

# HATÉKONY MONITOROZÁS ÉS ERŐFORRÁS- GAZDÁLKODÁS ÉRZÉKELŐ HÁLÓZATOKKAL

*PhD Disszertáció*

***Tornai Kálmán***

okleveles mérnök informatikus

Témavezető:

**Dr. Levendovszky János**  
az MTA Doktora



Pázmány Péter Katolikus Egyetem  
Információs Technológiai és Bionikai Kar  
Interdiszciplináris Műszaki Tudományok Doktori Iskola

Budapest, 2013



*„If we knew what it was we were doing, it would not be called research, would it?“*

Albert Einstein



## KÖSZÖNETNYILVÁNÍTÁS

Köszönöm témavezetőmnek, Dr. Levendovszky János professzor úrnak az elmúlt években nyújtott támogatását, építő kritikáit, amelyekkel a disszertáció színvonalát emelte. Köszönettel tartozom a Doktori Iskola korábbi és jelenlegi vezetőinek, Dr. Roska Tamás és Dr. Szolgay Péter professzor uraknak, akik biztosították munkához szükséges háttérrel. Ezenkívül külön hálával tartozom Nyékyné dr. Gaizler Judit (Pro-)Dékánasszonynak, aki a legkülönbözőbb időpontokban fogadott és hasonlóan különféle problémákra segített megoldást találni. Hasonlóan hálával tartozom a kari tanulmányi és gazdasági osztályon dolgozóknak és könyvtárosnak, akikhez szintén bármikor fordulhattam ügyes-bajos dolgaikkal.

A kutatási munka és a publikációk elkészítése közben felbecsülhetetlen segítséget nyújtottak szerzőtársaim, elsősorban a témavezetőm, továbbá Treplán Gergely, Fogarasi Norbert, László Endre, Oláh András, akiknek ezúton köszönöm a közös munkát és türelmet.

Köszönetet mondok dr. Oláh Andrásnak, valamint a WSN kutatócsoport tagjainak, illetve a Doktori Iskolában velem együtt munkálkodó doktorandusz társaimnak különösen azoknak, akikkel az elmúlt években több-kevesebb ideig egy szobában voltam: Gelencsér András, Varga Balázs, Balogh Ádám, Szabó Vilmos, Füredi László, Tisza Dávid, Manno-Kovács Andrea, Tibold Róbert, Vizi Péter, Nemes Csaba, Juhász Imre, Jani Mátyás, Nagy Gábor, Borbély Bence. Ők hasznos tanácsokkal, beszélgetésekkel és együttérző fülekkel voltak irántam, valamint a felmerült problémák megvitatásában mindvégig lelkesek maradtak.

Számítási kapacitást biztosítottak a robotika labor tagjai, a rendszergazdák és mindenki, aki hardvert vagy processzoridőt ajánlott fel. Nélkülük nem sikerült volna a szimulációkat lefuttatni, ezért hálás vagyok.

Köszönet illeti azt a kiskollégiumi közösséget, amelyben biztosított volt az elmélyedt tanuláshoz és kutatáshoz szükséges légkör és baráti segítség. Ebben társak voltak több éven keresztül: Gyarmati Gábor, Zsedrovits Tamás, Daubner Lóránt, Tornai Gábor, Józsa Csaba, Füredi László, Pásztor Csaba, Gál Árpád.

A dolgozat elkészültében jelentős segítséget jelentett a magyar és angol nyelvi lektorálás, amelyet ezúton is köszönök Tornai Katinkának és Dalmadi Nórának.

Különös és kiemelt hálával tartozom feleségemnek, Gabicának, aki mellettem állt és elviselte a hosszú és fáradt napjaimat is, és aki ajándék nekem – masnival.

Végül, de nem utolsósorban hálás vagyok családomnak, szüleimnek, nagyszüleimnek, testvéreimnek a szeretetért, törődésért és segítségért, amellyel mindig támogattak.



# HATÉKONY MONITOROZÁS ÉS ERŐFORRÁS- GAZDÁLKODÁS ÉRZÉKELŐ HÁLÓZATOKKAL

*Tornai Kálmán*

## KIVONAT

Számos olyan alkalmazás jelent meg a kommunikációs- és szenzoros technológia fejlődésének köszönhetően, amely alkalmas a mobilitási és flexibilitási igények kielégítésére. Az alkalmazások során az egyik leggyakoribb feladat a megfigyelt komplex rendszerben előforduló krízishelyzetek és események detekciója.

Annak érdekében, hogy hatékony monitorozó rendszert lehessen kialakítani, figyelembe kell venni, hogy az adatátvitelhez és adatfeldolgozáshoz rendelkezésre álló erőforrások végesek. Ez jelentősen korlátozza a vezeték nélküli rádiós hálózatban elérhető kommunikációs átvitel sebességét, valamint a hálózat elemeinek élettartamát és a spektrális kihasználtságot. A jelenlegi kutatások olyan monitorozási rendszerek kidolgozására fókuszálnak, amelyekben a teljesítőképesség, a pontosság és az erőforrások optimális kihasználásának elérése sikerül mindamellet, hogy az előzőleg említett kényszereket be kell tartani. A hatékony erőforrás felhasználás kérdései számos vezeték nélküli, mobil technológiát érintenek, amelyek során fellép valamilyen kényszer, amit figyelembe kell venni.

*A disszertációmban egy konkrét alkalmazás osztályt bemutatva vizsgálom a problémát. Az általam bemutatott algoritmusok és metódusok segítségével lehetővé válik a monitorozó rendszerekben a vezeték nélküli érzékelő hálózatokkal az 1) optimális közeghozzáférés, valamint a 2) kritikus események kis hibavalószínűségű detektálása a mérési eredmények felhasználásával. A közeghozzáférési és más ütemezési problémákat kvadratikus optimalizálási feladattá transzformáltam, amelyet Hopfield neurális hálózattal oldok meg. Az események detekciójára egy predikció alapú, az idősorokban szokatlan értékeket detektáló algoritmust felhasználó módszert implementáltam. Ennél a módszernél analitikusan meghatározom az optimális paramétereket. Az ismertetésre kerülő eljárás elosztottan, illetve párhuzamos architektúrán is futtatható.*

A bemutatott módszerek és algoritmusok nem kizárólagosan csak a vezeték nélküli monitorozó rendszerek esetén alkalmazhatóak, hanem más erőforrás optimalizálással, pénzügyi folyamatok ütemezésének optimalizálásával, vagy adatfeldolgozással és döntésemeléttel kapcsolatos kérdésekre és problémákra is választ adnak, továbbá sémákat mutat be az elosztott adatfeldolgozás megvalósítására.





## TARTALOMJEGYZÉK

<i>Első fejezet</i>	8
<b>1. Bevezetés.....</b>	<b>8</b>
1.1. Előszó .....	8
1.1.1. A kutatási terület áttekintése.....	9
1.1.2. Erőforrás kezelés és csomagütemezés .....	9
1.1.3. Outlier- és esemény detekció.....	11
1.2. A disszertáció célkitűzései.....	13
1.3. A disszertáció felépítése .....	13
 <i>Második fejezet</i>	 14
<b>2. Erőforrások ütemezése.....</b>	<b>14</b>
2.1. Bevezetés, technológiai motivációk és nyitott kérdések.....	14
2.2. Hagyományos megközelítések .....	15
2.2.1. Jelenleg használt erőforrás ütemező módszerek .....	15
2.2.2. Jelenleg használt csomagütemező protokollok.....	16
2.3. Az ütemezési probléma modellje.....	18
2.4. Új algoritmusok az ütemezési problémák megoldására.....	23
2.4.1. Új heurisztikus algoritmusok .....	23
2.4.2. Új, kvadratikus optimalizáláson alapuló megoldás az ütemezési feladatokra .....	29
2.4.3. A kvadratikus optimalizálási feladatok megoldása Hopfield neurális hálózattal .....	36
2.4.4. HNN alapú megoldó párhuzamos futtathatósága .....	39
2.5. A kvadratikus optimalizáció eredményeinek bemutatása.....	41
2.5.1. Teljesítőképesség összehasonlítás a TWT probléma esetén.....	41
2.5.2. Teljesítőképesség bemutatása az egyszerűsített WSN feladat esetén.....	43
2.5.3. Teljesítőképesség összehasonlítása a teljes WSN csomagütemezés esetén.....	44
2.5.4. Összefoglalás .....	46
 <i>Harmadik fejezet</i>	 48
<b>3. Hatékony monitorozás vezeték nélküli érzékelő hálózatokkal .....</b>	<b>48</b>
3.1. Bevezetés, technológiai motivációk és nyitott kérdések.....	48

3.2.	Hagyományos megközelítések a feladat megoldására.....	49
3.2.1.	Outlier detekciós módszerek.....	49
3.2.2.	Jelenleg használatos esemény detekciós módszerek.....	51
3.3.	Új, kis komplexitású outlier detekciós algoritmus.....	54
3.3.1.	Optimális megoldás lineáris predikcióval.....	55
3.3.2.	Megoldás nem lineáris predikcióval.....	58
3.4.	Új, klaszterezésen alapuló esemény detekciós algoritmus .....	60
3.5.	Az esemény detekciós módszer megvalósítása WSN alapú monitorozó környezetben .....	63
3.5.1.	A WSN hálózat mérési modellje .....	63
3.5.2.	Az esemény detekciós algoritmus protokollja .....	64
3.6.	A módszer komponenseinek párhuzamos futtathatósága .....	65
3.6.1.	Lokális, kis komplexitási outlier detekciós algoritmus.....	66
3.6.2.	Klaszterezés alapú esemény detekció .....	66
3.6.3.	A lokális detekció döntési paramétereinek meghatározása és a detektor adaptálása ..	67
3.6.4.	Optimális klaszterek meghatározása az egyes node-okhoz .....	67
3.6.5.	A párhuzamosíthatóság összefoglalása.....	67
3.7.	Az új módszer eredményei, teljesítőképessége.....	69
3.7.1.	Eredmények mérőszámai .....	69
3.7.2.	Szimulációs tesztek.....	70
3.7.3.	Valós szenzorhálózatban végrehajtott tesztek .....	74
3.7.4.	Elért eredmények összefoglalása .....	76
	<i>Negyedik fejezet</i> .....	78
<b>4.</b>	<b>Összefoglalás .....</b>	<b>78</b>
4.1.	Új tudományos eredmények .....	78
4.1.1.	Ütemezési algoritmusok .....	78
4.1.2.	Outlier és esemény detekció .....	81
4.2.	Az új eredmények felhasználási területe.....	83
	<i>Ötödik fejezet</i> .....	85
<b>5.</b>	<b>Függelék.....</b>	<b>85</b>
5.1.	A disszertációban használt jelölések és rövidítések.....	85
5.1.1.	A második fejezet jelölésrendszere.....	85
5.1.2.	A harmadik fejezet jelölésrendszere .....	86

---

5.2. Rövidítések jegyzéke .....	88
<b>6. A szerző publikációi.....</b>	<b>90</b>
6.1. Folyóiratokban.....	90
6.2. Konferenciákon.....	90
<b>7. Bibliográfia.....</b>	<b>91</b>

*Első fejezet***BEVEZETÉS****1.1. Előszó**

A kommunikációs- és szenzoros technológia fejlődésének köszönhetően, számos olyan alkalmazás jelent meg, amely képes a mobilitási és flexibilitási igények kielégítésére. Ezek az alkalmazások elsősorban komplex rendszerek megfigyelésére és az ezekben előforduló krízisek detekciójára összpontosítanak.

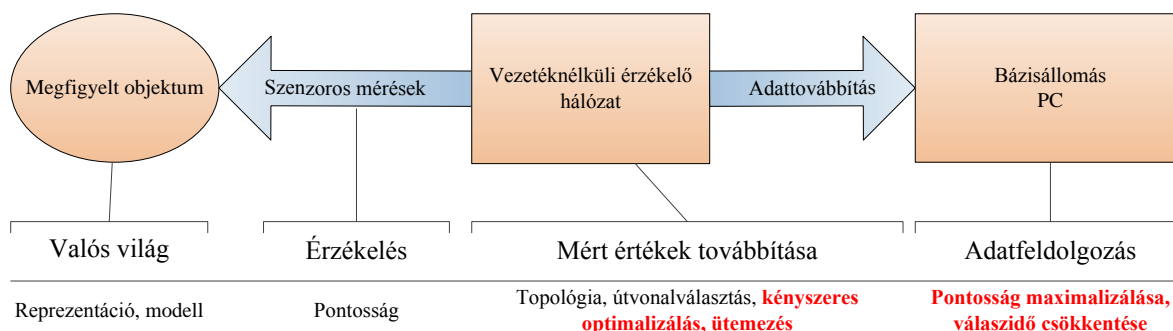
A hatékony rendszer-monitorozás eléréséhez ugyanakkor figyelembe kell venni, hogy az adatátvitelhez rendelkezésre álló erőforrások – úgymint felhasználható energiamennyiség, adóteljesítmény, sávszélesség – végesek, és ezek jelentős korlátozást jelentenek a vezeték nélküli rádiós hálózatban elérhető kommunikációs átvitel sebességére, a hálózat elemeinek élettartamára illetve a spektrális kihasználtságra vonatkozóan. A jelenlegi kutatások fókusza olyan monitorozási rendszerek kidolgozása, amelyek során sikerül a teljesítőképesség, pontosság és az erőforrások optimális kihasználásának elérése, az előzőleg említett kényszerek betartása mellett. A kérdés számos vezeték nélküli, mobil technológiát érint, melyeknél a kényszerek valamelyike fellép. Mivel ezeket a célkitűzéseket nem lehet általánosan – alkalmazástól függetlenül – megvalósítani, a disszertáció ezen feltételeket egy alkalmazás osztály során vizsgálja, és optimalizálja.

Az általam bemutatott algoritmusok és metódusok segítségével lehetővé válik a monitorozó rendszerekben a vezeték nélküli érzékelő (WSN) hálózatokkal az 1) optimális közeghozzáférés, valamint a 2) kritikus események kis hibaválószerűségű detektálása a mérési eredmények felhasználásával. A bemutatott algoritmusok és metódusok azonban nem kizárólagosan e területen alkalmazhatóak. A bemutatásra kerülő ütemező módszert alkalmazni lehet számos véges kapacitással rendelkező berendezés hozzáféréseinek ütemezésével, hívásengedélyezés ütemezésére telekommunikációs rendszerekben, memória és számítási kapacitás ütemezésére általános és pénzügyi-informatikai rendszerekben. Az esemény detekciós módszer során bemutatott technikák és matematikai megközelítések alkalmazhatók további mérési idősorok, vagy adatok feldolgozása során szokatlan értékek előrejelzésére és megtalálására, hiba előrejelzésre és becslésére valamint döntésmérettel kapcsolatos kérdésekre és problémákra is megoldást nyújthatnak. A séma adaptálható elosztott adatfeldolgozó rendszerek esetén is, például crowdsourcing alkalmazások során.

A következő szakaszokban az egyes téziscsoportjaimhoz kapcsolódó technikai háttér, az eddig elért eredmények és a nyitott kérdések rövid áttekintése következik.

### 1.1.1. A kutatási terület áttekintése

A disszertáció kutatási területeit a vezeték nélküli hálózatban előforduló technológiai és mérnöki problémák adják. A WSN hálózatok alkalmazásai során felmerülő feladatokat és lépéseket az 1.1. ábra szerinti blokkokra lehet felosztani. A disszertációban a kutatási területek közül a mért értékek továbbításához használható kommunikációs protokollal illetve a rögzített értékek feldolgozásával foglalkozom. Ezen belül vizsgáltam ütemezési és optimalizálási kérdéseket a csomagtovábbítás esetén, valamint események és szokatlan értékek detektálásával kapcsolatos problémákat az adatfeldolgozás területén.



1.1. ábra Kutatási területek, problémák és eredmények a WSN hálózatok esetén

A kidolgozott módszerek esetén minden esetben a teljesítmény maximalizálása a célom olyan módszerek bevezetésével, amelyek hatékonyan futtathatók párhuzamos architektúrán is.

### 1.1.2. Erőforrás kezelés és csomagütemezés

Monitorozó hálózatok nagy komplexitású adatforgalmazásának és számítási feladatainak lebonyolítására hatékony ütemező algoritmusok szükségesek, amelyek képesek véges kapacitás esetén is optimális megoldást biztosítani. Ennek megfelelően vizsgálok ezt a kérdést disszertációmban.

Az elosztott számítási rendszerek használatának ugrásszerű fejlődésével, a klasszikus erőforrás kezeléssel és ütemezéssel foglalkozó elméletek új alkalmazási területen is használhatóak. A létező megoldásokat, amelyeket a véges erőforrásokkal rendelkező gyárakban és kiszolgáló iparban használnak, adaptálta a telekommunikáció, a számítógép ipar és más nagy komplexitású számításokat igénylő diszciplínák is: többek között a kvantitatív biológia, a kémia és a pénzügyi szektor [1], [2]. A probléma általános megfogalmazása egy véges kapacitású erőforrás ütemezésének a megoldása. Figyelembe kell venni az igényekhez tartozó mennyiségeket, korlátokat, súlyozást, illetve el kell kerülni a feladatok időkorláton túli végrehajtását, az igények ütközését, továbbá az erőforrás kapacitását meghaladó ütemezését.

Sahni [3] egy  $\mathcal{O}(n \log nm)$  idejű algoritmust mutatott be, amely érvényes ütemezést állít elő betartva a határidőket, amennyiben az létezik  $n$  feladat és  $m$  erőforrás esetén. A módszert azonos funkciójú, de különböző számítási sebességgel rendelkező eszközökre is [4] kiterjesztették. Amennyiben nincs érvényes megoldása a feladatnak, úgy a probléma nehezebbé válik. Ekkor cél lehet a feladatok összes

késletetésének (*total tardiness*, TT) csökkentése, vagy a kapacitás túlterhelésnek a kiegyenlítése. (A késletetést a határidő és a tényleges befejezési idő különbségéből számítjuk minden egyes feladatra, ami nulla, ha a határidőig befejeződött a feladat végrehajtása.)

A maximális késletetés minimalizálására Lawler [5] megmutatta, hogy polinom időben megoldható, még precedencia feltételek megtartása mellett is. Martel [4] ezt konstrukciót felhasználva létrehozta az algoritmust. Mindamellet bizonyított, hogy a TT probléma NP nehéz, még egyetlen erőforrás esetén is [6]. Lawler [7] kifejlesztett egy pszeudo polinom idejű algoritmust a probléma megoldására, ugyanakkor a gyakorlati futásideje nem teszi lehetővé, hogy valós körülmények között használni lehessen. Azizoglu et al. [8] vizsgálta a preemptív TT probléma optimális megoldásának algoritmusát. Meghatároztak egy branch-and-bound (BB) algoritmust, azonban lassúsága miatt a gyakorlatban nem lehet használni tizenötnél több feladat ütemezésére.

A gyakorlati felhasználás során a feladatok fontosságát is figyelembe kell venni, amelyet egy-egy súly reprezentál, emiatt egy súlyozott TT (*total weighted tardiness*, TWT) problémát kell megoldani. Rachamadugu et al. [9] a TWT problémára ad heurisztikát, szintén nem-preemptív esetben.

A monitorozó rendszerekben a vezeték nélküli szenzorhálózatokkal történő adatgyűjtés energia és komplexitási kényszereket jelent, ezért ezekben a környezetekben az erőforrások optimalizálása nagyon fontos. Az erőforrás ütemezés problémája a WSN hálózatokban megjelenhet egyrészt a kommunikációs csatorna ütemezése, másrészt a kisteljesítményű feldolgozó egység használatának ütemezése során. A disszertációban vizsgált alkalmazás esetén a közeghozzáférés (MAC) és csomagütemezés a kritikus feladat, amire00 számtalan protokoll létezik a klasszikus hálózatok esetén, de ezek javarészt nem alkalmazhatóak WSN környezetben. A MAC protokollok két nagy csoportja ismert: egyrészt a versengés alapú, véletlen, ütközést érzékelő módszerek, másrészt az ütemezés alapú (TDMA) módszerek. A versengés alapú protokollok a keletkezett csomagot azonnal megkísérlik továbbküldeni, így az áteresztőképességet növelni és a végpontok közötti késletetést minimalizálni. (Ez a feladat ekvivalens a TT problémával.) A TDMA protokollok előnye az energiahatékonyság, mivel nincs szükség a csomagütközések detektálására, újraküldésre, illetve csak a számára meghatározott időrésben ébred fel a node. TDMA esetben a determinisztikus adatforgalom és feszítőfa ismerete nagy előnyt jelent. Ez az előny a monitorozó rendszereknél gyakran megjeleni.

A csomagütemezés megoldására energialimitált környezetben az S-MAC [11] a node-ok időnkénti kikapcsolásával, a B-MAC [12] pedig alacsony fogyasztású ébredési stratégia használatával alkalmazkodik. A B-MAC továbbfejlesztése az X-MAC [13], mely az áthallás következtében fellépő energiafelhasználást igyekszik csökkenteni.

A TDMA protokollok esetén a végpontok közötti késletetés jelentősen megnő, a hálózat áteresztőképessége alacsony. A hagyományos TDMA protokollok késletetése az időrés kiosztás változtatásával javítható [14]. Ezek közül a TreeMAC protokollt [15] WSN környezetre tervezték, figyelembe véve az egyes node-ok sáv szélesség igényét. Az előző két protokoll-család előnyeit ötvözve kompromisszumos megoldások is születtek, amelyek a több rétegű (cross-layer)

protokolltervezés eredményei. Goldsmith [16] és Jurdak [17] az útvonalválasztásra, közeghozzáférésre és az adóteljesítményre közösen optimalizált megoldást ad.

A fentiek alapján nem létezik olyan, a gyakorlatban is alkalmazható problémák megoldására alkalmas algoritmus, amely a TWT problémát kezeli. Továbbá kérdés, hogy a létező WSN MAC protokollok teljesítmény növelhető-e vagy sem.

**A disszertáció második fejezetében az erőforrás ütemezés problémáját vizsgálom az egyenletes terhelés, a TT és a TWT feladata, valamint WSN hálózatok esetén. Olyan módszert mutatok be, amely a probléma megoldására közel optimális eredményt ad, ugyanakkor egy olyan a gyakorlatban is használható polinom idejű heurisztikával oldom meg, amely sokprocesszoros architektúrákon hatékonyan futtatható.**

### 1.1.3. Outlier- és esemény detekció

Monitorozó hálózatok esetén a megfigyelt területen bekövetkező események és krízisek detektálása kulcsfontosságú probléma, amelynek megoldására több módszer is létezik. Ugyanakkor WSN környezetben nemcsak a pontos detekciónak, hanem az energiahatékonyságnak is nagy szerepe van.

Az esemény detekciót gyakorta vissza lehet vezetni más adatelemzési feladatra, például eltérő, szokatlan értékek keresésére, ami az outlier detekció feladata. A valós időben futó, idősorokon futtatható magas detekciós rátával rendelkező eljárások jelenleg is kutatott módszerek, azonban megfelelő megoldás pillanatnyilag nem elérhető el.

A létező outlier detekciós módszerek három csoportba oszthatóak a meglévő információk és modellek alapján: i) Nem felügyelt módszerek, mely esetben semmilyen feltételezésünk nincsen a mérési értékekről; ii) Felügyelt klasszifikáció, amikor a normál adatokra és a kiugró értékekre egyaránt létezik modell; iii) Részben felügyelt módszerek, ilyenkor kizárólag a normális adatokra létezik modell. A WSN hálózatok esetén a nem felügyelt módszerek alkalmasak a használatra, mivel a mérendő objektumról, valamint az egyes bekövetkezendő eseményekről, vagy hibás mérésekről gyakran körülményes, vagy lehetetlen megfelelő modellt felépíteni, megfigyelni. A klasszikus módszerek között nagyon sok olyan parametrikus, modell alapú létezik, amelyek az előzőek miatt csak pontatlan eredmények mellett használhatóak [18], [19], [20]. Helyettük több olyan algoritmust dolgoztak ki, amelyek adaptív eljárásokon alapulnak. Ezek közül egy fontosabb csoportot képvisel a mesterséges neuronhálózaton alapuló módszerek családja, amelyek jó detekciós aránnyal rendelkeznek, könnyű az adaptálhatóságuk és viszonylag alacsony a számítási komplexitásuk [21], [22], [23].

A WSN hálózatokban alkalmazott esemény detekciós algoritmusok közül az *SVM alapú megoldások* csoportjába tartozik Havinga et al. [24], [25] Quarter Sphere SVM módszere, amely a node-ok mérését SVM segítségével klasszifikálja outlier és nem outlier értéként. Az SVM-ben a klasszifikáció negyed hipergömb alkalmazásával történik. Bezdek et al. [26] Centered Hyperellipsoidal SVM (CESVM) módszere a negyed hipergömb helyett negyed hiperellipszoidot használ az SVM-ben, ezzel is

pontosabbá téve a klasszifikációt. További módszer a *neurális hálózat alapú klasszifikátor*, amely egy olyan kétfázisú módszer [23], ahol az első fázisban egyes szenzorok végrehajtanak egy klasszifikációt, amelynek eredményeit továbbküldve a bázisállomásra (BS) a második fázisban azok fúziójával, egy második klaszterezéssel történik az esemény detekció. A *szavazás-alapú módszerek* esetén az egyes node-ok lokálisan létrehoznak egy döntési fát, majd a döntésük eredményét a BS szavazás-alapon (többségi, vagy súlyozott) feldolgozza, és detektál eseményt. *Szabály és mintázat illesztés alapú módszerek* esetében az eseményeket előre definiáljuk, vagy adatbányászati eszközökkel meghatározzuk, és ezekre a szabályokra keresünk illeszkedéseket. Ezzel bonyolult feltételekre is illeszkedő események és eseménysorozatok detekciójára nyílik lehetőség, viszont a szabályok meghatározása gyakran nem automatikus [27]. Az illesztési, illetve döntési algoritmusra fuzzy logikát felhasználó megoldás is született [28], amely az esemény valószínűségét is kezeli. A *feature extraction alapú módszerekkel* lehet az átvitt adatmennyiséget mérsékelni, mivel az egyes szenzor node-ok előfeldolgozó műveleteket hajtanak végre a méréseken és azok eredményét juttatják el a BS-re [29]. Ezeknél a módszereknél az átvitt adatmennyiség csökkentésével az energiafogyasztás mérsékelhető.

Mindezen módszerekben azonban nem fektetnek nagy hangsúlyt az energiahatékonyságra, továbbá kevés az olyan módszer, amely a szenzorok méréseinek térbeli és időbeli korreláltságában rejlő információt is felhasználja.

**A disszertáció harmadik fejezetében a monitorozó WSN hálózatok problémáját vizsgálom, olyan algoritmust bemutatva, amely egyrészt magas detekciós rátával rendelkezik, másrészt a rádiós csomagok számának csökkentésével energiahatékony is. Az algoritmus további fontos tulajdonsága a gyors kiértékelési képesség, amelyet sokprocesszoros architektúrára való adaptálással oldok meg.**



## 1.2. A disszertáció célkitűzései

A disszertáció célkitűzése olyan algoritmikus eszközök létrehozása, amelyekkel monitorozó vezeték nélküli hálózatban lehet az esemény- és krízis detekciót végrehajtani magas detekciós ráta mellett, energiahatékonyan, az erőforrások optimalizálásával.

A kidolgozott eljárások és tézisek kutatási, WSN technológiai és általános felhasználási területeit az 1.1. táblázat foglalja össze.

1.1. táblázat A disszertációban bemutatott kutatási területek és eredmények összefoglalása

<i>Kutatási terület</i>	<i>WSN felhasználás</i>	<i>Eredmények további felhasználása</i>
<b>1. téziscsoport</b> Erőforrás optimalizáció, ütemezés	Közeghozzáférési protokollok, Csomagütemezés	Telekommunikáció, Számítási erőforrások ütemezése, Pénzügy, Kvantitatív diszciplínák
<b>2. téziscsoport</b> Esemény detekció, idősorok analízise	Szenzoradatok kiértékelése, mérési hibák, események detektálása	Outlier detekció idősorokon; Időben és térben korrelált adatok elemzése; Döntési rendszerek; Smart Metering és Smart Grid, Crowdsourcing

## 1.3. A disszertáció felépítése

A disszertáció következő, második fejezetében az erőforrás ütemezés problémáját tárgyalom a létező módszerek és megközelítések ismertetésével, majd az általam kidolgozott algoritmusokkal. A bemutatásra kerülő algoritmusok közül elsőként az erőforrás ütemezés általános problémájával foglalkozom, majd annak WSN hálózatokra történő adaptálását ismertetem. Ebben a fejezetben az első téziscsoportomban leírt eredmények kerülnek összefoglalásra. A harmadik fejezetben az adatsorokban jelenlevő kiugró értékek detektálásának problémájával foglalkozom. A meglévő megoldások mellett bemutatom a saját algoritmusomat, amelyet követően az arra épülő esemény detekciós módszert írom le. Ezek a módszerek a második téziscsoportban bemutatott eredményeim. A dolgozat negyedik fejezetében összefoglalom a bemutatott eredményeket, az ötödik fejezetben pedig függelék, a jelölésrendszer és rövidítések összefoglalása, publikációim és a bibliográfia található.

*Második fejezet***ERŐFORRÁSOK ÜTEMEZÉSE**

Ebben fejezetben erőforrás ütemezési problémákkal foglalkozom. A véges kapacitású erőforrások különböző kényszerek melletti optimális ütemezésére vonatkozó algoritmusok bemutatását megelőzően, az azokra vonatkozó modellek és a már létező algoritmusok ismertetésére kerül sor. Az általános, kényszeres erőforrás optimalizálás mellett tárgyalom a WSN környezetben történő alkalmazását a módszernek, amely esetben a kommunikációs csatorna ütemezésével foglalkozom.

**2.1. Bevezetés, technológiai motivációk és nyitott kérdések**

Az ismertetésre kerülő új eljárások fejlesztésekor az általános motiváció a monitorozó hálózatok számára olyan algoritmusok létrehozása, amelyek egyaránt figyelembe veszik a limitált számítási képességet valamint a rendelkezésre álló véges energiát. Mindezek mellett kimagasló eredményt is nyújtanak. A véges kapacitású erőforrások ütemezése esetén a feladat az optimális ütemezés megoldása, figyelembe véve az igényekhez tartozó mennyiségeket, korlátokat, súlyozást, illetve minimalizálva a késleltetést, elkerülve az igények ütközését valamint a közös erőforrás kapacitását meghaladó ütemezést. Az ütemezés egy speciális esete a vezeték nélküli monitorozó hálózatokban történő alkalmazás. Ebben a környezetben a vezeték nélküli kommunikáció ütemezésére szükségesek olyan módszerek, amelyek a kommunikációs csatorna természetéből fakadó kényszerek figyelembe vétele mellett az előre definiált cél szerinti optimális csomagütemezést határozzák meg. A célok között szerepelhet: a nagy áteresztőképességű csomagátvitel; a minimális energiafelhasználás; a határidő betartása melletti csomagtovábbítás; vagy az egyenletes rádiós terheltség.

A véges erőforrások kényszeres ütemezésének témakörében jól definiált problémaosztályok léteznek, amelyek egy része bizonyítottan NP nehéz. A feladatokra számos létező heurisztikus módszert lehet használni, azonban például a jobb eredmény elérésére alkalmas módszerek gyakran csak kisméretű problémák megoldására alkalmasak. A WSN környezetben történő csomagütemezés megoldására két protokoll-család ismert. Azonban a protokollok előnyeit egyesítő módszerek száma alacsony, teljesítőképességük mérsékelt. Ebben a fejezetben olyan módszert mutatok be, amely a meghatározott kényszerek melletti ütemezési és csomagütemezési probléma megoldására közel optimális eredményt nyújt, továbbá a gyakorlatban is használható polinom idejű heurisztikával oldom meg a feladatot.

A létező kihívások és problémák, amelyre ebben a fejezetben megoldásokat mutatok be a következők:

- *Az egyenletes terhelés; preemptivitás minimalizálása; minimális késleltetés; minimális súlyozott késleltetés célfüggvényei esetén milyen algoritmussal lehet meghatározni az ütemezési feladatok optimális megoldásait?*

- *Hogyan lehet az ütemezési módszereket módosítani illetve kiterjeszteni a WSN csomagütemezés feladatának megoldására? Olyan algoritmus szükséges, amely figyelembe veszi a WSN szűkös erőforrásait.*
- *Milyen módon lehetséges az eredmény polinom időben történő meghatározása, illetve lehetséges-e növelni az algoritmus futási sebességét?*

*Létrehoztam egy modellt, amelyben a feladat az egyes problémák esetén meghatározott célfüggvények mellett, a peremfeltételek betartásával az optimális ütemezési mátrix meghatározása.*

## 2.2. Hagyományos megközelítések

Röviden áttekintem a szakirodalomban elérhető, és a teljesítmény analízis során az összehasonlítás alapjául szolgáló módszereket. Külön mutatom be az erőforrás ütemezés általános feladatát és a WSN csomagütemezési protokollokat.

### 2.2.1. Jelenleg használt erőforrás ütemező módszerek

A disszertációban, a 2.3 szakaszban megfogalmazott modellben felírható ütemezési problémákat vizsgáltam meg. A feltételezés szerint a feladatokat végrehajtás közben le lehet állítani és folytatni később (preemptivitás), és a kiszolgáló számítási eszközök azonosak, valamint a végrehajtási idő alatt számuk állandó. A problémaosztályban szereplő egyes problémák megoldására több algoritmust is kifejlesztettek. A TT és TWT feladat megoldására számos közelítő heurisztika létezik [10], amelyeket az algoritmusok teljesítményének vizsgálatkor az összehasonlítás alapjául szokásosan használnak. Ennek megfelelően az EDD és a WSPT módszer működését ismertetem.

Az *EDD (Earliest Due Date)* közelítő megoldás során az ütemezendő feladatokat azok határideje szerinti sorrendbe rendezzük a legrövidebb határidővel kezdve. A feladatokat hozzárendeljük az azokat feldolgozó véges erőforráshoz a sorrend és a kapacitás figyelembe vételével úgy, hogy a feldolgozó minden időpillanatban a megengedett maximális terhelésnek megfelelő feladaton dolgozzon. Pinedo megmutatta [2], hogy az EDD heurisztika az optimális megoldást találja meg, amennyiben egyetlen feldolgozó egység áll rendelkezésre, illetve ha maximális késleltetést szeretnénk minimalizálni. Azonban az EDD nem veszi figyelembe az egyes feladatok fontosságát, így a TWT problémára nem nyújt kellően jó megoldást.

A súlyokat is figyelembe vevő közelítő heurisztika a *WSPT (Weighted Shortest Processing Time)*, amely az EDD módszerhez hasonlóan a feladatokat sorba rendezi figyelembe véve az egyes feladatok méretének és súlyának hányadosát. A legkisebb hányadostól kezdve az egyes feladatokat a feldolgozó egységhez rendeljük, az EDD módszer esetén ismertetett feltétel megtartása mellett. A módszer egy variánsa esetén a sorba rendezés során a határidőt is figyelembe kell venni úgy, hogy a feladat méretének és határidejének szorzatát osztjuk a feladat prioritásával.

A későbbi teljesítmény analízis során a referencia módszert képviseli az a random módszer, amely az ütemezést véletlen módon állítja elő, mégpedig a kapacitási feltétel betartásával. Ennél a módszernél, amikor a végrehajtó egységben szabad kapacitás keletkezik, akkor a még fennmaradó feladatok közül egyet véletlenszerűen kiválaszt futtatásra.

### 2.2.2. Jelenleg használt csomagütemező protokollok

Az erőforrás ütemezésnek – a monitorozó vezeték nélküli hálózatok esetén – az egyik fontos vetülete a közeghozzáférés ütemezése. Ebben a szakaszban a legfontosabb csomagütemezési módszereket ismertetem.

A MAC protokollok két fontos, egymástól eltérő filozófiát követő csoportja ismert: versengés alapú, véletlen és az ütközést érzékelő módszerek; illetve az ütemezés alapú (TDMA) módszerek. A versengés alapú protokollok a keletkezett, illetve átküldendő beérkezett adatsomagot azonnal igyekeznek továbbküldeni a szülő node-nak, a BS felé. Ezzel a megközelítéssel a teljes hálózat áteresztőképességet maximalizáljuk, aminek következtében a feladat könnyen megfogalmazható TT problémaként. Szemben ezzel a TDMA módszerek elkerüli a csomagütközést és az újraküldések okozta többletenergia-felhasználást, azonban ezért a nyereségért az áteresztőképesség alacsony mértékével fizetünk.

A versengés alapú protokollok közül elsőként a *B-MAC* [12] protokollt részletezem. Ez a protokoll a Mica2 [31] típusú node-ok segítségével terjedt el, az ezen node-ból épült hálózatok szabványos protokollja. (A disszertáció teljesítmény analíziseiben leírt fizikai hálózatát ilyen típusú node-ok alkotják.) A B-MAC protokoll használata során a küldő node egy hosszú preambulum részt illeszt a hasznos adatsomagra, amely mindig hosszabb, mint a node-ok alvási ideje. Ennek köszönhetően a preambulum adása alatt a cél node bizonyosan felébred és fogadja az üzenetet. A protokoll akkor hatékony, ha a forgalom ritka. Amikor gyakran kell rövid üzeneteket átküldeni, akkor arányaiban nagy mennyiségű energia vész kárba, mivel a preambulum hossza több, mint a tényleges adatrész. A protokoll esetén további hátrány az, hogy a rejtett terminál problémáját nem oldja meg a B-MAC [32], [33]. Az *X-MAC* protokoll [13] a B-MAC továbbfejlesztése, amely az áthallás következtében fellépő energiafelhasználást igyekszik csökkenteni.

Egy másik gyakran alkalmazott versengés alapú protokoll az *S-MAC* [11], amely esetében a szomszédos node-ok egymással periodikusan szinkronizálnak. Ennél a protokollnál minden egyes node egy előre meghatározott ütemben fogadó állapotba kapcsol, és várakozik a szomszédos node-ok szinkronizációs, vagy küldési kísérletére. Az aktív intervallum így szinkronizációs, és adatkommunikációs periódusra osztható szét. Az adatkommunikációs szakaszban RTS/CTS megoldás segít abban, hogy a csomagütközést elkerüljék a node-ok. Ennek megfelelően a B-MAC protokollhoz hasonló módon az alvó periódus hosszával szabályozható az aktív/inaktív arány. Az S-MAC és B-MAC protokollok tehát az energiafogyasztás minimalizálásával nem foglalkoznak, kizárólag az end-to-end késleltetést igyekeznek csökkenteni.

*TDMA protokollok* esetén az időszeltek triviális kiosztása jelentősen megnöveli a végpontok közötti késleltetést. Több node esetén kihívást jelent az optimális TDMA kerethossz és az időszeltek node-ok közötti felosztásának a meghatározása. (Egy node több időszelket is birtokolhat, továbbá lehetséges, hogy egy időszelket egyidejűleg több node által is felhasználhat párhuzamosan anélkül, hogy csomagütközés lenne a hálózatban.) A Varaiya által bemutatott TDMA ütemezés [34] a késleltetés (illetve ütemezési idő) minimalizálását tűzte ki célul. Ez a módszer a következő két fázisból áll. Elsőként – figyelembe véve az interferenciát – az egyes node-okhoz különböző színeket rendel a lehető legkevesebb szín felhasználásával. (Például a [35]–ben ismertetett RAND algoritmussal.) Ezt követően azonos időszeltekben az azonos színű node-ok kerülnek, valamint további olyanok, amelyek interferencia szempontjából nem zavarják egymást. Goldsmith [16] módszere minimális energiafogyasztású ütemezést valósít meg úgy, hogy először az útvonal-választási fa szerinti levelek csomagtovábbítását ütemezi a lehető legtöbb szimultán kommunikációt megengedve, majd ha az összes levél node-on keletkezett csomagot továbbítottak a szülő felé, akkor a levél node-ok elhagyásával a következő, egy szinttel kisebb gráfban folytatódik a csomagok ütemezése. Ezzel a módszerrel az adatátvitel burst jellegű lesz, és a node-ok az összes továbbítandó csomagot egyszerre adják át szülőjüknek. A TDMA protokollok közül a teljesítmény analízis során több időszeltek kiosztási stratégia szerinti implementációnak az eredményét is feltüntettem a [36]-ben bemutatottak szerint, illetve a további protokollok eredményei is szerepelnek: PEDAMACS (Varaiya et al, [37]), és a TreeMAC [15].

A bemutatott két protokoll-család előnyeit ötvözve több kompromisszumos megoldás született, ami a több rétegű (cross-layer) protokolltervezés eredménye. Goldsmith [16] és Jurdak [17] az útvonalválasztásra, közeghozzáférésre és adóteljesítményre közösen optimalizált megoldást ad.

### 2.3. Az ütemezési probléma modellje

Ebben a szakaszban egy alapmodellt ismertetünk, amelyet felhasználva több ütemezési probléma felírható. A bemutatásra kerülő építőkockák segítségével a TT, TWT probléma valamint a WSN rendszerek csomagütemezési feladatai felírhatók.

Az erőforrás ütemezési probléma során adott  $J$  számú kiszorgálandó feladat, amelyet  $V$  kapacitású kiszorgáló egységnek kell feldolgoznia. Az egyes  $j$  feladatok diszkrét egységekre bonthatók fel, továbbá minden feladatnak ismert a mérete és minden egyes feladathoz tartozik egy határidő, aminek lejártá előtt szükséges annak végrehajtása. Az egyes feladatok méretét  $X_j$ , a hozzá tartozó határidőket  $K_j$  jelöli, ahol  $j = 1 \dots J$ . Az ütemezendő feladatok mérete és a határideje egyaránt egész értékek. Minden egyes időszelvényben külön-külön kell eldönteni, hogy egy adott feladat abban futtatásra kerül-e vagy sem. Ennek megfelelően egy  $j$  feladat ütemezését fel lehet írni a következő ütemezési vektorral:

$$\mathbf{c}_j \in \{0,1\}^L, \quad (2.1)$$

ahol az  $L$  jelöli a teljes ütemezés hosszát. A vektor minden egyes pozíciójában 1 vagy 0 érték van, amely jelzi, hogy az adott pozícióhoz tartozó időszelvényben a feladat ütemezve van-e futásra a kiszorgáló egységen. Ezeket az ütemezési vektorokat felhasználva egy  $\mathbf{C}$  ütemezési mátrix hozható létre, amelyben a sorok felelnek meg az egyes feladatok ütemezésének. A mátrix mérete így  $J \times L$ . Tekintsük az alábbi egyszerű példát:  $J = 6$ ,  $X_1 = 5$ ,  $X_2 = 5$ ,  $X_3 = 4$ ,  $X_4 = 3$ ,  $X_5 = 3$ ,  $X_6 = 7$ ,  $K_1 = 6$ ,  $K_2 = 8$ ,  $K_3 = 7$ ,  $K_4 = 6$ ,  $K_5 = 4$  és  $K_6 = 8$ . Az értékeket felhasználva egy lehetséges ütemezést a következőképpen lehet felírni:

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad (2.2)$$

Az előzőeknek megfelelően a mátrixban, ha a  $c_{j,i}$  érték 1, akkor a  $j$ -edik feladat egy egysége az  $i$ -edik időszelvényben kiszorgálásra kerül a feldolgozó egység által.

Egy ütemezési mátrix akkor érvényes, amikor az előírt peremfeltételek teljesülnek. Az ütemezési mátrix  $j$ -edik sorában található egységek száma  $X_j$ , illetve egyetlen időpillanatban sincs több mint  $V$  darab ütemezett feladat, valamint a feladatok befejezési ideje az előírt  $K_j$  időkorlátan belül van a  $j$ -edik node esetén. A megoldás mindkét esetben hibásnak számít, egyrészt, ha az ütemezésben bármely feladat csak részben kerül végrehajtásra, másrészt, ha több időszelvény rendelünk a feladathoz, mint amekkora a feladat maga:

$$\forall j = 1, \dots, J : \sum_{l=1}^L c_{j,l} = X_j. \quad (2.3)$$

Az egyes időszelvényekhez tartozó ütemezésben nem lehet több feladatot végrehajtani, mint a végrehajtó egység kapacitása, ami a következő feltétel betartásával kerülhető el:

$$\forall l = 1 \dots L: \sum_{j=1}^J c_{j,l} \leq V. \quad (2.4)$$

Az egyes feladatokhoz kapcsolódó végrehajtási határidőt be kell tartani, azaz a megoldás nem tartalmazhat határidőn túli ütemezést:

$$\forall j = 1, \dots, J: \sum_{l=K_j+1}^L c_{j,l} = 0. \quad (2.5)$$

A kiterjesztett modell esetén az egyes ütemezendő feladatokhoz prioritással is rendelkeznek. Ezeket a prioritásokat az egyes ütemezések meghatározásánál figyelembe kell venni. A súlyokat a következőképpen jelöljük:

$$\mathbf{w} = \{w_1, w_2, w_3, \dots, w_J\} \in \mathbb{R}^N, w_j \geq 0, \forall j = 1, \dots, J. \quad (2.6)$$

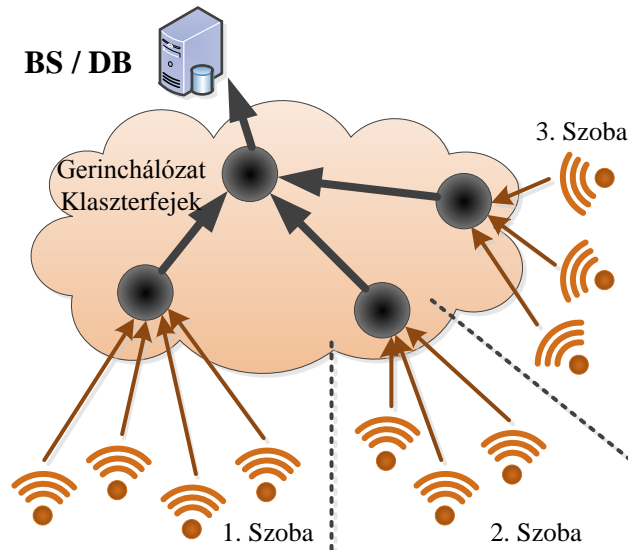
A (2.3), (2.4), (2.5) összefüggések felhasználásával a TT, és egyenletes terhelés problémája írható fel. Az előzőeket kiegészítve a (2.6) összefüggéssel a TWT probléma megfogalmazását kapjuk meg.

Az erőforrás ütemezés modelljét a vezeték nélküli érzékelő hálózat tulajdonságaihoz lehet igazítani, ahol a közös erőforrás a kommunikációs csatorna. A WSN környezetben a csatorna kapacitása  $V < \infty$ . Ennek megfelelően az egyidejűleg továbbítható csomagok száma korlátos. Az egyes  $j$  csomagforrások esetén a csomagok számát  $X_j$ , és a csomagküldés végső határidejét  $K_j$  jelöli.

A WSN modell egyszerűbb változata esetén az egyes forrás node-ok a csomagjaikat egyetlen lépésben küldik el a klaszterfejhez, a szülő node-hoz, amelynek a véges kapacitása az egyes időablakok során fogadható csomagok száma. A hálózat elrendezését a 2.1. ábra mutatja be.

Egy  $j$  node-hoz tartozó csomagküldési stratégiát a következő vektor reprezentálja:

$$\mathbf{c}_j \in \{0,1\}^L. \quad (2.7)$$



2.1. ábra Egyszerűsített WSN modell klaszterfejekkel

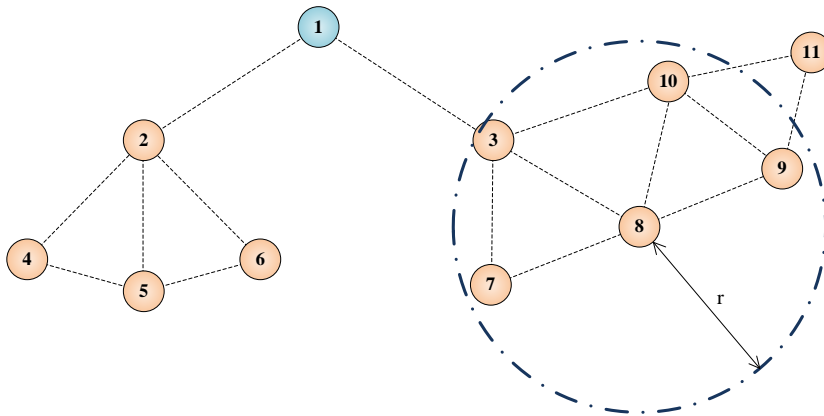
A vektorokat felhasználva egy  $\mathbf{C}$  ütemezési mátrix hozható létre, amelyben a sorok felelnek meg az egyes feladatok ütemezésének. A csomagütemezés esetén szintén meg kell valósulnia a

peremfeltételeknek, amelyek az ütemezési mátrix validitását garantálják. A feltételeket a (2.3), (2.4) és (2.5) egyenletek definiálják.

A kiterjesztett WSN modell esetén egy teljes vezeték nélküli hálózat adatforgalmazását optimalizálom. Minden egyes node esetén adott az átküldendő csomagok száma és a forgalmazás határideje. Az ütemezést – hasonlóan az eddigiekhez – a  $\mathbf{C}$  ütemezési mátrix írja le, amelyben az egyes sorok az egyes node-okhoz tartozó ütemezést tartalmazzák. A hálózatot egy  $G = (V, E)$  irányított gráf modellezi, ahol  $V = \{v_1, v_2, \dots, v_N\}$  csúcsok a node-ok halmazát illetve  $E$  a lehetséges kommunikációs linkek halmazát írja le. Az élek halmazában olyan összeköttetések tartoznak, ahol  $\{v_i, v_j\} \subseteq V$  ott,  $e = (v_i, v_j) \in E$  akkor és csak akkor, ha a két node egymástól adótávolságra található. A  $G$  gráfhoz definiálható annak szomszédsági mátrixa:

$$\mathbf{A} = [a_{ij}]_{i,j=1,\dots,N}. \quad (2.8)$$

Ez a mátrix bináris,  $a_{i,j} = 1$  amennyiben az  $i$  és  $j$  node között húzódik él, tehát  $e = (v_i, v_j) \in E$ . Mivel a rádiós linkek adótávolság szempontjából szimmetrikusak és ezeket megbízhatónak feltételezem, az  $\mathbf{A}$  mátrix szimmetrikus. A mátrix fading modellek, vagy mérések segítségével határozható meg egy már telepített hálózat esetén. A 2.2. ábra egy topológia példát mutat be, amelyen jelöltem az egyes node-okat, valamint az azok között létező kommunikációs linkeket.



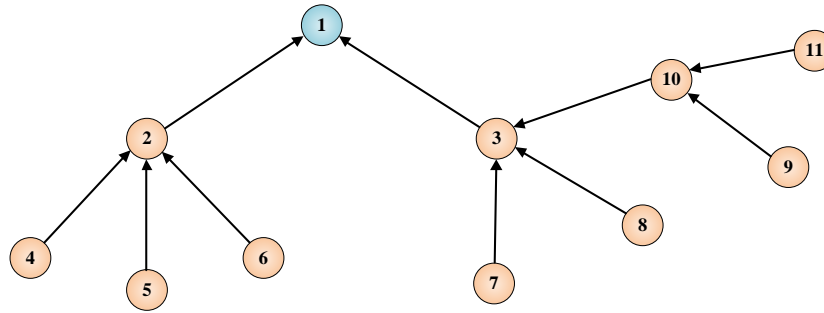
2.2. ábra Hálózati topológia egy példa hálózat esetén

Adatgyűjtő hálózat esetén az egyes node-ok a csomagokat a BS irányába küldik, amely az előző példában legyen a jelölt, 1-es node. A csomagok eljuttatásához szükséges útvonal meghatározását kereső algoritmusok segítségével lehet végrehajtani, ahol a gráfban az élekhez tartozó súlyokat a linkek mértékéből lehet meghatározni.

A routing határozza meg egyértelműen, hogy az egyes node-oknak melyik node a szülője. A routing-ot szintén egy mátrixszal tudjuk leírni:

$$\mathbf{R} = [r_{ij}]_{i,j=1,\dots,N}. \quad (2.9)$$





2.3. ábra Útvonalválasztás a példa hálózat esetén

A mátrixban  $r_{i,j} = 1$ , ha az  $j$  node-nak  $i$  node a szülője. Ennek megfelelően a routing mátrix egyértelműen meghatározza minden egyes csomag küldésekor a csomag forrását és címzettjét. Az előző példához kapcsolódóan egy lehetséges routing-ot mutat be a 2.3. ábra.

A vezeték nélküli hálózatban olyan csomagütemezésre van szükség, amely az egyik legnagyobb erőforrás pazarlást okozó jelenséget, a rádiós ütközéseket kiküszöböli. Ehhez az alábbi szempontokat figyelembe vevő interferencia modellel bővítettem a fentebb definiált hálózati modellt: [30]

- adott node egyidejűleg egy rádiós tevékenységet folytathat: ad, vagy vesz (half-duplex);
- egy node egyetlen csomagot forgalmaz egyszerre (single packet radio);
- ha egy  $i$  node fogad  $j$  node-tól, akkor egyetlen  $l$  node sem forgalmazhat, amely része az  $i$  node ütközési tartományának.

Az előző feltételek megtartása esetén a kézenfekvő megoldás az a TDMA alapú rendszer, amelyben minden egyes node számára kijelölünk egy időszelést, amikor adatot küldhet. Ez azonban lényegesen rontja az átviteli időt és növeli a késleltetést. Vezeték nélküli hálózatokban az ütközési és interferencia tartományok végesek, emiatt definiálható egy olyan környezet, amikor bizonyos node-ok kommunikálhatnak szimultán is. Ezt a következő interferencia mátrix bevezetésével lehet megoldani:

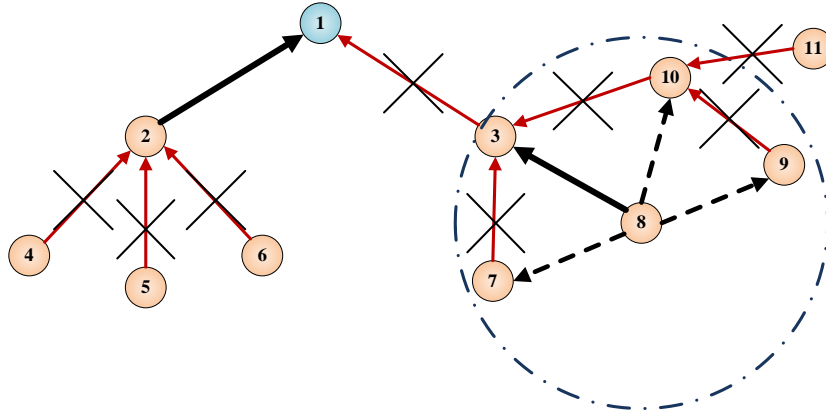
$$\mathbf{F} = \mathbf{A} + \mathbf{A}^2 - \text{diag}(\mathbf{A} + \mathbf{A}^2) \quad (2.10)$$

Az  $\mathbf{F}$  mátrix

$$\mathbf{F} = [f_{ij}]_{i,j=1,\dots,N} \quad (2.11)$$

ahol az  $f_{i,j} = 1$ , ha  $i$  node nem adhat egyidejűleg  $j$  node-dal. A 2.4. ábra ezt szemlélteti.

A példában szereplő 8. node kommunikációja mellett párhuzamosan tiltott: a vevő (3) node-nak adni, és fogadni (a 7, vagy 10 node-tól); valamint a vevő node-dal azonos ütközési tartományban levő node-nak (7, 9, 10) adni; továbbá a küldő node ütközési tartományában levő node-nak küldeni (11).



2.4. ábra Interferencia a példa hálózatban

Az előző modellt feltételeinek figyelembe vételével kell egy érvényes ütemezést létrehozni. A  $\mathbf{C}$  ütemezési mátrix – hasonlóan az egyszerűsített modellhez – meghatározza, hogy az egyes node-ok mikor kezdeményeznek adást, illetve implicit módon azt is, hogy a node-okon mikor fogadnak csomagot. Minden egyes  $j$  node  $k$ -adik ütemhez tartozó aktuális forgalmi állapotát a  $q_j(k)$  írja le. Kezdetben  $q_j(0) = x_j$ , ami minden egyes ütemben megváltozik. Az ütemezésnek megfelelően a forgalmi állapot a következőképpen írható fel:

$$\mathbf{q}(k+1) = \mathbf{q}(k) - c_{*,k} + c_{*,k} \mathbf{R} \quad (2.12)$$

Érvényes ütemezés létrehozásához az alábbi feltételeknek kell teljesülnie:

- Előírt csomagmennyiség átküldése a levél node-oknál, továbbá minden egyéb node a forgalmának megfelelő mennyiségű csomagot küldjön, az  $\mathbf{x}$  és  $\mathbf{R}$  adatoknak megfelelően:

$$\forall j: \sum_{l=1}^L c_{j,l} = p_j, \quad (2.13)$$

ahol a  $p$  jelöli egyetlen node forgalmát összegezve a node-hoz érkező összes csomagot

$$p_j = x_j + \sum_{l=1}^L c_{*,l} \mathbf{R}. \quad (2.14)$$

- A csomagokat egy node csak akkor küldje, ha az rendelkezésre áll, azaz a csomagok érkezési és továbbküldési sorrendjét meg kell tartani. Formálisan a következőnek kell teljesülnie:

$$\forall j, \forall k: C_{j,k} = 1 \rightarrow q_j(k) > 0. \quad (2.15)$$

- Ne legyen interferencia illetve ütközés a hálózatban, a sikeres csomagküldések garantálása érdekében

$$\forall j, \forall k, \forall l: \{C_{j,l} = 1 \wedge C_{j,k} = 1\} \rightarrow f_{l,k} = 0. \quad (2.16)$$

## 2.4. Új algoritmusok az ütemezési problémák megoldására

Az előzőekben bemutatott modell esetén felírható különböző ütemezési problémákra kidolgozott megoldásaim ismertetésére kerül sor ebben a szakaszban. Elsőként két egyszerű heurisztikát mutatok be, amelyek egy-egy speciális problémát oldanak meg. Ezt követően egy általánosabb apparátust ismertetek, amely kiterjeszhető bonyolultabb ütemezési feladatok megoldására is. A kiterjeszhetőséget az WSN hálózatokban történő alkalmazáson keresztül mutatom be. Végül a bemutatott módszerek numerikus megoldásával, illetve a megoldások hatékonyságával foglalkozom.

### 2.4.1. Új heurisztikus algoritmusok

Ebben a szakaszban olyan egyszerű algoritmusokat ismertetek, amelyek egy-egy specifikus ütemezési problémát optimálisan, vagy közel optimálisan képesek megoldani.

#### 2.4.1.1. Algoritmus a TWT probléma megoldására

Az összegzett súlyozott késleltetés minimalizálásának feladata esetén feltételezzük, hogy nem áll rendelkezésre elegendő számítási kapacitás a feladatok végrehajtásához, azonban a feladatokat a lehető legkevesebb késéssel kell ütemezni, a prioritásoknak megfelelően. A célfüggvény a következőképpen alakul:

$$\mathbf{C}_{opt} := \arg \min_{\mathbf{c}} \sum_{j=1}^J \left( w_j \sum_{l=K_j}^L c_{j,l} \right). \quad (2.17)$$

A TWT probléma megoldására alkalmas WSPT módszer módosításával létrehoztam egy olyan algoritmust, amelynek teljesítménye felülmúlja az egyszerű heurisztikákat.

---

#### LWPF( $\mathbf{X}, \mathbf{K}, \mathbf{w}$ )

---

Állítsuk sorba a feladatokat a súlyok szerint  $w_1 \geq w_2 \geq w_3 \geq \dots \geq w_J$

**FOR**  $i=1 \rightarrow J$

**WHILE**  $X_j > 0$

**IF**  $\sum_{k=1}^J c_{k,j} < V$

$c_{i,j} \leftarrow 1$

$X_j \leftarrow X_j - 1$

**END IF**

**END WHILE**

**END FOR**

**RETURN**  $\mathbf{C}_{LWPF}$

---

2.5. ábra LWPF heurisztika

A TWT probléma esetén a megoldás szempontjából azok az esetek a legrosszabbak, amikor legnagyobb súlyú problémák szenvednek késlekedést. Ezért kézenfekvő a feladatokat úgy ütemezni, hogy a legnagyobb súlyú problémákat ütemezzük először és így haladunk az egyre csökkenő súlyok

feladatok felé, feltöltve az ütemezési mátrixot. A kidolgozott algoritmust (*LWPF*, *Largest Weighted Process First*) a 2.5. ábra foglalja össze.

Az algoritmus gyors, azonban nem optimális az ütemezés, amelyet létrehoz. Az eredményen a futási idő elhanyagolható növelésével az alábbi módon sikerült javítani. Az LWPF eredménymátrixát véletlenszerűen, de a kapacitásra vonatkozó kényszer figyelembe vételével megváltoztatom, és az így létrejött megoldások közül a legjobbat tekintem végeredménynek. A 2.6. ábra illusztrálja az elgondolást.



2.6. ábra Perturbált LWPF módszer (PLWPF)

A perturbációs algoritmust tehát a felhasznált kapacitást nem befolyásolja, azonban az előírt feladatmennyiségekre vonatkozó kitétel sérülhet. Az ezeket a hibákat kijavító korrekciós algoritmust a 2.7. ábra mutatja be.

---

**REQUIRE  $X, w, K, V, C$**

---

**FOR**  $l = 1 \rightarrow L$

**WHILE**  $\sum_{j=1}^J c_{j,l} > V$

Távolítsunk el 1-est abból az  $i$  sorból a  $l$  oszlopban, ahol a sorhoz tartozó feladat súlya a legkisebb és  $c_{i,l}=1$

**END WHILE**

**END FOR**

**FOR**  $k = 1 \rightarrow J$

**WHILE**  $\sum_{k=1}^L c_{l,k} > X_l$

Távolítsunk el 1-est a  $l$  sorban levő  $i$  oszlopból, ahol az  $i$  az a jobbszélső oszlop, amelynél  $c_{l,i}=1$

**END WHILE**

**WHILE**  $\sum_{k=1}^L c_{l,k} < X_l$

Adjunk hozzá 1-est a  $l$  sorban levő  $i$  oszlophoz, ahol az  $i$  az a balszélső oszlop, amelyben nincs 1-es és a  $V$  kapacitást nem sértjük meg

**END WHILE**

**END FOR**

**RETURN C**

---

2.7. ábra A perturbációs módszer korrekciós algoritmus

Szimulációkat hajtottam végre az LWPF módszercsalád teljesítőképességének ellenőrzésére, amelyek során az alább leírt paraméterekkel dolgoztam. Matlab-ban véletlen méretű, súlyú és határidővel

rendelkező feladatokat generáltam a (2.18) szerint, ahol a  $\text{random}(\Gamma)$  függvény a  $\Gamma$  intervallumon generál egyenletes eloszlású véletlen számokat.

$$\begin{aligned} \mathbf{X}_j &= \text{random}([1, o_1]), \\ \mathbf{K}_j &= \mathbf{X}_j + \text{random}([o_1, 1.5 \cdot o_1]), \\ \mathbf{w}_j &= \text{random}([1, o_2]). \end{aligned} \quad (2.18)$$

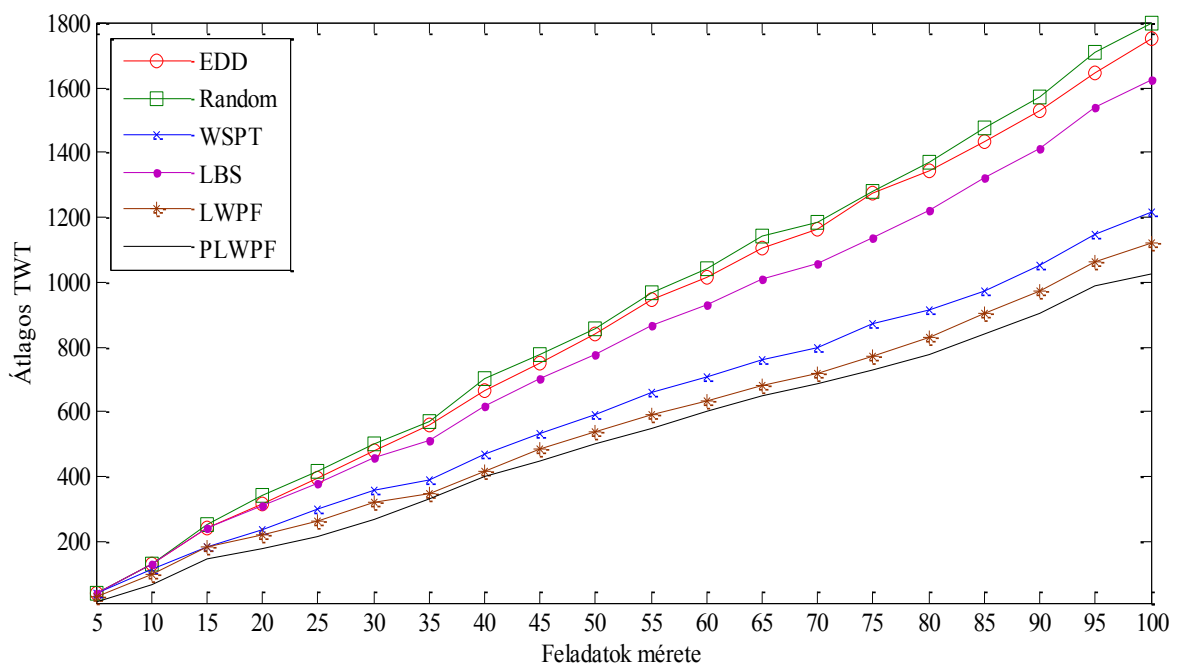
A szimulációimban a két konstans rendre 10 és 5 volt, továbbá a rendelkezésre álló számítási kapacitás a következők szerint lett meghatározva:

$$V = 0.25 \cdot J. \quad (2.19)$$

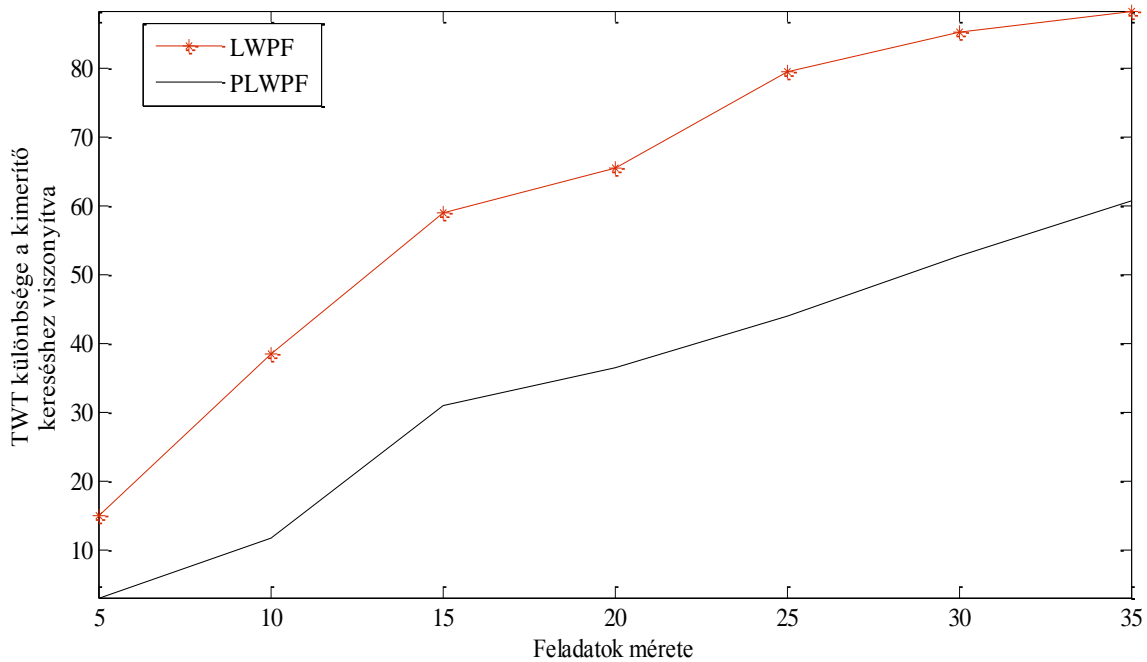
Az érték megválasztásának az oka az, hogy a probléma akkor oldható meg készletelés nélkül bizonyosan, ha a kapacitás egyezik az összes ütemezendő feladatok számával, vagyis  $V=J$ . Tehát a (2.19) garantálja, hogy lesznek olyan feladatok a megoldásban, amelyek bizonyosan készletetést szenvednek.

A szimulációkat több problémaméretre lefuttattam, minden egyes problémaosztály esetén ezerszer megismételve a futtatást és az így kialakult eredmények átlagát vettem figyelembe a kiértékelésnél.

A különböző feladtméretekre lefutott szimuláció eredményeit a 2.8. ábra mutatja be, ahol a TWT a megoldásokat jellemző metrika. Az LWPF és PLWPF módszerek a szokásos ütemezési heurisztikáknál jobb eredményt nyújtanak ebben az esetben. Látható, hogy a módszerek megoldásai a kimerítő kereséshez viszonyítva elmaradnak az optimális megoldástól, ugyanakkor a heurisztikus algoritmus futási ideje jelentősen kompenzálja ezt a hátrányt. A kimerítő kereséssel történő összehasonlítást a 2.9. ábra mutatja be.



2.8. ábra Heurisztikus módszerek megoldása a TWT problémára



2.9. ábra A két új heurisztikus algoritmus teljesítménye a kimerítő kereséssel szemben

#### 2.4.1.2. Algoritmus az egyszerűsített WSN probléma megoldására

A 2.3-as szakaszban bemutatott egyszerű WSN modell esetén a feladat egy olyan ütemezés létrehozása, amely a csomagmennyiségre (ez ekvivalens a feladatmennyiséggel) és a határidőre vonatkozó peremfeltételeket pontosan betartja. Itt nem engedhető meg a késleltetés, mivel az adott node összes hátralevő csomagjának elvesztését jelentené. Ugyanakkor az ütemezés létrehozásakor kapacitás-túlterhelés lehetséges, amit az ütemezés meghatározását követően az egy időszakokhoz tartozó, kapacitást meghaladó csomagok véletlenszerűen elhagyásával kompenzálom. A véletlenszerűség miatt a csomagvesztés egyenletesen eloszlik az összes node között. A módszer olyan monitoring rendszerben alkalmazható, ahol egy-egy mérési érték kiesése nem kritikus, azonban minden egyes node a működési határidejéig kell, hogy csomagot küldjön. (Ez a módszer csak alacsony túlterheltség mellett használható valós környezetben.) A célfüggvény az egyenletes csomagvesztési valószínűség garantálása, azaz:

$$P(\text{csomagvesztés az } i \text{ node-on, } l \text{ időpillanatban}) = \frac{\sum_{j=1}^J c_{j,l} - V}{\sum_{j=1}^J c_{j,l}}. \quad (2.20)$$

A megoldást a következőkben leírt algoritmus jelenti. A létrehozandó ütemezési mátrixban jobbról balra haladva minden egyes lépésben megkeressük azt a node-ot, amelynek a határideje az aktuális időszakhoz legközelebb van, de az aktuális időszakban még nincs ütemezett csomagja. A megtalált node-nak ütemezünk egy csomagot, majd folytatjuk az eggyel korábbi időszakban egy újabb csomag ütemezésével. Amennyiben elértük az első időszakot akkor újból a legutolsó időszakkal folytatjuk egészen addig, amíg van ütemezendő csomag. Az *LBS (Load Balancing Scheduling)* algoritmust a 2.10. ábra mutatja be.

**REQUIRE X,K**

$$L \leftarrow \max_l K_l$$

$$C \leftarrow \mathbf{0}_{J \times L}$$

$$S \leftarrow \sum_{j=1}^J X_j$$

**WHILE S > 0****FOR**  $l = L \rightarrow 1$ 

$$j \leftarrow \arg \min_j K_j \mid (l < K_j) \wedge (X_j > 0)$$

$$S \leftarrow S - 1, X_j \leftarrow X_j - 1$$

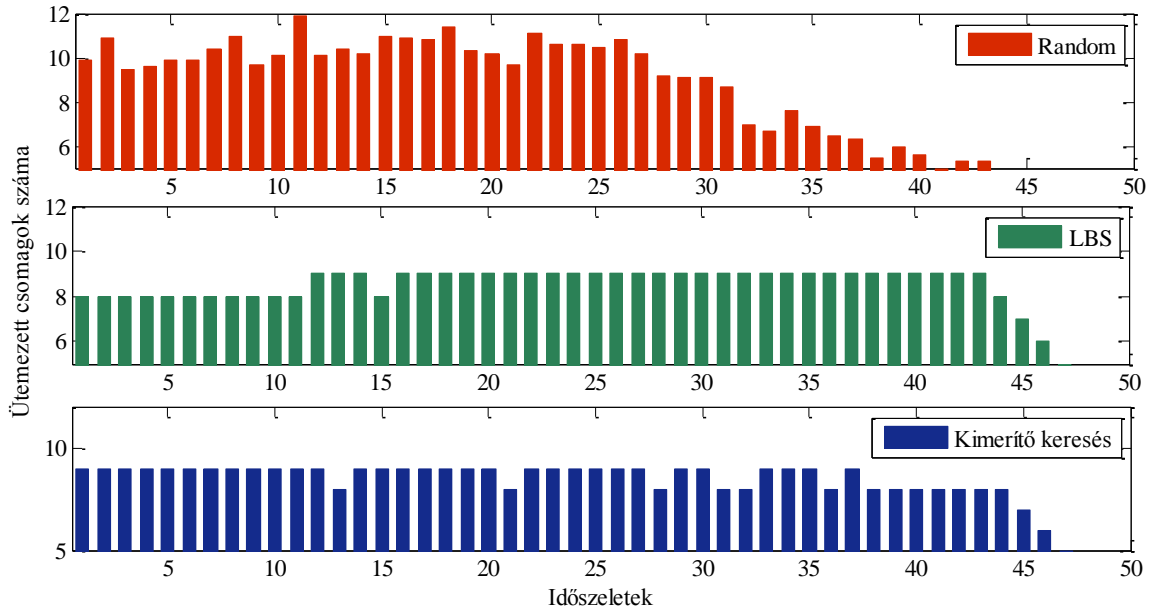
$$c_{j,l} \leftarrow 1$$

**END FOR****END WHILE****RETURN C**

2.10. ábra Az LBS algoritmus

A módszer teljesítőképességét az egyes időszeltekhez tartozó beütemezett csomagok számának egyenletességével lehet jellemezni. Ennek érdekében bevezetem az entrópia mérőszámot a következő képlet szerint:

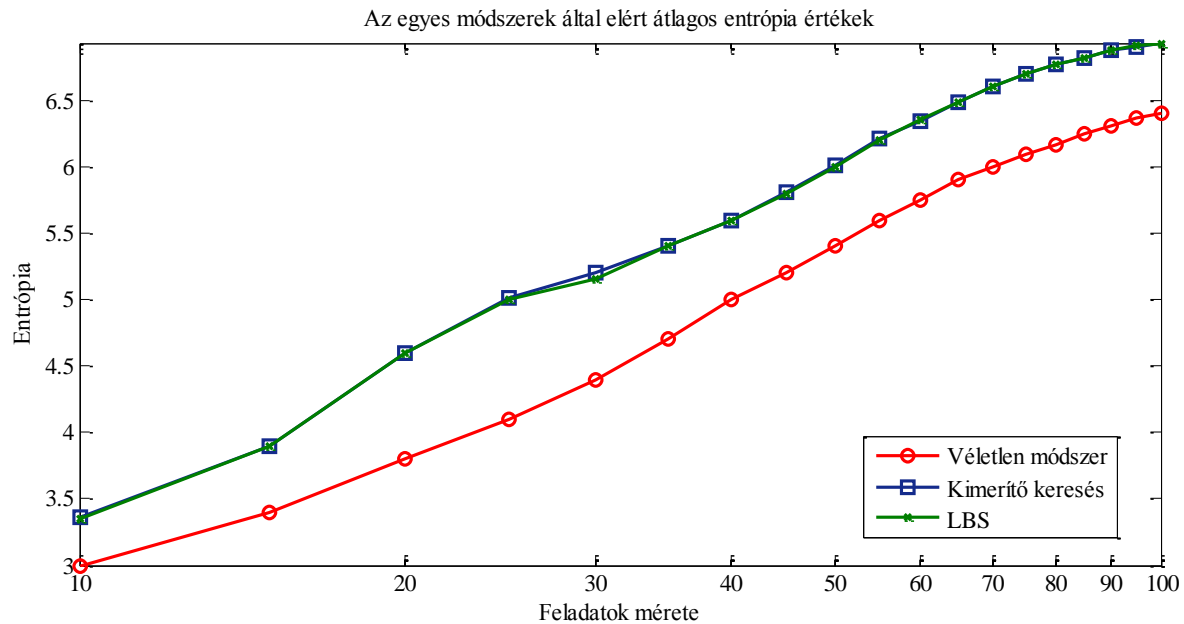
$$H(\mathbf{p}) = \sum_{l=1}^L -p_l \ln \left( \frac{\sum_{j=1}^J c_{j,l}}{\sum_{k=1}^L \sum_{j=1}^J c_{j,k}} \right). \quad (2.21)$$



2.11. ábra Az LBS módszer eredménye a kimerítő kereséssel szemben egy konkrét esetben.

A tesztet, hasonlóan az előző szakaszban ismertetett módszerhez, több véletlen sorsolt kiinduló adaton futtattam le, különböző problémaméret mellett. A kapott ütemezések entrópiáját meghatároztam, és az azonos feladatméretek melletti entrópiákat átlagoltam. Az ebből adódó

eredmények azt mutatták meg, hogy a bemutatott LBS algoritmus és a kimerítő kereséssel meghatározott optimális ütemezés azonosak. A 2.11. ábra egy konkrét ütemezési esetet mutat be az LBS, a kimerítő keresés és egy a peremfeltételek betartásával meghatározott 100 véletlen ütemezés átlaga esetén. A részletes összehasonlítást a 2.12. ábra ismerteti, ahol látható, hogy több problémaméret esetén is eléri az LBS módszer a kimerítő keresés által meghatározott eredményt.



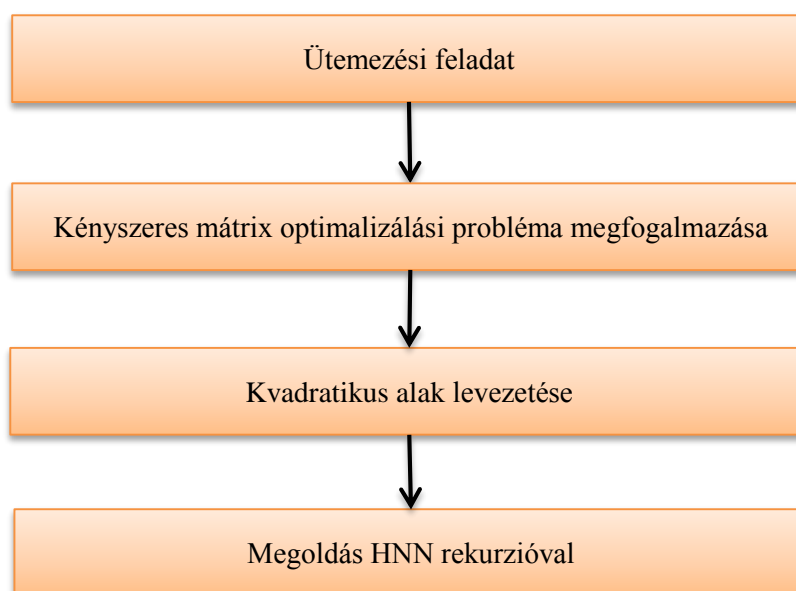
2.12. ábra Átlagos entrópia értékek a feladatméretek függvényében



### 2.4.2. Új, kvadratikus optimalizáláson alapuló megoldás az ütemezési feladatokra

A korábban megfogalmazott ütemezési problémák egy-egy kényszeres mátrix optimalizálási feladatot határoznak meg. Ebben a fejezetben ismertetem, hogy miként transzformáltam a kényszeres mátrix optimalizálási problémákat kvadratikus optimalizálási problémákká.

A módszer motivációját és menetét a 2.13. ábra mutatja be. A cél, hogy olyan kvadratikus alakokat határozzak meg, amelyet Hopfield neurális hálózat segítségével lehet minimalizálni.



2.13. ábra Ütemezési feladat megoldásának menete

Kvadratikus optimalizálás esetén adott egy szimmetrikus  $n \times n$  méretű  $\mathbf{W}$  mátrix és egy  $n$  hosszú  $\mathbf{b}$  vektor. Keressük azt az  $\mathbf{y}$  vektort, amely minimalizálja a következő kvadratikus alakot: [38]

$$f(\mathbf{y}) = \frac{1}{2} \mathbf{y}^T \mathbf{W} \mathbf{y} + \mathbf{b}^T \mathbf{y}, \quad (2.22)$$

a következő feltételek mellett:

$$\begin{aligned} \mathbf{A} \mathbf{y} &\leq \mathbf{v}, \\ \mathbf{B} \mathbf{y} &= \mathbf{u}. \end{aligned} \quad (2.23)$$

Amennyiben a feladat csak a lineáris feltételt tartalmazza, akkor a probléma megoldható [39]. Más esetekben, ha a  $\mathbf{W}$  mátrix pozitív definit, akkor az  $f(\mathbf{y})$  konvex függvény, és a probléma az ellipszoid módszer segítségével megoldható [40], amennyiben a  $\mathbf{W}$  mátrix indefinit, a feladat NP-nehéz [41].

Ahhoz, hogy a mátrix optimalizálási problémákat átranszformálhassam kvadratikus optimalizálási feladatokká, az egyes peremfeltételeket (például: (2.3), (2.4), (2.5)) és célfüggvényeket (például: (2.17)) az előző, (2.22) egyenlet szerinti alakra kell átalakítani. Továbbá, mivel a kvadratikus optimalizálás során az  $\mathbf{y}$  vektort optimalizálom, a  $\mathbf{C}$  ütemezési mátrixot át kell alakítani vektor alakra. A mátrix transzformációját soronkénti kiolvasással teszem meg a következők szerint:

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,L} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,L} \end{pmatrix} \rightarrow \mathbf{y} = (c_{1,1}, c_{1,2}, \dots, c_{1,L}, c_{2,1}, \dots, c_{2,L}, c_{N,1}, \dots, c_{N,L})^T. \quad (2.24)$$

A kvadratikus alakban szereplő súlymátrix és -vektor meghatározása minden egyes feladat esetén különböző, ezeket egyesével tárgyalom a következő szakaszokban.

#### 2.4.2.1. A TT probléma kvadratikus alakja

Elsőként a TT problémán keresztül mutatom be a három legalapvetőbb peremfeltétel kvadratikus formára átírt alakját. E probléma esetén a teljes késleltetést kell minimalizálnom, azaz a kapacitásra és feladatméretre vonatkozó (2.3), (2.4) peremfeltételek mellett a következő célfüggvényt kell használnom:

$$\mathbf{C}_{opt} := \arg \min_{\mathbf{c}} \sum_{j=1}^J \sum_{l=K_j}^L c_{j,l}. \quad (2.25)$$

Az ütemezési mátrix (2.24) formula szerinti átalakítását követően a peremfeltételeket a következőekben leírt módon transzformálok. A feladatméretre vonatkozó peremfeltétel (2.3), átalakítva a következőképpen írható fel:

$$\sum_{j=1}^J \left( \sum_{l=1}^L c_{j,l} - X_j \right)^2. \quad (2.26)$$

A peremfeltételhez tartozó súlymátrix és vektor meghatározásához az alábbi egyenlet megoldása szükséges:

$$\sum_{j=1}^J \left( \sum_{l=1}^L c_{j,l} - X_j \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_A \mathbf{y} + \mathbf{b}_A^T \mathbf{y}. \quad (2.27)$$

A (2.26) átalakítható a következő módon:

$$\begin{aligned} \sum_{j=1}^J \left( \sum_{l=1}^{K_j} c_{j,l} - X_j \right)^2 &= \sum_{j=1}^J \left( \left( \sum_{l=1}^{K_j} c_{j,l} \right)^2 - 2 \sum_{l=1}^{K_j} (c_{j,l}) X_j + X_j^2 \right) = \\ &= \sum_{j=1}^J \left( \sum_{l=1}^{K_j} c_{j,l} \right)^2 - 2 \sum_{j=1}^J \left( \sum_{l=1}^{K_j} c_{j,l} \right) X_j + \sum_{j=1}^J X_j^2. \end{aligned} \quad (2.28)$$

Ebből a harmadik tag

$$\sum_{j=1}^J X_j^2 = \text{konstans}, \quad (2.29)$$

ezért a minimalizálás szempontjából elhagyható, így az egyenlet végső alakja:

$$\sum_{j=1}^J \left( \sum_{l=1}^{K_j} c_{j,l} \right)^2 - 2 \sum_{j=1}^J \left( \sum_{l=1}^{K_j} c_{j,l} \right) X_j = \frac{1}{2} \mathbf{y}^T \mathbf{W}_A \mathbf{y} + \mathbf{b}_A^T \mathbf{y}. \quad (2.30)$$

A megoldás, ahol  $\mathbf{1}_{M,M}$  jelöli azt az  $M \times M$  méretű mátrixot, aminek minden eleme 1, valamint:

$$\mathbf{d}_{A_j} = X_j \cdot \begin{pmatrix} \mathbf{1}_{1 \times K_j} & \mathbf{0}_{1 \times L - K_j} \end{pmatrix}. \quad (2.31)$$

a következő:

$$\mathbf{b}_A = 2(\mathbf{d}_{A_1} \quad \mathbf{d}_{A_2} \quad \cdots \quad \mathbf{d}_{A_J}) \quad (2.32)$$

és

$$\mathbf{W}_A = 2 \begin{pmatrix} \mathbf{1}_{K_1 \times K_1} & \mathbf{0}_{K_1 \times L - K_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{L - K_1 \times K_1} & \mathbf{0}_{L - K_1 \times L - K_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{K_2 \times K_2} & \mathbf{0}_{K_2 \times L - K_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{L - K_2 \times K_2} & \mathbf{0}_{L - K_2 \times L - K_2} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{K_J \times K_J} & \mathbf{0}_{K_J \times L - K_J} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}_{L - K_J \times K_J} & \mathbf{0}_{L - K_J \times L - K_J} \end{pmatrix} \quad (2.33)$$

A kapacitásra vonatkozó peremfeltételt (2.4) átalakítva a következő egyenletet kapjuk:

$$\sum_{l=1}^M \left( \left( \sum_{j=1}^J c_{j,l} \right) - V \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_B \mathbf{y} + \mathbf{b}_B^T \mathbf{y}, \quad (2.34)$$

ahol

$$L \geq M \geq \left\lceil \frac{\sum_{j=1}^J x_j}{V} \right\rceil. \quad (2.35)$$

A kapacitás teljes felhasználását feltételként csak addig az időpillanatig lehet előírni, amikor még rendelkezésre áll megfelelő mennyiségű ütemezendő feladat. Ezen időpillanat értékére vonatkozó korlátokat a (2.35) egyenlet írja le. (A tényleges időpillanat egy közelítő algoritmus használatával határozható meg.) A (2.34) egyenlet megoldása a következő:

$$\mathbf{b}_B = \left[ \mathbf{V}_{M \times 1}, \mathbf{0}_{(L-M) \times 1}, \mathbf{V}_{M \times 1}, \mathbf{0}_{(L-M) \times 1}, \dots, \mathbf{V}_{M \times 1}, \mathbf{0}_{(L-M) \times 1} \right], \quad (2.36)$$

$$\mathbf{W}_B = 2 \begin{pmatrix} \mathbf{D}_B & \mathbf{D}_B & \cdots & \mathbf{D}_B \\ \mathbf{D}_B & \mathbf{D}_B & \cdots & \mathbf{D}_B \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_B & \mathbf{D}_B & \cdots & \mathbf{D}_B \end{pmatrix}, \quad (2.37)$$

itt

$$\mathbf{D}_B = \begin{pmatrix} \mathbf{I}_{M \times M} & \mathbf{0}_{M \times (L-M)} \\ \mathbf{0}_{(L-M) \times M} & \mathbf{0}_{(L-M) \times (L-M)} \end{pmatrix}. \quad (2.38)$$

A TT probléma esetén a célfüggvény (2.25) a következő formára hozható, mivel a négyzetre emelés nem változtatja meg a kifejezés minimumértékét és bináris az ütemezési mátrix:

$$\sum_{j=1}^J \sum_{l=K_j+1}^L c_{j,l}^2. \quad (2.39)$$

Az egyenlet, amelyet a transzformációhoz meg kell oldani:

$$\sum_{j=1}^J \sum_{l=K_j+1}^L c_{j,l}^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_C \mathbf{y} + \mathbf{b}_C^T \mathbf{y} \quad (2.40)$$

Az egyenlet megoldása szerint a súlymátrix és vektor a következőképpen írható fel:

$$\mathbf{b}_C = \mathbf{0}_{JL \times 1}, \quad (2.41)$$

$$\mathbf{W}_C = 2 \begin{pmatrix} \mathbf{D}_{C_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{C_2} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_{C_J} \end{pmatrix}, \quad (2.42)$$

ahol

$$\mathbf{D}_{C_j} = \begin{pmatrix} \mathbf{0}_{K_j \times K_j} & \mathbf{0}_{K_j \times (L-K_j)} \\ \mathbf{0}_{(L-K_j) \times K_j} & \mathbf{I}_{(L-K_j) \times (L-K_j)} \end{pmatrix}. \quad (2.43)$$

A teljes kvadratikus alakhoz szükséges paraméterek a fentebb kifejtett mátrixok és vektorok összegei, azaz

$$\begin{aligned} \mathbf{b}_{TT} &= \alpha \mathbf{b}_A + \beta \mathbf{b}_B + \gamma \mathbf{b}_C, \\ \mathbf{W}_{TT} &= \alpha \mathbf{W}_A + \beta \mathbf{W}_B + \gamma \mathbf{W}_C. \end{aligned} \quad (2.44)$$

Az összegzésben bevezettem az  $\alpha, \beta, \gamma$  súlyozó heurisztikus paramétereket, amelyek az egyes peremfeltételek és célfüggvény fontosságát szabályozzák.

#### 2.4.2.2. A TWT probléma kvadratikus alakja

A TWT probléma egy egyszerű módosítása a TT problémának, amely során az összesített súlyozott képletet minimalizáljuk a (2.17) egyenlet szerint. A peremfeltételek az előző szakaszban ismertetettek szerint történik. A célfüggvényt a következőképpen módosítottuk:

$$\sum_{j=1}^J \sum_{k=K_j+1}^L w_j c_{j,l}^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}'_C \mathbf{y} + \mathbf{b}'_C{}^T \mathbf{y}. \quad (2.45)$$

Ennek megfelelően átalakítottuk a kapcsolódó mátrixot:

$$\mathbf{W}'_C = 2 \begin{pmatrix} \mathbf{D}'_{C_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}'_{C_2} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}'_{C_J} \end{pmatrix}, \quad (2.46)$$

amelyben az átalakított mátrix:

$$\mathbf{D}'_{C_j} = \begin{pmatrix} \mathbf{0}_{K_j \times K_j} & \mathbf{0}_{K_j \times (L-K_j)} \\ \mathbf{0}_{(L-K_j) \times K_j} & w_j \mathbf{I}_{(L-K_j) \times (L-K_j)} \end{pmatrix} \in \mathbb{R}^{L \times L}. \quad (2.47)$$

Így teljes megoldáshoz összegzünk a módosított mátrixot felhasználva:

$$\begin{aligned} \mathbf{b}_{TWT} &= \alpha \mathbf{b}_A + \beta \mathbf{b}_B + \gamma' \mathbf{b}_C, \\ \mathbf{W}_{TWT} &= \alpha \mathbf{W}_A + \beta \mathbf{W}_B + \gamma' \mathbf{W}'_C. \end{aligned} \quad (2.48)$$

A teljesíthetőség analízis során a TWT problémára nyújtott megoldáson keresztül mutatom be a módszer eredményeit.

### 2.4.2.3. Az egyenletes terhelést biztosító ütemezés kvadratikus alakja

Ebben a szakaszban az egyenletes terhelést garantáló ütemezési problémának megoldásával foglalkozom. Ennek a problémának egy közvetlen alkalmazási lehetősége az egyszerűsített WSN modell (2.3 szakasz) szerinti csomagütemezés megvalósítása. (Az alkalmazás fontossága és egy polinom idejű algoritmus, az LBS a 2.4.1.2 fejezetben bemutatásra került.) Ennél a problémánál az előírt feladatmennyiségre (2.3) és időkorlátokra (2.5) vonatkozó feltételek megegyeznek a TT problémánál bemutatott feltételekkel, így a  $\mathbf{W}_A$ ,  $\mathbf{W}_C$  mátrixok, valamint  $\mathbf{b}_A$  és  $\mathbf{b}_C$  vektorok már rendelkezésre állnak. Azonban szükséges az egyenletes terhelési feltételhez tartozó kvadratikus alak meghatározása. Az egyenletes terhelés megvalósítása érdekében egy olyan célfüggvényt írtam fel, amely garantálja, hogy két tetszőleges időszakban az ütemezett feladatok számának különbsége minimális legyen:

$$\forall l, k: \left( \sum_{j=1}^J c_{j,l} - \sum_{j=1}^J c_{j,k} \right)^2 = 0 \quad (2.49)$$

A feltételhez tartozó kvadratikus alak meghatározása a következő egyenlet megoldásával történik:

$$\sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} - \sum_{j=1}^J c_{j,k} \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_D \mathbf{y} + \mathbf{b}_D^T \mathbf{y} \quad (2.50)$$

A bal oldali kifejezést szétbontottam és a következő módon alakítottam:

$$\begin{aligned} \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} - \sum_{j=1}^J c_{j,k} \right)^2 &= \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 + \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,k} \right)^2 - 2 \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} \sum_{j=1}^J c_{j,k} \right) = \\ &= 2 \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 - 2 \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} \sum_{j=1}^J c_{j,k} \right) = 2L \sum_{l=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 - 2 \sum_{l=1}^L \sum_{k=1}^L \left( \sum_{j=1}^J c_{j,l} \sum_{j=1}^J c_{j,k} \right) = \\ &= 2L \sum_{l=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 - 2 \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \right) \left( \sum_{k=1}^L \sum_{j=1}^J c_{j,k} \right) = 2L \sum_{l=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 - 2 \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \right)^2. \end{aligned} \quad (2.51)$$

A végső forma az alábbi két egyenletet eredményezi:

$$2L \sum_{l=1}^L \left( \sum_{j=1}^J c_{j,l} \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_{D_1} \mathbf{y} + \mathbf{b}_{D_1}^T \mathbf{y}, \quad (2.52)$$

$$-2 \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_{D_2} \mathbf{y} + \mathbf{b}_{D_2}^T \mathbf{y}. \quad (2.53)$$

Az egyenletek megoldásai a következők:

$$\mathbf{b}_{D_1} = \mathbf{b}_{D_2} = \mathbf{0}_{JL \times 1}, \quad (2.54)$$

és

$$\mathbf{W}_{D_1} = 2L \begin{pmatrix} \mathbf{I}_{L \times L} & \mathbf{I}_{L \times L} & \cdots & \mathbf{I}_{L \times L} \\ \mathbf{I}_{L \times L} & \mathbf{I}_{L \times L} & \cdots & \mathbf{I}_{L \times L} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{L \times L} & \mathbf{I}_{L \times L} & \cdots & \mathbf{I}_{L \times L} \end{pmatrix}, \quad (2.55)$$

$$\mathbf{W}_{D_2} = 2\mathbf{1}_{JL \times JL}. \quad (2.56)$$

Ennél a problémánál a végső súlymátrix és súlyvektor, a fentebb részletezett peremfeltételek és célfüggvény összege a súlyozó paraméterekkel együtt:

$$\begin{aligned} \mathbf{b}_{BC} &= \alpha \mathbf{b}_A + \gamma \mathbf{b}_C + \delta_1 \mathbf{b}_{D_1} + \delta_2 \mathbf{b}_{D_2}, \\ \mathbf{W}_{BC} &= \alpha \mathbf{W}_A + \gamma \mathbf{W}_C + \delta_1 \mathbf{W}_{D_1} + \delta_2 \mathbf{W}_{D_2}. \end{aligned} \quad (2.57)$$

A kiszámított mátrix segítségével lehetséges a probléma megoldása kvadratikus optimalizálással.

#### 2.4.2.4. A WSN csomagütemezés probléma kvadratikus alakja

Ebben a szakaszban egy teljes WSN hálózat csomagütemezését mutatom be. A korábbi transzformációval szemben itt a kvadratikus alakok paramétereinek meghatározásához az ütemezési mátrixot nem sorfolytonosan kódolom (ahogyan azt a (2.24) formulában bemutattam), hanem oszlopfolytonosan a következők szerint:

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,L} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ c_{J,1} & c_{J,2} & \cdots & c_{J,L} \end{pmatrix} \rightarrow \mathbf{y} = (c_{1,1}, c_{2,1}, \dots, c_{J,1}, c_{1,2}, \dots, c_{J,2}, c_{1,L}, \dots, c_{J,L})^T. \quad (2.58)$$

A következő eredmények és számítások ennek megfelelően kerülnek bemutatásra. A 2.3-es szakaszban leírt feltételek közül az első a (2.13) feltétel, amely az egyes node-ok által kezdeményezett csomagküldések számát garantálja. Az elküldendő csomagok a kezdeti csomagszám és az áthaladó forgalom összessége a (2.14) egyenlet szerint. Az ennek megfelelő kvadratikus alak közvetlen módosítása a feladatok méretére vonatkozó kvadratikus alaknak:

$$\sum_{j=1}^J \left( \sum_{l=1}^L c_{j,l} - z_j \right)^2. \quad (2.59)$$

Ehhez az egyenlethez tartozó transzformáció:

$$\sum_{j=1}^J \left( \sum_{l=1}^L c_{j,l} - z_j \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}'_A \mathbf{y} + \mathbf{b}'_A{}^T \mathbf{y}, \quad (2.60)$$

és megoldása:

$$\mathbf{W}'_A = 2 \begin{pmatrix} \mathbf{I}_{J \times J} & \mathbf{I}_{J \times J} & \cdots & \mathbf{I}_{J \times J} \\ \mathbf{I}_{J \times J} & \mathbf{I}_{J \times J} & \cdots & \mathbf{I}_{J \times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{I}_{J \times J} & \mathbf{I}_{J \times J} & \cdots & \mathbf{I}_{J \times J} \end{pmatrix}, \quad (2.61)$$

$$\mathbf{b}'_A = (\mathbf{z}, \mathbf{z}, \dots, \mathbf{z})_{JL \times 1}. \quad (2.62)$$

Egy ehhez kapcsolódó peremfeltétel szerint minden egyes node esetén garantálni kell a kimenő és bemenő csomagok számának egyezőségét, a levél node-ok és bázisállomás kivételével:

$$\sum_{i=1}^J \left( \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \mathbf{R}_{j,i} \right) - \left( \sum_{l=1}^L c_{i,l} \right) \right)^2 = 0. \quad (2.63)$$

A megoldandó egyenlet ennek megfelelően:

$$\sum_{i=1}^J \left( \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \mathbf{R}_{j,i} \right) - \left( \sum_{l=1}^L c_{i,l} \right) \right)^2 = \frac{1}{2} \mathbf{y}^T \mathbf{W}_E \mathbf{y} + \mathbf{b}_E^T \mathbf{y}. \quad (2.64)$$

A kifejezés bal oldalát a következő módon bontottam szét:

$$\begin{aligned} & \sum_{i=1}^J \left( \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \mathbf{R}_{j,i} \right) - \left( \sum_{l=1}^L c_{i,l} \right) \right)^2 = \\ & = \sum_{i=1}^J \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \mathbf{R}_{j,i} \right)^2 + \sum_{i=1}^J \left( \sum_{l=1}^L c_{i,l} \right)^2 - 2 \sum_{i=1}^J \left( \sum_{l=1}^L \sum_{j=1}^J c_{j,l} \mathbf{R}_{j,i} \right) \left( \sum_{l=1}^L c_{i,l} \right). \end{aligned} \quad (2.65)$$

A középső tag megoldása a (2.52) megoldásának módosítása, ez megjelenik ennek az egyenletnek a megoldásában is. A teljes súlymátrix és súlyvektor ehhez a peremfeltételhez:

$$\mathbf{b}_E = \mathbf{0}_{JL \times 1}, \quad (2.66)$$

$$\mathbf{W}_E = 2 \begin{pmatrix} \mathbf{P}_{J \times J} & \mathbf{P}_{J \times J} & \cdots & \mathbf{P}_{J \times J} \\ \mathbf{P}_{J \times J} & \mathbf{P}_{J \times J} & \cdots & \mathbf{P}_{J \times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{J \times J} & \mathbf{P}_{J \times J} & \cdots & \mathbf{P}_{J \times J} \end{pmatrix}, \quad (2.67)$$

ahol

$$\mathbf{P} = \mathbf{D}_E + \mathbf{I}_{J \times J} - 2\mathbf{R}_{J \times J}, \quad (2.68)$$

$$\mathbf{D}_E = \begin{pmatrix} \mathbf{R}_{1,1}^2 + \mathbf{R}_{1,2}^2 + \cdots + \mathbf{R}_{1,N}^2 & \mathbf{R}_{1,1}\mathbf{R}_{2,1} + \cdots + \mathbf{R}_{1,N}\mathbf{R}_{2,N} & \cdots & \mathbf{R}_{1,1}\mathbf{R}_{N,1} + \cdots + \mathbf{R}_{1,N}\mathbf{R}_{N,N} \\ \mathbf{R}_{1,1}\mathbf{R}_{2,1} + \mathbf{R}_{1,2}\mathbf{R}_{2,2} + \cdots + \mathbf{R}_{1,N}\mathbf{R}_{2,N} & \mathbf{R}_{2,1}^2 + \mathbf{R}_{2,2}^2 + \cdots + \mathbf{R}_{2,N}^2 & \cdots & \mathbf{R}_{2,1}\mathbf{R}_{N,1} + \cdots + \mathbf{R}_{2,N}\mathbf{R}_{N,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{1,1}\mathbf{R}_{N,1} + \mathbf{R}_{1,2}\mathbf{R}_{N,2} + \cdots + \mathbf{R}_{1,N}\mathbf{R}_{N,N} & \mathbf{R}_{2,1}\mathbf{R}_{N,1} + \mathbf{R}_{2,2}\mathbf{R}_{N,2} + \cdots + \mathbf{R}_{2,N}\mathbf{R}_{N,N} & \cdots & \mathbf{R}_{N,1}^2 + \mathbf{R}_{N,2}^2 + \cdots + \mathbf{R}_{N,N}^2 \end{pmatrix} \quad (2.69)$$

A következő peremfeltétel (2.16) az interferencia mennyiségét minimalizálja. A feltételhez tartozó transzformációs egyenlet:

$$\sum_{k=1}^L c_{*,k} \mathbf{F} c_{*,k}^T = \frac{1}{2} \mathbf{y}^T \mathbf{W}_F \mathbf{y} + \mathbf{b}_F^T \mathbf{y}. \quad (2.70)$$

Ennek peremfeltétel mátrix alakját felhasználva a megoldás a következőképpen alakul:

$$\mathbf{W}_F = 2 \begin{pmatrix} \mathbf{F}_{J \times J} & \mathbf{0}_{J \times J} & \cdots & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathbf{F}_{J \times J} & \cdots & \mathbf{0}_{J \times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{J \times J} & \mathbf{0}_{J \times J} & \cdots & \mathbf{F}_{J \times J} \end{pmatrix}, \quad (2.71)$$

$$\mathbf{b}_F = \mathbf{0}_{JL \times 1}, \quad (2.72)$$

ahol az  $\mathbf{F}$  jelöli az interferencia mátrixot.

A WSN hálózatban az energiahatékonyság garantálása a célfüggvényünk, továbbá szeretnénk elkerülni azt, hogy a node-ok akkor forgalmazzanak csomagot, amikor a pufferük üres. Ennek érdekében minimalizálni szeretnénk a forgalmi állapotnak megfelelően azokat a beütemezett

küldéseket, amik nem felelnek meg a kívánalmaknak. A következő képlet tartalmazza a probléma és a transzformáció megfogalmazását:

$$\forall k = 1 \dots L: \sum_{i=1}^J \left( \sum_{l=1}^k \left( \sum_{j=1}^J c_{j,l} R_{j,i} \right) - \sum_{l=1}^k c_{i,l} \right)^2 \rightarrow \frac{1}{2} \mathbf{y}^T \mathbf{W}_G \mathbf{y} + \mathbf{b}_G^T \mathbf{y}. \quad (2.73)$$

A transzformáció megoldása:

$$\mathbf{b}_G = (\mathbf{x}-1, \mathbf{x}-1, \dots, \mathbf{x}-1), \quad (2.74)$$

$$\mathbf{W}_G = 2(\mathbf{D}_G \cdot \mathbf{D}_G^T), \quad (2.75)$$

ahol

$$\mathbf{D}_G = \begin{pmatrix} \mathbf{I}_{J \times J} - \mathbf{R} & \mathbf{0}_{J \times J} & \cdots & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} - \mathbf{R} & \cdots & \mathbf{0}_{J \times J} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{J \times J} & \mathbf{0}_{J \times J} & \cdots & \mathbf{I}_{J \times J} - \mathbf{R} \end{pmatrix}. \quad (2.76)$$

A célfüggvény pedig egy node esetén a csomagküldés csoportosítását írja elő. Emiatt a rádióinicializálások és RX/TX módok váltásának száma minimális lesz. Az ehhez tartozó egyenlet és annak általunk meghatározott megoldása:

$$\sum_{j=1}^J \sum_{l=1}^{L-1} (c_{j,l} + c_{j,l+1} - 2c_{j,k} c_{j,l+1}) = \frac{1}{2} \mathbf{y}^T \mathbf{W}_H \mathbf{y} + \mathbf{b}_H^T \mathbf{y}. \quad (2.77)$$

$$\mathbf{b}_H = (-0.5, -1, -1, \dots, -1, -0.5)_{(JL \times 1)} \quad (2.78)$$

$$\mathbf{W}_H = - \begin{pmatrix} \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} & \cdots & \cdots & \mathbf{0}_{J \times J} \\ \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} & \cdots & \mathbf{0}_{J \times J} \\ \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \cdots & \mathbf{0}_{J \times J} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_{J \times J} & \cdots & \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} \\ \mathbf{0}_{J \times J} & \cdots & \cdots & \mathbf{0}_{J \times J} & \mathbf{I}_{J \times J} & \mathbf{0}_{J \times J} \end{pmatrix}. \quad (2.79)$$

A kiterjesztett WSN modellnek megfelelő ütemezés, a peremfeltételekkel és célfüggvénnyel együtt a következő paramétereket adja a kvadratikus optimalizálási feladathoz:

$$\begin{aligned} \mathbf{b}_{WSN} &= \alpha' \mathbf{b}'_A + \varepsilon \mathbf{b}_E + \xi \mathbf{b}_F + \zeta \mathbf{b}_G + \mathbf{b}_H \\ \mathbf{W}_{WSN} &= \alpha' \mathbf{W}'_A + \varepsilon \mathbf{W}_E + \xi \mathbf{W}_F + \zeta \mathbf{W}_G + \mathbf{W}_H \end{aligned} \quad (2.80)$$

### 2.4.3. A kvadratikus optimalizálási feladatok megoldása Hopfield neurális hálózattal

Az előzőekben megmutattam a különböző erőforrás ütemezési problémák kvadratikus optimalizálási feladatokká történő transzformációját. Ehhez meghatároztam a (2.22) egyenlet szerinti alaknak megfelelő paramétereket. Ebben a szakaszban a kvadratikus optimalizálási feladatot egy gyakorta alkalmazott heurisztikus módszer segítségével, a Hopfield típusú mesterséges neurális hálózatokkal (HNN) oldom meg. A HNN használatát indokolja, hogy polinom időben lehetséges a megoldás



meghatározása és, hogy meghatározott körülmények esetén bizonyítottan optimális eredményt biztosít.

Egy  $N$  neuront tartalmazó HNN működése az alábbi állapot változási egyenlettel írható le:

$$y_i(k+1) = \operatorname{sgn}\left(\sum_{j=1}^N \hat{W}_{i,j} y_j(k) - \hat{b}_i\right), \quad i = \operatorname{mod}_N k, \quad (2.81)$$

ahol a HNN rekurzió stabilitásának megőrzése érdekében az alábbi módosítások szükségesek:

$$\begin{aligned} \mathbf{d} &= \operatorname{diag}(\mathbf{W}), \\ \hat{\mathbf{W}} &= \mathbf{W} - \operatorname{diag}(\mathbf{d}), \\ \hat{\mathbf{b}} &= \mathbf{b} + \frac{1}{2}\mathbf{d}. \end{aligned} \quad (2.82)$$

A módosítást az indokolja, hogy a HNN rekurzió csakis akkor fog konvergálni saját fix pontjába, ha a  $\mathbf{W}$  mátrix szimmetrikus és nincsen az egyes neuronoknak önvisszacsatolása. Hopfield [73] bizonyította, hogy a fenti rekurzió, betartva a feltételeket a hálózat konvergál kvadratikus Lyapunov függvényének a minimumába, amely

$$\mathcal{L}(\mathbf{y}) := \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \hat{W}_{i,j} y_i y_j + \sum_{i=1}^N y_i \hat{b}_i = \frac{1}{2} \mathbf{y}^T \hat{\mathbf{W}} \mathbf{y} + \hat{\mathbf{b}}^T \mathbf{y}. \quad (2.83)$$

Ez lehetővé teszi, hogy a HNN-t alkalmazzam a kvadratikus problémák megoldására, mivel azok mindegyike minimalizálási feladat és megfelelnek az előző kritériumoknak. Számos hasonló kvadratikus problémára megmutatták, hogy az HNN-el megoldható [42], [43], [44].

A kvadratikus problémák HNN-el létrehozott megoldása gyakran nem felel meg a feladatokhoz tartozó a peremfeltételeknek amiatt, hogy a HNN rekurzió bennragad lokális (ál-globális) minimum értékekben, a konvergencia feltételek betartása ellenére, aminek az oka az, hogy diszkrét tér felett történik az optimalizáció. (Ekkor több lokális minimum hely is létezhet.) Ennek egyik alapvető elkerülési módja az, hogy több véletlen sorsolt kezdőállapotból indítjuk el a HNN rekurziót, és a legjobb metrikával rendelkező megoldást tekintjük a módszer végeredményének. A továbbiakban olyan módszereket mutatok be, amelyekkel a hagyományos HNN rekurzió megoldását felül lehet múlni.

Az általam bemutatott kvadratikus optimalizálási feladatokhoz tartozó súlymátrix és súlyvektor mindegyike több komponensből áll össze, amely komponensek fontosságát heurisztikus paraméterek szabályozzák. A legjobb paraméter konstelláció meghatározása egy külön probléma, amely túlmutat az eredeti optimalizálás feladatán. Ehelyett egy olyan heurisztikus módszert alkalmaztam, amely a legkritikusabb peremfeltételt priorizálja, és a többi feltétel fontosságát növeli a hibák számának csökkentése érdekében. Erre az 2.14. ábra által bemutatott algoritmust alkalmaztam, amely a TWT probléma esetén mutatja be a paraméterek korrekcióját. A módszer során használt korrekciós algoritmus egy változatát a 2.7. ábra mutatja be.

---

**REQUIRE**  $\mathbf{x}, \mathbf{K}, V, e$ 

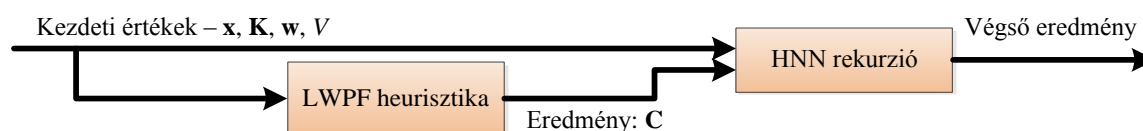

---

 $\alpha \leftarrow 0.1, \beta \leftarrow 5, \gamma \leftarrow 5$  $i \leftarrow 0$ **REPEAT** $i \leftarrow i + 1$  $\mathbf{C}_i \leftarrow \text{HNN}(\mathbf{x}, \mathbf{K}, V, \alpha, \beta, \gamma)$  $\alpha \leftarrow \alpha + 0.01$ **UNTIL**  $\text{error}(\mathbf{C}_i) \leq e$ **FOR**  $k = 1 \rightarrow i$  $\mathbf{C}_k \leftarrow \text{correct}(\mathbf{C}_k)$  $\mathbf{T}_k \leftarrow \text{calculateTWT}(\mathbf{C}_k)$ **END FOR****RETURN**  $\min(\mathbf{T})$ 


---

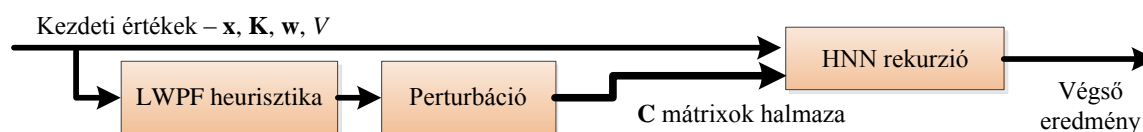
2.14. ábra A kvadrátikus optimalizáláshoz tartozó paraméterek komponenseinek súlyozására használt heurisztikus algoritmus

Előzőleg megmutattam, hogy a diszkrét térben Hopfield hálózat segítségével történő optimalizálás esetén miként lehet növelni a megoldás minőségét, elkerülve bizonyos lokális minimum pontokat. A heurisztikus paraméterek irányított megválasztásán túl létrehoztam egy módszert, amellyel tovább lehet csökkenteni annak esélyét, hogy a rekurzió ne az optimális megoldást találja meg. Ebben az esetben a HNN rekurziót irányítottan indítom el, olyan iniciális pontból, amely egy közelítő módszer megoldása, és amely ütemezés részben optimális. A módszer a *Smart HNN (SHNN)*, amelyet a 2.15. ábra mutat be. Az SHNN módszer alkalmazásával nem szükséges több véletlen sorsolt kezdőpontból indítani a HNN rekurziót, amely jelentősen lerövidíti a módszer futtatási idejét.



2.15. ábra SHNN módszer

Ugyanakkor az így meghatározott megoldás minősége nem mindig jobb a hagyományos HNN módszerhez viszonyítottan. Ezt elkerülni a PSHNN módszerrel lehet, amely a HNN rekurzióhoz hasonlóan több pontból indítja el a rekurziót, azonban az iniciális pontokat nem véletlen módon határozom meg, hanem egy közelítő megoldás eredményét felhasználva generálok, úgy hogy a közelítő heurisztika megoldását véletlenszerűen megváltoztatom a PLWPF esetén is alkalmazott módszerrel. A technika blokkdiagramját a 2.16. ábra mutatja be.



2.16. ábra PSHNN módszer

#### 2.4.4. HNN alapú megoldó párhuzamos futtathatósága

A bemutatott HNN alapú megoldó módszerek esetén több iniciális pontból azonos súlymátrix és vektor mellett történik a HNN rekurzió lefuttatása. Mindhárom megoldó esetén heurisztikus paraméterek számos kombinációjára megismételjük a HNN iteráció lefuttatását. Amennyiben a HNN iteráció egyetlen futtatásának idejét  $t$  jelöli, akkor a teljes futtatási idő:

$$T = R_{init} (R_{heuristics} t) \quad (2.84)$$

Ekkor  $R_{init} = 500$ ;  $R_{heuristics} \sim 10^2$ ;  $t = 10^{-3} s$  esetén a futási idő  $T \sim 50$  másodperc, egy  $200 \times 200$  méretű  $W$  mátrix esetén egy asztali PC (Intel Core i5) esetében, egyetlen processzormag felhasználásával.

Mivel az egyes ismétlődő iterációk egymástól függetlenek, ezért kézenfekvő ezeknek az iterációknak a párhuzamos lefuttatása, amellyel elérhető az, hogy egyetlen HNN rekurzió konvergencia ideje alatt számos iniciális pontból indított rekurzió lefusszon, amelyekből a legjobb megoldást tekinthető a végeredménynek. Egy az irodalomban már leírt létező GPGPU alapú HNN implementációt [45] figyelembe véve a következő párhuzamosítási lehetőséget terveztem meg, amely módszer az nVidia Fermi architektúrán alkalmazható [46]:

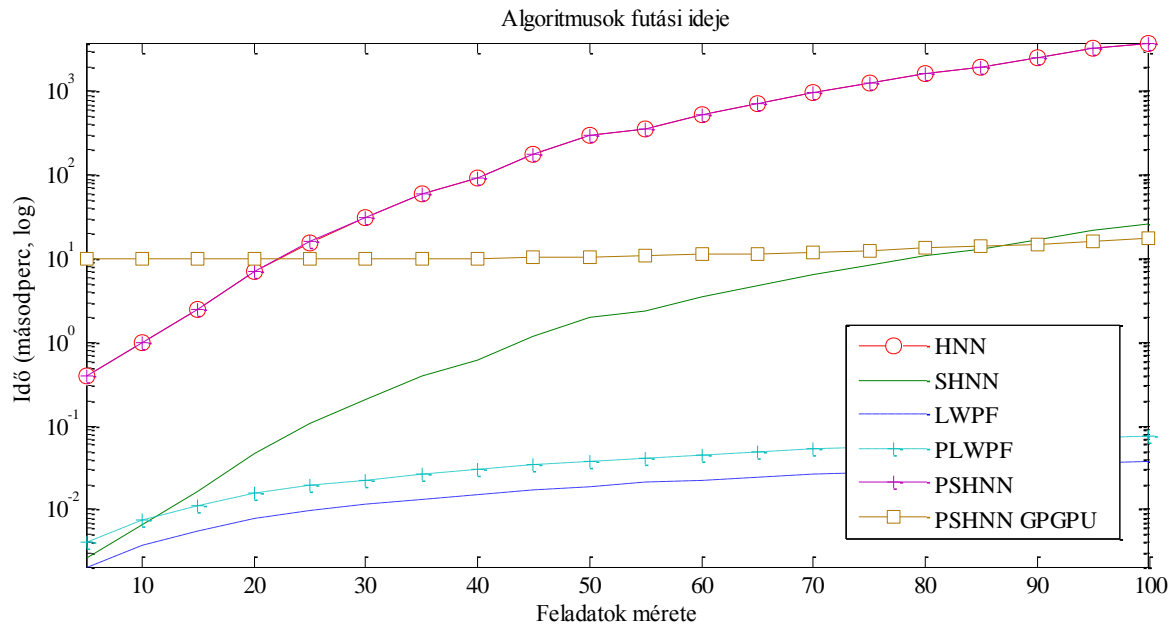
- A  $W$  mátrix ritkás (5-10%-a nem nulla érték), mely lehetővé teszi nagyméretű  $W$  mátrixszal rendelkező HNN-ek esetén is, hogy a HNN iterációkat egy-egy stream processzor dolgozza fel, mivel a mátrix az osztott memóriában elfér.
- Több párhuzamosan feldolgozott iniciális állapotú, vagy különböző heurisztikus paraméterekkel bíró HNN-t szimultán különböző stream processzorokon futtatnak.

Ennek megfelelően a Fermi sokprocesszoros architektúrán a HNN/PSHNN algoritmus futási ideje jelentősen lerövidíthető, és így megfelelő méretű  $W$  mátrix esetén több nagyságrenddel jobb idő alatt kapunk eredményt. Ez lehetővé teszi, hogy a hagyományos heurisztikáknál nemcsak jobb eredményhez jutunk, hanem azt az eredményt a gyakorlatban is elfogadható időben határozzuk meg [47].

A futási idők összehasonlítását az 2.17. ábra mutatja be, ahol a HNN rekurziók egyes javításai mellett az LWPF és PLWPF közelítő megoldás futási ideje is látható, a TWT probléma esetén. A futási idők összehasonlításnál a referencia architektúrák a következők:

- Párhuzamos architektúra: nVidia Fermi architektúra, nVidia 440GTX chippel 1 Gb memóriával.
- A hagyományos architektúra: Intel Core i5@2.8 Ghz, 4 Gb memóriával.

Az ábra szerint a párhuzamos implementáció végrehajtási ideje kis feladatméretek esetén azonos, amely egy küszöböt meghaladva növekedik. Ennek az oka, hogy kisebb feladatméretek esetén a rendelkezésre álló CUDA magok nem mindegyike kap feladatot, ugyanakkor a teljes futási idő ettől nem függ. A párhuzamos futtatás architekturális korlátainak átlépésekor több ütemben kerül lefuttatásra az algoritmus. Ez a futtatási időt a rekurziók számában megmutató függvényét lépcsőzetessé teszi.



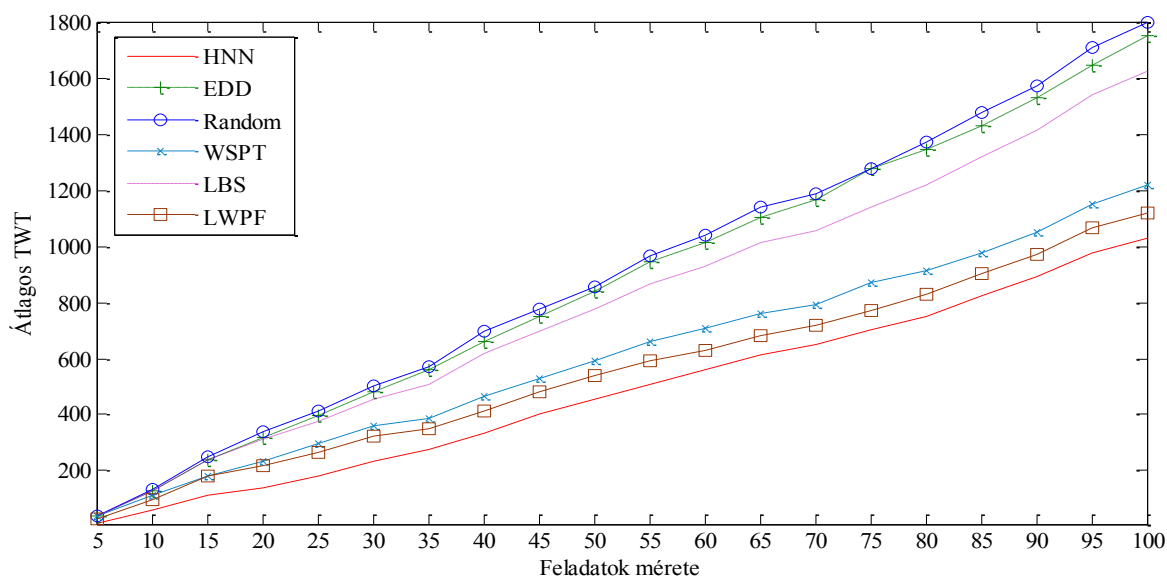
2.17. ábra Az egyes HNN alapú implementáció futási idejének összehasonlítása a közelítő megoldások futási idejével és a GPGPU-s implementációval (A HNN és PSHNN módszer futási ideje közel azonos)

## 2.5. A kvadratikus optimalizáció eredményeinek bemutatása

Az előzőleg bemutatott problémák és az arra megadott kvadratikus optimalizáláson alapuló megoldások teljesítőképességét vizsgálom ebben a szakaszban. Az egyes problémák esetén külön-külön megmutatom a létező módszerekhez viszonyított eredményt.

### 2.5.1. Teljesítőképesség összehasonlítás a TWT probléma esetén

A TWT probléma megoldására a 2.4.1.1. szakaszban bemutatott közelítő heurisztikát, amelynek eredményét fel lehet használni a SHNN és PSHNN módszereknél. A tesztek során az LWPF módszert bemutató szakaszban leírt körülményeket alkalmazom: véletlen módon sorsolt feladat méreteket és határidőket állítok elő különböző nagyságrendekben. A HNN alapú megoldási módszerek eredményeit az LWPF/PLWPF módszerek, valamint a klasszikus heurisztikus algoritmusok eredményeivel vetem össze. A 2.18. ábra a különböző feladatméretek mellett meghatározott ütemezésekhez tartozó TWT értékeket mutatja be. A TWT átlagolt, továbbá az azonos feladatméret mellett különböző sorsolt feladatok megoldásához tartozó értékeket szintén átlagoltam. A HNN alapú módszer esetén a többször lefuttatott HNN rekurziók által szolgáltatott ütemezési mátrixok közül a legjobbát vettem figyelembe.

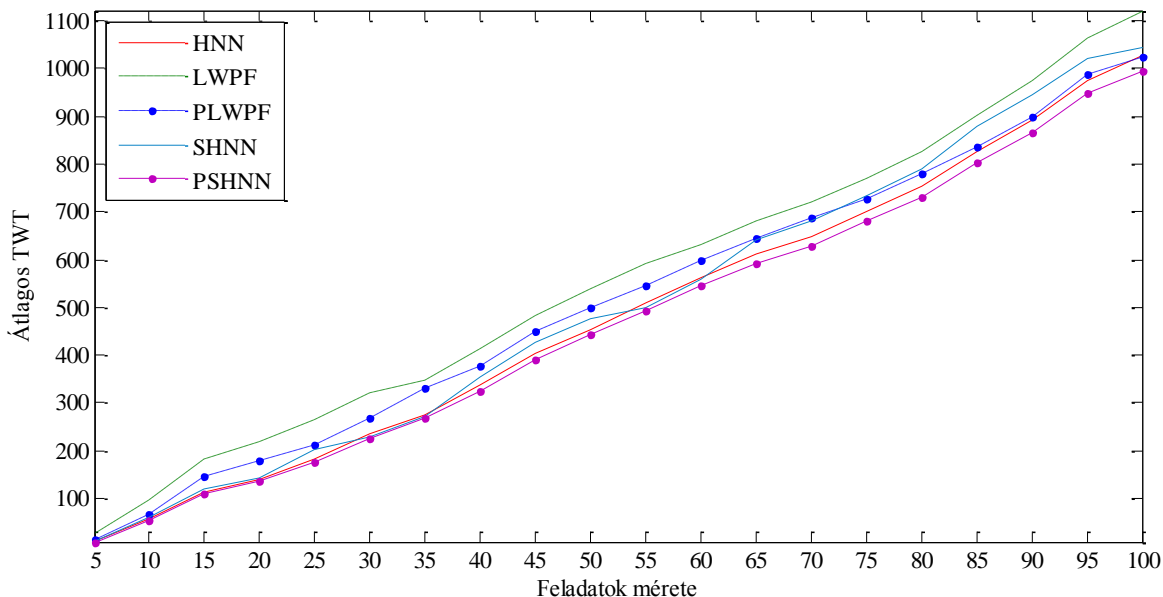


2.18. ábra Közelítő heurisztikák és a HNN módszer megoldásának TWT-je

Megállapítható, hogy a véletlen módon meghatározott ütemezési mátrixnál bármilyen módszer jobb eredményt szolgáltat. Az EDD és LBS módszer nem veszi figyelembe a feladatok súlyozását, ezek a következő módszerek a sorrendben. A legjobb eredményt a HNN alapú megoldás nyújtja, azonban emellett ez a leglassúbb módszer.

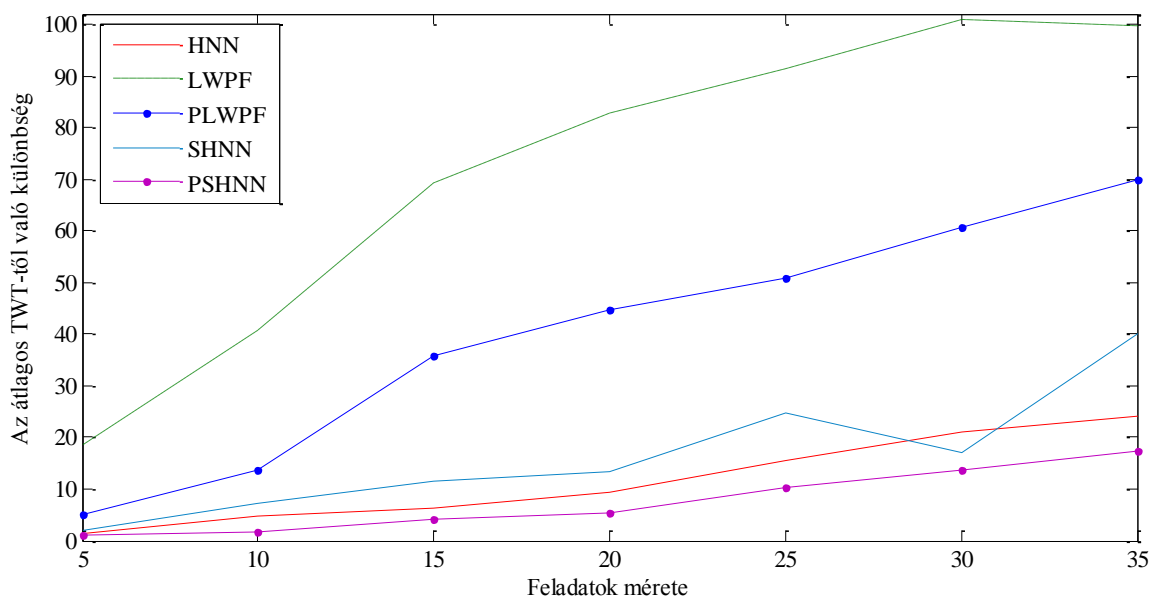
A szimulációk eredményét átlagolva és az egyes módszerek arányait vizsgálva megállapítható, hogy a HNN alapú módszer a vizsgált esetekben legalább 9%-kal jobb eredményt biztosít, mint a következő legjobb módszer. Ugyanakkor a feladat méretének növekedésével a HNN alapú módszer előnye csökken.

A TWT probléma esetén a javított módszerek közül a PSHNN az, ami a legjobb eredményt nyújtja, és ezeket az eredményeket a 2.19. ábra mutatja be. A PLWPF módszer a feladatméret növekedésével egyre jobban megközelíti a PSHNN módszer megoldását. A SHNN módszer átlagosan nem biztosít olyan jó eredményt, mint a HNN módszer.



2.19. ábra A javított módszerek eredményeinek összehasonlítása

A 2.20. ábra a javított módszerek optimális megoldáshoz viszonyított különbségét illusztrálja. Az optimális megoldást kimerítő kereséssel határoztam meg, ez azonban valós felhasználási körülmények között nem alkalmazható. A tesztek során kiderült, hogy nem heurisztikus megoldások közel optimális megoldást biztosítanak, ugyanakkor a feladatméret növekedésével az optimális megoldáshoz viszonyítva egyre rosszabb eredményt nyújtanak.



2.20. ábra Javított módszerek eredményei az optimális megoldáshoz viszonyítva

A tesztek eredményei azt is megmutatták, hogy több azonos méretű feladat esetén az egyes megoldási módszerek eredménye milyen módon viszonyul egymáshoz. Vizsgáltam azt is, hogy a PSHNN módszer azonos problémaméret esetén hány esetben nyújt jobb megoldást, mint a HNN módszer. Az egyes módszerek futási idejét a 2.4.4 szakaszban leírtam, és ezeket az eredményeket a 2.17. ábra illusztrálja, ahol látható, hogy a HNN módszer párhuzamos implementációja jelentős előnyökkel jár.

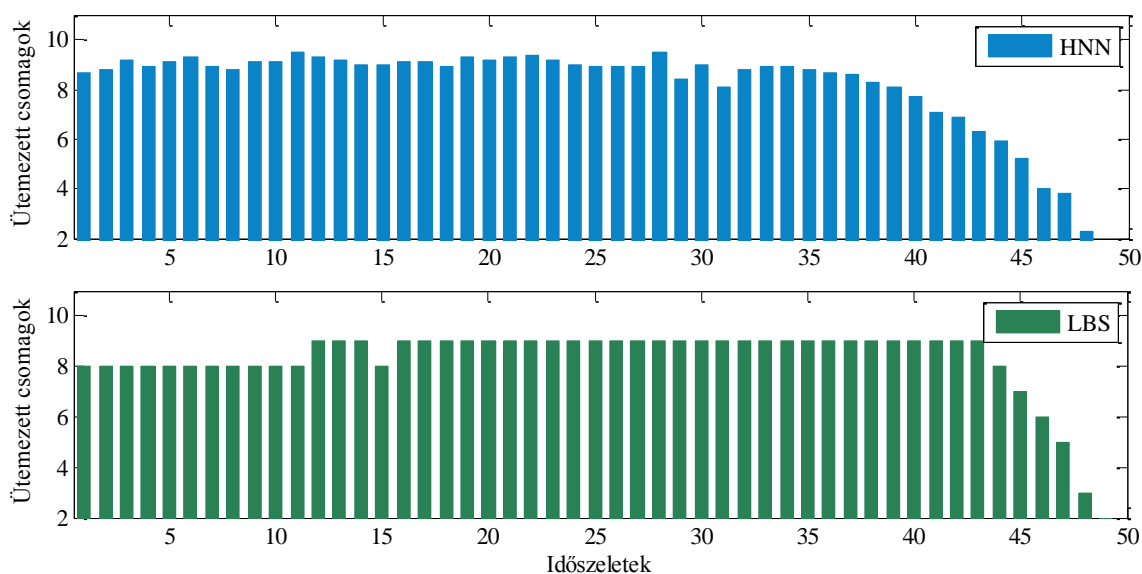
2.1. táblázat A HNN-LWPF és a PSHNN-HNN módszerek eredményeinek páronkénti aránya

Feladat mérete	5	10	15	20	30	40	50	75	100
HNN/ LWPF	99,9%	100%	99,8%	99,5%	99,4%	99,3%	99,3%	98,6%	98,8%
PSHNN/ HNN	100%	100%	99,9%	99,8%	99,8%	99,9%	99,7%	99,7%	99,5%

### 2.5.2. Teljesítőképesség bemutatása az egyszerűsített WSN feladat esetén

Az egyszerűsített WSN modell esetén az LBS algoritmus eredménye optimális, ahogyan azt a 2.4.2.3. szakaszban megmutattam. Itt azt vizsgálom, hogy a HNN alapú megoldás milyen teljesítményt képes elérni az optimális LBS módszerhez viszonyítva.

A módszerek összehasonlítását az LBS algoritmusnál leírtak szerint tettem meg. A módszer mérőszámait a (2.21) egyenletben leírt entrópia érték segítségével jellemzem. Az ütemezések karakterisztikáját a 2.21. ábra mutatja meg, az entrópia értékeket pedig a 2.2. táblázat. A HNN módszer láthatólag nem ad olyan pontos eredményt a feladatra, ugyanakkor érdemes megjegyezni, hogy a HNN alapú megoldások esetén tetszőleges célfüggvénnyel lehet bővíteni a feladatot, ami az LBS algoritmus esetén nem feltétlenül valósítható meg. Az entrópia értékek a korábban közölt eredményekkel összhangban állnak.



2.21. ábra A HNN és LBS módszer összevetése

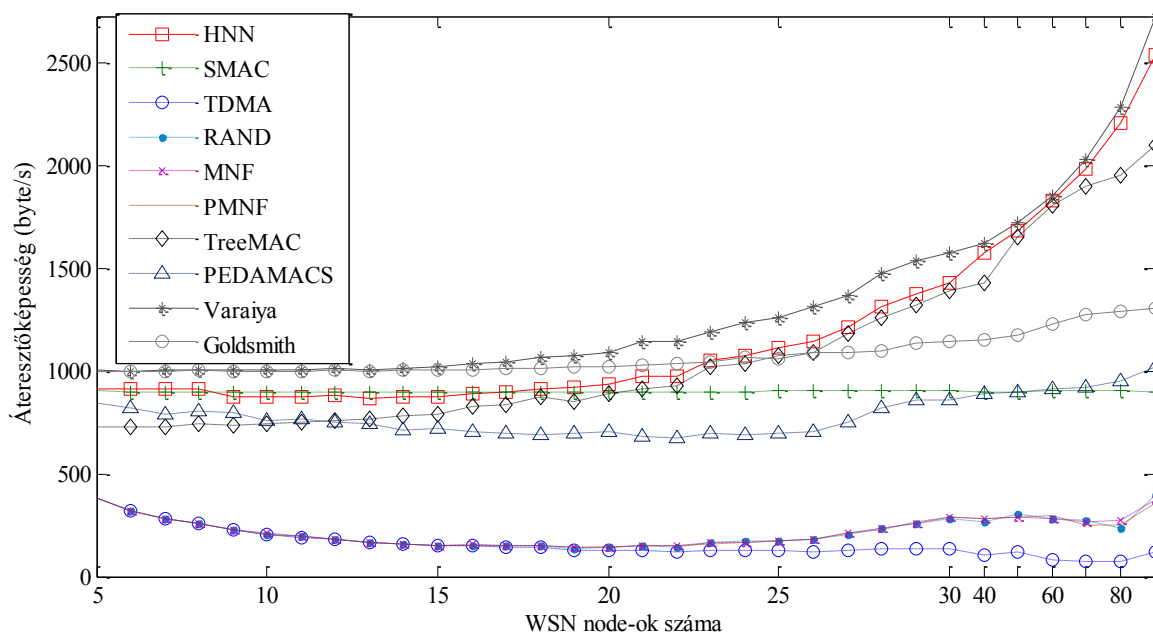
2.2. táblázat Az egyes módszerekhez ütemezéseihez tartozó entrópia érték

	Véletlen módszer eredménye	HNN eredménye	LBS eredménye	Optimális megoldás
<b>Entrópia</b>	5,45	5,78	5,89	5,89

Az eredmények ismeretében elmondható, hogy az LBS módszer gyors és pontos eredményt nyújt egyetlen specifikus probléma esetén, míg a HNN alapú megoldás is képes jó eredmény elérésére, azonban, ahogyan az bemutatásra kerül, más célfüggvénnyel bíró optimalizálási problémák esetén is bevethető.

### 2.5.3. Teljesítőképesség összehasonlítása a teljes WSN csomagütemezés esetén

A WSN csomagütemezés megvalósítása esetén a célfüggvény segítségével olyan ütemezést igyekeztünk megvalósítani, amely magas áteresztőképesség mellett energiahatékony működést tesz lehetővé azáltal, hogy az egyes node-ok egymást követően több csomagot küldjenek, illetve fogadjanak. Ezzel valósítható meg, hogy a node-ok minimális mennyiségű energiát használjanak fel rádióinicializálásra és az RX/TX (fogadási és küldési) módok közötti váltásra. Ehhez a célhoz alkottam meg a kommunikációs feltételek betartásához szükséges peremfeltételeket. Az ütemezési mátrixok meghatározását követően ellenőriztem a peremfeltételek teljesülését, és szükség esetén korrigáltam az ütemezési mátrixokat. Ezt követően a tesztek során az áteresztőképességet és energiahatékonyt vizsgáltam, amely a következőkben ismertetett eredmények nyújtotta.



2.22. ábra Az egyes csomagütemezők áteresztőképessége homogén csomagterheltség mellett

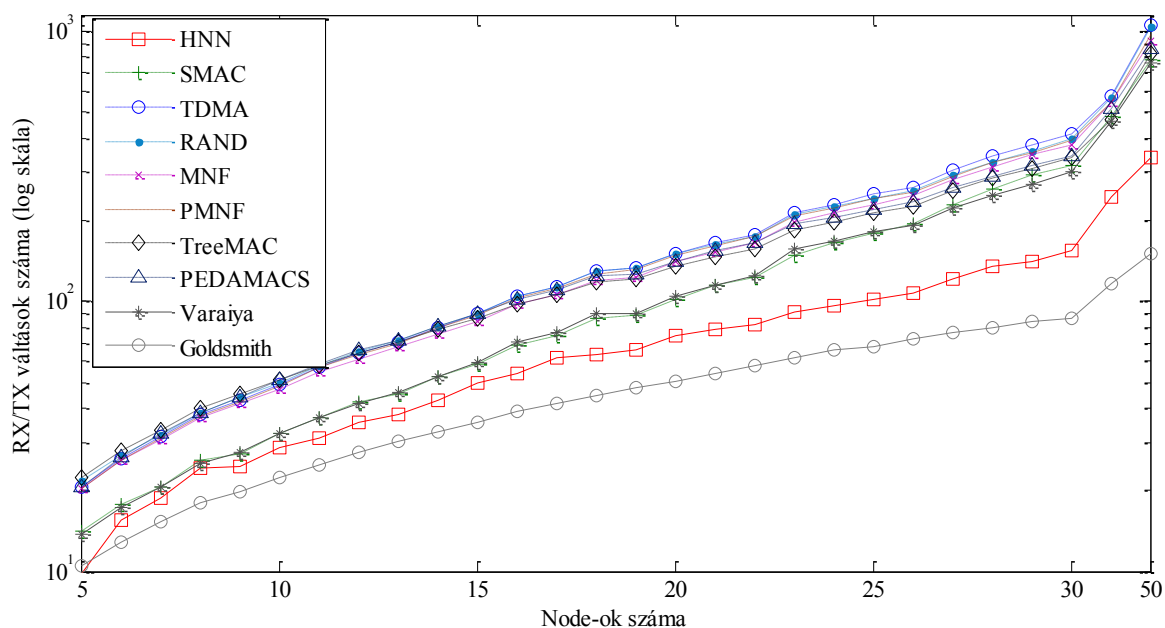
A 2.22. ábra bemutatja, hogy az egyes WSN méretekhez tartozóan milyen áteresztőképesség érhető el az egyes csomagütemezők során, amely esetben azt vizsgáltam, hogy az időegységenként mennyi információt lehet a BS-re juttatni. A tesztek folyamán homogén csomaggenerálódást alkalmaztam,



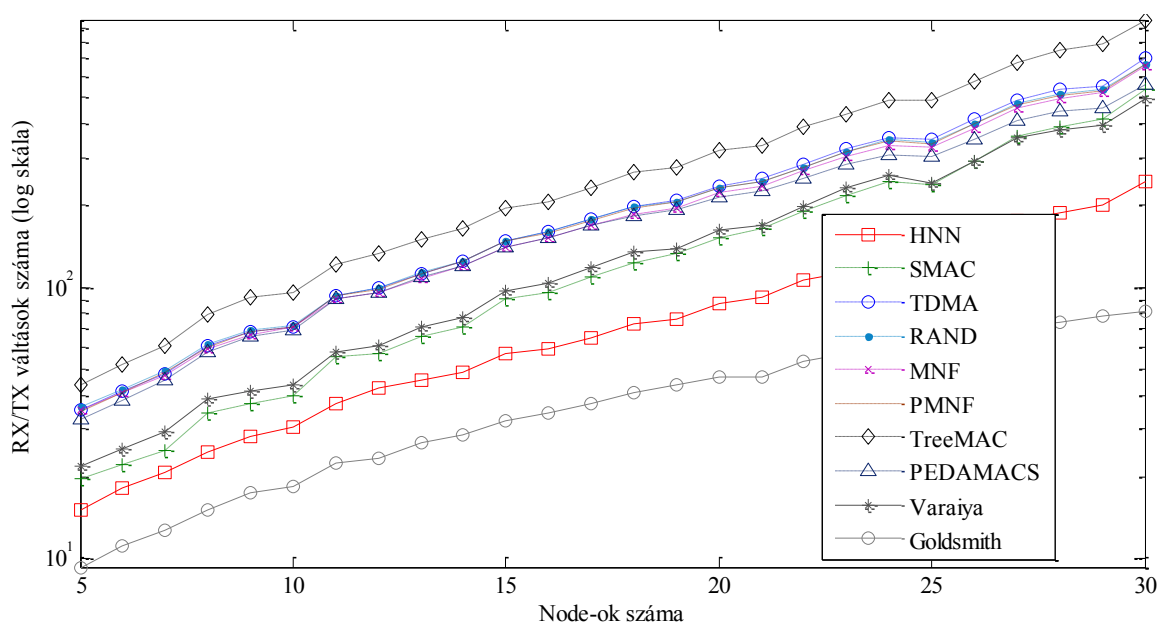
amely szerint minden egyes node-on azonos számú csomag keletkezik. Megfigyelhető, hogy a HNN alapú módszerrel létrehozott ütemezés áteresztőképessége megközelíti a legjobb tradicionális ütemező protokoll által biztosított áteresztőképességet.

Az ábrákon a WSN csomagütemező protokollok két családjába tartozó eljárások eredményei kerülnek összehasonlításra, amelyeket a 2.2.2 szakaszban ismertettem. (Az MNF és PMNF optimalizált TDMA protokoll.)

A 2.23. ábra homogén csomagterhelés mellett, a 2.24. ábra pedig heterogén csomagterhelés mellett illusztrálja azt, hogy a HNN alapú ütemező energiahatékonyság szempontjából olyan eredményt biztosít, amely megközelíti a legjobb hagyományos algoritmusok által meghatározott ütemezéseket. (A két ábrán az alacsonyabb érték, ami az összesített RX/TX váltások számát jelzi, a jobb.)



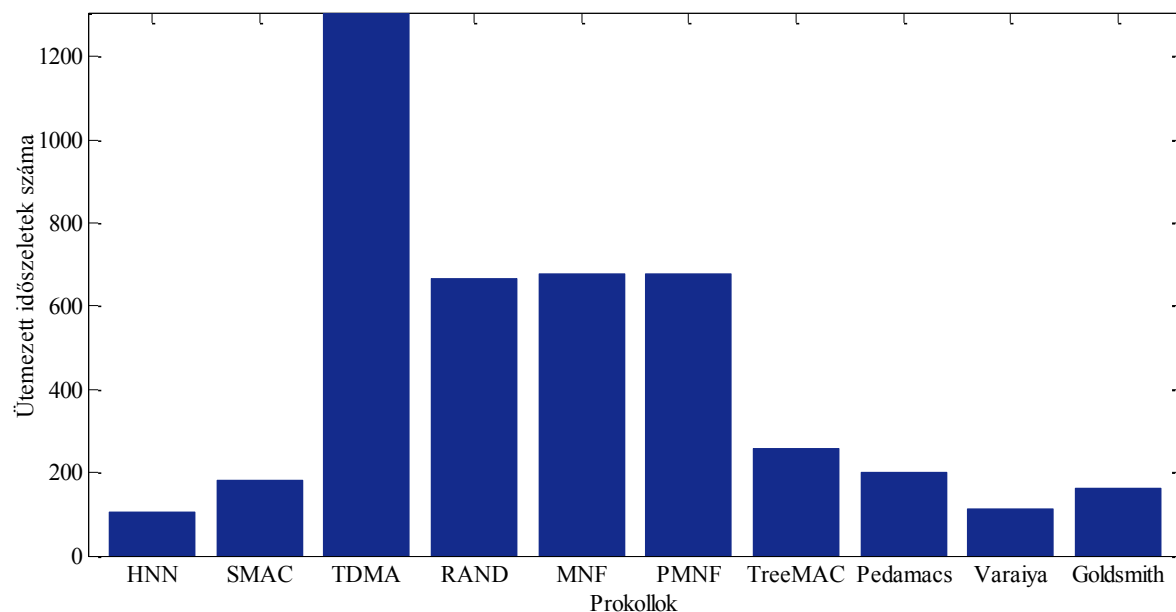
2.23. ábra Rádiós átkapcsolások száma homogén csomagterhelés mellett



2.24. ábra Rádiós átkapcsolások száma heterogén csomagterhelés mellett

A 2.25. ábra megmutatja, hogy a HNN alapú megoldás által megadott ütemezésben felhasznált időszeltek száma közel annyi, mint a legjobb megoldás.

Ezeket figyelembe véve kijelenthető, hogy a HNN alapú, kvadratikus optimalizálásra visszavezetett ütemezési módszer bonyolultabb feladat esetén is alkalmazható, ahol bonyolultabb peremfeltételek és célfüggvények mellett is lehetséges érvényes ütemezést létrehozni.



2.25. ábra Az ütemezett időszeltek átlagos száma heterogén terhelés mellett, 30 node-ot tartalmazó hálózatban

#### 2.5.4. Összefoglalás

A tesztek során megmutattam, hogy a fejezetben leírt apparátus alkalmas arra, hogy valós ütemezési problémákat oldjunk meg vele.

Az ütemezési feladatokat, amelyek kényszeres mátrix optimalizálásként írhatóak fel, áttranszformáltam kvadratikus optimalizálási feladatokká. Ezeket a kvadratikus optimalizálási feladatokat Hopfield neurális hálózat segítségével oldottam meg. A módszerrel létrehozott ütemezések jobb megoldások, mint a létező megoldók eredményei.

Több komplex feladaton keresztül illusztráltam a módszer modularitását, ugyanis az egyes célfüggvények szabadon megváltoztathatóak és kicserélhetőek. Elméletileg korlátlan számú célfüggvény, illetve peremfeltétel hozzáadható. Ugyanakkor az egyes feltételek esetén előfordulhat, hogy azok súlymátrixainak összege nullmátrix, amit külön kezelni kell. Továbbá minél nagyobb a HNN, annál nagyobb a valószínűsége annak, hogy lokálisan optimális megoldást talál a módszer. Emiatt az egyes futtatásokat többször meg kell ismételni, aminek következtében a teljes futtatási idő megnövekedhet.

A HNN alapú módszerhez kapcsolódóan foglalkoztam a módszer pontosságával, javaslatot tettem annak párhuzamos futtatására. A TWT problémára kapott eredmények alapján kijelenthető, hogy a javított HNN módszerek valóban képesek jobb ütemezési mátrix meghatározására.

A fentieknek megfelelően az ütemezési problémák hatékonyan és közel optimálisan megoldhatóak, egyszerűbb és bonyolultabb esetekben egyaránt.

*Harmadik fejezet***HATÉKONY MONITOROZÁS VEZETÉK NÉLKÜLI ÉRZÉKELŐ  
HÁLÓZATOKKAL**

A harmadik fejezetben a vezeték nélküli hálózaton alapuló komplex monitorozó rendszer adatfeldolgozásának optimalizálásával foglalkozom. A bevezetést, motivációkat és a jelenleg létező módszerek ismertetését követően egy olyan algoritmus kerül bemutatásra, amely a mérési idősorok elemzésével detektál kiugró értékeket és eseményeket a detekciós hibák valószínűségének minimalizálásával. A kidolgozott módszer a WSN hálózatok szűkös erőforrásaira szabott olyan értelemben, hogy minimalizálja a felhasznált energiát.

**3.1. Bevezetés, technológiai motivációk és nyitott kérdések**

Az előző fejezetben olyan ütemezési módszereket ismertettem, amelyek segítségével a szűkös erőforrások optimalizált kihasználása oldható meg különböző kényszerfeltételek alkalmazása mellett. A módszerek lehetséges alkalmazási területei között egy fontos csoportot képviselnek a monitorozó vezeték nélküli hálózati alkalmazások. Kapcsolódóan az előzőekhez, ebben a fejezetben a WSN elemei által rögzített mérési adatok feldolgozásával foglalkozom, ezen belül is az adatelemzésen alapuló esemény detekcióval. Fontos az alkalmazási környezet által támasztott követelményeket figyelembe venni, ugyanis a WSN hálózat node-jain kis komplexitású algoritmus futtatható, továbbá a hálózat elemei számára rendelkezésre álló energia véges. Ennek megfelelően egy olyan kis komplexitású módszer létrehozása a cél, amely elosztottan, a hálózat node-jain futtatható. A módszer az adatokban rejlő tér- és időbeli korrelációban rejlő információt használja fel a végső döntés meghozásához.

Az adatfeldolgozás számos módon történhet az adatbányászat eszközeivel, amelyek feladata a rejtett összefüggések, kapcsolatok felderítése a rendelkezésre álló adatokon. Számos precíz és gyors algoritmus áll rendelkezésre, azonban ezek a módszerek hagyományos számítási architektúrára tervezettek, és legtöbbjük nem, vagy nehezen adaptálható a monitorozó vezeték nélküli hálózat eszközeire és véges erőforrásaira. A létező adatbányászati problémák közül ebben a fejezetben az eltéréselemzéssel és detektálással foglalkozom, amelynek feladata a szokatlan értékek megtalálása [49]. Az elemeket, amelyek nem felelnek meg az adathalmaz általános jellemzőinek, tulajdonságaik nagymértékben eltérnek az általánostól, külön pontoknak vagy kiugró értékeknek (outlier) nevezzük. A WSN kontextusába helyezve szokatlan érték az alábbiak miatt fordulhat elő:

- adatrögzítési vagy, mérési hiba valamint kommunikációs probléma esetén;
- csalás, visszaélés, beavatkozás esetén, amikor a rádiós csomagok harmadik szereplő által módosulnak, meghamisíttatnak;

- esemény vagy krízis szituáció bekövetkezése esetén (például egy hirtelen növekvő hőmérséklet jelenthet tüzet);

Az outlier érték keletkezési okától függetlenül rendkívül fontos az értékek megtalálása egyrészt a mérési adatsor megtisztítása, másrészt a bekövetkező események detektálása érdekében. A detekciót követően lehetőség nyílik a beavatkozásra, amely a bekövetkezett esemény elhárítására, vagy további információ gyűjtésére vonatkozik.

Az outlier értékek és események detektálására számos megoldás létezik, azonban ezek nem, vagy csak részben optimalizáltak a WSN hálózatok során felmerülő körülményekre. Kevés az olyan létező módszer, amely a WSN hálózatok szűkös energiaforrását figyelembe veszi, ugyanakkor, a térben és időben korrelált értékeket úgy vizsgálja, hogy a korreláltságban rejlő információt is felhasználja a detekció során. Az általam bevezetett új módszer túl az előzőeken rendelkezik azzal a tulajdonsággal, hogy a hálózat egyes node-jai által mért értékeket képes párhuzamosan feldolgozni, továbbá lehetséges az algoritmus párhuzamos számítási architektúrára történő adaptációja.

A létező kihívások és problémák, amelyre ebben a fejezetben megoldásokat mutatok be a következők:

- *Hogyan lehet alacsony energiafogyasztás mellett WSN hálózatokban megvalósítani az esemény detekciót?*
- *Milyen kis számítási komplexitású algoritmussal lehetséges az outlier és esemény detekció megvalósítása?*
- *Mik az optimális döntési paraméterei az outlier detekciós módszerek és az esemény detekciós módszerek?*
- *Miként lehetséges a detekció és a paraméterek meghatározása elosztott módon, illetve sokprocesszoros számítási architektúrákon?*

## 3.2. Hagyományos megközelítések a feladat megoldására

A WSN alkalmazások során futtatott esemény detekció számos módon megvalósítható, amelyek áttekintésére ebben a szakaszban kerül sor. Az általam bemutatott esemény detekciós séma az outlier értékek detektálásán alapul, ezért az esemény detekciós módszerek bemutatását megelőzően a létező outlier detektálására kifejlesztett módszereket ismertetem.

### 3.2.1. Outlier detekciós módszerek

Az outlier értékek definiálására a két legtöbbet használt a Barnett által megfogalmazott „az outlier egy olyan megfigyelés (vagy megfigyelések részhalma), amely inkonzisztensnek tűnik az adathalmaz többi részével.” [50], [51]; illetve a Hawkins definíció: „az outlier egy olyan megfigyelés, amely olyannyira különbözik más megfigyelésektől, hogy feltételezhető, hogy eltérő folyamat generálta azt” [18].

Az ezeken a definíciókon alapuló outlier detekciós módszerek három csoportba oszthatóak az adatsorra vonatkozó információk és modellek alapján:

- Felügyelt módszerek, amelyek esetén mind a normál adatokra mind a kiugró értékekre felírunk egy modellt, ami alapján végrehajtjuk az optimális klasszifikációt.
- Nem felügyelt módszerek, amikor nincs kidolgozott modell a mérési értékekkel és az outlier értékekkel kapcsolatosan. Ekkor a mérési értékek megfigyelésével és folyamatos adaptációval tud az algoritmus helyes döntést hozni.
- Részben felügyelt módszerek esetén kizárólag a normális adatfolyamra írunk fel modellt. Ezt követően a normális folyamathoz nem illeszkedő értékeket minősítjük kiugró értéknek.

A WSN hálózatok esetén alkalmazható outlier detekciós módszereket Havinga valamint Hodge foglalja össze, amely alapján áttekintem a legfontosabb kategóriákat és algoritmusokat [52], [53]. Az eljárások közül a legrégebbi és egyben legegyszerűbb csoportot a *statisztikai alapú* módszerek jelentik, amelyek modell alapúak, azaz a megfigyelt adatok alapján megbecülik az adatok eloszlásának paramétereit. Akkor minősítenek egy értéket outlier-nek, ha az érték a tulajdonságai alapján alacsony valószínűséggel tartozhat az adathalmazhoz. A módszercsaládon belül léteznek parametrikus, és nem parametrikus módszerek egyaránt. A parametrikus módszerek esetén feltételezünk egy háttérmodellt, amely leggyakrabban normál eloszlást feltételez [54], [55], [56]. Nem parametrikus esetben a megfigyelt értékek alapján becsléssel határozza meg az eljárás az egyes értékek közötti korrelációt [57], [58]. (Például az értékek sűrűségét, vagy távolságát becsüli meg, és a távolság alapján hoz döntést a kiugró értékekről.)

A *klaszterezés alapú* algoritmusok esetén, a hálózat egyes elemei klaszterezéssel határozzák meg a mérési értékekről, hogy kiugró értékek-e vagy sem [26].

A *klasszifikációs alapú* módszerek családjába számos neurális hálózat alapú eljárás tartozik:

- Előrecsatolt neurális hálózatokkal megvalósított detekció. Ezen mesterséges neuronhálózatokat különböző mintahalmazzal tanítjuk, amely tartalmaz az adatsor értékeiből vett mintát, a klasszifikálandó értéket és az elvárt helyes döntés értékét. A tanítást követően a hálózat kimenete egy olyan bitsorozat, amely ezeket az outlier döntéseket reprezentálja. Ilyen módszert ismertet [59], [60].
- Replicator Neural Network (RNN) alapú módszerrel az egyes értékek kiugrási mértékét becsülik meg. Amennyiben ez a kiugrási mérték meghalad egy küszöbértéket a döntés szerint outlier értékről van szó [61].
- Support Vector Machine (SVM) alapú klasszifikációs módszercsalád. Ez a módszercsoport számos változó komplexitású és felépítésű rendszert alkalmaz, amelyekben egy SVM klasszifikálja egy adatsor elemeit [21], [25], [62].

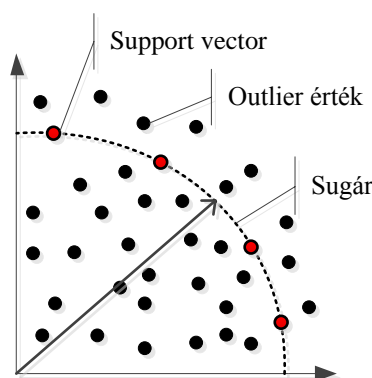
Szintén klasszifikáció alapú a *Bayesi hálókkal* megvalósított módszerek, ahol valószínűségi modellt építünk fel a változók tulajdonságaik reprezentálására [63], [64].

WSN hálózatok esetén csak a nem felügyelt tanuláson alapuló adaptív módszerek alkalmasak, ugyanis a mérendő objektumról és az egyes outlier értékekről gyakran körülményes, vagy nem lehetséges előzetesen modellt felállítani [65].

### 3.2.2. Jelenleg használatos esemény detekciós módszerek

Ebben a szakaszban, az irodalomban jelenleg ismertett WSN-ben működő esemény detekciós módszerek és megközelítések rövid áttekintése található. Az itt bemutatott módszerek jelentik az összehasonlítás alapját az általam bevezetett módszer teljesítőképesség-analízisének, ahol a detekciós pontosságot és az energiafelhasználást vizsgálom.

Az *SVM alapú* esemény detekciós módszerek családjába több megoldás is tartozik, amelyek a kernelfüggvényben és az SVM felhasználási módjában különböznek. Ugyanakkor mindegyik módszer klasszifikációs alapú, a döntés során az SVM az összes rendelkezésre álló adatot felhasználva osztályoz, a korábban megtanult paraméterek segítségével. Egy fontos és gyakran alkalmazott algoritmus-család az ellipszoidikus SVM-eken alapul [66], amely módszerek működését a 3.1. ábra vázolja.

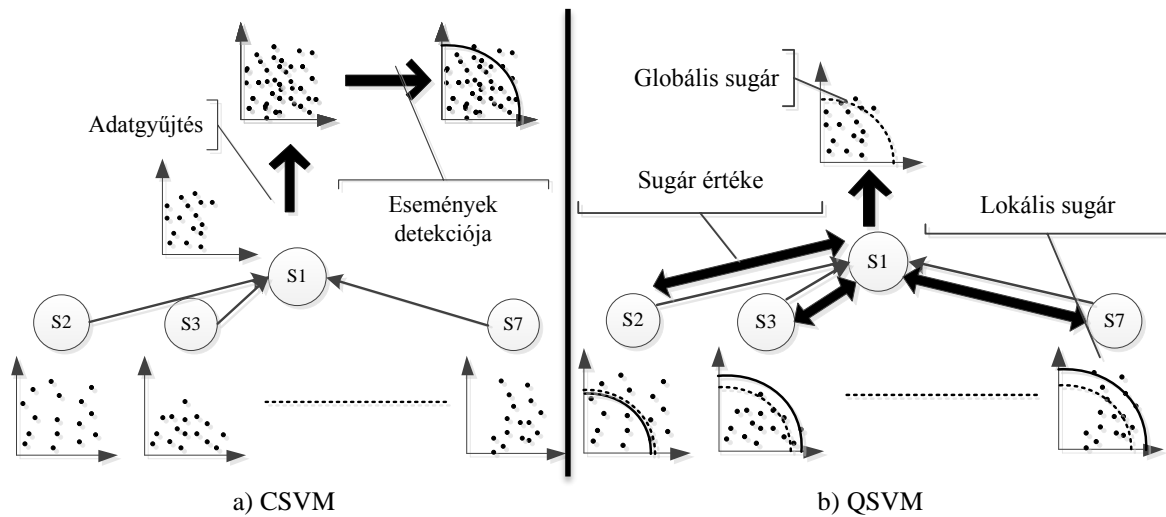


3.1. ábra Az SVM alapú módszerek működése

Quarter Sphere SVM módszer negyed-gömböt alkalmaz az SVM-ben [24], amely az előző ábrán is látható. A módszer centralizált változatát több kernelfüggvény használatával is bemutatták már, amelyek közül pontosság szempontjából az RBF kernel alapú QSVM bizonyult a legjobbnak [25].

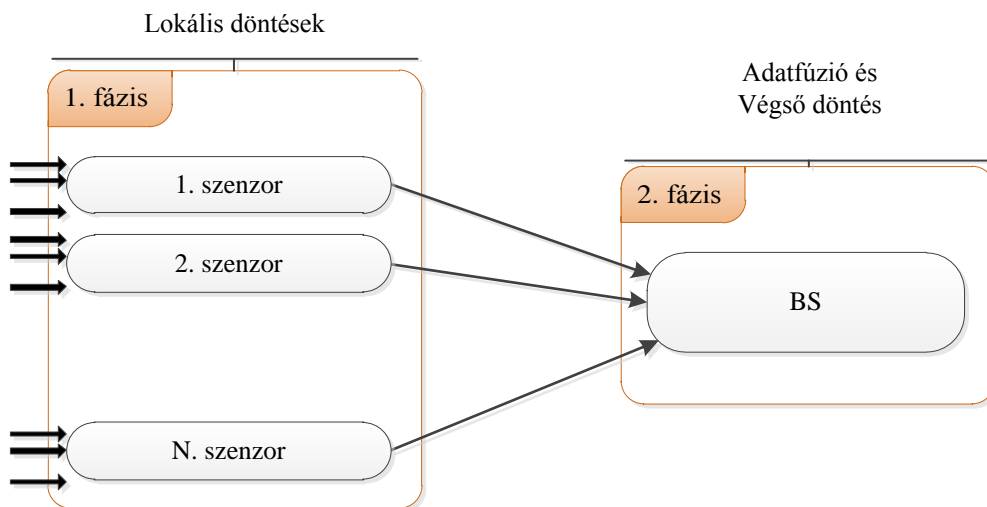
A módszercsaládkhoz tartozik az elosztottan működő Distributed QSVM, amely a detekciót lokálisan a node-on hajtja végre kooperálva a szomszédos node-okkal, és felhasználva azok információit. A pontosság növeléséhez ugyanakkor a node-ok közötti rádiós kommunikációk számát növeli [26]. Az elosztott és centralizált módszer működését a 3.2. ábra illusztrálja.

Hasonló eljárás a QSVM-hez a Bezdek által leírt Centered Hyperellipsoidal SVM (CESVM) módszer, amely a negyed hipergömb helyett negyed hiperellipszoidot használ az SVM-ben, ezzel is pontosabbá téve a klasszifikációt [26].



3.2. ábra a) Centralizált QSVM módszer és b) Elosztott QSVM módszer alapú esemény detekció működése

A SVM klasszifikáció helyett lehetséges más *neurális hálózat alapú* klasszifikátor használata. Egy előrecsatolt neurális hálózat alapú módszer kétlépcsős, az egyes szenzorok végrehajtanak egy klasszifikációt, amelynek eredményeit továbbküldve a BS-re a második lépésben azok fúziójával, klaszterezéssel hajtják végre a tényleges esemény detekciót [23]. A módszer sematikus működését a 3.3. ábra mutatja be.



3.3. ábra FFNN alapú klasszifikációs és adatfúzió

Ahogy a CESVM módszer során láthattuk a szomszédos node-ok adatainak felhasználásával a pontosságot lehet növelni. Ezt az elvet több további módszer használja, ahol a legközelebbi szomszédok méréseit kölcsönösen feldolgozzák a node-ok saját méréseik kiértékelésére. A szomszédok számát, vagyis a klaszter méretét a pontosság javítása érdekében lehet növelni, ugyanakkor ez a rádiós kommunikációk számát is megemeli, ezáltal a hálózat élettartama csökken [62], [67].

Léteznek továbbá *szavazás-alapú módszerek*, amelyekben az egyes node-ok lokálisan létrehoznak egy döntési fát és a döntésük eredményét a BS szavazás-alapon (többségi, vagy súlyozott) dolgozza fel és jelent eseményt.



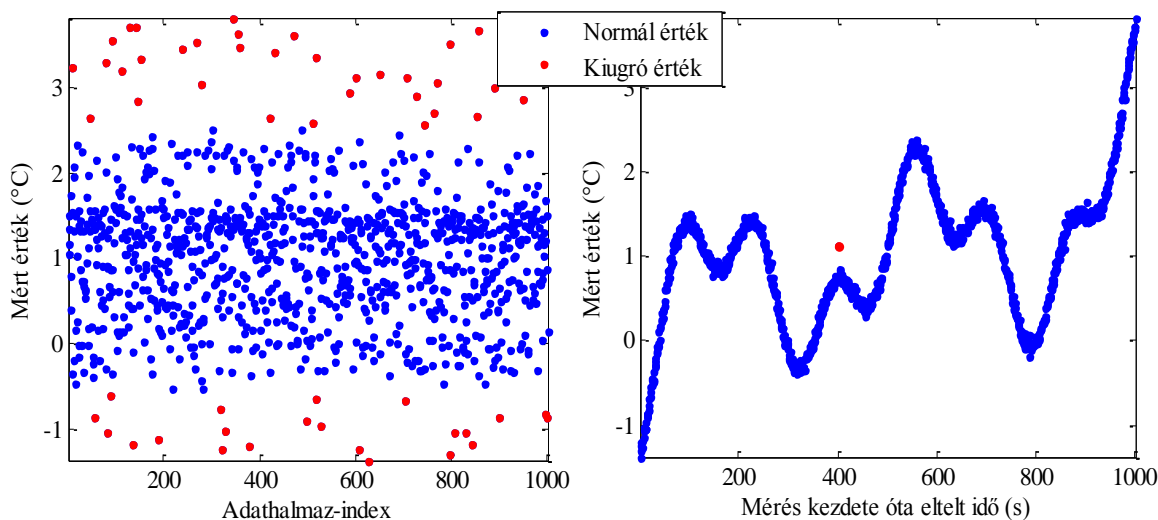
A *szabály és mintázat illesztés* alapú módszerek esetén az eseményeket előre definiáljuk, vagy adatbányászati eszközökkel meghatározzuk, és ezekre a szabályokra keresünk illeszkedéseket. Ezzel akár bonyolult feltételekre is illeszkedő események és eseménysorozatok detekciójára nyílik lehetőség, azonban a szabályok meghatározása nem mindig történhet automatikusan [27]. Az illesztési illetve döntési algoritmusra fuzzy logikát felhasználó megoldás is született [28].

A módszerek egy része a node-on hajtja végre a detekció egy részét, vagy egészét. Az efféle előfeldolgozás nyilvánvalóan csökkenti a rádión átküldött információ mennyiségét. Szintén lehet az átvitt adatmennyiséget mérsékelni a lényegkiemelés alapú módszerekkel, ahol az egyes szenzor node-ok előfeldolgozzák a méréseket, és azok eredményét juttatják el a BS-re [29]. Az előfeldolgozás mértékétől függően a módszerek két csoportra oszthatóak:

- A node-on nem történik klasszifikáció, mert azt a BS-en hajtjuk végre és ott történik az esemény detekció is fuzionálást követően.
- A node-on klasszifikáció is történik és a BS egy szavazás alapú döntést hoz, hasonlóan egy a korábban említett módszerhez.

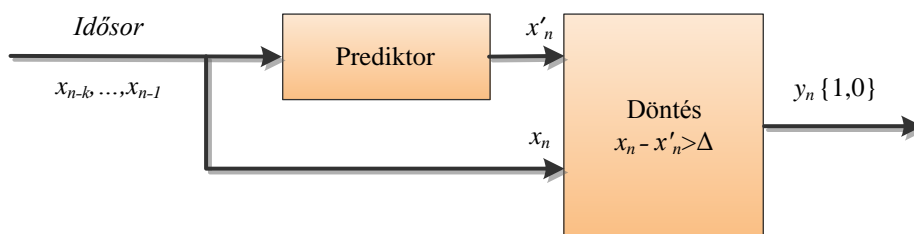
### 3.3. Új, kis komplexitású outlier detekciós algoritmus

Az outlier detekciós problémájának megoldására egy olyan kis komplexitású, a hálózat node-jain is futtatható algoritmus javasolok, amely a mérési idősorok predikcióján alapul. A módszertől alacsony detekciós hiba-valószínűséget várok el, más eljárásokhoz viszonyítottn, mivel kifejezetten olyan adatsorokhoz terveztem, amelyek esetén az egymást követő értékek korreláltak. Ez megfelel a monitorozó rendszerek által mért idősorok modelljének, aminek elvi háttérét a 3.4. ábra mutatja be. (A mérési adatsorokról feltételezem, hogy egy olyan idősor elemei, ahol az egymást követő értékek korreláltak. A rögzített értékek mérési pontatlanságát normál eloszlású zajjal modellezem, ami a megfigyelt objektum egy tulajdonságának valós értékéről pontatlan mérési információt szolgáltat.) Amennyiben egy mérési idősor értékeiből képezett halmazon eloszlás alapú detekciót hajtunk végre, a halmaz átlagának és szórásának ismeretében, outlier-nek minősítünk számos értéket. (Normál eloszlást feltételező módszer esetén az ábra bal oldalán látható eredményt kapjuk.) Amennyiben a mérési értékeket egy idősor elemeiként vizsgáljuk az outlier detekció eredménye szignifikánsan eltér.



3.4. ábra Adatok rendezetlen módon és idősor értékeiként ábrázolva és figyelembe véve

Mivel az idősor egymást követő elemei korreláltak egymással, valamint a korábbi mérési értékekből megbecsülhető a következő mérési érték (amennyiben kellően gyakran történik mintavételezés), ezért a becült és tényleges mérés összehasonlításából következtethetünk outlier jelenlétére. A módszert a 3.5. ábra foglalja össze.



3.5. ábra Predikciós módszer blokkdiagramja

Formálisan megfogalmazva a módszer a tényleges mérési értéket ( $x_k^{(i)}$ ) egy, a korábbi  $l$  mérésekből előre becsült értékkel ( $\hat{x}_k^{(i)}$ ) veti össze. A becsült érték az idősor korábbi elemeiből határozható meg a prediktor tanult paramétereinek ( $\mathbf{W}^{(i)}$ ) felhasználásával, ahol a  $\Gamma(\cdot)$  függvény a prediktor:

$$\hat{x}_k^{(i)} = \Gamma(\mathbf{W}^{(i)}, \mathbf{x}_{k-1, \dots, k-l}^{(i)}) \quad (3.1)$$

Amennyiben a különbségük meghalad egy előre definiált küszöböt ( $\Delta^{(i)}$ ), akkor az értéket outlier-nek minősítjük. A bináris döntés eredménye:

$$y_k^{(i)} = \text{sgn}\left(-|x_k^{(i)} - \hat{x}_k^{(i)}| + \Delta^{(i)}\right) \in \{0, 1\} \quad (3.2)$$

Amennyiben több idősorunk van minden egyes  $i$  idősorhoz különböző  $\Delta^{(i)}$  küszöbérték tartozhat. Az optimális döntési küszöb meghatározását külön tárgyalom.

A predikció megvalósítására számos algoritmus áll rendelkezésre [49], amelyek közül a legfontosabb módszerek: (i) lineáris regresszió; (ii) logaritmikuss regresszió; (iii) SVM alapú regresszió; (iv) perceptron-hálózat alapú regresszió [68], [69], [70]. A megfelelő numerikus prediktor paramétereit tanulási folyamat során kell az idősorhoz igazítani. Továbbá a paraméterek reguláris frissítésével a predikció pontosságát a kívánt szint felett lehet tartani.

### 3.3.1. Optimális megoldás lineáris predikcióval

A predikciós alapú outlier detekciós módszer esetén fontos kérdés az optimális döntéshez tartozó döntési küszöb meghatározása. Ebben a fejezetben megmutatom, hogy amennyiben az outlier detekciós módszerben a soron következő érték becslését a jól ismert lineáris predikció [72] módszerével végzem, akkor az optimális döntési paraméterek analitikusan meghatározhatók.

**Tétel.** A lineáris predikciót felhasználó outlier detekciós módszer optimális paramétereit meghatározhatók a 3.6. ábra szerinti algoritmussal. A számítás során felhasználom a mérési értékeket befolyásoló zaj statisztikai paramétereit ( $\mathcal{N}(0, \sigma)$ ), továbbá az outlier érték bekövetkezésének valószínűségét ( $p$ ), illetve az outlier-eket generáló folyamat összegzett eloszlási függvényét ( $F(\cdot)$ ). Így felírható egy modell, amelyben analitikusan meghatározhatóak a hibás döntésekhez tartozó valószínűségek. Ezek a minimalizálásával határozható meg az optimális döntési paraméterek.

**1. Lépés:** A statisztikai modellhez állapítsuk meg a paramétereket:  $\sigma, p, F()$

**2. Lépés:** Konvolúcióval határozzuk meg  $g^{(l)}(x), l=1, \dots, L$  függvényt

**3. Lépés:** Számítsuk ki a feltételes valószínűségi értékeket a modell paramétereit felhasználva:

$$P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \mid \alpha_k = 0\right), P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \mid \alpha_k = 1\right)$$

**4. Lépés:** Oldjuk meg a kényszeres optimalizálási feladatot  $\Delta_{opt} : \min_{\Delta} P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \mid \alpha_k = 0\right)$  a

$$P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \mid \alpha_k = 1\right) = \varepsilon \text{ feltétellel.}$$

3.6. ábra Algoritmus az optimális küszöbparaméter meghatározása

**Bizonyítás.** Lineáris prediktor alkalmazása esetén a predikált érték a következőképpen számítható a korábbi (3.1) egyenlettel összhangban:

$$\hat{x}_k = \sum_{i=1}^l w_i x_{k-i}, \quad (3.3)$$

ahol a  $w_i$  súlyok beállítása az  $x(n)$  tanulóhalmaz segítségével az alábbi összefüggés szerint történik:

$$\mathbf{w}_{opt}^{(i)} : \min_{\mathbf{w}} \sum_{k=1}^N |x_k^{(i)} - \hat{x}_k^{(i)}|^2. \quad (3.4)$$

A  $\Delta$  küszöbparaméter meghatározásához a következő modellt alkalmazom. Az idősort egy autoregresszív folyamat (AR) segítségével írom le: [71]

$$\xi_k = \sum_{j=1}^L a_j \xi_{k-j} + \nu_k, \quad (3.5)$$

ahol  $\nu$  egy normáloszlású valószínűségi változó:  $\mathcal{N}(0, \sigma)$ , amely a külső hatások okozta zajt reprezentálja. (Ez lehet például mérési zaj, vagy egyéb a mérési érték pontosságát rontó tényező.)

Az outlier értékeket egy másik valószínűségi változó reprezentálja a következők szerint:

$$\chi_k = \alpha_k \beta_k, \quad (3.6)$$

ahol  $\alpha_k \in \{0, 1\}$  és  $P(\alpha_k = 1) = p$ . (Azaz  $p$  valószínűséggel az idősor egy értékét megváltozik (hozzáadással), így outlier érték keletkezik.) A valószínűségi változó sűrűségfüggvénye és eloszlásfüggvénye a következő:

$$F(x) := P(\beta_k < x), f(x) = \frac{dF(x)}{dx} \quad (3.7)$$

és

$$P(\beta_{k-i_1} + \dots + \beta_{k-i_l} < x) = G^{(l)}(x), g^{(l)}(x) = \frac{dG^{(l)}(x)}{dx} = f * \dots * f(x). \quad (3.8)$$

A  $*$  operátor a konvolúciót jelenti. A ténylegesen rögzített mérési értékek az eredeti megfigyelt folyamat, valamint a hozzáadódott értékek összegeként írható fel:

$$\hat{\xi}_k = \xi_k + \chi_k. \quad (3.9)$$

Tételezzük fel, hogy a rendszer installálásakor a lineáris prediktor tanítható az outlier-ektől mentes adatok segítségével ( $\hat{\xi}_k = \xi_k$ ), tehát az optimális prediktor paramétereit képesek vagyunk meghatározni:

$$\tilde{\xi}_k = \sum_{j=1}^L w_j \xi_{k-j}, \quad (3.10)$$

ahol az optimális súlyparaméterek meghatározásához a két folyamat közötti különbséget minimalizáljuk:

$$\mathbf{w}_{opt} : \min_{\mathbf{w}} \mathbb{E}(\xi_k - \tilde{\xi}_k)^2. \quad (3.11)$$

A minimalizálandó különbségbe behelyettesítve a két folyamat paramétereit:

$$\mathbb{E}(\xi_k - \tilde{\xi}_k)^2 = \mathbb{E}\left(\sum_{j=1}^L a_j \xi_{k-j} + \nu_k - \sum_{j=1}^L w_j \xi_{k-j}\right)^2. \quad (3.12)$$

Amennyiben a prediktor adaptálása megtörtént, akkor az optimális súly

$$\mathbf{w}_{opt} = \mathbf{a}, \quad (3.13)$$

azaz

$$\min_{\mathbf{w}} \mathbb{E}(\xi_k - \tilde{\xi}_k)^2 = \mathbb{E}(\nu_k^2) = \sigma^2. \quad (3.14)$$

Legyen  $\eta_k = \sum_{j=1}^L w_j \hat{\xi}_{k-j}$ , ahol a  $w_j$  paraméterek optimálisak. (A továbbiakban a tanítást sikeresnek feltételezem, azaz  $w_{opt,i} = a_i, i=1, \dots, L$ ). A döntési szabály szerint, a  $\Delta$  küszöbparamétert használatával, ha  $|\hat{\xi}_k - \eta_k| > \Delta$  akkor outlier-ként dönt az algoritmus, ha  $|\hat{\xi}_k - \eta_k| \leq \Delta$  akkor normális értéként. A küszöbparaméter optimális megválasztásához az elsőrendű hibát vizsgálom. Annak a valószínűsége, hogy a döntés hibás és a kérdéses érték nem outlier a következő:

$$P\left(|\hat{\xi}_k - \eta_k| < \Delta | \alpha_k = 1\right). \quad (3.15)$$

A (3.15) átírható a következők szerint:

$$P\left(|\hat{\xi}_k - \eta_k| < \Delta | \alpha_k = 1\right) = \sum_{l=1}^{L-1} P\left(|\hat{\xi}_k - \eta_k| < \Delta | \alpha_k = 1, l \text{ outlier}\right) P(l \text{ outlier}). \quad (3.16)$$

ahol „ $l$  outlier” jelentése az, hogy a vizsgált  $L$  hosszú ablakban levő  $l$  érték outlier-e. A kifejezés kibontva:

$$\sum_{l=1}^{L-1} P\left(|\hat{\xi}_k - \eta_k| < \Delta | \alpha_k = 1, l \text{ outlier}\right) \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.17)$$

Az outlier-ek valamint az idősor modelljét behelyettesítve azt kapom, hogy:

$$\sum_{l=1}^{L-1} P\left(\left|\sum_{j=1}^L a_j \xi_{k-j} + \nu_k + \beta_k - \sum_{j=1}^L w_j \xi_{k-j} - a_i \beta_{k-i} - \dots - a_i \beta_{k-i}\right| < \Delta | \alpha_k = 1, l \text{ outlier}\right) \cdot \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.18)$$

A  $\beta_k = y$  és  $a_i \beta_{k-i} + \dots + a_i \beta_{k-i} = z$  behelyettesítést követően:

$$\sum_{l=1}^{L-1} \iint_{y,z} P(|\nu_k + y - z| < \Delta | \alpha_k = 1, l \text{ outlier}) f(y) g^{(l)}(z) \mathbf{d}y \mathbf{d}z \cdot \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.19)$$

Végül a  $\nu$  zaj paramétereit is behelyettesítve az eredmény:

$$\sum_{l=1}^{L-1} \iint_{y,z} \left\{ \Phi\left(\frac{\Delta - y + z}{\sigma}\right) - \Phi\left(\frac{-\Delta - y + z}{\sigma}\right) \right\} f(y) g^{(l)}(z) \mathbf{d}y \mathbf{d}z \cdot \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.20)$$

Hasonlóan megvizsgáltam azt az esetet, amikor a rendszer által meghozott hibás döntés szerint a kérdéses érték outlier:

$$P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \mid \alpha_k = 0\right) = \sum_{l=1}^{L-1} P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \mid \alpha_k = 0, l \text{ outlier}\right) \cdot n \cdot P(l \text{ outlier}). \quad (3.21)$$

Átírva és behelyettesítve:

$$\begin{aligned} & \sum_{l=1}^{L-1} P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \mid \alpha_k = 0, l \text{ outlier}\right) \binom{L}{l} p^l (1-p)^{L-l} = \\ & = \sum_{l=1}^{L-1} P\left(\left|\sum_{j=1}^L a_j \xi_{k-j} + v_k - \sum_{j=1}^L w_j \xi_{k-j} - a_{i_1} \beta_{k-i_1} - \dots - a_{i_l} \beta_{k-i_l}\right| \geq \Delta \mid \alpha_k = 0, l \text{ outlier}\right) \\ & \binom{L}{l} p^l (1-p)^{L-l} \end{aligned} \quad (3.22)$$

Az következő átírással  $\beta_k = y$ ,  $a_{i_1}$  és  $\beta_{k-i_1} + \dots + a_{i_l} \beta_{k-i_l} = z$  a formula átírható a

$$\sum_{l=1}^{L-1} \int_z P(|v_k - z| \geq \Delta \mid \alpha_k = 0, l \text{ outlier}) g^{(l)}(z) dz \binom{L}{l} p^l (1-p)^{L-l} \quad (3.23)$$

alakba. Hasonlóan az előzőekhez a modell paramétereit behelyettesítve, azt kapjuk, hogy:

$$\sum_{l=1}^{L-1} \int_z \left\{1 - \Phi\left(\frac{\Delta + z}{\sigma}\right) + \Phi\left(\frac{-\Delta + z}{\sigma}\right)\right\} g^{(l)}(z) dz \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.24)$$

Az  $1 - \Phi(\cdot)$ -re vonatkozó összefüggést felhasználva a végső formula:

$$\sum_{l=1}^{L-1} \int_z \left\{\Phi\left(\frac{-\Delta - z}{\sigma}\right) + \Phi\left(\frac{-\Delta + z}{\sigma}\right)\right\} g^{(l)}(z) dz \binom{L}{l} p^l (1-p)^{L-l}. \quad (3.25)$$

Felhasználva a (3.20) és (3.25) képletekben leírt eredményeket a  $\Delta_{opt}$  megtalálható egy kényszeres optimalizálási feladat megoldásával, ahol:

$$\Delta_{opt} : \min_{\Delta} P\left(\left|\hat{\xi}_k - \eta_k\right| \geq \Delta \mid \alpha_k = 0\right) \quad (3.26)$$

az optimalizálás feltétele pedig:

$$P\left(\left|\hat{\xi}_k - \eta_k\right| < \Delta \mid \alpha_k = 1\right) = \varepsilon, \quad (3.27)$$

ahol az  $\varepsilon$  egy előre definiált konstans. A megfigyelt valós folyamat paramétereinek ismeretében az optimális döntési küszöb meghatározható. ■

### 3.3.2. Megoldás nem lineáris predikcióval

A predikció nemcsak lineáris predikcióval, hanem nem lineáris módszerrel is végrehajtható. A szükségességét a nemlineáris módszerek bevetésének a későbbi 3.14. ábra illusztrálja. Nem lineáris prediktor használatkor a valós folyamat becslése kisebb összegzett négyzetes hiba mellett valósítható meg, megfelelő tanítás, illetve adaptáció esetén.

A lineáris prediktor helyett előrecsatolt neurális hálózatot (FFNN) alkalmaztam. Az FFNN-ek [73] megfelelő tanítás esetén tetszőleges  $L^2$ -beli függvény tetszőleges approximációjára alkalmasak [74], [75], [76].

A (3.1) egyenlettel összhangban az FFNN kimenete az  $\mathbf{x}$  bemenet és a  $\mathbf{W}$  súlyok esetén nemlineáris kombinációja a következők szerint:

$$\text{Net}(\mathbf{x}, \mathbf{w}) = \hat{x} = \phi \left( \sum_{i=1}^{N_L} \mathbf{W}_i^{(L)} \cdot \phi \left( \sum_{j=1}^{N_{L-1}} \mathbf{W}_{i,j}^{(L-1)} \cdot \phi \left( \sum_{m=1}^{N_L} \mathbf{W}_{l,m}^{(1)} \mathbf{x}_m \right) \right) \right) \quad (3.28)$$

A tanításhoz szükséges egy tanulóhalmaz előállítás, amelyet a valós mérésekből lehet megkonstruálni.

$$\tau = \{(\mathbf{x}_i, d_i), i = 1 \dots K\} \quad (3.29)$$

A tanulóhalmaz  $K$  számú tanulómintát tartalmaz, amely minták két komponensből állnak:

- $\mathbf{x}_i$  egy vektor, ami az idősorból egy  $L$  hosszú ablakot tartalmaz, illetve
- kívánt kimenet, azaz idősor következő eleme.

FFNN helyett Radial Basis Function (RBF) alapú hálózat is alkalmazható, ekkor a hálózat kimenete szintén nemlineáris kombinációja a bemenetnek a következők szerint:

$$\text{Net}(\mathbf{x}, \mathbf{w}) = \hat{x} = \sum_{i=1}^N w_i \phi(\mathbf{x}, \mathbf{c}_i) + w_0. \quad (3.30)$$

Itt a  $\phi(\cdot)$  függvény körszimmetrikus vagy gömbszimmetrikus függvény:

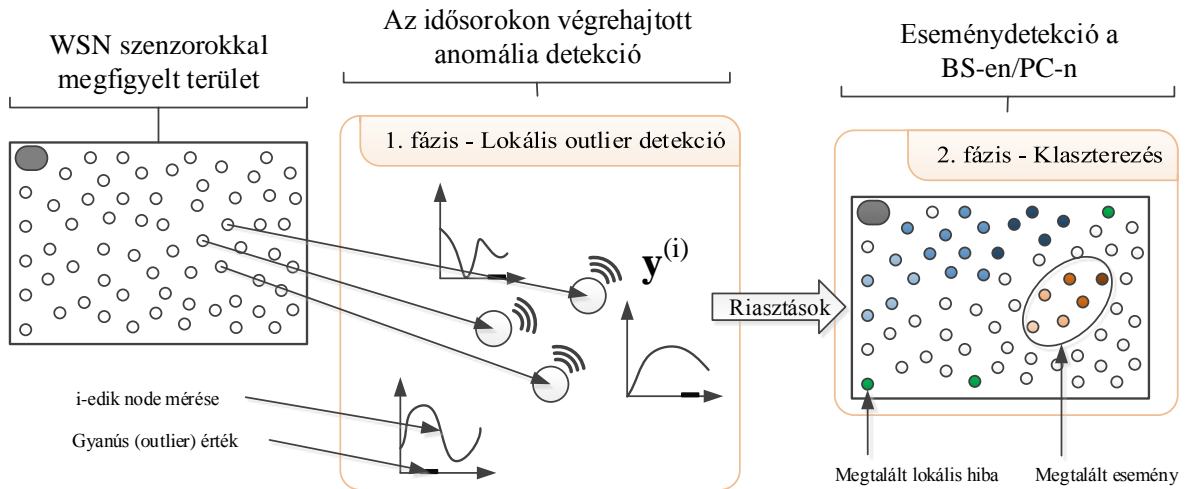
$$\phi_G = \exp\left(-\frac{r^2}{2\delta^2}\right) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_i\|^2}{\delta_i^2}\right) \quad (3.31)$$

A dinamikus tanulási eljárás egy mátrix invertálásra redukálható [77].

Az eredményeket bemutató fejezetben látható, hogy a nemlineáris predikcióval végrehajtott detekció jobb eredmény elérésére is képes. Mindazonáltal a helyes döntési küszöb meghatározása jóval összetettebb probléma, amelyre jelenleg analitikus eredmény nem létezik.

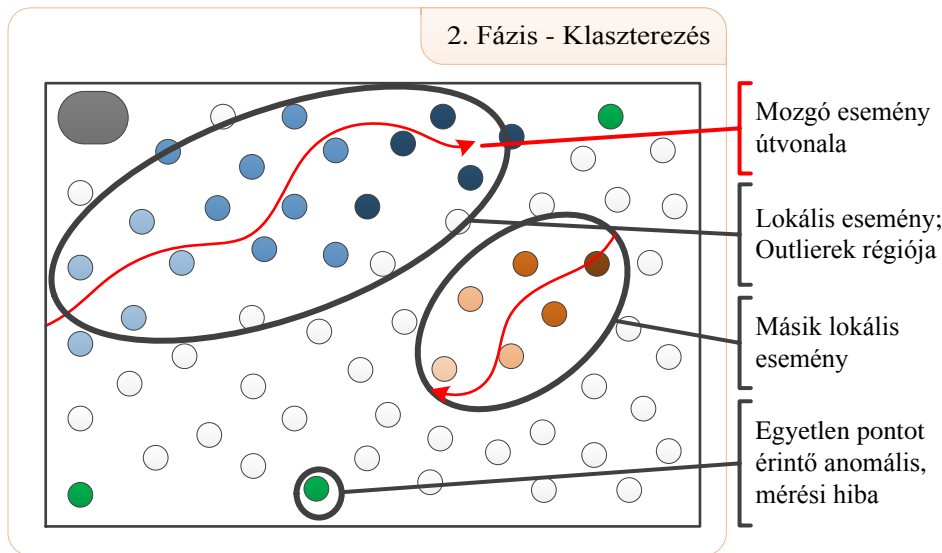
### 3.4. Új, klaszterezésen alapuló esemény detekciós algoritmus

Az ismert outlier detekciós algoritmusra alapozva, egy adott lokális környezetből rögzített adatsorok elemzésével eldönthető, hogy abban a környezetben bekövetkezett-e szokatlan esemény. A komponensek viszonyát a 3.7. ábra demonstrálja.



3.7. ábra A predikció alapú outlier detekció és a klaszterező komponensek viszonya

A 3.8. ábra olyan lehetséges eseteket mutat be, ahol egy adott lokális környezetből egy, néhány, vagy sok node jelentett outlier-t. Amennyiben egy adott környezetben a node-ok többsége outlier jelez, akkor az adott környezetben esemény következett be. A detekciós módszer így az esemény helyzetéről és kiterjedéséről (érintettségéről) is nyújt információkat. Ha csak szórványosan néhány node jelez, vagy elszigetelt outlier esetén lokális mérési hibát, vagy meghibásodást feltételezhetünk.



3.8. ábra Lokális események valamint egyetlen pontot érintő anomália demonstrációja

A végleges döntést az outlier értéket jelző node-ok régiójában található összes node-ok által nyújtott információ alapján többségi szavazási alapon lehet meghozni.

Fontos kérdés az, hogy a legkisebb döntési hibavalószínűség elérése érdekében hogyan válasszuk meg a klasztereket. A következőkben erre a kérdésre írok le egy algoritmust.



**Tétel.** Az alacsony hibavalószínűség elérése érdekében az optimális klaszterek meghatározhatóak. Az algoritmus, amelyet a 3.9. ábra mutat be, felhasználja a monitorozó rendszerben található mérési pontok pozícióját és egymástól való távolságát. Továbbá figyelembe veszi azt a ténytet, hogy az eseményt generáló jelenség minden mérési pontra más hatással van, a hatás fordítottan arányos a jelenség és mérési pont közötti távolsággal.

**FORALL**  $l = 1 \dots N$

**1. lépés:** Teszt eseményt generálunk, amely az  $l$  mérési pont által érzékelhető a legjobban, és amelyre lefuttatjuk a predikciós alapú outlier detekciót.

**2. lépés:** Kiszámítjuk a detektálás valószínűségét minden egyes mérési pontra az alábbiak szerint  $P_i := \Phi\left(-\frac{\Psi(r_{li})}{\sigma}\right)$ .

**3. lépés:** Meghatározzuk a  $C_l$  klasztert, a  $\max_c \sum_{\substack{\mathbf{y} \in \{0,1\}^{C_l} \\ w(\mathbf{y}) \geq \lceil \frac{L}{2} \rceil}} \sum_{i=1}^L y_i P_i$  mennyiség maximalizálásával.

Erre a célra tetszőleges keresési algoritmus felhasználható.

**END FOR**

3.9. ábra Klaszterek optimális meghatározása

**Bizonyítás.** A döntési folyamat formálisan megfogalmazva a következő: adott a mérési pontok egy halmaza  $i = 1, \dots, N$ . Tételezzük fel, hogy az eseményhez a legközelebbi mérési pont az  $l$ -edik. Ekkor ebben az idősorban az esemény egy  $m_l$  értéket generál, míg a többi idősorban az  $m_i = \Psi(r_{li})$  érték formájában fejeződik ki. A  $\Psi(\cdot)$  függvény egy monoton csökkenő függvény, reprezentálva a fizikai valóságban fellépő csillapodást. Az érzékelt érték az esemény által generált mennyiség és az érzékelés hibájából eredő zaj összege. Ennek megfelelően a ténylegesen rögzített érték  $\xi_i = \nu + m_i$ ;  $i = 1, \dots, N$  és  $\nu \sim \mathcal{N}(0, \sigma)$ . A  $\nu$  egy normális eloszlású változó, amely a mérést megzavaró zaj/hiba, amely minden egyes node-on megjelenik. Az  $N$  mérési pontból  $L$  felhasználásával egy a bekövetkezett esemény helyszínéhez közel található klaszter képezhető:

$$C_l := \{i_1, \dots, i_L\}, \quad (3.32)$$

ahol a klaszter számossága  $|C_l| = L$ . A predikció alapú döntéshez meghatározott optimális döntési küszöböt ( $\Delta$ ) felhasználva egy indikátor függvényt írok fel:

$$I_{i_j} := \begin{cases} 1 & \text{if } \xi_{i_j} > \Delta^{(i)} \\ 0 & \text{különben} \end{cases}. \quad (3.33)$$

Amennyiben az alábbi egyenlőtlenség teljesül a módszer detektálja az eseményt:

$$\sum_{j=1}^L I_{i_j} > \left\lceil \frac{L}{2} \right\rceil, \quad (3.34)$$

azaz a klaszterben levő mérési pontok több mint fele érzékelt az eseményt és ez outlier értéként detektálásra került.

Az optimális klasztereket a korrekt esemény detekció valószínűségének maximalizálásával lehet meghatározni. Egy  $j$  esemény, amennyiben az  $i$  idősorban detektálunk a következő valószínűséggel jellemezhető:

$$P_{i_j} := P(\xi_{i_j} > \Delta) = P(v + m_{i_j} > \Delta). \quad (3.35)$$

A felállított modell felhasználásával ez a következőképpen írható át:

$$P_{i_j} = 1 - \Phi\left(\frac{m_{i_j}}{\sigma}\right) = \Phi\left(-\frac{\Psi(r_{i_j})}{\sigma}\right). \quad (3.36)$$

Egyetlen klaszter esetén a korrekt detekció valószínűsége a következő módon írható fel:

$$P(\text{korrekt esemény detekció}) = \sum_{\substack{\mathbf{y} \in \{0,1\}^{C_l} \\ w(\mathbf{y}) \geq \lfloor \frac{L}{2} \rfloor}} \sum_{i=1}^L y_i P_i \quad (3.37)$$

Itt összegzem az egyes bekövetkező outlier értékeknek a valószínűségét egy adott klaszter esetén. Ennek megfelelően a feladat olyan  $C_l := \{i_1, \dots, i_L\}$  klaszter meghatározása, amely maximalizálja a helyes döntés valószínűségét. ■

Az adaptáció során meghatározott klasztereket az algoritmus a tényleges, valós működés során felhasználja a többségi szavazás során a lokalitásokhoz. Amennyiben egy mérési pont egyidejűleg több klaszter tagja is, akkor az adott mérési pontról beérkező outlier riasztást minden egyes, a mérési pontot tartalmazó klaszterre ki kell értékelni. Ezzel garantálható, hogy az egyes riasztások a klaszterhatároktól függetlenül kiértékelődjenek.

## 3.5. Az esemény detekciós módszer megvalósítása WSN alapú monitorozó környezetben

Az előzőleg bemutatott módszer képességeinek figyelembe vételével az alábbiakban ismertetésre kerülő WSN hálózat működési modelljében alkottam meg az esemény detekciós algoritmust.

### 3.5.1. A WSN hálózat mérési modellje

A WSN hálózat mérési protokolljával kapcsolatos feltételezéseim:

- Az idősor elemei időben korrelált értékek, autoregresszív folyamatként modellezhetőek.
- A hálózatban a mérő node-ok kellőképp sűrűen helyezkednek el ahhoz, hogy a mérési értékek térben korreláltak legyenek egymással.
- Minden egyes vizsgált idősor esetén az idősorban előforduló outlier értékek független valószínűségi változóval jellemezhetőek.
- A valós eseményeket egyidejűleg több szenzor is érzékeli.

Az alkalmazással kapcsolatosan a következő működési modellt feltételezem, amely megfelel egy low duty-cycled hálózat paramétereinek [78]:

- A megfigyelt értékekre valós időben nincsen szükség, mert azt későbbi kiértékelés céljából tároljuk.
- A bekövetkező eseményekről a rendszer azonnali értesítést kell, hogy küldjön.
- A hibás értékeket a mérések közül eliminálni kell.

Az előzőek figyelembe vételével a tipikus alkalmazási terület a következő: egy megfigyelés alatt álló környezetben keresünk mozgó objektumok által generált eseményeket [48].

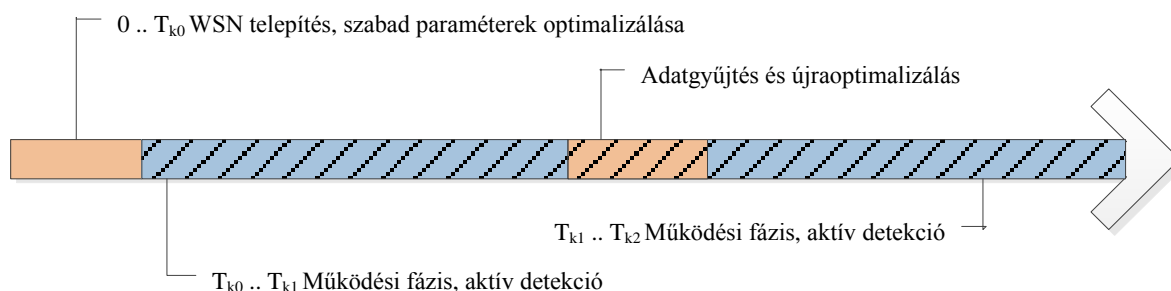
Az algoritmus energiahatékonyságát [79], [80] illető feltételezések és célok a következőek:

- Az outlier döntésre vonatkozó ( $\mathbf{y}^{(i)}$ ) információk beküldése az egyes node-ok által végzett detekciót követően kevesebb számú és kisebb payload-dal rendelkező csomagok forgalmazását jelenti.
- Minden mérési érték szervezett és tömörített módon történő begyűjtése az összes rádióforgalom mértékét csökkenti.
- A paraméterek, a tanuló algoritmusok adaptációja és a kiszámolt paraméterek node-okra juttatása extra rádiós forgalom. Azonban ezt lehetséges részben vagy egészben a node-ok közötti szinkronizációra, vagy a topológia felderítésre használt adatcsomagokba integrálni.

A leírt WSN környezetben működő esemény detekciós algoritmus működését a következő szakaszban foglalom össze.

### 3.5.2. Az esemény detekciós algoritmus protokollja

A létrehozott módszer működése során a következő 3.10. ábra által összefoglalt működési fázisok léteznek.



3.10. ábra Esemény detekciós algoritmus munkafázisai

A fázisok részletesen a következők:

- Az  $N$  node-ból álló hálózat telepítése, inicializálása, mely során  $T_{k_0}$  ideig mérést végeznek a node-ok, majd a méréseket elküldik a BS-re, ahol a minden  $i$  node-ra a paraméterek kiszámítása megtörténik. A paramétereket a node-okra küldi a BS.
- $T_{k_1} \dots T_{k_i}$  periódusban a hálózat node-jain a lokális detekció, a BS-en a teljes megfigyelt területre vonatkozó esemény detekció fut.
- A  $T$  időperiódus leteltét követően a szabad paramétereket újra meghatározza a BS a pontosság fenntartásának érdekében. Ehhez a teljes mérési anyagot továbbítják egyesítve a node-ok a BS-re egy energiahatékony protokoll használatával, ami lehet az előző fejezetben ismertetett WSN csomagütemezési protokoll.

Az egyes fázisok ismétlési gyakoriságát és hosszát a következőkben ismertetett körülmények befolyásolják. Az optimális döntési paraméterek ismételt meghatározására szükség van, ha amennyiben mérési idősor tanult paraméterei drasztikusan megváltoznak, és erről tudomást szerzünk. A predikált idősor és a mért idősor különbségének jelentős eltérése erre a megváltozásra utalhat.

Az optimális klaszterek meghatározását statikus hálózat topológia esetén, illetve amennyiben a szenzorok hatóköre, vagy típusa nem változik, akkor nem kell ismételni. Egyéb esetekben a változás mértékétől függően van szükség az újratanításra.

Az adatgyűjtést pedig a szenzorok memóriának méretéhez kell igazítani, azaz a legtöbb mérést végző legkisebb memóriával rendelkező szenzorhoz igazítva érdemes a mérési értékek gyűjtését elvégezni.

### 3.6. A módszer komponenseinek párhuzamos futtathatósága

A bemutatott esemény detekciós módszer egyes komponenseihez tartozó algoritmus, valamint az optimális paraméterek meghatározásának lépései végrehajthatóak szimultán is. Ebben a szakaszban a párhuzamos futtathatóság kérdéseit és a megoldását vizsgálom.

Az algoritmusok adaptációja több sokprocesszoros platform esetén lehetséges. Ennek oka, hogy a módszer komponenseiben futó algoritmusok SIMD (Single Instruction Multiple Data, azaz azonos műveletet kell több adaton végrehajtani) típusú utasítások sorozataként épülnek fel. Továbbá a WSN hálózat egyes elemeinek méréseit a hálózat node-jain részben fel lehet dolgozni. Ez a számítási módszer megfelel az elosztott számítás definíciójának [81].

Az egyes algoritmusok párhuzamos implementációinak ismertetése előtt röviden felsorolom a potenciális sokprocesszoros architektúrákat, amelyeken az módszerek futtatathatók:

- GPGPU (General Purpose Graphical Processing Unit). A modern GPU chipek egy-két ezer processzormagot tartalmaznak, több architektúra szerint, amely magok csökkentett utasításkészlettel kicsi lokális memóriával rendelkeznek, ugyanakkor alkalmasak térbeli információk hatékony feldolgozására az egyes magok közötti osztott memóriát kihasználva. Különösen hatékony olyan műveletek végrehajtása, ahol azonos utasításokat szükséges különböző adatokon végrehajtani. A programozás a modern GPGPU-k esetén történhet OpenCL, vagy az nVidia platform esetén CUDA környezetben [82].
- FPGA-k esetén szintén lehetséges olyan áramköri blokkok kialakítása, amelyek ugyanazt a műveletsort (vagy eltérő műveletsort) párhuzamosan végrehajtják a beérkező adatokon. Továbbá lehetséges olyan műveleti pipeline kialakítása is, amely a hatékonyabb adatfeldolgozást és paraméter optimalizálást tesz lehetővé [83].

Az általam kidolgozott esemény detekciós algoritmust az alábbi két működési fázisra lehet felbontani:

1. Lokális adatgyűjtés és outlier detekció a WSN hálózat node-jain.
2. Klaszterezés alapú esemény detekció a beérkezett lokális döntések paramétereit felhasználva.

A mérések kiértékeléséhez szükséges lépések kiegészülnek az optimális döntési paraméterek meghatározásának lépéseivel, amelyek a következők:

3. A lokális detekció döntési paramétereinek meghatározása a BS-en.
4. A lokális detektor adaptálása, amely a BS-en történik.
5. Optimális klaszterek és klasztersugarak meghatározása a döntés pontosságának maximalizálása érdekében.

A továbbiakban az egyes fázisokban futó algoritmusokat vizsgálom és javaslatot teszek az egyes módszerek párhuzamos futtatására, illetve elemzem az elérhető sebességnövekedést.

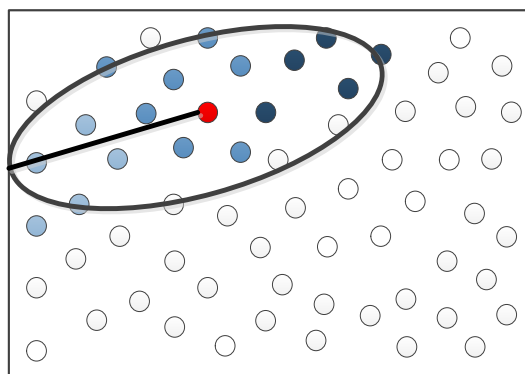
### 3.6.1. Lokális, kis komplexitási outlier detekciós algoritmus

A detekció első fázisa, ami a predikció alapú outlier detekció elosztott módon fut a WSN node-jain. Minden egyes node saját maga határozza meg, hogy mely mérési értékek lehetnek outlier értékek, a mérést követően azonnal. Ennek köszönhetően az összes node a saját új mérésének kiértékelését a többi node-dal egyidejűleg végzi, azaz ez a módszer már további lehetőséget nem biztosít a párhuzamosíthatóságra. Elméletben lehetséges volna, hogy az egyes node-ok méréseit a komolyabb számítási teljesítménnyel rendelkező BS dolgozza fel, azonban az adattovábbítás okozta késleltetést, illetve megnövekedett energiafelhasználást nem kompenzálja az elérhető sebességnövekedés [81].

### 3.6.2. Klaszterezés alapú esemény detekció

A detekció második fázisa, a klaszterezés során az egyes szenzoroktól beérkezett adatok egymástól függetlenül kiértékelhetőek. A kiértékelés során fel kell használni a szomszédos, azonos klaszterben található node-ok mérési értékeit, illetve outlier információit. A párhuzamosított döntés során olyan algoritmusra van szükség, amely a térben elhelyezhető információkat az elhelyezkedés alapján értékeli ki. (A döntés elvét a 3.11. ábra szemlélteti.)

A párhuzamosítás során kézenfekvő egy olyan architektúra választása, ahol azonos műveleteket eltérő adatokon egyidejűleg végre lehet hajtani, ezért az GPGPU-kban használatos sokprocesszoros architektúrák használata egy lehetőség. Az egyes node-ok kiértékelését egy-egy processzor hajtja végre, így a teljes mérési terület eseményeinek detekciója szimultán végrehajtható.



3.11. ábra Egyetlen mérési pont (piros) körül levő klaszter mérési pontjainak információit kell felhasználni a döntéshez. Minden egyes mérési pontra ezt külön-külön kell vizsgálni

Az elosztott outlier detekciónak és párhuzamosított esemény detekciónak köszönhetően az esemény detekciós algoritmus futtatásában a legnagyobb késleltetés a rádiós tevékenységek okozzák, nevezetesen az esemény bekövetkezése és az információ BS-re juttatása.

Az esemény detekciós algoritmus egy továbbfejlesztési iránya a hálózattól érkező flag információ, mint bináris kép feldolgozása szegmentálással. Ez esetben az optimális klaszterek meghatározásának problémájával már kell foglalkozni, továbbá elérhetőek olyan szegmentáló algoritmusok, amelyeket kifejezetten GPGPU-s környezetre terveztek és implementáltak [84].

### **3.6.3. A lokális detekció döntési paramétereinek meghatározása és a detektor adaptálása**

Minden egyes node-ra ugyanazon lépések segítségével történik a prediktor beállítása, miszerint a node specifikus tanulóhalmaz segítségével az egyes node-ok tanítása párhuzamosan történhet. Ennek megfelelően ezek a kiértékelési lépések egymástól függetlenül végrehajthatók. Az algoritmus szimultán futtatására ezáltal tetszőleges párhuzamos architektúra megfelel, mivel a node-ok számának megfelelő számú adatokon kell ugyanazokat az ismétlődő lépéseket végrehajtani. Ugyanez igaz a predikció alapú detekció optimális küszöbparaméterének meghatározása esetén is, ugyanis az egyes küszöbértékek a node-okra jellemző adatok felhasználásával egymástól függetlenül párhuzamosan kiszámíthatók.

### **3.6.4. Optimális klaszterek meghatározása az egyes node-okhoz**

Egy szimulált esemény esetén minden egyes node-hoz meg lehet határozni a hozzátartozó optimális klasztert.

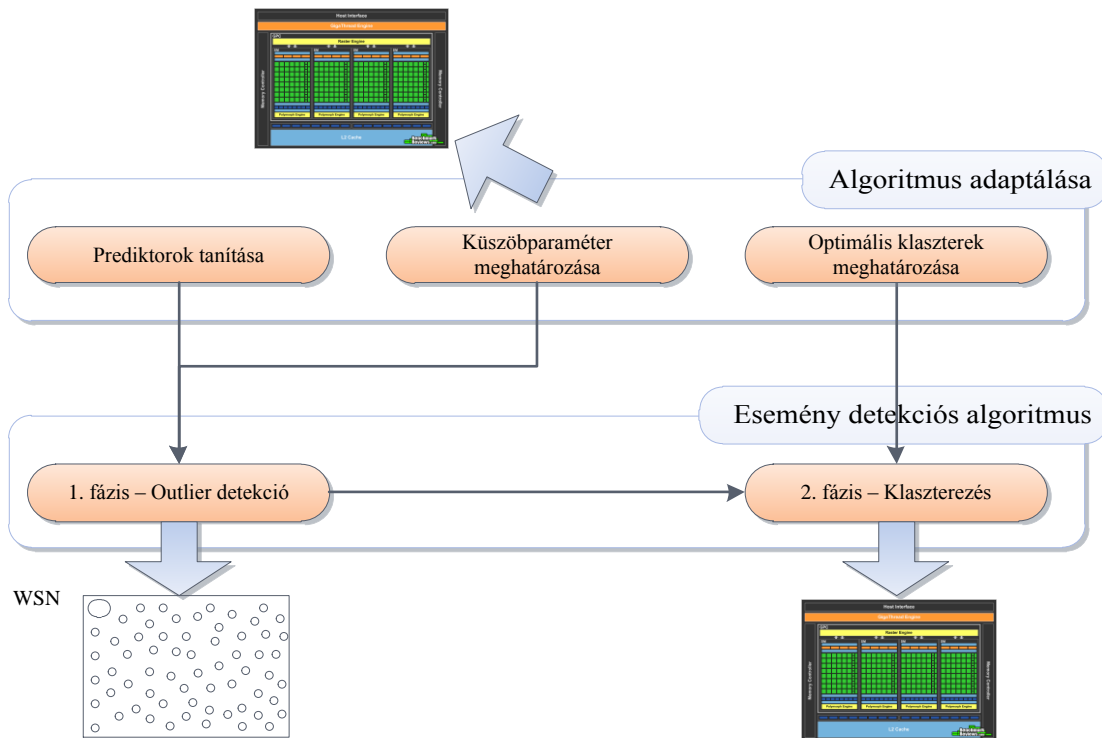
Hasonlóan a második fázisban működő klaszterezés alapú esemény detekciós algoritmus párhuzamos futtatási módszeréhez, az optimális klaszterek meghatározásához elégségesek az egyes node-okhoz tartozó bemeneti adatok és a node-ok pozíciójára vonatkozó adatok. Amennyiben ezek az információk rendelkezésre állnak az adott node körüli klaszter meghatározása lehetséges a 3.9. ábra által bemutatott algoritmus szerint. Mivel az összes node-hoz egymás eredményétől független módon meghatározhatóak az optimális klaszterek, ezért a művelet párhuzamosan is futtatható az arra alkalmas GPGPU architektúrán [84].

### **3.6.5. A párhuzamosíthatóság összefoglalása**

Az előző szakaszokban ismertettem, hogy milyen módon lehetséges a WSN hálózatokra tervezett esemény detekciós módszer komponenseinek párhuzamos architektúrán történő futtatása. A leírt módszereknek köszönhetően a WSN hálózatban található node-ok számától független az esemény detekciós algoritmus futtatása mind a kiértékelés, mind az optimális paraméterek meghatározása szempontjából. A hálózat méretének növekedtével a rádiós kommunikáció okozta késletetés és terhelés okoz egyre növekvő lefutási időt.

A klaszterek kiértékelési ideje függ a klaszterek számától, azonban ez a lépés párhuzamosítható. Amennyiben a node-ok száma meghaladja a kiértékelő processzormagok számát, akkor több ütemben történik a klaszterek kiértékelése, tehát a gyakorlatban némileg elmarad a számítási sebesség az elméleti eredményektől.

A párhuzamosíthatóság lehetőségeit a 3.12. ábra foglalja össze, ahol a csoportosítva jelöltem az esemény detekciós algoritmus két lépését, illetve az adaptáció három lépését.



3.12. ábra Az algoritmus párhuzamos futtathatóságának összefoglalása. Az outlier detekció az első fázisban elosztottan történik, míg a többi lépés esetén meghatározható egy a GPGPU-n futtatáshoz szükséges párhuzamosítás, amely során az egyes számítási feladatok függetlenül, szimultán végrehajthatók.



### 3.7. Az új módszer eredményei, teljesítőképessége

Ebben a szakaszban a bemutatott esemény detekciós módszer teljesítőképességét vizsgálom. Elsőként a teljesítményt jellemző mérőszámokat definiálom, majd a teszteléshez használt környezetet ismertetem, végül pedig a teszteredményeket mutatom be.

#### 3.7.1. Eredmények mérőszámai

A tesztek során az alábbi mérőszámokkal jellemeztem a vizsgált módszerek hatékonyságát:

- *TP* – true positive – Helyesen detektált események száma.
- *FP* – false positive – Olyan találatok száma, amelyek valójában nem események.
- *FN* – false negative – Nem detektált események száma.
- *DR* – detection rate: a *TP* és az összes esemény hányadosa. Implicit módon tartalmazza az *FN* és összes esemény hányadosát.
- *FPR* – false positive rate – az *FP* és az összes esemény hányadosa.

Az optimális algoritmus a *DR*-t maximalizálja, miközben az *FPR* értéket minimális marad. A leírt mérőszámokból az *F* mérték definiálható, amely mérték a 0 és 1 közé esik. Ez a hibák számát és a detekciós pontosságot egyaránt jellemzi:

$$F = \frac{2RP}{R + P}, R = \frac{TP}{TP + FN}, P = \frac{TP}{TP + FP}. \quad (3.38)$$

Az energiahatékonyságot a rádiós forgalmazások számával és átküldött bájtok számával jellemzem, amelyhez a Berkeley által fejlesztett Mica2 rendszer paramétereit használom a 3.1. táblázat szerint. A táblázatban feltüntettem a tipikus teljesítmény- és energiaértékeket [31]. Fontos megemlíteni, hogy a vizsgálatok során figyelembe vettem a rádió inicializálásánál felmerülő energiafogyasztást is. Az egyes kommunikációknál különböző hosszú adatsomagokat forgalmaz a hálózat, melyek tipikus hosszát a 3.2. táblázat összegzi.

3.1. táblázat Idő, energia és teljesítmény értékek a referencia Mica2 hálózatban

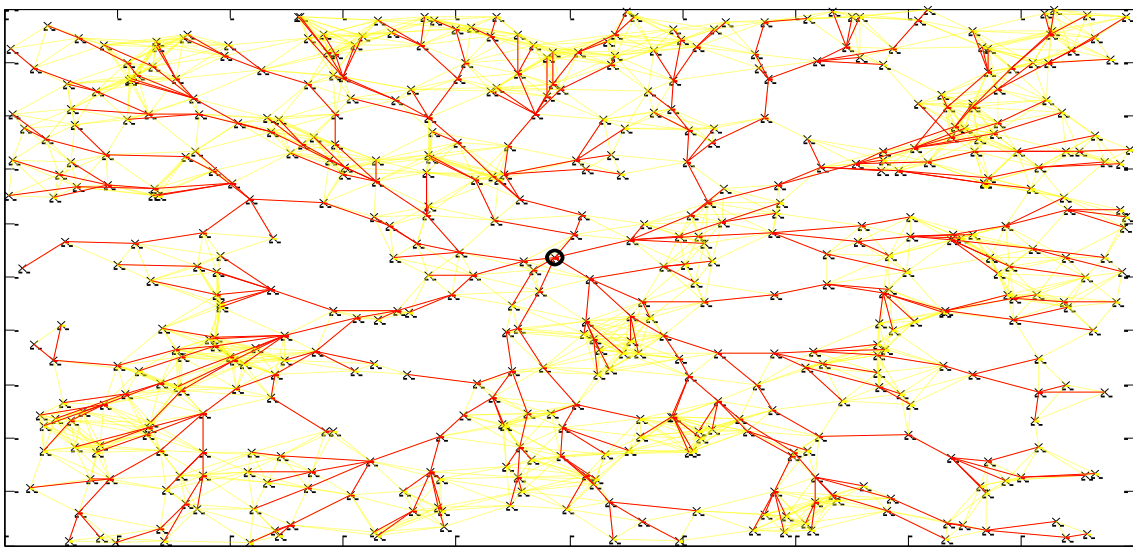
Működési fázis	Időtartam	Teljesítmény	Energia
Alvás	-	90 $\mu$ W	-
Rádióinicializálás	0,35 ms	18 mW	6,3 $\mu$ J
Rádió bekapcsolása	1,5 ms	3 mW	4,5 $\mu$ J
RX/TX állapot váltás	0,25 ms	45 mW	11,3 $\mu$ J
1 bájt fogadása	0,42 ms	45 mW	18,7 $\mu$ J
1 bájt küldése	0,42 ms	60 mW	24,9 $\mu$ J

3.2. táblázat Tipikus adatsomag felépítése és a részek hossza

<b>Hossz (bájtban)</b>	
Preambulum	8
Szinkronizáció	0
Fejléc	6
Lábléc	2
Adathossz	26
<b>Összesen</b>	<b>42</b>

### 3.7.2. Szimulációs tesztek

Az esemény detekciós módszert és annak komponenseit teszteltem szimulációkkal Matlab környezetben. A szimulációkhoz véletlen sorsolt adatokat és valós mérési értékeket egyaránt felhasználtam. Ezekhez a tesztekhez a hálózati topológiát és routing-ot véletlen módon generáltam a node-ok számának meghatározásával, téglalap alakú területen, egyenletes eloszlással. Egy tipikus mérési elrendezést a 3.13. ábra mutat be. A node-ok sűrűn helyezkednek el, annak érdekében, hogy egy kiváltott esemény hatása több mérési ponton is jelentkezzen.



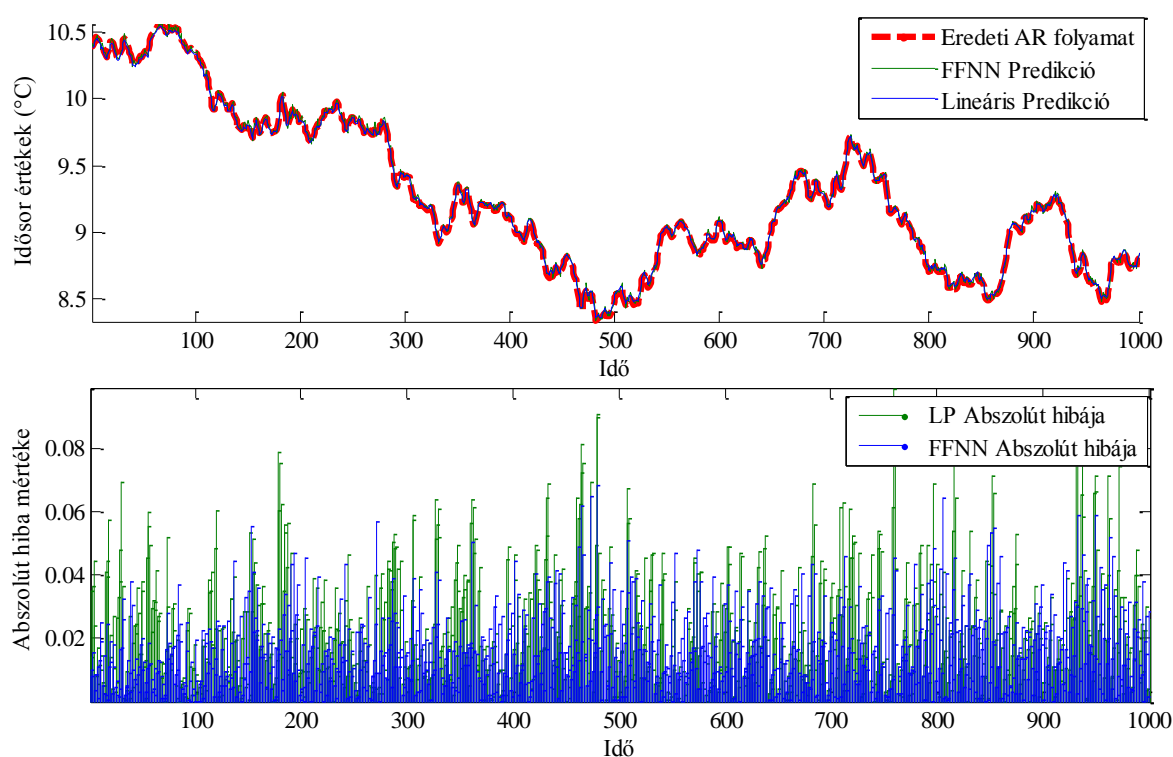
3.13. ábra Tipikus sorsolt topológia a Matlab szimulációkhoz. Az ábrán látható a útvonalválasztási fa (piros vonal), rádiós szomszédság (sárga vonal) a BS (o) és a node-ok (x).

A szimulációs tesztek során a nem idősorként generált adatsorok normál eloszlással sorsolt értékek, amelyek közé mesterségesen helyeztem el outlier-re, vagy eseményre utaló mérési értékeket. Ez az eredeti érték megváltoztatását jelenti különböző kiugrasztási tényezővel ( $t$ ), ahol a tényező jelenti annak a mértékét, hogy az eredeti  $x_j$  érték mennyire változott meg:

$$\text{Outlier}(x_j) = x_j + (x_j \cdot t \cdot (-1)^j) \quad (3.39)$$

A szintetikus idősorokat AR folyamat generálja, amelyben az outlier-eket az előzőekhez hasonlóan helyeztem el a node-ok elhelyezkedését is figyelembe véve. Ennek segítségével lehetséges a node-ok közötti térbeli korreláció szimulálása. A legjobb összehasonlítás érdekében az egyes vizsgált módszerek paramétereit változtatva a szimulációkat többször megismételtem a különböző paraméterekkel, majd az eredményeket átlagoltam.

Elsőként megvizsgáltam külön a bemutatott predikció alapú outlier detekció pontosságát. Az előzetes várakozásoknak megfelelően jól teljesített az idősorok esetén a klasszikus outlier detekciós módszerekhez viszonyítva. A predikció pontosságát a 3.14. ábra szemlélteti. Megfigyelhető, hogy az FFNN alapú prediktor pontosabb, mint a lineáris predikció AR folyamat esetén is. Általánosabb idősoroknál az FFNN módszer használatával pontosabb eredményt lehet elérni.

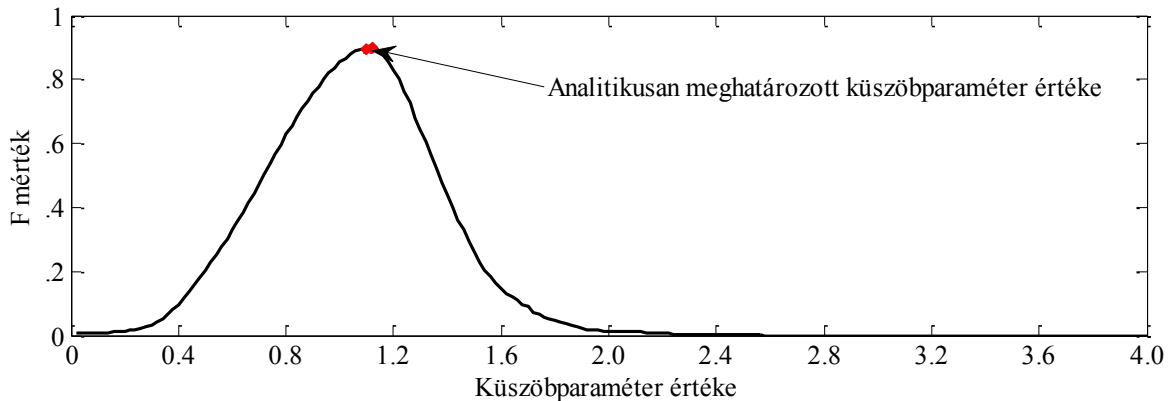


3.14. ábra A lineáris és nemlineáris prediktor képességeinek összevetése

A lineáris predikciót alkalmazva meghatározható az optimális küszöbérték, amely alkalmazásával lehetséges a detekciós hibavalószínűség minimalizálása. Megvizsgáltam, hogy az analitikus módon megállapított küszöbérték használatakor valóban optimális döntést lehet-e elérni a lineáris predikcióval végrehajtott outlier detekció esetén. Az eredményt a 3.15. ábra illusztrálja. A piros vonallal jelzett pont az analitikusan meghatározott küszöbértékhez tartozó  $F$  mérték, amely a tesztek során minden küszöbértékre meghatározott  $F$  mértékek maximumával egyenlő.

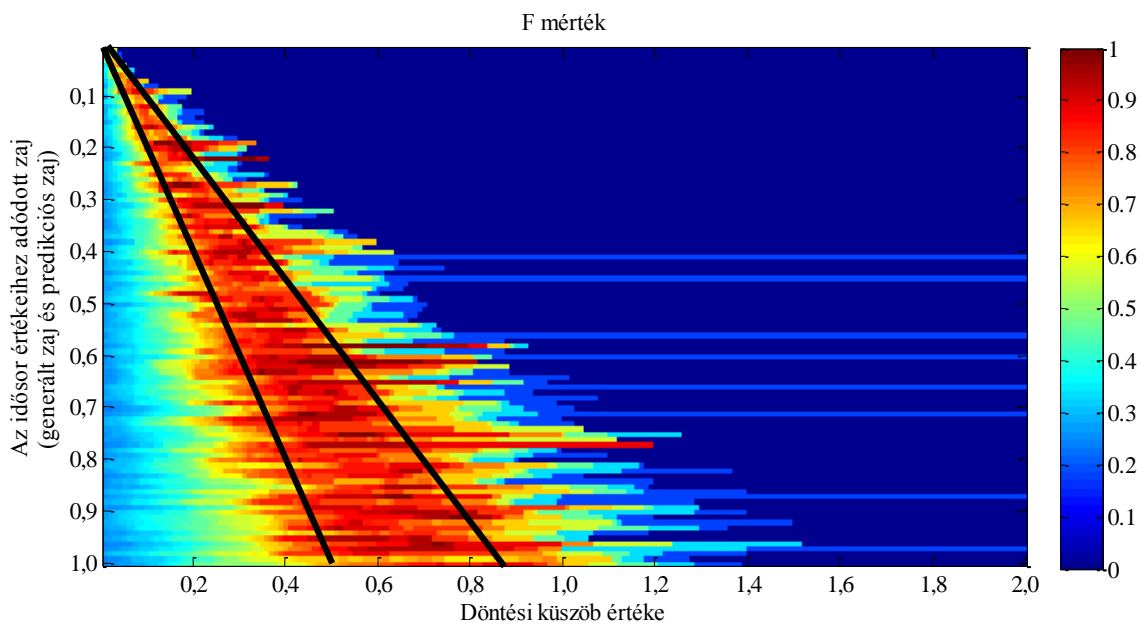
Nemlineáris prediktor alkalmazásakor nem AR folyamatot feltételezünk az idősor modelljében. Ekkor az optimális küszöbparaméter meghatározása még nyitott kérdés, ugyanakkor a generáló folyamat paramétereinek változtatásával, illetve a hozzáadott zaj paramétereinek változtatásával folytattam

szimulációkat. Az  $F$  mérték, az idősorban megjelenő zaj valamint a küszöbparaméter viszonyát a 3.16. ábra mutatja be.



3.15. ábra Az analitikusan meghatározott küszöbparaméter vizsgálata szimulációkkal

Megfigyelhető, hogy a maximális eredményhez kapcsolódó küszöbérték – hasonlóan a lineáris predikciónál analitikusan meghatározott értékhez – egy sáv tartományon belül található. Analitikus eredmény hiányában lehetséges egy adaptív algoritmus alkalmazása, amely az egyes idősorok paramétereit felhasználva megbecsüli az alkalmas küszöbparaméter értéket.



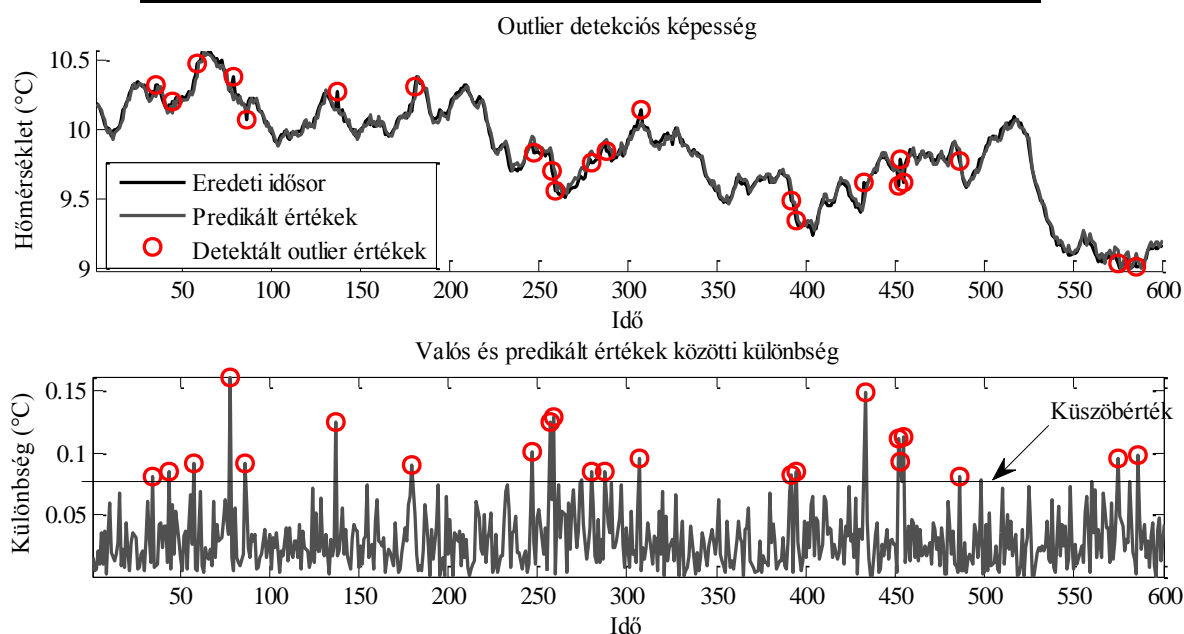
3.16. ábra Döntési küszöbérték vizsgálata az FFNN alapú predikció esetén

A 3.3. táblázat összefoglalja a létező klasszikus outlier detekciós módszerek és a predikciós alapú módszer detekciós képességét generált idősorok és abban mesterségesen elhelyezett outlier-ek esetén. A predikció alapú módszer esetében a téves riasztások száma jóval alacsonyabb, mint a klasszikus modell alapú algoritmusoknál. (A táblázatban fel nem tüntetett módszerek teljesítőképessége a Grubbs módszer képességeit nem haladja meg.) Ugyanakkor a hagyományos módszerek alkalmazhatóak olyan adatok esetén is, amelyek nem idősorba rendezettek.

Egy konkrét idősor esetén az optimális küszöbparaméter használatával végrehajtott detekciót a 3.17. ábra mutatja be. Az outlier értékek mindegyikét detektálja a módszer.

3.3. táblázat A predikációs alapú módszerek összehasonlítása a legjobb klasszikus módszerekkel, mesterségesen generált idősorok esetén

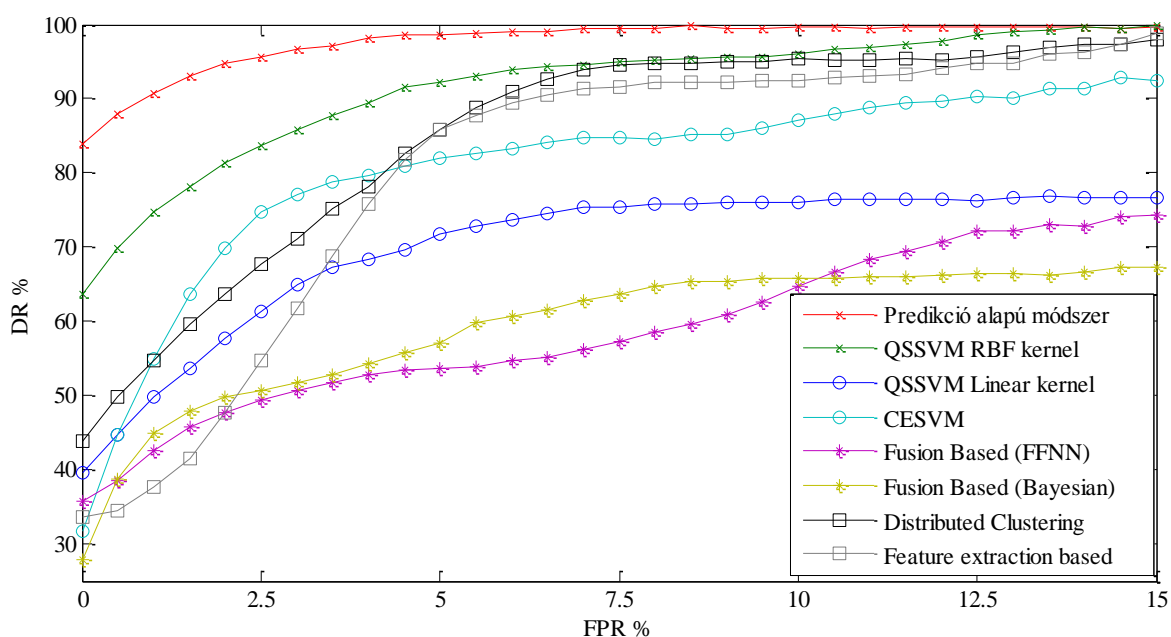
	FN %	FP %	F mérték
Grubbs módszer	0,6 %	56,8 %	0,44
Naiv Bayesi döntés	3,1 %	20,1 %	0,83
Lineáris prediktor	1,0 %	2,4 %	0,95
FFNN prediktor	0,9 %	1,0 %	0,98
RBF prediktor	0,9 %	2,0 %	0,97



3.17. ábra Outlier detekció valós idősoron

A predikció alapú outlier detekcióra alapozom az esemény detekciós módszert. A teljes módszert vizsgáltam a szimulációs környezetben véletlen generált adatsoron és abban mesterségesen elhelyezett eseményeken.

A tényleges idősorokon végrehajtott tesztek esetén azt tapasztaljuk, hogy a legjobb eredményeket az új módszer szolgáltatja. A 3.18. ábra, a ROC ábra, amely a *DR* és a *FPR* viszonyát mutatja be. Minél nagyobb találati arányra törekszünk, annál több olyan találat is van, amely nem tekinthető eseménynek, tehát az optimális képességekkel rendelkező detekciós algoritmus konstans „1” értéket venne fel. Az idősorokra lefuttatott tesztek esetén a kidolgozott hibrid eljárás túlteljesíti a már létező módszereket. A bemutatott és tesztelt módszerek többségében elérik a maximális *DR* értéket, ugyanakkor jóval több *FP* riasztást is generálnak a *DR* érték növekedésével.



3.18. ábra Esemény detekciós módszer teljesítménye meglévő módszerekkel szemben. Az optimális algoritmus a konstans 100% DR értéket veszi fel, függetlenül az FPR értéktől.

### 3.7.3. Valós szenzorhálózatban végrehajtott tesztek

A bemutatott algoritmust valós szenzorhálózat segítségével teszteltem, azok méréseit felhasználva. A szenzorhálózat egy beltéri alkalmazás, amely Berkeley Mica2 típusú [31] node-okból épül fel. A hálózat node-jait több szobában helyeztem el egymástól nem rendezett módon – hozzávetőlegesen 1-1,5 méter távolságban – biztosítva ezzel a mérések közötti térbeli korreláltságot (tehát egyetlen eseményt mindig érzékel több mérési pont is). A tesztek több azonos számú node-ot tartalmazó, de különböző topológiára; valamint több különböző számú node-ot tartalmazó topológiára egyaránt elvégeztem. Minden node képes fényerősséget, hőmérsékletet és akusztikus információt mérni. A kidolgozott detekciós protokollnak megfelelően a hálózat node-jain és a bázisállomáson futtatott algoritmusok szabad paramétereit a tanítási fázis során beállítottam, majd a tesztek az azt követően végeztem el.

A 3.4. táblázatban a bemutatott esemény detekciós módszer fontosabb tulajdonságait a fogalom össze, feltüntetve a referenciaként használt további módszerek tulajdonságait is. Ezen adatok közül a legfontosabb a valós környezetben elért detekciós arány, amelyet a DR oszlop tartalmaz. Látható, hogy valós körülmények között a módszer kimagasló detekciós arányt biztosít.

A táblázatban a következő paramétereket tüntettem fel az egyes módszerek jellemzéséhez:

- DR – Detekciós arány, azaz az azonosított események hány százalékát detektálja
- Elosztott? – A módszer centralizált, vagy elosztott algoritmus-e?
- Energiahatékony? – A tervezési szempontok között szerepelt-e az energiahatékonyság?
- Térben/Időben korrelált adatok? – Az algoritmus figyelembe veszi-e az adathalmaz térbeli/időbeli korreláltságát

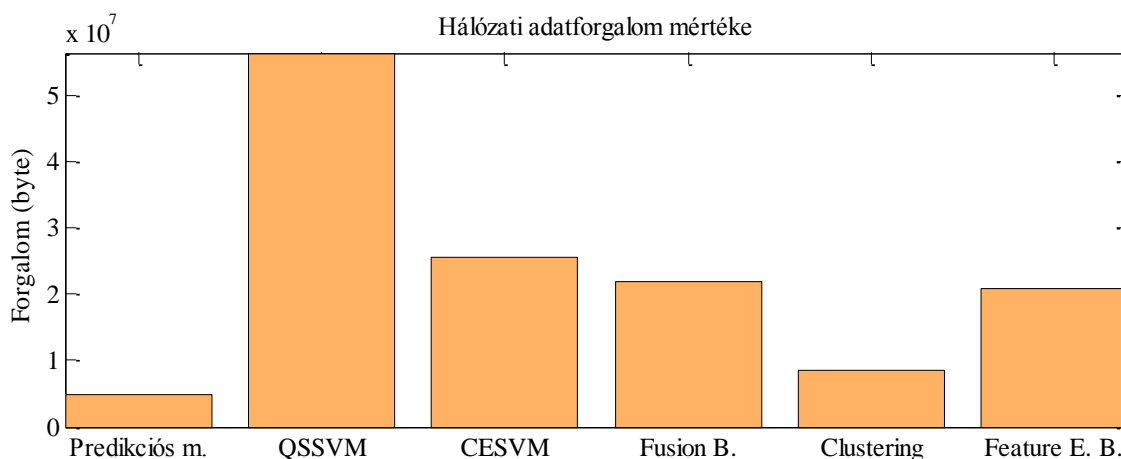
- Multiszenzoros detekció? – A módszer figyelembe veszi-e a detekcióhoz egy node több szenzorának mérési értékét, illetve az egy mérési ponton keletkezett különböző szenzorok mérései közötti korrelációt?

3.4. táblázat Összehasonlított esemény detekciós módszerek tulajdonságainak összefoglalása

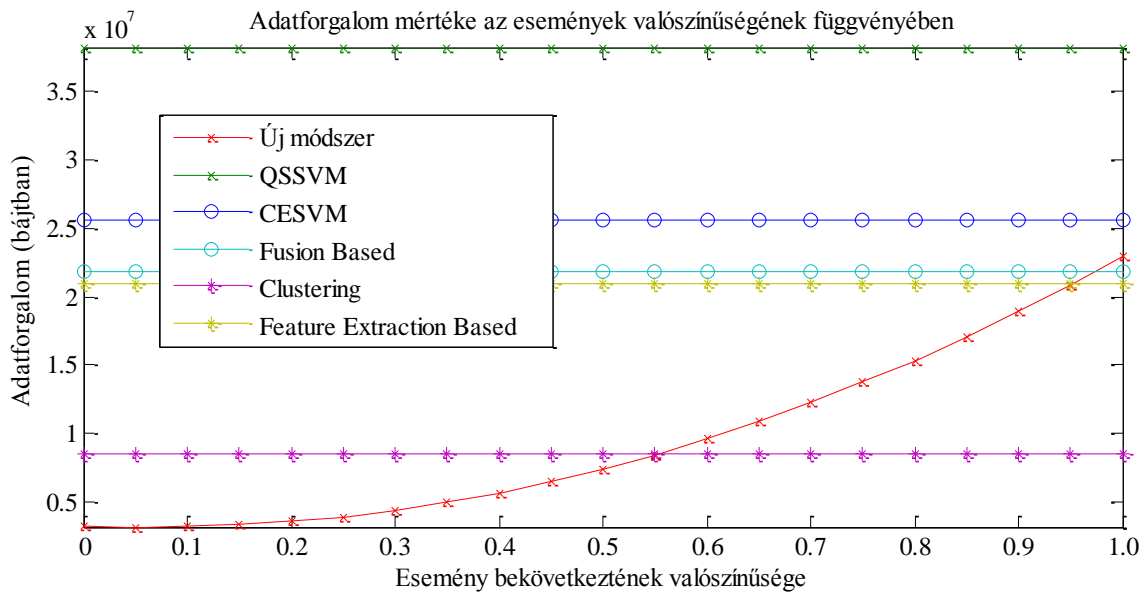
Módszer	DR	Elosztott?	Energia hatékony?	Térben korrelált adatok?	Időben korrelált adatok?	Multi-szenzoros detekció?
Predikció alapú	95	I	I	I	I	N
QSSVM RBF kernellel	85	I/N	N	I	részben	I
QSSVM Linear kernel	60	I/N	N	I	részben	I
CESSVM	69	I	részben	I	részben	I
Fusion Based (FFNN)	75	I	részben	I	N	I
Fusion Based (Bayesian)	78	I	részben	I	N	I
Distributed Clustering	83	I	I	I	N	N
Feature Extraction Based	79	I	részben	I	N	N,

Vizsgáltam a kidolgozott esemény detekciós módszer energiahatékonyosságát a hálózat adatforgalmának analízisével. A tesztek esetén, az eredmények meghatározásához a detekciós módszerekkel összefüggésbe hozható rádiózások számát vettem figyelembe. A WSN hálózat szervezési és egyéb csomagjait nem számoltam, mivel az minden módszer esetén azonos.

Az 3.19. ábra az egyes módszerek alkalmazása során a hálózat adatforgalmát mutatja be a korábban leírt topológia esetén. Egyértelmű előnye az új módszernek abból fakad, hogy i) az adatbeküldés esemény vezérelt, illetve ii) a csomagtovábbítás aggregált módon történik. Az előforduló események számától is függ az elküldött csomagok száma és mérete. Vizsgáltam azt is, hogy miként változnak ezek az értékek, amennyiben az egyes node-okon az outlier értékek megjelenésének valószínűségét változtatjuk. Ennek eredményét a 3.20. ábra mutatja be.



3.19. ábra Egyes esemény detekciós módszerek adatforgalmazási mértéke WSN hálózati alkalmazás esetén



3.20. ábra Összefüggés az események bekövetkezési valószínűsége és a hálózat adatforgalma között egyes algoritmusok alkalmazásakor

Bizonyos esemény-gyakoriság felett a módszer veszít energiahatékonyságából, sőt nagyon gyakori esemény-bekövetkezések során más módszerek hatékonyabbnak bizonyulnak. Ugyanakkor a valós alkalmazások többségében az eseményeket jelentő értékek és a normál értékek hányada alacsony. Megvizsgáltam azt is, hogy miként befolyásolja a klaszterek és a hálózat mérete az adatforgalmazást. A teszt megmutatta, hogy kisebb méretű hálózatok esetén a kidolgozott módszerem energiahatékonyságából fakadó előnye csekély, illetve az adatforgalom topológia független, abból a szempontból, hogy egy node csak akkor riaszt, ha lokálisan detektált outlier értéket. Az adatbegyűjtés összes adatforgalmát pedig a begyűjtésre használt protokoll befolyásolja. Az előzőeknek megfelelően leghatékonyabban olyan nagyméretű, sokszenzoros környezetben lehet használni az algoritmust, amelyben az események relatív gyakorisága alacsony.

### 3.7.4. Elért eredmények összefoglalása

Ebben a fejezetben bemutatásra került egy outlier és egy erre épülő esemény detekciós módszer, amely WSN hálózatban használható. Ez az új módszer egy moduláris megoldás, amelyben az outlier detekció lokálisan kerül végrehajtásra a node-okon, és döntést a lokális eredmények bázisállomáson történő összegzése során hoz. A kidolgozott séma egyik előnye, hogy felhasználja a szenzorok térbeli és időbeli korreláltságában levő információkat a következőképpen: a node-okon predikciós alapú algoritmus működik, amely a korábbi mérésekből becsléssel meghatározott érték, és a tényleges mérési érték különbségéből hozza meg a döntést. A bázisállomáson végzett összesítéshez klaszterezést alkalmazok, amely után a klaszterekben az outlier-t jelentő mérési pontok arányából lehet következtetni esemény jelenlétére.

A módszer tipikus alkalmazási területe egy megfigyelő/érezkelő hálózat, amely riasztási eseményt generál amennyiben szokatlan értékeket mér. A szokatlan érték behatolásnak, vagy egyéb zavaró



körülménynek lehet a következménye. A többi mérési adat továbbítása másodlagos feladat, így kizárólag az események bekövetkeztekor történik azonnali csomagküldés. Az energiahatékonyságot ennek megfelelően a node-okon történő lényegkiemelő algoritmus teszi lehetővé.

A bemutatott módszert kiterjedt teljesítőképesség-vizsgálatnak vettem alá, amely megmutatta, hogy a teljes esemény detekciós séma is megfelelően jó eredményt nyújt. Az analízisekből látható, hogy a kidolgozott módszer egyaránt pontos és energiahatékony. A pontosságot az idősor értékek korreláltságban rejlő információk felhasználásával, míg az energiahatékonyságot a rádiós csomagok számának csökkentésével, illetve a rádiós tevékenységek összevonásával értem el.

*Negyedik fejezet*

## ÖSSZEFOGLALÁS

**4.1. Új tudományos eredmények**

Ebben a fejezetben összefoglalom a disszertáció eredményeit, kimondva az egyes téziseket. A két téziscsoportomhoz tartozó eredmények külön alfejezetben kerülnek ismertetésre.

**4.1.1. Ütemezési algoritmusok**

Új, polinom idejű algoritmusokat vezettem be erőforrás ütemezésre, amelyekkel előírt peremfeltételek mellett optimális ütemezést lehet létrehozni. Az ütemezési módszer felhasználható egy számítási egység kapacitásának ütemezésére, ahol az egyes feladatok súlyozottak, illetve a határidőn túli ütemezés elkerülendő. Ilyen környezet például pénzügyi kereskedelmi rendszerek esetén két kereskedési időszak közötti portfólió-optimalizálás és –elemzés. Ezeket az algoritmusokat kiterjesztettem vezeték nélküli ad- hoc és szenzorhálózatok csomagütemezésének az optimalizálására.

*Létrehoztam egy polinom idejű heurisztikus ütemező algoritmust, amely az általános célfüggvények mellett az összesített súlyozott késleltetést is minimalizálja. A létrehozott LWPF és PLWPF algoritmusok a létező súlyozást figyelembe vevő algoritmusok variánsai, amelyek a TWT problémák esetén jól teljesítenek. (2.4.1.1 szakasz)* Az ütemezési probléma egy változata, amikor a rendelkezésre álló feldolgozó kapacitás nem elegendő. Így a bejövő feladatokat a megadott határidőkön belül nem lehet megoldani. A feladat megkívánja a modellt kiterjesztését úgy, hogy minden  $j$  feladathoz rendelünk egy súlyértéket. Ekkor a cél az összegzett, súlyozott késleltetés (TWT) minimalizálása. A Largest Weighted Process First (LWPF) módszer az ütemezéshez a feladatokat a súly alapján rendezi sorba, és illeszti bele az ütemezési mátrixba a kapacitásnak figyelembe vételével. A módszer lényegesen jobb eredményt ér el a létező EDD és WSPT heurisztikákhoz képest. A Perturbed LWPF (PLWPF) módszer továbbfejlesztése az előző heurisztikának, mivel annak az eredménye a TWT-t tekintve nem éri el a HNN alapú megoldás eredményét. A LPWPF módszer analóg a PSHNN módszerhez. A heurisztika a létrehozott ütemezési mátrixban az egy adott időszakhoz tartozó ütemezést véletlenszerűen megváltoztatja. Ez a lépés a felhasznált kapacitást nem befolyásolja, azonban az előírt feladatmennyiségekre vonatkozó kitétel sérülhet. Számos létrehozott, véletlenszerűen megváltoztatott és korrigált ütemezési mátrix közül a legjobb TWT-vel rendelkezőt kiválasztva olyan ütemezést kapunk eredményül, amely közel olyan jó eredményt nyújt, mint a kvadratikusan optimalizált megoldó heurisztikával elért eredmény. Ez a módszer ugyanakkor rugalmatlan a peremfeltételek változásával szemben.

*Az ütemezési feladatot visszavezettem kvadratikusan optimalizálási feladatra. A problémát az erőforrás egyenletes terhelése mellett oldom meg, miközben az előírt feladatmennyiségre és*

*időkorlátra vonatkozó megszorításokat figyelembe veszem. Az eredményeket részletes szimulációkkal ellenőriztem. (2.4.2.1 és 2.4.2.2 szakaszok)* Az előírt feladatmennyiség teljesítése egyenletes terhelés megvalósítása mellett az egyes oszlopokban található egyesek száma az egyes oszlopok között minimális különbségű. A C ütemezési mátrixot egy sorfolytonos kiolvasással átalakítom egy c vektorba, majd az egyesített optimalizálási függvényt transzformálok úgy, hogy egy kvadratikus alak minimuma legyen a célfüggvény megoldása. A meghatározott mátrixokból és vektorokból a kvadratikus alak paramétereit összegzéssel kaptam meg. Az egyes peremfeltételeket az  $\alpha$ ,  $\gamma$ ,  $\delta_1$ ,  $\delta_2$  heurisztikus paraméterekkel priorizálom. A bemutatott módszer általánosítható, további kényszerfeltételek vezethetők be, amelyekkel más, hasonló ütemezési problémák oldhatók meg, ahogyan azt következőkben összefoglalom.

*Az ütemezési feladat megoldásának módszerét felhasználva új csomagütemezési protollokat vezettem be vezeték nélküli érzékelő hálózatban. A módszerek kezelik a WSN környezetben felmerülő további kényszerfeltételeket, és azoknak megfelelő, érvényes csomagütemezést biztosítanak. (2.4.2.3 és 2.4.2.4 szakaszok)* Megmutattam, hogy az általam ismertetett erőforrás-ütemezési módszer használható WSN hálózatokban. Két alkalmazási lehetőséget vizsgáltam és adaptáltam. Ezek közül az első egy kétszintes hierarchiával rendelkező WSN hálózat, amelyben egyenletes csomagvesztési valószínűség mellett kell megoldani a csomagütemezési problémát.

Ez a probléma közvetlenül visszavezethető az előzőleg bemutatott módszerre. A második, kiterjesztett WSN modell esetén, egy komplex hálózat teljes csomagütemezését kell megvalósítani, amely során biztosítjuk az alábbi feltételek teljesülését:

- Előírt csomagmennyiség átküldése a levél node-oknál.
- Minden egyéb node a forgalmának megfelelő mennyiségű csomagot küldjön, és csak akkor küldje, ha rendelkezésre áll. (Előidejűség megtartása.)
- Ne legyen interferencia a hálózatban, a csomagküldések garantálása érdekében.

A feltételek megtartásával szintén konstruálható egy kvadratikus optimalizálási feladat, amely a meghatározott paraméterek mellett megoldja a problémát. Az érvényes ütemezés létrehozásához a kvadratikus optimalizálás súlymátrixai és súlyvektorai a következők: az interferenciát gátló  $\mathbf{W}_F$ ,  $\mathbf{b}_F$ ; a forgalmi paramétereket betartó  $\mathbf{W}'_A$ ,  $\mathbf{b}'_A$ ,  $\mathbf{W}_E$ ,  $\mathbf{b}_E$ ; és a node-ok felesleges felébredését meggátoló  $\mathbf{W}_G$ ,  $\mathbf{b}_G$  mátrixok és súlyvektorok. Ezeket felhasználva a végső optimalizálási feladat a tetszőleges célfüggvényhez tartozó súlyokkal ( $\mathbf{W}_H$ ,  $\mathbf{b}_H$ ) együttesen a  $\mathbf{W} = \alpha \mathbf{W}'_A + \varepsilon \mathbf{W}_E + \zeta \mathbf{W}_F + \zeta \mathbf{W}_G + \mathbf{W}_H$  és  $\mathbf{b} = \alpha \mathbf{b}_A + \varepsilon \mathbf{b}_E + \zeta \mathbf{b}_F + \zeta \mathbf{b}_G + \mathbf{b}_H$  alakban írható fel, ahol az összegben szereplő tényezőket az  $\alpha$ ,  $\varepsilon$ ,  $\zeta$ ,  $\zeta$  heurisztikus paraméterek súlyozzák. Ezek közül a forgalmat szabályozó súlymátrixokat határoztam meg.

*A kvadratikus optimalizálási feladatokat Hopfield neurális hálózatot alkalmazva oldottam meg. Kidolgoztam egy eljárást a HNN által szolgáltatott eredmény pontosságának növelésére, egyrészt a kiindulási állapotok megválasztásával, másrészt a heurisztikus paraméterek variálásával. (2.4.3 szakasz)* A bemutatott kvadratikus optimalizálási feladatra transzformált ütemezési feladatokat

polinom időben, a Hopfield neurális hálózatoknál használt rekurziós formula segítségével oldottam meg.

A HNN alapú módszer esetén több különböző véletlen pontból indítjuk a szimulációt, és a legjobb eredményt tekintjük a módszer kimenetének. Az ilyen módon végrehajtott megoldásnak más megoldó heurisztikákkal szembeni előnye a nagymértékű párhuzamosíthatóság, amely az algoritmus futási idejét jelentősen lerövidíti.

A HNN megvalósítás egy javított változatát (SHNN módszer) is létrehoztam, amely akkor használható, ha a probléma megoldására létezik olyan eredmény, amelyet más heurisztika szolgáltat. Ekkor a létező megoldást fel lehet használni a HNN kiindulási pontjaként, illetve a létező eredmény véletlenszerűen módosítva lehet létrehozni több kiindulási pontot a HNN számára (PSHNN módszer). Minderre azért van szükség, mert a HNN rekurzió a gyakorlati alkalmazásokban a lokális szélsőértékekben megrekedhet. A HNN megvalósítás egy másik problémája az egyes célfüggvényeket súlyozó heurisztikus paraméterek meghatározása. Az implementáció során alkalmazott heurisztikus paraméterek hangolására használt algoritmust a második fejezetben található 2.7 ábra mutatja be.

***A HNN/PSHNN iterációk futási idejének lerövidítése érdekében meghatároztam, hogy a módszer sokprocesszoros számítási architektúrán milyen módon párhuzamosítható. A módszer akár 400 százalékkal gyorsabban hozza létre az ütemezési mátrixot, mint szekvenciális implementáció. (2.4.4 szakasz)*** A bemutatott, kvadratikus optimalizálásra használt HNN egy lehetséges heurisztika több közül, amely megoldja polinom időben a kitűzött problémát. A megfelelő eredményhez több iniciális pontból indítva, illetve a heurisztikus paraméterek számos kombinációjára megismételjük a HNN iteráció lefuttatását. Ez két különböző szintű újraitertálást jelent. Az egyes ismétlődő iterációk függetlenek, ezért párhuzamosan is futtathatóak. Felhasználva egy létező GPGPU-s implementációt [45] a következő párhuzamosítási lehetőséget javaslom, amely az nVidia Fermi architektúrán alkalmazható [46]:

- A  $\mathbf{W}$  mátrix ritkás (5-10%-a nem nulla érték), ez lehetővé teszi nagyméretű  $\mathbf{W}$  mátrixszal rendelkező HNN-ek esetén is, hogy a HNN iterációkat egy-egy stream processzor dolgozza fel, mivel a mátrix az osztott memóriában elfér.
- Több párhuzamosan feldolgozott iniciális állapotú, vagy különböző heurisztikus paraméterekkel bíró HNN-t párhuzamosan, különböző stream processzorokon futtatok.

Ennek megfelelően a Fermi sokprocesszoros architektúrán a PSHNN algoritmus futási ideje jelentősen lerövidíthető, és így megfelelő méretű  $\mathbf{W}$  mátrix esetén több nagyságrenddel jobb idő alatt kapunk eredményt. A megfogalmazásban szereplő 400%-os sebességnövekedést az nVidia 440GTX chippel lehet elérni az Intel Core i5@2.8 Ghz rendszerrel szemben.

Ez teszi lehetővé, hogy a hagyományos heurisztikáknál nemcsak jobb eredményt kapjunk, hanem azt az eredményt a gyakorlatban is elfogadható időben határozzuk meg [46], [47].

#### 4.1.2. Outlier és esemény detekció

Olyan algoritmusokat fejlesztettem ki, amelyek az idősorokban előforduló kiugró értékek és események detektálására alkalmasak alacsony tévesztési ráta mellett. A létrehozott séma WSN hálózatokban alkalmazható energiahatékony módszer csökkenti a rádiós kommunikációk számát, és olyan elosztott detekciós rendszert valósít meg, ahol a lokális detekció a node-okon történik. Gyakori alkalmazás a WSN-ek esetén a felügyeleti és monitorozó felhasználás, amikor a vizsgált célterületen bekövetkező eseményeket is detektálni kell. Több esemény detekciós algoritmus létezik, azonban ezek az energia felhasználás csökkentésére nem fókuszálnak. Az általam kidolgozott módszernek két építőkockája van, amelyek lehetővé teszik, hogy az algoritmus elosztottan a szenzor node-jain működjön, illetve a rádiós csomagküldések számának csökkentésével a hálózat élettartamát jelentősen növelje.

*Szenzoros idősorokban előforduló kiugró (outlier) értékek detektálására predikció alapú algoritmust alkalmaztam, amely a valós mérési értékek és a korábbi mérésekből becsült érték összehasonlításából hozza meg a döntést. Analitikusan meghatároztam az optimális döntéshez a küszöbparaméter értékeit, továbbá részletes szimulációkkal igazoltam a módszer hatékonyságát.*

(3.3 szakasz) A módszer a rendelkezésre álló adatok időbeli korreláltságának kihasználásán alapul a következőkben leírt módon. Egy idősor következő értékét a korábbi értékek felhasználásával megbecsüljük, és a becsült értéket a ténylegessel összevetve egy bináris döntést hozunk. A szenzorok által érzékelt idősor és az outlier folyamat modelljét felhasználva meghatározható az optimális döntéshez tartozó  $\Delta$  küszöbérték, amelyet úgy kell megállapítani, hogy akkor jelezzen outlier értéket, amikor az valóban bekövetkezett, illetve normál értéket jelezzen ellenkező esetben. A hibák valószínűségének minimalizálásával az optimális döntési paramétereket meghatározhatóak egy kényszeres szélsőérték keresési feladat során.

A kidolgozott módszerben a lineáris prediktort olyan idősor esetén alkalmaztam, amelyet lineáris autoregresszív (AR) folyamat generál. Általánosabb esetben a lineáris prediktor helyett előrecsatolt neurális hálózatot alkalmaztam (FFNN). Az összegzett abszolút hiba az FFNN-nél alacsonyabb, ami a generált nemlineáris folyamatnak a következménye. Megfelelő döntési küszöb alkalmazásával az outlier értékek közel 100%-os detekciós rátával megtalálhatóak. Ez a részben elosztott outlier detekciós módszer használható olyan rendszerek esetén, amelyekben kis számítási kapacitással rendelkező mérőegységek kerülnek telepítésre és a lokális hibákat kell kiszűrni. (Például WSN környezetben, vagy Smart Gridekben az intelligens mérőkben.)

*A predikció alapú outlier detekciós módszert felhasználva kidolgoztam egy eljárást, amely vezeték nélküli szenzorhálózatban alacsony tévedési valószínűséggel események detekciójára használható, minimális energiafelhasználás mellett, maximalizálva a hálózat élettartamát. A módszer a számításokat elosztottan végzi a hálózat node-jait felhasználva. Megmutattam az optimális klaszterek meghatározásának algoritmusát.* (3.4 szakasz) Az outlier detekciós módszerre építve létrehoztam egy esemény detekciós sémát, amely a node-ok által rögzített mérési értékek közötti térbeli korrelációban rejlő információt használja fel.

Az első fázisban, a hálózat node-jain végrehajtott outlier detekció kimeneti eredményei a bázisállomáson kerülnek feldolgozásra. A második fázisban történik a lokális döntések összegzése, amelyet klaszterezés alapú eljárással oldottam meg. Ennek segítségével i) kiszűrhetővé válnak a lokális, az egyetlen node-ot érintő mérési hibák, vagy szenzorhibák; és ii) meghatározhatóvá válnak az egy környezetben – szomszédos node-ok által érzékelt – események, az esemény sugarát is figyelembe véve. A klaszterezés alapú detekció során keressük a helyes döntéshez tartozó optimális klasztert. Ennek meghatározásához a helyes esemény detekció valószínűségét kell maximalizálni. Emiatt olyan klasztereket kell találni, ahol a klaszterben levő node-ok több mint a felénél jelentkezik pozitív outlier döntés egy esemény bekövetkeztekor. A maximalizáláshoz szükséges értékeket valós hálózatban generált események segítségével lehet megállapítani. Az optimalizációt minden node-ra és minden klaszterre el kell végezni.

***Kidolgoztam az esemény detekciós eljárás párhuzamosított változatát. Ez alkalmas a sokprocesszoros (például GPGPU) architektúrákon történő futásra, jelentősen csökkentve az algoritmus futási idejét. A párhuzamosítással elért sebességnövekedés a node-ok számával arányos. (3.6 szakasz)*** A kidolgozott esemény detekciós séma első fázisa a predikció alapú outlier detekció a hálózat node-jain elosztott módon fut. A második fázis a klaszterezés, ahol az egyes node-okhoz tartozó információk kiértékelése történik a BS-en. Az egyes node-ok kiértékelése független folyamat és párhuzamosan végrehajtható.

A tanulási és adaptációs folyamatnak az alábbi lépései vannak, amelyeket végre lehet hajtani párhuzamosítva:

- *Prediktorok adaptálása:* Minden egyes node-ra ugyanazon lépések segítségével történik a prediktor beállítása, miszerint a node specifikus tanulóhalmaz segítségével az egyes node-ok tanítása párhuzamosan történhet
- *Predikció alapú detekció optimális küszöbparaméterének meghatározása:* Az egyes küszöbértékek a node-okra jellemző adatok felhasználásával egymástól függetlenül párhuzamosan kiszámíthatóak.
- *Optimális klaszterek meghatározása az egyes node-okhoz:* Egy szimulált esemény esetén minden egyes node-hoz meg lehet határozni a hozzátartozó optimális klasztert. Ehhez elégségesek a node-okhoz tartozó bemeneti adatok és a node-ok pozíciójára vonatkozó adatok, mely információk birtokában az optimális klaszterek párhuzamosan kereshetők, amelyre a GPGPU architektúra alkalmas.

Az elosztott eljárás mintájára olyan módszerek dolgozhatók ki, amelyek használhatók például a crowdsourcing alkalmazások során is.

## 4.2. Az új eredmények felhasználási területe

Mindkét téziscsoportban sikerült olyan új módszereket és algoritmusokat kifejleszteni, amelyek

- teljesítőképesség szempontjából meghaladják a már létező módszereket;
- komplexitásukat illetően polinom időben képesek működni;
- sokprocesszoros architektúrán párhuzamosíthatók, ezzel jelentősen csökkentve a futtatási időt;
- új kényszerfeltételek mellett is közel optimális megoldás megtalálására alkalmasak.

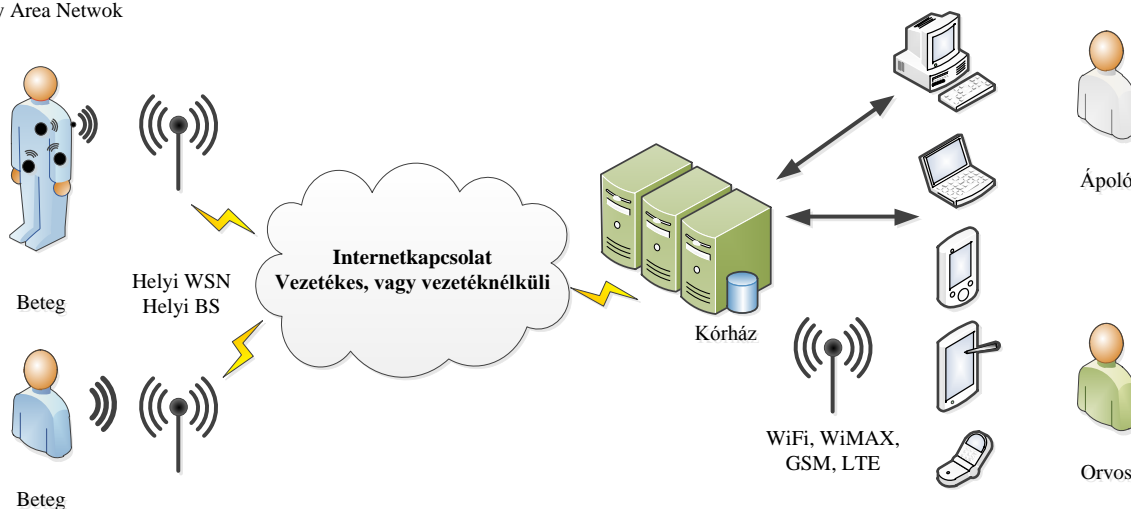
A kidolgozott módszerek monitorozó WSN hálózatok hatékony működését segítik. Az előző követelmények teljesítésével a disszertáció kitézéseit sikerült elérni. Az eredmények például monitoring rendszerekben használhatóak fel, amelyek többek között a következők lehetnek:

*Egészségügyi, környezeti monitorozás*, amely során a vizsgált területen a környezet paramétereit monitorozzák adatgyűjtés és/vagy előrejelzés céljából. Számos szenzor létezik, amelyek klíma, geológiai, vagy egyéb környezeti tulajdonságokat képesek érzékelni.

*Ipari berendezések monitorozása*, ahol az egyes berendezések állapotát monitorozzák, hasonlóan a környezeti monitorozáshoz.

*Területi megfigyelés*, amely során az adott területen előforduló jelenségeket monitorozzák, vagy a területen bekövetkező szokatlan eseményeket (például: behatolás) detektálják.

Body Area Network



4.1. ábra Intelligens páciensmonitorozás

A fenti alkalmazások során a téziseimben leírt eredmények felhasználhatóak, kiváltképpen a területi monitorozás esetében, ahol szükséges:

- az erőforrások megfelelő ütemezése a szenzorhálózat véges kapacitásainak optimális felhasználása érdekében;
- a csomagtovábbítás ütemezése, magas áteresztőképesség és energiahatékony működés érdekében;
- magas detekciós rátájú és energiahatékony esemény detekciós algoritmus, az egyes előforduló szokatlan jelenségek/események/behatolás detektálására.

Az eredményekkel elért teljesítőképességet az alábbi 4.1. táblázat foglalja össze.

4.1. táblázat A tézisek összefoglalása

<i><b>Kutatási terület</b></i>	<i><b>Teljesítőképesség és mérőszáma</b></i>		<i><b>Az eredmény alkalmazhatósága</b></i>	
			<i><b>WSN hálózatokban</b></i>	<i><b>más területen</b></i>
Erőforrás ütemezés, optimális csatorna hozzáférés	Késletetés, ütemezés összköltsége	5 % - 10 %	energiahatékonyságot igénylő alkalmazások: riasztás, lokalizáció, monitorozás, kontroll	hívásengedélyezés, processzorütemezés (például mobil eszközökben, vagy adatfeldolgozás során)
Idősor analízis, adatbányászat	Detekciós arány, energiafogyasztás	10 % - 20 %		adatbányászat: adattisztítás, krízis detekció; pénzügyi idősorokon detekció; crowdsourcing alkalmazások során; Smart Grid rendszerekben Smart Meter-ek esetén

A tézisek megoldást nyújtanak más területen is létező problémákra az optimális ütemezés, a pontos outlier detekció, a HNN algoritmus párhuzamosíthatóságának területén.



*Ötödik fejezet*

## FÜGGELÉK

**5.1. A disszertációban használt jelölések és rövidítések**

Ezen fejezetben a disszertációban alkalmazott jelöléseket foglalom össze.

**5.1.1. A második fejezet jelölésrendszere**

5.1. táblázat Az ütemezési problémák tárgyalása során használt jelölések és magyarázataik

Jelölés	Magyarázat
<b>A</b>	A $G$ gráfot leíró szomszédsági mátrix
<b>C</b>	A teljes ütemezési mátrix, amely a $c_j$ vektorokból állítható elő
$c_{*,k}$	A <b>C</b> ütemezési mátrix $k$ -adik oszlopvektora
$c_j$	A $j$ feladathoz tartozó ütemezési vektor
diag( . )	Egy mátrix diagonálisát reprezentáló függvény
<b>F</b>	A $G$ gráf által leírt WSN hálózat interferencia mátrixa
$G = (V, E)$	WSN hálózatot reprezentáló gráf
H(.)	Entrópia az ütemezés egyenletes terheltségének jellemzésére
$J$	Az ütemezendő feladatok száma, illetve az node-ok száma a WSN-ben
$K_j$	Az $j$ feladathoz tartozó határidő, illetve az $j$ node maximális élettartama
$L$	A teljes ütemezés hossza
$o_1, o_2$	Szimulációk során használt konstans értékek, az intervallumok meghatározásához
$p_j$	A $j$ node-hoz tartozó összes átküldendő (továbbküldendővel együttesen) csomagok száma
$q(k)$	A $k$ időszelethez tartozó forgalmi vektor, amely leírja az egyes node-ok pufférének állapotát

<b>R</b>	A $G$ gráf által leírt WSN hálózatban az útvonalválasztást meghatározó mátrix (routing)
random( $\Gamma$ )	A $\Gamma$ intervallumon egyenletes eloszlású véletlen számokat generáló függvény
$T_j$	A $j$ ütemezett feladathoz tartozó késleltetés mértéke
$V$	A kiszolgáló egység egy időszelethez tartozó kapacitása, illetve a WSN klaszterfej egy időszelvényben fogadható maximális csomagok száma
<b>W, b</b>	A kvadratikus optimalizálás súlymátrixa és súlyvektora
<b>W<sub>A</sub>, b<sub>A</sub>, W<sup>'</sup><sub>A</sub>, b<sup>'</sup><sub>A</sub></b>	Feladatméretre vonatkozó peremfeltételhez tartozó súlyok
<b>W<sub>B</sub>, b<sub>B</sub></b>	Kapacitásra vonatkozó peremfeltételhez tartozó súlyok
<b>W<sub>C</sub>, b<sub>C</sub>, W<sup>'</sup><sub>C</sub>, b<sup>'</sup><sub>C</sub></b>	TT és TWT feladat célfüggvényére vonatkozó feltételhez tartozó súlyok
<b>W<sub>D1</sub>, b<sub>D1</sub>, W<sub>D2</sub>, b<sub>D2</sub></b>	Egyenletes terhelést garantáló célfüggvényhez tartozó súlyok
<b>W<sub>E</sub>, b<sub>E</sub></b>	Forgalmi paramétereknek való megfelelést biztosító súlyok
<b>W<sub>F</sub>, b<sub>F</sub>, W<sub>G</sub>, b<sub>G</sub>, W<sub>O</sub>, b<sub>O</sub></b>	Interferenciát és puffert kiürülést gátló peremfeltételhez tartozó súlyok, illetve a WSN hálózat ütemezéséhez tartozó célfüggvény súlyai
$w_j$	A $j$ feladathoz tartozó súlytényező
$X_j$	A $j$ feladat mérete, illetve az $j$ node által küldendő csomagok száma
<b>y</b>	Kvadratikus alak optimalizálandó vektora, HNN állapotvektora
$\alpha, \beta, \gamma, \delta, \varepsilon, \zeta, \xi$	Az egyes összegzett súlymátrixok és súlyvektorok tagjait prioritizáló heurisztikus paraméterek

### 5.1.2. A harmadik fejezet jelölésrendszere

5.2. táblázat Az esemény detekciós feladat tárgyalása során használt jelölések és magyarázataik

Jelölés	Magyarázat
$C_l$	$l$ -edik klaszter
$a, v$	Autoregresszív folyamat paraméterei (súlyok és zaj)
$F()$	Az outlier értékeket generáló folyamat eloszlásfüggvénye
$I_i$	Bináris indikátor, amely a $j$ node predikciós döntését reprezentálja

---

$L$	Klaszter számossága
$m_l$	Az egyes node-ok által esemény hatására érzékelt érték
$N$	Node-ok száma
$t$	Outlier kiugrasztási tényező
$T_{kl} \dots T_{kl}$	Esemény detekciós módszer működési periódusainak hossza
$TP, FP, TN, DR, FPR, F$	Esemény detekció pontosságát jellemző hibákat jellemző értékek és döntési metrikák (lásd a rövidítések jegyzékében)
$\mathbf{w}^{(i)}$	Az $i$ -edik node-hoz tartozó prediktor szabad paraméterei (prediktor súlyai)
$\mathbf{x}^{(i)}$	Az $i$ -edik node-hoz tartozó mérési idősor
$x'_n$	A mérési idősor predikált következő $n$ -edik eleme
$\mathbf{y}^{(i)}$	Az $i$ -edik node-hoz tartozó predikciós döntések vektora
$\Gamma(\cdot)$	Általános prediktor függvény (lineáris/nemlineáris)
$\Delta^{(i)}$	Az $i$ -edik node-hoz tartozó döntési küszöb
$\eta^{(i)}$	Az $i$ -edik node-hoz tartozó optimális prediktor által generált folyamat
$\xi^{(i)}$	Az $i$ -edik node-hoz tartozó modell, autoregresszív folyamat leírása
$\xi^{(i)} = \xi^{(i)} + \chi^{(i)}$	A modell szerinti tényleges érzékelt folyamat
$\tau$	Tanulóhalmaz a prediktorok optimalizálásához
$\Phi(\cdot)$	Gaussi eloszláshoz tartozó valószínűségi sűrűségfüggvény
$\varphi(\cdot)$	A neurális hálózat alapú prediktorokban szereplő nemlineáris kernelfüggvény
$\chi^{(i)}$	Outlier modell az $i$ -edik node-hoz tartozó mérésekhez
$\Psi(\cdot)$	Monoton függvény, amely az esemény által gyakorolt hatás fizikai csillapodását reprezentáló függvény

---

## 5.2. Rövidítések jegyzéke

Ezen fejezetben a disszertációban előforduló rövidítéseket és azok jelentését gyűjtöm össze betűrendben.

Rövidítés	Magyarázat
AR	Autoregresszív folyamat
BS	Base Station - Bázisállomás
CESVM	Centered HyperEllipsoidal SVM – Normalizált hiperellipszist alkalmazó SVM
CUDA	Párhuzamost programozást támogató architektúra GPGPU-ra az nVidia fejlesztésében
DR	Detection Rate – (Helyes) detekciós arány
EDD	Earliest Due Date – Ütemező protokoll, amely a legrövidebb határidővel rendelkező feladatot ütemezi
FFNN	Feedforward Neural Network – Előrecsatolt neurális hálózat
FN	False Negative – Kihagyott detekció
FP/TP	False Positive/True Positive – Hibás detekció/Helyes detekció
FPGA	Field-Programmable Gate Array – A felhasználás helyén programozható logikai kapumátrix
GPGPU	General Purpose Graphical Processing Unit – Általános célú grafikai processzor
GPU	Graphical Processing Unit – Grafikai processzor
HNN	Hopfield Neural Network – Hopfield féle csak visszacsatolást tartalmazó mesterséges neuronhálózat
LBS	Load Balance Scheduling – Egyenles ütemezést biztosító ütemezés
LP	Linear Prediction – Lineáris predikció
LWPF	Largest Weighted Process First – Ütemező protokoll, amely a súlyt veszi figyelembe az ütemezés kialakításánál
MAC	Media Access Control – Közeghozzáférés, és –szabályozás
PC	Personal Computer – Személyi számítógép

---

PLWPF	Perturbed LWPF
QSVM	Quarter Sphere SVM – Negyedgömböt használó SVM
RBF	Radial Basis Function – Hálózat, amelyben az egyes egységekben a nemlinearitás gömbszimmetrikus függvény
ROC	Receiver Operating Characterictics – A detekció pontosságát jellemző görbe
RX/TX	Receive/Transmit – Fogadás/Küldés
SHNN/PSHNN	Smart HNN/Perturbed SHNN
SIMD	Single Instruction Multiple Data
SVM	Support Vector Machine – Támogatóvektor alapú gép
TDMA	Time Division Multiple Access – Időosztály alapú többszörös hozzáférést engedélyező protokoll
TT	Total Tardiness – Összes késletetés
TWT	Total Weighted Tardiness – Összes súlyozott késletetés
WSN	Wireless Sensor Network – Vezeték nélküli érzékelő hálózat
WSPT	Weighted Shortest Processing Time – Ütemező protokoll, amely figyelembe veszi a súlyt és a feldolgozási időt

---

## A SZERZŐ PUBLIKÁCIÓI

### 6.1. Folyóiratokban

- [S1] G. Treplán, **K. Tornai**, J. Levendovszky, “Quadratic Programming for TDMA Scheduling in Wireless Sensor Networks”, *Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks* (Sensor Networks for High-Confidence Cyber-Physical Systems (HCPS)), 2011
- [S2] N. Fogarasi, **K. Tornai**, J. Levendovszky, “A Novel Hopfield Neural Network Approach for Minimizing Total Weighted Tardiness of Jobs Scheduled on Identical Machines”, *Informatica Acta Universitatis Sapientiae*, **4** (1), 2012, pp. 48-66
- [S3] **K. Tornai**, A. Oláh, J. Levendovszky, “Monitoring Algorithm for Intrusion and Danger Detection in Wireless Sensor Networks”, reviewed and resubmitted to *Ad hoc & Wireless Sensor Networks*, Old City Publishing, 2013
- [S4] **K. Tornai**, N. Fogarasi, J. Levendovszky, “Improvements to the Hopfield Neural Network Solution of the Total Weighted Tardiness Scheduling Problem”, in *Electrical Engineering and Computer Science*, Periodica Polytechnica, **57** (1), 2013, DOI 10.3311/PPee.2090

### 6.2. Konferenciákon

- [S5] J. Levendovszky, E. László, **K. Tornai**, and G. Treplán, “Optimal pricing based resource management”, *International Conference on Operations Research Munich 2010*, September 2010, pp. 169
- [S6] E. László, **K. Tornai**, G. Treplán, J. Levendovszky, “Novel load balancing scheduling algorithms for Wireless Sensor Networks”, *The Fourth International Conference on Communication Theory, Reliability, and Quality of Service*, CTRQ 2011 April 17-22, 2011
- [S7] J. Levendovszky, A. Oláh, **K. Tornai**, G. Treplán, “Novel load balancing algorithms ensuring uniform packet loss probabilities for WSN”, *2011 IEEE 73rd Vehicular Technology Conference* Budapest, May 15-18, 2011

## BIBLIOGRÁFIA

- [1] P. Brucker, *Scheduling Algorithms*, 5th edition, New York: Springer, 2007
- [2] M. Pinedo, *Scheduling – Theory, Algorithms and Systems*, 3rd edition, New York: Springer, 2008
- [3] S. Sahni, “Preemptive scheduling with due dates”, *Operations Research*, **27** (5), 1979, pp. 925–934
- [4] C. Martel, “Preemptive Scheduling with Release Times, Deadlines, and Due Times”, *Journal of the ACM*, **29** (3), 1982, pp. 812–829
- [5] E. L. Lawler, “Preemptive scheduling of precedence–constrained jobs on parallel machines, deterministic and stochastic scheduling”, *Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems*, 1981, pp. 101–123
- [6] J. Du, J. Y. T. Leung, “Minimizing total tardiness on one machine is NP–hard”, *Mathematics of Operations Research*, **15** (3), 1990, pp. 483–495
- [7] E. L. Lawler, “A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness”, *Annals of Discrete Mathematics*, 1, 1977, pp. 331–342
- [8] M. Azizoglu, O. Kirca, “Tardiness minimization on parallel machines”, *International Journal of Production Economy*, 55, 1998, pp. 163–168
- [9] R. M. V. Rachamadugu, T. E. Morton, “Myopic heuristics for the weighted tardiness problem on identical parallel machines”. Technical report, (CMU–RITR–83–17) Carnegie–Melon University, the Robotics Institute, 1983
- [10] D. Biskup, J. Herrman, J. N. D. Gupta, “Scheduling identical parallel machines to minimize total tardiness”, *International Journal of Production Economy*, 115, 2008, pp. 134–142
- [11] W. Ye, J. Heidemann, and D. Estrin, “An energy–efficient MAC protocol for wireless sensor networks”, *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, New York, NY, USA 2002, pp. 1567–1576
- [12] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks”, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, ACM, Baltimore, Md, USA, 2004, pp. 95–107
- [13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X–MAC: a short preamble MAC protocol for duty–cycled wireless sensor networks”, *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, ACM, New York, NY, USA, 2006, pp. 307–320
- [14] S. Ramanathan, “A unified framework and algorithm for channel assignment in wireless networks”, *Wireless Networks*, **5** (2), 1999, pp. 81–94
- [15] W.Z. Song, R. Huang, B. Shirazi, and R. LaHusent, “TreeMAC: Localized TDMA MAC protocol for realtime high–data–rate sensor networks”, *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, 2009, pp. 1–10

- [16] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, “Cross-layer energy and delay optimization in small-scale sensor networks”, *IEEE Transactions on Wireless Communications*, **6** (10), 2007, pp. 3688–3699
- [17] A. G. Ruzzelli, G. O’Hare, M. O’Grady, and R. Jurdak, “Merlin: cross-layer integration of MAC and routing for low duty-cycle sensor networks”, *Ad Hoc Networks*, **6** (8), 2007, pp. 1238–1257
- [18] D. M. Hawkins, *Identification of outliers*, Chapman and Hall, 1980
- [19] J. Xi, “Outlier detection algorithms in data mining”, *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application*, 2008
- [20] Y. Zhang and J. Liu, “An outlier mining algorithm based on probability”, *Power Electronics and Intelligent Transportation System (PEITS)*, 2009 2nd International Conference on, 2009, pp. 209–212
- [21] V. Karioti and C. Caroni, *Detecting an outlier in a set of time series*, Chania, 2002
- [22] E. M. Jordaan and G. E. Smits, “Robust Outlier Detection using SVM Regression”, *Neural Networks*, IEEE International Joint Conference on, 3, 2004
- [23] M. Bahrepour, N. Meratnia, and P. Havinga, “Sensor fusion-based event detection in wireless sensor networks”, *Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous*, 2009. MobiQuitous ’09. 6th Annual International, 2009, pp. 1–8
- [24] Y. Zhang, N. Meratnia, P. Havinga, “Adaptive and Online One-Class Support Vector Machine-based Outlier Detection Techniques for Wireless Sensor Networks”, *2009 International Conference on Advanced Information Networking and Applications Workshops*, IEEE, CIS, 2009
- [25] M. Bahrepour, N. Meratnia, P.J. Havinga, “An Online Outlier Detection Technique for Wireless Sensor Networks using Unsupervised Quarter-Sphere Support Vector Machine”, *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2008
- [26] S. Rajasegarar, C. Leckie, M. Palaniswami, J. C. Bezdek “Quarter Sphere Based Distributed Anomaly Detection in Wireless Sensor Networks”, *ICC 2007 proceedings*, IEEE, 2007
- [27] G. Amato, S. Chessa, C. Gennaro, C. Vairo, “Efficient Detection of Composite Events in Wireless Sensor Networks: Design and Evaluation”, *2011 Computers and Communications (ISCC)*, IEEE Symposium on, 2011
- [28] Q. Liang, L. Wang, “Event Detection in Wireless Sensor Networks Using Fuzzy Logic System”, *CIHSPS 2005 – IEEE International Conference on computational Intelligence for Homeland Security and Personal Safety*, 2005
- [29] N. Dziengel, G. Wittenburg, J. Schiller, “Towards Distributed Event Detection in Wireless Sensor Networks”, *DCOSS ’08*, 2008
- [30] P. Gupta and P.R. Kumar, “The capacity of wireless networks”, *IEEE Transactions on Information Theory*, **46**, 2000, pp 388–404
- [31] CAPSIL, Mica2 technical parameters, 2010
- [32] I. Demirkol, C. Ersoy, and F. Alagoz, “MAC protocols for wireless sensor networks: a survey”, *IEEE Communications Magazine*, **44**, 2006, pp. 115-121
- [33] C.J. Merlin and W.B. Heinzelman, “Duty cycle control for low-power-listening MAC protocols”, *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, IEEE, 2008, pp. 497-502



- [34] S.C. Ergen and P. Varaiya, “TDMA scheduling algorithms for wireless sensor networks”, *Wireless Networks*, **16**, 2009, pp. 985-997
- [35] I. Rhee and a Warriier, “DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks”, *IEEE Transactions on Mobile Computing*, **8**, 2009, pp. 1384-1396
- [36] A.S. Tanenbaum, *Computer Networks (4th Edition)*, Prentice Hall, 2002
- [37] S.C. Ergen and P. Varaiya, “PEDAMACS: power efficient and delay aware medium access protocol for sensor networks”, *Mobile Computing, IEEE Transactions on*, **5**, 2006, pp. 920-930
- [38] J. Nocedal and S. J.Wright, *Numerical Optimization*, Springer, Berlin, Germany, 2006
- [39] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann, Berlin, Germany, 1988
- [40] Z. Q. Luo, W. K. Ma, A.-C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems”, *IEEE Signal Processing Magazine*, **27** (4), 2010, pp. 20–34
- [41] S. Sahni, “Computationally related problems”, *SIAM Journal on Computing*, **3** (4), 1974 pp. 262–279
- [42] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems”, *Biological Cybernetics*, **55** (3), 1985, pp. 141–146
- [43] J. Mandziuk, “Solving the travelling salesman problem with a Hopfield—type neural network”, *Demonstratio Mathematica*, **29** (1), 1996, pp. 219–231
- [44] J. Mandziuk and B. Macuk, “A neural network designed to solve the N–queens Problem”, *Biological Cybernetics*, **66** (4), 1992, pp. 375–379
- [45] L. Liang, “Parallel implementations of Hopfield neural networks on GPU”, Technical Report, Dépôt Universitaire de Mémoires Apres Soutenance
- [46] R. Farber, *CUDA Application design and development*, Morgan Kaufmann 1<sup>st</sup> edition, 2009
- [47] D.B. Kirk, W.W. Hwu, *Programming massively parallel processors: A hands–on approach*, Morgan Kaufman, 2010
- [48] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey”, *Computer Networks*, **38** (4), 2002, pp. 393–422
- [49] M. Last, A. Kandel, and H. Bunke, *Data mining in time series databases*, World Scientific, 2004
- [50] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley, 1994.
- [51] V. Barnett, “The study of outliers: Purpose and model”, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1978
- [52] V. J. Hodge and J. Austin, “A survey of outlier detection methodologies”, *Artificial Intelligence Review*, **22** (2), 2004, pp. 85–126
- [53] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier Detection Techniques For Wireless Sensor Networks: A Survey”, University of Twente, Department of Computer Science, P.O.Box 217 7500AE, Enschede, The Netherlands, Technical Report, 2008
- [54] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng, “Localized Outlying and Boundary Data Detection in Sensor Networks”, *IEEE Transactions on Knowledge and Data Engineering*, **19** (8), 2007, pp. 1145–1157
- [55] L.A. Bettencourt, A. Hagberg, and L. Larkey, “Separating the Wheat from the Chaff: Practical Anomaly Detection Schemes in Ecological Applications of Distributed Sensor Networks”,

- Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems*, 2007
- [56] Y. Hida, P. Huang, and R. Nishtala, *Aggregation Query under Uncertainty in Sensor Networks*, 2003
- [57] B. Sheng, Q. Li, W. Mao, and W. Jin, “Outlier Detection in Sensor Networks”, *Proceedings of MobiHoc*, 2007
- [58] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, “Distributed Deviation Detection in Sensor Networks”, *ACM Special Interest Group on Management of Data*, 2003, pp. 77–82
- [59] D. Luo, X. Wang, X. Wu, and H. Chi, “Outliers learning and its applications”, *Neural Networks and Brain*, ICNN & B ’05. International Conference on, 2005, pp. 661–666
- [60] M. Bahrepour, Y. Zhang, N. Meratnia, and P. J. Havinga, “Use of Event Detection Approaches for Outlier Detection in Wireless Sensor Networks”, *International Conference on Intelligent Sensors*, Sensor Networks and Information Processing, 2009
- [61] S. Harkins, H. He, G. Williams, and R. Baster, „Outlier detection using replicator neural networks”, *DaWaK’02*, 2002, pp. 170–180
- [62] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, P.J.M. Havinga, “Disaster Management”, 2010 *International Conference on Intelligent Networking and Collaborative Systems*, 2010
- [63] D. Janakiram, A. Mallikarjuna, V. Reddy, and P. Kumar, “Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks” , *Proceedings of IEEE Comsware*, 2006
- [64] E. Elnahrawy and B. Nath, “Context–Aware Sensors”, *Proceedings of EWSN*, 2004
- [65] Y. Zhang and J. Liu, “An outlier mining algorithm based on probability”, *Power Electronics and Intelligent Transportation System (PEITS)*, 2<sup>nd</sup> International Conference on, 2009, pp. 209 – 212
- [66] M. Momma, K. Hatano, and H. Nakayama, “Ellipsoidal Support Vector Machines”, 2nd Asian Conference on Machine Learning (ACML2010), Tokyo, Japan, Nov. 8, 10, 2010.
- [67] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, “Distributed anomaly detection in wireless sensor networks”, *Communication systems*, 2006. ICCS 2006. 10th IEEE Singapore International Conference on, 2006
- [68] I. H. Witten and E. Frank, *Data Mining, Practical Machine Learning Tools and Techniques*, Elsevier, 2nd edition, 2005
- [69] R. G. Brown, *Smoothing, Forecasting and Prediction of Discrete Time Series*, Dover Publication Inc., 2004
- [70] K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, *Artificial Neural Networks – LNCS: Predicting Time Series with Support Vector Machines*. Springer, 1997.
- [71] G. E. P. Box, G. M. Jenkins, G. C. Reinsel *Time Series Analysis: Forecasting and Control* , Wiley, 4<sup>th</sup> ed, 2008
- [72] J. Makhoul, “Linear prediction: A tutorial review”. *Proceedings of the IEEE*, **63** (4), 1975, pp 561–580
- [73] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the National Academy of Sciences of the United States of America*, **79** (8), 1982, pp. 2554–2558

- 
- [74] K. Taehwan, A. Tülay, “Approximation by Fully Complex Multilayer Perceptrons”, *Neural Computation*, **15** (7), 2003, pp 1641–1666
- [75] S. Haykin *Neural Networks, A Comprehensive Foundation*, Pearson, Prentice Hall, 2<sup>nd</sup> edition, 2005
- [76] K. Gurney, *An Introduction to Neural Networks*, UCL Press, 1997
- [77] N. Christianini, J. Shawe–Taylor, *An introduction to Support Vector Machines and other kernel–based learning methods*, Cambridge University Press, 2003
- [78] Y. Zhu, Y. Liu, and L. Ni, “Optimizing event detection in low duty–cycled sensor networks”. *Wireless Networks*, 18, 2012, pp. 241–255
- [79] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors”, *Communications of the ACM*, **43** (5), 2000, pp 51–58
- [80] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, “Energy–aware wireless microsensor networks”, *Signal Processing Magazine, IEEE*, **19** 2, 2002, pp 40–50
- [81] H. Huang *Distributed Computing in Wireless Sensor Networks*, IGI Global, 2007
- [82] NVIDIA CUDA™ NVIDIA CUDA C Programming Guide, nVidia, 2012
- [83] Erik H. D’Hollander, D. Stroobandt, and A. Touhafi: *Parallel Computing with FPGAs - Concepts and Applications* in *Parallel Computing: Architectures, Algorithms and Applications*, NIC, **38**, 2007.
- [84] B. Varga and K. Karacs “GPGPU Accelerated Scene Segmentation Using Nonparametric Clustering” in *2010 International Symposium on Nonlinear Theory and its Applications*, 2010