinterpoláció

és a hozzákapcsolódó elméleti háttér tárgyalása teszi ki. A fejezet végén a hierarchikus szabálybázisok alapfogalmai és alkalmazásai találhatók.

A 9. fejezetben alkalmazási példákon keresztül mutatjuk be az egyszerű szabályalapú következtető algoritmusok és a szabályinterpolációs módszerek működését. A fejezet bevezetőjében összefoglalást adunk az eddigi jellemző ipari és kereskedelmi felhasználásokról.

A 10. fejezetben a lágy számítástudomány másik nagy területét, az evolúciós módszereket mutatjuk be. Részletesen tárgyaljuk a genetikus algoritmusokat, a genetikus programozást és a bakteriális evolúciós algoritmusokat. Röviden utalunk egyéb evolúciós technikákra is.

A 11. fejezetben a lágy számítástudomány harmadik nagy csoportját, a mesterséges neurális hálózatok ismertetjük. Röviden bemutatjuk a két legelterjedtebb neurális hálózattípust, az MLP és az RBF hálózatokat, majd kicsit részletesebben ismertetjük a B-spline neurális hálózatokat. A hálózatok bemutatása után a tanuló algoritmusokat tárgyaljuk. A klasszikus back-propagation módszer ismertetése után bemutatjuk a Levenberg-Marquardt tanuló algoritmust.

A 12. fejezetben számítási modellek identifikációjával foglalkozunk. Fuzzy rendszerek és B-spline típusú neurális hálózatok identifikációját tárgyaljuk evolúciós és gradiens alapú módszerek alkalmazásával.

2. fejezet

Alapfogalmak

2.1. A hagyományos halmazelmélet rövid áttekintése

A fuzzy halmazok ismertetése előtt először tekintsük át a (hagyományos) halmazelmélet néhány alapvető fogalmát és azok tulajdonságait. A fuzzy halmazoktól való megkülönböztetés céljából a hagyományos, nem fuzzy halmazokra az irodalomban elterjedt *crisp halmaz* (éles, határozott körvonalú) terminológiát használjuk.

A továbbiakban feltesszük, hogy az Olvasó tájékozott a hagyományos halmazelmélet alapfogalmait illetően, ezért ezen szakasz célja csupán a fogalmak felidézése, és a későbbiekben, a fuzzy halmazok tárgyalása során is használt kifejezések és jelölések bevezetése.

A halmazok jelölésére az ábécé nagybetűit használjuk. Ha másképp kifejezetten nem állítjuk, akkor az *alaphalmaz* — amely az adott kontextusban a lehetséges összes elemet tartalmazza — X -szel jelöljük. Az egyetlen elemet sem tartalmazó, ún. *üres halmaz*ra a szokásos $\emptyset$ jelölést használjuk.

Egy tetszőleges crisp halmaz az alábbi három módon adható meg. Ha a halmaz *véges*, akkor elemei felsorolásával (pl. $A = \{1, 2, 4, 8, 16\}$), tetszőleges számosságú halmazt általában az elemeire teljesülő szabály segítségével (pl. $B = \{x \in X \mid x = 2^n, n \text{ egész}\}$, vagyis azon x értékek melyre teljesül a $|$ jelet követő feltétel), vagy a halmaz *karakterisztikus függvényével* definiálható. A χ_A karakterisztikus függvény kizárólag azon alaphalmazbeli értékekre vesz fel 1 értéket, melyek az A halmaznak elemei, azaz

$$\chi_A(x) = \begin{cases} 1, & \text{ha } x \in A \\ 0, & \text{ha } x \notin A \end{cases}$$

Ha A halmaz minden eleme B halmaznak is eleme, akkor A a B *rész-halmaza*, amit $A \subset B$ -vel vagy $A \subseteq B$ -vel jelölünk, ez utóbbi esetben kihangsúlyozva azt, hogy egyenlőség is megengedett. Minden halmaz rész-halmaza önmagának és az alaphalmaznak.

Ha $A \subseteq B$ és $B \subseteq A$, akkor a két halmaz azonos: $A = B$. Ellenkező esetben $A \neq B$.

Ha $A \subseteq B$ és $A \neq B$, akkor B -nek létezik legalább egy olyan eleme, amely nem eleme A -nak. Ekkor A valódi részhalmaza B -nek, jelölése: $A \subsetneq B$.

Egy A halmaz összes részhalmazának halmazát, $\mathcal{P}(A)$ -t, az A hatványhalmazának hívjuk.

A véges A halmaz elemeinek számát (számosságát) $|A|$ jelöli. Ha A véges, akkor $|\mathcal{P}(A)| = 2^{|A|}$.

Az A halmaz komplemente, $\overline{A}$, az alaphalmaz A -ban nem szereplő elemeit tartalmazza. A komplement képzés legfontosabb tulajdonságait a 2.1. táblázat tartalmazza.

A és B halmazok egyesítése, másszóval uniója, $A \cup B$, azon elemeket tartalmazza, melyek legalább vagy az A vagy a B halmaznak eleme (természetesen mindkettőnek is lehet eleme egyidejűleg):

$$A \cup B = \{x | x \in A \text{ vagy } x \in B\}.$$

Az unió művelete tetszőleges számú argumentumra általánosítható:

$$\bigcup_{i \in I} A_i = \{x | x \in A_i \text{ valamely } i \in I\text{-re}\},$$

ahol $\{A_i | i \in I\}$ egy halmazcsalád, I pedig egy tetszőleges indexhalmaz.

A és B halmazok metszete, $A \cap B$, azon elemeket tartalmazza, melyek mind az A , mind a B halmaznak elemei:

$$A \cap B = \{x | x \in A \text{ és } x \in B\}.$$

A metszet művelete is általánosítható tetszőleges számú argumentumra:

$$\bigcap_{i \in I} A_i = \{x | x \in A_i \text{ minden } i \in I\text{-re}\} \quad (\{A_i | i \in I\}).$$

Az egyesítés és a metszet műveletekre, valamint ezeknek a komplement-sel való kapcsolatára vonatkozó tulajdonságokat a 2.1. táblázat ismerteti. Ezen műveletek tulajdonságai a táblázatban páronként szerepelnek. Vegyük észre, hogy e párok tagjai az $\cup, \cap, \emptyset, X$ jelek rendre $\cap, \cup, X, \emptyset$ jelekre történő cserélésével egymásba alakíthatók. Ezt a tulajdonságot a metszet és az unió dualitásának nevezzük. Mint látható, a duális műveletpárok egyikére vonatkozó tetszőleges állításból a fenti cserék végrehajtásával megkapjuk az állítás duálisát.

Az alaphalmaz hatványhalmazának ($\mathcal{P}(X)$) elemein a részhalmaz művelet egy részben rendezést valósít meg, ezért $\mathcal{P}(X)$ -n egy háló definiálható,

amelyben a *legkisebb felső korlát* az unió, a *legnagyobb alsó korlát* pedig a metszet művelete. A $\langle \mathcal{P}(X), \cup, \cap \rangle$ hálót, amely disztributív (ld. 2.1. táblázat) és komplementumos (hiszen minden $A \in \mathcal{P}(X)$ -nek létezik komplemente $\mathcal{P}(X)$ -ben), *BOOLE-hálónak* vagy *BOOLE-algebrának* nevezzük.

Ha A és B halmazoknak nincs közös elemük, azaz $A \cap B = \emptyset$, akkor *diszjunktak*. Valamely A halmaz páronként diszjunkt, nem üres részhalmazainak családját az A egy partíciójának hívjuk, amennyiben ezen részhalmazok uniója A -val egyenlő:

$$\pi(A) = \{A_i | i \in I, A_i \subset A, A_i \neq \emptyset, \text{ és } \forall i, j \in I, i \neq j : A_i \cap A_j = \emptyset\}.$$

Az A és a B halmaz *DESCARTES-szorzata*, $A \times B$, olyan rendezett párokat tartalmazó halmaz, ahol az első elem az A , a második elem a B halmaznak eleme, azaz:

$$A \times B = \{\langle a, b \rangle | a \in A, b \in B\}.$$

2.1. táblázat. Halmazműveletek alaptulajdonságai

Involúció (kettős negáció törvénye):	$\overline{\overline{A}} = A$
Kommutativitás:	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Asszociativitás:	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Disztributivitás:	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Idempotencia:	$A \cup A = A$ $A \cap A = A$
Elnyelési törvények:	$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$
Elnyelési törvények (X és $\emptyset$):	$A \cup X = X$ $A \cap \emptyset = \emptyset$
Identitás:	$A \cup \emptyset = A$ $A \cap X = A$
Az ellentmondás törvénye:	$A \cap \overline{A} = \emptyset$
A harmadik kizárásának törvénye:	$A \cup \overline{A} = X$
DE MORGAN-azonosságok:	$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$

Ha $A \neq B$ és egyik halmaz sem üres, akkor $A \times B \neq B \times A$.

A DESCARTES-szorzat tetszőleges számú argumentumra általánosítható:

$$A_1 \times A_2 \times \cdots \times A_n = \bigtimes_{1 \leq i \leq n} A_i = \{ \langle a_1, a_2, \dots, a_n \rangle \mid a_i \in A_i \text{ minden } i = 1, 2, \dots, n\text{-re} \},$$

ahol $\{A_1, A_2, \dots, A_n\}$ valamely halmazcsalád. A többdimenziós alaphalmazt általában $X = X_1 \times X_2 \times \cdots \times X_n$ alakban feltételezzük. A DESCARTES-szorzatok részhalmazai a *relációk*, melyekkel részletesen a 4. fejezetben foglalkozunk.

2.2. Fuzzy halmazok alapvető típusai

Amint az előző fejezetben felidéztük, a crisp halmazok karakterisztikus függvénye minden alaphalmazbeli elemhez 0-t vagy 1-et rendel hozzá. A karakterisztikus függvény fogalmát úgy általánosíthatjuk, hogy az alaphalmaz minden eleméhez valamely rögzített tartományból — ez általában a $[0,1]$ intervallum — rendelhető érték. Ezen érték nagysága a halmazbeli tagság mértékével arányos, azaz minél kisebb (nagyobb) mértékben tagja a halmaznak valamely elem, annál kisebb (nagyobb) az elemre vonatkozó függvényérték. Ezt a függvényt *tagsági függvénynek*, az általa definiált halmazt pedig *fuzzy halmaznak* nevezzük.

Tehát a tagsági függvény valamely crisp alaphalmaz minden eleméhez az értékkészletéből egy *tagsági értéket* rendel. (A fuzzy halmazok alaphalmazára az irodalomban gyakran az *univerzum* kifejezést használják.) Ha mást kifejezetten nem állítunk, akkor a továbbiakban a tagsági függvény értékkészletének a $[0,1]$ intervallumot tekintjük.

Mivel a tagsági függvény egyértelműen meghatározza az általa definiált fuzzy halmazt, vagyis valamely fuzzy halmaz és tagsági függvénye között egy-egy megfeleltetés vonható, ezért a jelölésükre használt szimbólumok felcserélhetők. Az irodalomban a

$$\mu_A : X \rightarrow [0,1], \quad \text{illetve} \quad A : X \rightarrow [0,1] \quad (2.1)$$

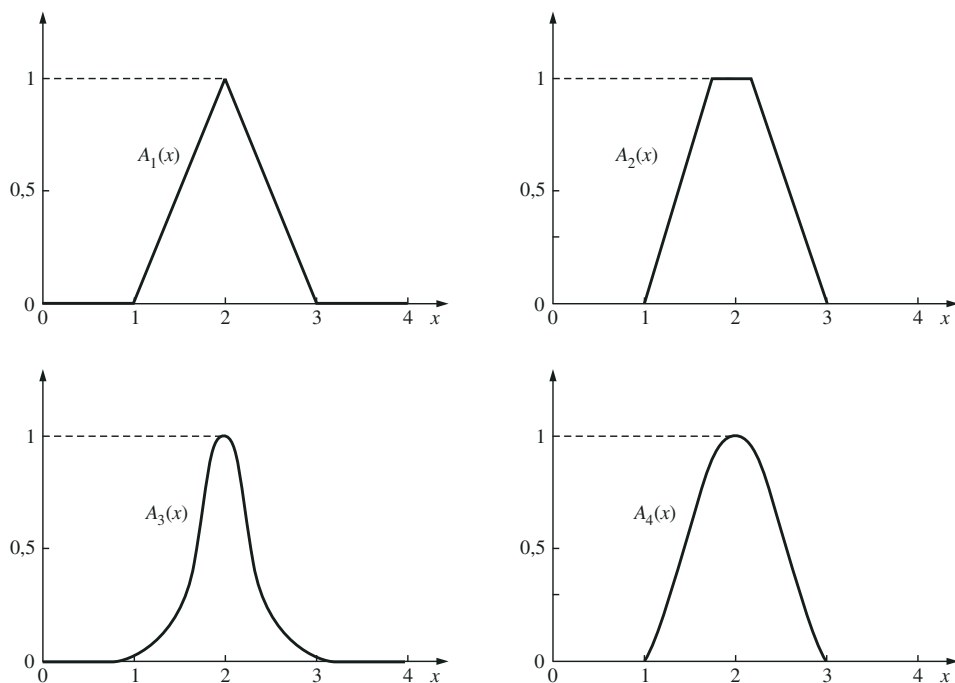
írásmód egyaránt használatos. E könyvben, az egyszerűbb, második jelölésmódot alkalmazzuk.

A bevezetőben láttuk, hogy a fuzzy halmazok alkalmasak a bizonytalan határokkal rendelkező természetes nyelvi fogalmak reprezentálására. Ez a

reprezentáció kontextusfüggő, hiszen nyilván lényegesen különböző fuzzy halmazokkal írható le például a *magas* fogalom, ha az emberek vagy épületek alaphalmazán értelmezzük. Sőt hasonlóan — bár kisebb mértékben — különbözhetnek az e fogalmat leíró fuzzy halmazok a (köztudomásúan kismövésű) pigmeusok és az (általában magasnövésű) svédek között.

Valamely fogalomnak egy rögzített kontextusban is több különböző modellezése lehetséges. A 2.1. ábrán látható fuzzy halmazok mindegyike a „körülbelül 2” koncepciót valósítja meg. Bár a halmazok közt lényeges különbségek vannak, általában igaz rájuk, és a példák is ezt illusztrálják, hogy:

1. A_i tengelyesen szimmetrikus 2-re nézve, azaz $A_i(2 - x) = A_i(2 + x)$ minden valós számra. (Ez egyébként nem szükségszerűen van így; értelmezhető például olyan „körülbelül 2” halmaz, amelyik a 2-nél nagyobb értékek felé „elnyúltabb” tagsági függvénnel rendelkezik.)
2. $A_i(x)$ monoton csökken a $|2 - x|$ különbség növekedésével.
3. $A_i(2) = 1$, és A_2 -t kivéve $A_i(x) < 1$ ha $x \neq 2$. (A_2 esetében a szélesebb



2.1. ábra. A „körülbelül 2” fogalmat reprezentáló különböző alakú fuzzy halmazok

1 tagsági függvényű tartomány modellezheti például a mérőeszköz kiküszöbölhetetlen hibáját.)

4. Az $[1,3]$ intervallumon kívül a tagsági függvények értéke elhanyagolható vagy 0. (Természetesen e határok is választhatók volnának más módon is, például nem teljesen szimmetrikus módon.)

Ezek a tulajdonságok a modellezett fogalom reprezentálásához általában szükségesek, ezért ezeket az olyan halmazoknak teljesítenie kell, mely a „körülbelül 2” fogalmat írja le.

Noha azonos fogalmat modelleznek, a 2.1. ábra fuzzy halmazai jelentősen különböző alakkal rendelkeznek. Az alkalmazások a fuzzy halmazok alakjára általában nem túl érzékenyek, azonban mindig az adott modelltől függ, hogy valamely fuzzy halmaz alakja megfelelő-e. Egyszerűségük miatt leginkább háromszög (A_1), trapéz (A_2), vagy ehhez nagyon hasonló szakaszonként lineáris alakú (ld. 2.2. ábra) alakú tagsági függvényeket használnak. (A 2.2. ábrán látható fuzzy halmazokat használta a bevezetőben már említett cikkében MAMDANI [130] a nagy bonyolultságú gőzgépes rendszer egyik változójának irányítására).

A 2.1. ábrán szereplő négy fuzzy halmaz mindegyike valamely parametrizált függvénycsalád tagja:

$$A_1 = \begin{cases} p_1(x - r) + 1, & \text{ha } x \in [r - (1/p_1), r], \\ p_1(r - x) + 1, & \text{ha } x \in [r, r + (1/p_1)], \\ 0, & \text{különben;} \end{cases}$$

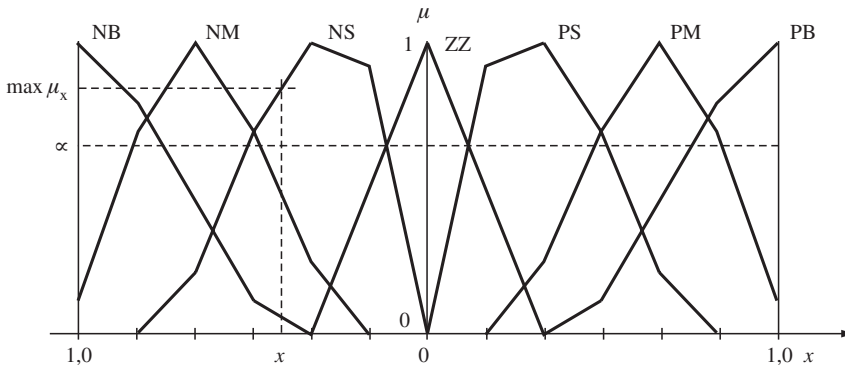
$$A_2 = \begin{cases} 1, & \text{ha } x \in [r - p_2, r + p_2], \\ p_3((x + p_2) - r) + 1, & \text{ha } x \in [r - (1/p_3) - p_2, r - p_2], \\ p_3(r - (x - p_2)) + 1, & \text{ha } x \in [r + p_2, r + (1/p_3) + p_2], \\ 0, & \text{különben;} \end{cases}$$

$$A_3 = \frac{1}{1 + p_4(x - r)^2};$$

$$A_4 = \begin{cases} (1 + \cos(p_5\pi(x - r)))/2, & \text{ha } x \in [r - 1/p_5, r + 1/p_5], \\ 0, & \text{különben;} \end{cases}$$

ahol r a halmaz középpontját (példánkban 2), p_i ($i = 1, \dots, 5$) pedig a halmazok oldaléleit meghatározó konstansok.

Eddig csak a leggyakrabban használt, (2.1) alakú egyszerű fuzzy halmaz típusával foglalkoztunk, melynek többféle általánosítása létezik.

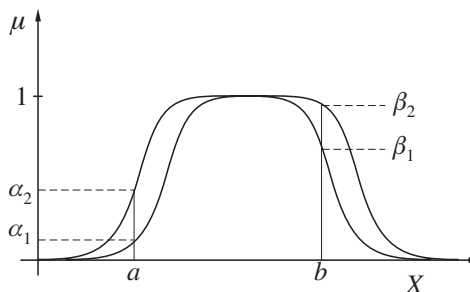


2.2. ábra. A MAMDANI által használt szakaszonként lineáris fuzzy halmazok reprodukciója [130] alapján

Az általánosítást motiváló egyik ok az, hogy valamely alaphalmazbeli elemhez rendelt tagsági érték a valóságban rendelkezésre álló információkhoz képest gyakran túl precíznek bizonyul. Ezért az egyes elemekhez pontos tagsági érték helyett egy intervallumot is rendelhetünk, amely megadja az adott elem tagsági értékének alsó és felső korlátját:

$$A : X \rightarrow \mathcal{E}([0,1]),$$

ahol $\mathcal{E}([0,1])$ a valós számok $[0,1]$ intervallumának zárt intervallumait jelöli. Az ilyen típusú tagsági függvényrel rendelkező halmazokat *intervallumértékű fuzzy halmazoknak* nevezzük. Ezen halmazok ábrázolása két görbe segítségével történik, melyek az egyes elemek alsó és felső korlátját jelölik (2.3. ábra).



2.3. ábra. Intervallumértékű fuzzy halmaz

Az intervallumértékű fuzzy halmazok segítségével az elemekhez rendelt tagságifüggvény-értékek bizonytalansága is modellezhető, amitől egy

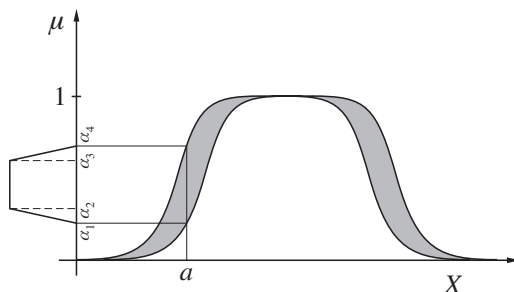
ilyen típusú halmazokat alkalmazó rendszer működésének pontossága csökken, de megbízhatósága nő. Az intervallumértékű fuzzy halmazok alkalmazásának legnagyobb hátránya az, hogy jelentősen növeli a számítási igényt. Ennek következtében — mivel a kisebb tagsági függvény változásra az alkalmazások döntő többsége nem érzékeny — gyakorlati jelentőségük igen korlátozott.

Az intervallumértékű fuzzy halmazok tovább általánosíthatók, ha az intervallumoknak fuzzy értéket is megengedett felvenni. Eszerint minden intervallum maga is lehet egyszerű fuzzy halmaz, ezáltal egy fuzzy halmaz minden eleméhez egy másik fuzzy halmazt rendelünk tagsági értéként. Az

$$A : X \rightarrow \mathcal{F}([0,1])$$

tagsági függvénnyel rendelkező fuzzy halmazokat, *2-es típusú* vagy *másodfajú fuzzy halmaznak* nevezzük, ahol $\mathcal{F}([0,1])$ a $[0,1]$ halmazon definiálható fuzzy halmazok halmaza, másnéven $[0,1]$ *fuzzy hatványhalmaza*.

A másodfajú fuzzy halmazra mutat példát a 2.4. ábra, amelyen $a \in X$ -re a hozzátartozó fuzzy jellegű tagsági érték is ábrázolva van. Minden alaphalmazbeli elem tagsági értékét négy szám jellemzi, melyek a megfelelő trapéz alakú halmaz töréspontjai. Így például a elemhez az $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ rendezett négyes tartozik, melyet a 2.4. ábra baloldalán ábrázoltunk.



2.4. ábra. Példa 2-es típusú vagy másodfajú fuzzy halmazra

Gyakorlati szempontból a másodfajú fuzzy halmazok alkalmazásának hátrányát a szintén igen jelentős számításigény jelenti.

Még bonyolultabb fuzzy típusú halmazok nyerhetők, ha a tagsági értékül nem egyszerű, hanem például másodfajú fuzzy halmazokat rendelünk az egyes elemekhez. Így *3-as típusú* vagy *harmadfajú fuzzy halmazokat* kapunk. Hasonló eljárással tetszőleges, magasabb típusú fuzzy halmazokhoz juthatunk, melyek azonban gyakorlati szempontból már nem bírnak jelentőséggel.

További általánosított fuzzyhalmaz-típushoz vezet, ha nem ragaszkodunk ahhoz, hogy az egyes elemekhez tagsági értéként a $[0,1]$ intervallum valós számait rendeljük. Legyen a tagsági függvény értékészlete egy (leg-
alább részben rendezett) L halmaz:

$$A : X \rightarrow L,$$

ha L -ben létezik valamilyen általános metszet és unió művelet, akkor a rendezés miatt L *háló* (angolul lattice), amiből az L -fuzzy halmaz kifejezés ered.

Mivel L -nek csak részben rendezettségét követeljük meg, ezáltal nagyon általános fogalomhoz jutunk, melyben bennefoglaltatik az eddig tárgyalt összes fuzzyhalmaz-típus.

Másfajta általánosításhoz vezet, ha olyan alaphalmazon definiáljuk a tagsági függvényt mely maga is fuzzy halmaz:

$$A : \mathcal{F}(X) \rightarrow [0,1],$$

ahol $\mathcal{F}(X)$ valamely X halmaz fuzzy hatványhalmaza. Ezeket 2-es szintű fuzzy halmazoknak nevezzük.

Ez a megközelítés lehetővé teszi, hogy bizonytalan, közelítő, csak fuzzy halmazzal leírható alaphalmaz elemeihez is tagsági függvényt rendeljünk. Például az alaphalmazban „ r -hez közeli x ” típusú elemek vannak, ahol r egy konkrét érték, x pedig egy változó. Ahhoz, hogy x értékét meghatározzuk egy egyszerű A fuzzy halmazban, r értékét pontosan meg kellene adni, míg 2-es szintű fuzzy halmazt alkalmazva ez elkerülhető. Feltéve, hogy az „ r -hez közeli x ” tagsági értékét a B fuzzy halmaz reprezentálja, x értéke az A 2-es szintű fuzzy halmazban $A(B)$ lesz.

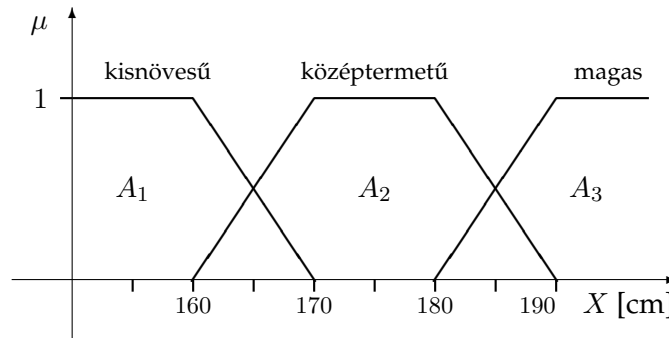
2-es szintű fuzzy halmazok tovább általánosíthatók 3-as és magasabb szintű fuzzy halmazokra, például a 3-as szintű fuzzy halmazok alaphalmaza 2-es szintű fuzzy halmazokból áll.

További általánosítás érhető el például a 2-es szintű és másodfajú fuzzy halmazok kombinációjából, melyek tagsági függvénye

$$A : \mathcal{F}(X) \rightarrow \mathcal{F}([0,1])$$

alakú.

A szakaszban szereplő különböző típusú fuzzy halmazokat egyrészt azért ismertettük, hogy az egyszerű fuzzy halmaz fogalmát általánosító definíciókat is megmutassuk, másrészt elképzelhető, hogy a jövőben egyes általánosított típusok jelentősége megnő, így hasznos, ha az Olvasó ismeri a vonatkozó alapfogalmakat. Könyvünk további fejezeteiben azonban csak 1-es szintű, elsőfajú, azaz egyszerű fuzzy halmazokkal foglalkozunk.



2.5. ábra. Emberek magasságára vonatkozó „kisnövésű”, „középtermetű” és magas fogalmakat reprezentáló fuzzy halmazok.

2.3. Fuzzy halmazok jellemzői

Ebben a szakaszban a fuzzy halmazokkal kapcsolatos alapvető fogalmakat és kifejezéseket vezetjük be. Ezek illusztrálását a 2.5. ábrán végezzük, ahol három trapéz alakú fuzzy halmazzal modellezzük az emberek magasságára vonatkozó „kisnövésű”, „középtermetű” és „magas” fogalmakat. A három fuzzy halmaz tagsági függvényei a $[150, 200]$ intervallumon az alábbi formulákkal adhatók meg:

$$\begin{aligned}
 A_1 &= \begin{cases} 1, & \text{ha } x \leq 160, \\ (170 - x)/10, & \text{ha } 160 < x < 170, \\ 0, & \text{ha } x \geq 170, \end{cases} \\
 A_2 &= \begin{cases} 0, & \text{ha } x \leq 160 \text{ vagy } x \geq 190, \\ (x - 160)/10, & \text{ha } 160 < x < 170, \\ 1, & \text{ha } 170 \leq x \leq 180, \\ (190 - x)/10, & \text{ha } 180 < x < 190, \end{cases} \\
 A_3 &= \begin{cases} 0, & \text{ha } x \leq 180, \\ (x - 180)/10, & \text{ha } 180 < x < 190, \\ 1, & \text{ha } x \geq 190. \end{cases}
 \end{aligned}$$

Az α -vágat (másként α -szint) az egyik legfontosabb fuzzy halmazokkal kapcsolatos fogalom. Valamely adott A fuzzy halmazhoz az A_α α -vágat minden $\alpha \in [0, 1]$ értékre az

$$A_\alpha = \{x | A(x) \geq \alpha\} \quad (2.2)$$

formulával adható meg. Ha (2.2) jobboldalán lévő halmaz definíciójában egyenlőséget nem engedünk meg, akkor *szigorú α -vágatot* kapunk, melyet az $A_{\alpha+}$ szimbólummal jelölünk.

Minden α -vágat (szigorú α -vágat) crisp halmaz, mely az alaphalmaz minden olyan elemét tartalmazza, melynek az adott halmazbeli tagsági értéke α -nál nem kisebb (nagyobb). A 2.5. ábra halmazainál például

$$\begin{aligned}
 A_{1,0} &= A_{2,0} = A_{3,0} = [150, 200] = X; \\
 A_{1,\alpha} &= [150, 170 - 10\alpha], \\
 A_{2,\alpha} &= [160 + 10\alpha, 190 - 10\alpha], \\
 A_{3,\alpha} &= [180 + 10\alpha, 200], \quad \alpha \in (0, 1]; \\
 A_{1,\alpha+} &= [150, 170 - 10\alpha), \\
 A_{2,\alpha+} &= (160 + 10\alpha, 190 - 10\alpha), \\
 A_{3,\alpha+} &= (180 + 10\alpha, 200], \quad \alpha \in [0, 1); \\
 A_{1,1+} &= A_{2,1+} = A_{3,1+} = \emptyset.
 \end{aligned}$$

Az A halmaz összes egymástól különböző α -vágatát tartalmazó halmazt A *színthalmazának* nevezzük:

$$\Lambda(A) = \{\alpha \mid A(x) = \alpha \text{ valamilyen } x \in X\text{-re}\}.$$

2.2. táblázat. A 2.5. ábrán szereplő A_2 halmaz közelítése a diszkrét $\{150, 152, 154, \dots, 200\}$ alaphalmazon

x	$D_2(x)$
$x \notin \{162, 164, \dots, 188\}$	0,0
$x \in \{162, 188\}$	0,2
$x \in \{164, 186\}$	0,4
$x \in \{166, 184\}$	0,6
$x \in \{168, 182\}$	0,8
$x \in \{170, 172, \dots, 180\}$	1,0

Példánk esetében $\Lambda(A_1) = \Lambda(A_2) = \Lambda(A_3) = [0, 1]$, ha azonban diszkrétizáljuk az alaphalmazt — például az A_2 halmaz helyett annak diszkrét D_2 közelítését véve a $\{150, 152, 154, \dots, 200\}$ alaphalmazon (ld. 2.2. táblázat) —, illetve ha folytonos alaphalmaz esetén a tagsági függvény nem folytonos, akkor a színthalmaz a $[0, 1]$ intervallumtól különbözik:

$$\Lambda(D_2) = \{0, 0, 0, 2, 0, 4, 0, 6, 0, 8, 1, 0\}.$$

Szakaszonként lineáris fuzzy halmazok esetén (például háromszög vagy trapéz alakúaknál) azon $\alpha \in [0,1]$ értékeket, melyeknél a tagsági függvénynek töréspontja van, *lényeges α -vágatok*nak nevezzük. Példánk esetében $\Lambda^*(A_1) = \Lambda^*(A_2) = \Lambda^*(A_3) = \{0,1\}$, azaz mindössze kételemű halmaz. Ha nem félrevezető, a $*$ felsőindexet elhagyjuk. A lényeges α -vágatoknak fontos szerepe van számos redukciós eljárásban, így a szabályinterpolációs technikák esetében is, ahol a kimeneti fuzzy halmazokat közelítő módon, ezen α -vágatok segítségével állítják elő (ld. 8. fejezet).

Az α -vágatok fontos tulajdonsága, hogy megfordítják az eredeti $\alpha \in [0,1]$ értékek rendezettségét, azaz minden $\alpha_1, \alpha_2 \in [0,1]$, $\alpha_1 < \alpha_2$ esetén

$$A_{\alpha_1} \supset A_{\alpha_2},$$

valamint

$$A_{\alpha_1} \cap A_{\alpha_2} = A_{\alpha_2}, \quad A_{\alpha_1} \cup A_{\alpha_2} = A_{\alpha_1}.$$

Ebből következik, hogy az α -vágatok (és hasonlóan a szigorú α -vágatok is) egymásba ágyazott halmazcsaládot alkotnak.

Egyes kitüntetett fontosságú α -vágatokra a szakirodalom külön elnevezéseket használ. Valamely A fuzzy halmaznak az alaphalmaz 0-nál nagyobb tagsági értékű pontjainak összességét a halmaz *tartójának* nevezzük, jelölése $\text{supp}(A)$. Formálisan:

$$\text{supp}(A) = \{x | A(x) > 0\} = A_{0+},$$

azaz megegyezik az $\alpha = 0$ értékhez tartozó szigorú α -vágattal. Valamely A fuzzy halmaz *magján* az alaphalmaz 1 tagsági értékkel rendelkező pontjainak összességét értjük. Ez nem más, mint A_1 , vagyis az A halmaz 1-vágata, melyet a $\text{core}(A)$ szimbólummal jelölünk:

$$\text{core}(A) = \{x | A(x) = 1\} = A_1.$$

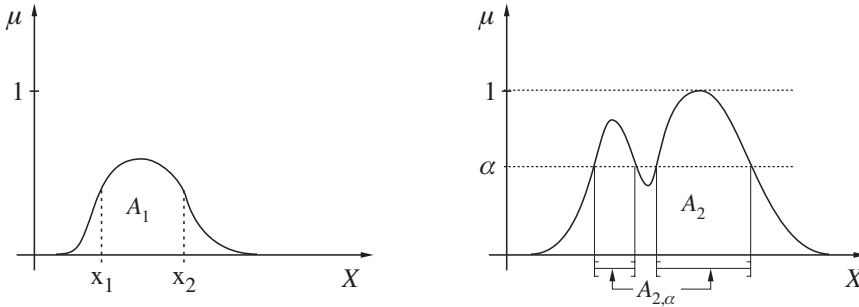
Egy fuzzy halmaz *magasságán* a tagsági függvényének legnagyobb értékét, azaz szuprémumát értjük:

$$h(A) = \sup_{x \in X} A(x).$$

Az A fuzzy halmaz *normális*, ha $h(A) = 1$. Ha ez nem áll fenn ($h(A) < 1$), akkor A *szubnormális*.

Fuzzy halmazokon is értelmezhető a *konvexitás* fogalma, amely a hagyományos halmazokon értelmezett konvexitás általánosítása. Legyen például az X alaphalmaz az $\mathbb{R}^n$ vektortér. Valamely $A \in \mathbb{R}^n$ fuzzy halmaz *konvex*,

ha valamennyi $\alpha \in (0,1]$ vágata a hagyományos értelemben véve konvex. (Itt az $\alpha = 0$ értéket kizárjuk, hiszen ez mindig azonos az alaphalmazzal.) A 2.5. ábra összes halmaza konvex és normális. A 2.6. ábrán látható A_1 halmaz konvex és szubnormális, az A_2 halmaz viszont normális, de nem konvex, hiszen ez utóbbi esetben az ábrán kiemelt α -vágat nem összefüggő, vagyis nem konvex.



2.6. ábra. Példa konvex és szubnormális (A_1), továbbá nemkonvex és normális (A_2) fuzzy halmazokra

Egy fuzzy halmaz konvexitása az alábbi tétel segítségével dönthető el.

2.1. Tétel. Az $\mathbb{R}$ alaphalmazon értelmezett A fuzzy halmaz akkor és csak akkor konvex, ha

$$A(\lambda x_1 + (1 - \lambda)x_2) \geq \min[A(x_1), A(x_2)] \quad (2.3)$$

teljesül minden $x_1, x_2 \in \mathbb{R}$ és $\lambda \in [0,1]$ esetén.

Megjegyzendő, hogy egy fuzzy halmaz konvexitása, nem jelenti azt, hogy a halmaz tagsági függvénye analitikus értelemben konvex a teljes értelmezési tartományon. Ezt jól illusztrálja a 2.6. ábrán látható A_1 fuzzy halmaz, amely ugyan konvex, de tagsági függvénye az $[x_1, x_2]$ szakaszon konkáv.

A szakasz végén néhány elterjedt jelölést ismertetünk. Valamely diszkrét alaphalmazon definiált A fuzzy halmazt az alaphalmaz pozitív tagsági értékű elemeinek és a hozzá tartozó tagsági értékek páronkénti felsorolásával adhatunk meg az alábbi módon:

$$A = a_1/x_1 + a_2/x_2 + \cdots + a_n/x_n, \quad (2.4)$$

ahol $x_1, x_2, \dots, x_n$ rendre az alaphalmaz $a_1, a_2, \dots, a_n > 0$ tagsági értékű elemei. A tört vonal itt az egyes elemek és tagsági értékük összekapcsolására

szolgál, a pluszjel pedig azt szimbolizálja, hogy az adott A halmazt a felsorolt párok összessége definiálja. A 2.2. táblázatban megadott D_2 halmaz ezzel a jelöléssel

$$D_2 = 0,2/162 + 0,2/188 + 0,4/164 + 0,4/186 + 0,6/166 + 0,6/184 + \\ + 0,8/168 + 0,8/182 + 1,0/170 + \dots + 1,0/180$$

módon definiálható. Véges vagy megszámlálhatóan végtelen számosságú alaphalmaz esetén (2.4) helyett az

$$A = \sum_{i=1}^n a_i/x_i, \quad \text{illetve} \quad A = \sum_{i=1}^{\infty} a_i/x_i \quad (2.5)$$

alak is használható. Hasonlóan, ha X a valós számegeyes valamely intervalluma, akkor A az

$$A = \int_X A(x)/x \quad (2.6)$$

alakban is megadható. A (2.5) és (2.6) egyenletekben a szumma- és az integráljel jelentése nem a szokásos, hanem csak az adott (tagsági érték, elem) párok összességét jelöli.

3. fejezet

Műveletek fuzzy halmazokon

A hagyományos (nem fuzzy, crisp) halmazokon értelmezett három alapműveletet, a komplementumképzést (negáció), a metszetet (konjunkció) és az egyesítést (unió, diszjunkció) többféle módon, sőt végtelensokféleképpen lehet általánosítani fuzzy halmazokra. A gyakorlati alkalmazásokban legelterjedtebb, és ezért talán a legjelentősebb ezek közül az ún. ZADEH-féle (*standard*) fuzzy halmazműveletek vagy *alapvető fuzzy műveletek* (melyet ZADEH a már többször idézett legelső cikkében [205] javasolt).

Az X alaphalmazon értelmezett $A \in \mathcal{F}(X)$ fuzzy halmaz ZADEH-féle *komplementense* $\bar{A}$, melyet az alábbi egyenlet határoz meg ($\forall x \in X$):

$$\bar{A}(x) = 1 - A(x). \quad (3.1)$$

Az alaphalmaz azon értékeit, melyre $A(x) = \bar{A}(x)$ fennáll, az A halmaz *egyensúlyi pontjainak* nevezzük. A ZADEH-féle komplementens esetén az egyensúlyi pontok a 0,5 tagságfüggvény-értékű pontok. A 3.1. ábrán látható A_2 halmaz esetén az egyensúlyi pontok értéke 29 és 61.

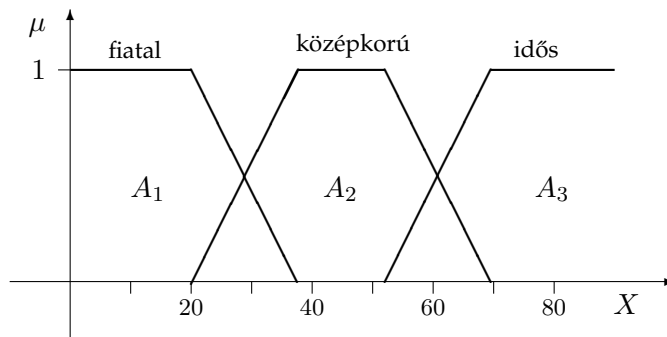
Legyen $A, B \in \mathcal{F}(X)$ két fuzzy halmaz. Ezeknek ZADEH-féle *metszete*, illetve ZADEH-féle *uniója* az alábbi módon határozható meg ($\forall x \in X$):

$$(A \cap B)(x) = \min[A(x), B(x)], \quad (3.2)$$

$$(A \cup B)(x) = \max[A(x), B(x)]. \quad (3.3)$$

Mivel a min és a max művelet asszociatív, ezért ezek a definíciók kiterjeszthetők tetszőleges véges számú fuzzy halmaz esetére is.

Minden fuzzy hatvány halmaz $\mathcal{F}(X)$ (az X alaphalmazon értelmezett összes fuzzy halmazok halmaza) egyben *algebrai háló*, amelyben a fuzzy metszet és fuzzy unió szerepel rendre mint a (háló)metszet (meet) és a (háló)unió (join). Ezt a struktúrát a ZADEH-féle fuzzy komplementessel kiegészítve olyan hálót kapunk, amely a BOOLE-algebrák csaknem minden tulajdonságát (2.1. táblázat) teljesíti, kivétel az ellentmondás és a harmadik kizárásának törvénye. Az ilyen típusú hálót gyakran DE MORGAN-hálónak vagy DE MORGAN-algebrának nevezzük.



3.1. ábra. Példák „fiatal”, „középkorú” és „idős” fogalmakat reprezentáló tagsági függvényekre

Könnyen ellenőrizhetjük, hogy például a harmadik kizárásának törvénye nem teljesül, ugyanis elég megmutatni, hogy a

$$\max[A(x), 1 - A(x)] = 1$$

egyenlet legalább egy $x \in X$ esetén nem áll fenn. Ez nyilvánvaló, hiszen minden $A(x) \in (0,1)$ -re a fenti érték 1-nél kisebb lesz, és csak $A(x) \in \{0,1\}$ esetén teljesül az egyenlőség, azaz a harmadik kizárásának törvénye csak crisp halmazokra áll fenn.

Ha a ZADEH-féle fuzzy műveleteket a $\{0,1\}$ halmazon alkalmazzuk, akkor ugyanúgy működnek mint crisp megfelelőik, ezért ezeket a crisp halmazműveletek általánosításának tekinthetjük. Azonban nem ez az egyetlen lehetőség a halmazműveletek „fuzzifikálására”. Egyik lehetséges alternatívát az ún. *algebrai műveletek* jelentik, melyeket ZADEH is megemlített egy lábjegyzetben az első cikkében [205]. Ő az *interaktív fuzzy halmazműveletek* elnevezést használta, ezzel is utalva arra, hogy a halmazok argumentumai hatással vannak egymásra (ld. (3.14) és (3.15)). Az interaktív műveletek által generált struktúrával, röviden I-fuzzy struktúrával, a 3.5. szakaszban foglalkozunk részletesebben.

Mindhárom alpművelethez léteznek olyan függvényosztályok, melyeknek elemei a hagyományos halmazműveletek fuzzy általánosításai. A következő szakaszokban megadjuk ezen függvényosztályok axiómáit. Itt említjük meg, hogy a fuzzy metszetet és fuzzy uniót a szakirodalomban gyakran *t-normának*, illetve *t-konormának* nevezik, mivel ezek axiomatikus tulajdonságait illetőleg megegyeznek a valószínűségi mértékek egyik geometriai interpretációja alapján nyerhető műveletekkel, melyeket a háromszög-egyenlőtlenség teljesülése miatt trianguláris (háromszög-

)normáknak, röviden t-normáknak nevezték el [154, 155].

Az előbbiek szerint a fuzzy műveletek, ellentétben a hagyományos megfelelőikkel, többfélék lehetnek, ezért különböző problémák esetén más függvények lehetnek a legalkalmasabbak megvalósításukra. Tehát a tagsági függvényen kívül az egyes alpműveletek meghatározása is függhet az adott feladattól. A megfelelő tagsági függvények és műveletek kiválasztása rendkívül fontos a problémák hatékony modellezése érdekében.

Igen lényeges itt látni, hogy a fuzzy halmazokon értelmezhető összes műveletek száma (kontinuum) végtelen, így semmilyen véges művelet-együttes sem alkothat funkcionálisan teljes rendszert. A fuzzy halmazok körében a funkcionális teljességnek nincs is értelme, ezért a választott alpműveletek körét mindig csak analógiás alapon határozzuk meg, valamilyen, a BOOLE-algebrában funkcionálisan teljes műveletrendszerhez hasonló operátorokként. Így értelmes lehet ugyanazon rendszeren belül például többféle t-norma (metszet), stb. egyidejű alkalmazása is. A következőkben ezen analógiás elven fogunk tárgyalni néhány alapvető fontosságú műveletcsaládot.

3.1. Fuzzy komplementek

Legyen A az X fuzzy halmaza. Definíció szerint valamely $x \in X$ -nek az A halmazhoz való tartozásának mértéke $A(x)$. Ekkor az A halmaz c típusú komplementjét (cA) -val jelölve, $(cA)(x)$ az az érték, amilyen mértékben x nem tartozik A -hoz. Tehát a (cA) fuzzy komplementet az alábbi módon definiálhatjuk:

3.1. Definíció. Fuzzy komplementnek nevezzük a

$$c : [0,1] \rightarrow [0,1]$$

függvényt, amely minden $A(x)$ tagsági függvény-értékhez tetszőleges A fuzzy halmaz esetén a $c(A(x))$ értéket rendeli hozzá olyan módon, hogy teljesüljön a fuzzy komplement axiomatikus váza, $c1$ és $c2$ axiómák.

c1 axióma. $c(0) = 1$ és $c(1) = 0$ (peremfeltételek).

c2 axióma. Minden $a, b \in [0,1]$ esetén, ha $a \leq b$, akkor $c(a) \geq c(b)$ (monotonitás).

Az első axióma azt biztosítja, hogy a komplementképzés hagyományos halmazokra a BOOLE-algebrai negációval azonos eredményt adjon. A második axióma azt írja elő, hogy a komplement monoton csökkenő legyen: az A halmaz tagságfüggvény-értékének növekedésével, a komplement $c(A)$ értéke nem nőhet.

Mivel c_1 és c_2 axiómákat igen nagy számú függvény elégíti ki, indokolt lehet még, főleg a gyakorlati szempontokból, további megszorításokat tenni. A szakirodalomban általában még az alábbi két feltétel szerepel a fuzzy komplementek axiómái között.

c3 axióma. c folytonos függvény.

c4 axióma. c involutív, azaz minden $a \in [0,1]$ -re $c(c(a)) = a$.

Ez a négy feltétel (c_1 – c_4) nem független egymástól, amint azt a következő tétel mutatja.

3.1. Tétel. Legyen $c : [0,1] \rightarrow [0,1]$ olyan, ami kielégíti c_2 és c_4 axiómákat. Ekkor c kielégíti a c_1 és c_3 axiómákat is, továbbá c bijekció.

Bizonyítás. 1. Mivel c értékészlete a $[0,1]$ intervallum, ezért $c(0) \leq 1$ és $c(1) \geq 0$. c_2 axióma miatt $c(c(0)) \geq c(1)$, valamint c_4 miatt $0 = c(c(0)) \geq c(1)$, tehát $c(1) = 0$. Ebből és c_4 -ből következik, hogy $c(0) = c(c(1)) = 1$. Azaz a c_1 axióma feltételei teljesülnek.

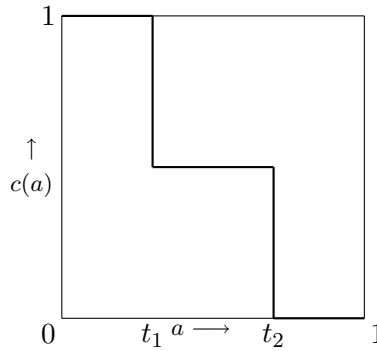
2. A bijekció megmutatásához először vegyük észre, hogy minden $a \in [0,1]$ esetén létezik $b = c(a) \in [0,1]$, amire $c(b) = c(c(a)) = a$ (c_4 felhasználásával). Azaz c ráképezés avagy szürjekció (az értelmezési tartomány minden értékét felveszi). Tegyük fel most, hogy $c(a_1) = c(a_2)$. Ekkor c_4 miatt

$$a_1 = c(c(a_1)) = c(c(a_2)) = a_2,$$

vagyis c egyben injektív függvény, így bijekció.

3. Mivel c bijektív és monoton, ezért folytonos is. ■

A 3.1. tételből következik, hogy minden involutív komplement egyben folytonos komplement is, és a folytonos komplementek halmaza a fuzzy komplementeknek részhalmaza. Összefoglalva tehát a c_1 – c_2 , c_1 – c_3 , és c_1 – c_4 axiómáknak eleget tevő komplementek halmazai szűkülő, egymásba ágyazott struktúrát alkotnak.



3.2. ábra. Kettős küszöb típusú komplement

Az ún. kettős küszöb típusú komplement, melyet a

$$c(a) = \begin{cases} 1, & \text{ha } a \leq t_1, \\ 1/2, & \text{ha } t_1 < a \leq t_2, \\ 0, & \text{ha } a > t_2 \end{cases}$$

formula definiál (ld. 3.2. ábra) például csak az első két axiómát elégíti ki.

Folytonos, de nem involutív a

$$c(a) = \frac{1}{2}(1 + \cos \pi a)$$

függvény, amit könnyen ellenőrizhetünk: például $c(1/3) = 0,75$, $c(0,75) \approx 0,15 \neq 1/3$.

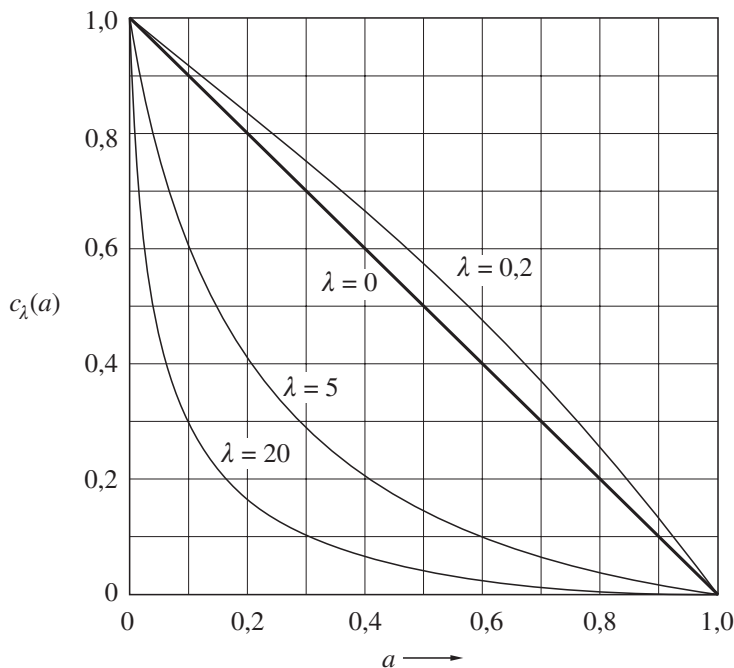
Involutív függvényekből áll a SUGENO-komplementek [170] osztálya, amit a

$$c_\lambda(a) = \frac{1 - a}{1 + \lambda a} \quad (3.4)$$

egyenlet határoz meg, ahol $\lambda \in (-1, \infty)$. A λ paraméter minden egyes értéke különböző involutív fuzzy komplementeket definiál. A 3.3. ábra a SUGENO-osztály néhány elemét illusztrálja, melyek jól mutatják a függvény grafikonjának alakja és a λ értéke közötti összefüggést. Vegyük észre, hogy $\lambda = 0$ esetben a SUGENO-komplement azonos a ZADEH-féle fuzzy komplementtel (3.1).

Egy másik nevezetes, és szintén involutív fuzzy komplementet definiál a

$$c_w(a) = (1 - a^w)^{1/w} \quad (3.5)$$



3.3. ábra. SUGENO-típusú komplementek

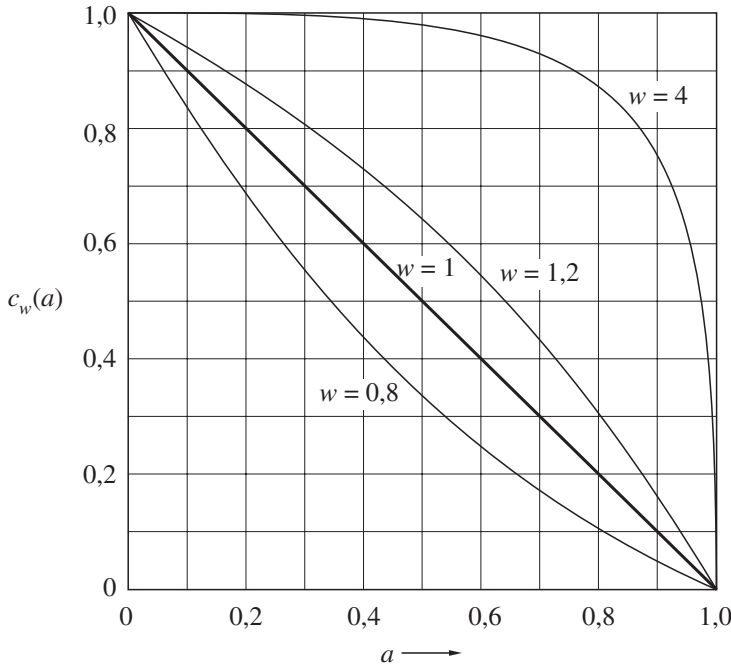
összefüggés, ahol $w \in (0, \infty)$. Ennek az osztálynak az elemeit YAGER-komplementeknek [197] nevezik, melyek közül néhány a 3.4. ábrán látható. Itt azt is megfigyelhetjük, hogyan változik a w paraméter értékétől függően a függvény alakja. $w = 1$ esetén a YAGER-komplement is megegyezik a ZADEH-féle komplementtel.

Világosan látható, hogy mindkét komplementosztály egy paraméter beépítésével keletkezett a ZADEH-féle komplementből.

A fuzzy komplementek tulajdonságaival kapcsolatban még két fogalmat tárgyalunk részletesebben ebben a szakaszban. Először a fejezet bevezetőjében már említett *egyensúlyi pont* jellegzetességeit vizsgáljuk.

A 43. oldalon már definiáltuk a fuzzy halmazok egyensúlyi pontját. A c fuzzy komplement egyensúlyi pontja az az a érték, amire $c(a) = a$ teljesül. Más szóval ez az az érték, amely az A fuzzy halmazban és annak $c(A)$ komplementében azonos tagságfüggvény-értékkel szerepel. A ZADEH-féle komplement egyensúlyi pontja például 0,5, amit az $1 - a = a$ egyenlet megoldásaként kapunk meg.

Könnyen belátható, hogy



3.4. ábra. YAGER-típusú komplementek

3.2. Tétel. Minden fuzzy komplementnek legfeljebb egy egyensúlyi pontja van.

Bizonyítás. Legyen c tetszőleges fuzzy komplement. c egyensúlyi pontja a

$$c(a) - a = 0$$

egyenlet megoldása, ahol $a \in [0,1]$. Megmutatjuk, hogy minden $c(a) - a = b$ alakú egyenletnek maximum egy megoldása van, amiből következik a tétel. Tegyük fel, hogy az egyenletnek létezik két különböző, a_1 és a_2 megoldása ($a_1 < a_2$). Ekkor

$$c(a_1) - a_1 = c(a_2) - a_2. \quad (3.6)$$

Mivel c monoton nemnövekvő, $c(a_1) \geq c(a_2)$, továbbá $a_1 < a_2$ miatt

$$c(a_1) - a_1 > c(a_2) - a_2$$

ami ellentmond (3.6)-nak. ■

Ebből következik, hogy ha létezik c -nek egyensúlyi pontja e_c , akkor az egyértelmű, és $a \leq c(a)$ akkor és csak akkor, ha $a \leq e_c$, valamint $a \geq c(a)$ akkor és csak akkor, ha $a \geq e_c$.

Bizonyítás nélkül megemlítjük, hogy

3.3. Tétel. Minden c folytonos fuzzy komplementnek van egyensúlyi pontja.

A SUGENO-osztály c_λ elemeinek egyensúlyi pontjait az

$$\frac{1 - e_{c_\lambda}}{1 + \lambda e_{c_\lambda}} = e_{c_\lambda}$$

egyenlet pozitív megoldásai adják:

$$e_{c_\lambda} = \begin{cases} ((1 + \lambda)^{1/2} - 1)/\lambda, & \text{ha } \lambda \neq 0, \\ 1/2, & \text{ha } \lambda = 0. \end{cases}$$

Ha adott a c fuzzy komplement és valamely a tagsági érték, akkor a

$$c({}^d a) - {}^d a = a - c(a) \quad (3.7)$$

egyenletnek eleget tevő ${}^d a$ értéket az a (c -re vonatkozó) duálisának nevezzük.

A 3.2. tétel bizonyításából következik, hogy ha a és c adott, akkor a (3.7) egyenletnek legfeljebb egy megoldása van. Sőt, ha c folytonos komplement, akkor könnyen belátható, hogy minden $a \in [0,1]$ pontnak létezik duálisa.

Vizsgáljuk meg a duális pont és az egyensúlyi pont kapcsolatát!

3.4. Tétel. Ha c fuzzy komplementnek létezik egyensúlyi pontja (e_c), akkor

$${}^d e_c = e_c.$$

Bizonyítás. Ha $a = e_c$, akkor a definíció szerint $a = c(a)$, és így $a - c(a) = 0$. Ezenfelül, ha ${}^d a = e_c$, akkor $c({}^d a) = {}^d a$, és így $c({}^d a) - {}^d a = 0$, tehát

$$c({}^d a) - {}^d a = a - c(a)$$

teljesül, ha $a = {}^d a = e_c$. Azaz tetszőleges komplement egyensúlyi pontjának duálisa önmaga. ■

A duális pont fogalmának és az involutív fuzzy komplementek kapcsolatára világít rá a következő

3.5. Tétel. Minden $a \in [0,1]$ esetén, ${}^d a = c(a)$, akkor és csak akkor, ha c involutív.

Bizonyítás. Legyen ${}^d a = c(a)$. Ekkor (3.7)-ben ${}^d a$ helyére $c(a)$ -t helyettesítve

$$c(c(a)) - c(a) = a - c(a).$$

Ezért $c(c(a)) = a$. Az ellenkező irány belátására legyen $c(c(a)) = a$. Ekkor (3.7)-ben a helyére $c(c(a))$ -t helyettesítve a

$$c({}^d a) - {}^d a = c(c(a)) - c(a)$$

összefüggést kapjuk, melynek megoldása ${}^d a = c(a)$. ■

Így tehát minden involutív fuzzy komplementum esetén igaz, hogy bármely tagságfüggvény-érték duálisa azonos ezen érték komplementumával. Abban az esetben, ha a komplementum nem involutív, akkor vagy nem létezik duális vagy nem esik egybe a komplementum értékével.

Az egyensúlyi és duális pontok fogalmának a fuzzyság mértékének vizsgálatában van jelentős szerepe (YAGER [197, 198]). Szintén a fuzzyság mértékének vizsgálata motiválta a [74] tanulmány szerzőit, amely alapos áttekintést is nyújt a témáról, csakúgy mint [86].

3.2. Fuzzy metszetek (t-normák)

Általánosan az A és B fuzzy halmazok metszetét az egységnégyzeten való bináris operátorként adhatjuk meg:

$$t : [0,1] \times [0,1] \rightarrow [0,1],$$

ahol t a t-norma (trianguláris norma) elnevezésre utal. Korábban említettük, hogy a trianguláris (háromszög) norma terminológia használatát az indokolja, hogy a valószínűségi mértékek egy érdekes geometriai interpretációja alapján nyerhető művelet — amire teljesül a háromszög-egyenlőtlenség — axiomatikus tulajdonságait illetőleg megegyezik a fuzzy metszettel.

A következőkben felsorolt axiómák a fuzzy metszet azon minimálisan elvárt tulajdonságait fogalmazzák meg, melyek a BOOLE-féle metszetfogalom természetes általánosítását adják.

t1 axióma. $t(a,1) = a$ minden $a \in [0,1]$ -re (peremfeltétel).

t2 axióma. $b \leq c$ -ből következik, hogy $t(a,b) \leq t(a,c)$ minden $a,b,c \in [0,1]$ -re (monotonitás).

t3 axióma. $t(a,b) = t(b,a)$ minden $a,b \in [0,1]$ -re (kommutativitás).

t4 axióma. $t(a, t(b, c)) = t(t(a, b), c)$ minden $a, b, c \in [0, 1]$ -re (asszociativitás).

Ezeket az axiómákat a *fuzzy metszetek (t-normák) axiomatikus vázának* hívjuk.

Könnyen belátható, hogy az első három axióma azt biztosítja, hogy hagyományos (nem fuzzy) halmazokra a kétváltozós fuzzy metszet, mint a hagyományos metszet általánosítása, a szokásos eredményeket adja. Az első axióma alapján $t(0, 1) = 0$ és $t(1, 1) = 1$, a kommutativitás miatt $t(1, 0) = 0$, míg $t(0, 0) = 0$ a monotonitásból következik. A monotonitás és a kommutativitás azt a természetes követelményt fejezik ki, hogy ha A -ban vagy B -ben a tagságfüggvény-érték csökken, az nem eredményezheti a metszet növekedését. Az utolsó axióma segítségével a t-normák definíciója tetszőleges véges számú argumentumra is kiterjeszthető.

A szakirodalomban még az alábbi megszorításokat szokták tenni a t-normákra:

t5 axióma. t folytonos függvény

t6a axióma. $t(a, a) < a$ (szubidempotencia), vagy **t6b** $t(a, a) = a$ (idempotencia).

t7 axióma. Ha $a_1 < a_2$ és $b_1 < b_2$, akkor $t(a_1, b_1) < t(a_2, b_2)$ (szigorú monotonitás).

A folytonosság megkövetelése biztosítja az olyan szituációk elkerülését, mikor az egyik argumentum kicsiny megváltozása a metszetben nagy (nem folytonos) változást idéz elő. A *szubidempotencia* a nemfuzzy metszetre vonatkozó *idempotencia* gyengébb formája, mely azt az esetet kezeli, amikor a két argumentum megegyezik.

A t7 axióma a monotonitásnak egy erősebb formája.

Ha egy t-norma folytonos és szubidempotens, akkor *arkhimédészi t-normának*, ha ezen felül szigorúan monoton, akkor *szigorú arkhimédészi t-normának* nevezzük.

E fejezet bevezetésében már említettük, hogy a ZADEH-féle fuzzy műveletek DE MORGAN-algebrát alkotnak, azaz idempotensek. Most megmutatjuk, hogy a fuzzy metszetek közül az idempotencia csak a ZADEH-féle metszetre áll fenn.

3.6. Tétel. A ZADEH-féle fuzzy metszet az egyetlen idempotens t-norma [15].

Bizonyítás. Nyilvánvaló, hogy a minimum művelet idempotens: $\min(a,a) = a$ minden $a \in [0,1]$ esetén. Tegyük fel, hogy t idempotens. Ekkor minden $a, b \in [0,1]$, $a \leq b$ -re

$$a = t(a,a) \leq t(a,b) \leq t(a,1) = a$$

t_1 és t_2 axiómák felhasználásával. Azaz $t(a,b) = a = \min(a,b)$. Hasonlóan $a \geq b$ esetén

$$b = t(b,b) \leq t(a,b) \leq t(1,b) = b$$

azaz $t(a,b) = b = \min(a,b)$. Tehát, ha egy t -norma idempotens, akkor az a ZADEH-féle fuzzy metszet. ■

Alábbiakban a fuzzy metszetként leggyakrabban használt t -normákat mutatjuk be, grafikonjaik a 3.5. ábrán láthatók:

ZADEH-féle metszet: $t(a,b) = \min(a,b)$.

Algebrai szorzat: $t(a,b) = ab$.

Korlátos különbség: $t(a,b) = \max(0, a + b - 1)$.

Drasztikus metszet: $t_{\min}(a,b) = \begin{cases} a, & \text{ha } b = 1, \\ b, & \text{ha } a = 1, \\ 0, & \text{egyébként.} \end{cases}$

A 3.5. ábrán is megfigyelhető a fuzzy metszetek közötti alábbi összefüggés:

3.7. Tétel. Minden $a, b \in [0,1]$ esetén

$$t_{\min}(a,b) \leq t(a,b) \leq \min(a,b). \quad (3.8)$$

Bizonyítás. Felső korlát. t_1 és t_2 axiómák felhasználásával

$$t(a,b) \leq t(a,1) = a,$$

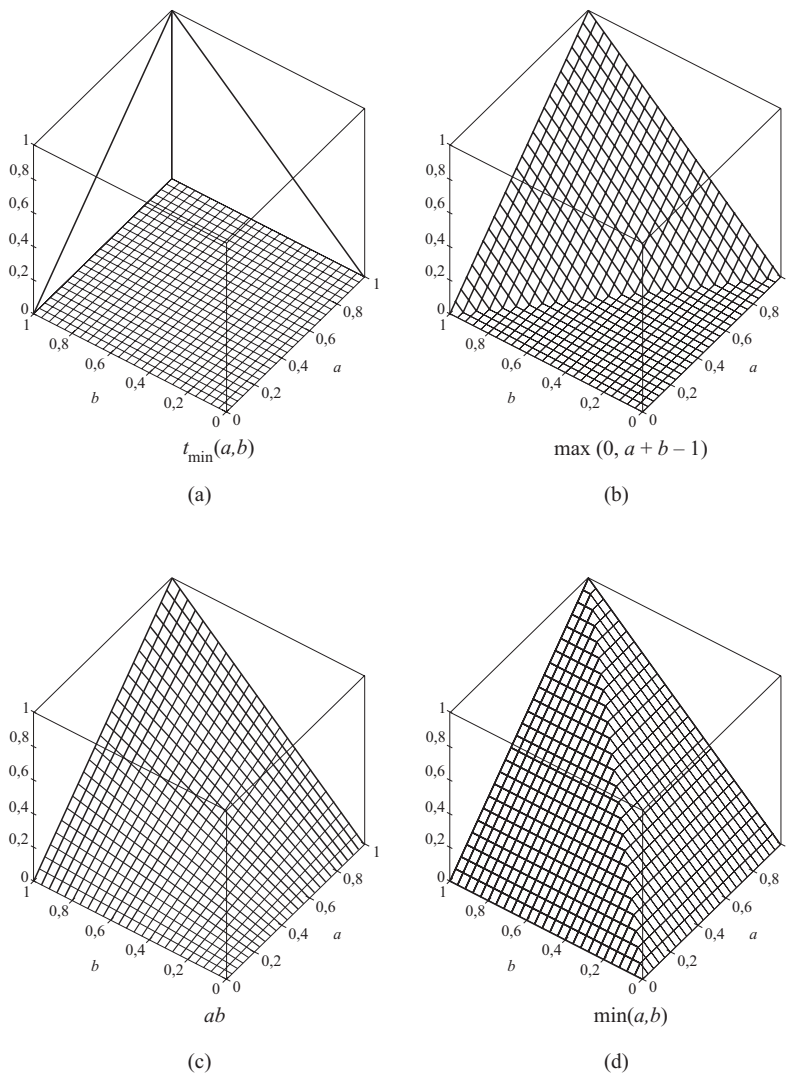
továbbá a kommutativitás miatt

$$t(a,b) = t(b,a) \leq t(b,1) = b.$$

Azaz $t(a,b) \leq a$ és $t(a,b) \leq b$, így $t(a,b) \leq \min(a,b)$.

Alsó korlát. t_1 axiómából következik, hogy $t(a,b) = a$, ha $b = 1$, és $t(a,b) = b$, ha $a = 1$. Mivel $t(a,b) \leq \min(a,b)$ és $t(a,b) \in [0,1]$ ezért

$$t(a,0) = t(0,b) = 0.$$



3.5. ábra. Fuzzy metszetek grafikonjai

A monotonitás miatt

$$t(a,b) \geq t(a,0) = t(0,b) = 0.$$

Tehát a fuzzy metszetek alsó korlátja az erős metszet. ■

Végül néhány ismertebb t-norma fontosabb adatait mutatjuk be a 3.1. táblázatban. További fuzzy metszetek találhatóak DOMBI [44], FRANK [61]

és WEBER [193] közleményeiben.

3.3. Fuzzy uniók (t-konormák, s-normák)

Mivel a fuzzy metszet és unió duális művelet pár, a t-konormák (vagy s-normák) tulajdonságai hasonlóak a t-normákéhoz, s ezért tárgyalásuk során gyakran az előző szakasz analóg eredményeire hivatkozunk. A és B halmazok fuzzy uniója az egységnégyzeten való bináris operátorként definiálható:

$$s : [0,1] \times [0,1] \rightarrow [0,1].$$

A fuzzy uniók alaptulajdonságait leíró axiómák a következők:

s1 axióma. $s(a,0) = a$, minden $a \in [0,1]$ -re (peremfeltétel).

s2 axióma. $b \leq c$ -ből következik, hogy $s(a,b) \leq s(a,c)$, minden $a,b,c \in [0,1]$ -re (monotonitás).

s3 axióma. $s(a,b) = s(b,a)$, minden $a,b \in [0,1]$ -re (kommutativitás).

s4 axióma. $s(a,s(b,c)) = s(s(a,b),c)$, minden $a,b,c \in [0,1]$ -re (asszociativitás).

Ezt a négy axiómát a *fuzzy uniók (t-konormák) axiomatikus vázának* hívjuk.

Az s1–s4 és t1–t4 axiómákat összehasonlítva láthatjuk, hogy csak a peremfeltételben különböznek. Az első három axióma — hasonlóképpen mint a t-normáknál — biztosítja, hogy a fuzzy unió crisp halmazok esetén a hagyományos halmazműveletekkel megegyező eredményt adjon.

További kiegészítő feltételeket fogalmaznak meg az alábbi axiómák:

s5 axióma. s folytonos függvény.

s6a axióma. $s(a,a) > a$ (szuperidempotencia), **s6b** $s(a,a) = a$ (idempotencia).

s7 axióma. Ha $a_1 < a_2$ és $b_1 < b_2$, akkor $s(a_1,b_1) < s(a_2,b_2)$ (szigorú monotonitás).

A fenti három axióma a t5–t7 axiómától csak abban különbözik, hogy a szubidempotenciát szuperidempotencia helyettesíti.

Arkhimédészinek, illetve szigorú arkhimédészinek nevezzük a monoton, illetve szigorúan monoton, szuperidempotens és folytonos t-konormákat.

3.1. táblázat. Fuzzy metszetek ismertebb osztályai ([86] alapján)

Hivatkozás	formula	paraméterérték	a formula értéke, ha a paraméter 0-hoz konvertál	a formula értéke, ha a paraméter ∞-hez konvertál
ŁUKASIEWICZ/ ZADEH [205]	$\min(a,b)$			
SCHWEIZER és SKLAR [155]	$\{\max(0,a^p + b^p - 1)\}^{1/p}$	$p \neq 0$	ab	$t_{\min}(a,b)$ ha $p \rightarrow \infty$ $\min(a,b)$ ha $p \rightarrow -\infty$
HAMACHER [69]	$\frac{ab}{r + (1-r)(a+b-ab)}$	$r \in (0,\infty)$	$\frac{ab}{a+b-ab}$	$t_{\min}(a,b)$
YAGER [198]	$1 - \min[1, ((1-a)^w + (1-b)^w)^{1/w}]$	$w \in (0,\infty)$	$t_{\min}(a,b)$	$\min(a,b)$
DUBOIS és PRADE [48]	$\frac{ab}{\max(a,b,\alpha)}$	$\alpha \in [0,1]$	$\min(a,b)$	

3.8. Tétel. A ZADEH-féle unió az egyetlen idempotens t-konorma [15].

A bizonyítás analóg a 3.6. tételnél leírtakkal, így azt az Olvasóra hagyjuk.

Az alábbiak a gyakorlatban is gyakran használt fuzzy uniók (3.6. ábra):

ZADEH-féle unió: $s(a,b) = \max(a,b)$.

Algebrai összeg: $s(a,b) = a + b - ab$.

Korlátos összeg: $s(a,b) = \min(1, a + b)$.

Drasztikus unió: $s_{\max}(a,b) = \begin{cases} a, & \text{ha } b = 0, \\ b, & \text{ha } a = 0, \\ 1, & \text{egyébként.} \end{cases}$

Hasonló összefüggés áll fenn a ZADEH-féle és a drasztikus unióra, mint a metszeteknél ismertetett megfelelőikre:

3.9. Tétel. Minden $a, b \in [0,1]$ és s fuzzy unióra:

$$\max(a,b) \leq s(a,b) \leq s_{\max}(a,b) \quad (3.9)$$

A bizonyítás menete azonos a 3.7. tételével.

Végül a 3.2. táblázatban néhány fontosabb t-konormát ismertetünk. További fuzzy uniókat definiáltak DOMBI [44], FRANK [61] és WEBER [193] is.

Ugyanúgy, mint a klasszikus halmazelméletben, fuzzy kontextusban is a DE MORGAN-azonosságok kapcsolják össze a metszetet és az uniót.

3.2. Definíció. A t t-norma és az s t-konorma a c fuzzy komplementtel összekapcsolva, akkor és csak akkor alkot duális, ha teljesítik a DE MORGAN-azonosságokat, azaz ha a

$$c(t(a,b)) = s(c(a),c(b)) \quad (3.10)$$

$$c(s(a,b)) = t(c(a),c(b)). \quad (3.11)$$

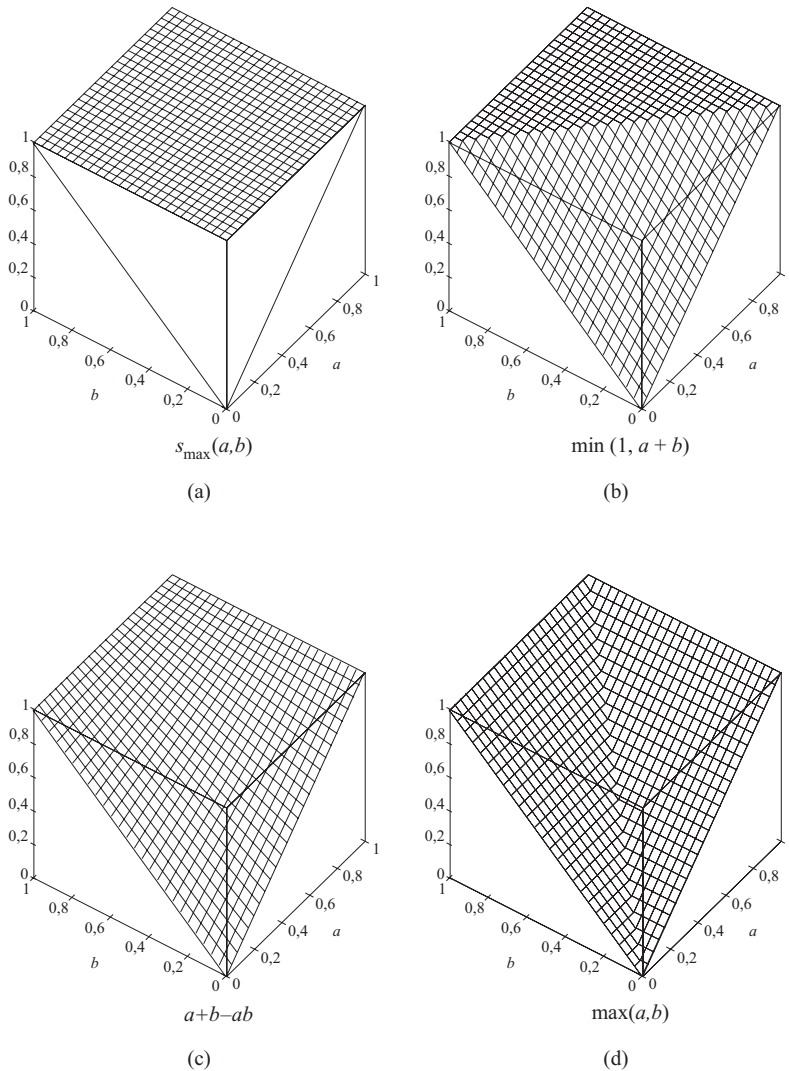
egyenletek fennállnak. Ekkor a $\langle t,s,c \rangle$ hármast duális, vagy DE MORGAN-hármasnak nevezzük.

Duális hármast alkotnak például a ZADEH-féle fuzzy komplementtel kiegészítve az 57. oldalon ismertetett műveletek a megfelelő párjaikkal. Sőt, ezek közül a $\min$, $\max$ és $t_{\min}$, $s_{\max}$ párosok tetszőleges fuzzy komplementtel teljesítik a DE MORGAN-azonosságokat.

A t-normák és t-konormák további általános vizsgálata található például FODOR [56, 57] munkáiban. A témáról alapos áttekintést nyújt még GUPTA és QI [67] is.

3.2. táblázat. Fuzzy uniók ismertebb osztályai ([86] alapján)

Hivatkozás	formula	paraméterérték	a formula értéke, ha a paraméter 0-hoz konvertál	a formula értéke, ha a paraméter ∞-hez konvertál
ŁUKASIEWICZ/ ZADEH [205]	$\max(a,b)$			
SCHWEIZER és SKLAR [155]	$1 - \{\max(0, (1-a)^p + (1-b)^p - 1)\}^{1/p}$	$p \neq 0$	$a + b - ab$	$s_{\max}(a,b)$ ha $p \rightarrow \infty$ $\max(a,b)$ ha $p \rightarrow -\infty$
HAMACHER [69]	$\frac{a + b - (2-r)ab}{1 - (1-r)ab}$	$r \in (0,\infty)$	$\frac{a + b - 2ab}{1 - ab}$	$s_{\max}(a,b)$
YAGER [198]	$\min[1, (a^w + b^w)^{1/w}]$	$w \in (0,\infty)$	$s_{\max}(a,b)$	$\max(a,b)$
DUBOIS és PRADE [48]	$\frac{a + b - ab - \min(a,b, 1-\alpha)}{\max(1-a, 1-b, \alpha)}$	$\alpha \in [0,1]$	$\max(a,b)$	



3.6. ábra. Fuzzy uniók grafikonjai

3.4. Aggregációs operátorok

Fuzzy halmazokon értelmezett *aggregációs operátorok* több fuzzy halmaz megfelelő módon történő egyesítése által egyetlen fuzzy halmazt állítanak elő. Példaként nézzük azt az esetet, mikor egy hallgató tanulmányi átlageredményét szeretnénk meghatározni. Tegyük fel, hogy ehhez rendel-

kezésünkre állnak a jeles, jó, közepes, elégséges és elégtelen fogalmakat a $[0,100]$ -as skálán definiáló fuzzy halmazok. Ekkor a tanuló átlagteljesítményét aggregációs művelet felhasználásával egyetlen fuzzy halmazzal adhatjuk meg.

3.3. Definíció. A

$$h : [0,1]^n \rightarrow [0,1]$$

függvényt n ($n \geq 2$) fuzzy halmazokon értelmezett aggregációs operátornak nevezzük. Ha a h függvény argumentumai az X alaphamazon értelmezett $A_1(x), \dots, A_n(x)$ fuzzy halmazok, akkor h minden $x \in X$ -re fuzzy halmazt állít elő az argumentumok tagsági értékeinek segítségével, azaz

$$A(x) = h(A_1(x), \dots, A_n(x)).$$

Egy jól definiált aggregációs műveletnek az alábbi három axiomatikus feltételt kell kielégítenie:

h1 axióma. $h(0, \dots, 0) = 0$ és $h(1, \dots, 1) = 1$ (peremfeltételek).

h2 axióma. Ha adott két tetszőleges n -es $\langle a_1, \dots, a_n \rangle$ és $\langle b_1, \dots, b_n \rangle$, ahol $a_i, b_i \in [0,1]$ és $a_i \leq b_i$ minden $i \in [1,n]$ -re, akkor

$$h(a_1, \dots, a_n) \leq h(b_1, \dots, b_n),$$

azaz h monoton növekvő minden argumentumában.

h3 axióma. h folytonos függvény.

Az aggregációs operátorokra vonatkozóan e három feltételen kívül még a további megszorításokat tehetjük:

h4 axióma. h szimmetrikus minden argumentumában, azaz

$$h(a_1, \dots, a_n) = h(a_{p(1)}, \dots, a_{p(n)}),$$

ahol p az $1, \dots, n$ számok tetszőleges permutációja.

h5 axióma. h idempotens, azaz $h(a, \dots, a) = a$, minden $a \in [0,1]$ esetén.

A h4 axióma az argumentumok egyenrangúságát fejezi ki. Az ötödik axióma azt a megközelítést írja le, mely szerint ha azonos halmazokat aggregálunk, akkor az eredmény legyen ugyanaz a halmaz. Vegyük észre, hogy h5-ből következik h1.

Könnyen igazolható, hogy az előző szakaszokban tárgyalt t-normák és t-konormák szintén aggregációs operátorok. Ezek ugyan a h_1 – h_3 esetében csak kétargumentumos műveletek, de mint már utaltunk rá, az asszociativitás segítségével (t_4 , s_4 axiómák) tetszőleges véges argumentumszámra kiterjeszthetők. E műveletek azonban, ahogy azt a 3.6. és 3.8. tételekben beláttuk, a ZADEH-féle operátoroktól eltekintve nem idempotensek.

Most megmutatjuk, hogy

3.10. Tétel. *A h_2 és h_5 axiómáknak eleget tevő aggregációs műveletek minden $\langle a_1, \dots, a_n \rangle \in [0,1]^n$ esetén teljesítik a*

$$\min(a_1, \dots, a_n) \leq h(a_1, \dots, a_n) \leq \max(a_1, \dots, a_n) \quad (3.12)$$

egyenlőtlenséget.

Bizonyítás. Legyen $a_* = \min(a_1, \dots, a_n)$ és $a^* = \max(a_1, \dots, a_n)$. Ha h monoton nő és idempotens, akkor

$$a_* = h(a_*, \dots, a_*) \leq h(a_1, \dots, a_n) \leq h(a^*, \dots, a^*) = a^*.$$

Fordítva, ha h kielégíti (3.12)-t, akkor

$$a = \min(a, \dots, a) \leq h(a, \dots, a) \leq \max(a, \dots, a) = a$$

miatt h_5 axiómát is minden $a \in [0,1]$ -re. ■

Így minden aggregációs operátor, amely a ZADEH-féle fuzzy műveletek közé esik idempotens, és megfordítva — a 3.6. és 3.8. tételekből következően — csak a (3.12) egyenlőtlenségnek eleget tevő aggregációs operátorok azok. Ezeket gyakran átlagoló operátoroknak is nevezzük.

Az átlagoló operátorok egyik osztálya, mely a minimum és maximum közt lévő teljes intervallumot befutja az *általános (hatvány)közép*, amit a

$$h_\alpha(a_1, \dots, a_n) = \left(\frac{a_1^\alpha + \dots + a_n^\alpha}{n} \right)^{1/\alpha} \quad (3.13)$$

egyenlet definiál, ahol $\alpha \neq 0$, és $\prod_{i=1}^n a_i \neq 0$, ha $\alpha < 0$. α néhány kitüntetett értékére nevezetes közepeket kapunk. Például, ha $\alpha \rightarrow 0$, akkor h_α a geometriai középhez konvergál, abban az esetben pedig, ha $\alpha = 1$ vagy $\alpha = -1$, akkor h_α rendre a számtani, illetve a harmonikus középkel azonos.

Szintén lefedi az előbb említett teljes intervallumot a rendezett súlyozott átlagoló operátorok osztálya, melyet angol nyelvű megfelelőjének rövidítéséként OWA (ordered weighted averaging) operátornak is nevezünk [199].

Legyen $\underline{w} = \langle w_1, \dots, w_n \rangle$ súlyvektor, amire minden $w_i \in [0,1]$ esetén

$$\sum_{i=1}^n w_i = 1.$$

Ekkor a $\underline{w}$ súlyvektorhoz tartozó OWA operátor a

$$h_{\underline{w}}(a_1, \dots, a_n) = w_1 b_1 + \dots + w_n b_n$$

formulával adható meg, ahol b_i az i -edik legnagyobb elem $a_1, \dots, a_n$ közül. Vagyis a $\langle b_1, \dots, b_n \rangle$ vektor az $\langle a_1, \dots, a_n \rangle$ vektor csökkenő sorrendben rendezett permutációja: $b_i \geq b_j$, ha $i < j$, $i, j \in [1, n]$.

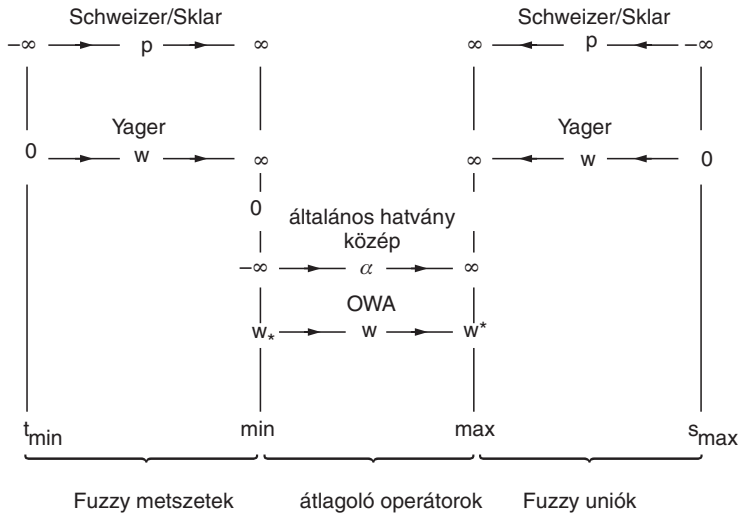
Legyen például $\underline{w} = \langle 0,2, 0,6, 0,15, 0,05 \rangle$, ekkor $h_{\underline{w}}(0,6, 0,8, 0,1, 1,0) = 0,2 \cdot 1 + 0,6 \cdot 0,8 + 0,15 \cdot 0,6 + 0,05 \cdot 0,1 = 0,775$.

Egyszerűen belátható, hogy $h_{\underline{w}}$ kielégíti a h1–h5 axiómákat, és így a (3.12) egyenlőtlenséget is. Az alsó és felső korlátot rendre a $\underline{w}_* = \langle 0, \dots, 0,1 \rangle$ és $\underline{w}^* = \langle 1,0, \dots, 0 \rangle$ súlyvektorok esetén kapjuk meg.

Az aggregációs operátorokat a 3.7. ábra összegzi, melyen csak néhány jelentősebb t-norma, t-konorma és átlagoló operátor osztály lett feltüntetve. Minden esetben jelöltük az odatartozó paraméter értékkészletét.

3.5. I-fuzzy struktúrák

A fuzzy logika bevezetésének egyik fő célja az volt, hogy segítségével könnyebben lehessen modellezni az emberi gondolkodásmódot és a természetes nyelvi fogalmakat, ugyanis a BOOLE-algebra merev tulajdonságai többnyire távol állnak a mindennapi gondolkodásunk, érvelésünk igazi menetétől. A ZADEH-féle fuzzy műveletek már közelebb állnak az emberi gondolkodáshoz, de még mindig túl merevnek bizonyulnak, inkább a tudományos vagy mérnöki megközelítést írják le. Nézzük példaként az idempotencia fogalmát. Akárhány azonos fuzzy halmaznak vesszük a maximumát, az eredmény mindig ugyanaz lesz. A bevezetésben is említettük, hogy a többek közt „metatudománnyal” is foglalkozó SELYE JÁNOS [157] szerint egyes lágy természettudományos területeken, ahol a tudományos eredmények nem feltétlenül matematikai értelemben pontos tapasztalatok kiértékelése alapján jönnek létre (például biológia), a kutatóknak más



3.7. ábra. Fuzzy aggregációs operátorok

modellt kell követniük eredményeik feldolgozásában. Minél több esetben adnak a kísérletek vagy vizsgálatok egybehangzó, a hipotézist alátámasztó eredményt, annál inkább meg lehet győződve a kutató hipotézise igazáról. Ugyanezt a módszert alkalmazzuk mindennapi döntéseink meghozatalában is. Tegyük fel, hogy kirándulást terveztünk mára, de némileg felhős az ég. Ha többen is azt mondogatják, hogy „nem rossz az idő, nem fog esni”, akkor egyre inkább meggyőzzük magunkat feltételezésünk igazáról. Minél többször, minél több forrásból halljuk ugyanazt az állítást, az annál inkább hihetőbbnek tűnik. A szubjektív emberi logika tehát közel sem idempotens, hanem inkább szuperidempotens (illetve szubidempotens)!

SELYE érvelését a következőképpen fogalmazhatjuk meg fuzzy műveletekre: „A t-norma és a t-konorma legyen szigorúan monoton, kivéve a szélső értékekre, amikor legalább egy argumentum 0 vagy 1 (t_{6a} és s_{6a} axiómák).” Ez a megkötés eleve kizárja a ZADEH-féle fuzzy műveleteket. A következőkben egy olyan minimális axiómarendszert mutatunk be, mely a fentiekben megfogalmazott feltételeknek eleget tesz. A két alapl művelet az algebrai vagy interaktív fuzzy metszet és unió:

$$t(a,b) = ab \quad (3.14)$$

$$s(a,b) = a + b - ab, \quad (3.15)$$

melyet egy bináris különbség művelet (d) egészít ki az unáris komplement helyett. A komplement a $d(1,a)$ művelet eredményeként adható meg. Ez

az általánosabb megfogalmazás egyes alkalmazásokban különösen előnyös lehet (például fuzzy flip-flop [103, 144]).

3.4. Definíció. *A fuzzy különbség vagy differencia bináris művelet*

$$d : [0,1] \times [0,1] \rightarrow [0,1],$$

mely kielégíti az I-fuzzy algebra megfelelő axiómáit.

Az I-fuzzy algebra olyan legalább két elemből és a $\langle t, s, d \rangle$ műveleti hármasból álló algebrai struktúra, mely az i1–i13 axiómákban megszabott feltételeket kielégíti. Ezen axiómák közül az i1, i2, i3 és i4 axióma rendre azonos az s1, s3, s4 (t-konorma peremfeltétele, asszociativitása és kommutativitása) és t1 (t-norma peremfeltétele) axiómával.

i5 axióma. $s(t(a,b), t(a,c)) > t(a, s(b,c))$, minden $a, b, c \in [0,1]$ -re akkor és csak akkor, ha $s(t(a,b), t(a,c)) \neq 0$ és $t(a, s(b,c)) \neq 1$ (t -re vonatkozó disztributív egyenlőtlenség).

i6 axióma. $s(a,b) > a$, minden $a \in [0,1]$ és $b \in (0,1]$ esetén (s -re vonatkozó szigorú monotonitása).

i7 axióma. $d(a,a) = 0$, minden $a \in [0,1]$ -re (fuzzy halmaz távolsága önmagától).

i8 axióma. Ha $a < b < c$, akkor $d(b,c) < d(a,c)$, minden $a, b, c \in [0,1]$ esetén (d szigorú monotonitása).

i9 axióma. $d(a,b) = d(b,a)$, minden $a, b \in [0,1]$ -re (d kommutativitása).

i10 axióma. $d(1, d(1, a)) = a$ (involúció d -vel kifejezve).

i11 axióma. $d(1, s(a,b)) = t(d(1,a), d(1,b))$ (DE MORGAN-azonosság).

i12 axióma. $d(a,b) = d(d(1,a), d(1,b))$ (szimmetria).

i13 axióma. Az $s(a,x) = b$ egyenletnek legfeljebb egy megoldása van, ha $a \in [0,1]$, $b \in [0,1]$ (s feltételes invertálhatósága).

A fenti axiómák közül az s -re kimondott axiómák t -re vonatkozó duális párjai az axiómarendszer segítségével bebizonyíthatók, ezért az I-fuzzy struktúrák axiómarendszere duális t -re és s -re.

Az alábbiakban az I-fuzzy struktúrák néhány egyszerűbb, de lényeges tulajdonságát ismertetjük, melyek három csoportra oszthatók: az első tartalmazza az axiómarendszerben t-konormára kimondott tulajdonságok t-normára vonatkozó duálisait (i8, i9, i13 axiómák duálisai); a másodikban a 3.2. és 3.3. szakaszban t1–t7 és s1–s7 axiómák közül szerepel néhány, melyek i5–i13 között nincsenek, és nem mondanak ellent a szigorú monotonitásnak (t3, t4, valamint t1 és s1 axiómák párjai); a harmadik csoportban a fuzzy különbségre vonatkozó tulajdonságok vannak. Mivel a fuzzy különbség a fuzzy komplement általánosítása, ezért ez utóbbi csoportban olyan tulajdonságok is előfordulnak, melyekre vonatkozó analóg állítás a fuzzy komplementre nem mondható ki. A bizonyítások a [90] és [107] közleményekben megtalálhatóak.

Az I-fuzzy algebrák leglényegesebb különbsége az általános fuzzy struktúrákkal szemben a t-norma és a t-konorma szigorú monotonitása. Létezik-e olyan műveleti hármas, amely eleget tesz a fenti megszorításoknak? A válasz igen, a legegyszerűbb példa a szakasz elején említett algebrai műveleti páros egy megfelelő fuzzy különbséggel kiegészítve.

3.11. Tétel. *Az algebrai (vagy interaktív) műveletek a $d(a,b) = |a - b|$ fuzzy különbséggel I-fuzzy algebrát alkotnak.*

Ezekon kívül is végtelen sok megfelelő műveleti hármas létezik. Ilyen a már említett HAMACHER-féle [69] (fuzzy differencia vagy komplement nélkül) művelethármas. A 70-es évek közepén RÖDDER a természetes nyelvekben használt logikai műveletek kiértékelését, az „intuitív” logikát vizsgálta [150], és megfigyelése igen jól összecseng a SELYE-féle megállapításokkal, valamint HAMACHER műveleteivel [69]. E munkájában meghatározta a lehetséges legáltalánosabb I-fuzzy struktúrának eleget tevő *racióális függvényeket*, melyek érdekes összefüggést mutatnak KÓCZY által javasolt — SELYE [157] munkáját felhasználó — I-fuzzy struktúrákkal [90]. Egyszerű behelyettesítéssel ellenőrizhető, hogy ezek a HAMACHER-féle műveletek, melyek a 3.1. és 3.2. táblázatokban megtalálhatóak, a szokásos fuzzy különbséggel kiegészítve I-fuzzy algebrát alkotnak.

4. fejezet

Fuzzy relációk

A hagyományos relációk két vagy több halmaz elemei közötti összefüggést, kapcsolatot vagy éppen annak hiányát fejezik ki. Ennek alapján két (vagy több) halmazbeli elem vagy relációban van egymással, vagy nem. Ezt a fogalmat általánosítja és árnyalja a *fuzzy reláció* fogalma, amellyel két halmaz elemei közötti kapcsolat 0 és 1 közötti mértékét is modellezhetjük. Egy fuzzy relációhoz való tartozást ugyanúgy *tagsági értékkel* lehet kifejezni, mint egy elemnek valamely fuzzy halmazhoz való tartozását. A klasszikus relációk tehát a fuzzy relációk speciális esetének tekinthetők, ahol a tagsági függvény értéke csak 0 vagy 1 lehet.

Az $X_1, X_2, \dots, X_n$ halmazok közötti R relációt úgy definiáljuk, mint a reláció alaphalmazai DESCARTES-szorzatának részhalmazát:

$$R(X_1, \dots, X_n) \subseteq X_1 \times \dots \times X_n,$$

azaz ekkor a $\times_{i=1}^n X_i$ szorzathalmaz az univerzum, ennek az elemeire vonatkozik a reláció. A relációt hagyományos esetben például *karakterisztikus függvényével* lehet reprezentálni, amelyet szintén R -rel jelölünk:

$$R(x_1, \dots, x_n) = \begin{cases} 1, & \text{akkor és csak akkor, ha } \langle x_1, \dots, x_n \rangle \in R, \\ 0, & \text{egyébként.} \end{cases}$$

Fuzzy esetben a karakterisztikus függvény azonos a reláció tagsági függvényével:

$$R(x_1, \dots, x_n) = \mu_R \langle x_1, \dots, x_n \rangle, \quad (4.1)$$

tehát a relációban bármely $\langle x_1, \dots, x_n \rangle$ n -es tetszőleges 0 és 1 közötti értékkel szerepelhet; ez a reláció tagsági függvényének értéke az adott argumentumra.

A relációk egyik lehetséges osztályozása a relációban szereplő halmazok számán alapul. Eszerint két alaphalmaz esetén *bináris*, három esetén *ternáris*, általános esetben n alaphalmaz esetén *n -áris* relációról beszélünk. Ennek megfelelően — egy másik gyakori reprezentációs módszerként, mely

főként a számítógépes modellezésben jelentős — a véges elemszámú halmazok relációit rendezett n -esekként is felírhatjuk. Legyen $\underline{R} = [r_{i_1, i_2, \dots, i_n}]$ egy n -dimenziós tömb (másnéven mátrix). Ekkor i_1 dimenzió minden eleme az X_1 halmaz pontosan egy eleméhez tartozik, hasonlóan i_2 dimenzió minden eleme X_2 -höz, és így tovább. Azaz a $\langle x_1, \dots, x_n \rangle$ n -est $r_{i_1, \dots, i_n}$ mátrixelemmel is reprezentálhatjuk.

Tekintsük az alábbi példát: legyenek az alaphalmazok $X = \{\text{CH}, \text{D}, \text{B}, \text{F}\}$, $Y = \{\text{frank}, \text{márka}\}$, $Z = \{\text{német}, \text{francia}, \text{olasz}, \text{flamand}\}$, és az R reláció kapcsolja össze egy ország autós felségjelzését, valutánemét és hivatalos nyelvét vagy nyelveit. Ekkor

$$R(X, Y, Z) = \{ \langle \text{CH}, \text{frank}, \text{német} \rangle, \langle \text{CH}, \text{frank}, \text{francia} \rangle, \langle \text{CH}, \text{frank}, \text{olasz} \rangle, \\ \langle \text{B}, \text{frank}, \text{flamand} \rangle, \langle \text{B}, \text{frank}, \text{francia} \rangle, \langle \text{F}, \text{frank}, \text{francia} \rangle, \\ \langle \text{D}, \text{márka}, \text{német} \rangle \}$$

hármaskok tartoznak a relációba, amit az alábbi két háromdimenziós mátrixszal is szemléltethetünk:

	CH	D	B	F		CH	D	B	F
német	1	0	0	0	német	0	1	0	0
francia	1	0	1	1	francia	0	0	0	0
olasz	1	0	0	0	olasz	0	0	0	0
flamand	0	0	1	0	flamand	0	0	0	0
frank					márka				

Hasonló módon ábrázolhatunk fuzzy relációkat is. Legyen R bináris reláció, mely a „nagyon távoli” fogalmat modellezi az $X = \{\text{Bp.}, \text{Sydney}, \text{London}\}$ és az $Y = \{\text{Hong Kong}, \text{Bp.}\}$ halmazok között. A reláció elemeit felsorolhatjuk

$$R(X, Y) = 0,9/(\text{Bp.}, \text{HK}) + 0,5/(\text{Sydney}, \text{HK}) + 1/(\text{London}, \text{HK}) \\ + 1/(\text{Sydney}, \text{Bp.}) + 0,3/(\text{London}, \text{Bp.}),$$

vagy mátrixszerűen is ábrázolhatjuk:

	Bp.	Sydney	London
HK	0,9	0,5	1
Bp.	0	1	0,3

Felsorolás esetében a nulla tagsági értékű párokat általában elhagyjuk.

4.1. Projekció és hengeres kiterjesztés

A projekció és a hengeres kiterjesztés fuzzy relációkon értelmezett műveletek, melyek rendkívül fontos szerepet játszanak a fuzzy szabályalapú irányítási rendszerekben (ld. 7.3. szakasz). Ezeket a fogalmakat szintén ZADEH definiálta elsőként [209, 210].

Tekintsük az $\mathcal{X} = \{X_i | i \in \mathbb{N}_n\}$ halmazok DESCARTES-szorzatát, és legyen $\underline{x} = \langle x_i | i \in \mathbb{N}_n \rangle$ az $\times_{i \in \mathbb{N}_n} X_i$ halmaz egy eleme, valamint $\underline{y} = \langle y_j | j \in J \rangle$ az $\times_{j \in J} X_j$ halmaz eleme. Így ha $J \subset \mathbb{N}_n$ és $|J| = r \leq n$, továbbá ha $y_j = x_j$ minden $j \in J$, akkor $\underline{y}$ -t az $\underline{x}$ részsorozatának nevezzük, amit az $\underline{y} \prec \underline{x}$ szimbólummal jelölünk. Ekkor a J halmazt szokás az $\mathbb{N}_n$ halmaz alterének is nevezni.

4.1. Definíció. Legyen $R(X_1, \dots, X_n)$ egy fuzzy reláció, ekkor $[R \downarrow \mathcal{Y}]$ jelöli R -nek az $\mathcal{Y}$ halmazcsaládra vetített projekcióját, mely R -nek csak az $\mathcal{Y} = \{X_j | j \in J \subset \mathbb{N}_n\}$ halmazokon vett értékét veszi figyelembe. Ekkor a projekció tagsági függvényét, mely az $\mathcal{Y}$ halmazon értelmezett, az alábbi módon adhatjuk meg:

$$[R \downarrow \mathcal{Y}](\underline{y}) = \max_{\underline{y} \prec \underline{x}} R(\underline{x}). \quad (4.2)$$

Bizonyos feltételek mellett (4.2) általánosítható úgy, hogy a max műveletet tetszőleges t-konormával helyettesítjük.

A projekció inverzének tekinthető bizonyos értelemben a hengeres kiterjesztés.

4.2. Definíció. Legyenek $\mathcal{X}$ és $\mathcal{Y}$ a 4.1. definícióban meghatározott halmazcsaládok. Továbbá legyen R az $\mathcal{Y}$ halmazainak DESCARTES-szorzatán értelmezett reláció. Ekkor a $[R \uparrow \mathcal{X} - \mathcal{Y}]$ szimbólummal jelöljük az R reláció ($\mathcal{X} - \mathcal{Y}$ halmazok DESCARTES-szorzatára való) hengeres kiterjesztését, melynek értékét az

$$[R \uparrow \mathcal{X} - \mathcal{Y}](\underline{x}) = R(\underline{y}) \quad (4.3)$$

egyenlet határozza meg, minden $\underline{x}$ -re, ahol $\underline{y} \prec \underline{x}$.

Vegyük észre, hogy a hengeres kiterjesztés a legnagyobb fuzzy relációt állítja elő egy adott projekcióhoz abban az értelemben, hogy a kiterjesztett halmazokon vett tagságfüggvény-értékek a lehető legnagyobbak. Ez egyszersmind a legkevésbé meghatározott reláció, mely kompatibilis az adott projekcióval. A hengeres kiterjesztésnek ezen tulajdonságát úgy is megfogalmazhatjuk, hogy a művelet maximalizálja egy n -dimenziós reláció meghatározottlanságát (nonspecificity) a reláció r -dimenziós projekciói

egyikének a felhasználásával. Tehát a bővebb reláció elkészítéséhez nem használunk fel olyan információt, mely az adott projekcióban nincs meg.

E két művelet illusztrálására nézzük az alábbi példát. Az $X_1 \times X_2 \times X_3$ halmazon értelmezett R reláció értékeit a 4.1. táblázatban adtuk meg, ahol $X_1 = \{A, B\}$, $X_2 = \{A, B\}$ és $X_3 = \{A, B, C\}$. $[R_{ij} \downarrow \{X_i, X_j\}]$ és $[R_i \downarrow \{X_i\}]$ jelöli rendre a két, illetve egy halmazra vett projekciókat, ahol $i, j \in \{1, 2, 3\}$.

A 4.1. táblázatból könnyen meghatározhatjuk az egyes projekciók hengeres kiterjesztéseit. Például

$$[R_{13} \uparrow \{X_2\}](A, A, C) = [R_{13} \uparrow \{X_2\}](A, B, C) = R_{23}(A, C) = 0,6$$

vagy

$$\begin{aligned} [R_2 \uparrow \{X_1, X_3\}](A, A, A) &= [R_2 \uparrow \{X_1, X_3\}](A, A, B) = \\ [R_2 \uparrow \{X_1, X_3\}](A, A, C) &= [R_2 \uparrow \{X_1, X_3\}](B, A, A) = \\ [R_2 \uparrow \{X_1, X_3\}](B, A, B) &= [R_2 \uparrow \{X_1, X_3\}](B, A, C) = R_2(A) = 0,9. \end{aligned}$$

Megfigyelhetjük, hogy a felsoroltak között nincs olyan projekció, amelynek az $X_1 \times X_2 \times X_3$ halmazra való kiterjesztése azonos az eredeti R relációval. Ez azt jelenti, hogy ebben a példában minden projektálás során elvesztünk valamennyi információt, amit a hengeres kiterjesztéssel már nem lehetett rekonstruálni, sőt ehhez az összes projekció sem elég!

Bár létezik olyan reláció, mely rekonstruálható összes ortogonális projekciójának hengeres kiterjesztéséből, ám ez igen ritkán fordul elő. Az

4.1. táblázat. Példa ternáris relációra (R) és projekcióira

$\langle x_1, x_2, x_3 \rangle$	$R(x_1, x_2, x_3)$	R_{12}	R_{13}	R_{23}	R_1	R_2	R_3
A A A	0,1	0,3	0,4	0,7	0,6	0,9	1,0
A A B	0,2	0,3	0,5	0,8	0,6	0,9	0,8
A A C	0,3	0,3	0,6	0,9	0,6	0,9	0,9
A B A	0,4	0,6	0,4	1,0	0,6	1,0	1,0
A B B	0,5	0,6	0,5	0,5	0,6	1,0	0,8
A B C	0,6	0,6	0,6	0,6	0,6	1,0	0,9
B A A	0,7	0,9	1,0	0,7	1,0	0,9	1,0
B A B	0,8	0,9	0,8	0,8	1,0	0,9	0,8
B A C	0,9	0,9	0,9	0,9	1,0	0,9	0,9
B B A	1,0	1,0	1,0	1,0	1,0	1,0	1,0
B B B	0,0	1,0	0,8	0,5	1,0	1,0	0,8
B B C	0,5	1,0	0,9	0,6	1,0	1,0	0,9

általánosabb az, hogy néhány projekciójának segítségével egy relációt pontosan vissza lehet kapni. Az ily módon előállított relációt *hengeres lezárt*nak hívjuk. Mikor a ZADEH-féle uniót (max) használjuk projektálásra, akkor a hengeres lezártat a ZADEH-féle metszet segítségével határozzuk meg.

4.3. Definíció. Legyen adott az $\mathcal{X}$ halmazon definiált R reláció projekcióinak egy halmaza, $\{P_i | i \in I\}$. Ekkor a reláció ezen projekció által generált hengeres lezártja

$$\text{cyl}\{P_i\}(\underline{x}) = \min_{i \in I} [P_i \uparrow \mathcal{X} - \mathcal{Y}_i](\underline{x}), \quad (4.4)$$

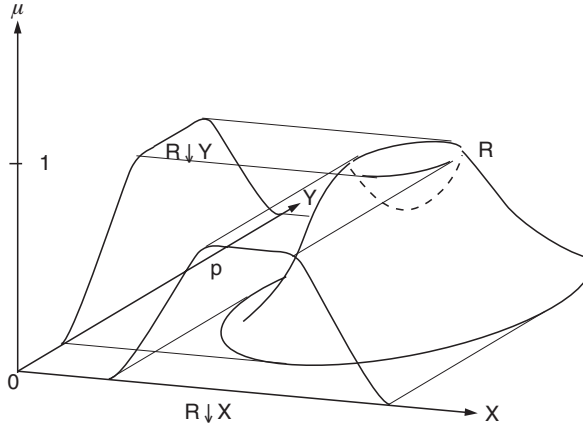
ahol $\mathcal{Y}_i$ jelöli azt a halmazcsaládot, amin a P_i projekció definiálva van.

Tekintsük a 4.1. táblázatban megadott relációt. A 4.2. táblázatban három projekciócsalád segítségével előállított hengeres lezártat mutatunk be. Figyeljük meg, hogy egyik sem azonos az eredeti relációval, vagyis a projekcióiból R nem rekonstruálható teljesen.

4.2. táblázat. R reláció három projekciócsalád által generált hengeres lezártja

$\langle x_1, x_2, x_3 \rangle$	$\text{cyl}\{R_{12}, R_{13}, R_{23}\}$	$\text{cyl}\{R_1, R_2, R_3\}$	$\text{cyl}\{R_{12}, R_3\}$
A A A	0,3	0,6	0,3
A A B	0,3	0,6	0,3
A A C	0,3	0,6	0,3
A B A	0,4	0,6	0,6
A B B	0,5	0,6	0,6
A B C	0,6	0,6	0,6
B A A	0,7	0,9	0,9
B A B	0,8	0,8	0,8
B A C	0,9	0,9	0,9
B B A	1,0	1,0	1,0
B B B	0,5	0,8	0,8
B B C	0,6	0,9	0,9

A jelenség magyarázatához vizsgáljuk meg a 4.1. ábrán látható relációt. Nyilvánvaló, hogy a reláció egyik projekciója sem fogja tartalmazni az eredeti relációban meglévő kráterszerű bemélyedést, ezért például az ehhez hasonló alakú relációk nem teljesen rekonstruálhatók.



4.1. ábra. Példa nem teljesen rekonstruálható fuzzy relációra

4.2. Bináris fuzzy relációk

A bináris relációk megkülönböztetett jelentőséggel bírnak az n -dimenziós relációk között, hiszen bizonyos tekintetben a matematikai függvények általánosításai. Ugyanis míg egy X -ből Y -ba képező függvény csak egy értéket rendelhet valamely $x \in X$ -hez, addig egy reláció bármely X -beli elemhez tetszőleges számú Y -belit. Először néhány, a függvényeknél is közismert fogalom bináris fuzzy relációkra vonatkozó megfelelőjét ismertetjük.

Legyen $R(X, Y)$ bináris (másképpen binér) fuzzy reláció. Ekkor értelmezési tartományát, X -et, $\text{dom } R$ -rel, értékkészletét, Y -t, $\text{ran } R$ -rel jelöljük, melyeket az alábbi összefüggések határoznak meg:

$$\text{dom } R(x) = \max_{y \in Y} R(x, y) \quad \forall x \in X, \quad (4.5)$$

$$\text{ran } R(y) = \max_{x \in X} R(x, y) \quad \forall y \in Y. \quad (4.6)$$

Azaz valamely $x \in X$ a relációban előforduló maximális értékével tartozik a reláció értelmezési tartományába. Hasonlóan, valamely $y \in Y$ a relációban előforduló maximális értékével tartozik a reláció értékkészletéhez. Definiálható továbbá egy reláció magassága is:

$$h(R) = \max_{x \in X} \max_{y \in Y} R(x, y). \quad (4.7)$$

Vagyis $h(R)$ megegyezik a legnagyobb tagságfüggvény-értékű $\langle x, y \rangle$ pár értékével.

A bináris relációkat *tagsági mátrixszal* (4.8) vagy *páros gráffal* szokták ábrázolni; ez utóbbit a szakirodalomban általában „íjszerű” *diagramnak* nevezik. A második esetben a két alaphalmazhoz tartozó elemeket csúcsokkal (vagy csomópontokkal) jelöljük úgy, hogy a különböző alaphalmazba tartozó elemek jól elkülönüljenek egymástól. A pozitív tagságfüggvény-értékű párokat vonallal kötjük össze, amin a köztük levő reláció értéke szerepel (ld. 4.2. ábra.)

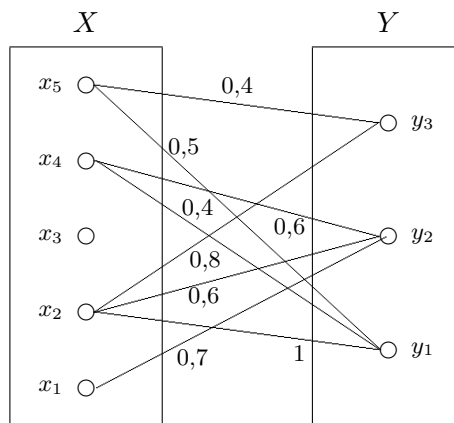
$$\underline{R} = \begin{bmatrix} 0,7 & 0 & 0 \\ 1 & 0,6 & 0,8 \\ 0 & 0 & 0 \\ 0,4 & 0,6 & 0 \\ 0,5 & 0 & 0,4 \end{bmatrix} \quad (4.8)$$

Az $R(X,Y)$ fuzzy reláció *inverze* az $Y \times X$ szorzathalmazon értelmezett $R^{-1}(Y,X)$ reláció, melynek értékeit az

$$R^{-1}(y,x) = R(x,y)$$

egyenlet határozza meg, minden $x \in X$ és $y \in Y$ esetén. Az inverz reláció mátrixa, $\underline{R}^{-1}$ az eredeti reláció mátrixának traszponáltja lesz, azaz $\underline{R}^{-1}$ sorai $\underline{R}$ oszlopaival egyeznek meg és fordítva. Nyilvánvaló, hogy a fuzzy relációkon végzett inverzió involutív:

$$(\underline{R}^{-1})^{-1} = \underline{R}. \quad (4.9)$$



4.2. ábra. Reláció ábrázolása páros gráffal („íjszerű” diagrammal)

Legyen adott két fuzzy reláció $P(X,Y)$ és $Q(Y,Z)$. Ezen relációk *max-min kompozíciója*, melyet $P \circ Q(X,Z)$ -vel jelölünk az $X \times Z$ szorzathalmazon értelmezett

$$R(x,z) = [P \circ Q](x,z) = \max_{y \in Y} \min[P(x,y), Q(y,z)] \quad (4.10)$$

reláció, minden $x \in X$ és $z \in Z$ -re. Látható, hogy a kompozíció képzéséhez a ZADEH-féle uniót és metszetet használtuk, innen ered a művelet neve. A (4.10) összefüggés segítségével rögtön belátható, hogy

$$[P(X,Y) \circ Q(Y,Z)]^{-1} = Q^{-1}(Z,Y) \circ P^{-1}(Y,X), \quad (4.11)$$

$$[P(W,X) \circ Q(X,Y)] \circ R(Y,Z) = P(W,X) \circ [Q(X,Y) \circ R(Y,Z)], \quad (4.12)$$

azaz a max-min kompozíció asszociatív, és az inverze azonos az inverz relációk fordított kompozíciójával. Ugyanakkor a kommutativitás már nem teljesül, hiszen $X \neq Z$ esetén a művelet nem is értelmezhető, de többnyire még $X = Z$ esetén sem lesz igaz. A max-min kompozíció illusztrálására nézzük az alábbi példát.

Legyen a két fuzzy reláció tagsági mátrixukkal megadva: $\underline{P} = [p_{ij}]$, $\underline{Q} = [q_{jk}]$. Ekkor a kompozíciójuk az

$$r_{ik} = [\max_j \min(p_{ij}, q_{jk})] \quad (4.13)$$

összefüggés alapján számolandó. Eszerint

$$\underline{R} = \begin{bmatrix} 0,2 & 0,4 & 0,5 & 0,3 \\ 0,1 & 0,7 & 0,8 & 0,7 \\ 1 & 0 & 0,2 & 0,9 \end{bmatrix} \circ \begin{bmatrix} 0,6 & 0,3 \\ 0,5 & 0,1 \\ 0 & 0,4 \\ 0,8 & 1 \end{bmatrix} = \begin{bmatrix} 0,4 & 0,4 \\ 0,7 & 0,7 \\ 0,8 & 0,9 \end{bmatrix},$$

ahol például az r_{11} és r_{32} elemek értékét (4.13) alapján az alábbi módon kapjuk:

$$\begin{aligned} r_{11} &= 0,4 = \max[\min(0,2, 0,6), \min(0,4, 0,5), \min(0,5, 0), \min(0,3, 0,8)] \\ &= \max[\min(p_{11}, q_{11}), \min(p_{12}, q_{21}), \min(p_{13}, q_{31}), \min(p_{14}, q_{41})] \\ r_{32} &= 0,9 = \max[\min(1, 0,3), \min(0, 0,1), \min(0,2, 0,4), \min(0,9, 1)] \\ &= \max[\min(p_{31}, q_{12}), \min(p_{32}, q_{22}), \min(p_{33}, q_{32}), \min(p_{34}, q_{42})]. \end{aligned}$$

Az előbb ismertetett max-min kompozíciótól lényegében csak az értelmezési tartományában tér el a *relációs összekapcsolás*, amely — megtartva a

fenti jelöléseket — az $X \times Y \times Z$ halmazon van definiálva. Tehát a $P(X,Y)$ és $Q(Y,Z)$ fuzzy relációk $P * Q$ -val jelölt (relációs) összekapcsolása egy ternáris relációt határoz meg az alábbi szerint:

$$R(x,y,z) = [P * Q](x,y,z) = \min[P(x,y), Q(y,z)], \quad (4.14)$$

minden $X \in X, y \in Y$ és $z \in Z$ esetén.

A fenti műveletek közötti legfontosabb különbség, hogy az egyik bináris, míg a másik ternáris fuzzy relációt eredményez. Valójában a max-min kompozíciót megkaphatjuk úgy is, hogy az összekapcsolás megfelelő elemeit a ZADEH-féle fuzzy unióval aggregáljuk:

$$[P \circ Q](x,z) = \max_{y \in Y} [P * Q](x,y,z), \quad \forall x \in X, y \in Y, z \in Z. \quad (4.15)$$

A max-min kompozíción és a hozzátartozó összekapcsoláson kívül más hasonló célú műveletek is képezhetők, ha a ZADEH-féle műveletek helyett tetszőleges t-normát és t-konormát használunk, például a *max-algebrai kompozíció* a

$$R(x,z) = [P \circ_{\max\text{-alg}} Q](x,z) = \max_{y \in Y} (P(x,y) \cdot Q(y,z))$$

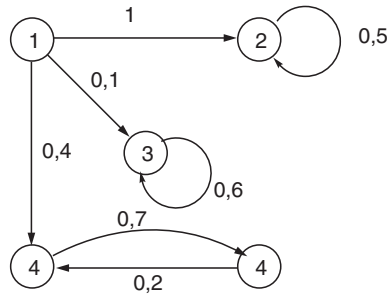
egyenlettel definiálható.

4.3. Írányított gráfok

A bináris relációk egyik jellegzetes csoportját képviselik azok a relációk, melyek értékészlete és értelmezési tartománya megegyezik. Az ilyen relációkat *írányított gráfoknak* is nevezzük, és általában $R(X,X)$ -szel, vagy $R(X^2)$ -tel jelöljük. Az elnevezés arra utal, hogy az ilyen típusú relációk grafikusán írányított gráfként reprezentálhatók. Egy ilyen reláció ugyanis az előző szakaszban bemutatott ábrázolási módszeren kívül úgy jeleníthető meg, hogy X elemeinek gráfcsúcsokat feleltetünk meg, és a relációban lévő elemeket reprezentáló csúcspárokat írányított élekkel kötjük össze. Ezt az ábrázolási módot szemlélteti a 4.3. ábra.

Az $R(X,X)$ alakú (fuzzy) relációkat a következő három fő jellemző alapján osztályozhatjuk: reflexivitás, szimmetria, tranzitivitás. Először tekintsük át röviden ezeket a tulajdonságokat binér crisp relációkra.

Az $R(X,X)$ crisp reláció *reflexív*, ha minden $x \in X$ elem relációban van önmagával, azaz $\langle x,x \rangle \in R$. Ilyen például a $\leq$ (kisebb vagy egyenlő) és a $\equiv$ (kongruencia: rögzített $m \in \mathbb{N}$ modulóval való maradékos osztás). Ha



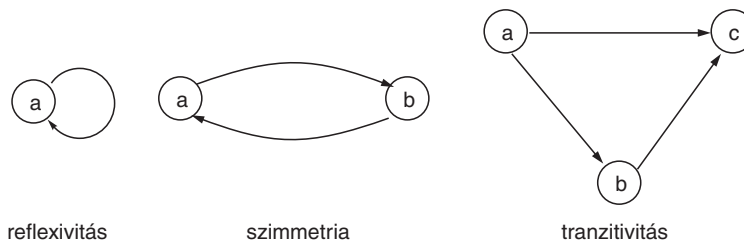
4.3. ábra. Reláció reprezentálása irányított gráffal $X = Y$ esetén

ez a tulajdonság nem teljesül valamely $x \in X$ -re, akkor a reláció *irreflexív*. Ha minden $x \in X$ elem esetén $\langle x, x \rangle \notin R$, akkor a reláció *antireflexív*, mint például a $\neq$ (nem egyenlő).

Az $R(X, X)$ crisp reláció akkor és csak akkor *szimmetrikus*, ha minden relációbeli $\langle x, y \rangle$ párra az $\langle y, x \rangle$ pár is a relációban van. Erre példa a „házastárs” vagy az egyenlőség reláció. Ha van olyan pár, amire ez a tulajdonság nem áll fenn, akkor a reláció *aszimmetrikus*. Ha $\langle x, y \rangle \in R$ és $\langle y, x \rangle \in R$ -ből az következik, hogy $x = y$, akkor a reláció *antiszimmetrikus*, mint például a $\leq$ reláció. Továbbá, ha $x \neq y$ esetén pontosan az egyik pár van a relációban, akkor a reláció *szigorúan antiszimmetrikus*.

Az $R(X, X)$ crisp reláció akkor és csak akkor *tranzitív*, amennyiben valamely y -ra $\langle x, y \rangle, \langle y, z \rangle \in R$, akkor az $\langle x, z \rangle \in R$ is igaz. A $\leq, \equiv, =$ relációk tranzitívak, de a $\neq$, vagy a „házastárs” reláció nem az. Ha ez az összefüggés nem teljesül valamely $x, y, z \in X$ hármasra, akkor a relációt *nontranzitív*nak, továbbá ha minden $\langle x, y \rangle, \langle y, z \rangle \in R$ esetén $\langle x, z \rangle \notin R$, akkor a relációt *antitranszitiv*nak nevezzük.

E három alapvető jellemzőt ábrázolja grafikusan a 4.4. ábra.



4.4. ábra. Reflexivitás, szimmetria és tranzitivitás reprezentálása irányított gráffal

Ezen három tulajdonság könnyen kiterjeszthető fuzzy relációkra is. Ennek alapján $R(X, X)$ fuzzy reláció *reflexív*, ha

$$R(x, x) = 1, \quad \text{minden } x \in X\text{-re.} \quad (4.16)$$

Ha (4.16) nem áll fenn valamely elemre, akkor R *irreflexív*, ha egy elemre sem áll fenn, akkor *antireflexív*. A reflexivitás gyöngített formája az ún. ε -*reflexivitás*, amikor az

$$R(x, x) \geq \varepsilon$$

egyenlőtlenség teljesülését követeljük meg valamely $0 < \varepsilon < 1$ értékre.

Az $R(X, X)$ fuzzy reláció *szimmetrikus*, ha

$$R(x, y) = R(y, x) \quad \text{minden } x, y \in X\text{-re.}$$

Ha ez valamely elempárra nem teljesül, akkor *aszimmetrikus* fuzzy relációról beszélünk. Továbbá, ha minden $x, y \in X$ párra $R(x, y) > 0$ és $R(y, x) > 0$ -ból következik, hogy $x = y$, akkor a reláció *antiszimmetrikus*.

Egy fuzzy reláció *transzítív* (pontosabban *max-min transzítív*), ha minden $\langle x, z \rangle \in X^2$ párra

$$R(x, z) \geq \max_{y \in X} \min[R(x, y), R(y, z)] \quad (4.17)$$

teljesül. Ha bizonyos elemekre a reláció nem elégíti ki a (4.17) összefüggést, akkor *nontranszítív*, valamint ha

$$R(x, z) < \max_{y \in X} \min[R(x, y), R(y, z)]$$

minden $\langle x, z \rangle \in X^2$ párra, akkor *antitranszítív* fuzzy relációról beszélünk. Nyilvánvaló, hogy a (4.17) max-min transzitivitás definíciója a (4.10) max-min kompozíción alapszik. Más t-normák, illetve t-konormák segítségével alternatív fuzzy transzitivitásfogalmakat lehet alkotni, melyek egyes alkalmazásokban hasznosnak bizonyulhatnak.

Az eddig tárgyalt három alapfogalom segítségével, mely háromféle reflexivitást, valamint négyféle szimmetriát, illetve transzitivitást foglal magába, összesen 36 különböző irányított gráf típusú fuzzy relációt különböztethetünk meg. Ezek közül a leglényegesebbeket tartalmazza a 4.5. ábra, melyekből az ekvivalencia, a kompatibilitási és a fuzzy rendezési relációkat a következő 4.4. szakasz tárgyalja részletesebben.

Egy crisp reláció *transzítív lezártjának* azt a relációt nevezzük, mely transzítív, tartalmazza $R(X, X)$ -et, és a lehető legkevesebb elemet tartalmazza.

	reflexivitás	szimmetria	transzitivitás	antireflexivitás	antiszimmetria
ekvivalencia					
kompatibilitás vagy tolerancia					
részben rendezés vagy parciális rendezés					
kvázi ekvivalencia					
szigorú rendezés					

4.5. ábra. Az $R(X, X)$ alakú relációk fontosabb típusai

Fuzzy relációk esetén az utolsó feltétel általánosabban azt követeli meg, hogy a lehető legkisebb tagságfüggvény-értékek mellett teljesüljön az első két feltétel.

Valamely R reláció tranzitív lezártja, $R_T(X, X)$, egy egyszerű, három lépésből álló algoritmussal határozható meg, mely crisp és fuzzy relációk esetén egyaránt alkalmazható:

1. $\underline{R}' = \underline{R} \cup (\underline{R} \circ \underline{R})$.
2. Ha $\underline{R}' \neq \underline{R}$, akkor legyen $\underline{R} = \underline{R}'$, és folytassuk az első lépéssel.
3. Állj, ha $\underline{R}' = \underline{R}$, s ekkor $\underline{R}' = \underline{R}_T$.

Fontos, hogy az első lépésben alkalmazott kompozíció összhangban legyen a tranzitivitás definíciójával, tehát például max-min tranzitivitás esetén max-min kompozíció szükséges. Ebben az esetben $\underline{R}_T$ -t *tranzitív max-min lezárt*nak nevezzük. A következő példában is ezek a műveletek szerepelnek.

Legyen az R reláció az alábbi mátrixszal adott

$$\underline{\underline{R}} = \begin{bmatrix} 0,7 & 0,5 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0,4 & 0 & 0 \\ 0 & 0 & 0,8 & 0 \end{bmatrix}.$$

Alkalmazva az algoritmus első lépését,

$$\underline{\underline{R}} \circ \underline{\underline{R}} = \begin{bmatrix} 0,7 & 0,5 & 0 & 0,5 \\ 0 & 0 & 0,8 & 0 \\ 0 & 0 & 0 & 0,4 \\ 0 & 0,4 & 0 & 0 \end{bmatrix}, \quad \underline{\underline{R}} \cup (\underline{\underline{R}} \circ \underline{\underline{R}}) = \begin{bmatrix} 0,7 & 0,5 & 0 & 0,5 \\ 0 & 0 & 0,8 & 1 \\ 0 & 0,4 & 0 & 0,4 \\ 0 & 0,4 & 0,8 & 0 \end{bmatrix} = \underline{\underline{R'}}\text{-t}$$

kapjuk. Mivel a befejezési feltétel ($\underline{\underline{R'}} = \underline{\underline{R}}$) nem teljesül, ezért ismét az első lépéssel folytatjuk az algoritmust, $\underline{\underline{R}} := \underline{\underline{R'}}$ helyettesítéssel:

$$\underline{\underline{R}} \circ \underline{\underline{R}} = \begin{bmatrix} 0,7 & 0,5 & 0,5 & 0,5 \\ 0 & 0,4 & 0,8 & 0,4 \\ 0 & 0,4 & 0,4 & 0,4 \\ 0 & 0,4 & 0,4 & 0,4 \end{bmatrix}, \quad \underline{\underline{R}} \cup (\underline{\underline{R}} \circ \underline{\underline{R}}) = \begin{bmatrix} 0,7 & 0,5 & 0,5 & 0,5 \\ 0 & 0,4 & 0,8 & 1 \\ 0 & 0,4 & 0,4 & 0,4 \\ 0 & 0,4 & 0,8 & 0,4 \end{bmatrix} = \underline{\underline{R'}}.$$

Ugyanezt a műveletsort még egyszer elvégezve, az eredményként kapott mátrix már nem változik, tehát az $R(X, X)$ reláció max-min tranzitív lezártja:

$$\underline{\underline{R}}_T = \begin{bmatrix} 0,7 & 0,5 & 0,5 & 0,5 \\ 0 & 0,4 & 0,8 & 1 \\ 0 & 0,4 & 0,4 & 0,4 \\ 0 & 0,4 & 0,8 & 0,4 \end{bmatrix}.$$

4.4. Hasonlóság, kompatibilitás, fuzzy rendezések

A reflexív, szimmetrikus és tranzitív crisp relációkat — mint a 4.5. ábrán is láttuk — *ekvivalenciarelációknak* nevezzük. Az ekvivalenciarelációk az alaphalmazt ún. *ekvivalenciaosztályokra* particionálják, ugyanis minden X -beli x elemhez hozzárendelhető egy A_x halmaz, amelybe az x -szel relációban lévő elemek tartoznak:

$$A_x = \{y | \langle x, y \rangle \in R(X, X)\}.$$

A reflexivitás miatt x maga is eleme az A_x halmaznak, továbbá a szimmetria és a tranzitivitás következményeként A_x minden eleme relációban

van a halmaz többi elemével is. Az is megállapítható, hogy A_x -en kívüli elemmel egy A_x -beli elem sincs relációban. Az A_x halmaz az R reláció egy ekvivalenciaosztálya, melynek reprezentáns eleme x . Mivel minden X -beli elem pontosan egy ekvivalenciaosztályba tartozik, ezért ezek az osztályok a reláció alaphalmazának egy particionálását adják (melyet X/R -rel jelölünk).

A reflexív, szimmetrikus és tranzitív relációkat a fuzzy kontextusban *fuzzy ekvivalenciareláció*nak vagy *hasonlósági reláció*nak hívjuk. A crisp relációktól való megkülönböztetés végett a könyvben többnyire az utóbbi elnevezést használjuk.

A hasonlósági relációkat kétfajta megközelítés szerint lehet interpretálni. Az első alapján az elemeket crisp halmazokba csoportosíthatjuk úgy, hogy a halmazon belüli elemek közti reláció értéke egy adott küszöbértéket haladjon meg. Természetesen ha ez az érték 1, akkor crisp ekvivalenciarelációt kapunk.

Második lehetőség, hogy az X elemein egy kitüntetett $x \in X$ elemhez való hasonlóságot definiálunk. Ekkor minden $x \in X$ elemhez rendelhető egy fuzzy halmazként definiálható *hasonlósági osztály*, ahol az elemhez való hasonlóság mértékét a tagsági érték adja meg. Ez a definíció is az ekvivalenciareláció általánosításának tekinthető, hiszen ha egy osztályban minden elem 1 mértékben hasonló x -hez, míg más elemhez 0 mértékben, akkor egyben egy crisp ekvivalenciaosztályt kapunk.

A felbontási elv szerint (ld. 146. old.; 8.4.1. pont), minden fuzzy reláció α -vágatok uniójára dekomponálható:

$$R = \bigcup_{\alpha \in (0,1]} \alpha \cdot R_\alpha.$$

Az olvasóra hagyjuk annak az egyszerű állításnak a belátását, hogy ha R egy hasonlósági reláció, akkor R minden egyes α -vágata ($\alpha \in (0,1]$) egy crisp ekvivalenciarelációt ad (R_α). Minden α értékhez tartozó ekvivalenciareláció particionálást definiál X -en. Jelöljük $\pi(R_\alpha)$ -val az R_α ekvivalenciarelációhoz tartozó particionálást. Két elem nyilván akkor és csak akkor tartozik azonos partícióba, ha $R(x,y) \geq \alpha$. Minden hasonlósági relációhoz hozzárendelhető az általa indukált α -particionálások halmaza:

$$\Pi(R) = \{\pi(R_\alpha) | \alpha \in (0,1]\}.$$

melyek egymásba ágyazottak abban az értelemben, hogy $\pi(R_\alpha)$ a $\pi(R_\beta)$ particionálás finomítása, ha $\alpha \geq \beta$.

A hasonlósági osztályokat a fentebb leírt módon kaphatjuk meg a hasonlósági relációkból. Egy $R(X,X)$ reláció minden x eleméhez rendelhető egy az alaphalmazon értelmezett fuzzy halmaz, és minden $y \in X$ -re, a tagsági függvény értéke $R(x,y)$. Leszámítva a crisp ekvivalenciaosztály szélsőséges esetét, a hasonlósági osztályok fuzzyk, és így nem diszjunktak.

A hasonlósági osztályokat rendszerint tagsági mátrixokkal ábrázoljuk. Ha adott egy hasonlósági reláció, akkor egy tetszőleges elem hasonlósági mátrixa az eredeti mátrixnak az a sora, mely az adott elemhez tartozik.

Azokat a relációkat, melyek csupán reflexívek és szimmetrikusak (de nem tranzitívak), *kompatibilitási* vagy *tolerancia* relációnak, néha *szomszédsági relációnak* nevezzük.

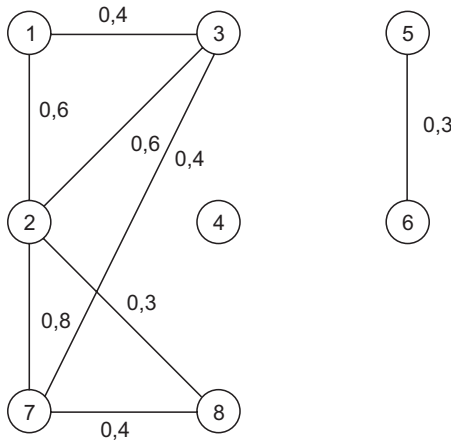
Fontos fogalom a kompatibilitási relációkkal kapcsolatban a *kompatibilitási osztály*. Legyen adott egy $R(X,X)$ tolerancia reláció. Ekkor az $A \subset X$ halmazt, melynek minden x,y elemére $\langle x,y \rangle \in R$ (tehát amelyen belül érvényes a tranzitivitás), kompatibilitási osztálynak nevezzük. Az ún. *legnagyobb kompatibilitási osztály* olyan tulajdonsággal is rendelkezik, hogy nem részhalmaza egyetlen más kompatibilitási osztálynak sem. Az R reláció legnagyobb kompatibilitási osztályainak családja az X (R által indukált) *teljes lefedése*.

Ha R fuzzy reláció, akkor a kompatibilitási osztályokat általánosabban, tetszőleges α értékre definiálhatjuk. Így az α -kompatibilis osztály, egy olyan részhalmaza a relációnak, amelyre

$$A(\alpha) = \{x,y \in A \mid R(x,y) \geq \alpha\}$$

fennáll. Hasonlóképpen az előző bekezdésben ismertetett crisp megfelelők értelemszerű általánosításaként adhatjuk meg a *legnagyobb α -kompatibilitási osztály* és a *teljes α -lefedés* fogalmait.

A kompatibilitási relációkat általában reflexív irányítatlan gráfokkal ábrázoljuk. A reflexivitás miatt minden csúcshoz tartozik egy hurokél (olyan él, mely a csúcsot önmagával köti össze), amit a gráf megjelenítésénél az egyszerűség és átláthatóság kedvéért elhagyunk ugyan, de úgy tekintjük, mintha ott lenne. Mivel a szimmetrikus reláció a kapcsolat meglétét mindkét irányban garantálja, a csúcsok közti élek irányítatlanok. Az élek mellett feltüntetjük a megfelelő tagsági értékeket (ld. 4.6. ábra).



4.6. ábra. Kompatibilitási reláció ábrázolása reflexív irányítatlan gráffal
(a hurokélek elhagyásával)

Példaként tekintsük az alábbi relációt:

$$\underline{\underline{R}} = \begin{bmatrix} 1 & 0,6 & 0,4 & 0 & 0 & 0 & 0 & 0 \\ 0,6 & 1 & 0,6 & 0 & 0 & 0 & 0,8 & 0,3 \\ 0,4 & 0,6 & 1 & 0 & 0 & 0 & 0,4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0,3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,3 & 1 & 0 & 0 \\ 0 & 0,8 & 0,4 & 0 & 0 & 0 & 1 & 0,4 \\ 0 & 0,3 & 0 & 0 & 0 & 0 & 0,4 & 1 \end{bmatrix},$$

melyet a 4.6. ábrán is megfigyelhetünk. Ez kompatibilitási reláció, mivel a mátrix szimmetrikus és minden főátlójában szereplő érték 1. A teljes α -lefedés a lényeges $\alpha > 0$ -szintekre $\Lambda_R = \{0, 0,3, 0,4, 0,6, 0,8, 1\}$, amint azt a 4.7. ábra mutatja.

Általában valamely kompatibilitási reláció α -lefedése nem képezi az alaphalmaz particionálását, noha ez természetesen előfordulhat. Ilyen például a 4.7. ábra relációja a 0,8 és 1 értékekre. Mivel éppen a tranzitivitás hiánya az, ami a kompatibilitási és hasonlósági relációkat megkülönbözteti egymástól, bármely kompatibilitási reláció tranzitív lezártja hasonlósági reláció lesz.

A harmadik jelentős bináris relációtípus, mellyel kiemelten foglalkozunk, a rendezések csoportja.

A reflexív, antiszimmetrikus és tranzitív crisp relációkat *részben rendezésnek* (vagy *parciális rendezésnek*) hívjuk. Jelöljük a részben rendezést $\prec$

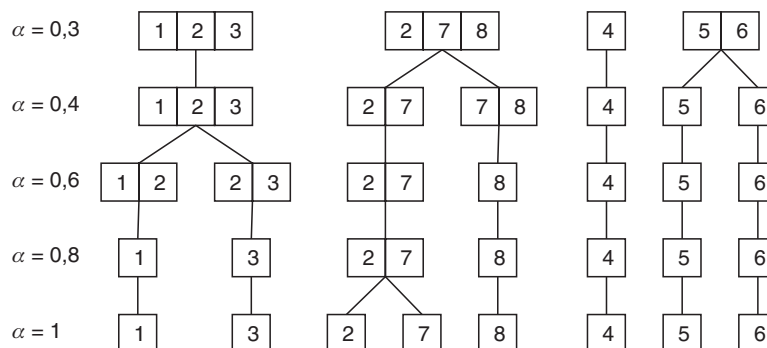
jellel, ahol $x \prec y$ azt jelenti, hogy x megelőzi y -t, és $\langle x, y \rangle \in R$. Az $R^{-1}(X, X)$ inverz részben rendezési relációt a $\succ$ szimbólummal jelöljük, ahol $y \succ x$ azt jelenti, hogy y az x rákövetkezője. Ha nincs olyan z elem, amire $x \prec z$ és $z \prec y$, de $x \prec y$ teljesül, akkor x az y közvetlen megelőzője, analóg módon, ha nincs olyan z , hogy $y \succ z$ és $z \succ x$, de $y \succ x$, akkor y az x közvetlen rákövetkezője.

Vegyük észre, hogy a részben rendezés tulajdonságai nem garantálják, hogy bármely két elemre az $x \prec y$ és $y \prec x$ reláció közül valamelyik is fennáll. Vannak olyan párok, melyben az x sem nem megelőzője, sem nem rákövetkezője az y -nak; az ilyeneket *nem összehasonlítható* pároknak nevezzük.

A parciális rendezéssel összefüggésben a következő alapvető fogalmakat vezetjük még be.

- Ha $x \in X$ és $x \prec y$ minden $y \in X$ -re, akkor x -et a $\prec$ szimbólummal jelölt reláció *első elemének* nevezzük.
- Ha $x \in X$ és $y \prec x$ minden $y \in X$ -re, akkor x -et a $\prec$ szimbólummal jelölt reláció *utolsó elemének* nevezzük.
- Ha $x \in X$ és $y \prec x$ -ből következik, hogy $x = y$, akkor x -et a $\prec$ szimbólummal jelölt reláció *minimális elemének* nevezzük.
- Ha $x \in X$ és $x \prec y$ -ből következik, hogy $x = y$, akkor x -et a $\prec$ szimbólummal jelölt reláció *maximális elemének* nevezzük.

Figyeljük meg, hogy valamely részben rendezésnek legfeljebb egy első, illetve utolsó eleme lehet, de minimális vagy maximális eleme több is. Ha létezik első/utolsó elem, akkor csak egy minimális/maximális elem van, és



4.7. ábra. Kompatibilitási reláció teljes α -lefedése

az megegyezik az első/utolsó elemmel. Valamely parciális rendezés első, illetve utolsó eleme az inverz relációnak rendre az utolsó, illetve első eleme.

Legyen adott az X halmaz és ezen egy R részben rendezés, A pedig legyen X részhalmaza: $A \subset X$. Ha $x \in X$ és $x \prec y$ minden $y \in A$ esetén, akkor x az A halmaz X -en való parciális rendezés szerinti *alsó korlátja*. Ha $x \in X$ és $y \prec x$ minden $y \in A$ esetén, akkor x az A halmaz X -en való parciális rendezés szerinti *felső korlátja*. Ha egy alsó korlát minden alsó korlátnak a rákövetkezője, akkor *legnagyobb alsó korlátnak* nevezzük, hasonlóan, ha egy felső korlát az összes többi felső korlátnak megelőzője, akkor *legkisebb felső korlátnak* hívjuk.

Az olyan rendezést, mely X minden kételemű részhalmazához tartalmaz legnagyobb alsó korlátot és legkisebb felső korlátot, *hálónak* nevezzük.

A (crisp) rendezésekkel kapcsolatos fogalmak ismertetése után térjünk rá ezek fuzzy megfelelőire. A reflexív, antiszimmetrikus és (valamilyen értelemben) tranzitív fuzzy relációkat *fuzzy részben rendezésnek* nevezzük. Tetszőleges max-min tranzitivitással rendelkező fuzzy részben rendezés felbontható crisp rendezésekre ugyanolyan módon, ahogy azt a hasonlósági relációknál láttuk: a reláció minden jelentős α -vágataként képzett crisp rendezés létrehozásával, melyek a fuzzy rendezés fokozatos finomítását adják.

Bármely fuzzy rendezés esetén az alaphalmaz minden x eleméhez két fuzzy halmazt rendelhetünk. Az elsőt x *domináló osztályának* nevezzük, melyet $R_{\geq[x]}$ szimbólummal jelölünk, s melynek értéke

$$R_{\geq[x]} = R(x, y), \quad y \in X.$$

Ebben a halmazban tehát a rendezés szerint megadott mértékben szerepelnek az x -et domináló elemek.

A második fuzzy halmaz az x *dominált osztálya*, melyet a $R_{\leq[x]}$ szimbólum jelöl

$$R_{\leq[x]} = R(y, x), \quad y \in X.$$

Ebben a halmazban a relációban megadott tagságfüggvény-értékkel szerepelnek az x által dominált elemek.

Az $x \in X$ elem *nemdominált*, illetve *nemdomináló* akkor és csak akkor, ha rendre $R(x, y) = 0$, illetve $R(y, x) = 0$ minden $y \in X$ -re.

Legyen X az R reláció alaphalmaza, s A ennek részhalmaza. Ekkor az A halmaz fuzzy felső korlátját az

$$U(R, A) = \bigcap_{x \in A} R_{\geq[x]}$$

összefüggéssel definiálhatjuk, ahol $\cap$ egy megfelelő fuzzy metszetet (t-normát) jelöl. Ha létezik az A halmaznak legkisebb felső korlátja, akkor az az $U(R, A)$ halmaz azon (egyetlen) eleme melyre,

$$U(R, A)(x) > 0 \text{ és } R(x, y) > 0$$

teljesül az $U(R, A)$ tartójának minden y elemére.

Legyen az R fuzzy részben rendezés az $X = \{a, b, c, d, e, f\}$ alaphalmazon az alábbi tagsági mátrixszal megadva:

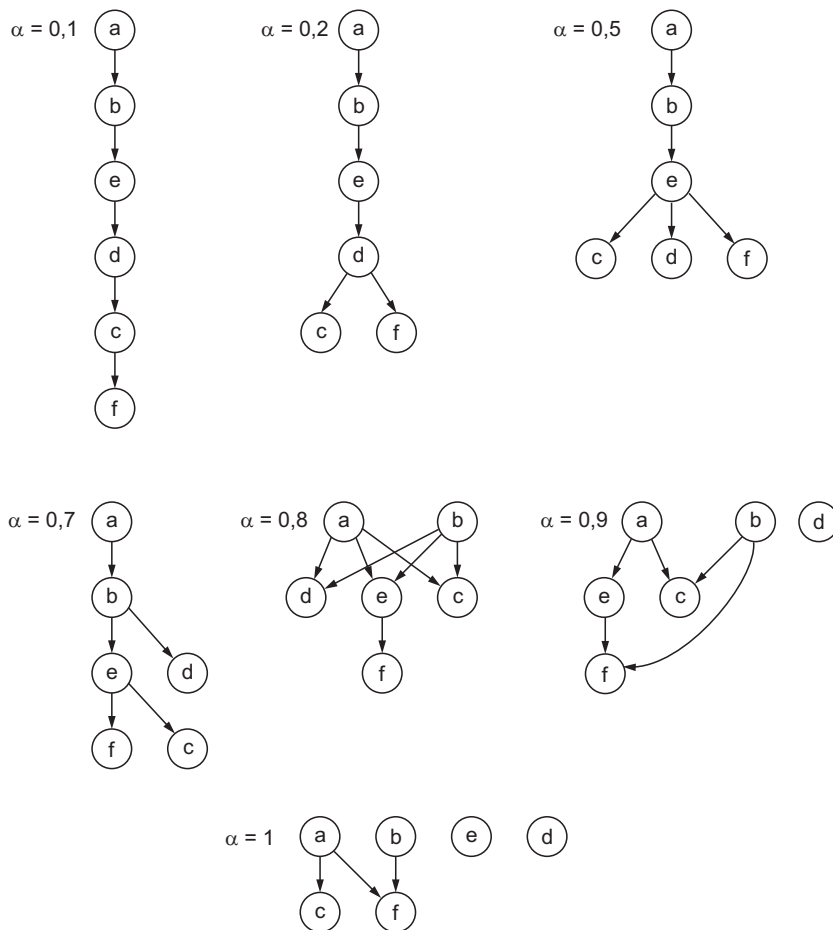
$$\underline{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0,7 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0,9 & 1 & 0,2 & 0,7 & 0 \\ 0,8 & 0,8 & 0 & 1 & 0,5 & 0 \\ 0,9 & 0,8 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0,1 & 0,2 & 0,9 & 1 \end{bmatrix}.$$

Az egyes elemek domináló osztályát a mátrixnak az adott elemhez tartozó sora adja. A mátrix oszlopai az elemek dominált osztályát határozzák meg. A példában szereplő mátrixban a nemdominált, f pedig nemdomináló elem. Az $A = \{c, b\}$ részhalmaz felső korlátja a c és b elemek domináló halmazainak metszeteként állítható elő:

$$U(R, \{c, b\}) = 0,7/a + 0,9/b.$$

Jelen esetben a ZADEH-féle metszetet alkalmaztuk. Az A halmaz legkisebb felső korlátja az a elem. A 4.8. ábrán az egyes α -vágatok által képzett crisp rendezéseket mutatjuk be. Megfigyelhető, hogy α növelésével a rendezés egyre gyengébb lesz.

A fuzzy relációk, valamint az e szakaszban tárgyalt hasonlósági, kompatibilitási és fuzzy rendezési relációk fogalmát elsőként ZADEH vezette be [206]. A bináris relációkat a fuzzy elméletről megjelent legelső monográfiában KAUFMANN tanulmányozta részletesen [83].



4.8. ábra. Fuzzy részbenrendezés α -vágatai

II. rész

Fuzzy irányítási rendszerek és alkalmazásaik

5. fejezet

A fuzzy irányítási rendszerek áttekintő bevezetése

A fuzzy logika és fuzzy halmazok elméletének megalkotása során ZADEH-t az az elgondolás vezette, hogy az igen bonyolult és analitikus módon nem modellezhető rendszerek algoritmikus értelemben kezelhető leírására találjon olyan eszközt, amelynek modelljéül az emberi gondolkodás, illetve a biológiai rendszerek szolgálnak. Nem véletlen az, hogy a fuzzy halmaz fogalmát ZADEH már 1965-ös híres cikke [205] előtt is felvetette, mégpedig rendszerelméleti, irányításelméleti munkáiban. A következő években, sőt évtizedben a fuzzy elmélet lassan fejlődött, és bizonytalan volt, hogy milyen területen sikerül először tényleges műszaki alkalmazásokat létrehozni.

1973-ban publikálta ZADEH azt a kulcsfontosságú tanulmányát [208], amelyben javasolta a nagy bonyolultságú rendszerek leíró modelljeiben a lingvisztikai, tehát természetes nyelvi változók fogalmának bevezetését, ahol a konkrét, precíz, számszerű érték helyett pontatlan, valamilyen tipikus megszerű érték környezetében fuzzy tagsági függvénnyel leírt, a tipikus értéktől távolodva egyre csökkenő tagsági értékű fuzzy számokkal, illetve általánosított fuzzy intervallumokkal — tehát konvex és normális fuzzy halmazokkal — modellezte az egyes értékeket. Ennek a megközelítésnek nagy előnye a korábban a mesterséges intelligenciában már használatos ún. szimbolikus logikai leírásokkal szemben, hogy míg ez utóbbiak meglehetősen precíz diszkretizálását igénylik az állapottérnek — hiszen ha a szimbolikus logikai értékek túlságosan nagy állapottérbeli hiperintervallumot jelentenek, akkor a modell pontossága jelentősen csökken —, a fuzzy modell esetében lehetőség van arra, hogy néhány tipikus értéket a fuzzy értékek magjaként feltüntetve, a közbenső területeken a magtól távolodva monoton csökkenő tagsági függvények (melyek egymásra részlegesen átlapolnak) valamilyen konvex kombinációjaként közelítsék az adott pontra jellemző tipikus értéket.

A ZADEH által javasolt megoldás a fuzzy halmazok és a már használatos

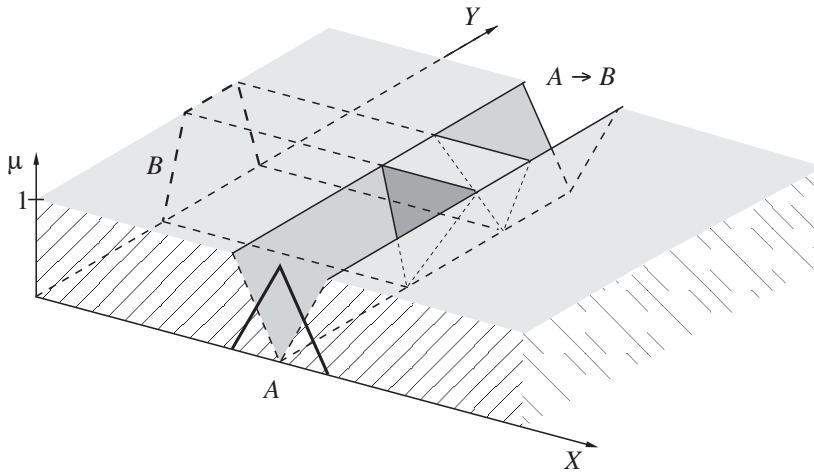
ha–akkor típusú szabályok kombinációja volt. A korábbi szimbolikus megközelítéshez képest a ZADEH-féle módszer komplexitáscsökkenést eredményezett, habár az is világos, hogy a leíráshoz szükséges szimbólumok számának redukciója csak valamilyen konstans faktorra történhetett; tehát amennyiben egy k dimenziós állapottérben modellezhető rendszer leírásához a szimbolikus megközelítésben $O(T^k)$ nagyságrendű szimbólumra van szükség, a fuzzy megoldásban $O((T/c)^k)$, azaz a redukció tényezője c^k . Az így megmaradó modellméret még mindig exponenciális az állapotváltozók számának függvényében.

A ZADEH-féle modell hátránya az volt, hogy a k -dimenziós állapottérben közvetlenül a tényleges k -dimenziós tér lehetőségeit kihasználó általános fuzzy relációkra vezette vissza a modellt. Nem sokkal ZADEH tanulmánya után, 1975-ben MAMDANINAK [130] sikerült olyan egyszerűsített modellt alkotnia, ahol a modellek kezelése az egyes dimenziókban függetlenül történhetett, ilyen módon drasztikusan csökkentve a számítási igényt. Ez ugyan a ZADEH-féle megközelítésnél kisebb rugalmasságot biztosított, mivel ebben a modellben a k -dimenziós fuzzy relációk helyett k számú egydimenziós relációvetület hengeres kiterjesztésének metszete által létrehozott speciális típusú relációk voltak csak megengedhetők.

MAMDANI ezt a projekciókon alapuló algoritmust sikeresen alkalmazta egy valós irányítási feladat megoldására. Vizsgálatait egy erősen nemlineáris gőzgépes–gőzkazános rendszeren végezte. Kísérletében az irodalom alapján rendelkezésre álló különböző nem hagyományos irányítástechnikai megoldásokat hasonlította össze, többek között a szimbolikus logikán és a fuzzy elméleten alapuló szabálybázisos szakértő jellegű irányítást. Az összehasonlító vizsgálat eredményeképpen a fuzzy modell adta a legjobb irányítást.

Ettől a pillanattól kezdődött a fuzzy irányítási rendszerek karrierje. MAMDANI eljárását a későbbiekben többen módosították. Így az egyik igen természetesen módosítás a LARSEN-féle algoritmus [125], mely megváltoztatta a következtetés végső lépését, a tényleges beavatkozás kiszámításának módját, később azonban SUGENO és TAKAGI egy olyan látszólag lényegesen különböző modelltypust javasolt [178], melyről azonban később KÓCZY kimutatta, hogy a MAMDANI-moddellel aszimptotikusan ekvivalens [92]. E modellek részletes ismertetését ld. a 7.3. és 7.5. szakaszokban.

Az itt megismert modellek a fuzzy elmélet művelői körében alapvetően kétféle elméleti interpretációt tettek lehetővé. Az egyik az ún. logikai interpretáció, amely egy ha–akkor típusú szabályt logikai implikációként értelmez. Ezzel a módszerrel a szabályok együttese tulajdonképpen az



5.1. ábra. Az $A \rightarrow B$ fuzzy szabály logikai implikációként való interpretációja

azonosan igaz logikai térben a szabályok által olyan korlátos területeket definiál, amelyeken belül az egyes implikációk hamis területeinek figyelembevételével, az azonosan igaznál kisebb, helyenként 0 igazságértékű, azaz hamis területek keletkeznek (ld. 5.1. ábra).

Ez az implementáció igen érdekes elméleti fejtegetésekre és tételbizonyításokra adott lehetőséget, ezek az elméleti megközelítések azonban nem magyarázták a konkrét MAMDANI-féle alkalmazást és az ezt követő, egyre nagyobb számban megjelenő valódi ipari alkalmazásokat. Lényeges nehézséget jelentett itt a MAMDANI-féle algoritmusban a min konjunkció alkalmazása — melyet egyes szerzők kezdetben MAMDANI-implikációnak neveztek —, nyilvánvaló azonban, hogy a konjukció nem rendelkezik az implikáció tulajdonságaival, így ez a művelet egyáltalán nem is értelmezhető implikációként.

A másik megoldás, mely ZADEH fejtegetéseit is felhasználja, a ha-akkor szabályokat úgy értelmezi, mint a bemeneti változók teréről a kimeneti állapotváltozók terére történő függvényyszerű leképezés egy-egy pontjának példászerű megadását. Ezek a pontok azonban nem a hagyományos értelemben vett térbeli crisp pontok, hanem „fuzzy pontok”, vagy akár fuzzy hiperintervallumok, és kiterjedésük a pozitív tagsági értékek figyelembevételével olyan, hogy ezek a szabálybázis szerint szomszédos pontok minden esetben részlegesen átlapolnak. Ezt a megközelítést ZADEH „fuzzy függvénygörbe” (vö. az 1.1. ábrát a 17., és a 7.2. ábrát a 111. oldalon) interpretációnak nevezte (fuzzy graph).

Ha a szabályokat eszerint értelmezzük, a szabálybázis grafikus képe egy nullában elhelyezett síkból kiemelkedő „hipergulákat” tartalmaz, természetesen az egy pontnál nagyobb kiterjedésű magú antecedensek (ha-részek, előzmények) esetében a gúla helyett csonkgúla alakú relációk keletkeznek. Ilyen módon a szabályok egymásba láncolódó (csonka)gúlaiból kialakított közelítőleg megadott fuzzy függvénygörbét rajzolnak le. A magokat összekötve megkapjuk azt a tényleges területet, amin belül a függvénygörbe tipikus értéke található. Az egynél kisebb, de pozitív tagsági értékek kevésbé tipikus, egyre kevésbé igaz területeken haladnak.

Ez az értelmezés nagyban segíti a ZADEH-, illetve MAMDANI-féle irányítási rendszerek műszaki alkalmazóit, mivel a szemlélettel nagyon jól összhangban áll. Ez a magyarázata, hogy különösebb irányításméleti ismeretek nélkül is lehetőség van fuzzy irányítási modellek megalkotására, mégpedig a szemlélet alapján olyan módon, hogy a tervező összerendelt, közelítőleg ismert bemenet–kimenet párok sokaságát valósítja meg, minden egyes közelítő bemenet–kimenet értékpárt egy fuzzy szabállyal reprezentálva. Amennyiben a reprezentáns pontokra vonatkozó információk pontatlannak bizonyulnak, igen könnyű a „fuzzy függvénygörbét” lokálisan módosítani, egy, vagy néhány, egy adott környezetben elhelyezkedő szabály antecedenisének és konzekvensének (akkor-rész vagy következmény rész) valamilyen mértékű alakváltoztatásával, illetve a mag helyzetének módosításával. Ilyen értelemben egy fuzzy szabálybázis egy statikus transzferfüggvény közelítő megadását is jelenti.

Megjegyzendő, hogy ugyanazzal a logikával, amellyel ZADEH a szimbolikus szabálybázison alapuló szakértő irányítási rendszerek helyett redukált bonyolultságú fuzzy irányítási rendszereket javasolt, tovább csökkenthető a bonyolultság, extrém esetben akár — az adott modell típuson belül maradva — a lehetséges minimumig, amely 2^k számú szabály (itt k továbbra is az állapotváltozók számát jelenti). A redukció megvalósítása a sűrű szabálybázisokról a ritka szabálybázisokra történő áttéréssel lehetséges. Ritka szabálybázisok esetén sem a MAMDANI-, sem a rokon LARSEN-, TAKAGI–SUGENO-féle, stb. szabályrendszerek és a hozzájuk kapcsolódó következtetési eljárások, illetve az irányító rendszerben alkalmazott következtető gépek nem alkalmazhatók. Ilyenkor sajátos, interpolatív következtető gépeket kell alkalmazni; az első ilyen következtetési módszereket KÓCZY és HIROTA javasolták ([95, 96]; ld. 8.4. szakasz), a későbbiekben ennek a módszernek számos általánosítása, illetve módosított változata készült el ([7, 102, 117, 180, 188]; részletesebben ld. a 8.5. szakaszt).

Mindazonáltal, ezen eljárásoknak a közös korlátját az jelenti, hogy egy

k állapotváltozós modell mindenképpen exponenciális, mégpedig k -adik hatvánnyal arányos bonyolultságú. A bonyolultság tovább csökkenthető, amennyiben lehetőség van a bemeneti állapotter valamilyen particionálására. Az itt alkalmazott megoldás a számítási algoritmusok körében ismert „oszd meg és uralkodj” (divide and conquer) eljárás alapötletén nyugszik olyan módon, hogy a modell állapotterét legalább két altérre particionáljuk, melyek direktszorzata adja a tényleges állapotteret. Az egyik altérben az állapotváltozóknak egy olyan csoportja szerepel, amelyek alkalmasak arra, hogy segítségükkel a modell további állapotváltozóit lokálisan redukáljuk; tehát a teljes állapotteret ebben az altérben particionáljuk, majd a partíció minden egyes elemében egymástól független, és lehetőség szerint a teljes állapotváltozó-készlethez képest csökkentett állapotváltozó-számú alszabálybázisok, azaz részmodellek alkothatók. Ezzel az eljárással a bonyolultság igen drasztikusan csökkenthető, hiszen az eredeti állapotváltozó-számhoz képest lényegesen kisebb hatványkitevőjű exponenciális bonyolultság is elérhető. Ezen az elven alapszik SUGENO vezetől nélküli helikopter kísérlete [172, 174], majd ezt követően több más sikeres irányítási és következtetési, döntéstámogatási alkalmazás.

A fuzzy irányítási rendszerek nagy előnye, hogy a modell közvetlenül bemenet–kimenet párok megfigyelésének segítségével állítható fel, és a kvázioptimális irányítási algoritmus hangolás segítségével állítható be. Természetesen az előnyök hátrányokkal járnak együtt: az ilyen modell mindig csak közelítő lehet, tehát olyan rendszerek esetében, ahol lehetőség van a pontos analitikus modell felállítására — és ennek, valamint ismert irányításelméleti tételek alapján az optimális irányításnak a meghatározására —, nem érdemes fuzzy megközelítéssel dolgozni, hiszen a fuzzy megoldás mindig szuboptimális lesz; adott esetben aszimptotikusan konvergálhat az egyébként analitikusan ismert optimumhoz.

Tehát a fuzzy irányítási rendszerek alkalmazásának területe elsősorban a vagy analitikusan nem ismert rendszerek modellezése és irányítása, vagy az olyan nagy bonyolultságú rendszereké, melyeknél az analitikus modell ugyan ismert, de a modell még numerikus módszerek alkalmazásával sem kezelhető valós időn belül.

Ezzel érintettünk egy általános filozófia jellegű problémát, amely a pontosság és a kezelhetőség egymáshoz való viszonyára vonatkozik. Általában jellemző az, hogy minél pontosabb egy közelítő modell, annál nagyobb a számítási bonyolultsága, azaz annál kevésbé kezelhető. Minél alacsonyabb a számítási bonyolultság, tehát minél kezelhetőbb egy modell, annál pontatlanabb, hiszen annál durvább a benne alkalmazott közelítés, legyen ez

hagyományos, nem fuzzy jellegű intervallumos, vagy pedig a magában interpolációs lehetőséget hordozó fuzzy halmazokkal történő közelítés.

Valamely probléma fuzzy irányítással való megoldhatóságát mindig az dönti el, hogy milyen a probléma eredendő bonyolultsága és milyen mérvű közelítés, az eredeti pontos rendszertől milyen mértékű eltérés engedhető meg az adott probléma még elfogadható megoldása során.

A fuzzy modell ötlete az emberi gondolkodás másolásán alapult, hiszen számos olyan feladat van, amelyet mind a mai napig nem sikerült megnyugtató módon számítógépesen vagy automatizáltan megoldani. Ugyanakkor egy esetleg nem is különlegesen képzett kezelő képes az adott probléma megoldására. Jól mutatja ezt a bevezetésben említett autóvezetési példa (ld. 13. oldal). Természetesen még a fuzzy irányító rendszerek sem érték el azt a szintet, hogy egy ilyen bonyolultsú feladatot képesek lennének megoldani. A vezetől nélküli helikopter irányítás sikeressége jól alátámasztja azonban, hogy ehhez hasonló feladatoknál a fuzzy irányítás alkalmazása sikerrel kecsegtethet, hiszen SUGENO kísérlete előtt semmilyen más megoldással nem sikerült ezt a nagyon komplex feladatot megoldani; a helikopter ugyanis a repülőgéphez képest sokkal több szabadsági fokkal rendelkezik, és emiatt sokkal bonyolultabb modellt kíván. Megjegyezzük azonban, hogy jelenleg még a SUGENO-féle helikopter irányítás sem került tényleges ipari alkalmazásra.

A hagyományos irányításelmélet szemszögéből természetesen felmerül egy sereg kérdés a fuzzy irányítási rendszerekkel kapcsolatban. E kérdések elsősorban arra vonatkoznak, hogy egy szabálybázissal adott modell és az ezen alapuló következtető eljárás eredményeképpen létrejött irányítási algoritmus a hagyományos irányításelmélet eszközeivel hogyan jellemezhető például a rendszer stabilitása szempontjából.

Ezen a téren ma már számos eredmény ismeretes, de korántsem áll rendelkezésünkre a válasz minden kérdésre. Tény, hogy a sikeres fuzzy irányítási alkalmazások egy jelentős része kísérleti hangoláson és elméleti megalapozottság nélkül, egyszerűen a stabil viselkedés megfigyelésén alapul. Szerencsére azonban a bevezetőben említett fuzzy függvénygörbés interpretáció olyan további gondolatokat vet fel, amelyek közelebb visznek a stabilitásvizsgálat teljes és elméletileg megalapozott elvégzésének lehetőségéhez. Amennyiben ugyanis a fuzzy szabálybázisokon alapuló modelleket úgy tekintjük, mint egy közelítőleg megadott bemenet–kimenet leképezési függvény valamilyen a tervezőmérnök számára intuitíve jól megközelíthető és megfogható megvalósulását, óhatatlanul felmerül az a kérdés, hogy milyen analitikus matematikai modell írná le ezen, a kezelő

szempontjából nagyon kellemes felülettel rendelkező rendszer viselkedését. Minden olyan fuzzy irányítóban, amely nem fuzzy megfigyelésen és nem fuzzy beavatkozáson alapul, tehát ahol a következtető gép fuzzy kimenetének meghatározását végső soron defuzzifikáció követi, lehetőség van az adott fuzzy irányító rendszer fekete dobozként, nem fuzzy, közelítő függvénygenerátorként való elemzésére.

Egyes, a gyakorlatban használatos egyszerű szabálytípusok (háromszög, trapéz, stb. alakú szabályok) és következtető rendszerek esetében az átviteli függvények explicit képletének meghatározása megtörtént [53, 108, 109]. Az így meghatározott függvényosztályok meglepő módon számos különböző fuzzy irányítási algoritmus esetében is hasonlónak bizonyultak. Amennyiben a tagsági függvények szakaszonként lineárisak, akkor mind a MAMDANI-, mind a LARSEN-, mind a TAKAGI-SUGENO-, mind pedig az interpolációs KH-féle eljárás család esetében viszonylag egyszerű, racionális tört függvényosztályt sikerül ilyen módon előállítani. Ez a függvényosztály nem túl jelentős számításigényű, és korlátozott közelítési tulajdonságokkal bír.

Amennyiben a szabálysorszám nem korlátos, természetesen ez a függvényosztály is univerzális közelítő tulajdonságú. Ezen a tényen alapulnak a fuzzy rendszereket univerzális eszközként, univerzális közelítőként tárgyaló matematikai eredmények. Abban az esetben azonban, ha a gyakorlati alkalmazásoknál ténylegesen felmerülő szabálysorszám korlátozásokat is figyelembe vesszük, az univerzális közelítési tulajdonság elvész, ugyanis az ilyen racionális törtfüggvények osztálya matematikai értelemben véve sehol sem sűrű a közelített (folytonos) függvények terében, tehát ezek a függvények pontos közelítésre nem alkalmasak. Szerencsére a gyakorlati feladatok nagy részénél nem is cél a minden határon túli pontosságú közelítés, hanem csupán egy olyan viszonylag jó reprezentáció, amely az adott feladatot reális méretben, szuboptimális módon oldja meg.

Megjegyzendő, hogy sok gyakorlati feladatnál elvileg sem létezik az a függvény, melyet az adott szabálybázissal igyekszünk közelíteni, hanem létezik a valóságban a függvényeknek egy végtelen elemből álló családja, melyekből egy jó szabálybázis egy jó reprezentációt kiválaszthat, amely rendelkezhet az adott függvény nyaláb minden lényeges tulajdonságával. Az explicit függvények közelítési tulajdonságaival, illetve általában a fuzzy rendszerek által generált függvényosztályok matematikai tulajdonságaival a 7.6. és 7.7. szakaszokban fogunk foglalkozni.

Érdekes tény, hogy ez a korlátos szabálysorszám esetén viszonylag kedvező közelítési tulajdonságokkal rendelkező függvényosztály adott esetben

igen előnyös matematikai tulajdonságokkal bír. Itt elsősorban az adott interpoláció matematikai stabilitására, vagy — más oldalról közelítve — érzékenységére utalunk [80, 180]. E függvényosztályok, ld. például a KH-interpolációt, ugyanis olyan értelemben matematikailag stabilak, hogy a bemeneti értékek kismértékű, azaz korlátos megváltozása a kimeneten is csak kismértékű, azaz korlátos változást idéz elő. Valamely konkrét modell esetén kompakt intervallumon ezek a korlátok a teljes univerzumra érvényesen megadhatók. E kérdéskörrel szintén foglalkozunk röviden (ld. 8.4.2. pont, 152. old).

6. fejezet

Tudásbázis-alapú szakértő rendszerek

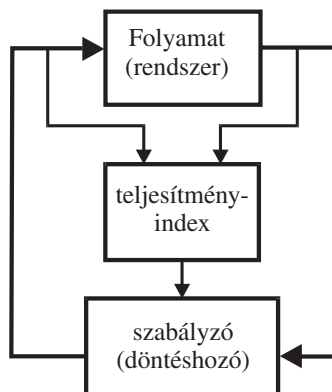
6.1. Hagyományos irányítási és szakértő rendszerek

Mielőtt a fuzzy szabályalapú következtetési rendszereket részletesen tárgyalnánk, világos képet kell alkotnunk arról, hogy mindennapos, de nehezen algoritmizálható és gépiesíthető, számítási értelemben igen bonyolult feladatokat — mint például az autóvezetés, törékeny tárgy mozgatása, vagy akár ismerősünk arcának felismerése — hogyan old meg az ember, és milyen hagyományos automatizált eljárások ismertek e témakörben [145]. Bár a felsorolt feladatok egyszerűnek tűnnek, mégis állandó kihívást jelentenek a mesterségesintelligencia-rendszerek tervezőinek, hiszen az ilyen berendezések teljesítménye és képessége messze elmarad egy átlagos emberétől.

Egy bizonyos feladat megoldása során, mint például egy mozgó akadály kikerülése, az adott szituáció megoldásához szükséges összes rendelkezésre álló információt összegyűjtjük, így például: a terep topológiáját, az akadály adott helyzetben fontos jellemzőit (méretek, sebesség, a mozgás iránya). Ezen adatok és a hasonló szituációkkal kapcsolatban meglévő tapasztalatok felhasználásával következtetési lépések sorozatát hajtjuk végre, amellyel megfelelő algoritmus esetén elérjük a kitűzött célt. Ezt a módszert az alábbiak szerint lehet modellezni. Minden egyes következtetési lépésnek (érvelésnek) egymástól gyakorlatilag független művelet felel meg. Ha van visszacsatolás az irányított rendszer és az irányító személy között, valamint ha a rendszer működéséről rendelkezésre áll elegendő információ, akkor a végső célt irányítási lépések véges sorozatával elérhetjük (ld. 6.1. ábra).

A vázolt irányítási modell automatizálását úgy valósíthatjuk meg, ha helyettesítésére olyan egységet hozunk létre, mely képes az összes számottevő irányítói következtetés meghozatalára.

Szakértő rendszernek nevezzük az olyan számítástechnikai rendszereket, melyek az emberi szakértő következtetési folyamatát emulálják va-



6.1. ábra. Zárthurkú irányítási rendszer vázlata

lamely jól behatárolt szakmai területen. A szakértő rendszerek megalkotásának elsődleges célja az volt, hogy egyes szakterületek szakértőinek tapasztalatát, hozzáértését és problémamegoldó-képességét elérhetővé és érthetővé tegyék az adott területen tapasztalattal nem rendelkezők számára is. E rendszerek a szakértelem megismerésén kívül többek közt konzultációs, diagnosztizáló, döntéstámogató, tanulási, tervezési, vagy kutatási tevékenységek támogatására is alkalmazhatók.

A szakértő rendszerek gondolata egyébként a fuzzy szakértő rendszerekénél sokkal régebbi, és a klasszikus mesterséges intelligencia kutatáshoz kötődik. A szakértő rendszerek általában ha–akkor típusú szabályokból felépített tudásbázist alkalmaznak, ahol a szabályokban szereplő logikai szimbólumok lényegében a BOOLE-algebrai struktúrát követő logika alapján állnak. Ennek megfelelően egy ha–akkor típusú szabályt implikációként lehet értelmezni. Tehát „ha x az A , akkor y a B ” értelmezése A implikálja B -t ($A \rightarrow B$). Az ilyen szakértő rendszerekben a BOOLE-algebra ismert azonosságai, illetve a már évszázadok óta ismert formális logikai tautológiák segítségével lehet következtéseket levonni. Legismertebbek ezek közül a bevezetőben már említett (15. old.) *modus ponens*, a *modus tollens* és a *hipotetikus szillogizmus*, illetve ezek tetszőleges kombinációja.

Az ilyen szakértő rendszerek hátránya az, hogy az alkalmazott szimbólumok a modellezett jelenséghez nem illeszkednek jól. Mivel az irányítási feladatok jelentős részében olyan változókkal kell dolgozni, amelyek folytonos értékkészletűek és analóg jellegűek, ezeknek az értékeknek formális leírására végtelen sok szimbólumra volna szükség. Ez természetesen lehetetlen, ezért a folytonos értékkészletet diszkrét intervallumokra osztják fel, és minden ilyen intervallum egy-egy szimbolikus nevet kap. Az inter-

vallumok valamilyen értelemben vett legtipikusabb reprezentáns értéke szerepel a szabályokban.

Természetesen ugyanilyen probléma merül fel az eredendően nem irányítási, hanem valamilyen emberi szakértői döntést igénylő területeken is. Ilyen például az orvostudomány, ahol a diagnosztizálás alapja számos olyan megfigyelés, amelyek egy jelentős része szintén folytonos értékkeszletű változók mérésén vagy becslésén alapul; ilyen például a vérnyomás, testhőmérséklet, stb.

Ugyanilyen problémák adódnak a gazdasági döntéseknél is, ahol tulajdonképpen nagyobb mérvű képletes formalizálásra nyílna lehetőség, de a vizsgált rendszerek általában igen bonyolultak és igen sok változótól függnek, ezért az emberi intuíció szerepe kiemelten fontos ezen a területen is.

A fuzzy szakértő rendszerek nagy előnye a klasszikus szakértő rendszerekkel szemben, hogy itt nincs szükség olyan nagy számú szimbólum használatára, hanem az egyes szimbólumokhoz tagsági függvények rendelődnek, amelyek a szimbólumhoz rendelt tipikus értéktől való távolodásnak megfelelően egyre kisebb igazságértéket hordoznak. Ezek minden esetben konvex és normális fuzzy halmazok, általában fuzzy számok vagy fuzzy intervallumok. Az ilyen szakértő rendszerek a megfigyelt jellegzetes pontokon a tudásbázis alapvető elemeit alkotó szabályokból állnak, ezek között pedig a részlegesen átlapoló tagsági függvények figyelembevételével interpolációs jellegű közelítés történik.

Feltehető, hogy ez a fajta interpolatív közelítés jellemző az emberi gondolkodásra is. Nehezen feltételezhető ugyanis, hogy például egy diagnosztizáló orvos agya olyan mennyiségű adatot raktározna, amely a praxisában előforduló minden — vagy csaknem minden — esetet külön szabály formájában tartalmazna. A diagnózis általában analógiás interpolatív módon történik: jellegzetes, hasonló, vagy valamilyen értelemben közrefogó példák segítségével sikerül meghatározni, hogy az adott tünetegyüttes milyen betegséget takar, illetve milyen kezelést igényel.

Az automatikus, következtetés-alapú irányítási rendszer és a klasszikus zárthurkú irányítás összehasonlításához tekintsük át először ez utóbbi rendszerek tervezési nehézségeit. Az összes hagyományos irányító rendszer tervezési stratégia az alábbi két lényeges feltételezésen alapszik:

- Az irányított rendszer ismert. A rendszert valamely modellje segítségével reprezentáljuk (identifikáljuk), amelynek létrehozásához szükség van a rendszerről rendelkezésre álló összes lényeges információra. Ekkor a rendszer kimeneti válasza a modell alapján tetszőleges bemenet

esetén kiszámolható.

Az *identifikációs* fázis a rendszer későbbi helyes működése szempontjából alapvető fontosságú.

- Az irányítás függvénye tömör matematikai formulák formájában adott, melyek tartalmazzák a rendszer változó paramétereit. (Ezt az információt nevezzük a rendszer *teljesítményindexének*.)

Ha a fenti feltételek teljesülnek, akkor az adott rendszer modellje a klasszikus irányításelmélet módszereivel megalkotható, és meghatározható a működését irányító optimális rendszer, illetve kiszámíthatók ez utóbbi paraméterei. A rendszer modellje segítségével végezzük az irányítás optimalizálását, vagyis a rendszer kimenete és a célfüggvény által generált elméleti optimum közötti eltérés minimalizálását.

Abban az esetben azonban, ha a modellezett rendszer túl bonyolult (például erősen nemlineáris), vagy modellje eleve ismeretlen, az irányításelmélet hatékony és elegáns módszerei és matematikai háttere nem használható, az irányítás alkalmazásának feltételei megvalósíthatatlanná válnak. Ha a rendszer nemlineáris, jellege nem stacionárius, vagy ha a rendszer működését leíró adatok hiányoznak, akkor modellje általában nem alkotható meg pontosan. Ekkor a rendszeridentifikáció rendelkezésre álló algoritmusai, melyek többek közt statisztikai módszereken, tapasztalati megfigyeléseken és többváltozós függvényoptimalizáción alapulnak, nem, vagy csak megszorításokkal alkalmazhatók.

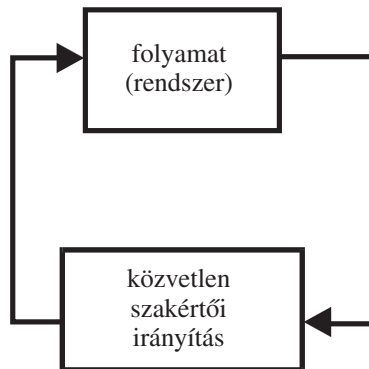
A bonyolult rendszerek másik gyakori problémája az, hogy a létrehozott modell túlságosan is pontos, túlzottan specifikus, s így a modellt leíró egyenletek bonyolultsága és a bennük szereplő paraméterek száma kezelhetetlenül magas. Ezt a jelenséget nevezi SCHWEPPE a „túlmodellezés hibájának” [156]. Továbbá, ahogy ZADEH is rámutatott [207], az irányításelméletben megfigyelhető az irányítási modellek „matematizálódási” trendje. Bonyolult rendszereknél nem sikerül meghatározni, hogy milyen optimalizálási stratégia szerint működjék az irányítás.

Szintén nehezen megoldható az a szakértő rendszereknél előforduló hasonló szituáció, mikor a szakértő a feladatot közel optimálisan végre tudja hajtani, ám a végrehajtás folyamatát és az alkalmazott (kvázi-) optimalizálási eljárást nem tudja megindokolni, és így a folyamat jellegzetességeit nem képes megismertetni, az automatizálást nem tudja elősegíteni.

Fontos megjegyezni, hogy ha képesek volnánk a szakértő kezelő irányítási protokollját automatizálni, akkor ezáltal két alapvető az irányítástechnikában jelentkező problémát is ki tudnánk küszöbölni: a rendszeridenti-

fikálás és modellalkotás időigényes lépését, valamint a teljesítményindex explicit matematikai formulákban való megadását.

Ha feltesszük, hogy a folyamat ismerete és a teljesítményindex magában az irányítási protokollban van elrejtve, akkor a probléma implicit módon megoldható. A gyakorlatban ugyanis a szakértő kezelők akkor is képesek ésszerű irányítási döntéseket hozni, ha a rendszer karakteristikája időben változó, nemlineáris, vagy zaj lép fel. Ezt felfoghatjuk úgy is, hogy a rendszer ismerete és a teljesítményindex az irányítási protokollban implicit módon, „kódoltan” jelenik meg. Ez a módszer jelentős előnyökkel jár, különösen az irányítási céljának meghatározásánál, ugyanis a helyes teljesítményindex specifikálása olykor bonyolultabb feladat még a rendszeridentifikációs eljárásnál is. Ezen módszeren alapuló gyakorlati megoldásokban a teljesítményindex kompromisszumot képez az irányítás-elmélet valódi követelményei és az egyszerűen megvalósítható irányítási stratégia között. (Például lineáris modellek esetén kvadratikus alakú teljesítményindex is elfogadható, amelyből analitikus módszerekkel a megfelelő formulák meghatározhatóak.)



6.2. ábra. Közvetlen tudásalapú szakértő rendszer vázlata

A felsorolt okok teszik a tudásbázis- (vagy automatikus következtetés-) alapú szakértő rendszereket hatékonyvá sok alkalmazásban. Bizonyos esetekben a szakértő rendszerek képesek az emberéhez hasonló döntések meghozatalára, és az emberi irányítási protokoll megközelítésére. A szakértő rendszer tudásbázisa a rendszer működését ismerő, azt sikeresen irányítani képes operátor gyakorlati tapasztalatainak segítségével megalkotott irányítási stratégiák formalizálásával valósítható meg.

Ha a tudásbázis alapú szakértő rendszer közvetlenül helyettesíti az irányítási körben az irányító modult (vagy az emberi segítséget), akkor

közvetlen szakértő rendszerről [72] beszélünk (6.2. ábra).

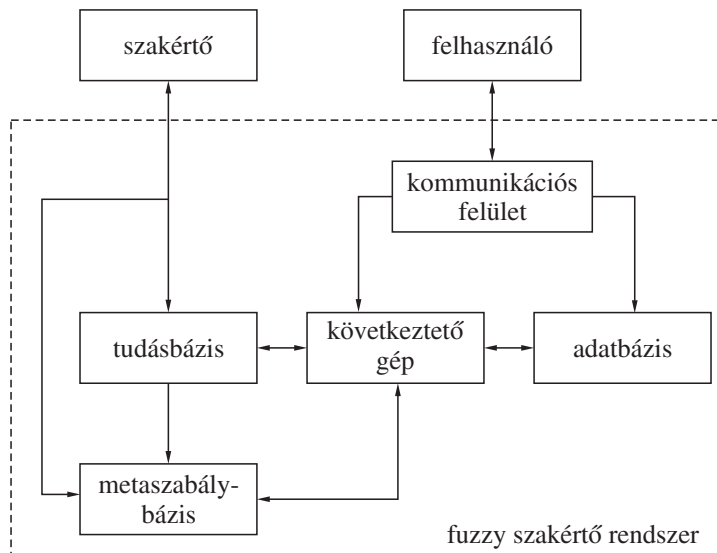
A szakértő rendszerek irányítási algoritmusának implementálása mindazonáltal további problémákat vet fel. Először is olyan irányítási protokollt kell létrehozni, mely megvalósítja az irányítási stratégia főbb tulajdonságait. Másodsor, a hatékonyság növelése érdekében olyan eszközre van szükség, mely egyrészt elég rugalmas ahhoz, hogy képes legyen az irányítási protokoll nyelvi fogalmaival operálni, másrészt elég pontos ahhoz, hogy számítógépen implementálható legyen. A következő fejezetben bemutatásra kerülő fuzzy irányítási rendszer eszközt nyújt erre a célra, mely reprezentálni képes a pontos határok (definíció) nélküli nyelvi fogalmakkal kifejezett következtetéseket, és így megfelelő formális keretet biztosít az imént megfogalmazott követelmények ötvözésére.

6.2. Fuzzy szakértő rendszerek

A fuzzy szakértő rendszerek szerkezetét és főbb összetevőit illusztrálja a 6.3. ábra.

A szakértő rendszer lényegét a *tudásbázis* (hosszú távú memória), az *adatbázis* (rövid távú memória) és a *következtető gép* alkotja.

A tudásbázis tartalmazza a problémakörrel vagy szakterülettel kap-



6.3. ábra. Fuzzy szakértő rendszerek szerkezeti vázlata

csolatos általános információkat. Fuzzy szakértő rendszerek esetén ezt az információt *fuzzy produkciós szabályokkal* adjuk meg, melyek többnyire ha–akkor alakban teremtenek kapcsolatot a feltételek (antecedensek) és következmények (konzekvensek) között. A szabályok általános alakja „Ha A , akkor B ”, ahol A és B a bemeneti és kimeneti univerzumok fuzzy halmazai.

Az adatbázis célja a szakértő rendszer bizonyos feladataival kapcsolatos adatok tárolása, melyet például a rendszer a felhasználóval való kommunikáció során szerez meg. Ezek jellemzően az adott feladat végrehajtásához szükséges paraméterértékek.

A következtető gép (egység) a rendelkezésre álló tények (adatok) és a fuzzy produkciós szabályok felhasználásával fuzzy következtetéseket hoz. A produkciós szabályok kiértékelése két típusú lehet: vagy adat-vezérelt, amikor a megadott adatok és a produkciós szabályok feltételrészének illesztésével a rendszer az összes lehetséges következtetést előállítja; vagy célvezérelt, amikor a cél és a produkciós szabályok következményrészének illesztésével keres a rendszer olyan tényeket (megfigyeléseket), melyek az adott állapotban fennállnak. Az adatvezérelt módszer előrehaladó, a célvezérelt pedig hátrafelé haladó következtetéseket végez. Időigény szempontjából az utóbbi módszer előnyösebb, mivel csak a célhoz vezető szabályokat értékeli ki.

A következtető egység a szabályok alkalmazási sorrendjére vagy a szabályok kiválasztására *metaszabályokat* is felhasználhat, melyek leállási feltételeket, szabályok közötti (esetleg állapottól függő) precedenciákat, vagy a felhasználóval történő kommunikációt határozzák meg. A *metaszabálybázis* alapvető célja, hogy a felesleges szabályok alkalmazását elkerülve egyszerűsítse a rendszer működését. A bonyolultabb rendszerek modellezéséhez hierarchikus szabálybázis és következtető gép alkalmazása szükséges. Ilyen rendszerekben kiemelt jelentőségű a metaszabálybázisok szintje, mely a számítási bonyolultság csökkentésében alapvető szerepet játszik.

A *kommunikációs/magyarázó felület* a felhasználó és a rendszer kapcsolatát szolgálja, például a konklúzióhoz vezető következtetési szabályok sorozatának megadásával segítheti a felhasználót a szakértő rendszer működésének megértésében.

A fuzzy szakértő/irányító rendszereknek jelentős irodalma van. Ezek közül kiemeljük a legfontosabb könyveket, melyekben további nagyszámú hivatkozás található folyóiratokban és konferencia-kiadványokban megjelent cikkekre: [66, 82, 121, 140, 189].

A következő fejezetben részletesen tárgyaljuk a fuzzy irányítási rendsze-

reket, melyek a fuzzy szakértő rendszerek legelterjedtebb és legsikeresebb alkalmazását jelentik.

7. fejezet

Fuzzy irányítási rendszerek

7.1. A fuzzy irányítási rendszerek felépítése

A *fuzzy irányítási rendszerek* (vagy röviden *fuzzy irányítók*) lényegi eleme a *szabálybázis* alapú modell. Ez a modell „ha a bemenet A , akkor a kimenet B ” (A és B fuzzy halmazok) típusú *szabályokból* áll. Az egyszerű modellek, mint a ZADEH- vagy MAMDANI-féle, általában homogén szabálytípusból épülnek fel, és bár létezik a szabályoknak adott esetben tömörített változata, amely esetleg egyes bemeneti állapotváltozókat egy-egy szabályban kiküszöböl, ezek a szabályok azonban tulajdonképpen több elemi szabály egyesítéséből keletkezett kumulatív szabályoknak tekinthetők. A bonyolultabb hierarchikusan strukturált szabálybázisokban, ahol az állapottér partíciója is megvalósul a szabálybázisokat felépítő alszabálybázisok strukturálisan is különbözhetnek. Ez azonban természetes, mivel az egyes alszabálybázisokban a leíráshoz szükséges állapotváltozók száma és jellege eltérhet egymástól. Szintén különbözik a partíciót leíró ún. meta-szabálybázis. Elvileg lehetséges a kettőnél több fokozatú vagy többlépcsős szabálybázis megalkotása is, ilyenre azonban eddig a gyakorlatban még nem került sor.

A szabálybázis szerkezetileg hasonlít a szimbolikus, mesterséges intelligenciában használatos szakértő szabálybázisokra, lényeges különbség azonban, hogy a szimbólumok mellett szubszimbolikus információkat is tartalmaz, mégpedig az egyes szimbólumokhoz rendelt *fuzzy tagsági függvények* formájában. Hasonló mondható el a neurális hálózatokon alapuló modellek esetéről is, ahol az egyes neuronok *gerjesztési függvényei* hordoznak hasonló szubszimbolikus információt.

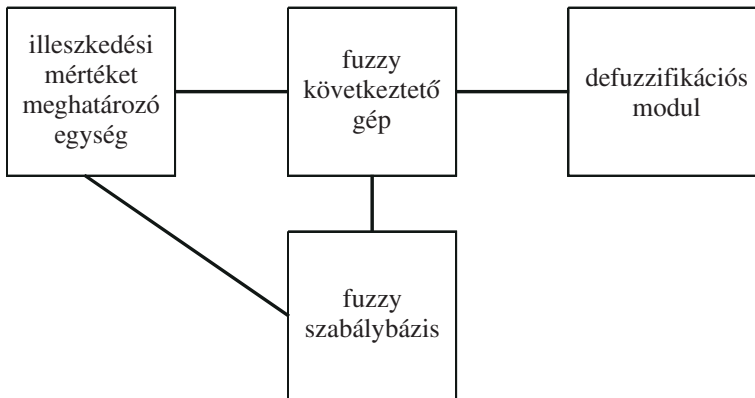
A fuzzy irányítási rendszerek további összetevője az *illeszkedési mértéket meghatározó egység*, amely lényegében hasonló módon működik mind fuzzy, mind pedig nem fuzzy, azaz crisp bemenetek esetében. Ez az egység a szabálybázis antecedens elemeit hasonlítja össze az aktuális megfigyelés tagsági függvényével vagy konkrét értékével, és a *tűzelő szabályoknál* — tehát azon szabályoknál, ahol az antecedens rész metszete a megfigyeléssel nem

üres — meghatároz egy 0 és 1 közötti fuzzy illeszkedési mértéket. Általános esetben ez nagy számú tüzelő szabályhoz tartozó illeszkedési mértéket fog megadni, melyek ismeretében a következtető gép a szabálybázis tüzelő szabályainak konzekvens részeit értékeli ki az illeszkedési mérték valamilyen módon történő figyelembevételével, melyek a konzekvens részeknél súlytényezőként szerepelnek, és a tüzelő szabályok súlyozott, illetve módosított konzekvenszei, azaz akkor-részei kerülnek be a következtető gépbe.

A fuzzy irányítási rendszereket alkotó harmadik egység a *következtető gép*. A következtető gép lényege, hogy az illeszkedési mérték meghatározása után a kapott súlyokat a fuzzy szabálybázisban található tüzelő szabályok konzekvenszeivel (általában egy konjunkció segítségével) kombinálja. A MAMDANI-módszer a minimum, a LARSEN-módszer pedig az algebrai szorzat konjunkcióját alkalmazza. Értelemszerűen a TAKAGI-SUGENO-szabályoknál ez a kombinálás más módon történik, hiszen ott a konzekvens oldalon nem fuzzy tagsági függvények, hanem kimenetbemenet közötti tényleges crisp függvények szerepelnek. A következtető gép kimenete MAMDANI-, LARSEN- és hasonló eljárásoknál, beleértve az interpoláción alapuló módszereket is, valamilyen általában nem konvex és normális fuzzy tagsági függvény formájában jelenik meg. Kivétel ezalól a TAKAGI-SUGENO-, és az ennek speciális esetét alkotó SUGENO-irányító, ahol a konzekvenssek eleve defuzzifikált formában vannak megadva.

A fuzzy irányítóknál szükség van arra, hogy valamilyen konkrét crisp beavatkozó érték jelenjék meg, amely a következtető gép kimenetén előálló fuzzy tagsági függvény defuzzifikálásával történik. Ezért a fuzzy irányító rendszerek negyedik alkotóeleme a *defuzzifikáló egység*, amely a következtető gép eredményeként kapott fuzzy tagsági függvény valamilyen értelemben vett legjellemzőbb, legtipikusabb elemét, vagy középértéket választja ki. A defuzzifikálásra számos eljárás ismert. A gyakorlatban leginkább alkalmazott módszerek közül egyes megoldások a mag középítő vagy szélső tipikus értékét választják ki, más technikák pedig a tagsági függvény alatti területnek a középpontját, illetve — a függvény alatti területet egy mechanikai lemeznek felfogva — a súlypontját adják meg. A defuzzifikációs technikákat a későbbiekben részletesen elemezzük (ld. 7.4. szakasz). A fuzzy irányítók négy alkotóelemét ábrázolja a 7.1. ábra.

Megjegyzendő, hogy a nem irányítási célra alkalmazott olyan következtető vagy döntéstámogató rendszerekben, amelyek az előbb leírtakkal lényegében azonos struktúrájúak, ám a kimenet emberi kezelő számára készül, nem szükséges a defuzzifikáció, hiszen sokszor a kapott fuzzy tagsági függvény informatívabb egyetlen konkrét crisp középértéknél. Ebben



7.1. ábra. Általános fuzzy irányítási rendszer vázlata

az esetben is előfordulhat azonban, hogy a kapott szabálytalan alakú nem konvex és nem normális függvényt egy hozzá lehetőség szerint minél közelebb álló — valamilyen szabványos készletből választott, vagy pedig legalább előírt tulajdonsággal rendelkező, tehát mindenképpen konvex és normális, esetleg trapéz vagy háromszög alakú, stb. — legjobban közelítő tagsági függvényre cseréljük. Ilyen esetben a defuzzifikáló egység helyett *lingvisztikus közelítő egység* vagy *CNF függvénygenerátor* szerepel.

Megjegyzendő továbbá, hogy egyes szakirodalmi munkák az illeszkedésimérték-generálót fuzzifikáló egységnek hívják. Ez az elnevezés nagyon félrevezető, mivel azt sugallja, hogy a fuzzifikálás és a defuzzifikálás egymásnak inverz műveletei. Ez a valóságban nem áll fenn, a két művelet egymással semmiképpen sem rokonítható. Ez például abból is könnyen látható, hogy míg az illeszkedési mértéket meghatározó egység bemenete minden esetben lehet egyszerre fuzzy vagy nem fuzzy érték, addig a defuzzifikáló egység mindig crisp kimeneti értéket generál. Az illeszkedésimérték-meghatározó a szabálybázis elemein mint univerzumon generál egy fuzzy halmazt, a defuzzifikáló viszont az eredeti univerzum fuzzy halmazát alakítja át.

A bemenet és a kimenet szimmetrikus vagy inverz viselkedésének feltételezésén alapul a KWONG-féle tagsági függvény nélküli fuzzy irányító [123], mely ugyan matematikailag korrekt összefüggéseket használ, de alkalmazása során éppen a fuzzy irányítók legelőnyösebb tulajdonsága, a könnyű ember-gép kommunikáció, illetve a kellemes ember-gép interfész vész el.

7.2. A fuzzy irányítási rendszerek alkotóegységei

7.2.1. A szabálybázis szerkezete

A fuzzy következtető rendszerek a szakértő irányítási rendszerek egy típusát alkotják, és így alkalmasak arra, hogy szakértőktől származó információt építsenek be a tudásbázisuk által reprezentált modellbe. Ennek hatalmas jelentősége van olyan irányítási problémáknál, melyek matematikai modellje bonyolult vagy egyáltalán nem ismert, vagy ahol a szükséges hagyományos irányítási rendszer használata nehézkes, illetve drága. Ezek a nehézségek általában a folyamat nemlinearitására, időben való változására, a környezeti tényezőkben fellépő jelentős zavarokra — melyek akadályozzák a pontos és hiteles mérések elkészítését — vagy más tényezők által kiváltott okokra vezethetők vissza. Az a tapasztalat azonban, hogy szakértő operátorok még ilyen körülmények között is képesek a rendszer hatékony irányítására.

A gyakorlott operátortól kapott információk összessége a folyamat irányításelméleti modelljének alternatívájaként használható. Annak ellenére, hogy ennek az ismeretanyagnak a pontos matematikai fogalmakkal való kifejezése szintén gondot okoz, mégis könnyebben leírhatók az irányítás lépései nyelvi fogalmak segítségével.

A *tudásbázis analízisének* nevezzük a szakértőnek az irányított folyamatra vonatkozó ismereteinek rendszerezését és kiértékelését [63], melyre több, széles körben elterjedt módszer ismert.

- Az ún. közvetlen eljárás, ha a rendszert „manuálisan” irányítani képes szakértő nyelvi szabályok formájában írja le a rendszer működéséről kialakult tudását.
- Közvetett módszernek nevezzük, amikor bizonyos ideig megfigyeljük az operátor munkáját irányítás közben, és ezalatt a szükséges információkat (bemenetek, irányítás értéke, rendszerparaméterek) feljegyezzük. Ezután az adatok közvetlen feldolgozásával vagy klaszterezési eljárással, (ld. például [18]) a szakértő irányítási stratégiáját elemezve nyelvi szabályokat hozhatunk létre. Ehhez lényegében hasonló módszer, amikor a rendszer működéséről mintaadatok állnak rendelkezésre, amelyek alapján megalkothatjuk a nyelvi szabályokat. Ezt a módszert részletesebben a neurofuzzy irányítási rendszereknél fogjuk tárgyalni a 11. fejezetben.
- A szabályokat közvetlenül is megkaphatjuk, ha a folyamat (rendszer) működése fuzzy modell segítségével van leírva.

- Végül az irányítási rendszer maga is alkothat szabályokat, illetve tanulhat saját működéséből, ha rendelkezésre áll egy metaszabálybázis, melynek felhasználásával az irányítási rendszer képes kiértékelni saját viselkedését, és eldönteni, hogy az adott irányítási művelet hatására a rendszer jobb vagy rosszabb állapotba kerül [147, 158].

A fuzzy szabálybázis alkotói *természetes nyelvi szabályok*

$$R : \text{Ha } x = A \text{ akkor } y = B \quad (7.1)$$

formájúak, ahol $x \in X$ a *bemeneti változó*, $y \in Y$ a *kimeneti változó* vagy *következtetés*, X , illetve Y rendre a bemeneti, illetve kimeneti változók alaphalmaza, továbbá A és B *nyelvi változók*. A az R *szabály antecedense* (előzménye), B pedig az R *szabály konzekvensze* (következménye). Ha a szabályban szereplő nyelvi változók, azaz az antecedens és konzekvens fuzzy halmazok, akkor *fuzzy szabályról* beszélünk. Tegyük fel, hogy egy közlekedési lámpa működését irányító fuzzy rendszer szabálybázisa tartalmazza a „**Ha** a forgalom erős északi irányban, **akkor** a lámpa legyen hosszabb ideig zöld” szabályt. Ebben az esetben az x bemeneti változó az „északi irányú forgalom”, a következtetés y , azaz hogy mi a teendő a zöld lámpával. Az A szabályantecedensnek az „*erős forgalom*” nyelvi fogalmat, a B konzekvensnek a „*hosszabb ideig legyen zöld*” nyelvi fogalmat leíró fuzzy halmaz felel meg.

A rendszer működését leíró nyelvi szabályok összességét nevezzük *fuzzy szabálybázisnak* (vö. 7.1. ábra.). A szabályok antecedense fuzzy halmazokkal írja le a bemeneti változók valamely „körülbelüli” állapotát. A konzekvens az adott antecedenshez tartozó kimeneti fuzzy értéket határozzák meg.

A modellezett rendszer bonyolultságától függően a szabálybázis általában többdimenziós szabályokat tartalmaz. Ha a rendszernek n bemenete és m kimenete van, akkor az i -edik szabály általánosan

$$R_i : \text{Ha } \underline{x} = A_i \text{ akkor } \underline{y} = B_i \quad (7.2)$$

alakú, ahol a $\underline{x} = \langle x_1, \dots, x_n \rangle$ a bemeneti értékek vektora, $x_j \in X_j$, $X = X_1 \times \dots \times X_n$ az alaphalmaz, $\underline{A}_i = \langle A_{1i}, \dots, A_{ni} \rangle$ az antecedens halmazok vektora, $A_i \in X$; $\underline{y} = \langle y_1, \dots, y_m \rangle$ a kimeneti változók vektora, $y_j \in Y_j$, $Y = Y_1 \times \dots \times Y_m$ a kimeneti változók alaphalmaza, $\underline{B}_i = \langle B_{1i}, \dots, B_{mi} \rangle$ a konzekvens halmazok vektora, $B_i \in Y$, és $i \in [1, r]$, ahol r a szabályok száma. A (7.2) szabály felírható

$$R_i : \text{Ha } x_i = A_{1,i} \text{ és } \dots \text{ és } x_n = A_{n,i} \text{ akkor } \underline{y} = B_i \quad (7.3)$$

formában is, amely jobban kifejezi hogy a szabály alkalmazásának feltétele, hogy az összes bemeneti változó értéke pozitív mértékben essen a megfelelő antecedens halmazba. Vegyük észre, hogy a kimenő változók értékei függetlenek egymástól, azaz az m kimenetű szabályok m darab, egymástól független, egy kimenetű szabály halmazára dekomponálhatók. Formálisan:

$$R_i \longrightarrow \{R_{1,i}, \dots, R_{m,i}\}$$

ahol

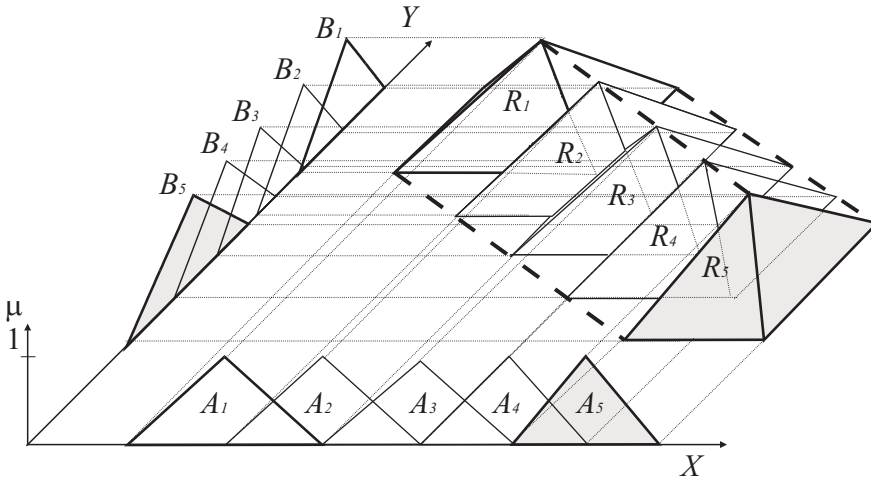
$$\begin{aligned} R_{1,i} : & \quad \text{Ha } x_i = A_{1,i} \text{ és } \dots \text{ és } x_n = A_{n,i} \text{ akkor } y_1 = B_{1,i}, \\ & \quad \vdots \\ R_{m,i} : & \quad \text{Ha } x_i = A_{1,i} \text{ és } \dots \text{ és } x_n = A_{n,i} \text{ akkor } y_m = B_{m,i}. \end{aligned}$$

A szabályok kimeneti oldalának dekomponálásával egyszerűbb szabályokat kapunk. A rendszer valós idejű működése szempontjából alapvető jelentőségű az, hogy az időigény ilyen módon csökken, hiszen a különböző kimeneti változók értékei párhuzamosan számolhatók lineáris időben. Ezért a továbbiakban csak egykimenetű rendszerekkel foglalkozunk.

7.2.2. A szabályok ábrázolása fuzzy relációkkal

A fuzzy szabályok interpretálásának többféle megközelítése létezik [51] (vö. Bevezetés 16. oldal). Az egyik, széles körben elterjedt felfogás szerint a (7.1) alakú szabály egy $A \times B$ „fuzzy pont”. A szabályok összessége (azaz a szabálybázis) pedig egy r pontból álló „fuzzy függvénygörbe” (r a szabályok száma). A fuzzy függvénygörbe a bemenő és kimenő változók (x és y) közötti reláció hozzávetőleges leírásának tekinthető [213] (ld. a 7.2. ábrát). A fuzzy szabályokat valamely konjunkció (t-norma) segítségével adjuk meg (az $A \times B$ DESCARTES-szorozattal). Az irodalomban elterjedt az az interpretáció is, ahol a ha-akkor típusú szabályokat implikációként értelmezik (ld. a 91. oldalon az 5.1. ábrát). A gyakorlati alkalmazások azonban a konjunkciós megoldáson alapulnak, ezért az implikációs változatot itt nem tárgyaljuk. A konjunkció alapú modell az egyes szabályokat adatpárokként kezeli, tehát például a (7.1) szabályt az (A, B) adatpárnak tekinti, mely az A szabályantecedens és B szabálykonzekvens között meglévő relációt írja le. A szabályok egyszerűsített jelölésére a félreérthetőség kizárásával az implikáció jelét használjuk, azaz a (7.1)-et az

$$R_i : A_i \rightarrow B_i$$



7.2. ábra. Fuzzy szabályok ábrázolása fuzzy függvénygörbével

formában adjuk meg tömören.

Az R_i fuzzy szabály-reláció, az $X \times Y$ DESCARTES-szorozattéren értelmezett fuzzy halmaz, amely az

$$R_i(x, y) : \mu_{R_i(x, y)}(x, y) = t(A_i(x), B_i(y)), \quad (x, y) \in X \times Y$$

képlettel adható meg, ahol t egy tetszőleges t-norma, a gyakorlatban többnyire a min művelet (ld. 7.3. ábra), azaz ha a ZADEH-féle t-normát használjuk, az R_i reláció

$$\mu_{R_i(x, y)}(x, y) = \min(A_i(x), B_i(y)), \quad (7.4)$$

alakú lesz.

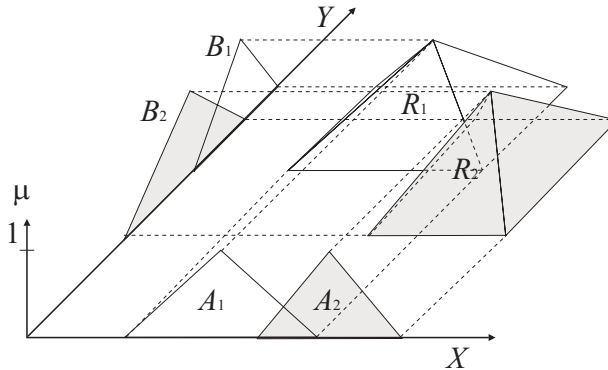
A szabálybázisban szereplő összes szabályok uniójaként adhatjuk meg az R fuzzy szabálybázis-relációt, amely a szabályokban megtalálható összes információt tartalmazza:

$$R = \bigcup_{i=1}^r R_i$$

Ha a ZADEH-féle uniót használjuk t-konormaként, akkor a teljes R relációt

$$\mu_{R(x, y)}(x, y) = \max_{i=1}^r (\mu_{R_i(x, y)}(x, y)) = \max_{i=1}^r (\min(A_i(x), B_i(y))) \quad (7.5)$$

alakban írhatjuk fel. A (7.4) és (7.5) kifejezések többdimenziós bemenet



7.3. ábra. Fuzzy szabály-reláció. A szabálybázis az $A_1 \rightarrow B_1$ és az $A_2 \rightarrow B_2$ szabályokat tartalmazza.

esetén értelemszerűen változnak:

$$\mu_{R_i(x_1, \dots, x_n, y)}(x_1, \dots, x_n, y) = \min(A_{1,i}(x_1), \dots, A_{n,i}(x_n), B_i(y)),$$

$$\mu_{R(x_1, \dots, x_n, y)}(x_1, \dots, x_n, y) = \max_{i=1}^r (\mu_{R_i(x_1, \dots, x_n, y)}(x_1, \dots, x_n, y)),$$

ahol $(x_1, \dots, x_n, y) \in X_1 \times \dots \times X_n \times Y$.

7.2.3. Nyelvi változók és fuzzy halmazok szemantikája

A fuzzy szabályokkal megfogalmazott irányítási stratégiák jelentős előnye a hagyományos módszerekkel szemben, hogy a szabályok közvetlen, természetes nyelvi voltak miatt könnyen érthetőek, ugyanakkor numerikus számolásoknál is alkalmazhatóak. A numerikus felhasználhatóságot a nyelvi változók (szabályantecedensek és -konzekvensek) fuzzy halmazként való reprezentálása teszi lehetővé.

A *nyelvi (lingvisztikai) változó* elnevezést ZADEH vezette be [209], a nyelvi változó értékei természetes (vagy mesterséges) nyelvi szavak vagy kifejezések lehetnek. Például a „sebesség” nyelvi változó, ha értékei nem numerikusan, hanem szavakkal definiáltak, azaz 5, 20, 50 vagy 200 km/h helyett *nagyon lassú*, *lassú*, *átlagos sebességű*, illetve *nagyon gyors* értékeket vehet fel.

A lingvisztikai változókat tehát fuzzy halmazokkal adhatjuk meg. A fuzzy halmazoknak többféle szemantika (értelmezés) feleltethető meg [50]. Történetileg az első felfogás a konvex és normális fuzzy halmazokat a hasonlóság, közelség, megkülönböztethetlenség leírásaként értelmezték.

Eszerint azok az elemek, melyeknek tagsági értéke 1, azaz a magban találhatók az adott fuzzy halmaz prototípusa, míg a többi elem 1-nél kisebb tagsági értéke a prototípuselem(ek)hez való közelséget határozza meg. Ez a megközelítés például az osztályozási és alakfelismerési feladatoknál használatos, ahol egy vizsgált objektum akkor kerül egy adott (fuzzy) osztályba, ha valamilyen elbírálás szerint elégséges mértékben hasonló a prototípushoz, azaz elég nagy a tagsági értéke az adott halmazban [16].

Egy más értelmezés szerint a fuzzy halmazok lényegében bizonytalan állapotokat írnak le szubjektív valószínűségi eloszlások esetén. Eszerint a fuzzy halmazok pontatlan vagy bizonytalan információk modellezésére alkalmasak [211].

A harmadik szemantikai magyarázat szerint a fuzzy halmazok rugalmas kényszerfeltételek, specifikációk vagy célok esetén a feltételektől függő, különböző, többé-kevésbé elfogadható megoldások közötti döntési preferenciákat testesítik meg [17]. A fuzzy halmazok elmélete a fokozatosság bevezetésével lehetővé tette a kétpólusú igen-nem típusú döntések finomítását, s így módon a döntési skála kiterjesztését a két szélsőérték, a teljesen elfogadható és a teljesen elfogadhatatlan között. Ennek az értelmezésnek igen komoly szerepe van döntéstámogatási feladatok esetén. Fuzzy halmazok felhasználásával a hagyományos kényszerfeltétel-megoldó algoritmusok és optimalizációs technikák is kiegészíthetők oly módon, hogy képesek legyenek egyszerre kezelni rugalmas feltételeket és bizonytalan adatokat.

A felsorolt három értelmezés szerint egy fuzzy halmaz tagsági értékei a kontextustól függően (legalább) három különböző módon értelmezhetők. Legyen a példa a „magas” nyelvi címkével ellátott halmaz. Az első szemantika szerint a „magas” fogalom a magasságoknak egy fuzzy osztályát határozza meg, mely közelíti a magas prototípusának értékeit. Másodszor jelenthet egy bizonytalan állapotot, amennyiben csak azt tudjuk, hogy például „János magas”, de további információval nem rendelkezünk a magasságáról. Ekkor szubjektív valószínűségi eloszlást adhatunk meg János magasságának konkrét értékeit illetően. Végül a halmaz kifejezhet egy rugalmas feltételt, azaz ha valamilyen célból olyasvalakit keresünk aki „magas”, azaz aki valamilyen értelemben megfelel egy feltételnek.

Fuzzy irányítási rendszerek esetén a fuzzy halmazoknak mindhárom szemantikus értelmezését használjuk. Az elsőt, mikor nyelvi címkéket és változókat hozunk létre, a másodikat a fuzzy szabályok megalkotásánál, a harmadikat pedig a bemeneti halmazok (megfigyelés) fuzzy halmazzá alakításánál.

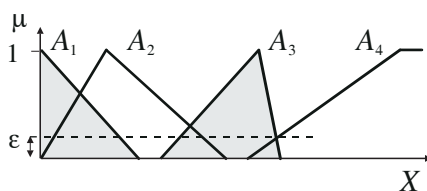
7.2.4. Fuzzy partíciók és tulajdonságaik

A szabálybázis szabályai a bázis által reprezentált információ egységei. Az információegységeknek minden dimenzióban nyelvi változók értékei felelnek meg, melyeket fuzzy halmazokkal modellezünk. Az egyes nyelvi változók lehetséges értékei általános értelemben felosztják, illetve részlegesen lefedik a változóhoz tartozó alaphalmazt.

Valamely bemeneti nyelvi változóhoz tartozó fuzzy halmazokra az alábbi feltételnek kell teljesülnie: Együttesen fedjék le az alaphalmazt olyan értelemben, hogy minden lehetséges bemeneti értékre létezzék valamilyen pozitív tagsági értékű információ. Formálisan megfogalmazva, ha az X alaphalmazon értelmezett változóhoz az $\{A_1, \dots, A_n\}$ fuzzy halmazok tartoznak, akkor

$$\forall x \in X, \exists i \in [1, n] : A_i(x) \geq \varepsilon,$$

ahol $\varepsilon > 0$ az X lefedettségének mértéke (7.4. ábra). Erre azért van szükség, hogy minden megfigyeléshez létezzék a szabálybázisban olyan szabály, amelynek alapján az irányítási rendszer képes valamilyen következtetés meghozatalára. Az $\mathcal{A} = \{A_1, \dots, A_n\}$ fuzzy halmazcsaládot az X alaphalmaz *fuzzy partíciójának* nevezik.



7.4. ábra. Az alaphalmaz ε -lefedése fuzzy halmazokkal

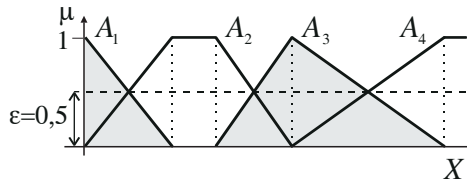
Ha az A_i halmazok tagsági értékének összege minden x alaphalmazbeli elemre vonatkozóan 1, akkor az $\mathcal{A}$ halmazcsalád ún. RUSPINI-partíciót alkot [153]:

$$\sum_{i=1}^n A_i(x) = 1, \quad \forall x \in X. \quad (7.6)$$

Az igen elterjedt háromszög vagy trapéz alakú antecedens halmazok esetén a (7.6) feltétel könnyen teljesíthető, ha a

$$\begin{aligned} \sup(\text{supp}(A_i(x))) &= \inf(\text{core}(A_{i+1}(x))) \\ \sup(\text{core}(A_i(x))) &= \inf(\text{supp}(A_{i+1}(x))) \end{aligned}$$

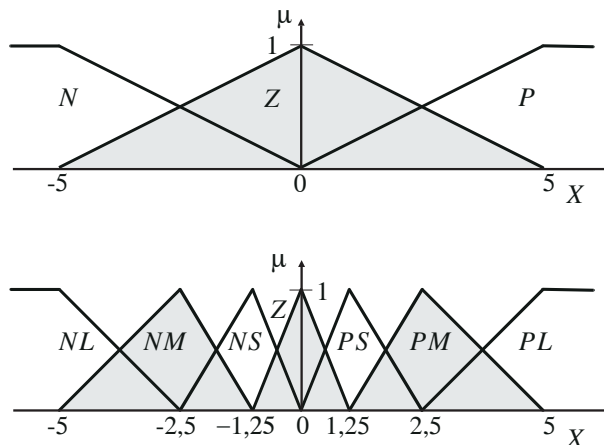
összefüggések fennállnak, vagyis ha minden fuzzy halmaz magjának szélsőértékei megegyeznek a megelőző és a rákövetkező fuzzy halmaz tartójának maximumával, illetve minimumával (ld. 7.5. ábra.).



7.5. ábra. Fuzzy halmazok RUSPINI-partíciója

Nagyon lényeges a megfelelő alaphalmaz kiválasztása. Ha a megfigyelés numerikus jellegű, akkor célszerű az alaphalmaz alsó és felső korlátját oly módon meghatározni, hogy tartalmazzon minden lehetséges megfigyelést. Az alaphalmaz skálázását úgy kell megoldani, hogy az lehetőleg viszonylag kis számú fuzzy halmazzal lefedhető legyen, ugyanis a végrehajtási idő és a szabálybázis tárolásához szükséges tárigény exponenciálisan arányos a szabályok (azaz az antecedens) halmazok számával (ld. még 8.1. szakasz).

Az $\mathcal{A}$ fuzzy partíció specifikusabb, mint az $\mathcal{A}'$, ha minden eleme specifikusabb valamilyen mérték szerint. Ekkor az $\mathcal{A}$ elemeinek száma nagyobb $\mathcal{A}'$ elemszámánál, azaz több fuzzy halmazt tartalmaz. Például az $\mathcal{A}' = \{N, Z, P\}$ partíciónál az $\mathcal{A} = \{NL, NM, NS, Z, PS, PM, PB\}$ speci-



7.6. ábra. Az $\mathcal{A}$ fuzzy partíció hét, míg az $\mathcal{A}'$ három nyelvi kifejezést tartalmaz

fikusabb (ld. 7.6. ábra). Itt jegyezzük meg, hogy a könyvben az ilyen jellegű nyelvi kifejezések értékére az irodalomban elterjedt angol rövidítéseket alkalmazzuk: N negatív, Z körülbelül nulla, P pozitív, L nagy, M közepes, S kicsi; eszerint például a PM közepes pozitív értéket jelent.

Ugyanakkor megfigyelhető, hogy minél több nyelvi kifejezést tartalmaz egy fuzzy partíció, a nyelvi címkék kifejezőereje annál kisebb lesz, hiszen a fuzzy partíciók e két tulajdonsága kölcsönösen gyengíti egymást. Szélsőséges esetben, ha a fuzzy halmazok egyelemű numerikus értékekhez közelítenek, a partíció specifikussága nagy lesz, de a nyelvi kifejezőképesség teljesen eltűnik. Tehát a nyelvi címkék, azaz a felhasznált fuzzy halmazok számának meghatározásánál ésszerű kompromisszumra kell törekedni a pontosság és a nyelvi kifejezőerő (és mint később látni fogjuk, a számítási bonyolultság) között.

7.3. Mamdani-féle fuzzy irányítási rendszerek

A fuzzy irányítási rendszerek alapelvét először ZADEH javasolta 1973-ban [208] a nagy bonyolultságú rendszerek modellezését tekintve elsődleges célnak. E módszer lényege, hogy $\langle X \times Y, \mu_R \rangle$ formában, fuzzy relációként interpretálja a szabálybázist, ahol

$$\mu_R : X \times Y \rightarrow [0,1].$$

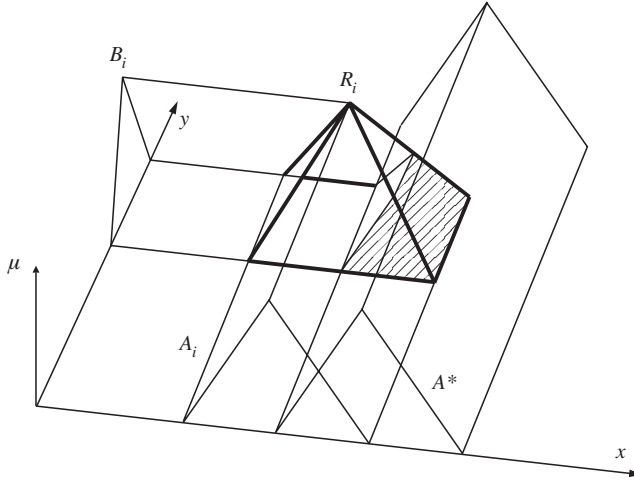
A megfigyelés ekvivalenciarelációként fogalmazható meg:

$$A^* : X \times X \rightarrow [0,1],$$

ilyen módon lehetővé téve a következtetés (például a max és min normákon alapuló) fuzzy kompozícióként való előállítását (ld. 7.7. ábra):

$$B^* = A^* \circ R.$$

A nagy számításigény miatt azonban a gyakorlati alkalmazásokban az algoritmusnak a MAMDANI által egy évvel később módosított változata terjedt el [130], mely többdimenziós X bemenet esetén nem magán az R reláción, hanem annak ortogonális projekcióin működő algoritmust használ. Ezzel az eljárás erősen megszorítja ugyan a szóba jöhető modellek körét, ugyanakkor a számítási bonyolultság szempontjából lényegesen kedvezőbb helyzetet teremt. Az alábbiakban ismertetjük a MAMDANI-irányítók működési elvét.



7.7. ábra. A kompozíciós következtetési szabály

A fuzzy irányítási rendszerek általános felépítését a fejezet elején tárgyaltuk (ld. 7.1. ábra). Ezeknek leglényegesebb alapegysége a következtető gép által használt következtetési algoritmus, mely előállítja a megfigyelésből a következtetést.

A következtetési algoritmus első lépése az aktuális megfigyelés (bemeneti értékek) és a szabályok antecedenseinek illesztése. Minden egyes szabályantecedenshez meg kell határozni a megfigyeléssel való *illeszkedés (tüzelés vagy hasonlóság) mértékét*, melynek alapján meghatározható, hogy az egyes szabályok milyen mértékben játszanak szerepet a konklúzió megalakításában.

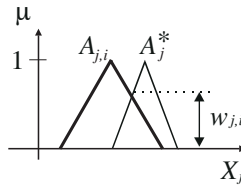
Legyen az $\underline{A}^* \in X_1 \times \dots \times X_n$ az n -dimenziós megfigyelésvektor, az r darab szabály pedig (7.3) alakú. Az illeszkedés mértéke a j -edik dimenzióban ($j \in [1, n]$) a

$$w_{j,i} = \max_{x_j} \{ \min \{ A_j^*(x_j), A_{j,i}(x_j) \} \} \quad (7.7)$$

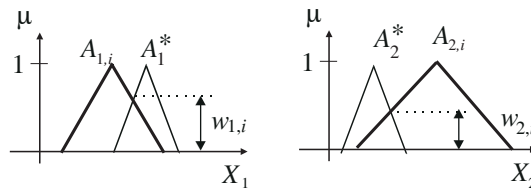
súlyfaktor kiszámításával határozható meg (ld. 7.8. ábra). A $w_{j,i}$ súlyfaktor az A_j^* megfigyelés és az $A_{j,i}$ szabályantecedens kapcsolatát jellemzi.

Az R_i szabály alkalmazhatóságát (illeszkedésének mértékét) a szabály feltételoldalán lévő összes antecedenshez tartozó súlyfaktorok minimumaként határozhatjuk meg (7.9. ábra):

$$w_i = \min_{j=1}^n w_{j,i}. \quad (7.8)$$



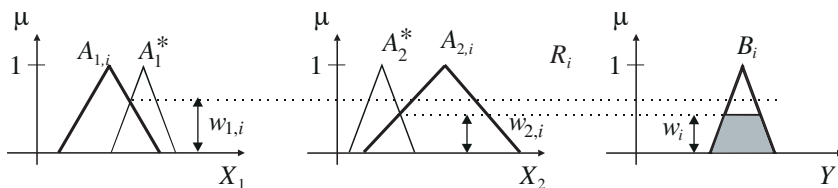
7.8. ábra. Az illeszkedés mértékének meghatározása egy dimenzióban



7.9. ábra. Az illeszkedés mértékének meghatározása több dimenzióban

A w_i súlyfaktor adja meg, hogy az R_i szabály konzekvensé milyen mértékben szerepel a végső következtetés előállításában. A B_i konzekvenst w_i „magasságban” csonkoljuk, s így kapjuk meg az adott megfigyeléshez és szabályhoz tartozó B_i^* következtetést (7.10. ábra). Formálisan

$$B_i^*(y) = \min(w_i, B_i(y)) \quad (7.9)$$



7.10. ábra. Az R_i szabályhoz tartozó következtetés meghatározása

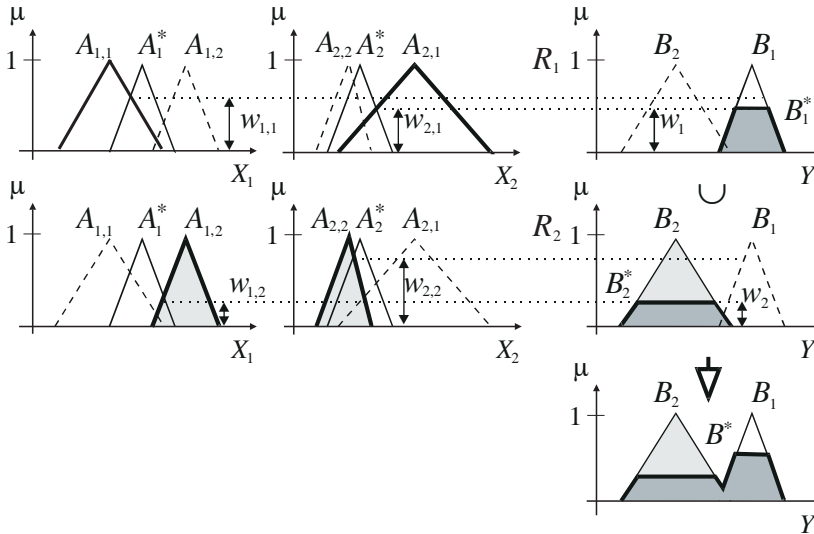
Vegyük észre, hogy ha a megfigyelés az antecedenssel minden dimenzióban egyezik, vagy elfedi azt, akkor a súlyfaktor értéke 1 lesz, és a B_i^* következtetés megegyezik a szabály konzekvensével. Ugyanakkor, ha bármelyik dimenzióban a megfigyelés és az antecedens metszete üres, azaz létezik $j = 1, \dots, n$, hogy $w_{ji} = 0$, tehát $w_i = 0$, akkor a szabályhoz tartozó következtetés üres fuzzy halmaz lesz.

Az egész szabálybázishoz tartozó összesített következtetés az egyes szabályokhoz tartozó B_i^* konklúziók uniójaként áll elő:

$$B^* = \bigcup_{i=1}^r B_i^*, \text{ azaz } B^*(y) = \max_{i=1}^r B_i^*(y). \quad (7.10)$$

A végső konklúzió meghatározása a MAMDANI-módszer esetén interpolatív jellegű abban az értelemben, hogy azt több szabály következtetésének egyfajta súlyozott átlagolásával kapjuk, ahol az egyes következtetéseket a bemenet és a megfigyelés illeszkedésének mértékével súlyozzuk.

A MAMDANI-féle módszer egészének működéséről a 7.11. ábra ad áttekintést.



7.11. ábra. MAMDANI-irányító algoritmus

Figyeljük meg a módszerben a fuzzy relációk megjelenését. A szabálybázis reláció

$$R(x_1, \dots, x_n, y) = \max_{i=1}^r \left\{ \min_{x,y} \{A_{1,i}(x_1), \dots, A_{n,i}(x_n), B(y)\} \right\}$$

$$R(\underline{x}, y) = \max_{i=1}^r \left\{ \min_{\underline{x}, y} \{A_i(\underline{x}), B(y)\} \right\} \quad (7.11)$$

alakban írható fel. Összegezve a (7.7)–(7.10) képleteket, az alábbi egyenlete-

ket kapjuk:

$$w_{j,i} = \max_{x_j} \left\{ \min_{x_j} \{A_j^*(x_j), A_{j,i}(x_j)\} \right\}, \quad (7.12)$$

$$\begin{aligned} w_i &= \min_j \left\{ \max_{x_j} \left\{ \min_{x_j} \{A_j^*(x_j), A_{j,i}(x_j)\} \right\} \right\} \\ &= \max_{x_j, j} \left\{ \min_j \left\{ \min_{x_j} \{A_j^*(x_j), A_{j,i}(x_j)\} \right\} \right\} \\ &= \max_{\underline{x}} \left\{ \min_{\underline{x}} \{A^*(\underline{x}), A_i(\underline{x})\} \right\}, \end{aligned} \quad (7.13)$$

$$\begin{aligned} B_i^*(y) &= \min_y \left\{ B_i(y), \max_{\underline{x}} \left\{ \min_{\underline{x}} \{A^*(\underline{x}), A_i(\underline{x})\} \right\} \right\} \\ &= \max_{\underline{x}} \left\{ \min_y \left\{ B_i(y), \min_{\underline{x}} \{A^*(\underline{x}), A_i(\underline{x})\} \right\} \right\} \\ &= \max_{\underline{x}} \left\{ \min_{\underline{x}, y} \{B_i(y), A^*(\underline{x}), A_i(\underline{x})\} \right\}, \end{aligned} \quad (7.14)$$

$$\begin{aligned} B^*(y) &= \max_{i=1}^r \left\{ \max_{\underline{x}} \left\{ \min_{\underline{x}, y} \{B_i(y), A^*(\underline{x}), A_i(\underline{x})\} \right\} \right\} \\ &= \max_{\underline{x}} \left\{ \max_{i=1}^r \left\{ \min_{\underline{x}, y} \left\{ A^*(\underline{x}), \min_{\underline{x}, y} \{A_i(\underline{x}), B_i(y)\} \right\} \right\} \right\} \\ &= \max_{\underline{x}} \left\{ \min_{\underline{x}, y} \left\{ A^*(\underline{x}), \max_{i=1}^r \min_{\underline{x}, y} \{A_i(\underline{x}), B_i(y)\} \right\} \right\}. \end{aligned} \quad (7.15)$$

A (7.15) egyenletben a szabálybázis relációjának (7.11) képletét behelyettesítve a

$$B^*(y) = \max_{\underline{x}} \left\{ \min_{\underline{x}, y} \{A^*(\underline{x}), R(\underline{x}, y)\} \right\} \quad (7.16)$$

összefüggéshez jutunk. Könnyen észrevehetjük, hogy a (7.16) kifejezés a max-min kompozíció alakú fuzzy reláció (vö. (4.10)), a MAMDANI-módszer következtetési algoritmusá által előállított konklúzió a megfigyelés és a szabálybázis reláció max-min kompozíciója:

$$B^* = A^* \circ R.$$

Ezért ezt a következtetési eljárást *kompozíciós következtetési szabálynak* is nevezik a szakirodalomban.

MAMDANI a fent ismertetett eljárását sikeresen alkalmazta egy félüzemi gőzgépes rendszer kvázioptimális irányítására. Ezt az erősen nemlineáris rendszert más ismert technikákkal csak ennél rosszabb eredménnyel lehetett irányítani [130].

A MAMDANI-féle eljárásban alkalmazottól eltérő t-normák és t-konormák használatával hasonló módszereket kaphatunk. A legismertebb a LARSEN által javasolt algoritmus [125], melyben a ZADEH-féle metszetet az algebrai szorzattal helyettesítve kedvező tulajdonságú következtető eljárást kapunk (ld. 7.12. ábra). Ennek alapján a (7.10) kifejezés a

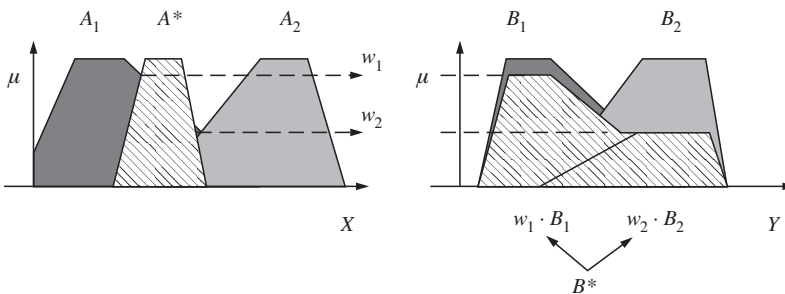
$$B^*(y) = \max_{i=1}^r \{w_i \cdot B_i(y)\}$$

egyenletre módosul.

7.4. Defuzzifikációs módszerek

A MAMDANI-típusú következtetési algoritmusok fuzzy halmazt adnak eredményül. Ez az elsődleges konklúzió, mely általában lingvisztikai kifejezésekkel közelíthető, vagy összetett rendszerek esetén más fuzzy irányítási rendszer bemeneti adataként hasznosítható. A gyakorlati alkalmazások zömében azonban a fuzzy irányítási rendszer kimeneteként egyszerű crisp numerikus értékre van szükség. A fuzzy konklúzióból tehát ki kell választani egy konkrét értéket, mely az adott fuzzy halmazt az alkalmazástól, illetve modellezett rendszertől függően a legjobban jellemzi. Ezt az eljárást *defuzzifikációnak* nevezzük. Az alkalmazás típusától függően a fuzzy halmaz értelme eltérő lehet, ezért a megfelelő eredmény eléréséhez különböző defuzzifikációs módszereket célszerű használni.

A fuzzy szakirodalomban számos defuzzifikációs módszer ismert, melyek közül a legismertebbeket és leggyakrabban alkalmazottakat mutatjuk be. A defuzzifikációs eljárásokról átfogó ismertetés található a [73] közleményben.



7.12. ábra. LARSEN-típusú következtető eljárás által számolt konklúzió

7.4.1. Súlypont módszer (COG)

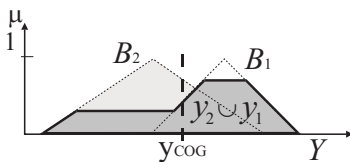
A módszer alkalmazásának előfeltétele, hogy a B^* tartója intervallum legyen, valamint hogy a

$$\text{MAX}(B^*) = \{y \in \text{supp}(B^*) | \forall y' \in \text{supp}(B^*) : B^*(y') \leq B^*(y)\} \quad (7.17)$$

halmaz nemüres és (BOREL-)mérhető legyen [63]. A B^* halmaz legjellemzőbb pontjául a súlypontot (Center Of Gravity) adjuk meg, melyet az egyes B_i^* részkonklúziók súlypontjának átlagaként kapunk meg:

$$\begin{aligned} y_i^* &= \frac{\int_{y \in \text{supp}(B_i^*)} B_i^*(y) y dy}{\int_{y \in \text{supp}(B_i^*)} B_i^*(y) dy} \\ w_i^* &= \int_{y \in \text{supp}(B_i^*)} B_i^*(y) dy \\ y_{\text{COG}} &= \frac{\sum_{i=1}^r (y_i^* \cdot w_i^*)}{\sum_{i=1}^r w_i^*} \end{aligned} \quad (7.18)$$

ahol y_i^* a B_i^* részkonklúzió súlypontja, w_i^* pedig a súlyozási faktor (ld. 7.13. ábra).

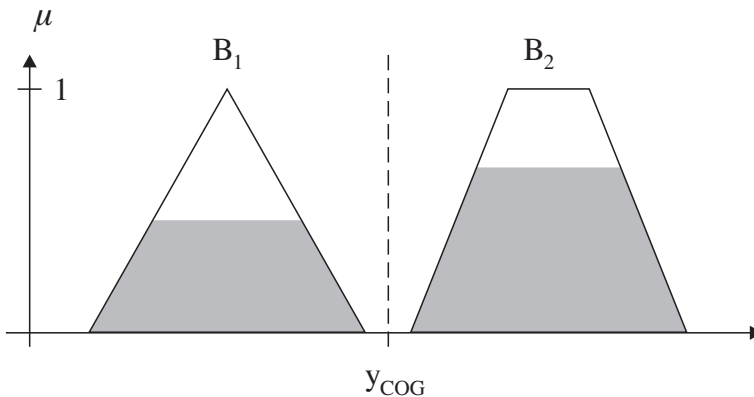


7.13. ábra. Defuzzifikálás súlypont módszerrel

Ez az egyik leggyakrabban használt defuzzifikációs módszer. Előnyei közé tartozik, hogy háromszög és trapéz alakú szabályoknál viszonylag egyszerűen számolható, valamint hogy közvetlen irányítás esetén majdnem mindig folytonos viselkedést eredményez: ha a megfigyelés s ezzel együtt a szabályok alkalmazhatóságának mértéke kis mértékben változik, az nem okoz nagy eltérést a crisp következmény értékében sem. Ebből adódóan a módszer minden tüzelő szabályt az illeszkedési mértékének megfelelően vesz tekintetbe, így minden tüzelő szabálynak van befolyása a defuzzifikált érték meghatározásában.

A módszer hátránya, hogy az eredmény szemantikusan nehezen értelmezhető, ez a valószínűségszámítási (várható érték) analógiájának következménye. Ezenkívül előfordulhat az is (ld. 7.14. ábra), hogy a módszer

olyan értéket határoz meg, amelyre a konklúzió tagsági értéke nulla. Tekintsük azt a példát, mikor az irányítás célja egy jármű akadályok közötti automatikus irányítása. Abban az esetben, ha a járművel éppen szemben van egy akadály, akkor vagy jobb, vagy bal oldalra kell kerülni. Ekkor a fuzzy következtetésnek a „kerüld el az akadályt jobbra vagy balra kormányzással” utasítás lehet a nyelvi interpretálása. Ugyanakkor a súlypont módszer a két alternatívát átlagolva pontosan az akadálynak irányítaná a járművet.



7.14. ábra. Rossz defuzzifikáláshoz vezető szituáció

Az ilyen helytelen defuzzifikálás természetesen konvex következtetés halmazok esetén nem fordulhat elő. A 7.14. ábrán látható szabálytípusban operátor által megadott nemdeterminisztikus irányítási stratégia jelenik meg. Ilyen esetekben a defuzzifikáció kétcélú: egyrészt a megfelelő crisp érték előállítás, másrészt a lehetséges irányítási műveletek közül való választás. Ha a fuzzy konklúzió egyetlen crisp értéket reprezentál, akkor a második feladat fölösleges. A problémát legegyszerűbben úgy oldhatjuk meg, ha a szabályokba determinisztikus döntési stratégiát kódolunk, ami általában nem jelent lényeges megszorítást, és javítja az irányítás megbízhatóságát is. Az egymásnak ellentmondó szabályokat tartalmazó szabálybázisok esetén alkalmazható technikákat tárgyalják például [89, 204] tanulmányok.

7.4.2. Geometriai középpont módszer (COA)

Nagyon hasonló a súlyponti módszerhez, s ezért itt említjük a geometriai középpont módszert (Center Of Area). A két módszer közötti különbség,

hogy a súlypont módszer a több részkonklúzió által fedett területeket többszörösen számolja, míg a geometriai középpont módszer csak a B^* következtetés alakját veszi figyelembe, így az átlapolt területeket természetesen csak egyszeres súllyal veszi figyelembe. Komoly hátránya a súlypont módszerhez képest, hogy bonyolult alakú részkonklúziók esetén igen nehezen számolható. A defuzzifikált érték a geometriai középpont eljárással az

$$y_{\text{COA}} = \frac{\int_{y \in B^*} B^*(y) y dy}{\int_{y \in B^*} B^*(y) dy} \quad (7.19)$$

kifejezés alapján számolható. Diszkrét kimenet esetén, ha a B^* konklúzió az $\{y_1, \dots, y_m\}$ halmazon van definiálva, a (7.19) képlet a

$$y_{\text{COA}} = \frac{\sum_{i=1}^m B^*(y_i) y_i}{\sum_{i=1}^m B^*(y_i)}$$

kifejezésre módosul. Ebben az esetben, ha y_{COA} nem azonos az univerzum egyik elemével sem, azaz nem létezik olyan i , amire $y_{\text{COA}} = y_i$, akkor a legközelebbi értéket választjuk.

7.4.3. Maximumok közepe módszer (MOM)

A módszer alkalmazásának előfeltételei megegyeznek a súlypont módszerével (7.17). A defuzzifikált érték a (7.17) halmaz középértéke (Mean Of Maxima) (7.15. ábra):

$$y_{\text{MOM}} = \frac{\int_{y \in \text{MAX}(B^*)} y dy}{\int_{y \in \text{MAX}(B^*)} dy}. \quad (7.20)$$

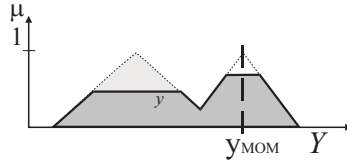
Ha a $\text{MAX}(B^*)$ halmaz véges vagy megszámlálható számosságú, akkor a

$$y_{\text{MOM}} = \frac{\sum_{y \in \text{MAX}(B^*)} y}{|\text{MAX}(B^*)|}.$$

kifejezést kapjuk.

A módszert leginkább véges elemszámú univerzum esetén alkalmazzák. Előnye, hogy egyszerűen számolható.

Hátrányai közül a legjelentősebb, hogy nemfolytonos irányítási függvényt eredményez. A legnagyobb illeszkedési mértékű szabály csúcspontja körül helyezkedik el a $\text{MAX}(B^*)$ halmaz, amiből a választott crisp érték kikerül. Abban az esetben, ha a megfigyelés úgy változik, hogy egy másik



7.15. ábra. Defuzzifikáláshoz a maximumok közepe módszerrel

szabálynak lesz a legmagasabb a tüzelési értéke, akkor a $\text{MAX}(B^*)$ halmaz ez utóbbi csúcsa körül lesz, így előfordulhat, hogy a megfigyelés kismértékű megváltozása az eredményben nagy eltérést okoz. Tehát a domináns szabály megváltozása esetén az eredmény „ugrálni” fog.

Az eljárás átlagoló jellegéből következik, hogy a súlypont módszernél bemutatott „ütközési” jelenség (7.14. ábra) szintén előfordulhat.

7.4.4. Középső maximum módszer (COM)

Az eljárás a következtetés legnagyobb tagságfüggvény-értékű elemeiből választja ki a középsőt (Center Of Maxima). Legyen $h(B^*)$ a következtetés magassága, ekkor

$$y_{\text{COM}} = \frac{\inf M + \sup M}{2}, \quad (7.21)$$

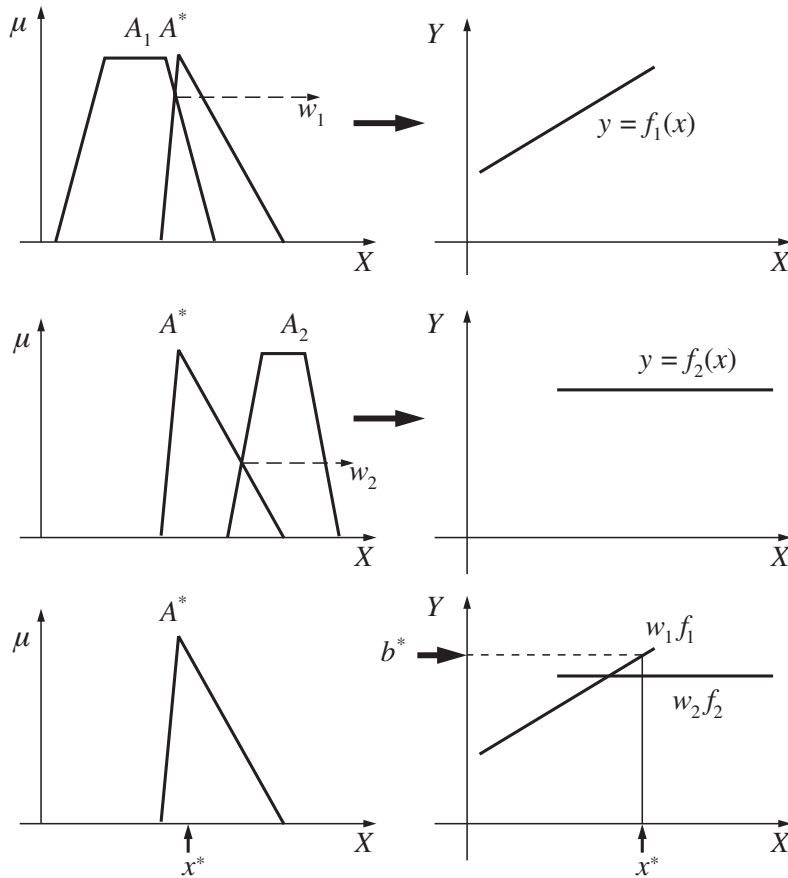
ahol $M = \{y | B^*(y) = h(B^*)\}$. Diszkrét esetben

$$y_{\text{COM}} = \frac{\min\{y_k | y_k \in M\} + \max\{y_k | y_k \in M\}}{2}.$$

Az eljárás egyszerűen számolható, de az előző eljárással azonos hátrányokkal bír.

7.5. Függvény kimenetű fuzzy irányítási rendszerek

A 80-as évek közepétől SUGENO és iskolája olyan alternatív fuzzy irányítási modellt javasolt, melyben a szabályok konzekvens oldalán nem fuzzy halmazok szerepelnek, hanem konstans, lineáris, esetleg más, bonyolultabb (nem fuzzy) függvények [171, 173, 178]. Ennek egyik előnye, hogy kiküszöböli a defuzzifikálás olykor időigényes és bizonyos esetekben lingvisztikailag nehezen megindokolható lépését, amivel a számítási idő és a modell bonyolultsága csökken. (Ez utóbbi természetesen csak akkor,



7.16. ábra. TAKAGI-SUGENO-típusú irányítók működése

ha a szabálykonzekvensekben szereplő függvények nem túl bonyolultak.) A másik előnye, hogy struktúrája és működése egyszerűbb, mint a 7.3. szakaszban ismertetett MAMDANI-irányítóké.

A szabályok általános alakja

$$\text{Ha } x_1 = A_{1,i}, \dots, x_n = A_{n,i} \text{ akkor } y_i = f_i(x_1, \dots, x_n), \quad (7.22)$$

ahol $x_i, i \in [1, n]$ a bemenő változók, f_i pedig tetszőleges n -dimenziós függvény. A szakirodalomban az f_i függvény bonyolultságától függően az alábbi irányítótípusokat különböztetik meg. Ha f_i konstans, akkor (nul-ladrendű) SUGENO-irányítóról, ha a bemenetek lineáris függvénye akkor elsőrendű SUGENO- vagy TAKAGI-SUGENO-irányítóról, ha magasabbren-

dű függvény, akkor általános SUGENO- vagy TAKAGI–SUGENO–KANG-irányítóról beszélünk.

Ilyen típusú irányítókkal a bemeneti állapottér minden egyes, a szabályok által megkülönböztetett régiójához egy f_i irányítási függvényt rendelhetünk hozzá. Ugyanezt hagyományos irányítási rendszerek segítségével is megtehetjük, ám a fuzzy halmazok bevezetésével a kimeneti függvények között sima átmenetet biztosíthatunk.

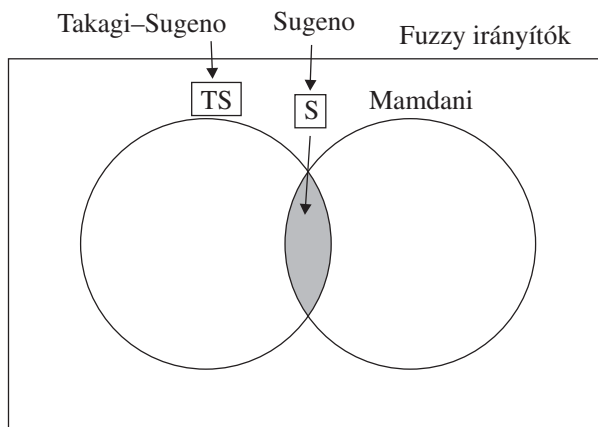
A SUGENO-típusú irányítók működési elve megegyezik a MAMDANI-féle irányítókéval. A bemenetek fuzzifikálása után a megfigyelés és a szabályok kiértékelésével meghatározható az egyes szabályok w_i illeszkedési mértéke (7.7)–(7.8), illetve (7.12)–(7.13) segítségével. Ennek alapján meghatározható a következtetés (ld. 7.16. ábra):

$$y = \frac{\sum_{i=1}^r w_i \cdot y_i}{\sum_{i=1}^r w_i} = \frac{\sum_{i=1}^r w_i \cdot f_i(x_1, \dots, x_n)}{\sum_{i=1}^r w_i}. \quad (7.23)$$

Nullandrendű SUGENO-irányítók esetén a (7.23) kifejezés az

$$y = \frac{\sum_{i=1}^r w_i \cdot c_i}{\sum_{i=1}^r w_i}. \quad (7.24)$$

összefüggésre egyszerűsödik, ahol c_i konstans. Ez az egyenlet még tovább redukálható egydimenziós bemenet esetén, ha a szabálybázis RUSPINI-partíciót alkot. Ekkor ugyanis (vö. (7.6)) az illeszkedési mértékek összege 1



7.17. ábra. SUGENO- és MAMDANI-irányítók kapcsolata

lesz, így

$$y = \sum_{i=1}^r w_i \cdot c_i.$$

Az általános SUGENO-irányítók és a MAMDANI-irányítók halmazának metszetét a (nulladrendű) SUGENO-irányítók jelentik, hiszen a konstans szabálykonzekvens egyelemű, ún. singleton fuzzy halmazként is felfogható. A SUGENO-, és MAMDANI-irányítók kapcsolatát a 7.17. ábrán mutatjuk be.

7.6. Fuzzy irányítási rendszerek explicit függvényei

A fuzzy irányítási rendszerek funkcionális szempontból függvénygenerátornak tekinthetők. Az irányító ugyanis felfogható egy „fekete doboznak”, mely a többnyire valós (azaz nem fuzzy) bemenetből, vagy bemenetvektorból az előző szakaszokban ismertetett módon előállít egy valós kimenetet, vagy kimenetek vektorát. Felmerül tehát a kérdés, hogy milyen függvényekkel lehet a fuzzy irányítókat helyettesíteni, azaz melyek a fuzzy irányítók explicit függvényei, illetve melyik az a legtágabb függvényosztály, mely az egyes fuzzy irányítótípusokkal megvalósítható.

A fenti kérdések megválaszolása további problémákat vet fel. Miért részesítsük előnyben a fuzzy irányítókat más függvénygenerátorokkal szemben? Helyettesíthetők-e a fuzzy irányítási rendszerek más, tagsági függvényeket nem használó irányítási algoritmusokkal, mint azt bizonyos szerzők javasolták [123]? Milyen következtetés szűrhető le az explicit függvényekből a megvalósítható függvényosztállyal kapcsolatban? Valamilyen értelemben jobb irányítást nyújt-e a fuzzy megközelítés, mint a hagyományos metodológia? Ebben a szakaszban megkísérlünk válaszolni ezekre a kérdésekre.

7.6.1. Explicit függvények egyenlő szárú háromszög alakú szabályok esetén

Ha a rendszerfeljesztés és hangolás aspektusait nem vesszük számításba, akkor a valós be- és kimenetű fuzzy irányítók valóban helyettesíthetők valós függvényekkel. Az első eredményeket ebben az irányban EL HAJJAJI és RACHID ismertették [53]. Munkájukban egy igen speciális modellt vizsgáltak, melyben mind az antecedens, mind a konzekvens fuzzy halmazok olyan egyenlő szárú háromszögek, melyek úgy helyezkednek el, hogy egy

háromszög tartójának két végpontja a két szomszédos halmaz csúcspontjával esik egybe (ugyanilyen elrendezés látható például a 9.2. ábrán, 173. old.). Az elrendezés miatt teljesül a RUPINI-partíció (7.6), amelynek következménye, hogy minden lehetséges megfigyelés legfeljebb két szabályt aktivál, azaz

$$w_1 = A_i(x^*), \quad w_2 = A_{i+1}(x^*) \quad \text{és} \quad w_1 + w_2 = 1, \quad (7.25)$$

ahol w_1 , illetve w_2 jelöli a megfigyelés és a két egymást követő antecedens illeszkedésének mértékét.

Elsőként MAMDANI-féle következtetési algoritmus, valamint geometriai középpont defuzzifikáció esetén határozták meg a konklúziót:

$$y_{\text{COA}}^* = \left(i + \frac{1}{2}\right) b + \frac{(w_1 - 1)^2 - (w_2 - 1)^2}{2w_1 - w_1^2 + w_2} \cdot \frac{b}{2}, \quad (7.26)$$

ahol b a konzekvens fuzzy halmazok tartója. A képlet helyességét az Olvasó maga is könnyen ellenőrizheti a megfelelő részkonklúziók, a végső következtetés, valamint ez utóbbi geometriai középpontjának kiszámításával.

(7.25) miatt a (7.26) kifejezés

$$y_{\text{COA}}^* = \left(i + \frac{1}{2}\right) b + \frac{1 - w_1}{1 + w_1 + w_2} \cdot \frac{b}{2} \quad (7.27)$$

egyenletre egyszerűsíthető. Ha ebbe behelyettesítjük az x^* megfigyelés értékét, akkor a

$$y_{\text{COA}}^* = c_1 i + c'_1 \frac{c_2 + x^*}{c_3 + c_4 x^* + c_5 (x^*)^2}, \quad (7.28)$$

kifejezést kapjuk, mely a bemenet és a kimenet kapcsolatát jellemzi. A (7.28) összefüggésben szereplő c_i konstansok a szabályokból levezethetők.

Érdeemes megjegyezni, hogy a c_4 és c_5 konstans sosem lehet zérus. Az eredményül kapott racionális törtfüggvény viselkedése nehezen áttekinthető. A függvény korlátosságát a szabályokban szereplő fuzzy halmazok geometriai tulajdonságai biztosítják, belátható továbbá, hogy két fuzzy halmaz csúcspontja között a függvény „majdnem monoton” [53]. Vagyis az egyenlő szárú háromszögekkel megvalósított fuzzy irányítási rendszer viselkedése egy intervallumokon alapuló crisp szakértő rendszerével azonos.

Vizsgáljuk meg, mennyiben módosul a (7.28) kifejezés, ha az elterjedtebb súlypont defuzzifikációs eljárást alkalmazzuk! Ekkor a megfigyelés és a következtetés között az

$$y_{\text{COG}}^* = \left(i + \frac{1}{2}\right) b + \frac{w_1(1 - w_1)(1 - 2w_1)}{(1 + w_1 - w_1^2)(1 + 2w_1 - 2w_1^2)} \cdot \frac{b}{2}, \quad (7.29)$$

összefüggés áll fenn, ahol ismét felhasználtuk a (7.25) egyenletet (részletesen ld. például [108, 109]). A (7.29) kifejezésnek szerkezete (7.27)-éhez hasonló:

$$y_{\text{COG}}^* = c_1 i + c'_1 \frac{c_2 + c_3 x^* + c_4 (x^*)^2}{c_5 + c_6 x^* + c_7 (x^*)^2 + c_8 (x^*)^3}, \quad (7.30)$$

azzal a különbséggel, hogy a racionális törtfüggvény-rész számlálójának és nevezőjének foka egyvel magasabb. Az eredményül kapott függvény viselkedése (7.28) kifejezésével azonos. Korlátossága a geometriai interpolációból következik, a csúcspontok közti monotonitást pedig a súlypont módszer átlagoló jellege biztosítja.

A két kifejezés — (7.27) és (7.29) — szerkezetének hasonlósága felveti a kérdést, hogy mekkora az eltérés a két defuzzifikációs módszer által előállított következtetés között. Egyszerű számításokkal megmutatható, hogy egyenlő szárú háromszög alakú halmazok esetén ez a különbség nem több, mint a konzekvens halmazok tartójának (b) 2%-a [108, 109], amely igen csekélynek mondható figyelembe véve a fuzzy érvelési rendszerekben eredendően meglévő bizonytalanságot. Az eredmény tükrében érthető, hogy a kisebb számításigénnyel rendelkező súlypont módszert a gyakorlati alkalmazások többségében előnyben részesítik, hiszen a két defuzzifikációs eljárás közel azonos eredményt ad.

Szintén érdemes megvizsgálni, hogy a következtetési algoritmusban alkalmazott t-norma milyen módon befolyásolja a következtetés függvényének szerkezetét. Ha az eljárásban az algebrai t-normát használjuk (LARSEN-típusú következtetés [125]), akkor az explicit függvények az alábbiak szerint módosulnak. Ha a geometriai középpont defuzzifikációs módszert alkalmazzuk, akkor az

$$y_{\text{COA}}^* = \left(i + \frac{1}{2}\right) b + \frac{7w_1 w_2^2 - 7w_2 w_1^2 + 6w_2^3 - 6w_1^3}{6(2 - w_1 w_2)} \cdot b \quad (7.31)$$

kifejezést kapjuk, a súlypont módszer pedig az

$$y_{\text{COG}}^* = \left(i + \frac{1}{2}\right) b + (1 - 2w_2) \cdot \frac{b}{2} \quad (7.32)$$

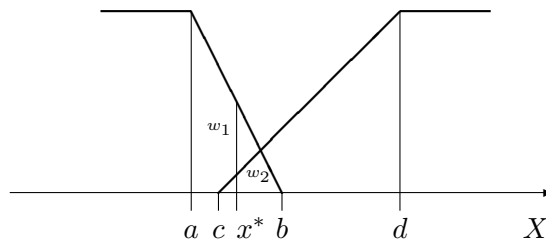
összefüggést adja. A (7.31) kifejezés hasonló szerkezetű, mint a MAMDANI-eljárás alkalmazásával kapott egyenletek, viszont a (7.32) összefüggés szerkezete lényegesen egyszerűbb: a kimenet a megfigyelés lineáris függvénye. A két módszer által kapott eredmény közti eltérés valamelyest nagyobb, a konzekvens halmazok tartójának 6%-a, de még így sem jelentős.

Összességében megállapítható, hogy a következtetési algoritmusban alkalmazott trianguláris normától és a defuzzifikáló eljárástól függetlenül az irányítás explicit függvénye a csúcspontok között közelítően lineáris, a közelítés hibája pedig egy monoton, lineáristól nem lényegesen eltérő racionális törtfüggvénnyel adható meg. Ettől csak a LARSEN-módszer és a súlypont eljárás kombinációja tér el, ahol az eljárás pontosan szakaszonként lineáris explicit függvénnyel jellemezhető [108, 109].

7.6.2. Explicit függvények trapéz alakú szabályok esetén

Bár a gyakorlati alkalmazásokban is előszeretettel használnak egyenlő szárú háromszög alakú tagsági függvényeket (ld. 9. fejezet), a fuzzy halmazok alakjának ilyen szabályossága azonban általában nem teljesül. Még abban az esetben is, ha a kiindulási rendszer szabályos halmazokat tartalmaz, a szabályok hangolásával az antecedens és konzekvens halmazok alakja módosulhat [30]. A tagsági függvények formája még szabálytalanabb lehet abban az esetben, ha a szabályokat kvalitatív modellezés alapján generáljuk [175]. Éppen ezért indokolt megvizsgálni, hogy hogyan módosulnak a fuzzy irányítási rendszerek által megvalósított explicit függvények, ha a szabályokban szereplő fuzzy halmazok alakja általánosabb. Figyelembe véve ugyanakkor azt, hogy a számítási igény jelentősen növekszik, amennyiben a szabályok alakja tetszőleges, a továbbiakban szakaszonként lineáris, azon belül is az általános trapéz alakú tagsági függvényeket tartalmazó rendszereket tanulmányozzuk.

Ezenkívül továbbra is tegyük föl azt, hogy minden megfigyelés legfeljebb két szabályt aktivál. Ha pontosan két szabály tüzel, akkor az x^*



7.18. ábra. Az illeszkedés mértékének meghatározása általános trapéz alakú tagsági függvények esetén, ha pontosan két szabály tüzel

megfigyelés és a tüzelő szabályok antecedenseinek illeszkedési mértékét a

$$w_1 = \frac{a - x^*}{b - a} + 1 \text{ és } w_2 = \frac{x^* - c}{d - c}$$

egyenletek határozzák meg, ahol a és d , illetve c és b rendre a trapézok magjának és tartójának szélsőértékeit jelölik (részletesen ld. a 7.18. ábrán). Abban az esetben, ha $a \leq c \leq x^* \leq b \leq d$ nem áll fenn, akkor egyik vagy mindkét illeszkedési mérték 0-ra vagy 1-re változik, hiszen ha $x^* < a$, akkor $w_1 = 1$, ha $x^* < c$, akkor $w_2 = 0$, ha $x^* > b$, akkor $w_1 = 0$, valamint ha $x^* > d$, akkor $w_2 = 1$. Mivel a következtetés kiszámítása meglehetősen hosszadalmas még az egyszerűbb súlypont módszer esetén is, ezért ennek ismertetésétől eltekintünk. A részletek megtalálhatók a [108, 109] munkákban.

A következtetésre kapott formula igen bonyolult és nehezen áttekinthető, strukturális felépítése viszont hasonló az előző pontban kapott kifejezésekéhez:

$$y_{\text{COG}}^* = c_1 + c_2 x^* + \frac{c_3 + x^*}{c_4 + c_5 x^* + c_6 (x^*)^2}. \quad (7.33)$$

A (7.33) kifejezés a korábbi explicit függvényekhez képest két lényeges eltérést mutat. Egyrészt a lineáris rész ezúttal nem konstans, azaz nemcsak attól függ a következtetés értéke, hogy mely két csúcspont közötti intervallumba esik a megfigyelés, hanem ennek intervallumon belüli pozíciója is befolyásolja azt; másrészt a kifejezés több paramétert tartalmaz (itt jegyezzük meg, hogy a c_i ($i \in [1, 6]$) konstansok értéke 12 paramétertől függ, melyek az aktivált antecedens és konzekvens halmazokat írják le.) A (7.33) kifejezés racionális törtresze nem eliminálható, ugyanis c_5 és c_6 akkor és csak akkor zérus, ha a konklúziók nem fuzzy halmazok. Nem egyszerűsödik az összefüggés alakja lényegesen akkor sem, ha a rendszerben szereplő nyelvi fogalmakat általános háromszög alakú halmazokkal írjuk le. Mindazonáltal a szabályok geometriai formájának segítségével belátható, hogy a (7.33) következtetés monoton módon változik két csúcspont között. Hasonló eredményt kapunk akkor is, ha csak egy szabály tüzel [108, 109].

Mindeddig csak egyváltozós rendszereket elemeztünk. A fuzzy irányítók explicit függvényeinek jellegére többváltozós esetben is hasonló formulákat kaphatunk. A kétdimenziós eset vizsgálatával a [186, 187] közlemények foglalkoznak. Ezen eredmények felhasználásával — melyek lényegüket tekintve megegyeznek az egydimenzióssal, azaz racionális törtfüggvények — tetszőleges dimenziószámra általánosítható formulák nyerhetők. Az explicit függvényekről igen átfogó és részletes képet nyújt a [93] tanulmány 5.1. szakasza is.

7.6.3. Az explicit függvények jelentősége

Az előző két pontban megmutattuk, hogy ha adott a fuzzy következtetési rendszer algoritmus, akkor lehetőség van az irányítási függvény $y^* = f(x^*)$ alakban történő explicit megadására, s így módon az irányító nemfuzzy függvénnyel való helyettesítésére. Bár részletesen csak az egyváltozós rendszereket vizsgáltuk, nyilvánvaló, hogy valamely következtetési eljárás explicit függvényének meghatározása a konzekvens halmazok alakjától és a w_i illeszkedési mértékek értékétől függ. Ennek alapján a módszer könnyen általánosítható többdimenziós rendszerekre, sőt olyan esetekre is kiterjeszthető, amikor az illeszkedési mértékek nem teljesítik a (7.25) feltételt, vagy egyszerre több mint két szabály is tüzel. Ha biztosítható, hogy az esetleg többdimenziós w_i a megfigyelés (x^*) lineáris függvénye, akkor az egydimenziós esetre kapott $y^* = f(x^*)$ függvény rangja többdimenzióban is megőrződik, tehát lineáris vagy polinomiális explicit függvény típusa többdimenzióban is ugyanaz lesz, azonos ranggal.

Az explicit függvényekre kapott eredmények néhány kutatót arra inspirálták, hogy a tagsági függvények elhagyásával olyan eljárást javasoljanak, mely csak a fuzzy halmazok magját (középpontját) használja fel [123], holott a fuzzy következtetési eljárások legnagyobb előnye nem a megvalósított explicit függvények jellegében rejlik, hanem abban hogy felhasználóbarát technológiát kínál irányítási, döntéshozói, vagy más típusú problémákat megoldó rendszerek létrehozására és beállítására anélkül, hogy az irányított folyamat vagy rendszer matematikai háttere analitikusan ismert volna. Ebben a fejlesztési és hangolási folyamatban viszont a tagsági függvények szerepe rendkívül jelentős.

Az explicit formulák ismerete a fuzzy érvelési és következtetési rendszerek további analitikus vizsgálatában jelent segítséget, alkalmazásukkal a behangolt, beállítás utáni rendszerek futási ideje is csökkenthető valame-lyest.

7.7. Fuzzy irányítási rendszerek univerzális közelítő tulajdonsága

A szabálybázisos fuzzy következtetési rendszerek gyors elterjedése és sikere felvetette azt a kérdést, hogy mi a matematikai háttere és magyarázata a fuzzy irányítási rendszerek használhatóságának. Meglepő módon az 1990-es évek elejéig igényes matematikai vizsgálat nem történt. Az első ilyen irányú eredményre is gyakorlati tapasztalatok vezettek, melyek azt

mutatták, hogy bármely nemlineáris, kompakt halmazon definiált folytonos függvény fuzzy irányítási rendszerrel tetszőlegesen jól közelíthető.

Ezt az empirikus eredményt nem sokkal később elméleti oldalról is alátámasztották. 1992-ben WANG [190] és KOSKO [113, 114] egymástól lényegében függetlenül kimutatták, hogy a MAMDANI-féle fuzzy irányító kompakt értelmezési tartományban elvileg univerzális függvényapproximátorként működik. Hasonló állításokat láttak be különböző fuzzy következtető rendszerekre a [28, 29, 192] munkákban is. Eredményeiket NGUYEN és KREINOVICH [142] általánosították tetszőleges dimenziószámra, majd CASTRO adott teljesen általános megfogalmazást [35]. Ezen eredmények alapján szakmai körökben elterjedt az a vélemény, hogy az alkalmazásokban a fuzzy irányítási rendszerek nemlineáris transzferfüggvények univerzális uniform közelítőjeként működnek, s előnyüket elsősorban ez a tulajdonság jelenti.

Az itt felsorolt eredmények közös vonása az, hogy a tetszőlegesen pontos (azaz formálisan: minden $\varepsilon > 0$ -nál kisebb hibájú) közelítést nagyon sok, sűrűn elhelyezkedő szabályt tartalmazó szabálybázisok segítségével érik el. Ez KOSKO munkájában — amely a MAMDANI-féle irányítókat vizsgálta — például azt jelenti, hogy az előírt ε -nál kisebb hibájú közelítéshez a szomszédos antecedens halmazok távolsága nem haladhatja meg a

$$|y_i - y_{i+1}| \leq \frac{\varepsilon}{2p - 1}$$

értéket, ahol p a tüzelő szabályok száma, tehát $2p - 1$ értéke általában 3. Így a jó közelítés megvalósításához túlzottan nagyméretű szabálybázis szükséges.

WANG és MENDEL [190, 192] a STONE-WEIERSTRASS-tétel felhasználásával egzakt bizonyítást adtak GAUSS- (vagyis harang-) görbe alakú tagsági függvények és LARSEN-következtetést használó irányítási algoritmus univerzális közelítő tulajdonságra. Eredményükben nemcsak a közelítéshez felhasznált szabályok száma, hanem még az alkalmazott tagsági függvények tartója sem korlátos.

Valójában ezek az állítások nem a fuzzy irányítási rendszerek alkalmazhatóságát támasztják alá. Elegendően nagyszámú szabály esetén ugyanis megmutatható [108, 109], hogy a fentiekkel ekvivalens állítás igaz nem fuzzy (crisp) szakértő rendszerek esetén is, ahol a szabályok

$$R_i : \text{Ha } x_{ij} \in [x_{jk}, x_{j,k+1}) \text{ akkor } y = b_i$$

alakúak, és a x_{jk} az alaphalmaz elegendően finom felosztása. Tehát a függvényapproximációs tulajdonság háttérében nem a tervezési paraméterek változatossága, hanem a szabályok minden határon túl növekvő száma áll.

Ha a szabályokban szereplő halmazok alakja nem rögzített, akkor lényegesen kisebb számú szabállyal is *pontosan* elő lehet állítani tetszőleges folytonos függvényt fuzzy következtetési rendszer transzferfüggvényeként. Ebben az esetben — megfelelő tagsági függvények használatával — legalább két szabály szükséges. Ha a halmazok konvexitását is feltételként szabjuk, a szabályok száma akkor sem haladja meg a közelített függvény inflexiók pontjai számának kétszeresét plusz kettőt [14]. Ezek azonban csak egzisztenciaeredmények, és a konstrukcióra nem adnak módszert, sőt a megfelelő tagsági függvények és a transzferfüggvény közvetlen előállítása gyakorlatilag azonos nehézségű feladat, s ha a rendszer analitikus modellje nem ismert, ez szinte lehetetlen. Ez a megoldás tehát a transzferfüggvény előállításának nehézségét egyenesen a tagsági függvények megvalósítására viszi át.

Igen érdekes, hogy a fuzzy irányítási rendszerek univerzális approximátor voltának kérdése belehelyezhető egy sokkal tágabb problémakör kontextusába. 1900-ban a II. Matematikai Világkongresszuson D. HILBERT (1862–1943), a híres német matematikus 23 érdekes matematikai problémát fogalmazott meg, melyek megoldása véleménye szerint a XX. század matematikusainak fontos feladat lesz. A problémák közül a 13. sejtésben azt feltételezte, hogy létezik olyan folytonos háromváltozós függvény, amelyik nem dekomponálható folytonos, kétváltozós függvények véges szuperpozíciójaként.

A helyzet a fuzzy és más lágy számítási rendszerek (például neurális hálózatok) szempontjából akkor vált igazán érdekessé, amikor 1957-ben A. N. Kolmogorov bebizonyította, hogy ez a hipotézis nem igaz [112]. Cáfolatában egy sokkal általánosabb állítást bizonyított, amikor belátta, hogy nem csupán minden háromváltozós függvény, hanem tetszőleges n -változós folytonos függvény is felírható mindössze egyváltozós függvények segítségével.

7.1. Tétel. Minden $n \geq 2$ egész esetén létezik $n(2n + 1)$ olyan folytonos, monoton növekvő, egyváltozós, a $[0,1]$ intervallumon értelmezett függvény, melyek segítségével tetszőleges valós, n -változós, folytonos $f : [0,1]^n \rightarrow \mathbb{R}$ függvény az

$$f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \phi_q \left(\sum_{p=1}^n \psi_{pq}(x_p) \right) \quad (7.34)$$

alakban előállítható.

Kolmogorov tételét továbbfejlesztve 1965-ben SPRECHER megmutatta [168], hogy az ismeretlen leképezés a (7.34) egyenletben használt függvény-rendszerek helyett mindössze két nemlineáris függvénnyel is előállítható.

DE FIGUEIREDO 1980-ban megmutatta, hogy KOLMOGOROV tétele vonatkoztatható többszintes előrecsatolt neurális hálózatokra, s így ezek univerzális approximátornak tekinthetők [39]. KŰRKOVÁ 1992-ben bizonyította, hogy a feladat általánosított szigmoid függvényekkel megoldható [119]. A szakasz elején ismertetett fuzzy irányítási rendszerekre vonatkozó tételek lényegében ezekkel az eredményekkel rokoníthatók [35]. Belátható azonban, hogy mindezek a tételek minden határon túl növekvő sokaságú közelítő függvényekre vonatkoznak és az exponencialitás nem küszöbölhető ki [120]. A probléma abból adódik, hogy az approximált függvény bonyolultsága — hasonlóan a korábban említett [14] esetéhez — a nagyon speciális alakú nemlineáris függvények bonyolultságára transzformálódik, amelyek megvalósítása exponenciális méretű hálózatot igényel.

Ha mind a tagsági függvények számát, mind azok alakját korlátozzuk, akkor korántsem marad igaz az univerzális közelítő tulajdonság. Az univerzális közelítés másképpen úgy fogalmazható meg, hogy a közelítő függvények által generált függvénytér sűrű a közelített függvények terében. Ezzel szemben SUGENO-irányítókra MOSER belátta [134, 135], hogy korlátozott szabálysorszám esetén a SUGENO-irányítók által generált tér a folytonos függvények L_p terében *sehol sem sűrű*. A „sehol sem sűrűség” a topológiában használt fogalom. Azt jelenti, hogy a halmaznak nincs belső pontja, vagyis annak tetszőleges kis környezetében van a halmazba nem tartozó pont. Eszerint nemcsak hogy nem lehet korlátozott szabálysorszámú SUGENO-típusú következtető rendszerekkel tetszőleges folytonos függvényt közelíteni, de az ezek által generált tér a folytonos függvények terében „majdnem diszkrét”. A tételt TIKK általánosította olyan irányítókra [180, 181], ahol a szabályok konzekvense *tenzorszorzat* alakban felírható (ún. T-irányítók: ide tartoznak a TAKAGI–SUGENO- [178], valamint a TAKAGI–SUGENO–KANG-típusú [173] irányítók is), és természetesen a szabályok száma korlátozott.

Ezek az eredmények azt a véleményt támasztják alá, mely szerint a fuzzy irányítási rendszerek sikeres alkalmazhatósága nem az univerzális közelítő voltukban rejlik, hanem azon múlik, hogy segítségükkel bonyolult, nemlineáris, vagy akár ismeretlen viselkedésű rendszerek is jól modellezhetők, mivel az ilyen rendszerek komplexitása fuzzy modellezés alkalmazásával jelentősen csökkenthető.

8. fejezet

Fuzzy redukciós módszerek

8.1. Klasszikus fuzzy következtető algoritmusok komplexitása

8.1.1. Algoritmusok bonyolultsága

Az algoritmusok lehetséges jellemzői közül gyakorlati szempontból az a legfontosabb, hogy milyen módon függ az adott probléma mérete és a probléma megoldásához szükséges idő-, valamint tárigény [1]. Így például minden problémához egy egész számot rendelhetünk hozzá, amely a probléma méretét, azaz a bemeneti adatok nagyságát jellemzi. Mátrixszorzás esetén ez a mennyiség lehet a szorzandó mátrixok legnagyobb dimenziója, gráfokkal kapcsolatos problémák esetén a csúcsok vagy az élek száma, stb.

Az algoritmus *időkomplexitása* (másképpen *időigénye*) a probléma méretének függvényében adható meg. Az időigény határfüggvényét *aszimptotikus időkomplexitás*nak nevezzük. Hasonló módon határozható meg a *tárkomplexitás* (tárigény) és az *aszimptotikus tárkomplexitás* fogalma is. Ha az algoritmus során végrehajtandó műveletekhez szükséges időt egységesen azonosnak tekintjük, akkor *uniform*, ellenkező esetben *logaritmikus* bonyolultságról beszélünk.

A továbbiak során a vizsgált algoritmusok uniform bonyolultságú aszimptotikus komplexitását vizsgáljuk, mely fogalmat röviden az algoritmus komplexitásának vagy bonyolultságának hívunk. Eszerint ha valamely algoritmus egy n méretű problémát cn^2 lépésben old meg (c pozitív konstans), akkor az algoritmus bonyolultsága $O(n^2)$ (ejtsd: nagy ordó n^2). A pontos definíció szerint a $g(n)$ függvény nagyságrendje $O(f(n))$, ha létezik olyan c konstans, amelyre $g(n) \leq c \cdot f(n)$ fennáll majdnem minden nemnegatív n értékre.

A számítástechnikában tapasztalt ugrásszerű műveletisebesség-növekedés ellenére bizonyos algoritmusok alkalmazhatóságának nagy komplexitásuk eleve korlátokat szab. Tekintsük az alábbi példát! Tegyük fel, hogy számítógépünk másodpercenként 1000 művelet elvégzésére képes.

A 8.1. táblázatból látható, hogy a különböző időigényű algoritmusok közül milyen méretűt képes végrehajtani egy másodperc, egy perc vagy egy óra alatt.

8.1. táblázat. Az algoritmus időigénye és a probléma méretének korlátjára vonatkozó összefüggések [1]

Algoritmus	Időigény	Maximális problémaméret		
		1 másodperc	1 perc	1 óra
A_1	n	1000	$6 \cdot 10^4$	$3,6 \cdot 10^6$
A_2	$n \log n$	140	4893	$2,0 \cdot 10^5$
A_3	n^2	31	244	1897
A_4	2^n	9	15	21
A_5	$n!$	6	8	10

Tegyük fel, hogy a következő számítógép-generáció tízszer gyorsabb a jelenleginél. A 8.2. táblázat megmutatja, hogy a sebességnövekedés hatására a megoldható problémák mérete miként növekszik a különböző komplexitású algoritmusok esetén.

8.2. táblázat. Tízszeres sebességnövekedés hatása a megoldható problémák méretére [1]

Algoritmus	Időigény	Maximális problémaméret a sebességnövekedés	
		előtt	után
A_1	n	s_1	$10s_1$
A_2	$n \log n$	s_2	$\approx 10s_2$
A_3	n^2	s_3	$3,16s_3$
A_4	2^n	s_4	$s_4 + 3,3$
A_5	$n!$	s_5	$s_5 + 10/n$

Megfigyelhető, hogy a tízszeres sebességnövekedés az A_4 algoritmussal megoldható probléma méretét csak hárommal, míg az A_3 algoritmussal megoldhatóét több mint háromszorosára növeli.

ERDŐS PÁLTÓL (1913–1998) származik állítólag az alábbi példa, amely érzékletesen szemlélteti a sebességnövekedéssel elérhető problémaméret-

növekedés elvi határát. Tegyük fel, hogy az elméletileg lehetséges leggyorsabb és leghatalmasabb számítógép, melyben a világ összes atomja (10^{80}) fénysebességgel ($c = 3 \cdot 10^8$ m/s) az univerzum kezdete óta (kb. 10 milliárd év $= 10^{10} \cdot 365 \cdot 24 \cdot 3600 = 3,15 \cdot 10^{17}$ s) egy $n!$ nagyságrendű probléma megoldásán dolgozik. Még ez a gép is csak $9,45 \cdot 10^{105}$ számú műveletet, vagyis $n = 73$ méretű problémát lett volna képes megoldani ezalatt az idő alatt.

8.1.2. Klasszikus algoritmusok bonyolultsága

Térjünk rá az alapvető fuzzy irányítási algoritmusok bonyolultságának tárgyalására. Vizsgáljuk meg először, hány szabály szükséges a k dimenziós $X = X_1 \times \dots \times X_k$ bemeneti alaphalmaz teljes ε -lefedéséhez ($\varepsilon > 0$). Legyen az egyes X_i bemeneti halmazok lefedéséhez felhasznált fuzzy halmazok száma legfeljebb T . Ekkor az alaphalmaz teljes ε -lefedéséhez

$$|R| = O(T^k) \quad (8.1)$$

szabály szükséges, amely rendkívül magas érték, amennyiben k értéke nem kicsiny. Még abban a szélsőséges esetben is, amikor állapotváltozónként mindössze két nyelvi változót adunk meg (két különböző állapot mindenképpen szükséges, ugyanis ellenkező esetben a változó hatástalan lenne, és a modellből ki kellene zárni), a teljes lefedéshez 2^k , azaz exponenciális nagyságrendű szabálysorszám szükséges.

A klasszikus fuzzy következtető eljárások bonyolultságát először KÓCZY vizsgálta [88, 91]. Ennek alapján a ZADEH-féle CRI eljárás bonyolultságára, amely a fuzzy relációként felírt szabálybázisnak és a megfigyelésnek a $(k + 1)$ dimenziós hipertérben képzett metszetéből számolja a következtetést, a

$$C_{Z,\text{idő}} = O(r \cdot T^{k+1}) = O(T^{2k+1}) \quad (8.2)$$

kifejezés adható, ahol r a szabályok száma (ld. (8.1)). A tárkomplexitásra hasonlóan exponenciális bonyolultságú érték adható:

$$C_{Z,\text{tár}} = O(r(k + 1)T) = O((k + 1)T^{k+1}).$$

Ennél lényegesen kisebb, de még mindig exponenciális komplexitással rendelkeznek a MAMDANI-típusú következtető eljárások (MAMDANI, LARSEN, SUGENO), mivel ezek nem a szorzattéren, hanem annak vetületein számolják a konklúziót, amellyel számottevő idő takarítható meg:

$$C_{M,\text{idő}} = O(r(k + 1)T) = O((k + 1)T^{k+1}). \quad (8.3)$$

Itt említjük meg, hogy kis módosítással csökkenthető a ZADEH-féle eljárás bonyolultsága (érzékenységeinek megőrzése mellett), ha az alaphalmaz kompakt [87, 88]. Ez az eljárás gyakorlati alkalmazásokban azonban nem terjedt el, ugyanis általában elégséges a MAMDANI-jellegű algoritmusok által nyújtott érzékenység és pontosság.

8.2. Csökkentési lehetőségek

Az előző pontban vázolt komplexitási tényező jelentősen korlátozza a fuzzy irányítási algoritmusok valós idejű alkalmazhatóságát bonyolult, sokdimenziós feladatok megoldására. A fuzzy következtető eljárások exponenciális bonyolultsága miatt gyakorlati alkalmazásokban az állapotváltozók száma szinte sosem haladja meg a tizet, de a jellemző érték többnyire öt alatt van. A problémát az is fokozza, hogy a fuzzy algoritmusokat alapvetően matematikailag nem ismert rendszerekhez alkalmazzák, és éppen ezért nincsen kidolgozott matematikai háttér a szabályok számának és helyének valamilyen optimalizálási kritérium szerinti meghatározására. A matematikai modell hiányában sokszor a szükségesnél jóval több antecedens halmaz kerül felhasználásra, ami bár jobb közelítést eredményezhet, ám ezáltal nagymértékben nő a szabálybázisban lévő fölösleges információ. Így az egyre jobban elterjedő és egyre szélesebb problémakörben alkalmazni kívánt fuzzy következtetésen alapuló eljárások számítási idő- és tárkomplexitásának csökkentése fontos kérdéssé vált, aminek felismerése a 90-es évek elejétől egyre több kutatót sarkallt alternatív redukciós eljárások kidolgozására.

A redukciós módszerek jellegük alapján két fő csoportba sorolhatók. Az elsőbe azok az eljárások tartoznak, melyek olyan új vagy módosított következtetési módszert alkalmaznak, amelynek számítási bonyolultsága kisebb mint az eredeti algoritmusé. Ebbe a csoportba tartozik a ZADEH-féle eljárás — már korábban említett — kompakt alaphalmazon működő változata [87, 88]. Másik megoldás a STOICA által javasolt, szakaszonként lineáris nyelvi változók esetén alkalmazható módszer, mely α -vágatokként számítja közvetlenül a defuzzifikált eredményt, s így jelentősen csökkenti az időigényt [169]. Szintén csökkentett számítási eljárás a YU és BIEN által javasolt minimum távolságalapú eljárás [204].

A komplexitás csökkentésének másik módja a szabályok (r), illetve a nyelvi változók (T) számának, összefoglalóan a szabálybázisnak a redukálása. Amint azt korábban említettük, a szabálysorszám minimumának elvi alsó korlátja 2^k , ami ugyan továbbra is exponenciális kifejezés, de még így

is jelentős csökkenést eredményez, különösen akkor, ha a redukció előtt T értéke nagy volt. További lehetőség a kitevő (k) csökkentése az esetleges redundáns állapotváltozók elhagyásával, illetve összevonásával.

Ebbe a csoportba tartozó eljárások elsősorban nem új következtető algoritmusok, hanem a már behangolt szabálybázisok információtartalmának tömörítésére, redundanciájuk megszüntetésére alkalmas módszerek. Ezen eljárásoknak akkor van nagy jelentőségük, ha a szabálybázis előre elkészíthető, alkalmazása közben további hangolást nem igényel és így a tömörítés után már kisebb memória- és számításkapacitással rendelkező módszerekben is alkalmazhatók. Ebbe a csoportba sorolható a BRUINZEL és munkatársai által ismertetett módszer, melynek célja egyes bemeneti változók összevonása [27]. Szinguláris értékfelbontáson (SVD) alapuló információ-tömörítő módszert javasol a szabálybázis redukálására WANG és munkatársai [191], és YAM [200] eljárása SUGENO-típusú irányítás esetén. Ezen eredmények tetszőleges szabálybázisra történő általánosítása található az [8, 9, 12, 201] munkákban. Szintén szinguláris értékfelbontáson alapul a különlegesen nagy szabálybázisok tömörítését elvégző algoritmus [13], amely csak az aktuálisan tüzelő szabályokhoz kapcsolódó információkat csomagolja ki futás közben, amivel lényegesen csökkenti a tárigényt. Kedvező esetben a módszer alkalmazásával kiiktatható a háttértárból történő adatbeolvasás, s mivel a legtöbb számítógépes architektúrán az operatív memóriában tárolt információ sokkal gyorsabban elérhető, ezáltal a futási idő is számottevően csökkenhet.

A szabálybázisredukciós-módszereken belül külön figyelmet érdemel a szabályok hierarchikus rendezését javasoló technika, amelyet először SUGENO alkalmazott a vezetőnélküli helikopter vezérlőrendszerének irányítására [172, 174]. A hierarchikus szabálybázisokra a fejezet végén külön szakaszban térünk vissza.

Mindkét csoportba besorolhatóak a fuzzy szabályinterpolációs algoritmusok, melyeket a következő szakaszokban részletesen tárgyalunk.

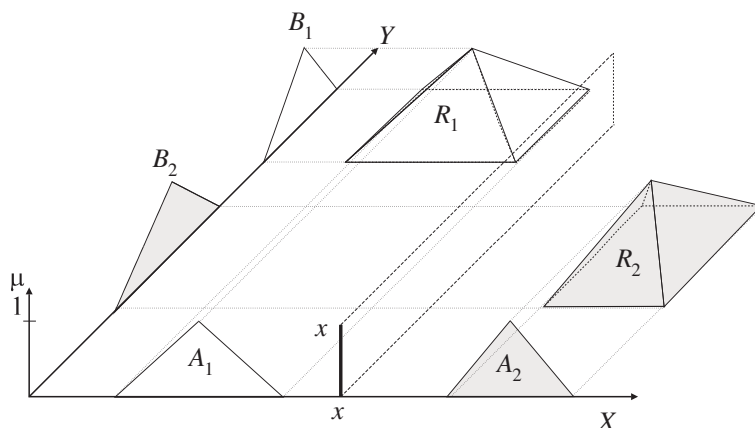
8.3. Ritka szabálybázisok

A szabálysám csökkentésének egyik módja a bemeneti halmazokon megadott nyelvi változók, azaz a szabályantecedensek számának (T) mérséklése. Fennáll a lehetősége annak, hogy a szabályantecedensek elhagyásával olyan szituációhoz jutunk, amikor a bemenet ε -fedettsége már nem áll fenn semmilyen pozitív ε értékre sem, vagyis valamely bemenetnek van legalább egy olyan pontja, amelyhez nem rendelhető egyetlen (megtartott)

szabály sem. Az ilyen „lyukas”, nem teljes fedettséget biztosító szabálybázist *ritka szabálybázis*nak nevezzük. Ritka szabálybázisok esetén létezik olyan A^* megfigyelés, amelyre

$$A^* \cap \bigcup_{i=1}^r \text{supp}(A_i) \subset X = \emptyset, \quad (8.4)$$

ahol az i -edik ($1 \leq i \leq r$) szabály $R_i : A_i \rightarrow B_i$ alakú (ld. 8.1. ábra). Ebben az esetben a klasszikus (ZADEH-, MAMDANI-féle) következtetési eljárások alapján nem lehet a konklúziót meghatározni, ezért ezek az eljárások itt egyáltalán nem alkalmazhatók.



8.1. ábra. Ritka szabálybázis: a megfigyelés a szabályokkal diszjunkt

Érdemes megjegyezni, hogy a szabálybázis ritkításán túl más okok is vezethetnek ritka szabálybázisokhoz. Függetlenül attól, hogy milyen eljárást alkalmazunk valamely szabálybázis létrehozására, ha a modellezett rendszerről csak részleges információ áll rendelkezésre, az eredményként kapott szabálybázis eleve lehet ritka. A szabálybázis összeállításához ZADEH több tanulmányában közvetlenül a szakértői tudás felhasználását javasolta. Újabban viszont egyre gyakrabban alkalmaznak például neurális hálózat alapú tanulási technikákat a szabálybázis megalkotásához, melyek alapjául a rendelkezésre álló numerikus mintaadatok szolgálnak. Ez utóbbi esetben az eredményezhet ritka szabálybázist, ha a mintaadatok nem kellően reprezentálják a bemeneti paramétereket, az előbbi esetben pedig természetesen az, ha a szakértő nem rendelkezik kellő információval egyes rendszerkonfigurációkról.

Ritka szabálybázishoz juthatunk *hangolás* eredményeként is (ld. 8.2. ábra). A szabályantecedensek eltolása és/vagy zsugorítása következtében előfordulhat olyan szituáció, mikor a hangolt modell lyukakat tartalmaz, noha a kiinduló antecedenshalmaz még teljes fedettséget biztosított [30].

Hierarchikus rendszerek esetén definiálható két szabálybázis távolsága úgy, hogy köztük lyuk legyen [97].

Ahhoz, hogy ritka szabálybázisokkal kapott szabálysámcsökkenés ténylegesen kiaknázható legyen, teljesen új következtetési eljárás szükséges. A ritka szabálybázisokon is alkalmazható technika alapötlete az, hogy a lyukak helyén a szomszédos szabályok segítségével közelítő következtetést határozunk meg. Ezt az eljárást (*fuzzy*) *szabályinterpoláció*nak nevezzük.

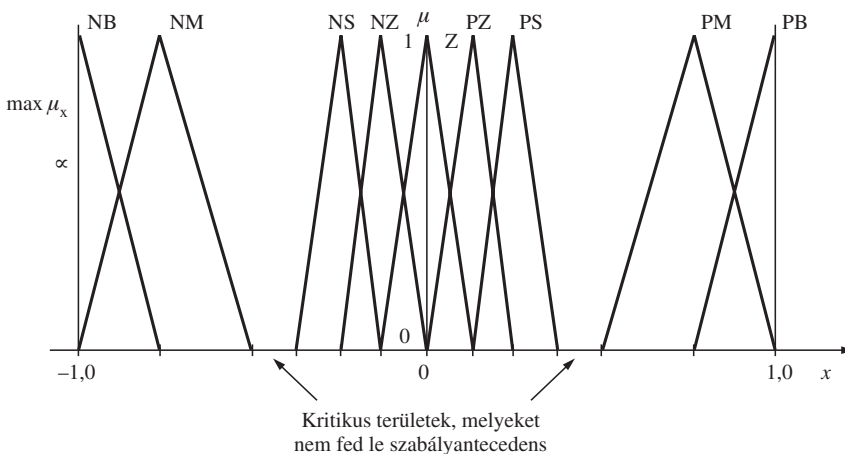
8.4. Fuzzy szabályinterpoláció

A szabályok közti interpoláció egy egyszerű példán jól szemléltethető. Tegyük fel, hogy adott két szabály:

R_1 : Ha A_1 = „a paradicsom piros”, akkor B_1 = „érett”,

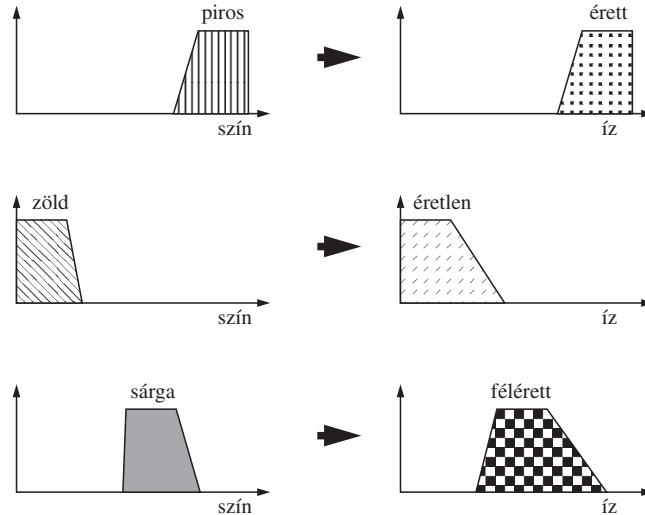
R_2 : Ha A_2 = „a paradicsom zöld”, akkor B_2 = „éretlen”.

Legyen a megfigyelés A^* = „a paradicsom sárga”, melynek metszete mindkét antecedens halmazzal üres, így a klasszikus algoritmusok nem alkal-



8.2. ábra. Hangolás eredményeként keletkezett ritka szabálybázis [30]

mazhatók, de intuitíve ismerjük a B^* = „félérett” választ (ld. 8.3. ábra). Ezt hivatott a fuzzy szabályinterpoláció formálisan is megadni.



8.3. ábra. Példa fuzzy szabályinterpolációs következtetés alkalmazására

Természetesen az interpolációs technika nem alkalmazható minden esetben. Például, ha rendszerünk az

R_1 : Ha A_1 = „a közlekedési lámpa piros” akkor B_1 = „állj meg (hajts 0 sebességgel)”

R_2 : Ha A_2 = „a közlekedési lámpa zöld” akkor B_2 = „hajts át a maximális sebességgel”

szabályokat tartalmazza, akkor az A^* = „a közlekedési lámpa sárga” megfigyelés esetén hibás volna a B^* = „hajts tovább közepes sebességgel” következtetésre jutni. A példa szemlélteti, hogy a fuzzy interpoláció akkor alkalmazható, ha az antecedens és konzekvens univerzumok strukturáltak és metrikusak, ahol a távolság- vagy hasonlóság-mérték és a halmazok közötti rendezés definiálható.

8.4.1. A lineáris (KH)-szabályinterpolációs eljárás

Az előbbi példa jól mutatta az interpoláció alkalmazásának feltételét, azaz hogy a szabályantecedensek és -konzekvenszek valamely részben rendezési reláció segítségével összehasonlíthatók legyenek, ezért először a

fuzzy halmazok, s azon belül is a lineáris interpolációban alkalmazott konvex és normális fuzzy (CNF) halmazok részben rendezését vizsgáljuk [99].

8.1. Tétel. Legyen $X = \times_{i=1}^k X_i$ -ben adott $a \leq = \times_{i=1}^n \leq_i$ reláció úgy, hogy $\leq_i$ az X_i -beli rendezés. Ekkor létezik $a \leq X$ -beli részben rendezés abban az értelemben, hogy $x_1 \leq x_2$ akkor és csak akkor, ha minden i -re: $x_{1i} \leq_i x_{2i}$.

Ennek segítségével definiálhatjuk a CNF halmazokon értelmezett részben rendezést:

8.2. Tétel. Legyen $\tilde{C}(X_i)$ az X_i univerzum $\tilde{P}(X_i)$ hatványhalmazának CNF halmazait és az üres halmazt tartalmazó részhalmaza. Ekkor minden i esetén létezik $\tilde{C}(X_i)$ -ben egy $\prec_i$ részben rendezés úgy, hogy $A_{1i} \prec_i A_{2i}$ akkor és csak akkor, ha minden $\alpha \in [0,1]$ -re

$$\begin{aligned} \inf\{A_{1i\alpha}\} &\leq_i \inf\{A_{2i\alpha}\}, \\ \sup\{A_{1i\alpha}\} &\leq_i \sup\{A_{2i\alpha}\} \end{aligned}$$

fennáll. Létezik továbbá ez alapján $\tilde{C}(X)$ -ben egy $\prec$ részben rendezés abban az értelemben, hogy $A_1 \prec A_2$ akkor és csak akkor, ha minden i -re

$$A_{1i} \prec_i A_{2i}.$$

A $\prec$ reláció bevezetése és tulajdonságai lehetővé teszik CNF halmazok egymással való összehasonlítását, továbbá az összehasonlítható CNF halmazok alsó és felső távolságainak definícióját.

8.3. Tétel. Legyen $A_1 \prec A_2$, $A_1, A_2 \in \tilde{C}(X)$, valamint X egydimenziós. Ekkor a két halmaz alsó és felső távolságát az alábbi különbségek határozzák meg:

$$d_{\alpha L}(A_1, A_2) = \inf\{A_{2\alpha}\} - \inf\{A_{1\alpha}\} \quad (8.5)$$

$$d_{\alpha U}(A_1, A_2) = \sup\{A_{2\alpha}\} - \sup\{A_{1\alpha}\} \quad (8.6)$$

Legyen most $X = \times_{i=1}^k X_i$ többdimenziós. Ekkor minden i -re létezik a fenti tulajdonságú $d_{i\alpha L/U}(A_{1i}, A_{2i})$, ha $A_{1i} \prec_i A_{2i}$. Normalizáljuk X_i -t úgy, hogy minden i -re $|X'_i| = 1$, ekkor $A_1 \prec A_2$ esetén $(A_1, A_2 \in \tilde{C}(X')) = \tilde{C}(\times_{i=1}^k X'_i)$ létezik a

$$d_{\alpha L/U}(A_1, A_2) = \left(\sum_{i=1}^k (d_{i\alpha L/U}(A_{1i}, A_{2i}))^u \right)^{1/u} \quad (8.7)$$

MINKOWSKI-távolság.

A fenti tételekben szereplő távolságdefiníció tulajdonságai alkalmasak a „közelség” számszerű leírására.

Az itt ismertetett definíciók alapján lehetőség nyílt ritka szabálybázison működő szabályinterpolációs eljárások kidolgozására. Az első fuzzy szabályinterpolációs eljárást KÓCZY és HIROTA javasolta [95, 96, 98]. A kiindulási ötlet a *kiterjesztési* és a *felbontási elven* alapszik. Az előbbi azt fejezi ki, hogy fuzzy halmaz formájában keresett megoldás megkapható úgy, hogy először megoldjuk a problémát tetszőleges α -vágatra, majd az így kapott megoldásokat fuzzy esetre kiterjesztjük. Az utóbbi a fuzzy halmazok α -vágatokra történő dekomponálhatóságát írja le:

$$A = \bigcup_{\alpha \in [0,1]} \alpha A_{\alpha}, \quad (8.8)$$

(ahol az unió maximumot jelent). Eszerint minden fuzzy halmaz leírható α -vágatai összességével. Elméletileg a végtelen számosságú α -vágatot külön kell kezelni, gyakorlatilag azonban, szakaszonként lineáris tagsági függvények esetén, elég csak néhány tipikus vágatot figyelembe venni.

Ha (az egydimenziós) X -ben és Y -ban létezik valamilyen $\prec$ -típusú rendezés, továbbá ha $R_i : A_i \rightarrow B_i$ ($i = 1, 2$) szabályok és A^* megfigyelés esetén

$$A_1 \prec_X A^* \prec_X A_2 \text{ és } B_1 \prec_Y B_2, \quad (8.9)$$

fennáll, akkor a $B^*(y)$ következtetés a lineáris interpoláció *alapegyenlete* alapján a következő módon számolható:

$$d(A^*, A_1) : d(A_2, A^*) = d(B^*, B_1) : d(B_2, B^*). \quad (8.10)$$

Távolságfogalomként például a (8.5)–(8.7) egyenleteket használva, az eljárás a következtetés α -vágatait oly módon határozza meg, hogy a következtetés és a konzekvensek távolságának aránya minden α -vágatra ($\alpha \in [0, 1]$) megegyezzek a megfigyelés és az antecedensek megfelelő arányaival:

$$d_{\alpha L/U}(A^*, A_1) : d_{\alpha L/U}(A_2, A^*) = d_{\alpha L/U}(B^*, B_1) : d_{\alpha L/U}(B_2, B^*). \quad (8.11)$$

Az *alapegyenlet* a lineáris interpoláció elvét terjeszti ki konvex és normális fuzzy halmazokon alapuló szabályokra, összhangban azzal a DUBOIS és PRADE által 1992-ben javasolt szemantikai szabályértelmezéssel [47, 49], mely szerint „minél hasonlóbb a megfigyelés valamely antecedenshez,

annál hasonlóbbnak kell lennie a következtetésnek az adott antecedenshez tartozó konzekvenshez”, és amelyet a ha–akkor szabályok „fokozatos” értelmezése alapján javasoltak. Ez egyben továbbfejleszti a TÜRKSÜN által javasolt analógiás következtetés elvét [185], mely szerint „minél közelebb van a megfigyelés egy antecedenshez, annál közelebb kell legyen a következtetés a megfelelő konzekvenshez” (a távolságot egyetlen globális crisp értékkel, az átlapolás mértékével mérve), valamint az ehhez közel álló DING, SHEN és MUKAIDONO által kifejlesztett „revíziós elvet” [43, 137, 159, 160, 161], mely a következtetést egy ún. szemantikai görbe segítségével szerkeszti meg (szintén átlapoló megfigyeléseknél).

Az interpolációs alapelv előnye az összes itt felsorolt módszerrel szemben, hogy akkor is működik, ha

$$\text{supp}(A^*) \cap \text{supp}(A_1) = \text{supp}(A^*) \cap \text{supp}(A_2) = \emptyset,$$

ahol A_1 és A_2 az A^* megfigyelést közrefogó antecedensek.

A (8.11) alapegyenlet alapján $\inf\{B^*\}$ és $\sup\{B^*\}$ egyértelműen számolható (ld. 8.4. ábra):

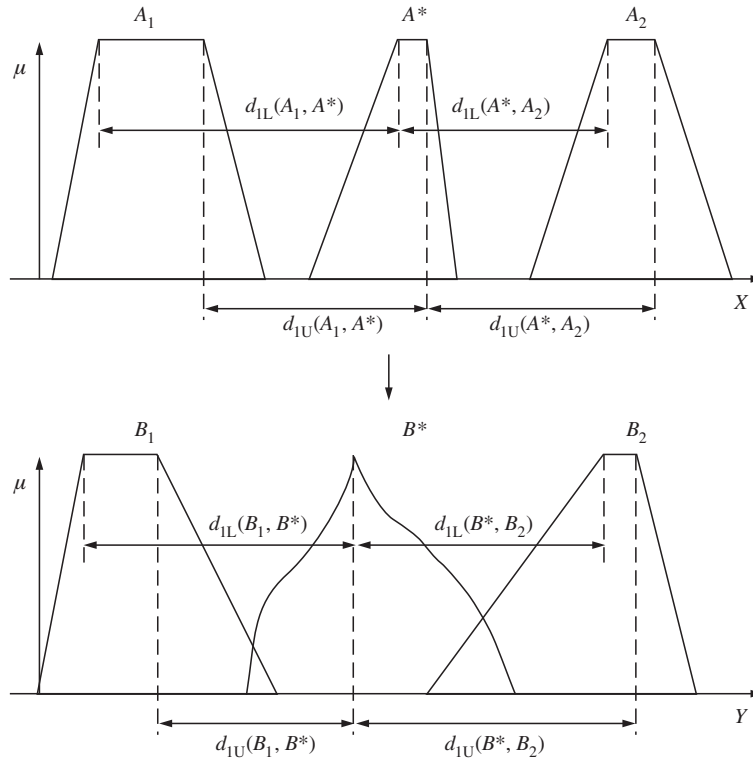
$$\begin{aligned} \inf\{B_\alpha^*\} &= \frac{\frac{\inf\{B_{1\alpha}\}}{d_L(A_\alpha^*, A_{1\alpha})} + \frac{\inf\{B_{2\alpha}\}}{d_L(A_\alpha^*, A_{2\alpha})}}{\frac{1}{d_L(A_\alpha^*, A_{1\alpha})} + \frac{1}{d_L(A_\alpha^*, A_{2\alpha})}}, \\ \sup\{B_\alpha^*\} &= \frac{\frac{\sup\{B_{1\alpha}\}}{d_U(A_\alpha^*, A_{1\alpha})} + \frac{\sup\{B_{2\alpha}\}}{d_U(A_\alpha^*, A_{2\alpha})}}{\frac{1}{d_U(A_\alpha^*, A_{1\alpha})} + \frac{1}{d_U(A_\alpha^*, A_{2\alpha})}}. \end{aligned} \quad (8.12)$$

Miután a feladatot a következtetés összes α -vágatára megoldottuk a felbontási elv (8.8) alapján a fuzzy következtetés megalkotható.

Ezek az egyenletek geometriailag éppen az $R_{1\alpha}$ és $R_{2\alpha}$ halmazok minimális, illetve maximális pontjai közötti, az A_α^* hengeres kiterjesztése szerinti lineáris interpolációt írnak le (vö. 8.5. ábra).

8.4.2. A lineáris interpolációs eljárás elemzése

Megvizsgálva a (8.12) által előállított eredményt, könnyen található olyan szituáció, hogy az eredményül kapott következtetés közvetlenül nem értelmezhető fuzzy halmazként, ugyanis a $d_{\alpha L/U}$ távolságok nem biztosítják, hogy növekvő α értékekre a $(\min\{B_\alpha^*\}, \max\{B_\alpha^*\})$ párok egymásba



8.4. ábra. Lineáris szabályinterpolációval számolt következtetés

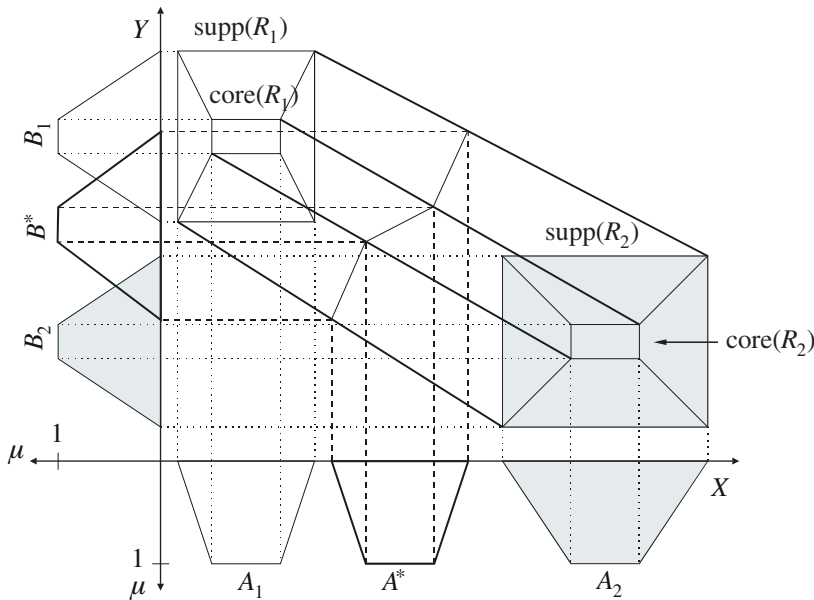
skatulyázott intervallumsorozatot alkossanak, s így előfordulhatnak a 8.6. és a 8.7. ábrán látható abnormális kimenetek is.

A probléma feloldására két eljárás ismert. Elsőként bizonyos rögzített tagságfüggvény-típusokra megadhatók olyan, a tagsági függvény alakjára vonatkozó feltételek, melyek biztosítják a konklúzió normalitását [104, 105, 106]. Ha a fenti feltétel nem teljesül, ezek az eljárások ún. „normalizációs” módszert javasolnak, mely a következtetés értelmezhetőségét garantálja. Az abnormális elkerülésének szükséges feltétele például trapéz alakú tagsági függvények esetén

$$b_2^* = \min\{B_1^*\} \leq \max\{B_1^*\} = b_3^*,$$

azaz

$$\frac{(a_{22} - a_2^*)b_{12} + (a_2^* - a_{12})b_{22}}{a_{22} - a_{12}} \leq \frac{(a_{23} - a_3^*)b_{13} + (a_3^* - a_{13})b_{23}}{a_{23} - a_{13}}, \quad (8.13)$$

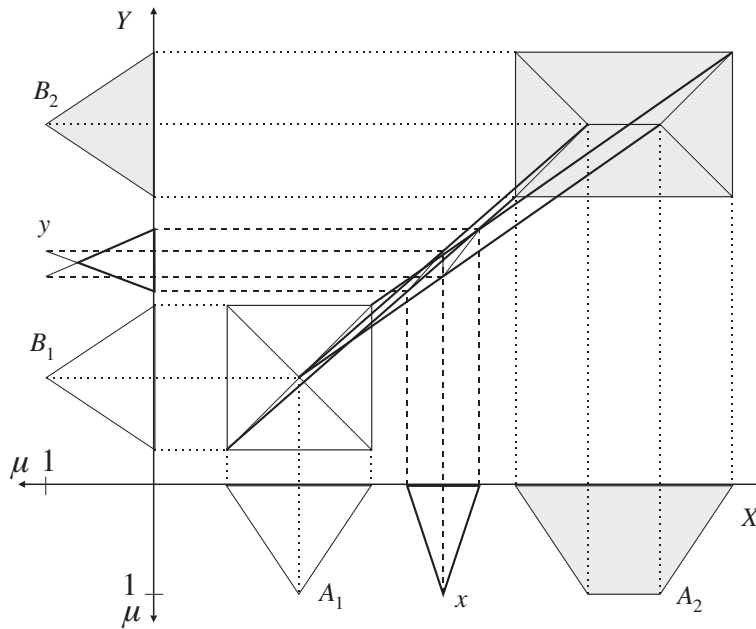


8.5. ábra. A lineáris szabályinterpoláció geometriai jelentése trapéz alakú tagsági függvények esetén

ahol az antecedenseket az a_{ij} ($i = 1, 2; j = 1, \dots, 4$) karakterisztikus pontok jellemzik. Hasonlóan, az a_j^* , b_{ij} és b_j^* karakterisztikus pontok rendre a megfigyelés, a konzekvensek és a következtetés megfelelő paraméterét jelölik. (A tagsági függvények karakterisztikus pontjaira a továbbiak során is ezt a jelölésmódot fogjuk használni.) Nem meglepő, hogy ebben a feltételben a tartóhoz kapcsolódó paraméterek nem játszanak közvetlen szerepet. További feltételeket találhatók SHI és MIZUMOTO munkáiban [163, 164, 165].

A másik abnormalitás-feloldó módszercsalád úgy módosítja a KH-eljárást, hogy az mindig normális konklúziót eredményezzen. Erről részletesebben a továbbfejlesztett interpolációs technikák tárgyalása során lesz szó (ld. 8.5. szakasz).

Mivel a szabályinterpolációs eljárások bevezetésének fő indoka a szabálysám és ezzel a számítási bonyolultság redukciója, ezért hatékony működésük feltétele, hogy a szabályok formája egyszerű, néhány karakterisztikus pont segítségével leírható, azaz szakaszosan lineáris, például lehetőleg háromszög vagy trapéz alakú legyen [101], ellenkező esetben ugyanis, amit nyerünk a réven, azt elvesztjük a vámon, azaz hiába csökkentjük a szabályok számát, ha túl bonyolulttá válik a megmaradó szabályok leírása.



8.6. ábra. Fuzzy halmazként közvetlenül nem értelmezhető konklúzióhoz vezető szituáció, ahol a konklúzió transzformálása után értelmes eredmény adódik

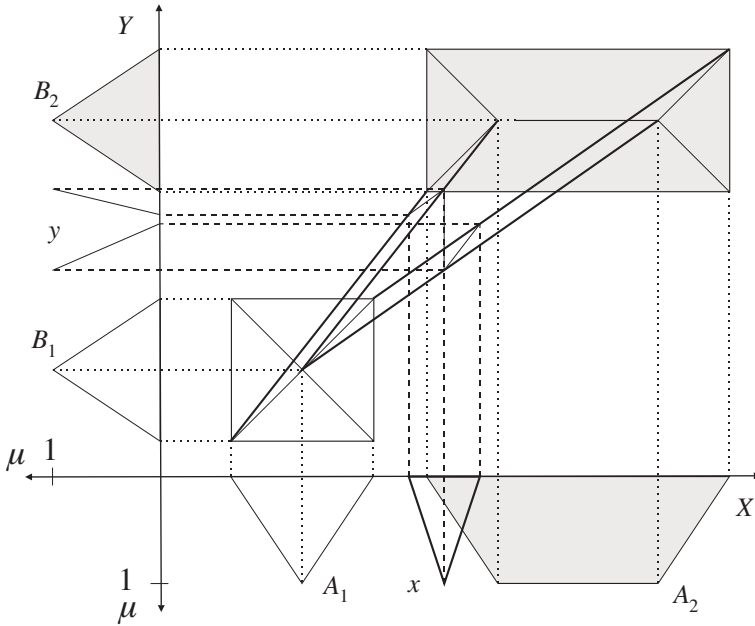
Azoknak az α értékeknek halmazát, melyekre a (8.12) egyenleteket mindenképpen ki kell számolni, *lényeges α -vágatoknak* nevezzük, és ezek halmazát Λ -val jelöljük (vö. 40. old). Λ minimális elemszáma kettő; ekkor csak a tartó és a mag tartozik a lényeges vágatok közé: $\Lambda = \{0,1\}$.

A tényleges redukció további feltétele, hogy szakaszosan lineáris tagsági függvények esetén elegendő legyen a karakterisztikus pontokra számolni a következtetést. Ez azt jelenti, hogy a szakaszonkénti linearitás megőrződik a konklúzióra is. Bár ez általánosságban csak bizonyos megszorító feltételek mellett teljesül (KÓCZY és KOVÁCS [104, 105, 106], KAWASE és CHEN [84]), de a vizsgálatok megmutatták, hogy a lineáristól való eltérés általános esetben is igen kicsiny, gyakorlati szempontból elhanyagolható, ezáltal teljesül a redukció feltétele.

Trapéz alakú szabályok esetében a lineáris oldalélekből kiszámított B^* oldaléle általában

$$Y_{\alpha L/U}^* = \frac{c_1 \alpha^2 + c_2 \alpha + c_3}{c_4 \alpha + c_5}$$

alakú, azaz távolról sem lineáris, sőt nem is polinomiális. B^* szakaszonként



8.7. ábra. Fuzzy halmazként közvetlenül nem értelmezhető konklúzióhoz vezető szituáció, ahol még transzformálással sem lehet értelmes eredményt elérni

polinomiális (kvadratikus) lesz, ha

$$\begin{aligned} a_{12} - a_{11} &= a_{22} - a_{21} = d, \\ a_{13} - a_{14} &= a_{23} - a_{24} = d', \end{aligned}$$

azaz az antecedensek oldalélei párhuzamosak, továbbá B^* szakaszonként lineáris lesz, ha $d^* = a_2^* - a_1^*$ és $d^{*'} = a_3^* - a_4^*$ jelölés mellett

$$(d - d^*)(b_{12} - b_{11} - b_{22} + b_{21}) = 0$$

és

$$(d' - d^{*'})(b_{13} - b_{14} - b_{23} + b_{24}) = 0.$$

Két tipikus, és a gyakorlat szempontjából fontos eset, ha

$$d = d^*, \quad d' = d^{*'},$$

azaz a megfigyelés és az antecedensek oldalélei párhuzamosak, illetve ha

$$b_{12} - b_{11} = b_{22} - b_{21} \text{ és } b_{13} - b_{14} = b_{23} - b_{24},$$

vagyis a konzekvensek megfelelő oldalélei párhuzamosak egymással.

Vizsgáljuk meg ezek után, hogy milyen mértékben képes a lineáris KH-eljárás csökkenteni a fuzzy következtetési algoritmusok eredendő exponenciális bonyolultságát. Tegyük fel, hogy $T' = T/S$ az egy dimenzióban szükséges ritka címkék számának felső korlátja ($S > 1$, lehetőség szerint $\gg 1$), és $\lambda = |\Lambda|$. Ekkor a lineáris interpolációs algoritmus bonyolultsága k dimenziós esetben

$$C_{K,id\delta} = \mathcal{O} \left(\lambda(k+1)T'^k \right), \quad (8.14)$$

de mivel λ a legtöbb gyakorlati esetben konstans, ezért

$$C_{K,id\delta} = \mathcal{O} \left((k+1)T'^k \right), \quad (8.15)$$

Azonban ha $k = 1$, akkor

$$C_{K,id\delta} = \mathcal{O} \left((k+1)T' + \lambda(k+1) \right),$$

ahol a λ -s tag fog dominálni. Szakaszonként lineáris B^* mellett jó közelítéssel $\lambda = 2$ és $C_{K,id\delta} = \mathcal{O}(k)$, míg gyenge linearitás mellett a nemlinearitás esetén mind λ , mind $C_{K,id\delta}$ értéke nő.

A lineáris interpolációs módszer kiterjeszthető kettőnél több szabályra is. Ha adott $2n$ szabály, úgy hogy $A_i \prec A^* \prec A_j$ és $B_i \prec B_j$, $i \in \{1, \dots, n\}$, $j \in \{n+1, \dots, 2n\}$, akkor az alábbi képletek alapján számítható B^* :

$$\begin{aligned} \inf B_{\alpha}^* &= \frac{\sum_{i=1}^{2n} \frac{1}{d_{\alpha L}(A_{\alpha}^*, A_{i\alpha})}}{\sum_{i=1}^{2n} \frac{1}{d_{\alpha L}(A_{\alpha}^*, A_{i\alpha})}} \inf \{B_{i\alpha}\}, \\ \sup B_{\alpha}^* &= \frac{\sum_{i=1}^{2n} \frac{1}{d_{\alpha U}(A_{\alpha}^*, A_{i\alpha})}}{\sum_{i=1}^{2n} \frac{1}{d_{\alpha U}(A_{\alpha}^*, A_{i\alpha})}} \sup \{B_{i\alpha}\}. \end{aligned} \quad (8.16)$$

Az egyes szabályok a megfigyeléstől való távolságukkal fordítottan arányos súllyal játszanak szerepet a következtetés megalkotásában. E módszer a környező $2n$ szabály alapján a szakaszonként lineáris interpolációnál finomabb közelítést ad, és nemlineáris interpolációt is lehetővé tesz.

A kiterjesztett KH-módszer érdekes tulajdonsága, hogy matematikai értelemben véve stabil, azaz ha a megfigyelés kismértékben változik, akkor az antecedens halmazok helyzetétől függetlenül a következtetés sem módosul jelentősen. Más megfogalmazással ez azt jelenti, hogy ha a bemeneteken megadott halmazok (antecedensek) száma minden határon túl nő, akkor a

kiterjesztett KH-eljárás (8.16) interpolációs operátora, $K_n(f, x)$, a bemene-
tek egyenletes eloszlása esetén (tetszőleges részhalmazra eső antecedensek
aránya megegyezik e részhalmaz és az alaphalmaz LESBEGUE-mértékének
arányával) egyenletesen konvergál a következtetési eljárással közelített
folytonos függvényhez, ahol

$$K_n(f, x) = \sum_{k=1}^n f(x_k^{(n)}) \frac{\frac{1}{\|x - x_k^{(n)}\|_p^N}}{\sum_{j=1}^n \frac{1}{\|x - x_j^{(n)}\|_p^N}} \quad (8.17)$$

az interpolációs operátor, f a közelített N dimenziós folytonos függvény, p
a távolság meghatározásához alkalmazott L_p norma paramétere, és $x_k^{(n)}$ a
 k -adik antecedens megfelelő értéke. Ekkor $\lim_{n \rightarrow \infty} K_n(f, x) = f(x)$ [180].

Az állításban szereplő $K_n(f, x)$ operátor szoros kapcsolatban áll a függ-
vényapproximáció területén behatóan vizsgált BALÁZS–SHEPARD inter-
polációs operátorral (ld. pl. [40, 41, 42, 162, 176, 177]). Belátható (TIKK
[180]), hogy a kiterjesztett KH-eljárás interpolációs függvénye a BALÁZS–
SHEPARD interpolációs operátor fuzzy általánosítása.

8.5. Interpolációs módszerek áttekintése

A KH-módszer legjelentősebb hátránya az, hogy nem ad mindig köz-
vetlenül fuzzy halmazként értelmezhető eredményt. További megkötés,
hogy csak CNF halmazokra alkalmazható. Ezen problémák több kutatót a
módszer javítására, illetve más, lényegesen különböző módszerek kidolgo-
zására ösztönöztek.

8.5.1. VKK-eljárás

Elsőként VASS, KALMÁR és KÓCZY javasolt olyan módszert [188] (VKK-
interpoláció), mely csökkenti ugyan az alkalmazhatóság korlátait, de nem
szünteti meg azokat teljesen. A VKK-eljárás az alsó és felső távolságok
alternatívájaként bevezetett középső távolság és szélességi viszony segítsé-
gével definiálja a következtetést. Két szabályra az alapegyenlet megoldása

a következő:

$$\begin{aligned} \text{centr}\{B_{\alpha}^*\} &= \frac{\frac{1}{d_{\alpha}(A^*, A_1)} \text{centr}\{B_{1\alpha}\} + \frac{1}{d_{\alpha}(A^*, A_2)} \text{centr}\{B_{2\alpha}\}}{\frac{1}{d_{\alpha}(A^*, A_1)} + \frac{1}{d_{\alpha}(A^*, A_2)}}, \\ \text{width}\{B_{\alpha}^*\} &= \frac{\frac{\text{width}\{A_{\alpha}^*\}}{d_{\alpha}(A^*, A_1)} \cdot \frac{\text{width}\{B_{1\alpha}\}}{\text{width}\{A_{1\alpha}\}} + \frac{\text{width}\{A_{\alpha}^*\}}{d_{\alpha}(A^*, A_2)} \cdot \frac{\text{width}\{B_{2\alpha}\}}{\text{width}\{A_{2\alpha}\}}}{\frac{1}{d_{\alpha}(A^*, A_1)} + \frac{1}{d_{\alpha}(A^*, A_2)}} \end{aligned} \quad (8.18)$$

(d_{α} a középső α -távolságot jelenti).

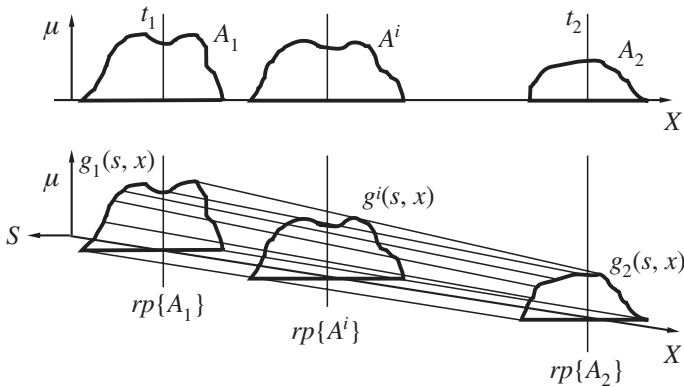
A (8.18) képlet szintén kiterjeszthető oly módon, hogy a konklúzió megalkotásánál több szabálypárt veszünk figyelembe. Mivel a következtetés megalkotásában résztvevő szabályok számának növelésével $x \rightarrow \pm\infty$ esetén a kifejezés lineáris függvényhez tart, ez lehetővé tesz bizonyos extrapolációs alkalmazásokat is [188].

8.5.2. Szabályinterpoláció testmetszéssel

Alapjaiban különböző eljárásokat javasoltak BARANYI és munkatársai, melyek a fuzzy halmazok közötti reláció [6, 7], illetve a szemantikus görbe és interreláció interpolációján alapulnak [10]. A módszerek egydimenziós változata a következő lépésekből áll. Az eljárás először meghatározza a kiszámítandó következtetés helyét (annak a referencia vagy legjellemzőbb pontját) a megfigyelés és az antecedensek referenciapontjai arányának segítségével. Ezután kerül sor a konklúzió megalkotásában résztvevő összes halmaz referenciapontjuk körüli 90°-kal történő elforgatására, majd az elforgatott halmazok megfelelő pontjainak összekötésére, melynek segítségével két test keletkezik, egyik a bemeneti, másik pedig a kimeneti halmazok terében (8.8. ábra).

A testeknek a megfigyelés és a következtetés referenciapontjánál történő elmetszésével két halmazt kapunk (8.8. ábra): $A^{*'}-t$ a bemeneti, és $B^{*'}-t$ a kimeneti alaphalmazon. Végül a végső következtetés (B^*) meghatározása az A^* és az „interpolált” megfigyelés $A^{*'}-t$ hasonlóságát felhasználó ún. *revíziós függvény* segítségével történik.

A testmetszéses módszer több előnyös tulajdonsággal rendelkezik. Tet- szőleges bemeneti halmazrendszer esetén mindig közvetlenül értelmezhető fuzzy halmazt ad következtetésként. Nincsen korlátozva a bemeneti halmazok alakja, vagyis sem a normalitás, sem a konvexitás nem szükséges; az alkalmazás kizárólagos feltétele az, hogy a halmazok referenciapontja (8.9)



8.8. ábra. A testmetszéses módszer alap gondolata

szerint rendezett legyen. Ez a feltétel még azt is megengedi, hogy a megfigyelés egy része valamelyik antecedens tartóján túlnyúlják. A módszer szakaszosan lineáris halmazokra kifejlesztett változata minden esetben pontosan (tehát nem közelítőleg) szakaszonként lineáris következtetést eredményez.

A módszer egyetlen komolyabb hátránya az, hogy a revíziós függvény számítása még szakaszosan lineáris halmazok esetén is jelentős időt igényel, viszont ezzel a fuzzy szabályinterpolációs módszerek bevezetésének legfőbb indoka sérül. A testmetszéses módszer részletes leírása és további hivatkozások [5]-ben található.

8.5.3. További szabályinterpolációs módszerek

A relatív fuzziság megőrzésén alapuló, CNF halmazokon működő eljárást javasolt GEDEON és KÓCZY 1996-ban [64], amely azonban bizonyos crisp halmazokra nem alkalmazható. E módszer javítása található [102]-ben, mely kiküszöböli ezt a hibát. A szerzők rámutattak ezen módszer és a fuzzy szabályinterpoláció alapegyenletének (8.10) közvetlen kapcsolatára is.

A fuzzy szabályok bizonytalan környezetben való közelítésén alapszik a KOVÁCS és KÓCZY által javasolt eljárás [115, 116, 117].

8.5.4. A MACI-módszer

A felsoroltak alapján megállapítható, hogy bár sok olyan interpolációs módszer készült a KH-interpoláció nyomán, melyek az eredeti algoritmus hátrányait csökkentették vagy kiküszöbölték, ám ezek számítási bonyolultsága kisebb-nagyobb mértékben meghaladja az eredeti algoritmusét. A cél ezért egy olyan módszer kidolgozása volt, mely megtartja a KH-módszer előnyeit — s ezek közül is leginkább annak alacsony időigényét —, ugyanakkor megszünteti az abnormális következtetés lehetőségét. A következőkben ismertett módosított α -vágat alapú módszer, amelyet angol elnevezéséből (Modified Alpha-Cut based Interpolation) alkotott betűszó alapján MACI-módszernek hívunk, megfelel ezen kritériumoknak.

Először a módszerben felhasznált vektorrepresentációs eljárást ismer-tetjük [202]. Legyen A háromszög alakú fuzzy halmaz. Ekkor A az $\underline{a} = \langle a_{-1}, a_0, a_1 \rangle^T$ vektorral egyértelműen megadható, ahol a_{-1} és a_1 jelöli A tartójának két végpontját, és a_0 az A halmaz egyelemű magját (csúcsát). Ezeket a paramétereket az A fuzzy halmaz *karakterisztikus pontjainak* nevez-zük. Minden háromszög alakú A fuzzy halmazhoz hozzárendelhetünk tehát egy $\underline{a}$ vektort, amelynek karakterisztikus pontjaira fennáll az

$$a_{-1} \leq a_0 \leq a_1 \quad (8.19)$$

egyenlőtlenség. Fordítva, minden olyan $\underline{a} = \langle a_{-1}, a_0, a_1 \rangle^T$ vektor, melyre (8.19) teljesül, egyértelműen meghatároz egy háromszög alakú A fuzzy halmazt.

Az $\underline{a}$ vektort két újabb vektorra bonthatjuk:

$$\underline{a}_L = \langle a_{-1}, a_0 \rangle^T, \text{ és } \underline{a}_U = \langle a_0, a_1 \rangle^T, \quad (8.20)$$

melyek rendre a bal, illetve jobb oldalél karakterisztikus pontjait tartalmaz-zák. Az egyszerűség kedvéért mostantól csak a jobb (azaz felső) oldalélel foglalkozunk, a bal oldalélre a megfelelő állítások analóg módon beláthatók (ld. pl. [180]). Ha másképpen kifejezetten nem állítjuk, akkor egy fuzzy halmazt reprezentáló vektoron ezután a jobb oldalélet reprezentáló vektort értjük.

Hasonlóképpen, minden konvex (és nem feltétlenül normális), szakaszonként lineáris fuzzy halmazhoz egyértelműen hozzárendelhetünk egy $n + 1$ elemű vektort, amelynek elemei a halmaz $n + 1$ karakterisztikus pontját tartalmazzák:

$$\underline{a} = \langle a_0, a_1, \dots, a_n \rangle^T \quad (8.21)$$

(a jobb oldaléltre). A vektor elemei monoton nőnek (vö. (8.19)-el). Kisebb módosítással a reprezentációs módszert folytonos fuzzy halmazokra is kiterjeszthetjük [180].

Tegyük fel, hogy adottak a $A_1 \rightarrow B_1$ és $A_2 \rightarrow B_2$ szabályok, valamint az A^* megfigyelés, amelyre (8.9) fennáll. Vektorreprezentációs megközelítésben a KH-módszert az alábbi módon írhatjuk le:

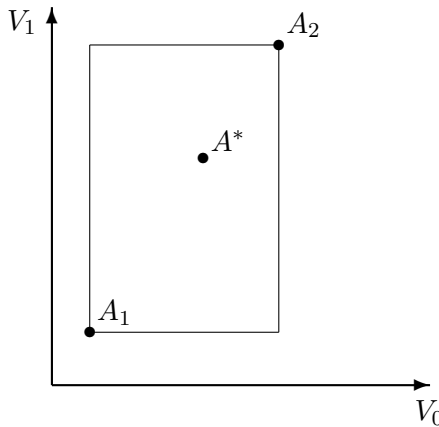
$$\underline{b}^* = (\underline{I} - \underline{I}\underline{\Lambda})\underline{b}_1 + \underline{I}\underline{\Lambda}\underline{b}_2, \quad (8.22)$$

ahol $\underline{I}$ az identitásmátrix és

$$\underline{\Lambda} = \langle \lambda_0, \lambda_1 \rangle, \quad \lambda_k = \frac{a_k^* - a_{1k}}{a_{2k} - a_{1k}}, \quad k = 0, 1. \quad (8.23)$$

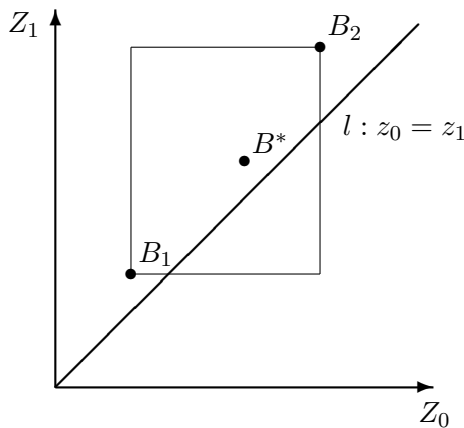
Itt a_k^* , a_{1k} , és a_{2k} rendre a megfigyelés és a két antecedens k -adik karakterisztikus pontja, vagyis egyúttal a megfelelő vektor k -adik eleme.

Ábrázoljuk ekkor a fenti reprezentációval a $V_0 \times V_1$ kétdimenziós térben az antecedenseket és a megfigyelést, a $Z_0 \times Z_1$ térben pedig a konzekvenseket. Hogyan jellemezhetők ezekben a terekben lineáris interpoláció esetén a szóban forgó halmazok?



8.9. ábra. Az antecedensek és a megfigyelés vektorreprezentációja (jobb oldalél)

Mivel az antecedens halmazok és a megfigyelés rendezett (8.9) szerint, ezért a hozzájuk rendelt vektorban a második koordináta sosem kisebb az elsőnél, azaz az oldaléleket reprezentáló pontok az $x = y$ egyenes és a V_1 tengelyek közé esnek (a tengelyeket is beleértve; 8.9. ábra). Hasonló igaz a konzekvens halmazokra is (8.10. ábra).



8.10. ábra. A konzekvensek és a következtetés vektorrepresentációja (jobb oldalél)

Ahhoz, hogy a következtetés fuzzy halmaz legyen, a $\underline{b}^*$ vektornak az l egyenes és a Z_1 tengely közé kell esnie (ezekre való illeszkedés is megengedett; 8.10. ábra):

$$b_0^* \leq b_2^* \quad (8.24)$$

A KH-módszer feltételei miatt $a_{1k} < a_k^* < a_{2k}$, így a λ_k törtek ($k = 0, 1$) értéke nemnegatív, és a $[0, 1]$ intervallumba esik. Ez azonban csak azt garantálja, hogy a következtetés — az antecedensek és a megfigyelés értékétől függően — az ábrán látható téglalapon belül lesz, ahol nem lehet kizárni az abnormalis következtetés lehetőségét, ha a téglalap metszi az l egyenest. Az egész téglalap viszont csak abban az esetben esik az l egyenes fölé, ha B_1 és B_2 konzekvensek nem diszjunktak.

A megoldást a következő ötlet adja: transzformáljuk a B_1, B_2 konzekvenseket egy másik koordináta-rendszerbe, amely kizárja az abnormalitás lehetőségét. A következtetés meghatározásához helyettesítsük átmenetileg a Z_0 tengelyt az $l : z_0 = z_1$ egyenessel, míg a Z_1 maradjon változatlan. Vegyük észre, hogy B_1 és B_2 konvexitása biztosítja az új rendszerbeli nemnegatív koordinátákat. Ezután számítsuk ki a következtetés helyét az új koordináta-rendszerben, végül transzformáljuk vissza az így kapott eredményt az eredeti rendszerbe. Ez a konstrukció biztosítja, hogy a következtetés koordinátái monoton növekedjenek, azaz a (8.24) egyenlőtlenség teljesüljön.

Tetszőleges $\underline{b}$ vektor esetén a transzformáció az alábbi:

$$\begin{aligned}\underline{b} = \langle b_0, b_1 \rangle &\rightarrow \underline{b}' = \langle b'_0, b'_1 \rangle \\ b'_0 &= b_0 \cdot \sqrt{2} \\ b'_1 &= b_1 - b_0.\end{aligned}\tag{8.25}$$

Mátrixos írásmóddal

$$\underline{b}' = \underline{b} \underline{T},$$

ahol

$$\underline{T} = \begin{bmatrix} \sqrt{2} & 0 \\ -1 & 1 \end{bmatrix}.\tag{8.26}$$

Ha a konzekvenseket már eszerint transzformáltuk, akkor a konklúziót a (8.23) összefüggés szerinti λ_k ($k = 0, 1$) értékek felhasználásával a

$$b^{*'}_0 = (1 - \lambda_0)b'_{10} + \lambda_0 b'_{20},\tag{8.27}$$

$$b^{*'}_1 = (1 - \lambda_1)b'_{11} + \lambda_1 b'_{21}\tag{8.28}$$

egyenletekkel kapjuk. Mátrix alakban

$$\underline{b}^{*'} = (\underline{I} - \underline{I}\underline{\Lambda})\underline{b}'_1 + \underline{I}\underline{\Lambda}\underline{b}'_2.\tag{8.29}$$

Mivel (8.9) teljesül és a λ_k együtthatók nem változnak a (8.27) és a (8.28) kifejezésekben, ezért az új koordináták — mint nemnegatív számok konvex kombinációja — nemnegatívok lesznek.

A konklúzió visszatranszformálását a

$$\begin{aligned}b^*_0 &= b^{*'}_0 / \sqrt{2}; \\ b^*_1 &= b^{*'}_1 + b^*_0 = b^{*'}_1 + (b^{*'}_0 / \sqrt{2})\end{aligned}$$

egyenletek alapján végezzük, másképpen

$$\underline{b}^* = \underline{b}^{*'} \underline{T}^{-1},\tag{8.30}$$

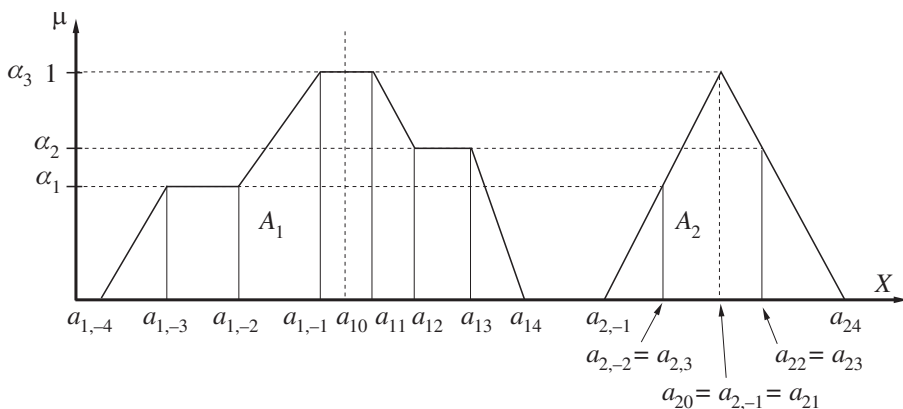
ahol

$$\underline{T}^{-1} = \begin{bmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1 \end{bmatrix}.$$

Megjegyezzük, hogy a középpont (b^*_0) értéke nem változik az eljárás során és ugyanez teljesül a bal oldaléle is [11, 180], ezért a két oldalél felső végpontja összeér, és a végső konklúzió háromszög alakú lesz.

A transzformációs eljárás bonyolultabb alakú tagsági függvények esetén is működik. $n + 1$ karakterisztikus pont esetén egy fuzzy halmaz (8.21)

formában reprezentálható. Abban az esetben, ha a konklúzió meghatározásában szerepet kapó halmazok karakterisztikus pontjainak száma nem azonos, vagyis például egyaránt van köztük háromszög és trapéz alakú is, akkor általánosan a következőképpen kell eljárni. Ha van olyan $\alpha_1 \in [0,1]$ érték, amely csak bizonyos halmazok esetén tartozik a fontos vágatok közé, akkor az ilyen α_1 értékhez a többi halmaz esetén is karakterisztikus pontot kell rendelni, méghozzá olyan multiplicitással, amekkorával az α_1 valódi töréspontként maximálisan szerepel (ld. 8.11. ábra). A halmazok konvexitása biztosítja, hogy olyan halmazokra, ahol α_1 nem töréspont, oldalélként csak egy $x \in X$ elem tartozik; ezt az x -et választjuk az α_1 -hez tartozó karakterisztikus pontnak.



8.11. ábra. Különböző töréspontok esetén a karakterisztikus pontok meghatározása

A következtetésre a

$$b_i^* \leq b_j^*, \quad \forall i \leq j \in [0, n]$$

egyenlőtlenségnek kell teljesülnie. Az eredeti $Z_0, \dots, Z_n$ koordinátatengelyeket a

$$Z'_i = \{(z_i, \dots, z_n) | z_\ell = z_m, \ell, m \in [i, n]\}, \quad i \in [0, n]$$

tengelyekkel helyettesítjük. Vegyük észre, hogy a Z_n tengely nem változik, ezért a transzformáció során a b_0^* értéke sem változik.

A $\underline{\Lambda}$ együtthatóvektor a

$$\underline{\Lambda} = \langle \lambda_0, \dots, \lambda_n \rangle, \quad \lambda_k = \frac{a_k^* - a_{1k}}{a_{2k} - a_{1k}} \quad (k = 0, \dots, n) \quad (8.31)$$

vektorra bővül. A transzformációs mátrix értéke

$$\underline{T} = \begin{bmatrix} \sqrt{n+1} & 0 & 0 & 0 & \dots & 0 \\ -\sqrt{n} & \sqrt{n} & 0 & 0 & \dots & 0 \\ 0 & -\sqrt{n-1} & \sqrt{n-1} & 0 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ \vdots & & & & \ddots & \vdots \\ 0 & \dots & 0 & -\sqrt{2} & \sqrt{2} & 0 \\ 0 & \dots & \dots & 0 & -1 & 1 \end{bmatrix},$$

melynek inverze a

$$\underline{T}^{-1} = \begin{bmatrix} 1/\sqrt{n+1} & 0 & 0 & \dots & 0 \\ 1/\sqrt{n+1} & 1/\sqrt{n} & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 1/\sqrt{n+1} & 1/\sqrt{n} & \dots & 1/\sqrt{2} & 1 \end{bmatrix}$$

mátrix lesz. Tehát a transzformált következtetés koordinátái

$$\begin{aligned} b'_0 &= b_0\sqrt{n+1}, \\ b'_1 &= b_1\sqrt{n} - (b'_0/\sqrt{n+1})\sqrt{n}, \\ b'_2 &= b_2\sqrt{n-1} - (b'_1/\sqrt{n} + b'_0/\sqrt{n+1})\sqrt{n-1}, \\ &\vdots \\ b'_k &= b_k\sqrt{n-k+1} - \left(\sum_{i=0}^{k-1} b'_i/\sqrt{n-i+1} \right) \sqrt{n-k+1}, \\ &\vdots \\ b'_n &= b_n - \left(\sum_{i=0}^{n-1} b'_i/\sqrt{n-i+1} \right), \end{aligned}$$

lesznek, míg a végső konklúzió koordinátáit a

$$\begin{aligned} b^*_0 &= b'^*_0/\sqrt{n+1} \\ b^*_1 &= b'^*_1/\sqrt{n} + b'^*_0/\sqrt{n+1} \\ b^*_2 &= b'^*_2/\sqrt{n-1} + b'^*_1/\sqrt{n} + b'^*_0/\sqrt{n+1} \\ &\vdots \end{aligned} \tag{8.32}$$

$$\begin{aligned}
 b_k^* &= \sum_{i=0}^k (b_i^* / \sqrt{n-i+1}) \\
 &\vdots \\
 b_n^* &= \sum_{i=0}^n (b_i^* / \sqrt{n-i+1})
 \end{aligned}$$

egyenletrendszer adja meg. Ennek alapján belátható [11, 180], hogy

8.4. Tétel. *A koordinátatranszformációs-módszer CNF bemenetek esetén mindig CNF halmazt ad, azaz zárt a CNF halmazok körében.*

8.5.5. A MACI-módszer vizsgálata

A módszer vizsgálata során elsőként hasonlítsuk össze az új és a KH-eljárás által számolt következtetést. A következtetések koordinátánkénti kapcsolatára az alábbi összefüggés áll fenn (a jobb oldaléle vonatkozóan) [11, 180]:

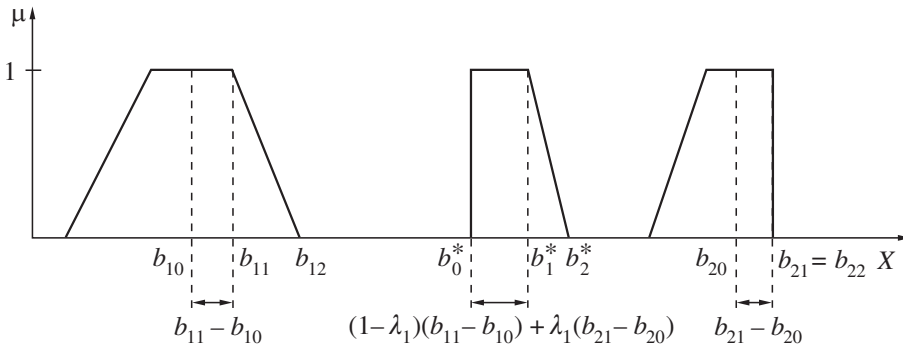
$$b_k^* = {}^{KH}b_k^* + \sum_{i=0}^{k-1} (\lambda_i - \lambda_{i+1})(b_{2i} - b_{1i}), \quad k \in [0, n], \quad (8.33)$$

ahol ${}^{KH}b_k^*$ a KH-interpoláció által számolt k -edik karakterisztikus pont értéke (vö. (8.22)-vel). A másik oldaléle hasonló eredmény kapható.

A (8.33) összefüggés alapján a módszer által adott következtetést a KH-eljárás eredményéhez képest a következőképpen lehet az előző alpontban bemutatott említett transzformáció nélkül jellemezni. Mint már korábban említettük, a referenciapont koordinátája, azaz mindkét oldalél esetén az első, közös b_0^* koordináta nem változik. A következő koordinátákat — mindkét irányban — a konzekvensek és az adott iránybeli első karakterisztikus pontok közötti távolság ($b_{i1} - b_{i0}$; $i = 1, 2$) határozza meg, ami a (8.33)-ból egyszerűen kiszámolható

$$b_k^* = b_{k-1}^* + ((1 - \lambda_k)(b_{1k} - b_{1,k-1}) + \lambda_k(b_{2k} - b_{2,k-1}))$$

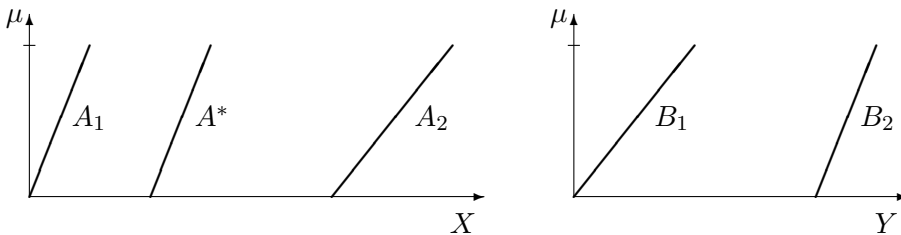
összefüggés alapján világosan látszik (ld. még [11, 180]). A további koordinátákat is rendre a konzekvensek aktuális és a megelőző karakterisztikus pontjai közötti különbség határozza meg. Ez azt jelenti, hogy ebben az esetben az egymás utáni karakterisztikus pontok között explicit összefüggés van, ami az eredeti KH-eljárás esetén hiányzik.



8.12. ábra. A következtetés koordinátái között fennálló összefüggés geometriai interpretációja (jobb oldalélre)

A módszer által nyújtott következtetést geometriailag tehát a 8.12. ábrán látható módon lehet interpretálni.

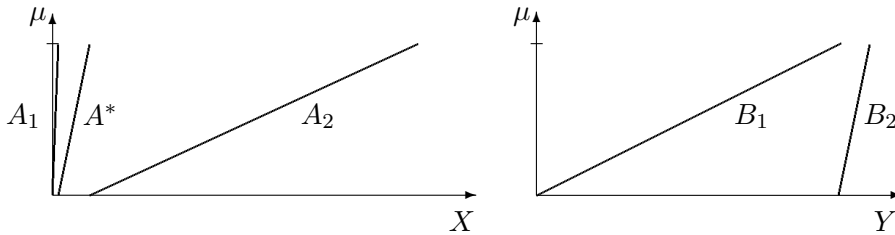
A (8.33) képlet alapján nyilvánvaló, hogy a két eljárás akkor és csak akkor ad azonos következtetést, ha az összes λ_k arány megegyezik minden k -ra. Egyszerű becslések segítségével belátható, hogy a két eljárás által adott következtetés különbsége korlátos, gyakorlati esetekben nem számottevő [11, 180].



8.13. ábra. Példa a linearitás közelítő megtartására. A háromszög alakú tagsági függvények bal oldalélét ábrázoltuk: $A_1 : (0,1)$, $A_2 : (5,7)$, $A^* : (2,3)$, $B_1 : (0,2)$, $B_2 : (4,5)$

Megmutatjuk, hogy a MACI-módszer számítási bonyolultsága nagyságrendileg megegyezik a KH-módszerével. Az nyilvánvaló, hogy az egyes koordináták kiszámítása azonos időt vesz igénybe, hiszen a két módszer hasonló eljárással adja meg a következtetést. Kérdés, hogy elegendő-e a következtetést a MACI-módszernél is csak a karakterisztikus pontokra számolni, azaz megtartja-e — legalábbis közelítőleg — a konklúzió a szakaszos linearitást a karakterisztikus pontok közti intervallumokra.

A fenti kérdésre a válasz egyértelműen igen. A [182]-ben elvégzett össze-



8.14. ábra. Szélsőségesebb példa esetén is jó a lineáris közelítés. A bal oldalélek:

$A_1 : (0,1)$, $A_2 : (10,100)$, $A^* : (1,10)$, $B_1 : (0,10)$, $B_2 : (10,11)$

hasonlítás egyrészt megmutatta, hogy [104]-ből vett példák és háromszög alakú tagsági függvények esetén a lineáristól való gyakorlati eltérés még a KH-módszernél tapasztaltnál is kisebb (a példákat és a KH-módszerrel összehasonlított eredményeket a 8.13. és 8.14. ábrák, valamint a 8.3. és 8.4. táblázatok mutatják). Másrészt, a kísérleti tényeket matematikailag becslésekkel is alátámasztva belátható, hogy a lineáristól való eltérés igen kicsiny.

A MACI-módszer további érdekessége, hogy a KH-módszer matematikai stabilitását is megőrzi [183]. Ezt a tulajdonságot — hasonlóan, mint a KH-módszer esetében —, az eljárás kiterjesztett változatára lehet belátni, amikor nem csupán két, hanem oldalanként n környező szabályt veszünk

8.3. táblázat. A következtetés számított és becsült értékei $\alpha = 0,1$ -es felosztás esetén a 8.13. ábra halmazaira

α	KH-módszer		MACI-módszer	
	számolt	becsült	számolt	becsült
0	1,600	1,60	1,400	1,40
0,1	1,729	1,74	1,553	1,56
0,2	1,862	1,88	1,708	1,72
0,3	1,996	2,02	1,864	1,88
0,4	2,133	2,16	2,022	2,04
0,5	2,273	2,30	2,182	2,20
0,6	2,414	2,44	2,343	2,36
0,7	2,558	2,58	2,505	2,52
0,8	2,703	2,72	2,669	2,68
0,9	2,851	2,86	2,834	2,84
1,0	3,000	3,00	3,000	3,00

8.4. táblázat. A következtetés számított és becsült értékei $\alpha = 0,1$ -es felosztás esetén a 8.14. ábra halmazaira

α	KH-módszer		MACI-módszer	
	számolt	becsült	számolt	becsült
0	1,000	1,000	1,000	1,000
0,1	1,867	1,91	1,914	1,918
0,2	2,767	2,82	2,832	2,836
0,3	3,676	3,73	3,750	3,755
0,4	4,589	4,64	4,668	4,673
0,5	5,505	5,55	5,587	5,591
0,6	6,421	6,45	6,506	6,509
0,7	7,338	7,36	7,425	7,427
0,8	8,255	8,27	8,344	8,345
0,9	9,173	9,18	9,263	9,264
1,0	10,091	10,091	10,182	10,182

figyelembe a konklúzió előállításához.

Ezek alapján megállapítható, hogy a MACI-módszer valóban megőrzi a KH-eljárás előnyös tulajdonságait (alacsony számítási igény, matematikai stabilitás), és emellett az abnormalitás lehetőségét is kiküszöböli.

8.6. Hierarchikus szabálybázisok

Habár a ritka szabálybázisok és fuzzy interpolációs algoritmusok használata kevés állapotváltozó esetén számottevően csökkentheti a futási időt, 3–5-nél több állapotváltozó esetén nem biztosít jelentős időigénynyereséget. Áttörést igazából csak az jelenthet, ha valamilyen módon az időigény $O(T^k)$ nagyságrendű kifejezésében a k kitevőt (is) csökkenteni tudjuk. Erre esetenként megoldást jelenthet a felesleges változók kiküszöbölése [27], de ez általánosan nem alkalmazható. Nagy változósám esetén csak radikálisabb módszerrel csökkenthető a szabályok száma, illetve a számítási bonyolultság: a szabálybázis szerkezetének megváltoztatásával, hierarchikus szabálybázisrendszer kialakításával. Az első példa hierarchikus szabályrendszer megalkotására és alkalmazására a SUGENO által publikált vezető nélküli helikopter vezérlése [172, 174].

A strukturált szabálybázis alapötlete a következő. Egy bonyolult rend-

szer leírásához ugyan többnyire sok, nélkülözhetetlen állapotváltozó szükséges, azonban előfordulhat az, hogy *lokálisan* a változók egy valódi rész-halmaza is elégséges a modell kellő pontosságú leírására. Természetesen ez a részhalmaz az állapottér különböző régióiban más és más lehet. Ha ilyen lokális változórészhalmazok ismertek, a teljes állapotteret particionáljuk, és minden egyes résztérhez lokális modellt készítünk. Szerencsés esetben a lokális rendszerek lényegesen kevesebb változót használnak, és így az összesített szabálysám is jelentősen csökkenhet. A helikopteres példánál maradva, más változók dominálnak az „emelkedés”, és megint mások például az „előre repülés” művelete közben.

Minden lokális modellhez egy *alszabálybázis* tartozik. A felső, ún. *metaszinten* először — a megfigyelés környezete vagy a rendszer előírt reakciója alapján — a megfelelő alszabálybázis kiválasztására kerül sor. Ezt a lépést az ún. *metaszabályok* határozzák meg, amelyek bizonyos, a lokális modelleket lényegében elkülönítő változók értéke, vagy speciálisan a rendszer lokális működését szabályozó változók értéke alapján választják ki a megfelelő lokális modellt.

Lehetőség van több metaszint alkalmazására is, ilyenkor az egyes metaszintek a modell egyre pontosabb finomítását végzik, és a kiválasztott állapotváltozókhoz tartozó lokális szabálybázis szerinti következtetés meghatározása a legalsó szinten történik.

Az egy metaszintet tartalmazó strukturált szabálybázis az alábbiak szerint adható meg formálisan:

Metaszint (R_0):

Ha $z_0 = D_1$ akkor vegyük a D_1 tartományt és a hozzátartozó R_1 bázist

Ha $z_0 = D_2$ akkor vegyük a D_2 tartományt és a hozzátartozó R_2 bázist

⋮

Ha $z_0 = D_n$ akkor vegyük a D_n tartományt és a hozzátartozó R_n bázist

R_1 szabálybázis D_1 tartományhoz:

Ha $z_1 = A_{11}$ akkor $Y = B_{11}$

Ha $z_1 = A_{12}$ akkor $Y = B_{12}$

⋮

Ha $z_1 = A_{1m_1}$ akkor $Y = B_{1m_1}$

R_2 szabálybázis D_2 tartományhoz:

Ha $z_2 = A_{21}$ akkor $Y = B_{21}$

Ha $z_2 = A_{22}$ akkor $Y = B_{22}$

⋮

Ha $z_2 = A_{2m_2}$ akkor $Y = B_{2m_2}$

stb., R_n szabálybázis D_n tartományhoz:

Ha $z_n = A_{n1}$ akkor $Y = B_{n1}$

Ha $z_n = A_{n2}$ akkor $Y = B_{n2}$

⋮

Ha $z_n = A_{nm_n}$ akkor $Y = B_{nm_n}$

ahol R_i ($i \in [1, n]$) a D_i tartományhoz tartozó lokális szabálybázis; $Z_0, Z_1, \dots, Z_n$ az $X = X_1 \times X_2 \times \dots \times X_n$ állapotter részterei és az R_i szabálybázisok bemeneti alaphalmazai, $z_i \in Z_i$ ($i \in [0, n]$), valamint D_i a $\Pi = \{D_1, D_2, \dots, D_n\}$ partíció i -edik eleme. Z_i -t *valódi részhalmaznak* nevezzük, ha $Z_i \subset X_i$. A $\Pi = \{D_1, D_2, \dots, D_n\}$ partíció *teljes*, amennyiben az egész X alaphalmazt lefedi:

$$\bigcup_{i=1}^n D_i = Z_0.$$

Tekintsük az alábbi igen egyszerű példát:

Metaszint (R_0):

Ha $x_1 = A_{11}$ és $x_2 = A_{21}$ akkor vegyük a D_1 tartományt

Ha $x_1 = A_{12}$ és $x_2 = A_{22}$ akkor vegyük a D_2 tartományt ,

R_1 szabálybázis D_1 tartományhoz:

Ha $x_3 = A_{31}$ és $x_4 = A_{41}$ akkor $Y = B_1$

Ha $x_3 = A_{32}$ és $x_4 = A_{42}$ akkor $Y = B_2$,

R_2 szabálybázis D_2 tartományhoz:

Ha $x_5 = A_{51}$ akkor $Y = B_3$

Ha $x_5 = A_{52}$ akkor $Y = B_4$,

ahol $\Pi = \{D_1, D_2\}$, és $Z_0 = X_1 \times X_2, Z_1 = X_3 \times X_4, Z_2 = X_5$ az $X = \times_{i=1}^5 X_i$ alaphalmaz valódi részhalmazai. Vegyük észre, hogy ekkor $O(T^5)$ szabály helyett elegendő $O(T^2)$ szabály, ami például $T = 7$ érték esetén 16 807 szabály helyett kevesebb mint $2 \cdot 49 + 7 = 105$ szabályt jelent.

Általánosan, ha a k bemenő változót egy k_0 (metaszint) és n darab k_i elemszámú csoportba osztjuk, akkor legrosszabb esetben is

$$r = T^{k_0} + T^{k_1} + T^{k_2} + \dots + T^{k_n} \quad (8.34)$$

számú szabály szükséges, vagyis nagyságrendileg

$$C_{\text{hier}} = O\left(T^{k_0}\right) \cdot O\left(T^{k_1} + T^{k_2} + \dots + T^{k_n}\right) = O\left(n \cdot T^{k_0 + \max_{i=1}^n k_i}\right), \quad (8.35)$$

ami jelentős időmegtakarítást eredményez.

Az imént tárgyalt legegyszerűbb esetben Π az alaphalmaz klasszikus partícióját adja, ahol minden megfigyeléshez pontosan egy tartomány és szabálybázis tartozik. A hierarchikus szabálybázis koncepciója azonban kiterjeszthető. A Π egyszerű partíció általánosításával az egyes tartományok érvényességének meghatározását rugalmasabbá tehetjük. Megengedhetünk fuzzy lefedéseket is, ahol egy tipikus tartomány határán a hozzátartozó szabálybázis érvényessége csökkenhet, sőt egy megfigyeléshez több lokális szabálybázis is tartozhat, és a következtetés a lokális bázisok által meghatározott részkövetkeztetések kombinációjaként áll elő. Továbbgonddolva a lehetőségeket még az is előfordulhat, hogy valamely (meta)szint szabálybázisa ritka; erre az esetre általános megoldást az előző szakaszban tárgyalt interpolációs algoritmusok és a hierarchikus szabálystruktúrák kombinálása jelenthet: a következtetést a metaszinten végzett szabályinterpoláció segítségével határozva meg. Ezen kiterjesztések segítségével a szabálybázis strukturálása általánosabb esetekben is lehetővé válhat. Ez a gondolat nyilván több komoly matematikai és algoritmikus problémát is felvet.

Hogyan kombináljuk a különböző változóhalmazokhoz tartozó lokális szabálybázisokat fuzzy lefedettségű alaphalmaz esetén? Hogyan kezeljük azt, amikor a metaszinten több egymást részben átfedő tartomány tüzel? Hogyan súlyozzuk a különböző lokális szabálybázisok által számított részkonklúziókat, melyek esetleg eltérő változóhalmazokhoz tartoznak?

Lokális szabálybázisok interpolálása esetén először is egy egységes, az érintett változókat tartalmazó legszűkebb részhalmazt kell meghatározni, majd a lokális bázisok minden szabályát ebben a közös, kiterjesztett térben kell felírni. Ekkor azonban elképzelhető az, hogy az összes változót az ily módon meghatározott legszűkebb szupertér tartalmazza, és így a számítási bonyolultság ismét megnő. Ezért [100] szerzői kiterjesztés helyett projekció alapú algoritmust javasoltak, amit az alábbiakban ismertetünk.

Legyen $\tilde{\Pi}$ *ritka fuzzy partíció*, ahol a partíció minden elemében ki van jelölve a változók egy valódi részhalmaza, amely az adott tartományban domináns:

$$\tilde{\Pi} = \{D_1, D_s, \dots, D_n\}, \quad (8.36)$$

ahol

$$\bigcup_{i=1}^n \text{core}(D_i) \subset Z_0$$

valódi részhalmaz értelemben, azaz a partíció fuzzy; sőt

$$\bigcup_{i=1}^n \text{supp}(D_i) \subset Z_0$$

is megengedett, azaz a partíció ritka. Az algoritmus tehát az alábbi lépésekből áll:

1. Határozzuk meg az A^* megfigyelésnek a $\tilde{\Pi}$ partícióra vetített projekcióját (A_0^*). Keressük meg a projektált megfigyeléshez tartozó szomszédos tartományokat (8.36)-ban.
2. Határozzuk meg minden $D_i \in \Pi_i$ -re a hasonlóság mértékét (w_i).
3. Minden $w_i \neq 0$ -ra határozzuk meg R_i -ben A_i^* -ot, az A^* megfigyelés Z_i térre vetített projekcióját. Határozzuk meg A_i^* -gal szomszédos elemeket R_i -ben.
4. Számítsuk ki az R_i -beli ($w_i \neq 0$) B_i^* részkonklúziókat.
5. Helyettesítsük R_0 -beli metaszabályokban a tüzelő R_i lokális szabálybázisokat az általuk generált részkonklúziókkal, és határozzuk meg a végső B^* következtetést a w_i arányok segítségével.

9. fejezet

Alkalmazások

Ebben a fejezetben a fuzzy irányító és szakértő rendszerek gyakorlati alkalmazási lehetőségeit tekintjük át, majd azokat néhány egyszerű példán keresztül illusztráljuk is.

A MAMDANI-típusú irányítók elvének publikálása [125, 130] a 70-es évek végétől egyre több kutató és ipari szakember érdeklődését keltette föl a fuzzy irányítók alkalmazási lehetőségeinek vizsgálata iránt. A kutatások kezdeti eredményeként néhány laboratóriumi prototípus született, majd 1982-ben az első tényleges ipari alkalmazás is megjelent [76], amely egy cementmű irányítását végezte. A kezdeti időszak (1975–1985) eredményeiről a [184] tanulmány ad jó áttekintést. A 80-as évek közepétől a MAMDANI-technikán és variánsain alapuló módszerek igen elterjedtek viszonylag kevés bemenettel rendelkező, explicite nem ismert modellű irányítástechnikai problémák megoldásaiban.

A „fuzzy robbanás” 1987-ben kezdődött Japánban, a *Nemzetközi Fuzzy Rendszerek Szövetsége* (IFSA) Tokióban tartott második világkongresszusával szinte egyidőben, ahol különböző japán egyetemek és vállalatok már számos sikeres fuzzy irányítási alkalmazást mutattak be. Ezek között szerepelt víztisztító berendezés (Sagamihava), vezető nélküli metróvonal (Sendai), mobil robot és több olyan demonstrációs összeállítás, melyek lényegében univerzális célú ipari irányítási rendszerek tulajdonságaival bírtak. A következő években számos kereskedelmi termékben és ipari rendszerben jelent meg a fuzzy irányító, így elsősorban háztartási gépekben (mosógép, porszívó, klímaberendezés, vízmelegítő, rizsfőző, villanyborotva, stb.), a video- és fényképtechnikában (autofókusz, white balance, képstabilizáció), a gépjárműgyártásban (fogyasztáscsökkentés, ABS-rendszer, stb.), víz- és levegőtisztító, illetve szellőző rendszerekben, ipari és mobil robotokban (ideértve a talán legfejlettebb ilyen alkalmazást, a „repülő robotot”, azaz mezőgazdasági célú vezető nélküli helikoptert is [172, 174], melyet SUGENO laboratóriumában, a Tokiói Műszaki Egyetemen fejlesztettek ki), és számos más területen.

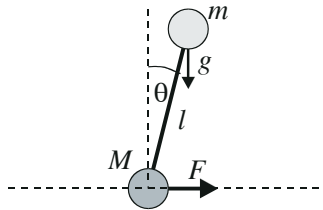
Az ipari alkalmazásokban Japánt először Dél-Korea és Tajvan követte, majd Európában elsősorban Németország (Siemens, Volkswagen, stb.), de

más országok is, például Olaszország, s egyidejűleg Új-Zéland, stb. Az Egyesült Államok alkalmazott fuzzy kutatása a felsorolt témakon kívül az űrkutatás és a haditechnika területére koncentrál, utóbbi téren a legnagyobb sikert az Öböl-háborúban alkalmazott éjszakai célazonosító-rendszer aratta, mely fuzzy eljárásokat alkalmazva infravörös képeket osztályozott, és lehetővé tette a nem harckocsiként azonosított célpontok megkímélését.

Mára a fuzzy eljárást (elsősorban irányító vagy szakértő rendszert) alkalmazó kereskedelmi termékek száma rendkívül jelentős.

9.1. Egy demonstrációs példa: a fordított inga szabályozása

A fuzzy irányítási rendszerek működésének illusztrálását először egy egyszerű példán, a fordított inga szabályozásán mutatjuk be. Az irányítás célja egy vízszintes tengellyel rögzített rúd függőleges helyzetben való tartása, melyet a tengelyt tartó kocsi vízszintes irányú mozgatásával érünk el (ld. 9.1. ábra). Az egyszerűsített fizikai modell a rúd aljánál lévő M , és a rúd felső részén lévő m tömegpontból áll. A két tömegpont egy tömör, elhanyagolható tömegű, l hosszúságú rúddal van összekötve. Az inga egyensúlyi helyzetbe való visszahozásához (megtartásához) szükséges F erő meghatározására a rúd függőlegessel bezárt ϑ szögét, és ennek a szögnek a $\Delta\vartheta$ -ból becsült $\dot{\vartheta} = \frac{d\vartheta}{dt}$ változását (közelítő szögsebesség) mérjük. Tehát a rendszer bemenő változói ϑ és $\Delta\vartheta$, ezek aktuális értéke a megfigyelés. Az irányítás célja, hogy a megfelelő F mozgató erő segítségével mindkét értéket nullán tartsuk.



9.1. ábra. Fordított inga esetén fellépő erőhatások

Elsőként a bemeneti és kimeneti alaphalmazt kell meghatároznunk. A ϑ szög értéke az $X_1 = [-90, 90]$ fokos tartományban változhat. Elméletileg a $\Delta\vartheta$ szögsebesség értéke bármekkora lehet, de egyrészt szélsőséges értékeket csak mesterségesen idézhetünk elő, másrészt a mérőeszköz is csak egy

adott mérési tartományban működik. Ezért a továbbiakban feltesszük, hogy $\Delta\vartheta \in X_2 = [-45, 45]$ (fok/másodperc). Hasonló megfontolások alapján a kimenő változó alaphalmazául az $Y = [-10, 10]$ (N) tartományt határozzuk meg.

A hagyományos szabályozás a modell formális, differenciálegyenlet-rendszer formájában megadott leírásán alapszik. Ennek a differenciálegyenlet-rendszernek a megoldása adja meg a megfelelő irányítási értéket.

A fordított inga modellje differenciálegyenlet segítségével is leírható:

$$(m + M) \cdot \sin^2 \vartheta \cdot l \cdot \ddot{\vartheta} + m \cdot l \cdot \sin \vartheta \cdot \cos \vartheta \cdot (\dot{\vartheta})^2 - \\ -(m + M) \cdot g \sin \vartheta = -F \cdot \cos \vartheta,$$

ahol g a gravitációs állandó. A cél az $F = F(t)$ erőnek meghatározása az egyenletből úgy, hogy a ϑ és $\dot{\vartheta}$ változók lehetőleg gyorsan nullához konvergáljanak. Általában ahhoz, hogy az egyenletrendszer megoldása hatékony szabályozást eredményezzen, előfeltétel, hogy a modell jól közelítse a valóságot, melyhez a fizikai folyamat alapos ismerete szükséges.

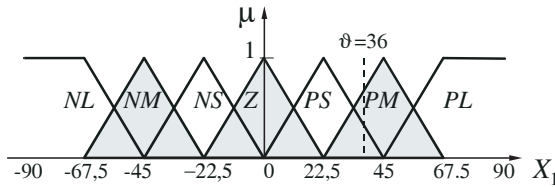
A folyamat matematikai modelljének differenciálegyenletekkel való pontos leírása azonban sok esetben lehetetlen, vagy legalábbis rendkívül bonyolult feladat. Nyilvánvaló, hogy az ilyen rendszerek szabályozása többnyire a fizikai-matematikai modell pontos ismerete nélkül is megvalósítható. Ezért tud például szinte bárki kerékpározni anélkül, hogy akárcsak tudna a differenciálegyenletek létezéséről. A rendszer irányításához elegendő, ha a rendelkezésünkre áll a rendszer kvalitatív működését leíró

R_i : Ha a szög $\vartheta = A_{i,1}$ és a szögsebesség $\Delta\vartheta = A_{i,2}$ akkor az erő $F = B_i$

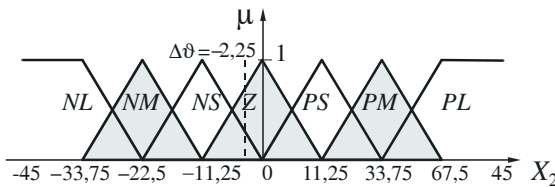
alakú szabályok halmaza, ahol $A_{i,1}$, $A_{i,2}$ és B_i fuzzy halmazokkal reprezentált nyelvi kifejezések. A szabályok definiálása előtt meg kell határozni a fuzzy partíciókat, azaz azt, hogy milyen nyelvi kifejezésekre, illetve fuzzy halmazokra osztjuk fel az alaphalmazokat.

Az X_1 alaphalmazt a 9.2. ábrán látható módon hét fuzzy nyelvi kifejezésre osztjuk fel, melyek a két szélsőtől eltekintve egyenlőszárú háromszög alakúak. Nagyon hasonló megoldást alkalmazunk a másik két alaphalmaz particionálásakor (9.3. és 9.4. ábra). A nyelvi kifejezések értékeire a 116. oldalon definiált jelöléseket használjuk.

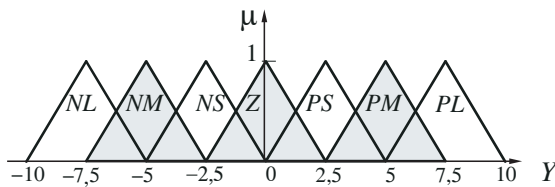
A fordított inga szabályozásához a 9.1. táblázatban megadott szabályokat alkalmazzuk. A szabályok értelmezése a következő módon történik (tekintsük például az első sor harmadik oszlopát): „Ha a szög kicsi negatív



9.2. ábra. Az X_1 alaphalmaz és a mért szög lehetséges értékei



9.3. ábra. Az X_2 alaphalmaz és a becsült szögsebesség lehetséges értékei



9.4. ábra. Az Y alaphalmaz és a mozgató erő értékei

és a becsült szögsebesség nagy negatív, akkor az erő legyen kicsi pozitív”;

Ha $\vartheta = NS$ és $\Delta\vartheta = NL$, akkor $F = PS$.

A táblázat nem definiál szabályt minden lehetséges bemenet esetére, az üres pozíciók olyan szituációkhoz tartoznak, melyek a gyakorlatban fizikailag nem fordulnak elő. Ennek ellenére, adódhat olyan helyzet, hogy a megfigyelés egyetlen szabályt sem aktivál. Ez bizonyos rendszerek esetén katasztrófához vezethet, azaz ilyenkor a modell nem alkalmazható, más következtetési módszert kell alkalmazni, például szabályinterpolációs/extrapolációs eljárásokat (ld. 8. fejezet és 9.2. szakasz). Az adott példában megfelelő kiindulási pozíció esetén nem fordulhat elő olyan szélsőséges szituáció, ahol nem lehet (az adott erőhatárok között) a rúd eldőlését megakadályozni.

9.1. táblázat. Fordított inga (hiányos) szabálybázisa

ϑ $\Delta\vartheta$	NL	NM	NS	Z	PS	PM	PL
NL			PS	PL			
NM				PM			
NS	NM		NS	PS			
Z	NL	NM	NS	Z	PS	PM	PL
PS				NS	PS		PM
PM				NM			
PL				NL	NS		

Az inga irányítását MAMDANI következtetési algoritmussal végezzük. Legyen például az aktuális megfigyelés

$$\vartheta = 36^\circ, \quad \Delta\vartheta = -2,25^\circ.$$

Amint az a 9.2. és 9.3. ábrán látható, a megfigyelésnek két szabály anteceden-sével van nem üres metszete. Ennek alapján a 9.1. táblázat felhasználásával az

R_1 : Ha $\vartheta = PS$ és $\Delta\vartheta = Z$, akkor $F = PS$

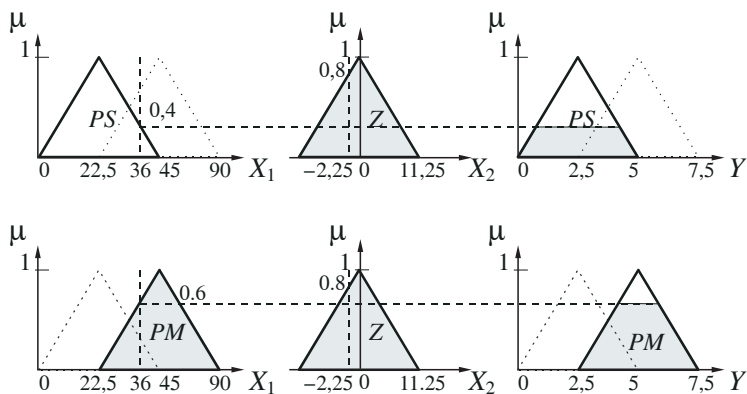
R_2 : Ha $\vartheta = PM$ és $\Delta\vartheta = Z$, akkor $F = PM$

szabályok tüzelnek. Az első lépés a súlyfaktorok meghatározása (vö. (7.7) és (7.8); 9.5. ábra):

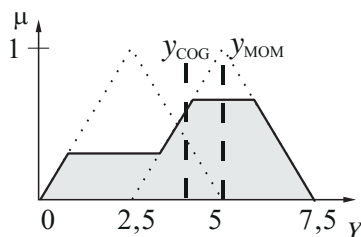
$$w_1 = \min\{0,4, 0,8\} = 0,4 \text{ és } w_2 = \min\{0,6, 0,8\} = 0,6.$$

A részkonklúziók meghatározása a (7.9) egyenlet alapján történik (ld. 9.5. ábra), a végső következtetést pedig ezek uniójaként kapjuk. Az eredmény a 9.6. ábrán látható. Ha crisp következtetésre van szükségünk, akkor a 7.4. szakaszban ismertetett defuzzifikációs módszerek közül alkalmazhatjuk valamelyiket. Például a maximumok közepe módszerrel (MOM) $y = 5$, a geometriai középpont módszerrel (COA) $y \approx 3,95$, azaz a döntés-től függően 3,95 vagy 5 N erőt kell alkalmaznunk az inga egyensúlyban tartásához.

Amint a példa is mutatja, akár ilyen jelentős eltérés is adódhat a kü-lönböző defuzzifikációs módszerek által számított eredmények között. Ez



9.5. ábra. Részkonklúziók meghatározása



9.6. ábra. A következtetésként kapott fuzzy halmaz és a két defuzzifikációs módszer eredménye

a magyarázata annak, hogy a szabályok beállítása általában bonyolult „hangolási folyamat” (tuning) keretében történik. A tagsági függvény alakjának megváltoztatása, helyzetének módosítása ugyanis kompenzálhatja a defuzzifikáció eltérését — vagy éppen fordítva.

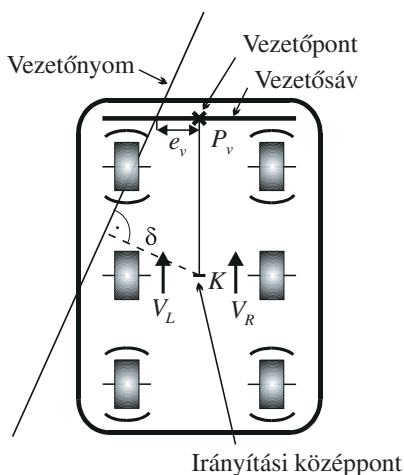
9.2. Vezetőnélküli targonca irányítása

Ebben a szakaszban egy ún. vezetőnélküli targonca irányítását mutatjuk be [36, 116], melyet anyagmozgató feladatokra használnak nagy raktárakban. Először egy hagyományos, MAMDANI-típusú irányítást mutatjuk be a feladatnak, majd pedig a redundáns szabályok elhagyását követően a KH-féle szabályinterpolációs eljárás egy változatát alkalmazzuk az irányítás elvégzésére.

9.2.1. A targonca modellje és irányítási stratégiája

Az irányítás általában az ún. *vezetőnyom* alapján történik [70], ez a megoldás a vezérlő rendszer egyszerűsége miatt igen népszerű. A vezetőnyom a valóságban többnyire festett jelzés, de lehet a padlóra ragasztott, vagy abba beépített vezeték vagy mágnescsík is. A targonca irányításának egyik célja a vezetőnyom követése, melyet a targoncán elhelyezett ún. vezetőnyom-érzékelő(k) segítségével valósítanak meg. A bemutatott példában modellezett vezetőnélküli targoncának két rögzített irányú hajtott kereke van, melynek irányítása differenciális kormányzással (tankhajtással) történik, azaz a járművet a kerekek fordulatszámának (egymástól eltérő) módosításával lehet kormányozni. A fordulási képességet a hajtott kerekek fordulatszámának különbsége határozza meg. A nyomkövető rendszer nem rendelkezik térképpel, ezért mindig a vezetőnyomot leíró pillanatnyi, illetve a megelőző mérési adatok alapján kell navigálni. Az irányítás másik célja a járműnek a kijelölt beállási (dokkolási) pozícióba való eljuttatása. Összefoglalva, olyan irányítási (nyomkövető) stratégiát keresünk, amely a teljes útvonalon biztosítja a nyomkövetési hiba minimális szinten tartását, valamint a beállási távolság minimalitását.

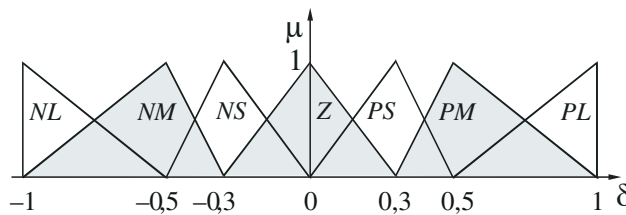
Az irányítási stratégia a rendszert ismerő szakértőtől megszerzett információk segítségével adható meg. A lényege igen egyszerű: tartsuk a lehető legközelebb a targonca irányított középpontját (azaz a hajtott kerekek tengelyvonalának felezőpontját; ld. 9.7. ábra) a vezetőnyomhoz, majd ha ez



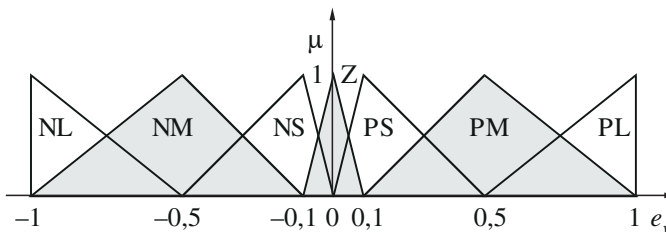
9.7. ábra. A vezetőnélküli targonca modellje

a pont elég közel van már a vezetőnyomhoz, fordítsuk a járművet a dokkolási irányba. Annak érdekében, hogy a targonca a vezetőnyomot minél hamarabb elérje az ún. *vezetősáv* használata javasolt. Ennek lényege, hogy a vezetőnyomtól való távolság mérése nem egy kijelölt ponton, hanem egy összefüggő szakaszon, több érzékelő felhasználásával történik. Ekkor a kormányzás célja kevésbé szigorú, mégpedig az, hogy a vezetősáv által kijelölt tartomány ne hagyja el a vezetőnyomot, illetve, hogy annak helyzetét mindig érzékelni lehessen a rendelkezésre álló érzékelők valamelyikével. Ez a módszer biztosítja a vezetőnyom gyorsabb megközelítését [36].

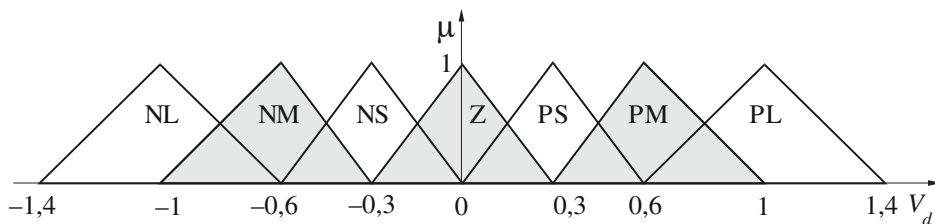
A következőkben ismertetésre kerülő irányítási stratégiához csupán két adatot kell mérni: a vezetőnyom és az irányítási középpont, valamint a vezetőnyom és a vezetőpont távolságát (9.7. ábra). A vezetőnyom és a vezetőpont távolsága a vezetősáv felhasználásával meghatározható ugyan, de a nyomvonalkövetés hibája ekkor még nem áll rendelkezésre. A nyomvonalkövetés hibájának pillanatnyi értékét [36] alapján a vezetőpont és vezetőnyom távolságának előző és aktuális értékei segítségével becsülhetjük. A fenti számítás elvégzése után a megfigyelést a becsült nyomvonalkövetés hibája (δ), valamint a vezetőnyom és a vezetőpont távolsága (e_v) alkotja.



9.8. ábra. A becsült nyomvonalkövetés hibájának (δ) fuzzy partíciója



9.9. ábra. A vezetőnyom és vezetőpont távolságának (e_v) fuzzy partíciója



9.10. ábra. A pillanatnyi irány (V_d) fuzzy partíciója

9.2.2. Irányítás Mamdani-módszerrel

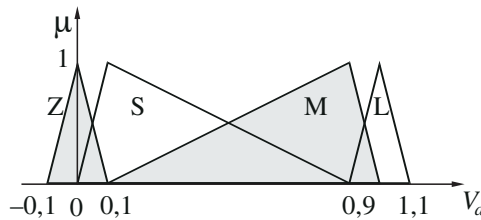
A következő lépés a szabálybázis felépítése. Olyan szabályokat veszünk fel a szabálybázisba, amelyek jellemző kiindulási helyzetek esetére írja le azokat a szükséges manőverezési (sebesség és irányváltoztatás) utasításokat, amelyek a minimális dokkolási távolság közelítő elérését garantálják. A kimeneti változók a jármű sebessége (V_a) és iránya (V_d). Tankhajítás esetén ezek a mennyiségek a

$$V_d = V_L - V_R, \quad V_a = \frac{V_L + V_R}{2}$$

egyenletek segítségével számolhatók ki, ahol V_L és V_R a bal, illetve jobb oldali kerék kerületi sebességét jelöli. A szabályok két csoportba sorolhatók; az elsőbe a sebességet, a másodikba az irányt meghatározó szabályok tartoznak. Mindkét fajta szabálynak két antecedense és egy konzekvensze van.

A bemeneti alaphalmazokat mindkét változó esetében a $[-1, 1]$ intervallumra vetítettük, és ezeken hét-hét fuzzy halmazt definiáltunk, melyek RUSPINI-partíciót alkotnak. Az alaphalmazok partíciói a 9.8. és 9.9. ábrán láthatók. A kimeneti alaphalmaz a sebesség esetén a $[-0, 1, 1, 1]$, az irány meghatározásához a $[-1, 4, 1, 4]$ intervallum, melyek négy, illetve hét fuzzy halmazra vannak particionálva (ld. 9.10. és 9.11. ábra).

A szabályokban szereplő nyelvi fogalmak leírását a kiinduláskor egyenlőszárú háromszög alakú fuzzy halmazokkal valósíthatjuk meg, amelyeket az irányítás optimalizálása céljából hangolni kell. Ennek érdekében elkészítettük egy működő vezetónélküli targonca szimulációs modelljét. A szimuláció során megkíséreltük a lehető legkisebb dokkolási távolság elérését az adott vezetónyomon. Az így kapott eredmények segítségével módosítottuk a szabályokban szereplő halmazok csúcspontjának pozícióját, ennek eredményei láthatók a 9.10. és 9.11. ábrán. A szabályokat a 9.2. és 9.3.



9.11. ábra. A pillanatnyi sebesség (V_a) fuzzy partíciója

táblázatok tartalmazzák.

9.2. táblázat. Vezetőnélküli targonca pillanatnyi irányának (V_d) meghatározására használt szabályok

δ e_v	NL	NM	NS	Z	PS	PM	PL
NL	PM	PS	Z	Z	NL	NL	NL
NM	PL	PS	PS	PS	PS	Z	NL
NS	PL	PM	PS	PS	Z	Z	NL
Z	PL	PM	PS	Z	NS	NM	NL
PS	PL	Z	Z	NS	NS	NM	NL
PM	PL	Z	NS	NS	NS	NS	NL
PL	PL	PL	PL	Z	Z	NS	NM

A 7.3. szakaszban ismertetett max-min kompozíciós következtetési eljárással és súlypont defuzzifikációs módszerrel (ld. 7.4.1. pont) végzett irányítás esetén a kapott („behangolt”) szabálybázissal a 9.12. ábrán látható irányítási felületeket nyertük.

Az így létrehozott szabálybázisnak és a MAMDANI-algoritmussal történő irányítás teljesítményének ellenőrzése a szimulációs modell segítségével elvégezhető. Az eredmény — a legkisebb vezetőnyom sugarának függvényében kifejezett minimális dokkolási távolság — azt mutatja, hogy a modell a targoncát kielégítően irányítja, és a [36] közleményben publikált vezetőszám bevezetése a vezetőnyom megközelítésének sebességét észrevehetően javítja. A probléma további részletes vizsgálata KOVÁCS SZ. munkáiban található meg [36, 116, 117].

9.3. táblázat. Vezetőnélküli targonca pillanatnyi sebességének (V_a) meghatározására használt szabályok

δ e_v	NL	NM	NS	Z	PS	PM	PL
NL	M	S	S	S	S	Z	Z
NM	S	M	M	M	M	M	S
NS	Z	S	L	L	L	M	S
Z	S	M	L	L	L	M	S
PS	S	M	L	L	L	S	Z
PM	S	M	M	M	M	M	S
PL	Z	Z	S	S	S	S	M

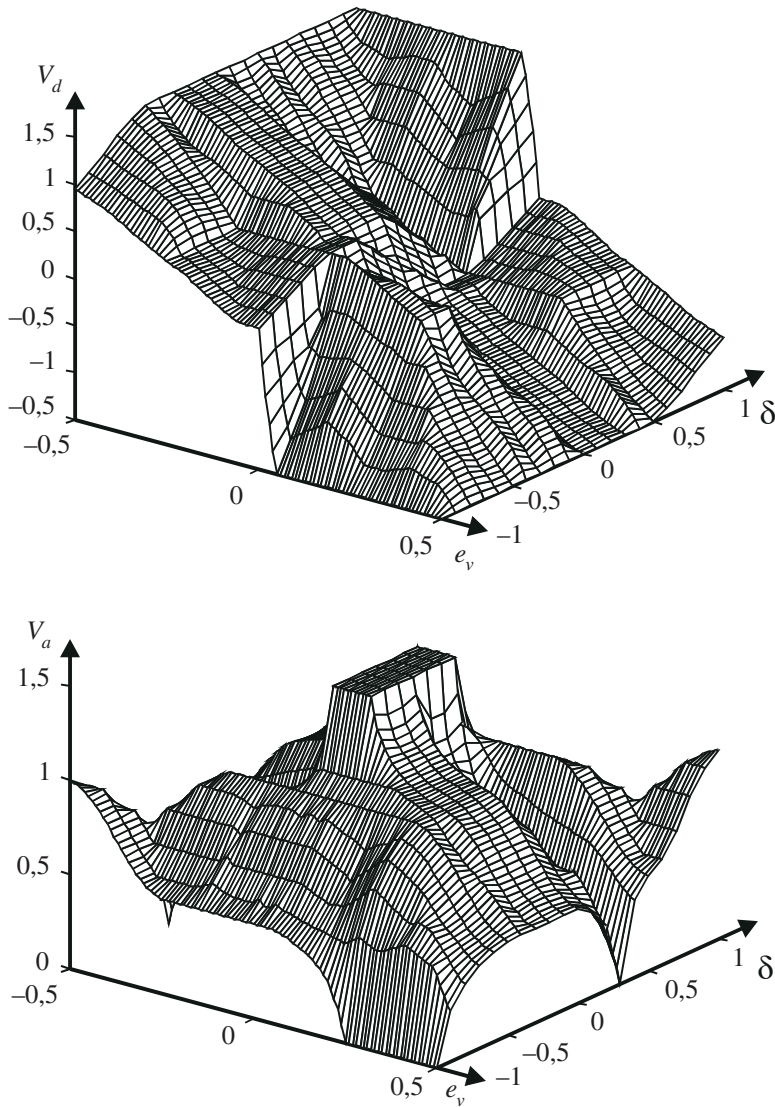
9.2.3. Írányítás szabályinterpolációs módszerrel

A 9.2. és 9.3. szabálybázisok együttesen kétszer 49, azaz összesen 98 szabályt tartalmaznak. Ezek között természetesen vannak olyan szabályok, melyek elhagyhatók, illetve más szabályok segítségével kiválthatók.

A szabálybázis redukcióját egyes redundáns fuzzy halmazok elhagyásával kezdjük. A redukált szabálybázis két bemenetén öt-öt, kimenetén három, illetve öt fuzzy halmazt definiálunk. Ezután elhagyjuk a más szabályok segítségével előállítható szabályokat. A végeredményül kapott redukált szabálybázis, mely nem redukálható tovább, a 9.4. és 9.5. táblázatokban található szabályokat tartalmazza. A redukált szabálybázis az egyes kimenetekben tizenkét, illetve öt szabályt, vagyis összesen tizenhét szabályt tartalmaz. Ez az eredeti szabálybázis méretének kevesebb, mint 35%-a.

9.4. táblázat. A pillanatnyi irány (V_d) redukált szabálybázisa

δ e_v	NL	NM	Z	PM	PL
NL				NL	
NM	PL		PS	PS	NL
Z		PL		NL	
PM	PL	NS	NS		NL
PL		PL			



9.12. ábra. A pillanatnyi irány (V_d) és sebesség (V_a) irányítási felülete

Érdekes megfigyelni, hogy míg az irány meghatározásánál a (Z, Z) bemenethez tartozó szabály és konzekvensé a környező szabályok segítségével pótolható, addig a sebesség esetében ez az egyik legfontosabb szabály, mely az elhagyható, környező szabályok rekonstrukciójában alapvető szerepet játszik. A szabályinterpolációs eljárással végzett irányítási algoritmus

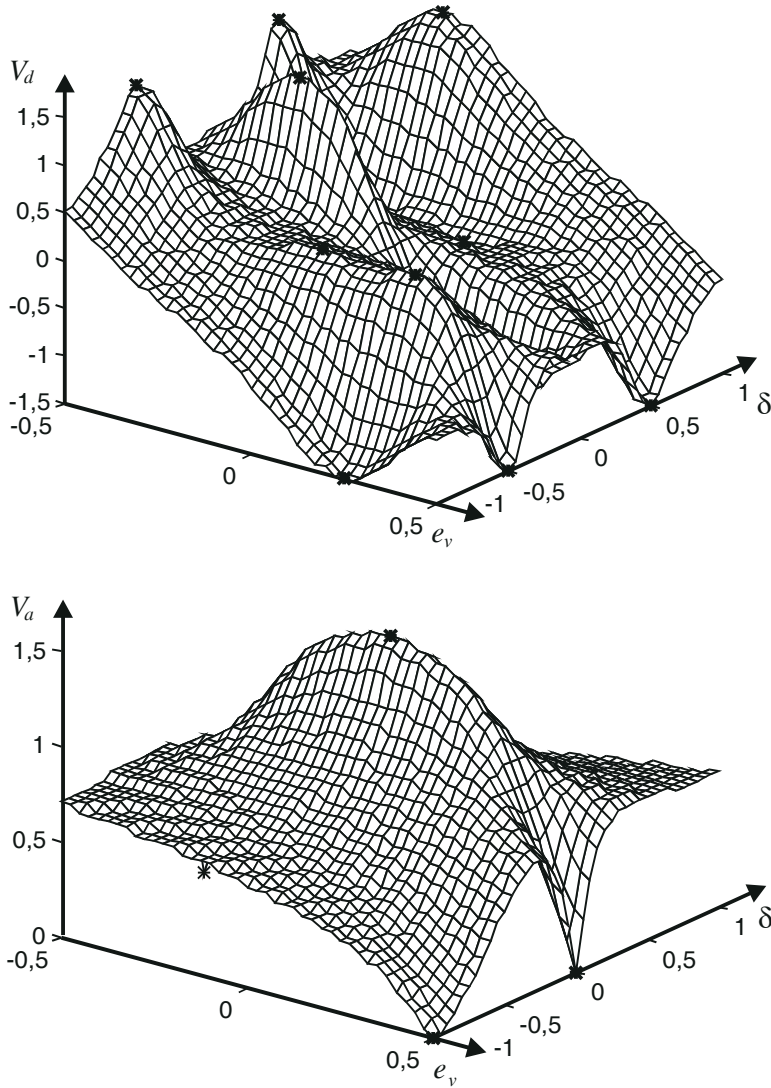
a 9.13. ábrán látható irányítási felületeket generálja.

Ha még a redukált szabálybázis is meglehetősen nagyméretű, akkor további lehetőség a redukálásra a szabálybázis tömörítése [5, 8, 9]. Ekkor a szabálybázist leíró paramétertömböt egy tömörítő eljárással becsomagoljuk, és a következtetés számítása közben „interaktívan” csak a tüzelő szabályok paramétereit emeljük ki anélkül, hogy az egész paramétertömböt kitömörítsük. Természetesen ekkor a tüzelő szabályok konzekvensének számítása némileg több időt igényel. Azonban ha a tömörített szabálybázis elfér az operatív memóriában, akkor a nagyobb elérési időt igénybe vevő merevlemez használata nem szükséges, s így összességében az eljárás válaszadási ideje jelentősen csökkenhet.

További lényeges redukciót eredményezhet, ha sikerül az irányítási probléma állapotterét olyan módon alterekre particionálni, hogy az így nyert egyik alterben metasabálybázis állítható fel, a megmaradó alterben vagy alterekben pedig a fennmaradó változók száma lokálisan csökkenthető (ld. 8.6. szakasz).

9.5. táblázat. A pillanatnyi sebesség (V_a) redukált szabálybázisa

δ e_v	NL	NM	Z	PM	PL
NL					Z
NM					
Z	S		L		S
PM					
PL	Z				



9.13. ábra. A pillanatnyi irány (V_d) és sebesség (V_a) irányítási felülete szabályinterpolációs eljárás esetén

10. fejezet

Evolúciós algoritmikus módszerek

Az evolúciós módszerek a természetben lejátszódó, az élőlények biológiai folyamatait utánzó optimalizációs technikák. Ezen módszerek előnye, hogy képesek megoldani a problémákat és közelítő megoldást találni akkor is, ha a probléma nemlineáris, sokdimenziós, nem folytonos. Hatékony eszköznek bizonyulnak nemlineáris, multikritériumú, kényszerekkel kiegészített optimalizációs feladatok megoldására. Alapelvük a megoldások egy populációján történő keresés, melyet a biológiából megismert törvényszerűségek vezérelnek. A modellezni kívánt rendszertől semmilyen speciális tulajdonságot nem követelnek meg. Ebben a fejezetben ennek a területnek az alapjait tekintjük át. Az első részben a klasszikusnak számító genetikus algoritmusokat mutatjuk be röviden. Utána a genetikus programozást részletezzük, majd pedig a bakteriális evolúciós algoritmusokat tárgyaljuk. Végezetül egyéb technikákra is utalunk.

10.1. Genetikus algoritmusok

A *genetikus algoritmusok* alapötlete J. Holland-tól származik [75], aki Darwin evolúciós elméletét [38] felhasználva hatékony optimalizációs módszert fejlesztett ki. Az evolúciós elmélet szerint az élőlénypopulációk folyamatosan fejlődnek, az élőlények egyre tökéletesebbé válnak, egyre jobb egyedek fejlődnek ki. A fejlődési folyamatot a természetes kiválasztódás vezérli, jobb egyedek nagyobb eséllyel maradnak fent, míg a gyengébb élőlények elpusztulnak. Ezt az elvet felhasználva, az evolúciós folyamatot szimulálva optimalizálási feladatok megoldására alkalmas algoritmusokat kapunk. Ezekben az algoritmusokban egy egyed a feladat egy megoldását jelenti. A populáció folyamatos fejlődése biztosítja, hogy egyre jobb egyedeket kapunk, azaz a problémát egyre jobban sikerül megoldani, a közelítés egyre pontosabb lesz.

10.1.1. Gyakran használt fogalmak

Az alábbiakban összefoglaljuk a evolúciós algoritmikus módszerekben leggyakrabban használt fogalmakat:

gén: funkcionális entitás, amely az egyed egy speciális tulajdonságát kódolja (pl. fuzzy szabály)

allél: a gén értéke (pl. a fuzzy szabályt alkotó fuzzy halmazok paramétereinek az értékei)

genotípus: az egyed alléljainak kombinációja

fenotípus: az egyed tulajdonságainak az összessége

egyed: kromoszóma, egy jelölt a feladat megoldására

populáció: egyedek összessége

generáció: egy időben létező egyedek

alkalmassági (fitness) függvény: az egyedek jóságának, alkalmasságának mérése

evolúció: a populáció fejlődése, tökéletesedése

szelekció, kiválasztás: bizonyos egyedek vagy egyedcsoportok életben maradását vagy szaporodását gátolja, másokét pedig megengedi

keresztelés: két egyed kromoszómájának kombinálása

mutáció: véletlenszerű változás a kromoszómában

migráció: egyed vándorlása egyik populációból a másikba

konvergencia: az egyedek tulajdonságainak fokozatos közeledése egymáshoz és az optimumhoz

10.1.2. Az algoritmus

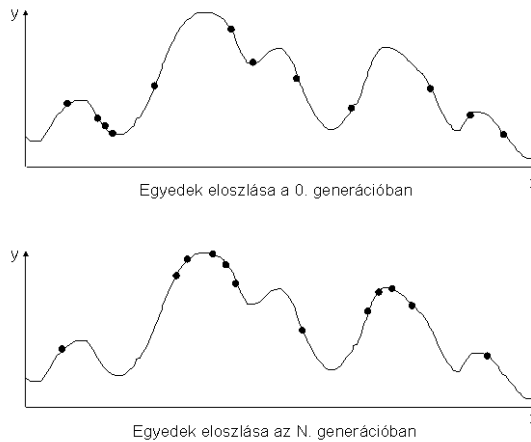
Az eredeti genetikus algoritmus folyamata a következő:

```
kezdeti populáció létrehozása
generáció=0
amíg generáció < max. generáció
{
    egyedek rangsorolása az alkalmassági érték alapján
    szelekció
    keresztezés
    mutáció
    visszahelyettesítés
    generáció = generáció + 1
}
```

Először a kezdeti populációt alakítjuk ki. A populáció egyedei a probléma egy-egy megoldását adják. Megkülönböztetünk bináris és valóskódolású genetikus algoritmusokat. Előbbinél az egyedek kromoszómája egy bináris string, utóbbi esetén pedig valós számokból álló vektor. A kezdeti populáció kialakítása véletlenszerűen történik, a keresési térből N_{egyed} pont kiválasztásával. Az algoritmus futása során az egyedek egyre jobban közelednek az optimális megoldás felé. Ezt szemlélteti a 10.1. ábra maximumkeresési probléma esetén. A maximális generáció szám és az egyedek száma (N_{egyed}) paraméterek, melyeket előre megadunk. A maximális generáció szám helyett megadhatunk más leállási feltételt is, például megfelelő megoldás elérésekor megállhatunk.

10.1.3. Az alkalmassági (fitness) függvény

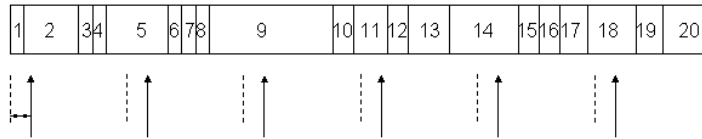
Az *alkalmassági (fitness) függvény* segítségével rangsoroljuk az egyedeket a populációban. Az algoritmus megfelelően hatékony működéséhez fontos az alkalmassági függvény helyes megválasztása. Minél nagyobb egy egyed alkalmassági értéke, annál nagyobb a túlélési esélye. Minimumkeresési probléma esetén az adott függvényt transzformáljuk az alkalmassági érték kiszámításához, hiszen minél kisebb a függvény értéke annál nagyobb alkalmassági érték tartozik hozzá. Ezenkívül az alkalmassági értékek pozitívak. Gyakran egy probléma megoldása során a cél a hibaérték minimalizálása. Ilyenkor a hibából valamilyen transzformációval állíthatjuk elő az alkalmassági értéket.



10.1. ábra. Az evolúció folyamata egy maximumkeresési példában

10.1.4. Szelekció

A szelekció (kiválasztás) során azokat az egyedeket választjuk ki a populációból, melyek az utódok létrehozásában fognak részt venni. Egy egyed több utódot is létrehozhat, az alkalmassági értékétől függően. A legegyszerűbb kiválasztási algoritmus a rulett kerék módszer, vagyis a sztochasztikus kiválasztás cserével („stochastic sampling with replacement”). A populációt egy rulett keréken ábrázoljuk, ahol az egyes egyedekhez az alkalmassági értékükkel arányos méretű körcikk tartozik. A kerék kerülete az egyedek alkalmassági értékeinek összege. Ezután generálunk annyi véletlen számot a $[0, \text{alkalmassági-összeg}]$ tartományban, ahány egyed van a populációban. Amelyik egyedhez tartozó körcikkbe a generált szám beleesik, azt az egyedet kiválasztjuk. Így a nagyobb alkalmassági értékű egyedek nagyobb valószínűséggel kerülnek kiválasztásra, többször is kiválasztódhatnak (azaz többször szerepelhetnek majd szülőként), míg a kis alkalmassági értékkel rendelkezők ritkábban vagy egyszer sem. Ennél az eljárásnál van egy hatékonyabb, általánosan használt módszer, a sztochasztikus univerzális kiválasztás („stochastic universal sampling”). Itt nem cseréljük le az összes egyedet a populációban, hanem csak az egyedek bizonyos hányadát. A populációnak csak egy részét jelöljük ki utódok létrehozására. A szülők számát tehát úgy határozzuk meg, hogy a populáció méretét megszorozzuk egy 0 és 1 közötti számmal. Az alkalmas egyedek többször is lehetnek szülők, éppúgy, mint az előző módszer esetén. A kihalók száma – vagyis akik-

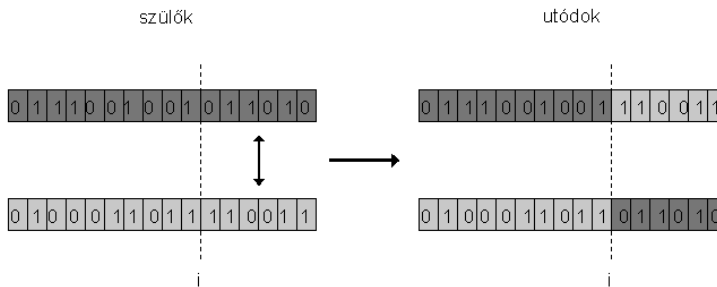


10.2. ábra. A szelekció művelete

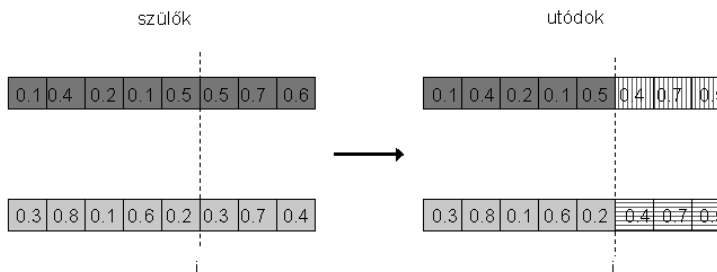
nek a helyére az utódokat tesszük majd – megegyezik a szülők számával. A kiválasztás a rulett kerék módszerhez hasonlóan történik, de most csak egy véletlenszám generálásával a $[0, \text{összalkalmassági/szülők száma}]$ intervallumban. Ehhez a számhoz adjuk hozzá az $\text{összalkalmassági/szülők száma}$ érték többszöröseit. Amelyik körcikkkbe mutat az aktuális szám, az ahhoz tartozó egyedet kiválasztjuk. A 10.2. ábrán például egy 20 egyedből álló populáció látható egy kiterített rulett keréken. A szülők aránya legyen 0,3, azaz 6 egyed kerül kiválasztásra. A kiválasztott egyedek a 10.2. ábrán: a 2., 5., 9., 11., 14., és a 18. egyed.

10.1.5. Keresztezés

A kiválasztott egyedeket párokba rendezzük, és a párok (szülők) között végrehajtjuk a *keresztelést* (*crossover*). Ez a művelet valósítja meg a két szülő „genetikus anyagának” a keverését. Egy pontos keresztezés esetén generálunk egy egyenletes eloszlású i véletlenszámot a kromoszóma hosszának intervallumában. A két egyed között kicseréljük az információt az i indextől balra és jobbra, így két új egyed, utód keletkezik (10.3. ábra). Többpontos keresztezés esetén több véletlenszerű keresztezési pontot választunk, és felváltva cseréljük ki a két egyednél az információt a keresztezési pontok között. Ez azonban az egy pontos keresztezés többszöri alkalmazásával ekvivalens, így nem jelent lényegi változást. Lehetséges uniform keresztezés használata is, melynek során a kromoszóma minden egyes génjénél véletlenszerűen döntünk, hogy melyik szülő bitjét kapja az első utód és melyiket a második. Valós kódolású genetikai algoritmusok esetén is többféle keresztezési módszer közül választhatunk. Az egyik lehetséges változatot mutatja a 10.4. ábra.



10.3. ábra. A keresztezés operáció (bináris példa)



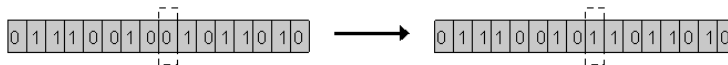
10.4. ábra. A keresztezés operáció (valós példa)

10.1.6. Mutáció

Az egyed génkészletének véletlenszerű megváltozása a *mutáció*. Az algoritmusban ez a kromoszóma egy véletlenszerűen kiválasztott bitjének vagy bitscsoportjának megváltoztatását jelenti (10.5. ábra). Valós kódolású esetben a kiválasztott gén értékét változtatjuk meg a lehetséges intervallumból új értéket véve. A mutáció meghatározó paramétere a mutációs arány, amely azt adja meg, hogy az egyed mekkora eséllyel vesz részt a mutációban.

10.1.7. Visszahelyettesítés

Az egyszerű rulett kerék módszerrel történő kiválasztási algoritmus esetén a populáció összes egyedét lecseréljük. Ilyenkor tehát a következő generációt csupa új egyed fogja alkotni, melyeket a szelektálás utáni keresztezés és mutáció operátorokkal kapunk meg. Sztochasztikus univerzális



10.5. ábra. A mutáció művelete

kiválasztás esetén viszont a populációnak csak valamekkora hányadát választottuk ki (10.2. ábra). Ahhoz, hogy a populáció mérete ne változzon, szükséges, hogy meghatározzuk azon egyedeket, amelyeket megsemmisítünk, vagyis azokat, melyek helyére az új egyedeket tesszük. A kihalt egyedeket hasonlóan választhatjuk ki, mint ahogy a szelekciónál a szülő egyedeket, viszont a kihaltók esetében olyan rulett kereket definiálunk, ahol az egyes egyedek a jósági értékükkel fordítottan arányos méretű körcikket kapnak. A többi egyed, amelyek nem vettek részt új egyedek létrehozásában és ki sem haltak, változatlanul hagyjuk a populációban, azaz a következő generációban is benne lesznek.

10.1.8. Migráció

Több populációs esetben a populáció alpopulációkból áll. Az egyes alpopulációk a fent leírt algoritmussal kezelhetők. Lehetőség van azonban egyedek vándorlására a populációk között. Ezt nevezzük *migrációnak*, mellyel kapcsolatban szükséges annak a megadása, hogy az egyedek hány százaléka vesz részt benne, milyen alapon kerülnek kiválasztásra az egyes egyedek, és hogy milyen topológia szerint játszódik le [146].

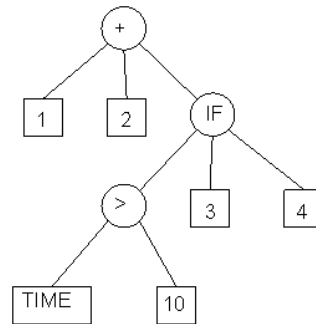
10.2. Genetikus programozás

A *genetikus programozás* ötletét J. Koza javasolta először 1992-ben [118]. A módszer a genetikus algoritmus alapötletét használja fel egy adott feladat megoldására alkalmas számítógépes programok terében való keresésre. Számos különbözőnek tűnő, különböző területről vett problémát át lehet fogalmazni olyan feladattá, amely a problémát megoldó optimális programot keresi. A genetikus algoritmusokhoz képest az az alapvető különbség, hogy az egyedek itt nem egyetlen stringbe vannak bekódolva, hanem kifejezésfáival adottak. A 10.6. ábrán egy program és a kifejezésfája látható. A fát kétféle csomópontok alkotják, a *terminális csomópontok*, melyek a fa leveleiben helyezkednek el és a *függvény csomópontok*, amelyek a fa többi

```

int prog (int time)
{
    int temp1, temp2;
    if (time > 10)
        temp1 = 3;
    else
        temp1 = 4;
    temp2 = temp1 + 1 + 2;
    return (temp2);
}

```



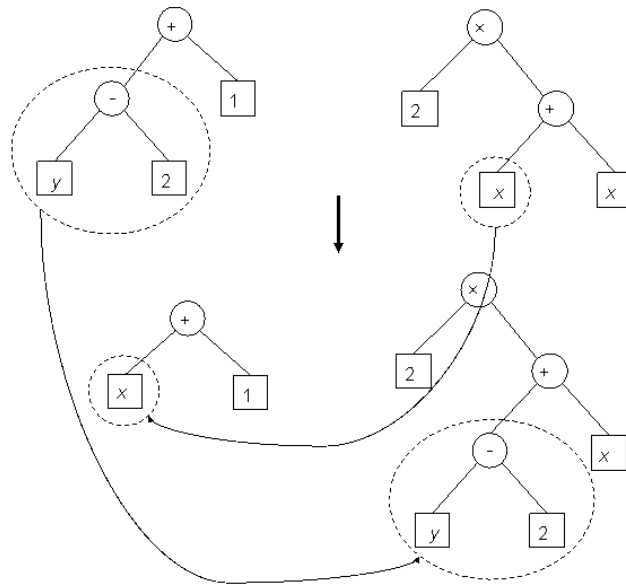
10.6. ábra. Példa program és a hozzátartozó kifejezésfa

részét alkotják. A módszer alkalmazása esetén a következő előkészítő lépések szükségesek. Defináljuk a terminálisok halmazát, mely a lehetséges terminális szimbólumokat tartalmazza, valamint a függvények halmazát a függvényekkel. Meghatározzuk az alkalmassági érték számítási módját, és beállítjuk a következő paramétereket: az egyedek számát, a szülők arányát (vagyis azt, hogy a teljes populáció mekkora része vesz részt utódok létrehozásában), a mutációs arányt, a maximális generációsszámot, illetve ehelyett a leállási feltételt.

Ezután következik az evolúciós folyamat végrehajtása, amelynek során a cél az optimális kifejezésfa megtalálása. A populációt alkotó fák különböző méretűek és alakúak lehetnek. Az első generációt véletlenszerűen létrehozott, de szintaktikailag érvényes fák alkotják.

10.2.1. Keresztezés

A keresztezés során először mindkét szülő fájában kiválasztunk egy véletlenszerű csomópontot. Ezután kicseréljük a csomópontokhoz tartozó részfat a két szülőben és így két utódot nyerünk. A következő lépésben ellenőrizzük, hogy a kapott utódok szintaktikailag érvényesek-e, hiszen előfordulhat, hogy a létrejött utód olyan fát reprezentál, amelyik az adott feladat szempontjából értelmezhetetlen. Ebben az esetben a szülőkből más csomópontokat választunk a keresztezéshez, és ezt addig ismételjük, míg olyan utódokat nem kapunk, amelyek szintaktikailag érvényesek. A 10.7. ábrán egy példán követhetjük nyomon a keresztezés folyamatát. A $\{+, -, \times, ./\}$ jelek, műveletek alkotják a lehetséges függvények halmazát, az $\{x, y, 1, 2\}$

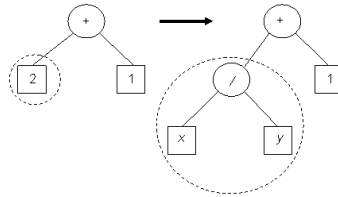


10.7. ábra. A keresztezés operáció

elemek pedig a terminálisokat. A keresztezési csomópont a baloldali szülő esetén a „-” függvény, a jobboldali szülő esetén pedig a baloldali „x”. Ezeket cseréljük ki, így kapjuk meg az alsó két ábrán látható utódokat.

10.2.2. Mutáció

A mutációban résztvevő egyeden véletlenszerűen kiválasztunk egy csomópontot. Töröljük a csomópont alatt lévő részfát, és egy újat növesztünk helyette. A szintaktikai érvényességre itt is ügyelünk. A 10.8. ábrán egy példán láthatjuk a mutációt, a terminálisok és a függvények halmaza ugyanaz, mint a keresztezéses példában. A „2”-es értékű terminális csomópontot választottuk, melynek helyére egy új részfa került.



10.8. ábra. A mutáció művelete

10.3. Bakteriális evolúciós algoritmusok

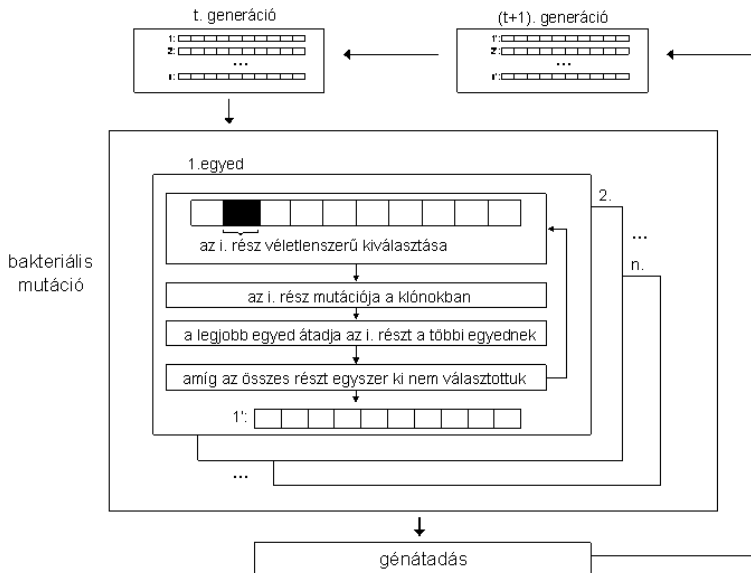
A természet sok ötletet ad különböző evolúciós módszerek alkotására a hagyományos genetikai algoritmus mintájára. Az egyik legújabb ilyen módszer a 1990-es évek második felében kifejlesztett *bakteriális evolúciós algoritmus*. Ez az eljárás a baktériumok evolúciós jelenségén alapul, a módszert a baktériumok génátadási jelensége inspirálta. Először a pszeudobakteriális genetikai algoritmust publikálták [139], mely a genetikai algoritmusokban alkalmazott mutáció helyett az úgynevezett bakteriális mutációt alkalmazza. Később megjelent a bakteriális evolúciós algoritmus [138], mely a bakteriális mutáción kívül a génátadási (géntranszfer) operációt is magába foglalja. Az algoritmus folyamata a következő:

```

kezdeti populáció létrehozása
generáció=0
amíg generáció < max. generáció
{
    bakteriális mutáció alkalmazása minden egyedre
    génátadás
    generáció = generáció + 1
}

```

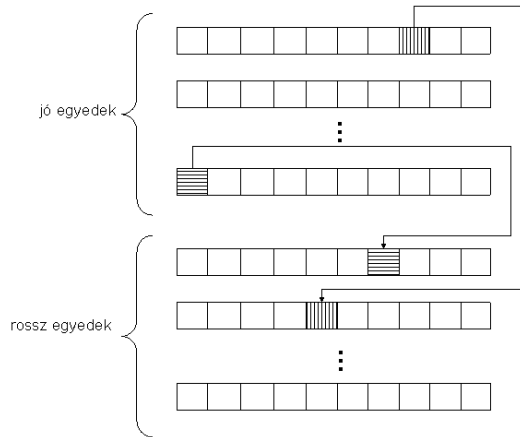
A genetikai algoritmusokban megismert szelekcióra nincs szükség, és az operátorok is eltérőek. Az egyedek kromoszómába vannak kódolva, valós számokkal. A kezdeti populáció kialakítása a genetikai algoritmusokhoz hasonlóan történik, a keresési térből N_{egyed} pont véletlenszerű kiválasztásával. Ezután következik az evolúciós ciklus, melyben a bakteriális mutációt minden egyedre alkalmazzuk, a génátadást pedig a populáción hajtjuk végre.



10.9. ábra. Bakteriális evolúciós algoritmus

10.3.1. Bakteriális mutáció

A *bakteriális mutáció* egy egyeden végrehajtott operátor, melyet azonban minden egyedre végrehajtunk. Az eljárás elején az egyedet lemásoljuk $N_{klón}$ példányban (klónok). A kromoszóma egy véletlenszerűen kiválasztott i . részét megváltoztatjuk a klónokban (mutáció), az eredeti baktériumban viszont nem. Utána kiválasztjuk a legjobb egyedet a klónok és az eredeti egyed közül, és az a kromoszómájának az i . részét átadja a többi egyednek. Ez azt jelenti, hogy a többi egyed kromoszómájának i . részét helyettesítjük a legjobb egyed kromoszómájának i . részével. Ezt a folyamatot, mely a mutáció – kiértékelés – kiválasztás – behelyettesítés lépéssorozatot jelenti, ismétljük addig, amíg a kromoszómának mindegyik részét egyszer ki nem választottuk. Amikor az egész kromoszómával végeztünk, kiválasztjuk a legjobb egyedet, a többi $N_{klón}$ egyedet pedig megszüntetjük. A folyamat végén tehát az eredeti egyednél jobb, vagy legrosszabb esetben, sikertelen mutációs ciklusok esetén, azzal megegyező egyedet kaptunk. A bakteriális mutáció a teljes algoritmussal együtt a 10.9. ábrán látható.



10.10. ábra. A génátadás művelete

10.3.2. Génátadás

A *génátadási* (*géntranszfer*) operátor segítségével valósul meg az információcsere az egyes egyedek között a baktériumpopulációban. A bakteriális mutáció az egyes baktériumokat külön-külön optimalizálja, szükség van azonban az egyedek közötti információátadásra, a genetikai információ terjesztésére a populációban. Első lépésben a populációt két részre osztjuk, az egyik felébe kerülnek a jó egyedek, a másikba pedig a rosszak. Nincs szükség a hagyományos értelemben vett alkalmassági érték számításra, elég ha a baktériumokat kiértékeljük az adott feladatban alkalmazott kiértékelési kritérium szerint. Így a baktériumok összehasonlíthatók, a két fél populáció könnyen létrehozható. Második lépésben kiválasztunk egy baktériumot a jó egyedek közül, ez lesz a forrásbaktérium, egyet pedig a rossz egyedek közül, ez lesz a célbaktérium. A harmadik lépésben kiválasztunk egy részt a forrásbaktérium kromoszómájából, és átadjuk a célbaktériumnak. A célbaktérium ezzel felülírhatja a kromoszómájának egy részét, vagy egyszerűen hozzáadhatja a kromoszómájához, ha különböző hosszúságú egyedeket is megengedünk. Ezt a három lépésből álló eljárást ismételjük N_{inf} -szer, ahol N_{inf} egy paramétere az algoritmusnak, és az „infekciók” számát jelöli. A génátadás illusztrációja a 10.10. ábrán látható.

10.3.3. Összehasonlítás, paraméterek

A bakteriális evolúciós algoritmusokban a korábban megismert szelekció, keresztezés, és mutáció operátorokat a génátadás, bakteriális mutáció és osztódás váltja fel. A módszert a baktériumpopulációk génátadási jelensége (transzdukció) inspirálta. Ennek segítségével egy baktérium gyorsan szét tudja terjeszteni a genetikus információt. Az egyik jelenség, amikor egy baktérium a kromoszómájának egy részét átadja a saját klónjainak (ez történik a bakteriális mutációnál a legjobb egyeddel), a másik pedig, amikor egy populációban az egyik baktérium egy másiknak ad át genetikus információt (génátadás). Ez utóbbi művelet helyettesíti a hagyományos genetikus algoritmus keresztezés operátorát, megengedve az információ átadását különböző egyedek között, még jobb egyedeket létrehozva ezáltal. Az algoritmusnak 4 paramétere van: a baktériumok száma (N_{egyed}), a generációk száma (N_{gen}), a klónok száma ($N_{klón}$), és az infekciók száma (N_{inf}). A maximális generációk száma helyett ennél a módszernél is használható más leállási feltétel. A bakteriális megközelítés a benne alkalmazott operátorok természete miatt a genetikus algoritmusoknál kedvezőbb konvergencia viselkedéssel rendelkezik. Kevesebb generáció elegendő a kvázioptimális megoldás eléréséhez. Ezenkívül a bakteriális mutációban alkalmazott klónok miatt elegendő kevesebb egyedet alkalmazni a populációban.

10.4. Egyéb módszerek

Az előbbieken említett módszereken kívül sok egyéb evolúciós és élőlényeket, élőlénypopulációk viselkedését utánzó módszert fejlesztettek ki. Már a téma korai időszakában megjelentek a J. Holland-féle módszernek alternatívái, az evolúciós programozás [58], és az evolúciós stratégiák [4, 148]. Később javasolták még például a többkritériumú genetikus algoritmusokat [59, 60], a többpopulációs genetikus algoritmust [146, 214], az ún. hangykolóniákat [45, 46], a vírus alapú evolúciós algoritmust [122], a „ragadozózsákmány” módszert [3], a méhkirálynő evolúciós algoritmust [81], a mesterséges immunrendszereket [37, 68], és a részecske-sereg optimalizációt [52, 85]. Megemlítjük ezenkívül a memetikus algoritmusokat is [133]. Ezek az evolúciós módszereket kombinálják lokális szélsőérték-kereső eljárásokkal. Ezek az algoritmusok az evolúciós technikák kvázi-optimumkeresését és lassú konvergencia tulajdonságait küszöbölik ki, gyorsabb konvergenciát és pontosabb optimum megtalálását teszik lehetővé.

11. fejezet

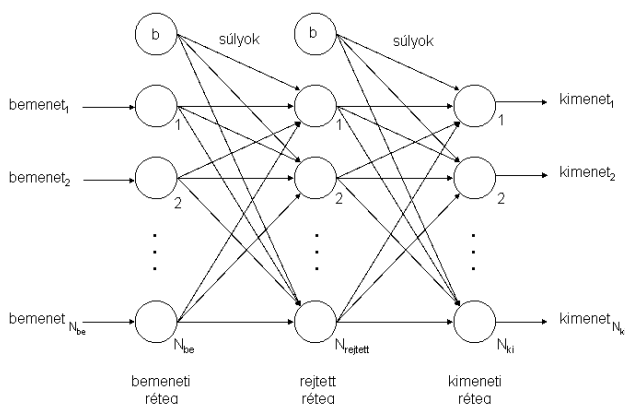
Neurális hálózatok

A számítási intelligencia harmadik fontos csoportját a neurális hálózatok alkotják, melyek az agyban található neuronok hálózatában végbemenő folyamatok utánzását megvalósító számítási rendszerek. Számos probléma létezik, melyet nem tudunk megoldani hagyományos algoritmusokkal, de az ember könnyedén megoldja őket. Az agy tevékenységeinek megfigyelésével utánozhatjuk annak működését, és ezek alapján számítási rendszereket hozhatunk létre. A mesterséges neurális hálózat tehát az evolúciós technikákhoz hasonlóan a biológiából elerett módszer. A neurális hálózat elemi egységekből, neuronokból áll, melyek valamilyen lokális feldolgozást hajtanak végre. Ezek a neuronok valamilyen topológia szerint vannak összekapcsolva egy hálózattá. A neurális hálózat minták alapján képes tanulni, a megtanult tudás a neuronok közötti összeköttetésekben, úgynevezett súlyokban tárolódik. A minták alapján végbemenő tanulás alternatívát jelent a fuzzy rendszerek esetén alkalmazott szabályokra, a neurális hálózatok és a fuzzy rendszerek jól kiegészítik egymást.

Ebben a fejezetben a neurális hálózatok alapjait tekintjük át, előrecsatolt típusú hálózatokkal foglalkozunk. A következőkben röviden bemutatjuk a két legelterjedtebb neurális hálózattípust, az MLP és az RBF hálózatokat, majd kicsit részletesebben ismertetjük a B-spline neurális hálózatokat. A hálózatok bemutatása után a tanuló algoritmusokat tárgyaljuk. A klasszikus backpropagation módszer ismertetése után bemutatjuk a Levenberg-Marquardt tanuló algoritmust.

11.1. Többrétegű perceptron

Annak ellenére, hogy a neurális hálózatoknak számos fajtája ismert, az egyik leggyakrabban használt típus a *többrétegű perceptron* (*Multi-layer perceptron*, *MLP*). Az MLP egy előrecsatolt hálózat, mely számos egységet (neuron) tartalmaz, melyek súlyozott összeköttetésekkel kapcsolódnak egymáshoz. A neuronok rétegekbe vannak rendezve, egy rétegbe a hasonló kapcsolatokkal rendelkező neuronok tartoznak. Háromféle réteget

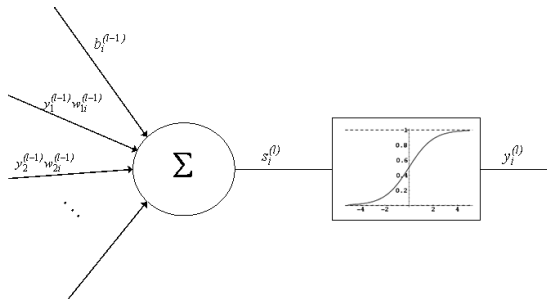


11.1. ábra. Többrétegű perceptron (MLP)

különböztetünk meg, a bemeneti, rejtett és kimeneti réteget. A bemeneti réteg a kívülről érkező bemeneti vektort fogadja, és adja át a súlyozott kapcsolatokon keresztül az első rejtett rétegnek. Az itt található neuronok feldolgozzák a kapott értékeket, majd a következő rétegnek továbbítják. Több rejtett réteg is előfordulhat a hálózatban. Végül a kimeneti rétegen keresztül az információ a környezet felé továbbítódik. Egy egy rejtett réteget tartalmazó MLP látható a 11.1. ábrán. Távlatból nézve tulajdonképpen egy tetszőleges bemeneti vektort terjesztünk előre felé a hálózaton, mely a kimeneti rétegen valamilyen aktiválást okoz. A teljes hálózati függvény, ami a bemeneti vektort a kimeneti vektorba képzí, a hálóban található súlyokkal határozható meg. A hálóban lévő minden neuron egy egyszerű processzáló elem, ami kiszámítja saját kimenetét a gerjesztésének alapján. Az l . réteg i . neuronjának bemenete (11.2):

$$s_i^{(l)} = \sum_{j \in \text{pred}(i)} y_j^{(l-1)} w_{ji}^{(l-1)} + b_i^{(l-1)},$$

ahol $\text{pred}(i)$ az i . egységet megelőző neuronok halmaza, azaz azon neuronoké, ahonnan összeköttetés érkezik a vizsgált i . neuronba; w_{ji} a j . és az i . egység közötti összeköttetés súlyának az értéke. A b_i konstans a neuron „torzítási (bias)” értékének nevezzük (amely egy járulékos, egység kimenetű neuron minden rétegben, és szerepe, hogy a hálózat akkor is produkáljon



11.2. ábra. Egy neuron bemenetei és kimenete

kimenetet, ha nincs gerjesztés). A homogén reprezentáció kedvéért ezt az értéket gyakran egy konstans 1-es kimenetű bias-egységből induló súllyal veszik figyelembe. Ez azt jelenti, hogy a bias értékek súlyként kezelhetők. A 11.1. ábrán b betűkkel vannak jelölve a bias egységek, melyek minden processzáló neuronhoz kapcsolódnak valamekkora súllyal. Az i . egység kimenetét úgy határozzuk meg, hogy az előbb számított $s_i^{(l)}$ eredő bemenetet egy nemlineáris aktiváló vagy más néven gerjesztési függvénynek vetjük alá. Általában a szigmoid függvényt alkalmazzák, mely alapján a neuron kimenete a következőképpen határozható meg:

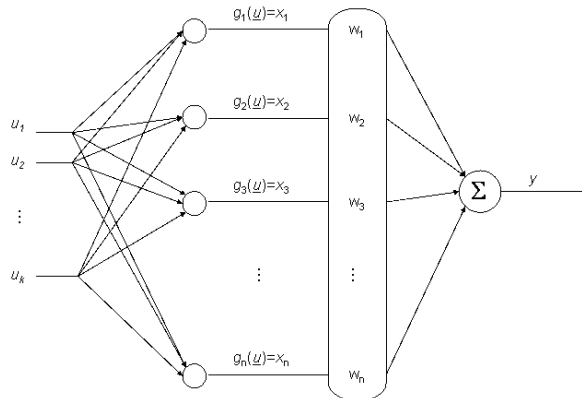
$$y_i^{(l)} = \frac{1}{1 + e^{-s_i^{(l)}}}.$$

Egy érdekes tulajdonsága ennek a függvénynek, hogy a deriváltja könnyen számítható:

$$\frac{\partial y_i}{\partial s_i} = y_i(1 - y_i).$$

A hálózat első rétegében minden egyes neuron a maga szigmoid függvényével, súlytényezőivel és eltolásával az n dimenziós teret egy $n - 1$ dimenziós hipersíkkal egyértelműen két térfélre képes osztani. A következő réteg neuronjai e hipersíkokból konstruálhatnak konvex tartományokat, a rákövetkező réteg neuronjai e konvex tartományokból gyárthatnak konkáv tartományokat, végül az utolsó réteg súlyai e tartományokon való leképezések értékeit állíthatják be, így az értelmesen felhasználható rétegek száma elméletileg véges, és az MLP méretezésének problémája többnyire a rétegekben alkalmazandó neuronok számától függ.

Minták alapján beállíthatjuk a súlyokat valamilyen tanuló algoritmussal,



11.3. ábra. Radiális bázisfüggvény hálózat (RBF)

hogyan a hálózat a kívánt leképezést valósítsa meg. A tanuló algoritmusokat a fejezet későbbi részében ismertetjük.

11.2. Radiális bázisfüggvény hálózatok

A neurális hálózatok másik elterjedt fajtája a *radiális bázisfüggvény (Radial Basis Function, RBF)* hálózat, melynek felépítését a 11.3. ábrán láthatjuk. A bemenet utáni első rétegbeli neuronok aktiváló függvényei, az ábrán $g_i(\underline{u})$ -val jelölt függvények körszimmetrikusak. Aktiváló függvényként leggyakrabban a Gauss-függvényeket alkalmazzák, melyek a következő alakban írhatók fel:

$$g_i(\underline{u}) = e^{-\frac{\|\underline{u} - \underline{c}_i\|^2}{2\sigma_i^2}}.$$

A c_i érték az i . függvény középpontját adja meg, mely egy paramétere a függvénynek, a σ_i érték pedig az i . függvény szélességparamétere. A hálózat kimenete az egyes Gauss-függvények súlyokkal történő lineáris kombinációjával számítható:

$$y = \sum_{i=1}^n w_i g_i(\underline{u}).$$

A paramétereket tanítással állítjuk be.

A hálózat működését azzal a szemléletes hasonlattel képzelhetjük el, hogy az általában „nem különösebben sima” módon közelíthet egy nagyobb, sima felületet ugyanúgy, mintha egy sík területet valami ponyvával szeretnénk lefedni oly módon, hogy azt néhány sátorbottal néhány diszkrét pontban kitámasztjuk. A botok magassága felel meg a függvény adott környéken való közelítő értékének, a bázisfüggvények szélessége a ponyva lehajlási szélességének. Bár ezeknek a Gauss-függvényeknek a súlyozott összegű kimenete „matematikai értelemben” ugyan „sima”, azaz akárhányszor deriválható, az elvégzendő függvényközelítést illetően gyakorlatilag azonban nem az, túlságosan „fodros”. E fodrosság elkerülhető, ha Gauss-függvények helyett egymáshoz simán illeszthető szakaszokból álló, a következő alfejezetben ismertetendő B-spline függvényeket alkalmazunk nagyjából egyenletesen elosztott csomópontokkal teljesen hasonló módon.

Az MLP és RBF hálózatokról és egyéb neurális hálózatokról további részletek találhatók pl. a [71, 215] könyvekben.

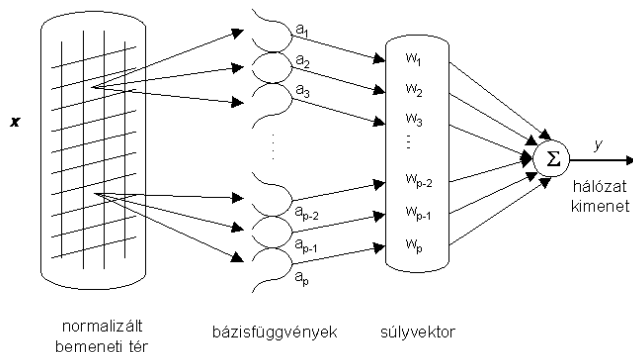
11.3. B-spline neurális hálózatok

A *B-spline típusú neurális hálózatok* számos előnyös tulajdonságot mutatnak az MLP és az RBF típusú hálózatokhoz képest. Az információt lokálisan tárolják, ami azt eredményezi, hogy a bemeneti tér egy bizonyos részén végrehajtott tanítás a tér többi részét csak kismértékben érinti. Emiatt jól használhatók on-line adaptív modellezési és irányítási alkalmazásokban [203]. A rács-alapú felépítésük transzparenssé teszi őket, így más típusú hálózatokkal ellentétben a tárolt tudásuk könnyebben megérthető [26]. Sokdimenziós problémákra jobb az általánosító képességük mint néhány RBF hálózatnak [19].

A B-spline neurális hálózatok a rács-alapú asszociatív memória hálózatok (AMN) osztályába tartoznak. Ezek a hálózatok három réteget tartalmaznak: a normalizált bemeneti tér réteget, a bázisfüggvények rétegét és a súlyvektor réteget. A hálózat felépítése a 11.4. ábrán látható.

11.3.1. Normalizált bemeneti tér réteg

Ez a réteg egy rácsozat, amelyen a bázisfüggvényeket definiáljuk. Ahhoz, hogy egy rácsot definiáljunk a bemeneti téren, minden egyes tengelyhez csomópontok egy vektorát kell meghatározni. Általában minden



11.4. ábra. B-spline neurális hálózat

dimenzióban más és más a csomópontok száma, és más-más pozíciókban is helyezkednek el.

Jelöljük az i . tengely belső csomópontjait $\lambda_{i,j}, j = 1, \dots, r_i$ -vel melyekre érvényes, hogy $x_i^{\min} < \lambda_{i,1} \leq \lambda_{i,2} \leq \dots \leq \lambda_{i,r_i} < x_i^{\max}$, ahol $x_i^{\min}$ és $x_i^{\max}$ az i . bemenet minimum ill. maximum értéke. Minden tengely szélén szükséges külső csomópontokat is definiálni, melyekre teljesül, hogy $\lambda_{i,-(k_i-1)} \leq \dots \leq \lambda_{i,0} = x_i^{\min}$ és $x_i^{\max} = \lambda_{i,r_i+1} \leq \dots \leq \lambda_{i,r_i+k_i}$. Ezek a külső csomópontok azért szükségesek, hogy azokat a bázisfüggvényeket is definiálni tudjuk, amelyek közel vannak a határokhoz. A külső csomópontok általában a bemeneti tengely szélével egybeesnek, vagy az adott tengely szélén kívül egymástól egyenlő távolságra helyezkednek el. A hálózat bemeneti tere $[x_1^{\min}, x_1^{\max}] \times \dots \times [x_n^{\min}, x_n^{\max}]$, ezért a külső csomópontok csak a bázisfüggvények definiálása céljából szükségesek. Az i . bemenet j . intervallumát $I_{i,j}$ -vel jelöljük:

$$I_{i,j} = \begin{cases} [\lambda_{i,j-1}, \lambda_{i,j}] & \text{ha } j = 1, \dots, r_i \\ [\lambda_{i,j-1}, \lambda_{i,j}] & \text{ha } j = r_i + 1. \end{cases}$$

Az i . tengelyen $r_i + 1$ intervallum van (melyek között lehetnek üresek, ha bizonyos csomópontok egybeesnek), azaz összesen $p' = \prod_{i=1}^n (r_i + 1)$ n -dimenziós cella van a rácsozaton.

11.3.2. A bázisfüggvények rétege

A bázisfüggvények rétege p darab B-spline bázisfüggvényt tartalmaz, amelyeket az n -dimenziós rácsozaton értelmezünk. A B-spline függvények könnyen állíthatók, számíthatók és implementálhatók. A bázisfüggvények alakja, mérete és eloszlása az adott AMN-re jellemző, és a bázisfüggvények tartója behatárolt. A B-spline neurális hálózatokban a spline rendje automatikusan meghatározza a függvény tartóját és alakját. A k rendű egyváltozós B-spline bázisfüggvény tartója k intervallum széles. Ezért minden bemenet k darab bázisfüggvényt aktivál. A j . egyváltozós k rendű bázisfüggvényt $N_k^j(x)$ -vel jelöljük, és a következőképpen adhatjuk meg:

$$N_k^j(x) = \left(\frac{x - \lambda_{j-k}}{\lambda_{j-1} - \lambda_{j-k}} \right) N_{k-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-k+1}} \right) N_{k-1}^j(x)$$

$$N_1^j(x) = \begin{cases} 1, & \text{ha } x \in I_j \\ 0, & \text{egyébként.} \end{cases}$$

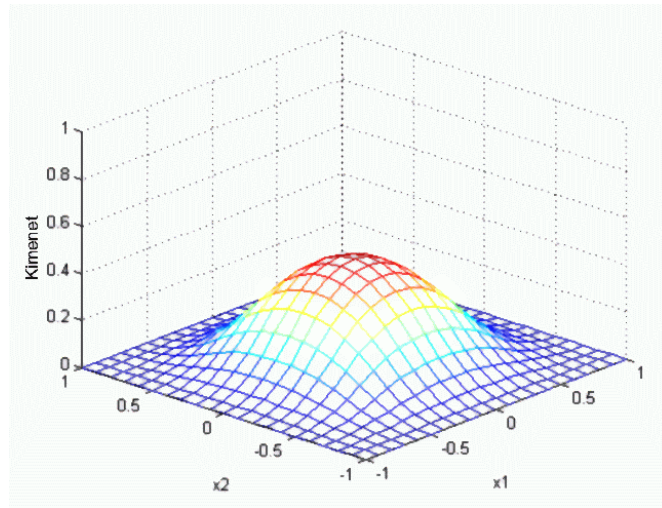
Többsváltozós bázisfüggvényeket az egyváltozósak tenzor szorzatával nyerünk, azaz: $N_{\mathbf{k}}^j(\mathbf{x}) = \prod_{i=1}^n N_{\mathbf{k}_i, i}^j(\mathbf{x}_i)$ (ld. 11.5. ábra). A k_i rendű bázisfüggvények száma az i . tengelyen r_i belső csomópont esetén $r_i + k_i$. Ezért a többsváltozós B-spline-ok száma $p = \prod_{i=1}^n (r_i + k_i)$. Ez a szám exponenciálisan függ a bemenet méretétől, ezért a B-spline-ok csak alacsony dimenziószámú problémákra használhatók.

11.3.3. A súlyvektor réteg

A hálózat kimenete a bázisfüggvények kimeneteinek lineáris kombinációja. A kombinációs tagok az állítható súlyparaméterek: $y = \sum_{i=1}^p \mathbf{a}_i \mathbf{w}_i = \mathbf{a}^T \mathbf{w}$, ahol $\mathbf{a}_i = N_{\mathbf{k}}^i(x)$, $i = 1, \dots, p$. Mivel csak $p'' = \prod_{i=1}^n k_i$ cella aktív egyszerre, ezért a kimenet számítása a következő formára egyszerűsödik: $y = \sum_{i=1}^{p''} \mathbf{a}_{act(i)}(\mathbf{x}) \mathbf{w}_{act(i)}$, ahol $\mathbf{a}_{act(i)}(\mathbf{x})$ jelöli az i . aktív bázisfüggvényt $\mathbf{x}$ bemeneti vektor esetén.

11.3.4. Almodulok

A magas dimenziójú esetekben keletkező problémák leküzdésére a nagy-dimenzió számú modell helyett, amely az összes bemeneti változót tartalmazza, kisebb dimenzió számú almodellek segítségével próbáljuk az adott feladatot megoldani. Egy ilyen módon összeállított hálózat kimenete:



11.5. ábra. Kétváltozós B-spline függvény

$y(\mathbf{x}) = \sum_{u=1}^{n_u} S_u(\mathbf{x}_u)$, ahol $S_i(\mathbf{x}_i)$ jelöli az i . almodellt, és $\mathbf{x}_i$ azon bemeneti változók halmaza, amelyek az i . almodellt alkotják.

11.3.5. B-spline neurális hálózatok tervezése

A B-spline hálózatok tervezése az alábbi fázisokat foglalja magába:

1. Az almodellek meghatározása
2. Mindegyik almodellre annak bemeneteinek meghatározása
3. A spline-ok rendjének meghatározása minden bemenetre
4. A belső csomópontok számának meghatározása mindegyik bemenetre
5. A belső csomópontok helyének meghatározása mindegyik bemenetre
6. A súlyok meghatározása

Az utolsó két fázis a később részletezendő tanuló algoritmusokkal megoldható [26, 151]. Az első négy tervezési fázis összetettebb probléma, melynek megoldására más módszereket fogunk alkalmazni.

11.4. Más típusú neurális hálózatok

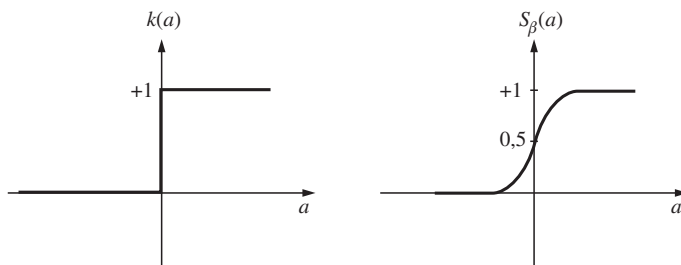
Az előrecsatolt neurális hálózatokon kívül számos más típusú hálózat is elterjedt az alkalmazásokban. Fontos kategóriát alkotnak a visszacsatolt hálózatok, melynek egyik fontos típusa az Elman-hálózat [54]. Az Elman-hálózatban előforduló belső visszacsatolások nemcsak arra alkalmasak, hogy valamiféle „dinamikát” vigyenek be a hálózat által megvalósított leképezésbe, ami pl. belsőégésű motorok modellezésére kiváltképp alkalmassá tette ezt a hálózat-típust, hanem a legújabb kutatások szerint arra is, hogy azok az így keletkező „memória” révén mintegy megszűrjék, megsimítsák a bemeneti tanító adatokat, így a közönséges MLP-nél általában jobbak (pontosabbak) függvénymodellezési célokra, pl. időtől függő adat-sorok analízisében is.

11.5. Neurofuzzy irányítási rendszerek

*Adaptív neurális hálózat*nak nevezik az olyan hálózatokat, amelyeknek értéke a csomópontokhoz (vagy egy részhalmazukhoz) rendelt paraméterek értékétől függ. Ekkor a tanulási folyamat során a hibát ezen paraméterértékek állításával lehet minimalizálni, az összeköttetések csak a csomópontok közti információáramlás irányát jelzik, súllyal nem rendelkeznek. Adaptív neurális hálózatok alapvető tanulási algoritmus a gradiens módszeren és a láncszabályon alapul [194]. Mivel a gradiens módszer jellemzően lassú konvergenciát biztosít és gyakran csak lokális minimumot talál meg, ezért abban az esetben, ha a kimenet a paraméterhalmaz egy részhalmazától lineárisan függ, ezen értékek optimalizálására a legkisebb négyzetek módszere is használható. Ekkor az egész hálózat optimalizálását hibrid tanulási algoritmussal [77, 78] végzik, amely a gradiens és a legkisebb négyzetek módszerének ötvözte oly módon, hogy a lineáris paramétereket a legkisebb négyzetek módszerével, a nemlineárisokat pedig a gradiens módszerrel optimalizálják. A fuzzy irányítót megvalósító neurális hálózat esetén — mint látni fogjuk — teljesül a hibrid tanulási algoritmus használatának feltétele.

Az adaptív hálózat egy *többrétegű visszacsatolatlan (vagy előrecsatolt) hálózat*, amelyben minden neuron a bemeneti értékein és a neuronhoz tartozó paramétereken hajtja végre hozzátartozó *aktiváló* vagy *gerjesztési függvényt*.

Az aktiváló függvények csomópontonként különbözhetnek, egyetlen megkötés, hogy szakaszosan differenciálhatók legyenek. A gyakorlatban gyakran használnak küszöb vagy szigmoid függvényeket, melyeket a 11.6.



- 11.6. ábra.** Példák aktiváló függvényekre.
- (a) küszöbfüggvény: $k(a) = 1, \text{ ha } a \geq 0; 0, \text{ ha } a < 0$
- (b) szigmoid függvény: $s_\beta(a) = (1 + e^{\beta a})^{-1}$

ábra illusztrál.

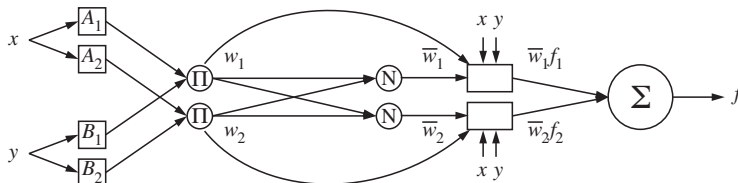
Az ANFIS (Adaptiv-Network-based Fuzzy Inference System) fuzzy irányítási rendszerrel ekvivalens adaptív neurális hálózatot valósít meg [79]. Az egyszerűség kedvéért két bemenettel (x_1 és x_2) és egy kimenettel (y) rendelkező, valamint két TAKAGI–SUGENO-típusú [178] (7.5. szakasz) szabályt tartalmazó irányítási rendszerrel azonos neurális hálózat felépítését ismertetjük.

Legyen a két szabály

$$R_1 : \text{ Ha } x_1 = A_1 \text{ és } x_2 = B_1 \text{ akkor } y = f_1 = p_1 x + q_1 y + r_1$$

$$R_2 : \text{ Ha } x_1 = A_2 \text{ és } x_2 = B_2 \text{ akkor } y = f_2 = p_2 x + q_2 y + r_2$$

formában megadva.



11.7. ábra. Két bemenetű, két szabályt tartalmazó TAKAGI–SUGENO irányítóval ekvivalens ANFIS struktúra

A TAKAGI–SUGENO-irányítóval ekvivalens adaptív neurális hálózatot a 11.7. ábra szemlélteti. A négyzetek az állítható paraméterekkel rendelkező, a körök a paraméter nélküli csomópontokat jelölik. Az egyes rétegekben hasonló függvények vannak.

- *Első réteg:* Minden csomópont parametrizált az

$$O_i^1 = A_j(x_1)$$

($i = 1,2$; $j = 1,2$), illetve

$$O_i^1 = B_j(x_2),$$

($i = 3,4$; $j = 1,2$) aktiváló függvénnyel, ahol i a csomópont száma. Más szóval O_i^1 az A_j és a B_j tagsági függvényeket valószínűsíti meg, és meghatározza az adott x_1 és x_2 bemeneti értékek illeszkedési mértékét. Ha haranggörbe alakú tagsági függvényeket használunk, akkor a tagsági függvényeket

$$A_i = \frac{1}{1 + \left[\left(\frac{x_1 - c_i}{a_i} \right)^2 \right]^{b_i}}$$

vagy

$$A_i = \exp \left\{ - \left(\frac{x_1 - c_i}{a_i} \right)^2 \right\}$$

alakban definiálhatjuk, ahol $\{a_i, b_i, c_i\}$ a paraméterhalmaz, amelyeket *bemeneti paramétereknek* nevezünk. Hasonló módon a másik bemenet (x_2) tagsági függvényei is megadhatók. A tagsági függvény alakjának megváltozását a paraméterek megfelelő módosításával érhetjük el. Más — például szakaszosan lineáris, trapéz, vagy háromszög alakú — tagsági függvényeket is alkalmazhatunk, amelyek eleget tesznek a szakaszonként differenciálhatóság feltételének.

- *Második réteg:* A csomópontokhoz nem tartozik paraméter, a kimeneten a bejövő jelek szorzatát továbbítják. Például:

$$w_i = A_i \cdot B_i, \quad i = 1,2.$$

A csomópontok a szabályok illeszkedési mértékét, tüzelési értékét számítják ki. Az algebrai metszet helyett tetszőleges más t-norma is alkalmazható.

- *Harmadik réteg:* Szintén paraméterhalmaz nélküli csomópontokat tartalmaz, melyek az i -edik szabály és az összes szabály tüzelési értékének arányát, vagyis a *normalizált tüzelési (vagy illeszkedési) értéket* határozzák meg:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1,2.$$

- *Negyedik réteg:* Parametrizált csomópontokat tartalmaz, amelyek a

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i)$$

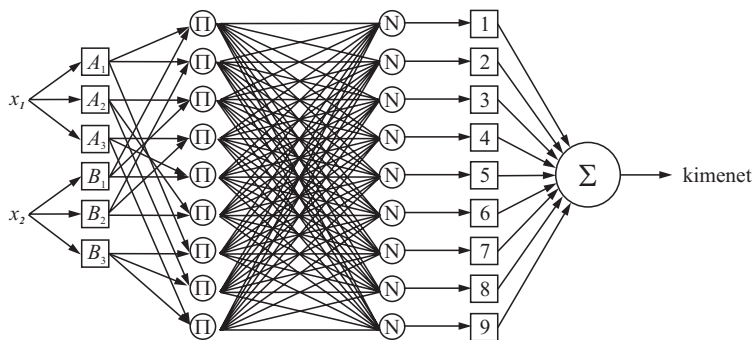
aktiváló függvényt valósítják meg, ahol $\{p_i, q_i, r_i\}$ ($i = 1, 2$) a csomópontokhoz tartozó *kimeneti paraméterhalmaz*.

- *Ötödik réteg:* Egyetlen paraméter nélküli csomópontot tartalmaz, amely a végeredményt számolja ki:

$$O_1^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Az így konstruált adaptív neurális hálózat funkcionálisan ekvivalens a TAKAGI–SUGENO-típusú következtetési rendszerrel. A negyedik réteg megfelelő módosításával SUGENO-irányítót is megvalósíthatunk. Diszkrét defuzzifikációs módszer alkalmazása esetén MAMDANI-típusú irányítás is helyettesíthető adekvát ANFIS-struktúrával.

A bemeneti alaphalmazok finomabb particionálása (azaz magasabb szabálysám) esetén a csomópontok száma a szabályokéval exponenciálisan nő. Ha például bemenetenként három nyelvi változót definiálunk, akkor a szabályok száma kilencre nő, így a második, harmadik és negyedik rétegben is kilenc csomópont szerepel (ld. 11.8. ábra).



11.8. ábra. Két bemenetű, kilenc szabályt tartalmazó TAKAGI–SUGENO irányítást megvalósító ANFIS struktúra

Vegyük észre, hogy rögzített bemeneti paraméterek esetén a végeredmény a konzekvens paraméterek lineáris kombinációjaként írható fel:

$$\begin{aligned} y &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ &= (\bar{w}_1 x_1) p_1 + (\bar{w}_1 x_2) q_1 + (\bar{w}_1) r_1 \\ &\quad + (\bar{w}_2 x_1) p_1 + (\bar{w}_2 x_2) q_1 + (\bar{w}_2) r_1 \end{aligned} \quad (11.1)$$

Legyen S_1 a bemeneti, S_2 pedig a kimeneti paraméterhalmaz. (11.1) miatt teljesül a hibrid tanulási algoritmus feltétele, így az közvetlenül alkalmazható [77, 78]. Az S_1 halmaz paramétereit gradiens módszerrel, az S_2 halmaz paramétereit pedig a legkisebb négyzetek módszerével optimalizálhatjuk.

Az eredmények azt mutatják [79], hogy a fuzzy és neurális technikát vegyesen alkalmazó rendszer hatékonyabban működik az egyik technikát kizárólagosan alkalmazóhoz képest. A fuzzy szabályok segítségével ugyanis a kiinduló hálózatba is kódolható problémafüggő információ — ezeket az értékeket csak neurális technika alkalmazása esetén véletlenszerűen generálják —, a szabályok paramétereinek beállítása pedig a különösen hatékony hibrid tanulási módszerrel igen gyors konvergenciát eredményez.

11.6. Backpropagation eljárás

A neurális hálózat paramétereit, azaz a súlyokat úgy szeretnénk beállítani, hogy a hálózat megvalósítsa a bemenet és kimenet közötti kívánt leképezést. Ezt a súlybeállítási folyamatot nevezzük a hálózat „tanításának”. A tanításhoz szükséges leképezés minták formájában adott. A mintakészlet minden egyes mintája egy-egy bemenet-kimenet párt határoz meg, amelyek a hálózat választ írnak le az adott bemeneti gerjesztésre. A tanítás célja ennek megfelelően az, hogy a hálózat súlyait úgy állítsuk be, hogy a hálózat minden egyes minta bemenetre olyan kimenetet adjon, mint a mintához tartozó kívánt kimenet, minél kisebb legyen a hálózat által számított kimenet és a kívánt kimenet közötti eltérés. A hálózat által meghatározott kimenet és a kívánt kimenet közötti eltérés matematikailag a következő formában írható fel:

$$E = \frac{1}{2} \sum_{p=1}^m \sum_{n=1}^k (y_n^{(p)} - t_n^{(p)})^2.$$

Ebben a hibadefinícióban $y_n^{(p)}$ jelöli a hálózat által a p . mintára számított kimenet n . komponensét, $t_n^{(p)}$ a p . mintához tartozó megkívánt kimenet n . komponensét, k a kimenet dimenziószáma, (azaz a neuronok száma a kimeneti rétegben), m pedig a minták száma. Egy neuront tartalmazó kimeneti réteg esetén a hiba vektoros formában a következőképpen írható fel egyszerűbb formában:

$$E = \frac{\|\underline{y} - \underline{t}\|^2}{2}.$$

Itt y az m -dimenziós kimeneti vektor, azaz a hálózat kimeneteinek összege az m mintára, t az m -dimenziós megkívánt kimeneti vektor, $\|\cdot\|$ az euklideszi norma. Ez a kifejezés a hibák négyzetösszege (Sum of Square Errors, SSE). A tanítás célja ennek alapján az E (kvázi-)minimumának a megkeresése. A legkorábbi tanuló eljárás az úgynevezett *backpropagation* algoritmus [152, 195], amely egy legmeredekebb lejtő típusú módszer. Az algoritmus a hiba deriváltjait felhasználva iteratíván állítja be a hálózat súlyait, a deriváltak alapján a súlyokat az optimum felé terelve. A súlyokat w -vel jelölve, az algoritmus minden iterációja a következő általános alakba írható:

$$\underline{w}[k+1] = \underline{w}[k] - \eta g[k],$$

ahol k ill. $k+1$ a lépésszám azonosítói, η a tanítás paramétere $\underline{g}$ a hiba gradiens vektora:

$$\underline{g}[k] = \frac{\partial E(\underline{w}[k])}{\partial \underline{w}^T[k]}.$$

A gradiens meghatározása után a súlyok új értéke számítható. Ezt az iteratív lépést ismételjük amíg az E hiba megfelelően kicsi nem lesz.

11.7. Levenberg-Marquardt algoritmus

A backpropagation algoritmus elsőrendű gradiens típusú módszer, mely a hiba elsőrendű deriváltjait használja. A gyakorlati feladatoknál nem garantált a konvergencia, de ha van is általában lassú. Emiatt a probléma miatt később más tanuló algoritmusok is megjelentek a közleményekben, ezekkel hatékonyabban lehet a neurális hálózatokat tanítani [149, 151]. Az egyik leghatékonyabb módszer a Levenberg-Marquardt eljárás, melyet eredetileg Levenberg [126] és Marquardt [132] nemlineáris paraméterek legkisebb négyzetes becslésére javasolt. A módszer kiválóan alkalmazható neurális hálózat súlyainak beállítására és egyéb optimalizációs problémákra.

Az algoritmus minden lépésében szükséges a *Jacobi-mátrix* kiszámítása, mely a hálózat különböző mintákra számított kimenetének súlyok szerinti parciális deriváltjait tartalmazza:

$$\mathbf{J} = \frac{\partial y(\underline{x}^p)}{\partial \underline{w}^T}.$$

A súlyok értékének $\underline{w}[k+1] = \underline{w}[k] + \underline{s}[k]$ módon történő meghatározásához az $\underline{s}[k]$ vektort a következő egyenlet megoldása adja:

$$(\mathbf{J}^T[k]\mathbf{J}[k] + \alpha\mathbf{I})\underline{s}[k] = -\mathbf{J}^T[k]\underline{e}[k], \quad (11.2)$$

ahol $\underline{e}[k]$ a k . lépésben vett hibavektor, azaz $\underline{e}[k] = y[k] - \underline{t}[k]$.

A 11.2. összefüggésben α regularizációs paraméter, amely egyaránt szabályozza a keresés irányát és a módosítás nagyságát. A keresési irány a Gauss-Newton és a legmeredekebb irány között α értékétől függően változik. Ha $\alpha \rightarrow 0$, akkor az algoritmus a Gauss-Newton módszerhez konvergál, ha $\alpha \rightarrow \infty$, akkor pedig a legmeredekebb lejtő típusú megközelítést adja. Az α paraméter kulcsfontosságú az algoritmusban, szükségessége matematikailag több megközelítésből is igazolható. Levenberg eredetileg [126] azzal indokolta az α paraméter szükségességét, hogy segítségével megfelelő keretek között tarthatók az optimalizálandó paraméterértékek megváltozásai, helyes approximációt biztosítva ezáltal. Az α paraméterrel kapcsolatos kérdéskör azonban megközelíthető a Hesse-mátrixhoz kapcsolódó megfontolásokból és a Tyihonov regularizációval [179] kapcsolatosan is.

A 11.2. egyenlet a következő formára egyszerűsíthető:

$$\underline{s}[k] = - \left[\begin{array}{c} \mathbf{J}[k] \\ \sqrt{\alpha} \mathbf{I} \end{array} \right]^+ \left[\begin{array}{c} \underline{e}[k] \\ \underline{0} \end{array} \right], \quad (11.3)$$

ahol a $^+$ operátor a mátrix pszeudo-inverzét jelenti. Az $\underline{s}[k]$ uniform kiszámítási költsége $O(n^3)$, ahol n a Jacobi-mátrix oszlopainak száma.

Minden egyes iterációs lépésben szükség van a Jacobi-mátrix elemeinek meghatározására, azaz a parciális deriváltak számítására is.

Az α paraméter értéke a tanulás közben változtatható. Így az algoritmus k . iterációs lépése [55] alapján a következő:

1. adott $\underline{w}[k]$ és $\alpha[k]$
(Inicializálásként tetszőleges pozitív α értéket választhatunk, azaz $\alpha[1] > 0$)
2. $\mathbf{J}[k]$ és $\underline{e}[k]$ meghatározása
3. $\underline{s}[k]$ számítása (11.3) alapján.
4. Az úgynevezett megbízhatósági régió (trust region), $r[k]$ számítása a következőképpen: $r[k] = \frac{E(\underline{w}[k]) - E(\underline{w}[k] + \underline{s}[k])}{E(\underline{w}[k]) - \frac{1}{2} \|\mathbf{J}[k]\underline{s}[k] + \underline{e}[k]\|^2}$.
5. Az α paraméter értékét dinamikusan állítjuk, $r[k]$ értékétől függően:
 - Ha $r[k] < 0.25$ akkor $\alpha[k+1] = 4\alpha[k]$
 - Ha $r[k] > 0.75$ akkor $\alpha[k+1] = \alpha[k]/2$
 - Egyébként $\alpha[k+1] = \alpha[k]$

6. Ha $r[k] \leq 0$ akkor $\underline{w}[k+1] = \underline{w}[k]$, különben $\underline{w}[k+1] = \underline{w}[k] + \underline{s}[k]$.

Ha teljesül a megállási feltétel, vagy elértünk egy előre definiált maximális iterációs számot, akkor megállunk, különben folytatjuk a $(k+1)$. iterációs lépéssel.

12. fejezet

Intelligens számítási modellek identifikációja

Ebben a fejezetben számítási modellek identifikációjával foglalkozunk. Az identifikáció célja olyan fuzzy szabályalapú modell illetve neurális hálózat létrehozása, amely egy adott mintakészletre valamilyen hibakritérium alapján a lehető legjobban illeszkedik.

Az első alfejezetben egyenlő szárú háromszög alakú tagsági függvényeket alkalmazó fuzzy szabálybázis bakteriális evolúciós algoritmussal történő optimalizálását tárgyaljuk.

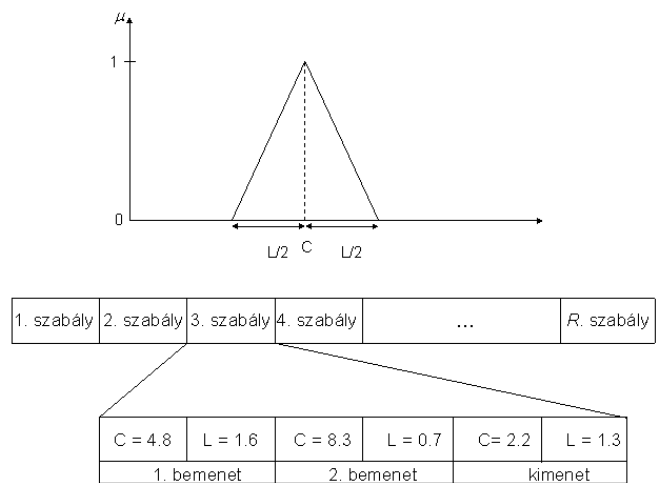
A következő három alfejezetben trapéz alakú tagsági függvényeket használó Mamdani-típusú, COG defuzzifikációs módszert alkalmazó fuzzy rendszerek szabálybázisának identifikációjával foglalkozunk. A 12.2. részben a szabályredukciós operátorokkal kiegészített bakteriális evolúciós algoritmust, a 12.3. részben a Levenberg-Marquardt módszert, a 12.4. részben pedig a bakteriális memetikus algoritmust mutatjuk be a szabályidentifikációs feladat megoldására.

A 12.5. részben bemutatjuk a bakteriális programozást B-spline típusú neurális hálózatok struktúrájának meghatározására alkalmazva.

12.1. Fuzzy szabályoptimalizálás bakteriális evolúciós algoritmussal

A 10.3. részben megismert bakteriális evolúciós algoritmust a módszert javasoló japán kutatók fuzzy szabálybázis optimalizálására alkalmazták [138]. Céljuk olyan háromszög alakú tagsági függvényekből felépülő szabálybázis meghatározása volt, amely egy adott mintakészletre valamilyen hibakritérium alapján a lehető legjobban illeszkedik.

A Nawa és Furuhashi által vizsgált fuzzy szabálybázisban a tagsági függvények egyenlő szárú háromszögekkel adottak, ily módon minden tagsági függvényt két paraméter jellemez: a magpont (a háromszög csúcsa)



12.1. ábra. A kódolási elrendezés

és a tartó hossza (a háromszög alapja). Az egyes szabályok szabálybázisbeli sorrendjük szerint szerepelnek a baktériumban, szabályonként a dimenziók indexének sorrendjében, legvégül a kimeneti dimenzió tagsági függvényével. Minden egyes baktérium egy ilyen szabálybázis kódját tartalmazza. Például a 12.1. ábrán szereplő baktériumhoz tartozó 3. szabály:

R_3 : Ha $x_1 = T(4,8; 1,6)$ és $x_2 = T(8,3; 0,7)$ akkor $y = T(2,2; 1,3)$,

ahol $T(C,L)$ olyan szimmetrikus háromszög alakú tagsági függvényt jelöl, melynek C a magpontja, L pedig a háromszög alapjának hossza.

A kódolási elrendezésen kívül az egyedek kiértékelési módját is meg kell adni. Ez azt mutatja meg, hogy az egyednek megfelelő szabálybázis mennyire illeszkedik jól a tanító mintakészletre. A hivatkozott cikkben [138] az átlagos relatív hibán alapuló kritériumot alkalmazták:

$$E = \frac{1}{m} \sum_{i=1}^m \frac{|t_i - y_i|}{t_i},$$

ahol t_i az i . mintára vonatkozó megkívánt kimenet, y_i az i . mintára számított modell kimenet, és m a minták száma.

A kezdeti populáció kialakítása véletlenszerűen létrehozott szabályokat jelent az egyes baktériumokban. A bakteriális mutációs és génátadási operátorokban szükség van a „kromoszóma” egy egységnyi részletének kiválasztására. A kromoszóma egy egységnyi részlete ebben az esetben egyetlen szabályt jelent. A bakteriális mutáció esetén tehát a klónokban egy véletlenszerűen kiválasztott szabály paraméterei módosulnak. A génátadás művelete során a célbaktérium egy egységnyi részletet, azaz egyetlen szabályt kap a forrásbaktériumtól.

12.2. Szabályredukciós operátorokkal kiegészített bakteriális evolúciós algoritmus alkalmazása

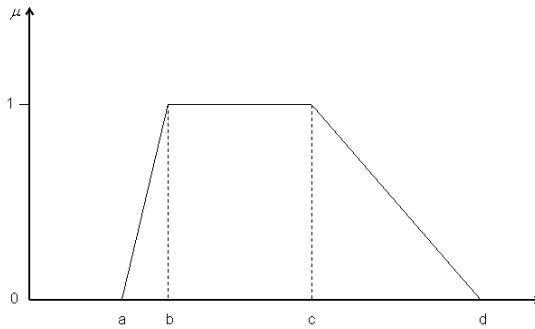
Ebben a részben a 12.1. alfejezetben megismert bakteriális evolúciós algoritmus olyan továbbfejlesztését ismertetjük, mely háromszög alakú tagsági függvények helyett az általánosabb trapéz alakú tagsági függvényeket használja, és ezenkívül újszerű szabályredukciós operátorokat is tartalmaz, melyek segítségével a szabálybázis mérete is optimálisan beállítható [23, 24].

12.2.1. A javasolt algoritmus

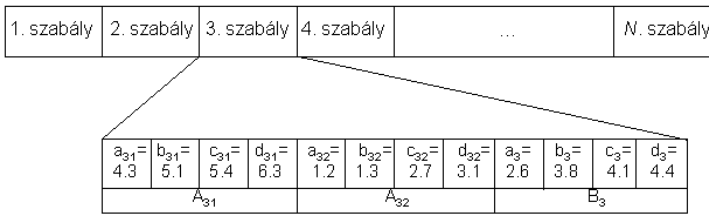
A bakteriális evolúciós algoritmus szabályredukciós operátorokkal kiegészített változata a következő:

```
kezdeti populáció létrehozása
generáció=0
amíg generáció < max. generáció
{
    bakteriális mutáció alkalmazása minden egyedre
    szabályredukciós operátorok minden egyedre:
    {
        szabály megszüntetés
        szabály egyesítés
        szemantikus elemzés
        szabály eltávolítás
    }
    génátadás
    generáció = generáció + 1
}
```

A baktérium kromoszómájában, csakúgy mint a 12.1. fejezetben tárgyalt első bakteriális evolúciós algoritmusban, a szabálybázis paraméterei, azaz a tagsági függvények vannak bekódolva. Az eredeti módszerrel ellentétben nem háromszög alakú, hanem trapéz alakú tagsági függvényeket használunk. Egy trapézt négy paraméterrel lehet megadni, szokásosan (és legegyszerűbben) a négy töréspontjával, ahogy a 12.2. ábrán látható. A négy paraméterre a trapéz tulajdonságaiból adódóan teljesülnie kell annak, hogy $a \leq b \leq c \leq d$. Az ábrán látható, hogy ha $b = c$, akkor a háromszög alakú tagsági függvényhez jutunk. Ha még $b - a = d - c$ (azaz $2b = a + d$) is teljesül, akkor az algoritmus eredeti változatában ([138]) alkalmazott egyenlő szárú háromszög alakú tagsági függvényt kapjuk. A trapéz alakú tagsági függvény a négy paraméterén kívül rendelkezik két indexszel is, melyek a helyét határozzák meg a szabálybázisban. Az $A_{ij}(a_{ij}, b_{ij}, c_{ij}, d_{ij})$ tagsági függvény a j . bemeneti változóhoz tartozik az i . szabályban. A $B_i(a_i, b_i, c_i, d_i)$ tagsági függvény a kimeneti változóhoz tartozik az i . szabályban. Egy bemeneti vektor j . dimenziója, x_j , az i . szabályban a következő tagsági értéket



12.2. ábra. A tagsági függvény paraméterei



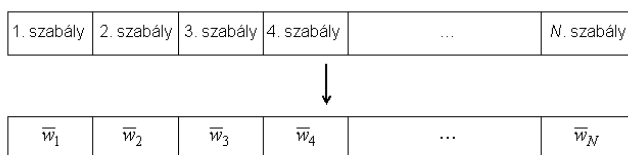
12.3. ábra. A kódolási elrendezés

eredményezi:

$$A_{ij}(x_j) = \begin{cases} \frac{x_j - a_{ij}}{b_{ij} - a_{ij}}, & \text{ha } a_{ij} < x_j < b_{ij} \\ 1, & \text{ha } b_{ij} \leq x_j < c_{ij} \\ \frac{d_{ij} - x_j}{d_{ij} - c_{ij}}, & \text{ha } c_{ij} \leq x_j < d_{ij} \\ 0, & \text{egyébként} \end{cases} \quad (12.1)$$

ahol $a_{ij} \leq b_{ij} \leq c_{ij} \leq d_{ij}$ szükségszerűen teljesül a töréspontok egymáshoz képesti elhelyezkedése miatt. Egy szabálynak összesen $4(n+1)$ paramétere van, ahol n a szabályok bemeneteinek száma. A szabályok a kromoszómában egymás után, felsorolásszerűen szerepelnek, ahogy az a 12.3. ábrán látható. A 12.3. ábrán egy két bemenetű, N darab szabályból álló szabálybázis elrendezése látható. A 3. szabály az ábra alapján a következő:

$$R_3: \quad \text{Ha } x_1 = A_{31}(4,3; 5,1; 5,4; 6,3) \text{ és } x_2 = A_{32}(1,2; 1,3; 2,7; 3,1) \text{ akkor} \\ y = B_3(2,6; 3,8; 4,1; 4,4).$$



12.4. ábra. Kromoszóma a génátadásnál

A kromoszóma összesen $4N(n + 1)$ valós számot tartalmaz, egy tagsági függvényen belül megtartva a rendezettséget.

A kezdeti populáció létrehozása a populációt alkotó N_{egyed} számú egyed véletlenszerű létrehozását jelenti. Minden egyedhez $N(n + 1)$ véletlenszerű tagsági függvényt generálunk, a tagsági függvényt alkotó trapéz négy paraméterénél ügyelve a trapéz paramétereinek a rendezettségére, továbbá arra is, hogy az adott változóhoz, melyre a tagsági függvény éppen generálódik, tartozó intervallumba beleessenek a trapéz paraméterei, vagy pedig valamekkora előre definiált megengedett túlnyúlási százalékot ne haladjanak meg.

A bakteriális mutáció hasonlóan történik, mint a háromszög alakú tagsági függvényt használó módszerénél, itt azonban az egyes klónokban a trapézok változnak meg. A trapézok megváltoztatásánál arra kell ismét ügyelni, hogy az új trapéz paraméterei megfeleljenek a korábban említett feltételeknek.

A bakteriális mutáció után a következő alfejezetben ismertetendő szabályredukciós operátorok alkalmazása történik meg. Végül a génátadási operátor használata következik, mely hasonlóan történik az eredeti módszerénél alkalmazotthoz, de egy továbbfejlesztett lehetőséggel. Nevezetesen, hogy a forrásbaktériumtól a célbaktériumnak adandó génrészletet nem muszáj véletlenszerűen kiválasztani, hanem jobb alternatíva lehet nagy aktivációs értékű szabályok átadása, mely kis aktivációs értékű szabályt ír felül a célbaktériumban. Ehhez az egyedeket alkotó szabályok aktivációs értékét, azaz az átlagos tüzelési értékét határozzuk meg a következőképpen (12.4. ábra):

$$\bar{w}_i = \frac{1}{m} \sum_{j=1}^m w_i^{(j)}, \quad (12.2)$$

ahol i a szabály indexe, $w_i^{(j)}$ az i . szabály tüzelési értéke a j . mintára, m a minták száma.

Az algoritmusban ügyelni kell arra, hogy a keletkezett szabálybázis ne legyen ritka, azaz minden lehetséges megfigyelésvektor esetén legyen tüzelő szabály. Ezt úgy garantáljuk, hogy nem engedünk olyan egyedet létrehozni, amelyik ezt a feltételt nem teljesíti. Az operátorok alkalmazása során körültekintően kell eljárni ennek a feltételnek a betartásához. A másik lehetőség büntető függvény alkalmazása lenne azokra az egyedekre, melyek a feltételt nem teljesítik, vagyis az ilyen egyedek kiértékelése a hibatagon kívül egy járulékos büntető tagot is tartalmaz. Ebben az esetben azonban a büntető függvény rossz megválasztásával, illetve nem megfelelő súlyozásával létrejöhetnek nem megfelelő szabálybázist reprezentáló egyedek.

Szabályredukciós operátorok

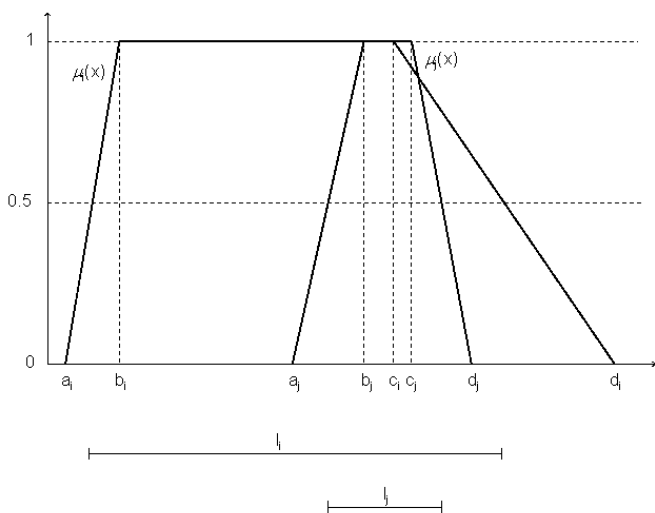
A bakteriális operátorok az optimálishoz egyre közelebb kerülő szabálybázist hoznak létre. Sokszor szükség van azonban arra is, hogy ne csak optimális legyen a szabálybázis, hanem minél kisebb méretű is legyen. A hatástalan szabályokat megszüntetve, a hasonlókat pedig összevonva egy szabályba, csökkenthető a szabálybázis mérete anélkül, hogy a szabálybázis által adott hiba jelentősen megnőne ezen egyszerűsítő műveletek eredményeképpen. Szabályredukciós operátorokat korábban más szerzők is javasoltak [136, 167]. Ezekben a művekben azokban nem az általános és igen elterjedt trapéz alakú tagsági függvényekre definiálják az egyszerűsítő operátorokat, hanem háromszög illetve Gauss-görbe alakú tagsági függvényekre. Ezen operátorok helyett a könyvben a következőkben ismertetendő, trapéz alakú tagsági függvényekre definiált szabálybázis egyszerűsítő műveletek végrehajtását javasoljuk minden egyedre.

Szabály megszüntetés

Ha egy tagsági függvény túl keskennyé válik, akkor megszüntetjük a szabályt, amelyik használja. Ilyenkor ugyanis ennek a szabálynak a tüzelési értéke elenyésző lesz a többi szabályéhoz képest. A megszüntetés kritériuma a következő:

$$l_i \mu_i \left(\frac{a_j + b_j + c_j + d_j}{4} \right) \geq \beta l_j, \quad (12.3)$$

ahol l_i és l_j az adott változóhoz tartozó tagsági függvény középvonalának hossza az i . és a j . szabályban, β pedig a megszüntetés paramétere. A 12.5. ábrán láthatóan az ugyanahhoz a változóhoz tartozó két tagsági



12.5. ábra. Tagsági függvény megszüntetése

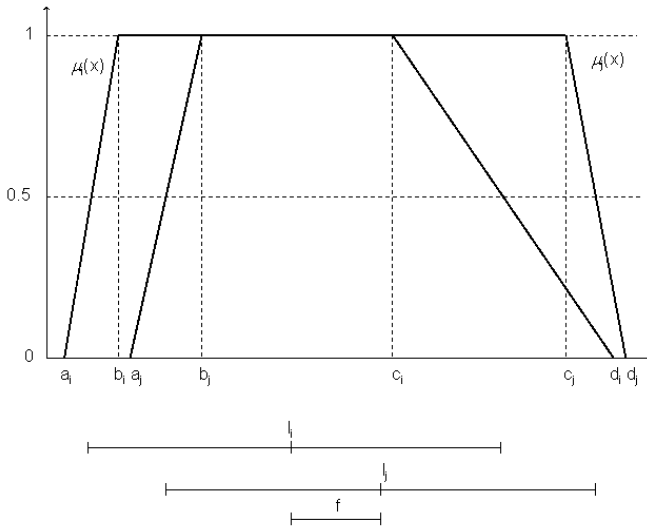
függvény az egyik szabályban sokkal szélesebb, mint a másik szabályban, és a szélesebb majdnem teljesen magába foglalja a keskenyebbet. Ilyenkor megszüntetjük azt a szabályt, amelyikben a keskenyebb tagsági függvény van. A megszüntetés szigorúsága a β paraméterrel szabályozható. Minél nagyobb β értéke, annál szigorúbb a megszüntetés feltétele.

Szabály egyesítés

Ha ugyanahhoz a változóhoz tartozó két tagsági függvény hasonlóvá válik, akkor egyesíthetők egyetlen közös tagsági függvénné. A hasonlóság fogalmától két kritériumot követelünk meg. Egyrészt a két tagsági függvény szélessége legyen körülbelül ugyanakkora, és legyenek közel egymáshoz (12.6. ábra). Ennek alapján a két feltétel:

$$\left| \frac{l_i}{l_j} - 1 \right| < \gamma \text{ és } |f| < \gamma, \quad (12.4)$$

ahol l_i és l_j az adott változóhoz tartozó tagsági függvény középvonalának hossza az i . és a j . szabályban, f pedig az l_i és l_j középpontjai közötti távolság. A tagsági függvények egyesítéséhez mindkét feltételt teljesíteni kell, de csak egy paramétert használunk, γ -t, mellyel az egyesítés szigorúságát



12.6. ábra. Tagsági függvények egyesítése

befolyásoljuk. Minél kisebb γ értéke, annál szigorúbb az egyesítés feltétele. Az egyesítés után a két tagsági függvény meg fog egyezni, a paramétereik pedig a következők lesznek:

$$z_{új} = \frac{z_i l_i + z_j l_j}{l_i + l_j}, \quad (12.5)$$

ahol z rendre behelyettesítendő a, b, c és d -vel.

Szemantikus elemzés

Ha két szabály feltételrészre megegyezik, viszont a következményük különböző, akkor a kimeneti változóhoz tartozó tagsági függvényeket (azaz a következményeket) egyesítjük egy közös tagsági függvénnyé, az egyesítésnél megismert képlet segítségével.

Szabály eltávolítás

Ha két szabály megegyezik, akkor az egyiket eltávolítjuk. Az egyesítések során előfordul, hogy két szabály feltételrészre azonos lesz. Ezután a szemantikus elemzés egyesíti a következményeiket is, tehát a

12.1. táblázat. Rögzített szabálysám

N_{egyed}	$N_{klón}$	N_{inf}	legjobb egyed	átlag	legrosszabb egyed	teszt hiba	szabályok száma
10	10	4	5,09	5,82	6,12	12,3	6
10	10	8	5,04	5,76	6,23	11,9	6
6	10	11	5,21	5,93	6,41	12,2	5

két szabály megegyező lesz, az egyiket töröljük. Az egyesítések hatása ennél fogva ezen két utóbbi operátor alkalmazása után fog érvényesülni.

12.2.2. A módszer tesztelése

Szimulációs vizsgálatokat hajtottunk végre az algoritmusban szereplő operátorok elemzése céljából. A vizsgálatokat egy az irodalomban ismert problémán hajtottuk végre. Ennél a referencia problémánál a cél a következő hat változós függvény fuzzy szabálybázissal történő approximációja:

$$y = x_1 + \sqrt{x_2} + x_3x_4 + 2e^{2(x_5-x_6)}, \tag{12.6}$$

ahol $x_1 \in [1; 5]$, $x_2 \in [1; 5]$, $x_3 \in [0; 4]$, $x_4 \in [0; 0,6]$, $x_5 \in [0; 1]$, $x_6 \in [0; 1,2]$.
A bakteriális operátorokban az egyedeket a következő hibadefiníció alapján értékeltük ki, és a kapott eredményeket is ezen hibakritérium alapján hasonlítottuk össze:

$$E = \frac{1}{m} \sum_{i=1}^m \frac{|t_i - y_i|}{I_{max} - I_{min}}, \tag{12.7}$$

ahol t_i az i . mintára vonatkozó megkívánt kimenet, y_i az i . mintára számított modell kimenet, m a minták száma, I_{max} a kimeneti változó felső korlátja, I_{min} pedig az alsó korlátja, azaz a hiba normálva van a kimeneti változó intervallumának hosszával.

Az első futtatási eredmények a 12.1. táblázatban láthatók. Ennek során nem használtuk a szabályredukciós operátorokat, a szabályok száma rögzítve volt. A tanító minták száma és a tesztelési minták száma egyaránt 200, a generációs szám 40. A szimuláció célja a szabály csökkentő operátorok alkalmazásának vizsgálata a szabálybázisban lévő redundancia redukálására. Számos szimulációt hajtottunk végre annak érdekében, hogy megtaláljuk

12.2. táblázat. *A megszüntető paraméter (β) hatása*

β	átlagos szabálysorszám	tanítási hiba	tesztelési hiba
5	4,7	5,23	8,94
6	5,4	5,02	8,47
8	7,6	4,52	8,32
10	7,8	4,13	8,17

ezen operátorok paramétereinek optimális értékeit, ami azért szükséges, hogy elkerüljük a „túl egyszerű” szabálybázisok generálását. A legjobb baktériumra vonatkozó futtatási eredmények a 12.2. táblázatban láthatók. Az egyesítés operátor paramétere $\gamma = 5\%$, a kezdeti szabálysorszám 10, az egyedek száma 10, a klónok száma 20, az infekciók száma pedig 4. A táblázat eredményei 10 futtatás átlagát mutatják. A β paraméter értékének a növelésével a megmaradó szabályok száma is növekszik, mert a szabályok megszüntetésének kritériuma szigorodik ilyenkor. A táblázatból látszik, hogy ha a szabályok száma a végső szabálybázisban nagyobb, akkor ennek megfelelően a tanítási hiba kisebb, viszont a tesztelési hiba csökkenése csak kisebb mértékű. Ennek oka, hogy egy nagyobb szabálybázis túl specifikus a tanító készletre, és nincs sok előnye, amikor egy független mintakészletre alkalmazzák. Ennek alapján megállapítható, hogy nem szükséges β -nak nagy értéket adni, mert akkor a megszüntetés szigorúsága miatt az túl komplex szabálybázist eredményez, mely hasonló képességű, mint amilyen egy kisebb szabálybázis lenne.

12.3. A Levenberg-Marquardt algoritmus alkalmazása Mamdani-típusú fuzzy szabálybázis optimalizálására

Amint azt a 11.7. alfejezetben láttuk, a Levenberg-Marquardt algoritmus jó konvergencia tulajdonságokkal rendelkezik lokális környezetben történő kereséseknél. Az algoritmus általános nemlineáris optimalizációs módszer olyan problémák megoldására, ahol ismertek a célfüggvény deriváltjai. A módszert sikeresen alkalmazták neurális hálózatok paramétereinek a meghatározására [151]. Ebben a fejezetben az algoritmus Mamdani-típusú, COG defuzzifikációs eljárást alkalmazó, trapéz alakú tagsági függvénye-

ket használó fuzzy rendszerek paramétereinek meghatározására történő alkalmazását ismertetjük [25, 20].

12.3.1. A Jacobi-mátrix meghatározása

Az algoritmus minden egyes iterációs lépésében szükséges a Jacobi-mátrix elemeinek, azaz az összes parciális deriváltak a kiszámítása. A fuzzy rendszer szabályainak számát a teljes algoritmus működése során rögzítjük. A szabályokban trapéz alakú tagsági függvényeket használunk, amelyek a következő formában írhatók fel:

$$\mu_{ij}(x_j) = \frac{x_j - a_{ij}}{b_{ij} - a_{ij}} N_{i,j,1}(x_j) + N_{i,j,2}(x_j) + \frac{d_{ij} - x_j}{d_{ij} - c_{ij}} N_{i,j,3}(x_j). \quad (12.8)$$

Ebben az összefüggésben az a_{ij} , b_{ij} , c_{ij} , d_{ij} értékek az i . szabály j . bemeneti változójához tartozó tagsági függvény négy paraméterét jelentik. A kimeneti változóhoz az i . szabály esetén az a_i , b_i , c_i , d_i értékek tartoznak, továbbá:

$$\begin{aligned} N_{i,j,1}(x_j) &= \begin{cases} 1, & \text{ha } x_j \in [a_{ij}, b_{ij}] \\ 0, & \text{ha } x_j \notin [a_{ij}, b_{ij}] \end{cases} \\ N_{i,j,2}(x_j) &= \begin{cases} 1, & \text{ha } x_j \in (b_{ij}, c_{ij}) \\ 0, & \text{ha } x_j \notin (b_{ij}, c_{ij}) \end{cases} \\ N_{i,j,3}(x_j) &= \begin{cases} 1, & \text{ha } x_j \in [c_{ij}, d_{ij}] \\ 0, & \text{ha } x_j \notin [c_{ij}, d_{ij}] \end{cases} \end{aligned} \quad (12.9)$$

A trapézokat tehát ugyanúgy, ahogy korábban, a négy törésponttal írjuk le a 12.2. ábrán látható módon.

Az eredeti Mamdani-algoritmusnak megfelelően a következtetés során standard (min) t-normát alkalmazunk, azaz az i . szabály illeszkedési mértéke valamely n -dimenziós crisp $\underline{x}$ megfigyelés vektor esetén:

$$w_i = \min_{j=1}^n \mu_{ij}(x_j). \quad (12.10)$$

A fuzzy következtetés kimenete az alkalmazott trapézok esetében a COG defuzzifikációs eljárás használatával a következő explicit alakba írható:

$$y(\underline{x}) = \frac{\sum_{i=1}^R 3w_i(d_i^2 - a_i^2)(1 - w_i) + 3w_i^2(c_i d_i - a_i b_i) + w_i^3(c_i - d_i + a_i - b_i)(c_i - d_i - a_i + b_i)}{\sum_{i=1}^R 2w_i(d_i - a_i) + w_i^2(c_i + a_i - d_i - b_i)}. \quad (12.11)$$

A szabályok száma R , a szabályok n bemenetűek. Ezen összefüggések alapján meghatározzuk a kimenetnek a modell paramétereire szerinti parciális deriváltjait, majd ezek alapján a Jacobi-mátrixot. A mátrix egy-egy sorát egy-egy minta alapján határozzuk meg. A p . sor az alábbi alakot veszi fel:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial y(\underline{x}^{(p)})}{\partial a_{11}} & \frac{\partial y(\underline{x}^{(p)})}{\partial b_{11}} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial a_{12}} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial d_1} & \dots & \frac{\partial y(\underline{x}^{(p)})}{\partial d_R} \end{bmatrix}, \quad (12.12)$$

ahol p a minta azonosítója és

$$\begin{aligned} \frac{\partial y(\underline{x}^{(p)})}{\partial a_{ij}} &= \frac{\partial y}{\partial w_i} \frac{\partial w_i}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial a_{ij}} \\ \frac{\partial y(\underline{x}^{(p)})}{\partial b_{ij}} &= \frac{\partial y}{\partial w_i} \frac{\partial w_i}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial b_{ij}} \\ \frac{\partial y(\underline{x}^{(p)})}{\partial c_{ij}} &= \frac{\partial y}{\partial w_i} \frac{\partial w_i}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial c_{ij}} \\ \frac{\partial y(\underline{x}^{(p)})}{\partial d_{ij}} &= \frac{\partial y}{\partial w_i} \frac{\partial w_i}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial d_{ij}}. \end{aligned} \quad (12.13)$$

A 12.10. összefüggésből kitűnik, hogy a w_i értékek csak a tagsági függvényektől függenek, és minden tagsági függvénynek négy paramétere van. Ezért w_i deriváltjai a következők lesznek:

$$\frac{\partial w_i}{\partial \mu_{ij}} = \begin{cases} 1, & \text{ha } \mu_{ij} = \min_{k=1}^n \mu_{ik} \\ 0, & \text{egyébként.} \end{cases} \quad (12.14)$$

A tagsági függvények deriváltjai a következő módon számíthatók:

$$\begin{aligned} \frac{\partial \mu_{ij}}{\partial a_{ij}} &= \frac{x_j^{(p)} - b_{ij}}{(b_{ij} - a_{ij})^2} N_{i,j,1}(x_j^{(p)}) \\ \frac{\partial \mu_{ij}}{\partial b_{ij}} &= \frac{a_{ij} - x_j^{(p)}}{(b_{ij} - a_{ij})^2} N_{i,j,1}(x_j^{(p)}) \\ \frac{\partial \mu_{ij}}{\partial c_{ij}} &= \frac{d_{ij} - x_j^{(p)}}{(d_{ij} - c_{ij})^2} N_{i,j,3}(x_j^{(p)}) \\ \frac{\partial \mu_{ij}}{\partial d_{ij}} &= \frac{x_j^{(p)} - c_{ij}}{(d_{ij} - c_{ij})^2} N_{i,j,3}(x_j^{(p)}). \end{aligned} \quad (12.15)$$

$\frac{\partial y}{\partial w_i}$ -t és a kimeneti tagsági függvények deriváltjait szintén ki kell számítani. A 12.11. összefüggés alapján a következő írható fel:

$$\begin{aligned}\frac{\partial y}{\partial w_i} &= \frac{1}{3} \frac{N \frac{\partial F_i}{\partial w_i} - S \frac{\partial G_i}{\partial w_i}}{N^2} \\ \frac{\partial y}{\partial a_i} &= \frac{1}{3} \frac{N \frac{\partial F_i}{\partial a_i} - S \frac{\partial G_i}{\partial a_i}}{N^2} \\ \frac{\partial y}{\partial b_i} &= \frac{1}{3} \frac{N \frac{\partial F_i}{\partial b_i} - S \frac{\partial G_i}{\partial b_i}}{N^2} \\ \frac{\partial y}{\partial c_i} &= \frac{1}{3} \frac{N \frac{\partial F_i}{\partial c_i} - S \frac{\partial G_i}{\partial c_i}}{N^2} \\ \frac{\partial y}{\partial d_i} &= \frac{1}{3} \frac{N \frac{\partial F_i}{\partial d_i} - S \frac{\partial G_i}{\partial d_i}}{N^2},\end{aligned}\tag{12.16}$$

ahol N a 12.11 tört nevezője, S pedig a számlálója. F_i az összeg i -ik tagja a számlálóban, G_i az i . tag a nevezőben. A deriváltak a következőképpen számíthatók:

$$\begin{aligned}\frac{\partial F_i}{\partial w_i} &= 3(d_i^2 - a_i^2)(1 - 2w_i) + 6w_i(c_i d_i - a_i b_i) + 3w_i^2[(c_i - d_i)^2 - (a_i - b_i)^2] \\ \frac{\partial G_i}{\partial w_i} &= 2(d_i - a_i) + 2w_i(c_i + a_i - d_i - b_i)\end{aligned}\tag{12.17}$$

$$\begin{aligned}\frac{\partial F_i}{\partial a_i} &= -6w_i a_i + 6w_i^2 a_i - 3w_i^2 b_i - 2w_i^3(a_i - b_i) & \frac{\partial G_i}{\partial a_i} &= -2w_i + w_i^2 \\ \frac{\partial F_i}{\partial b_i} &= -3w_i^2 a_i + 2w_i^3(a_i - b_i) & \frac{\partial G_i}{\partial b_i} &= -w_i^2 \\ \frac{\partial F_i}{\partial c_i} &= 3w_i^2 d_i - 2w_i^3(d_i - c_i) & \frac{\partial G_i}{\partial c_i} &= w_i^2 \\ \frac{\partial F_i}{\partial d_i} &= 6w_i d_i - 6w_i^2 d_i + 3w_i^2 c_i + 2w_i^3(d_i - c_i) & \frac{\partial G_i}{\partial d_i} &= 2w_i - w_i^2.\end{aligned}\tag{12.18}$$

A mátrix oszlopainak száma $4(n + 1)R$, sorainak száma pedig a minták számával egyezik meg.

12.3.2. A módszer alkalmazása

Szimulációs vizsgálatokat hajtottunk végre, amelyekben az irodalomban található referenciaproblémákon összehasonlítottuk az eljárást más módszerekkel.

A Levenberg-Marquardt algoritmus trapéz alakú tagsági függvényeket használó fuzzy rendszer paramétereinek optimalizálása során olyan paraméter módosító vektort adhat eredményül, mely megváltoztathatja a trapéz töréspontjainak egymáshoz képesti helyzetét. Ezért ilyen esetekben az eredményül kapott paraméter módosító vektort át kell alakítani, hogy az ne módosítsa a trapézok töréspontjainak egymáshoz képesti helyzetét. Ennek végrehajtása az irodalomból ismert korrekciós technikával történik [151]. Ha a $p_i < p_{i+1}$ rendezettségnek fenn kell állnia két szomszédos paraméter között, és a paraméter módosító vektor ezt megsérti, akkor a következő korrekciós együtthatót határozzuk meg:

$$g = \frac{p_{i+1}[k] - p_i[k]}{2 \cdot (\Delta p_{i+1}[k] - \Delta p_i[k])}. \quad (12.19)$$

Az új paraméter értékek pedig a g korrekciós tag segítségével a következőképpen számíthatók:

$$\begin{aligned} p_{i+1}[k+1] &= p_{i+1}[k] - g \cdot \Delta p_{i+1}[k] \\ p_i[k+1] &= p_i[k] - g \cdot \Delta p_i[k]. \end{aligned} \quad (12.20)$$

Így garantálható a trapéz töréspontjainak megfelelő sorrendje a módosítás meghatározása után.

Referenciaproblémák

Az algoritmus tesztelésére két tudományos problémát választottunk. Mindkét feladatnál a következő megállási feltételt használtuk a Levenberg-Marquardt algoritmusra:

$$\begin{aligned} E[k-1] - E[k] &< \theta[k] \\ \|\underline{par}[k-1] - \underline{par}[k]\| &< \sqrt{\tau_f}(1 + \|\underline{par}[k]\|) \\ \|\underline{g}[k]\| &\leq \sqrt[3]{\tau_f}(1 + |E[k]|), \end{aligned} \quad (12.21)$$

ahol $\theta[k] = \tau_f(1 + |E[k]|)$ és $\tau_f = 10^{-4}$. A $\underline{g}[k]$ a 11.6. alfejezetben definiált gradiens vektor, a $\underline{par}$ vektor pedig a tagsági függvények paramétereit tartalmazza sorrendben, szabályonként, és egy szabályon belül pedig az egyes dimenziók mentén haladva. A $\underline{par}$ vektor a 11.6. alfejezetbeli $\underline{w}$ súlyvektornak, azaz az optimalizálandó elemeknek felel meg. Az optimalizálandó paraméterek száma, mely megegyezik a Jacobi mátrix oszlopainak számával $4(n+1)R$, mert R a szabályok száma, mindegyikben n bemeneti és egy

kimeneti fuzzy halmaz található, és egy fuzzy halmazt négy paraméterrel adunk meg.

Az egyik referencia feladat az ún. *pH probléma*, mely egyváltozós. A cél ennél a feladatnál egy titrálási görbe inverzének a közelítése. Ez a fajta nemlinearitás a kémiai pH értékkel van kapcsolatban. A tanítómintákat a következő egyenlet alapján generáljuk:

$$pH' = -\frac{\log\left(\sqrt{\frac{y^2}{4} + 10^{-14}} - \frac{y}{2}\right) + 6}{26}, \quad y = 2 \cdot 10^{-3}x - 10^{-3} \quad (12.22)$$

A tanításhoz 101 mintát generálunk.

A másik referencia feladat az ún. *inverz koordináta transzformációs probléma* (ICT), amely kétváltozós. Ennél a feladatnál inverz transzformáció történik két koordináta és egy kétkarú manipulátor egyik szöge között. A tanítómintákat a következő összefüggések alapján állítjuk elő:

$$\begin{aligned} \Theta_2 &= \pm \tan^{-1}\left(\frac{s}{c}\right) \\ s &= \sqrt{1 - c^2} = \sin(\Theta_2) \\ c &= \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} = \cos(\Theta_2). \end{aligned} \quad (12.23)$$

A cél az $(x, y) \mapsto \Theta_2$ függvény közelítése. A tanításhoz 110 mintát generálunk.

Hibadefiníciók

Az algoritmus tesztelésére a következő hibadefiníciókat alkalmazzuk: az *átlagos négyzetes hibát* (Mean Square of Error, MSE), az *átlagos négyzetes relatív hibát* (Mean Square of Relative Error, MSRE), és az *átlagos relatív százalékos hibát* (Mean Relative Error Percentage, MREP), melyeket a következő módon értelmezzünk:

$$\begin{aligned} MSE &= \frac{1}{m} \sum_{i=1}^m (t_i - y_i)^2 \\ MSRE &= \frac{1}{m} \sum_{i=1}^m \frac{(t_i - y_i)^2}{y_i^2} \\ MREP &= \frac{100}{m} \sum_{i=1}^m \left| \frac{t_i - y_i}{y_i} \right|, \end{aligned} \quad (12.24)$$

12.3. táblázat. Az eredmények összefoglalása a pH problémára

módszer	kezdőpont	végpont	kezdeti MSE	vég MSE	iterációk száma
LM	{0,12; 0,20}	{0,355; 0,744}	0,019	0,0100	6
LM	{0,40; 0,50}	{0,387; 0,763}	0,013	0,0100	4
LM	{0,60; 0,65}	{0,355; 0,747}	0,011	0,0100	8
BProp	{0,12; 0,20}	{0,278; 0,744}	0,019	0,0100	86
BProp	{0,40; 0,50}	{0,398; 0,743}	0,013	0,0100	38
BProp	{0,60; 0,65}	{0,355; 0,747}	0,011	0,0100	8

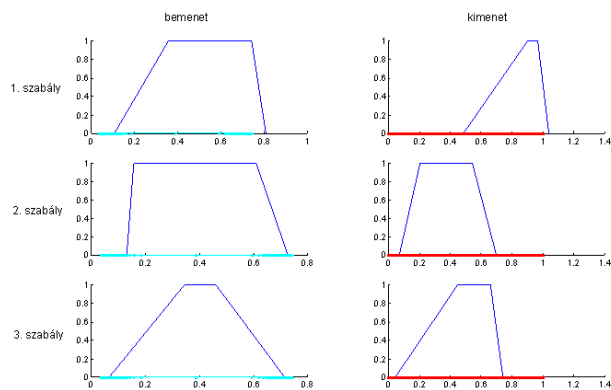
ahol t_i az i . mintára vonatkozó megkívánt kimenet, y_i az i . mintára számított modell kimenet, és m a minták száma.

Két paraméter optimalizációja

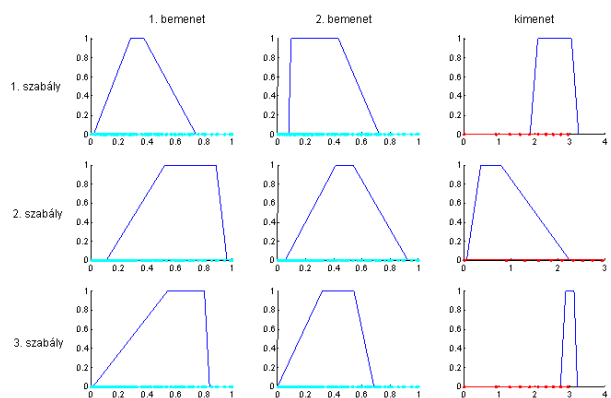
Az első szimulációs vizsgálat a Levenberg-Marquardt (LM) algoritmus tanítási képességeit mutatja, mindkét problémára, de az egyszerűség kedvéért csak két paramétert optimalizálva. Ezek az első szabály első bemeneti tagsági függvényének b és c paraméterei. Az LM algoritmus teljesítményét a hiba-visszaterjesztéses módszerrel (backpropagation, BProp, 11.6. alfejezet) hasonlítjuk össze. A fuzzy rendszert három szabályból építjük fel, a kezdeti szabálybázisok a 12.7. és a 12.8. ábrán láthatók. Egy lokális minimum a pH probléma esetén kb. a $\{b, c\} = \{0,349; 0,800\}$ pontban található, az MSE érték 0,01; az ICT esetén pedig a $\{b, c\} = \{0,128; 0,245\}$ pontban, ahol az MSE érték 0,865. Három különböző kezdeti paraméter-elrendezést vizsgáltunk, a kezdeti- és a végállapotbeli hiba (MSE) és a paraméter értékek a következő ábrákon (a 12.9. ábrától a 12.12. ábráig) és a 12.3. és 12.4. táblázatban figyelhetők meg. A táblázatok jól szemléltetik, hogy az LM módszer sokkal gyorsabban konvergál, mint a BProp. Mindkét algoritmus a lokális optimum közelébe vezet és jól közelítik a minimális hibát, de az LM algoritmusnak sokkal kevesebb iterációra van ehhez szüksége, mint a BProp módszernek.

Az összes paraméter egyidejű optimalizációja

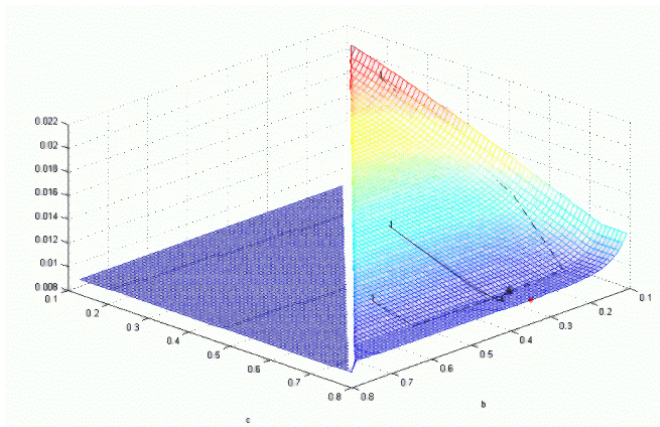
A következő szimuláció célja az volt, hogy a fuzzy szabálybázis minden egyes paraméterét optimalizáljuk. A 12.13. ábrán a szabálybázis paramétereinek alakulását látjuk az LM iterációk függvényében. A pH problémát



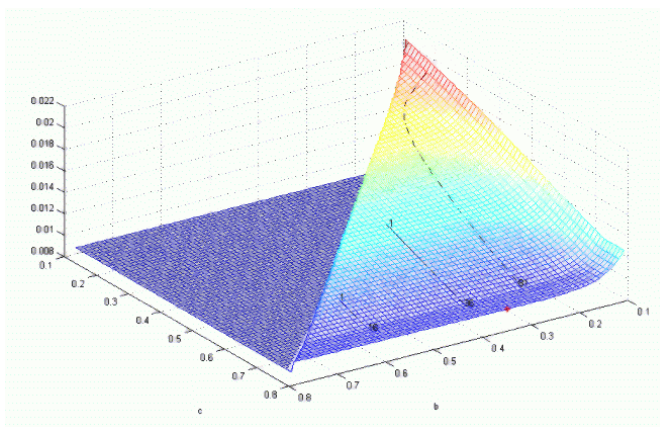
12.7. ábra. Kezdeti szabálybázis a pH problémára



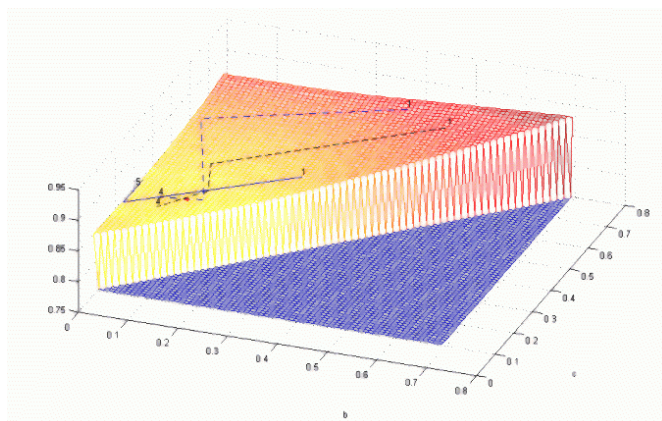
12.8. ábra. Kezdeti szabálybázis az ICT problémára



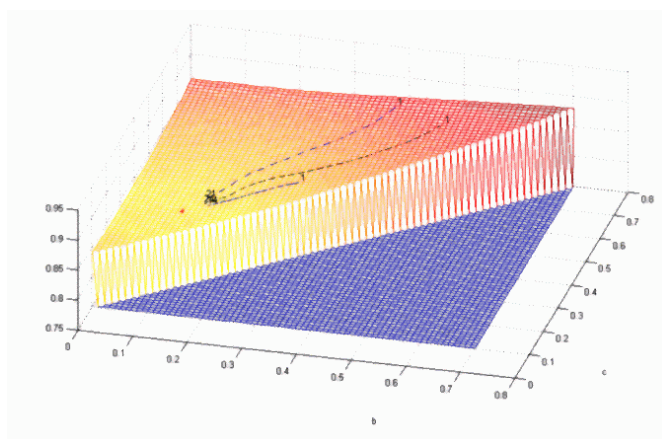
12.9. ábra. LM módszer teljesítménye a pH problémára



12.10. ábra. BProp módszer teljesítménye a pH problémára



12.11. ábra. *LM módszer teljesítménye az ICT problémára*

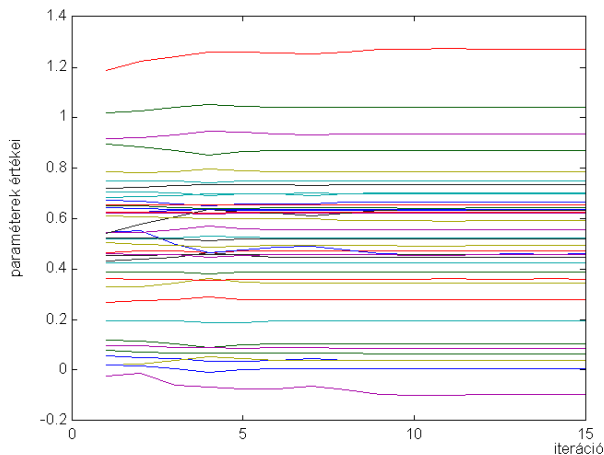


12.12. ábra. *BProp módszer teljesítménye az ICT problémára*

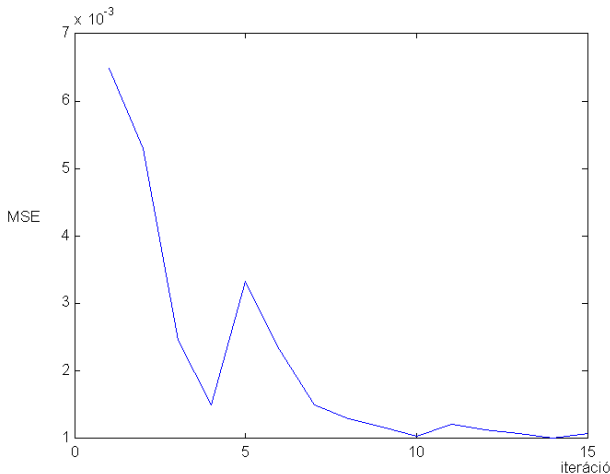
12.4. táblázat. Az eredmények összefoglalása az ICT problémára

módszer	kezdőpont	végpont	kezdeti MSE	vég MSE	iterációk száma
LM	{0,40; 0,70}	{0,100; 0,239}	0,890	0,8650	4
LM	{0,30; 0,40}	{0,190; 0,256}	0,870	0,8651	4
LM	{0,50; 0,65}	{0,113; 0,218}	0,890	0,8653	7
BProp	{0,40; 0,70}	{0,155; 0,301}	0,890	0,8652	21
BProp	{0,30; 0,40}	{0,162; 0,279}	0,870	0,8650	14
BProp	{0,50; 0,65}	{0,156; 0,283}	0,890	0,8650	24

vizsgáljuk, 15 iterációt hajtunk végre az algoritmusban, a szabálybázis 5 szabályt tartalmaz (azaz 40 paramétert optimalizálunk). A 12.14. ábrán az átlagos négyzetes hiba (MSE) alakulása látható. A 12.5. táblázatban egy összehasonlító vizsgálat eredményét láthatjuk. Különböző típusú hibaértékeket számítva hasonlítottuk össze az LM algoritmust és a Bakteriális Evolúciós Algoritmust (12.2. fejezet). Utóbbiban 10 baktériumot használtunk 40 generáción keresztül, 8 klónt és 4 infekciót alkalmazva a bakteriális operátorokban rögzített szabálysorszám mellett. A 12.14. ábrán látható MSE görbét figyelve megállapíthatjuk, hogy az LM algoritmus valóban optimali-



12.13. ábra. A fuzzy rendszer paramétereinek alakulása



12.14. ábra. Az MSE érték fejlődése

12.5. táblázat. A teljes fuzzy szabálybázis optimalizálása a pH problémára a Levenberg-Marquardt módszerrel és a Bakteriális Evolúciós Algoritmussal

fuzzy szabálybázis	MSE	MSRE	MREP
LM – kezdeti	$6,5 \cdot 10^{-3}$	$1,98 \cdot 10^{-1}$	25,5
LM – végállapot	$9,83 \cdot 10^{-4}$	$1,42 \cdot 10^{-1}$	16,7
BEA – végállapot	$7,01 \cdot 10^{-4}$	$1,23 \cdot 10^{-1}$	14,9

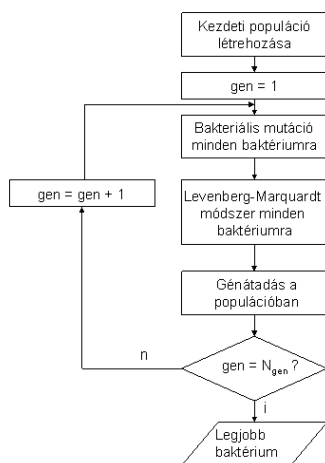
zálja a szabályokat, igyekszik megtalálni a legközelebbi lokális optimumot a hibát minimálisra csökkentve. Tizenöt iteráció elegendő volt a lokális optimum közelébe jutásához. A 12.5. táblázatból látható, hogy a kiindulási ponthoz képest az MSE hiba milyen mértékben csökkent a szabálybázison. Habár a bakteriális evolúciós algoritmus valamelyest jobb eredményt adott, ez annak köszönhető, hogy ez a módszer az LM algoritmussal ellentétben a globális optimumba konvergál. A megfelelő pontból indított LM módszerrel 15 iterációs lépés után elért eredmény így sem sokkal marad el a 10 egyed 40 generáción át használó bakteriális megközelítéstől. A két módszert kombinálva viszont még jobb eredményt érhetünk el (12.4. fejezet).

12.4. Bakteriális memetikus algoritmus alkalmazása Mamdani-típusú fuzzy szabálybázis optimalizálására

A bakteriális evolúciós algoritmusok sikeresen alkalmazhatók fuzzy szabálybázisok identifikációjára. A bakteriális operátorok biztosítják a populáció fejlődését, „evolúcióját”, azaz egyre jobb szabálybázisok kialakulását. A Levenberg-Marquardt algoritmust is sikeresen alkalmaztuk trapéz alakú tagsági függvényeket használó fuzzy szabályok lokális hangolására. A bakteriális megközelítés evolúciós típusú módszer, mely ennél fogva globális jellegű keresést tesz lehetővé, viszont az algoritmus által talált megoldás meglehetősen lassan konvergál. A Levenberg-Marquardt algoritmus, mely gradiens alapú technika, képes pontosabb megoldást találni, viszont hátránya, hogy ezt valamely lokális környezetben teszi, és így az optimalizáció során könnyen lokális minimumba ragadhatunk. Az evolúciós és gradiens alapú technikák kombinálásával mindkét típusú módszer előnyei kihasználhatók, és hátrányaik kiküszöbölhetők. A kétféle megközelítés kombinációját az irodalomban memetikus algoritmusnak szokás nevezni [133]. Ezen módszerek nem a „darwinizmus evolúciós modell” [38] alkalmazzák, hanem az ún. „lamarcki evolúciót” [124], melynek lényege, hogy az egyedek nemcsak az örökölt tulajdonságaikat adják tovább utódaiknak, hanem az „életük során szerzett”, azaz a tanult tulajdonságaikat is. Habár a biológiában ez az elv hibás, kiválóan alkalmazható az informatikában, számos memetikus eredmény található a szakirodalomban [2, 143, 166]. A hagyományos evolúciós operátorok mellett ezekben a módszerekben megjelenik a tanulás is, mely valamely lokális keresést jelent, amely által az egyed tökéletesebbé válik. A bakteriális megközelítés és a Levenberg-Marquardt algoritmus külön-külön a saját kategóriájuk legjobb módszerei közé tartoznak. Kézenfekvő a gondolat, hogy e két módszert célszerű kombinálni. A bakteriális operátorok által megvalósított hatékony evolúciót az egyes baktériumokon alkalmazott Levenberg-Marquardt algoritmus teszi még tökéletesebbé. A kombinált módszert *bakteriális memetikus algoritmusnak* nevezzük [21].

12.4.1. A javasolt algoritmus

A bakteriális memetikus algoritmus folyamatábrája a 12.15. ábrán látható. A bakteriális mutációs és a génátadási lépés között a Levenberg-Marquardt algoritmust alkalmazzuk minden egyes egyedre (baktériumra).



12.15. ábra. Bakteriális memetikus algoritmus

A módszert, csakúgy mint az előző alfejezetekben javasolt technikákat, Mamdani-típusú, trapéz alakú tagsági függvényeket és COG defuzzifikációs módszert alkalmazó fuzzy szabálybázis optimalizációjára alkalmaztuk. A kódolási elrendezés és a szükséges definíciók megegyeznek az előző alfejezetekben alkalmazottakkal.

Az algoritmusnak kétféle változatát ismertetjük. Az első változatban a szabálybázis mérete a folyamat során állandó, azaz az egyedeket mindig ugyanannyi szabály alkotja, vagyis a baktériumok hossza a fejlődés során állandó és a baktériumok egyforma hosszúak [21, 94]. Ilyenkor a már megismert bakteriális mutációt és génátadást használjuk ebben az algoritmusban is, változatlan formában. A génátadásnál a forrásbaktériumtól kapott szabály egy szabályt ír felül a célbaktériumban.

Az algoritmus másik változatában az egyes baktériumok hossza a folyamat során változhat, és egymástól is különbözhet [31]. A cél nemcsak a szabályok optimalizálása, hanem a szabálybázis optimális méretének automatikus meghatározása is. Ebben az esetben a bakteriális mutáció és a génátadás okozhatják a baktériumok hosszának megváltozását.

A bakteriális mutáció a következőképpen történik: miután az adott baktérium klónjai létrejöttek, az egyes klónokban a mutáció során három lehetőség közül lehet véletlenszerűen választani. Az egyik lehetőség az éppen megváltoztatandó szabály törlése a klónból, a másik lehetőség a

kijelölt szabály paramétereinek véletlenszerű módosítása (ez az eredeti bakteriális mutációnak megfelelő lehetőség), a harmadik pedig a kijelölt szabály paramétereinek módosítása és ezzel egyidejűleg egy új szabály véletlenszerű létrehozása.

A génátadás végrehajtásakor két lehetőség közül lehet véletlenszerűen választani: a forrásbaktériumtól kapott szabály vagy felülír egy szabályt a célbaktériumban, vagy pedig új szabályként hozzáadódik a célbaktériumhoz. Az ily módon továbbfejlesztett bakteriális mutációs és génátadási operátorok lehetővé teszik a baktériumok hosszának növekedését illetve csökkenését. Az egyedek olyan kritérium alapján értékelődnek ki az operátorokban, amely nemcsak a szabálybázis által számított approximációs hibát veszi figyelembe, hanem a szabálybázis méretét is. Több szabály ugyanis általában kedvezőbb hibát ad, viszont növeli a modell komplexitását, ezért cél a minél kevesebb szabály elérése is. A Bayes-i információs kritérium segítségével mindkét célt, azaz a minél kisebb hiba és minél kevesebb szabály elérését, egy összefüggésben lehet megfogalmazni:

$$BIC = m \cdot \ln(MSE) + n \cdot \ln(m), \quad (12.25)$$

ahol m a tanítóminták száma, n pedig a szabályok száma. A Levenberg-Marquardt algoritmus a 12.3 alfejezetben leírt módon kerül alkalmazásra, ez a lépés nem változtatja meg a baktérium hosszát.

12.4.2. Az algoritmus alkalmazása

Szimulációs vizsgálatokat végeztünk, amelyekben a bakteriális memetikus algoritmust hasonlítottuk össze a bakteriális evolúciós algoritmussal az irodalomban található referenciaproblémákon. Az előző alfejezetekben szereplő referenciaproblémákat alkalmaztuk, és a módszert olyan bakteriális evolúciós algoritmussal hasonlítottuk össze, mely minden tekintetben megegyezik a bakteriális memetikus algoritmussal, kivéve természetesen, hogy nem tartalmazza a Levenberg-Marquardt lépést.

Rögzített szabálysám

Először a rögzített szabálysámú változattal végeztünk szimulációs futtatásokat, a szabályok száma 3. A pH probléma esetén a tanítóminták száma 101, az ICT problémánál 110, míg a hatváltozós függvény esetén 200. A tesztelési minták száma megegyezik a tanítóminták számával. Az algoritmusokban 10 egyedet használtunk, a klónok száma 8, az infekciók száma pedig 4. A Levenberg-Marquardt módszer 10 iterációs lépést használ,

12.6. táblázat. MSE, MSRE és MREP átlagértékek a tanító és a tesztelési mintákra minden probléma esetén a BEA használatával

hibadef.	pH	ICT	6 dim.
MSE	$1,5 \cdot 10^{-2}$	$5,0 \cdot 10^{-1}$	$3,4 \cdot 10^1$
MSRE	$2,8 \cdot 10^3$	$1,1 \cdot 10^{14}$	$6,8 \cdot 10^{-1}$
MREP	$2 \cdot 10^2$	$2,4 \cdot 10^8$	$7,2 \cdot 10^1$
MSE-teszt	$7,4 \cdot 10^{-3}$	$6,0 \cdot 10^{-1}$	$3,5 \cdot 10^1$
MSRE-teszt	$1,1 \cdot 10^4$	$8,2 \cdot 10^{-2}$	$6,1 \cdot 10^{-1}$
MREP-teszt	$3,9 \cdot 10^2$	$2,5 \cdot 10^1$	$6,8 \cdot 10^1$

a 12.21 leállási feltétellel. A generációs szám 20, és mindkét algoritmus 10-szer futott mindegyik problémára. A vizsgálat során az MSE, MSRE, és MREP hibakritériumokat használtuk (ld. 12.24).

A Bakteriális Evolúciós Algoritmus (BEA) által 10 futtatás során kapott átlageredményeket mutatja a 12.6. táblázat, a Bakteriális Memetikus Algoritmus (BMA) által kapott átlageredmények pedig a 12.7. táblázatban láthatók. A táblázatokból leolvasható, hogy a BMA minden problémára jobb eredményeket ad az MSE kritérium alapján, mint a BEA. Az egyváltozós pH probléma esetén a legszembevetőbb a különbség, ahol is a BMA kb. 16-szor jobb eredményt (kisebb hibát) ad. Általában mindegyik hibakritérium szerint jobb eredményt ad a BMA, kivéve a relatív hibakritériumokat a pH problémára. Ezt majd később vizsgáljuk, amikor a legjobb egyedet elemezzük.

Az ICT probléma tanítókészletében van néhány nullához közeli minta, emiatt a relatív hiba igen magas, összehasonlítva a tesztelési készletre kapott hibákkal, ahol ilyen minták nem fordulnak elő.

Az átlagos hibák tanulmányozása is fontos, azonban lényegesebb elemezni a legjobb egyedek által adott eredményeket. A 12.8. táblázatban az MSE kritérium alapján az összes futtatás során kapott legjobb egyed eredményei láthatók. Mindegyik esetre a BMA adja a legjobb MSE értékű megoldást nemcsak a tanítóminták tekintetében, hanem a tesztelési minták vonatkozásában is. A 12.8. táblázatból látszik viszont az is, hogy a BEA jobb relatív hibaeredményt ad a pH problémára. Ez főleg annak köszönhető, hogy van néhány minta viszonylag alacsony értékkel (kb. 10^{-4}). Viszont ha a relatív hibakritérium alapján kapott legjobb egyedet hasonlítjuk össze akkor megállapítható, hogy ebből a szempontból is a BMA adja az egyértelműen jobb eredményt, ahogy az a 12.9. táblázatban látható. A táblázatok

12.7. táblázat. MSE, MSRE és MREP átlagértékek a tanító és a tesztelési mintákra minden probléma esetén a BMA használatával

hibadef.	pH	ICT	6 dim.
MSE	$8,9 \cdot 10^{-4}$	$1,4 \cdot 10^{-1}$	$3,0 \cdot 10^1$
MSRE	$4,9 \cdot 10^3$	$2,3 \cdot 10^{13}$	$5,9 \cdot 10^{-1}$
MREP	$5,8 \cdot 10^2$	$9,8 \cdot 10^7$	$6,6 \cdot 10^1$
MSE-teszt	$1,0 \cdot 10^{-3}$	$9,9 \cdot 10^{-2}$	$3,3 \cdot 10^1$
MSRE-teszt	$1,9 \cdot 10^4$	$1,3 \cdot 10^{-2}$	$5,5 \cdot 10^{-1}$
MREP-teszt	$1,2 \cdot 10^3$	9,3	$6,4 \cdot 10^1$

12.8. táblázat. Az MSE kritérium alapján az összes futtatás során talált legjobb egyed

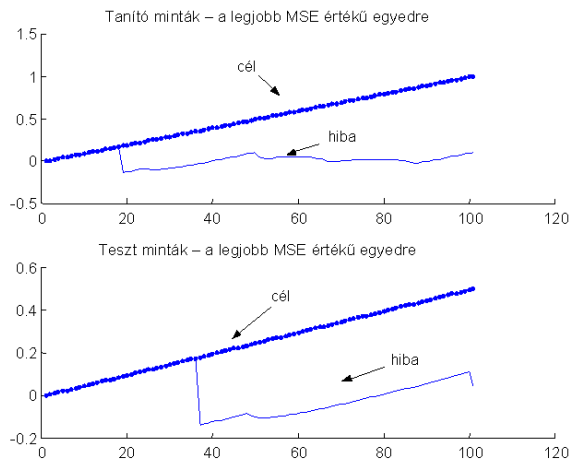
hibadef.	BEA	BMA	probléma
MSE	$4,5 \cdot 10^{-5}$	$1,5 \cdot 10^{-5}$	pH
MSRE	$2,1 \cdot 10^{-1}$	$3 \cdot 10^2$	pH
MREP	$2,8 \cdot 10^1$	$1,7 \cdot 10^2$	pH
MSE-teszt	$7,6 \cdot 10^{-3}$	$3 \cdot 10^{-5}$	pH
MSRE-teszt	$4,2 \cdot 10^{-1}$	$1,2 \cdot 10^3$	pH
MREP-teszt	$5,1 \cdot 10^1$	$3,5 \cdot 10^2$	pH
MSE	$3,4 \cdot 10^{-1}$	$8,5 \cdot 10^{-2}$	ICT
MSRE	$4,7 \cdot 10^{13}$	$3,6 \cdot 10^{13}$	ICT
MREP	$1,4 \cdot 10^8$	$1,7 \cdot 10^7$	ICT
MSE-teszt	$2,0 \cdot 10^{-1}$	$2,0 \cdot 10^{-2}$	ICT
MSRE-teszt	$2,6 \cdot 10^{-1}$	$2,7 \cdot 10^{-3}$	ICT
MREP-teszt	$1,5 \cdot 10^1$	4,4	ICT
MSE	$2,5 \cdot 10^1$	$2,0 \cdot 10^1$	6 dim.
MSRE	$5,4 \cdot 10^{-1}$	$4,4 \cdot 10^{-1}$	6 dim.
MREP	$6,3 \cdot 10^1$	$5,4 \cdot 10^1$	6 dim.
MSE-teszt	$2,6 \cdot 10^1$	$2,3 \cdot 10^1$	6 dim.
MSRE-teszt	$4,5 \cdot 10^{-1}$	$4,3 \cdot 10^{-1}$	6 dim.
MREP-teszt	$5,6 \cdot 10^1$	$5,6 \cdot 10^1$	6 dim.

12.9. táblázat. Az MREP kritérium alapján az összes futtatás során talált legjobb egyed

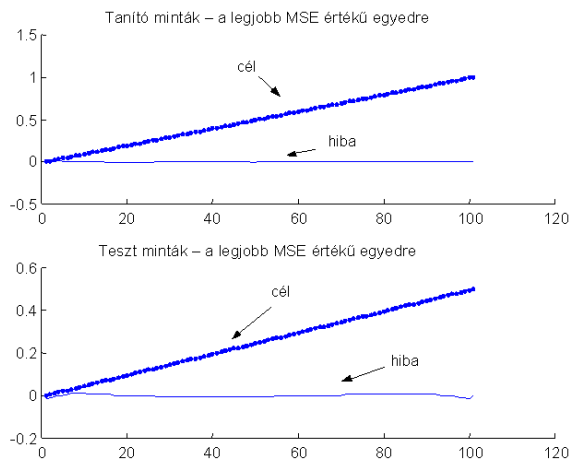
hibadef.	BEA	BMA	probléma
MSE	$1,5 \cdot 10^{-2}$	$8 \cdot 10^{-4}$	pH
MSRE	$1,6 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	pH
MREP	$2,6 \cdot 10^1$	$1,5 \cdot 10^1$	pH
MSE-teszt	$2,3 \cdot 10^{-3}$	$1,3 \cdot 10^{-3}$	pH
MSRE-teszt	$2,8 \cdot 10^{-1}$	$2 \cdot 10^{-1}$	pH
MREP-teszt	$3,5 \cdot 10^1$	$2,8 \cdot 10^1$	pH
MSE	$3,5 \cdot 10^{-1}$	$8,5 \cdot 10^{-2}$	ICT
MSRE	$8,4 \cdot 10^{13}$	$3,6 \cdot 10^{13}$	ICT
MREP	$2,0 \cdot 10^7$	$1,7 \cdot 10^7$	ICT
MSE-teszt	$1,8 \cdot 10^{-1}$	$2,0 \cdot 10^{-2}$	ICT
MSRE-teszt	$2,3 \cdot 10^{-2}$	$2,7 \cdot 10^{-3}$	ICT
MREP-teszt	$1,4 \cdot 10^1$	4,4	ICT
MSE	$2,5 \cdot 10^1$	$2,0 \cdot 10^1$	6 dim.
MSRE	$5,4 \cdot 10^{-1}$	$4,4 \cdot 10^{-1}$	6 dim.
MREP	$6,3 \cdot 10^1$	$5,4 \cdot 10^1$	6 dim.
MSE-teszt	$2,6 \cdot 10^1$	$2,3 \cdot 10^1$	6 dim.
MSRE-teszt	$4,5 \cdot 10^{-1}$	$4,3 \cdot 10^{-1}$	6 dim.
MREP-teszt	$5,6 \cdot 10^1$	$5,6 \cdot 10^1$	6 dim.

egyértelműen azt mutatják, hogy a BMA jobb eredményeket ad, mint a BEA. A következő néhány ábra ugyanezt támasztja alá. A 12.16. és 12.17. ábra a legjobb MSE értékű egyed cél- és a hibaértékeit mutatja az egyes mintákra a pH probléma esetén BEA illetve BMA használatával. A BMA-ra vonatkozó hiba teljesen sima, látható, hogy a módszer a mintakészlet minden elemére milyen kicsi hibát ad.

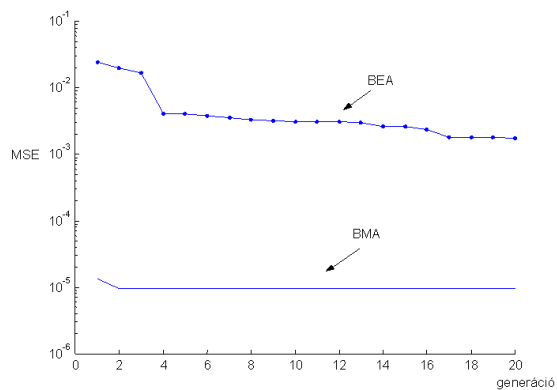
Magasabb dimenziószámú problémára a hiba is magasabb. Ennek oka, hogy összesen csak 3 szabályt tartalmaz a szabálybázis mindegyik problémára, ezért a hiba nagyobb a bonyolultabb problémák esetén. A 12.18. 12.19. és 12.20. ábrák az MSE értékek alakulását mutatják egy-egy futtatásra a különböző problémákra. Ezek alapján is jól látszik a memetikus megközelítésben a Levenberg-Marquardt lépés által okozott javulás. A kapott optimális szabályokat is bemutatjuk az ICT problémára. A tagsági függvényekben szereplő trapézok paramétereinek nem szükségszerűen kell az adott változó intervallumán belül elhelyezkedniük, a trapéz kinyúlhat a változó intervallumából.



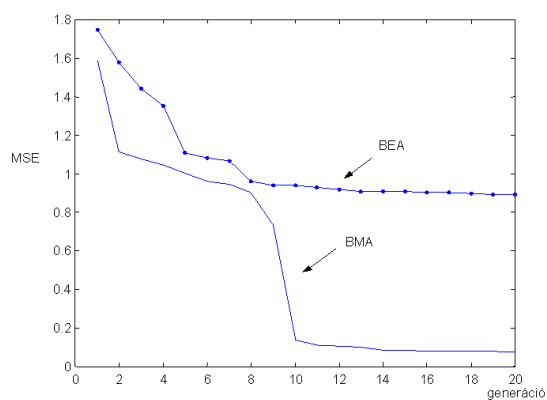
12.16. ábra. A legjobb MSE értékű egyed cél- és hibaértékei az egyes mintákra a pH probléma esetén BEA használatával



12.17. ábra. A legjobb MSE értékű egyed cél- és hibaértékei az egyes mintákra a pH probléma esetén BMA használatával



12.18. ábra. MSE alakulása a pH probléma esetén



12.19. ábra. MSE alakulása az ICT probléma esetén

A BEA által kapott optimális szabályok az ICT problémára:

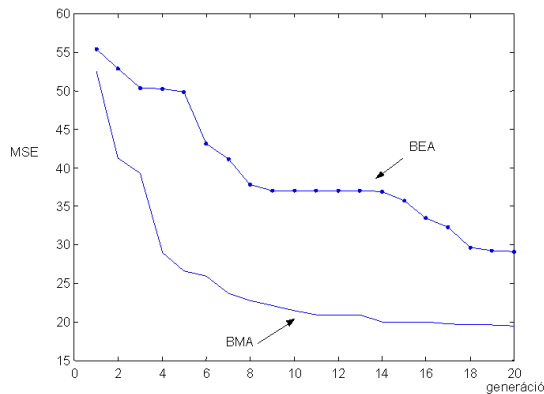
$$\begin{aligned} R_1: & \text{Ha } x_1 = [-0,027427; 0,013643; 0,027407; 0,27058] \text{ és} \\ & x_2 = [-0,072697; 0,1739; 0,43876; 0,88877] \text{ akkor} \\ & y = [1,9286; 1,9644; 2,2663; 2,8419] \\ R_2: & \text{Ha } x_1 = [0,023605; 0,82814; 0,83875; 0,91526] \text{ és} \\ & x_2 = [0,039025; 0,37465; 0,76291; 0,94203] \text{ akkor} \\ & y = [0,083984; 0,599; 0,83059; 1,6556] \\ R_3: & \text{Ha } x_1 = [0,0055574; 0,14652; 0,57419; 0,57709] \text{ és} \\ & x_2 = [-0,090017; 0,16984; 0,88549; 0,89452] \text{ akkor} \\ & y = [1,5032; 2,4553; 2,6697; 3,2334] \end{aligned}$$

A BMA által kapott optimális szabályok az ICT problémára:

$$\begin{aligned} R_1: & \text{Ha } x_1 = [-0,05557; 0,096831; 0,096831; 0,61764] \text{ és} \\ & x_2 = [-0,38741; 0,15828; 0,42377; 0,75708] \text{ akkor} \\ & y = [1,9286; 2,5199; 2,841; 3,6312] \\ R_2: & \text{Ha } x_1 = [-0,13014; 0,49711; 0,79235; 0,99016] \text{ és} \\ & x_2 = [0,30314; 0,72457; 0,76876; 1,114] \text{ akkor} \\ & y = [0,01816; 0,38369; 1,0447; 1,3301] \\ R_3: & \text{Ha } x_1 = [0,11687; 0,62462; 0,69643; 0,80899] \text{ és} \\ & x_2 = [-0,083077; 0,22048; 0,24413; 0,70602] \text{ akkor} \\ & y = [1,0631; 1,6966; 1,9513; 2,1227] \end{aligned}$$

Változó szabálysám

Az algoritmus másik, általánosabb változata automatikusan beállítja a szabálybázis optimális méretét is. Ezzel a változattal is végeztünk szimulációkat, melynek eredményei a 12.10. és 12.11. táblázatban szerepelnek. A szimuláció paraméterei megegyeznek az előző szakaszban használt paraméterekkel, viszont a szabálysám nincs rögzítve. A megengedett maximális szabálysám 10. A táblázatokban a kapott szabályok száma is fel van tüntetve. A kapott eredmények az előző szakaszhoz hasonlóan azt mutatják, hogy a BMA lényegesen jobb eredményt ad. Ebben a változatban a nagyobb dimenziószámú problémák esetén is viszonylag nagy a javulás, mert itt nem 3 szabály a megengedett maximum, hanem a szabálysám fokozatosan növekedve hozzáidomul a probléma bonyolultságához.



12.20. ábra. MSE alakulása a 6 változós probléma esetén

12.10. táblázat. Hiba átlagértékek változó szabályszám esetén

hibadef.	pH		ICT		6 dim.	
	BEA	BMA	BEA	BMA	BEA	BMA
MSE	$6,28 \cdot 10^{-3}$	$2,3 \cdot 10^{-6}$	$8,69 \cdot 10^{-1}$	$3,67 \cdot 10^{-2}$	3,21	1,29
MSRE	$4,06 \cdot 10^4$	$1,02 \cdot 10^1$	$5,63 \cdot 10^{13}$	$1,80 \cdot 10^{13}$	$6,20 \cdot 10^{-2}$	$3,17 \cdot 10^{-2}$
MREP	$2,05 \cdot 10^3$	$2,69 \cdot 10^1$	$1,77 \cdot 10^8$	$1,08 \cdot 10^8$	$1,86 \cdot 10^1$	$1,21 \cdot 10^1$
MSE-teszt	$7,76 \cdot 10^{-3}$	$2,08 \cdot 10^{-6}$	1,30	$1,12 \cdot 10^{-2}$	4,29	2,22
MSRE-teszt	$1,63 \cdot 10^5$	$4,18 \cdot 10^1$	$1,92 \cdot 10^{-1}$	$1,62 \cdot 10^{-3}$	$6,50 \cdot 10^{-2}$	$3,37 \cdot 10^{-2}$
MREP-teszt	$4,12 \cdot 10^3$	$5,46 \cdot 10^1$	$2,56 \cdot 10^1$	3,04	$1,91 \cdot 10^1$	$1,32 \cdot 10^1$
átlagos szabályszám	4,90	7,40	2,20	5,00	6,50	7,30

12.11. táblázat. Legjobb egyedek változó szabályszám esetén

hibadef.	pH		ICT		6 dim.	
	BEA	BMA	BEA	BMA	BEA	BMA
MSE	$2,73 \cdot 10^{-3}$	$4,7 \cdot 10^{-7}$	$8,42 \cdot 10^{-1}$	$1,04 \cdot 10^{-2}$	2,07	$4,10 \cdot 10^{-1}$
MSRE	$3,71 \cdot 10^4$	3,63	$5,32 \cdot 10^{13}$	$7,09 \cdot 10^{12}$	$4,78 \cdot 10^{-2}$	$9,56 \cdot 10^{-3}$
MREP	$1,97 \cdot 10^3$	$1,92 \cdot 10^1$	$2,03 \cdot 10^8$	$6,39 \cdot 10^7$	$1,59 \cdot 10^1$	7,06
MSE-teszt	$5,12 \cdot 10^{-3}$	$6,07 \cdot 10^{-7}$	1,26	$2,60 \cdot 10^{-3}$	2,66	$9,9 \cdot 10^{-1}$
MSRE-teszt	$1,48 \cdot 10^5$	$1,53 \cdot 10^1$	$1,87 \cdot 10^{-1}$	$3,74 \cdot 10^{-4}$	$4,28 \cdot 10^{-2}$	$1,5 \cdot 10^{-2}$
MREP-teszt	$3,96 \cdot 10^3$	$3,93 \cdot 10^1$	$2,34 \cdot 10^1$	1,48	$1,47 \cdot 10^1$	9,28
szabályszám	9	8	2	5	7	10

12.5. Bakteriális programozás alkalmazása B-spline neurális hálózatok identifikációjára

Ahogy a 11.3. fejezetben láttuk, a B-spline típusú neurális hálózatok tervezési folyamata 6 lépésből áll. Az utolsó két fázis nemlineáris legkisebb négyzetes problémának tekinthető és ennél fogva komplett felügyelt tanuló algoritmust lehet a megoldására alkalmazni, Levenberg-Marquardt módszerrel a feladat kiválóan megoldható [151]. Az első négy tervezési fázis összetett kombinatorikai probléma, melynek megoldására különböző konstruktív algoritmusokat javasoltak, például az ASMOD (Adaptive Spline Modelling of Observed Data) algoritmust [196], a MARS (Multivariate Adaptive Regression Splines) algoritmust [62], és a LOLIMOT (Local Linear Model Trees) algoritmust [141]. Ezeket követően 2003-ban alkalmazták a genetikus programozást is a feladat megoldására, mellyel az előbbi módszereknél kedvezőbb eredményt sikerült elérni [34]. Ebben a fejezetben a bakteriális programozás alkalmazását ismertetjük a probléma megoldására.

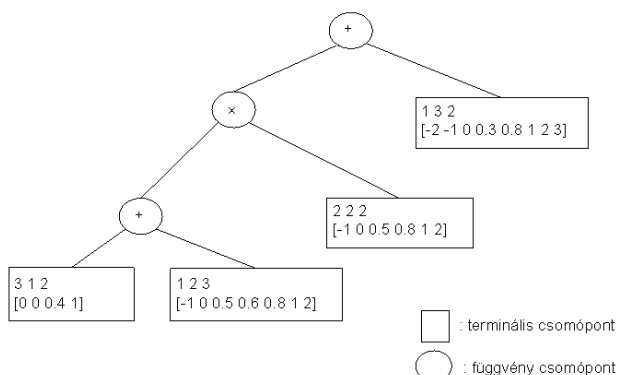
12.5.1. A javasolt módszer

A bakteriális programozás a genetikus programozás (10.2. fejezet) és a bakteriális evolúciós algoritmusok ötvözéséből keletkezik [32, 33, 22]. Az egyedek csakúgy mint a genetikus programozás esetén kifejezésfával adottak, viszont a bakteriális programozás az evolúciós folyamatban nem az irodalomból ismert genetikus operátorokat (keresztelés, mutáció), hanem a bakteriális algoritmus operátorait (bakteriális mutáció, génátadás) használja.

A kódolási elrendezés

A B-spline neurális hálózatok hierarchikus struktúrája jobban ábrázolható fával, mint egy kromoszómába kódolt stringgel. Emiatt a kromoszómát használó genetikus illetve bakteriális algoritmus helyett a genetikus programozásban megismert fastruktúra alkalmasabb a hálózat struktúrájának ábrázolására.

A 12.21. ábrán egy példa kifejezésfa látható B-spline neurális hálózatra [34]. A B-spline hálózat tervezésekor a következő műveletek végrehajtására lehet szükség: almodellek összeadására (+), kisebb dimenziójú almodellekből nagyobb dimenziójú almodellek létrehozására ($\times$), és nagyobb dimenziójú almodell szétbontására kisebb dimenziójú almodellekre ($/$). Ezek



12.21. ábra. Példa B-spline neurális hálózat kifejezésfájára

a műveletek alkotják a B-spline hálózatra definiált függvények halmazát, melyek a kifejezésfa függvény csomópontjaiban előfordulhatnak. A fa terminális szimbólumai nem csak a dimenzió azonosítóját tartalmazzák, hanem az adott dimenzióhoz tartozó változón definiált spline görbék rendjét, az azokhoz tartozó belső csomópontok számát illetve a belső csomópontok elhelyezkedését is. A 12.21. ábrán látható hálózat esetén például a fa bal szélső levele azt jelzi, hogy a bemeneti változó a levélhez tartozó almodell esetén 3, a spline görbe rendje 1 és két belső csomópont van a változóhoz tartozó koordinátatengelyen, amelyek helye a 0 és a 0,4-es pozícióban van. Az ábrán szereplő modell kimenete 3 almodell kimenetének összeadásaként számítható, a következő módon:

$$y(\mathbf{X}) = f(\mathbf{X}_3 \times \mathbf{X}_2) + f(\mathbf{X}_1 \times \mathbf{X}_2) + f(\mathbf{X}_1),$$

ahol $\mathbf{X}_i$ az i . bemeneti változót jelöli, azaz a modell 2 kétváltozós és 1 egyváltozós almodell összegeként adódik. A függvényeket és terminálisokat körültekintően kell definiálni, hogy bármilyen értéket kaphassanak argumentumként, amit a lejjebb található részfák függvényei és terminálisai visszaadnak.

Ha egy fához tartozó modell komplexitása nagyobb, mint a tanítóhalmazban lévő minták száma, akkor az egyed törlése helyett érdemes inkább a struktúráján olyan változtatást eszközölni, hogy az egyed továbbra is

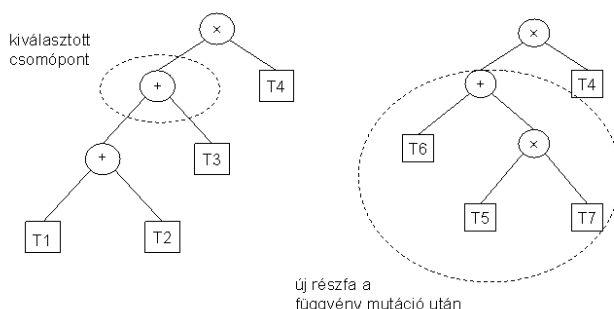
részt vehessen az evolúciós folyamatban. Annak érdekében, hogy érvényes egyedeket kapjunk, az egyed kiértékelése során áttekintjük a kifejezésfát, és minden csomópontban kiértékeljük a csomópont alatti részfa komplexitását. Ha ez az érték nagyobb, mint a minták száma, akkor a komplexitást a következőképpen csökkentjük: a tenzor szorzat ($\times$) függvényt az összeadás (+) függvénnyel helyettesítjük, ha a csomópont tenzor-szorzatfüggvény; ha pedig a csomópont összeadásfüggvény, akkor azt a csomópont alatt található legkisebb komplexitású ághoz tartozó részfával helyettesítjük. Ezt a két lépést ismételjük addig, amíg az egyed szintaktikailag érvényes nem lesz, azaz amíg olyan fát nem reprezentál, amelyik a feladat szempontjából értelmezhető. Ezen lépések során néhány terminális és függvény csomópont eltűnhet a fából.

Az evolúciós folyamat

Az evolúciós folyamat megegyezik a bakteriális evolúciós algoritmusnál alkalmazottal, itt azonban az egyedek fagráfokat jelentenek. A kezdeti populációt véletlenszerűen létrehozott fák alkotják. Az egyedek száma N_{egyed} . Ezután következik az evolúciós ciklus, mely a bakteriális mutációt és a génátadást alkalmazza. A maximális generációs szám (N_{gen}) elérésekor a folyamat véget ér.

Bakteriális mutáció

A bakteriális mutációt a populáció minden egyedére egyenként alkalmazzuk. Az egyed $N_{klón}$ példányban lemásoljuk (klónok). A baktérium egy véletlenszerűen kiválasztott részét megváltoztatjuk az így nyert klónokban (mutáció), az eredeti baktériumban viszont nem. Jelen esetben azonban az egyedek kétféle típusú csomópontot tartalmaznak, ezért a klónokban kétféle mutáció megengedett. Az egyik a függvény mutáció, amelynek során a kijelölt függvény-csomópont-hoz tartozó részfat egy új, véletlenszerűen létrehozott részfával cseréljük le. A másik a terminális mutáció, amelynél az adott terminálist változtatjuk meg valamilyen módon. A klónokban végrehajtott mutáció után az összes klónt és az eredeti baktériumot kiértékeljük, és a legjobb közülük a megváltoztatott részt a többi egyednek adja át, illetve, ha az eredeti baktérium maradt a legjobb, akkor ez adja át a szóban forgó részt a klónoknak. Ezt a folyamatot, mely a mutáció – kiértékelés – kiválasztás – behelyettesítés lépéssorozatot jelent, ismételjük addig, amíg a kifejezésfa mindegyik részét egyszer ki nem választottuk. Ha a fa egy részének mutációja során egy új részfa keletkezett, annak csomópont-



12.22. ábra. A függvény mutáció művelete

jait ezen a lépéssorozatot belül már nem választjuk ki többször mutációra (csak majd a következő generációban). Amikor az egész fával végeztünk, kiválasztjuk a legjobb egyedet, a többi $N_{klón}$ egyed pedig megszüntetjük.

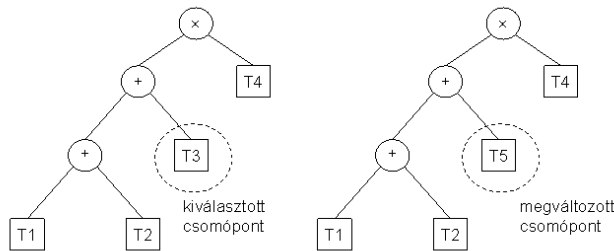
A 12.22. ábra a függvény típusú mutációt illusztrálja. Ilyenkor a kiválasztott függvény-csomópont alatt új részfa jön létre. A 12.23. ábrán a terminális típusú mutáció látható, mely a terminálisban szereplő információkat változtatja meg véletlenszerűen. B-spline neurális hálózatok esetén a terminális mutáció a következő hatféle lehet:

1. a teljes terminális csomópont lecserélése
2. a változó azonosítójának megváltoztatása
3. a spline görbe rendjének megváltoztatása
4. egy belső csomópont áthelyezése
5. N véletlenszerű belső csomópont hozzáadása ($N = 5$)
6. N belső csomópont eltávolítása (csomópontok hiánya esetén nincs végrehajtás)

A terminális mutációs arányt a $p_{mut-terminal} = [\%1, \%2, \%3, \%4, \%5, \%6]$ vektorral definiáljuk, ahol $\%i$ az i . típusú terminális mutáció valószínűségét jelenti.

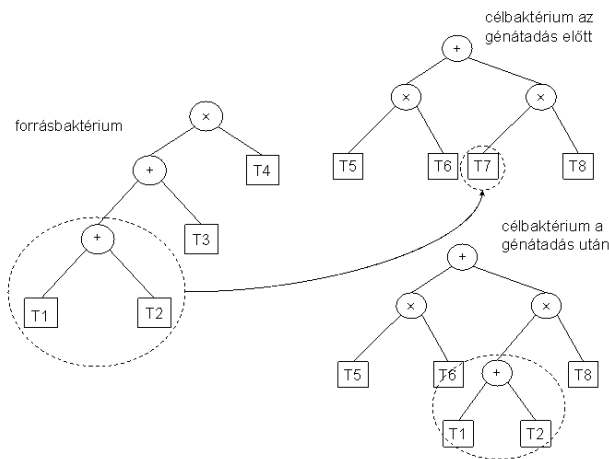
Génátadás

A génátadás a korábban megismert módon zajlik. A populáció kettéosztása után a „jó” egyedek közül kiválasztunk egy forrásbaktériumot, és



12.23. ábra. A terminális mutáció művelete

a „rosszak” közül egy célbaktériumot. A forrásbaktérium fájából kiválasztunk véletlenszerűen egy részfát, és ez a részfa felülírja a célbaktérium egy véletlenszerűen kiválasztott részfáját. A részfa bármekkora lehet, akár mindössze egyetlen terminális csomópontot tartalmazó részfa is. A génátadási folyamat a 12.24. ábrán látható. Ezt a három lépésből álló folyamatot (populáció kettéosztás; forrás- és célbaktérium kiválasztás; génátadás) N_{inf} -szer ismétljük.



12.24. ábra. A génátadás művelete

A baktériumok kiértékelése

A baktériumokat többféle különböző kritérium alapján lehet kiértékelni. Az előző alfejezetekben bemutattunk néhányat ezek közül. Legmegfelelőbb jelen esetben a 12.4. fejezetben alkalmazott Bayes-i információs kritérium, mely az egyed által adott hibán kívül a komplexitást is figyelembe veszi. Ahogy már korábban is szerepelt (12.25) a következőképpen számítható:

$$BIC = m \cdot \ln(MSE) + n \cdot \ln(m), \quad (12.26)$$

ahol m a tanítóminták száma, n pedig a modell komplexitása, azaz a bázisfüggvények száma.

12.5.2. A módszer alkalmazása

Szimulációs vizsgálatokat végeztünk, amelyekben az eljárást a genetikus programozással hasonlítottuk össze az irodalomban található referenciaproblémákon. Az előző alfejezetekben megismert referenciaproblémákat alkalmaztuk ismét.

A bakteriális programozás paramétereinek meghatározása

Az első szimuláció célja annak meghatározása, hogy mik a bakteriális programozás legmegfelelőbb paraméterértékei. A módszerben használt paraméterek az egyedek száma (N_{egyed}), a klónok száma ($N_{klón}$), az infekciók száma (N_{inf}) és a generációk száma (N_{gen}). Az optimális paraméter értékek meghatározása a pH probléma alapján történik.

A szimuláció 10-szer futott le, melyek során a Bayes-i információs kritérium (BIC), az átlagos négyzetes hiba (MSE), az átlagos négyzetes relatív hiba (MSRE), és az átlagos relatív százalékos hiba (MREP) értékek kerültek meghatározásra.

A relatív hibakritériumok hasznosak lehetnek akkor, amikor a kimenet széles tartományt fog át, mert ezek a kritériumok minden egyes mintához tartozó hibát a modell kimenetéhez viszonyítva vesznek figyelembe. Mindemellett viszont a BIC és MSE értékek fontosabbak mert az egyedek az evolúciós folyamat során a BIC kritérium alapján értékelődnek ki, és a BIC az MSE kritériumot foglalja magába, nem pedig a relatív hibát.

A példában az egyedek száma és a generációk száma is 20. Először a klónok száma állítható, 5, 10 és 15-ös értéket vizsgálva, az infekciók száma 5-re rögzített. A második esetben az infekciók száma állítható 5, 10 és 15-ös értéket vizsgálva, 8 klónt használva.

12.12. táblázat. A BIC, MSE, MSRE, és MREP átlagértékei és a modellbonyolultság az $N_{klón}$ paraméter értékét változtatva

$N_{klón}$	5	10	15
BIC	-1695,9	-1834,3	-1913
MSE	$3,6 \cdot 10^{-8}$	$6,9 \cdot 10^{-9}$	$2,3 \cdot 10^{-9}$
MSRE	$5,8 \cdot 10^{-2}$	$2,9 \cdot 10^{-3}$	$1,2 \cdot 10^{-3}$
MREP	1,2	$2,7 \cdot 10^{-1}$	$1,7 \cdot 10^{-1}$
komplexitás	57,8	73	81

12.13. táblázat. A BIC, MSE, MSRE, és MREP átlagértékei és a modellbonyolultság az N_{inf} paraméter értékét változtatva

N_{inf}	5	10	15
BIC	-1823,9	-1792,8	-1934,3
MSE	$8,4 \cdot 10^{-9}$	$4,2 \cdot 10^{-8}$	$2,2 \cdot 10^{-9}$
MSRE	$7,1 \cdot 10^{-3}$	$7,4 \cdot 10^{-2}$	$3,4 \cdot 10^{-3}$
MREP	$4,1 \cdot 10^{-1}$	$9,1 \cdot 10^{-1}$	$3,3 \cdot 10^{-1}$
komplexitás	75,9	76,2	87,4

A 12.12. táblázatban közölt eredmények azt mutatják, hogy minél több a klón, annál jobb a kimenet pontossága. Viszont több klón több számítási költséget is jelent. Ezért a cél ilyenkor az optimális egyensúly megtalálása, és a túl sok klón használatának elkerülése.

A 12.13. táblázatot elemezve megállapítható, hogy az itt kapott eredmények nem különböznek annyira egymástól, mint az előző esetben, a klónok számának meghatározásakor. Több infekció alkalmazása a populációt lokális optimumba vezetheti a korai konvergencia miatt. Kisebb értékű N_{inf} jobb eredményt adhat és kevesebb számítási teljesítményt igényel.

A bakteriális és genetikus programozás összehasonlítása

Ennek a résznek a célja, hogy a genetikus programozás (GP) és a bakteriális programozás (BP) által referenciaproblémák megoldására kapott eredményeket bemutassa. Ahogy az előző részben, itt is 10 futtatás történik, és a BIC, MSE, MSRE és MREP értékekre kapott átlageredmények képezik a vizsgálódás tárgyát. A tanítóminták száma a pH problémánál 101, az ICT-nél 110, a 6 változós függvényénél pedig 200.

12.14. táblázat. *A két módszerben (genetikus és bakteriális) használt paraméter értékek*

Paraméterek	GP	BP
$N_{klón}$	-	8
N_{inf}	-	5
N_{egyed}	160	20
N_{gen}	20	20
keresztelési arány	a populáció 50%-a	-
mutációs arány	0,8	-

Annak érdekében, hogy a két módszer esetén azonos legyen a számítási komplexitás, és ezáltal az összehasonlítás igazságos legyen, az algoritmusok a 12.14. táblázatban szereplő paraméter értékeket használják.

A terminális mutációs arány mindkét algoritmus esetén: $p_{mut-terminal} = [5\%, 10\%, 5\%, 10\%, 60\%, 10\%]$.

A paraméter-értékek táblázata alapján látható, hogy a populáció a BP módszer esetén sokkal kisebb. A 8 klón használata viszont hasonló számítási komplexitást jelent, mint a GP módszer esetén alkalmazott 160 egyedből álló populáció, mivel a bakteriális mutációban mind a 20 baktériumnak 8 klónja jön létre. A BP megközelítés egyik előnye, hogy nem szükséges ilyen nagy populációt kezelni; 20 baktérium evolúciós folyamata elég a két módszer összehasonlításához.

A 12.15. táblázatban a pH problémára kapott átlagértékek láthatók. Ebben az esetben a GP és a BP megközelítés hasonló eredményt ad. Viszont a BP átlagban kisebb komplexitású modellt hoz létre. A 12.16. táblázatban a BP és a GP által létrehozott legjobb egyedhez tartozó modell struktúrája látható. Ezekből az eredményekből megállapítható, hogy a két módszer hasonló végső modelleket hozott létre. Ennek oka, hogy a pH probléma egy egyváltozós feladat, ezért a B-spline neurális hálózat struktúrája nem túl bonyolult, és ennél fogva mindkét módszer könnyen megoldja ezt a modelltervezési feladatot.

Az ICT problémára kapott eredmények a 12.17. és a 12.18. táblázatban találhatók. A BP jobb eredményt ad nemcsak az átlagértékekre vonatkozóan, hanem a legjobb egyedre (a legkisebb BIC értékű egyedre) vizsgálva is. Bár a GP a legjobb egyedre kisebb relatív hibaértékeket ad, az evolúciós folyamatot mindkét algoritmus esetén a BIC kritérium vezérli (ami az MSE-t is magába foglalja), ezért a BIC és MSE kritériumok fontosabbak. Ezen

12.15. táblázat. A BIC, MSE, MSRE, és MREP átlagértékei és a modellbonyolultság a pH problémára

	GP	BP
BIC	-1784,6	-1786,7
MSE	$1,1 \cdot 10^{-8}$	$1,3 \cdot 10^{-8}$
MSRE	$1,3 \cdot 10^{-2}$	$2,6 \cdot 10^{-2}$
MREP	$6,1 \cdot 10^{-1}$	$6,9 \cdot 10^{-1}$
komplexitás	82,6	72,4

12.16. táblázat. Az összes futtatás során talált legkisebb BIC értékű egyedhez tartozó modellstruktúra a pH probléma esetén

	GP	BP
almodellek	(1)	(1)
komplexitás	33	40
BIC	-1903,1	-1874,3
MSE	$1,4 \cdot 10^{-9}$	$1,8 \cdot 10^{-9}$
MSRE	$2,7 \cdot 10^{-3}$	$7,5 \cdot 10^{-5}$
MREP	$5,3 \cdot 10^{-1}$	$1,0 \cdot 10^{-1}$
W	3,3	3,6

kritériumok alapján a BP jobb eredményt nyújt.

A bakteriális megközelítés fő előnye a 6 változós problémára kapott eredmények elemzésekor válik világossá. Ezek az eredmények azt mutatják, hogy a bakteriális módszer jó teljesítményt nyújt nagyobb dimenziószámú

12.17. táblázat. A BIC, MSE, MSRE, és MREP átlagértékei és a modellbonyolultság az ICT problémára

	GP	BP
BIC	-1344,4	-1539,7
MSE	$2,5 \cdot 10^{-7}$	$8,6 \cdot 10^{-8}$
MSRE	$1,6 \cdot 10^5$	$2,1 \cdot 10^3$
MREP	1331,6	125,8
komplexitás	33,4	31,7

12.18. táblázat. Az összes futtatás során talált legkisebb BIC értékű egyedhez tartozó modellstruktúra az ICT probléma esetén

	GP	BP
almodellek	$(1 \times 2)(1)(2)$	$(2 \times 1)(1)$
komplexitás	106	106
BIC	-1533,9	-2048,3
MSE	$1,3 \cdot 10^{-8}$	$8,8 \cdot 10^{-11}$
MSRE	$1,6 \cdot 10^{-7}$	22,45
MREP	$1,3 \cdot 10^{-2}$	79,7
$ W $	686,7	$2,6 \cdot 10^7$

12.19. táblázat. A BIC, MSE, MSRE, és MREP átlagértékei és a modellbonyolultság a 6 változós problémára

	GP	BP
BIC	-380,1	-552,9
MSE	$2,0 \cdot 10^{-1}$	$2,7 \cdot 10^{-2}$
MSRE	$2,1 \cdot 10^{-3}$	$5,2 \cdot 10^{-4}$
MREP	2,1	0,98
komplexitás	38,8	44,6

problémák esetén. A 12.19. és a 12.20. táblázat alapján látható, hogy a BP jobb, mint a GP, mind az átlagértékek, mind pedig a legjobb egyedek vonatkozásában.

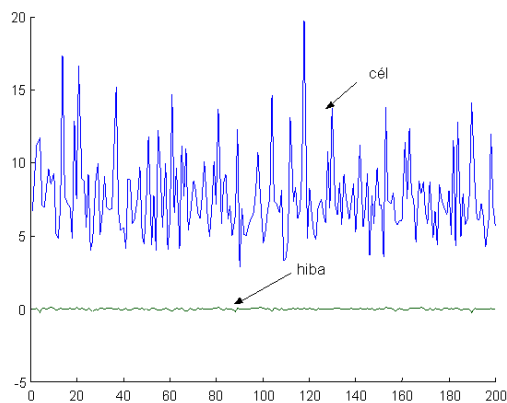
A 12.25. és a 12.26. ábrán a megkívánt kimenet (cél) és a hiba értékei láthatók az egyes mintákra vonatkozóan mindkét módszer által adott legjobb egyedek tekintve. Az ábrákat összehasonlítva megállapítható, hogy a bakteriális technika kisebb hibát ad.

Statisztikai elemzés

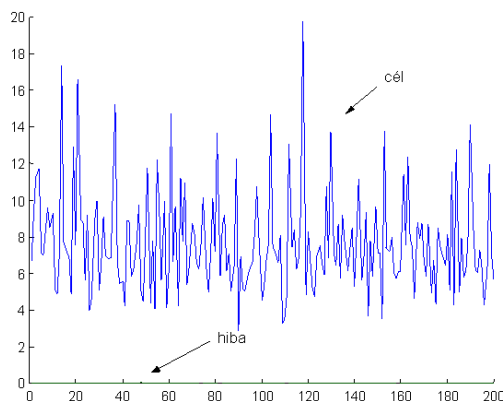
Az előzőekben bemutatott szimulációk alapján megállapítható, hogy a bakteriális programozás hatékonyabb, mint a genetikus programozás. Ennek oka a bakteriális megközelítésben alkalmazott operátorok természetének különbözősége. A baktériumok a keresési tér nagyobb részét képesek felfedezni a bakteriális mutációban alkalmazott hatékony klónok segítségével.

12.20. táblázat. Az összes futtatás során talált legkisebb BIC értékű egyedhez tartozó modellstruktúra a 6 változós probléma esetén

	GP	BP
almodellek	(5)(4)(2) (3 × 4)(5 × 3 × 6) (3 × 1)	(6 × 5)(5 × 2) (6 × 1)(3 × 6) (3 × 4)(1)
komplexitás	98	156
BIC	-593,2	-702
MSE	$3,8 \cdot 10^{-3}$	$4,8 \cdot 10^{-4}$
MSRE	$7,3 \cdot 10^{-5}$	$1,2 \cdot 10^{-5}$
MREP	$6,6 \cdot 10^{-1}$	$2,4 \cdot 10^{-1}$
W	423,8	128,4



12.25. ábra. Cél- és hibaértékek a 6 változós problémára GP használatával



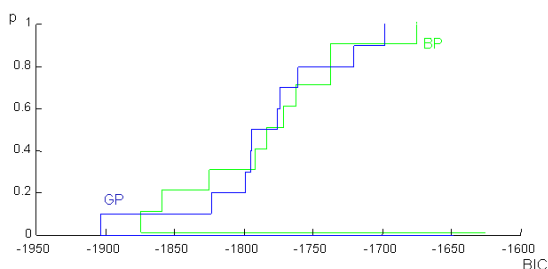
12.26. ábra. Cél- és hibaértékek a 6 változós problémára BP használatával

Ez a rész néhány hipotézis tesztet mutat be, amelyek a 12.21. táblázatban találhatók meg. Feltéve, hogy a hipotézis elvetésének szignifikancia szintje $\alpha = 5\%$, a nullhipotézisek konfidencia aránya 95%-ra adódik.

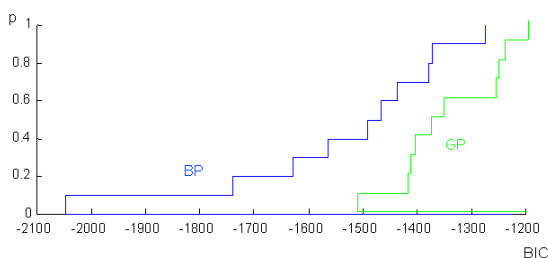
Az eredmények egyenlőségének becslésére az egyik legnépszerűbb két-mintás módszert, a Mann-Whitney tesztet alkalmaztuk [131]. Ez a teszt annak valószínűségét mutatja (p), hogy két különböző algoritmusból azonos értékű mintákat nyerünk. Annak kiderítésére, hogy a két algoritmusnak egyenlő mediánja van-e, a medián teszt készíthető el. Adott futásszám és az egyik algoritmus alapján kapott mintákból létrehozott rendezett összesítés esetén a medián teszt annak valószínűségére ad véleményt, hogy két populáció közötti medián különböző, kisebb vagy nagyobb-e. A 12.21. táblázatban mutatott eredmények azt jelzik, hogy a két algoritmus hasonló teljesítményű amikor az egyváltozós pH problémára keresnek megoldást, hiszen p értéke magas. Amikor viszont többváltozós feladatokat kezelnek, akkor látható, hogy lecsökken annak a valószínűsége, hogy hasonló eredményeket adnak, mivel a p érték alacsonyabb, mint az 5%-os elvetési határ. Ezenkívül a medián érték a bakteriális esetben alacsonyabb, ami azt jelenti, hogy a legtöbb esetben ez a módszer jobb kiértékelési eredményt ad. A 12.27. 12.28. és 12.29. ábra mindegyik problémára, mindkét módszerrel nyert tapasztalati valószínűségi eloszlásfüggvényeket mutat. Ezen ábrák alapján ugyanaz a következtetés vonható le, ami az előző részekben. Az

12.21. táblázat. A BP-re és a GP-re vonatkozó statisztikai következmények a Mann-Whitney és a medián teszt módszerekkel

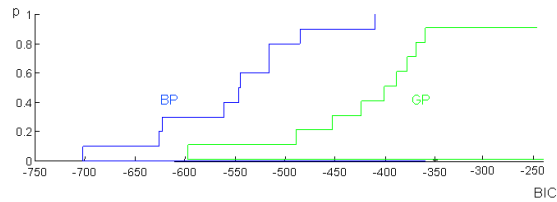
Probléma	Mann-Whitney teszt (p érték)	Medián teszt
pH	0,7624	a BP-re alacsonyabb medián, mint a GP-re
ICT	0,0156	a BP-re alacsonyabb medián, mint a GP-re
6 változós függvény	0,00194	a BP-re alacsonyabb medián, mint a GP-re



12.27. ábra. Tapasztalati valószínűségi eloszlásfüggvény a pH problémára



12.28. ábra. Tapasztalati valószínűségi eloszlásfüggvény az ICT problémára



12.29. ábra. *Tapasztalati valószínűségi eloszlásfüggvény a 6 változós problémára*

egyváltozós feladatra a két megközelítés hasonló eredményt ad. A kétváltozós ICT problémára a legjobb egyed BIC értékei kb. -2050 és -1750 között vannak a BP-vel, és kb. -1500 és -1420 között a GP-vel, ami azt jelenti, hogy a GP módszer gyengébb eredményt ad, mint a bakteriális technika. A 12.29. ábra a 6 változós problémára vonatkozik. Míg a GP esetben a legjobb egyed BIC értékei csak kb. -600 és -500 között vannak, addig a BP által létrehozott legjobb egyed BIC értékei kb. -700 és -630 közöttiek.

Ábrák jegyzéke

1.1. a) R_1, R_2, R_3 szabálybázis által generált hozzárendelés; b) Ezen hozzárendelés („fuzzy függvény”) α -vágatai	17
2.1. A „körülbelül 2” fogalmat reprezentáló különböző alakú fuzzy halmazok	33
2.2. A MAMDANI által használt szakaszonként lineáris fuzzy halmazok reprodukciója [130] alapján	35
2.3. Intervallumértékű fuzzy halmaz	35
2.4. Példa 2-es típusú vagy másodfajú fuzzy halmazra	36
2.5. Emberek magasságára vonatkozó „kisnövésű”, „középtermetű” és magas fogalmakat reprezentáló fuzzy halmazok.	38
2.6. Példa konvex és szubnormális (A_1), továbbá nemkonvex és normális (A_2) fuzzy halmazokra	41
3.1. Példák „fiatal”, „középkorú” és „idős” fogalmakat reprezentáló tagsági függvényekre	44
3.2. Kettős küszöb típusú komplement	47
3.3. SUGENO-típusú komplementek	48
3.4. YAGER-típusú komplementek	49
3.5. Fuzzy metszetek grafikonjai	54
3.6. Fuzzy uniók grafikonjai	59
3.7. Fuzzy aggregációs operátorok	63
4.1. Példa nem teljesen rekonstruálható fuzzy relációra	71
4.2. Reláció ábrázolása páros gráffal („íjszerű” diagrammal)	72
4.3. Reláció reprezentálása irányított gráffal $X = Y$ esetén	75
4.4. Reflexivitás, szimmetria és tranzitivitás reprezentálása irányított gráffal	75
4.5. Az $R(X, X)$ alakú relációk fontosabb típusai	77
4.6. Kompatibilitási reláció ábrázolása reflexív irányítatlan gráffal (a hurokélek elhagyásával)	81
4.7. Kompatibilitási reláció teljes α -lefedése	82
4.8. Fuzzy részbenrendezés α -vágatai	85

5.1. Az $A \rightarrow B$ fuzzy szabály logikai implikációként való interpretációja	91
6.1. Zárthurkú irányítási rendszer vázlata	98
6.2. Közvetlen tudásalapú szakértő rendszer vázlata	101
6.3. Fuzzy szakértő rendszerek szerkezeti vázlata	102
7.1. Általános fuzzy irányítási rendszer vázlata	107
7.2. Fuzzy szabályok ábrázolása fuzzy függvénygörbével	111
7.3. Fuzzy szabály-reláció. A szabálybázis az $A_1 \rightarrow B_1$ és az $A_2 \rightarrow B_2$ szabályokat tartalmazza.	112
7.4. Az alaphalmaz ε -lefedése fuzzy halmazokkal	114
7.5. Fuzzy halmazok RUSPINI-partíciója	115
7.6. Az $\mathcal{A}$ fuzzy partíció hét, míg az $\mathcal{A}'$ három nyelvi kifejezést tartalmaz	115
7.7. A kompozíciós következtetési szabály	117
7.8. Az illeszkedés mértékének meghatározása egy dimenzióban	118
7.9. Az illeszkedés mértékének meghatározása több dimenzióban	118
7.10. Az R_i szabályhoz tartozó következtetés meghatározása	118
7.11. MAMDANI-irányító algoritmusa	119
7.12. LARSEN-típusú következtető eljárás által számolt konklúzió	121
7.13. Defuzzifikálás súlypont módszerrel	122
7.14. Rossz defuzzifikáláshoz vezető szituáció	123
7.15. Defuzzifikáláshoz a maximumok közepe módszerrel	125
7.16. TAKAGI-SUGENO-típusú irányítók működése	126
7.17. SUGENO- és MAMDANI-irányítók kapcsolata	127
7.18. Az illeszkedés mértékének meghatározása általános trapéz alakú tagsági függvények esetén, ha pontosan két szabály tüzel	131
8.1. Ritka szabálybázis: a megfigyelés a szabályokkal diszjunkt	142
8.2. Hangolás eredményeként keletkezett ritka szabálybázis [30]	143
8.3. Példa fuzzy szabályinterpolációs következtetés alkalmazására	144
8.4. Lineáris szabályinterpolációval számolt következtetés	148
8.5. A lineáris szabályinterpoláció geometriai jelentése trapéz alakú tagsági függvények esetén	149
8.6. Fuzzy halmazként közvetlenül nem értelmezhető konklúzióhoz vezető szituáció, ahol a konklúzió transzformálása után értelmes eredmény adódik	150

8.7. Fuzzy halmazként közvetlenül nem értelmezhető konklúzióhoz vezető szituáció, ahol még transzformálással sem lehet értelmes eredményt elérni	151
8.8. A testmetszéses módszer alapgondolata	155
8.9. Az antecedensek és a megfigyelés vektorrepresentációja (jobb oldalél)	157
8.10. A konzekvensek és a következtetés vektorrepresentációja (jobb oldalél)	158
8.11. Különböző töréspontok esetén a karakterisztikus pontok meghatározása	160
8.12. A következtetés koordinátái között fennálló összefüggés geometriai interpretációja (jobb oldalélre)	163
8.13. Példa a linearitás közelítő megtartására. A háromszög alakú tagsági függvények bal oldalélét ábrázoltuk: $A_1 : (0,1)$, $A_2 : (5,7)$, $A^* : (2,3)$, $B_1 : (0,2)$, $B_2 : (4,5)$	163
8.14. Szélsőségebb példa esetén is jó a lineáris közelítés. A bal oldalélek: $A_1 : (0,1)$, $A_2 : (10,100)$, $A^* : (1,10)$, $B_1 : (0,10)$, $B_2 : (10,11)$	164
9.1. Fordított inga esetén fellépő erőhatások	171
9.2. Az X_1 alaphalmaz és a mért szög lehetséges értékei	173
9.3. Az X_2 alaphalmaz és a becsült szögsebesség lehetséges értékei	173
9.4. Az Y alaphalmaz és a mozgató erő értékei	173
9.5. Részkonklúziók meghatározása	175
9.6. A következtetésként kapott fuzzy halmaz és a két defuzzifikációs módszer eredménye	175
9.7. A vezetónélküli targonca modellje	176
9.8. A becsült nyomvonalkövetés hibájának (δ) fuzzy partíciója	177
9.9. A vezetőnyom és vezetőpont távolságának (e_v) fuzzy partíciója	177
9.10. A pillanatnyi irány (V_d) fuzzy partíciója	178
9.11. A pillanatnyi sebesség (V_a) fuzzy partíciója	179
9.12. A pillanatnyi irány (V_d) és sebesség (V_a) irányítási felülete	181
9.13. A pillanatnyi irány (V_d) és sebesség (V_a) irányítási felülete szabályinterpolációs eljárás esetén	183
10.1. Az evolúció folyamata egy maximumkeresési példában	187
10.2. A szelekció művelete	188
10.3. A keresztezés operáció (bináris példa)	189
10.4. A keresztezés operáció (valós példa)	189

10.5. A mutáció művelete	190
10.6. Példa program és a hozzátartozó kifejezésfa	191
10.7. A keresztezés operáció	192
10.8. A mutáció művelete	193
10.9. Bakteriális evolúciós algoritmus	194
10.10A génátadás művelete	195
11.1. Többrétegű perceptron (MLP)	198
11.2. Egy neuron bemenetei és kimenete	199
11.3. Radiális bázisfüggvény hálózat (RBF)	200
11.4. B-spline neurális hálózat	202
11.5. Kétváltozós B-spline függvény	204
11.6. Példák	aktiváló függvényekre.
(a) küszöbfüggvény: $k(a) = 1$, ha $a \geq 0$; 0 , ha $a < 0$	
(b) szigmoid függvény: $s_{\beta}(a) = (1 + e^{\beta a})^{-1}$	206
11.7. Két bemenetű, két szabályt tartalmazó TAKAGI–SUGENO irányítóval ekvivalens ANFIS struktúra	206
11.8. Két bemenetű, kilenc szabályt tartalmazó TAKAGI–SUGENO irányítást megvalósító ANFIS struktúra	208
12.1. A kódolási elrendezés	214
12.2. A tagsági függvény paraméterei	217
12.3. A kódolási elrendezés	217
12.4. Kromoszóma a génátadásnál	218
12.5. Tagsági függvény megszüntetése	220
12.6. Tagsági függvények egyesítése	221
12.7. Kezdeti szabálybázis a pH problémára	230
12.8. Kezdeti szabálybázis az ICT problémára	230
12.9. LM módszer teljesítménye a pH problémára	231
12.10BProp módszer teljesítménye a pH problémára	231
12.11LM módszer teljesítménye az ICT problémára	232
12.12BProp módszer teljesítménye az ICT problémára	232
12.13A fuzzy rendszer paramétereinek alakulása	233
12.14Az MSE érték fejlődése	234
12.15Bakteriális memetik algoritmus	236
12.16A legjobb MSE értékű egyed cél- és hibaértékei az egyes mintákra a pH probléma esetén BEA használatával	241
12.17A legjobb MSE értékű egyed cél- és hibaértékei az egyes mintákra a pH probléma esetén BMA használatával	241

12.18MSE alakulása a pH probléma esetén	242
12.19MSE alakulása az ICT probléma esetén	242
12.20MSE alakulása a 6 változós probléma esetén	244
12.21Példa B-spline neurális hálózat kifejezésfájára	246
12.22A függvény mutáció művelete	248
12.23A terminális mutáció művelete	249
12.24A génátadás művelete	249
12.25Cél- és hibaértékek a 6 változós problémára GP használatával	255
12.26Cél- és hibaértékek a 6 változós problémára BP használatával	256
12.27Tapasztalati valószínűségi eloszlásfüggvény a pH problémára	257
12.28Tapasztalati valószínűségi eloszlásfüggvény az ICT probléma mára	257
12.29Tapasztalati valószínűségi eloszlásfüggvény a 6 változós prob- lémára	258

Irodalomjegyzék

- [1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass., 1974. (Magyar nyelven: *Számítógépes algoritmusok tervezése és analízise*, Műszaki Könyvkiadó, Budapest, 1982.).
- [2] A. Alkan and E. Ozcan. Memetic algorithms for timetabling. In *Proceedings of the 2003 Congress on Evolutionary Computation, CEC 2003*, pages 1796–1802, Canberra, Australia, 2003.
- [3] D. Ashlock and A. Sherk. Non-local adaptation of artificial predators and prey. In *Proceedings of the IEEE Congress on Evolutionary Computation 2005*, pages 41–48, Edinburgh, UK, 2005.
- [4] T. Bäck, F. Hoffmeister, and H.-P. Schwefel. A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 2–9, San Diego, July 1991.
- [5] P. Baranyi. *Fuzzy információ-tömörítő eljárások irányítási algoritmusokban*. PhD disszertáció, Budapesti Műszaki Egyetem, Budapest, 1999.
- [6] P. Baranyi, T. D. Gedeon, and L. T. Kóczy. A general interpolation technique in fuzzy rule bases with arbitrary membership functions.

- In *Proc. of the IEEE Int. Conf. on System Man and Cybernetics (IEEE-SMC'96)*, pages 510–515, Beijing, 1996.
- [7] P. Baranyi and L. T. Kóczy. A general and specialized solid cutting method for fuzzy rule interpolation. *BUSEFAL*, 67:13–22, 1996.
- [8] P. Baranyi, A. Martinovics, D. Tikk, L. T. Kóczy, and Y. Yam. A general extension of fuzzy SVD rule base reduction using arbitrary inference algorithm. In *Proc. of IEEE Int. Conf. on System Man and Cybernetics (IEEE-SMC'98)*, pages 2785–2790, San Diego, USA, 1998.
- [9] P. Baranyi, A. Martinovics, D. Tikk, Y. Yam, and I. Nagy. Fuzzy rule base reduction for arbitrary inference algorithm using singular value decomposition. In *Proc. of the 5th Int. Conf. on Soft Computing and Information/Intelligent Systems*, volume I, pages 487–490, Iizuka, Japan, 1998.
- [10] P. Baranyi, S. Mizik, L. T. Kóczy, T. D. Gedeon, and I. Nagy. Fuzzy rule base interpolation based on semantic revision. In *Proc. of the IEEE Int. Conf. on System Man and Cybernetics (IEEE-SMC'98)*, pages 1306–1311, San Diego, 1998.
- [11] P. Baranyi, D. Tikk, Y. Yam, L. T. Kóczy, and L. Náday. A new method for avoiding abnormal conclusion for α -cut based rule interpolation. In *Proc. of the 8th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'99)*, volume 1, pages 383–388, Seoul, Rep. of Korea, 1999.
- [12] P. Baranyi, Y. Yam, C. T. Yang, and A. Várkonyi-Kóczy. Complexity reduction of the rational general form. In *Proc. of the 8th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'99)*, volume I, pages 366–371, Seoul, Korea, 1999.
- [13] P. Baranyi, Y. Yam, C. T. Yang, and A. Várkonyi-Kóczy. Practical extension of the SVD based reduction technique for extremely large fuzzy rule bases. In *Proc. of the IEEE Int. Workshop on Intelligent Signal Proc. (WISP'99)*, pages 29–33, Budapest, Hungary, 1999.
- [14] P. Bauer, E. P. Klement, A. Leikermoser, and B. Moser. Modeling of control functions by fuzzy controllers. In H. Nguyen, M. Sugeno, R. Tong, and R.R. Yager, editors, *Theoretical Aspects of Fuzzy Control*, pages 91–116. Wiley, New York, 1995.
- [15] R. Bellman and M. Giertz. On the analytic formalism of the theory of fuzzy sets. *Information Sciences*, 5:149–156, 1973.

- [16] R. Bellman, R. Kalaba, and L. A. Zadeh. Abstraction and pattern classification. *J. of Math. Analysis and Applications*, 13(1):1–7, 1966.
- [17] R. Bellman and L. A. Zadeh. Decision making in a fuzzy environment. *Management Science*, 17(4):141–164, 1970.
- [18] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Functions Algorithms*. Plenum Press, New York, 1981.
- [19] C. Bishop. Improving the generalization properties of radial basis function neural networks. *Neural Computation*, 3:579–588, 1991.
- [20] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. E. Ruano. Estimating fuzzy membership functions parameters by the Levenberg-Marquardt algorithm. In *Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2004*, pages 1667–1672, Budapest, Hungary, July 2004.
- [21] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. E. Ruano. Fuzzy rule extraction by bacterial memetic algorithms. In *Proceedings of the 11th World Congress of International Fuzzy Systems Association, IFSA 2005*, pages 1563–1568, Beijing, China, July 2005.
- [22] J. Botzheim, C. Cabrita, L. T. Kóczy, and A. E. Ruano. Genetic and bacterial programming for B-spline neural networks design. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 11(2):220–231, February 2007.
- [23] J. Botzheim, B. Hámori, and L. T. Kóczy. Extracting trapezoidal membership functions of a fuzzy rule system by bacterial algorithm. In B. Reusch, editor, *Computational Intelligence, Theory and Applications*, volume 2206 of *Lecture Notes in Computer Science*, pages 218–227. Springer-Verlag, Berlin-Heidelberg, 2001.
- [24] J. Botzheim, B. Hámori, L. T. Kóczy, and A. E. Ruano. Bacterial algorithm applied for fuzzy rule extraction. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, IPMU 2002*, pages 1021–1026, Annecy, France, July 2002.
- [25] J. Botzheim, L. T. Kóczy, and A. E. Ruano. Extension of the Levenberg-Marquardt algorithm for the extraction of trapezoidal and general piecewise linear fuzzy rules. In *Proceedings of the 2002 IEEE World Congress on Computational Intelligence, WCCI 2002*, pages 815–819, Honolulu, Hawaii, May 2002.

- [26] M. Brown and C. Harris. *Neurofuzzy Adaptive Modelling and Control*. Prentice-Hall, 1994.
- [27] J. Bruinzeel, V. Lacrose, A. Titli, and H. B. Verbruggen. Real time fuzzy control of complex systems using rule-base reduction methods. In *Proc. of the 2nd World Automation Congress (WAC'96)*, Montpellier, 1996.
- [28] J. J. Buckley. Sugeno type controllers are universal controllers. *Fuzzy Sets and Systems*, 53(3):299–304, 1993.
- [29] J. J. Buckley. System stability and the fuzzy controller. In H. Nguyen, M. Sugeno, R. Tong, and R.R. Yager, editors, *Theoretical Aspects of Fuzzy Control*, pages 51–63. Wiley, New York, 1995.
- [30] D. G. Burkhardt and P. P. Bonissone. Automated fuzz knowledge base generation and tuning. In *Proc. of the 1st IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'92)*, pages 179–196, San Diego, 1992.
- [31] C. Cabrita, J. Botzheim, T. D. Gedeon, A. E. Ruano, L. T. Kóczy, and C. M. Fonseca. Bacterial memetic algorithm for fuzzy rule base optimization. In *Proceedings of the World Automation Congress, WAC 2006*, Budapest, Hungary, July 2006.
- [32] C. Cabrita, J. Botzheim, A. E. Ruano, and L. T. Kóczy. Design of B-spline neural networks using a bacterial programming approach. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2004*, pages 2313–2318, Budapest, Hungary, July 2004.
- [33] C. Cabrita, J. Botzheim, A. E. Ruano, and L. T. Kóczy. A hybrid training method for B-spline neural networks. In *Proceedings of the IEEE International Symposium on Intelligent Signal Processing, WISP 2005*, pages 165–170, Faro, Portugal, September 2005.
- [34] C. Cabrita, A. E. Ruano, and C. M. Fonseca. Single and multi-objective genetic programming design for B-spline neural networks and neuro-fuzzy systems. In *Proceedings of the IFAC Workshop on Advanced Fuzzy-Neural Control 2001, AFNC01*, pages 75–80, Valencia, Spain, Oct. 2001.
- [35] J. L. Castro. Fuzzy logic controllers are universal approximators. *IEEE Trans. on SMC*, 25(4):629–635, 1995.
- [36] J. Cselényi, Sz. Kovács, L. Pap, and L. T. Kóczy. New concepts in the fuzzy logic controlled path tracking strategy of the differential steered

- AGVs. In *Proc. of the 5th Int. Workshop on Robotics in Alpe-Adria-Danube Region*, Budapest, Hungary, 1996. p. 6.
- [37] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis. An immune algorithm for protein structure prediction on lattice models. *IEEE Transactions on Evolutionary Computation*, 11(1):101–117, Feb. 2007.
- [38] C. Darwin. *The Origin of Species*. John Murray, London, 1859.
- [39] R. J. P. De Figueiredo. Implications and applications of Kolmogorov’s superposition theorem. *IEEE Trans. on Autom. Control*, pages 1227–1230, 1980.
- [40] B. Della Vecchia. Direct and converse results by rational operators. *Constr. Approx.*, 12:271–285, 1996.
- [41] B. Della Vecchia, G. Mastroianni, and J. Szabados. Balázs–Shepard operators on infinite intervals. *Annales Uni. Sci. Budapest, Sectio Comput.*, 16:93–102, 1996.
- [42] B. Della Vecchia, G. Mastroianni, and V. Totik. Saturation of the Shepard operators. *Appr. Theory and its Appl.*, 6(4):76–84, 1990.
- [43] L. Ding, L. Shen, and M. Mukaidono. Revision principle for approximate reasoning, based on linear revising method. In *Proc. of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA’92)*, pages 305–308, Iizuka, 1992.
- [44] J. Dombi. A general class of fuzzy operators, the De Morgan class of fuzzy operator and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, 8(2):149–163, 1982.
- [45] M. Dorigo and L. M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [46] M. Dorigo, V. Maniezzo, and A. Colomi. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B Cybernetics*, 26(1):29–41, 1996.
- [47] D. Dubois, M. Grabisch, and H. Prade. Synthesis of real-valued mappings based on gradual rules and interpolative reasoning. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence (IJCAI ’93)*, Chambery, France, 1993.

- [48] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, New York, 1980.
- [49] D. Dubois and H. Prade. Gradual rules in approximate reasoning. *Information Science*, 61:103–122, 1992.
- [50] D. Dubois and H. Prade. Information engineering and fuzzy logic. In *Proc. of the 5th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'96)*, pages 1525–1531, New Orleans, 1996.
- [51] D. Dubois and H. Prade. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84:169–185, 1996.
- [52] R. C. Eberhart and J. Kennedy. *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [53] A. El Hajjaji and A. Rachid. Explicit formulas for fuzzy controllers. *Fuzzy Sets and Systems*, 62(2):135–141, 1994.
- [54] J. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.
- [55] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000.
- [56] J. C. Fodor. A remark on constructing t-norms. *Fuzzy Sets and Systems*, 41(2):195–199, 1991.
- [57] J. C. Fodor. A new look at fuzzy connectives. *Fuzzy Sets and Systems*, 57(2):141–148, 1993.
- [58] L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. Wiley, New York, 1966.
- [59] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithm in multiobjective optimization. *Evolutionary Computation*, 3:165–180, 1995.
- [60] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, 28(1):26–37, Jan. 1998.
- [61] M. J. Frank. On the simultaneous associativity of $f(x,y)$ and $x + y + f(x,y)$. *Aequationes Mathematicae*, 19(2–3):194–226, 1979.
- [62] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

- [63] J. Gebhart, F. Klawon, and R. Kruse. *Foundations of Fuzzy Systems*. John Wiley, New York, 1994.
- [64] T. D. Gedeon and L. T. Kóczy. Conservation of fuzziness in rule interpolation. In *Proc. of the Symp. on New Trends in Control of Large Scale Systems*, volume 1, pages 13–19, Herlány, 1996.
- [65] D. N. Godbole and J. Lygeros. Longitudinal control of a lead car of a platoon. *IEEE Trans. on Vehicular Technology*, 43(4):1125–1135, Nov 1994.
- [66] M. M. Gupta, A. Kandel, W. Bandler, and J. B. Kiszka. *Approximate Reasoning in Expert Systems*. North-Holland, New York, 1985.
- [67] M. M. Gupta and J. Qi. Theory of t-norms and fuzzy inference methods. *Fuzzy Sets and Systems*, 40(3):431–450, 1991.
- [68] R. Halavati, S. B. Shouraki, M. J. Heravi, and B. J. Jashmi. An artificial immune system with partially specified antibodies. In *Proceedings of Genetic and Evolutionary Computation Conference, GECCO'07*, pages 57–62, July 2007.
- [69] H. Hamacher. Über logische Verknüpfungen unscharfer Aussagen und deren Zugehörige Bewertungsfunktionen. In R. Trappl, G. J. Klir, and L. Ricciardi, editors, *Progress in Cybernetics and Systems Research*, volume 3, pages 276–288. Hemisphere, Washington D.C., 1978.
- [70] G. Hammond. *AGVS at Work – Automated Guided Vehicle Systems*. Springer Verlag, Heidelberg–New York, 1986.
- [71] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1990.
- [72] H. Hellendoorn, D. Driankov, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, Berlin, 1993.
- [73] H. Hellendoorn and C. Thomas. Defuzzification in fuzzy controllers. *J. of Intelligent and Fuzzy Systems*, 1(2):109–123, 1993.
- [74] M. Higashi and G. J. Klir. On measures of fuzziness and fuzzy complements. *Intern. J. of General Systems*, 8(3):169–180, 1982.
- [75] J. H. Holland. *Adaption in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts, 1992.
- [76] L. P. Holmblad and J. J. Ostergaard. Control of a cement kiln by fuzzy logic. In M. M. Gupta and E. Sanchez, editors, *Fuzzy Information and Decision Processes*, pages 389–399. North-Holland, New York, 1982.

- [77] J-S. R. Jang. Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In *Proc. of the 9th Nat. Conf. on Artificial Intelligence (AAAI'91)*, pages 762–767, 1991.
- [78] J-S. R. Jang. Rule extraction using generalized neural networks. In *Proc. of the 4th IFSA World Congress (IFSA'91)*, 1991.
- [79] J-S. R. Jang. ANFIS: adaptive-network based fuzzy inference system. *IEEE Trans. on SMC*, 23(3):665–685, 1993.
- [80] I. Joó, L. T. Kóczy, D. Tikk, and P. Várlaki. Stability of interpolative fuzzy KH-controllers. In *Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, volume I, pages 93–97, Barcelona, Spain, 1997.
- [81] S. H. Jung. Queen-bee evolution for genetic algorithms. *Electronics Letters*, 39(6):575–576, 2003.
- [82] A. Kandel. *Fuzzy Expert Systems*. CRC Press, Boca Raton, FL, 1991.
- [83] A. Kaufmann. *Introduction to the Theory of Fuzzy Subsets*. Academic Press, New York, 1975.
- [84] S. Kawase and Q. Chen. On fuzzy reasoning by Kóczy's linear rule interpolation. Technical report, Teikyo Heisei University, Ichihara, 1996.
- [85] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, Perth, Australia, 1995.
- [86] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1995.
- [87] L. T. Kóczy. Complexity of bounded compact rule based fuzzy inference. In *Towards a Unified Fuzzy Sets Theory. Proc. of the 3rd Joint IFSA-EC and EURO-WG Workshop on Fuzzy Sets*, pages 59–60, Visegrád, Hungary, 1990.
- [88] L. T. Kóczy. Computational complexity of various fuzzy inference algorithms. *Annales Univ. Sci. Budapest, Sect. Comp.*, 12:151–158, 1991.
- [89] L. T. Kóczy. Reasoning and control with incomplete and contradicting fuzzy rule bases. In *Proc. of Int. Symposia on Information Science of Kyushu Inst. of Technology (ISKIT '92)*, pages 67–70, Iizuka, Japan, 1992.

- [90] L. T. Kóczy. I-fuzzy structures: the world of strictly monotonous norms. In P. Z. Wang and K. F. Foe, editors, *Between Mind and Computer: Fuzzy Science and Engineering*, Advances in Fuzzy Systems — Applications and Theory, chapter 4, pages 105–137. World Scientific, Singapore, 1994.
- [91] L. T. Kóczy. Algorithmic aspects of fuzzy control. *Int. J. of Approximate Reasoning*, 12:159–217, 1995.
- [92] L. T. Kóczy. Fuzzy if then rules models and their transformation into one another. *IEEE Trans. on SMC*, 26(5):621–637, 1996.
- [93] L. T. Kóczy, editor. *Fuzzy Logic. Texts. Fuzzy systems II. (Fuzzy Reasoning and Control)*, volume II. TEMPUS JEP MODIFY 07759/94 Modify, Budapest, 1997.
- [94] 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*, pages 111–131. Springer, Berlin-Heidelberg, 2007.
- [95] L. T. Kóczy and K. Hirota. Rule interpolation by α -level sets in fuzzy approximate reasoning. *BUSEFAL*, 46(Printemps):115–123, 1991.
- [96] L. T. Kóczy and K. Hirota. Rule interpolation in approximate reasoning based fuzzy control. In R. Lowen and M. Roubens, editors, *Proc. of 4th IFSA World Congress*, pages 89–92, Brussels, Belgium, 1991.
- [97] L. T. Kóczy and K. Hirota. Approximate inference in hierarchical structured rule bases. In *Proc. of 5th IFSA World Congress (IFSA'93)*, pages 1262–1265, Seoul, 1993.
- [98] L. T. Kóczy and K. Hirota. Approximate reasoning by linear rule interpolation and general approximation. *Internat. J. Approx. Reason.*, 9:197–225, 1993.
- [99] L. T. Kóczy and K. Hirota. Ordering, distance and closeness of fuzzy sets. *Fuzzy Sets and Systems*, 60:281–293, 1993.
- [100] L. T. Kóczy and K. Hirota. Interpolation in hierarchical fuzzy rule bases with sparse meta-levels. Technical Report 97/3, Hirota Lab., Dept. of Comp. Intelligent and Sys. Sci., Tokyo Institute of Technology, Yokohama, 1997.

- [101] L. T. Kóczy and K. Hirota. Size reduction by interpolation in fuzzy rule bases. *IEEE Trans. on SMC*, 27:14–25, 1997.
- [102] L. T. Kóczy, K. Hirota, and T. D. Gedeon. Fuzzy rule interpolation by the conservation of relative fuzziness. Technical Report 97/2, Hirota Lab, Dept. of Comp. Intelligent and Sys. Sci., Tokyo Institute of Technology, Yokohama, 1997.
- [103] L. T. Kóczy, K. Hirota, and K. Ozawa. Knowledge representation and accumulation by fuzzy flip-flops. *Fuzzy Sets and Systems*, 39(1):1–13, 1991.
- [104] L. T. Kóczy and Sz. Kovács. On the preservation of the convexity and piecewise linearity in linear fuzzy rule interpolation. Technical Report TR 93–94/402, LIFE Chair of Fuzzy Theory, Tokyo Institute of Technology, Yokohama, 1993.
- [105] L. T. Kóczy and Sz. Kovács. The convexity and piecewise linearity of the fuzzy conclusion generated by linear fuzzy rule interpolation. *BUSEFAL*, 62(Automne):23–29, 1994.
- [106] L. T. Kóczy and Sz. Kovács. Shape of the fuzzy conclusion generated by linear interpolation in trapezoidal fuzzy rule bases. In *Proc. of the 2nd European Congress on Intelligent Techniques and Soft Computing*, pages 1666–1670, Aachen, 1994.
- [107] L. T. Kóczy and C. Magyar. On the minimal axiomatic system of I-fuzzy algebra. *BUSEFAL*, 32:19–31, 1987.
- [108] L. T. Kóczy and M. Sugeno. Explicit functions of fuzzy control systems. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 4:515–535, 1996.
- [109] L. T. Kóczy and D. Tikk. Approximation of transfer functions by various fuzzy controllers. In L. Reznik, V. Dimitrov, and J. Kacprzyk, editors, *Fuzzy System Design: Social and Engineering Applications*, number 17 in *Studies in Soft Computing and Fuzziness*, pages 202–224. Physica-Verlag, Heidelberg–New York, 1998.
- [110] L. T. Kóczy and A. Zorat. Fuzzy systems and approximation. *Fuzzy Sets and Systems*, 85:203–222, 1997.
- [111] L. T. Kóczy, A. Zorat, and T. D. Gedeon. The Cat and Mouse Problem: optimizing the size of fuzzy rule bases. In *Proc. of the 5th International*

- Workshop on Current Issues on Fuzzy Technologies (CIFT'95)*, pages 139–151, Trento, Italy, 1995.
- [112] A. N. Kolmogorov. O predstavlenii nyeprerivnih funkcij nyeszkolkih peremennih v vigye szuperpozicij nyeprerivnih funkcij odnovo peremennovo i szlozsennyija. (On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition). *Dokl. Akad. SSSR*, 114:953–956, 1957. (oroszul).
- [113] B. Kosko. Fuzzy systems as universal approximators. In *Proc. of the 1st IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'92)*, pages 1153–1162, San Diego, 1992.
- [114] B. Kosko. Function approximation with additive fuzzy systems. In H. Nguyen, M. Sugeno, R. Tong, and R. R. Yager, editors, *Theoretical Aspects of Fuzzy Control*, pages 313–347. Wiley, New York, 1995.
- [115] Sz. Kovács and L. T. Kóczy. Fuzzy rule interpolation in vague environment. In *Proc. of the 3rd European Congress on Intelligent Techniques and Soft Computing (EUFIT'95)*, pages 95–98, Aachen, 1995.
- [116] Sz. Kovács and L. T. Kóczy. Application of the approximate fuzzy reasoning based on interpolation in the vague environment of the fuzzy rule base in the fuzzy logic controlled path tracking strategy of differential steered AGV's. In *Computational Intelligence — Theory and Applications*, number 1226 in Lecture Notes in Computer Science, pages 456–467. Springer, Heidelberg, 1997.
- [117] Sz. Kovács and L. T. Kóczy. The use of the concept of vague environment in approximate fuzzy reasoning. *Tatra Mountains Math. Publ.*, 12:169–181, 1997.
- [118] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.
- [119] V. Kůrková. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, pages 501–506, 1992.
- [120] V. Kůrková. Kolmogorov's theorem. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 501–502. MIT Press, 1995.

- [121] R. Kruse, E. Schwecke, and J. Heinsohn. *Uncertainty and Vagueness in Knowledge-Based Systems: Numerical Methods*. Springer-Verlag, New York, 1991.
- [122] N. Kubota, K. Shimojima, and T. Fukuda. The role of virus infection in a virus-evolutionary genetic algorithm. *Journal of Applied Mathematics and Computer Science*, 6(3):415–429, 1996.
- [123] C. P. Kwong. Fuzzy inference without membership functions. Technical report, Div. of Info. Engineering, The Chinese University of Hong Kong, 1993.
- [124] J. B. Lamarck. *Zoological Philosophy*. Paris, Paris, 1809.
- [125] P. M. Larsen. Industrial application of fuzzy logic control. *Int. J. of Man Machine Studies*, 12(4):3–10, 1980.
- [126] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2(2):164–168, 1944.
- [127] J. Łukasiewicz. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. *C. R. Scéances Soc. Sci. Lettres de Varsovie*, 23:51–77, 1930.
- [128] J. Łukasiewicz and A. Tarski. Untersuchungen über den Aussagenkalkül. *C. R. Scéances Soc. Sci. Lettres de Varsovie*, 23:33–50, 1930.
- [129] J. Lygeros and D. N. Godbole. An interface between continuous and discrete event controllers for vehicle automation. *IEEE Trans. on Vehicular Technology*, 46(1):229–241, February 1997.
- [130] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. of Man Machine Studies*, 7(1):1–13, 1975.
- [131] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947.
- [132] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11(2):431–441, Jun. 1963.
- [133] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report. 826, California Institute of Technology, Pasadena, California, USA, 1989.

- [134] B. Moser. *A new approach for representing control surfaces by fuzzy rule bases*. PhD dissertation, Johannes Kepler University of Linz, Department of Mathematics, 1995.
- [135] B. Moser. Sugeno controllers with a bounded number of rules are nowhere dense. *Fuzzy Sets and Systems*, 104(2):269–277, 1999.
- [136] M.Salmeri, M.Re, E. Petrongari, and G.C.Cardarilli. A novel bacterial algorithm to extract the rule base from a training set. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 759–761, May 2000.
- [137] M. Mukaidono, L. Ding, and Z. Shen. Approximate reasoning based on revision principle. In *Proc. of the NAFIPS'90*, volume I, pages 94–97, 1990.
- [138] N. E. Nawa and T. Furuhashi. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems*, 7(5):608–616, Oct. 1999.
- [139] N. E. Nawa, T. Hashiyama, T. Furuhashi, and Y. Uchikawa. Fuzzy logic controllers generated by pseudo-bacterial genetic algorithm. In *Proceedings of the IEEE Int. Conf. Neural Networks (ICNN97)*, pages 2408–2413, Houston, 1997.
- [140] C. V. Negoita. *Expert Systems and Fuzzy Systems*. Benjamin Cummings, Menlo Parko, CA, 1985.
- [141] O. Nelles. *Nonlinear Systems Identification with Local Linear Neuro-Fuzzy Models*. PhD thesis, TU Darmstadt, Germany, 2000.
- [142] H.T. Nguyen and V. Kreinovich. On approximations of controls by fuzzy systems. Technical Report TR 92-93/302, LIFE Chair of Fuzzy Theory, Tokyo Institute of Technology, Tokyo, 1992.
- [143] Y. S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, Apr. 2004.
- [144] K. Ozawa, K. Hirota, and L. T. Kóczy. Algebraic fuzzy flip-flops. *Fuzzy Sets and Systems*, 39(2):215–226, 1991.
- [145] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. John Wiley, New York, 1989.

- [146] H. Pohlheim. The multipopulation genetic algorithm: Local selection and migration. Technical report, Technical University Ilmenau, 1995.
- [147] T. J. Procyk and E. H. Mamdani. A linguistic self-organizing process controller. *Automatica*, 15(1):15–30, 1979.
- [148] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.
- [149] M. Riedmiller. Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms. *Int. J. Computer Standards and Interfaces*, 16:265–278, 1994.
- [150] W. Rödder. On „and” and „or” connective in fuzzy set theory. Operations res., Technical University of Aachen, 1975.
- [151] A. E. Ruano, C. Cabrita, J. V. Oliveira, and L. T. Kóczy. Supervised training algorithms for B-spline neural networks and neuro-fuzzy systems. *International Journal of Systems Science*, 33(8):689–711, 2002.
- [152] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [153] E. H. Ruspini. A new approach to clustering. *Information Control*, 15(1):22–32, 1969.
- [154] B. Schweizer and A. Sklar. Associative functions and statistical triangle inequalities. *Publ. Math. Debrecen*, 8:169–186, 1961.
- [155] B. Schweizer and A. Sklar. Associative functions and abstract semigroups. *Publ. Math. Debrecen*, 10:69–81, 1963.
- [156] F. C. Schweppe. *Uncertain Dynamic Systems*. Prentice Hall, Upper Saddle River, New Jersey, 1973.
- [157] J. Selye. *Álomtól a felfedezésig. Egy tudós vallomásai*. Akadémiai Kiadó, Budapest, 1967. Eredeti: From Dream to Discovery. Confession of a Scientist. (Ford. Józsa Péter). New York–Toronto–London, McGraw-Hill.
- [158] S. H. Shao. Fuzzy self-organizing controller and its application for dynamic processes. *Fuzzy Sets and Systems*, 26(2):151–164, 1988.
- [159] Z. Shen, L. Ding, H. C. Lui, P. Z. Wang, and M. Mukaidono. Revision principle based on semantics revising method. In *Proc. of the IEEE*

- Proc. of ISMVL, 22nd Int. Symposium on Multi-Valued Logic*, pages 467–473, 1992.
- [160] Z. Shen, L. Ding, and M. Mukaidono. Fuzzy resolution principle. In *Proc. of the 18th Int. Symposium on Multi-Valued Logic*, pages 210–215, 1988.
- [161] Z. Shen, L. Ding, and M. Mukaidono. Methods of revision principle. In *Proc. of the 5th IFSA World Congress (IFSA'93)*, pages 246–249, Seoul, 1993.
- [162] D. Shepard. A two dimensional interpolation function for irregularly spaced data. In *Proc. of the 23rd ACM International Conference*, pages 517–524, 1968.
- [163] Y. Shi and M. Mizumoto. On Kóczy's interpolative reasoning method in sparse rule bases. In *Proc. of the 10th Fuzzy Systems Symposium*, pages 211–224, Osaka, 1994.
- [164] Y. Shi and M. Mizumoto. Some considerations on Kóczy's interpolative reasoning method. In *Proc. of the 4th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE/IFES'95)*, pages 2117–2122, Yokohama, 1995.
- [165] Y. Shi and M. Mizumoto. Some considerations on Kóczy's linear interpolative reasoning method. *Journal of SOFT*, 8:147–157, 1996. (In Japanese, with English abstract).
- [166] J. E. Smith. Coevolving memetic algorithms: A review and progress report. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(1):6–17, Feb. 2007.
- [167] B.G. Song, R.J. Marks II, S. Oh, P. Arabshahi, T.P. Caudell, and J.J. Choi. Adaptive membership function fusion and annihilation in fuzzy if-then rules. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 961–967, March 1993.
- [168] D. A. Sprecher. On the structure of continuous functions of several variables. *Trans. Amer. Math. Soc.*, 115:340–355, 1965.
- [169] A. Stoica. Fuzzy processing based on alpha-cut mapping. In *Proc. of the 5th IFSA World Congress (IFSA'93)*, pages 1266–1269, Seoul, 1993.
- [170] M. Sugeno. Fuzzy measures and fuzzy integrals: A survey. In M. M. Gupta, G. N. Sadiris, and B. R. Gaines, editors, *Fuzzy Automata and Decision Processes*, pages 89–102. North-Holland, Amsterdam–New York, 1977.

- [171] M. Sugeno. An introductory survey of fuzzy control. *Information Science*, 36(1–2):59–83, 1985.
- [172] M. Sugeno, M. F. Griffin, and A. Bastian. Fuzzy hierarchical control of an unmanned helicopter. In *Proc. of the 5th IFSA World Congress (IFSA'93)*, pages 1262–1265, Seoul, 1993.
- [173] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28(1):15–33, 1988.
- [174] M. Sugeno and G. K. Park. An approach to linguistic instruction based learning. *Intern. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 1(1):19–56, 1993.
- [175] M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Trans. on FS*, 1:7–31, 1993.
- [176] J. Szabados. On a problem of R. DeVore. *Acta Math. Acad. Sci. Hungar.*, 27:219–223, 1976.
- [177] J. Szabados. Direct and converse approximation theorems for Shepard-operator. *J. Approx. Th. and its Appl.*, 7:63–76, 1991.
- [178] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on SMC*, 15(1):116–132, 1985.
- [179] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-posed Problems*. Winston, Washington, 1977.
- [180] D. Tikk. *Investigation of fuzzy rule interpolation techniques and the universal approximation property of fuzzy controller*. PhD disszertáció, Budapesti Műszaki Egyetem, Budapest, 1999.
- [181] D. Tikk. On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules. *Tatra Mountains Math. Publ.*, 16:369–377, 1999.
- [182] D. Tikk, P. Baranyi, Y. Yam, and L. T. Kóczy. On the preservation of piecewise linearity of a modified rule interpolation approach. In *Proc. of the EUROFUSE-SIC'99 conference*, pages 550–555, Budapest, Hungary, 1999.
- [183] D. Tikk, P. Baranyi, Y. Yam, and L. T. Kóczy. Stability of a new interpolation method. In *Proc. of the IEEE Int. Conf. on System, Man,*

and Cybernetics (IEEE-SMC'99), volume III, pages 7–9, Tokyo, Japan, October, 1999.

- [184] R. M. Tong. An annotated bibliography of fuzzy control. In M. Sugeno, editor, *Industrial Application of Fuzzy Control*, pages 249–269. North-Holland, New York, 1985.
- [185] I. B. Türkşen and Z. Zhong. An approximate analogical reasoning approach of functions. In *Proc. of the 2nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA'92)*, pages 629–632, Iizuka, Japan, 1992.
- [186] J. Varga and L. T. Kóczy. Explicit formulae of two-input fuzzy control. *BUSEFAL*, 63(Été):58–66, 1995.
- [187] J. Varga and L. T. Kóczy. Explicit function of multiple input controller. In *Proc. of the EUFIT'95*, pages 103–106, Aachen, 1995.
- [188] Gy. Vass, L. Kalmár, and L. T. Kóczy. Extension of the fuzzy rule interpolation method. In *Proc. of the Int. Conf. on Fuzzy Sets Theory and its Applications*, Liptovský Mikuláš, 1992.
- [189] J. L. Verdegay and M. Delgado. *Approximate Reasoning Tools for Artificial Intelligence*. Verlag TÜV Rheinland, Köln, 1990.
- [190] L. X. Wang. Fuzzy systems are universal approximators. In *Proc. of the 1st IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'92)*, pages 1163–1169, San Diego, 1992.
- [191] L. X. Wang, R. Langari, and J. Yen. Principal components, B-splines, and fuzzy system reduction. In W. Chiang and J. Lee, editors, *Fuzzy Logic for the Applications to Computer Systems*, pages 255–259. World Scientific, 1996.
- [192] L. X. Wang and J. Mendel. Generating fuzzy rules from numerical data with suplications. Technical Report TR USC-SIPI #169, Signal and Image Processing Institute, University of Southern California, 1991.
- [193] S. Weber. A general concept of fuzzy connectives, negations and implications based on t -norms and t -conorms. *Fuzzy Sets and Systems*, 11(2):115–134, 1983.
- [194] P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. Ph.D. dissertation, Harvard University, Cambridge, 1970.

- [195] P. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Committee on Appl. Math., Harvard Univ., Cambridge, MA, USA, Nov. 1974.
- [196] E. Weyer and T. Kavli. The ASMODO algorithm. Some new theoretical and experimental results. Technical Report SINTEF Report STF31 A95024, SINTEF, Oslo, 1995.
- [197] R. R. Yager. On the measure of fuzziness and negation. part i: Membership in the unit interval. *Intern. J. of General Systems*, 5(4):221–229, 1979.
- [198] R. R. Yager. On the measure of fuzziness and negation. part ii: Lattices. *Information and Control*, 44(3):236–260, 1980.
- [199] R. R. Yager. On ordered weighted averaging aggregation operators in multilateral decision making. *IEEE Trans. on SMC*, 18(1):183–190, 1988.
- [200] Y. Yam. Fuzzy approximation via grid point sampling and singular value decomposition. *IEEE Trans. on SMC*, 27(6):933–951, 1997.
- [201] Y. Yam, P. Baranyi, and C. T. Yang. Reduction of fuzzy rule base via singular value decomposition. *IEEE Trans. on FS*, 7(2):120–132, 1999.
- [202] Y. Yam and L. T. Kóczy. Representing membership functions as points in high dimensional spaces for fuzzy interpolation and extrapolation. Technical Report CUHK-MAE-97-03, Dept. of Mechanical and Automation Eng., The Chinese Univ. of Hong Kong, 1997.
- [203] K. F. C. Yiu, S. Wang, K. L. Teo, and A. C. Tsoi. Nonlinear system modeling via knot-optimizing B-spline networks. *IEEE Trans. on Neural Networks*, 12:1013–1022, 2001.
- [204] W. Yu and Z. Bien. Design of fuzzy logic controller with inconsistent rule base. *J. of Intelligent and Fuzzy Systems*, 3:20 p., 1995.
- [205] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
- [206] L. A. Zadeh. Towards a theory of fuzzy systems. In R. E. Kalman and R. N. De Clairis, editors, *Aspects of Networks and Systems Theory*, pages 469–490. Holt, Rinehart & Winston, New York, 1971.
- [207] L. A. Zadeh. A rationale for fuzzy control. *J. of Dynamical Systems, Measurement and Control (Trans. ASME Ser G.)*, 94(1):3–4, 1972.

- [208] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. on SMC*, 1(1):28–44, 1973.
- [209] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning I, II, III. *Information Science*, 8:199–251, 301–357, 9:43–80, 1975.
- [210] L. A. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30(1):407–428, 1975.
- [211] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.
- [212] L. A. Zadeh. Fuzzy sets and information granularity. In R. K. Ragade, R. R. Yager, and M. M. Gupta, editors, *Advances in Fuzzy Set Theory and Applications*, pages 3–18. North Holland, Amsterdam, 1979.
- [213] L. A. Zadeh. The calculus of fuzzy if/then rules. *AI Expert*, 7(3):23–27, 1992.
- [214] W. Zhang, D. Ma, H.-J. Zhang, B.-L. Wang, and Y.-T. Chen. An application of multipopulation genetic algorithm for optimization of adversaries's tactics and strategies in battlefield simulation. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, pages 1704–1707, Xi'an, November 2003.
- [215] J. M. Zurada. *Introduction to Artificial Neural Systems*. West Publishing Co., St. Paul, 1992.

Tárgymutató

- aggregációs operátor, 59
 - folytonossága, 60
 - idempotens, 60
 - monotonitása, 60
 - peremfeltételei, 60
 - szimmetrikus, 60
- aktiváló függvény, 206
- alaphalmaz, 29
 - diszkretizálása, 39
- α -vágat, 39
 - lényeges, 40, 150
 - szigorú, 39
- algebrai szorzat, 53
- alszabálybázis, 166
- általános közép, 61
- ANFIS, 207
 - bemeneti paraméterek, 208
 - kimeneti paraméterek, 209
- antecedens, 109
- antireflexivitas, 75
- antiszimmetria, 75
 - szigorú, 75
- antitranszitivitas, 75
- aszimmetria, 75
- asszociativitas
 - crisp halmazoké, 31
 - fuzzy metszeté, 52
 - fuzzy unióé, 55
- átlagoló operátor, 61
- BOOLE-algebra, 10, 31, 62
- CALPATH, 13
- CNF függvénygenerátor, 107
- crisp halmaz
 - megadása, 29
- crisp relációk
 - reflexivitása, 75
 - szimmetriája, 75
 - transzitiv lezártja, 76
 - transzitivitása, 75
- DE MORGAN-algebra *v.* -háló, 43, 52
- DE MORGAN-azonosság, 31, 57, 64
- defuzzifikáció, 121
 - geomteriai középpont módszer (COA), 123, 129
 - középső maximum módszer (COM), 125
 - maximumok közepe módszer (MOM), 124
 - súlypont módszer (COG), 122, 129
- defuzzifikáló egység, 106
- DESCARTES-szorzat
 - crisp halmazoké, 31
 - n halmazé, 32
- diszjunkt halmazok, 31
- disztributív egyenlőtlenség
 - t -normára vonatkozó, 64
- disztributivitas
 - crisp halmazoké, 31
- domináló osztály, 83
 - nem-, 83
- dominált osztály, 83
 - nem-, 83
- duális hármas, 57
- duális pont, 50
- dualitás
 - crisp metszet és unióé, 30

- egyensúlyi pont
 - fuzzy halmazé, 43
 - komplementésé, 48
- egyesítés
 - crisp halmazoké, 30
- egymásba ágyazott halmazcsalád, 40
- ekvivalenciaosztály, 78
- ekvivalenciareláció, 77, 78
 - fuzzy, 79
- ellentmondás törvénye, 11, 23, 31, 43
- elnyelési törvények, 31
- előrendezés, 77
- felbontási elv, 146
- feltételes invertálhatóság
 - t-konormáé, 64
- fordított inga szabályozása, 171
- fuzzy differencia *lásd* fuzzy különbség 64
- fuzzy függvénygörbe, 110
- fuzzy halmaz
 - 2-es szintű, 37
 - 2-es típusú, 36
 - 3-as típusú, 36
 - általánosításai, 34
 - CNF, 145
 - egyensúlyi pontja, 43
 - értelmezései, 113
 - fogalma, 32
 - haranggörbe alakú, 34, 134, 208
 - harmadfajú, 36
 - háromszög alakú, 34
 - intervallumértékű, 35
 - karakterisztikus pontjai, 156
 - konvex, 40
 - L*-, 37
 - magassága, 40
 - magja, 40
 - másodfajú, 36
 - normális, 40
 - szakaszonként lineáris, 34
 - szubnormális, 40
 - tartója, 40
 - trapéz alakú, 34
 - vektorrepresentációja, 156
- fuzzy halmazműveletek, 43
- fuzzy hatványhalmaz, 36
- fuzzy irányítási rendszer
 - alkalmazásai, 170
 - felépítése, 105
 - univerzális közelítő tulajdonsága, 133
- fuzzy irányítási rendszer *lásd* fuzzy irányító 105
- fuzzy irányító
 - MAMDANI-típusú, 117
 - SUGENO-típusú (nullandrendű), 127
 - TAKAGI–SUGENO–KANG-típusú, 127
 - TAKAGI–SUGENO-típusú, 127, 207
- fuzzy komplement, 45
 - axiomatikus váza, 45
 - egyensúlyi pontja, 48
 - folytonossága, 46
 - involutivitása, 46
 - küszöb-típusú, 47
 - monotonitása, 45
 - peremfeltételei, 45
 - SUGENO-típusú, 47
 - YAGER-típusú, 48
 - ZADEH-féle, 43, 47
- fuzzy különbség, 64
 - szimmetriája, 64
- fuzzy metszet, 51

- algebrai v. interaktív, [53, 63](#)
 - asszociativitása, [52](#)
 - axiomatikus váza, [52](#)
 - folytonossága, [52](#)
 - kommutativitása, [51](#)
 - monotonitása, [51](#)
 - peremfeltétele, [51](#)
 - szubidempotenciája, [52](#)
 - ZADEH-féle, [43, 52](#)
- fuzzy műveletek
 - algebrai, [44](#)
 - ZADEH-féle, [43](#)
- fuzzy partíció, [114](#)
- fuzzy pont, [110](#)
- fuzzy produkciós szabály, [103](#)
- fuzzy reláció, [66](#)
 - bináris, [71](#)
 - értékkészlete, [71](#)
 - értelmezési tartománya, [71](#)
 - hengeres kiterjesztése, [68](#)
 - hengeres lezártja, [70](#)
 - „íjszerű” diagramja, [72](#)
 - inverze, [72](#)
 - magassága, [71](#)
 - projekciója, [68](#)
 - rekonstruálhatósága, [70](#)
 - reprezentálása, [67](#)
 - tagsági függvénye, [66](#)
 - tagsági mátrixa, [72](#)
 - transzítív lezártja, [77](#)
 - max-min, [77](#)
- fuzzy relációk
 - max-min kompozíciója, [73](#)
- fuzzy szabály, [109](#)
 - relációja, [111](#)
- fuzzy szabálybázis, [109](#)
 - felépítése, [108](#)
 - relációja, [111, 119](#)
- fuzzy unió, [55](#)
- algebrai v. interaktív, [57, 63](#)
 - asszociativitása, [55](#)
 - axiomatikus váza, [55](#)
 - feltételes invertálhatósága, [64](#)
 - folytonossága, [55](#)
 - kommutativitása, [55](#)
 - monotonitása, [55](#)
 - peremfeltétele, [55](#)
 - szuperidempotenciája, [55](#)
 - ZADEH-féle, [43](#)
- függvény
 - racionális, [65](#)
- geometriai közép, [61](#)
- gerjesztési függvény, [105, 206](#)
- gradiens módszer, [206](#)
- halmaz
 - altère, [68](#)
 - crisp
 - azonossága, [29](#)
 - számossága, [30](#)
 - üres, [29](#)
- halmazcsalád, [30](#)
- halmazműveletek
 - alaptulajdonságai, [31](#)
- háló, [31, 37, 83](#)
 - algebrai, [43](#)
- harmadik kizárásának törvénye,
 - [11, 23, 31, 43](#)
- harmonikus közép, [61](#)
- hasonlósági osztály, [79](#)
- hasonlósági reláció, [79](#)
- hatványhalmaz, [30](#)
 - fuzzy, [36](#)
- hengeres kiterjesztés, [68](#)
- hengeres lezárt, [70](#)
- hipotetikus szillogizmus, [15, 98](#)
- homokkupac-paradoxon, [9](#)
- idempotencia, [31, 52, 55, 62](#)

- identifikáció, 100
- identitás, 31
- IFSA, 24
- I-fuzzy algebra
 - axiómái, 64
- I-fuzzy struktúrák, 44, 62
- illeszkedés mértéke, 117
 - normalizált, 208
- illeszkedési mértéket meghatározó egység, 105
- involúció, 31, 46, 50
 - fuzzy különbséggel, 64
- irányított gráfok, 74
 - reflexivitása, 76
 - szimmetriája, 76
 - transzitivitása, 76
- irreflexivitás, 75
- karakterisztikus függvény, 29
- KH-interpoláció
 - abnormális kimenete, 148
 - kiterjesztése, 152
 - lineáris, 147
 - linearitás megőrzése, 150
 - matematikai stabilitása, 152
- kiterjesztési elv, 146
- klaszterezés, 108
- kommunikációs felület, 103
- kommutativitás
 - crisp halmazoké, 31
 - fuzzy különbségé, 64
 - fuzzy metszeté, 51
 - fuzzy unióé, 55
- kompatibilitási
 - osztály, 80
 - α -, 80
 - legnagyobb, 80
 - legnagyobb α -, 80
 - reláció, 77, 80
- komplements
 - crisp halmazé, 30
 - fuzzy, 45
- komplexitás
 - aszimptotikus, 137
 - idő-, 137
 - tár-, 137
- kompozíciós következtetési szabály, 24, 117, 120
- konvexitás, 40
- konzekvens, 109
- korlát
 - alsó, 83
 - felső, 83
 - legkisebb felső, 83
 - legnagyobb alsó, 83
- következtetés, 109
- következtetési algoritmus, 117
- következtető egység *lásd* következtető gép 102
- következtető gép, 106
- kvázi-ekvivalencia, 77
- lefedés
 - teljes, 80
 - teljes α -, 80
- legkisebb felső korlát, 83
 - hálóban, 31
- legnagyobb alsó korlát, 83
 - hálóban, 31
- LESBEGUE-mérték, 153
- lingvisztikai változó *lásd* nyelvi változó 112
- lingvisztikus közelítő egység, 107
- logika
 - háromértékű, 11
 - alpműveletei, 11
 - n -értékű, 12
- macska-egér probléma, 18
- mag, 40

- magasság, [40](#)
- max-algebrai kompozíció, [74](#)
- max-min kompozíció, [73](#), [120](#)
- megelőző, [82](#)
 - közvetlen, [82](#)
- meghatározottlanság, [69](#)
- mesterséges intelligencia (AI), [13](#)
- metaszabály, [103](#), [166](#)
- metaszabálybázis, [103](#)
- metaszint, [166](#)
- MINKOWSKI-távolság, [145](#)
- modell
 - szabálybázis alapú, [105](#)
- modus ponens*, [15](#), [98](#)
- modus tollens*, [15](#), [98](#)
- monotonitás
 - szigorú
 - fuzzy különbségé, [64](#)
 - t-konormáé, [55](#), [64](#)
 - t-normáé, [52](#)
- nem összehasonlítható
 - pár, [82](#)
- neurális hálózat
 - adaptív, [206](#)
 - többrétegű előrecsatolt, [206](#)
 - többrétegű visszacsatolatlan, [206](#)
- neuron
 - aktiváló v. gerjesztési függvénye, [105](#), [206](#)
- nontranszitivitás, [75](#)
- normális, [40](#)
- nyelvi változó, [109](#), [112](#)
- oszd meg és uralkodj eljárás, [93](#)
- OWA operátor, [62](#)
- partíció, [167](#)
 - crisp halmazé, [31](#)
 - fuzzy, [114](#)
 - ritka fuzzy, [168](#)
 - RUSPINI-, [114](#)
 - teljes, [167](#)
- projekció, [68](#)
- ráképezés, [46](#)
- rákövetkező, [82](#)
 - közvetlen, [82](#)
- reflexivitás, [74](#), [75](#)
 - ε -, [76](#)
- reláció, [32](#)
 - bináris, [66](#)
 - ekvivalencia-, [77](#), [78](#)
 - hasonlósági, [79](#)
 - karakterisztikus függvénye, [66](#)
 - kompatibilitási, [80](#)
 - n -áris, [66](#)
 - szomszédsági, [80](#)
 - ternáris, [66](#)
 - tolerancia, [80](#)
- rendezés
 - elő-, [77](#)
 - kvázi-, [77](#)
 - parciális lásd részben rendezés [81](#)
 - részben, [77](#), [81](#)
 - CNF halmazokon, [145](#)
 - első eleme, [82](#)
 - fuzzy, [83](#)
 - fuzzy halmazokon, [145](#)
 - maximális eleme, [82](#)
 - minimális eleme, [82](#)
 - utolsó eleme, [82](#)
 - szigorú, [77](#)
- részhalmaz, [29](#)
 - valódi, [30](#), [167](#)
- részsorozat, [68](#)
- revíziós függvény, [154](#)
- RUSPINI-partíció, [114](#), [127](#)

- sehol sem sűrű, 136
- STONE–WEIERSTRASS-tétel, 134
- súlyvektor, 62
- szabály
 - fuzzy, 105
 - ha–akkor, 15, 105
 - tüzelő, 106
- szabálybázis, 105
 - felépítése, 142
 - hangolása, 143
 - lokális, 166
 - ritka, 142
- szabályinterpoláció, 143
 - alapegyenlete, 146
 - GK-interpoláció, 155
 - KH-interpoláció, 147
 - testmetszéssel, 154
 - VKK-interpoláció, 154
- szakértő rendszerek
 - adatbázisa, 102
 - fuzzy, 102
 - következtető gépe, 102
 - közvetlen, 102
 - tudásbázis alapú, 97
 - tudásbázisa, 102
- számtani közép, 61
- szimmetria, 74, 75
- szomszédsági reláció, 80
- szubidempotencia, 52, 63
- szubnormális, 40
- szuperidempotencia, 55, 63
- szűrjekció, 46
- tagsági érték, 32
- tagsági függvény, 32, 105
 - értéke, 32
 - fuzzy relációé, 66
 - haranggörbe alakú, 34, 134, 208
 - háromszög alakú, 34
 - trapéz alakú, 34
- tanulási algoritmus
 - gradiens módszer, 206
 - hibrid, 206
- tartó, 40
- távolság
 - alsó, 145
 - CNF halmazoké, 145
 - felső, 145
- teljesítményindex, 100
- tenzorszorzat, 136
- természetes nyelvi szabály, 109
- t-konorma, 44, 55
 - arkhimédészi, 55
 - szigorú arkhimédészi, 55
- t-norma, 44, 51
 - arkhimédészi, 52
 - szigorú arkhimédészi, 52
- toleranciareláció, 80
- transzitivitás, 74, 75
 - max–min, 76
- tudásbázis, 102
 - analízise, 108
- tüzelési érték, 117
 - normalizált, 208
- tüzelő szabály, 106
- unió
 - crisp halmazoké, 30
- univerzális közelítő, 134
- univerzum, 32
- üres halmaz, 29
- vezetőnélküli targonca irányítása, 175
- vezetőnyom, 176
- vezetősáv, 177
- VKK-interpoláció, 154