



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**KABLOSUZ ALGILAYICI AĞLARDA RSSI
TABANLI KÜMELEME ALGORİTMASININ
GELİŞTİRİLMESİ**

Naim KARASEKRETER

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Aralık-2020
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Naim KARASEKRETER tarafından hazırlanan “*Kablosuz Algılayıcı Ağlarda RSSI Tabanlı Kümeleme Algoritmasının Geliştirilmesi*” adlı tez çalışması 17/12/2020 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Halife KODAZ

.....

Danışman

Prof. Dr. Fatih BAŞÇİFTÇİ

.....

Üye

Prof. Dr. Harun UĞUZ

.....

Üye

Doç. Dr. Humar KAHRAMANLI ÖRNEK

.....

Üye

Dr. Öğr. Üyesi Ömer Kaan BAYKAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.



İmza

Naim KARASEKRETER

Tarih: 17.12.2020

ÖZET

DOKTORA TEZİ

KABLOSUZ ALGILAYICI AĞLARDA RSSI TABANLI KÜMELEME ALGORİTMASININ GELİŞTİRİLMESİ

Naim KARASEKRETER

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Fatih BAŞÇİFTÇİ

2020, 128 Sayfa

Jüri

**Prof. Dr. Halife KODAZ
Prof. Dr. Fatih BAŞÇİFTÇİ
Prof. Dr. Harun UĞUZ
Doç. Dr. Humar KAHRAMANLI ÖRNEK
Dr. Öğr. Üyesi Ömer Kaan BAYKAN**

Kablosuz Algılayıcı Ağ (KAA), çok sayıda algılayıcı düğümün (AD) düzenli ya da düzensiz formlarda alana yayılması ve algıladıkları verileri birbiri üzerinden atlamalı olarak merkeze iletmeleri şeklinde tanımlanabilir. KAA'yı oluşturan AD'ler, kısıtlı enerjiye sahip, üzerinde algılayıcı birimi olan küçük ve düşük maliyetli birimlerdir. Kablosuz haberleşmede verinin iletim mesafesi tüketilen enerji ile doğru orantılı olduğundan iletim mesafenin kısaltılması gerekmektedir. Bu nedenle veri iletimi çok atlamalı (multi-hop) yapılmaktadır. Çok atlamalı yapıda, AD'ler verilerini merkeze iletmek için ağa yayınlar ve yakın düğümler bu yayını alır ve onlarda ağa yayınlarlar. Bu şekilde veri merkeze ulaşıncaya kadar süreç devam ettirilir. Ancak bu durum, çarpışma, örtüşme veya düğümlerin aşırı yüklenmesi gibi pek çok sorunu beraberinde getirmektedir. Bu problemlerin en aza indirilerek verinin en düşük enerji maliyetiyle merkeze iletilmesi KAA uygulamalarının temel amacıdır.

Bu tez çalışmasında, ağın enerji verimini arttıran ve gereksiz veri yükünü ortadan kaldırmayı amaçlayan yeni bir yönlendirme algoritması önerilmiştir. Önerilen Merkez Eğilimli Kümeleme Algoritması (MEKA), düğümlerin kendi içlerinde kümelere ayrılmasını ve bir lider seçerek verilerini bu lider üzerinden merkeze aktarılması mantığını esas alır. MEKA, literatürdeki benzer algoritmalarla enerji verimi, kapsama alanı ve hayatta kalan düğüm sayıları yönünden karşılaştırmalı analiz edilmiştir. Sonuç olarak MEKA'nın LEACH'e göre hayatta kalan düğüm sayısı yönünden %30, gönderilen paket sayısı bakımından %18, ağda kalan toplam enerji yönünden %58,7 ve kapsama alanı yönünden %13 iyileşme sağladığı ve k-ortalamlar algoritmasındaki kapsama alanı başarımına yaklaştığı gözlemlenmiştir.

Anahtar Kelimeler: Kablosuz Algılayıcı Ağ, Kümeleme, Yönlendirme, Yönlendirme Algoritması, Öz-örgütlü Kümeleme.

ABSTRACT

PhD THESIS

DEVELOPING RSSI-BASED CLUSTERING ALGORITHM IN WIRELESS SENSOR NETWORKS

Naim KARASEKRETER

**Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering**

Advisor: Prof. Dr. Fatih BAŞÇİFTÇİ

2020, 128 Pages

Jury

**Prof. Dr. Halife KODAZ
Prof. Dr. Fatih BAŞÇİFTÇİ
Prof. Dr. Harun UĞUZ
Doç. Dr. Humar KAHRAMANLI ÖRNEK
Dr. Öğr. Üyesi Ömer Kaan BAYKAN**

Wireless Sensor Network (WSN) can be defined as the spreading of a large number of sensor nodes (SN) in regular or irregular forms to the area and transmitting the data they perceive to the center by hopping over each other. Sensor nodes forming wireless sensor networks; They are units with low cost, limited energy and with a sensor unit. Since the transmission distance of data in wireless communication is directly proportional to the energy consumed, the distance must be shortened. For this reason, data transmission in WSN is preferred as multi-hop. In a multi-hop structure, SNs broadcast their data to the network to transmit their data to the base station (BS) and close nodes receive and broadcast this broadcast on them. In this way, the process continues until it reaches the base station. However, this situation brings with it many problems such as collision, overlap or overload of nodes. Minimizing these problems and transmitting data to the BS with the lowest energy cost is the main purpose of WSN applications.

In this thesis, a new routing algorithm is proposed that increases the energy efficiency of the network and aims to eliminate unnecessary data load. The proposed Center Oriented Clustering Algorithm (CCA) is based on the logic of separating the nodes into clusters and selecting a leader and transferring their data to the BS through this leader. CCP was analyzed comparatively with similar algorithms in the literature in terms of energy efficiency, packet numbers, coverage and surviving node numbers. As a result, it was observed that MEKA achieved 30% improvement in the number of surviving nodes, 18% in the number of packets sent, 58.7% in terms of total energy remaining in the network, and 13% in terms of coverage area according to LEACH and approached the coverage area performance in the k-means algorithm.

Keywords: Clustering, Routing, Routing Algorithm, Self-organized Clustering, Wireless Sensor Network.

ÖNSÖZ

Her insanın hayatında, ona değerli tecrübeler kazandıran olaylar ve zamanlar olmuştur, olacaktır. Benim için doktora eğitimi bunlardan biridir. Bana çok değerli tecrübeler kazandırmış, insanları tanımama vesile olmuş, kendi iç dünyamı ve maneviyatımı şekillendirmiş ve güçlendirmiştir. O nedenle çok kıymetli ve unutulmazdır ama bir o kadar zor olmuştur.

Bu tez çalışmasında, alana rastgele bırakılan algılayıcı düğümlerin kendi aralarında kümelenerek bir ağ oluşturmaları ve verilerini merkezi birime iletmelerini sağlayan enerji verimli bir algoritma önerilmiştir. Önerilen algoritma, literatürdeki benzer algoritmalar ile simülasyon ortamında karşılaştırılmış ve elde edilen sonuçlar detaylı bir şekilde sunulmuştur.

Bu zorlu sürecin her aşamasında bana yardımcı olan kıymetli hocam Sayın Prof. Dr. Fatih BAŞÇİFTÇİ' ye, değerli TİK üyelerim Sayın Prof. Dr. Harun UĞUZ ve Doç. Dr. Humar KAHRAMANLI ÖRNEK' e, çalışmanın her aşamasında desteğini esirgemeyen Sayın Dr. Öğr. Üyesi Uğur FİDAN hocama ve Sayın Dr. Öğr. Üyesi Mehmet Akif ŞAHMAN'a,

Değerli büyüklerim; Sayın Ceylan EMET, Sayın Abdülkadir EMRULLAHOĞLU ve Sayın Prof. Dr. Şahabettin YİĞİTBAŞI'na,

Hayat arkadaşım, dert ortağım sevgili eşim Meral Hanıma, oğullarım Nail ve Talha'ya,

Kıymetlerini kelimelerle anlatamayacağım Babama ve Anneme,

İsimlerini saymadığım tüm dostlarıma ve mesai arkadaşlarıma şükranlarımı bir borç bilirim. Hepinize sonsuz teşekkürler...

Naim KARASEKRETER
KONYA-2020

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER VE ÇİZELGELER	xiii
Şekiller	xiii
Çizelgeler	xv
1. GİRİŞ	1
1.1. Tezin Amacı ve Önemi	3
1.2. Tezin Organizasyonu	5
2. KAYNAK ARAŞTIRMASI	6
2.1. Makale Çalışmaları	7
2.2. Yüksek Lisans Tezleri	19
2.3. Doktora Tezleri	20
3. MATERYAL VE YÖNTEM	24
3.1. Algılayıcılar	24
3.1.1. Algılayıcı düğümün donanım özellikleri	25
3.1.1.1. Algılama bölümü	26
3.1.1.2. İşlem bölümü	26
3.1.1.3. Alıcı/Verici	26
3.1.1.4. Enerji bölümü ve güç tüketimi.....	26
3.1.2. İşletim sistemi	28
3.2. Kablosuz Algılayıcı Ağlar	29
3.2.1. Ortam erişim protokolleri	29
3.2.1.1. TDMA (Time Division Multiple Access).....	30
3.2.1.2. FDMA (Frequency Division Multiple Access)	31
3.2.1.3. CSMA (Carrier Sense Multiple Access).....	31
3.2.2. KAA yapıları.....	32
3.2.3. Veri iletim yöntemleri.....	33
3.2.4. Veri merkezli protokoller.....	35
3.2.4.1. Flooding protokolü	35
3.2.4.2. Rumor protokolü.....	36
3.2.4.3. SPIN protokolü	36
3.2.4.4. Energy Aware Routing (EAR).....	37

3.2.5. Konum tabanlı yönlendirme protokolleri	37
3.2.6. Hiyerarşik yönlendirme protokolleri	37
3.2.6.1. LEACH	39
3.2.6.2. TEEN ve APTEEN	41
3.2.6.3. HEED	41
3.2.6.4. PEGASIS	42
3.2.6.5. k-means algoritması	43
3.2.6.6. TD-PSO algoritması	44
4. MERKEZ EĞİLİMLİ KÜMELEME ALGORİTMASI (MEKA)	46
4.1. Kurulum Aşaması	47
4.1.1. RSSI'dan mesafe hesaplama	49
4.2. KB Seçimi ve Ağın Kümelenmesi	56
4.3. Radyo Modeli	63
4.4. Simülasyon Programı	64
5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	67
5.1. Algoritmaların Küme Oluşumu ve Dağılımı Açısından Değerlendirilmesi	67
5.2. Kapsama Alanlarının Değerlendirilmesi	72
5.3. Paket Sayıları ve Enerji Tüketimleri Açısından Değerlendirme	79
5.4. Ölen ve Hayatta Kalan Düğümler Açısından Değerlendirme	84
5.5. Algoritmaların Zaman Karmaşıklığı Bakımından Değerlendirilmesi	86
5.6. Tartışma	88
6. SONUÇLAR VE ÖNERİLER	96
6.1. Sonuçlar	96
6.2. Öneriler	98
KAYNAKLAR	99
TURNİTİN	109
ÖZGEÇMİŞ	119

SİMGELER VE KISALTMALAR

Simgeler

d_{avr}	: Düğümün diğer düğümlere olan ortalama mesafesi
Nid	: Düğüm kimliği
N	: Düğüm sayısı
$E_{residual}$: Düğümün kalan enerjisi
E_{max}	: Düğüm referans ya da başlangıç enerjisi
E_{elec}	: Elektronik devrelerin harcadığı enerji
d_{max}	: En uzak düğümün mesafesi
d_{min}	: En yakın düğümün mesafesi
$P_i(t)$: Eşik değer fonksiyonu
ω	: Fonksiyon sabiti
d_x	: İletim mesafesi
$C_i(t)$: İndikatör fonksiyonu
r	: İterasyon sayısı
d_i	: i. Dügüme olan mesafe
c_i	: Kümenin ağırlık merkezi
p	: Küme başı olma yüzdesi
C_{prob}	: Küme başı olma yüzdesi
n_t	: Küme içindeki düğüm sayısı
CH_{prob}	: Küme başı seçilme olasılık değeri
k	: Küme sayısı
μA	: Mikroamper
ma	: Miliamper
Avr_i	: Ortalamalardan oluşan mesafe dizisi
d_0	: Referans mesafe
$E_i(t)$: t. Dügümün enerjisi
$Mf(x)$: Üyelik fonksiyonu
l	: Verinin büyüklüğü
D	: Veri seti
η	: Yayılım/yol kaybı
E_{amp}	: Yükselteç birimin harcadığı enerji

Kısaltmalar

AB	: Algılayıcı Birim
AD	: Algılayıcı Düğüm
ADC	: Analog/Sayısal Dönüştürücü
ADRP	: Adaptive Decentralized Re-Clustering Protocol (Uyarlanabilir Merkezi Olmayan Yeniden Kümeleme Protokolü)
AEERRP	: Adaptive Energy Efficient Forwarding Routing (Uyarlanabilir Enerji Verimli Yönlendirme)
ANFIS	: Adaptif Neuro-Fuzzy Inference System (Adaptif Nöro-Bulanık Çıkarım Sistemi)
AMN	: Ağırlık Merkezi Noktaları
APTEEN	: Adaptive Periodic Threshold-Sensitive Energy Efficient Sensor Network Protocol (Periyodik Uyarlamalı Eşik Algılamalı Enerji Verimli Kümeleme Protokolü)
BCDCP	: Base-Station Controlled Dynamic Clustering Protocol (Baz İstasyon Kontrollü Dinamik Kümeleme Protokolü)
CHFL	: Cluster Head Fuzzy Logic
CSMA	: Carrier Sense Multiple Access (Taşıyıcı Algılamalı Çoklu Erişim)
CTA	: Closer Tracking Algorithm (Daha Yakın İzleme Algoritması)
ÇYVLD	: Çift Yönlü Verimli Lokasyon Bazlı Dedikodu
DBS	: Distance-Based Segmentation (Mesafeye Dayalı Segmentasyon)
DEECIC	: Distributed Energy-Efficient Clustering with Improved Coverage (İyileştirilmiş Kapsama ile Dağıtılmış Enerji Açısından Verimli Kümeleme)
DECSA	: Distance-Energy Cluster Structure Algorithm (Uzaklık-Enerji Küme Yapısı Algoritması)
DGMA	: Distributed Group Mobility Adaptive (Dağıtılmış Grup Hareketliliği)
DHSCA	: Dual Head Static Clustering Algorithm
DPC-EERRP	: Distributed Power Control with Energy Efficient and Reliable Routing Protocol (Enerji Açısından Verimli ve Güvenilir Yönlendirme Protokolü ile Dağıtılmış Güç Kontrolü)
EADUC-II	: Energy Aware Distributed Unequal Clustering
EBCAG	: Energy-Balancing Unequal Clustering Approach for Gradient-Based Routing (Gradyan Tabanlı Enerji Dengeleyici Eşitsiz Kümeleme Yaklaşımli Yönlendirme)
EBCS	: Entropy Based Clustering Algorithm (Entropi Tabanlı Kümeleme Algoritması)
EBUC	: Energy-Balanced Unequal Clustering (Enerji Dengeli Eşitsiz Kümeleme)
ECDC	: Energy and Coverage-aware Distributed Clustering Protocol (Enerji ve Kapsama Duyarlı Dağıtılmış Kümeleme Protokolü)
ECRA	: Energy Efficient Clustering and Routing Algorithm
EEBCDA	: Energy Efficient And Balanced Cluster-Based Data Aggregation Algorithm (Enerji Verimli ve Dengeli Küme Tabanlı Veri Toplama Algoritması)
EECPK	: Energy Efficient Clustering Protocol Based on k-means
FDMA	: Frequency Division Multiple Access (Frekans Bölmeli Çoklu Erişim)
FHRP	: Fuzzy-Based Hyper Round Policy
FL-EEC/D	: Fuzzy Logic-based Energy-Efficient Clustering (Bulanık Mantık Tabanlı Enerji Verimli Kümeleme)

GA	: Genetik Algoritma
GPS	: Global Positioning System (Küresel Konumlama Sistemi)
GSM	: Global System for Mobile Communications (Küresel Mobil İletişim)
HEEC	: Hybrid Unequal Energy Efficient Clustering (Hibrit Enerji Verimli Kümeleme)
HEED	: Hybrid Energy-Efficient Distributed Clustering Approach (Enerji Verimli Dağıtık Hibrit Kümeleme Yaklaşımı)
HFAPSO	: Hybrid Approach of Firefly Algorithm with Particle Swarm Optimization (Parçacık Sürü Optimizasyonu ve Ateşböceği Algoritmalarının Hibritlenmesi)
HUCL	: Hybrid Unequal Clustering With Layering
IDS	: Independent Dominating Set (Bağımsız Hâkim Set)
IEEE	: Institute of Electrical and Electronics Engineers (Elektrik-Elektronik Mühendisliği Enstitüsü)
IIH	: Iterative Improvement Heuristic (Yinelemeli Sezgisel İyileştirme)
IoT	: Internet of Things (Nesnelerin İnterneti)
I-LEACH	: Improved-LEACH (Gelişmiş LEACH)
IP	: Internet Protocol (İnternet Protokolü)
IR	: Kızıl Ötesi
KA	: Kablosuz Algılayıcı Ağ
KAD	: Kablosuz Algılayıcı Düğüm
KB	: Küme Başı
KOCA	: Multi-Hop Clustering Algorithm (Çoklu Atlamalı Kümeleme Algoritması)
KTDOP	: Karma Tam Sayılı Doğrusal Olmayan Programlama
LBCP	: Load-Balanced Clustering Problem (Yük Dengeli Kümeleme Sorunu)
LEACH	: Low Energy-Adaptive Clustering Hierarchy (Düşük Enerji Uyarlamalı Kümeleme Hiyerarşisi)
LEACH-C	: Low Energy Adaptive Clustering Hierarchy- Centralized (Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi – Merkezileştirilmiş)
LLACA	: Localized Learning Automata Based Clustering Algorithm (Yerelleştirilmiş Öğrenme Otomatına Dayalı Kümeleme Algoritması)
LND	: Last Node Died (Ölen Son Düğüm)
MAC	: Medium Access Control (Ortam Erişim Protokolü)
MEKA	: Merkez Eğilimli Kümeleme Algoritması
M-ICHB	: Modified Intelligent CH Election Based on Bacterial Foraging Optimization Algorithm (Güncellenmiş CH Seçimli Optimizasyon Algoritması)
Mİ	: Merkez İstasyon
MK-MEANS	: Modified k-means Algorithm (Güncellenmiş k-means)
M-LEACH	: Multi-hop LEACH (Çok Atlamalı LEACH)
MPC	: Multiple Parameter Based Clustering (Çoklu Parametre Tabanlı Kümeleme)
OCM-FC	: Optimal Clustering Mechanism Based On Fuzzy-C Means (Fuzzy-C Yaklaşımına En Uygun Kümeleme Mekanizması)
PEGASIS	: Power-Efficient Gathering in Sensor Information Systems (Sensör Bilgi Sistemlerinde Güç Verimli Bilgi Toplama)
PSNR	: Peak Signal to Noise Ratio (Tepe Sinyali- Gürültü Oranı)
PSO	: Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)
PSO-NMDC	: PSO Based Malicious Node Detection and Clustering

PTPC	: Practical- Transmission Power Control (Pratik- İletim Güç Kontrolü)
RAT	: Routing by Adaptive Targeting (Uyarlanabilir Hedeflemeye Göre Yönlendirme)
RE-EERP	: Residual Energy- Energy Efficient Routing Protocol (Kalan enerjiye göre- Enerji Verimi Yönlendirme Protokolü)
RF	: Radyo Frekansı
RSSI	: Receive Signal Strength Indicator (Alınan Sinyalin Güç İndikatörü)
RX	: Receive (Alma)
SECA	: Saving Energy Clustering Algorithm
SEECH	: Scalable Energy Efficient Clustering Hierarchy
SFS	: Siber-Fiziksel Sistem
S-MAC	: Sensör MAC (Sensor Medium Access Control)
TCAC	: Topology-Controlled Adaptive Clustering
TEEN	: Threshold Sensitive Energy Efficient Sensor Network Protocol (Eşik Algılamalı Enerji Verimli Kümeleme Protokolü)
TDMA	: Time Division Multiple Access (Zaman Bölmeli Çoklu Erişim)
TX	: Transmit (Gönderme)
TSA	: Telsiz Sensör Ağları
TSL	: Timing-Based Sensors Localization (Zamanlama Tabanlı Sensör Lokalizasyonu)
Q-LEACH	: Quadrature-LEACH (Dördün LEACH)
UCR	: Unequal Cluster based Routing (Eşitsiz Küme Tabanlı Yönlendirme)
XMPP	: Extensible Messaging and Presence Protocol (Genişletilebilir Mesajlaşma ve Durum Protokolü)
YÖK	: Yüksek Öğretim Kurumu

ŞEKİLLER VE ÇİZELGELER

Şekiller

Şekil 3.1. Örnek algılayıcılar	24
Şekil 3.2. Mica2 algılayıcı düğümü	25
Şekil 3.3. AD mimarisi	25
Şekil 3.4. AD'lerde kullanılan bataryalar	27
Şekil 3.5. Kablosuz algılayıcı ağ yapısı	29
Şekil 3.6. TDMA iletim modeli	30
Şekil 3.7. FDMA iletim modeli	31
Şekil 3.8. CSMA iletim modeli	32
Şekil 3.9. Tarımsal KAA uygulaması (AD: Algılayıcı Düğüm, Mİ: Merkez İstasyon). 32	
Şekil 3.10. Düzensiz dağıtık yapıdaki ağ modeli (Mİ: Merkez İstasyon)	33
Şekil 3.11. Örnek bir veri iletim modeli	34
Şekil 3.12. Örnek veri paketi	35
Şekil 3.13. Flooding protokolü çalışma prensibi	36
Şekil 3.14. SPIN protokolü çalışma prensibi	36
Şekil 3.15. LEACH algoritması akış diyagramı	40
Şekil 3.16. PEGASIS protokolü	43
Şekil 3.17. k-means algoritmasının sözde kodu	44
Şekil 4.1. Sorgu yayını TDMA şeması	48
Şekil 4.2. Örnek ağ modeli.	48
Şekil 4.3. Sorgu yayını veri paketi.....	49
Şekil 4.4. Cevap yayını veri paketi	49
Şekil 4.5. Alana rastgele bırakılan 2 düğüm arasındaki mesafenin örnekleme.....	50
Şekil 4.6. Hesaplanan ortalama yayını veri paketi.....	54
Şekil 4.7. Sıradaki düğümün yayına başlayabilmesi için gönderilen veri paketi	55
Şekil 4.8. Alıcı/verici düğüm kurulum yayını sözde kodu	55
Şekil 4.9. KB düğümün seçilmesi.....	56
Şekil 4.10. Eşik değeri yayın paketi.....	57
Şekil 4.11. MEKA algoritması kaba kodu.....	63
Şekil 4.12. Simülasyon programı.....	65
Şekil 4.13. Örnek şekil penceresi.....	65
Şekil 4.14. Örnek bir ağ modeli oluşturmak için kullanılan kod bloğu.....	66
Şekil 5.1. Ağ modeli	69
Şekil 5.2 LEACH algoritması KB seçimi (X: KB düğümler)	70
Şekil 5.3. k-means algoritması KB seçimi (*: Ağırlık merkezleri)	71
Şekil 5.4. MEKA algoritması KB seçimi	72
Şekil 5.5. Kapsama alanlarının gösterimi	73
Şekil 5.6. Örnek ağ modeli	74
Şekil 5.7. Aynı ağ modelinde algoritmaların kümeleme davranışları	75
Şekil 5.8. Aynı ağ modelinde algoritmaların KB seçim davranışları	76
Şekil 5.9. 20 farklı ağ modeli için, seçilen KB düğümlerin toplam birleşim alanları	78
Şekil 5.10. 20 iterasyon sonunda algoritmaların enerji tüketim davranışları ve ağda kalan toplam enerji (joule).....	81
Şekil 5.11. 50 farklı ağ modeli için ağda kalan toplam enerji miktarları	82
Şekil 5.12. Gönderilen toplam paket sayıları.....	83
Şekil 5.13. 20 farklı ağ modelinde paket gönderilemeyen iterasyonlar.....	83

Şekil 5.14. Ölü düğümlerin tespiti için kullanılan ağ modeli.....	84
Şekil 5.15. LEACH algoritmasında hayatta kalan düğümler.....	84
Şekil 5.16. MEKA algoritmasında hayatta kalan düğümler	85
Şekil 5.17. İterasyonlara göre ölü düğüm sayıları	85
Şekil 5.18. 20 farklı ağ modeline göre hayatta kalan düğümler ($E_0=1,5j$, iterasyon=50)	86
Şekil 6.1. MEKA'nın diğer algoritmalara göre ağ ömrü iyileştirme oranı.....	97
Şekil 6.2. Son düğümün öldüğü iterasyon sayıları	97



Çizelgeler

Çizelge 3.1. Literatürde yer alan bazı algılayıcı düğümler ve bu düğümlerin farklı durumlara ait enerji tüketimleri (https://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes)	27
Çizelge 3.2. A noktasından Mİ' ye ulaşmak için izlenecek rotalar ve enerji tüketimleri (AD _T : Düğümlerin mevcut toplam enerjileri, Y _T : Yolda harcanan toplam enerji, D _s : Atlanan düğüm sayısı)	34
Çizelge 3.3. Literatürdeki algoritmaların gruplandırılması	39
Çizelge 4.1. 15 düğüm için oluşturulan arama tablosu (Sarı ile işaretli olan bölgeler düğümlerin kendisine olan mesafesidir)	51
Çizelge 4.2. Her düğümün diğer düğümlere olan ortalama uzaklıklarının olduğu arama tablosu	53
Çizelge 4.3. Örnek dizi	54
Çizelge 4.4. 8. Düğüm için eşik değer fonksiyonunda kullanılan değerler (dmax : 8. düğümün en uzak olduğu düğüme olan mesafesi, dmin : 8. düğümün en yakın olduğu düğüme olan mesafesi)	57
Çizelge 4.5. 1. Turda KB seçilen düğümün üyelerinin belirlenmesi	59
Çizelge 4.6. 1. Turda kümelenen düğümlerin çıkarılması ve 2. tur KB seçimi	60
Çizelge 4.7. 9. Düğüm için eşik değer fonksiyonunda kullanılan değerler (dmax : 9. düğümün en uzak olduğu düğüme olan mesafesi, dmin : 9. düğümün en yakın olduğu düğüme olan mesafesi)	61
Çizelge 4.8. İlk 2 turda kümelenen düğümler	61
Çizelge 4.9. 2. Turda KB seçilen düğümün üyelerinin belirlenmesi	62
Çizelge 5.1 Ağın kümelenmesinde kullanılan parametreler ve değerler	68
Çizelge 5.2. Hesaplanan alan değerleri	77
Çizelge 5.3. Farklı alan ölçülerinde algoritmaların kapsama alanı performansları	78
Çizelge 5.4. Algoritmaların kapsama alanları yüzdesi	79
Çizelge 5.5. Enerji tüketimlerinin ve paket sayılarının hesaplanmasında kullanılan değerler	80
Çizelge 5.6. 20 İterasyon sonunda ağın enerji tüketim verileri	80
Çizelge 5.7. DHSCA algoritmasının test senaryoları	93
Çizelge 5.8. DHSCA algoritmasının LEACH, TCAC ve SEECH ile LND yönünden kıyaslanması	93
Çizelge 5.9. Tartışılan algoritmaların sayısal karşılaştırması	95

1. GİRİŞ

Çağının en önemli icatlarından olan tekerleğin bulunması ile başlayan ve matbaanın icadı ile devam eden teknoloji serüveni, 18.yy'da sanayi devriminin başlaması ve makineleşme ile önemli bir döneme girmiştir. Makinelerin, iş gücünü ve üretim kapasitesini arttırıyor olması, muhtemel insan hatalarını en aza indiriyor olması gibi avantajlarından dolayı ekonomik gelişmeye çok önemli katkılar sağlamıştır. Bu nedenlerden dolayı özellikle batı toplumları için vazgeçilmez bir hale gelmiştir. Bu durum şehirleşmenin önünü açmış, sosyal hayatta ve iş hayatında tarz değişikliklerine yol açmıştır (İşman, 2001). 19. yy. ortalarına doğru elektronikte birlikte programlanabilir cihazların geliştirilmesi 3. sanayi devrimini başlatmıştır. Bu süreçte, bilgisayar ve bilişim sistemlerinde, makine teknolojilerinde önemli ilerlemeler kaydedilmiş, internet kullanımının yaygınlaşması ile bilgi ve bilgiye ulaşımın önemi ortaya çıkmıştır. Aynı zamanda üretimde, yazılım ile çalışan otomasyon sistemlerinin kullanılmaya başlanması, üretim kapasitelerinin artmasına ve maliyetlerin düşmesine yol açmıştır. Bu devrimleri erken yakalayabilen toplumlar kendilerini ağır sanayiye dayalı bir ekonomik yapıdan teknolojinin geliştirilmesi ve ihraç edilmesine dayanan bir yapıya dönüştürmüş ve bu sayede çok ciddi bir ekonomik güce ve gelişmişliğe sahip olmuşlardır (Bulut ve Akçacı, 2017; Özkan ve ark. 2018).

Dünya genelinde internet altyapısının fiber optik teknolojisi ile güçlendirilmesi ve mobil haberleşme teknolojilerindeki hızlı gelişme bilgiye ulaşımı oldukça hızlandırmış ve 20. yy.'lin en önemli olayı olan 4. sanayi devriminin başlamasında önemli rol oynamıştır. Bilgiye ulaşım, teknolojik gelişmenin yaygınlaşmasına ve dolaylı olarak temel donanımların ucuzlamasına yol açmıştır. Örneğin, 1990'lı yılların sonunda mobil cihazlara sahip olma oranı oldukça düşükken 2000'li yılların başında bu tür cihazlara sahip olmanın yanında bu cihazları oluşturan Global System for Mobile Communications (GSM) modüller gibi donanımsal parçalara ulaşılmaya ve yaygın olarak kullanılmaya başlanmıştır. Bu durum, 4. sanayi devriminin en önemli farklarından olan ve Nesnelerin İnterneti olarak adlandırılan yeni bir kavramın doğmasına neden olmuştur. Dilimize, Nesnelerin İnterneti olarak geçen, İngilizcesi Internet of Things (IoT) olan kavram, gerçekte bilgisayar olmayan nesnelerin internete bağlanarak veri alıp gönderebilir olmaları şeklinde tanımlanabilir (Dengiz, 2017). Ancak nesnenin sadece internet ortamında veri alma/gönderme yapabiliyor olması yeterli değildir. Aynı zamanda nesnenin, algılayıcılar (sensör) yardımıyla bulunduğu ortamdan veri alması ve topladığı

verileri internet üzerinden paylaşıyor olması gerekmektedir. Algılayıcılar, fiziksel ortamdaki ısı, nem, ses vb. değişimleri algılayan ve bunu elektriksel bir işarete çeviren sistemlerdir. Entegre çalıştığı elektronik ve yazılım sistemleri sayesinde elektriksel işaret sayısal bir veriye dönüştürülür ve anlamlandırılır.

Algılayıcıların boyutlarının küçültülmesi, hassasiyetlerinin yükseltilmesi ve en önemlisi maliyetlerinin ucuzlaması IoT'un hızlı bir şekilde yaygınlaşmasına ve buna bağlı olarak farklı yapıların doğmasına sebep olmuştur, "*Akıllı Sistemler*". Akıllı sistemler, IoT kavramının bir üst sürümü olarak düşünülebilir. Bunun sebebi ise, algılanan verinin nesne tarafından sadece internet bulutuna iletilmesinin ötesinde yapay zekâ yöntemleri ile işlenmesi ve kendi karar süreçlerini nesnenin kendi içinde yönetebilmesidir. Örneğin; akıllı ev sistemleri, fiziksel temas olmadan sesli komutla çalışabilen "*SIRI*" (iOS sesli yönlendirme uygulaması) benzeri sistemler ya da cihazlar. IoT ile başlayan, akıllı teknolojiler ile devam eden süreç yerini sayısal dönüşüme bırakmıştır. Sayısal dönüşüm, akıllı cihazların bir ağa bağlı olmaları ya da ağ kurabilme kabiliyetlerine sahip olmaları şeklinde tanımlanabilir. Jirkovsk'y ve arkadaşları (2017); algılama, hesaplama, haberleşme ve kontrol yeteneğine sahip kablolu veya kablosuz çalışabilen sistemleri, Siber-Fiziksel Sistem (Cyber-Physical System- SFS) olarak tanımlamışlar ve 4. sanayi devriminin en önemli dayanaklarından birisi olarak kabul etmişlerdir.

Haberleşme, tarım, sağlık ve savunma gibi hayatın her alanında büyük yenilikler ve kolaylıklar sağlayan bu sistemlerin de zayıf olduğu yönler bulunmaktadır. Bunların başında enerji gelmektedir. Özellikle enerjisini pil veya batarya gibi kaynaklardan sağlayan sistemler için enerjinin verimli kullanımı gerçek bir sorundur ve üzerinde en çok durulması gereken konuların başında gelmektedir. Kablosuz sistemler için haberleşmede kullanılan donanımlarda, iletim mesafesinin harcanan güç ile doğru orantılı olması problemin ciddiyetini açıkça ortaya koymaktadır. Benzer durum mikroişlemci ve mikrodenetleyiciler içinde geçerlidir. Bu probleme karşı üretim ve tasarım aşamalarında farklı önlemler alınmaktadır. Düşük enerji tüketimi sağlayan yeni modeller geliştirilmesinin yanında uygulamaya göre farklılık göstermekle birlikte, tasarımcılar da tasarladıkları sistemin enerji verimini artırmak için, o anda kullanılmayan donanımların enerjilerini keser ve sistemi daha çok uyku halinde tutmayı tercih ederler. Yapılan tüm iyileştirmelere rağmen özellikle kablosuz haberleşmede mesafelerin uzaması enerji tüketimini olumsuz etkilemektedir. Bu duruma karşı son yıllarda önemli çalışmalar yapılmaya başlanmıştır. Yapılan çalışmalardaki ana hedef algılayıcı birimlerin verilerini

sorunsuz iletmeleri ve bunu mümkün olan en düşük enerji tüketimi ile gerçekleştiriyor olmasıdır. Bunun için önerilen metodolojinin temelinde birimlerin topladıkları verileri direkt olarak merkeze iletmesi yerine kendisine daha yakın başka bir algılayıcı birime iletmesi ve bu şekilde bir ağ kurularak verinin merkeze ulaştırılmasına dayanmaktadır. Bu sayede birimler daha kısa mesafedeki başka bir birim üzerinden verisini iletebildiğinden, enerji tüketimi düşecek ve algılayıcı birimin ömrü daha da arttırılmış olacaktır. Bahsedilen bu yapı, literatürde *Kablosuz Algılayıcı Ağlar (KAA)*, ağı oluşturan algılayıcı birimler ise *algılayıcı düğümler* olarak adlandırılmışlardır.

KAA ile getirilen bu mantıksal yapı savunmadan sağlığa pek çok alanda uygulama imkânı bulmuştur. Çok geniş bir araştırma alanına sahip olan KAA'larda, araştırmacıların 3 farklı alt başlık üzerinde yoğunlaştıkları gözlemlenmiştir. Bunlar,

- Algılayıcı birimlerin yerlerinin kestirilmesi için lokalizasyon ve konum belirleme,
- Algılanan verinin kurulan ağ içinde en kısa yoldan merkeze ulaştırılması için geliştirilen yönlendirme (routing) algoritmaları,
- Ağın kümelenmesi ve bir küme başı (KB) seçerek veriyi onun üzerinden merkeze iletmesi temeline dayanan çalışmalardır.

Kümelemede, algılayıcılar kümeler oluşturur ve her küme için bir KB seçerler. Seçilen KB'lere göre algılayıcılar en uygun küme başına tabi olarak kümeleri oluştururlar. Her algılayıcı verisini kendi küme başına iletir ve KB toplanan veriyi merkeze iletir. Literatürde, konuyla ilgili çalışmalarda ağın en yüksek enerji verimiyle hayatta kalmasına yönelik farklı kümeleme algoritmaları ve protokolleri geliştirilmiştir.

1.1. Tezin Amacı ve Önemi

Literatürde, veri iletimi için kümeleme yöntemi önemli bir yer tutmaktadır. Kümelemede iki temel yaklaşım benimsenmiştir. İlki, KB seçimi ve ağın kümelenmesinin Mİ tarafından yapılmasıdır. Bu yöntemde, düğümlerin Mİ ile haberleşebildiği var sayılır ve enerji seviyeleri, GPS (Global Positioning System) konum bilgileri vb. bilgilerini Mİ'ye iletir. Mİ, bu bilgilere göre KB seçimlerini yapar ve KB'lere bağlanacak düğümleri belirler. Sonucu ağa yayımlar ve bu şekilde ağ kümelenmiş olur. KAA'da veri iletimi sürekli bir işlemdir. Kümelemede ise veri iletim yükünü KB düğümler üzerine alırlar ve

bundan dolayı enerjileri hızla sönümlenmektedir. Bu takdirde ağın tekrar kümelenmesi ve yeni KB seçilmesi gerekmektedir. Bu işlem iterasyonlar halinde devam eder ve her iterasyonda kümeleme işlemi Mİ tarafından yapılmaktadır. Mİ yönlendirmeli algoritmaların önemli dezavantajları bulunmaktadır. Bunlar şu şekilde sıralanabilir;

- Mİ'nin haberleşemediği düğümler kümelemenin dışında kalmış olacak ve bağımsız hareket edeceklerdir. Bu durum, ağın kapsama alanını olumsuz yönde etkileyecektir.
- Düğümlerin konumlarının bilinmesi için AD üzerinde GPS modül olması gerekmektedir. AD'lerde GPS entegrasyonu ise ilave maliyet ve enerji yükü getirecektir.

İkinci yaklaşım ise, ağın herhangi bir müdahale olmadan kendi kendine kümelenmesinin sağlanmasıdır. Bu yaklaşıma öz-örgütlü kümelenme denilmektedir. Öz-örgütlü kümelenme yöntemi, literatürdeki pek çok kaynakta tavsiye edilmektedir. Çünkü, uygulama ve coğrafi alandan bağımsızdır. Örneğin, ormanlık bir alanda yangını önceden haber almak ve erken refleks göstermek amacıyla alanda algılama yapılması durumunda, algılayıcı düğümlerin alana düzenli olarak bırakılmaları mümkün değildir. Bu gibi uygulamalarda algılayıcıların hava aracından serpilerek alana yayılması sağlanmalıdır. Serpme yönteminde düğümlerin alandaki dağılımları homojen olmayacağından Mİ ile aralarındaki mesafeler ve coğrafi farklılıklar veri iletimini önemli şekilde etkileyen faktörlerdir. Her düğüm Mİ ile direk iletişim kuramayabilir ancak her düğümden toplanan verinin işlenmesi gerekebilir. O nedenle, düğümlerin öz-örgütlü bir şekilde organize olmaları ve ağı kurmaları, veri iletim hattını kurmaları gerekmektedir.

Bu yöntemi kullanan en popüler algoritma LEACH (Low Energy Adaptive Clustering Hierarchy) algoritmasıdır. LEACH'de, KB seçimi rastgele seçim yöntemiyle yapılır. Düğümler 0-1 arasında rastgele bir sayı üretirler. Önceden tanımlı bir eşik değer fonksiyonundan elde edilen eşik değerle karşılaştırılır. Rastgele üretilen sayı eşik değerinin altında kalırsa o düğüm kendini KB ilan eder ve ağa yayılır. Diğer bir algoritma ise HEED (Hybrid Energy-Efficient Distributed Clustering Approach) algoritmasıdır. HEED, LEACH 'e göre daha gelişmiştir ancak KB seçimini aynı şekilde yapmaktadır. Rastgele KB seçimi kendi içinde pek çok dezavantaja sahiptir. Bunlardan en önemlisi ise KB dağılımının ağda homojen olmama ihtimalidir. Bu ise enerji tüketimini ve ağın kapsama alanını olumsuz etkileyen bir faktördür.

Kaynaklarda, çalışmaların pek çoğunda ağın Mİ tarafından kümelendiği belirtilmektedir. Bu tür çalışmalarda düğüm maliyetleri göz ardı edilir ve ağı oluşturan düğümlerin tamamının Mİ ile haberleşebildikleri varsayılır. Çalışmalarda önerilen algoritmaların performans analizleri ise LEACH ile kıyaslanarak yapılmaktadır. Ancak bu, hatalı sonuçlar üretir çünkü LEACH’de düğümler kendi kendine organize olur ve kümelendir.

Bu tez çalışmasında, KAA’yı oluşturan algılayıcı birimlerin kendi içinde kümelmesini sağlayan yeni bir algoritma önerilmiştir. Önerilen, Merkez Eğilimli Kümeleme Algoritması (MEKA), RSSI (Receive Signal Strength Indicator) verisi kullanılarak düğümler arasındaki mesafenin kestirilmesi ve mesafeye dayalı bir kümeleme yöntemini benimser. MEKA’da KB seçimi homojen bir şekilde yapılır ve ağın kapsama alanı arttırılmış olur. Önerilen MEKA algoritmasının literatürdeki çalışmalardan farkı ise kümeleme işlemini merkez istasyondan bağımsız, öz-örgütlü bir şekilde yapıyor olması ve KB seçimini rastgele seçim yöntemiyle değil mesafeye dayalı bir şekilde yapıyor olmasıdır.

1.2. Tezin Organizasyonu

Tez çalışmasının 2. Bölümünde konuyla ilgili yapılan literatür taraması analiz edilerek sunulmuştur. Literatür taraması; ulusal-uluslararası indexli dergilerde yayınlanmış makaleler, bildiriler, yüksek lisans ve doktora tezleri üzerinde yapılmıştır. 3. Bölümde, KAA’nın temel yapı taşı olan algılayıcılar, KAA’daki veri iletim yöntemleri haberleşme ve veri iletim protokolleri detaylı bir şekilde anlatılmıştır. 4. Bölümde, tez kapsamında geliştirilen MEKA algoritması anlatılmış, 5. Bölümde ise araştırma sonuçları verilerek tartışılmıştır. 6. Bölümde sonuçlar ve önerilerden bahsedilmiş, son bölümde ise çalışmanın kaynakçası verilerek tez sonlandırılmıştır.

2. KAYNAK ARAŞTIRMASI

Kaynak araştırması, KAA yapıları için önerilmiş kümeleme algoritmaları ve enerji tüketimini düşürmek için yapılan çalışmalar odaklanarak yapılmıştır. Kaynaklar, ulusal/uluslararası çapta tamamlanmış tez çalışmaları ve akademik makaleler olmak üzere 2 ana başlık altında incelemeye alınmıştır. Kaynak analizi ise konu ve tarih sırasına uyularak verilmeye çalışılmıştır. Ulusal tez araması YÖK tez arama sayfasından yapılmış ve konumuzla ilgili tüm tez çalışmaları kaynakçaya alınmıştır. Yurtdışında yapılan tezlere ise UMULIB veri tabanı kanalıyla ulaşılmıştır. Akademik makaleler ise Science & Direct, Elsevier E-Book, IEEE, IEEE E-Book, Springer Link, Web of Science çevrimiçi veri tabanlarından taranmıştır. Araştırmada, aşağıdaki anahtar kelimeler kullanılmıştır.

Anahtar Kelimeler:

- Kablosuz algılayıcı ağlar
- Kablosuz sensör ağlarda kümeleme,
- Yönlendirme algoritmaları,
- Enerji verimli yönlendirme,
- KAA'da enerji verimi
- Kümeleme algoritmaları,
- Kümeleme,
- Kümeleme metotları,
- Kümeleme teknikleri,
- RSSI tabanlı kümeleme,
- Kablosuz sensör ağlar RSSI,
- RSSI kümeleme

Veri Tabanları:

- Science&Direct
- IEEE
- Web of Science
- Dergipark
- Google Akademik

2.1. Makale Çalışmaları

KAA'da kümeleme yöntemleri üzerine yapılan makale çalışmaları incelendiğinde, kümelemenin teknik olarak ortaya konulduğu Heinzelman ve ark. (2002) tarafından yapılan çalışma ve bu çalışmada önerilen LEACH algoritması öne çıkmaktadır. Sonraki yıllarda, bu algoritmanın avantajlı veya dezavantajlı olduğunun anlatıldığı ve farklı kümeleme metotlarının anlatıldığı algoritmalar geliştirilmiştir.

Heinzelman ve ark. tarafından yapılan başka bir makale çalışmada ise LEACH algoritmasının enerji verimliliği politikası incelenmiş ve LEACH algoritmasını baz alan farklı bir öneri yapılmıştır. LEACH algoritmasında, düğümler bir random sayı üretirler ve bu sayı belirlenen eşik değerinden düşükse o düğüm KB olur. Fakat KB olan düğümün kümedeki elemanlara olan uzaklıkları ve merkezi birime olan uzaklıkları homojen değildir bu nedenle KB olan düğümlerin enerji verimliliklerinin düşük olduğundan bahsedilir. Fakat bu çalışmada ise LEACH algoritmasından farklı olarak KB seçimi ağ içindeki düğümlerin merkezinde ya da merkezine en yakın olan düğümden seçilir. Bu seçim işlemi ise düğümler üzerinde bulunan GPS sayesinde yapılmaktadır (Heinzelman ve ark., 2002).

2001 yılında yapılan çalışmalarda ise TEEN ve APTEEN algoritmaları geliştirilmiştir. Bu algoritmaların LEACH algoritmasından farkı ise, verinin belli zaman aralıklarında değil algılamada farklılık oluştuğu zaman veri KB'ye aktarılması ve KB'den merkeze gönderilmesidir (Manjeshwar ve ark., 2002; Manjeshwar ve Agrawal, 2000).

2005 yılında yapılan bir makale çalışmasında önerilen BCDP (Base-Station Controlled Dynamic Clustering Protocol) algoritmasında, ağın merkezden yönlendirilerek kümelendiği bir yapı sunulmaktadır. Bu algortmada, ağ merkez tarafından kümeler halinde gruplandırılır. Kümedeki bir KB seçilir ve diğer üye düğümler algıladıkları veriyi KB'ye iletirler. KB ise merkez istasyona iletir. Bu algortmada KB seçimi ise şu şekilde yapılmaktadır, merkez tarafından kümelenen düğümler merkez istasyona enerji seviyelerini bildire bir mesaj gönderirler. Merkez istasyon ise tüm düğümlerden gelen enerji değerlerinin aritmetik ortalamasını alır ve bir eşik değeri bulur. Bu değerin üzerinde bir enerji seviyesinde bulunan düğüm KB olarak seçilmektedir. Ağın kümelere bölünme işlemi ise optimum KB sayısına ulaşıncaya kadar iterasyonlar halinde merkezden yönetilerek devam etmektedir. Fakat optimum sayının ne olduğuna ilişkin bir bilgi bulunamamıştır (Moruganathan ve ark., 2005).

2007 yılında yapılan bu makale çalışmasında, ağ farklı yetkilerde kümelere ayrılmıştır. En alttaki küme düğümlerden topladıkları verileri bir üst düğümdeki KB'ye o ise eğer varsa bir üstteki kümeye verisini ulaştırarak verinin merkeze gönderilmesi sağlanmaktadır. Ağın enerji verimliliği ise düğümlerin çalışmayan zaman dilimlerinde uyutulması temeline dayanmaktadır (Misić, 2007).

2008 yılında yapılan bu makale çalışmasında, küme içinde KB olacak düğümlerin seçildikten sonra bunların arasında haberleşme trafiğinin ya da bilgi yüklerinin optimum seviyeye getirilmesi için Load-Balanced Clustering Problem (LBCL) adında bir algoritma geliştirilmiştir. KB olan düğümlerin aralarındaki veri yükünün azaltılması hedef alınarak bir problem ortaya konulmuş ve bu problem NP-Hard tipinde tanımlanarak çözülmeye çalışılmıştır (Low ve ark., 2008).

Chen ve ark. (2008) yılında yaptıkları makale çalışmasında, kümelenmiş bir ağda, kümeye bağlı düğümlerden verilerin verimli şekilde toplanmasına yönelik bir araştırma yapmışlardır. Çalışmada vurgulanan olgu, veri tekrarlarının engellenmesi ve bu sayede hem enerji tasarrufu ve hem de efektif bir veri toplama sisteminden bahsedilmektedir.

2008 yılında yapılan diğer bir makale çalışmasında, KAA'nın ömrünün arttırılmasının için ağ içinde bulunan KB'lerin adil bir şekilde dağıtılması ve bu sayede ağ ömrünün arttırılması üzerine durulmuştur. LEACH algoritmasının küm başı bulma algoritması eleştirilmiş ve bu algoritma ile $1/p$ alanında tüm düğümlerin KB olma ihtimalinin bulunduğu bildirilmiştir. Çalışmada, mevcut algoritmaların yeniden düzenlenmesi önerilmiştir. Önerilen algoritmada farklı olan yönler, mevcut KB'nin kendinden sonra gelecek KB'yi belirleyebiliyor olmasıdır. Bu çalışmada mevcut algoritmaların güncellenmesi sureti ile ağın en optimum seviyeye getirilmesi önerilmektedir. Fakat KB seçiminin nasıl yapılacağı anlatılmamış KB'lerin var olduğu kabul edilmiştir (Kim ve ark., 2008).

Akkaya ve ark. (2009) yılında yaptıkları makale çalışmasında, düğümlerin alana yayılmadan önce KB düğümlerin belirlenmesi için Serbest hâkim küme (Independent Dominating Set (IDS)) tabanlı bir algoritma çalıştırılır ve KB'ler belirlenir. Daha sonra belirlenen KB'ler alanda belirlenen konumlara yerleştirilir ve bu şekilde ağ kurgulanır.

2009 yılında yapılan bu makale çalışmasında, mobil ağlar için Distributed Group Mobility Adaptive (DGMA) algoritması önerilmektedir. Bu algoritma ile kümenin ömrünün arttırılması amaçlanmaktadır. Çalışmada, düğümlerin üzerinde GPS konum belirleme sistemlerinin olduğu var sayılmış bu yolla mobil düğümlerin o anki

konumlarının belirlenmesi sağlanmıştır. Ayrıca düğümlerin aralarındaki mesafelerin belirlenebilmesi için RSSI verilerine bakılmıştır (Zhang ve ark., 2009).

2009'da Heinzelman ve Soro, ağ içinde bulunan KB'lerin kapsama alanı ve ağın yapısını bozmadan elimine edilerek azaltılmasına yönelik bir çalışma yapılmıştır. Önerilen algoritma, enerjisi azalan KB'lerin yerini enerji seviyesi daha yüksek düğümlere bırakması temelinde çalışmaktadır (Soro ve Heinzelman, 2009).

2010 yılında yapılan bu çalışmada ise, LEACH algoritması ile kümelenecek ve ayrışık olan KAA'da, küme içinde enerji verimliliği açısından en uygun veri iletim protokolünün belirlenmesine çalışılmıştır (Zhou ve ark., 2010).

Fazackerly (2010), RSSI değerleri kullanılması ağ içinde KB seçimine yönelik bir öneri sunulmuştur. Çalışma ile ilgili simülasyon çalışmaları yapılmış ve LEACH algoritması ile karşılaştırılmıştır. Önerilen çalışmada, düğümlerin KB seçilmeleri 6 adımda tanımlanmıştır. İlk olarak düğümler etrafında KB olup olmadığını kontrol ederek KB var ise ona bağlanmaktadır. İkinci adımda, mevcut KB yok ise kendi içlerinde bir timeout zamanı belirlemektedirler. Düğümler timeout zamanı içinde sürekli mesaj yayınlıyor ve cevap alırlar. Daha sonra KB olma durumları promotion factor olarak tanımlanır ve Denklem 2.1 ile hesaplanır.

$$P_{fact} = \begin{cases} \frac{e^n \sum_{i=1}^n RSSI_i}{n}, & \text{if } n > 0 \\ \infty, & \text{if } n = 0 \end{cases} \quad (2.1)$$

Burada n düğüm tarafından alınan cevapları, $RSSI_i$ değeri alınan en yüksek RSSI değerini temsil etmektedir. Kısaca düğümler etraflarındaki düğümlerden alınan tüm RSSI verilerinin ortalamasını bulurlar ve P_{fact} değerini elde ederler. Ayrıca, düğümlerde daha önceden belirlenmiş bir $RSSI_{thres}$ değeri belirlenmiştir. P_{fact} değeri $RSSI_{thres}$ değerinin üzerinde ise o düğüm KB olur.

2009 yılında yapılan bu çalışmada, kümeleme için düğümlerde kalan enerji miktarları kullanılmıştır. Başlangıçta tüm düğümlerin enerji seviyelerinin aynı oldukları varsayılmış ve düğümler random olarak dağıtılmıştır. Düğümlerin enerji miktarları baz alınarak KB seçimi yapılmıştır (Kumar ve ark., 2009).

2010 ve 2011 yılında yapılan bu makalelerde, teorik olarak LEACH, MLEACH (Multi-hop LEACH), HEED ve UCR (Unequal Cluster Based Routing) algoritmalarının ağ ömürlerinin belirlenmesi için performans analizleri yapılmıştır. Bu çalışmada, diğer çalışmalardan farklı olarak ilk başta bazı düğümlerin ölü olabileceği ve hepsinin de ölü

olabileceği hesaba katılarak analizler yapılmıştır. Aynı zamanda, ağ farklı yerlerde ve farklı zamanlarda test edilmiş ve sonuçları alınmıştır. Çalışmada Omnet++ yazılımı kullanılarak simülasyon çalışmaları yapılmıştır (An-Feng ve ark., 2011; Liu ve ark., 2011; Khalil ve Attea, 2011).

2010 yılında yapılan diğer bir makalede, kümelerin oluşturulması ve KB seçimi işlemi ise PSO (Partical Swarm Optimization) metodu ile yapıldığı EBUC (Energy-Balanced Unequal Clustering) protokolü önerilmiştir. Çalışmada, KB'ler arasında multi-hop bir yapının kurulduğu ve KB olan düğümlerin verilerini başka bir KB'ye iletirken seçeceği KB'yi Mİ 'ye en yakın olan düğüm seçildiği bir yapıdan bahsedilmektedir (Chang ve ark., 2010).

2011 yılında yapılan diğer bir makalede, data transferini rahatlatan ve düğümlerin enerji verimliliklerini arttıran bir kümeleme yaklaşımından bahsedilmiştir. Önerilen metotta, KB seçimi şu şekilde yapılmaktadır; Her bir düğüm KB olma potansiyelindedir ve içinde 0-1 arasında rastgele bir sayı üretir ve bu sayı o düğümün KB olduğunu ilan etmesi için gerekli olan bekleme süresi (delay time) ifade etmektedir. Eğer düğüm bu bekleme zamanında başka bir düğümden KB olma ilanını alırsa o düğüme biat eder ve ona katılır. KB yayını almayan ve süresi dolan düğüm kendi durumunu ilan eder ve kendini KB tayin eder (Chuang ve ark., 2011).

2011 yılında yapılan bu makalede, ağın düzenli parçalara bölünmesi suretiyle kümelere ayrılması ve her bölümden bir KB seçilmesiyle ilgili bir algoritma önerilmiş ve ağın enerji verimliliği gözlemlenmiştir. Önerilen algoritma LEACH, EECS ve HEED algoritmaları ile performans açısından kıyaslanmış ve daha iyi bir performans sergilediği bildirilmiştir (Aslam, 2011).

Yu ve Xiaohui (2011) tarafından yapılan bu çalışmada, PSO algoritması kullanılarak bir kümeleme modeli önerilmiştir. Bu algoritmada KB'lerin ağın merkezinde bulunan düğümlerden seçildiği kabul edilir.

2011 yılında yapılan bu makalede, ağa yayılan tüm düğümler merkeze enerji seviyelerini ve GPS yardımıyla almış oldukları koordinat bilgilerini gönderir. Merkez tarafından alınan bu bilgilere göre ortalama bir enerji seviyesi belirler. Bu seviyenin üzerinde bulunan düğümleri KB olarak seçer ve ona göre kümeleme işlemi merkez tarafından yapılır. Kümeleme bilgileri merkez tarafından tüm ağa yayılır ve kümeleme tamamlanmış olur. Bu kümeleme algoritması ADRP (Adaptive Decentralized Re-Clustering Protocol) olarak adlandırılmaktadır (Bajaber ve Awan, 2011).

2011 yılında yapılan bu makalede, LLACA (Localized Learning Automata Based Clustering Algorithm) algoritmasından bahsedilmektedir. Bu algoritmada düğümler içinde kendi kendine karar verebilen bir yapı bulunmakta ve bu yapı learning automata olarak adlandırılmaktadır. Burada düğümler tamamen rastgele bir şekilde yanındaki düğümlere bir mesaj yayınlarlar ve her düğüm kendine gelen mesaja küme içindeki seviyesini bildiren bir mesaj yayınlarlar. Bahsedilen algoritmaya göre küme seviyesi (cluster degree) eşik seviyesinin (threshold value) altında ise o düğüm KB olarak kabul edilir ve o KB'ye bağlanmak isteyen düğümlerin bağlanması sureti ile ağ kümelenir (Torkestani ve Meybodi, 2011).

KAA'larda Social Insect Colonies tabanlı bir kümeleme algoritmasının önerildiği bu makalede, KB seçimi şu şekilde yapılmaktadır. Düğümler arasında enerji seviyesi en yüksek olan düğümler kendileri backoff time (vazgeçme zamanı) belirlerler. Bu zaman dilimi batarya durumuna göre belirlenmektedir. Yani Batarya seviyesi yüksek olan düğüm daha kısa bir zamana sahiptir. Bu zaman diliminde kendisine başka bir düğümden KB mesajı gelmeyen düğüm kendi mesajını yayınlar ve kendini KB ilan eder. KB ile düğümler arasındaki mesafenin kestirilmesi amacıyla RSSI verilerinin kullanıldığından bahsedilmiştir (Cheng ve ark., 2011).

Amini ve ark. (2011) tarafından yapılan bu makalede, KAA'ları enerji verimliliği ve kapsama alanı açısından incelenmiş ve DBS (Distance-Based Segmentation) algoritması önerilmiştir. Bu algoritmada, ağ alanı eşit parçalara ayrılmış ve her alanda farklı kümeleme politikaları kullanılarak ağın ve düğümlerin enerji kullanımlarının ayarlanmasına çalışılmıştır.

2012 yılında yapılan bu çalışmada yazarlar Mİ 'ye yakın olan düğümlerin veya KB'lerin daha az enerji tüketmeleri için küme içindeki düğüm sayılarının düşürülmesine yönelik bir çalışma yapmışlardır. Sonuç olarak LEACH, Mr-LEACH ve BCDCP algoritmalarından daha efektif bir çözüm sunduklarını bildirmişlerdir (Lai ve ark., 2012).

2012 yılında yapılan bu çalışmada, ağ içindeki düğümlerin enerji harcamalarının ayarlanması üzerine bir çalışma yapılmış ve EBCAG (Energy-Balancing Unequal Clustering Approach for Gradient-based routing) algoritması önerilmiştir. Bu algoritmada KB seçimi ise düğümlerin kalan enerji miktarları üzerinden yapılmaktadır (Liu ve ark., 2012).

Wanga ve ark. (2012) Tarafından yapılan bu çalışmada ise, LEACH algoritmasının, düğümlerin enerji seviyelerinin ve KB seçiminde bu parametrenin de hesaba katılması ile güncellenen bir çalışma yapılmıştır. Çalışmada, ağ içinde KB

seçiminin daha verimli bir hale geldiği bunun yanında enerji seviyesi bilgisinin düğümler arasında dolaşması nedeni ile %2,5 oranında ağa yük getirdiği bildirilmiştir.

2012 yılında yapılan başka bir çalışmada ise ağ ömrünün uzatılması için düğümlerin veri iletimi esnasında harcadıkları gücün minimize edilmesi için RSSI tabanlı DPC-EERRP (Distributed Power Control with Energy Efficient and Reliable Routing Protocol) ismiyle bir algoritma önerilmiştir. Önerilen algoritmada, düğümlerin verilerini merkeze ulaştırmada RSSI değerlerinden ve aralarındaki link kalitelerine bakılarak uygun yolun bulunmasına yönelik bir öneridir. Simülasyon çalışmalarında önerilen algoritma ile AEERRP (Adaptive Energy Efficient Forwarding Routing) algoritmaları kıyaslanmıştır. Sonuç olarak önerilen algoritmanın diğerine göre daha iyi bir performans sergilediği bildirilmiştir (Mathapati ve ark., 2012).

2012 yılında yapılan bu makalede, k-means algoritması kullanılarak çok parametrelili bir kümeleme algoritmasının MPC (Multiple Parameter based Clustering) olabilirliğine dair uygulamalı bir çalışma yapılmıştır. Her bir düğümün enerji yoğunlukları, verilerin aktarılması yaşanan gecikme değerleri ve düğümlerin merkeze olan uzaklıkları ölçüm parametreleri olarak alınmıştır. Bu parametreler kendi aralarında 2ⁿ kombinasyonunda NP karmaşıklık problemi olarak ele alınmıştır (Khan ve ark., 2012).

Fu ve ark. (2012) tarafından yapılan bu makalede, AD'lerin iletim esnasında harcadıkları gücün kontrolüne yönelik (transmission power control) pratik güç yönetim algoritması (Practical- Transmission Power Control) önermişlerdir. Bu algoritma ile farklı link kalitelerinde farklı güçler uygulanarak algoritmanın enerji verimliliğine bakılmıştır.

Yuea (2012) yaptığı çalışmada, KAA'larda kümelemenin ağın enerji verimliliği açısından ne kadar önemli olduğunu açıklamış ve EEBCDA (Energy Efficient And Balanced Cluster-Based Data Aggregation Algorithm) algoritmasını önermiştir. Bu algoritma temelde şu şekilde çalışmaktadır; Ağ dikdörtgenler şeklinde bölümlere ayrılır ve KB olarak kabul edilen düğümler kendi bölgelerinde tüm düğümleri dolaşır ve verilerini toplayarak merkeze iletmektedirler.

Geetha ve ark., (2012) yılında yaptıkları çalışmada, LEACH ve LEACH-C (Centralized) algoritmalarının karşılaştırmalı olarak incelemişlerdir. Sonuç olarak LEACH algoritmasının, merkez istasyonla haberleşmeye öncelik tanınmadan küme içindeki düğümlerin yerleşimleri koordine edildiğinde yani merkez istasyona aşırı yüklenme yapmadan tercih edilebilir olduğundan bahsedilmektedir. Örneğin merkez istasyon KB seçimleri ile uğraşmamaktadır. Ama düzenli ve merkezi bir ağ kurulmak

isteniyorsa LEACH-C algoritmasının daha verimli olduğu fakat burada da merkez istasyona ekstra yük getirmesinin de bir dezavantaj olduğundan bahsedilmektedir.

Kümedeki düğüm sayılarının optimizasyonu da son yıllarda üzerinde durulan konular arasında yer almaktadır. 2012 yılında yapılan makalelerde farklı yöntemler kullanılarak kümedeki düğüm sayılarının optimizasyonuna çalışılmıştır. Çalışmalarda, KAA'da enerji verimliliğinin önemli olduğu ve bu problemin çözümünde kümeleme algoritmalarının katkısının bulunduğu sonucuna varmışlardır.

Amini ve ark. (2012), LEACH, LEACH-C ve DBS algoritmaları incelenmiş ve kümelerin boyutlarını optimizasyonu üzerinde durulmuştur. Optimizasyonda, KB sayının ayarlanması gerektiği yani, KB sayısı optimumun altında olursa düğümlerin datalarını daha uzağa göndermek zorunda kalacakları ve daha fazla enerji harcayacakları, fazla olursa da aynı şekilde merkezle iletişim kuracak olan düğüm sayısının artacağı dolayısı ile her iki durumunda istenmediğini belirtmiştir. Ayrıca çalışmada, ana istasyonun ağın farklı noktalarında bulunması durumları incelenmiştir. Burada merkez istasyon ile düğümlerin aralarında mesafenin RSSI değerleri baz alınarak kestirildiği belirtilmiştir. Ağın ve düğümlerin enerji durumları ile ilgili ise düğüm tasarımında kullanılan devre yapısının önemli olduğu vurgulanmıştır. Örneğin Xbee firmasına ait düğümlerin göndermede almaya göre 6 kat fazla enerji harcadığı, crosbow firmasının düğümlerinde ise bu oranın 2,5 katı olduğu vurgulanmıştır.

Dutta ve ark., ise istatistik metotları kullanılarak küme içindeki düğüm sayılarının optimizasyonu hakkında çalışmışlardır (Dutta ve ark., 2012).

2012 yılında yapılan diğer bir makalede, DEECIC (Distributed Energy-Efficient Clustering with Improved Coverage) algoritması geliştirilmiştir. Algoritmada, ağın n sayıda KB ile temsil edilmesi ve tek bir ID kullanması amaçlanmış, kümenin enerji seviyelerine göre de kendini yenileyebildiği bildirilmiştir. Ağın kümelenmesi süresince Mİ aktiftir ve tüm düğümlere 4 byte'lık paketler halinde mesaj gönderir. Bu paketleri alabilen düğümler aldıkları paket sayısına göre kendilerine bir derece belirlerler. Derecesi yüksek olan düğümün KB olma ihtimalinin yüksek olması prensibi ile KB seçimi yapılır (Liu ve ark., 2012).

Rocha ve arkadaşlarının yaptığı bu makale çalışmasında ise anlamsal (semantic) bir kümeleme metodundan bahsedilmiştir. Semantic yaklaşımda, algılanmak istenen verinin doğru algılanıp algılanmadığı kaygısı göz önüne alınmış ve buna göre bir kümeleme yapılması gerektiğinden bahsedilmiştir. Buna göre, örneğin ormanda kullanılan bir KAA uygulamasında, algılanacak veri sıcaklıktır fakat her zaman sensörler

bu veriyi doğru algılamamaktadır. Yani düğümlerde, 50 derecenin üzerinde veri alırsan ilet denir ama sensör 49,9 olarak algılsa veriyi göndermez. Bu durumun önüne geçilmek için ilgili bölgede bulunan düğümlerin kendi aralarında iş birliği yaparak veri toplaması ve alınan verilerin bulanık mantık kontrollerine sokularak verinin analizi, daha sonra da bu düğümlerin kendi aralarında kümelenmesi temeline dayanmaktadır (Rocha ve ark., 2012).

2012 yılında yapılan bu makalede, DECSA (Distance-Energy Cluster Structure Algorithm) algoritmasını açıklamıştır. Algoritma, mesafe ve enerji açısından kümeleme yapılmasına olanak sağlayan bir algoritmadır. Burada kullanılan kümeleme ve KB seçiminde LEACH modeli kullanılmıştır. LEACH'den farklı olarak, KB düğümlerin merkeze en yakın olan düğümlerden seçilmesidir. KB seçiminde kullanılan metot ise rastgele seçimliliklidir (Yonga ve Peia, 2012).

2012 yılında yapılan bu makalede, birbiri ile örtüşen kümelerden oluşan bir kümeleme yapısından bahsedilmiştir. Multi-Hop Clustering Algorithm (KOCA) algoritmasının diğer geleneksel kümeleme metotlarına göre daha verimli olduğu ve bu yöntemle KB'nin kapsama alanının daha da artacağından bahsedilmiştir. Burada farklı olan kısım, düğümler birden fazla kümeye üye olabilirler. Bu düğümler KB'ler arasında bir gateway olarak görev yaparlar (Jia-qi ve ark., 2012).

2013 yılında yapılan bu makalede, KAA'da bir kümeleme algoritmasından ve mobil veri toplama sisteminde bahsedilmektedir. Çalışmada, kümelenmiş olan ağ içerisinde KB olan düğümlerin mobil bir araç ile dolaşılıp en az veri kaybı ile kümelerdeki verilerin toplanması temeline dayanmaktadır. Burada aracın KB'leri gezerken izleyeceği yolu ise NP hard problemi olarak görülmüştür. Ayrıca, çalışmada kümeleme şu şekilde yapılmaktadır; Düğümler komşu düğümler ile veri paketi gönderirler ve bu sayede aralarında bir link kalitesi değeri bulurlar. Link kalitesi 0-1 arasında bir ağırlık değeri olarak kabul edilir ve "selection weight" olarak adlandırılmıştır. Ağırlığı düşük olan düğüm KB olma mesajı yayınlar. Eğer etrafında KB olan bir düğüm var ise ona katılır yok ise kendisini KB ilan eder. Çalışmada link kalitesinin ölçümü için alternatif olarak RSSI verisinin de kullanılabileceğinden bahsedilmektedir (Gong ve ark., 2013).

Gu ve ark. (2013), ağın ve kümelerin kapsama alanı probleminin yeterince üzerine düşülmediği ve bu problemin de önemli bir problem olduğunu vurgulayan bir makale çalışması yapmışlar ve ECDC (Energy and Coverage-aware Distributed Clustering Protocol) protokolünü önermişlerdir. Yapılan teorik çalışmalar ve simülasyonlar sonucunda düğümlerin enerji verimliliklerinin arttırıldığı ve ağ ömrünün uzatıldığından

bahsedilmiştir. Kümeleme algoritması ve KB seçme algoritması olarak LEACH algoritması kullanılmıştır. Farklı olarak düğümlerin kalan enerji miktarları göz önüne alınmıştır.

Kuila ve arkadaşları (2013), KAA’da kümeleme algoritmalarının önemli olduğu fakat KB durumunda olan AD’lerin veri toplama, irdeleme ve bu verileri merkeze iletme gibi görevlerinden dolayı aşırı yüklü olduklarından bahsetmişlerdir. Bu tip sensör ağlarda KB’lerin yük dengelerinin incelenmesi konusunda bir çalışma yapmışlardır. Bu sorunun NP-hard cinsinden bir problem olduğu o şekilde çözülebileceğinin bildirmişlerdir. Fakat bu çalışmada, bu problemin çözümünde genetik algoritmalar kullanılarak da çözülebileceği hakkında yeni bir çözüm önerisi getirilmiştir. Önerilen çalışma literatürde bulunan benzer algoritmalarla kıyaslanmış ve daha iyi bir performans sergilediği bildirilmiştir.

Krunz ve arkadaşları 2013 yılında yaptıkları çalışmalarında, KAA’da kümeleme ve enerji verimliliği açısından yeni bir algoritma önermişlerdir. Bu algortmada, kümeleme şu şekilde yapılmaktadır; Düğümler etrafındaki düğümlere “Hello” mesajı gönderirler. Bu mesajı alan düğümler karşı bir cevap ile birlikte batarya seviye bilgilerini gönderirler. Batarya seviyesi en yüksek olan düğüm kendisini KB ilan eder. Bunun yanında KB olan düğüm yardımcı bir düğüm seçimi için kendi kümesine davetiye gönderir. Bu davetiyeyi alan üyeler bir süre beklerler ve davetiyeyi kabul eden düğüm yardımcı KB olur (Krunz ve ark., 2013).

Hacıoğlu ve ark. (2016) çok parametrelili bir kümeleme algoritması önermişlerdir. Önerilen algoritma LEACH algoritması ile karşılaştırılmıştır. Ancak çalışmada, merkezi kontrol ile kümeleme yapıldığı, düğümlerin yerlerinin bilindiği ve her düğümün uzak mesafeli haberleşme yapabildikleri varsayılmıştır. Bu varsayımlarla, önerdikleri algoritmanın LEACH’den daha iyi sonuçlar ürettiği bildirilmiştir.

2018 yılında Zahedi, homojen dağılmış KAA modelleri için bir kümeleme algoritması önermişlerdir. KB seçimi rastgele seçim metodu ile yapılmış ve önerilen algoritma LEACH ve Heed algoritmaları ile kıyaslanmıştır (Zahedi, 2018).

2018 yılında Su ve Zhao tarafından yapılan makalede, Fuzzy-C tabanlı bir kümeleme algoritması önerilmiştir. Önerilen algoritma, ağda düzensiz dağıtılmış olan düğümler merkezlenerek kümeleyen ve KB sayısını optimize eden bir yaklaşım sunmaktadır. Ancak, ağ içinde yayılmış olan düğümlerin konum bilgilerinin GPS ile alındığı varsayılmıştır. Bu durum ise algoritmanın maliyet ve enerji verimliliği açısından uygulanabilirliğini düşürmektedir (Su ve Zhao, 2018).

2018’de yapılan diğerk bir makalede ise düğümlemler konum ve enerji seviyeleri göz önüne alınarak bir kümeleme algoritması önerilmiştir. Her düğümün konumunun bilindiğı ve Mİ ile haberleşebildiğı var sayılmıştır. Önerilen algoritma LEACH ile kıyaslanmıştır (Panag ve ark., 2018).

Kanmaz ve Aydın (2019) tarafından yapılan bir makalede ise iç mekânda hareketli cisimlerin konumlarının tespiti üzerine bir çalışma yapılmıştır. Çalışmada, ağda rastgele yayılan düğümlemler kümeleme yöntemi ile sınıflandırıldıkları ve konum tespitindeki hata oranının en düşük seviyeye ulaştığı bildirilmiştir.

Gupta ve ark. (2019) tarafından yapılan bu makalede, rastgele KB seçim prensibinden farklı olarak düğümlemler kalan enerjilerinin dikkate alındığı Modified Intelligent CH election based on Bacterial foraging optimization algorithm (M-ICHB) algoritması önerilmiştir. Önerilen algoritma LEACH algoritması ve türevleri ile karşılaştırılmış ve ağdaki son düğümün LEACH algoritmasında 1201.iterasyonda, önerilen algoritmada ise 3291.iterasyonda öldüğü bildirilmiştir.

Goswami ve ark. (2019) yılında yapılan bu makalede, optik alıcılardan oluşan sensör düğümlemlerinin oluşturdukları KAA’da Firefly algoritması ve Hierarchical Maximum Likelihood (HML) algoritmalarından türetilmiş yeni bir kümeleme algoritmasından bahsedilmiştir. Aynı zamanda birer ışık kaynağı olan algılayıcı düğümlemler yaydığı ışınım, Mİ tarafından algılanarak aradaki yansımanın şiddetinden düğümlemler Mİ’ye olan mesafeleri hesaplanır ve ağın bu şekilde kümeleneşmesi sağlanmış olur.

Hamzah ve ark. (2019) yılında yapılan bu makalede, Fuzzy Logic-based Energy-Efficient Clustering (FL-EEC/D) algoritmasını önermişlerdir. Önerilen algoritma LEACH, k-means algoritmaları ile kıyaslanmıştır. Bildirilen sonuçlara göre, düğümlemlerinin yarısının öldüğü iterasyon sayısı bakımından LEACH ve k-means sırasıyla 129. ve 145. iterasyonlarda ölürken FL-EEC/D algoritmasında 187.iterasyonda öldüğü görülmüştür.

Bu makalelerde, ağdaki küme sayılarının ve kümedeki düğüm sayılarının optimizasyonu için öneriler sunulmuştur. Ardından, enerji tasarruflu sıkıştırılmalı algılama tabanlı kümeleme yönlendirme (EECSR) protokolü ayrıntılı olarak sunulmuştur. Algoritmada, alan katmanlara ayrılır her katmanda bir KB olmasına dikkat edilir. Mİ, ağın bilgilerine göre KB düğümlemleri seçer ve KB düğümlemleri alana yayınlar. Yayını alan düğüm KB düğüm ID’siyle “HELLO” mesajı yayınlar. Düğüm, KB düğümün kapsama alanına giriyorsa KB’ye dahil olur. Yazarlar yaptıkları diğerk bir çalışmada ise mevcut

yapıya ek olarak katmanda yedek KB belirleyerek ağın enerji tüketimine olumlu katkıda buldukları bildirilmiştir (Wang ve ark., 2019; Lin ve Wang, 2019).

Bu makalede, KAA'da verilerin toplandığı KB düğümlerde bu verilerin Mİ'ye iletilmesi için akıllı bir veri birleştirme (data fusion) algoritması ve buna bağlı olarak yeni bir KB seçim algoritması önerilmektedir. KB seçim algoritmasında Mİ'ye yakın düğümlerin KB seçilmesi ve buna bağlı olarak veri birleştirmenin getireceği enerji yükünün azaltılması hedeflenmektedir (Liu ve ark.,2020).

Osamy ve ark. (2020) tarafından yapılan bu makalede, düzensiz yapıdaki KAA modellerinde, düğümlerin kalan enerji miktarları ve Mİ'ye olan mesafelerine göre öz-örgütlü yapıda kümeleme yapan Entropy based clustering algorithm (EBCS) algoritmasından bahsedilmiştir. Önerilen algoritma, DEC, EDDEEC, MEDDEEC, SEP-E, CREEP ve DMSEP algoritmaları ile kıyaslanmış ve daha iyi sonuçlar ürettiği bildirilmiştir.

2020'de yapılan bu çalışmada, Hybrid Approach of Firefly Algorithm with Particle Swarm Optimization (HFAPSO) algoritması önerilmiş ve sonuçları LEACH-C algoritması ile karşılaştırılmıştır. Sonuçlara göre ağdaki son düğümün LEACH-C algoritmasında 550.iterasyonda öldüğü, HFAPSO algoritmasında ise 1550. İterasyonda öldüğü bildirilmiştir (Pitchaimanickam ve Murugaboopathi, 2020).

Tez çalışması ile dolaylı olarak ilgili olabilecek çalışmalar da incelenmiş ve analiz edilmiştir. Buna göre 2003, 2006 ve 2011 yıllarında yapılan çalışmalarda, ise MicaZ AD'ler ve TinyOS işletim sisteminin kullanıldığı uygulamalar yapılmıştır. Simülasyon çalışmaları ise Matlab ve farklı simülasyon programları kullanılarak yapılmıştır (Musznicki ve Zwierzykowski, 2012; Jens ve ark., 2006; Ali ve ark., 2011; Levis ve ark., 2003).

2012 yılında yapılan bu çalışmada, geniş bir alanda çalışacağı düşünülen bir KAA'da, ölçülen değerlerin sürekli olarak aktarılmasından ziyade, bir anormallik olması durumunda sistemin aktive edildiği ve anormallik olan bölgelerin mesafe ölçümleri baz alınarak kümelenin yapıldığı bir sistemden bahsedilmektedir (Devia ve ark., 2012).

Enerji verimliliğinde önemli olan değişkenlerden birisi almaç/göndermeçlerin çıkış güçlerinin ayarlanabilir olmasıdır. 2007 yılında yapılan bu çalışmada, yazarlar düğümler üzerinde bulunan kablosuz haberleşme ekipmanlarının gönderme anında harcadıkları gücün kontrolüne yönelik yeni bir öneri sunmuşlardır. Önerilen bu yeni yöntem B-MAC metoduyla performans açısından karşılaştırılmıştır (Correia ve ark., 2007).

2008 yılında yapılan benzer bir makalede, düğümlerin gönderme sırasında kullanmış olduğu güç tüketiminin düşürülmesine yönelik bir çalışma yapmışlar ve bir algoritma sunmuşlardır. Farklı güç değerlerindeki link kaliteleri ve veri kayıpları analiz edilmiştir. Bu işlem yapılırken komşuların aralarındaki RSSI seviyelerinden faydalanılmıştır (Sheu ve ark., 2008).

Bu tez çalışmasında kullanılan temel parametrelerden birisi olan düğümler arası mesafelerin belirlenebilmesi için yapılan çalışmalarda kaynak analizine eklenmiştir. 2008 yılında yapılan bu çalışmada Adaptif Neuro-Fuzzy Inference System (ANFIS) kullanılarak KAA içinde kullanılan düğümlerin aralarındaki mesafenin ölçülmesine yönelik bir bildiri yayınlanmıştır. Bildiride, RSSI seviyeleri baz alınarak ANFIS eğitilmiş ve sonuç olarak mesafe değerlerine ulaşılmaya çalışılmıştır (Feng ve ark., 2008).

2010 yılında yapılan bu çalışmada, 802.15.4 standardında çalışan CC2420 Zigbee RF Alıcı-Verici ailesinin kullanıldığı bir uygulama yapılmış ve bu uygulama ile ağ içinde konumu belli olmayan düğümlere ilişkin mesafe kestirim çalışması yapılmıştır. Uygulamada RF alıcı-vericilerin RSSI değerlerinden faydalanılmıştır. Sonuç olarak, düşük hata payı ile kestirim yapıldığı bildirilmektedir (Faheem ve ark., 2010).

2011 yılında yapılan diğer bir çalışmada ise, RSSI seviyesi parametre alınarak ağ içindeki düğümlerin konumlandırılmalarına yönelik bir çalışma yapılmıştır. Düğümlerin ağ içindeki konumlarının tespitinde lineer regresyon ve yol kaybı (Path-loss) metotları kullanılarak, konumu bilinmeyen düğüm ile sunucu düğüm arasındaki mesafe kestirilmeye çalışılmıştır. Çalışmanın uygulaması kapalı ve açık alan sensör ağlarda denenmiş, açık alanda yapılan uygulamada konum kestiriminin kapalı alana göre daha doğru olduğu sonucuna ulaşılmıştır (Alawi, 2011).

2012 yılında yapılan bir çalışma, KAA'larda konumu bilinmeyen düğümlerin konumlarının belirlenmesi için RSSI seviyene göre mesafe kestiriminin en popüler metot olduğundan bahsedilmiştir. Fakat bu yönteminde tam olarak doğru sonuç vermediğine değinildiği genel analiz tarzında yapılmış bir çalışmadır. Analiz 250 düğümün olduğu bir ağda yapılmış ve RSSI seviyesine alternatif olarak Spring-Relaxation algoritması ile karşılaştırmalar yapılmıştır (Heurtefeux ve Velois, 2012).

2.2. Yüksek Lisans Tezleri

Malik, 2003 yılında yaptığı yüksek lisans tezinde, KAA kullanılarak hedef izleme algoritması geliştirmiştir. Ağ içerisinde hedefi algılayan 3 AD'den hedefin 4. Düzüme olan mesafesi matematiksel modellerle tespit edilmeye çalışılmıştır (Malik, 2005).

Yılmaz'ın 2007 yılında yaptığı yüksek lisans tezinde, temel rota tespit yöntemleri ve doluluk boşluk algoritmaları incelenmiştir. Doluluk boşluk algoritmalarından biri olan RACP algoritması incelenmiş ve ns2 kullanılarak simüle edilmiştir. Yeni bir doluluk boşluk algoritması SIFP önerilmiş ve RACP ile, uyuyan algılayıcı oranı ve kapsama alanının zaman üzerinden ortalaması bakımından karşılaştırılmıştır (Yılmaz, 2007).

2007 yılında Ayran tarafından yapılan yüksek lisans tez çalışmasında, Telsiz Sensör Ağları (TSA) için tasarlanmış olan taşıma katman protokollerinin çoğul ortam iletişimindeki başarımları incelenmiştir. Mevcut protokollerin, paket aktarım hızı uçtan uca paket gecikmesi, bant ve enerji kullanım verimliliği çerçeve sinyal gürültü oran tepe değeri PSNR, zaman sınırlı çerçeve PSNR'si, başarılı çerçeve aktarım olasılığı, uçtan uca çerçeve gecikmesi ve seğirmesi gibi basarım metrikleri açısından çok düşük basarım sergiledikleri gözlemlenmiştir (Ayran, 2007).

Karasabun (2009), yüksek lisans tezinde aktif sensor aygıtlarının belirlenebilmesi için etkin ve hızlı çalışan tekrarlamalı iyileştirme gerçekleştiren buluşsal IIIH algoritmasını önermektedir. İkinci olarak ise aktif sensor aygıt kümesine seçilen sensor aygıt sayısı azaltılırken, bu kümeye seçilen sensor aygıtlarının yüksek enerjiye sahip olabilmelerine imkân veren bir yarar fonksiyonu önermektedir. Detaylı simülasyonlarla ileri sürdükleri bu yaklaşımın hem kablosuz sensor ağının işleme süresi bakımından, hem de algoritma çalışma zaman bakımından iyi sonuçlar ortaya koyduğu görülmektedir.

Çetin (2009), sunduğu yüksek lisans tezinde, kablosuz sensor ağlarının biyomedikal bir uygulamasını gerçekleştirmiş ve geliştirilen sistem Ege Üniversitesi Hastanesi'nde denenmiştir. Kablosuz modüller (mote) nesC diliyle programlanmış, pulse oximeter algılayıcılar bu modüllere bağlanarak hastaların nabız, pletismogram ve kandaki oksijen oranı verileri ZigBee standardı kullanılarak kablosuz ağ üzerinden merkezi veri tabanına aktarılmıştır. Sistemin performansı, değişik ağ topolojilerinde, paket kaybı yüzdesi olarak ölçülmüştür.

Taş (2010), yaptığı yüksek lisans tez çalışmasında, KAA'nın karasal hudut güvenliğinde kullanılmasına yönelik bir öneri getirilmiş ve sistem tasarımı yapılmıştır fakat sistemin gerçekleştirilmesi yapılmamıştır.

Seçkin (2011), yüksek lisans tezinde KAA düğümleri ile IP tabanlı ağlar arasında haberleşme yapılıp internete girilebilin her noktadan sensör ağı düğümlerine ulaşılabilen bir sistem geliştirilmeye çalışılmıştır. Tasarlanan sistem ile sunucu tabanlı yöntem kullanılarak ağlar arası haberleşmede ZigBee standardının avantajlarının korunması sağlanmıştır. Ayrıca sunucu tabanlı sistemlerde klasik web sunucusu yerine XMPP sunucusunun kullanılabilirliği sınanmıştır.

İbrahim (2011) yüksek lisans çalışmasında, KAA hakkında genel bilgi, donanım ve Ad'lerin ağ mimarisi, KAA uygulamaları, kümeleme ve yönlendirme protokollerine değinilmiş ve bazı iyi bilinen enerji verimli yönlendirme protokolleri arasında derin ve entegre bir karşılaştırma yapıldıktan sonra ÇYVL Dedikodu protokolü adlı yeni bir yönlendirme protokolü önerilmiştir.

Özcan (2011) tarafından yapılan yüksek lisans tezinde, AD tasarımı yapılmıştır. Yapılan tasarımda Microchip firmasına ait “nanoWatt Technology” destekli PIC’ ler hem hafıza hem de işlem birimi olarak kullanılmışlardır. Kablosuz iletişim birimi olarak da yerli bir firma olan UDEA firmasına ait kablosuz iletişim birimleri kullanılmıştır.

2.3. Doktora Tezleri

Elson (2003) tarafından yapılan doktora çalışmasında, farklı senkronizasyon tekniklerinin uygulama şekilleri ve karakterizasyonları ve bunların örneklendirilmesi yapılmıştır.

- Reference-Broadcast Senkronizasyon
- Post-Facto Senkronizasyon

Sonuç olarak, Reference-Broadcast Senkronizasyon yönteminde, sensör ağı içindeki kablosuz haberleşmede düşük enerji tüketiminde de yüksek hassasiyet elde edilebileceği sonucuna ulaşılmıştır.

2003 yılında Hill tarafından yapılan doktora çalışmasında, KAA ihtiyacını karşılamak için geliştirilen 3 farklı donanım platformu ve TinyOS adı verilen açık kaynak kodlu işletim sistemi ve sistem mimarisi üzerinde çalışılmıştır (Hill, 2003).

Shah (2007) doktora çalışmasında, sensör yerlerinin tespiti, ağ konfigürasyonu, veri birikimi, gerçek zaman veri iletimi, sensor-aktör/aktör-aktör koordinasyonu ve enerji tasarrufu mekanizmaları gibi konuları içeren tümleşik bir çerçeve önermiştir. Öneri, üç boyutta yeni yöntemler içermektedir;

- (1) Zamana dayalı TSL,
- (2) Yol verme algoritmaları,
- (3) Enerji tasarrufu.

TSL sensor düğümlerinin aktörlere göre yerlerini belirler. Dağıtık gerçek-zaman koordinasyon, düğüm ID ve bencil yol vermeye göre çalışmaktadır. O beklemeye bağlı gerçek-zaman koordinasyon ve yol verme protokolü, düğüm ID modunda çalışmaktadır. Sonuç olarak, Bencil yol verme yöntemi (RAT), durumsuz en kısa bir yol verme tekniğidir. RAT, yüksek yoğunluklu ağlarda enerji bakımından diğer yöntemlere göre daha iyi sonuç verdiği tespit edilmiştir.

Onur (2007) tarafından yapılan doktora çalışmasında, kablosuz algılayıcı ağların gözetim kalitesini analiz etmek için ölçütler önerilmekte ve konuşlandırılacak algılayıcı sayısı ile arasındaki ilişki incelenmektedir. En zayıf sızma yolu problemi tanımlanarak belli bir gözetim kalitesi için gerekli algılayıcı sayısını hesaplamak için yöntemler önerilmektedir.

Hussain (2008) doktora tez çalışmasında, KAA için ölçeklenebilir, enerji haberdar ve esnek bir iletim çözümü sunmuştur. Önerilen iletim çözümü, sensör-aktör ve aktör-sensör güvenilir iletim olmak üzere iki ana parçaya bölünmüştür. Olayların güvenilirlik gereksinimlerini karşılamak amacıyla sensör-aktör iletim daha sonra dört ayrı iletim moduna ayrılmıştır: basit, adil, önceliklendirilmiş ve gerçek zamanlı. Algılayıcılardan aktöre olay bilgilerinin aktarımındaki ani taleplerin tıkanıklığa neden olmasından dolayı, paket iletim süresi ve düğümlerin arabelleklerinin büyüklüğüne dayanan özgün bir tıkanıklık denetim planı da bu çalışmada sunulmuştur.

Tümer (2011) doktora tezinde, bina içi kablosuz algılayıcılar için enerji verimli yönlendirme protokolleri önermiştir. Çalışmada, ağın kümelere ayrılması ve verinin bu şekilde toplama merkezine ulaştırılması hedeflenmiştir. Küme başı seçimi ve ağın kümelemesi, LEACH algoritmasının prensiplerine göre yapılmıştır. İletim planlamasında bir farklılık ortaya konulmuş, algılayıcıların sürekli iletme geçmesi yerine algılanan verinin belirli bir eşiğin üstüne çıktığında iletme geçilmesi şeklinde planlanmıştır. Sonuç olarak LEACH algoritmasından daha iyi sonuçlar elde edildiği bildirilmiştir.

Karataş (2012) doktora tez çalışmasında, saha araması amacıyla kullanılan bistatik ve multistatik sualtı akustik algılayıcıların etkinliğinin analitik yöntemlerle tahmin edilebilmesi ve planlayıcılara referans oluşturması için çeşitli teorik formüller üzerinde çalışmış, farklı tipteki hedeflerin tespit olasılıklarının en büyüklenmesine

yönelik optimum sensör yerleşim planları oluşturulmuştur. Bistatik sistemler için yapılan çalışmada integral geometrisi ve geometri olasılığı teorilerinden faydalanılmış; multistatik sistemler için yapılan çalışmada Karma Tam Sayılı Doğrusal Olmayan Programlama (KTDOP) ve Genetik Algoritma (GA) teknikleri kullanılmıştır. Elde edilen tüm teorik sonuçlar Monte Carlo simülasyonları ile test edilmiş ve doğrulanmıştır.

Literatür analizi kablosuz algılayıcı ağlarda kümeleme algoritmaları, RSSI değerleri baz alınarak yapılmış çalışmalar ve düğümlerde kullanılan alıcı-vericilerin güç yönetimleri ile ilgili yapılan çalışmalardan oluşmaktadır. İncelenen makalelerde önerilen kümeleme algoritmaları, RSSI tabanlı yönlendirme ve bulma algoritmaları ve alıcı-verici güç yönetimi algoritmaları aşağıda maddeler halinde sıralanmıştır;

Kümeleme

- LEACH
- LEACH-C
- MLEACH
- HEED
- UCR
- MPC
- TEEN
- APTEEN
- BCDCP
- EEBCDA
- PSO

RSSI

- DPC-EERRP
- AEERRP
- CTA

Güç Yönetimi

- PTPC

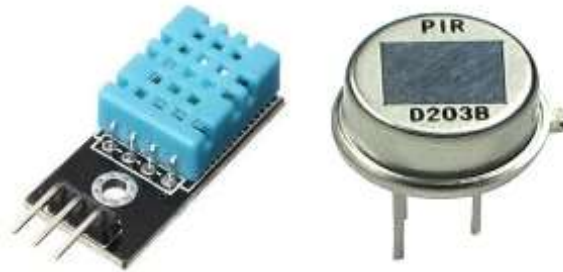
Kablosuz algılayıcı ağlarda kümeleme için önerilen algoritmalarda LEACH algoritması kümeleme mantığının ortaya ilk atıldığı bir çalışma olması dolayısı ile öne çıkmakta yapılan çalışmalar performanslarını bu algoritma ile kıyaslamaktadırlar. Önerilen diğer algoritmalar ise genel anlamda, ağın düzenli şekillerde bölümlere ayrılması ve bu bölümlere denk düşen düğümlerin kendi aralarında kümelenmesi temeline dayanmaktadır. Burada, en önemli nokta KB seçimidir. Seçimler, düğümlerin enerji yoğunlukları, merkezi istasyon tarafından seçme veya LEACH algoritmasında olduğu gibi rastgele seçim yöntemleri kullanılarak yapılmaktadır. Literatürdeki çalışmalarda seçimlerin Mİ tarafından yapılması durumunda; düğümlerin konumlarının, enerji seviyelerinin veya düğümlerin birbirlerine olan uzaklıklarının bilindikleri varsayılmaktadır. Ancak bu varsayımların her biri bir problemdir ve göz ardı edilemezler. Rastgele seçim modelinde ise KB'lerin homojen dağılımları ihtimali çok düşüktür. Bu durumun ağın enerji verimliliğini olumsuz etkileyeceği oldukça açıktır.

3. MATERYAL VE YÖNTEM

Tez çalışmasının temelini oluşturan 2 ana unsur bulunmaktadır. Bunlardan ilki algılayıcı düğümler diğeri ise bu düğümlerin bir araya gelerek oluşturdukları ağ yapılarıdır. Bu iki ana unsur bölümde iki alt başlık altında ele alınmaktadır. Birinci alt başlıkta algılayıcı düğümler ve nitelikleri hakkında bilgi verilmektedir. İkinci alt başlıkta ise, Kablosuz Algılayıcı Ağ (KAA) yapıları, KAA’ da kullanılan haberleşme protokolleri, veri yönlendirme algoritmaları hakkında bilgiler verilmektedir. Tez çalışması ile direkt ilgili olan kısımlar detaylandırılmış, diğeri kısımlar ise kısaca bahsedilerek anlatılmıştır.

3.1. Algılayıcılar

Ortamdaki ısı, ışık, nem, basınç vb. fiziksel değişiklikleri elektriksel büyüklüğe dönüştüren elektronik elemanlara *algılayıcı (sensör)* denir. Bu elemanlar tek başlarına bir anlam ifade etmediklerinden elektronik kontrolcülerle kullanılırlar ve bunlara girdi sağlarlar (Özcan, 2011). Şekil 3.1’de örnek algılayıcılar verilmektedir.



Şekil 3.1. Örnek algılayıcılar

Doğanın etkin korunabilmesi, bireylerin ve toplumların yaşam konforlarının artırılabilmesi ve güvenliklerinin sağlanabilmesi, bulunulan ortamdan düzenli veri alınması ve muhtemel olumsuzluklara karşı sorun oluşmadan önce haberdar olup ona göre önlem alınmasına bağlıdır. Örneğin, yangın potansiyeli yüksek ormanlık alanlardan ısı, nem, vb. verilerin düzenli olarak toplanması halinde muhtemel yangın tehlikesi önceden sezilebilir ve hızlıca müdahale edilebilir. Verilen örnekteki gibi geniş ve ulaşımı zor alanların izlenmesi gerekliliğini çıkarmaktadır. İzleme ve veri toplama insan eli ile yapılabilecek bir iş olmadığından bu gibi alanlarda, üzerindeki algılayıcıları ile ortamdan veri toplayabilen, verilerini merkez istasyona (Mİ) kablosuz iletebilen küçük ve düşük

maliyetli elektronik aygıtlar kullanılarak yapılmalıdır. Bu şekilde geniş alanlardan veri toplama amacıyla kullanılan aygıtlar algılayıcı düğüm (AD) olarak adlandırılmaktadır.

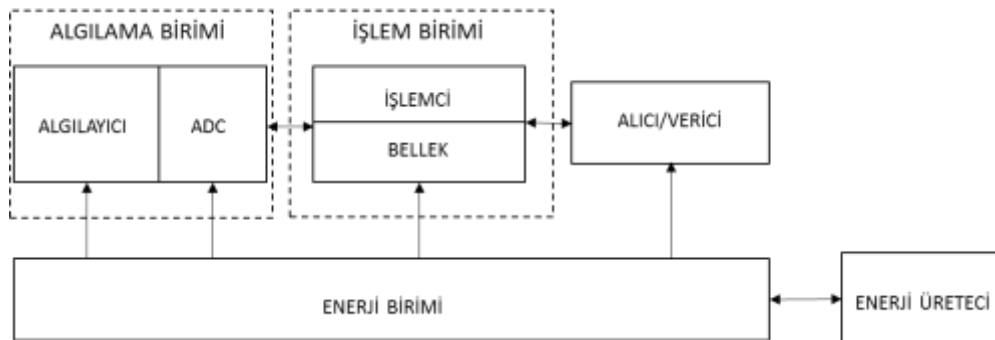
3.1.1. Algılayıcı düğümün donanım özellikleri

AD, kablosuz algılayıcı ağlarda hesaplama, algısal bilgi toplama ve ağdaki diğer bağlantılı düğümlerle haberleşme yeteneklerine sahip düğümlerdir. AD'lerin geliştirilmesine 1998 yılındaki SMARTDUST projesi ile başlanmıştır. Bu proje erken bitmesine rağmen, Berkeley NEST ve CENS projelerinin doğmasına neden olmuştur (Shah 2007). Şekil 3.2'de örnek düğümler gösterilmektedir.



Şekil 3.2. Mica2 algılayıcı düğümü

Şekil 3.3'de genel donanım mimarisi verilen AD; Algılama bölümü, İşlem bölümü, Alma/Gönderme bölümü ve Enerji bölümü olmak üzere 4 ana bölümden oluşmaktadır.



Şekil 3.3. AD mimarisi

3.1.1.1. Algılama bölümü

Dış ortamdaki mekanik, termal, kimyasal vb. değişiklikleri algılayan ve işlem bölümüne gönderen birimlerdir. Bu birime bağlı çalışan analog/sayısal dönüştürücü (ADC) modüller ile algılanan işaret sayısal dönüştürülür ve işlem bölümüne iletilir.

3.1.1.2. İşlem bölümü

Bu bölüm işlemci/denetleyici ve dahili bellek birimlerinden oluşmaktadır. Düğümün kontrol merkezi olarak çalışır ve tüm sistem bu birimden yönetilir. ADC ile sayısal dönüştürülen veri bu birimde işlenir, depolanır ve Alıcı/Verici yardımıyla merkez istasyona iletilir. Düğüm tasarımlarında enerji tüketim kontrolü önemli bir husus olduğundan kullanılacak mikroişlemci / mikrodenetleyicilerde ona göre seçilmelidir. Sıklıkla kullanılan mikroişlemci/mikrodenetleyiciler arasında Texas firması tarafından üretilen MSP 430 serisi ve Atmel ATmega sayılabilir.

3.1.1.3. Alıcı/Verici

Algılanan verinin başka bir noktaya kablosuz iletilmesini veya iletim ortamından gelen sinyallerin çözümlenerek işlem bölümüne iletilmesini sağlayan birimdir. Donanımsal farklılıklara göre değişmekle birlikte genellikle kısa mesafe haberleşmeye olanak sağlarlar. Kızılötesi (IR), radyo frekansı (RF) vb. haberleşme ortamlarını kullanırlar. Kablosuz algılayıcı ağlarda kullanılan alıcı/verici birimleri arasında RFM TR 1001, Chipcon CC100 & CC 2420 ve Inineon TDA525x ürünleri sayılabilir (Ata ve Ark., 2016).

3.1.1.4. Enerji bölümü ve güç tüketimi

AD genellikle coğrafi koşullar yönünden ulaşılması zor bölgelerde düşük maliyetlerle veri toplama amacıyla tasarlanmışlardır. Algılama yapılacak alanın büyüklüğüne göre çok sayıda AD alana rastgele serpilerek yerleştirilir. O nedenle düğümlerin sürekli bir enerji kaynağına bağlı çalıştırılmaları mümkün olmadığından enerji kaynağı olarak batarya kullanılmaktadırlar. Şekil 3.4'de kullanılan bataryalar

gösterilmektedir. Bataryalarının tekrar şarj edilmesinin mümkün olmadığı durumlarda enerjisi biten düğüm ağ içinde ölü düğüm olarak kabul edilir.



Şekil 3.4. AD'lerde kullanılan bataryalar

AD'ler algılama, işleme ve veri transferinde enerji harcarlar ancak en fazla enerji tüketen birim alıcı/verici birimlerdir. AD'lerden bazıları ve hangi durumda ne kadar enerji tükettikleri Çizelge 3.1'de verilmektedir.

Çizelge 3.1. Literatürde yer alan bazı algılayıcı düğümler ve bu düğümlerin farklı durumlara ait enerji tüketimleri (https://en.wikipedia.org/wiki/List_of_wireless_sensor_nodes)

Düğüm Adı	İşlemci	Alıcı/Verici	Uyku	Nominal	RX	TX
AquisGrain	ATmega128L	Zigbee Chipcon CC2420	47 μ A	31 mA	18,8-19,7 mA	17,4 mA
Mica2	ATmega128L	Zigbee Chipcon CC1000	N / A	15 μ A	8 mA	9,6 mA
MantaroBlocks	ATxMega32A4	Atmel_AT86RF231	3,0-3,6 μ A	637,8-728 μ A	8,9-13,2 mA	10,4-14,4 mA
iMote2	PXA271 Xscale	Zigbee Chipcon CC2420	390 μ A	31 mA	44-66 mA	44-66 mA
AWAIRS1	StrongARM SA-1100	Conexant Systems RDSSS9M	44 μ A	11.1-16.7 mA	5,56 mA	5,56 mA
AS-XM1000	TI MSP430F2618	Zigbee Chipcon CC2420	1 μ A	N/A	18,8-19,7 mA	17,4 mA

Çizelge 3.1'den görüldüğü gibi düğümlerin en düşük enerji tüketimleri uyku halinde gerçekleşmektedir. Mikrodenetleyici kontrollü çalışan hemen hemen tüm sistemler için bu durum benzerdir. Teksas Ins. firmasının geliştirmiş olduğu MSP430 serisi işlemcilerin kullanıldığı düğümlerde uyku halindeki tüketim değerleri diğerlerine göre çok daha düşüktür. Aynı düğüm RX/TX durumuna geçtiğinde ise enerji tüketimi μ A seviyesinden mA seviyelerine çıkmaktadır. Bu durumda, KAA uygulamalarında

kullanılacak AD'lerin maksimum uyku halinde kalmaları ve mümkün olduğu kadar az alma/gönderme yapmalarının sağlanması, düğümlerin enerji verimliliği için geliştirilecek metotlar açısından ana hedefdir.

3.1.2. İşletim sistemi

AD'deki çeşitli işlevsellikleri desteklemek ve donanım alt yapısını kontrol edebilmek için çalışan yerleşik kodlardan oluşan yapıya işletim sistemi denilmektedir. KAA'da kullanılacak AD'lerde en yaygın bilinen işletim sistemi açık kaynak kodlu dağıtılan TinyOS işletim sistemidir. Uluslararası TinyOS Alliances birliği tarafından C dilinin bir çeşidi olan nesC dilinde geliştirilmiştir. Daha az güç tüketimi için mikrodenetleyicinin daha uzun süre uyumasını sağlayan bir strateji izlemektedir (Dener, 2014).

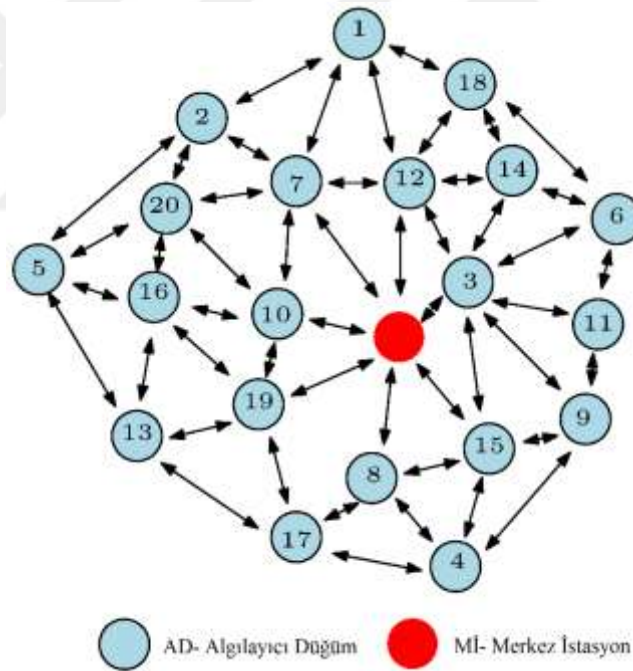
İzleme yapılacak yerin büyüklüğüne göre değişmekle birlikte, alanın büyük çoğunluğunun kapsama alanına girebilmesi için bu düğümlerden yüzlerce atılması gerekebilir ki bu durum hem düğümlerin enerji verimliliklerinin önemini hem de düğüm maliyetlerinin önemini ortaya koymaktadır. Düğüm maliyetlerinin düşük olabilmesi ekonomik ve gelişmişlik gibi farklı parametrelere bağlıdır ve bu düğümlerin sahada kullanılabilir olması için maliyetinin bir doların (USD) altında olması gerektiği daha önce yapılan çalışmalarda vurgulanmıştır (Akyıldız ve ark. 2002). Ancak küçük boyutlu ve düşük maliyetli cihazlar olarak planlandıklarından üzerinde bulunacak enerji kaynakları bir o kadar sınırlı olacaktır. O nedenle AD'lerde önemli olan husus enerji verimliliğidir. Düğümün tasarımı, üzerinde bulunan elektronik devre elemanlarının özellikleri ve işlemci içinde çalıştırılan algoritma vb. faktörler düğümlerin enerji tüketimlerini doğrudan etkileyen unsurlardır.

AD'lerin geliştirilme amaçları, ortamdan algıladıkları verileri Mİ'ye iletmektir. İletim işlemi konvansiyonel yapılarda her düğüm tarafından bireysel yapılmaktadır ancak bu durum düğümlerin enerji tüketimlerini olumsuz etkilemektedir. Sınırlı enerji kapasitesine sahip AD'lerin uzun ömürlü olabilmeleri enerjilerini dengeli kullanmalarına bağlıdır. Bu nedenle, AD'lerin, verisini Mİ'ye direkt kendisi ulaştırmak yerine yakınındaki başka bir algılayıcı düğüm üzerinden Mİ'ye ulaştırma ihtiyacı, KAA yapılarının doğmasına neden olmuştur.

3.2. Kablosuz Algılayıcı Ağlar

KAA, fiziksel veya çevresel koşulları izlemek için algılayıcıları kullanan, rastgele dağıtılan özerk aygıtlardan oluşan ağlardır (Akyıldız ve ark. 2002). KAA' da verinin düğümler üzerinden en kısa yoldan ve en az enerji maliyeti ile Mİ' ye iletilmesi problemi, KAA teknolojisinin gelişmesine sebep olmuştur. KAA' da temel prensip, herhangi bir AD tarafından algılanan verinin ağı oluşturan diğer AD' ler üzerinden Mİ' ye iletilmesine dayanmaktadır. Bu nedenle düğümler kendi aralarında sürekli iletişimde kalmak durumundadırlar.

KAA' da enerji verimliliği üzerinde durulması gereken en önemli hususların başında gelmektedir. Enerji verimliliğini etkileyen iki temel unsur bulunmaktadır. Bunlardan ilki, düğümler arasındaki ortam erişim protokollerinin belirlenmesi diğeri ise verinin Mİ' ye iletim şeklidir. Şekil 3.5' de örnek bir KAA gösterilmektedir.



Şekil 3.5. Kablosuz algılayıcı ağ yapısı

3.2.1. Ortam erişim protokolleri

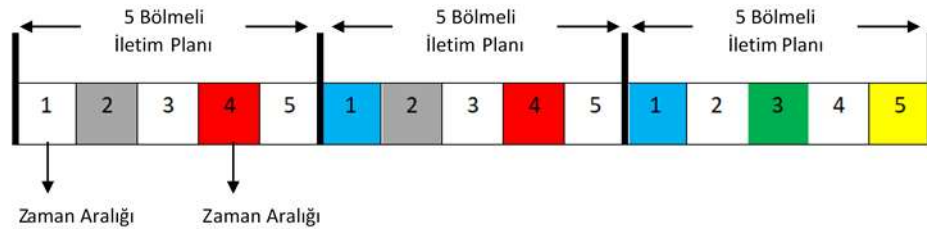
Önceki bölümlerde belirtildiği gibi algılayıcı düğümlerin en fazla enerji tüketimleri nominal durum ve RX/TX durumunda olmaktadır. En düşük enerji tüketimi ise uyku durumunda gerçekleşmektedir. O nedenle algılayıcı düğümlerin olabildiğince

uyku modunda kalmaları ve gereksiz RX/TX iletişiminden kaçınmaları yaşam ömürleri açısından önem arz etmektedir. Bunun için düğümlerin hangi koşullarda uyanık kalması ve ne zaman RX/TX durumuna geçmesi gerektiği ile ilgili kuralların geneli Ortam Erişim Protokolleri (Medium Access Control (MAC)) olarak adlandırılmaktadır. IEEE tarafından 802.11 koduyla standartları belirlenmiş bu kurallar, ağı oluşturan düğümlerin veri toplama ve düğümler arasındaki iletişim esasları ile ilgili olduğundan ayrı bir katman olarak değerlendirilmekte ve bu katmana MAC katmanı denilmektedir.

MAC katmanında gereksiz enerji sarfiyatına sebep olan unsurların başında kontrol paketlerinin fazlalığı, gereksiz dış ortam dinlemeleri ve istem dışı alım talepleri sayılabilir. Ortam dinlenmesinden anlaşılan, düğümlerin kendine ait olmayıp başka düğümlerden gelen veri paketlerin iletilmesi taleplerinin alınması ve alınan paketlerin diğer düğümlere iletilmesidir. Bu ise düğümün sürekli uyanık kalmasına ve sıklıkla RX/TX durumuna geçmesine neden olduğundan en fazla enerji sarfiyatına sebep olan unsurdur. Bu gibi durumlarda enerji tüketimini azaltmak için farklı protokoller önerilmiştir. KAA' da öne çıkan protokoller TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access), CSMA (Carrier Sense Multiple Access) sayılabilir (Okay ve Özdemir, 2014).

3.2.1.1. TDMA (Time Division Multiple Access)

TDMA, zaman bölmeli çoklu erişim protokolüdür. Bir iletim periyodu için gerekli zaman ağıdaki düğümlere eşit oranda paylaştırılmıştır. Her düğüm kendine ayrılan zaman aralığında iletme geçer ve verisini iletir. Şekil 3.6'da örnek bir iletim modeli verilmiştir.



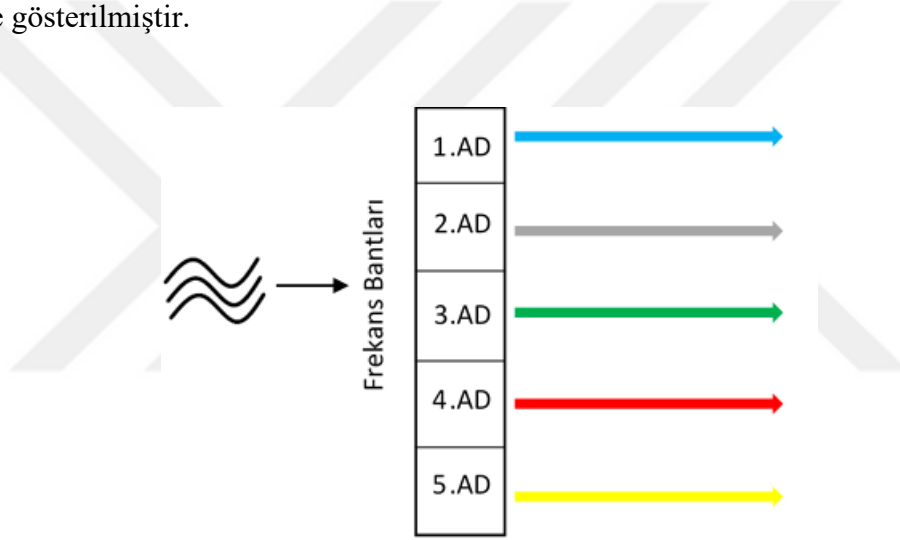
Şekil 3.6. TDMA iletim modeli

Şekil 3.6'da verilen modelde, 5 farklı düğüm için 5 farklı zaman aralığı belirlenmiştir. Düğümler kendilerine ayrılan zaman aralıklarında iletme geçerler. Verisi olmayan düğüm iletme geçmez ve o zaman aralığı boş geçer. Örneğin, 1. Periyotta 1, 3

ve 5. düğümler iletim yapmamış o zaman aralıkları beklenmiştir. Bu durum, ağda oluşan muhtemel çarpışma ve örtüşme gibi problemlerin önüne geçer ve her düğüm kendine ayrılan zaman aralığında tüm frekans bandını kullanabildiğinden hızlı bir iletim gerçekleşir ancak boş geçen zamanlar hesaba katıldığında protokolün verimi düşmektedir.

3.2.1.2. FDMA (Frequency Division Multiple Access)

FDMA, frekans bölmeli çoklu iletim protokolüdür. Ağın haberleşmede kullandığı frekans bandı ağdaki düğümlere bölünür ve her düğüm kendi frekans aralığından yayın yapar. Bu sayede düğümler aynı anda ilettime geçebilirler. Örnek bir iletim modeli Şekil 3.7’de gösterilmiştir.

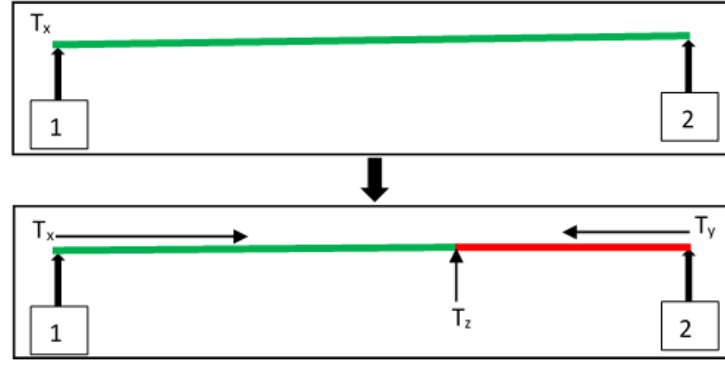


Şekil 3.7. FDMA iletim modeli

Burada, frekans bandı 5 düğüm için ayrılmıştır. Düğümler kendi frekans aralıklarının zamandan bağımsız olarak ilettime geçebilirler. Burada da çarpışma veya örtüşme problemi yaşanmaz ancak bant aralığı bölündüğünden iletim hızı da bölünmüş olur hızda düşmeler yaşanmaktadır.

3.2.1.3. CSMA (Carrier Sense Multiple Access)

CSMA, iletim hattını kontrol eden, hattın boş olması durumunda iletimi başlatan bir protokoldür. Zaman ve frekans bağımlılığı yoktur ancak çarpışma riski yüksek bir protokoldür. Çalışma prensibi Şekil 3.8’de gösterilmiştir.

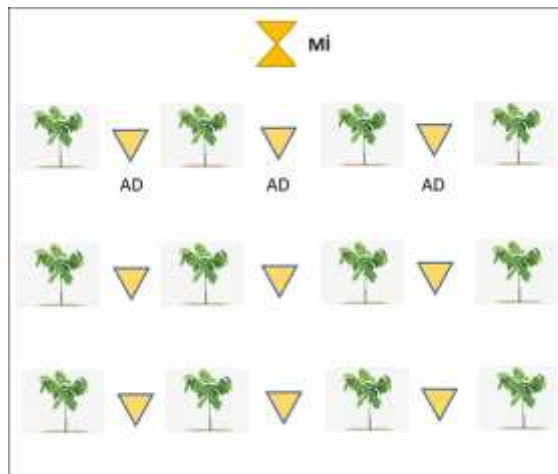


Şekil 3.8. CSMA iletim modeli

Burada, 1. düğüm hattı kontrol eder ve hattın boş olduğunu görünce T_x zamanında iletme geçer. İletim sinyali 2. düğümüne ulaşmadan 2. düğüm iletme geçmek için hattı kontrol eder ve hat boş olduğu için T_y zamanında iletme geçer. Ancak bu iki sinyal T_z zamanında hatta çarpışmaya neden olur ve iletim gerçekleşmez.

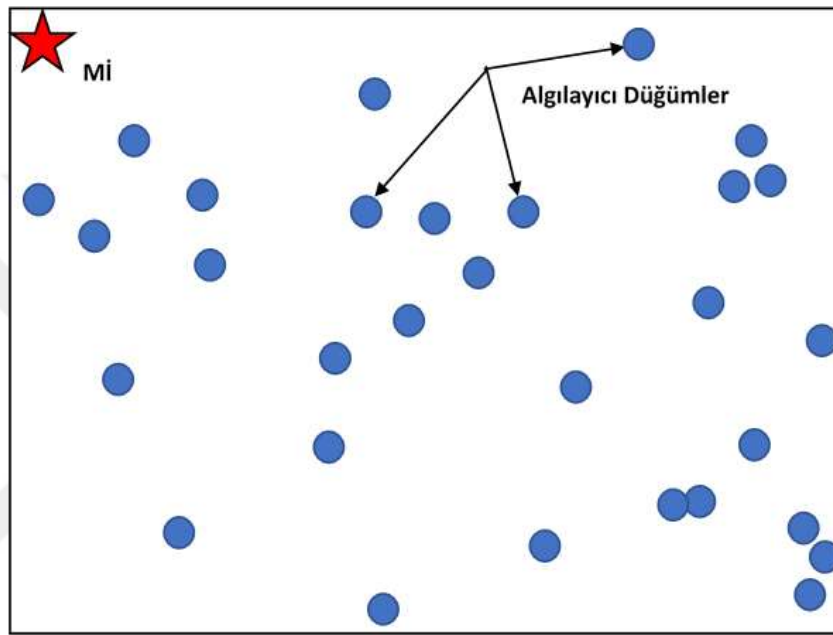
3.2.2. KAA yapıları

KAA uygulamalarında oluşturulan ağ yapıları, düzenli dağıtık ağlar ve düzensiz dağıtık ağlar olmak üzere 2 kısımda incelenebilir. Bu farklılık, uygulamanın tipi ve uygulama yapılacak alanın coğrafi koşulları ile ilgilidir. Tarım uygulamalarında toprak neminin takip edilmesi ve sulama planının bu veriye göre yapılması için kurulan ağlar, hastane veya fabrika gibi kapalı alan içinde düzenli olarak yerleştirilen algılayıcılardan kurulan ağlar düzenli dağıtık yapı içerisinde değerlendirilebilir. Düzenli yapı örneği Şekil 3.9'da verilmektedir.



Şekil 3.9. Tarımsal KAA uygulaması (AD: Algılayıcı Düğüm, Mİ: Merkez İstasyon)

Coğrafi olarak ulaşılması zor alanlarda algılama ve veri toplama uygulamalarında, AD'lerin düzenli olarak yerleştirilmesi mümkün olmadığından hava aracıyla serpilerek alana yayılması sağlanmaktadır. Orman yangınlarında yangının çıktığı konumu kestirmek veya askerî amaçlarla açık alandaki hareketliliği izlemek amacıyla yapılan uygulamalarda en uygun yöntem serpme yöntemidir. Bu tarz uygulamalar ise düzensiz dağıtık yapıdaki KAA'lar olarak tanımlanabilir. Düzensiz dağıtık yapıdaki ağ modeline örnek Şekil 3.10'da gösterilmiştir.

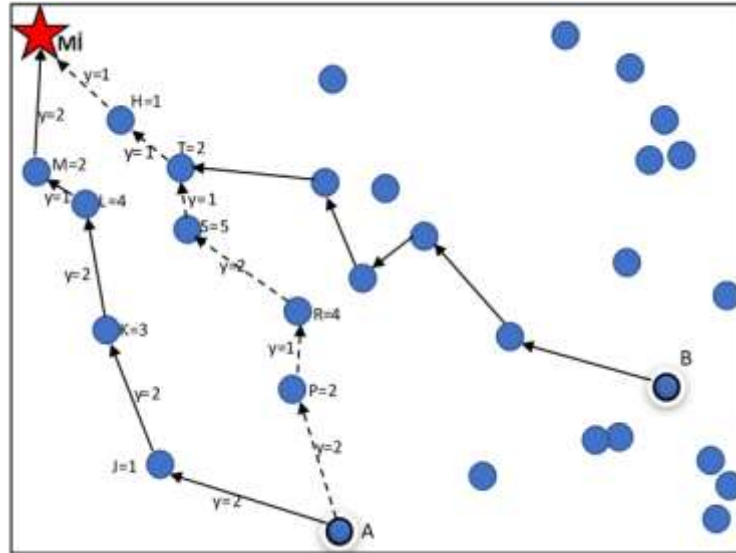


Şekil 3.10. Düzensiz dağıtık yapıdaki ağ modeli (Mİ: Merkez İstasyon)

3.2.3. Veri iletim yöntemleri

Geleneksel kablosuz algılayıcı teknolojilerinde, algılanan veri direk olarak Mİ'ye ulaştırılır. Bu metot, enerji tüketiminin göz ardı edildiği durumlarda tercih edilebilir ancak AD'nin Mİ ile haberleşebildiği var sayılır. AD'nin Mİ'ye olan uzaklığı veya coğrafi şartların zorluğu verinin sağlıklı iletimini olumsuz etkileyen en önemli unsurdur.

KAA'da veri iletimi, geleneksel kablosuz iletim teknolojilerinden farklı olarak ağı oluşturan düğümler üzerinden atlayarak sağlanmaktadır. Uygulamalarında, algılama yapılacak alana çok sayıda AD bırakılır ve veri iletimi AD'ler üzerinden gerçekleştirilir. Ancak verinin hangi yoldan iletileceği önemlidir. Bu tercih, düğümlerin ve dolayısıyla ağın enerji tüketimini, iletim hızını ve iletim kalitesini etkilemektedir. Örnek bir iletim modeli Şekil 3.11'de gösterilmektedir.



Şekil 3.11. Örnek bir veri iletim modeli

Verinin Mİ'ye ne şekilde ulaştırılacağını konu alan çalışmalar ise yönlendirme protokolleri (Routing Protocols) olarak adlandırılır (Sarkar ve Murugan, 2016; Rathi ve ark.,2012; Shabbir ve Hassan, 2017; Harmankaya, 2007). Yönlendirmede uygun rotanın belirlenmesi en önemli unsurdur. Uygun rota belirlenirken düğümlerin enerji seviyelerine, yolda harcanacak enerjiye ve verinin kaç düğüm üzerinden atlayacağına dikkat edilmektedir. Örneğin, Şekil 3.11'de görüldüğü gibi A noktasından Mİ'ye ulaşmak için pek çok rota tercihi bulunmaktadır. Bunların içinden, *JKLM* ve *PRSTH* rotaları en uygun 2 rota olarak seçilebilir. Bu iki rota için yukarıda bahsedilen parametrelere göre Çizelge 3.2 oluşturulabilir.

Çizelge 3.2. A noktasından Mİ'ye ulaşmak için izlenecek rotalar ve enerji tüketimleri (AD_T : Düğümlerin mevcut toplam enerjileri, Y_T : Yolda harcanan toplam enerji, D_S : Atlanan düğüm sayısı)

Rota	AD_T	Y_T	D_S
<i>PRSTH</i>	$E=2+4+5+2+1=14$	$Y=2+1+2+1+1+1=8$	5
<i>JKLM</i>	$E=1+3+4+2=10$	$Y=2+2+2+1+2=9$	4

Buna göre; hangi yolun tercih edileceği önceliklere göre değişebilmektedir. Minimum enerji tüketimi için *PRSTH* yolu tercih edilir. Çünkü, hem rota üzerindeki düğümlerin enerjileri daha yüksek hem de yolda harcayacağı enerji daha düşüktür. Diğer taraftan, *JKLM* rotasında ise diğer rotaya göre atlama sayısı daha azdır. Atlama sayısının fazla olması gecikme zamanına ve aşırı yüklenmeye sebep olabilir. Bu gibi durumlarda,

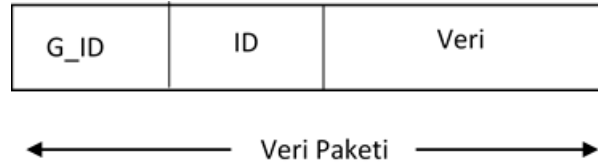
uygulamanın yeri, uygulamadaki öncelikler gibi nitelikler göz önüne alınarak optimum rotanın belirlenebilmesi için farklı protokoller geliştirilmiştir.

3.2.4. Veri merkezli protokoller

Bu yaklaşımda, önemli olan verinin kendisidir ve diğer parametreler göz ardı edilebilir. Verinin iletimi için algılama yapan düğüm tarafından herhangi bir hedef belirtmeksizin ağa yayın yapılır. Ağdaki diğer düğümler tarafından alınan yayın Mİ' ye ulaştırılır. Bu yaklaşımda verinin önceliği düğümlerin enerji tüketimi, çarpışma riski veya aşırı yüklenme gibi olumsuzlukların önündedir. Veri merkezli protokollerden öne çıkan dört tanesi şunlardır;

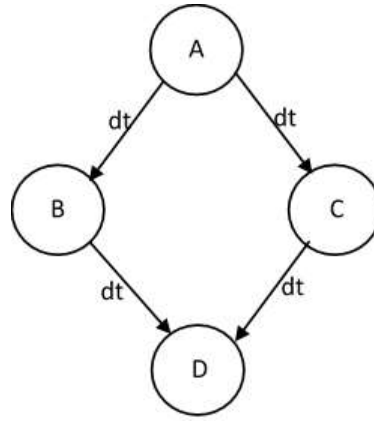
3.2.4.1. Flooding protokolü

Bu protokolde, verinin Mİ'ye ulaştırılması öncelikli konudur. Yönlendirmede en iyi yol aranmaz. Algılanan veri ilgili düğüm tarafından ağa yayın yapılarak gönderilir. Veriyi alan düğüm gönderenin bilgilerini kayıt ederek yeni bir veri paketi hazırlar ve gönderen düğüm dışında ağdaki diğer düğümlere yayın yapar ve bu şekilde verinin Mİ'ye ulaştırılması sağlanmış olur. Örnek bir veri paketi Şekil 3.12'de verilmektedir. Burada, veri göndericinin ID'si ve başka bir düğümden atlama varsa onun ID'si ile birlikte veri gönderilir.



Şekil 3.12. Örnek veri paketi

Bu protokolün önemli dezavantajları bulunmaktadır. En önemlileri ise çökme ve örtüşme olaylarıdır. Çökme, aynı verinin farklı düğümlerden aynı düğüme iletilmesi olayıdır. Örtüşme ise aynı alan içinde aynı veriyi algılayan farklı düğümlerin bulunması ve aynı verinin farklı düğümler tarafından yayınlanması şeklinde gerçekleşmektedir. Flooding protokolünün çalışma prensibi Şekil 3.13'de verilmektedir.



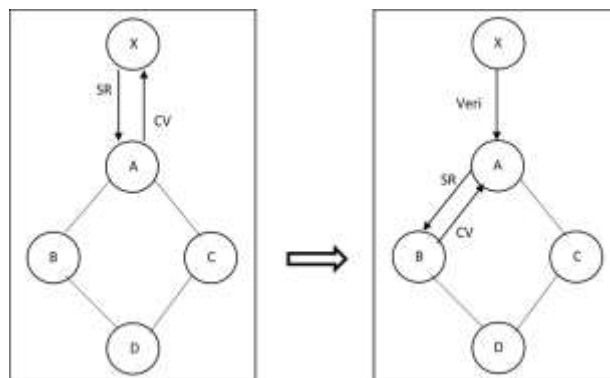
Şekil 3.13. Flooding protokolü çalışma prensibi

3.2.4.2. Rumor protokolü

Rumor protokolüne göre, veri ile birlikte enerji tüketiminin planlaması da öncelikli konudur. Ağın kurulumunda, düğümler komşu düğümlerle ilgili bir liste tutarlar. Algılama olduğunda, veri ağa yayınlanır ve komşu listesine göre düğüm atlama sayısı göz ardı edilerek en kısa yollardan oluşan bir rota üzerinden veri Mİ'ye ulaştırılır.

3.2.4.3. SPIN protokolü

Bu protokolde diğerlerinden farklı olarak, veri ağın geneline yayılmaz. Ağ içinde düğümlerin tamamına bakılır ve konumu, enerji seviyesi gibi nitelikler bakımından üstün olan düğümler belirlenir. Algılanan veri önce bu düğümlere sonra Mİ'ye ulaştırılır. Benzer özelliklere sahip diğer bir protokol ise Cougar protokolüdür. SPIN protokolü çalışma prensibi Şekil 3.14'da anlatılmaktadır.



Şekil 3.14. SPIN protokolü çalışma prensibi

Şekil 3.14’da gösterildiği gibi, X düğümü A düğümüne iletme başlamadan önce ona bir SR mesajı gönderir. A düğümü ona cevap olarak CV mesajı yayınlarsa X düğümü veri gönderimine başlar. Bu işlem ağdaki tüm düğümler için geçerlidir.

3.2.4.4. Energy Aware Routing (EAR)

Veri merkezli olmasına rağmen enerji verimliliğini önceleyen bir protokoldür. Algılayıcı düğümler, Mİ ile aralarındaki en kısa yolu belirlerler. Her düğüm kendine göre en kısa yolu belirlediğinden, verinin her koşulda en kısa yoldan Mİ ulaştırılması ve bu şekilde ağın ömrünün uzayacağı öngörülmektedir.

3.2.5. Konum tabanlı yönlendirme protokolleri

Bu protokolda düğümlerin ağ içindeki konumları bilinmektedir. Konumların biliniyor olması AD ve Mİ arasındaki en kısa yolun belirlenmesi anlamına geleceğinden enerji verimi artırılmış ve ağın ömrü uzatılmış olacaktır. Ancak farklı yöntemler kullanılsa da konum bilgisi GPS modülü ile elde edilebilmektedir. AD’nin entegre bir GPS modülüne sahip olması düğüm maliyetini olumsuz yönde etkileyecektir.

3.2.6. Hiyerarşik yönlendirme protokolleri

Veri merkezli yönlendirme protokollerinde, verinin iletimi diğer şartların önünde tutulduğundan ağ üzerinde önemli problemlerin oluşmasına neden olmuştur. Örtüşme ve benzeri olaylarından dolayı ağın aşırı yüklenmesi, veri çökmelerinin yaşanması ve en önemlisi enerji tüketiminin göz ardı edilmesi, gerçek uygulamalarda KAA kullanımını olumsuz yönde etkilemiştir. KAA, yapısı itibari ile çok sayıda düğümden oluşan ve dolayısı ile düğüm maliyetleri düşük olması gereken yapılardır. Bugünün şartlarında, maliyetin düşürülmesi ise imkanlarının kısıtlanması ile mümkün olabilmektedir. Bu da daha düşük enerji kapasitesi, daha kısa iletim mesafesi anlamına gelmekte, mevcut imkanların verimli kullanılmasının önemini açık bir şekilde ortaya koymaktadır.

Hiyerarşik yönlendirme protokollerinde asıl amaç, ağın enerji veriminin artırılması, ağda oluşan fazla yükün düşürülmesi ve veri iletim kalitesinin ön plana alınmasıdır. Algılanan veri, geleneksel çok atlamalı (multi-hop) yönlendirmeli bir şekilde Mİ’ye iletilmez. Ağın ömrü düğümlerin enerji seviyesi ve iletim mesafesi ile orantılı

olduğundan düğümlerin kısa mesafede ve ağda belirlenmiş aktarıcılar yardımıyla Mİ'ye ulaşması beklenir. Ayrıca, veri iletiminin sürekli ya da zaman bölmeli yapılması yerine koşullu veri toplama ve iletme stratejisi benimsenmiştir. Örneğin, “sıcaklık eşik değerinin üstüne çıktığında” veya “nem eşik değerinin altına düştüğünde” veri iletimi gerçekleşmektedir.

Hiyerarşik yönlendirmenin diğer bir özelliği ise iletim mesafesinin algoritma içinde önemli bir değişken olmasıdır. Bu değişkene göre düğümler, çıkış güçlerini ayarlayabilmekte ve bu sayede enerji tüketimlerini düşürebilmektedirler. Hiyerarşik yönlendirmenin tüm özellikleri kümeleme başlığı altında toplanmış ve farklı algoritmalar geliştirilmiştir. LEACH, TEEN, HEED, APTEEN ve PEGASIS algoritmaları bunlardan başlıca olanlarıdır.

Kümelemede 2 ana metot belirlenmiştir. Birincisi, Mİ tarafından yapılan kümeleme işlemidir. Ağdaki tüm düğümler konum, enerji seviyesi vb. bilgilerini Mİ'ye gönderir. Mİ ise bu bilgilere göre bir algoritma yürüterek ağı kümelere ayırır ve Küme Baş (KB) larını belirler. Sonuç, Mİ tarafından yapılan bir yayımla ağa bildirilir ve kümeleme işlemi bir tur için tamamlanmış olur. KB düğümlerin enerjileri sönümlenmeye yakın bu işlem belirlenen bir tur sayısı kadar tekrar edilir. Bu metodun uygulanabilir olması için ağa yayılan düğümlerinin konumlarının bilinmesi ve her düğümün Mİ ile haberleşebilir olması gerekmektedir. Yapılan çalışmalarda konum bilgisinin ağırlıklı olarak GPS üzerinden alınması AD'lerin maliyetini olumsuz etkilemekte ve uygulanabilirliğini düşürmektedir. Ayrıca Mİ ile haberleşemeyen düğümlerin var olması durumu, ağın enerji verimliliği ve veri toplama kapasitesini olumsuz etkileyecektir.

Diğer metot ise ağın kendi içinde öz örgütlenecek şekilde yapılan kümelemedir. Dışarıdan herhangi bir müdahale olmadan düğümler kendi aralarında KB'leri seçerler ve onlara bağlı kümeler oluştururlar. Bu metotta önemli olan husus KB düğümlerin nasıl seçileceği ve kümelerin nasıl oluşturulacağıdır. Ayrıca, kümeleme sonucunda oluşan ağ yapısının enerji verimli olması beklenmektedir. Öz-örgütlü yapılarda KB seçimi 2 şekilde yapılabilmektedir. İlki rastgele seçim, diğeri belli parametreler göz önüne alınarak yapılan seçimdir. Rastgele seçimli algoritmalarda her düğüm, algoritmada açıklanan bir eşik değer fonksiyonundan elde edilen değeri baz alarak 0-1 arasında rastgele sayı üretir ve bu sayının eşik değerinin altında ya da üstünde olmasına göre o düğüm KB ilan edilir. Diğer KB seçim algoritmalarında ise düğümlerin mevcut enerji seviyeleri, yakınlık-uzaklık gibi parametreler baz alınarak yapılan seçimdir. Literatürdeki Mİ yönlendirmeli ve öz-örgütlü yapıda olan algoritmalar gruplandırılarak ve Çizelge 3.3'de verilmiştir.

Çizelge 3.3. Literatürdeki algoritmaların gruplandırılması

Algoritma	Yıl	Grup
LEACH	2000	Öz-Örgütlü
PEGASIS	2002	Öz-Örgütlü
HEED	2004	Öz-Örgütlü
MEKA	2020	Öz-Örgütlü
TEEN-APTEEN	2002	Mİ Yönlendirmeli
BCDCP	2005	Mİ Yönlendirmeli
RE-EERP	2011	Mİ Yönlendirmeli
EEBCDA	2012	Mİ Yönlendirmeli
DPC-EERRP	2012	Mİ Yönlendirmeli
Q-LEACH	2013	Mİ Yönlendirmeli
Mk-means	2016	Mİ Yönlendirmeli
EECPK	2016	Mİ Yönlendirmeli
OCM-FC	2018	Mİ Yönlendirmeli
DHSCA	2018	Mİ Yönlendirmeli
I-LEACH	2019	Mİ Yönlendirmeli
PSO-NMDC	2019	Mİ Yönlendirmeli
HEEC	2019	Mİ Yönlendirmeli
TDPSO	2020	Mİ Yönlendirmeli

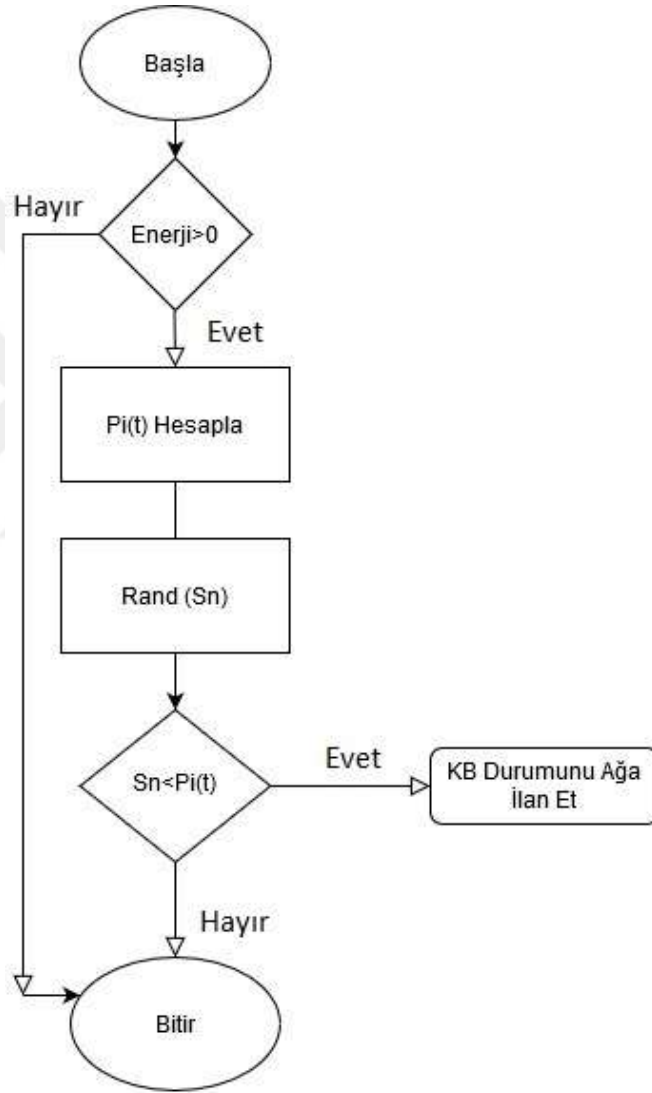
3.2.6.1. LEACH

LEACH (Low Energy Adaptive Clustering Hierarchy), KAA'larda düğümlerin kendi kendilerine kümelenebilmelerini sağlayan öz örgütlü yapıya sahip bir kümeleme algoritmasıdır. Algoritma, kurulum aşaması ve kararlı durum aşaması olmak üzere iki aşamadan oluşmaktadır. Kurulum aşamasında düğümler algoritmada tanımlı Denklem 3.1'de gösterilen eşik değer fonksiyonunu çalıştırlar.

$$P_i(t) = \begin{cases} \frac{p}{N - p * (r * \text{mod} \frac{N}{p})} & : C_i(t) = 1; \\ 0 & : C_i(t) = 0 \end{cases} \quad (3.1)$$

Burada N KAD sayısını, p küme başı olma yüzdesini r ise iterasyon sayısını ifade eder. $C_i(t)$, indikatör fonksiyonu olarak belirtilmiştir ve düğüm i 'nin KB olup olmayacağını ifade eder. Bir düğümün r . iterasyonda KB seçimine katılıp katılmaması enerjisine ($E_i(t)$) bağlıdır. Eğer $E_i(t) < 0$ ise düğüm i ölü kabul edilir ve indikatör

fonksiyonu $C_i(t) = 0$ deęerini alır. Eęer, $C_i(t) = 1$ ise dđęüm i KB seęimine katılır. Bu durumda KB olabilecek dđęümlerin toplamı, $\sum_{i=1}^N C_i(t)$ şeklinde ifade edilmektedir. KB seęiminde ortama daęıtılan her bir dđęüm, hesapladıęı $P_i(t)$ deęeri ile 0-1 arasında rastgele ürettięi (S_n) sayısını karşılaştırır. Eęer, $S_n < P_i(t)$ ise, dđęüm i küme başı olarak seęilir. Küme başına üye olacak dđęümler oklit mesafelerine göre belirlenir (Heinzelman, 2002, Singh ve ark., 2017). Algoritmanın akış diyagramı Şekil 3.15’de verilmektedir.



Şekil 3.15. LEACH algoritması akış diyagramı

LEACH algoritmasının dezavantajları:

- KB düğümler rastgele seçim modeli ile belirlenmektedir. Bu durum KB düğümlerin alana dengeli dağılamamaları halinde enerji tüketimini olumsuz etkileyecektir.
- KB olan bir düğüm, etrafına bir mesaj yayınlarsın ve bu mesajı alan düğümler o KB'ye tabii olurlar. Fakat aynı zamanda birden fazla düğümün KB olma mesajı yayınlaması halinde iletim hattında çarpışma veya örtüşme gibi senkronizasyon bozukluklarına neden olacaktır.
- Birbirine yakın olan düğümlerin KB olması durumunda ağın kapsama alanı düşecektir. Bu durumda ise hangi düğümün hangi KB'ye tabii olacağı kararsızlık durumu oluşturmaktadır.

3.2.6.2. TEEN ve APTEEN

LEACH algoritmasından farklı olarak KB seçimini Mİ tarafından yaptırır. Düğümlerin düzgün bir dizilimde olduğu varsayılır. İletim modeli olarak TDMA protokolü kullanılır ve algılanan her veri Mİ'ye ulaştırılır. TEEN ve APTEEN' de ise veride önemli bir farklılık algılandığı zaman ya da verinin belirlenen eşik değerin altında ya da üstünde kalması durumlarında Mİ'ye iletilir. Bu sayede, ağda oluşan gereksiz veri yükü engellenmiş ve enerji tüketimi azaltılmış olur.

3.2.6.3. HEED

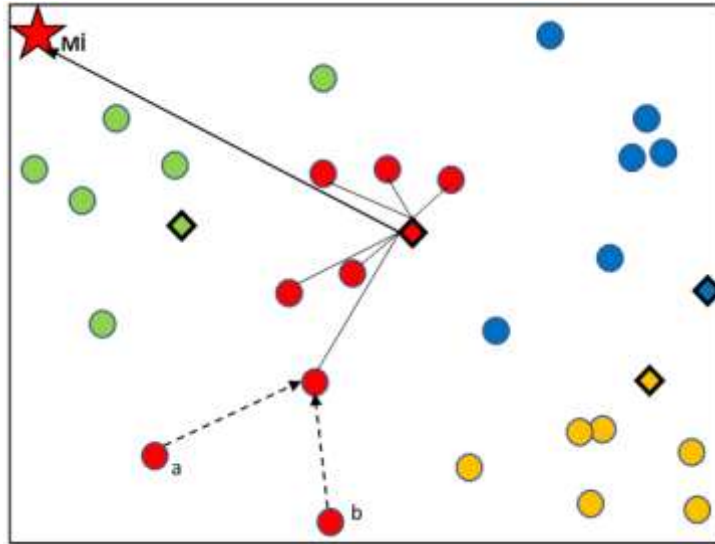
HEED (Hybrid Energy-Efficient Distributed Clustering Approach) algoritması LEACH ile önemli benzerlikler göstermektedir. Rastgele seçim metodunu benimseyen HEED, LEACH' den farklı olarak küme başı düğümleri belirlerken düğümlerin enerji seviyelerini dikkate almaktadır. Enerjisi düşük olan düğüm eşik değerin altında bir değer bile yakalasa o düğüm KB olarak seçilmez. Algoritmada KB seçimi, Denklem 3.2'ye göre yapılmaktadır.

$$CH_{prob} = C_{prob} * \frac{E_{residual}}{E_{max}} \quad (3.2)$$

Burada, C_{prob} , KB olma yüzdesini (Örn: %5), $E_{residual}$, düğümün kalan enerji miktarını, E_{max} ise düğümün referans enerjisini göstermektedir. Bu denkleme göre, $CH_{prob} > 1$ olan düğümler KB seçilebilirler. Örneğin; başlangıç enerjisi $E_{max} = 0.5j$ ve $C_{prob} = 5$ değerlerine göre; iterasyonlar sonunda enerjisi düğüm için $0.05j$ kalan bir düğüm için $CH_{prob} = 0.5$ sonucu elde edilir. $CH_{prob} < 1$ olduğundan düğüm KB seçilemez. Ayrıca, düğümler iletim mesafelerine göre farklı çıkış güçleri kullanarak ağdaki enerji verimini arttırlar (Younis ve Fahmy, 2004).

3.2.6.4. PEGASIS

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) protokolü LEACH protokolünün geliştirilmiş bir modeli olarak sunulmuştur. LEACH ile temel prensipleri aynıdır ancak farklı olarak kümedeki düğümler verilerini hemen küme başına iletmezler. Düğümün küme başına olan mesafesi fazla ise küme başına daha yakın komşu bir düğüme veriyi iletir oradan küme başına ulaştırılması sağlanır. Örnek bir durum Şekil 3.16'da gösterilmiştir. Küme başı seçimi ise her turda rastgele seçim modeli ile yapılmaktadır. PEGASIS algoritması zaman içinde geliştirilmiş ve farklı öneriler getirilmiştir. EEPB (Energy Efficient PEGASIS Based Algorithm) ve PEGASIS-Ant algoritması bunlardan önemli olanlarıdır. Bu iki algoritmada PEGASIS' den farklı olarak küme başı seçiminde düğümlerin enerji seviyelerine ve merkez istasyona olan mesafelerine bakılır. Küme başı düğümler M_1 ' ye yakın olan düğümlerden seçilmektedir (Lindsey ve Raghavendra, 2002; Hussein ve Khalid, 2019).



Şekil 3.16. PEGASIS protokolü

3.2.6.5. k-means algoritması

k-means, Öklid uzayında D veri setini oluşturan n tane nesnenin ağırlık merkezine göre k kümeye ayrılmasını sağlayan kümeleme algoritmasıdır. Hızlı ve kolay olmasından dolayı genellikle veri madenciliği uygulamalarında tercih edilmektedir. Algoritma şu şekilde çalışır; ağırlık merkezi noktaları başlangıçta verilen k sayısı kadar random seçilir. Ağın içinde bulunan tüm KAD'lerin ağırlık merkezlerine olan mesafeleri Öklid metodu ile hesaplanır. Kümeleme işlemi, karesel hata (Denklem 3.3) minimum noktaya ulaşıncaya kadar iterasyonlar halinde devam ettirilir.

$$E = \sum_{i=1}^k \sum_{n_t \in C_i} \|n_t - c_i\|^2 \quad (3.3)$$

Burada, n_t küme içindeki düğümü, c_i ise kümenin ağırlık merkezini ve k ise ayrılmak istenen küme sayısını ifade eder. Denklem 3.4'deki küme üyelik fonksiyonu ile düğümlerin hangi kümeye dahil olacağı belirlenir. Küme üyelik fonksiyonu $m(n_t, i)=1$ olduğunda n_t düğümü km_i kümesine dahil olur (Jiawei ve ark., 2012; Zalik, 2008). k-means algoritmasının çalışma prensibi ile ilgili sözde kodu Şekil 3.17'de görülmektedir.

$$m(n_t, km) = \begin{cases} 1; & \text{Eğer } km = \arg \min(\|n_t - c_i\|^2) \quad i = 1, \dots, k \\ 0; & \text{Diğer durumlarda} \end{cases} \quad (3.4)$$

```

1: Başla
2: Oku k (İstenen küme sayısı)
3: n= düğüm sayısı
4: j=0
5: Rasgele  $c_i$  belirle
6: Eğer  $j \leq k$  ise  $j=j+1$  ve 4. Adıma Git
7: Tekrar Et
8:   i=0
9:   J=0
10: Her  $n_{th}$  düğüm için  $m(n_t, i)$  hesapla
11: Eğer  $m(n_t, i) == 1$  ise  $n_t$  düğümünü  $i_{th}$  kümeye dahil et.
12: Eğer  $j \leq n$  ise 10. Adıma Git
13: Eğer  $i < k$  ise 9. Adıma Git
14: k adet küme için yeni  $C_i$  hesapla
15:  $C_i$  noktalarında değişiklik olmayıncaya kadar
16: Bitir

```

Şekil 3.17. k-means algoritmasının sözde kodu

3.2.6.6. TD-PSO algoritması

Parçacık sürü optimizasyon (PSO) algoritması, kuş ve balık sürülerinin sosyal davranışlarından esinlenen popülasyon temelli stokastik bir optimizasyon algoritmasıdır. PSO algoritmasında “parçacık” olarak adlandırılan bireyler optimum sonucu aramak için mevcut konumlarını ve hızlarını kendinin ve popülasyonun tecrübelerinden faydalanarak güncellerler.

PSO algoritmasının, amaç fonksiyonu dışında bir bilgiye ihtiyaç duymaması, düzenlemeyi gerektiren parametre sayısının az olması ve problemlere kolay uyarlanabilmesinden dolayı en çok tercih edilen optimizasyon tekniklerinden biridir. Sürekli problemlerin çözümü için geliştirilen PSO algoritması birçok alanda uygulanmıştır ve kabul edilebilir sonuçlar ürettiği görülmüştür (Altun ve Şahman, 2013; Chatterjee ve ark., 2017; Basu, 2015). Ayrıca PSO algoritması “Gezgin Satıcı Problemi (Traveling Salesman Problem- (TSP))” gibi ayrık problemler için modifiye edilerek uygulanmıştır (Wang ve ark., 2003; Clerc, 2004; Marinakis ve Marinaki, 2010; Huilian, 2010).

Sürekli problemlerinin çözümü için önerilen PSO algoritması farklı ayrık (TSP etc.) problemlerin çözümü için modifiye edilmiştir. TSP kolay formüle edilebilmesi, zor çözülmesi ve birçok uygulaması olduğundan dolayı araştırmacıların üzerinde çalıştığı önemli ayrık NP-Hard problemlerden birisidir. Huilian FAN, PSO algoritmasını TSP problemlerinde uygulanabilmesi için modifiye etmiştir.

TSP’de, verilen şehirlerin listesine ve şehirlerarasındaki mesafe bilgisine göre her şehre sadece bir kez uğranılarak en kısa muhtemel yol bulunmaya çalışılır (Denklem 3.5).

$$f(x) = \sum_{i=1}^{n-1} d(c_i + c_{i+1}) + d(c_n + c_1) \quad (3.5)$$

$f(x)$, x yoluna ait toplam mesafeyi, $d(c_i + c_{i+1})$ ise c_i ve c_{i+1} arasındaki mesafeyi belirtmektedir. Çözüm sonunda bulunan çözüm bitişik şehirlere ait en kısa yolu gösterecektir. TSP’de çözüm için n adet şehir kullanılmakta ve nxn bir mesafe matrisi ile şehirlerarasındaki uzaklık bilinmektedir.

WSN probleminde n adet düğüm bulunmaktadır. Belirlenen adet kadar KB, n adet düğüm içinden seçilir ve geri kalan düğümler KB’lere tabi olarak Mİ ile iletişime geçerler. TSP’den farklı olarak, WSN probleminde hem KB’ler hem de KB’lere tabi olan düğümler söz konusudur. Dolayısıyla WSN problemlerinde 2 boyutta işlem yapılmaktadır.

4. MERKEZ EĞİLİMLİ KÜMELEME ALGORİTMASI (MEKA)

KAA' da veri iletim protokollerinde, ağın kümelerine ayrılarak seçilen KB üzerinden verinin Mİ' ye iletilmesi son yıllarda kabul görmüş önemli bir yöntemdir. Ağın kümelerine ayrılmasında ise iki farklı yaklaşım vardır.

1. KB'nin Mİ tarafından seçilmesi ve ağın Mİ tarafından oluşturulmasıdır. Bu yöntemin uygulanabilir olması için tüm düğümlerin enerji seviyelerinin ve konumlarının bilinmesi ve bunları Mİ' ye iletebiliyor olmalıdır. Dolayısıyla tüm düğümlerin Mİ ile iletişim kurabiliyor olması gerekmektedir.
2. Diğerisi ise ağın kendi içinde örgütlenerek KB seçmesi ve kümelerini oluşturması yöntemidir. Bu yaklaşım, saha uygulamaları açısından daha verimli ve olması gereken olarak kabul edilmektedir. Bu şekilde çalışan yapılar incelendiğinde, KB seçiminin rastgele seçim yöntemiyle yapıldığı ve seçilen KB'lerin kendi kümelerini oluşturdukları görülmüştür. Bu durumun ise pek çok olumsuz yönü bulunmaktadır.

Bu tez çalışmasında, literatürdeki çalışmaların eksik olan yönleri tespit edilmiş ve düğümlerin kendi içinde örgütlenerek KB seçmelerine ve kümelerini oluşturmalarına imkân sağlayan yeni bir algoritma olarak Merkez Eğilimli Kümeleme Algoritması (MEKA) önerilmiştir. Önerilen algoritma, LEACH ve k-means algoritmaları ile ağın enerji verimliliği, hayatta kalan düğüm sayıları, gönderilen paket sayıları ve kapsama alanları yönünden tartışılmıştır. Bu algoritmaların seçilme nedenleri ise şunlardır;

1. Literatürdeki kümeleme algoritmalarının önemli bir kısmı Mİ yönlendirmeli algoritmalarlardır. MEKA, öz-örgütlü bir yapıya sahip olduğundan bunlarla kıyaslanması doğru değildir. Öz-örgütlü yapıda kümeleme yapabilen algoritmalar önceki bölümlerde anlatılmıştır. Bu algoritmaların ortak özellikleri KB seçimini ve kümelemeyi LEACH algoritmasında belirtilen rastgele seçim modeline göre yapıyor olmalarıdır. O nedenle, çalışmada tartışma için LEACH seçilmiştir.
2. MEKA, KB seçiminde düğümler arasındaki mesafeyi baz almaktadır. Mesafe, düğümler arasındaki RSSI verisinden elde edilebilir ancak tezin deneysel çalışmaları simülasyon ortamında gerçekleştirilmiştir. Simülasyonda ise alana

rastgele dağıtılan noktalar arasında Öklid mesafeleri baz alınmıştır. Literatürde, rastgele dağıtılmış noktaların Öklid mesafesine göre kümelenmesinde k-means algoritması öne çıkmaktadır. k-means algoritması, Mİ kontrolü yönlendirme algoritmaları içinde yer alıyor olmasına rağmen, literatürde kabul edilmiş bir kümeleme algoritması ve kümelemeyi Öklid'e göre yapıyor olmasından dolayı tartışmaya dahil edilmiştir.

MEKA, kurulum aşaması ve kararlı durum aşaması şeklinde iki kısımda değerlendirilir.

- Kurulum aşamasında, düğümler alana serpidikten sonra kendi aralarında kümelenmeleri ve haberleşme ağını kurmaları için gerekli olan adımları içerir.
- Kararlı durum aşaması ise kurulum tamamlandıktan sonra veri iletimlerinin yapıldığı zamanı temsil etmektedir.

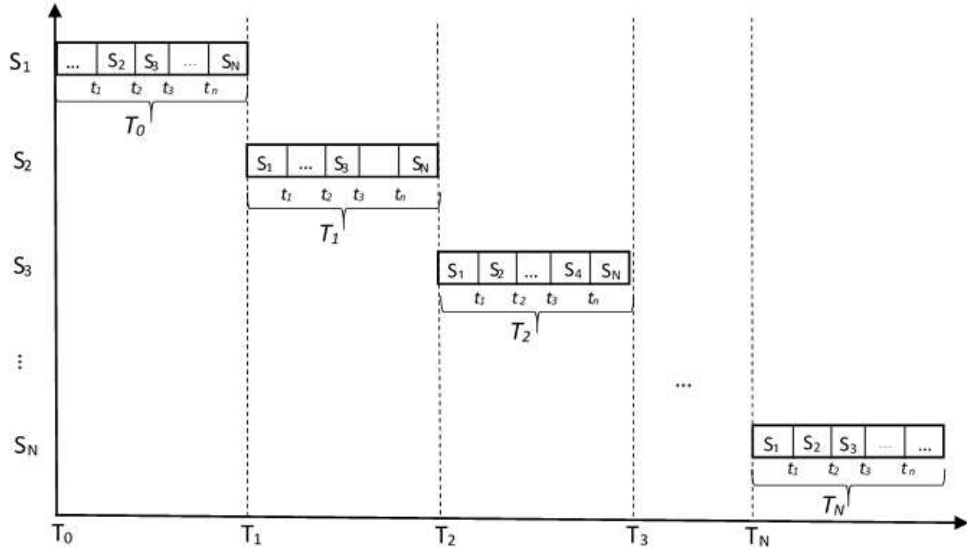
4.1. Kurulum Aşaması

Bu aşamada, düğümler alana düzensiz bir formda ve rastgele serpme yöntemiyle yayılırlar. Serpme işlemi;

- Hava aracından bırakma,
- Fırlatma mekanizması ile atma,
- Elle serpme şekillerinde olabilmektedir.

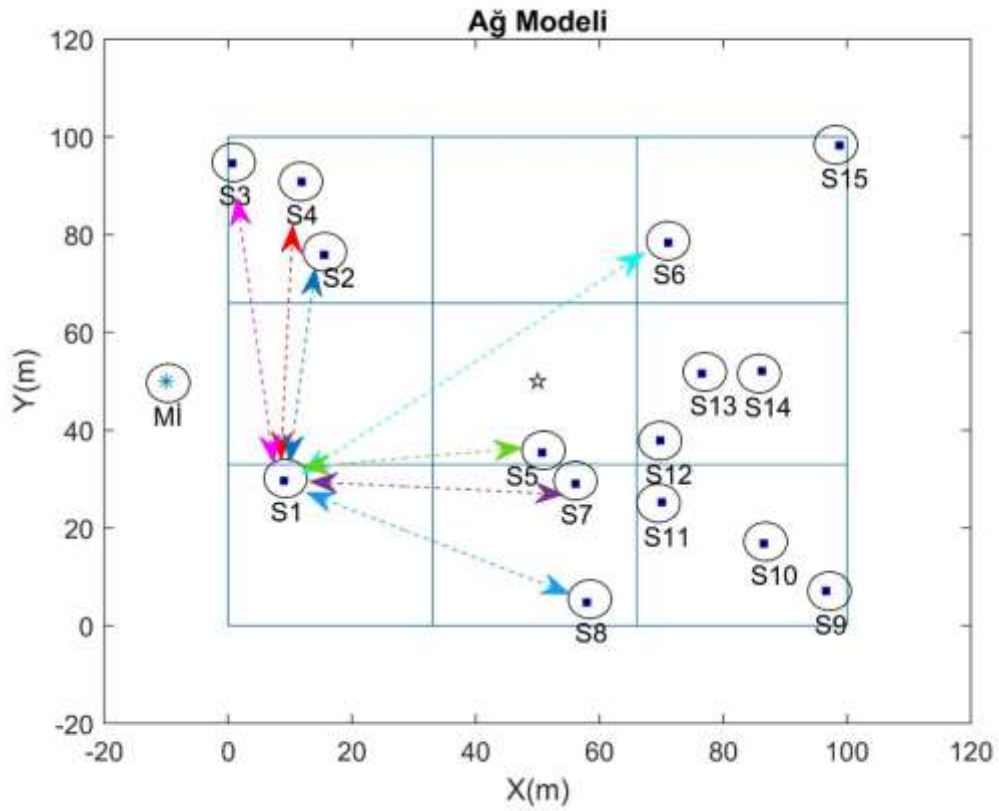
Başlangıçta, düğümler aynı enerji seviyesine sahiptirler ve her düğümün bir kimlik numarası vardır. Bu aşamada yapılacak en önemli iş, düğümler arasındaki mesafenin hesaplanmasıdır.

Her düğüm bir kimlik numarasına (Id) ($NId_1, NId_2, NId_3 \dots, NId_N$) sahiptir ve ağa kendi kimliğinden oluşan bir veri paketi yayınlarlar. Olası çarpışma risklerine karşı haberleşme protokolü olarak TDMA kabul edilmiştir. Örnek TDMA şeması Şekil 4.1'de gösterilmiştir. İlk yayın sorgu yayınıdır ve Id 'si 1 olan düğümden başlayarak büyük olana doğru sırasıyla devam ettirilir.



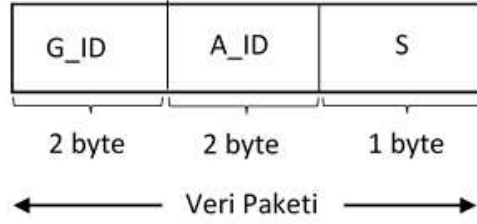
Şekil 4.1. Sorgu yayını TDMA şeması

Örneğin; Şekil 4.2’de gösterilen ve 15 düğümden oluşan ağ modeline göre S_1 düğümü kendine ayrılan T_0 - T_1 zamanı içinde kendi sorgu döngüsünü başlatır ve diğer düğümlerle arasındaki mesafeyi kestirmek için sırasıyla sorgu gönderilir.

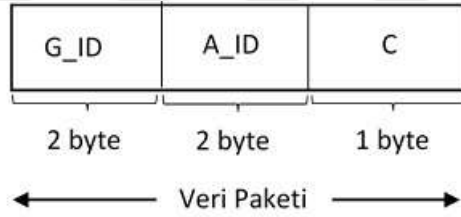


Şekil 4.2. Örnek ağ modeli.

Sorgu yayını, toplam 5 byte'dan oluşan bir veri paketi içerir. İlk 2 byte gönderici düğümün *Id*'sini (G_ID), sonraki 2 byte ise alıcı düğümün *Id*'sini (A_ID) ifade eder. Sonraki 1 byte ise 'S' karakterini içerir. Veri paketi yapısı Şekil 4.3'de gösterilmiştir. Paketteki son byte, sonlandırıcı byte'dır ve herhangi bir karakter olabilir. Bu çalışmada 'S' olarak tercih edilmiştir. (Örn: string sorgu="S1S2S"). Bu paket ağa yayınlanır ve tüm düğümler tarafından alınır. Alıcı *Id*'de belirtilen düğüm bu yayına cevap verir ve Şekil 4.4'de gösterilen paketi yayımlar. Cevap paketindeki G_ID, cevabı veren düğümü, A_ID ise sorgu yayını başlatan düğümü temsil etmektedir. (Örn: string cevap="S2S1C").



Şekil 4.3. Sorgu yayını veri paketi



Şekil 4.4. Cevap yayını veri paketi

Cevap paketini alan düğüm, *Id* ve *RSSI*'dan elde edilen mesafe verilerini kaydedicide saklar.

4.1.1. RSSI'dan mesafe hesaplama

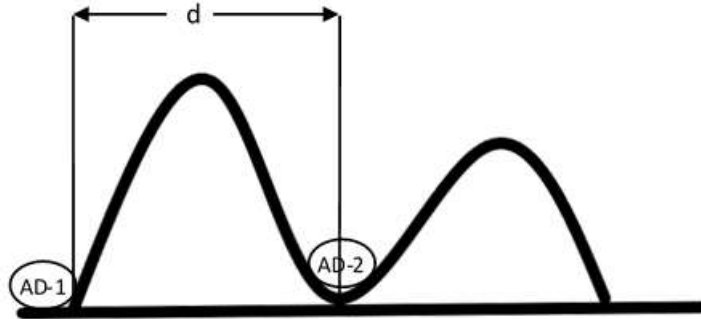
Kurulum aşamasının en önemli bölümü, düğümler arası mesafelerin belirlenmesidir. Düğümlerin coğrafi koordinatlarının bilinmesi durumunda bu işlem daha hızlı bir şekilde ve daha kesin sonuçlarla yapılabilmektedir. Ancak bunun için düğümlerde GPS eklenmesi gibi donanımsal düzenlemeler yapmak gerekmektedir. Bu ise maliyet ve enerji tüketiminin artmasına neden olacağından daha basit bir yolla yapılması gerekmektedir. Alıcı verici birimler arasında gönderilen sinyalin gücünü ifade

eden *RSSI* (Receive Signal Strength Indicator), düğümler arası mesafenin hesaplanması için kullanılabilecek basit ve etkili yollardan biridir. Kullanılan yongaya göre değişmekle birlikte (-255,0) arasında değer alır ve birimi dBm' dir. Bu veri kullanılarak alıcı/verici arasındaki mesafe yaklaşık olarak tahmin edilebilmektedir ve piyasadaki pek çok alıcı/verici ürünlerinde standart olarak sunulan bir özelliktir. Toplanan *RSSI* değerleri Denklem 4.1'de (Botta ve Simek, 2013) verilen fonksiyon ile mesafeye dönüştürülerek arama tablosuna kaydedilir. Arama tablosu, her düğümün diğer düğümler ile arasındaki mesafe değerlerini ve kimliklerini tuttuğu bir diziyi ifade eder. Örnek bir arama tablosu Çizelge 4.1'de gösterilmiştir.

$$d_{i,j} = d_0 \cdot 10^{\left(\frac{RSSI_{d_0} - RSSI_{i,j}}{10 \cdot \eta}\right)} \quad (4.1)$$

Denklem 4.1'de kullanılan η yayılım kaybı katsayısını, $RSSI_{i,j}$ i . düğüm ile j . düğüm arası sinyal gücünü, $d_{i,j}$ i . düğüm ile j . düğüm arası mesafeyi, $RSSI_{d_0}$ ise d_0 referans mesafedeki sinyal gücünü ifade etmektedir. Referans mesafe, alıcının uzaklığı bilinen bir noktaya olan mesafesi anlamına gelmektedir. Bu uzaklık farklı çalışmalarda 1m, 10m ve 100m olarak seçilebilmektedir.

Son yıllarda, *RSSI* ile mesafe tahmini ve konum belirleme üzerine çok sayıda önemli çalışmalar yapılmış, avantaj ve dezavantajları bildirilmiştir. Belirtilen en önemli dezavantajı ise, alıcı/verici arasındaki farklı yansıtıcılar veya engellerden dolayı yayılım kaybının artması ve dolayısıyla sinyal gücünün düşmesi ve buna bağlı olarak aradaki mesafenin hatalı hesaplanmasıdır. Örnek bir durum Şekil 4.5'de gösterilmiştir.



Şekil 4.5. Alana rastgele bırakılan 2 düğüm arasındaki mesafenin örnekleme

Çizelge 4.1. 15 düğüm için oluşturulan arama tablosu (Sarı ile işaretli olan bölgeler düğümlerin kendisine olan mesafesidir)

AD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,000	89,604	93,069	77,007	68,387	88,197	8,480	76,812	80,763	79,591	61,182	28,187	93,030	30,453	48,545
AD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	89,604	0,000	67,098	44,363	26,423	1,748	81,337	13,381	51,001	59,873	74,643	72,842	29,060	77,648	73,983
AD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,000	89,604	93,069	77,007	68,387	88,197	8,480	76,812	80,763	79,591	61,182	28,187	93,030	30,453	48,545
2	89,604	0,000	67,098	44,363	26,423	1,748	81,337	13,381	51,001	59,873	74,643	72,842	29,060	77,648	73,983
3	93,069	67,098	0,000	25,434	77,224	67,638	88,518	60,527	18,017	13,854	35,880	96,731	94,778	64,045	111,270
4	77,007	44,363	25,434	0,000	51,807	44,559	70,925	35,856	7,445	15,553	34,587	75,102	70,503	52,078	87,598
5	68,387	26,423	77,224	51,807	0,000	24,682	59,908	19,863	59,250	66,169	71,866	47,825	26,318	64,179	47,576
6	88,197	1,748	67,638	44,559	24,682	0,000	79,908	12,377	51,315	60,099	74,317	71,201	28,006	76,669	72,238
7	8,480	81,337	88,518	70,925	59,908	79,908	0,000	68,654	75,190	74,817	58,510	21,639	84,600	28,656	42,778
8	76,812	13,381	60,527	35,856	19,863	12,377	68,654	0,000	43,114	51,248	62,986	61,884	34,685	64,298	65,892
9	80,763	51,001	18,017	7,445	59,250	51,315	75,190	43,114	0,000	9,265	32,457	80,888	77,687	54,126	94,206
10	79,591	59,873	13,854	15,553	66,169	60,099	74,817	51,248	9,265	0,000	25,602	82,929	85,931	51,164	97,832
11	61,182	74,643	35,880	34,587	71,866	74,317	58,510	62,986	32,457	25,602	0,000	72,257	95,978	30,765	90,811
12	28,187	72,842	96,731	75,102	47,825	71,201	21,639	61,884	80,888	82,929	72,257	0,000	69,317	46,051	21,309
13	93,030	29,060	94,778	70,503	26,318	28,006	84,600	34,685	77,687	85,931	95,978	69,317	0,000	90,483	61,964
14	30,453	77,648	64,045	52,078	64,179	76,669	28,656	64,298	54,126	51,164	30,765	46,051	90,483	0,000	66,696
15	48,545	73,983	111,270	87,598	47,576	72,238	42,778	65,892	94,206	97,832	90,811	21,309	61,964	66,696	0,000

Şekil 4.5’den da anlaşılacağı üzere bir tepenin iki yanında bulunan AD-1 ve AD-2 düğümleri arasındaki mesafe d ’dir. Fiziksel olarak d mesafesi, kuş bakışı ve engelsiz boş bir ortamda haberleşme için kısa mesafe olarak değerlendirilebilir. Ancak düğümler Şekil 4.5’de olduğu gibi bir tepenin iki yanında olduğunda yayılım kaybı katsayısı yükseleceğinden sinyal gücü düşecek ve aradaki mesafe ona göre hesaplanacaktır. Bu durum, çok atlamalı yönlendirmelerde veya Mİ kontrollü yönlendirmelerde önemli bir problem olarak karşımıza çıkar. Ancak bu durum MEKA algoritması ile kolaylıkla aşılabilmektedir. Fiziksel olarak yakın olsalar bile iki düğüm arasındaki haberleşme kalitesinin zayıf olması düğümlerin farklı kümelerde olmasına neden olacak ve birbirleri ile haberleşmek zorunda kalmayacaklardır.

Yayılım kaybı katsayısı hesaplanırken ortamdaki engeller, engellerin cinsi, kullanılan antenin boyu ve tipi önemlidir. KAA uygulamalarında kullanılan düğümlerin çeşitli kısıtlara sahip olması nedeniyle çok küçük boyutlarda anten kullanılmaması ve düşük enerjiye sahip olmalarından dolayı çıkış güçlerinin düşük olması yayılım kaybını artırmaktadır. Literatürdeki çalışmalarda, yayılım kaybı genellikle 2-6 (birimi yoktur) arasında bildirilmiştir. Bu çalışmada ise yayılım kaybı 2 olarak kabul edilmiştir.

Tüm düğümlere sorgu yayını yapıldıktan ve Çizelge 4.1’deki gibi kendi arama tablosunu oluşturduktan sonra Denklem 4.2’deki fonksiyonu çalıştırarak diğer düğümler ile arasındaki mesafenin ortalamasını hesaplar ve bulduğu değeri ağa yayımlar. Ortalama değerleri ile oluşturulan yeni tablo Çizelge 4.2 verilmiştir.

$$d_{avr} = \begin{cases} \frac{\sum_{i=1}^n d_i}{N} & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases} \quad (4.2)$$

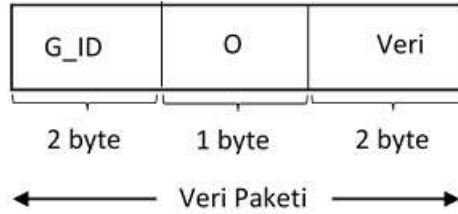
$$Avr_i = \begin{vmatrix} d_{avr_{i_1}} & \dots & d_{avr_{i_n}} \\ d_{i_1} & \dots & d_{i_n} \end{vmatrix} \quad (4.3)$$

Denklemlerdeki d_i düğümün i . düğüme olan mesafesini, N ağdaki düğüm sayısını, d_{avr} düğümün diğer düğümlere olan ortalama mesafesini, Avr_i ise ortalamalardan oluşan mesafe dizisini ifade etmektedir.

Çizelge 4.2. Her düğümün diğer düğümlere olan ortalama uzaklıklarının olduğu arama tablosu

AD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ortalama
1	0,000	89,604	93,069	77,007	68,387	88,197	8,480	76,812	80,763	79,591	61,182	28,187	93,030	30,453	48,545	61,55376
2	89,604	0,000	67,098	44,363	26,423	1,748	81,337	13,381	51,001	59,873	74,643	72,842	29,060	77,648	73,983	50,86688
3	93,069	67,098	0,000	25,434	77,224	67,638	88,518	60,527	18,017	13,854	35,880	96,731	94,778	64,045	111,270	60,93887
4	77,007	44,363	25,434	0,000	51,807	44,559	70,925	35,856	7,445	15,553	34,587	75,102	70,503	52,078	87,598	46,18793
5	68,387	26,423	77,224	51,807	0,000	24,682	59,908	19,863	59,250	66,169	71,866	47,825	26,318	64,179	47,576	47,43173
6	88,197	1,748	67,638	44,559	24,682	0,000	79,908	12,377	51,315	60,099	74,317	71,201	28,006	76,669	72,238	50,19687
7	8,480	81,337	88,518	70,925	59,908	79,908	0,000	68,654	75,190	74,817	58,510	21,639	84,600	28,656	42,778	56,26134
8	76,812	13,381	60,527	35,856	19,863	12,377	68,654	0,000	43,114	51,248	62,986	61,884	34,685	64,298	65,892	44,77182
9	80,763	51,001	18,017	7,445	59,250	51,315	75,190	43,114	0,000	9,265	32,457	80,888	77,687	54,126	94,206	48,98161
10	79,591	59,873	13,854	15,553	66,169	60,099	74,817	51,248	9,265	0,000	25,602	82,929	85,931	51,164	97,832	51,59517
11	61,182	74,643	35,880	34,587	71,866	74,317	58,510	62,986	32,457	25,602	0,000	72,257	95,978	30,765	90,811	54,78946
12	28,187	72,842	96,731	75,102	47,825	71,201	21,639	61,884	80,888	82,929	72,257	0,000	69,317	46,051	21,309	56,54407
13	93,030	29,060	94,778	70,503	26,318	28,006	84,600	34,685	77,687	85,931	95,978	69,317	0,000	90,483	61,964	62,82249
14	30,453	77,648	64,045	52,078	64,179	76,669	28,656	64,298	54,126	51,164	30,765	46,051	90,483	0,000	66,696	53,15411
15	48,545	73,983	111,270	87,598	47,576	72,238	42,778	65,892	94,206	97,832	90,811	21,309	61,964	66,696	0,000	65,51322

Şekil 4.6’da yapısı verilen ve ortalama değerin ağa ilan edildiği bu yayın alıcı ID içermediğinden tüm düğümler tarafından alınır ve alıcı düğümler tarafından gönderici düğümün ID’ si ile birlikte hafızaya Denklem 4.3’deki gibi ifade edilen bir dizi formatında kaydedilir. Dizi örneği Çizelge 4.3’de gösterilmiştir.

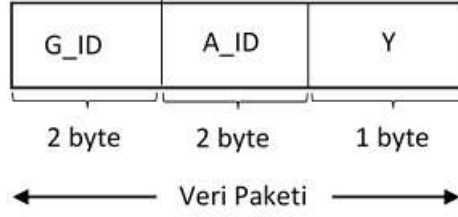


Şekil 4.6. Hesaplanan ortalama yayını veri paketi

Çizelge 4.3. Örnek dizi

AD	Ortalama
1	61,55376
2	50,86688
3	60,93887
4	46,18793
5	47,43173
6	50,19687
7	56,26134
8	44,77182
9	48,98161
10	51,59517
11	54,78946
12	56,54407
13	62,82249
14	53,15411
15	65,51322

Kurulum yayını tamamlayan düğüm, sıradaki düğümün yayınına başlayabilmesi için Şekil 4.7’de gösterilen veri paketini yayınlar ve kendi sürecini tamamlar. Bu pakette A_ID, G_ID’ nin 1 fazlasıdır. Dolayısıyla A_ID’ye sahip düğüm yayını alır almaz kendi yayın sürecini başlatır. Düğümlerin kurulum aşamasında çalıştırdığı algoritmanın kaba kodu Şekil 4.8’de verilmiştir.



Şekil 4.7. Sıradaki düğümün yayına başlayabilmesi için gönderilen veri paketi

```

Program Alıcı/Verici ()
  i ← 0, t_mesafe ← 0, ortalama ← 0
  yayın ← yanlış, N ← düğüm sayısı, ID ← düğüm ID
  GID ←gönderen ID, AID ← alıcı ID
döngü ()
  veri ← Seri port oku
  Eğer veri'nin son byte = 'S' ise
    Seri porta yaz (ID+GID+'C')
  eğer son
  Eğer veri'nin son byte = 'C' ise
    cevap() metoduna git
  eğer son
  Eğer veri'nin son byte = 'Y' veya ID = 1 ise
    yayın ← doğru
  eğer son
  Eğer veri'nin ikinci byte = 'O' ise
    EEPROM'a yaz (GID, ortalama)
  eğer son
  Eğer yayın = doğru ve i<=N ise
    Eğer EEPROM(i) ≠ 0 ise
      Seri porta yaz (ID +AID+S)
      i ← i+1
    değilse
      i ← 0
      ort_kontrol ← doğru
    eğer son
  eğer son
  Eğer ort_kontrol = doğru
    ortalama ← t_mesafe / N
    Seri porta yaz (ID, 'O', ortalama)
    Seri porta yaz (ID, AID,'Y')
    // AID = ID+1
    yayın ← yanlış
  eğer son
döngü son

cevap ()
  Eğer gelen_veri ≠ boşluk ise
    d mesafesini hesapla
    EEPROM'a yaz(i,d)
    t_mesafe ← t_mesafe+ d
  değilse
    EEPROM'a yaz(i, 0)
  eğer son
  geri dön
cevap son
program son

```

Şekil 4.8. Alıcı/verici düğüm kurulum yayını sözde kodu

4.2. KB Seçimi ve Ağın Kümelenmesi

Her düğüm kendine ait d_{avr} değeri ile Avr_i dizisini karşılaştırır. Örneğin, Şekil 4.9'da gösterildiği gibi, 8. düğümün ortalaması diğer düğümlerin içinde en düşük olduğundan KB seçilir. MEKA' da literatürdeki diğer çalışmalardan farklı olarak seçim sonucunun ağa ilan edilmesine gerek yoktur. Düğümler, diğer düğümlerin d_{avr} değerlerini bildikleri için hangi düğümün KB olduğunu bilmektedirler.

AD	Ortalama
1	61,55376
2	50,86688
3	60,93887
4	46,18793
5	47,43173
6	50,19687
7	56,26134
8	44,77182
9	48,98161
10	51,59517
11	54,78946
12	56,54407
13	62,82249
14	53,15411
15	65,51322

AD	Ortalama
1	61,55376
2	50,86688
3	60,93887
4	46,18793
5	47,43173
6	50,19687
7	56,26134
8	44,77182
9	48,98161
10	51,59517
11	54,78946
12	56,54407
13	62,82249
14	53,15411
15	65,51322

Şekil 4.9. KB düğümün seçilmesi

Seçilen bu düğüm ağın ilk KB düğümüdür. KB düğüm seçildikten sonra, kümesine dahil olacak düğümler belirlenir ve kümesini oluşturur. Üye seçiminde, Denklem 4.4'de verilen $Mf(x)$ üyelik fonksiyonu çalıştırılır.

$$Mf(x_i) = \frac{d_{max} - d_{min}}{\sqrt{N}} * \left(\frac{\omega}{p}\right) \quad (4.4)$$

Denklemden, d_{max} KB düğümün kendisine en uzaktaki düğüme olan mesafesini, d_{min} KB düğümün kendisine en yakın düğüme olan mesafesini, N ağdaki düğüm sayısını, ω fonksiyon sabitini ve p küme başı olma yüzdesini ifade etmektedir. ω sabiti

sabit bir sayıdır ve 0-100 arasında seçilebilir. Bu sayının artması veya azalması ters orantılı olarak küme içindeki düğüm sayısını etkilemektedir. p ise ağda kaç düğümün KB olması isteniyorsa ona göre belirlenmelidir. Örneğin, 100 düğümünden oluşan bir ağda düğümlerinin %10'unun KB olması isteniyorsa $p = 10$ olarak verilmelidir.

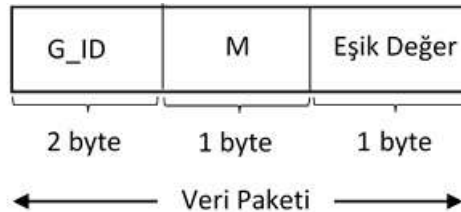
Çizelge 4.1'de verilen örnek arama tablosuna göre, 15 düğümden oluşan bir ağ modelinde 8. düğüm ilk turda KB düğüm seçilmiştir. Buna göre, Çizelge 4.4'de verilen değerler, Denklem 4.4'de yerine konularak Denklem 4.5'de gösterildiği gibi 36,61m değeri elde edilmiştir.

Çizelge 4.4. 8. Düğüm için eşik değer fonksiyonunda kullanılan değerler (d_{max} : 8. düğümün en uzak olduğu düğüme olan mesafesi, d_{min} : 8. düğümün en yakın olduğu düğüme olan mesafesi)

Parametre	Değer
d_{max}	76,812
d_{min}	12,377
N	15
ω	33
p	15

$$Mf(x) = \frac{76,812 - 12,377}{\sqrt{15}} \cdot \frac{33}{15} = 36,61m \quad (4.5)$$

$Mf(x)$ fonksiyonundan elde edilen eşik değeri, hangi düğümün KB'ye üye olup olmayacağını belirlemek için kullanılır. İlk KB olan 8. düğüm, hesapladığı eşik değeri, Şekil 4.10'da gösterilen 4 byte'lık veri paketi ile ağa yayımlar ve tüm düğümler tarafından bu yayım alınır. Yayını alan düğümler KB'ye olan mesafeleri ile eşik değeri karşılaştırırlar. Eğer ($d_{KB} \leq \text{eşik değeri}$) ise o düğüm KB'ye üye olur.



Şekil 4.10. Eşik değeri yayım paketi

Buna göre, 8 numaralı düğüm KB ve 2, 4, 5, 6 ve 13 numaralı düğümler o kümenin üyeleri olarak kabul edilir. Ancak üyelerin hangi düğümler olduklarının ağa ilan

edilmesine gerek yoktur. Örneğin, eşik değeri alan 2. düğüm, 8. düğüme olan mesafesine bakar ve eğer ($d_{KB} \leq \text{eşik değeri}$) ise kendisini KB'ye dahil eder.

Bu durum Çizelge 4.5'de açıkça gösterilmektedir. Tüm düğümler ($d_{KB} \leq \text{eşik değeri}$) koşulunu kendi tablolarında kontrol ettiklerinden dolayı şartı sağlayıp sağlayamadığını bildiklerinden ağa ilan edilmezler. Şartı sağlayamayan düğümler o turda oluşan kümeye üye olamaz, diğer tura kalırlar ve kendi aralarında tekrar kümelenmeye başlarlar. Düğümler, 1. turda KB ve ona bağlanan düğümleri, diğer turun başında Çizelge 4.6'da gösterildiği gibi kendi tablolarından çıkarır ve yeni tabloya göre KB seçimi yaparlar. Çıkarılan kısımlar "X" ile işaretlenmiştir.



Çizelge 4.5. 1. Turda KB seçilen düğümün üyelerinin belirlenmesi

AD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Ortalama
1	0,000	89,604	93,069	77,007	68,387	88,197	8,480	76,812	80,763	79,591	61,182	28,187	93,030	30,453	48,545	61,55376
2	89,604	0,000	67,098	44,363	26,423	1,748	81,337	13,381	51,001	59,873	74,643	72,842	29,060	77,648	73,983	50,86688
3	93,069	67,098	0,000	25,434	77,224	67,638	88,518	60,527	18,017	13,854	35,880	96,731	94,778	64,045	111,270	60,93887
4	77,007	44,363	25,434	0,000	61,807	44,559	70,925	35,856	7,445	15,553	34,587	75,102	70,503	52,078	87,598	46,18793
5	68,387	26,423	77,224	51,807	0,000	24,682	59,908	19,863	59,250	66,169	71,866	47,825	26,318	64,179	47,576	47,43173
6	88,197	1,748	67,638	44,559	24,682	0,000	79,908	12,377	51,315	60,099	74,317	71,201	28,006	76,669	72,238	50,19687
7	8,480	81,337	88,518	70,925	59,908	79,908	0,000	68,654	75,190	74,817	58,510	21,639	84,600	28,656	42,778	56,26134
8	76,812	13,381	60,527	35,856	19,863	12,377	68,654	0,000	43,114	51,248	62,986	61,884	34,685	64,298	65,892	44,77182
9	80,763	51,001	18,017	7,445	59,250	51,315	75,190	43,114	0,000	9,265	32,457	80,888	77,687	54,126	94,206	48,98161
10	79,591	59,873	13,854	15,553	66,169	60,099	74,817	51,248	9,265	0,000	25,602	82,929	85,931	51,164	97,832	51,59517
11	61,182	74,643	35,880	34,587	71,866	74,317	58,510	62,986	32,457	25,602	0,000	72,257	95,978	30,765	90,811	54,78946
12	28,187	72,842	96,731	75,102	47,825	71,201	21,639	61,884	80,888	82,929	72,257	0,000	69,317	46,051	21,309	56,54407
13	93,030	29,060	94,778	70,503	26,318	28,006	84,600	34,685	77,687	85,931	95,978	69,317	0,000	90,483	61,964	62,82249
14	30,453	77,648	64,045	52,078	64,179	76,669	28,656	64,298	54,126	51,164	30,765	46,051	90,483	0,000	66,696	53,15411
15	48,545	73,983	111,270	87,598	47,576	72,238	42,778	65,892	94,206	97,832	90,811	21,309	61,964	66,696	0,000	65,51322

Çizelge 4.6. 1. Turda kümelenen düğümlerin çıkarılması ve 2. tur KB seçimi

AD	1	X	3	X	X	X	7	X	9	10	11	12	X	14	15	Ortalama
1	0,000	X	93,069	X	X	X	8,480	X	80,763	79,591	61,182	28,187	X	30,453	48,545	61,55376
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	93,069	X	0,000	X	X	X	88,518	X	18,017	13,854	35,880	96,731	X	64,045	111,270	60,93887
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	8,480	X	88,518	X	X	X	0,000	X	75,190	74,817	58,510	21,639	X	28,656	42,778	56,26134
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	80,763	X	18,017	X	X	X	75,190	X	0,000	9,265	32,457	80,888	X	54,126	94,206	48,98161
10	79,591	X	13,854	X	X	X	74,817	X	9,265	0,000	25,602	82,929	X	51,164	97,832	51,59517
11	61,182	X	35,880	X	X	X	58,510	X	32,457	25,602	0,000	72,257	X	30,765	90,811	54,78946
12	28,187	X	96,731	X	X	X	21,639	X	80,888	82,929	72,257	0,000	X	46,051	21,309	56,54407
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14	30,453	X	64,045	X	X	X	28,656	X	54,126	51,164	30,765	46,051	X	0,000	66,696	53,15411
15	48,545	X	111,270	X	X	X	42,778	X	94,206	97,832	90,811	21,309	X	66,696	0,000	65,51322
9	80,763	X	18,017	X	X	X	75,190	X	0,000	9,265	32,457	80,888	X	54,126	94,206	48,98161

1.Turda kümelenen düğümlerin dışında kalan düğümler, yeni bir küme oluşturmak için yeni bir KB seçerler. Çizelge 4.6'dan anlaşılacağı üzere, 9 numaralı düğüm diğer düğümler içinde ortalaması en düşük olduğundan 2. turda KB seçilmiştir. Buna göre, Çizelge 4.7'de verilen değerler, Denklem 4.4'de yerine konulduğunda Denklem 4.6'da gösterildiği gibi 48,26m değeri elde edilmiştir. Bu değer 9. Düğümün eşik değeridir ve buna göre kümesini oluşturacaktır.

Çizelge 4.7. 9. Düğüm için eşik değer fonksiyonunda kullanılan değerler (d_{max} : 9. düğümün en uzak olduğu düğüme olan mesafesi, d_{min} : 9. düğümün en yakın olduğu düğüme olan mesafesi)

Parametre	Değer
d_{max}	94,206
d_{min}	9,265
N	15
ω	33
p	15

$$Mf(x) = \frac{94,206 - 9,265}{\sqrt{15}} \cdot \frac{33}{15} = 48,26 \quad (4.6)$$

Elde edilen yeni eşik değer, ilk turda olduğu gibi tekrar ağa ilan edilir. Buna göre, 3,10 ve 11 nolu düğümler 9. Düğümün kümesine dahil olurlar. Bu durum Çizelge 4.8'de gösterilmiştir. Buna göre, ilk 2 turda seçilen KB'ler ve onlara üye olan düğümler ve o turda küme dışında kalan düğümler Çizelge 4.9'da verilmiştir.

Çizelge 4.8. İlk 2 turda kümelenen düğümler

KB Düğüm	Üyeler	Kalan Düğümler
8	2,4,5,6,13	1,3,7,10,11,12,14,15
9	3,10,11	1,7,12,14,15

Ağdaki düğümlerin tamamı kümeleninceye kadar bu işlem iterasyonlar halinde devam eder. Ağın kümelenme işlemi tamamlandıktan sonra ağ kararlı durum aşamasına geçer ve veri algılama ve gönderme işlemleri başlatılır. Şekil 4.11'de MEKA algoritmasının kaba kodu verilmiştir.

Çizelge 4.9. 2. Turda KB seçilen düğümün üyelerinin belirlenmesi.

AD	1	X	3	X	X	X	7	X	9	10	11	12	X	14	15	Ortalama
1	0,000	X	93,069	X	X	X	8,480	X	80,763	79,591	61,182	28,187	X	30,453	48,545	61,55376
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3	93,069	X	0,000	X	X	X	88,518	X	18,017	13,854	35,880	96,731	X	64,045	111,270	60,93887
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
7	8,480	X	88,518	X	X	X	0,000	X	75,190	74,817	58,510	21,639	X	28,656	42,778	56,26134
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9	80,763	X	18,017	X	X	X	75,190	X	0,000	9,265	32,457	80,888	X	54,126	94,206	48,98161
10	79,591	X	13,854	X	X	X	74,817	X	9,265	0,000	25,602	82,929	X	51,164	97,832	51,59517
11	61,182	X	35,880	X	X	X	58,510	X	32,457	25,602	0,000	72,257	X	30,765	90,811	54,78946
12	28,187	X	96,731	X	X	X	21,639	X	80,888	82,929	72,257	0,000	X	46,051	21,309	56,54407
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
14	30,453	X	64,045	X	X	X	28,656	X	54,126	51,164	30,765	46,051	X	0,000	66,696	53,15411
15	48,545	X	111,270	X	X	X	42,778	X	94,206	97,832	90,811	21,309	X	66,696	0,000	65,51322

```

1: Program MEKA
2:  $i \leftarrow 0$ ;
3:  $n \leftarrow$  Düşüm Sayısı
4: Donanımsal Kontrol  $NId_i$ 
5: Eğer  $NId_i$  herhangi bir kümeye üye ise 27. Adıma Git
6: Tüm düşümlerden  $RSSI$  değerlerini oku
7:  $ort \leftarrow \frac{\sum_{m=0}^n RSSI_m}{n}$ 
8: Eğer  $ort = 0$  ise 27. Adıma Git
9:  $j \leftarrow 0$ 
10:  $RSSI'$  dan  $d_j$  hesapla
11: Ortalama  $d_{avr_i}$  bul
12: Eğer  $j < n$  ise  $j++$  ve 9. Adıma Git
13: Ağa  $d_{avr_i}$  yayınla
14: Diğer düşümlerden  $d_{avr}$  oku
15:  $k=0$ 
16: Eğer  $d_{avr_i} < d_{avr_k}$  ise
17:      $NId_i$  küme başıdır (KB)
18: Değilse
19:     KB düşümden davet bekle
20: Eğer  $d_{avr_i} < \lceil KB \rceil_{avr}$  ise
21:     KB'ye dahil ol
22: Eğer  $k < n$  ise 15. Adıma Git
23:  $m \leftarrow 0$ 
24: Eğer  $d_{avr_m} < M \lceil f(x) \rceil_i$  ise
25:     Node  $m$  KB kümesine dahil ol
26: Eğer  $m < n$  ise
27:      $m++$  ve 23. Adıma Git
28: Eğer  $i < n$  ise
29:      $i++$  ve 2. Adıma Git
30: Program Son

```

Şekil 4.11. MEKA algoritması kaba kodu.

4.3. Radyo Modeli

Bu tezde, alıcı-verici arasındaki yol kaybı katsayısı göz ardı edilerek sadece iletim çıkış gücünün dikkate alındığı iki ışın yansıma modeli kullanılmıştır. Bu modelde, alıcı ve vericilerin haberleşmede harcadıkları enerji, gönderilen verinin büyüklüğü ve iletim mesafesine bağlıdır. KAA' da iletim mesafesi, yönlendirme protokolüne veya algoritmasına bağlı olarak değişmektedir. Atlama kullanmadan direk haberleşme yapan sistemlerde, her düşümün iletim mesafesi M_i ' ye olan uzaklığıdır. Atlamalı modellerde ise iletim mesafesi verinin atlayacağı düğüme olan uzaklığı kabul edilir. Kümeleme modelinde ise iletim mesafesi 2 farklı şekilde ele alınmaktadır. Bunlar;

- Düşümlerin KB düğüme olan mesafeleri
- KB düşümün M_i ' ye olan mesafesi

Herhangi bir kümeye dahil olmadan yalnız kalan düğümler ise Mİ ile direk haberleşme kuracaklarından iletim mesafeleri Mİ' ye olan mesafeleri kabul edilmektedir. Radyo modelinde gönderme için harcanan enerji Denklem 4.11 ile hesaplanmaktadır.

$$E_{TX} = (E_{elec} * l) + (E_{amp} * l * d_x^2) \quad (4.11)$$

Burada E_{elec} , elektronik devrelerin harcadığı enerjiyi, l bit cinsinden gönderilen verinin büyüklüğünü, E_{amp} yükselteç biriminin harcadığı enerjiyi ve d_x ise mesafeyi ifade etmektedir. Veri almak için harcanan enerji ise Denklem 4.12 ile hesaplanmaktadır.

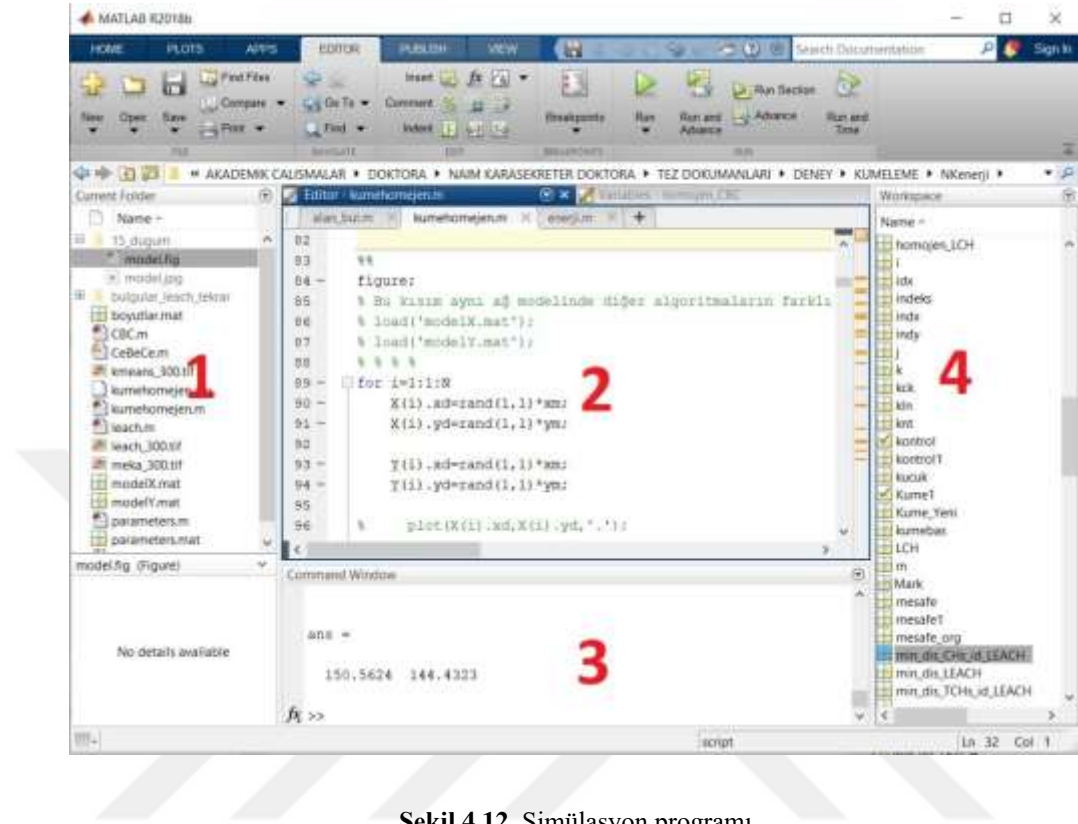
$$E_{RX} = (E_{elec} * l) \quad (4.12)$$

Denklem 4.11 ve Denklem 4.12'ye göre hesaplanan enerji tüketimleri her veri gönderme ve alma sürecinde çalıştırılır ve düğümlerin mevcut enerji değerlerinden harcanan enerjinin çıkarılması ile düğümlerin enerji seviyeleri güncellenir. Düğümlerdeki enerji seviyesi, gönderme için harcanan enerjinin altına düştüğünde o düğüm ölü olarak kabul edilir ve çevrimden çıkarılır. Bu durum üye düğümler ve KB düğümler için aynı şekilde çalıştırılır. Ancak KB düğümlerin transfer yükü daha fazla olduğundan enerjileri diğerlerine göre daha hızlı tükenir ve ölürlür. KB düğümler öldüklerinde ağ tekrar kümelenmeye geçer ve yeni KB düğümler seçilir. Bu durum, ağdaki düğümlerin enerjileri tükeninceye kadar iterasyonlar halinde devam ettirilir.

4.4. Simülasyon Programı

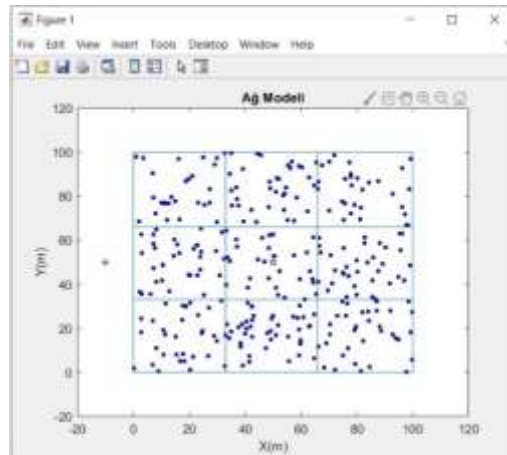
Algoritmaların karşılaştırmalı olarak analiz edilebilmesi için Matlab ortamında bir simülasyon programı geliştirilmiştir. Matlab; C, C++ ve Java gibi dillerle etkileşim kurabilen, o dillerde yazılmış kütüphaneleri ve fonksiyonları kullanabilen bir programlama dilidir. Mühendislik alanlarında, sistem analizi ve matematiksel hesaplamaların kolayca yapılabildiği bir platformdur. Program ara yüzü Şekil 4.12'de verilmiştir. Ara yüzde, 1 numaralı kısım mevcut dosya penceresi, 2 numaralı kısım ise kod editörü penceresidir. 3 numaralı kısım, komut penceresidir. Matlab'da, yazılan kodun dışında farklı bir komut/komutların çalıştırılmasına izin verilmektedir. Yeni bir

dosya ya da proje oluşturmaya gerek yoktur. 4. Kısımda ise, matrislerin ve değişkenlerin görüntülediği kısımdır.



Şekil 4.12. Simülasyon programı

Düğümün serpildiği alan olarak ise şekil pencereleri kullanılmış ve tüm gösterimler burada verilmiştir. Örnek bir şekil penceresi Şekil 4.13’de verilmiştir. Burada (100,100) koordinatları arasında kalan kısım 100m*100m’lik bir alan olarak kabul edilmiş ve düğümler bu alan içine serpilmiştir.



Şekil 4.13. Örnek şekil penceresi

LEACH, MEKA ve k-means algoritmalarının KB seçimi ve kümelene karakteristilerinin görülmesi için “Leach.m” fonksiyonu yazılmış ve çalıştırılmıştır. Programda ilk olarak belirtilen sayıda alana rastgele dağıtılan düğümlerden oluşan bir ağ modeli oluşturulur. Kod örneği Şekil 4.14’de verilmiştir. Daha sonra ilk olarak LEACH algoritmasında KB seçimi ve kümeleme işlemleri tamamlanır. Onu MEKA ve k-means algoritmaları takip eder. MEKA algoritması için “CBC.m” adında başka bir fonksiyon yazılmış buradan çağırarak suretiyle kullanılmıştır.

```

for i=1:1:N
    X(i).xd=rand(1,1)*xm;
    X(i).yd=rand(1,1)*ym;

    Y(i).xd=rand(1,1)*xm;
    Y(i).yd=rand(1,1)*ym;

    plot(X(i).xd,X(i).yd, '.');

    plot(X(i).xd,X(i).yd, '--gs', ...
        'LineWidth',1, ...
        'MarkerSize',3, ...
        'MarkerEdgeColor','b', ...
        'MarkerFaceColor',[0 0 0])

    title('Ağ Modeli');
    xlabel('X(m)');
    ylabel('Y(m)');
    hold on;
end

```

Şekil 4.14. Örnek bir ağ modeli oluşturmak için kullanılan kod bloğu

Yukarıda sayılan işlemler tamamlandıktan sonra oluşturulan kümelere göre düğümlerin enerji tüketimleri hesaplanır ve düğümlerin belirli bir süre (*Steady state time*) iletim yaptıkları kabul edilir. Bu zaman dolduğunda ağ tekrar kümelenemeye başlar ve bu süreç bir iterasyon olarak kabul edilir. Program içinde iterasyon sayısı önceden bir değişken içinde belirtilir ve program iterasyon sayısı kadar ağı çalıştırır.

İterasyonlar tamamlandığında, ağın iterasyonlara göre tüketim değerlerindeki değişim, paket sayıları, ölen ve hayatta kalan düğümlerin değişimleri vb. tüm sonuçlar elde edilir. Daha sonra, KB düğümlerin kapsama alanlarının hesaplanması için yazılan “kumehomojen.m” ve “alanbul.m” fonksiyonları kullanılarak kapsama alan değerleri bulunur. Hesaplama metodu önceki bölümlerde detaylı şekilde anlatılmıştır.

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

LEACH, k-means ve MEKA algoritmaları önceki bölümlerde detayları ile anlatılmıştır. Bu algoritmalar için MATLAB’da bir simülasyon programı yazılmıştır. Bu simülasyon programında;

- Algoritmaların kümelenme karakteristikleri,
- Kümelerin kapsama alanları,
- Belirli bir iterasyon sonunda hayatta kalan ve ölen düğümlerin dağılımları,
- Gönderilen toplam veri paketi sayıları,
- Ağın enerji verimlilikleri analiz edilmiştir.

Simülasyonda, alana rastgele serpmeye yöntemiyle yerleştirilmiş farklı düğüm sayılarından oluşan ağ yapıları modellenmiştir. Geliştirilen MEKA algoritması, tüm yönleriyle literatürde öz-örgütlü yapıda kümeleme yapan tek algoritma olan LEACH algoritması ile karşılaştırılmış, öz-örgütlü yapıda olmamasına rağmen kümeleme karakteristiği açısından benzerlik gösterdiği için k-means algoritması ile de küme dağılımları ve kapsama alanları yönünden karşılaştırmalı olarak analiz edilmiştir.

5.1. Algoritmaların Küme Oluşumu ve Dağılımı Açısından Değerlendirilmesi

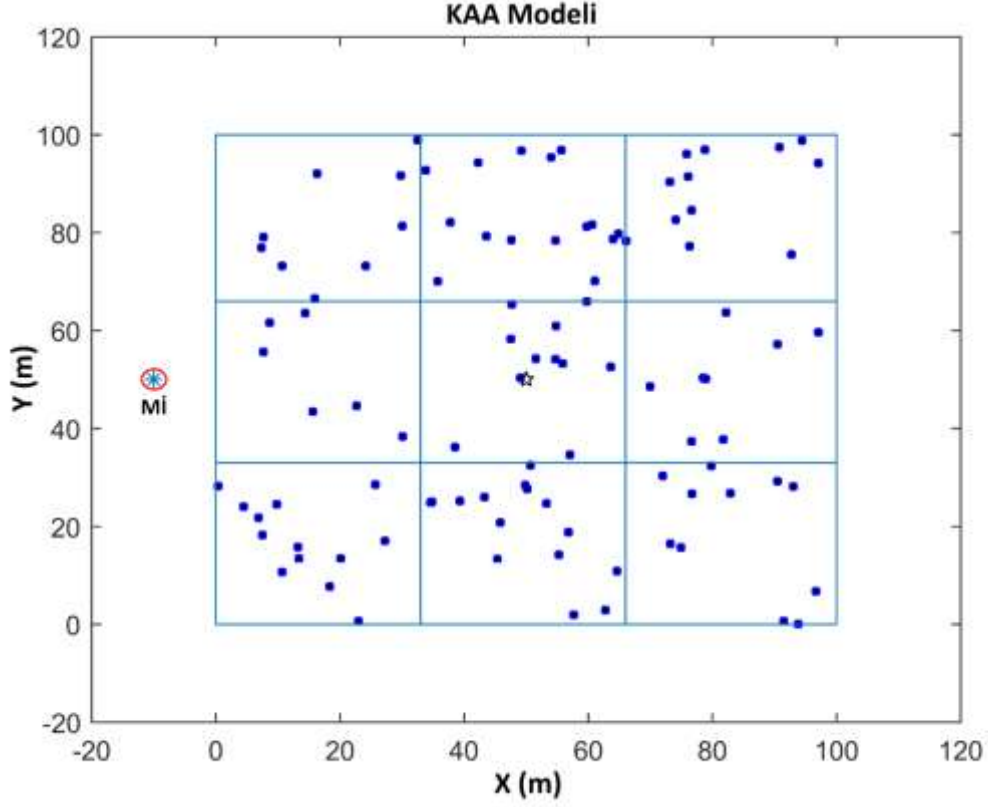
Simülasyon programında açılan form ekranı, 100m*100m bir alan olarak kabul edilmiş ve rastgele noktalar yerleştirilerek bu noktalar da düğümler olarak kabul edilmişlerdir. Tüm düğümlerin aralarında haberleşebildikleri var sayılmış ve düğümler arasındaki mesafe Öklid teoremine göre hesaplanmıştır. İlk olarak, algoritmalar aynı ağ modeli üzerinde çalıştırılarak KB seçimlerine bakılmıştır. KB düğümlerin alandaki homojen dağılımı, ağın ömrünü olumlu yönde etkileyeceğinden önemlidir.

KB düğümlerin alandaki dağılımlarına toplam kapsama alanları hesaplanarak bakılmıştır. Her KB düğümün 20m çapındaki bir alanda bulunan düğümlerle haberleşebildiği var sayılmış ve her KB düğüm etrafına 20m çapında daireler çizilerek alanları hesaplanmıştır. Bu alanların toplamına göre KB dağılımlarının başarımları ölçülmüştür. Çizelge 5.1’de algoritmaların kümelemede ve alan hesabında kullandıkları parametreler verilmiştir.

Çizelge 5.1 Ağın kümelenmesinde kullanılan parametreler ve değerler

Parametre	Değer
Düğüm sayısı	100 adet
Alan	100m x 100m
Her düğümün tahmini alma-gönderme mesafesi	20m
MEKA ve LEACH için küme başı olama yüzdesi (p)	%10
k-means ağırlık merkezi sayısı (k)	10
MEKA için $Mf(x)$ fonksiyon sabiti (ω)	22,2

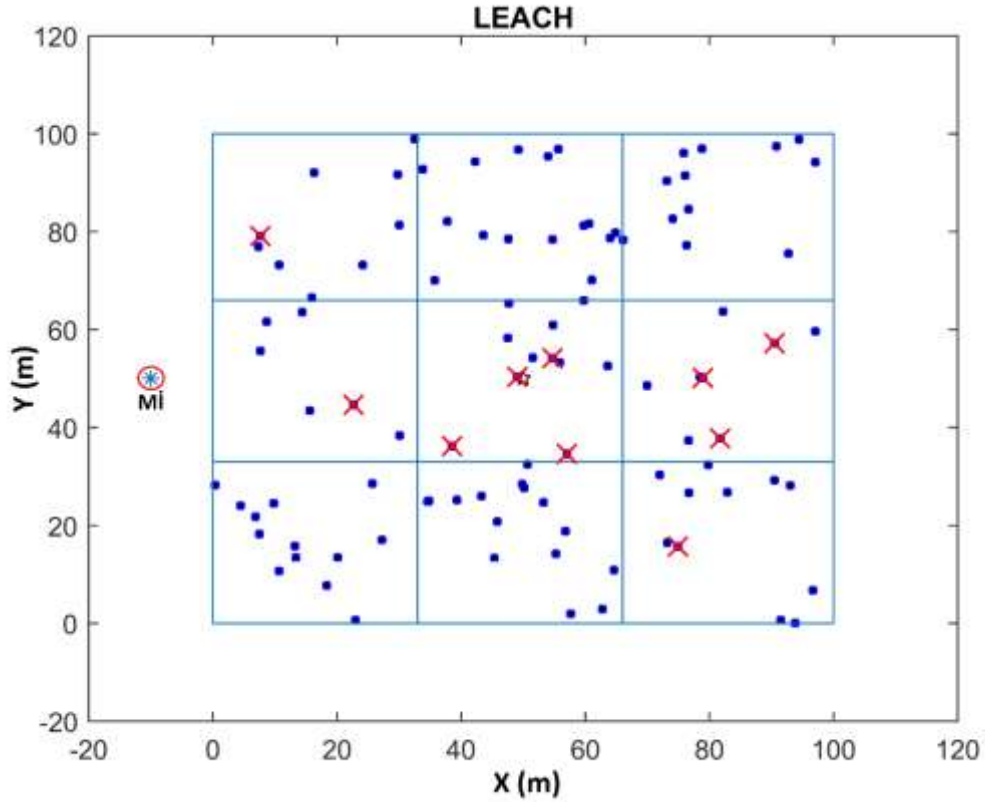
Burada; p , MEKA ve LEACH algoritmalarında küme başı yüzde değeri olarak kullanılmaktadır. Örneğin; $p = 10$ verildiğinde, 100 düğüm içinden 10 tane düğümün KB seçilmesi beklenmektedir. Bu değer arttırıldığında, örneğin %30 verildiğinde 30 düğümün KB olması beklenir. KB sayısının fazla olması enerji verimliliği açısından uygun değildir. Bu sayının azaltılması durumunda ise örneğin %5 verildiğinde 5 düğümün KB olması beklenir. Bu durumda ise KB başına düşecek olan düğüm sayısı artmaktadır. Literatürde ise, genel olarak bu oran 10 olarak kabul edilmiştir. Bu sebeplerden dolayı, uygulama sonuçlarının anlamlı olması adına $p = 10$ olarak alınmıştır. Alandaki toplam düğüm sayısına göre optimum KB ve üye sayılarının belirlenmesi ise farklı bir çalışma konusu olarak ele alınabilir. k değeri ise ağırlık merkezi sayısıdır. k-means algoritmasında ağın kaç kümeye ayrılması isteniyorsa k değeri ona göre ayarlanmaktadır. Algoritmaların karşılaştırmasında eşit şartların sağlanabilmesi için $k=10$ olarak alınmıştır. Düğümlerin alma/gönderme mesafeleri ise varsayılmış ve 20m olarak kabul edilmiştir. ω değeri ise, farklı ağ modellerinde yapılan denemeler sonucunda optimum değer olarak belirlenmiş ve 22.2 olarak kabul edilmiştir. Buna göre, algoritmalar Şekil 5.1’de verilen ağ modeli üzerinde çalıştırılmış ve KB seçim davranışları aşağıda gösterilmiştir.



Şekil 5.1. Ağ modeli

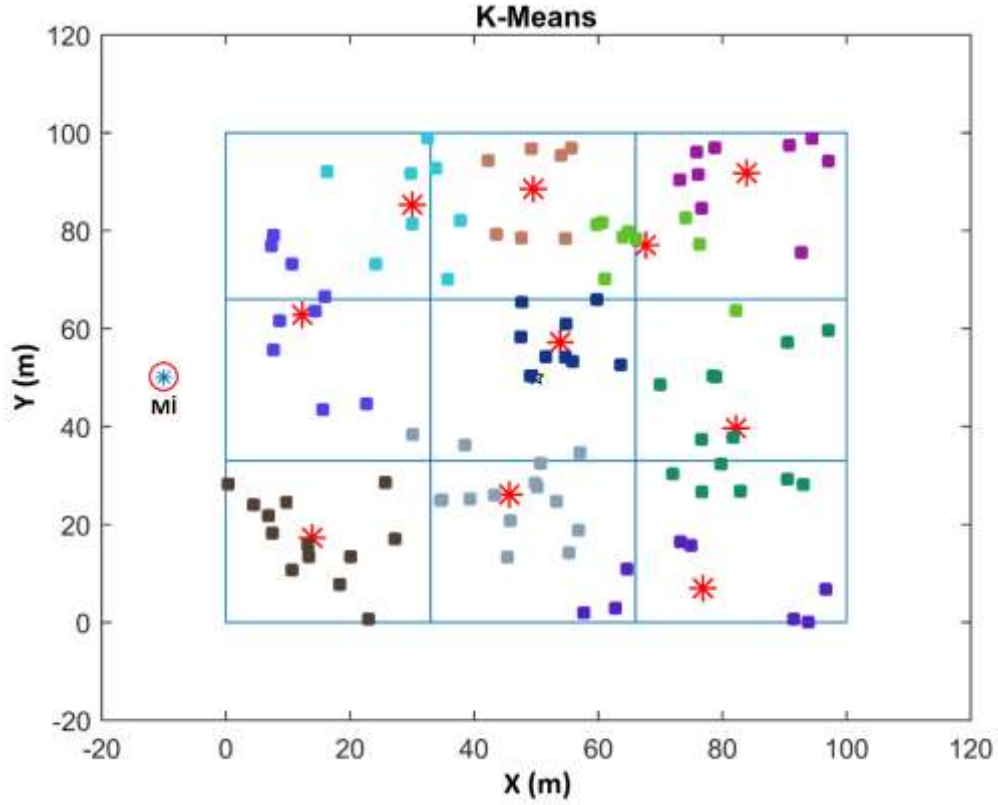
KB'lerin ağ içerisindeki dağılımlarını net bir şekilde gözlemlemek için alan 9 bölgeye ayrılmıştır. Alanın büyüklüğüne göre bu sayı artırılabilir ya da azaltılabilir. Kümeleme işleminden sonra algoritmaların her bölgeden en az 1 KB belirlemesi beklenir. Bu sayede görsel olarak KB düğümlerin alandaki eşit dağılım durumları belirlenebilmektedir. Düzensiz dağıtık yapıdan dolayı düğüm yoğunluklarının belirli bölgelerde toplanması durumunda ise algoritmaların yoğunluğa göre ağı dengeli bir şekilde kümelemesi beklenmektedir.

Şekil 5.2 incelendiğinde, düğüm dağılımlarının üst ve alt bölümlerde yoğunlaşmış olmasına rağmen LEACH algoritması 10 adet KB'yi yoğunluğun daha az olduğu orta kısımlarda ve birbirine çok yakın yerlerde seçmiştir. LEACH, KB seçimini rastgele yaptığından dolayı düğümlerin ağ içerisindeki dağılım yoğunluğunu göz ardı etmiş ve 4 bölge için KB seçimi yapamamıştır.



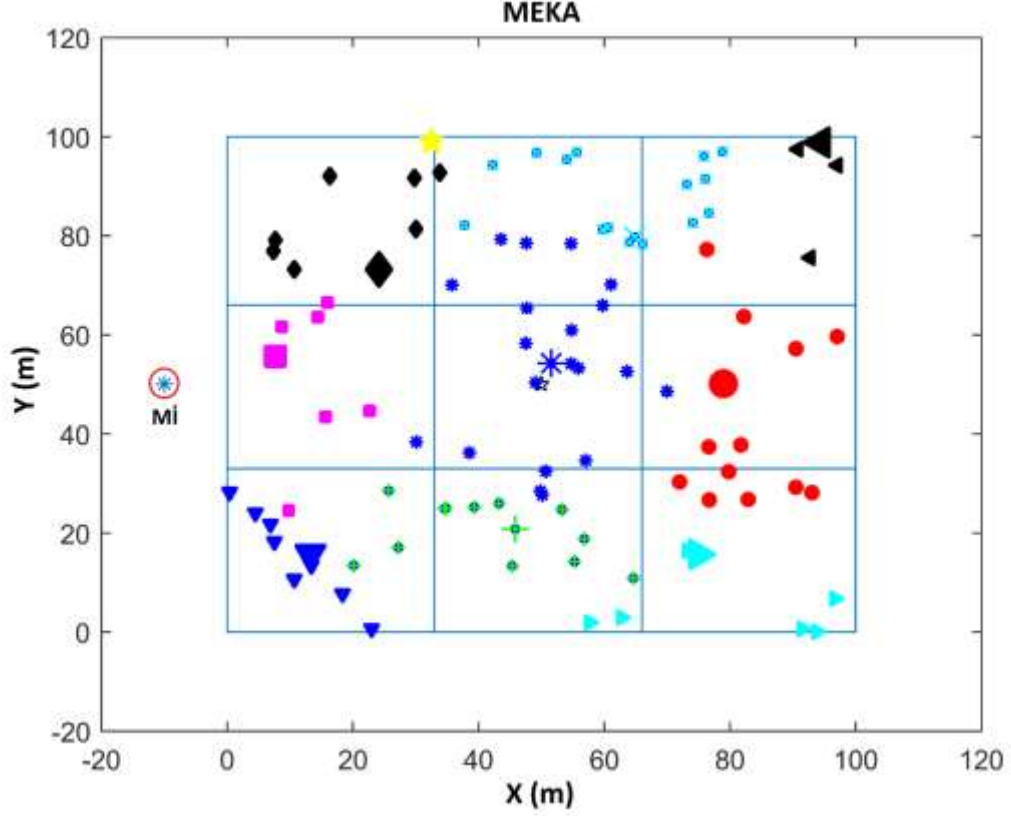
Şekil 5.2 LEACH algoritması KB seçimi (X: KB düğümler)

Şekil 5.3 incelendiğinde ise, ağırlık merkezi prensibine dayalı olarak çalışan k-means algoritmasının her bölgede ağırlık merkezi hesapladığı görülmektedir. Ancak belirlenen bu ağırlık merkezleri ağ içerisindeki herhangi bir düğümü ifade etmemektedir. k-means, ilk olarak rastgele seçtiği ağırlık merkezlerini en iyiyi buluncaya kadar iterasyonlar halinde günceller. En son iterasyonda en iyi ağırlık merkezini seçer ve etrafındaki noktaları ona göre kümeler. Buna göre, ağırlık merkezinin gerçek bir düğümü temsil etmemesi ve kümelene aşamasındaki iteratif yapısından dolayı k-means' in KAA' daki kümeleme karakteristiğini öz-örgütlü yapıdan çıkarır ve Mİ yönlendirmeli kategorisine koyar. k-means' de, ağırlık merkezine en yakın düğüm KB seçilebilir. Ancak bu işlem için de Mİ' ye ihtiyaç duyulmaktadır. Bu durum ise k-means algoritmasının öz örgütlü yapıda olmadığını açıkça göstermektedir.



Şekil 5.3. k-means algoritması KB seçimi (*: Ağırlık merkezleri)

Şekil 5.4’de görüldüğü gibi MEKA algoritmasının aynı ağ modeli için her bir bölgede gerçek bir düğümü KB seçtiği görülmektedir. Toplam 10 KB seçerek ağın kümelenmesi gerçekleştirilmiştir. Her küme farklı renkler ve semboller ile ayrılmıştır. Kümede büyük sembol ile gösterilen düğüm KB düğümdür. Algoritmaların KB seçim ve kümelenme karakteristikleri pek çok farklı ağ modelinde denenmiştir. KB seçimindeki başarımlar ise daha önce de bahsedildiği gibi KB düğümlerin alanda oluşturdukları toplam kapsama alanı hesaplanarak belirlenmeye çalışılmıştır.



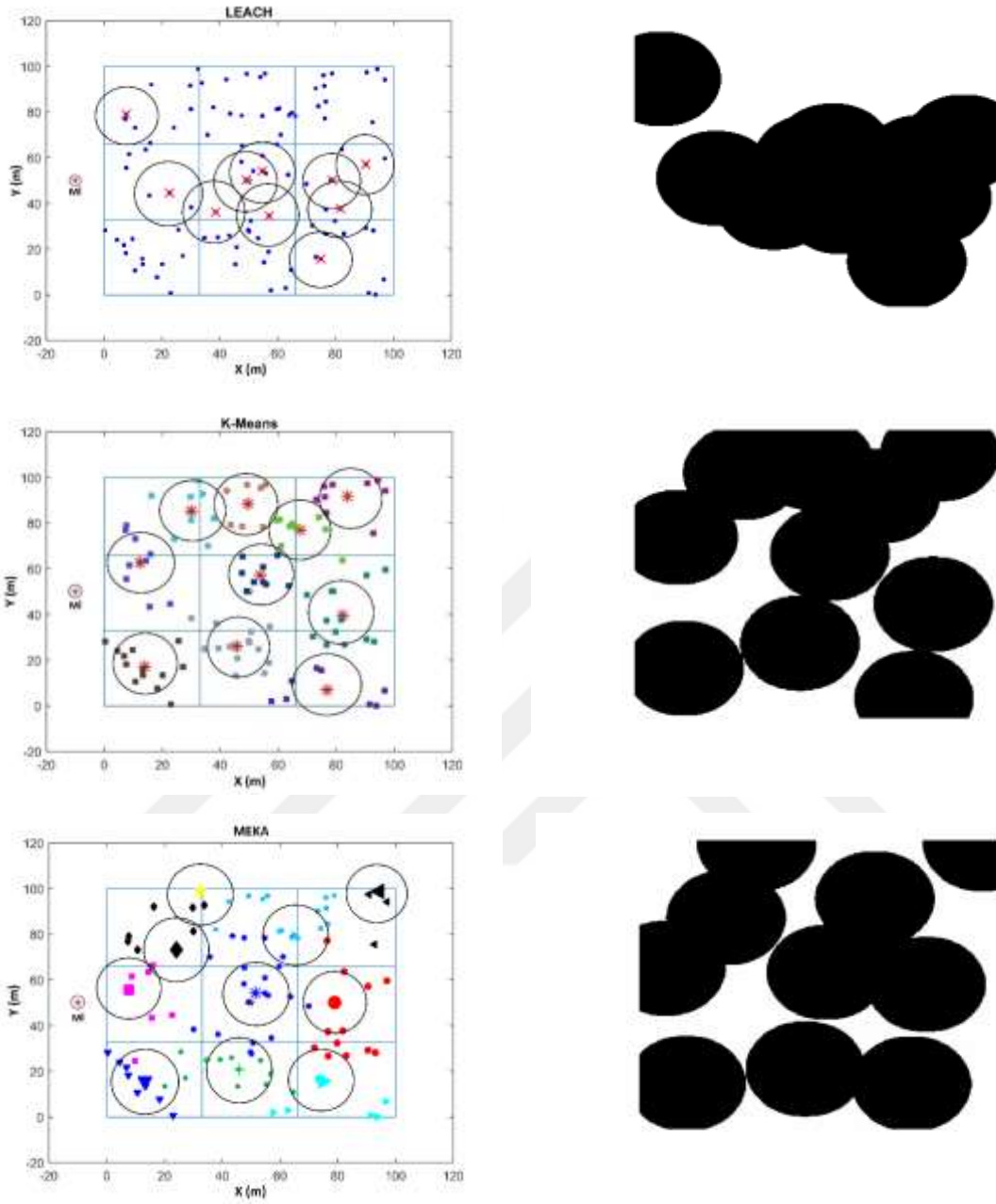
Şekil 5.4. MEKA algoritması KB seçimi

5.2. Kapsama Alanlarının Değerlendirilmesi

KB'lerin alandaki homojen dağılımlarını ve kümelerin kapsama alanlarını incelemek için alanda KB'ler dışındaki düğümler silinmiş ve her KB etrafında 20m çapında daireler çizilmiştir. Daha sonra, dairelerin içi siyah piksellerle doldurularak toplam siyah piksellerin sayısı hesaplanmış ve toplam piksele oranlanarak yüzdelik dilimi bulunmuştur. Alanın gerçekte, $100m \times 100m = 10.000m^2$ olduğu varsayıldığından buna oranlanmış ve m^2 cinsinden toplam kapsama alanı hesaplanmıştır. Hesaplama, Denklem 5.1'de gösterilen şekilde yapılmıştır.

$$BirleşimAlan = \frac{ToplamSiyahPiksel}{ToplamPiksel} * 10000m^2 \quad (5.1)$$

Buna göre, Şekil 5.1'e göre kümelenen algoritmaların kapsama alanları Şekil 5.5'de gösterilmiştir. Şekilde 'a' ile belirtilen KB düğümlerin etrafında oluşturulan dairesel alanları, 'b' ile belirtilenler ise toplam kapsama alanlarını temsil etmektedir.



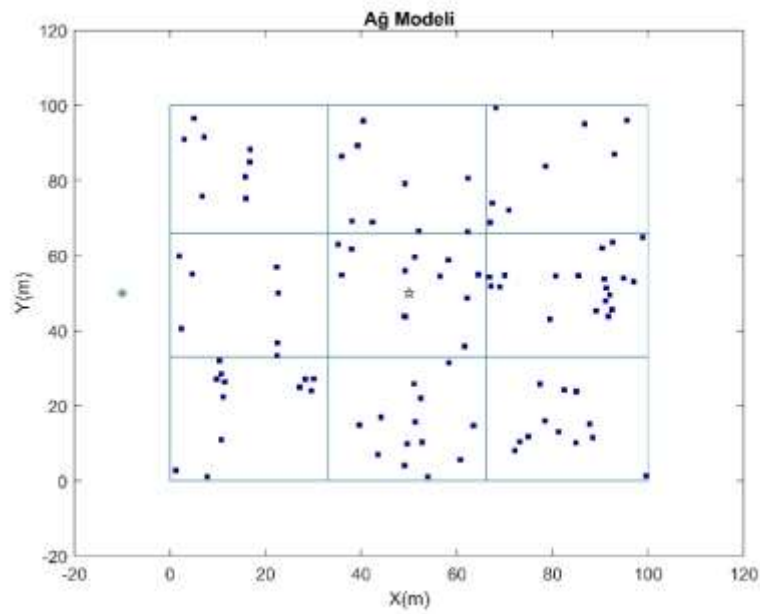
a: KB düğümlerin dairesel alanları

b: KB düğümlerin kapsama alanları

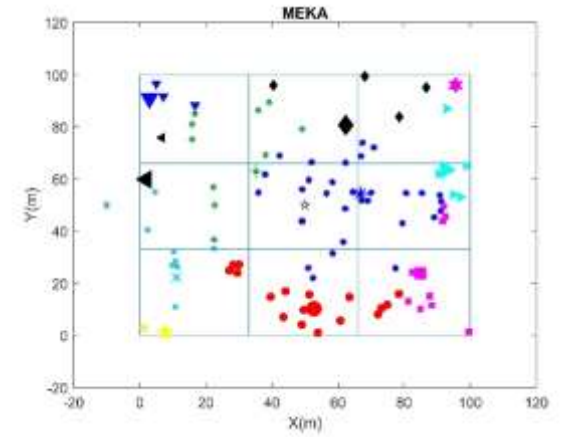
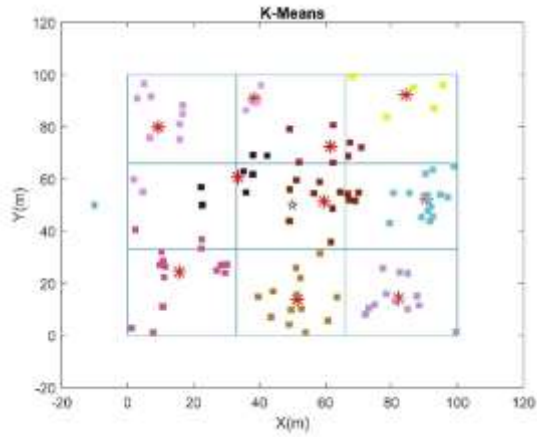
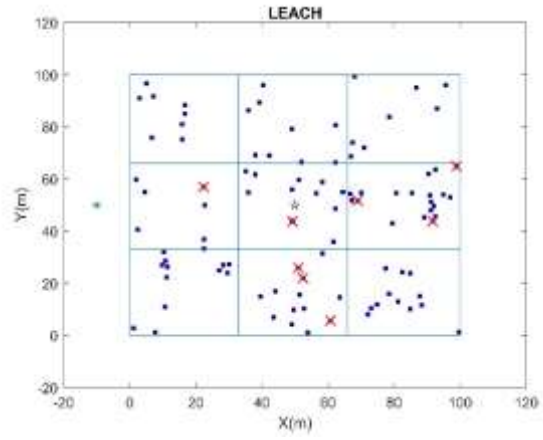
Şekil 5.5. Kapsama alanlarının gösterimi

Şekil 5.5'in b sütunu incelendiğinde LEACH' in KB düğümleri birbirine yakın olduklarından alana homojen dağılamamış ve kapsama alanlarında örtüşmeler olmuştur. k-means ve MEKA algoritmalarında ise KB düğümlerin homojen dağılımları, örtüşmeleri engellemiş ve daha geniş bir alanın kapsanmasını sağlamıştır. Ayrıca aynı ağ modelinde algoritmaların aynı kararlılıkla çalışması beklenmektedir. Örneğin, aynı şartlarda ve aynı ağ modelinde algoritmalar çalıştırıldığında aynı düğümlerin KB olması beklenir. Ancak

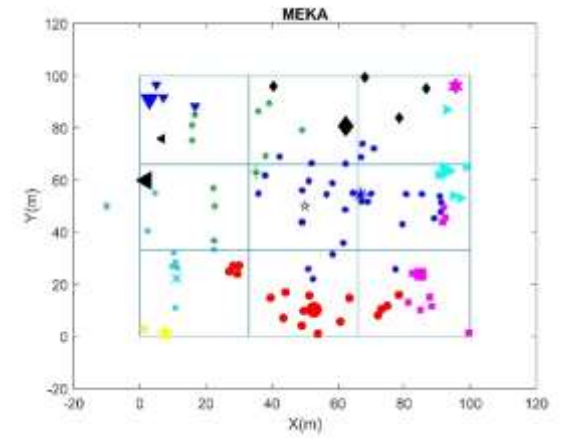
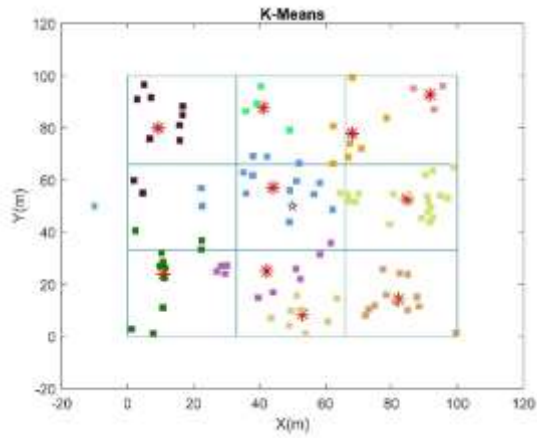
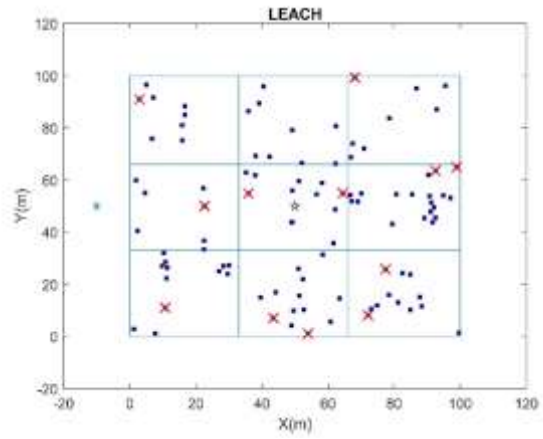
bu kararlılık rastgele seçim yaparak KB seçen algoritmalar için söz konusu değildir. Bunun için, Şekil 5.6'da verilen ağ modeli üzerinde algoritmalar ikişer defa çalıştırılmışlar ve her seferinde aynı düğümlerin KB seçilmesi ve ağın aynı şekilde kümelenmesi beklenmiştir. Bu sayı daha da arttırılabilir ama her seferinde LEACH farklı bir sonuç verecek, k-means yaklaşık bir sonuç üretecek ama MEKA aynı şekilde ağı kümeleyecektir. O nedenle 2 deneme, durumu açıklamak için yeterli görülmüştür. Aynı ağ modelinde, algoritmaların kümelenme davranışları Şekil 5.7'de, KB seçim davranışları ise, daha anlaşılır olması için Şekil 5.8'de verilmiştir.



Şekil 5.6. Örnek ağ modeli

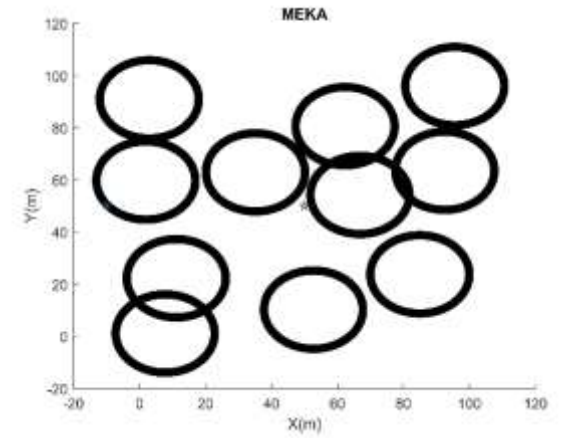
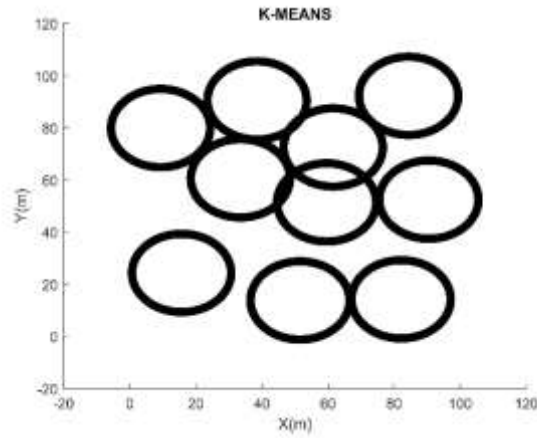
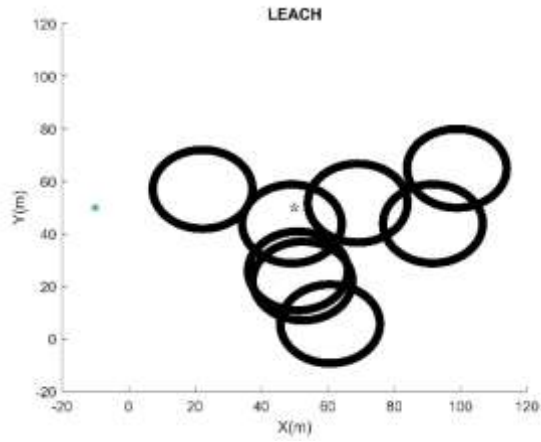


1.deneme

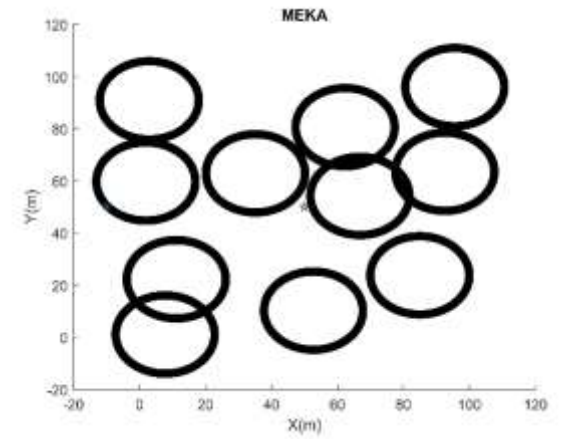
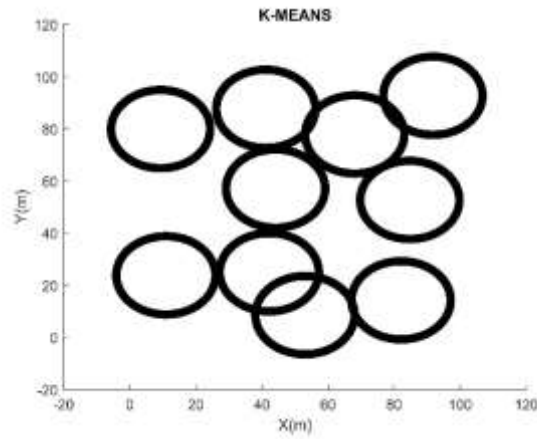
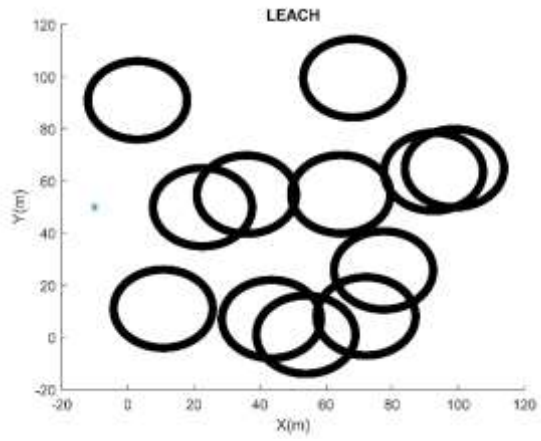


2.deneme

Şekil 5.7. Aynı ağ modelinde algoritmaların kümeleme davranışları



1.deneme



2.deneme

Şekil 5.8. Aynı ağ modelinde algoritmaların KB seçim davranışları

Şekil 5.7 ve Şekil 5.8 incelendiğinde ise, MEKA ve k-means algoritmaları aynı ağ modelinde birden çok kez çalıştırıldığında aynı sonuçları üretmekte fakat LEACH algoritması her çalıştırmada farklı KB seçimi yapmıştır. Dolayısıyla LEACH' in davranışı kestirilememekte, k-means ve MEKA ise kararlı bir davranış göstermektedir. O nedenle, algoritmaların aynı ağda birden çok kez çalıştırılması sonucu değiştirmeyecek, farklı ağ modellerinde denenmesi daha doğru sonuçlar üretecektir.

Buna göre, Şekil 5.5'de gösterilen ve 100 düğümden oluşan ağ modeli de dahil 200 ve 300 düğümden oluşan farklı ağ modellerinde algoritmaların kapsama alanları m^2 cinsinden değerlendirilmiş ve elde edilen bulgular Çizelge 5.2'de verilmiştir.

Çizelge 5.2. Hesaplanan alan değerleri

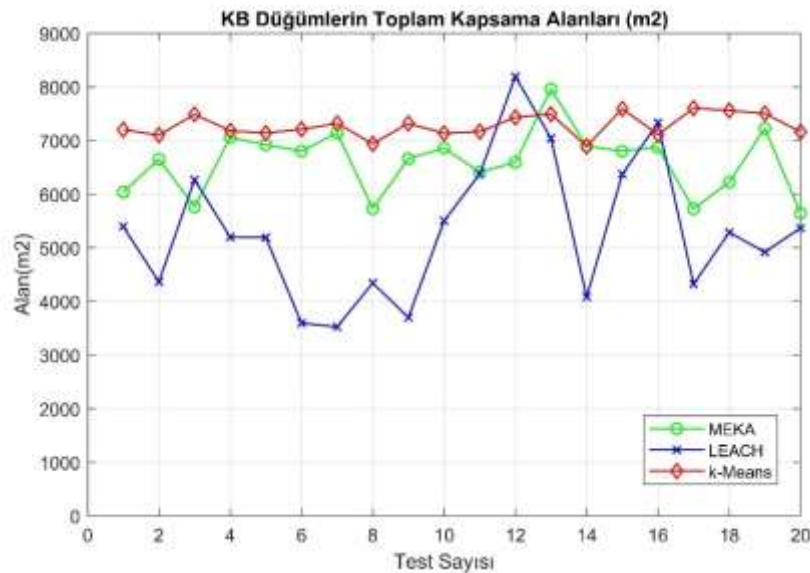
	Algoritma	Alan (m^2)	KB sayısı
100 Düğüm	LEACH	5196,51	10
	MEKA	7060,69	10
	k-means	7183,007	10
200 Düğüm	LEACH	7812,41	21
	MEKA	9485,69	20
	k-means	9501,40	20
300 Düğüm	LEACH	7610,14	25
	MEKA	9991,34	25
	k-means	10000	25

Çizelge 5.2'den de anlaşılacağı üzere, alan sabit tutulup düğüm sayısı arttırıldığında aynı oranda KB sayısı da artmış ve 300 düğümden k-means ve MEKA algoritmaları alanın tamamını kapsarken LEACH algoritması mevcut dezavantajlarından dolayı başarısız bir performans göstermiştir. Benzer bir uygulama farklı alan büyüklüklerinde de yapılmış ve elde edilen sonuçlar Çizelge 5.3'de sunulmuştur.

Çizelge 5.3. Farklı alan ölçülerinde algoritmaların kapsama alanı performansları

	Algoritma	Düğüm Sayısı	Alan (m ²)	KB sayısı
100m*100 m	LEACH	100	7240	11
	MEKA	100	8376	10
	k-means	100	8358	10
200m*200m	LEACH	200	13985	16
	MEKA	200	17725	19
	k-means	200	24040	20
300m*300m	LEACH	300	21372	33
	MEKA	300	20350	26
	k-means	300	23394	30
400m*400m	LEACH	400	69121	37
	MEKA	400	76546	36
	k-means	400	84729	40
500m*500m	LEACH	500	108000	46
	MEKA	500	119600	46
	k-means	500	132390	50

Çizelge 5.2’de elde edilen bulguların doğrulanması için algoritmalar Çizelge 5.1’de verilen parametrelere bağlı kalarak 20 farklı ağ modelinde test edilmiştir. Algoritmalar, 20’den fazla modelde de denenmiştir. Ancak sonuçlarda önemli bir farklılık gözlemlenmediğinden burada 20 farklı ağ modelinden alınan veriler paylaşılmıştır. Buna göre elde edilen alan değerlerinin grafiksel gösterimi Şekil 5.9’da sunulmuştur.

**Şekil 5.9.** 20 farklı ağ modeli için, seçilen KB düğümlerin toplam birleşim alanları

Farklı ağ modellerinden elde edilen alan değerlerinin anlamlandırılması için ortalamaları alınmış ve sonuçları Çizelge 5.4’de sunulmuştur.

Çizelge 5.4. Algoritmaların kapsama alanları yüzdesi

Algoritma	Ortalama Alan (m ²)	Kapsama Alanı (%)	KB sayısı
LEACH	5287,718	52	9
MEKA	7100,785	71	11
k-means	7828,024	78	10

Buna göre, LEACH'in ortalama kapsama alanı %52, k-means'in %78, MEKA'nın ise %71 olarak tespit edilmiştir. Elde edilen bulgulara göre, öz-örgütlü kümeleme yapısına sahip MEKA ve LEACH algoritmaları kıyaslandığında MEKA'nın LEACH'e göre kararlı ve kapsama alanının daha geniş olduğu görülmektedir. k-means, doğası gereği daha başarılı kapsama alanlarına ulaşmıştır ancak gerekçeleri önceki bölümlerde anlatılmıştır. MEKA'nın ise, öz-örgütlü yapıda olmasına rağmen kümelenme performansı bakımından Mİ yönlendirmeli algoritmalar gibi davrandığı yaklaşık bir başarıyı sağladığı açıkça görülmektedir.

5.3. Paket Sayıları ve Enerji Tüketimleri Açısından Değerlendirme

Ağın ömrü ve algoritmaların performanslarının değerlendirilebilmesi için düğümlerin enerji tüketimlerinin kontrol edilmesi gerekmektedir. Enerji tüketim değerlerinin hesaplanmasında, karşılaştırmaların daha doğru yapılabilmesi ve basit yapısından dolayı LEACH algoritması ile aynı radyo modeli kullanılmıştır. Kullanılan model önceki bölümlerde detaylı bir şekilde anlatılmıştır. Bu radyo modeline göre tüketim değerleri hesaplanmaktadır. Oluşan ağ modelinde 1. tur kümeleme işlemi tamamlandıktan sonra düğümlerin KB'ye belli aralıklarda ve belli büyüklüklerde veri paketleri gönderdikleri varsayılmıştır. KB ise düğümlerden toplandığı verileri Mİ'ye iletir. Bu döngü KB düğümün enerjisini tüketmesiyle sonlanır ve ağın yeniden kümelenmesi ve yeni KB seçilmesiyle devam etmektedir.

Deneysel çalışmada, bu tur sayısı ya da iterasyon sayısı sınırlandırılmış ve sonunda ağda kalan toplam enerji seviyesine ve gönderilen paket sayılarına göre değerlendirme yapılmıştır. Analiz için Çizelge 5.5'de sunulan değerler kullanılmıştır.

Çizelge 5.5. Enerji tüketimlerinin ve paket sayılarının hesaplanmasında kullanılan değerler

Parametre	Değer
Düğüm Sayısı	100
Yol kaybı katsayısı η	2
Paket boyutu	4000 bit
Her iterasyonda düğümler tarafından üretilen paket sayısı	72 / iterasyon
Her iterasyonda 1 düğümden gönderilen toplam veri boyutu	35kB
Toplam iterasyon (tur) sayısı	20
Düğüm başlangıç enerjisi (E_0)	0,5 Joule
Ağın toplam enerjisi (E_{total})	50 Joule
Alma/İletmede harcanan enerji (E_{elec})	0,05 nJ/bit
İletim yükselticisinin bit başına enerji tüketimi (E_{amp})	0,13 nJ/bit

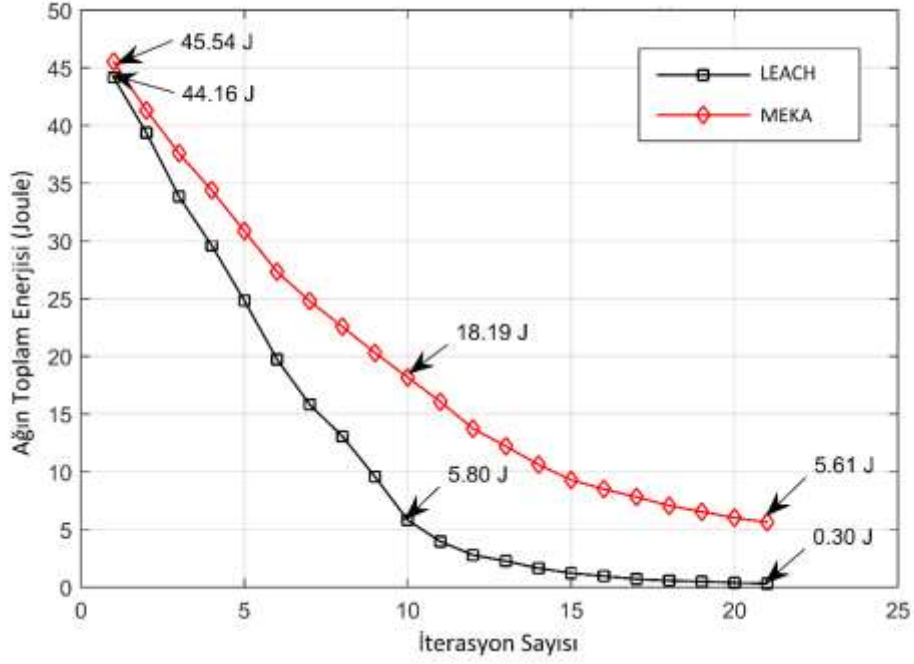
Çizelge 5.5’de verilen düğüm sayısı, paket boyutu ve düğüm başlangıç enerjisi değerleri literatürde kullanılan değerlerdir ve bu değerlere göre elde edilen sonuçlar kullanılarak MEKA’nın performansı ortaya konulmuştur. Ancak tartışılan algoritmalarda, bu parametrelerde farklı değerler kullanılmışsa o değerlere göre simülasyonda elde edilen sonuçlar kullanılarak tartışma yapılmıştır. Yol kaybı katsayısı ise 2 olarak kullanılmıştır. Sebebi ise bölüm 4.1.1’de detaylı olarak anlatılmıştır. Diğer değerler ise sabittir ve literatürde kabul edilmiş değerler olduklarından burada da aynı şekilde kullanılmıştır.

Buna göre, 20 iterasyon sonunda Şekil 5.1’de verilen ağ modeline göre kümelenen LEACH ve MEKA algoritmalarının enerji tüketim davranışları ve ağda kalan toplam enerji değerleri Çizelge 5.6’de verilmiştir. Burada, daha öncede tüm detayları ile belirtildiği gibi k-means algoritması öz-örgütlü bir yapıda olmadığından enerji tüketimi yönünden değerlendirmeye alınmamıştır.

Çizelge 5.6. 20 İterasyon sonunda ağın enerji tüketim verileri

Algoritma	10. İterasyon		20. İterasyon	
	Ağda Kalan Enerji (j)	Oran	Ağda Kalan Enerji (j)	Oran
MEKA	18,19	37	5,61	11,2
LEACH	5,8	11	0,3	1

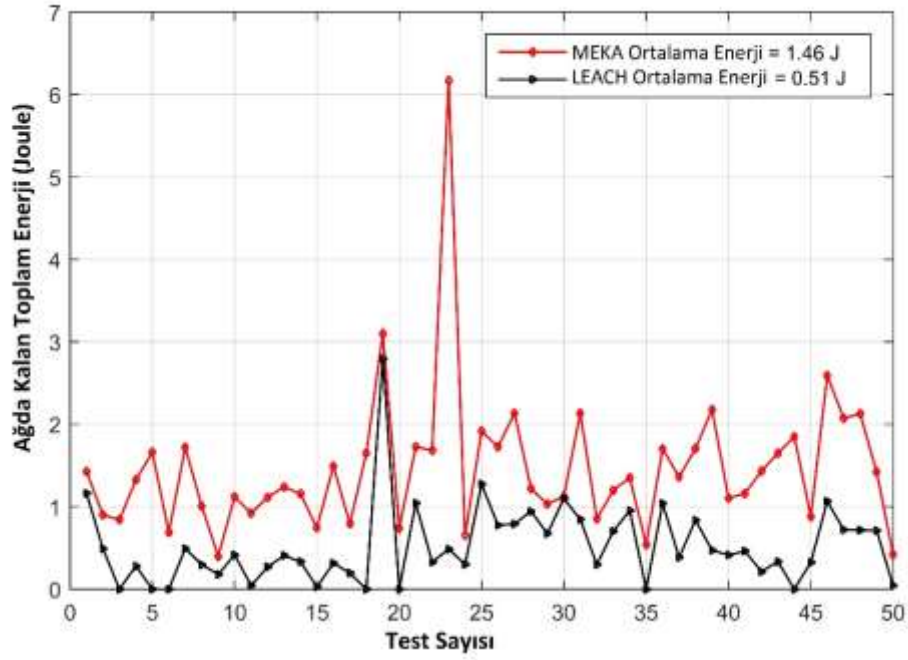
10. iterasyon sonunda kalan toplam enerjiye bakıldığında, LEACH’ de toplam enerjinin %11’i, MEKA’ da ise toplam enerjinin %37’sinin ağda kaldığı görülmüştür. 20 iterasyon sonunda ise LEACH’ de ağın enerjisi tükenmiş ve toplam enerjinin ancak %1’ini saklayabilmiştir. MEKA ise toplam enerjisinin %11,2’sini saklayabilmiştir. Ağın iterasyonlara göre toplam enerji miktarındaki değişim grafiği Şekil 5.10’da verilmiştir.



Şekil 5.10. 20 iterasyon sonunda algoritmaların enerji tüketim davranışları ve ağda kalan toplam enerji (joule)

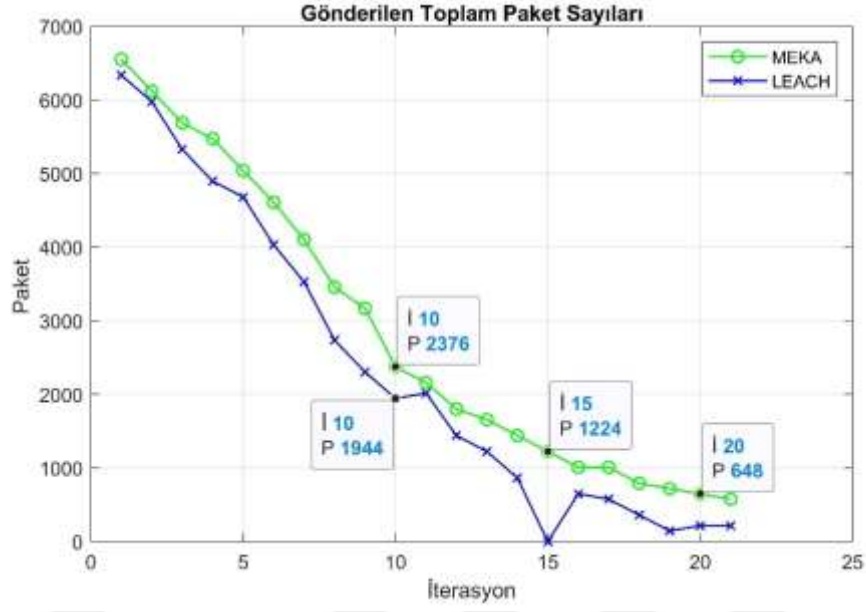
Şekil 5.10’da elde edilen verilerin doğruluğunu göstermek için algoritmalar 50 farklı ağ modelinde aynı şartlarda test edilmiştir. Burada test sayısı olarak 50 seçilmesinin nedeni ise şu şekilde açıklanabilir; her ağ modelinde düğümler alana rastgele serpilerek dağıtılırlar ve her dağıtımda özellikle LEACH algoritmasının KB seçimi ve kümeleme davranışı farklılık göstermektedir. Bir modelde çok iyi seçimler yapabiliyorken bir diğerinde daha kötü KB seçimlerine bağlı olarak daha kötü tüketim değerleri çıkarabilmektedir. O nedenle karşılaştırmanın objektif olması açısından test sayısı 50 olarak belirlenmiştir.

Yapılan testlerin sonunda ağda kalan toplam enerji miktarlarının grafiği Şekil 5.11’de verilmiştir. Elde edilen verilerin anlamlandırılması için her test sonunda ağda kalan enerji miktarları toplanarak aritmetik ortalamaları alınmıştır. Buna göre, LEACH’de ağda kalan ortalama toplam enerji 0,51J iken MEKA’da 1,46J olduğu tespit edilmiştir.



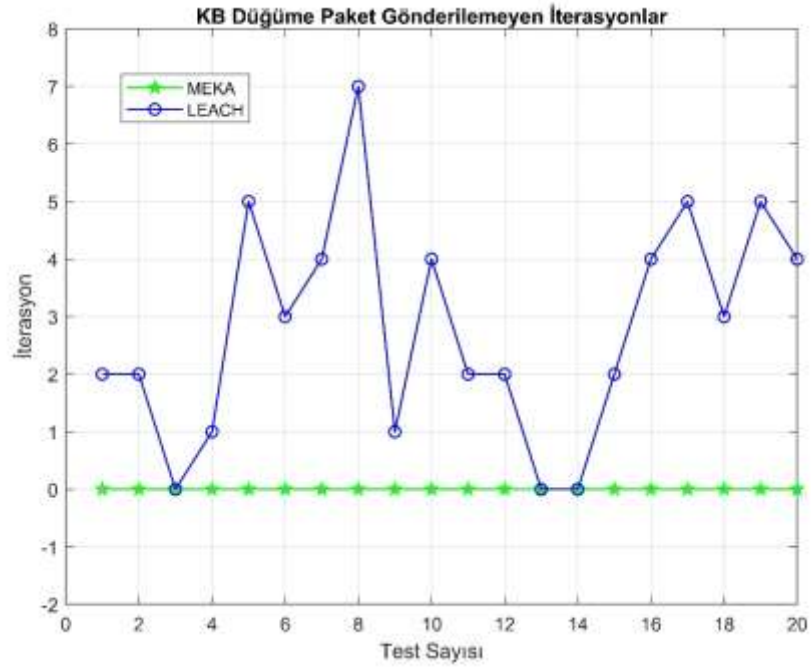
Şekil 5.11. 50 farklı ağ modeli için ağda kalan toplam enerji miktarları

Aynı parametrelerle ve aynı ağ modelinde, gönderilen paket sayısı yönünden bakıldığında ise Şekil 5.12’de verilen sonuçlar elde edilmiştir. Grafikten de anlaşılacağı üzere MEKA algoritması veri gönderiminde kararlılık gösterirken LEACH 15. iterasyonda hiç veri gönderimi yapamamıştır. Gönderim yapılamaması, o iterasyonda algoritmanın KB seçememesi ya da KB seçilen düğümlerin çok düşük enerjili düğümlerden seçilmiş olması ile ilgilidir. Gönderilen paket sayılarının ortalamasına bakıldığında, LEACH 2355,43 paket gönderebilirken MEKA 2838,86 paket göndermeyi başarmıştır.



Şekil 5.12. Gönderilen toplam paket sayıları

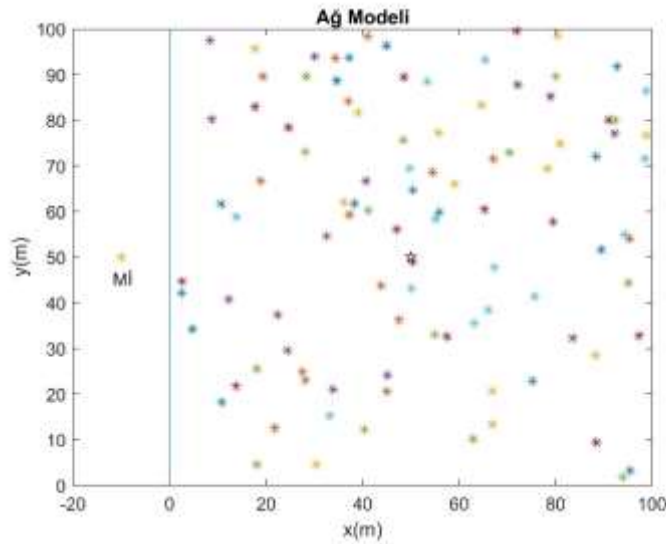
Algoritmaların veri gönderimindeki sürekliliğinin ortaya konulması için algoritmalar aynı şartlarda 20 farklı ağ modeli üzerinde denenmiş ve veri gönderilemeyen iterasyonlar tespit edilmiştir. Buna göre elde edilen veriler Şekil 5.13’de sunulmuştur.



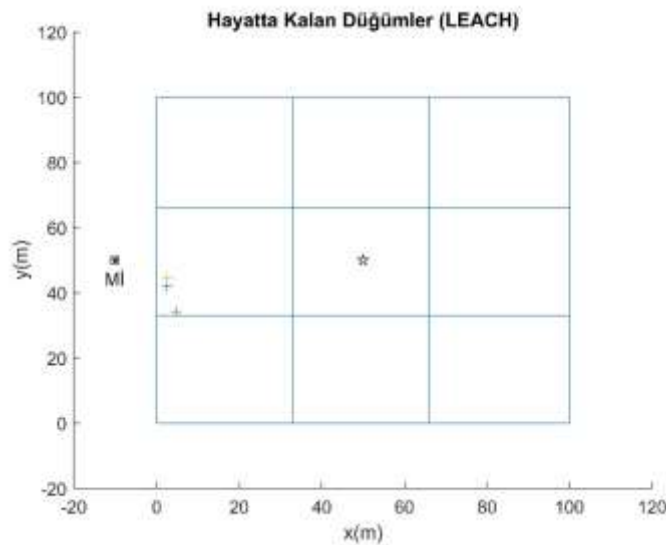
Şekil 5.13. 20 farklı ağ modelinde paket gönderilemeyen iterasyonlar

5.4. Ölen ve Hayatta Kalan Düğümler Açısından Değerlendirme

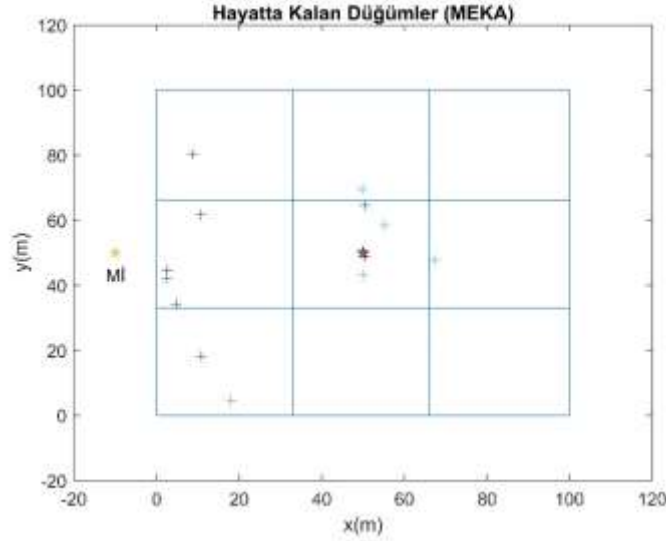
Başlangıç enerjileri 0,5J olarak kabule edilen düğümler, belirli bir süre iletim yaptıktan sonra enerjileri sönümlenmektedir. Bu düğümler ölü düğüm olarak kabul edilmektedir. Algoritmalarda, belirli bir iterasyon sonunda ölen düğümlerin sayısı, ağın ömrünü dolayısıyla algoritmanın performansını gösteren önemli bir göstergedir. O nedenle, algoritmalar Şekil 5.14’da verilen ağ modeline göre çalıştırılmışlar ve Şekil 5.15 ve Şekil 5.16’de sırasıyla LEACH ve MEKA algoritmalarında 20 iterasyon sonunda hayatta kalan düğümler gösterilmiştir.



Şekil 5.14. Ölü düğümlerin tespiti için kullanılan ağ modeli.

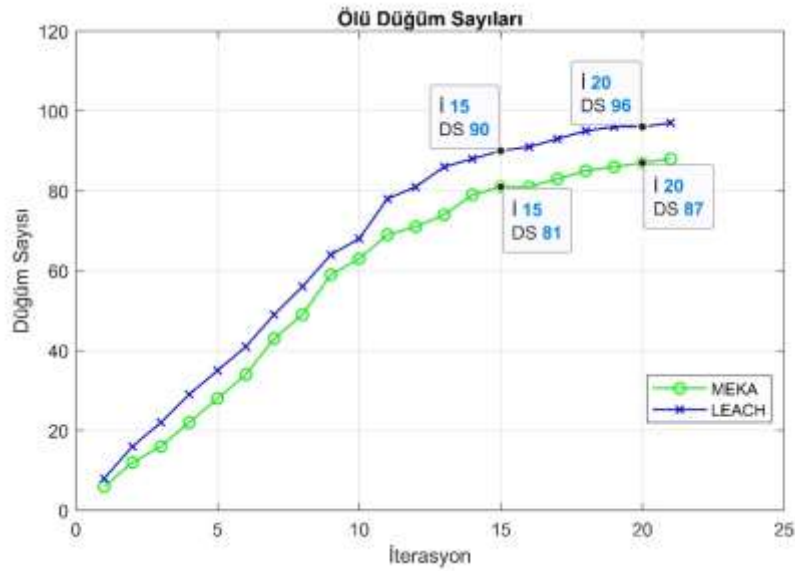


Şekil 5.15. LEACH algoritmasında hayatta kalan düğümler.



Şekil 5.16. MEKA algoritmasında hayatta kalan düğümler

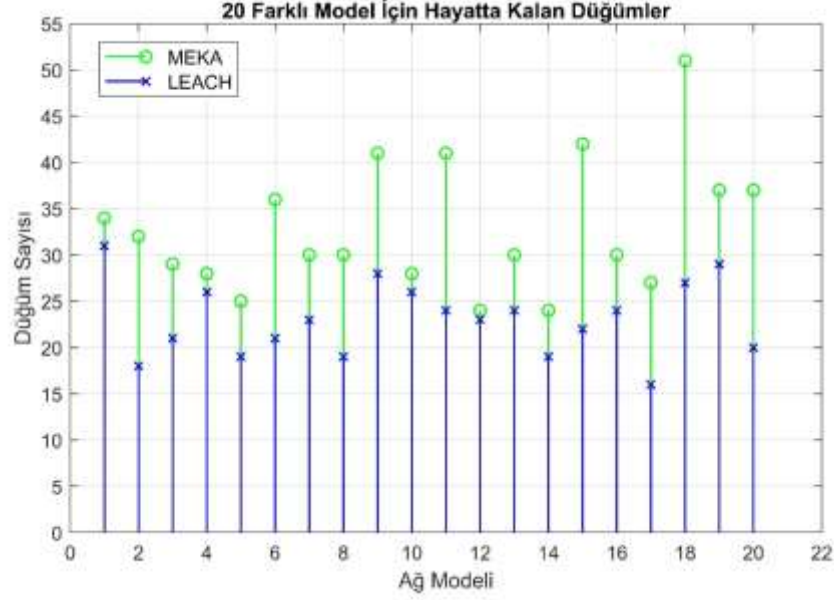
Buna göre, iterasyona bağlı olarak ölen düğüm sayılarının değişimini gösteren grafik Şekil 5.17’de sunulmuştur. 20.iterasyonun sonunda, LEACH algoritmasında ağda toplam 4 düğüm hayatta kalırken MEKA’da 13 düğüm hayatta kalmıştır. Aradaki bu farklılık düğüm başlangıç enerjileri ve iterasyon sayısı arttırıldığında daha da açılmaktadır.



Şekil 5.17. İterasyonlara göre ölü düğüm sayıları

Örneğin simülasyon parametrelerinden, düğüm başlangıç enerjisi 1,5J ve iterasyon sayısı 50 olarak değiştirildiğinde, algoritmalar 20 farklı ağ modelinde çalıştırılmış ve hayatta kalan düğüm sayılarına bakılmış ve Şekil 5.18’de verilen grafik

elde edilmiştir. Buna göre, LEACH' de hayatta kalan düğümlerin ortalaması 23 iken MEKA' da 32,8 olarak tespit edilmiştir.



Şekil 5.18. 20 farklı ağ modeline göre hayatta kalan düğümler ($E_0=1,5j$, iterasyon=50)

5.5. Algoritmaların Zaman Karmaşıklığı Bakımından Değerlendirilmesi

Algoritmaların, kümelenme hızları ve gerçek bir mikro kontrolcü üzerindeki performansları karmaşıklıkları ile ilgilidir. Algoritmaların karmaşıklıkları hesaplanırken her adım hesaba katılır ve atama vb. sabit zamanlı işlemler 1 (T) zaman birimi olarak kabul edilir. Örneğin; aşağıda verilen kod parçasının karmaşıklığı analiz edildiğinde $T_1+T_2+T_3+n*(T_4+T_5+T_6)$ zamanda çalışacaktır. Burada T_1, T_2 , şeklinde gösterilen zaman ifadeleri 1 olarak ya da aynı zaman dilimi şeklinde de kullanılabilir.

```

i = 1; →  $T_1$  zaman
n=10; →  $T_2$  zaman
toplam = 0; →  $T_3$  zaman
while (i <= n) →  $T_4$  zamanda n+1 kez yapılacak;
{
    i = i + 1; →  $T_5$  zamanda n kez yapılacak
    toplam = toplam + i; →  $T_6$  zamanda n kez yapılacak
}

```

Bu algoritmada, sabit zamanlı işlemler göz ardı edilir ve algoritmanın karmaşıklığı $O(n)$ olarak ifade edilir. Aşağıdaki algoritmada gösterildiği gibi iç içe döngü olması

durumunda zaman karmaşıklığı $T_1+T_2+T_3+n*(T_4+T_5+n*(T_6+T_7+T_8+T_9))$ şeklinde hesaplanır. Bu durumda, karmaşıklık üstel bir fonksiyona dönüşür $O(n^2)$ ve bu aynı zamanda algoritmanın büyüme fonksiyonu olarak da kabul edilebilir.

```

i = 1; → T1 zaman
n=10; → T2 zaman
toplam = 0; → T3 zaman
while (i < n) → T4 zamanda n kez yapılacak
{
  j=0; → T5 zaman
  While(j<n) → T6 zamanda n kez yapılacak
  {
    i = i + 1; → T7 zamanda n kez yapılacak
    j=j+1; → T8 zamanda n kez yapılacak
    toplam = toplam + i; → T9 zamanda n kez yapılacak
  }
}

```

Buna göre; Şekil 4.11’de verilen MEKA algoritması, aşağıda gösterildiği gibi kaba kod haline dönüştürülüp zaman karmaşıklığı hesaplanmıştır.

Program MEKA

```

{
  i = 0; → T zaman
  n = Düşüm Sayısı → T zaman
  Donanımsal Kontrol  $Nid_i$  → T zaman
  Tüm düşümlerden  $RSSI$  değerlerini oku → T zaman
  Ort hesapla → T zaman
  if(ort!=0) → T zaman
  {
    for(j=0; j<n; j++) → T zamanda n kez yapılacak
    {
       $RSSI$ 'dan  $d_j$  hesapla; → T zamanda n kez yapılacak
      Ortalama  $d_{avr_i}$  bul; → T zamanda n kez yapılacak
    }
    Ağa  $d_{avr_i}$  yayınla → T zaman
    Diğer düşümlerden  $d_{avr}$  oku → T zaman
    for(k=0; k<n-1; k++) → T zamanda n kez yapılacak
    {
      if( $d_{avr_i} < d_{avr_k}$ ) → T zamanda n kez yapılacak
      {
        KB=true; T zamanda
      }
    }
    if(KB==false) → T zaman
    {
      for(m=0; m<n; m++) → T zamanda n kez yapılacak
      {
        if(  $d_{avr_m} < M[f(x)]_i$  ) T zamanda n kez yapılacak
        {
          Node_m KB kümesine dahil ol T zaman
        }
      }
    }
  }
}

```

Zaman karmaşıklığı $6T+(n*3T) + 2T+(n*2T) + 3T = 11T+n*5T$ olarak bulunan **MEKA**'nın karmaşıklık fonksiyonu $O(n)$ olarak ifade edilir.

Çalışmadaki diğer algoritmalar buna göre analiz edildiğinde, **LEACH**' in küme başı seçiminde her bir düğüm için ölçülen eşik, düğümün olasılığı nedeniyle sadece rastgele bir sayıdır. Bu eşik değeri ağdaki tüm düğümler için uygulanır. Dolayısıyla **LEACH** için zaman karmaşıklığı $O(n)$ olarak ifade edilmektedir (Su ve Zhao, 2018). k-means' in zaman karmaşıklığı ise $O(T, K, n)$ fonksiyonundaki T , K ve n değişkenlerine bağlıdır. Burada n girdi sayısını, K istenen küme sayısını ve T kümeleme işlemini tamamlamak için gereken yineleme sayısını ifade eder ve **k-means** algoritmasının zaman karmaşıklığı $O(n^2)$ olarak ifade edilmektedir (Panag ve Dhillon, 2018).

Büyüme fonksiyonundaki üstel veri, algoritmaların verimlilikleriyle ters orantılı olarak değerlendirilmektedir. Üç farklı algoritmanın karmaşıklık fonksiyonları kıyaslandığında **MEKA** ile **LEACH**' in benzer, k-means algoritmasının karmaşıklık fonksiyonunun ise yüksek yani diğerlerine göre daha verimsiz olduğu görülmektedir.

5.6. Tartışma

Geliştirilen **MEKA** algoritması, kapsama alanı, gönderilen paket sayıları, ağda kalan toplam enerji ve hayatta kalan ve ölen düğüm sayıları yönünden **LEACH** ve k-means algoritmaları ile kıyaslanmıştır. **LEACH**, literatürdeki en önemli öz-örgütlü algoritma, k-means ise verimli bir kümeleme algoritması olduğu için seçilmişlerdir. Elde edilen bulgular önceki bölümlerde tüm detaylarıyla sunulmuştur.

Düzensiz formdaki **KAA** modellerinde ağın kümeleme başarımı ve enerji verimliliğini inceleyen birçok makale ve tez çalışması bulunmaktadır. Bu çalışmalar genel olarak ele alındığında aşağıdaki maddelerden söz edilebilir. Bunlar;

1. Kümeleme yönteminin ağın enerji verimliliğine olan olumlu etkisi açıkça anlaşılmaktadır.
2. Literatürdeki algoritmalar Mİ yönlendirmeli ve öz-örgütlü olmak üzere iki üst başlık altında toplanmaktadır.
3. Öz-örgütlü olarak değerlendirilen ve literatüre çok güçlü katkılar yapmış iki algoritma **LEACH** ve **HEED** algoritmalarıdır.
4. Çalışmaların tamamına yakınında, ağın enerji verimliliği için önerilen algoritmaların hemen hemen hepsinde **LEACH** ve **HEED** algoritmaları ile

kıyaslama yapılmış ve kendi performanslarını bu algoritmalara göre kanıtlamışlardır.

5. Genel kümeleme algoritmaları içinde ise k-means ve türevleri kapsama alanı yönünden başarılıdır ancak enerji tüketimi yönünden aynı başarıyı gösteremedikleri açıktır.
6. Düğümlerin alandaki dağılımları düzenli ve düzensiz olmak üzere 2 farklı formda ele alınmıştır.
7. Coğrafi olarak algılama yapılması zor olan bölgelerdeki KAA uygulamalarında düğümlerin alana düzenli bir formda yerleştirilmesi ve kontrolü pek mümkün olmadığından ağır kümelenmesi ve veri iletiminde düğümlerin kendi içinde organize olması beklenmektedir. O nedenle, düzensiz dağıtık yapılar için öz-örgütlü kümelemenin önemi vurgulanmıştır.
8. Çalışmaların tamamında enerji tüketimlerinin hesaplanması için bu tez çalışmasında kullanılan radyo modelinin aynısı kullanılmıştır.
9. Çalışmaların genelinde aynı değerlerle kullanılan parametreler şunlardır; radyo modelinde kullanılan $E_{elec} = 0,05$ nJ/bit , $E_{amp} = 0,1$ nJ/bit ve düğümlerin dağıldığı alan büyüklükleri (100m*100m).
10. Çalışmalarda farklı değerlerle kullanılan parametreler ise düğüm başlangıç enerjisi (E_0), ilemede gönderilen paket boyutları ve düğüm sayılarıdır (N). E_0 için tercih edilen değerler sırasıyla 0.5J, 1.25J, 1.5J ve 2J'dür. Düğüm sayıları genelde 100 kabul edilmiştir ancak bazı çalışmalarda 700 düğüme kadar farklı sayılarda yapılan simülasyonlar da görülmektedir. Paket büyüklüğü olarak kullanılan değerler ise sırasıyla 1000 bit, 2000 bit ve 4000 bit' dir.

Tartışma ilkeleri

MEKA, yapısı itibariyle LEACH ve HEED ile aynı kategoride değerlendirilmektedir. Bu iki algoritmanın ortak problemi KB seçimindeki rastgele seçim metodudur. MEKA, rakiplerine göre ayakları yere basan, mantıksal bir temele dayanan, basit, uygulanabilir ve enerji verimli bir algoritmadır. MEKA, aynı kategoride olmasalar bile enerji verimini ön plana alan literatürdeki diğer algoritmalarla sonuçları yönünden tartışılmıştır. Buna göre tartışılan çalışmalarda aşağıdaki ilkeler esas alınmıştır;

1. Simülasyon parametreleri ve değerleri kontrol edilmiş ve aynı değerler kullanılarak MEKA algoritması simüle edilmiştir. Elde edilen sonuçlara göre tartışma yapılmıştır.
2. Çalışmanın sonuçları sayısal olarak verilmemişse, sunulan grafiklerden yaklaşık değer okuması yapılmış ve o sonuçlarla tartışılmıştır.
3. Literatürdeki diğer çalışmalara nazaran yapılan iyileştirme yüzdesel olarak verilmişse, tartışma aynı şekilde yüzdesel olarak yapılmıştır.
4. Önerilen esas algoritma referans alınmış, onun kendisini karşılaştırdığı ve daha iyi olduğunu iddia ettiği diğer algoritmalarla tartışılmamıştır. MEKA'nın önerilen algoritmadan daha iyi sonuçlar verdiği durumlarda, diğer algoritmalarından da daha iyi olduğu kabul edilmiştir.
5. Farklı alan ölçülerinde ölçümler alınmışsa, tartışmada 100m*100m alandan alınan sonuçlar kullanılmıştır.
6. Son düğümün öldüğü iterasyon sayıları, algoritmaların ağ ömrü başarımlarının hesaplanması için kullanılmıştır. Örnek: 2013 yılında yayınlanan Q-LEACH algoritmasının sonuçlarına göre, ağdaki son düğüm 2900.iterasyonda ölürken aynı verilere göre LEACH algoritmasında son düğümün 1700.iterasyonda öldüğü bildirilmiştir. Aynı verilerle MEKA algoritması çalıştırıldığında ise son düğümün 5400.iterasyonda öldüğü tespit edilmiştir. Buna göre, 6000 iterasyonu %100 kabul edersek,
 - LEACH = $1700 * 100 / 6000 = \%28,33$
 - Q-LEACH = $2900 * 100 / 6000 = \%48,33$
 - MEKA = $5400 * 100 / 6000 = \%90$ çıkmaktadır. Buna göre; Q-LEACH, LEACH'e göre %20 iyileşme gösterirken MEKA ise %61,67 iyileşme göstermiştir.
7. Tartışılan algoritma, kendi içinde LEACH algoritması ile tartışmamışsa ağ ömrü iyileştirme oranı verilmemiş sadece son düğümün öldüğü iterasyon sayısı verilerek tartışılmıştır.

Tartışma bulguları

Yukarıda sayılan ilkelere göre aşağıdaki çalışmalarla tartışılmış ve sonuçları aşağıda sunulmuştur.

1. HEED algoritmasının sonuçları incelendiğinde ağdaki son düğümün olduğu iterasyon sayısı parametre olarak kabul edilmiştir. HEED algoritmasında, 300 ve 700 düğümden oluşan ağlarda yapılan denemelerde ağdaki son düğümün LEACH'de 780. iterasyonda, HEED algoritmasında ise 920. iterasyonda öldüğü bildirilmiştir (Younis ve Fahmy, 2004). Aynı değerlerle MEKA algoritması çalıştırıldığında, son düğümün 1177. iterasyonda öldüğü tespit edilmiştir.
2. Tümer'in 2011 yılında yaptığı doktora tezinde, kapalı mekanlarda metan gazının algılanması için kullanılan çok sayıda düğüm ve bunlardan alınan verilerin Mİ'ye iletilmesi amacıyla enerji verimli kümeleme algoritmaları önermiştir. Önerilen algoritmalarındaki temel prensip, belirli bir eşğin üstünde olan verilerin iletilmesidir. Önerilen algoritmalarından en verimli olanı RE-EERP'nin sonuçlarında ilk düğümün öldüğü iterasyon sayısı ve düğümlerin yarısının öldükleri iterasyon sayılarına göre değerlendirme yapılmış ve RE-EERP algoritmasının LEACH'e göre %16 iyileşme sağladığı tespit edilmiştir (Tümer, 2011). Simülasyonda bu algoritma ile tartışmak için iletme geçmeden önce 0-50 arasında rastgele bir sayı üretilmiş ve örneğin üretilen sayının 25'in üzerinde olması halinde iletme geçilmiştir. Aynı parametre verileri ve aynı çalışma prensibi ile MEKA çalıştırıldığında RE-EERP'ye göre 4 kat daha iyi sonuçlar elde edilmiştir.
3. Manzoor ve ark. 2013'de yaptıkları çalışmada Q-LEACH algoritması önerilmiştir. Bu algortmada, maksimum 3000 iterasyondan oluşan simülasyon sonucunda, ağdaki son düğüm LEACH' de 1700. iterasyonda ölürken, Q-LEACH algoritmasında 2900. iterasyonda öldüğü ve LEACH'e göre %40 oranında daha iyi sonuç verdiği bildirilmiştir (Manzoor ve ark.,2013). Aynı verilerle MEKA algoritması çalıştırıldığında son düğümün 5400.iterasyonda öldüğü tespit edilmiştir.
4. Periyasamy ve ark. 2016 yılında yaptıkları çalışmada, Mk-means algoritmasını önermişler ve sonuçlarını k-means, ECRA, SECA-M, LEACH

ve HEED algoritmaları ile tartışmışlardır. Yayınlanan sonuçlara göre ağda kalan toplam enerji bakımından yapılan değerlendirmede, LEACH, HEED ve k-means algoritmalarında 1000. iterasyonda ağın enerjisi sönümlenirken SECA-M ve ECRA algoritmalarında 1200. iterasyonda toplam enerjinin yaklaşık %37'si, Mk-means algoritmasında ise yaklaşık %60'ı saklanabilmiştir. Çalışmada, simülasyonda kullanılan parametreler aynı değerleri ile MEKA algoritmasında çalıştırıldığında MEKA algoritması 1200. iterasyonda ağdaki enerjinin %82,5'ini saklayabilmiştir (Periyasamy ve ark., 2016).

5. Ray ve De 2016 yılında yaptıkları çalışmada ise EECPK algoritmasını önermişler ve Mk-means ile karşılaştırmışlardır. EECPK'nın, ağda kalan toplam enerji bakımından aynı iterasyon sayısı ile Mk-means'den %6 daha iyi sonuç verdiği bildirilmiştir. Buna göre MEKA algoritması değerlendirildiğinde, MEKA'nın EECPK algoritmasından da daha iyi olduğu görülmektedir (Ray ve De, 2016).
6. 2018 yılında Su ve ark. Fuzzy-C tabanlı kümeleme algoritması ile küme başı sayısını optimize eden bir yaklaşım sunmaktadır. Ağ içinde yayılmış olan düğümlerin konum bilgilerinin GPS ile alındığı varsayılmış ve optimizasyon için Mİ kullanılmıştır. Makalede belirtilen parametreler ve değerlerle, önerilen OCM-FC algoritmasının diğer algoritmalarla göre daha iyi sonuçlar ürettiği ve 600 iterasyon sonunda tüm enerjisini tükettiği bildirilmiştir (Su ve Zhao, 2018). Aynı değerlerle MEKA algoritması çalıştırıldığında, son düğümün 2400 iterasyon sonunda öldüğü tespit edilmiştir.
7. 2018'de Panag ve Dhillon, düğümlerin konum ve enerji seviyelerinin göz önüne alındığı yeni bir kümeleme algoritması önermişlerdir. Önerilen DHSCA algoritması, iterasyonlara bağlı olarak hayatta kalan düğümlerin sayıları bakımından LEACH, TCAC ve SEECH algoritmaları ile kıyaslanmıştır. Kıyaslama 3 farklı senaryo ile yapılmıştır. Senaryolarda, radyo modeli ve gönderilen paketlerin büyüklükleri sabit kalmış, düğümlerin dağıldığı alanın büyüklüğü, Mİ'nin alana olan uzaklığı, düğüm sayıları ve düğüm başlangıç değeri değiştirilmiştir. Senaryolarda değişken parametreler Çizelge 5.7'da verilmiştir.

Çizelge 5.7. DHSCA algoritmasının test senaryoları

	Düğüm sayısı (N)	Alan (m)	Mİ Konumu (m)	E_0 (joule)
Senaryo-1	100	100*100	50*175	0.5
Senaryo-2	400	100*100	50*200	0.5
Senaryo-3	1000	200*200	100*350	1

Yukarıdaki tabloda belirtilen senaryolara göre algoritmaların son düğümün öldüğü iterasyon (LND) yönünden kıyaslanması sonucu elde edilen veriler Çizelge 5.8’de sunulmuştur.

Çizelge 5.8. DHSCA algoritmasının LEACH, TCAC ve SEECH ile LND yönünden kıyaslanması

	DHSCA (ite)	TCAC (ite)	SEECH (ite)	LEACH (ite)
Senaryo-1	1099	1006	1099	1209
Senaryo-2	1182	1071	1140	1274
Senaryo-3	2126	1664	2202	2014

Bu sonuçlara göre, DCSCA algoritması, 1 ve 2. senaryolarda LEACH’e göre daha kötü bir sonuç verirken 3. Senaryoda en iyi sonucu yakalamış ve 2126. iterasyonda son düğüm ölümü gerçekleşmiştir (Panag ve Dhillon, 2018). MEKA algoritması Senaryo 1’e göre 2000 iterasyon çalıştırılmış ve sonunda MEKA’daki düğümlerden 2 tanesinin hayatta kaldığı görülmüştür.

8. Zahedi 2018 yılında yaptığı çalışmada, KB seçiminde farklı katsayılar önermiş ve o katsayıları KB seçim parametresi olarak kullanmıştır. Çalışmanın sonuçlarında, HEED ve LEACH algoritmaları ile karşılaştırılmıştır. Toplam 500 iterasyondan oluşan bir simülasyonda, önerilen katsayılarla yapılan KB seçimine göre ağdaki son düğüm 480. iterasyonda, HEED’de 480. iterasyonda ve LEACH’de ise 380. iterasyonda ölmüştür (Zahedi,2018). Aynı değerlerle MEKA çalıştırıldığında ise son düğüm 770.iterasyonda öldüğü görülmüştür.
9. Behera ve ark. 2018 yılında IoT uygulamaları için LEACH algoritmasını geliştirmişler ve I-LEACH algoritmasını önermişlerdir. Önerdikleri algoritmanın sonuçlarına göre, LEACH algoritmasında ağda kalan son düğüm 1450. iterasyonda ölürken I-LEACH algoritmasında 1750. iterasyonda öldüğü tespit edilmiştir (Behera ve ark. 2018). Aynı verilerle MEKA çalıştırıldığında, 3000 iterasyon sonunda ağdaki düğümlerin yarısının hayatta kaldığı ve son düğümün 4500.iterasyonda öldüğü gözlemlenmiştir.

- 10.** Kumar ve Mehfuz 2019 yılında yaptıkları çalışmada PSO-NMDC algoritmasını önermişler ve CHFL algoritması ile karşılaştırmışlardır. İterasyonlara göre ağda kalan toplam enerji miktarları ve ağdaki son düğümün öldüğü iterasyon sayısına göre ağın ömrü değerlendirilmiştir. CHFL algoritmasında, 320. iterasyonda ağ ömrü tükenirken PSO-NMDC algoritmasında ağ enerjisinin 450. İterasyonda tükendiği bildirilmiştir (Kumar ve Mehfuz, 2019). Bu çalışmada kullanılan simülasyon parametreleri ile MEKA algoritması çalıştırılmış ve 600 iterasyon sonunda ağdaki enerjinin %18,5'inin saklandığı tespit edilmiş ve son düğümün 711.iterasyonda öldüğü görülmüştür. Toplam iterasyon sayısını 1000 kabul edersek, MEKA algoritmasının PSO-NMDC'ye göre %26 daha iyi olduğu belirlenmiştir.
- 11.** Bozorgi ve Bidgoli 2019 yılında yaptıkları çalışmada, KAA'da enerji verimli bir kümeleme için HEEC algoritmasını önermişlerdir. Önerilen algoritma Mİ yönlendirmeli bir yapıdadır ancak bir yönüyle bu tez çalışması ile benzerlik göstermektedir. Algoritmada, KB seçimi ve ağın kümelenmesi Mİ tarafından yapılır ve bu işlem için Mİ'nin düğümlere olan uzaklıkları kullanılır. Uzaklıklar Mİ ile düğümler arasındaki RSSI seviyesinden elde edilmektedir. Önerilen algoritma LEACH, FHRP, EADUC-II ve HUCL algoritmaları ile kıyaslanmıştır. Deneysel çalışmada, Mİ'nin ağa olan uzaklıklarının değiştirildiği 4 farklı senaryo kullanılmıştır. Bu senaryolarda, düğümlerin tamamı LEACH'de 1400.iterasyonda ölürken HEEC'de ise 2000. iterasyonda öldükleri bildirilmiştir (Bozorgi ve Bidgoli, 2019). Çalışmada kullanılan aynı simülasyon değerleriyle MEKA algoritması çalıştırıldığında ise, 2000. iterasyonda ağdaki düğümlerin 87 tanesinin hayatta olduğu gözlemlenmiştir.
- 12.** Karasekreter ve ark. (2020) tarafından yapılan bir makale çalışmasında geliştirilen TSPSO algoritmasıyla ağın kümelenmesi sağlanmış ve LEACH algoritması ile karşılaştırılmıştır. Elde edilen sonuçlara göre LEACH algoritmasında son düğümün 27.iterasyonda öldüğü, TDPSO'da ise son düğümün 47.iterasyonda öldüğü bildirilmiştir. Aynı verilerle MEKA algoritması çalıştırıldığında son düğümün 61.iterasyonda öldüğü tespit edilmiştir.

Tartışmada sunulan verilerin daha iyi anlaşılması için, tartışılan algoritmaların LEACH'e göre sağladığı iyileştirme oranı ve aynı verilerle MEKA'nın LEACH'e göre sağladığı iyileştirme oranları tartışma ilkelerinin 6. Maddesinde verilen örneğe göre hesaplanmış ve elde edilen sonuçlar Çizelge 5.9'de sunulmuştur. Çizelgede, "ALGORİTMA" sütunundaki veriler ilk sütunda isimleri verilen algoritmaları temsil etmektedir.

Çizelge 5.9. Tartışılan algoritmaların sayısal karşılaştırması

	Yıl	Ağ Ömrü İyileştirme Oranı (%)		Son Düğümün Öldüğü İterasyon Sayısı	
		ALGORİTMA	MEKA	ALGORİTMA	MEKA
HEED	2004	11	51	920	1177
RE-EERP	2011	16	64	-	-
Q-LEACH	2013	20	61,67	2900	5400
Mk-means	2016	60	82,5	1920	2190
EECPK	2016	66	82,5	1992	2190
OCM-FC	2018	-	-	600	2400
DHSCA	2018	-4.4	31,64	1099	2000
Zahedi	2018	10	29	480	770
I-LEACH	2019	6,6	67,8	1750	4500
PSO-NMDC	2019	-	-	450	711
HEEC	2019	15	58,5	2000	3740
TDPSO	2020	20	41	47	61

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

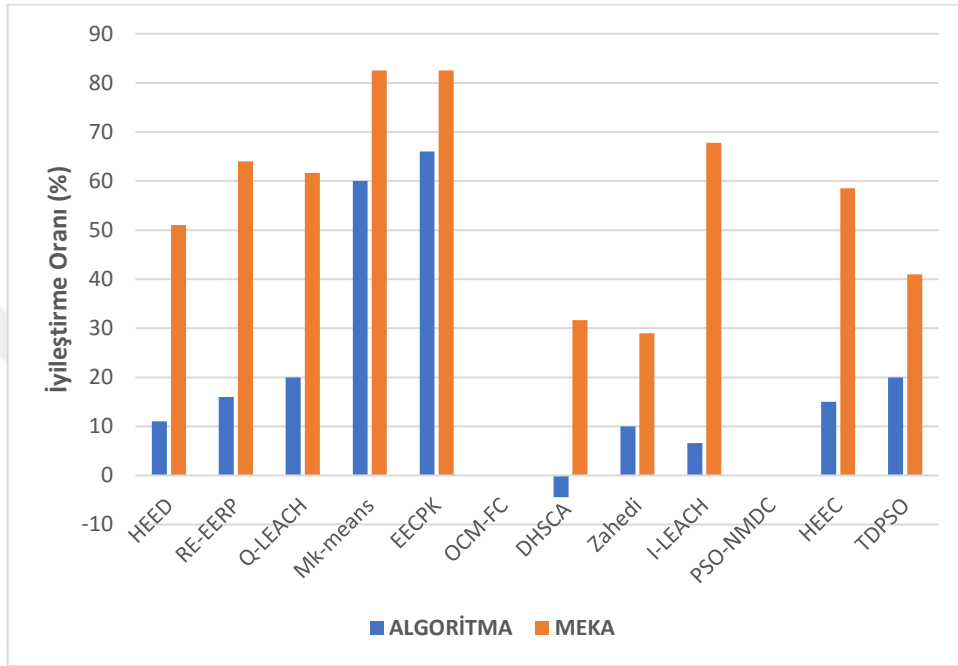
Bu çalışmada düzensiz dağıtık yapıdaki kablosuz algılayıcı ağlarda merkez eğilimli öz örgütlenmeli kümeleme yapabilen yeni bir algoritma önerilmiştir. Önerilen üyelik fonksiyonu ve algoritmanın performansı MATLAB ortamında geliştirilen uygulamalar ile ortaya konulmuştur. Elde edilen bulgular MEKA algoritmasının kendisi gibi öz örgütlü kümeleme yapabilen LEACH' e göre kapsama alanı, hayatta kalan düğüm sayıları, gönderilen veri paketleri ve enerji verimliliği yönünden karşılaştırılmıştır. Elde edilen sonuçlara göre;

- Hayatta kalan düğüm sayıları yönünden %30,
- Gönderilen paket sayıları yönünden % 18,
- Ağda kalan toplam enerji yönünden %58,7,
- Kapsama alanı yönünden ise %13 iyileşme sağlandığı tespit edilmiştir.

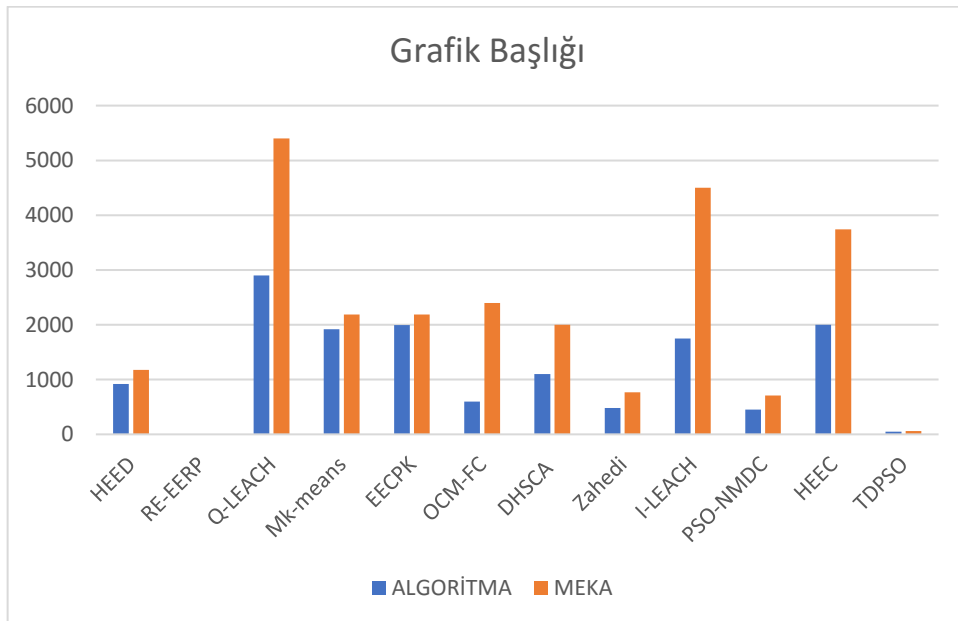
MEKA, öz örgütlü kümeleme yapısında olmamasına rağmen k-means algoritması ile kümeleme karakteristiği ve kapsama alanı yönünden kıyaslanmıştır. Bulgulara göre, k-means' de alanın %78'i kapsama alanına alınırken bu oran MEKA' da %71 olarak tespit edilmiş ve MEKA' nın kapsama alanı yönünden k-means algoritmasına yakın bir davranış sergilediği görülmüştür. MEKA algoritması, literatürdeki diğer öz-örgütlü algoritma olan HEED algoritmasıyla da karşılaştırılmıştır. HEED, LEACH ile son düğümün öldüğü iterasyon sayısı bakımından kıyaslandığında %11 iyileşme sağladığı görülmüştür. HEED algoritmasındaki aynı simülasyon değerleri MEKA algoritması bu oranı %51'e çıkarmayı başarmıştır.

Literatürde, enerji verimini önceleyen tüm kümeleme tabanlı algoritmaların sonuçları LEACH ile kıyaslanmış ve tartışılmıştır. İlk düğümün ve son düğümlerin öldükleri iterasyon sayıları, ağın iterasyonlara göre enerji tüketimleri vb. parametrelere bakılmış ve önerilen algoritmaların LEACH'e göre yaptığı performans iyileştirmeleri bildirilmiştir. Bu tez çalışmasında da aynı şekilde LEACH ile karşılaştırma yapılmıştır. Dolayısıyla elde edilen bulgular direk olarak diğer çalışmaların iyileştirme oranlarıyla karşılaştırılabilir. Ancak her çalışmada parametre değerleri farklı kullanıldığından ve

farklı yönleriyle değerlendirme yapıldığından direk karşılaştırma yapılamamıştır. O nedenle, çalışmada kullanılan değerlerle MEKA çalıştırılmış ve direk sonuçlar karşılaştırılmıştır. MEKA algoritmasının direk uygulanamadığı durumlarda ise yorum yöntemine başvurulmuştur. Elde edilen karşılaştırma bulguları grafiksel olarak Şekil 6.1 ve Şekil 6.2’de sunulmuştur.



Şekil 6.1. MEKA'nın diğer algoritmalara göre ağ ömrü iyileştirme oranı



Şekil 6.2. Son düğümün öldüğü iterasyon sayıları

6.2. Öneriler

KAA, son yılların en önemli konu başlıklarından birisidir ve önemi günden güne daha iyi anlaşılmaktadır. Savunma sanayimiz başta olmak üzere tüm alanlarda insansız teknolojilerin ağırlık kazanması bu konunun önemini vurgulamaktadır. Bahsedilen bu teknolojik gelişim, geleneksel yöntemlerle başlamış olsa da sürdürülebilir olmadığı açıkça görülmektedir. Yeni nesil teknolojilerde ise karar destek yöntemleri veya yapay zekâ entegrasyonu olmazsa olmazlar arasında yerini almaya başlamıştır. O nedenle, tasarlanan sistemlerin veya önerilen algoritmalarında bu yapıya uygun nitelikte olması gerekmektedir. Bu tez kapsamında geliştirilen MEKA algoritması, içinde literatürde bilinen bir yapay zekâ modeli kullanmıyor olsa da yapısı itibariyle düğümlerin kendi kendine organize olabilecekleri bir yapıdadır. İleriki çalışmalarda,

1. MEKA' nın sezgisel yöntemlerle veya bilinen yapay zekâ modelleri ile desteklenmesi düşünülmektedir.
2. MEKA'nın mevcut yapısında, düğümler arasındaki mesafeler bir kere hesaplanır ve KB seçimi buna göre yapılmaktadır. Ancak, birinci KB düğüm seçildikten sonra tekrar mesafelerin hesaplanmasıyla farklı sonuçlar çıkabileceğinden ileriki çalışmalarda denenecektir.

MEKA, basit algoritması ve öz-örgütlü mimarisinden dolayı bir mikrokontrolcünün kapasitesi dahilinde rahatça uygulanabilir olarak tasarlanmıştır. Bu durum MEKA için uygulama alanı yelpazesinin diğerlerine göre daha geniş olduğunu göstermektedir. Orman yangınlarının izlenmesi, teletıp, fizyolojik işaretlerin takibi, açık alan gözetleme sistemleri bunlar arasında sayılabilir. Ayrıca, düğümler arasında farklı mesafe metriklerinin kullanılması ve farklı haberleşme birimlerinin entegrasyonu ile otonom sürü teknolojilerinin geliştirilmesinde de kullanılabilir.

KAYNAKLAR

- Akkaya, K., Senel, F. and McLaughlan, B., 2009, Clustering of wireless sensor and actor networks based on sensor distribution and connectivity, *J. Parallel Distrib. Comput.*, 69, 573-587.
- Akyıldız, İ. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., 2002, Wireless Sensor Networks: a survey, *Computer Networks*, 38, 393-422.
- Alawi, A. R., 2011, RSSI based location estimation in wireless sensors networks, *17th IEEE International Conference, Networks (ICON)*, Singapore 2011,118-122.
- Ali, Q. İ., Abdulmaowjod, A. and Mohammed, H.M., 2011, Simulation & Performance study of wireless sensor network (WSN) using MATLAB, *Iraq J. Electrical and Electronic Engineering*, 7(2), 112-119.
- Altun, A. A., and Şahman, M. A., 2013, Cost optimization of mixed feeds with the particle swarm optimization method, *Neural Computing and Applications*, 22(2), 383-390.
- Amini, N., Vahdatpour, A., Xu, W., Gerla, M. and Sarrafzadeh, M., 2010, Cluster size optimization in sensor networks with decentralized cluster-based protocols, *Computer Communications*, 35, 207-220.
- Amini, N., Vahdatpour, A., Xu, W., Gerla, M. and Sarrafzadeh, M., 2012, Cluster size optimization in sensor networks with decentralized cluster-based protocols, *Computer Communications*, 35, 207-220.
- An-Feng, L., Peng-Hui, Z. and Zhi-Gang, C., 2011, Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks, *J. Parallel Distrib. Comput.*, 71, 1327–1355.
- Aslam, N., Phillips, W., Robertson, W. and Sivakumar, S., 2011, A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks, *Information Fusion*, 12, 202–212.
- Ata, O., Uçar, E. and Balık H. H., 2016, Kablosuz Algılayıcı Ağlarda Kullanılan Teknoloji ve Protokoller Üzerine Bir İnceleme, *İstanbul Aydın Üniversitesi Dergisi*, 3, 51-68.
- Ayran, O., 2007, Reliable Real-Time Video Communication in Wireless Sensor Networks, Yüksek Lisans Tezi, *Ortadoğu Teknik Üniversitesi*, Ankara.
- Bajaber, F. and Awan, I., 2011, Adaptive decentralized re-clustering protocol for wireless sensor networks, *Journal of Computer and System Sciences*, 77, 282–292.
- Basu, M., 2015, Modified particle swarm optimization for nonconvex economic dispatch problems, *International Journal of Electrical Power & Energy Systems*, 69, 304-312.

- Behera, T.M., Samal, U.C. and Mohapatra, S.K., 2018, Energy-efficient modified LEACH protocol for IoT application, *IET Wireless Sensor Systems*, 8(5), 223-228.
- Botta, M. and Simek, M., 2013, Adaptive Distance Estimation Based on RSSI in 802.15.4 Network, *Radioengineering*, 22(4), 1162-1168.
- Bozorgi, S.M. and Bidgoli, A.M., 2019, HEEC: a hybrid unequal energy efficient clustering for wireless sensor networks, *Wireless Network*, 25, 4751–4772.
- Bulut, E. ve Akçacı, T., 2017, Endüstri 4.0 ve İnovasyon Göstergeleri Kapsamında Türkiye Analizi, *ASSAM Uluslararası Hakemli Dergi*, 7, 50-72.
- Chang-jiang, J., Wei-ren, S., Min, X. and Xian-lun, T., 2010, Energy-balanced unequal clustering protocol for wireless sensor networks, *The Journal of China Universities of Posts and Telecommunications*, 17, 94-99.
- Chatterjee, S., Sarkar, S., Hore, S., Dey, N., Ashour, A. S., and Balas, V. E. ,2017, Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings, *Neural Computing and Applications*, 28(8), 2005-2016.
- Chen, Y.T, Yang, C. L., Chang, Y. K. and Chu, C. P, 2009, A RSSI-based algorithm for indoor localization using ZigBee in wireless sensor network, *15th International Conference on Distributed Multimedia Systems (DMS 2009)*, San Francisco, USA, 2009, 70-75.
- Cheng, C. T., Chi K. Tse, Fellow and Lau, F. C. M., 2011, A Clustering Algorithm for Wireless Sensor Networks Based on Social Insect Colonies, *IEEE Sensors Journal*, 11(3), 711-721.
- Chuang, P., Yang, SH. and Lin, CS., 2011, An Energy-efficient clustering approach for wireless sensor networks, *Journal of the Chinese Institute of Engineers*, 32 (7), 951-957.
- Clerc, M., 2004, Discrete particle swarm optimization, illustrated by the traveling salesman problem, In *New optimization techniques in engineering*, Springer, Berlin, Heidelberg, 219-239.
- Correia, L.H.A., Macedo, D. F., dos Santos, A. L., Loureiro, A. F and Nogueira, S., 2007, Transmission power control techniques for wireless sensor networks, *Computer Networks*, 51(17), 4765-4779.
- Çetin, H.E., 2009, Kablosuz Sensor Ağlarının Micaz Tabanlı Biyomedikal Uygulaması, Yüksek Lisans Tezi, *Ege Üniversitesi Fen Bilimleri Enstitüsü*, İzmir.
- Dener, M., 2014, Kablosuz Algılayıcı Ağlarda Güvenli İzleme Sistemi Tasarımı, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 29, 745-754.
- Dengiz, O., 2017, Endüstri 4.0: Üretimde Kavram ve Algı Devrimi, *Makine Tasarım ve İmalat Dergisi*, 15(1), 38-45.

- Devia, N. C., Palanisamyb, V., Baskaranc K. and Prabeelad, S., 2012, A Novel Distance for Clustering to Support Mixed Data Attributes and Promote Data Reliability and Network Lifetime in Large Scale Wireless Sensor Networks, *Procedia Engineering*, 30, 669 – 677.
- Dutta, R., Gupta, S. and Das, M.K., 2012, Efficient Statistical Clustering Techniques for Optimizing Cluster Size in Wireless Sensor Network, *Procedia Engineering*, 38, 1501-1507.
- Elson, J.E., 2003, Time Synchronization in Wireless Sensor Networks, Doktora Tezi, *University of California*, USA.
- Faheem, A., Virrankoski, R. and Elmusrati, M., 2010, Improving RSSI based distance estimation for 802.15.4 wireless sensor networks, *Wireless Information Technology and Systems (ICWITS), IEEE International Conference*, Honolulu, HI 2010,1-4.
- Fazackerley, S., Paeth, A. and Lawrence, R., 2009, Cluster head selection using rf signal strength, *Electrical and Computer Engineering. CCECE '09. Canadian Conference* St. John's, NL, 2009,334-338.
- Feng, X., Gao, Z., Yang, M. and Xiong, S., 2008, Fuzzy distance measuring based on RSSI in wireless sensor network, *Intelligent System and Knowledge Engineering, 3rd International Conference*, Xiamen, ISKE 2008,395-400.
- Fu, Y., Sha, M., Hackmann, G. and Lu, C., 2012, Practical control of transmission power for wireless sensor networks, *Network Protocols (ICNP), 20th IEEE International Conference*, Auistin, TX 2012,1-10.
- Geetha, V., Kallapur, V. P. and Tellajeera, S., 2012, Clustering in wireless sensor networks: Performance comparison of LEACH & LEACH-C protocols using NS2, *Procedia Technology*, 4, 163-170.
- Gong, D., Yang, Y. and Pan, Z., 2013, Energy-efficient clustering in lossy wireless sensor networks, *J. Parallel Distrib. Comput.*, 73, 1323-1336.
- Goswami, P., Yan, Z., Mukherjee, A., Yang, L., Routray and S., Palai, G., 2019, An energy efficient clustering using firefly and HML for optical wireless sensor network, *Optic*, 182, 181-185.
- Gu, X., Yu, J., Yu, D., Wang, G. and Lv, Y., 2014, ECDC: An energy and coverage-aware distributed clustering protocol for wireless sensor networks, *Computers and Electrical Engineering*, 40(2), 384-398.
- Gupta, P. and Sharma, A.K., 2019, Designing of energy efficient stable clustering protocols based on BFOA for WSNs, *J Ambient Intell Human Comput*, 10, 681–700.

- Hacıoğlu, G., Kand, V. F. A. and Sesli, E., 2016, Multi objective clustering for wireless sensor network, *Expert Systems with Applications*, 59, 86-100.
- Hamzah, A., Shurman, M., Al-Jarrah, O. and Taqieddin, E., 2019, Energy-Efficient Fuzzy-Logic-Based Clustering Technique for Hierarchical Routing Protocols in Wireless Sensor Networks, *Sensors*, 19, 561.
- Harmankaya, A. O. (2007) Kablosuz Algılayıcı Ağ Yönlendirme Protokollerinin Karşılaştırılması, Yüksek Lisans, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli.
- Heinzelman W. B., Chandrakasan, A. P. and Balakrishnan, H., 2002, An Application-Specific Protocol Architecture for Wireless Micro sensor Networks, *IEEE Transactions on Wireless Communications*, 1(4), 660-670.
- Heurtefeux, K. and Valois, F., 2012, Is RSSI a good choice for localization in wireless sensor network? *Advanced Information Networking and Applications, (AINA), IEEE 26th International Conference*, Fukuoka 2012, 732-739.
- Hill, J. E., 2003, System Architecture for Wireless Sensor Networks, Doktora Tezi, *University of California, USA*.
- Huilian, F.A.N., 2010, Discrete particle swarm optimization for TSP based on neighborhood, *Journal of Computational Information Systems*, 6(10), 3407-3414.
- Hussain, F. B., 2008, Reliable Transport for Wireless Sensor and Actor Networks, Doktora Tezi, *Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü*, İzmir.
- Hussein, A. A. and Khalid, R., 2019, Improvements of PEGASIS Routing Protocol in WSN, *International Advance Journal of Engineering Research*, 2(11), 1-14.
- İbrahim, A., 2011, Wireless Sensor Networks, Protocols and Applications, Yüksek Lisans Tezi, *Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü*, İzmir.
- İşman, A., 2001, Teknolojinin Felsefi Temelleri, *Sakarya Üniversitesi Eğitim Fakültesi Dergisi*, 1.
- Jens M., Chris A., John C., Alex H. and Damon T., 2006, Sensor network lab exercises using TinyOS and MicaZ motes, *Proceedings of the International Conference on Pervasive Systems & Computing*.
- Jia-qi, X., Lei, W., Can, M. and Lei, S., 2012, Study of impacts of duty-cycle on overlapping multi-hop clustering in wireless sensor networks, *The Journal of China Universities of Posts and Telecommunications*, 19(2), 19-22.
- Jiawei, H., Kamber, M., Han, J. and Kamber, M., & Pei, J., 2012, Data Mining: Concepts and Techniques, San Francisco, CA, *Morgan Kaufmann*.

- Jirkovsk'y, V., Obitko, M. and Marik, V., 2017, Understanding Data Heterogeneity in the Context of Cyber-Physical Systems Integration, *IEEE Transactions on Industrial Informatics*, 13(2), 660-667.
- Kanmaz, M. and Aydın, M.A., 2019, Comparison of dv-hop based indoor positioning methods in wireless sensor networks and new approach with K-means ++ clustering method, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34:2, 975-986.
- Karasabun, E., 2009, Active Node Determination for Correlated Data Gathering in Wireless Sensor Networks, Yüksek Lisans Tezi, *Bilkent Üniversitesi Fen Bilimleri Enstitüsü*, Ankara.
- Karasekreter, N., Şahman, M.A., Başçiftçi, F., and Fidan, U., 2020, PSO-based clustering for the optimization of energy consumption in wireless sensor network, *Emerging Materials Research*, 9(3), 776-783.
- Karataş, M., 2012, Bistatik ve Multistatik Suallı Sensör Ağlarının Optimizasyonu, Doktora Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli.
- Khalil, E. A. and Attea B. A., 2011, Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks, *Swarm and Evolutionary Computation*, 1, 195-203.
- Khan, A., Tamim, I., Ahmed, E. and Awal, M.A., 2012, Multiple parameter-based clustering (MPC): prospective analysis for effective clustering in wireless sensor network (WSN) using K-means algorithm, *Wireless Sensor Network*, 4, 18-24.
- Kim, J., Park, S., Han, Y. and Chung, T., 2008, CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks, *2008 10th International Conference on Advanced Communication Technology*, Gangwon-Do, 654-659.
- Kuila, P., Gupta, K. S. and Jana, K. P., 2013, A novel evolutionary approach for load balanced clustering problem for wireless sensor networks, *J. Parallel Distrib. Comput.*, 71, 1327-1355.
- Kumar, D., Aseri, T. and Patel, R.B., 2009, EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks, *Computer Communications*, 32, 662-667.
- Kumar, S. and Mehruz, S., 2019, A PSO Based Malicious Node Detection and Energy Efficient Clustering in Wireless Sensor Network, *6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 859-863.
- Krunz, M., Siam M.Z. and Nguyen, D.N., 2013, Clustering and power management for virtual MIMO communications in wireless sensor networks, *Ad Hoc Networks*, 11, 1571-1587.

- Lai, W.K., Fan, C. S. and Lin, L.Y., 2012, Arranging cluster sizes and transmission ranges for wireless sensor networks, *Information Sciences*, 183, 117-131.
- Levis, P., Lee, N., Welsh, M. and Culler, D., 2003, TOSSIM: Accurate and scalable simulation of entire TinyOS applications, *Proceedings of the 1st international conference on Embedded networked sensor systems*, 126-137.
- Lin, D., and Wang, Q., 2019, An Energy-Efficient Clustering Algorithm Combined Game Theory and Dual-Cluster-Head Mechanism for WSNs, *IEEE Access*, 7, 49894-49905.
- Lindsey, S and Raghavendra, C.S., 2002, PEGASIS: Power-efficient gathering in sensor information systems, *Proceedings, IEEE Aerospace Conference*, Big Sky, MT, USA, pp. 3-3.
- Liu, T., Li, Q. and Liang, P., 2012, An energy-balancing clustering approach for gradient-based routing in wireless sensor networks, *Computer Communications*, 35, 2150-2161.
- Liu, Z., Zhenga, Q., Xuea, L. and Guana, X., 2012, A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks, *Future Generation Computer Systems*, 28, 780–790.
- Liu Z., a, Wang, j. and Xue, Y., 2013, PCRLB-based sensor selection for maneuvering target tracking in range-based sensor networks, *Future Generation Computer Systems*, 29, 1751–1757.
- Liu, A.P., Xian-You, W., Zhi-Gang, C. and Wei-Hua, G. 2010, Research on the energy hole problem based on unequal cluster-radius for wireless sensor networks, *Computer Communications*, 33, 302–321.
- Liu, X., Zhu, R., Anjum, A., Wang, J., Zhang, H. and Ma, M., 2020, Intelligent Data Fusion Algorithm Based on Hybrid Delay-Aware Adaptive Clustering In Wireless Sensor Networks, *Future Generation Computer Systems*, 104, 1-14.
- Low, C. P., Fang, C., Ng, J. M. and Ang, Y. H., 2008, Efficient Load-Balancing Clustering Algorithms for wireless sensor networks, *Computer Communications*, 31, 750-759.
- Malik, T. A., 2005, Target Tracking in Wireless Sensor Networks, Yüksek Lisans Tezi, *Louisiana State University*, USA.
- Manjeshwar, A. and Agrawal, D.P., 2002, APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, *Parallel and Distributed Processing Symposium*, Proceedings International, Ft. Lauderdale, FL, USA, 2002.

- Manjeshwar, A. and Agrawal, D. P., 2000, TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks, *Parallel and Distributed Processing Symposium., Proceedings 15th International*, DOI:10.1109/IPDPS.2001.925197.
- Manzoor, B., Javaid, N., Rehman, O., Akbar, M., Nadeem, Q., Iqbal, A. and Ishfaq, M., 2013, Q-LEACH: A New Routing Protocol for WSNs, *Procedia Computer Science*, 19, 926-931.
- Marinakis, Y., and Marinaki, M., 2010, A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem, *Computers & Operations Research*, 37(3), 432-442.
- Mathapati, B.S., Patil, S. R. and Mytri, V.D., 2012, Power control with energy efficient and reliable routing mac protocol for wireless sensor networks, *International journal of computer engineering & technology*, 3, 223-231.
- Misic, J., 2007, Algorithm for equalization of cluster lifetimes in a multi-level Beacon enabled 802.15.4 sensor network, *Computer Networks*, 51, 3252-3264.
- Muruganathan, S. D., Ma, D. C. F., Bhasin, R. I. and Fapojuwo, A. O., 2005, A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks, *IEEE Radio Communications*, 43(3), 8-13.
- Musznicki, B. and Zwierzykowski, P., 2012, Survey of simulators for wireless sensor networks, *International Journal of Grid and Distributed Computing*, 5(3), 23-49.
- Okay, F. Y. ve Özdemir, S., 2014, Kablosuz Algılayıcı Ağlarda Ortam Erişim Protokolleri, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 29, 49-59.
- Onur, E., 2007, Deployment Quality Measures in Surveillance Wireless Sensor Networks, Doktora Tezi, *Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.
- Osamy, W., Salim, A. and Khedr, A.M., 2020, An information entropy based-clustering algorithm for heterogeneous wireless sensor networks. *Wireless Netw.*, 26, 1869–1886.
- Özcan, Ö., 2011, Kablosuz Sensör Ağları İçin PIC Tabanlı Sensör Düğümü Tasarımı, Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya.
- Özkan, M., Al, A. ve Yavuz, S., 2018, Uluslararası Politik Ekonomi Açısından Dördüncü Sanayi-Endüstri Devrimi'nin Etkileri ve Türkiye, *Marmara Üniversitesi Siyasal Bilimler Dergisi*, 1 (1), 1-30.
- Panag, T.S., Dhillon, J.S., 2018, Dual head static algorithm for wireless sensor networks, *Int. J. Electron. Commun.*, 88, 148-156.
- Periyasamy, S., Khara, S. and Thangavelu, S., 2016, Balanced Cluster Head Selection Based on Modified k-means in a Distributed Wireless Sensor Network, *International Journal of Distributed Sensor Network*, 12(3), 1-11.

- Pitchaimanickam, B. and Murugaboopathi, G., 2020, A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks, *Neural Comput & Applic*, 32, 7709–7723.
- Qin, J., Fu, W., Gao, H. and Zhen, WX., 2017, Distributed k-means Algorithm and Fuzzy c-Means Algorithm for Sensor Networks Based on Multiagent Consensus Theory, *IEEE Transactions on Cybernetics*, 47(3), 772-783.
- Rathi, N., Saraswat, J. and Btattacharya, P.P., 2012, A Review on Routing Protocols for Application in Wireless Sensor Networks, *International Journal of Distributed and Parallel Systems*, 3(5), 39-58.
- Ray, A. and De, D., 2016, Energy Efficient Clustering Protocol based on k-means (EECPK-Means)-midpoint algorithm for enhanced network lifetime in wireless sensor network, *IET Wireless Sensor Systems*, 6(6), 181-191.
- Rocha, A. R., Pirmez, L., Delicato, F. C., Lemos, E., Santos, I., Gomes, D.G. and Souza, J.S., 2012, WSNs clustering based on semantic neighborhood relationships, *Computer Networks*, 56, 1627-1645.
- Sarkar, A. and Murugan, S.T, 2016, Routing protocols for wireless sensor networks: What the literature says?, *Alexandria Engineering Journal*, 55, 3173-3183.
- Shah, G.A., 2007, An Energy-Efficient Real-Time Coordination and Routing Framework for Wireless Sensor Actor Networks, Doktora Tezi, *Ortadoğu Teknik Üniveristesi*, Ankara.
- Shabbir, N. and Hassan, S.R., 2017, Routing protocols for wireless sensor networks (WSNs), *Wirel. Sens. Netw. Insights Innov.*, DOI: 10.5772/intechopen.70208
- Sheu, J. P., Hsieh, K. Y. and Cheng, K. Y., 2009, Distributed Transmission Power Control Algorithm for Wireless Sensor Networks, *Journal of information science and engineering*, 25(5), 1447-1463.
- Singh, S.K., Kumar, P. and Singh, J.P., 2017, A Survey on Successors of LEACH Protocol, *IEEE Access*, 5, 4298-4328.
- Soro, S. and Heinzelman, W.B., 2009, Cluster head election techniques for coverage preservation in wireless sensor networks, *Ad Hoc Networks*, 7, 955-972.
- Su, S. and Zhao, S., 2018, An optimal clustering mechanism based on Fuzzy-C means for wireless sensor networks, *Sustainable Computing: Informatics and Systems*, 18, 127-134.
- Taş, V., 2010, Kablosuz Sensör Ağlarının Karasal Hudut Güvenliğinde Kullanılması Hakkında Bir Öneri, Yüksek Lisans Tezi, *Beykent Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul.

- Torkestani, J. A. and Meybodi, M. R., 2011, LLACA: An adaptive localized clustering algorithm for wireless ad hoc networks, *Computers and Electrical Engineering*, 37, 461-474.
- Tümer, A.E., 2011, Bina İçi Kablosuz Algılayıcı Ağlar İçin Enerji Verimli Yönlendirme Protokollerinin Geliştirilmesi, Doktora Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*, Konya.
- Wang, K. P., Huang, L., Zhou, C. G., and Pang, W., 2003, Particle swarm optimization for traveling salesman problem, *In Machine Learning and Cybernetics*, 2003 International Conference, 3, 1583-1585.
- Wang, Q., Lin, D., Yang, P and Zhang, Z., 2019, An Energy-Efficient Compressive Sensing-Based Clustering Routing Protocol for WSNs, *IEEE Sensors Journal*, 19(10), 3950-3960.
- Wanga, A., Yang, D. and Sun, D., 2012, A clustering algorithm based on energy information and cluster heads expectation for wireless sensor networks, *Computers and Electrical Engineering*, 38, 662-671.
- Yılmaz, M., 2007, Duty Cycle Control in Wireless Sensor Networks, Yüksek Lisans Tezi, *Ortadoğu Teknik Üniversitesi*, Ankara.
- Yonga, Z. and Peia, Q., 2012, An Energy-Efficient Clustering Routing Algorithm Based on Distance and Residual Energy for Wireless Sensor Networks, *Procedia Engineering*, 29, 1882-1888.
- Younis, O. and Fahmy, S., 2004, HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks, *IEEE Transactions on Mobile Computing*, 3(4), 366-379.
- Yu, H. and Xiaohui, W., 2011, PSO-based energy-balanced double cluster-heads clustering Routing for wireless sensor networks, *Procedia Engineering*, 15, 3073-3077.
- Yuea, J., Zhang, W., Xiao, W., Tang, D. and Tang, J., 2012, Energy efficient and balanced cluster-based data aggregation algorithm for wireless sensor networks, *Procedia Engineering*, 29, 2009-2015.
- Zahedi, A., 2018, An efficient clustering method using weighting coefficients in homogeneous wireless sensor networks, *Alexandria Engineering Journal*, 58, 695-710.
- Zalik, K. R., 2008, An efficient k-means clustering algorithm, *Pattern Recognition Letters*, 29(9), 1385-1391.
- Zhang, Y., Ng, J.M. and Low, C.P., 2009, A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks, *Computer Communications*, 32, 189-202.

Zhou, H., Wu, Y., Hu, Y. and Xie, G., 2010, A novel stable selection and reliable transmission protocol for clustered heterogeneous wireless sensor networks, *Computer Communications*, 33, 1843-1849.



TURNİTİN

KABLOSUZ ALGILAYICI AĞLARDA RSSI TABANLI KÜMELEME ALGORİTMASININ GELİŞTİRİLMESİ

by Naim Karasekreter

Submission date: 21-Dec-2020 10:58PM (UTC+0300)

Submission ID: 1480257263

File name: naim_tez.docx (5.07M)

Word count: 27178

Character count: 177394

KABLOSUZ ALGILAYICI AĞLARDA RSSİ TABANLI KÜMELEME ALGORİTMASININ GELİŞTİRİLMESİ

ORIGINALITY REPORT

7%	6%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Selçuk Üniversitesi Student Paper	1%
2	etd.lib.metu.edu.tr Internet Source	1%
3	acikerisim.deu.edu.tr Internet Source	<1%
4	silo.pub Internet Source	<1%
5	Submitted to Kahramanmaraş Sütçü İmam University Student Paper	<1%
6	acikerisim.selcuk.edu.tr:8080 Internet Source	<1%
7	repository.bilkent.edu.tr Internet Source	<1%
8	www.scribd.com Internet Source	<1%

9	docplayer.biz.tr Internet Source	<1%
10	Submitted to Osmaniye Korkut Ata University (deneme) Student Paper	<1%
11	acikerisim.pau.edu.tr Internet Source	<1%
12	Submitted to TechKnowledge Turkey Student Paper	<1%
13	www.cmpe.boun.edu.tr Internet Source	<1%
14	www.hindawi.com Internet Source	<1%
15	www.mdpi.com Internet Source	<1%
16	Naim Karasekreter, Uğur Fidan, Fatih Başçiftçi. "Energy Efficient Clustering Algorithm for WSN with Distributed Structure", Emerging Materials Research, 2020 Publication	<1%
17	www.springerprofessional.de Internet Source	<1%
18	link.springer.com Internet Source	<1%

19	www.ahmetcevahircinar.com.tr Internet Source	<1 %
20	Prateek Gupta, Ajay K. Sharma. "Designing of energy efficient stable clustering protocols based on BFOA for WSNs", Journal of Ambient Intelligence and Humanized Computing, 2018 Publication	<1 %
21	www.researchgate.net Internet Source	<1 %
22	www.megoras.com.tr Internet Source	<1 %
23	www.iaeme.com Internet Source	<1 %
24	hasanbalik.com Internet Source	<1 %
25	www.openaccess.hacettepe.edu.tr:8080 Internet Source	<1 %
26	ermt.net Internet Source	<1 %
27	www.freepatentsonline.com Internet Source	<1 %
28	acikerisim.istanbul.edu.tr Internet Source	<1 %

"Intelligent Computing, Networking, and

29	Informatics", Springer Science and Business Media LLC, 2014 Publication	<1%
30	Submitted to Istanbul Aydin University Student Paper	<1%
31	Submitted to University of Hertfordshire Student Paper	<1%
32	openaccess.maltepe.edu.tr Internet Source	<1%
33	"Inventive Computation Technologies", Springer Science and Business Media LLC, 2020 Publication	<1%
34	polen.itu.edu.tr Internet Source	<1%
35	Submitted to Konya Necmettin Erbakan University Student Paper	<1%
36	arro.anglia.ac.uk Internet Source	<1%
37	hdl.handle.net Internet Source	<1%
38	Recep Ozdag, Ali Karci. "A novel routing protocol based on EM-L algorithm for energy efficiency of Wireless Sensor Networks", 2015 23rd Signal Processing and Communications	<1%

Applications Conference (SIU), 2015

Publication

39	pt.scribd.com	<1%
Internet Source		
40	docplayer.net	<1%
Internet Source		
41	traglor.cu.edu.tr	<1%
Internet Source		
42	en.wikipedia.org	<1%
Internet Source		
43	Devika G., Ramesh D., Asha Gowda Karegowda. "chapter 7 A Study on Energy-Efficient Wireless Sensor Network Protocols", IGI Global, 2020	<1%
Publication		
44	Naim Karasekreter, Mehmet Akif Şahman, Fatih Başçiftçi, Uğur Fidan. "PSO Based Clustering for the Optimization of Energy Consumption in Wireless Sensor Network", Emerging Materials Research, 2020	<1%
Publication		
45	Nauman Israr. "Multilayer Cluster Based Energy Efficient Routing Protocol for Wireless Sensor Networks", International Journal of Distributed Sensor Networks, 04/2008	<1%
Publication		

46	www.tojsat.net Internet Source	<1%
47	"Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014", Springer Science and Business Media LLC, 2015 Publication	<1%
48	www.degruyter.com Internet Source	<1%
49	edergi.bilecik.edu.tr Internet Source	<1%
50	ŞEKER, Soner, DOĞAN ² , Erkan and KOZANOĞLU ² , Celal. "Yarı Rijit Çelik Çerçevelerin Stokastik Arama Yöntemleri Kullanılarak Optimum Boyutlandırılması", Celal Bayar Üniversitesi, 2017. Publication	<1%
51	acikerisim.ktu.edu.tr Internet Source	<1%
52	indexonensorship.org:8080 Internet Source	<1%
53	Murat DENER. "WiSeN: A new sensor node for smart applications with wireless sensor networks", Computers & Electrical Engineering,	<1%

2017

Publication

54	Ahmed H. Gabr. "Effect of Sensor Mobility and Channel Fading on Wireless Sensor Network Clustering Algorithms", Computer and Information Science, 12/28/2011 Publication	<1%
55	www.inass.org Internet Source	<1%
56	www.mmfdergi.gazi.edu.tr Internet Source	<1%
57	Computer Communications and Networks, 2009. Publication	<1%
58	usir.salford.ac.uk Internet Source	<1%
59	Muhammad Akram, Tae Cho. "Energy Efficient Fuzzy Adaptive Verification Node Selection-Based Path Determination in Wireless Sensor Networks", Symmetry, 2017 Publication	<1%
60	ktun.edu.tr Internet Source	<1%
61	Liu, Xuxun. "A Survey on Clustering Routing Protocols in Wireless Sensor Networks", Sensors, 2012.	<1%

Publication

62 Jain, Aarti, and B.V. Ramana Reddy. "Eigenvector centrality based cluster size control in randomly deployed wireless sensor networks", *Expert Systems with Applications*, 2015.

Publication

63 Bharat Bhushan, Gadadhar Sahoo. "Recent Advances in Attacks, Technical Challenges, Vulnerabilities and Their Countermeasures in Wireless Sensor Networks", *Wireless Personal Communications*, 2017

Publication

64 Sasikumar Periyasamy, Sibaram Khara, Shankar Thangavelu. "Balanced Cluster Head Selection Based on Modified -Means in a Distributed Wireless Sensor Network ", *International Journal of Distributed Sensor Networks*, 2016

Publication

65 Lahcène Dehni, Younès Bennani, Francine Krief. "Chapter 39 LEA2C: Low Energy Adaptive Connectionist Clustering for Wireless Sensor Networks", *Springer Science and Business Media LLC*, 2005

Publication

66 web-tools.uts.edu.au

Internet Source

<1%

Exclude quotes Off

Exclude matches < 5 words

Exclude bibliography On

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : NAİM KARASEKRETER
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Bolvadin-Afyonkarahisar/1980
Telefon : 0505 933 25 57
Faks : -
E-Posta : karasekreter@aku.edu.tr

EĞİTİM

Derece	Adı	Bitirme Yılı
Lise	: Pendik Lisesi	1997
Üniversite	: Doğu Akdeniz Üniversitesi	2003
Yüksek Lisans	: Afyon Kocatepe Üniversitesi	2009
Doktora	: Konya Teknik Üniversitesi	Devam E.

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2003-2004	Sampaş A.Ş.	YAZILIM UZM.
2004-2006	Serbest Ticaret	-
2006-	Afyon Kocatepe Üniversitesi	ARŞ.GRV.

UZMANLIK ALANI

- Kablosuz Algılayıcı Ağlar, Yazılım Teknolojileri ve Gömülü Sistemler

YABANCI DİLLER

- İngilizce

YAYINLAR

7.1. Uluslararası hakemli dergilerde yayımlanan makaleler (SCI, SCI-E)

- 1) Karasekreter N, Fidan U and Başçiftçi F (2020), Energy-efficient clustering algorithm for a WSN with a distributed structure, Emerging Materials Research 9(3): 784–788
- 2) Karasekreter N, Şahman MA, Başçiftçi F and Fidan U, (2020), PSO-based clustering for the optimization of energy consumption in wireless sensor network, Emerging Materials Research 9(3): 776–783
- 3) Karasekreter, N., Başçiftçi., F., Fidan, U., “A New Suggestion of Irrigation Schedule with Artificial Neural Network”, The Journal of Experimental and Theoretical Artificial Intelligence, Vol:25-1, pp:93-104, 2013

7.2 Uluslararası diğer hakemli dergilerde yayınlanan makaleler

- 1) Naim Karasekreter, Fatih Başçiftçi, Uğur Fidan, “RSSI Based Clustering Algorithm For Wireless Sensor Networks”, The Online Journal of Science and Technology, 7-1, pp. 97-102, 2017
- 2) Uğur Fidan, Naim Karasekreter, “Designing And Implementation Of Outdoor Transformer Security System”, The Online Journal of Science and Technology, 7-3, pp. 34-40, 2017
- 3) N. Karasekreter, U. Fidan, “Comparison Spectral Analysis Methods According To The Performance And Selectivity For The S1-S2 Heart Sounds” e-Journal of New World Sciences Academy, Volume: 6, Number: 2, pp: 620-630, 2011
- 4) U. Fidan, N. Karasekreter, “Gsm/Sms Tabanlı Sulama Otomasyonu Kontrol Biriminin Geliştirilmesi ve Uygulanması”, e-Journal of New World Sciences Academy, Volume: 6, Number: 1, pp: 71-77, 2011

7.3. Uluslararası bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler

- 1) Y. Sarpkaya, N.Karasekreter, M. Doğan, "Uzaktan Eğitim Yazılım Altyapısının Bilginin Kalıcılığına ve Geçerliliğine Etkisi", Akademik Bilişim 2007 Dumlupınar Üniversitesi, 31 Ocak-2 Şubat 2007, Kütahya/TÜRKİYE
- 2) Ö. Deperlioğlu, G. E. Güraksın, N. Karasekreter, "Dc-Dc Power Converters Control with Fuzzy Logic", International Wokshop On Nonlinear Signal And Image Processing 2007, Polytechnica University Of Bucharest, 10 Eylül-12 Eylül 2007, Bükreş/Romanya
- 3) İ. Yılmaz, İ.Tiryakioğlu, F.Taktak, A.K.Telli, N. Karasekreter, “Attribute Database for Dynamic Web Based GIS: Campus Sample”, 5.Uluslararası Coğrafi Bilgi Sistemleri Konferansı, 02–05 Temmuz 2008
- 4) Ö. Deperlioğlu, N. Karasekreter, U. Fidan, “Oskültasyon Seslerinin PDA Ortamında Görüntülenmesi ve Dinlenmesi”, 5. International Advanced Technologies Symposium, 13-15 Mayıs 2009, Karabük / Türkiye
- 5) U. Fidan, N. Karasekreter, G. Ütebay, “Ekg ve Fkg İşaretlerinin Kablosuz Olarak Bilgisayar Ortamında Gerçek Zamanlı Görüntülenmesi”, 15. Biyomedikal Mühendisliği Ulusal Toplantısı, BİYOMUT 2010, 21-24 Nisan 2010, Antalya/Türkiye

- 6) Karasekreter, N., Fidan, U., “Developing Agricultural Irrigation Technology Compatible With National Energy Efficiency Policy”, Innovations in Intelligent Systems and Applications (INISTA),19-21 Haziran 2013, Albena/Bulgaristan, Doi: 10.1109/INISTA.2013.6577661
- 7) Karasekreter, N., Başçiftçi, F., Fidan, U., “RSSI Based Clustering Algorithm For Wireless Sensor Networks”, ISTEC 2016, Vienna-Austria, July 13-15, 2016
- 8) Uğur Fidan, Naim Karasekreter, “Designing And Implementation Of Outdoor Transformer Security System”, ISTEC 2016, Vienna-Austria, July 13-15, 2016
- 9) Karasekreter, N., “Bolvadin’in Sosyo-Ekonomik Gelişmişlik Düzeyinin İncelenmesi ve Yeni Bir Kentsel Fonksiyon Önerisi”, Uluslararası Bolvadin Sempozyumu, 13-15 Ekim 2017 / Bolvadin- Afyonkarahisar
- 10) Karasekreter, N., Fidan, U., Başçiftçi, F., “Wireless Sensor Network Based System Design and Application in Concept of Internet of Things (IoT): Open Area Security System (OASS)”, International Advanced Researches & Engineering Congress, Osmaniye/Türkiye, 16-18 November 2017
- 11) Karasekreter, N., Fidan, U., Başçiftçi, F., “Application of Chlorine Monitoring System based on Internet of Things (IoT)”, International Advanced Researches & Engineering Congress, Osmaniye/Türkiye, 16-18 November 2017
- 12) Karasekreter, N., Fidan, U., Başçiftçi, F., “Investigation of Clustering Algorithms Used in Wireless Sensor Networks, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:11, No:10, 2017
- 13) Karasekreter, N., Fidan, U., Başçiftçi, F., “Self-organizing Clustering Protocol Recommendation and Comparative Complexity Analysis”, 5th International Conference On Computational And Experimental Science And Engineering (ICCESEN-2018) 12-16 October 2018, Antalya-Türkiye
- 14) Karasekreter, N., Fidan, U., Başçiftçi, F., “Optimization of Parameters Used in Threshold Values and Membership Functions in Clustering Protocols with PSO”, 5th International Conference On Computational And Experimental Science And Engineering (ICCESEN-2018) 12-16 October 2018, Antalya-Türkiye

7.5. Ulusal hakemli dergilerde yayımlanan makaleler

- 1) Fidan,U., Çalığışu, İ., Karasekreter, N., “Dkd R5-7 Standardına Göre Etüv Cihazı Kalibrasyonunun Gerçekleştirilmesi”, Selçuk Üniversitesi Teknik Online, Cilt:9, Sayı:2, S: 175-191, 2010.

8. PROJELER

- 1) Kalp Seslerinin Sayısal Analiz Yöntemleri ile Sınıflandırılması”, BAP (07.AFMYO.2007), 2009, Tamamlandı, Bütçe: 7.500 TL
- 2) Kablosuz Afyon Kocatepe Üniversitesi Sulama Otomasyon Sisteminin Gerçekleştirilmesi ve Enerji Tasarrufuna Etkisinin Ölçülmesi Afyon Kocatepe Üniversitesi Bilimsel Araştırma Komisyonu, BAP, No: 12.MUH.10, 2010, Tamamlandı, Bütçe: 12.500 TL
- 3) Ulusal Enerji Verimliliği Politikası ile Uyumlu Tarımsal Sulama Teknolojisinin Afyonkarahisar ve Çevresinde Uygulanması, BAP, No: 12 Tematik 04, 2012, Tamamlandı, Bütçe: 40.000 TL
- 4) Mikrobiyolojik Disk Difüzyon Yöntemi ile Elde Edilen Görüntülerin Gerçek Zamanlı Görüntü İşleme Teknikleri ile Analizi, BAP, No: 13MUH09, 2013, Tamamlandı, Bütçe:15.000 TL
- 5) Açık Alan PIR Detektör Karakterlerinin Geliştirilmesi ve Yeni Nesil Trafo Güvenlik Sistemi Tasarımı, KOSGEB, 2013-2016. Bütçe: 259.000 TL
- 6) YAREN: Yalnızlık Korkunuz Olmasın, Gözünüz Arkada Kalmasın, TUBITAK, 2019, Bütçe: 2.500 TL
- 7) Gerçek Zamanlı Uzaktan İzleme Sistemi Tasarımı ve Kontrol Birimi Yazılımının Geliştirilmesi, BAP Tematik, 2018, Bütçe: 99.900 TL

9. PATENTLER

- 1) GSM/SMS Kontrollü Programlanabilir Sulama Otomasyon Cihazı, No: 2011/09885, Belge Tarihi: 2014
- 2) Açık Alan PIR Detektör Karakterlerinin Geliştirilmesi ve GSM/GPRS/TCP-IP/SMS/MMS Tabanlı Yeni Nesil Trafo Güvenlik Sistemi, No:2013/04712, Belge Tarihi: 2018

10. HAKEMLİKLER

- 1) Computers and Electronics in Agriculture (SCI)
- 2) Journal of Medical Systems (SCI)
- 3) Applied Artificial Intelligence (SCI)
- 4) Turkish Journal of Electrical Engineering & Computer Sciences (SCI)
- 5) Iranian Journal of Fuzzy Systems
- 6) World Journal of Engineering (E-SCI)
- 7) Afyon Kocatepe Üniversitesi Mühendislik Bilimleri Dergisi (TR-Index)
- 8) International Journal of Intelligent Systems and Applications in Engineering (TR-Index)

11. ÖDÜLLER

- 1- Tübitak Patent Teşvik Ödülü, 2012
- 2- Tübitak Patent Teşvik Ödülü, 2013