

T.C. MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

KABLOSUZ ALGILAYICI AĞLARDA OPTİMUM YAŞAM SÜRESİ İÇİN
ENERJİ VERİMLİ SEZGİSEL BİR YÖNTEM GELİŞTİRİLMESİ

DOKTORA TEZİ

Hüseyin Fehmi Selim BAYRAKLI


Tez Danışmanı


Yrd. Doç. Dr. Şenol Zafer ERDOĞAN


İSTANBUL - 2013

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 25/06/2013 tarih ve 2013/09 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği Doktora Tezi** olarak kabul edilmiştir.

JÜRİ



Prof.Dr. İlhami YAVUZ
(Başkan)


Yrd.Doç.Dr. Şenol Zafer ERDOĞAN
(Danışman)


Yrd.Doç.Dr. Önder TOMBUŞ
Üye

Yrd.Doç.Dr. Turgay BİLGİN
Üye




Yrd.Doç.Dr. Fatih YÜCALAR
Üye

ÖZET

Bu tezde kablosuz algılayıcı ağların yaşam süresini optimize etmek için genetik algoritma tabanlı bir yöntem (GABEEC) önerilmektedir. Önerilen yöntem LEACH algoritması gibi kümeleme tabanlı bir metottur. Ağın yaşam süresini maksimize etmek için genetik algoritma kullanılmıştır. Yöntemin iki safhası vardır. Kurulum safhası ve iletişim safhası. Kurulum safhasında kümeler oluşturulur ve oluşturulan kümeler ağ yaşam süresi boyunca sabit kalır. Her turda dinamik olarak değişen küme başları olan sabit kümeler vardır. Önerilen yöntemi doğrulamak için Visual C# 2010 program geliştirme ortamı kullanılarak bir simülatör geliştirilmiştir. Yöntem simüle edilip alınan sonuçlar diğer ilgili çalışmalarla karşılaştırılmıştır. Sonuçlar önerilen yöntemin diğer yöntemlerden daha iyi sonuç ürettiğini göstermektedir.

Tez çalışmasında, önerilen GABEEC yöntemi, küme başları arasındaki iletişim metodu değiştirilerek geliştirilmiştir. Ağın yaşam süresini uzatmak için çok atlamalı (multihop) iletişim kullanılmıştır. Yöntemin adı Multihop-GABEEC (M-GABEEC) olup simülasyon sonuçları M-GABEEC yönteminin GABEEC'e göre daha enerji verimli olduğunu göstermektedir.

Bu tez 2013 yılında tamamlanmıştır ve 102 sayfadan oluşmaktadır.

Anahtar kelimeler: algılayıcı ağlar, genetik algoritma, enerji verimli, çok atlama, kümeleme

ABSTRACT

In this thesis, a genetic algorithm based energy efficient clustering (GABEEC) is proposed to optimize the lifetime of wireless sensor networks (WSNs). The proposed method is cluster based method like LEACH and generates energy efficient clusters by considering several aspects of the network. The WSN is represented as a graph and the genetic algorithm (GA) is used to obtain the approximate near-optimal solution in order to maximize the lifetime of the network. GABEEC generates energy efficient clusters, where cluster heads are changed dynamically. A simulator is developed in MS Visual C# 2010 development environment to validate the proposed method. The simulation results show that the proposed method outperforms the other methods.

Further in this thesis, GABEEC is improved by using multi-hop communication between cluster heads. Multi-hop communication has been used to extend the lifetime of the network. The name of the method is Multihop-GABEEC (M-GABEEC). The simulation results reveal that multi-hop approach is even more energy-efficient than single-hop approach, GABEEC.

This thesis has been completed in 2013 and consists of 102 pages.

Keywords: sensor networks, genetic algorithm, energy-efficient, multi-hop, clusters, static clustering

TEŐEKKÜR

Bu tez konusunu seçmemde beni yönlendiren, destek ve yardımlarını esirgemeyen danışman hocam sayın Yrd. Doç. Dr. Őenol Zafer ERDOĐAN'a, ikinci danışman görevini üstlenen hocam sayın Yrd. Doç. Dr. Önder TOMBUŐ'a, tezi tamamlamam için bana gereken fırsatı sağlayan Maltepe Üniversitesi Mühendislik ve Dođa Bilimleri Fakültesi Dekanı sayın Prof. Dr. Murat TAYLI'ya, maddi ve manevi desteklerini benden hiçbir zaman esirgemeyen eşime, anneme, babama ve kardeşlerime teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	iii
ABSTRACT.....	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER	vi
KISALTMALAR	x
ŞEKİL LİSTESİ.....	xii
ÇİZELGE LİSTESİ.....	xiv
1. GİRİŞ.....	1
1.1 Kablosuz Algılayıcı Ağlar.....	2
1.2 Kablosuz Algılayıcı Düğümün Bileşenleri	3
1.2.1 Algılama birimi	4
1.2.2 İşlemci birimi	5
1.2.3 Alıcı-verici birimi.....	6
1.2.4 Güç birimi	6
1.3 Kablosuz Algılayıcı Ağların Üstünlükleri.....	7
1.4 Kablosuz Algılayıcı Ağların Geleneksel Ağlarla Karşılaştırılması	7
1.5 Kablosuz Algılayıcı Ağların Uygulama Alanları.....	9
1.5.1 Askeri uygulamalar	9
1.5.2 Çevresel gözlemeleme uygulamaları	10
1.5.1 Sağlık uygulamaları.....	12
1.5.2 Endüstriyel uygulamalar	14
2. KABLOSUZ ALGILAYICI AĞLARDA KÜMELEME	16
2.1 Farklı Uygulamaların Ağ Mimarisine Etkisi.....	18

2.2	Küme Tasarımında Göz Önüne Alınması Gereken Nitelikler	19
2.3	Kümeleme Süreci	20
2.3.1	Küme başı seçimi	20
2.3.2	Algılayıcı düğümün küme başını seçmesi.....	21
2.4	Kümeleme Algoritmaları.....	21
2.4.1	LEACH (Low-Energy Adaptive Clustering Hierarchy)	21
2.4.2	TL-LEACH (Two Level LEACH).....	25
2.4.3	HEED (Hybrid Energy Efficient Distributed Clustering).....	27
2.4.4	DWEHC (Distributed Weight-Based Energy-Efficient Hierarchical Clustering).....	28
2.4.5	EECS (Energy Efficient Clustering Scheme).....	29
2.4.6	HCR (Hierarchical Cluster-Based Routing).....	30
2.4.7	MR-LEACH (Multihop Routing with LEACH)	32
2.4.8	E-LEACH (Energy LEACH)	33
2.4.9	BEC (Based Energy Clustering)	33
2.4.10	LEACH-DCHS (LEACH-Deterministic Cluster-Head Selection)	34
2.4.11	LEACH-C (LEACH-Centralized).....	34
2.4.12	BCDCP (Base-Station Controlled Dynamic Clustering Protocol).....	36
2.4.13	LCA (Linked Cluster Algorithm).....	36
2.4.14	FLOC (Fast Local Clustering Service).....	37
2.4.15	Multihop-LEACH	37
2.4.16	MS-LEACH (Multihop-Singlehop LEACH)	38
2.4.17	ECHSSD (Efficient Cluster Head Selection Scheme for Data Aggregation in Wireless Sensor Network)	38
3.	GENETİK ALGORİTMA.....	40
3.1	Genetik Algoritmanın Bazı Uygulamaları	40

3.1.1	Bazı özel problemlere uygulanışı.....	41
3.2	Genetik Algoritmanın İşleyişi.....	42
3.2.1	Kromozom kodlaması.....	43
3.2.2	Seçim.....	45
3.2.3	Çaprazlama.....	48
3.2.4	Mutasyon.....	49
3.3	Genetik Algoritma Parametreleri.....	51
4.	Kablosuz algılayıcı ağlarda kümeleme optimizasyonu.....	54
4.1	Genetik Algoritma ile Optimum Kümelerin Bulunması.....	54
4.1.1	Genetik algoritma kullanılarak algılayıcı ağ optimizasyonu.....	55
4.1.2	Genetik algoritma kullanılarak algılayıcı ağların kendi kendine organizasyonu.....	57
4.1.3	Dağıtık genetik algoritma kullanılarak algılayıcı ağlar için yaşam süresi bilinçli kaynak yönetimi.....	59
4.1.4	Büyük çaplı gözetim uygulamalarında kablosuz algılayıcı ağın yaşam süresini geliştirmek için etkili bir genetik algoritma.....	60
4.1.5	Hiyerarşik kablosuz algılayıcı ağlar için genetik algoritma.....	61
4.1.6	İki katmanlı algılayıcı ağlarda enerji verimli yönlendirme için genetik algoritma tabanlı bir yaklaşım.....	64
4.1.7	Hareketli tasarsız ağlarda kümelemeyi optimize etmek için genetik algoritma kullanımı.....	66
4.1.8	Çevresel ölçümler için algılayıcı ağlarda enerji optimizasyonu.....	67
4.1.9	Genetik algoritma kullanılarak kablosuz algılayıcı ağlar için yeni bir kümeleme protokolü.....	69
5.	KABLOSUZ ALGILAYICI AĞLARDA GENETİK ALGORİTMA TABANLI ENERJİ VERİMLİ KÜMELER.....	71
5.1	Yöntemin İşleyişi.....	71

5.2	Genetik Algoritmanın Uyarlanması	72
5.2.1	Uygunluk fonksiyonu	72
5.3	Çok Atlamalı (Multihop) İletişim ile GABEEC'in İyileştirilmesi.....	73
5.4	Simülasyon ve Sonuçlar	74
6.	SONUÇ	80
	KAYNAKLAR	82
	ÖZGEÇMİŞ	88

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
GABEEC	Genetik Algorithm Based Energy Efficient Clusters	Genetik Algoritma Tabanlı Enerji Verimli Kümeler
M-GABEEC	Multihop-GABEEC	Çok atlamalı GABEEC
SHARC	Super Harvard Architecture	Süper Harvard Mimarisi
ADC	Analog-Digital Converter	Analog-Sayısal Dönüştürücü
ALERT	Automated Local Evaluation in Real Time	Gerçek Zamanlı Otomatikleştirilmiş Yerel Ölçüm
RF ID	Radio Frequency Identification	Radyo Frekansı Kimliği
BP	British Petroleum	İngiliz Petrol
GPS	Global Positioning System	Küresel Konumlandırma Sistemi
LPG	Liquefied Petroleum Gas	Sıvılaştırılmış Petrol Gazı
LEACH	Low-Energy Adaptive Clustering Hierarchy	Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi
TL-LEACH	Two Level LEACH	İki Katmanlı LEACH
HEED	Hybrid Energy Efficient Distributed Clustering	Hibrit Enerji Verimli Dağıtık Kümeleme
DWEHC	Distributed Weight-Based Energy-Efficient Hierarchical Clustering	Dağıtık Ağırlık Tabanlı Enerji Verimli Hiyerarşik Kümeleme
EECS	Energy Efficient Clustering Scheme	Enerji Verimli Kümeleme Şeması
HCR	Hierarchical Cluster-Based Routing	Hiyerarşik Kümeleme Tabanlı Yönlendirme
MR-LEACH	Multihop Routing with LEACH	LEACH ile Çok Atlamalı Yönlendirme
E-LEACH	Energy LEACH	Enerji LEACH
BEC	Based Energy Clustering	Enerji Tabanlı Kümeleme
LEACH-DCHS	LEACH-Deterministic Cluster-Head Selection	Deterministik Küme Başı Seçimiyle LEACH

LEACH-C	LEACH-Centralized	Merkezi LEACH
ID	Identification	Kimlik
BCDCP	Base-Station Controlled Dynamic Clustering Protocol	Baz İstasyonu Kontrollü Dinamik Kümeleme Protokolü
LCA	Linked Cluster Algorithm	Bağlı Küme Algoritması
FLOC	Fast Local Clustering Service	Hızlı Yerel Kümeleme Servisi
MS-LEACH	Multihop-Singlehop LEACH	Çok Atlamalı-Tek Atlamalı LEACH
ECHSSD	Efficient Cluster Head Selection Scheme for Data Aggregation in Wireless Sensor Network	Kablosuz Algılayıcı Ağlarda Veri Birleştirme için Verimli Küme Başı Seçim Şeması
GA	Genetic Algorithm	Genetik Algoritma
DSC	Disjoint Set Covers	Ayrık Küme Kapsamaları
DD	Direct Distance	Doğrudan Mesafe
C	Cluster	Küme
SD	Standard Deviation	Standart Sapma
E	Energy	Enerji
T	Transmission	İletim

ŞEKİL LİSTESİ

Şekil 1.1 - Ufak kablosuz algılayıcı aygıtları.....	1
Şekil 1.2 - Temel kablosuz algılayıcı ağ mimarisi.....	3
Şekil 1.3 - Temel kablosuz algılayıcı mimarisi.....	4
Şekil 1.4 - Algılama işleyişi.....	5
Şekil 1.5 - Temsili askeri uygulama örneği.....	10
Şekil 1.6 - Örnek orman yangını uygulaması.....	12
Şekil 1.7 - Temsili tarım uygulaması örneği.....	13
Şekil 1.8 - Felçli hasta rehabilitasyonunda kol-bacak hareket takibi.....	13
Şekil 2.1- Kümelere ayrılmış kablosuz algılayıcı ağı ve bileşenleri.....	16
Şekil 2.2 - LEACH'de kullanılan kümeleme örneği.....	22
Şekil 2.3 - LEACH protokolünün karşılaştırma grafiği. Algılayıcıların başlangıç enerjisi 0,5J'dür.....	25
Şekil 2.4- TL-LEACH iletişim hiyerarşisi. SN: Algılayıcı düğüm, CH _{xx} : İkincil küme başı, CH _x : Birincil küme başı, BS: Baz istasyonu.....	26
Şekil 2.5- TL-LEACH ve LEACH karşılaştırması.....	27
Şekil 2.6 - HCR ve LEACH karşılaştırma grafiği.....	31
Şekil 2.7 - MR-LEACH protokolünün hiyerarşik yapısı.....	32
Şekil 2.8 - LEACH-C ve LEACH karşılaştırması.....	35
Şekil 3.1 - Genetik algoritmanın işleyiş yapısı.....	43
Şekil 3.2 - Temsili rulet tekeri.....	46
Şekil 3.3 - Genetik algoritmanın detaylı işleyiş akış şeması.....	52
Şekil 4.1 - Genetik algoritma sonucu oluşturulmuş kümeler ve algılayıcıların görev dağılımı.....	58
Şekil 4.2 - Algılayıcı grupları için kromozomun tam sayı olarak temsil edilmesi.....	61
Şekil 4.3 - (a) Rastgele düzen (b) tek tip ızgara düzeni (c) küme ızgara düzeni.....	63
Şekil 4.4 - Algılayıcı ağın graf ve kromozom olarak temsili.....	65
Şekil 4.5 - Numaralı kromozomlarda çaprazlama örneği.....	66
Şekil 4.6 - Izgara şeklinde dağıtılan algılayıcıların iki bitlik olarak kodlanması.....	68
Şekil 4.7 - Önerilen yöntemin tur sayısı bazında LEACH ile karşılaştırılması.....	70

Şekil 5.1 - Çok atlamalı iletişimin temsili gösterimi.....	74
Şekil 5.2 - Geliştirilen simülâtörün ana ekran görüntüsü.....	75
Şekil 5.3 - GABEEC ve LEACH kullanılarak 0,5J başlangıç enerjisili algıalyıcı düğümlerle ağ yaşam süresi	76
Şekil 5.4 - GABEEC, HCR ve HCR-GA karşılaştırması.....	77
Şekil 5.5 - GABEEC ve ECHSSDA karşılaştırma grafiği.....	77
Şekil 5.6 - GABEEC ve M-GABEEC karşılaştırma grafiği	79

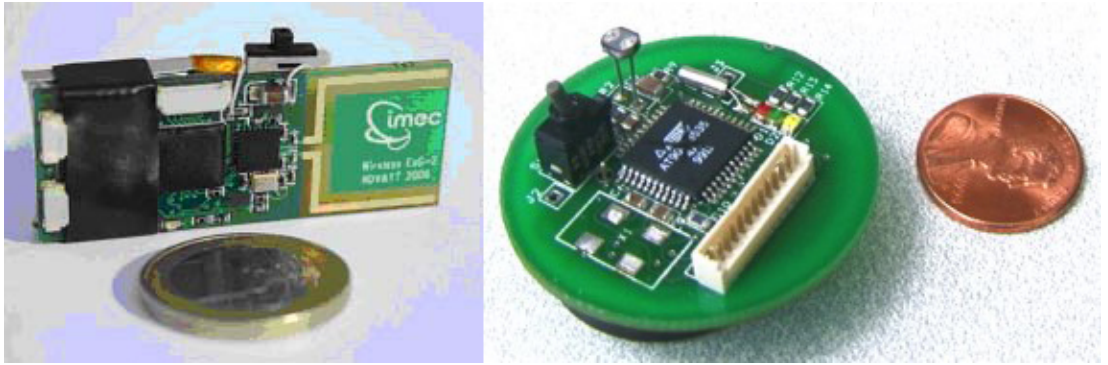
ÇİZELGE LİSTESİ

Çizelge 1.1 - Algılayıcı sınıflandırması ve örnekler	2
Çizelge 1.2 - Geleneksel ağlarla kablosuz algılayıcı ağların karşılaştırılması	8
Çizelge 2.1 - LEACH protokolünün tur sayısı bazında simülasyon sonuçları	25
Çizelge 3.1 - Kromozom için ikili kodlama örneği.....	44
Çizelge 3.2 - Kromozom için permutasyon kodlama örneği	44
Çizelge 3.3 - Kromozom için değer kodlama örneği	45
Çizelge 3.4 - Tek noktalı çaprazlama örneği	48
Çizelge 3.5 - Çift noktalı çaprazlama örneği	49
Çizelge 3.6 - Tek tip çaprazlama örneği	49
Çizelge 3.7 - Basit mutasyon örneği	50
Çizelge 3.8 - Rastgele seçilmiş komşu iki genin yer değiştirmesi.....	50
Çizelge 3.9 - Rastgele seçilmiş iki genin yer değiştirmesi.....	51
Çizelge 3.10 - Rastgele seçilen üç genin yer değiştirmesi.....	51
Çizelge 3.11 - Farklı araştırmacılar tarafından önerilen genetik algoritma parametre değerleri.....	53
Çizelge 4.1 - Kablosuz algılayıcı ağın ikili kodlamayla temsili	55
Çizelge 4.2 - Kablosuz algılayıcı ağlarda kromozomlara çaprazlama uygulanması .	56
Çizelge 4.3 - Mutasyon sonucu kromozomda iki bit değer değiştirmiştir	56
Çizelge 5.1 - GABEEC'in farklı başlangıç enerjileriyle tur sayısı temel alınarak LEACH'le kıyaslanması.....	75
Çizelge 5.2 - GABEEC ve HCR-GA karşılaştırmasının detaylı sonuçları. Sonuçlar iletim sayısı cinsinden alınmıştır.....	78

1. GİRİŞ

Bir kablosuz algılayıcı düğümü bulunduğu konumdaki bir durumu (sıcaklık, nem, hareket v.b.) algılayan, analiz edebilen ve bu veriyi belirli bir noktaya iletebilen akıllı, ucuz, düşük güç ile çalışan, çok fonksiyonlu bir elektronik ayardır [1] [2]. Örneğin insan vücudu çevredeki optik bilgiyi yakalayabilen (göz), akustik bilgiyi yakalayabilen (kulak) ve kokuları hissedebilen (burun) algılayıcılara sahiptir. Gözlemlenen nesneden bilgi almak için dokunmaya ihtiyaç yoktur. Teknik açıdan bakacak olursak bir algılayıcı fiziksel dünyadaki olayları ölçülebilen ve analiz edilebilen sayısal sinyallere dönüştüren ve bilgisayar sistemine aktarabilen bir cihazdır [3]. Ayrıca veriyi depolama ve veri üzerinde bir takım işlemler yapabilme yeteneğine de sahiptir. Fiziksel olarak da çok küçük boyutlarda olmaları (Şekil 1.1) kullanılabilirliklerini arttırmakta ve uygulama yelpazesini de genişletmektedir. Bunların yanında pille çalıştılarından dolayı enerji kısıtları vardır.

Birçok çeşit kablosuz algılayıcı vardır. Bunlar; sıcaklık, basınç, optik, akustik, mekanik, hareket, konum, elektromanyetik, kimyasal, nem ve radyasyon tipleridir (Çizelge 1.1).



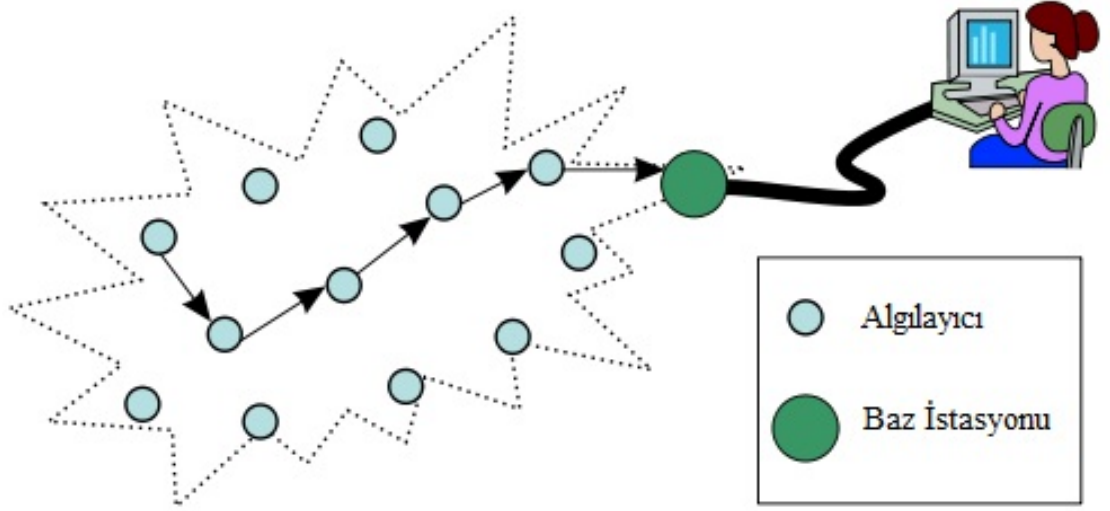
Şekil 1.1 - Ufak kablosuz algılayıcı aygıtları [4] [5]

Çizelge 1.1 - Algılayıcı sınıflandırması ve örnekler [3]

Tür	Örnekler
Sıcaklık	Isıldirenç (termistör), ısılıçift (thermocouple)
Basınç	Basınç ölçer, barometre, iyonlaşma ölçer
Optik	Fotodiyot, Fototransistör, kızılöresi algılayıcı
Akustik	Piezoelektrik tınlatici, mikrofon
Mekanik	Gerginlik ölçer, dokunma algılayıcısı
Hareket, titreme	İvme ölçer, jiroskop,
Akış	Rüzgar ölçer, yoğun hava akışı algılayıcısı
Pozisyon	GPS, kızılötesi tabanlı algılayıcılar, ses ötesi tabanlı algılayıcılar
Elektromanyetik	Hall etkisi algılayıcıları, manyetik ölçer
Kimyasal	pH algılayıcıları, elektrokimyasal algılayıcılar, kızılötesi gaz algılayıcılar
Nem	Kapatif ve dirençli algılayıcılar, nem ölçer
Radyasyon	İyonizasyon belirleyiciler, Geiger-Mueller sayaçları

1.1 Kablosuz Algılayıcı Ağlar

Küçük algılayıcı düğümleri, çok sayıda düğümü bir araya getirerek ortak çalışmalarını sağlayacak bir kablosuz algılayıcı ağı fikrini ortaya çıkartmıştır. Bir algılayıcı ağı çok sayıda algılayıcı düğümünden oluşur ve algılanacak alana yoğun bir şekilde dağıtılırlar (Şekil 1.2). Algılayıcı düğümlerin konumları önceden belirlenmek zorunda değildir. Erişilemez bölgelere veya afet yardım operasyonlarında rastgele dağıtılabirler. Bu da algılayıcı ağ protokollerinin ve algoritmalarının kendi kendini organize edebilen yetenekte olmasını gerektirmektedir. Algılayıcı ağların diğer bir önemli özelliği de algılayıcı düğümlerin ortak çalışmalarıdır. Bu anlatılan özellikler algılayıcı ağlar için çok çeşitli uygulama alanları sağlamaktadır. Sağlık, çevre ve güvenlik bu alanlardan bazılarıdır [1].

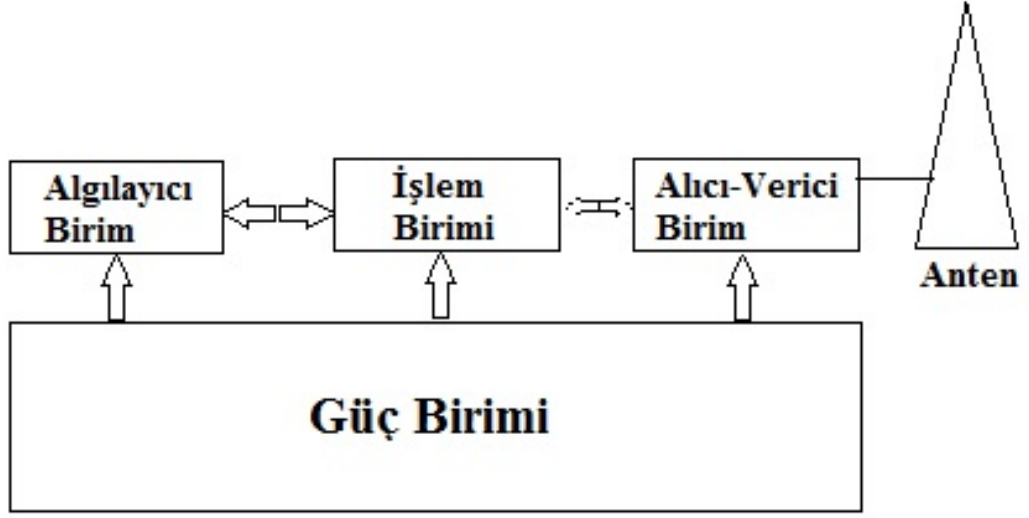


Şekil 1.2 - Temel kablosuz algılayıcı ağ mimarisi [6]

Birçok kablosuz algılayıcı ağda algılayıcılar çalıştırıcı (actuator) olarak da görev yapar. Çalıştırıcılar doğrudan fiziksel ortamı kontrol edebilirler. Örneğin bir algılayıcı sıcak su akışını kontrol eden bir vana olabilir, bir kapıyı veya pencereyi açıp kapayan bir motor olabilir ya da bir motora doldurulan yakıtın miktarını kontrol eden bir pompa olabilir. Böyle bir algılayıcı ve çalıştırıcı ağ işlemci cihazdan (kontrol kutusu) komutları alır ve algılayıcı için girdi sinyallerine dönüştürür. Böylece fiziksel süreçle etkileşim kurabilir [3].

1.2 Kablosuz Algılayıcı Düğümün Bileşenleri

Bir kablosuz algılayıcı düğümün temel bileşenleri algılayıcı birim, işlem birimi, alıcı-verici birim, anten ve güç birimi olarak söylenebilir. Şekil 1.3’de temsili olarak algılayıcı yapısı gösterilmiştir.

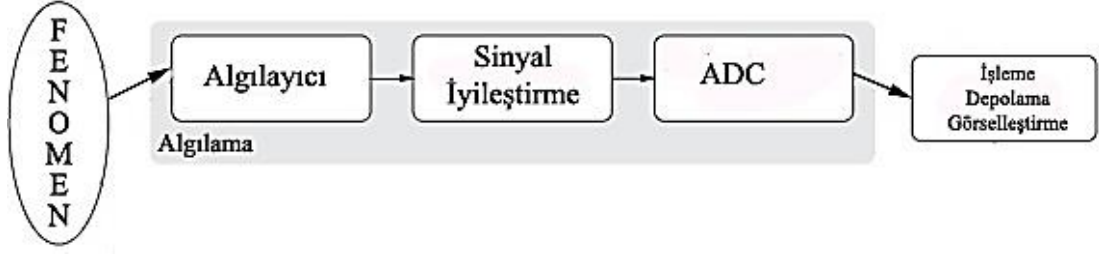


Şekil 1.3 - Temel kablosuz algılayıcı mimarisi

1.2.1 Algılama birimi

Algılama birimi düğümün ana bileşenidir. Uygulama bazlı bir algılayıcı içerir. Eğer uygulama sıcaklık takibi ise ısı algılayıcısı olacaktır. Uygulamaya göre algılayıcı birimi değişecektir [2].

Bir algılayıcı aygıtı fiziksel dünyadaki fenomeni (gözlemlenen nesne ya da olay) algıladığı zaman oluşan elektrik sinyalleri henüz iletmeye hazır değildir. İşlemeye geçmeden önce *sinyal iyileştirme* safhasından geçmelidir. Bu safhada algılanan sinyal analogtan sayısala dönüşüm için gerekli olan sinyal büyüklüğüne yükseltilmesi amacıyla sinyal güçlendirme işlemine tabi tutulur. Sinyal iyileştirme safhasında yapılan başka bir işlem de belirli bir frekans aralığında istenmeyen gürültüleri kaldırmak için filtreler uygulamaktır. Bu safhadan sonra analog sinyal bir ADC dönüştürücü (Analog-to-digital) kullanılarak sayısal sinyale dönüştürülür ve artık sinyal işleme, depolanma ve görselleştirme için hazır hale gelir (Şekil 1.4) [3].



Şekil 1.4 - Algılama işleyişi [3]

1.2.2 İşlemci birimi

İşlemci birimi programlanabilir bir mikro denetleyicidir. İşleme birimi ilgili aygıtta kontrol komutları göndererek düğümün bütün işlemlerini kontrol eder. Atmel, Siemens, Texas instruments önde gelen mikro denetleyici üreten firmalardır [2].

İşlemciler üç temel bilgisayar mimarisi kullanılarak tasarlanabilirler. Bunlar; Von Neumann, Harvard ve Super-Harvard'dır [3].

- **Von Neumann mimarisi:** Bu mimaride program komutlarının ve verilerin kullanabileceği tek bellek boşluğu vardır. İşlemciyle bellek arasında veri transferi için tek veri yolu sağlar. Bu da mimarinin işlem hızını oldukça düşürmektedir. [3]
- **Harvard mimarisi:** Bu mimari Von Neumann mimarisindeki yapıyı geliştirerek program komutları ve veriler ayrı bellek boşlukları sağlamıştır. Her bellek boşluğu ayrı bir veri yolu ile işlemciye bağlanır. Böylelikle program komutlarına ve verilere aynı anda erişilebilir. Bu iyileştirmeye ek olarak tek komutla çoklu veri işlemi, özel aritmetik işlem ve tersine bit adresleme gibi özelliklerde sunmaktadır. [3]
- **Super-Harvard mimarisi:** Bu mimari de Harvard mimarisinin geliştirilmesiyle ortaya çıkmıştır ve SHARC olarak bilinmektedir. Getirdiği yeniliklerden biri işlemci biriminin performansını arttıran dahili komut önbelleğidir. Bunun yanında mimari tam manasıyla kullanılmayan program belleğini veriler için geçici yer değiştirme alanı olarak kullanılmaktadır. [3]

1.2.3 Alıcı-verici birimi

Alıcı-verici, veri iletimi ve alımı görevlerini yerine getiren bir aygıttır [2]. Bu aygıt sayesinde algılayıcı düğüm ağa bağlanarak diğer düğümlerle haberleşebilir. Bu haberleşmeyi gerçekleştirebilmesi için de antene ihtiyacı vardır.

Birim, iletişim yapmak için radyo dalgası kullanır. Radyo dalgasının yanı sıra lazer ve kızıl ötesi kullanılabilir fakat bu iki tür radyo dalga kadar elverişli değildir. Lazer daha az enerji gerektirmesine rağmen atmosfer olaylarından çok etkilendiği ve iletişim için görüş alanı gerektirdiğinden ötürü tercih edilmez. Kızılötesi de lazer gibi anten gerektirmez ve az enerji gerektirir fakat iletim mesafesi çok düşük olduğundan tercih edilmez. Kablosuz algılayıcı ağlarda kullanılan radyo dalga frekansı 433 MHz – 2.4 GHz arasındadır [7].

1.2.4 Güç birimi

Bu birim algılayıcı düğüme enerji sağlayan birimdir. Üzerinde sınırlı enerjisi olan bir pil bulunur. Günümüzde bazı algılayıcı modellerinde güneş enerjisiyle yeniden doldurulabilir piller mevcuttur. Bu tarz pillere sahip algılayıcıların oluşturduğu ağ, güneşli açık alanlardaki uygulamalarda diğer uygulamalara nazaran çok daha uzun ömürlü olacaktır [2].

Bir kablosuz algılayıcı algılama yaparken, veri iletişimi yaparken ve veri işlerken enerji harcar. Bunların içinde en fazla enerji gerektiren işlem veri iletişimidir [7]. Bu nedenle kablosuz algılayıcı ağların araştırma alanlarının en önemlilerinden biri veri iletişimi yaparken enerjiyi en az şekilde kullanmanın yolunu bulmaktır.

1.3 Kablosuz Algılayıcı Ağların Üstünlükleri

Algılayıcı ağların temel hedefleri güvenilirlik, doğruluk, esneklik, maliyet verimliliği ve kolay kurulumdur. Algılayıcı ağların faydaları ve anahtar karakteristikleri şunlardır [7]:

- **Algılama doğruluğu:** Çok sayıda ve çeşitli algılayıcıların kullanımı az sayıda algılayıcı kullanımına kıyasla toplanan bilginin daha doğru ve kesin olma potansiyelini sağlar.
- **Alan kapsaması:** Verimli bir algılayıcı ağı genel ağ maliyetine ters etki etkisiz geniş çapta coğrafi alana yayılabilir.
- **Hata toleransı:** Aygıt fazlalığı ve bilgi fazlalığının tekil algılayıcılarda bir hata tolerans seviyesi sağlamak için değerlendirilebilir.
- **Bağlanırlık:** Birden fazla algılayıcı ağı taşıyıcı düğümler sayesinde internet gibi kablolu ağlar boyunca birbirine bağlanabilir. Ağların kümelenmesi her ağın belli bir alana veya olaya odaklanmasını ve sadece ilgili bilgiyi paylaşmasını sağlar.
- **Asgari insan etkileşimi:** En az insan etkileşimi sayesinde olası sistem kesintilerinin önüne geçilmiş olunur.
- **Zorlu ortamlarda çalışabilme:** Sağlam algılayıcı tasarımına ve yüksek seviyeli hata toleransına sahip algılayıcı düğümleri zorlu ortamlara dağıtılabilirler ve verimli bir algılayıcı ağı oluşturabilirler.
- **Dinamik algılayıcı zamanlaması:** Algılayıcı ağı bazı zamanlama şemalarının uygulanmasıyla veri iletiminde algılayıcı düğümler için öncelik ayarlayabilir durumundadır.

1.4 Kablosuz Algılayıcı Ağların Geleneksel Ağlarla Karşılaştırılması

Kablosuz algılayıcı ağlar geleneksel ağ sistemlerinden birkaç durumda oldukça farklıdır. Klasik bilgisayar ve iletişim ağları sabit sayıda düğüme ve sabit bir

topolojiye sahiptir. Ağın veri hızını maksimize etmek için tasarlanırlar. Buna karşılık kablosuz algılayıcı ağların amacı bilinmeyen ortamlarda birçok bilinmeyeni algılamak, belirlemek ve tanımlamaktır. Düğümler aynı tipte veya farklı tipte algılayıcılar olabilir. Her düğümün alıcı-vericisi ve işlemcisi vardır.

Algılayıcı düğümlerin konuşlanması da benzersizdir ve uygulamadan uygulamaya farklılık gösterir. Algılayıcılar bir fabrikanın etrafına veya binanın içine belirli mesafelerde el ile yerleştirilebilir. Bir araç veya uçak vasıtasıyla gerekli bölgelere rastgele serpiştirilebilirler. Böylelikle tamamen rastgele tasarsız (ad-hoc) bir topoloji oluştururlar. Bu konuşlanmalarda algılayıcı düğümlerin sayısı onlarca, yüzlerce ve hatta binlerce olabilir [9].

Kablosuz algılayıcı ağlarla geleneksel ağ sistemlerinin karşılaştırılması Çizelge 1.2'de özetlenmiştir.

Çizelge 1.2 - Geleneksel ağlarla kablosuz algılayıcı ağların karşılaştırılması [3]

Geleneksel ağlar	Kablosuz algılayıcı ağlar
Genel amaç için tasarlanır, birçok uygulamaya hizmet verir	Tek bir amaç için tasarlanır, özel bir uygulamaya hizmet verir
Öncelikli tasarım gereksinimleri ağ performansı ve gecikmelerdir. Enerji konusu öncelikli bir gereksinim değildir.	Ana tasarım gereksinimi enerjidir.
Ağlar planlara göre tasarlanır ve yönetilir.	Konuşlanma, ağ altyapısı ve kaynak kullanımını çoğunlukla tasarsızdır (plansız).
Cihazlar ve ağ kontrol edilebilir ve sakin ortamlarda çalışır.	Algılayıcı ağlar çoğunlukla çeşit koşullar bulunun ortamlarda çalışırlar.
Bakım ve onarımları basit ve alışılmış yöntemlerdir ve ağa erişim çoğunlukla çok kolaydır.	Algılayıcı düğümlere fiziksel olarak erişmek çok zor hatta bazı durumlarda imkansızdır.
Bileşenler arızalandığında bakım ve onarımla giderilebilir.	Bileşen arızalanması daha olasıdır ve ağın tasarımıyla giderilebilir.

1.5 Kablosuz Algılayıcı Ağların Uygulama Alanları

Algılayıcı düğümleri sürekli algılama, olay tespiti, konum algılaması ve tetikleyicilerin yerel kontrolü için kullanılabilir. Bu düğümlerin mikro-algılama ve kablosuz bağlantı özellikleri birçok yeni uygulama alanlarının ortaya çıkmasına olanak sağlamıştır. Kablosuz algılayıcı ağların uygulamaları genel olarak askeri, çevresel, sağlık, ev ve diğer ticari alanlar olarak sınıflandırılmaktadır [1].

1.5.1 Askeri uygulamalar

Askeri uygulamalar kablosuz algılayıcı ağların konseptiyle çok yakından alakalıdır. Bu alandaki kullanım amaçları düşman takibi, savaş alanı gözetlemesi ve hedef sınıflandırma olarak gruplandırılabilir [10] (Şekil 1.5).

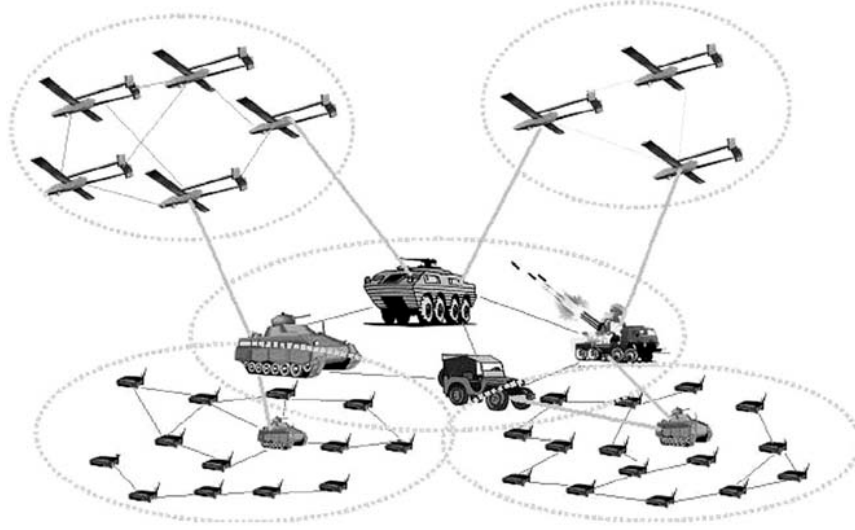
Sınıflandırma algoritmalarında girdi verisi sismik ve akustik sinyal algılamasından gelmektedir [11] [12]. Örneğin mayınlar çok tehlikeli olarak görülebilir ve gelecekte işe yaramaz hale gelebilirler. Bunun yerine düşman birimlerin izinsiz girişlerini saptamak için binlerce dağıtılmış algılayıcıdan faydalanılabilir. Bununla ilgili bir çalışma Virginia üniversitesinde yapılmıştır [13].

[12]'deki diğer bir uygulamada kaçış-kovalama (pursuit-evasion) oyununda çoklu araç takibi yapılmaktadır. Kaçanlar ve kovalayanlar olarak iki rakip takım vardır. Üçüncü parça da kovalayanların rakiplerinin yerini bulmakta yardımcı olarak kullanılan algılayıcı ağıdır. Algılayıcı ağ kovalayanları düşman birimlerinin yaklaşık konumu ve hareketleri ile bilgilendirir.

Savaş alanı gözlemlenmesinde kritik araziler, yaklaşım rotaları, yollar ve geçitler algılayıcı ağlarla sarılabilir ve düşman kuvvetlerinin aktiviteleri yakından takip edilebilir [1].

Komutanlar sürekli savaş alanındaki dost birliklerin durumunu, ekipmanın durumunu ve mühimmat durumunu algılayıcı ağlar kullanarak izleyebilir. Her birliğe, taşıta, ekipmana ve kritik mühimmata durumlarını rapor edecek ufak bir algılayıcı

cı yerleştirilir. Bu raporlar taşıyıcı algılayıcılarda toplanır ve birlik liderine gönderilir [1].



Şekil 1.5 - Temsili askeri uygulama örneği [15]

1.5.2 Çevresel gözleme uygulamaları

Kablosuz algılayıcı ağlarda çevresel gözlemler kapalı alan gözleme ve acil durum servisleri, açık alan gözleme ve ekolojik uygulamalar, açık alan gözleme ve tarımsal uygulamalar olarak sınıflandırılabilir [10].

a) Kapalı alan gözleme ve acil durum servisleri

Berkeley California Üniversitesi Elektronik Mühendisliği ve Bilgisayar Bilimleri Fakültesi araştırmacıları ve lisansüstü öğrencileri bilişim teknolojileri merkezine ışık ve sıcaklığı gözlemek için kibrit kutusu büyüklüğünde elli adet “Smartdust” adında kablosuz algılayıcı yerleştirmiştir [16]. Kapalı alan gözlemeyi optimuma getirmek için sıcaklığı, ışığı, kapı ve pencerelerin durumunu, binanın hava kirliliği durumunu algılayabilme kabiliyetinden yararlanılabilir. Bunun yanında binaları ısıtmak veya soğutmak için çok büyük enerji israfı oluşmaktadır. Algılayıcılar sayesinde ısıtıcıların, fanların ve diğer ekipmanların gerektiği zaman gerektiği kadar

kullanılması sağlanıp büyük enerji tasarrufu sağlanabilir. Bu tarz bir uygulama Washington'da Pasifik Güneybatı Ulusal Laboratuvarında yapılmıştır [17].

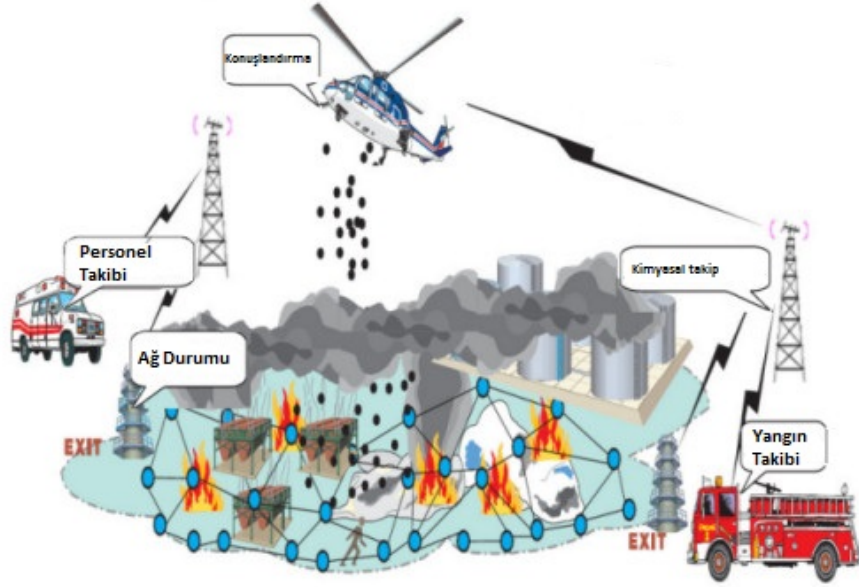
Başka kapalı alan uygulamaları da yangının zararını azaltmayı amaçlayan uygulamalardır. Bu gibi durumlarda algılayıcı ağın görevi apartman sakinlerine rehberlik edip çıkış için en güvenli yolu göstermektir. Bunun için bina içinde ışıklandırma sistemi kullanılabilir.

b) Açık alan gözleme ve ekolojik uygulamalar

Açık alan gözlemede algılayıcı ağ uygulamaları geniş bir yelpazeye sahiptir [10]. Bu alanla ilgili en temsili örnek Büyük Ördek Adası'na kurulan 32 düğümlü algılayıcı ağıdır [18]. Bu ağ yaşam alanı (habitat) gözleme için kullanılmıştır. Kullanılan algılayıcılar sıcaklığı, basıncı ve nemi algılayabiliyorlardı. Bunların yanında ışıdırengler (fotorezistör) de kullanılmıştır. Uygulamanın amacı bir kuşun doğal yaşam alanını ve iklimsel değışikliklerde nasıl bir tutum sergilediğini gözlemlemektir. Bu nedenle bazı düğümler kuşun yuvasına diğerleri de çevreye yerleştirilmiştir. Ağ haberleşmesinde algılayıcılar topladıkları verileri bir ağ geçidine gönderiyor ve ağ geçitleri de baz istasyonuna gönderiyordu. Bu uygulama heterojen ve çok katmanlı ağ kullanan bir gözleme örneğidir.

Eski uygulamalardan biri de ALERT (Automated Local Evaluation in Real Time) dir [19]. Bu uygulama California ve Arizona'da yağmurları gözlemleyerek sel tespiti yapmak için geliştirilmiştir. Sistemde sıcaklık, su seviyesi ve rüzgar algılayıcıları vardır.

Algılayıcı ağlar, orman yangınlarını engellemek için de kullanılabilir [1]. Eğer algılayıcılar ormana stratejik olarak yoğun şekilde rastgele dağıtılsa algılayıcı düğümler yangının başladığı kesin noktayı son kullanıcıya anında ileterek yangının büyümeden, hemen başında engellenmesini sağlayabilir (Şekil 1.6). Ormanlık araziye milyonlarca algılayıcı konumlandırılabilir. Bunun yanında algılayıcılara güneş pili gibi farklı güç kaynakları da eklenebilir [20]. Çünkü bu gibi alanlarda algılayıcılar aylarca hatta yıllarca başıboş kalacaktır. Enerjilerini yenileyebilmeleri için bu gibi çözümler kaçınılmazdır.



Şekil 1.6 - Örnek orman yangını uygulaması [21]

c) Açık alan gözlemeleme ve tarımsal uygulamalar

Ziraatın verimliliğinin artırılması ve gelişmesini sağlamak için algılayıcı ağlar tarım sektöründe de kullanılmıştır [10] (Şekil 1.7). Bunların bir örneği üzüm bağlarında yapılan uygulamadır [22]. 121.000 metrekarelik alana nemi, rüzgarı, suyu, toprak durumunu ve hava sıcaklığını ölçen algılayıcılar yerleştirilmiştir. Amaçlardan biri sulama yönetimidir. Topraktaki nem, havadaki nem ve hava durumu tahmini sayesinde çok daha verimli ve ekonomik sulama yapılabilir. Bunun dışında ağda donma tespiti ve uyarı sistemi, böcek ilacı uygulaması gibi farklı işlemler de yapılabilmektedir.

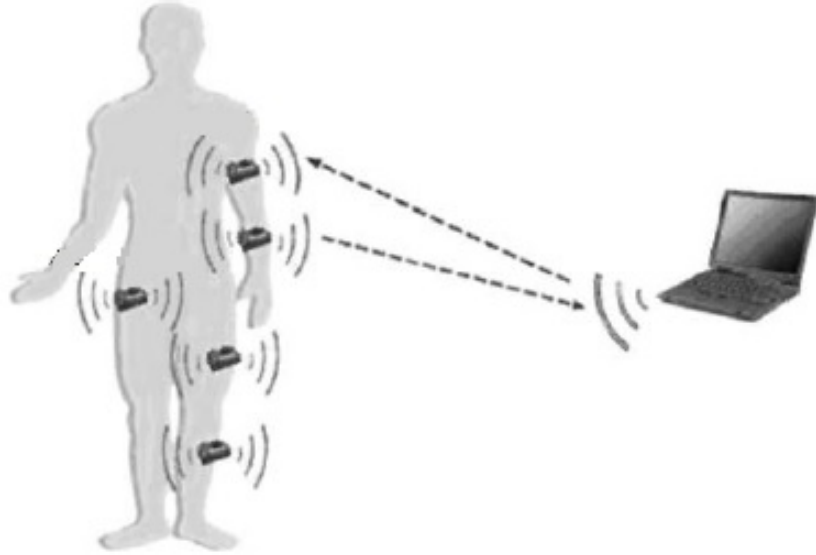
1.5.1 Sağlık uygulamaları

Kablosuz algılayıcılar sağlık sektöründe de faydalı işlerde kullanılabilir. Alzheimer hastalığına götüren kognitif bozuklukların başlangıç safhalarında gözlemlenebilmesi ve kontrol edilebilmesi kablosuz algılayıcılarla mümkün olabilmektedir [10]. Düğümler, hastanın yaptığı eylemleri kaydeder ve gerekli kişiler hastanın

ne yaptığını, nasıl durumda olduğunu uzaktan takip edebilir [1] [23]. Şekil 1.8’te sağlık uygulaması için temsili bir resim gösterilmiştir.



Şekil 1.7 - Temsili tarım uygulaması örneği [15]



Şekil 1.8 - Felçli hasta rehabilitasyonunda kol-bacak hareket takibi [15]

Bunların dışında hastanedeki doktorların ve hastaların takibi de yapılabilir. Her hastanın üzerinde ufak ve hafif bir algılayıcı bulunur. Her algılayıcı düğümün kendine has bir görevi vardır. Örneğin bir düğüm kalp ritmini ölçerken, diğeri kan basıncını ölçebilir. Doktorlar da bir algılayıcı düğüm taşıyabilir. Böylelikle hastane içinde doktorların nerede olduğu kolaylıkla ortaya çıkar [1].

1.5.2 Endüstriyel uygulamalar

Envanter kontrolü, büyük firmalar için önemli bir problemdir. Malların kontrolü zor bir hale gelebilir. Özellikle dünya geneline yayılmış şirketler için, bu problem daha geniş kapsamlı olmaktadır. Bu problemle baş etmek ve mal takibini sağlayabilmek için RF ID ve kablosuz algılayıcılar kullanılmaktadır [10]. BP (British Petroleum) firması bu alanda araştırmalar yapmıştır ve “akıllı vekiller” uygulamasını geliştirmiştir [24]. Uygulama ambarlarla ve varillerin depo yönetimiyle ilgilidir. Algılayıcılar varillere takılmıştır ve yakındaki nesnelerin yerini saptayabilmektedir. Bu nesnelerin içeriğini belirleyip uygunsuz bir durum halinde alarm verebilmektedir. Bu uygulama güvenliği arttırmakta ve ürün kalitesini garanti etmektedir. BP ve Accenture Teknoloji firmaları kablosuz algılayıcı ağları filo yönetiminde de kullanmışlardır [10]. Kamyonların, vagonların ve bunlarla taşınan eşyaların takibi algılayıcılar ve GPS sistemi sayesinde mümkün olabilmektedir. Böylelikle telemetri (uzaktan ölçüm) ve kablosuz algılama, akıllı nesnelere ve araçlar oluşturmak için bir araya getirilebilir.

Endüstriyel uygulamaların başka bir tanesini de Intel firması gerçekleştirmiştir [10]. Bu uygulamada yarı iletken fabrika ekipmanlarının durum gözlemlemesi yapılmıştır. Algılayıcılar titreşimleri algılamıştır. Parçaların titreşimleri analiz edilerek hangi parçaların onarıma veya değişime ihtiyacı olduğu ortaya çıkmaktadır.

Taşıma ve dağıtım sistemleri de kablosuz algılayıcılar için başka bir uygulama alanıdır. Yine BP firmasının yaptığı bir uygulamada, LPG gazının müşterilere dağıtımının yönetilmesinde kablosuz algılayıcı teknolojisini kullanılmıştır [24]. Her müşteri tankının içeriği gözlemlenir. Böylelikle şirketin tedarik departmanı geriye kalan miktarı bilebilmektedir. Bu, tedarik araçlarının yolculuklarının programlanma-

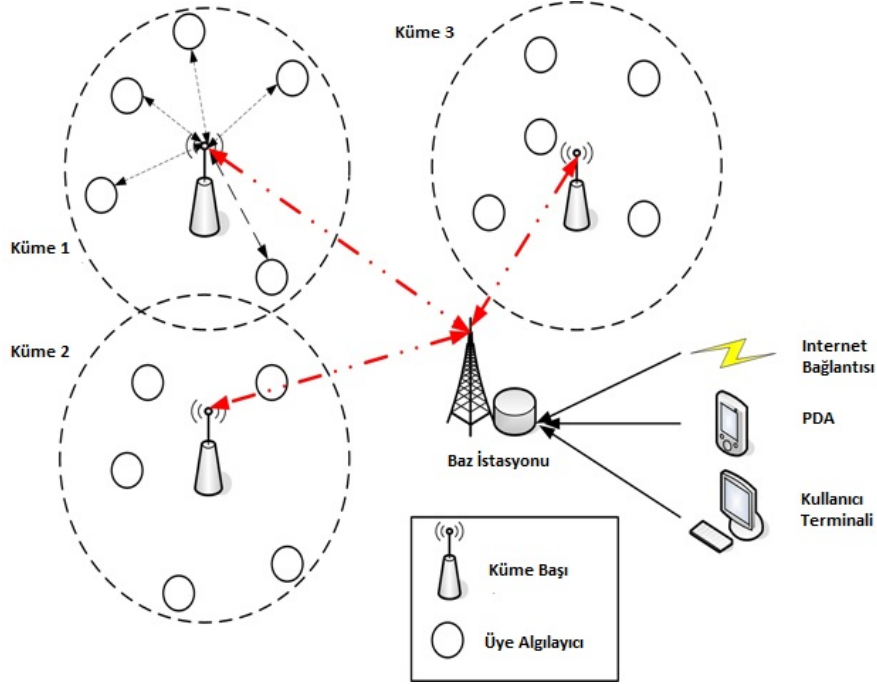
sında kullanışlı olmaktadır ve sonuç olarak taşıma sisteminin verimliliğini arttırmaktadır.

Başka bir endüstriyel uygulama da Helsinki Teknoloji Üniversitesi'nde gerçekleştirilmiştir [25]. Bu çalışmada algılayıcılar kağıt üretim sürecinde kullanılmıştır. Algılayıcılar kağıt kurulum safhasında kullanılan ısı rulolarına yerleştirilmiştir ve görevleri ısı rulolarını kontrol etmek için sıcaklığı ölçmektir.

Bir diğer uygulama da elektrik enerji sistemleri için geliştirilmiştir [26]. Elektrik ekonomisi sürekli maksimum elektrik tüketim değerleriyle ilgilenir. Bu değer ekonomik nedenlerden ötürü olabildiğince düşük tutulmalıdır. Elektrik talebindeki uç değerler elektronik aygıtlara kablosuz algılayıcılar yerleştirilerek azaltılabilir.

2. KABLOSUZ ALGILAYICI AĞLARDA KÜMELEME

Algılayıcı ağların ölçeklenebilirliğini sağlamak, enerji tasarrufu sağlayarak ağ yaşam süresini uzatmak için algılayıcıların grup olarak ortak çalışması kümeleme (clustering) olarak adlandırılmaktadır ve araştırmacılar tarafından sıklıkla kullanılmaktadır [27]. Kümelere ayrılmış bir kablosuz algılayıcı ağın bileşenleri şunlardır (Şekil 2.1) [28]:



Şekil 2.1- Kümelere ayrılmış kablosuz algılayıcı ağı ve bileşenleri [29]

- **Algılayıcı düğüm:** Bir algılayıcı düğüm, kablosuz algılayıcı ağın temel bileşenidir. Algılayıcı düğümler algılama, veri depolama, yönlendirme ve veri işleme gibi işlevleri yerine getirirler.
- **Kümelere:** Kümelere, kablosuz algılayıcı ağların hiyerarşik birimleridir. Büyük algılayıcı ağları baz istasyonu ve küme başıyla iletişim gibi görevleri basitleştirmek için kümelere bölünmek durumundadır.

- **Küme başları:** Küme başları kümelerin liderleridir. Küme başları çoğunlukla küme içindeki aktiviteleri yönetirler. Bu aktiviteler veri birleştirme, küme içindeki iletişimi organize etme ve baz istasyonu ile iletişim olarak sıralanabilir.
- **Baz istasyonu:** Baz istasyonu algılayıcı ağda elde edilen verilerin toplandığı noktadır. Son kullanıcıyla algılayıcı ağ arasındaki iletişim bağına sağlar.
- **Son kullanıcı:** Son kullanıcı kablosuz algılayıcı ağın elde ettiği veriye erişen ve bu veriyi çeşitli uygulamalarda kullanan kişidir.

Bir küme başı, kümedeki algılayıcılar tarafından seçilebileceği gibi ağ tasarımcısı tarafından da önceden belirlenebilir. Küme başı, algılayıcılardan herhangi biri olabileceği gibi kaynak bakımından daha verimli olan bir düğüm de olabilir. Küme üyelikleri sabit olabileceği gibi değişken de olabilir. Küme başı bir alt ağ katmanı oluşturarak arada köprü görevi görebileceği gibi aldığı veriyi doğrudan ilgili yere (örn: baz istasyonu) iletebilir [27].

Kümeleme yönteminde göz önüne alınması gereken birkaç kısıt vardır. Bu kısıtları şu şekilde sıralayabiliriz [28] [29]:

- **Limitli enerji:** Kablosuz algılayıcı düğümler pille çalıştıkları ve çoğunlukla bu piller yeniden doldurulamayacağı veya değiştirilemeyeceği için limitli enerjileri vardır. Bu enerjinin verimli kullanılması, kablosuz algılayıcı ağın yaşam süresi ve etkin çalışması için hayati önem arz etmektedir.
- **Ağ yaşam süresi:** Düğümlerdeki enerji limiti ağ yaşam süresini de sınırlamaktadır. Kümeleme yöntemleri enerjiyi verimli kullanma adına oldukça etkilidirler ve ağ yaşam süresini uzatabilirler.
- **Limitli kabiliyetler:** Fiziksel büyüklüğünün ufak olması ve limitli enerjiye sahip olması algılayıcı düğümün veri işleme ve iletişim gibi kabiliyetlerini de sınırlamaktadır. Kümeleme yöntemi kullanılarak algılayıcı düğümlerin ortak çalışıp kaynaklarını aralarında paylaşmalarını sağlayıp bu kısıtların önüne geçilebilir.

- **Uygulama bağımlılığı:** Algılayıcı ağda kümeleme yöntemi uygulanacakken üzerinde çalışılan uygulamaya göre bir küme tasarımı yapılmasına dikkat edilmelidir.

2.1 Farklı Uygulamaların Ağ Mimarisine Etkisi

Birçok ağ mimarisi algılayıcı düğümlerinin konumunun sabit olduğunu varsayar. Eğer düğümler gezici (mobil) olsaydı kümeleme işlemi çok zorlaşır. Çünkü üye düğümler sık sık dinamik olarak değişecektir ve kümeler sık aralıklarla değişime uğrayacaktır. Diğer taraftan bir algılayıcı tarafından gözlemlenen olaylar uygulamaya göre aralıklı olabileceği gibi sürekli devam eden olay da olabilir. Örneğin takip uygulaması veya hedef belirleme uygulaması. Bu uygulamalarda takip edilecek veya belirlenecek şey dinamik bir yapıya sahiptir. Yani sürekli izlenmelidir. Öte yandan yangın için orman gözlemlemesi aralıklı olaylara bir örnektir. Aralıklı olay gözlemlemesi tepkili (duyarlı) durumda çalışır. Sadece raporlama sırasında ağ trafiği oluşur. Sürekli olaylarda ise birçok uygulamada periyodik raporlama gerekmektedir ve bu genellikle çok sık aralıklarla meydana gelir. Sonuç olarak da merkeze giden iletimde trafik yoğunlaşır. Sürekli olaylar çoğunlukla kümeleri sabit kılar. Bu da küme başlarına üye düğümlere kıyasla daha fazla yük yükler. Bu nedenle küme başlarının değiştirilmesine ihtiyaç duyulabilir [27].

Diğer bir düşünce düğümlerin topolojik konuşlanmasıdır. Bu işlem uygulama bağımlıdır ve ağ kümelemesinin hedefini ve ihtiyacını etkiler. Konuşlanma belirli yerlere olabileceği gibi rastgele de organize edilebilir. Belirli yer konuşlanmasında algılayıcılar el ile yerleştirilir ve veriler de önceden belirlenmiş yollardan yönlendirilir. Rastgele konuşlanan sistemlerde algılayıcı düğümleri rastgele dağıtılır ve plansız bir yapı oluşturur [30] [31]. Bu alt yapıda baz istasyonunun veya küme başının konumu enerji verimliliği ve performans açısından çok önemlidir. Tek tip algılayıcıları olan bir ağda küme başları konuşlanmış düğümlerden seçilir. Bu durumda küme başları dikkatlice görevlendirilmelidir. Örneğin algılama görevlerine dahil edilmemelidirler. Bu şekilde enerjileri çabuk tükenmez. Buna ek olarak iletişim mesafesi ve

küme başlarının baz istasyonuna olan yakınlığı kısıt olabilir ve bu nedenle göz önünde bulundurulması gerekmektedir. Algılayıcının iletişim mesafesi kısıtlıdır. Uzun mesafeye iletim yapıldığında aşırı enerji tüketecektir. Bu nedenle küme başları arasında çok atlamalı (multihop) iletişim kurulması enerji verimliliği açısından etkili olacaktır [27].

2.2 Küme Tasarımında Göz Önüne Alınması Gereken Nitelikler

Kablosuz algılayıcı ağı kümelere bölerken ağ tasarımcılarının bazı kriterleri göz önünde bulundurmaları gerekmektedir. Bu önemli nitelikler aşağıda verilmiştir [29]:

- **Kümeleme maliyeti:** Kümeleme işlemi algılayıcı ağ topolojisinin organizasyonunda çok önemli rol oynamaktadır. Bununla beraber kümelemenin yapılması ve sürdürülmesinde ihtiyaç duyulan iletişim ve işlem gibi gerekli kaynaklar vardır. Bu kaynaklar veri iletimi veya algılama işlemlerinde kullanılmamaktadırlar.
- **Küme başlarının ve kümelerin seçimi:** Kümeleme yöntemi kablosuz algılayıcı ağlar için büyük faydalar sağlamaktadır. Bu faydalardan en önemlisi enerji tasarrufudur. Düğümlerin uzun mesafelere iletim yapmasının önüne geçerek enerjilerinin daha geç tükenmesini sağlamaktadır. Bununla beraber tasarımcılar belirli bir uygulama için tasarım yaparken kümelerin ağdaki düzenini dikkatlice irdelemelidir. Uygulamanın tipine göre kümedeki düğümlerin sayısı ağın işlevinde önemli rol oynayabilir.
- **Veri birleştirilmesi:** Kablosuz algılayıcı ağların önemli avantajlarından biri ağda veri birleştirme işlemi yapabilmesidir. Çok sayıda algılayıcı düğüm içeren ağlarda genellikle birçok düğüm benzer bilgileri algırlar. Veri birleştirme sayesinde algılanan veri ve faydalı veri farkı ortaya çıkmaktadır ve bu işlem birçok algılayıcı ağ şemasında olmazsa olmaz bir özelliktir [32]. Bu sayede ağ üzerinde taşınan veri miktarı en aza indirilmiş olur. Bu işlemde harcanan enerji miktarı veri iletiminde harcanan enerji miktarından çok daha dü-

şüktür. Kümeleme yaklaşımı seçilirken bu özellik dikkatlice göz önüne alınmalıdır.

- **Onarım mekanizması:** Kablosuz algılayıcı ağların doğası gereği düğümlerin yer değişimi, bozulması ve kapsama dışında kalması olasıdır. Bu durumların hepsi bağlantı hatası oluşturabilir. Kümeleme şemasında bağlantı düzeltme ve güvenilir veri iletişimi mekanizması olması kaçınılmazdır.
- **Servis kalitesi:** Kablosuz ağda servis kalitesini yakalayabilmek için yukarıda bahsedilen niteliklere çok önem gösterilmesi gerekmektedir. Bu nitelikler uygulama bağımlıdır. Bu sebeple yapılacak uygulamaya göre kümelerin oluşturulması servis kalitesini arttırmada önemli rol oynar.

2.3 Kümeleme Süreci

Kümelemede iki temel adım vardır: Küme başı seçimi ve küme düzenleme. Küme başı seçimi üç tipte sınıflanabilir: Baz istasyonu tarafından, algılayıcı düğümler tarafından veya baz istasyonu ve düğümlerden elde edilen bilgiler sayesinde karma bir şekilde [28].

2.3.1 Küme başı seçimi

Bu işlemde göz önüne alınması gereken durumlar şu şekildedir:

- Küme başları ve baz istasyonu arasındaki mesafe çok uzak olmamalıdır. Aksi takdirde uzun mesafelerde veri iletimi çok enerji harcatacağından algılayıcının erken ölmesine sebep olacaktır.
- Küme başlarının tek tip (uniform) şekilde konuşlanmaları küme başı olmayan diğer algılayıcılar arasındaki mesafenin uzak olmasını engelleyecektir. Fakat bu işlemi rastgele konuşlandırılan algılayıcı ağlarda yapmak pek mümkün değildir.

- Küme başları kablosuz algılayıcı ağlarda veri birleştirme, uzak mesafeye veri iletimi gibi fazladan işler yapmaktadırlar. Bu nedenle enerjileri diğer düğümlere nazaran daha erken bitecektir. Ağ iletişim sürecinde küme başının değiştirilmesi kümeleme yönteminde göz ardı edilmemesi gereken başka bir konudur.
- Küme başı seçilecek olan algılayıcı düğümün geriye kalan enerjisi yüksek olmalıdır.

2.3.2 Algılayıcı düğümün küme başını seçmesi

- Bir düğümler küme başı arasındaki mesafe düğümün hangi küme başının kümesine dahil olacağı konusunda en etkili parametredir. Bu durumda düğüm veri gönderirken en az enerjiyi harcamak için kendisine en yakın olan küme başını seçmelidir.
- Uygulamanın tipine göre kümenin büyüklüğü de bir düğümün kümeye dahil olup olmayacağı konusunda parametre olabilir.

2.4 Kümeleme Algoritmaları

Literatürde birçok farklı tarzda kümeleme algoritması mevcuttur. Bu algoritmalar ve özellikleri bu bölüm içerisinde anlatılmaktadır.

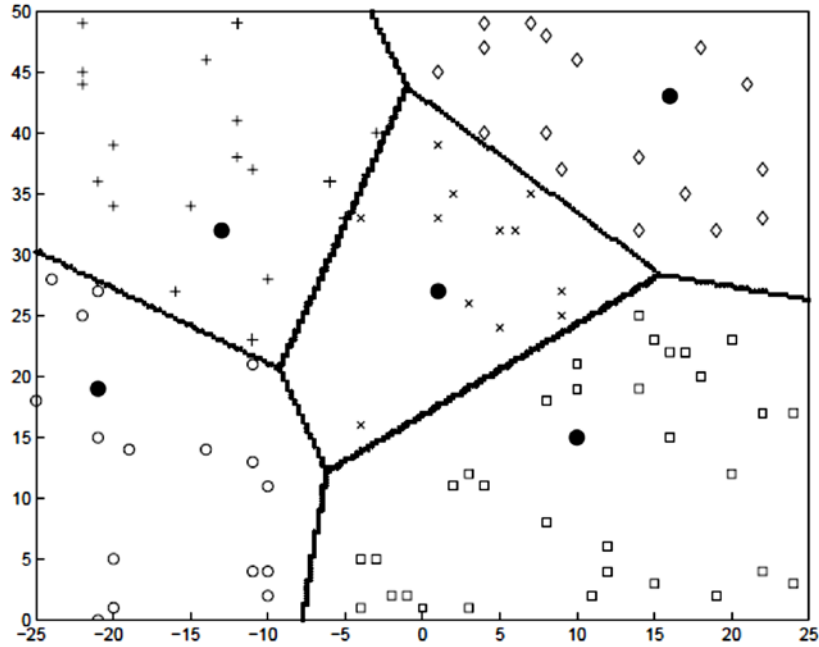
2.4.1 LEACH (Low-Energy Adaptive Clustering Hierarchy)

LEACH kümeleme algoritmalarının en eskilerinden ve en bilindik olanlarından [33]. Algılayıcılar kendi kendilerini organize edip kümelere bölünürler. Bunu yaparken de küme başlarına olan mesafelerini değerlendirirler. Bu mesafeyi de küme başlarından aldıkları yayın sinyalinin gücüne bakarak anlarlar. Küme başları da baz

istasyonuna yönlendirici olarak kullanılırlar. Bütün veri işleme işlemleri kümelerde yapılır. Bu çalışmada şu varsayımlar yapılmıştır:

- Baz istasyonu sabittir ve ağdan uzak bir mesafeye konumlanmıştır.
- Ağdaki bütün algılayıcılar tek tiptir ve sınırlı enerjileri vardır.

LEACH klasik kümeleme yöntemlerinden farklı olarak uyarlanabilir kümeler kullanır ve küme başlarını rotasyonla değiştirir. Böylelikle ağın enerji gereksinimlerini algılayıcılar arasında dağıtır. Küme başları değiştirilince de yeni kümeler oluşturulur. Şekil 2.2’de bir kümeleme örneği gösterilmiştir. Bunun yanında LEACH algoritmasında yerel hesaplama yapılır. Yani küme başları üye düğümlerden aldıkları veriyi direk baz istasyonuna göndermezler. Bunun yerine bütün üye düğümlerden verileri aldıktan sonra tek paket halinde sıkıştırıp gönderirler. Bu da enerji tüketimini oldukça azaltmaktadır. Çünkü veriler üzerinde işlem yapmak iletişimden çok daha az enerji gerektirir.



Şekil 2.2 - LEACH'de kullanılan kümeleme örneği [33]

Bu çalışmada basit bir radyo modeli kullanılmıştır. Radyo vericisini veya alıcısını çalıştırmak için $E_{elec} = 50$ nJ/bit enerji harcar. İletim güçlendiricisi için de $\epsilon_{amp} = 100$ pJ/bit/m² enerji harcar. Böylece k bitlik mesajı d mesafesine göndermek için harcanan enerji Denklem (1)'de gösterilmektedir.

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2 \quad (4.1)$$

Mesajı almak için de harcanan enerji Denklem (2)'de gösterilmektedir.

$$E_{Rx}(k) = E_{elec} * k \quad (4.2)$$

LEACH algoritmasında işlem süreci turlara (rounds) ayrılmıştır ve her tur iki safhadan oluşur. Birinci safha kurulum safhası (set-up phase), ikinci safha da iletişim safhasıdır (steady-state phase). Kurulum safhasında her düğüm küme başı olup olmayacağına karar verir. Bu karar ağ için önceden belirlenmiş küme başı oranı ve o ana kadar algılayıcı düğümün kaç kere küme başı olduğu baz alınarak verilir. n düğümü 0 ve 1 arasında rastgele bir sayı seçilerek küme başı olup olmayacağına karar verir. Eğer seçtiği sayı eşik değerinden az ise düğüm o tur için küme başı olur. Eşik değeri, $T(n)$, Denklem (2.3)'de aşağıdaki gibi hesaplanır:

$$T(n) = \begin{cases} \frac{P}{1 - P * \left(r \bmod \frac{1}{P}\right)}, & n \in G \text{ ise} \\ 0, & \text{diğer} \end{cases} \quad (4.3)$$

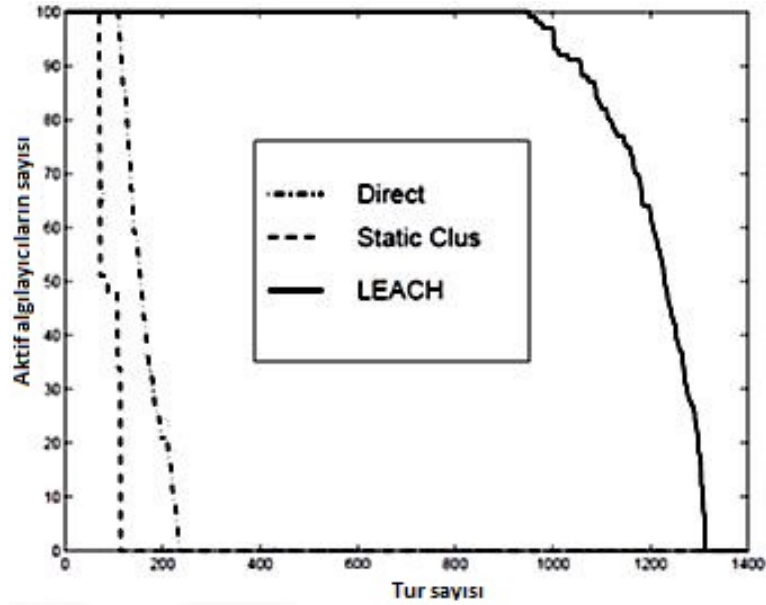
P , önceden belirlenen küme başı oranıdır. Bu çalışmada $P = 0,05$ olarak alınmıştır. r , o anki turun sayısı ve G de, son $1/P$ turda küme başı olmamış düğümlerin kümesidir. Bu eşik değeri kullanılarak her düğüm $1/P$ tur içerisinde her hangi bir noktada küme başı olacaktır. İlk turda ($r = 0$) her düğümün küme başı olmada P olasılığı vardır. İlk turda küme başı olan algılayıcılar bir dahaki $1/P$ tur sayısınca küme başı olamazlar. Bu nedenle tur sayısı arttıkça geriye kalan düğümlerin küme başı olma olasılığı da

artacaktır. Her düğüm küme başlığı yaptıktan sonra bütün düğümler tekrar küme başı olabilme durumuna geçerler.

Kendini küme başı seçen her düğüm diğer küme başı olmayan düğümlere bildirim mesajı yayınlırlar. Küme başı olmayan düğümler kendilerine en yakın mesafede olan küme başının kümesine katılmak durumumdadır. Çünkü gönderilecek mesaj mesafesi ne kadar fazla olursa o kadar fazla enerji harcanır. Mesajı alan düğümler o tur için hangi küme başının kümesine dahil olacaklarına karar verirler. Bu kararı verirken de bildirim mesajının sinyal gücüne bakarlar. Aldıkları sinyalin en güçlüsü kendilerine en yakın olan küme başıdır. Çünkü mesafe arttıkça sinyal gücü zayıflamaya başlar. Her düğüm hangi küme başının kümesine dahil olacağına karar verdikten sonra ilgili küme başına kümeye katılım için bildirim mesajı gönderir.

Kurulum safhası bu şekilde tamamlandıktan sonra veri iletişim safhası başlar. Her düğümün gönderecek verisi olduğu var sayılır. Üye düğümler sırayla verilerini küme başına gönderirler. Küme başı tüm üyelerinden verileri aldıktan sonra sıkıştırma işlemini gerçekleştirip veriyi tek paket halinde baz istasyonuna iletir. Tüm küme başları verilerini baz istasyonuna ilettikten sonra bir tur tamamlanmış olur ve yeni bir kurulum safhası başlar.

LEACH'in direkt iletişim ve sabit kümeleme yöntemleriyle performans kıyaslaması Şekil 2.3'de gösterilmiştir. Çizelge 2.1'de de aynı yöntemlerle algılayıcı düğümlerin farklı başlangıç enerji seviyeleriyle (0.25J, 0.5J, 1J) yapılan denemeler sonucunda elde edilen sonuçlar tur bazında gösterilmektedir. Görüldüğü üzere ağ yaşam süresi ilk düğümün öldüğü tur olarak düşünüldüğünde yaklaşık 8 kat, son düğümün öldüğü tur olarak düşünüldüğünde de yaklaşık 3 kat arttırılmıştır. LEACH'te kullanılan dinamik kümelemenin sabit kümeleme yöntemine göre avantajı Şekil 2.3'te açıkça görülmektedir. Sabit kümelemede küme başı öldükten sonra o kümede yeni bir küme başı seçilmediğinden dolayı algılayıcıların enerjisi çok çabuk tükenmektedir.



Şekil 2.3 - LEACH protokolünün karşılaştırma grafiği. Algılayıcıların başlangıç enerjisi 0,5J'dür [33]

Çizelge 2.1 - LEACH protokolünün tur sayısı bazında simülasyon sonuçları [33]

Başlangıç Enerjisi (J/düğüm)	Protokol	İlk düğümün öldüğü tur	Son düğümün öldüğü tur
0,25	Direkt	55	117
	Sabit Kümeleme	41	67
	LEACH	394	665
0,5	Direkt	109	234
	Sabit Kümeleme	80	110
	LEACH	932	1312
1	Direkt	217	468
	Sabit Kümeleme	106	240
	LEACH	1848	2608

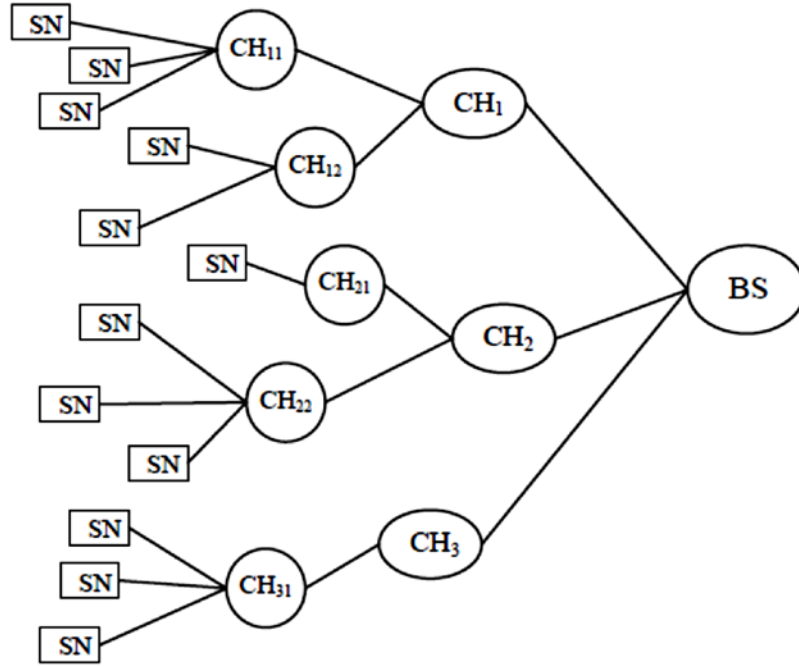
2.4.2 TL-LEACH (Two Level LEACH)

Bu protokol LEACH'in geliştirilmiş bir şeklidir [34]. Küme başları arasında iki katman oluşturulur (birincil ve ikincil). Her kümedeki birincil küme başları ikincil küme başlarıyla iletişim kurarlar ve ilgili ikinciller de üye düğümleriyle iletişim kurarlar (Şekil 2.4). Birincil ve ikincil küme başlarını seçme işlemi LEACH ile aynıdır.

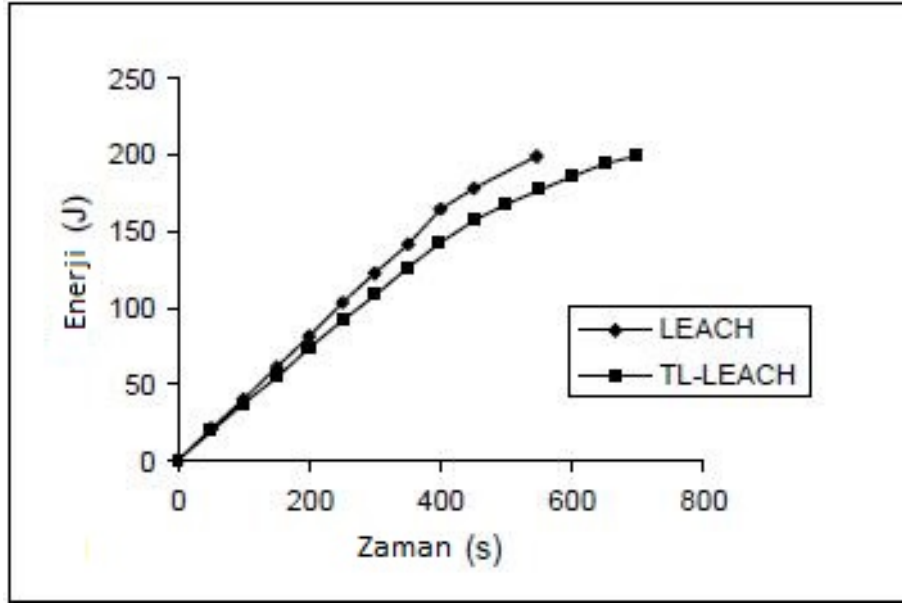
Birincil küme başlarının seçilme olasılığı, sayıları ikincil küme başlarından daha az olacağından ikincil küme başı seçilme olasılığından daha düşüktür. Kaynak düğümünden taşıyıcıya olan iletişim iki adımda gerçekleşir:

1. İkincil küme başları kümesindeki üye düğümlerden verileri toplarlar. Veri sıkıştırma işlemi bu adımda gerçekleşebilir.
2. Birincil küme başları ilgili ikincil küme başlarından verileri toplarlar. Veri sıkıştırma işlemi bu adımda da gerçekleştirilebilir.

Yapılan karşılaştırmalarda TL-LEACH’de kullanılan bu yöntem LEACH’e göre genel enerji tüketimini oldukça düşürmektedir. Şekil 2.5’te görüldüğü gibi TL-LEACH protokolü ağ yaşam süresince LEACH’e göre daha az enerji harcamaktadır ve verimliliği arttırmaktadır.



Şekil 2.4- TL-LEACH iletişim hiyerarşisi. SN: Algılayıcı düğüm, CH_{xx}: İkincil küme başı, CH_x: Birincil küme başı, BS: Baz istasyonu [34]



Şekil 2.5- TL-LEACH ve LEACH karşılaştırması [34]

2.4.3 HEED (Hybrid Energy Efficient Distributed Clustering)

HEED kümeleme şemasında küme başları konuşlandırılmış algılayıcılardan seçilir [31]. HEED küme başları seçerken enerji ve iletişim maliyetlerini hibrit olarak ele alır. LEACH'den farklı olarak küme başları rastgele seçilmez. Sadece fazla enerjisi kalan algılayıcılar küme başı olabilir. HEED'in üç temel özelliği vardır:

- Birbirlerinin iletim alanında olan iki düğümün beraber küme başı olma olasılığı çok düşüktür. LEACH'den farklı olarak bu, küme başlarını ağ içinde iyi şekilde dağıtıyor demektir.
- Enerji tüketiminin bütün düğümler için tek tip (eşdeğer) olduğu varsayılmaz.
- Bir düğümün iletim mesafesine bakılarak küme başı seçim olasılığı, küme başları arasındaki bağlantıdan emin olmak için ayarlanabilir.

Protokol üç safhaya ayrılmıştır.

1. **Başlangıç safhası:** İlk başta bütün algılayıcılar arasından bir başlangıç yüzdesi miktarında küme başı seçilir. Yüzde değeri (C_{prob}), diğer algılayıcılara

yapılan başlangıç küme başı bildirimini kısıtlamak için kullanılır. Her algılayıcı, küme başı olabilmek için olasılığını (CH_{prob}) şu şekilde ayarlar:

$$CH_{\text{prob}} = C_{\text{prob}} * E_{\text{residual}} / E_{\text{max}}$$

E_{residual} = Algılayıcının o anda geriye kalan enerjisi

E_{max} = Algılayıcının başlangıç enerjisi

CH_{prob} 'un belirli bir eşik değerinin (p_{min}) altında olmasına izin verilmez. p_{min} eşik değeri de E_{max} 'a ters orantılı olarak seçilir.

2. **Tekrarlama safhası:** Bu safhada her algılayıcı en az iletim maliyetli olan küme başını bulana kadar birkaç tekrarlama girer. Eğer herhangi bir küme başından haber alamazsa düğüm kendini küme başı seçer ve komşularına durumun değiştiğini haber veren bir mesaj yayınlar. En sonunda her algılayıcı CH_{prob} değerini ikiye katlar ve bu safhanın bir sonraki tekrarına geçer. CH_{prob} değeri 1 oluncaya kadar bu safha devam eder. Bu nedenle algılayıcının komşularına duyurabileceği iki tip küme başı durumu vardır.

- a. **Kararsız durum:** Eğer düğümün CH_{prob} değeri 1'in altındaysa düğüm kararsız küme başı olur. Sonraki bir tekrarlama eğer düşük maliyetli bir küme başı bulursa durumunu normal düğüm olarak değiştirebilir.

- b. **Son durum:** Eğer düğümün CH_{prob} değeri 1'e ulaşırsa düğüm kalıcı olarak küme başı olur.

3. **Sonlandırma safhası:** Bu safhada her algılayıcı kendi durumu için son bir karar verir. Ya en düşük maliyetli küme başını seçer ya da kendini küme başı yapar.

2.4.4 DWEHC (Distributed Weight-Based Energy-Efficient Hierarchical Clustering)

Bu protokolda her algılayıcı kendi alanındaki düğümlerin konumlarını belirledikten sonra kendi ağırlığını hesaplar [35]. Ağırlık, algılayıcının enerji rezervi ve

komşularına olan mesafenin bir fonksiyonudur. Bir alanda en büyük ağırlığa sahip olan düğüm küme başı olarak seçilir ve diğer düğümler üye düğüm olur. Bu safhada düğümler birinci seviye düğümler olarak düşünülür çünkü küme başına doğrudan bağlıdırlar. Bir düğüm bu üyeliğini küme liderine ulaştırmak için en az miktarda enerji kullanımını temel alarak kademeli biçimde ayarlar. Temel olarak bir düğüm, küme başı olmayan komşu düğümleriyle en düşük maliyetli küme başını bulmaya çalışır. Bir düğüm, komşularına olan mesafelerine göre birinci seviye üye düğüm olarak kalabilir veya küme başına iki atlamalı bir yol üzerinden ulaşarak ikinci seviye üye düğüm olacağını belirleyebilir. Bu süreç düğümlerin en enerji verimli küme içi topolojisini oturtana dek devam eder. Seviyelerin sayısını limitlemek için her küme bir alan sınırı atanır.

2.4.5 EECS (Energy Efficient Clustering Scheme)

EECS’de küme başı adayları belirli bir tur için küme başı olmak için gerekli nitelikleri sağlamak adına yarışır [36]. Bu yarışta adaylar kalan enerjilerini komşu adaylara bildirirler. Eğer bir düğüm daha fazla enerjisi kalan başka bir düğüm bulamazsa kendisi küme başı olur. Küme oluşumu LEACH’ten farklıdır. EECS kümenin baz istasyonuna olan mesafesini baz alarak dinamik olarak küme büyüklüğünü değiştirir. Böylelikle algoritma baz istasyonundan uzak mesafede olan kümelerin fazla enerji harcama probleminin önüne geçmeye çalışmaktadır. Bu da ağdaki enerji dağılımını geliştirmektedir ve ağın yaşam süresini uzatmaktadır. Algoritmanın küme başı seçim safhasında ufak bir kontrol destek işlemiyle iyi dağıtılmış küme başları seçilir. Küme oluşturma safhasında da kümelerde yük dengelemesi için yeni bir ağırlıklı fonksiyon geliştirilmiştir.

2.4.6 HCR (Hierarchical Cluster-Based Routing)

Bu hiyerarşik kümeleme yönteminde baz istasyonu operasyona dahil edilmektedir [37]. Bu yönlendirme şemasında “uzak mesafelere mesaj göndermede harcanan enerji kısa mesafe iletimde harcanan enerjiden çok daha fazla ilkesi” göz önüne alınmıştır. Tek bir küme başı kullanmak yerine lider kümesi (head-set) adı verilen bir işbirlikçi kümesi kullanarak LEACH protokolünü genişletmişlerdir. Bunun yanında, sonuç paketleri gönderildiği zaman baz istasyonu düğümlerin enerji durumları hakkında bilgilendirilir. Böylelikle baz istasyonu düğümlerin enerji durumlarını kullanarak enerji verimli kümeler oluşturur. Lider kümesinin üyeleri, mesajları uzaktaki baz istasyonuna göndermekle sorumludurlar. Bir anda sadece bir lider küme üyesi aktif olurken diğerleri uyku halinde olur. Baz istasyonuna iletim görevi lider kümesi üyeleri arasında eşit oranda dağıtılmaktadır.

Bu çalışmada kullanılan radyo modelinde LEACH’dekine ek olarak uzak ve yakın mesafeler için harcanan enerjiler farklı denklemlerle ele alınmıştır. Yani algılayıcılar arasındaki iletişimde yakın mesafe enerji harcama denklemi; baz istasyonu ile iletişimde uzak mesafe enerji harcama denklemi kullanılmıştır. l bit uzunluğuna sahip bir mesajı d_l uzun mesafesine göndermek için harcanan enerji Denklem (2.4)’te gösterilmiştir.

$$E_T = lE_e + l\varepsilon_l d_l^4 \quad (4.4)$$

Benzer şekilde d_s kısa mesafesine iletimde harcanan enerji Denklem (2.5)’te gösterilmiştir.

$$E_T = lE_e + l\varepsilon_s d_s^2 \quad (4.5)$$

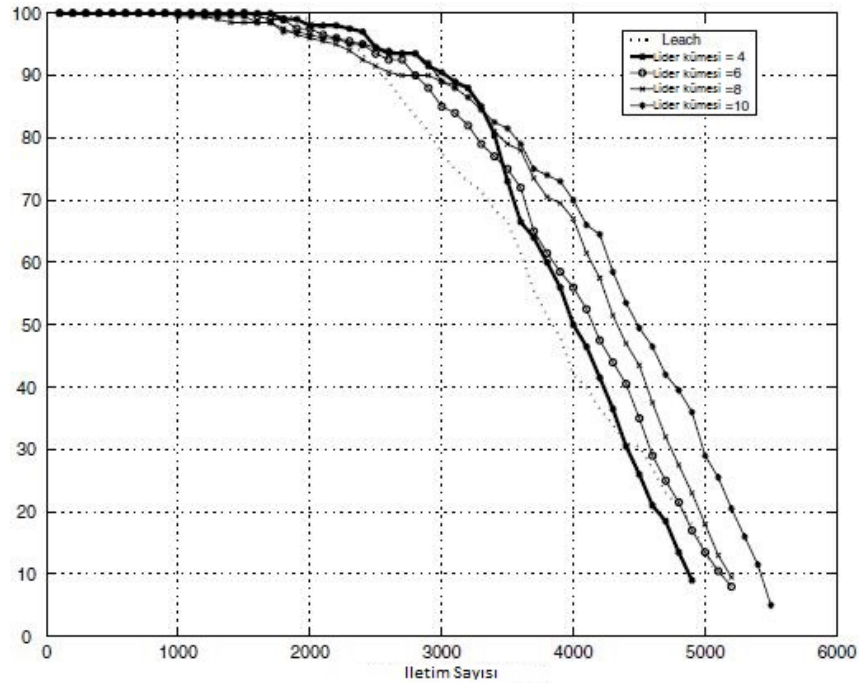
Mesajı alırken harcanan enerji Denklem (2.6)’da gösterilmiştir:

$$E_R = lE_e + lE_{BF} \quad (4.6)$$

Denkleimde kullanılan sabit deęerler řu řekildedir: $\epsilon_s = 10$ pJ/bit/m², $\epsilon_l = 0,0013$ pJ/bit/m⁴, $E_e = 50$ nJ/bit, mesaj sıkıřtırmada harcanan enerji $E_{BF} = 5$ nJ/bit, $l = 4000$ bit.

Çalıřmanın seim safhasında baz istasyonu aktif dđęümlerin sayısından haberdardır ve dđęümlerin enerji seviyelerini kullanarak uygun küme sayısına karar verir. LEACH'ten farklı olarak bařlangıçta baz istasyonu bir grup küme bařı seer ve bu bilgiyi aęda yayımlar. Daha sonra kümelere ayrılma, yani üye dđęümlerin seimi LEACH ile aynıdır. Seim safhasının son adımında her küme bařı kendi kümesi için üye dđęümlerden aldıęı bildirim mesajının sinyal seviyesine bakarak bir iřbirlikçi kümesi (lider kümesi) oluřturur.

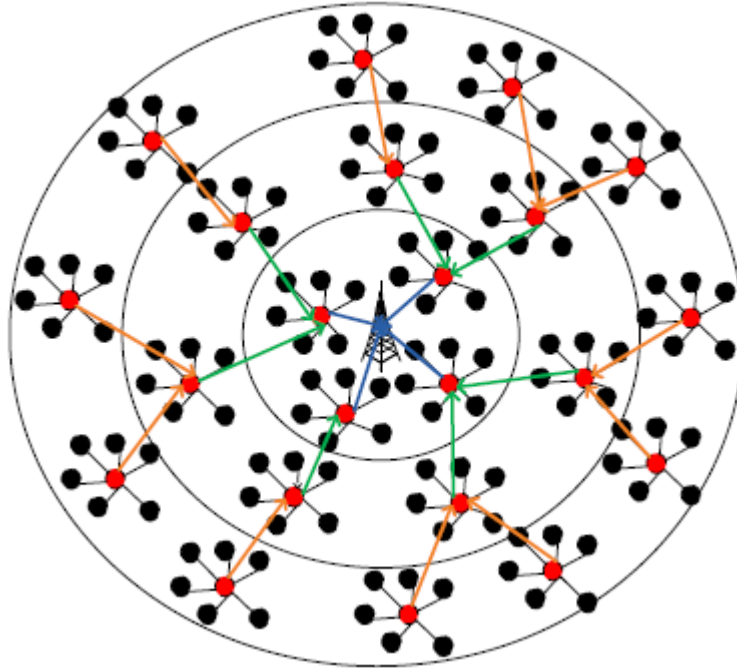
Simülasyonda farklı dđęüm daęılımlarında, farklı büyüklükte iřbirlikçi kümeleri ile LEACH protokolüyle karřılařtırma yapılmıřtır. Protokolün LEACH'e göre daha fazla iletiřim gerekleřtirdięi görölmüřtür (řekil 2.6).



řekil 2.6 - HCR ve LEACH karřılařtırma grafięi [37]

2.4.7 MR-LEACH (Multihop Routing with LEACH)

Bu yöntemde çok atlamalı (multihop) bir yönlendirme algoritması öne sürülmektedir [38]. Çalışmadaki amaç kümeleme hiyerarşisini arttırarak algılayıcı düğümlerinin enerji tüketimini azaltmaktır. Eşit sayıda kümeler oluşturmak için baz istasyonu devreye girer ve kümeleme hiyerarşisini belirler. MR-LEACH'te küme başları sadece üye düğümlerden veri toplamazlar. Aynı zamanda düşük seviyedeki küme liderleri için de taşıyıcı görevi yaparlar. Çünkü düşük seviyedeki küme başları baz istasyonuna veri gönderimini kendilerinden yüksek seviyedeki küme başları aracılığıyla gerçekleştirirler. Böylelikle küme başları, kökü baz istasyonu olan bir ağaç yapısı oluştururlar (Şekil 2.7). Bu ağaçta ara düğümler küme başları, yaprak düğümler ise üye düğümlerdir. Bu şema düşük seviyeden yüksek seviyeye doğru çok atlamalı bir yapı sayesinde ağ yaşam süresini uzatmaktadır. Simülasyon sonuçlarında MR-LEACH'in LEACH'e ve direkt iletme göre çok daha performanslı olduğu ortaya çıkmıştır. Toplam iletim sayısı neredeyse LEACH'in iki katındadır.



Şekil 2.7 - MR-LEACH protokolünün hiyerarşik yapısı [38]

2.4.8 E-LEACH (Energy LEACH)

Bu protokol LEACH'teki küme başı seçme prosedürünü geliştirmiştir [39]. İlk turda her düğümün eşit enerjisi vardır ve her birinin küme başı olmak için aynı olasılığı vardır. İlk turdan sonra her düğümün enerji seviyeleri kontrol ediliyor ve ikinci tur için fazla enerjisi olan düğümler küme başı seçiliyor. Bütün düğümler ölene kadar bu işlem devam ediyor. Diğer bütün işlemler LEACH ile aynıdır. Yapılan simülasyon sonuçları bu şekilde bir küme başı seçiminin ağ yaşam süresini oldukça uzattığını göstermektedir.

2.4.9 BEC (Based Energy Clustering)

Bu geliştirilmiş protokolde küme başlarının sayısına şu şekilde karar verilmektedir [40]:

$$k = \frac{\sqrt{27} * M}{2 * r^2 * \pi^2} \quad (4.7)$$

Denklem (2.7)'de M, gözlemlenen bölgenin alanıdır. r, düğümlerin algılama çapıdır. k da küme başı sayısıdır.

Küme başlarının seçim stratejisi enerji faktörü ve eşik değeri göz önüne alınarak işlenir. Eşik değeri Denklem (2.8)'deki gibi hesaplanır.

$$T(n) = \frac{E_i * k}{E_{total}} \quad (4.8)$$

E_i , düğümde geriye kalan enerji miktarı; E_{total} , bütün düğümlerin geriye kalan enerjilerinin toplamı; k da küme başlarının sayısıdır. Küme başları belirlendikten sonra BEC algoritması LEACH ile aynı kümeleme işlemlerini gerçekleştirir. Simülasyon sonucu yapılan karşılaştırmalarda protokolün ağ yaşam süresini LEACH'e göre uzattığı görülmektedir.

2.4.10 LEACH-DCHS (LEACH-Deterministic Cluster-Head Selection)

Bu protokolda LEACH'i geliřtirmek için düğümlerin enerji seviyelerini göz önüne almıřtır [41]. LEACH'te kullanılan eřik deęerini düğümün geriye kalan enerjisine göre ayarlamayı hedeflemiřtir. Geliřmiř eřik deęeri Denklem (2.9)'da gösterilmiřtir.

$$T(n) = \frac{P}{1 - P(r \bmod 1/P)} \frac{E_{n_current}}{E_{n_max}} \quad (4.9)$$

$E_{n_current}$, düğümün o anki geriye kalan enerjisidir. E_{n_max} , düğümün bařlangıç enerjisidir. Simülasyon sonuçlarında bu deęiřiklięin LEACH'e kıyasla ilk düğümün ölümünü %30, düğümlerin yarısının ölümünü %20 geciktirdięi görülmüřtür. Bununla beraber eřik deęerinde böyle bir deęiřiklik yapmanın bir dezavantajı vardır. Belli bir turdan sonra aę tıkanabiliyor. Bunun nedeni küme bařı eřik deęeri çok düşük olması. çünkü geriye kalan düğümlerin enerji seviyeleri düşük olmaktadır. Bu sorunu ařmak için eřik deęeri Denklem (2.10)'da görüldüğü gibi biraz daha genişletilmiřtir.

$$T(n) = \frac{P}{1 - P(r \bmod 1/P)} \left[\frac{E_{n_current}}{E_{n_max}} + \left(r_s \operatorname{div} \frac{1}{P} \right) \left(1 - \frac{E_{n_current}}{E_{n_max}} \right) \right] \quad (4.10)$$

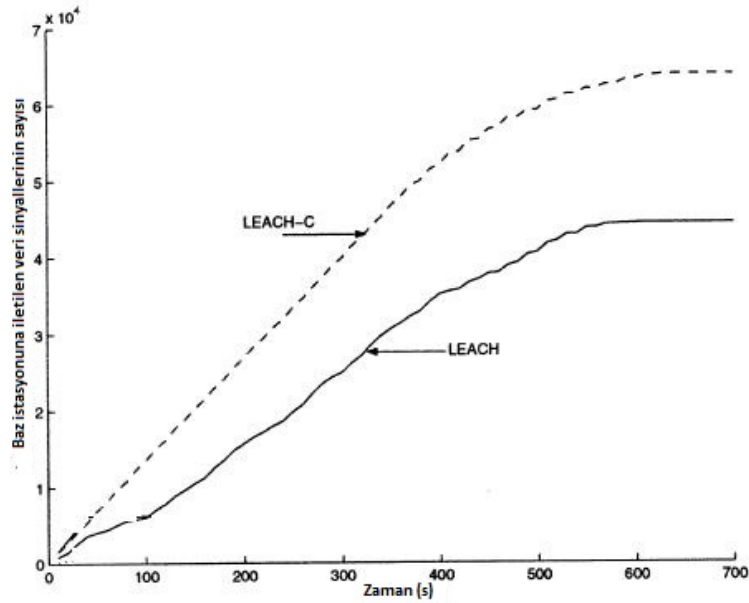
r_s , küme bařı olmamıř bir düğümün geçirdięi tur sayısıdır. r_s $1/P$ deęerine ulařınca $T(n)$ eřik deęeri denkleme "geriye kalan enerji" parametresi eklenmemiř haline döndürülür. Böylelikle eřik deęeri arttırılarak n düğümünün küme bařı olma řansı arttırılır.

2.4.11 LEACH-C (LEACH-Centralized)

Bu çalışmada LEACH'deki kurulum safhasını tamamen deęiřtirerek merkezi bir kontrol algoritması kullanıp daha iyi sonuç üretecek kümeler oluřturulması hedeflenmiřtir [42]. Veri iletim safhası LEACH ile aynıdır.

LEACH-C'nin kurulum safhasında her düğüm konum bilgisini ve enerji seviyesini baz istasyonuna iletir. Baz istasyonu enerjini yükünü düğümler arasında dengelemek için düğümlerin ortalama enerjisini hesaplar ve enerjisi ortalamanın altında olan düğümler o turda küme başı olamazlar. Baz istasyonu geriye kalan olası küme başları ile optimal kümeleri bulmak için bir benzetimli tavlama (simulated annealing) algoritması kullanır. Bu algoritma üye düğümlerin verilerini küme başlarına gönderirken harcadıklarını enerjii en aza indirmeye çalışır. Bunu da bütün üye düğümlerin en yakın küme başıyla arasındaki mesafeleri en aza indirmekle yapar. İşlem tamamlandıktan sonra baz istasyonu bulduğu optimal küme bilgilerini ağdaki algılayıcılara yayımlar. Yayımlanan mesajda her düğüm için küme başı ID'si bulunur. Eğer bir düğümün aldığı mesajdaki ID kendi ID'si ile eşitse o düğüm küme başı durumuna geçer, aksi halde hangi küme başının kümesine dahil olacağını öğrenmiş olur. Kümeler kurulduktan sonraki veri iletişim safhası LEACH'deki gibi gerçekleşir. İletişimde kullanılan radyo modeli [37]'da kullanılan modelle aynıdır.

Simülasyon sonuçlarında yapılan karşılaştırmalarda LEACH-C'nin ağ yaşam süresini arttırdığı ve baz istasyonun aldığı veri miktarının oldukça arttığı gözlemlenmiştir (Şekil 2.8).



Şekil 2.8 - LEACH-C ve LEACH karşılaştırması [42]

2.4.12 BCDCP (Base-Station Controlled Dynamic Clustering Protocol)

Bu protokol ağ enerjisini tüm düğümler arasında eşit şekilde dağıtmayı denemektedir ve küme başlarını seçmek için özel bir yaklaşım önermektedir [43]. [33]'deki gibi ağı kümelere bölmektedir ve bu kümeler [37]'daki gibi baz istasyonu tarafından yönetilir. Her turdan sonra baz istasyonu bütün düğümlerin enerji ortalamasını hesaplar ve bu değeri belirli bir eşik değeriyle karşılaştırır. Enerjisi eşik değerinden yüksek olan düğümler bir sonraki tur için küme başı olma şansına sahip olur. Daha sonra kümelere bölmek için bir algoritma kullanarak ağı birkaç kümeye ayırır. Bunu yaparken de küme başlarının ağda düzgün bir şekilde yerleştirildiğinden emin olur. Bunun ilerisinde BCDCP her kümenin aşağı yukarı aynı sayıda üye düğümü olacak şekilde bir kümeleme tekniği kullanır. Küme sayısı ve küme başları belirlendikten sonra veri iletim safhası başlar. BCDCP bu safhada çok atlamalı bir iletim şeması kullanır. Simülasyon sonuçlarında BCDCP'nin enerji tüketimini LEACH'e göre %15 oranında düşürdüğü gözlemlenmiştir.

2.4.13 LCA (Linked Cluster Algorithm)

Bu yöntem ilk kümeleme yöntemlerindedir [44] [45]. Bu yöntem hareketli algılayıcı içeren ağlar için geliştirilmiştir. Kümeleme sayesinde küme başlarının, üye düğümler hareket halindeyken bağlanabilecekleri bir omurga ağ oluşturulması beklenmektedir. Yöntemde her düğüme bir ID atanmıştır ve küme başı olmanın iki yolu vardır. İlk yolda kümedeki en yüksek ID numarasına sahip olan düğüm kendini küme başı seçer. İkinci yolda da eğer düğümün komşularının hiçbiri küme başı değilse o zaman küme başı olur.

2.4.14 FLOC (Fast Local Clustering Service)

Bu çalışmada düğümler küme başlarına olan mesafelerine göre iç düğüm (i-band) ve dış düğüm (o-band) olarak sınıflandırılırlar [46]. İç düğümler küme başlarıyla iletişimde pek kopukluk yaşamazken dış düğümlerin küme başları ile aralarındaki mesafe çok olduğundan dolayı bu düğümlerden küme başına gelen mesajlar kaybolabilir. Bir düğüm rastgele bir sürede kendisine uygun bir küme başından davet alana dek bekleme halinde olur. Eğer bir davet gelmezse düğüm aday küme başı durumuna geçer ve ağa aday küme başı olduğunu mesajla haber verir. Bu aday mesajını alan bir k iç düğümü (zaten C_k kümesinin üye olan), aday küme başına başka kümenin üyesi olduğu mesajını ileterek cevap verir. Cevabı alan aday küme başı, durumdaki çelişkinin farkına varır ve C_k kümesine dış düğüm olarak katılır. Eğer aday küme başı, herhangi bir çelişki cevabı almazsa küme lideri durumuna geçer ve diğer düğümleri kümesine davet etmeye başlar.

Beklemede olan normal bir düğüm de herhangi bir davet mesajı almazsa bir kümeye dış düğüm olarak katılır. Eğer düğüm daha sonra yakın bir küme başından davet alırsa iç düğüm olarak yeni kümesine katılır.

2.4.15 Multihop-LEACH

Bu çalışmada [38]'den farklı olarak küme içi düğüm iletişimde de çok atlamalı yönlendirme işlemi gerçekleştirilmiştir [47]. Genelde küme içinde kullanılan tek atlamalı iletişim optimum çözüm olmayabilir. Algılayıcı düğümleri yoğun bitkili bir alanda veya düzgün olmayan/engebeli bir arazidelerse üye düğümlerin küme başına ulaşmasında çok atlamalı yönlendirme kullanılması faydalı olmaktadır. Üye düğümler küme başlarının kapsama alanında olmadığı durumlarda ağdan kopacakları için her düğüm kendisine yakın başka bir üye düğümü küme başı gibi kullanma isteği gönderebilmelidir. Simülasyonda alınan sonuçlarda diğer protokollere nazaran bu protokolün ağ yaşam süresini uzattığı gözlemlenmiştir.

2.4.16 MS-LEACH (Multihop-Singlehop LEACH)

Bu yöntemde yine [33] baz alınmıştır fakat veri iletişim safhasında değişiklik yapmıştır [48]. Veri iletim safhasında CH_i küme başı düğümü $Q_{critical}$ ile temsil edilen küme alanı büyüklüğünün kritik değerini hesaplar. Bu değer yaklaşık değeri kümedeki düğüm sayısı ve düğüm pozisyonlarıyla alakalı olup Q ile temsil edilmektedir.

Eğer Q değeri $Q_{critical}$ değerinden küçükse CH_i bir şey yapmayıp üye düğümlerden gelecek paketleri bekler. Eğer büyükse CH_i Dijkstra algoritması kullanarak [49] [50] bir yönlendirme yol ağacı hesaplar ve bu ağacı bir veri paketine dönüştürüp kümedeki düğümlere yayınlar.

Aynı zamanda küme başı olmayan düğüm N_j bir T zamanlayıcısı oluşturur ve CH_i den yönlendirme yol ağacının yayınlanmasını bekler. Eğer N_j yönlendirme yol ağacını alırsa ve T pozitif değerdeyse N_j yönlendirme yol ağacına bakarak sonraki hopu (küme başına ulaşan yolda kendisine en yakın düğüm) bulur ve veri paketlerini bu düğüme yollar. Aksi halde paketleri doğrudan CH_i ye yollar. Simülasyonda protokol küçük çapta ve büyük çapta uygulama alanlarında denenmiştir. Büyük çapta alanda %200 oranında iyileştirme yapmıştır. Küçük çapta alanda ise LEACH ile aynı oranlarda performans sağlamıştır.

2.4.17 ECHSSD (Efficient Cluster Head Selection Scheme for Data Aggregation in Wireless Sensor Network)

Bu yöntem yeni küme başı seçimi konusunda [42]'e benzer bir prosedür önermektedir [51]. Fakat optimum kümeleri bulmak için herhangi bir sezgisel yöntem kullanmamaktadır.

[42]'deki küme başı değiştirme prosedüründe algılayıcıların geriye kalan enerjileri dikkate alınıp tüm ağın enerji ortalaması alınıyordu. Eğer küme başının geriye kalan enerjisi bu ortalamanın altında ise o küme için enerjisi daha fazla olan bir düğüm yeni küme başı olarak seçiliyor. ECHSSD de ise tüm ağın enerji ortalamasını hesaplamak yerine ilgili kümenin enerji ortalaması hesaplanmaktadır. Eğer küme

başının enerji seviyesi üye düğümlerin enerji ortalamasının altındaysa bir sonraki tur için kümedeki en fazla enerjisi olan düğüm yeni küme başı seçilir. Yeni lider seçildikten sonra bir sonraki tur için yeni kümeler oluşturulur. Diğer prosedürler [33] ile aynıdır. Simülasyon sonuçlarında ECHSSD [33] ve [42]'e göre ağ yaşam süresini daha uzatmıştır.

3. GENETİK ALGORİTMA

Genetik algoritmalar (GA) evrimsel sistemleri modellemek ve problemleri çözmek için biyolojik evrim sürecini taklit eden bilgisayar programlarıdır [52]. GA ilk olarak 1960lı yıllarda John Holland tarafından tanımlanmış ve ilerleyen yıllarda Michigan Üniversitesinde mesai arkadaşları ve öğrencileriyle beraber geliştirilmiştir.

Evrime mekanizmasının birçok alanda hesaba dayalı problemlerin çözümü için çok uygun olduğu görülmektedir. Birçok hesaba dayalı problem çözümleri çok büyük sayıda olasılığın içerisinde aramayı gerektirir. Örnek olarak hesaplamalı protein mühendisliğinin problemini verebiliriz. Bu problemde belirli özelliklere sahip bir protein için büyük sayıda olası aminoasit dizileri arasında arama yapacak bir algoritmaya ihtiyaç duyulur.

GA da bu ve buna benzer problemlerin çözümü için geliştirilmiş, biyolojik evrimdeki doğal seçim ilkesinin bilgisayar ortamına uyarlanmasıyla elde edilen bir arama yöntemidir [53]. Genetik algoritmalar, yeni aday çözümler üretmek için çaprazlama ve mutasyon gibi operatörleri kullanır. Büyük sayıda aday sonuç içerisinde arama yaparken bu operatörleri kullanarak verilen parametreler doğrultusunda optimum sonucu bulur [54].

3.1 Genetik Algoritmanın Bazı Uygulamaları

Genetik algoritmalar birçok fen ve mühendislik problemlerinde ve modellerinde kullanılmıştır. Bazı örnek uygulama alanları şunlardır [52]:

- **Optimizasyon:** Genetik algoritmalar devre düzeni ve iş çizelgeleme gibi tümleşik optimizasyon problemlerinde ve nümerik optimizasyon problemleri gibi birçok optimizasyon işinde kullanılmaktadır.
- **Otomatik Programlama:** Genetik algoritmalar bilgisayar programlarını spesifik görevlerde geliştirmek için kullanılmaktadır. Bunun yanında hücresel

otomat ve ağ sıralaması gibi hesaba dayalı yapıların tasarımında da kullanılmaktadır.

- **Makine öğrenimi:** Genetik algoritmalar sınıflandırma, hava tahmini ve protein yapısı tahmini gibi birçok makine öğrenimi uygulamasında kullanılmaktadır. Ayrıca genetik algoritmalar, özel bazı makine öğrenimi sistemleri geliştirmek için de kullanılmıştır. Sinir ağları ağırlıkları, sembolik üretim sistemleri ve robotlar için algılayıcılar gibi.
- **Ekonomik modeller:** Genetik algoritmalar yenilenme süreçlerini modellemek ve fiyat teklifi stratejisini geliştirmek için de kullanılmaktadır.
- **Bağışıklık sistemi modelleri:** Genetik algoritmalar doğal bağışıklık sisteminin çeşitli yönlerini modellemek için kullanılmaktadır.
- **Sosyal sistem modelleri:** Genetik algoritmalar işbirliği gelişimi, iletişim gelişimi ve karıncaların kuyruk takibi davranışı gibi sosyal sistemlerin gelişimsel yanlarını araştırmak için kullanılmaktadır.

3.1.1 Bazı özel problemlere uygulanışı

Genetik algoritmalar üretim alanında özellikle kaynaklardan en verimli şekilde yararlanmak, zamandan ve maliyetten tasarruf etmek için kullanılır [55].

Endüstride çok önemli rolü olan montaj hattı dengeleme probleminde en iyi sonucu bulabilmek için genetik algoritma kullanılmıştır. Bir iş istasyonundaki toplam harcanan işlem zamanlarını en aza indirmek için genetik algoritma ve bulanık küme mantığı beraber kullanılmıştır ve problemin çözümü gerçekleştirilmiştir [56].

Genetik algoritma atama problemlerinde de kullanılmıştır. Atama problemi k tane elemanın k tane farklı işe atanması problemidir. Genetik algoritmada uygunluk fonksiyonu, optimum atama kümesinin bulunmasında en aza indirgenecek şekilde tanımlanabilir. 1996'da Chu ve Beasley en az maliyetli atamayı hedefleyen problem için genetik algoritma kullanmışlardır. Zhao, Tsujimura ve Gen 1995 yılında iş istasyonu atama problemini çözmek için genetik algoritmadan faydalanmışlardır [57].

3.2 Genetik Algoritmanın İşleyişi

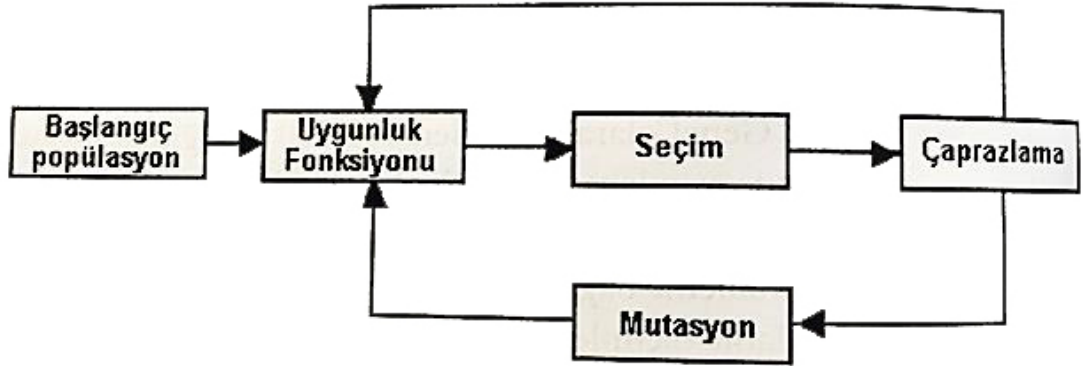
GA'da aday sonuçlar eşit boyutlu diziler şeklinde temsil edilir. Bu dizilerin her birine *kromozom* denir. Bir kromozomu (diziyi) oluşturan elemanlara da *gen* denir. Algoritmanın başlangıcında rastgele bir grup kromozom seçilerek önceden belirlenmiş büyüklükte bir *popülasyon* oluşturulur. Bu popülasyon içerisindeki kromozomlar (aday sonuçlar) algoritma süresince değişikliklere uğrar ve yeni nesiller (popülasyonlar) oluşturur (Şekil 3.1) [53].

Her popülasyonda önceden belirlenmiş uygunluk fonksiyonu sayesinde kromozomların ne kadar iyi sonuç üretip üretmediği kontrol edilir. Bir sonraki popülasyon oluşturulurken önceki popülasyondaki bazı kromozomlar değiştirilir ve yeni kromozomlar oluşturulur. Bu işlemde çaprazlama ve mutasyon operatörleri sayesinde olur. Bu operatörler de kromozomlara belirli olasılıklarda uygulanırlar. Bunun yanında popülasyon içerisinde uygunluk değeri düşük olan kromozomlar yeni oluşturulacak popülasyona katılmayabilirler, yani eleneceklerdir. Çünkü genetik algoritma biyolojik evrim sürecindeki iyi nesillerin kendilerini koruyup yaşama devam ederken, kötü nesillerin yok olması ilkesine dayanır.

Genetik algoritmanın işleyişini maddeler halinde sıralayacak olursak:

1. Başlangıçta rastgele olası çözümlerin (kromozomların) kodlandığı bir popülasyon oluşturulur. Popülasyonda bulunacak kromozom sayısı için bir standart yoktur. Bu sayıda yapılan işlemlerin karmaşıklığı ve aramanın ne kadar derin olacağı önemlidir. Kromozomların kodlanması uygulamaya göre farklı biçimlerde olabilir.
2. Popülasyondaki her bir kromozomun ne kadar iyi sonuç üretip üretmediği kontrol edilir. Bunun için önceden belirlenen uygunluk fonksiyonu kullanılır. Uygunluk fonksiyonu genetik algoritmanın en önemli unsurudur, çünkü algoritma sonucunda ortaya çıkacak çözümün kalitesini bu fonksiyon belirlemektedir. Kromozomların uygunluk değerlerinin bulunmasına *evrimleşme* adı verilir. Yapılan uygulamaya göre uygunluk fonksiyonu değişir.

3. Bu adımda uygunluk değerlerine göre kromozomlar seçilir. Seçilen kromozomlar çift olarak eşlenerek çaprazlama ve mutasyon işlemleri uygulanır. Böylelikle yeni kromozomlar meydana gelen yeni bir popülasyon oluşturulur. Seçim işlemi yapılırken yeni oluşturulan popülasyonda önceki popülasyondaki en iyi kromozomun kaybolacağı düşünülebilir. Bunun önüne geçmek için *seçkinlik* uygulanır. En azından iyi bir sonuç üzerinde hiçbir değişiklik yapılmadan yeni popülasyona aktarılır. Seçim işlemi için birkaç farklı yöntem vardır: rulet tekerleği seçimi, turnuva seçimi gibi.
4. Popülasyonda yeni kromozomlara yer açmak için kötü kromozomlar çıkartılır. Yeni popülasyondaki kromozomların uygunluk değerleri yeniden hesaplanır.
5. Belirlenmiş popülasyon oluşturma sınırı içerisinde yukarıdaki işlemler tekrar edilir ve en iyi kromozom (çözüm) elde edilmeye çalışılır.



Şekil 3.1 - Genetik algoritmanın işleyiş yapısı [53]

3.2.1 Kromozom kodlaması

GA ile çözüme başladığımızda yapılması gereken öncelikli işlem algoritmada kullanılacak kromozomların nasıl kodlanacağıdır. Bu kodlama biçimi problemin tipine göre değişmektedir. Kodlama biçimleri aşağıda verilmektedir [53].

a) İkili kodlama

Bu kodlama biçimi genetik algoritmanın ilk uygulamalarında kullanılmıştır. Günümüzde de en çok kullanılan kodlama biçimidir. Bu biçimde kromozomun genleri 0 ve 1 ile temsil edilerek bir bit dizisi olarak kodlanır (Çizelge 3.1). Dizideki her bit problemde rol oynayan bir özelliği taşır.

Çizelge 3.1 - Kromozom için ikili kodlama örneği

A Kromozomu	110100111001
B Kromozomu	010011101010

b) Permutasyon kodlama

Bu kodlama biçimi iş sıralama problemleri gibi permutasyon problemlerinde kullanılır. Burada her bir kromozom numara dizisinden oluşur (Çizelge 3.2).

Çizelge 3.2 - Kromozom için permutasyon kodlama örneği

A Kromozomu	4 0 5 7 2 6 1 3
B Kromozomu	2 4 1 5 3 7 6 0

c) Değer kodlama

Bu kodlama biçimi kompleks sayıların kullanıldığı problemlerde kullanılır. Bu tür problemlerde ikili kodlama kullanılması zor olduğundan ötürü bu şekilde farklı bir kodlama türü geliştirilmiştir. Bu biçimde kromozomlar bir takım değerler dizisidir. Bu değerler problemin türüne göre farklılaşabilir. Ondalık sayı, alfa-betik karakter veya kelime gibi (Çizelge 3.3).

Çizelge 3.3 - Kromozom için değer kodlama örneği

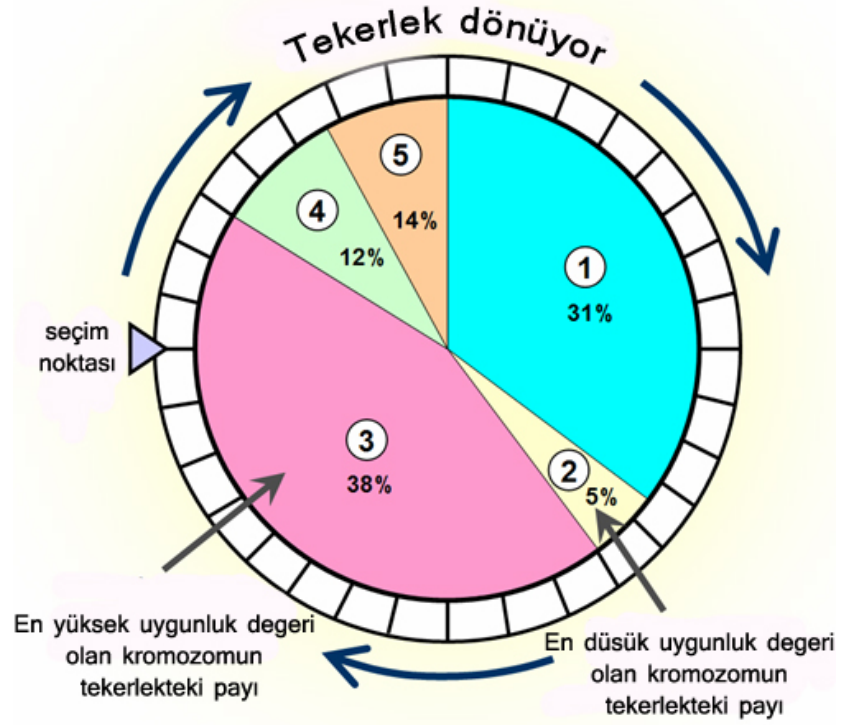
A Kromozomu	3,865 5,221 6,348 0,399 1,562
B Kromozomu	BCAJFKLHDERYFDLFFSE
C Kromozomu	(ileri), (geri), (sağ), (sol), (ileri)

3.2.2 Seçim

İlk popülasyondaki kromozomların uygunlukları hesaplandıktan sonra yeni popülasyon oluşturmak için ilk yapılması gereken işlem ebeveyn kromozomların seçim işlemidir. Bu işlem yukarıda bahsettiğimiz biyolojik evrimdeki gibi iyi olan kromozomların kurtulup kötü olan kromozomların elenmesi şeklinde olmalıdır. Seçim işleminin birkaç çeşidi vardır. Bunlar; rulet tekeri seçimi (roulette wheel selection), sıralama seçimi (rank selection) ve sabit durum seçimidir (steady state selection).

a) Rulet tekeri seçimi

Bu seçim işlemi kromozomların uygunluk değerine göre şansa bağlı olarak yapılmaktadır. Yani uygunluk değeri en büyük olan kromozomun seçileceği kesin değildir ama seçilme olasılığı çok yüksektir. Kromozomların seçilme olasılıklarını belirlemek için tüm kromozomların uygunluk değerleri toplanır. Daha sonra her kromozomun uygunluk değeri toplam değere bölünerek 0-1 aralığında seçilme olasılıkları belirlenir [53]. Sonra bu olasılıklar bir rulet tekerleğine konur. Uygunluğu yüksek olan kromozom tekerlekte daha büyük paya sahip olacaktır (Şekil 3.2). Teker popülasyondaki kromozom sayısı kadar döndürülür. Her döndürülüş sonunda bir kromozom eşleme için seçilir. Eğer genetik algorithmada seçkinlik uygulanacaksa tekerleğin döndürülme sayısı azalacaktır.



Şekil 3.2 - Temsili rulet tekeri [58]

Rulet tekerleği seçiminde kromozomların uygunluk değerlerinin pozitif olması gerekir. Negatif olma durumunda kromozomun seçilme şansı olmaz. Popülasyondaki kromozomların çoğunluğunun uygunluk değeri negatif olursa sonraki nesillerde daha iyi sonuçlar elde edilemeyebilir. Yeni oluşacak kromozomların uygunluk değerleri eski kromozomlarla benzer olacaktır ve belli bir noktada genetik algoritma takılı kalacaktır. Bunun önüne geçmek için uygunluk ölçeklendirme yapılır. Uygunluk fonksiyonunu f ile ifade edersek uygunluk ölçeklendirme (u) Denklem (3.1)'de gösterilmektedir

$$u = a * f + b \quad (3.1)$$

Bu denklemde “a” için 1.0-2.0 arasında değerler kullanılır. “b” ise uygunluk fonksiyonunun negatif değerini pozitive çevirecek bir değerdir.

b) Sıralama seçimi

Bu seçim işlemi kromozomların uygunluk değerlerinin birbirlerinden çok farklı olduğu durumlarda kullanılabilir. Örneğin bir kromozomun uygunluk değeri %90 oranına sahip olmuşsa geriye kalan kromozomların seçilme olasılığı çok düşük olacaktır. Bu da genetik algoritmanın performansını olumsuz yönde etkileyecektir.

Sıralama seçiminde popülasyondaki kromozomlar en kötü uygunluk değerinden en iyi uygunluk değerine doğru sıralanırlar. Bu sıralama sonucunda kromozomların uygunluk değerleri artık sıradaki dereceleriyle ölçülür. Yani en kötü kromozom 1 uygunluk değerinde, ikinci kötü kromozom 2 uygunluk değerinde ve en iyi kromozom k uygunluk değerinde olacaktır. k popülasyondaki toplam kromozom sayısıdır.

Bu seçim işleminin dezavantajı, en iyi kromozomlar diğer kromozomla yakın uygunluk değerlerine sahip olmasıdır. Bunun sonucunda genetik algoritmanın çözüme varması yavaşlayacaktır.

c) Sabit durum seçimi

Bu seçim tipinde diğer iki tipte olduğu gibi derecelendirme yapılmaz. Bu seçimin ana fikri popülasyondaki birçok kromozom bir sonraki popülasyona hiç değiştirilmeden aktarılmasıdır. Tabii bu aktarılacak olan kromozomlar yüksek uygunluk değerine sahip olan kromozomlardır. Örneğin popülasyonun ortalama uygunluk değerinin üstünde uygunluk değerine sahip olan kromozomlar. Düşük uygunluk seviyesine sahip olan kromozomlar bir sonraki popülasyona aktarılmazlar. Popülasyonda kalan boşluğu doldurmak için seçilen kromozomlardan birkaçı yeni kromozom üretmek için ebeveyn olarak seçilir ve popülasyon tamamlanmış olur.

Seçkinlik

Seçim sonrası yeni kromozomlar oluşturulurken eski popülasyonun içindeki en iyi kromozomlar yok olacaktır. Seçkinlik sayesinde eski popülasyondaki en iyi

kromozom veya birkaç iyi kromozom yeni popülasyona doğrudan kopyalanır. Bu şekilde popülasyondaki en iyi sonuçlar kaybolmayacağından genetik algoritmanın performansı arttırılmış olur.

3.2.3 Çaprazlama

Yeni kromozomları oluşturacak ebeveyn seçildikten sonra çaprazlama (crossover) operatörü devreye girer. Çaprazlama işleminde ebeveyn kromozom dizilerinde rastgele bir veya birden fazla nokta seçilir ve iki ebeveyn arasında bu noktalardan itibaren gen takası yapılır ve yeni kromozom dizileri meydana gelir.

Bit dizisi olarak kodlanmış kromozomlarda 4 farklı çaprazlama gerçekleştirilebilir [53].

- a) **Tek noktalı çaprazlama:** Bu çaprazlamada ebeveyn kromozom dizilerinde bir nokta seçilir. Seçilen noktaya kadar olan genler (bitler) birinci ebeveynden, geriye kalanlar ikinci ebeveynden alınır ve yeni kromozom (çocuk) oluşturulur (Çizelge 3.4).

Çizelge 3.4 - Tek noktalı çaprazlama örneği

Ebeveyn Kromozom 1	11011 00100110110
Ebeveyn Kromozom 2	10110 11100011010
Çocuk Kromozom 1	11011 11100011010
Çocuk Kromozom 2	10110 00100110110

- b) **Çift noktalı çaprazlama:** Bu çaprazlamada ebeveyn kromozom dizilerinde iki nokta seçilir. Seçilen ilk noktaya kadar olan genler birinci ebeveynden, iki nokta arasındaki genler ikinci ebeveynden ve geriye kalan genler yine birinci ebeveynden alınır ve yeni kromozom oluşturulur (Çizelge 3.5).

Çizelge 3.5 - Çift noktalı çaprazlama örneği

Ebeveyn Kromozom 1	110 010 10
Ebeveyn Kromozom 2	001 001 11
Çocuk Kromozom 1	110 001 10
Çocuk Kromozom 2	001 010 11

c) **Tek tip çaprazlama:** Bu çaprazlama biçiminde iki ebeveyn arasında genler önceden belirlenmiş bir karıştırma olasılığıyla değiştirilir. Örneğin karıştırma olasılığı 0,5 ise genlerin yaklaşık olarak yarısı takas edilecektir. Çizelge 3.6’da tek tip çaprazlama örneği gösterilmiştir. Örnekteki çocuk kromozomların genlerinin tabanında yazan rakamlar o genin hangi ebeveynden geldiğini göstermektedir. Takas işlemi kromozom üzerindeki herhangi bir gende gerçekleşebilir.

Çizelge 3.6 - Tek tip çaprazlama örneği

Ebeveyn Kromozom 1	1 1 0 0 1 0 1 0
Ebeveyn Kromozom 2	0 0 1 0 0 1 1 1
Çocuk Kromozom 1	1 ₁ 0 ₂ 1 ₂ 0 ₁ 0 ₂ 0 ₁ 1 ₁ 1 ₂
Çocuk Kromozom 2	0 ₂ 1 ₁ 0 ₁ 0 ₂ 1 ₁ 1 ₂ 1 ₂ 0 ₁

3.2.4 Mutasyon

Mutasyon, çaprazlama sonrasında çeşitliliği arttırmak amaçlı uygulanan bir diğer operatördür. Çaprazlama operatörü yeterince farklı kromozomlar oluşturamayacağından ileriki nesillerde oluşan kromozomlar birbirlerine benzeyecektir. Bu da en iyi çözümü bulmaya engel olacaktır. Bu benzerliklerin önüne geçmek için mutas-

yon operatörünü kullanmak kaçınılmazdır. Bu işlemde kromozomun kendi içerisindeki genler rastgele değer veya yer değiştirir. Ancak mutasyon operatörünün gen üzerinde uygulanma olasılığı doğru belirlenmelidir. Aksi halde kromozomlar çok fazla değişime uğrayacak ve genetik algoritma çözüm amacından sapacaktır [54]. Mutasyon işleminin de farklı uygulanış biçimleri vardır.

Bunlardan biri çocuk kromozomun genlerinin her biri mutasyon olasılığında değiştirilir (Çizelge 3.7). Yani o anda genlerin yalnız biri değişebilir, birden fazla gen de değişebilir veya hiçbir gen değişmeyebilir. Bu durum bizim önceden belirlediğimiz mutasyon olasılığıyla ilgili bir durumdur.

Çizelge 3.7 - Basit mutasyon örneği

Mutasyon öncesi	Mutasyon sonrası
1 0 1 1 0 1 1 0	1 0 1 1 0 0 1 0

Başka bir mutasyon biçimi de rastgele seçilmiş iki komşu genin yer değiştirmesidir (Çizelge 3.8).

Çizelge 3.8 - Rastgele seçilmiş komşu iki genin yer değiştirmesi

Mutasyon öncesi	Mutasyon sonrası
1 0 1 1 0 1 1 0	1 1 0 1 0 1 1 0

Bir diğer biçim rastgele seçilen iki genin yer değiştirmesidir (Çizelge 3.9).

Çizelge 3.9 - Rastgele seçilmiş iki genin yer deęiřtirmesi

Mutasyon öncesi	Mutasyon sonrası
1 0 1 1 0 1 1 0	1 0 0 1 0 1 1 1

Dördüncü bir biçim de rastgele seçilen üç genin rastgele birbirlerinin yerine geçmeleridir (Çizelge 3.10).

Çizelge 3.10 - Rastgele seçilen üç genin yer deęiřtirmesi

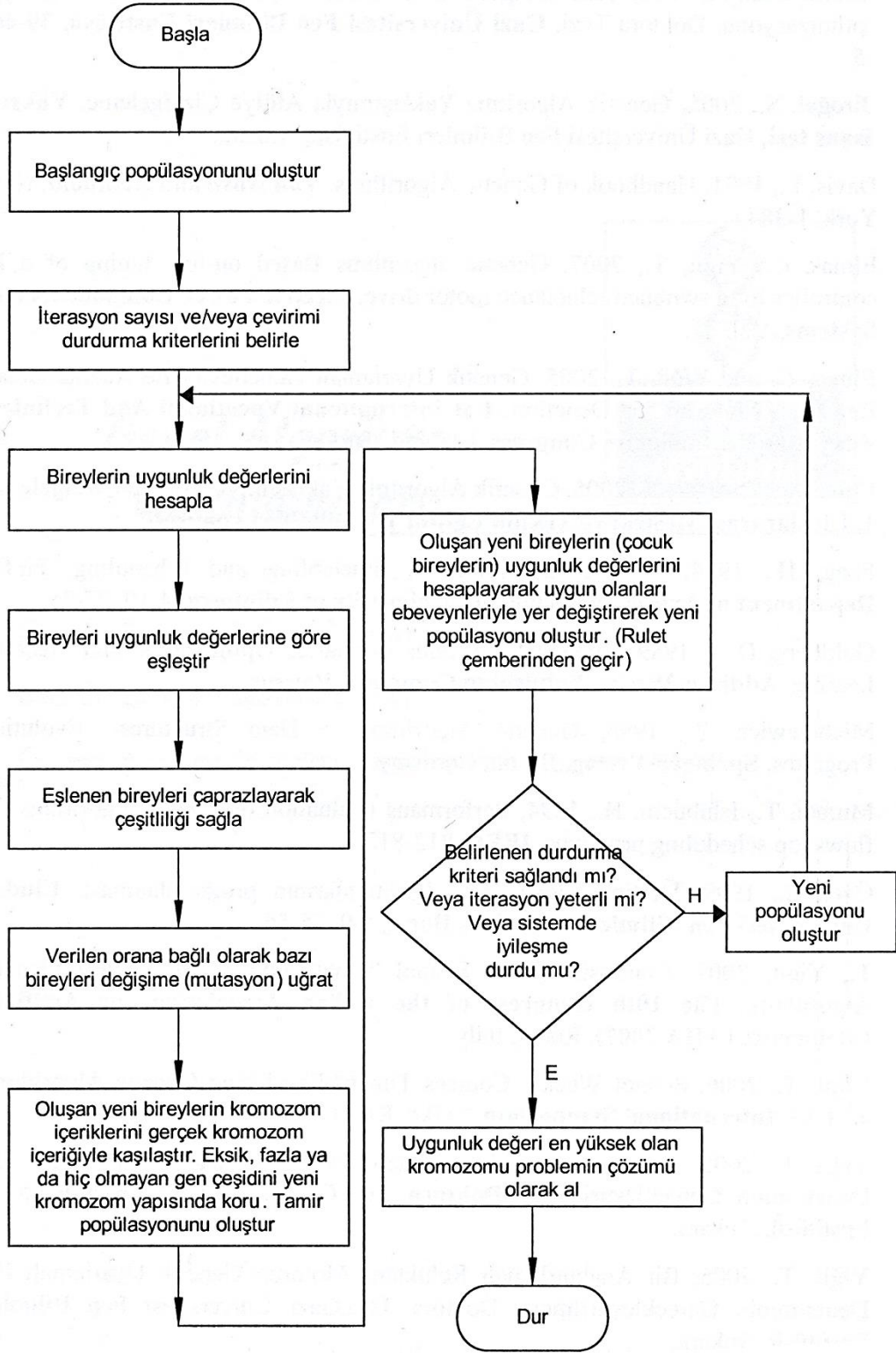
Mutasyon öncesi	Mutasyon sonrası
2 5 7 3 4 2 6 2	2 5 2 3 7 2 6 4

Genetik algoritmanın detaylı işleyiş biçimi Şekil 3.3'te gösterilmektedir.

3.3 Genetik Algoritma Parametreleri

Genetik algoritmanın performansını etkileyen bazı parametreler mevcuttur. Bu parametrelerin uygulamaya göre doğru bir şekilde ayarlanması problem için optimum çözümü bulmada çok etkilidir. Bu parametreler popülasyon büyüklüğü, çaprazlama olasılığı ve mutasyon olasılığıdır [53], [59], [60]. Şimdi kısaca bu parametrelere değinelim.

- a) **Çaprazlama olasılığı:** Bu olasılık değeri seçim işleminden sonra hangi sıklıkla ebeveyn kromozomlar arasında çaprazlama işlemi yapılacağını belirtir. Eğer çaprazlama olasılığı %100 olursa seçilen ebeveynler kesinlikle çaprazlama işlemine uğrayacaktır. Eğer algoritmada seçkinlik işlemi de yapılmıyorsa yeni popülasyona eski popülasyondan kromozom aktarımı kesinlikle olmayacaktır. Çaprazlama olasılığı çok yüksek tutulursa iyi kromozomlar



Şekil 3.3 - Genetik algoritmanın detaylı işleyiş akış şeması [54]

sürekli yok olacaktır ve algoritmanın ortaya çıkaracağı sonuç vasatın altında olacaktır. Olasılığı çok düşük tutmak da az sayıda yeni kromozom oluşturmaya neden olacaktır. Bu da çeşitliliği azaltacak ve optimum sonuca ulaşmamızı engelleyecektir.

b) Mutasyon olasılığı: Bu olasılık değeri çaprazlama işleminden sonra hangi sıklıkla yeni oluşan kromozom üzerinde mutasyon gerçekleştirileceğini belirtir. Eğer oran %100 olursa çaprazlamadan sonra kromozom muhakkak mutasyona uğrayacaktır. Mutasyon olasılığı çok yüksek olursa kromozomların yapısı çok değişecektir ve rastgele kromozom üremeye başlayıp gerçek sonuca ulaşılmasını engelleyecektir.

c) Popülasyon büyüklüğü: Popülasyonda bulunan kromozomların sayısı o popülasyonun büyüklüğünü göstermektedir. Popülasyon içindeki kromozom sayısı az tutulursa çeşitlilik az olacağından optimum sonucu bulmak zor olacaktır. Popülasyon büyüklüğü çok olursa bu sefer algoritma yavaşlayacaktır ve verimliliği çok düşecektir.

Çizelge 3.11’de farklı araştırmacıların yukarıdaki parametrelere önerdikleri değerler gösterilmektedir.

Çizelge 3.11 - Farklı araştırmacılar tarafından önerilen genetik algoritma parametre değerleri [53]

Parametreler	Negnevitsky	D. Jong	Michalewicz	Schaffer	Grefenstette
Popülasyon büyüklüğü	50	50-100	50-100	20-30	30
Çaprazlama oranı	0,7	0,6	0,5-1,0	0,75-0,95	0,95
Mutasyon oranı	0,001-0,01	0,001	0,001-0,01	0,005-0,01	0,01

4. KABLOSUZ ALGILAYICI AĞLARDA KÜMELEME OPTİMİZASYONU

Kablosuz algılayıcı ağlarda kümeleme yönteminin kullanılması, ağda çok az sayıda algılayıcı düğümün uzak mesafedeki baz istasyonuna veri iletmesini sağlayıp, çok sayıda düğümün kısa mesafelerde iletişim yapmasını sağlayarak algılayıcıların enerji tüketimini azaltıp ağ yaşam süresini arttırmaktadır. Ancak yüzlerce algılayıcı barındıran bir ağda kümeleme işleminin en verimli şekilde yapılması çok zor bir işlemdir (NP-zor problem). 100 algılayıcılı bir ağı örnek alırsak tüm olası çözümleri aramak Denklem (4.1)'in sonucu kadar farklı kombinasyon gerektirecektir ve bu problemin çözümü günümüzdeki bilgisayarlarla bile çok çok zordur [61].

$$c_{100}^1 + c_{100}^2 + \dots + c_{100}^{100} = 2^{100} - 1 \quad (1)$$

İkinci bölümde anlatılan kümeleme yöntemlerinde genellikle önceden belirlenen sabit sayıda küme başı fikri vardır ve algılayıcılar ağda kendi kendilerini organize ederler. Bu gibi durumlarda en uygun kümelerin ve küme liderlerinin bulunması çok güçtür. Bunun yerine en uygun küme ve küme başlarının sayısını bulan ve organizasyon işlemini baz istasyonunun yaptığı bir yöntem çok daha faydalı olacaktır. İşte bu noktada devreye sezgisel arama algoritmaları girmektedir.

4.1 Genetik Algoritma ile Optimum Kümelerin Bulunması

Üçüncü bölümde anlatıldığı gibi genetik algoritma birçok NP-zor (en iyi çözümleri bulmanın imkansız olduğu problemler) probleme uygulanarak optimum çözümleri bulabilen etkili bir arama algoritmasıdır. Bu kısımda genetik algoritmanın bu etkili arama işleminin kablosuz algılayıcı ağlarda en uygun kümeleri bulması için nasıl uygulandığı irdelenecektir.

4.1.1 Genetik algoritma kullanılarak algılayıcı ağ optimizasyonu

Uygun küme başlarının ve kümelerin bulunması için mesafenin en aza indirgenmesi çok önemlidir. Genetik algorithmada kablosuz algılayıcı ağın temsil edilmesi için ikili kodlama kullanılır. Bu kodlamada her bit bir algılayıcı düğümü temsil eder. Kodlamadaki “1” ilgili düğümün küme başı olduğunu, “0” ise normal düğüm olduğunu ifade eder [61].

Çizelge 4.1 - Kablosuz algılayıcı ağın ikili kodlamayla temsili

a1	a2	a3	a4	a5	a6	a7	a8	a9
1	0	0	1	0	0	1	0	0

Çizelge 4.1’de a1, a4 ve a7 algılayıcı düğümleri küme liderleridir. Geri kalanları da normal düğümlerdir. Başlangıç popülasyonunda kromozomlar rastgele oluşturulur. Bu rastgele oluşturulan kromozomlarda (ağ yapılandırması) bilindik protokol işlemleri yapılır. Yani küme başı olmayan algılayıcı düğümler kendilerine en yakın küme başlarının kümesine dahil olur ve iletişim başlar. Her kromozomun ne kadar iyi bir ağ yapılandırması oluşturduğunu anlamak için genetik algoritmanın uygunluk fonksiyonundan yararlanır. [61]’deki çalışmada kullanılan uygunluk fonksiyonu Denklem (4.2)’de gösterilmiştir.

$$Uygunluk = w * (D - mesafe_i) + (1 - w) * (N - H_i) \quad (2)$$

Bu uygunluk fonksiyonunda D , tüm algılayıcı düğümlerin baz istasyonuna olan mesafelerinin toplamıdır. $Mesafe_i$ kümelerdeki normal algılayıcı düğümlerin küme başına olan mesafelerinin toplamıyla küme başlarının baz istasyonuna olan mesafelerinin toplamının toplamıdır. H_i küme başlarının sayısıdır. N toplam algılayıcı sayısıdır ve w da önceden belirlenmiş bir ağırlık değeridir. Bir ağ için $mesafe$ ve H parametreleri dışındaki diğer parametreler sabit değerlidir. Mesafe ne kadar kısa

olursa küme başlarının sayısı o kadar az olacak ve uygunluk fonksiyonunun değeri o kadar yüksek olacak. Bu çalışmadan kullanılan genetik algoritma uygunluk fonksiyonunun değerini maksimize etmeye çalışmaktadır.

W ağırlık değeri $0 \leq w \leq 1$ arasında bir değerdir ve uygulama bağımlıdır. Hangi parametrenin uygunluk hesaplamasında daha baskın olacağını ayarlar. Örneğin $w=1$ olursa ağ yalnızca iletişim mesafesi açısından optimize edilir. Eğer $w=0$ olursa yalnızca küme başı sayısı açısından optimize edilir.

Her kromozomun uygunluğu hesaplandıktan sonra sıra seçim işlemine gelir. [61]'deki çalışmada rulet tekeri seçim işlemi uygulanmıştır. Yüksek uygunluk değerine sahip kromozomların seçilme olasılığı daha yüksektir.

Seçilen kromozomlar daha sonra çaprazlama ve mutasyon işlemlerine tabi tutularak bir sonraki nesil (popülasyon) için yeni kromozomalar oluşturulur. Çalışmada tek noktali çaprazlama uygulanmıştır (Çizelge 4.2). Mutasyon operatörü de kromozomun her bitine belli bir olasılıkla (mutasyon olasılığı) uygulanır (Çizelge 4.3).

Çizelge 4.2 - Kablosuz algılayıcı ağlarda kromozomlara çaprazlama uygulanması

Ebeveyn Kromozom 1	1 1 1 0 0 1 0 1
Ebeveyn Kromozom 2	1 0 1 1 1 1 1 0
Çocuk Kromozom 1	1 1 1 0 1 1 1 0
Çocuk Kromozom 2	1 0 1 1 0 1 0 1

Çizelge 4.3 - Mutasyon sonucu kromozomda iki bit değer değiştirmiştir

Mutasyon öncesi yeni kromozom	1 0 1 1 0 1 0 1
Mutasyon sonrası yeni kromozom	1 1 1 0 0 1 0 1

Bu işlemler sonucunda yeni ağ yapılandırmalarına (kromozomlar) sahip yeni bir popülasyon oluşturulur. Her kromozomun temsil ettiği ağ yapılandırması simüle edilerek uygunluk değeri hesaplanır ve aynı işlemler belirlenen nesil sayısına kadar devam eder. [61]'de yapılan simülasyonda 100 algılayıcı düğüm kullanılmıştır ve baz istasyonu alanın (0, 0) ve (200, 200) noktalarına konarak iki farklı durumda ağ test edilmiştir.

Simülasyon sonuçlarında yöntemin hızlıca sonuca ulaştığı görülmüştür. 100 düğümü olan bir ağda iyi bir sonuç en erken 120 nesil sonra bulunmaktadır. Baz istasyonu ağın ortasına konumlandığı zaman ağın köşesine konumlandığı durumdakinden daha çok küme başına ihtiyaç duyulmaktadır. Yoğun olarak konuşlandırılmış bir bölgede çoğunlukla ortalarda bulunan algılayıcılar küme başı olarak seçilmektedir.

4.1.2 Genetik algoritma kullanılarak algılayıcı ağların kendi kendine organizasyonu

Bu çalışmada çok atlamalı (multi-hop) algılayıcı ağların optimizasyonu için karmaşıklığı azaltılmış bir genetik algoritma önerilmektedir [62]. Bu sistemin amacı da alana rastgele dağıtılmış algılayıcıların oluşturduğu tasarsız topolojide en uygun sayıda küme ve küme başlarını bulmak ve bunu yaparken de tek veya çok atlama kullanarak baz istasyonuna giden maliyeti en az iletişim yolunu bulmaktır. Ancak bu yöntemde ağ yapılandırması görevi sadece baz istasyonu tarafından yapılmamaktadır. Baz istasyonu alandaki algılayıcılarla ortak çalışarak genetik algoritmaya parametre üretmektedir. Ayrıca yöntemde iki farklı genetik algoritma kullanılmaktadır. Birincisi algılayıcı düğümlerin seçimi yani görev dağılımları için kullanılmaktadır. İkincisi de küme başları, kümeler arası yönlendiriciler arasında en uygun yolu bulmak için kullanılmaktadır. Her iki algoritma için farklı uygunluk fonksiyonu vardır.

Genetik algortmada kullanılan kodlama sistemi bir önceki çalışmadaki sistemden biraz farklıdır. Bu çalışmada algılayıcı düğümler 3 bit ile temsil edilmektedir.

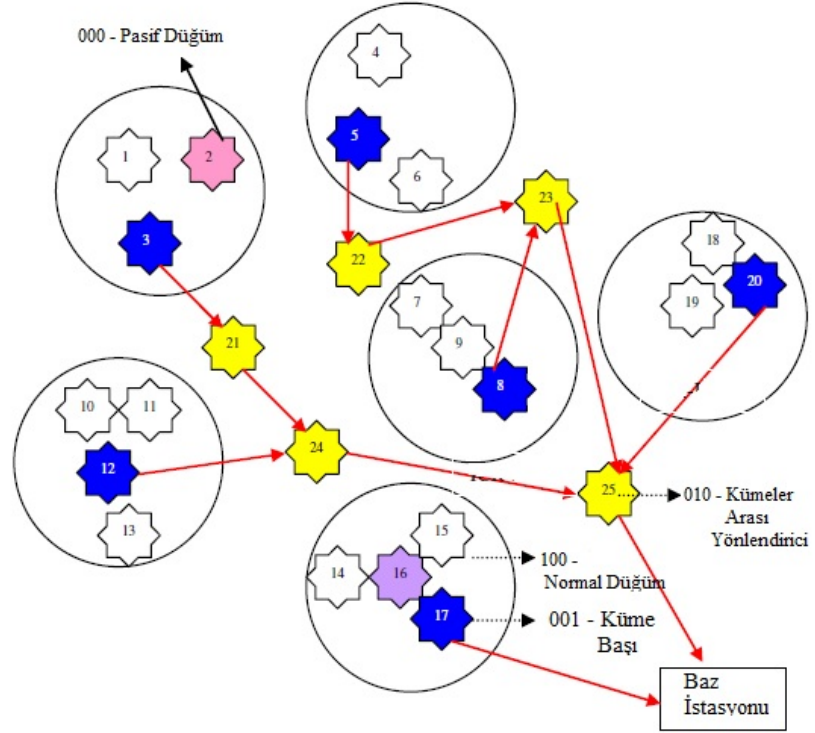
000 – Aktif olmayan düğüm (kapatılmış)

001 – Küme başı olan düğüm

010 – Kümeler arası yönlendirici düğüm

100 – Normal düğüm

Her küme bir küme başı, aktif ya da pasif olan küme üyeleri ve kümeler arası yönlendiriciler ile temsil edilir. Küme liderleri üye düğümlerden verileri toplayıp sıkıştırmakla sorumludur. Kümeler arası yönlendiriciler de küme liderlerinden aldıkları veriyi baz istasyonuna iletmekle sorumludurlar (Şekil 4.1).



Şekil 4.1 - Genetik algoritma sonucu oluşturulmuş kümeler ve algılayıcıların görev dağılımı [62]

Simülasyonda farklı sayıda algılayıcıların oluşturduğu farklı ağlarda yöntem denenmiştir. Sonuçlarda amaçladıkları enerjiyi maksimize eden optimum sonuca ulaşmışlardır.

4.1.3 Dağıtık genetik algoritma kullanılarak algılayıcı ağlar için yaşam süresi bilinçli kaynak yönetimi

Bu çalışmada kablosuz algılayıcı ağlarda kaynak yönetimi ile alakalı bir çalışma yapılmıştır [63]. Kaynak yönetimi, farklı işleme elemanlarına görev ataması yapan, başlangıç vakitlerini zamanlayan, enerji dağılımı ve iletişim bant genişliği gibi kaynakların kullanımını belirleyen servis kalitesi seviyesine karar veren süreç olarak tanımlanmaktadır. Kaynak yönetim problemi çok hedefli bir optimizasyon problemi olarak formüle edilebilir. Kazancı maksimize ederken maliyeti minimize etmek gibi söylenebilir.

Bu çalışmada çevre kirliliği, sel baskını ve yangın gibi fiziksel süreçlerin tahmini, modellenmesi ve gözlemlenmesi için kullanılan bir çevre gözleme algılayıcı ağında enerji kaynağının yönetimine odaklanılmıştır. Ağda farklı tipte durumları algılayabilen düğümler mevcuttur.

Problemin optimum çözümünü bulabilmek için dağıtık genetik algoritma kullanılmıştır. Yani genetik algoritmanın işlem yükü birden fazla işlemciye paylaştırılmıştır. Her işlemci kendine ayrılan popülasyon üzerinde eş zamanlı olarak çalışmaktadır. Bu da genetik algoritmanın sonuç üretmesini oldukça hızlandırmaktadır. Genetik algortmada kullanılan kromozomlar n tane sembolden oluşmaktadır. n de kümedeki görevlerin toplam sayısıdır. Her bir görevin bir tek algılayıcı düğüm tarafından seçilebileceği varsayılmıştır. Çünkü aynı görevin bir kereden fazla çalıştırılması gereksiz bilgi üretimine neden olacaktır. Eğer j . görev x düğümü için ayrılmışsa o zaman kromozomun j . genin değeri x olacaktır.

Eğer ilgili kaynak yönetim şeması kullanıcının belirlediği tespit eşik değeriyle uyuşmuyorsa o zaman o şemayı temsil eden kromozomun uygunluk değeri 0 olur. Aksi halde uygunluk değeri düğümlerin geriye kalan en az yaşam süresine eşit olur. Algoritmada tek noktalı çaprazlama kullanılmıştır. Mutasyon olasılığı %1 olarak ayarlanmıştır. İşlemcilere pay edilen popülasyonlar rastgele sırada başlatılıp değerlendirilir ve önceden belirlenmiş nesil sayısı kadar birbirlerinden bağımsız şekilde evrilirler.

Genetik algoritmanın performansını değerlendirmek için C++ ile bir simülasyon geliştirilmiştir. Simülasyonda 10 tane algılayıcı düğüm ve 100 tane görev tanımlanmıştır. Algılama olasılığı ve her görevin güç tüketimi rastgele oluşturulmuştur. Farklı varyasyonlarla farklı dağıtımlar test edilmiştir. Alınan sonuçlarda geliştirilen genetik algoritmanın ağ yaşam süresini %14.4 oranında arttırdığı görülmüştür.

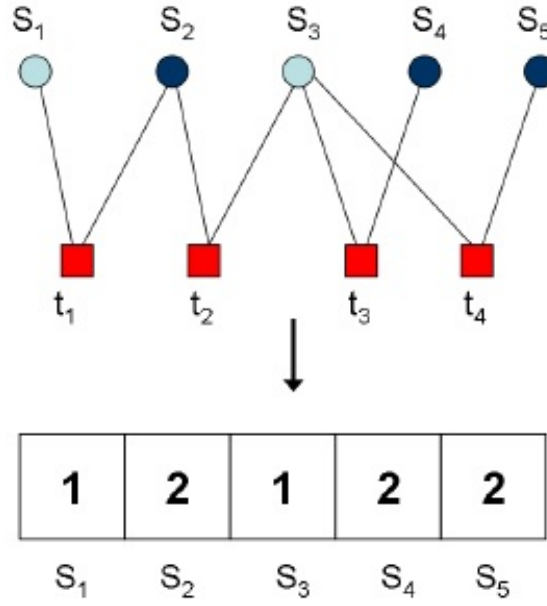
4.1.4 Büyük çaplı gözetim uygulamalarında kablosuz algılayıcı ağın yaşam süresini geliştirmek için etkili bir genetik algoritma

Bu çalışmada ayrık hedeflerin algılayıcılar tarafından tamamen kapsanması gerektiği bir ayrık algılayıcı gözetim uygulaması düşünülmüştür [64]. Eğer bir hedef artık kapsanamıyorsa sistem güvenilir hale gelecektir. Örneğin savaş sahasında ilgilenilen her konum en azı bir algılayıcı tarafından kapsanmak zorundadır. Aksi takdirde önemli bilgi eksiklikleri meydana gelebilir. Bu nedenle bu çalışmadaki amaç bütün hedeflerin algılayıcılar tarafından kapsandığı maksimum sayıdaki ayrık algılayıcı kapsamalarını bulmaktır. Bu problem literatürde Ayrık Küme Kapsamaları (Disjoint Set Covers – DSC) problemi adı altında çözümlenmektedir. Bu problem NP-zor problem olduğu için en iyi çözümü bulmak imkansızdır. Bu yüzden yaklaşık en uygun (optimum) çözümü bulmak için sezgisel yöntemler kullanılmıştır.

Çalışmada DSC problemini çözmek için genetik algoritma tabanlı bir çözüm önerilmiştir. Ayrık kapsamaları bulmanın sezgisel bir yolu her algılayıcının önceden belirlenmiş gruplara rastgele dahil olmasıdır. Bir grup eğer bütün hedefleri kapsayabiliyorsa bir kapsama oluşturur. Bu fikre dayanarak algılayıcıların gruplama kombinasyonunu kodlamak için tam sayı gösterimi kullanılmıştır. bir genin değeri algılayıcının katıldığı grubun indeksini göstermektedir. Şekil 4.2’de S_1 algılayıcısı grup 1’e katılmıştır, S_2 algılayıcısı grup 2’ye katılmıştır. Her genin değeri 1’den başlayarak önceden belirlenmiş grup sayısı aralığında değişebilir.

Kromozomun uygunluk değeri bulunan ayrık kapsamalar sayısı ile ifade edilmektedir. Bir algılayıcı grubun bir kapsam oluşturup oluşturamadığını kontrol etmek için gruptaki her algılayıcı sırayla kontrol edilir ve bütün hedeflerin dahil edilip

edilmediği kontrol edilir. Şekil 4.2'deki kromozomu göz önüne alırsak 1. ve 2. gruplar bütün hedefleri kapsamaktadır. Yani kromozomun uygunluk değeri 2'dir. İki tip simülasyon gerçekleştirilmiştir. Birincisi algılama çapını ayarlayan simülasyondur. İkincisi de algılayıcıların sayısını kontrol eden simülasyondur. Alınan sonuçlarda klasik yöntemden %16 daha iyi kapsama gerçekleştirildiği gözlemlenmiştir.



Şekil 4.2 - Algılayıcı grupları için kromozomun tam sayı olarak temsil edilmesi [64]

4.1.5 Hiyerarşik kablosuz algılayıcı ağlar için genetik algoritma

Bu çalışmada ikinci bölümde anlattığımız HCR protokolünün [37] genetik algoritmayla optimize edilmesidir [65]. Belirlenmiş iletim sayısı için enerji verimli kümeleri bulabilmek için genetik algoritma kullanılmıştır. Kromozomların kodlanması [61]'de olduğu gibidir. Bu yöntemde baz istasyonu algılayıcıların konum bilgisine sahiptir. Bütün ağ yapılanmasını baz istasyonu yapmaktadır. Genetik algoritma sonucu elde edilen en uygun yapılandırmayı algılayıcı düğümlere yayınlar. Bu yayın mesajında hangi düğümlerin o anki turda küme başı olacağı, küme başı olmayan diğer düğümlerin hangi kümeye dahil olacakları yer almaktadır.

Bu yöntemde kullanılan uygunluk fonksiyonu diğerlerinkinden daha farklıdır. Kullanılan uygunluk parametreleri şu şekildedir:

- *Baz İstasyonuna Doğrudan Mesafe (DD)*: Bu mesafe bütün algılayıcı düğümlerin baz istasyonuna olan mesafelerinin toplamıdır. Özellikle geniş ağlar için bu değer minimize edilmesi gerekmektedir.
- *Küme Mesafesi (C)*: Bu mesafe üye düğümlerden küme başına olan mesafelerin toplamıyla küme başından baz istasyonuna olan mesafenin toplamıdır.
- *Küme mesafesi standart sapması (SD)*: Algılayıcı düğümlerin alana tek tip (uniform) olarak dağıtıldığı durumlar için küme mesafelerinin aralarındaki farklılık az olmalıdır. Bununla beraber tek tip olmayan, yani rastgele dağıtılmış durumlarda küme mesafeleri birbirlerine yakın değerlerde olmak zorunda değildir. Eğer algılayıcı dağılımı tek tipse küme mesafelerinin arasındaki farklılığın büyük olması zayıf ağ yapılandırmasına neden olacaktır. Bu nedenle tek tip dağılım olan uygulamalarda bu farklılığın göz önünde bulundurulması gerekmektedir.
- *İletim Enerjisi (E)*: Bu parametre üye düğümlerin algıladıkları veriyi küme başına gönderirken harcadıkları enerji, küme başının bu verileri alırken harcadığı enerji ve yine küme başının aldığı veriyi baz istasyonuna iletilirken harcanan enerji miktarının toplamını temsil etmektedir.
- *İletim Sayısı (T)*: Bu değer baz istasyonu tarafından her bir veri aktarım safhasında atanan bir sayıdır. Ağ durumuna ve o anki enerji seviyelerine göre ayarlanabilir.

Bu parametreler göz önüne alınarak oluşturulan uygunluk fonksiyonu Denklem (4.3)'te gösterilmiştir.

$$F = \sum_i (w_i * f_i), \forall f_i \in \{C, DD, E, SD, T\} \quad (4.3)$$

Başlangıçta her parametreye rastgele w_i ağırlık değeri atanır. Daha sonra her nesilde en iyi kromozom elde edilir ve uygunluk parametrelerinin ağırlık değerleri Denklem (4.4)'deki gibi güncellenir:

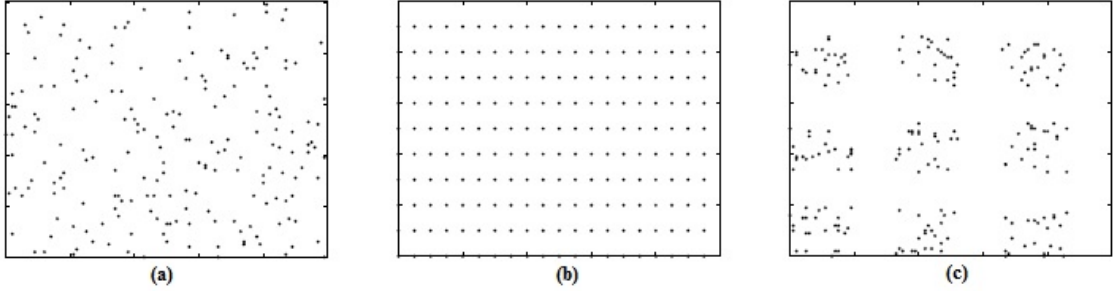
$$\Delta f_i = f_i - f_{i-1} \quad (4.4)$$

Δf_i ifadesi uygunluk parametresinin deęerindeki deęiřimi gsterir.

$$w_i = w_{i-1} + c_i \Delta f_i \quad (4.5)$$

Denklem (4.5)'te $c_i = \frac{1}{1+e^{-f_i}}$ dir. Bu deęer aęırlık deęerlerini nceki durumlara bakarak geliřtirir.

nerilen yntemi gereklemek iin Java programlama diliyle nesne tabanlı programlama teknikleri kullanılarak bir simlatr geliřtirilmiřtir. Simlasyonda  tip algılayıcı daęıtım dzeni ele alınmıřtır. Rastgele dzen, tek tip ızgara dzeni ve kme ızgara dzeni (řekil 4.3).



řekil 4.3 - (a) Rastgele dzen (b) tek tip ızgara dzeni (c) kme ızgara dzeni [65]

Simlasyonda 200 algılayıcı dğmn bulunduęu, baz istasyonunun aędan 200 m uzakta olduęu 100 m x 100 m lik bir alan varsayılmıřtır ve  farklı daęılım dzeninde denemeler yapılmıřtır. Sonular [33], [37] ve [61] ile karřılařtırılmıřtır. Karřılařtırma sonucunda bu yntemin dięerlerine gre aę yařam sresini arttırdıęı grlmřtir.

4.1.6 İki katmanlı algılayıcı ağlarda enerji verimli yönlendirme için genetik algoritma tabanlı bir yaklaşım

Bu çalışmada ağ yaşam süresini uzatmak için iki katmanlı algılayıcı ağlarda küme başı olarak yüksek enerjili taşıyıcı düğümlerin (küme başları) kullanılabilceği ileri sürülmüştür [66]. Taşıyıcı düğümler verinin baz istasyonuna doğru yönlendirme için kendi aralarında bir ağ oluşturabilirler. Çalışmada taşıyıcı düğümlerin verileri toplamasını çizelgelemek için genetik algoritma tabanlı etkili bir çözüm geliştirilmiştir.

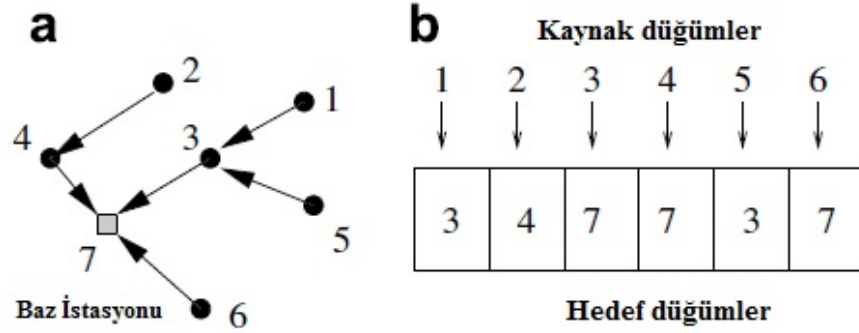
Taşıyıcı düğümlerden yapılan veri transferi tek atlamalı olabileceği gibi çok atlamalı da olabilir. Özellikle geniş ağlar için çok atlamalı iletişimin kullanılması kaçınılmazdır. Aksi halde taşıyıcı düğümlerin enerjileri erken tükenecektir. Çok atlamalı model ikiye ayrılmaktadır:

- *En az iletim enerjisi modeli:* Bu modelde her taşıyıcı düğüm en yakın komşu taşıyıcı düğüme iletimi gerçekleştirir. Komşu olan taşıyıcı düğüm baz istasyonuna daha yakındır.
- *En az atlamalı yönlendirme modeli:* Bu modelde her taşıyıcı düğüm baz istasyonuna doğru atlama sayısını en aza indirgeyen bir yol bulur.

Bu yöntemde de ağdaki algılayıcıların sabit olduğu ve konumlarının bilindiği öngörülmektedir. Taşıyıcı düğümler için geliştirilecek en uygun yol hesabını enerji kısıtlaması olmayan merkezi birim, yani baz istasyonu yapmaktadır. Genetik algoritmanın sonucunda elde edilen taşıyıcı düğümler arasındaki en uygun yol bilgisi baz istasyonu tarafından ağdaki bütün algılayıcılara yayınlanır. Bu yayında algılayıcılar tarafından harcanan enerji çok az olduğundan ağın yaşam süresine neredeyse hiç etkisi yoktur ve bu nedenle önemsenmez. Yayın bilgisi alındıktan sonra her algılayıcı düğüm hangi taşıyıcı düğümün kümesine dahil olacağını bilir. Taşıyıcı düğümlerde yönlendirme tablolarını günceller ve artık ağ iletişime hazırdır.

Bu çalışmadaki genetik algorithmada kullanılan kromozom yapısı sadece ağdaki taşıyıcı düğümleri temsil etmektedir. Diğer normal düğümler kromozomlara dahil edilmemiştir. Yani kromozomların uzunluğu taşıyıcı düğümlerin sayısına eşit-

tır. Kromozomlar ikili kodlamayla değil düğümlerin sayısını gösteren bir dizi şeklinde kodlanmıştır. Şekil 4.4a’da 6 taşıyıcı düğüm ve baz istasyonu için bir yönlendirme şeması gösterilmekte. Şekil 4.4b’de bu şemayı temsil eden kromozom yapısı gösterilmektedir. Kromozomu açıklarsak; birinci sıradaki genin değeri 3’tür. Yani 1 numaralı taşıyıcı düğüm 3 numaralı taşıyıcı düğüme iletim yapacaktır. Benzer şekilde üçüncü sıradaki genin değeri 7’dir. Yani 3 numaralı taşıyıcı düğüm 7 numaralı düğüme (baz istasyonuna) iletim yapacaktır. Buradan anlaşıldığı üzere baz istasyonu da sanki bir düğüm gibi kromozoma dahil edilmiştir.

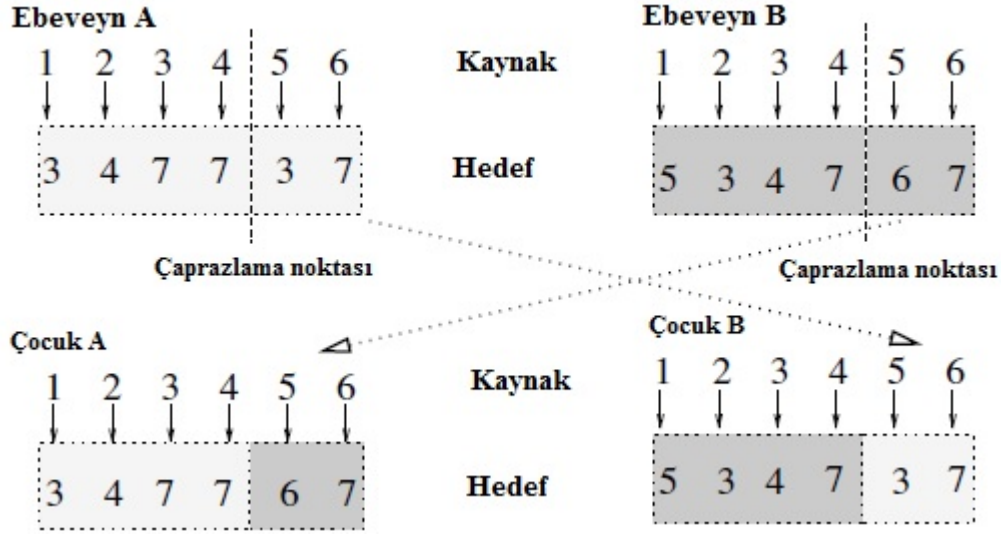


Şekil 4.4 - Algılayıcı ağıın graf ve kromozom olarak temsili [66]

Kromozomun uygunluk değeri Denklem (4.6)’daki gibi hesaplanmaktadır.

$$F = E_{initial}/E_{max} \quad (4.6)$$

Bu uygunluk denkleminde $E_{initial}$ taşıyıcı düğümün başlangıç enerji seviyesidir ve bütün algılayıcı düğümlerin başlangıçtaki enerji seviyeleri önceden bilinmektedir. E_{max} da bir turda kromozomdaki en fazla enerji harcayan taşıyıcı düğümün harcadığı enerji miktarıdır. Enerji harcamasını hesaplamak için [33]’deki radyo modeli kullanılmaktadır. Şekil 4.5’de bu çalışmada yapılan çaprazlama işleminin bir örneği gösterilmiştir.



Şekil 4.5 - Numaralı kromozomlarda çaprazlama örneği [66]

Simülasyonda farklı durumlar denenerek yukarıda bahsedilen iki farklı çok atlamalı model ile karşılaştırmalar yapılmıştır. Çok atlamalı modellerde sezgisel bir yöntem (genetik algoritma gibi) kullanılmamıştır. Düğümlerin sayısı 100 ile 3000 arasında değiştirilmiştir. Baz istasyonunun konumu ağın sol alt köşesi ve ağın tam ortası şeklinde değiştirilmiştir. Taşıyıcı düğümlerin sayısı 10 ile 60 arasında değiştirilmiştir. Alınan sonuçlarda ağın yaşam süresinin diğer yöntemlere göre çok daha uzatıldığı gözlemlenmiştir.

4.1.7 Hareketli tasarsız ağlarda kümelemeyi optimize etmek için genetik algoritma kullanımı

Bu çalışmada önerilen yöntem gezici tasarsız ağlarda ağırlıklı kümeleme algoritması üzerine yapılandırılmıştır [67]. Diğer çalışmalardan temel farklılığı düğümlerin alanda sabit olmayışıdır. Ağırlıklı kümeleme algoritmasında küme başları algılayıcı düğümlerin (v) ağırlıkları temel alınarak (W_v) seçilir. Ağırlık hesaplaması Denklem (4.7)'de gösterilmiştir.

$$W_v = w_1\Delta_v + w_2D_v + w_3M_v + w_4P_v \quad (4.7)$$

Bu denklemde Δ_v , derece farkı; D_v , küme üye düğümlerinin küme başına olan mesafelerinin toplamı; M_v , düğümlerin ortalama hızı; P_v , bir düğümün küme başı olmasının toplam zamanıdır. Ağırlık değeri en düşük olan algılayıcı düğüm küme başı olarak seçilir.

Çalışmada bu yöntem genetik algoritma kullanılarak optimize edilmeye çalışılmıştır. Küme başı sayısını azaltmak için her küme başı kümesinde olası maksimum sayıda düğüm içermelidir. Düğümlerin kümeler arasında dengeli dağıtılmasıyla her bir düğümün yaşam süresi arttırılmış olmaktadır. Genetik algoritmanın hedefi o anki popülasyonda en düşük uygunluk değerine sahip kromozomu seçmektir. Bunun yanında algorithmda seçkinlik operatörü de kullanılmaktadır. Yani popülasyondaki en iyi kromozomlar kaybedilmeden bir sonraki popülasyona aktarılmaktadır.

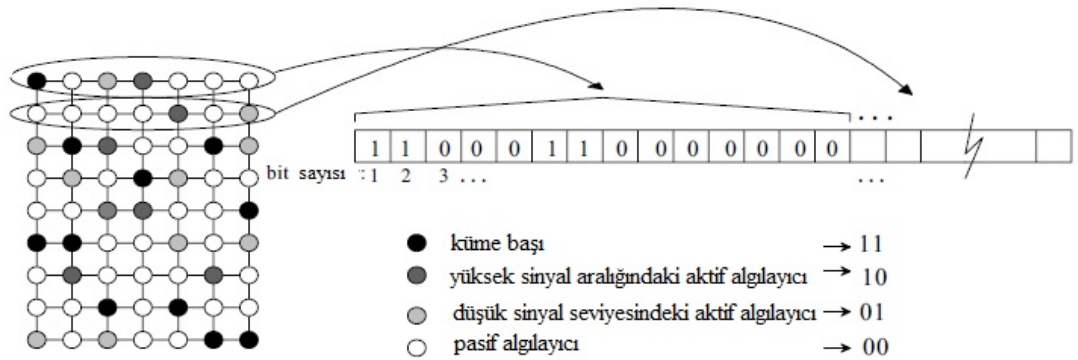
Genetik algorithmda kromozom kodlaması için tam sayılar kullanılmıştır. Kromozomun her bir geninin temsil edildiği tam sayılar ayrıca ilgili algılayıcının ID numarasıdır.

Yapılan simülasyonda 100 m x 100m'lik bir alanda 20 ile 70 arasında düğüm sayısı ile denemeler yapılmıştır. Sonuçlarda düğümler arası yük dağılımının daha dengeli yapıldığı ortaya çıkmıştır.

4.1.8 Çevresel ölçümler için kablosuz algılayıcı ağlarda enerji optimizasyonu

Bu çalışmada tarımsal bir uygulamada kullanılan kablosuz algılayıcı ağda optimum tasarımı bulabilmek için genetik algoritma kullanılmıştır [68]. Önerilen kablosuz algılayıcı ağın ana özellikleri şöyledir: 30 a 30 uzunluğunda kare bir ızgara oluşturulmuştur ve algılayıcılar bu ızgaranın 900 noktasına yerleştirilmiştir. Böylelikle bütün alan kapsanmıştır. Algılayıcılar tek tiptir ve aktif veya pasif durumda olabilirler. Bunun yanında algılayıcılar yüksek sinyal aralığında ve düşük sinyal aralığında olan algılayıcılar olarak sınıflandırılmaktadır. Eğer bir algılayıcı 10 birim uzunluğundaki çapta bir alanı kapsıyorsa yüksek sinyal aralığındadır. Eğer 5 birim uzunluğundaki çapta bir alanı kapsıyorsa düşük sinyal aralığındadır.

Genetik algılayıcıların gösterimi için iki bitlik kodlama kullanılmıştır. Pasif olan algılayıcı “00”, düşük sinyal aralığında olan aktif algılayıcı “10”, yüksek sinyal aralığında olan aktif algılayıcı “01” ve küme başı olan algılayıcı “11” olarak kodlanmıştır. Izgaradaki bütün noktalar satır satır bir dizisinde kodlanmıştır. Her gende kodlama için iki bite ihtiyaç olduğuna göre ve ızgarada da 900 tane algılayıcı olduğuna göre kromozomun bit uzunluğu 1800 olacaktır ().



Şekil 4.6 - Izgara şeklinde dağıtılan algılayıcıların iki bitlik olarak kodlanması [68]

Uygunluk fonksiyonuna etki eden faktörler içinde algılayıcıların bağımlılıkları, algılayıcıların tiplerine göre sistemin işleyiş maliyeti ve üye algılayıcıların iletişim kurduğu küme başıyla aralarındaki mesafeye dayanan iletişim maliyeti parametreleri vardır.

Algoritmada yapılan deneme yanımlar sonucunda en uygun popülasyon büyüklüğü 300 kromozom, çaprazlama olasılığı %80 ve mutasyon olasılığı %5 olarak ayarlanmıştır. Başlangıç popülasyonu rastgele oluşturulmaktadır. Uygunluk fonksiyonuyla popülasyondaki her bir kromozom değerlendirilmektedir. En iyi uygunluk değerine sahip olan kromozom seçkinlik kullanılarak bir sonraki popülasyona aktarılmaktadır. Seçim işlemi için rulet tekeri yöntemi kullanılmıştır.

Simülasyon sonucunda elde edilen sonuçlarda daha fazla algılayıcı kullanımının iletişimde daha az enerji tüketimi sağlayacağı görülmüştür. Bunun yanında ilgili ağ yapılandırmasında işlevsel ve iletişim enerji tüketimi minimize edilmiştir.

4.1.9 Genetik algoritma kullanılarak kablosuz algılayıcı ağlar için yeni bir kümeleme protokolü

Bu çalışmada kablosuz algılayıcı ağların enerji tüketimi ve yaşam süresinin optimizasyonu incelenmiştir [69]. Enerji tüketimini belirlemek için yazılım servislere bağlı olarak genetik algoritma parametreleri uyarlanmıştır.

Bu çalışmaya göre gereğinden fazla küme oluşturmak enerji tüketimini arttırmaktadır. Bu yüzden her düğüm için kullanılan enerjini ortalama miktarını elde etmek için toplam enerji tüketiminin toplam düğümlerin mesafesine oranı kullanılmaktadır. Algılayıcı ağda en uygun enerji tüketimi ve kapsamayı elde etmek için bir uygunluk fonksiyonu önerilmiştir. Fonksiyon Denklem (4.8)'deki gibidir.

$$F(i) = \left(\frac{e_i * T}{D_a * \#Düğüm} \right) * \left(\frac{e_j * T}{D_b * \#KümeBaşlı} \right) \quad (4.8)$$

$$: D_a = \frac{Genişlik * g_i}{\sqrt{\#Küme}}$$

Fonksiyondaki $((e_i * T) * (e_j * T))$ kullanılan toplam enerjidir ve $((D_a * \#düğüm) * (D_b * \#KümeBaşlı))$ da her kümedeki düğümler arası mesafenin toplamıyla küme başlarının arasındaki mesafenin toplamının çarpımıdır. g_i 'nin alabileceği değer kümesi [65]'deki uygunluk fonksiyonu parametrelerinden oluşturulmuştur. $F(i)$ fonksiyonu bu oranın olası en büyük değerini elde etmeye çalışmaktadır. D_a değişkeninde kapsama problemi yüzünden alan genişliği düşünülmüştür. "Genişlik" algılayıcıların konuşlandığı alanın uzunluğudur. D_a ve D_b değişkenleri algılayıcı üye düğümlerin ilgili küme başına olan mesafeyi ve küme başlarının baz istasyonuna olan mesafelerini göstermektedir. e_i ve e_j sabitleri de üye düğümlerden küme başına iletim için gerekli olan enerji miktarını ve küme başından baz istasyonuna iletim yapmak için gerekli olan enerji miktarını temsil etmektedir. $F(i)$ hem kümeleme tabanlı yöntemde hem de doğrudan iletim yapan yöntemde her kromozoma bir ağırlık değeri atamaktadır. Denklem (4.9)'daki fonksiyon kullanılarak $F(i)$ fonksiyonundaki değişkenlerin etkileri dengelenmektedir.

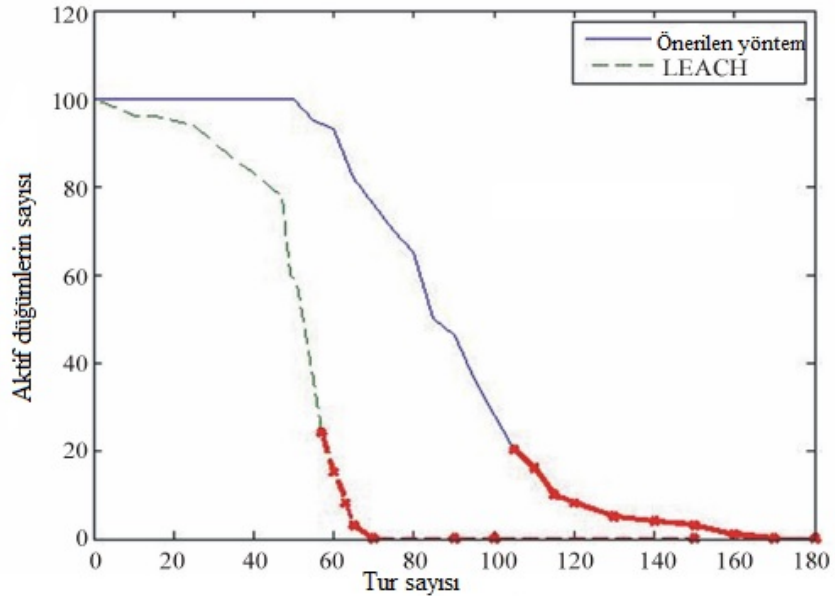
$$\forall g_i \in \{DDBS, CD, CDSD\}, [g_i = DDBS] = 1 \therefore [\#Küme = 1] = 1 \quad (4.9)$$

$$\therefore (D_a = D_b = \#KümeBaşlı = 1)$$

Uygunluk fonksiyonunda e_i ve e_j için gerekli olan enerji miktarı sırasıyla üye düğümlerin iletim sayısı ve küme başlarının iletim sayılarıyla çarpılmaktadır. Bu şekilde kromozomun uygunluğu belirlenmektedir.

Simülasyon ağ kurulum safhasıyla başlamaktadır. Bu safhada sayısı önceden belirlenmiş algılayıcı düğümlerle ağ için başlangıç değerleri ve uygunluk fonksiyonunda düşünülen sabitlerin değerleri ayarlanır. Her bir düğüme bir x ve y koordinat noktası atanır ve başlangıç enerjileri 2J olarak ayarlanmıştır. Daha sonra gelen safhalarda kümeler oluşturulur ve algılayıcı düğümler iletişime başlarlar. Küme başları aldıkları verileri birleştirerek baz istasyonuna iletirler.

Simülasyonda 200 algılayıcı düğüm 100 m x 100 m'lik bir alana dağıtılmıştır. Baz istasyonun da ağdan 200 m uzakta olduğu varsayılmıştır. Radyo modeli olarak [42]'deki model kullanılmıştır. Elde edilen sonuçlar LEACH ile karşılaştırılmıştır (Şekil 4.7). Ağ yaşam süresi tur sayısı olarak ölçülmüştür ve önerilen yöntem LEACH'e göre daha iyi sonuç üretmiştir.



Şekil 4.7 - Önerilen yöntemin tur sayısı bazında LEACH ile karşılaştırılması [69]

5. KABLOSUZ ALGILAYICI AĞLARDA GENETİK ALGORİTMA TABANLI ENERJİ VERİMLİ KÜMELER

Bu tezde yaptığımız çalışmada kablosuz algılayıcı ağların yaşam süresini optimize etmek için genetik algoritma tabanlı bir yöntem önerilmiştir (Genetic Algorithm Based Energy Efficient Clusters - GABEEC) [70]. Çalışmadaki ana amacımız gözlemlenecek olan alanın her bölgesini olabildiğince uzun süre takip etmektir. Yani enerjisi ilk bitecek olan algılayıcı düğümün yaşam süresini olabildiğince uzatmaktır.

5.1 Yöntemin İşleyişi

Yöntemimizin işleyişi [33]'te olduğu gibi iki safhaya ayrılmıştır. Bunlar kurulum safhası ve veri iletişim safhasıdır.

Kurulum safhası bir kez gerçekleştirilir. Bu safhada önceden belirlenmiş sayıda algılayıcı düğüm küme başı olarak seçilir. Küme başlarının sayısı aynı zamanda ağda bulunacak kümelerin sayısını da belirtmektedir. Küme başı olmayan algılayıcı düğümler küme başlarına olan mesafelerine bakılarak kendilerine en yakın olan küme başının kümesine dahil edilirler.

Veri iletişim safhasında bütün üye algılayıcı düğümler küme başlarıyla iletişime başlarlar. Her düğüm sırayla küme başıyla iletişime geçecektir. Küme başı kümesindeki bütün düğümlerden verileri topladıktan sonra verileri tek pakette birleştirir ve bu tek paket veriyi baz istasyonuna iletir. Bütün küme başları topladıkları veriyi baz istasyonuna iletince bir *tur* tamamlanmış olur. [51]'de olduğu gibi her turun sonunda baz istasyonu küme başlarının ve üye düğümlerin enerji seviyelerini kontrol eder. Eğer bir kümedeki küme başının enerjisi kümesindeki üye düğümlerin enerji ortalamasının altına düşerse üye düğümlerden enerjisi en fazla olan düğüm yeni küme başı olarak seçilir. Eski küme başı üye düğüm durumuna geçer.

Bu tez çalışmasında kurulum safhasında oluşturulan kümeler sonraki turlarda değiştirilmemektedir. Sadece küme başları enerji seviyelerine göre değiştirilmektedir. Yani her turda sabit kümeler ve dinamik olarak değişen küme başları vardır.

5.2 Genetik Algoritmanın Uyarlanması

Algılayıcı ağda en uygun enerji verimli kümeleri bulabilmek ve ağ yaşam süresini arttırmak için genetik algoritma kullanılmıştır. Algılayıcı ağın genetik algılamada temsili için [61]'deki gibi ikili kodlama kullanılmıştır. Başlangıç durumunda genetik algılamada önceden tanımlı sayıda rastgele genler ile oluşturulmuş kromozomlar kullanılmıştır. Daha sonra her kromozomun uygunluk değeri hesaplanır. Bir kromozomun uygunluk değeri bazı uygunluk parametrelerine bağlıdır. Popülasyondaki her kromozomun uygunluk değeri hesaplandıktan sonra rulet tekeri seçim yöntemi kullanılarak yeni popülasyon oluşturmak için ebeveyn kromozomlar seçilir. Seçimden sonra çaprazlama ve mutasyon operatörleri uygulanarak yeni çocuk kromozomlar oluşturulur. Seçim esnasında seçkinlik işlemi de uygulanır. Yani popülasyonun en iyi kromozomu yeni oluşturulacak popülasyona doğrudan aktarılır.

5.2.1 Uygunluk fonksiyonu

Uygunluk fonksiyonu bir kromozomun ne kadar iyi sonuç verdiğini değerlendirir. Bunu da bazı uygunluk parametrelerine bağlı olarak yapar. Bu tez çalışmasında üç uygunluk parametresi kullanılmıştır. Bunlar:

- R_{FND} : İlk düğümün kaçınca turda öldüğü
- R_{LND} : Son düğümün kaçınca turda öldüğü
- C : Küme mesafesi

Küme mesafesi [65]'te olduğu gibi Denklem (5.1)'de de tanımlanmaktadır.

$$C = \sum_{i=1}^k d_{ih} + d_{hs} \quad (5.1)$$

Bu denklemde d_{ih} , i üye düğümünden h küme başına olan mesafeyi; d_{hs} , h küme başından s baz istasyonuna olan mesafeyi temsil etmektedir. Yani küme mesafesi üye düğümlerin küme başına olan mesafelerinin toplamıyla küme başının baz istasyonuna olan mesafesinin toplamıdır.

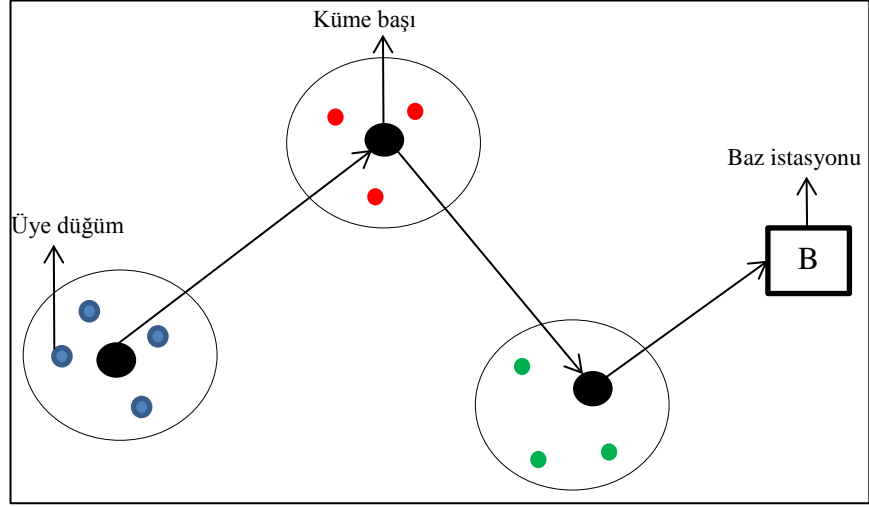
Uygunluk fonksiyonu bu üç parametrenin oluşturduğu fonksiyondur ve Denklem (5.2)'deki gibi tanımlanmaktadır.

$$F = \sum_i (f_i * w_i), \forall f_i \in (R_{FND}, R_{LND}, -C) \quad (5.2)$$

Fonksiyondaki w değeri uygulama bağımlı bir ağırlık değerini temsil eder. Yani uygulamada hangi uygunluk parametresinin daha baskın olacağını belirlemek için kullanılır.

5.3 Çok Atlamalı (Multihop) İletişim ile GABEEC'in İyileştirilmesi

GABEEC yönteminde küme başlarının baz istasyonu ile aralarındaki iletişim yolu yeniden tanımlanmaktadır. Uzun mesafelere veri paketleri yollamak enerjiyi fazla harcatmaktadır. GABEEC'de her küme lideri topladığı verileri doğrudan baz istasyonuna iletmektedir. Bu iletim küme liderlerinin erken ölmesine neden olmaktadır. Doğrudan iletmek yerine baz istasyonu, küme başları arasında mesafeyi temel olarak bir çok atlamalı (multihop) iletişim zinciri kurmaktadır (Şekil 5.1). Küme başlarının baz istasyonuna olan mesafelerine bakılarak bir zincir oluşturulur. Baz istasyonuna en uzak olan küme başı verisini kendisine en yakın küme başına gönderir. En son baz istasyonuna en yakın olan küme başı baz istasyonuna iletimi gerçekleştirir. Bu yöntemde sadece tek bir küme başı uzun mesafede iletim yapmaktadır. Bu da ağın enerjisini daha verimli kullanmayı sağlamaktadır. Bu yöntem Multihop-GABEEC (M-GABEEC) olarak isimlendirilmektedir.



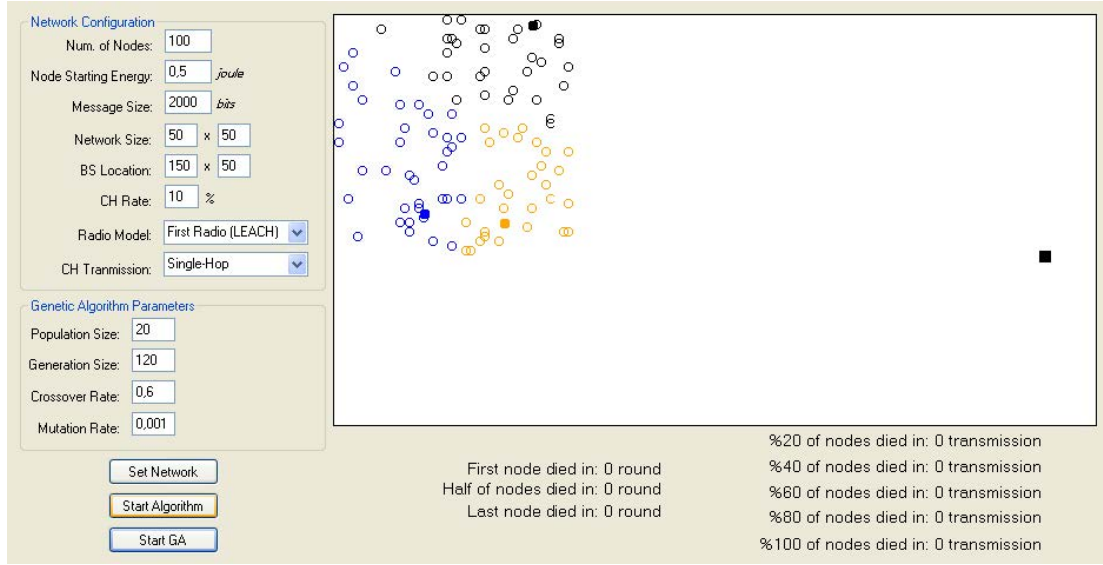
Şekil 5.1 - Çok atlamalı iletişimin temsili gösterimi

5.4 Simülasyon ve Sonuçlar

MS Visual C# 2010 geliştirme ortamı kullanılarak bir simülator geliştirilmiştir. Şekil 5.2'de simülatorün ana ekranı gösterilmektedir. İlk karşılaştırma LEACH protokolüyle yapılmıştır. 50 m x 50 m'lik alana 100 algılayıcı düğüm rastgele dağıtılmıştır ve baz istasyonu da ağdan 100 m uzağa konumlandırılmıştır. Radyo modeli olarak [33]'deki model kullanılmıştır.

Genetik algoritmanın parametreleri şu şekilde ayarlanmıştır:

- Popülasyon büyüklüğü: 20
- Nesil sayısı: 120
- Çaprazlama oranı: 0,60
- Mutasyon oranı: 0,001

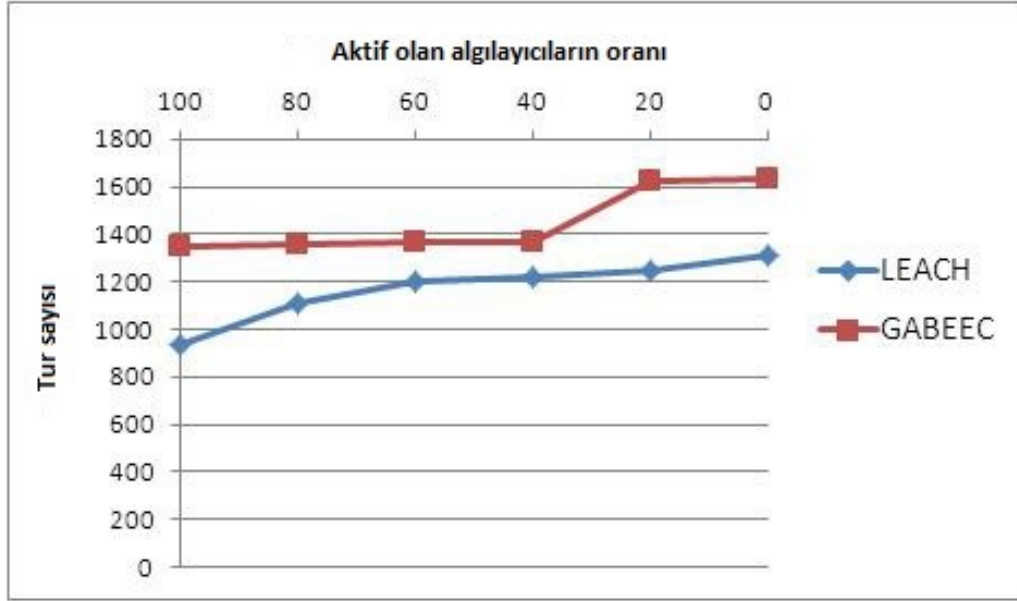


Şekil 5.2 - Geliştirilen simülâtörün ana ekran görüntüsü

Simülasyonda algılayıcı düğümler için üç farklı başlangıç (0.25J, 0.5J, 1J) enerjisi kullanılarak sonuçlar elde edilmiştir. Çizelge 5.1’de ve Şekil 5.3’te önerdiğimiz yöntemin simülasyon sonuçlarının LEACH protokolüyle karşılaştırması gösterilmektedir. Alınan sonuçlarda görülmektedir ki LEACH’e kıyasla ilk düğümün ölümü yaklaşık olarak 1,5-1,6 oranında geciktirilmiştir. Son düğümün ölümü de 1,15-1,2 oranında geciktirilmiştir.

Çizelge 5.1 - GABEEC’in farklı başlangıç enerjileriyle tur sayısı temel alınarak LEACH’le kıyaslanması

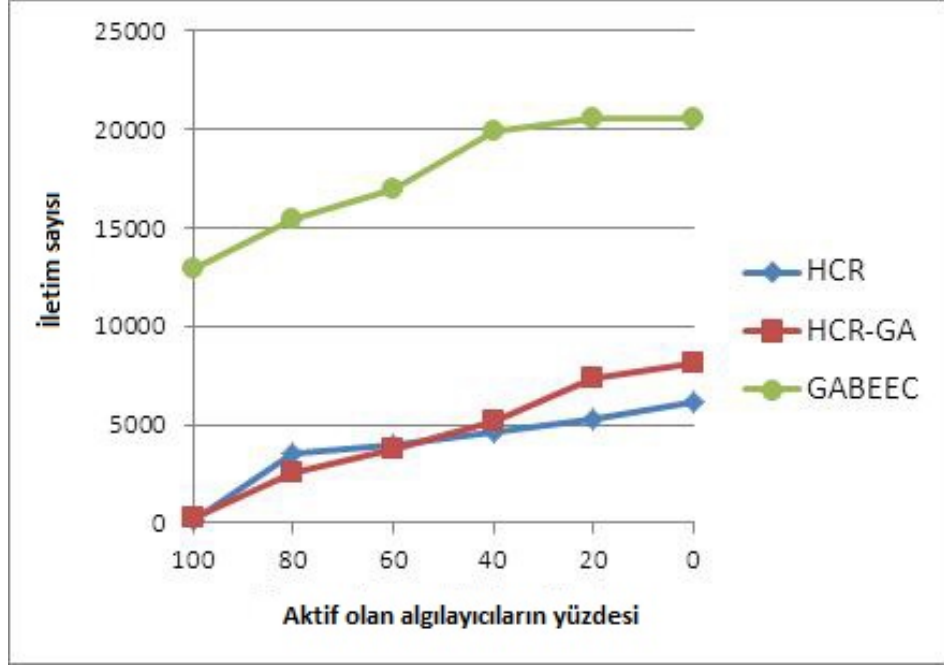
Başlangıç Enerjisi (j/node)	Metot	İlk düğümün öldüğü tur	Son düğümün öldüğü tur
0,25	LEACH	394	665
	GABEEC	651	771
0,5	LEACH	932	1312
	GABEEC	1226	1571
1	LEACH	1848	2608
	GABEEC	2776	2918



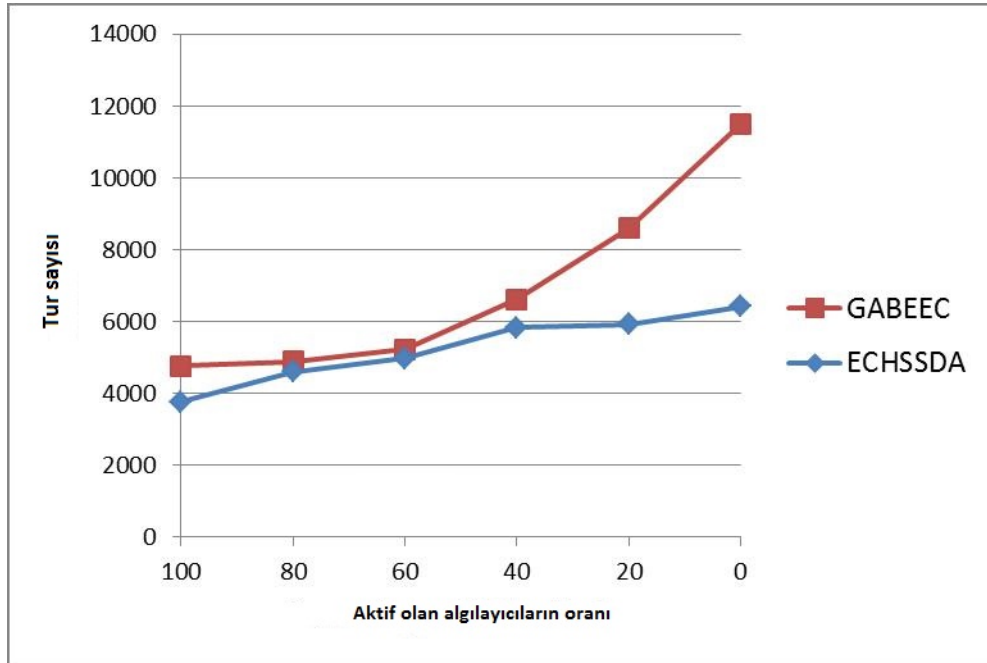
Şekil 5.3 - GABEEC ve LEACH kullanılarak 0,5J başlangıç enerjisili algılayıcı düğümlerle ağ yaşam süresi

İkinci karşılaştırma da farklı bir ağ yapılandırması kullanan [37] ve [65]'deki çalışmalarla yapılmıştır. 100 m x 100 m'lik alana 200 algılayıcı düğüm rastgele dağıtılmıştır ve baz istasyonu da ağdan 200 m uzağa konumlandırılmıştır. Radyo modeli olarak [65]'deki model kullanılmıştır. Radyo alıcı ve vericisinin çalıştırılması için $E_{elec} = 50 \text{ nJ/bit}$ enerji harcanmaktadır. İletim güçlendiricisinde kısa mesafe iletim için $\epsilon_{amp} = 10 \text{ pJ/bit/m}^2$, uzun mesafe iletim için $\epsilon_{amp} = 0,0013 \text{ pJ/bit/m}^4$ enerji harcamaktadır. Veri birleştirmek için harcanan enerji de $E_{DA} = 5 \text{ nJ/bit}$ tir. Genetik algoritma parametreleri bir önceki ile aynı şekilde ayarlanmıştır. Şekil 5.4'te simülasyon sonucunda algılayıcıların iletim sayısı temel alınarak yöntemimizin daha iyi sonuç ürettiği görülmektedir. Çizelge 5.2'de HCR-GA [65] ile yapılan karşılaştırmada iletim sayısı cinsinden alınan sonuçların detaylı dökümü vardır.

Bir diğer karşılaştırma da [65]'deki yapılandırma kullanılarak sonuçlar [51] de önerilen ECHSSDA ile karşılaştırılmıştır. Simülasyonda [33]'teki radyo modeli kullanılırken algılayıcı düğümlerin başlangıç enerjisi 2J olarak ayarlanmıştır. Şekil 5.5'de gösterilen simülasyon sonuçlarına baktığımızda GABEEC'in ECHSSDA'dan daha iyi sonuç ürettiği görülmektedir.



Şekil 5.4 - GABEEC, HCR ve HCR-GA karşılaştırması

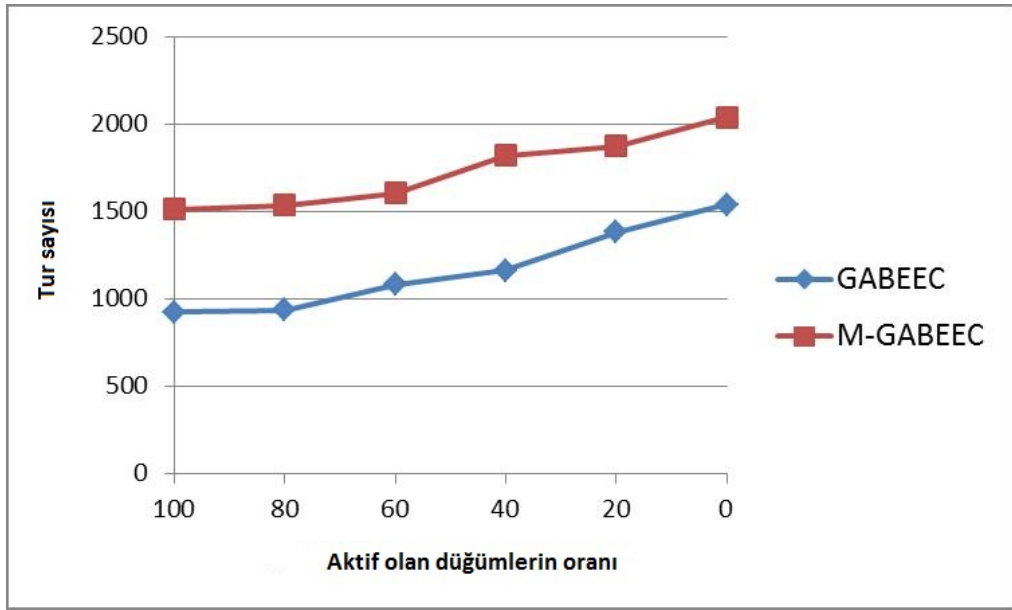


Şekil 5.5 - GABEEC ve ECHSSDA karşılaştırma grafiği

Çizelge 5.2 - GABEEC ve HCR-GA karşılaştırmasının detaylı sonuçları.

Simülasyon	HCR-GA						GABEEC					
	İlk Ölüm	%20 Ölü	%40 Ölü	%60 Ölü	%80 Ölü	%100 Ölü	İlk Ölüm	%20 Ölü	%40 Ölü	%60 Ölü	%80 Ölü	%100 Ölü
1	10980	114550	173780	191010	207430	214680	250569	254156	255146	255900	256234	256376
2	11990	98260	148210	169130	182550	188400	252642	255486	257000	257502	257799	257898
3	15980	103960	131890	159760	170480	177680	257103	260720	261704	262114	262462	262561
4	10980	114550	173780	191010	207430	214680	258588	261788	262624	263127	263500	263631
5	12990	131520	171920	187810	200090	206320	255492	258558	259473	260067	260424	260567
6	11990	103590	165040	185100	195240	201120	262647	265890	266737	267240	267550	267636
7	13980	132290	178960	193460	202870	208880	255419	260002	261102	261599	261898	262026
8	12990	92680	139870	166200	178890	182970	252549	256223	257301	257716	258068	258185
9	10980	97550	141940	159700	172900	180090	190806	193709	194407	195183	207542	207643
10	12950	107030	145500	172860	190850	194630	253143	256533	258007	258577	258829	258934
11	11980	122330	155760	174250	183600	191140	127846	130187	130864	177337	177875	177992
12	10990	97770	139920	162550	170620	183300	258390	261748	262604	263397	263734	263796
13	11990	98360	143550	170380	183930	189920	257787	261128	261919	262463	262735	262863
14	12980	128760	178120	196850	208600	216170	256697	259889	260791	261339	261645	261751
15	13990	118720	174230	188200	199240	207240	249876	253229	254496	254819	255147	255248
16	17980	125690	184000	201560	210480	217730	252153	254955	255938	256478	256795	256924
17	12980	110370	145780	173030	184990	189980	264690	267049	267813	268228	268594	268693
18	10960	115930	125750	166850	187030	198670	252840	255157	256017	256399	256690	256816
19	12990	111070	156540	168420	178940	186430	247599	250985	251672	252270	252570	252640
20	10970	121010	166410	185500	197810	204300	254294	256600	257435	257883	258258	258370
Ortalama	12681	112299,5	157047,5	178181,5	190698,5	197716,5	245556,5	248699,6	249652,5	252481,9	253417,5	253527,5
Minimum	10960	92680	125750	159700	170480	177680	127846	130187	130864	177337	177875	177992
Maksimum	17980	132290	184000	201560	210480	217730	264690	267049	267813	268228	268594	268693
Ortalama Sapma	1300	10235,5	15281,25	11686,65	11305,5	11262,5	17246,1	17350,32	17403,4	13265,57	12226,54	12230,75
Standart Sapma	1810,12	12233,95	17517,14	13094,31	13113,37	12950,95	31478,14	31668,38	31745,99	23177,85	21684,33	21683,62

Bu deęerlendirmelerden sonra GABEEC’i bu tez alıřmasında nerdięimiz ok atlamalı yntemle (M-GABEEC) karřılařtırdık. Simlasyonda 100 dęm 100m x 100 m’lik bir alana rastgele daęıtılmıřtır. Baz istasyonu aędan 50 m uzaęa konumlanmıřtır. [42]’de kullanılan radyo modeli kullanılarak simlasyon gerekleřtirilmiřtir. Őekil 5.6’daki grafik M-GABEEC’in GABEEC’den daha iyi sonu verdięini gstermektedir.



Őekil 5.6 - GABEEC ve M-GABEEC karřılařtırma grafięi

6. SONUÇ

Bu tez çalışmasında kablosuz algılayıcı ağlar alanında önemli bir yer teşkil eden kümeleme algoritmaları ve genetik algoritma açıklanmaktadır. Bu anlatım yapılırken de geçmişten günümüze literatürde var olan eski ve yeni yöntemler üzerinde durulmuştur.

Kümeleme yöntemi ağdaki algılayıcı düğümlerin bir grup mantığında ortak çalışmasını sağlayıp enerji tasarrufu sağlamak için kullanılmıştır. Tez çalışmasında bu kümeleme yöntemi üzerinde genetik algoritma kavramı kullanılarak yeni ve iyileştirilmiş bir yöntem ortaya konmaktadır.

İlk önerdiğimiz GABEEC yöntemi literatürdeki diğer çalışmalara göre ağır yaşam süresini daha uzatmaktadır. Simülasyon sonuçları göstermektedir ki genetik algoritma tabanlı bir yöntem kullanmak en uygun küme yapılandırmasını bulup yaşam süresini arttırmaktadır. Tezde önerilen genetik algoritmanın diğerlerinden farkı uygunluk fonksiyonudur. Yaptığımız çalışmada ilk düğümün ölümünü olabildiğince geciktirmeye çalıştık. Çünkü ortamın en verimli şekilde algılanması için her düğümün çalışıyor olması gerekmektedir.

Yöntemimiz rastgele düğümler oluşturularak ve bu düğümler içerisinde rastgele küme liderleri seçilerek başlar. Daha sonra bu rastgele oluşturulan yapı üzerinde genetik algoritma çalıştırılarak enerji verimini en iyi sağlayan kümeler bulunabilmektedir.

Tezde ortaya konulan bu yöntemi, küme başlarının iletişim yöntemini değiştirerek iyileştirdik. Küme başları arasında bir iletişim zinciri kurarak çok atlamalı (multihop) iletişim yapabilmelerini sağladık. Böylelikle küme başlarının da uzak mesafeye veri gönderiminin yani fazladan enerji tüketiminin önüne geçmiş olduk. Simülasyonda önerdiğimiz iki yöntemi de deneyerek sonuçlarını karşılaştırdık. Karşılaştırma sonucunda ikinci önerdiğimiz yöntemin ağ yaşam süresini birinci önerdiğimiz yöntemden çok daha fazla arttırdığını gözlemledik.

İleride bu protokole iki katmanlı bir yapı uygulanarak iyileştirme yapılması düşünülmektedir. Bunun yanında yoğun kümeler içerisinde de üye düğümler arasında çok atlamalı iletişim gerçekleştirilip daha fazla enerji tasarrufu yapılabilir.

KAYNAKLAR

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks*, cilt 38, no. 4, pp. 393-422, 2002.
2. G. R. Sakthidharan, S. Chitra, "A Survey on Wireless Sensor Network : An Application Perspective" *International Conference on Computer Communication and Informatics*, 2012.
3. W. Dargie, C. Poellabauer, "Fundamentals of Wireless Sensor Networks", Wiley, 2010, pp. 4-5.
4. <http://edageek.com/primages/2007q1/20070402c.jpg> (19.04.2013)
5. <http://www.cs.wmich.edu/wsn/images/mote.jpg> (26.04.2013)
6. http://en.wikipedia.org/wiki/Wireless_sensor_network (13.04.2013)
7. T. E. Kalaycı, "Kablosuz Sensör Ağlar, Uygulamaları", *Akademik Bilişim Konferansı*, 2009.
8. A. K. Pathan, C. S. Hong, H. Lee, "Smartening the environment using wireless sensor networks in a developing country", *The 8th International Conference on Advanced Communications Technology*, 2006.
9. S. S. Iyengar, R. R. Brooks, "Distributed Sensor Networks", Chapman & Hall/CRC, 2005, p. 9.
10. T. Arampatzis, J. Lygeros, S. Manesis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks", *13th Mediterranean Conference on Control and Automation*, 2005.
11. D. Li, K. D. Wong, Y. H. Hu, A. M. Sayeed, "Detection, classification and tracking of targets", *IEEE Signal Processing Magazine*, cilt 19, pp. 17-29, 2002.
12. C. Meesookho, S. Narayanan, C. Raghavendra, "Collaborative classification applications in sensor networks", *Sensor Array and Multichannel Signal Processing Workshop*, 2002.
13. T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, B. Krogh, "Energy-efficient surveillance system using wireless sensor networks", *2nd international conference on Mobile systems, applications and services*, 2004.

14. B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, S. Sastry, "Distributed control applications within sensor networks", *Proceedings of the IEEE*, cilt 91, no. 8, pp. 1235-1246, 2003.
15. A. Nayak, I. Stojmenovic, "Single-Hop Wireless Sensor Networks", *Wireless Sensor and Actuator Networks*, Wiley, 2010, p. 3.
16. <http://coe.berkeley.edu/labnotes/0701brainybuildings.html> (27.04.2013)
17. M. Meyer, M. Brambley, "Pros & Cons of Wireless" *ASHRAE Journal*, pp. 54-59, 2002.
18. A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, J. Anderson, "Wireless sensor networks for habitat monitoring", *1st ACM international workshop on Wireless sensor networks and applications*, 2002.
19. <http://www.alertsystems.org/> (27.04.2013)
20. A. Chandrakasan, R. Amirtharajah, C. Seonghwan, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, A. Wang, "Design considerations for distributed microsensor systems", *IEEE Custom Integrated Circuits*, 1999.
21. <http://www.mobilab.unina.it/images/WSNs.jpg> (27.04.2013)
22. <http://www.accenture.com/SiteCollectionDocuments/PDF/pickberry.pdf> (27.04.2013)
23. N. Noury, T. Herve, V. Rialle, E. M. G. Virone, G. Morey, A. Moro, T. Porcheron, "Monitoring Behaviour in Home Using a Smart Fall Sensor", *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, 2000.
24. T. Knott, "Smart Surrogates", *BP Frontiers*, no. 9, pp. 6-10, 2004.
25. E. Blomqvist, H. N. Koivo, "Security in Sensor Networks - A Case Study", *IEEE 12th Mediterranean Conference on Control and Automation*, 2004.
26. J. Rabaey, E. Arens, C. Federspiel, A. Gadgil, D. Messerschmitt, W. Nazaroff, K. Pister, S. Oren, P. Varaiya, "Smart Energy Distribution and Consumption: Information Technology as an Enabling Force", *University of California at Berkeley*, 2001.
27. A. A. Abbasi, M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks", *Computer Communications*, no. 30, pp. 2826-2841, 2007.

28. O. Boyinbode, H. Le, A. Mbogho, M. Takizawa, R. Poliah, "A Survey on Clustering Algorithms for Wireless Sensor Networks", *13th International Conference on Network-Based Information Systems*, 2010.
29. D. J. Dechene, A. E. Jardali, M. Luccini, A. Sauer, "A Survey of Clustering Algorithms for Wireless Sensor Networks", *The University of Western Ontario, Ontario*.
30. K. Sohrabi, J. Gao, V. Ailawadhi, G. J. Pottie, "Protocols for SelfOrganization of a Wireless Sensor Network", *IEEE Personal Communications*, cilt 7, no. 5, pp. 16-27, 2000.
31. O. Younis, S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks", *IEEE Transactions on Mobile Computing*, cilt 3, no. 4, p. 366–379, 2004.
32. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, F. Silva, "Directed Diffusion for Wireless Sensor Networking", *IEEE/ACM Transactions on Networking*, cilt 11, no. 1, pp. 2-16, 2003.
33. W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *33rd Hawaii International Conference on System Sciences*, 2000.
34. V. Loscri, G. Morabito, S. Marano, "A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH)", *IEEE hicular Technology Conference*, cilt 3, pp. 1809 - 1813 , 2005.
35. P. Ding, J. Holliday, A. Celik, "Distributed energy efficient hierarchical clustering for wireless sensor networks", *IEEE International Conference on Distributed Computing in Sensor Systems*, 2005.
36. M. Ye, C. Li, G. Chen, J. Wu, "EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks", *IEEE Performance Computing and Communications Conference*, 2005.
37. S. Hussain, A. W. Matin, "Base station assisted hierarchical cluster-based routing", *IEEE/ACM International Conference on Wireless and Mobile communications Networks*, 2006.
38. M. O. Farooq, A. B. Dogar, G. A. Shah, "MR-LEACH: Multi-hop Routing with Low Energy Adaptive Clustering Hierarchy", *IEEE Fourth International Conference on Sensor Technologies and Applications*, 2010.
39. F. Xiangning, S. Yulin, "Improvement on LEACH Protocol of Wireless Sensor Network", *IEEE International Conference on Sensor Technologies and*

Applications, 2007.

40. L. Wang, J. Liu, W. Wang, "An Improvement and Simulation of LEACH Protocol for Wireless Sensor Network", *First International Conference on Pervasive Computing, Signal Processing and Applications*, 2010.
41. M. J. Handy, M. Haase, D. Timmermann, "Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection", *Fourth IEEE Conference on Mobile and Wireless Communications Networks*, 2002.
42. W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks" *IEEE Transactions on Wireless Communications*, cilt 1, no. 4, pp. 660-670, 2002.
43. S. D. Muruganathan, D. C. F. Ma, B. R. I., F. A. O., "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks", *IEEE Radio Communications*, cilt 43, no. 3, pp. 8-13, 2005.
44. D. J. Baker, A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm", *IEEE Transactions on Communications*, cilt 29, no. 11, pp. 1694-1701, 1981.
45. D. J. Baker, A. Ephremides, J. Flynn, "The design and simulation of a Mobile Radio Network with Distributed Control", *IEEE Journal on Selected Areas in Communications*, cilt 2, no. 1, pp. 226-237, 1984.
46. M. Demirbas, A. Arora, V. Mittal, V. Kulathumani, "Design and Analysis of a Fast Local Clustering Service for Wireless Sensor Networks", *First International Conference on Broadband Networks*, 2004.
47. R. Biradar, S. R. Sawant, R. Mudholkar, V. Patil, "Multihop Routing In Self-Organizing Wireless Sensor Networks", *International Journal of Computer Science Issues*, cilt 8, no. 1, pp. 155-164, 2011.
48. T. Qiang, W. Bingwen, D. Zhicheng, "MS-Leach: A Routing Protocol Combining Multi-hop Transmissions and Single-hop Transmissions", *Pacific-Asia Conference on Circuits, Communications and System*, 2009.
49. E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, cilt 1, no. 1, pp. 269-271, 1959.
50. http://en.wikipedia.org/wiki/Dijkstra%27s_algorithm#cite_note-Dijkstra_Interview-1 (06.04.2013).
51. K. Maraiya, K. Kant, N. Gupta, "Efficient Cluster Head Selection Scheme for Data Aggregation in Wireless Sensor Network", *International Journal of*

- Computer Applications*, cilt 23, no. 9, pp. 10-18, 2011.
52. M. Mitchell, "Genetic Algorithms: An Overview", *Complexity*, cilt 1, no. 1, pp. 31-39, 1995.
 53. V. V. Nabiyev, "Genetik Algoritmalar", *Yapay Zeka*, Seçkin Yayınları, 2010, pp. 586-597.
 54. Ç. Elmas, "Genetik Algoritma, Tarihçesi", *Yapay Zeka Uygulamaları*, Seçkin Yayınları, 2007, pp. 379-384.
 55. G. G. Emel, Ç. Taşkın, "Genetik Algoritmalar, Uygulama Alanları" *Uludağ Üniversitesi İktisadi, İdari Bilimler Fakültesi Dergisi*, cilt 21, no. 1, pp. 129-152, 2002.
 56. Y. Tsujimura, M. Gen, E. Kubota, "Solving fuzzy assembly-line balancing problem with genetic algorithms", *17th International Conference on Computers and Industrial Engineering*, 1995.
 57. N. Özçakar, "Genetik Algoritmalar", *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, cilt 27, no. 1, pp. 69-82, 1998.
 58. <http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/> (04.05.2013)
 59. S. Sarmady, "An Investigation on Genetic Algorithm Parameters", *Universiti Sains Malaysia*, 2007.
 60. A. H. Wright, "Genetic Algorithms for Real Parameter Optimization" *University of Montana*, 1991.
 61. S. Jin, M. Zhou, A. S. Wu, "Sensor Network Optimization Using a Genetic Algorithm", *7th World Multiconference on Systemics Cybernetics and Informatics*, 2003.
 62. R. Khanna, H. Liu, H. Chen, "Self-Organization of Sensor Networks Using Genetic Algorithms", *IEEE International Conference on Communications*, 2006.
 63. Q. Qiu, Q. Wu, D. Burns, D. Holzhauer, "Lifetime Aware Resource Management for Sensor Network Using Distributed Genetic Algorithm", *International Symposium on Low Power Electronics and Design*, 2006.
 64. C. Lai, C. Ting, R. Ko, "An Effective Genetic Algorithm to Improve Wireless Sensor Network Lifetime for Large-Scale Surveillance Applications", *IEEE Congress on Evolutionary Computation*, 2007.
 65. S. Hussain, A. W. Matin, O. Islam, "Genetic Algorithm for Hierarchical Wireless

- Sensor Networks”, *Journal of Networks*, cilt 2, no. 5, pp. 87-97, 2007.
66. A. Bari, S. Wazed, A. Jaekel, S. Bandyopadhyay, “A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks”, *Ad Hoc Networks*, cilt 7, no. 4, pp. 665-676, 2009.
 67. D. Turgut, S. K. Das, R. Elmasri, B. Turgut, “Optimizing Clustering Algorithm in Mobile Ad hoc Networks Using Genetic Algorithmic Approach”, *IEEE Global Telecommunications Conference*, 2002.
 68. K. P. Ferenitos, T. A. Tsiligiridis, K. G. Arvanitis, “Energy Optimization of Wireless Sensor Networks for Environmental Measurements”, *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2005.
 69. A. Norouzi, F. S. Babamir, A. H. Zaim, “A New Clustering Protocol for Wireless Sensor Networks Using Genetic Algorithm Approach”, *Wireless Sensor Network*, cilt 3, no. 11, pp. 362-370, 2011.
 70. S. Bayraklı, S. Z. Erdogan, “Genetic Algorithm Based Energy Efficient Clusters (GABEEC) in Wireless Sensor Networks”, *The 3rd International Conference on Ambient Systems, Networks and Technologies*, 2012.

ÖZGEÇMİŞ

Hüseyin Fehmi Selim BAYRAKLI, 1984 yılında İstanbul / Üsküdar'da doğdu. Öğrenimlerini sırasıyla Faik Reşit Unat İlkokulu ve Kartal Anadolu İmam Hatip Lisesi'nde tamamladı. 2002 yılında Maltepe Üniversitesi Mühendislik – Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünü kazandı ve 2006 yılında bölüm birincisi olarak mezun oldu. Ekim 2006'da, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans programında yüksek lisans öğrenimi yapmaya hak kazandı. Kasım 2006'da, halen çalışmalarını sürdürmekte olduğu Maltepe Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne Araştırma Görevlisi olarak atandı. Eylül 2007 – 2008 tarihleri arasında geliştirilen Maltepe Üniversitesi Uzaktan Eğitim Sistemi yazılım projesinde görev aldı. Eylül 2008'de yüksek lisans öğrenimini tamamladı ve aynı zaman içerisinde Maltepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Doktora programında doktora öğrenimi yapmaya hak kazandı. Kasım 2010'da Maltepe Üniversitesi Mühendislik Fakültesi Yazılım Mühendisliği Bölümü'ne Öğretim Görevlisi olarak atandı ve halen Öğretim Görevlisi olarak görev yapmaktadır.