

SZEGEDI TUDOMÁNYEGYETEM
Természettudományi és Informatikai Kar
Számítógépes Algoritmusok és Mesterséges Intelligencia
Tanszék

Informatika Doktori Iskola

**A fuzzy operátorok egy speciális
osztályának alkalmazása hatások
modellezésére és döntési problémákra**

Doktori értekezés tézisei

Dombi József Dániel

Témavezető:
Dr. Csirik János

Szeged, 2013

1. Bevezetés

A Fuzzy halmazokat 1965-ben vezette be Zadeh azért, hogy az emberi gondolkodás matematikai modellezése megvalósítható legyen a számítástudományban. Már az első munkájában meghatározta a fuzzy elmélet alap definícióit, azaz a klasszikus matematikai jelölések fuzzy kiterjesztését, így a logikai operátorokat, szabályok alkalmazását, fuzzy relációkat és mennyiségi jellemzők transzformációit.

Az elmúlt évtizedekben a fuzzy halmazok és a fuzzy logika egyre népszerűbb terület lett és egyre több alkalmazás készült a segítségével. Így felhasználta a számítás tudomány és mérnöki tudomány. Ennek következménye, hogy nagyon sok cikket publikáltak, amely a fuzzy logikával foglalkozik. A könyvek száma is több száz, amelyek különböző módon vizsgálják a fuzzy elméletet és annak módszertanát. A sok tudományos eredmény ellenére nem lehet egy egységes fuzzy elméletről beszélni, ugyanúgy, ahogy egységes következtetési eljárásról sem.

A fuzzy elmélet egy speciális osztálya a Pliant rendszer (rugalmas rendszer), amely a fuzzy elméletben felmerülő problémákra keres megoldást [2]. A két rendszer között az operátorok választásában van különbség. A fuzzy elméletben fontos szerepet játszik a halmazhoztartozási függvény, de ennek nincs rögzített definíciója. A Pliant rendszerben úgy nevezett felfújó függvény (distending) helyettesíti a halmazhoztartozási függvényt, amely "lágú (soft) egyenlőtlenséget" reprezentál. A Pliant rendszerben az operátorok (conjunction, disjunction, aggregative operátor) között egyértelmű kapcsolat van.

A disszertációm főbb célja, hogy bemutassam, hogy a Pliant rendszert hogyan lehet alkalmazni valós problémákon. A kutatás során a gyakorlatiasságot és a pragmatikusságot tartottam folyamatosan szem előtt. Elsőként a dinamikus rendszerek területeken alkalmaztam a Pliant koncepciót. Itt hasonló rendszert hoztunk létre, mint a Fuzzy Cognitive Map. Továbbá a Pliant rendszert függvény közelítésre alkalmaztam, amely praktikus szempontból bizonyult nagyon hasznosnak. Végül megmutattam, hogy a Pliant koncepciót döntés támogató eszközként is lehet alkalmazni.

2. Fogalmak és előzmények

Ebben a fejezetben ismertetem azokat az általános fogalmakat jelöléseket, amelyek a tézishez szükségesek.

2.1. Negációs operátor

1. Definíció. Azt mondjuk, hogy $n(x)$ negációs operátor, ha $n: [0, 1] \rightarrow [0, 1]$ teljesíti az alábbi feltételeket

- C1: $n: [0, 1] \rightarrow [0, 1]$ (Continuity)
C2: $n(0) = 1, n(1) = 0$ (Boundary conditions)
C3: $n(x) < n(y)$ for $x > y$ (Monotonicity)
C4: $n(n(x)) = x$ (Involution)

2.2. Conjunctive, disjunctive és a módosító operátorok

A Pliant koncepció során úgy választjuk meg az operátorokat (szigorúan monoton t-norm és szigorúan monoton t-conorm), hogy különböző negációkra is DeMorgan osztályt alkossanak. A negáció ν_* fixpontját vagy a ν neutrális értékét döntési szintnek tekinthetjük. Azon operátor osztályok, amelyekhez sok negáció tartozhat különösen hasznosak, mert a küszöb érték változhat.

1. Tétel. $c(x, y)$ és $d(x, y)$ DeMorgan osztályt alkot az $n_{\nu_*}(x)$, ahol $n_{\nu_*}(\nu_*) = \nu_*$ minden $\nu_* \in (0, 1)$ akkor és csak akkor, ha

$$f_c(x)f_d(x) = 1. \quad (1)$$

Bizonyítás itt található [3].

2. Definíció. A Pliant rendszer általános formulája a következő

$$o_\alpha(x, y) = f^{-1} \left((f^\alpha(x) + f^\alpha(y))^{1/\alpha} \right) \quad (2)$$

$$n_\nu(x) = f^{-1} \left(f(\nu_0) \frac{f(\nu)}{f(x)} \right) \quad \text{or} \quad (3)$$

$$n_{\nu_*}(x) = f^{-1} \left(\frac{f^2(\nu_*)}{f(x)} \right), \quad (4)$$

ahol $f(x)$ egy szigorúan monoton t-norma operátor generátor függvénye és $f: [0, 1] \rightarrow [0, \infty]$ folytonos és szigorúan monoton csökkenő függvény.

3. Definíció. A Pliant rendszerben a módosító (modifier) operátor általános formája a következő

$$\kappa_{\nu, \nu_0}^{(\lambda)}(x) = f^{-1} \left(f(\nu_0) \left(\frac{f(x)}{f(\nu)} \right)^\lambda \right) \quad (5)$$

2.3. Aggregative operátor

Yager and Rybalov [7] vezette be először az *uninorm* kifejezést. Az uninormák a t-normák és t-conormák általánosítása, melynek során elhagyjuk az egység elemre vonatkozó feltételt a $\{0, 1\}$ intervallumban. Fodor, Yager és Rybalov [4] definiálták az uninormák egy részosztályát, a reprezentatív uninormákat. Az alábbi módon definiáljuk az aggregative operátort [1]

4. Definíció. Az aggregative operátor egy $a : [0, 1]^2 \rightarrow [0, 1]$ függvény a következő tulajdonságokkal

1. Folytonos a $[0, 1]^2 \setminus \{(0, 1), (1, 0)\}$
2. $a(x, y) < a(x, y')$ if $y < y', x \neq 0, x \neq 1$
 $a(x, y) < a(x', y)$ if $x < x', y \neq 0, y \neq 1$
3. $a(0, 0) = 0$ és $a(1, 1) = 1$ (boundary conditions)
4. $a(x, a(y, z)) = a(a(x, y), z)$ (associativity)
5. Létezik olyan n erős negáció, amelyre teljesül $a(x, y) = n(a(n(x), n(y)))$ (ön-DeMorgan azonosság) ha $\{x, y\} \neq \{0, 1\}$ or $\{x, y\} \neq \{1, 0\}$
6. $a(1, 0) = a(0, 1) = 0$ vagy $a(1, 0) = a(0, 1) = 1$

Az aggregative operátor multiplikatív formulája a következő

$$a_{\nu_*}(\mathbf{w}, \mathbf{x}) = f_a^{-1} \left(f_a^{1 - \sum_{i=1}^n w_i} (\nu_*) \prod_{i=1}^n f_a^{w_i}(x_i) \right) \quad (6)$$

2.4. Felfújó függvény

A fuzzy koncepcióban a legfontosabb fogalom a halmazhozzartozási függvény. Mind a mai napig a kutatói közösségi nem tudott egy egyértelmű definíciót meghatározni erre a fogalomra. A Pliant koncepcióban

meg tudunk adni egy olyan definíciót, amely az operátor rendszerhez szorosan kapcsolódik. A Pliant koncepcióban az alábbi jelölést használjuk a felfújó függvényre

$$\delta(x) = \text{truth}(0 < x) \quad x \in R$$

Ezt tudjuk általánosítani a következő módon

$$\delta(g(\mathbf{x})) = \text{truth}(0 < g(\mathbf{x})) \quad \mathbf{x} \in R^n$$

A klasszikus reláció helyett egy függvényt definiálunk, amely a reláció "érvényességéről" ad információt.

A fuzzy logika elméletében a halmazhoz tartozási függvénynek más jelentése van. A Pliant koncepcióban a halmazhoz tartozási függvényt "lágú intervallumra" cseréljük. A matematikai leírása az alábbi

$$\delta_{a,b}^{\lambda_1, \lambda_2}(x) = \text{truth}(a <_{\lambda_1} x <_{\lambda_2} b)$$

5. Definíció. *A Pliant rendszerben, ha az alábbi feltételek teljesülnek*

$$\delta_{a,b}^{\lambda_1, \lambda_2}(a) = \nu_0 \quad \delta_{a,b}^{\lambda_1, \lambda_2}(b) = \nu_0, \quad (7)$$

akkor a felfújó intervallum a következő

$$\delta_{a,b}^{\lambda_1, \lambda_2}(x) = f^{-1} \left(\frac{1}{A} \left(A_1 e^{-\lambda_1(x-a)} + A_2 e^{-\lambda_2(b-x)} \right) \right), \quad (8)$$

ahol

$$\begin{aligned} A &= \frac{1}{f(\nu_0)} \left(1 - e^{-(\lambda_1 + \lambda_2)(b-a)} \right) \\ A_1 &= 1 - e^{-\lambda_2(b-a)} \\ A_2 &= 1 - e^{-\lambda_1(b-a)} \end{aligned} \quad (9)$$

2.5. Pliant rendszer

A Pliant rendszer operátorai az alábbiak

$$c(\mathbf{x}) = \frac{1}{1 + \left(\sum_{i=1}^n w_i \left(\frac{1-x_i}{x_i} \right)^\alpha \right)^{1/\alpha}} \quad (10)$$

$$d(\mathbf{x}) = \frac{1}{1 + \left(\sum_{i=1}^n w_i \left(\frac{1-x_i}{x_i} \right)^{-\alpha} \right)^{-1/\alpha}} \quad (11)$$

$$a_{\nu_*}(\mathbf{x}) = \frac{1}{1 + \left(\frac{1-\nu_*}{\nu_*} \right) \prod_{i=1}^n \left(\frac{1-x_i}{x_i} \frac{1-\nu_*}{\nu_*} \right)^{w_i}} \quad (12)$$

$$n(x) = \frac{1}{1 + \left(\frac{1-\nu_*}{\nu_*} \right)^2 \frac{x}{1-x}}, \quad (13)$$

$$\kappa_\nu^{(\lambda)}(x) = \frac{1}{1 + \frac{1-\nu_0}{\nu_0} \left(\frac{\nu}{1-\nu} \frac{1-x}{x} \right)^\lambda}$$

ahol $\nu_* \in]0, 1[$, a következő generátor függvényekkel

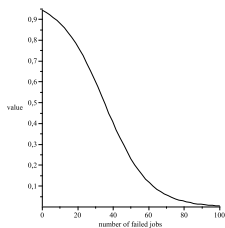
$$f_c(x) = \left(\frac{1-x}{x} \right)^\alpha \quad f_d(x) = \left(\frac{1-x}{x} \right)^{-\alpha}, \quad (14)$$

ahol $\alpha > 0$. A c , d és n operátorok teljesítik a DeMorgan azonosságot minden ν -ra. Az a and n operátorok rész-DeMorgan azonosságot adnak minden ν -ra és az aggregative operátor disztributív a szigorúan monoton t-normára vagy szigorúan monoton t-conormára.

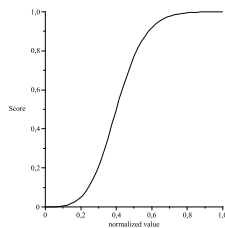
3. A disszertáció új tudományos eredményei

3.1. Döntés támogatás

A Pliant rendszert döntéstámogató algoritmusként alkalmaztam Grid rendszereknél. A disszertáció harmadik fejezet elején bemutattam a Grid rendszereket és azok alapelemeit. A Grid Metabróker egy olyan web szolgáltatás alapú elem, amely ki tudja szolgálni a felhasználókat és a grid portálokat és a brókerekkel közvetlen kapcsolatban áll. Egy újszerű ütemező algoritmus felhasználása lehetővé tesz egy magasabb szintű, együttműködő bróker használatot úgy, hogy felhasználunk már meglévő más grid rendszerben lévő brókereket. A Grid Metabróker



1. ábra. A hibás feladat mutató normalizálásának szigmoid függvénye



2. ábra. Normalizált paraméter érték a Kappa függvény segítségével

azért gyűjti össze és hasznosítja a különböző rendszerben lévő brókerek metaadatait, hogy létrehozzon egy adaptív metabroker szolgáltatást. A Meta-Broker ütemező algoritmus teljesítményét úgy tudjuk növelni, ha a feladatot annak a brókernek küldjük el, amely megfelel a feltételeknek, a feladatot hiba nélkül képes lefuttatni és a feladatot a lehető leghamarabb lefuttatja. A brókereknek négy tulajdonsága van, amelyre az algoritmus támaszkodhat: siker mutató, hiba mutató, terhelési mutató és aktuálisan futó feladat mutató.

A disszertációban bemutatok több ütemező komponenst, amely a Metabrokerhez készült. Ezen algoritmusok a fent nevezett bróker tulajdonságokat használják fel a döntéshozásban. A legjobb algoritmus "Pliant function with random generation" nevű algoritmus adta, amely még nagy bizonytalanság mellett is képes a megfelelő brókert kiválasztani a megadott feladatnak. Ez az algoritmus minden brókerhez egy pont értéket rendel és egy generátor függvényt használ a brókerek kiválasztásánál. A brókerek pont érték meghatározásakor a Kappa függvény kerül alkalmazásra. Az algoritmus főbb lépései a következők. Mivel a Pliant rendszer a $[0, 1]$ intervallumon van értelmezve ezért normalizálnunk kell a bemenő adatokat. Ez az algoritmus a szigmoid függvényt használja az adatok normalizálására.

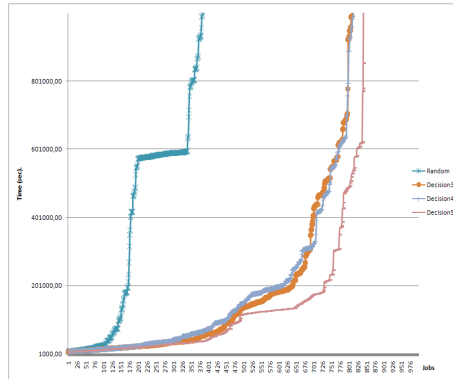
Továbbra is hangsúlyozni szeretném, hogy pont (score) érték meghatározásakor az érték minél közelebb van az egyhez, annál jobbnak számít a bróker, és az érték minél közelebb van a nullához akkor a bróker annál rosszabb. Például ha a hibás számláló magas, akkor normalizált értéknek nullához közelinek kell lennie, mert nem jó, ha egy bróker sokszor nem tudja lefuttatni a feladatot (lásd 1 képen). A "siker mutató" pont ennek ellentétje.

A következő lépésben a normalizált értéken ugyanazt a Kappa függvényt alkalmazzuk (lásd a 2 ábrán). Ennek során ν és λ paramétereket határoztuk meg, amelyekkel az elvárt értéket határoztuk meg.

Az adott pont (score) érték kiszámításához a conjunction, vagy aggregative operátort tudjuk használni. Több teszt futtatást elvégezve arra a megállapításra jutottam, hogy az aggregation operátor használatával jobb eredményt kapunk. Az eredmény minden esetben egy valós szám lesz a $[0, 1]$ intervallumban, amelyet megszoroztunk százszal, hogy megkapjuk a brókerek pont értékét.

Kifejlesztettünk egy "tanító" eljárást, hogy tovább növeljük az ütemező algoritmus hatékonyságát. Ez az eljárás a szimuláció előtt fut le azért, hogy az első és második jellemző releváns értéket tartalmazzon. Ez a kis eljárás kisméretű különböző jellemzőkkel ellátott feladatokat küld a brókereknek, hogy meg tudjuk határozni a sikeres és hibás mérőszámot.

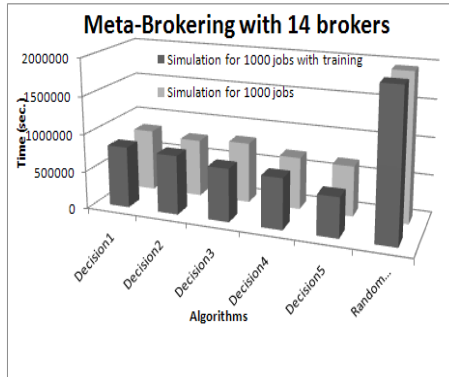
A kifejlesztett algoritmusokat Grid alapú grid szimulációs környezetben futtattuk. A szimuláció során valós terhelési adatokkal dolgoztunk.



3. ábra. Három döntéstámogató algoritmus szimulációs eredménye a véletlen döntéshozóval összehasonlítva

A tanító eljárással futtatott szimulációs eredmények láthatóak a 3 ábrán. Ahogy korábban említettem a szimuláció futtatása előtt egy tanító eljárás segítségével a brókerek jellemzőit beállítottam. Ez által az első körben beküldött feladatok gyorsabban teljesültek. Az eredményeket elemezve azt láttam, hogy a Decision 4 nevű algoritmus hasonló eredményt ért el, mint a Decision3 algoritmus, de a Decision5 nevű

algoritmus határozottan jobban teljesít a többihez képest.



4. ábra. Szimulációs eredmények összehasonlítása

A 4 ábrán felhasználásával megmutattam, hogy az egyes döntéstámogató algoritmusok milyen eredményt értek el. Az egyes oszlopok az átlagos futási értéket mutatják a szimulációknak ugyanazon paraméter használat mellett. Az eredmények azt mutatják, hogy minél szofisztikáltabb megoldást rakunk a rendszerbe annál jobb lesz a rendszer teljesítménye. Az eredmények nagy nyereséget mutattak, azaz a javított ütemező algoritmus sokkal hatékonyabb feladat futási eredményt adott.

3.2. Függvény közelítés

A disszertáció negyedik fejezetben általános közelítő eljárásokat ismerttettem. Ráműtöttem ezen eljárások néhány hiányosságára. Ezek alapján kifejlesztettem egy új típusú nem lineáris regressziós eljárást, amely a felfújó függvényen alapul és a függvény természetes leírását adja. A közelítés során a felfújó függvényt két féle módon is lehet alkalmazni. A felfújó függvény létrehozása során az egyik esetben a függvény csúcsát, a másikban egy intervallumot használtam fel. A következőben a függvény létrehozást ismertetem.

Mivel az aggregációnak van egy neutrális értéke, ezért el kell transzformálnunk az intervallumot erre $[0, \nu]$ intervallumra, vagy $[\nu, 1]$ intervallumra. Definiáltam a pozitív és negatív hatásokat, amelynek során a felfújó függvényt használtam. Ez a következő:

$$P_{a_1, a_2}^{\lambda_1, \lambda_2}(x) = \frac{1}{2} \left(1 + \gamma \sigma_{a, b}^{\lambda_1, \lambda_2}(x) \right) \quad (15)$$

$$N_{a_1, a_2}^{\lambda_1, \lambda_2}(x) = \frac{1}{2} \left(1 - \gamma \sigma_{a, b}^{\lambda_1, \lambda_2}(x) \right), \quad (16)$$

amelyben $\gamma \in [0, 1]$ nagyítás paraméter a hatás intenzitását határozza meg.

Ezt az impulzus függvényt tudjuk felhasználni a függvény interpolációhoz. A függvény létrehozására az aggregációs operátort használtam. Ha λ_1, λ_2 értékei nem túl magasak, akkor sima közelítést kaptam.

A következőben egy adott függvény felbontását fogom megmutatni ezen eszközök felhasználásával. Ebben az esetben a függvényt koordinátákkal határozzuk meg. Egy sűrű mintavételezésű eljárást használok. Minden egyes példa során 100 ekvidisztáns koordinátát használtam a megadott intervallumon. Válasszuk egy alábbi $F : R \rightarrow [0, 1]$ függvényt, amelyet közelíteni szeretnénk. A feladatunk e függvény felbontása hatásokra. Ezt a korábban ismertetett felfújó függvény, vagy az impulse függvény segítségével tudjuk megtenni. Első lépésként az $F(x)$ függvényen végrehajtottunk egy simítást.

A felfújó függvényt használó algoritmus

1. Keressük meg az $F(x)$ függvény lokális minimum és maximum pontjait

$$F(c_i) = A_i \quad \text{úgy mint} \\ F(x) < A_i \quad \text{ha } x \in (c_i - \varepsilon, c_i + \varepsilon)$$

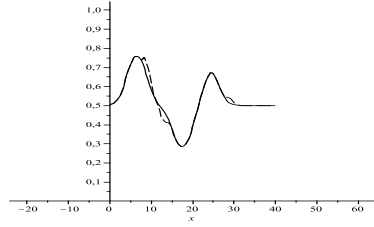
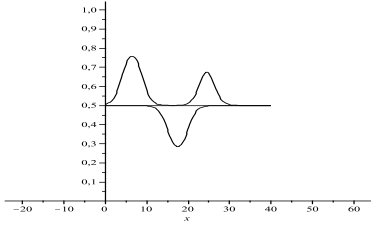
$$F(c_j) = A_j \quad \text{úgy mint} \\ F(x) > A_j \quad \text{ha } x \in (c_i - \varepsilon, c_i + \varepsilon)$$

2. Határozzuk meg az $[a_i, b_i]$ intervallumokat

$$a_1 = c_1 - \frac{c_1 + c_2}{2}, \quad b_1 = \frac{c_1 + c_2}{2}, \dots \\ a_n = \frac{c_{n-1} + c_n}{2}, \quad b_n = c_n + \frac{c_{n-1} + c_n}{2},$$

ahol

$$c_1 < c_2 < c_3 < \dots < c_k$$



5. ábra. Optimális komponensek 6. ábra. A függvény és közelítése.

3. Határozzuk meg a kezdeti λ_{i_1} és λ_{i_2} paraméter értékeket

$$\lambda_{i_1} = \frac{f(c_i) - f(a_i)}{c_i - a_i} \quad \lambda_{i_2} = 2 \frac{f(b_i) - f(c_i)}{b_i - a_i}$$

4. Határozzuk meg a kezdeti értékek alapján a függvényt

$$G_{a,b}^{\lambda_1, \lambda_2}(x) = \sum_{i=1}^n \delta_{a_i, b_i}^{\lambda_{i_1}, \lambda_{i_2}}(x)$$

Lásd a (5) ábra.

5. Ezek után pedig keressük meg az $a_i, b_i, \lambda_{i_1}, \lambda_{i_2}$ optimális értékeit a korábban meghatározott kezdeti értékeket felhasználva.

$$\min_{\mathbf{a}, \mathbf{b}, \lambda_1, \lambda_2} \sum \left(G_{\mathbf{a}, \mathbf{b}}^{\lambda_1, \lambda_2}(x_i) - F(x_i) \right)^2$$

A megtalált minimum nem biztos, hogy globális minimum, mivel azonban a kezdeti értékeket megfelelően állítottuk be, ezért a globális minimumot legtöbbször megkapjuk. A közelítés eredménye a (6) ábrán látható.

Impulzus függvényt használó algoritmus

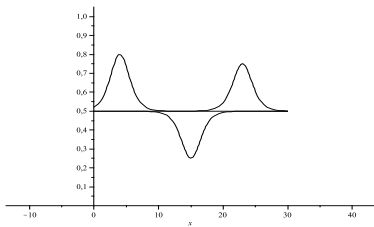
Keressük meg a minimum és maximum értékeit az $F(x)$ függvénynek

$$c_1 < c_2 < c_3 < \dots < c_k,$$

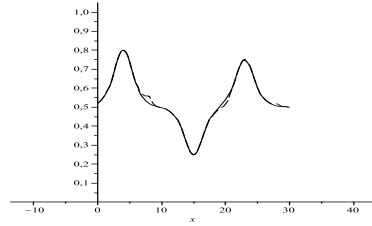
ahol c_i és c_{i+1} a minimum maximum (vagy maximum minimum) pont-párok.

Ha $f(c_i) = A_i$, legyen a közelítő függvény kezdeti értéke a következő:

$$A_i = f(c_i) - \frac{1}{2}, \quad \lambda_{i_1} = \frac{c_i - c_{i-1}}{A_i - A_{i-1}} \quad \text{és} \quad \lambda_{i_2} = \frac{c_{i+1} - c_i}{A_{i+1} - A_i}$$



7. ábra. Eredeti és optimális hatás



8. ábra. Az eredeti függvény és a közelítés eredménye.

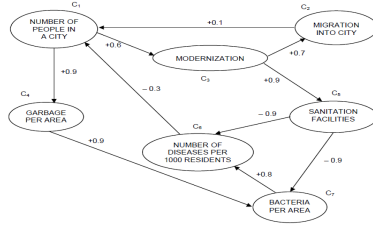
Lásd a (7) ábra. Az eljárás további lépései ugyanazok, mint a korábbi esetben.

A (8) ábrán az eljárás egyik eredménye látható. Ez az algoritmus a BFGS eljárást is használja. Így pontosabb közelítést érhető el. Megmutattam, hogy az eljárás gyors, hatékony (csak néhány lépést szükséges az optimalizáló eljárásnak) és egyszerűen használható. Ezzel a módszerrel lehetséges a függvény egy részét direkt módon megváltoztatni szemben a szokásos közelítő eljárásokkal.

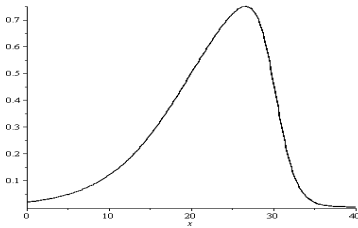
3.3. Cognitive Map

A disszertáció utolsó fejezetében a "Cognitive Map"-el foglalkozom. Célom a komplex, dinamikusan változó rendszerek modellezése volt a Pliant rendszer segítségével. A kidolgozott koncepció hasonló az irodalomban már ismert "Fuzzy Cognitive Map"-hez. Az általam kidolgozott esetben mind, az alkalmazott függvények és mind az összegzések mások. A klasszikus FCM egy hibrid megoldásnak tekinthető, amely egyszerre tekinthető fuzzy rendszernek és neurális hálónak. A rendszerben itt is állapotok és események vannak. Mindegyik numerikus értékkel jellemzett. A 9. ábrán egy tipikus FCM modellt láthatunk, amely egy irányított gráf.

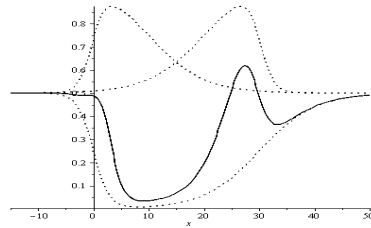
A módszer a folytonos logikán alapul és a paramétereknek szemantikus jelentése is van. A Pliant Cognitive Map-ben új típusú hatásokat tudunk definiálni. A szigmoid függvény az értékeket természetes módon a $(0,1)$ intervallumra transzformálja. Pozitív hatást úgy definiálok, hogy két szigmoid függvényre alkalmazom a conjunction operátort. Az általános pozitív, negatív hatást a kövekező függvénnyel írható le



9. ábra. Az FCM modell



10. ábra. Egy általános hatás.



11. ábra. Hatások aggregációja.

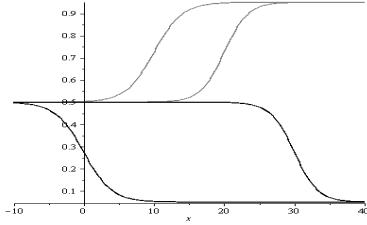
$$c(t, u, v, a, b) = \frac{1}{1 + ue^{-\lambda_1(t-a)} + ve^{-\lambda_2(t-b)}}$$

ahol u és v súlyok. A 11. ábrán egy általános hatás látható. Ha rendszerben nem definiálunk hatásokat, akkor $1/2$ értéket rendelünk az elemekhez. Pozitív hatást akkor érünk el, ha az érték $1/2$ -nél nagyobb és a hatás maximuma 1 is lehet. Negatív hatás a pozitív hatás ellentettje. Ezen hatásokat az alábbi transzformációval hoztam létre

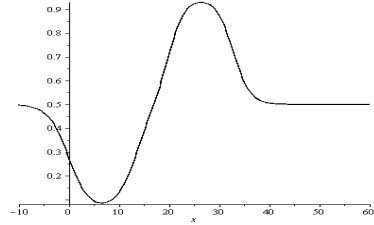
$$P(t, u, v, a, b) = \frac{1}{2} (1 + c(t, u, v, a, b))$$

$$N(t, u, v, a, b) = \frac{1}{2} (1 - c(t, u, v, a, b))$$

A 11. ábrán bemutatom a pozitív és negatív hatások aggregációját. Itt a Pliant függvényre aggregative operátorát használtam. Megmutattam azt is, hogy arra is van lehetőség, hogy csupán a szigmoid függvényeket használjak fel a hatás létrehozására. Ez esetben a modellnek más jelentése van. A módszer akkor hasznos, ha nem tudjuk a hatás hosszát. Ebben az esetben a hatást, mint egy létrejövő impulzust modellezzük. Az értékkészlet megegyezik a korábbi módszerrel, azaz az



12. ábra. Általános hatás leírása szigmoid függvénnyel.



13. ábra. Szigmoid függvények aggregációja.

1/2 érték jelzi a neutrális értéket. A pozitív hatást úgy definiáltam, hogy a szigmoid függvényt transzformáltam a $[0.5, 1.0]$ tartományra, a negatív hatás esetében pedig $[0.0, 0.5]$ -re. A fentiek alapján a hatások létrehozását az alábbi módon határoztam meg

$$P(x, a, \lambda_1) = \frac{1}{2} (1 + \sigma(x, a, \lambda_1))$$

$$N(x, a, \lambda_2) = \frac{1}{2} (1 - \sigma(x, a, \lambda_2)),$$

ahol $\lambda_1 > 0$ és $\lambda_2 > 0$. Meg kell említenünk, hogy ha a a hatás eléri a nulla vagy egy értéket, akkor a hatások aggregációja nem értelmezhető. Ennek elkerülésére a szigmoid függvény összenyomtam a $[0.15, 0.95]$ érték készletre tartományra.

A 13. ábrán bemutatom a pozitív és negatív hatások aggregációját, amelyeknél csak a szigmoid függvényt használtam fel.

PCM létrehozása A rendszer szimulációjához a hatásokat kell aggregálnunk. Összefoglalva a szimuláció végrehajtásához az alábbi lépések kell megtenni:

1. Gyűjtsük össze a koncepciókat.
2. Határozzuk meg az egyes koncepciókhoz tartozó elvárt értékét (Az aggregative operátor küszöbértéke lesz).
3. Építsük fel a "Cognitive Map"-et, azaz határozzuk meg a koncepciók közötti irányított éleket.
4. Határozzuk meg a hatásokat (minden egyes élre adjuk meg, hogy az pozitív, vagy negatív).

Az iterációs eljárás:

1. Olvassuk be a bemenő adatokat, amelyet meghatározhat egy függvény, vagy pontpárok sorozata.
2. Számítsuk ki a pozitív és negatív hatások értékét.
3. Alkalmazzuk az aggregative operátort a hatások összegzésére, ahol ν_0 értéknek a C_j koncepció korábbi értékét válasszuk.

A disszertációban bemutattam az általam kifejlesztett keretrendszert. Példákon keresztül mutattam be, hogy hogyan működik a "Pliant Cognitive Map". Végezetül megmutattam, hogy a rendszer klasszikus hőcserélő példán is megfelelően működik. A hőcserélő a vegyi és feldolgozó ipar egy általános eszköze [5]. Ez egy speciális tartály, ahol a hő kontrollálása komoly problémákat jelent, mivel különböző feltételeket kell egyidejűleg figyelembe venni. A rendszer nem lineáris viselkedésű és erősen függ a hűtőközeg áramlástól és köztes réteg hőmérsékletétől. Erre a problémára a klasszikus FCM modell is alkalmazható [6]. A dolgozatomban a PCM modellt alkalmaztam erre a problémára. A szimuláció kiértékelése során megállapítottam, hogy a szimulációs lépések közötti értékváltozások egyenletesebben csökkennek, amely a szabályozás szempontjából jóval jobb, azonban több lépés szükséges az egyensúlyi állapot eléréséhez. A valós példán a hatások értékeit nem változtattam meg. A PCM modell segítségével azonban lehetséges a hatások paramétereinek időbeni változtatása és így a valós folyamatok jobban modellezhetőek.

4. Tézispontok

A disszertációban elért eredmények három tézispontba csoportosíthatóak.

4.1. Döntés támogatás

A Pliant rendszert döntéstámogató algoritmusként alkalmaztam Grid rendszereknél. A fejezetben bemutattam az általam kifejlesztett döntéstámogató ütemező algoritmusokat, amely a Metabróker komponenshez készültek. Ezen algoritmusok a döntés meghozatalához a brókerek jellemzőit használják fel. A legjobb megoldást a "Pliant function with random generation" nevű algoritmus adta. Ez a felhasználó által beküldött feladatokhoz a rendszer nagy bizonytalansága mellett is a legjobban teljesítő brókert választotta. A tesztek valós terhelési adatokon

a GridSim szimulációs környezetben kerültek kiértékelésre. Az eredmények nagy nyereséget mutattak, azaz a javított ütemező algoritmus sokkal hatékonyabb feladat futási eredményt adott.

4.2. Függvény közelítés

A általános közelítő eljárásokat ismertettem. Ráműtattam ezen eljárások néhány hiányosságára. Ezek alapján kifejlesztettem egy új típusú nem lineáris regressziós eljárást, amely a felfújó függvényen alapul és a függvény természetes leírását is megadja. A közelítés során a felfújó függvényt kétféle módon is lehet alkalmazni. A felfújó függvény létrehozása során az egyik esetben a függvény csúcsát, a másikban egy intervallumot használtam fel. Példafüggvényeken megmutattam, hogy hogyan lehet létrehozni és felbontani az általam definiált technikával komponensekre a függvényt. Az algoritmusom a BFGS eljárást is használja. Így nagy pontosságú közelítést érhető el. Megmutattam, hogy az eljárás gyors, hatékony (csak néhány lépést szükséges az optimalizáló eljárásnak) és egyszerűen használható. Ezzel a módszerrel lehetséges a függvény egy részét direkt módon változtatni ellentétben a szokásos közelítő eljárásokkal.

4.3. Cognitive Map

Megvizsgáltam a "Cognitive Map"-et. Célom a komplex, dinamikus változó rendszerek modellezése volt a Pliant rendszer segítségével. A kidolgozott koncepció hasonló az irodalomban már ismert "Fuzzy Cognitive Map"-hez. Az általam kidolgozott algoritmus esetében mind az alkalmazott függvények és mind az összegzések mások. A módszer a folytonos logikán alapul és a paramétereknek szemantikus jelentés tulajdonítható. Két módszert is meghatároztam a hatások leírására. Megmutattam, hogy miként kell felépíteni a "Pliant Cognitive Map"-et. Továbbá bemutattam az általam kifejlesztett keretrendszert. Példákon keresztül mutattam be, hogy hogyan működik a "Pliant Cognitive Map". Végezetül megmutattam, hogy a rendszer valós példákon is megfelelően működik. A szimuláció kiértékelése során megállapítottam, hogy a szimulációs lépések közötti értékváltozások egyenletesen csökkennek, több lépés szükséges a kiegyensúlyozott állapot eléréséhez. A valós példán a hatások értékeit nem változtattam meg. A PCM mo-

dell segítségével azonban lehetséges a hatások paramétereinek időbeni változtatása és így a valós folyamatok jobban modellezhetőek.

5. A tézispontokhoz kapcsolódó közlemények

1. tézispont

- 2008: Attila Kertész, József Dániel Dombi, József Dombi: Adaptive scheduling solution for grid meta-brokering, In Acta Cybernetica 2009, Volume 19, 105-123
- 2010: József Dániel Dombi, Attila Kertész: Scheduling solution for grid meta-brokering using the pliant system, In Second International Conference on Agents and Artificial Intelligence, Valencia-Spain (2010)

2. tézispont

- 2009: J. Dombi, J. D. Dombi: Function decomposition using distending function In: Proceedings of IEEE International Workshop on Soft Computing Applications, Szeged-Hungary and Arad-Romania (2009)
- 2010: József Dániel Dombi, József Dombi: Create and decompose function by using fuzzy functions, In 12th International Conference on Enterprise Information System, Funchal-Madeira-Portugal (2010)

3. tézispont

- 2005: József Dombi, József Dániel Dombi: Cognitive Maps based on pliant logic In: International Journal of Simulations Systems, Science Technology Special Issue, April 2005, Volume 6, Number 6
- 2005: J. Dombi, J. D. Dombi: Pliant Cognitive Map using conjunctive operator In: Proceedings of IEEE International Workshop on Soft Computing Applications, Szeged-Hungary and Arad-Romania (2005)

Hivatkozások

- [1] J. Dombi. Basic concepts for a theory of evaluation: the aggregative operator. *European Journal of Operations Research*, 10:282–293, 1982.
- [2] J. Dombi. A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, 8:149–163, 1982.
- [3] Jozsef Dombi. Demorgan systems with an infinitely many negations in the strict monotone operator case. *Information Sciences*, 181(8):1440 – 1453, 2011.
- [4] R. R. Yager J. Fodor and A. Rybalov. Structure of uninorms. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 5(4):411–427, 1997.
- [5] O. Nelles M. Fischer and R. Isermann. Knowledge-based adaption of neurofuzzy models in predictive control of a heat exchanger. *Soft Computing and Intelligent Systems*, pages 469–489, 2000.
- [6] Chrysostomos D. Stylios and Petros P. Groumpos. Modeling complex systems using fuzzy cognitive maps. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 34:155–162, 2004.
- [7] R.R. Yager and A. Rybalov. Uninorm aggregation operators. *Fuzzy Sets and Systems*, 80(1):111–120, 1996.