

**ACİL DURUMLARDA YAYA TAHLİYESİ İÇİN
GRUP DAVRANIŞI İÇEREN
SOSYAL KUVVET MODELİ ÖNERİSİ**

YÜKSEK LİSANS TEZİ

Yakup TURGUT

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Y.Doç.Dr. Cafer Erhan BOZDAĞ

Aralık 2017

**ACİL DURUMLARDA YAYA TAHLİYESİ İÇİN
GRUP DAVRANIŞI İÇEREN
SOSYAL KUVVET MODELİ ÖNERİSİ**

YÜKSEK LİSANS TEZİ

**Yakup TURGUT
(507151130)**

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Y.Doç.Dr. Cafer Erhan BOZDAĞ

Aralık 2017

İTÜ, Fen Bilimleri Enstitüsü'nün 507151130 numaralı Yüksek Lisans Öğrencisi Yakup TURGUT, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “ACİL DURUMLARDA YAYA TAHLİYESİ İÇİN GRUP DAVRANIŞI İÇEREN SOSYAL KUVVET MODELİ ÖNERİSİ” başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Y.Doç.Dr. Cafer Erhan BOZDAĞ**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç.Dr. Ayberk Soyer**
İstanbul Teknik Üniversitesi

Doç.Dr. Nihan Çetin Demirel
Yıldız Teknik Üniversitesi

.....

Teslim Tarihi : **17 Kasım 2017**
Savunma Tarihi : **18 Aralık 2017**





Aileme,



ÖNSÖZ

Yüksek lisans tez çalışmamın konusunun belirlenmesinden mevcut halini almasına kadar geçen süre yaklaşık olarak 1 yıldır. Bu süreçte destek olan tüm arkadaşlarıma teşekkür ederim. Varlığında en büyük paya sahip olan anneme teşekkürü bir borç bilirim. Son olarak yüksek lisans tezimin yazılması sürecinde sunduğu yardımlar ve sağladığı motivasyon desteği için değerli tez danışmanım Y.Doç.Dr. Cafer Erhan BOZDAĞ'a çok teşekkür ederim. Bu çalışmanın bu alanda çalışan ya da çalışacak olan araştırmacılara küçük de olsa bir yardımı olmasını temenni ederim.

Aralık 2017

Yakup TURGUT
(Endüstri Mühendisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR.....	xiii
SEMBOLLER	xv
ÇİZELGE LİSTESİ.....	xvii
ŞEKİL LİSTESİ.....	xix
ÖZET	xxi
SUMMARY	xxiii
1. GİRİŞ	1
1.1 Tezin Amacı.....	1
1.2 Tezin Bölümleri	2
2. YAYA HAREKETİ MODELLERİ.....	5
2.1 Modellerin Sınıflandırılması.....	5
2.2 Mikroskopik Modeller.....	5
2.2.1 Ajan temelli benzetim.....	5
2.2.1.1 Hücresel otomat modeli	6
2.2.1.2 Kuvvete dayalı modeller	7
2.2.2 Kural tabanlı modeller	8
2.2.3 Oyun teorisine dayalı modeller	8
2.3 Makroskopik Modeller	9
2.3.1 Kafes gaz modeli	9
2.3.2 Kuyruk modelleri.....	9
2.3.3 Ağ tabanlı modeller	10
2.3.4 Akış tabanlı modeller	10
2.3.5 Akışkan dinamiği modelleri	11
2.4 Mezoskopik Modeller.....	11
2.5 Modellerin Karşılaştırılması.....	11
3. SOSYAL KUVVET MODELİ.....	15
3.1 Sosyal Kuvvet Modeli Temel Yapısı	15
3.2 Sosyal Kuvvet Modeli Formülasyonu	16
3.2.1 Yönlendirici kuvvet.....	16
3.2.2 İtici kuvvet.....	17
3.2.2.1 İçsel itici kuvvet.....	17
3.2.2.2 Dışsal itici kuvvet	18
3.2.3 Çekici kuvvet.....	20
3.2.4 Toplam kuvvet	20
3.3 Sosyal Kuvvet Modeline Yönelik Eleştiriler.....	21

3.4 Yaya Hareketi Temel Davranışlar (Kalıplar)	22
3.4.1 Normal durum	22
3.4.2 Panik durumu.....	23
4. YAYA GRUP DAVRANIŞI.....	27
4.1 Literatür Taraması.....	27
4.2 Boids Algoritması.....	34
4.3 Algoritmanın Yapısı ve Çalışma Şekli	34
4.3.1 Ayırma	34
4.3.2 Birleştirme	36
4.3.3 Hizalama.....	36
5. YAYA DAVRANIŞI İÇİN ÖNERİLEN MODEL.....	37
5.1 Model Öncelikleri.....	37
5.1.1 Diferansiyel denklemler	37
5.1.1.1 Diferansiyel denklemlerin çözümünde nümerik yaklaşımlar	38
5.1.1.2 Euler(Euler-Cauchy) yöntemi.....	38
5.1.2 İnsan gözü görüş alanı	39
5.1.3 Sosyal uzaklık teorisi.....	39
5.2 YGDTS Modeli	40
5.2.1 Yayaların gruplandırılması	41
5.2.2 Model içerisinde yer alan kuvvetler	42
5.2.2.1 Yaya yönlendirici kuvveti	42
5.2.2.2 Yayalar arasındaki itici kuvvetler	43
5.2.2.3 Duvarlar ve engeller tarafından uygulanan itici kuvvetler.....	43
5.2.2.4 Yardımcı yönlendirici kuvvet	43
5.2.2.5 Grup kuvveti	44
5.2.2.6 Lider kuvveti.....	45
5.2.2.7 Tüm kuvvetler toplamı	47
6. SİMÜLASYON VE DENEY SONUÇLARI.....	49
6.1 Simülasyon Parametreleri.....	49
6.1.1 Simülasyon adım büyüklüğü	50
6.2 Modelin Doğrulanması.....	50
6.2.1 Literatür ile karşılaştırma	50
6.2.2 Acil durumda meydana gelen davranışlar(Kalıplar)	51
6.2.2.1 Engellerden ve diğer yayalardan kaçınma	51
6.2.2.2 Freezing-by-heating etkisi	51
6.2.2.3 Faster-is-slower etkisi	51
6.2.2.4 Kuyruk oluşturma	51
6.3 Simülasyon Deneyleri	51
6.3.1 Yaya sayısı.....	52
6.3.2 Çıkışın(Kapı) genişliği	52
6.3.3 Çıkış sayısı.....	52
6.3.4 Engellerin olması.....	53
6.3.5 Liderin varlığı ve sayısı	53
6.4 Deney Sonuçlarının Yorumlanması.....	55
7. SONUÇ VE ÖNERİLER	59

KAYNAKLAR.....	61
EKLER	65
EK A.1: AnyLogic Programı	67
EK A.2: Java Kodları.....	69
EK A.3: Deney Sonuçları Tabloları	91
ÖZGEÇMİŞ	95





KISALTMALAR

SKM	: Sosyal Kuvvet Modeli
BA	: Boids Algoritması
ATS	: Ajan Temelli Simülasyon
YGDTS	: Yaya Grup Davranışı Tahliye Simülasyonu





SEMBOLLER

$v_{\alpha}^0(0)$	: α yayasının başlangıçta ulaşmak istediği hız
$v_{\alpha}^0(t)$	: α yayasının t anında ulaşmak istediği hız
$\bar{v}_{\alpha}(t)$	: α yayasının t anına kadar geçen süredeki ortalama hızı
$n_{\alpha}(t)$	: α yayasının t anındaki sabırsızlık ölçüsü
$v_{\alpha}^{\max}$	: α yayasının ulaşabileceği maksimum hız
$v_{\alpha}(t)$	: α yayasının t anındaki hızı
m_{α}	: α yayasının kütlesi
$\vec{e}_{\alpha}(t)$	: α yayasının t anındaki hareket yönü vektörü
A_{α}	: Yaya bölgesel etkileşim gücü katsayısı
B_{α}	: Yaya bölgesel etkileşim gücü aralığı katsayısı
τ_{α}	: Gevşeme süresi
$\vec{n}_{\alpha\beta}$	: β yayasından α yayası doğrultusunda normalize edilmiş vektör
$\vec{n}_{\alpha w}$	: w duvarından α yayası doğrultusunda normalize edilmiş vektör
$r_{\alpha\beta}$	: α ve β yayalarının yarıçapları toplamı
$d_{\alpha\beta}$	: α ve β yayalarının kütle merkezleri arasındaki uzaklık
$d_{\alpha w}$	: α yayasının w duvarı sınırına olan dik uzaklığı
λ_{α}	: α yayasının anizotropik karakteristiği
C_1	: Sabit bir sayı
C_2	: Sabit bir sayı
$\vec{t}_{\alpha\beta}$	: Normalize vektöre teğet olan vektör
$\xi(t)$	: Yaya davranışlarındaki sistematik olmayan dalgalanmalar
ρ	: Yaya yoğunluğu seviyesi
m_x	: Yayaların kütle merkezinin x eksenindeki değeri
m_y	: Yayaların kütle merkezinin y eksenindeki değeri
g	: Tek bir yaya kümesini temsil eder
G	: Tüm yaya kümelerini içine alan evrensel yaya kümesi
s_g	: g kümesinde yer alan yaya sayısı
$l(x,y)$	: Yayaların iki boyutlu düzlemdeki konumlarını temsil eder.
S	: Sabit bir sayı
K_1	: Sabit bir sayı
K_2	: Sabit bir sayı
K_3	: Sabit bir sayı
K_4	: Sabit bir sayı
K_5	: Sabit bir sayı
$\vec{F}_{\alpha}^{\text{Yönlendirici}}(t)$	: α yayasına t anında etki eden yönlendirici kuvvet
$\vec{F}_{\alpha\beta}^{\text{iik}}(t)$	: α yayasına t anında etki eden içsel itici kuvvet
$\vec{F}_{\alpha\beta}^{\text{dik}}(t)$	: α yayasına t anında etki eden dışsal itici kuvvet

$\vec{F}_{\alpha\beta}^{itici}(t)$	: α yayasına t anında β yayası tarafından uygulanan net itici kuvvet
$\vec{F}_{\alpha w}^{iik}(t)$	: α yayasına t anında w duvarı tarafından uygulanan içsel itici kuvvet
$\vec{F}_{\alpha w}^{dik}(t)$	: α yayasına t anında w duvarı tarafından uygulanan dışsal itici kuvvet
$\vec{F}_{\alpha w}^{itici}(t)$	: α yayasına t anında w duvarı tarafından uygulanan net itici kuvvet
$\vec{F}_{\alpha\beta}^{çekici}(t)$	: α yayasına t anında β yayası tarafından uygulanan çekici kuvvet
$\vec{F}_{\alpha grup}(t)$	: α yayasına t anında üyesi olduğu grup tarafından uygulanan kuvvet
$\vec{F}_{\alpha lider}(t)$	: α yayasına t anında üyesi olduğu grubun lideri tarafında uygulanan kuvvet
$\vec{F}_{\alpha w}^{yrd}(t)$	: α yayasına t anında uygulanan yardımcı kuvvet
$\vec{f}_{\alpha}^{toplam}(t)$	: α yayasına t anında uygulanan toplam kuvvet



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1: Model sınıflandırması.....	12
Çizelge 2.2: Tahliye Simülasyonu Yazılımları ve Sınıflandırılması.....	13
Çizelge 3.1: Yönlendirici Kuvvet Formülasyonu Terminolojisi.	17
Çizelge 3.2: İçsel İtici Kuvvet Formülasyonu Terminolojisi.....	18
Çizelge 3.3: Dışsal İtici Kuvvet Formülasyonu Terminolojisi.	20
Çizelge 4.1: Literatür özeti.	33
Çizelge 6.1: SKM Simülasyon Parametreleri.....	49
Çizelge 6.2: Yaya sayısı ve tahliye süresi ilişkisi	52
Çizelge C.1 Çıkışın genişliği ve tahliye süresi ilişkisi	92
Çizelge C.2 Çıkış sayısı ve tahliye süresi ilişkisi.....	93
Çizelge C.3 Engel konumu ve tahliye süresi ilişkisi.....	94
Çizelge C.4 Grup Sayısı ve tahliye süresi ilişkisi	94



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 1.1 : Maslov'un ihtiyaçlar hiyerarşisi.	2
Şekil 2.1 : Game of Life simülasyonuna ait anlık ekran görüntüleri.	7
Şekil 2.2 : Bina kuyruk ağı gösterimi	10
Şekil 2.3 : Hesaplama süresi, detay düzeyi ve model ilişkisi	12
Şekil 3.1 : Sosyal Kuvvet Modeli Gösterimi	15
Şekil 3.2 : Yayalar arası etkileşimde normalize vektör ve teğet vektör ilişkisi ..	19
Şekil 3.3 : Duvar ve yaya etkileşiminde normalize vektör ve teğet vektör ilişkisi.....	19
Şekil 3.4 : Şerit oluşumu	22
Şekil 3.5 : Dar boğazlarda salınım.....	22
Şekil 3.6 : Kavşaklarda yaya akışı olgusu	23
Şekil 3.7 : "Freezing-by-heating" etkisi.....	24
Şekil 3.8 : "Faster is slower" etkisi	24
Şekil 3.9 : "Transition to incoordination due to clogging"	25
Şekil 3.10 : "Ignorance of available exits"	25
Şekil 4.1 : Kuş sürü davranışı.	35
Şekil 5.1 : İnsan gözü görüş alanı(field of view).	39
Şekil 5.2 : Proksemi(Proxemics).....	40
Şekil 5.3 : Yaya etki alanı.	43
Şekil 5.4 : Yardımcı yönlendirici kuvvet.	44
Şekil 5.5 : Grup Kuvveti.	44
Şekil 5.6 : Lider Kuvveti.....	45
Şekil 6.1 : Literatür Karşılaştırması.....	50
Şekil 6.2 : Çıkış genişliği ve tahliye süresi ilişkisi.	52
Şekil 6.3 : Çıkış sayısı ve tahliye süresi ilişkisi.	53
Şekil 6.4 : Modelin 6.27, 5.70 ve 3.90 saniyesinde yayaların hareketinin görüntüsü.....	53
Şekil 6.5 : Engelin çıkışa uzaklığı ve tahliye süresi ilişkisi.	54
Şekil 6.6 : Engel ve engelsiz durumda tahliye süresi.....	54
Şekil 6.7 : Modelin 1.75, 4.07 ve 4.68 saniyesinde yayaların hareketinin görüntüsü.....	54
Şekil 6.8 : Modelin 1.15 ve 0.79 saniyesinde yayaların hareketinin görüntüsü.	55
Şekil 6.9 : Lider(grup) sayısı ve tahliye süresi ilişkisi.....	55
Şekil 6.10 : Modelin 10 saniyesinde yayaların hareketinin görüntüsü.	55



ACİL DURUMLARDA YAYA TAHLİYESİ İÇİN GRUP DAVRANIŞI İÇEREN SOSYAL KUVVET MODELİ ÖNERİSİ

ÖZET

Güvenlik insanların temel ihtiyaçlarından birisidir. Öte yandan yangın, terör saldırısı ve doğal felaket gibi durumlar insanların ve diğer canlıların can ve mal güvenliği için büyük bir tehdit oluşturmaktadır. Bu tür felaketlerle yaşamımız boyunca çok sık karşılaşmamıza rağmen, karşılaşmamız durumunda hazırlıksız yakalanmamıza bağlı olarak çok dramatik sonuçlarla karşılaşabiliriz. Bu felaketlerden çoğu zaman kaçınmak mümkün olmadığından dolayı, karşılaşmamız anında olabileceklere karşı alınabilecek önlemler kayıpları en aza indirmek açısından faydalı olacaktır.

Acil durumlardaki kayıpları en aza indirebilmek için etkili tahliye planlarına ihtiyaç vardır. Deneysel çalışmalar ve bilgisayar destekli analizler bu planların oluşturulmasında uygun yöntemler olarak görülmektedir. Matematiksel denklemler, bazı fizik teorileri ve simülasyon modelleri tahliye planlamasında bilgisayar destekli analizin temelini oluşturmaktadır.

Acil durumdaki yaya davranışlarının modellenmesi yeni bina tasarımından robot planlama ve kontrolü gibi birçok farklı alanda önemli bir yere sahiptir. Trafik ve yapı mühendisleri, yaya trafiği planlamacıları ve tahliye planlamacıları en etkili tahliye ve yaya akışını sağlamak amacıyla bu modelleri yaptıkları yapıların tasarımı aşamasında bir girdi olarak kullanabilir.

Yaya dinamiklerinin modellemesinde kullanılan modellere son yıllarda olan ilgi artmaktadır. Çünkü bu modeller yaya davranışlarının tahmin edilmesinde ve tahliye bölgelerinin tasarımında kritik bir role sahiptir. Bu modeller literatürde farklı şekillerde sınıflandırılmaktadır. En yaygın olarak kullanılan sınıflandırmalar Mikroskopik-Makroskopik, Sürekli-Kesikli, Deterministik-Stokastik şeklindedir. Bu çalışma içerisinde ise modeller Mikroskopik-Mezoskopik-Makroskopik şeklinde sınıflandırılmıştır.

Çalışma içerisinde bu üç temel başlık altında toplanan modelleri incelediğimiz zaman, hesaplama zamanı ve sağlanan detay seviyesi noktalarında farklılıklar olduğu görülmektedir. Mikroskopik modeller daha fazla hesaplama zamanına ihtiyaç duyarken, daha detaylı ve daha gerçekçi yaya davranışları ortaya çıkardığı görülmektedir. Makroskopik modeller ise mikroskopik modellerin tam tersi daha az hesaplama zamanı ve daha az detaya sahip modeller ortaya koymaktadır. Mezoskopik modeller ise iki yaklaşımın hibrid hali olarak göze çarpmaktadır. Literatürde kullanılan modellere baktığımız zaman Mikroskopik modeller içerisinde yer alan Ajan Temelli modelleme yaklaşımının daha yaygın olarak kullanıldığı görülmektedir.

Ajan temelli benzetim bireylerin otonom ve birbirleriyle etkileşim kurabilen ajanlar olarak tanımlandığı modellerdir. Ajanlar sahip oldukları kurallar doğrultusunda bilgi alma, alınan bilgiyi işleme, değişen ortama uyum sağlama ve karar verme yeteneğine

sahip otonom bireylerdir Ajan temelli modelleme ve simülasyon kesikli olay simülasyonu ve nesneye yönelik programlama yönteminden faydalananak kompleks uyarlanabilir sistemleri yönetir.

Yaya davranışlarının modellenmesinde kullanılan mikroskopik modellerden biri Sosyal Kuvvet Modelidir. SKM yayaların hareketlerinin belirli kuvvetlerin etkisi altında ortaya çıktığını ileri süren, Newton'un ikinci hareket yasasına dayanan matematiksel bir modeldir. Modelin bu ismin almasının sebebi yayaların hareketlerinin ortaya çıkmasında etkili olan içsel motivasyonları birer kuvvet olarak düşünmesidir. Modelin bazı eksik yanları olmasına rağmen, yaya davranışlarının modellenmesinde literatürde çok önemli bir yere sahiptir.

Bu tez çalışmasında acil durumda kaçan yaya gruplarını modellemek amacıyla matematiksel bir model oluşturulmuş ve AnyLogic yazılımı kullanılarak modelin implementasyonu sağlanmıştır. Önerilen model Helbing'in temel SKM dayanmakta olup, model üzerinde grup davranışı doğrultusunda bazı değişiklikler yapılmıştır. Temel modele, modelin eksikleri ve grup davranışı doğrultusunda bazı yeni kuvvetler eklenmiştir.

Ajan temelli modelleme yaklaşımı kullanılmış ve yayalar belirli kuvvetlerin etkisi altında hareket eden ajanlar olarak modellenmiştir. Yayaların belirli bir t anındaki yeni konumlarının ve hızlarının elde edilmesinde Euler nümerik yönteminden faydalanılmıştır. Önerilen model iki boyutlu düzlemde sürekli iken, adi diferansiyel denklemin çözümünde kullanılan Euler nümerik yaklaşımdan dolayı zaman kesiklidir.

Yayalar gruplara bölünmüş ve gruplar halinde hareket etmeleri sağlanmıştır. Yaya grupları oluşturulurken Sosyal uzaklık teorisi ve K-Means kümeleme algoritmasından faydalanılmıştır. Yaya yoğunluğunun 0.125 den küçük olduğu durumlarda yayalar sosyal uzaklık teorisine göre gruplandırılırken yaya yoğunluğunun daha büyük olduğu durumlarda K-means kümeleme algoritmasından faydalanılmıştır. Yayaların birarada hareket etmelerini sağlamak amacıyla Boids algoritmasından esinlenilmiş ve grup içerisinde yer alan yayaların algılanan kütle merkezleri hesaplanarak ve bu kütle merkezinden grup üyelerine hayali bir grup kuvveti uygulanarak grup üyelerinin birbirlerine yakın hareket etmeleri sağlanmıştır. Ayrıca grup davranışının bir alt türü olan lideri takip etme davranışını da incelemek amacıyla her bir grup içerisinde bir lider olma durumunu incelenmiş, lider çekici kuvveti sosyal kuvvet modeline eklenmiştir. Ayrıca liderin hedefe doğru gidişinin engellenmemesi amacıyla lider itici kuvveti de eklenmiştir. Önerilen modelin etkinliğini ortaya koymak amacıyla simülasyon deneyleri yapılmıştır. Yaya sayısı, çıkış sayısı, çıkışın genişliği, engellerin varlığı, konumu, grup sayısı ve lider sayısı gibi farklı senaryolar test edilmiş ve tüm senaryolar için tahliye süreleri elde edilmiştir. Her bir deney için 100 koşum yapılmış ve yapılan koşumların ortalaması alınarak tahliye süreleri elde edilmiştir.

Modelin doğrulanmasını sağlamak amacıyla önerilen model literatürde yer alan çalışmalar ile karşılaştırılmış ve çalışmalara benzer sonuçlar verdiği görülmüştür. Daha sonra ise modelin acil durumlarda meydana gelen yaya olgularını meydana getirme durumu incelenmiş ve bu durumların birçoğunu ortaya koyduğu gözlemlenmiştir.

PROPOSAL OF SOCIAL FORCE MODEL INCLUDING GROUP BEHAVIOUR FOR PEDESTRIAN EVACUATION IN EMERGENCY SITUATIONS

SUMMARY

Security is one of the basic needs of people. On the other hand, situations such as fire, terrorist attacks and natural disasters pose a major threat to the lives and property of people and other living things. Although we have not met with these kinds of disasters very often throughout our lives, we may encounter very dramatic results depending on whether we are caught unprepared or not. Because these disasters are often impossible to avoid, Disaster countermeasures will be useful in terms of minimizing the losses in case we meet with a disaster.

The modeling of pedestrian behavior in emergency situations is important for many disciplines. The modeling of pedestrian behavior is an important factor for the design of new buildings, robot programming, pedestrian traffic planning and emergency planners. Traffic and civil engineers can use these models as an input to design structures that will provide the most efficient evacuation and pedestrian flow. Particularly in countries where the number of pedestrians is increasing, such as Istanbul, the modeling of pedestrian behavior has a critical prescription.

Effective evacuation plans are needed to reduce the most casualties in emergencies. Experimental studies and computer based analysis studies related to evacuation planning of structures (buildings) at these points are seen as suitable methods. Evacuation planning involves iterative processes and aims to ensure that people in a particular area are removed from this area as soon as possible in a risky situation and identify the most appropriate routes to provide these exits. Computer-aided analysis studies seem to be used more widely in the literature in evacuation planning, as they are less costly than experimental (practical) studies. Mathematical equations, some physics theories and simulation models are the basis of the computer assisted analysis studies used in the evacuation planning.

Models used to simulate pedestrian dynamics have begun to attract much attention in recent years, because these models play an important role in predicting pedestrian behavior and in designing evacuation zones. The models used in modeling the behavior exhibited by the publications are based on physical rules and mathematical equations. According to the results of literature review, there are various approaches in classification of models in studies. Microscopic-macroscopic, homogeneous-heterogeneous, intermittent-continuous and stochastic-deterministic models are the most common types of classifications. The majority of names used in these classifications refer to a state or a feature in a model. In this thesis, models will be introduced and examined under macroscopic and microscopic and mesoscopic titles in classification.

When we examine the models collected under the three basic headings, it is seen that there are differences in the complexity of the calculation and the level of detail

between the models. While microscopic models require the most computation time, they provide a more detailed way to create realistic pedestrian patterns. On the other hand, where the number of pedestrians is too great, the macroscopic models may become more suitable due to the exponential increases during the calculation. Where interactions between pedestrians are not important, macroscopic models may be a suitable modeling approach. It is seen that the number of simulation software based on the Agent Based modeling approach is higher than the number of other model types. The most important reason for this is that the Agent Based modeling approach can provide more realistic results in modeling human behaviors.

Agent-based models define pedestrians as autonomous agents that can interact with each other and their environment. Based on agent-based modeling and simulation, the work is based on complex adaptive systems. Complex adaptive systems have the characteristics of autonomous individuals in the system, interaction between these individuals and adaptation of individuals to changing environment conditions. In complex systems, behaviors that are difficult to predict, which come into play due to the interaction of each individual in the system and their interactions, are called emergent behaviours. Agents are units that have the ability to make decisions in a complex adaptive system. Agents are autonomous individuals with the ability to receive information, process information, adapt to changing environment and make decisions in line with the rules they have. Agent-based modeling and simulation utilizes discrete event simulation and object oriented programming methods to manage complex adaptive systems. It is also referred to as the down-up modeling approach because it allows the generic behavior of the system to be revealed by one-to-one modeling of each agent in the system.

The Social force model is based on the assumption that the movements of the pedestrians are facilitated under the influence of certain forces. It is basically based on Newton's second fundamental law of motion. The model received this name because the internal motivations of the pedestrians are considered as forces. Although the social force model is lacking, it is a very valuable model for modeling pedestrian movements. Models in which the movements of the pedestrians in an emergency are modeled as robots, only physical ones as particles, will always be missing. The influence of factors such as psychological, sociological, educational, etc. they have under the behaviors exhibited by the victims in an emergency must not be ignored. In addition, pedestrians tend to interact with other pedestrians around as a social entity. It is seen that the effects of these factors and the behavioral situations arising from these factors must be examined when modeling the movements of the pedestrians in the emergency situations.

We can divide the behaviors exhibited by the pedestrians into two main groups: group behavior and individual behavior. It is unacceptable to ignore group behavior in modeling pedestrian behavior. The pedestrian has a tendency to exhibit largely group behavior from individual behaviors in emergencies.

In this thesis, a mathematical model has been created using AnyLogic to simulate escaping pedestrians behaviour during an emergency situation. The proposed model is based on the basic SKM model of Helbing and some changes and additions have been made in the direction of the group behavior suggested on Helbing's model in order to make the pedestrians behaviour more realistic. The model has additional

forces implemented both for improving social force model and to implement group behaviour.

Agent-based modeling approach has been used as modeling approach and the pedestrians has been modeled as acting agents acting under specific forces. Depending on the size of the determined step, pedestrians are accelerated under the influence of the forces acting on them and they changes their speed and position depending on this acceleration. Each simulated pedestrian is evaluated as an independet agent that move according to its local perception of the dynamic environment and the movements of the pedestrians are obtained by solving the ordinary differential equation which is formed by the forces acting on them by the Euler method. While the generated model is continuous in the two dimensional plane, time is discrete due to the numerical approach used to solve the ordinary differential equation.

In this work, the pedestrians are divided into groups and they allow to move in groups. The social distance theory and the K-Means clustering algorithm have been utilized while building pedestrian groups. In cases where the pedestrian density is less than 0.125, the pedestrians are grouped according to the social distance theory and the K-means clustering algorithm is used where pedestrian density is greater. Inspired by the Boids algorithm to calculate the perceived mass centers of the pedestrians in the group, and by applying an imaginary group force to the group members from the center of mass, the group members were made to move close to each other. In addition, the leading attractor force was added to the social force model in order to examine the leader behavior within each group to examine the behavior of the leader, a subtype of group behavior. A leader driving force is also added to prevent the leader from going to the target.

Modeling of pedestrians with different size and number of group members is provided. The creation of realistic pedestrian groups has been achieved with the help of the theory of social distance utilized in the formation of groups. In case of emergency, the feeling of hastiness of the pedestrians is included in the model by the desired speed being dependent on the time.

Simulation experiments were carried out to demonstrate the effectiveness of the proposed model. Different scenarios such as pedestrian number, exit number, exit width, presence of obstacles, location, group number and leader number were tested and evacuation times were obtained for all scenarios. Obtained evacuation times were obtained by taking Monte Carlo approach and taking averages of at least 100 runs with a margin of error of 0.05 in 95 confidence intervals.

To verify the proposed model proposed, two apprhach was applied. First, the model was compared with the studies in the literature and it was seen that the results gave similar results to the studies. Then, the situation of bringing the pedestrian phenomenon that emerged in an emergency into the model was examined, and it was observed that most of these cases were revealed.

In the further studies we aims to create more realistic pedestrian models with new models to be created based on pedestrian data analysis and fuzzy logic. We also consider to compare our model with some real evacuation experiments for validation.

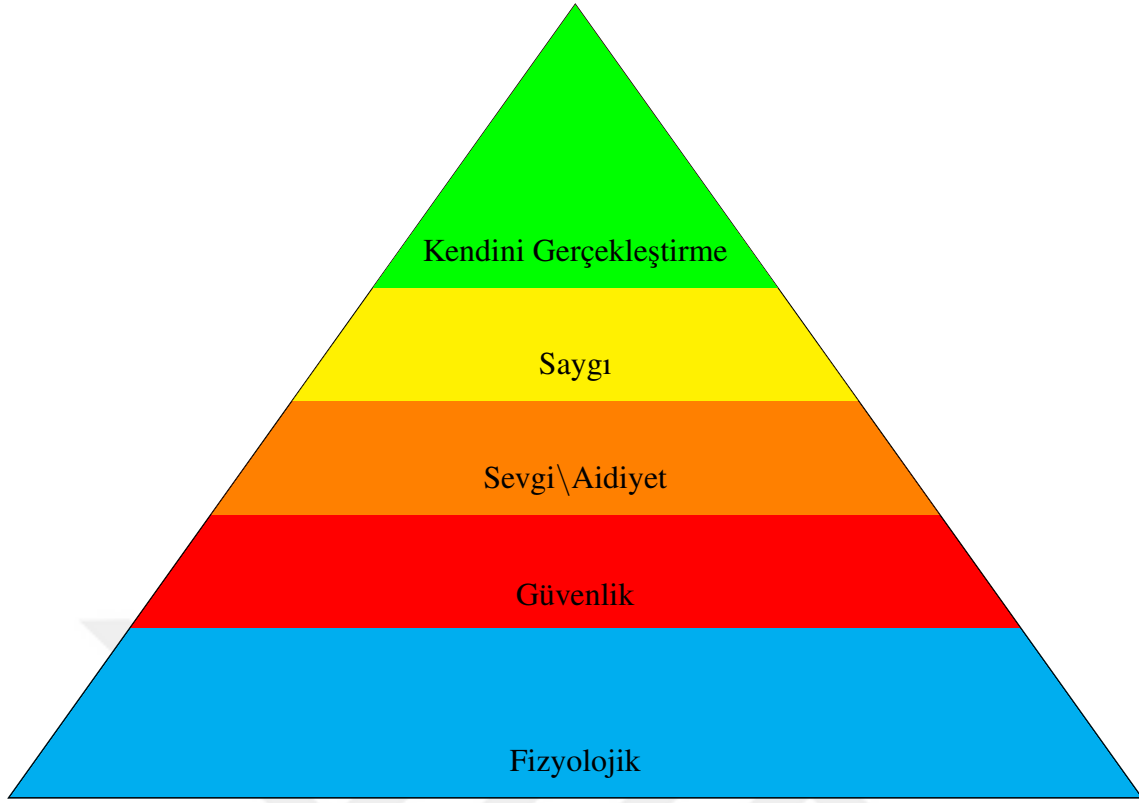


1. GİRİŞ

Güvenlik insanların temel ihtiyaçlarından birisidir. Şekil 1.1’de Maslovun ihtiyaçlar hiyerarşisinde belirttiği gibi güvenlik ihtiyacı piramidin 2. sırasında yer almaktadır. Öte yandan yangın, terör saldırısı ve doğal felaket gibi durumlar insanların ve diğer canlıların can ve mal güvenliği için büyük bir tehdit oluşturmaktadır. Bu tür felaketlerle yaşamımız boyunca çok sık karşılaşmamıza rağmen, karşılaşmamız durumunda hazırlıksız yakalanmamıza bağlı olarak çok dramatik sonuçlarla karşılaşabiliriz. Bu felaketlerden çoğu zaman kaçınmak mümkün olmadığından dolayı, karşılaşmamız anında olabileceklere karşı alınabilecek önlemler kayıpları en aza indirmek açısından faydalı olacaktır. Bu noktada yapıların (binaların) tahliye planlaması ile ilgili deneysel (tatbikat vb.) çalışmalar ve bilgisayar tabanlı analiz çalışmaları uygun metodlar olarak görülmektedir. Tahliye planlaması iteratif süreçleri içermekte olup, belirli bir alandaki kişilerin riskli bir durumda en kısa süre içerisinde bu alandan dışarı çıkarılmasına ve bu çıkışları sağlayacak en uygun rotaların tanımlanmasını sağlamayı amaçlar. Bilgisayar destekli analiz çalışmaları, deneysel(uygulamalı) çalışmalardan daha az maliyetli olduğu için, tahliye planlamasında daha yaygın bir şekilde literatürde kullanıldığı görülmektedir. Tahliye planlaması yaparken kullanılan bilgisayar destekli analiz çalışmalarını temelini matematiksel denklemler, bazı fizik kuramları ve simülasyon modelleri oluşturmaktadır. Bu çalışma kapsamında tahliye planlamasında önemli bir yere sahip olan felaket bölgesinden taşınan kişilerin (evacuee) hareketlerinin, seçimlerinin ve davranışlarının modellenmesi ve tahliye süresine olan etkisi incelenmiştir.

1.1 Tezin Amacı

İnsanların acil durumlar karşısında nasıl harekete geçtikleri, tahliye analizi çalışmalarında önemli bir yere sahiptir. Yaya dinamiklerinin ve davranışlarının modellenmesi, bu modeller üzerinde acil durumlarda gerçekleşebilecek olaylar ile ilgili deneysel çalışmalar yapmayı ve çıkarımlarda bulunmayı mümkün hale



Şekil 1.1 : Maslov'un ihtiyaçlar hiyerarşisi.

getirmektedir. Öte yandan yaya davranışlarının mevcut modellere dahil edilmesi ile daha gerçekçi yaya modelleri elde edileceği düşünülmektedir. Belirtilen bu temel noktalar bu tezin konusunun belirlenmesindeki çıkış noktalarıdır.

Bu çalışmanın amacı ise acil durumlarda bir yeri terk eden (biryerden kaçan) yayaların davranışlarını modellemektir. Bu bağlamda yaya davranışlarından biri olan grup davranışı ve grup davranışının bir türü olan lideri takip etme davranışı modellenmeye (matematiksel ve simülasyon modeli) çalışılmıştır. Bu tezin diğer bir amacı ise grup davranışlarını modellemede çalışma içerisinde kullanılan temel SKM'nin eksik yanlarını iyileştirip, daha gerçekçi bir yaya davranışı modeli ortaya koymaktır.

1.2 Tezin Bölümleri

Bu tez çalışması 7 bölümden oluşmaktadır. Tezin birinci bölümde tezin konusu ile ilgili giriş yapılmış ve tezin amacı aktarılmaya çalışılmıştır. Tezin ikinci kısmında literatürde yaya hareketlerinin modellenmesi ile ilgili yaklaşımlar hakkında bilgi verilmiştir. Tezin üçüncü bölümünde çalışma kapsamında kullanılan Sosyal Kuvvet Modeli detaylı bir şekilde açıklanmıştır. Daha sonra normal ve acil durumlarda

meydana gelen yaya hareketi olguları incelenmiştir. Tezin dördüncü bölümünde grup davranışı hakkında literatür taramasına yer verilmiş ve Boids algoritması anlatılmıştır. Tezin beşinci kısmında ise çalışma kapsamında önerilecek olan modelin önceliklerine kısaca değindikten sonra önerilen model verilmiştir. Tezin altıncı kısmında ilk olarak önerilen modelin doğrulanmasına yönelik bazı yaklaşımlar sunulmuş, daha sonra ise önerilen model için farklı senaryolar için deneyler yapılmış ve bu deneylerin sonuçlarına yer verilmiştir. Tezin son kısmında ise Bölüm 6’da elde edilen deney sonuçlarının gerçek hayatta nasıl kullanılabileceği, tez çalışması ile ilgili genel sonuçlar, çalışmanın öne çıkan yanlarına değinilmiş ve gelecekte yapılması düşünülen çalışmalar hakkında öneriler sunulmuştur.



2. YAYA HAREKETİ MODELLERİ

Bu bölüm yaya hareketlerinin modellenmesinde literatürde karşılaşılan modellerin tanıtılması ve sınıflandırılmasına yöneliktir.

2.1 Modellerin Sınıflandırılması

Literatür incelemesi sonuçlarına göre yapılan çalışmalarda modellerin sınıflandırılmasında çeşitli yaklaşımlar olduğu göze çarpmaktadır. Bunlar arasında en yaygın sınıflandırma türleri; Mikroskopik-makroskopik, homojen-heterojen, kesikli-sürekli ve stokastik-deterministik modellerdir. Bu sınıflandırmalarda kullanılan isimler modele ait bir durumu ya da model içerisindeki bir özelliğe işaret etmektedir. Bu çalışma içerisinde ise makroskopik ve mikroskopik ve mezoskopik başlıkları altında modeller tanıtılacak ve incelenecektir.

2.2 Mikroskopik Modeller

Kalabalık içerisinde yer alan her bir yayanın hareketlerinin modellenmesine odaklanılır. Bütün içerisinde yer alan her bir yayanın davranışları modellenerek, kalabalığın davranışları ortaya çıkarılmaya çalışılır. Bu tip modeller esasında ajan temelli modellemeye dayanmaktadır. Ajan temelli modellemenin en basit hali olan, hücresel otomat modeli, kuvvete dayalı modeller(sosyal kuvvet modeli vb.) ve oyun teorisine dayalı modeller mikroskopik modeller içerisinde yer almaktadır.

2.2.1 Ajan temelli benzetim

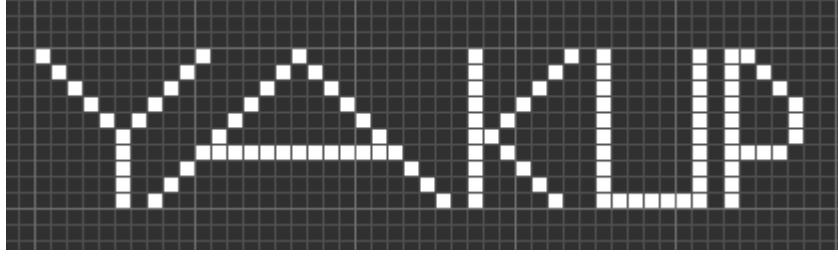
Ajan temelli modelleme ve benzetimin temelleri kompleks uyarlanabilir sistemlerle ilgili çalışmalara dayanmaktadır. Kompleks uyarlanabilir sistemler, sistem içerisinde yer alan otonom bireyler, bu bireyler arasında oluşan etkileşim ve bireylerin değişen ortam koşullarına uyum sağlayabilmesi özelliklerine sahiptir. Kompleks sistemlerde sistem içerisinde yer alan bireylerin biraraya gelmesi ve aralarındaki etkileşim ile meydana gelen ve önceden tahmin edilmesi zor olan davranışlar "beliren davranışlar"

olarak adlandırılır. Ajanlar kompleks uyarlanabilir sistemde karar verme yeteneğine sahip birimlerdir. Ajanlar sahip oldukları kurallar doğrultusunda bilgi alma, alınan bilgiyi işleme, değişen ortama uyum sağlama ve karar verme yeteneğine sahip otonom bireylerdir. Ajan temelli modelleme ve simülasyon, kompleks uyarlanabilir sistemleri yönetebilmek için kesikli olay simülasyonu ve nesneyi yönelik programlama yöntemlerinden yararlanır. Sistem içerisinde yer alan her bir ajanın birebir modellenmesi ile sistemin genel davranışlarını ortaya çıkarmayı sağlaması nedeniyle aşağıdan-yukarıya modelleme yaklaşımı olarak da adlandırılır [1].

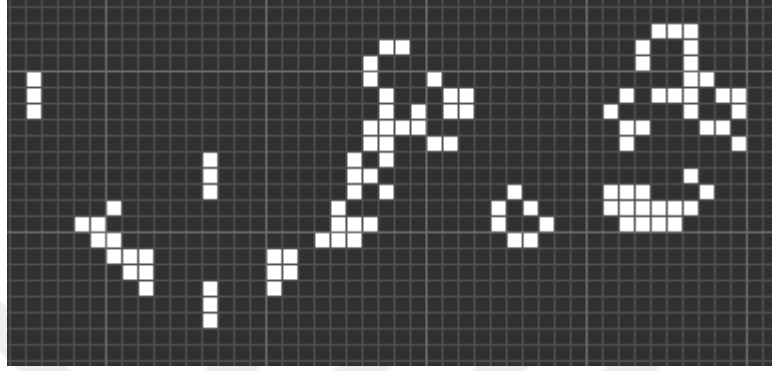
2.2.1.1 Hücresel otomat modeli

Ajan temelli modelleme ve simülasyon yaklaşımına ait temel olguyu örneklemenin en basit yolu olarak kabul edilebilir [1]. Modelin temeli fizikçi Stanislaw'ın, 21.yüzyılda yaşamış bir matematikçi olan John von Neumann'a sorduğu şu soruya cevap olarak geliştirilmiştir. "Bir makine kendi kopyasını yapmak üzere programlanabilir mi?". Soru temelinde, kompleks bir sistemin, kendisi gibi kompleks bir sistemi oluşturup oluşturamayacağını içermektedir. Bu sorunun cevabı hücresel otomat formunda bir makinenin soyut matematiksel temsili yardımıyla evet olarak verilmiştir.

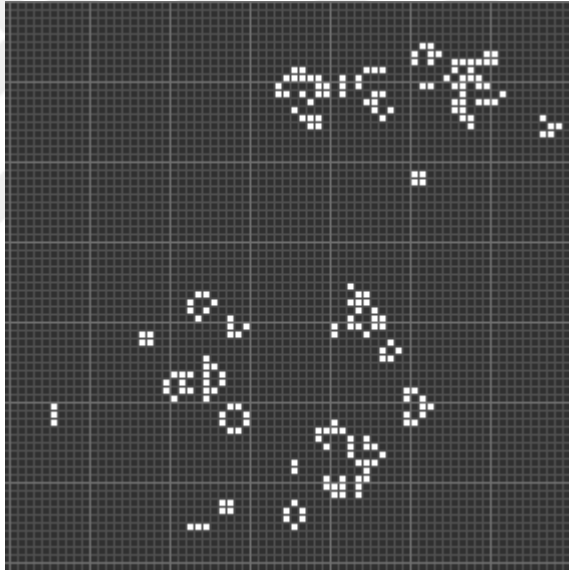
Hücresel otomat sistemi genellikle iki boyutlu ızgaralar şeklinde hücrelere bölünmüş bir yapıya sahiptir. Her hücrenin belirli bir t anında sınırlı sayıdaki durumlardan birine sahip olabileceği kabul edilir. Ve daha önceden belirlenmiş olan bazı kurallara bağlı olarak hücreler sahip oldukları değerleri ya da durumlarını değiştirirler. Her hücre belirli zaman periyotlarında durumunu günceller. Hücrenin sahip olacağı bir sonraki değer hücrenin o anki durumuna ve etrafında yer alan komşu hücrelerin sahip olduğu değere bağlı olarak değişir. Aynı duruma sahip hücre ve komşu hücreler için yeni durumda sahip olacak değerler her zaman için aynı şekilde güncellenmektedir. Bu da bize hücresel otomatın deterministik bir yaklaşım olduğunu göstermektedir. John Conway'ın geliştirdiği "Game of Life", hücresel otomat modeli için güzel bir örnektir [1]. "Game of Life" simülasyonuna ait bir örneğe ait ekran görüntüleri Şekil 2.1'de görülmektedir. Simülasyon çalışmasında başlangıç popülasyonu verilmiş ve daha sonraki jenerasyonlarda popülasyonun yeni durumu gösterilmiştir.



(a) Başlangıç Durumu



(b) Jenerasyon 11



(c) Jenerasyon 205

Şekil 2.1 : Game of Life simülasyonuna ait anlık ekran görüntüleri.

2.2.1.2 Kuvvete dayalı modeller

Bu modeller yayaları birer parçacık olarak düşünmekte ve yayaların hareketlerinin itici ve çekici bazı kuvvetlerin etkisi altında gerçekleştiği fikrine dayanmaktadır. Bu modellerin ortaya çıkarılmasında bazı fiziksel kuramlardan yararlanıldığı görülmektedir. Kuvvete dayalı modeller arasında en yaygın olarak bilinen model Helbing tarafından ortaya atılan sosyal kuvvet modelidir. Sosyal kuvvet modeli

Newtonun ikinci temel hareket yasasına dayanmaktadır. Sosyal kuvvet modeline benzer bir yapıya sahip olmakla birlikte bazı farklılıkları da içinde barındıran diğer bir kuvvet modeli ise Merkezkaç kuvvet modelidir. Modelde merkezkaç kuvveti, itici bir kuvvet olarak ele alınmış ve bu kuvvetin yayaların ilerleme yönünü ve göreceli hızını etkilediği düşünülmüştür [2]. Fiziksel kuramlardan yararlanılarak kurulan kuvvete dayalı diğer bir model ise Okazaki ve arkadaşları tarafından önerilen Manyetik kuvvet modelidir. Önerilen modele göre yayaların hareketleri manyetik alanda hareket eden manyetik nesnelerin hareketine benzetilmiştir. Modelde itici ve çekici kuvvetler manyetik alan teorisindeki eksi ve artı kutuplara göre belirlenmiştir. Yayalar ve yayaların uzak durması gereken engeller yani itici nesneler pozitif kutuplar olarak belirlenmiş iken, yayaların ulaşmak istedikleri çıkış noktası gibi hedef noktalar, yani çekici nesneler negatif kutuplar olarak tanımlanmıştır [3].

2.2.2 Kural tabanlı modeller

Parçacıkların hareketlerinin belirli kurallara bağlı olarak oluşturulduğu modellerdir. Sürü algoritmaları kural tabanlı modeller içerisinde yer almaktadır. Yaya yoğunluğunun fazla olmadığı durumlarda sürü halinde hareket eden yayaların gerçekçi davranışlar sergilediği görülmektedir. Fakat yaya yoğunluğunun fazla olduğu durumlarda yayaların birbirlerine itme davranışını sergilemede yetersiz kalmaktadır. Yani yayaların arasındaki etkileşimi dikkate almamaktadır. Reynolds'ın [4] Boids algoritması kural tabanlı modellere örnek olarak verilebilir.

2.2.3 Oyun teorisine dayalı modeller

Oyun teorisi matematiğin bir dalı olup, sosyal etkileşim içerisinde giren oyuncuların seçimlerinin rasyonel kararlara göre şekillenmesine dayanır. Yayaların tahliye esnasındaki seçimleri de oyun teorisi yaklaşımına göre modellenebilir. Lo [5], yayaların acil durumlardaki çıkış seçimi sürecinin rekabetçi oyun teorisine dayalı olarak gerçekleştiği bir model önerisinde bulunmuş ve bu modeli "Fine-Grid" tahliye modeli ile entegrasyonunu sağlamıştır. Mesmer [6] geleneksel davranış odaklı yaya hareketi modellerine alternatif olarak karar verme sürecine dayalı yeni bir yaya hareket modeli önermiştir. Önerilen modelde karar teorisi ve oyun teorisi kullanılarak yayaların karar verme süreci modellenmiştir. Guan [7] tahliye sürecinde yayalar

arasında oluşan çatışmalardan kaynaklı karmaşık davranışları modellemek amacıyla "Snowdrift" oyun teorisini hücresel otomat yöntemiyle birlikte kullandığı bir model önerisinde bulunmuştur.

2.3 Makroskopik Modeller

Mikroskopik modellerin aksine, yayaların bireysel davranışlarını modellemek yerine tüm yayaları tek bir varlık olarak kabul eder ve bu varlığı modellemeye çalışır.

2.3.1 Kafes gaz modeli

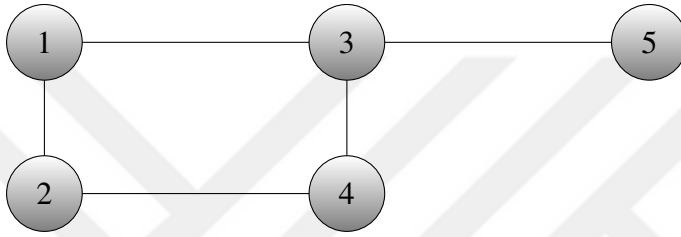
İlk olarak 1986 yılında Frisch ve arkadaşları tarafından akışkanlar dinamiği alanında sıvı içerisinde moleküllerin hareketlerini daha gerçekçi bir şekilde ortaya koyabilmek amacıyla kafes gaz modelini önermişlerdir. Model üçgensel bir kafes ve altıgen simetrisine sahip ızgara yapısına sahiptir ve bu yapısıyla hücresel otomat yaklaşımına benzemektedir. Belirli bir yöne gitme eğiliminde olan yayalar üçgen bir kafes içerisinde hareket eden birer parçacık olarak modellenmektedir. Yayalar sahip oldukları yön arzusu yanında akış yönünde gitme eğilimine de sahiptirler. Model iki temel dinamiğe sahiptir. Bu dinamiklere göre hareket halindeki parçacıklar hareket alanı içerisinde komşu hücrelere doğru hareket etmekte ve bu komşu hücrelerde meydana gelen çarpışma ve bu çarpışma sonucunda farklı yönlere yayılma davranışı göstermektedir.

2.3.2 Kuyruk modelleri

Kuyruk modellerinde yayaların hareket alanları ağ şeklinde modellenmekte olup bu alanlarda yaya akışının modellenmesinde kuyruk teorisinden faydalanılmaktadır. Ağ içerisinde yer alan her bir düğüm ve bu düğümler arasında oluşan hareket ve bekleme süreleri kuyruk teorisine göre belirlenmektedir. Bakuli [8] vd. tahliye problemini ele alırken binaların tasarımı aşamasında kullanılabilecek bir model önerisinde bulunmuştur. Kaynak paylaşımı algoritması ve kuyruk teorisine dayalı önerilen model yardımıyla normal ve acil durumlar esnasında yaya trafiğini etkin bir şekilde sağlayacak optimum kaynak paylaşımı belirlenmeye çalışılmıştır.

2.3.3 Ağ tabanlı modeller

Yayaların haket alanları olan tesislerin alt bölümlere bölünmesi ile oluşan bu bölümlerin her birinin düğüm olarak tanımlandığı ve bu düğümler arasındaki bağlantıların ise bu bölümler arasındaki geçişleri gösterdiği modellerdir. Şekil 4.2' de 4 odadan oluşan bir binaya ait ağ yapısı verilmiştir. Burada 5 numaralı düğüm çıkışı temsil etmektedir. Okazaki yaya ortamının ağ şeklinde modellendiği ve yayaların bu ağ içerisindeki hareketlerinin kuyruk ağ süreci yardımıyla tanımlandığı stokastik bir yaya trafiği akış modeli önermiştir [3].



Şekil 2.2 : Bina kuyruk ağ gösterimi

2.3.4 Akış tabanlı modeller

Kalabalık simülasyonu problemine global bir pencereden (genel) bakmaktadır. Yayalar bireysel varlıklar olarak değil bir akışın parçası olarak modellenirler. Model içerisinde akışın fiziksel özellikleri ve çevresel özellikleri tahliye senaryoları içerisinde tanımlar. EVACNET4 akış tipi modele bir örnektir. Ortam düğümlerden oluşan bir ağ olarak tasarlanır. Düğümler oda, koridor vb. sınırlandırılmış parçalardır. Düğümlerin durumu parametreler tarafından belirlenir. Bu parametreler doluluk, kişiler arası ortalama mesafe vb. şeylerdir. İki parça arasında yer alan giriş yerleri, iki düğüme denk gelen bir bağlantı arkı olarak modellenir. Kullanıcı her bir arkın akış kapasitesini ve geçiş süresini tanımlar. Simülasyonun hazırlık aşamasında kullanıcı her bir düğümün başlangıç durumunu tanımlamak zorundadır (Her bir düğümdeki insan sayısı, diğer parametrelerin durumu ve hedef düğümleri). Sonuç olarak model ortama ait geometriyi net bir şekilde açıklamaz, sadece ağ düğümlerini kullanır. Yani yol planlaması geometrik pencereden incelenmez. EVACNET4 minimum tahliye süresi için araştırma yapar ve ileri ağ akışı aktarma algoritmaları yardımıyla çözüm bulmaya çalışır. EESCAPE ve FireWind akış tipi modeller için diğer örneklerdir.

2.3.5 Akışkan dinamiği modelleri

Yapılan gözlem sonuçlarına göre yaya dinamiklerinin akışkanlarla ve gazlarla benzer özelliklere sahip olduğu görülmüştür. Bu benzer özelliklerin bazıları incelenen çalışmalardan elde edilen sonuçlara göre aşağıdaki gibidir [9].

- Yayaların kardaki ayak izleri akışkanların akış çizgilerine benzemektedir.
- Zıt yürüme yönlerinde sınır bölgelerde, vizkoz dokunma durumunun görüldüğü gözlemlenmiştir.
- Karşı tarafa geçen yayaların hareketlerinin, bir nehirin akışına benzediği görülmüştür.
- Kalabalık içerisinde hareket etmeye çalışan yayaların davranışları şok dalgalarının yayılmasına benzemektedir.

2.4 Mezoskopik Modeller

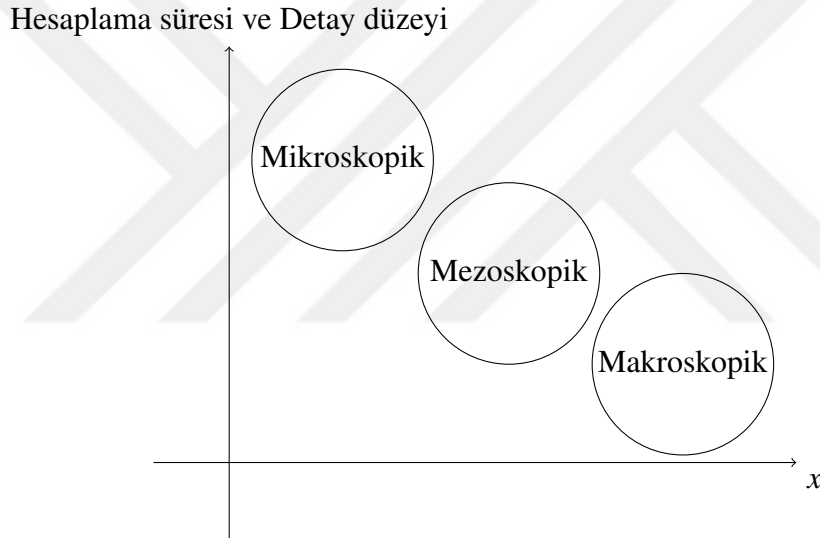
Mezoskopik modeller, mikroskopik ve makroskopik modellerin her ikisinin de karakteristik özelliklerinin bazılarını sahip olup bu iki yaklaşımın bir kombinasyonudur. Yayaların mekansal hareketleri her bir yayaya özgü olarak tanımlanırken, yayaların hareketleri kalabalığın toplu akışına bağlı olarak gerçekleşir. Yani her bir yayanın kararları modellenir, fakat davranışları belirlenirken bütüncül ölçüler dikkate alınır. Modelleme yaklaşımı olarak genellikle Hücresel Otomat yönteminin kullanıldığı görülmektedir. Henderson [10] yaptığı çalışmada yayaların hareketleri ile akışkanların davranışı arasında bir benzerlik olduğunu öne sürmüştür. Hoogendoorn [11] gaz kinetiği modeline dayalı bir yaya hareketi davranışı modeli ortaya koymuştur.

2.5 Modellerin Karşılaştırılması

Üç temel başlık altında toplanan modelleri incelediğimizde modeller arasında hesaplama karmaşıklığı ve detay düzeyi açısından farklılıklar olduğu görülmektedir. Şekil 2.3'te görüldüğü üzere Mikroskopik modeller en fazla hesaplama zamanına ihtiyaç duyarken, daha detaylı diğer bir deyişle gerçekçi yaya modelleri oluşturmaya imkan sunmaktadır. Öte yandan yaya sayısının çok fazla olduğu durumlarda hesaplama süresindeki exponansiyel artışlardan ötürü, makroskopik modeller daha

uygun bir hale gelebilir. Yayalar arasında meydana gelen etkileşimlerin önemli olmadığı durumlarda da makroskopik modeller uygun bir modelleme yaklaşımı olabilir. Literatürde yer alan yaya modelleme yaklaşımları ve yazılımlarına ait sınıflandırmalar sırasıyla Çizelge 2.1 ve Çizelge 2.2’de verilmiştir.

Çizelge 2.2’ yi incelediğimiz zaman Ajan Temelli modelleme yaklaşımına dayalı simülasyon yazılımlarının diğer model tiplerinden sayıca daha üstün olduğu görülmektedir. Bunun en önemli sebebi ise Ajan Temelli modelleme yaklaşımının insan davranışlarını modellemede daha gerçekçi sonuçlar ortaya koyabilmesinden kaynaklanmaktadır. Öte yandan zaman açısından sahip olduğu komplekslikten dolayı sayıları çok fazla olmasa da mezoskopik (hibrid) modellerle ilgili çalışmaların da olduğu görülmektedir.



Şekil 2.3 : Hesaplama süresi, detay düzeyi ve model ilişkisi

Çizelge 2.1 : Model sınıflandırması.

Model	Model Tipi	Homojen/Heterojen	Kesikli/Sürekli
Hücreli Otomat	Mikroskopik	Heterojen	Kesikli
Kuvvete Dayalı Modeller	Mikroskopik	Heterojen	Sürekli
Kuyruk Modelleri	Makroskopik	Heterojen	Kesikli
Akış Tipi Modeller	Makroskopik	Homojen	Kesikli
Kafes Gaz Modeli	Makroskopik	Homojen	Sürekli
Ağ Tabanlı Modeller	Makroskopik	Homojen	Sürekli
Akış Tabanlı Modeller	Makroskopik	Homojen	Sürekli
Gaz Kinetiği Modeli	Mezoskopik	Homojen	Sürekli

Çizelge 2.2 : Tahliye Simülasyonu Yazılımları ve Sınıflandırılması

Model	Model Tipi	Uzay ve Zaman	Modelleme Yaklaşımı
ALLSAFE	Makroskopik	Sürekli	Akış Tabanlı
Egress-Pro	Makroskopik	Sürekli	Akış Tabanlı
E-SCAPE	Makroskopik	Sürekli	Akış Tabanlı
EVACNET4	Makroskopik	Sürekli	Ağ/Akış Tabanlı
FPETool	Makroskopik	Sürekli	Akış Tabanlı
TIMTEX	Makroskopik	Sürekli	Akış Tabanlı
Egress Complexity	Mezoskopik	Sürekli	Akış/Ajan Tabanlı
PathFinder	Mezoskopik	Kesikli	Hücresele otomat
PedSim	Mezoskopik	Sürekli	Ajan Temelli
WAYOUT	Mezoskopik	Coarse	Akış Tabanlı
ASERI	Mikroskopik	Sürekli	Ajan Temelli
BGRAF	Mikroskopik	Kesikli	Ajan Temelli
BldEXO	Mikroskopik	Kesikli	Ajan Temelli
CRISP	Mikroskopik	Kesikli	Ajan Temelli
CrowdDMX	Mikroskopik	Sürekli	Ajan Temelli
EGRESS 2002	Mikroskopik	Kesikli	Hücresele Otomat
EvacSim	Mikroskopik	Sürekli	Ajan Temelli
EXIT89	Mikroskopik	Sürekli	Ağ Tabanlı
EXITT	Mikroskopik	Sürekli	Ajan Temelli
EXODUS	Mikroskopik	Kesikli	Ajan Temelli
FDES+Evac	Mikroskopik	Sürekli	Ajan Temelli
GridFlow	Mikroskopik	Sürekli	Ajan Temelli
Legion	Mikroskopik	Sürekli	Ajan Temelli
Magnetic Model	Mikroskopik	Kesikli	Ajan Temelli
MASSEgress	Mikroskopik	Sürekli	Ajan Temelli
Myriad	Mikroskopik	Sürekli	Ajan Temelli
PED/PAX	Mikroskopik	Sürekli	Akış Tabanlı
PedGo	Mikroskopik	Kesikli	Hücresele Otomat
SGEM	Mikroskopik	Kesikli	Hücresele Otomat
Simulex	Mikroskopik	Sürekli	Ajan Temelli
STEPS	Mikroskopik	Kesikli	Ajan Temelli

Not: Sınıflandırma [12], [13], [14] kaynaklarında verilen bilgilere dayanarak yapılmıştır.

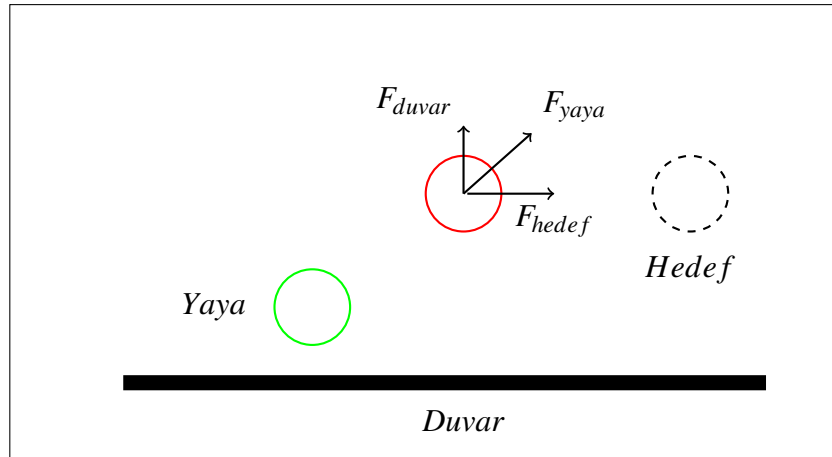


3. SOSYAL KUVVET MODELİ

Bu bölüm Sosyal Kuvvet Modeli'nin tanıtılmasına ve yaya hareketi hakkında temel olguların (normal ve acil durumlar) incelenmesine yöneliktir.

3.1 Sosyal Kuvvet Modeli Temel Yapısı

Sosyal kuvvet modeli yayaların hareketlerinin belirli kuvvetlerin etkisi altında ortaya çıktığını ileri sürmektedir. Temel olarak Newton'un ikinci temel yasası olan hareket yasasına dayanmaktadır. Bu kuvvetler yayanın bulunduğu ortam tarafından doğrudan uygulanmamakta olup, yayaların içsel motivasyonlarını etkileyerek yayaların harekete geçmesine neden olmaktadır [15]. Öte yandan yaya yoğunluğunun çok fazla olduğu ortamlarda, yayaların diğer yayalarla ve sınırlarla (duvar, engel vb.) fiziksel temas kurması halinden bunlar tarafından uygulanan kuvvetler de mevcuttur. Bu kuvvetler ise birer dışsal kuvvet olarak düşünülebilir.



Şekil 3.1 : Sosyal Kuvvet Modeli Gösterimi

Helbing tarafından önerilen sosyal kuvvet modeli temelde 3 farklı kuvvet bileşenine dayanmaktadır. Bu kuvvetler yayaların hareketlerinin oluşmasında en temel ve gerekli olan kuvvet bileşenleri olarak görülmüştür. Bu kuvvetlerden birincisini yayaların sahip olduğu belirli bir yere gitme arzusu oluşturmaktadır. İkincisi (itici kuvvetler), yayaların hareketleri sırasında önlerine çıkan engellerden ve diğer yayalardan uzak durma (belirli bir mesafeyi koruma) isteğinden kaynaklanmaktadır. Üçüncü kuvveti

(çekici kuvvetler) ise yayaların belirli yayalarla olan yakın ilişkilerinden dolayı bu yayalara yakın durup bu yayalarla birlikte hareket etme isteği ya da diğer bazı cansız etmenler (pencere vb.) tarafından kendilerine çekilme isteği oluşturmaktadır.

3.2 Sosyal Kuvvet Modeli Formülasyonu

Modelin formülasyonuna başlamadan önce modelin temelini oluşturan Newtonun ikinci yasasına değinilmesi, modelin konseptini anlamak açısından faydalı olacaktır. Newtonun hareket yasasına göre cisimler kendilerine uygulanan kuvvetlerin etkisi altında uygulanan kuvvetin büyüklüğüne ve doğrultusuna bağlı olarak bir ivme kazanmaktadır. Cisim üzerine uygulanan net kuvvet ve ivme arasındaki ilişki Denklem 3.1’de görülmektedir. Öte yandan fizikte yer alan mekanik kanunlarına göre cisimlerin belirli bir t anındaki ivmeleri, hızları ve konumları arasındaki ilişki Denklem 3.2’de görüldüğü gibidir. Buna göre belirli kuvvetlerin etkisi altında ivmelenen cismin belirli bir t anındaki konumunu Denklem 3.2’de verilen ikinci dereceden adi diferansiyel denklemi çözerek bulabiliriz.

$$F = ma \quad (3.1)$$

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2x}{dt^2} \quad (3.2)$$

SKM ile ilgili üç farklı formülasyon bulunmaktadır. Bunlardan ilki kapalı formda verilmiş olup anlaşılması biraz daha güçtür [15]. Diğer ikisi [16] [17] ise daha açık bir şekilde ifade edilmiş olup formülasyonu daha anlaşılır bir formdadır. Bu üç form da içerdiği formülasyon ve gösterim farklılıklarına rağmen üç temel kuvvet olan yönlendirici kuvvet, itici kuvvetler ve çekici kuvvetlerden oluşan bir yapıya sahiptir. Bu çalışmada acil durumlardaki yaya davranışları ile ilgilenildiğinden dolayı [16] çalışmadaki formülasyon kullanılmıştır.

3.2.1 Yönlendirici kuvvet

Yayalar mümkün olduğu kadar hızlı ve rahat bir şekilde arzu ettikleri noktaya ulaşmak istemektedirler. Bu nokta $\vec{a}_\alpha^0$ ile gösterilmektedir. Yayaların bu noktaya ulaşırken en kısa yolu kullanacakları ve bu yolun da çokgen şeklinde olduğu düşünülmüştür. Bu çokgenin uç noktaları $\vec{a}_\alpha^1, \vec{a}_\alpha^2, \dots, \vec{a}_\alpha^n$ ile gösterilmektedir. Yönlendirici kuvvetin büyüklüğü Denklem 3.3a’deki gibi bulunmaktadır.

$$\vec{F}_{\alpha}^{\text{Yönlendirici}}(t) = m_{\alpha} \frac{v_{\alpha}^0(0)\vec{e}_{\alpha}(t) - \vec{v}_{\alpha}(t)}{\tau} \quad (3.3a)$$

$$v_{\alpha}^0(t) = [1 - n_{\alpha}(t)]v_{\alpha}^0 + n_{\alpha}(t)v_{\alpha}^{\text{max}} \quad (3.3b)$$

$$n_{\alpha}(t) = \frac{\bar{v}_{\alpha}(t)}{v_{\alpha}^0} \quad (3.3c)$$

$$\vec{e}_{\alpha}(t) = \frac{\text{hedef} - \text{konum}}{||\text{hedef} - \text{konum}||} \quad (3.3d)$$

Çizelge 3.1 : Yönlendirici Kuvvet Formülasyonu Terminolojisi.

Parametre	Birimi	Açıklaması
$v_{\alpha}^0(0)$	m/sn	α yayasının başlangıçta ulaşmak istediği hız
$v_{\alpha}^0(t)$	m/sn	α yayasının t anında ulaşmak istediği hız
$\bar{v}_{\alpha}(t)$	m/sn	α yayasının t anına kadar geçen süredeki ortalama hızı
$n_{\alpha}(t)$	skaler	α yayasının t anındaki sabırsızlık ölçüsü
v_{α}^{max}	m/sn	α yayasının ulaşabileceği maksimum hız
$v_{\alpha}(t)$	m/sn	α yayasının t anındaki hızı
m_{α}	kg	α yayasının kütlesi
$\vec{e}_{\alpha}(t)$	vektör	α yayasının t anındaki hareket yönü vektörü
τ_{α}	sn	Gevşeme süresi
$\vec{F}_{\alpha}^{\text{Yönlendirici}}(t)$	Newton	t anında α yayasına etki eden yönlendirici kuvvet

3.2.2 İtici kuvvet

Yayaların diğer yayalarla ve engellerle çarpışmasını önlemeye yöneliktir. İçsel ve dışsal itici kuvvetler olmak üzere 2 alt başlığa ayrılmıştır.

3.2.2.1 İçsel itici kuvvet

Yayaların hareketleri diğer yayaların hareketinden etkilenmektedir. Yayalar hareketleri esnasında diğer yayalar ile aralarında belirli bir mesafe bırakacak şekilde hareket etmektedir. Bu mesafe mevcut yaya yoğunluğuna ve yayanın arzu ettiği hıza bağlı olarak değişmektedir. Bu yüzden her bir yaya için özel bir küresel bölge tanımlanmıştır. Yayaların diğer yayalardan uzak kalma isteği itici bir kuvvet olarak tanımlanmış olup α yayasına β yayası tarafından uygulanan bu içsel itici kuvvet Denklem 3.4’de görüldüğü gibi tanımlanmıştır. Yayalar diğer yayalardan uzak durma isteğine sahip oldukları gibi duvarlar, engeller vb., ile de arasında bir mesafe bulundurma isteğine sahiptir. Buna bağlı olarak duvar, engel vb., kaynaklı içsel itici kuvvet Denklem 3.5 şeklinde tanımlanmıştır.

$$\vec{F}_{\alpha\beta}^{\text{ik}}(t) = A_{\alpha} \exp\left[\frac{r_{\alpha\beta} - d_{\alpha\beta}}{B_{\alpha}}\right] \vec{n}_{\alpha\beta} \left(\lambda_{\alpha} + (1 - \lambda_{\alpha}) \frac{(1 + \cos(\varphi_{\alpha\beta}))}{2}\right) \quad (3.4)$$

$$\vec{F}_{\alpha w}^{iik}(t) = A_{\alpha} \exp\left[\frac{r_{\alpha} - d_{\alpha w}}{B_{\alpha}}\right] \vec{n}_{\alpha w} \quad (3.5)$$

$$n_{\alpha\beta} = (x_{\alpha} - x_{\beta})/d_{\alpha\beta} \quad (3.6)$$

Çizelge 3.2 : İçsel İtici Kuvvet Formülasyonu Terminolojisi.

Parametre	Birimi	Açıklaması
A_{α}	skaler	Yaya bölgesel etkileşim gücü
B_{α}	skaler	Yaya bölgesel etkileşim aralığı
r_{α}	metre	α yayasının yarıçapı
$\vec{n}_{\alpha\beta}$	vektör	β yayasından α yayası doğrultusunda normalize edilmiş vektör
$r_{\alpha\beta}$	metre	α ve β yayalarının yarıçapları toplamı
$d_{\alpha\beta}$	metre	α ve β yayalarının kütle merkezleri arasındaki uzaklık
$\vec{n}_{\alpha w}$	vektör	w duvarından α yayası doğrultusunda normalize edilmiş vektör
$d_{\alpha w}$	metre	α yayasının w duvarı sınırına olan dik uzaklığı
λ_{α}	skaler	α yayasının anizotropik karakteristiği
$\vec{F}_{\alpha\beta}^{iik}(t)$	Newton	α yayasına β yayası tarafından uygulanan içsel itici kuvvet
$\vec{F}_{\alpha w}^{iik}(t)$	Newton	α yayasına w duvarı tarafından uygulanan içsel itici kuvvet

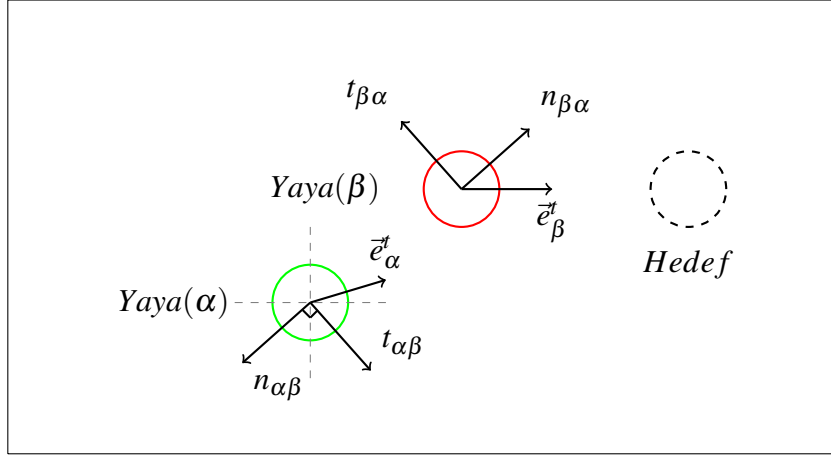
3.2.2.2 Dışsal itici kuvvet

Bu kuvvet özellikle panik durumunda ve yaya yoğunluğunun çok fazla olduğu durumlarda yayaların birbirleriyle ve duvarlarla(sınırlarla) etkileşime girmeleri durumunda meydana gelen fiziksel etkileşim kuvvetini ortaya koymaktadır. Bu kuvvet diğer yayalarla ve sınırlarla meydana gelen fiziksel etkileşim sonucunda ortaya çıkmaktadır. Yayalar için kuvvet Denklem 3.7a şeklinde tanımlanmıştır. Denklemin sol toplam kısmı beden kuvvetini temsil ederken, sağ taraf kinetik sürtünme kuvvetini temsil etmektedir. Denklem içerisinde yer alan kavramlara ait formülasyon ve açıklamalar için Denklem 3.7 alt denklem grubu, Şekil 3.2 ve Çizelge 3.1 incelenebilir. Sınırlar için bu kuvvetin hesaplanması benzer olmakla birlikte farklılıklar; Denklem 3.8 alt denklem grubu, Şekil 3.3 ve Çizelge 3.1’de incelenebilir.

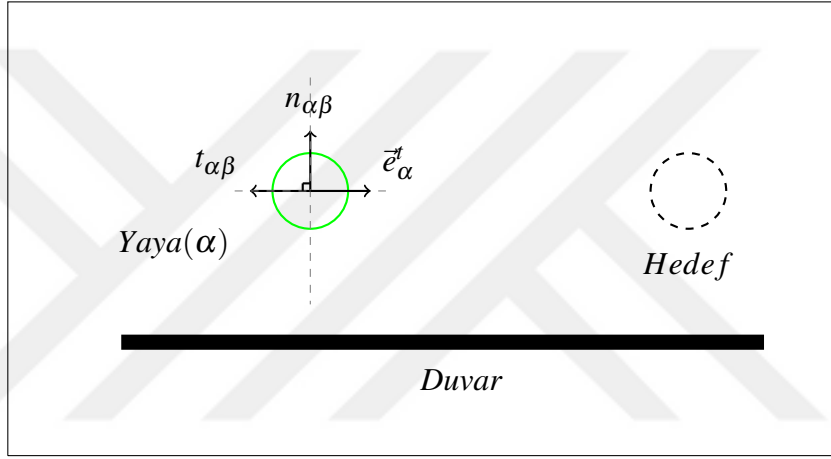
$$\vec{F}_{\alpha\beta}^{dik} = C_1 H(r_{\alpha\beta} - d_{\alpha\beta}) n_{\alpha\beta} + C_2 H(r_{\alpha\beta} - d_{\alpha\beta}) \Delta_{\beta\alpha}^t t_{\alpha\beta} \quad (3.7a)$$

$$H(r_{\alpha\beta} - d_{\alpha\beta}) = \begin{cases} 0 & r_{\alpha\beta} - d_{\alpha\beta} < 0 \\ r_{\alpha\beta} - d_{\alpha\beta} & r_{\alpha\beta} - d_{\alpha\beta} \geq 0 \end{cases} \quad (3.7b)$$

$$\Delta_{\beta\alpha}^t = (v_{\beta}(t) - v_{\alpha}(t)) t_{\alpha\beta} \quad (3.7c)$$



Şekil 3.2 : Yayalar arası etkileşimde normalize vektör ve teğet vektör ilişkisi



Şekil 3.3 : Duvar ve yaya etkileşiminde normalize vektör ve teğet vektör ilişkisi

$$\vec{F}_{\alpha w}^{dik} = C_1 H(r_\alpha - d_{\alpha w}) n_{\alpha w} - C_2 H(r_\alpha - d_{\alpha w}) \Delta_{\alpha w}^t t_{\alpha w} \quad (3.8a)$$

$$H(r_\alpha - d_{\alpha w}) = \begin{cases} 0 & r_\alpha - d_{\alpha w} < 0 \\ r_{\alpha w} - d_\alpha & r_\alpha - d_{\alpha w} \geq 0 \end{cases} \quad (3.8b)$$

$$\Delta_{\alpha w}^t = v_\alpha(t) t_{\alpha w} \quad (3.8c)$$

Sonuç olarak yayalar üzerindeki itici kuvvet içsel ve dışsal kuvvetlerin toplamı olarak Denklem 2.10 alt denklem grubunda verilmiştir.

$$\vec{F}_{\alpha\beta}^{itici}(t) = \vec{F}_{\alpha\beta}^{iik}(t) + \vec{F}_{\alpha\beta}^{dik}(t) \quad (3.9a)$$

$$\vec{F}_{\alpha w}^{itici}(t) = \vec{F}_{\alpha w}^{iik}(t) + \vec{F}_{\alpha w}^{dik}(t) \quad (3.9b)$$

Çizelge 3.3 : Dışsal İtici Kuvvet Formülasyonu Terminolojisi.

Parametre	Birimi	Açıklaması
$v_{\alpha}(t)$	m/sn	α yayasının t anındaki hızı
$v_{\beta}(t)$	m/sn	α yayasının t anındaki hızı
r_{α}	metre	α yayasının yarıçapı
$\vec{n}_{\alpha\beta}$	vektör	β yayasından α yayası doğrultusunda normalize edilmiş vektör
$\vec{n}_{\alpha w}$	vektör	w duvarından α yayası doğrultusunda normalize edilmiş vektör
$r_{\alpha\beta}$	metre	α ve β yayalarının yarıçapları toplamı
$d_{\alpha\beta}$	metre	α ve β yayalarının kütle merkezleri arasındaki uzaklık
$d_{\alpha w}$	metre	α yayasının w duvarı sınırına olan dik uzaklığı
C_1	skaler	Sabit bir sayı
C_2	skaler	Sabit bir sayı
$\vec{t}_{\alpha\beta}$	Vektör	Normalize vektöre teğet olan vektör

3.2.3 Çekici kuvvet

Yayalar diğer yayalar ile bir arada bulunma ya da pencere, görüntü(yönlendirici işaretler) vb., etmenlere doğru hareket etme isteği oluşturan çekici bir kuvvete maruz kalmaktadır. Bu çekici kuvvet daha önce belirtilen itici kuvvetlere benzer şekilde tanımlanabilir. Çekici kuvvet formülasyon olarak itici kuvvetle aynı formülasyona sahip olup, $B_{\alpha\beta}$ parametresi daha büyük bir değere, $A_{\alpha\beta}$ daha küçük bir değere sahip ve negatif olup zamana bağımlı hale gelir.

3.2.4 Toplam kuvvet

Yayalara etki ettiği düşünülen bu kuvvetlerin aynı anda etki ettiği varsayıldığından bu kuvvetlerin toplamı yaya üzerinde etkili olan toplam kuvveti verecektir. Bu kuvvet Denklem 3.10 şeklinde tanımlanmıştır. Denklem 3.10 içerisinde yer alan $\xi(t)$ ifadesi yayaların davranışlarındaki sistematik olmayan dalgalanmaları temsil etmektedir.

$$\vec{f}_{\alpha}^{toplaml}(t) = \vec{F}_{\alpha}^{Yonlendirici} + \sum_{\beta} \vec{F}_{\alpha\beta}^{itici}(t) + \sum_w \vec{F}_{\alpha w}^{itici}(t) + \sum_{\beta} \vec{F}_{\alpha\beta}^{cekici}(t) + \xi(t) \quad (3.10)$$

3.3 Sosyal Kuvvet Modeline Yönelik Eleştiriler

Helbing'in sosyal kuvvet modelini önermesinin ardından, araştırmacılar model içerisindeki gördükleri eksik kısımları tamamlamak ve yaya hareketlerini daha gerçekçi hale getirebilmek amacıyla, modelin temel yapısını koruyacak şekilde bazı yeni yaya dinamiği modelleri öne sürmüşlerdir.

Heliövaara vd. [18] sosyal kuvvet modeli gibi fiziksel kuvvetlere dayalı modellerin yayaların davranışlarını modellemede yetersizliğinden yola çıkarak, yayaların zıt yönlü akış durumundaki davranışlarını modelleyen sosyal kuvvet modeline dayalı ajan temelli bir çalışma önermişlerdir. Önerilen modelde yayaların karşılıklı akış esnasında bedenlerini döndürmelerinden dolayı, yayalar üst üste örtüşen üç çember şeklinde tanımlanmışlardır.

Han ve Liu [19] tahliye durumundaki yaya hareketlerini ve davranışlarını bir grup davranışı olarak ele almışlar ve yayalar arasındaki bilgi transferi yoluyla oluşan işbirliği mekanizmasına bağlı olarak yayaların hareket yönlerinin oluştuğunu belirterek modele yeni bir yaklaşım getirmişlerdir.

Yang vd. [20] acil durumlardaki yaya davranışlarının daha doğru bir şekilde yansıtılabilmesine yönelik olarak, sosyal kuvvet modeline seyir kuvveti olarak tanımladıkları yeni bir kuvvet eklemiştir. Acil durum esnasında nereye gideceği konusunda bilgisi olmayan kişilerin bu seyir kuvveti yardımıyla rehber bir diğer yayayı takip etmeleri durumu ele alınmıştır.

Kretz [21] yaya hareketleri esnasında meydana gelen salınım (geriye doğru hareket) hareketinin hangi durumlarda oluştuğu ve bu salınım hareketinin ortadan kaldırılmasına yönelik olarak, model içerisindeki parametre seçimlerine odaklanmıştır.

Farina vd. [22] yayaların hareket yörüngelerini daha iyi yansıtabilmek amacıyla yayaların başlarını da modele dahil etmişlerdir.

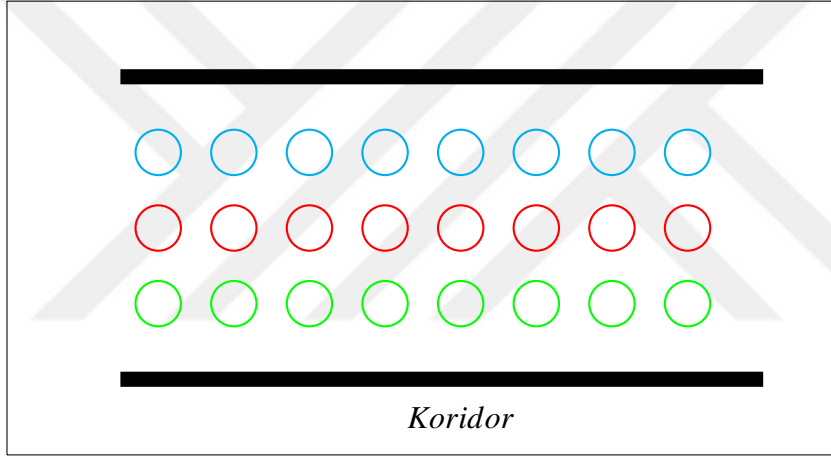
Johansson vd. [23] acil durumlar esnasında bekleyen yayaların yaya trafiği üzerindeki etkilerini araştırmak amacıyla sosyal kuvvet modeli üzerinde bazı genişletmeler yaparak sosyal kuvvet modelinde bekleyen yayaları oluşturmuşlardır.

3.4 Yaya Hareketi Temel Davranışlar (Kalıplar)

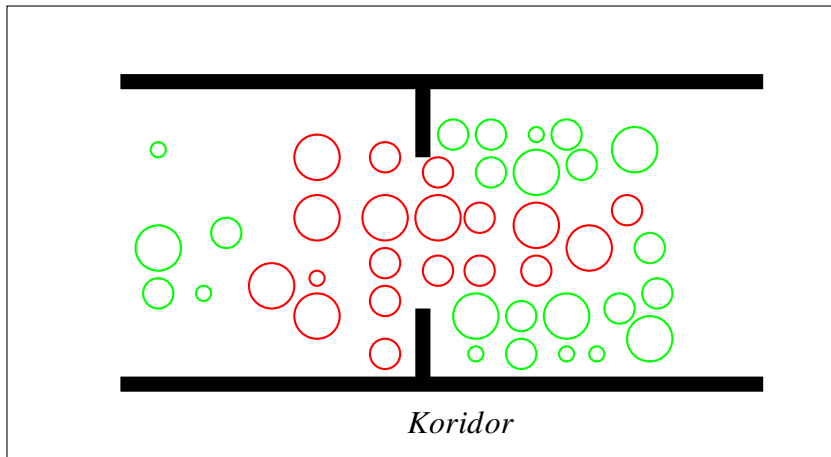
Yayaların hareketleri esnasında kendiliğinde meydana gelen temel davranışlar kurulan yaya hareketi modellerini doğrulamak amacıyla bakılması gereken bir durumdur. Yaya hareketi ile ilgili davranışlar Helbing vd. [16] tarafından normal durumda meydana gelen davranışlar ve panik durumunda meydana gelen davranışlar olmak üzere ikiye ayrılmıştır.

3.4.1 Normal durum

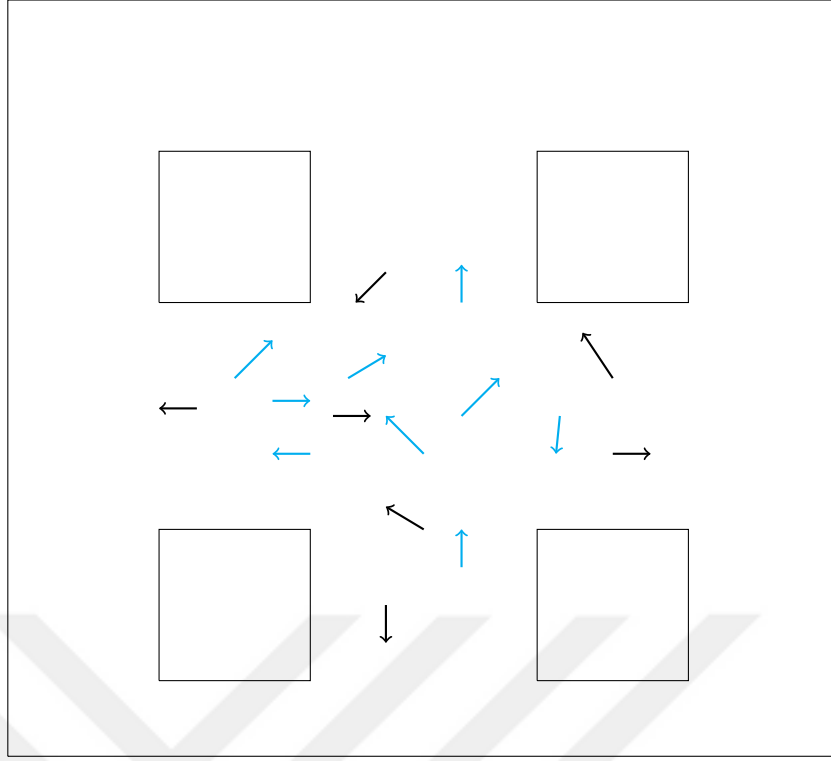
Yayaların hareketlerinin sonucu kendiliğinde meydana gelen temel davranışlar şerit oluşumu, darboğazlarda salınım hareketi ve kesişim noktalarındaki dinamiklerdir [16]. Meydana gelen bu davranışlar sırasıyla Şekil 3.4 Şekil 3.5’de örneklendirilmiştir.



Şekil 3.4 : Şerit oluşumu



Şekil 3.5 : Dar boğazlarda salınım



Şekil 3.6 : Kavşaklarda yaya akışı olgusu

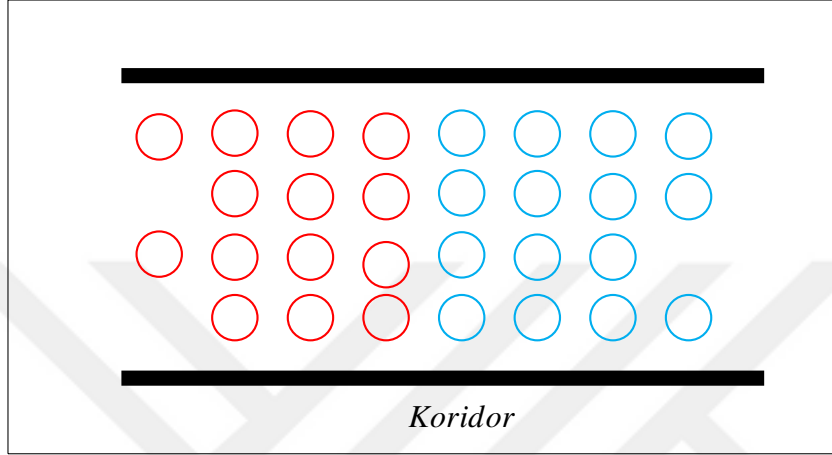
3.4.2 Panik durumu

Helbing vd. [16] göre yayalar panik durumunda aşağıda belirtilen davranışları sergileme eğilimindedir.

- Panik durumundaki yayalar endişeli olmakta ve buna bağlı olarak kararsızlık düzeyleri artmaktadır .
- Paniğin kaynağından biran önce uzaklaşma eğiliminde olmalarından dolayı arzu edilen hızlarında çok önemli artışlar gözlemlenmektedir.
- Kompleks durumlar karşısında ne yapması gerektiğini bilmeyen yayalar, çevresindeki yayaları izlemekte ve onların sergilediği davranışları sergilemektedir. Yani bu davranış bir sürü davranışı olarak ya da diğer yayalar tarafından uygulanan bir çekici kuvvet olarak nitelendirilebilir.

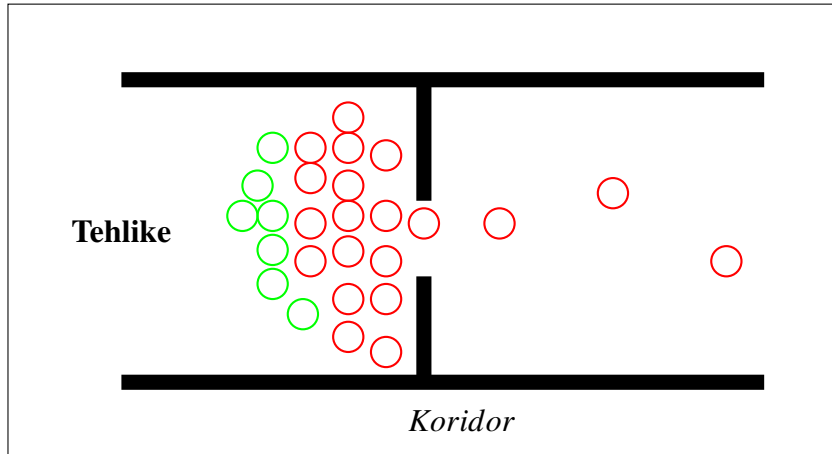
Öte yandan panik durumunda yayaların hareket alanının yoğunluğuna bağlı olarak meydana gelen bazı kolektif olgular ise aşağıdaki gibidir.

- "Freezing-by-heating" etkisi: Yayalar panik anında kendilerini sınırlı ve sabırsız hissetmelerine bağlı olarak hareket alanının yoğunluğuna bağlı olarak da hareket anında yürüme yönlerini sürekli olarak değiştirme ihtiyacı hissetmektedirler. Yayaların hareket yönlendirindeki bu düzensizlik gürültü olarak adlandırılmaktadır. Bu gürültülere bağlı olarak tek bir şerit halinde yayaların birarada hareket ettiği görülmektedir. Bu durum Şekil 3.7'de görselleştirilmiştir.



Şekil 3.7 : "Freezing-by-heating" etkisi

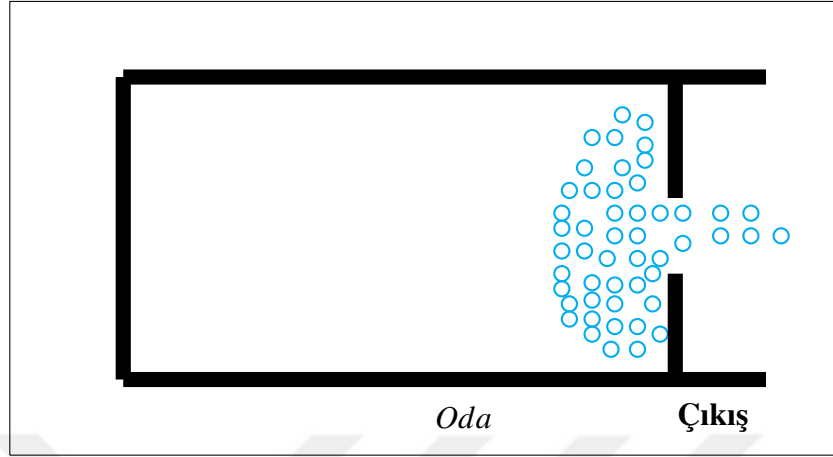
- "Faster-is-slower" etkisi: Bu terim belirli durumların (tahliye durumu) yüksek hızda gerçekleşmesi durumunda daha fazla süre aldığı gözlemine dayanmaktadır. Birbirleriyle rekabet halinde olan birimlerin toplamından oluşan aktiviteleri koordine etmek için beklemek ortalama süreci hızlandırabilir. Bu durum Şekil 3.8'de görselleştirilmiştir.



Şekil 3.8 : "Faster is slower" etkisi

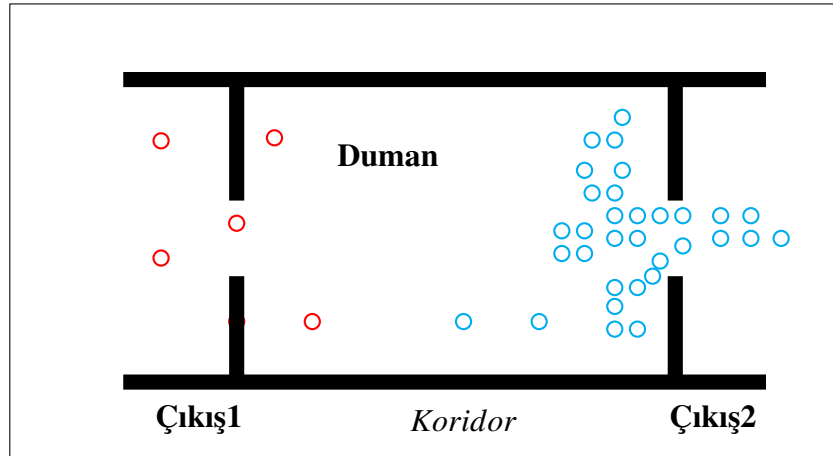
- "Transition to incoordination due to clogging": Yayaların herhangi bir ortamdan dışarıya doğru akışları oldukça koordineli ve normal bir hızda gerçekleşmektedir.

Yayaların yoğunluğunun arttığı durumlarda ise yaya akışı koordinasyonu bozulmakta ve çıkış noktalarında çok büyük yığılmalar oluşmaktadır. Bu durum Şekil 3.9'da görselleştirilmiştir.



Şekil 3.9 : "Transition to incoordination due to clogging"

- "Phantom panic": Herhangi acil bir durum olmayan durumlarda da panik oluşabilir. Bu gibi durumlarda "faster-is-slower" etkisinden kaynaklı olarak yaya akışlarında sıkışmalar meydana gelebilir.
- "Ignorance of available exits": Yayalar yangın gibi zehirli gazların ve dumanların salındığı bir ortamda görüş alanının kapalı olması ve stres düzeyine bağlı olarak çıkışları göremeyebilir. Bu durum Şekil 3.10'da görselleştirilmiştir.



Şekil 3.10 : "Ignorance of available exits"



4. YAYA GRUP DAVRANIŞI

Bu bölümde ilk olarak yayaların sergilediği grup davranışı ile ilgili literatürde yer alan çalışmalar incelenmiş özet bir tablo Çizelge 4.1’de sunulmuştur. Daha sonra grup davranışı kapsamında Boids algoritması incelenmiş ve açıklanmıştır. Grup davranışının en belirgin özelliği yayaların bir arada ve birbirlerini takip eder bir durumda hareket etmeleridir. Ayrıca büyük yaya gruplarında yayaların sergiledikleri davranış doğadaki canlıların (kuşlar vb.) sergilediği sürü davranışına benzemektedir. Bu sebeplerden dolayı grup davranışının modellenmesinde uygun bir yardımcı algoritma olabileceği düşüncesiyle Boids algoritması incelenmiştir.

4.1 Literatür Taraması

Fang vd. [24]:

- Çalışmanın Amacı:Yayaların bir yerden çıkışlarında sergilediği kolektif davranışları yaya hareket modellerine eklemek.
- Davranış Tipi: Lideri takip etme
- Modelleme Yaklaşımı: Ajan Temelli Modelleme
- Hareket Modeli: Skaler Alan Yöntemi
- Elde Edilen Sonuçlar: Oluşturulan model yapılan deneysel çalışmalar yardımıyla kalibre edilerek, yaya çıkış davranışında sosyal kolektif davranışların dahil edildiği yeni bir model oluşturulmuştur.

Sivers vd. [25]

- Çalışmanın Amacı: Sosyal kimlik teorisi ve öz kategorizasyon teorilerinin yaya davranışları üzerindeki etkisini ortaya koymak. Ve bu teorilere dayalı olarak ortaya çıkan diğer yayalara yardım etme davranışının etkilerini incelemek.

- Davranış Tipi: Yardım Etme.
- Modelleme Yaklaşımı: Hücresel Otomat
- Hareket Modeli: Optimal Adımlar Modeli
- Faydalanılan Teoriler: Sosyal kimlik teorisi ve Öz kategorizasyon
- Elde Edilen Sonuçlar: Çalışmanın güvenlik planlayıcılarına yaralı insanların olduğu durumlar için daha doğru tahliye süresi tahminlemelerine yardımcı olacağı düşünülmektedir.

Song vd. [26] :

- Çalışmanın Amacı: Bencillik/özgecilik davranışlarının tahliye süresi üzerine etkisini araştırmak.
- Davranış Tipi: Bencillik/Özgecilik davranışları ve sempati/rekabet duygusu
- Modelleme Yaklaşımı: Hücresel otomat ve Kafes Gaz Modeli
- Faydalanılan Teoriler: Oyun Teorisi
- Elde Edilen Sonuçlar: Bencilliğin tahliye süresini uzattığı, sempati duygusunun ise işbirliğini artırarak daha kısa tahliye sürelerinin elde edilmesini sağladığı görülmüştür. Aynı zamanda rekabet duygusundaki artışın da tahliye süresini uzattığı görülmüştür.
- Geometrik Özellik: Grid tabanlı

Lemercier ve Auberlet [27]:

- Çalışmanın Amacı: Yayaların sahip oldukları algıya bağlı olarak çevreleriyle kurduğu etkileşim sonucu, mevcut durum doğrultusunda bir davranış seçmelerini sağlamak.
- Davranış Tipi: Takip etme ve Grup olma/olmama
- Modelleme Yaklaşımı: Ajan Temelli Modelleme
- Hareket Modeli: RVO2 çarpışmadan kaçınma modeli (hız tabanlı)

- Elde Edilen Sonuçlar: Farklı davranışların birarada kullanılması sonucunda daha gerçekçi simülasyon sonuçlarını elde edildiği ve kolektif davranışların ortaya çıktığı görülmüştür.

Zhou vd. [28] :

- Çalışmanın Amacı: İki farklı yaya grubu oluşturularak bu yayaların çarpışmalarını önleyecek şekilde bir arada hareket etmelerini sağlamak.
- Davranış Tipi: Grup davranışı, kaçınma davranışı
- Modelleme Yaklaşımı: Ajan Temelli Modelleme
- Hareket Modeli: Hız Engel Yöntemi
- Elde Edilen Sonuçlar: Elde edilen deney sonuçlarına ve literatürde yer alan modellerle yapılan karşılaştırmalar sonucunda model doğrulanmış ve modelin daha fazla yaya grubu oluşturularak genişletilebileceği görülmüştür.

You vd. [29]:

- Çalışmanın Amacı: Küçük yaya gruplarının (1-3 yaya) tahliye etkinliği üzerindeki etkilerini araştırmak.
- Davranış Tipi: Grup davranışı
- Modelleme Yaklaşımı: Hücresel Otomat
- Faydalanılan Teoriler: Von Neumann modeli
- Geometrik Özellik: Grid tabanlı
- Elde Edilen Sonuçlar: Grup büyüklüğündeki artışın tahliye etkinliği azalttığı görülmüştür.

Qiu ve Hu [30] :

- Çalışmanın Amacı: Grup içi yapıların, grup büyüklüğünün ve gruplar arası ilişkilerin kalabalık davranışlar üzerine etkilerini ortaya çıkarmak.

- Davranış Tipi: Grup davranışı
- Modelleme Yaklaşımı: Ajan Temelli
- Hareket Modeli: Reynolds Boids Algoritması
- Elde Edilen Sonuçlar: Farklı yaya grup yapılarının oluşturulması modellenmiş ve elde edilen deney sonuçlarına göre, farklı grup büyüklüklerinin, grup içi yapıların ve grup içi ilişkilerin yaya davranışları üzerinde önemli etkilere sahip olduğu görülmüştür.

Li vd. [31] :

- Çalışmanın Amacı: Yayalar arasındaki sosyal ilişkiye dayalı olarak oluşacak yaya gruplarının ve grup içerisinde yer alan lideri takip etme davranışının tahliye üzerine etkisini araştırmak.
- Davranış Tipi: Grup davranışı ve lideri takip etme
- Modelleme Yaklaşımı: Ajan Temelli modelleme
- Hareket Modeli: Sosyal Kuvvet Modeli
- Elde Edilen Sonuçlar: Grup halinde hareket eden yayaların çıkışlarda meydana gelen rekabeti azaltmasına bağlı olarak tahliye süresinin mevcut sosyal kuvvet modeli ile karşılaştırıldığında azaldığı görülmüştür.

Lovreglio vd. [32] :

- Çalışmanın Amacı: Tahliye esnasında yayaların sergilediği sürü davranışının yayaların çıkış seçimi üzerine etkisini araştırmak.
- Davranış Tipi: Sürü Davranışı
- Modelleme Yaklaşımı: Ajan Temelli modelleme
- Faydalanılan Teoriler: Sosyal Etki teorisi
- Elde Edilen Sonuçlar: En kalabalık ve en az kalabalık çıkışa yakın olan yaya sayısı arasındaki fark arttıkça veya en az kalabalık çıkışa yakın insan sayısı arttıkça sürü davranışının meydana gelme ihtimalinin azaldığı görülmüştür.

Ma vd. [33] :

- Çalışmanın Amacı: Tahliye esnasında kalabalık içerisinde yer alan lider sayısının ve bu liderlerin kalabalık içerisindeki pozisyonunun tahliye üzerindeki etkisini araştırmışlardır.
- Davranış Tipi: Lideri takip etme
- Modelleme Yaklaşımı: Ajan Temelli modelleme
- Hareket Modeli: Sosyal kuvvet modeli
- Elde Edilen Sonuçlar: Elde edilen deney sonuçlarına göre yaya sayısının fazla olduğu durumlarda az sayıda lider, yaya sayısının az olduğu durumlarda ise daha fazla sayıda liderin olmasının tahliye etkinliğini artırdığı görülmüştür. Ayrıca kalabalık tahliyesinde lideri takip etmenin her zaman pozitif etkiye sahip olmadığı durumlar da gözlemlenmiştir.

Dossetti, Bouzat ve Kuperman [34] :

- Çalışmanın Amacı: Farklı davranışsal profiller arasındaki etkileşimin tahliye süresi üzerindeki etkisini araştırmak.
- Davranış Tipi: İşbirlikçilik ve bencillik
- Modelleme Yaklaşımı: Ajan Temelli modelleme
- Faydalanılan Teoriler: Oyun teorisi
- Elde Edilen Sonuçlar: Tamamıyla bencilce ya da işbirlikçi davranışlar yerine her iki davranışından orta seviyede yer aldığı davranışların daha etkili olduğu görülmüştür.

Zheng ve Cheng [35] :

- Çalışmanın Amacı: Yayaların sahip olduğu rekabetçi ve işbirlikçi davranışların tahliye süresine etkisini araştırmak.
- Davranış Tipi: Rekabetçi ve işbirlikçi davranış
- Modelleme Yaklaşımı: Hücreli otomat

- Faydalanılan Teoriler: Oyun teorisi
- Elde Edilen Sonuçlar: İşbirlikçi davranışların tahliye süresini kısalttığı rekabetçi davranışın ise tahliye süresini artırdığı gözlemlenmiştir.

Qiu vd. [36] :

- Çalışmanın Amacı: Dinamik olarak oluşacak yaya gruplarının kalabalık davranışları üzerindeki etkilerini araştırmak.
- Davranış Tipi: Grup davranışı
- Modelleme Yaklaşımı: Ajan Temelli modelleme
- Faydalanılan Teoriler: Fayda teorisi ve Sosyal karşılaştırma teorisi
- Elde Edilen Sonuçlar: Deney sonuçları oluşturulan modelin dinamik grup davranışlarını ortaya koymada başarılı olduğunu ve sosyal faktörlerin yaya davranışları üzerinde büyük bir etkiye sahip olduğunu göstermiştir.

Ren vd. [37]:

- Çalışmanın Amacı: Farklı yaya grubu tiplerini(sosyal gruplar, yürüyüş grupları, lideri takip eden gruplar vb.,) modellemek ve simüle etmek.
- Davranış Tipi: Grup davranışı
- Modelleme Yaklaşımı: Ajan Temelli Modelleme
- Hareket Modeli: Hız engel yaklaşımı
- Elde Edilen Sonuçlar: Önerilen model farklı grup tiplerini modelleyebildiği ve yaya davranışları sırasında zaman zaman meydana gelen grupların ayrılıp tekrar birleşmesi ya da yayaların grup değiştirebilmesi gibi davranışları da ortaya koyabildiği görülmüştür.

Çizelge 4.1 : Literatür özeti.

Çalışmalar	Grup Davranışı Türü	Faydalanılan Teoriler	Modelleme Yaklaşımı	Hareket Modeli
Fang vd. [24]	Lideri Takip Etme	-	ATM	Skaler Alan Yöntemi
Sivers vd. [25]	Yardım Etme	SKT ve ÖKT	HO	Optimal Adımlar Modeli
Song vd. [26]	Bencillik/Özgeçiclik	OT	HO ve KGM	-
Lemercier ve Auberlet [27]	Takip Etme	-	ATM	RVO2
Zhou vd. [28]	GD	-	ATM	Hız Engel Yöntemi
You vd. [29]	GD ve Lideri Takip Etme	VNM	HO	-
Qiu ve Hu [30]	Sürü Davranışı	-	ATM	Reynold Boids Algoritması
Li vd. [31]	Lideri Takip Etme	-	ATM	Sosyal Kuvvet Modeli
Lovreglio vd. [32]	İşbirlikçilik ve Bencillik	SET	ATM	-
Ma vd. [33]	Rekabetçi ve İşbirlikçi	-	ATM	Sosyal Kuvvet Modeli
Dossetti, Bouzat ve Kuperman [34]	GD	OT	ATM	-
Zheng ve Cheng [35]	GD	OT	HO	-
Qiu vd. [36]	GD	FT ve SKTT	ATM	-
Ren vd. [37]	GD	-	ATM	Hız Engel Yaklaşımı

Not: SKT: Sosyal Kimlik Teorisi, ÖKT:Öz Kategorizasyon Teorisi, SET: Sosyal Etki Teorisi, OT: Oyun Teorisi, FT: Fayda Teorisi, SKTT: Sosyal Karşılaştırma Teorisi, HO: Hücrel Otomat,ATM: Ajan Temelli Modelleme, KGM: Kafes Gaz Modeli

4.2 Boids Algoritması

Doğadaki canlıların sürü halinde gerçekleştirdikleri toplu hareketlerden esinlenme sonucu Craig Reynold tarafından ilk kez 1986 yılında ortaya atılmıştır [4]. Algoritma temelinde kuşların, balıkların ve diğer hayvanların birarada (sürü) hareket etmelerini inceleyen simülasyon modellerinde kullanılmıştır. Bu algoritma canlıların gerçekleştirdiği sürü hareketinden yola çıkarak grup davranışlarını modellemeye, ya da grup davranışlarından biri olan lideri takip etme davranışını modellemek amacıyla kullanılmaktadır. Sürü içerisinde yer alan her bir canlı kendi çevresiyle oluşturduğu algı sonucu bağımsız olarak hareket etmekte iken, bu sürü içerisindeki her bir canlının oluşturduğu davranışlar senkronize bir sürü(grup) davranışına dönüşmektedir. Algoritma oyun programlama, parçacık simülasyonları, robot kontrolü gibi birçok alanda yaygın olarak kullanılmaktadır. Sürü halinde hareket eden kuşların oluşturduğu grup davranışı Şekil 4.1'deki gibi gösterilmiştir [38].

4.3 Algoritmanın Yapısı ve Çalışma Şekli

Algoritma parçacıkların(canlıların) sürü hareketlerini modellemeye 3 temel yönlendirici hareketi(kuralı) kullanmaktadır. Bunlar sırasıyla ayırma(separation), birleştirme(cohesion) ve hizalamadır(alignment). Sürü içerisinde yer alan her bir parçacık bu üç kural çerçevesinde hareketlerini gerçekleştirmektedir. Bu kuralların her biri yayanın o anki hızının ve yönünün değişmesine sebep olan birer yönlendirici kuvvet olarak düşünülebilir. Bu kuvvetlerin her biri detaylı bir şekilde açıklanmadan önce algoritmanın çalışma yapısı Algoritma 1 ile gösterilmiştir.

4.3.1 Ayırma

Bu kural canlıların(parçacıkların) birbirlerine çok fazla yaklaşmamalarını sağlamaya yöneliktir. Canlıların birbirleriyle çarpışmalarını önler.



Şekil 4.1 : Kuş sürü davranışı.

Algoritma 1: Boids algoritması sözde kodu

Sonuç değişkenler: Güncel pozisyon vektörleri

Girdi Değişkenler: Sürü üyelerinin başlangıç pozisyon vektörleri

Çıktı değişkenler : Sürü üyelerinin yeni pozisyonları

```

1  $v1 \leftarrow 0$  ;
2  $v2 \leftarrow 0$  ;
3  $v3 \leftarrow 0$  ;

4 foreach Sürü Üyesi  $b$  do
5    $v1 \leftarrow \text{Ayırma}(b)$  ;
6    $v2 \leftarrow \text{Birleştirme}(b)$  ;
7    $v3 \leftarrow \text{Hizalama}(b)$  ;
8    $b.hiz \leftarrow b.hiz + v1 + v2 + v3$  ;
9    $b.pozisyon \leftarrow b.pozisyon + b.hiz$  ;
10 end
```

Algoritma 2: Boid algoritması Ayırma Fonksiyonu

```

1 Fonksiyon  $\text{Ayırma}(b_j)$ 
2 Vektör  $c=0$ 
3 foreach Boid  $b$  do
4   if  $b \neq b_j$  then
5     if  $b.pozisyon - b_j.pozisyon < \text{dayırma}$  then
6        $c \leftarrow c - (b.pozisyon - b_j.pozisyon)$ 
7     end
8   end
9 end
10 return  $c$ 
11 end
```

4.3.2 Birleřtirme

Canlıların sürü halinde hareket etmesini saęlayan en önemli kuraldır. Canlıların birarada hareket etmesini (birarada kalmalarını) bu kural saęlar. Kuralın işleyişinde her canlı sürü içerisinde yer alan tüm canlıların merkezine doęru hareket edecek şekilde davranırlar.

Algoritma 3: Boid algoritması Birleřtirme Fonksiyonu

```
1 Fonksiyon Birleřtirme( $b_j$ )  
    /*  $pc_j$ : Boid  $j$  için algılanan kutle merkezi */  
2  $pc_j \leftarrow (b_1.pozisyon + b_2.pozisyon + \dots + b_{j-1}.pozisyon + \dots +$   
    $b_{j+1}.pozisyon + \dots + b_N.pozisyon) / N - 1$   
    /*  $N$ : toplam Boid sayısı */  
    /*  $A$ : her bir adımdaki yaklařma yüzdesini belirler.  
    */  
3 foreach Boid  $b$  do  
4     if  $b \neq b_j$  then  
5         |  $pc_j \leftarrow pc_j + b.pozisyon$   
6     end  
7 end  
8  $pc_j \leftarrow pc_j / N - 1$   
9 return  $(pc_j - b_j.pozisyon) / A$   
10 end
```

4.3.3 Hizalama

Canlıların (örneğin kuřların) dięer canlıları takip etmelerini saęlayan bu kuraldır. Bu kural sayesinde canlılar dięer canlılar ile belirli bir mesafede kalarak hızlarını korurlar ve dięer canlıları takip ederler. Sürü içerisinde yer alan her canlı aynı yönü işaret etmektedir.

Algoritma 4: Boid algoritması Hizalama Fonksiyonu

```
1 Fonksiyon hizalama( $b_j$ )  
    /*  $pv_j$ : Boid  $j$  için algılanan hız */  
2  $pv_j \leftarrow (b_1.hiz + b_2.hiz + \dots + b_{j-1}.hiz + \dots + b_{j+1}.hiz + \dots + b_N.hiz) / N - 1$   
    /*  $N$ : toplam Boid sayısı */  
3 foreach Boid  $b$  do  
4     if  $b \neq b_j$  then  
5         |  $pv_j \leftarrow pv_j + b.hiz$   
6     end  
7 end  
8 return  $pv_j \leftarrow pv_j / N - 1$   
9 end
```

5. YAYA DAVRANIŞI İÇİN ÖNERİLEN MODEL

Bu bölümde ilk olarak önerilecek model içerisinde kullanılacak olan bazı temel bilgiler ve teorilere değinilecektir. Daha sonra yayaların hareketlerini modellemede kullanılan Sosyal Kuvvet modelinin implementasyon süreci, karşılaşılan durumlar için önerilen çözüm önerilerinden bahsettikten sonra, yayaların acil durumda sergiledikleri bir davranış olan grup davranışı ile ilgili sosyal kuvvet modeline dayalı olarak önerilen modelleme yaklaşımı sunulacaktır.

5.1 Model Öncelikleri

Önerilecek yaya davranış modeli Sosyal Kuvvet Modeline dayanmaktadır. SKM'nin fizik kuramlarına dayanması ve içerdiği yapı itibari ile adi diferansiyel denklem formunda olması nedeniyle modelin tanıtımından önce diferansiyel denklemler konusuna bu bölümün altında değinilme ihtiyacı duyulmuştur. Öte yandan önerilen modelde yaya gruplarının oluşumunda Sosyal Uzaklık teorisinden faydalandığı için sosyal uzaklık teorisine de kısaca bu bölüm altında değinilecektir. Ayrıca insan gözü görüş alanı konusuna da kısaca bu bölüm altında değinilecektir..

5.1.1 Diferansiyel denklemler

Karşılaşılan mühendislik problemlerinin büyük bir kısmında kullanılan fiziksel kurallar ve matematiksel ilişkiler diferansiyel denklemler formundadır. Sosyal kuvvet modelinin dayandığı Newtonun hareket kanunlarının da diferansiyel denklemler formunda olduğu görülmektedir. Bu yüzden çalışma içerisinde diferansiyel denklemler konusunda temel bilgileri vermenin faydalı olacağı düşünülmüştür.

Tanım 5.1 Diferansiyel Denklem: Bilinmeyen bir fonksiyonun türevini içerisinde barındıran denklemler diferansiyel denklem olarak adlandırılır.

Tanım 5.2 Adi Diferansiyel Denklem: Bilinmeyen fonksiyonun değerinin tek bir değişkene dayandığı diferansiyel denklemlerdir. N.dereceden bir diferansiyel

denklemin temel formu Denklem 5.1’de gösterildiği gibidir.

$$F(x, y, y', \dots, y^{(n)}) = 0 \quad (5.1)$$

Tanım 5.3 Kısmi Diferansiyel Denklem: Bilinmeyen fonksiyonun değerinin birden fazla değişkene dayandığı diferansiyel denklemlerdir. N bağımsız değişkene sahip, birinci dereceden kısmi bir diferansiyelin temel formu Denklem 5.2’de gösterildiği gibidir.

$$F(x_1, x_2, \dots, x_n, v, \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2}, \dots, \frac{\partial v}{\partial x_n}) = 0 \quad (5.2)$$

5.1.1.1 Diferansiyel denklemlerin çözümünde nümerik yaklaşımlar

Diferansiyel denklemlerin analitik çözüm yöntemleri ile çözülemediği ya da çözümünün zor olduğu durumlarda nümerik yöntemlere başvurulmaktadır. Aslında nümerik yöntemler günümüzde kullanılan birçok yazılım paketi içerisinde gömülü olarak yaygın bir şekilde kullanılmaktadır. Bu çalışma içerisinde yer alan denklem adi bir diferansiyel denklem olması nedeniyle bu bölümde yalnızca adi diferansiyel denklemlerin çözümünde kullanılan nümerik yöntemlere değinilecektir. Bu nümerik yöntemler hesaplamaların dayandırıldığı adıma göre temelde iki gruba ayrılmaktadır. Eğer mevcut adımdaki hesaplama yalnızca bir önceki adıma dayanıyorsa tek adımlı, bir önceki adımdan daha fazla önceki adıma dayanıyorsa çok adımlı yöntemler olarak adlandırılır. Euler yöntemi ve Runge-Kutta yöntemi tek adımlı nümerik yöntemler iken, Adams-Bashforth ve Adams-Moulton yöntemleri çok adımlı yöntemlere örnektir. Bu çalışma içerisinde Euler yöntemi hesaplama zamanı açısından uygun olması nedeniyle tercih edilmiştir. Bu yüzden yalnızca Euler yöntemine kısaca değinilecektir.

5.1.1.2 Euler(Euler-Cauchy) yöntemi

Euler yöntemi temelde Denklem 5.3a ile gösterilen Taylor serisine dayanmaktadır. Denklem 5.3 grubunda yer alan denklemler içerisinde geçen h değeri adım büyüklüğü olmak üzere, adım büyüklüğünün çok küçük değerleri için Denklem 5.3a de gösterilen Taylor serisinin Denklem 5.3b’de gösterilen ifadeye yakınsaması ve bu yakınsayan ifadenin Denklem 5.3c de verilen eşitliğe dönüştüğü kabul edilir. Euler yönteminin sunduğu yaklaşım Denklem 5.3d ile ifade edilmektedir.

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \dots \quad (5.3a)$$

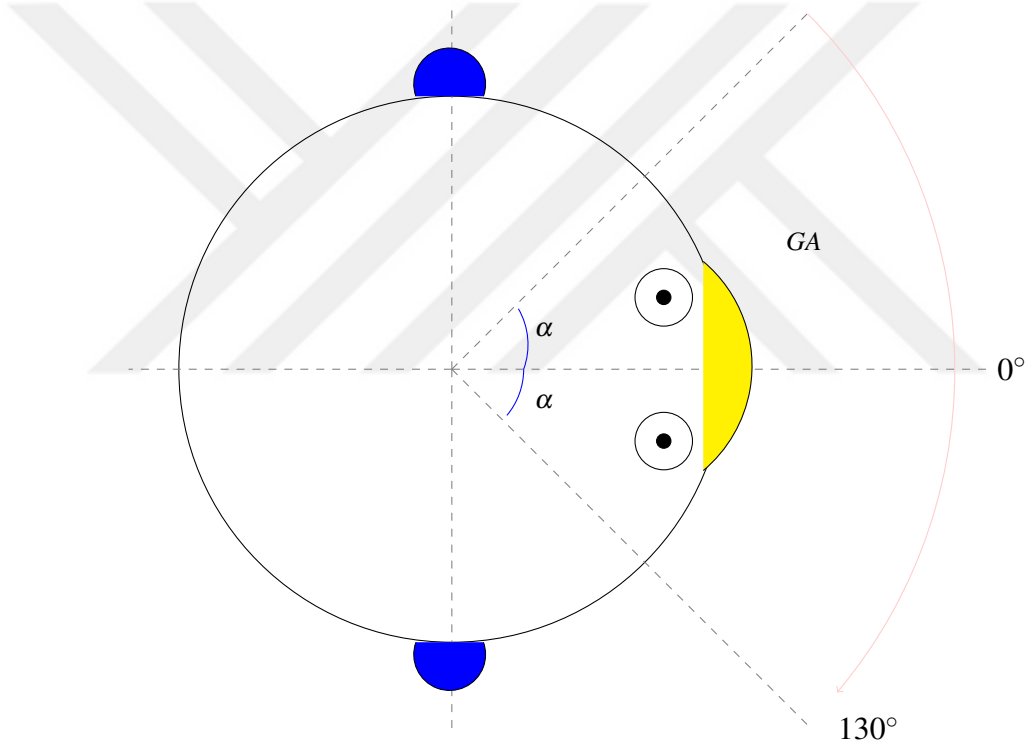
$$y(x+h) \approx y(x) + hy'(x) \quad (5.3b)$$

$$y(x+h) = y(x) + hf(x,y) \quad (5.3c)$$

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (n = 0, 1, \dots) \quad (5.3d)$$

5.1.2 İnsan gözü görüş alanı

Sağlıklı bir göze sahip insanın geniş görüş alanı 180° olup, iki gözü tarafından aynı anda görülebilen alan 130° iken, gözün yüksek çözünürlüğe sahip olarak odaklanıp görebildiği alan 2° açıya sahiptir [39]. 130° derece dışında kalan alanlar(kenardan görme) her iki göz tarafından algılanamayan alanlar olmakla beraber görüş alanı içinde kabul edilir.



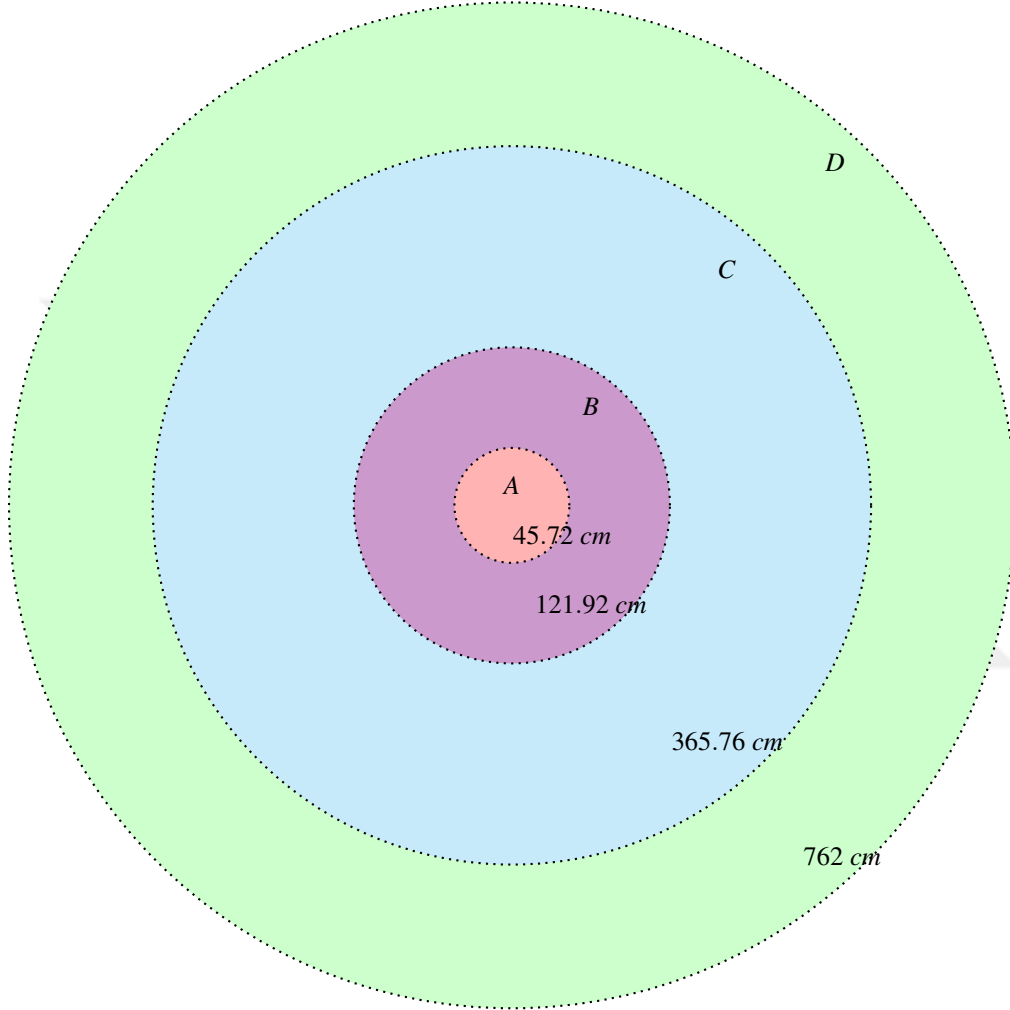
Şekil 5.1 : İnsan gözü görüş alanı(field of view).

5.1.3 Sosyal uzaklık teorisi

Hall [40] insanların arasındaki ilişkinin seviyesine bağlı olarak aralarında fiziksel mesafeyi ölçmeye yönelik çalışmasında 4 farklı mesafe tipi tanımlamıştır. Bu mesafe ölçüleri Şekil 5.2’de görülmektedir.

- Samimi mesafe (A): Çok yakın ilişkiye sahip olunan kişilerle (örneğin:sevgili) oluşturulan mesafe.

- Kişisel mesafe (B): Yakın arkadaşlarla ya da aile üyeleriyle iletişimde iken oluşturulan mesafe.
- Sosyal mesafe (C): Tanıdığımız kişilerle iletişim halinde iken oluşturulan mesafe.
- Halka açık mesafe (D): Bir topluluğa hitap ederken oluşan mesafe.



Şekil 5.2 : Proksemi(Proxemics).

5.2 YGDTS Modeli

Önerilen model Helbing'in temel SKM modeline dayanmakta olup, yayaların daha gerçekçi davranışlar sergilemesini sağlamak amacıyla Helbing'in modeli üzerinde önerilen grup davranışı doğrultusunda bazı değişiklikler ve eklemeler yapılmıştır. Modelleme yaklaşımı olarak ajan temelli modelleme yaklaşımı kullanılmış ve yayalar belirli kuvvetlerin etkisinde hareket eden ajanlar olarak modellenmiştir. Yayaların hareketleri, üzerlerine etki eden kuvvetlere bağlı olarak oluşturulan adi

diferansiyel denklemin Bölüm 5.1.1.2 de anlatılan Euler yöntemi ile çözülmesiyle elde edilmiştir. Oluşturulan model iki boyutlu düzlemde sürekli iken, adi diferansiyel denklemin çözümünde kullanılan nümerik yaklaşımdan dolayı zaman kesiklidir. Ve belirlenen adım büyüklüğüne bağlı olarak yayalar etki eden kuvvetlerin etkisi altında ivmelenmekte ve bu ivmelenmeye bağlı olarak hızlarını ve konumlarını değiştirmektedir.

Tanım 5.4 Yaya: Yayalar belirli bir kütlesi, başlangıç konumu ve hızı, istenen hızı, ulaşılabilir maksimum hızı, başlangıç yönü, ulaşmak istediği bir hedefi, belirli bir görüş alanı olan ajanlar olarak tanımlanmıştır.

Tanım 5.5 Engel: Yayaların hareket alanı içerisinde yer alan duvarlar ya da farklı şekillere sahip olup yayanın hareket alanını kapatan nesneler model içerisinde engel olarak tanımlanmıştır. Bu nesneler model içerisinde çizgilerle tasvir edilmiştir.

Tanım 5.6 ρ : Yaya yoğunluğunu temsil etmektedir. Birim metrekare alan başına düşen yaya sayısını göstermektedir. Birimi yaya sayısı/ m^2 'dir.

Tanım 5.7 Tahliye Alanı: İki boyutlu düzlemde tanımlanmıştır. Tanımlanan düzlemsel alan sürekli dir.

Tanım 5.8 Lider: Diğer yayaları yönlendiren ve kendisine doğru çeken yayaların sahip olduğu özelliktir.

Tanım 5.19 Grup: Aralarındaki yakınlık ilişkisine dayalı olarak bir arada hareket etme isteğine sahip olan yaya topluluğudur.

Tanım 5.10 Grup Büyüklüğü: Bir grup içerisinde yer alan yaya sayısıdır.

5.2.1 Yayaların gruplandırılması

Tanım 5.11 Yaya Kümesi: Tahliye edilecek olan tüm yayaları içeren ve n elemandan oluşan yaya kümesi $E=\{y_1, y_2, y_3, \dots, y_n\}$ şeklinde tanımlanmıştır.

Tanım 5.12 Grup Kümesi: Aynı grup içerisinde yer alan yayaları içeren ve m elemandan oluşan grup kümesi $G=\{g_1, g_2, g_3, \dots, g_m\}$ şeklinde tanımlanmıştır.

Tanım 5.13 Yaya Konumu: Yayaların konumları 2 boyutlu uzayda gösterilmiş olup $l = \{x, y\}$ şeklinde tanımlanmıştır.

$$mesafe(l_{y1}(x_1, y_1), l_{y2}(x_2, y_2)) = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \quad (5.4)$$

Yayaların gruplandırılmasında iki farklı yöntem kullanılmıştır.

- $\rho \leq 0.125$ olduğu durumda yaya grup büyüklüğü [1,5] arasında değiştiği küçük yaya grupları için Sosyal Uzaklık teorisinde belirtilen kişisel mesafeye bağlı olarak yayaların aynı grup içerisinde yer almaları sağlanmıştır.
- $\rho > 0.125$ olduğu durumlarda Sosyal Uzaklık teorisinde belirlenen mesafe ölçülerini yayaların koruması mümkün olamayacağı için, K-means kümeleme algoritmasından faydalanarak yayalar gruplandırılmıştır. Grup büyüklükleri birbirinden farklılıklar gösterebilmektedir.

5.2.2 Model içerisinde yer alan kuvvetler

Bu bölümde model içerisinde yer alan kuvvetler tanıtılacak ve kuvvetlerin model içerisindeki implementasyon süreci detaylandırılacaktır. Bu kuvvetlerin ilk üç tanesi Helbing'in SKM' sinde yer alan kuvvetler iken diğer üç kuvvet SKM'ye eklenen yeni kuvvetlerdir.

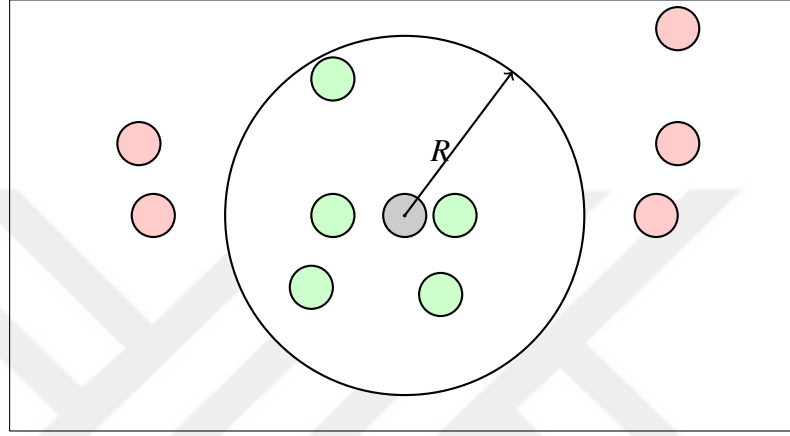
5.2.2.1 Yaya yönlendirici kuvveti

Bu kuvvetin büyüklüğü üzerinde etkili olan en önemli etmen yayaların sahip olmak istedikleri ve sahip olabilecekleri maksimum hızdır. Helbing önerdiği formülasyonda yayaların sahip olabilecekleri istenen hızı zamana bağlı bir denklem haline getirmiş ve bu denklemde yayaların sabırsızlık seviyelerindeki artışa bağlı olarak sahip olmak istedikleri hızın artacağını ileri sürmüştür. Ve sabır düzeyinin yayaların bekleme sürelerindeki artışa bağlı olarak artacağı varsayımında bulunmuştur. Önerilen model içerisinde bu formülasyonun yanı sıra yayaların ulaşabileceği maksimum ulaşılabilir hız Denklem 5.5 ile yaya yoğunluğuna bağlı hale getirilmiştir. Denklem içerisinde yer alan S bir sabittir.

$$v_{\alpha}^{max} = v_{\alpha}^{max} \exp^{(1-\rho)/S} \quad (5.5)$$

5.2.2.2 Yayalar arasındaki itici kuvvetler

Yayalar arasında meydana gelen itici kuvvetlerin hesaplanmasında Denklem 3.5 'den yararlanılmıştır. Bir yaya üzerine diğer yayalar tarafından uygulanan itici kuvvetler hesaplanırken her bir yaya için Şekil 5.3' deki gibi etki alanı tanımlanmıştır. Ve yaya üzerine yalnızca bu etki alanı içerisinde kalan diğer yayalar tarafından etki eden itici kuvvetler hesaplanmıştır.



Şekil 5.3 : Yaya etki alanı.

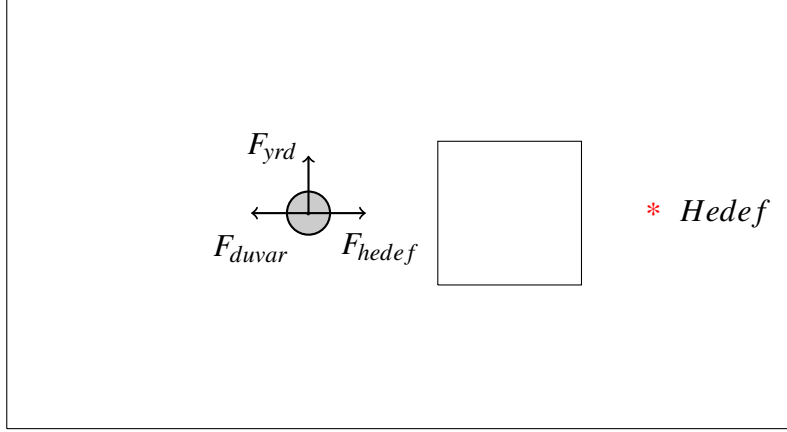
5.2.2.3 Duvarlar ve engeller tarafından uygulanan itici kuvvetler

Duvarların ve engellerin uyguladığı kuvvet Denklem 3.6 ile hesaplanmıştır. Yaya hareket içerisinde yer alan duvarlar içerisinde en yakın duvarın yaya üzerinde itici bir etkiye sahip olduğu yaklaşımla yalnızca bu duvar tarafından uygulanan kuvvet dikkate alınmıştır.

5.2.2.4 Yardımcı yönlendirici kuvvet

Modelin implementasyonu esnasında karşılaşılan bir problemi çözmek amacıyla SKM içerisine yardımcı bir kuvvet eklenmesi ihtiyacı oluşmuştur. Şekil 5.4 de gösterildiği gibi yayanın hedef noktası ile aynı doğrultuda bir engel olması durumunda yaya hedef noktasına ulaşamamaktadır. Bu problemi çözmek amacıyla yayanın engelin etrafını dolaşarak hedefe ulaşmasını sağlayacak yardımcı bir kuvvet eklenmiştir. Bu kuvvet duvar ya da engel tarafından uygulanan kuvvetin büyüklüğüne hedefin konumuna ve yayanın konumuna bağlı olarak duvar kuvvetinin Denklem 5.6'da belirtilen A matrisi ile çarpılmasıyla elde edilmektedir. Duvar kuvvetinin, A matrisi ile çarpılmasıyla duvar kuvveti çeşitli yönlerde döndürülerek yardımcı yönlendirici

kuvveti hesaplanmaktadır. EK A.2’de verilen Java kodları içerisinde farklı durumlar için A matrisinin alacağı değerin nasıl belirlendiğini görülebilir.

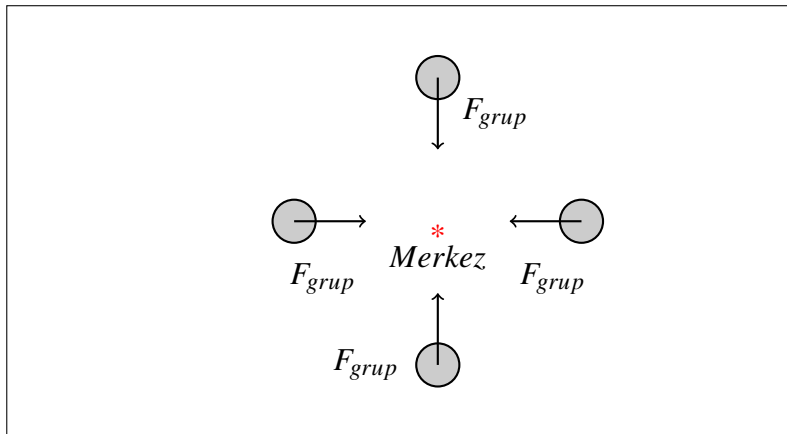


Şekil 5.4 : Yardımcı yönlendirici kuvvet.

$$\vec{F}_{\alpha W}^{yrd}(t) = A_{\alpha W} \exp\left[\frac{r_{\alpha} - d_{\alpha W}}{B_{\alpha W}}\right] \vec{n}_{\alpha W} \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \quad (5.6)$$

5.2.2.5 Grup kuvveti

Grup davranışında en belirgin özellik yayaların birarada hareket etme istekleridir. Aynı grup içerisinde yer alan yayaların birarada hareket etmelerini sağlamak amacıyla Bölüm 4.3.2 ve 4.3.3 de değinilen Boids algoritmasının Birleştirme ve Hizalama fonksiyonlarından yararlanılmıştır. Aynı grup içerisinde yer alan yayaların birarada durmalarını sağlamak amacıyla $\vec{F}_{\alpha grup}$ kuvveti SKM’ye eklenmiştir. Şekil 5.5’te 4 yayanın yer aldığı bir yaya grubuna etki eden grup kuvveti temsil edilmiştir. Bu kuvvetin hesaplanmasında her bir yaya grubu için grubun merkezi (m_x, m_y) Denklem 5.7a ve Denklem 5.7b ile bulunmuştur.



Şekil 5.5 : Grup Kuvveti.

$$m_x = \sum_{g \in G} l_g(x) / s_G \quad (5.7a)$$

$$m_y = \sum_{g \in G} l_g(y) / s_G \quad (5.7b)$$

$$\vec{n} = (m_x - l_\alpha(x), m_y - l_\alpha(y)) \quad (5.7c)$$

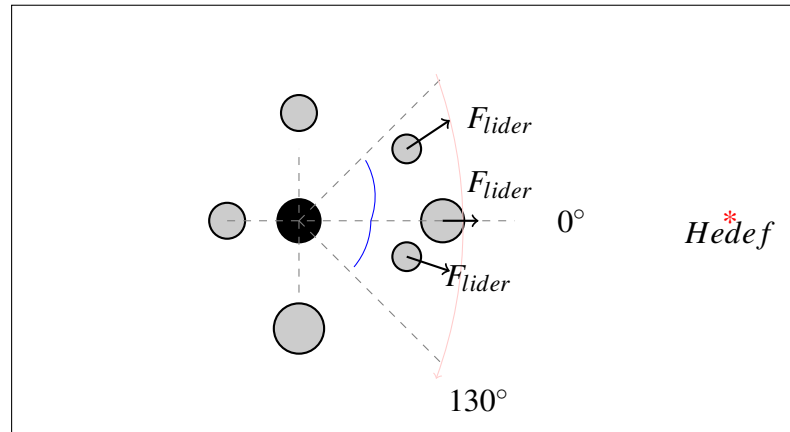
$$\vec{F}_{\alpha group}(t) = \vec{n} [-K_1 \exp(\sqrt{(m_y - l_\alpha(y))^2 + (m_x - l_\alpha(x))^2} / K_2)] \quad (5.7d)$$

5.2.2.6 Lider kuvveti

Grup halinde hareket eden yayaların, grup üyelerinden birinin lider olması durumunda yayalar bu lideri takip etme davranışı sergileme eğilimindedirler. Lider temelde 2 farklı kuvvet uygulamaktadır.

- Lider Çekici Kuvveti
- Lider İtici Kuvveti

Grup içerisindeki yayalara lider tarafından uygulanan çekici kuvvet yardımıyla grup üyelerinin lidere yaklaşmaları ve takip etmeleri sağlanmaktadır. Öte yandan tüm yayalara lider tarafında uygulanan çekici kuvvete bağlı olarak liderin etrafından adeta bir kalkan oluşturabilmektedir. Bunu engellemek amacıyla Şekil 5.6 de görüldüğü gibi, liderin görüş alanındaki yayalar üzerinde itici bir kuvvet uygulamasıyla liderin yolunun açılması sağlanmıştır.



Şekil 5.6 : Lider Kuvveti.

$$d_{\alpha p} = \sqrt{(p_y - l_{\alpha}(y))^2 + (p_x - l_{\alpha}(x))^2} \quad (5.8a)$$

$$\vec{n} = (l_{\alpha}(x) - p_x, l_{\alpha}(y) - p_y) \quad (5.8b)$$

$$\vec{F}_{\alpha l i d e r c e k i c i}(t) = \vec{n}[-K_3 \exp((r_{\alpha p} - d_{\alpha p})/K_4)] \quad (5.8c)$$

$$\vec{F}_{\alpha l i d e r i t i c i}(t) = \vec{n}[K_5 \exp(1/d_{\alpha p})] \quad (5.8d)$$

$$\vec{F}_{\alpha l i d e r}(t) = \vec{F}_{\alpha l i d e r c e k i c i}(t) + \vec{F}_{\alpha l i d e r i t i c i}(t) \quad (5.8e)$$

Algoritma 5: YGDTS Modelinin Sözde Kodu

Sonuç değişkenler: Güncel pozisyon ve hız vektörleri

Girdi değişkenler: Yayaların başlangıçtaki pozisyon ve hız vektörleri

Çıktı değişkenler : Yayaların yeni hızları ve pozisyonları

```

1   $f_1 \leftarrow 0$  ;
2   $f_2 \leftarrow 0$  ;
3   $f_3 \leftarrow 0$  ;
4  foreach Yaya yaya do
5      Arzu edilen yönü hesapla (Denklem 3.3d);
6      Ulaşılmak istenen hızı hesapla (Denklem 3.3b) ;
7      Yönlendirici kuvveti hesapla (Denklem 3.3a);
8       $f_1 \leftarrow f_1 + \text{yönlendiriciKuvvet}$ 
9      foreach yaya1  $\neq$  yaya do
10         if Yaya görüş alanında mı then
11             Yayaların arasındaki uzaklığı hesapla ;
12             if uzaklık  $<$  Yaya etki alanı then
13                 Yaya itici kuvvetini hesapla (Denklem 3.4);
14                  $f_2 \leftarrow f_2 + \text{yayaİticiKuvveti}$ 
15             end
16         end
17     end
18      $\text{enküçük} \leftarrow \text{sonsuz}$  ;
19     foreach Engel ve Duvar do
20         Ele alınan engel ya da duvara olan uzaklığı hesapla ;
21         if uzaklık  $<$  enküçük then
22              $\text{Enyakınduvar} \leftarrow \text{duvar}$  ;
23              $\text{enküçük} \leftarrow \text{uzaklık}$  ;
24             En Yakın duvar üzerindeki en yakın noktayı hesapla ;
25             Duvar kuvvetini hesapla (Denklem 3.5) ;
26             Duvar yönlendirme kuvvetini hesapla (Denklem 5.5) ;
27              $f_3 \leftarrow f_3 + \text{duvarKuvveti}$  ;
28              $f_4 \leftarrow f_4 + \text{yardımcıKuvvet}$ 
29         end
30     end
31 end

```

5.2.2.7 Tüm kuvvetler toplamı

Bu bölümde model içerisinde tanımlanan kuvvetlerin toplamı verilmiştir. Verilen bu kuvvet herhangi bir yaya üzerine etki eden toplam kuvveti ifade etmekte olup, yaya bu kuvvetin etkisi altında hareketini gerçekleştirmektedir. Tüm kuvvetlerin toplamı Denklem 5.9 ile verilmiştir.

$$\vec{f}_{\alpha}^{toplamlam}(t) = \vec{F}_{\alpha}^{Yönlendirici}(t) + \sum_{\beta} \vec{F}_{\alpha\beta}^{İtici}(t) + \sum_W \vec{F}_{\alpha W}^{İtici}(t) + \sum_{\beta} \vec{F}_{\alpha\beta}^{Çekici}(t) + \vec{F}_{\alpha grup}(t) + \vec{F}_{\alpha lider}(t) + \xi(t) \quad (5.9)$$

Algoritma 6: YGDTS Modelinin Sözde Kodu Devamı

```
32  $f_5 \leftarrow 0$  ;
33  $f_6 \leftarrow 0$  ;
34  $f_{toplamlam} \leftarrow 0$  ;
35 foreach Yaya yaya do
36   foreach Grup do
37     if Yaya grubun elemanı mı then
38       Yaya ve grup merkezi arasındaki uzaklığı hesapla ;
39       Grup kuvvetini hesapla (Denklem 5.7d);
40        $f_5 \leftarrow f_5 + grupKuvveti$ 
41     end
42   end
43   foreach yaya!=yaya do
44     if yaya==lider ve yaya ile aynı grupta then
45       Lider Çekici Kuvvetini Hesapla(Denklem 5.8c);
46       if yaya lider yayanın görüş alanında mı then
47         Lider İtici Kuvvetini Hesapla(Denklem 5.8d);
48         Lider kuvvetini hesapla (Denklem 5.8e);
49          $f_6 \leftarrow f_6 + liderKuvveti$ 
50       end
51     end
52   end
53    $f_{toplamlam} \leftarrow f_{toplamlam} + f_1 + f_3 + f_3 + f_4 + f_5 + f_6$  ;
54    $ivme \leftarrow f_{toplamlam}/kütle$  ;
55    $hız \leftarrow hız + ivme * dt$ ;
56    $konum \leftarrow konum + hız * dt$ 
57 end
```



6. SİMÜLASYON VE DENEY SONUÇLARI

Bu bölümde, Bölüm 5’te önerilen modele yönelik farklı senaryolar ele alınmış ve bu durumlar ajan temelli simülasyon modelleme tekniği ile AnyLogic programı yardımıyla gerçekleştirilmiştir. AnyLogic programında yazılan java kodları EK A.2’de verilmiştir.

6.1 Simülasyon Parametreleri

Model parametrelerinin belirlenmesinde Helbing’in çalışmasında yer alan önerilerden faydalanılmıştır [41]. Çalışmada yer alan öneriler dışında modele eklenen yeni parametreler ve mevcut parametreler için oluşturulan animasyon modeli incelenmiş ve meydana gelen bazı hataları (yayaların üst üste çıkışması ya da geri salınım hareketi yapması) önlemeye yönelik olarak Çizelge 6.1’de verilen parametre değerleri belirlenmiştir. Yayalar arasında heterojenliği sağlamak amacıyla, yayalara ait parametreler düzgün dağılıma uygun şekilde tanımlanmış ve böylece yayalar özdeş birer parçacık gibi davranmasının önüne geçilmeye çalışılmıştır.

Çizelge 6.1 : SKM Simülasyon Parametreleri.

Parametre	Birimi	Değeri
$v_{\alpha}^0(0)$	m/sn	düzgün(3;4)
v_{α}^{max}	m/sn	5
m_{α}	kg	düzgün(60,80)
τ_{α}	sn	0,5
A_{α}	N	2000
B_{α}	metre	0.08
r_{α}	metre	düzgün(0,25;0,35)
λ_{α}	skaler	1
C_1	skaler	12
C_2	skaler	24
S	skaler	10
K_1	skaler	0,2
K_2	skaler	1000
K_3	skaler	0,9
K_4	skaler	1000
K_5	skaler	0,2

6.1.1 Simülasyon adım büyüklüğü

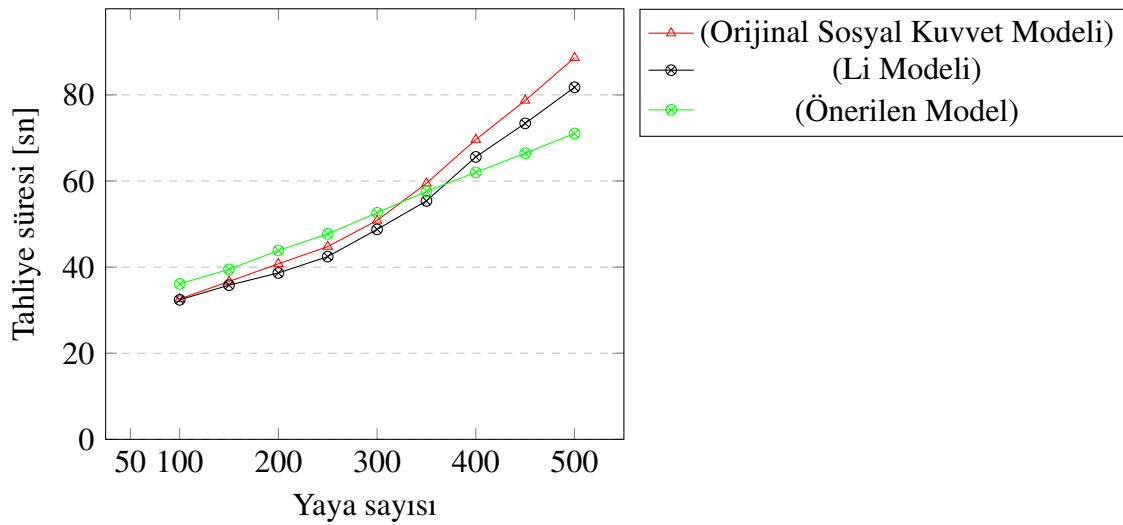
Modelin sahip olduğu diferansiyel denklem yapının çözümü için Euler yönteminden faydalandığına Bölüm 5'te değinilmişti. Euler yönteminin ürettiği çözümün hata oranını azaltmada adım büyüklüğü kritik bir öneme sahiptir. Çalışma içerisinde 0,1; 0,001; 0,005 gibi farklı birçok adım büyüklüğü denenmiş ve bunlar içerisinde en gerçekçi yaya davranışının ve makul işlemci süresinin 0,005 değeri ile olduğu görülmüş ve adım büyüklüğü değeri 0,005 olarak alınmıştır.

6.2 Modelin Doğrulanması

Modelin doğrulanmasında iki farklı yaklaşım kullanılmıştır. Bunlardan birincisinde kurulan model literatürde yer alan çalışmalar ile karşılaştırılmış, ikincisinde ise kurulan modelin acil durumda meydana gelen yaya kalıplarını ortaya çıkarıp çıkarmadığına bakılmıştır.

6.2.1 Literatür ile karşılaştırma

Modelin doğrulanması aşamasında literatürde yer alan çalışmalar ile karşılaştırma yapılmıştır. Karşılaştırma amacıyla Li vd. [31] ile aynı deney ortamı oluşturulmuştur. Çalışmada belirtilemeyen yaya yarıçapları ve çıkış genişliği ise sırasıyla düzgün(0,25; 0,35) m, ve çıkış genişliği 2 m olarak alınmıştır.



Şekil 6.1 : Literatür Karşılaştırması

6.2.2 Acil durumda meydana gelen davranışlar(Kalıplar)

Yayaların panik durumunda iken ve normal durumda iken sergiledikleri temel olgular Bölüm 3.5 de verilmişti. Ortaya konulan modelin panik durumunda meydana gelen olguları ortaya koyma seviyesi modelin doğrulanması amacıyla kullanılmıştır.

6.2.2.1 Engellerden ve diğer yayalardan kaçınma

Önerilen modelde yayalar diğer yayalardan ve engellerden uzak durarak (çarpmadan) hareketlerini gerçekleştirmektedir. Bu durum modelin gerçeği yansıtmasında bir kriter olarak düşünülmüştür.

6.2.2.2 Freezing-by-heating etkisi

Yaya yoğunluğundaki artış ve yayaların çıkışa doğru yönelme isteklerinden dolayı yayaların sürekli olarak yönlerini değiştirdikleri ve farklı yönlerde ilerledikleri görülmüştür. Normal durumlarda meydana gelen düzenli şeritler oluşturma davranışı yerine düzensiz hareket yönleri sergiledikleri görülmüştür.

6.2.2.3 Faster-is-slower etkisi

Oluşturulan model bu etkiyi sergilemekte yetersiz kalmaktadır. Modelin eksik bir yanı olarak kabul edilebilir.

6.2.2.4 Kuyruk oluşturma

Yayaların çıkış bölgelerinde, kuyruk oluşturmaları yayaların normal ya da acil durumda sergiledikleri bir davranış tipidir. Modelin ortaya çıkardığı animasyonu gözlemlediğimiz zaman yayalar çıkışlarda kuyruk oluşturmaktadır.

6.3 Simülasyon Deneyleri

20×20 metre ölçülerinde iki boyutlu bir oda tasarlanmış ve daha sonra bu oda içerisinden yayaların çıkışları üzerinde etkili olabilecek farklı faktörler sırasıyla ele alınmıştır. Gerçekleştirilen deneylerde elde edilen çıktının (tahliye süresi) analizi için Monte Carlo yaklaşımı kullanılmış ve %95 güven aralığı ve 0,5 hata oranı ile en az 100 koşum yapılmıştır. Ve elde edilen koşum sonuçlarının ortalama değeri deney çıktısı olarak kabul edilmiştir.

6.3.1 Yaya sayısı

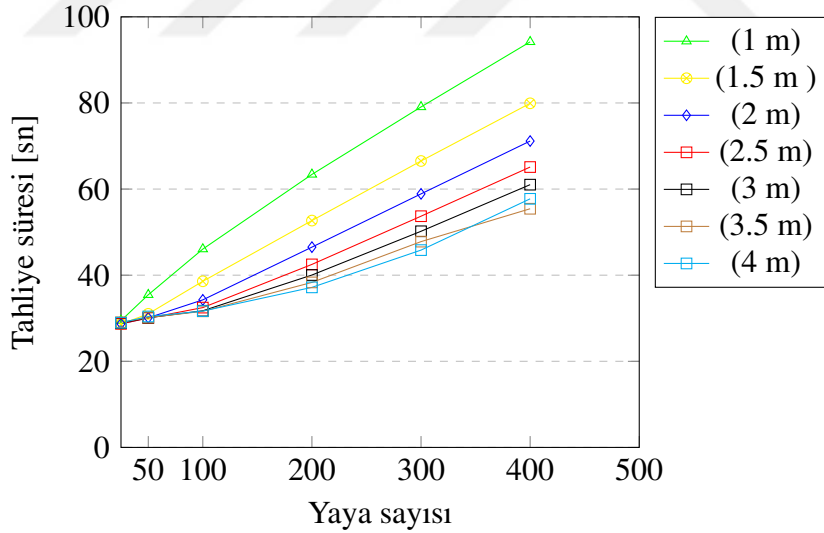
Tüm yayaların bireysel hareket ettiği ve doğrudan çıkışa doğru hareket ettiği durumda yaya sayısı ve simülasyon süresi arasındaki ilişki incelenmiştir. Deneylerde çıkışın genişliği 2 m kabul edilmiştir.

Çizelge 6.2 : Yaya sayısı ve tahliye süresi ilişkisi

Yaya Sayısı	Yaya Yoğunluğu	Tahliye Süresi(sn)
25	0,0625	28,6498
50	0,125	30,112
100	0,25	34,2698
200	0,50	46,5181
300	0,75	58,8858
400	1	71,1543

6.3.2 Çıkışın(Kapı) genişliği

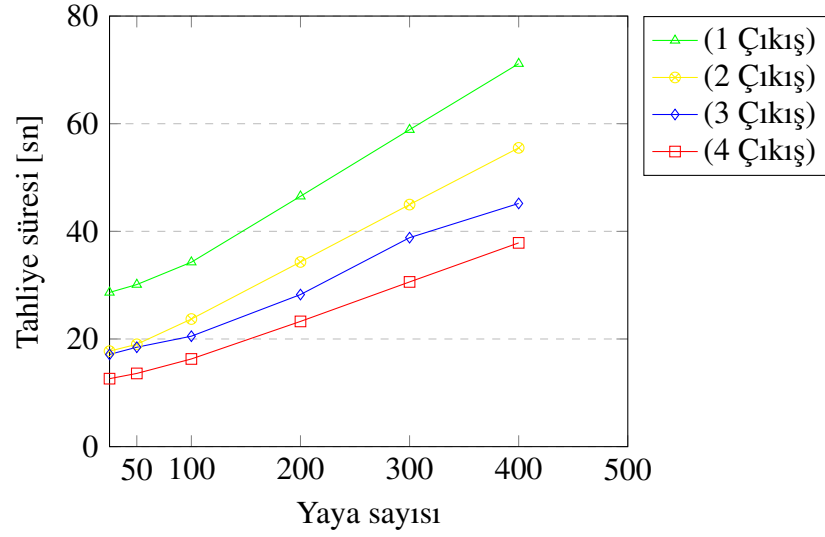
Yayaların çıkış noktasının genişliğinin tahliye süresi üzerindeki etkisini incelemek amacıyla Çizelge C.1’de verilen farklı genişlikler ve yaya sayıları için simülasyon modeli çalıştırılmış ve ortalama tahliye süreleri bulunmuştur.



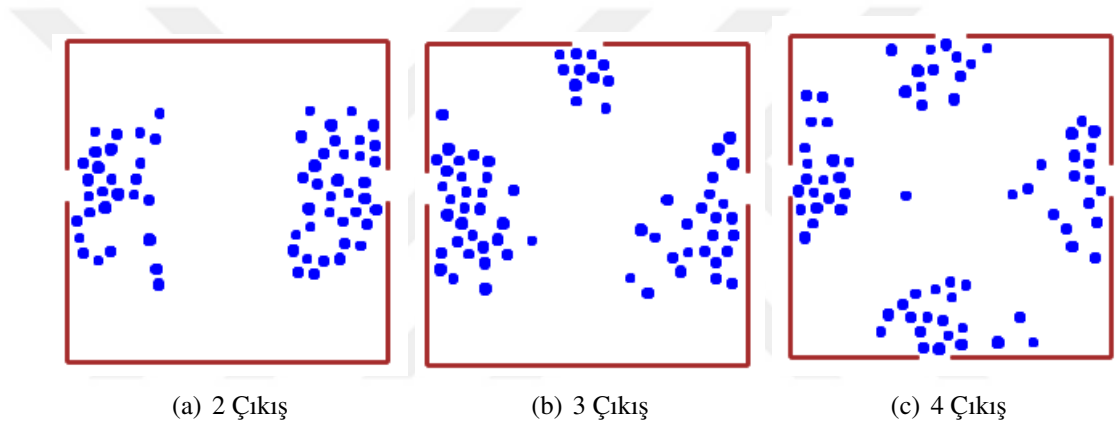
Şekil 6.2 : Çıkış genişliği ve tahliye süresi ilişkisi.

6.3.3 Çıkış sayısı

Tahliye alanı içerisinde yer alan çıkış sayısının tahliye süresi üzerindeki etkisini araştırmak amacıyla Çizelge C.2’de verilen çıkış sayıları ve yaya sayıları için simülasyon modeli çalıştırılmış ve ortalama tahliye süreleri bulunmuştur.



Şekil 6.3 : Çıkış sayısı ve tahliye süresi ilişkisi.



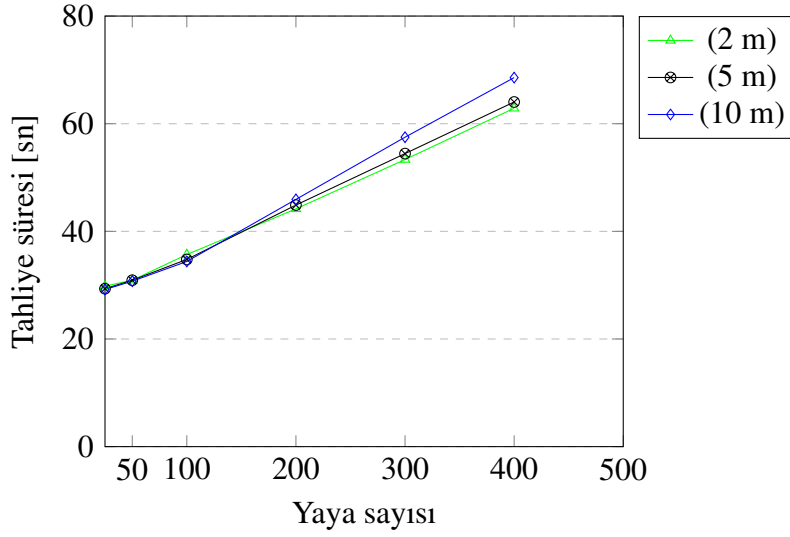
Şekil 6.4 : Modelin 6.27, 5.70 ve 3.90 saniyesinde yayaların hareketinin görüntüsü.

6.3.4 Engellerin olması

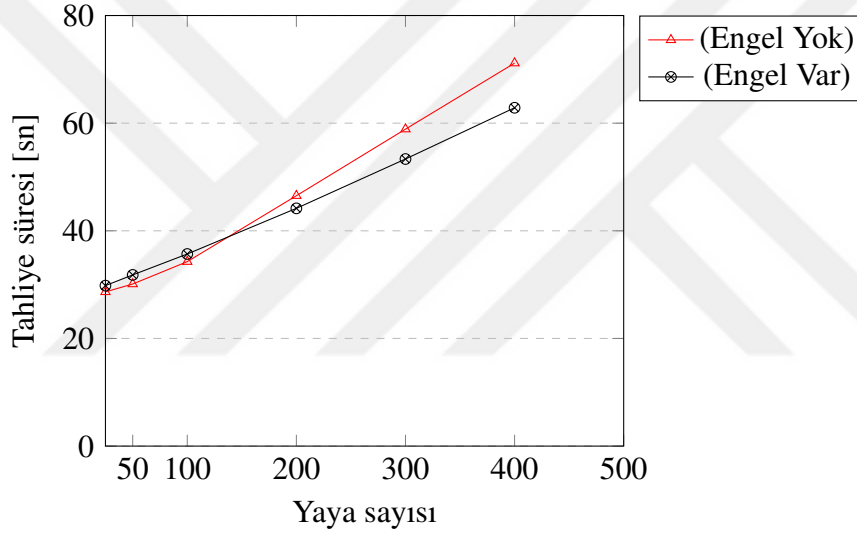
Tahliye alanı içerisinde engel olması durumu ve bu engelin konumunun tahliye süresi üzerindeki etkisini araştırmak amacıyla Çizelge C.3’de verilen farklı engel konumları ve yaya sayıları için simülasyon modeli çalıştırılmış ve ortalama tahliye süreleri bulunmuştur.

6.3.5 Liderin varlığı ve sayısı

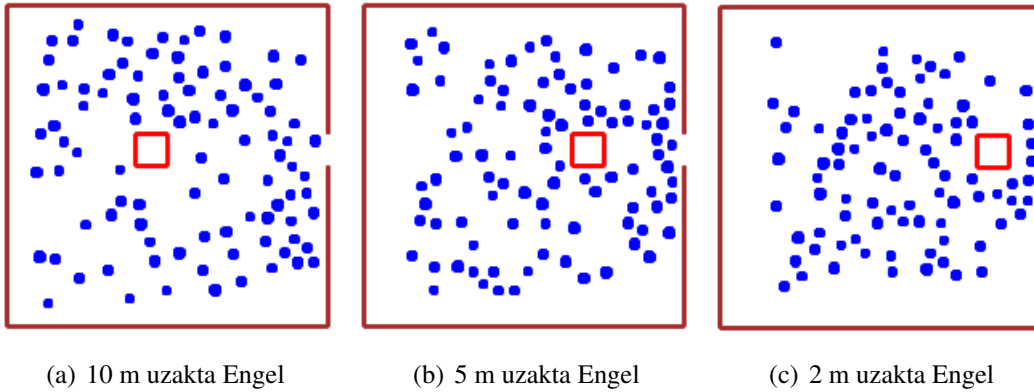
Her bir grup içerisinde bir lider olması ve grup içerisinde yer alan diğer yayaların bu lideri takip etmesinin etkisi incelenmiştir. Şekil 6.10’da animasyon görüntüsünde siyah renkli yayalar gruplar içerisinde yer alan liderleri temsil etmektedir. Grup sayısı kadar lider bulunmaktadır. Burada liderin varlığında yaya gruplarının bölünmesinin tahliye süresine etkisi araştırılmıştır. Gruplar K-means algoritmasına göre oluşturulmuştur,



Şekil 6.5 : Engelin çıkışa uzaklığı ve tahliye süresi ilişkisi.

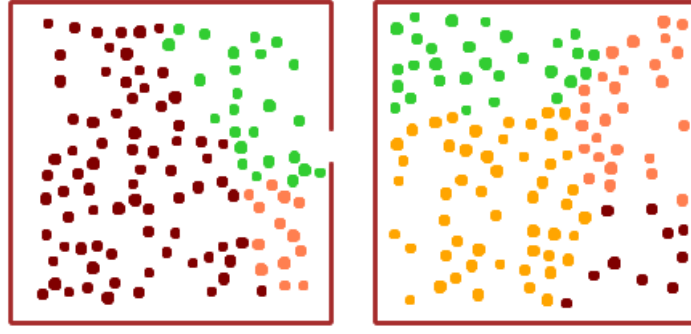


Şekil 6.6 : Engel ve engelsiz durumda tahliye süresi.



Şekil 6.7 : Modelin 1.75, 4.07 ve 4.68 saniyesinde yaya hareketinin görüntüsü.

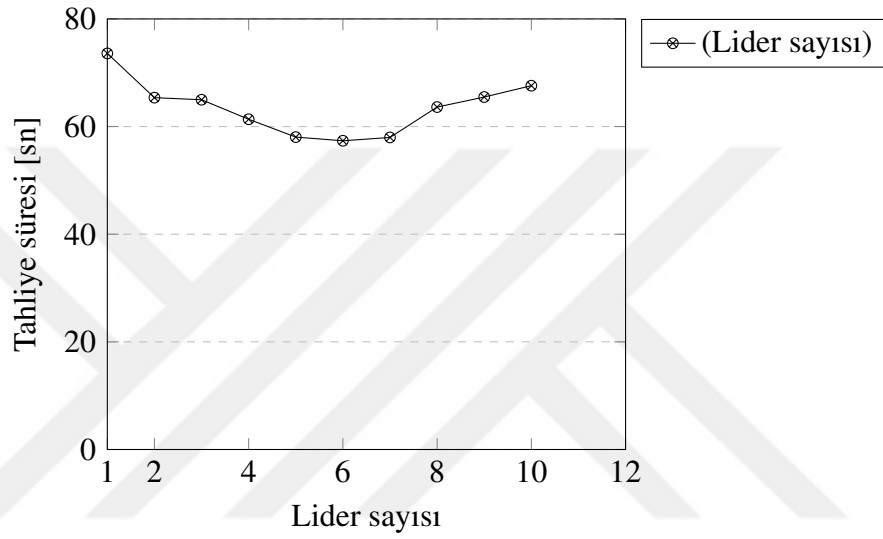
bu yüzden gruplar farklı büyüklüklerde olabilmektedir. Burada liderin varlığında yaya gruplarının bölünmesinin tahliye süresine etkisi araştırılmıştır.



(a) 3 Grup

(b) 4 Grup

Şekil 6.8 : Modelin 1.15 ve 0.79 saniyesinde yaya hareketinin görüntüsü.



Şekil 6.9 : Lider(grup) sayısı ve tahliye süresi ilişkisi.



Şekil 6.10 : Modelin 10 saniyesinde yaya hareketinin görüntüsü.

6.4 Deney Sonuçlarının Yorumlanması

- Yaya sayısındaki(yoğunluğundaki) artış tahliye süresini artırmaktadır. Bu beklenen bir durumdur. Fakat bu artışlar dramatik olmayıp Çizelge 6.2 de görüldüğü gibi yaya yoğunluğunun 0.50'ye çıktığı noktada belirginleşmeye başlamıştır. Bunun sebebi yaya yoğunluğuna bağlı olarak çıkış noktasında meydana gelen bekleme

süreleridir. Yaya yoğunluğ 0.5 ve üzerine çıktığı andan itibaren bekleme sürelerinin arttığı söylenebilir.

- Çıkışın genişliğindeki artışın tahliye süresini azaltan bir etkileye sahiptir. Şekil 6.2'yi incelediğimiz zaman bunu görebiliriz. Fakat bu etki yaya sayısından(yoğunluğundan) bağımsız değildir. Yaya yoğunluğunun 0,0625'den daha küçük olduğu durumlarda çıkışın genişliğinin bir etkisi olmadığı görülmektedir. Bunun üzerindeki yoğunluklarda tahliye süresi aynı şekilde yoğunluğa bağlı olarak farklılık göstermektedir. Bu yüzden çıkış genişlikleri belirlenirken yaya yoğunluğu dikkate alınmadan belirlenen çıkışlar gereksiz geniş ya da yetersiz genişlikte olabilir.
- Çıkış sayısı da tahliye süresini etkileyen diğer bir etmendir. Fakat bu etmende çıkış genişliğinde olduğu gibi yaya yoğunluğuna bağlıdır. Şekil 6.3'ü incelediğimiz zaman yaya yoğunluğu 0,0625 iken 2 çıkış ile 3 çıkış arasında gözle görülür bir fark olmadığı görülmektedir. Dolayısıyla çıkış sayıları belirlenirken hedeflenen tahliye süresi ve yaya yoğunluğuna bağlı olarak belirlenmelidir.
- Tahliye üzerinde etkili olan diğer bir faktör ise engellerin varlığıdır. İncelenen senaryolardan biri engellerin var olma durumudur. Şekil 6.6'yı incelediğimiz zaman yaya yoğunluğu 0,25 ve altındaki iken engel olmaması durumunda yayaların tahliye süresinin daha kısa olduğu görülmektedir. Fakat yaya yoğunluğunun 0,375 ve üzerine çıktığı noktadan itibaren tahliye süresinin engel varlığında daha düşük olduğu görülmektedir. Bunun sebebi olarak çıkış doğrultusunda yer alan bir engelin yüksek yaya yoğunluklarında daha koordineli çıkışlar sağlaması olarak düşünülmektedir. Engeller ile ilgili incelenen diğer bir durum ise engelin konumudur(çıkışa uzaklığı). Şekil 6.5'i incelediğimiz zaman bu etmenin yaya yoğunluğu 1 civarındaki iken etkili olabildiği görülmüştür. Burada da çıkışa en yakın engel durumunda daha düşük tahliye süresinin elde edildiği görülmektedir. Bu da engelin çıkış noktasında daha koordineli davranışa sebep olması düşüncesini destekleyen bir sonuçtur.
- Son olarak incelenen senaryo ise liderlerin (grupların) sayısının tahliye süresine etkisidir. Şekil 6.9'u incelediğimiz zaman, ortaya çıkan sonucun, lider (grup)

sayısındaki artışın yayaların arasındaki rekabeti artırması ile yaya çıkışlarındaki işbirliği ve koordinasyonu azalmasından kaynaklandığı düşünülmektedir.





7. SONUÇ VE ÖNERİLER

Bu bölümde ilk olarak Bölüm 6’da elde edilen deney sonuçlarının gerçek hayatta kullanımı ile ilgili çıkarımlar yapılacak, daha sonra ise genel sonuçlar, öne çıkan farklılıklar ve önerilere değinilecektir.

- Toplu kullanıma açık olan hastane, okul, alışveriş merkezi, stadyum, depo vb. alanlarda insanlar (yayalar) diğer insanlarla birlikte hareket etme eğilimindedir. Bu alanların tasarımı aşamasında acil durum tahliyesi planlamaları yapılırken, tasarlanan çıkışların genişliği, sayısı ve konumunun planlanmasında önerilen model yardımcı bir araç olarak kullanılabilir.
- Belirtilen toplu kullanım alanlarının tahliyesinde koordineli (yönlendirici yardımı) çıkışlarda, yayaların etkin gruplara (grup sayısı) bölünmesinde yardımcı bir araç olarak kullanılabilir.
- Çıkış noktalarına yakın yerlerde yer alan engellerin, tahliyesi süresi üzerindeki pozitif etkisinden yola çıkarak, bölünmüş çıkışlar tasarlanması düşünülebilir.

Bu çalışma kapsamında acil durumda kaçan yaya gruplarını modellemek amacıyla tahliye simülasyonu modeli önerilmiştir. Yapılan çalışmanın temel katkıları şu şekilde özetlenebilir:

- Grup davranışı içeren SKM ortaya konulmuş ve SKM’nin bazı eksiklikleri giderilmeye çalışılmıştır.
- Model içerisinde yayaların sahip olduğu heterojen davranışların etkisini de göstermek amacıyla model parametreleri düzgün dağılıma uygun olarak verilmiştir.
- Yaya gruplarının oluşturulmasında Sosyal Uzaklık teorisinden faydalanılarak küçük yaya gruplarının oluşturulması sağlanmış ve sosyal bir teoriye dayalı olarak gerçekçi yaya grupları üretilmiştir.

- Büyük yaya gruplarının oluşması durumu için ise K-means kümeleme algoritması kullanılmış ve büyük yaya gruplarının (sürülerinin) oluşma durumu dikkate alınmıştır.
- Acil durumda yayaların sergiledikleri davranışların sürü davranışına benzer olmasından yola çıkarak yaya gruplarını bir arada tutmayı sağlamak amacıyla Boids algoritması birleştirme fonksiyonundan faydalanarak SKM'ye grup kuvveti eklenmiştir.
- Grup davranışı tiplerinden biri olan lideri takip etme davranışı incelenmiş ve bu amaçla SKM'ye lider çekici ve itici kuvvetleri eklenmiştir.
- Literatürde yer alan çalışmaların çoğu yaya hareket alanını modellemede kesikli bir yaklaşım olan hücresel otomat yöntemini kullanmaktadır. Fakat yayaların hareket alanları hiçbir zaman kesikli değildir. Bu yüzden bu çalışmada yayaların hareket alanı sürekli olarak tanımlanmıştır.
- Literatürde yer alan bazı çalışmalar da SKM'nin implementasyonunda zaman sürekli olarak alınmış ve diferansiyel denklem çözücüler yardımıyla modeller çözülmüştür. Bu tür bir yaklaşım daha doğru olmakla birlikte yaya sayısındaki artışa bağlı olarak çok hantal bir hale gelmekte ve hesaplama süresi çok uzun olmaktadır. Bu problemi çözmek amacıyla bu çalışmada zaman kesikli olarak alınmış ve uygun bir adım büyüklüğü ile doğruya yakın ve makul hesaplama süreleri elde edilmiştir.

Gelecek çalışmalarda bulanık mantığa dayalı yaya hareket modellerinin daha gerçekçi yaya davranışının ortaya çıkarılmasında faydalı olacağı düşünülmektedir. Bu yüzden bulanık SKM oluşturulması amaçlanmaktadır. Öte yandan yaya gruplarının oluşturulmasında Minimum Kapsayan Ağaç algoritması ya da farklı sezgisel algoritmaların kullanılması düşünülmektedir.

KAYNAKLAR

- [1] **North, M.J. ve Macal, C.M.** (2007). *Managing business complexity: discovering strategic solutions with agent-based modeling and simulation*, Oxford University Press.
- [2] **Yu, W., Chen, R., Dong, L. ve Dai, S.** (2005). Centrifugal force model for pedestrian dynamics, *Physical Review E*, 72(2), 026112.
- [3] **Okazaki, S. ve Matsushita, S.** (1993). A study of simulation model for pedestrian movement with evacuation and queuing, *International Conference on Engineering for Crowd Safety*, cilt271.
- [4] **Reynolds, C.W.** (1987). Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH computer graphics*, 21(4), 25–34.
- [5] **Lo, S.M., Huang, H.C., Wang, P. ve Yuen, K.** (2006). A game theory based exit selection model for evacuation, *Fire Safety Journal*, 41(5), 364–369.
- [6] **Mesmer, B.L. ve Bloebaum, C.L.** (2016). Modeling decision and game theory based pedestrian velocity vector decisions with interacting individuals, *Safety science*, 87, 116–130.
- [7] **Guan, J., Wang, K. ve Chen, F.** (2016). A cellular automaton model for evacuation flow using game theory, *Physica A: Statistical Mechanics and its Applications*, 461, 655–661.
- [8] **Bakuli, D.L. ve Smith, J.M.** (1996). Resource allocation in state-dependent emergency evacuation networks, *European Journal of Operational Research*, 89(3), 543–555.
- [9] **Helbing, D., Molnár, P., Farkas, I.J. ve Bolay, K.** (2001). Self-organizing pedestrian movement, *Environment and planning B: planning and design*, 28(3), 361–383.
- [10] **Henderson, L.F.** (1974). On the fluid mechanics of human crowd motion, *Transportation research*, 8(6), 509–515.
- [11] **Hoogendoorn, S. ve Bovy, P.** (2000). Gas-kinetic modeling and simulation of pedestrian flows, *Transportation Research Record: Journal of the Transportation Research Board*, (1710), 28–36.
- [12] **Kormanová, A. ve diğeri** (2014). A review on macroscopic pedestrian flow modelling, *Acta Informatica Pragensia*, 2(2), 39–50.

- [13] **Di Gangi, M.** (2009). Safety of users in road evacuation: modelling and DSS for pedestrian outflow, *WIT Transactions on Ecology and the Environment*, 120, 497–508.
- [14] **Kuligowski, E.D.** (2004). Review of 28 egress models, *NIST SP*, 1032, 339–352.
- [15] **Helbing, D. ve Molnar, P.** (1995). Social force model for pedestrian dynamics, *Physical review E*, 51(5), 4282.
- [16] **Helbing, D., Farkas, I.J., Molnar, P. ve Vicsek, T.** (2002). Simulation of pedestrian crowds in normal and evacuation situations, *Pedestrian and evacuation dynamics*, 21(2), 21–58.
- [17] **Helbing, D., Buzna, L., Johansson, A. ve Werner, T.** (2005). Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions, *Transportation science*, 39(1), 1–24.
- [18] **Heliövaara, S., Korhonen, T., Hostikka, S. ve Ehtamo, H.** (2012). Counterflow model for agent-based simulation of crowd dynamics, *Building and Environment*, 48, 89–100.
- [19] **Han, Y. ve Liu, H.** (2017). Modified social force model based on information transmission toward crowd evacuation simulation, *Physica A: Statistical Mechanics and its Applications*, 469, 499–509.
- [20] **Yang, X., Dong, H., Wang, Q., Chen, Y. ve Hu, X.** (2014). Guided crowd dynamics via modified social force model, *Physica A: Statistical Mechanics and its Applications*, 411, 63–73.
- [21] **Kretz, T.** (2015). On oscillations in the social force model, *Physica A: Statistical Mechanics and its Applications*, 438, 272–285.
- [22] **Farina, F., Fontanelli, D., Garulli, A., Giannitrapani, A. ve Prattichizzo, D.** (2017). Walking Ahead: The Headed Social Force Model, *PloS one*, 12(1), e0169734.
- [23] **Johansson, F., Peterson, A. ve Tapani, A.** (2015). Waiting pedestrians in the social force model, *Physica A: Statistical Mechanics and its Applications*, 419, 95–107.
- [24] **Fang, J., El-Tawil, S. ve Aguirre, B.** (2016). Leader–follower model for agent based simulation of social collective behavior during egress, *Safety Science*, 83, 40–47.
- [25] **von Sivers, I., Templeton, A., Künzner, F., Köster, G., Drury, J., Philippides, A., Neckel, T. ve Bungartz, H.j.** (2016). Modelling social identification and helping in evacuation simulation, *Safety science*, 89, 288–300.
- [26] **Song, X., Ma, L., Ma, Y., Yang, C. ve Ji, H.** (2016). Selfishness-and Selflessness-based models of pedestrian room evacuation, *Physica A: Statistical Mechanics and its Applications*, 447, 455–466.
- [27] **Lemercier, S. ve Auberlet, J.M.** (2016). Towards more behaviours in crowd simulation, *Computer Animation and Virtual Worlds*, 27(1), 24–34.

- [28] **Zhou, Z., Cai, Y., Ke, R. ve Yang, J.** (2017). A collision avoidance model for two-pedestrian groups: Considering random avoidance patterns, *Physica A: Statistical Mechanics and its Applications*, 475, 142–154.
- [29] **You, L., Hu, J., Gu, M., Fan, W. ve Zhang, H.** (2016). The simulation and analysis of small group effect in crowd evacuation, *Physics Letters A*, 380(41), 3340–3348.
- [30] **Qiu, F. ve Hu, X.** (2010). Modeling group structures in pedestrian crowd simulation, *Simulation Modelling Practice and Theory*, 18(2), 190–205.
- [31] **Li, Y., Liu, H., Liu, G.p., Li, L., Moore, P. ve Hu, B.** (2017). A grouping method based on grid density and relationship for crowd evacuation simulation, *Physica A: Statistical Mechanics and its Applications*, 473, 319–336.
- [32] **Lovreglio, R., Fonzone, A., dell’Olio, L. ve Borri, D.** (2016). A study of herding behaviour in exit choice during emergencies based on random utility theory, *Safety science*, 82, 421–431.
- [33] **Ma, Y., Yuen, R.K.K. ve Lee, E.W.M.** (2016). Effective leadership for crowd evacuation, *Physica A: Statistical Mechanics and its Applications*, 450, 333–341.
- [34] **Dossetti, V., Bouzat, S. ve Kuperman, M.** (2017). Behavioral effects in room evacuation models, *Physica A: Statistical Mechanics and its Applications*, 479, 193–202.
- [35] **Zheng, X. ve Cheng, Y.** (2011). Modeling cooperative and competitive behaviors in emergency evacuation: A game-theoretical approach, *Computers & Mathematics with Applications*, 62(12), 4627–4634.
- [36] **Qiu, F. ve Hu, X.** (2010). Modeling dynamic groups for agent-based pedestrian crowd simulations, *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, cilt 2, IEEE, s.461–464.
- [37] **Ren, Z., Charalambous, P., Bruneau, J., Peng, Q. ve Pettr , J.** (2016). Group Modeling: A Unified Velocity-Based Approach, *Computer Graphics Forum*.
- [38] **Benchang,** (2008), Emergence and Navigating Space, <http://spacecollective.org/userdata/XkgE2Ij2/1200470109/boidfields.jpg>.
- [39] **Kushalnagar, R.S., Ludie, S.A. ve Kushalnagar, P.** (2011). Multi-view platform: an accessible live classroom viewing approach for low vision students, *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, ACM, s.267–268.
- [40] **T.Hall, E.** (1966). *The Hidden Dimensions*, ANCHOR BOOKS EDITION.
- [41] **Helbing, D., Farkas, I. ve Vicsek, T.** (2000). Simulating dynamical features of escape panic, *Nature*, 407(6803), 487–490.



EKLER

EK A.1 : AnyLogic Programı

EK A.2 : Java Kodları

EK A.3 : Deney Sonuçları Tabloları





EK A.1: AnyLogic Programı

Çalışma kapsamında önerilen modelin implementasyonu sürecinde AnyLogic programından faydalanılmıştır. AnyLogic, ajan temelli benzetim, sistem dinamiği modellemesi ve kesikli olay benzetim modellerinin implementasyonuna imkan sağlayan çok yönlü bir simülasyon aracıdır. Java temelli bir program olup, kullanıcıların kendi yazdıkları java kodları yardımıyla benzetim modellerin oluşturmalarına imkan sağlamakta ve animasyon desteği ile oluşturulan modellerin görselleştirilmesini sağlamaktadır. Üretim, tedarik zinciri, yaya ve araç trafiği, havacılık, sağlık hizmetleri, proje yönetimi, bilgi teknolojileri alt yapısı vb. alanlarda benzetim modellerinin oluşturulması AnyLogic ile mümkündür. Bu çalışmada sağladığı esneklik ve animasyon desteğinden dolayı bu program tercih edilmiştir.





EK A.2: Java Kodları

```
/**
 * Vektor
 */
public class Vektor implements Serializable {
    private double x;
    private double y;

    public Vektor() {
        x=0;
        y=0;
    }

    public Vektor(double x,double y) {
        this.x=x;
        this.y=y;
    }

    public double getX() {
        return x;
    }

    public double getY() {
        return y;
    }

    public void setX(double x) {
        this.x=x;
    }

    public void setY(double y) {
        this.y=y;
    }

    @Override
    public String toString() {
        return super.toString();
    }

    private static final long serialVersionUID = 1L;

}

/**
```

```

* VektorHesabi
*/
public abstract class VektorHesabi implements Serializable {
public static Vektor toplam(Vektor v1, Vektor v2) {
return new Vektor(v1.getX()+v2.getX(), v1.getY()+v2.getY());
}

public static Vektor fark(Vektor v1, Vektor v2) {
return new Vektor(v1.getX()-v2.getX(), v1.getY()-v2.getY());
}

public static Vektor topla(Vektor v1, Vektor v2) {
return new Vektor(v1.getX()+v2.getX(), v1.getY()+v2.getY());
}
public static Vektor skalar(Vektor v1, double a) {
return new Vektor(a*v1.getX(), a*v1.getY());
}

public static double uzunluk(Vektor v1) {
return sqrt(v1.getX()*v1.getX()+v1.getY()*v1.getY());
}

public static Vektor birim(Vektor v1) {
return skalar(v1, (double) 1/uzunluk(v1));
}

public static double noktacarpim(Vektor v1, Vektor v2){

return (v1.getX()*v2.getX()+v1.getY()*v2.getY());
}

public static Vektor caprazcarpim(Vektor v1, Vektor v2) {

return new Vektor(v1.getX()*v2.getY(), v1.getY()*v2.getX());
}

public static Vektor yon (Vektor v1, Vektor v2){

Vektor ara=VektorHesabi.fark(v2,v1);
double carpan=1/VektorHesabi.uzunluk(ara);
Vektor aranan=VektorHesabi.skalar(ara,carpan);
return aranan;

}

public static double yonacisi(Vektor v1){

```

```

return (atan(v1.getY()/v1.getX()))*PI/180;
}

public static double Vektorlerarasi_aci
(Vektor v1, Vektor v2){

double a=noktacarpim(v1,v2)/(uzunluk(v1)*uzunluk(v2));

return acos(a);
}

public static double aciHesabi(Vektor v1, Vektor v2){

double aci=atan((v2.getY()-v1.getY())/
(v2.getX()-v1.getX()));
if (v1.getX()<v2.getX()) aci+=PI;
return aci;
}

public static Vektor rotate(Vektor v1) {
return new Vektor(-v1.getY(), v1.getX());
}
public static Vektor rotate_saatyonu(Vektor v1) {
return new Vektor(v1.getY(), -1*v1.getX());
}
public static Vektor rotate2(Vektor v1) {
return new Vektor(v1.getY(), v1.getX());
}

public static Vektor rotate3(Vektor v1) {
return new Vektor(-1*v1.getX(), v1.getY());
}

public static Vektor rotate4(Vektor v1) {
return new Vektor(v1.getX(), -1*v1.getY());
}

public static Vektor rotate1(Vektor v1) {
return new Vektor(cos(45)*v1.getX(), sin(45)*v1.getX());
}

@Override
public String toString() {
return super.toString();
}

```

```

private static final long serialVersionUID = 1L;

}

/**
 * DuvarHesabi
 */
public abstract class DuvarHesabi implements Serializable {

/**
 * Default constructor
 */

public DuvarHesabi() {

}

public static double uzunluk(Vektor v1, Vektor v2){
return sqrt(pow((v2.getY()-v1.getY()),2) +
pow((v2.getX()-v1.getX()),2));
}

public static double uzunluktek(Vektor v1)
{

return sqrt(pow(v1.getX(),2)+pow(v1.getY(),2));
}

public static double noktaUzaklik(Vektor nokta ,
Vektor v1, Vektor v2){

return abs(((v2.getY()-v1.getY())*nokta.getX())-
((v2.getX()-v1.getX())*nokta.getY())+(v2.getX()*v1.getY())-
(v1.getX()*v2.getY()))/uzunluk(v1,v2);

}

public static double enKisa(Vektor nokta ,Vektor v1, Vektor v2){

double A=nokta.getX()-v1.getX();
double B=nokta.getY()-v1.getY();
double C=v2.getX()-v1.getX();
double D=v2.getY()-v1.getY();

```

```

double nokta = A * C + B * D;
double uzunluk = C * C + D * D;
double bayrak = -1;
if (uzunluk != 0)
    bayrak = nokta / uzunluk;

double X, Y;

if (bayrak < 0) {
    X = v1.getX();
    Y = v1.getY();
}
else if (bayrak > 1) {
    X = v2.getX();
    Y = v2.getY();
}
else {
    X = v1.getX() + bayrak * C;
    Y = v1.getY() + bayrak * D;
}

double DX = nokta.getX() - X;
double DY = nokta.getY() - Y;
return sqrt(DX * DX + DY * DY);

}

public static Vektor enyakin (Vektor nokta ,
Vektor v1, Vektor v2)
{

    Vektor XY = VektorHesabi.fark(nokta , v1);
    Vektor XZ = VektorHesabi.fark(v2, v1);
    double buyuklukXZ = pow(uzunluktek(XZ),2);
    double XYZ = VektorHesabi.noktacarpim(XY, XZ);
    double mesafe = XYZ/buyuklukXZ;
    if(mesafe < 0) return v1;
    else if(dist > 1) return v2;
    else return VektorHesabi.topla(v1, VektorHesabi.
skalar(XZ, mesafe));

}

```

@Override

```

public String toString() {
    return super.toString();
}

/**
 * This number is here for model snapshot storing purpose<br>
 * It needs to be changed when this class gets changed
 */
private static final long serialVersionUID = 1L;

}

/**
 * Group
 */
public class Group implements Serializable {

    ArrayList<Yaya> elemanlar;

    /**
     * Default constructor
     */
    public Group() {
        elemanlar=new ArrayList<Yaya>();
    }

    public int size() {
        return elemanlar.size();
    }
    public void setEleman(Yaya eleman) {
        elemanlar.add(eleman);
    }

    public void silEleman(Yaya eleman) {
        elemanlar.remove(eleman);
    }

    public Yaya getEleman(int no) {
        return elemanlar.get(no);
    }

    @Override
    public String toString() {
        return super.toString();
    }

    /**
     * This number is here for model snapshot storing purpose<br>

```



```

* It needs to be changed when this class gets changed
*/
private static final long serialVersionUID = 1L;

}

/**
 * Color
 */
public class UniqueColor implements Serializable {

/**
 * Default constructor
 */
public UniqueColor() {

}

public static Color Next(int _colorIndex)
{
Color col;
Random rand = new Random();

byte red = ( byte )( _colorIndex & 0x000000FF );
byte green = ( byte )( ( _colorIndex & 0x0000FF00 ) >> 8 );
byte blue = ( byte )( ( _colorIndex & 0x00FF0000 ) >> 16 );
if(_colorIndex==0)
col=new Color(128,0,0);

else if(_colorIndex==1)
col=new Color(255,127,80);

else if(_colorIndex==2)
col=new Color(50,205,50);

else if(_colorIndex==3)
col=new Color(255,165,0);

else if(_colorIndex==4)
col=new Color(0,0,128);

else if(_colorIndex==5)
col=new Color(0,0,0);

else if(_colorIndex==6)

```

```

col=new Color(255,0,0);

else if(_colorIndex==7)
col=new Color(0,128,128);

else if(_colorIndex==8)
col=new Color(255,255,0);

else if(_colorIndex==9)
col=new Color(128,128,0);

else if(_colorIndex==10)
col=new Color(245,255,250);

else if(_colorIndex==11)
col=new Color(0,128,0);

else if(_colorIndex==12)
col=new Color(0,255,255);

else if(_colorIndex==13)
col=new Color(0,0,255);

else if(_colorIndex==14)
col=new Color(128,0,128);

else if(_colorIndex==15)
col=new Color(255,0,255);

else if(_colorIndex==16)
col=new Color(128,128,128);

else if(_colorIndex==17)
col=new Color(255,192,203);

else if(_colorIndex==18)
col=new Color(165,42,42);

else if(_colorIndex==19)
col=new Color(245,245,220);

else
col=new Color(255,255,255);
return col;
}

```

@Override

```

public String toString() {
return super.toString();
}

private static final long serialVersionUID = 1L;

}

-----
/** SKModel */

// Parameters

public double d_t;
public int agentNumber;
public int kume_sayisi;

// Plain Variables

public double sayac;

// Collection Variables
public ArrayList <Group > kumeler = new ArrayList<Group>();

double x,y;
boolean sonuc;

for (int i=0;i<agentNumber;i++) {
Yaya yay=add_yayalar();

do {
x=uniform(103,297);

y=uniform(103,297);

sonuc=false;

for (Yaya yaya: yayalar) {
if (yay!=yaya) {
if (sqrt(sqr(yaya.konum.getX()-x)+
sqr(yaya.konum.getY()-y))<yay.yaricap+yaya.yaricap+1) {
sonuc=true;
break;
}
}
}
} while (sonuc);

```

```
yay.konum=new Vektor(x,y);
yay.ilkKonum=new Vektor(x,y);

}
```

```
Duvar duvar;
```

```
duvar=add_duvarlar();
duvar.ciz(new Vektor(100,100), new Vektor(300,100));
duvar.line.setColor(
brown);
```

```
duvar.line.setLineWidth(2);
duvar=add_duvarlar();
duvar.ciz(new Vektor(100,100), new Vektor(100,300));
duvar.line.setColor(
brown);
duvar.line.setLineWidth(2);
duvar=add_duvarlar();
duvar.ciz(new Vektor(100,300), new Vektor(300,300));
duvar.line.setColor(
brown);
duvar.line.setLineWidth(2);
duvar=add_duvarlar();
duvar.ciz(new Vektor(300,100), new Vektor(300,180));
duvar.line.setColor(
brown);
duvar.line.setLineWidth(2);
```

```
duvar=add_duvarlar();
duvar.ciz(new Vektor(300,300), new Vektor(300,200));
duvar.line.setColor(
brown);
duvar.line.setLineWidth(2);
```

```
ShapePixel hedef=new ShapePixel();
hedef.setX(820);
hedef.setY(350);
presentation.add(hedef);
```

```
// int iterasyon=1000;
// int sayac=0;
```

```

int kumeNo=0;
boolean degisti;
double [] mx=new double[kume_sayisi];
double [] my=new double[kume_sayisi];
double toplamx,toplamy;

Color [] color=new Color[kume_sayisi];
//Rassal merkezler...

for (int i=0;i<kume_sayisi;i++)
{
mx[i]=uniform(103,297);
my[i]=uniform(103,297);
kumeler.add(new Group());

}

//Kumeleri olustur...

for (Yaya a: yayalar)
{
kumeNo=uzaklik(a,mx,my);
if (kumeNo!=a.grup_no)
{
a.grup_no=kumeNo;
degisti=true;
kumeler.get(kumeNo).setEleman(a);

}
}

do
{

degisti=false;
for (Yaya a: yayalar)
{
kumeNo=uzaklik(a,mx,my);
if (kumeNo!=a.grup_no)
{
kumeler.get(a.grup_no).silEleman(a);
a.grup_no=kumeNo;
degisti=true;
kumeler.get(kumeNo).setEleman(a);

}
}
}

```

```

}
for (int i=0;i<kume_sayisi;i++)
{
toplamx=0;
toplamy=0;
for (Yaya a:kumeler.get(i).elemanlar)
{
toplamx+=a.getX();
toplamy+=a.getY();
}
mx[i]=toplamx/kumeler.get(i).elemanlar.size();
my[i]=toplamy/kumeler.get(i).elemanlar.size();
}
sayac++;
} while (degisti);

for (int i=0;i<kume_sayisi;i++)
{
Color rsK=UniqueColor.Next(i);
for (Yaya a:kumeler.get(i).elemanlar) a.renk=rsK ;
}

for (Group grup2:kumeler) grup2.getEleman(0).lider=true;

/** Uzaklik fonksiyonu */

int uzaklik( Yaya yaya, double [] x, double [] y ) {

int enk=0;
double uzaklik=infinity;
double uz;

for (int i=0;i<kume_sayisi;i++)
{
uz=sqr(x[i]-yaya.konum.getX())+sqr(y[i]-yaya.konum.getY());
if (uz<uzaklik)
{
uzaklik=uz;
enk=i;
}
}
}

```

```

return enk;
}

```

```

-- -- -- -- --
/** Yaya ajani */

```

```

// Parameters

```

```

public double dt;
public double v0;
public double vmax;
public double yaricap;

```

```

// Plain Variables

```

```

public Vektor konum;
public Vektor toplamKuvvet;
public static double tahliye;
public Color renk;
public Vektor hiz;
private double istenenHiz;
private double A1;
private double A2;
private double B1;
private double B2;
public Vektor hedef;
public double etkiAlani;
public int grup;
public static final double gorusYCapi = 50 ;
public double liderlik;
public boolean lider;
public static final double gorusAcisi = PI*1.05555555 ;
public int grup_no;

```

```

void kuvvetHesabi( ) {

```

```

    Vektor yon=VektorHesabi.birim(VektorHesabi.fark
    (hedef,this.konum));

```

```

    if(time()!=0){
        Vektor alinanYol=new Vektor();
        alinanYol=VektorHesabi.fark(konum,ilkKonum);
        double yolMesafesi=VektorHesabi.uzunluk(alinanYol);
        double ortalamaHiz=yolMesafesi/time();
    }
}

```

```

double bar=ortalamaHiz/v0;
istenenHiz=(1-bar)*v0+bar*vmax;

}

Vektor yeniHiz=VektorHesabi.skalar(yon,istenenHiz);

Vektor farkHiz=VektorHesabi.fark(yeniHiz,hiz);

Vektor yonlendiriciKuvvet=VektorHesabi.skalar
(VektorHesabi.skalar(farkHiz,(double) 1/dt),1);

toplamKuvvet=yonlendiriciKuvvet;

Vektor yayaIticiKuvvet=new Vektor();
Vektor yayaIticiKuvvet1;
Vektor yayaIticiKuvvet2;
Vektor yayaEtkilesimKuvvet;
Vektor yayaSurtunmeKuvvet;
Vektor t;
Vektor liderKuvveti=new Vektor(0,0);
Vektor liderKuvveti1=new Vektor();
Vektor liderCekiciKuvveti=new Vektor(0,0);
Vektor liderCekiciKuvveti1=new Vektor(0,0);
double uzak=0;

double yonaci=VektorHesabi.yonacisi(yon);
double Q=0;

double uzaklik=0,carpan=0;
for (Yaya yaya:sKModel.yayalar) {
    uzaklik=VektorHesabi.uzunluk(VektorHesabi.fark(konum,yaya.konum));
    if (yaya!=this && uzaklik<etkiAlani) {

        if (gorusAlaninda(VektorHesabi.aciHesabi
            (konum,yaya.konum),yonaci)){
            Vektor n=VektorHesabi.fark(konum,yaya.konum);
            n=VektorHesabi.skalar(n,(double)1/uzaklik);

            if (((yaricap+yaya.yaricap)-uzaklik)>=0)
                carpan=0;
            else

```



```

carpan=A1*exp(((yaricap+yaya.yaricap)-uzaklik)/B1);
yayaIticiKuvvet1=VektorHesabi.skalar(n,carpan);

if((yaricap+yaya.yaricap)-VektorHesabi.uzunluk
(VektorHesabi.fark(konum,yaya.konum))>=0)
Q=(yaricap+yaya.yaricap)-VektorHesabi.uzunluk
(VektorHesabi.fark(konum,yaya.konum));
else
Q=0;
yayaEtkilesimKuvvet=VektorHesabi.skalar(n,Q);

yayaEtkilesimKuvvet=VektorHesabi.skalar
(yayaEtkilesimKuvvet,12);

yayaSurtunmeKuvvet=VektorHesabi.skalar
(VektorHesabi.rotate(n),
(VektorHesabi.noktacarpim(VektorHesabi.fark
(hiz,yaya.hiz),n)));
yayaSurtunmeKuvvet=VektorHesabi.skalar
(yayaSurtunmeKuvvet,Q);
yayaSurtunmeKuvvet=VektorHesabi.skalar
(yayaSurtunmeKuvvet,24);
yayaIticiKuvvet=VektorHesabi.topla(yayaIticiKuvvet,
yayaIticiKuvvet1);
yayaIticiKuvvet=VektorHesabi.topla(yayaIticiKuvvet,
yayaEtkilesimKuvvet);
yayaIticiKuvvet=VektorHesabi.topla(yayaIticiKuvvet,
yayaSurtunmeKuvvet); yayaIticiKuvvet=VektorHesabi.skalar

}
}

toplamKuvvet=VektorHesabi.toplam
(toplamKuvvet,yayaIticiKuvvet);

Vektor grupMerkezKuvveti=new Vektor();
double uzakGrup=0;
double [] totalKonumX=new double[sKModel.kume_sayisi];
double [] totalKonumY=new double[sKModel.kume_sayisi];

for(int i=0;i<sKModel.kume_sayisi;i++){
for(Yaya a:sKModel.kumeler.get(i).elemanlar){

```

```

totalKonumX[i]=totalKonumX[i]+a.konum.getX();
totalKonumY[i]=totalKonumY[i]+a.konum.getY();

}

totalKonumX[i]=totalKonumX[i]/sKModel.kumeler.get(i).
elemanlar.size();
totalKonumY[i]=totalKonumY[i]/sKModel.kumeler.get(i).
elemanlar.size();

}

for(int i=0;i<sKModel.kume_sayisi;i++){

if(sKModel.kumeler.get(i).elemanlar.contains(this)){
uzakGrup=sqrt(sqr(totalKonumX[i]-this.konum.getX()+
sqr(totalKonumY[i]-this.konum.getY())));

Vektor n=VektorHesabi.fark(konum,new Vektor(totalKonumX[i],
totalKonumY[i]));
grupMerkezKuvveti=VektorHesabi.skalar(n,-0.2*exp(uzakGrup/1000));

}

}

toplamKuvvet=VektorHesabi.toplam(toplamKuvvet,grupMerkezKuvveti);

for(int i=0;i<sKModel.kume_sayisi;i++){
for(Yaya yaya:sKModel.kumeler.get(i).elemanlar){
if(yaya!=this && yaya.lider==true &&
sKModel.kumeler.get(i).elemanlar.contains(this)){

uzak=DuvarHesabi.uzunluk(konum,yaya.konum);

Vektor n=VektorHesabi.fark(konum,yaya.konum);

liderCekiciKuvveti=VektorHesabi.skalar(n,
-0.9*exp((uzak-(yaricap+yaya.yaricap))/1000));

}
}
}

```

```

toplamKuvvet=VektorHesabi.toplam(toplamKuvvet ,
liderCekiciKuvveti);

for(int i=0;i<sKModel.kume_sayisi;i++){
for(Yaya yaya:sKModel.kumeler.get(i).elemanlar){
if(yaya!=this && yaya.lider==true &&
sKModel.kumeler.get(i).elemanlar.contains(this)){
if(gorusAlaninda(VektorHesabi.aciHesabi(konum,
yaya.konum), yonaci)){
uzak=DuvvarHesabi.uzunluk(konum,yaya.konum);

Vektor n=VektorHesabi.fark(konum,yaya.konum);

liderKuvveti=VektorHesabi.skalar(n,0.2*exp(1/uzak));
}
}
}

toplamKuvvet=VektorHesabi.toplam(toplamKuvvet ,
liderKuvveti);

Vektor duvarKuvvet=new Vektor(0,0);
Vektor yardimciYonlendiriciKuvvet=new Vektor(0,0);
Duvvar enYakinDuvvar=new Duvvar();
Vektor enYakinNokta=new Vektor();
boolean sacma=false;
double enKisaUzaklik=0;
double enkucuk = infinity;
double uzaklik1;
double vektorArasiaci=0;
Vektor duvarEtkilesimKuvvet=new Vektor();
double delta=0;
Vektor del=new Vektor();
Vektor del2=new Vektor();
double C1=0;
double C2=0;
for(Duvvar duvar: sKModel.duvarlar){

uzaklik1 = DuvvarHesabi.enKisa(konum,
duvar.baslangic , duvar.bitis);

```

```

if(uzaklik1<enkucuk) {
enkucuk = uzaklik1;
enYakinDuvar = duvar;
enYakinNokta = DuvarHesabi.enyakin(konum,
    enYakinDuvar.baslangic , enYakinDuvar.bitis );
duvarKuvvet = VektorHesabi.yon(enYakinNokta ,this.konum);
duvarKuvvet = VektorHesabi.skalar(duvarKuvvet ,
    enYakinDuvar.A*exp((yaricap-DuvarHesabi.enKisa(konum,
        enYakinDuvar.baslangic , enYakinDuvar.bitis ))/enYakinDuvar.B));
}

if(yaricap-(DuvarHesabi.uzunluk(konum,enYakinNokta ))>=0 ){
C2=2400*(yaricap-DuvarHesabi.uzunluk(konum,enYakinNokta ));
C1=1200*(yaricap-DuvarHesabi.uzunluk(konum,enYakinNokta ));
}

delta=VektorHesabi.noktacarpim(hiz ,VektorHesabi.rotate
(VektorHesabi.yon(enYakinNokta ,konum)));
del=VektorHesabi.skalar(VektorHesabi.rotate
(VektorHesabi.yon(enYakinNokta ,konum)),delta );

del2=VektorHesabi.skalar(VektorHesabi.rotate2
(VektorHesabi.yon(enYakinNokta ,konum)),delta );
duvarEtkilesimKuvvet=VektorHesabi.fark
(VektorHesabi.skalar(VektorHesabi.yon(enYakinNokta ,konum),C1),
VektorHesabi.skalar(del ,C2));

if(duvarKuvvet.getY()==0 && duvarKuvvet.getX()<0){
if(hedef.getY()>=konum.getY())
yardimciYonlendiriciKuvvet=VektorHesabi.skalar(VektorHesabi.
rotate2(duvarKuvvet),-1);
else
yardimciYonlendiriciKuvvet=VektorHesabi.rotate2(duvarKuvvet);
}

else if(duvarKuvvet.getY()<0 && duvarKuvvet.getX()==0){
if(konum.getX()>hedef.getX())
yardimciYonlendiriciKuvvet=VektorHesabi.rotate2(duvarKuvvet);
else
yardimciYonlendiriciKuvvet=VektorHesabi.rotate(duvarKuvvet);
}
}

```

```

else if (duvarKuvvet.getY() > 0 && duvarKuvvet.getX() < 0) {
    if (hedef.getY() > konum.getY())
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate3(duvarKuvvet);
    else
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate4(duvarKuvvet);

}

else if (duvarKuvvet.getY() < 0 && duvarKuvvet.getX() < 0) {
    if (hedef.getY() > konum.getY())
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate4(duvarKuvvet);
    else
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate3(duvarKuvvet);

}

else if (duvarKuvvet.getY() < 0 && duvarKuvvet.getX() > 0) {
    if (hedef.getY() > konum.getY())
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate4(duvarKuvvet);
    else
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate3(duvarKuvvet);

}

else if (duvarKuvvet.getY() > 0 && duvarKuvvet.getX() > 0) {
    if (hedef.getY() > konum.getY())
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate3(duvarKuvvet);
    else
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate4(duvarKuvvet);

}

else if (duvarKuvvet.getY() > 0 && duvarKuvvet.getX() == 0) {
    if (konum.getX() > hedef.getX())
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate(duvarKuvvet);
    else
        yardimciYonlendiriciKuvvet = VektorHesabi.rotate2(duvarKuvvet);

}

else if (duvarKuvvet.getY() == 0 && duvarKuvvet.getX() > 0) {
    if (hedef.getY() >= konum.getY())

```

```

yardimciYonlendiriciKuvvet=VektorHesabi.rotate2(duvarKuvvet);
else
yardimciYonlendiriciKuvvet=VektorHesabi.rotate_saatyonu
(duvarKuvvet);

```

```

}

```

```

sacma=true;

```

```

}

```

```

if (sacma) toplamKuvvet=VektorHesabi.toplam(VektorHesabi.toplam
(VektorHesabi.toplam(toplamKuvvet,duvarKuvvet),
yardimciYonlendiriciKuvvet),duvarEtkilesimKuvvet);

```

```

}

```

```

}

```

```

/** GorusAlaninda Fonksiyonu */
boolean gorusAlaninda( double aci, double yon ) {
if (abs(yon-aci)>PI) aci=2*PI-abs(yon-aci);
else aci=abs(yon-aci);
if (aci<gorusAcisi/2) return true;
else return false;
}

```

```

-----
/** Duvar ajani */
// Parameters

```

```

public double A;
public double B;

```

```

// Plain Variables

```

```

public Vektor baslangic;
public Vektor bitis;
private double m;
private double payda;

```

```

// Functions

```

```

void ciz( Vektor baslangic, Vektor bitis ) {

```

```

this.baslangic=baslangic;

```

```
this.bitis=bitis;  
m=( bitis .getY() – baslangic .getY() )/( bitis .getX() –  
baslangic .getX() );  
payda=sqrt (m*m+1);  
  
}  
  
public double uzaklik( Vektor nokta ) {  
  
return abs (m*(nokta .getX() – baslangic .getX() )+ baslangic .getY() –  
nokta .getY() )/ payda;  
  
}
```





EK A.3: Deney Sonuçları Tabloları



Çizelge C.1 : Çıkışın genişliği ve tahliye süresi ilişkisi

Çıkış Genişliği	Yaya Sayısı	Tahliye Süresi
1	25	29.4496
	50	35.4525
	100	46.0640
	200	63.4024
	300	79.0739
	400	94.1932
1.5	25	28.6314
	50	31.0265
	100	38.5840
	200	52.6837
	300	66.5195
	400	79.9185
2	25	28.7063
	50	30.1059
	100	34.2698
	200	46.5181
	300	58.8858
	400	71.1543
2.5	25	28.6974
	50	30.0379
	100	32.4719
	200	42.5055
	300	53.7002
	400	65.1165
3	25	28.8986
	50	30.1046
	100	31.7364
	200	40.0370
	300	50.1979
	400	61.0252
3.5	25	29.0666
	50	30.1483
	100	31.6142
	200	38.3242
	300	47.7960
	400	57.7347
4	25	28.9583
	50	30.4116
	100	31.6639
	200	37.1706
	300	45.8262
	400	55.4296

Çizelge C.2 : Çıkış sayısı ve tahliye süresi ilişkisi

Çıkış Sayısı	Yaya Sayısı	Tahliye Süresi
1	25	28.6498
	50	30.112
	100	34.2698
	200	46.5181
	300	58.8858
	400	71.1543
2	25	17.7392
	50	19.0159
	100	23.7073
	200	34.3104
	300	44.9715
	400	55.5118
3	25	17.1381
	50	18.4691
	100	20.5186
	200	28.2548
	300	38.8240
	400	45.1845
4	25	12.6268
	50	13.6005
	100	16.2897
	200	23.2655
	300	30.6009
	400	37.8588

Çizelge C.3 : Engel konumu ve tahliye süresi ilişkisi

Engel Çıkış Arası Uzaklık	Yaya Sayısı	Tahliye Süresi
2 m	25	29.8164
	50	31.8056
	100	35.6900
	200	44.1711
	300	53.3387
	400	62.8607
5 m	25	29.3444
	50	30.9198
	100	34.7818
	200	44.8805
	300	54.4433
	400	64.0321
10 m	25	29.1955
	50	30.7396
	100	34.4030
	200	45.9358
	300	57.4905
	400	68.5619

Çizelge C.4 : Grup Sayısı ve tahliye süresi ilişkisi

Yaya Sayısı	Grup Sayısı	Tahliye Süresi
200	2	65.37193
	3	64.98291
	4	61.35613
	5	58.0440
	6	57.3599
	7	57.97647
	8	63.62295
	9	65.48780
	10	67.59176

ÖZGEÇMİŞ

Ad Soyad:Yakup TURGUT

Doğum Tarihi ve Yeri: 23.04.1993/Buca

E-Posta:turgut16@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans:** 2015, Yıldız Teknik Üniversitesi, Makine Fakültesi, Endüstri Mühendisliği Bölümü