

# Ad Hoc Ağları İçin Kuyruk Ağ Analizi Ve Yapay Arı Kolonisi Algoritmalarının Birleştirilerek Routing Probleminin Simülasyonu

<sup>1</sup>Hakan Üçgün <sup>\*2</sup>Mustafa Danacı

<sup>1</sup>Mühendislik Fakültesi, Bilgisayar Mühendisliği Bilecik Şeyh Edebali Üniversitesi, Türkiye

<sup>\*2</sup>Mühendislik Fakültesi, Bilgisayar Mühendisliği Erciyes Üniversitesi, Türkiye

## Özet

Mobil Gezgin Ağlar (MANET) alanı, son yıllarda çok popüler hale gelmiş ve araştırmacıların önemli bir bölümünün ilgisini kazanmıştır. MANET, sabit altyapı olmadan çalışabilir ve ağ topolojisinde hızlı değişimler yaşayabilir. MANET de zaman içinde topolojilerin değerleri değişebilir ve bu değişimler grafiksel ve biçimsel olarak ele alınabilir. MANET 'in performansını değerlendirmek için ana yöntem olarak simülasyon uygulamaları kullanılmaktadır. Başarıyla sonuçlanan uygulama, Yapay Arı Kolonisi Optimizasyonu ve Ağ Gecikme Analizi algoritmalarının birlikte kullanılarak MANET için yeni adaptif bir dinamik yönlendirme algoritması sunmaktadır.

Ad Hoc İsteğe Bağlı Mesafe Vektörleri olarak bilinen, routing protokollerinden en çok öne çıkan bir yapı ile oluşturulacak hibrid simülasyon algoritması arasındaki karşılaştırma ve çalışma ile ilgili yapılan önceki uygulamalarda kullanılan algoritmalar ile bir kıyaslama çalışması yapılmaktadır. Farklı düğüm yoğunluğu ve duraklama süreleri ile çeşitli simülasyon senaryoları araştırılmıştır. Oluşturulan yeni algoritma da, artan duraklama süresini veya azalan düğüm yoğunluğu gibi belirli koşullar altında farklı sonuçlar vermektedir.

**Anahtar Kelimeler:** Yapay Arı Koloni Algoritması (ABC), Mobile Gezgin Ağlar (MANET), Kuyruk Ağ Analizi, Routing Problemi

## Abstract

In the past few years there has been a growing interest in the field of Mobile Ad hoc Networks (MANETs). MANETs can run independently of the fixed infrastructure. Additionally, they are able to accommodate fluctuations in the network topology. In MANETs, the values of the topologies may change in time and the changes can be represented as graphs. Simulation applications are the widely adopted method for evaluating the performance of MANETs. In the resulted successfully study the Artificial Bee Colony Optimization (ABC) algorithms together with network delay analysis algorithms, this study offers a new adaptive and dynamic routing algorithm for MANETs.

A comparative study between Ad hoc On-Demand Distance Vector (AODV) which is a well known router protocol and the proposed hybrid simulation algorithm is propounded. Different node densities and pause times are used to scrutinize different simulation scenarios. The proposed algorithm produces different results in different conditions as in increasing the pause time and decreasing node density.

**KEYWORDS:** Artificial Bee Colony Optimization (ABC), Mobile Ad hoc Network (MANET), Queuing Network Analysis, Routing Problem

## 1. Giriş

Çoklu ortamlardaki gezgin kullanıcılara gerekli hizmetleri vermeyi amaçlayan gezgin ağ teknolojileri piyasaya sürüldüğü 1970 yılından itibaren bilişim alanında gitgide daha büyük bir ilgi çekmeye başlamıştır [1]. Telsiz erişimini kullanarak haberleşen gezgin ağları iki bölüm de inceleyebiliriz; Çalışabilmesi için önceden ayarlanmış bir altyapıya ihtiyaç duyan "hücresele ağlar" ve çalışabilmesi için önceden ayarlanmış bir altyapıya ihtiyaç duymayan ad hoc (gezgin) ağlarıdır. Kablosuz olarak merkezi bir yapıya sahip olmayan, istemcilerin birbirleri arasındaki haberleşmesi şeklinde gerçekleşen iletişim teknolojileri genel olarak "Ad hoc ağları" şeklinde adlandırılır.

Ad hoc ağları kendi kendilerine yapılabilmekte, geniş bir alanda kolayca yayılmakta ve sabit bir altyapıya gerek duymamaktadır. Belirlenen alan içerisinde her yöne doğru hareket edebilmekte ve kısıtlı olanaklarına rağmen birbirleri arasındaki iletişimi devam ettirmek için aralarında sürekli işbirliği halinde bulunan telsiz düğümlerden oluşmaktadır. Ad hoc ağlarındaki düğümler, sabit bir şekilde durabildikleri gibi sürekli hareket halinde de olabilirler. Geliştirilen yeni mimarinin amacı, sabit altyapılı olan gezgin ağlara göre, kullanıcılarına esneklik, hareketlilik sağlamak ve ağ yönetimini daha fazla kolaylaştırmaktır. Bu sayılan özellikler, hücresele ağlarda kullanılan fiziksel altyapının yerine ad hoc düğümlerinin arasında dinamik iletişim omurgaları kurularak sağlanmaktadır [2].

Arı sürülerinin yiyecek arama davranışları, sürü içerisindeki bireylerin birbirleri arasındaki bilgi paylaşımı, öğrenme ve ezberleme özellikleri geçmiş zamanlarda sürü zekası alanındaki araştırma konularından biri olmuştur. Bal arıları üzerinde yapılan çalışmalar son yıllarda literatürde bir artış göstermektedir. Doğal bir arı kolonisinde sürü içerisindeki her arının yapacağı bir görev bulunmaktadır. Arılar, kendi aralarındaki iş paylaşımını kendi kendilerine yapmaktadırlar. Bir sürü içerisindeki iş paylaşımı ve bir merkeze bağlı olmadan kendi kendilerine organize olabilmeleri durumu sürü zekasının iki önemli özelliğidir. Tereshko'nun geliştirmiş olduğu reaktif difüzyon denkleminde de sürü içerisindeki ortak zekanın ortaya çıkmasını sağlayan minimum yiyecek arama modelinde üç temel bileşen bulunmaktadır [3]. Bu bileşenler, görevsiz işçi arılar, görevli işçi arılar ve yiyecek kaynaklarıdır.

## 2. Kullanılan Materyaller ve Metotlar

### 2.1. Yapay Arı Koloni Algoritması

Bilim insanları doğadaki canlıların sürü davranışlarından faydalanıp, özellikle sistemlerin analizinde ve kontrolünde kullanılabileceğini göstermişlerdir. Karıncaların yiyecek ararken kullandıkları yöntemleri, balık sürülerinin birlikte yüzmesi veya birlikte kaçması, kuş sürülerinin göç sırasındaki uçuş düzenleri gibi farklı gözlemlerden çeşitli sezgisel algoritmalar oluşturulmuştur. Geçtiğimiz son yıllarda ise bilgisayar uzmanları ve biyologların birlikte çalışmaları ile "Yapay Yaşam" alanı kapsamında bu sürülerin davranışlarının algoritmik olarak nasıl modellenilebileceği ve sürü bireylerinin birbirleri arasındaki iletişim mantığı üzerinde çeşitli çalışmalar yürütülmüştür [4].

Yapay Arı Koloni (ABC) Algoritması sürü zekası temelli olarak çalışmakta ve sürünün davranışlarını çeşitli yollardan modellemektedir. Özellikleri ise arı sürüleri, işlerini doğal olarak paylaşabilen ve çevresel değişimlere sürü zekasıyla kendilerini uyarlayabilen sürülerdir [5]. Koloni halinde yaşayan bu sosyal yaşamda 3 arı çeşidi bulunmaktadır: Bunlar dişi olan işçi arılar, kraliçe arı ve erkek arılar'dır. Bu modelde, arıların yiyecek arama davranışı incelenerek yapay arı

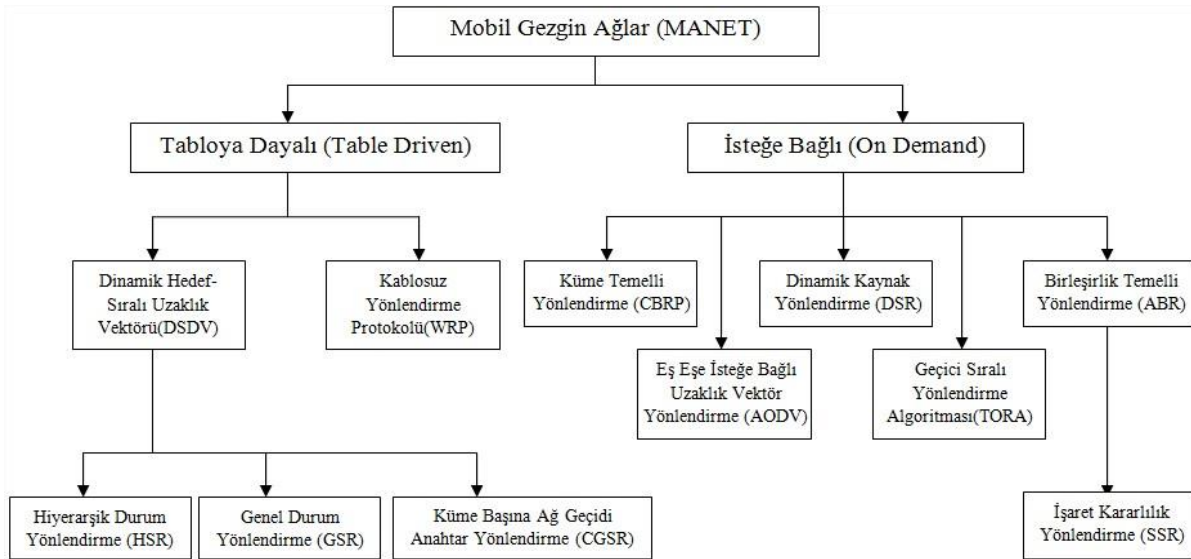
kolonisi algoritması geliştirilmiştir. İşçi arılar, gözlemci arılar ve kaşif arılar olmak üzere, yapay arı kolonisi algoritmasında da, üç grup arı bulunmaktadır. Bu üç gruptaki arıların farklı görevleri bulunmaktadır.

Gözlemci arıların görevi, yeni bal kaynaklarının yerlerini tespit etmek ve işçi arıları bu kaynaklara yönlendirmektir. Yeni kaynakların belirlenebilmesi için işçi arıların önceki kaynaklardan edindiği gözlemlerin analiz edilmesi gerekmektedir. İşçi arıların görevi, gözlemci arılar tarafından belirlenmiş olan bal kaynaklarına gidip, buradaki kaynaklardan çiçek özlerini toplamak ve topladığı çiçek özlerini kovana boşaltmaktır. Diğer bir görevi ise, çiçek özlerini topladığı kaynağın kalitesini gözlemci arılara bildirmektir. Kaşif arıların görevi ise, gözlemci arılardan farklı olarak rastgele çevreyi dolaşmak ve sadece kendi gözlemlerini analiz ederek yeni bal kaynaklarının yerini aramaktır.

## 2.2. Mobil Gezgin Ağlar (MANET)

Çalışması için sabit bir kablo alt yapısına gerek duymayan ağ üzerinde bulunan cihazların minimum konfigürasyon ile kısa sürede birbirleri arasında oluşturdukları kablosuz ağlar mobil gezgin ağlar olarak adlandırılır. Diğer bir tanımda ise hareketli kablosuz istasyonlar arasında herhangi bir erişim noktası bulunmaksızın, geçici olarak oluşturulan ve dinamik olarak değişebilen ağ yapısı mobil gezgin ağlar (MANET) olarak adlandırılmaktadır. Mobil gezgin ağlar, ortam koşullarına hızlı bir şekilde ve önceden yapılmış bir hazırlık olmadan kendi kendini hazırlayarak hizmet ettiği birimler arasında iletişimi sağlayan yapılardır.

Ağ içindeki haberleşmenin kolay yapılabilmesi için düğümlerin yerlerinin bulunmasını sağlayan yönlendirme protokolleri kullanılabilir. Kablosuz ağlar için kullanılan yönlendirme protokolleri, düğümler arasındaki iletişim için doğru ve hızlı yollar oluşturabilir. Bu işlemin yapılması için izlenecek yolun, minimum bant genişliğine sahip olmasına dikkat edilmesi gereklidir [6]. Mobil Gezgin Ağlar, düğümler arasında uçtan uca haberleşmeyi sağlamak için geleneksel TCP/IP yapısını kullanır. Kablosuz ağlardaki sınırlı kaynak ve hareketlilik kısıtlamaları yüzünden TCP/IP modelindeki her katman, Mobil Gezgin Ağlar da tekrar tanımlama ve düzenleme gerektirir. Mobil Gezgin Ağlar da kullanılan birçok yönlendirme protokolleri bulunmaktadır. Bunları farklı başlıklar altında gruplandırılabilir. Tabloya Dayalı (Table Driven) ve İsteğe Bağlı (On-Demand) olmak üzere 2 sınıfa ayrılırlar.



Şekil 1. MANET Yönlendirme Algoritmaları

Hareketlilik modelleri, ad hoc ağlarının simülasyonunda önemli bir rol oynamaktadır. Simülasyon sürecinde, Rastgele Geçiş Noktası Hareketlilik Modeli simüle edilmektedir.

### 2.2.1. Rastgele Geçiş Noktası Hareketlilik Modeli (RWM)

Rastgele geçiş noktası modeli, mobil kullanıcıların hareketini ve onların zamanla değişen konum, hız ve ivme değerlerini hesaplamak için kullanılan bir modeldir. Yeni ağ protokolleri değerlendirildiğinde, hareketlilik modelleri simülasyon amaçlı kullanılmaktadır. Rastgele geçiş noktası modeli ilk olarak Johnson ve Maltz tarafından önerilmiştir. Sadeliği ve geniş kullanılabilirliği sayesinde, mobil ad hoc ağ (MANET) yönlendirme protokollerini değerlendirmek için en popüler hareketlilik modellerinden biri haline gelmiştir [7]. Rastgele Geçiş Noktası Hareketlilik Modeli, mobil düğümlerin kısıtlama olmaksızın rastgele ve serbestçe hareket etmesine olanak sağlar. Daha spesifik olmak gerekirse, düğümler için hedef, hız, yön, tüm rastgele ve bağımsız seçenekler ayarlanabilmektedir. Model bu tür pek çok simülasyon çalışmalarında kullanılmıştır. Düğümler, simülasyon alanında dağıtılmış noktalara gitmeden önce bir süre duraklayabilir ve daha sonra simülasyon alanında bulunan yeni noktalara hareket edebilir.

### 2.3. Kuyruk Ağ Analizi

Simülasyon uygulamasında, bir paketin gecikme süresinin hesaplanması için Kleinrock'un Bağımsızlık Varsayımı kullanılmaktadır.

$$\text{End-to-End delay} = T_{\text{destination receives packet}} - T_{\text{source wants to send packet}} \quad (1)$$

Kaynak, bir hedefe paket göndermek istediğinde ve paket hedefe ulaştığı andaki zaman farkı Uçtan Uca gecikme olarak tanımlanmaktadır. Bugün birçok uygulama da (örneğin IP telefon) kullanılabilir sonuçlar sunmak için küçük bir gecikme gerekmektedir. Bu gecikme uygulamalar için kullanılan protokollerin yatınlığını göstermektedir. Bir kaynak düğüm S'den, hedef düğüm D'ye paket göndermek için beklenen tepki süresinin gecikme değeri, paketin hedefe gideceği yol boyunca ziyaret ettiği tüm linkler ve düğümlerdeki tepki sürelerinin toplamıdır [8].

$$E[R(S,D)] = \sum E[R_{ij}] + \sum E[R_i] \quad (2)$$

Tüm linklerdeki beklenen gecikme süresi  $E[R_{ij}] = 1/(\mu_{cij} - \lambda_{ij})$ ' dir.

$\mu_{cij}$  = bir link  $ij$ (paket/sn.) üzerinde iletilen paketlerin sayısı,

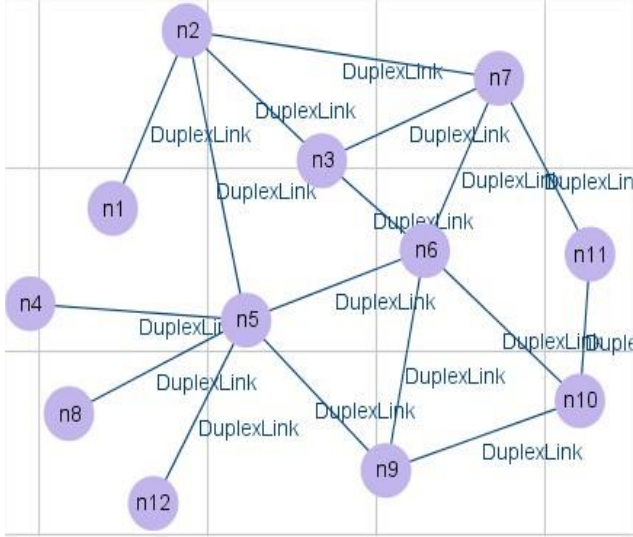
$\lambda_{ij}$  = bir linkin  $ij$ (paket/sn.) üzerine varış oranıdır.

$E[S_i] = 1/\mu_i$  ve  $\rho_i = \lambda_i/\mu_i$  ile  $i$  düğümü için beklenen gecikme süresi,

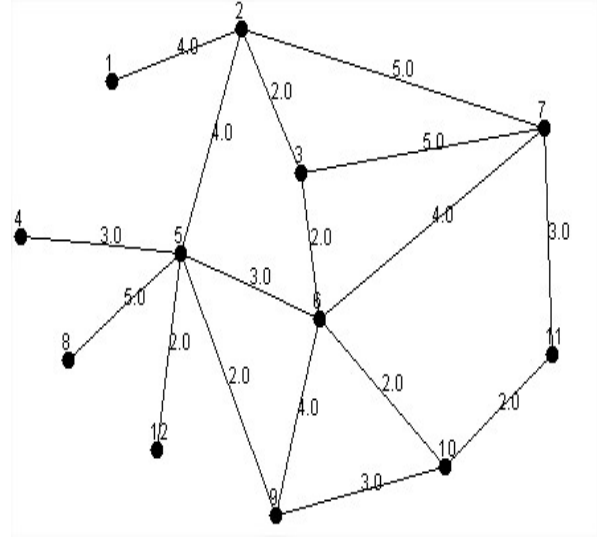
$E[R_i] = (\rho_i * E[S_i]) / (1 - \rho_i) + E[S_i]$ ' dir.  $i = 1, \dots, N$ .

### 2.4. Graf Veri Modeli ve Dijkstra Algoritması

Bir ağ içerisindeki düğümlerin durumunu ve düğümlerin arasındaki bağıntıyı göstermek için diyagram çizmek günümüzde oldukça yaygınlaşmıştır. Bu diyagramlar da düğümler noktalar ile ve düğümler arasındaki bağlantılar ise çizgilerle temsil edilirler. Şekil 2.a da bir ağ topolojisi ve Şekil 2.b de, 2.a da verilen ağ topolojisinin graf veri modeli gösterilmektedir. Bu ağ topolojisinde düğümler noktalar ile düğümler arası bağlantılar ise çizgilerle temsil edilmektedir. Graf veri modeli, aynı küme içerisinde bulunan noktaların ve çizgilerin birleştirilmesi ile oluşturulur. Noktalar birleşme yerlerini, çizgiler ise noktaların bağlantı ilişkilerini gösterir [9].



Şekil 2.a. Örnek Ağ Topolojisi



Şekil 2.b. Örnek Ağ Topolojisinin Graf Yapısı

Dijkstra Algoritması, graf yapıları içerisinde birbirine bağlı noktalar arasındaki en kısa yolu bulmak için kullanılan algoritmalardan birisidir [10]. Dijkstra Algoritmasının çalışma prensibi;

1. Kaynak ve hedef düğüm belirlenir. Kaynak düğümünden hedef düğüme gidebilmek için diğer düğümler seçilir.
2. Bu düğümlerde en az ağırlıklı(maliyetli) olan düğüm işaretlenir, seçilmemiş olana diğer düğümlerin maliyet değerleri sonsuz olarak işaretlenir.
3. Daha sonra 2. adım tekrar ettirilip maliyeti en az olan düğümler vasıtasıyla hedef düğüme doğru ilerleme devam ettirilir.

### 3. Simülasyon Algoritmasının Senaryosu

#### 3.1. Simülasyon Programı

Çalışmanın temel amacı, kuyruk ağ analizi ile birlikte ABC Algoritması tabanlı Mobil Ad Hoc Ağlarında dinamik yönlendirme için bir simülasyon modeli oluşturmaktır. Yönlendirme algoritması kaynak düğümünden hedef düğüme data paketlerinin deterministik olarak iletilmesini sağlamaktadır. Tüm bağlantıların çift yönlü olarak çalıştığını ve algoritmanın çalışması sırasında ağda bulunan bütün düğümlerin tam olarak işbirliği halinde olduğunu varsayılmaktadır.

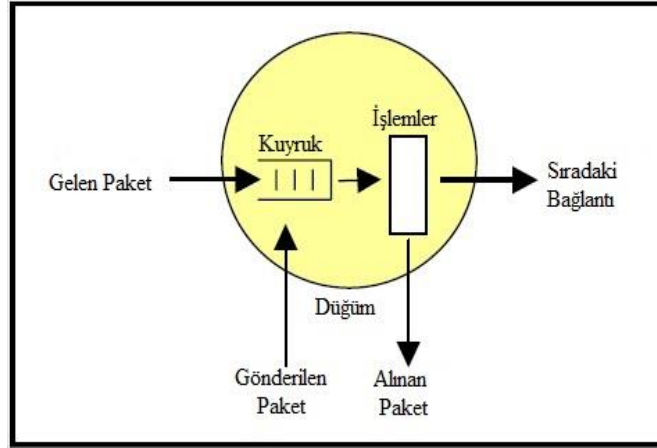
#### 3.2. Simülasyon Ortamı ve Gerekli Kurulumlar

##### 3.2.1. Ağ (Network) Kurulumu

Manet uygulamasında N düğümleri ve M bağlantıları yönlendirilmiş bir grafik üzerinde eşleştirilmektedir. Tüm bağlantılar da, bant genişliği (bit / sn) ve iletim gecikmesi (sn) karakterize bir bit olarak görülmektedir. Bu amaçla, depola ve ilet tipi her bir düğüm (yani, anahtar elemanının) dışarıdan gelen ve dışarıya çıkan paketlerin depolandığı tampon bir boşlukta (kuyruk) tutulmaktadır.

Bu tampon boşluk, her gelen ve giden linke bağlı tüm kuyruklar arasında paylaşılan bir kaynaktır. İletilen paketler, veri paketleri ya da yönlendirme paketleri olabilir. Paketler sıraya alınmış veya ilk giren ilk çıkar (FIFO) ilkesi temelinde iletilebilmektedir. Bir paket, yönlendirme tablosundan

hedef düğüme doğru giden yolu takip etmek için kullanılacak bağlantı hakkındaki bilgileri okumaktadır. Bağlantı kaynakları kullanılabilir olduğunda, kaynaklar ayrılır ve kaynakların transferleri ayarlanır. Bir paketin bir komşu düğüme geçmek için gereken zaman, paketin büyüklüğü ve bağlantı iletiminin özelliklerine bağlı olarak değişebilmektedir. Paketi tutmak için yeterli arabellek alanı yoksa paket iletdikten sonra atılır. Ağda bulunan iki gezgin düğümden her  $i$  ve  $j$  düğümleri arasındaki bağlantılar çift yönlüdür.



Şekil 3. Örnek bir Düğüm Modeli

### 3.2.2. Topoloji Kurulumu

Simülasyon ortamında sürekli seyahate devam eden mobil düğümler oluşturulmuştur, kare bir bölgeye kıyasla düğümler arasında mesafeler olması için simülasyon alanı  $800 \text{ m} \times 600 \text{ m}$  seçilmiştir, paketler bu mesafeler arasında daha çok düğüm üzerinden gönderilmektedir.

### 3.2.3. Yönlendirme(Routing) Tablosu Kurulumu

Yönlendirme tablosu, satırlar ve sütunlar içeren, yerel olarak depolanan, konvansiyonel bir tablodur ve ağın her bir düğümü için bir değere sahiptir. Bir düğümün yönlendirme tablosu, bu düğümden ulaşılabilir komşu düğümlerinin temsil edildiği sütunları içermektedir. Yönlendirme tablosu ağdaki her bir düğüm için, düğümün kendisi dışında, hedeflerin temsil edildiği değerleri içermektedir. Tablodaki değerler, kısa sürede komşu düğümler arasındaki satırların temsil edildiği, diğer düğümlerin ulaşma olasılığını temsil eden 0 ve 1 arasındaki sayısal değerlerdir. Her satırdaki olasılıklar toplamı 1'dir. Girişler,  $N$  düğümünün komşularının sayısı, ilgili hedef için bir sonraki atlama olana kadar her komşu için olasılıklar  $1 / N$  ile başlatılmaktadır. Hedef düğüm  $i$  olduğunda bir sonraki düğüm olarak  $n$  seçme iyiliğini ifade olasılık değeri ' $P_{in}$ ', depolama ile her bir  $(i, n)$  çifti için saklanır:

$$\sum_{n \in N_k} P_{in} = 1 \in [1, N_k], N_k = \{\text{neighbours}(k)\}; \quad (3)$$

### 3.2.4. Parametrelerin Kurulumu

Simülasyon sürecinin başında, her zaman biriminde düğüme gelen paketlerin ortalama sayısı olan varış oranı  $\lambda$  'nın tanımlanması gerekmektedir. Diğer bir parametre ise her zaman biriminde veri paketlerinin ortalama sayısı, ağdaki her  $i$  ve  $j$  düğümleri arasındaki iletim oranı olan,  $\mu_{ij}$  (Bant Genişliği) 'nin tanımlanması gerekmektedir. Bu  $\mu_{ij}$  parametresi, iletim oranı tablosu adı verilen bir matris olarak verilmektedir.

### 3.2.5. Zamanlayıcı Kurulumu

Simülasyon sürecinin başında, iki farklı zamanlayıcı tanımlanmaktadır. Arı zamanlayıcısı, Tbee verilen bir ağ topolojisi üzerinde abc algoritmasının çalışması için gerekli olan zamanı göstermektedir (ABC algoritması için sona erme zamanı). Örneğin, Tbee = 30 saniye Abc algoritmasının 30 sn de bir her yeni ağ topolojisi için çalışmasını bitirmesi anlamına gelmektedir. Simülasyon zamanlayıcı Tsimul, Abc algoritmasının her Tbee zamanı süre tekrar edileceği simülasyon işlemi için gereken bütün zamandır. Örneğin, Tsimul = 180 saniye ve Tbee = 30 saniye ise Abc algoritması 6 farklı ağ topolojisinde 6 kez tekrarlanır.

## 4. Simülasyon Sonuçları ve Yapılacak Çalışmalar

Bu bölümde, yukarıda bölümlerde açıklanan kurulumlar ve değişkenler kullanılarak E[V] veri paketleri için Mobil Gezgin Ağlar (Manet) aracılığıyla Uçtan Uca(End-to-End) gecikme değerlendirmektir. Simülasyon uygulaması için Network Simulator 2 (NS-2) simülasyon yazılımı ve Ubuntu Masaüstü 12.04 sürümlü Linux İşletim sistemi kullanılmıştır. Simülasyon senaryosu 800x600 m<sup>2</sup> dikdörtgen bir alanda RWM hareketlilik modeline göre rastgele yerleştirilmiş ve sürekli hareket halinde olan bir kaç dizi düğümden oluşmaktadır. Simülasyon sürecinde düğümler bu alan içerisinde herhangi bir yere hareket ettirilebilmektedir. Her düğüm rastgele bir noktaya doğru hareket etmekte ve daha sonra saniyenin kalan periyodunda beklemektedir. Bekleme süresinin ardından düğüm rastgele seçilen başka bir noktaya doğru hareket etmektedir. Bu işlem simülasyon boyunca tekrarlanmaktadır. Bu işlem Kleinrock 'un gecikme analizi ile değerlendirildiğinde; yönlendirme tablosundaki değerler, arıların düğümler arasındaki davranışlarının her simülasyondan sonra, ağ topolojisinin değişmesinden dolayı simülasyon bitene kadar değişmektedir. Son olarak, bir kaynak düğümden, hedef düğüme gecikme değerinin en düşük seviyede olduğu varsayılmaktadır. Simülasyon için kullanılacak parametreler ve parametrelerin değerleri tablo 1 de verilmiştir.

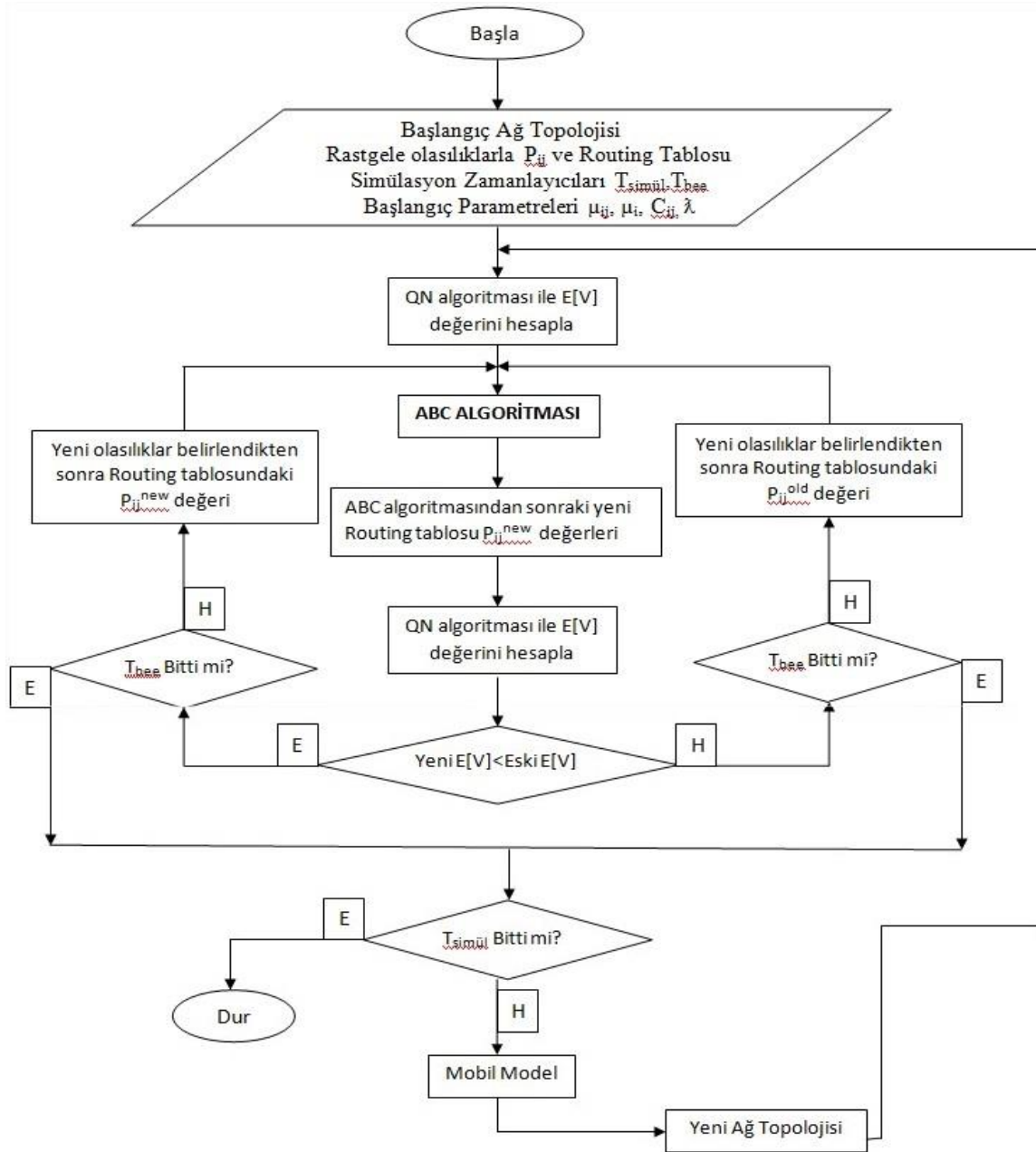
**Tablo 1.** Simülasyon Parametreleri

Parametreler	Örnek Değerler
Düğüm Sayısı	10, 20, ..., 100
Geliş oranı	150 Kbps
İletim aralığı	200 m
Hız / Yön	10 m/sn & 45°
Paket boyutu	64 byte
Duraklama süresi (Bee zaman)	10, 20, ..., 100 sn.
Simülasyon zamanı	180 sn.
Bağlantı bant genişliği	1 Mbps
Simülasyon alanı	800 m × 600 m
Hareketlilik modeli	RWM modeli
Yönlendirme protokolü	ABC algoritması

Ad hoc yönlendirme algoritmasını değerlendirmek için aşağıdaki gibi bir senaryo kullanılmıştır. Varsayımlar:

- 2 boyutlu dikdörtgen ve engelsiz bir alan,
- Düğümler arasında çift yönlü bağlantılar,
- Sabit düğüm sayısı (Farklı Topolojiler),
- Düğümler simülasyon boyunca çalışmakta,
- Rastgele geçiş noktası hareketlilik (RWM) modeli: Düğümler, simülasyon alanında eşit olarak bölünmüş noktalara gitmeden önce bir miktar duraklayabilmekte daha sonra simülasyon alanda yeni noktalara hareket ettirilebilmektedir.

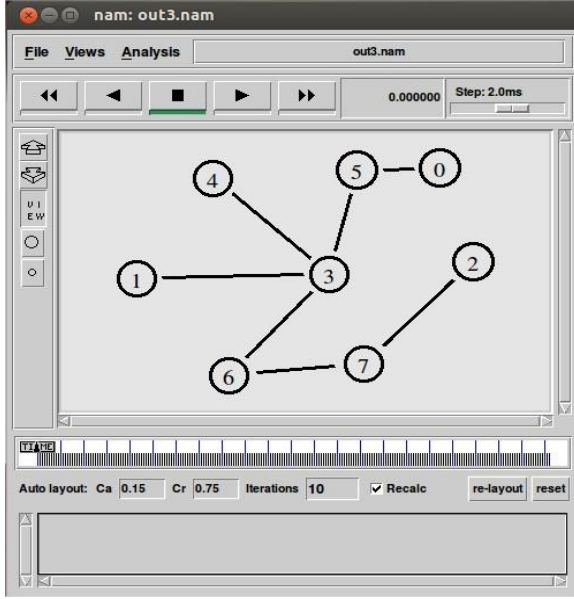
Simülasyon sürecinde Şekil 2.a da görüldüğü gibi oluşturulmuş farklı ağ topolojilerinde, bir veri paketini kaynak düğümden hedef düğüme göndermek için graf veri modeli örnek olarak alınmaktadır. Graf veri modelinde, noktalar arasındaki en kısa yolu bulmak için kullanılan algoritmalarından bir tanesi olan Dijkstra algoritması, simülasyon sürecinde ağ topolojisi üzerinde bulunan düğümler arasında, kaynak düğüm ile hedef düğüm arasındaki en kısa yolu bulmak için kullanılmaktadır. Simülasyon algoritmasının akış şeması Şekil 5 de verilmiştir.



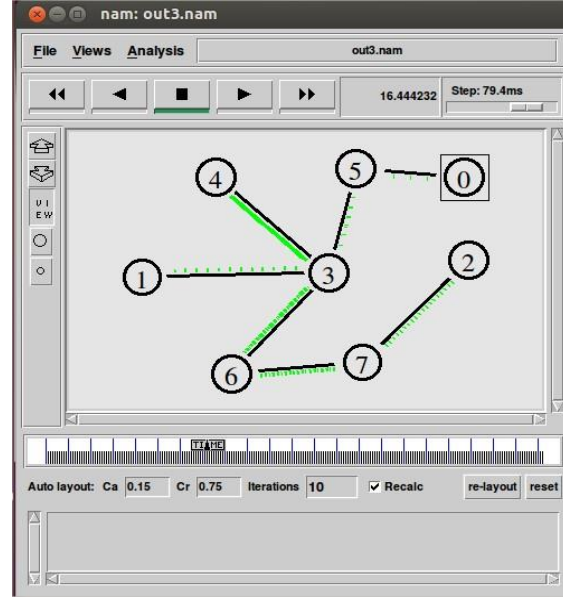
Şekil 5. Simülasyon Algoritmasının Akış Şeması

Şekil 6.a da simülasyon başlangıç durumunda bulunmaktadırlar ve düğümler iletişime 3. Düğüm üzerinden geçmektedirler. Simülasyon başlatıldığı zaman Şekil 6.b deki gibi düğümler 3. Düğüm üzerinden birbirleri ile paket alışverişini yapmaktadırlar. Simülasyon süresi 60 sn. olarak belirlenmiştir. Bu süre bittiğinde iletişime geçen düğümlerin paket iletimleri Şekil 6.c de verilen grafikte gösterilmektedir.

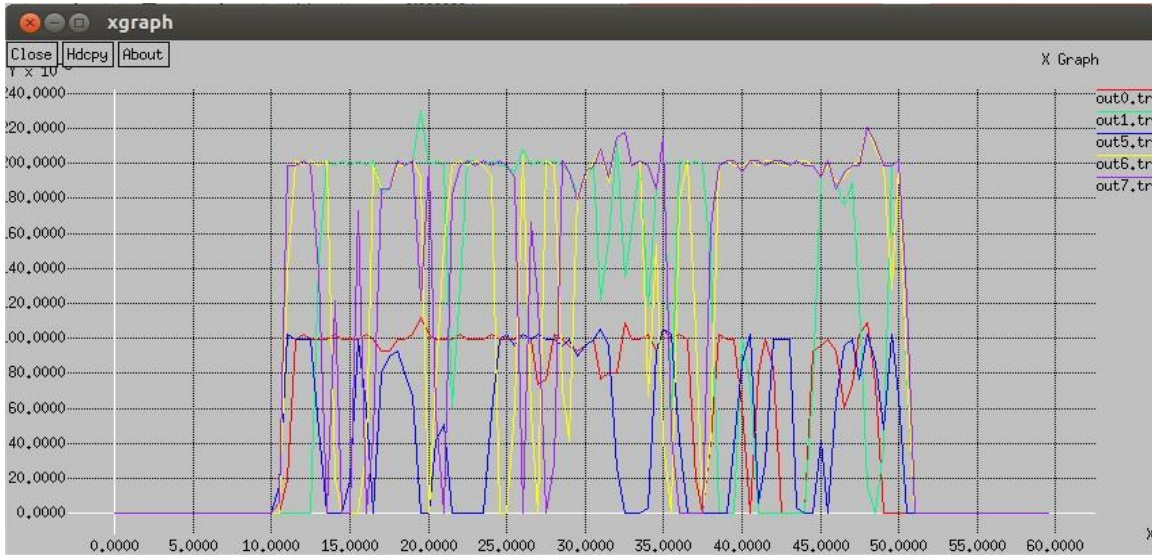




Şekil 6.a. 1. Durum



Şekil 6.b. 2. Durum



Şekil 6.c. Düğümün Durum Grafikleri X (sn) ve Y (İletilen Veri Paket)

Şekil 6.c de, 60 saniyelik simülasyon boyunca düğümler arasındaki iletilen veri paketlerinin gösterildiği bir grafik verilmektedir. İlk 10 sn. ve son 10 sn. de düğümler arasında iletişim yapılmamaktadır. Veri paketlerinin iletiliminin düğümlere göre farklılık göstermesinin nedenleri arasında; iletilen veri paketlerinin büyüklükleri, bir düğümün diğer düğümlere göre daha fazla düğüm üzerinden veri göndermesi veya gönderilen düğümün kaynaktan hedef düğüme ulaşmadan ağ topolojisinden atılması gibi sebepler yer almaktadır. Farklı simülasyon topolojilerinde alınan değerlerin farklı olduğu gözlemlenmiştir.

#### 4.1. Yapılacak Çalışmalar

1. Farklı topolojileri ve farklı routing problemleri için simülasyonlar yapıp sonuçlar yapılmış olan simülasyon ile karşılaştırılacaktır.
2. Simülasyonu oluşturulmuş sistem, önceden yapılmış veya yapılmakta olan uygulamalarla karşılaştırılacaktır.

## Sonuç

MANET 'in sahip olduğu farklı araştırma alanları sayesinde kullanımı son yıllarda artmaktadır. MANET deki yönlendirme işlemi zor bir uygulamadır, bu zorluğun temel nedeni, ağda sürekli olarak gerçekleştirilen değişiklikler yüzündendir. Bu tür proaktif ve reaktif protokollerin, avantajları ve dezavantajları için her birinin çözümünde bazı geleneksel yöntemler bulunmaktadır. Buna rağmen, gerçekleştirilen uygulama bu çözümlerin yerine daha iyi bir performans sunmak için geliştirilmiştir.

Başarıyla sonuçlanan uygulama da, bir ad hoc ağında optimum bir yönlendirme davranış modelini tanımlamak için öğrenme desteklemesi kullanılarak faaliyet gösteren bir ad hoc yönlendirme algoritması tasarlanmıştır. Tasarlanan algoritma da optimum davranış, hem kısa döngü yollarını aramakta hem de bu yolları oluşturan bağlantıların kalitesini dikkate almaktadır.

Ağ boyunca bilgiyi yönlendirme hareketi ve ağdaki bağlantıların sürekli izlenmesi için bir öğrenme stratejisi geliştirilmiştir. Geliştirilen öğrenme stratejisi Yapay Arı Koloni (ABC) Algoritması'na dayanmaktadır. Yapay Arı Koloni algoritmasının benzersiz özelliklerinden bazıları Ad hoc yönlendirme sorununa uygulanabilmektedir. Örneğin; protokol tarafından yönlendirilen her paket, yapay arı kolonisinde bir arıya eşdeğer olabilmektedir.

Sonuç olarak, bu şekildeki dinamik ağlar da çözüm üretebilmek için ABC algoritmasının kullanabileceğini ve özellikle gerçekleştirilen model de sistem performansının artırılabilirdiğinin sonucuna varılabilmektedir.

## Kaynaklar

- [1] Royer EM, Toh CK. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. IEEE Personal Communications, Apr. 1999;10:1-2.
- [2] McDonald AB. A Mobility-Based Framework for Adaptive Dynamic Cluster-Based Hybrid Routing in wireless Ad Hoc Networks. Ph.D. Dissertation Proposal, Uni. Of Pittsburgh, 1999.
- [3] Tereshko V. Reaction-Diffusion Model of a Honey Bee Colony's Foraging Behaviour, 6th International Conference on Parallel Problem Solving from Nature, 807-816, 3-540-41056-2, Springer-Verlag, London, UK, 2000.
- [4] Korb K, Randall M, Hendtlass T. Artificial Life: Borrowing from Biology, 4th Australian Conference ACAL; 2009
- [5] Karaboğa D, Akay B. A Survey: Algorithms Simulating Bee Swarm Intelligence, Springer, Cilt 31, No 1-4, 61-85, 2010
- [6] Bayılmış C, Ertürk İ, Ceken C, Bandırmalı N. DSR ve AODV Manet Yönlendirme Protokollerinin Başarım Değerlendirmesi, 2005;4:1-2
- [7] Navidi W, Camp T. Stationary Distributions for the Random Waypoint Mobility Model, IEEE Transactions On Mobile Computing, Vol. 3, No. 1, January-March 2004
- [8] Haverkort B. Performance Of Computer Communication System, A Model Based Approach, John Wiley & Sons, Ltd., 1998.
- [9] Bayzan Ş, Tokat S. Araç Rotalama Probleminde Talep Noktasında Bekleme Süresinin Alınan Toplam Yola Etkisi, III. Otomasyon Sempozyumu Denizli, pp. 255-259, Kasım, 2005.
- [10] Noto M, Sato H. A Method for the Shortest Path Search by Extended Dijkstra Algorithm, Systems, Man, and Cybernetics, 2000 IEEE International Conference on (Volume:3 ), 2000.
- [11] Helmi T. PhD Thesis Modeling And Simulation Of Routing Protocol For Ad Hoc Networks Combining Queuing Network Analysis And ANT Colony Algorithms, April 2005