

Egységes megerősítéses tanulási keretrendszer adaptív rajintelligencia-algoritmusokhoz vezeték nélküli szenzorhálózatokban

A Unified Reinforcement Learning Framework for Adaptive Swarm Intelligence in Wireless Sensor Networks

KHELIFI Wajih¹, Prof. Dr. GÁL Zoltán²

¹Debreceni Egyetem, Informatikai Kar, Információs Rendszerek és Hálózatok Tanszék,
4028 Debrecen, Kassai út 26.,

{¹khelifi.wajih, ²gal.zoltan}@inf.unideb.hu

Abstract

This paper presents RL-SI, a network-aware reinforcement-learning framework for controlling ABC, continuous ACO, GWO, and PSO during energy-aware cluster-head selection in wireless sensor networks. Algorithm-specific Double-DQN policies employ a shared 13-variable state representation and six control actions. Fixed and RL-guided variants are evaluated in 120 simulations covering networks of 50, 250, and 1000 nodes and five layouts per configuration. RL-SI most consistently improves packet-delivery reliability and often reduces optimizer runtime, particularly at larger scales, while its effects on network lifetime and energy preservation remain algorithm-dependent.

Keywords: Wireless Sensor Networks; Reinforcement Learning; Swarm Intelligence; Cluster-Head Selection; Energy Efficiency

Kivonat

A tanulmány az RL-SI, a hálózat állapotát figyelembe vevő megerősítéses tanulási keretrendszert mutatja be az ABC, a folytonos ACO, a GWO és a PSO energiahatékony klaszterfej-kiválasztásának vezérlésére vezeték nélküli szenzorhálózatokban. Az algoritmus-specifikus Double-DQN szabályozók közös, 13 változós állapotleírást és hat vezérlési műveletet alkalmaznak. A fix és az RL-vezérelt változatokat 120 szimulációban, 50, 250 és 1000 csomópontos hálózatokon értékeltük. Az RL-SI legkövetkezetesebben a csomagkézbesítés megbízhatóságát javította, és gyakran csökkentette a futási időt, míg az élettartamra és az energiamegőrzésre gyakorolt hatása algoritmusfüggő maradt.

Kulcsszavak: vezeték nélküli szenzorhálózatok; megerősítéses tanulás; rajintelligencia; klaszterfej-kiválasztás; energiahatékonyaság

1. BEVEZETÉS

A vezeték nélküli szenzorhálózatok (WSN-ek) többek között környezeti megfigyelési, ipari automatizálási, egészségügyi és dolgok internete alkalmazásokat támogatnak. A szenzorcsomópontok rendszerint korlátozott, nem újratölthető energiaforrással működnek; ezért a nem hatékony kommunikáció, a kedvezőtlenül elhelyezett klaszterfejek (CH-k) és a kiegyensúlyozatlan továbbítási terhelés lerövidítheti a hálózat élettartamát.

A klaszterezés csökkenti a kommunikáció energiaigényét, mivel a klaszterfejek összegyűjtik és aggregálják a tagsomópontok adatait, majd továbbítják azokat a nyelőcsomópontnak [1]. A megfelelő klaszterfejek kiválasztásakor azonban figyelembe kell venni azok számát, helyét, kommunikációs távolságát és maradék energiáját. Ez különösen nagy hálózatokban nagyméretű kombinatorikus optimalizálási problémát eredményez. A részecske-raj-optimalizálást (PSO), a mesterséges méhkolónia-algoritmust (ABC), a hangyakolónia-optimalizálást (ACO) és a szürkefarkas-optimalizálót (GWO) ezért széles körben alkalmazzák energiahatékony WSN-klaszterezésre és útválasztásra [2]–[5]. Hatékonyságuk nagymértékben függ a feltárás és a kiaknázás egyensúlyától. A hagyományos megvalósítások rögzített paramétereket vagy előre meghatározott

ütemezéseket használnak, amelyek nem reagálnak közvetlenül a fitness javulására, a populáció diverzitására, a keresés stagnálására vagy a hálózat állapotváltozásaira.

A tanulmány az RL-SI elnevezésű, hálózatállapot-tudatos, megerősítéses tanulással támogatott keretrendszert javasolja az ABC, a folytonos ACO, a GWO és a PSO dinamikus vezérlésére a klaszterfejek kiválasztása során. Minden optimalizáléhoz külön Double-DQN vezérlési politika tartozik, amely minden WSN-epoch elején egy közös, 13 változós Markov-állapotot figyel meg. Az állapot a hálózati jellemzőket az aktuális optimalizálási mutatókkal egyesíti. A vezérlő hat művelet közül választ: a keresés változatlan folytatása, a feltárás növelése, a kiaknázás növelése, a legrosszabb megoldások diverzifikálása, a legrosszabb megoldások újraindítása vagy helyi finomítás alkalmazása. A műveletek jelentése közös, de megvalósításuk optimalizálóspecifikus, így az eredeti keresési mechanizmusok változatlanok maradnak.

A tanulmány fő hozzájárulása egy egységes, tanult keresésvezérlési felület strukturálisan eltérő rajoptimalizálókhoz. Hatását az F-SI és az RL-SI azonos feltételek melletti összehasonlításával különítjük el. A párosított kiértékelés 50, 250 és 1000 csomópontos hálózatokat, valamint öt független elrendezést foglal magában. Összesen 120 szimuláció vizsgálja, hogy a tanult keresésvezérlés következetes előnyt biztosít-e különböző algoritmusok és hálózatméretek esetén.

2. SZAKIRODALMI ÁTTEKINTÉS

A klaszterezést széles körben alkalmazzák a vezetékek nélküli szenzorhálózatok kommunikációs energiaigényének csökkentésére. A Low-Energy Adaptive Clustering Hierarchy (LEACH) véletlenszerű klaszterfejróciót vezetett be az energiaigényes szerep csomópontok közötti elosztására [1]. Valószínűségi kiválasztása azonban nem optimalizálja közvetlenül a kommunikációs távolságot, a maradékenergia-egyensúlyt vagy a klaszterek számát. Latiff és munkatársai PSO-alapú, központosított, energiahatékony klaszterezést javasoltak klaszteren belüli távolság- és hálózatienergia-kritériumok alkalmazásával [2]. Li és munkatársai olyan ACO-alapú útválasztást dolgoztak ki, amely módosított útvonalfelderítéssel és feromonfrissítéssel egyensúlyozza a forgalmat és az energiafogyasztást [3]. Wang és munkatársai továbbfejlesztett ABC-protokollt alkalmaztak energiahatékony WSN-útválasztásra [4], míg Kaddi és munkatársai a GWO-t továbbfejlesztett közeg-hozzáférési mechanizmussal kombinálták [5]. Bár ezek a módszerek javítják a klaszterfej-kiválasztást, az útválasztást és a hálózati élettartamot, keresésüket általában rögzített paraméterek, előre meghatározott ütemezések vagy algoritmus-specifikus szabályok vezérlik.

A megerősítéses tanulás lehetővé teszi az optimalizálási folyamat állapotfüggő vezérlését. Durgut és munkatársai RL-alapú adaptív operátorkiválasztást építettek az ABC-be [6], Zhao és munkatársai pedig központi vezérlővel rendelkező, többágenses RL-vezérelt ABC-módszert javasoltak [7]. Az adaptív operátorkiválasztással foglalkozó kutatások szintén azt mutatják, hogy az állapotalapú tanulás az optimalizálás különböző szakaszaiban megfelelő keresési operátorokat választhat [8]. A legtöbb RL-lel támogatott metaheurisztikát azonban általános optimalizálási problémákon vagy egyetlen optimalizálással vizsgálták; skálázható, energiahatékony WSN-klaszterezésben való értékelésük korlátozott.

A javasolt RL-SI keretrendszer ezt a hiányosságot közös, 13 változós állapotrepresentációval és hatműveletes vezérlési felülettel kezeli az ABC, a folytonos ACO, a GWO és a PSO esetében. A különálló Double-DQN politikák figyelembe veszik az egyes optimalizálók keresési sajátosságait, és lehetővé teszik az algoritmusok és hálózatméretek egységes kiértékelését.

3. A JAVASOLT RL-SI KERETRENDSZER

Az RL-SI megerősítéses tanulással támogatott rajintelligencia-keretrendszer dinamikusan vezérli egy természetihletésű optimalizáló keresési viselkedését a klaszterfejek kiválasztása során. A keretrendszert négy reprezentatív optimalizálással értékeljük: mesterséges méhkolónia-algoritmussal (ABC), folytonos hangyakolónia-optimalizálással (ACO), szürkefarkas-optimalizálással (GWO) és részecskeraj-optimalizálással (PSO). A megerősítéses tanulási vezérlő szükség esetén úgy avatkozik be a keresésbe, hogy közben nem módosítja az algoritmusok eredeti keresési egyenleteit.

3.1. WSN-rendszer és energiamodell

Egy statikus, N szenzorcsomópontból és egy központi nyelőcsomópontból álló WSN-t vizsgálunk. Minden csomópont kezdeti energiája $E_0 = 0,5$ J, és periodikusan egy $k = 1024$ bites adatcsomagot generál. Minden hálózati epochban az optimalizáló az élő csomópontok számával megegyező dimenziójú döntési vektort állít elő. Egy csomópont akkor válik klaszterfejjé, ha a hozzá tartozó döntési változó értéke legalább 0,5. Minden

nem klaszterfej csomópontot a legközelebbi kiválasztott klaszterfejhez rendelünk. A csomópont azonban csak akkor továbbít a klaszterfejen keresztül, ha a tag-CH átvitel, a vétel, az aggregáció és a CH-nyelő átvitel együttes energiaköltsége kisebb a közvetlen továbbítás költségénél. Az útválasztás ezért az $SN \rightarrow$ nyelő vagy az $SN \rightarrow CH \rightarrow$ nyelő kapcsolatot választhatja. A klaszterfejek közvetlenül kommunikálnak a nyelőcsomóponttal; klaszterfejek közötti többugrásos útválasztást nem alkalmazunk.

A kommunikáció energiaköltségét az elsőrendű rádiómodellel számítjuk. Egy k bites adatsomag d távolságra történő továbbításának energiaigénye:

$$E_{Tx}(k, d) = \begin{cases} kE_{elec} + k\varepsilon_{fs}d^2, & d < d_0, \\ kE_{elec} + k\varepsilon_{mp}d^4, & d \geq d_0, \end{cases} \quad (1)$$

ahol $(d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}})$.

A vételhez és az adataggregációhoz szükséges energia:

$$E_{Rx}(k) = kE_{elec}, \quad E_{DA}(k) = kE_{agg}. \quad (2)$$

Az elfogadott paraméterek: $(E_{elec} = 50)\text{nJ/bit}$, $(\varepsilon_{fs} = 10)\text{pJ/bit/m}^2$, $(\varepsilon_{mp} = 1.3)\text{fJ/bit/m}^4$ és $(E_{agg} = 5)\text{nJ/bit}$. Egy csomópont inaktívvá válik, amikor a maradék energiája eléri a nullát. A szimuláció 5000 epoch után, vagy ha az eredeti csomópontoknak legfeljebb 10%-a marad életben, véget ér.

3.2. Fitnessfüggvény

Minden jelölt megoldást normalizált, többcélú fitnessfüggvénnyel értékelünk, amely a kommunikációs energiaköltséget, a klaszterek tömörségét, az energiaegyensúlyt, az alacsony energiaszintű csomópontok védelmét és a megoldás megvalósíthatóságát egyesíti:

$$F = E_{norm} + 10D_{norm} + 5\sigma_{norm} + L_{norm} + 10^4P_{CH} + P_{dead}. \quad (3)$$

Itt (E_{norm}) a normalizált kommunikációs energiaköltséget jelöli, beleértve az átvitelt, vételt és adataggregációt. (D_{norm}) A tagsomópontok és a hozzájuk rendelt CH-k közötti távolságot méri. A tag (σ_{norm}) bünteti az egyenlőtlen maradékegyenlőtlenséget, míg (L_{norm}) a a legalacsonyabb előrejelzett maradékegyenlőtlenséggel rendelkező csomópontot védi. (P_{CH}) bünteti azokat a megoldásokat, amelyek CH-inak száma túlzottan eltér a 10%-os célaránytól, és (P_{dead}) megakadályozza a halott csomópontok kiválasztását. Minden tag összehasonlítható tartományokra van normalizálva, és az alacsonyabb fitnessérték jobb klaszterezési konfigurációt jelez. Ez a megfogalmazás nem kedvez azoknak a megoldásoknak, amelyek minimalizálják az azonnali energiafogyasztást azáltal, hogy ismételten kiválasztják ugyanazokat a kiváló minőségű csomópontokat, mivel az ilyen viselkedés növelné az energiaegyensúlyhiányt és lerövidítené a hálózat élettartamát.

3.3. Algoritmusspecifikus RL-politikák tanítása

Az ABC-hez, a folytonos ACO-hoz, a GWO-hoz és a PSO-hoz külön Double-DQN politikát tanítunk. A közös vezérlési felület mellett így minden politika megtanulhatja a saját optimalizálójának keresési sajátosságait. Minden WSN-epoch elején a politika normalizált, 13 változós állapotot figyel meg, amely tartalmazza a hálózati előrehaladást, a csomópontok túlélési arányát, a maradék energiát, az energiaegyensúlyt, az előző klaszterfejarányt, a csomagkézbesítési teljesítményt, az optimalizálási dimenziót, a közelmúltbeli fitnessjavulást, az energiamegtartást, a futásidő-hatékonyságot, a közelmúltbeli csomópontvesztést, az előző műveletet és egy konfigurációs indikátort. A vezérlő a keresés folytatása, a feltárás növelése, a kiaknázás növelése, a legrosszabb megoldások diverzifikálása, a legrosszabb megoldások újraindítása vagy helyi finomítás közül választ. A kiválasztott művelet az aktuális epochban vezérli az optimalizálót, majd a frissített hálózati állapot alkotja a következő állapotot.

A tanítás során a kiválasztott műveletet és egy determinisztikus, szabályalapú kontrafaktuális döntést ugyanabból a hálózati állapotból, azonos véletlenmagokkal hajtunk végre. A kontrafaktuális ág kizárólag a jutalom kiszámítását szolgálja, míg a kiválasztott művelet ága folytatja a tanítási trajektóriát. A jutalom a két ág normalizált teljesítménykülönbségeit egyesíti az energiamegtartás, a csomóponttúlélés, az energiaegyensúly, a csomagkézbesítés, a fitness és a futásidő alapján:

$$[r_t = 0.35R_E + 0.20R_A + 0.15R_B + 0.10R_P + 0.15R_F + 0.05R_T - P_{switch} - P_{invalid}.] \quad (4)$$

A jutalomkomponenseket normalizáljuk és korlátozzuk a szélsőséges értékek hatásának mérséklésére. Az egymást követő epochok közötti műveletváltás 0,005 büntetést kap, a nem megvalósítható optimalizálási eredményekhez pedig további büntetés tartozik. A Double-DQN tanítását tapasztalati visszajátzás, célhálózat és ϵ -mohó stratégia stabilizálja.

3.4. RL-vezérelt optimalizálási eljárás

Minden epoch elején az adott optimalizáláshoz tartozó politika az aktuális állapot alapján kiválaszt egy műveletet. A művelet az egész epoch alatt érvényben marad, és algoritmus-specifikus paramétermódosításokká alakul. A feltérési és kiaknázási műveletek a natív keresési együttthatókat módosítják, míg a diverzifikálás, az újraindítás és a helyi finomítás kiválasztott megoldásokat újonnan generált jelöltekkel helyettesít. A műveletek erősségét optimalizálónként külön állítjuk be. Az optimalizálás után dekódoljuk a legjobb megoldást, a csomópontokat klaszterfejekhez rendeljük, kiválasztjuk az útvonalakat, majd frissítjük a csomópontok energiáját. Az eredményül kapott hálózati állapot alkotja a következő állapotot. Az eljárás a maximális epochszám vagy az élőcsomópont-küszöb eléréséig folytatódik.

4. KÍSÉRLETI MÓDSZERTAN

4.1. Hálózati konfigurációk

Három hálózati méretet vizsgálunk: 50, 250 és 1000 csomópontot. A hozzájuk tartozó telepítési területek rendre $300 \times 300 \text{ m}^2$, $671 \times 671 \text{ m}^2$ és $1342 \times 1342 \text{ m}^2$. A terület csomópontszámmal arányos növelése közel állandó csomópontsűrűséget tart fenn, így a hálózati méret hatása nem keveredik a telepítési sűrűség hatásával. Minden méretet öt független topológiai véletlenmaggal értékelünk. Ugyanazokat a magokat használjuk az ABC, a folytonos ACO, a GWO és a PSO, valamint az F-SI és az RL-SI esetében, biztosítva a párosított hálózati megvalósításokat. Minden kísérlet egyenletes telepítést, központi nyelőcsomópontot, helyhez kötött csomópontokat és külsőleg előidézett meghibásodások nélküli működést feltételez. Az optimalizálók konfigurációit előzetes érzékenységvizsgálattal választottuk ki. Az ABC és az ACO 20 jelöltet és 750 iterációt, a GWO 50 jelöltet és 750 iterációt, a PSO pedig 50 jelöltet és 500 iterációt használ.

Kísérleti paraméterek

1. táblázat

Paraméter	Érték	Paraméter	Érték
Algoritmusok	ABC, folytonos ACO, GWO, PSO	Rádióparaméterek	$E_{\text{elec}}=50\text{nJ/bit}$; $\epsilon_{\text{fs}}=10\text{pJ/bit/m}^2$; $\epsilon_{\text{mp}}=0.0013\text{pJ/bit/m}^4$; $E_{\text{agg}}=5\text{nJ/bit}$
Csomópontok (N) és területek	50: $300 \times 300 \text{ m}^2$; 250: $671 \times 671 \text{ m}^2$; 1000: $1342 \times 1342 \text{ m}^2$	Maximális epochszám és leállítási feltétel	5000 epoch vagy leállás 10% élő csomópontnál
Telepítés és nyelőcsomópont	Egyenletes telepítés; központi nyelőcsomópont	Optimalizálóbéállítások (populációméret × iterációs szám)	ABC/ACO: 20×750 ; GWO: 50×750 ; PSO: 50×500
Kezdeti energia és csomagméret	0,5 J; 1024 bit	Végso értékelési magok	11, 23, 37, 51, 79

4.2. Tanítási és értékelési protokoll

A tanításhoz, a validáláshoz, az elkülönített teszteléshez és a végso értékeléshez egymást nem átfedő véletlenmagokat használunk. Az algoritmus-specifikus Double-DQN politikák tanítása a 3001–3020 magokkal, algoritmusonként 480 epizódon történik; a validálás a 4001–4005 magokkal, algoritmusonként 120 epizódot használ. Egy tanítási trajektória 12 egymást követő WSN-epochból, egy validálási trajektória pedig 20 epochból áll. A betanított politikákat ezután az 5001–5005 elkülönített tesztmagokkal ellenőrizzük. A végso teljesítményt öt korábban nem látott hálózati elrendezésen értékeljük, amelyeket a 11, 23, 37, 51 és 79 véletlenmagokkal generálunk. A négy algoritmus, két vezérlési változat, három hálózati méret és öt értékelési mag összesen 120 teljes élettartamú szimulációt eredményez. A 12 algoritmus–hálózati méret csoport mindegyike vezérlési változatonként öt futást tartalmaz. A hálózat telepítéséhez, az optimalizáló inicializálásához és a vezérlő végrehajtásához külön véletlenszám-folyamokat használunk.

Mivel az optimalizálók eltérő populációméreteket és iterációs számokat alkalmaznak, a jelentett számításiköltség-mutató az optimalizáló futásideje; a fitnessfüggvény-kiértékelések száma a szimulációs naplóban marad. Az epochszintű hálózati és vezérlési nyomkövetések az élő csomópontok számát, a maradék

energiát, a csomagkézbesítési arányt, a legjobb fitneszt, a kiválasztott műveleteket, a futásidőt és a függvénykiértékelések számát tartalmazzák.

5. EREDMÉNYEK ÉS ÉRTÉKELÉS

A hálózati élettartamot az első csomópont kiesésének időpontjával (FND), a csomópontok fele kiesésének időpontjával (HND) és az utolsó 10%-os túlélési ponttal (LTPA) jellemezzük. Az energiateljesítményt a maradékenergia-görbe alatti területtel (RE-AUC) és a végső maradék energiával (Final RE), a csomóponttúlélést pedig az élőcsomópont-görbe alatti területtel (AN-AUC) mérjük. A kommunikáció megbízhatóságát a csomagkézbesítési arány (PDR), a számítási költséget pedig az optimalizáló futásideje jellemzi. A 2. táblázat minden algoritmusra, vezérlőre és hálózatméretre az öt értékelési elrendezés átlagát és szórását közli.

5.1. Az RL-alapú vezérlés hatása

N = 50 esetén az RL-SI az ABC-vel mutatja a legnagyobb javulást. Az FND-t 713,0-ról 1027,8 epochra késlelteti, ami körülbelül 44,2%-os növekedésnek felel meg, miközben a futási időt 12057,1-ről 8231,6 másodpercra csökkenti. Ezt a korai stabilitási javulást kissé alacsonyabb HND, RE-AUC és AN-AUC értékek kísérik, ami azt jelzi, hogy az első csomópont meghibásodásának késleltetése nem feltétlenül hosszabbítja meg a teljes élettartamot. Az RL-vezérelt ACO kiegyensúlyozottabb eredményt ad: a futási idő 15,5%-kal csökken, a HND 2488,4-ről 2501,4 epochra nő, az LTPA 4599,2-ről 4665,8 epochra nő, és mind az energia, mind az élő csomópont AUC javul. A GWO szinte azonos élettartam- és energiaeredményeket produkál mindkét kezelés során, bár az RL-SI körülbelül 5,1%-kal növeli a futási időt. PSO esetén az RL-SI 7,5%-kal csökkenti a futási időt, míg a fennmaradó metrikák szinte változatlanok. Az összes 50 csomópontos konfiguráció megtartja a túlélő csomópontokat a megfigyelési időszak végén, és PDR-értékeik közel egy maradnak. N = 250 esetén az RL-SI minden optimalizáló esetében csökkenti a futási időt. Az F-SI-hez viszonyított csökkentések körülbelül 29,3% az ABC, 18,9% az ACO, 1,5% a GWO és 18,4% a PSO esetében.

Összefoglaló eredmények

2. táblázat

Hálózatméret N= 50										
Algoritmus	Vezérlő	Statisztika	Futásidő (s)	FND	HND	LTPA	RE-AUC	AN-AUC	Végső RE	PDR
ABC	F-SI	Átlag	12057.1	713	2566.2	Túlélő	4749.6	3.91E+04	1.37E+05	0.738
		Szórás	1013.4	164.4	274.3	Túlélő	452.7	3.46E+03	1.39E+04	0.095
	RL-SI	Átlag	8231.6	1027.8	2440	Túlélő	4604.6	3.61E+04	1.35E+05	0.764
		Szórás	1089.4	260.5	299.6	Túlélő	701.4	3.76E+03	1.54E+04	0.183
ACO	F-SI	Átlag	4093.9	949.2	2488.4	Túlélő	4599.2	3.66E+04	1.35E+05	0.761
		Szórás	509.1	219.4	281.6	Túlélő	635.2	3.44E+03	1.49E+04	0.157
	RL-SI	Átlag	3457.5	916.2	2501.4	Túlélő	4665.8	3.75E+04	1.36E+05	0.741
		Szórás	360.8	225.1	271.3	Túlélő	582.0	3.49E+03	1.46E+04	0.138
GWO	F-SI	Átlag	8672.9	970.2	2512.4	Túlélő	4620.0	3.70E+04	1.35E+05	0.764
		Szórás	1156.9	253.4	282.8	Túlélő	660.6	3.51E+03	1.50E+04	0.17
	RL-SI	Átlag	9116.9	971.2	2492.6	Túlélő	4637.8	3.70E+04	1.36E+05	0.75
		Szórás	1233.3	243	271.3	Túlélő	613.7	3.50E+03	1.48E+04	0.155
PSO	F-SI	Átlag	7863.7	753.4	2245.6	Túlélő	4777.4	3.29E+04	1.29E+05	0.709
		Szórás	662.7	145.9	325.4	Túlélő	367.6	3.32E+03	1.37E+04	0.137
	RL-SI	Átlag	7272.2	715.6	2245.2	Túlélő	4792.6	3.29E+04	1.29E+05	0.707
		Szórás	543.6	93.3	323.5	Túlélő	341.1	3.27E+03	1.36E+04	0.135

Hálózatméret N= 250										
Algoritmus	Vezérlő	Statisztika	Futásidő (s)	FND	HND	LTPA	RE-AUC	AN-AUC	Végső RE	PDR
ABC	F-SI	Átlag	6721.2	27	169.4	2199.6	3.58E+04	1.33E+05	5.017	0.997
		Szórás	787.5	4.5	20.1	352.8	3.16E+03	1.03E+04	0.887	2.50E-04
	RL-SI	Átlag	4755	19.2	146	2067.8	3.32E+04	1.25E+05	5.002	0.998
		Szórás	399	3.1	18.3	252.3	2.32E+03	9.42E+03	0.933	2.10E-04
ACO	F-SI	Átlag	2987.3	23.4	172.8	2176.6	3.53E+04	1.31E+05	4.92	0.997
		Szórás	462	5	21.6	361.8	2.91E+03	1.16E+04	1.189	3.60E-04
	RL-SI	Átlag	2421.6	27.4	140.6	2080.0	3.34E+04	1.25E+05	4.927	0.998
		Szórás	191.8	3.6	10.8	210.6	2.35E+03	9.21E+03	0.762	1.70E-04
GWO	F-SI	Átlag	5932.3	22.2	171	2175.0	3.61E+04	1.31E+05	5.005	0.997
		Szórás	628.6	4.6	15.2	329.9	2.92E+03	1.11E+04	1.05	2.90E-04
	RL-SI	Átlag	5846.3	22.8	172.4	2186.2	3.61E+04	1.32E+05	4.927	0.998
		Szórás	671.3	3.5	18	338.8	3.06E+03	1.12E+04	1.101	3.40E-04
PSO	F-SI	Átlag	5068.1	22	153.4	2015.0	3.26E+04	1.19E+05	4.988	0.997
		Szórás	1094	3.6	19.7	247.2	2.72E+03	1.01E+04	0.729	3.50E-04
	RL-SI	Átlag	4136.1	21.6	148.4	2017.6	3.27E+04	1.19E+05	4.984	0.998
		Szórás	395.5	1.1	24.1	236.1	2.74E+03	1.00E+04	0.702	1.80E-04

Hálózatméret N= 1000										
Algoritmus	Vezérlő	Statisztika	Futásidő (s)	FND	HND	LTPA	RE-AUC	AN-AUC	Végső RE	PDR
ABC	F-SI	Átlag	4927.5	1	10.6	269.4	1.49E+04	5.04E+04	26.412	0.963
		Szórás	372.8	0	0.5	94.0	4.08E+03	1.20E+04	1.25	8.07E-03
	RL-SI	Átlag	2250	1	9	228.4	1.32E+04	4.24E+04	28.211	0.983
		Szórás	261.5	0	0.7	57.2	2.90E+03	7.97E+03	1.539	4.01E-03
ACO	F-SI	Átlag	979.8	1	10.6	253.4	1.45E+04	4.85E+04	27.161	0.962
		Szórás	120	0	0.5	71.1	3.34E+03	9.67E+03	1.654	6.54E-03
	RL-SI	Átlag	1059.7	1	9.2	223.6	1.29E+04	4.12E+04	28.479	0.983
		Szórás	128	0	0.4	55.2	2.80E+03	7.57E+03	1.63	3.76E-03
GWO	F-SI	Átlag	2410.3	1	14.4	254.6	1.54E+04	5.02E+04	28.267	0.936
		Szórás	320.4	0	0.5	78.1	3.65E+03	1.05E+04	1.476	1.35E-02
	RL-SI	Átlag	2319.4	1	14.4	251.8	1.53E+04	4.99E+04	28.12	0.973
		Szórás	318.7	0	0.5	71.8	3.41E+03	9.74E+03	1.82	7.44E-03
PSO	F-SI	Átlag	1632.7	1	9.2	232.6	1.35E+04	4.35E+04	27.907	0.966
		Szórás	214.8	0	0.4	60.5	3.01E+03	8.41E+03	1.453	6.76E-03
	RL-SI	Átlag	1586.4	1	9.6	231.2	1.34E+04	4.35E+04	27.842	0.983
		Szórás	171.9	0	0.5	47.1	2.56E+03	6.99E+03	2.128	3.43E-03

Az RL-SI a PDR-t is növeli körülbelül 0,997-ről 0,998-ra mind a négy algoritmus esetében. Az élettartamra vonatkozó eredmények továbbra is algoritmusfüggőek. Az RL-vezérelt ABC alacsonyabb FND, HND, LTPA, RE-AUC és AN-AUC értékekkel rendelkezik, mint az F-SI. Az ACO segítségével az RL-SI 23,4-ről 27,4 epochra javítja az FND-t, de csökkenti a későbbi élettartam-mértékeket. A GWO adja a legkonzisztensebb eredményt ezen a skálán: az RL-SI kismértékben javítja az FND, HND, LTPA és AN-AUC értékeket, miközben megőrzi a RE-AUC értéket. A PSO csak kisebb különbségeket mutat, alacsonyabb futási idővel és kissé magasabb LTPA-val RL-SI alatt.

Az RL-SI fő előnye N = 1000 esetén a kommunikáció megbízhatósága. PDR-értéke eléri a 0,983-at ABC, ACO és PSO, valamint a 0,973-at GWO esetén. A megfelelő F-SI értékek 0,936 és 0,966 között mozognak. A legnagyobb javulás a GWO esetében tapasztalható, ahol a PDR 0,936-ról 0,973-ra nő. Az RL-SI minden optimalizáló esetében alacsonyabb PDR-variabilitást is eredményez. A futási idő 54,3%-kal csökken ABC, 3,8%-kal GWO és 2,8%-kal PSO esetén, bár az ACO futási ideje 8,2%-kal nő. Az élettartam-előny kevésbé következetes: az F-SI magasabb HND-t és LTPA-t biztosít ABC és ACO esetén, míg a GWO szinte változatlan marad, és az RL-vezérelt PSO kismértékben javítja a HND-t. Összességében az RL-SI következtetesebben javítja a futási időt és a csomagkézbesítést, mint az energiatakarékosság vagy a hálózat élettartama.

5.2. Skálázhatóság

A hálózat élettartama meredeken csökken a hálózat növekedésével. Az FND 713,0 és 1027,8 epoch között fordul elő N = 50 esetén, 19,2 és 27,4 epoch között N = 250 esetén, és minden N = 1000 konfiguráció első epochjában. Hasonlóképpen, a HND körülbelül 2245–2566 epochról N = 50 esetén 141–172 epochra csökken N = 250 esetén, és mindössze 9–14 epochra N = 1000 esetén. Az LTPA, a RE-AUC és az AN-AUC ugyanazt a csökkenő tendenciát mutatja. Ez azt jelzi, hogy a hasonló csomópont-sűrűség fenntartása a telepítési terület növelésével nem őrzi meg a hálózat élettartamát, mivel a nagyobb kommunikációs távolságok növelik az átviteli költségeket.

A GWO mutatja a legerősebb késői viselkedést a legnagyobb skálán, mindkét kezelés során 14,4 epochos HND-t ér el, és 251 epochos felett megőrzi az LTPA-t. Mindazonáltal az RL-SI hatékonyabban skálázódik a csomagkézbesítés szempontjából. PDR-je csak körülbelül 0,999-ről csökken N = 50-nél 0,973–0,983-ra N = 1000-nél, míg az F-SI akár 0,936-ra is visszaesik. A PDR és LTPA alacsonyabb RL-SI szórása N = 1000-nél szintén következetesebb viselkedést sugall az elrendezések között.

A teljes futási idő csökkenését nagyobb méretekben nem szabad a számítási komplexitás csökkenésének tekinteni. A nagyobb hálózatok lényegesen kevesebb epoch után fejeződnek be, ezért az optimalizálók kevesebbszer futnak le. A hálózatméretek közötti futási idő-összehasonlításokat következtetésképpen a teljes élettartammal együtt kell értelmezni. Ezek az eredmények azt mutatják, hogy az RL-SI javítja a kommunikáció megbízhatóságát és gyakran csökkenti az optimalizálási időt, de további energiatudatos útválasztási mechanizmusokra van szükség a nagyméretű hálózatok gyors élettartam-romlásának kezeléséhez.

6. KÖVETKEZTETÉSEK

A tanulmány az RL-SI egységes megerősítéssel tanulósi keretrendszerrel mutatta be az ABC, a folytonos ACO, a GWO és a PSO vezérlésére energiahatékony klaszterfej-kiválasztás során. Az algoritmus-specifikus Double-DQN politikák közös hálózati állapot- és műveleti felületet használnak a feltárás, a kiaknázás, a diverzifikálás, az újraindítás és a helyi finomítás közötti egyensúly beállítására. Az eredmények szerint az RL-SI

nem teljesít minden mutatóban következetesen jobban az F-SI-nél. Legkövetkezetesebb előnye a nagyobb csomagkézbesítési arány és a kisebb optimalizálási futásidő, különösen nagyobb hálózatokban. Ezer csomópont esetén az RL-SI mind a négy optimalizálónál számottevően javítja a PDR-t, és csökkenti annak elrendezések közötti szórását. Az FND, HND, LTPA, RE-AUC és AN-AUC javulása ugyanakkor az optimalizálótól és a hálózatmérettől függ; több konfigurációban az F-SI hatékonyabban őrzi meg az energiát vagy a hálózati élettartamot. A tanult vezérlés tehát hasznos adaptációt biztosít, de nem szünteti meg a nagy telepítési területek és a hosszú kommunikációs távolságok alapvető energiaköltségét. A vizsgálat négy optimalizálóra, három hálózatméretre, konfigurációként öt elrendezésre, egyenletes és statikus telepítésre, központi nyelőcsomópontra és elsőrendű rádiómodellre korlátozódik. A szimulációk nem modellezik az interferenciát, a fadinget, az újraküldéseket, az üresjáratot, a csomópontmobilitást, a külső meghibásodásokat vagy a klaszterfejek közötti többugrós út választást. Az elrendezések kis száma a statisztikai erőt és az eredmények általánosíthatóságát is korlátozza.

A jövőbeni kutatás további hálózati feltételeket, nyelőpozíciókat, telepítési mintákat és meghibásodási forgatókönyveket vizsgál majd több független futással és párosított statisztikai próbákkal. A modell klaszterfejek közötti többugrós út választással és valószínűbb kommunikációs modellekkel is bővíthető. Alternatív jutalomfüggvények, a hálózatméretek közötti transzfertanulás és többcélú RL-politikák tovább javíthatják a csomagkézbesítést, az energiamegőrzést, a hálózati élettartamot és a számítási költség közötti egyensúlyt.

HIVATKOZÁSOK

- [1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proceedings of the 33rd Hawaii International Conference on System Sciences*, IEEE, 2000. doi:10.1109/HICSS.2000.926982.
- [2] N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Energy-Aware Clustering for Wireless Sensor Networks Using Particle Swarm Optimization," *18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, 2007, pp. 1–5. doi:10.1109/PIMRC.2007.4394521.
- [3] X. Li, B. Keegan, F. Mtenzi, T. Weise, and M. Tan, "Energy-Efficient Load Balancing Ant Based Routing Algorithm for Wireless Sensor Networks," *IEEE Access*, vol. 7, pp. 113182–113196, 2019. doi:10.1109/ACCESS.2019.2934889.
- [4] Z. Wang, H. Ding, B. Li, L. Bao, and Z. Yang, "An Energy Efficient Routing Protocol Based on Improved Artificial Bee Colony Algorithm for Wireless Sensor Networks," *IEEE Access*, vol. 8, pp. 133577–133596, 2020. doi:10.1109/ACCESS.2020.3010313.
- [5] M. Kaddi, M. Omari, K. Salameh, and A. Alnoman, "Energy-Efficient Clustering in Wireless Sensor Networks Using Grey Wolf Optimization and Enhanced CSMA/CA," *Sensors*, vol. 24, no. 16, Article 5234, 2024. doi:10.3390/s24165234.
- [6] R. Durgut, M. E. Aydin, and I. Atli, "Adaptive Operator Selection with Reinforcement Learning," *Information Sciences*, vol. 581, pp. 773–790, 2021. doi:10.1016/j.ins.2021.10.025.
- [7] F. Zhao, Z. Wang, L. Wang, T. Xu, and N. Zhu, "A Multi-Agent Reinforcement Learning Driven Artificial Bee Colony Algorithm with the Central Controller," *Expert Systems with Applications*, vol. 219, Article 119672, 2023. doi:10.1016/j.eswa.2023.119672.
- [8] J. Pei, Y. Mei, J. Liu, M. Zhang, and X. Yao, "Adaptive Operator Selection for Meta-Heuristics: A Survey," *IEEE Transactions on Artificial Intelligence*, vol. 6, no. 8, pp. 1991–2012, 2025. doi:10.1109/TAI.2025.3545792.