DOI: 10.14750/ME.2023.035

MISKOLCI EGYETEM

GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR

Fuzzy szabály-interpolációs állapotgép modellek és hangolási eljárásaik

Ph.D. értekezés

Készítette:

Tompa Tamás
okleveles villamosmérnök

Hatvany József Informatikai Tudományok Doktori Iskola

Doktori iskola vezető

Prof. Dr. Szigeti Jenő
egyetemi tanár

Témavezető

Prof. Dr. Kovács Szilveszter
egyetemi tanár

Miskolc
2023

DOI: 10.14750/ME.2023.035

SZERZŐI NYILATKOZAT

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

A dolgozat bírálatai és a védésről készült jegyzőkönyv a későbbiekben, a Miskolci Egyetem Gépészmérnöki és Informatikai Karának Dékáni Hivatalában lesz elérhető.

Miskolc, 2023. május 10.

_____
Tompa Tamás

A disszertáció bírálatai és a védésről készült jegyzőkönyv megtekinthető a Miskolci Egyetem Gépészmérnöki és Informatikai Karának Dékáni Hivatalában, valamint a doktori iskola weboldalán az Értekezések menüpont alatt: http://www.hjphd.iit.uni-miskolc.hu

DOI: 10.14750/ME.2023.035

TÉMAVEZETŐI AJÁNLÁS

Tompa Tamással a Miskolci Egyetem Műszaki Informatikus hallgatójaként találkoztam először 2008-ban. Mind a tudományos diákköri munkájának konzulenseként, mind az azt követő közös kutatásaink, majd PhD tanulmányainak témavezetőjeként is egy megbízható, önálló tudományos munkára alkalmas lelkiismeretes kollégaként ismertem meg.

Tompa Tamás a közös kutatásaink során célként kitűzött feladatokat sikeresen megvalósította. Az általa elért eredmények jól alkalmazhatók a fuzzy szabály alapú rendszermodellezésben, a megerősítéses Q tanulás során a szakértői szabályok felhasználásában, illetve azok validálásában.

Tompa Tamás a kutatásai során elért eredményeit magyar és angolnyelvű kiadványokban, valamint konferenciákon is publikálta.

Az értekezés tézisei Tompa Tamás saját kutatási munkájának eredményeit foglalják össze. Az értekezés és a tézisekhez kapcsolódó publikációk alapján messzemenően támogatom és javaslom Tompa Tamás számára a Ph.D. cím odaítélését.

Miskolc, 2023. május 10.

Prof. Dr. Kovács Szilveszter
egyetemi tanár

DOI: 10.14750/ME.2023.035

KÖSZÖNETNYILVÁNÍTÁS

Ezúton szeretnék köszöntet mondani **Prof. Dr. Kovács Szilveszter** témavezetőmnek az éveken át tartó segítőkészségéért, tanácsaiért, iránymutatásaiért, amely nélkülözhetetlen támogatást jelentett kutatómunkám során. Külön köszönet a bíztatásáért és bátorításáért ami sokszor lendületet adott a kutatás során.

Szeretnék köszönetet mondani továbbá mindazoknak, akik támogatásukkal, segítségükkel hozzájárultak az értekezés elkészítéséhez, külön köszönet **Dr. Vincze Dávid**nak a számtalan segítségéért illetve kollégáimnak és barátaimnak a megértésükért és kitartó támogatásukért.

Miskolc, 2023. május 10.

Tompa Tamás

DOI: 10.14750/ME.2023.035

RÖVIDÍTÉSEK JEGYZÉKE

AdaGrad	Adaptive Gradient
COA	Center Of Area
COG	Center Of Gravity
CRI	Compositional Rule of Inference
DP	Dynamic Programming
FIVE	Fuzzy Interpolation in the Vague Environment
FQ-learning	Fuzzy Q-learning
FRI	Fuzzy Rule Interpolation
FRIQ-learning	Fuzzy Rule Interpolation-based Q-learning
GD	Gradient Descent
GM	Generalized Methodology of fuzzy rule interpolation
GOAL	Goal-Oriented Agent Language
HA	Heuristically Accelerated
HAQL	Heuristically Accelerated Q-learning
HARL	Heuristically Accelerated Reinforcement Learning
KH	Kóczy-Hirota-féle lineáris fuzzy szabály-interpoláció
MDP	Markov Decision Process
MOM	Mean Of Maxima
MSE	Mean Squared Error
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
SARSA	State-Action-Reward-State-Action
SGD	Stochastic Gradient Descent
TD	Temporal Difference
VE	Vague Environment
VKK	Vass-Kalmár-Kóczy-féle fuzzy szabály-interpoláció

DOI: 10.14750/ME.2023.035

SZÓTÁR

Adaptive Gradient	adaptív gradiens módszer
Agent	ágens
Center Of Area	geometriai középpont módszer
Center Of Gravity	súlypont módszer
Compositional Rule of Inference	kompozíciós fuzzy következtetés
Crisp	diszkrét
Deep Learning	mély tanulás
Deep Reinforcement Learning	mély megerősítéses tanulás
Discount factor	diszkontálási tényező / faktor
Distance Rate	közelségarány
Dynamic Programming	dinamikus programozás
Exploration-exploitation	felderítés-kiaknázás
Fuzzy Interpolation in the Vague Environment	bizonytalan környezet alapú fuzzy szabály-interpoláció
Fuzzy Q-learning	fuzzy Q-tanulás
Fuzzy Rule Interpolation	fuzzy szabály-interpoláció
Fuzzy Rule Interpolation-based Q-learning	fuzzy szabály-interpoláció alapú Q-tanulás
Goal-Oriented Agent Language	cél-orientált ágens programozási nyelv
Gradient Descent	gradiens süllyedés
Greedy	mohó
Heuristically Accelerated	heurisztikusan gyorsított
Heuristically Accelerated Reinforcement Learning	heurisztikusan gyorsított megerősítéses tanulás
Initial state	kezdeti állapot
Learning rate	tanulási ráta
Markov Decision Process	Markov döntési folyamat
Mean Of Maxima	maximumok közepe módszer
Mean Squared Error	átlagos négyzetes hiba
Model-based	modell alapú
Model-free	modellmentes
Off-policy	politika független
On-policy	politikafüggő
Particle Swarm Optimization	részecske-raj alapú optimalizálás
Policy iteration	politika iteráció
Q-learning	Q-tanulás
Quality function	Q-függvény
Reinforcement Learning	megerősítéses tanulás

DOI: 10.14750/ME.2023.035

Singleton	egyértékű
Step	lépés (iteráció)
Stochastic Gradient Descent	sztochasztikus gradiens süllyedés
Temporal Difference learning	időbeli-különbség tanulás
Terminal state	végállapot
Trial-end-error	próbálkozás típusú
Vague Environment	bizonytalan környezet
Value iteration	érték iteráció

DOI: 10.14750/ME.2023.035

TARTALOMJEGYZÉK

DOI: 10.14750/ME.2023.035

1 BEVEZETÉS

Az egyre növekvő gépi számítási kapacitás és ennek a hétköznapi eszközökben történő megjelenése következtében a mesterséges intelligencia [80], a gépi tanulás [17][68] témaköre illetve az ezen módszerek nyújtotta lehetőségek kiaknázására való törekvés egyre inkább aktuálissá válik, egyre nagyobb jelentőséggel bír. A gépi tanulás olyan módszerek összesége, amelyek tapasztalatszerzés útján tanulnak, ezáltal lépésről-lépésre építve fel a rendszer működtető tudásbázisát. Több típusa elterjedt, vannak olyan algoritmusok amelyek külső mintaadatok (példa-halmazok, tanítóminták) alapján igyekeznek törvényszerűségeket feltárni a rendszer működésére vonatkozóan, majd ezek alapján a még „nem látott” ismeretlen helyzetekre is „helyes” döntést hozni. Másik típusa mikor nem áll rendelkezésre külső tanítóminta majd a rendszer próbálkozások és a próbálkozásokra kapott válaszok (megerősítések) alapján térképezi fel a megoldás miként jét, hozza létre a működtető tudásbázist. Ez az úgynevezett - jelenleg is népszerű és egyre jobban kutatott tudományterület - megerősítéses tanulás.

A megerősítéses tanulási módszereken alapuló rendszerek hasonló módon tanulnak, mint az ember is teszi a gyermekkorától kezdve. A környezettel való kölcsönhatásba lépés során az abból érkező megerősítési információk (amelyek lehetnek jutalmak vagy büntetések) alapján igyekszik lehetséges cselekedetei, döntései közül a legmegfelelőbbet végrehajtani, megfigyelni a környezet arra adott reakcióját (megerősítését), tapasztalatait ennek megfelelően bővíteni, hogy elérje a kívánt célt. Ezen tanulási algoritmusok összesége általában üres tudásbázissal indítja a tanulási folyamatot, ahogyan az ember is gyermek korában (kezdetben semmit sem ismer környezetből, majd törekszik felfedezni azt), majd a cselekedetekre kapott megerősítések következtében lépésről-lépésre igyekszik gyarapítani tudását, tapasztalatait. Ennek következtében ezen módszerek jól használhatók olyan rendszerekben ahol a működés egzakt folyamata nem ismert, az elérendő cél definiálása után a megerősítések alapján térképezik fel a megoldásul szolgáló működtető tudásbázist, modellt. A megoldás keresésének folyamatát és annak hosszúságát nagymértékben befolyásolja a definiált állapot-cselekvési dimenziók és a jutalomfüggvény, ezek megfelelő meghatározása kulcsfontosságú lépés. Abban az esetben ha rendelkezésre áll részleges információ a megoldás miként jére vonatkozóan, akkor ennek a megerősítéses tanuló rendszerekbe történő beépítésével felgyorsítható a teljes tanulási folyamat.

A kutatás (jelen doktori értekezés) célja egy olyan fuzzy szabály-interpoláción alapuló megerősítéses tanulási módszer továbbfejlesztése és kidolgozása, amely alkalmas emberi szakértő által megadott előzetes (*a priori*) tudásbázis (mint heurisztika) injektálásra a rendszerbe, a rendszerbe adoptált szakértői heurisztika hangolására (optimalizálására), illetve a rendszer tudásbázisát leíró fuzzy szabálybázis méretének tanulási folyamat közbeni csökkentése (redukálása). A kutatás alapjául a „Fuzzy szabály-interpoláció alapú Q-learning" [97][98] (*Fuzzy Rule Interpolation-based Q-learning - FRIQ-learning, D. Vincze, Sz. Kovács, 2009*) rendszer szolgál, a kutatás ezen Fuzzy szabályinterpoláció alapú Q-learning módszer szakértői heurisztikával való kiterjesztésére, hangolására és tudásbázisának redukálására irányul.

1.1 A KUTATÁS CÉLKITŰZÉSEI

Elsődleges cél a „Fuzzy szabály-interpoláció alapú Q-learning" (*Fuzzy Rule Interpolation-based Q-learning - FRIQ-learning*) megerősítéses tanulási módszer továbbfejlesztése oly módon, hogy alkalmas legyen egy előre megadott (a priori) szakértői tudásbázis (mint heurisztika) befogadására úgy, hogy az a későbbiekben, szükség esetén a tudásbázis többi részével együtt hangolható legyen. Ez olyan módszerek kifejlesztését jelenti, amelyek lehetővé teszik a szakértői tudásbázis valamilyen (magasabb szintű) formában történő leírását és a rendszerbe történő injektálását, és képesek a szakértői szabályrendszer hangolására és validálására (helyességének ellenőrzésére) is. Ezek alapján a kutatás célkitűzései a következőképpen fogalmazhatók meg:

- Olyan előzetes tudásbázis, szakértői szabályrendszer leírási forma kidolgozása, melyben az előzetes szakértői heurisztika megadható és a FRIQ-learning rendszerbe építhető.
- Kezdeti Q-érték becslési módszer kialakítása, amely lehetővé teszi a szakértői produkciós szabályrendszer kezdeti Q-függvényt leíró fuzzy szabályokká való alakítását és így azok FRIQ-learning rendszer tanulási folyamatába való injektálását.
- Hangolási eljárás kidolgozása, amely lehetővé teszi az előzetesen megadott szakértői tudásbázis hangolását, optimalizálását. A hangolási eljárásnak alkalmasnak kell lennie a nem feltétlenül helyes szakértői heurisztika (szabályrendszer) negatív hatásainak kompenzálására, a téves vagy nem teljesen helyes szakértői szabályok korrekciójára.
- A Q-függvényt leíró fuzzy szabálybázis szabályszámának csökkentésére (redukálására) alkalmas módszer kidolgozása, amely a tanulási folyamat közben, a közel ugyanazon

információt leíró szabályok összevonásával csökkenti a szabálybázis méretét és így a rendszer komplexitását.

- Módszer kidolgozása a Q-függvényt leíró fuzzy szabálybázis hangolása és redukciója során a szakértői szabályok követésére és hangolást követő kinyerésére. A módszerrel az eredeti és a hangolást követően kinyert szakértői szabálybázis összevethető, a kezdeti szakértői heurisztika helyessége ellenőrizhető, validálható. A hangolás előtt megadott és a hangolás utáni előállt szakértői tudásbázis összehasonlításával következtetni lehet a szakértői szabályok helyességének mértékére. A hangolást követő kismértékű eltérés igazolhatja a szakértői szabályok helyességét, nagyobb mértékű eltérés értelmezhető a kezdeti heurisztika pontosításaként, a jelentős eltérések, vagy az eredeti szabályrendszernek ellentmondó produkciós szabályok pedig utalhatnak a kezdeti heurisztika egyes részeinek helytelenségére. A szabálybázis redukciója során eltűnő szabályok a szakértői heurisztika redundanciájára utalhatnak.

A kutatási téma célkitűzése tehát kettős. Egyrészt olyan fuzzy interpolációs állapotgép viselkedésmodell kidolgozása, amelyben az a priori tudás viszonylag egyszerű módon implementálható, másrészt olyan automatikus hangolási eljárás kidolgozása, amellyel ezen a priori elemeket is tartalmazó modell hiányos minta alapján hangolható. Az így kialakítandó modell és módszer jelentősége amellett, hogy egy nyelvi leírási formából kiindulva (pl. etológiai modell, mint a priori tudás) működtető modell kialakítására alkalmas, megfelelő teljesítmény mérték választása és minták megléte esetén akár etológiai modell hangolására és akár annak validálására is lehetőséget nyújthat.

1.2 AZ ÉRTEKEZÉS FELÉPÍTÉSE

A kutatás célkitűzéseinek bemutatása után az értekezés további fejezeteiben, az adott fejezet témájához kapcsolódó szakirodalom felhasználásával bemutatásra kerül a fuzzy rendszerek felépítése, a megerősítéses tanulás, a fuzzy szabály-interpolációs módszerek illetve a kutatás alapjául szolgáló Fuzzy szabály-interpoláció alapú Q-learning (FRIQ-learning) módszer.

A 4. fejezetben kerül részletesen bemutatásra a megvalósított kutatómunka, a javasolt heurisztikusan gyorsított FRIQ-learning rendszer és a hozzá kapcsolódóan fejlesztett módszerek, algoritmusok illetve a tézisek és publikációk.

A 4.5. alfejezetben a javasolt heurisztikusan gyorsított FRIQ-learning módszer lehetséges alkalmazásai kerülnek bemutatásra elterjedt megerősítéses tanulási mintapéldák által.

Az 5. (és a 6. angol nyelvű) fejezetben összefoglalásra kerülnek az értekezésben részletezett, a tématerületen elért saját, új tudományos eredmények, a tézisek bemutatásával és a további célkitűzések ismertetésével.

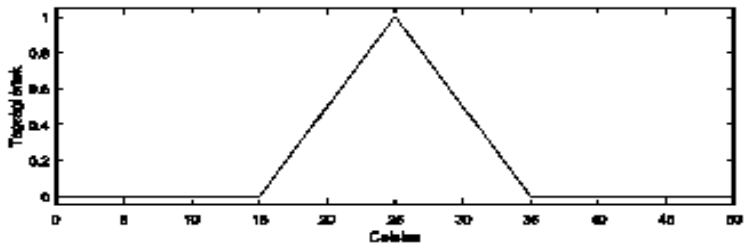
A 7. és 8. fejezet tünteti fel az értekezéshez kapcsolódó szakirodalmi jegyzéket illetve a saját publikációkat és az azokra történő hivatkozásokat.

2 FUZZY RENDSZEREK

A fuzzy logika, azaz az elmosódott halmazok logikájának matematikai alapjait Lotfi A. Zadeh vezette be 1965-ben [102]. Ezen elgondolás alapján az emberi gondolkodási módhoz közelálló, de matematikailag is egzaktul leírható halmazelméleti modellt definiált. A fuzzy halmazelmélet szerint nincs éles elkülönítés egy elem halmazba való tartozásának megadására, tehát nem az éles "vagy eleme a halmaznak vagy nem eleme a halmaznak" lehetőségek vannak, hanem az elmélet szerint minden elem beletartozik a halmazba de különböző mértékben. A halmazba való tartozás mértékét és egyben magát a fuzzy halmazt egy μ tagsági függvény [102] jellemzi, amely $[0,1]$ intervallumban definiálja az adott halmaz elemeinek halmazba való tartozásának mértékét. Egy *A* jelölésű fuzzy halmaz és egy *X* jelölésű univerzum esetében ez $\mu_A: X \rightarrow [0,1]$ által definiálható [53]. Ezen módszer segítségével leírható például, hogy egy adott hőmérséklet mikor számít hidegnek, melegnek vagy éppen kellemesnek. A következő példa a *'meleg'* fuzzy halmaz leírását szemlélteti:

$$\mu_{meleg} = \begin{cases} 1, & if\ hőmérséklet(x) = 25 \\ \left(\dfrac{hőmérséklet(x) - 15}{25 - 15}\right), & if\ 15 < hőmérséklet(x) < 25 \\ \left(\dfrac{35 - hőmérséklet(x)}{35 - 25}\right), & if\ 25 < hőmérséklet(x) < 35 \\ 0, & if\ hőmérséklet(x) < 15\ or\ hőmérséklet(x) > 35 \end{cases} \quad (1)$$

A tagsági függvény 1 értéket vesz fel 25 fok esetében, ez jelenti a meleg hőmérsékletet 1 mértékben, azaz teljes mértékben igaz rá, hogy meleg. 0 értéket vesz fel 15 fok alatt illetve 35 fok felett, ez is a meleg hőmérsékletet jelenti de 0 mértékben, azaz egyáltalán nem igaz rá, hogy meleg (15 fok alatt nagyon hideg, 35 fok felett pedig forró). 15 és 25 fok között a függvény alakja lineárisan nő, 25 és 35 fok között pedig lineárisan csökken, amely által a függvény háromszög alakot vesz fel. A következő 1. ábra ezt a háromszög alakú *'meleg'* tagsági függvényt (fuzzy halmazt) szemlélteti:



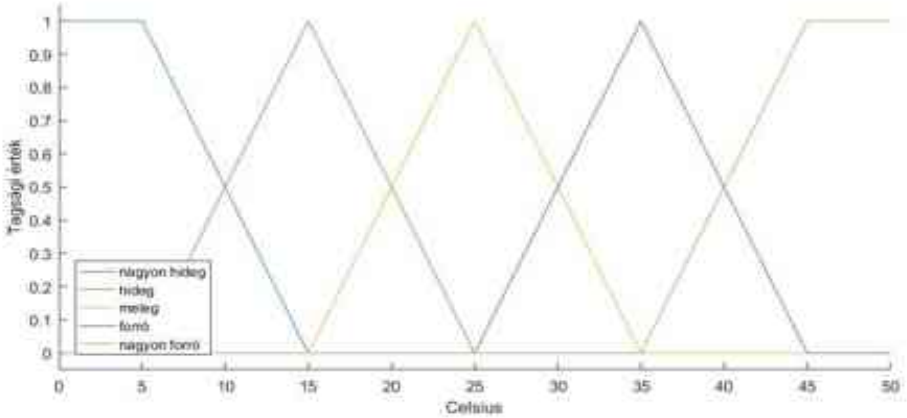
1. ábra: A 'meleg' fuzzy halmaz tagsági függvénye

A tagsági függvények alakját az adott alkalmazási terület határozza meg [24], de általában Gauss, háromszög, trapéz, harang, vagy sigmoid alakúak. A háromszög alakú fuzzy halmazok esetében általában a halmaz magjával (magnak nevezik azt, ahol a tagsági érték 1) és a függvény meredekségével jellemezhető a tagsági függvény alakja.

A fuzzy halmazokhoz nyelvi változók (*lingvisztikai változók*) rendelhetők, amelyek az alaphalmaz elmeit valamilyen szempontból felosztják és a numerikus értékekhez (*értelmezési tartományokhoz*) beszédes neveket (*nyelvi értékeket*) rendelnek. Az előző példában a nyelvi változó (x) a *'hőmérséklet'*, amely a *'nagyon hideg'*, *'hideg'*, *'meleg'*, *'forró'* és *'nagyon forró'* nyelvi értékeket $\big(T(x)\big)$ veheti fel, melyekhez fuzzy számok illetve azok univerzuma (U) rendelhető. A nyelvi értékek tehát fuzzy halmazokhoz rendelt beszédes nevek, a nyelvi változók nyelvi értékeket kaphatnak értékül. Egy lehetséges felosztást a következő példa szemléltet:

```
T(hőmérséklet) = { nagyon hideg, hideg, meleg, forró, nagyon forró }
U = [0,50] (Celsius)
nagyon hideg = { 0 0 5 15 }
hideg = { 5 15 25 }
meleg = { 15 25 35 }
forró = { 25 35 45 }
nagyon forró = { 35 45 50 50 }
```

A következő ábrán a fentieknek megfelelően definiált fuzzy halmazok láthatóak, amelyekből a *'nagyon hideg'* és a *'nagyon forró'* trapéz, a *'hideg'*, *'meleg'* és a *'forró'* nyelvi értékű halmazok pedig háromszög alakú tagsági függvényekkel rendelkeznek:



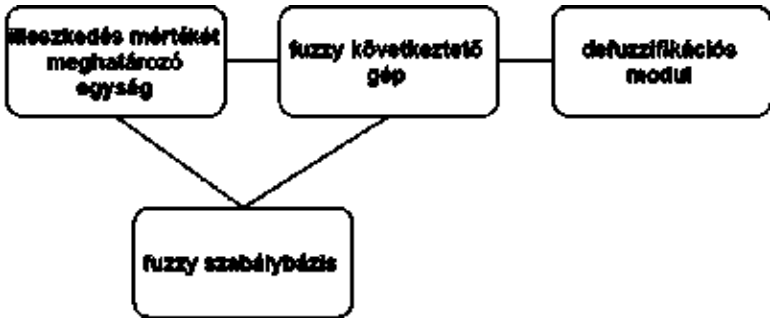
2. ábra: A 'hőmérséklet' fuzzy halmazait leíró tagsági függvények

A tagsági függvények által definiált fuzzy halmazokon értelmezhetők különböző halmazelméleti műveltek, mint például a fuzzy komplemens képzés, a fuzzy metszet (*t-norma*), a fuzzy unió (*t-conorma, s-norma*) és így tovább. Ezek a műveletek különféleképpen

értelmezhetők a fuzzy halmazokra így különböző t-normák (és *s-normák*) léteznek [26]. A legelterjedtebb ezek közül a ZADEH-féle halmazműveletek, melyet [102]-ben is javasolt, de ilyen még a YAGER [76] és a DOMBI [37] norma is. A legelterjedtebb t-norma a ZADEH-féle minimum, a legelterjedtebb s-norma pedig a ZADEH-féle maximum függvény [102]. A további elterjedtebb, gyakorlatban is használt t-normákról a [67] jelölésű irodalom ad bővebb áttekintést.

2.1 FUZZY IRÁNYÍTÁSI RENDSZEREK

Az alábbi 3. ábra a fuzzy irányítási rendszerek általános felépítését szemlélteti [53]:



3. ábra: Fuzzy irányítási rendszer felépítése [53]

A fuzzy irányítási rendszerek alapja egy olyan modell amelyet, a fuzzy szabályokat magába foglaló fuzzy szabálybázis alkot, majd ezen szabálybázis szabályai alapján következtet. Többféle következtetési eljárás létezik, a legelső a Zadeh által 1973-ban javasolt kompozíciós fuzzy következtetés (*Compositional Rule of Inference - CRI*), melyet Mamdani 1975-ben átdolgozott, így ez a Mamdani nevet viseli. A rendszer tudásábrázolása a szabálybázis által van biztosítva. A szabálybázisban lévő szabályok mindegyike „ha a bemenet *A*, akkor a kimenet *B*", tehát „HA-AKKOR" típusú szabályokból áll, ahol *A* és *B* fuzzy halmazok [53]:

$$R_i\colon\ HA\ x = A\ \ AKKOR\ \ y = B \tag{2}$$

Ahol $x \in X$ bemeneti változó, $y \in Y$ kimeneti változó, X, Y a be- és kimeneti változók alaphalmaza, A, B nyelvi változók, A az előzménye (*antecedense*), B pedig a következménye (*konzekvense*) az i-edik R szabálynak. Az R szabálybázist általában többdimenziós $x_n = A_{n,i}$ antecedenssel (és akár többdimenziós $y_m = B_{m,i}$ konzekvenssel) rendelkező szabályok alkotják.

Az illeszkedés mértékét meghatározó egység a szabályok antecedenseit a megfigyelés értékével hasonlítja össze és tüzelő szabályok esetében meghatároz egy $0-1$ közötti illeszkedési mértéket.

A fuzzy következtető gép az illeszkedés mérték meghatározása következtében kapott súlyokat a tüzelő szabályok konzekvenseivel kombinálja. A kombinálási módszerek eltérőek lehetnek az egyes irányító típusok esetében.

A defuzzifikációs modul a fuzzy kimeneti értékből egy diszkrét (*crisp)* értéket állít elő, célja a fuzzy halmazra leginkább jellemző, nem fuzzy érték meghatározása. Ez a defuzzifikálás folyamata. Különböző defuzzifikációs módszerek ismertek, ilyenek például a maximumok közepe módszer (*Mean Of Maxima - MOM*), a súlypont keresésének módszere (*Center of Gravity - COG*), a geometriai középpont módszer (*Center of Area - COA*) és a maximális tagsági értékű elem keresésnek módszere [53][64].

Különböző típusú irányítási rendszerek találhatóak meg a szakirodalomban, attól függően, hogy a rendszer hogyan állítja elő a teljes szabályrendszerre nézve a következményt, azaz milyen módon következtet. Az elterjedtebbek a Mamdani-, a Zadeh- [61][64][103], illetve a Takagi-Sugeno-féle [83][86] rendszerek.

A Mamdani-féle kompozíciós következtetésben (*CRI*) a B^* következmény az A^* megfigyelés és az R szabálybázis reláció (max, min normákon alapuló) kompozíciójaként áll elő [32][53]:

$$B^* = A^* \circ R \tag{3}$$

Ebben az esetben a megfigyelés minél jobban illeszkedik valamelyik szabály antecedensére, azon szabály konzekvense annál nagyobb ω súllyal vesz részt a végső következmény meghatározásában. Azonban ha bármelyik dimenzióban a megfigyelés és az antecedens metszete üres, akkor a szabályhoz tartozó következtetés fuzzy halmaza üres lesz.

Legyen $A^* \in X_1 \times X_2 \times \ldots \times X_n$ az n-dimenziós megfigyelés, ekkor a $\omega_{j,i}$ illeszkedés mértéke ami az A_j^* megfigyelés és az $A_{j,i}$ antecedens kapcsolatát írja le a j-edik ($j \in [1, n]$) dimenzióban a következő módon határozható meg [53]:

$$\omega_{j,i} = \max_{x_j}\left\{\min\left\{A_j^*(x_j), A_{j,i}(x_j)\right\}\right\} \tag{4}$$

A ω_i súlyfaktor ami meghatározza, hogy az R_i szabály konzekvense, milyen mértékben vesz részt a végső konklúzióban, az összes antecedenshez tartozó súlyfaktorok minimumaként áll elő [53]:

$$\omega_i = \min_{j=1\ldots n} \omega_{j,i} \tag{5}$$

Az adott megfigyeléshez és szabályhoz tartozó B_i^* következtetést a B_i konzekvens ω_i „magasságában” történő csonkolása után kapjuk meg [53]:

$$B_i^*(y) = \min\big(\omega_i, B_i(y)\big) \tag{6}$$

A teljes szabálybázisra vonatkozó következtetés pedig az egyes szabályokhoz tartozó B_i^* konklúziók uniójaként áll elő [53]:

$$B^* = \bigcup_{i=1}^{r} B_i^* \text{ azaz } B^*(y) = \max_{i=1\dots r} B_i^*(y) \tag{7}$$

A Mamdani típusú következtetés esetében a végső konklúzió egy fuzzy halmaz, így defuzzifikációs lépésre van szükség a crisp érték előállításához.

A Sugeno-féle következtetés hasonló a Mamdani-féléhez, azzal az eltéréssel, hogy ebben az esetben a kimenet nem fuzzy halmaz lesz (így nincs szükség defuzzifikációs lépésre) illetve a szabályok konzekvensében nem fuzzy halmazok szerepelnek, hanem matematikai függvények. Az R szabálybázis r ($r \in R$) szabályainak általános formátuma az alábbi [53]:

$$r_i\colon HA\ x_1 = A_{1,i}, \dots, x_n = A_{n,i}\ AKKOR\ y_i = f_i(x_1, \dots, x_n) \tag{8}$$

Ahol f_i n-dimenziós függvény, x_i ($i \in [1, n]$) pedig a bementi változók. Ha az f függvény konstans, akkor nulladrendű *Sugeno*, ha az a bementek lineáris függvénye, akkor elsőrendű *Takagi-Sugeno* [86], ha magasabb rendű függvény, akkor pedig általános *Takagi-Sugeno-Kang* típusú irányító. A konklúzió meghatározása súlyozott átlagolással áll elő, ahol a súly a megfigyelés illeszkedésének a mértéke [53]:

$$y = \frac{\sum_{i=1}^{r} \omega_i * y_i}{\sum_{i=1}^{r} \omega_i} = \frac{\sum_{i=1}^{r} \omega_i * f_i(x_1, \dots, x_n)}{\sum_{i=1}^{r} \omega_i} \tag{9}$$

A bemutatott, klasszikus Mamdani- és Sugeno-féle fuzzy irányítási rendszerek fedő (teljes) szabálybázist igényelnek működésükhöz, azaz olyan szabályrendszert, amelyben minden egyes lehetséges megfigyelésre létezik legalább egy olyan szabály, amely előzménye minden egyes bemeneti dimenzióban metszi (vagy fedi) a megfigyelést. Ezen rendszerek esetében azonban az antecedens dimenziók számának növekedése a szabályszám exponenciálisan növekedését eredményezi [49], amely a szabályrendszer komplexitásának növekedéséhez vezet [60], aminek következtében megnövekedhet a rendszer következtetési ideje [53].

A fuzzy rendszerek előnye, hogy alkalmasak bizonytalan, pontatlan formában megadott (pl. kicsi, nagy, közel, távol, hideg, meleg stb.) fogalmak kezelésére. Például egy légkondicionáló berendezés esetében megvalósítható annak hőmérsékletfüggő és energiahatékony szabályozása

[22][23], ipari alkalmazás esetében egy tankhajtású targonca nyomvonal követése [59], de alkalmas akár egy autó biztonságos haladási sebességének meghatározására az éppen aktuális időjárási és útviszonyok illetve az autó karbantartottsági állapota alapján [1][71].

A fuzzy következtető rendszerek alkalmasak lehetnek továbbá szakértőtől származó információk rendszerbe történő beépítésére is. A szakértői tudásbázis lehetőséget ad az adott irányított folyamatra vonatkozó előzetes ismeret beépítésére, amely olyan problémák esetében bírhat nagy jelentőséggel, ahol a folyamat matematikai modellje bonyolult vagy csak részben ismert [53]. A fuzzy irányító rendszerek további felépítéséről, működéséről a [46][51] és [53] jelölésű szakirodalmakban található részletesebb áttekintést.

2.2 FUZZY SZABÁLY-INTERPOLÁCIÓ

A klasszikus fuzzy irányítási rendszerek (amelyek a hagyományos fuzzy következtetést alkalmazzák) [61][64][86][103] működésének feltétele a szabálybázis fedő jellege, azaz, hogy bármilyen megfigyelés esetén létezzen legalább egy olyan szabály, amelynek antecedense nagyobb, mint nulla mértékben ($\varepsilon > 0$) fedi a megfigyelést minden egyes bemeneti dimenzióban. Tehát elengedhetetlenül fontos, hogy minden létező megfigyelés esetében a rendszer állítson elő következtetést a kimeneten. Abban az esetben azonban, ha ez a fedő jelleg nem teljesül (ritka szabálybázis), nem biztosított a bemeneti tér teljes lefedettsége, akkor előfordulhat olyan eset, hogy a rendszer nem ad következtetést [53][56][58]. Ez a beágyazott rendszerekben, gyakorlati alkalmazásokban nagy gondot okozhat, hiszen nem áll elő a szükséges beavatkozó jel, így ez elkerülendő szituáció. A fuzzy szabály-interpolációs módszerek (*Fuzzy Rule Interpolation - FRI*) célja, hogy ritka szabálybázisok alkalmazásának esetében is valamilyen módon határozzon meg a rendszer következményt a kimenetén.

A fuzzy szabály-interpolációs módszereket két csoportba sorolhatjuk attól függően, hogy közvetlenül állítják-e elő a konklúziót vagy sem. Ez alapján megkülönböztetünk egylépéses és kétlépéses interpolációs módszereket. Az egylépéses interpolációs eljárások a megfigyelés illetve két vagy több közrefogó szabály (az első szabály antecedens halmaza megelőzi a megfigyelés halmazt és a második szabály antecedens halmaza követi a megfigyelés halmazt, minden egyes antecedens dimenzióban) figyelembevételével állítják elő a konklúziót. Számos egylépéses interpolációs módszer megtalálható a szakirodalomban. Az ismertebbek az α-vágat alapú, távolságokon és azok aránya alapján működő Kóczy-Hirota-féle lineáris szabály-interpoláció (*KH módszer* [50]), a Tikk-Baranyi féle módosított α-vágat alapú MACI [89], a bizonytalan környezetet alkalmazó Kovács-Kóczy-féle „FIVE” [54][55][57], a Wong-Gedeon-

Tikk által javasolt javított többdimenziós módosított α-vágat alapú IMUL [100], az új távolságmértéket bevezető majd az α-vágatok szélessége alapján működő Vass-Kalmár-Kóczy-féle VKK [93] és a Kóczy-Hirota-Gedeon-féle CRF [52]. A kétlépéses módszerek első lépésben egy segédszabályt interpolálnak, majd annak felhasználásával második lépésben állítják elő a következményt. Ezen módszerek a Baranyi-Kóczy-Gedeon-féle általánosított fuzzy szabály-interpolációs módszertant (*Generalized Methodology of Fuzzy Rule Interpolation - GM*) [4] követik, melyek működése egy referencia pont alkalmazásával való távolságmérésen és rendezésen alapszik. Ezen módszerek családjába sorolhatók a Baranyi és társai által kidolgozott eljárások [4][5][6], a hasonlóság megőrzési módszeren alapuló Yan-Mizumoto-Qiao-féle ST [101] valamint a Johanyák által kidolgozott VEIN [41], LESFRI [40] és FRIPOC [39] módszerek.

Az FRI módszerek erőssége abban rejlik tehát, hogy képesek ritka szabálybázis közvetlen alkalmazására. A szabálybázisnak a klasszikus fuzzy következtetési módszerekkel ellentétben elég csak a lényegi szabályokat tartalmaznia, így egyes esetekben a rendszer leírásának komplexitása is csökkenthető [58]. További fuzzy szabály-interpolációs módszereket illetve azok különböző szempontok által történő vizsgálatát a [38], [42], [69] és [70] sorszámú publikációk foglalják össze.

2.2.1 A „FIVE" Fuzzy szabály-interpolációs módszer

A „FIVE" (*Fuzzy Interpolation in the Vague Environment, bizonytalan környezet alapú fuzzy szabály-interpoláció*) Kovács-Kóczy [54][55][57] által kifejlesztett egylépéses szabály-interpolációs módszer, amely az interpolációs feladatot egy úgynevezett bizonytalan környezetbe (*Vague Environment, VE*) [47] helyezi át. A Klawonn-féle bizonytalan környezet alapgondolata az univerzum elemei közötti hasonlóságon illetve megkülönböztethetetlenségen alapszik [47]. Az elemek hasonlóságának mértéke súlyozott távolság által definiálható ($hasonlóság\ mértéke = 1/(1 + távolság)$), ahol a súlytényező az úgynevezett $s(x)$ skálafüggvény [47][55][56]:

$$s(x) = |\mu'(x)| = \left|\frac{d\mu}{dx}\right| \qquad (10)$$

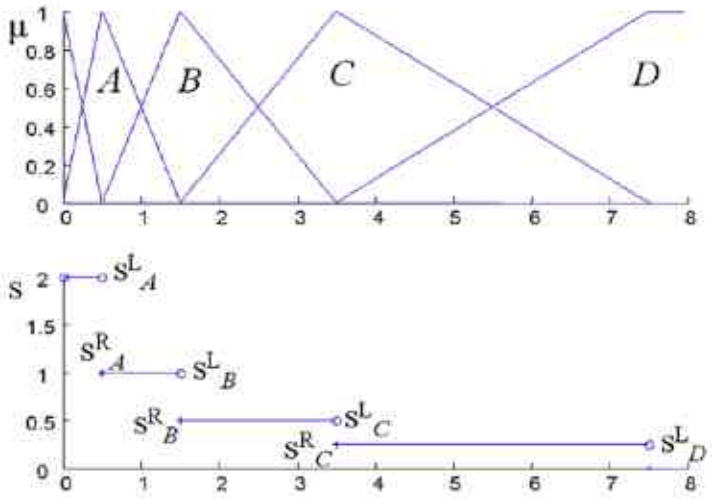
létezik ha: $\min\{\mu_i(x), \mu_j(x)\} > 0 \rightarrow |\mu'_i(x)| = |\mu'_j(x)|$

Ahol $\mu'(x)$ a fuzzy halmaz tagsági függvényének deriváltja, ebben az esetben tehát az $s(x)$ skálafüggvény a $\mu(x)$ tagsági függvény deriváltja [55]. Az (x_1, x_2) elemek közötti, bizonytalan környezetbeli δ_s távolságuk meghatározásának módja az $s(x)$ skálafüggvényen alapszik [47]:

$$\delta_s(x_1, x_2) = \left| \int_{x_2}^{x_1} s(x)dx \right| \tag{11}$$

Az X bizonytalan környezetben az (x_1, x_2) elemek ε mértékben hasonlónak (megkülönböztethetetlennek) tekinthetőek, ha a közöttük lévő δ_s távolság legfeljebb ε mértékű, azaz $\delta_s(x_1, x_2) \leq \varepsilon$ [55][56]. A bizonytalan környezetek (antecedens, konzekvens, szabálybázis) előre számolhatók (így biztosítva a módszer gyorsaságát) amelyben minden szabály egy-egy szabálypontként ábrázolható.

Az alábbi 4. ábra fuzzy halmazokat (az ábra felső részében lévő grafikon) és az őket jellemező skálafüggvényeket (az ábra alsó részében lévő grafikon) szemlélteti, amelyek ezáltal alkalmasak az adott fuzzy partíció alakjának leírására [56]:



4. ábra: Fuzzy halmazok (felső grafikon) és az őket jellemző skálafüggvények (alsó grafikon) [56]

Minél kisebb a skálafüggvény értéke az elemek annál kevésbé megkülönböztethetők egymástól, azaz ugyanolyan alaphalmazbeli távolság esetében egyre közelebb vannak egymáshoz. Ha a skálafüggvény értéke nulla az azt eredményezi, hogy az elemek nem megkülönböztethetőek, mert egyformán közel helyezkednek el. Egy valós gyakorlati alkalmazás esetében ez azt eredményezi, hogy például minden olyan esetben amikor 2 méternél messzebb találhatok objektumok akkor nem kell fékezni, amikor pedig ennél közelebb vannak akkor egyre jobban kell fékezni.

A „FIVE” módszer a multidimenziós mivolta következtében a Shepard interpolációs operátort [25] alkalmazza, amely által a singleton (egyértékű) c_k konzekvens a következő összefüggés alapján, további defuzzifikációs lépések nélkül határozható meg [56]:

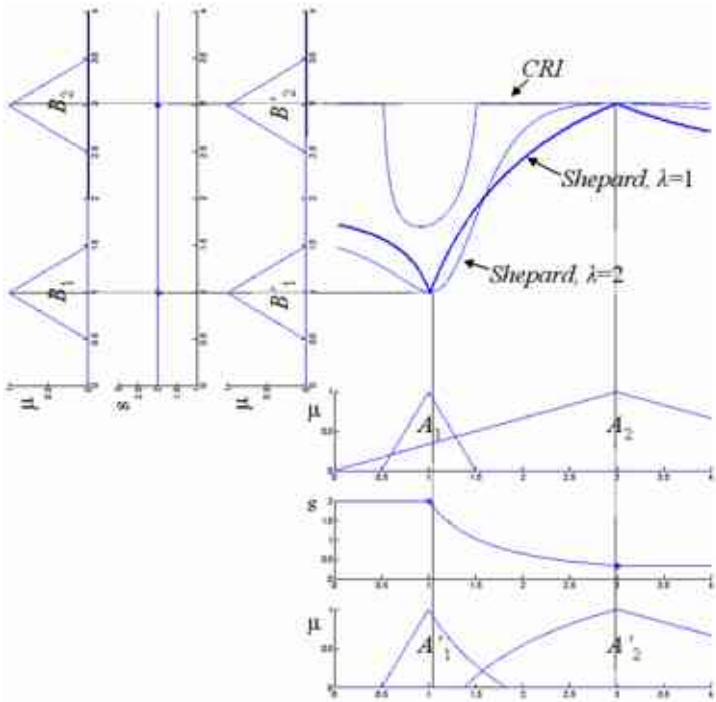
$$y(\boldsymbol{x}) = \begin{cases} c_k & \text{ha } x = a_k \text{minden } k\text{-ra,} \\ \sum_{k=1}^{r} \left(\left({}^{c_k}/_{\delta_{s,k}^{\lambda}} \right) / \left(\sum_{k=1}^{r} {}^{1}/_{\delta_{s,k}^{\lambda}} \right) \right) & \text{egyébként} \end{cases} \tag{12}$$

Ahol $\boldsymbol{x}$ a többdimenziós megfigyelés, c_k a k-adik létező szabály konzekvense, r a szabályok száma az R szabálybázisban, λ a Shepard-kivető, $\delta_{s,k}$ pedig a súlyozott (Euklideszi) távolság, amely a következő formával által definiálható [55][56]:

$$\delta_{s,k} = \delta_s(\boldsymbol{a_k}, \boldsymbol{x}) = \left[\sum_{i=1}^{m} \left(\int_{a_{k,i}}^{x_i} s_{x_i}(x_i) dx_i \right)^2 \right]^{1/2} \tag{13}$$

Ahol $\boldsymbol{x}$ az m-dimenziós crisp megfigyelés, $\boldsymbol{a_k}$ a magja az m-dimenziós szabály antecedens $\boldsymbol{A_k}$-nak, s_{x_i} pedig az i-edik skálafüggvény az m-dimenziós antecedens univerzumban.

A következő 5. ábra [58] a „FIVE” módszer működését szemlélteti, egy egydimenziós antecedens-konzekvens tér esetében (*singleton* megfigyelés és *singleton* konklúzió), 2 szabállyal ($R_i: A_i \rightarrow B_i$), az interpoláció adta eredményt a klasszikus min-max CRI módszerhez hasonlítva:



5. ábra: A „FIVE” működése 2 szabály ($R_i: A_i \rightarrow B_i$) estében és eredményének hasonlítása a min-max CRI módszerhez képest (COG defuzzifikáció alkalmazásával) [58]

Az ábra bal felső és jobb alsó részében, az első grafikonon a fuzzy partíciók (B_1, B_2, A_1, A_2), azok alatt az azokból számolt S skálafüggvény, majd a 3. grafikonon pedig a skálafüggvényből visszaszámolt (közelített) fuzzy partíciók (B_1', B_2', A_1', A_2') helyezkednek el. Az ábra jobb felső részében az interpolációs grafikonok helyezkednek el. A vastag vonallal jelzett görbe a FIVE módszer által adott eredmény $\lambda = 1$ Shepard kitevő, illetve a vékonyabb vonallal jelezett görbén pedig a $\lambda = 2$ Shepard kitevő esetében kapott interpolációs görbe. A „CRI”-vel jelzett görbe a klasszikus min-max CRI-módszer által (*COG defuzzifikáció alkalmazásával*) kapott következtetés eredménye.

A „FIVE” tehát egy alkalmazás orientált, többdimenziós térben is működő FRI módszer, amely viszonylag kis számításigénye következtében jól alkalmazható valós idejű, beágyazott rendszerekben [7][9] illetve robotikához kapcsolódó gyakorlati alkalmazásokban is [8]. A „FIVE FRI Matlab Toolbox” a [28] hivatkozáson érhető el, tölthető le.

3 MEGERŐSÍTÉSES TANULÁS

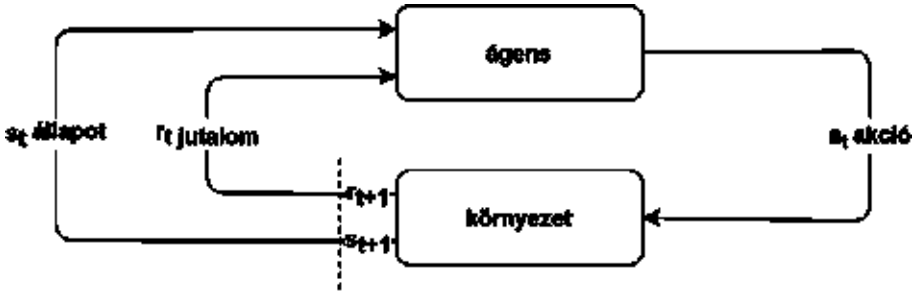
A gépi tanulás [17][68] témaköre alapvetően három, nem teljesen élesen elkülöníthető csoportba sorolható, melyek a felügyelt tanulás, a felügyelet nélküli tanulás illetve a megerősítéses tanulás.

A felügyelt tanulás (*vagy ellenőrzött tanulás, supervised learning*) esetében rendelkezésre állnak tanítóminták (összetartozó be-kimeneti értékek), majd a rendszer feladata a tanítóminták által adott ki- és bementi leképzés megtanulása. A leképzés révén ismertek az adott bemenethez tartozó kimeneti, elvárt válaszok, így a rendszer bementre adott kimeneti válaszát össze lehet hasonlítani az elvárt kimeneti válasszal, majd ez alapján módosítani a viselkedést.

A felügyelet nélküli tanulás (*nemellenőrzött tanulás, unsupervised learning*) esetében az elvárt kimeneti értékek nem állnak rendelkezésre az adott bemenetekhez, a rendszer célja a bemeneti minták alapján az adott viselkedés kialakítása. Ebben az esetben nincs visszajelzés a rendszer számára, hogy mely kimeneti válasz lett volna a helyes.

A megerősítéses tanulás (*reinforcement learning - RL*) [84] esetében a rendszer megerősítési információk alapján alakítja ki a viselkedést, minden lépésben kap visszajelzést, úgynevezett megerősítést (jutalmat vagy büntetést) az adott döntés vagy döntések végrehajtását követően, de ezekből arra nem lehet következtetni, hogy ezt mely döntéssorozatának köszönhetően kapta (nincs külső tanár, aki minden esetben adna visszajelzést arról, hogy mi volt a helyes cselekvés). A megerősítési információk egy jutalomfüggvényből származnak, amely definiálja a rendszer számára, hogy mely esetekben jutalmazhat vagy büntethet. Ezen módszerek próbálkozás típusú (*trial-end-error*) algoritmusok összesége, melyek a megoldásra vonatkozó ismeret nélkül, a környezettől kapott megerősítési információk alapján térképezik fel a rendszer elvárt viselkedését. A jutalomfüggvény által van definiálva az elérendő cél, hogy mely esetben jár nagy jutalom az adott cselekvésért (vagy cselekvéssorozatért), mely döntés (döntéssorozat) milyen mértékben volt helyes, de a megoldás mikéntjére vonatkozóan nincs információ. A tanulási módszer alapötlete tehát, hogy a visszajelzéseket ne csak az ágens (azaz a tanuló entitás) jelenlegi cselekvéseinek kialakítására használják fel, hanem arra is, hogy javítsa a jövőbeli döntésekre irányuló képeséget, tehát a tanulás során lépésről-lépésre egyre helyesebben oldja meg az adott feladatot. A tanulási folyamat epizodikus, azaz véges hosszúságú időszeletekre, úgynevezett epizódokra bontott. Minden egyes epizód egy kezdeti állapot (*initial state*) és egy végállapot (*terminal state*) között játszódik, egy epizódon belül bármennyi véges számú lépés

(*step*) lehetséges és az egyes epizódok egymástól függetlenek. A tanulási folyamat konvergencia sebességét a tanulási fázis során lejátszódó epizódok száma határozza meg, azaz hogy mennyi epizód alatt találta meg a rendszer a megoldást, mennyi epizódot igényelt a tanulási folyamat. A megerősítéses tanulás ágens-környezet modelljét a következő 6. ábra szemlélteti [84]:



6. ábra: A megerősítéses tanulás ágens-környezet modellje [84]

A tanuló entitás az ágens (*agent*). Az ágens minden egyes diszkrét t időpillanatban kapcsolatba kerül a környezettel és végrehajt egy adott a_t akciót. Az a_t akciót a lehetséges A akciók halmazából választja, ezek az általa végrehajtható cselekvések. A lehetséges akciók választásának módját a π politika (stratégia) írja le, egyfajta állapot-akció leképzést valósít meg, meghatározva ezáltal az ágens viselkedését. Ez a politika akkor optimális, ha a várható összjutalmat maximalizálja. A környezet az a_t akció végrehajtását követően válaszol egy r_{t+1} numerikus megerősítési értékkel (amely lehet jutalom vagy büntetés) és egy új s_{t+1} állapottal, ez a folyamat a $(s_{t-1}, a_{t-1}, r_t, s_t, a_t, r_{t+1}, s_{t+1}, \dots)$ szekvenciával realizálható.

Az S állapothalmaz az ágens által felvehető állapotváltozók lehetséges értékeit, az r jutalom értéke pedig az ágens által végrehajtott akció jóságát határozza meg. A jutalom értékét egy jutalomfüggvény határozza meg, amely definiálja, hogy az egyes állapotátmenetekhez mekkora mértékű megerősítés társítható. Ha ez a megerősítés pozitív akkor jutalmat, ha negatív akkor pedig büntetést (negatív jutalmat) szimbolizál. Az ágens célja, hogy a gyűjtött jutalmak értékét hosszútávon maximalizálja, azaz az $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$ diszkontált kumulált jutalom értéke legyen hosszútávon maximális (ne az azonnali r jutalomé). A γ $(0 < \gamma \leq 1)$ diszkontálási tényező (*discount factor*) a súlyt határozza meg, hogy az idő teltével mekkora mértékben vegye figyelembe a rendszer a későbbi (jövőbeli) r jutalom értékeket.

Mind az állapotok, mind az akciók, mind pedig a megerősítések halmaza véges számú diszkrét értékek halmaza. Az ágens az adott s állapotban, az adott a akció végrehajtását követően mindig egy új s' állapotba kerül, melyet a környezet határoz meg. A cél a jutalom értékének hosszútávú maximalizálásra való törekvés, azaz optimális (vagy közel optimális) stratégia keresése az adott környezethez.

Az alkalmazott modell Markov döntési folyamat (*Markov Decision Process - MDP*) [10], azaz egy diszkrét idejű sztochasztikus folyamat, ahol az egyes állapotokhoz tartozó állapot-átmeneti függvény nem függ a megelőző állapotoktól, azaz a rendszer nem képes az „emlékezésre". A folyamat jövőbeli állapota csakis a jelenlegitől függ, a múlt és a jövő függetlenek egymástól.

A modell elemei a következőképpen definiálhatók [84]:

- t: diszkrét időpillanat ($t = 0,1,2,3, \ldots$)
- S: lehetséges állapotok halmaza
- s_t: egy lehetséges állapot, $s_t \in S$
- A: lehetséges akciók halmaza
- a_t: egy lehetséges akció, $a_t \in A$
- $P(s', s, a) = Pr(s_{t+1} = s' | s_t = s, a_t = a)$: annak a valószínűsége, hogy az a_t akció végrehajtása az s_t állapotból az s_{t+1} állapotba vezet a t időpillanatban
- $R(s', s, a) = Pr(r_{t+1} | s_{t+1} = s' | s_t = s, a_t = a)$: jutalomfüggvény, amely az r_{t+1} megerősítés, azaz jutalom vagy büntetés (negatív jutalom) értékét határozza meg az s-ből az s' állapotátmentre az a akció végrehajtását követően
- $\pi(s, a) = Pr(s_t = s, a_t = a)$: politika, amely az a akció választását határozza meg az A akcióhalmazból az s állapotban, állapot-akció leképezés
- $V^{\pi}(s)$: állapot-érték függvény, a kumulált megerősítések összesége s állapotból kiindulva a π politika követése mellett
- $Q^{\pi}(s, a) = E(R(s', s, a) + \gamma V^{\pi}(s'))$: állapot-akció-érték függvény (Q-függvény), ami várható jutalmak összeségét határozza meg az s állapotban az a akció végrehajtását követően a π politika követése mellett. A Q értéke az s állapotban végrehajtott a akció jóságát (Q-értékét) határozza meg. A γ értéke a jelen s állapot jövőre vetíthetőségének mértéke, azaz, hogy mekkora súllyal legyen figyelembe véve egy jövőbeli állapot hasznossága

Az állapotok hasznosságának meghatározására a Bellman-egyenlet [11] szolgál, amely alapján egy állapot hasznossága megegyezik az adott állapotban való tartózkodás r_0 megerősítésének és a következő $s' = s_{t+1}$ állapot diszkontált $\gamma V(s')$ várható hasznosságának az összegével (a π politika követése mellett) [11]:

$$\mathrm{V(s)} = \max_{\pi} \mathrm{E(r_0 + \gamma V(s'))} \tag{14}$$

A megerősítéses tanulási algoritmus mindegyikének célja egy $\pi^* = arg\max_{\pi} \mathrm{E}(\mathrm{r}_0 + \gamma \mathrm{V}(\mathrm{s}'))$ optimális politika keresése az adott környezethez.

A megerősítéses tanulási módszerek a dinamikus programozás (*Dynamic Programming - DP*) témakörébe sorolhatók [11][12], amely szekvenciális döntési problémákra ad numerikus megoldásokat illetve, hogy egy optimalizációs probléma hogyan osztható fel több kisebb rekurzív optimalizációs részproblémára [20]. Abban az esetben ha ismert a környezet modellje, azaz az állapotátmenteket és a jutalmakat előrejelző függvények (*model-based RL*), akkor az úgynevezett *„actor-critic"* módszerek alkalmazhatók az optimális állapot-érték függvény (érték-iteráció módszer) és az optimális politika (politika-iteráció módszer) keresésére. A *critic* komponens az értékeket, az *actor* komponens pedig a politikát tanulja.

A politika-iteráció (*policy iteration*) egy többlépéses optimalizációs eljárás, amely az optimális V^{π} állapot-érték függvényt keresi a π politika követése mellett. Első lépésben a követett π politikát rögzíti, majd ennek ismeretében értékeli ki az állapot-érték függvényt. További lépésben az állapot-érték függvényt rögzíti, majd a politikát optimalizálja olyan módon, hogy az adott állapotban különböző akciókat hajt végre a várható jutalom maximalizálása céljából. Tehát a mohó politikát keresi úgy, hogy egy véletlen politikával indul, majd ezt a politikát módosítja iterációról-iterációra. Általában néhány iteráció szükséges a konvergálásához, ami miatt viszonylag gyors.

Az érték-iterációs módszer (*value iteration*) hasonló a politika-iterációhoz, azzal a különbséggel, hogy minden egyes iterációban csak az állapot-érték függvényt frissíti, majd az optimális politika ez alapján áll elő a folyamat végén [20]. Véletlenszerű állapot-érték függvénnyel indul, majd ezt a függvényt frissíti minden egyes iterációban. Számításigénye magasabb a politika-iterációs módszerhez képest és általában jóval több lépés alatt konvergál, amely következtében lassabb is. Az „actor-critic" módszerekből az „actor" politika-alapú, ezért a politikát tanulja, a „critic" pedig érték-alapú, így az állapot-érték függvényt tanulja.

Azon algoritmusok egy csoportját, amelyek esetében nincs szükség a környezet modelljének ismeretére (*model-free RL*), időbeli-különbség tanulásnak (*Temporal Difference learning - TD-learning*) nevezik [31][77][81]. Ebben az esetben a hasznosságnak (Q-érték) egy becsült értéke kerül figyelembevételre, így az ágensnek nem kell elérnie a végső terminális állapotot és nem kell várnia a teljes összjutalom létrejöttére ahhoz, hogy a becsült hasznosság értékeket frissítse. Az összjutalom értékének későbbi ismeretében a Q-érték visszamenőleg kerül frissítésre (egyes módszerek esetében neurális háló alkalmazásával [88]). Ilyen „modell-free" algoritmusok az

eredetileg diszkrét állapot- és akciótér felbontással rendelkező Q-learning [99] és SARSA (*State-Action-Reward-State-Action*) [79] módszerek.

3.1 Q-LEARNING ÉS SARSA

A Q-learning (*Q-tanulás*) algoritmus [99], amely eredeti megfogalmazásban diszkrét állapot- és akció tér felbontással rendelkezik - azaz véges számú és diszkrét értékű állapot-akció értékek lehetségesek - a (14) Bellman egyenlet fixpont megoldásait keresi iterációkon keresztül. Az állapot-akció-érték függvény (*Q-függvény - Quality function*) frissítési szabálya a következő [99]:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha * (r_{t+1} + \gamma * \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \qquad (15)$$

Ahol t az adott időpillanat, r_{t+1} a kapott jutalom az $s_t \rightarrow s_{t+1}$ állapotátmenetre, $Q^{new}(s_t, a_t)$ az új frissített Q-érték, $Q(s_t, a_t)$ a régi Q-érték (s_t, a_t)-ban, α tanulási ráta (*learning rate*, $0 < \alpha \leq 1$), γ a diszkontálási tényező (*discount factor*, $0 < \gamma \leq 1$), $\max_a Q(s_{t+1}, a)$ pedig az a becsült érték, ami az s_{t+1} állapotba vezető feltehetően legjobb a akció végrehajtása mellett érhető el. Az α értéke azt határozza meg, hogy az új frissített értékek milyen mértékben kerüljenek figyelembe vételre a régi értékekhez képest (ha ez az érték 0 akkor a Q-értékek nem kerülnek frissítésre, aminek következtében nem tanul a rendszer).

A $Q(s_t, a_t)$ függvény tehát az adott állapotokban az adott akciók végrehajtása melletti jóság értékeket (Q-érték) írja le. Ezen Q-függvény értékei általában egy Q-táblában tárolódnak, amely az összes állapot-akció párra vonatkozó Q-értéket tárolja, majd a tanulási fázis közben a (15) összefüggés alapján frissíti azokat. Minél finomabb az állapot-akció tér felbontása annál nagyobb méretű a Q-tábla, mérete a dimenziók számának és felbontásának növekedésével rohamosan nő.

A Q-learning politikafüggetlen (*off-policy*) algoritmus, amely a (15) formula alapján a Q-értékeket a legjobb akció (mohó akció) alapján frissíti. A mohó akció azt az akciót jelöli, amely végrehajtása mellett az adott állapotban a legnagyobb hasznosságérték (vagy Q-érték) várható. A Q-learning algoritmus tehát mohó akcióválasztási politika alapján frissíti Q-értékeit függetlenül attól, hogy a mohó akció volt-e ténylegesen végrehajtva, azaz a mohó politika volt-e alkalmazva [84]:

$$\pi(s) = arg \max_a Q^{\pi}(s, a) \qquad (16)$$

A felderítés-kiaknázás (*exploration-exploitation*) technika következtében előfordul olyan eset mikor nem mindig a feltételezhető legjobb (legnagyobb hasznosságértékkel járó) akciót választja a rendszer, hanem ε valószínűséggel ($\varepsilon \in [0,1]$) választ véletlen cselekvést, majd $1-\varepsilon$ valószínűséggel pedig a mohó akciót választja. Ez az úgynevezett ε–mohó (ε-*greedy*) politika, ahol a nagyobb ε érték több véletlen akció végrehajtását eredményezi, amely következtében inkább felderít, mint kiaknáz a rendszer. A felderítést annak reményében teszi, hogy a várhatóan legnagyobb hasznosságértékkel járó cselekvésen kívül hátha talál egy olyan akciót (és ez által olyan állapotot, esetleg bejáratlan utat), amely még nagyobb hasznosságértéket eredményezhet.

Ha az adott állapotokban mindig a ténylegesen legjobb akció kerül végrehajtásra, amely mellett a ténylegesen legnagyobb hasznosságérték várható, akkor ez a π^* optimális politikát, és ennek következtében pedig az optimális Q^*-függvényt (optimális q^* értékeket) eredményezi [84].

A Q-learning algoritmusának pszeudokódja az alábbi [99]:

Algoritmus: Q-learning

Algoritmus paraméterei: $\alpha, \gamma \epsilon (0,1]$

$Q(s,a)$ inicializálása
Ismétlés (minden egyes epizódra):
    s inicializálása
    Ismétlés (az epizód minden egyes iterációjára):
        a választása s-ből adott politika alkalmazásával (például ε-mohó)
        a akció végrehajtása, r, s' megfigyelése
        $Q(s,a) \leftarrow Q(s,a) + \alpha * (r + \gamma * \max_a Q(s',a) - Q(s,a))$
        $s \leftarrow s'$
    amíg s terminális állapot nem lesz

1. algoritmus: A Q-learning módszer algoritmusa [99]

A „SARSA" (*State-Action-Reward-State-Action*) [79] algoritmus működése hasonló a Q-learning algoritmuséhoz, azaz annak egy módosított verziója, amely egy politikafüggő (*on-policy*) módszer. Ebben az esetben a Q-értékek nem a mohó politika alapján kerülnek frissítésre, hanem a ténylegesen követett politika, azaz a ténylegesen végrehajtott akciók alapján. A SARSA algoritmus frissítési szabálya a következő [79]:

$$Q(s,a) \leftarrow Q(s,a) + \alpha * (r + \gamma * Q(s',a') - Q(s,a)) \qquad (17)$$

A SARSA algoritmusának pszeudokódja az alábbi [79]:

Algoritmus: SARSA

Algoritmus paraméterei: $\alpha, \gamma \epsilon (0,1]$

$Q(s,a)$ inicializálása
Ismétlés (minden egyes epizódra):
  s inicializálása
  a választása s-ből adott politika alkalmazásával (például ε-mohó)
  Ismétlés (az epizód minden egyes iterációjára):
    a akció végrehajtása, r, s' megfigyelése
    a' választása s'-ből adott politika alkalmazásával (például ε-mohó)
    $Q(s,a) \leftarrow Q(s,a) + \alpha * (r + \gamma * \max_a Q(s',a') - Q(s,a))$
    $s \leftarrow s'; a \leftarrow a';$
  amíg s terminális állapot nem lesz

2. algoritmus: A SARSA módszer algoritmusa [79]

3.2 FUZZY Q-LEARNING

A Fuzzy Q-learning (*FQ-learning*) [2][13][18][32] módszer a diszkrét felbontású Q-learning algoritmus kiterjesztése folytonos állapot-akció térre, fuzzy logika alkalmazásával. A folytonos állapot-akció tér (univerzum) folytonos értékű állapot- és akciódimenziót takar, végtelenszámú diszkrét értéket képviselve az adott dimenzión belül. Ebben az esetben a rendszer működtető tudásbázisa nem Q-táblában, hanem fuzzy szabályok formájában van tárolva, a tudásbázis mérete pedig a szabálybázisban lévő szabályok számával egyenértékű. Azonban a dimenziószám növekedése a szabálybázis méretének exponenciális növekedéséhez vezet, növelve ezáltal a rendszer komplexitását [49]. Az ezen módszerekben alkalmazott, általában 0-rendű Takagi-Sugeno következtetés illetve a működtető tudást leíró szabálybázis, mint univerzális függvény approximátor fogható fel, amely így a $\tilde{Q}(s,a)$ függvény közelítő leírását valósítja meg. A szabálybázis fuzzy szabályainak általános formája a következő [32]:

$$\boldsymbol{If}\ S\ is\ S_i\ \boldsymbol{And}\ A\ is\ A_u\ \boldsymbol{Then}\ \tilde{Q}(s,a) = Q_{i,u} \qquad i \in I, u \in U \tag{18}$$

Ahol, $\tilde{Q}(s,a)$ a folytonos, közelített Q-függvény, $Q_{i,u}$ a szingleton konklúzió, S_i az n-dimenziós állapottér i-edik tagsági függvénye, A_u az egydimenziós akciótér u-adik tagsági függvénye.

3.3 FUZZY SZABÁLY-INTERPOLÁCIÓ ALAPÚ MÓDSZEREK

A Fuzzy Q-learning módszerek esetében a rendszer működtető tudását leíró szabálybázis mérete exponenciálisan nő a dimenziószámmal [49]. Ennek következtében a szabályok száma a szabálybázis fedő jellege miatt, bizonyos problémák esetében (dimenziószám függő),

bizonyos idő eltelte után (epizódok száma) kezelhetetlen méretűvé válik. A klasszikus 0-rendű Takagi-Sugeno következtetési módszert kicserélve fuzzy szabály-interpolációs modellre, a szabálybázis mérete jelentősen csökkenthető, a szabálybázis ritka jellege következtében. Az egyik ilyen fuzzy szabály-interpolációs modellt alkalmazó Q-learning módszer a FRIQ-learning [97][98], de a szakirodalomban több ehhez hasonló módszer is megtalálható [36][45]. Ezen módszerek általában az alkalmazott fuzzy interpolációs modellben különböznek, kihasználva az adott interpolációs eljárás tulajdonságait.

3.3.1 Fuzzy szabály-interpoláció alapú Q-learning (FRIQ-learning)

A „Fuzzy szabály-interpoláció alapú Q-tanulás" (*Fuzzy Rule Interpolation-based Q-learning - FRIQ-learning*) [97][98] a „FIVE" fuzzy szabály-interpolációs módszert alkalmazó megerősítéses tanulási algoritmus. A „FIVE" FRI modell alkalmazása következtében a módszer folytonos állapot-akció (és Q-érték) dimenziókkal rendelkezik, a folytonos és interpolált $\tilde{Q}(s,a)$ függvényt a tanulási folyamat közben létrejött fuzzy szabályrendszer írja le, melyben a szabályok a $\tilde{Q}(s,a)$ függvény tartópontjai. A módszer Q-függvény reprezentációja a „FIVE" FRI modell alkalmazása következtében így kisebb, mint a klasszikus Q-learning módszerek esetében [S11], amely által a FRIQ-learning módszer hatékonyan alkalmazható a klasszikus megerősítéses tanulási alkalmazási példák vagy akár a „Pong" játék [S14] esetében is.

A rendszer működtető tudásbázisát (R szabálybázis) leíró i -edik ($i \in [1,m]$, m a szabályszám) r_i fuzzy szabály alakja a következő [97][98]:

$$r_i\text{: } \boldsymbol{If}\ s_1\ is\ S_1^i\ \boldsymbol{And}\ s_2\ is\ S_2^i\ \boldsymbol{And}\ \dots\ \boldsymbol{And}\ s_n\ is\ S_n^i\ \boldsymbol{And}\ a\ is\ A^i\ \boldsymbol{Then}\ \tilde{Q}(\boldsymbol{s},a) = q^i \qquad (19)$$

Ahol $\tilde{Q}(\boldsymbol{s},a)$ a közelített Q-függvény, q^i az i -edik szabály konzekvense. S_j^i ($j \in [1,n]$) a fuzzy halmaza az i–edik szabálynak a j–edik állapot univerzumban, $\boldsymbol{s} \in \boldsymbol{S}$ az n dimenziós állapot megfigyelés az n dimenziós állapot térben, s_j a j-edik ($j \in [1,n]$) dimenziója az állapot megfigyelés $\boldsymbol{s}$-nek. A^i a fuzzy halmaza az i -edik szabálynak az egydimenziós U akciótérben, $a, a \in U$ pedig a végrehajtott akció. A rendszer állapot-akció univerzuma $(n+1)$ dimenziós, ahol n az állapot dimenziók száma, a további dimenzió pedig az akcióteret jelöli. Az R szabálybázis m darab $r_i \in R$ ($i \in [1,m]$) fuzzy szabályt tartalmaz.

A „FIVE" FRI modellel közelített $\tilde{Q}(\boldsymbol{s},a)$ függvény i-edik fuzzy szabályának konzekvense a $(k+1)$-edik iterációban a következő [97][98]:

$$q_i^{k+1} = \begin{cases} q_i^k + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) & \text{ha } (\boldsymbol{s}, a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re,} \\ q_i^k + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) * \left(1/\delta_{v,i}^{\lambda}\right) \Big/ \left(\sum_{i=1}^{m} 1/\delta_{v,i}^{\lambda}\right) & \text{egyébként} \end{cases} \tag{20}$$

Ahol $\Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a)$ a Q-függvény $(k+1)$-edik iterációbeli frissítési értéke $(\mathbf{s}, \mathrm{a})$-ban amely a következő módon határozható meg [97][98]:

$$\tilde{Q}^{k+1}(\boldsymbol{s}, a) = \tilde{Q}^{k}(\boldsymbol{s}, a) + \Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) \tag{21}$$

$$\Delta\tilde{Q}^{k+1}(\boldsymbol{s}, a) = \alpha * \left(g(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma * \max_{a' \epsilon U} \tilde{Q}^{k}(\boldsymbol{s}', a') - \tilde{Q}^{k}(\boldsymbol{s}, a)\right) \tag{22}$$

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 $\boldsymbol{s} \rightarrow \boldsymbol{s}'$ állapotátmenetre, $\tilde{Q}^k$ a k-adik, $\tilde{Q}^{k+1}$ pedig a $k+1$-edik iteráció becsült konklúziója a „FIVE" FRI (*6*) által.

Összegezve, a $\tilde{Q}(\boldsymbol{s}, a)$ függvény alakja a következőképpen írható fel a „FIVE" FRI (12) modellbe való behelyettesítésével [97][98]:

$$\tilde{Q}(\boldsymbol{s}, a) = \begin{cases} q^i & \text{ha } (\boldsymbol{s}, a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re,} \\ \sum_{i=1}^{m} \left(\left(q^i \Big/ \left(\delta_v^i\right)^{\lambda} \right) \Big/ \left(\sum_{j=1}^{m} 1 \Big/ \left(\delta_v^j\right)^{\lambda} \right) \right) & \text{egyébként} \end{cases} \tag{23}$$

Ahol q^i az i-edik ($i \in [1, m]$) szabály konzekvense, $(\mathbf{s}, \mathrm{a})$ a crisp megfigyelés, λ a Shepard paraméter, amely értéke jelen esetben megegyezik az antecedens dimenziók számával [90], m pedig a szabálybázisban lévő szabályok száma. A δ_v^i a skálázott (súlyozott) távolság az $(\mathbf{s}, \mathrm{a})$ állapot-akció megfigyelés és az i-edik szabály $(\boldsymbol{s}^{\boldsymbol{i}}, a^i)$ állapot-akció antecedense között, amely a következőképpen fejezhető ki [56]:

$$\delta_v^i = \delta_v\left((\boldsymbol{s}, a), (\boldsymbol{s}^{\boldsymbol{i}}, a^i)\right) = \left[\sum_{j=1}^{n} \left(\int_{s_j^i}^{s_j} v_j(s_j) ds_j \right)^2 + \left(\int_{a^i}^{a} v(a) da \right)^2 \right]^{1/2} \tag{24}$$

Ahol $(\boldsymbol{s}, a)$ az állapot-akció megfigyelés, $(\boldsymbol{s}^{\boldsymbol{i}}, a^i)$ az i-edik szabály állapot-akció antecedense, s_j a j-edik $(j \in [1, n])$ dimenziója az n-dimenziós állapottér univerzumnak, s_j^i az i-edik szabály j-edik állapot dimenziója, a^i az i-edik szabály akció univerzuma, $v_j(s_j)$ az s_j állapot

univerzum skálafüggvénye, a $v(a)$ pedig az U akció univerzum skálafüggvénye. A skálafüggvények által az adott fuzzy halmazok alakja jellemezhető [47][56].

A tanulási fázis kezdetben 2^{n+1} darabszámú, 0 konzekvens értékkel rendelkező fuzzy szabállyal indul, amit az inkrementális szabálybázis építési módszer [94] 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 (20) 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. A lehetséges szabálypozíciók meghatározása egy állapot-akció tér rácsháló által történik, mely által az állapot-akció tér csak adott $\left(s_{k+1} = s_k, \forall k > i, s_{i+1} = \frac{s_i + s_{i+2}}{2}\right)$ pontjaiba illeszthetők be az új szabályok [94]. 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 módszer akcióválasztási politikája lehet mohó vagy akár ε-mohó is. Mohó politika követesé esetében a rendszer mindig azt az akciót hajtja végre (az adott állapotból elérhető akciók közül), amelyik várhatóan a legnagyobb Q-értéket eredményezi majd [97][98]:

$$\pi(s) = \arg\max_{a \in U} Q^{\pi}(\boldsymbol{s}, a) \tag{25}$$

Az inkrementális szabálybázis építési fázisban 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, csökkentve ez által a szabálybázis komplexitását és a működtető végleges tudásbázis méretét

[95][96]. Az elhagyható szabályok megállapítására eredetileg 3 dekrementális tudásbázis redukálási módszerrel (I., II., III.) [95][96] rendelkezik a FRIQ-learning rendszer, amelyek az inkrementális szabálybázis építési fázis után alkalmazhatók opcionálisan. Kifejlesztettem egy további inkrementális szabálybázis redukálási módszert (IV.) [S8], amely klaszterezési eljáráson alapszik, ez az értekezés későbbi 4.3.4 fejezetében kerül részletesen bemutatásra.

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. A IV. redukálási módszer inkrementális, azaz a végső redukált szabálybázis a tanulási fázis végén kapott teljes szabálybázisból a feltételezett lényegi szabályok kiemelésével keletkezik. 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 [95][96] 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 [95][96] 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 [95][96] 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.

A IV. jelölésű, általam fejlesztett szabálybázis redukálási stratégia [S8] egy klaszterezési módszeren alapszik, amely az értekezés későbbi 4.3.4. alfejezetében kerül részletesen bemutatásra.

A FRIQ-learning módszer működése tehát 2 fő lépésre bonható. Az első fázisban az inkrementális szabálybázis építési módszer [94] iterációról-iterációra bővíti, majd a tanulási fázis végeztével létrehozza a rendszert működtető végleges tudásbázist. A második fázisban a tanulási folyamat végén előállt szabálybázis méretének (szabályainak számának) csökkentésére van lehetőség az adott dekrementális szabálybázis csökkentési módszerek [95][96][S8] (I., II., III., és IV.) opcionális alkalmazása által.

3.4 HEURISZTIKÁVAL BŐVÍTETT MÓDSZEREK

A klasszikus megerősítéses tanulási módszerek problémája az esetleges lassú konvergencia sebesség, magas iterációszám [14]. Ennek oka ezen módszerek előnyében keresendő, amely által képesek az állapottér feltérképezésével, próbálkozásokkal megoldást találni egy olyan problémára, melyről kezdetben semmilyen előzetes információ nem állt rendelkezésre. Tehát a rendszer a tanulási fázis kezdetén nem rendelkezik semmilyen előzetes tudásbázissal az adott probléma megoldására vonatkozóan, így az annak méretétől (dimenziószámától) függően több-kevesebb epizód alatt, számos próbálkozással találja meg a helyes megoldást. A teljesítménymértékek (konvergencia sebesség, megoldáshoz vezető iterációk száma) értéke az állapottér dimenziószámának növekedésével pedig egyre csak növekszik.

Az említett problémák kiküszöbölésére jelenthetnek megoldást azon megerősítéses tanuló rendszerek, amely rendelkeznek valamilyen előzetes (*a priori*) tudással (*heurisztikával*) az adott feladat megoldására vonatkozóan. Heurisztika alatt ebben az esetben korábban megszerzett tapasztalat, az adott megerősítéses tanulási feladat megoldására vonatkozó előzetes (és

részleges) tudásbázis értendő, amely ember (azaz szakértő) által meghatározott és a rendszer szempontjából külső információ. Fontos megemlíteni, hogy ez az a priori heurisztika általában nem a teljes megoldás leírását jelenti, hiszen abban az esetben a megoldás már ismert, hanem a teljes megoldásnak csak a rendelkezésre álló, adott részét. Tehát ez az előzetes tudásbázis nem az optimális politikát definiálja és a rendszerhez viszonyítva kívülről származik, nem a tanulási folyamat közben jött létre. Megtalálhatóak olyan módszerek is, melyek a tanulási fázis közben létrejött tudást használják fel újra. Ilyen például a *„Transfer Learning"* [91], amely esetében egy korábbi tanulási folyamat során létrejött tudásbázis kerül felhasználásra egy másik, de nagyon hasonló probléma megoldására. Egy másik, már meglévő tudásbázist felhasználó módszer a multiágens rendszer [87], amely estében együttműködő ágensek használják fel egymás tudásbázisait.

Több szerző által is említésre kerül a megerősítéses tanulási rendszer valamilyen módon történő előzetes szakértői információval történő bővítése [15][21][33][78]. A következő alfejezet ezen szempontokból tekinti át a témához kapocslódó, publikációkban megtalálható, szakértői információval bővített megerősítéses tanulási módszereket.

3.4.1 Heurisztikusan gyorsított megerősítéses tanulás

A [15]-ben bevezetett heurisztikusan gyorsított megerősítéses tanulási módszerek az eredeti Q-learning és SARSA [99] algoritmusok módosított változatai. Ezen módszerek rendelkeznek a probléma megoldására vonatkozó részleges tudásbázissal [16], összefoglaló nevük magyar fordítása a „heurisztikusan gyorsított megerősítéses tanulás" (*Heuristically Accelerated Reinforcement Learning - HARL*). Ebben az esetben egy úgynevezett $H_t(s_t, a_t)$ heurisztikus függvény formájában van definiálva az előzetes heurisztika. Ez a H $(H: S \times A \to R)$ függvény egy politika módosítónak tekinthető, azt definiálja, hogy mely s_t állapotban, mely a_t akció végrehajtása preferált az adott t időpillanatban.

A kapcsolat a heurisztikus függvény és az akció-érték függvény között a következő [15]:

$$F_t(s_t, a_t) \bowtie \xi H_t(s_t, a_t)^{\beta} \tag{26}$$

Ahol $F: S \times A \to R$ az értékfüggvény becslése (Q-learning esetében $\tilde{Q}_t(s_t, a_t)$), $H: S \times A \to R$ a heurisztikus függvény, amely az adott a_t akció végrehajtásának preferálását határozza meg s_t-ben, $\bowtie$ függvény, amely a rendezett halmazból állít elő értéket (valós szám), ξ és β pedig a heurisztikus függvény paraméterei, melyek a H függvény rendszerre történő hatását befolyásolják, azaz, hogy H milyen mértékben érvényesüljön.

A heurisztikusan gyorsított megerősítéses tanulási algoritmusok a [15] publikáció szerzői által bevezetett HAQL (*Heuristically Accelerated Q-learning*), HA-Q(λ), HA-SARSA(λ) és HA-TD(λ) (*Heuristically Accelerated Temporal Difference*) algoritmusok, melyek az eredeti Q-learning, Q(λ), SARSA(λ) és TD(λ) módszerek heurisztika alapú változatai [14][15]. A heurisztikusan gyorsított megerősítéses tanulási módszerek általános algoritmusának pszeudokódja az alábbi [15]:

Algoritmus: Heuristically Accelerated Q-learning

Tetszőleges becslés létrehozása az értékfüggvényre
Kezdeti $H_t(s,a)$ heurisztikus függvény definiálása megfelelő módszerrel
Az aktuális s állapot megfigyelése
Ismétlés:
    a akció választása a heurisztikus és az értékfüggvény megfelelő kombinálásával
    a akció végrehajtása
    $r(s,a)$ megerősítés és s' állapot megfigyelése
    $H_t(s,a)$ heurisztikus függvény frissítése megfelelő módszerrel
    Értékfüggvény frissítése
    $s \leftarrow s'$ állapot frissítése
amíg a leállási feltétel nem teljesül
ahol $s = s_t, s' = s_{t+1}$ and $a' = a_t$

3. algoritmus: A heurisztikusan gyorsított megerősítéses tanulási módszerek általános algoritmusa [15]

3.4.2 Heurisztika leírásának elterjedtebb módszerei

Heurisztika definiálása alatt a rendszer szempontjából külső szakértői információ leírásának illetve az adott formában leírt külső információ megerősítéses tanulási rendszerbe történő injektálásának módja értendő. A 3.4.1 alfejezetben bemutatott heurisztikusan gyorsított megerősítéses tanulás esetében a rendszer számára külső információ (heurisztika) egy *H* heurisztikus függvény formájában definiálható, mint politikamódosító. A *H* függvény leírását megvalósító módszereket 2 csoportba lehet bontani. Az egyik csoportba azok a módszerek tartoznak melyek korábbi ismereteket alkalmaznak a heurisztika következtetésére, vagy újra felhasználjak egy korábbi feladatban megtanult akcióválasztási politikát („ad hoc" mód). A másik csoportba azon módszerek sorolhatók, melyek a tanulási folyamatból származó információkat használják fel, ilyen lehet az aktuális akcióválasztási politika, az értékfüggvény, állapottér trajektória [15].

Több szerző is javasol más, a heurisztikus függvény leírási módjától eltérő tudásbázis megadási formát, amely alkalmas lehet kezdeti szakértői tudásbázis injektálására a megerősítéses tanuló rendszerbe. Az egyik ilyen lehetséges leírási forma a „GOAL" (*Goal-Oriented Agent Language*) [35] azaz a cél-orientált ágens programozási nyelv, amely ember számára is olvasható *„if then"* típusú szabályok által írja le az ágens számára az akcióválasztás módját. Ezen tudásreprezentációs nyelv által különböző névvel ellátott egységekben, kapcsos

zárójelek között definiálhatók az adott funkciójú nevesített blokkok. A célállapotok a *„goals”* nevű blokkban, az előnyben részesített akciók az *„actionspec”* nevű blokkban, azok várható hatása *„pre”* és *„post”* kulcsszóval a blokkon belül, az állapotok a *„beliefs”* nevű blokkban, az *„if then”* típusú szabályok pedig a „program” nevű blokkon belül találhatók. Ez a GOAL nyelv alkalmas lehet külső (nem a rendszerből származó) információ leírására az ágens viselkedésére vonatkozóan [19]. Az ilyen módon megadott akcióválasztási szabályok a tanulási folyamat során nem változtathatók meg.

A fuzzy szabály alapú megerősítéses tanulási módszerekben (mint például a fuzzy Q-learning vagy a fuzzy szabály-interpoláció alapú Q-learning módszerek) kézenfekvő, hogy a rendszer tudásbázisát leíró fuzzy szabályok formájában lenne célszerű megadni az előzetes szakértői tudásbázist is. Ezt a leírási, megadási formát javasolják a [75] publikáció szerzői is.

3.4.3 Kezdeti Q-érték meghatározásának elterjedtebb módszerei

A diszkrét felbontású Q-learning módszer esetében a Q-táblában tárolt állapot-akció párokra vonatkozó Q-értékek kezdetben (a tanulási folyamat elején) 0 (zéró) értékkel vannak inicializálva. A fuzzy szabályalapú illetve a fuzzy szabály-interpoláció alapú Q-learning módszerek esetében (például a FRIQ-learning [98]) a kezdeti Q-értékek a szabálybázis kezdeti szabályinak konzekvenseiben jelennek meg zérusként. A fuzzy szabályalapú megerősítéses tanulási rendszerek esetében, az előzetes szakértői tudásbázis szabályaira célszerű lehet valamilyen kezdeti (tanulási fázis előtti), de 0-tól eltérő Q-érték (vagy állapot érték) meghatározása. Erre a szakértői tudásbázis megerősítéses tanuló rendszerbe történő injektálása miatt van szükség, valamint az előzetesen meghatározott Q-érték hatással lehet a rendszer konvergencia sebességére [72][S7]. A 0-tól eltérő Q- vagy állapot-érték a szakértői szabály konzekvensében megadott, adott állapotban lévő akció végrehajtásának előnyben részesítését jelzi. Mivel a Q-értékek szakértő által történő meghatározása nehézkes (szinte lehetetlen), így különféle módszerek alkalmazása, kidolgozása szükséges ezen előzetes jóságértékek számításához. Több publikációban is található javaslat illetve módszer kezdeti, azaz a tanulási fázis előtt inicializált Q-érték (vagy állapot érték) meghatározására [65][72][75]. Ez történhet szakértő által leírt, a rendszer szempontjából külső tudásbázis alkalmazása következtében illetve a tanulási folyamat iterációszámának csökkentése érdekében is.

Egyik lehetséges kezdeti állapot-érték számítási módszer fuzzy Q-learning alkalmazása esetében az egyes fuzzy univerzumok tagsági függvényei alapján történik [75]. Ebben az esetben a szakértői szabályrendszer által az egyes állapotokban preferált akciók vannak meghatározva, majd ezen állapotokban kerül kiszámításra az előzetes állapot-érték. A Q-learning módszer

frissítési formulája módosul az előzetesen számított állapot-értékek következtében úgy, hogy a meghatározott állapotérték adott súllyal (β) jelenik meg az összefüggésben [75]. Egy másik hasonló, folytonos állapot-akció térrel rendelkező fuzzy Q-learning módszerben [73] fuzzy szabályok által, az egyes fuzzy partíciók tagsági függvényeinek megadásával határozzák meg a kezdeti Q-értékeket, javítva ezzel a módszer hatékonyságát. A tanulási fázis előtt inicializált Q-értékek (Q_i) hatását vizsgálja [65] a tanulási folyamatra. Egyik esetben egy bináris jutalomfüggvény által határozza meg Q_i értékeket. A bináris jutalomfüggvény által a jutalom mindig végtelen ($r = r_\infty$) kivéve abban az esetben, mikor az aktuális meglátogatott állapot megegyezik a célállapottal, ekkor $r = r_g$. Ha $r_g = r_\infty$ akkor $Q_\infty = r_\infty/(1-\gamma)$, Q_i értéket ezen összefüggés alapján célszerű megválasztani. Folytonos állapottérrel rendelkező Q-learning módszer esetében folytonos jutalomfüggvény alkalmazható, amely következtében a kezdeti Q-értékek például Gauss eloszlási függvény segítségével kerülnek inicializálásra. Egy másik lehetséges eset mikor ugyanazon értékekkel kerül inicializálásra Q_i, ekkor $Q_i = \beta/(1-\gamma)$, ahol β konstans érték [65].

4 HEURISZTIKUSAN GYORSÍTOTT FRIQ-LEARNING

Ebben a fejezetben a szakértői tudásbázissal bővített (heurisztikusan gyorsított) FRIQ-learning módszer (HFRIQ-learning) és a hozzá kapcsolódó tézisek kerülnek bemutatásra, több alfejezetre bontva.

Elsőként az előzetes szakértői tudásbázis definiálásának, leírásának módja kerül bemutatásra, amely által az előzetes szakértői tudásbázis beépíthető az FRIQ-learning megerősítéses tanulási rendszerbe. Ezt követően egy Q-érték inicializációs módszer ismertetése következik, amely az előzetes szakértői heurisztika FRIQ-learning rendszerbe történő injektálása miatt szükséges.

Feltételezve annak lehetőségét, hogy az előzetesen definiált szakértői szabályrendszer tartalmazhat nem feltétlenül helyes szabályokat amelyek negatív hatással lehetnek a rendszer hatékonyságára, egy gradiens módszer alapú Q-függvény hangolási eljárás is kifejtésre kerül. Ennek alkalmazásával a tanulási folyamat közben, az előzetesen megadott fuzzy szabályrendszer (és az általa leírt Q-függvény) hangolható, optimalizálható.

A szabályrendszer antecedens és konzekvens értékeinek hangolása következtében a tanulási folyamat során hasonló szabályok kerülhetnek egymáshoz közel. Ezen szabályok valamilyen módszerrel történő egyesítése (vagy akár egyikük elhagyása) által, a szabálybázis mérete már a tanulási fázis során csökkenthető. Bevezetésre kerül ezért egy olyan fuzzy szabályok közötti távolságon alapuló szabálybázis redukálási módszer is, amely már a tanulási folyamat során alkalmazható.

4.1 SZAKÉRTŐI TUDÁSBÁZIS BEÉPÍTÉSE

Ebben az alfejezetben a szakértői tudásbázis megadásának illetve az FRIQ-learning rendszerbe történő beágyazásának, injektálásának módja kerül bemutatásra.

4.1.1 Szakértői tudásbázis leírási forma

A FRIQ-learning rendszer tudásbázisa egy ritka fuzzy szabályrendszer, amely az interpolált Q-függvényt írja le a szabálypontok (mint tartópontok) által. A szabályrendszer szabályai a (19) formula alapján meghatározott „állapot-akció-Q-érték” formátumúak. Az állapot-akció rész a szabály $n+1$ dimenziós antecedense, a Q-érték a szabály singleton konzekvense. Ennek megfelelően a rendszer szempontjából külső információként megjelenő szakértői tudásbázis is fuzzy szabályok formájában kerül leírásra, produkciós szabályokként, a megfelelő állapotokban

előnyben részesített akciók meghatározásával. A szakértői szabályrendszer $\left(R_{expert}\right)$ i-edik $\hat{r}_i$ $(i \in [1, \hat{m}])$ szabályának formája a következő:

$$\hat{r}_i\text{: } \textbf{If } s_1 \textbf{\textit{is}}\ \hat{S}_1^i \textbf{ \textit{And}}\ s_2 \textbf{\textit{is}}\ \hat{S}_2^i \textbf{ \textit{And}} \dots \textbf{\textit{And}}\ s_n \textbf{\textit{is}}\ \hat{S}_n^i \textbf{ \textit{Then}}\ \text{a} = \hat{A}^i \tag{27}$$

Ahol $\hat{A}^i$ az i-edik $(i \in [1, \hat{m}])$ szakértői szabály konzekvenseként definiált akció, $\hat{S}_n^i = \left[\hat{S}_1^i, \hat{S}_2^i, \dots \hat{S}_n^i\right]$ az n-dimenziós állapot megfigyelés, $\hat{m}$ az R_{expert} szakértői szabályrendszer szabályainak számossága, $\hat{r}_i$ az i-edik szakértői szabály, $\hat{r}_i \in R_{expert}$.

Az így megadott szabályrendszer szabályai hasonlóak a (19) formula által leírt szabályokhoz, azzal az eltéréssel, hogy ezekben az akció nem antecedensként hanem konzekvensként jelenik meg. Ennek oka, hogy a szakértő ezen szabályok konzekvenseként azaz akciójaként, azt határozza meg a rendszer számára, hogy mely állapotokban milyen akciók végrehajtása előnyös. Az n-dimenziós állapotok pedig a szakértői szabályok antecedensei. Lényeges, hogy a szakértő által bármennyi fuzzy szabály definiálható a (27) formátumban, a 2.2.1 fejezetben bemutatott interpolációs módszer következtében nem szükséges minden egyes lehetséges állapotra szakértői szabályt létrehozni, elég csak a meglévő ismeretet leírni.

A szakértő által megadott a priori szabályrendszer az eredetileg mohó vagy -mohó akcióválasztási politika módosítójaként jelenik meg. Ha az éppen megfigyelt állapotban áll rendelkezésre szakértő által meghatározott akció, azaz létezik olyan szakértői szabály amely antecedense illeszkedik a megfigyelésre, akkor a rendszer a szakértői szabály konzekvenseként megjelenő akciót fogja végrehajtani, a FRIQ-learning mohó (vagy ε-mohó) politikája által meghatározott akciója helyett. Ennek következtében a szakértői szabályrendszer egy heurisztikus politika módosítóként [15] is tekinthető és az alábbi módon módosítja a FRIQ-learning mohó (vagy ε-mohó) politikáját:

$$\pi(\boldsymbol{s}) = \begin{cases} \text{a} = \hat{A}^i & \text{ha } \boldsymbol{s_n} = \widehat{\boldsymbol{S}}_{\boldsymbol{n}}^{\boldsymbol{i}} \text{ minden } i\text{-re,} \\ \arg\max_{a \in U} Q^{\pi}(\boldsymbol{s}, a) & \text{egyébként} \end{cases} \tag{28}$$

Ahol $\boldsymbol{s_n}$ az n-dimenziós állapot megfigyelés, $\widehat{\boldsymbol{S}}_{\boldsymbol{n}}^{\boldsymbol{i}}$ az i-edik illeszkedő szakértői szabály antecendse (állapot dimenziója), $\hat{A}^i$ pedig az $\widehat{\boldsymbol{S}}_{\boldsymbol{n}}^{\boldsymbol{i}}$ szakértői szabály antecedenshez tartozó konzekvens akció. Ha az aktuális $\mathbf{s_n}$ megfigyelés illeszkedik valamelyik szakértői szabály $\widehat{\mathbf{S}}_{\mathbf{n}}^{\mathbf{i}}$ antecedensére, akkor a rendszer által végrehajtott akció a szakértői által konzekvensként megjelenő $\hat{A}^i$ akció lesz. Ellenkező esetben a végrehajtott akció a rendszer által követett mohó vagy ε-mohó politika által meghatározott akció lesz.

Mivel az előzetes szakértői szabályrendszer emberi szakértői által definiált, így annak helyessége első megközelítésként feltétlenül elfogadható, azaz, feltételezhető, hogy az összes szakértői szabály helyes. Helyesség alatt itt az értendő, hogy a megfelelő állapotokban olyan akciók lettek megadva, amelyek az ágens viselkedésére pozitívan hatnak és így ezáltal a rendszer konvergencia sebessége javulhat, a tanulási fázis lerövidülhet. Azonban olyan esetek is előfordulhatnak, mikor csak részben helyes, teljesen helytelen, vagy akár véletlenszerű szakértői szabályrendszer kerül megadásra. Az ilyen módon megadott szakértői tudásbázisoknak a tanulási fázis konvergencia sebességére gyakorolt hatása az értekezés későbbi részében kerül kifejtésre [S7].

4.1.2 A szakértői szabályok kezdeti Q-értékeinek meghatározása

Az előzetes tudásbázis a szakértő által szabályrendszer formájában kerül leírásra a (27) formula által meghatározott módon. A szabályrendszerben az antecedens a többdimenziós állapot univerzumot, a konzekvens pedig az akció dimenziót jelöli, azaz ezen szabályok definiálásukkor nem rendelkeznek Q-értékkel. A FRIQ-learning módszer szabályrendszere a (19) formula szerint állapot-akció-Q-érték formátumú, ahol az antecedens a többdimenziós állapot és az akció, a konzekvens pedig a Q-érték. Annak érdekében, hogy a szakértői szabályok a FRIQ-learning szabályrendszerébe illeszthetőek legyenek, meg kell határozni a szabályok kezdeti Q-érték konzekvensét. A szakértői szabályok akció konzekvensei az antecedens oldalra kerülnek, majd az új konzekvensük pedig ezen Q-érték lesz. Ezt a folyamatot még a tanulási fázis megkezdése előtt kell megvalósítani. A kezdeti Q-érték meghatározási módszer célja tehát még a tanulási fázis előtt, a szakértői szabályrendszer minden egyes szabályára becsült Q-érték $\left(\tilde{Q}_{init}\right)$ inicializálása, azaz a kezdeti Q-függvény meghatározásra. Ez a szakértői heurisztikából létrehozott kezdeti Q-függvény lesz hangolva a tanulási folyamat során.

Feltételezve, hogy a szakértő által megadott szabályok megkérdőjelezhetetlenül helyesek, az azokra meghatározott Q-értékeknek relatívan magasnak (és lehetőleg 0-tól eltérőnek) kell lenniük. A relatívan magas érték egy kezdeti becslés, amely a környezet által maximálisan adható g_{max} jutalom ismeretében következők alapján határozható meg [S7]:

$$\tilde{Q}_{init} = \eta * \tilde{Q}_{max} \tag{29}$$

$$\tilde{Q}_{max} = \lim_{k\to\infty} \tilde{Q}^{k+1}(\boldsymbol{s}^*, a^*) = \lim_{k\to\infty} (\tilde{Q}^k(\boldsymbol{s}^*, a^*) + \alpha * g(\boldsymbol{s}^*, a^*, \boldsymbol{s}^*) + \gamma * \tilde{Q}^k(\boldsymbol{s}^*, a^*) - \tilde{Q}^k(\boldsymbol{s}^*, a^*))) \tag{30}$$

$$\tilde{Q}^k(\boldsymbol{s}^*, a^*) = \max_{a' \in U} \tilde{Q}^k(\boldsymbol{s}^*, a') \text{ és } g(\boldsymbol{s}^*, a^*, \boldsymbol{s}^*) = \max_{\boldsymbol{s}\in\boldsymbol{S}, a\in U} g(\boldsymbol{s}, a, \boldsymbol{s}') = g_{max} \tag{31}$$

$$\begin{aligned} \tilde{Q}_{max} &= \lim_{k\to\infty} \tilde{Q}^{k+1}(\boldsymbol{s}^*, a^*) = \lim_{k\to\infty} (\tilde{Q}^k(\boldsymbol{s}^*, a^*) + \alpha * (g_{max} + (\gamma - 1) * \tilde{Q}^k(\boldsymbol{s}^*, a^*))) \\ &= \\ &= \tilde{Q}^k(\boldsymbol{s}, a) + \alpha * g(\boldsymbol{s}, a, \boldsymbol{s}') + \alpha * (\gamma - 1) * \tilde{Q}^k(\boldsymbol{s}', a') = \frac{\alpha * g_{max}}{-\alpha * (\gamma - 1)} = \\ &= \frac{g_{max}}{1 - \gamma} \end{aligned} \tag{32}$$

$$\tilde{Q}_{init} = \eta * \frac{g_{max}}{1-\gamma} \text{ ha } \gamma < 1 \tag{33}$$

Ahol $\tilde{Q}_{max}$ a becsült, maximálisan elérhető Q-érték a környezet által maximálisan adható jutalom g_{max} ismeretében, $\eta \in [0,1]$ a $\tilde{Q}_{init}$ skála faktora, amely azt határozza meg, hogy a számított $\tilde{Q}_{init}$ érték mekkora része (százaléka) kerüljön figyelembevételre, α a tanulási ráta, γ pedig a diszkontálási tényező. Az így kiszámított $\tilde{Q}_{init}$ érték a szakértői szabályrendszer minden egyes szabályára, azaz annak konzekvens részére egyforma értékű. A szabályrendszer formája az előzetes Q-érték meghatározási módszer alkalmazása után a (27) összefüggés alapján a következőképpen módosul:

$$\hat{r}_i\text{: } \textbf{If } s_1 \textbf{\textit{ is }} \hat{S}_1^i \textbf{ \textit{And}} \ s_2 \textbf{\textit{ is }} \hat{S}_2^i \textbf{ \textit{And}} \ \ldots \textbf{ \textit{And}} \ s_n \textbf{\textit{ is }} \hat{S}_n^i \textbf{ \textit{And}} \text{ a} = \hat{A}^i \textbf{ \textit{Then}} \ \tilde{Q}(\boldsymbol{s}, a) = \tilde{Q}_{init} \tag{34}$$

4.1.3 Szakértői tudásbázis adoptálása

Az FRIQ-learning rendszer 2^{n+1} (n az állapot univerzum dimenziószáma) darabszámú szabállyal rendelkező kezdeti szabálybázist hoz létre a tanulási folyamat kezdetével, amely egy kezdeti Q-függvényt határoz meg. Ezen kezdeti szabályok konzekvense, azaz Q-értéke rendre $q_i = 0$ értéket vesz fel, az $(n+1)$-dimenziós hiperkocka sarkaiban (sarokponti szabályok) [94]. A FRIQ-sarokponti szabálybázisát alkotó szabályokat (FRIQ-kezdeti szabályok) jelöljük a következőképpen:

$$r_i^{\square}\text{: } \textbf{\textit{If }} s_1 \textit{ is } S_1^{\square i} \textbf{ \textit{And}} \ s_2 \textit{ is } S_2^{\square i} \textbf{ \textit{And}} \ \ldots \textbf{ \textit{And}} \ s_n \textit{ is } S_n^{\square i} \textbf{ \textit{And}} \ a \textit{ is } A^{\square i} \textbf{ \textit{Then}} \ \tilde{Q}(\boldsymbol{s}, a) = 0 \tag{35}$$

Ahol $S_l^{\square i} \in [\min(S_l), \max(S_l)]$ $(\forall i \in [1, 2^{n+1}], \forall l \in [1, n])$ és $A^{\square i} \in [\min(A), \max(A)]$ $(\forall i \in [1, 2^{n+1}])$ a sarokponti állapot- és akcióértékek, $r_i^{\square} \in R$ $(i \in [1, 2^{n+1}])$ az i -edik sarokponti szabály, n pedig az állapotdimenziók száma. Például egy 2 állapotdimenzióval rendelkező probléma esetében $2^{2+1} = 8$ darab (35) formátumú szabályt hoz létre a rendszer a tanulási fázis kezdetén.

Mivel a rendszer tanulási folyamat kezdetén 2^{n+1} darabszámú fuzzy szabállyal rendelkezik, a szakértői szabályrendszer pedig ebbe a (35) formátumú sarokponti szabályrendszerbe beillesztésre kerül, így a 2 szabályrendszer, tehát a FRIQ sarokponti szabályok és a szakértői szabályok összefésülése szükséges. A szakértő által definiált előzetes heurisztikát leíró szabályrendszer szabályainak száma $\hat{m}$. A sarokponti szabályrendszer, amely 2^{n+1} darabszámú szabállyal rendelkezik az $\hat{m}$ darabszámú szakértői szabállyal rendelkező szabályrendszerrel fog kiegészülni, így a teljes (összefésült) kezdeti szabályrendszer szabályainak számossága $2^{n+1} + \hat{m}$. A két szabályrendszer formulája az előzetes Q-érték meghatározási módszer alkalmazása után ugyanaz, azzal a különbséggel, hogy a sarokponti szabályok konzekvense $\tilde{Q}(\boldsymbol{s}, a) = 0$, a szakértői szabályrendszer konzekvense pedig $\tilde{Q}(\boldsymbol{s}, a) = \tilde{Q}_{init}$ értékkel rendelkezik. A (35) formula a FRIQ sarokponti szabályrendszer, a (34) formula pedig az előzetes Q-érték meghatározási módszert követő szakértői szabályrendszer szabályainak felépítését szemlélteti.

Előfordulhat azonban olyan eset is, mikor a két összefésült kezdeti szabályrendszer azonos antecedenssel rendelkező szabályokat tartalmaz. Ez abban az esetben lehetséges, mikor a szakértő olyan szabályt definiál, amely éppen sarokponti szabály antecedensére illeszkedik, tehát a szakértői szabály teljes része (állapot antecedense és akció konzekvense) illeszkedik a sarokponti szabály állapot-akció antecedensére. Ekkor az előzetes Q-érték meghatározási módszert követően - amely következtében a szakértői szabályok akció konzekvenseiből antecedens lesz és kiegészülnek Q-érték konzekvenssel - a rendszerben két ugyanolyan antecedenssel, de különböző konzekvens értékkel rendelkező szabály lesz. Ez a két illeszkedő szabály ellentmondást eredményez, ugyanarra az antecedens állapot-akció pontra eltérő konzekvens értékeket definiálnak. Az ellentmondás feloldásaként a két szabályrendszer összefésülésekor ellenőrzi kell, hogy valamely szakértői szabály sarokponti szabályra esik-e. Ha igen, akkor az adott sarokponti és az ellentmondó szakértői szabály egyesítése valósul meg olyan módon, hogy a sarokponti szabály 0 értékű konzekvense (Q-értéke) lecserélése kerül az illeszkedő szakértői szabály konzekvensére (Q-értékére), majd a szakértői szabály törlésre kerül, aminek következtében az ellentmondás megszűnik. A sarokponti szabályok fontos

szerepet töltenek be a „FIVE” interpolációs módszer alkalmazása következtében, így ezek törlése nem megvalósítható. Az összefésült szabályrendszer szabályainak száma így csökkeni fog az ellentmondó szabályok számának felével, ennek oka, hogy az ellentmondó szabálypárokból az egyik szabály mindig eltávolításra kerül. Például ha két olyan különböző szakértői szabály lett definiálva, amely két különböző sarokponti szabályra illeszkedik, akkor a 2-2 ellentmondó szabálypárból 1-1 szakértői szabály eltávolításra kerül.

A szakértői szabálybázist és az FRIQ-sarokponti szabálybázist összefésülő, fejlesztett módszer algoritmusának pszeudokódja az alábbi [S7]:

Algoritmus: injectExpertRB(expertR)

Bemenet: szakértői szabálybázis
Kimenet: tanulási fázis kezdeti szabálybázisa

$\tilde{Q}_{init}$ számolása a szakértői szabálybázisra
Szakértői szabályok akció konzekvensének lecserélése $\tilde{Q}_{init}$ értékre, mint új konzekvens
Ciklus (minden egyes szakértői szabályra)
    If (szakértői szabály antecedense illeszkedik FRIQ-sarokponti szabály antecedensére)
        FRIQ-sarokponti szabály 0 konzekvensének lecserélése $q_i = \tilde{Q}_{init}$ konzekvensre
        illeszkedő (ellentmondó) szakértői szabály törlése
    end
return kezdeti szabálybázis

4. algoritmus: A szakértői szabálybázist és az FRIQ-sarokponti szabálybázist összefésülő, fejlesztett módszer algoritmusa

Jelöljük az ellentmondó szabályokat r_i^c -vel ($r_i^c \in \left(R\ U\ R_{expert}\right), i \in [1, c]$) , ahol az ellentmondó szabályok száma c, ekkor az összefésült $R\ U\ R_{expert}$ kezdeti szabályrendszer szabályainak száma az ellentmondó szabályok eltávolítása után $(2^{n+1} + \hat{m}) - (c/2)$.

Mivel a szakértő által megadott szabályok konzekvens része az előzetes Q-érték meghatározási módszer végrehajtását követően a számított és valószínűleg 0-tól különböző Q-érték lesz, így azok jelentősebb befolyással vannak a rendszer működésére, fontos szabályrendszerként (tudásbázisként) kezelendő.

4.1.4 Szakértői szabályrendszer adoptálásának blokkvázlata

A következő 7. ábra a fejlesztett, szakértői tudásbázis injektálására lehetőséget biztosító FRIQ-learning megerősítéses tanulási rendszer felépítését szemlélteti:

7. ábra: Az előzetes szakértői tudásbázissal kiegészített FRIQ-learning megerősítéses tanulási rendszer

4.1.5 „Mountain Car" mintapélda szakértői tudásbázis injektálásával

Az előzetes szakértői heurisztika (mint szakértői állapot-akció produkciós szabályrendszer) FRIQ-learning rendszerbe történő injektálását és az általam javasolt módszerek alkalmazását ezen alfejezet egy elterjedt megerősítéses tanulási mintapéldán keresztül mutatja be.

A választott mintaalkalmazás a népszerű megerősítéses tanulási problémák közül a „Mountain Car" nevezetű. Az alkalmazáspéldában az ágens egy autó, környezete pedig egy meredek völgy. Az autó a meredek völgy közepén helyezkedik el a tanulási folyamat indulásakor. Az ágens célja, hogy kijusson a meredek völgy közepéből a völgy tetején található dombra. A feladat akkor tekinthető megoldottnak, ha az autó, azaz az ágens valamennyi meghatározott lépés alatt (jelen esetben 1000 lépés alatt) kijut a völgyből és eléri a domb tetején lévő csillagot. A környezettől akkor érkezik nagy megerősítés (r) ha ez a feladat sikerül, azaz az ágens pozíciója eléri a csillag pozícióját, ellenkező esetben pedig büntetést ad a rendszer.

A „Mountain Car" nevezetű megerősítéses tanulási probléma állapottere 2 változós (s_1, s_2), az akció tér pedig 1 változóval (a) rendelkezik, melyek a következők:

- autó aktuális pozícióját: s_1 ($s_1 \in [-1.5, 0.5]$)
- autó aktuális sebessége: s_2 ($s_2 \in [-0.07, 0.07]$)
- az autó elmozdulása: a (jobbra, balra vagy nincs elmozdulás, $a = [-1, 0, 1]$)

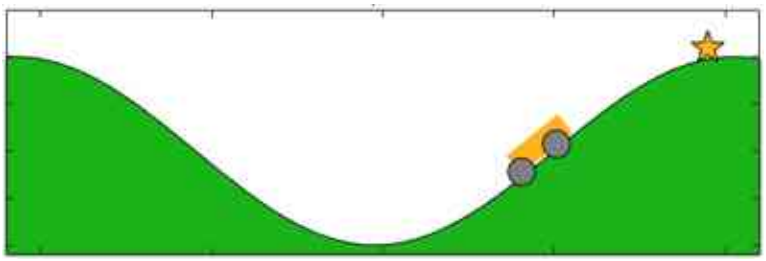
A jutalomfüggvény a következő:

Jutalomfüggvény: Mountain Car

```
Bemenet: s állapot
Kimenet: r megerősítés

bpright_position = 0.45;
if (s_1  >= bpright_position)
    r = 1000
else
    r = -10

end

return r
```

1. jutalomfüggvény: A "Mountain Car" probléma jutalomfüggvényének pszeudokódja

A következő ábrán a „Mountain Car” Matlab szimulációja látható, melyet diszkrét környezetre Jose Antonio Martin H. implementált [43]:



8. ábra: A „Mountain Car" probléma szimulációja [43]

A rendszer előzetes szakértői tudásbázisának megadása állapot-akció alakú fuzzy szabályok formájában lehetséges. Ezen szakértő által definiált tudásbázis rendszerre gyakorolt hatásának megállapításához 4 különböző futtatási esetet hoztam létre, melyek a következők:

1. helyesen megadott heurisztika
2. az előző 1. esetben megadott heurisztika csak egy része
3. részben helytelen szakértői szabályrendszer
4. „véletlenszerűen” generált szakértői szabálybázis

Az összehasonlítás alapja az üres tudásbázissal, azaz a szakértői heurisztika nélkül működő FRIQ-learning rendszer által adott konvergencia sebesség és fuzzy szabálybázis méret (azaz szabályszám). Minden egyes esetben 10 külön, egymástól független futtatás valósult meg, különböző kezdeti állapottér pozíciókkal. Az egyes futtatási eredmények adataiból, az egyes konvergencia sebességek és szabálybázis méretek átlagai kerültek meghatározásra. A szabálybázis méretek mindegyik esetben a szabálybázis redukálási stratégiák alkalmazása nélküli szabálybázis méreteket jelölik.

A rendszer futtatási paraméterei és azok értékei a következők:

- szakértő által megadott megerősítés értéke: $g_{max} = 100$
- tanulási ráta: $\alpha = 0.5$
- leszámítolási tényező: $\gamma = 0.99$

Amikor a rendszer szakértői tudásbázis nélkül kerül futtatásra akkor az átlagos konvergencia sebesség 28.3 epizód, szabálybázis méret pedig 91.7 szabály. A következő táblázat foglalja össze az ebben az esetben kapott futási eredményeket:

1. táblázat: Szakértői heurisztika nélküli futtatási eset eredményei

Futtatási eset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Átlag**
Konvergencia sebesség	23	36	34	35	20	34	25	26	29	21	28.3
Szabálybázis méret	80	85	82	96	105	90	89	98	99	93	91.7

Az 1. futtatási esetben a helyesen megadott szakértői tudásbázis kerül beillesztésre a rendszerbe és a tanulási fázis ezzel a szabálybázissal indul el. A helyes szakértői szabálybázist az előző futtatási esetből, az inkrementálisan felépített szabálybázisból nyertem ki a szabálybázis redukálási módszerek alkalmazását követően. Az így kapott redukált méretű szabálybázis 17 szabályát ($\hat{m} = 17$) a következő táblázat tartalmazza:

2. táblázat: A helyesen megadott szakértői szabályok felépítése

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
s1	-0.5	-0.475	-0.475	-0.27	-0.27	-0.475	-0.065	-0.475	-0.68
s2	0	-0.014	0.014	0.014	-0.014	0.042	-0.014	-0.042	0.042
a	-1	1	1	1	1	1	0	-1	-1

R#	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
s1	-0.065	-0.065	0.14	-0.27	-0.885	-0.65	-1.09	0.14
s2	0.042	0.014	-0.014	-0.042	0.042	0.042	0.042	-0.014
a	1	0	-1	-1	-1	0	-1	0

A maximális szakértő által megadott megerősítés ($g_{max} = 100$) következtében a 2. táblázatban lévő szabályokra a (33) összefügés által számított kezdeti Q-értékek $\tilde{Q}_{init} = 10000$. A következő lépésben a (33) összefüggésben lévő η érték rendszerre történő hatásának vizsgálatát valósítottam meg, azaz, hogy a lehetséges $\tilde{Q}_{init}$ érték adott részét figyelembe véve

hogyan változik a rendszer konvergencia sebessége. A kapott eredményeket a következő táblázat tartalmazza:

3. táblázat: Az η érték konvergencia sebességre gyakorolt hatása

η	Konvergencia sebesség (epizódok száma)	$\tilde{Q}_{init}$
100	23	10000
75	23	7500
60	30	6000
37	29	3700
7.5	25	750
0.015	27	1.5

A 3. táblázatban lévő futási eredményekből az látható, hogy a $\tilde{Q}_{init}$ érték egyre kisebb részét figyelembe véve egyre jobban romlik a rendszer konvergencia sebessége. A kapott adatok alapján jelen esetben (és a további alkalmazáspéldák esetében is) a $\tilde{Q}_{init}$ érték 100%-a kerül figyelembe vételre, azaz $\eta = 1$.

A helyesen megadott szakértői heurisztikával történő futtatás eredményeit (1. eset) a következő 4. táblázat tartalmazza. Ebben az esetben a rendszer átlagosan 10 epizód alatt és 124.3 szabállyal találta meg a megoldást.

4. táblázat: Helyes szakértői heurisztikával történő futtatás eredményei

Futtatási eset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Átlag**
Konvergencia sebesség	10	20	17	7	11	10	6	5	6	8	10
Szabálybázis méret	108	125	139	109	135	129	107	124	133	134	124.3

A következő 2. esetben a helyesen megadott szakértői szabályrendszer egy részével indult el a tanulási fázis. Ebben az esetben az előzőekben megadott 2. táblázatban lévő 17 szakértő szabályból kiemeltem néhány darabot véletlenszerűen, azaz pontosan a 10 darabot a 17-ből. Az így kapott szabályrendszer helyes, de kisebb méretű, mint az előzőekben megadott 2. táblázatban lévő szabályrendszer. Ebben az esetben átlagosan 14.4 epizód alatt és 114.3 szabállyal konvergált a rendszer. A pontos futási eredményeket a következő 5. táblázat tartalmazza.

5. táblázat: A helyes szakértői heurisztika egy részével történő futtatás eredményei

Futtatási eset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Átlag**
Konvergencia sebesség	20	13	10	7	7	15	29	15	22	6	14.4
Szabálybázis méret	107	85	102	85	98	96	111	107	110	98	114.3

A következő 3. esetben azt vizsgáltam, hogy milyen hatással van a rendszerre az, ha a helyesnek feltételezett szakértői szabályrendszer tartalmaz néhány „rossz” szabályt is. A rossz szabályok azt jelentik, hogy az adott szakértői szabály antecedenshez nem megfelelő konzekvens lett meghatározva. Ez azt jelentheti, hogy az ágens által végrehajtott cselekvéssorozat elromlik abban az értelemben, hogy az ágens ennek következtében nem fog eljutni a célállapotba. A 17 helyesen definiált szakértői szabályrendszerből 6 szabály konzekvensét elrontottuk úgy, hogy módosítottuk az adott akciót. Ezt a szabályrendszert a következő táblázat tartalmazza, amelyben csak az elrontott szabályokat tüntettem fel, ezek sorszáma a következő: 1, 2, 3, 15, 16 és 17.

6. táblázat: A részben helyes szakértői szabályrendszer helytelen szabályai

R#	**1**	**2**	**3**	**4…14**	**15**	**16**	**17**
s1	-0.5	-0.475	0.475	…	-0.68	-1.09	0.14
s2	0	-0.014	-0.014	…	0.042	0.042	-0.014
a	0	1	-1	…	0	0	1

Az így kapott futtatási eredményeket a 7. táblázat tartalmazza, ebben az esetben átlagosan 11.7 epizóddal és 120.1 szabályszámmal konvergált a rendszer:

7. táblázat: Helyes szakértői heurisztika egy részével történő futtatás eredményei

Futtatási eset	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Átlag**
Konvergencia sebesség	8	16	8	13	7	16	10	15	16	7	11.7
Szabálybázis méret	115	134	126	133	135	126	123	135	147	127	120.1

Az utolsó (4.) futtatási eset mikor „véletlenszerűen” generált szakértői heurisztikával indul a rendszer. Ebben az esetben szintén 17 szabályból áll a szakértői szabályrendszer, de ezek véletlenszerű állapotokkal (antecedenssel) és akcióval (konzekvenssel) rendelkeznek. Ezen állapot- és akció értékek véletlenszerűen lettek létrehozva, adott tartományon belül. Ezen szabályokat a 8. táblázat tartalmazza:

8. táblázat: A „véletlenszerűen” generált helytelen szakértői heurisztika

R#	1	2	3	4	5	6	7	8	9
s1	-0.475	-0.5	-0.475	-0.475	-0.27	-0.27	-0.27	-0.475	-0.475
s2	0	0	-0.014	0.014	0	-0.014	0	-0.042	0
a	1	-1	-1	0	-1	0	-1	1	1

R#	10	11	12	13	14	15	16	17
s1	-0.475	-0.065	0.14	-0.27	-0.885	0.885	-0.065	-1.09
s2	0	0	-0.014	-0.042	0.042	0.042	0.042	0.042
a	-1	0	1	-1	-1	1	0	-1

Ezen szabályrendszerrel történő futtatás eredményeit a 9. táblázat tartalmazza.

9. táblázat: A „véletlenszerűen” generált szakértői heurisztikával történő futtatás eredményei

Futtatási eset	1	2	3	4	5	6	7	8	9	10	Átlag
Konvergencia sebesség	29	56	19	16	24	18	37	29	20	17	26.6
Szabálybázis méret	122	127	118	124	131	120	130	124	127	121	124.4

Összegzésként az egyes futtatási esetek eredményeit a 10. táblázat tartalmazza:

10. táblázat: Az egyes futtatási estek eredményei összegezve

#	Szakértői heurisztika típusa	Átlagos konvergencia sebesség	Átlagos szabályszám
0.	üres (heurisztika nélkül)	28.3	91.7
1.	helyesen megadott	10	124.3
2.	helyesen megadottnak egy része	14.4	114.3
3.	részben helytelenül megadott	11.7	120.1
4.	véletlenszerűen generált	26.6	124.4

A kapott eredmények alapján megállapítható, hogy a 4.1.1-4.1.4. alfejezetekben bemutatott fejlesztett módszerek által külső szakértői tudásbázis fuzzy produkciós szabályok formájában injektálható a FRIQ-learning rendszerbe, továbbá egy helyesen megadott, a rendszer szempontjából külső szakértői tudásbázis nagymértékben javíthatja a FRIQ-learning rendszer hatékonyságát, pozitív módon befolyásolja a tanulási fázis konvergencia sebességét.

4.1.6 „Cart-Pole” mintapélda szakértői tudásbázis injektálásával

A „Cart-Pole” vagy másnéven a fordított inga szintén egy klasszikus megerősítéses tanulási probléma. Ebben az esetben az ágens egy autó melynek célja, hogy a közepén elhelyezkedő rudat megtanulja függőleges pozícióban tartani. A probléma 4 állapotleíróval és 1 akcióváltozóval rendelkezik, melyek a következők:

- autó aktuális vízszintes pozíciója: s_1
- autó aktuális sebessége: s_2
- inga aktuális pozíciója (szög): s_3
- inga szögsebessége: s_4
- autó elmozdulása adott erővel: a (jobbra, balra vagy nincs elmozdulás)

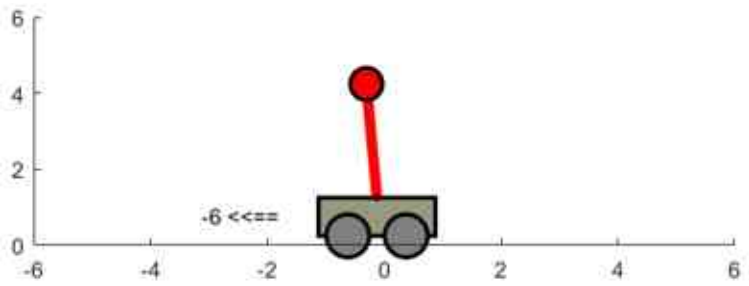
A rendszer jutalmat (r) ad, ha az autó pozíciója illetve az inga szögpozíciója adott határokon belül van, ellenkező esetben (vagy ha az inga eldől) pedig büntetést:

Jutalomfüggvény: Cart-Pole

```
Bemenet: s állapot
Kimenet: r megerősítés

x = s_1;
theta = s_3;
theta_dot = s_4;
r = 10 - 10*abs(10*theta)^2 - 5*abs(x) - 10*theta_dot;
fourtyfive_degrees = deg2rad(45);

if (x < -4.0 || x > 4.0  || theta < -fourtyfive_degrees || theta > fourtyfive_degrees)
    r = -10000 - 50*abs(x) - 100*abs(theta);
end

return r;
```

2. jutalomfüggvény: A "Cart-Pole" probléma jutalomfüggvényének pszeudokódja

A következő ábrán a „Cart-Pole” Matlab szimulációja látható, melyet diszkrét környezetre Jose Antonio Martin H. implementált [43]:



9. ábra: A „Cart-Pole" probléma szimulációja [43]

A szakértői tudásbázis hatásának vizsgálatához 4 különböző futtatási esetet hoztam létre 4 különböző kezdeti szakértői szabályrendszerrel, melyek a következők:

1. szakértői szabályrendszer nélkül („üres” tudásbázis)
2. helyes szakértői szabályrendszer
3. részben helyes szakértői szabályrendszer
4. teljesen helytelen szakértői szabályrendszer

A rendszer futtatási paraméterei és azok értékei a következők:

- szakértő által megadott megerősítés értéke: $g_{max} = 100$
- tanulási ráta: $\alpha = 0.3$
- leszámítolási tényező: $\gamma = 0.99$

Az első esetben a rendszer szakértői tudásbázis nélkül, azaz üres tudásbázisból indulva kezdte el működtetni a tanulási fázist. Ebben az esetben a FRIQ-learning módszer 58 epizód alatt 182 szabállyal találta meg a végső megoldást, amely által az autó úgy változtatja pozícióját, hogy az inga folyamatosan (minimum 1000 lépésen keresztül) függőleges pozícióban marad.

A második esetben a helyes szakértői tudásbázist úgy hoztam létre, hogy alkalmaztam az első esetben létrejött 182 szabállyal rendelkező tudásbázisra a 3.3.1 alfejezetben bemutatott III. szabálybázis redukálási módszert. A redukálási módszer alkalmazását követően hét szabály maradt a tudásbázisban, melyek a következők:

11. táblázat: A helyes szakértői szabályrendszer szabályai

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**
s_1	1	1	1	1	-1	-1	1
s_2	0	0	0	0	0	1	0
s_3	0	-0.0524	0	-0.0524	-0.2094	-0.2094	0.2094
s_4	1	-1	-1	1	-1	-1	1
a	1	-1	-0.8	0.8	1	-0.8	0.4

A helyes szakértői szabályrendszert injektálva a rendszerbe a javasolt módszer által, majd ezzel elindítva a tanulási fázist a rendszer öt epizód alatt és 46 szabállyal találta meg a megoldást. A 11. táblázat szabályai teljes egészében leírják a probléma megoldását, a további 39 szabály a rendszer szabálybázis építési módszere következtében került hozzáadásra a szabálybázishoz.

A részben helyes szakértői szabályrendszer szabályai a második futtatási esetben a helyes szakértői szabályrendszer szabályainak elrontásával jöttek létre. Ebben az esetben három darab

szabály (#1, #2 és a #6) akcióját (konzekvensét) módosítottam egy teljesen más értékre, mint amely a 11. táblázatban látható, ezt a következő táblázat szemlélteti:

12. táblázat: A részben helyes szakértői szabályrendszer szabályai

R#	1	2	3	4	5	6	7
s_1	1	1	1	1	-1	-1	1
s_2	0	0	0	0	0	1	0
s_3	0	-0.0524	0	-0.0524	-0.2094	-0.2094	0.2094
s_4	1	-1	-1	1	-1	-1	1
a	-1	0.8	-0.8	0.8	1	1	0.4

Ebben az esetben a rendszer 65 epizód alatt és 263 szabállyal találta meg a megoldást, tehát több epizód alatt és több szabállyal, mint az előző esetben. Ennek oka, hogy a helytelen szakértői szabályokat a rendszer úgy próbálja korrigálni, hogy számos új szabályt vesz fel, amely kioltja a helytelen szabályok hatását.

Az utolsó negyedik esetben teljesen helytelen szakértői szabályrendszerrel futott a tanulási fázis. A teljesen helytelen szabályrendszert úgy hoztam létre, hogy módosítottam a 11. táblázat összes szabályának akcióját. Az így létrejött szabályrendszer a következő:

13. táblázat: A teljesen helytelen szakértői szabályrendszer szabályai

R#	1	2	3	4	5	6	7
s_1	1	1	1	1	-1	-1	1
s_2	0	0	0	0	0	1	0
s_3	0	-0.0524	0	-0.0524	-0.2094	-0.2094	0.2094
s_4	1	-1	-1	1	-1	-1	1
a	-1	1	-1	-1	0.8	0.4	1

Ebben az esetben a rendszer nem konvergált a megoldáshoz, 350 epizód eltelte után 400 feletti szabályszámmal rendelkezett a tanulási fázis, tehát a helytelen szakértői szabályrendszer jelentősen rontotta a rendszer hatékonyságát.

Az egyes futási esetekben kapott eredményeket a következő táblázat foglalja össze:

14. táblázat: Futási eredmények

#	Szakértői heurisztika típusa	Konvergencia (epizód szám)	Szabálybázis méret
1.	üres	58	182
2.	helyes	5	46
3.	részben helyes	65	263
4.	teljesen helytelen	nem konvergál	>400

A „Cart-Pole” mintapélda által kapott eredmények alapján is elmondható, hogy egy helyesen definiált szakértői szabályrendszer következtében a rendszer konvergencia sebessége javítható, ellenben ha nem megfelelő információt hordozó külső tudásbázis kerül beágyazásba akkor az negatív hatással van a FRIQ-learning módszer tanulási folyamatára [S6]. Bebizonyosodott továbbá, hogy az általam javasolt, 4.1.1-4.1.4. alfejezetekben bemutatott módszerek által külső szakértői heurisztika fuzzy produkciós szabályok formájában megfelelő módon injektálható a FRIQ-learning rendszerbe.

4.1.7 I. tézis

A FRIQ-learning megerősítéses tanulási rendszer konvergencia sebessége javítható a kezdeti Q-érték szabálybázisba illesztett helyes szakértői produkciós szabályokból képzett Q fuzzy szabályokkal, ahol ezen beillesztett szabályok kezdeti konzekvens Q-értéke a környezet által adható maximális megerősítés érték alapján becsülhető.

I.1. Altézis: *A konvergencia sebesség az esetben is javulhat, ha a felhasznált helyes szakértői produkciós szabályok csak részben fedik le a teljes állapotteret.*

I.2. Altézis: *Amennyiben a felhasznált szakértői produkciós szabályok helytelen szabályokat is tartalmaznak, azaz egyes szabályok esetén az érintett állapotban javasolt akció választása csökkentené a várható megerősítés értékét, a teljes FRIQ-learning rendszer konvergencia sebessége romolhat.*

Az I. tézishez kapcsolódó saját publikációk: [S2], [S4], [S6], [S7], [S15]

4.2 A FRI Q-FÜGGVÉNYT LEÍRÓ SZABÁLYBÁZIS HANGOLÁSA

A 3.3.1 fejezetben bemutatott FRIQ-learning módszer a tudásbázist leíró fuzzy szabályok konzekvensét (azaz a Q-értékét) hangolja a (20) frissítési formula alapján, azonban az újonnan felvett szabályok antecedens része változatlan marad a teljes tanulási folyamat során. Abban az esetben, ha a rendszer vesz fel új szabályokat a szabálybázisba, akkor az újonnan létrehozott szabály állapot-akció antecedense (szabálypont) az állapot-akció tér rácsháló [94] az aktuális állapot-akció értékéhez legközelebbi pontjába kerül.

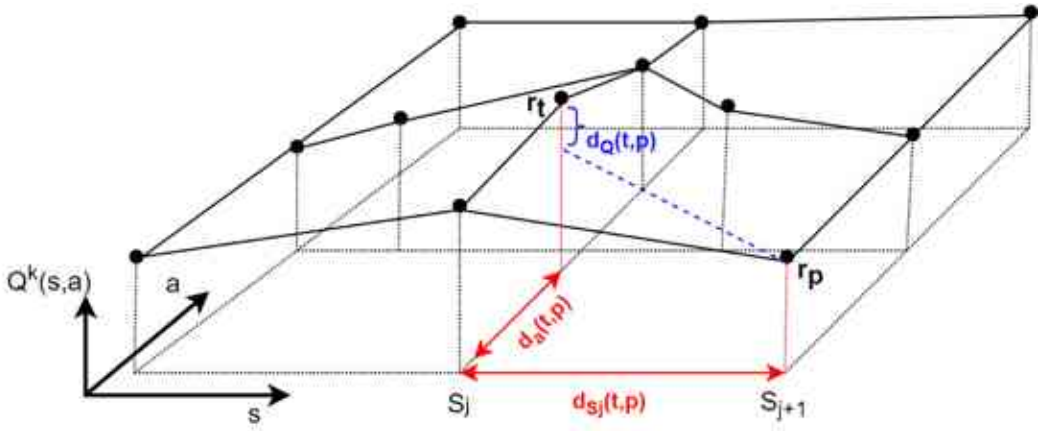
A javasolt szakértői tudásbázissal bővített rendszer esetében az eredetileg alkalmazott állapot-akció tér rácsháló [94] elhagyásra került, ezért az újonnan létrehozott szabályok antecedense pontosan az aktuális megfigyelés állapot-akció pontjába kerül. Ennek következtében a szabályok nem a rácsháló által meghatározott pontokban, hanem tetszőleges állapot-akció pontokban helyezkedhetnek el. A szakértő által definiált a priori szabályrendszer esetében, amennyiben a megadott szakértői produkciós szabály valamelyik állapothoz nem megfelelő akció értéket rendel (azaz a szakértő szabályrendszer csak részben tekinthető helyesnek), úgy a szabálybázisba felvett szakértői szabály állapot-akció antecedense is rossz helyre kerül.

Ebben az esetben, a csak részben helyes szabályrendszer negatív hatással lehet a tanulási folyamat hatékonyságára [S7], így szükség lehet egy hangolási eljárásra, amely képes a szabályok állapot-akció pontját, azaz antecedensét elhangolni (optimalizálni) a megfelelő irányba, szabálypontba.

A fentiek alapján a javasolt szakértői tudásbázissal bővített FRIQ-learning (HFRIQ-learning) módszer az alábbi főbb lépésekből áll:

- A tanulási fázis, azaz a szabálybázis létrehozási folyamat során a rendszer kezdeti (sarokponti és szakértői) szabálybázisa kiegészül a rendszer által felvett új szabályokkal.
- Ha az éppen vizsgált állapot-akció (megfigyelés) pontban még nem létezik szabály és a legközelebbi szabály is távolinak számít, akkor a rendszer felvesz ebbe a pozícióba egy új szabályt, amely állapot-akció pontja megegyezik az éppen aktuális megfigyelés állapot-akció pontjával.
- Új szabály felvétele a szabályok közötti közelségmérték és egy megengedett minimális szabálytávolság meghatározása alapján (amely által két szabály egymáshoz közelinek tekinthető).

- Új szabály felvétele esetében a szabályok közötti távolság számítása antecedens dimenziónként. Két szabály akkor tekinthető közelinek, ha minden egyes antecedens univerzumban közeliek [S9]. A következő 10. ábra két szabály (r_t, r_p) közötti távolságot szemlélteti az antecedens $(d_{S_j}(t,p),\ d_a(t,p))$ és a konzekvens $(d_Q(t,p))$ dimenziókban:

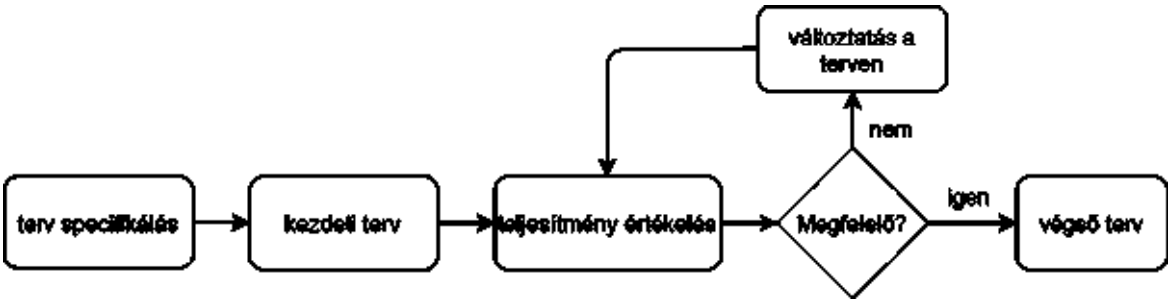


10. ábra: Az r_t és az r_p fuzzy szabályok közötti távolság dimenziónként

- Ha az éppen vizsgált állapot-akció pontban, vagy ahhoz közel már van létező szabály, akkor a szabálybázis szabályai hangolódnak, azaz a szabálypontok vándorolnak (gradiens alapú optimalizációs módszer esetén a Q-függvény gradiensének megfelelően).
- Ha a hangolási folyamat során két szabály közel kerül egymáshoz a szabályvándorlás következtében, akkor ezen szabályok egyetlen kardinális szabályként egyesülnek (csökkentve a szabálybázis méretét már a tanulási fázis során) [S5].

4.2.1 Elterjedtebb hangolási módszerek a megerősítéses tanulásban

Az optimalizálási módszerek célja változók olyan értékeinek megtalálása, amelyek minimalizálnak (vagy maximalizálnak) egy célfüggvényt. Ezen algoritmusok az adott változók értékeit addig módosítják (hangolják) amíg a valamilyen szempontból optimális megoldás elő nem áll. A következő ábra az általános optimalizálási folyamatot szemlélteti:

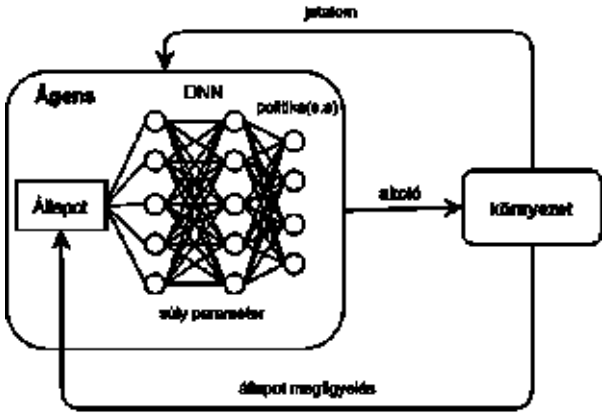


11. ábra: Általános optimalizálási folyamat [48]

Az optimalizálási folyamat általában egy függvény minimum vagy maximum pontjának (szélsőértékének) keresését jelenti. A legtöbb esetben egy hibafüggvény minimum pontjának keresése a cél, azaz a változók azon értékeinek a megtalálása, amely mellett a függvény értéke a legkisebb majd ez által úgy igyekszik elmozdítani (elhangolni) az adatpontokat, hogy a hiba értéke csökkenjen.

Az iteratív módszerek közül a gradiens módszer (*gradient descent, GD – gradiens süllyedés*) a legelterjedtebb, amely iterációkon keresztül jut el egy függvény minimumához, a függvény gradiense, azaz az iránymenti deriváltjai által úgy, hogy mindig a legmeredekebb lejtő irányba halad. Gradiens módszer alapú hangolási eljárást a mély tanulás [62] (*Deep Learning*) alapú „Deep Q-learning Network" (*DQN*) [92] is alkalmaz. A DQN egy olyan Q-learning algoritmus, amely a mély tanulás, azon belül is a mély megerősítéses tanulás (*Deep Reinforcement Learning*) [3][29][63] módszerek csoportjába sorolható. A mély megerősítéses tanulás a megerősítéses tanulás és a mély tanulás kombinációja. Ezen módszerek közös jellemzője, hogy általában nagyszámú bemeneti adattal dolgoznak és többrétegű neurális hálózatokat (*deep neural network – DNN*) alkalmaznak, melyek topológiájában több rejtett réteg is található (innen a „mély" elnevezés). Minden egyes réteg egy másfajta reprezentációját adja a bementi adatoknak, amely egyfajta jellemzőkinyerésnek tekinthető, így rétegről-rétegre előre haladva egyre komplexebb összefüggések kerülnek beazonosításra (például: pixelek → élek → alakzatok → tárgyak). Az elterjedtebb mély megerősítéses tanulási módszerekről bővebb áttekintést a [3][29] és [63] hivatkozások adnak. Az optimalizálási módszereket a Deep Learning algoritmusok esetében általában a neurális hálózat súlyainak optimalizálására alkalmazzák, amely által az úgy igyekszik módosítani a hálózat súlyainak értékét, hogy a hiba értéke csökkenjen.

A mély megerősítéses tanulás modelljét a következő ábra szemlélteti [62]:



12. ábra: A mély megerősítéses tanulás modellje [62]

A DQN módszer neurális hálózatot alkalmaz a Q-függvény leírására, amely által a Q-függvény a következő módon írható fel [20]:

$$Q(s,a) \approx Q(s,a,\theta) \tag{36}$$

ahol θ a neurális hálózat súlyait reprezentálja. A hiba értéke ebben az esetben a TD-hiba értékének feleltethető meg, amelyet az adott optimalizálási módszer minimalizálni igyekszik [20]:

$$\mathrm{L} = E\left[\left(r_k + \gamma \max_a Q(s_{k+1}, a_{k+1}, \theta) - Q(s_k, a_k, \theta)\right)^2\right] \tag{37}$$

A klasszikus gradiens módszernek több változata is található a szakirodalomban, ilyen például a sztochasztikus gradiens módszer (*Stochastic Gradient Descent - SGD*), amely a klasszikus gradiens módszerrel ellentétben nem az összes lehetséges hibafüggvény pontban határozza meg a gradienst és ez által az összes rendelkezésre álló minta alapján hangol minden egyes iterációban, hanem egyesével, véletlenszerűen veszi a mintapontokat iterációnként. Ennek előnye, hogy nagy dimenziószámmal rendelkező problémák esetében csökken a számítási- és memória igény. A „mini-batch” gradiens módszer a mintaadatok halmazát kisebb részhalmazokra bontja, majd ezen részhalmazok alapján határozza meg a hibát és frissíti (azaz hangolja) a modell paramétereit. A gradiens módszer alapú optimalizálási eljárások hátránya, hogy nem garantálják a globális optimum megtalálását. A gradiens módszerek elterjedtebb változatairól és azok hatékonyságának összehasonlításáról a [34] és [82] irodalmak adnak bővebb áttekintést.

A részecske-raj alapú optimalizálás (*Particle Swarm Optimization - PSO*) [44] szintén egy iteratív algoritmus, amely működése a keresési térben, általában egyenletes eloszlás szerint elhelyezett részecskéken (raj) alapszik, amely részecskék matematikai összefüggések alapján mozognak. A részecskék a legjobb pontot keresik a térben majd a saját legjobb pozíciójuk és a raj legjobb pozíciója alapján mozdulnak el. A legjobb ismert pozíció frissül iterációnként attól függően, hogy a raj talált-e a legjobb ismert pozíciónál még jobbat vagy sem, tehát a részecskék mozgását (és a mozgás irányát) az adott iterációban megtalált legjobb pozíció befolyásolja. Ha a raj már nem talál jobb pozíciót, akkor a leállási feltétel bekövetkezése után a raj legjobbja (részecskéje) adja meg a függvény optimum, azaz minimum pontját. A PSO algoritmus esetében nincs szükség a függvény gradiensének és így annak parciális deriváltjainak meghatározásra (a gradiens módszereknél ez alapkövetelmény) így ez a módszer jól

alkalmazható olyan függvények esetében melyek gradiense nem ismert illetve zajjal terhelt, valamint időben változó problémák esetében is.

További optimalizálási módszerekről a [48] sorszámú szakirodalom, a megerősítéses tanulásban alkalmazott elterjedtebb optimalizálási módszerekről pedig a [66] hivatkozás ad bővebb áttekintést.

4.2.2 Szabálytávolság és közelségmérték meghatározása

Az új szabályok felvétele és az egymáshoz közeli szabályok egyesítése kapcsán is felmerül a szabályok közötti távolság meghatározása. Új szabály akkor kerül beszúrásra a szabálybázisba, ha a hozzá legközelebb eső szabály is távol van, azaz távolsága nagyobb, mint egy meghatározott távolságmérték, illetve két már létező szabály akkor vonható össze (redukálható) egyelten szabállyá, ha azok nagyon közel kerülnek egymáshoz.

A javasolt módszerben a szabályok közeliségének meghatározása egy antecedens dimenziónként számított $\boldsymbol{dtr} = [dtr_1, dtr_2, \dots, dtr_n, dtrU]$ távolságküszöb alapján történik, ahol $dtr_1, dtr_2, \dots, dtr_n$ az állapot dimenzióra, $dtrU$ pedig az akció dimenzióra számított távolságküszöb. Ezáltal a $\boldsymbol{dtr}$ vektor elemszáma megegyezik az antecedens dimenziók számával, azaz $(n+1)$. A távolságküszöbök számításának alapja a szintén antecedens dimenziónként meghatározott d_j távolság, ahol a j az antecedens univerzumok számát ($j \in [1, n+1]$) jelöli.

Ha a szabálybázis egy szabályának távolsága az aktuális megfigyeléstől (állapot-akció ponttól) minden egyes állapot-akció dimenzióban kisebb, mint az adott dimenziókra számított dtr_j ($j \in [1, n+1]$) távolságküszöb értéke, akkor a szabálypont közelinek tekinthető az adott megfigyeléshez [S9]:

$$\underset{t,p\in[1,m+\hat{m}]}{\exists} t,p \;\; ahol \underset{j\in[1,n+1]}{\forall} (d_j(t,p) < dtr_j) \tag{38}$$

Azaz két szabály közelinek tekinthető, ha létezik olyan t és p szabálysorszám az $m+\hat{m}$ méretű szabályrendszerben (m számosságú FRIQ szabályok + $\hat{m}$ számosságú szakértői szabályok), amire igaz, hogy minden egyes j-edik ($j \in [1, n+1]$) antecedens dimenzióban (az $n+1$ dimenziószámú állapot-akció térben) az adott szabály $d_j(t,p)$ távolsága kisebb, mint a dtr_j távolságküszöbök értéke. A $d_j(t,p)$ a t és p indexű szabályok ($t,p \in [1, m+\hat{m}]$) közötti távolságot jelöli a j-edik antecedens dimenzióban ($j \in [1, n+1]$):

$$d_j(t,p) = \left|s_j^t - s_j^p\right| \qquad j \in [1,n] \tag{39}$$

$$d_j(t,p) = |a^t - a^p| \qquad j = n+1$$

Ahol s_j^t a j-edik antecedens fuzzy halmaza a t-edik indexű szabálynak, s_j^p a j-edik antecedens fuzzy halmaza a p-edik indexű szabálynak, a^t t-edik indexű szabály akciója, a^p pedig a p-edik indexű szabály akciója, a $||$ az abszolútértéket jelöli. A t-edik és p-edik indexű szabályok közötti távolság tehát nem egyetlen számérték, hanem egy $[d_1(t,p), d_2(t,p), \dots, d_n(t,p), d_{n+1}(t,p)]$ vektor, amely elemszáma megegyezik az antecedens dimenziók számával $(n+1)$ és dimenziónként tartalmazza az adott szabályok antecedensei közötti távolságokat.

Az egymáshoz közeli szabályok meghatározásához szükség van a szabálybázis összes szabálya közötti távolság meghatározására. Ez leírható egy többdimenziós *D* távolságmátrix által, amely a szabálybázis szabályai közötti távolságokat tartalmazza minden egyes antecedens dimenzióban. Mivel mindegyik szabály önmagától számított távolsága 0, így a többdimenziós mátrix főátlójában csupa nullák szereplenek illetve a (t,p) indexű szabályok közötti távolság az ugyanaz, mint a (p,t) indexűek közötti $(d_j(t,p) = d_j(p,t))$, így a számításoknál a főátló alatti elemeket (alsóháromszög mátrix) szükséges figyelembe venni.

A javasolt módszerben az akció és állapot dimenziókra számított távolságküszöb értékek az egyes antecedens univerzumok méretének közelségarányhoz viszonyított része (*distance rate* - dR), amely egy a szakértő által definiált numerikus konstans, amely az egyes antecedens univerzumokban eltérő lehet. Azaz a távolságküszöb értékek az egyes dimenziók hosszának valamekkora része. A továbbiakban ezen távolságküszöb értékek határozzák majd meg a szabályok között megengedett minimális távolságot:

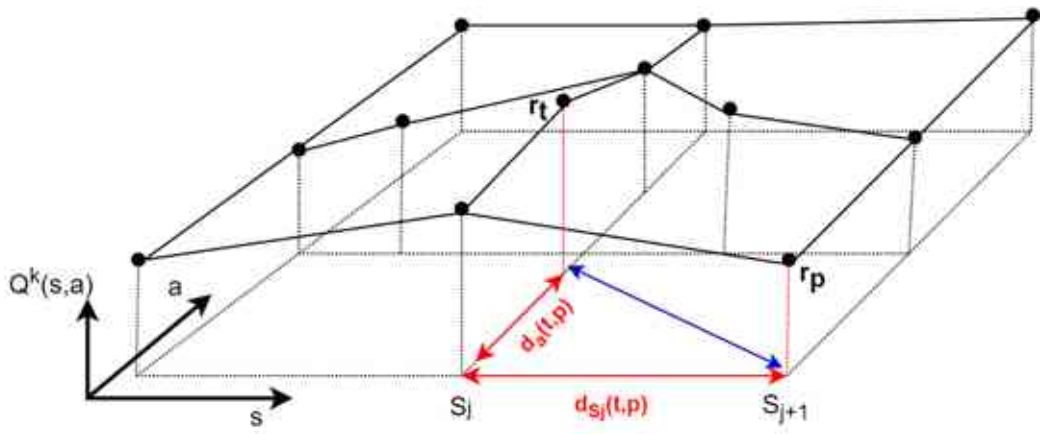
$$\begin{aligned} dtr_j &= \frac{length(S_j)}{\mathrm{dR}_S}, & j \in [1,n], \\ dtrU &= \frac{length(U)}{\mathrm{dR}_U}, & j = n+1, \end{aligned} \tag{40}$$

ahol $length(S_j)$ az állapot univerzum, $length(U)$ pedig az akció univerzum legkisebb és legnagyobb elemei közötti különbség abszolútértéke, azaz az értelmezési tartományuk hossza:

$$\begin{aligned} length(s_j) &= \left|\max(S_j) - \min(S_j)\right|, \\ length(U) &= |\max(U) - \min(U)|, \end{aligned} \tag{41}$$

ahol max (S_j) a maximum (legnagyobb), min (S_j) pedig a minimum (legkisebb) eleme a j-edik ($j \in [1, n]$) S_j állapot és U akció univerzumnak.

A 13. ábra a t és a p indexű szabályok közötti $d_j(t, p)$ antecedens (állapot és akció) dimenzióbeli távolságokat szemlélteti:



13. ábra: A p és a t indexű szabályok közötti $d_j(t, p)$ távolság az antecedens (állapot és akció) univerzumban

Az ábrán látható módon, a kék nyíllal jelölt eredő távolság helyet a piros vonalakkal jelölt távolságok kerültek meghatározása (antecedens dimenziónként).

Összefoglalva, a (38) összefüggésnek eleget tevő t és p indexű szabálypárok tekinthetők egymáshoz közeli szabályoknak. A szabályközelséget és ezáltal a közeli szabályokat meghatározó, fejlesztett algoritmus pszeudokódja az alábbi [S9]:

Algoritmus: isCloseRuleExist(observation, R, dR)

Bemenet: megfigyelés (szabály), szabálybázis, dR paraméterek
Kimenet: igaz vagy hamis (a megfigyelés közeli szabály-e vagy sem)

távolságmátrix (D) inicializálása
távolságküszöbök (dtr) számítása minden egyes antecedens dimenzióra
Ciklus (minden egyes szabályra)
    szabálytávolság $(d \in D)$ számítása az aktuális megfigyeléstől
    Ciklus (minden egyes antecedens dimenzióra)
        $If \left(\underset{t,p \in [1, m+\widehat{m}]}{\exists} t, p \ \ hogy \ \underset{j \in [1, n+1]}{\forall} \left(d_j(t, p) < dtr_j \right) \right) \quad j \in [1, n+1]$
            a megfigyelés (szabály) közeli szabálynak tekinthető
            return igaz
        else
            a megfigyelés (szabály) nem tekinthető közeli szabálynak
            return hamis
    ciklus vége

5. algoritmus: A szabályközelséget és ezáltal a közeli szabályokat meghatározó fejlesztett algoritmus

4.2.3 A gradiens módszer alkalmazása a szabályrendszer hangolására

Abban az esetben, ha az éppen aktuális megfigyelés közelében már található létező fuzzy szabály, akkor nem kerül új szabály felvételre a megfigyelés pozíciójába, hanem a szabályrendszer antecedense és konzekvense kerül hangolásra a gradiens módszer alkalmazásával.

A gradiens módszer egy iteratív optimalizációs algoritmus, amely célja egy F függvény minimum pontjának meghatározása, amelyet úgy valósít meg, hogy egy adott x_0 függvénypontból kiindulva iterációról-iterációra valamekkora α lépést megtéve halad a legmeredekebb lejtő irányába, amely irányt a függvény változói szerinti deriváltjai (iránymenti deriváltjai) határozzák meg. A módszer alkalmazhatóságának feltétele (a gradiens számítása következtében) az adott függvény differenciálhatósága. Többváltozós függvény esetében minden egyes változóra szükséges a parciális deriváltak meghatározása. A parciális deriváltakból képzett ∇ vektor a gradiens vektor, amely egy adott függvénypontban megmutatja, hogy merre növekszik a függvény a leginkább. Az F függvény minimum pontjának keresése során egy x_k pontból kiindulva az iteráció rákövetkező x_{k+1} értéke úgy áll elő, hogy az egy α tanulási ráta által súlyozott mértékben a $\nabla F(x_k)$ gradiens által meghatározott növekedési iránnyal ellentétesen változik:

$$x_{k+1} = x_k - \nabla F(x_k) * \alpha \tag{42}$$

A klasszikus gradiens módszerek esetében az α tanulási ráta értéke minden iterációban konstans, de léteznek olyan megoldások is melyeknél ez a paraméter iterációként változik, ilyen például az AdaGrad (*Adaptive Gradient*) [27].

A Q-függvény hangolása során a cél a Q-függvényt leíró fuzzy szabálypontok hangolása úgy, hogy az egyes iterációs lépésekben a TD-hiba értékével frissülő Q-függvényt minél kisebb hibával írja le.

A hiba a legkisebb négyzetek módszerével az elvárt kimeneti értékek és a tényleges kimeneti értékek különbség négyzeteinek az összege, azaz az átlagos négyzetes hiba (*Mean Squared Error - MSE*):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - F(x_i)\right)^2 \tag{43}$$

ahol y_i az elvárt kimenet leíró a i-edik ($i \in [1, N]$) mintaadat, $F(x_i)$ az F függvény értéke az x_i pontban, N pedig a mintadatok száma.

Jelen esetben hibának a Q-learning TD-hiba értéke tekinthető, amely a jutalom értéke összegezve a diszkontált várható Q-érték és a tényleges Q-érték közötti eltéréssel. A (43) formulában így y_i -nek a $g(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma * \max_{a' \epsilon U} \tilde{Q}^k(\boldsymbol{s}', a')$, $F(x_i)$ -nek pedig a $\tilde{Q}^k(\boldsymbol{s}, a)$ feleltethető meg:

$$TDerror = g(\boldsymbol{s}, a, \boldsymbol{s}') + \gamma * \max_{a' \epsilon U} \tilde{Q}^k(\boldsymbol{s}', a') - \tilde{Q}^k(\boldsymbol{s}, a) \tag{44}$$

A fentiek alapján a gradiens módszerben alkalmazott MSE értéke (melynek a minimalizálása a cél) az alábbi módon határozható meg:

$$MSE = \frac{1}{\hat{m}} \sum_{i=1}^{\hat{m}} (TDerror)^2 \tag{45}$$

A fuzzy szabályok antecedens és konzekvens értékeinek hangolása a (42) összefüggés alapján történik, ahol az $F(x_k)$ függvény parciális deriváltja ($\nabla F(x_k)$ gradiens) a láncszabály alkalmazásával a következőképpen határozható meg:

$$\nabla F(x_k) = \frac{\partial MSE(x_k)}{\partial x_k} = \frac{\partial (TDerror)^2}{\partial x_k} = 2 * TDerror * \frac{\partial \tilde{Q}^k(\boldsymbol{s}, a)}{\partial x_k} \tag{46}$$

A (42) összefüggésbe $\nabla F(x_k)$-t a (46) szerint behelyettesítve az x_{k+1} értéke a következő lesz:

$$x_{k+1} = x_k - \left(2 * TDerror * \frac{\partial \tilde{Q}^k(\boldsymbol{s}, a)}{\partial x_k} \right) * \alpha \tag{47}$$

A frissítési szabályt a $\tilde{Q}^k(\boldsymbol{s}, a)$ függvény minden egyes $\boldsymbol{s}, a$ és q dimenziójának parciális deriváltjaira alkalmazva az új $\boldsymbol{s}_{k+1}$ állapot, az új a_{k+1} akció és az új q_{k+1} konzekvens értékek a következő módon számíthatók:

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k - \left(2 * TDerror * \frac{\partial \tilde{Q}(\boldsymbol{s}, a)}{\partial \boldsymbol{s}} \right) * \alpha \tag{48}$$

$$a_{k+1} = a_k - \left(2 * TDerror * \frac{\partial \tilde{Q}(\boldsymbol{s}, a)}{\partial a} \right) * \alpha \tag{49}$$

$$q_{k+1} = q_k - \left(2 * TDerror * \frac{\partial \tilde{Q}(\boldsymbol{s}, a)}{\partial q} \right) * \alpha \tag{50}$$

A javasolt algoritmus pszeudokódja az alábbi [S10][S12]:

Algoritmus: updateRBGradDesc(R, numOfIter, alpha)

Bemenet: szabálybázis (vagy egy szabály), iterációk száma, tanulási ráta
Kimenet: hangolt szabálybázis (vagy egy szabály)

x súlyok inicializálása a szabályok antecedense és konzekvenseként
Ismétlés
    $\nabla F(x_k)$ gradiens számolása $\boldsymbol{s}, a, q$ szerint
    x súly frissítése: $x_{k+1} = x_k - \left(2 * TDerror * \frac{\partial \tilde{Q}^k(\boldsymbol{s},a)}{\partial x_k}\right) * \alpha$
    leállási feltétel változók értékeinek frissítése
amíg a hiba már nem csökken vagy az iterációk száma véget nem ér
return hangolt szabálybázis (vagy egy szabály)

6. algoritmus: A gradiens módszer alapú szabálybázis hangolás javasolt algoritmusa

A $\tilde{Q}$-függvény $\boldsymbol{s}$ állapot szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial \boldsymbol{s}}$, a akció szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial a}$ és q konzekvens szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial q}$ parciális deriváltjainak meghatározása a következő 4.2.4 alfejezetben történik.

4.2.4 Az FRI Q-függvény parciális deriváltjainak meghatározása

A szabálybázis antecedensének és konzekvensének hangolásához a (47) összefüggésben szereplő többváltozós $\tilde{Q}(\boldsymbol{s}, a)$ FRI Q-függvény parciális deriváltjainak, azaz a $\nabla\tilde{Q}$ gradiens vektornak a meghatározása szükséges. A függvény változói az $\boldsymbol{s}$,a és q, ahol $\boldsymbol{s}$ az n-dimenziós $(s_1, s_2, \dots, s_n)$ állapot univerzum, a az egydimenziós akcióváltozó, q pedig a szintén egydimenziós konzekvens univerzum. A parciális deriváltakat tartalmazó $\nabla\tilde{Q}$ gradiens vektor a következő formában írható fel:

$$\nabla\tilde{Q} = grad(\tilde{Q}) = \begin{bmatrix} \frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial \boldsymbol{s}} \\ \frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial a} \\ \frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial q} \end{bmatrix} \tag{51}$$

A $\tilde{Q}(\boldsymbol{s}, a)$ függvény matematikai modellje a (23) formulával definiálható, így ennek következtében a parciális deriváltak analitikus módon (differenciálkalkulus szabályait alkalmazva) meghatározhatók. A $\tilde{Q}(\boldsymbol{s}, a)$ függvényt leíró (23) összefüggésben a δ_v^i súlyozott távolság a (24) formula által meghatározott, ahol $v_j(s_j)$ a j-edik ($j \in [1, n]$) állapot univerzum skálafüggvénye, a $v(a)$ pedig az akció univerzum skálafüggvénye. A probléma (és így a deriválás) egyszerűsítése végett tekintsük a $v_j(s_j)$ és a $v(a)$ skálafüggvényeket konstans

függvényeknek és jelöljük őket c_j-vel illetve c_a-val, amelyek következtében a (24) formula által meghatározott távolság képlete a következőképpen módosul:

$$\delta_v^i = \delta_v\left((\boldsymbol{s}, a), (\boldsymbol{s^i}, a^i)\right) = \left[\sum_{j=1}^{n}\left(\int_{s_j^i}^{s_j} c_j\, dx_j\right)^2 + \left(\int_{a^i}^{a} c_a dx_a\right)^2\right]^{1/2} \tag{52}$$

ahol $(\boldsymbol{s}, a)$ az állapot-akció megfigyelés, $(\boldsymbol{s^i}, a^i)$ az i-edik szabály állapot-akció antecedense, s_j a j-edik $(j \in [1, n])$ dimenziója az n-dimenziós állapottér univerzumnak, s_j^i az i-edik szabály j-edik állapot dimenziója, a^i az i-edik szabály akció univerzuma, c_j az s_j állapot univerzum konstans skálafüggvénye, c_a pedig az U akció univerzum konstans skálafüggvénye.

Abban az esetben, mikor a skálafüggvények konstans függvényeknek tekintettek, az $\boldsymbol{s}$ állapot szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial \boldsymbol{s}}$, az a akció szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial a}$ és q konzekvens szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial q}$ parciális deriváltak a következő (53), (55) és (56) összefüggések által határozhatók meg. A q konzekvens szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial q}$ parciális derivált a következő:

$$\frac{\partial \tilde{Q}(\boldsymbol{s}, a)}{\partial q} = \begin{cases} 1 & \text{ha } (\boldsymbol{s}, a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re} \\ \dfrac{1}{\left(\delta_v^i\right)^\lambda} \Big/ \left(\displaystyle\sum_{i=1}^{m} \frac{1}{\left(\delta_v^i\right)^\lambda}\right) & \text{egyébként} \end{cases} \tag{53}$$

ahol δ_v^i az (52) formulával meghatározott távolság, m a szabályok száma $(i \in [1, m])$. Ha $(\boldsymbol{s}, a) = (\boldsymbol{s}^i, a^i)$ akkor a megfigyelés éppen szabálypontra esik, ebben az esetben q deriváltja q szerint 1 lesz.

Az állapot és akció szerinti parciális deriváltak összefüggéseinek jobb olvashatósága érdekében vezessünk be egy új változót, legyen x a következő:

$$x = \sum_{i=1}^{m} \frac{q}{\left(\sum_{i=1}^{m} \frac{1}{\delta_v^i(s,a)}\right)\delta_v^i(s,a)} * \left(\frac{q\left(\sum_{i=1}^{m} \frac{1}{\delta_v^i(s,a)}\right)\delta_v^i(s,a)}{\left(\sum_{i=1}^{m} \frac{1}{\delta_v^i(s,a)}^2\right)\delta_v^i(s,a)^3} - \frac{q\delta_v^i(s,a)}{\left(\sum_{i=1}^{m} \frac{1}{\delta_v^i(s,a)}\right)\delta_v^i(s,a)^2}\right) \tag{54}$$

ahol δ_v^i a (52) formulával meghatározott távolság, m pedig a szabályok száma $(i \in [1, m])$. Az (54) alkalmazásával az $\boldsymbol{s}$ állapotuniverzum szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial \boldsymbol{s}}$ és az a akcióuniverzum szerinti $\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial a}$ parciális deriváltak a következőképpen számíthatók:

$$\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial \boldsymbol{s}} = \begin{cases} 0 & \text{ha } (\boldsymbol{s},a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re,} \\ \sum_{j=1}^{n}(x_j) & \text{egyébként} \end{cases} \quad (55)$$

$$\frac{\partial \tilde{Q}(\boldsymbol{s},a)}{\partial a} = \begin{cases} 0 & \text{ha } (\boldsymbol{s},a) = (\boldsymbol{s}^i, a^i) \text{ valamennyi } i\text{-re,} \\ x & \text{egyébként} \end{cases} \quad (56)$$

Ahol x az (54) összefüggés által definiált, n pedig az állapotdimenziók száma ($j \in [1,n]$). Ha $(\boldsymbol{s},a) = (\boldsymbol{s}^i, a^i)$ akkor a megfigyelés éppen szabálypontra esik, ebben az esetben q deriváltja $\boldsymbol{s}$ illetve a szerint 0 lesz. Az (55) és az (56) összefügésekből az látszik, hogy az állapot és az akció szerinti parciális deriváltak megegyeznek azzal a különbséggel, hogy az $\boldsymbol{s}$ állapot n-dimenziós, így a parciális deriváltak számítása ebben az esetben $\boldsymbol{s}$ minden egyes j-edik ($j \in [1,n]$) univerzumára szükséges.

4.2.5 A hangolandó szabálypontok meghatározása

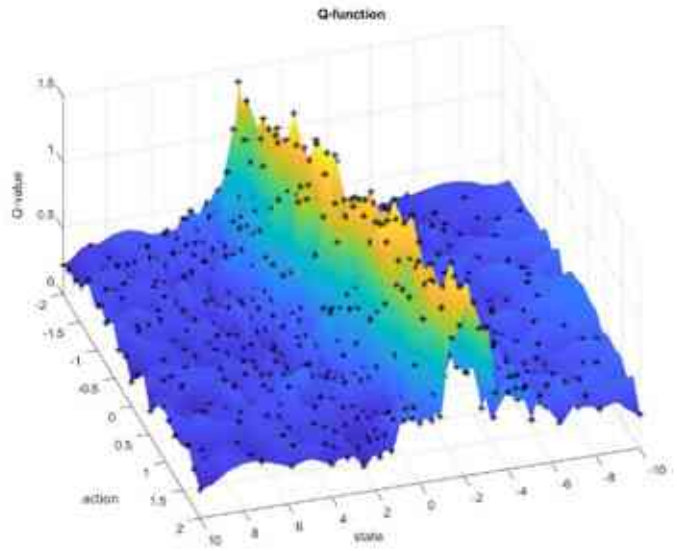
Az FRIQ-learning Q-függvény tartópontjainak hangolása (optimalizálása) során meghatározandó, hogy mely szabálypontok kerüljenek optimalizálásra a tanulási fázis során. A javasolt módszerben a Q-függvény a fuzzy szabálypontok 2.2.1 fejezetben bemutatott „FIVE" FRI módszer fuzzy interpolációjával áll elő.

A gradiens módszernek több verziója ismert a 4.2.1 fejezetben bemutatottak alapján. A GD módszer minden egyes iterációban minden egyes mintapontot figyelembe vesz a hibafüggvény illetve a deriváltak számításakor, míg a SGD módszer egyesével (véletlenszerűen) veszi a mintapontokat majd ezek alapján határozza meg a gradienst és hibafüggvényt iterációnként. Jelen esetben mintapontoknak a szabálypontok tekinthetők, melyek hangolása a TD-hiba (44) értékének függvényében történik. A szabálypontok a Q-függvény frissítésekor a ∇F hibafüggvény parciális deriváltjai (46) szerint változnak.

A tanulási folyamat során a szabálypontok abban az esetben hangolódnak, ha a Q-frissítés értéke nagynak tekinthető $(\Delta\tilde{Q} > \varepsilon_Q)$ és az éppen aktuális állapot-akció megfigyelés közelében (a 4.2.2 alfejezetben meghatározott közelségmérték alapján) található már létező szabály.

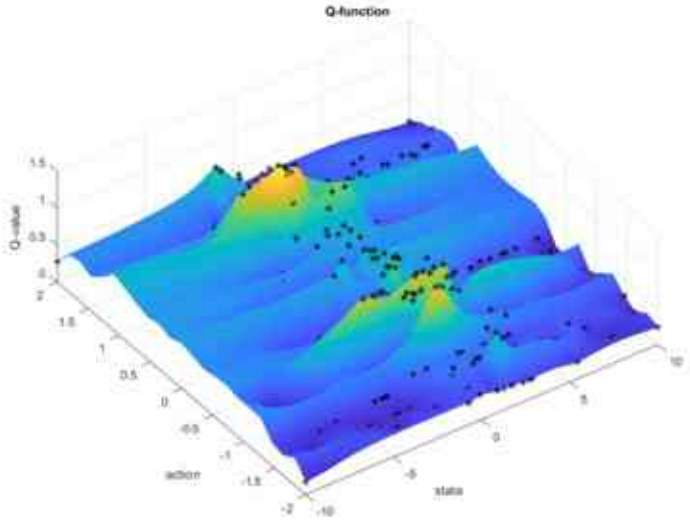
Annak vizsgálata, hogy az összes szabálypont antecedens és konzekvens értékeinek hangolása a tanulási folyamat során milyen hatással van a Q-függvényre egy egyszerű, egy állapot- és egy akciódimenzióval rendelkező mintapéldán történik. A mintapélda (illetve a futás

során alkalmazott paraméterek) és az így kapott futási eredmények a disszertáció későbbi 4.5.1 fejezetében lesznek részletesen bemutatva, itt csak az egyes esetekben a kapott felületek kerülnek összehasonlításra. A 14. ábrán az összehasonlítás alapjául szolgáló, a javasolt gradiens módszeren alapuló hangolási eljárás alkalmazása nélkül kapott „referencia” Q-függvény felülete látható, ahol a „*” (csillag) karakterek a szabálypontokat (530 darab) jelölik, a szabályok között megengedett minimális távolság pedig az univerzumok hosszának 100-ad része:



14. ábra: Az összehasonlítás alapjául szolgáló "referencia" Q-függvény felülete

Az első vizsgált esetben a javasolt hangolási módszerrel az összes szabálypont hangolásra kerül, beleértve a szakértői szabályokat is. Az ebben a futási esetben kapott Q-függvény felületét a 15. ábra szemlélteti:



15. ábra: Összes létező szabálypont hangolása esetében kapott Q-függvény felülete

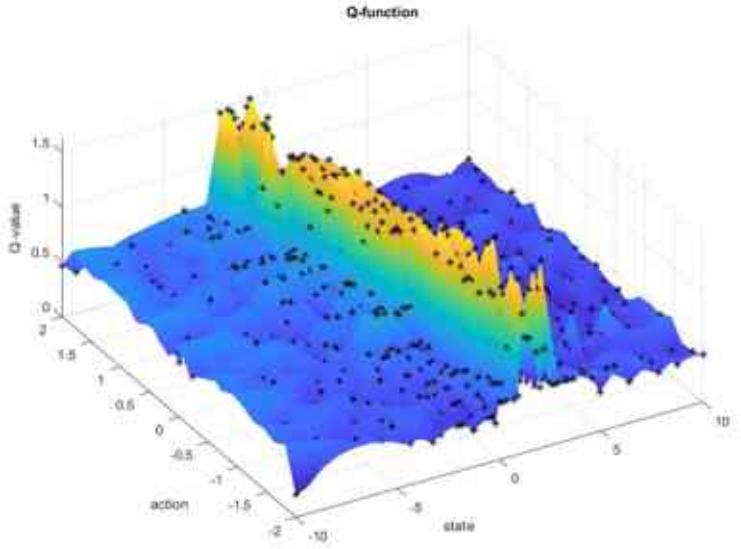
Megállapítható, hogy az összes szabálypont hangolásával a 14. ábrán látható referencia függvény felületéhez képest a függvény felülete romlott, nem alakult ki a felület „gerince”, több kisebb csúcs keletkezett csupán. Ennek oka az lehet, hogy az állapot-akció tér azon pontjainak antecedense és Q-értéke, melyek ritkán kerülnek bejárásra és így frissítésre (a felderítés-kiaknázás következtében), az összes szabálypont hangolása miatt elromlik. A kiaknázás során bejárt út frissítései hatással vannak a ritkán, csak a felderítés során tesztelt területekre. A visszajelzés nélküli (kevésbbé bejárt) területek Q-értéke a gyakorta bejárt területek átlag Q-értéke felé hangolódik. Emiatt egyes szabályok Q következmény értéke elromlik, azaz rossz irányban módosul.

Az összes szabálypont egyidejű hangolása ezért nem járható út. Megoldás lehet az, hogy csak azon szabályok kerüljenek hangolásra, amely az éppen aktuális állapot-akció megfigyelési pont közelében találhatók, azaz a legközelebb helyezkednek el ahhoz. A javasolt módszer megoldja a fentebb említett problémát és helyes irányban hangolja a Q-függvény szabálypontjait, nem változtatva azon szabályokat melyek a aktuális állapot-akció megfigyelési ponttól távol, a kevésbé látogatott területekre esnek. A javasolt módszerben a közelség mértékének meghatározása a 4.2.2 fejezetben bemutatottak alapján történik.

A hangolási folyamat során tekintettel kell lenni a hangolandó (az éppen közeli) szabálypont típusára. A sarokponti típusú szabályok antecedensei kötöttek (az állapot-akció tér univerzum hiperkocka sarokpontjai), a hangolás során nem változhatnak. A hangolás ezért a sarokponti típusú szabályok esetében csak a szabályok konzekvensére vonatkozik. A szakértői és a rendszer által felvett típusú szabályok esetén a szabályok antecedense (állapot-akció pontja) és konzekvense (Q-értéke) is hangolásra kerül.

Abban az esetben, ha a tanulási folyamat során valamely szakértői szabály antecedense kerül hangolására, akkor ezen produkciós (állapot-akció) szakértői szabály módosul. Abban az esetben, ha a hangolási folyamat során az antecedens nem, vagy csak kismértékben változik, akkor a szakértői szabály a környezeti tesztek által igazoltnak, helyesen definiált szakértői szabálynak tekinthető.

A 16. ábrán az az eset látható mikor nem az összes szabálypont, hanem csak a megfigyeléshez közeli szabályok kerültek hangolásra. Ebben az esetben a függvény felülete nem romlott el, a függvény gerince viszonylag összefüggően kialakult, a függvény formája a referencia Q-függvényhez hasonló lett, azzal a különbséggel, hogy az kisimult, a felülete simább lett (a gradiens alapú hangolásnak köszönhetően).



16. ábra: Csak a megfigyeléshez legközelebbi szabálypontok hangolása esetében kapott Q-függvény felülete

A kapott Q-függvény felületeken sok olyan szabálypont található (a „*” karakter által jelölt pozíciókban), melyek viszonylag egymáshoz közel (és sűrűn) helyezkednek el. Ezeken az állapot-akció területeken a gradiens módszer alapú hangolási eljárásnak köszönhetően a közeli szabályok egyre közelebb kerülnek egymáshoz. A javasolt hangolási módszer következő lépése az, hogy az egymáshoz viszonylag közel kerülő szabályok valamilyen módszer (a szabályok közötti távolság, illetve távolságküszöbök) alapján kerüljenek összevonásra (egyesítésre), a szabályok számának csökkentése érdekében. Ennek a lehetőségnek a vizsgálata, illetve egy ilyen lehetséges módszer kidolgozása és bemutatása a disszertáció következő 4.3 fejezetében valósul meg.

4.2.6 II. tézis

A HFRIQ-learning megerősítéses tanulási rendszer inkrementális szabálybázis építési fázisában a Q-függvényét leíró fuzzy szabálybázis szabályainak (fuzzy Q-szabályok) antecedensei és konzekvensei gradiens módszerrel optimalizálhatók, hangolhatók. Az aktuális állapot-akció pontban egy új szabály beillesztésének feltétele a már meglévő szabályoktól vett távolsága és a Q-függvény frissítésének mértéke alapján meghatározható.

II.1. Altézis: *Amennyiben nincs olyan fuzzy Q-szabály, melynek antecedense valamennyi antecedens dimenzióban vett távolsága kisebb az egyes dimenziókra meghatározott távolságküszöbnél és a Q-függvény frissítésének mértéke nagyobb, mint egy küszöbérték, úgy az aktuális állapot-akció pontba egy új szabály kerül beillesztésre. Ellenkező esetben a meglévő fuzzy Q-szabályok kerülnek hangolásra.*

II.2. Altézis: *Abban az esetben, ha szabálybázisba illesztett kezdeti szakértői szabályrendszer helytelen szakértői produkciós szabályokat is tartalmaz, akkor azok a tanulási fázisban a szabálybázis többi szabályával együtt hangolhatók, korrigálhatók.*

II.3. Altézis: *A HFRIQ-learning megerősítéses tanulási rendszer tudásbázisának hangolása során az állapot-akció tér ritkán bejárt területein lévő fuzzy Q-szabályok elhangolódása csökkenthető, ha az összes fuzzy szabálypont egyidejű hangolása helyett, csak azon szabályok kerülnek hangolásra, amely az éppen aktuális állapot-akció megfigyelési pont közelében találhatók.*

A II. tézishez kapcsolódó saját publikációk: [S1], [S9], [S10], [S12]

4.3 SZABÁLYBÁZIS REDUKCIÓ

A megerősítéses tanulási módszerek tudásábrázolási formája eltérő, a klasszikus Q-learning algoritmus Q-táblát (többdimenziós mátrix), a fuzzy szabályalapú megerősítéses tanuló rendszerek pedig fuzzy szabálybázist alkalmaznak a rendszer működtető tudásbázisának leírására. A végső tudásbázis mérete, azaz a Q-tábla elemeinek száma, a fuzzy szabályrendszer szabályainak száma függ az adott probléma méretétől, dimenzióinak számától, így előfordulhatnak olyan esetek mikor ezek mérete igen nagy. A fuzzy szabály interpoláció alapú megerősítéses tanulási rendszerekben a rendszer végleges működtető tudásbázis méretének csökkentésére szabálybázis redukálási (csökkentési) módszerek alkalmazhatók. A FRIQ-learning rendszerben [97][98] a tudásbázist leíró szabálybázis méretének csökkentésére a tanulási fázis után van lehetőség az elhagyható szabályok keresésével a 3.3.1 alfejezetben bemutatott módon. A szakértői tudásbázissal bővített FRIQ-learning (HFRIQ-learning) rendszerben a tudásbázis építési módszer működéséből adódóan a szabály antecedensek hangolása során elfordulhatnak olyan esetek mikor több szabály közel kerül egymáshoz. Ha az egymáshoz közel lévő szabályok közel ugyanazt az információt írják le, akkor célszerű azokat valamilyen stratégia alapján egyetlen szabállyá egyesíteni (összevonni), csökkentve ezáltal a szabálybázis méretét.

Jelen alfejezet célja a szakértői tudásbázissal bővített FRIQ-learning rendszerben egy olyan tudásbázis redukálási módszer bemutatása, amely már a tanulási fázis közben a közel hasonló információt leíró fuzzy szabályok összevonásával csökkenti a szabálybázis méretét.

A szakértői tudásbázissal bővített FRIQ-learning rendszerben [S7] a tanulási folyamat a szakértő által definiált szabályrendszer beépítésével indul. Ezt követően a tanulási fázis során epizódról-epizódra számos új szabály kerül beillesztésre, inkrementálisan bővítve a tudásbázist. A szakértői szabályrendszer bármennyi, a szakértő által meghatározott szabályt tartalmazhat, amelyek tetszőleges (a szakértő által meghatározott) szabálypontokban helyezkedhetnek el. A szabálybázis bővítése során új szabály akkor kerül beszúrásra az éppen aktuális megfigyelés pozíciójába, ha a legközelebbi szabálypont is távol van a megfigyeléstől és a Q-frissítési érték nagyobb, mint az előre meghatározott ε_Q küszöbérték $\left(\Delta\tilde{Q} > \varepsilon_Q\right)$. A szabálybázis hangolása során előfordulhat olyan eset, mikor kettő vagy több szabály közel kerül egymáshoz. Amennyiben ezek a hasonló antecedenssel rendelkező szabályok közel ugyanazt az információt írják le (q konzekvens értékük is hasonló), úgy a tanulási fázis után alkalmazható szabálybázis redukálási módszerek ezen szabályok valamelyikét valószínűleg el fogja távolítani a szabályrendszerből. Azonban ha már a tanulási folyamat közben megállapítható, hogy egyes

szabályok az állapot-akció térben (antecedens) közel kerülnek egymáshoz, akkor ezen szabályok egyesítésére, azaz valamilyen módszer alapján történő összevonására, már a tanulás fázisban is sor kerülhet. Azaz egy olyan szabálybázis redukciós (módosított tudásbázis építési) módszer fejleszthető, amely már a tanulási fázis közben ellenőrzi a közel ugyanazt az információt leíró szabályok előfordulását, majd valamilyen stratégia alapján egyiküket elhagyja, vagy összevonja (egyesíti) azokat.

A javasolt szabálybázis redukciós módszer elvárt jellemzői a következők:

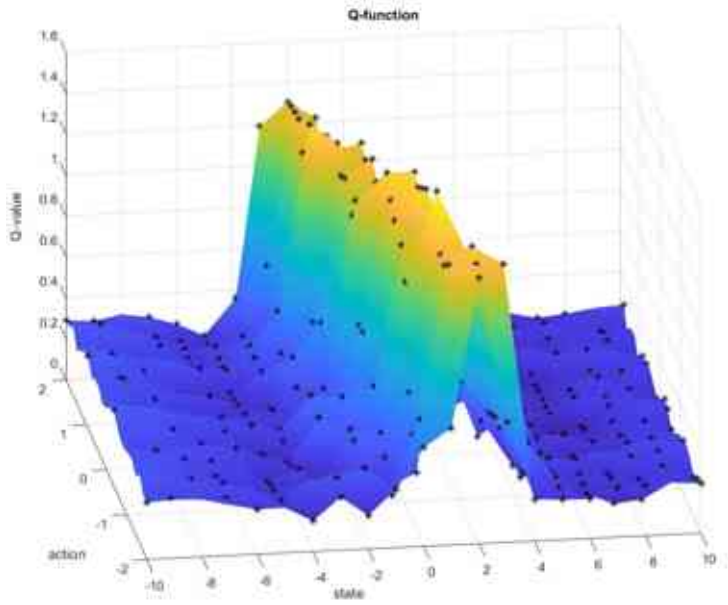
- A közel ugyanazt az információt leíró (egymáshoz hasonló) fuzzy szabályok összevonása (redukálása) egyetlen szabályként történjen a szabálybázis építési folyamat, azaz tanulási fázis közben a szabályok közötti távolság alapján.
- A szabályok közötti távolság és távolságküszöbök meghatározása történjen univerzumonként. Akkor tekinthető két szabály egymáshoz közelinek (hasonlónak), ha minden egyes dimenzióban (a konzekvens dimenzióban is) közelinek számít.
- A szabályok összevonása során legyen lehetőség a vizsgált szabályok típusának figyelembevételére, azaz a szakértői szabályok súlyának (fontosságának) beszámítására.
- A javasolt szabálycsökkentési módszer használatával a 3.3.1 fejezetben bemutatott I.-IV. szabálybázis redukciós módszerek utólagos alkalmazása egyes esetekben el is hagyható.

4.3.1 A közeli szabályok egyesítése

A tanulási fázis során a szabálybázis hangolási eljárása [S1][S10] következtében a szabályok állapot-akció pontja elmozdulhat (nem csak a konzekvense változhat), amely következtében előfordulhat olyan eset, mikor két vagy esetleg több szabály közel kerül egymáshoz. Ha az egymáshoz nagyon közel kerülő szabályoknak hasonló a konzekvense, akkor közel ugyanazt az információt írják le. Az ilyen hasonló szabályok egyesítésével (összevonásával) a szabálybázis mérte csökkenthető, redukálható.

Az egymáshoz közel kerülő szabályok összevonásának alapja az előző 4.2.2 alfejezetben bemutatott $\boldsymbol{dtr}$ távolságmérték. Ez az antecedens dimenziókra meghatározott $\boldsymbol{dtr}$ távolságküszöb azonban nem elegendő a hasonlóság definiálásához, mert előfordulhat olyat eset mikor két szabály ezen távolságküszöb alapján (azaz az antecedens dimenzióban) egymáshoz közelinek számít a (38) összefüggést alkalmazva, de a konzekvensükben (Q-értékükben) nagy az eltérés, ezért nem hasonló szabályok. Ebben az esetben nem célszerű a két (forrás) szabályt összevonni és egyetlen szabállyá redukálni mert lehet, hogy a közeli antecedensek és különböző konzekvensek az általuk leírt Q-függvényben egy meredek lejtőt

vagy emelkedőt jelentenek. Egy ilyen meredek „hegyoldallal” rendelkező Q-függvényt szemléltet a 17. ábra (illetve a 4.2.5 fejezetben bemutatott 14. ábra), ahol * karakter jelöli a szabálypontokat.



17. ábra: Meredek töréspontot tartalmazó Q-függvény

Ennek következtében a konzekvens (Q-érték) dimenzióra is szükséges közelségmérték definiálása, amely által csak akkor tekinthető két szabály egymáshoz hasonlónak, ha azok a távolságmértékek alapján az antecedens és a konzekvens univerzumokban is közelinek számítanak. A konzekvens dimenzióbeli távolságot d_Q jelöli, amely a két szabály Q-érték különbségének az abszolútértéke. A $d_Q(t,p)$ a t és p indexű szabályok ($t,p \in [1, m+\hat{m}]$) közötti távolság a konzekvens dimenzióban a következőképpen írható fel:

$$d_Q(t,p) = |Q^t - Q^p| \tag{57}$$

A konzekvens univerzumbeli közelségmérték meghatározása szintén egy távolságküszöb alapján történik, emiatt a $\boldsymbol{dtr}$ vektor kiegészül a konzekvens dimenzióra is vonatkozó $dtrQ$ távolságküszöbbel, így $\boldsymbol{dtr} = [dtr_1, dtr_2, \dots, dtr_n, dtrU, dtrQ]$. Ennek értéke a teljes (éppen aktuális) Q-érték tartomány valamekkora része, azaz a legnagyobb és a legkisebb Q-érték különbségének (a teljes tartomány hosszának) a szakértő által definiált dR_q része:

$$length(Q) = |\max(Q) - \min(Q)| \tag{58}$$

$$dtrQ = \frac{length(Q)}{dR_q} \tag{59}$$

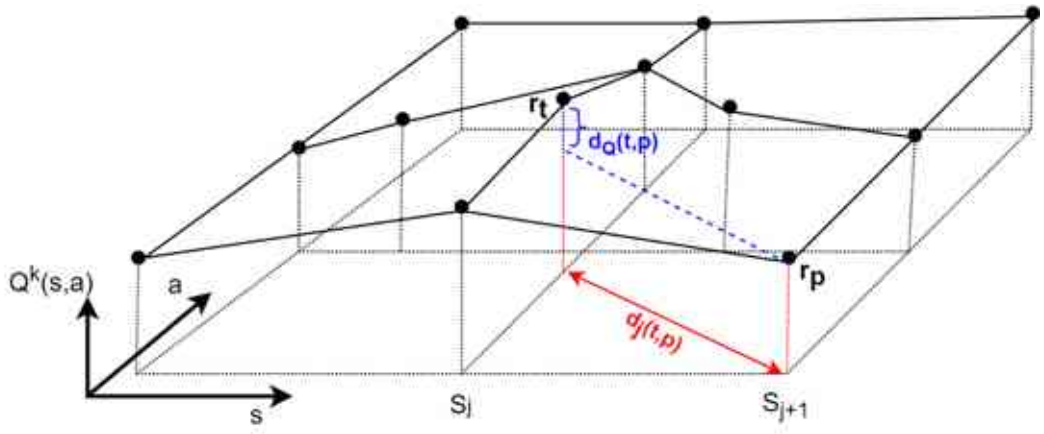
Mivel a Q-értékek a tanulási fázis során iterációnként változnak (a módszer által hangolásra kerülnek), így a $length(Q)$ értéke minden egyes olyan iterációban újraszámításra kerül ahol szabálybázis redukálás történik.

Összegezve, a szabálybázis redukálás során akkor tekintjük a szabálybázis két t és p indexű r_t és r_p szabályát egymáshoz hasonlónak és ennek következtében akkor kerülnek összevonásra (redukálásra) egyetlen r_{red} szabályként, ha a (38) összefüggés teljesül rájuk és még az r_t és r_p szabályok konzekvensében (Q-értékében) nincs nagy eltérés. Tehát minden egyes antecedens dimenzióban a távolságuk egymáshoz képest közelinek számít $\left(d_j(t,p) < dtr_j\right)$ és a két szabály $d_Q(t,p)$ Q-érték különbségének abszolútértéke kisebb, mint a dtr_Q küszöbérték:

$$\underset{t,p\in[1,m+\hat{m}]}{\exists} t,p \text{ hogy } d_Q(t,p) < dtrQ \tag{60}$$

Ebben az esetben a t indexű r_t és p indexű r_p szabályok összevonásra kerülnek egyetlen szabállyá. Akkor azonban, ha az antecedens dimenzióban közelinek tekinthetőek, de a konzekvensük a dtr_Q küszöbértéknél nagyobb, akkor nem kerülnek összevonásra.

A 18. ábra a t és a p indexű szabályok közötti konzekvens dimenzióbeli $d_Q(t,p)$ és az antecedens dimenzióbeli $d_j(t,p)$ távolságokat szemlélteti:



18. ábra: A p és a t indexű szabályok közötti $d_j(t,p)$ antecedens dimenzióbeli és $d_Q(t,p)$ konzekvens univerzumbeli távolság

Az új szabálypont, azaz az egyesítés után létrejövő új r_{red} szabály antecedensének és konzekvensének meghatározása a két forrásszabály antecedens és konzekvens értékeinek az átlagolásával történik. Mivel a szakértő által definiált produkciós szabályrendszer formátuma (27) a FRIQ-learning rendszerbe történő beillesztésük után a fuzzy Q szabály formára (19) változik, így az új szabály antecedensének és konzekvensének számítása az eredetileg szakértő által definiált szabályok esetén is a rendszer szabályainak egyesítésével azonos módon

történhet. A javasolt megoldás az, hogy az új szabály állapot-akció értéke (antecedense) és Q-értéke (következménye) legyen a forrás szabályok antecedens és konzekvens értékeinek az átlaga.

A fejlesztett, tanulási fázis közben alkalmazható szabályösszevonás (szabálybázis redukálás) algoritmusának pszeudokódja az alábbi [S3][S10][S13]:

Algoritmus: distBasedRBreduce(R, dR)

Bemenet: szabálybázis, dR paraméterek
Kimenet: redukált szabálybázis

redukált szabálybázis inicializálása
r_{red} redukált szabály inicializálása
távolságmátrix (D) inicializálása
távolságküszöbök (dtr) értékeinek számítása minden egyes antecedens és konzekvens dimenzióra dR alapján
Ciklus (minden egyes szabályra)
    szabálytávolságok $(d \in D)$ számítása minden egyes szabály között
    Ciklus (minden egyes antecedens dimenzióra)
        $If\ \left(\underset{t,p\in[1,m+\hat{m}]}{\exists} t,p \text{ hogy } \underset{j\in[1,n+1]}{\forall} \left(d_j(t,p) < dtr_j\right) \text{ és } \left(d_Q(t,p) < dtrQ\right)\right)\ , j \in [1, n+1]$
            r_t és r_p szabályok egyesítése egyetlen r_{red} szabállyá
            r_{red} antecedensének és konzekvensének számítása r_t és r_p szabálypontok átlagolása által
            r_{red} redukált szabály hozzáadása a redukált szabálybázishoz
        end
return redukált szabálybázis

7. algoritmus: A fejlesztett szabálybázis redukálási módszer algoritmusa, amely a tanulási folyamat közben alkalmazott

Továbbfejlesztésként célszerű lehet majd egy olyan módszer kidolgozása, amely alkalmas lehet a szakértő által meghatározott konstans dR paraméterek (és így a távolság alapú küszöbértékek) hangolására, optimalizálására.

4.3.2 Az összevont szabály típusának meghatározása

A szakértői tudásbázissal kiegészített FRIQ-learning rendszerben a fuzzy szabályoknak három típusa különböztethető meg: a szakértő által definiált szabályrendszer (R_{expert}), a rendszer által létrehozott szabályrendszer (R) és a rendszer által létrehozott sarokponti szabályok $(r^{\square} \in R)$. A sarokponti szabályrendszer az interpolációs eljárás miatt szükséges 2^{n+1} darabszámú sarokponti szabályt, a rendszer által létrehozott szabályrendszer pedig az újonnan létrehozott szabályokat tartalmazza. A szabályok különböző típusa miatt az egyesítés (redukálás) során előálló új szabály típusát célszerű a forrásszabályok típusa alapján meghatározni. Mivel a szakértő által definiált szabályok feltételezhetően helyes tudást írnak le a rendszer működésére vonatkozóan, így azok nagyobb fontossággal (súllyal) kerülnek

figyelembe vételre. Ezen fontosság (vagy súly) határozza meg az új szabály típusát illetve azt, hogy a forrás szabályok milyen módon kerülnek, vagy éppen nem kerülnek összeolvasztásra egyetlen szabállyá.

Két egymáshoz közel kerülő szabály esetében az új, egyesített szabálytípus meghatározásának módja a következő: ha a két szabály közül az egyik szakértői a másik pedig a rendszer által felvett új szabály, akkor az egyesített szabály szakértői szabály lesz. Ha a két szabály közül mindkét szakértői szabály volt, akkor az új összevont szabály is szakértői szabály lesz. Ha a két szabály közül mindkét szabály a rendszer által újonnan felvett szabály, akkor az új egyesített szabály is újonnan beszúrt szabályként lesz jelölve.

Speciális eset mikor az egymáshoz közeli szabályok közül az egyik sarokponti típusú a másik pedig attól különböző, azaz szakértői vagy újonnan létrehozott szabály, tehát a sarokpontot leíró szabály közelébe kerül egy attól különböző típusú szabály. Annak következtében, hogy a sarokponti szabályok a „FIVE" interpolációs módszer alkalmazása miatt fontos szerepet töltenek be a szabálybázis építés folyamatában, így ezek eltérő fontossági súllyal kerülnek figyelembevételre a szabályegyesítés során. Ha a közeli szabály a rendszer által létrehozott szabály, akkor az törlésre kerül a sarokponti szabály közeléből. Ha a közeli szabály szakértői szabály volt, akkor az változatlanul megmarad. Ezekben az esetekben így nem történik szabályösszevonás, mert a sarokponti szabály antecedense nem változhat.

A szabályok típusának meghatározásához illetve az egyes szabályok változásának (hangolásának) nyomon követése céljából a szabályrendszer minden egyes szabálya egy egyedi azonosítóval (*ID*) rendelkezik. Az egyedi szabályazonosítók meghatározásának módja szabálytípusonként az alábbi:

- sarokponti szabály azonosító: 1000-1999
- szakértői szabály azonosító: 2000-2999
- új, rendszer által felvett szabály azonosító: 3000-3999

Ezen egyedi azonosítóval a szabályok hangolása és egyesítése nyomon követhető, a hangolási folyamat végén kinyerhető. Utólagosan meghatározható, hogy a hangolási, optimalizálási folyamat során a szakértő által megadott szabályok milyen mértékben változtak.

Egy, a szakértői által definiált szabályt jelöljünk $\hat{r}$-el $\left(\hat{r} \in R_{expert}\right)$, egy a rendszer által felvett szabályt r-el $(r \in R)$, egy sarokponti szabályt pedig $r^{\square}$-el $(r^{\square} \in R)$. A (61) összefüggés az adott típusú, r_t és r_p indexű szabályok egyesítését követően létrejött új r_{red} szabály típusát

szemlélteti. A „⊔” operátor a két szabály egyesítését, a „→” operátor pedig a szabályegyesítés eredményét, azaz a szabályösszevonás során létrejött új r_{red} szabály típusát jelöli:

r_t		r_p		r_{red}
r	⊔	$\hat{r}$	→	$\hat{r}$
$\hat{r}$	⊔	$\hat{r}$	→	$\hat{r}$
r	⊔	r	→	r
$r^{\square}$	⊔	r	→	$r^{\square}$
$r^{\square}$	⊔	$\hat{r}$	→	$r^{\square}, \hat{r}$

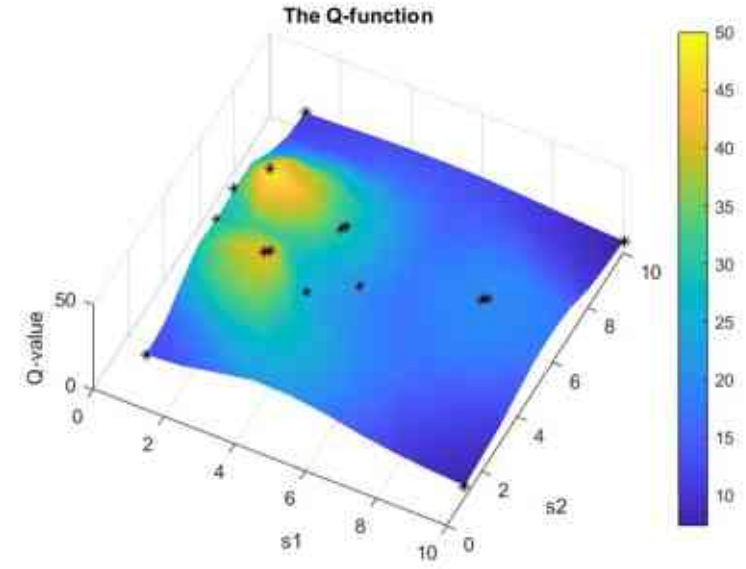
(61)

A 15. táblázat egy lehetséges szabályegyesítési példát szemléltet, ahol $\hat{r}$ egy szakértői szabály, r pedig egy rendszer által beszúrt (új) szabály. Feltételezzük, hogy r és $\hat{r}$ távolsága közelinek tekinthető, továbbá s_1, s_2 az állapot dimenziók, a az akcióérték, q a Q-érték, $\hat{r} \sqcup r \rightarrow \hat{r}$ pedig az egyesített új szabály, amely szakértői szabályként kerül megjelölésre:

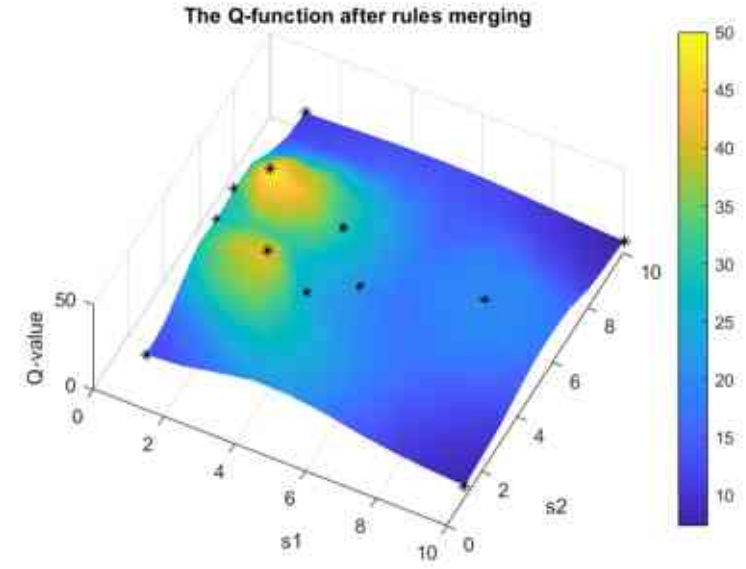
15. táblázat: Szabályegyesítés egy szakértői és egy rendszer által felvett szabály esetében

Szabály	**s_1 állapot**	**s_2 állapot**	**a akció**	**q-érték**
$\hat{r}$	1	2	4	123.45
r	2	2	5	145.23
$\hat{r} \sqcup r \rightarrow \hat{r}$	1.5	2	4.5	134.34

A 19. és 20. ábrák egy mintapéldán keresztül vizuálisan is szemléltetik a szabályegyesítés folyamatát. Az ábrákon egy Q-függvény felülete látható, melyet a „*” által jelölt fuzzy szabályok (tartópontok) írnak le. A függvény két állapotdimenzióval (s_1 és s_2) és egy akciódimenzióval rendelkezik. A Q-függvény megjelenítésénél a legjobb akció melletti Q-értékek kerültek kirajzolásra az s_1 és s_2 univerzumok mentén, az akciódimenziót a megjelenítésből. A 19. ábra a szabályegyesítés (redukálás) előtti állapot, mikor 2-2-2 darab (3 pár) egymáshoz közeli szabály található, a 20. ábra pedig a szabályegyesítés utáni állapotot szemlélteti, miután ezen közeli szabályok összevonásra kerültek. Így a 15 darab szabály helyett már csak 12 darab, „*” által jelölt tartópont (fuzzy szabály) írja le a függvény felületét:



19. ábra: Q-függvény felülete szabályegyesítés előtt (15 darab „”-al jelölt fuzzy szabállyal)*



20. ábra: Q-függvény felülete szabályegyesítés után (12 darab „”-al jelölt fuzzy szabállyal)*

A szabályösszevonások eredménye táblázatban szemléltetve:

16. táblázat: A 18. és 19. ábrákon látható szabályegyesítések eredménye

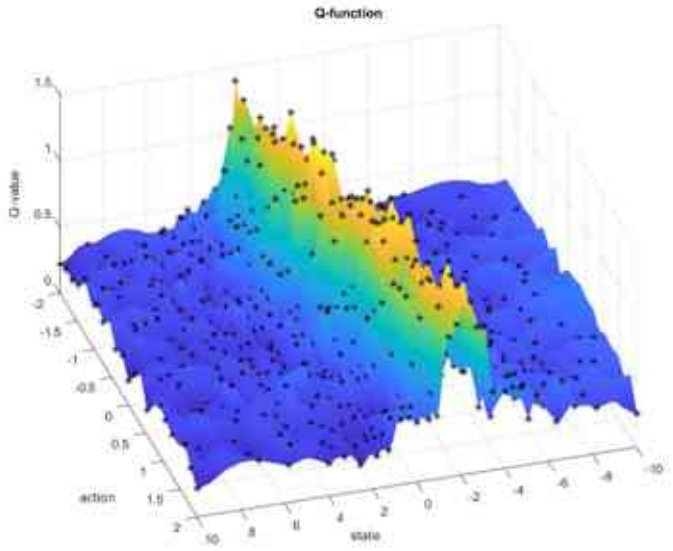
Szabály	s_1	s_2	a	q
r_1	4	6	1	10
r_2	4.1	6.1	1	10.2
$r_1 \sqcup r_2 \rightarrow r$	4.05	6.05	1	10.1
r_3	3	4	1	42
r_4	2.9	3.9	1	41.8
$r_3 \sqcup r_4 \rightarrow r$	2.95	3.95	1	41.9
r_5	8	6	1	20
r_6	8.1	6.1	1	20.3
$r_5 \sqcup r_6 \rightarrow r$	8.05	6.05	1	20.15

4.3.3 Mintapéldák

A javasolt távolság alapú szabálybázis redukálási módszer hatékonyságát három futási eset felhasználásával vizsgáltam. Az első eset a szabálybázis redukálás nélküli, a második eset a szabálybázis redukálás (szabályegyesítés) során a szabályok hasonlóságát csak az antecedens dimenziók alapján vizsgálja, a harmadik eset a szabályok hasonlóságát az antecedens és konzekvens dimenziók távolságküszöbei alapján vizsgálja.

Mindegyik futási esetben 10000 iteráción keresztül futott a szimuláció, a második és harmadik esetben alkalmazásra került a gradiens alapú hangolási eljárás is, amely α paramétere 0.05 volt és egyetlen darab szakértői szabály került felvételre a 0 állapotban 0 értékű akcióval.

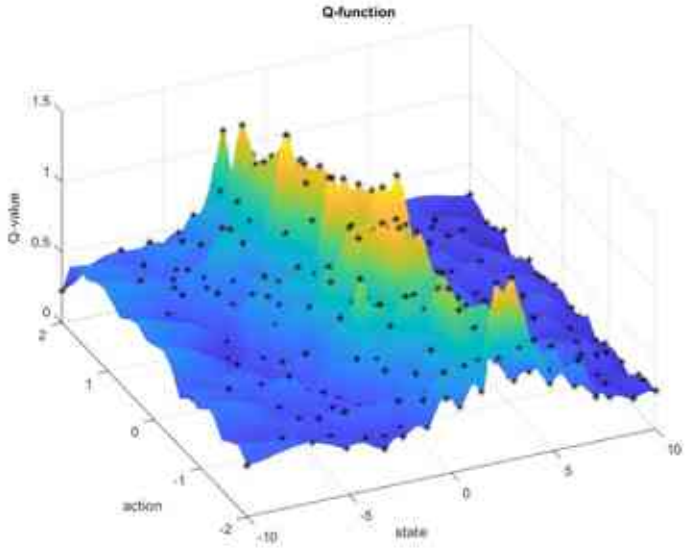
Az összehasonlítás alapja a 21. ábra szabálybázis redukálás nélküli, egy állapot- és egy akcióváltozóval rendelkező („referencia”) Q-függvény, ahol a szabályfelvétel során a szabályok között megengedett minimális távolság az univerzumok hosszának a 100-ad része ($\mathrm{dR}_{\boldsymbol{S}} = \mathrm{dR}_U = 100$). A futás során kapott szabálybázis ebben az esetben 530 darab szabályt tartalmaz:



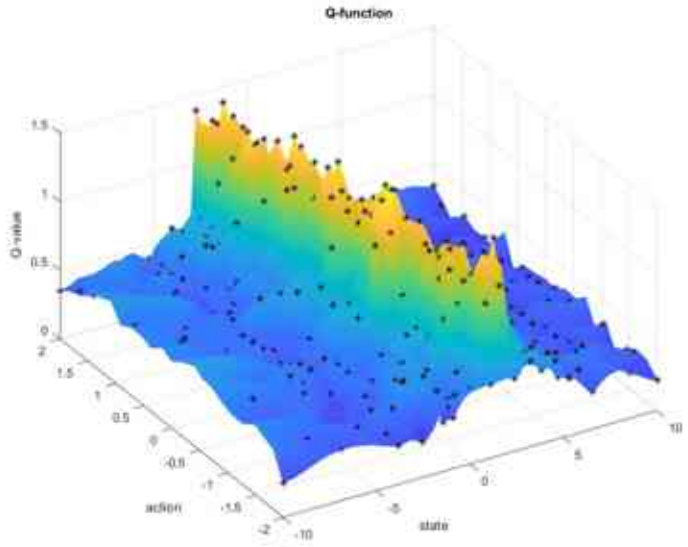
21. ábra: Az összehasonlítás alapjául szolgáló „referencia" Q-függvény

A második futási esetben a futás a szabálybázis redukálási módszer alkalmazásával történik, de a minimális szabálytávolságok csak az antecedens dimenziókra vannak meghatározva. Ebben az esetben, ha két (vagy több) szabály az antecedens univerzumokban meghatározott távolságküszöbök által közelinek számít, akkor azok összevonásra kerülnek egyetlen szabállyá. Ez olyan esetekben okozhat problémát, ha a két forrás szabály konzekvensében (Q-értékében) nagy az eltérés, azaz a konzekvens univerzumban távolinak tekinthetők, de mégis egyesítésre kerülnek egyetlen szabállyá. Ekkor a forrás szabályok egyesítése során, az antecedens és konzekvens értékeiknek átlagolása következtében, az általuk leírt szabálypontban a Q-függvény alakja rossz irányban módosulhat. Ez által elromolhat a szabályok által leírt Q-függvény, hamis információt adva az állapot-akció értékekről.

A 22. ábra futási esetében a szabályegyesítés csak az antecedens univerzumban történő szabálytávolságok és távolságküszöbök alapján történik. A 23. ábra futási esetében a szabályegyesítés az antecedens és a univerzumban is vizsgálja a szabályok távolságát. Mindkét futási esetben a szakértő által megadott, szabálybázis redukálásra vonatkozó *R* paraméterek értéke 50 volt, de az első futási esetben csak $\mathrm{dR}_{\boldsymbol{S}}$ és dR_U értékei voltak megadva $(\mathrm{dR}_{\boldsymbol{S}} = \mathrm{dR}_U = 50)$, a második futási esetben pedig ez kiegészült a konzekvens univerzumra vonatkozó dR_q távolságküszöbbel is $\left(\mathrm{dR}_{\boldsymbol{S}} = \mathrm{dR}_U = \mathrm{dR}_q = 50\right)$.



22. ábra: A 2. futási esetben létrejött Q-függvény, mikor a távolságküszöbök csak az antecedens univerzumokra meghatározottak



23. ábra: A 3. futási esetben létrejött Q-függvény, mikor a távolságküszöbök az antecedens és a konzekvens univerzumokra is alkalmazásra kerültek

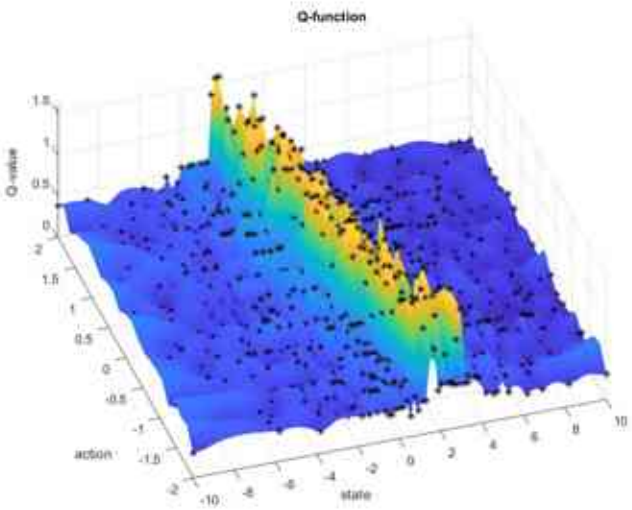
Az egyes futási esetekben kapott eredményeket a következő táblázat foglalja össze:

17. táblázat: Az egyes futási esetekben kapott szabálybázis mérete

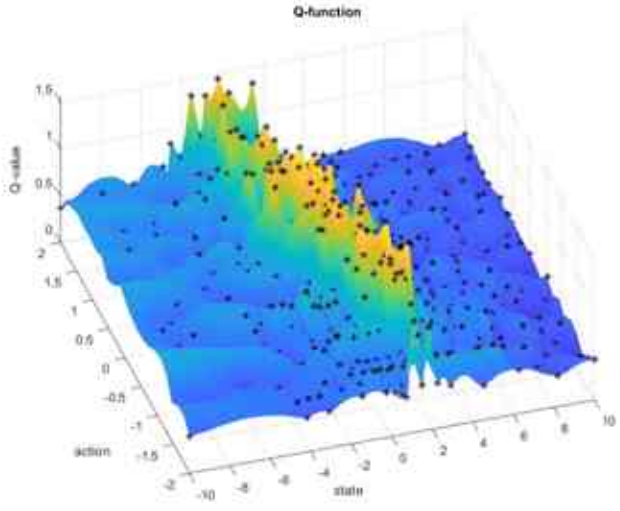
#	Futási eset	Szabályszám
1.	szabálybázis redukálás nélkül	530
2.	szabálybázis redukálás csak az antecedens univerzumra alkalmazott szabálytávolságokkal és távolságküszöbökkel	334
3.	szabálybázis redukálás a szabálytávolságok és távolságküszöbök konzekvens univerzumra történő kiterjesztésével	403

A második futási esetben kevesebb szabályt tartalmaz a szabálybázis, de a 22. ábrán látható, hogy a függvény „gerince” nem alakult ki teljesen (a 21. ábrához viszonyítva), tartalmaz több alacsonyabb csúcsot, mert olyan szabályok is összevonásra kerültek, melyek Q-értékében nagy volt az eltérés és az átlagolás következtében ezen csúcsok alacsonyabbak lettek. Ebben az esetben a kapott összejutalom értéke is kisebb. A harmadik futási esetben a szabálybázis több szabályt tartalmaz, mint a második futási esetben, de az 23. ábrán látható, hogy a függvény gerince kialakult, a távolságok és távolságküszöbök konzekvens univerzumra történő kiterjesztése miatt nem kerültek olyan szabályok összevonásra, melyek Q-értékében nagy az eltérés, így ez által nem romlott el a Q-függvény formája.

A 24. és 25. ábrákon látható Q-függvény felületek újabb, az előzőtől példától független futási eseteket szemléltetnek:



24. ábra: Q-függvény a szabálytávolság alapú szabálybázis redukálás nélkül, 901 fuzzy szabály



25. ábra: Q-függvény a szabálytávolság alapú szabálybázis redukálási módszer alkalmazásával, 485 fuzzy szabály

Mindkét esetben a rendszer 5000 iteráción keresztül futott, a HFRIQ-learning gradiens alapú hangolási módszer α paraméterének értéke 0.05 volt. Az első esetben a szabálytávolság alapú

szabálybázis redukálás nélkül (24. ábra), mikor a szabályfelvételkor megengedett minimális szabálytávolság az antecedens univerzumok hosszának 200-ad része volt ($\mathrm{dR}_S = \mathrm{dR}_U = 200$). A szabálybázis ebben az esetben 901 szabályt tartalmazott. A második esetben, mikor a távolságalapú szabálybázis redukálás alkalmazásra került (25. ábra), a szabálybázis redukálásra vonatkozó távolságküszöbök értékei pedig $\mathrm{dR}_S = \mathrm{dR}_U = 45$, $\mathrm{dR}_q = 100$, ekkor a szabálybázis mérete kisebb lett az előző futási esethez képest, 485 darab szabállyal írta le ugyanazt a Q-függvényt így a szabályszám körülbelül 53%-al csökkent.

4.3.4 Klaszterezési módszeren alapuló szabálybázis redukció

A 4.3.1-4.3.3 alfejezetekben bemutatott, tanulási fázis közben alkalmazható, fuzzy szabályok közötti távolságokon (és távolságküszöbökön) alapuló szabálybázis redukciós módszeren kívül egy olyan klaszterezés alapú szabálybázis redukálási eljárást [S8] is kifejlesztettem, amely a tanulási folyamat után alkalmas a Q fuzzy szabálybázis optimalizálására. A javasolt 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.

A javasolt módszer 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 [55] 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 és j-edik indexű szabálya (azaz adatpontja) közötti távolságot d_{ij} jelölti, ahol $i, j = [1, \dots, m]$.

$$D_{[ij]} = d_{ij} \tag{62}$$

A következő fázisban a távolságmátrix alapján p_1, p_2 pivot objektumok (a klaszterezési módszer speciális pontjai, jelen esetben a fuzzy szabályok) kerülnek meghatározásra, amelyek

a két egymástól legtávolabbi szabályok lesznek. Ezek tulajdonképpen a D távolságmátrix d_{ij} távolságértékei közöl a legkisebb és a legnagyobb távolságértékkel rendelkező, adott indexű szabályok:

$$p_1 = arg \min_{d_{ij} \in D} (d_{ij}) \tag{63}$$

$$p_2 = arg \max_{d_{ij} \in D} (d_{ij}) \tag{64}$$

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 [55] az alkalmazásával történik. A szabálybázis szabályai a pivot objektumoktól való távolságuk alapján kerülnek besorolásra klaszterbe. A klaszterhez való hozzárendelés alapja egy ε távolságküszöb, amely a következőképpen számítható:

$$\varepsilon = \frac{\max_{p_1, d_{ij} \in D} (d_{ij})/2 + \max_{p_2, d_{ij} \in D} (d_{ij})/2}{2} \tag{65}$$

Az ε távolságküszöb értéke tehát, amely alapján az egyes klaszterek (majd alkalszterek) kerülnek kialakításra, a p_1 és p_2 adatpontoktól vett legtávolabbi távolságértékek felének az összegei, majd az ezek alapján számított átlag.

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 az (65) 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 [30]. 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 algoritmus pszeudokódja az alábbi [S8]:

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 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

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

A javasolt klaszterezési eljáráson alapuló szabálybázis redukálási algoritmus hatékonysága a „Cart-Pole” és a „Mountain Car” mintapéldákon lett vizsgálva és a 3.3.1 fejezetben bemutatott I., II. és III. jelölésű szabálybázis redukálási módszerek által kapott eredményekkel összevetve. A javasolt klaszterezési eljáráson alapuló szabálybázis redukálási módszer (az ábrán 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. Az egyes futási esetekben kapott eredményeket a 18. táblázat és 19. táblázat foglalja össze:

18. 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

19. 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 Cart-Pole mintapélda esetében a javasolt klaszterezési eljáráson alapuló szabálybázis redukálási módszer (IV. jelű) 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 a javasolt szabálybázis redukálási módszer (IV. jelű) 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. A 26. és 27. ábrák a tanulási fázis után alkalmazott egyes szabálybázis redukálási módszerek által kapott futási időket szemléltetik:

26. ábra: Szabálybázis redukálási idők a Cart-Pole esetében

27. ábra: Szabálybázis redukálási idők a Mountain Car esetében

A javasolt klaszterezési eljáráson alapuló szabálybázis redukálási módszer futási idejének nagy részét nem a hierarchikus klaszterezési algoritmus (illetve a „FIVE” FRI távolságszámítás) lépései teszi 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 20. táblázat foglalja össze:

20. 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

4.3.5 III. tézis

A HFRIQ-learning megerősítéses tanulási módszer tudásbázisának mérete a tanulási folyamat során csökkenthető a hasonló fuzzy Q-szabályok összevonásával. A fuzzy szabályok hasonlósága becsülhető antecedenseik és konzekvenseik távolságával.

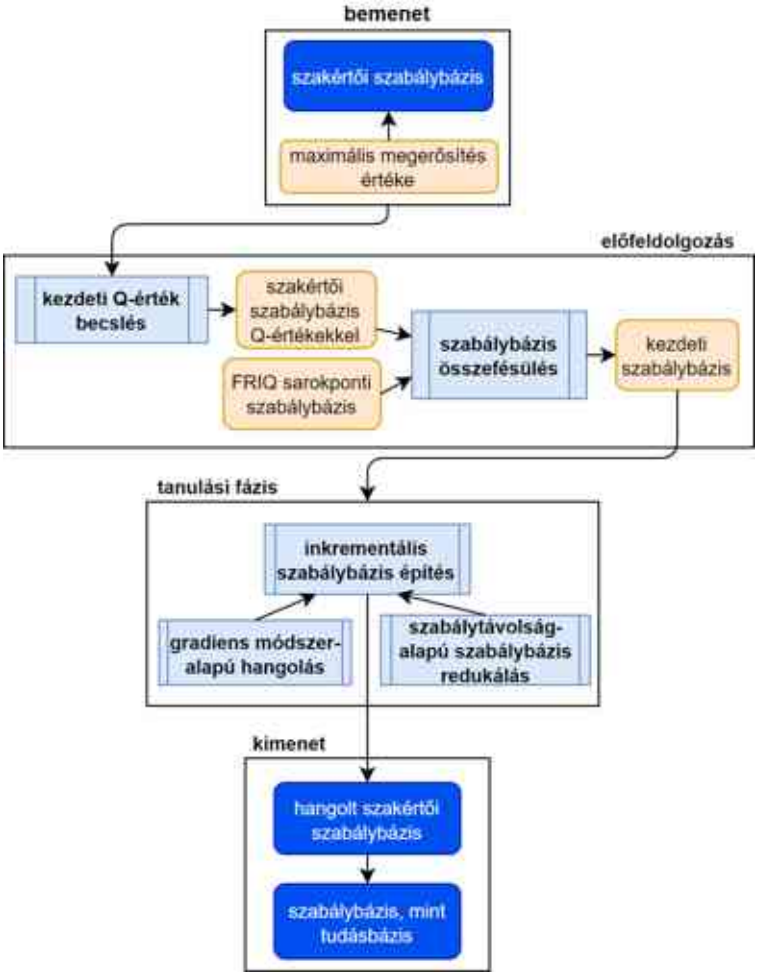
III.1. Altézis: *A szabályok hangolása és összevonása során követhető a szabályok típusa, így a tanulási folyamat végeztével a HFRIQ-learning Q fuzzy szabályrendszeréből a kezdeti szakértői szabályok visszanyerhetők. Az eredetileg megadott és a hangolást követően visszanyert szakértői szabályok összevetésével következtetni lehet a kezdeti szakértői szabályok helyességére.*

III.2. Altézis: *A hierarchikus klaszterezési módszer alkalmas a HFRIQ-learning tanulási fázisának végeztével előállt Q-függvényt leíró fuzzy szabályrendszer méretének csökkentésére.*

A III. tézishez kapcsolódó saját publikációk: [S3], [S5], [S8], [S13]

4.4 A HFRIQ-LEARNING

A fejlesztett, heurisztikusan gyorsított FRIQ-learning (HFRIQ-learning) rendszer felépítésének blokkvázlatát az alábbi 28. ábra szemlélteti.



28. ábra: A heurisztikusan gyorsított FRIQ-learning (HFRIQ-learning) rendszer felépítése

A rendszer bemenete a szakértő által definiált kezdeti állapot-akció formátumú (produkciós) fuzzy szabályrendszer. A szintén szakértő által definiált lehetséges maximális megerősítés értéknek ismeretében a kezdeti Q-érték becslő módszer Q-értékeket határoz meg a szakértői szabályokra, amely által annak formátuma állapot-akció-Q-érték-re módosul. Ellenőrzésre kerül, hogy az előző lépésben létrehozott Q-értékekkel rendelkező szakértői szabálybázis valamelyik szabálya illeszkedik-e az FRIQ-learning rendszer valamely kezdeti (sarokponti) szabályára, azaz előfordulnak-e ellentmondó szabályok, melyek így azonos antecedenssel de eltérő konzekvenssel rendelkeznek. Ha igen, akkor az ellentmondás feloldásaként a FRIQ-sarokponti szabály megkapja az illeszkedő szakértői szabály konzekvensét majd a szakértői szabály eltávolításra került a szabálybázisból.

Ez a folyamat a javasolt szabálybázis összefésülési módszer által valósul meg. Az így kialakult, ellentmondó szabályokat már nem tartalmazó, összefésült szakértői és FRIQ-sarokponti szabályrendszer a rendszer tanulási fázisának a kezdeti szabálybázisa. A tanulási fázis, azaz az inkrementális szabálybázis építési folyamat során a szakértő által definiált kezdeti szabálybázis kerül kiegészítésre új szabályokkal majd hangolásra és redukálásra. Új szabály akkor kerül felvételre a szabálybázisba, ha az aktuális megfigyelés közelében még nincs létező szabály és a Q-frissítés értéke is meglehetősen nagy. Az új szabály állapot-akció pontja a megfigyelés állapot-akció pontjával megegyező lesz. Ellenkező esetben, ha a megfigyelés közelében már van létező szabály és a Q-frissítés értéke is viszonylag kicsi, akkor a megfigyelés közelében lévő szabályok kerülnek hangolásra a javasolt gradiens-módszer alapú optimalizálási eljárással. Ilyenkor az adott szabály antecedense (állapot-akció pontja) és a konzekvense (Q-értéke) is hangolásra, finomításra kerül. A szabálypontok vándorlása (elmozdulása) következtében előfordulhat olyan eset, mikor egy vagy több szabály közel kerül egymáshoz. Abban az esetben, ha az egymáshoz közel elhelyezkedő szabályok nagyon hasonló információt (akciót) írnak le, akkor ezen szabályok a tanulási folyamat során összevonásra (egyesítésre) kerülnek a javasolt szabálytávolság alapú szabálybázis redukálási módszerrel. A tanulási fázis, azaz az inkrementális szabálybázis építési folyamat végeztével - mikor már nem kerül új szabály felvételre a szabálybázisba és a Q-frissítés értéke is elenyészően kicsi – előáll a rendszert működtető tudásbázis, azaz szabálybázis. Ezen szabálybázis részei a szakértő által megadott produkciós szabályok, melyeket a tanulási folyamat során javasolt szabálybázis redukálási módszer nyomon követ, így azok paraméterei a hangolási folyamat befejezése után kiolvashatók, új értékük a kezdeti szakértői szabályokkal összevethető. A tanulási folyamat végeztével a szabálybázis mérete a javasolt klaszterezési módszeren alapuló szabálybázis redukálási eljárással tovább csökkenthető.

A disszertáció és a kutatás tárgyát képező, továbbfejlesztett FRIQ-learning módszer, amely alkalmas szakértő által előzetesen definiált tudásbázis FRIQ-learning rendszerbe történő beillesztésére, majd annak hangolására és a tanulási folyamat közbeni redukálására, elnevezése: **Heuristically Accelerated Fuzzy Rule-Interpolation based Q-learning** (*HFRIQ-learning*).

A fejlesztett HFRIQ-learning módszer algoritmusának pszeudokódja az alábbi:

Algoritmus: HFRIQ(expertR)

```
Bement: szakértői szabálybázis, α, γ, g_max
Kimenet: hangolt szabálybázis

Q-értékek számítása a szakértői szabálybázisra
szakértői és FRIQ-sarokponti szabálybázis egyesítése
Ismétlés (minden egyes epizódra):
    állapot inicializálása
    Ismétlés (az epizód minden egyes iterációjára)
        a választása s-ből adott politika alkalmazásával (például ε-mohó)
        a akció végrehajtása, r, s' megfigyelése
        ΔQ̃ Q-frissítési érték számítása
        if (Q-frissítési érték nagy: ΔQ̃ > ε_Q)
            if (megfigyelés szabálypontra illeszkedik)
                szabálybázis konzekvensének frissítése
            else
                minden szabály távolságának számítása a megfigyeléstől
                if (szabálytávolság < távolságküszöb)
                    közeli szabálypont hangolása a gradiens-módszer alkalmazásával
                    szabálybázis redukálása a szabálytávolság alapú redukálási módszerrel
                else
                    új szabály létrehozása és felvétele az aktuális megfigyelés pozíciójába
                end
            end
        else
            szabálybázis konzekvensének frissítése
        end
    állapot- és akció változók értékeinek frissítése
    amíg s terminális állapot nem lesz
amíg új szabály felvétele történik és a Q-frissítés értéke relatívan nagy
return hangolt szabálybázis, ami a szakértői szabályokat is tartalmazza
```

9. algoritmus: A fejlesztett HFRIQ-learning módszer algoritmusa

4.5 HFRIQ-LEARNING ALKALMAZÁSPÉLDÁK

Ezen alfejezet a javasolt HFRIQ-learning módszer működésének hatékonyságát mutatja be egyetlen állapot- és egyetlen akcióváltozóval rendelkező mintapéldán és további klasszikus megerősítéses tanulási alkalmazáspéldákon keresztül.

4.5.1 Egy állapot-akció változós mintapélda

A mintapélda nem egy klasszikus megerősítéses tanulási alkalmazáspélda, hanem a Q-függvény egyszerűbb vizualizációjának érdekében, egyetlen állapot és egyetlen akciódimenzióval rendelkező feladat. Az s_1 állapotváltozó értéktartománya -10 és +10 közé esik ($s_1 \in [-10,10]$), az a akcióváltozó pedig -2 és +2 között vehet fel értékeket ($a \in [-2,2]$), a tanulási módszer α paramétere 0.5 értékű, a γ diszkontálási tényező értéke 0.4, az ε-mohó akcióválasztás ε értéke pedig 0.5. A szakértő által meghatározott tudásbázis egyetlen

állapot-akció formátumú szabályt tartalmaz, amely a 0 állapotpontban 0 értékű akciót határoz meg, és az erre megadott megerősítés értéke $g_{max} = 1$. A környezet +1 jutalmat ad, ha az ágens állapotváltozójának értéke -1 és +1 közötti, ellenkező esetben a jutalom értéke 0. A jutalomfüggvény a következő:

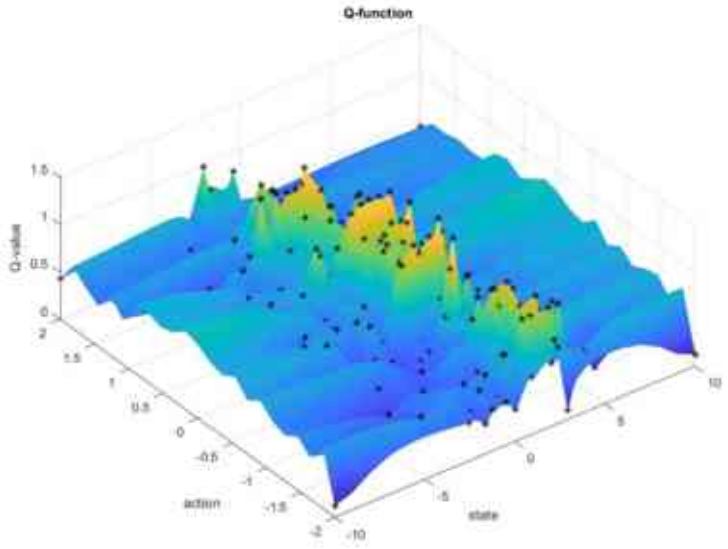
Jutalomfüggvény: 1D_mintapélda

```
Bemenet: s1 állapot
Kimenet: r megerősítés

if (s1 <= 1 and s1 >= -1)
    r = 1
else
    r = 0
end

return r
```

3. jutalomfüggvény: Az egy állapot-akció változós mintapélda jutalomfüggvényének pszeudokódja

Az így kapott HFRIQ-learning futási eredmények összehasonlításának alapja a FRIQ-learning azon verziójának futása során kapott szabálybázis méret (szabályszám) illetve Q-függvény forma, ahol az eredeti állapot-akció tér rácsháló elhagyásra került és a szabályok között megengedett minimális szabálytávolság volt csak figyelembe véve. Az első futási esetben tehát a FRIQ-learning eredeti verziójában került futtatásra a mintapélda 10000 iteráción keresztül, sem a szakértői tudásbázis, sem a javasolt gradiens módszer alapú hangolási eljárás, sem pedig a szabálytávolság alapú szabálybázis redukálási módszer nem került alkalmazásra. Ebben az esetben a kapott szabálybázis 530 szabályt tartalmazott, amelyek által leírt Q-függvényt a 4.3.3 alfejezetben bemutatott 21. ábrán látható „referencia” Q-függvény szemléltet.

A második futási esetben egyetlen darab szakértői szabály lett a rendszerbe illesztve, illetve alkalmazásra került a javasolt gradiens módszer alapú szabálybázis hangolási eljárás és a szabálytávolság alapú szabálybázis redukálási módszer (HFRIQ-learning). A szakértői szabály a 0 állapot pozícióban preferált 0 értékű akciót definiálja. A szabálybázis építési folyamat során a szabályok között megengedett minimális szabálytávolság az univerzumok méretének a 200-ad része ($\mathrm{dR}_{\boldsymbol{S}} = \mathrm{dR}_U = 200$). A szabálytávolság alapú szabálybázis redukálás dR paramétereinek értéke: $\mathrm{dR}_{\boldsymbol{S}} = 45$, $\mathrm{dR}_U = 45$ és $\mathrm{dR}_q = 100$. Ebben a futási esetben a szabálybázis 327 szabályt tartalmazott, amelyek által leírt Q-függvény a 29. ábrán látható:



29. ábra: A fejlesztett HFRIQ-learning rendszer alkalmazása esetében kapott Q-függvény

A 29. ábrán látható, hogy az így kapott Q-függvényt kevesebb tartópont (szabály) írja le a szabálytávolság alapú szabályösszevonás következtében, illetve a függvény formája a gradiens módszer–alapú szabálypont optimalizálás miatt kisimult a 21. ábrán látható referenciaként szolgáló Q-függvényhez képest. A 21. táblázat az egyes futási esetekben kapott szabálybázis méreteket (szabályszámokat) foglalja össze:

21. táblázat: Az egyes futási esetekben kapott szabálybázis méretek

#	Futási eset	Szabálybázis méret
1	szakértői tudásbázis nélkül illetve a gradiens módszer alapú hangolási eljárás és a szabálytávolság alapú szabálybázis redukálás nélkül	530 szabály
2	szakértői tudásbázissal és a gradiens módszer alapú hangolási eljárás és a szabálytávolság alapú szabálybázis redukálási módszerek alkalmazásával	327 szabály

A 22. táblázatban a tanulási fázis előtt, a szakértői által eredetileg definiált szabály illetve a tanulási fázis (a javasolt hangolási módszerek alkalmazása) után előállt szakértői szabály található:

22. táblázat: A szakértő által definiált eredeti, illetve a hangolási folyamat után előállt szabály

#	Szakértői szabály	állapot	akció	Q-érték
1	eredetileg (tanulási fázis előtt) megadott	0	0	0.1
2	hangolt (tanulási fázis után)	0.06	0	0.59

Mivel a példában a szakértői szabály helyesen definiált volt, ezért a hangolás során a szabály állapot-akció pontja (azaz a szakértői szabály maga) csak kismértékben (akció értéke egyáltalán nem) változott (lásd 22. táblázat 1., 2. sora). A szabálypont hangolása után előállt pozitív Q-érték alapján a szabály helyes, a Q-érték változása csak a szabály hasznosságára vonatkozó becslést pontosítja.

4.5.2 „Mountain Car" alkalmazáspélda

Ezen alfejezet egy klasszikus megerősítéses tanulási mintapéldán keresztül mutatja be a javasolt rendszer működését, hatékonyságát. A választatott mintapélda a „Mountain Car" elnevezésű, amely a disszertáció korábbi 4.1.5 fejezetében részletesen bemutatásra került. A mintapélda futása során, a tanulás paramétereinek értéke ($\alpha = 0.5,\ \gamma = 0.99$) ugyanaz maradt jelen esetben is, mint ahogyan azok a 4.1.5 fejezetben bemutatásra kerültek, de azzal az eltéréssel, hogy egy epizód nem 1000 lépésen (step), hanem 2000 lépésen keresztül futott a szimulációban.

Az összehasonlítás alapjául három futási esetet hoztam létre, amelyek a következők:

1. Eredeti FRIQ-learning verzió [97][98] (melyben az állapot-akció tér rácsháló alkalmazásra kerül, amely által az új szabályok állapot-akció pontja meghatározott).
2. Az előző 1. eset kiegészítve szakértői tudásbázissal a 4.1.1-4.1.3 fejezetben bemutatott szakértői tudásbázis leírási forma és az előzetes Q-érték meghatározási módszerek alkalmazásával. A futási eredmények a 4.1.5 fejezetben kerültek bemutatásra.
3. Az előző második eset, szakértői tudásbázis injektálásával, de az állapot-akció rácsháló elhagyásával, illetve a 4.2 és 4.3 fejezetekben bemutatott, javasolt gradiens módszer alapú hangolási eljárás és a szabálytávolság alapú szabálybázis redukálási módszerek alkalmazásával. Ez az futási eset négy további esetet takar az injektált szakértői tudásbázis milyenségétől függően, a 4.1.5 fejezetben bemutatottak alapján:
    a. helyesen definiált szakértői szabályrendszer,
    b. részben helyesen definiált szakértői szabályrendszer (a helyesen megadott szakértői szabályoknak csak egy része),
    c. részben helytelenül definiált szakértői szabályrendszer,
    d. „véletlenszerűen" generált szakértői szabályrendszer.

Az összehasonlítás alapja, a rendszer tanulási hatékonyságára jellemző változók, azaz a konvergencia sebesség (a betanuláshoz szükséges epizódok száma), a megoldás

megvalósításához szükséges lépések (step) száma (mennyi lépés alatt jut ki a völgyből az ágens) és a tudásbázis, azaz a fuzzy szabálybázis mérete.

Az első futási esetben a FRIQ-learning 29 epizód alatt konvergál és a szabálybázis 110 darab fuzzy szabályt tartalmaz (szabálybázis redukálás nélkül). Ezzel a tudásbázissal az ágens („kisautó”) 472 lépés (step) alatt jutott ki a völgyből és érte el a jobb oldali dombtetőn található sárga csillagot. A második esetben kapott futási eredményeket a 4.1.5 fejezet 10. táblázata tartalmazza.

A harmadik futási esetben négy további esetet hoztam létre a 4.1.5 fejezetben leírtaknak megfelelően, azaz helyes, részben helyes, részben helytelen, majd véletlenszerű szakértői szabályrendszerrel futott a szimuláció. A helyes szakértői szabályrendszert a 2. táblázat, a részben helyes szakértői szabályrendszer a 4.1.5 fejezetben került bemutatásra (17 darab szakértői szabályból csak 10 darab), a részben helytelen szakértői szabályrendszer helytelen szabályait a 6. táblázat, a véletlenszerűen generált szakértői szabályrendszert pedig a 8. táblázat tartalmazza.

Jelen futási esetekben alkalmazásra kerültek a javasolt gradiens módszer alapú szabálybázis hangolás (4.2 fejezet) és a szabálytávolság alapú szabálybázis redukálási (4.3 fejezet) módszerek, melyek paramétereinek értékei az alábbiak:

- gradiens módszer $\alpha = 0.01$
- új szabály felvételénél a szabályok közötti minimális szabálytávolságot meghatározó dR paraméterek értékei:
  - $\mathrm{dR}_S = \mathrm{dR}_U = 40$
- a tanulási folyamat során alkalmazott szabálytávolság alapú szabálybázis redukálási módszer dR paramétereinek értékei:
  - $\mathrm{dR}_S = 15, \mathrm{dR}_U = 15, \mathrm{dR}_q = 100$

Az egyes futási esetekben kapott eredményeket a 23. táblázat foglalja össze, ahol a 3.a.-3.d.-vel jelölt futási esetekben a javasolt szabálybázis hangolási eljárás és a szabálybázis redukálási módszerek is alkalmazásra kerültek:

23. táblázat: Az egyes futási esetekben kapott eredmények

Futási eset	Szakértői heurisztika típusa	Konvergencia sebesség (epizódok száma)	Megoldáshoz szükséges lépések száma	Szabálybázis méret (szabályok száma)
1.	üres	29	472	110
2.	a 4.1.5 fejezet 10. táblázata szerint			
3.a.	helyesen megadott	1	1199	79
3.b.	helyesen megadottnak egy része	9	1997	81
3.c.	részben helytelenül megadott	20	1554	88
3.d.	véletlenszerűen generált	37	1178	86

A táblázatban lévő futási eredmények alapján elmondható, hogy a tanulási folyamat konvergencia sebességét (és részben a végső tudásbázis méretét is) jelentős mértékben befolyásolja a szakértő által állapot-akció formátumban megadott tudásbázis helyessége. Ennek oka, hogy a helytelenül (vagy részben helytelenül) megadott előzetes tudásbázis esetében a szakértői szabályokat javítani (hangolni) kell. A szintén a szakértő által meghatározott dR paraméterek értékei alapján lettek kiszámítva a szabályfelvétel és a szabályösszevonási módszerek távolságküszöbeinek értékei, melyek a két szabály közötti minimális szabálytávolságot határozzák meg, illetve a szabályösszevonás folyamatát irányítják. Ezen paraméter értékek függvényében mindig a megfigyeléshez legközelebbi szabálypont kerül hangolása. Ezért több iterációra (ennek következtében több időre) van szükség ahhoz, hogy a helytelenül megadott szakértői szabályok hangolására is sor kerüljön.

Abban az esetben (3.a.) amikor a helyesen definiált szakértői szabályrendszerrel futott a szimuláció, akkor a rendszer jelentősen gyorsabban konvergált (egyetlen epizód alatt), mint a FRIQ-learning eredeti verziójában és a szabályszám is csökkent, 110-ről 79-re. Ennek oka, hogy a megadott szakértői tudásbázison nem kellett hangolnia a rendszernek, a szabályszámot pedig a tanulási folyamat közben alkalmazott szabálytávolság alapú szabálybázis redukciós módszer csökkentette.

Abban az esetben mikor a részben helyes szakértői szabályokkal futott a tanulási folyamat (3.b.), akkor a rendszer 9 epizód alatt konvergált 81 darab szabállyal, tehát valamennyivel több epizódra volt szükség, mint az előző futási esetben.

Mikor a részben helytelen szakértői szabályokkal futott a szimuláció (3.c), azaz több elrontott (helytelen) szabályt is tartalmazott a szakértői tudásbázis, akkor 20 epizódra volt szükség ahhoz, hogy a javasolt módszer kijavítsa a helytelenül megadott szakértői szabályokat.

Mikor a tejesen helytelen előzetes tudásbázis lett injektálva a tanulási folyamatba (3.d.), akkor is konvergált a rendszer de a 37 epizódra (epizódonként 2000 iterációra) volt szükség ahhoz, hogy a helytelen szakértői szabályok hangolása megvalósuljon.

A következő táblázatok a „véletlenszerűen” generált (helytelen) szakértői szabályrendszer szabályait tartalmazzák a tanulási fázis előtt (24. táblázat, a 4.1.5 fejezet 8. táblázata alapján) és a tanulási fázis, azaz a javasolt szabálybázis hangolási (és redukálási) módszerek alkalmazását követően (25. táblázat):

24. táblázat: A „véletlenszerűen” generált (helytelen) szakértői szabályrendszer szabályai a tanulási fázis előtt

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
s1	-0.475	-0.5	-0.475	-0.475	-0.27	-0.27	-0.27	-0.475	-0.475
s2	0	0	-0.014	0.014	0	-0.014	0	-0.042	0
a	1	-1	-1	0	-1	0	-1	1	1

R#	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
s1	-0.475	-0.065	0.14	-0.27	-0.885	0.885	-0.065	-1.09
s2	0	0	-0.014	-0.042	0.042	0.042	0.042	0.042
a	-1	0	1	-1	-1	1	0	-1

25. táblázat: A „véletlenszerűen” generált (helytelen) szakértői szabályrendszer szabályai a tanulási (hangolási) fázis után

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
s1							-0.52		-0.39		-0.21	-0.47	-0.31		0.885		-0.81
s2	x	x	x	x	x	x	-0.04	x	0.05	x	-0.03	-0.016	-0.03	x	0.042	x	0.03
a							-1		1		0	1	-1		1		-1

A 25. táblázat alapján látható, hogy a tanulási folyat során a 17 darab helytelen szakértői szabályból csupán 7 darab szakértői szabályt (7, 9, 11, 12, 13, 15, 17 sorszámú szabályok) hagyott meg a módszer, mely 7 szakértői szabály közül 6 darab állapot-akció pontját jelentősen elhangolta (optimalizálta), a 15. szakértői szabályt viszont érintetlenül hagyta. Ezek alapján megállapítható, hogy a kezdeti szakértői szabályrendszerben csak 15. sorszámú az igazoltan helyesen definiált szakértői szabály. Az „x” jelölésű szakértői szabályok a szabálybázis redukálás, azaz az a szabályösszevonások során törlésre kerültek annak következtében, hogy összeolvadtak más szakértői, vagy újonnan beszúrt szabályokkal a (61) összefüggésnek megfelelően. A szabályok száma olyan módon csökkent, hogy összeolvadtak más szabályokkal

a szabálybázis hangolás és a szabálybázis redukálás során, így a szabálybázisból elhagyott (törölt) szabályok redundáns szabályoknak tekinthetők.

A kapott futási eredmények alapján megállapítható, hogy a javasolt HFRIQ-learning rendszer (és így a javasolt szabálybázis hangolási és redukálási módszerek) alkalmazása által a rendszerbe injektált szakértői tudásbázis hangolható (korrigálható) azokban az esetekben mikor az helytelen információkat tartalmaz.

4.5.3 „Cart-Pole" alkalmazáspélda

A javasolt HFRIQ-learning rendszer működése és hatékonysága egy újabb klasszikus megerősítéses tanulási mintapéldán, a „Cart-Pole" nevezetű szimulációs példán keresztül kerül vizsgálatra. A mintapélda a disszertáció korábbi 4.1.6 fejezetében már részletesen bemutatásra került, a rendszer a futási paramétereinek értékei ($\alpha = 0.3,\ \gamma = 0.99$) nem változtak ezen fejezetben bemutatottakhoz képest. A vizsgált futási esetek megegyeznek a 4.1.6 alfejezetben részletezett futási esetekkel, de azzal az eltéréssel, hogy jelen esetben a javasolt HFRIQ-learning rendszer (a gradiens módszer alapú hangolási eljárás és a szabálytávolság alapú szabálybázis redukálási módszer) került alkalmazásra. Az injektált szakértői tudásbázis milyenségétől függően a futási esetek a következők:

1. szakértői tudásbázis nélkül (FRIQ-learning)
2. helyes szakértői szabályrendszer
3. részben helyes szakértői szabályrendszer
4. teljesen helytelen szakértői szabályrendszer

Az összehasonlítás alapja a szakértői tudásbázis nélküli 1. futási eset konvergencia sebessége és szabálybázis mérete, amikor az eredeti FRIQ-learning rendszer [97][98] 58 epizóddal és 182 darab fuzzy szabállyal konvergált (4.1.6 fejezet 14. táblázatának 1. sora). Azon esetekben kapott futási eredményeket mikor a FRIQ-learning rendszer beágyazott szakértői tudásbázissal, de a gradiens módszer alapú hangolási eljárás és a szabálytávolság alapú szabálybázis redukálási módszerek alkalmazása nélkül került futtatásra, a 4.1.6 fejezet 14. táblázata foglalja össze. A szakértői szabálybázisokat és azok megválasztásának módját szintén a 4.1.6 fejezet részletezi, a helyes szakértői szabálybázis 7 darab szakértői szabályát a 11. táblázat, a részben helyes szakértői szabálybázis szabályait az 12. táblázat, a teljesen helytelen szakértői szabálybázist pedig a 13. táblázat tartalmazza.

A szabálybázis építési folyamat során a szabályok között megengedett minimális szabálytávolság az univerzumok méretének a 10-ed része ($\mathrm{dR}_{\boldsymbol{S}} = \mathrm{dR}_{U} = 10$), a

szabálytávolság alapú szabálybázis redukálás dR paramétereinek értéke $\mathrm{dR}_S = 4$, $\mathrm{dR}_U = 5$, $\mathrm{dR}_q = 10$, a gradiens módszer alapú hangolási eljárás α tanulási ráta paraméterének értéke pedig $\alpha = 0.01$.

Az egyes futási esetekben kapott eredményeket a 26. táblázat foglalja össze:

26. táblázat: Az egyes futási esetekben kapott eredmények

Futási eset	**Szakértői heurisztika típusa**	**Konvergencia sebesség (epizódok száma)**	**Szabálybázis méret (szabályok száma)**
1.	üres	58	182
2.	helyes	2	74
3.	részben helyes	26	140
4.	teljesen helytelen	92	95

Ezen alkalmazáspélda esetében kapott futási eredmények alapján is elmondható, hogy a szakértő által megadott tudásbázis helyessége jelentős mértékben befolyásolja a HFRIQ-learning rendszer konvergencia sebességét és végső szabálybázisának méretét. Abban az esetben (2. eset) mikor helyes szakértői szabályrendszer került injektálásra akkor a rendszer 2 epizód alatt konvergált, aminek oka, hogy a rendszernek nem kellett hangolnia a megadott szakértői tudásbázison. Azokban az esetekben (3. és 4. eset) amikor helytelen szakértői szabályrendszer került injektálásra a rendszerbe akkor a javasolt hangolási módszer alkalmazása következtében a rendszernek több iterációra volt szükség a szabályok hangolásához.

A 4.1.6 fejezetben bemutattak szerint, a javasolt hangolási eljárás nélkül, ugyanezen szakértői tudásbázis esetén 350 epizód után (400 darab szabállyal) sem konvergált a rendszer, nem találta meg a megoldást leíró tudásbázist. Ezt a hibát a javasolt szabálybázis hangolási (és a szabálytávolság alapú szabálybázis redukálási) módszer kiküszöbölte.

A következő táblázatok a „véletlenszerűen” generált (helytelen) szakértői szabályrendszer szabályait tartalmazzák a tanulási fázis előtt (27. táblázat a 4.1.6 fejezet 13. táblázata alapján) és a tanulási fázis után, azaz a javasolt szabálybázis hangolási (és redukálási) módszerek alkalmazását követően (28. táblázat):

27. táblázat: A teljesen helytelen szakértői szabályrendszer szabályai a tanulási fázis előtt

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**
s_1	1	1	1	1	-1	-1	1
s_2	0	0	0	0	0	1	0
s_3	0	-0.0524	0	-0.0524	-0.2094	-0.2094	0.2094
s_4	1	-1	-1	1	-1	-1	1
a	-1	1	-1	-1	0.8	0.4	1

28. táblázat: A teljesen helytelen szakértői szabályrendszer szabályai a tanulási (hangolási) fázis után

R#	**1**	**2**	**3**	**4**	**5**	**6**	**7**
s_1	0.9340	1.0115	1.0095	0.9630	-1	-1	
s_2	-1.3334	0.5245	0.1166	-0.3350	0	1	
s_3	0.0169	-0.0173	-0.0156	0.0819	-0.2094	-0.2094	x
s_4	1.520	-0.7900	-0.1966	0.6808	-1	-1	
a	-0.9475	0.9692	-0.9108	0.975	0.8	0.4	

A 28. táblázatban látható, hogy a 7 darab helytelenül megadott szakértői szabály közül az első 4 darab (1-4) szabályt hangolta a rendszer, az 5. és 6. szabályokat változatlanul hagyta, a 7. szabály pedig összevonta más szabállyal (törölte). A tanulási folyamat ebben az esetben 92 epizódot vett igénybe és a végső szabálybázis 95 szabályt tartalmazott.

Összességében elmondható, hogy a javasolt (és fejlesztett) módszerek következtében a rendszer tanulási fázisának konvergencia sebessége javulhat, de csak olyan esetekben mikor a szakértői heurisztika helyes. Ellenkező esetben, mikor helytelen szakértői szabályok is injektálásra kerülnek, a tanulási módszer továbbra is konvergál, de a szabályok hangolásához (optimalizálásához) több epizódra (és így több iterációra) van szükség.

5 ÖSSZEFOGLALÁS

A disszertáció tárgya a gépi tanuláshoz, azon belül a megerősítéses tanulás témaköréhez kapcsolódik, amely egy Fuzzy szabály-interpoláción alapú Q-learning (FRIQ-learning) módszer szakértői tudásbázissal való kibővítését, a rendszerbe injektált szakértői tudásbázis hangolását (optimalizálását), valamint a tanulási folyamat közben létrejött tudásbázis méretének a csökkentését (redukálását) takarja.

Az értekezés első fejezetében bemutatásra került a kutatás célkitűzése, a 2.-3. fejezetekben részletezésre került a Fuzzy logika és a megerősítéses tanulás témaköre, valamint a kutatás alapjául szolgáló Fuzzy szabály-interpoláción alapú Q-learning (FRIQ-learning) módszer. A 4. fejezetben került bemutatásra a javasolt „Heurisztikusan gyorsított FRIQ-learning" (HFRIQ-learning) módszer, a hozzá kapcsolódó kutatás eredményeit összefoglaló tézisek és saját publikációk.

Első tézisként javaslatot tettem a szakértő által definiálható előzetes tudásbázis leírási módjára, valamint az a priori tudásbázist leíró fuzzy szabályrendszerre történő kezdeti Q-érték meghatározási módszerre, amely által a kezdeti szakértői tudásbázis beépíthető az FRIQ-learning módszer tanulási folyamatába. A javasolt módszerek alkalmazása által vizsgáltam továbbá az injektált szakértői tudásbázis FRIQ-learning rendszer tanulási hatékonyságára gyakorolt hatását.

Második tézisként kidolgoztam egy gradiens módszeren alapuló hangolási (optimalizálási) eljárást, amely alkalmas a tanulási folyamat közben, a tudásbázist leíró Q-függvény tartópontjainak hangolására, azaz a fuzzy szabályrendszer antecedens (állapot-akció) és konzekvens (Q-érték) értékeinek pontosítására.

Harmadik tézisként kidolgoztam egy olyan tudásbázis csökkentési (szabálybázis redukálási) módszert, amely a rendszer tanulási folyamata közben alkalmazható. A második tézisként javasolt szabálybázis hangolása során a szabálypontok vándorlása miatt előfordulhat olyan eset, amikor több szabály kerül egymáshoz közel az $(n+2)$-dimenziós térben. Abban az esetben, ha az egymáshoz közel kerülő szabályok nagyon hasonló információt írnak le (azaz ilyen szempontból redundánsak), a szabályok egyesíthetők, a szabályrendszer mérete csökkenthető. A javasolt szabálybázis redukálási módszer a rendszer tanulási fázisa közben a fuzzy szabályok közötti távolságok (és távolságküszöbök) alapján egyesíti az egymáshoz hasonló szabályokat, csökkentve a rendszer tudásbázisának méretét. Kidolgozásra került továbbá egy olyan

tudásbázis redukálási módszer is, amely hierarchikus klaszterezési eljárással a tanulási folyamatot követően képes a Q-függvényt leíró szabálybázis méretét csökkenteni.

További kutatási terv egy olyan módszer kidolgozása, amely alkalmas lehet a szakértő által megadott kezdeti, állapot-akció típusú fuzzy szabályok és a tanulási (hangolási) folyamat végeztével előállt optimalizált szakértői szabályok összehasonlítására, információt adhat arról, hogy a kezdeti szakértői szabályrendszer milyen mértékben voltak helyesek. Azaz alkalmas a kezdeti szakértői heurisztika helyességének igazolására validálására. További fejlesztési terv az előzetes szakértői tudásbázis definiálásának egyszerűsítése egy fuzzy viselkedésleíró nyelv [74] alkalmazásával. Az így kialakítandó modellek és módszerek jelentősége amellett, hogy egy nyelvi leírási formából kiindulva (például etológiai modell [85], mint a priori tudás) valamilyen rendszert közvetlenül működtető modellként használhatók, megfelelő teljesítmény mérték választása és minták megléte esetén a kezdeti szakértői heurisztika validálására is lehetőséget nyújthatnak.

A továbbiakban röviden összefoglalom a disszertáció négy fejezetében és annak alfejezeteiben részletesen bemutatott téziseket:

5.1 I. TÉZIS

A FRIQ-learning megerősítéses tanulási rendszer konvergencia sebessége javítható a kezdeti Q-érték szabálybázisba illesztett helyes szakértői produkciós szabályokból képzett Q fuzzy szabályokkal, ahol ezen beillesztett szabályok kezdeti konzekvens Q-értéke a környezet által adható maximális megerősítés érték alapján becsülhető.

I.1. Altézis: *A konvergencia sebesség az esetben is javulhat, ha a felhasznált helyes szakértői produkciós szabályok csak részben fedik le a teljes állapotteret.*

I.2. Altézis: *Amennyiben a felhasznált szakértői produkciós szabályok helytelen szabályokat is tartalmaznak, azaz egyes szabályok esetén az érintett állapotban javasolt akció választása csökkentené a várható megerősítés értékét, a teljes FRIQ-learning rendszer konvergencia sebessége romolhat.*

Az I. tézishez kapcsolódó saját publikációk: [S2], [S4], [S6], [S7], [S15]

5.2 II. TÉZIS

A HFRIQ-learning megerősítéses tanulási rendszer inkrementális szabálybázis építési fázisában a Q-függvényét leíró fuzzy szabálybázis szabályainak (fuzzy Q-szabályok) antecedensei és konzekvensei gradiens módszerrel optimalizálhatók, hangolhatók. Az aktuális

állapot-akció pontban egy új szabály beillesztésének feltétele a már meglévő szabályoktól vett távolsága és a Q-függvény frissítésének mértéke alapján meghatározható.

II.1. Altézis: *Amennyiben nincs olyan fuzzy Q-szabály, melynek antecedense valamennyi antecedens dimenzióban vett távolsága kisebb az egyes dimenziókra meghatározott távolságküszöbnél és a Q-függvény frissítésének mértéke nagyobb, mint egy küszöbérték, úgy az aktuális állapot-akció pontba egy új szabály kerül beillesztésre. Ellenkező esetben a meglévő fuzzy Q-szabályok kerülnek hangolásra.*

II.2. Altézis: *Abban az esetben, ha szabálybázisba illesztett kezdeti szakértői szabályrendszer helytelen szakértői produkciós szabályokat is tartalmaz, akkor azok a tanulási fázisban a szabálybázis többi szabályával együtt hangolhatók, korrigálhatók.*

II.3. Altézis: *A HFRIQ-learning megerősítéses tanulási rendszer tudásbázisának hangolása során az állapot-akció tér ritkán bejárt területein lévő fuzzy Q-szabályok elhangolódása csökkenthető, ha az összes fuzzy szabálypont egyidejű hangolása helyett, csak azon szabályok kerülnek hangolásra, amely az éppen aktuális állapot-akció megfigyelési pont közelében találhatók.*

A II. tézishez kapcsolódó saját publikációk: [S1], [S9], [S10], [S12]

5.3 III. TÉZIS

A HFRIQ-learning megerősítéses tanulási módszer tudásbázisának mérete a tanulási folyamat során csökkenthető a hasonló fuzzy Q-szabályok összevonásával. A fuzzy szabályok hasonlósága becsülhető antecedenseik és konzekvenseik távolságával.

III.1. Altézis: *A szabályok hangolása és összevonása során követhető a szabályok típusa, így a tanulási folyamat végeztével a HFRIQ-learning Q fuzzy szabályrendszeréből a kezdeti szakértői szabályok visszanyerhetők. Az eredetileg megadott és a hangolást követően visszanyert szakértői szabályok összevetésével következtetni lehet a kezdeti szakértői szabályok helyességére.*

III.2. Altézis: *A hierarchikus klaszterezési módszer alkalmas a HFRIQ-learning tanulási fázisának végeztével előállt Q-függvényt leíró fuzzy szabályrendszer méretének csökkentésére.*

A III. tézishez kapcsolódó saját publikációk: [S3], [S5], [S8], [S13]

6 SUMMARY

The subject of the dissertation is related to the machine learning area, especially to the reinforcement learning, which includes extending the Fuzzy Rule Interpolation-based Q-learning (FRIQ-learning) method by embedding a priory expert knowledge, tuning the injected expert knowledge (optimization), and reducing the size of the rule-base representing the knowledge base during and after the incremental learning process.

In the first section of the dissertation, the main goal of the study is summarized and the related literature is reviewed. The next two chapters briefly introduce the fuzzy logic, the reinforcement learning and the Fuzzy Rule Interpolation-based Q-learning (FRIQ-learning). The fourth section introduce the suggested "Heuristically Accelerated FRIQ-learning" method, the related theses and our papers supporting them.

In the first thesis, I proposed a way to describe the preliminary (a prior) knowledge base which is defined by a human expert as a set of production (state-action) fuzzy rules. I also suggest a way for embedding the expert defined rules to the initial rule-base of the FRIQ-learning, by reformulation the state-action fuzzy rules to state-action-Q-value fuzzy rules required by the FRIQ-learning. The way as the required initial Q-values of the rules are determined is also part of the firs thesis. Furthermore, I investigated the effect of the injected expert knowledge base on the learning efficiency of the FRIQ-learning system applying the proposed methods.

My second thesis, is a gradient descent-based optimization method for optimizing the antecedents (state-action) and consequent (Q-value) parameters of the fuzzy rule-base describing the Q-function.

My third thesis, is a Q-function fuzzy rule-base reduction method, based on merging the similar rules during the learning phase. During the learning phase, rule antecedents could shift to be close to each other, and in the case if the rule consequents are also close, the rules are similar. This case the similar rules can be merged to reduce the rule-base size with a negligible change in the Q-function described. This thesis also describes a method based on hierarchical clustering for reducing the Q-function fuzzy rule-base size after the learning phase.

Further research will focus on the development of a method for validating the preliminary expert knowledge base by comparing the initial expert fuzzy rules with the rules fetched back form the optimized Q-function fuzzy rule-base after the learning phase. Another future goal is

to apply a fuzzy behavior description language (FBDL, introduced in [74]), for simplifying and standardizing the form as the initial state-action fuzzy expert rules are specified. These models and methods could support the validation of practical heuristic expert models like ethological models [85].

In the following, I repeat the three theses of the dissertation.

6.1 THESIS I.

The convergence speed of the FRIQ-learning reinforcement learning system can be improved by embedding Q fuzzy rules created from valid expert production rules into the initial Q-value rule-base, where the initial consequent Q-values of the embedded rules can be estimated based on the maximum reinforcement value that can be provided by the environment.

I.1. Subthesis: *The convergence speed can be improved even if the embedded valid expert production rules are only partly covering the complete state space.*

I.2. Subthesis: *If the expert production rules are also containing invalid rules, i.e., in some rules, the action suggested in the affected state would decrease the expected reinforcement value, it may negatively effects the convergence speed of the entire FRIQ-learning system.*

My publications related to Thesis I: [S2], [S4], [S6], [S7], [S15]

6.2 THESIS II.

In the incremental rule-base construction phase of the HFRIQ-learning reinforcement learning system, the antecedents and consequents of the rules of the fuzzy rule-base describing the Q-function (fuzzy Q-rules) can be optimized and tuned using a gradient based method. The conditions for inserting a new rule at the current state-action point can be determined based on the distance from the existing rules and the extent of the Q-function update.

II.1. Subthesis: *If there is no fuzzy Q-rule, whose distances in all antecedent dimensions are smaller than the distance thresholds determined for each dimensions and the Q-function update is also greater than a threshold value, then a new rule is inserted to the current state-action position. Otherwise, the existing fuzzy Q-rules are tuned.*

II.2. Subthesis: *If the initial expert rule system, which is embedded into the rule-base also contains incorrect expert production rules, then they can be tuned and corrected together with the other rules of the rule-base during the learning phase.*

II.3. Subthesis: *During the tuning of the knowledgebase of the HFRIQ-learning reinforcement learning system, the detuning of fuzzy Q-rules in the sparsely explored areas of the state-action*

space can be reduced by tuning the rules that are located close to the current state-action observation point only, instead of the simultaneous tuning of all the fuzzy rule points.

My publications related to Thesis II: [S1], [S9], [S10], [S12]

6.3 THESIS III.

During the learning process the size of the knowledgebase of the HFRIQ-learning reinforcement learning method can be reduced by merging the similar fuzzy Q-rules. The similarity of fuzzy rules can be estimated by the distance between their antecedents and consequents.

III.1. Subthesis: *During the tuning and merging of rules, the type of the rules can be tracked, thus at the end of the learning process from the Q fuzzy rule system of the HFRIQ-learning, the initial expert rules can be recovered. By comparing the originally defined and the recovered expert rules after tuning, it is possible to evaluate the correctness of the initial expert rules.*

III.2. Subthesis: *At the end of the HFRIQ-learning phase, the hierarchical clustering method is suitable for reducing the size of the fuzzy rule-base describing the Q-function.*

My publications related to Thesis III: [S3], [S5], [S8], [S13]

7 IRODALOMJEGYZÉK

[1] Almadi, A. I., Al Mamlook, R. E., Almarhabi, Y., Ullah, I., Jamal, A., & Bandara, N. (2022). A fuzzy-logic approach based on driver decision-making behavior modeling and simulation. Sustainability, 14(14), 8874.

[2] Appl, M.: Model-based Reinforcement Learning in Continuous Environments. Ph.D. thesis, Technical University of München, München, Germany, dissertation.de, Verlag im Internet (2000)

[3] Arulkumaran, Kai, et al. "Deep reinforcement learning: A brief survey." *IEEE Signal Processing Magazine* 34.6 (2017): 26-38.

[4] Baranyi, P., Kóczy, L. T., Gedeon, T. D.: A Generalized Concept for Fuzzy Rule Interpolation, IEEE Trans. on Fuzzy Systems, vol. 12, No. 6, 2004, pp 820-837.

[5] Baranyi, P., Kóczy, L. T.: A General and Specialised Solid Cutting Method for Fuzzy Rule Interpolation, in Journal BUSEFAL, URA-CNRS, Vol. 66., Toulouse, France, 1996, pp. 13-22.

[6] Baranyi, P., Mizik, S., Kóczy, L.T., Gedeon, T. and Nagy, I.: Fuzzy Rule Base Interpolation Based on Semantic Revision, in Proceedings of the IEEE International Conference on System Man and Cybernetics (IEEE SMC'98), San Diego, USA, 1998, pp.1306-1311.

[7] Bartók, Roland, and József Vásárhelyi. "A fuzzy rule interpolation base algorithm implementation on different platforms." *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*. IEEE, 2015.

[8] Bartók, Roland, and József Vásárhelyi. "Fuzzy Rule Interpolation Based Object Tracking and Navigation for Social Robot." *Vehicle and Automotive Engineering*. Springer, Cham, 2018.

[9] Bartók, Roland, and József Vásárhelyi. "Parallelization of FIVE method on multicore embedded system." *2018 19th International Carpathian Control Conference (ICCC)*. IEEE, 2018.

[10] Bellman, R. (1957). A Markovian decision process. Journal of mathematics and mechanics, 679-684.

[11] Bellman, R. E.: Dynamic Programming. Princeton University Press, Princeton, NJ 1957

[12] Bellman, Richard, and Robert Kalaba. "On the role of dynamic programming in statistical communication theory." IRE Transactions on Information Theory 3.3 (1957): 197-203.

[13] Berenji, H.R.: Fuzzy Q-Learning for Generalization of Reinforcement Learning. Proc. of the 5th IEEE International Conference on Fuzzy Systems, pp. 2208-2214., 1996

[14] Bianchi, Reinaldo AC, Carlos HC Ribeiro, and Anna Helena Reali Costa. "Heuristically Accelerated Reinforcement Learning: Theoretical and Experimental Results." ECAI. 2012.

[15] Bianchi, Reinaldo AC, Carlos HC Ribeiro, and Anna HR Costa. "Accelerating autonomous learning by using heuristic selection of actions." *Journal of Heuristics* 14.2 (2008): 135-168.

[16] Bianchi, Reinaldo AC, Carlos HC Ribeiro, and Anna HR Costa. "Heuristically Accelerated Q–Learning: a new approach to speed up Reinforcement Learning." Brazilian Symposium on Artificial Intelligence. Springer, Berlin, Heidelberg, 2004.

[17] Bonaccorso, Giuseppe. Machine learning algorithms. Packt Publishing Ltd, 2017.

[18] Bonarini, A.: Delayed Reinforcement, Fuzzy Q-Learning and Fuzzy Logic Controllers. In Herrera, F., Verdegay, J. L. (Eds.) Genetic Algorithms and Soft Computing, (Studies in Fuzziness, 8), Physica-Verlag, Berlin, D, (1996), pp. 447-466.

[19] Broekens, Joost, Koen Hindriks, and Pascal Wiggers. "Reinforcement learning as heuristic for action-rule preferences." *International Workshop on Programming Multi-Agent Systems*. Springer Berlin Heidelberg, 2010.
[20] Brunton, Steven L., and J. Nathan Kutz. Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press, 2019.
[21] Brys, Tim. *Reinforcement Learning with Heuristic Information*. Diss. PhD thesis, PhD thesis, Vrije Universitet Brussel, 2016.
[22] Chaudhari, Swati R., and Manoj E. Patil. "Comparative analysis of fuzzy inference systems for air conditioner." International Journal of Advanced computer research 4.4 (2014): 922.
[23] Chiou, C. B., Chiou, C. H., Chu, C. M., & Lin, S. L. (2009). The application of fuzzy control on energy saving for multi-unit room air-conditioners. Applied thermal engineering, 29(2-3), 310-316.
[24] Csaba, Johanyák Zsolt, and Kovács Szilveszter. "A fuzzy tagsági függvény megválasztásáról." *A GAMF közleményei, Kecskemét, XIX. évfolyam, ISSN*: 0230-6182.
[25] D. Shepard, "A two dimensional interpolation function for irregularly spaced data", Proc. 23rd ACM Internat. Conf., 1968, pp. 517-524.
[26] Dubios, D., Ostasiewicz, W., Prade, H.: Fuzzy Sets: History and Basic Notions, in: Fundamentals of Fuzzy Sets, ISBN 978-0-7923-7732-0, Kluwer Academic, 2000
[27] Duchi, John, Elad Hazan, and Yoram Singer. "Adaptive subgradient methods for online learning and stochastic optimization." Journal of machine learning research 12.7 (2011).
[28] FIVE FRI MATLAB Toolbox: http://fri.uni-miskolc.hu/
[29] François-Lavet, Vincent, et al. "An introduction to deep reinforcement learning." *arXiv preprint arXiv:1811.12560* (2018).
[30] 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.
[31] G. Tesauro et al. Temporal difference learning and td-gammon. Communications of the ACM, 38(3):58–68, 1995.
[32] Glorennec, P. Y., & Jouffe, L. (1997, July). Fuzzy Q-learning. In Proceedings of 6th international fuzzy systems conference (Vol. 2, pp. 659-662). IEEE.
[33] Hailu, Getachew, and Gerald Sommer. "Embedding knowledge in reinforcement learning." *International Conference on Artificial Neural Networks. Springer*, London, 1998.
[34] Haji, Saad Hikmat, and Adnan Mohsin Abdulazeez. "Comparison of optimization techniques based on gradient descent algorithm: A review." PalArch's Journal of Archaeology of Egypt/Egyptology 18.4 (2021): 2715-2743.
[35] Hindriks, K. V., De Boer, F. S., Van Der Hoek, W., & Meyer, J. J. C. (2000, July). Agent programming with declarative goals. In *International Workshop on Agent Theories, Architectures, and Languages* (pp. 228-243). Springer, Berlin, Heidelberg.
[36] Horiuchi, T., Fujino, A., Katai, O., & Sawaragi, T. (1996, September). Fuzzy interpolation-based Q-learning with continuous states and actions. In *Proceedings of IEEE 5th International Fuzzy Systems* (Vol. 1, pp. 594-600). IEEE.
[37] J. Dombi. A general class of fuzzy operators, the De Morgan class of fuzzy operator and fuzziness measures induced by fuzzy operators. Fuzzy Sets and Systems, 8(2):149–163, 1982.
[38] Jenei, S.: Interpolation and Extrapolation of Fuzzy Quantities revisited - (I). An Axiomatic Approach, in Soft Computing, ISSN: 1432-7643, Vol. 5, 2001, pp. 179-193.
[39] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy Rule Interpolation Based on Polar Cuts, in Computational Intelligence, Theory and Applications, Springer Berlin Heidelberg, 2006, pp. 499-511.

[40] Johanyák, Zs. Cs. and Kovács, Sz.: Fuzzy Rule Interpolation by the Least Squares Method, in Proceedings of the 7th International Symposium of Hungarian Researchers on Computational Intelligence (HUCI 2006), Budapest, Hungary, 2006, pp. 495-506.

[41] Johanyák, Zs. Cs. and Kovács, Sz.: Vague Environment-based Two-step Fuzzy Rule Interpolation Method, in Proceedings of the 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics (SAMI 2007), Poprad, Slovakia, 2007, pp. 189-200.

[42] Johanyák, Zsolt Csaba, and Szilveszter Kovács. "A brief survey and comparison on various interpolation based fuzzy reasoning methods." *Acta Polytechnica Hungarica* 3.1 (2006): 91-105.

[43] Jose Antonio Martin H. PhD, Software Tools for Reinforcement Learning, Artificial Neural Networks and Robotics (Matlab and Python)

[44] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-international conference on neural networks. Vol. 4. IEEE, 1995.

[45] Kim, M. S., Hong, G. G., & Lee, J. J. (1999, October). Online fuzzy Q-learning with extended rule and interpolation technique. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289)* (Vol. 2, pp. 757-762). IEEE.

[46] King, P.J., Mamdani, E.H.: The application of fuzzy control systems to industrial processes, Automatica, Vol. 13, Issue 3, May 1977, pp. 235–242.

[47] Klawonn, F.: Fuzzy Sets and Vague Environments, in Fuzzy Sets and Systems, Vol. 66, 1994, pp. 207-221.

[48] Kochenderfer, Mykel J., and Tim A. Wheeler. *Algorithms for optimization*. Mit Press, 2019.

[49] Kóczy, L. T. and Hirota, K.: Size reduction by interpolation in fuzzy rule bases, IEEE Transactions on Systems, Man, and Cybernetics, vol. 27, 14 - 25, 1997.

[50] Kóczy, L. T., Hirota, K., Rule interpolation by α-level sets in fuzzy approximate reasoning, In J. BUSEFAL, Automne, URA-CNRS. Vol. 46. Toulouse, France, 1991, pp. 115-123.

[51] Kóczy, L. T., Sugeno, M.: Explicit functions of fuzzy control systems, in International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 4:515-535, 1996.

[52] Kóczy, L.T., Hirota, K. and Gedeon, T. D.: Fuzzy rule interpolation by the conservation of relative fuzziness, in Journal of Advanced Computational Intelligence, Vol. 4/1, 2000, pp. 95-101.

[53] Kóczy, László T., and Domonkos Tikk. "Fuzzy rendszerek." TypoTEX, Budapest (2000).

[54] Kovács, Sz., Kóczy, L. T.: Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI. Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, 1997, pp. 144-149.

[55] 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.

[56] Kovács, Sz.: Extending the Fuzzy Rule Interpolation 'FIVE' by Fuzzy Observation, Advances in Soft Computing, Computational Intelligence, Theory and Applications, Bernd Reusch (Ed.), Springer Germany, ISBN 3-540-34780-1, 2006, pp. 485-497.

[57] Kovács, Sz.: New Aspects of Interpolative Reasoning. Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 477-482.
[58] Kovacs, Szilveszter. "Fuzzy Rule Interpolation in Practice." *SCIS & ISIS SCIS & ISIS 2006*. Japan Society for Fuzzy Theory and Intelligent Informatics, 2006.
[59] Kovács, Szilveszter: Fuzzy logic control, M.Phil. theses, Technical University of Budapest, Faculty of Informatics and Electrical Engineering, Budapest, Branch of Computer Science, p.116, (1993).
[60] L. T. Kóczy, Computational complexity of various fuzzy inference algorithms, Annales Univ. Sci. Budapest, Sect. Comp. 12, pp 151-158, (1991)
[61] Larsen, P. M.: Industrial application of fuzzy logic control. Int. J. of Man Machine Studies, (12) 4, 3-10., 1980
[62] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444.
[63] Li, Yuxi. "Deep reinforcement learning: An overview." *arXiv preprint arXiv:1701.07274* (2017).
[64] Mamdani, E. H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. of Man Machine Studies, (7), 1 -13., 1975
[65] Matignon, Laëtitia, Guillaume J. Laurent, and Nadine Le Fort-Piat. "Reward function and initial values: better choices for accelerated goal-directed reinforcement learning." *International Conference on Artificial Neural Networks*. Springer, Berlin, Heidelberg, 2006.
[66] Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research, 134, 105400.
[67] Mesiar, R.: Triangular Norms - An Overview, Computational Intelligence in Theory and Practice, Advances in Soft Computing Vol. 8, 2001, pp. 35-54.
[68] Mitchell, T. M., & Mitchell, T. M. (1997). Machine learning (Vol. 1, No. 9). New York: McGraw-hill.
[69] Mizik, S., Baranyi, P., Korondi, P. and Kóczy, L.T.: Comparison of fuzzy interpolation techniques, 4th Meeting of the Euro Working Group on Fuzzy Sets and 2nd International Conference on Soft and Intelligent Computing (EUROFUSE-SIC'99), 1999, Budapest, pp.544-549.
[70] Mizik, S., Szabó, D. and Korondi, P.: Survey on fuzzy interpolation techniques, in Proceedings of the IEEE International Conference on Intelligent Engineering Systems, Poprad, Slovakia, 1999, pp. 587–592.
[71] Murphy, R. R. (2001). Fuzzy logic for fusion of tactical influences on vehicle speed control. Fuzzy logic techniques for autonomous vehicle navigation, 73-96.
[72] Oh, Chi-Hyon, Tomoharu Nakashima, and Hisao Ishibuchi. "Initialization of Q-values by fuzzy rules for accelerating Q-learning." *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*. Vol. 3. IEEE, 1998.
[73] Oh, Chi-Hyon, Tomoharu Nakashima, and Hisao Ishibuchi. "Initialization of Q-values by fuzzy rules for accelerating Q-learning." *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*. Vol. 3. IEEE, 1998.
[74] Piller, Imre, and Szilveszter Kovács. "FBDL: A Declarative Language for Interpolative Fuzzy Behavior Modeling." *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*. IEEE, 2019.

[75] Pourhassan, Mojgan, and Nasser Mozayani. "Incorporating expert knowledge in Q-learning by means of fuzzy rules." *Computational Intelligence for Measurement Systems and Applications, 2009. CIMSA'09. IEEE International Conference on*. IEEE, 2009.
[76] R. R. Yager. On the measure of fuzziness and negation. part ii: Lattices. Information and Control, 44(3):236–260, 1980.
[77] R. S. Sutton. Learning to predict by the methods of temporal differences. Machine learning, 3(1):9–44, 1988
[78] Ribeiro, Carlos HC. "Embedding a priori knowledge in reinforcement learning." *Journal of Intelligent and Robotic Systems* 21.1 (1998): 51-71.
[79] Rummery, G. A., Niranjan, M.: On-line Q-learning using connectionist systems. CUED/F-INFENG/TR 166, Cambridge University, UK., 1994
[80] Russell Stuart, J., & Norvig, P. (2009). Artificial intelligence: a modern approach. Prentice Hall.
[81] S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. Machine learning, 22(1):33–57, 1996.
[82] Santra, Santanu, Jun-Wei Hsieh, and Chi-Fang Lin. "Gradient descent effects on differential neural architecture search: A survey." IEEE Access 9 (2021): 89602-89618.
[83] Sugeno, M.: An introductory survey of fuzzy control. Information Science, (36), 1985, pp. 59-83.
[84] Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction, MIT Press, Cambridge (1998)
[85] Sz. Kovács, D. Vincze, M. Gácsi, Á. Miklósi, P. Korondi, "Ethologically inspired robot behavior implementation", Proc. 4th International Conference on Human System Interaction (HSI 2011), Keio University, Yokohama, Japan, 2011, pp. 64–69.
[86] Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. on SMC, (15), 1985, pp. 116-132.
[87] Tan, Ming. "Multi-agent reinforcement learning: Independent vs. cooperative agents." *Proceedings of the tenth international conference on machine learning*. 1993.
[88] Tesauro, G.: Temporal difference learning and TD-Gammon, in Communications of the ACM, 1995, 38.3: pp. 58-68.
[89] Tikk, D. and Baranyi, P.: Comprehensive analysis of a new fuzzy rule interpolation method, in IEEE Transactions on Fuzzy Systems, Vol. 8, 2000, pp. 281-296.
[90] Tikk, D., Joó, I., Kóczy, L., Várlaki, P., Moser, B., & Gedeon, T. D. (2002). Stability of interpolative fuzzy KH controllers. Fuzzy Sets and Systems, 125(1), 105-119.
[91] Torrey, Lisa, and Jude Shavlik. "Transfer learning." Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, 2010. 242-264.
[92] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529, 2015
[93] Vass, G., Kalmár, L. and Kóczy, L. T.: Extension of the fuzzy rule interpolation method, in Proceedings of the International Conference on Fuzzy Sets Theory Applications (FSTA '92), Liptovsky Mikulas, Czechoslovakia, 1992, pp. 1-6.
[94] Vincze, D., Kovács, Sz.: Incremental Rule Base Creation with Fuzzy Rule Interpolation-Based Q-Learning, I. J. Rudas et al. (Eds.), Computational Intelligence in Engineering, Studies in Computational Intelligence, Volume 313/2010, Springer-Verlag, Berlin Heilderberg, 2010, pp. 191-203.
[95] Vincze, D., Kovács, Sz.: Reduced Rule Base in Fuzzy Rule Interpolation-based Q-learning, Proceedings of the 10th International Symposium of Hungarian Researchers on

Computational Intelligence and Informatics, CINTI 2009, November 12-14, 2009, Budapest Tech, Budapest, pp. 533-544.
[96] Vincze, D., Kovács, Sz.: Rule-Base Reduction in Fuzzy Rule Interpolation-Based Q-Learning, Recent Innovations in Mechatronics (RIiM) Vol. 2. (2015) No. 1-2.
[97] Vincze, Dávid, and Szilveszter Kovács. "Fuzzy rule interpolation-based Q-learning." 2009 5th International Symposium on Applied Computational Intelligence and Informatics. IEEE, 2009.
[98] Vincze, Dávid.: "Fuzzy Rule Interpolation-based Q-learning." PhD dissertation, 2013.
[99] Watkins, C. J. C. H., Dayan, P.: Q-learning, in Machine Learning, Vol. 8 (3/4), 1992., pp. 279-292.
[100] Wong, K. W., Gedeon, T. D. and Tikk, D.: An improved multidimensional α-cut based fuzzy interpolation technique, In Proc. Int. Conf. Artificial Intelligence in Science and Technology (AISAT'2000) , Hobart, Australia, 2000, pp. 29–32.
[101] Yan, S., Mizumoto, M. and Qiao, W. Z.: An Improvement to Kóczy and Hirota's Interpolative Reasoning in Sparse Fuzzy Rule Bases, in International Journal of Approximate Reasoning, Vol. 15, 1996, pp. 185-201.
[102] Zadeh, L. A.: Fuzzy Sets, in Information and Control, Vol. 8, 1965, pp. 338-353.
[103] Zadeh, L. A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on SMC, (3), 1973, pp. 28-44.

Hivatkozások ellenőrzésének utolsó dátuma: 2023.05.10.

8 SAJÁT PUBLIKÁCIÓK

[S1] Tamás, Tompa, and Kovács Szilveszter. "Expert heuristic tuning design for the FRIQ-learning." Multidiszciplináris Tudományok 10.4 (2020): 119-125.

[S2] Tamás, Tompa, and Kovács Szilveszter. "Heurisztikusan gyorsított megerősítéses tanulási módszerek-áttekintés." Multidiszciplináris Tudományok 10.3 (2020): 394-402.

[S3] Tamás, Tompa, and Kovács Szilveszter. "Szabálytávolság alapú szabálybázis redukció a szakértői tudásbázissal bővített FRIQ-learning környezetben." Multidiszciplináris Tudományok 12.1 (2022): 90-102.

[S4] Tamás, Tompa, and Kovács Szilveszter. "Szakértői heurisztika alkalmazása a FRIQ-learning megerősítéses tanulási módszerben." Multidiszciplináris Tudományok 9.4 (2019): 356-368.

[S5] Tamás, Tompa, and Kovács Szilveszter. "Tudásbázis redukció a szakértői szabályrendszerrel bővített FRIQ-learning módszerben." Multidiszciplináris Tudományok 11.4 (2021): 70-80.

[S6] Tompa, T., Kovács, S., Vincze, D., & Niitsuma, M. (2021, January). Demonstration of expert knowledge injection in Fuzzy Rule Interpolation based Q-learning. In 2021 IEEE/SICE International Symposium on System Integration (SII) (pp. 843-844). IEEE.

[S7] Tompa, Tamás, and Szilveszter Kovács. "Applying Expert Heuristic as an a Priori Knowledge for FRIQ-Learning." Acta Polytechnica Hungarica 17.4 (2020).

[S8] Tompa, Tamás, and Szilveszter Kovács. "Clustering-based fuzzy knowledgebase reduction in the FRIQ-learning." 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI). IEEE, 2017.

[S9] Tompa, Tamás, and Szilveszter Kovács. "Determining the minimally allowed rule-distance for the incremental rule-base contruction phase of the FRIQ-learning." 2018 19th International Carpathian Control Conference (ICCC). IEEE, 2018.

[S10] Tompa, Tamás, and Szilveszter Kovács. "Heuristically accelerated FRIQ-learning." 2022 IEEE 20th Jubilee International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 2022.

[S11] Tompa, Tamás, and Szilveszter Kovács. "Q-learning vs. FRIQ-learning in the Maze problem." 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom). IEEE, 2015.

[S12] Tompa, Tamás, and Szilveszter Kovács. "TUDÁSBÁZIS HANGOLÁSA A FRIQ-LEARNING MEGERŐSÍTÉSES TANULÁSI RENDSZERBEN." Production Systems and Information Engineering 10.4 (2022): 32-41.

[S13] Tompa, Tamás, and Szilveszter Kovács. "Tudásbázis redukálás a heurisztikusan gyorsított FRIQ-learning rendszerben." Production Systems and Information Engineering 11.2 (2023): 1-12.

[S14] Tompa, Tamás, Dávid Vincze, and Szilveszter Kovács. "The Pong game implementation with the FRIQ-learning reinforcement learning algorithm." Proceedings of the 2015 16th International Carpathian Control Conference (ICCC). IEEE, 2015.

[S15] Tompa, Tamás; Kovács, Szilveszter. „Szakértői tudás alapú FRIQ-learning”, Nemzetközi Energetika-Elektrotechnika Konferencia SzámOkt 2018 XXVIII. Nemzetközi Számítástechnika és Oktatás Konferencia Erdélyi Magyar Műszaki Tudományos Társaság (EMT), (2018) pp. 320-325., 4 p.

Publikációs statisztika (2023.05.10.):
Publikációk száma: 15, Hivatkozások száma: 65 (ebből független: 17)