

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GEZGİN TASARSIZ AĞLAR İÇİN YENİ BİR KONUM
TABANLI MELEZ YÖNLENDİRME ALGORİTMASI

DOKTORA TEZİ

Resul KARA

Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK MÜH.
Enstitü Bilim Dalı : ELEKTRONİK
Tez Danışmanı : Yrd. Doç. Dr. İbrahim ÖZÇELİK
Ortak Tez Danışmanı : Prof. Dr. Hüseyin EKİZ

Şubat 2009

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GEZGİN TASARSIZ AĞLAR İÇİN YENİ BİR KONUM
TABANLI MELEZ YÖNLENDİRME ALGORİTMASI

DOKTORA TEZİ

Resul KARA

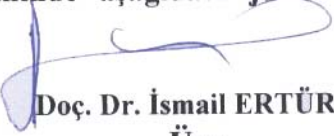
Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK MÜH.

Enstitü Bilim Dalı : ELEKTRONİK

Bu tez 12/02/2009 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.


Prof. Dr. Etem KÖKLÜKAYA


Jüri Başkanı


Doç. Dr. İsmail ERTÜRK

Üye


Doç. Dr. M. Ali AKCAYOL

Üye


Yrd. Doç. Dr. İbrahim ÖZÇELİK

Üye


Yrd. Doç. Dr. Ahmet ZENGİN

Üye

TEŐEKKÜR

Bu tez alıőmasının tamamlanmasında deęerli katkılarını esirgemeyen danıőmanım Yrd. Do. Dr. İbrahim ÖZELİK'e ve Prof. Dr. Hüseyin EKİZ'e, alıőmalarım sırasında bana sürekli ve sabırla destek olan aileme teőekkürü bir bor bilirim.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ	vii
ŞEKİLLER LİSTESİ	x
TABLolar LİSTESİ	xiv
ÖZET.....	xvii
SUMMARY	xviii

BÖLÜM 1.

GİRİŞ	1
1.1. Kablosuz Ağların Tarihi ve Gelişimi.....	3
1.2. Kablosuz Tasarsız Ağların (MANETs) Karakteristikleri	4
1.3. Kablosuz Tasarsız Ağlar için Yönlendirme Algoritmaları ve Literatürde Yer Alan Geliştirme Çalışmaları	6
1.4. Çalışmanın Amacı, Önerilen Çözüm Yöntemi ve Katkıları	10
1.5. Tez Organizasyonu	15

BÖLÜM 2.

GEZGİN TASARSIZ AĞLARDA YÖNLENDİRME.....	17
2.1. Gezgin Tasarsız Ağlarda Yönlendirme Algoritmalarının Sınıflandırması	17
2.2. Tabloya Dayalı (Table-Driven) Yönlendirme Algoritmaları	18
2.2.1. Destination sequenced distance vector algoritması	18
2.2.2. Wireless routing protocol.....	19
2.3. İsteğe Bağlı Yönlendirme Algoritmaları	20
2.3.1. Dynamic source routing	20
2.3.2. Ad hoc on-demand distance vector	22
2.3.3. Temporally ordered routing algoritması	23

2.4. Melez Yönlendirme Algoritmaları	24
2.4.1. Konum tabanlı yönlendirme algoritmaları	24
2.4.1.1. Geographic distance routing	25
2.4.1.2. Most forward with in radius ve directional routing	26
2.4.1.3. Distance routing effect algorithm for mobility	27
2.4.2. Multi point relaying tabanlı algoritmalar	28
2.5. Tasarsız Yönlendirme Algoritmalarının Karşılaştırma Tablosu.....	29
2.6. Sonuç	29

BÖLÜM 3.

KONUM TABANLI MELEZ YÖNLENDİRME ALGORİTMASI (KTMYA).....	31
3.1. Giriş	31
3.2. Düğümlerin Sahip Olması Gereken Özellikler.....	32
3.3. KTMYA'nın Adımları.....	32
3.4. KTMYA Algoritması.....	35
3.4.1. Master düğüm rolünü belirleme	42
3.4.2. Yönlendirme bilgisi isteği ve cevabı.....	43
3.4.3. Master düğüm duyuru paketlerinin ağda dağıtımı	44
3.4.4. Veri paketlerinin ağda dağıtımı.....	46
3.4.5. Konum bilgisi güncelleme paketlerinin ağda dağıtımı	47
3.5. Bulanık Mantık Yöntemiyle Maliyet Değerlerinin Belirlenmesi	47
3.5.1. Bulanık mantığın KTMYA'ya uygulanması	48
3.5.2. Örnek bir ağ üzerinde maliyet değerlerinin belirlenmesi.....	54
3.5.3. Bulanık mantık kullanmadan düğümler arası maliyet	56
3.6. KTMYA'da Öçekleme	64
3.6.1. Fuzzy c-means kümeleme.....	65
3.6.2. KTMYA'da düğümlerin kümelere ayrılması.....	66
3.6.3. Örnek ağ üzerinde kümeleme işleminin uygulanması	71
3.7. KTMYA'nın Örnek Bir Ağ Üzerinde Uygulanması	74
3.8. Sonuç	81

BÖLÜM 4.

KTMYA'NIN PERFORMANS DEĞERLENDİRMESİ	82
--	----

4.1. MANET’te Performans Değerlendirme Kriterleri.....	82
4.1.1. Ölçülebilir performans kriterleri	82
4.1.2. Niteliksel performans kriterleri	83
4.1.3. Ağ performansını etkileyen değişken parametreler	85
4.2. Konum Tabanlı Melez Yönlendirme Algoritmasında Performans	
Değerlendirme İşlem Adımları	85
4.2.1. Performans değerlendirme ortamı.....	85
4.2.2. Düğümler arası mesafeler	86
4.2.3. Master düğümü belirleme	86
4.2.4. Bulanık mantık ile düğümler arası maliyet değerlerinin	
belirlenmesi.....	87
4.2.5. Düğümler arası en kısa yol (en maliyetsiz yol) belirleme işlemi.....	87
4.2.6. Düğümlerin paket üretimi	88
4.2.7. Düğümlerin hareketliliği	88
4.2.8. Düğümlerin enerji tüketimleri.....	89
4.3. Performans Değerlendirme	90
4.3.1. Benzetim parametreleri	90
4.3.2. Yönlendirme yükü.....	91
4.3.3. Paket dağıtım oranı	95
4.3.4. Paket kayıp oranı.....	98
4.3.5. Uçtan uca ortalama paket gecikmesi.....	99
4.3.6. Enerji Verimliliği	104
4.4. Performans Karşılaştırması.....	106
4.5. Sonuç	111

BÖLÜM 5.

SONUÇLAR VE DEĞERLENDİRMELER	112
5.1. Sonuçlar	112
5.2. Çalışmanın Getirdiği Katkıları.....	115
5.3. Tartışma ve Öneriler	116
KAYNAKLAR	119
EKLER.....	127

ÖZGEÇMİŞ	158
----------------	-----

SİMGELER VE KISALTMALAR LİSTESİ

ABR	: Associativity Based Routing
AGNES	: AGglomerative NEsting
AODV	: Ad hoc On-Demand Distance Vector
APR	: Alternate Path Routing
BLR	: Beacon-less Routing
CBR	: Cluster Based Routing
CGSR	: Clusterhead Gateway Switch Routing
CSMA/CA	: Carrier Sense Multiple Access/Collision Avoidance
CURE	: Clustering Using REpresentatives
DFR	: Direction Forward Routing
DIR	: Directional Routing
DNVR	: Dynamic Nix-Vector Routing
DREAM	: Distance Routing Effect Algorithm For Mobility
DSDV	: Destination Sequenced Distance-Vector
DSR	: Dynamic Source Routing,
DYMO	: DYnamic Manet On-demand Routing
FCM	: Fuzzy C Means
FIFO	: First In First Out
FSR	: Fisheye State Routing
GEDIR	: Geographical Distance Routing
GloMo	: Global Mobile Information Systems
gp	: Güncelleme Paketi
GPS	: Global Positioning System
GSM	: Global System for Mobile Communications
GSR	: Global State Routing
HSR	: Hierarchical State Routing

IARP	: Intrazone Routing Protocol
IEEE	: Institute of Electrical and Electronics Engineers
IETF	: Internet Engineering Task Force
KTMYA	: Konum Tabanlı Melez Yönlendirme Algoritması
LAR	: Location Aided Routing
LAR	: Load Aware Routing
MAC	: Medium Access Control
MANET	: Mobile Ad Hoc Networking
MAODV	: Multirate Ad hoc On-Demand Distance Vector Routing
mcp	: Master Düğüm Öneri Cevap Paketi
mdp	: Master Düğüm Duyuru Paketi
MFR	: Most Forward With in Radius
Mop	: Master Düğüm Öneri Paketi
MPR	: Multi-Path Routing
MPR	: Multi Point Relaying
MREP	: Maximum Residual Energy Path
NTDR	: Near-Term Digital Radio
OSI	: Open System Interconnect
OSPF	: Open Shortest Path First
PAR	: Power-Aware Routing
PDA	: Personal Digital Assistant
PRNET	: Packet Radio Network
REEF	: Reliable and Efficient Forwarding
RFC	: Request for Comments
RIP	: Routing Information Protocol
SI	: Uluslararası Birim Sistemi
SSA	: Signal Stability Based Adaptive Routing
SSR	: Signal Stability Routing
STAR	: Source Tree Adaptive Routing
SURAN	: Survivable Adaptive Radio Networks
TAI	: Turkish Aerospace Industries
TDMA	: Time Division Multiple Access
TORA	: Temporally-Ordered Routing Algorithm

US DoD	: United States Department of Defence
WLAN	: Wireless Local Area Network
WRP	: Wireless Routing Protocol
WRP-Lite	: Wireless Routing Protocol-Lite
ycp	: Yol Cevap Paketi
yip	: Yol İstek Paketi
ZHLS	: Zone-based Hierarchical Link State
ZRP	: Zone Routing Protocol

ŞEKİLLER LİSTESİ

Şekil 1.1. Altyapılı kablosuz ağ	2
Şekil 1.2. Altyapısız kablosuz ağ	2
Şekil 2.1.a. DSR protokolünde yol bulma mekanizması ve istek paketinin izlediği yol [41]	21
Şekil 2.1.b. Yol cevap paketinin izlediği yol [41].....	21
Şekil 2.2. Yol istek paketi yapısı.....	22
Şekil 2.3. Yol cevap paketi	22
Şekil 2.4. GEDIR örnek ağı üzerinde yol belirlenmesi.....	26
Şekil 2.5. GEDIR örnek ağı üzerinde algoritmanın durması ve hata oluşması.....	26
Şekil 2.6. DREAM algoritmasında mesafe etkisi	28
Şekil 2.7. MPR tekniği yayın akışı.....	29
Şekil 3.1. KTMYA akış diyagramı	34
Şekil 3.2. Master düğüm duyuru paketlerinin (mdp) çerçeve yapısı	35
Şekil 3.3. Düğüm bilgi güncelleme paketlerinin (gp) çerçeve yapısı	36
Şekil 3.4. Master düğüm aday önerisi paketi	38
Şekil 3.5. Master düğüm adaylık kabul veya red paketi	39
Şekil 3.6. Veri paketi çerçeve yapısı	40
Şekil 3.7. Yol İstek Paketi çerçeve yapısı	40
Şekil 3.8. Yol Cevap Paketi çerçeve yapısı.....	41
Şekil 3.9. Master düğüm rolünü belirleme işlemi akış diyagramı	43
Şekil 3.10. Yönlendirme bilgisi isteği ve cevabı işlemlerinin akış diyagramı.....	45

Şekil 3.11. Master düğüm duyuru paketlerinin ağda dağıtımı	46
Şekil 3.12. Veri paketlerinin ağda dağıtımı	47
Şekil 3.13. Güncelleme paketlerinin ağda dağıtımı	48
Şekil 3.14. Bulanık mantık giriş ve çıkış değişkenlerinin blok görünümü	49
Şekil 3.15. Mesafe giriş değişkeni üyelik fonksiyonlarının grafik görünümü	50
Şekil 3.16. Batarya ömrü giriş değişkeni üyelik fonksiyonlarının grafik görünümü	50
Şekil 3.17. Yoğunluk giriş değişkeni üyelik fonksiyonlarının grafik görünümü.....	51
Şekil 3.18. Maliyet çıkış değişkeni üyelik fonksiyonlarının grafik görünümü.....	52
Şekil 3.19.a. Batarya ömrü ve mesafeye karşı maliyetin değişimi	53
Şekil 3.19.b. Yoğunluk ve mesafeye karşı maliyetin değişimi	53
Şekil 3.20. 10 düğümlü örnek ağın düğüm yerleşimleri	55
Şekil 3.21. 10 düğümlü örnek ağın düğüm yerleşimleri	57
Şekil 3.22. 50 düğümlü ağda bulanık mantık kullanılarak belirlenmiş 3. düğümden 8. düğüme olan yol	59
Şekil 3.23. 50 düğümlü ağda 3. düğümden 8. düğüme klasik yöntemle bulunmuş en maliyetsiz yol.....	59
Şekil 3.24. 100 düğümlü ağda 1. düğümden 2. düğüme doğru bulanık mantıkla bulunmuş yol	60
Şekil 3.25. 100 düğümlü ağda 1. düğümden 2. düğüme, bulanık mantık kullanılmadan bulunmuş yol	62
Şekil 3.26. 1000 düğümlü ve kümelere ayrılmamış ağın düğüm görünümü	67
Şekil 3.27. 1000 düğümlü ağın 3 kümeye ayrılmış hali.....	68
Şekil 3.28. 1000 düğümlü ağın 4 kümeye ayrılmış hali.....	68
Şekil 3.29. 1000 düğümlü ağın 5 kümeye ayrılmış hali.....	69
Şekil 3.30. 1000 düğümlü ağın 6 kümeye ayrılmış hali.....	69
Şekil 3.31. Kümeleme ve kümeler arası iletili düğüm belirleme akış diyagram.....	72

Şekil 3.32. 20 düğümlü ağın kümelenmemiş ve iki kümeye ayrılmış görünümü.....	73
Şekil 3.33. 20 düğümlü ağda düğümlerin görünüşleri.....	75
Şekil 3.34. 20 düğümlü ağda düğümlerin 100 saniye hareket etmiş görünüşleri.....	77
Şekil 3.35. 20 düğümlü ağda 1. düğümün 10 saniye boyunca izlediği yol.....	78
Şekil 3.36. 20 düğümlü ağın iki kümeye ayrılmış hali	80
Şekil 4.1. 50 düğümlü bir ağın başlangıç ve 100 saniye hareket etmiş konumları	89
Şekil 4.2. 100 düğümlü bir ağın başlangıç ve 100 saniye hareket etmiş konumları	89
Şekil 4.3. 10 düğümlü, 5 ve 10 kaynaklı ağın yönlendirme yükü.....	92
Şekil 4.4. 20 düğümlü, 10 ve 20 kaynaklı ağın yönlendirme yükü.....	93
Şekil 4.5. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü.....	93
Şekil 4.6. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü.....	94
Şekil 4.7. 10 düğümlü, 5 ve 10 kaynaklı ağın paket dağıtım oranları.....	96
Şekil 4.8. 20 düğümlü, 10 ve 20 kaynaklı ağın paket dağıtım oranları.....	96
Şekil 4.9. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları.....	97
Şekil 4.10. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları.....	98
Şekil 4.11. 10 düğümlü, 5 ve 10 kaynaklı ağın uçtan uca ortalama paket gecikmesi.....	101
Şekil 4.12. 20 düğümlü, 10 ve 20 kaynaklı ağın uçtan uca ortalama paket gecikmesi.....	102
Şekil 4.13. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi.....	102
Şekil 4.14. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi.....	103
Şekil 4.15. 10 düğümlü, 5 ve 10 kaynaklı ağdaki düğümlerin toplam enerji tüketimi	104

Şekil 4.16. 20 düğümlü, 10 ve 20 kaynaklı ağdaki düğümlerin toplam enerji tüketimi	105
Şekil 4.17. 50 düğümlü, 10 ve 20 kaynaklı ağdaki düğümlerin toplam enerji tüketimi	106
Şekil 4.18. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için yönlendirme yükü değerleri.....	107
Şekil 4.19. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için paket dağıtım oranları.....	108
Şekil 4.20. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için uçtan uca paket gecikmesi	110
Şekil A.1. 6 düğümlü yönlendirilmiş graf.....	127
Şekil A.2. 6 düğümlü yönlendirilmemiş graf.....	128
Şekil A.3. Ağırlık değerli yönlendirilmiş ve yönlendirilmemiş graf	128
Şekil B.1. Bulanık sistem yapısının genel gösterimi.....	132
Şekil B.2. Üçgen, yamuk ve çan eğrisi (Gauss) üyelik fonksiyonları.....	134

TABLolar LİSTESİ

Tablo 2.1. Tasarsız ağlar yönlendirme algoritmalarının karşılaştırması.....	30
Tablo 3.1. Mesafe giriş değişkeni üyelik fonksiyonu sınırları.....	50
Tablo 3.2. Batarya ömrü değişkeni üyelik fonksiyonu sınırları.....	51
Tablo 3.3. Yoğunluk giriş değişkeni üyelik fonksiyonu sınırları.....	51
Tablo 3.4. Mesafe giriş değişkeni üyelik fonksiyonu sınırları.....	52
Tablo 3.5. Örnek giriş değerleri için, bulanık mantıkla hesaplanmış maliyet değerleri.....	53
Tablo 3.6. 10 düğümlü örnek ağ için konum, batarya ve yoğunluk değerleri	54
Tablo 3.7. 10 düğümlü örnek ağ için düğümler arası mesafe değerleri.....	55
Tablo 3.8. 10 düğümlü örnek ağ için düğümler arası mesafe değerleri.....	56
Tablo 3.9. 50 düğümlü ağın konum, batarya ve yoğunluk bilgileri.....	58
Tablo 3.10. 100 düğümlü ağın konum, batarya ve yoğunluk bilgileri.....	61
Tablo 3.11. 20 düğümlü ağın konum bilgileri.....	71
Tablo 3.12. 20 düğümlü ağın küme aitlik dereceleri.....	73
Tablo 3.13. 20 düğümlü ağın P matrisi verileri	74
Tablo 3.14. 20 düğümlü ağda düğümler arası mesafeler	75
Tablo 3.15. 20 düğümlü ağda toplam mesafe değerleri	76
Tablo 3.16. 20 düğümlü ağın bulanık mantık sonucu elde edilen düğümler arası maliyet değerleri.....	77
Tablo 3.17. 20 düğümlü ağın paket sayıları ve güç tüketimleri.....	79
Tablo 3.18. Batarya azalması durumunda değişen en kısa yollar	79

Tablo 3.19. 20 düğümlü ağın küme aitlik dereceleri.....	81
Tablo 4.1. 10 düğümlü, 5 ve 10 kaynaklı ağın yönlendirme yükü değerleri	92
Tablo 4.2. 20 düğümlü, 10 ve 20 kaynaklı ağın yönlendirme yükü değerleri	92
Tablo 4.3. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü değerleri	94
Tablo 4.4. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü değerleri ..	94
Tablo 4.5. 10 düğümlü, 5 ve 10 kaynaklı ağın paket dağıtım oranları	95
Tablo 4.6. 20 düğümlü, 10 ve 20 kaynaklı ağın paket dağıtım oranları	97
Tablo 4.7. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları.....	97
Tablo 4.8. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları.....	98
Tablo 4.9. 10, 20, 50 ve 100 düğümlü, 10 kaynaklı ağların paket kayıp oranı değerleri.....	100
Tablo 4.10. 10 düğümlü, 5 ve 10 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri	101
Tablo 4.11. 20 düğümlü, 10 ve 20 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri	101
Tablo 4.12. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri.....	103
Tablo 4.13. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri.....	103
Tablo 4.14. 10 düğümlü, 5 ve 10 kaynaklı ağdaki düğümlerin toplam enerji tüketimi değerleri	105
Tablo 4.15. 10 düğümlü, 5 ve 10 kaynaklı ağdaki düğümlerin toplam enerji tüketimi değerleri	105
Tablo 4.16. 50 düğümlü, 10 ve 20 kaynaklı ağdaki düğümlerin toplam enerji tüketimi değerleri	106
Tablo 4.17. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için yönlendirme yükü değerleri.....	107

Tablo 4.18. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için paket dağıtım oranları.....	109
Tablo 4.19. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için uçtan uca paket gecikmesi değerleri.....	109
Tablo 4.20. 50 ve 100 düğümlü, 20 kaynaklı ağların KTMYA ve DREAM için performans değerleri	110
Tablo D.1. 10 düğümlü ağın 100 saniye boyunca kalan batarya ömrü değerleri....	146
Tablo D.2. 20 düğümlü ağın 100 saniye boyunca kalan batarya ömrü değerleri....	146

ÖZET

Anahtar kelimeler: Ad hoc, yönlendirme, yönlendirme algoritması, kablosuz ağ

Ad hoc kablosuz ağlar, baz istasyonu veya erişim cihazı gibi bir altyapı olmaksızın birbirleriyle iletişim kuran gezgin düğümlerden oluşur. Bu tür ağlarda düğümlerin hareketliliğine bağlı olarak sık sık topoloji değişikliği olur. Bundan dolayı düğümlerin birbirleriyle iletişimde kalmaları için ağda yayın paketlerinin sıklıkla gönderimine gereksinim duyulur. Gezgin tasarsız ağlarda tabloya dayalı ve isteğe bağlı yönlendirme algoritmaları kullanılmaktadır. Her iki tür algoritmada da düğüm sayısının çok olduğu ve aşırı düğüm hareketliliği durumlarında yönlendirme ek yükü artmaktadır. Sınırlı bant genişliği ve düğüm batarya ömürlerinin iki önemli kaynak olduğu bu ağlarda, yönlendirme ek yükünün en düşük seviyeye indirilerek sınırlı kaynakların verimli bir şekilde kullanılması gerekmektedir.

Bu tez çalışmasında, tasarsız ağların sınırlı ve en önemli iki kaynağı olan bant genişliği ve batarya kullanımını optimize eden yeni bir yönlendirme algoritması geliştirilmiştir. Konum Tabanlı Melez Yönlendirme Algoritması (KTMYA) adı verilen bu algoritma ile yönlendirme ek yükü azaltılarak bant genişliğinin etkin kullanımı sağlanmaktadır. Bununla birlikte, ağdaki her bir düğüm tarafından yönlendirme amacıyla yapılacak işlem sayısı ve bellekte tutulacak veri miktarı azaltılarak batarya ömrü uzatılmaktadır. KTMYA hem tabloya dayalı hem de isteğe bağlı algoritmaların çalışma esaslarından faydalanılarak geliştirilmiştir. Ancak KTMYA, yönlendirme kararı verilirken düğüm pozisyon bilgileri, düğüm batarya miktarları ve düğüm yoğunluk bilgileri kullanımı açısından bu iki yöntemin dışında yeni bir yaklaşım içermektedir.

Geliştirilen algoritma, MATLAB 7.0.1'de kodlanan bir benzetim programı ile başarımlı değerlendirilerek yapılarak, paket dağıtım oranı, yönlendirme ek yükü ve uçtan uca paket gecikmesi açısından tabloya dayalı, isteğe bağlı ve konum tabanlı yönlendirme algoritmaları ile karşılaştırmalı olarak sunulmaktadır. KTMYA, yönlendirme ek yükü, paket dağıtım oranı ve uçtan uca paket gecikmesi açısından daha iyi sonuçlar vermektedir.

A NEW POSITION BASED HYBRID ROUTING ALGORITHM FOR MOBILE AD HOC NETWORKS

SUMMARY

Keywords: Ad hoc, routing, routing algorithm, wireless network

Ad hoc wireless networks consist of nodes which communicate with each other without infrastructure such as base station and access point. In these type networks, the network topology frequently changes due to the movements of the nodes. Broadcast packets are regularly sent to network so that nodes in network can stay in communication with each other. Reactive and proactive routing algorithms are used in ad hoc networks, where routing overhead increases in the case of large number of nodes and raised mobility. Bandwidth and battery lives of nodes are two main sources in ad hoc networks. The reduction of routing overhead and efficiently use of sources are very important issues in these networks.

In this study, a new routing algorithm called Position Based Hybrid Routing Algorithm (PBHRA) was developed to optimize bandwidth usage of ad hoc networks. The main goal of PBHRA is effective use of bandwidth by reducing the routing overload. Additionally, the other goals of the algorithm are to extend battery life of the mobile devices by reducing the required number of operations for route determination and reduce the amount of memory used. Although in the PBHRA, some features of both table driven and on-demand algorithms were used to achieve these goals at some stages, PBHRA algorithm is a completely different approach from them in terms of giving routing decision that use node position information, node battery life and node buffer value.

The PBHRA was coded and simulated in MATLAB 7.0.1 to evaluate its performance and compared with other algorithms. The results showed that PBHRA performs better in terms of routing overload, packet delivery fraction, and end-to-end average packet delay compared to table driven, on demand, and position based algorithms.

BÖLÜM 1. GİRİŞ

Kablosuz ağlar ilk ortaya çıktığı 1970'lerden beri oldukça popüler olmuştur. Popülaritesi kullanıcının coğrafi yeri ne olursa olsun bilgiye erişim imkânı sağlamasından kaynaklanmaktadır. İnternetin gelişimine paralel olarak kablosuz sistemlerin gelişiminde de gözle görülür bir ilerleme gözlemlenmektedir. Kullanıcıların teknolojik eğilimi, kablosuz ve gezgin cihazlarla iletişimi sağlama yönündedir. Son zamanlarda kullanımları hızla artan hücreli telefonlar, kablosuz yerel alan ağı (WLAN- Wireless Local Area Network) özelliğine sahip taşınabilir bilgisayarlar ve el bilgisayarları (PDA-Personal Digital Assistant) bunun en büyük göstergesidir.

Kablosuz ağlar “altyapılı” ve “altyapısız” olmak üzere iki sınıfa ayrılabilir. Altyapılı kablosuz ağlar birbirlerine linklerle bağlanmış sabit baz istasyonlarına veya erişim noktalarına sahiptir. Gezgin düğümler birbirleriyle bu baz istasyonları aracılığıyla haberleşirler. Altyapılı kablosuz ağlar hücreli ağlar olarak da adlandırılır. GSM ve IEEE 802.11 ortam erişim kontrolünü kullanarak bir kablosuz erişim cihazıyla haberleşen bilgisayarların oluşturduğu ağlar, bu tür ağlardandır.

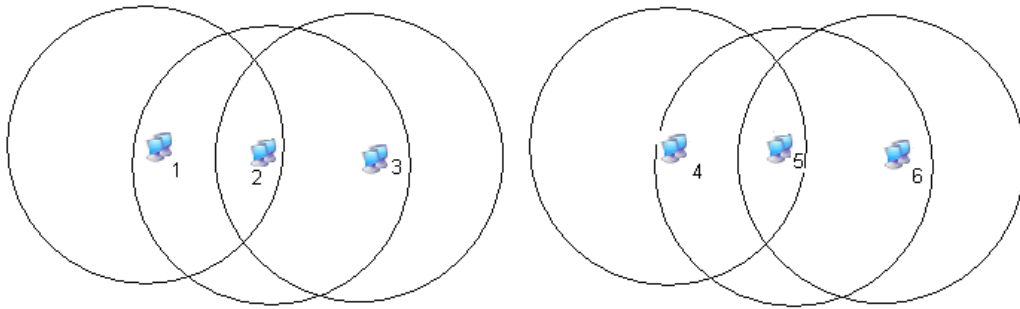
Altyapılı kablosuz ağlarda yönlendirme görevini yerine getirmek için erişim noktası cihazları veya baz istasyonları kullanılır. Şekil 1.1'de görülen 1, 2 ve 3 numaralı düğümler kapsama alanlarında olan erişim noktası cihazıyla, 4, 5 ve 6 numaralı düğümler de kendi kapsama alanlarındaki erişim cihazıyla haberleşebilir. Böyle bir durumda iki grubun haberleşmesi için kablolu bir ağ anahtarına ihtiyaç vardır [1].



Şekil 1.1. Altyapılı kablosuz ağ

Altyapısız kablosuz ağlar ise önceden tanımlanmış altyapı veya baz istasyonları gibi merkezi bir kontrol olmadan bir araya gelmiş gezgin düğümlerdir. Altyapısız kablosuz ağlar askeri uygulamalar, kurtarma senaryoları, ofisler gibi yerel alanlar, WLAN (Wireless Local Area Network) kurulumları, ev ağları, robot ağları, sensör (algılayıcı) ağları, kişisel alan ağları, kablosuz cihazların ara bağlantıları, vahşi yaşam alanları ve mikro iklimlendirme uygulamalarının görüntülenmesi aşamaları vb. birçok alanda kullanılır.

Altyapısız kablosuz ağlarda ağdaki her bir düğüm bir yönlendirici olarak davranır ve yönlendirme işlemini yapar. Şekil 1.1’de verilen düğümler altyapısız düğümlerle değiştirildiğinde Şekil 1.2’deki yapı elde edilir. Düğümlerin etrafındaki daireler, onların kapsama alanını göstermektedir. Buna göre 1 ve 2 düğümleri birbirlerinin kapsama alanında oldukları için haberleşebilirler. 1 ve 3 düğümleri ise ancak 2 düğümü yönlendirme yaparsa haberleşebilirler. Aynı durum 4-5-6 düğümleri için de geçerlidir. Fakat 1-2-3 düğümleri, 4-5-6 düğümleri ile haberleşemezler. Bu durumda iki grup düğümün iletişimi için aralarında bir bölgede başka bir düğüme daha ihtiyaç vardır [1].



Şekil 1.2. Altyapısız kablosuz ağ.

Endüstriyel otomasyon, bina otomasyonu, çevresel kontroller ve askeri uygulamalar gibi alanlarda kullanılan sensör ağları, kuruluş yapısına bağlı olarak altyapılı veya altyapısız ağlar sınıfına girebilir. Bir sensör ağı içindeki düğümler, verilerini iletmek için herhangi bir altyapı cihazı kullanmıyorlarsa, bu ağ tasarsız ağlar sınıfında yer alır. Eğer erişim noktası cihazı gibi bir cihaz kullanıyorlarsa, altyapılı ağlar sınıfında ele alınır.

1.1. Kablosuz Ağların Tarihi ve Gelişimi

Kablosuz erişimin tarihi aslında 1896'lara dayanır. Guglielmo Marconi bu tarihte kablosuz telgrafi icat etti. 1901'de, Cornwall'dan Atlantik Okyanusu'nun diğer kıyısındaki Newfoundland'a yaklaşık 3200 km öteye ilk telgraf sinyallerini göndermeyi başardı. Bu icat, kablosuz erişim teknolojilerinin hızla gelişmesini sağladı. Telgrafın icadını radyo, televizyon, gezgin telefon ve haberleşme uyduları konusunda gelişmeler takip etti. Geçtiğimiz yüzyıl özellikle haberleşme uyduları ve gezgin hücreli telefonlarının çağı oldu.

Kablosuz bilgisayar ağlarının geçmişi ise 1970'lere dayanmaktadır. Hawaii Üniversitesi'nde Norman Abramson'un liderliğinde, ALOHAnet adı verilen bir bilgisayar ağı oluşturularak iki yönlü linklere sahip yedi bilgisayar yıldız topoloji kullanılarak kablosuz olarak birbirine bağlandı. Yine 1970'de Amerikan Savunma Bakanlığı'nın (US DoD) desteklediği Packet Radio Network (PRNET) projesinin ardından, kablosuz ağlardaki evrim, 1980'lerin başında SURAN (Survivable Adaptive Radio Networks) projesi ile devam etti. PRNET projesi ortam erişim katmanında uzaklık vektör (distance vector) algoritmasını, SURAN projesi ise bağlantı-durumu (link state) algoritmasını kullanmıştır.

1990'lı yılların başında, radyo frekans ve infrared iletişim ekipmanlarıyla donatılmış taşınabilir bilgisayarların popüler olması sebebiyle, kablosuz ağlar yeni bir gelişim sürecine girdi. Altyapısız gezgin düğümleri bir araya getiren bu süreç, IEEE 802.11 alt grubu tarafından "ad hoc networks" olarak tanımlandı.

Yine 1990'lı yılların başında DoD, tasarsız ağlarla ilgili projeleri fonlarla desteklemeye devam etti. Bu sayede Global Mobile Information Systems (GloMo) ve Near-Term Digital Radio (NTDR) projeleri ortaya çıktı. GloMo, el terminallerinin çevre birimleriyle Ethernet ile kablosuz olarak haberleşmesini sağladı. Bu proje sonucunda CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance-Taşıyıcı sezme çoklu erişim/Çarpışmadan kaçınma) ile TDMA (Time Division Multiple Access-Zaman bölmeli çoklu erişim) mekanizmaları ve birkaç yönlendirme ve topoloji kontrol şeması ortaya çıktı. NTDR, bağlantı durumu yönlendirme algoritması kullanan bir sistemdir ve hâlihazırda Amerikan ordusu tarafından da kullanılmaktadır.

1990'ların ortalarında, tasarsız ağ standartlarıyla ilgili çok sayıda çalışma yapılmıştır. İnternet ve ağ protokolleri geliştirme ile ilgili konularda çalışma yapan ve bir sivil toplum kuruluşu olan IETF (Internet Engineering Task Force) bünyesinde, MANET (Mobile Ad Hoc Networking) çalışma grubu kurulmuştur. MANET çalışma grubu, tasarsız yönlendirme algoritmalarının standartlaştırma çalışmalarına yoğunlaşmıştır. Bu süreçte, IEEE 802.11 alt komitesi tarafından, çarpışmadan kaçınmaya dayalı bir ortam erişim protokolü standart hale getirilmiştir. Böylece 802.11 PCMCIA taşınabilir bilgisayar kartları standart hale getirilip, tasarsız ağların oluşturulması için ilk prototip aşaması sağlanmıştır.

Ad hoc ağlar, ilk olarak savunma sektöründe kullanımının ardından ticari uygulamalarda da yaygın olarak kullanılmaya başlanmıştır. Taşınabilir bilgisayarlar üzerinde IEEE 802.11 kablosuz erişim kartlarının gömülü olarak gelmesi, tasarsız ağların yaygınlaşmasına katkı sağlamıştır. Günümüzde el terminalleri, navigasyon cihazları ve yüksek özellikler kazandırılmış gezgin telefonlar endüstride, kişisel ve ticari uygulamalarda yoğun bir biçimde kullanılmaktadır.

1.2. Kablosuz Tasarsız Ağların (MANETs) Karakteristikleri

Kablosuz tasarsız ağlar (MANETs-Mobile Ad Hoc Networks), keyfi ve rastgele hareket eden düğümlerden oluşur. Bu düğümler aynı zamanda, kablolu ağlarda

kullanılan yönlendiricilerin yaptığı yönlendirme görevini de yerine getirirler. Kablolu ağlardan farklı olarak, MANET’te veri düğümünden düğüme yönlendirilerek iletilir [2].

MANET teknolojisi, endüstriyel ve ticari uygulamalarda yaygın olarak kullanılır. Gezgin düğümler hava alanlarında, gemilerde, kamyonlarda, otomobillerde, insanların üzerinde ve çok küçük cihazların içinde kullanılmaktadır. Kullanım yerleri, düğümlerin hareketlilik özelliğine sahip olmasını zorunlu kılmaktadır.

Kablosuz gezgin düğümler, gönderici, alıcı ve anten içerirler. Antenler, yönlendirmesiz (omnidirectional), yönlendirmeli (directional), yönetilebilir-akıllı (smart) veya bunların birleşimi şeklinde yapıya sahip olabilirler. Yönlendirmesiz antenler, dairesel bir yapıda her yöne iletim yaparlar ve yayın (broadcast) tabanlı uygulamalarda kullanılırlar. Yönlendirmeli antenler, belli bir yönde açılabilir olarak iletim yaparlar ve noktadan noktaya (point to point) uygulamalarda kullanılırlar. Akıllı antenler ise programlanabilir bir biçimde yönlendirme yaparlar ve istenilen tür uygulamada kullanılabilirler.

Kablosuz tasarsız ağların sahip olduğu karakteristik özellikler aşağıda sunulmaktadır:

1. Değişken topoloji: Düğümlerin serbest olarak hareket etmesi sonucu ağ topolojisinde değişiklikler oluşur. Düğüm konumları bilinmeyen zamanlarda ve hızla değişir.
2. Sınırlı bant genişliği: Kablosuz ağlar, kablolu ağlardan çok daha düşük link kapasitesine sahiptirler. Ayrıca gerçekleşen link kapasitesinin, radyo frekans bölgesinin en yüksek kapasitesinden düşük kalmasının sebebi; çoklu erişim, solma (fading), gürültü ve etkileşimdir. Ayrıca MANET kullanıcıları, kablolu ağların sağladığı yoğun trafik gerektiren çoklu ortam uygulamalarının servislerini kullanmak istediklerinde ağda tıkanmalar da ortaya çıkabilir.
3. Kısıtlı enerji: MANET içindeki düğümlerin bazıları veya tümü, enerji kaynağı olarak batarya veya diğer tükenbilir kaynaklar kullanırlar. Bu düğümler için enerji kullanımı optimizasyonu en önemli tasarım kriteri olmalıdır.
4. Sınırlı fiziksel güvenlik: Gezgin kablosuz ağlar, kablolu ağlara göre daha yüksek fiziksel güvenlik tehlikesi ile karşı karşıyadır. Başkaları tarafından

dinlenme, hizmet dışı bırakma atakları gibi artan tehditlerin göz önünde bulundurulması gereklidir.

Kablosuz tasarsız ağların bu karakteristikleri, yönlendirme algoritması geliştirilme işlemlerinde göz önünde bulundurulması gerekmektedir.

1.3. Kablosuz Tasarsız Ağlar için Yönlendirme Algoritmaları ve Literatürde Yer Alan Geliştirme Çalışmaları

Bu bölümde literatürde yer alan tasarsız yönlendirme algoritmalarına ve onların üzerinde yapılan iyileştirme çalışmalarına yer verilmiştir. Yönlendirme algoritmalarının çalışma prensipleri 2. Bölüm'de ayrıca ele alınmaktadır.

Kablolu ağlarda en kısa yol (shortest path) bulma işlemi, genellikle mesafe vektörü (distance vector) veya bağlantı durumu (link state) yönlendirme algoritmalarını esas alan protokoller kullanılarak gerçekleştirilir. Bu algoritmalar kablosuz tasarsız ağlarda verimli çalışmazlar. Çünkü kablosuz tasarsız ağlar sınırlı bant genişliğine sahiptir ve merkezi kontrol yoktur. Bu yüzden kablosuz tasarsız ağlar, bu protokoller üzerinde değişiklikler yapılarak veya tamamen yeni protokoller geliştirilerek yönlendirilebilir [3,4,5].

Tasarsız kablosuz ağlardaki yönlendirme protokolleri, tabloya dayalı (table driven), isteğe bağlı (On-demand) ve melez (hybrid) yönlendirme protokolleri olmak üzere üç ana sınıfa ayrılır. Geliştirilmiş algoritmalar, ortaya çıkış tarihlerine göre aşağıda verilmiştir.

Tabloya dayalı yönlendirme algoritmaları kablolu ağlarda kullanılan RIP (Routing Information Protocol), uzaklık vektörü (distance-vector), OSPF (Open Shortest Path First) ve bağlantı durumu (link-state) algoritmalarına benzer. Bu algoritmalar tüm ağa ait yönlendirme bilgilerini sürekli güncel tutmaya çalışırlar. Ağdaki her bir düğüm, yönlendirme bilgilerini bir veya daha fazla yol tablosunda depolarlar ve ağ topolojisindeki değişiklikleri yayın yoluyla birbirlerine bildirirler. Bu kategorideki yönlendirme algoritmaları şunlardır: DSDV (Destination Sequenced Distance-

Vector, 1994), WRP (Wireless Routing Protocol, 1995), CGSR (Clusterhead Gateway Switch Routing, 1997), GSR (Global State Routing, 1998), FSR (Fisheye State Routing, 1999), HSR (Hierarchical State Routing, 1999), ZHLS (Zone based Hierarchical Link State, 1999), STAR (Source Tree Adaptive Routing, 2000), WRP-Lite (Wireless Routing Protocol-Lite, 2000), IARP (Intrazone Routing Protocol, 2002), DFR (Direction Forward Routing, 2006).

İsteğe bağlı algoritmalar, geleneksel yol tablolarını ve topoloji değişimlerinde güncelleme işlemini devre dışı bırakmaya çalışır. Tabloya dayalı algoritmaların aksine yönlendirme bilgileri sürekli değil sadece istenildiğinde oluşturulur. Kaynak bir düğüm, bir hedefe paket göndermek istediğinde, hedefi bulabilmek için yol bulma prosedürü başlatır. Bu kategorideki algoritmalar şunlardır: DSR (Dynamic Source Routing, 1996), ABR (Associativity Based Routing, 1996), TORA (Temporally-Ordered Routing Algorithm, 1997), SSR (Signal Stability Routing, 1997), PAR (Power-Aware Routing, 1998), LAR (Location Aided Routing, 1998), CBR (Cluster Based Routing, 1999), AODV (Ad hoc On-Demand Distance Vector Routing, 1999), Dynamic NIX-Vector Routing (2005), DYMO (DYnamic Manet On-demand Routing, 2006), MAODV (Multirate Ad hoc On-Demand Distance Vector Routing, 2007).

Melez yönlendirme algoritmaları ise isteğe bağlı ve tabloya dayalı kategorilerine girmeyen veya her ikisinin özelliklerini taşıyan algoritmalarlardır. Konum bilgisine dayalı algoritmalar da bu sınıfa dâhil edilebilir. Bu kategoride yer alan algoritmalar şunlardır: MFR (Most Forward With in Radius, 1984), DIR (Directional Routing, 1999), GEDIR (Geographical Distance Routing, 1999), ZRP (Zone Routing Protocol, 2002).

Kablosuz tasarsız ağların Bölüm 1.2’de verilen kısıtlarından kaynaklanan dezavantajları ortadan kaldırmak veya azaltmak için, yönlendirme algoritmaları üzerinde değişiklik yapılmıştır. Literatürde yer alan geliştirme çalışmaları şu konulara yoğunlaşmıştır:

- a. Table driven (tabloya dayalı) ve On demand (isteğe bağlı) protokollerin yönlendirme ek yükünü azaltma ile ilgili çalışmalar [6,7]

- b. Enerji tüketimini azaltmaya yönelik yönlendirme çalışmaları [8,9,10,11]
- c. Simetrik (çift yönlü-bidirectional) linklerin olmaması veya link bozulmalarını iyileştirme ile ilgili çalışmalar [12]
- d. Tasarsız ağlarda güvenlik artırma ile ilgili yönlendirme çalışmaları [13]
- e. Konum bilgilerini kullanarak yönlendirme kararı veren algoritmalar üzerinde yapılan geliştirme çalışmaları [14,15,16,17]

Kablosuz ağlarda kullanılan algoritmaların performanslarını karşılaştırmak için yapılmış çalışmalar da yayınlanmıştır. Bu çalışmalar, protokollerin hareketlilik modellerini ve performans ölçülerini ele almışlardır. Monarch projesi kapsamında yapılan bir çalışmada DSDV, AODV, DSR ve TORA protokollerini karşılaştırmak için kullanılan performans ölçütleri verilmiştir [18]. [19]'da isteğe bağlı protokollerle aynı kısıtlamaları kullanan bir tabloya dayalı distance vector yönlendirme protokolü önerilmiştir. Düğümler arası linkler olabildiğince geçerli olabilecek ve güncellemeler yalnızca linklerden biri geçersiz olduğunda gönderilecek bu öneriye WRP-lite adı verilmiştir. Bu isimlendirmenin kullanılmasının sebebi, WRP'de tutulan tablo sayısının azaltılmasını hedeflemesidir.

Geliştirilen bazı yönlendirme protokolleri ağ yükünü dengelemeyi, tıkanıklığı azaltmayı ve kararlı paket dağıtımını sağlamayı hedeflemiştir. [20]'de LEE ve arkadaşları tarafından 2001 yılında Load Aware Routing (LAR) algoritması tasarlanmıştır. LAR, ara düğümlerin yönlendirme yükünü birincil yol seçim ölçütü olarak kullanmaktadır. Algoritmaya göre, ağa gönderilen ve ağda dolaşan tüm yol istek paketleri her bir düğümün tamponunda tutulur. Veri paketi gönderecek düğüm hedefi seçerken tamponundaki bilgilere göre en iyi seçimi yapar. Bu yöntemde tek yönlü yol seçim mekanizması kullanıldığı için yük dengeleme kapasitesi sınırlıdır. [21]'deki Alternate Path Routing (APR, 2000 yılı) ve [22]'deki Multi-Path Routing (MPR, 2002 yılı) algoritmalarında, farklı yolları belirleme en önemli konudur. Bu algoritmalarda yol belirleme için, kaynaktan hedefe olan çoklu yollar bulunur. Amaç, trafiği çoklu yollar üzerine dağıtmaktır. APR, MPR ve LAR'ın eksikliklerini gidermek için [23]'de Dynamic Nix-Vector Routing (DNVR, 2005 yılı) algoritması tasarlanmıştır. Bu algoritma ile trafik hacmini ve hareketliliği ölçeklemeyi sağlayan bir yük dengeleme yöntemi önerilmiştir. DNVR'ye göre, depolanmış yol bilgileri

kullanılmadan önce doğrulanır, sonrasında ağ topolojisi güncellenir ve hedefe olan en iyi birkaç yol seçilerek yönlendirme gerçekleştirilir.

Performans artırma ile ilgili yapılan çalışmalardan birinde Beacon-less Routing (BLR, 2004 yılı) algoritması önerilmektedir [7]. BLR yönlendirme ek yükünü azaltmak için konum bilgisini kullanır. Periyodik yayın paketlerine (Hello mesajı) ihtiyaç duymayarak batarya kullanımından ve performanstan kazanç sağlamayı hedefler. BLR, komşuları arasından bir iletici seçerek, onların ayakta olup olmadığını ve konum bilgilerini kullanır. En iyi ileticiyi seçerek paket gönderme işlevi gerçekleştirir.

[12]'de yapılan çalışmada, ikinci ve üçüncü katmandaki kontrol döngülerini kullanarak ağdaki link kopmalarını önceden belirleme öne sürülmektedir. Bu yöntem IEEE 802.11a'da kullanılan link uyarılma mekanizmasına dayanır. Link uyarılma işlemi, link sürekliliği ve link ömrünü tahmin bilgisini içerir.

[9]'da düğümlerin enerji tüketimlerini azaltmak için, yönlendirilmiş antenler kullanma önerilmektedir. Kablosuz düğümlerin kullandığı yönlendirilmiş antenlerin, belirli bir yönde gönderim yapmasından dolayı enerji tüketimini azaltma öne sürülmüştür. [10]'da düğümlerin enerji tüketimini azaltmak için MAC katmanında değişiklik yapılması önerilmiştir. [11]'de Maximum Residual Energy Path (MREP) algoritmasının gönderme ve alma yönünde harcanan enerjileri azaltma ile ilgili önerilerde bulunulmuştur.

[24]'de düğümlerin iletişim performansını artırmayı ve güvenli iletişim kurmayı hedefleyen RELiable and Efficient Forwarding (REEF) algoritması öne sürülmektedir. REEF, ağ tıkanıklığını azaltmayı ve kaynak-hedef çiftleri arasında güvenli linkler kurmayı ele almaktadır.

Konum bilgilerini kullanarak iyileştirme için geliştirilen algoritmalarından biri [14]'de önerilmiştir. Distance Routing Effect Algorithm For Mobility (DREAM) algoritmasına göre, her düğümün yönlendirme tablosunda, ağdaki tüm düğümlerin konum bilgileri depolanmakta ve konum bilgileri GPS kullanılarak elde

edilmektedir. Bu algoritma tüm düğümlere ait bilgi bulundurduğu için tabloya dayalı bir algoritmadır. Algoritmaya göre, ağdaki bir A düğümü m mesajını B düğümüne gönderirken, B 'nin yönünü belirleyebilmek için onun konum bilgilerini kullanır. Ardından m mesajını, B yönündeki 1-hop komşularına gönderir. Her bir komşu aynı işlemleri tekrar eder. Bu işlemin mesaj B 'ye ulaşmaya kadar (eğer mümkünse) devam etmesi yönüyle de, isteğe bağlı algoritmalara benzer. Algoritmada mesafe ve hareketlilik önemli rol oynadığı için DREAM adı verildiği ifade edilmektedir.

Konum tabanlı algoritma çalışmalarından biri [15]'de ileri sürülmüştür. V-GEDIR adı verilen bu algoritma, Geographic Distance Routing (GEDIR)'e dayanmaktadır. Bu yöntemde, kesişim düğümleri hedefin olası konumlarının dairesel veya dikdörtgensel voronoi diyagramı ile belirlenir. Bir diğer konum tabanlı algoritma, yol isteği iletilen düğüm sayısını azaltmayı önermektedir [16]. [17]'de önerilen Location Information Aided Routing (LAR) algoritması, arama bölgesini azaltarak yol bulmayı ele almaktadır.

1.4. Çalışmanın Amacı, Önerilen Çözüm Yöntemi ve Katkıları

Tasarsız kablosuz ağlar; merkezi kontrol olmaması, tüm düğümlerin kablosuz arayüze sahip olması, düğümlerin serbestçe hareketi sonucu sıklıkla oluşan topoloji değişikliği, düğümlerin sınırlı miktarda kaynağa (bant genişliği ve batarya ömrü gibi) sahip olması ve simetrik (iki yönlü) linklerin eksikliği gibi karakteristiklerden dolayı diğer ağlardan ayrılır.

Gezgin kablosuz ağlarla ilgilenilmek istendiğinde, “mobil” ve “kablosuz” kelimelerinden kaynaklanan bazı ihtiyaçlar ortaya çıkar. Gezginlik, düğümlerin hareketinden dolayı sınırlı ömre sahip komşuluk veya sürekli değişen topoloji bilgisini ima eder. Bu durum, bilginin düzenli olarak yenilenmesi aksi halde geçersiz olacağı anlamına gelir. Bilgi ne kadar sık güncellenirse, düğümlerin gezginliği o kadar doğru biçimde elde edilebilir. Kablosuz ortam ise kullanılabilir frekans bandında, sınırlı bant genişliğini ima eder. Bant genişliğinin sınırlı olması, kablosuz dünyada kıymetli kaynakların akıllıca tüketilmesi ihtiyacını doğurur. Bu yüzden

kablosuz ađlar için bir protokol tasarlanırken, ilk görev bant genişliğinin gereksiz kullanımını azaltmak olmalıdır.

“Gezginlik” ve “kablosuz” kavramları, sahip oldukları gereksinimleri açısından birbirine zıttır. Gezginlik, diğer düğümlerdeki deęişikliği hissedebilmek için ađda dolaşan bilgi trafiğinin çok olmasını gerektirir. Diğer taraftan, kablosuz ortam kapasiteyi boşa kullanmamak için gereksiz veya fazla trafiği istemez. Sınırlı bant genişliği kullanılırken düğümlerin hareketliliğini belirleyebilecek bir mekanizma gereklidir.

Kablosuz ađlarda kıt olan kaynakların etkin bir biçimde kullanılması için yönlendirme algoritmalarının iyileştirilmesi gerekir. Yönlendirme algoritmalarının kontrol ve bağlantı için harcadığı bant genişliği ve buna baęlı olarak batarya tüketiminin de en aza indirilmesi gerekir.

Bu tez çalışmasında, tasarsız ađların en önemli iki kaynağı olan bant genişliği ve batarya ömrü kullanımını optimize etmeyi saęlayan yeni bir yönlendirme algoritması geliştirilmiştir. Düğümlerin konum bilgilerinin, içlerinde tümleşik olarak bulundukları GPS (Global Positioning System) ile elde edilen ve “Konum Tabanlı Melez Yönlendirme Algoritması” adı verilen bu algoritma ile yönlendirme ek yükü azaltılarak bant genişliğinin etkili bir şekilde kullanımı saęlanmaktadır. Bununla birlikte, ađdaki her bir düğüm tarafından yönlendirme amacıyla yapılacak işlem sayısı ve bellekte tutulacak veri miktarı azaltılarak batarya ömrü de etkin bir biçimde kullanılmaktadır. Bu hedefleri gerçekleştirmek için hem tabloya dayalı, hem de isteğe baęlı algoritmaların çalışma mantığından faydalanılmıştır. Ancak önerilen yöntem bu iki yöntemin de dışında yeni bir yaklaşımdır.

Bilindiği gibi altyapılı kablosuz ađlarda merkezi bir istasyon veya düğüm vardır ve bu istasyon hareketsizdir. Bu istasyonun kapsama alanında olan düğümler yönlendirme için gerekli olan bilgileri bu istasyondan öğrenirler ve ayrıca veri gönderme-alma işlemini de bu istasyon üzerinden gerçekleştirirler. Ancak tasarsız ađlarda merkezi bir düğüm olmadığı veya diğer bir deyişle bütün düğümler hareketli olduğu için altyapılı ađlardaki çalışma yöntemi tasarsız ađlarda doğrudan

kullanılmaz. Fakat altyapılı kablosuz ağların çalışma mantığı, tasarsız ağlara uyarlanabilir. Bu noktada, geliştirdiğimiz KTMYA’da altyapılı ağlarda olduğu gibi merkezi bir düğüm atanır ve yönlendirme bilgilerini yönetir. Bu merkezi düğüme “master” düğüm adı verilir. Düğümler, bir hedef düğüme veri göndermek istediklerinde hedef düğümün yerini ve ona gidecek en kısa yolu master düğümden öğrenirler ve bu yol üzerinden verilerini gönderirler. Bu aşamada algoritma altyapılı ağlardan ayrılır. Çünkü altyapılı ağlarda veri de merkezi istasyon üzerinden gönderilir. Ancak KTMYA’da merkezi istasyon olarak davranan master düğüm sadece hedefe ait yolu bulmada kullanılır.

Geliştirilen algoritma, fiziksel, veri-bağlantı, ağ, aktarım, sunum, oturum ve uygulama olarak yedi katmandan oluşan OSI (Open System Interconnect) referans modelinin 3. katmanı olan ağ katmanında çalışır. Alt katmanda bulunan ortam erişim kontrolü (MAC) katmanının servislerini kullanır ve üst katmandaki aktarım katmanına servis sağlar. Bu çalışmada MAC katmanı olarak IEEE 802.11 kullanılmış ve yeni bir öneride bulunulmamıştır.

Bu tez çalışmasında öne sürülen ve bilimsel katkı sağlamayı amaçlayan çalışmalar şöyle sıralanabilir:

1. Düğümlerin konum bilgilerini kullanarak yönlendirme kararı veren KTMYA’yı geliştirmek,
2. Geliştirilen algoritmayı kullanan protokol tasarımını gerçekleştirmek için paket yapılarını çıkarmak,
3. Konum bilgilerini kullanarak yönlendirme kararı veren klasik algoritmalarda da kullanılan düğümler arası maliyet değerini oluşturmak için bulanık mantık yöntemini uygulamak ve bulanık mantık kullanılmamış duruma göre de üstünlükleri ortaya çıkarmak,

4. Ağdaki düğüm sayısının artması durumunda, ağın ölçeklendirilebilmesi için fuzzy C-means algoritmasını kullanarak kümelere ayırma işlemini yapabilmek,
5. Geliştirilen algoritmanın modellenmesi ve performans değerlendirmesini yapmak için MATLAB 7.0.1 kullanılarak yazılım geliştirmektir.

Geliştirilen algoritma ile mevcut algoritmaların aşağıdaki problemlerine çözümler oluşturulmuştur:

1. Tabloya dayalı algoritmalar, yönlendirme bilgilerini oluşturabilmek için sürekli olarak yönlendirme tablosunu güncellerler ve bu işlem için yayın tabanlı paketler kullanırlar. Bu durum yönlendirme ek yükünün artmasına yol açar. KTMYA'ya göre bütün düğümlerde yol tablosu tutulmaz. Sadece master düğüm üzerinde yol tablosu tutulur. Periyodik güncellemeler kullanılmaz, olay tabanlı güncellemeler kullanılır. Ayrıca olay tabanlı güncelleme işlemi, yayın (broadcast) yöntemiyle yapılmaz. Güncelleme işlemleri, hedefi, ağın merkezinde yer alan ve yönlendirme kararını veren düğüm (master düğüm) olan paketlerle yapılır.
2. İsteğe bağlı algoritmalar sadece bir düğümün yol ihtiyacı olduğunda keşif paketleri ile yol oluştururlar. Bu durum, uçtan uca paket gönderim zamanının uzamasına yol açar. KTMYA'da yol istekleri master düğüm tarafından sağlandığı için, bu süre azaltılmıştır.
3. İsteğe bağlı ve tabloya dayalı algoritmalarda yol bulmak için yapılan işlem sayısı fazla olduğu için enerji tüketimi yüksektir. KTMYA ile işlem sayısı ve yönlendirme ek yükü azaltıldığı için enerji tüketimi iyileştirilmiştir.
4. Tabloya dayalı algoritmalarda, ağdaki düğüm sayısının artması durumunda, her bir düğümde bulundurulmuş yol tablolarının büyümesi ve sürekli periyodik güncellemelerden dolayı, ağın verimi düşer. KTMYA'da periyodik

güncellemeler gönderilmez. Ayrıca ağdaki düğüm sayısının artması durumunda, ağın kümelere ayrılması çözümü de önerilmiştir.

5. İsteğe bağlı algoritmalarda, hareketliliğin yüksek olması durumunda, yol keşif paketlerinin yol bulma süresi ve yol bulamama ihtimali artar. Oysa KTMYA'da hareket durumunda konum güncelleme bilgileri gönderimiyle bu problem çözülmüştür.
6. DSDV, AODV ve TORA algoritmalarında, her bir düğümde tutulan yol tablosu yol bulma işlemlerinde tüm düğümlerin yer alması enerji tüketimini artırır. KTMYA'da ise, master düğüm dışındaki düğümlerde yol tablosu bulundurulmadığı ve tüm düğümlerin yol bulma işlemine katılmamasından dolayı enerji tüketimi azaltılmıştır.
7. Konum tabanlı melez yönlendirme algoritmalarından DREAM, MFR, GEDIR ve DIR, yönlendirme kararını vermek için sadece düğümler arası mesafeleri kullanırlar. Bu durumda, yoğun bir düğüm yol üzerinde bulunursa, uçtan uca paket gecikmesi artar. KTMYA'da yönlendirme kararını vermek için sadece düğümler arası mesafe değil, buna ek olarak düğüm yoğunluğu ve batarya miktarı bilgisi de kullanılarak bulanık mantık yöntemiyle yönlendirme kararı verme öne sürülmüştür.

Aşağıda sayılan konularda ise iyileştirme yönünde herhangi bir çözüm önerisinde bulunulmamıştır:

1. Geliştirilen algorithmada, ağın merkezinde yer alan ve yönlendirme kararını veren düğümü (master düğüm), ağdaki diğer düğümlere duyuran paketler yayın tabanlı paketlerdir. Yayın paketleri kullanmadan bir yöntem geliştirme konusunda her hangi bir çözüm önerilmemiştir.
2. Geliştirilen algorithmaya göre, ağdaki düğümlerin konum, batarya ömrü ve yoğunluk bilgilerini içeren güncelleme paketleri için, gidiş yolu olarak, master düğümden gelen duyuru paketlerinin ters yolu kullanılmaktadır. Bu durum düğümler arası bağlantıların simetrik (çift yönlü) olmasını zorunlu kılar. Tek

yönlü (unidirectional) linklerin olduğu durumlarla ilgili herhangi bir çözüm önerilmemiştir.

1.5. Tez Organizasyonu

Bu tez çalışması aşağıda özetleri verilen bölümler şeklinde organize edilmiştir:

1. Bölümde, kablosuz ağların tarihsel gelişimi ve kullanım alanlarına yer verilmektedir. Literatürde yer alan yönlendirme algoritmaları ele alınmakta ve geliştirme çalışmaları hakkında bilgi verilmektedir. Bu tez çalışmasının amacı olan ve yeni geliştirilen yönlendirme algoritmasının, literatürde yer alan algoritmalarla karşılaştırması yapılmaktadır.

2. Bölümde, kablosuz tasarsız ağlar için geliştirilmiş yönlendirme algoritmalarının sınıflandırması yapıp, her bir sınıfın çalışma prensipleri hakkında bilgi verilmiştir. Ayrıca algoritmaların yetenekleri hakkında da bir karşılaştırma tablosu oluşturulmaktadır.

3. Bölümde, geliştirilen yönlendirme algoritmasının sınıflamadaki yeri belirlenmekte ve çalışma prensibi de ayrıntılarıyla ele alınmaktadır. Bu bölümde algoritmanın matematiksel ayrıntıları verilerek protokole geçiş için kullanılacak paket çerçeve yapıları oluşturulmaktadır. Algoritmada öne çıkan bulanık mantıkla “maliyet” değerinin belirlenmesi işlemi üzerinde durulmakta ve bulanık mantık kullanılmadığı duruma göre karşılaştırma işlemleri de yapılmaktadır. Ayrıca ağdaki düğüm sayısının ve düğümlerin dağınıklığının artması durumunda ölçekleme için kümeleme çözümü üzerinde durulmaktadır.

4. Bölümde, geliştirilen algoritma MATLAB 7.0.1’de kodlanan bir benzetim programı ile performans değerlendirmesine tabi tutulmakta, başarı oranı, yönlendirme ek yükü ve enerji tüketimi açısından diğer yönlendirme algoritmaları ile karşılaştırılmaktadır.

5. Bölümde, önceki bölümde elde edilen performans sonuçları değerlendirilmekte ve bu çalışma ışığında yapılabilecek çalışmalarla ilgili önerilerde bulunmaktadır.

Ek A'da, Bölüm 3'te kullanılan graf teorisi hakkında bilgiler verilmektedir.

Ek B'de, Bölüm 3'te algoritmaya uygulanan bulanık mantık yöntemi detaylı olarak ele alınmaktadır.

Ek C'de, 3. Bölüm'de KTMYA'ya uygulanan bulanık mantık yönteminde oluşturulan bulanık kuralların tamamı verilmektedir.

Ek D'de, kablosuz ağlar için enerji tüketim analizi yapılmakta ve küresel yer belirleme sistemi (GPS-Global Positioning System) cihazlarının enerji tüketimleri ele alınmaktadır. Ayrıca kapasitesi bilinen bir bataryanın paket alma veya gönderme yeteneği belirlenmektedir.

Ek E'de, literatürde yer alan kümeleme algoritmaları ve çalışma prensipleri hakkında bilgi verilmektedir.

Ek F'de, algoritmanın benzetimini yapmak için geliştirilen MATLAB .m dosyalarının içerikleri verilmektedir.

BÖLÜM 2. GEZGİN TASARSIZ AĞLARDA YÖNLENDİRME

Bölüm 1’de literatür özeti verilen algoritmalarından, performans değerlendirme işlemlerinde sık kullanılan algoritmaların sınıflandırması ve çalışma prensipleri bu bölümde ele alınmıştır. Ayrıca algoritmalar için bölüm sonunda karşılaştırma ve kontrol tablosu da oluşturulmaktadır.

2.1. Gezgin Tasarsız Ağlarda Yönlendirme Algoritmalarının Sınıflandırması

Kablosuz tasarsız ağlarda kullanılan yönlendirme algoritmaları; tabloya dayalı, isteğe bağlı ve melez algoritmalar olmak üzere üç ana kategoriye ayrılır. Her bir kategori alt kategorileri ile birlikte aşağıda ifade edilmiştir.

- Tabloya dayalı (table-driven) yönlendirme algoritmaları
 - Destination Sequenced Distance Vector (DSDV) [5,25,26]
 - Wireless Routing Protocol (WRP) [27]
 - Clustered Gateway Switch Routing (CGSR) [28]
 - Optimized Link State Routing (OLSR) [29]
- İsteğe bağlı (on-demand) yönlendirme algoritmaları
 - Dynamic Source Routing (DSR) [30,31]
 - On-Demand Distance Vector Routing (AODV) [3,30,32,33]
 - Temporally Ordered Routing Algorithm (TORA) [3]
 - Associativity Based Routing (ABR)[34]
 - Zone Routing Protocol (ZRP) [35]
 - Signal Stability Based Adaptive Routing (SSA) [36]
- Melez (hybrid) yönlendirme algoritmaları
 - Konum tabanlı algoritmalar [37]
 - Geographic Distance Routing (GEDIR) [37]
 - Directional Routing Algorithm (DIR) [38]

- Most Forward within Radius (MFR) [39]
- Distance Routing Effect Algorithm For Mobility (DREAM) [14]
- Multi Point Relaying (MPR) tabanlı algoritmalar [25]

Bir sonraki bölümde, her bir sınıfın genel çalışma prensipleri ele alınmaktadır. Ayrıca, bu bölümde, her bir sınıftan öne çıkan ve performans değerlendirme işlemlerinde çoğunlukla ele alınan algoritmaların yönlendirme esasları ve avantaj-dezavantajlarına da yer verilmektedir.

2.2. Tabloya Dayalı (Table-Driven) Yönlendirme Algoritmaları

Tabloya dayalı yönlendirme algoritmaları aynı zamanda ihtiyatlı (proaktif) olarak da adlandırılır. Bu algoritmayı kullanan protokoller, ağdaki tüm kaynak-hedef çiftleri arasındaki yolları bulurlar ve periyodik yol güncellemeleriyle en yeni yol bilgilerini oluştururlar. Güncelleme mesajları ağ topolojisinde hiç değişiklik olmasa bile gönderilir. Bu kategorideki protokoller uzaklık vektör (distance vector) ve bağlantı durum (link state) algoritmaları değiştirilerek geliştirilmiştir. Protokoller yönlendirme bilgilerini yönlendirme tablolarında saklarlar. Periyodik güncellemeler sayesinde bu protokoller çok yavaş sonuca ulaşabilirler ve çok miktarda yönlendirme ek yükü oluştururlar. Bu durum çok düğümlü tasarsız kablosuz ağlar için çok uygun değildir [3].

2.2.1. Destination sequenced distance vector algoritması

Bu protokol Routing Information Protocol (RIP) üzerine geliştirilmiş bir uyarlamadır. RIP'teki yönlendirme tablosuna "sıra numarası" alanı eklenmesi ile RIP'ten farklılaşır. Sıra numarası alanı, eski ve yeni yolları birbirinden ayırt etmede kullanılır.

Her bir düğüm erişebileceği hedefler yönündeki sonraki düğüm bilgisini içeren yönlendirme tablosu oluşturur. Yönlendirme tablosundaki her bir kayıt, hedef düğümün adresi, hedefe ulaşmak için atlanması gereken düğüm sayısı ve o hedeften aldığı sıra numarası bilgilerini içerir. Sıra numaraları her bir yolun yeniliğinin

belirlenmesi için birleştirilir. Bir düğüm belli bir yoldan yeni bir bilgi alırsa önceki sıra numarasıyla sonraki sıra numarasını karşılaştırır ve büyük sıra numaralı satırı tutup diğerini yok sayar. Eğer aynı sıra numarasına sahip iki güncelleme alırsa, atlama sayısı düşük olanı dikkate alır.

Yönlendirme tablosu periyodik duyurularla veya yeni bir bilgi alındığı zaman güncellenir. Düğümler, sürekli veya artışı olmak üzere iki tür güncelleme gönderirler. Sürekli güncellemeler periyodik olarak gönderilirken, artışı güncellemeler yol değişikliği gibi durumlarda olay tabanlı olarak gönderilirler. Sürekli güncellemede tüm yönlendirme tablosu gönderilirken, artışı güncellemede sadece en son güncellenen bilgi satırı gönderilir [3,26].

Protokolün performansı periyodik güncellemelerin gönderilme aralığı değerine büyük ölçüde bağlıdır. Bu aralık çok küçükse, çok büyük miktarda yönlendirme ek yükü oluşur. Bu ek yük sürekli ve artışı güncellemelerin çok sık gönderilmesinden kaynaklanır. Eğer aralık büyük olursa, en son yol bilgilerinin alınmasında bir gecikme olur. Eğer ağdaki düğümler çoksa ve hareketlilerse bu protokol elverişli değildir [3].

2.2.2. Wireless routing protocol

WRP protokolünde her bir düğüm, mesafe, yönlendirme, bağlantı-maliyet ve mesaj tekrarlama listesi olmak üzere dört tablo kullanır. Mesaj tekrarlama listesi, güncelleme mesajlarının sıra numaralarını ve güncelleme mesajlarını bulundurur. Mesaj tekrarlama sayısı ne kadar fazla olursa, bağlantı kurma başarısı o kadar yüksek olur. Yönlendirme işlemi için tablo sayısının fazlalığı bu protokolü kullanan düğümlerde ağ büyüklüğü ile orantılı olarak önemli bellek ihtiyacı gerektirir [12,40].

WRP'yi tanımlamak için yönlendirilmemiş graf modeli kullanılır (Yönlendirilmemiş graflarla ilgili bilgi bu tezin Ek A bölümünde yer almaktadır). $G(V,E)$ grafında V düğümler kümesi ve E linkler veya kenarlar kümesi olarak ifade edilir. V kümesindeki her bir düğüm bir yönlendiriciyi gösterir. Bu yönlendiriciler işlemci, bellek ve giriş çıkış tamponlarına sahiptir. Kablosuz ağda bir düğüm, birden çok

düğümle radyo bağlantısına sahiptir. Yönlendirme tablolarının güncellenmesi amacıyla, bir A düğümü eğer aralarında radyo bağlantısı varsa, komşusu olan bir B düğümünden güncelleme mesajları alır. Bu yüzden çoklu düğümlerle noktadan noktaya yayın yoluyla bağlantı kurulur. Kurulan linkler için her iki yöne pozitif ağırlık değerleri atanarak çift yönlü graf kenarları oluşturulur. Bir düğüm tarafından alınan tüm mesajlar giriş kuyruğuna alınır. İlk giren ilk çıkar (FIFO) algoritmasına göre mesajlar işlem görür. Ağdaki güncelleme mesajları gibi haberleşme linkleri, işlemsel kenarlar üzerinden sonlu zamanlarda gönderilir. İki düğüm arasındaki radyo bağlantısı koparsa, düğümlerin mesafe ve yönlendirme tablolarındaki kayıtlar sonsuz olarak işaretlenir [27].

2.3. İsteğe Bağlı Yönlendirme Algoritmaları

İsteğe bağlı yönlendirme algoritmaları düğümler arası kullanılacak yol bilgilerini sürekli oluşturmazlar. Yollar sadece gerektiğinde yani düğümlerden herhangi birisi paket göndermek istediğinde oluşturulur. Bu yüzden bu kategorideki protokollere isteğe bağlı (on-demand) yönlendirme protokolleri denir.

Literatürde yer alan isteğe bağlı yönlendirme protokollerinden bazıları şunlardır: Dynamic Source Routing (DSR), On-Demand Distance Vector Routing (AODV), Temporally Ordered Routing Algorithm (TORA).

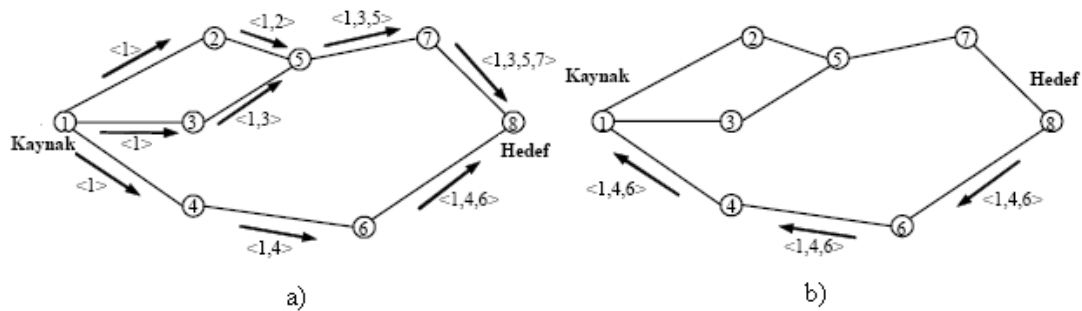
2.3.1. Dynamic source routing

Kaynaktan yönlendirme algoritmasında, gönderici düğüm, gönderdiği paketin izleyeceği tüm yolu belirler ve belirlediği yol bilgisini paketin başlık kısmına ekler. Bu işlem statik veya dinamik olarak yapılabilir. DSR protokolü dinamik olan kaynak yönlendirmesini kullanır [30].

Her bir düğüm bir yol önbelleği tutar. Yol önbelleğinde farklı düğümlere olan tüm yollar bulunur. Yol belirleme ya önbellekten bir istemde bulunularak ya da yol bulma işlemi ile yapılır. Kaynak düğüm, hedef düğüme bir paket göndermek istediğinde ilk olarak önbelleğini kontrol eder. Eğer ön bellekte o hedefe ait bir kayıt varsa

doğrudan o yol kullanılır. Aksi halde bir yol bulma işlemi başlatılır ve bu işlem hedefe olan yol tamamen belirleninceye kadar tekrarlamalı olarak devam eder. Bir yol bulunmaz bulunmaz paketin başlığına eklenerek paketin iletilmesi sağlanır [31].

Yol belirleme işleminde gönderici, “yol istek paketi” oluşturur. Yol istek paketi, içinde sonraki düğüm bilgisini içeren ve biricik olan istek kimlik numarası (id) ile tutulan bir Yol kaydı bulundurur. Kaynak adres-istek id çifti her bir Yol istek paketini biricik olarak tanımlar. Kaynak düğüm Yol istek paketini yayınlar. Tüm komşular bu isteği alırlar. Şekil 2.1a’da bu adımlar ifade edilmiştir. Komşulardan birinin ön belleğinde hedef düğümüne karşılık gelen bir kayıt varsa, istek sahibine tüm yolu içeren cevap gönderir. Eğer ön belleğinde bir kayıt bulunmazsa istek paketini daha sonraki düğümlere gönderir [42,43]. Şekil 2.1b’de yol istek paketine gönderilen paketin izlediği yol gösterilmiştir [41].



Şekil 2.1a. DSR protokolünde yol bulma mekanizması ve istek paketinin izlediği yol [41]

Şekil 2.1.b. Yol cevap paketinin izlediği yol [41]

Bir yol cevap paketi kaynak düğümüne, diğer düğümün yönlendirme tablosundan gelebilir. Bunun sebebi düğümler bir yol cevap paketi gönderirken en kısa yola sahip düğüm ilk olarak cevabı göndermesidir.

Düğümler aynı zamanda “gelişigüzel dinleme” modunda çalışabilir. Böylece düğümler yeni yolları çeşitli yol isteklerinden cevaplandığı kadarıyla öğrenirler. Fakat gelişigüzel dinleme, işlem gören paket sayısı artması durumunda CPU yükünü artırır.

DSR algoritması periyodik güncellemeler göndermez. Ancak her bir veri paketi içine tüm yol bilgisi eklendiği için yönlendirme ek yükü vardır. Bu ek yük hareketlilik ve trafik yoğunluğu durumunda daha da artar.

2.3.2. Ad hoc on-demand distance vector

AODV algoritmasına göre her bir düğüm yönlendirme tablosu tutar. Ancak tabloya dayalı bir algoritma olan DSDV'nin aksine diğer tüm düğümlere olan yolları mutlaka tutmak zorunda değildir. DSR algoritmasına benzer şekilde yayın yoluyla yol belirleme işlemi yapılır [30].

AODV'de kaynaktan yönlendirme yerine ara düğümlerin yönlendirme tablosundaki kayıtlar dinamik olarak oluşturulur. Yönlendirme tablosunda hiçbir kayıt bulunmayan bir hedefe gönderilmek üzere bir paket oluşturulduğunda, Şekil 2.2'de paket yapısı verilen bir yol istek mesajı yayınlanır. Tüm komşu düğümler bu paketi alır ve kendi tablolarını kontrol ederler. Eğer tablolarında isteğe ait bir kayıt yoksa paketi yayınlarlar. Aynı zamanda isteği gönderen düğümü tablolarına kaydederler. Bu kayıt daha sonra bir ters yol belirleme işlemi gerektiğinde kullanılır. Yol istek paketi tüm ağ boyunca yayılıp bir cevap paketi gelinceye kadar kayıtlar tabloda tutulur. İstek mesajları, tablosunda hedefe ait bilgi bulunan düğümlere veya hedefe ulaşmaya kadar düğümden düğüme iletilir. Son düğüm bir cevap paketi gönderir. Ara düğümler yönlendirme tablosundaki kayıtları ters yola dönüştürür [3,32].

Kaynak adres	İstek numarası (id)	Hedef adres	Kaynak sıra no	Hedef sıra no	Atlama sayısı
--------------	---------------------	-------------	----------------	---------------	---------------

Şekil 2.2. Yol istek paketi yapısı

Yol cevap paketi ters yol boyunca yol alır. Yol cevap paketini alan her bir düğüm paketi kimden aldığına dair bir ileri yön işaretçisini set eder. Bu yolla, paket seyahatini tamamladığında kaynaktan hedefe doğru bir ileri yön yolu oluşturulmuş olur. Şekil 2.3'de yol cevap paketi yapısı görülmektedir.

Kaynak adres	Hedef adres	Hedef sıra no	Atlama sayısı	Paket ömrü
--------------	-------------	---------------	---------------	------------

Şekil 2.3. Yol cevap paketi

AODV protokolü bütün linklerin simetrik olduğunu varsayar. Varsayıma göre ağda tek yönlü link yoktur. Bu yüzden cevap mesajları için, istek mesajları için kullanılan yol kullanır. Eski ve yeni yol bilgilerini ayırt etmek için sıra numaralarından faydalanılır [33].

Bu protokol, yerel bağlanabilirliği belirlemek için periyodik HELLO mesajları kullanır. Bu mekanizma aynı zamanda link hatalarını da belirler. Bu protokolün ek yükü DSDV'ninki kadar değildir. Ancak ek yük ortamdaki düğüm sayısının artışıyla artar [3].

Protokol, kaynak ve hedef çiftleri arasındaki çoklu yolları bulur. Bu durum yollardan birinin bağlantısında bir bozukluk varsa yeni bir yol belirleme işleminin ek yükünden kaçınmayı sağlar. Aynı zamanda kullanıcıya yük dengeleme ve benzeri işlemler için yolları seçme ve kontrol etme izni verir. Yol önbelleği, düşük hareketlilik durumlarında çok kullanışlıdır. Ancak topolojinin çok hızlı değiştiği yüksek hareketlilik durumlarında, yol ön belleği bir ek yük oluşturur.

2.3.3. Temporally ordered routing algoritması

TORA, topolojik değişimlere olan reaksiyonu en aza indirmek için tasarlanmıştır. Tasarımdaki bir anahtar kavram; potansiyel olarak çok durumu etkileyen kontrol mesaj yayılımının üretimini, topolojik değişiklik oranından ayırmasıdır.

Protokolün temel işlevselliği şunları içerir: Yolların oluşturulması, sürdürülmesi ve silinmesi. Protokol, ağı bir graf olarak modeller. Graftaki tüm kenarlar yönlendirilmemiş graf yapısındadır. Düğüm i 'den düğüm j 'ye veya düğüm j 'den düğüm i 'ye her bir düğüm yönlü veya yönsüz olabilir. Her bir düğüm, yükseklik ölçütüne sahiptir. Bu ölçüt düğümün komşularına olan bağlantılarını atamak için kullanılır [3].

Yollar reaktif veya proaktif modda oluşturulabilir. Reaktif modda yol oluşturma kaynaktan hedef düğüme doğru linkler kurmayı gerektirir. Bu, hedefte yönlendirilmiş çevrimsiz bir grafla sorgu/istek işlemi kullanılarak yapılır. Bir yol gerektiğinde,

kaynak komşularına QRY (query) paketi yayınlar. İstek paketi, hedefe ait bilgi içeren bir veya daha fazla düğüme ulaşmaya kadar yayılmaya devam eder. İstenen yol bilgisine sahip olan düğüm de UPD (update) paketi yayınlar. UPD paketini alan düğüm, cevabı aldığı düğüme ait “yükseklik” değerini öncekinden bir fazlasına ayarlar. Proaktif modda ise hedef bir optimizasyon (OPT) paketi yayınlayarak yol bulma işlemini başlatır. Bu paket komşularına ve daha uzak komşulara yayılır [3].

Yollar sadece boş olmayan yükseklik değeri ile tutulur. Yükseklik değeri boş olan yollar hesaplamalarda kullanılamazlar. Link kopmalarına karşı tepki, sadece düğümün kendisinden dışarıya doğru olan son linkin kopması durumunda başlatılır. Protokol, link kopmalarına reaksiyon gösterecek düğüm sayısını, en aza indirecek şekilde tasarlanır.

2.4. Melez Yönlendirme Algoritmaları

Melez yönlendirme algoritmaları konum tabanlı yönlendirme algoritmaları ve Multipoint Relaying (MPR) tabanlı algoritmalar olarak iki ana başlık altında toplanmaktadır. Melez yönlendirme algoritmaları, isteğe bağlı ve tabloya dayalı algoritmaların özelliklerini birlikte içeren algoritmalarlardır. Tasarlanan algoritmalar, hem isteğe bağlı hem de tabloya dayalı algoritmaların mekanizmalarına sahiptir.

2.4.1. Konum tabanlı yönlendirme algoritmaları

Konum tabanlı yönlendirme algoritmaları tabloya dayalı ve isteğe bağlı protokollerin özelliklerini içerirler ve genellikle yerleştirilmiş düğümlerle ilgilenirler. Yerleştirme, düğümlerin, coğrafi konumlarını belirlemelerini sağlayan GPS (Global Positioning System) veya Galileo (Global Navigasyon Sistemi) ile gerçekleştirilir [44].

MANET’te düğümlerin hareketliliğinden dolayı oluşan konum değişiklikleri, düğümlerin yol tablolarının değişmesine yol açar. Ancak tabloya dayalı ve isteğe bağlı algoritmaların aksine, konum tabanlı algoritmalarda tablodaki bilgilerin güncellenmesinde, düğümlerin içerdiği GPS’ler kullanılır.

Konum tabanlı algoritmalar, her bir düğümün aktif olmayan (1-hop veya 2-hop komşuları dışındaki düğümler) düğümlerden ayırık olduğunu varsayar. Düğümler yalnızca 1-hop ve 2-hop komşulukları üzerinden paket gönderme işlevini yerine getirirler [44].

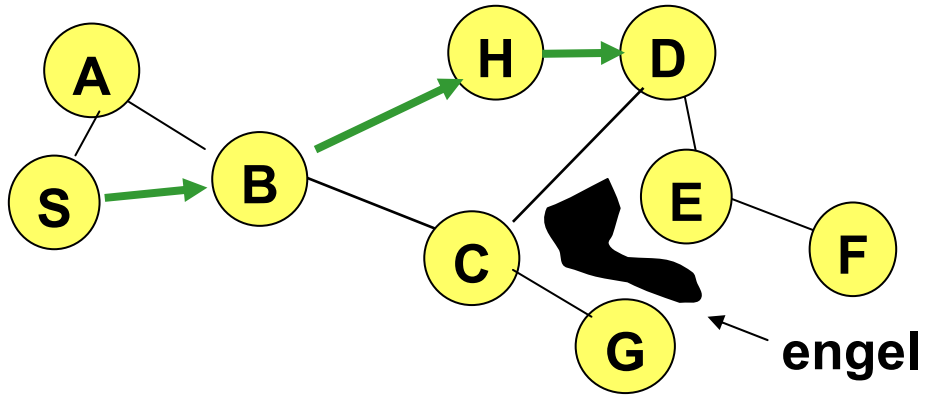
GPS sistemleri düşük maliyetle ve yüksek doğrulukta enlem, boylam ve yükseklik değerlerini sağladığı için tercih edilen bir sistem olmuştur. Bu tür protokollerde düğümlerin yönlendirme kararını vermelerinde etkili olan onların ve diğer düğümlerin coğrafi yerleridir [31]. GPS tabanlı melez yönlendirme algoritmalarından bazıları şunlardır: Geographic Distance Routing (GEDIR), Most Forward within Radius (MFR), Directional Routing Algorithm (DIR).

2.4.1.1. Geographic distance routing

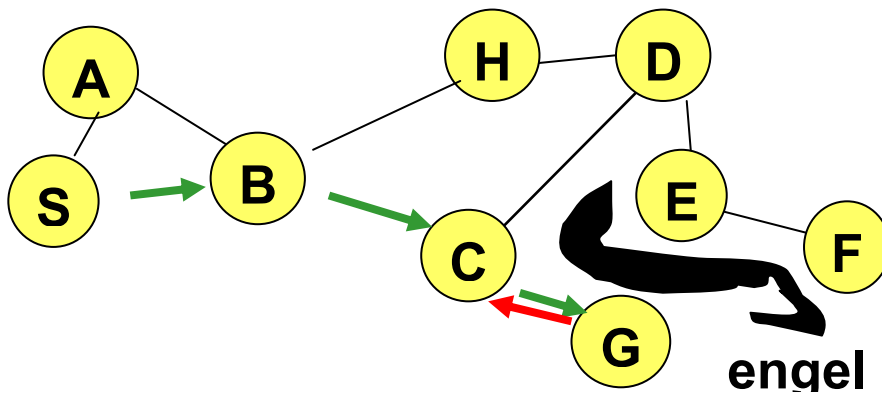
GEDIR algoritması mesaj paketi alıcılarını belirlemek için komşu ve hedef düğümlerin coğrafi bilgilerini kullanır. Komşu düğümden kastedilen hedefe en yakın olan düğümdür. Algoritma birkaç CPU döngüsünde hedefi belirler: İki adet toplama, iki çarpma, bir toplama ve düğüm sayısı kadar karşılaştırma işlemi yapar [37].

GEDIR, tüm düğümlerin enlem ve boylam olarak yer bilgilerini kullanır. Her bir düğüm, doğrudan komşusu olan düğümlerin coğrafi koordinatlarını bilir. Gönderici düğüm, aynı zamanda hedef düğümün yerini bilir. Şekil 2.4'deki örneğe göre S düğümü m mesajını D düğümüne göndermek istediğinde, D 'nin yer bilgisini ve onun 1-hop komşusu olan düğümlerden D 'ye en yakın olan düğümün yer bilgisini kullanır.

Algoritma, aynı kenardan arka arkaya iki kez geçildiğinde durur. Şekil 2.5'deki örnek ağa göre, S kaynak düğümünden, E hedef düğümüne olan yol bozulur. Çünkü yol üzerindeki C düğümünden E düğümüne gidilebilmesi için hedefe en yakın düğüm olan G , E ile haberleşemez.



Şekil 2.4. GEDIR örnek ağı üzerinde yol belirlenmesi



Şekil 2.5. GEDIR örnek ağı üzerinde algoritmanın durması ve hata oluşması

2.4.1.2. Most forward with in radius ve directional routing

İletim mesafesi içindeki en ileri düğüm üzerinden iletme (MFR) algoritması, her bir düğümün, kendisine R mesafe uzaklıktaki düğümlerin konumlarını bildiğini varsayar. Bir düğümün hedefe göndereceği paketi, konumunu bildiği düğümler arasından en ileride olanına gönderir. R mesafesindeki en ileri düğüm, hedef yönündeki düğümler içinden seçilir. Gönderme yönünde en ileride düğüm bulunmuyorsa, en az yakın düğüm iletim için kullanılır [39].

Ağdaki düğüm kendisine paket göndermez. R yarıçaplı daire içinde hiçbir düğüm bulunmaması durumunda, paket gönderilmez. MFR algoritması hedefe ulaşmayı sağlamak için R mesafesini değiştirmez.

Directional Routing yönteminde, bir düğüm bir mesajını kendisinden hedefe kadar olan aralıktaki tüm komşularına gönderir. Belleksiz bir yönlendirme yöntemidir. Bu yüzden yönlendirme işleminde döngüler oluşabilir [38].

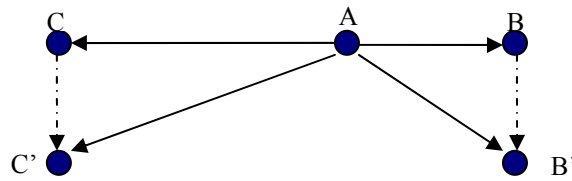
2.4.1.3. Distance routing effect algorithm for mobility

DREAM algoritmasında, her bir düğüm, ağdaki bütün düğümlere ait konum bilgilerini içeren bir veritabanı bulundurur. Konum bilgisi veritabanındaki kayıtlarda düğümlere ait düğüm tanımlayıcı bilgisi, düğümlerin yönü (açısı), düğümlerin mesafeleri ve zaman bilgisi içerilir.

Her bir düğüm, konum bilgileri veritabanının oluşturulabilmesi için güncelleme paketleri gönderirler. Güncelleme paketleri flooding mekanizması ile gönderilir. Bir düğümün konum bilgilerinin doğruluğunun diğer düğümler tarafından bilinebilmesi için iki parametre vardır. Bunlar:

1. Konum bilgisi güncelleme paketleri gönderim sıklığı (zamansal çözünürlük)
2. Güncelleme paketlerinin yok sayılmadan önce dolaşacağı düğüm sayısı (düzlemsel çözünürlük)

Bunlardan birincisi, düğümlerin hareketliliği ile ilgilidir. Düğüm ne kadar hareketliyse, o kadar sık güncelleme paketi gönderimine ihtiyaç vardır. İkincisi ise, komşu düğümlerin birbirinden daha iyi haberdar olacağını ifade eder. Daha uzak düğümler, birbirleri hakkında daha az doğru bilgiye sahip olurlar. Ayrıca düğümler birbirinden ne kadar uzakta olurlarsa, birbirlerinden o kadar geç haberdar olurlar. Buna “distance effect” denir. Şekil 2.6’daki düğümlerden A hareketsiz, B ile C ise aynı yönde ve aynı hızda hareket eden düğümlerdir. A’dan bakıldığında, B’ye olan açı C’dekinden daha fazla artacaktır. Bu durum “distance effect”i ortaya çıkarır. Hedef düğümden çok uzakta olan düğümlerle ilgili düşük düzlemsel çözünürlük sonucunu doğurur. Ara düğümlerin konum bilgileri güncelleme paketleri içinde yer almak şartıyla, düğümler hakkında bilgiye sahip olunur [14,45].



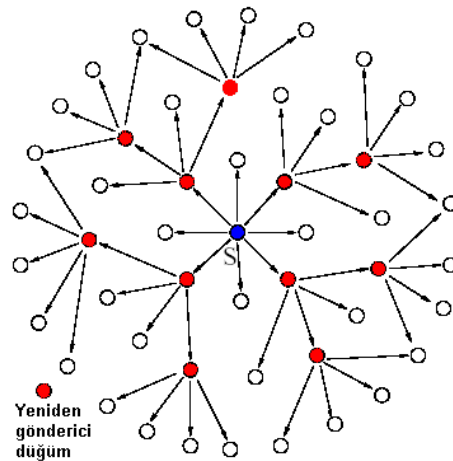
Şekil 2.6. DREAM algoritmasında mesafe etkisi

DREAM algoritması yüksek sayıda düğüm içeren ağlar için her bir düğüm üzerinde büyük miktarda veritabanı bulundurulması gerektiğinden, çok elverişli değildir.

2.4.2. Multi point relaying tabanlı algoritmalar

Melez yönlendirme algoritmalarından bir kısmı Multi Point Relaying (MPR) tekniği üzerine olan algoritmalarlardır. MPR, yönlendirme ek yükünü azaltmak için kaynaktan çıkan bir kontrol paketinin tüm düğümler tarafından alınıp yeniden iletildiği akış tekniğinin oluşturduğu ek yükü azaltmayı hedefler. Kontrol paketlerinin bütün düğümler tarafından değil de önceden belirlenmiş bir küme tarafından yeniden iletilmesini ileri sürer. Böylece düğüm sayısı çok olan ağlarda bir paketin defalarca yeniden iletilmesini önlemeye çalışır.

MPR'nin amacı yayın paketi yeniden gönderim fazlalığını azaltmaktır. Bu teknik “yeniden gönderici” sayısını küçük bir altküme seçerek mümkün olduğunca kısıtlamak anlamına gelir. Bu altküme bir ağ bölgesindeki 1-hop komşuluk tarafından örtülen düğümlerden oluşur. Bu küçük altküme, ilgili düğümün Multi Point Relay'ları (MPRs) denir. MPR planı ağdaki yayın mesajı akışını yeterli miktarda azaltan bir çözüm olabilir. Böylece bütün düğümlere ulaşılmış olur. Şekil 2.7'de yayın mesajları S kaynağından yayınlanıp MPR'lar kullanarak yeniden gönderim yapılmış olur. 3-hopta bulunan komşulara 11 yeniden gönderimle ulaşılmış olur [25].



Şekil 2.7. MPR tekniği yayın akışı

MPR tekniği dağıtık bir yapıda çalışır. Gezgin ve dağınık yapıdaki düğümlerden oluşur. Her düğüm kendi MPR kümesini hesaplar. Bir düğümün MPR seçimi diğerlerinin MPR seçiminden bağımsızdır. Her düğüm komşulukları değiştiğinde bu duruma göre 2-hop komşuluklarını örtecek şekilde MPR kümesini yeniden oluşturur.

2.5. Tasarsız Yönlendirme Algoritmalarının Karşılaştırma Tablosu

Kablosuz tasarsız ağlarda yaygın olarak performans değerlendirmesi yapılan algoritmaların, çeşitli açılardan birbirlerine göre karşılaştırmaları Tablo 2.1'de verilmektedir.

2.6. Sonuç

Bu bölümde, tasarsız ağlarda ön plana çıkan ve performans değerlendirme işlemlerinde sıklıkla kullanılan algoritmaların çalışma esaslarına yer verilmiş ve bir karşılaştırma tablosu sunulmuştur.

Tablo 2.1. Tasarsız ağlar yönlendirme algoritmalarının karşılaştırması

	<i>Periyodik Güncelleme</i>	<i>Yol Keşif Paketli</i>	<i>Konum Bilgisi Kullanımı</i>	<i>Ölçeklenebilirlik</i>	<i>Çoklu Yol</i>	<i>Simetrik Link</i>	<i>Yönlendirme Ek Yükü</i>	<i>Graf yapısı</i>	<i>Uçtan Uca Paket Gecikmesi</i>
<i>OLSR</i>	E	H	H	H	E	H	Y	H	D
<i>DSDV</i>	E	H	H	H	H	E	Y	H	D
<i>WRP</i>	E	H	H	H	H	E	Y	H	D
<i>AODV</i>	H	E	H	H	H	E	Y	H	Y
<i>DSR</i>	H	E	H	H	H	E	Y	H	Y
<i>TORA</i>	E	E	H	H	E	H	Y	E	Y
<i>GEDIR</i>	H	H	E	H	H	E	Y	E	D
<i>V-GEDIR</i>	H	H	E	H	E	E	Y	E	D
<i>DNVR</i>	E	H	H	E	E	E	D	H	-
<i>MPR</i>	E	H	H	H	E	E	-	H	-
<i>APR</i>	E	H	H	H	E	E	-	H	-
<i>DREAM</i>	E	H	E	H	H	E	-	E	D
<i>LAR</i>	H	H	E	H	H	E	-	E	-

E: Evet, H: Hayır, Y: Yüksek, D: Düşük, -: Belirlenmiş değil.

BÖLÜM 3. KONUM TABANLI MELEZ YÖNLENDİRME ALGORİTMASI (KTMYA)

3.1. Giriş

Bölüm 2’de MANET için geliştirilmiş algoritmalar tabloya dayalı, isteğe bağlı ve melez algoritmalar olmak üzere üç kategoride sınıflandırılmıştı. KTMYA, melez ana kategorisi içinde yer alan konum tabanlı algoritmalar sınıfında yer alır.

KTMYA, hem yönlendirme ek yükünü azaltarak bant genişliğinin etkin bir şekilde kullanımını, hem de ağdaki her bir düğüm tarafından yönlendirme için yapılacak işlem sayısını ve tutulacak veri miktarını azaltarak, batarya ömrünü etkin bir biçimde kullanmayı hedeflemektedir. Bu hedefleri gerçekleştirmek için hem isteğe bağlı, hem de tabloya dayalı algoritmaların çalışma esaslarından faydalanılmıştır. Ancak önerilen yöntem bu iki algoritma türünün dışında yeni bir yaklaşım içermektedir.

Önerilen yöntemde, altyapılı kablosuz ağların çalışma esaslarından da faydalanılmıştır. Bilindiği gibi altyapılı kablosuz ağlarda erişim noktası veya baz istasyonu adı verilen merkezi bir düğüm vardır. Bu düğüm hareketsizdir. Merkezi istasyonun kapsama alanında olan düğümler yönlendirme için gerekli olan bilgileri bu istasyondan öğrenirler ve ayrıca veri gönderme-alma işlemini de bu istasyon üzerinden gerçekleştirirler. Ancak tasarsız ağlarda merkezi bir düğüm olmadığı veya diğer bir deyişle bütün düğümler hareketli olduğu için altyapılı ağlardaki çalışma yöntemi tasarsız ağlarda kullanılamaz.

Geliştirilen KTMYA’da altyapılı ağlarda olduğu gibi merkezi bir düğüm diğer bir ifadeyle bir master düğüm atanır ve yönlendirme bilgilerini yönetir. Düğümler hedef bir düğüme veri göndermek istediklerinde, hedef düğümün yerini ve ona gidecek yolu master düğümünden öğrenirler ve bu yol üzerinden verilerini gönderirler. Bu

aşamada algoritma altyapılı ağlardan ayrılır. Çünkü altyapılı ağlarda veri de merkezi istasyon üzerinden gönderilir. Ancak bu algorithmada merkezi istasyon olarak davranan master düğüm sadece hedefe ait yolu bulmada yardımcı olur.

3.2. Düğümlerin Sahip Olması Gereken Özellikler

KTMYA'ya göre, ağdaki düğümler konumlarını bilmeleri gerektiği için mutlaka konum belirleme sistemine (GPS) sahip olmaları gerekir. Ayrıca düğümler arasında algoritmanın yol belirleme amacıyla iletilecek paketlerinin ağda gönderilebilmesi için düğümler arası bağlantıların iki yönlü (simetrik) olması gerekir.

Düğümlerin KTMYA'yı kullanabilmeleri için şu özelliklere sahip olması gerekir.

- En az 6 MB bellek içermeli
- En az 200 MHz işlemciye sahip olmalı
- Üzerlerinde GPS (Global Positioning System) bulundurmalı
- İki yönlü linklerin çalışıyor olmalı

6 MB bellek kriterini belirlemek için, en az 1000 düğümlü bir ağın yönlendirme bilgilerini anlık olarak depolayabilme kapasitesi göz önünde bulundurulmuştur. Ayrıca, yönlendirme işlemlerini yapabilmek için 200 MHz işlemci kriteri ileri sürülmüştür.

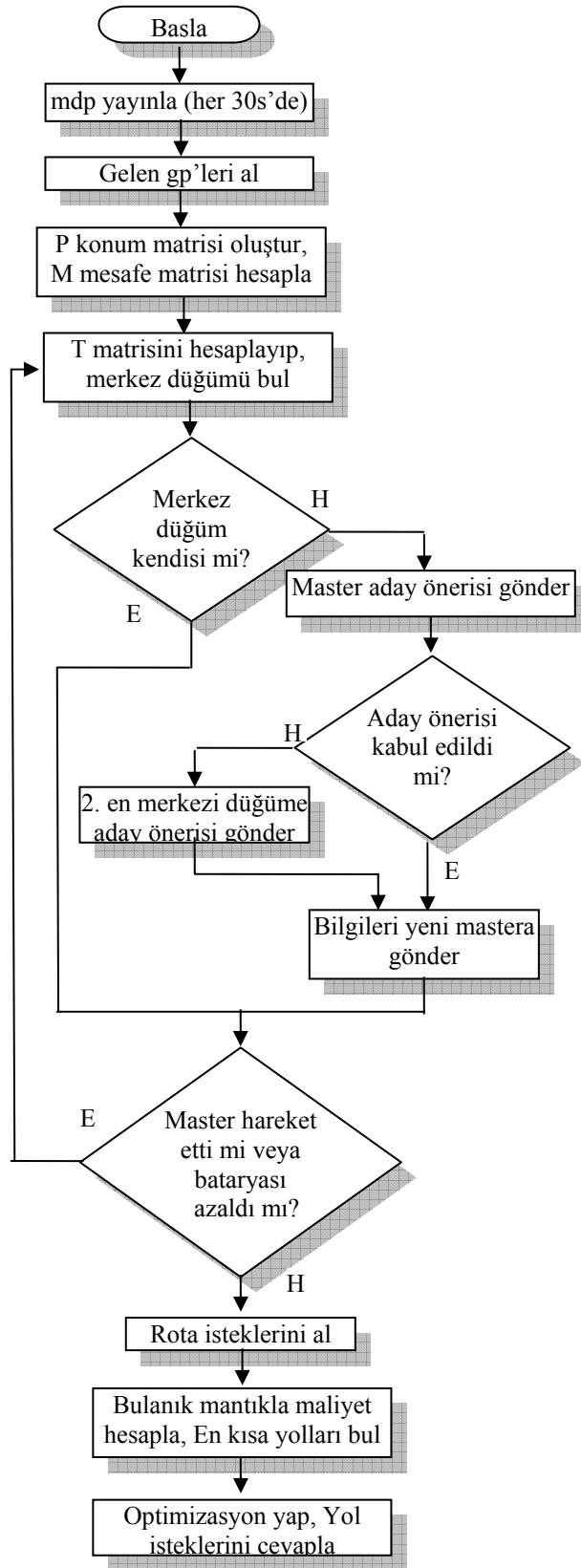
3.3. KTMYA'nın Adımları

KTMYA aşağıdaki adımlara sahiptir:

- a. Ağda ilk aktif olan düğüm, master düğüm olarak atanır.
- b. Master düğüm belirli aralıklarla master düğüm duyuru paketi yayınlarken ağa master düğüm olduğunu duyurur.
- c. Ağdaki diğer düğümler master düğüme kendi konumlarının bilgisini içeren güncelleme paketleri gönderirler.
- d. Master düğüm güncelleme bilgilerini kullanarak P konum matrisini oluşturur.
- e. Master düğüm konum bilgilerini kullanarak her bir düğümün birbirlerine olan uzaklığını hesaplayıp M mesafe matrisini oluşturur.

- f. M matrisinin satır elemanları toplamı yapılarak en küçük değerli olan satıra karşılık gelen düğüm, en merkezde olarak belirlenir ve aday master düğüm tespit edilir.
- g. Mevcut master düğüm, aday master düğüme öneri paketi göndererek yeni master düğüm olma önerisinde bulunur.
- h. Yeni master düğüm ağa master düğüm duyuru paketleri gönderir.
- i. Diğer düğümler güncelleme paketlerini eğer gerekliyse yeni master düğüme gönderirler.
- j. Master düğüm M ve P matrislerini kullanarak bulanık mantık yöntemiyle her düğümün birbirlerine olan maliyet kriterini belirler ve G matrisini oluşturur.
- k. Düğümler arası maliyeti en az yolların belirlenmesi için G matrisi kullanılarak optimizasyon yapılır.
- l. Düğümler bir başka düğüme veri göndereceklerinde master düğüme yol istek paketi göndererek en kısa yolu sorarlar. Master düğüm optimizasyon sonucuna göre en kısa yolu istekte bulunan düğüme bildirir.
- m. Master düğüm kendisi merkezden uzaklaşırsa veya batarya ömrü eşik olarak kullanılan bir değer altına düşerse, master düğüm rolünü başka bir uygun düğüme devreder.
- n. Ağdaki diğer düğümler sadece master düğümün kimliğini ve yerini belleklerinde tutarlar.

Kısa çalışma adımları burada verilen algoritmanın akış diyagramı Şekil 3.1'de görülmektedir.



Şekil 3.1. KTMYA akış diyagramı

3.4. KTMYA Algoritması

Algoritmanın Bölüm 3.3’de verilen çalışma adımlarının ayrıntıları şöyledir:

- a. Ağda aktif olan ilk düğüm master düğüm olarak atanır. Eğer aynı anda iki düğüm açılır ve ağda iki master düğüm oluşursa, birbirlerinden aldıkları ilk paketler içindeki MAC adreslerini karşılaştırırlar ve MAC adresi küçük değerli olan master düğümlükten vazgeçer. Master düğüm belirleme işleminin ayrıntıları Bölüm 3.4.1’de verilmiştir.
- b. Master düğüm belirli aralıklarla broadcast paketleri yayınlar ve kendisinin master düğüm olduğunu ağdaki diğer düğümlere bildirir. Bu paketlere, “*master düğüm duyuru paketi (mdp)*” adı verilir. Master düğüm duyuru paketinin çerçeve yapısı Şekil 3.2’de verilmiştir. Bu paket içindeki alıcı adresi alanı yayın (broadcast) adresidir. Kaynak adres alanı ise master düğümün adresini ifade eder. Buradaki sıra numarası alanı, master düğümün duyuru paketi broadcast paketi olarak yayımlandığında paketin geçtiği bütün düğümlerin kendi adreslerinin eklenmesi için kullanılır. Toplam 10 düğümden daha fazla düğüm geçilmesi gereken durumlarda, ağın Bölüm 3.6’da verilen yöntemi kullanarak kümelere ayrılması gerekir. Böylece master düğümü öğrenen düğüm, master düğüme giden yolu da öğrenmiş olur.

Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Sıra Numarası 48 Byte
---------------------------	----------------------------	-----------------------------

Şekil 3.2. Master düğüm duyuru paketlerinin (mdp) çerçeve yapısı

- c. Master düğüm dışındaki düğümler, master düğüme *güncelleme paketleri (gp)* gönderirler. Bu paketlerin içinde buldukları yerin *coğrafi konumu* (x,y,z koordinatları olarak), kalan *batarya ömürlerinin yüzde olarak değeri* ve *düğüm yoğunluğu* bilgileri bulunur. Paket içinde hedef adres, kaynak adres ve id alanları da bulunur. Id alanı master düğümün oluşturacağı konum bilgileri matrisinin ilgili satırını güncellemek amacıyla ve istatistik işlemleri için kullanılır. Güncelleme paketlerinin çerçeve yapısı Şekil 3.3’de verilmiştir.

Alıcı Adresi ve Yolu 48 Byte	Kaynak Adresi 6 Byte	Konum (x,y,z) 12 Byte	Batarya Miktarı(%) 1 Byte	Yoğunluk 1 Byte	Id 2 Byte
---------------------------------	-------------------------	--------------------------	------------------------------	--------------------	--------------

Şekil 3.3. Düğüm bilgi güncelleme paketlerinin (gp) çerçeve yapısı

Alıcı adresi o anki master düğümün adresidir. Kaynak adresi ise güncelleme paketi gönderen düğümün adresidir. Paket içindeki id alanını gönderici düğüm her defasında bir artırır. Bu formattaki güncelleme bilgisi master düğüm üzerinde tutulan P matrisine bir satır elemanı olarak işlenir. Eğer aynı düğümden daha önce güncelleme bilgisi alınmış ise id değerleri karşılaştırılır. Daha büyük id değerine sahip paket en güncel bilgi olarak kaydedilir ve önceki kayıt değiştirilir. Konum alanı içinde, düğümün GPS aracılığıyla belirlediği konum bilgisi yer alır. Her bir konum parametresi double integer formatında olacağı için toplam 12 Byte alan boyutu öngörülmüştür. Batarya miktarı ve yoğunluk alanları içinde düğümün kalan batarya miktarı ve tampon doluluğu yüzde cinsinden gönderilir. Bu veri Byte türünden ifade edilebileceği için 1Byte alan boyutu ayrılmıştır.

d. Master düğüm diğer düğümlerden gelen paketleri kullanarak *konum bilgileri matrisi* oluşturur. P matrisi adı verilen bu matrisin sütunlarında her bir düğümün x_i konum bilgisi, y_i konum bilgisi, z_i konum bilgisi, batarya ömrü (b_i), yoğunluk (d_i) ve düğüm güncelleme id'si (id_i) yer alır. Matrisin satır sayısı ağdaki düğüm sayısına eşittir. k düğümlü bir ağ için bu matris denklem 3.1'de verilmiştir.

$$P = \begin{bmatrix} x_1 & y_1 & z_1 & b_1 & d_1 & id_1 \\ x_2 & y_2 & z_2 & b_2 & d_2 & id_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_k & y_k & z_k & b_k & d_k & id_k \end{bmatrix} \quad (3.1)$$

e. Master düğüm denklem 3.1’de verilen P matrisinin birinci, ikinci ve üçüncü sütunlarını kullanarak her bir düğümün birbirlerine olan uzaklığını hesaplar. Bu işlemi, denklem 3.2’de verilen eşitliği kullanarak yapar. Bunun sonucunda boyutu ağdaki düğüm sayısına eşit olan bir kare matris oluşur. k düğümlü bir ağ için M mesafe matrisi denklem 3.3’de verildiği gibi bulunur.

$$l_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (3.2)$$

$$M = \begin{bmatrix} l_{1,1} & l_{1,2} & \cdot & \cdot & l_{1,k} \\ l_{2,1} & l_{2,2} & \cdot & \cdot & l_{2,k} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{k,1} & l_{k,2} & \cdot & \cdot & l_{k,k} \end{bmatrix} \quad (3.3)$$

Denklem 3.3’deki M matrisinde her düğümün kendisine olan uzaklığı 0 olacağı için esas köşegen üzerindeki elemanlar 0 olacaktır. Ayrıca 1 numaralı düğümün 2 numaralı düğüme olan uzaklığı ile 2 numaralı düğümün 1 numaralı düğüme olan uzaklığı eşit olacağı için matris esas köşegene göre simetrik olacaktır. Bu yüzden master düğüm M matrisinin sadece esas köşegen üstündeki elemanlarını hesaplar. Esas köşegen altındaki elemanları ise simetriklerini kullanarak doldurur. Bu sayede master düğüm M matrisini oluştururken daha az işlem yapar. Bu da düğümün batarya kullanımı için önemli bir faktördür.

f. Ağın merkezindeki düğüm hesaplanır. Denklem 3.3’de verilen M mesafe matrisinin satır elemanlarının toplamı alınarak denklem 3.4’de verilen T sütun matrisine aktarılır. T matrisinin en küçük elemanına sahip olan indis numarası, ağın merkezindeki düğümün numarasına eşittir [46].

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \cdot \\ \cdot \\ t_k \end{bmatrix} \quad (3.4)$$

T matrisinin elemanları aşağıdaki ifade ile hesaplanır;

$$t_m = \sum_{n=1}^k l_{m,n} \quad (3.5)$$

T matrisinin her bir elemanı, o elemana ait indis numaralı düğümün diğer tüm düğümlere olan toplam mesafesini verir. Ayrıca, ağın dağınıklığı hakkında da bilgi verir. T matrisinin elemanlarının cebirsel toplamı büyüdükçe, o ağın dağınıklığı artar. T matrisinin elemanlarının cebirsel toplamı küçüldükçe, ağdaki düğümler birbirlerine yakındır.

g. Yeni master düğüm adayı, ağın merkezindeki düğümdür. Master düğüm aday düğüme yeni master düğüm olup olamayacağını sorar. Eğer olumlu cevap almışsa kendinde bulunan bütün yönlendirme bilgilerini yeni master düğüme veri paketi ile gönderir. Aynı zamanda diğer düğümlere de yeni master düğümü ve onun konum bilgisini bildirir. Eğer olumsuz cevap almışsa T matrisine göre ikinci en merkezi düğüm yeni master adayıdır. Aynı işlemler bu düğüm için gerçekleştirilir. Aday master düğüm batarya ömrü veya yoğunluk gibi sebeplerle master düğümlüğü reddedebilir. Master düğüm önerisini reddetme için eşik değer kararı benzetim esnasında belirlenecektir. *Master düğüm öneri paketi (mop)* ve *master düğüm öneri-cevap paketinin (mcp)* çerçeve yapıları sırasıyla Şekil 3.4 ve Şekil 3.5'te verilmiştir.

Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Öneri Verisi 2 Byte
---------------------------	----------------------------	------------------------

Şekil 3.4. Master düğüm aday önerisi paketi

Paket içindeki alıcı adresi, aday master düğümün adresi, kaynak adres ise mevcut master düğümün adresidir. Bu paketteki öneri verisi içindeki bir byte 0 değerinde diğer byte ise 1 değerindedir. “Master düğüm önerisi” yapmak için aktif master düğüm tarafından, aday master düğümüne gönderilen paketin Öneri verisini onluk olarak “01” değerine ayarlar.

Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Öneri Verisi 2 Byte
---------------------------	----------------------------	------------------------

Şekil 3.5. Master düğüm adaylık kabul veya red paketi

Paket içindeki alıcı adresi mevcut master düğümün adresi, kaynak adres ise aday master düğümün adresidir. Aday master düğüm öneri paketini aldığı anda eğer master düğüm olmaya istekliyse öneri verisinin değerini 11 yaparak, değilse 10 yaparak aktif master düğümüne gönderir.

h. Yeni master düğüm ağa kendisinin master düğüm olduğuna dair yayın (broadcast) paketleri gönderir. Bu paketin çerçeve yapısı Şekil 3.2’de verilmiştir. Diğer düğümlerden gelecek güncelleme paketlerini de (güncelleme paketi gönderimi gerekli ise) önceki master düğümün yaptığı gibi P matrisinde toplar. Yeni master düğüm a-h arasındaki adımları tekrar eder.

i. Diğer düğümler master düğümüne

1. Konum değiştirdiklerinde
2. Batarya ömürleri eşik bir değer altına düştüğünde
3. Yoğunlukları arttığında

olay tabanlı olarak güncelleme paketlerini gönderirler. Paketler içinde gönderilen id değeri sayesinde master düğüm P matrisinde o düğümle ilgili eski bilgiyi silip yeni bilgiyi yazar. Çünkü diğer düğümler her gönderdikleri güncelleme paketi içinde bir öncekinin bir büyüğü olan id değeri gönderirler.

j. Bu algoritmaya göre normal düğümler, hedef bir düğümüne, Şekil 3.6’da verilen çerçeve yapısında bir veri göndermek istediklerinde, master düğümden, hedef

düğümüne olan yol bilgisini sorarlar. Master düğüm kendisine herhangi bir hedefle ilgili bir istek geldiğinde M matrisi ve P matrisini kullanarak, bulanık mantık yöntemiyle düğümler arası kenarlara birer maliyet değeri atar. Böylece düğümler ve kenarlardan oluşan bir graf oluşturur. Grafın maliyet değerlerini tutmak için G matrisi oluşturulur. G matrisinin oluşturulması Bölüm 3.5’te ayrıntılı olarak ele alınmaktadır.

Çerçeve Kontrol 2 Byte	Çerçeve Boyut 2 Byte	Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Adres3 6 Byte	Adres4 6 Byte	Sıra Kontrol 2 Byte	Yol Bilgisi 32 Byte	Veri 0-2280 Byte	Kontrol Alanı 4 Byte
---------------------------	-------------------------	------------------------	-------------------------	------------------	------------------	------------------------	------------------------	---------------------	-------------------------

Şekil 3.6. Veri paketi çerçeve yapısı

Alıcı adresi, veriyi alacak düğümün adresi, kaynak adresi de veriyi gönderen düğümün adresidir. Adres3 ve Adres4 alanlarında yol üzerindeki ilk iki düğümün adresleri bulunur. Yol bilgisi alanında da kaynaktan hedefe kadar olan ilk iki düğümden sonraki düğümlerin adresleri bulunur. Yol bilgisi (32 Byte) ve Veri (0-2280 Byte) alanları, frame gövdesidir.

- k. Master düğüm, oluşan graf üzerinden kaynak ve hedef arasındaki en düşük maliyetli yolu bulabilmek için bir optimizasyon yapar. En kısa yol yani en maliyetsiz yol, Dijkstra veya Bellman-Ford algoritması kullanılarak belirlenir.
- l. Master düğüm, j ve k adımlarından elde ettiği sonucu istekte bulunan düğüme bildirir ve ilgili düğümde verisini bildirilen yoldan gönderir. Herhangi bir düğüm master düğümden yönlendirme isteğinde bulunacağı zaman, master düğüme bir “*yol istek paketi (yip)*” gönderir. Master düğüm de yol isteğinde bulunan düğüme “*yol cevap paketi (ycp)*” gönderir. Yol istek ve yol cevap paketlerinin çerçeve yapıları sırasıyla Şekil 3.7 ve Şekil 3.8’de verilmiştir.

Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Hedef Adresi 6 Byte
------------------------	-------------------------	------------------------

Şekil 3.7. Yol İstek Paketi çerçeve yapısı

Yol istek paketi içindeki “Hedef adresi” alanı, istekte bulunan düğüm tarafından veri gönderilmek istenen düğümün adresidir. Alıcı adresi, master düğümün adresi ve kaynak adresi istekte bulunan düğümün adresidir.

Alıcı Adresi 6 Byte	Kaynak Adresi 6 Byte	Hedef Adresi 6 Byte	Yol Bilgisi 48 Byte
------------------------	-------------------------	------------------------	------------------------

Şekil 3.8. Yol Cevap Paketi çerçeve yapısı

Master düğüm yol istek paketini aldığı anda oluşturmuş olduğu graf yapısı üzerinde optimizasyon yaparak istekte bulunan düğümünden hedef düğüme olan en optimum yolu belirleyip yol bilgisi alanına bu yolu yerleştirerek istek sahibi düğüme cevap verir. Alıcı adresi, istekte bulunan düğümün adresini, kaynak adresi master düğümün adresini ve hedef adresi de, istekte bulunan düğüm tarafından veri gönderilmek istenen düğümün adresidir.

m. Master düğüm kendisi hareket ederek merkezden uzaklaşırsa, batarya ömrü eşik bir değerin altına düşerse veya işlem yoğunluğu artarsa yeni bir master düğüme görevi devreder. Düğümler master düğüm olup olmamaya, batarya ömürlerine ve yoğunluklarına göre karar verirler. Herhangi bir sebeple master düğümün kapanması durumunda, ağın master düğümsüz kalmaması için, ağda “ikincil master” düğüm atanır. Bu atama işlemi, geçerli master düğüm tarafından yapılır. Master düğüm, ikincil master olarak, kendisine en yakın düğümü seçer. Kendi üzerinde tutmuş olduğu yönlendirme bilgilerini belirli periyotlarla ikincil master’a gönderir. İkincil master’a bilgi gönderim sıklığı, master düğüm duyuru paketleri gönderim aralığının dört katıdır. Bu süre belirlenirken, hem ağın master düğümü kaybetmesi durumunda yönlendirme kaybına uğramasını azaltmak, hem de birincil ve ikincil master arasındaki paket gönderimlerinden dolayı ek trafik oluşturmayı en aza indirmek hedeflenmiştir.

n. Diğer düğümler master düğüm gibi bütün düğümlere ait bilgileri tutmazlar ve yönlendirme ile ilgili herhangi bir işlem yapmazlar. Ancak, master düğümünden gelen *mdp* içindeki master’ın adresini ve ona gidecek yolu belleklerinde bulundurlar.

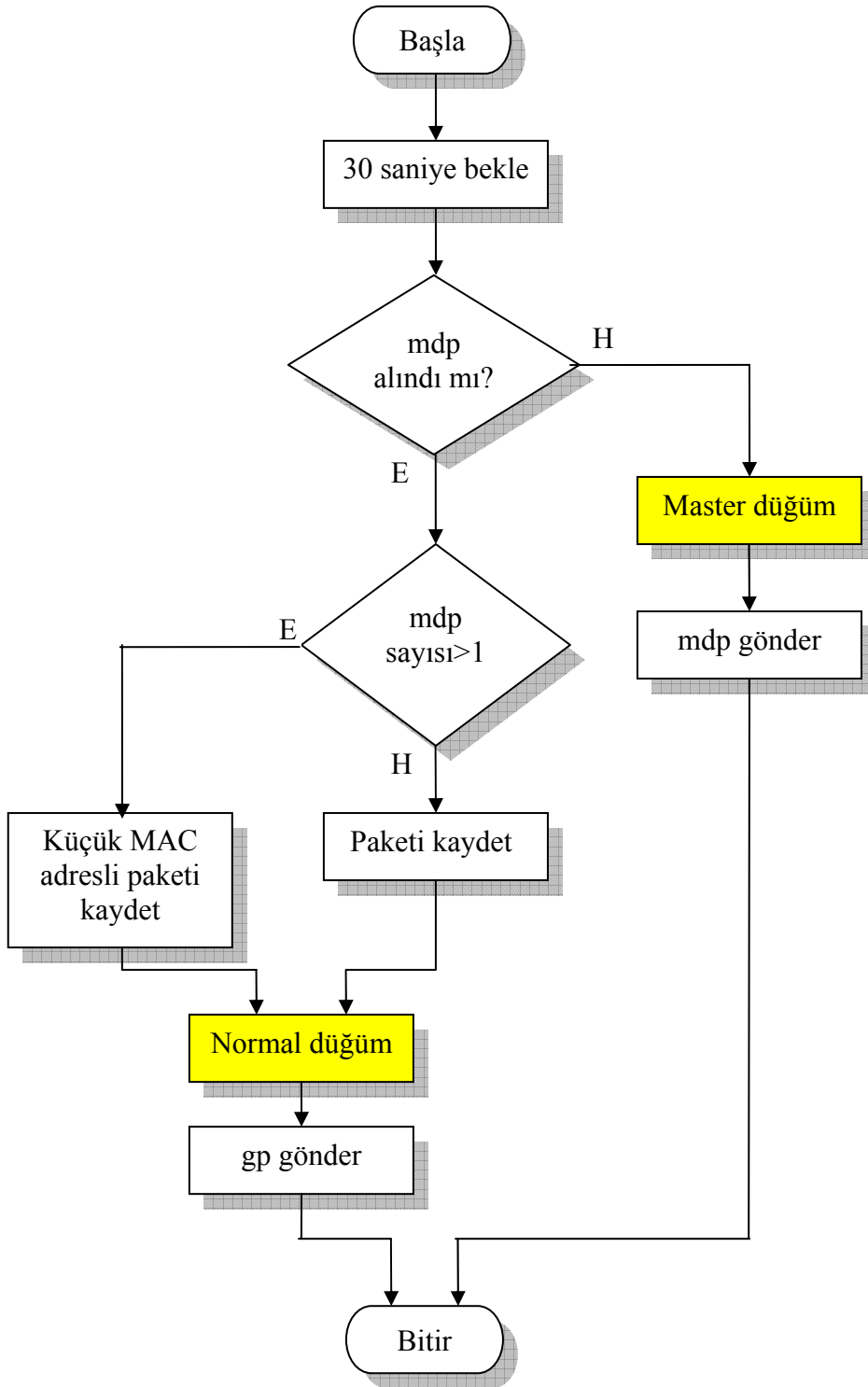
3.4.1. Master düğüm rolünü belirleme

KTMYA'ya göre, ağdaki bir düğüm için üç rol vardır. Bunlar; master, ikincil master ve normal düğüm. İkincil master rolünün belirlenme işlemi, master düğüm tarafından yapılır. Bu yüzden bir düğümün kendisinin master düğüm veya normal düğüm olup olmadığını bilme zorunluluğu vardır.

Master düğümlüğü belirleme işlemi aşağıdaki adımlarla gerçekleştirilir:

1. Ağdaki bir düğüm ayağa kalktıktan sonra 30 saniye süre bekler.
2. Bu sürede düğüm, master düğüm duyuru paketi (*mdp*) aldı mı?
3. 2. adımın cevabı Evet ise;
 - a. Bir *mdp* mi aldı yoksa birden fazla *mdp* mi aldı?
 - i. Bir *mdp* aldıysa, paketi aldığı düğümün adresini ve konumunu belleğine master düğüm olarak kaydeder. Kendisinin normal düğüm olduğuna karar verir.
 - ii. Birden fazla *mdp* aldıysa, aldığı paketler içindeki adresleri karşılaştırır. Küçük adresli olanı ve konumunu belleğine master düğüm olarak kaydeder. Kendisinin normal düğüm olduğuna karar verir.
 - b. Belleğine kaydettiği master'a konumunu içeren güncelleme paketi (*gp*) gönderir.
4. 2. adımın cevabı Hayır ise;
 - a. Ağda mevcut bir master düğüm yoktur. Kendisinin master düğüm olduğuna karar verir.
 - b. Ağa 30 saniyelik periyotlarla *mdp* yayınlar.
5. Bitiş.

Master düğümlüğü belirleme işlemi Şekil 3.9'da verilen akış diyagramında ifade edilmiştir.



Şekil 3.9. Master düğüm rolünü belirleme işlemi akış diyagramı

3.4.2. Yönlendirme bilgisi isteği ve cevabı

KTMYA'ya göre gönderecek verisi olan bir düğüm, algoritmanın l adımı gereğince,

master düğümden hedefin yol bilgisini ister. Master düğüm, kendisinde bulunan bilgileri kullanarak, Dijkstra algoritması sonucunda elde ettiği en maliyetsiz yolu, istekte bulunan düğüme gönderir.

Master düğüm, en maliyetsiz yol belirleme işlemi için, geçerli ağ yapısını kenarlardan ve düğümlerden oluşan bir graf yapısında ifade eder. Kenarlara, ağırlık değeri olarak, bulanık mantık sonucu bulunan maliyet değerleri verilir.

Yol istek ve cevap işlemleri aşağıdaki adımlarda ifade edilir:

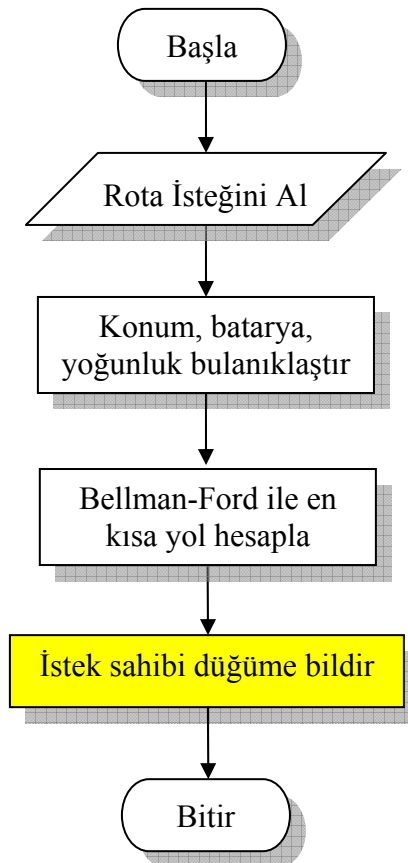
- Düğüm yol isteğinde bulunur.
- Master düğüm, konum bilgileri matrisi içindeki batarya ömrü ve yoğunluk, mesafe matrisi içindeki düğümler arası uzaklıklar bilgilerini bulanıklaştırarak düğümler arası maliyet değerlerini elde eder.
- Master düğüm, Dijkstra algoritmasını kullanarak istekte bulunan düğüm ile onun veri göndermek istediği düğüm arasındaki en maliyetsiz yolu elde eder.
- Master düğüm bulduğu yolu istekte bulunan düğüme gönderir.
- İlgili düğüm, gelen yol bilgisini, göndereceği veri paketinin başlık kısmına yazarak ağa gönderir.

Yönlendirme bilgisi isteği ve cevabı işlemlerinin yukarıda verilen maddelerinin akış diyagramı Şekil 3.10'da verilmiştir.

3.4.3. Master düğüm duyuru paketlerinin ağda dağıtımı

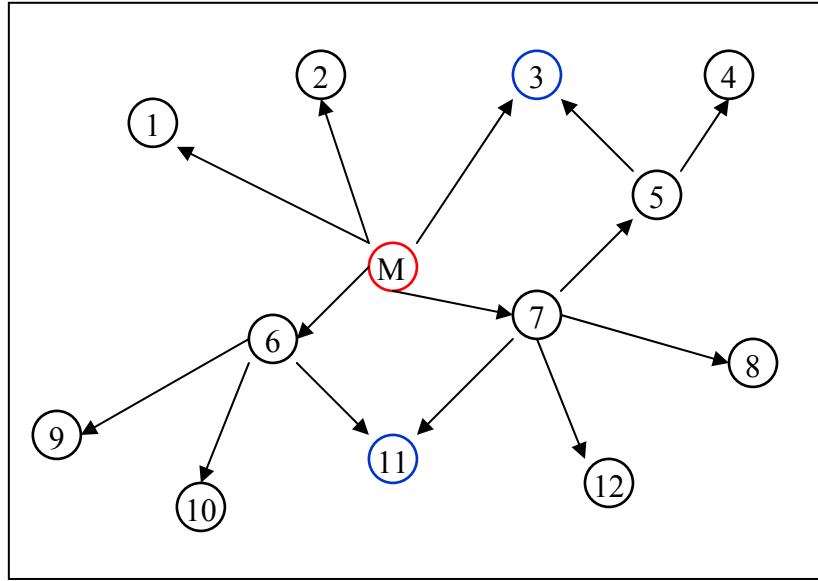
Master düğüm duyuru paketleri, ağda en öncelikli paketlerdir. Master düğüm duyuru paketinin önceliği, Şekil 3.2'de verilen çerçeve yapısında yer alan alıcı adresi alanı ile belirlenir. *mdp*'de alıcı adresi yayın (FF-FF-FF-FF-FF-FF) adresidir. Ağda gönderilen hiçbir kontrol paketi yayın paketi değildir. Herhangi bir düğüm bir başka düğüme iletilmek üzere bir *mdp* aldığı anda, ilk olarak bu paketi iletir. *mdp* master düğümden çıktıktan sonra, master düğümün yayın mesafesinde olan düğümlere gönderilir. *mdp* paketini alan her bir düğüm, paket içindeki yol bilgisi alanına kendi adresini ekledikten sonra, yayın mesafesinde bulunan düğümlere gönderir. Eğer bir düğüm aynı *mdp*'yi başka düğümlerden birden fazla alırsa, bunlardan sadece bir

tanmasını (ilkini) iletir. Aynı paketleri yeniden iletmez. Normal düğüm, kendi veritabanına master düğümüne ait bilgiyi kaydettiği için, *mdp* paketlerinin birden fazla alınmasını kendi veritabanlarındaki ile karşılaştırıp, yeniden iletip iletmeyeceğine karar verir.



Şekil 3.10. Yönlendirme bilgisi isteği ve cevabı işlemlerinin akış diyagramı.

Düğüm *mdp*'yi kendisine ileten düğümüne geri göndermez. Bu da *mdp* paketlerinin ağda tek yönlü olarak iletilmesini sağlar. Bu durum ağın *mdp* paketleri ile yoğun bir biçimde meşgul edilmesini önler. Şekil 3.11'de *M* ile gösterilen master düğümünden çıkan *mdp* paketlerinin dağıtımını gösterilmiştir.

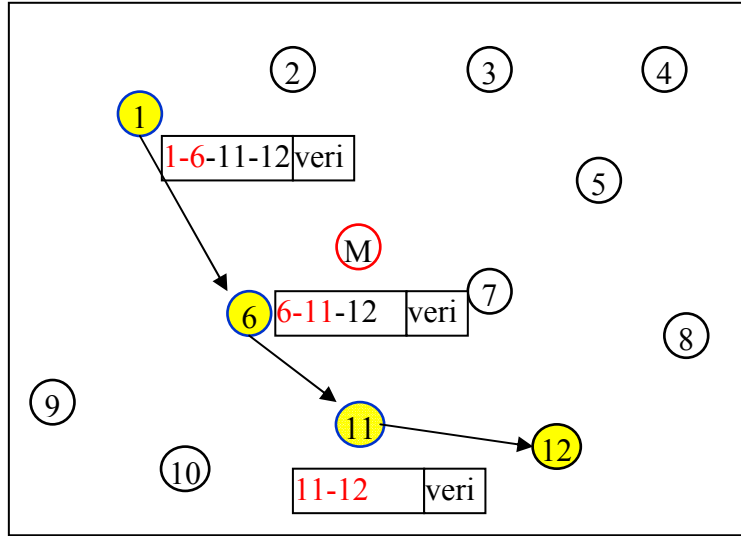


Şekil 3.11. Master düğüm duyuru paketlerinin ağda dağıtımı

3.4.4. Veri paketlerinin ağda dağıtımı

Veri paketleri ağda, kaynaktan yönlendirme yöntemine göre iletilir. Veri paketi gönderecek düğüm, paketin başlık kısmına kendisinden hedefe kadar olan tüm yol bilgisini yazar. Bir düğüm, iletilmek üzere bir veri paketi aldığı anda, paketin başlık kısmında bulunan hedef alanındaki, kendisine ait olan adres bilgisini çıkartıp, sonraki adresin sahibi olan düğüme paketi iletir.

Veri paketinin gönderileceği bir sonraki düğüm, gönderici düğümün yayın mesafesinde olmasından dolayı, KTMYA algoritması tarafından garanti edilmiştir. Bu işlem Bölüm 3.2’de verilen algoritmanın j adımına göre gerçekleştirilmiştir. Veri paketini iletmek üzere alan düğüm, paket başlığında yazan yol bilgisine göre, bir sonraki düğüme gönderir. Bu işlemin tekrar edilmesi sonucu veri paketi hedef düğüme ulaşmış olur. Şekil 3.12’de örnek bir veri paketinin izlediği yol gösterilmiştir.



Şekil 3.12. Veri paketlerinin ağda dağıtımı

3.4.5. Konum bilgisi güncelleme paketlerinin ağda dağıtımı

Ağdaki normal düğümler ilk aktif olduklarında, konum değiştirdiklerinde, bataryaları eşik bir seviyenin altına düştüğünde ve tamponlarındaki yoğunluk eşik bir seviyenin üzerine çıktığında, master düğüme güncelleme paketi (gp) gönderirler.

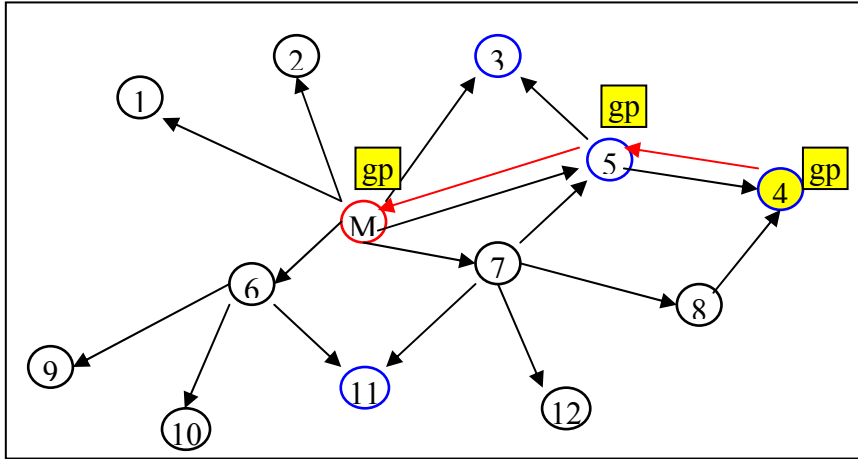
Düğümler, güncelleme paketini master düğüm duyuru paketinin (mdp) geldiği yoldan geri gönderirler. Şekil 3.1’de verilen mdp’nin “sıra numarası” alanına, geçilen her bir düğümün adresi eklenir. Sıra numarası alanındaki adresler tersine çevrildiğinde düğümden master düğüme doğru bir yol oluşmuş olur. “gp” bu yol üzerinden master düğüme kadar ulaştırılır.

Düğümler aynı mdp’yi farklı düğümler üzerinden birden fazla sayıda alırlarsa, mdp’ler içindeki sıra numarası alanından, geçilmiş düğüm sayısına bakarlar. En az sayıda düğümden oluşan yol, gp paketlerinin gönderimi için kullanılır (Şekil 3.13).

3.5. Bulanık Mantık Yöntemiyle Maliyet Değerlerinin Belirlenmesi

Bu kısımda, KTMYA algoritmasına göre, ağdaki düğümler arası maliyet değerlerinin belirlenmesi ve ağın graf yapısına göre de kenarlara ağırlık değerinin atanması

işlemleri ele alınmaktadır. Konunun anlaşılması açısından, graf teorisi ile ilgili olarak Ek A'da ve bulanık mantık teorileri hakkında Ek B'de detaylı bilgiler verilmektedir.



Şekil 3.13. Güncelleme paketlerinin ağda dağıtımı

3.5.1. Bulanık mantığın KTMYA'ya uygulanması

Algoritmada bulanık mantık yönteminin kullanılmasının sebebi, düğümlerin yönlendirme amacıyla daha etkin kullanımını sağlamaktır. Sadece konum bilgileri içindeki düğümler arası mesafe esas alınarak yapılacak bir yönlendirmede, ağdaki bazı düğümler çok fazla kullanılacağından, bataryaları kısa sürede bitebilir. Ayrıca, mesafeleri çok düşük olan iki düğümden birinin tampon yoğunluğu yüksekse, yönlendirilmiş paketin uçtan uca ulaşım zamanı artacaktır. Bütün bunlara dayalı olarak, düğümlerin enerji kullanımlarını optimize etmek ve uçtan uca gecikmeyi azaltmak amacıyla, algoritmada bulanık mantık yöntemi kullanılmaktadır.

İlk olarak, 3. Bölümde verilen algoritma içindeki P ve M matrisleri kullanılarak kablosuz ağın o anki durumunun topolojisini ifade eden graf oluşturulur. Grafın sonucu, M matrisi ile aynı boyutta olan G matrisinde tutulur.

Düğümlerin aşağıdaki varsayımları sağladıkları düşünülerek graf oluşturulur:

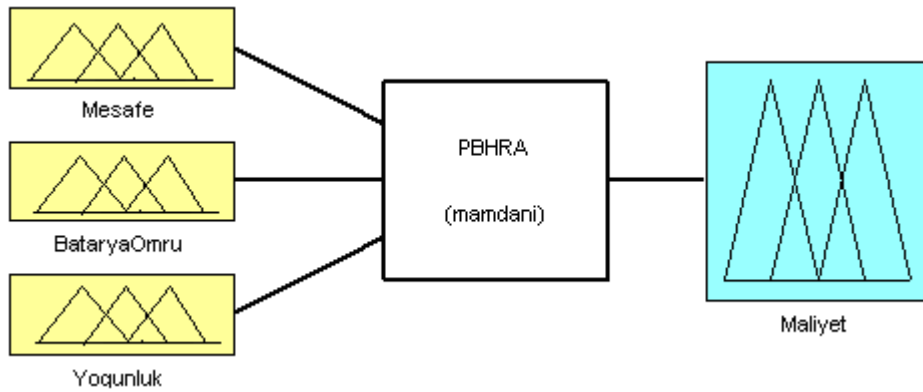
- Her bir düğüm kendisinden l_T (düğümlerin yayın mesafesi) uzaklıktaki düğümlere doğrudan paket gönderebilir. Daha uzak düğümlere ancak diğer düğümler üzerinden paket gönderebilir. Yani her bir düğüm kendi yayın

mesafesi (kapsama alanı) içinde kalan düğümlerle doğrudan haberleşebilir, yayın mesafesi dışında kalan düğümlerle de diğer düğümler üzerinden haberleşebilir.

- b. Düğümler arasındaki linkler iki yönlüdür. Yani komşu iki düğüm karşılıklı olarak paket gönderebilirler.

Bulanık mantık yöntemi aşağıdaki adımlara dayalı olarak uygulanmıştır:

- a. M mesafe matrisinin elemanları l_T değeri ile karşılaştırılır. l_T 'den büyük elemanlar “sonsuz” olarak işaretlenip, küçük elemanlar aynen korunarak M matrisi oluşturulur.
- b. G graf matrisi bulanık mantık kullanılarak oluşturulur.
- b.1. Üç adet bulanık giriş değişkeni vardır. Bunlar mesafe, batarya ömrü ve yoğunluktur. Bulanık çıkış değişkeni ise maliyet olarak adlandırılır.
- b.2. Blok görünümü Şekil 3.14’de verilen bulanıklaştırma işlemi, MATLAB 7.0.1 Fuzzy Logic Toolbox ile oluşturulmuştur.

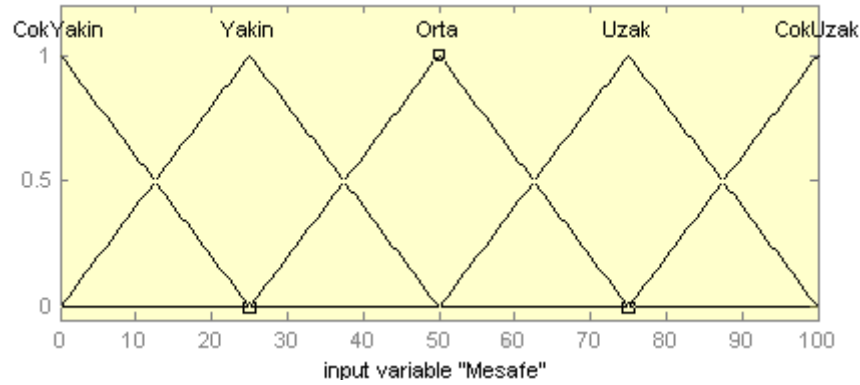


Şekil 3.14. Bulanık mantık giriş ve çıkış değişkenlerinin blok görünümü

- b.3. Mesafe 0 ile l_T arasında değişir. Bu aralık beş bölgeye ayrılarak üçgen üyelik fonksiyonları oluşturulur. Belirlenen l_T değerine göre mesafeler $[0,100]$ arasında ölçeklenir. l_T 'nin değeri 100'e karşılık gelir. Üyelik fonksiyonlarının adları: “Çok yakın”, “yakın”, “orta”, “uzak” ve “çok uzak”tır. Mesafe giriş değişkenine ait üyelik fonksiyonlarının sınırları Tablo 3.1’de ve grafik görünümü Şekil 3.15’de verilmiştir.

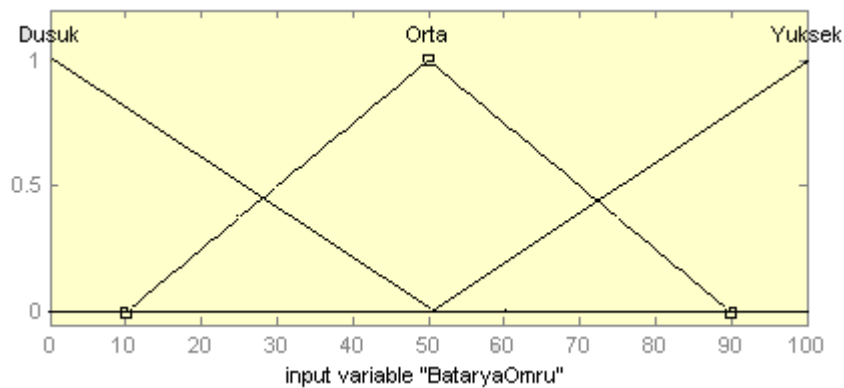
Tablo 3.1. Mesafe giriş değişkeni üyelik fonksiyonu sınırları

Dilsel Değişken	Üyelik Fonksiyonu sınırları		
Çok Yakın	0	0	25
Yakın	0	25	50
Orta	25	50	75
Uzak	50	75	100
Çok Uzak	75	100	100



Şekil 3.15. Mesafe giriş değişkeni üyelik fonksiyonlarının grafik görünümü

b.4. Batarya ömrü yüzde olarak 0 ile 100 arasında değişir. Bu giriş değişkeni için üç üyelik fonksiyonu oluşturulmuş olup “düşük”, “orta” ve “yüksek” dilsel değişkenleri üyelik fonksiyonlarına isim olarak verilmiştir. Batarya ömrü giriş değişkenine ait üyelik fonksiyonlarının sınırları Tablo 3.2’de ve grafik görünümü Şekil 3.16’da verilmiştir.



Şekil 3.16. Batarya ömrü giriş değişkeni üyelik fonksiyonlarının grafik görünümü

b.5. Yoğunluk yüzde olarak 0 ile 100 arasında değişir. Bu giriş değişkeni için üç üyelik fonksiyonu oluşturulmuş olup “düşük”, “orta” ve “yüksek” dilsel değişkenleri üyelik fonksiyonlarına isim olarak verilmiştir. Yoğunluk

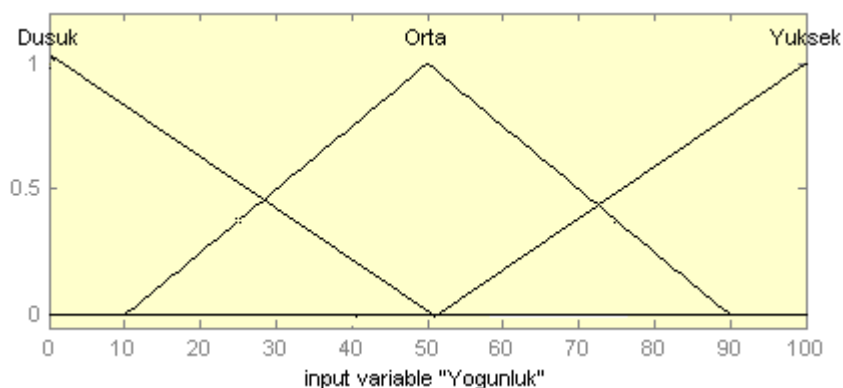
değişkenine ait üyelik fonksiyonlarının sınırları Tablo 3.3’de ve grafik görünümü Şekil 3.17’de verilmiştir.

Tablo 3.2. Batarya ömrü değişkeni üyelik fonksiyonu sınırları

Dilsel Değişken	Üyelik Fonksiyonu sınırı		
Düşük	0	0	50
Orta	10	50	90
Yüksek	50	100	100

Tablo 3.3. Yoğunluk giriş değişkeni üyelik fonksiyonu sınırları

Dilsel Değişken	Üyelik Fonksiyonu sınırı		
Düşük	0	0	50
Orta	10	50	90
Yüksek	50	100	100

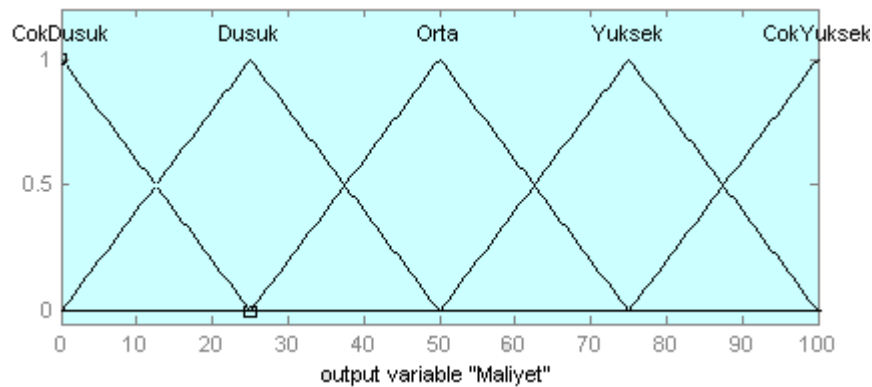


Şekil 3.17. Yoğunluk giriş değişkeni üyelik fonksiyonlarının grafik görünümü

b.6. Yukarıda verilen üç giriş değişkeni kullanılarak maliyet değeri elde edilir. Bu amaçla maliyet çıkış değişkeni için beş üyelik fonksiyonu oluşturulmuştur. Üyelik fonksiyonlarına “çok düşük”, “düşük”, “orta”, “yüksek” ve “çok yüksek” dilsel değişkenleri verilmiştir. Maliyet çıkış değişkenine ait üyelik fonksiyonlarının sınırları Tablo 3.4’de ve grafik görünümü Şekil 3.18’de verilmiştir.

Tablo 3.4. Mesafe giriş değişkeni üyelik fonksiyonu sınırları

Dilsel Değişken	Üyelik Fonksiyonu sınırı		
Çok Düşük	0	0	25
Düşük	0	25	50
Orta	25	50	75
Yüksek	50	75	100
Çok Yüksek	75	100	100



Şekil 3.18. Maliyet çıkış değişkeni üyelik fonksiyonlarının grafik görünümü

b.7. Oluşturulan üyelik fonksiyonları kullanılarak 45 adet kural oluşturulup bulanıklaştırma işlemi yapılmıştır. Oluşturulan kuralların bir kısmı aşağıya çıkarılmıştır.

If mesafe="Çok yakın" AND bataryaomru="Yuksek" AND yogunluk="Dusuk"
THEN maliyet="cok dusuk"

If mesafe="Çok yakın" AND bataryaomru="Yuksek" AND yogunluk="orta"
THEN maliyet="dusuk"

If mesafe="Çok yakın" AND bataryaomru="Orta" AND yogunluk="Dusuk"
THEN maliyet="Dusuk"

If mesafe="Yakın" AND bataryaomru="Orta" AND yogunluk="Dusuk" THEN
maliyet="Dusuk"

If mesafe="Yakın" AND bataryaomru="Orta" AND yogunluk="Orta" THEN
maliyet="Orta"

If mesafe="Çok Yüksek" AND bataryaomru="Düşük" AND
yogunluk="Yüksek" THEN maliyet="Çok Yüksek"

(Bulanık kuralların tamamı Ek C'de verilmiştir)

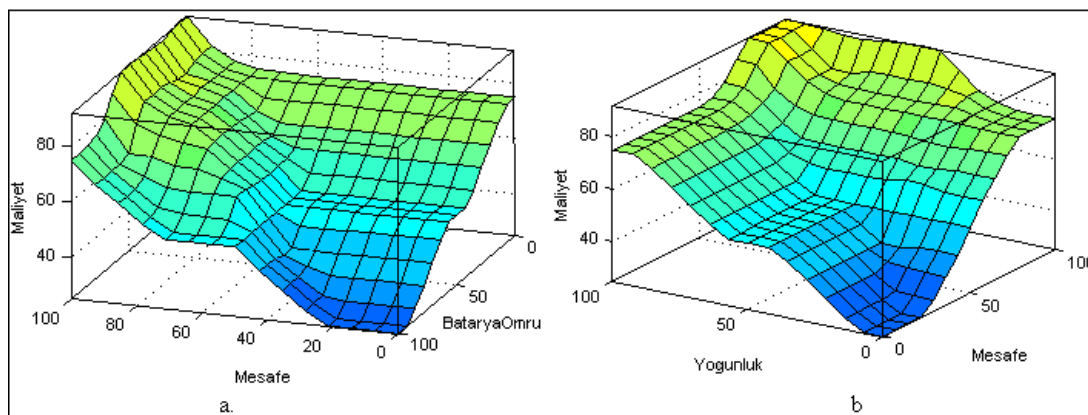
b.8. Durulaştırma (defuzzification) ile maliyet değerleri elde edilir. Durulaştırma için Mamdani yöntemi kullanılmıştır.

Tablo 3.5’de örnek giriş değerleri için bulanık mantık işlemi uygulanması sonucu elde edilen maliyetler verilmiştir.

Tablo 3.5. Örnek giriş değerleri için, bulanık mantıkla hesaplanmış maliyet değerleri

Mesafe	Batarya Ömrü	Yoğunluk	Maliyet
50	50	50	50
10	90	60	25
30	25	80	66
70	25	100	80
80	20	50	76

b.9. Batarya ömrü ve mesafeye karşı maliyetin değişim grafiği, Şekil 3.19.a’da ve yoğunluk ve mesafeye karşı maliyetin değişim grafiği de Şekil 3.19.b’de verilmiştir.



Şekil 3.19.a. Batarya ömrü ve mesafeye karşı maliyetin değişimi

Şekil 3.19.b. Yoğunluk ve mesafeye karşı maliyetin değişimi

c. Elde edilen düğümler arası maliyet değerleri G matrisine aktarılır. Bunun sonucunda k düğümlü bir ağ için oluşan G matrisi denklem 4.6’da verilmiştir.

$$G = \begin{bmatrix} 0 & m_{1,2} & \cdot & \cdot & m_{1,k} \\ m_{2,1} & 0 & \cdot & \cdot & m_{2,k} \\ \cdot & \cdot & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 & \cdot \\ m_{k,1} & m_{k,2} & \cdot & \cdot & 0 \end{bmatrix} \quad (3.6)$$

Matrisin esas köşegen elemanlarının değeri 0 olacaktır. Çünkü köşegenlerdeki değerler bir düğümün kendisine olan erişim maliyetini ifade eder. Düğümün kendisine olan erişim maliyeti de 0'dır.

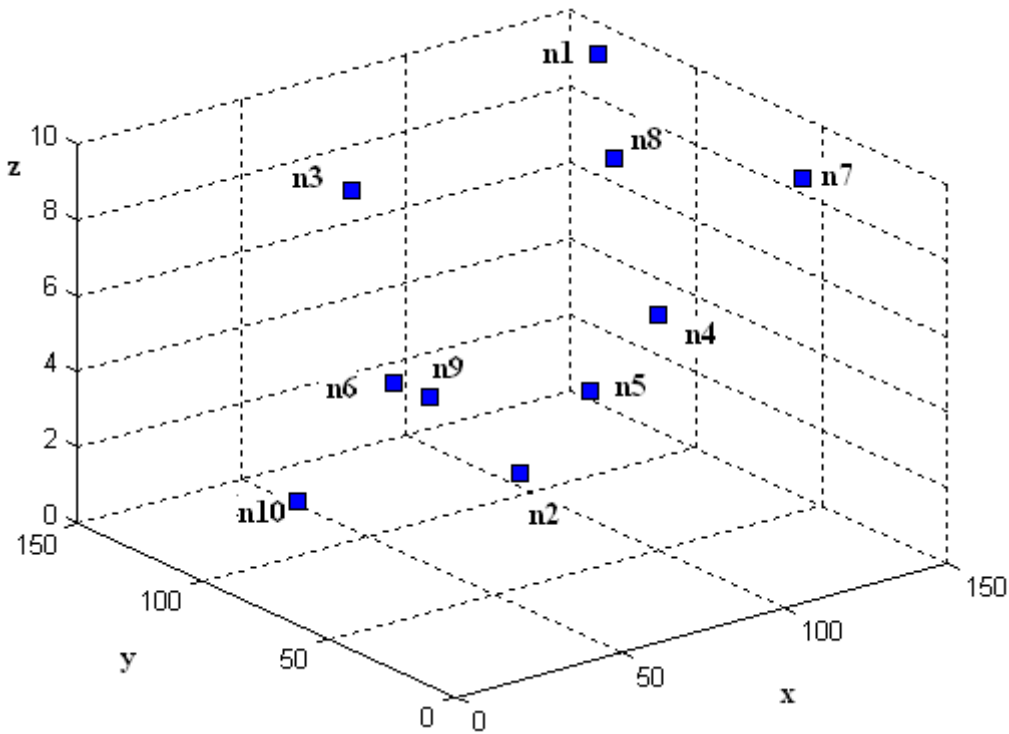
i düğümü ile j düğümü birbirinin yayın mesafesi (kapsama alanı) içinde değilse yapılan işlemler sonucu, G matrisinin $G(i,j)$ elemanının değeri sonsuz olarak bulunur.

3.5.2. Örnek bir ağ üzerinde maliyet değerlerinin belirlenmesi

10 düğümlü bir örnek ağ üzerinde düğümler arası maliyet değerlerinin hesaplanması için konum bilgileri ile yoğunluk ve batarya ömrü değerleri rastgele sayılar üretilerek seçilmiştir. Konum bilgileri için 220 metre x 220 metre x 10 metre alana dağılmış düğümler, batarya ve yoğunluk bilgileri için 0-100 arası değerler üretilmiştir. Elde edilen değerler Tablo 3.6'da verilmiştir. Tabloda X,Y ve Z; düğümlerin konumlarını, Batarya; düğümün batarya ömrünün yüzde cinsinden değerini, yoğunluk; düğümün tampon doluluk oranının yüzde cinsinden değerini ifade eder. Düğümlerin grafik görünümleri Şekil 3.20'dedir.

Tablo 3.6. 10 düğümlü örnek ağ için konum, batarya ve yoğunluk değerleri

	X	Y	Z	Batarya	Yoğunluk
n1	137	122	10	19	25
n2	90	92	1	50	59
n3	49	105	9	41	51
n4	72	13	8	47	47
n5	90	64	4	62	55
n6	24	56	6	8	95
n7	125	25	10	32	35
n8	144	125	7	61	41
n9	89	126	2	18	31
n10	4	68	3	63	42



Şekil 3.20. 10 düğümlü örnek ağıın düğüm yerleşimleri

Düğümler arası mesafeler Denklem 3.2’de verilen eşitlik kullanılarak hesaplanmış ve bulunan mesafeler metre cinsinden Tablo 3.7’de verilmiştir.

Tablo 3.7. 10 düğümlü örnek ağı için düğümler arası mesafe değerleri

Mes.	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
n1	0,00	56,48	89,63	126,93	74,89	130,92	97,74	8,19	48,83	143,71
n2	56,48	0,00	43,75	81,33	28,16	75,35	76,13	63,57	34,03	89,31
n3	89,63	43,75	0,00	94,84	58,20	55,09	110,35	97,10	45,72	58,57
n4	126,93	81,33	94,84	0,00	54,23	64,48	54,38	133,15	114,43	87,60
n5	74,89	28,16	58,20	54,23	0,00	66,51	52,75	81,52	62,04	86,10
n6	130,92	75,35	55,09	64,48	66,51	0,00	105,73	138,43	95,61	23,52
n7	97,74	76,13	110,35	54,38	52,75	105,73	0,00	101,83	107,52	128,60
n8	8,19	63,57	97,10	133,15	81,52	138,43	101,83	0,00	55,24	151,21
n9	48,83	34,03	45,72	114,43	62,04	95,61	107,52	55,24	0,00	102,91
n10	143,71	89,31	58,57	87,60	86,10	23,52	128,60	151,21	102,91	0,00

Tablo 3.6’deki batarya-yoğunluk bilgileri ve Tablo 3.7’deki hesaplanmış mesafe değerleri kullanılarak bulanıklaştırma işlemi sonucu elde edilen maliyet değerleri Tablo 3.8’de verilmiştir. Tablo 3.8’de “inf” olarak gösterilen değerler, düğümlerin birbirlerinin kapsama alanı dışında olduğunu göstermektedir.

Tablo 3.8. 10 düğümlü örnek ağ için düğümler arası mesafe değerleri

Maliyet	n1	n2	n3	n4	n5	n6	n7	n8	n9	n10
n1	0,00	75,73	Inf	Inf	Inf	Inf	Inf	48,85	67,82	Inf
n2	76,12	0,00	62,50	Inf	47,95	Inf	Inf	78,99	64,88	Inf
n3	Inf	62,50	0,00	Inf	74,51	91,68	Inf	Inf	67,82	73,79
n4	Inf	Inf	Inf	0,00	73,28	91,31	75,25	Inf	Inf	Inf
n5	Inf	50,00	76,41	75,17	0,00	91,68	75,02	Inf	78,67	Inf
n6	Inf	Inf	75,35	81,24	80,35	0,00	Inf	Inf	Inf	47,14
n7	Inf	Inf	Inf	75,20	73,10	Inf	0,00	Inf	Inf	Inf
n8	53,72	80,25	Inf	Inf	Inf	Inf	Inf	0,00	75,47	Inf
n9	66,06	50,00	64,77	Inf	76,62	Inf	Inf	74,43	0,00	Inf
n10	Inf	Inf	76,59	Inf	Inf	76,57	Inf	Inf	Inf	0,00

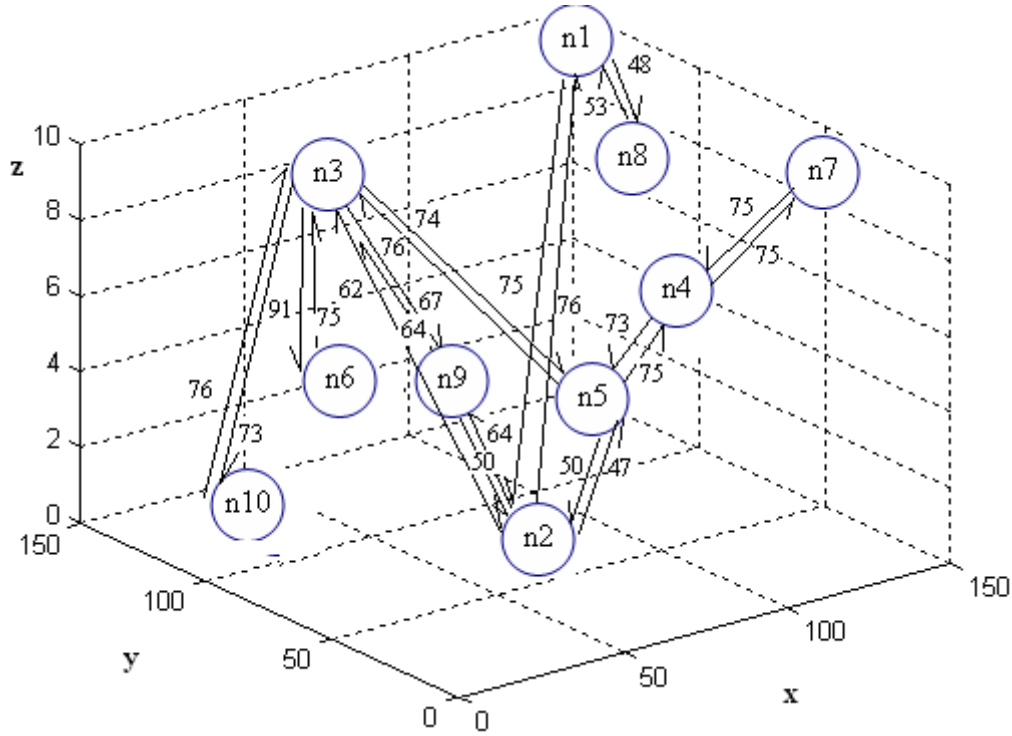
Elde edilen maliyet değerleri kullanılarak ağdaki düğümler graf yapısında oluşturulur. Geliştirilen algoritmada kullanılan graflar, Ek A'da özellikleri verilen yönlendirilmiş ve ağırlıklandırılmış graflardır.

Ek A'da detayları verilen $G(V,E)$ grafında V kümesinin elemanları; $n1$ 'den $n10$ 'a kadar düğümler, E kümesinin elemanları da maliyet değerleridir. Örnek 10 düğümlü ağın graf gösterimi Şekil 3.21'de verilmiştir. Bu graf, yönlendirilmiş bir graftır.

Tablo 3.7'den veya Şekil 3.21'den görüldüğü gibi, herhangi iki düğüm için, düğümler arası maliyet değerleri $G(i,j)$ ile $G(j,i)$ eşit olmak zorunda değildir. Böyle olmasının sebebi düğümlerin batarya ve yoğunluk değerlerinin farklı olmasıdır. Sadece mesafe değeri, maliyet olarak kullanılmış olsaydı, $G(i,j)=G(j,i)$ olurdu.

3.5.3. Bulanık mantık kullanmadan düğümler arası maliyet

Bu bölümde, önerdiğimiz algoritmada yönlendirme işleminin daha gerçekçi yapıldığını göstermek amacıyla maliyet değerlerinin belirlenmesinde bulanık mantık kullanmayan klasik yöntem de denenmiştir. Bu bağlamda 50 ve 100 düğümlü tasarsız ağlar oluşturulmuş ve bu ağlar içindeki düğümler arası maliyet değerleri klasik yöntem ve bulanık mantık yöntemiyle belirlenmiştir. Klasik yöntemde maliyet sadece düğümler arası mesafe olarak ele alınmıştır. Bulanık mantık yönteminde ise Bölüm 3.5.1'de verilen işlem sırası kullanılmıştır.



Şekil 3.21. 10 düğümlü örnek ağın düğüm yerleşimleri

Bu örnek ağlar üzerinde de klasik yöntem ve bulanık mantık kullanan yöntemlere göre maliyet değerleri belirlenmiştir. Her iki yöntemi birbiriyle karşılaştırmak amacıyla, 50 ve 100 düğümlü örnek ağlar oluşturulmuştur. Bu durumda maliyeti belirleyen düğümler arası mesafe olduğu için, bataryası tükenmek üzere olan veya yoğunluğu fazla olan düğüm yönlendirme amacıyla kullanıldığı görüldü. Aşağıda bu örnek ağlar ile alakalı bilgiler verilmektedir.

50 düğümlü örnek ağ: Benzetim programı tarafından, 400 metre x 400 metre x 10 metre alana dağılmış, 50 düğümlü bir ağ oluşturulmuştur. Bu ağ üzerindeki düğümlerin konum, batarya ve yoğunluk bilgileri de Tablo 3.9’da verilmiştir.

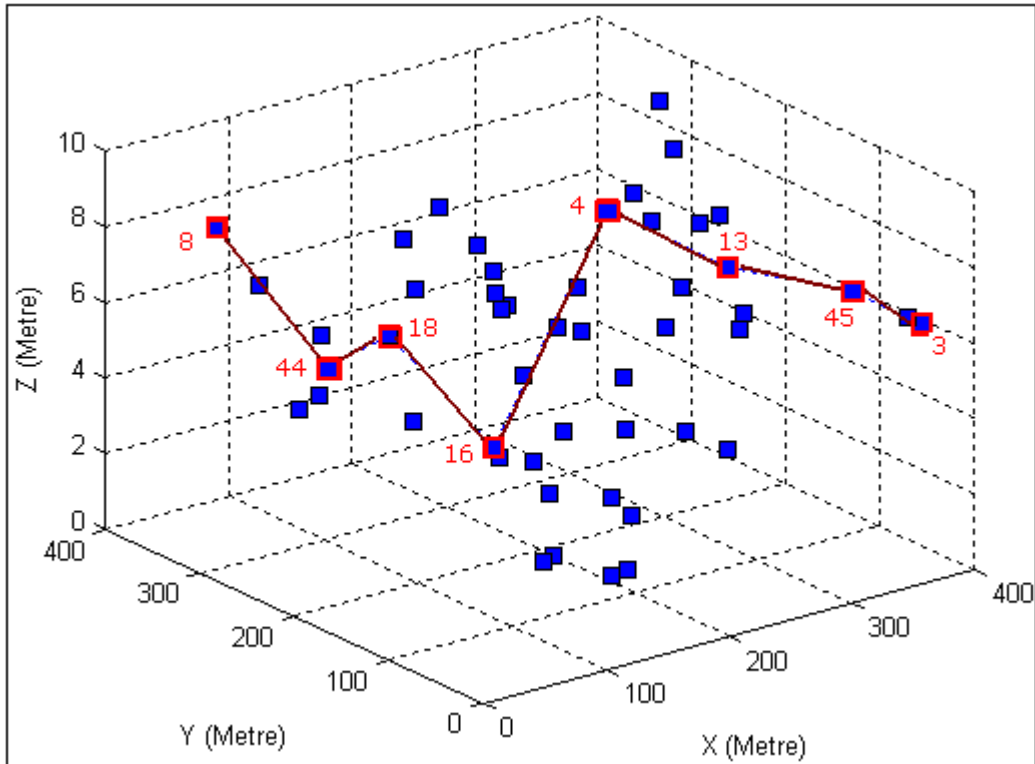
KTMYA Tablo 3.9’deki verilere göre bulanık mantık kullanılarak çalıştırıldığında 3. düğümden 8. düğüme olan en kısa yol {3 45 13 4 16 18 44 8} olarak bulunmuştur. Bu yol bilgilerine göre yönlendirmede kullanılacak yolların ve ağdaki düğümlerin görünümü Şekil 3.22’de verilmiştir. 3. düğüm ve 8. düğümün seçilmesinin sebebi, bu düğümlerin birbirlerine uzak noktalarda olmasıdır.

Tablo 3.9. 50 düğümlü ağın konum, batarya ve yoğunluk bilgileri

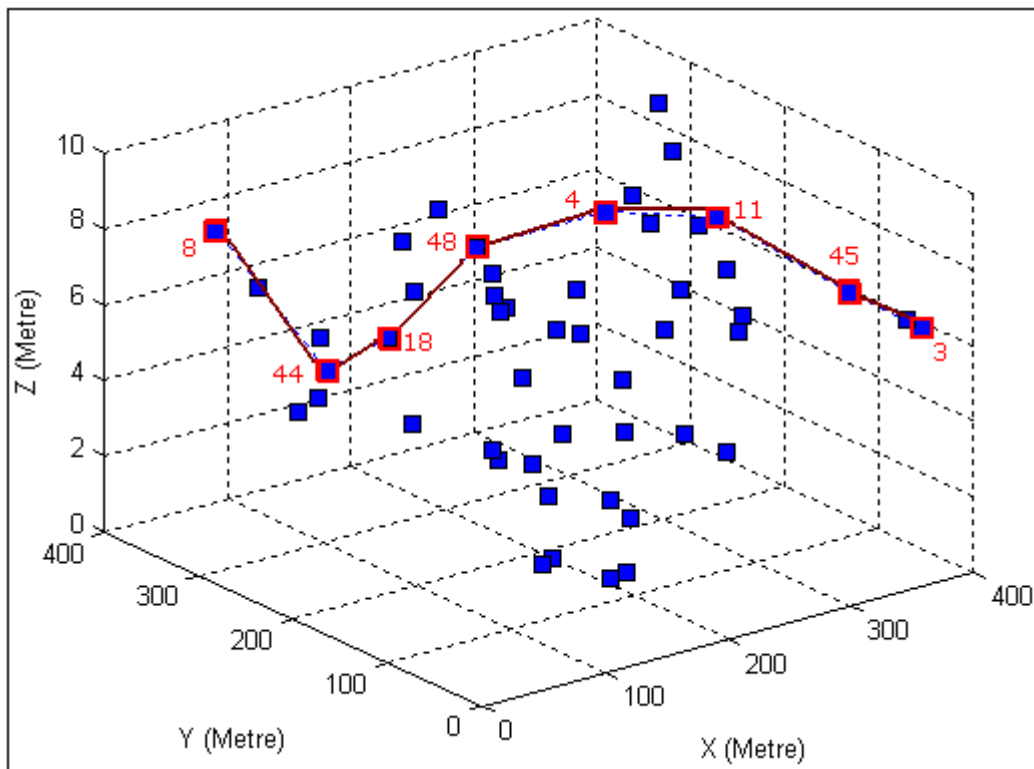
Düğüm	X	Y	Z	Bat.	Yoğ.	Düğüm	X	Y	Z	Bat.	Yoğ.
1	282	108	3	53	90	26	168	115	7	56	31
2	169	166	3	97	89	27	181	88	6	99	55
3	391	45	6	53	79	28	244	140	9	18	37
4	234	174	9	5	30	29	98	365	6	48	93
5	242	159	1	45	45	30	335	235	9	56	57
6	184	30	9	87	76	31	61	16	3	24	85
7	218	90	7	45	52	32	238	290	5	43	84
8	57	356	8	71	88	33	65	280	4	49	21
9	119	76	8	35	4	34	222	137	0	8	72
10	393	63	6	39	77	35	210	1	8	24	32
11	284	119	9	18	2	36	32	213	7	95	76
12	210	260	6	57	44	37	208	136	0	26	90
13	273	97	8	79	12	38	167	82	3	33	50
14	246	294	5	99	38	39	210	122	4	8	45
15	234	294	6	32	25	40	90	75	7	92	43
16	206	258	2	70	31	41	173	126	8	32	27
17	272	78	7	30	68	42	266	277	0	17	28
18	150	296	5	12	48	43	158	132	1	26	76
19	47	134	9	15	33	44	124	328	4	12	23
20	218	199	3	56	23	45	352	67	7	14	83
21	237	382	1	98	87	46	364	260	1	100	49
22	265	186	9	32	48	47	380	309	9	38	17
23	392	282	6	3	14	48	168	226	8	62	59
24	137	264	8	22	86	49	48	236	5	10	17
25	282	350	0	90	18	50	234	351	7	35	12

KTMYA aynı ağ üzerinde bulanık mantık kullanılmadan 3. düğümden 8. düğüme olan en kısa yol {3 45 11 4 48 18 44 8} olarak bulunmuştur. Bu yol bilgilerine göre yönlendirmede kullanılacak yolların görünümü Şekil 3.23'de verilmiştir.

Her iki yöntem sonucunda bulunan yollara bakıldığında, kaynaktan hedefe ulaşmak için atlama yapılacak düğüm sayısı aynıdır. Hatta yol üzerinde geçilecek düğümlerin 6 tanesi de aynıdır. Sadece iki atlama noktasında düğümler farklıdır. Bu fark yöntemlerin farklılığından kaynaklanmaktadır. Klasik yöntemde maliyeti belirleyen ölçüt düğümler arası mesafe olduğu için, bataryası tükenmek üzere olan veya yoğunluğu fazla olan düğümlerin kullanıldığı görülmüştür.



Şekil 3.22. 50 düğümlü ağda bulanık mantık kullanılarak belirlenmiş 3. düğümden 8. düğüme olan yol

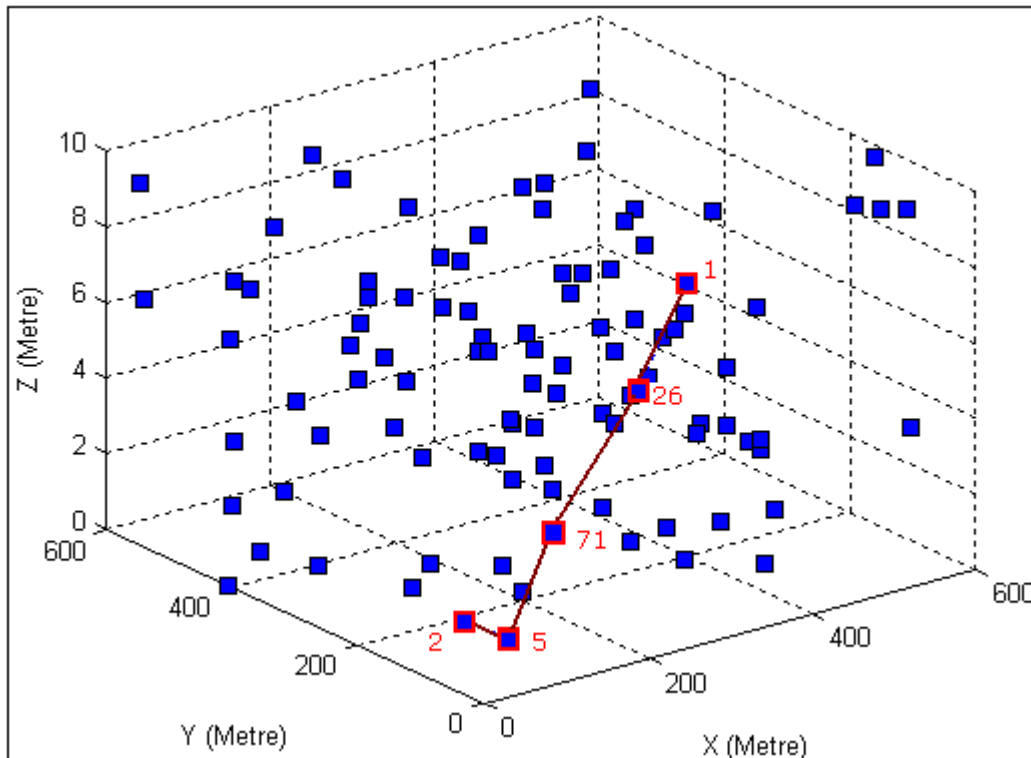


Şekil 3.23. 50 düğümlü ağda 3.düğümden 8. düğüme klasik yöntemle bulunmuş en maliyetsiz yol

50 düğümlü ağ örneği için ilgili durumlar Tablo 3.9’da yansıtılmıştır. Atlama noktası olarak seçilen düğümlerden 11 ve 13 nolu düğümlerde oluşan fark, batarya değerlerinden kaynaklanmaktadır. 16 ve 48 nolu düğümlerde oluşan fark yoğunluk değerlerinden kaynaklanmaktadır. Bu örnek sonuçtan da görüldüğü gibi bulanık mantık kullanımı, yönlendirmede görev yapacak düğümlerin Ek D’de enerji tüketim analizleri yapılan düğümlerin batarya ve yoğunluk bilgilerinin de belirleyici olmasını sağlamaktadır.

100 düğümlü örnek ağ: Benzetim programı tarafından, 600 metre x 600 metre x 10 metre alana dağılmış, 100 düğümlü bir ağ oluşturulmuştur. Bu ağ üzerindeki düğümlerin konum, batarya ve yoğunluk bilgileri de Tablo 3.10’da verilmiştir.

KTMYA Tablo 3.10’deki verilere göre bulanık mantık kullanılarak çalıştırıldığında 1. düğümden 2. düğüme olan en kısa yol {1 26 71 5 2} olarak bulunmuştur. Bu yol bilgilerine göre yönlendirmede kullanılacak yolların ve ağdaki düğümlerin görünümü Şekil 3.24’de verilmiştir.

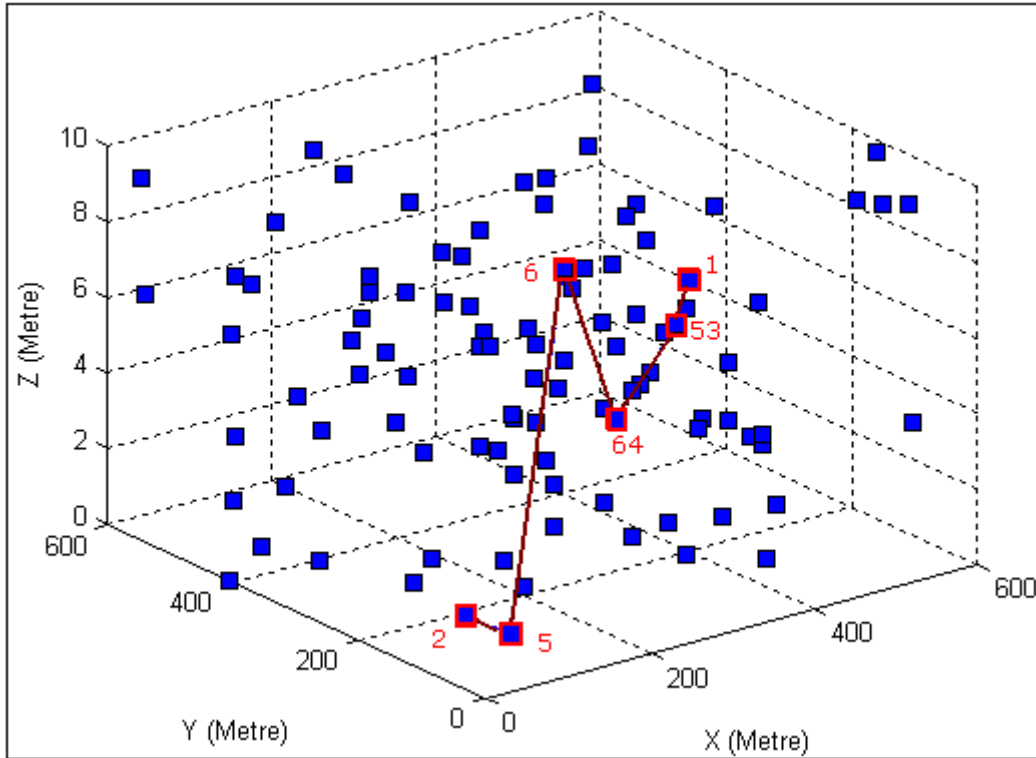


Şekil 3.24. 100 düğümlü ağda 1.düğümden 2. düğüme doğru bulanık mantıkla bulunmuş yol

Tablo 3.10. 100 düğümlü ağın konum, batarya ve yoğunluk bilgileri

Düğ	X	Y	Z	Bat	Yoğ	Düğ	X	Y	Z	Bat	Yoğ	Düğ	X	Y	Z	Bat	Yoğ
1	350	133	8	1	67	36	54	468	3	67	89	71	211	166	2	85	46
2	57	106	1	28	10	37	272	540	5	68	11	72	534	17	4	80	6
3	43	355	5	75	34	38	308	161	10	16	19	73	506	459	5	35	63
4	312	474	5	30	12	39	443	481	0	70	24	74	517	3	10	97	38
5	129	129	0	42	63	40	319	301	4	36	14	75	242	528	4	69	25
6	208	146	9	21	44	41	174	181	5	19	11	76	548	125	9	56	16
7	88	54	2	84	83	42	141	254	9	73	14	77	67	347	4	56	27
8	283	261	2	60	85	43	434	276	0	60	22	78	369	95	6	17	89
9	288	571	4	38	41	44	581	135	10	50	69	79	596	553	5	77	32
10	393	522	3	99	72	45	430	399	9	9	48	80	445	141	3	13	56
11	232	304	6	31	10	46	35	171	7	39	34	81	436	431	5	17	18
12	380	419	3	98	91	47	491	278	8	24	10	82	373	254	4	50	26
13	569	396	1	62	91	48	36	595	9	68	67	83	199	579	0	74	86
14	254	77	10	21	66	49	181	30	8	67	32	84	206	223	3	75	32
15	452	168	3	40	57	50	300	491	1	54	35	85	557	443	3	61	23
16	138	580	0	40	68	51	425	216	3	70	5	86	245	289	0	67	7
17	310	562	3	53	21	52	397	133	4	24	55	87	328	110	1	17	56
18	38	312	1	54	17	53	326	123	7	73	58	88	97	531	5	72	5
19	425	234	6	69	91	54	318	181	1	7	22	89	302	301	9	89	61
20	498	19	10	77	91	55	272	365	8	68	21	90	352	302	7	71	59
21	65	286	6	22	91	56	460	533	3	6	94	91	464	165	3	19	67
22	404	565	5	80	92	57	573	559	0	62	81	92	327	547	7	66	21
23	5	379	8	36	57	58	10	156	6	80	85	93	127	251	1	12	21
24	260	564	8	89	25	59	65	419	9	8	15	94	75	213	1	36	47
25	0	406	0	16	76	60	225	303	3	59	45	95	486	256	0	89	21
26	262	97	6	82	26	61	553	287	5	61	18	96	20	20	9	21	24
27	463	418	4	99	12	62	260	278	10	9	26	97	213	551	9	77	55
28	300	577	5	43	74	63	187	163	7	34	9	98	50	25	7	16	86
29	352	272	1	45	6	64	250	120	5	97	88	99	73	119	9	78	58
30	6	407	8	8	67	65	33	584	6	35	74	100	300	517	5	30	36
31	16	378	1	37	76	66	499	572	0	18	52						
32	346	356	9	9	54	67	542	446	2	7	84						
33	385	56	1	68	92	68	548	251	0	51	85						
34	539	532	9	12	21	69	232	283	3	28	39						
35	312	168	7	72	96	70	117	28	8	92	29						

KTMYA aynı 100 düğümlü ağ üzerinde bulanık mantık kullanılmadan 1. düğümden 2. düğüme olan en kısa yol {1 53 64 6 5 2} olarak bulunmuştur. Bu yol bilgilerine göre yönlendirmede kullanılacak yolların görünümü Şekil 3.25'te verilmiştir.



Şekil 3.25. 100 düğümlü ağda 1.düğümde 2. düğüme, bulanık mantık kullanılmadan bulunmuş yol

100 düğümlü örnek ağda, bulanık mantık kullanılarak ve bulanık mantık kullanılmadan en kısa yol belirlenmesi işlemleri arasında hem düğüm sayısı hem de yol üzerindeki düğümler açısından farklılıklar oluşmuştur. Ele alınan örnekte yönlendirme için bir fazla sayıda düğüm kullanılmıştır. Fazla sayıda düğüm kullanmak, hem uçtan uca gecikmeyi artırır hem de düğümlerin enerji kullanımına olumsuz etki yapar. Bu sebeplerden dolayı en kısa yol belirleme işleminde KTYA için bulanık mantık yönteminden faydalanmak gerekmektedir.

Ayrıca 50 ve 100 düğümlü ağlara ait tüm düğümlerin birbirlerine olan en kısa yolları, bulanık mantık yöntemiyle ve klasik yöntemle bulunmuştur. Bulunan en kısa yol bilgileri üzerinde yapılan analiz çalışmasının sonuçları aşağıda verilmiştir.

50 düğümlü ağ için:

Toplam yol bilgisi: $50 \times 50 - 50 = 2450$

Kaynaktan hedefe kadar geçilen maksimum düğüm sayısı: 6 (kaynak ve hedef hariç)

Klasik ve Bulanık mantık yöntemi arasında farklı düğümleri içeren yol bilgisi sayısı:

1163

Klasik ve Bulanık mantık yöntemi tarafından aynı düğümler içeren yol bilgisi sayısı:
1287

Klasik ve Bulanık mantık yöntemi arasında farklı düğümleri içeren yol bilgisinin,
toplam yol bilgisine oranı: % 47,5

Klasik ve Bulanık mantık yöntemi arasında farklı sayıda düğüm içeren yol bilgisi
sayısı: 167

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısı: 152

Klasik yöntemde daha az atlamaya sahip yol bilgisi sayısı: 15

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısının, toplam yol bilgisi
sayısına oranı: % 6,2

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısının, klasik mantıktan farklı
düğümleri kullanan yol bilgisi sayısına oranı: % 13

100 düğümlü ağ için:

Toplam yol bilgisi: $100 \times 100 - 100 = 9900$

Kaynaktan hedefe kadar geçilen maksimum düğüm sayısı: 11 (kaynak ve hedef
hariç)

Klasik ve Bulanık mantık yöntemi arasında farklı düğümleri içeren yol bilgisi sayısı:
5056

Klasik ve Bulanık mantık yöntemi tarafından aynı düğümler içeren yol bilgisi sayısı:
4844

Klasik ve Bulanık mantık yöntemi arasında farklı düğümleri içeren yol bilgisinin,
toplam yol bilgisine oranı: % 51,07

Klasik ve Bulanık mantık yöntemi arasında farklı sayıda düğüm içeren yol bilgisi
sayısı: 582

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısı: 582

Klasik yöntemde daha az atlamaya sahip yol bilgisi sayısı: 0

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısının, toplam yol bilgisi
sayısına oranı: % 5,88

Bulanık mantıkta daha az atlamaya sahip yol bilgisi sayısının, klasik mantıktan farklı
düğümleri kullanan yol bilgisi sayısına oranı: % 11,51

Yukarıdaki sonuçlardan da görüldüğü gibi, bulanık mantık kullanımı, batarya ömrü ve düğüm yoğunluğu bilgilerini de göz önünde bulundurarak en kısa yol belirleme işeminin, sadece mesafeye dayalı klasik yöntemle göre üstünlük sağlamaktadır.

3.6. KTMYA'da Ölçekleme

Bu kısımda bir gezgin tasarsız ağda düğüm sayısının artması durumunda, ağın ölçeklenebilirliğinin sağlanması konusu üzerinde durulmaktadır. Geliştirilen algoritmanın, yüksek düğüm sayılı ağlara uygulanabilmesi için bir çözüm ileri sürülmektedir. Düğümlerin konum bilgilerine göre kategorize edilmesini öneren bu çözüm, örnek ağlar üzerinde uygulanmaktadır. Bu amaçla öncelikli olarak Ek E'de literatürde yer alan kümeleme algoritmaları hakkında bilgi verilmektedir. KTMYA'ya uygulanacak kümeleme algoritmasının kararı da bu bölümde verilmektedir.

Ağdaki düğümlerin sayısı veya dağılımı arttığında;

- i. Veri paketlerinin bir uçtan diğer uca iletilmesi için harcanacak enerji artar.
- ii. Master düğüm üzerine düşen hesaplama yükü artar
- iii. Ağda paket alamayan düğüm sayısının artması ihtimali artar.
- iv. Yol isteklerine cevap vermede gecikmeler oluşabilir.

Gezgin tasarsız ağların Bölüm 1'de verilen kısıtları dikkate alınarak, tasarımı yapılan algoritmanın bu problemlere çözüm üretmesi gerekir. KTMYA ölçekleme için düğümlerin sınıflara ayrılması önerilmektedir. Oluşturulan her sınıf içinde bir master düğüm atanarak o sınıfa ait yönlendirme bilgilerini tutması ve yol isteklerine cevap vermesi sağlanmaktadır.

Algoritmada sınıflandırma işlemine başlanması için üç kriter kullanılmaktadır.

Bunlar;

1. Ağdaki düğüm sayısı
2. Ağdaki düğümlerin dağılımı

3. Kaynaktan hedefe paket gönderimi esnasında geçilebilecek en fazla düğüm sayısı

Ağdaki düğüm sayısı zaten ölçülebilir bir büyüklüktür. Düğümlerin dağılımına karar vermek için ise M mesafe matrisi kullanılacaktır. Bir matrisin normu onun dağılımı hakkında bilgi verir. Norm değeri büyüdükçe matrisin dağılımı artar [47]. KTMYA ağdaki düğümlerin birbirlerine uzaklıkları M matrisinde tutulduğu için M matrisinin normu belirlenen bir eşik değerin üzerine çıktığında, ağda hedefe ulaşmak için atlanması gereken düğüm sayısı belli bir eşik üzerine çıktığında veya ağdaki düğümlerin sayısı belirlenen bir eşik değeri aştığında sınıflandırma işlemi başlatılır.

Ağın kaç sınıfa ayrılacağına kararını da düğüm sayısı ve norm değeri ile belirler.

Literatürde kullanılan sınıflandırma algoritmalarından bazıları şunlardır:

- Hiyerarşik kümeleme
- K-Means kümeleme
- Eksiltmeli kümeleme
- Fuzzy C-Means (FCM) kümeleme

Bunlardan hiyerarşik kümeleme, K-means kümeleme ve eksiltmeli kümeleme yöntemleri Ek E’de sunulmuştur. Fuzzy C-means kümeleme yöntemi ise bu bölümde incelenmiştir.

3.6.1. Fuzzy c-means kümeleme

FCM, her bir veri noktasını bir kümeye belli bir üyelik derecesi ile dahil eden bir sınıflandırma algoritmasıdır. 1981 yılında Jim Bezdek tarafından ilk olarak ortaya konulmuştur. FCM, çok boyutlu uzaydaki noktaları belli sayıdaki gruplara ayırmayı sağlayan bir metottur [48].

FCM algoritması $X = \{x_1, x_2, x_3, \dots, x_n\}$ veri setinin elemanlarını verilen kriterlere uygun olarak parçalara ayırmayı amaçlar. Algoritma giriş parametresi olarak sonlu veri kümesi alıp, $V = v_i, i = 1, 2, \dots, c$ kümesi ile verilen küme merkezlerini ve

$U = u_{ij}, i = 1, 2, \dots, c, j = 1, 2, \dots, n$ matrisi ile verilen küme elemanları matrisini geri döndürür. U matrisi içinde u_{ij} 0 ile 1 arasında sayısal değerler alır. Bu değerlere x_j 'nin, i . kümeye aitlik derecesi denir.

FCM algoritmasının çalışma adımları aşağıdadır:

1. Küme sayısı c ($2 \leq c \leq n$), üstel ağırlık μ ($1 < \mu < \infty$), başlangıç küme matrisi U^0 ve sonlandırma kriteri ε seçilir. Aynı zamanda iterasyon indisi 1'den 0'a olarak ayarlanır.
2. Fuzzy küme merkezleri olan $\{v_i^1 | i=1, 2, \dots, c\}$, U^1 kullanılarak hesaplanır.
3. Yeni küme matrisi U^{l+1} , $\{v_i^1 | i=1, 2, \dots, c\}$ kullanılarak hesaplanır.
4. Yeni küme matrisi $\Delta = \|U^{l+1} - U^l\| = \max_{i,j} |u_{ij}^{l+1} - u_{ij}^l|$ hesaplanır. l iterasyon değeri 1 artırılıp ($l=l+1$) 2. adıma gidilir. Eğer $\Delta \leq \varepsilon$ ise durur.

MATLAB 7.0.1 Fuzzy Logic Toolbox'ın *fcm* komutu, küme merkezini tahmini bir değer olarak işleme başlar. Amaç her bir kümenin orta noktasını işaretlemektir. Başlangıçta kullanılan tahmini değer doğru olmayabilir. Her bir noktanın seçilen küme merkezlerine olan uzaklıklarına göre üyelik dereceleri atanır. İterasyonlarla küme merkezleri ve her bir kümenin üyelik dereceleri bulunur. Bu iterasyon bir nesnel fonksiyonun minimize edilmesine dayanır ve bir nokta ile merkezin arasındaki mesafeyi noktanın üyelik derecesi ile ağırlıklandırır [49].

m ; nokta sayısı, n ; noktaların koordinatları olmak üzere $(m \times n)$ boyutundaki bir matrisin elemanlarını sınıflandırma yapmak için, bir U matrisinde her bir noktanın her bir kümeye ait üyelik dereceleri tutulur. Üyelik derecesi 0 ile 1 arasında bir değerdir. İterasyonlarla küme merkezi ve küme üyeleri bulunur.

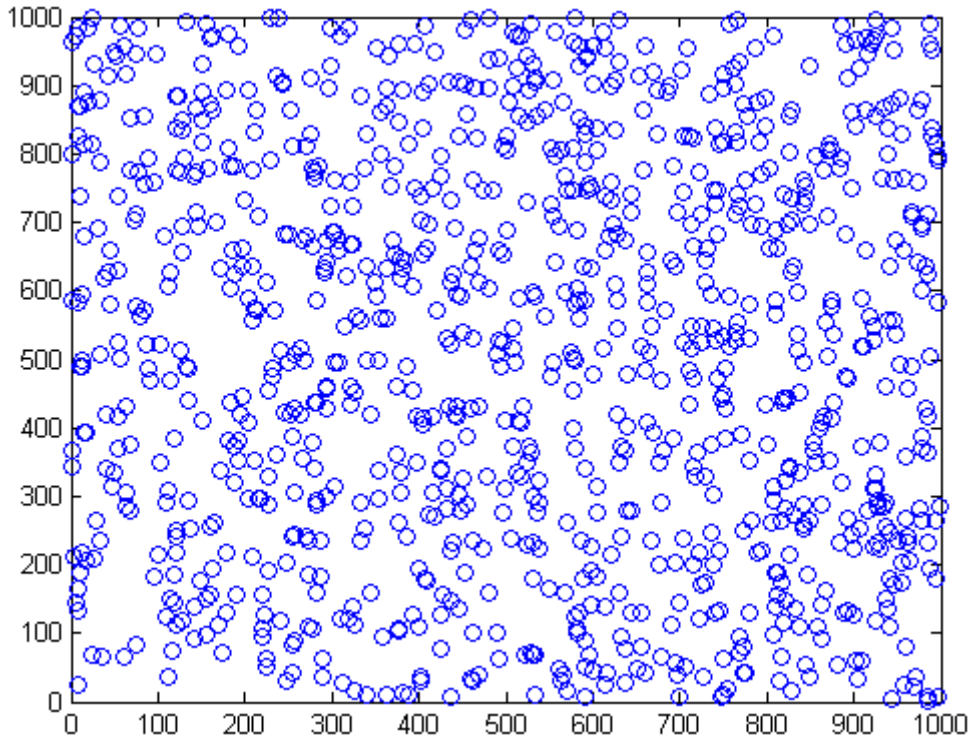
3.6.2. KTMYA'da düğümlerin kümelere ayrılması

Bu tez çalışmasında, KTMYA'da düğümlerin kümelere ayrılması işlemi için, Ek E'de verilen kümeleme algoritmaları ve Bölüm 3.6.1'de verilen fuzzy c-means (FCM) algoritması arasında seçim yapılması konusunda [50]'de yapılmış çalışmadan faydalanılmıştır. [50]'deki enerji verimliliği açısından düğümlerin kümelere

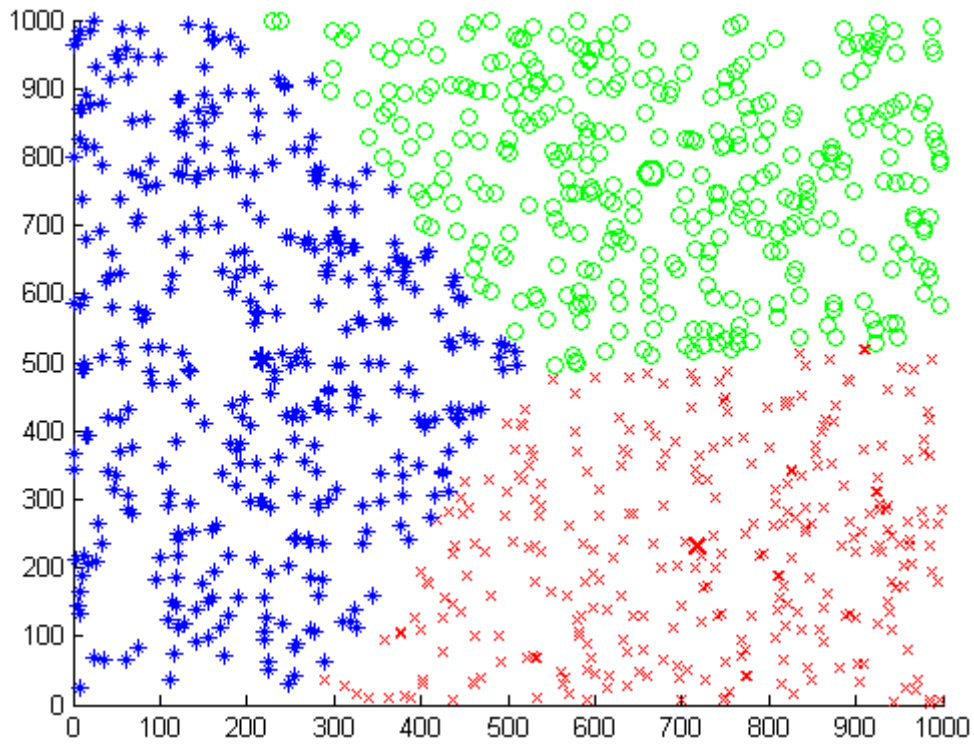
ayrılması ile ilgili çalışmada, kümeleme algoritmaları karşılaştırılmış ve en az enerji tüketimine FCM ile ulaşılmıştır. Bu yüzden bu tez çalışmasında da FCM tercih edilmiştir. Ayrıca eksiltmeli kümeleme yöntemi, küme sayısını kendi belirlediği için daha uzun sürede sonuca ulaşır.

Denklem 3.1’de verilen P konum bilgileri matrisi FCM algoritmasına veri seti olarak uygulanır. FCM, P matrisindeki düğüm konumlarını kullanarak birbirlerine yakın olan düğümleri, belirlenen küme sayısına sınıflandırır. Algoritmanın giriş değişkenleri, P matrisi ve küme sayısı, çıkış değişkenleri ise V küme merkezleri matrisi ve U küme elemanları aitlik dereceleri matrisidir.

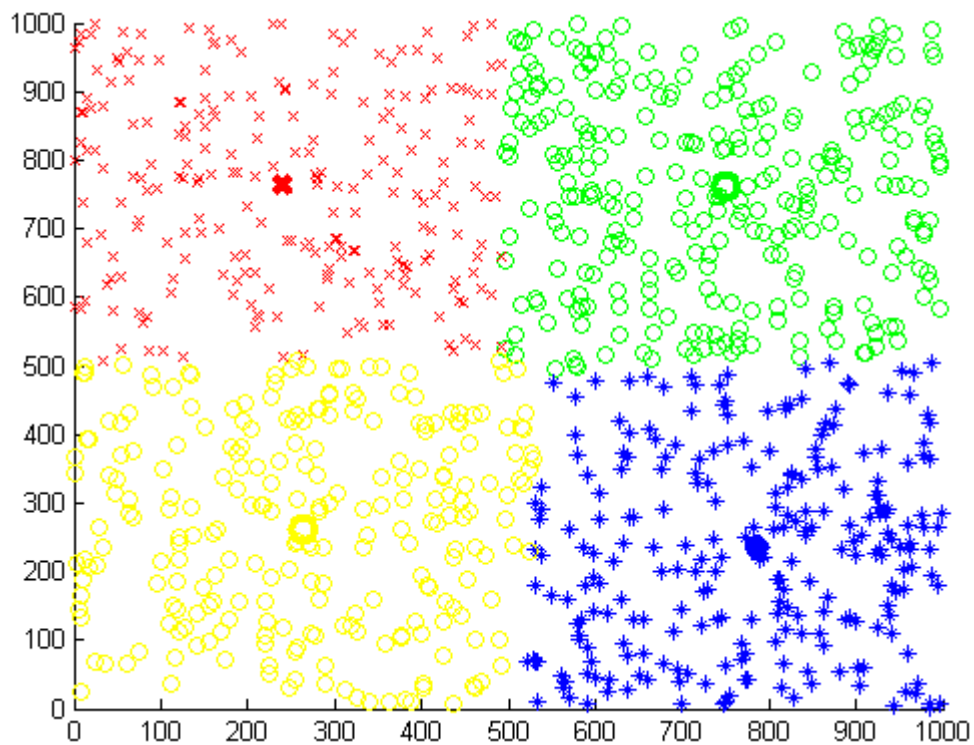
Şekil 3.26’da 1000 metre x 1000 metre alana rastgele dağılmış, 1000 düğümlü bir ağ görülmektedir. Şekil 3.27’de ise 1000 düğümlü ağın, FCM kullanılarak 3 kümeye ayrılmış hali görülmektedir. Şekilde her bir kümenin merkezi, koyu renkle işaretlenmiştir. Şekil 3.28’de aynı ağın, FCM kullanılarak 4 kümeye ayrılmış hali, Şekil 3.29’da 1000 düğümlü ağın, FCM kullanılarak 5 kümeye ayrılmış hali görülmektedir. Şekil 3.30’da 1000 düğümlü ağın, FCM kullanılarak 6 kümeye ayrılmış hali verilmiştir.



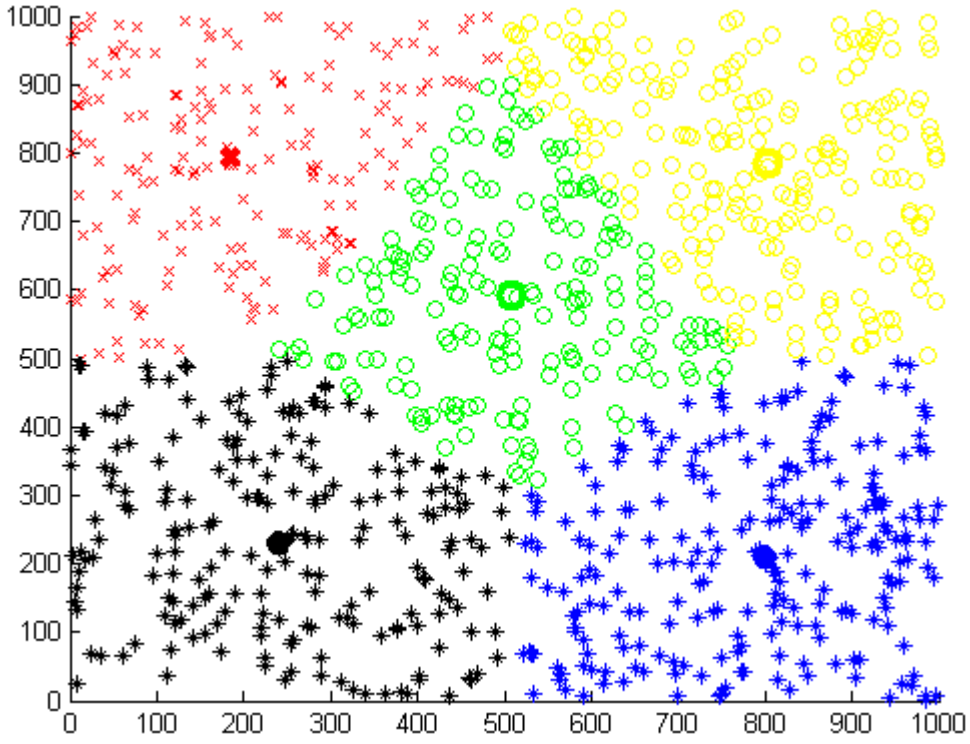
Şekil 3.26. 1000 düğümlü ve kümelere ayrılmamış ağın düğüm görünümü



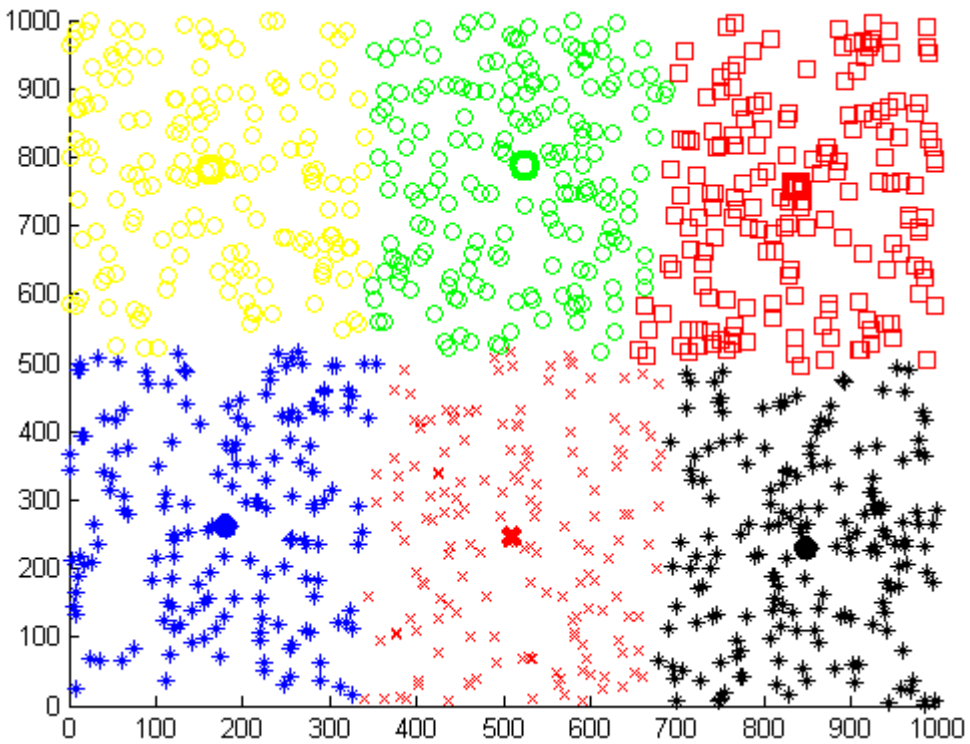
Şekil 3.27. 1000 düğümlü ağın 3 kümeye ayrılmış hali



Şekil 3.28. 1000 düğümlü ağın 4 kümeye ayrılmış hali



Şekil 3.29. 1000 düğümlü ağın 5 kümeye ayrılmış hali



Şekil 3.30. 1000 düğümlü ağın 6 kümeye ayrılmış hali

Algoritmaya küme sayısı ve düğüm konumlarının giriş verileri olarak uygulanması sonucunda, her bir düğümün hangi kümeye ne ölçüde üye olduğu ve her bir kümenin merkez koordinatları üretilir. Ağdaki düğümlerden hangilerinin küme masterları olduğunun belirlenmesi gerekir. Bunun için, elde edilen küme merkezinin koordinatlarına en yakın düğümden başlanılarak düğümlerin küme masterları olup olamayacakları sorgulanır. Küme merkezinin koordinatlarına en yakın düğümü belirlemek için yeni bir işlem yapılmasına gerek yoktur. Bir küme içindeki en yüksek üyelik derecesine sahip olan düğüm, kümenin en merkezinde olan düğümdür. Çünkü FCM algoritması küme merkezine olan uzaklığa göre üye aitlik derecelerini oluşturur.

KTMYA’da kümeleme işlemi aşağıdaki adımlar kullanılarak uygulanır.

1. Küme sayısı ve düğüm konumları verilir.
2. MATLAB 7.0.1 *fcm* fonksiyonu kullanılarak küme merkezleri ve düğümlerin her bir kümeye olan aitlik dereceleri 0 ile 1 arasında bir değer olarak üretilir.
3. Her bir kümeye ait küme masterı, küme içindeki en yüksek aitlik derecesine sahip düğüm olarak belirlenir. Aşağıdaki ifade U aitlik derecesi matrisinin küme masterlarını verir.

$$C_c = \max \left\{ \left\{ U_{c,j} \right\} \right\} \quad (3.7)$$

Denklem 3.7’deki C_c , c ’inci küme merkezidir. c , U matrisinin satır numarası ve j , U matrisinin sütun numarasıdır.

4. Eğer küme merkezindeki düğüm batarya veya yoğunluk gibi bir sebepten dolayı küme masterı olmayı kabul edemeyecekse, ikinci sıradaki aitlik derecesine sahip düğüm küme masterı olarak belirlenir.
5. Master düğüm kendisinde bulunan yönlendirme bilgilerini (P konum matrisi, M mesafe matrisi ve G düğümler arası maliyet marisi) küme masterlarına gönderir. Her bir küme masterına sadece kendi kümesinde olan düğümlerin bilgisini ve diğer kümelerin küme masterlarının hangi düğümler olduğunu bildirir.

6. Küme masterları, aynı kümedeki düğümlere kendisinin master düğüm olduğunu bildirir ve Bölüm 3'te verilen algoritma adımları işletilir.
7. Kümeler arası iletişim için, her iki kümenin arasında bulunan bir düğüm ara düğüm olarak seçilir. Ara düğümler kümeler arası yönlendirme elemanı olarak çalışırlar. Bu düğümler, ilgili kümelere aitlik dereceleri arasındaki farkın mutlak değerinin en küçük olduğu düğümlerdir. Aşağıdaki ifade kullanılarak ara düğüm bulunur.

$$I_{r,s} = \min \left\{ |U_{r,j} - U_{s,j}| \right\} \quad (3.8)$$

Denklem 3.8'de r,s ara düğümleri bulunacak küme numaralarıdır.

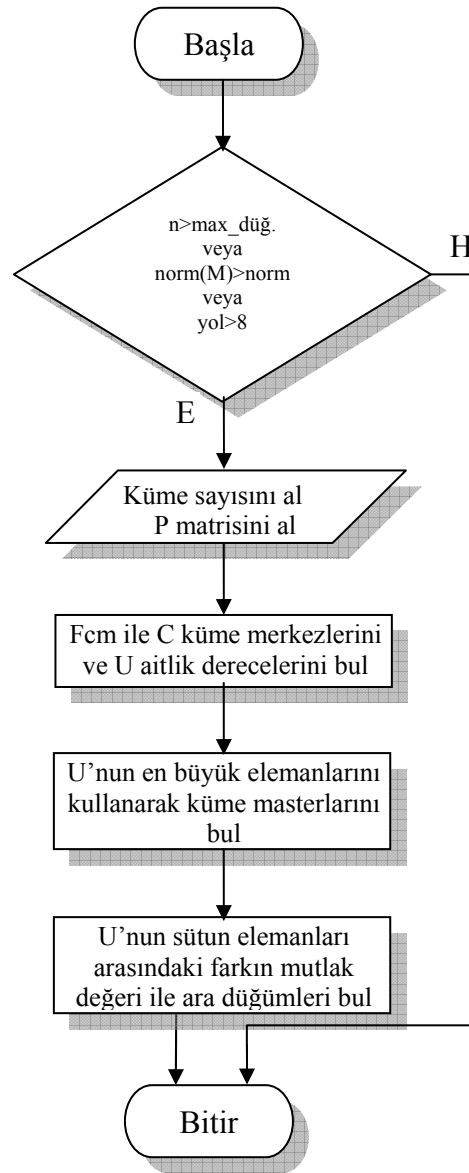
Yukarıda verilen işlem adımları Şekil 3.31'deki akış diyagramı ile de ifade edilmiştir.

3.6.3. Örnek ağ üzerinde kümeleme işleminin uygulanması

Yukarıda aşamaları verilen kümeleme adımlarının nasıl çalıştığını göstermek için bir örnek ağ sunulmuştur. Bu ağ 10x10 metre kare alana rastgele dağılmış 20 düğümlü bir ağdır. Bu ağa ait koordinat bilgileri de Tablo 3.11'de verilmiştir.

Tablo 3.11. 20 düğümlü ağın konum bilgileri

Düğüm No	X	Y	Düğüm No	X	Y
1	3	3	11	10	5
2	2	0	12	0	1
3	4	5	13	5	3
4	2	4	14	6	0
5	1	3	15	1	4
6	6	0	16	9	2
7	0	10	17	0	8
8	8	6	18	4	2
9	0	6	19	8	9
10	10	0	20	3	1



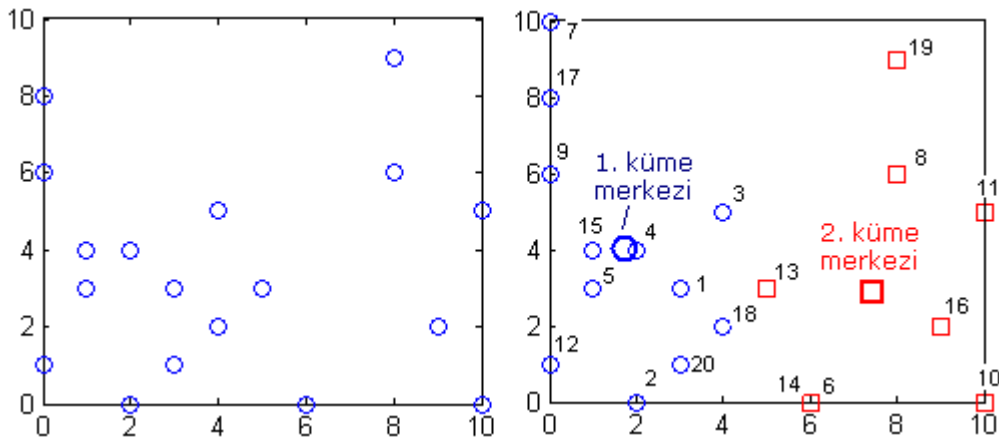
Şekil 3.31. Kümeleme ve kümeler arası iletici düğüm belirleme akış diyagramı

Bu ağın iki kümeye ayrılabilmesi için oluşturulan aitlik dereceleri Tablo 3.12’de verilmiştir. Bu ağın düğümlerinin ve iki kümeye ayrılmış halinin görünümü Şekil 3.32’de verilmiştir. Şekil 3.32’de daire ile gösterilen düğümler birinci kümenin elemanlarını, kare ile gösterilen düğümler ikinci kümenin elemanlarını simgelemektedir.

Tablo 3.12. 20 düğümlü ağıın küme aitlik dereceleri

Düğüm	1	2	3	4	5	6	7	8	9	10
1.Küme	0,881	0,697	0,733	0,998	0,961	0,231	0,734	0,19	0,905	0,149
2.Küme	0,119	0,303	0,267	0,002	0,039	0,769	0,266	0,81	0,095	0,851
Düğüm	11	12	13	14	15	16	17	18	19	20
1.Küme	0,138	0,826	0,339	0,231	0,987	0,053	0,814	0,58	0,373	0,681
2.Küme	0,862	0,174	0,661	0,769	0,013	0,947	0,186	0,42	0,627	0,319

1. Küme merkezi (1,7512 ; 4,0523) ve 2. küme merkezi (7,4422 ; 2,8806) olarak elde edilmiştir. 1. kümeye ait düğümler: 1,2,3,4,5,7,9,12,15,17,18,20 ve 2. kümeye ait düğümler: 6,8,10,11,13,14,16,19 olarak bulunmuştur.



Şekil 3.32. 20 düğümlü ağıın kümelenmemiş ve iki kümeye ayrılmış görünümü

Tablo 3.12 ve Şekil 3.31 birlikte incelendiğinde; 18 numaralı düğümün her iki küme merkezine de yakın olduğu görülmektedir. Bunun yanında bu düğümün birinci kümeye aitlik derecesi 0,58 ve ikinci kümeye aitlik derecesi 0,42'dir. Bir başka örnek ele alırsak; 5 numaralı düğümün birinci kümeye aitlik derecesi 0,961 iken, ikinci kümeye aitlik derecesi 0,039'dur. Bu değerler, 5 numaralı düğümün birinci küme merkezine çok daha yakın olduğunu göstermektedir.

Tablo 3.12 ve Şekil 3.31'den çıkarılabilecek bir başka sonuç, birinci kümeye ait düğümlerden, aitlik derecesi en yüksek olan düğüm 4 numaralı düğümdür. Şekil 3.31 üzerinde de küme merkezine en yakın düğüm 4 numaralı düğüm olarak görülmektedir. Aynı değerlendirme diğer küme için de yapıldığında 16 nolu düğüm küme merkezine en yakın düğüm olarak görülmektedir.

3.7. KTMYA'nın Örnek Bir Ağ Üzerinde Uygulanması

KTMYA'nın etkisinin daha iyi anlaşılabilmesi için 20 düğümlü bir ağ üzerinde, tez içinde açıklanan bütün özellikler uygulanarak, elde edilen sonuçlar değerlendirilmektedir.

Örnek ağ uygulanması işlemi adım adım açıklanarak alınan sonuçlar aşağıda değerlendirilmektedir.

- 1) 280x280 m² alana rastgele dağılmış 20 düğümlü ağ oluşturuldu. Denklem 3.1'de P matrisi olarak tanımlanan, düğümlerin konum bilgileri ile batarya ömrü ve yoğunluk bilgilerini içeren veriler Tablo 3.13'te verilmektedir.

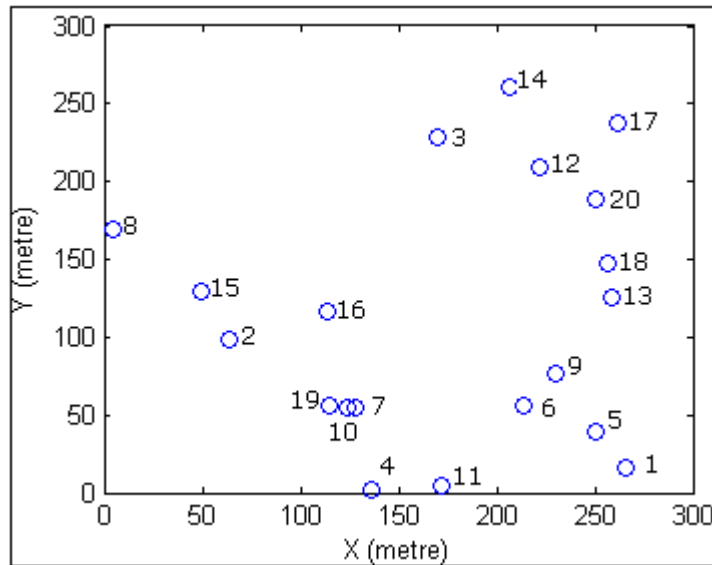
Tablo 3.13. 20 düğümlü ağın P matrisi verileri

Düğüm	X	Y	Z	Bat.	Yoğ.
1	266	16	9	50	80
2	64	99	0	90	96
3	170	228	7	83	53
4	136	2	4	65	89
5	250	39	9	82	18
6	214	56	5	67	98
7	128	55	7	35	28
8	5	169	4	29	26
9	230	76	3	35	88
10	124	55	2	54	74
11	172	4	2	73	14
12	222	209	7	31	2
13	259	125	3	84	90
14	207	261	5	57	20
15	49	130	1	38	30
16	114	117	7	71	67
17	262	237	4	55	29
18	257	147	9	45	47
19	115	56	9	70	7
20	251	188	6	63	99

- 2) Ağdaki düğümler arası mesafeler denklem 3.3'te verilen M matrisinde tanımlandığı haliyle oluşturulmuş ve Tablo 3.14'de verilmiştir. Ağdaki düğümlerin yerleşimleri ise Şekil 3.33'te verilmiştir.

Tablo 3.14. 20 düğümlü ağda düğümler arası mesafeler

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	218,6	232,7	130,9	28	65,7	143,4	302,6	70,2	147,4	95	198	109,4	252	245,3	182,5	221,1	131,3	156,2	172,7
2	218,6	0	167,1	120,9	195,7	156,1	78	91,6	167,6	74,4	143,9	192,7	196,8	216,1	34,5	53,6	241,4	199,1	67,3	207,2
3	232,7	167,1	0	228,6	205,2	177,6	178	175,3	163,5	179,1	224,1	55,4	136,2	49,6	155,8	124,3	92,5	118,9	180,6	90,3
4	130,9	120,9	228,6	0	120	94,9	53,7	212,3	119,6	54,4	36,1	224,2	174	268,6	154,8	117,1	266,7	188,9	58,2	218,7
5	28	195,7	205,2	120	0	40	123,1	277,4	42,5	127,2	85,8	172,3	86,7	226,2	220,8	156,8	198,4	108,2	136,1	149
6	65,7	156,1	177,6	94,9	40	0	86	237,6	25,7	90,1	66,9	153,2	82,4	205,1	180,9	117,2	187,3	100,7	99,1	137,1
7	143,4	78	178	53,7	123,1	86	0	167,7	104,2	6,4	67,5	180,4	148,6	220,6	109,1	63,6	226	158,5	13,2	181,2
8	302,6	91,6	175,3	212,3	277,4	237,6	167,7	0	243,5	164,8	234,8	220,7	257,8	222	58,9	120,8	265,8	253	157,8	246,7
9	70,2	167,6	163,5	119,6	42,5	25,7	104,2	243,5	0	108,1	92,5	133,3	56,9	186,4	188,9	123,1	164,2	76,2	116,9	114
10	147,4	74,4	179,1	54,4	127,2	90,1	6,4	164,8	108,1	0	70	182,6	152,1	222,1	106,1	63	228,4	161,9	11,4	183,9
11	95	143,9	224,1	36,1	85,8	66,9	67,5	234,8	92,5	70	0	211,1	149	259,4	176,1	127,1	249,8	166,5	77,5	200,3
12	198	192,7	55,4	224,2	172,3	153,2	180,4	220,7	133,3	182,6	211,1	0	91,9	54,2	190,3	141,9	48,9	71,2	186,7	35,8
13	109,4	196,8	136,2	174	86,7	82,4	148,6	257,8	56,9	152,1	149	91,9	0	145,6	210,1	145,3	112	22,9	159,8	63,6
14	252	216,1	49,6	268,6	226,2	205,1	220,6	222	186,4	222,1	259,4	54,2	145,6	0	205,3	171,4	60	124,6	224,7	85,2
15	245,3	34,5	155,8	154,8	220,8	180,9	109,1	58,9	188,9	106,1	176,1	190,3	210,1	205,3	0	66,6	238,4	208,9	99,5	210,2
16	182,5	53,6	124,3	117,1	156,8	117,2	63,6	120,8	123,1	63	127,1	141,9	145,3	171,4	66,6	0	190,6	146,1	61	154,3
17	221,1	241,4	92,5	266,7	198,4	187,3	226	265,8	164,2	228,4	249,8	48,9	112	60	238,4	190,6	0	90,3	233,2	50,3
18	131,3	199,1	118,9	188,9	108,2	100,7	158,5	253	76,2	161,9	166,5	71,2	22,9	124,6	208,9	146,1	90,3	0	168,7	41,5
19	156,2	67,3	180,6	58,2	136,1	99,1	13,2	157,8	116,9	11,4	77,5	186,7	159,8	224,7	99,5	61	233,2	168,7	0	189,6
20	172,7	207,2	90,3	218,7	149	137,1	181,2	246,7	114	183,9	200,3	35,8	63,6	85,2	210,2	154,3	50,3	41,5	189,6	0



Şekil 3.33. 20 düğümlü ağda düğümlerin görünüşleri

- 3) Ağdaki master düğümü belirleyebilmek için Tablo 3.14'deki mesafelerin satır veya sütun toplamı yapılarak Tablo 3.15'te aktarılmıştır.
- 4) Tablo 3.15'in minimum değerli elemanına sahip olan 9 numaralı düğüm ağın master düğümü olarak belirlenir.

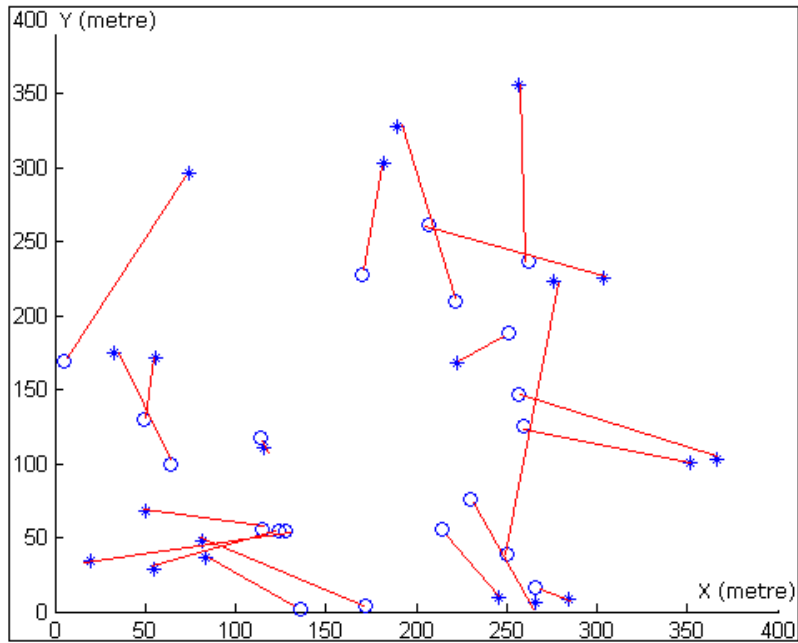
Tablo 3.15. 20 düğümlü ağda toplam mesafe değerleri

Düğüm No	Toplam Mesafe
1	3.103
2	2.823
3	2.935
4	2.843
5	2.699
6	2.304
7	2.309
8	3.911
9	2.297
10	2.333
11	2.733
12	2.745
13	2.501
14	3.399
15	3.061
16	2.326
17	3.365
18	2.537
19	2.398
20	2.732

- 5) Elde edilen P, M ve düğümlerin kapsama alanı olarak kabul edilen 100 metre bilgileri kullanılarak bulanık mantıkla düğümler arası maliyet değerleri belirlenmiştir. Bu amaçla öncelikli olarak, birbirlerinin kapsama alanında olmayan düğümlerin birbirlerine paket göndermeleri mümkün olmayacağı için, G bulanık mantık matrisinin ilgili elemanları sonsuz (inf) olarak işaretlenmiştir. Bulunan G matrisinin içeriği Tablo 3.16’da verilmiştir.
- 6) Benzetim modeli 100 saniye süre ile çalıştırılmıştır. Bu süre boyunca 0-20 m/s rastgele değişken hızlarla, rastgele yönlerde hareket eden düğümlerin eski ve yeni konumları Şekil 3.34’de görülmektedir. Düğümlerin ilk konumları “o” ile yeni konumları “*” ile gösterilmektedir.

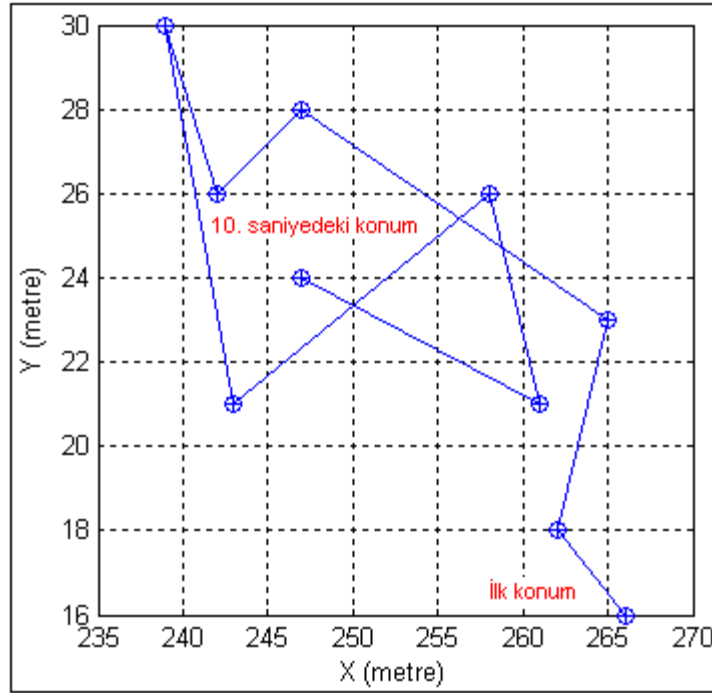
Tablo 3.16. 20 düğümlü ağın bulanık mantık sonucu elde edilen düğümler arası maliyet değerleri

Düğüm	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	Inf	Inf	Inf	32	80	Inf	Inf	80	Inf	67	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
2	Inf	0	Inf	Inf	Inf	Inf	75	78	Inf	76	Inf	Inf	Inf	Inf	44	57	Inf	Inf	65	Inf
3	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	58	Inf	49	Inf	Inf	80	Inf	Inf	87
4	Inf	Inf	Inf	0	Inf	83	56	Inf	Inf	62	38	Inf	Inf	Inf	Inf	Inf	Inf	Inf	60	Inf
5	66	Inf	Inf	Inf	0	69	Inf	Inf	73	Inf	64	Inf	78	Inf	Inf	Inf	Inf	Inf	Inf	Inf
6	69	Inf	Inf	83	39	0	78	Inf	73	78	64	Inf	76	Inf	Inf	Inf	Inf	Inf	73	Inf
7	Inf	75	Inf	74	Inf	83	0	Inf	Inf	62	64	Inf	Inf	Inf	Inf	63	Inf	Inf	24	Inf
8	Inf	81	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	60	Inf	Inf	Inf	Inf	Inf
9	71	Inf	Inf	Inf	41	69	Inf	Inf	0	Inf	65	Inf	75	Inf	Inf	Inf	Inf	75	Inf	Inf
10	Inf	75	Inf	74	Inf	83	45	Inf	Inf	0	64	Inf	Inf	Inf	Inf	63	Inf	Inf	24	Inf
11	82	Inf	Inf	68	62	80	65	Inf	88	69	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	66	Inf
12	Inf	Inf	56	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	81	56	Inf	Inf	48	70	Inf	72
13	Inf	Inf	Inf	Inf	63	83	Inf	Inf	74	Inf	Inf	75	0	Inf	Inf	Inf	Inf	50	Inf	79
14	Inf	Inf	49	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	58	Inf	0	Inf	Inf	61	Inf	Inf	87
15	Inf	60	Inf	Inf	Inf	Inf	Inf	62	Inf	Inf	Inf	Inf	Inf	Inf	0	65	Inf	Inf	74	Inf
16	Inf	75	Inf	Inf	Inf	Inf	63	Inf	Inf	65	Inf	Inf	Inf	Inf	65	0	Inf	Inf	62	Inf
17	Inf	Inf	66	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	56	Inf	61	Inf	Inf	0	80	Inf	75
18	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	88	Inf	Inf	69	55	Inf	Inf	Inf	79	0	Inf	72
19	Inf	75	Inf	75	Inf	89	45	Inf	Inf	62	64	Inf	Inf	Inf	81	63	Inf	Inf	0	Inf
20	Inf	Inf	65	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	44	75	76	Inf	Inf	51	50	Inf	0



Şekil 3.34. 20 düğümlü ağda düğümlerin 100 saniye hareket etmiş görünümleri

Şekil 3.35'te ise hareketlerin daha iyi görülebilmesi için, ağdaki birinci düğümün ilk 10 saniye boyunca yaptığı hareket sırasında izlediği yol verilmiştir.



Şekil 3.35. 20 düğümlü ağda 1. düğümün 10 saniye boyunca izlediği yol

- 7) Örnek ağdaki düğümlerin gönderdikleri, aldıkları ve üzerlerinden ilettikleri paketlerin sayısı belirlenmiştir. Tablo 3.17’de verilen değerlerden “gonderilen” sütunu, bir düğümün gönderdiği veri paketi sayısını, “alınan” sütunu, bir düğümün aldığı (kendisine gönderilmiş olan) toplam veri paketi sayısını ve “iletilen” sütunu, bir düğümün yönlendirdiği toplam veri paketi sayısını ifade etmektedir.
- 8) Her bir düğümün gönderdiği ve aldığı paket sayıları kullanılarak, enerji tüketimleri belirlenmiştir. Tablo 3.17’de her bir düğümün paket gönderme, paket alma ve paket iletme için harcadığı toplam güç değerleri verilmiştir. Ayrıca her bir düğümün tam dolu halde 100 Joule enerjiye sahip olduğu varsayılarak, kalan batarya yüzdeleri verilmiştir. Batarya kullanımı ile ilgili hesaplama işlemlerinde Ek D’de verilen Denklem D.5 ve D.6 kullanılmıştır.

Tablo 3.17. 20 düğümlü ağı paket sayıları ve güç tüketimleri

Düğüm No	Bat. Ömrü %	Gönd. Paket Sayısı	Alınan Paket Sayısı	İletilen Paket Sayısı	Gönd. Gücü (Joule)	Alma Gücü (Joule)	İletme Gücü (Joule)	Toplam Güç (Joule)	Kalan Bat. (%)
1	50	205	146	0	0,8405	0,029	0,000	0,870	49,13
2	90	247	131	59	1,0127	0,026	0,254	1,293	88,707
3	83	192	145	0	0,7872	0,029	0,000	0,816	82,184
4	65	291	69	0	1,1931	0,014	0,000	1,207	63,793
5	82	222	110	57	0,9102	0,022	0,245	1,177	80,823
6	67	185	102	1157	0,7585	0,020	4,975	5,754	61,246
7	35	163	116	178	0,6683	0,023	0,765	1,457	33,543
8	29	248	96	0	1,0168	0,019	0,000	1,036	27,964
9	35	247	94	58	1,0127	0,019	0,249	1,281	33,719
10	54	170	311	0	0,6970	0,062	0,000	0,759	53,241
11	73	233	243	712	0,9553	0,049	3,062	4,066	68,934
12	31	217	287	779	0,8897	0,057	3,350	4,297	26,703
13	84	244	258	1370	1,0004	0,052	5,891	6,943	77,057
14	57	215	208	0	0,8815	0,042	0,000	0,923	56,077
15	38	217	309	455	0,8897	0,062	1,957	2,908	35,092
16	71	218	349	0	0,8938	0,070	0,000	0,964	70,036
17	55	244	393	0	1,0004	0,079	0,000	1,079	53,921
18	45	188	299	255	0,7708	0,060	1,097	1,927	43,073
19	70	217	378	773	0,8897	0,076	3,324	4,289	65,711
20	63	199	318	0	0,8159	0,064	0,000	0,880	62,12

Düğümlerin enerjilerinde değişim olduktan sonra, kaynaktan hedefe giden en kısa yollarda da değişimler olmaktadır. Tablo 3.18’de başlangıç batarya değerinde ve değişmiş batarya değeri durumlarında kullanılan en kısa yolların bir kısmı görülmektedir.

Tablo 3.18. Batarya azalması durumunda değişen en kısa yollar

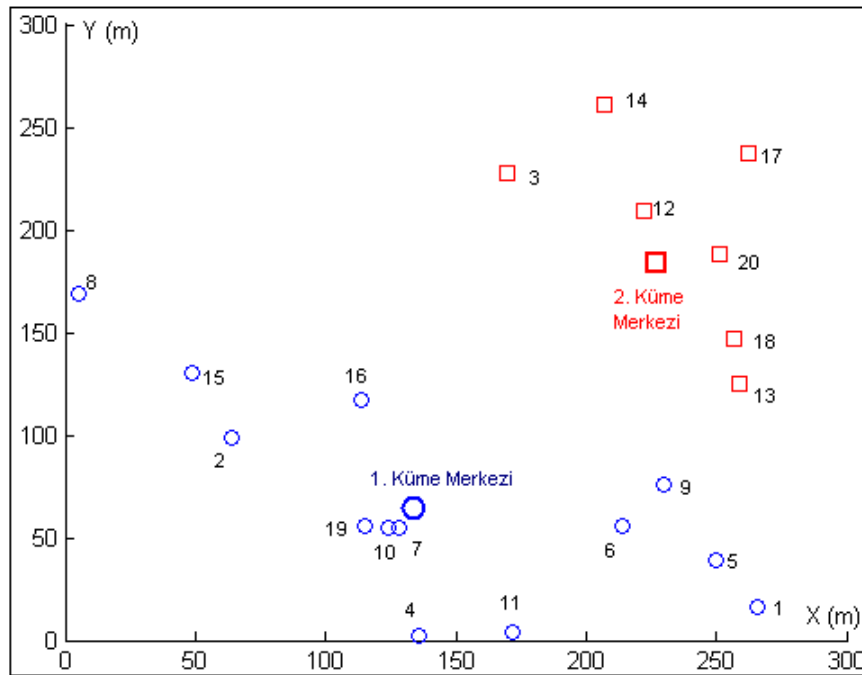
Başlangıç					100 s sonrası					Başlangıç					100 s sonrası								
2	19	11	5		2	19	6	5		16	7	6	13	12	14	16	19	6	13	12	14		
8	15	19	11	5	8	15	19	6	5	16	7	6	13	12	17	16	19	6	13	12	17		
15	19	11	5		15	19	6	5		16	7	6	13	18		16	19	6	13	18			
16	7	4			16	19	4			16	7	6	13	20		16	19	6	13	20			
16	7	6	13	12	3	16	19	6	13	12	3	16	19	11	1	16	7	11	1				
16	7	6	5		16	19	6	5		16	19	11	9		16	7	11	9					
16	7	6			16	19	6			16	19	11			16	7	11						
16	7	6	13	12	16	19	6	13	12	17	18	13	5		17	12	13	5					
16	7	6	13		16	19	6	13		17	18	13	6		17	12	13	6					
16	7	6	13	12	14	16	19	6	13	12	14	17	18	13	6	19	15	17	12	13	6	19	15

Tablo 3.18’deki her bir satırdaki sarı renkle dolu hücreler kaynak düğümü, mavi renkle dolu hücreler hedef düğümü göstermektedir.

Tablo 3.18'den görüldüğü gibi, bataryası başlangıç durumuna göre azalmış olan düğüm yerine, yoğunluk durumuna göre, yönlendirme yolunda, daha iyi batarya değerine sahip bir başka düğüm kullanılabilir.

Ayrıca ağdaki başlangıçta master düğüm olan 9 numaralı düğüm, 13. saniyeden itibaren master düğüm rolünü 2 numaralı düğüme devretmektedir. 100 saniyelik benzetim sürecinin çoğu boyunca master düğüm olarak kalan 2 numaralı düğüm yaklaşık 1,2 Joule enerji tüketmektedir.

9) Ağdaki düğümlerin dağınıklığı arttığında veya en kısa yol üzerinde bulunan düğüm sayısı belli bir eşik değerini üzerine çıktığında, ağ kümelerine ayrılır. 20 düğümlü örnek ağ iki kümeye ayrılırsa Şekil 3.36'daki görünüm elde edilir. Her bir düğümün kümelerine aitlik dereceleri ise Tablo 3.19'da verilmiştir. 1. Küme merkezinin koordinatları (134,02;64,558) ve 2. Küme merkezinin koordinatları (226,91;183,88) 'dir. 1. kümenin master düğümü 1. küme üyelik derecesi en büyük olan 3. düğüm ve 2. kümenin master düğümü 2. küme üyelik derecesi en büyük olan 20. düğümdür.



Şekil 3.36. 20 düğümlü ağın iki kümeye ayrılmış hali

Tablo 3.19. 20 düğümlü ağın küme aitlik dereceleri

Düğüm	1	2	3	4	5	6	7	8	9	10
1.Küme	0,6	0,85	0,16	0,91	0,6	0,72	1	0,64	0,55	0,99
2.Küme	0,4	0,15	0,84	0,09	0,4	0,28	0,01	0,36	0,45	0,01
Düğüm	11	12	13	14	15	16	17	18	19	20
1.Küme	0,87	0,02	0,19	0,13	0,75	0,85	0,08	0,09	0,99	0,01
2.Küme	0,13	0,98	0,81	0,87	0,25	0,15	0,92	0,91	0,01	0,98

Denklem 3.8 kullanılarak, kümeler arası paket iletimi gerçekleştirecek düğüm belirlenir. Örnek ağ için bu düğüm 9 numaralı düğümdür.

3.8. Sonuç

Bu bölümde, KTMYA algoritmasını oluşturan çalışma adımları ve ağda paket dağıtım mekanizmaları verilmiştir. Algoritmanın yönlendirme kararı verirken kullandığı bulanık mantık yöntemi ve klasik yöntem üzerine sağladığı üstünlükler sunulmuştur. Ayrıca, ağdaki düğüm sayısının ve düğümlerin dağınıklığının artması durumunda, FCM ile kümeleme çözümü sunulmuştur.

BÖLÜM 4. KTMYA’NIN PERFORMANS DEĞERLENDİRMESİ

Bu bölümde, öncelikle MANET’te kullanılan yönlendirme algoritmalarının performans değerlendirme ölçütleri üzerinde durulmaktadır. Ardından algoritmanın performans değerlendirmesini yapabilmek amacıyla MATLAB 7.01 ortamında geliştirilen yazılım sunularak, karşılaştırmalı olarak, geliştirilen yeni yöntemin olumlu ve olumsuz yanları vurgulanmaktadır.

4.1. MANET’te Performans Değerlendirme Kriterleri

Internet Engineering Task Force (IETF), MANET protokollerini oluşturmaktan sorumlu olan organizasyondur. IETF organizasyonu altında alt çalışma grubu olan MANET, kablosuz tasarsız ağların performans değerlendirmesi için gerekli ölçütleri RFC 2501 (Request For Comments) belgesinde yayınlamıştır [2]. Bu belgeye göre performans kriterleri ölçülebilir ve nitel kriterler olarak iki gruba ayrılmıştır.

4.1.1. Ölçülebilir performans kriterleri

Performans değerlendirmesi için IETF tarafından önerilen ölçülebilir kriterler şunlardır [2]:

1. Üretilen yönlendirme trafiği veya yönlendirme ek yükü: Ağda iletilen veri paketi başına düşen kontrol paketi sayısıdır. Yönlendirme yükü olarak da adlandırılan bu ölçüm kriterinin düşük değerde olması, algoritmanın performansı açısından önemlidir.
2. Paket dağıtım oranı: Throughput (birim zamanda bir kanaldan dağıtılacak veri miktarı), ağda bir düğümden çıkan paketlerin hedefine ulaşma oranıdır. Yüksek paket dağıtım oranı, algoritmanın performansının yüksek olduğu anlamına gelir.

3. Paket kayıp oranı: Hedefe ulaşmayan paketlerin oranıdır. Paket kayıp oranı düşükse, algoritmanın performansı yüksektir.
4. Veri paketlerinin uçtan uca ortalama gecikmesi: Bir düğümden çıkan veri paketinin hedefe ulaşması için geçen süredir. Algoritmanın performansı açısından bu değerin düşük olması gereklidir.
5. Yol bulma hızı: Bir hedefe ait yolun bulunma süresidir. Bu kriter sadece isteğe bağlı algoritmalarda kullanılır. Yol bulma hızının yüksek olması olumlu yönde performans katkısı sağlar.
6. Enerji verimliliği: Tasarsız ağlardaki birçok düğüm sınırlı batarya ömrüne sahiptir. Bu yüzden enerjilerini en iyi şekilde kullanmaları gerekir. Protokol, mümkün olduğunca en az işlem ve iletme gereksinim duymalıdır.
7. Verimlilik: Performans değerlendirmesi için üç tür verimlilik hesabı yapılabilir. Bunlar:
 - a. Gönderilen veri bitlerinin dağıtılan veri bitlerine oranı: Ağdaki bit verimliliğini ve veri paketlerinin hedefe ulaşmak için geçtikleri ortalama hop sayısını verir.
 - b. Gönderilen kontrol bitlerinin veri bitlerine oranı: Protokolün bit verimliliğini verir. Aynı zamanda yönlendirme ek yükünün hesabını sağlar.
 - c. Gönderilen kontrol ve veri paketlerinin dağıtılan veri paketlerine oranı: Protokolün kanal erişim verimliliğinin hesabını sağlar. Aynı zamanda ikinci katman trafiği hakkında bilgi verir.

4.1.2. Niteliksel performans kriterleri

Aşağıdaki performans kriterleri, ölçülemeyen veya nitel kriterlerdir:

1. En uygun yol bulunması: Protokol, seçilen optimalite ölçütlerine dayalı olarak en uygun yolu bulabilmelidir. Optimalite metrikleri hop sayısı, gecikme, bant genişliği, yük veya güvenilirlik olabilir.
2. Yük dengeleme (load balancing): Protokol bir düğüme aşırı yüklenmemeli ve tüm düğümlere yükü dağıtacak şekilde tasarlanmalıdır. Bu, belli bazı düğümlerde meydana gelebilecek çakışmalardan kaçınmaya yardımcı olur.

3. Ölçeklenebilirlik: Protokolün performansı, ağdaki düğümlerin sayısının artmasından veya azalmasından etkilenmemelidir.
4. Bant genişliği verimliliği: Kablosuz ağlarda her bir düğüm sınırlı bant genişliğine sahiptir. Bu yüzden protokol mümkün olduğunca en az trafik üretmeyi amaçlamalıdır. Bu da periyodik güncellemelerin azaltılmasıyla olur.
5. Yakınsama: Ağ topolojisinde bir değişiklik olduğunda, protokol oldukça küçük bir zamanda kararlı bir duruma gelmelidir.
6. Döngü oluşmama (loop free): Bazı paketler hedeflerine ulaşamayıp ağ içinde kaldıkları zaman döngüler oluşur. İyi bir yönlendirme protokolü, döngü oluşmadığından emin olmalıdır. Çünkü döngüler bant genişliğinin çoğunu tüketirler ve döngüdeki paketler asla hedeflerine ulaşamazlar.
7. Yol bozulması ve düzeltilmesi: Yol bozulmalarını bulmak ve düzeltmek için bazı mekanizmalar geliştirilmelidir.
8. Eski yolları belirleme: Ağdaki düğümler gezgin olduğu için düğümlerin hareketinden dolayı yollar eskir. Protokol eski yolları belirleme için bir mekanizma tanımlamalıdır.
9. Düğümlerin hareketlilik hızı: Protokol düğüm hareketlerinin tüm hız ve türlerini iyi bir şekilde desteklemelidir.
10. Bölünmüş ağlar: Düğümlerin serbest hareketinin bir sonucu olarak, birkaç düğüm geriye kalan düğümlerden, iletim bölgesinin dışında kalarak izole olabilir. Farklı düğüm grupları oluşur ve bu senaryoya bölünmüş düğümler denir. Yönlendirme protokolü bu problemi anlayabilmeli ve bazı uygun çözüm önerilerinde bulunabilmelidir.
11. Çoklu yol: Bir yönlendirme protokolü, özel bir yoldaki çakışmayı azaltmak için bir tek kaynak-hedef çifti için çoklu yolu mümkün hale getirebilmelidir. Çoklu yollar gezgin ağlar için daha kullanışlıdır. Çünkü bu ağlarda sık sık düğüm hareketleri olduğu için bir çok link hatası meydana gelir. Eğer alternatif bir yol varsa, gecikmeyi azaltıp paket dağıtım oranını artırır.
12. Uyuma özelliği (sleep function): Tasarsız kablosuz ağlarda düğümlerin enerjileri kısıtlı olduğundan, bazı düğümler rastgele bir zaman periyodunca uyku moduna (pasif moda) geçmeye karar verebilirler. Protokol bu durumda olan düğümlerden ağdaki diğer düğümlerin etkilenmesini önleyebilmelidir.

4.1.3. Ağ performansını etkileyen değişken parametreler

Aşağıdaki parametreler yönlendirme protokol performansını değerlendirirken değiştirilebilir [2]:

1. Ağın büyüklüğü (düğüm sayısı): Ağdaki farklı düğüm sayıları için performans değerlendirmesi yapılabilir.
2. Gezginlik: Ağdaki düğümler değişken hızlarda hareket edebilmelidir.
3. Trafik deseni: Ağdaki düğümler düzgün dağıtılmamış (non-uniform) olmalıdır veya düğümleri düzgün dağıtılmamış ağlar oluşturulabilir.
4. Ağ bağlantırlılığı (bir düğümün ortalama derecesi) : Bir tek düğüme bağlı olan ortalama düğüm sayısı veya bir düğümün kapsama alanı içinde bulunan komşu düğüm sayısıdır.
5. Link kapasitesi: Bit/saniye olarak ölçülen link hızıdır. Değişik link hızları için performans ölçümü ve değerlendirmesi yapılabilir.
6. Tek yönlü linklerin oranı: Algoritma içinde kullanılan tek yönlü linkler değiştirilerek performans ölçümü gerçekleştirilebilir.
7. Utku modundaki düğümlerin oranı

4.2. Konum Tabanlı Melez Yönlendirme Algoritmasında Performans Değerlendirme İşlem Adımları

Bu tez çalışmasında geliştirilen KTMYA'nın performans değerlendirmesini yapabilmek amacıyla, MATLAB 7.01'de yazılım geliştirilmiştir. Geliştirilen yazılımda iş ve süreçleri gerçekleştirmek için fonksiyonlar yazılmış ve modüler bir yazılım ortamı oluşturulmuştur.

Performans değerlendirmesinde, Bölüm 4.1'de verilen performans ölçüm kriterleri kullanılmış ve RFC 2501'de [2] öngörülen değerlendirme ortam ve parametreleri kullanılmıştır.

4.2.1. Performans değerlendirme ortamı

Yazılımda ağ ortamı olarak yatay, düşey ve yükseklik boyutlarında rastgele dağılmış

düğümler oluşturmaktadır. Oluşturulacak ağın düğüm sayısı ve düğümlerin dağıtılacağı ortamın büyüklüğü değiştirilebilmektedir. Bu sayede Bölüm 4.1.3’de verilen değiştirilebilir ortam parametrelerinden, düğüm sayısı (ağın büyüklüğü) ve trafik deseni kriterleri sağlanmış olmaktadır. Düğümler rastgele dağıtılmış olduğu için homojen olmayan (non-uniform) bir dağılım gerçekleşmektedir.

Benzetim yazılımında bu adımı gerçekleştirmek için Ek F’de kodları verilen *node.m* fonksiyonu oluşturulmuştur. Fonksiyonun giriş parametreleri düğüm sayısı ile x ve y eksenlerinin yani ortamın büyüklüğüdür. Fonksiyonun çıkış parametresi ise denklem 3.1’de verilen P konum bilgileri matrisidir. P matrisinin boyutu, n düğüm sayısı olmak üzere “ $n \times 6$ ”dır. Matrisin ilk üç sütunu düğüm konumlarını tutarken, dördüncü ve beşinci sütunlar sırasıyla düğümlerin yüzde cinsinden batarya ömürleri ve tampon (buffer) değerleridir. Altıncı sütun olan *id* değeri bu aşamada 0 olarak atanır. *id* alanı, ağdaki düğümlerden güncelleme paketleri alındıkça arttırılan bir değer içerir.

4.2.2. Düğümler arası mesafeler

KTMYA’nın Bölüm 3’te verilen adımlarına göre, “düğümler arası mesafe” yönlendirme kararının verilebilmesi için gerekli olan parametrelerden biridir. Bu amaçla düğümler arası mesafeler Denklem 3.2 kullanılarak, yazılım tarafından hesaplanabilmektedir. Bu işlem için, kodları Ek F’de verilen *mesafe.m* fonksiyonu kullanılmaktadır.

mesafe.m fonksiyonunun giriş parametresi P konum bilgileri matrisidir. Çıkış parametresi ise Denklem 3.3’te verilen M mesafe matrisidir.

4.2.3. Master düğümü belirleme

KTMYA’da düğüm yönlendirme bilgilerini ve diğer bilgileri tutmak için master düğüm kullanılır. Benzetim yazılımında master düğümü belirlemek için *master.m* fonksiyonu yazılmıştır. Kodları Ek F’de verilen fonksiyon giriş parametresi olarak M mesafe matrisini almaktadır. Çıkış parametresi olarak master düğümü ve matrisin

normunu vermektedir. Master düğüm adayını belirlemek için denklem 3.4 kullanılmaktadır.

4.2.4. Bulanık mantık ile düğümler arası maliyet değerlerinin belirlenmesi

Algoritmada düğümlerin graf yapısında oluşturulabilmesi için düğümler arası maliyetin hesaplanması gerekir. Maliyet hesabı bulanık mantık kullanılarak gerçekleştirilir. Bu amaçla MATLAB 7.01'in Fuzzy Logic Toolbox kullanılarak *PBHRA.fis* bulanık mantık çıkarım mekanizması oluşturulmuştur. *PBHRA.fis*'i de kullanarak maliyet hesabı yapan *fuzzy.m* fonksiyonu yazılmıştır.

Fonksiyonun giriş parametreleri denklem 3.1'de verilen P konum matrisi, denklem 3.3'de verilen M mesafe matrisi ve yayılım mesafesi (düğüm kapsama alanı)'dır. Çıkış parametresi ise denklem 4.6'da verilen G düğümler arası maliyet matrisidir. *Fuzzy.m* fonksiyonunun kodları Ek F'de verilmiştir.

Algoritmada bulanık mantığın etkisini gösterebilmek için, bulanık mantık kullanılmadan da maliyet değerleri hesaplanmıştır. Bu işlem için *fuzzyless.m* fonksiyonu oluşturulmuştur. Giriş ve çıkış parametreleri *fuzzy.m* fonksiyonu ile aynıdır. *Fuzzyless.m* fonksiyonunun kodları Ek F'de verilmiştir.

4.2.5. Düğümler arası en kısa yol (en maliyetsiz yol) belirleme işlemi

Algoritmada veri gönderici düğüm, master düğümden yol bilgisi sorgusunda bulunduğu master düğüm optimizasyon yaparak yol bilgisi cevabı gönderir. Dijkstra algoritması kullanılarak bulanık mantık ile elde edilmiş maliyet değerleri minimum olacak şekilde, düğümler arası yolların belirlenmesi işlemi gerçekleştirilebilmektedir. Bu amaçla *enkisayol.m* fonksiyonu yazılmış ve kodları Ek F'de verilmiştir.

Enkisayol.m fonksiyonunun giriş parametresi denklem 4.6'da verilen G maliyet matrisi ve çıkış parametresi v yol bilgisi matrisidir. En maliyetsiz yolun

hesaplanmasında Sergiy Iglin tarafından kodları yazılmış Dijkstra algoritması fonksiyonu olan *grShortPath.m* fonksiyonu da kullanılmıştır [51].

Fonksiyon çıkışında hesaplanan v en kısa yol matrisi bir satır matristir. Elemanları ise kaynaktan hedefe kadar olan düğümlerin numaralarıdır.

4.2.6. Düğümlerin paket üretimi

Her bir düğümün birim zamanda ürettiği paketlerin sayısı ve bu paketlerin ağda gönderileceği düğümler rastgele olarak belirlenebilmektedir. Bir düğüm eğer paket gönderecekse, kendisi dışındaki bir düğüme paket gönderebilecek şekilde hedef düğüm rastgele seçilmektedir. Düğümlerin paket üretim miktarları Poisson dağılımı ile bulunmaktadır.

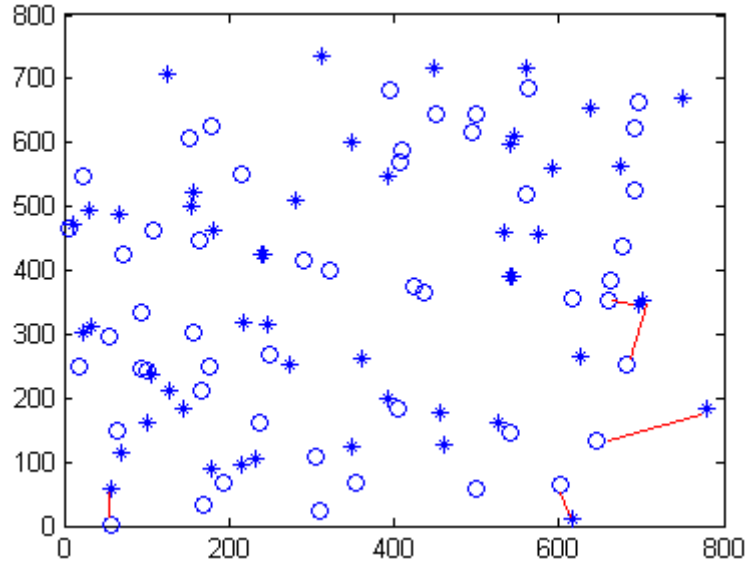
Paket üretim miktarları ve alıcılarının belirlenmesi için *poisson.m* fonksiyonu yazılmıştır. Kodları Ek F'de verilen fonksiyonun giriş parametresi ağın düğüm sayısıdır. Çıkış parametreleri ise üretilen paket sayısı ve paketlerin alıcısı olan hedef düğüm numaralarıdır.

4.2.7. Düğümlerin hareketliliği

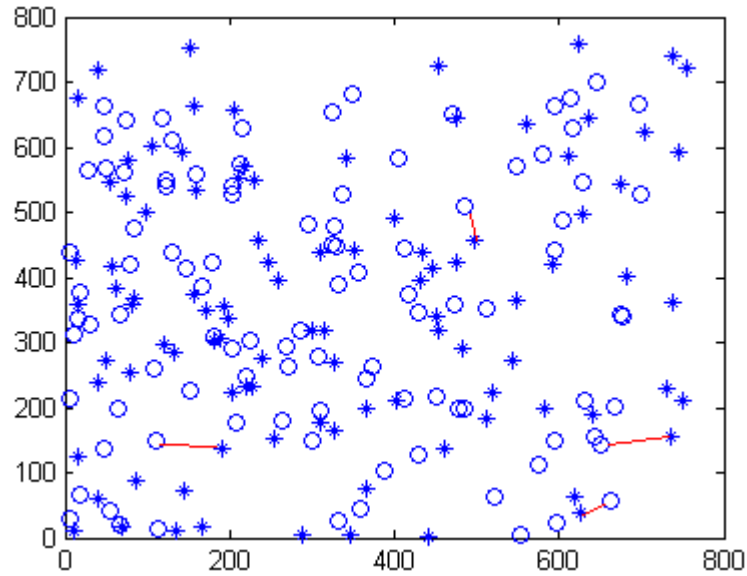
Bölüm 4.1.3'de verilen değiştirilebilir performans ölçüm kıstaslarından gezginliğin sağlanabilmesi için düğümler rastgele hızlarda hareket ettirilmişlerdir. 0'dan 20 m/s'ye kadar değişken hızlarda düğümlerin hareket etmelerini sağlayacak biçimde benzetim kodları yazılmıştır. *Move.m* fonksiyonu ile düğümlerin hareket etmeleri sağlanmıştır. En büyük hızın 20 m/s seçilmesinin sebebi, bir aracın şehir içi hızına yaklaşık olmasıdır.

Kodları Ek F'de verilen *move.m* fonksiyonunun giriş parametresi denklem 3.1'de verilen P konum bilgileri matrisi ve çıkış parametresi hareket etmiş haldeki düğümlerin konum bilgileri matrisidir. *Move.m* fonksiyonu kullanılarak herhangi bir düğümün, herhangi bir yönde ($0-360^0$) hareket etmesi sağlanmakta ve düğümün yeni konumuna ait koordinatları üretilebilmektedir.

Şekil 4.1’de 50 düğümlü bir ağın, Şekil 4.2’de 100 düğümlü bir ağın 0-20 m/s rastgele hızlarda ve rastgele bir yönde 100 saniye boyunca hareket etmiş halleri görülmektedir. Şekillerde “o” ile gösterilenler düğümlerin başlangıç konumlarını ve “*” ile gösterilenler 100 saniye sonundaki konumlarını ifade etmektedir.



Şekil 4.1. 50 düğümlü bir ağın başlangıç ve 100 saniye hareket etmiş konumları



Şekil 4.2. 100 düğümlü bir ağın başlangıç ve 100 saniye hareket etmiş konumları

4.2.8. Düğümlerin enerji tüketimleri

Geliştirilen algoritmada düğümlerin harcadıkları enerji miktarları

hesaplanabilmektedir. Enerji tüketimleri, Ek D’de verilen enerji tüketim ifadeleri kullanılarak bulunmaktadır. Düğümlerin aldıkları, gönderdikleri ve ilettikleri paket sayıları kullanılarak enerji tüketimlerinin hesaplandığı *guc.m* fonksiyonu kullanılmaktadır.

Ek F’de kodları verilen *guc.m* fonksiyonunun giriş parametreleri alınan, gönderilen ve iletilen paket sayıları ile gönderilen paket boyutu ve gönderme mesafesidir. Çıkış parametresi ise harcanan toplam enerjidir.

4.3. Performans Değerlendirme

Geliştirilen KTMYA Bölüm 4.1’de verilen değerlendirme kriterlerinden yönlendirme yükü, paket dağıtım oranı, paket kayıp oranı, uçtan uca ortalama paket gecikmesi ve enerji verimliliği açısından performans ölçümleri yapılmıştır. Performans ölçümlerinde Bölüm 4.1.3’de verilen değiştirilebilir performans ölçüm parametrelerine farklı değerler verilerek sonuçlar alınmıştır.

4.3.1. Benzetim parametreleri

Düğüm sayısı: Benzetim için 10, 20, 50 ve 100 düğümlü ağlar kullanılmıştır.

Ortam boyutu: Yatay ve düşeyde düğüm başına yaklaşık 5000 m² alan düşecek şekilde ve 10 metre yüksekliğe sahip bir ortamda rastgele dağılım kullanılmıştır.

Paket boyutu: 512 ve 1024 Byte sabit boyutlarda veri paketleri kullanılmıştır.

Kaynak Sayısı: 5, 10, 20 ve 30 kaynaklı ağlar ele alınmıştır. Kaynak sayısı, ağda aynı anda veri paketi gönderebilecek düğümleri belirtmektedir.

Hareketsizlik zamanı: 0, 10, 20, 50 ve 100 saniye hareketsizlik zamanları kullanılmıştır. Hareketsizlik zamanı, düğümün hareket etmeden, sabit kaldığı zamandır. 0 saniye, düğümün sürekli hareket halinde olduğu durumu, 100 saniye ise düğümün iki hareket arasında 100 saniye boyunca hareketsiz kaldığını ifade etmektedir.

Yayılmı mesafesi: Dügümlerin kapsama alanı 100 metre yarıçapında dairedir.

Benzetim süresi: Benzetim modelleri 100 saniye çalıştırılmaktadır.

Tekrar sayısı: Benzetim yazılımı ile ele alınan her farklı ağ on defa çalıştırılıp, ortalama sonuçlar alınmıştır.

4.3.2. Yönlendirme yükü

Yönlendirme yükü, bant genişliği verimliliği ve enerji verimliliği yönünden en önemli performans ölçütüdür. Ağda bir veri paketini hedefine ulaştırabilmek için gönderilmesi gereken kontrol paketleri sayısı olarak ifade edilir. Diğer bir ifadeyle veri paketi başına düşen yönlendirme paketi sayısıdır.

KTMYA, ağda değişik düğüm sayıları ve paket gönderici (kaynak) düğüm sayıları kullanarak benzetimi gerçekleştirilmiştir. Ağdaki düğümler 10, 20, 50 ve 100 olarak kullanılmıştır. Ağdaki kaynak sayıları ise 5, 10, 20 ve 30 olarak alınmıştır.

Yönlendirme yükünü hesaplamak için denklem 4.1'deki ifade kullanılmıştır.

$$YY = \frac{n_r}{n_T} \quad (4.1)$$

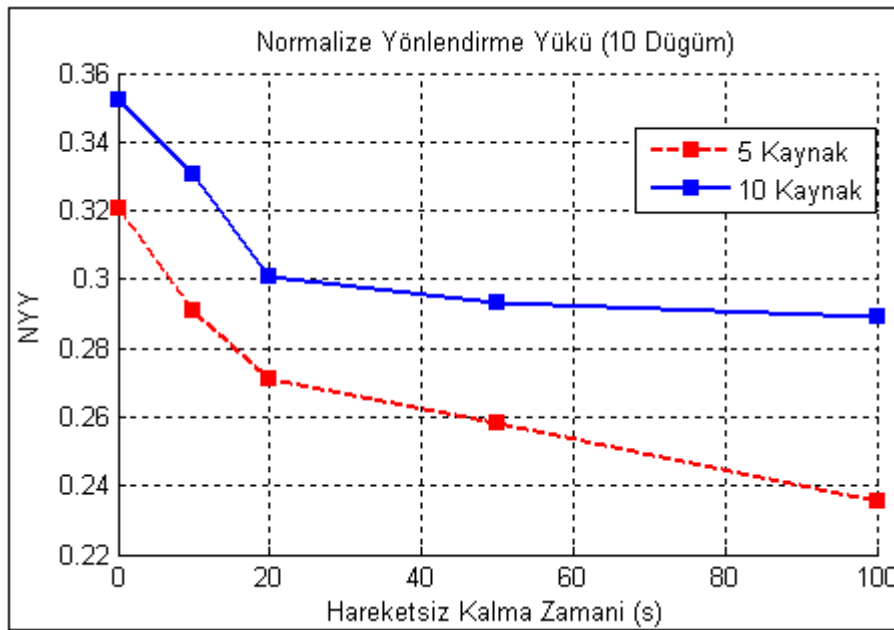
Bu ifadede n_r , yönlendirme paketleri sayısı, n_T , gönderilen toplam veri paketleri sayısıdır.

Geliştirilen benzetim programında yönlendirme yükünü hesaplamak için kodları Ek F'de verilen $nrl.m$ fonksiyonu yazılmıştır. 10 düğümlü ağ için $nrl10.m$, 20 düğümlü ağ için $nrl20.m$, 50 düğümlü ağ için $nrl50.m$ ve 100 düğümlü ağ için $nrl100.m$ fonksiyonları oluşturulmuştur. Benzetim programıyla elde edilen yönlendirme yükü sonuçları grafik olarak ifade edilmekte ve küsüratlı performans değerlerinin daha net görülebilmesi için tablo olarak da ifade edilmektedir.

10 düğümlü, 5 ve 10 kaynaklı ağ için benzetim sonucunda elde edilen yönlendirme yükü Tablo 4.1’de verilmiş ve Şekil 4.3’de grafik olarak gösterilmiştir.

Tablo 4.1. 10 düğümlü, 5 ve 10 kaynaklı ağın yönlendirme yükü değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
10 Düğüm-5 Kaynak	0,32062	0,29104	0,27102	0,25795	0,2357
10 Düğüm-10 Kaynak	0,35207	0,33053	0,30061	0,2933	0,28889



Şekil 4.3. 10 düğümlü, 5 ve 10 kaynaklı ağın yönlendirme yükü

Tablo 4.1’deki veya Şekil 4.3’deki değerlere göre 10 düğümlü bir ağda 5 kaynağın veri gönderdiği durum, 10 kaynağın veri gönderdiği duruma göre daha iyi yönlendirme yükü değerine sahiptir.

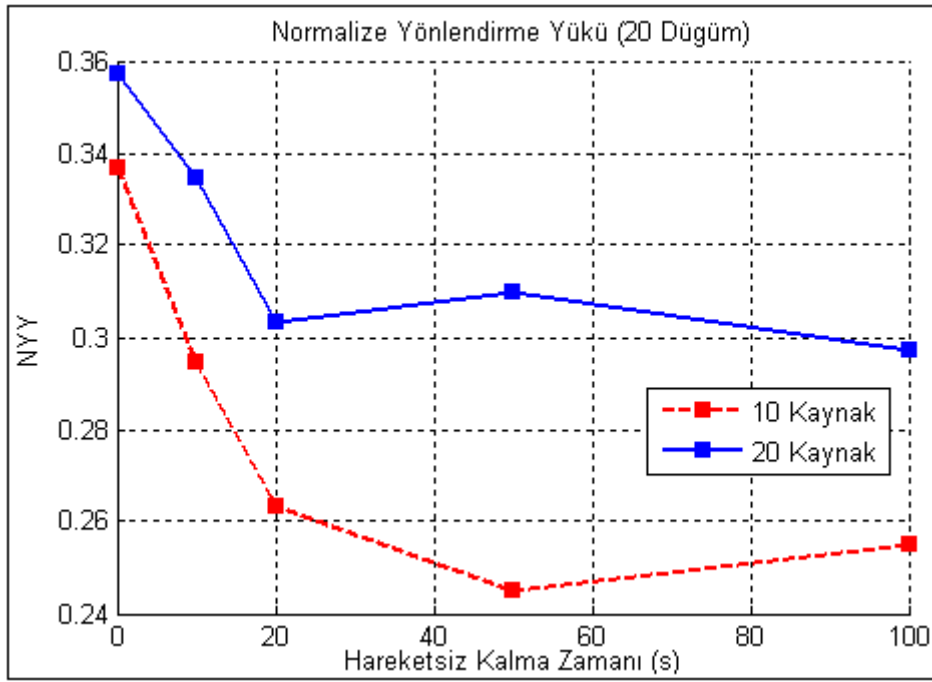
20 düğümlü, 10 ve 20 kaynaklı ağ için benzetim sonucunda elde edilen yönlendirme yükü Tablo 4.2’de verilmiş ve Şekil 4.4’de grafik olarak gösterilmiştir.

Tablo 4.2. 20 düğümlü, 10 ve 20 kaynaklı ağın yönlendirme yükü değerleri

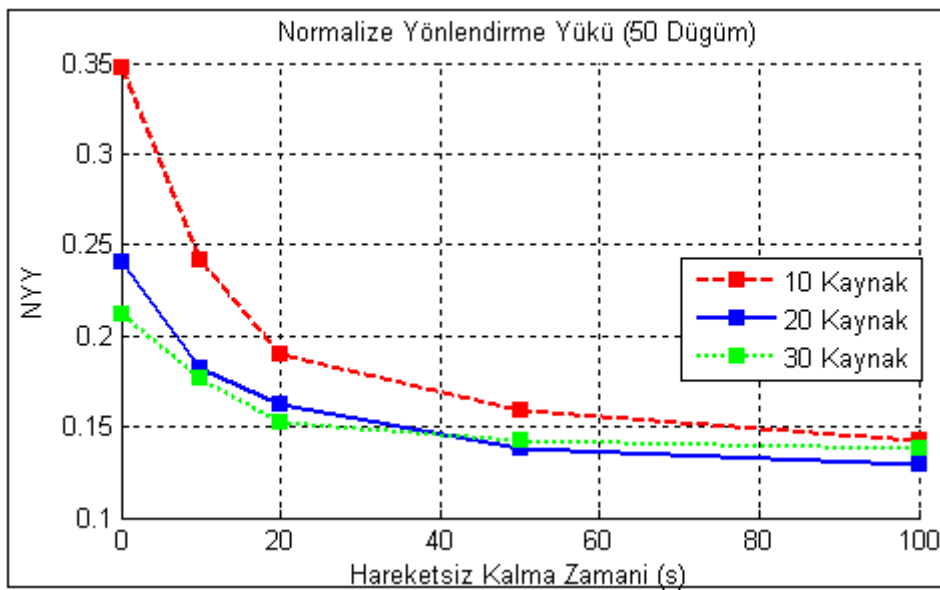
Hareketsizlik zamanı (s)	0	10	20	50	100
20 Düğüm-10 Kaynak	0,33659	0,29459	0,26316	0,24513	0,25496
20 Düğüm-20 Kaynak	0,35704	0,33475	0,30337	0,30967	0,29706

Tablo 4.2'deki veya Şekil 4.4'deki değerlere göre 20 düğümlü bir ağda, 10 kaynaklı durum, 20 kaynaklı duruma göre daha iyi yönlendirme yükü değerine sahiptir.

50 düğümlü, 10, 20 ve 30 kaynaklı ağ için benzetim sonucunda elde edilen yönlendirme yükü Tablo 4.3'de verilmiş ve Şekil 4.5'de grafik olarak gösterilmiştir.



Şekil 4.4. 20 düğümlü, 10 ve 20 kaynaklı ağın yönlendirme yükü



Şekil 4.5. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü

Tablo 4.3. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü değerleri

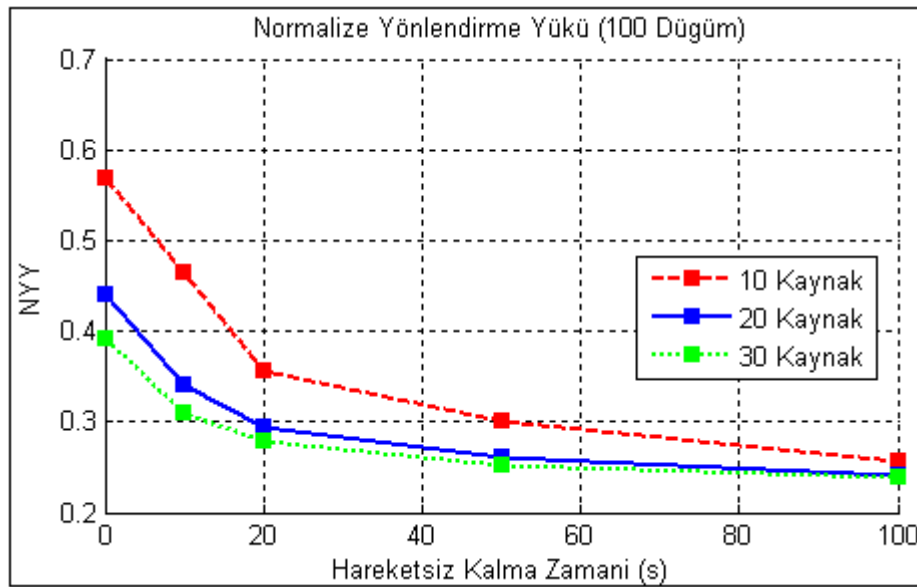
Hareketsizlik zamanı (s)	0	10	20	50	100
50 Düğüm-10 Kaynak	0,34729	0,24106	0,18958	0,15848	0,14292
50 Düğüm-20 Kaynak	0,24075	0,18157	0,16203	0,1385	0,12903
50 Düğüm-30 Kaynak	0,21231	0,17666	0,15236	0,14248	0,13818

Tablo 4.3'deki veya Şekil 4.5'deki değerlere göre 50 düğümlü ağda, 30 kaynaklı durumda ve yüksek hareketlilik halinde daha iyi yönlendirme yükü değerlerine sahiptir.

100 düğümlü, 10, 20 ve 30 kaynaklı ağ için benzetim sonucunda elde edilen yönlendirme yükü Tablo 4.4'de verilmiş ve Şekil 4.6'da grafik olarak gösterilmiştir.

Tablo 4.4. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
100 Düğüm-10 Kaynak	0,5685	0,46341	0,35674	0,30115	0,25547
100 Düğüm-20 Kaynak	0,44066	0,33952	0,29364	0,26131	0,24
100 Düğüm-30 Kaynak	0,3923	0,30951	0,27837	0,25099	0,23836



Şekil 4.6. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın yönlendirme yükü

Tablo 4.4'deki veya Şekil 4.6'daki değerlere göre 100 düğümlü ağda, 30 kaynaklı durumda ve yüksek hareketlilik halinde daha iyi yönlendirme değerlerine sahiptir.

Yukarıdaki yönlendirme yükü değerlerinden görüldüğü gibi, KTMYA, yüksek düğüm sayısına sahip ağlar için çok kaynaklı durumunda, düşük düğüm sayısına sahip ağlar için az kaynaklı durumda daha iyi yönlendirme yükü değerleri vermektedir.

4.3.3. Paket dağıtım oranı

Paket dağıtım oranı, ağdaki veri gönderici düğümlerin gönderdikleri paketlerin, alıcılarına ulaşma oranlarını belirleyen bir performans ölçüm kriteridir. Yüzde cinsinden hedefine ulaşan paketlerin oranı olarak ifade edilir. Ağda gönderici düğümden çıkan paketler hedeflerine ulaşamamışlarsa, paket dağıtım oranı düşüktür. Hedefine ulaşan paketlerin sayısı artarsa, paket dağıtım oranı da artar.

Paket dağıtım oranını belirlemek için denklem 4.2'deki ifade kullanılmıştır.

$$PDO = \frac{n_h}{n_T} \cdot 100 \quad (4.2)$$

Bu ifadede n_h , hedefine ulaşan paketlerin sayısı ve n_T , gönderilen toplam veri paketleri sayısıdır.

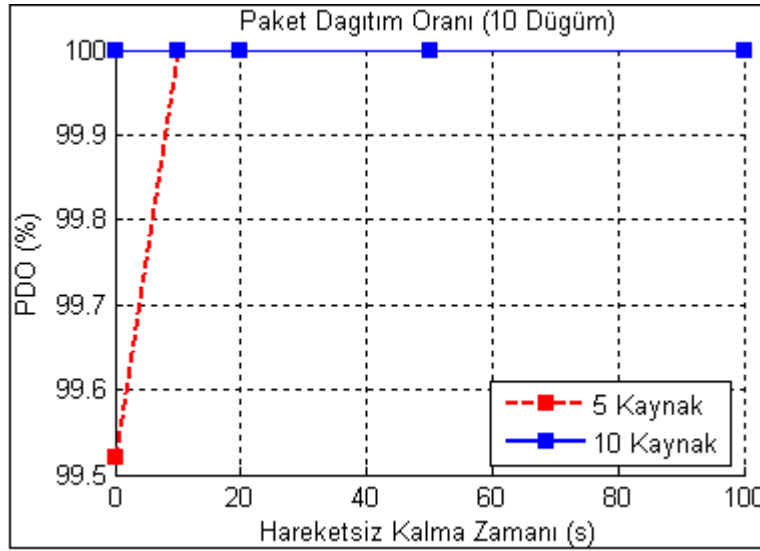
Geliştirilen benzetim programında paket dağıtım oranını hesaplamak için kodları Ek F'de verilen *padf.m* fonksiyonu yazılmıştır. 10 düğümlü ağ için *padf10.m*, 20 düğümlü ağ için *padf20.m*, 50 düğümlü ağ için *padf50.m* ve 100 düğümlü ağ için *padf100.m* fonksiyonları oluşturulmuştur.

10 düğümlü, 5 ve 10 kaynaklı ağ için paket dağıtım oranları Tablo 4.5'te verilmiş ve Şekil 4.7'de grafik olarak gösterilmiştir.

Tablo 4.5. 10 düğümlü, 5 ve 10 kaynaklı ağın paket dağıtım oranları

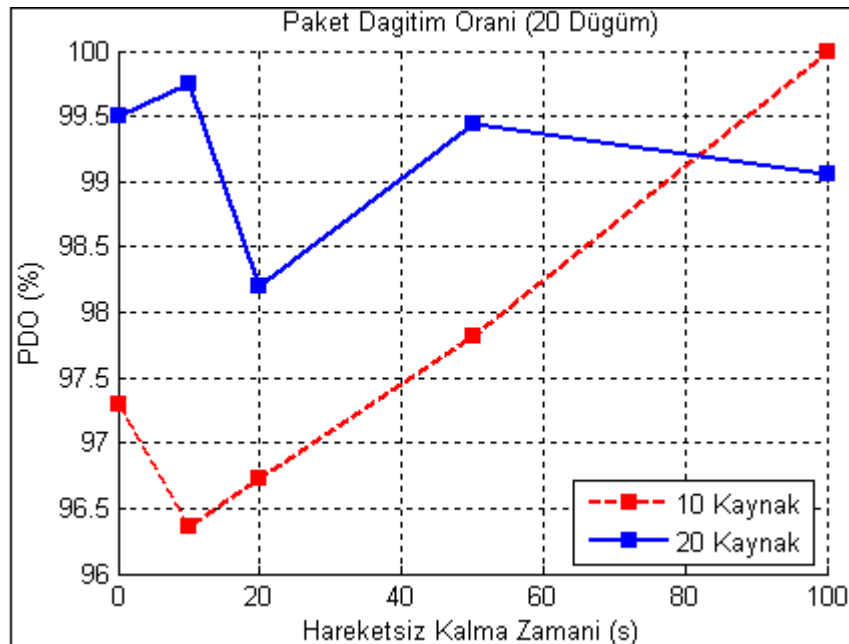
Hareketsizlik zamanı (s)	0	10	20	50	100
10 Düğüm-5 Kaynak	99,52	100	100	100	100
10 Düğüm-10 Kaynak	100	100	100	100	100

Tablo 4.5’de verilen değerlere göre, 10 düğümlü ağ, 10 kaynaklı durumda daha iyi paket dağıtım oranı sağlamaktadır.



Şekil 4.7. 10 düğümlü, 5 ve 10 kaynaklı ağın paket dağıtım oranları

20 düğümlü, 10 ve 20 kaynaklı ağ için paket dağıtım oranları Tablo 4.6’de verilmiş ve Şekil 4.8’de grafik olarak gösterilmiştir.



Şekil 4.8. 20 düğümlü, 10 ve 20 kaynaklı ağın paket dağıtım oranları

Tablo 4.6. 20 düğümlü, 10 ve 20 kaynaklı ağın paket dağıtım oranları

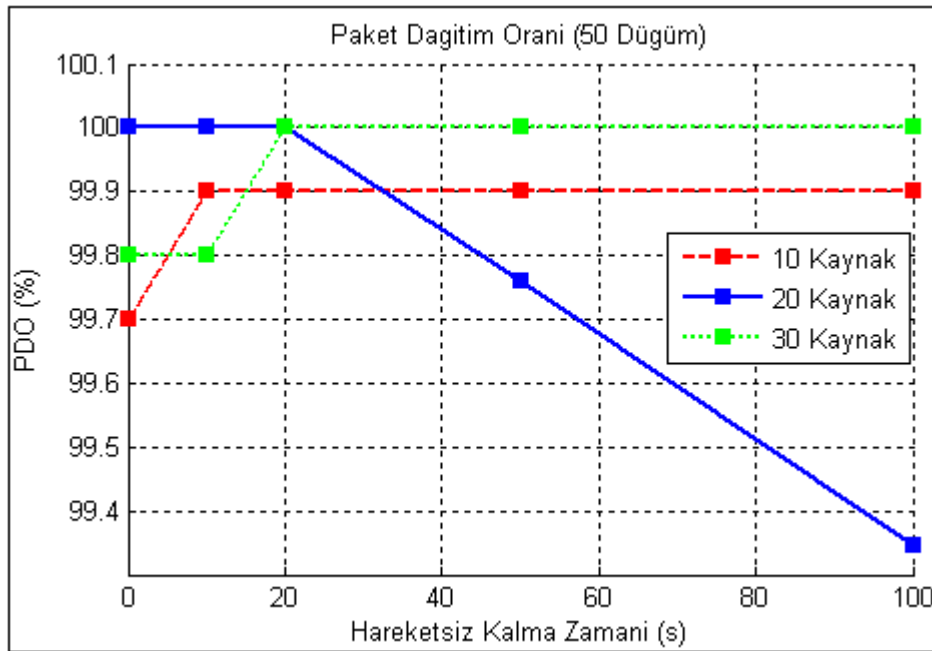
Hareketsizlik zamanı (s)	0	10	20	50	100
20 Düğüm-10 Kaynak	97,30029	96,35286	96,72257	97,81829	100
20 Düğüm-20 Kaynak	99,5	99,75	98,20567	99,43567	99,062

Tablo 4.6’da verilen değerlere göre, 20 düğümlü ağ, 20 kaynaklı durumda daha iyi paket dağıtım oranı sağlamaktadır.

50 düğümlü, 10, 20 ve 30 kaynaklı ağ için paket dağıtım oranları Tablo 4.7’de verilmiş ve Şekil 4.9’da grafik olarak gösterilmiştir.

Tablo 4.7. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları

Hareketsizlik zamanı (s)	0	10	20	50	100
50 Düğüm-10 Kaynak	100	100	100	100	100
50 Düğüm-20 Kaynak	100	100	100	99,759	99,345
50 Düğüm-30 Kaynak	100	100	100	100	100



Şekil 4.9. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları

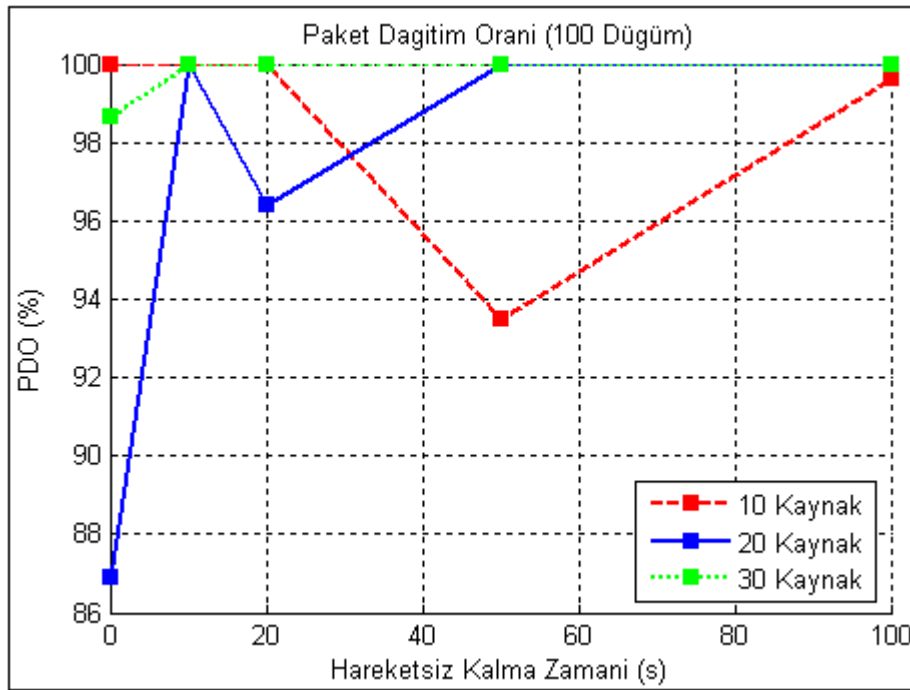
Tablo 4.7’de verilen değerlere göre, 50 düğümlü ağ, yüksek hareketlilik halinde 30 kaynaklı durum için daha iyi paket dağıtım oranı sağlamaktadır.

100 düğümlü, 10, 20 ve 30 kaynaklı ağ için paket dağıtım oranları Tablo 4.8’de verilmiş ve Şekil 4.10’da grafik olarak gösterilmiştir.

Tablo 4.8’de verilen değerlere göre, 100 düğümlü ağ, yüksek hareketlilik halinde 30 kaynaklı durum için daha iyi paket dağıtım oranı sağlamaktadır.

Tablo 4.8. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları

Hareketsizlik zamanı (s)	0	10	20	50	100
100 Düğüm-10 Kaynak	100	100	100	93,473	99,623
100 Düğüm-20 Kaynak	86,872	100	96,397	100	100
100 Düğüm-30 Kaynak	98,663	100	100	100	100



Şekil 4.10. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın paket dağıtım oranları

Yukarıdaki paket dağıtım oranı sonuçlarından görüldüğü gibi, KTYA, yüksek düğümlü ağlarda çok kaynaklı durum için, düşük düğümlü ağlarda az kaynaklı durum için daha iyi paket dağıtım oranı sağlamaktadır.

4.3.4. Paket kayıp oranı

Paket kayıp oranı, ağda hedefine ulaşamayan paketlerin oranıdır. Kaynak düğümlerden çıkan tüm veri paketlerinden hedefine ulaşamayanların yüzde cinsinden

oranı hesaplanarak belirlenir.

Paket kayıp oranı aşağıdaki ifade ile hesaplanır:

$$PKO = \frac{n'_h}{n_T} \cdot 100 \quad (4.4)$$

Denklem 4.4'de, *PKO*: Paket Kayıp Oranı, n'_h : hedefine ulaşmayan paketlerin sayısı, n_T : gönderilen toplam paket sayısıdır.

Paket kayıp oranında kullanılan n'_h denklem 4.5'deki ifade ile bulunur.

$$n'_h = n_T - n_h \quad (4.5)$$

Denklem 4.5'de, n_T : gönderilen toplam paket sayısı, n_h : hedefine ulaşan paket sayısıdır. Bu durumda Paket kayıp oranı, Paket dağıtım oranı türünden de bulunabilir.

$$PKO = 1 - PDO \quad (4.6)$$

Denklem 4.6'da *PKO*: Paket kayıp Oranı ve *PDO*: Paket Dağıtım Oranıdır.

Bölüm 4.3.3'de verilen paket dağıtım oranları kullanılarak 10, 20, 50 ve 100 düğümlü, 10 kaynaklı ağlar için paket kayıp oranları Tablo 4.9'da verilmiştir.

Tablo 4.9'daki değerlere göre, KTMYA için, yüksek düğümlü ve çok hareketli ağlarda paket kayıp oranlarının artmasına yol açmaktadır.

4.3.5. Uçtan uca ortalama paket gecikmesi

Uçtan uca paket gecikmesi, hedefine ulaşan bir paketin, hedefe ulaşma süresini belirlemede kullanılan bir kriterdir. Kaynak düğümden çıkan veri paketlerinin hedeflerine ulaşma sürelerinin ortalamasıdır.

Tablo 4.9. 10, 20, 50 ve 100 düğümlü, 10 kaynaklı ağların paket kayıp oranı değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
10 Düğüm-5 Kaynak	0,48	0	0	0	0
10 Düğüm-10 Kaynak	0	0	0	0	0
20 Düğüm-10 Kaynak	2,69971	3,64714	3,27743	2,18171	0
20 Düğüm-20 Kaynak	0,5	0,25	1,79433	0,56433	0,938
50 Düğüm-10 Kaynak	0	0	0	0	0
50 Düğüm-20 Kaynak	0	0	0	0,241	0,655
50 Düğüm-30 Kaynak	0	0	0	0	0
100 Düğüm-10 Kaynak	0	0	0	6,527	0,377
100 Düğüm-20 Kaynak	13,128	0	3,603	0	0
100 Düğüm-30 Kaynak	1,337	0	0	0	0

Ağdaki her bir düğümün gönderdiği veri paketlerinin, kaynaktan çıktıkları zaman ile hedef düğüme ulaştıkları zaman farkı belirlenir. Belirlenen zaman farklarının ortalaması alınarak, uçtan uca paket gecikmesi hesaplanır.

Tasarsız ağlarda bir düğümden komşu düğüme gönderilen paketin gecikmesi denklem 4.3’de verilen ifade kullanılarak hesaplanır [52].

$$d = \frac{S}{B} \quad (4.3)$$

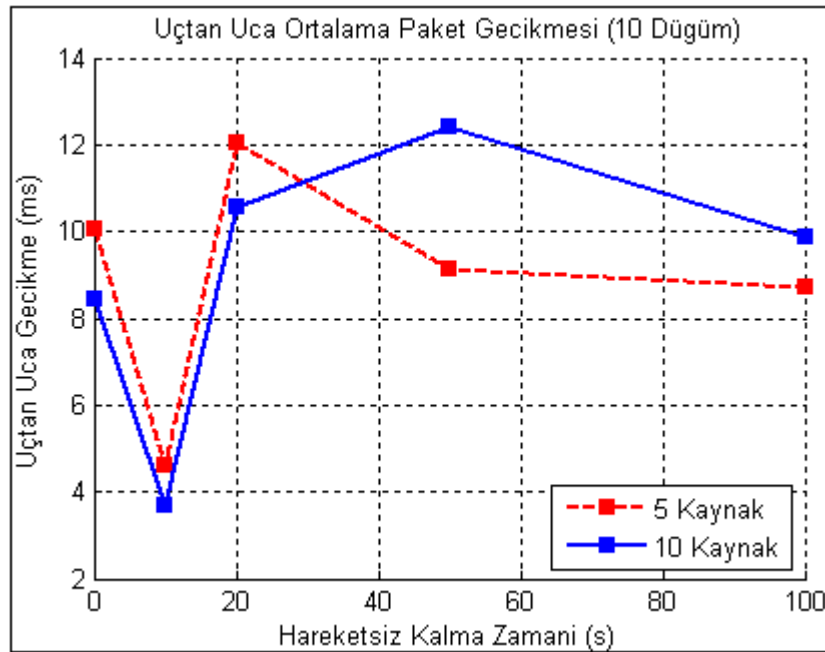
Bu ifadede d , zaman gecikmesi, S , gönderilen verinin bit sayısı, B , efektif bant genişliğidir.

Geliştirilen benzetim programında uçtan uca ortalama paket gecikmesini hesaplamak için, kodları Ek F’de verilen $delay.m$ fonksiyonu yazılmıştır. 10 düğümlü ağ için $delay10.m$, 20 düğümlü ağ için $delay20.m$, 50 düğümlü ağ için $delay50.m$ ve 100 düğümlü ağ için $delay100.m$ fonksiyonları oluşturulmuştur.

10 düğümlü, 5 ve 10 kaynaklı ağ için benzetim sonucunda elde edilen uçtan uca ortalama paket gecikmesi değerleri Tablo 4.10’da verilmiş ve Şekil 4.11’de grafik olarak gösterilmiştir.

Tablo 4.10. 10 düğümlü, 5 ve 10 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
10 Düğüm-5 Kaynak (ms)	10,037	4,6082	12,044	9,1398	8,7234
10 Düğüm-10 Kaynak (ms)	8,4226	3,672	10,557	12,399	9,8539



Şekil 4.11. 10 düğümlü, 5 ve 10 kaynaklı ağın uçtan uca ortalama paket gecikmesi

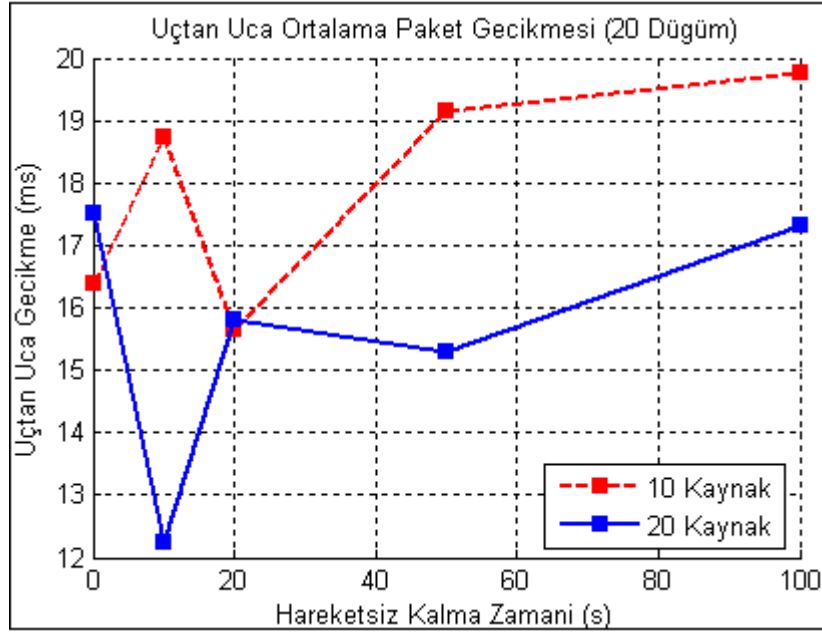
Tablo 4.10'daki değerlere göre 10 düğümlü ağ, 10 kaynaklı durumda daha iyi uçtan uca ortalama paket gecikmesi değerlerine sahiptir.

20 düğümlü, 10 ve 20 kaynaklı ağ için benzetim sonucunda elde edilen uçtan uca ortalama paket gecikmesi değerleri Tablo 4.11'de verilmiş ve Şekil 4.12'de grafik olarak gösterilmiştir.

Tablo 4.11. 20 düğümlü, 10 ve 20 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri

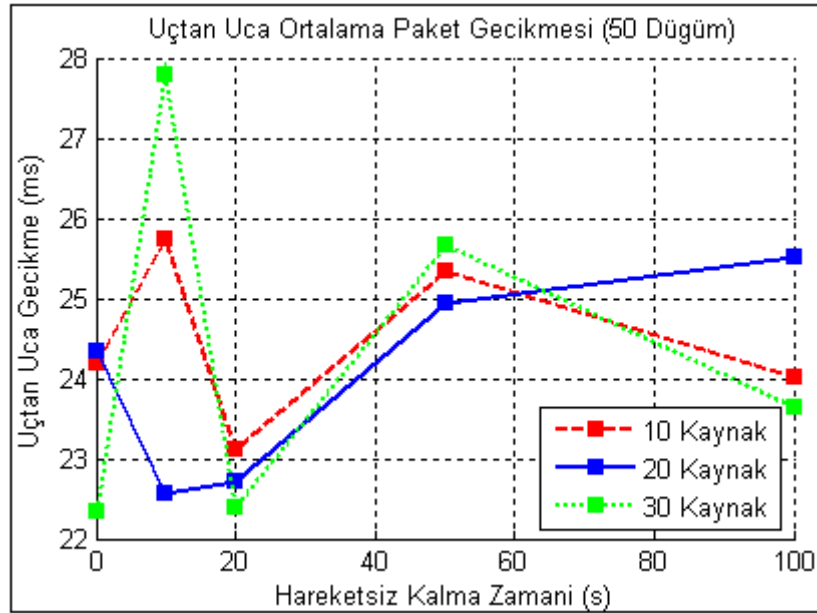
Hareketsizlik zamanı (s)	0	10	20	50	100
20 Düğüm-10 Kaynak (ms)	16,39	18,718	15,65	19,145	19,774
20 Düğüm-20 Kaynak (ms)	17,499	12,228	15,819	15,283	17,332

50 düğümlü, 10, 20 ve 30 kaynaklı ağ için benzetim sonucunda elde edilen uçtan uca ortalama paket gecikmesi değerleri Tablo 4.12'de verilmiş ve Şekil 4.13'de grafik olarak gösterilmiştir.



Şekil 4.12. 20 düğümlü, 10 ve 20 kaynaklı ağın uçtan uca ortalama paket gecikmesi

Tablo 4.11'deki değerlere göre 20 düğümlü ağ, 10 kaynaklı durumda daha iyi uçtan uca ortalama paket gecikmesi değerlerine sahiptir.



Şekil 4.13. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi

Tablo 4.11'deki değerlere göre 50 düğümlü ağ, 30 kaynaklı durumda daha iyi uçtan uca ortalama paket gecikmesi değerlerine sahiptir.

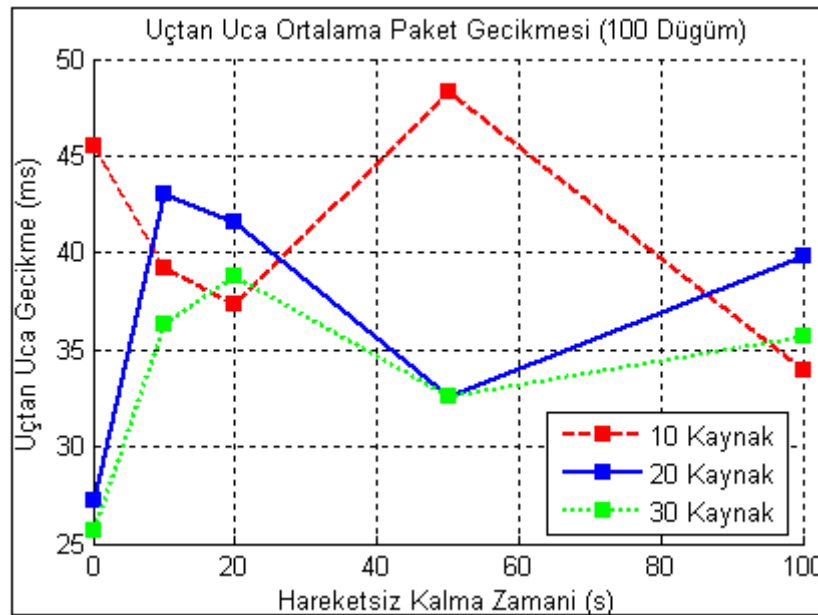
Tablo 4.12. 50 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
50 Düğüm-10 Kaynak (ms)	24,184	25,728	23,114	25,331	24,005
50 Düğüm-20 Kaynak (ms)	24,331	22,557	22,714	24,945	25,511
50 Düğüm-30 Kaynak (ms)	22,337	27,778	22,387	25,665	23,637

100 düğümlü, 10, 20 ve 30 kaynaklı ağ için benzetim sonucunda elde edilen uçtan uca ortalama paket gecikmesi değerleri Tablo 4.13’de verilmiş ve Şekil 4.14’de grafik olarak gösterilmiştir.

Tablo 4.13. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
100 Düğüm-10 Kaynak (ms)	45,455	39,219	37,39	48,314	33,902
100 Düğüm-20 Kaynak (ms)	27,217	43,052	41,597	32,554	39,789
100 Düğüm-30 Kaynak (ms)	25,722	36,359	38,788	32,548	35,723



Şekil 4.14. 100 düğümlü, 10, 20 ve 30 kaynaklı ağın uçtan uca ortalama paket gecikmesi

Tablo 4.11’deki değerlere göre 50 düğümlü ağ, 10 kaynaklı durumda daha iyi uçtan uca ortalama paket gecikmesi değerlerine sahiptir.

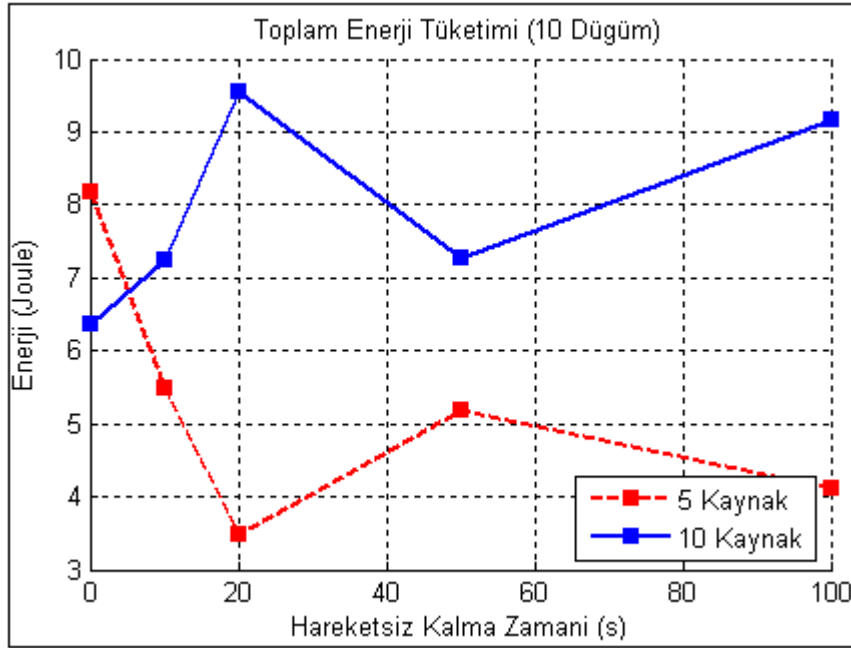
Yukarıdaki ortalama uçtan uca ortalama paket gecikmesi değerlerine göre, KTMYA, düşük kaynaklı ağlarda daha iyi performans sağlamaktadır.

4.3.6. Enerji Verimliliği

Ağdaki düğümlerin bataryalarını etkin bir şekilde kullanabilmeleri için, enerji tüketimini en aza indirmeleri gereklidir. Yapılan benzetim yazılımıyla düğümlerin enerji tüketimleri hesaplanabilmektedir.

Geliştirilen benzetim programında uçtan uca ortalama paket gecikmesini hesaplamak için, kodları Ek F’de verilen *enerji.m* fonksiyonu yazılmıştır. 10 düğümlü ağ için *enerji10.m*, 20 düğümlü ağ için *enerji20.m*, ve 50 düğümlü ağ için *enerji50.m* fonksiyonları oluşturulmuştur. Bu fonksiyonların yazılması esnasında, Ek D’de verilen enerji ifadeleri kullanılmıştır.

10 düğümlü, 5 ve 10 kaynaklı ağ için benzetim sonucunda elde edilen düğümlerin toplam enerji tüketimi değerleri Tablo 4.14’de verilmiş ve Şekil 4.15’de grafik olarak gösterilmiştir.



Şekil 4.15. 10 düğümlü, 5 ve 10 kaynaklı ağdaki düğümlerin toplam enerji tüketimi

Tablo 4.14. 10 düğümlü, 5 ve 10 kaynaklı ağıdaki düğümlerin toplam enerji tüketimi değerleri

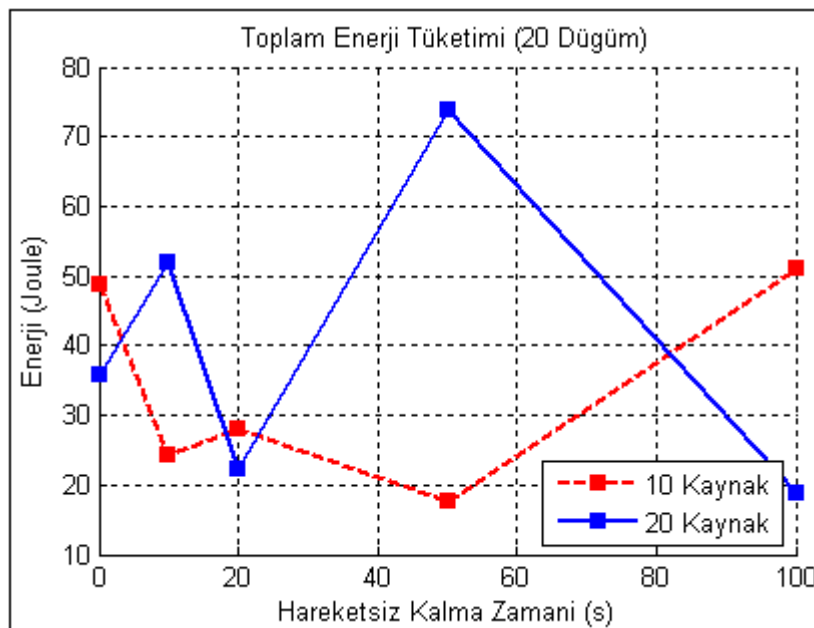
Hareketsizlik zamanı (s)	0	10	20	50	100
10 Düğüm-5 Kaynak (Joule)	8.162	5.4772	3.4846	5.179	4.1178
10 Düğüm-10 Kaynak (Joule)	6.3571	7.2369	9.5454	7.2775	9.1551

Tablo 4.14'e göre, 10 düğümlü ağ, 5 kaynak durumunda daha düşük enerji tüketimine sahiptir.

20 düğümlü, 10 ve 20 kaynaklı ağ için benzetim sonucunda elde edilen düğümlerin toplam enerji tüketimi değerleri Tablo 4.15'de verilmiş ve Şekil 4.16'de grafik olarak gösterilmiştir.

Tablo 4.15. 20 düğümlü, 10 ve 20 kaynaklı ağıdaki düğümlerin toplam enerji tüketimi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
20 Düğüm-10 Kaynak (Joule)	48.74	24.2881	28.1471	17.6119	51.093
20 Düğüm-20 Kaynak (Joule)	35.849	51.7793	22.1121	73.862	18.6575



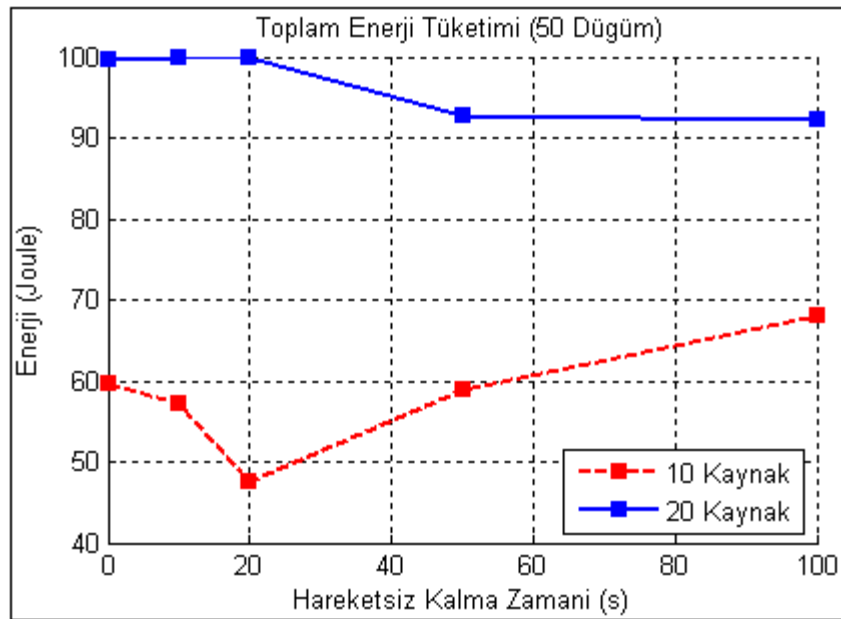
Şekil 4.16. 20 düğümlü, 10 ve 20 kaynaklı ağıdaki düğümlerin toplam enerji tüketimi

Tablo 4.15'e göre, 20 düğümlü ağ, 10 kaynak durumunda daha düşük enerji tüketimine sahiptir.

50 düğümlü, 10 ve 20 kaynaklı ağ için benzetim sonucunda elde edilen düğümlerin toplam enerji tüketimi değerleri Tablo 4.16’da verilmiş ve Şekil 4.17’de grafik olarak gösterilmiştir.

Tablo 4.16. 50 düğümlü, 10 ve 20 kaynaklı ağdaki düğümlerin toplam enerji tüketimi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
50 Düğüm-10 Kaynak (Joule)	59.6897	57.0488	47.4201	58.9503	68.015
50 Düğüm-20 Kaynak (Joule)	92.7544	103.7397	74.8659	88.8059	113.7368



Şekil 4.17. 50 düğümlü, 10 ve 20 kaynaklı ağdaki düğümlerin toplam enerji tüketimi

Tablo 4.16’ya göre, 50 düğümlü ağ, 10 kaynak durumunda daha düşük enerji tüketimine sahiptir.

Yukarıdaki enerji tüketimi değerlerine göre, KTMYA, az kaynaklı ağlarda daha düşük enerji tüketimine sahiptir.

4.4. Performans Karşılaştırması

KTMYA için benzetim sonucunda elde edilen performans değerleri, performans karşılaştırma işlemlerinde sıklıkla kullanılan diğer algoritmalarla da

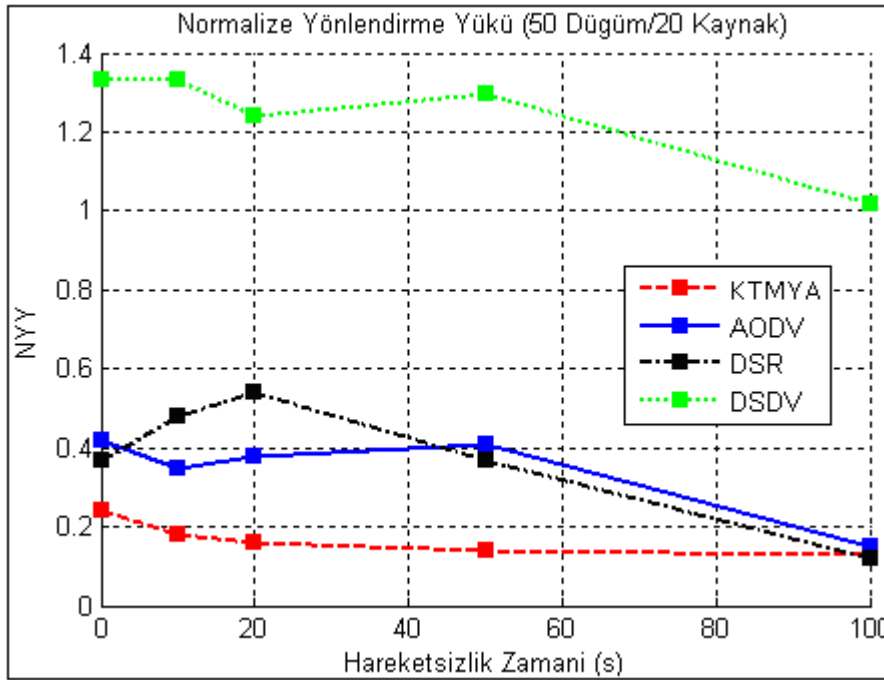
karşılaştırılmıştır. Bu algoritmalar Bölüm 2’de detaylandırılan, literatürde öne çıkmış algoritmalarıdır.

İsteğe bağlı algoritmalarından AODV ve DSR algoritmaları, tabloya dayalı algoritmalarından DSDV ve konum tabanlı algoritmalarından DREAM karşılaştırma amacıyla kullanılmıştır. [3,53]’deki AODV, DSDV ve DSR için kullanılan performans sonuçları ile [54]’deki DREAM için kullanılan performans sonuçları farklı biçimlerde olduğu için, iki grup karşılaştırma ayrı ayrı ele alınmaktadır.

50 düğümlü ve 20 kaynaklı bir ağ için KTMYA, AODV, DSDV ve DSR algoritmalarının yönlendirme yükü değerleri Tablo 4.17’de verilmekte ve Şekil 4.18’de grafik olarak gösterilmektedir.

Tablo 4.17. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için yönlendirme yükü değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
KTMYA (YY)	0,2408	0,1816	0,162	0,1385	0,129
AODV (YY)	0,42	0,35	0,38	0,41	0,15
DSR (YY)	0,37	0,48	0,54	0,37	0,12
DSDV (YY)	1,3333	1,3333	1,2427	1,2941	1,0161

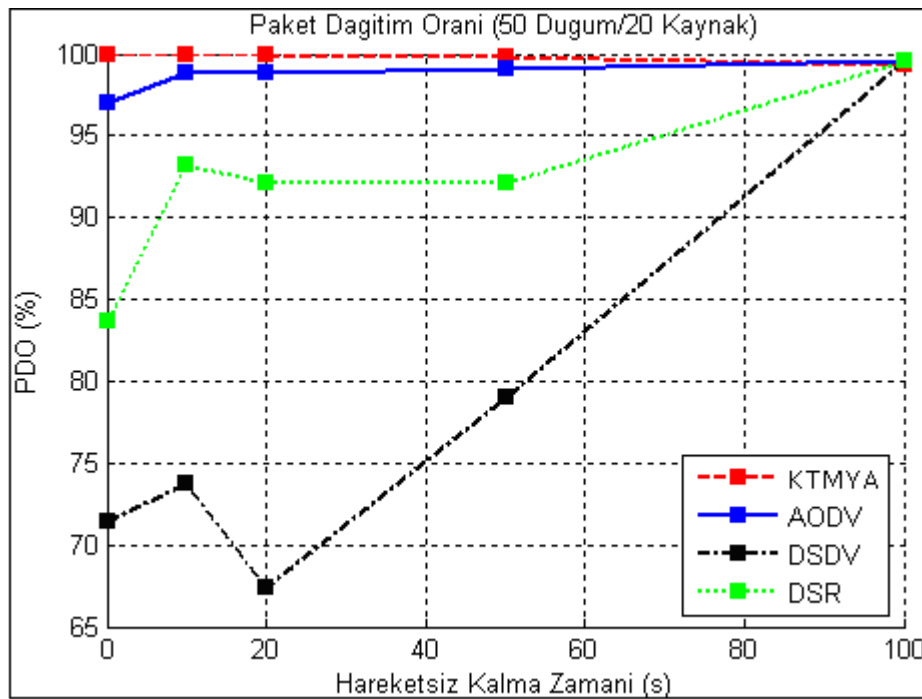


Şekil 4.18. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için yönlendirme yükü değerleri

Tablo 4.17 ve Şekil 4.18'den de görüleceği gibi, en düşük normalize yönderme yükü KTMYA'ya aittir. MANET tarafından belirlenen ölçülebilir performans kriterlerinden olan bu kriterin düşük değerde olması, algoritmanın performansının yüksek olduğunu göstermektedir.

Yönlendirme yükü açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 10,702 tabloya dayalı algoritmalarından DSDV'ye göre en az % 12,69 oranında daha iyi performans değerlerine sahiptir.

50 düğümlü ve 20 kaynaklı bir ağ için KTMYA, AODV, DSDV ve DSR algoritmalarının paket dağıtım oranları Tablo 4.18'de verilmiş ve Şekil 4.19'da grafik olarak gösterilmiştir.



Şekil 4.19. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için paket dağıtım oranları

Tablo 4.18 ve Şekil 4.19'dan da görüleceği gibi, en yüksek paket dağıtım oranı değeri KTMYA'ya aittir. MANET tarafından belirlenen ölçülebilir performans kriterlerinden olan bu kriterin yüksek değerde olması, algoritmanın performansının yüksek olduğunu göstermektedir.

Tablo 4.18. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için paket dağıtım oranları

Hareketsizlik zamanı (s)	0	10	20	50	100
KTMYA (PDO %)	100	100	100	99,759	99,345
AODV (PDO %)	96,97	98,83	98,83	99,07	99,53
DSDV (PDO %)	71,39	73,72	67,44	79,06	99,53
DSR (PDO %)	83,72	93,25	92,09	92,09	99,53

Paket dağıtım oranı açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 1,17, tabloya dayalı algoritmalarından DSDV'ye göre en az % 20 oranında daha iyi performans değerlerine sahiptir.

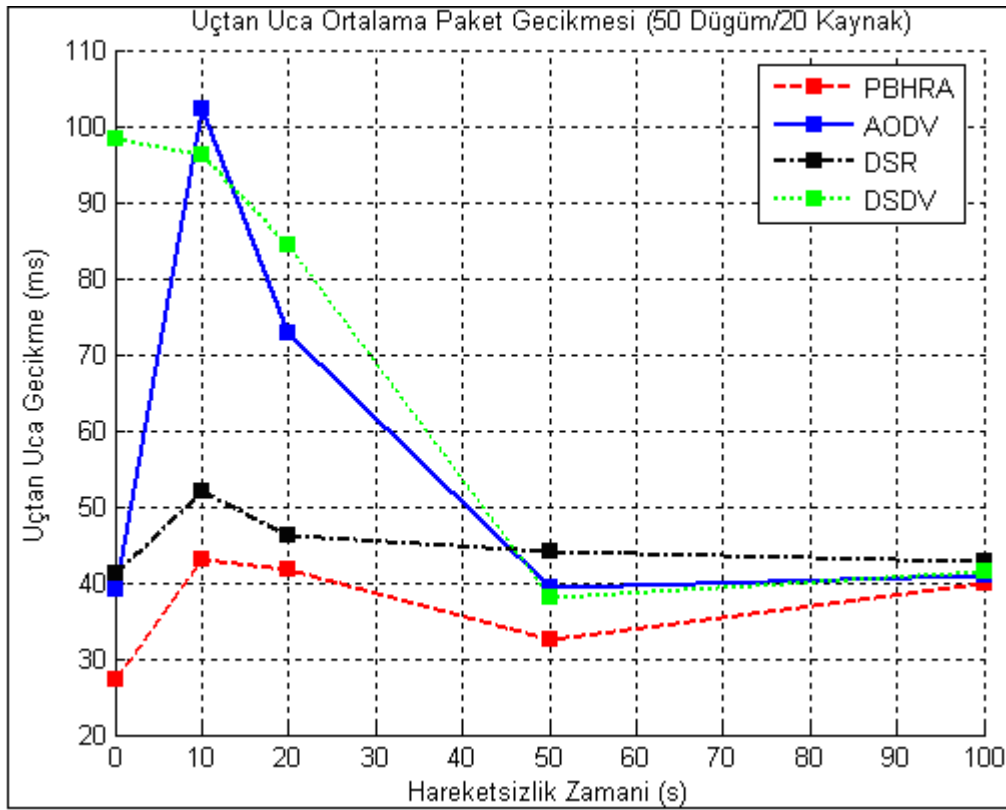
50 düğümlü ve 20 kaynaklı bir ağ için KTMYA, AODV, DSDV ve DSR algoritmalarının uçtan uca ortalama paket gecikmesi değerleri milisaniye cinsinden Tablo 4.19'da verilmiş ve Şekil 4.20'de grafik olarak gösterilmiştir.

Tablo 4.19. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için uçtan uca paket gecikmesi değerleri

Hareketsizlik zamanı (s)	0	10	20	50	100
KTMYA (Gecikme-ms)	27,217	43,052	41,597	32,554	39,789
AODV (Gecikme-ms)	39,1	102,2	72,65	39,45	40,81
DSDV (Gecikme-ms)	98,34	96,12	84,43	38	41,34
DSR (Gecikme-ms)	41,23	52,02	46,23	44,12	42,65

Tablo 4.19 ve Şekil 4.20'den de görüleceği gibi, en düşük uçtan uca ortalama paket gecikmesi değeri KTMYA'ya aittir. MANET tarafından belirlenen ölçülebilir performans kriterlerinden olan bu kriterin düşük değerde olması, algoritmanın performansının yüksek olduğunu göstermektedir.

Uçtan uca ortalama paket gecikmesi kriteri açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 2,6, tabloya dayalı algoritmalarından DSDV'ye göre en az % 3,9 oranında daha iyi performans değerlerine sahiptir.



Şekil 4.20. 50 düğümlü, 20 kaynaklı ağın KTMYA, AODV, DSR ve DSDV algoritmaları için uçtan uca paket gecikmesi

Geliştirilen algoritma konum bilgilerine dayalı olarak yönlendirme kararı veren algoritmalardan DREAM algoritması ile performans karşılaştırmasına tabi tutulmuştur. DREAM algoritmasına ait performans değerleri [54]'den alınmıştır. Tablo 4.20'de 50 ve 100 düğümlü, 20 kaynaklı ağ için, yönlendirme yükü, uçtan uca ortalama paket gecikmesi ve paket dağıtım oranı değerleri karşılaştırmalı olarak verilmiştir.

Tablo 4.20. 50 ve 100 düğümlü, 20 kaynaklı ağların KTMYA ve DREAM için performans değerleri

Konum tabanlı algoritmalar	Normalize Yönlendirme Yükü		Uçtan Uca Ortalama Gecikme (ms)		Paket Dağıtım Oranı (%)	
	DREAM	KTMYA	DREAM	KTMYA	DREAM	KTMYA
50 düğüm 20 Kaynak	1,07	0,24	90	24,331	65	100
100 düğüm 20 Kaynak	18	0,44	120	27,217	48	86,8

Tablo 4.20'den görüleceği üzere; yönlendirme yükü açısından KTMYA, DREAM'a göre % 22,43 oranında, paket dağıtım oranı açısından KTMYA, DREAM'a göre %

38,8 oranında, uçtan uca ortalama paket gecikmesi açısından % 275 oranında performans üstünlüğüne sahiptir.

4.5. Sonuç

Bu bölümde, öncelikli olarak tasarsız ağlarda kullanılması gereken performans değerlendirme parametreleri üzerinde durulmuştur. Geliştirilen KTMYA'nın benzetim için modellenmesi gerçekleştirilmiş ve yönlendirme yükü, paket dağıtım oranı, paket kayıp oranı, uçtan uca ortalama paket gecikmesi ve enerji tüketimi açısından performans değerlendirmesi yapılmıştır. Ayrıca KTMYA'nın isteğe bağlı algoritmalarından AODV ve DSR, tabloya dayalı algoritmalarından DSDV ve konum tabanlı algoritmalarından DREAM ile performans karşılaştırması yapılmıştır. Karşılaştırma sonuçlarına göre KTMYA'nın daha iyi performans değerlerine sahip olduğu görülmüştür.

BÖLÜM 5. SONUÇLAR VE DEĞERLENDİRMELER

5.1. Sonuçlar

Bu çalışmada, gezgin tasarsız ağlarda daha düşük yönlendirme ek yükü oluşturacak ve düğümlerin daha düşük enerji tüketmelerini sağlayacak yeni bir yönlendirme algoritması geliştirilmiştir. Algoritmayı diğer algoritmalarından farklı kılan özellikleri, yönlendirme kararı verilmesinde, düğümlerin konum bilgilerine ek olarak, batarya ve yoğunluk değerlerinin de kullanılmasıdır. “Konum Tabanlı Melez Yönlendirme Algoritması (Position Based Hybrid Routing Algorithm-PBHRA) adı verilen bu algoritmanın performans değerlendirmesi yapılmıştır.

Aşağıda, yapılan çalışmalar, tezin sunumuna uygun olarak, kısaca hatırlatılmış ve elde edilen bulgular sunulmuştur.

1. Gezgin tasarsız ağların, düğüm hareketlerinden kaynaklanan sık topoloji değişiklikleri ve bu ağların karakteristik özelliklerinden kaynaklanan kısıtlarından dolayı, bant genişliği ve düğüm bataryaları gibi kaynakları daha verimli kullanacak yönlendirme algoritmalarının tasarlanması gerekmektedir (Bölüm 1).
2. Altyapılı (infrastructured) ve altyapısız (infrastructereless) kablosuz ağların karşılaştırması yapılmış ve farklılıkları ortaya konulmuştur. Ayrıca sensör (algılayıcı) ağları ile tasarsız ağların benzer ve farklı yanları üzerinde durulmuştur. Altyapılı ağlarda baz istasyonu veya erişim noktası olmasına karşın, altyapısız ağlarda böyle bir donanım yoktur. Algılayıcı ağ düğümleri ise tasarsız ağ düğümlerine göre çok küçük boyutta ve sınırlı kapasiteye sahiptir (Bölüm 1).

3. Yeni bir algoritma tasarımı yapabilmek için, literatürde yer alan algoritmaların çalışma prensipleri tartışılmış ve düğümlerin konum bilgilerine dayalı olarak yönlendirme kararı veren algoritmalar üzerinde durulmuştur (Bölüm 1 ve Bölüm 2).
4. Düğümlerin konum bilgilerine, batarya miktarlarına ve tampon yoğunluğuna bağlı olarak yönlendirme kararı veren KTMYA'nın adımları oluşturulmuştur (Bölüm 3).
5. Altyapılı ve altyapısız kablosuz ağların çalışma yöntemleri kullanılarak, KTMYA adı verilen algoritmanın ayrıntılı çalışma adımları verilmiş ve algoritmadan protokol gerçekleştirimine geçişte temel olacak çerçeve (frame) yapıları verilmiştir Ağda düğümler arasında, kontrol, güncelleme ve veri paketlerinin dağıtım mekanizmaları belirlenmiştir (Bölüm 3).
 - a. Ağda “master düğüm” adı verilen bir düğümde, diğer tüm düğümlere ait bilgiler bulundurulur. Düğümlerin paket gönderecekleri düğüme giden yolu bulmaları işlemini de master düğüm gerçekleştirir.
 - b. Ağda master düğüm görevini üstlenecek düğümün belirlenmesi işlemi gerçekleştirilmiştir.
 - c. Master düğüm yönlendirmede kullanılacak yolu bulur, ancak veri paketlerinin dağıtımını yapmaz. Bu açıdan, geliştirilen algoritma, altyapılı ağlarda kullanılan algoritmalarından ayrılır.
6. Geliştirilen algoritma düğümlerin konum, batarya ve yoğunluk bilgilerine bağlı olarak yönlendirme kararı verdiği için, bu üç bilgi kullanılarak bulanık mantık yöntemiyle düğümler arasında “maliyet” değerleri oluşturulmuştur. Maliyet değerlerinin oluşturulması için “mesafe”, “batarya ömrü” ve “yoğunluk” giriş değişkenleri dilsel değişkenler kullanılarak bulanıklaştırılmış ve mamdani (bulanık kümelerin ağırlık merkezleri) bulanık çıkarım yöntemi

kullanılarak durulaştırma işlemi gerçekleştirilmiştir. Bu bağlamda bulanık mantık ve uygulanması ile ilgili bilgi de verilmiştir (Bölüm 3).

7. Ağdaki düğüm sayısının yüksek olması durumunda master düğüm üzerinde bulundurulmuş yönlendirme bilgisi yüksek boyutlara ulaşabilir ve yüksek boyutlu matrisler üzerinde işlem yapılacağından dolayı yönlendirme isteklerine cevap verme süresi artacağından dolayı, ağda kümeleme önerisinde bulunulmuştur. Kümeleme işleminin uygulanabilmesi için öncelikli olarak, kümeleme algoritmaları verilmiş ve içlerinden KTMYA için uygulanacak kümeleme algoritmasının seçimi yapılmıştır. Fuzzy C-Means (FCM) kümeleme algoritması geliştirilen algoritmaya uygulanmış ve örnek ağlar üzerinde çalışması gösterilmiştir (Bölüm 3).
8. Algoritmanın performans değerlendirme işlemini yapmak için öncelikli olarak, IETF MANET çalışma grubu tarafından belirlenmiş ve RFC 2501 olarak yayınlanmış olan performans ölçüm kriterleri verilmiştir [2]. Yönlendirme yükü, Paket dağıtım oranı, uçtan uca ortalama paket gecikmesi, enerji tüketimi ve paket kayıp oranı yönlerinden 10, 20, 50 ve 100 düğümlü ağlar, MATLAB 7.01'de kodlanan bir benzetim yazılımı kullanılarak benzetime tabi tutulmuştur. Elde edilen benzetim sonuçları tabloya dayalı, isteğe bağlı ve konum tabanlı algoritmaların performans sonuçları ile karşılaştırılmıştır (Bölüm 4).
 - a. İsteğe bağlı ve tabloya dayalı algoritmalarla yapılan performans karşılaştırma sonuçları aşağıda verilmiştir.
 - i. Yönlendirme yükü açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 10,702 tabloya dayalı algoritmalarından DSDV'ye göre en az % 12,69 oranında daha iyi performans değerlerine sahiptir.
 - ii. Paket dağıtım oranı açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 1,17, tabloya dayalı algoritmalarından

DSDV'ye göre en az % 20 oranında daha iyi performans değerlerine sahiptir.

iii. Uçtan uca ortalama paket gecikmesi kriteri açısından KTMYA, isteğe bağlı algoritmalarından AODV ve DSR'ye göre en az % 2,6, milisaniye, tabloya dayalı algoritmalarından DSDV'ye göre en az % 3,9 oranında daha iyi performans değerlerine sahiptir.

b. Konum tabanlı algoritmalarından DREAM algoritması ile yapılan performans karşılaştırma sonuçları aşağıda verilmiştir.

iv. Yönlendirme yükü açısından KTMYA, DREAM'a göre % 22,43 oranında daha iyi performans sağlamıştır.

v. Paket dağıtım oranı açısından KTMYA, DREAM'a göre % 38,8 oranında performans üstünlüğüne sahiptir.

vi. Uçtan uca ortalama paket gecikmesi açısından % 275 oranında performans üstünlüğüne sahiptir.

9. Geliştirilen algoritmanın daha iyi anlaşılabilmesi için örnek bir ağ üzerinde uygulama yapılmıştır. 20 düğümlü örnek ağın master düğüm seçiminden, enerji tüketimi hesabına kadar bütün aşamaları uygulanmıştır (Bölüm 7).

5.2. Çalışmanın Getirdiği Katkılar

Bu çalışmanın bilime ve teknolojiye getirdiği ana katkılar aşağıda maddeler halinde sunulmaktadır:

1. Bilime ve teknolojiye yenilik getirme:

a. Gezgin kablosuz ağlar için “Konum Tabanlı” algoritmalar sınıfında yeni bir algoritma geliştirilmiştir. Geliştirilen algoritma ağın bant

genişliğini ve düğümlerin batarya ömürlerini daha etkin kullanmayı amaçlamıştır.

- b. Konum tabanlı algoritmaların yönlendirme kararını vermelerinde düğümler arası mesafe bilgilerine ek olarak düğümlerin batarya ömürleri ve tampon yoğunluk bilgileri de kullanılmıştır.
- c. Geliştirilen algortmada kullanılan paketlerin çerçeve yapıları belirlenmiştir.
- d. MATLAB 7.0.1 kullanılarak geliştirilen algoritmanın benzetimi yapılmıştır.

2. Bilinen bir yöntemi yeni bir alana uygulama:

- a. Konum tabanlı algoritmaların yönlendirme kararını vermelerinde bulanık mantık yöntemi kullanılmıştır.
- b. FCM kümeleme yöntemi konum tabanlı yönlendirme algoritmalarını kullanan düğümlerin oluşturduğu ağlara uygulanmıştır.

5.3. Tartışma ve Öneriler

Tez çalışmasında geliştirilen algoritma ve benzetim sonuçlarından faydalanarak gelecekte yapılabilecek çalışmalara ışık tutmak üzere aşağıda altı öneri sunulmaktadır:

i. Protokol gerçekleştirilmesi:

Bu tez çalışmasında KTMYA, algoritma olarak geliştirilmiş ve performans ölçümleri yapılmıştır. Algoritmanın protokol olarak uygulanması ile ilgili çerçeve yapıları da verilmiştir. Ancak bu tez çalışmasında algoritmanın

protokol olarak uygulaması yapılmamıştır. Tez içinde verilen çerçeve yapıları kullanılarak, çalıştırılabilir ve uygulanabilir bir protokol haline getirilebilir.

Algoritmanın protokol gerçekleştirilmesi yapıldıktan sonra, RFC olarak yayınlanması konusunda çalışmalar yapılabilir.

ii. En kısa yol optimizasyonu:

KTMYA'ya göre, master düğüm yönlendirme kararını verirken en kısa yol bulma işlemi gerçekleştirilmektedir. Bu işlem için bulanık mantık sonucu bulunmuş maliyet değerleri üzerinde optimizasyon uygulanmaktadır. Optimizasyon için, geleneksel yöntem olan Bellman-Ford algoritması kullanılmıştır. Genetik algoritma veya karınca kolonisi yöntemlerinden biri veya her ikisi ile optimizasyon işlemi gerçekleştirilip, performans katkısı sağlayıp sağlamadığı araştırılabilir.

iii. Kümelemede ara düğümler:

Algoritma kümelere ayrıldıktan sonra, kümeler arası paket iletimi için, kümeler arasında kalan ara düğümlerin kullanılması önerilmiştir. Bu düğümlerin enerji tüketimleri ve enerji tüketiminin optimizasyonu ile ilgili çalışmalar yapılabilir.

iv. Performans değerlendirme:

Bu tez çalışmasında, geliştirilen algoritma yönlendirme yükü, paket dağıtım oranı, paket kayıp oranı ve uçtan uca ortalama paket gecikmesi kriterlerine göre performans değerlendirmesine tabi tutulmuştur. Bölüm 4.1.1'de verilen ölçülebilir kriterlerden yol bulma hızı ve verimlilik yönünden performans değerlendirme çalışmaları, geliştirilen benzetim yazılımı üzerine yeni modüller eklenerek yapılabilir.

KTMYA'nın tasarımı aşamasında, Bölüm 4.1.2'de verilen ölçülemeyen performans ölçütlerinden en uygun yol bulma, ölçeklenebilirlik, bant genişliği

verimliliği, eski yolları belirleme, düğümlerin hareketlilik hızı ve bölünmüş ağlar konularına çözüm üretmesi sağlanmıştır. Ancak, yük dengeleme, yol bozulmalarının düzeltilmesi ve çoklu yol konularında ise çözüm üretilmemiştir. Bu tez çalışması kullanılarak, çözüm bulunmamış konular üzerinde çalışmalar yapılabilir.

v. OSI 2. katman uygulaması:

Bu çalışmada OSI'nin 3. katmanında çalışan bir yönlendirme algoritması geliştirilmiştir. Alt katman protokolü olarak IEEE 802.11 kullanılmıştır. Alt katmanla ilgili çalışmalar yapıp, KTMYA daha performanslı hale getirilebilir.

Yönlendirme algoritması üzerinde güvenlik ve servis kalitesi ile ilgili çalışmalar yapılabilir.

vi. Fiziksel katman uygulaması:

Algoritmanın fiziksel katmanı ile ilgili olarak, akıllı anten veya yönlendirilmiş anten kullanımı ile ilgili çalışmalar yapıp, paket dağıtım oranında artış ve enerji tüketiminde azalma sağlanabilir. Yönlendirilmiş antenler ile yönlendirme yolu üzerinde sadece ilgili düğüme yayın yapıp diğer düğümlerin bant genişlikleri boşa tüketilmemiş olabilir. Akıllı anten ile gerekli olan duruma, mesafeye ve açığa göre uyarlanabilir yayın yapılarak bant genişliği kullanımına ve düşük enerji tüketimine katkı sağlanabilir.

KAYNAKLAR

- [1] AHMAD, A., *Wireless And Mobile Networks, John Wiley And Sons Inc.*, 2005, Sayfa:240-241
- [2] CORSON, S., MACKER, J., *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, IETF*, <http://www.ietf.org/rfc/rfc2501.txt>
- [3] EHSAN, H., UZMI, Z. A., *Performance Comparison Of Ad Hoc Wireless Network Routing Protocols, 0-7803-8680-IEEE, INMIC 2004*
- [4] WATTENHOFER, R., *Algorithms for ad hoc and sensor Networks, Computer Communications*, Volume 28, Issue 13, 2 August 2005, Pages 1498-1504
- [5] PANTHONG, S., JANTARANG, S., *3G Mobile Wireless Routing Optimization By Genetic Algorithm, CCECE 2003 - CCGEI 2003*, Montreal, May 2003.
- [6] KURUVILA, J., NAYAK, A., STAJMENOVIC, I., *Greedy localized routing for maximizing probability of delivery in wireless ad hoc networks with a realistic physical layer, Journal of Parallel Distributed Computing*, Volume 66, Issue 4, April 2006, Pages 499-506
- [7] HEISSENBUTTEL, M., BRAUN, T., BERNOULLI, T., WALCHLI, M., *BLR: beacon-less routing algorithm for mobile ad hoc networks, Computer Communications*, Volume 27, Issue 11, 1 July 2004, Pages 1076-1086
- [8] KARAYIANNIS, N.B., NADELLA, S., *Power-conserving routing of ad hoc mobile wireless networks based on entropy-constrained algorithms, Ad Hoc Networks*, Volume 4, Issue 1, January 2006, Pages 24-35
- [9] DAI, F., DAI, Q., WU, J., *Power efficient routing trees for ad hoc wireless Networks using directional antenna, Ad Hoc Networks*, Volume 3, Issue 5, September 2005, Pages 621-628
- [10] RABAEY, J.M., AMER, M.J., DA SILVA, J.L., PATEL, D., ROUNDY, S., *PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking, IEEE Computer Society 2000 Published Conference Preceedings*, Volume 33, Issue 7, July 2000 Page(s):42 - 48

- [11] FLEISCHER, R., GOLIN, M., LEA, C.T., WONG, S., Finding optimal paths in MREP routing, *Information Processing Letters*, Volume 89, Issue 2, 31 January 2004, Pages 57-63
- [12] WEISS, E., HIERTZ, G., XU, B., HICKS, S., WALKE, B., GROSS, S., Improving routing performance in wireless ad hoc networks using cross-layer interactions, *Ad Hoc Networks*, Volume 5, Issue 5, July 2007, Pages 579-599
- [13] CONTI, M., GREGORI, E., MESELLI, G., Reliable and efficient forwarding in ad hoc Networks, *Ad Hoc Networks*, Volume 4, Issue 3, May 2006, Pages 398-415
- [14] BASAGNI, S., CHIAMTAC, I., VIOLET, R., Woodward, B.A., A Distance Routing Effect Algorithm for Mobility (DREAM)
- [15] STOJMENOVIC, I., SEDDIGH, M., ZUNIC, J., Dominating Sets And Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks, *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, January 2002
- [16] IMIELINSKI, T., NAVAS, J., GPS Based Geographic Addressing Routing and Resource Discovery, *Communications of The ACM*, Vol. 42, No. 4, April 1999
- [17] KO, Y., VAIDYA, N., Location Aided Routing (LAR) in Mobile Ad Hoc Networks, *The 4th Annual International Conference on Mobile Computing and Networking MOBICOM'98*, Dallas, Texas, October, 1998
- [18] BROCH, J., MALTZ, D.A., A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, October 25-30, 1998, Dallas, Texas,
- [19] RAJU, J.J., ACEVES, G.L., A Comparison of On-Demand and Table Driven Routing for Ad-Hoc Wireless Networks, Volume 3, 18-22 June 2000 Page(s):1702 - 1706
- [20] LEE, S., GERLA, M., Dynamic load-aware routing in ad hoc networks, *IEEE International Conference on Communications (ICC 2001)*, Volume 10, 11-14 June 2001 Page(s):3206 - 3210
- [21] PEARLMAN, M., HAAS, Z., SCHOLANDER, P., TABRIZI, S., Alternate path routing in mobile ad hoc networks, *IEEE Military Communications Conference MILCOM 2000*, Volume 1, 2000, pp:501 - 506
- [22] PHAM, P., PERREAU, S., Multi-path routing protocol with load balancing policy in mobile ad hoc network, *IFIP Int'l Conference on Mobile and Wireless communications Networks (MWCN 2002)*, September 2002

- [23] LEE, Y.J., Routing And Efficient Evaluation Techniques For Multi-Hop Mobile Wireless Networks, School of Electrical and Computer Engineering Georgia Institute of Technology December 2005
- [24] CONTI, M., GREGORI, E., MASELLI, G., Reliable and efficient forwarding in ad hoc Networks, *Ad Hoc Networks* 4 (2006) 398–415
- [25] RAJU, J., ACEVES, J.J., A comparison of on-demand and table driven routing for ad hoc wireless Networks, *Communications 2000. ICC 2000*, IEEE International Conference on Volume 3, June 2000, pp:1702-1706
- [26] CHARLES E., BHZTGAWAT, P., Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, *Comp. Cummunication Rev.*, pp.234-44, Oct. 1994.
- [27] MURTHY, S., ACEVES, G.L., An Efficient Routing Protocol For Wireless Networks, *Mobile Networks and Applications*, Volume 1 , Issue 2 (October 1996), Pages: 183 - 197
- [28] CHIANG, C.C., WU, H.K., LIU, W., GERLA, M., Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel, IEEE Singapore International Conference on Networks, SICON'97, pp. 197-211, Singapore, 16.-17. April 1997, IEEE
- [29] JACQUET, P., MUHLETHALER, P., QAYYUM, A., LAOUITI, A., VIENNOT, L., CLAUSEN, T., OLSR (Optimized Link State Routing Protocol), Optimized Link State Routing Protocol (OLSR), RFC 3626.
- [30] KURUVILA, J., NAYAK, A., STAJMENOVIC, I., Greedy localized routing for maximizing probability of delivery in wireless ad hoc networks with a realistic physical layer, *Journal of Parallel Distributed Computing*, Volume 66, Issue 4, April 2006, Pages 499-506
- [31] FENG, K.T., LU, T.E., Velocity and location aided routing for mobile ad hoc Networks, Vehicular Technology Conference VTC2004-Fall, Volume 4, 26-29 Sept. 2004, pp:2789-2793
- [32] HEISSENBUTTEL, M., BRAUN, T., BERNOULLI, T., WALCHLI, M., BLR: beacon-less routing algorithm for mobile ad hoc networks, *Computer Communications*, Volume 27, Issue 11, 1 July 2004, Pages 1076-1086
- [33] CHENG, A.M.K., RAJAN, K., A digital map/GPS based routing and addressing scheme for wireless ad-hoc Networks, Proceedings of IEEE Intelligent Vehicles Symposium, 9-11 June 2003, pp:17-20
- [34] KARAYIANNIS, N.B., NADELLA, S., Power-conserving routing of ad hoc mobile wireless networks based on entropy-constrained algorithms, *Ad Hoc Networks*, Volume 4, Issue 1, January 2006, Pages 24-35

- [35] DAI, F., DAI, Q., WU, J., Power efficient routing trees for ad hoc wireless Networks using directional antenna, *Ad Hoc Networks*, Volume 3, Issue 5, September 2005, Pages 621-628
- [36] RABAEY, J.M., AMER, M.J., DA SILVA, J.L., PATEL, D., ROUNDY, S., PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking, *IEEE Computer Society 2000 Published Conference Preceedings*, Volume 33, Issue 7, July 2000 Page(s):42 - 48
- [37] PHAM P., PERREAU, S., Multi-path routing protocol with load balancing policy in mobile ad hoc network, *IFIP Int'l Conference on Mobile and Wireless communications Networks (MWCN 2002)*, September 2002
- [38] JUNG, J., PARK, T., KIM, C., A forwarding scheme for reliable and energy-efficient data delivery in cluster-based sensor Networks, *IEEE Communications Letters*, Volume 9, Issue 2, Feb. 2005, pp:112-114
- [39] LEE, Y.J., Routing And Efficient Evaluation Techniques For Multi-Hop Mobile Wireless Networks, School of Electrical and Computer Engineering Georgia Institute of Technology December 2005
- [40] FLEISCHER, R., GOLIN, M., LEA, C.T., WONG, S., Finding optimal paths in MREP routing, *Information Processing Letters*, Volume 89, Issue 2, 31 January 2004, Pages 57-63
- [41] BAYILMIŞ, C., ERTÜRK, İ., ÇEKEN, C., BANDIRMALI, N., “DSR ve AODV Yönlendirme Protokollerinin Başarım Değerlendirmesi”, Elektrik, Elektronik, Bilgisayar Mühendisliği 11. Ulusal Kongresi, EMO, İstanbul, Türkiye, 22-25 Eylül, 2005.
- [42] TAMILARASI, M., SUNDER, V.R., HAPUTHANTHRI, U.M., SOMATHILAKA, C., BABU, N.R., CHANDRAMATHI, S., Scalability Improved DSR Protocol for MANETs, Conference on Computational Intelligence and Multimedia Applications, International Conference on Volume 4, 13-15 Dec. 2007, pp:283-287
- [43] AHMAD, S., AWAN, I., WAQQAS, A., AHMAD, B., Performance Analysis of DSR & Extended DSR Protocols, Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference, 13-15 May 2008, pp:191-196
- [44] YANG, L., HONG, M., Delay Based Load Aware Routing Protocol for Ad Hoc Networks, 4th IEEE Consumer Communications and Networking Conference, Jan. 2007, pp:79-83
- [45] MAUVE, M., WIDMER, A., HARTENSTEIN, H., A survey On Position-Based Routing in Mobile Ad Hoc Networks, Volume 15, *IEEE Network*, Issue 6, 2001, Page: 30-39.

- [46] DEMİRCİ, R., “Similarity relation matrix-based color edge detection”, *International Journal of Electronics and Communication (AEÜ)*, Vol. 61, Issue 7, July 2007, Pages 469-477
- [47] LEON, S.J., *Linear Algebra with Applications*, Prentice-Hall, London, 1998
- [48] BEZDEK, J.C., *Pattern Recognition With Fuzzy Objective Function Algorithms*, Plenum Pres, 1981, New York
- [49] Mathworks Inc Help (Online),
<http://www.mathworks.com/helpdesk/help/toolbox/fuzzy/subclust.shtml>, 10.12.2008
- [50] HALGAMUGE, M.N., GURU, S.M., JENNINGS, A., Energy efficient cluster formation in wireless sensor Networks, *ICT-2003 10th International Conference on Telecommunications*, Volume 2, Issue 23, Page(s): 1571 – 1576, 2003.
- [51] Matworks Inc. File Exchange Center,
<http://www.mathworks.in/matlabcentral/fileexchange/4266>, 10.12.2008
- [52] CHEN, K., XUE, Y., SHAH, S.H., NAHRSTEDT, K., Understanding bandwidth-delay product in mobile ad hoc Networks, *Computer Communications*, Volume 27, Issue 10, 20 June 2004, Pages 923-934
- [53] PERKINS, C.E., ROYER, E.M., DAS, S.R., MARINA, M.K., Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks, *IEEE Personal Communication*, 2001.
- [54] BAKHOUYA, M. COTTIN, N., Performance Evaluation of the Location-Based Protocol DREAM for Large Mobile Ad Hoc Networks, *New Technologies Mobility and Security-2008*, sayfa:1-6, 2008.
- [55] GLISIC, S.G., *Advanced Wireless Networks : 4G Technologies*, *John Wiley And Sons Ltd*, 2006, Sayfa: 191-195
- [56] www.ta-eng.com/industry/mforum/fuzzy/preface.htm (08.08.2002)
- [57] ŞEN, Z., “Bulanık (Fuzzy) Mantık Ve Modelleme İlkeleri”, Bilge Sanat Yapım Yayınevi, İstanbul, 2001
- [58] ÇİFTÇİ, H., “Fuzzy Logic Approximation For Some Mathematical Functions”, *OGÜ Lisans Tezi*, Eskişehir, 2002
- [59] <http://members.tripod.com/~Bagem/bagem/yz3.html> (10.06.2003)
- [60] ŞENOL, F., “Bulanık Mantık Kontrolcüsü”, *Gazi Ü.Lisans Tezi*, Ankara, 2000

- [61] AKDEMİR, M., “Indirect Adaptive Fuzzy Control For A Tank Using Gradient And RLS Methods”, OGÜ Lisans Tezi, Eskişehir, 2001
- [62] http://www.hkmo.org.tr/yayin/odadergi/s87/bulanik_mantik.htm, 10.06.2003
- [63] YEN, J., LANGARI, R., ZADEH, L., “Industrial Applications Of Fuzzy Logic And Intelligent Systems”, IEEE Press, New York, 1995
- [64] CHEN, G., PHAM, T.T., “Introduction To Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems”, CRC Press, Florida, 2000
- [65] www.bumat.itu.edu.tr/dokuman_BULANIK_KuMELER.doc (10.06.2003)
- [66] YEN, J., LANGARI, R., “Fuzzy Logic Intelligence, Control, and Information”, Prentice Hall, New Jersey, 1999
- [67] NGUYEN, H.T., WALKER, E.A., “A First Course In Fuzzy Logic”, Chapman&Hall/CRC, New York, 1999
- [68] <http://eros.science.ankara.edu.tr/~ozbek/bulanik-1.htm> (08.08.2002)
- [69] PIEGAT, A., “Fuzzy Modeling and Control”, Physica-Verlag Press, New York, 2001
- [70] AL, M., “Determination Of Fuzzy Sets By Using Genetic Algorithms”, OGÜ Lisans Tezi, Eskişehir, 1998
- [71] NEGOTIA, C.V., RALESCU, D., “Simulation, Knowledge-Based Computing, and Fuzzy Statistics”, Van Nostrand Reinhold Company Inc., New York, 1987
- [72] İTÜ Bulanık Mantık ve Teknoloji Kulübü, “Bulanık Mantık Kurs Notları”, İstanbul, 2003
- [73] RAGHUNATHAN, V., SCHURGERS, C., PARK, S., SRIVASTANA, M.B., Energy-aware wireless microsensor networks, IEEE Signal Processing Magazine 19 (2) (2002) 40–50
- [74] OYMAN, E.İ., ERSOY, C., Overhead energy considerations for efficient routing in wireless sensor Networks, Computer Networks 46, 2004, 465–478
- [75] RAPPAPORT, T.S., Wireless communications: Principles and Practice, Prentice Hall, 1996

- [76] HEINZELMAN, W.R., CHANDRAKASAN, A., BALAKRISHNAN, H., Energy-efficient routing protocols for wireless microsensor networks, Proc. HICSS, Hawaii, January 2000.
- [77] SEARS, R., ZEMANSKY, W., Modern Üniversite Fiziği Cilt 2 Elektrik, Çağlayan Yayınevi, 1990.
- [78] GIARDONA, S., STAJMENOVIC, I., BLAZEVIC, L., Position Based Routing Algorithms For Ad Hoc Networks: A Taxonomy, Cluwer Publishing, 103-136, 2003.
- [79] SPOONER, J.G., Motorola: New Chip will bring GPS to all, <http://news.com.com/2100-1040-959085.html>, September 2002
- [80] BERKHIN, P., (2002), Survey of Clustering Data Mining Techniques, San Jose, California, USA, Accrue Software Inc.
- [81] BİLGİN, T., (2003), Veri Madenciliginde Kümeleme Analizi Yöntemi Uygulaması, Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar ve Kontrol Eğitimi.
- [82] BOUTSINAS, B., GNARDELLIS, T., "On Distributing the Clustering Process", Pattern Recognition Letters 23, pp:999-1008, 2002
- [83] HAN, J., KAMBER, M., Data Mining Concepts and Techniques, Morgan Kauffmann Publishers Inc., 2001
- [84] JAIN, A.K., DUBES, R.C., "Algorithms for Clustering Data", Englewood Cliffs, New Jersey, 07632, Prentice Hall, 1998
- [85] JAIN, A.K., MURTHY, M.N., FLYNN, P.J., "Data Clustering: A Review", ACM Computing Surveys, pp:31-34, 1998
- [86] KARYPIS, G., HAN, E.H., KUMAR, V., "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling", IEEE Computer, pp:68-75, 1999
- [87] MERCER, D.P., "Clustering Large Datasets", <http://www.stats.ox.ac.uk/~mercer/documents/transfer.pdf> ; Erisim tarihi:13/05/2005, 2003
- [88] WITTEN, I.H., FRANK, E., Data Mining: Practical machine learning tools with Java implementations, San Francisco, Morgan Kaufmann., 1999
- [89] DEMİRALAY, M., ÇAMURCU, Y., "CURE, AGNES ve K-means Algoritmalarındaki Kümeleme Yeteneklerinin Karşılaştırılması", İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, Yıl 4, Sayı 8, Güz-2005, Sayfa 1-18

- [90] CHENG, S., Fuzzy Clustering with an Application to Scheduling, Mechanical Engineering Msc. Thesis, Concordia University, Canada, 1999.
- [91] GUHA, S., RASTOGI, R., SHIM, K., "CURE: An Efficient Clustering Algorithm for Large Databases", Information Systems 26, pp:35-58, 2002
- [92] HALKIDI, M., BATISTAKIS, Y., VAZIRGIANNIS, M., On Clustering Validation Techniques, Kluwer Academic Publishers, 2001
- [93] KAUFMAN, L., ROUSSEEUW, P.J., Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, 1990

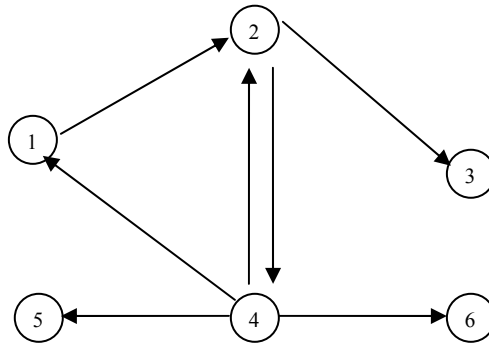
EKLER

Ek A. Graf Teorisi [55]

Yönlendirme protokolleri tasarlanırken ve analiz edilirken, graf teorisi kullanılır. Bir kablosuz ağ, gezgin düğümlerin nokta ile düğümler arası linklerin de kenarlarla tanımlandığı graflarla gösterilebilir. Bir graf $G(V,E)$ şeklinde iki küme ile ifade edilir. Burada V , noktalar (vertex) yani düğümler kümesi, E ise kenarlar (edge) kümesidir. Graf, noktalar ile bu noktaları birleştiren hatlardan oluşur. Grafın büyüklüğünü, düğümlerin sayısı $|V|$ ve kenarların sayısı $|E|$ belirler. Aşağıda Graf teorisinde öne çıkan temel tanımlar verilecektir.

Yönlendirilmiş Graflar (Directed Graphs): Yönlendirilmiş bir graf $e \in E$ olmak üzere, sıralı (u,v) çiftleri ve aralarındaki e kenarlarından oluşan graftır $(u,v \in V)$. Burada u başlangıç düğümü, v varış düğümüdür. Ayrıca $u \neq v$ dir.

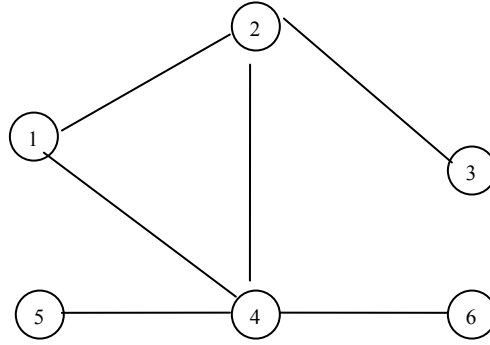
Örnek olarak; $V=\{1,2,3,4,5,6\}$ $|V|=6$ ve $E=\{(1,2),(2,3),(2,4),(4,1),(4,2),(4,5),(4,6)\}$ $|E|=7$ olarak verilen V ve E kümelerinin gösterimi Şekil A.1'deki gibidir.



Şekil A.1. 6 düğümlü yönlendirilmiş graf

Yönlendirilmemiş Graflar (Unidirected Graphs): $e \in E$ kenarı sıralanmamış $(u,v)=(v,u)$ çiftlerinden oluşan graflardır.

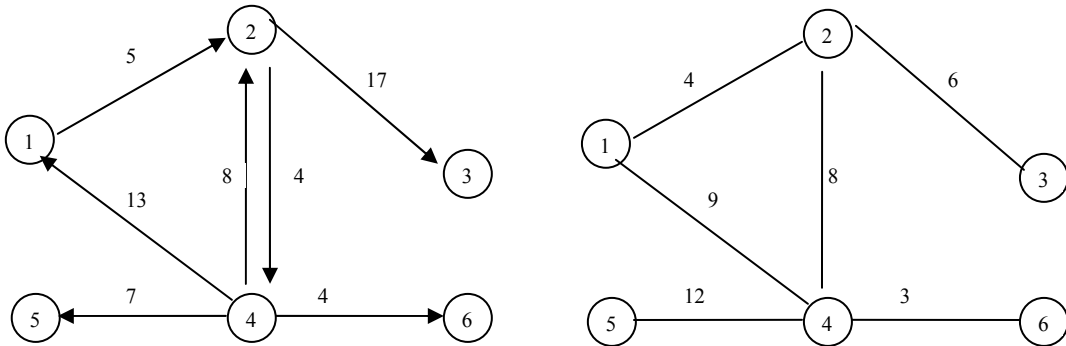
$V=\{1,2,3,4,5,6\}$ $|V|=6$ ve $E=\{(1,2),(2,3),(2,4),(4,1),(4,5),(4,6)\}$ $|E|=6$ olarak verilen graf Şekil A.2’de verilmiştir.



Şekil A.2. 6 düğümlü yönlendirilmemiş graf

Bir düğümün derecesi: Yönlendirilmemiş bir grafta bir düğümün derecesi, onun kenarlarının sayısıdır. Yönlendirilmiş bir grafta, düğümün dış derecesi; ondan ayrılan kenarların sayısı, iç derecesi de ona yönelen kenarların sayısıdır. Yönlendirilmemiş graf için Şekil A.2’de 2 numaralı düğümün derecesi 3’tür. Yönlendirilmiş grafta Şekil A.1’de 2 numaralı düğümün iç derecesi 2, dış derecesi 2’dir.

Ağırlıklandırılmış graflar: Ağırlıklandırılmış bir grafta, her kenar bir ağırlık değerine sahiptir. Bu genellikle $w: E \rightarrow R$ şeklinde bir fonksiyonla verilir. Şekil A.3’de ağırlık değerleri verilmiş yönlendirilmiş ve yönlendirilmemiş graflar verilmiştir.



Şekil A.3. Ağırlık değerli yönlendirilmiş ve yönlendirilmemiş graf

Bağlı (Connected) graflar: Bir yönlendirilmemiş grafta, bir düğümden diğer düğüme bir sıra izlenerek gidilebiliyorsa veya düğümler bir yol ile bağlılarsa bu grafa bağlı graf denir. Bir yönlendirilmiş grafta, bir düğümden diğer düğüme yönlendirilmiş bir yol varsa, bu grafa tam bağlı graf denir. Eğer $|E| \approx |V|$ (düğüm sayısı kenar sayısına yakın) ise seyrek bağlı graf, $|E| \approx |V|^2$ (düğüm sayısı kenar sayısına yakın) ise yoğun bağlı graf denir.

Graf teorisi ve graflar birçok bilim dalında modelleme amacıyla kullanılmaktadır. Coğrafyada, yerleşim yerleri ve aralarındaki mesafelerin gösteriminde, hidrolojide su kaynaklarının gösteriminde, elektrikte barajların ve dağıtım şebekelerinin gösteriminde kullanımı örnek olarak verilebilir. Bilgisayar ağlarında ise, kablolu ve kablosuz ağlar graflarla gösterilir. Özellikle OSI modelinin ağ katmanının incelenmesinde ve ağdaki düğümlerin yönlendirme algoritmaları açısından gösteriminde graflara başvurulur.

Ek B. Bulanık Mantık

Bulanık mantık, insan davranışlarına benzer bir şekilde mantıksal uygulamalarla, bilgisayarlara yardım eden bir bilgisayar mantık devrimidir. Bulanık mantığın endüstride kullanımı verimliliği artırır, daha uygun üretim sağlar, zamanın çok önemli olduğu günümüzde zamandan tasarruf ve ekonomik açıdan fayda getirir. Birçok uygulama alanından biri olan kontrol mühendisliğinde, bulanık mantık kullanılarak tasarlanan denetleyiciler, genellikle matematik modelleri zor türetilen ya da bilinen yöntemlerle denetlendiğinde verimli sonuç alınamayan sistemlerde kullanılır. Bulanık mantık kavramını basit bir şekilde anlamak için, ‘biraz sıcak’, ‘hemen hemen doğru’, ‘çok hızlı’ vs. cümlelerine bakılacak olursa, bu cümlelerin matematiksel açıdan bir durum ifade etmemelerine karşın, bir problemi çözme açısından günlük hayatta kullanılan ve sıkça karşılaşılan örnekler olduğu görülür. Bulanık mantık bir insanın anlayabileceği ve çözüme ulaştırabileceği şekilde sistemlerin ya da cihazların çalışmasına izin verir. Kelime anlamı olarak, belirsiz bir durum içeriyor gibi gözükse de, matematiksel uygulamalarda oldukça kullanışlı olmaktadır [56].

Bulanık mantığın en önemli özelliklerinden biri nesnel olmayıp, kişisel olmasıdır. Ortaya çıkan bu çoklu mantık içinden sadece birinin seçilmesiyle, ‘Aristo Mantığı’ adı verilen ve olayın doğruluğu ya da yanlışlığıyla ilgilenen mantığın uygulamalarda tekrar kullanılmasına ihtiyaç vardır. İşte bu şartlanmanın sonucunda gerçek hayatta çoklu mantık dışlanmış ve ikili mantığa göre sınıflandırmaya gidilmiş, bu durum nedeniyle de bir takım uygulamalarda yetersiz kalınmıştır. Çünkü ‘Aristo Mantığı’nda yapılan bir iş ya doğrudur ya da yanlıştır. Bunların bir karışımı yani kısmen doğru, kısmen de yanlış olamaz. Bulanık mantık bu durumu gideren ve çözüm arayışında, özellikle de modellemede, bireyin daha aktif kullanılmasına izin veren bir metottur. Aristo Mantığı’nın en fazla kullanıldığı sahalar içinde mühendislik konuları gelir. Matematik, klasik fizik ve kimya ilkeleri de Aristo mantığına göre gelişmiştir. Ancak doğadaki olayların incelenmesinde, durum tamamen bulanık dünya ve mantığa göre olması gerekirken, yapılan kabul, varsayım, idealleştirme ve lineerleştirme ile olay, insan aklının şartlı olarak ikili mantığa göre algılayabileceği seviyeye getirilir. 1930’larda ünlü Amerikan filozofu Max Black tarafından

belirsizliđi açıklayıcı öncü kavramlar geliştirilmiş olsa da, 1965’de Zadeh tarafından yayınlanan makale modern anlamda belirsizlik kavramının değerlendirilmesinde önemli bir nokta olarak kabul edilir. Zadeh, bu makalede, kesin olmayan sınırlara sahip nesnelere oluşturduđu bulanık küme teorisini ortaya koymuştur. Zadeh’in bu makalesinin önemi sadece ihtimaller teorisine karşı duruşu ile ilgili deđil, ayrıca ihtimaller teorisinin temelini oluşturan Aristo mantığına karşı da bir meydan okumadır. Zadeh’in “bulanık küme” kavramı, klasik sistem kuramının matematiksel yöntemlerinin gerçek dünyadaki pek çok sistemde, özellikle de işin içine insanları alan, kısmen karmaşık sistemlerde yetersiz kalmasından ortaya çıkmıştır. Zadeh, ‘uzun, kırmızı, durađan’ gibi yüklemelerin ikili üyelik fonksiyonuyla ifade edilen klasik kümeler yerine, dereceli üyelik fonksiyonuyla ifade edilen bulanık kümelerle tanımlamasını önermiştir. Bulanık mantığın temelde sağladığı avantajlar aşağıda sıralanmıştır [57,58]:

- İnsan düşünce sistemine ve tarzına yakındır.
- Uygulamasında mutlaka matematiksel bir modele gereksinim duymaz.
- Yazılımın basit olması nedeniyle, sistem daha ekonomik olarak kurulabilir.
- Bulanık Mantık kavramını anlamak kolaydır.
- Üyelik değerlerinin kullanımı sayesinde, diđer kontrol tekniklerine göre daha esnektir.
- Kesinlik arz etmeyen bilgilerin kullanılması söz konusudur.
- Doğrusal olmayan fonksiyonların modellenmesine izin verebilir.
- Sadece uzman kişilerin tecrübelerinden faydalanılarak, kolaylıkla bulanık mantığa dayalı bir modelleme ya da sistem tasarlanabilir.
- Geleneksel kontrol teknikleriyle uyum halindedir.
- İnsanların iletişimde kullandıkları sözel ifadelerin bulanık mantıkta kullanımı ile daha olumlu sonuçlar çıkmaktadır.

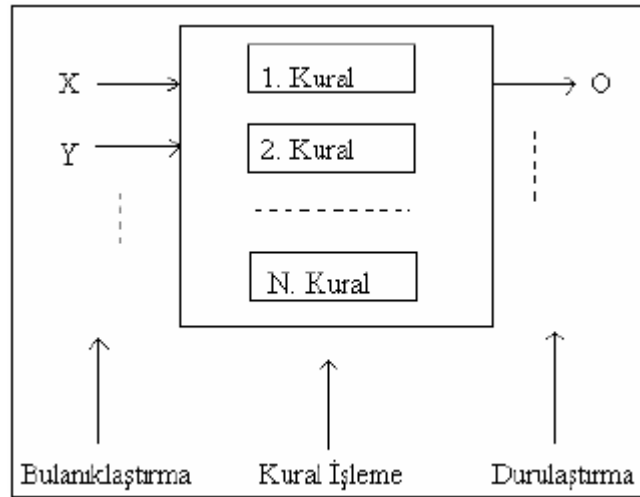
İsminin insanlarda çağrıştırdığının aksine bulanık mantık, belirsiz ifadelerle yapılan, belirsiz işlemler deđildir. Gelişmiş bir olasılık hesaplama yöntemi de deđildir. Aslında modelleme aşamasında deđişkenler ve kuralların esnek bir şekilde belirlenmesidir. Bu esneklik asla rastgelelik ya da belirsizlik içermez. Nasıl bir lastik içinde bulunduđu duruma göre şeklini deđiştirirken bütünlüğünü ve yapısını koruyabilirse, bir bulanık model de deđişen koşullara deđişen cevaplar verirken

özündeki yapıyı muhafaza eder. 1980'den sonra bulanık sistemin; elektrikli süpürgeler, çamaşır makineleri, asansörler, metro ve şirket işletimi gibi konularda kullanılmasında patlama olmuştur. Son yıllarda bir çok mühendislik dallarında, veri tabanlarının sözelleştirilmesinde, tele-sekreterlerin cevaplamasında ve bir çok konuda bulanık mantık bütün dünyada kullanılır hale gelmiştir [57].

Uzay araştırmaları ve havacılık endüstrisinde de kullanılmakta ve TAI (Turkish Aerospace Industries)'de araştırma geliştirme kısmında bulanık mantık konusunda çalışmalar yapılmaktadır. Ayrıca bulanık mantık, bir helikopter modelinin kontrolü, sözlü talimatla radyo kontrolü, yetersiz motor durumlarında otomatik yol girişi ve insansız helikopterlerin kontrolünde de kullanılmaktadır [59,60].

B.1. Bulanık sistemler

Bulanık bir süreç (fuzzy işlemi), genelde, üç ayrı birimden oluşmaktadır. Bu birimler; sırası ile bulanıklaştırıcı birim, kural işleme birimi, durulaştırıcı birim ve çıktı bilgileridir. Şekil B.1'de genel bir bulanık sistem yapısı gösterilmektedir [61].



Şekil B.1. Bulanık sistem yapısının genel gösterimi

Bu akış düzeninde, bulanıklaştırıcı birim, bulanık işlem sisteminin ilk birimi olarak devreye girmektedir. Kesin veya geri besleme sonuçları biçiminde bu birime giren bilgiler, burada bir ölçek değişikliğine uğrayarak bulanıklaştırılmaktadır. Başka bir deyişle; bu bilgilerin her birine bir üyelik değeri atanıp, dilsel bir yapıya

dönüştürülerek, buradan kural işleme birimine gönderilir. Kural işleme birimine gelen bilgiler, kural işleme biriminde depolanmış bir şekilde bulunan bilgi tabanına dayalı “if ... and ... then ... else” (eğer ... ise, ... olsun) gibi kural işleme bilgileri ile birleştirilir. Burada sözü edilen mantıksal önermeler, problemin yapısına göre sayısal değerlerle de kurulabilmektedir. Son adımda; problemin yapısına uygun mantıksal karar önermeleri kullanılarak elde edilen sonuçlar durulaştırıcı birime gönderilir. Durulaştırıcı birime gönderilen bulanık küme ilişkilerinde, bir ölçek değişikliği daha gerçekleştirilerek bulanık haldeki bilgilerin her biri gerçel sayılara dönüştürülür [62,63,64].

B.2. Bulanıklaştırma

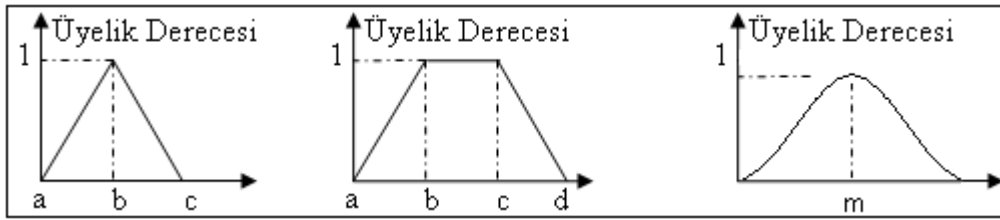
Matematikte, benzer özellikler gösteren elemanların bir arada gruplandırılmasıyla ‘küme’ adı verilen kavram oluşturulur. Klasik matematikte bir konunun bir bölümünün o kümeye ait olması gibi bir kavram düşünülmez ve kabul edilmez. Bu sınırlama, problemlerin her zaman uygun bir çözüme kavuşturulabilmesine engel teşkil etmektedir. Pratikte genel olarak, klasik küme şeklinde beliren değişim aralıklarının bulanıklaştırılması, bulanık küme, mantık ve sistem işlemleri için gereklidir. Bunun için, bir aralıkta bulunabilecek öğelerin hepsinin, 1’e eşit üyelik derecesine sahip olacak yerde, 0 ile 1 arasında değişik değerlere sahip olması düşünülür. Bu durumda, bazı öğelerin belirsizlik içerdikleri kabul edilir. Bu belirsizliklerin, sayısal olmayan durumlardan kaynaklanması halinde bulanıklıktan söz edilir [57].

Bulanıklaştırma sürecinde ele alınan üyelik fonksiyonları, problemin yapısına ve amacına uygun olmalıdır. Genel anlamda üyelik fonksiyonları sezgisel, matematik, geometrik ya da istatistiksel yaklaşımlara dayandırılabilir [62]. Bulanık kümelerin gerek üyelik derecelerinin gerekse bunların tümünü temsil edebilecek üyelik fonksiyonlarının belirlenmesinde, ilk başlayanlar tarafından bile kişisel sezgi, mantık ve tecrübelerin kullanılmasına sıkça rastlanır. Zaten pratikte bir çok sorunun üstesinden gelebilmek için bu yaklaşımlar çoğu zaman yeterlidir. Öyle olmasa bile, ilk yaklaşım olarak bu esaslara göre davranmaları faydalıdır. Üyelik fonksiyonlarının belirlenmesinde kullanılan başlıca yöntemler; a) Sezgi, b) Çıkarım, c) Mertebelenme,

d) Açılı bulanık kümeler, e) Yapay sinir ağları, f) Genetik algoritmalar, g) Çıkarımcı muhakeme gibi yaklaşımlardır [60].

Bulanık kümelerin kullanılabilirliği, farklı kavramlara uygun üyelik derecesi fonksiyonlarını oluşturabilme becerisine dayanmaktadır. En sık kullanılan fonksiyonlar kolaylık açısından “üçgen” ve “yamuktur” [65,66].

“A” bulanık kümesine ait elemanların, üçgen üyelik fonksiyonu, yamuk üyelik fonksiyonu ve çan eğrisi (Gauss) üyelik fonksiyonu ile gösterimi Şekil B.2’de verilmektedir. Üçgen üyelik fonksiyonunun matematiksel ifadesi denklem 4.1’de, yamuk üyelik fonksiyonunun matematiksel ifadesi denklem 4.2’de ve çan eğrisi üyelik fonksiyonunun matematiksel ifadesi denklem 4.3’de verilmiştir [67,68].



Şekil B.2. Üçgen, yamuk ve çan eğrisi (Gauss) üyelik fonksiyonları

$$\mu_A(x) = \mu_A(x; a, b, c) = \begin{cases} (x-a)/(x-b) & , & a \leq x < b \\ (c-x)/(c-b) & , & b \leq x \leq c \\ 0 & , & x > c \text{ veya } x < a \end{cases} \quad (\text{B.1})$$

$$\mu_A(x) = \mu_A(x; a, b, c, d) = \begin{cases} (x-a)/(b-a) & , & a \leq x < b \\ 1 & , & b \leq x < c \\ (d-x)/(d-c) & , & c < x \leq d \\ 0 & , & x > d \text{ veya } x < a \end{cases} \quad (\text{B.2})$$

$$\mu_A(x) = e^{-a(x-m)^2} \quad a > 0, m \in R \quad (\text{B.3})$$

B.3. Bulanık kural işleme

Bulanık mantıkta kurallar, “eğer ... ise, ... olsun” şeklinde koşullu durumlarla formüle edilirler [69]. Tüm girdi değişkenleri, sözel değişken değerlere çevrilerek, bulanık sonuç çıkarma adımı, güncel durum için kurallara dayandırılarak uygulanır ve çıkışta sözel değişkenlerin değerleri hesaplanır [70].

Öte yandan, bir bulanık kural, “eğer ... ise, ... olsun” şeklinde (örneğin X değeri A ise, Y değeri B olsun) sözel girdi ve çıktı terimlerine sahip olmalıdır. “eğer ...” bölümüne durum; “... olsun” bölümüne ise sonuç yada karar kısmı adı verilir [71].

Bu durumda bulanık sonuç çıkarma hesaplarının, iki bileşeni olduğu anlaşılır:

- Kümeleme: Kuralların “eğer ...” bölümlerinin hesaplanması
- Düzen: Kuralların ‘... olsun’ bölümlerinin hesaplanması

“X değeri A ise, Y değeri B olsun” örneğinde, A ve B sözel kelimelerdir ve bulanık kümelerde X ve Y değerlerinin, hangi duruma ait olduğunu gösterirler. Günlük hayatta kullanılan bazı bulanık ifadeler dayanan kurallar, örnek olması açısından aşağıda verilmektedir.

“Eğer basınç yüksekse, hacim küçük olsun.”

“Eğer bir domates kırmızı ise, o domates olgun bir domatestir.”

1973 yılında, Zadeh, bulanık değişkenler ya da sözel ifadeler ile ilgili kavramlar ortaya koymuştur. Bu kavramlardan önemli olanlardan bir tanesi bulanık nesne kavramıdır. Nesne olarak tanımlanmış sensör girdilerine örnek olarak ‘sıcaklık’, ‘yer değiştirme’, ‘hız’, ‘akış’, ‘basınç’ vb. gösterilebilir. Fark değeri olan ‘hata’ sinyalini de aynı kategoriye sokmak mümkündür.

B.4. Durulaştırma

Pratik uygulamalarda, özellikle mühendislik plan, proje ve tasarımlarında boyutlandırmalar için kesin sayısal değerlere gerek duyulmaktadır. Yapay zekâ çalışmalarındaki bulanık değişken, küme, mantık ve sistemlerin bulanık olabilecek

çıkarımlarının kesin sayılar haline dönüştürülmesi gerekir. Bulanık olan bilgilerin kesin sonuçlar haline dönüştürülmesi için yapılan işlemlerin tümüne birden durulaştırma işlemleri adı verilir. Durulaştırma işleminde kullanılan yöntemlerden ikisi yükseklik (Sugeno) ve ağırlık merkezi (Mamdani) yöntemidir.

Yükseklik durulaştırma yöntemi: Durulaştırmada kullanılan yöntemlerden bir tanesi yükseklik yöntemidir. Kullanılması için tepeleri olan çıkarım bulanık kümelerine gerek vardır. Yükseklik metoduna göre durulaştırma işlemi yapıldığında, kesin sonuç aşağıdaki ifadeden elde edilir.

$$y_0 = \frac{\sum \mu(y_i)y_i}{\sum \mu(y_i)} \quad (\text{B.4})$$

Denklem 4.4'de görülen y_i değerleri, bulanıklaştırmada oluşmuş her bir fonksiyonun üyelik derecesi en büyük olan elemanlarıdır. $\mu(y_i)$ değerleri ise, bu elemanlara karşılık gelen üyelik derecelerini belirtir.

Ağırlık merkezi durulaştırma yöntemi: Durulaştırma işlemlerinde, yaygın olarak kullanılan işlemlerden biri de ağırlık merkezi yöntemidir. Adından anlaşılacağı gibi bu yöntemle, çıkış fonksiyonunun altında kalan alanın ağırlık merkezi denklem 4.5'deki ifadeden faydalanılarak bulunur.

$$y_0 = \frac{\int \mu(y)y}{\int \mu(y)} \quad (\text{B.5})$$

Durulaştırma işleminde bu iki yöntemin dışında, üyelik derecesi en büyük olan elemanların aritmetik ortalamasına dayanan, en büyüklerin ortası yöntemi ve simetrik üyelik fonksiyonlarının bulunması halinde kullanılan ağırlıklı ortalama yöntemleri de mevcuttur [72].

Ek C. Bulanık Mantık Yöntemiyle Maliyet Değeri Belirlenmesi İşlemi Bulanıklaştırma Kuralları

Bölüm 3.5'te verilen bulanık kurallar aşağıda listelenmiştir.

1. If (Mesafe is CokYakin) and (BataryaOmru is Yuksek) and (Yogunluk is Dusuk) then (Maliyet is CokDusuk)
2. If (Mesafe is CokYakin) and (BataryaOmru is Yuksek) and (Yogunluk is Orta) then (Maliyet is Dusuk)
3. If (Mesafe is CokYakin) and (BataryaOmru is Yuksek) and (Yogunluk is Yuksek) then (Maliyet is Orta)
4. If (Mesafe is CokYakin) and (BataryaOmru is Orta) and (Yogunluk is Dusuk) then (Maliyet is Dusuk)
5. If (Mesafe is CokYakin) and (BataryaOmru is Orta) and (Yogunluk is Orta) then (Maliyet is Orta)
6. If (Mesafe is CokYakin) and (BataryaOmru is Orta) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)
7. If (Mesafe is CokYakin) and (BataryaOmru is Dusuk) and (Yogunluk is Dusuk) then (Maliyet is Orta)
8. If (Mesafe is CokYakin) and (BataryaOmru is Dusuk) and (Yogunluk is Orta) then (Maliyet is Yuksek)
9. If (Mesafe is CokYakin) and (BataryaOmru is Dusuk) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
10. If (Mesafe is Yakın) and (BataryaOmru is Yuksek) and (Yogunluk is Dusuk) then (Maliyet is Dusuk)
11. If (Mesafe is Yakın) and (BataryaOmru is Yuksek) and (Yogunluk is Orta) then (Maliyet is Dusuk)
12. If (Mesafe is Yakın) and (BataryaOmru is Yuksek) and (Yogunluk is Yuksek) then (Maliyet is Orta)
13. If (Mesafe is Yakın) and (BataryaOmru is Orta) and (Yogunluk is Dusuk) then (Maliyet is Dusuk)
14. If (Mesafe is Yakın) and (BataryaOmru is Orta) and (Yogunluk is Orta) then (Maliyet is Orta)

15. If (Mesafe is Yakın) and (BataryaOmru is Orta) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)
16. If (Mesafe is Yakın) and (BataryaOmru is Dusuk) and (Yogunluk is Dusuk) then (Maliyet is Orta)
17. If (Mesafe is Yakın) and (BataryaOmru is Dusuk) and (Yogunluk is Orta) then (Maliyet is Yuksek)
18. If (Mesafe is Yakın) and (BataryaOmru is Dusuk) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)
19. If (Mesafe is Orta) and (BataryaOmru is Yuksek) and (Yogunluk is Dusuk) then (Maliyet is Orta)
20. If (Mesafe is Orta) and (BataryaOmru is Yuksek) and (Yogunluk is Orta) then (Maliyet is Orta)
21. If (Mesafe is Orta) and (BataryaOmru is Yuksek) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)
22. If (Mesafe is Orta) and (BataryaOmru is Orta) and (Yogunluk is Dusuk) then (Maliyet is Orta)
23. If (Mesafe is Orta) and (BataryaOmru is Orta) and (Yogunluk is Orta) then (Maliyet is Orta)
24. If (Mesafe is Orta) and (BataryaOmru is Orta) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)
25. If (Mesafe is Orta) and (BataryaOmru is Dusuk) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
26. If (Mesafe is Orta) and (BataryaOmru is Dusuk) and (Yogunluk is Orta) then (Maliyet is Yuksek)
27. If (Mesafe is Orta) and (BataryaOmru is Dusuk) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
28. If (Mesafe is Uzak) and (BataryaOmru is Yuksek) and (Yogunluk is Dusuk) then (Maliyet is Orta)
29. If (Mesafe is Uzak) and (BataryaOmru is Yuksek) and (Yogunluk is Orta) then (Maliyet is Orta)
30. If (Mesafe is Uzak) and (BataryaOmru is Yuksek) and (Yogunluk is Yuksek) then (Maliyet is Yuksek)

31. If (Mesafe is Uzak) and (BataryaOmru is Orta) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
32. If (Mesafe is Uzak) and (BataryaOmru is Orta) and (Yogunluk is Orta) then (Maliyet is Yuksek)
33. If (Mesafe is Uzak) and (BataryaOmru is Orta) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
34. If (Mesafe is Uzak) and (BataryaOmru is Dusuk) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
35. If (Mesafe is Uzak) and (BataryaOmru is Dusuk) and (Yogunluk is Orta) then (Maliyet is Yuksek)
36. If (Mesafe is Uzak) and (BataryaOmru is Dusuk) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
37. If (Mesafe is CokUzak) and (BataryaOmru is Yuksek) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
38. If (Mesafe is CokUzak) and (BataryaOmru is Yuksek) and (Yogunluk is Orta) then (Maliyet is Yuksek)
39. If (Mesafe is CokUzak) and (BataryaOmru is Yuksek) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
40. If (Mesafe is CokUzak) and (BataryaOmru is Orta) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
41. If (Mesafe is CokUzak) and (BataryaOmru is Orta) and (Yogunluk is Orta) then (Maliyet is CokYuksek)
42. If (Mesafe is CokUzak) and (BataryaOmru is Orta) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
43. If (Mesafe is CokUzak) and (BataryaOmru is Dusuk) and (Yogunluk is Dusuk) then (Maliyet is Yuksek)
44. If (Mesafe is CokUzak) and (BataryaOmru is Dusuk) and (Yogunluk is Yuksek) then (Maliyet is CokYuksek)
45. If (Mesafe is CokUzak) and (BataryaOmru is Dusuk) and (Yogunluk is Orta) then (Maliyet is CokYuksek)

Ek D. Tasarsız Ağlarda Enerji Tüketimi

Tasarsız kablosuz ağlar, sürekli değişen topolojiye sahip oldukları için, kendilerine enerji sağlayan kaynaklar da hareketli olmak zorundadır. Hareketli enerji kaynakları bataryalar veya pillerdir. Belirli bir besleme kapasitesine sahip olan bu kaynakların ağdaki düğümler tarafından ekonomik bir biçimde kullanılması gereklidir. Bu bölümde kablosuz tasarsız ağlarda düğümlerin enerji tüketimleri ele alınmıştır. Ayrıca, geliştirilen KTMYA algoritmasına göre, düğümlerin sahip olması gereken GPS (Global Positioning System) cihazlarının enerji tüketimine etkileri üzerinde durulmuştur.

Anma değerleri verilen batarya veya pillerin paket iletme kapasitelerinin analizi yapıp, örnek besleme kaynakların paket alma ve gönderme kapasiteleri belirlenmiştir.

D.1 Düğümlerin enerji tüketimleri

Düğümlerin harcadıkları enerjiler; ayakta kalma enerjisi, gönderme enerjisi, alma enerjisi ve hesaplama enerjisidir.

Ayakta kalma enerjisi (E_a): Düğümün açık olduğu ve çevresinde bulunan (kapsama alanı içindeki) diğer düğümlerden haberdar olmak için harcadığı enerjidir [73].

Gönderme enerjisi (E_t): Bir veriyi herhangi bir mesafe uzaklıktaki bir düğüme göndermek için harcanan enerjidir. Tüketilen gönderme enerjisi, göndericinin gücüne, gönderme mesafesine, veri paketinin büyüklüğüne ve veri transfer hızına bağlıdır.

Alma enerjisi (E_r): Düğümün bir veriyi almak için harcadığı enerjidir. Alma enerjisi mesafeden bağımsızdır. Alma süresince harcanır. Dolayısıyla veri transfer hızına bağımlıdır.

Hesaplama Enerjisi (E_c): Düğüm içindeki işlemci tarafından, yönlendirme ve diğer işlemler için harcanan enerjidir. Hesaplama enerjisi tüketimi, diğerlerine göre oldukça düşüktür.

Bir düğümün toplam enerji tüketimi denklem D.1’de verilmiştir.

$$E_{toplam} = E_a + E_t + E_r + E_c \quad (D.1)$$

Ayakta kalma enerjisi (E_a) bütün yönlendirme algoritmaları için aynı olacağından dolayı algoritmalar arası karşılaştırma işlemlerinde dikkate alınmaz. Hesaplama enerjisi ise alma ve gönderme enerjisi yanında çok küçük kalacağı için ihmal edilir [73].

[74]’e göre gönderme gücü $\frac{1}{d^\alpha}$ ile doğru orantılı olarak azalır. Burada d , düğümler arasındaki mesafe ve α , ortamla ilgili bir sabittir. Ortamda bulunan engellere göre α değeri 2 ile 4 arasında değişir. Engellerin bulunmadığı ortamlarda α , genellikle 2 olarak alınır.

Denklem D.2’deki güç ifadesine göre:

$$P = \frac{T}{d^\alpha} \quad (D.2)$$

P : Alınan Güç, T : Gönderilen Güç, α : Ortam engellerine bağlı sabit ve d : Alıcı ile gönderici arasındaki mesafedir [75].

Denklem D.2’deki ifadenin solu ile sağı aynı fiziksel büyüklükte değildir. Yani sol tarafın birimi Watt iken, sağ tarafın birimi Watt/metre’dir. Bu yüzden d , bağıl bir mesafedir. Yani düğümler arası gerçek mesafe, birim mesafelere bölünmüştür. $d < 1$ için, alınan güç gönderilen güçten daha büyüktür. Bu aslında fiziksel olarak imkânsızdır. Böylece bu model diğer modeller tarafından reddedilmiştir.

[76]'da önerilen “simple radio model’e göre, bir düğüm gönderme alma için $E_{elec}=50$ nj/bit güç tüketir. Gönderici ile alıcı arasında 1-bit’in transfer edilebilmesi için Eelec enerjisi tüketilir. d , düğümler arası mesafe olmak üzere d^2 enerji kaybı olduğu varsayıldığında, gönderici düğümün gönderme yükselteci denklem D.3’de verilen E_{ty} enerjisini tüketir.

$$E_{ty} = E_{amp} \cdot d^2 \quad (D.3)$$

Burada $E_{amp}=100$ $pJ/bit/m^2$ dir. Böylece d mesafesine 1-bit göndermek için

$$E_{elec} + E_{amp} \cdot d^2 \quad (D.4)$$

enerji harcanırken, bu biti almak için E_{elec} enerjisi harcanır. Bu durumda;

$$E_r = E_{elec} \quad (D.5)$$

$$E_t = E_{elec} + E_{amp} \cdot d^2 \quad (D.6)$$

D.2 Besleme kaynaklarının paket alma-gönderme kapasiteleri

Bir yük depolama elemanında depolanan yükün ifadesi denklem D.7’de verilmiştir [77].

$$Q = I \cdot t \quad (D.7)$$

Denklem D.7’deki ifadede Q yükünün birimi Coulomb’dur. Coulomb’un SI birim sistemindeki karşılığı [77];

$$Coulomb = Amper \cdot saniye \quad (D.8)$$

Yük depolayan bir elemanın enerjisi aşağıda verilmiştir.

$$W = P \cdot t \quad (D.9)$$

Denklem D.9'deki ifadede W : Enerji, P : Güç ve t : Zaman olarak verilmektedir [77]. Enerjinin SI birim sistemindeki birimleri;

$$(Joule = Watt \cdot Saniye) \text{ veya } Joule = Volt \cdot Amper \cdot Saniye \quad (D.10)$$

Denklem D.8 ve D.10'daki birimler kullanılarak enerji ifadesi denklem D.11'deki gibi yazılabilir.

$$W = V \cdot Q \quad (D.11)$$

Denklem D.10'daki ifadede W : *Joule* olarak enerji, V : *Volt* olarak gerilim ve Q : *Amper \cdot Saniye* olarak yükür.

Batarya ve pillerin anma elektriksel büyüklükleri yük ve gerilimdir. Genel olarak üreticiler pil ve bataryaların üzerine yük değerini mAh (mili amper saat) ve gerilim değerini volt olarak yazmaktadır. Bu durumda, verilen yük biriminin As (amper saniyeye dönüştürülmesi gerekmektedir. mAh birimini, As birimine dönüştürmek için; $10^{-3} \times 3600 = 3,6$ ile çarpılması gereklidir.

Örnek olarak anma değerleri 10,8 V ve 4400 mAh olarak verilen bir dizüstü bilgisayar bataryasını ele alalım. Bu bataryanın enerjisi denklem D.11 kullanılarak hesaplanırsa:

$$E = 10,8 \cdot 4400 \cdot 3,6 = 171072 \text{Joule} \text{ olarak bulunur.}$$

Bir başka örnek olarak, AAA ölçüsündeki anma değerleri 1,2V ve 1000mAh olan pili ele alalım. Bu pilin sağlayabileceği enerji:

$$E = 1000 \cdot 1,2 \cdot 3,6 = 4320 \text{Joule} \text{ olarak bulunur.}$$

100 metre mesafeye 64 Byte veri paketi göndermek için harcanacak enerji denklem D.3 kullanılarak aşağıdaki gibi hesaplanır:

$$E_t = 100 \cdot 10^{-12} \cdot d^2 \cdot bit = 100 \cdot 10^{-12} \cdot 100^2 \cdot 64 \cdot 8 = 5,12 \cdot 10^{-4} \text{ Joule}$$

Yukarıda enerjisi hesaplanan dizüstü bilgisayar bataryası ile yaklaşık 334 milyon paket gönderilebilir. AAA ölçüsündeki pil ile yaklaşık 8,5 milyon paket gönderilebilir.

Bir düğümün 1 paket almak için harcayacağı enerji denklem D.5 kullanılarak:

$$E_r = 50 \text{ nJoule} = 50 \cdot 10^{-9} \text{ Joule}$$

Dizüstü bilgisayar, $3,42 \cdot 10^{12}$ paket, AAA pil, 86 milyar paket alabilir.

Geliştirilen algoritmaya göre üzerinden paket ileten bir düğüm gönderme enerjisi ve alma enerjisinin toplamı kadar enerji harcar.

D.3 GPS'lerin Enerji Tüketimi

Geliştirilen algoritmaya göre, her bir düğüm üzerinde konum bilgisinin belirlenebilmesi için GPS bulunması şarttır. Yukarıda verilen E_{toplam} enerjisi dışında GPS tarafından da güç tüketilir. GPS teknolojisindeki son gelişmeler sayesinde, yaygın kanı olarak bilinenin aksine, düşük güç tüketimine sahiptirler ve küçük boyutlara indirgenmiş alıcı yongaları şeklinde üretilirler [78]. Boyutlarının küçülmesi ile birlikte, PDA (Personal Digital Assistant) cihazları ve taşınabilir bilgisayarlar üzerinde tümleşik olarak bulunma imkanı doğmuştur. Ayrıca, bu chiplerin seri üretimi sonucu sahip olma maliyeti de azalmıştır. Örneğin Motorola tarafından üretilmiş GPS chipinin büyüklüğü 49 mm^2 ve fiyatı yaklaşık \$40'dır. Yine aynı chipin çalışma anında güç tüketimi 70 mW ve bekleme anında güç tüketimi 10 mW'tır [79]. Buna ek olarak birçok donanım üreticisi, GPS alıcı chiplerini varsayılan olarak bulundurma yönünde çalışmalar yapmaktadır.

Düğümelerde bulunan GPS'ler yer belirleme anında maksimum 70 mW güç harcarlar. Buna göre D.1'de verilen örnek dizüstü bilgisayar bataryası ile 4,8 milyon yer belirleme işlemi yapılabilir. Bu GPS bekleme zamanında ise 10 mW güç tüketir. Bu verilere göre; AAA ölçüsünde bir pil bulduran GPS, 120000 yer belirleme işlemi yapılabilir. 10 saniyede bir konum belirleme işlemi yapan bir GPS cihazı sürekli çalışma modunda 14 gün bir batarya ile beslenebilecek demektir.

Enerji tüketimleri, geliştirilen KTMYA algoritmasında her bir düğümün gönderdiği veya aldığı veriler göz önünde bulundurularak hesaplanıp, kalan batarya ömrünün hesaplanmasında kullanılmıştır. Hesaplama işlemleri MATLAB 7.0.1 ortamında geliştirilen benzetim yazılımında *gguc.m* ve *aguc.m* fonksiyonları ile yapılmıştır. Ek F'de fonksiyonların kodları verilmiştir. Benzetim işleminde her bir düğümün aldığı, gönderdiği ve ilettiği paketlerin sayısı belirlenmiştir. Ardından denklem D.6, D.9 ve D.11 kullanılarak harcanan enerji belirlenmiştir.

Algoritmada düğümlerin enerji tüketimlerinin belirlenmesi büyük önem arz etmektedir. Bölüm 3.5'te ayrıntıları açıklanan, yönlendirme kararı verilirken düğüm bataryalarının bir kriter olarak alındığı bulanık mantık yönteminin uygulanması sırasında, düğümlerin kalan batarya miktarları bilgisi kullanılmaktadır. Aynı zamanda algoritmaya göre, master düğümün "master düğümlüğü" devretmesi, normal düğümlerin güncelleme paketi göndermeleri de batarya ömrünü kullanan süreçlerdir.

D.4 Geliştirilen algoritmada düğümlerin enerji tüketimleri

PHRA algoritması Ek D'de verilen verilen enerji ifadeleri kullanılarak 100 saniye boyunca koşturulmuştur. 1024 Byte sabit boyutlu veri paketleriyle 10 düğümlü ve 20 düğümlü ağların kalan batarya ömürleri sırasıyla Tablo D.1 ve Tablo D.2'de verilmiştir. Tablolardaki "Bat" sütunu, düğümlerin başlangıç batarya ömürleridir.

Tablo D.1. 10 düğümlü ağın 100 saniye boyunca kalan batarya ömrü değerleri

Düğüm	Bat	10.s	20.s	30.s	40.s	50.s	60.s	70.s	80.s	90.s	100.s
1	67	66.972	66.9459	66.9347	66.9178	66.8981	66.8897	66.8585	66.8449	66.8323	66.7989
2	87	86.9805	86.9453	86.9216	86.8689	86.8595	86.8453	86.8073	86.7737	86.7516	86.7222
3	45	44.9801	44.9695	44.9408	44.9355	44.9138	44.8716	44.8364	44.8104	44.7686	44.7553
4	94	93.9909	93.9687	93.9316	93.8777	93.8584	93.8527	93.8523	93.8355	93.8135	93.8091
5	62	61.9329	61.823	61.754	61.7	61.6104	61.4659	61.3366	61.2476	61.1694	61.1003
6	96	95.9284	95.8983	95.8557	95.7537	95.6043	95.5345	95.462	95.4377	95.3251	95.2823
7	42	41.8909	41.8051	41.6637	41.5972	41.49	41.4701	41.3234	41.1699	41.0989	40.976
8	94	93.9218	93.822	93.7814	93.7057	93.6618	93.6072	93.5522	93.43	93.3511	93.2264
9	79	78.8112	78.4832	78.3305	77.9887	77.8408	77.7397	77.5398	77.2078	76.8824	76.6569
10	93	92.9632	92.9122	92.873	92.8511	92.8208	92.7977	92.7552	92.7042	92.6726	92.6597

Tablo D.2. 20 düğümlü ağın 100 saniye boyunca kalan batarya ömrü değerleri

Düğüm	Bat	10.s	20.s	30.s	40.s	50.s	60.s	70.s	80.s	90.s	100.s
1	20	19.8315	19.6603	19.5858	19.4377	19.2427	19.0325	18.9915	18.8137	18.7644	18.5768
2	3	2.8793	2.8383	2.7684	2.7055	2.6799	2.6558	2.6525	2.6069	2.5815	2.4965
3	94	93.8896	93.8262	93.727	93.6813	93.6213	93.5541	93.4672	93.4047	93.3363	93.2816
4	5	4.8129	4.5809	4.463	4.3289	4.1302	4.0653	3.9881	3.8604	3.7324	3.5715
5	17	16.7596	16.4998	16.1925	16.0037	15.7995	15.5237	15.2081	15.0118	14.6588	14.414
6	33	32.9354	32.8434	32.5983	32.3984	32.3116	32.0763	31.9117	31.7664	31.5386	31.3349
7	38	37.9194	37.8114	37.7873	37.638	37.574	37.4389	37.3958	37.2646	36.9462	36.8794
8	61	60.7865	60.5389	60.189	60.0162	59.7423	59.503	59.1388	58.8954	58.4773	58.2331
9	98	97.7262	97.3899	97.1384	96.7134	96.4644	96.1431	95.6948	95.3961	95.1294	94.8112
10	83	82.8302	82.4253	82.1371	81.806	81.182	80.3731	80.1931	79.7047	79.4494	79.2395
11	52	51.9927	51.9854	51.9692	51.8536	51.8471	51.8345	51.8187	51.8041	51.6934	51.6236
12	83	82.9854	82.9696	82.9639	82.9465	82.9351	82.9254	82.9152	82.9015	82.8913	82.8791
13	16	15.9919	15.9663	15.9566	15.9453	15.9355	15.9315	15.929	15.9124	15.9043	15.8966
14	19	18.8495	18.7319	18.5608	18.4516	18.3197	18.1776	17.991	17.869	17.473	17.3028
15	71	70.9939	70.9834	70.9789	70.97	70.9465	70.9197	70.9035	70.8881	70.8577	70.8406
16	44	43.9886	43.9822	43.9696	43.9586	43.9416	43.938	43.9274	43.9161	43.8937	43.8824
17	13	12.9818	12.9692	12.9505	12.9278	12.9246	12.9197	12.9075	12.8986	12.8783	12.871
18	12	11.9866	11.9785	11.8974	11.7406	11.697	11.6202	11.5754	11.4991	11.4813	11.4349
19	37	36.985	36.9696	36.9554	36.9262	36.9144	36.899	36.8885	36.869	36.8609	36.8504
20	45	44.9923	44.9858	44.9704	44.9562	44.9363	44.9221	44.9051	44.8921	44.8714	44.8544

Ek E.Kümeleme Algoritmaları

Kümeleme, en basit tanımıyla benzer özellik gösteren veri elemanlarının kendi aralarında gruplara ayrılmasıdır. Literatürde kümeleme analizini açıklayan birçok tanım bulunmaktadır [80,81,82,83,84,85,86,87,88]. Bu tanımlara göre her küme temsil ettiği nesnelere en iyi şekilde ifade edecek şekilde düzenlenir. Kümeleme işleminin uygulandığı veri setindeki her bir veriye nesne adı verilir. Bu nesnelere iki boyutlu düzlem üzerinde noktalarla gösterilir. Kümeleme analizi, veri indirgeme veya nesnelere doğal sınıflarını bulma gibi çeşitli amaçlarla kullanılmaktadır [80,82,85]. Kümeleme analizinin kullanıldığı sayısız uygulama alanı bulunmaktadır. Bu alanlardan en çok gündemde olanlar örüntü tanıma, veri analizi, resim tanıma, pazarlama, metin madenciliği, doküman toplama, istatistik araştırmaları, makine öğrenimi, şehir planlama, coğrafik analizler (deprem, meteoroloji, yerleşim alanları), uzaysal veritabanı uygulamaları, web uygulamaları, müşteri ilişkileri yönetimi, sağlık ve biyoloji alanında yapılan araştırmalardır [80,81,83,85].

Literatürde kullanılan sınıflandırma algoritmalarından bazıları şunlardır:

- Hiyerarşik kümeleme [89]
- K-Means kümeleme [83]
- Eksiltmeli kümeleme [90]
- Fuzzy C-Means (FCM) kümeleme [48]

E.1. Hiyerarşik Kümeleme

Hiyerarşik kümeleme yöntemine göre, her küme bir veri setindeki her bir nesnenin dizindeki bir sonraki nesnenin içinde yer aldığı nesnelere dizisidir [91]. Bu dizinin en üst seviyesinde tüm nesnelere içeren tek bir küme ve en alt seviyesinde ise ayrı noktalardan oluşan tekil kümeler yer alır. Bu iki seviye arasında kalan her seviyedeki küme, bu küme ve bu kümenin bir alt (veya bir üst) seviyesindeki kümenin birleşimidir (veya ayrışımıdır) [92].

Hiyerarşik kümeleme yönteminde genel olarak kullanılan bölümeleme algoritmaları, CURE (Clustering Using REpresentatives) ve AGNES (AGglomerative NEsting) dir.

CURE Algoritması: CURE algoritması, her kümenin sabit sayıda örneklem nokta ile temsil edildiği ve her adımda istenen küme sayısı elde edilene kadar örneklem noktaları en yakın olan kümelerin birleştirildiği aşağıdan yukarıya doğru çalışan hiyerarşik bir kümeleme algoritmasıdır. Her adımda yeni oluşturulan kümelerin örneklem noktalarını bulmak için birleşen kümelerin örneklem noktaları bir daraltma katsayısı ile çarpılır. Bu durumda algoritmanın doğru kümelenmeleri bulması üç parametrenin değerine bağlıdır: küme sayısı (k), örneklem nokta sayısı (rep_say), ve daraltma katsayısı (α). CURE algoritmasının çalışmasındaki işlem basamakları aşağıdadır [91]:

1. Her küme için sabit sayıda ve küme içinde dağınık olarak yerleşmiş c adet örneklem nokta seçilir
2. İki küme arasındaki uzaklık, bu kümelere ait örneklem noktalar arasındaki Öklit uzaklığı hesaplanarak elde edilir
3. En yakın küme çifti birleştirilir
4. Oluşan yeni kümenin örneklem noktaları bulunur. Bu işlem için yeni kümenin alt kümelerinden merkeze en yakın olan c adet nokta seçilir. Bu noktalar daraltma katsayısı α ile çarpılarak merkeze doğru yaklaştırılır
5. Küme sayısı, kümeleme algoritmasında giriş parametresi olarak verilen k değerine ulaşana kadar 2, 3 ve 4. adımlar tekrarlanır.

AGNES (AGglomerative NEsting) Algoritması: Kaufman ve Rousseeuw tarafından 1990 yılında sunulmuştur Aşağıdan yukarı doğru çalışan bir inşa yapısı izler. Başlangıçta her nesne ayrı bir küme olarak kabul edilir. Algoritmanın sonraki her adımında bu atomik kümelerden benzer özellik gösterenler birleştirilir. Herhangi bir sonlanma koşulu verilmezse kümeleme işlemi tamamlandığında bütün nesnelere tek bir kümede toplanır [92].

E.2. K-means Kümeleme

En eski kümeleme metodlarından biri olan k-means algoritmasının genel mantığı n adet veri nesnesinden oluşan bir veri setini, giriş parametresi olarak verilen k adet kümeye bölümlenektir. Amaç, gerçekleştirilen bölümlenme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası

benzerliklerinin minimum olmasını sağlamaktır. Küme benzerliği, kümenin ağırlık merkezi olarak kabul edilen bir nesne ile kümedeki diğer nesneler arasındaki uzaklıkların ortalama değeri ile ölçülmektedir [83].

K-means algoritmasının işlem basamakları şöyledir:

1. İlk küme merkezleri belirlenir. Bunun için iki farklı yol vardır. Birinci yol nesneler arasından küme sayısı k adet rastgele nokta seçilmesidir. İkinci yol ise merkez noktaların tüm nesnelerin ortalaması alınarak belirlenmesidir.
2. Her nesnenin seçilen merkez noktalara olan uzaklığı hesaplanır. Elde edilen sonuçlara göre tüm nesneler k adet kümeden kendilerine en yakın olan kümeye yerleştirilir.
3. Oluşan kümelerin yeni merkez noktaları o kümedeki tüm nesnelerin ortalama değeri ile değiştirilir.
4. Merkez noktalar değişmeye kadar 2. ve 3. adımlar tekrarlanır.

K-means algoritmasında her bir nesnenin merkez noktalara uzaklığını hesaplamak için kullanılan dört farklı formül aşağıda açıklanmaktadır [87]:

Öklit Uzaklığı - Öklit Uzaklığının Karesi (Euclidean Distance - Squared Euclidean Distance): Öklit uzaklığı ve Öklit uzaklığının karesi formülleri ile standartlaştırılmış verilerle değil, işlenmemiş verilerle hesaplama yapılır. Öklit uzaklıkları kümeleme analizine sıra dışı olabilecek yeni nesnelerin eklenmesinden etkilenmezler. Ancak boyutlar arasındaki ölçek farklılıkları öklit uzaklıklarını önemli ölçüde etkilemektedir. Öklit uzaklık formülü en yaygın olarak kullanılan uzaklık hesaplama formülüdür. Öklit ve Öklit uzaklığının karesinin formülleri sırasıyla denklem E.1 ve E.2'de görülmektedir.

$$d(x, y) = \left\{ \sum_i (x_i - y_i)^2 \right\}^{1/2} \quad (\text{E.1})$$

$$d(x, y) = \sum_i (x_i - y_i)^2 \quad (\text{E.2})$$

City-block (Manhattan) Uzaklığı : Manhattan uzaklığı boyutlar arasındaki ortalama farka eşittir. Bu ölçüt kullanıldığında farkın karesi alınmadığı için sıra dışılıkların etkisi azalır. Manhattan uzaklığının formülü denklem E.3’de görülmektedir.

$$d(x, y) = \sum_i |x_i - y_i| \quad (\text{E.3})$$

Chebychev Uzaklığı (Chebychev Distance): Chebychev uzaklığı iki nesne arasındaki mutlak maksimum uzaklığa eşittir. Chebychev uzaklığının ifadesi denklem E.4’de verilmiştir.

$$d(x, y) = \max |x_i - y_i| \quad (\text{E.4})$$

Denklem E.1, E.2, E.3 ve E.4’de verilen ifadelerdeki;

$d(x,y)$: x ve y noktaları arasındaki uzaklık, hata parametresidir.

x, y : aralarındaki uzaklık hesaplanan nesnelere uzayda temsil eden noktalardır.

K-means algoritmasının en büyük eksikliği k değerini tespit edememesidir. Bu nedenle başarılı bir kümeleme elde etmek için farklı k değerleri için deneme yanılma yönteminin uygulanması gerekmektedir [89].

E.3. Eksiltmeli kümeleme

Eksiltmeli kümeleme, bir veri seti için, küme sayısını ve küme merkezlerini tahmin etmeye yarayan hızlı bir algoritmadır. 1994 yılında Chi tarafından ortaya atılmıştır. Eksiltmeli kümeleme işlemi, alt kümeler kullanarak ve iterasyon yaparak küme merkezlerini bulur [49].

Bu yöntem Bölüm 3.6’da verilen FCM algoritmasını kullanarak küme merkezlerini ve düğümlerin aitlik derecelerini bulur. FCM’den farkı, küme sayısını algoritmanın kendisinin belirlemesidir. Algoritma bulanık mantık kullanarak ve iterasyon yaparak kaç küme merkezi bulunması durumunda daha iyi gruplama yapılacağını belirler.

Ek F. KTMYA Benzetim Modeli MATLAB Fonksiyonları

Node.m:

```
function P=node(nodenumber)

% Pkonum=[2 70 10 ;2 5 1 ; 65 50 25 ;40 30 15;140 5 30;95 55 15];
%düğüm sayısına göre düğüm alanı belirlenir. Düğüm başına yaklaşık
5000
%metrekare düşecek şekilde kare alanlar seçilir.
if nodenumber==10
Pxy=randint(nodenumber,2,[0,180]); %220
else
    if nodenumber==20
        Pxy=randint(nodenumber,2,[0,280]); %315
    else
        if nodenumber==50
            Pxy=randint(nodenumber,2,[0,400]); %500
        else
            Pxy=randint(nodenumber,2,[0,600]); %600
        end;
    end;
end;
Pz=randint(nodenumber,1,[0,10]);
Pkonum=horzcat(Pxy,Pz);
Pbatarya=randint(nodenumber,2,[1,100]);
P=horzcat(Pkonum,Pbatarya);
```

Master.m:

```
function d=master(M) %master düğüm belirleme işlemi
% Bu matrisin satır elemanlarını toplayarak merkez elemanı bulalım
[a,b]=size(M);
T=zeros(a,1);
for i=1:1:a;
for j=1:1:a;
    T(i,1)=T(i,1)+M(i,j);
end
end
T
%T matrisinin en küçük elemanını bulalım bu eleman merkez elemandır
[c,d]=min(T);
d %elemanı en merkezde olan düğümdür
n=norm(M);
%Me=zeros(a,a);
```

Fuzzy.m:

```
function G=fuzzy(M,P,t) %fuzzy kullanarak düğümler arası maliyet
değerlerini
[a,b]=size(M); %belirleme M mesafe matrisi ve t yayım
mesafesi giriş
for i=1:1:a; %parametreleridir.
    for j=1:1:a; %birbirlerinin yayılım mesafesinde olmayan
düğümler
        if M(i,j)<=t %arası mesafe inf sonsuz olarak işaretleniyor
            Me(i,j)=M(i,j)*100/t;
        else
            Me(i,j)=inf;
        end;
    end;
end
```

```

end
%Me
b=readfis('PBHRA');
G=zeros(a,a);
for i=1:1:a;
    for j=1:1:a;
        if Me(i,j)==0
            G(i,j)=0;
        else
            if Me(i,j)==inf
                G(i,j)=inf;
            else
                G(i,j)=(evalfis([Me(i,j) P(j,4) P(j,5)],b));
            end;
        end;
    end
end
end
%G

```

Fuzzyless.m:

```

function G=fuzzyless(M,P,t) %fuzzy kullanarak düğümler arası
malîyet değerlerini
[a,b]=size(M); %belirleme M mesafe matrisi ve t yayım
mesafesi giriş
for i=1:1:a; %parametreleridir.
    for j=1:1:a; %birbirlerinin yayılım mesafesinde olmayan
düğümler
        if M(i,j)<t %arası mesafe inf sonsuz olarak işaretleniyor
            G(i,j)=M(i,j);
        else
            G(i,j)=inf;
        end;
    end
end
end

```

enkisayol.m:

```

function [enkisa,v,V]=enkisayol(G)
u=1;
[p r]=size(G);
%y=input('hangi düğümden...:');
%z=input('hangi düğüme...:');
for i=1:1:p;
    for j=1:1:r;
        if (G(i,j)~=0 & G(i,j)~=inf)
            B(u,1)=i;B(u,2)=j;B(u,3)=G(i,j); u=u+1;
        end;
    end
end
end
for i=1:1:p;
    for j=1:1:r;
        [enkisa v]=grShortPath(B,i,j);
        [aa ba]=size(v);
        for rr=1:1:ba;
            V((i-1)*p+j,rr)=v(rr);
        end
    end
end
end

```

Poisson.m:

```

function [PD,hedef]=poisson(nodenummer)

```

```

PD=poissrnd(1,1,nodenumber);
hedef=zeros(1,nodenumber);
%düğüm sayısına göre poisson dağılımına göre paket üretim miktarları
ve
%üretilen paketlerin hedef düğümleri oluşturulmuştur. PD ile paket
sayı
%ları, hedef ile paketlerin hedef adresleri oluşturulmuştur.
for i=1:1:nodenumber/2;
    if PD(i)~=0
        hedef(i)=randint(1,1,[nodenumber/2,nodenumber]);
    end;
end
for i=nodenumber/2+1:1:nodenumber;
    if PD(i)~=0
        hedef(i)=randint(1,1,[1,nodenumber]);
    end;
end
end

```

move.m:

```

function P=move(Pa)
[a,b]=size(Pa);
h=randint(a,1,[0,20]);
alfa=randint(a,1,[0,360]);
alfa=alfa*pi/180;
%P2=randint(a,1,[1,2]);
P=Pa;
for i=1:1:a
    if (cos(alfa(i))==0 | sin(alfa(i))==0)
        P(i,1)=Pa(i,1);
        P(i,2)=Pa(i,2);
    else
        x=round(h(i)*cos(alfa(i)))+Pa(i,1);
        y=round(h(i)*sin(alfa(i)))+Pa(i,2);
        if x<0
            P(i,1)=Pa(i,1);
        else
            P(i,1)=x;
        end;
        if y<0
            P(i,2)=Pa(i,2);
        else
            P(i,2)=y;
        end;
    end;
end;
end;
P;

```

Gguc.m ve aguc.m:

```

function Et=gguc(mesafe,bit) %gönderici düğümün harcadığı güç
%Et=Eamp*d2
%Eamp=100pJ/bit/m^2 veya Eamp=100 pJ/(bit*m^2)
Et=100*10^(-12)*mesafe^2*bit
function Er=aguc(bit) %alıcı düğümün harcadığı güç
%Et=Eamp*d2
%Eamp=100pJ/bit/m^2
Er=50*10^(-9)*bit

```

Nrl.m:

```

function toplam=nrl(nodenum,kaysay,pause)
n=nodenum; % ağın düğüm sayısı
P=node(n);
M=mesafe(P);
d=master(M);
yayilim=100;
ptime=pause;
kaynak=kaysay; %ağda paket üretecek düğüm
sayısı(kaynak sayısı)
%kaynak=input('ağın kaynak sayısı...:');
%te=input('çalışma süresi giriniz...:');
te=100;
bb=0;
gonderilen=0; % her bir düğüm tarafından gönderilen paket
sayısı
alınan=zeros(1,n); % her bir düğüm tarafından alınan paket
sayısı
iletilen=zeros(1,n); %her bir düğümün üzerinden ilettiği paket
sayısı
for j=1:1:te;
    if mod(j,ptime)==0
        [PD,hedef]=paket(n,kaynak);
        gonderilen=gonderilen+PD;

[basar,iletilen,alınan]=basarimhesap_(M,P,yayilim,n,PD,hedef,iletile
n,alınan);
        bb(j)=basar
        P=move_(P);
        else
            [PD,hedef]=paket(n,kaynak);
            gonderilen=gonderilen+PD;

[basar,iletilen,alınan]=basarimhesap_(M,P,yayilim,n,PD,hedef,iletile
n,alınan);
        bb(j)=basar
        end;

end

bb;
basarimsonuc=mean(bb)
toplamgonderilen=sum(gonderilen);
toplamalınan=sum(alınan);
toplamiletilen=sum(iletilen);
toplam=sum(gonderilen);
%sonuc=toplamalınan/toplamgonderilen
toplam;

```

padf.m:

```

padf10_10(1)=padf(10,10,1);
padf10_10(2)=padf(10,10,10);
padf10_10(3)=padf(10,10,20);
padf10_10(4)=padf(10,10,50);
padf10_10(5)=padf(10,10,100);
padf10_5(1)=padf(10,5,1);
padf10_5(2)=padf(10,5,10);
padf10_5(3)=padf(10,5,20);

```

```
padf10_5(4)=padf(10,5,50);
padf10_5(5)=padf(10,5,100);
padf10_10
padf10_5
```

```
padf20_20(1)=padf(20,20,1);
padf20_20(2)=padf(20,20,10);
padf20_20(3)=padf(20,20,20);
padf20_20(4)=padf(20,20,50);
padf20_20(5)=padf(20,20,100);
padf20_10(1)=padf(20,10,1);
padf20_10(2)=padf(20,10,10);
padf20_10(3)=padf(20,10,20);
padf20_10(4)=padf(20,10,50);
padf20_10(5)=padf(20,10,100);
```

```
padf20_20
padf20_10
padf50_30(1)=padf(50,30,1);
padf50_30(2)=padf(50,30,10);
padf50_30(3)=padf(50,30,20);
padf50_30(4)=padf(50,30,50);
padf50_30(5)=padf(50,30,100);
```

```
padf50_20(1)=padf(50,20,1);
padf50_20(2)=padf(50,20,10);
padf50_20(3)=padf(50,20,20);
padf50_20(4)=padf(50,20,50);
padf50_20(5)=padf(50,20,100);
```

```
padf50_10(1)=padf(50,10,1);
padf50_10(2)=padf(50,10,10);
padf50_10(3)=padf(50,10,20);
padf50_10(4)=padf(50,10,50);
padf50_10(5)=padf(50,10,100);
padf50_30
padf50_20
padf50_10
```

```
padf100_30(1)=padf(100,30,1);
padf100_30(2)=padf(100,30,10);
padf100_30(3)=padf(100,30,20);
padf100_30(4)=padf(100,30,50);
padf100_30(5)=padf(100,30,100);
```

```
padf100_20(1)=padf(100,20,1);
padf100_20(2)=padf(100,20,10);
padf100_20(3)=padf(100,20,20);
padf100_20(4)=padf(100,20,50);
padf100_20(5)=padf(100,20,100);
```

```
padf100_10(1)=padf(100,10,1);
padf100_10(2)=padf(100,10,10);
padf100_10(3)=padf(100,10,20);
padf100_10(4)=padf(100,10,50);
padf100_10(5)=padf(100,10,100);
padf100_30
padf100_20
padf100_10
```

delay.m:

```

function delaymsn=padf(nodenum,kaysay,pause)
n=nodenum; % ağın düğüm sayısı
P=node(n);
M=mesafe(P);
d=master(M);
yayilim=100;
ptime=pause;
kaynak=kaysay; %ağda paket üretecek düğüm
sayısı(kaynak sayısı)
%kaynak=input('ağın kaynak sayısı...:');
%te=input('çalışma süresi giriniz...:');
te=100;
bb=0;
gonderilen=0; % her bir düğüm tarafından gönderilen paket
sayısı
alınan=zeros(1,n); % her bir düğüm tarafından alınan paket
sayısı
iletilen=zeros(1,n); %her bir düğümün üzerinden ilettiği paket
sayısı
ortdugum=0; %Kaynakyan hedefe paket giderken kaç düğüm
üzerinden geçtiğini tutacak
gondericisayisi=0;
for j=1:1:te;
    if mod(j,ptime)==0
        [PD,hedef]=paket(n,kaynak);
        for nn=1:1:n;
            if PD(nn)~=0
                gondericisayisi=gondericisayisi+1;
            end;
        end
        gonderilen=gonderilen+PD;

[basar,iletilen,alınan,ortdugum]=basarimhesap_(M,P,yayilim,n,PD,hede
f,iletilen,alınan,ortdugum);
        bb(j)=basar;
        P=move_(P);
    else
        [PD,hedef]=paket(n,kaynak);
        for nn=1:1:n;
            if PD(nn)~=0
                gondericisayisi=gondericisayisi+1;
            end;
        end
        gonderilen=gonderilen+PD;

[basar,iletilen,alınan,ortdugum]=basarimhesap_(M,P,yayilim,n,PD,hede
f,iletilen,alınan,ortdugum);
        bb(j)=basar;
    end;

end
delaymsn=3.9*2*ortdugum

```

enerji.m:

```

function toplamenerji=enerji(nodenum,kaysay,pause)
n=nodenum; % ağın düğüm sayısı
P=node(n);
Pilk=P;

```

```

Pg=P(:,4);
M=mesafe(P);
d=master(M);
yayilim=100;
ptime=pause;
kaynak=kaysay; %ağda paket üretecek düğüm
sayısı(kaynak sayısı)
%kaynak=input('ağın kaynak sayısı...:');
%te=input('çalışma süresi giriniz...:');
te=100;
bb=0;
gonderilen=0; % her bir düğüm tarafından gönderilen paket
sayısı
alınan=zeros(1,n); % her bir düğüm tarafından alınan paket
sayısı
iletilen=zeros(1,n); %her bir düğümün üzerinden ilettiği paket
sayısı
almagucu=zeros(1,n);
gondermegucu=zeros(1,n);
for j=1:1:te;

    if mod(j,ptime)==0
        [PD,hedef]=paket(n,kaynak);
        gonderilen=gonderilen+PD;

[basar,iletilen,alınan,almagucu,gondermegucu,Pg]=basarimhesap_(M,P,y
ayilim,n,PD,hedef,iletilen,alınan,almagucu,gondermegucu,Pg);

        bb(j)=basar;
        P=move_(P);
        else
            [PD,hedef]=paket(n,kaynak);
            gonderilen=gonderilen+PD;

[basar,iletilen,alınan,almagucu,gondermegucu,Pg]=basarimhesap_(M,P,y
ayilim,n,PD,hedef,iletilen,alınan,almagucu,gondermegucu,Pg);
            bb(j)=basar;
        end;

end
toplamenerji=sum(almagucu+gondermegucu);
Pilk;
Pg;
gonderilen;
alınan;
iletilen;

```


ÖZGEÇMİŞ

1974 yılında Nevşehir'in Derinkuyu ilçesinde doğdu. İlk ve orta öğrenimini Nevşehir'de tamamladı. 1991 yılında başladığı Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik Elektronik Mühendisliği Bölümü'nden Ocak 1996'da mezun oldu. Aynı yıl başladığı Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği ABD yüksek lisans programından Aralık 1999'da mezun oldu. Ocak 2005'de Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği EABD'da doktora programına başlamıştır. Halen Düzce Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Eğitimi Bölümü'nde öğretim görevlisi olarak çalışmaktadır.