Production Systems and Information Engineering
Volume 13 (3), pp. 1-10
doi: https://doi.org/10.32968/psaie.2025.3.1

KLASZTEREZÉSI MÓDSZEREN ALAPULÓ FUZZY SZABÁLYBÁZIS REDUKÁLÁS A FRIQ-LEARNING RENDSZERBEN

TOMPA TAMÁS
Miskolci Egyetem
Informatikai Intézet
Általános Informatikai Intézeti Tanszék
tamas.tompa1@uni-miskolc.hu

KOVÁCS SZILVESZTER
Miskolci Egyetem
Informatikai Intézet
Általános Informatikai Intézeti Tanszék
szilveszter.kovacs@uni-miskolc.hu

Absztrakt. A Fuzzy logikán alapuló megerősítéses tanulási módszerek tudásábrázolási formája fuzzy szabályok alkotta fuzzy szabályrendszer által leírt. A tanulási folyamat után előállt szabálybázis szabályainak száma, azaz a tudásbázis mérete meghatározza a rendszer komplexitását, számításigényét. Ennek következtében a szabálybázis méretének optimalizálása, az esetleges elhagyható szabályok kiszűrése kulcsfontosságú ezen tanulási módszerek hatékonyságának növelése érdekében. A túlzottan nagyméretű szabálybázis növelheti a számítási igényt és csökkentheti a döntéshozatali folyamat átláthatóságát. A cikk célja egy klaszterezési eljáráson alapuló fuzzy szabálybázis-redukciós módszer bemutatása és alkalmazása a FRIQ-learning megerősítéses tanulási rendszerben, amely a releváns szabályok kiemelésével lehetővé teszi a szabálybázis méretének csökkentését, miközben a rendszer döntéshozatali képessége változatlan marad.

Kulcsszavak: megerősítéses tanulás, Q-learning, Fuzzy Q-learning, tudásbázis redukálás, klaszterezés

1. Bevezetés

A klaszterezés [11] egy olyan adatbányászati módszer [10], amely célja az adatok csoportosítása olyan módon, hogy a hasonló tulajdonságokkal rendelkező adatelemek egy klaszterbe kerüljenek, míg a jelentősen eltérőek külön csoportokat alkossanak. Mivel a klaszterezés nem igényel előre meghatározott címkéket (mint az osztályozás [5]), a nem felügyelt tanulási algoritmusok [3] közé tartozik, és kizárólag az adatok közötti hasonlóság alapján végzi el a csoportosítást. Az adatok közötti hasonlóság mértéke jellemzően távolságfüggvényekkel [9] mérhető, a közel álló adatpontok hasonló tulajdonságokat mutatnak, míg a távol esők jelentősen különböznek. Az adatok közötti távolság mérésére többféle metrika létezik, például

az Euklideszi-, Manhattan-, Minkowski- és Hamming-távolság [9]. A klaszterezési technikák számos területen alkalmazhatók, többek között a képfeldolgozásban, a piackutatásban, a bioinformatikában és az egészségtudományban [4][8]. A különböző klaszterezési algoritmusok, például a K-means, a hierarchikus klaszterezés vagy a DBSCAN eltérő módszereket kínálnak az adatok szerkezetének feltárására és a rejtett mintázatok azonosítására [11].

A FRIQ-learning (Fuzzy Rule-Interpolation based Q-learning) [19][21] és a HFRIQ-learning (Heuristically Accelerated Fuzzy Rule Interpolation based Q-learning) [15][17] olyan fuzzy szabály-interpoláción [7][13] alapuló megerősítéses tanulási (reinforcement learning) [12] módszerek, amelyek tudásbázisa fuzzy szabálybázis által leírt. A probléma megoldását leíró fuzzy szabálybázis a tanulási folyamat végeztével áll elő és „ha-akkor" típusú fuzzy szabályokat tartalmaz, ahol a „ha" rész a szabály antecedense (állapot-akció érték), az „akkor" rész pedig a szabály konzekvense (Q-értéke). Ezen szabálybázis tartalmazhat olyan szabályokat, amelyek csak a tanulási folyamat során voltak relevánsak, így ezen szabályok törlésével adott esetben csökkenthető a végső szabálybázis mérete.

A cikk célja, egy olyan klaszterezési módszeren alapuló fuzzy szabálybázis redukálási algoritmus bemutatása, amely alkalmas a tanulási folyamat végeztével előállt szabálybázis méretének (azaz a szabályok számának) csökkentésére a lényegi szabályok kiemelésével és az elhagyható szabályok törlésével. A javasolt megközelítés koncepcionálisan hasonlít a Multi-Vantage Point Tree (MVP-tree) módszerhez [1], amely több referenciapont (vantage point - VP) alapján végzi az elemek csoportosítását és tér rekurzív felosztását, ezáltal csökkentve a redundanciát és növelve a hasonlóság keresés hatékonyságát. A fejlesztett módszer hatékonysága különböző megerősítéses tanulási mintapéldák (Cart-Pole, Mountain Car) alkalmazásán keresztül kerül bemutatásra.

2. A FRIQ-learning és a HFRIQ-learning megerősítéses tanulási módszerek

A FRIQ-learning (Fuzzy Rule-Interpolation based Q-learning) [19][21] és ennek a heurisztikusan gyorsított (szakértői tudásbázissal kiegészített) verziója (Heuristically Accelerated Fuzzy Rule-Interpolation based Q-learning - HFRIQ-learning) [15][17] olyan fuzzy szabály-interpoláció alapú Q-tanulási módszerek, amelyek a „FIVE" (Fuzzy Interpolation in the Vague Environment) [7] fuzzy szabályinterpolációs eljárást alkalmazzák a Q-függvény leírására. Ezen FRI (Fuzzy Rule Interpolation) módszer alkalmazása következtében a rendszer tudásbázisa egy ritka szabálybázis által leírt, amelyben egy r_i ($i \in [1, m]$) szabály formátuma az m méretű R szabálybázisban a következő [19][21]:

$$r_i: \textbf{\textit{If}}\ s_1\ is\ S_1^i\ \textbf{\textit{And}}\ s_2\ is\ S_2^i\ \textbf{\textit{And}}\ \dots\ \textbf{\textit{And}}\ s_n\ is\ S_n^i\ \ \textbf{\textit{And}}\ a\ is\ A^i\ \textbf{\textit{Then}}\ \tilde{Q}(\boldsymbol{s}, a) = q^i \qquad (1)$$

ahol $\boldsymbol{s} \in \boldsymbol{S}$ az n-dimenziós állapot megfigyelés, A^i az i-edik szabály egydimenziós akció univerzumának (U) fuzzy halmaza, $a \in U$ az akció, S_j^i az i-edik ($i \in [1, m]$) szabály j-edik ($j \in [1, n]$) állapot dimenziójának fuzzy halmaza az n-dimenziós $\boldsymbol{S}$ állapottérben, s_j a j-edik dimenziója az $\boldsymbol{s}$ állapot megfigyelésnek, $\tilde{Q}(\boldsymbol{s}, a)$ a FIVE FRI [7] által becsült Q-függvény. Az i-edik szabály antecedense az $s_1\ is\ S_1^i\ \textbf{\textit{And}}\ s_2\ is\ S_2^i\ \textbf{\textit{And}}\ \dots\ \textbf{\textit{And}}\ s_n\ is\ S_n^i\ \ \textbf{\textit{And}}\ a\ is\ A^i$, konzekvense (Q-értéke) pedig a q^i.

A FIVE FRI modellel közelített $\tilde{Q}(\boldsymbol{s}, a)$ függvény i-edik fuzzy szabályának konzekvense a $(k+1)$-edik iterációban a következő [19][21]:

$$q_i^{k+1} = \begin{cases} q_i^k + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) & \text{ha } (\boldsymbol{s}, a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re,} \\ q_i^k + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) * \left(1/\delta_{v,i}^{\lambda}\right) \Big/ \left(\sum_{i=1}^{m} 1/\delta_{v,i}^{\lambda}\right) & \text{egyébként} \end{cases} \quad (2)$$

Ahol q_i^k-edik az i-edik fuzzy szabály konzekvense a k-dik iterációban, $\delta_{v,i}^{\lambda}$ a skálázott távolság az ($\boldsymbol{s}$, a) állapot-akció megfigyelés és az i-edik szabály ($\boldsymbol{s^i}, a^i$) állapot-akció antecedense között, λ a Shepard parameter, $\Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a)$ a Q-függvény $(k+1)$-edik iterációbeli frissítési értéke az ($\boldsymbol{s}$, a) állapot-akció párban, amely a következő módon határozható meg [19][21]:

$$\tilde{Q}^{k+1}(\boldsymbol{s}, a) = \tilde{Q}^k(\boldsymbol{s}, a) + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) \quad (3)$$

$$\Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) = \alpha * \left(g(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma * \max_{a' \epsilon U} \tilde{Q}^k(\boldsymbol{s}', a') - \tilde{Q}^k(\boldsymbol{s}, a)\right) \quad (4)$$

Ahol γ a leszámítolási tényező, $\alpha \in [0,1]$ a tanulási ráta, q_i^{k+1} az i-edik szabály singleton konklúziója a $(k+1)$-edik iterációban, a a végrehajtott akció $\boldsymbol{s}$-ben, $\boldsymbol{s}'$ az új állapot megfigyelés, $g(\boldsymbol{s}, a, \boldsymbol{s}')$ a jutalom értéke az $\boldsymbol{s} \to \boldsymbol{s}'$ állapotátmenetre, $\tilde{Q}^k$ a k-adik, $\tilde{Q}^{k+1}$ pedig a $k+1$-edik iteráció becsült konklúziója (Q-értéke) a FIVE FRI által.

A FRIQ-learning tanulási fázisa kezdetben 2^{n+1} darabszámú, 0 konzekvens értékkel rendelkező fuzzy szabállyal indul, amit az inkrementális szabálybázis építési módszer [19][21] iterációról-iterációra bővít, illetve hangol. Ezek a kezdeti vagy úgynevezett sarokponti szabályok az $(n+1)$-dimenziós hiperkocka sarkaiban, azaz az univerzumok határain helyezkednek el. A továbbiakban a módszer ezt a kezdeti szabálybázist bővíti új szabályokkal vagy azok konzekvensét (Q-értékét) hangolja a (2) formula által, attól függően, hogy új szabály beillesztésére vagy csak a meglévő szabálybázis Q-értékének a frissítésére van-e szükség.
Új szabály szabálybázisba történő beszúrása az ágens környezetéből érkező megerősítési információk és a Q-függvény frissítési értékei $(\Delta\tilde{Q})$ alapján történik. Ha $\Delta\tilde{Q}$ értéke magasabb, mint egy előre meghatározott Q-frissítési limit $(\Delta\tilde{Q} > \varepsilon_Q)$ és a létező legközelebbi szabály is távol van az éppen beszúrandó szabály pozíciójához képest, akkor új szabály felvétele történik az adott lehetséges szabálypozícióba. Abban az esetben, ha $\Delta\tilde{Q}$ értéke kisebb, mint az előre meghatározott Q-frissítési limit $(\Delta\tilde{Q} < \varepsilon_Q)$, akkor nem történik új szabály beszúrása a szabálybázisba. Ebben az esetben a teljes szabálybázis konzekvensének, azaz Q-értékének a frissítése (hangolása) valósul meg. Az említett lépések minden egyes iterációban végrehajtásra kerülnek addig, amíg a tanulási fázis (azaz az inkrementális szabálybázis építési fázis) véget nem ér. Akkor áll elő a rendszert működtető, végleges tudásbázis (fuzzy szabályrendszer) és ér véget a tanulási folyamat, ha már nem kerül új szabály beszúrásra a szabálybázisba és a $\Delta\tilde{Q}$ frissítési értékek relatívan kicsik maradnak.
A HFRIQ-learning [15][17] ezen FRIQ-learning módszer kiterjesztése, amely alkalmas szakértő által megadott tudásbázis injektálására [14][16] a tanulási folyamatba, majd a megadott szakértői szabályrendszer hangolására [17] és redukálására [17][18] a tanulási folyamat során. Azáltal, hogy ezen módszer esetében már a tanulási folyamat során megtörténik a szabálybázis redukálás egy szabálytávolság-alapú módszer által [17][18], így ebben az esetben nem feltétlenül

van szükség további szabálybázis redukálásra.

2.1. FRIQ-learning szabálybázis redukálási módszerek

A FRIQ-learning inkrementális szabálybázis építési fázisában létrejött szabálybázis tartalmazhat redundáns szabályokat vagy olyan szabályokat melyeknek kiadódhatnak más, már létező szabályokból. Ezen szabályok a szabálybázisból való végleges törlésével a szabálybázis mérete csökkenthető [20], csökkentve ez által a szabálybázis komplexitását és a működtető végleges tudásbázis méretét. Az elhagyható szabályok megállapítására eredetileg 3 dekrementális tudásbázis redukálási módszerrel (I., II., III.) [20][21] rendelkezik a FRIQ-learning rendszer, amelyek az inkrementális szabálybázis építési fázis után alkalmazhatók opcionálisan. Ezen módszerek mindegyike a tanulási folyamat végén előállt teljes szabályrendszer szabályait vizsgálja, hogy az egyes szabályok lényegiek (kardinális), vagy kiadódók (redundáns). A szabálybázis redukálási módszerek eltávolítják a redundáns szabályokat a szabályrendszerből, így az eredetivel közel azonos az információt hordozó szabályrendszert alkotnak a lényegi szabályokból. A redukciós módszerek közös jellemzője, hogy a szabályok konzekvens értékét, azaz a Q-értéket vizsgálja. Az I.-III. redukálási módszerek dekrementálisak, azaz a végső redukált szabálybázis a tanulási fázis végén kapott teljes szabálybázis egyes szabályainak elhagyásával jön létre, fokozatosan csökkentve annak méretét. Az egyes szabálybázis redukálási módszerekkel kapott csökkentett méretű szabálybázisok közel ugyanazt a Q-függvényt (irányítási felületet) írják le, mint a redukálás előtti esetben, de kevesebb szabállyal (azaz interpolációs tartóponttal).

Az I. jelölésű szabálybázis redukálási stratégia [20][21] azon szabályokat törli a teljes szabálybázisból, amelyeknek abszolútértékben alacsony a Q-értékük (konzekvens értékük). Minden egyes szabály törlése után megvizsgálja, hogy az adott szabályt elhagyva a probléma még megoldható-e és ha igen, akkor folytatja a folyamatot. Ezt addig teszi, amíg az adott szabály törlése után kapott eredmény nem tér el lényegesen az azt megelőzőtől. Ha lényegesen eltér, azaz a feladat már nem oldható meg, akkor a törölt szabályt visszahelyezi a szabálybázisba és fontos szabályként jelöli meg. Ellenkező esetben azonban véglegesen törli azt a szabálybázisból.

A II. jelölésű tudásbázis redukálási módszer [20][21] hasonló, mint az I., de azzal a különbséggel, hogy ebben az esetben a legnagyobb Q-értékkel rendelkező szabályok kerülnek vizsgálatra, feltételezve, hogy a nagyobb Q-értékkel rendelkező szabályok jelentősebb befolyással bírnak.

A III. jelölésű szabálybázis redukciós módszer [20][21] nem egyesével vizsgálja a szabályokat, hanem szabálycsoportokat alakít ki, majd ezeket távoltja el. A szabálycsoportok kialakítása szintén Q-érték alapján történik, a módszer meghatározza a Q-értékek teljes tartományát (legkisebb és legnagyobb érték közötti értérést), majd ezen tartomány alapján hoz létre két szabálycsoportot, úgy hogy a tartomány fele lesz a tűréshatár. Ezt követően a nagyobb Q-értékkel rendelkező szabálycsoport kerül kiértékelésre. Ha ezzel a probléma még sikeresen megoldható, akkor az ebből a megmaradt szabálycsoportból kiindulva ismétlődik az eljárás. Ha nem oldható meg sikeresen, akkor a törölt szabályok visszakerülnek a szabályrendszerbe, de a vizsgált Q-érték tartomány újra megfelezésre kerül. Ez addig ismétlődik, amíg a tűréshatár értéke olyan nem lesz, hogy az adott szabálycsoport már eltávolítható. Abban az esetben, ha a tűréshatás alapján csak

egyetlen szabály marad a csoportban és a probléma így sem oldható meg, akkor ez a szabály fontos (állandó) jelölést kap, majd a továbbiakban ezen állandónak jelölt szabályokat már nem vizsgálja.

3. Klaszterezésen alapuló fuzzy szabálybázis-redukálási módszer

A fejlesztett módszer egy hierarchikus klaszterezési eljáráson alapszik, mely a tanulási folyamat végeztével előállt szabálybázis szabályait klaszterekbe (és alklaszterekbe) rendezi, majd az egyes klaszterekből az adott klasztert jellemző lényegi (kardinális) szabályokként kiemeli a legnagyobb és a legkisebb Q-értékű szabályokat. A módszer úgy csökkenti a teljes szabálybázis méretét, hogy az egyes klaszterek (és alklaszterek) lényegi szabályait tartja csak meg, a többi szabályt pedig elhagyja a szabálybázisból. A szabálybázis szabályai ebben az értelemben, mint objektumok, azaz az adathalmaz adatpontjaiként tekinthetők.

Az algoritmus első lépésként egy D távolságmátrix kerül meghatározásra, mely a szabálybázis minden egyes szabálya közötti távolságot tartalmazza. Az egyes adatpontok (szabályok) közötti távolságok meghatározása a „FIVE” FRI módszer által alkalmazott többdimenziós Euklideszi-távolság alapú távolságszámítási eljáráson [6] (jelöljük FIVE_dist-el) alapszik. A számított távolságmátrix egy négyzetes $(m * m)$ mátrix, melynek mérete a szabálybázis szabályainak a számától (m) függ. A mátrix főátlójában csupa nullák helyezkednek el, annak következtében, hogy egy szabálypont saját magától vett távolsága mindig 0. A szabálybázis két i-edik (r_i) és j-edik (r_j) indexű szabálya (azaz adatpontja) közötti távolságot d_{ij} jelölti, ahol $i, j = [1, \dots, m]$:

$$D_{[ij]} = d_{ij} = FIVE\_dist(r_i, r_j) \tag{5}$$

A következő fázisban a távolságmátrix alapján p_1, p_2 pivot objektumok (a klaszterezési módszer speciális pontjai, jelen esetben fuzzy szabályok) kerülnek meghatározásra, amelyek a két egymástól legtávolabbi szabályok lesznek. Ezen p_1 és p_2 pivot pontok megfelelnek az MVP-tree módszer vantage point-jainak. Tulajdonképpen a D távolságmátrix d_{ij} távolságértékei közöl a legkisebb és a legnagyobb távolságértékkel rendelkező i, j indexű szabályokpárok, amelyek között a távolság maximális:

$$(p_1, p_2) = arg \max_{d_{ij} \in D} (d_{ij}) \tag{6}$$

A következő iterációban a szabálybázis minden egyes szabályának a távolsága kerül kiszámításra a p_1 és p_2 adatpontokól. Ezen távolságszámítás szintén a távolságmátrix meghatározásánál is alkalmazott „FIVE” FRI távolságszámítási módszerének [6] az alkalmazásával történik:

$$d(r_i, p_1) = FIVE\_dist(r_i, p_1) \tag{7}$$

$$d(r_i, p_2) = FIVE\_dist(r_i, p_2) \tag{8}$$

A szabálybázis szabályai a pivot objektumoktól való távolságuk alapján kerülnek

besorolásra klaszterbe. A klaszterhez való hozzárendelés alapja egy ε távolságküszöb, amely a következőképpen számítható:

$$\varepsilon = \frac{\max_r(d(r_i, p_1))/2 + \max_r(d(r_i, p_2))/2}{2} = \frac{\max_r(d(r_i, p_1)) + \max_r(d(r_i, p_2))}{4} \quad (9)$$

Az ε távolságküszöb értéke tehát, amely alapján az egyes klaszterek (majd alklaszterek) kerülnek kialakításra, a p_1 és p_2 adatpontoktól vett legtávolabbi távolságértékek felének az összegei, majd az ezek alapján számított átlag, azaz a két klaszter „sugár" átlagának a fele. Ezen ε küszöbérték alapján a klaszterekhez való hozzárendelés a következő:

$$Cluster_\varepsilon(r_i) = \begin{cases} C_{left}, & ha\ d(r_i, p_1) \le \varepsilon \\ C_{right}, & egyébként \end{cases} \quad (10)$$

Kezdetben a teljes szabálybázis minden egyes szabálya egyetlen klasztert alkot, majd ez a kezdeti klaszter kerül felosztásra további két alklaszterre minden egyes lépésben (felosztó, top-down klaszterezés). Az alklaszterekre (right branch, left branch) való bontás alapja a (9) formula alapján számított ε távolságküszöb. Minden egyes szabály távolsága meghatározásra kerül a p_1 és p_2 adatpontoktól, majd ha az éppen vizsgált szabály távolsága az adott p_1 vagy p_2 pivot elemnél kisebb, mint a számított ε távolságküszöb értéke, akkor az adott szabály a bal alklaszterhez (left branch), ellenkező esetben pedig a jobb alklaszterhez (right branch) kerül hozzáadásra. Az egyes iterációkban kialakult alklaszterek szabályai közül a legnagyobb és a legkisebb Q-értékkel (konzekvenssel) rendelkező szabályok kerülnek fontos szabályként megjelölésre annak következtében, hogy a rendszer működésének hatékonyságát a legrosszabb Q-érték (illetve megerősítés) is befolyásolja [2]. Tehát, az adott iterációban kialakult alklaszterek szabályai közül mindig a legkisebb és a legnagyobb Q-értékkel rendelkező szabályok kerülnek lényegi szabályként megjelölésre (és a teljes szabálybázisból eltávolításra), majd egy átmeneti, kezdetben üres szabálybázishoz történő hozzáadásra, amely mérete ezért inkrementálisan növekszik. Így a fontos szabályként megjelölt szabályok kikerülnek az egyes klaszterekből és bekerülnek a fontos szabályok halmazába. A folyamat a kialakult alklaszterekre rekurzívan ismétlődik, az alklaszterek újabb alklaszterekre lesznek felosztva, majd a lényegi szabályok kiemelve. Ez a folyamat addig ismétlődik, amíg a lényegi szabályokat tartalmazó (átmeneti) redukált szabálybázis meg nem oldja az adott problémát, azaz az így kapott megerősítés értéke nagyobb, mint egy előre definiált megerősítési küszöbérték.

A fejlesztett fuzzy szabálybázis redukálási algoritmus pszeudokódja az alábbi:

Algoritmus: clusteringBasedRBreduce(R)

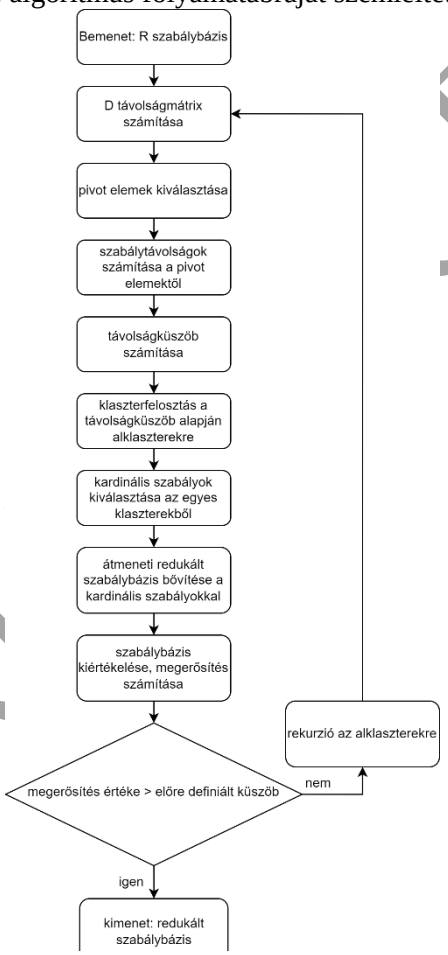
Bemenet: szabálybázis
Kimenet: redukált szabálybázis

1. D távolságmátrix számítása
2. p_1 és p_2 pivot objektumok meghatározása a számított távolságok alapján
3. szabályok távolságának számítása p_1 és p_2 objektumoktól
4. ε távolságküszöb számítása

5. ε távolságküszöb alapján szabálybázis felosztása bal és jobb alklaszterekre
6. legnagyobb és legkisebb Q-értékű (kardinális) szabályok kiválasztása az egyes alklaszterekből
7. a 6. lépésben meghatározott kardinális szabályok hozzáadása az átmeneti redukált szabálybázishoz
8. megerősítés számítása a 7. lépésben létrejött átmeneti redukált szabálybázisra (szabálybázis kiértékelése)
9. If (számított megerősítés > előre definiált megerősítés)
   vége, kardinális szabályok megtalálva, return redukált szabálybázis
   else
   következő rekurzív lépés a bal és jobb alklaszterekre
   end

1. algoritmus: A fejlesztett, klaszterezési eljáráson alapuló szabálybázis redukálási módszer algoritmusa, amely a tanulási folyamat után alkalmazható

A következő ábra az 1. algoritmus folyamatábráját szemlélteti:



3.1. Alkalmazáspéldák

A fejlesztett algoritmus hatékonysága a „Cart-Pole” és a „Mountain Car” mintapéldák alkalmazásával kerül bemutatásra, összevetve az előzőleg bemutatott I., II. és III. jelölésű szabálybázis redukálási módszerek által kapott eredményekkel. A klaszterezési eljáráson alapuló szabálybázis redukálási módszer (IV. jelölésű) alkalmazásával az egyes futási esetekben kapott szabálybázis méretek (szabályszámok) nem feltétlenül lettek kisebbek, mint az I.-III. jelöléssel ellátott redukálási módszerek alkalmazása esetében, de a futási idejében nagy eltérés mutatkozik.

A Cart-Pole mintapélda esetében kapott futási eredményeket a következő a 1. táblázat foglalja össze:

1. táblázat: A „Cart-Pole" mintapélda esetében kapott futási eredmények

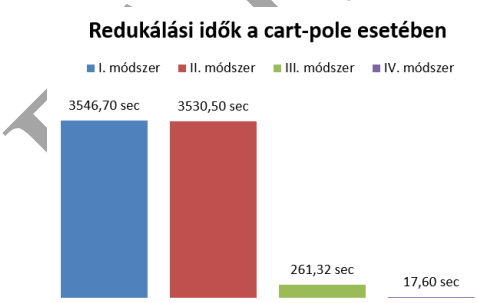
Redukciós módszer	Szabályszám	Futási idő (sec)	Sebesség
I.	7	3546,70	1x
II.	6	3530,50	1x
III.	8	261,32	13,6x
IV.	24	17,60	201,5x

A Cart-Pole mintapélda esetében a fejlesztett klaszterezési eljáráson alapuló szabálybázis redukálási módszer (IV. jelölésű) által kapott szabálybázis 24 darab szabályt tartalmaz, amely több, mint az I.-III. redukálási módszerek esetében kapott szabálybázis méretek, de a futási idejében jelentős javulás áll elő, amely így 201,5-ed része, mint az I. módszer esetében.
A Mountain Car mintapélda esetében kapott futási eredményeket a következő a 2. táblázat foglalja össze:

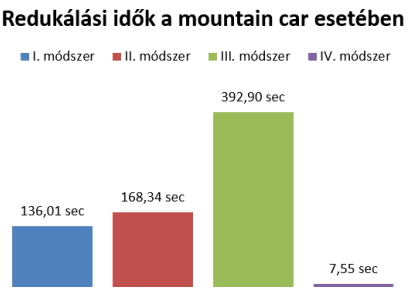
2. táblázat: A „Mountain Car" mintapélda esetében kapott futási eredmények

Redukciós módszer	Szabályszám	Futási idő (sec)	Sebesség
I.	27	136,01	1x
II.	53	168,34	0,8x
III.	26	392,90	0,3x
IV.	32	7,55	18x

A Mountain Car mintapélda esetében a javasolt szabálybázis redukálási módszer (IV. jelölésű) alkalmazásával a redukált szabálybázis 32 darab szabályt tartalmaz, amely a II. redukálási módszer esetében létrejött 53 darab szabályhoz viszonyítva kevesebb. Jelentős eltérés a szintén a futási időben mutatkozik meg, amely így 18-ad része mint az I. redukálási módszer esetében.
Az 1. és 2. ábrák a tanulási fázis után alkalmazott egyes szabálybázis redukálási módszerek által kapott futási időket szemléltetik (másodpercben):



1. ábra: Szabálybázis redukálási idők a Cart-Pole mintapélda esetében



2. ábra: Szabálybázis redukálási idők a Mountain Car mintapélda esetében

A fejlesztett szabálybázis redukálási módszer teljes futási idejének nagy részét nem a hierarchikus klaszterezési algoritmus (illetve a „FIVE" FRI távolságszámítás) lépései teszik ki, hanem az adott iterációban előálló, redukált szabálybázis tesztelése azaz, hogy az helyesen megoldja-e már az adott problémát vagy szükséges egy újabb rekurzív iteráció futtatása. A teljes szabálybázis redukálási

időt és annak a szabálybázisok tesztelésére fordított idő mértékét a 3. táblázat foglalja össze:

3. táblázat: Szabálybázis tesztelési ideje a teljes redukálási időből

Mintapélda	Teljes redukálási idő (sec)	Szabálybázis tesztelési idő (sec)	%
Cart-Pole	17,6	11,13	63
Mountain Car	7,55	6,49	86

4. Összefoglalás

A cikkben egy klaszterezési eljáráson alapuló fuzzy szabálybázis-csökkentési módszer került bemutatásra, amely a FRIQ-learning tanulási folyamatot követően lehetőséget biztosít a szabálybázis méretének optimalizálására. Az alkalmazott hierarchikus klaszterezési algoritmus a szabályok közötti távolságokat figyelembe véve alakítja ki a klasztereket, majd minden klaszteren belül a legkisebb és legnagyobb Q-értékű szabályokat kiválasztva építi fel a redukált (csökkentett méretű) szabálybázist. A kiválasztási folyamat után a módszer ellenőrzi, hogy a redukált szabályrendszerrel a probléma megfelelően megoldható-e és ezt a folyamatot mindaddig ismétli amíg a rendszer az így létrejött szabálybázissal meg nem oldja a problémát. A Cart-Pole és a Mountain Car mintapélda alkalmazása során kapott eredmények alapján elmondható, hogy a bemutatott szabálybázis-redukciós eljárás alkalmas a fuzzy szabálybázis méretőnek csökkentésére illetve végrehajtási ideje jelentősen rövidebb, mint a bemutatott I., II., és III. FRIQ-learning szabálybázis redukálási módszerek esetében.

Irodalomjegyzék

[1] Bozkaya, Tolga, and Meral Ozsoyoglu. "Distance-based indexing for high-dimensional metric spaces." Proceedings of the 1997 ACM SIGMOD international conference on Management of data. 1997.

[2] Fuchida, Takayasu, Kathy Thi Aung, and Atsushi Sakuragi. "A study of Q-learning considering negative rewards." Artificial Life and Robotics 15.3 (2010): 351-354. https://doi.org/10.1007/s10015-010-0822-7

[3] Ghahramani, Zoubin. "Unsupervised learning." Summer school on machine learning. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. 72-112.

[4] Ghosal, Attri, et al. "A short review on different clustering techniques and their applications." Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 (2020): 69-83. https://doi.org/10.1007/978-981-13-7403-6_9

[5] Gordon, Allan David. Classification. CRC Press, 1999.

[6] Kovács, Sz., Kóczy, L. T.: The use of the concept of vague environment in approximate fuzzy reasoning. Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, vol.12, 1997, pp. 169-181.

[7] Kovács, Szilveszter. "Extending the fuzzy rule interpolation" five" by fuzzy

observation." *Computational Intelligence, Theory and Applications: International Conference 9th Fuzzy Days in Dortmund, Germany, Sept. 18–20, 2006 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. https://doi.org/10.1007/3-540-34783-6_48

[8] Oyewole, Gbeminiyi John, and George Alex Thopil. "Data clustering: application and trends." Artificial intelligence review 56.7 (2023): 6439-6475. https://doi.org/10.1007/s10462-022-10325-y

[9] Pandit, Shraddha, and Suchita Gupta. "A comparative study on distance measuring approaches for clustering." International journal of research in computer science 2.1 (2011): 29. https://doi.org/10.7815/ijorcs.21.2011.011

[10] Pujari, Arun K. Data mining techniques. Universities press, 2001.

[11] Rokach, Lior, and Oded Maimon. "Clustering methods." Data mining and knowledge discovery handbook (2005): 321-352.

[12] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018. https://doi.org/10.1017/s0263574799211174

[13] Tikk, D., Johanyák, Z. C., Kovács, S., & Wong, K. W. (2011). Fuzzy rule interpolation and extrapolation techniques: Criteria and evaluation guidelines. Journal of advanced computational intelligence and intelligent informatics, 15(3), 254-263. https://doi.org/10.20965/jaciii.2011.p0254

[14] Tompa, Tamás, and Szilveszter Kovács. "Applying Expert Heuristic as an a Priori Knowledge for FRIQ-Learning." Acta Polytechnica Hungarica 17.4 (2020). https://doi.org/10.12700/aph.17.4.2020.4.2

[15] Tompa, Tamás, and Szilveszter Kovács. "Heuristically accelerated FRIQ-learning." 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY 2022). IEEE, 2022. https://doi.org/10.1109/iccc57093.2023.10178919

[16] Tompa, Tamás, and Szilveszter Kovács. "Integrating Expert Knowledge into Fuzzy Reinforcement Learning." 2024 IEEE 18th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2024. https://doi.org/10.1109/saci60582.2024.10619888

[17] Tompa, Tamás, and Szilveszter Kovács. "Knowledge Base Optimization of the HFRIQ-Learning." Acta Polytechnica Hungarica 21.10 (2024). https://doi.org/10.12700/aph.21.10.2024.10.6

[18] Tompa, Tamás, and Szilveszter Kovács. "Tudásbázis redukálás a heurisztikusan gyorsított FRIQ-learning rendszerben." Production Systems and Information Engineering 11.2 (2023): 1-12. https://doi.org/10.32968/psaie.2022.4.4

[19] Vincze, Dávid, and Szilveszter Kovács. "Fuzzy rule interpolation-based Q-learning." 2009 5th International Symposium on Applied Computational Intelligence and Informatics. IEEE, 2009. https://doi.org/10.1109/saci.2009.5136311

[20] Vincze, Dávid, and Szilveszter Kovács. "Rule-base reduction in Fuzzy Rule Interpolation-based Q-learning." Recent Innovations in Mechatronics 2.1-2. (2015): 1-6. https://doi.org/10.17667/riim.2015.1-2/10.

[21] Vincze, Dávid.: "Fuzzy Rule Interpolation-based Q-learning." PhD dissertation, 2013.