Fuzzy szabályalapú modellek és rendszerek felépítése evolúciós technikák segítségével

BALÁZS KRISZTIÁN, KÓCZY T. LÁSZLÓ

BME Távközlési és Médiainformatikai Tanszék
{balazs, koczy}@tmit.bme.hu
Széchenyi István Egyetem, Műszaki Tudományi Kar, Automatizálási Tanszék
koczy@sze.hu

Lektorált

Kulcsszavak: fuzzy rendszerek, evolúciós algoritmusok, gépi tanulás

Jelen cikk célja rövid összefoglalást adni a fuzzy szabályalapú gépi tanuló architektúrák koncepciójáról, illetve e rendszerek evolúciós számítási technikák segítségével történő létrehozásának lehetőségéről. Ennek során a fuzzy következtetés koncepciójának ismertetése, az evolúciós algoritmusokba nyújtott betekintés, valamint az ellenőrzött gépi tanulás sémájának felvázolását követően a fuzzy szabályalapú tanuló architektúrák kialakításának mikéntje kerül tárgyalásra.
Ezek után a cikk szimulációs futtatások eredményei alapján, az ily módon létrehozott fuzzy rendszerek tömör összehasonlítását mutatja be.

1. Bevezetés

Az úgynevezett „lágy számítási” módszerek az 1960-as években jöttek létre. Hatékonyságban felülmúlják a korábbi eljárásokat nagybonyolultságú, de ugyanakkor szuboptimális megoldásokat elfogadó problématerületeken. Ennek oka az, hogy ezek a technikák viszonylag alacsony idő- és tárkomplexitással oldják meg a problémákat, illetve alkalmazhatók olyan esetekben is, amikor a probléma analitikus leírása nem, vagy csak részben ismert, illetve amikor a területről szerezhető tudás bizonytalan. Ezeknek az előnyös tulajdonságoknak az árát a megoldás pontatlanságában, szuboptimalitásában kell megfizetni. Alkalmazhatóságuk így olyan problémákra korlátozódik, melyekben a hatékonyság, a gyorsaság fontos szempont, ellenben elfogadható némi pontosságbeli hiányosság.

A lágy számítási módszerek három fő ágát a *fuzzy rendszerek,* az *evolúciós számítási technikák,* illetve a *neurális hálózatok* jelentik. Bár a fenti tulajdonságokkal mind rendelkeznek, lényeges különbség van közöttük.

A fuzzy rendszerek, valamint a neurális hálózatok jó modellezőképességűek. Alkalmasak olyan rendszerek modellezésére, melyek szerkezetére nézve kezdetben semmilyen (feketedoboz probléma), vagy hiányos ismeretek állnak rendelkezésre (szürkedoboz probléma), viszont ismertek, vagy megismerhetők egyes bemenetekre adott válaszok. Ezekből a bemenet-kimenet párokból tanulási folyamat révén fel tudnak építeni egy modellt, melynek segítségével „utánozni tudják” a megtanult rendszereket. Nagy különbség a két lágy számítási megközelítés között az, hogy amíg a neurális hálózatok a problémáról nyert tudást az elemei közti összeköttetésekben lévő súlyokban hordozzák nehezen kinyerhető és még nehezebben, vagy egyáltalán nem interpretálható formában, addig a fuzzy rendszerek eleve olyan szabálybázis építésével halmozzák fel a tudást, ahol minden információ explicit módon, interpretálható formában van jelen. Ez nagy előny a fuzzy rendszerek oldalán.

Az evolúciós számítási technikák voltaképpen sztochasztikus numerikus optimalizálási eljárások, melyek a természetben megfigyelhető különböző evolúciós folyamatok mechanizmusát próbálják egyszerűsítve visszaadni, ezzel törekedve az egyre optimálisabb megoldások felé hasonlóan, mint ahogyan az az élővilágban is megfigyelhető versengésben zajlik.

Kedvező tulajdonságaiknak köszönhetően a lágy számítási módszereken alapuló intelligens műszaki alkalmazások köre folyamatos bővülést mutat a nagybonyolultságú, szuboptimális megoldásokat elfogadó problématerületeken a robotikától [1] a különböző szabályozástechnikai [2] területeken keresztül a híradástechnikán [3] át a kémiáig [4], vagy éppen a közgazdaságtanig [5]. Ennek következményeként az e módszerek közé tartozó fuzzy szabálybázis alapú tanuló és következtető rendszerek, mint intelligens rendszerkomponensek felhasználása is nő az említett, és megannyi más területeken.

A felsoroltak közül a híradástechnikát kiemelve elmondható, hogy alkalmaznak fuzzy rendszereket a távközlésben útvonalválasztáshoz [6,7], torlódásirányításhoz [8], hibaazonosításhoz [9] és még számos más részterületen.

Tanuló és következtető rendszerekről lévén szó, minősíteni, értékelni őket a következő alapvető tulajdonságaik segítségével lehet: a tanulás és a tanulást követően a megtanult szabályok alapján történő következtetés idő-, tárkomplexitása, valamint a tanulás pontossága, illetve hibája.

Jelen cikk célja egy rövid összefoglalást adni az ilyen típusú rendszerek koncepciójáról, illetve e rendszerek evolúciós számítási technikák segítségével történő létrehozásának lehetőségéről. (Felépítésük bővebb ismertetése és mélyrehatóbb analízise megtalálható például a [10] valamint [11] publikációkban.)

Ezt a célkitűzést szem előtt tartva a következő szakasz ismerteti a fuzzy modellezés és következtetés koncepcióját, betekintést ad a numerikus optimalizálás elméletébe, ezen belül is az evolúciós algoritmusokba, valamint felvázolja az ellenőrzött gépi tanulás sémáját. Ezután, a harmadik szakaszban a fuzzy szabályalapú tanuló architektúrák kialakításának lehetősége kerül tárgyalásra. A negyedik szakasz szimulációs futtatások eredményein alapuló tömör összehasonlítását mutatja be az ily módon létrehozott fuzzy rendszereknek. A cikket egy összefoglalás zárja, amely rávilágít az ismertetett megközelítések fő tulajdonságaira és a különböző területeken való alkalmazhatóságára.

2. Az alkalmazott modellező eszközök és technikák áttekintése

A fuzzy szabályalapú tanuló és következtető rendszerek létrehozása elméleti oldalról több területen is bizonyos jártasságot igényel. Magától értetődően az egyik ilyen terület a fuzzy szabályalapú modellező és következtető rendszerek elmélete [12,13], a másik pedig a gépi tanulás [14,15]. Az előbbihez szükséges a fuzzy rendszerek alapkoncepciójának [12,16], továbbá a fuzzy következtető módszereknek, amíg az utóbbihoz a numerikus optimalizálás egyes eljárásainak az ismerete [17-22].

A következő pontok ezeknek az elméleteknek a rövid, lényegre törő bemutatását tűzik ki célul.

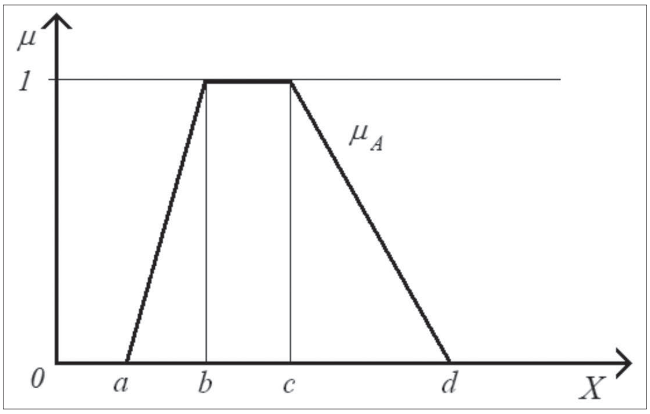
2.1. A fuzzy rendszerek alapkoncepciója

Gyakran felmerül az igény arra vonatkozóan, hogy matematikailag modellezni lehessen olyan helyzeteket, amelyekben adott tulajdonságok nem határozhatók meg teljes pontossággal, vagy nem dönthetők el teljes bizonyossággal, azaz egyfajta „bizonytalansági modellezésre” van szükség. A klasszikus halmazelmélet és az erre épülő klasszikus logika azonban nem, vagy csak nagyon körülményes módon alkalmas erre a feladatra. Ennek következtében a halmazelmélet olyan irányú általánosítása, mely egyszerűen, de hatékonyan alkalmazható eszközt ad az ilyen bizonytalansági modellezéssel kapcsolatos kihívásokra, előzőleg megoldatlan, vagy praktikusan megoldhatatlan problémákra szolgáltatott eredményt.

A *fuzzy halmazok* elmélete L. A. Zadehtől származik [16], aki az 1960-as években alkalmazta őket először bizonytalansági modellezésre. Az elgondolás a klasszikus *(crisp)* halmazelmélettel szemben nem csak azt engedi meg, hogy egy elem része legyen egy halmaznak vagy sem, hanem azt is, hogy bizonyos mértékben legyen csak része. Tehát amíg a crisp halmazelméletben egy halmaz definiálható úgy, hogy felsoroljuk az elemeit, vagy ezzel ekvivalens módon egy adott alaphalmaz minden eleméről megmondjuk, hogy az adott halmazhoz tartozó-e, addig fuzzy halmazok esetén nem csupán az elemek halmazhoz való tartozásának tényét, hanem annak mértékét is megadhatjuk. Vagyis a crisp halmazelmélettel szemben, ahol egy X alaphalmazbeli A halmazt meghatároz egy χ_A:$X \rightarrow \{0,1\}$ *karakterisztikus függvény:*

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

a fuzzy halmazelméletben az A halmazt az ő μ_A:$X \rightarrow [0,1]$ *tagsági függvénye* határozza meg.



1. ábra
Trapéz alakú tagsági függvénnyel definiált fuzzy halmaz

Az *1. ábrán* szereplő és a hozzá hasonló úgynevezett *trapéz alakú* tagsági függvények alkalmazása széles körben elterjedt. A jelen cikkben tárgyalt rendszerek is ilyen trapéz alakú tagsági függvényekkel definiálható fuzzy halmazokat alkalmaznak, illetve annak elfajult speciális eseteit: *háromszög* és *szingleton* (ez utóbbi esetén a halmaz egyelemű). A szakaszonként lineáris tagsági függvényeknek a töréspontjai a karakterisztikus pontok, melyek segítségével a függvény által leírt fuzzy halmazok könnyedén megadhatók. Az ábrán ezek az *a, b, c,* valamint *d* jelölésű pontok.

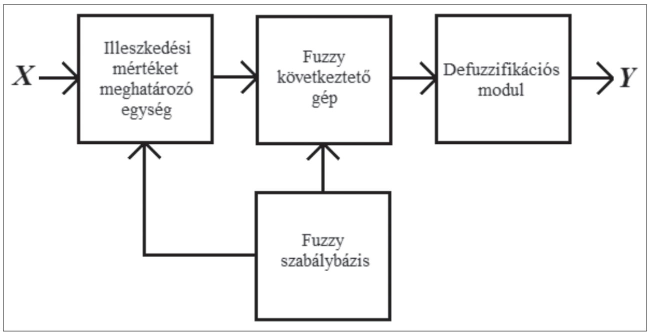
A fuzzy halmazok szolgálnak a fuzzy halmazelmélet alapjául. Segítségükkel többek között megkaphatjuk a (szűkebb értelemben vett) fuzzy logikát. (Tágabb értelemben véve minden fuzzy matematikát magába foglaló módszercsaládot szokás fuzzy logikának nevezni [23].)

Eszerint a fuzzy logikában kézenfekvő módon leírható például egy olyan állítás, hogy valamely formula „félig-meddig” igaz, vagy egy tulajdonság „többé-kevésbé” illik egy elemre, hiszen mind az igazság fogalma, mind pedig egy adott tulajdonsággal való rendelkezés relációja visszavezethető halmazokba való tartozásokra.

2.2. Fuzzy szabálybázis alapú következtető rendszerek

A fuzzy szabálybázis alapú következtető rendszerek egy adott k dimenziós X *bemeneti problématér* (alaphalmaz) egy fuzzy részhalmazához (a rendszer bemenete) – ami természetesen speciális esetben lehet az alaphalmaz egyetlen eleme is – rendelik hozzá az Y *kimeneti tér* egy fuzzy részhalmazát, illetve a defuzzifikáció (lásd lejjebb) után a kimeneti tér egy elemét. Tehát felfoghatók egy $\mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ függvényként (ahol $\mathcal{P}(X)$ és $\mathcal{P}(Y)$ rendre X, illetve Y fuzzy hatványhalmazai).

Mivel minden többkimenetű függvény triviális módon felírható egykimenetű függvények összességeként, ezért a továbbiakban csak egy kimeneti dimenziós rendszerekről esik szó.



2. ábra
A szabályalapú fuzzy következtető rendszerek felépítése

A fuzzy szabályalapú következtető rendszerek általános felépítését az *2. ábra* [12] mutatja.

Az *illeszkedési mérték* meghatározása a rendszer bemenetének (megfigyelés) az összehasonlítását jelenti a *szabálybázisban* szereplő *szabályok* (R_i, $i=1...n$) *feltétel (antecedens)* részével. A *következtető gép* az egyes szabályok *következmény (konzekvens)* részéből egy *eredő következményt (következtetés, konklúzió)* hoz létre annak megfelelően, hogy az egyes szabályok feltétel része milyen mértékben illeszkedett a bemenetre. A *defuzzifikációs* modul szerepe az, hogy az eredő következményként kapott fuzzy halmazból a kimeneti tér egy elemét, vagyis egy crisp (nem fuzzy) értéket állítson elő.

A szabályalapú fuzzy következtető rendszerek nagy előnye a klasszikus következtető rendszerekkel szemben többek között az alacsonyabb számítási komplexitás (sok bemenetű modellek esetén is), a következtetési szabályok könnyű interpretálása, valamint a modellezendő rendszer analitikus leírásának szükségtelensége. Az első kedvező tulajdonság abból ered, hogy a szabályokban szereplő tagsági függvények rendszerint egyszerűen kezelhető tulajdonságokkal bírnak, valamint a szabályok kiértékeléséhez könnyen elvégezhető műveletek szükségesek.

E rendszerek másik előnye, a könnyű interpretálhatóság abból ered, hogy a következtetési szabályokban úgynevezett „nyelvi változók” szerepelnek, vagyis a szabályokban az egyes bemenő paraméterek értékei természetes nyelven megfogalmazott „értékekkel” hasonlítódnak össze (melyek valójában fuzzy halmazok tagsági függvényei). A szabályok úgynevezett *Mamdani-féle, ortogonálisan dekomponált* alakja lehetőséget ad a szabályok feltétel részeinek és a megfigyelésnek a dimenziónkénti összevetésére, kialakítva az egyes szabályokhoz tartozó következtetéseket [24]:

$$\begin{aligned} R_i\text{: } &\mathbf{ha}\ X_{i,1} \text{ megegyezik } A_{i,1} - \text{gyel é}\mathbf{s} \ldots \\ &\text{é}\mathbf{s}\ X_{i,k} \text{ megegyezik } A_{i,k} - \text{val} \\ &\mathbf{akkor}\ Y_i \text{ megegyezik } B_i - \text{vel} \end{aligned}$$

Ez a tulajdonság más tanuló architektúrákkal (például a neurális hálózatokkal) összehasonlítva vitathatatlanul előnyös.

A harmadik tulajdonság oka az, hogy a fuzzy szabálybázis kinyeréséhez egyáltalán nem szükséges ismernünk a modellezendő struktúrát, ugyanis vagy egy szakértő (aki a modellezendő folyamatról, rendszerről szabályokként megfogalmazható tapasztalatokkal rendelkezik) bevonásával, vagy pedig úgynevezett *„tanítóminták”* (bemenet-kimenet párok) segítségével hozzuk létre a szabálybázist. Az utóbbi esetben (ellenőrzött) gépi tanulásról beszélünk (lásd 2.4. szakasz). Természetesen a két lehetőség egyike sem nyújthat pontos modellt, mivel sem a szakértő tudása, sem a tanítóminták nem fedhetnek le minden lehetőséget, továbbá mind a szakértői tapasztalatok, mind pedig a tanítóminták zajosak, pontatlanok.

Ennek ellenére törekedhetünk rendszerünk minél nagyobb pontosságára, hibájának minimalizálására, a létrehozott szabálybázisban szereplő paraméterek finom változtatásával. Ezt a folyamatot *hangolásnak* nevezzük, amit végezhetünk manuálisan, illetve automatizáltan optimalizáló technikák segítségével.

2.3. Numerikus optimalizálás

A *numerikus optimalizálás* feladata egy (kényszerfeltételek által meghatározott) halmaz azon p_{opt} optimális pontjának a megkeresése, amely pontban egy adott f_{opt} célfüggvény a globális optimumát (feladattól függően ez lehet maximum vagy minimum) felveszi. Tehát a cél egy globális szélsőérték-keresési feladat elvégzése.

Erre a feladatra léteznek determinisztikus, illetve sztochasztikus, valamint analitikus és iteratív eljárások is. Minél bonyolultabb, változatosabb a minimalizálandó függvény, annál inkább az iteratív eljárások nyernek teret az analitikusakkal szemben és minél több lokális minimummal rendelkeznek, annál inkább a sztochasztikusak érvényesülnek a determinisztikusak ellenében.

Az iteratív algoritmusok közül hatékony determinisztikus eljárások az úgynevezett *gradiens-módszerek,* mint például a legmeredekebb lejtő, momentum módszer, konjugált gradiens eljárás, Newton-módszer, vagy a Levenberg-Marquardt algoritmus. Sikeres sztochasztikus eljárások az úgynevezett *evolúciós számítások,* mint például az evolúciós stratégiák, evolúciós programozás, genetikus-, pszeudo-bakteriális-, vagy bakteriális algoritmusok, illetve a részecske-sereg módszer.

2.3.1. Gradiens-módszerek

A gradiens-módszerek lényege az, hogy az adott f_{opt} célfüggvényen elfoglalt aktuális p pozícióban kiszámoljuk a függvény gradiensét, majd a kapott értéket felhasználva „odébblépünk” a függvényen, vagyis módosítjuk p értékét azzal a céllal, hogy minél optimálisabb (feladattól függően minél nagyobb, illetve minél kisebb) függvényértéket kapjunk.

A lépegetések eredményeként kellően sok iterációt követően a gradiens-módszerek a kiindulási pozícióhoz eső „legközelebbi” lokális optimumot meglehetősen pontosan megtalálják, azonban a globális optimum eléréséhez, annak valamilyen környezetéből kell indulniuk.

2.3.2. Evolúciós számítási módszerek

Bizonyos optimalizálási technikák a természetben megfigyelhető evolúciós folyamatok absztrakt leutánzásai, ezért összefoglalóan *evolúciós számításoknak* nevezzük őket. Céljuk a *„populáció”* olyan formálása, ami során egyre „jobb” *„egyedek”* jönnek létre. Ha az „egyedeket” (vagy *„kromoszómákat”*) egy probléma egy adott megoldásának, a „populációt” a megoldások egy részhalmazának, a *„jóságot” („fitness”)* pedig az adott megoldás optimalitásának feleltetjük meg, akkor az evolúciós számítások célja nem más, mint egy problémára az optimális megoldás(ok) megtalálása.

Ennek érdekében először létrehozandó egy kezdeti populáció, ami történhet az egyedek véletlenszerű generálásával, vagy esetleg egy korábbi populáció felhasználásával. Ezt követően minden iterációs ciklusban *(„generációban”)* a technikák az úgynevezett *evolúciós operátorokat,* vagy *evolúciós műveleteket* alkalmazzák a populáció egyes egyedein, vagy az egészén. Ezek során az egyedek egyes *„génjei”* (a kromoszómáknak, vagyis a megoldásoknak az elemi részei) megváltoztatják értéküket. Új egyedek, úgynevezett *„utódok” („leszármazottak”)* is kialakulhatnak a populációban lévő kromoszómák felhasználásával. Azokat az egyedeket, melyek segítségével újak jönnek létre, *„szülőknek”* nevezzük. Az operátorok szerepe az is, hogy meghatározzák az egyedeknek azt a részét, amelyik átjut a következő generációba. *Elitista stratégia* alkalmazása során minden generáció populációjának legjobb egyede túlél, vagyis átjut a következő generációba. Ezzel garantált, hogy a mindenkori legjobb egyed nem veszik el, azaz valóban az optimalizálás folyamán adódott legjobb megoldást kapjuk meg optimálisként.

Ha az evolúciós algoritmusokban az egyedek különböző optimalizálandó p_i paramétervektorokat, a gének a különböző vektorok komponenseit reprezentálják, a fittness-érték pedig nő a célfüggvényen való jobb érték felvételekor, akkor e technikák segítségével numerikus optimalizálást végezhetünk. A továbbiakban tekintsük a kromoszómákat paramétervektoroknak.

2.3.3. Memetikus algoritmusok

Az evolúciós számítási technikák jellegükből adódóan feltérképezik az egész célfüggvényt, így kellően sok iterációt követően eljutnak minden lokális optimum közelébe. Azonban az egyes lokális optimumokhoz meglehetősen lassan konvergálhatnak.

A kétféle említett módszertípusok (gradiens és evolúciós) kombinációjával is dolgozhatunk, ha például egy genetikus algoritmus minden iterációjában valamennyi kiválasztott egyedre végrehajtunk egy gradiens eljárást, vagy ha ugyanezt megtesszük egy bakteriális algoritmus minden iterációjában valamennyi egyedre. Az előbbieket *memetikus* algoritmusoknak [21], az utóbbiakat pedig *bakteriális memetikus* [22] algoritmusoknak nevezzük.

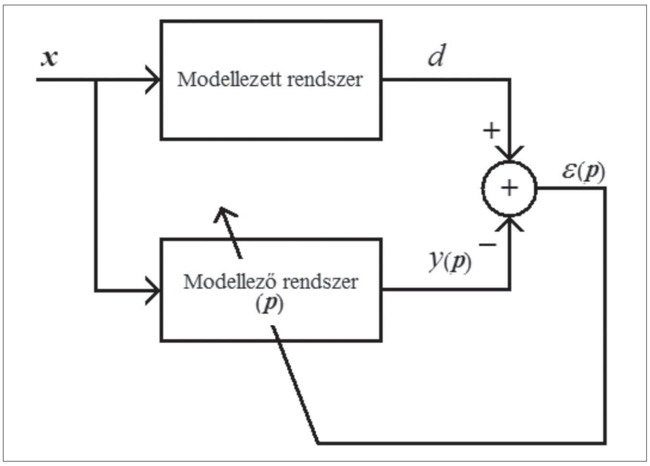
Ezeknek az az előnye adódik, hogy minden egyed bekerül a hozzá legközelebbi lokális optimumba. Ezzel ötvözni tudjuk a két megközelítés előnyeit, vagyis megtaláljuk a lokális optimumokat elég pontosan (kellően sok gradiens iteráció után) az egész célfüggvényen (kellően sok genetikus iteráció után). Tehát meglehetősen nagy pontossággal kiadódik a globális optimum, azaz az optimális p_{opt} paramétervektor.

2.4. Gépi tanulás

A *gépi tanulás* [14,15] elmélete arra törekszik, hogy módszereket adjon ismeretek, készségek automatizált elsajátítására. Eredményei számos területen alkalmazhatók. Ilyen például a műszaki, vagy orvosi diagnosztikai felismerési feladatok (képfelismerés, beszédfelismerés), folyamatszabályozás, vagy akár az előrejelzés. Valójában ezek a területek mind egy olyan közös, általános problémának a speciális esetei, melynek megoldása a gépi tanulás alapvető célja. A gépi tanulásról ugyanis általánosságban elmondható, hogy egy *„modellező rendszer”* paramétereinek hangolását jelenti annak érdekében, hogy viselkedése minél jobban hasonlítson a *„modellezett rendszer”* viselkedéséhez.

Ezt a viselkedést bemenet-kimenet párokkal jellemezhetjük. Attól függően, hogy milyen ismereteink vannak a modellezett rendszerről, különböző tanulási formák léteznek. Ha a rendszer struktúrájára nézve nincs előismeretünk, *feketedoboz-rendszermodellezésről* beszélünk. Ebben az esetben a tanulás úgynevezett *tanítóminták* segítségével történik. A tanítóminták lehetnek akár bemenet-kimenet párok *(felügyelt tanítás)*, vagy csak bemenetek *(nem ellenőrzött tanítás)*. Lehetséges, hogy a tanítóminta-halmazban bemenettel rendelkező és nem rendelkező minták is vannak *(félig ellenőrzött tanítás)*. Előfordulhat, hogy csak ritkán és akkor is csak pontatlan visszajelzést kapunk a bemenetekre *(megerősítéses tanítás)*. Az esetek mind különböző megközelítéseket kívánnak.

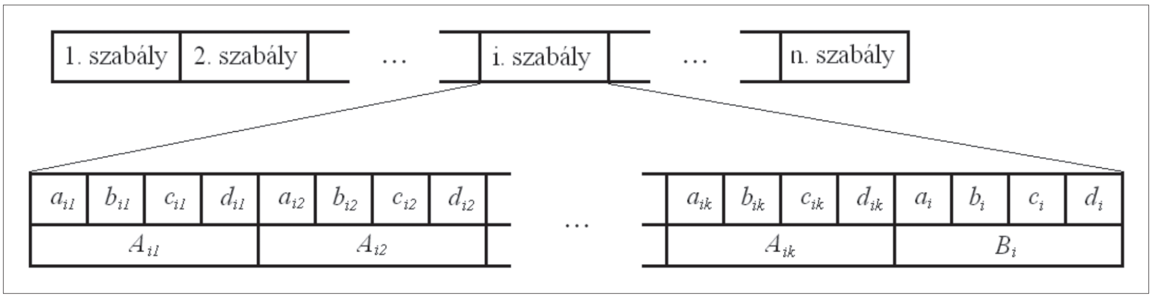
Jelen cikk ellenőrzött tanítású rendszerekkel foglalkozik. E rendszerek felépítését a *3. ábra* mutatja.



3. ábra Az ellenőrzött tanítású rendszerek felépítése

A modellező rendszer felparaméterezésétől (p) függő *hiba* ($\varepsilon(p)$) arról ad számot, hogy mennyire járunk közel a célunkhoz, azaz mennyire hasonlít a modellező a modellezett rendszerre. Értékét különböző módokon definiálhatjuk. Például *m* számú tanítóminta esetén tekinthetjük hibaként a modellező architektúra egyes bemeneti mintákra (x_i) adott válaszainak ($y_i(p)$) a megkívánt értékektől (d_i) vett távolságai négyzetösszegének számtani közepét *(Mean Squared Error, MSE)*:

$$\varepsilon(\boldsymbol{p}) := \frac{1}{m}\sum_{i=1}^{m}(d_i - y_i(\boldsymbol{p}))^2$$



4. ábra Szabálybázis kódolása kromoszómába

Ezek után a tanulás célja átfogalmazható azzá a törekvéssé, hogy ezt a p paramétervektortól függő hibafüggvényt minimalizáljuk. Ez pedig nem más, mint egy numerikus optimalizálási feladat, melynek megoldására többek között az előzőekben tárgyalt módszerek alkalmasak.

3. Fuzzy szabályalapú gépi tanuló rendszerek kialakítása

A modellező rendszer tulajdonképpen felfogható egy *tudásbázisként* és egy hozzá kapcsolódó *következtető gépként*. A tudásbázis valamilyen előre meghatározott struktúrában tárolja a tanulás folyamán „megszerzett tudást”, a következtető gép pedig egy adott megfigyelés hatására az „aktuális tudásnak” megfelelően egy következtetést végez a tároló struktúra szerint. A p paramétervektor a tudásbázis egyes elemeinek az értékét tartalmazza. Tehát például ha a modellező rendszer egy neurális hálózat, akkor a tudásbázis a neuronok struktúrája szerint tárolja az élsúlyokat (amelyek a p vektor elemei), a következtető gép pedig a hálózat válaszát adja.

Fuzzy szabálybázis alapú tanulás esetén a modellező rendszer következtető gépe egy fuzzy következtető gép, amely szabályalapú következtetést végez, tudásbázisa pedig egy fuzzy szabálybázis, melyben a paraméterek a fuzzy szabályok tagsági függvényeit definiálják (lásd 2.1. alszakasz). Például ha ezek a tagsági függvények trapéz alakúak, a p paramétervektor elemei megfeleltethetők a trapézok törés-, vagyis karakterisztikus pontjainak.

A tudásbázist leíró p paramétervektor optimális értékének megkeresését, azaz a szabálybázis hangolását numerikus optimalizálással tehetjük meg, többek között a determinisztikus legmeredekebb lejtő, illetve Levenberg-Marquardt eljárásokat, a sztochasztikus genetikus, bakteriális, illetve részecske-sereg evolúciós algoritmusokat, vagy például az előzőek kombinációjaként létrehozható memetikus technikákat alkalmazva. A keresendő optimum a globális minimum, hiszen a cél a tanuló rendszer hibájának minimalizálása a p paramétervektor megfelelő megválasztásával.

A gradiens módszerek alkalmazása a tanulási folyamat során kézenfekvő.

Az evolúciós algoritmusok segítségével úgy optimalizálhatjuk a tudásbázist, vagyis minimalizálhatjuk annak hibáját, ha az egyedeket különböző p_i paramétervektoroknak, a géneket a különböző vektorok komponenseinek feleltetjük meg, a fitness-függvényt pedig úgy definiáljuk, hogy az nőjön a rendszer hibájának csökkenésével.

Mivel az egyedek egy-egy (potenciálisan optimális) szabálybázist reprezentálnak, szükséges meghatározni egy megfeleltetést *(kódolást)* a kromoszómák génjei és e szabálybázisok között.

Trapéz alakú tagsági függvények esetén egy lehetséges kódolás a következő [22]. Mivel a szabályok karakterisztikus pontjainak adunk értéket az optimalizálás során, valamennyi egyedben minden egyes gén egy-egy karakterisztikus pontot ír le. Az egymás utáni gének trapézokat, azok pedig szabályokat határoznak meg kiadva a szabálybázist. Tehát az első négy gén az első szabály első dimenzióját magadó trapéz karakterisztikus pontjait jelenti, a következő négy a következő dimenziót megadó trapézt írja le, és így tovább.

A kódolást a *4. ábra* mutatja, ahol a_{ij}, b_{ij}, c_{ij}, illetve d_{ij} az i-edik szabály j-edik bemeneti dimenziójának, a_i, b_i, c_i, valamint d_i az i-edik szabály kimeneti dimenziójának a karakterisztikus pontjai.

Az algoritmusok nem tudják figyelembe venni, hogy az egymást követő gének között milyen viszonynak kell lennie. Ebből kifolyólag ügyelni kell az evolúciós operátorok következményeként esetlegesen létrejövő úgynevezett *abnormális* fuzzy halmazokra, vagyis azokra az esetekre, amikor a trapézok csúcsai rossz sorrendbe kerülnek (például a jobb alsó csúcsnak kisebb az értéke, mint a bal alsónak). Ez a nem kívánatos jelenség az imént részletezett génkódolás esetén elkerülhető azzal, ha azokon a helyeken, ahol esély van a kialakulásukra, egy sorrendezés kerül végrehajtásra.

4. A fuzzy szabályalapú, tanuló architektúrákat összehasonlító szimulációs vizsgálatok

Ebben a szakaszban a különböző fuzzy következtetési technikákon és optimalizálási algoritmusokon alapuló tanulási eljárások szimulációs futtatások segítségével történő összehasonlításának körülményei, valamint az eredmények tömör összefoglalása kerül ismertetésre.

4.1. A szimulációs vizsgálatok körülményei

A futtatások során a legmeredekebb lejtő és Levenberg-Marquard algoritmusok önállóan (tehát evolúciós algoritmussal történő párosítás nélkül) nem kerültek alkalmazásra, ugyanis pusztán gradiens technikák használata értelmetlen, hiszen azok a rendszert a hibafelületen legfeljebb csak a legközelebbi lokális minimumig tudnák eljuttatni.

A szimulációk futtatására három gépi tanulási problémán került sor: a kémia területéről származó egydimenziós, úgynevezett pH [22], a robotikában felmerülő kétdimenziós, úgynevezett inverz koordináta transzformációs (ICT) [22] és egy hatdimenziós függvény approximációs feladaton, amelyet a Nawa–Furuhashi szerzőpáros alkalmazott cikkében [20] a Bakteriális Evolúciós Algoritmus kiértékelésére. Bár ezek egyszerű, alacsony dimenziós problémák, mégis alkalmasak a vizsgált fuzzy rendszerek karakterisztikáinak összehasonlítására.

A futtatások során megfigyelésre kerültek többek között az aktuális populációk legjobb egyedeinek a fitness-értékei az idő függvényében. Ezen értékek a tanítómintákon mért MSE-n alapuló alábbi fitness-definíció által adódtak:

$$F(\boldsymbol{p}) = \frac{10}{\mathrm{MSE}(\boldsymbol{p}) + 1}$$

Az *5. ábra* a hatdimenziós tanulási probléma esetén mutatja a legjobb egyedek fitness-értékeinek időbeli lefolyását. A szaggatott vonalak a tisztán (gradiens lépések nélküli) evolúciós eljárásokat (genetikus, bakteriális és részecske-sereg), a pontozott vonalak a legmeredekebb lejtő technikát alkalmazókat, míg a folytonos vonalak a Levenberg-Marquardt algoritmust használókat jelzik.

4.2. A szimulációs eredmények összefoglalása

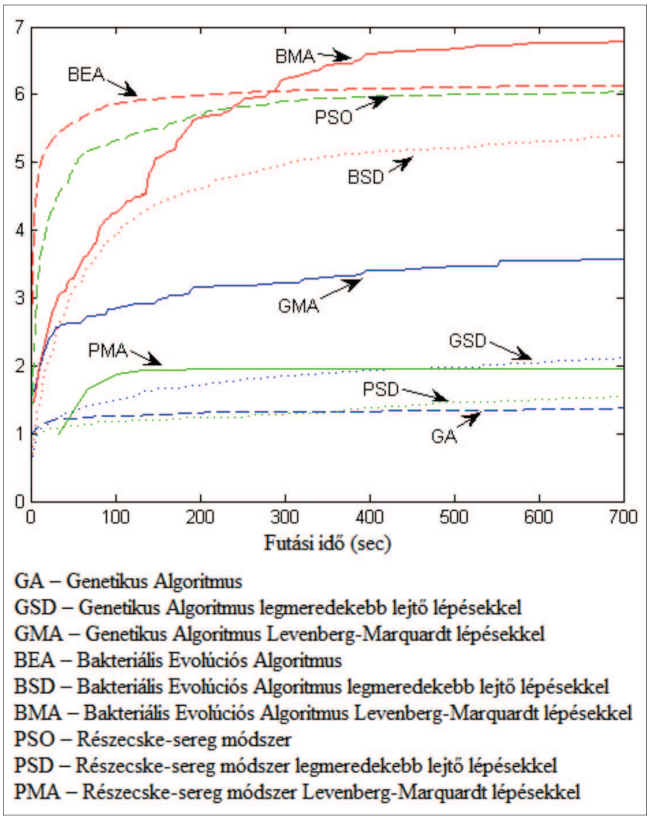
Az elvégzett szimulációs futtatások alapján a következő fő tendenciák figyelhetők meg:

- A bakteriális technikák jobbnak mutatkoztak, mint a megfelelő genetikus és részecske-sereg módszerek.
- A memetikus algoritmusok (főként a Levenberg-Marquardt gradiens lépéseket alkalmazók) hatékonyabbnak bizonyultak, mint a gradiens lépések nélküli, tisztán evolúciós technikák.
- Általában egy adott futási idő után a bakteriális evolúciós algoritmus alapú memetikus módszer (BMA) nem volt rosszabb, mint bármely más technika, és minél bonyolultabbnak mutatkozott egy feladat, annál inkább kiadódott e módszer elsőbbsége *(5. ábra)*.

5. Összefoglalás

A fentiekben a fuzzy rendszerek alapkoncepciójából kiindulva a fuzzy következtetésekbe, valamint a numerikus optimalizálás evolúciós módszereibe történő betekintést és az ellenőrzött gépi tanulás sémájának ismertetését követően a fuzzy szabályalapú tanuló rendszerek evolúciós technikák révén történő kialakításának lehetősége került tárgyalásra, melynek részét képezte egy, a kialakított rendszereket összehasonlító szimulációs vizsgálat.

Mint ahogyan azt a szimulációk során alkalmazott kémiából átvett pH és robotikából származó ICT problémákra adott eredmények is alátámasztják [10], a létrehozott fuzzy rendszerek számos tudományterület (természettudományok, műszaki tudományok, orvostudomány, gazdaságtudomány stb.) művelői számára hasznos modellezési segédeszközként szolgálhatnak.

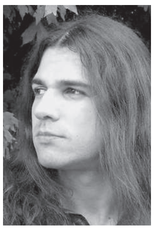


5. ábra
A legjobb egyedek fitness-értékeinek időbeli alakulása a hatdimenziós probléma esetén

Köszönetnyilvánítás

A kutatást az OTKA K75711, a TÁMOP 421 B és a SZE Kutatási Főirány Program támogatta.

A szerzőkről



BALÁZS KRISZTIÁN a Budapesti Műszaki és Gazdaságtudományi Egyetemen szerzett műszaki informatikus diplomát 2009-ben. Azóta a BME Távközlési és Médiainformatikai Tanszékének állami ösztöndíjas doktorandusza. Kutatási területe a fuzzy rendszerek és az evolúciós algoritmusok. Ipari tapasztalatokkal rendelkezik algoritmustervezési területen. Az International Fuzzy Systems Association, a Neumann János Számítógép-tudományi Társaság és a Magyar Fuzzy Szövetség tagja.

KÓCZY T. LÁSZLÓ a Budapesti Műszaki és Gazdaságtudományi Egyetemen szerzett villamosmérnöki (1975) és szakmérnöki (1976) diplomát, egyetemi doktori (1977), műszaki kandidátusi (1989), valamint MTA doktora (1998) címet. 1975 óta a BME, 2001 óta a győri Széchenyi István Egyetem oktatója. 2002 és 2011 között az SZE Műszaki Karának dékánja. Jelenleg egyetemi tanár. Vendégprofesszorként oktatott többek között az ausztráliai ANU, Murdoch és UNSW, a japán TIT, a koreai POSTECH, az osztrák J. Kepler, valamint az olasz trentói egyetemeken. Kutatási és oktatási területe a fuzzy rendszerek és más lágy számítási módszerek (evolúciós algoritmusok, neurális hálózatok). A területen több mint 450 publikációval rendelkezik. Társszerkesztője volt az IEEE Transactions on Fuzzy Systems folyóiratnak és jelenleg is társszerkesztője a Fuzzy Sets and Systems, International Journal of Fuzzy Systems, Journal of Advanced Computational Intelligence, Mathware and Soft Computing stb. folyóiratoknak. Az International Fuzzy Systems Association (IFSA) volt elnöke, az IEEE Computational Intelligence Society intézőbizottságának volt tagja, az IEEE Systems Council tagja.

Irodalom

[1] Kuttan, A.,
"Robotics",
I. K. International Pvt Ltd., 2009, p.336.

[2] Bubnicki, Z.,
"Modern control theory",
Springer, 2005, p.423.

[3] Ghosh, S., Razouqi, Q., Schumacher, H.J., Celmins, A.,
" A survey of recent advances in fuzzy logic in telecommunications networks and new challenges",
IEEE Transactions on Fuzzy Systems,
Vol. 6, No. 3, pp.443–447., 1998.

[4] Rouvray, D.H.,
"Fuzzy Logic in Chemistry",
Academic Press, 1997, p.364.

[5] Dymowa, L.,
"Soft Computing in Economics and Finance",
Springer, 2011, p.295.

[6] Mirabedini, S.J., Teshnehlab, M., Rahmani, A.M.,
"FLAR: An Adaptive Fuzzy Routing Algorithm for Communications Networks Using Mobile Ants",
Int. Conf. on Convergence Information Technology,
Gyeongju, South Korea, pp.1308–1315., 2007.

[7] Tanaka, Y., Hosaka, S.,
"Fuzzy control of telecommunications networks using learning technique",
Electronics and Communications in Japan,
Vol. 76, No. 12, pp.41–51., 1993.

[8] Pitsillides, A., Sekercioglu, A.,
"Fuzzy logic based Congestion control",
COST 257: Impacts of new services on the architecture and network performance of broadband networks,
Larnaca, Cyprus, 1999.

[9] Bellec, J.-H., Kechadi, M.-T.,
"Fuzzy Event Correlation Algorithm in Wide Telecommunication Networks",
Int. Journal of Multimedia and Ubiquitous Engineering,
Vol. 3, No. 2, pp.103–116., 2008.

[10] Balázs, K., Botzheim, J., Kóczy, L. T.,
"Comparative Analysis of Interpolative and Non-interpolative Fuzzy Rule Based Machine Learning Systems Applying Various Numerical Optimization Methods",
World Congress on Computational Intelligence,
WCCI 2010, Barcelona, Spain, pp.875–982., 2010.

[11] Balázs, K., Kóczy, L.T.,
"Hierarchical-interpolative Fuzzy System Construction by Genetic and Bacterial Programming Algorithms",
IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE),
Taipei, Taiwan, pp.2116–2122, 2011.

[12] Kóczy T. L., Tikk D.:
Fuzzy rendszerek,
TypoTEX, Budapest, 2000.

[13] Michels, K., Klawonn, F., Kruse, R., Nürnberger, A.,
"Fuzzy Control Fundamentals",
Stability and Design of Fuzzy Controllers Series: Studies in Fuzziness and Soft Computing,
Vol. 200, Springer, 2006.

[14] Altrichter M., Horváth G., Pataki B., Strausz Gy., Takács G., Valyon J.:
Neurális hálózatok,
Panem Kiadó, Budapest, 2006.

[15] Alpaydin, E.,
"Introduction to Machine Learning",
The MIT Press, 2004.

[16] Zadeh, L.A.,
"Fuzzy sets",
Inf. Control, Vol. 8, pp.338–353., 1965.

[17] Levenberg, K.,
" A method for the solution of certain non-linear problems in least squares",
Quart. Appl. Math., Vol. 2, No. 2, pp.164–168., 1944.

[18] Marquardt, D.,
"An algorithm for least-squares estimation of nonlinear parameters",
Journal Soc. Indust. Appl. Math.,
Vol. 11, No. 2, pp.431–441., 1963.

[19] Holland, J.H.,
"Adaption in Natural and Artificial Systems",
The MIT Press, Cambridge, Massachusetts, 1992.

[20] Nawa, N. E.. Furuhashi, T.,
"Fuzzy system parameters discovery by bacterial evolutionary algorithm",
IEEE Transactions on Fuzzy Systems,
Vol. 7, No. 5, pp.608–616., 1999.

[21] Moscato, P.,
"On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms",
Technical Report Caltech Concurrent Computation Program, Report. 826, California Inst. of Technology,
Pasadena, California, USA, 1989.

[22] Botzheim, J., Cabrita, C., Kóczy, L.T., Ruano, A. E.,
"Fuzzy rule extraction by bacterial memetic algorithms",
In Proc. of the 11th World Congress of International Fuzzy Systems Association, IFSA 2005,
Beijing, China, pp.1563–1568., 2005.

[23] Zadeh, L.A.,
"Fuzzy logic and the calculi of fuzzy rules, fuzzy graphs, and fuzzy probabilities",
Computers & Mathematics with Applications,
Vol. 37, pp.35–41., 1999.

[24] Mamdani, E.H.,
"Application of fuzzy algorithms for control of simple dynamic plant",
Procedings of IEEE, Vol. 121, No. 12., pp. 1585–1588, 1974.