



**DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ VE
KARINCA KOLONİSİ ALGORİTMASI İLE YOL PLANLAMA
PROBLEMLERİNİN OPTİMAL ÇÖZÜMÜ**

Fatih SUVAYDAN

ELEKTRİK EĞİTİMİ ANABİLİM DALI

**MAYIS 2011
DÜZCE**

**DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ
VE KARINCA KOLONİSİ ALGORİTMASI İLE YOL
PLANLAMA PROBLEMLERİNİN OPTİMAL ÇÖZÜMÜ**

FATİH SUVAYDAN

ELEKTRİK EĞİTİMİ ANABİLİM DALI

**MAYIS 2011
DÜZCE**

Fatih SUVAYDAN tarafından hazırlanan MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ VE KARINCA KOLONİSİ ALGORİTMASI İLE YOL PLANLAMA PROBLEMLERİNİN OPTİMAL ÇÖZÜMÜ adlı bu tezin olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Pakize ERDOĞMUŞ

Tez Danışmanı, Elektrik A.B.D.

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Elektrik Anabilim Dalında Yüksek Lisans olarak kabul edilmiştir.

Yrd. Doç. Dr. Pakize ERDOĞMUŞ

Elektrik A.B.D., Düzce Üniversitesi

Yrd. Doç. Dr. Ali ÖZTÜRK

Elektrik A.B.D., Düzce Üniversitesi

Yrd. Doç. Dr. Uğur GÜVENÇ

Elektrik A.B.D., Düzce Üniversitesi

Tarih : 25 / 05 / 2011

Bu tez ile Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Refik KARAGÜL

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Fatih SUVAYDAN

ÖNSÖZ

Tez çalışmam sırasında değerli fikirlerini ve zamanını benden esirgemeyen tez danışmanım olan Sayın Yrd. Doç. Dr. Pakize ERDOĞMUŞ' a yardımlarından dolayı teşekkür ederim.

Bugüne kadar üzerimde fazlaca emekleri olan ve vermiş oldukları bilgilerle alanımda gelişmemi sağlayan bütün Elektrik Eğitimi hocalarıma teşekkür ederim.

Tez çalışmam sırasında yardımlarını esirgemeyen Elektrik Eğitimi ile Elektronik ve Bilgisayar Eğitimi araştırma görevlilerine göstermiş oldukları sabırdan dolayı teşekkür ederim.

Hayatımın her anında olduğu gibi eğitim-öğretim hayatımda da hep yanımda olan, yardımlarını benden esirgemeyen anneme, babama ve kardeşlerime çok teşekkür ederim.

Ayrıca tez çalışmam sırasında bana hep moral olan Tuba YILMAZ' a, biricik yeğenim Cemre'ye ve tüm arkadaşlarıma teşekkür etmeyi bir borç bilirim.

Mayıs 2011

Fatih SUVAYDAN

İÇİNDEKİLER

Sayfa

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ	v
ÇİZELGE LİSTESİ	vii
SEMBOL LİSTESİ	viii
KISALTMALAR LİSTESİ	ix
ÖZ	x
ABSTRACT	xii
1. GİRİŞ	1
1.1. TEZ KONUSUNUN TANIMI VE ÖNEMİ	1
1.2. TEZİN AMACI	1
1.3. TEZİN İÇERİĞİ	2
2. LİTERATÜR İNCELENMESİ	3
3. MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ	13
3.1. GİRİŞ	13
3.2. GLOBAL ve LOKAL YOL PLANLAMA	16
3.3. YOL PLANLAMA UZAYI	17
3.4. YOL PLANLAMA YÖNTEM ve TEKNİKLERİ	20
3.4.1. Klasik Yol Planlama Teknikleri	21
3.4.1.1. Hücre Ayrıştırma Tekniği	21
3.4.1.2. Potansiyel Alanlar Tekniği	25
3.4.1.3. Matematiksel Programlama Tekniği	27
3.4.1.4. Yol Haritaları Tekniği	28

3.4.2. Olasılığa Dayalı Yol Planlama Teknikleri	33
3.4.2.1. Olasılığa Dayalı Yol Haritaları Tekniği	33
3.4.2.2. Rassal Bağlantıları Hızlı Keşif Tekniği	35
3.4.2.3. Seviye Kümeleri Tekniği	36
3.4.2.4. Dilbilimsel Geometri Tekniği	38
3.4.3. Sezgisel Yol Planlama Teknikleri	40
4. KARINCA KOLONİSİ OPTİMİZASYONU	43
4.1. GERÇEK KARINCA DAVRANIŞLARI	44
4.2. KARINCA KOLONİSİ ALGORİTMASI	47
4.2.1. Karınca Sistemi	48
4.2.1.1. Algoritmanın Başlaması ve Değerlerin Atanması	49
4.2.1.2. Karıncaların Turunun (Hareketinin) Oluşturulması	52
4.2.1.3. Feromen İzlerinin Güncelleştirilmesi	53
4.3. KARINCA KOLONİSİ ALGORİTMASI VERSİYONLARI	55
4.3.1. Elit (Elitist) Karınca Sistemi	55
4.3.2. Derece (Rank) Temelli Karınca Sistemi	56
4.3.3. Maksimum-Minimum Karınca Sistemi	56
4.4. SAYISAL PROBLEMLER İÇİN KARINCA ALGORİTMALARI	57
4.5. MOBİL ROBOTLARIN YOL PLANLAMA PROBLEMLERİ	
İÇİN KARINCA ALGORİTMALARI	61
5. MATERYAL VE YÖNTEM	63
5.1. MATERYAL	63
5.1.1. Matlab ve GUI	63
5.1.2. Tez Çalışması Arayüzünün GUI ile Tasarımı	65
5.2. YÖNTEM	66
5.2.1. Engelden Sakınma Yöntemi	66
5.2.2. Önerilen Karınca Kolonisi Algoritması Yöntemi	69
5.2.2.1. Algoritmanın İşleyişi	71
6. BULGULAR	74
6.1. TEST PROBLEMLERİ ve SONUÇLARI	75
6.1.1. Çevre - 1	76

6.1.2. Çevre - 2	80
6.1.3. Çevre - 3	84
6.1.4. Çevre - 4	88
7. TARTIŞMA VE SONUÇ	92
KAYNAKLAR	95
ÖZGEÇMİŞ	110

ŞEKİL LİSTESİ

Sayfa

Şekil 3.1	: Basit bir yol planlama örneğine ait kriterler.....	14
Şekil 3.2	: İki boyutlu çalışma uzayı.....	18
Şekil 3.3	: İki boyutlu konfigürasyon uzayı	18
Şekil 3.4	: Yol planlama teknikleri.....	20
Şekil 3.5	: Klasik yol planlama teknikleri.....	21
Şekil 3.6	: Yol planlama için dördün ağaç tekniği	23
Şekil 3.7	: Yol planlama için çerçeveli dördün ağaç tekniği.....	24
Şekil 3.8	: Yol planlama için tam hücre ayrıştırması örneği	25
Şekil 3.9.a	: Birinci engel için oluşturulan potansiyel alan	26
Şekil 3.9.b	: İkinci engel için oluşturulan potansiyel alan	26
Şekil 3.10.a	: Hedef için oluşturulan potansiyel alan.....	27
Şekil 3.10.b	: Toplam potansiyel alan	27
Şekil 3.11	: Yol planlama için görünürlük grafiği örneği.....	29
Şekil 3.12	: Yol planlaması için voronoi diyagramı örneği	30
Şekil 3.13	: Yol planlama için silüet eğrisi örneği	31
Şekil 3.14	: Yol planlama için alt hedef tekniği örneği.....	32
Şekil 3.15	: Olasılığa dayalı yol planlama teknikleri.....	33
Şekil 3.16	: Yol planlama için olasılığa dayalı yol haritası örneği	35
Şekil 3.17	: Yol planlaması için rassal bağlantıları hızlı keşif örneği	36
Şekil 3.18.a	: A ile B noktaları arasının bir çizgi ile gösterilmesi.....	37
Şekil 3.18.b	: A noktası etrafında yüzey yayılması.....	37
Şekil 3.18.c	: A noktasından B noktasına yol bulmak için geri dönüşün izlenmesi.....	37
Şekil 3.19.a	: A bölgesinin karlı, B bölgesinin kuru olduğunu gösterilmesi.....	38
Şekil 3.19.b	: A noktası etrafında yüzey yayılması.....	38
Şekil 3.19.c	: A noktasından B noktasına yol bulmak için geri dönüşün izlenmesi.....	38
Şekil 3.20	: Bir uçağın dilbilimsel geometri tekniği ile yol (rota) planlaması	39
Şekil 3.21	: Sezgisel yol planlama teknikleri	40
Şekil 3.22	: Yol planlama tekniklerinin yıllara göre dağılımı	41
Şekil 4.1	: Yuva ile gıda arasındaki en kısa yol	45
Şekil 4.2	: Karıncaların bir engelle karşılaştığında en kısa yolun bozulması	45
Şekil 4.3	: Karıncaların engelle karşılaşması sonucundaki durumları.....	46
Şekil 4.4	: Karıncaların belli bir süre sonunda kısa yolu bulmaları	46
Şekil 4.5	: Karınca sistemi algoritması işleyişi	48
Şekil 4.6	: α ve β değerleri için seçim aralıkları grafiği.....	51
Şekil 4.7	: Bir karınca tarafından oluşturulan çözüm	59
Şekil 5.1	: GUI çalışma arayüzü.....	64
Şekil 5.2	: Tez çalışması için tasarlanan arayüz.....	65
Şekil 5.3	: Engel orijininin doğruya olan uzaklığı.....	67
Şekil 5.4	: Algoritmanın akış diyagramı	69
Şekil 6.1	: Çevre-1 için elde edilen grafiksel görüntü	76

Şekil 6.2	: Çevre-1 için lokal planlamanın optimal değerime göre çizilen yollar.....	77
Şekil 6.3	: Çevre-1 için global planlamanın optimal değerime göre çizilen yollar.....	78
Şekil 6.4	: Çevre-2 için elde edilen grafiksel görüntü	80
Şekil 6.5	: Çevre-2 için lokal planlamanın optimal değerime göre çizilen yollar.....	81
Şekil 6.6	: Çevre-2 için global planlamanın optimal değerime göre çizilen yollar.....	82
Şekil 6.7	: Çevre-3 için elde edilen grafiksel görüntü	84
Şekil 6.8	: Çevre-3 için lokal planlamanın optimal değerime göre çizilen yollar.....	85
Şekil 6.9	: Çevre-3 için global planlamanın optimal değerime göre çizilen yollar.....	86
Şekil 6.10	: Çevre-4 için elde edilen grafiksel görüntü	88
Şekil 6.11	: Çevre-4 için lokal planlamanın optimal değerime göre çizilen yollar.....	89
Şekil 6.12	: Çevre-4 için global planlamanın optimal değerime göre çizilen yollar.....	90

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1	: Global ve lokal planlamasının karşılaştırılması	17
Çizelge 3.2	: Mobil robotların yol planlama problemlerinin çözümü için kullanılan sezgisel tekniklerin yıllara göre kullanım oranları.....	42
Çizelge 4.1	: Ayrık optimizasyon problemleri için ACO ile ilgili yapılan çalışmalar	58
Çizelge 6.1	: Test için oluşturulan çevrelere ait parametre değerleri	75
Çizelge 6.2	: Çevre-1'e ait KKA parametre değerleri ve sonuçlar	76
Çizelge 6.3	: Çevre-1'e ait çalışma süreleri	77
Çizelge 6.4	: Çevre-2'e ait KKA parametre değerleri ve sonuçlar	80
Çizelge 6.5	: Çevre-2'e ait çalışma süreleri	81
Çizelge 6.6	: Çevre-3'e ait KKA parametre değerleri ve sonuçlar	84
Çizelge 6.7	: Çevre-3'e ait çalışma süreleri	85
Çizelge 6.8	: Çevre-4'e ait KKA parametre değerleri ve sonuçlar	88
Çizelge 6.9	: Çevre-4'e ait çalışma süreleri	89

SEMBOL LİSTESİ

x	: Robotun pozisyonun koordinatı
x_{hedef}	: Hedef pozisyonun koordinatı
ρ_0	: Engelin etkili olduğu mesafe
$k_{çek} - k_{it}$: Pozitif ölçeklendirme faktörleri
N	: Şehirler (Noktalar) kümesi
A	: Noktaları birbirine bağlayan arklar
G	: Ağırlıklaştırılmış Graf
b_i	: i . şehirdeki karınca sayısı
m	: Toplam karınca sayısı
n	: Şehir sayısı
τ_{ij}	: i ve j noktaları arasında depolanan feromen madde miktarı
τ_0	: İlk (başlangıç) feromen madde miktarı
C^{nn}	: En yakın komşuluk sezgisiyle oluşturulan tur uzunluğu
ρ	: Buharlaştırma katsayısı
η_{ij}	: i ve j noktaları arasında seçilebilirlik parametresi
d_{ij}	: i ve j noktaları arasında mesafe
α	: Feromen miktarının önemini belirleyen parametre
β	: Yol uzunluklarının ve sezgiselliğin önemini belirleyen parametre
p_{ij}^k	: k . karınca için i şehirden j şehrine geçme olasılığı
N_i^k	: k . karıncanın henüz ziyaret etmediği şehirlerin seti
M^k	: Karıncaların ziyaret ettikleri şehirleri içeren hafıza
$\Delta\tau_{ij}$: Birim zamanda (i,j) hattına bırakılan feromen madde miktarı
L^k	: k . karıncanın toplam tur uzunluğu
Q	: Feromen güncellemesi için sabit bir değer
$\Delta\tau_{ij}^k$: k . karınca tarafından (i,j) hattına bırakılan feromen madde miktarı
t	: Zaman sabiti
$L_{en\ iyi}$: Geçerli turda bulunan en iyi tur uzunluğu
e	: Elit karınca sistemi için algoritmanın başlangıcından o ana kadar bulunan en iyi tura verilen önemi tanımlayan parametredir
$\Delta\tau_{ij}^{bs}$: Elit karınca sistemi için Algoritmanın başlangıcından o ana kadarki en iyi tur üzerindeki feroman madde miktarı
L_{bs}	: Elit karınca sistemi için en iyi tur uzunluğu
τ_{max}	: Üst feromen sınırı
τ_{min}	: Alt feromen sınırı.
a	: Maksimum-Minimum karınca sistemi için sabir değer
$L^{en\ iyi}$: Maksimum-Minimum karınca sistemi için en iyi tur uzunluğu

KISALTMALAR LİSTESİ

GPS	: Global positioning system (Global konumlandırma sistemi)
Wspace	: Çalışma uzayı
Cspace	: Konfigürasyon uzayı
Cfree	: Serbest uzay
KKO	: Karınca kolonisi optimizasyonu
ACO	: Ant colony optimization
KKA	: Karınca kolonisi algoritması
GSP	: Gezgin satıcı problemi
CACO	: Continuous ant colony optimization
TACO	: Touring ant colony optimization
GACO	: Grid ant colony optimization

MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ VE KARINCA KOLONİSİ ALGORİTMASI İLE YOL PLANLAMA PROBLEMLERİNİN OPTİMAL ÇÖZÜMÜ

(Yüksek Lisans Tezi)

Fatih SUVAYDAN

DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2011

ÖZ

Robotik sistemlerde, mobil robotların engellerden sakınması ve çarpışma olmadan yol planlaması zor bir problemdir. Buna bağlı olarak mobil robotların belirli bir alan içerisinde bir noktadan başka bir noktaya, alan içerisinde bulunan engeller ile çarpışma olmaksızın optimum yolun bulunması yol planlamasının temel hedefidir. Bu hedef doğrultusunda mobil robotlar için yol planlaması yapabilmek için literatürde birçok yol planlama tekniği bulunmaktadır. Özellikle son yıllarda klasik yol planlama tekniklerinin yetersizliklerini ve sorunlarını ortadan kaldırmak için sezgisel yol planlama teknikleri kullanılmaya başlanmıştır. Bu tez çalışmasında da, mobil robotların yol planlaması problemini çözme işlemi için sezgisel yol planlama tekniklerinden biri olan karınca kolonisi algoritması kullanılmış olup, buna göre bir alan içerisinde engellere çarpma olmaksızın optimum yolun bulunması amaçlanmıştır. Bunun için görsel bir simülasyon programı hazırlanmıştır. Bu simülasyon program ile başlangıç-bitiş koordinatları, engellerin sayısı ve boyutları ile karınca kolonisi algoritması parametre değerleri dışarıdan girilmektedir. Bu sayede parametre değerleri istenildiğinde değiştirilerek buna göre sonuçları sağlıklı bir şekilde değerlendirmesi sağlanmıştır. Hazırlanan simülasyon programı ile 4 farklı çevre oluşturulup, sonuçlar önerilen algoritmanın lokal feromon güncellemesine ve global feromon güncellemesine bağlı

olarak elde edilmiştir. Bununla birlikte karşılaştırmalar amacıyla yolların ceza fonksiyonu uygulamadan normal uzunlukları hesaplanmıştır ve bağıl hata değerleri bulunmuştur. Parametre değişikliklerine bağlı olarak program farklı engel boyutu ve sayısında defalarca çalıştırılarak algoritmanın performansı ve etkinliği değerlendirilmiştir.

Bilim Kodu : 400258

Anahtar Kelimeler : Yol Planlama, Karınca Koloni Algoritması, Optimal Yol

Sayfa Adedi : 127

Tez Yöneticisi : Yrd. Doç. Dr. Pakize ERDOĞMUŞ

**PATH PLANNING PROBLEM FOR MOBILE ROBOTS AND OPTIMAL SOLUTION
OF PATH PLANNING PROBLEMS WITH ANT COLONY OPTIMIZATION
ALGORITHM**

(M.Sc. Thesis)

Fatih SUVAYDAN

**DÜZCE UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

May 2011

ABSTRACT

It is a difficult problem to mobile robots obstacle avoidance and to path planning without clash for robotic systems. For that reason the main aim of path planning is to find optimal way without clash while it goes from one point to another point in a certain area. With that aim, we have lots of technics at literature for path planning to make a path plan for mobile robots. Especially in recent years, intuitive path planning technics are started to be used because of the insufficiency of classical path planning technics and eliminating the problem. This thesis includes Ant Colony Algorithm which is one of the ways of optimal path plans to avoidance the obstacles of mobile robots path planning. Nevertheless the aim of thesis is to find an optimal way without clashing the barriers in a certain area. Therefore a visual simulation programme is prepared. With this simulation programme start-finish coordinates, the number of obstacles and the size of them, ant colony algorithm and parameters accounts are added. Thanks to this feature, if one can change the parametric accounts and obtain safe results. There are four different circuits in this simulation programme and the results are obtained from proposed algorithm local pheromene update and global pheromene update. Notwithstanding normal lengths of the paths are calculated without penalty function in order to compare

and was calculated relative error values. Depending on parameter changes, the performances and the efficiency of algorithm is evaluated by run of programme with a variety of obstacle dimensions and numbers.

Science Code : 400258

Keywords : Path Planning, Ant Colony Optimization Algorithm, Optimal Path

Page Number : 127

Adviser : Yrd. Doç. Dr. Pakize ERDOĞMUŞ

1. GİRİŞ

1.1. TEZ KONUSUNUN TANIMI VE ÖNEMİ

Yol planlama problemleri işlemsel olarak çok karmaşık problemler olarak ifade edilebilir. Bunun nedeni ise yol planlama işlemi yapılacak alan içerisinde birçok alternatif yol seçeneğinin bulunması ve bu yollar arasında engellere çarpmaksızın optimum yolun bulmaya çalışılmasının amaçlanmasıdır. Yol planlamasında ilk hedef planlanan yolun alan içerisinde bulunan engeller ile temas probleminin halledilmesidir. Bunun için literatürde birçok engellerden kaçma veya engellerden sakınma problemlerini çözecek farklı yöntem ve teknikler vardır. Bu yöntemler bazı araştırmacılar tarafından yol planlama yöntemleri olarak da adlandırılmaktadır. Bu yol planlama yöntem ve teknikleri klasik yol planlama, olasılığa dayalı yol planlama ve sezgisel yol planlama yöntemleri olarak ifade edilebilir. Özellikle son yıllarda yol planlama problemlerinin çözümünde sezgisel yol planlama tekniklerinin daha fazla kullanıldığı görülmüştür. Bunu sebebi karmaşık işlemlerden kaçmak ve zamanın daha verimli hale getirilerek daha sağlıklı sonuçlar elde edilmesidir.

1.2. TEZİN AMACI

Bu tezin amacı mobil robotların yol planlama problemlerinin çözümü hakkında literatür taraması yapıp, yol planlaması hakkında bilgi birikimi sağlamak ve sezgisel bir teknik olan Karınca Kolonisi Algoritması yardımıyla bir noktadan hedeflenen bir noktaya engellere çarpmaksızın optimum kriterlere uygun yol planlamasının yapılmasıdır. Buna paralel olarak Karınca Kolonisi Algoritmasının lokal ve global feromon güncellemesine göre sonuçlar elde edilmeye çalışılmış bir simülasyon programı elde edilmiştir. Bu simülasyon programı sayesinde gereken değerler girilip değerlendirilmeler yapılarak en sağlıklı sonucun bulunması hedeflenmiştir.

1.3. TEZİN İÇERİĞİ

Tezin ikinci bölümünde, karınca kolonisi algoritması kullanılarak mobil robotlar için yapılan yol planlaması ile ilgili kapsamlı literatür taraması yapılarak elde edilen çalışmaların bir literatür özeti verilmiştir.

Tezin üçüncü bölümünde, ilk olarak yol planlama problemi tanıtılmış olup, daha sonra yol planlama yaklaşımları ve planlama yapılan alanlar hakkında bilgi verilmiştir. Bununla birlikte mobil robotların yol planlama problemleri için önerilen ve geliştirilen farklı teknikler hakkında detaylı bir şekilde bilgi verilmiştir.

Tezin dördüncü bölümünde, yol planlama problemi için kullanılan karınca kolonisi algoritması ile ilgili olarak gerçek karınca davranışlarının açıklaması ile başlanarak, karınca kolonisi algoritmasının gelişimi, matematiksel formülasyonu, bazı karınca kolonisi sistemleri ve bunların farklı problemlerine uygulama şekilleri anlatılmıştır.

Tezin beşinci bölümünde, ilk olarak Matlab GUI hakkında temel bilgi verilmiş olup, oluşturmuş olduğumuz simülasyon programının tanıtımı yapılarak hedeflenen yol planlama algoritması açıklanmıştır. Yine yol planlama probleminin çözümü için kullanmış olduğumuz yöntemler olan engel sakınma yöntemi ve geliştirdiğimiz karınca kolonisi algoritması ayrıntılı bir şekilde anlatılmıştır.

Tezin altıncı bölümünde, tasarlanmış olduğumuz simülasyon programı kullanılarak farklı parametre değerleri girilerek elde edilmiş bulgulara yer verilmiştir. Elde edilen bulgular üzerinden sonuçların karşılaştırılması yapılarak elde edilen sonuçlar sayısal ve grafiksel olarak gösterilmiştir.

Tezin yedinci bölümünde, elde edilen sonuçlar ile değerlendirme yapılarak, yapılacak çalışmalar için önerilere yer verilmiştir.

2. LİTERATÜR İNCELENMESİ

Mobil robotlar için yol veya hareket planlama problemleri ile ilgili çalışmalar 1960'lı yılların ortasında başlamış olup 1980'li ivme kazanarak, o yıllardan beri önemli bir araştırma konusu haline gelmiştir. Bu bölümde de, daha önceki yıllardan şimdiye kadar karınca kolonisi algoritması kullanılarak mobil robotlar için yapılan yol planlaması çalışmalarının bir literatür incelemesi yapılarak, özeti sunulmuştur. Karıncaların doğal yaşamlarından yola çıkarak yapılan ilk çalışma Deneubourg ve diğ. (1994) tarafından yapılmıştır. Bu çalışma, ulaşım sistemleri için uyarlamışlardır. Yaptıkları çalışmada karıncaların doğal yaşamlarının özellikle pozitif geri besleme davranışını örnek almışlardır ve böylece ulaşım ağları oluşturmayı hedeflemişlerdir.

Sauter ve diğ. (2002), mobil robotların engellerden kaçarak yol planlaması için KKA ile genetik algoritmayı aynı algorithmada kullanmıştır. Dinamik çevrelerde kısa yolları bulmak için, karınca kolonisi algoritmasında kullanıyor olan feromen maddesini otomatik olarak ayarlamak için genetik algoritmayı kullanmıştır ve buna sentetik feromen adını vermişlerdir. Simülasyon olarak yapılan çalışmaların sonucunda bu algoritmanın çalışmasından kısa bir süre sonra uygun feromen değeri bulunarak iyi sonuçlar verdiği görülmüştür.

Jin ve diğ. (2002), özgür uçuş sahalarında mobil robotun yol planlaması için bir karınca kolonisi algoritması geliştirmişlerdir. Oluşturdukları simülasyon programları ile bu sahalara farklı engeller (dağ,tepe,v.b.) ekleyerek sonuçların verimliliğini göstermişlerdir.

Fan ve diğ. (2003), yaptıkları çalışmada ilk kez mobil robotların yol planlamasında kullanılacak olan kısıt koşulları içeren sürekli bir optimizasyon problemi için yoğunlaştırılmış bir KKA geliştirmişlerdir. Bu algoritma ile önemli katsayılar arasında karşılaştırmalar yaparak, dinamik olarak olası yolları oluşturarak, karıncalar için geçiş

olasılıklarının seçimini yapmışlardır. Algoritmanın etkinliğini simülasyon sonuçları ile göstermişlerdir.

Hsiao ve diğ. (2004), çalışmalarında engel olmayan bir alanda optimal yolu planlamak için karınca kolonisi algoritmasını kullanmışlardır. Yapmış oldukları algoritmayı C++ dilinde simülasyon programı olarak oluşturmuşlardır. Bu program sayesinde yollar rastgele olarak oluşturulmuştur ve bu oluşturulan yollar arasından optimal yol planlaması yapılmıştır.

Fan ve diğ. (2004), mobil robotların yol planlaması için kullanılan tekniklerin var olan bazı zorluklarını ortadan kaldırmak için KKA geliştirmişlerdir. Bu algoritmayı, bazı kısıt koşulları içeren sürekli bir optimizasyon problemini çözmek için uyarlamışlardır. Algoritmayı farklı kompleks çevrelerde bir simülasyon programı ile uygulamışlardır ve sonuçların verimliliğini göstermişlerdir.

Garro ve diğ. (2005), mobil robotun hedefe varması için izleyeceği bir yolun optimize edilmesi için karınca kolonisinin değişik bir biçimini önermişlerdir. Bu algorithmada, genetik algoritma kullanılarak karınca kolonisi algoritmasının bazı parametrelerini ayarlanması sağlanmıştır. Yaptıkları bu çalışmada, oluşturdukları genetik algoritma-karınca kolonisi ile gerçek çevrelerde oluşan durgunlaşmaları klasik karınca kolonisi ile karşılaştırarak daha iyi sonuçlar elde ettiklerini görmüşlerdir.

Wen ve diğ. (2005), karınca kolonisi algoritmasını kullanarak denizden yüksekliğe veya alçaklığa göre hava aracı için bir yol planlaması algoritması geliştirmişlerdir. Karınca kolonisinin bazı parametreleri ile oynayarak daha iyi bir noktada birleşme daha iyi bir yol bulmaya çalışmışlardır ve algoritmanın verimliliğini simülasyon programı oluşturarak değerlendirmişlerdir.

Wang ve diğ. (2005), yol planlama problemleri için yeni bir KKA geliştirilerek bu metodu Java programı ile simule etmişlerdir. Önerdikleri metodun sonuçlarını farklı çevrelerde deneyip dijkstra algoritması ile kıyaslayarak algoritmanın etkinliğini ve verimliliğini görmüşlerdir.

Mohamad ve diğ. (2005), olasılığa dayalı yol harita yol haritası ile karınca kolonisi algoritmasını kullanarak yol planlama problemi için yeni bir yaklaşım önermişlerdir. Olasılığa dayalı yol harita ile yol planlaması yapılacak alan oluşturularak bu alan içersinde optimal yolu bulmak için karınca kolonisi algoritması kullanılmıştır. Bu önerilen algoritma ile yapılan deneylerin sonucunda, uygun bir çalışma zamanında başlangıç noktası ile hedef noktası arasında istenenleri bulmakta daha yetenekli ve etkili olduğu görülmüştür.

Liu ve diğ. (2005), mobil robotların iki boyutlu çevrelerde yol planlamasını yapmak için karınca kolonisi algoritmasını kullanarak yeni bir algoritma önermişlerdir. Önerdikleri algoritma ile yerel optimizasyonun sakıncalarını çözmeyi hedeflenmiştir. Yaptıkların çalışmanın sonucunda bu çevrelerde optimum değeri aramayı hızlandırdıklarını görmüşlerdir.

Mei ve diğ. (2006), yapmış oldukları çalışmada global yol planlaması için KKA ve yerel yol planlaması için de potansiyel alanlar tekniğine ait algoritmaları birleştirerek yeni bir algoritma elde etmişlerdir. Çalışmada, karınca kolonisi algoritmasında kullanılan feromen maddesi parametresini yapay potansiyel alanda yerel minimum değere takılmaktan kurtulmak için kullanmışlardır. Yaptıkları simülasyonun sonuçlarına göre bu yeni algoritmanın gerçek zamanlı talepleri karşılamada yeterli olduklarını görmüşlerdir ve bu algoritmanın sonuçlarını genetik algoritma ile yapılan çalışma ile karşılaştırmışlardır.

Liu ve diğ. (2006), mobil robotların yol planlaması için kompleks ve statik çevrelerde karınca kolonisi algoritmasını ve engel sakınması ile yerel bir arama algoritması sunmuşlardır. Önerdikleri algoritmada klasik karınca kolonisi algoritmasına özel bazı görevler eklemişlerdir. Yolları arama sırasında bir çarpışma ile karşılaştıklarında, bu yerler için bir tablo oluşturmuşlardır ve mobil robotun bu engellerden kaçınması için ceza fonksiyonu uygulayarak o bölgeleri etkisiz hale getirmeye çalışmışlardır. Algoritmayı simulasyon programı ile uyarlayarak, elde edilen sonuçlara göre algoritmanın etkili ve verimli olduğunu göstermişlerdir.

Lu ve Feng (2006), nümerik bir potansiyel alan tekniđi ile karınca kolonisi algoritmasını birlikte kullanarak yeni bir metot sunmuşlardır. Bu yeni metotta sayısal potansiyel alanı mobil robotun çalıştığı çevreyi oluşturması için kullanılmıştır. Çalışmada engellerin hareketi lokal potansiyel değerler değiştirilmesi ile göz önünde tutulmaktadır. Daha sonra yol arama ve optimal yolu bulmak için de karınca kolonisi algoritması kullanılmıştır. Algoritmanın uygulanması ile ilgili yapılan denemelerin sonucu algoritmanın etkili olduğunu göstermiştir.

Mohamad ve diğ. (2006), 3 boyutlu robot yol planlamasına karınca kolonisi algoritmasını uygulamışlardır. Bu çalışmada, önceki çalışmalarında kullanmış oldukları olasılığa dayalı yol haritası tekniđini tekrar kullanarak yol planlaması alanı oluşturmuşlardır. Yine bu çalışmanın daha etkili ve yetenekli olduğunu görmüşlerdir.

Ye ve diğ. (2006), mobil robotların engellerden kaçınması ile yol planlaması için kendinden uyarlamalı bir KKA önermişlerdir. Algoritmayı Matlab-Toolbox'da uygulayarak klasik karınca algoritması üzerinde bazı uyarlamalar yaparak daha iyi yolların bulunmasını gerçekleştirmişlerdir. Yapılan çalışmalar sonucunda algoritmanın klasik karınca algoritmasına göre daha etkili ve uygun olduğu görülmüştür.

Tan ve diğ. (2007), yapmış oldukları çalışmada gerçek zamanlı mobil robotlar ile global optimal yolun oluşması için karınca kolonisi algoritmasını önermişlerdir. Bu çalışma üç adımdan oluşmaktadır. Birinci adımda, mobil robotun özgür boşluk modelini kurması için MAKLINK graf teorisinden yararlanılmıştır. İkinci adımda, çarpışma olmayan alt bir optimal yolu bulmak için dijkstra algoritması kullanılmıştır. Üçüncü adımda ise alt optimal yolun yerine global olarak optimal yolun oluşması için karınca kolonisi algoritması kullanılmıştır. Bu algoritma yapılan simülasyon programı ile daha önce genetik algoritma ile yapılmış çalışma karşılaştırılmıştır ve bu algoritmanın bir noktada dinamik olarak birleşme açısından, sonuçlar ve çalışma hızı bakımından daha iyi sonuçlar verdiği görülmüştür.

Shi ve diğ. (2007), yol planlaması problemini çözmek için KKA ile parçacık sürü algoritmasını kullanarak yeni bir algoritma geliştirmişlerdir. Parçacık sürü algoritması, karınca kolonisi algoritmasının parametrelerini en verimli şekilde kullanmak için

uygulanmıştır. Bunu Matlab’da bir simülasyon programı hazırlayarak en iyi olduklarını düşündükleri parametre değerlerini engel bulunan farklı alanlarda uygulamışlardır ve sonuçların verimliliğini göstermişlerdir.

Chen ve diğ. (2007), aralıklı-seyrek bir graf üzerinde uygun bir zamanda optimal yolun bulan bir KKA önermişlerdir. Bu algoritmayı, gerçek insansız hava araçlarının yolunu planlamak için düşünmüşlerdir. Algoritma VC++ dilinde simülasyon olarak uygulayarak, çeşitli karmaşıklığa sahip test problemlerinde riskleri minimize ederek kalite çözüm yolları üretmiştir ve bu sayede algoritmanın verimli olduğunu göstermişlerdir.

Vien ve diğ. (2007), yaptıkları çalışmada mobil robotlar için engelden kaçınarak yol planlaması için Ant-Q algoritmasını kullanmışlardır. Çalışmalarında, güncellemenin takviye hassasiyetini araştırmışlardır ve bunları deneysel olarak oluşturdukları simülasyon programında uygulamışlardır. Bu uygulama sonucunda elde ettikleri sonuçları genetik algoritma ve klasik karınca kolonisi algoritması ile kıyaslamışlardır ve sonucunda algoritmanın daha iyi bir noktada birleşme sağladığı görülmüştür.

Lee ve diğ. (2008), yol planlaması için potansiyel alanlar tekniğini de kullanarak yeni bir KKA geliştirmişlerdir. Matlab’da yapmış oldukları simülasyon programı ile farklı çevrelerde ve geometrik şekillerde kontrol parametreleri ile oynayıp kompleks haritalara uygulayarak optimal çözümlere ulaşmaya çalışmışlardır. Sonuçları yine kendi oluşturdukları klasik bir karınca kolonisi algoritmasının sonuçları ile karşılaştırmışlardır ve yeni oluşturdukları algoritmanın daha iyi sonuçlar verdiğini görmüşlerdir.

Yu ve diğ. (2008), mobil robotların dinamik çevrelerde yol planlaması için bir KKA geliştirmişlerdir ve bu algoritmayı C++ dilinde simülasyon haline getirmişlerdir. Buna göre bir robotu oluşturmuş oldukları farklı dinamik çevrelerde gezdirerek yol planlaması gerçekleştirmişlerdir. Sonuçlara göre bu algoritmanın robotun dinamik çevrelerde yol planlaması için çok iyi sonuçlar verdiğini belirtmişlerdir.

Chen ve Yuan (2008), yol planlama problemi için alışılmışın dışında bir KKA sunmuşlardır. Sundukları çalışmada, feromen sınırlaması ve feromen buharlaşma

katsayıları ile oynayarak bir metot geliştirmişlerdir. Bu sayede algoritmanın durgunlaşmasını ve karınca kolonisi algoritmasının uygulama sürecinde global arama yeteneğini ortaya çıkarmaya çalışmışlardır. Oluşturulan simülasyon programı ile grid şeklinde haritalar oluşturularak bu algoritmayı uygulamışlardır. Simülasyon sonuçları algoritmanın etkili ve algoritmanın hızlı olduğunu göstermiştir.

Viet ve diğ. (2008), bir engel sakınma ile yol planlaması probleminde, iyi bir bilgi alışverişi ile iyi sonuçlar bulan işbirliği içindeki birkaç karınca kolonisi kullanımı ile oluşturulan bir multi karınca kolonisi sistemi önerilmiştir. Matlab dilinde oluşturmuş oldukları simülasyon programı ile koloniler arasında farklı bilgiler ile oluşan multi karınca kolonisinin davranışını göstermişlerdir. Bu program ile farklı davranışların kıyaslamasını yaparak algoritmanın verimli olduğunu göstermişlerdir.

Gao ve diğ. (2008), yol planlama problemlerinin bazı kusurlarını aşmak ve mobil robotların global yol planlaması için yeni bir karınca kolonisi algoritmasını önermişlerdir. Genetik algoritmanın mutasyon ve çaprazlama operatörleri karınca kolonisi algoritmasının kalitesini artırmak için kullanılmıştır ve sezgisel olasılık fonksiyonun ilk popülasyon oluşum sürecine eklenmiştir. Bu algoritma VC++ dilinde simülasyon olarak farklı çevrelerde uygulanmıştır ve yapılan testler sonucunda klasik karınca kolonisi algoritmasından daha yüksek kapasitede olduğu ve karınca kolonisi algoritmasını geliştirdiği görülmüştür.

Zhang ve diğ. (2009), karınca kolonisi algoritmasını bir kurtarma robotunun global yol planlaması için kullanmışlardır. Bu çalışmada, bir hedefe çekme fonksiyonu kompleks ve dinamik çevrelerde karınca kolonisi algoritmasının arama kalitesini yükseltmek için önerilmiştir. Bu algoritmanın etkinliği RoboCupRescue simülasyon sistemi üzerinde test edilmiştir.

Lee ve Jung Kim (2009), bazı yol planlama problemlerini çözmek için yeni bir KKA geliştirmişlerdir. Onlar daha önce Lee ve diğ. (2008) yaptıkları çalışmadan dört farklı yönü vardır. Birincisi, bu algorithmada karınca kolonisi algoritması içinde var olan karınca hareket kuralı için uzaklık yerine potansiyel alan kullanılmıştır ve bu potansiyel alan il feromen alanını oluşturmak için kullanılmıştır. İkincisi, feromen güncellemesi

yapılırken, yollar değere göre sınıflandırıldı. Üçüncü, belirli α , β gibi kontrol parametreleri değişiklikler yapılarak kullanıldı. Son olarak da, problemin çözümü için gereken süreyi azaltmak için klasik karınca kolonisi algoritmasını kullanmamışlardır. Matlab dilinde simülasyon olarak farklı çevrelerde uyguladıkları algoritmanın sonuçlarına göre fiziksel robotlar için optimal yolun diğer algoritmalara daha hızlı bulunduğunu görmüşlerdir.

Garcia ve diğ. (2009), özerk mobil robotların statik ve dinamik engellerden kaçınarak yol planlaması için bir bulanık mantık sistemi ile birlikte KKA kullanılarak yeni bir algoritma geliştirilmiştir ve bu metoda SACODm adı verilmiştir. Burada d mesafeler için, m ise hafıza içindir. Bu algoritma ile karar verme sürecine farklı özellikler eklenmiştir ve 10 civarında örnek çalışma yapılmıştır. Algoritmanın iki farklı çalışma biçimi vardır, birincisi sanal çevreler için, diğeri de kablosuz iletişim ile kurarak oluşturulandır. Yapılan çalışmalarına göre bu algoritmanın statik ve dinamik çevrelerde robotlar için çok iyi sonuçlar verdiğini göstermişlerdir.

Wang ve Xiong (2009), özerk sualtı araçlarının yol planlaması için bir KKA önermişlerdir. Önerdikleri çalışmada, çalışma uzayını görünürlük grafiği tekniğini kullanarak oluşturmuşlardır ve bu uzay üzerinde optimal yolları bulmak için karınca kolonisi algoritmasını kullanmışlardır. Algoritmayı simülasyon olarak farklı engeller ekleyip uygulayarak sualtı araçlarının yolunun daha kısa, daha emin ve daha düzgün olarak elde etmişlerdir.

Zotos (2009), yaptığı çalışmada satranç tahtası şeklinde bir grid haritası üzerinde insancıl robotların engellere çarpma olmaksızın yol planlaması için KKA kullanmıştır. Bu çalışmada Webost programı ile α ve β değerlerini 0-4 arasında değerler vererek simule ederek, sonuçlarını grafiksel hale getirerek algoritmanın performansını denemiştir.

Sariff ve Buniyamin (2010), global statik çevrelerde mobil robotlar için yol planlaması için bir KKA önermişlerdir. Önerdikleri algoritmada geçiş kurallarının sezgisel denklemi, optimal yolu bulan robot yol planlama çözümü için karınca kolonisini daha iyileştirmek içindir. Bu algoritma olası serbest uzay noktalarını kapsayan global statik

bir harita üzerinde uygulanmıştır. Algoritma Matlab dilinde yazılmış olup, sonuçlar zaman ve tur açısından performansı değerlendirilmiştir. Bu algoritmanın doğruluğu genetik algoritma ile yapılan başka bir çalışma ile kıyaslanarak değerlendirilmiştir.

Yang ve diğ. (2010), mobil robotların yol planlaması için yeni bir algoritma önermişlerdir. Bu algoritma, mobil robotların gerçek zamanlı global olarak optimal yol planlaması içindir. Algoritmada, mobil robotun hareket ettiği uzayı oluşturmak için Maklink graf tekniği kullanılmıştır ve bu oluşturulan uzay grafında başlangıç noktası ile bitiş noktası arasında optimal yolu bulmak için karınca kolonisi algoritması kullanılmıştır. Algoritma simulasyon ile uyarlanarak gerekli çevrelerde uygulanmıştır ve sonuçlar dijkstra algoritması ile karşılaştırılmıştır. Sonuçlara göre, önerdikleri algoritmanın daha etkili olduğu ve hızlı bir şekilde orijinal algoritmadan daha kısa yolları bulduğu görülmüştür.

Luo ve Wu (2010), karınca kolonisi algoritmasının ile potansiyel alanlar tekniğini kullanarak bilinmeyen bir çevrede mobil robotun yol planlaması için bir algoritma önermişlerdir ve buna ACOPF adını vermişlerdir. Bu algoritmada potansiyel alanlar tekniği engelden kaçmak ve robotun hareketi için kullanılmıştır. Bununla birlikte karınca kolonisi algoritması da başlangıç noktası ile bitiş noktası arasında optimal yolu bulmak için kullanılmıştır. C++ dilinde yazılmış simülasyon programı ile bu algoritmanın klasik karınca kolonisi algoritmasından daha verimli ve etkili sonuçlar verdiğini görmüşlerdir.

Wu ve diğ. (2010), yaptıkları çalışmada yol planlaması ile ilgili belli zorlukları ortadan kaldırmak ve aynı zamanda optimal yolu bulmak için çift yönlü bir KKA önermişlerdir. Bu çift karınca kolonisi farklı stratejilerde çalıştırılmıştır ve aynı anda iki yolu aramak bu algoritmayı global yapmıştır. Daha sonra bu bahsedilen stratejiler ile pozitif geri beslemeyi artırarak daha hızlı ve verimli bir algoritma elde edilmiştir ve sonunda da bu algoritma her bir yolu lokal olarak optimize etmektedir. Yapılan deneysel sonuçlar, algoritmanın basit bir şekilde uygulandığı sayısız çalışmadan sonra kısa zamanda optimal yolları bulduğu görülmüştür.

Shaogang ve Ming (2010), temel karınca kolonisi algoritmasına dayanan bir metodu birkaç kontrol noktasının arasında hareket eden bir denetleme robotunun yol planlaması için önerilmiştir. Önerdikleri algoritmada yol planlama problemini grid metodu ve görünürlük grafiği tekniğini uygulayarak oluşturdukları haritada gerçekleştirmişlerdir. Daha sonra bu harita üzerinde kontrol mekanizmaları arasında optimizasyonu sağlamak için karınca kolonisi algoritmasını kullanmışlardır. Algoritma Matlab dilinde simülasyon haline getirilerek sonuçlarının bu tür problemleri çözmeye uygulanabilirliğinin etkili olduğunu göstermişlerdir.

Zhao ve diğ. (2010), mobil robotların yol planlaması için KKA kullanmışlardır. Bu çalışmada ilk olarak α , β , ρ değerlerini verimli bir şekilde kullanımını sağlamak için iki bulanık denetleyici tasarlamışlardır. Sonra karıncalar için dinamik bir arama penceresi oluşturulmuştur ve son olarak yeni bir değerlendirme kriteri yolların nerede mükemmel olup-olmadığını ayırmak için önerilmiştir. Algoritma simülasyon programında uyarlanarak elde edilen sonuçlar bu algoritmanın 3 boyutlu karmaşık çevrelerde hızlı bir şekilde planlama yaptığını göstermiştir.

Lee ve diğ. (2010), global yol planlama probleminin çözümü için yeni bir karınca algoritması olan heterojen karınca kolonisi optimizasyonu adında bir algoritma önermişlerdir ve HACO olarak adlandırmışlardır. Onlar daha önce geliştirmiş oldukları karınca algoritmasını daha da geliştirmek için bu algoritmayı önermişlerdir. Geliştirmiş oldukları bu algoritma klasik karınca kolonisi algoritmasından üç açıdan farklıdır. Birincisi, geçiş olasılık fonksiyonunu ve feromen güncelleme kuralını değiştirmişlerdir. İkincisi, feromen kuralı içinde yeni bir yol çaprazlaması önermişlerdir. Son olarak ise, karınca kolonisi algoritmasında heterojen karıncalar kullanmışlardır. Bu algoritmayı Matlab' da oluşturdukları simülasyon programı ile farklı çevrelerde algoritmanın etkinliğini ve doğruluğunu göstermişlerdir.

Brand ve diğ. (2010), yaptıkları çalışmada bir grid haritası içersinde mobil robotların çarpışma olmaksızın en kısa yolu bulmak için ACO kullanmışlardır. Çalışmalarında sonuçları karşılaştırma açısından lokal ve global feromen güncellemesi olmak üzere 2 farklı güncelleme yaparak sonuçları bulmuşlardır ve çevreye herhangi bir engel eklediğinde robotun otomatik olarak ilk konumuna gidip tekrar optimum yolu araması

amaçlanmıştır. Çalışmalarını Python dilinde yazıp, simülasyon yaparak farklı engel ve dinamik çevrelerde deneyler yapmışlardır ve simülasyon sonuçları algoritmanın gayet başarılı ve iyi sonuçlar verdiğini göstermiştir.

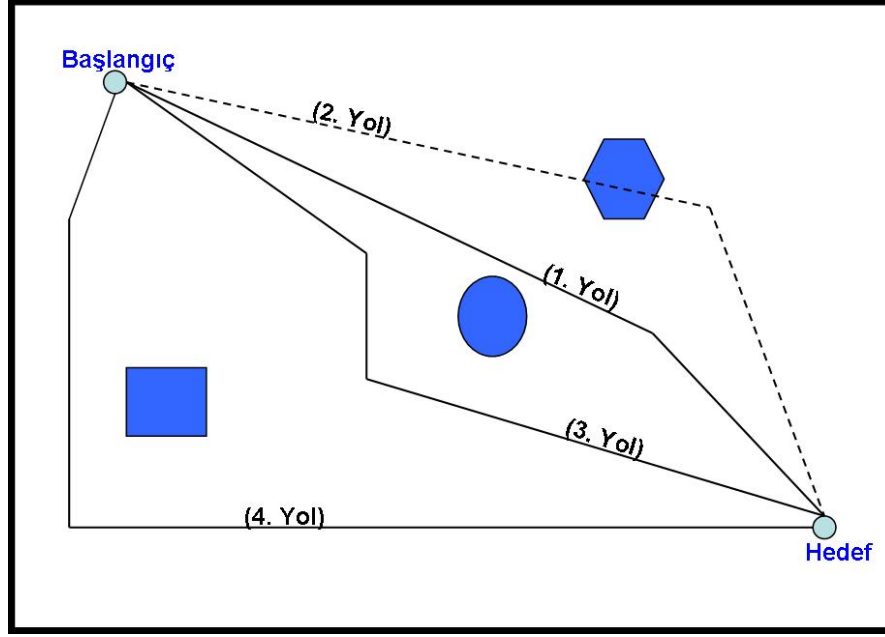
3. MOBİL ROBOTLAR İÇİN YOL PLANLAMA PROBLEMİ

3.1. GİRİŞ

Mobil robotlar keşfedildiği günden bugüne kadar hem özellikleri hem de kullanım alanları incelendiğinde büyük bir gelişim içinde olduğu görülmektedir. Mobil robotların bu kullanım alanları geçmişte sınırlı sayıda iken günümüzde özellikle endüstri alanında montaj, paketleme, depolama, taşıma gibi faaliyetlerin yanında sağlık hizmetlerinde, askeri ve güvenlik uygulamalarda, enerji kaynaklarında, temizlik sanayinde, tarım hizmetleri alanlarında kullanılmaya başlanmıştır. Bu geniş alanlarda kullanılması ile mobil robotlar için zamanla ortaya birçok problem çıkmıştır ve bunlar için birçok çözüm üretilmekle beraber çözümlerin iyileştirilmesine hala günümüzde devam edilmektedir. Mobil robotlar üzerinde yapılan araştırma konularından en önemli olanlarından birisi yol planlama problemidir.

Yol planlama problemi, belirli bir alan içerisinde istenilen bir noktadan hedeflenen başka bir noktaya, alan içerisindeki engellere çarpma olmaksızın optimal uzunluktaki yolu bulmayı hedefleyen bir problem türü olarak tanımlanabilir.

Yol planlama probleminin yukarıdaki tanımından anlaşılacağı üzere yol planlama işlemi yapılırken bazı kriterlerin sağlanması gerekir. Bunlardan en önemlileri planlanan yolun engeller ile temasının olmaması, düzgün bir yol olması ve en kısa yol olmasıdır. Şekil 3.1 de bir alan içerisinde bir yol planlamasının basit bir örneği gösterilmiştir. Farklı alternatif yollar verilerek bunlar arasında değerlendirmeleri yaparak yol planlamasının kriterlere uygun olup olmadığını daha rahat görebiliriz.



Şekil 3.1: Basit bir yol planlama örneğine ait kriterler

Şekil 3.1 de incelediğimizde; 1. Yol engellerle teması olmadığı ve optimum yol uzunluğu açısından da uygun bir yol olduğu görülmektedir. 2. Yol engellerden birisi ile direk olarak temas halinde olduğundan bu yolun uygun olmadığı görülmektedir. 3. Yol ise engellere çarpma olmaksızın 1. Yol'a göre engellerden uzaklık mesafesine göre daha emniyetli bir yol olduğu görülmektedir. Burada dikkat edilmesi gereken nokta engele çarpma olmamasına rağmen yol planlama probleminin tam olarak çözümü için optimal yol kriterinin de sağlanmış olmasıdır. 4.Yol engellere çarpma açısından uygun bir yol olduğu görülmektedir fakat optimal yol açısından 1. ve 3. yollara göre uygun olmayan bir yol olarak görülmektedir. Sonuç olarak 1. ve 3. yollar bu yol planlama problemi için birbirine çok yakın optimal çözümlerdir.

Yukarıdaki işlem basamaklarına bakıldığında yol planlama probleminin çözümü için sadece bir kriterin sağlanması yetmediği görülmektedir. Yol planlama probleminin çözümü için belirlenen bütün kriterlerin aynı anda sağlanmış olması gerekmektedir. Bu temel kriterlerin dışında bazı yol planlaması problemlerinde değişik kriterler (zaman, enerji v.b.) yol planlaması problemlerine eklenebilir. Tüm bu kriterler sağlandığında yol planlaması probleminin en doğru sonucu vereceği görülmektedir (Nagıb ve Gharie 2004).

Temel yol planlama problemleri genel olarak çevrenin bilindiđi durumlarda üzerinde sıkça çalışılmıřtır. Buradan yola çıkarak robotun hareket edeceđi çevre ve çevredeki engeller hakkında yol planlaması yaparken 4 ana madde ile sınıflandırabiliriz.

- Tamamen bilinen çevrelerde statik engeller için yol planlama.
- Kısmen bilinen veya hiç bilinmeyen çevrelerde statik engeller için yol planlama.
- Tamamen bilinen çevrelerde dinamik engeller için yol planlama.
- Kısmen bilinen veya hiç bilinmeyen çevrelerde dinamik engeller için yol planlama.

Yukarıdaki maddelerden ilk 2 tanesi engellerin durumundan dolayı belirli koşullar altında genellikle harika sonuçlar vermektedir (Zhuang ve diğ. 2006). Pratikte yine de mevcut çevre hakkında genellikle bilginin tam olmadığı bir problem kullanılır. Çođu durumda, belirlenmiř bütün engeller ile ayrıntılı bir haritanın oluşması gerçekçi görülmemektedir. Özellikle açık hava uygulamalarında örnek olarak GPS uygulaması ile gerekli koordinat sistemi elde edebilmektedir. Buna rağmen çevre hakkında elde edilen bilgi çok sınırlı olabilir. Bu koşullar altında çok ayrıntılı ve mantıklı bir planlama için ortaya çok fazla belirsizlik çıkmaktadır. Planları güncelleřtirmenin en dođal yolu mevcut bilgide bulunan bir yolun başlangıç olarak seçilip, daha sonra yeni bilgiler toplanıyor iken kısa bir zaman boyunca o yol üzerinde hareket ettirilmesidir. Yeni elde edilmiř bilgilere göre yol daha sonra tekrar planlanır. Bu yöntem bilinmeyen alanlarda yol planlaması için literatürde sıkça kullanılmaktadır (Ersson ve Hu 2001). Eđer kısmi bir harita üzerinde çalışıyorsa amaç, arzu edilen hedefe direk olarak yönelim davranışının daha çok olmasını sağlamaktır. Lumelsky ve Stepanov (1986) algoritmalarında robot arzu edilen hedefe giderken sanki hiç engel yokmuş gibi hareket ederek hedefe direk olarak hareket etmektedir. Bir engelle karşılařtığı zaman, robot engel üzerindeki bir noktaya en yakın hedefi buluncaya kadar engel etrafında hareket eder, hedefi bulduđu anda o hedefe direk olarak hareket etmeye başlar.

3.2. GLOBAL VE LOCAL YOL PLANLAMA

Yol planlaması için global yol planlaması ve lokal yol planlaması olmak üzere 2 farklı yaklaşım vardır. Global yol planlaması yapılırken çevre hakkında mükemmel bir bilgiye sahip olduğu varsayılır. Bu mükemmel bilgi ile yol planlaması yapılacak alan içerisindeki engellerin pozisyonları, şekil değişiklikleri, boyutları ve koordinatları tam olarak bilinmesi sağlanır. Çevre bilgisinin bilinmesi ile global yol planlaması algoritması çevrimdışı çalışır ve mobil robot veya benzeri araçların daha harekete başlamasını beklemeden başlangıç noktası ile hedef noktası arasındaki yolu planlar. Yani global yol planlamasında hem amacın tanımlanması yapılır hem de engellerin statik boşluğu müsait hale getirilir. Bu yaratılan yol kullanarak harekete başlanmadan önce yol planlaması için referans ve rehber sağlanmış olur (Buniyamin ve diğ. 2011).

Lokal yol planlaması yaklaşımında ise yol planlaması için gerekli bilgiler yerel bazı bilgilerin kaynağı ile gerçek zamanlı yapılır. Lokal yol planlaması, başlangıç pozisyonu ve hedef pozisyonu dışında çevre hakkında hiçbir bilgiye sahip değildir. Lokal planlamada işlem çevrimiçi yapıldığından yol planlaması sırasında planlama yapılır. Yani tepkisel bir süreç vardır. Dolayısıyla planlama yapılırken karşılaşılan engellere tepki verilerek, bu engellerden kaçınılarak planlama yapılır.

Global ve lokal yol planlama yaklaşımları arasındaki temel farklar Çizelge 3.1 de temel unsurları ile gösterilmiştir. Bu iki yaklaşımı karşılaştırdığımızda global yol planlamasının lokal yol planlamasına göre daha yavaş ve daha fazla süreçte sonuçlandığı görülmüştür. Bunun sebebi ise lokal yol planlamasını tepkisel bir süreç ve dinamik olmasıdır. Global yol planlama yaklaşımı engeller açısından yerele göre daha büyük engellerde daha yüksek düzeyde çalıştığı görülmektedir. Bununla birlikte global yol planlamasında çevre tamamen bilindiği için global planlayıcı yolun kalitesini yüksek derecede geliştirdiği görülmüştür.

Çizelge 3.1 : Global ve lokal yol planlamasının karşılaştırılması

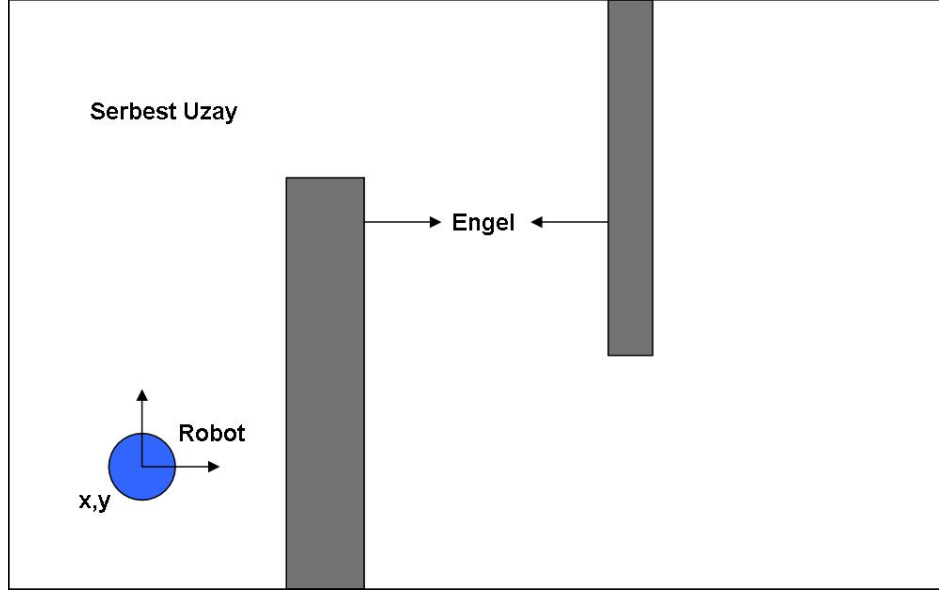
GLOBAL YOL PLANLAMA	LOKAL YOL PLANLAMA
<ul style="list-style-type: none"> • Biriktirilmiş ve önsel yol temelli bilgiler kullanılır. • Verimli ve uygun bir şekilde hedefe gitmeyi planlar. • Dikkatli ve yavaş bir süreç vardır. • Uzun mesafeler ve uzun periyotlar için planlama yapar. • Basit model araçların planlaması için kullanılır. • Daha büyük engellerin bulunduğu alanlarda kullanılır. 	<ul style="list-style-type: none"> • Acil, tepkisel sensör temelli bilgiler kullanılır. • En hızlı ve en mümkün seyahati planlar. • Tepkisel ve hızlı bir süreç vardır. • En yakın çevre ve en kısa periyotlar için planlama yapar. • Kompleks model araçların planlaması için kullanılır. • Küçük engellerin bulunduğu alanlarda kullanılır.

3.3. YOL PLANLAMA UZAYI

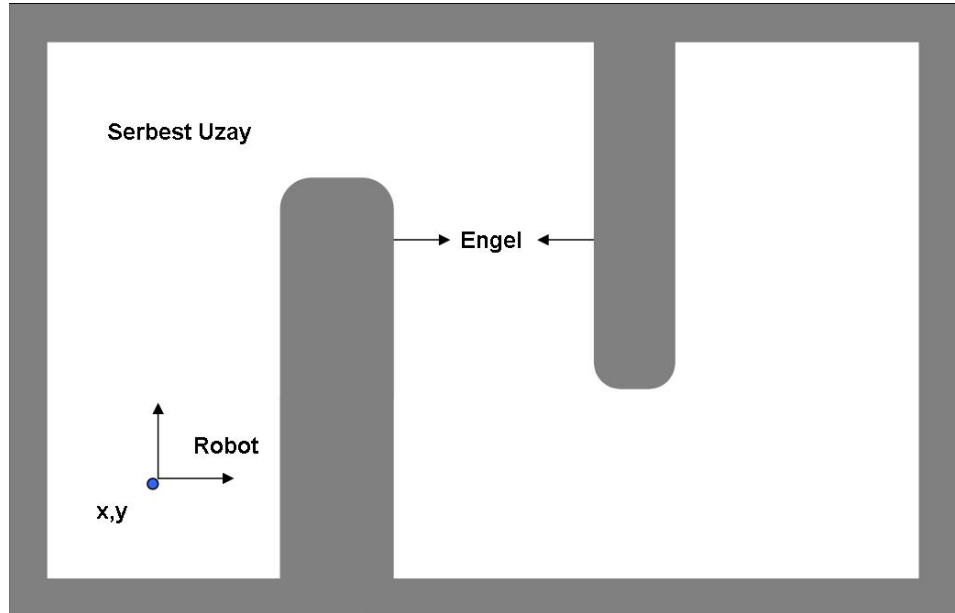
Yol planlama problemleri işlemsel olarak çok karmaşık problemler olarak ifade edilebilir. Bunun nedeni ise yol planlaması yapılan alan içerisinde birçok yol seçeneğinin bulunması ve bu yollar arasında karar vermeye çalışılmasıdır. Bu sebeple yol planlama problemleri konfigürasyon uzayı boyutlarına bağlı olarak üstel olarak artış gösterdiğinden NP-Hard veya P-Space-Hard problemler sınıfına girmektedir ve bütün mümkün çözümlerin denenmesi ile çözülebilen problemlerdir.

Yol planlama problemlerinin literatürdeki genel tanımına göre yol planlamada temel fikir mobil robotların insana ihtiyaç duymadan istenilen noktalar arasında otomatik olarak hareket edebilmesidir. Robot otomatik olarak yol planlaması yaparken bulunduğu alan içerisinde bilinen veya bilinmeyen engeller bulunabilir ve robot bu engellere çarpmadan engeller etrafında hareket edip, manevra yapabilmelidir. Buradan yola çıkarak genel tanımlarıyla çalışma uzayı (W-Space) robotun içinde operasyonlarını gerçekleştirdiği, içinde engelleri ve hedefleri barındıran bir uzay olarak tanımlanırken konfigürasyon uzayı (C-space), robotun mümkün olan bütün konfigürasyonlarını temsil eder. Bununla birlikte engeller dışında robota manevra imkânı veren fiziksel uygun bölgelerde serbest uzay (C-free) olarak tanımlanmaktadır.

Konfigürasyon uzayı robotun çalışma uzayının bir çeşit indirgenmiş hali olarak düşünülebilir. Çalışma uzayının Şekil 3.2 üzerinde ve indirgenmiş hali olarak konfigürasyon uzayı da Şekil 3.3 üzerinde açıklanmıştır.



Şekil 3.2: İki boyutlu çalışma uzayı



Şekil 3.3: İki boyutlu konfigürasyon uzayı

Şekil 3.2 de görülen basit bir disk robotu için çalışma uzayıdır. Robot 2 boyutlu yuvarlak bir şekildedir. Konfigürasyonlar bu yüzden x,y şeklindedir. Eğer robot burada

dairesel olmasaydı bir bağımsız hareket yönelimsel kademesi de belirtilmesi gerekecekti. Şekil 3.3 de ise yine basit bir robot için verilen konfigürasyon uzayıdır. Konfigürasyon uzayı görüldüğü gibi Şekil 3.2 de robot için verilen çalışma uzayına benzemektedir. Konfigürasyon uzayında robot bir nokta olarak belirtilir. Dolayısıyla her bir nokta boşluktaki gerçek bir noktadan ziyade, robotun her bir konfigürasyonuna benzemektedir. Nasıl oluştuğunu görmek için şekilde de gösterildiği gibi bir referans noktası olarak robot üzerinde bir noktayı seçelim. Çalışma alanının sınırları ve engellerin yüzeyleri boyunca yürütülen robot olarak varsayılan nokta tarafından dışa doğru kavislerin yayıldığı görülmektedir. Bu robotun nokta halinde geldiğinden dolayı engellerin de bu durumu karşılaması için genişletilmiştir ve bu nokta şeklindeki robotun engellerle temasının denetimi sırasında dikkate alınması gereken bir husustur.

Konfigürasyon uzayının boyut sayısı robotun bağımsız olarak hareket etme kademesine bağlıdır. Yani eğer robot n tane bağımsız olarak hareket kademesine sahipse konfigürasyon uzayı da genellikle n boyutludur. İyi bir konfigürasyon uzayı planlama algoritmalarının hızlı bir şekilde çalışabilmesi için az sayıda boyuta sahip olmalıdır. Kompleks bir sistem için bağımsız olarak hareket kademesi eklendiği anda konfigürasyon uzayı hızlı bir şekilde büyür ve boyut sayısı planlamayı zor hale getirir. Latombe (1991) önerdiği gibi konfigürasyon uzayında boyutların sayılarından dolayı planlama problemi zamanla artarak zor hale gelmektedir. Buda planlama yaparken konfigürasyon uzayının boyutunun dikkate alınması gerektiği anlamına gelmektedir. Bu sebeple araştırmacılar genellikle yol planlayıcısının hedefe daha hızlı bir şekilde ulaşması için problemlerde bazı basitleştirmeler kullanmışlardır. Örneğin bazı yol planlama problemleri için mümkün konfigürasyonları x,y koordinat sistemine indirgenmiştir.

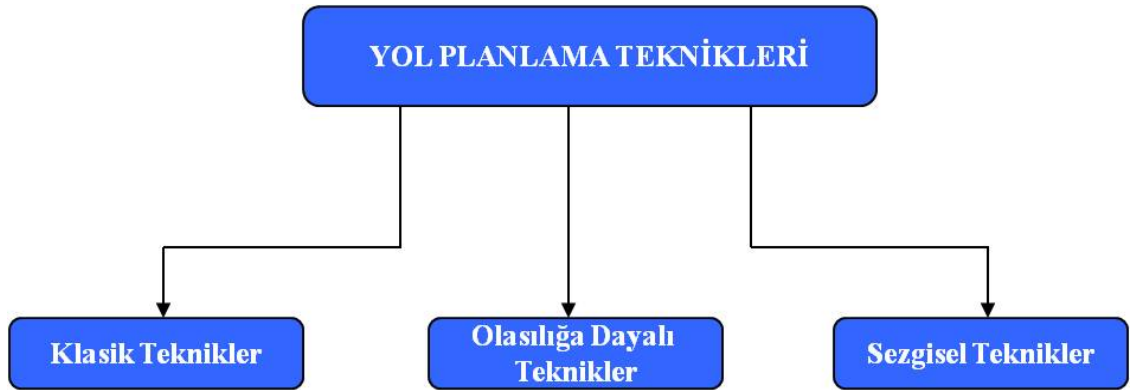
Bir yol planlaması için sürekli veya ayrık arama uzayı kullanılabilir. Sürekli bir durum uzayı olası durumların sonsuz sayısının sürekli bir kümesi gibi dünyayı modeller. Genellikle sürekli durum uzayında planlama yapmak daha zordur. Sürekli durum uzayı global yol planlamasından ziyade reaktif olarak engellerden sakınma için kullanılır. Ayrık bir durum uzayı kullanıldığı zaman arama uzayı robotun ve benzeri aracın bulunduğu olası ayrık durumların sonsuz sayısına bölünür. Ayrık ve sürekli durum uzayları birçok seviyede uygulanabilir. Örneğin yol planlamasının potansiyel alan

metotları, matematiksel fonksiyonlar ile tarif edilen sürekli bir alana uygulanabilir veya fonksiyonlar yol planlayıcısının baştan sona aradığı bir grid haritasına bölünmüş olabilir (Giesbrecht, 2004).

3.4. YOL PLANLAMA YÖNTEM VE TEKNİKLERİ

Mobil robotlar için yol planlama probleminin çözümü ile ilgili çalışmalar uzun yıllardan beri devam eden ve gün geçtikçe de üzerinde daha da çok çalışılmaya devam edilen bir problemdir. Araştırmacıların oldukça önemli bir kısmı yol planlama probleminin çözümü için etkili ve verimli teknikler geliştirmek için yoğun bir şekilde çaba sarf etmişlerdir. Araştırmacılar bu teknikleri geliştirirken 2 genel yöntemi ele almışlardır. Birincisi önceden bilinen global çevre veya engel bilgisi ve robot karakteristik özellikleri kullanarak yol planlama çözümüne ulaşılması ile ikinci olarak lokal bilgiler ve yine robot karakteristik özelliklerini kullanarak hareket planlama üzerinedir.

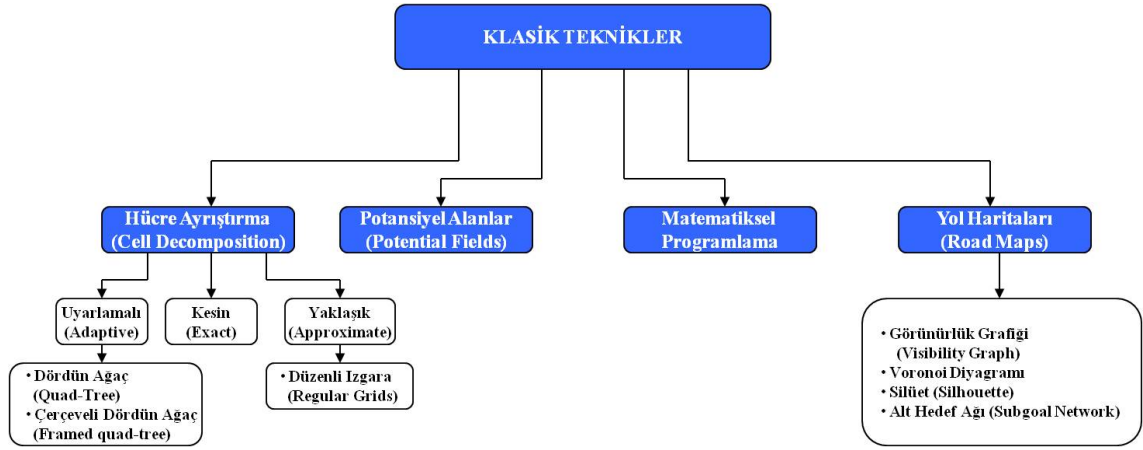
Yol planlama problemi çözümü için birçok teknik geliştirilmiştir. Yol planlama teknikleri çalışmaları, performansları, oluşturdukları çözümler açısından aşağıdaki Şekil 3.4 de görüldüğü gibi 3 ana başlık altında toplanmıştır.



Şekil 3.4: Yol planlama teknikleri

3.4.1. Klasik Yol Planlama Teknikleri

Klasik yol planlama teknikleri mobil robotlar için yol planlama probleminin çözümü için yapılan çalışmaların en eski teknikleri olarak nitelendirilebilir. Klasik yol planlama teknikleri ile yol planlama problemlerinin büyük bir çoğunluğu çözülebilir ve bu yüzden uzun yıllar yapılan çalışmalarda klasik teknikler kullanılmıştır. Klasik yol planlama teknikleri Şekil 3.5 de gösterildiği gibi 4 ana teknik ile ifade edilebilir.



Şekil 3.5: Klasik yol planlama teknikleri

3.4.1.1. Hücre Ayrıştırma Tekniği

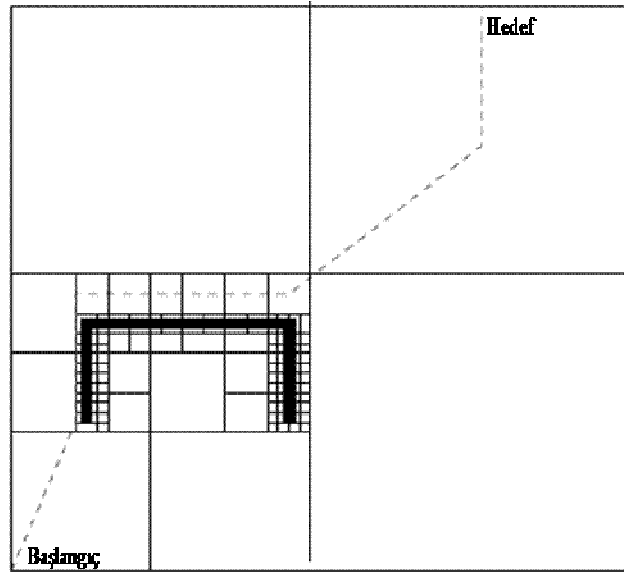
Hücre ayrıştırma teknikleri robotik uygulamalarında en çok çalışılmış ve yaygın olarak kullanılan bir tekniktir. Bu teknik ile serbest konfigürasyon uzayı basit bir hücre serisine ayrıştırılır ve çok yakın bağlantılar arasındaki hücreler hesaplanır. Robotun harekete başlayacağı nokta ile hedef arasında çarpışmadan olmadan hareket edeceği bir yol ilk önce tanımlanmış başlangıç ve hedef noktasını kapsayan 2 hücre tarafından bulunur ve daha sonra onlar hücrelerin bir ardışık serisine bağlanırlar. Bu teknikler bir hücreden bitişik hücrelere erişebilirlik için bağlayıcı grafikler kullanılırlar. Robot daha sonra bağlayıcı grafikler ve buna bağlı olarak üretilen yollar sayesinde ardışık hücreler serisi ile hedefe gitmeye çalışır. Bir hücreyi oluşturmanın değeri değişebilir ve planlayıcı optimal yolu belirlemek için iyi yöntemler kullanılmalıdır.

Hücre ayrıştırma teknikleri genellikle fiziksel boşluğu oluşturmak için kullanılırlar fakat bazı durumlarda boşluğun yapılandırması içinde kullanılabilirler. Hücre ayrıştırma teknikleri yaklaşık hücre ayrıştırması, uyarlamalı hücre ayrıştırması ve kesin hücre ayrıştırması olmak üzere 3 ana başlık altında toplanmıştır (Giesbrecht, 2004).

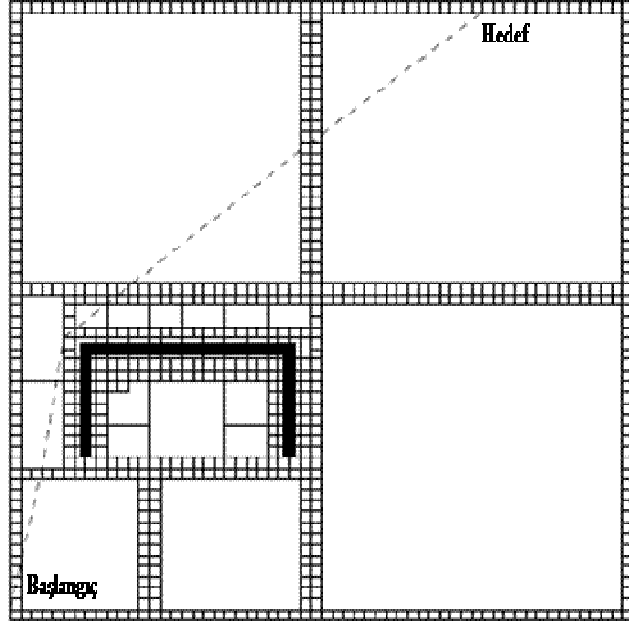
Yaklaşık hücre ayrıştırma, planlama yapılacak alan üzerinde yayılmış düzenli bir grid oluşturulur. Yaklaşık hücre ayrıştırması ilk olarak Brooks ve Perez (1983) tarafından ortaya çıkmıştır ve daha sonra Zhu ve Latombe (1989) tarafından geliştirilmiştir. Bu teknikte her hücrenin orta noktası, bir yolu bulmak için incelenecek olan arama grafiğinde bir düğüm olur. Bu tekniğin yaklaşık hücre ayrıştırması olarak tanımlanmasının sebebi fiziksel nesnelerin sınırları ile önceden tanımlanmış hücre sınırlarının uyuşmamış olmasıdır. Grid hücrelerin önceden tanımlanmış şekil ve boyutu olduğundan bu tekniği diğer hücre ayrıştırma tekniklerine göre dünya uzayına uygulamak son derece kolaydır ve ihtiyaca göre esnek tekniklerdir. Hücrelerin boyutları hesaplama zamanının azaltmak için uyarlanabilir, böylece baştan sona arama yapmak ve daha çok ayrıntıyı elde etmek için hücrelerin boyutları azaltılabilir. Dolayısıyla bu hücreler mobil robotlar için cazip hale gelir. Ayrıca ayrıştırma hücrelerin çapraz olarak çizgi oluşturması için bazı değerleri vardır. Eğer bu değer binary bir değerse harita basitçe kapalı robotikleri için çok ortak noktası olan engel ve serbest uzay bölgelerini kapsar. Bununla birlikte bu tekniğin uygulama kolaylığına rağmen bu teknikte birkaç sorun vardır. Örneğin bunlardan en önemlisi sayısal eğilimdir. Yaklaşık hücre ayrıştırması kullanılırken konfigürasyon uzayını bölmek için farklı birçok şekilden biri kullanılır (Giesbrecht, 2004).

Uyarlamalı hücre ayrıştırma tekniği ise, açık alanlarda kullanılan hücrelerin sayısını azaltmak ve yaklaşık hücre ayrıştırmasının sayısal eğilimini ortadan kaldırmak için kullanılır. Bu teknik robotun seyahati için oluşturulacak çapraz çizgi yeteneği ile büyük alanların olduğu geniş doğal araziler için verimli bir şekilde kullanılan bir uygulamadır. Uyarlamalı hücre ayrıştırması, düzenli hücre ayrıştırmasının serbest uzaydaki bilginin hepsinin lüzumsuz olduğu gerçeğine dayanmaktadır. Hücrelerin düzenli şekli yine de sürdürülür fakat hücreleri daha verimli bir şekilde kullanmak ve daha fazla detayı sağlamak için boşlukta bazı değişiklikler yapılır. Böylece sonuç daha az hafızadan ve daha az işlem zamanından meydana gelir. Uyarlamalı ayrıştırma tekniğinin en çok kullanılan türü dördün ağaç tekniğidir. Dördün ağaç tekniği ilk defa Samet (1988) ile Noborio ve diğ. (1990) tarafından ortaya atılmıştır. Dördün ağaç tekniği tüm planlama boşluğunun üzerine büyük boyutlu bir hücre koyarak başlar. Eğer grid hücresi kısmen işgal edilmişse hücre 4 eşit alt bölgeye ayrılır ve daha sonra planlama uzayına tekrar uygulanır. Daha sonra tekrar alt bölümlere ayrılır. Bu yapılan tekrar hücrelerinin her

birinin tamamıyla dolmasına veya tamamıyla boşalmasına kadar devam eder. Bu durumda ortaya çıkan haritada farklı boyutta ve yoğunlukta grid hücreler oluşur fakat hücre sınırları çok yakın benzer engel sınırları ile bitişiktir. Uyarlamalı hücre ayrıştırması robotun yeni veri elde etme ve yeni engellere dayalı olarak haritaların güncellemesinin yapıldığı dinamik ortamlar için bazı problemler teşkil etmektedir. Bu durumda haritanın tüm veri yapısı için tamamen düzeltilmesine ihtiyaç vardır. Sürekli değerlere sahip haritalar yani doğal araziler ile dördün ağaç tekniğini aynı anda kullanmanın bazı zorlukları da vardır. Çünkü işgal edilmiş veya özgür bölgelerde dünyayı alt bölümlere bölmek oldukça zordur. Buna ek olarak, dördün ağaç yakın optimal yolları sağlamada da bazı zorluklara sahiptir ve yoğunlukla sivri uçlu, çentikli yollarla sonuçlanırlar. Bundan dolayı da daha iyi bir çözüm olarak Chen ve diğ. (1995) tarafından çerçeveli dördün ağaç tekniği önerilmiştir. Dördün ağaç ile çerçeveli dördün ağaç arasındaki fark Şekil 3.6 de ve Şekil 3.7 de görülmektedir. Yine de yüksek kaba ve karışık çevrelerde çerçeveli dördün ağaç tekniği düzenli gridlere göre daha az verimli sonuçlar verebilmektedir.

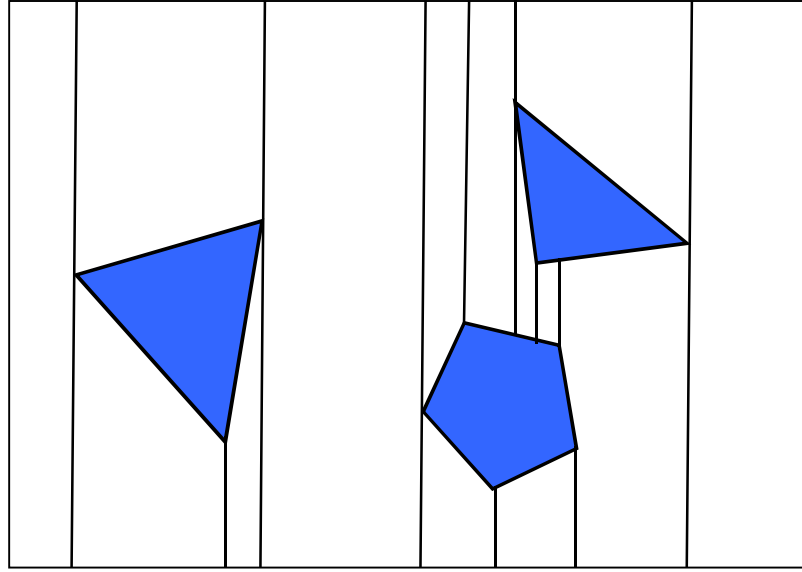


Şekil 3.6: Yol planlama için dördün ağaç tekniği (Giesbrecht, 2004).



Şekil 3.7: Yol planlama için çerçeveli dördün ağaç tekniği (Giesbrecht, 2004).

Tam hücre ayrıştırması ise, farklı bir yolda düzenli gridler ile yol planlama problemlerin bazı çözümleri için uğraşır. Hücrelerin önceden tanımlanmış bir boyutu veya şekli yoktur fakat hücreler dünya haritasına dayanarak, dünya haritası içindeki engellerin şekline göre belirlenir (Şekil 3.8). Hücre sınırları planlama uzayındaki sınırlar ile tam olarak uyuşur ve hücrelerin birleşimi tam olarak dünyanın özgür boşluğudur. Bu tam hücre ayrıştırmasını eksiksiz yapar ve onlar sürekli olarak bir yol bulmaya çalışacaktır. Fakat bu hücreler optimal yol ile sonuçlanmayacaktır. Bu tekniği düzensiz ve iyi nitelendirilmemiş engellerin bulunduğu çevreler için uygulamak oldukça zordur. Bu teknik ile örnek çalışmalar See Schwartz ve Sharir (1983) ile Avnaim ve diğ. (1988) tarafından yapılmıştır. Bununla birlikte mobil robotların yol planlaması için tam hücre ayrıştırması tekniğinin uygulaması Sleumer ve Gurman (1999) tarafından yapılan çalışmada daha detaylı görülmektedir.



Şekil 3.8: Yol planlama için tam hücre ayrıştırması örneği

3.4.1.2. Potansiyel Alanlar Tekniği

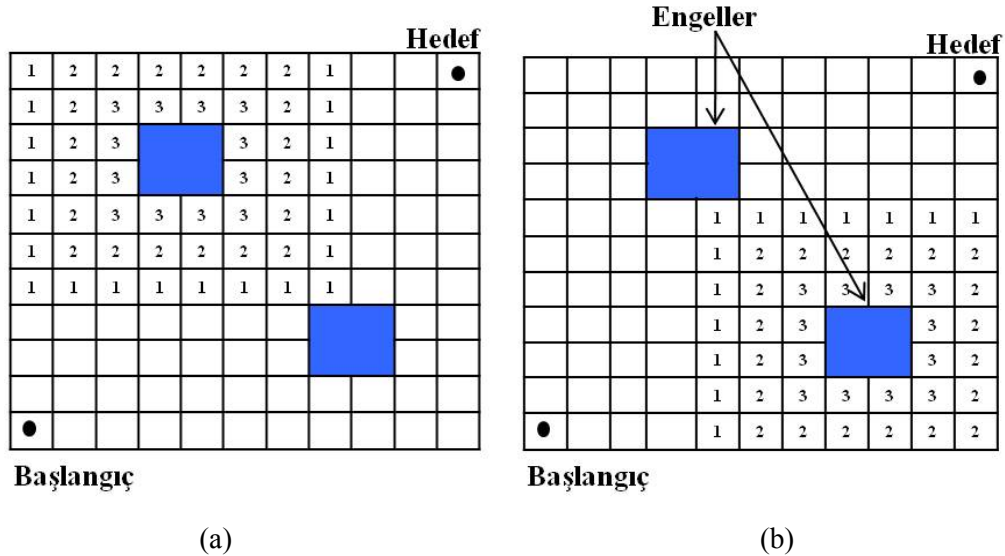
Potansiyel alan tekniği, diğer yol planlama tekniklerinden tamamen farklıdır ve geçmişte yaygın olarak kullanılmış bir tekniktir. Potansiyel alan tekniği robotun hareket edeceği tüm alan üzerinde bir matematiksel fonksiyon kullanır. Bu teknik, robota hedefler tarafından üretilen alanların etkisi altındaki bir nokta gibi ve dünya içerisindeki engeller elektriksel alan içerisindeki bir elektron gibi davranır. Dolayısıyla hedefler robotun hedefe çekilmesi için çekici kuvvetleri, robot yakın engeller ise itici kuvvetleri oluşturur (Giesbrecht, 2004). Potansiyel alanlar tekniğinin bu çekici ve itici kuvvetleri Denklem 3.1 ve Denklem 3.2 de görüldüğü gibi genel formüllerle bulunmaktadır.

$$F_{\text{çek}} = -k_{\text{çek}}(x - x_{\text{hedef}}) \quad (3.1)$$

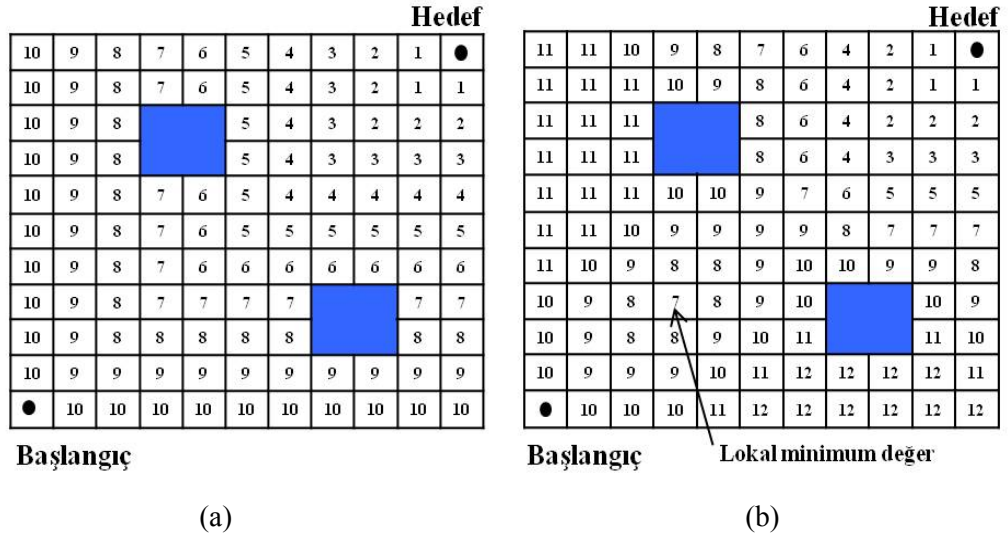
$$F_{\text{iti}} = \begin{cases} k_{\text{iti}} \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right) \frac{1}{\rho^2} \frac{\partial \rho}{\partial x} & \text{eğer } \rho \leq \rho_0 \\ 0 & \text{aksi takdirde} \end{cases} \quad (3.2)$$

Burada; x robotun pozisyonunun koordinatı, x_{hedef} hedef pozisyonunun koordinatı, ρ robottan engele olan uzaklık, ρ_0 engelin etkili olduğu mesafe, $k_{\text{çek}}$ ve k_{iti} pozitif ölçeklendirme faktörlerdir.

Potansiyel alan tekniđi gerekte ilk önceleri yol planlaması yerine gerek zamanlı engellerden kaçınmak için keşfedilmiş bir tekniktir. Potansiyel fonksiyonların engellerden kaçmak için kullanılması fikri ilk olarak Khatib ve diğ. (1978) ile güç kontrolü için Hogan (1985) tarafından ortaya atılmıştır. Bununla birlikte Miyazaki ve Arimoto (1985) ile Pavlov ve Voronin (1984) ve Koditschek(1987), Rimon ve Koditschek (1992) tarafından üzerinde çalışılıp daha verimli hale getirmek için farklı teoriler geliştirilmiştir. Bu teknik öncel bir bilgiden ziyade dinamik olarak aldıkları bilgilerden yararlanırlar. Bu metotların uygulaması nispeten kolaydır ve sayısal olarak etkili ve verimli çalışırlar. Fakat potansiyel alan tekniđinin en büyük sorunu yerel olarak minimum değere sahip olmasıdır. Yani robot başlangıçtan hedefe giderken bir yerel bir minimum değere takılı kalarak sanki hedefe varmış gibi davranır. Aşağıdaki şekillere bakıldığında, Şekil 3.9.a de birinci engel için, Şekil 3.9.b de ikinci engel için bir alan oluşturulmuştur. Bununla birlikte Şekil 3.10.a de hedefe çekici kuvvetler için bir alan oluşturulmuştur ve Şekil 3.10.b de tüm bu alanların toplamı olan bir alan oluşturulmuştur ve toplam alan robotu düşük potansiyel alan seviyeleri ile robotu yönetebilecekti ama görüldüğü gibi minimum lokal değerlere dikkat etmek gerekmektedir. Özellikle potansiyel alanın bu sorunundan dolayı başta potansiyel alanlar tekniđi üzerinde yapılan çalışmaların çoğu bu sorunu ortadan kaldırmak içindir.



Şekil 3.9:(a) Birinci engel için oluşturulan potansiyel alan.(b) İkinci engel için oluşturulan potansiyel alan



Şekil 3.10:(a) Hedef için oluşturulan potansiyel alan.(b) Toplam potansiyel alan

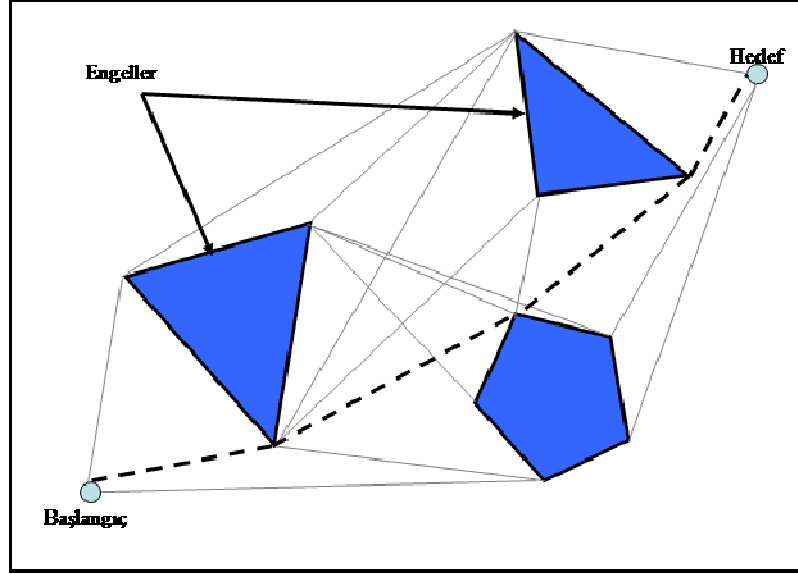
3.4.1.3. Matematiksel Programlama Tekniği

Matematiksel programlama tekniği yol planlama problemlerinin çözümü için pek fazla kullanılmamaktadır. Matematiksel programlama tekniği, konfigürasyon parametrelerine bağlı eşitsizliklerin bir kümesi ile engelden kaçınmanın gereksim olduğu yerlerde kullanılmaktadır. Bu teknik ile mobil robotlar için yol planlaması, başlangıç ve hedef konfigürasyonları arasında belirli bir sayısal miktarı minimize eden bir kavis veya eğri bulan matematiksel bir optimizasyon problemi olarak formüle edilir. Böyle bir optimizasyonun doğrusal değildir ve eşitsizlik sınırları vardır, sayısal bir teknik optimal çözümü bulmak için kullanılır (Hwang ve Ahuja, 1992). Maciejewski ve Klein (1985) tarafından bir matematiksel programlama ile dinamiksel çevrelerde gereksiz manipülasyonlar için yerel engel çarpışmasını önleyecek bir matematiksel algoritma geliştirmişlerdir. Bununla birlikte Chen ve Vidyasagar (1988) bu tekniği kullanarak yol planlama problemlerinin çözümü için uğraşmışlar.

3.4.1.4. Yol Haritaları Tekniđi

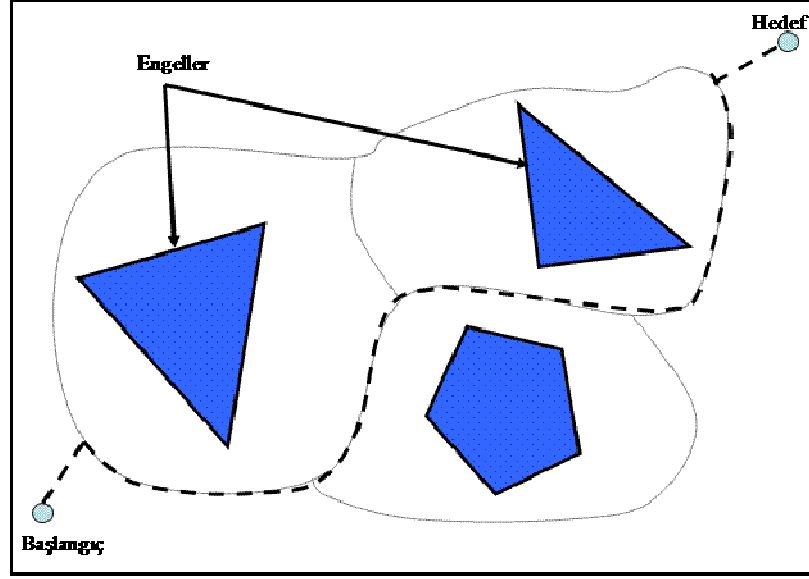
Yol haritaları tekniđi, mobil robotların yol planlama probleminin çözümlü için sıkça kullanılan tekniklerden birisidir. Yol haritaları, bir noktadan başka bir noktaya nasıl gidileceđini temsil eden grafikler olarak tanımlanabilir. Yol haritaları teknikleri tek boyutlu eğrilerin bir kümesi gibi robotun serbest uzayı arasındaki bağlantıları ortaya çıkarır. Yol haritaları oluşturulduđu andan itibaren planlamacının optimal çözümleri arayacak olduđu yolların bir kümesi olarak kullanılır. Grafik içersindeki düğümler genellikle robotun başarılı bir hareket yapabilmesi için yardımcı yol noktalarıdır. Yol haritaları tekniđi, planlamacının bir yolu bulmak için baştan sona aramaya ihtiyaç duyduđu düğümlerin sayısından dolayı hücre ayrıştırma tekniklerine göre büyük avantaja sahiptir. Bu düğümlerin kümesi konfigürasyonların hepsinden meydana gelmek yerine seçilmiş özel az bir kısımdan meydana gelmektedir. Yol haritaları genellikle hızlıdır ve bu yöntemleri uygulamak çođu zaman kolaydır. Bundan dolayı kullanım alanları çok fazladır. Yol planlama tekniđi grafiksel yapısından dolayı görünürlük grafiđi, voronoi diyagramı, silüet ve alt hedef ađı olmak üzere 4 alt başlık altında toplanabilir.

Görünürlük grafiđi tekniđi, 2 boyutlu konfigürasyon uzayına uygulanmış olan yol haritası tekniđinden birisidir ve özellikle Nilsson (1984) öncü olmak üzere birçok araştırmacı bu tekniđi kullanmıştır. Görünürlük yol haritası, Şekil 3.11 de görüldüđu gibi engellerin içinden geçmeden bu engellerin düğümlerini düz çizgiler yardımıyla bağlar. Bu düz çizgiler robot veya benzerinin gidebileceđi yolları ifade eder ve birkaç arama tekniđi ile istenilen optimal yol görünürlük yol haritası üzerinde bulunabilir. Görünürlük grafiđi tekniđi genellikle yetersiz olarak görülmektedir çünkü belirlenen yollar engeller ile görüldüđu gibi teđetseldir. Diđer bir problem, engellerin belirlenmiş olan şekilleridir. Bu engellerin şekliyle ilgili olan sorun özellikle dış hava robotları için büyük bir problemdir çünkü engeller neredeyse her zaman yuvarlak veya biçimsiz bir şekildedir. Basit bir görünürlük grafiđi Şekil 3.11 de görüldüđu gibi engellerin köşeleri ile bađlı olan yollardan en optimal yol kesikli çizgilerle belirtilmiştir.



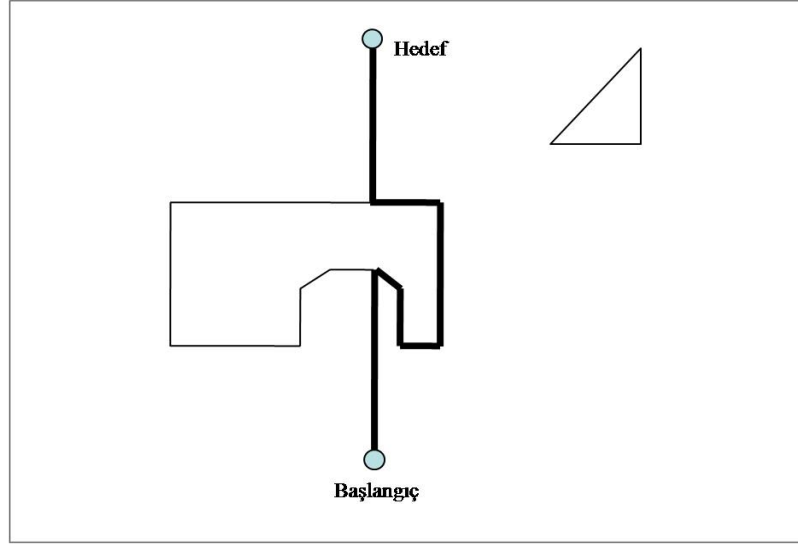
Şekil 3.11: Yol planlama için görünürlük grafiği örneği

Voronoi diyagramı tekniği, bir konfigürasyon uzayından bir yol haritası oluşturmak için kullanılan en popüler tekniklerden birisidir. Voronoi diyagramı birbirinden bağımsız küme elemanlarının birbirleri ile arasındaki uzaklıkları ifade eder. Voronoi diyagramı ismini Georgy Voronoi'den almasına rağmen çok daha önceleri de voronoi diyagramı birçok çalışmada kullanılmıştır. Voronoi diyagramı literatürde Dirichlet, Thiessen veya Wigner-Seitz diyagramı olarak da anılmaktadır. Bu diyagram robotun yeni bir çevreye girdiği gibi oluşturulabilir. Yollardan oluşan yol haritası veya voronoi kenarları engel bölgesinde tüm noktalardan eşit uzaklıktadır. Şekil 3.12 de örnek bir voronoi diyagramı gösterilmiştir. Burada voronoi kenarlarının birleştiği noktalar üzerinden robot kesikli çizgiler ile gösterildiği başlangıç noktası ile hedef noktası arasında hareket etmektedir. Görünürlük grafiği tekniği ile kıyaslandığında engellerden uzaklık bakımından daha uygun bir tekniktir.



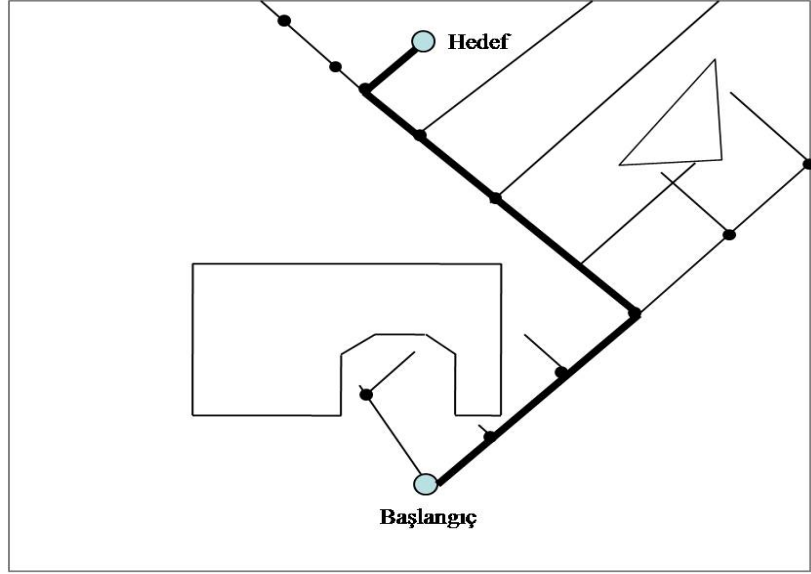
Şekil 3.12: Yol planlama için voronoi diyagramı örneği

Silüet tekniği, çalışma hücrelerinin silüetlerinin üretimi ile oluşur ve bu silüet eğrileri birbiri ile bağlanmasıyla yol haritası büyür ve oluşur. Silüet tekniği Canny (1987) tarafından rastgele veya isteğe göre boyutlarda bir yol haritası oluşturularak ortaya çıkmıştır. O daha küçük boyutlu bir uzayda bir nesneyi tasarlayıp ve daha sonra silüet olarak adlandırılan bir izdüşümü eğrilerinin krokisini oluşturmuştur. Bu, silüet eğrileri daha düşük boyutlu uzaya tek boyutlu çizgiler oluşuncaya kadar tekrarlanacak şekilde tasarlanmıştır. Sonra eğriler, bağlanmış eğrileri kullanarak yeni silüet eğrilerinin meydana geldiği veya ortadan kalktığı yerlerde bağlanırlar. Bu teknik, nesnelere içeriyor olan orijinal bir uzaydan daha düşük bir karmaşıklığa sahip tek boyutlu eğrilerin oluşturduğu bir grafikten bir yol bulmak üzere geliştirilmiştir. Bu teknik çoğunlukla, pratik algoritmaları geliştirmekten ziyade karmaşıklığı çözümlüyor olan teorik algoritmalarda kullanılırlar (Hwang ve Ahuja, 1992). Silüet eğrilerinden bulunan bir yol engel ve çalışma uzayı sınırları boyunca robota hareket imkânı verir (Şekil 3.13). Algoritmanın değişik bir versiyonu yine Canny ve Lin (1990) tarafından geliştirilmiştir. Bu çalışmalarında yine silüet tekniği ile birlikte potansiyel alanlar tekniğini de kullanarak yeni bir algoritma geliştirmişlerdir.



Şekil 3.13: Yol planlama için silüet eğrisi örneği

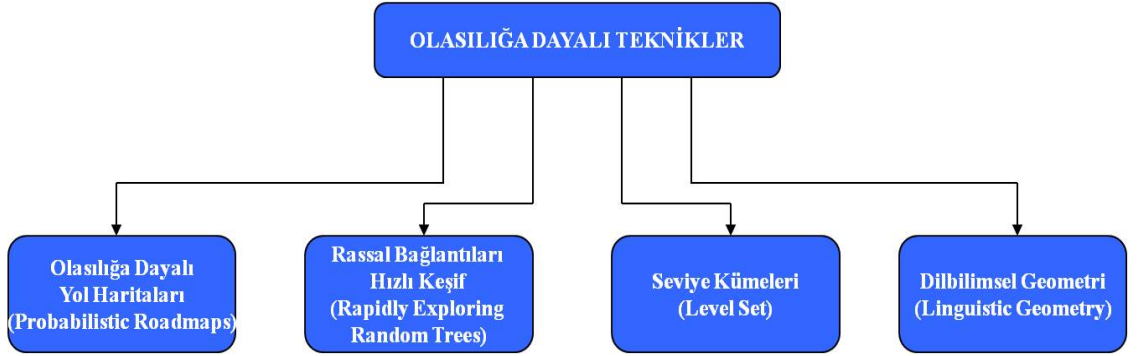
Alt hedef tekniği, konfigürasyon engellerinin net ve belirgin bir betimlemesini oluşturmaz. Onun yerine başlangıç konfigürasyonundan ulaşılabilir konfigürasyonların bir listesi oluşturulur. Hedef konfigürasyonuna ulaşıldığı anda yol planlama problemi çözülmüş olur. Bir konfigürasyondan diğerine ulaşılabilirliğe, oldukça basit bir hareket planlama algoritması olarak varsayılan bir lokal operatör sanki konfigürasyonlar arasındaki düz çizgide robotu hareket ettiriyormuş gibi karar verir. Başlangıçta bu teknik, alt hedefler olarak varsayılan arada bulunan konfigürasyonların bir aday serisini(ardışıklığını) bulur ve sırayla alt hedeflerden geçerek robotun hareketini sağlamak için lokal operatör kullanır. Bir keşifsel, genellikle ardışıklığı aramak için kullanılır fakat bunun yanında rastgele konfigürasyonların bir ardışıklığı da kullanılabilir. Eğer robot hedef konfigürasyonuna ulaşamıyorsa ulaşılan alt hedefler listede saklanır ve başka bir aday ardışıklığı hedef ile ulaşılan alt hedeflerin herhangi birisi arasında oluşturulur. Lokal operatör, ardışıklık aracılığıyla oluşabilecek hareketi kontrol etmek için tekrar kullanılır ve bu süreç defalarca tekrarlanır. Bu algoritmanın en önemli avantajı küçük hafıza gereksimi olmasıdır (Hwang ve Ahuja, 1992). Alt hedef tekniği ile ilgili bir yol planlama örneği Şekil 3.14 de görülmektedir.



Şekil 3.14: Yol planlama için alt hedef tekniği örneği

3.4.2. Olasılığa Dayalı Yol Planlama Teknikleri

Olasılığa dayalı yol planlama teknikleri, arama uzayında oluşturulan yolların ilk aşamada eşit seçilme olasılığına sahip olduğu ve sonuç olarak tüm bu arama uzayını temsil edecek şekilde bir örneklemelerin seçilmesiyle oluşur. Daha sonra bu uzay içindeki örneklenmiş noktalar birbiri ile bağlanır. Bu noktalar arasındaki bağlantılarda engellerde dikkate alınarak yol probleminin çözümü için bir harita oluşturulur. Olasılığa dayalı yol planlama teknikleri klasik yol planlama teknikleri sınıfına da girmektedir fakat işlevleri bakımından olasılığa dayalı yol planlama teknikleri olarak ifade edilmiştir. Örneğin; aşağıda açıklanacak olan olasılığa dayalı yol haritaları ve rassal bağlantıları hızlı keşif tekniği yukarıda bahsedilmiş olan klasik tekniklerde yol haritaları tekniği sınıfına girmektedir. Yinede olasılığa dayalı yol planlama teknikleri klasik tekniklerin verimliliğini geliştirmek için olasılığa dayalı teknikler geliştirilmiştir. Olasılığa dayalı teknikler Şekil 3.15 de görüldüğü gibi 4 ana teknik ile ifade edilebilir.



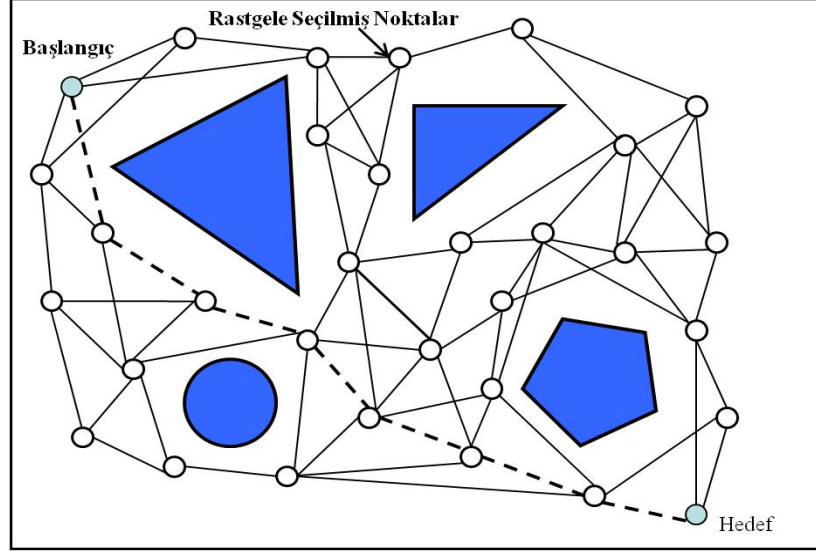
Şekil 3.15: Olasılığa dayalı yol planlama teknikleri

3.4.2.1. Olasılığa Dayalı Yol Haritaları Tekniği

Olasılığa dayalı yol haritaları tekniği genel olarak adından da anlaşılacağı gibi yol haritaları tekniğinin bir çeşidi olarak tanımlanabilir. Yine de yol planlamasının yapılacağı haritayı oluşturma aşamaları bakımından olasılığa dayalı yol planlama teknikleri arasında yer almaktadır. Olasılığa dayalı yol haritaları tekniği, son zamanlarda büyük ve geniş boyutlu uzaylarda yol planlama probleminin çözümü için kullanıp büyük bir ilerleme kaydetmiştir. Bu teknik, bir konfigürasyon uzayının rastgele olarak örneklenmesi ile meydana gelir ve daha sonra buradaki noktalar bir yol haritası içinde bağlanırlar. Olasılığa dayalı yol haritaları tekniği, yol planlamasının çözümü için

yapılan çalışmalarda büyük bir yol kat etmiştir çünkü diğer birçok planlama teknikleri özellikle de hücre ayırıştırma teknikleri planlama probleminin çözümünü bütün arama uzayında yaparlar. Bunun yanında olasılığa dayalı yol haritaları tekniği arama uzayının rastgele seçilmiş bir alt kümesinin oluşturmuş olduğu bir yol harita ile yol planlama problemini çözmeye çalışır. Olasılığa dayalı yol haritaları tekniği, noktaların ve kilometre taşlarının nispeten küçük bir sayısına dayanmaktadır ve yollar genellikle serbest uzayın bağlanırlığını elde etmek için yeterlidir. Bu varsayım da planlama problemlerinin çözümlerinin çoğunda süreci hızlandırabilir. Olasılığa dayalı yol haritaları tekniği yol haritasının oluşturması aşaması ve sorgu aşaması olmak üzere 2 aşamadan oluşmaktadır. Haritayı oluşturmak için konfigürasyon uzayında noktalar rastgele olarak seçilir ve özel noktaların bir listesine eklenir. Eğer bir nokta engel bölgesine bağlı ise o nokta iptal edilir. Haritalama algoritması bu noktayı daha önceden rastgele olarak seçilmiş özel noktaların listesinde var olan diğer konfigürasyonların bir alt kümesine bağlamaya çalışır. Bu bağlama işi iki nokta arasında eğer bir engel yoksa düz bir çizgi yardımıyla yapılır. Bu rastgele seçme işlemi ve bağlama işi değerlendirilerek defalarca tekrarlanır. Rastgele seçilmiş noktalar arasındaki uygun bağlantılarının bir listesi oluşturulur. İşte bu bağlantıların listesi, konfigürasyon uzayındaki engellerin etrafındaki yol güzergah noktalarının bir serisinin gösterildiği yol haritasını oluşturur. Sorgu aşamasında ise, eğer iki konfigürasyon arasında bir yol planlaması yapılacaksa, başlangıç ve bitiş konfigürasyonları arasındaki optimal yolu bulmak için yol güzergah nokta düğümlerinden geçerek arama yapmak için birinci aşamada oluşturulan harita kullanılır. İlk aşamada haritanın oluşturması çok masraflı ve pahalı olsa da bu harita oluşturduktan sonra arama çok verimli olur. Basit bir olasılığa dayalı yol haritaları tekniği örneği Şekil 3.16 de görülmektedir ve rastgele seçilmiş noktalara göre kesik çizgi ile gösterilmiş olan yol belirlenmiştir. Tüm bu açıklamalarla birlikte olasılığa dayalı yol planlama tekniğindeki en büyük sorun dar-sınırlandırılmış uzaylar için yetersiz ve verimsiz olmasıdır. Çünkü yol haritasını yapan noktalar rastgele seçilmiş olduğundan dar-sınırlandırılmış bir alanda rastgele oluşturulmuş bir noktayı yakalama şansı çok düşüktür ve haritanın bazı kısımları arasında istenilen bağlantılar kurulmayabilir. Buna çözüm olarak noktaların daha fazla sayıda olması ile noktaların daha fazla alanı kaplayacağı söylenebilir fakat noktaların sayısının artması yol planlama problemini kompleks hale getirir. Bu sorunu ortadan kaldırmak için Boor ve diğ. (1999) tarafından çalışmalar yapılmıştır. Olasılığa dayalı yol haritaları tekniğindeki diğer bir

sorun ise, engellerin haritadan çıkarıldığı veya eklendiği zaman ortaya çıkar ve bütün yol haritası bu durumda yeniden yapılandırılmalıdır. Çünkü yol haritasının oluşturulması yavaştır ve gerçek zamanda yapılamaz. Bu soruna bağlı olarak dinamik engel sınırlamalarını aşmak için Hsu ve diğ. (1999), Leven ve Hutchinson (2001) araştırmacılar tarafından birçok çalışmalar yapılmıştır.

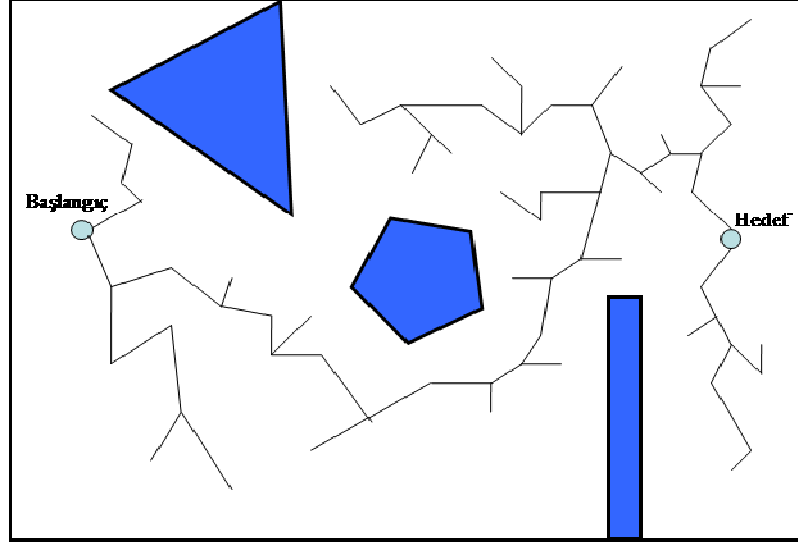


Şekil 3.16: Yol planlama için olasılığa dayalı yol haritası örneği

3.4.2.2. Rassal Bağlantıları Hızlı Keşif Tekniği

Rassal bağlantıları hızlı keşif tekniği, bir önceki teknik olan olasılığa dayalı yol haritaları tekniğinin birçok özelliğini içinde barındırdığından onun bir çeşidi olarak da tanımlanabilir. Bu teknikte olasılığa dayalı haritalar tekniğinde olduğu gibi daha az arama ile isteğe bağlı parametreler ile tasarlanmaktadır. Buda daha iyi bir performans ve daha tutarlı sonuçlar elde etmemizi sağlar. Bu tekniğin olasılığa dayalı yol haritaları metoduna göre en büyük avantajı kinodinamik planlama için direk olarak uygulanabilir olmasıdır. Olasılığa dayalı yol haritalarında olduğu gibi çözüm için birçok bağlantı gerekirken, bu teknikte konfigürasyon çiftleri arasında herhangi bir bağlantı yapılmasına gereksim yoktur. Yani rastgele olarak seçilmiş bir noktadan başlayarak bu noktaya en yakın noktaya hareket edip, daha sonra yine o noktaya yakın başka bir noktaya hareket edip hedefe ulaşmaya kadar dallanarak bir ağaç haline gelip bir yol haritası oluşturmaktadır. Bu dallanma henüz doldurulamayan alanlara doğru bağlı yolların genişlemesi ile devam eder. Bu yüzden bu teknikler ağaç tabanlı planlamaya dayalı bir tekniktir. Bu teknikle ilgili bir örnek Şekil 3.17 de gösterilmiştir. Bu teknikle ilgili

çalışmalar rastgele kinodinamik planlama adı altında Lavelle ve Kuffner (1999), Kuffner ve Lavelle (2000) Cheng ve diğ. (2001) ve Farazolli (2001) tarafından yapılmıştır.

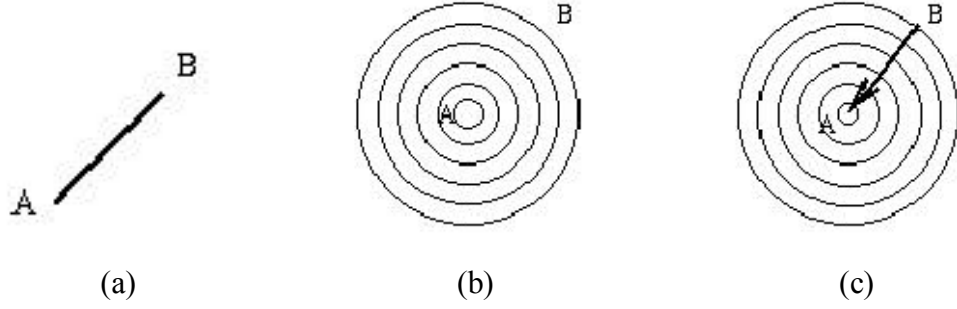


Şekil 3.17: Yol planlaması için rassal bağlantıları hızlı keşif örneği

3.4.2.3. Seviye Kümeleri Tekniği

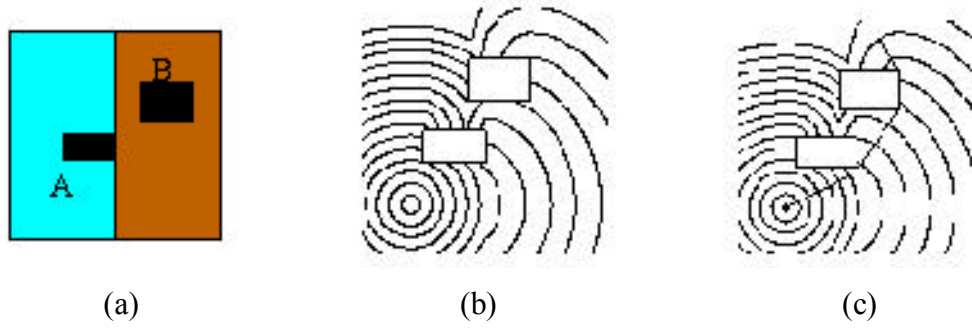
Seviye kümeleri tekniği, Osher ve Sethian (1988) tarafından önerilmiş kompleks hız kuralları altında yayılan ara yüzlerin hareketini izlemek ve topolojik değişimleri kontrol altında tutmak için kullanılan Hamilton-Jacobi denklemlerini çözmek için kullanılan verimli bir tekniktir. Bu teknikte hedef, hesaplama ve analiz etmek için bir çabukluk alanı altında ara yüzlerin bir sonraki hareketidir. Bu çabukluk ,konuma, zamana, ara yüzlerin geometrik şekline ve diğer fiziksel etkenlere bağlı olabilir. Özellikle son yıllarda Hassouna ve diğ. (2005) bu tekniği kullanarak yol planlama problemlerinin çözümü için birkaç algoritma geliştirmişlerdir. Bu teknikler mobil robotlar için bir alan içerisinde yol planlaması için engeller etrafında optimal yolları hesaplarlar. Bu teknik yardımıyla optimal yolların bulunması ile ilgili Sethian (1997) tarafından yapılan çalışma bir otopark örneği üzerinden çalışma yapılmıştır. Burada bir noktadan arabamızın olduğu yere gitmek için Şekil 3.18.a de görüldüğü gibi bir düz çizgi çizilmiştir. Burada A noktası bizim bulunduğumuz yer olurken B noktası da arabamızın olduğu yerdir. Buradaki ama A noktasından bütün yönlere doğru daire şeklinde yayılan bir yüzey oluşturmak olarak düşünülebilir. Burada yönlerin diğerleri üzerinden yürümesinin bize bir külfeti olmadığından, burada bütün yönlere 1 hızla yüzey yayılmasına izin verilir.

Bu yüzey yayılması A noktasından başlayıp daire şeklinde büyüyerek B noktasına temas edinceye kadar devam eder(Şekil.3.18.b). Bu yüzey B noktasına temas ettikten sonra, B noktasından başlayarak ve yüzey yayılmasına her zaman dik olan bir çizgi geriye doğru çizilerek optimal yol yaratılmış olur(Şekil.3.18.c).



Şekil 3.18: (a) A ile B noktaları arasının bir çizgi ile gösterilmesi.(b) A noktası etrafında yüzey yayılması.(c) A noktasından B noktasına yol bulmak için geri dönüşün izlenmesi (Sethian, 1997)

Yine diğer bir problemde otopark içersinde bir kısmı karlı,bir kısmın kuru olduğunu varsayarak ve bununla birlikte kendi arabasına giderken bir engel teşkil etmesini düşündüğü birkaç araba koyarak problemi kompleks hale getirmiştir. Yine burada da Şekil 3.19 de görüldüğü gibi A noktasından başlayarak tüm yönlere doğru bir yayılma gerçekleşmiştir fakat burada yayılma yüzeyleri karlı ve kuru yerlerde farklılık göstermiştir (Şekil 3.19.a). Mantık olarak düşünüldüğünde karlı yerde yürümek zor ve zaman açısından yavaş olduğundan buradaki yayılma farklı olmuştur (Şekil 3.19.b). Engel teşkil etmesi için koyulan arabaları sonsuz değerlere sahip yerler olarak tanımlamıştır, sebebi olarak da bir araba üzerinden geçmek imkânsız veya sonsuza kadar süreceğini düşünmesidir. Bu durumda Şekil 3.19.c de görüldüğü gibi yüzey keskin köşeler oluşturacaktır. Sonucunda da yine bu köşelerde dikkate alınarak yüzey yayılma problemin çözümü için geri dönüş izlenip, çizgilerle vasıtasıyla optimal yolu bulmuştur.

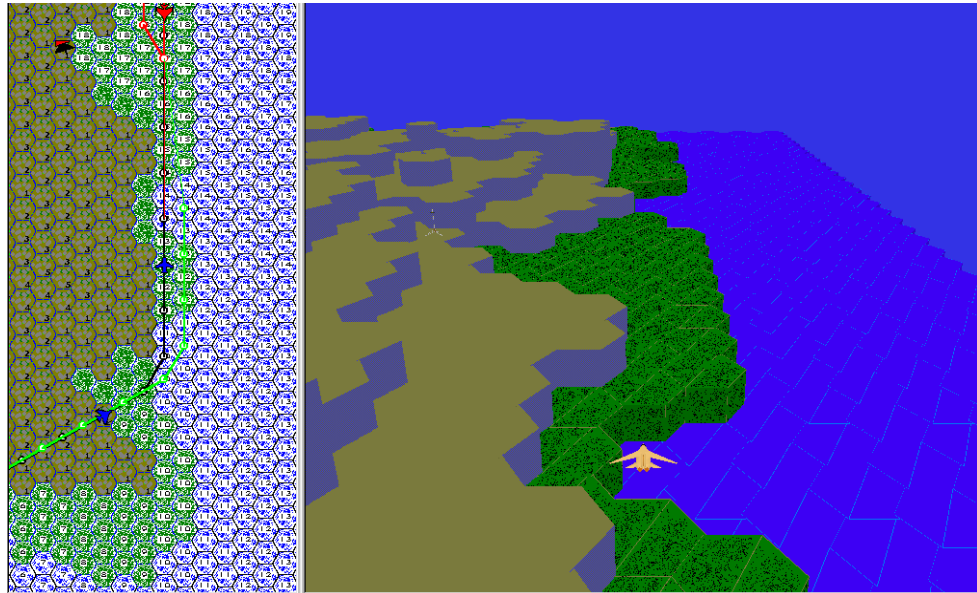


Şekil 3.19: (a) A bölgesinin karlı, B bölgesinin kuru olduğunu gösterilmesi.(b) A noktası etrafında yüzey yayılması.(c) A noktasından B noktasına yol bulmak için geri dönüşün izlenmesi (Sethian, 1997)

3.4.2.4. Dilbilimsel Geometri Tekniği

Dilbilimsel geometri, kompleks sistemlerin belirli bir sınıfı için geniş kapsamlı bir teknik olarak geliştirilmiştir. Bu teknik aslında bir oyun teoremidir ve bize pratik veya kompleks oyunların çözümü için kolaylıklar sağlar. Buradan yola çıkarak askeri karar mekanizmalarında, robotik uygulamalarında, yazılım mühendisliğinde, geleneksel eğlence oyununda uygulanabilir bir teknik haline gelmiştir. Dilbilimsel geometri ile ilgili özellikle Stilman (1993) tarafından yapılan birçok çalışmalar vardır ve bunları yol planlamasının da içinde olduğu kompleks sistemlere uygulamıştır. Dilbilimsel geometri tekniği, kompleks multi-ajan sistemler hakkında bilgi gösterimi ve ayrıca muhakeme etmek için kullanılabilecek sözdizimsel araçları içerir. Bu teknik, dinamik bir etkileşim içinde alt sistemlerin bir hiyerarşisi içinde kompleks sistemleri ayrıştırarak bu kompleks problemlerde aramayı azaltmak için bize güçlü araçlar sağlar. Bu araçlar, karmaşıklığın değerlendirilmesi ve çözümlerin kalitesi için bize bir bakış açısı oluşturur. Dilbilimsel geometrinin amacı çok büyük durum uzayları ile çeşitli problemlere çözümler sağlamaktır. Bu tür problemler, zeki veya insan yönlendirmesi ile hareket eden uçak, uzay araçları, kara ve deniz araçları gibi robot takımlarının ortak çalışmalarını veya yarışmalarını, gezgin keşif araçları gibi hareket eden uzaktan tam otomatik nesnelere için güvenli-kritik kontrol sistemlerini planlamayı, programlamayı ve satranç oyunu ile ilgili problemleri içerir. Genel olarak, yukarıdaki sistemlerde tek-bağımsız nesnelere optimal veya optimala yakın davranışlarını bulmak için dev arama ağaçlarındaki uygun dallarda arama yapılması gerekmektedir. Bu tür aramalar çoğunlukla karışık olup, modern bilgisayarların bile ötesindedir. Dilbilimsel geometri tekniği, ciddi şekilde bu arama ağaçlarının boyutlarını azaltarak problemleri sayısal olarak çözümler hale getirir. Bu

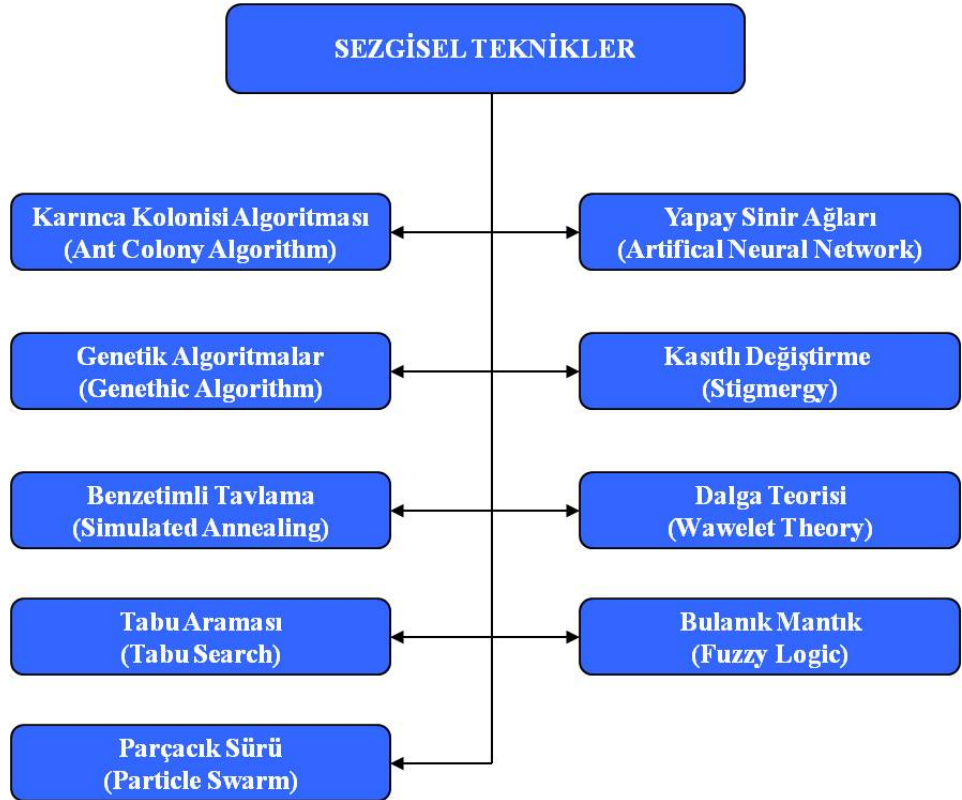
teknikğin önemli özelliklerinden birisi biçimlendirilmesidir ve çok iyi becerilere sahip uzman kişiler tarafından(satranç uzmanı) tarafından arama keşiflerinin kullanımı geliştirilmiştir. Bu uzmanlar kendi alanlarında aramaları olağanüstü bir şekilde azaltarak başarılı stratejiler geliştirmişlerdir. Yukarıda yazılanlardan yola çıkarak yol planlaması yapılacak çevreler dilimsel geometride bir oyun tahtası gibi düşünülmektedir. Dilbilimsel Geometri tekniği hareketleri tanımlıyor olan soyut ilişkilere veya bağlantılara ek olarak oyun tahtasının geometrisine başvurur. Buradaki soyut ilişkiler yukarıda yazılan uçaklar, füzeler, gemiler, tanklar, yangın ekipleri v.b. gibi savaş veya benzeri araçların hareketini ve onların kendi aralarında haberleşmesini ifade eder. Şekil 3.20 de görüldüğü gibi bir çeşit uçağın yol(rota) planlaması istenilen stratejilere göre yapılmıştır. Strateji gerektiren yerlerde kullanılan dilbilimsel geometri bir sistemin bir durumdan başka bir duruma hareket etmesi ile çalışır. Bu hareketi örneğin satranç oyununda oyuncu sağlarken, bir savaş uçağının yol planlaması için üst düzey kişilerin hareketidir. Bu hareket, bir noktadan başka bir nokta tarafında olan oyuncunu bir ögesinin yer değişmesi olur ve sistemin bir durumdan başka bir duruma bir geçiş yapmasına sebep olur.



Şekil 3.20: Bir uçağın dilbilimsel geometri tekniği ile yol(rota) planlaması (Stilman, 2009)

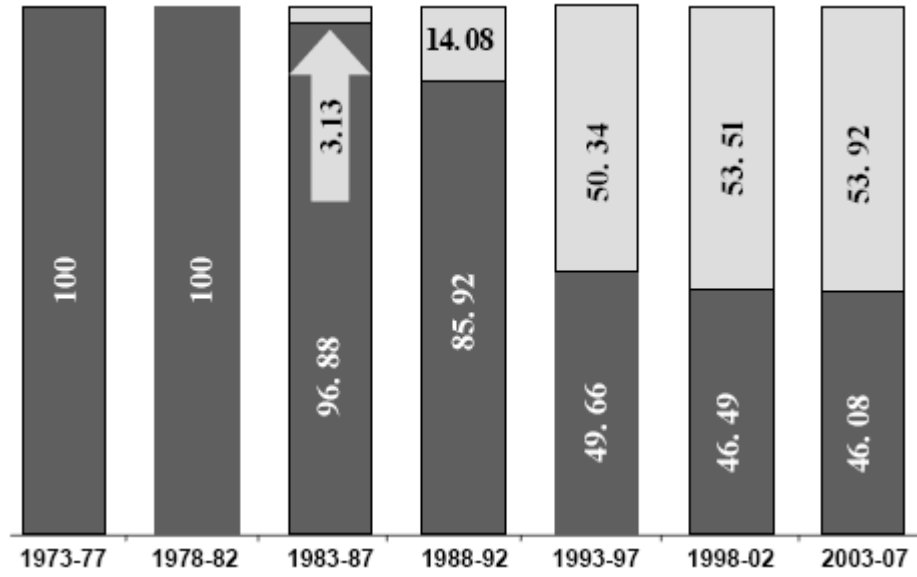
3.4.3. Sezgisel Yol Planlama Teknikleri

Sezgisel teknikler genel kullanım amaçları doğrultusunda klasik yöntemlerle çoğu zaman çözülemeyen ve NP-Hard gibi sonuca ulaşma zamanı üstel olarak artan ve karmaşık olan problemlerin çözümlerine etkili ve hızlı bir şekilde iyi çözüm veren yöntemlerdir. Sezgisel teknikler özellikle başta mühendislik olmak üzere neredeyse her alanda problem çözümleri için kullanılmaya başlanmıştır. Özellikle son yıllarda robotik alanında sezgisel teknikler ile yapılan birçok çalışma mevcuttur. Bu çalışmalardan birisi de mobil robotlar için yol planlama veya hareket planlama problemleridir. Sezgisel yöntemlerin mobil robotlar için yol planlamasının problemlerinin çözümünde kullanılmasının sebepleri yukarıda açıklanmış olan özellikle klasik tekniklerde olmak üzere bu tekniklerde ortaya çıkan sorunlardır. Tüm bunları göz önüne alarak araştırmacılar mobil robotlar için yol planlama veya hareket planlama problemlerinin çözümü için Şekil 3.21 de görüldüğü gibi birçok sezgisel tekniği kullanmışlardır.



Şekil 3.21: Sezgisel yol planlama teknikleri

Sezgisel tekniklerin kullanımı ile klasik ve bazı olasılığa dayalı yol planlama teknikleri yapılan çalışmalarda hatalı ve yüksek hesaplama değerleri yüzünden yetersizliği ortadan kaldırılması amaçlanmıştır. Bununla birlikte özellikle potansiyel alanlar tekniğinde olduğu gibi yerel değerlerde planlamayı doğrudan kestikleri görülmüştür. Belirtilen bu sorunlarda kurtulmak için yine bu sezgisel teknikler kullanılmıştır (Brand ve diğ. 2010). Masehian ve Sedighizadeh (2007) robotların yol ve hareket planlaması için kullanılan tekniklerle ilgili yaptıkları çalışmada bu konuyla ilgili 1973 yılından 2007 yılına kadar yapılmış 1381 çalışmayı incelemiştir. Bu incelemelerinin sonucunda 1973 yılından 1982 yılına kadar yaklaşık 10 yıl içerisinde yapılan çalışmaların hepsinde klasik tekniklerin kullanıldığı görülmüştür. 1983 yılı ve sonrasında sezgisel tekniklerin kullanımına başlanmış olup 2007 yılına kadar daha çok kullanılmış ve belirli oranlarda artış göstermiştir. Şekil 3.22 de bakıldığında koyu renkle klasik teknikler görülmekle birlikte, açık renk ile sezgisel teknikler ifade edilmiştir. Buradan sezgisel tekniklerin 15 yıl içerisinde klasik tekniklere oranla daha çok kullanılmaya başlandığı görülmüştür.



Şekil 3.22: Yol planlama tekniklerinin yıllara göre dağılımı (Masehian ve Sedighizadeh, 2007)

Yine bu 1381 çalışmanın incelemesi sonucunda ulaşılan verilerden biriside Çizelge 3.2 de görüldüğü gibi mobil robotlar için yol planlama veya hareket planlaması çözümü amacıyla kullanılan sezgisel tekniklerin kullanım oranlarıdır.

Çizelge 3.2 : Mobil robotların yol planlama problemlerinin çözümü için kullanılan sezgisel tekniklerin yıllara göre kullanım oranları (Masehian ve Sedighzadeh, 2007)

Teknik	83-87	88-92	93-97	98-02	03-07	Toplam
Yapay Sinir Ağları	0	60	36	44.26	25.53	34.58
Genetik Algoritma	0	20	34.67	32.24	37.23	34.54
Benzetimli Tavlama	0	5	4	1.09	2.13	2.35
Parçacık Sürü	0	0	0	0.55	4.26	2.04
Karınca Kolonisi	0	0	1.33	3.83	10.64	6.12
Kasıtlı Değiştirme	0	0	1.33	1.64	3.55	2.35
Dalga Teorisi	0	5	0.67	1.64	1.77	1.57
Bulanık Mantık	100	10	22	14.75	14.89	16.64
Toplam	100	100	100	100	100	100

Çizelge 3.2 ye bakıldığında yol planlaması için tezde kullanmış olduğumuz karınca kolonisi algoritması ile ilgili çalışmalar 1993 yılından itibaren günümüze belli oranlarda kadar artarak gelmektedir. Bu oran burada sezgisel teknikler ile yapılan çalışmalar arasında %6.12 olarak gözükmemektedir. Bunun yanında karınca kolonisi algoritması ile ilgili yapılan çalışmalar tüm tekniklerin arasında (klasik, olasılığa dayalı ve sezgisel) %2.83'lük bir orana sahiptir. Buradan da anlaşılacağı üzere her alanda olduğu gibi robotik alanında da karınca kolonisi algoritmasının kullanımı her geçen gün artmaktadır. Mobil robotların yol planlaması için kullanılan karınca kolonisi algoritması ile ilgili tüm bilgiler bir sonraki bölümde detaylı bir şekilde aktarılacaktır.

4. KARINCA KOLONİSİ OPTİMİZASYONU

Son yıllarda arařtırmacılar klasik tekniklerin çözemediđi bařta NP-Hard problemler olmak üzere birçok gerçek problemi sezgisel tekniklerin yardımıyla çözmeye çalıřmıřlardır. Bu sezgisel tekniklerin oluřturulmasında doğada yařayan canlılar ve onların davranıřları arařtırmacılara ve bilim adamlarına ilham kaynađı olmuřtur. Bu canlıların bařında karıncalar, arılar, balıklar ve kuřlar gelmektedir. Normalde basit gözükten doğadaki bu canlıların davranıřı arařtırılıp incelenirken en çokta bu canlıların hep birlikte iřbirliđine dayalı yapmıř olduđu davranıřlar ve davranıřları yaparken aralarındaki haberleřmenin nasıl olduđu konusu üzerinde durulmuřtur. Canlıların bu iřbirliđine dayalı yapmıř olduđu davranıřlar arařtırmacılar ve bilim adamları tarafından sürü zekâsı adı altında tanımlanmaktadır. Sürü zekâsı, canlıların tek bir canlı olarak buldukları çözümlerin ötesinde hepsinin iřbirliđine dayalı ortaya çıkarmıř oldukları çözüme dayanmaktadır ve yapay zekâyâ farklı bir bakıř açısı getirmiřtir. Yıllar geçtikçe canlıların sürü zekâsına dayanan davranıřları modellenip-formüle edilerek algoritma haline getirilmiřlerdir. Özellikle de son yıllarda bu algoritmalar sezgisel teknikler adı altında robotik, mühendislik, bilgisayar ve birçok alanda en iyi çözüme yakın çözümler üretmede bařarılı olduđu görölmüřtür.

Bu bölümde sezgisel tekniklerin bařında gelen ve tez çalıřmasında da kullanılmıř olan Karınca Kolonisi Optimizasyonu (KKO) gerçek karıncalara bađlı olarak gelişim evresi, algoritmanın temel özellikleri, diđer karınca kolonisi algoritması ile bunların farklı alanlara göre kullanımları ve KKO'nun mobil robotlar için yol planlamasına uygulaması hakkında detaylı bir bilgi verilmiřtir.

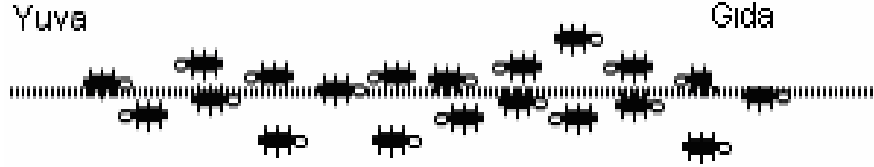
4.1. GERÇEK KARINCA DAVRANIŞLARI

Doğal yaşamda karıncalar koloni halinde yaşayan ve tek başlarına çözümler üretme açısından son derece basit canlılardır. Kendi yuvaları ile araştırma sonucunda buldukları yiyecek arasında belli bir süre içerisinde en kısa yolu bulma yeteneğine sahip canlılardır. Gerçek karıncalar ile ilgili deneyler Goss ve arkadaşları (1989) tarafından laboratuvar ortamında yetiştirilmiş arjantin karınca kolonileri üzerinde yapılmıştır. Bu çalışmanın sonuçlarına göre karıncaların görme yetisinin olmadığı, yuva ile buldukları yiyecek arasında yuvadan-gıdaya veya gıdadan-yuvaya gittiklerinde feromen (pheromone) denilen kimyasal bir madde salgıladıklarını ve karıncaların böylece geçtikleri yollarda bir feromen izi oluşturduklarını görmüşlerdir. Karıncaların yiyeceğe giderken bir yol seçmesi gerektiğinde, yuva ile yiyecek arasındaki tüm yollardaki feromen izinin miktarlarına göre seçim yapıldığı ve karıncaların bu davranışlarını merkezi bir kontrol sağlamadığı gözlenmiştir.

İzdeki feromen maddesi ne kadar yüksek olursa, karıncanın o izi takip etme olasılığı o kadar yüksektir. Bu kimyasal mesaj karıncaların başarısının anahtar etkenidir. Bu yapı kullanışlıdır, çünkü en iyi kararları veren karıncalar yuvaya ilk dönenler olacağından, geçtikleri yollara, henüz yiyecek aramakla meşgul diğer karıncalara göre iki kat daha fazla feromen bırakmış olurlar. Karıncalar yol seçiminde bir olasılık dağılımını kullandıklarından bir karıncanın keşfedilememiş bir yolu seçme ihtimali az da olsa vardır. Bu lokal bir optimum çözüme karşı bir güven unsurudur. Lokal olarak oluşabilecek bir çözüme kilitleyerek daha iyi çözümlerin bulunamaması ihtimali azaltılmalıdır (Öztürk, 2006).

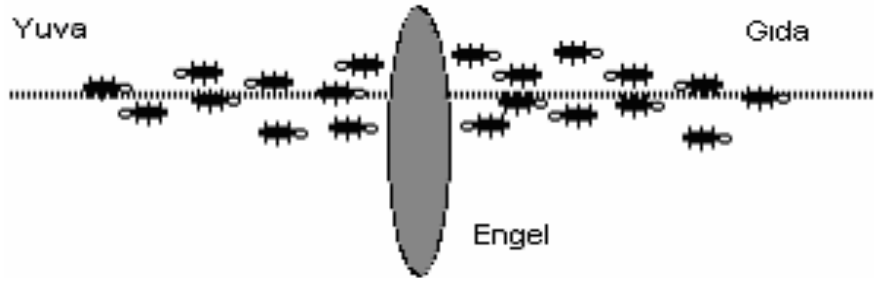
Karıncalar görme yeteneğine sahip olmamalarından dolayı çevrelerinde olup bittiğini göremeyen canlılardır. Buna rağmen yukarıda da bahsedilen feromen maddesi miktarına bağlı olarak kısa süre içerisinde çevredeki değişikliklere uyum sağlayan ve davranışlarını çevre şartlarına göre ayarlayan canlılardır. Örneğin, karıncalar koloni halinde yuva ile yiyecek arasında en kısa yol üzerinde ilerlerken veya bu yolu araştırdıklarında o yol üzerinde bir engel ortaya çıkması sonucunda karıncalar kısa bir süre içerisinde bu engellere göre davranışlarını değiştirerek yuva ile gıda arasında en kısa yolu bulurlar. Şekil 4.1 de görüldüğü gibi belli bir süre içerisinde yuva ile gıda arasında kolonideki tüm

karıncalar tarafından en kısa yolun bulunmuş olup karıncalar bu güzergah üzerinde hareketlerini gerçekleştirdiği durumu inceleyelim.



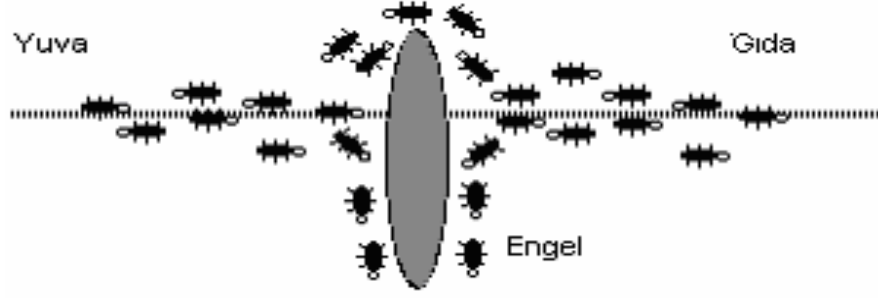
Şekil 4.1: Yuva ile gıda arasındaki en kısa yol

Şekil 4.1. de gösterildiği gibi karıncaların düz bir yolda giderken bu yola Şekil 4.2 de görüldüğü gibi bir engel konularak o yolun kısa olmaktan çıktığını varsayalım.



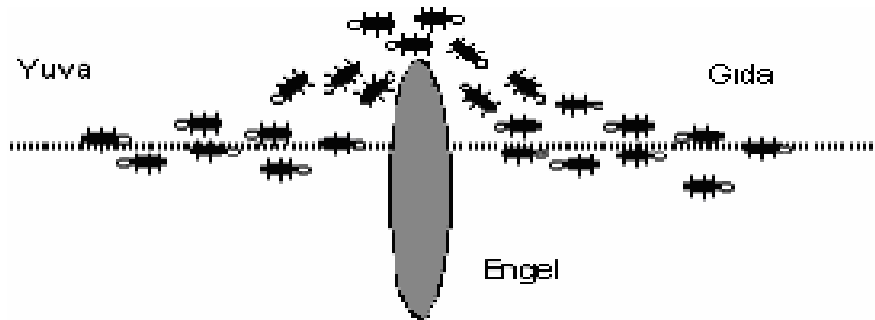
Şekil 4.2: Karıncaların bir engelle karşılaştığında en kısa yolun bozulması

Bu durumda engelin önündeki ve arasındaki karıncalar bu engel üzerindeki yollar daha önce kullanılmadığından feromon olmayan yönlerle karşılaşır. Bundan dolayı da gıdaya gidebilmeleri için yeni bir gidiş yönü seçmek zorunda kalacaklardır. Bu durumda karıncaların sağa ve sola doğru gitme olasılıkları eşittir. Karıncalar ilk etapta rastgele olarak seçim yaparak bir tarafa doğru ilerlerler. Şekil 4.3. de görüldüğü gibi sağ ve sola doğru giden karıncaların bu durumda sayısı birbirine çok yakın olmaktadır.



Şekil 4.3: Karıncaların engelle karşılaşması sonucundaki durumları

Burada karıncaların eşit olasılıkla seçmiş oldukları yolun gıdaya giden en kısa yol olmaması durumunda karıncalar yönlerini hızlı bir şekilde yeniden değiştirirler. Bunun sebebini açıklarsak kolonideki tüm karıncaların geçmiş oldukları yola ortalama olarak aynı hızda ve aynı miktarda feromen maddesi salgıladığını varsayarsak kısa yol üzerinden geçen karıncaların sayısı fazla olacaktır ve dolayısıyla bu yoldaki feromen maddesi miktarı karınca sayısına bağlı olarak birim zamanda artacaktır. Bu yolda feromen maddesinin artması ile birlikte sonradan gelen karıncalar için cazip hale gelip seçilme olasılığı artacaktır. Böylece kolonideki tüm karıncalar kısa bir süre sonra bu yol üzerinden gıdaya gitmeye başlayacaklardır ve böylece engelin konulması ile değişen çevre şartlarına uyum sağlayacaklardır. Burada feromen maddesine bağlı olarak bir pozitif geri beslemeden söz edilebilir. Yani bir yoldan geçen karınca sayısı arttıkça yoldaki feromen maddesi artacak, feromen maddesi arttıkça da sonradan gelen karıncalardan bu yolu tercih edenlerin sayısı artacaktır. Dolayısıyla belli bir süre sonunda kolonideki karıncaların tümü Şekil 4.4 de görüldüğü gibi gıdaya giden en kısa bulacaklardır.



Şekil 4.4: Karıncaların belli bir süre sonunda kısa yolu bulmaları

4.2. KARINCA KOLONİSİ ALGORİTMASI

Karınca kolonisi algoritması (KKA) ile ilgili ilk çalışma Dorigo ve arkadaşları tarafından 1991 yılında çalışmalara başlanarak 1992 yılında Dorigo tarafından doktora tezi olarak gerçekleştirilip, yayımlanmıştır. Dorigo bu çalışmasında Goss ve diğ. (1989) gerçek karıncaların davranışları ile ilgili çalışmalarının sonucundan yola çıkarak matematiksel formüller üzerine bir modelleme yapmıştır. Bu oluşturduğu sisteme Karınca Sistemi (Ant System), ortaya çıkan algoritmaya da Karınca Sistemi Algoritması (Ant System Algorithm) olarak adlandırmıştır. Bununla birlikte bu algoritmanın gerçek hayat problemlerine uygulanması ve daha verimli sonuçlar alınması için gerçek karıncaların temel özelliklerinin yanı sıra algoritmada kullanılacak olan yapay karıncalara aşağıda maddeler halinde belirtilen bazı özellikler eklenmiştir.

Gerçek karıncaların aynen alınan özellikleri:

- Karıncalar arasında feromen aracılığı ile kurulan iletişim.
- Feromen miktarının fazla olduğu yolların öncelikle tercih edilmesi.
- Kısa yollar üzerinde feromen miktarının daha hızlı artması.

Eklenen özellikler:

- Zamanın ayırık olarak hesaplandığı bir ortamda yaşarlar ve hareketleri ayırık durumlardan ayırık durumlara geçişleri içerir.
- Tamamen kör olmayıp, problem ile ilgili detaylara erişebilirler.
- Belli bir miktar hafıza ile problemin çözümü için oluşturdukları bilgileri tutabilirler.

Karınca tabanlı algoritmalarda temel fikir, basit iletişim mekanizmalarını kullanan yapay akıllı araçların (agent), birçok karmaşık problem için çözümler üretebilmesidir (Dalkılıç ve Türkmen, 2002).

Dorigo, geliştirdiği karınca sistemi algoritmasını ilk olarak değişik boyutlardaki gezgin satıcı problemlerine (GSP) uygulamıştır. GSP, bir satıcının herhangi bir noktadan başlayarak her noktaya bir kez uğramak koşuluyla optimum yolları kullanarak dolaşım tekrar başlamış olduğu noktaya dönmesini amaçlayan bir problemdir. GSP tamamı ağırlıklaştırılmış $G=(N,A)$ ile tanımlanabilir. Burada N şehirler(noktalar) kümesini, A

ise noktaları birbirlerine bağlayan arkları göstermektedir. Her bir ark $(i,j) \in A$ olarak tanımlanabilir. Burada (i,j) iki noktaları arasındaki uzunluğu, yani d_{ij} 'yi ifade etmektedir. GSP diğer yandan G grafi üzerinde her bir noktadan bir kez geçmek suretiyle yine başladığı noktaya gelen bir Hamilton turu üstünde minimum bir uzunluğu bulan problemdir (Dorigo ve Stützle, 2004).

4.2.1. Karınca Sistemi

Literatürde karınca-yoğunluk, karınca-miktar ve karınca-çevrim olmak üzere karınca sistemi için 3 farklı versiyon önerilmiştir. Bunların arasındaki fark ise feromen depolamasında yatmaktadır. Karınca-çevrim algoritmasında feromen sadece her karınca turunun sonunda o karıncanın turu üzerindeki her kenar, uzunluklarıyla ters orantılı olarak feromenle depolanmaktadır. Bununla birlikte her sanal karınca gerçeğinin aksine geçtiği yolları hatırlamak için bir hafızaya sahiptir. Diğer iki algoritmada ise feromen yolu, karınca bir şehirden (noktadan) diğer bir noktaya geçerken güncellenmektedir. Yapılan testler sonunda Karınca-çevrim algoritmasının daha iyi sonuçlar verdiği ortaya çıkmış ve genel bir yapı kazanmış ve karınca sistem algoritması denildiğinde bu algoritma kastedilir hale gelmiştir (Uğur ve Aydın, 2006).

Karınca sistemi algoritması genel adımları ile Şekil 4.5 de görüldüğü gibi işlemektedir.

1. **Adım:** Gerekli parametreler için başlangıç değerleri belirlenir.
2. **Adım:** Bütün karıncaları şehirlere rastgele olarak yerleştirilir.
3. **Adım:** Karıncaları buldukları şehirden olasılık formülüne göre başka bir şehre hareket etmesi için seç.
4. **Adım:** Her karınca turunu tamamladıktan sonra yolların uzunluğu hesaplanır ve en iyi değere göre feromen güncellemesi yapılır.
5. **Adım:** İterasyon sayısına veya başka bir kriteri sağlayıncaya kadar tabu listelerini boşalt ve 2. Adıma geri dön

Şekil 4.5: Karınca sistemi algoritması işleyişi

4.2.1.1. Algoritmanın Başlaması ve Değerlerin Atanması

Karınca Sistemi algoritması Şekil 4.5 de görüldüğü ilk olarak başlangıç parametre değerlerinin belirlenmesi ile başlamaktadır. Bu parametre değerleri problemin çözümü sırasında birçok nedenden dolayı performansını etkilemektedir, bu yüzden bu parametre değerlerinin iyi bir şekilde seçilip belirlenmesi gerekmektedir. Parametre değerlerinin bazıları araştırmacılar tarafından yapılan çalışmalar sonucunda belirli değerler aldıkça da bazı problemlerde bu değerler değişebilmektedir.

Bu parametre değerlerinden ilk akla gelen karıncaların sayısıdır. Karınca sayısının fazla olması daha iyi sonuçlar elde etmemize rağmen, algoritmanın çalışma zamanının artmasına neden olur. Bu yüzden Dorigo ve diğ. (1991) gezgin satıcı problemlerinin çözümü için yaptıkları deneylerin sonucunda karınca sayısının şehir sayısına eşit seçilmesinin uygun olacağı görülmüştür. Kolonideki toplam karınca sayısı m , şehir sayısı ise n ile ifade edilirse parametrelerin belirlenmesinde $m=n$ eşitliğinin sağlanması gerekmektedir. Bu iki büyüklük arasındaki bu ilişkiden dolayı da Denklem 4.1 deki eşitlik çıkmaktadır.

$$m = \sum_{i=1}^n b_i \quad (4.1)$$

Burada b_i i. Şehirdeki karınca sayısını ifade ederse buradan yola çıkarak şehir sayısına bağlı olarak toplam karınca sayısı elde edilebilir.

Karınca sisteminde diğer bir önemli parametrede ilk feromon değeridir. Feromon maddesi değeri yukarıda açıklanan tanımlamalardan anlaşılacağı üzere algoritmadaki yapay karıncaların kendi aralarında paralelliği sağlayan önemli parametrelerden birisidir. Dolayısıyla karıncaların bir noktadan diğer bir noktaya geçmeleri için gereken olasılık değerlerinin oluşturulması için algoritmanın başında bir feromon matrisi oluşturulur. Bu feromon matrisinin boyutu karınca ve şehir sayısına bağlıdır. Matris i ve j noktaları arasında depolanan feromon maddesi miktarını gösterir ve τ_{ij} şeklinde tanımlanır. Feromon matrisi başlangıcı için Dorigo ve Stützle (2004) tarafından önerilen başlangıç değeri $\forall(i, j)$ 'ler için Denklem 4.2 de verilmiştir. Burada feromon matrisindeki tüm yolların aynı değeri yani τ_0 başlangıç değerini aldığı görülmektedir.

$$\tau_{ij} = \tau_0 = m/C^{nn} \quad (4.2)$$

Burada m yine toplam karınca sayısını ifade ederken, C^{nn} değeri ise en yakın komşuluk sezgisiyle oluşturulan tur uzunluğunu ifade eder. Eğer ilk feromen değeri olan τ_0 değeri aşırı derecede düşük olsaydı, arama karıncaların ilk turlarında hızlı bir şekilde peşin hükümlü hale gelecektir. Diğer yandan, ilk feromen değeri aşırı derecede yüksek olsaydı birkaç turdan sonra feroman buharlaşması yapılmıncaya kadar farklı yolların aynı turlar seçileceğinden bir kayıp meydana gelecektir. Buradan yola çıkarak karınca sistemi algoritmasının başlangıçta dikkat edilerek alınması gereken bir parametre değeri de feromen buharlaşmasıdır.

Feromen buharlaşması olmadan karınca sistemi algoritması problemlerin çözümünde doğru sonuçlar vermez. Çünkü arama uzayındaki ilk keşifler tamamen rastgele olduğundan, bu aşamada bırakılan feromenler herhangi bir bilgi içermezler. Dolayısıyla diğer karıncaların bu işe yaramayan değerleri kolayca unutması ve iyi çözümler üzerinden yol olabilmesi için buharlaşma mekanizması gereklidir (Uğur ve Aydın, 2006). Buharlaşma katsayısı ρ ile gösterilmektedir ve bu değer 0 ile 1 arasındaki değerler ile belirlenir. Dorigo (1992) yapmış olduğu çalışmalar sonucunda bu parametreye en uygun değerinin 0,5 olduğunu belirlemiştir.

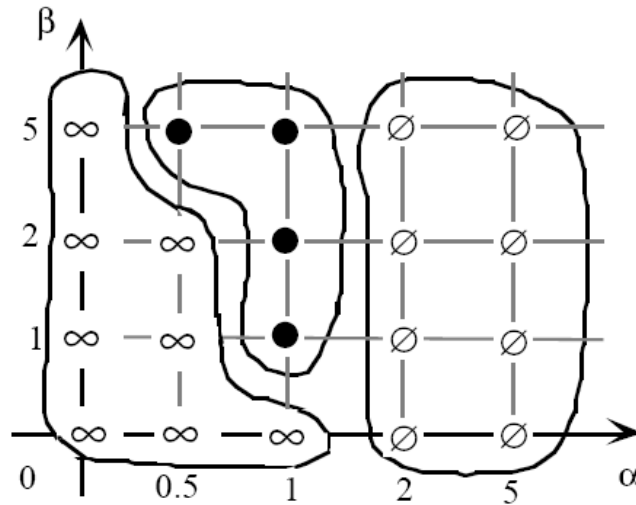
Algoritmanın başlangıçta belirlenmesi gereken en önemli parametrelerinden birisi de seçilebilirlik parametresidir. Seçilebilirlik parametresi i şehirden j şehrine hareketin sezgisel değerini ifade etmektedir. Bu seçilebilirlik parametresi olasılık değeri denkleminde sadece feromen değerine bağlı olarak seçim yapmak yerine, problemdeki tüm yollar arasındaki mesafede de göz önüne alınmaktadır. Seçilebilirlik parametresi problemdeki şehirler statik olduğundan algoritma boyunca sabittir ve algoritmanın başında bir seçilebilirlik parametre matrisi oluşturulur. Şehirlere bağlı olarak oluşturulan bu seçilebilirlik parametresi Denklem 4.3 de gösterildiği gibi bulunmaktadır.

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (4.3)$$

Burada d_{ij} i ve j şehirleri arasındaki mesafeyi göstermektedir. Bu mesafeler yine algoritmanın başında Denklem 4.4 de olduğu gibi Öklid (euclid) bağıntısına bağlı olarak hesaplanır.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (4.4)$$

Karıncaların bir şehirden diğer bir şehre gitmelerini belirleyen olasılık denklemi için en önemli parametre değerleri aslında α ve β değerleridir. α değeri i ve j şehirleri arasında feromen miktarının önemini belirlerken, β değeri i ve j noktaları arasındaki yol uzunluğunun önemini belirlemektedir. Dorigo (1992) yapmış olduğu çalışmalar sonucunda bu parametreye en uygun değerinin α için 1, β içinde 5 olduğunu belirlemiştir. Yine Dorigo ve diğ.(1996) yaptıkları çalışmada α ve β değerleri için Şekil 4.6 da görüldüğü gibi parametre seçim grafiğini elde etmişlerdir.



Şekil 4.6: α ve β değerleri için seçim aralıkları grafiği (Dorigo ve diğ. 1996)

Burada (●) ile gösterilen kısımda algoritma durgunluğa girmeden en iyi bilinen sonucu bulur. (∞) ile gösterilen kısımda algoritma durgunluk davranışına girmeden iyi çözümleri bulamaz. (∅) ile gösterilen kısımda ise algoritma iyi çözümleri bulamaz ve durgunluk davranışına girer (Dorigo ve diğ. 1996).

Algoritmanın başlangıcında belirtilmesi gereken bir diğer parametre de karıncanın tur(iterasyon) sayısıdır. İterasyon sayısı algoritmadaki tüm karıncaların aynı turu oluşturulmasına ve belirlenen bir maksimum tur sayısı NC_{max} değerine göre belirlenir.

4.2.1.2. Karıncaların Turunun(Hareketinin) Oluşturulması

Yukarıda bahsedilmiş olan tüm parametreleri ilk başta algoritma için belirlendikten sonra algoritmadaki m tane yapay karınca n tane şehre rastgele olarak yerleştirilerek algoritma çalıştırılmaya başlanır. Yapay karıncalar her turlarında bir noktadan sonraki adımda hangi noktaya hareket edeceklerini olasılığa dayalı oransal bir formül ile bulurlar. Buna göre, k. karıncanın i şehirden j şehrine hareketi için gerekli olasılık formülü Denklem 4.5 deki gibidir.

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha \cdot [\eta_{il}]^\beta} & \text{eğer } j \in N_i^k \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.5)$$

Burada;

τ_{ij} : i ve j noktaları arasındaki feromen maddesi miktarı.

η_{ij} : i noktasından j noktasına hareketin sezgisel değeri.

α : Feromen izi miktarını kontrol eden parametre.

β : Seçilebilirlik(sezgisel) parametresini kontrol eden parametre.

N : Şehir(düğüm) kümesi veya sayısı.

N_i^k : k. Karıncanın henüz ziyaret etmediği şehirlerin seti.

Denklem 4.5 deki olasılık formülünden görüldüğü gibi belirlenen parametrelerin özellikle α ve β değerlerinin seçiminin önemi görülmektedir.

Eğer, $\alpha = 0$ olursa en yakın şehirlerin seçilmiş olması daha olasıdır ve bu klasik stokastik bir aç gözlü algoritmaya karşılık gelmektedir. Eğer $\beta=0$ olursa sezgisel etki olmadan sadece feromen kullanılır ve feromen yükselmesi devam eder (Dorigo ve Stützle, 2004). Diğer yandan α değerinin yüksek olması feromenin yoğun olduğu yolların seçilme olasılığını arttırırken tesadüflüğü azaltmaktadır. Aynı şekilde β değeri arttıkça bir sonraki yolun seçiminde tesadüflük artmaktadır (Dorigo, 1996).

Bununla birlikte her k karıncası, daha önce önceden ziyaret ettikleri şehirleri sırayla içeren bir M^k hafızasına sahiptir. Bu hafıza, denklem 4.5 de belirlenmiş kural yapısına göre, olası gidilebilecek N_i^k komşularını tanımlamak için kullanılır. Ek olarak, M^k hafızası k. karıncanın oluşturduğu turu uzunluğunun hesaplamasına ve feromen koymak için yolu yeniden gözden geçirmeye izin verir (Dorigo ve Stützle, 2004).

4.2.1.3. Feromen İzlerinin Güncelleştirilmesi

Bütün karıncalar turlarını oluşturduktan sonra, feromen izleri güncellenir. Bu ilk olarak, sabit bir faktör ile bütün yollar üzerindeki feromen değerinin düşürülmesi ile yapılır ve daha sonra karıncalar turlarında geçtikleri yollara feromen eklemesi ile olur (Dorigo ve Stützle, 2004).

KKA değişik versiyonlarında farklı feromen yenileme kuralları kullanılmıştır. Karınca kolonisi algoritmasında feromen yenilemesi lokal ve global olmak üzere iki düzeyde gerçekleşmekte ve bir yoldaki toplam feromen düzeyi, lokal ve global feromen düzeyinin toplamından oluşmaktadır (Keskintürk ve Söyler, 2006). Lokal feromen güncellemesi, tüm karıncalar turlarını tamamladıktan sonra, eski feromen miktarları belli bir oranda buharlaştırılır, her bir karıncanın turu boyunca geçiş yapmış olduğu yollarda belli bir miktarda feromen artışı sağlanır (Söyler ve Keskintürk, 2007). Lokal feromen güncellemesi Denklem 4.6 da görüldüğü gibi yapılmaktadır.

$$\tau_{ij} = (1 - \rho) \tau_{ij}(t) + \Delta\tau_{ij} \quad (4.6)$$

Burada;

ρ : Feromen buharlaşma katsayısıdır ($0 < \rho \leq 1$).

$(1 - \rho)$: Feromen maddesinin (t) ile ($t + n$) süreleri arasında buharlaşan oranı.

$\Delta\tau_{ij}$: Birim zamanda (i,j) hattına bırakılmış feromen maddesi miktarıdır ve bu Denklem 4.7 de görüldüğü gibi ifade edilmektedir.

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (4.7)$$

Burada;

m : Toplam karınca sayısı.

$\Delta\tau_{ij}^k$: k. karınca tarafından (i,j) hattına bırakılan feromen madde miktarıdır ve Denklem 4.8 de görüldüğü gibi ifade edilmektedir.

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L^k} & \text{eğer karınca (i,j)yolunu kullanmışsa} \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.8)$$

Burada;

Q : Sabit bir değerdir.

L^k : k karıncasının toplam tur uzunluğudur.

Global feromon güncellemesi, tüm karıncalar turlarını tamamladıktan sonra yapılır. Karıncaların her birinin toplam yol uzunlukları hesaplandıktan sonra en kısa yolu kullanan karınca bulunur (Keskintürk ve Söyler, 2007). Global feromen güncellemesi Denklem 4.9 de görüldüğü gibi yapılır.

$$\tau_{ij} = (1 - \rho) \tau_{ij}(t) + \Delta\tau_{ij}^k \quad (4.9)$$

Burada;

ρ : Feromen buharlaşma katsayısıdır ($0 < \rho \leq 1$).

$(1 - \rho)$: Feromen maddesinin (t) ile (t + n) süreleri arasında buharlaşan oranı.

$\Delta\tau_{ij}^k$: En iyi turu bulan k. karınca tarafından (i,j) hattına bırakılan feromen madde miktarıdır ve Denklem 4.10 de görüldüğü gibi ifade edilmektedir.

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_{(en\ iyi)}} & \text{eğer (i,j)yolu en iyi tura ait ise} \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.10)$$

Burada; $L_{(en\ iyi)}$:Geçerli turda bulunan en iyi turun uzunluğudur (Ying ve Liao, 2004)

4.3. KARINCA KOLONİSİ ALGORİTMASI VERSİYONLARI

KKA ile gezgin satıcı problemlerinin çözümü için yapılan çalışmalarda karınca sistemi algoritmasından sonra farklı algoritmalar geliştirilmiştir. Bunların başında Elit(Elitist) Karınca Sistemi, Derece(Rank) Temelli Karınca Sistemi, Maksimum-Minimum Karınca Sistemi gelmektedir. Önerilen bu çalışmaların feromen güncellemesine dayalı değişiklikler yapılmış ve en iyi sonuca varılmaya çalışılmıştır.

4.3.1. Elit (Elitist) Karınca Sistemi

Karınca sistemi ile ilgili ilk gelişme, karınca sistemi için elitist stratejisi olarak adlandırılan çalışma Dorigo (1992) ile Dorigo ve diğ.(1996) tarafından önerilip geliştirilmiştir. Ana fikir, algoritmanın başlangıcından bu yana bulunan en iyi tura ait yollara ekstra feromen eklenmesinin sağlanmasıdır. Bu algorithmada feromen güncellemesi Denklem 4.11 de görüldüğü gibi yapılmaktadır.

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs} \quad (4.11)$$

Burada;

e : Algoritmanın başlangıcından o ana kadar bulunan en iyi tura verilen önemi tanımlayan parametredir

$\Delta\tau_{ij}^{bs}$: Algoritmanın başlangıcından o ana kadarki en iyi tur üzerindeki feroman madde miktarı ve Denklem 4.12 de görüldüğü gibi bulunur.

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{1}{L^{bs}} & \text{eğer } (i,j)\text{ yolu } \tau^{bs} \text{ turuna ait ise} \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.12)$$

Burada;

L^{bs} : En iyi turun uzunluğudur.

4.3.2. Derece (Rank) Temelli Karınca Sistemi

Derece(rank) temelli karınca sistemi Bullnheimer ve diğ. (1999) tarafından önerilmiştir ve yapılan deneyler sonucunda karınca sistemine göre biraz daha iyi performans gösterdiği görülmüştür. Bu sistemde, feromen izleri güncelleşmeden önce, karıncalar çoğalan uzunluklarına göre sıralanır ve bir karıncanın bırakacağı feromen miktarı r . karıncanın derecesine göre ağırlık kazanır. Her turda sadece en iyi sıradaki karıncalar ve o ana kadar en iyi turu üreten karıncaya feromen bırakması için izin verilir. O ana kadarki en iyi tur bir w ağırlık parametresi ile güçlü bir geri besleme sağlar. Yani o anki turun en iyi r . karıncası sağlayacağı feromen katkısı, $\max(0, w-r)$ değerleri arasından bir değer $1/L^r$ ifadesi ile çarpılması sonucu ortaya çıkan sonuca göredir (Dorigo ve Stützle, 2004). Buna göre bu sistemin feromen güncellemesi Denklem 4.13 de görüldüğü gibi yapılır.

$$\tau_{ij} = \tau_{ij} + \sum_{r=1}^{w-1} (w-r) \Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs} \quad (4.13)$$

Burada;

$\Delta\tau_{ij}^r$: r . sıradaki karınca tarafından i ve j hattına bırakılan feromen madde miktarıdır ve $1/L^r$ ile bulunur ($L^r = r$. sıradaki karıncanın tur uzunluğu)

$\Delta\tau_{ij}^{bs}$: Algoritmanın başlangıcından o ana kadarki en iyi tur üzerindeki feroman madde miktarıdır ve $1/L^{bs}$ ile bulunur.

4.3.3. Maksimum- Minimum Karınca Sistemi

Maksimum-minimum karınca sistemi ilk olarak Stützle ve Hoos (1997) tarafından önerilmiştir. Özellikle arama sırasında durgunlaşmayı önlemek için önerilen maksimum-minimum karınca sistemini, karınca sistemine göre 4 temel değişiklik getirmektedir. Birincisi, o anki turda veya o ana kadar olan turlardaki en iyi karıncanın feromen maddesi eklemesine izin verilir fakat bu bütün karıncaların aynı turu izlediği bir durgunluğa sebep olacaktır. Çünkü yeterli bir tur olmamasına rağmen iyi bir yol üzerindeki feromen izi sınırsız bir şekilde artacaktır. Dolayısıyla bu durgunlaşmayı önlemek için ikinci olarak, olası feromen izleri değerleri $[\tau_{min}, \tau_{max}]$ şeklinde değerler arasında sınırlandırılır. Üçüncü olarak, feromen izleri arama başındaki turların

keşiflerini arttırmak için küçük bir buharlaşma katsayısı ile birlikte üst feromen izi sınırına göre başlatılır. Son olarak, maksimum-minimum karınca sisteminde feromen izleri sistemin her durgunlaşmaya yaklaştığında veya belirli ardışık bir tur sayısı için iyiye giden bir tur bulamazsa başlangıca ger döndürülür (Dorigo ve Stützle, 2004). Tüm bunlara göre maksimum-minimum karınca sisteminde $[\tau_{min}, \tau_{max}]$ feromen sınırları Denklem 4.14 görüldüğü gibi bulunur.

$$\tau_{max} = \frac{1}{\rho \cdot L^{en\ iyi}} \quad , \quad \tau_{min} = \frac{\tau_{max}}{a} \quad (4.14)$$

Burada;

τ_{max} : Üst feromen sınırı.

τ_{min} : Alt feromen sınırı.

$L^{en\ iyi}$: Bulunan en iyi turun uzunluğu.

a : Sabit bir değerdir.

4.4. SAYISAL PROBLEMLER İÇİN KARINCA ALGORİTMALARI

Karınca kolonisi önerildiği günden beri mühendislik alanında özellikle sürekli problemleri çözmek için birçok çalışma yapılmıştır. Karınca algoritmaları yukarıda da açıklandığı gibi ilk olarak Dorigo ve diğ. (1991) ile Dorigo (1992) tarafından gezgin satıcı problemi ve karesel atama problemi gibi zor kombinasyonel problemleri için bir multi-ajan yaklaşımı olarak önerilmiştir. Bilim dünyasında şuanda farklı ayrık optimizasyon problemleri için karınca tabanlı algoritmaların geliştirilmesi amacıyla devam eden birçok çalışma vardır. (Corne ve diğ. 1999). Buna ara rotalama problemleri, grafik boyama, network rotalama, düzenli ve ortak sıralama problemleri örnek verilebilir. Bu çalışmalar ile ilgili bugüne kadar olan uygulamalardan bir kısmı Çizelge 4.1 de verilmiştir.

Çizelge 4.1: Ayrık optimizasyon problemleri için ACO ile ilgili yapılan çalışmalar (Dorigo ve diğ., 1999).

Problem	Yazar	Yıl	Algoritma
Gezgin Satıcı Problemi	Dorigo, Maniezzo & Colorni Gambardella & Dorigo Dorigo & Gambardella Stützle & Hoos Bullnheimer, Hartl & Strauss	1991, 1992, 1996 1995 1996, 1997a,b 1997a,b 1997	AS Ant-Q ACS & 3 opt MMAS ASrank
Karesel Atama	Maniezzo, Colorni & Dorigo Gambardella, Taillard & Dorigo Stützle & Hoos Maniezzo & Colorni Maniezzo Colorni, Dorigo & Maniezzo	1994 1997, 1999 1998 1998 1998 1994	AS-QAP HAS-QAP MMAS-QAP AS-QAP ANTS-QAP AS-JSP
Araç Rotalama	Bullnheimer, Hartl & Strauss Gambardella, Taillard & Agazzi	1996, 1997, 1998 1999	AS-VRP HAS-VRP
Bağlantı Temelli Network Rotalama	Schoonderwoerd, Holland, Bruten & Rothkrantz White, Pagurek & Oppacher Di Caro & Dorigo Bonabeau, Henaux, Guerin Snyers, Kuntz & Theraulaz	1996, 1997 1998 1998 1998	ABC ASGA AntNet-FS ABC-smart ants
Bağlantısız Network Rotalama	Di Caro & Dorigo Subramanian, Druschel & Chen Heusse, Guerin, Snyers & Kauntz Van der Put & Rothkrantz	1997, 1998 1997 1998 1998	AntNet & Ant NetFA Regular ants CAF ABC-backward
Grafik Boyama	Costa & Hertz	1997	ANTCOL
Düzenli Sıralama	Gambardella & Dorigo	1997, 2000	HAS-SOP
Ortak Sıralama	Michel & Middendorf	1998, 1999	AS-SCS

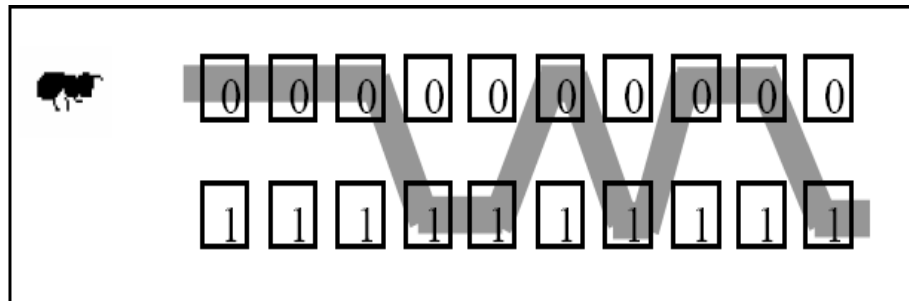
Karınca algoritmaları yukarıdaki gibi ayrık optimizasyon problemleri alanında başarılı bir şekilde kullanılmasına rağmen sürekli optimizasyon problemlerinde bu aynı şekilde değildir. İlk olarak Bilchev ve diğ. (1995) tarafından sürekli problemler için bir karınca algoritması önermişlerdir fakat karınca kolonisi algoritması mekanizması yalnızca lokal arama işleminde kullanılmıştır. Daha sonra Wodrich ve diğ. (1996) tarafından karınca mantığı kullanılarak etkili iki seviye arama işlemi önerilmiştir ve bu algoritma CACO olarak adlandırılmıştır. Algoritma daha sonra Mathur ve diğ. (2000) tarafından genişletilerek performansı önemli ölçüde yükseltilmiştir.

CACO lokal ve global arama seviyeleri olmak üzere iki aşamadan oluşmaktadır. Lokal arama karınca kolonisi algoritmasını kullanırken global arama genetik algoritma

tarafından yapılır. Aslında global karıncalar arama boşluğunda belirlenen bazı bölgelerin uygunluğunu güncellemek için sırayla basit bir değerlendirme yaparlar. Bazı yeni bölgelerin yaratılması, rastgele gezinti davranışı gibi bir davranışa sahip gerçek karınca kolonileri için araştırmacılar tarafından özümsemiş olan ortak işlevleri kullanan genetik algoritmaya benzer bir yöntem ile sağlanır (mutasyon ve çaprazlamanın kısımları ile oynanır). Lokal seviye, optimal değere yakın bölgelere düzenli bir şekilde hareket için, basit bir yoğunlaşma davranışı ile daha sistematik şekilde bölgeleri keşfeden karıncalar tarafından oluşturulur. Algoritma bazı lokal karıncaları bölgelere gönderir, bu karıncalar amaç fonksiyonunda bir düzelme-iyileşme bulduklarında o noktalara feromon maddesi bırakırlar ve bu noktalar kolonideki tüm karıncalar için cazip hale gelir. CACO klasik ACO çok yakın olmasına rağmen iki farklı uygulamanın kullanımı parametre değerlerinin hassas bir ayarlanmasını gerektirmektedir (Mohammad, 2006).

Sürekli optimizasyon problemleri için önerilen diğer algoritmalar ızgaralı karınca koloni optimizasyonu algoritması olan GACO ve tur atan karınca koloni optimizasyon algoritması olan TACO algoritmalarıdır. Bu algoritmalar, Hiroyasu (2000) tarafından önerilmiştir (Karaboğa, 2004).

TACO'da her bir çözüm, binary bitlerin dizisi ile temsil edilmektedir. Bundan dolayı yapay karıncalar bit dizisindeki bitlerin değerlerine karar vermeye çalışırlar. Bu yaklaşımın temel fikri Şekil 4.7'de gösterilmektedir.



Şekil 4.7: Bir karınca tarafından oluşturulan çözüm (Karaboğa ve diğ. 2006)

Bu yapıda karıncalar, bitlerin değerine karar vermek için sadece feromon bilgisini kullanırlar. Dizideki tüm bitlerin değerlerine karar verildikten sonra problem için çözüm

adayı üretilmiş demektir. Üretilen çözüm adayı problemde değerlendirilir ve bu çözüm adayına ait kalite değeri hesaplanır ve kalite değerinden faydalanılarak, çözümü üreten yapay karıncanın geçmiş olduğu yola bırakılacak yapay feromon maddesinin miktarı hesaplanır.

Örneğin, bir bit dizisinin herhangi bir pozisyonunda bulunan 0–1 bitleri arasındaki bağlantının (0→1) tercih edilme olasılığı Denklem 4.15 de görüldüğü gibi hesaplanır.

$$p_{01} = \frac{\tau_{01}}{\tau_{01} + \tau_{00}} \quad (4.15)$$

Burada;

p_{01} : (0 → 1) bağlantısının seçilme olasılığı,

τ_{00} : (0 → 0) bağlantılarının yapay feromon miktarı,

τ_{01} : (0 → 1) bağlantılarının yapay feromon miktarlarıdır ve yapay feromon miktarı

Denklem 4.16 da ifade kullanılarak hesaplanmaktadır.

$$\Delta\tau_{01}^k(t, t + 1) = \begin{cases} \frac{Q}{MF_k} & \text{eğer } k. \text{ karınca } (0 \rightarrow 1) \text{ yolundan} \\ & \text{geçerse} \\ 0 & \text{aksi takdirde} \end{cases} \quad (4.16)$$

Burada;

$\Delta\tau_{01}^k$: (0 → 0) bağlantısına k. karınca tarafından yapıştırılan yapay feromon miktarı,

Q : Pozitif bir sabit değer,

MF_k : Maliyet(Amaç) fonksiyonunun değeridir.

Kolonide bulunan M tane karınca araştırma işlemini tamamladıktan ve çözümleri ürettikten sonra (0→1) bağlantısına t-(t+1) zaman aralığında yapıştırılacak feromon miktarı Denklem 4.17 de görüldüğü gibi bulunur.

$$\Delta\tau_{01}(t, t + 1) = \sum_{k=1}^M \Delta\tau_{01}^k(t, t + 1) \quad (4.17)$$

(t+1) anında (0→1) ara bağlantısında bulunan feromon miktarı ise Denklem 4.18 de görüldüğü gibi bulunur.

$$\tau_{01}(t + 1) = \rho\tau_{01}(t) + \Delta\tau_{01}(t, t + 1) \quad (4.18)$$

Burada;

ρ : Buharlaşma katsayısı ve $(1 - \rho)$ ise feromonun buharlaşma miktarını temsil etmektedir.

Sürekli optimizasyon problemleri için yukarıda tanımlanan algoritmanın temel dezavantajı, yön tayin ederken sadece koku miktarını kullanmasıdır. Bu sebeple bazı optimizasyon problemlerinde erken yakınsama ve bölgesel minimuma takılma problemiyle karşı karşıya kalınmaktadır (Karaboğa ve diğ. 2006). Bu dezavantajı gidermek amacıyla, Kalınlı ve arkadaşları tarafından 2001 yılında alt-yolların koku miktarının rastgele değişimine dayanan bir strateji geliştirilmiştir.

4.5. MOBİL ROBOTLARIN YOL PLANLAMA PROBLEMLERİ İÇİN KARINCA ALGORİTMALARI

Mobil robotlar için yol planlama problemleri bilindiği gibi NP-Hard problemler sınıfına giren problem olarak adlandırılmaktadır. NP-Hard problemlerde çözüme ulaşabilmek için problem içinde olası bütün çözümlerin tek tek denenmesi gerekmektedir. Dolayısıyla çok karmaşık problemlerde bu zamanın hızla artması anlamına gelmektedir. Yol planlama problemleri zor yapan en önemli faktörler çalışma uzayının ve robota ait konfigürasyon uzayının boyutundaki artışıdır. Bu boyutun artması ile yol planlama problemi daha karmaşık hale gelecek ve daha da zor olacaktır. Bu zor problemin çözümünde klasik ve olasılığa dayalı yol planlama tekniklerine göre sezgisel yol planlama tekniklerinin en büyük avantajı hızlı ve çabuk bir şekilde kabul edilebilir bir çözüm üretmesidir ve bu yüzden sezgisel teknikler yol planlaması problemi gibi NP-Hard problemleri çözmek için uygun görünen tekniklerdir.

Mobil robotlar için yol planlama da kullanılan sezgisel tekniklerden birisi de KKA' dır ve yol planlaması için karınca kolonisi ile ilgili literatürde birçok çalışma vardır. Geçmişe göre özellikle son yıllarda karınca kolonisi ile yol planlaması çalışması sıkça yapılmaktadır. Karınca kolonisi ile yapılan yol planlaması araştırma ve çalışmalarda KKA çoğu zaman klasik ve olasılığa dayalı yol planlama tekniklerinin oluşturmuş oldukları bir uzayda arama yapmak için kullanılmıştır. Bununla birlikte, bazı çalışmalarda yalnız veya farklı sezgisel tekniklerle arama uzayları oluşturarak istenen çözümü bulmaya çalışmışlardır. Bu çalışmaların bazılarında farklı engel sakınma formülleri karınca kolonisi algoritmalarıyla modellenmiştir.

KKA ile yapılan çalışmalar genel olarak iyi sonuçlar elde edilmiş ve bu problemlerin çözümünde etkili teknikler olduğu belirtilmiştir. Özellikle problemlere hızlı bir şekilde optimal çözümler bulması bu tekniğin başlıca kullanılma sebebi olarak gösterilmektedir. Bununla birlikte sezgisel tekniklerin ortak özelliklerinden birisi olan algoritma üzerinde değişiklikler yapılabilmesi, anlaşılabilir olması ve değerlendirme için uygun özellikler sunması karınca kolonisinin yol planlama problemleri için uygulanabilirliğini göstermiştir.

5. MATERYAL VE YÖNTEM

5.1. MATERYAL

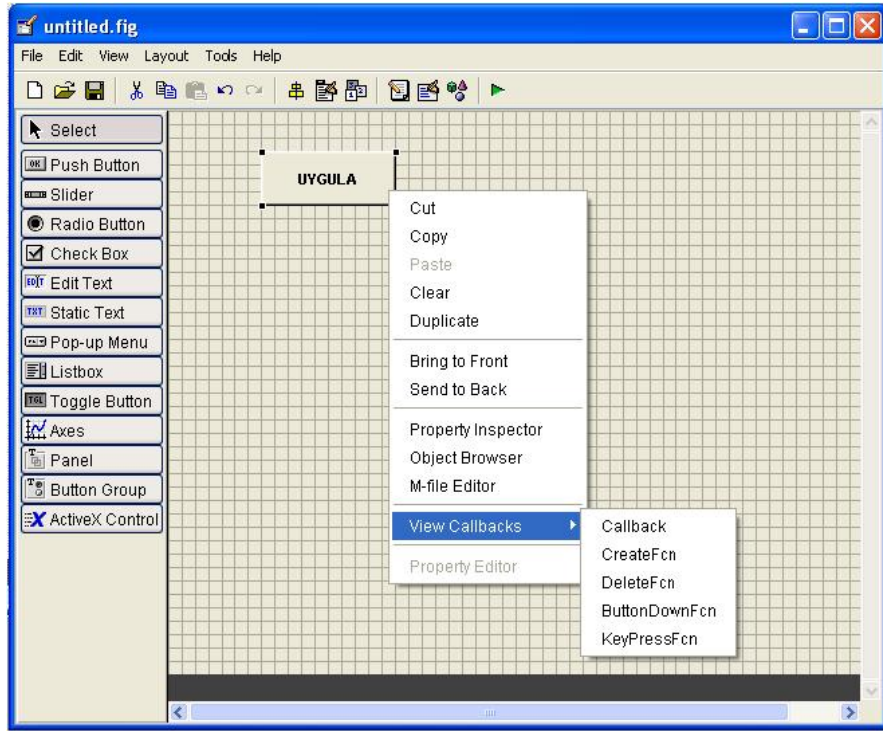
Bu bölümde tez çalışması konusunun uygulanması için kullanılan materyal ve yöntemlere değinilmiştir. Bu bölümde ilk olarak, yol planlaması için geliştirilmiş algoritmanın simülasyonunda kullanmış olduğumuz Matlab ve buna bağlı olarak görselliği sağladığımız GUI ara yüzü hakkında genel bir bilgi verilmiştir. Daha sonra tez için GUI yardımıyla oluşturulmuş ara yüz programımız tanıtılmıştır. Daha sonra, yapılan uygulamaların gerçekleşmesi için yararlanmış olduğumuz yöntemler hakkında temel bilgiler verilir, tez için oluşturmuş olduğumuz algoritma ana hatlarıyla açıklanmıştır.

5.1.1. Matlab ve GUI

Matlab, özellikle son yıllarda başta mühendislik ve bilgisayar alanları olmak üzere geniş bir kullanım alanına sahiptir. Bunun başlıca sebepleri kolay ve anlaşılabilir olması, kullanıcı etkileşimli olması, programcılığı geliştirmek ve farklı algoritmalar oluşturmak için uygun bir program olmasıdır. Matlab, Mathworks şirketinin geliştirmiş olduğu ve ismi Matrix Laboratory kelimelerinin kısaltılmasından meydana gelen bir paket programdır. Matlab programı, basit ve teknik hesaplamaları, kompleks matematiksel problemlerinin çözümünü ve analizini çok hızlı bir şekilde yapan ve bununla birlikte görüntü ve grafiksel olarak verileri oluşturma ve çözüm üretme yeteneğine sahip matris tabanlı bir programdır. Matlab programı, ilk olarak Fortran dilinde yazılsa da, son sürümleri C programlama dilinde oluşturulmaktadır. Matlab'ın matris tabanlı olmasından dolayı, matris formatında olmayan tüm değişkenler, işlemler matris formatına dönüştürülür. Dolayısıyla Matlab kütüphanesinde bu işlemleri yapacak birçok hazır fonksiyon bulunmaktadır. Ayrıca Matlab çok geniş bir grafik kütüphanesine sahiptir. Bu grafik kütüphanesinde her türlü verisel veya işlemsel grafik farklı değişkenlere bağlı olarak 2 veya 3 boyutlu olarak oluşturulabilir ve saklanabilir. Oluşturulan grafikler üzerinde istenilen bütün ayarlamalar yapılabilir. Matlab'ın farklı alanlarda kullanılmasının sebeplerinden birisi de simülük ve GUI eklentileridir. Simülük

ile gerçek hayatta kurmuş olduğumuz birçok deneysel çalışmayı sanal olarak kurma imkanı veren ve içerisinde kontrol sistemleri, generatörler, matematiksel fonksiyonlar, ölçü aletleri v.b. birçok farklı kütüphanesi bulunan bir sanal laboratuardır. Simulink kullanımı çok kolay olmakla birlikte, kurulan sistemlerin denenmesi sırasında kullanıcıya büyük kolaylık sağlayan özelliklere sahiptir. Diğer en önemli eklenti de tez çalışmasında da ara yüz oluşturulmasında kullanılan GUI eklentisidir.

GUI, Graphical User Interface kelimesinin baş harflerinden oluşan ve görsel olarak kütüphanesinde yer alan birçok nesne ile istenilen ara yüzleri oluşturmaya yarayan kullanıcı etkileşimli Matlab programının bir eklentisidir. Kütüphanesini içerisinde birçok farklı buton(push, radio v.b.), text(edit, statik v.b.) kutuları, liste kutuları, grafiksel çizim(axis) v.b. nesnelere bulunmaktadır. Bu nesnelerin kullanımı ile kullanıcı istediği programı bu nesneye uyarlayabilir, yani nesnenin kullanılması(bir butonun tıklanması v.b.) ile yazmış olduğu program çağrılır ve çalıştırılır. Bu program, nesnelerin içeriğinde var olan ve nesne üzerinde sağ tıklama ile görülebilen Callbacks seçeneğinin altında yapılmaktadır (Şekil 5.1).

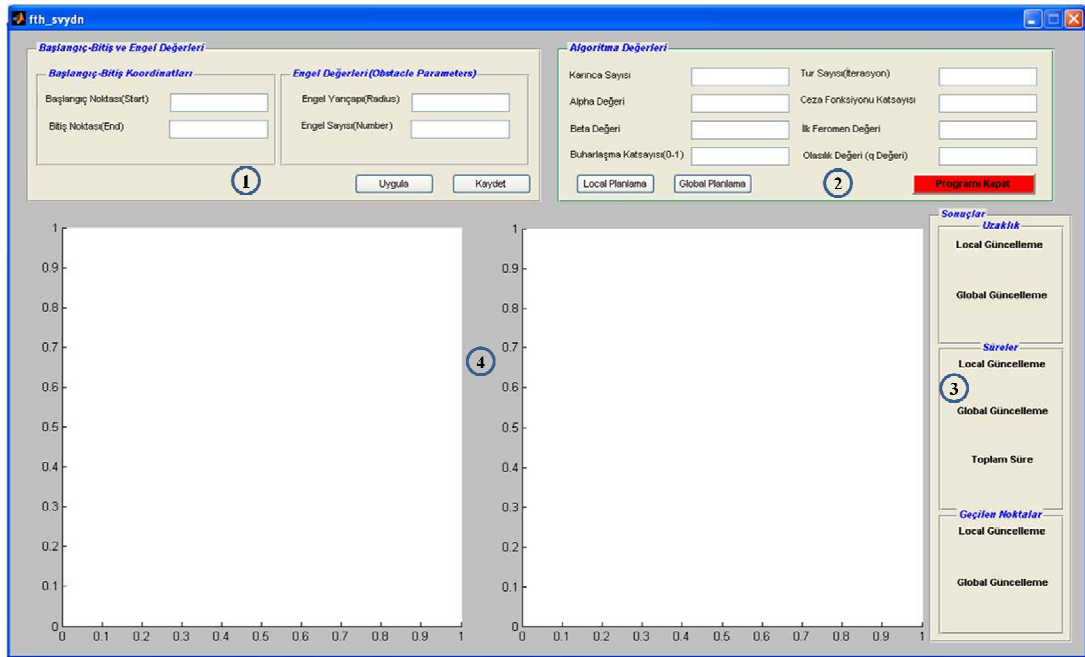


Şekil 5.1: GUI çalışma arayüzü

Arayüz de ne kadar nesne olursa olsun hepsi aynı m-file dosyasında olmak ile birlikte tüm nesnelere m-file dosyasında farklı dizinlerde bulunmaktadır. Yani her bir nesne altındaki yazılmış program birbirinden bağımsız olarak çalıştırılabilir. Arayüz üzerine eklenecek olan nesnelere Şekil 5.1 de görüldüğü gibi sol taraftaki nesne menüsünden mouse yardımıyla sürüklenerek istenilen ara yüz çalışma alanına eklenir ve isteğe göre hizalanabilir. Ara yüz de kullanılan her nesne üzerine çift tıklanarak, nesne üzerine yazı yazma, nesnenin adını değiştirme, nesne rengini, yazı rengini ve boyutunu değiştirme gibi birçok farklı seçenek ile karşılaşılmaktadır.

5.1.2. Tez Çalışması Arayüzünün GUI İle Tasarımı

Tez çalışması ile ilgili arayüz oluşturmadan önce yol planlamasında kullanılacak olan değişkenlerin ve bu değişkenlerin dışarıdan nasıl girileceğine karar verildi. Tez çalışmasının grafiksel arayüzü Şekil 5.2 de görüldüğü tasarlanmıştır.



Şekil 5.2: Tez çalışması için tasarlanan arayüz

Şekil 5.2 de görüldüğü gibi tasarlanan arayüz 4 temel kısımdan oluşmaktadır. 1 numaralı kısımda yol planlamasının yapılacağı başlangıç-bitiş noktalarının koordinatlarının ve çalışma alanına atanacak engellerin yarıçapları ve engel sayıları statik ve edit text nesnelere kullanılarak girilmesi sağlanmıştır. Bununla birlikte push buton nesnesi kullanarak oluşturulan uygula butonu ile başlangıç-bitiş noktaları ve

engeller çizilmektedir ve kaydet butonu da bu engellerin koordinatları daha sonra kullanmak üzere kaydetmektedir. 2 numaralı kısımda da geliştirilen karınca kolonisine ait tüm parametrelerinin yine statik ve edit text nesnelere kullanılarak girildiği kısımdır. Bu kısımda lokal feromen isimli buton lokal feromen güncellemesine göre yol planlaması yaparken, global planlama isimli butona tıkladığında global feromen güncellemesine göre yol planlaması yapılmaktadır. Programı kapat butonuna tıkladığında da oluşturulan arayüz kapatılmaktadır. 3 numaralı kısım da kullanılan önerilen karınca kolonisine ait sonuçlar, algoritmaların çalışma süreleri statik ve edit text nesnelere kullanılarak gösterildiği kısımdır. 4 numaralı kısım ise programın ilk aşamasında başlangıç-bitiş noktalarının, engellerin çizildiği ve program sonunda da yol planlaması için bulunan yolun axis nesnesi kullanılarak çizildiği ve sonuçların görüldüğü kısımdır. Bu nesnelere ek olarak bu 4 kısmı birbirinden ayırmak ve daha kolay görünüm için panel nesnesi kullanılmıştır.

5.2. YÖNTEM

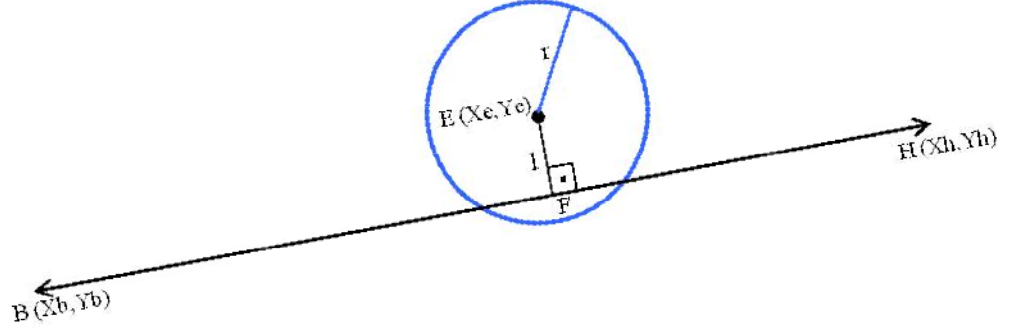
Tez çalışmasında yol planlaması için kullanılan ana yöntem olan karınca kolonisi algoritması yanında alt yöntem olarak bir engelden sakınma yöntemi kullanılmıştır.

5.2.1. Engelden Sakınma Yöntemi

Mobil planlama çalışmalarında yol planlama için optimal yolun bulunmasının yanında başlangıç noktasından hedef noktaya giderken belirlenmiş engellere çarpmamaya şartı vardır. Dolayısıyla bu engellere çarpmamak için önceki bölümlerde de detaylı bir şekilde anlatıldığı gibi birçok yöntem ve teknik vardır. Tez çalışmasında ise matematiksel bir engel sakınma yöntemi kullanılmıştır.

Tez çalışmasında engellerimiz daire şeklindedir ve başlangıç ve bitiş noktaları arasında optimal yolu bulmak için de bir noktanın bir doğruya olan uzaklığı formülünden yararlanılmıştır. Bir noktadan başka bir noktaya her geçişte engellerden sakınma için bu değer hesaplanacak ve dairenin belirlenmiş olan yarıçapı baz alınarak gerekli değerlendirme yapılmıştır, yani iki nokta arasına çizilen bir doğruya dairenin orijininden olan uzaklık hesaplanacak ve bu değer daire yarıçapı ile karşılaştırılıp, buna göre belirlenmiş bir ceza fonksiyonu uygulanıp artık bu yol optimal olmaktan çıkarak,

engelden sakınma sağlanılmıştır. Buna göre detaylı bir şekilde açıklayacak olursak, Şekil 5.3 de görüldüğü gibi bir durumumuz olsun.



Şekil 5.3: Engel orijininin doğruya olan uzaklığı

Şekil 5.3 de görüldüğü gibi daire engel ve bu engele temas etmiş iki nokta arasında doğru gösterilmiştir. Burada $B(X_b, Y_b)$ noktası herhangi bir başlangıç noktası, $H(X_h, Y_h)$ noktası da gidilecek diğer nokta olarak düşünersek, yani karınca kolonisi algoritmasına uyarlırsak $|BH|$ doğrusu karıncanın bir noktadan başka bir noktaya geçişini gösterir. $E(X_e, Y_e)$ noktası engelin orijin noktası olup $|BH|$ doğrusuna olan uzaklığı $|EF|$ uzunluğudur ve l harfi ile gösterilmiştir. Buradan yola çıkarak, engelin orjin noktası olan $E(X_e, Y_e)$ noktasının $|BH|$ doğrusuna uzaklığı, Denklem 5.1 de görüldüğü gibi genel bir formül olan bir noktanın bir doğruya olan uzaklığı formülü ile bulunmaktadır.

$$|EF| = l = \frac{|ax_e + by_e + c|}{\sqrt{a^2 + b^2}} \quad (5.1)$$

Burada $|EF|$ değeri bulunabilmesi için görüldüğü gibi $|BH|$ doğrusunun denkleminin oluşturulması ve buna a, b ve c değerleri bulunarak yerine konulması gerekmektedir. $|BH|$ doğrusunun denklemi $(ax+by+c)$ genel formül olan iki noktası bilinen doğrunun denklemi bulma kuralına dayanarak Denklem 5.2 de görüldüğü gibi bulunmaktadır.

$$\frac{Y_b - Y}{X_b - X} = \frac{Y_h - Y_b}{X_h - X_b} \quad (5.2)$$

Denklem 5.2 içler-dışlar çarpımı yapılmasıyla ve oranlanıp gerekli düzenlemeler yapılmasının sonucunda Denklem 5.3 de görüldüğü gibi $|BH|$ doğrusunun denklemini elde etmiş oluruz.

$$X(Y_h - Y_b) - Y(X_h - X_b) + Y_b(X_h - X_b) - X_b(Y_h - Y_b) \quad (5.3)$$

Burada denklemi $ax+by+c$ düşünürsek X değerinin çarpanı a değerini, Y çarpanının b değerini ve diğerleri c değerini verecektir. Bu elde etmiş olduğumuz değerleri Denklem 5.1 de yerine koyarsak, daire engelin orijininin iki nokta arasında bulunan doğruya olan uzaklığı Denklem 5.4 de görüldüğü gibi bulunmaktadır.

$$|EF| = l = \frac{|(Y_h - Y_b)X_e + (X_h - X_b)Y_e + Y_b(X_h - X_b) - X_b(Y_h - Y_b)|}{\sqrt{(X_h - X_b)^2 + (Y_h - Y_b)^2}} \quad (5.4)$$

Denklem 5.4 de parantezlerin düzenlenmesi ile yapılan düzeltmeler sonucu tez çalışmasında kullanmış olduğumuz engel sakınma denklemi Denklem 5.5 de görüldüğü gibi ortaya çıkmaktadır.

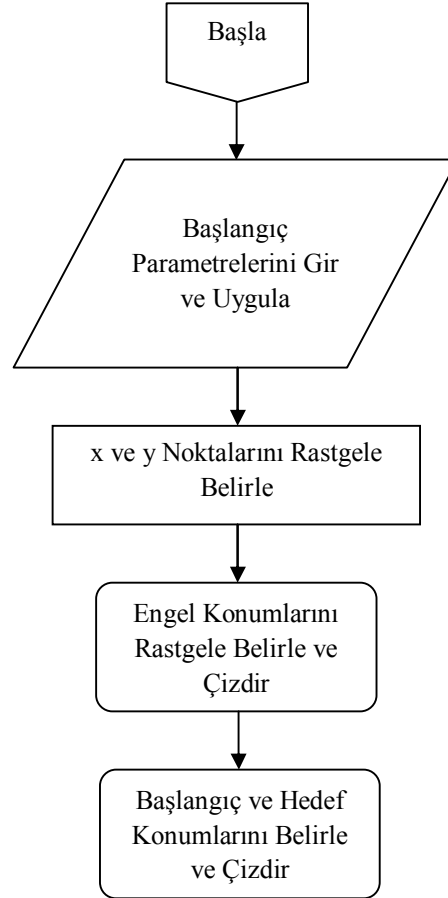
$$|EF| = l = \frac{|(X_e - X_b)(Y_h - Y_b) + (Y_b - Y_e)(X_h - X_b)|}{\sqrt{(X_h - X_b)^2 + (Y_h - Y_b)^2}} \quad (5.5)$$

Tez çalışmasında engelden sakınma yöntemi olarak kullanılan bu matematiksel formül yukarıda da bahsedildiği gibi engelimizin merkez noktasından iki noktaya olan uzaklığı bulmaktadır. Fakat bu formülün sonucuna göre, daire engelin yarıçapına bağlı bir değerlendirme yapılmaktadır. Buna göre algoritma içerisinde ceza fonksiyonu uygulanıp uygulanmayacağına karar verilmektedir. Bu karar verme işlemi, eğer $(r - l) \leq 0$ ise bu durumda ceza fonksiyonu uygulanmaz, eğer $(r - l) > 0$ ise ceza fonksiyonu uygulanarak yapılmaktadır. Karar verme işlemine göre uygulanan ceza fonksiyonları engel hangi yol üzerinde ise o yola eklenerek o yol artık optimal yol olmaktan çıkacaktır ve buna göre bu yol artık tercih sebebi olmayacaktır. Bu engel sakınma yöntemine bir sonraki bölümde önerilen karınca kolonisi algoritmasının açıklanmasında da değinilecektir.

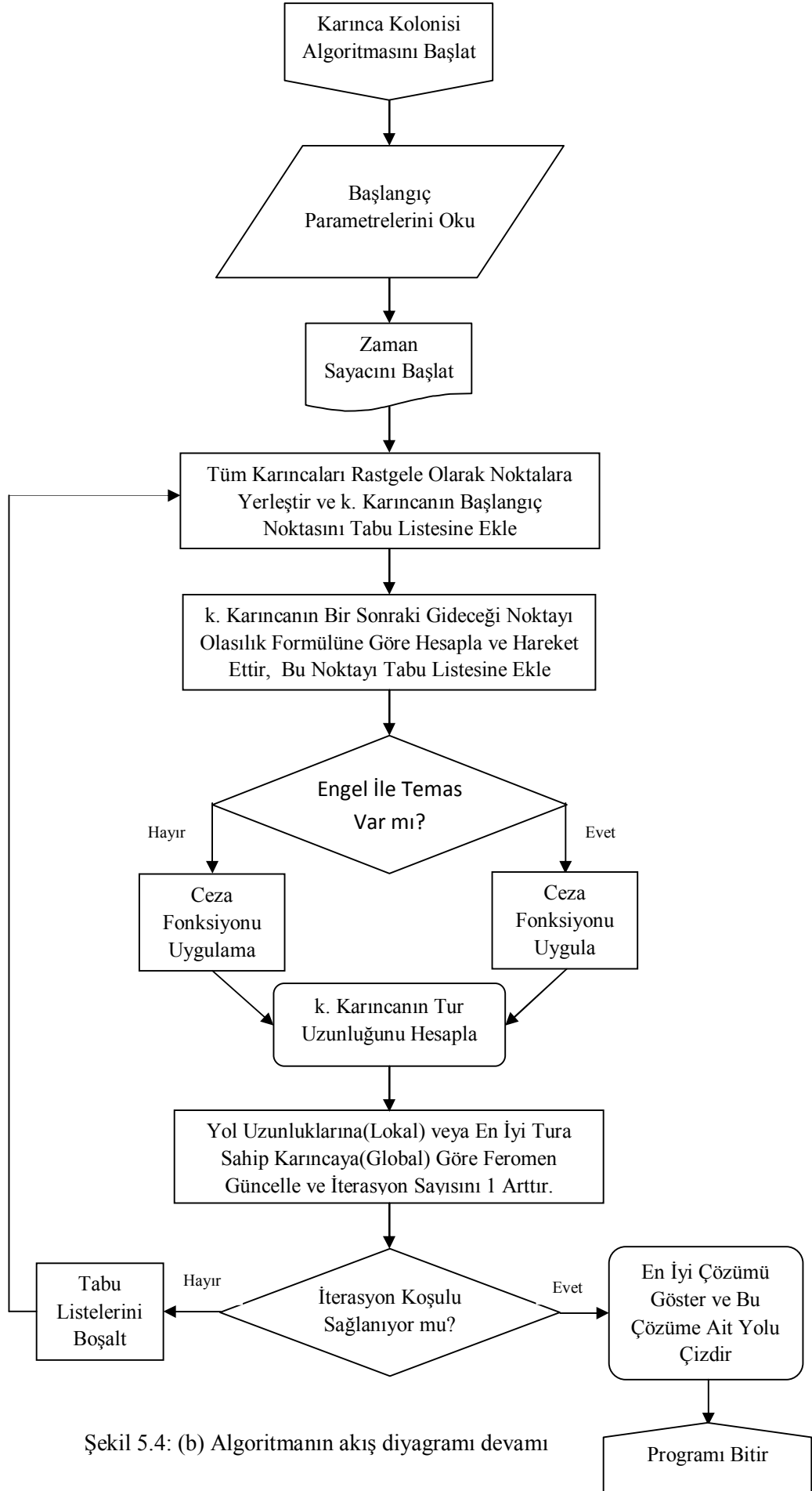
5.2.2. Önerilen Karınca Kolonisi Algoritması Yöntemi

Bu kısımda 4. Bölümde KKO' u temeli detaylı bir şekilde açıklandığından sadece tez çalışmasında mobil robotlar için yol planlama için geliştirilmiş olan karınca kolonisi algoritmasına yer verilmiştir.

Geliştirilen karınca kolonisi algoritması 3 temel kısımdan oluşmaktadır. Birinci kısımda, çalışma alanında başlangıç-hedef noktalarının, hedeflerin ve yolların konumlarının belirlenmesi için (x,y) düzlemine göre noktalar belirlenmekte ve bu noktalara göre başlangıç-hedef noktaları ile engeller çizdirilmektedir. İkinci kısımda, karınca kolonisi algoritmasının lokal güncellemesine göre işlemler yapılmaktadır. Üçüncü kısımda da ikinci kısma her şeyi benzer olmasına rağmen feromon güncellemesine bağlı olarak karınca kolonisi algoritmasının global güncellemesine göre işlemler yapılmaktadır. Buna göre önerilen algoritmanın akış diyagramı Şekil 5.4.a ve Şekil 5.4.b de gösterilmektedir.



Şekil 5.4: (a) Algoritmanın akış diyagramı



Şekil 5.4: (b) Algoritmanın akış diyagramı devamı

5.2.2.1. Algoritmanın İşleyişi

Algoritmamızın işleyişine ilk olarak algoritmanın birinci kısmından başlarsak, tez çalışmasının ana konusu yol planlaması için ilk olarak gerekli olan başlangıç-bitiş noktaları ve engellerin konumları belirlenerek, buna göre çizimlerin yapılmasıdır. Bunun için ilk olarak dışarıdan değerler girilir, uygulanır ve algoritma bu parametreleri program vasıtasıyla kodlayarak çalışmaya başlar.

Algoritma da ilk olarak x ve y noktaları için değerleri, rastgele olarak belirlenmektedir. Burada rastgele değerler, yol planlaması yapılacak grafiksel alanın sınırlarına bağlı kalınarak yapılmıştır ve bu değerler $a * a$ boyutlu bir matris halindedir. Örneğin, 100 tane noktamız var ise bizim x ve y değerlerimizi oluşturan matris $10 * 10$ boyutlu bir matris olacaktır. Oluşturulan bu matris algoritmanın birçok yerinde kullanılıp, gerekli işlemlerin yapılması sağlanmaktadır.

Engellere ait yarıçap ve engel sayısı parametreleri normal değerler olarak girilmektedir. Girilen engel sayısına göre ilk olarak engellerin x ve y konumları algoritma başında belirlenen rastgele x ve y noktalarına bağlı olarak bir matris şeklinde tanımlanır. Bu oluşturulan matrise göre kaç tane engel varsa bunların konumlarına göre ve girilen yarıçapa göre grafiksel olarak çizimi yapılmaktadır.

Başlangıç ve bitiş noktalarının değerleri dışarıdan (sıra, sütun) şeklinde girilmektedir, yani sıra değeri x değerine karşılık gelirken, sütun değeri de y değerine karşılık gelmektedir. Bunun sebebi ise bu noktaların değerleri, yine algoritmanın ilk başında rastgele belirlenmiş olan x ve y değerlerine bağlı olarak seçilmesidir. Burada, dışarıdan girilen sıra ve sütun değerine göre $a * a$ boyutlu matris içerisinde değerler bulunarak, grafiksel olarak çizimi yapılmaktadır.

Algoritmamızın ikinci ve üçüncü kısmında, algoritma başlangıç-bitiş noktaları ve engellerin konumları belirlendikten sonra ve çizildikten sonra çalışmaya başlar. Bu kısımda ilk olarak geliştirilen karınca kolonisi algoritmasına ait parametreler oluşturulan arayüz vasıtasıyla uygun değerler seçilerek girilmekte ve program sayesinde bu parametreler algoritma içerisinde kodlanmaktadır. Dolayısıyla bu belirlenen parametre değerleri bizim yol planlama problemimizin çözümünün etkinliği ve doğruluğu

açısından çok önemlidir. Bu parametre değerlerinin belirlenmesi ile temel bilgiler ilgili karınca kolonisi algoritmasının detaylı bir şekilde anlatıldığı 4. bölümde verilmişti. Buradaki parametre değerlerinin büyük bir kısmı oradaki bilgilere dayanarak belirlenmiş olup birçok değer denenip, en doğru sonuç bulunmaya çalışılmıştır. Algoritmamızda hani feromen güncellemesine göre yol planlaması yapılmak isteniyorsa arayüz üzerinde butona basılarak, buton menüsündeki Callbacks seçeneği altındaki program çalıştırılır.

Algoritmamızın ikinci kısmı çalışmaya başladığında ilk olarak başlangıç-bitiş noktalarının oluşturmuş olduğumuz x ve y noktaların bulunduğu $a * a$ boyutlu matrisin hangi dizininde olduğu bulunur, yani hangi noktalar oldukları bulunur. Bunun sebebi ise başlangıç noktasından bitiş noktasına kadar engelden sakınma formülünü kullanmak ve buna bağlı olarak yolun hesaplanmasında kullanılacak olmasıdır. Bununla birlikte olasılık formülünde kullanılmak için gerekli olan η_{ij} değeri hesaplanmaktadır. Bunun için ilk olarak bütün noktalar arasındaki mesafeler hesaplanmaktadır ve bu değerler artık tüm iterasyonlarda sabit olup algoritmamız boyunca kullanılacaktır.

Bu değerler ve istenilenler belirlendikten sonra artık temel karınca kolonisi algoritması çalışmaya başlar. Algoritma çalışmaya başlamasıyla, başlangıçta yapay karıncaların bir noktadan başka bir noktaya hareketi sırasında p_{ij} formülünü kullandıklarında o formülde bulunan τ_{ij} değerinin 0 olmaması için dışarıdan girilen değere göre bütün noktalara ilk feromen maddesi yerleştirilir ve zaman sayacı ile iterasyon sayısı başlatılarak karıncalar rastgele olarak noktalara yerleştirilir. Daha sonra artık yapay karıncalar ilk yerleşmiş oldukları i noktasından p_{ij} formülüne dayanarak j noktasına hareket ederek turlarını oluşturmaya başlarlar ve bu oluşturulan turların dizini oluşturulur yani tabu listesine eklenir. Dolayısıyla burada her gidilen nokta hafızaya alınıp o noktaya bir daha gidilmesi engellenmiştir. İlk karınca ilk iterasyona göre turlarını oluşturduktan sonra yol uzunluklarının hesabı yapılır.

Geliştirilen algortmada en dikkat edilmesi nokta tur uzunluklarının hesaplandığı kısımdır. Algoritmamızın ikinci kısmına başlarken başlangıç-bitiş noktalarını x ve y noktalarının bulunduğu matrisde hangi dizinde olduğunu algoritma bulmuştu. Bu bulunan dizinler her yapay karıncanın oluşturmuş olduğu tur noktaları dizinine göre bu

başlangıç-bitiş noktalarının hangi dizinde olduğu bulunur, yani başlangıç ve bitiş noktalarının k. karıncanın tur sırasında hangi sırada daha doğrusu dizinde kullandığı bulunur ve artık bu dizinler arasındaki noktaların uzunlukları hesaplanır. Algoritma bu dizinler bulunduktan sonra artık engel sayısı ve engellerin konumları dikkate alınarak yol uzunlukları hesaplanmaya başlar. Başlangıç ve bitiş noktaları arasındaki uzunluk hesaplanırken engel sakınma yönteminde anlatıldığı gibi engellerin durumuna göre ceza fonksiyonu değeri vardır. Algoritma ceza fonksiyonumuzun değerini yine oluşturmuş olduğumuz arayüz vasıtasıyla sağlamaktadır. Burada amaç i ve j noktaları arasındaki normal uzunluğuna ek olarak eğer bu noktalar arasındaki engel ile herhangi bir temas varsa ceza fonksiyonu uygulanıp yolun uzunluğu belli bir oranda artırmaktır. Dolayısıyla normal yol uzunluğuna ceza fonksiyonunun eklenmesiyle bu yol veya tur uzunluğu değeri normal değerinin üstünde olacak ve yapay karıncalar tarafından seçilme olasılığı düşük olacaktır.

Yol uzunlukları engel sakınma yöntemine bağlı olarak hesaplandıktan sonra, lokal güncelleme için her karıncanın turu sonunda hesaplanan uzunlukların değerlerine göre, global güncelleme için de en iyi tura sahip karıncanın kullanmış olduğu yolun uzunluğuna göre feromen güncelleme yapılmaktadır.

Algoritma feromen güncelleme yaptıktan sonra, başlangıç-bitiş noktaları arasındaki optimal yol bulunur ve eğer belirlenmiş iterasyon sayısı koşulu sağlıyorsa bulunan optimal yol yazdırılır ve çizimi grafiksel olarak çizdirilir. Bununla birlikte zaman sayacı durdurulup algoritmanın çalışma süresi ile yol planlaması için kullanılmış noktalar ile bu noktalar arasındaki mesafenin ceza fonksiyonu uygulanmamış hali hesaplanır ve yazdırılır. Başlangıç-bitiş noktaları arasındaki mesafelerin ceza fonksiyonu olmadan direk olarak hesaplanma sebebi ise, optimal değerimiz ile karşılaştırma yapmaktır.

6. BULGULAR

Mobil robotlar için yol planlama problemlerinin çözümü genel olarak statik çevrelerde yapılmaktadır. Diğer bir deyişle yol planlaması yapılacak olan çalışma uzayı içerisinde sadece geometrisi bilinen sabit engelleri bulundurur. Engellerin sabit olması yol planlama probleminin çözümü için daha iyi sonuçlar vermekle birlikte, uygulama açısından da kolaylık sağlamaktadır. Statik çevreler içerisindeki engel, başlangıç-bitiş noktaları gibi faktörlerin geometrik şekli, yarıçapları, uzunlukları ve koordinat sistemi gibi parametreleri probleme başlamadan önce belirlenmesi gerekmektedir ve bu parametreler problem boyunca sabit kalmaktadır.

Yapmış olduğumuz tez çalışmasında yol çalışmasının yapılacağı alan için farklı çevreler oluşturulmuştur. Oluşturulan bu çevreler içersine atanan engellerin yarıçapları ve sayıları farklıdır. Bununla birlikte başlangıç ve bitiş noktaları da farklı değerler verilmiştir. Çevreler oluşturulduktan sonra karınca kolonisi algoritması parametreleri değiştirilerek çalıştırılmış ve elde edilen yol planlama sonuçları üzerinde değerlendirmeler yapılmıştır.

Önerilen karınca kolonisi algoritması parametre değerleri olarak, karınca sayısı değeri her çevre için farklı verilmiştir. α ve β parametre değerleri için 10 farklı değer belirlenmiştir. Buharlaşma katsayısı değerini belirlemek için bütün çevrelerde α ve β için belirlenmiş 10 farklı değerde 0 ile 1 arasında değişen buharlaşma katsayısı değerleri için test ve denemeler yapılmıştır. Bu testlerin sonucunda tüm çevrelerdeki planlama problemleri için buharlaşma katsayısı 0.2 olarak belirlenmiştir. İlk feromen maddesi miktarı değeri ise çok az bir değer olduğundan 0.0001 olarak belirlenmiştir. Bununla birlikte tüm problemlerde ceza fonksiyonu katsayısı değeri sabit bir değer 450 olarak belirlenmiştir.

Problem çözümlerinin sonucunda lokal ve global feromen güncellemelerine bağlı olarak 4 farklı değer elde edilmiştir. Bu değerler amaç fonksiyonu ve uygunluk fonksiyonu

olarak tanımlanmıştır. Amaç fonksiyonumuz başlangıç ile hedef noktası ceza fonksiyonu uygulanmayarak yol uzunluğun bulunduğu fonksiyon değeridir. Uygunluk fonksiyonumuz ise başlangıç ile hedef noktası arasındaki ceza fonksiyonu uygulanarak uzunluğun bulunduğu fonksiyon değeridir. Bunlarla birlikte tüm çalışmaların ortalama zamanı değeri hesaplanarak α ve β değerlerine göre tablo haline getirilmiştir.

Önerilen algoritma Matlab dilinde yazılmış olup, GUI’de bir ara yüz hazırlanmıştır. Algoritma AMD 2.10 GHZ işlemci, 1 GB Ram ve 512 ekran kartı özelliklerine sahip bir bilgisayarda çalıştırılmış ve sonuçlar elde edilmiştir.

Elde etmiş olduğumuz sonuçların arasında gerekli yorumları ve değerlendirmeleri yapabilmek için Denklem 6.1 de görüldüğü gibi bağıl hata formülü kullanılarak aradaki farkların oranı belirlenmiştir.

$$\varepsilon_{bağlı} = \left| \frac{(\text{amaç fonk.} - \text{uygunluk fonk.})}{\text{amaç fonk.}} \right| \quad (6.1)$$

6.1. TEST PROBLEMLERİ ve SONUÇLARI

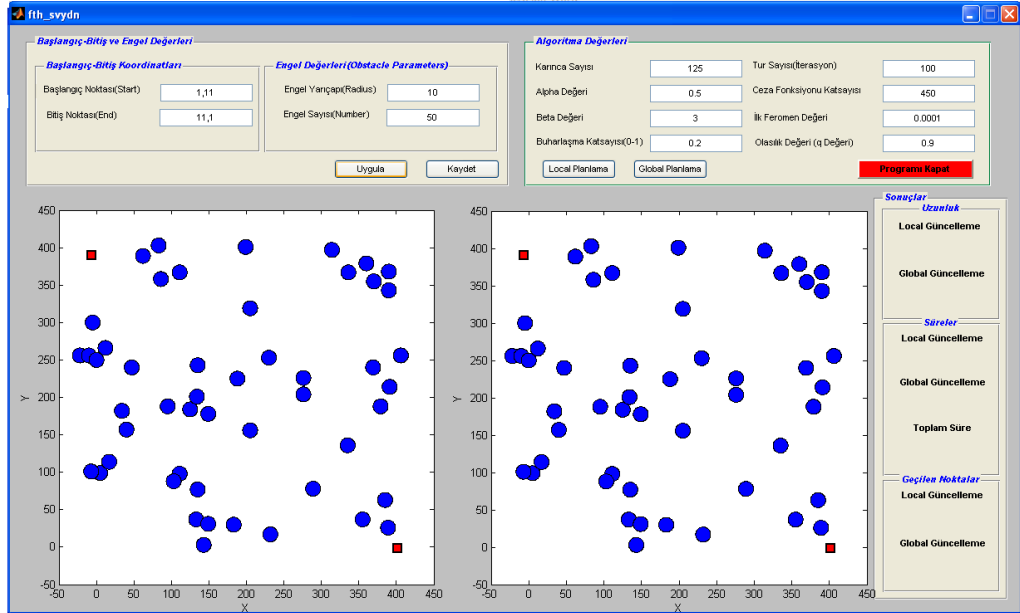
Tez çalışması için tarafımızdan önerilen KKA’nın performansını değerlendirmek için 4 farklı çevre oluşturulmuştur. Bu çevreler için belirlenen alanların değeri x ve y düzlemleri için -50 ile 450 sınır değerleri arasındadır. Çevrelerin engellerinin ve başlangıç-hedef noktalarının koordinatları rastgele olarak belirlenmiştir. Oluşturmuş çevre içersindeki engellerin ve başlangıç-bitiş noktalarının koordinatları bu sınır değerlerini geçmemektedir. Çevrelere ait değişkenler Çizelge 6.1 de görüldüğü gibidir.

Çizelge 6.1: Test için oluşturulan çevrelere ait parametre değerleri

Çevre No	Başlangıç Koordinatları		Hedef Koordinatları		Engel Yarıçapı	Engel Sayısı
	x	y	x	y		
Çevre-1	-6.1577	390.0092	401.7976	-1.9849	10	50
Çevre-2	-4.9365	402.1949	390.5052	1.8417	20	20
Çevre-3	7.1039	355.9986	364.7381	-6.5997	40	12
Çevre-4	-1.7546	365.8177	369.7193	-2.1597	50	8

6.1.1. Çevre – 1

Önerilen KKA algoritmasının performansını değerlendirmek için ilk olarak Şekil 6.1 de görüldüğü gibi yarıçapı küçük ve engel sayısı fazla olan bir çevre seçilmiştir. Çevrelerin engellerinin ve başlangıç-hedef noktalarının koordinatları rastgele olarak belirlenmiştir.



Şekil 6.1: Çevre-1 için elde edilen grafiksel görüntü

Program bu çevre için 50 kere, 100 iterasyon çalıştırılmıştır. Çevre-1'e göre yapılan yol planlamasının çözümünde kullanılan KKA parametre değerleri ve bunlara göre 50 çalıştırmanın ortalama sonuçları Çizelge 6.2 de, bu sonuçların ne kadar sürede elde edildiği de Çizelge 6.3 de gösterilmiştir.

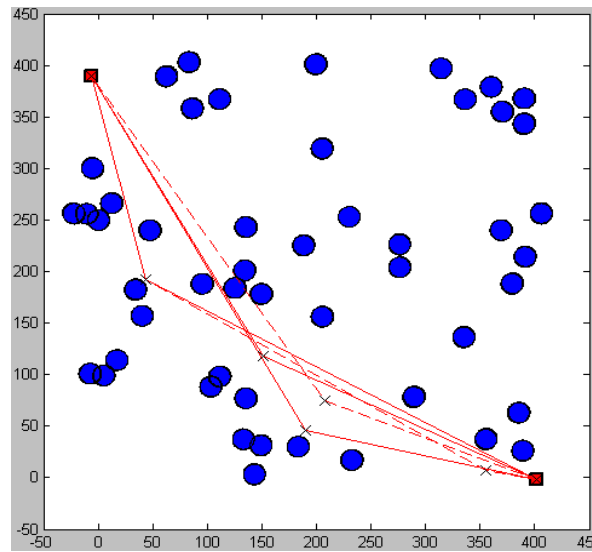
Çizelge 6.2: Çevre-1'e ait KKA parametre değerleri ve sonuçlar

	Parametre Değerleri			Sonuçlar					
	α	β	m	LOKAL PLANLAMA (feromen)			GLOBAL PLANLAMA (feromen)		
				Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata	Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata
Çevre 1	0.5	3	125	621.3452	1727.221	1.779809	616.1928	657.5776	0,067162
	0.5	5	125	615.9206	1967.123	2.193874	613.8949	690.301	0.12446
	0.75	2	125	622.4368	1011.629	0.625272	611.7277	611.7277	0
	0.75	4	125	616.1056	1593	1.585596	610.4085	671.8418	0.09144
	1	1	125	609.301	783.2513	0.285492	610.4054	721.5766	0.182127
	1	2	125	606.1063	987.1859	0.628734	612.0016	618.4833	0.010591
	1	5	125	655.8918	795.9787	0.122104	618.5384	618.5384	0
	2	1	125	594.4023	3555.176	4.981094	610.6849	683.7659	0.119671
	5	1	125	611.2555	2706.423	3.427646	609.4972	776.4892	0.273983
	5	5	125	622.1342	3105.445	3.9916	609.6464	758.2673	0.243782

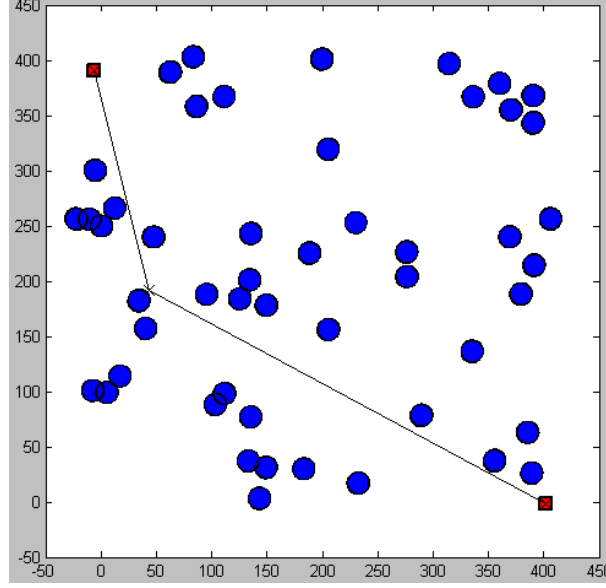
Çizelge 6.3: Çevre-1'e ait çalışma süreleri

Parametre Değerleri			Zaman (sn)	
α	β	Karınca Sayısı	Lokal Planlama	Global Planlama
0.5	3	125	379.9847	332.7092
0.5	5	125	423.1387	339.2438
0.75	2	125	414.321	345.3841
0.75	4	125	445.1971	351.3449
1	1	125	319.6908	269.5456
1	2	125	363.2172	281.238
1	5	125	393.381	286.5237
2	1	125	149.6472	250.5902
5	1	125	156.1897	322.8225
5	5	125	244.3745	309.977

Çevre-1'e ait sonuçlar incelendiğinde, koyu renk ile gösterilen sonuçlar feromen güncellemesine bağlı olarak en iyi sonuçlardır. Burada global yol planlaması lokal yol planlamasının elde ettiği sonuçtan daha optimal uzunlukta bir yol elde ettiği görülmektedir. Lokal ve global olarak bulunan en iyi sonuçların bağıl hata değerlerine bakıldığında da global planlamanın 0 fakat lokal planlamanın 0'ın üzerinde bir değer bulduğu görülmüştür. Yani lokal planlamada 50 çalıştırmadan herhangi birisinde oluşturulan optimal yol engel ile temas etmiş ve buna göre bir ceza fonksiyonu uygulanmıştır. Global planlamada ise herhangi bir ceza fonksiyonu uygulanmadığı ve engel ile temasın olmadığı sayısal olarak görülmektedir. Şekil 6.2 de lokal planlamanın ve Şekil 6.3 de global planlamanın oluşturduğu yollar grafiksel olarak görülmektedir.



Şekil 6.2: Çevre-1 için lokal planlamanın optimal değerime göre çizilen yollar



Şekil 6.3: Çevre-1 için global planlamanın optimal değerime göre çizilen yollar

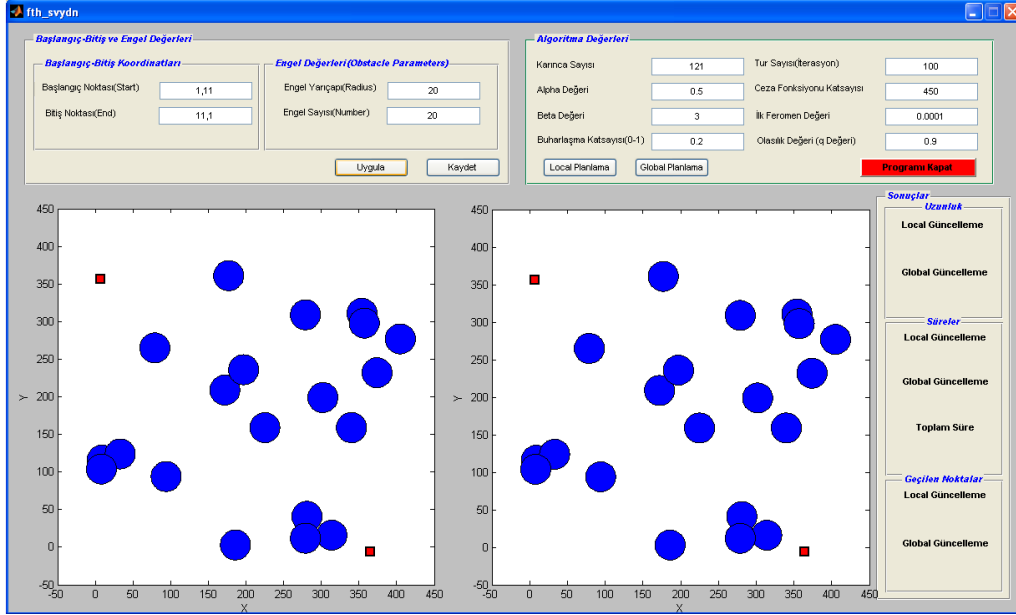
Çevre-1'e ait grafiklere bakıldığında başlangıç ile hedef noktaları arasında lokal planlama birçok farklı yol bulmuştur ve bu yollardan bazılarının engel ile temas ettiği görülmektedir. Bundan dolayı bir ceza fonksiyonu uygulanmıştır ve görüldüğü gibi lokal planlamanın elde ettiği uygunluk fonksiyonu değeri amaç fonksiyonu değerinden büyüktür. Global planlama ise sadece 1 yol elde etmişlerdir. Global planlamaya ait sayısal değere ve grafiğe bakıldığında engellerle herhangi bir temas olmadığı görülmektedir. Lokal feromen güncellemesi ile yapılan yol planlaması ile elde edilen sonuçlar 611.7277 ile 2128.513 arasında iken global planlamaya göre elde edilen sonuçlar sadece 611.7277 değerini almıştır. Yani lokal planlamanın elde ettiği maksimum değeri bir yolun engelle temas olduğunu göstermiştir ve farklı sonuçlar elde etmiştir. Global planlama ise 50 çalıştırmanın hepsinde optimal sonucu elde ederek tutarlılık göstermiştir.

Sürelere göre karşılaştırma yapıldığında lokal planlamanın optimal değeri bulmuş olduğu süre global planlamanın çalışma süresinden daha iyidir fakat probleminin tamamına bakıldığında optimal değeri aradığımızdan global planlamanın bulduğu sonuçlar daha optimaldir. KKA parametre değerleri olarak karşılaştırma yaptığımızda, lokal planlamanın α ve β değerlerinin eşit olduğu durumda optimal sonucu bulmuştur. Global planlamada ise β değerinin α değerinden daha büyük olduğu durumda optimal sonucu bulmuştur.

Yukarıda Çevre-1 için elde edilen sonuçların geneline bakıldığında KKA'nın bütün parametre değerleri için global planlamanın lokal planlamaya göre çok daha iyi sonuçlar verdiği görülmektedir. Bağlı hatalar incelendiğinde global planlama için 2 problemin sonucunda sıfır değeri elde edilmiştir ve bu değerler dışındaki değerlerde 1'den küçük olduğu yani optimala yakın değerler bulunduğu görülmüştür. Lokal planlamanın bağlı hata değerlerine bakıldığında da 5 tane değer 0'ın üstünde olduğu görülmektedir ve engel ile temas sonucunda optimal değerlerden büyük değerler elde ettiği görülmüştür. Problemin genelinde global planlama problemin çözümü için en optimal yolu bulmuştur. Lokal planlamanın optimal değeri bulunduğu durumda da global planlamanın daha iyi bir değer bulunduğu görülmektedir.

6.1.2. Çevre – 2

Çevre-2’de engellerin yarıçapları Şekil 6.4 de görüldüğü gibi bir önceki çevreye göre daha fazla artırılmış fakat sayısı azaltılmıştır. Burada engeller rastgele atanarak farklı bir çevre oluşturulmuştur



Şekil 6.4: Çevre-2 için elde edilen grafiksel görüntü

Program 50 kere 100 iterasyon çalıştırılmıştır. Çevre-2’ye göre yapılan yol planlamasının çözümünde kullanılan KKA parametre değerleri ve bunlara göre sonuçları Çizelge 6.4 de, bu sonuçların ne kadar sürede elde edildiği de Çizelge 6.5 de gösterilmiştir.

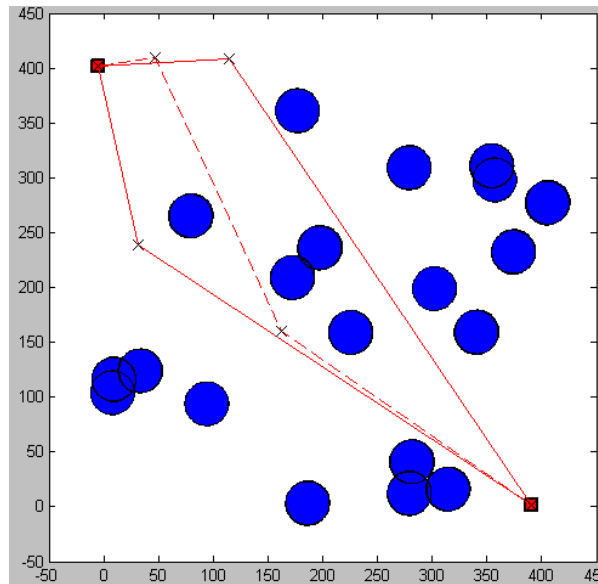
Çizelge 6.4: Çevre-2’ye ait KKA parametre değerleri ve sonuçlar

Çevre 2	Parametre Değerleri			Sonuçlar					
	α	β	m	LOKAL PLANLAMA (feromen)			GLOBAL PLANLAMA (feromen)		
				Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata	Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata
	0.5	3	121	607.7654	607.7654	0	599.2015	599.2015	0
	0.5	5	121	608.2404	608.2404	0	601.7805	601.7805	0
	0.75	2	121	603.6569	603.6569	0	601.2986	601.2986	0
	0.75	4	121	607.7428	607.7428	0	600.0849	600.0849	0
	1	1	121	610.3679	615.0485	0.007668	603.3636	618.537	0.025148
	1	2	121	613.8833	613.8833	0	602.777	630.2217	0.0455304
	1	5	121	604.1793	606.4456	0.003751	603.7238	609.3026	0.0092406
	2	1	121	641.1334	1124.734	0.75429	604.2294	627.6721	0.0387977
	5	1	121	645.753	1704.581	1.63968	623.1219	709.6293	0.1388289
	5	5	121	617.2072	734.8613	0.190623	610.9144	635.7222	0.0406077

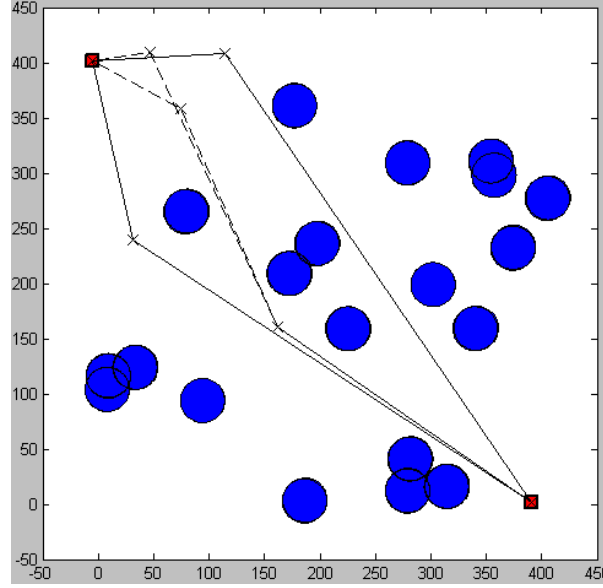
Çizelge 6.5: Çevre-2'e ait çalışma süreleri

Parametre Değerleri			Zaman (sn)	
α	β	Karınca Sayısı	Lokal Planlama	Global Planlama
0.5	3	121	210.2469	196.8532
0.5	5	121	209.713	191.9289
0.75	2	121	212.4653	190.3526
0.75	4	121	222.0755	191.6076
1	1	121	156.5017	140.1408
1	2	121	163.595	130.5414
1	5	121	192.4918	164.2224
2	1	121	109.6371	130.1539
5	1	121	127.1912	174.9206
5	5	121	164.4667	198.0282

Çevre-2'e ait sonuçlar incelendiğinde, optimal sonuçlar yine koyu renk ile gösterilmiştir. Burada global planlamanın lokal planlamaya göre daha iyi bir sonuç bulduğu görülmektedir. Lokal ve global olarak bulunan en iyi sonuçların bağlı hata değerlerine bakıldığında her ikisinin de bağlı hata değerinin 0 olduğu görülmektedir. Yani iki planlamada da engeller ile herhangi bir temas olmadığı sayısal olarak görülmektedir. Şekil 6.5 de lokal planlamanın ve Şekil 6.6 de global planlamanın oluşturduğu yollar grafiksel olarak görülmektedir.



Şekil 6.5: Çevre-2 için lokal planlamanın optimal değerime göre çizilen yollar



Şekil 6.6: Çevre-2 için global planlamanın optimal değerine göre çizilen yollar

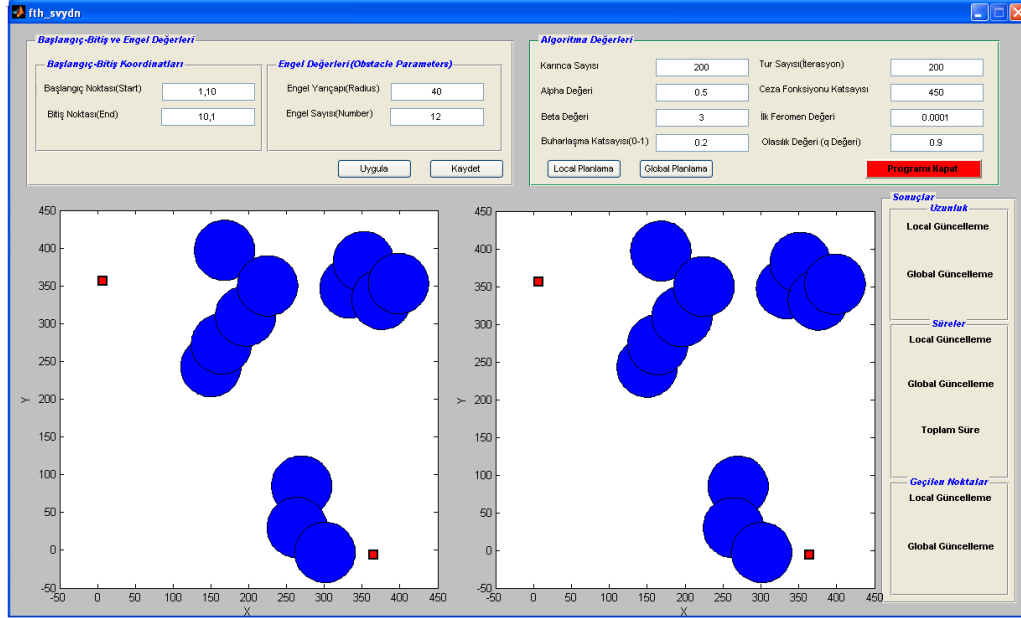
Grafiklere bakıldığında başlangıç ile hedef noktaları arasında lokal planlama 3 farklı yol, global planlama ise 4 farklı yol elde etmişlerdir. İki grafikten görüldüğü gibi yolların engellerin herhangi birisi ile teması bulunmamaktadır. Fakat global planlamanın 50 çalıştırma sonucunda elde etmiş olduğu yol değerlerinin ortalaması lokal planlamanın ortalama değerinden daha iyidir. Her iki planlamaya göre elde edilen optimal sonuçlar 598.6222 ile 611.4088 değerleri arasındadır. Burada lokal planlama 50 çalıştırmada uzun yolu global planlamaya göre daha fazla elde etmiştir ve burada ortalama değer artmasına sebep olmuştur. Yani global feromen güncellemesine göre elde edilen ortalama optimal sonuca bakıldığında 50 çalıştırmada tamamına yakınında minimum değeri ve çok yakın değerleri bularak tutarlılık göstermiştir.

Sürelere göre karşılaştırma yapıldığında, bir önceki çevreye göre daha küçük değerler bulunmuştur. Bunun sebebi ise engel sayısındaki azalmadır. Global planlamanın optimal değeri bulmuş olduğu süre lokal planlamanın çalışma süresinden daha iyidir. KKA parametre değerleri olarak karşılaştırma yaptığımızda, hem lokal planlama hem de global yol planlaması β değerinin α değerinden daha büyük olduğu durumda optimal sonucu bulmuştur. Fakat lokal planlamanın optimalı bulunduğu α ve β değerleri birbirine daha yakın değerlerdir.

Çevre-2 için elde edilen sonuçların geneline bakıldığında KKA'nın bütün parametre değerleri için bağıl hata değerleri, lokal planlamada 5 tane değer 0 iken global planlamada 4 tane bağıl değeri 0'dır. Buna rağmen global planlamanın genelinde elde etmiş olduğu sonuçların birbirine yakın fakat lokal planlamanın bazı parametre değerlerinde yüksek değerler bulunduğu ve bundan dolayı bu değerlerin bağıl hataları 1'in üstüne çıkmıştır. Dolayısıyla bu da o parametre değerlerine bağlı problemde engeller ile temas ettiği anlamına gelmektedir. Problemin genelinde lokal planlama bir optimal yol bulsa da sonuçlardan görüldüğü gibi global planlama problemin çözümü için en optimal yolu bulmuştur. Lokal planlamanın optimal değeri bulunduğu durumda da global planlamanın daha iyi bir değer bulunduğu görülmektedir.

6.1.3. Çevre – 3

Çevre-3’de engellerin yarıçapları Şekil 6.7 de görüldüğü gibi önceki çevrelere göre daha da fazla artırılmış ve engel sayısı azaltılmıştır. Engel yarıçapı büyük olmasına bağlı olarak engelin alan içersindeki kapsamış olduğu alanda artırılmıştır



Şekil 6.7: Çevre-3 için grafiksel görüntü

Program 50 kere, 200 iterasyon çalıştırılmıştır. Çevre-3’e göre yapılan yol planlamasının çözümünde kullanılan KKA parametre değerleri ve bunlara göre sonuçları Çizelge 6.6 de, bu sonuçların ne kadar sürede elde edildiği de Çizelge 6.7 de gösterilmiştir.

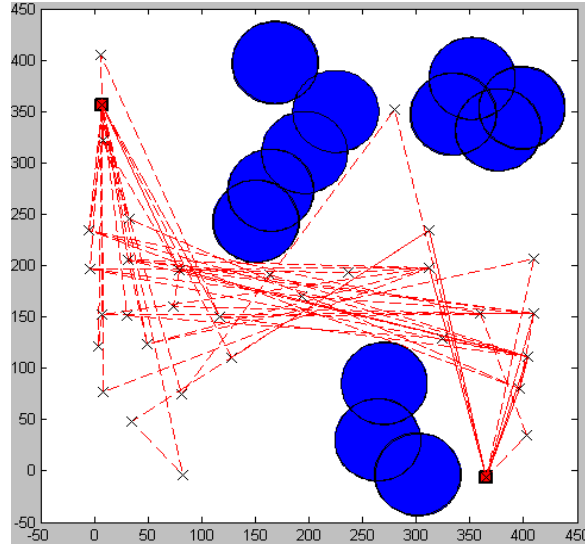
Çizelge 6.6: Çevre-3’e ait KKA parametre değerleri ve sonuçlar

	Parametre Değerleri			Sonuçlar					
	α	β	m	LOKAL PLANLAMA (feromen)			GLOBAL PLANLAMA (feromen)		
				Amaç Fonk.	Uygunluk Fonk.	Bağul Hata	Amaç Fonk.	Uygunluk Fonk.	Bağul Hata
Çevre 3	0.5	3	200	696.1491	1312.452	0.885303	706.4676	722.0239	0.02202
	0.5	5	200	716.2465	3595.227	4.019539	637.4935	638.1415	0.001016
	0.75	2	200	686.9913	1497,31	1.179518	696.0643	697.3255	0.001812
	0.75	4	200	720.5394	734.852	0.019864	705.0367	717.7032	0.017966
	1	1	200	668.1768	674.2354	0.009067	725.5209	742.3742	0.023229
	1	2	200	676.2855	714.0543	0.055847	710.3782	734.3398	0.033731
	1	5	200	715.8131	2530.786	2.53554	704.9494	717.0036	0.017099
	2	1	200	741.0038	3998.481	4.396033	735.1262	848.4086	0.154099
	5	1	200	644.2216	12513.79	18.42467	737.0211	714.3283	0.03079
	5	5	200	718.9835	5840.36	7.123079	751.6521	925.8329	0.231731

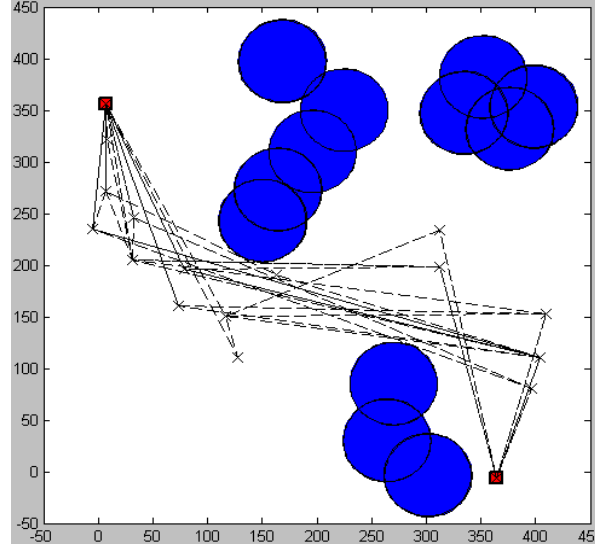
Çizelge 6.7: Çevre-3'e ait çalışma süreleri

Parametre Değerleri			Zaman (sn)	
α	β	Karınca Sayısı	Lokal Planlama	Global Planlama
0.5	3	200	561.031	464.9008
0.5	5	200	568.0734	512.085
0.75	2	200	611.5283	506.9586
0.75	4	200	633.2433	508.2681
1	1	200	399.42	324.6154
1	2	200	393.2235	297.4774
1	5	200	525.2002	411,8486
2	1	200	213.6071	283.7515
5	1	200	367.3538	459.6763
5	5	200	484.7281	554.6939

Çevre-3'e ait sonuçlar incelendiğinde, optimal sonuçlar koyu renk ile gösterilmiştir. Burada global planlamanın lokal planlamaya göre optimal sonuç bulduğu görülmektedir. Lokal ve global olarak bulunan en iyi sonuçların bağıl hata değerlerine bakıldığında her ikisinin de bağıl hata değerinin 0'a yakın olduğu görülmektedir. Yani iki planlamada da 50 çalıştırmanın sadece birkaçında az bir ceza fonksiyonu değeri uygulanmıştır. Şekil 6.8 de lokal planlamanın ve Şekil 6.9 de global planlamanın oluşturduğu yollar grafiksel olarak görülmektedir.



Şekil 6.8: Çevre-3 için lokal planlamanın optimal değerime göre çizilen yollar



Şekil 6.9: Çevre-3 için lokal planlamanın optimal değerime göre çizilen yollar

Grafiklere bakıldığında başlangıç ile hedef noktaları arasında lokal ve global planlama birçok farklı yol elde etmişlerdir. İki grafikten görüldüğü gibi birkaç tane yolun çok az engel ile temas ettiği görülmektedir ve global planlamanın daha düzgün yollar elde ettiği görülmüştür. Sayısal değerlere baktığımızda amaç fonksiyonu ile uygunluk fonksiyonu arasındaki farkın çok az olduğu görülmektedir. Lokal feromen güncellemesine göre elde edilen sonuçlar 662.4632 ile 1001.405 arasında iken global planlama ise 634.5829 ile 773.4362 arasında değerler elde edilmiştir. Buradan lokal planlamanın 50 çalıştırmanın bazılarında engelle temas sağladığı ve birbirinden uzak değerleri elde etmiştir. Global planlamada da minimum değerini daha fazla bulunduğu ve diğer değerlerinde yakın bulunmasıyla tutarlılık içinde olduğu görülmüştür.

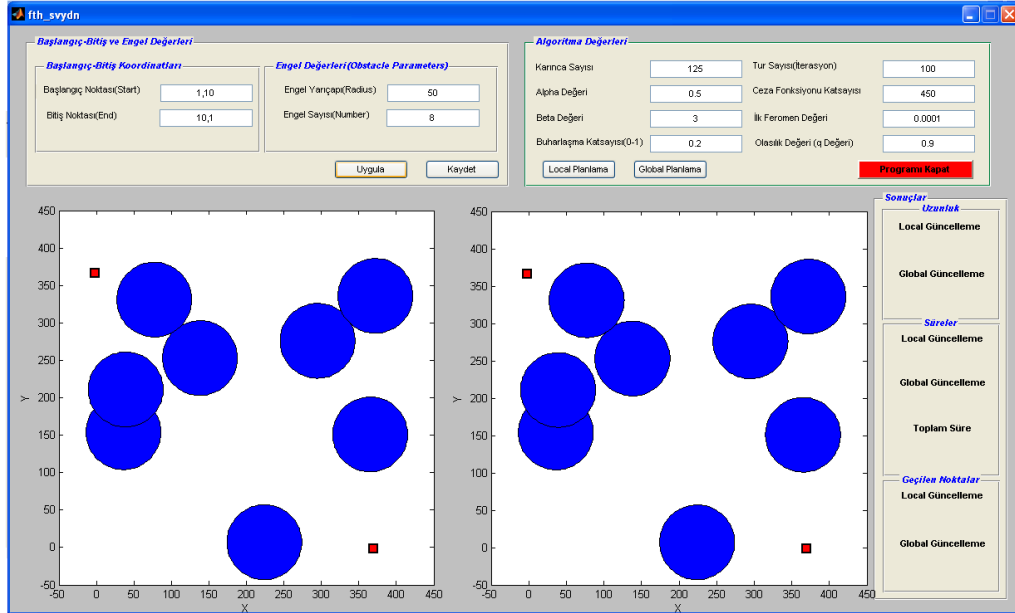
Sürelere göre karşılaştırma yapıldığında, bir önceki çevrelere göre daha fazla değerler bulunmuştur. Bunun sebebi ise karınca ve iterasyon sayısındaki artıştır. Lokal planlamanın optimal değeri bulmuş olduğu süre global planlamanın çalışma süresinden daha iyidir. KKA parametre değerleri olarak karşılaştırma yaptığımızda, lokal planlamanın yine α ve β değerlerinin eşit olduğu durumda optimal sonucu bulmuştur. Global planlama ise yine β değerinin α değerinden daha büyük olduğu durumda optimal sonucu bulmuştur.

Çevre-3 için elde edilen sonuçların geneline bakıldığında KKA'nın bütün parametre değerleri için global planlamanın lokal planlamaya göre çok daha iyi sonuçlar verdiği

görülmektedir. Bağlı hatalar incelendiğinde global planlama için değerlerin 0'a yakın olduğu görülmektedir. Lokal planlamanın bağlı hata değerlerine bakıldığında da 6 tane değer 1'in üstünde olduğu görülmektedir ve engel ile temas sonucunda optimal değerlerden büyük değerler elde ettiği görülmüştür. Problemin genelinde global feromen güncellemesine göre yapılan planlama, problemin çözümü için en optimal yolu bulmuştur.

6.1.4. Çevre – 4

Son çevre olarak oluşturulan Çevre-4'de Şekil 6.10 da görüldüğü gibi yarıçapı en büyük olan bir çevre seçilmiştir ve bu çevrede yine engel sayısı azaltılmıştır. Çevrelerin engellerinin koordinatları yine rastgele olarak belirlenmiştir.



Şekil 6.10: Çevre-4 için grafiksel görüntü

Program bu çevre için 30 kere, 100 iterasyon çalıştırılmıştır. Çevre-4'e göre yapılan yol planlamasının çözümünde kullanılan KKA parametre değerleri ve bunlara göre sonuçları Çizelge 6.8 de, bu sonuçların ne kadar sürede elde edildiği de Çizelge 6.9 de gösterilmiştir.

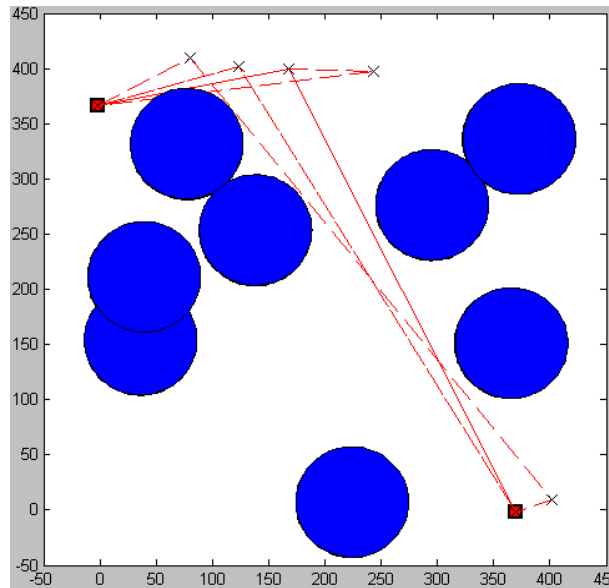
Çizelge 6.8: Çevre-4'e ait KKA parametre değerleri ve sonuçlar

	Parametre Değerleri			Sonuçlar					
	α	β	m	LOKAL PLANLAMA (feromen)			GLOBAL PLANLAMA (feromen)		
				Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata	Amaç Fonk.	Uygunluk Fonk.	Bağlı Hata
Çevre 4	0.5	3	125	617.1804	2302.396	2.730507	608.4889	938.6818	0.542644
	0.5	5	125	611.4278	8465.803	12.84596	603.6912	603.6912	0
	0.75	2	125	611.6132	1257.317	1.055739	603.8958	921.0271	0.525142
	0.75	4	125	639.4613	4277.537	5.689282	602.6173	602.6173	0
	1	1	125	608.0024	768.7844	0.264443	604.7607	667.2108	0.103264
	1	2	125	613.2049	1973.816	2.218852	605.1264	667.5765	0.103202
	1	5	125	617.1804	2302.396	2.730507	608.4889	938.6818	0.542644
	2	1	125	622.5451	4487.806	6.208805	613.7041	1313.617	1.140473
	5	1	125	598.7206	10470.26	16.48772	610.5484	1502.345	1.460648
	5	5	125	617.1835	1758.538	1.849295	606.8871	731.7873	0.205805

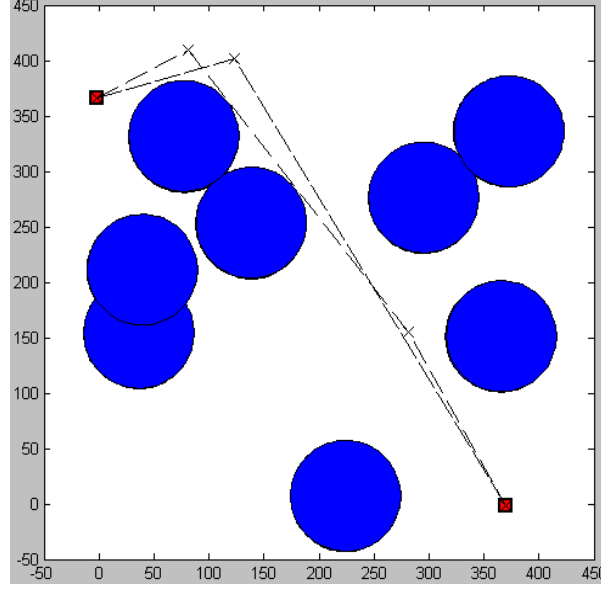
Çizelge 6.9: Çevre-4'e ait çalışma süreleri

Parametre Değerleri			Zaman (sn)	
α	β	Karınca Sayısı	Lokal Planlama	Global Planlama
0.5	3	125	155.6373	145.7037
0.5	5	125	159.148	143.5551
0.75	2	125	164.1026	151.8045
0.75	4	125	156.7070	140.2296
1	1	125	108.5174	99.3095
1	2	125	145.4792	132.6208
1	5	125	155.6373	145.7037
2	1	125	77.5416	87.7951
5	1	125	116.1945	145.0025
5	5	125	150.7892	179.6104

Çevre-4'e ait sonuçlar incelendiğinde, optimal sonuçlar yine koyu renk ile gösterilmiştir. Sonuçlara göre global planlamanın lokal planlamaya göre daha iyi bir değer bulduğu görülmektedir. Lokal ve global olarak bulunan en iyi sonuçların bağlı hata değerlerine bakıldığında global planlamanın bağlı değerinin 0, lokal planlamanın bağlı değerinin ise 0'dan büyük olduğu görülmektedir. Yani lokal planlamada 30 defa çalıştırmanın herhangi birisinde engel ile teması olduğu ve ceza fonksiyonuna göre değer arttığı sayısal olarak görülmektedir. Şekil 6.11 de lokal planlamanın ve Şekil 6.12 de global planlamanın oluşturduğu yollar grafiksel olarak görülmektedir.



Şekil 6.11: Çevre-4 için lokal planlamanın optimal değerime göre çizilen yollar



Şekil 6.12: Çevre-4 için global planlamanın optimal değerime göre çizilen yollar

Grafiklere bakıldığında başlangıç ile hedef noktaları arasında lokal planlama 4 farklı yol, global planlama ise 2 farklı yol elde etmişlerdir. Lokal planlama ait Şekil 6.7 de görüldüğü gibi elde edilen yollardan 2 tanesinin engelle temas ettiği görülmektedir. Lokal planlamada engele temas eden bu yollar yol uzunlukları ortalamasının artmasına sebep olmuştur. Lokal feromen güncellemesine göre elde edilen sonuçlar 603.6912 ile 1246.903 değerleri arasında iken global planlamada ise 601.1428 ile 603.0083 değerleri arasındadır. Yani lokal planlama için elde edilen optimal sonuç içinde engelle temas olduğu bazı durumları da bulunmaktadır. Global planlama ise engel ile temasın olmadığı optimal değeri ve bu optimale değere yakın değerleri elde ederek 30 çalıştırmada tutarlılık göstermiştir.

Çevre-4'ün çalışma sürelerine göre karşılaştırma yapıldığında, diğer çevrelere göre daha az değerler bulunmuştur. Bunun sebebi yine engel sayısındaki azalmadır. Lokal planlamanın optimal değeri bulmuş olduğu süre lokal planlamanın çalışma süresinden daha iyidir. KKA parametre değerleri olarak karşılaştırma yaptığımızda, lokal planlamada α ve β değerinin aynı olduğu, global planlamada β değerinin α değerinden daha büyük olduğu durumda optimal sonuç elde edilmiştir.

Çevre-4 için elde edilen sonuçların geneline bakıldığında KKA'nın bütün parametre değerleri için bağıl hata değerleri lokal planlamanın sadece optimal değeri bulunduğu durumun dışında diğer 9 tane durumda 1'in üstündedir. Yani lokal planlamanın elde etmiş olduğu sonuçlar optimal değerlerin bir hayli üstündedir. Lokal planlamanın sonuçlarına bakıldığında da sayısal olarak ceza fonksiyonunun uygulandığı yani engeller ile temas olan yolların çok kullanıldığı net bir şekilde görülmektedir. Global planlamada ise 2 tane bağıl değer 0'dır. Global planlamanın elde etmiş olduğu sonuçlara bakıldığında da genel olarak iyi sonuçlar bulunduğu görülmüştür. Fakat birkaç tanesinde optimal değerden uzak değerler bulunduğu görülmüştür. Problemin genelinde problemin çözümü için en optimal yol global planlama tarafından bulunmuştur. Lokal planlamanın optimal değeri bulunduğu durumda da global planlamanın lokal planlamaya göre daha iyi değerler bulunduğu görülmektedir.

7. TARTIŞMA ve SONUÇ

Bu tez çalışmasında, NP-Hard problem sınıfına giren mobil robotlar için yol planlama probleminin çözümü amaçlanmıştır. Bu yol planlama probleminin çözümü için bir KKA önerilmiştir. Algoritma içerisinde engeller ile temas etmemek için matematiksel bir engelden sakınma yöntemi kullanılmıştır. Bu yöntemle göre, engellerle temas olduğunda bir ceza fonksiyonu uygulanarak yol uzunluklarının değerinin artırılması ve böylece bu yolların tercih etme olasılığının azaltılması amaçlanmıştır.

Tez çalışması için tarafımızdan önerilen KKA 4 farklı çevrede test edilmiştir. Çevreler engellerin yarıçapları ve sayıları ile başlangıç-hedef noktaları değiştirilerek oluşturulmuştur. Oluşturulan her bir çevre için önerilen KKA'nın 10 farklı parametre değeri kullanılarak sonuçlar elde edilmiştir ve her bir çevre için feromon güncellemesine göre lokal ve global planlama olmak üzere 40 farklı değer elde edilmiş ve sonuçlar kaydedilmiştir. Sonuçlara göre bağıl hataları hesaplanmış ve oluşturulan çevrelerde bütün parametre değeri için çalışma süreleri hesaplanmıştır. Elde edilen sonuçlar tablo haline getirilerek, bu tablolar algoritmanın performansını değerlendirmek için kullanılmıştır.

Oluşturulan çevrelerde yapılan yol planlama çözümlerinin sonuçları incelendiğinde, bütün çevrelerde global planlamanın lokal planlamaya göre daha iyi sonuçlar elde ettiği görülmüştür. Lokal ve global planlamanın en iyi sonuçlar çevre-1'de elde etmiş oldukları sonuçlardır. En kötü sonuçları ise çevre-4'de elde etmişlerdir. Engel yarıçap değeri artırılıp engel sayısı azaltıldıkça algoritmanın elde edilen sonuçların kötü olduğu görülmüştür. Yani bir engelin çalışma uzayı içerisinde kaplamış olduğu alan arttığında mobil robotun hareket edeceği alan azaldığından engeller ile temas olasılığına bağlı olarak değerler artmıştır.

KKA'nın parametre değerlerinin, algoritmanın elde etmiş olduğu sonuçlar üzerinde etkisi incelendiğinde lokal planlamanın optimal değeri çevre-2 dışındaki 3 çevrede α ve β değerlerinin eşit olduğu durumda bulunmuştur. Global planlamada ise bütün optimal sonuçlarda, α değeri 0'a yakın iken β değeri ise 2'den yüksek değerler almıştır. Yani lokal feromen güncellemesine göre yapılan planlamada feromen maddesi değeri ile yol uzunluğu değerine göre orantılı olarak bir tercih de bulunulmuştur. Global planlamada ise β değeri yüksek olduğundan feromen maddesine göre daha çok yol uzunluklarına bağlı olarak optimal yollar bulunmaya çalışılmıştır. Elde edilen genel sonuçlara bakıldığında lokal planlama α değerinin büyük olduğu durumlarda, global planlama ise β değerinin büyük olduğu durumlarda optimal değere yakın çözümler elde etmiştir.

Algoritmanın tüm çevreler için çalışma sürelerine bakıldığında en fazla çalışma süreleri çevre-3'dedir. Bunun sebebi ise bu problemde karınca sayısının diğer çevrelere göre fazla seçilmesidir. Çalışma sürelerinin KKA α ve β parametre değerine göre bakıldığında α değerinin yüksek olduğu durumlarda lokal planlama sonuçları daha az süreler elde ederken, β değerinin yüksek olduğu durumlarda da global planlama sonuçları daha az süre elde etmiştir. Bütün çevrelerde lokal ve global planlamaya göre bulunan optimal sonuçların çalışma sürelerine bakıldığında lokal planlamanın daha az sürede sonuca ulaştığı görülmüştür.

Sonuç olarak bütün çevrelerde global planlama lokal feromen güncellemesine göre yapılan planlamaya göre problemin çözümü için optimal sonuçlar elde etmiştir. Lokal feromen güncellemesinde elde edilen sonuçların ortalaması ile minimum ve maksimum değerler arasında çok fark olduğu görülmüştür. Global feromen güncellemesine göre elde edilen sonuçlarda ise elde edilen ortalama sonuçlar ile minimum ve maksimum değerlerin birbirine çok değerler olduğu görülmüştür. Buna bağlı olarak çalıştırma sayılarına göre lokal planlama tutarsız sonuçlar elde ederken, global planlama neredeyse bütün çalıştırmalarda tutarlı sonuçlar elde etmiştir. Çevrelerin tamamında global planlama optimal yol uzunluklarına ulaşmıştır ve her çevre için gerçekleşen çalıştırmalarda %80 ile %100 oranı arasında aynı değeri bulduğu görülmüştür.

Yapılan bu tez ile literatürde sıkça karşılaşılmayan mobil robotların yol planlaması için kullanılan KKA'nın lokal ve global feromen güncellemesi arasında karşılaştırma yapılarak, yol planlaması yapılmıştır. Tez ile ilgili ileriki zamanlarda yapılacak olan çalışmalarda feromen güncellemelerini sınırlandırmak amacıyla Maksimum-Minimum karınca sistemi algoritması kullanılabilir. Bununla birlikte engel sakınma yöntemine bağlı olarak uygulanacak ceza fonksiyonu uygulama kriterleri üzerinde çalışılabilir ve ayrıca farklı geometrik şekillere sahip engeller ile yol planlama için çevreler oluşturularak çalışmalar yapılabilir.

KAYNAKLAR

- AKSOY, R., KURNAZ, S., 2009, Sektörel Grid Temelli Mihverleme Yöntemi ve Genetik Algoritmalarla İnsansız Kara Aracı Navigasyonu, *Havacılık ve Uzay Teknolojileri Dergisi*, 4(1), 33-45.
- ALAYKIRAN, K., ENGİN, O., 2005, Karınca Kolonileri Metasezgiseli ve Gezgin Satıcı Problemleri Üzerinde bir Uygulaması, *Gazi Üni. Müh. Mim. Fak. Dergisi*, 20(1), 69-76.
- ARICI, V., 2008, *Engellerin Bulunduğu Ortamda Gezgin Robotun En İyi Yolu Bulması ve İzlemesi*, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Başkent Üniversitesi.
- AVNAİM, F., BOÏSSONNAT, J.D., FAVERJON, B., 1988, *A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles*, Technical Report 890, INRIA.
- BİLCHEV, G., PARMEE, I.C., 1995, The Ant Colony Metaphor for Searching Continuous Design Spaces, *Lecture Notes in Computer Science*, 993, 25-39.
- BRAND, M., MASUDA, M., WEHNER, N., YU, X.H., 2010, Ant Colony Optimization Algorithm for Robot Path Planning, *International Conference On Computer Design And Applications*, 25-27 June 2010 Qinhuangdao, New York: IEEE, 436-440.
- BROOKS, R., PEREZ, T.L., 1983, A Subdivision Algorithm in Configuration Space for Findpath with Rotation, *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, August 1983 German, German: DBLP, 799-806.

- BOOR, V., OVERMARS, M.H., STAPPEN A.F., 1999, The Gaussian Sampling Strategy for Probabilistic Roadmap Planners, *Proceedings of the IEEE International Conference on Robotics & Automation*, May 1999 Detroit, New York: IEEE, 1018-1023.
- BULLNHEIMER B., HARTL, F.R., STRAUB, C., 1999, A New Rank-Based Version of the Ant System: A Computational Study, *Central European Journal for Operations Research and Economics*, 7(1), 25-38.
- BUNIYAMIN, N., SARIFF N., WAN, N.W.A.J., MOHAMAD, Z., 2011, Robot Global Path Planning Overview and a Variation of Ant Colony System Algorithm, *International Journal of Mathematics and Computers in Simulation*, 5(1), 9-16.
- CANNY, J.F., 1987, A New Algebraic Method For Robot Motion Planning And Real Geometry, *Proceedings of the IEEE 28th Annual Symposium on Foundations of Computer Science*, 12-14 October Los Angeles, Washington: IEEE, 39-48.
- CANNY, J.F., MING, C.L., 1990, An Opportunistic Global Path Planner , *Proceedings of the IEEE International Conference on Robotics and Automation*, 13-18 May Cincinnati, New York: IEEE, 1554-1561.
- CHEN, D.Z., SZCZERBA, R.J., UHRAN, J.J., 1995, Planning Conditional Shortest Paths Through An Unknown Environment: A Framed-Quadtree Approach, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 5-9 August 1995 Pittsburgh, New York: IEEE, 33-38.
- CHEN, X., YUAN, Y., 2008, Novel Ant Colony Optimization Algorithm For Robot Path Planning, *Systems Engineering and Electronics*, 5.

- CHEN, Y.C., VIDYASAGAR, M., 1988, Optimal Trajectory Planning for Planner N-Link Revolute Manipulators in The Presence of Obstacles, *Proceedings of the IEEE International Conference on Robotics and Automation*, 24-29 April Philadelphia, New York: IEEE, 202-208.
- CHEN, X.S., LIM, M.H., ONG, Y.S., 2007, An Ant Colony System Algorithm for Path Planning in Sparse Graphs, *International Conference on Intelligent and Advanced Systems*, 25-28 November 2007 Kuala Lumpur, New York: IEEE, 31-36.
- CHENG, P., SHEN, Z., LAVALLE, S.M., 2001, Rrt-Based Trajectory Design for Autonomous Automobiles and Spacecraft, *Archives of Control Sciences*, 11(3-4), 167-194.
- CORNE, D., DORİGO, M., GLOVER F., 1999, *New Ideas in Optimization*, McGraw-Hill, London UK, 0-07-709506-5.
- DALKILIÇ, G., TÜRKMEN, F., 2002, Karınca Kolonisi Algoritması, *1. Ulusal Yüksek Performanslı Bilişim Sempozyumu*, Ekim 2002 Kocaeli.
- DENEUBOURG, J., CLİP, P.L., CAMAZİNE S.S., 1994, Ants, Buses and Robots-Self-Organization of Transportation Systems, *From Perception to Action Conference*, 7-9 September 1994 Switzerland, New York: IEEE, 12-23.
- DORİGO, M., STÜTZLE, T., 2004, *Ant Colony Optimization*, MIT Press, London, 0-262-04219-3.
- DORİGO, M., 1992, *Optimization, Learning and Natural Algorithms*, Thesis(PHD), Dipartimento di Elettronica e Informazione, Politecnico di Milano.

- DORÍGO, M., MANÍEZZO V., COLORNÍ A., 1996, The Ant System: Optimization by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics*, 26-(1), 1-13.
- DORÍGO, M., MANÍEZZO V., COLORNÍ A., 1991, The Ant System: An Autocatalytic Optimizing Process, *Technical report 91-016 revised*, Dipartimento di Elettronica, Politecnico di Milano, Milan.
- DORÍGO, M., DÍ CARO, G., GAMBARDELLA, L.M., 1999, Ant algorithms for discrete optimization, *Artificial Life*, 5(2), 137-172.
- ERSSON, T., HU, X., 2001, Path Planning And Navigation of Mobile Robots in Unknown Environments, *RSJ International Conference on Intelligent Robots and Systems*, 29 Oct.-03 Nov. 2001 USA, New York: IEEE, 858-864.
- FAN, X.P., LUO, X., YÍ, S., ZHANG, H., 2004, Path Planning For Robots Based On Ant Colony Optimization Algorithm Under Complex Environment, *Control and Decision*, 19(2), 166-170.
- FAN, X., LUO, X., YÍ, S., YANG, S., and ZHANG H., 2003, Optimal Path Planning for Mobile Robots Based on Intensified Ant Colony Optimization Algorithm, *IEEE Intemational Conference on Robotics Intelligent Systems and Signal Processing*, October 2003 China, New York: IEEE, 131-136.
- FRAZZOLÍ, E., 2001, *Robust Hybrid Control for Autonomous Vehicle Motion Planning*, PhD thesis, Submitted to the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology.
- GAO, M., XU, J., TIAN, J., 2008, Mobile Robot Global Path Planning Based on Improved Augment Ant Colony Algorithm, *Second International Conference on Genetic and Evolutionary Computing*, 25-26 September 2008 Hubei, New York: IEEE, 273-276.

- GARCÍA, M.A.P., MONTIEL O., CASTILLO, O., SEPULVEDA R., and MELÍN P., 2009, Path Planning for Autonomous Mobile Robot Navigation with Ant Colony Optimization and Fuzzy Cost Function Evaluation, *Applied Soft Computing*, 9(3), 1102-1110.
- GIESBRECHT, J., 2004, *Global Path Planning for Unmanned Ground Vehicles*, Technical Memorandum, Defence R&D, Suffield University.
- GOSS, S., ARON, S., DENEUBOURG, J.L., Pasteels, J.M., 1989, Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, 76, 579-581.
- HASSOUNA, M.S., HAKIM, A.E.A., FARAG, A.A., 2005, PDE-Based Robust Robotic Navigation, *Proceedings of the 2nd Canadian conference on Computer and Robot Vision*, 9-11 May 2005 Canadian, New York: IEEE, 176-183.
- HOGAN, N., 1985, Impedance Control: An Approach to Manipulation: Part II-Implementation, *Journal of Dynamic Systems Measurement and Control ASME*, 107, 8-16.
- HSIAO, Y.T., CHNANG, C.L., CHIEN, C.C., 2004, Ant Colony Optimization for Best Path Planning, *IEEE International Symposium on Communications and Information Technology*, 26-29 October 2004 Japan, New York: IEEE, 109-113.
- HSU, D., LATOMBE, J., MOTWANI, R., 1999, Path Planning in Expansive Configuration Spaces, *International Journal of Computational Geometry and Applications*, 9(4/5), 495-512.
- HWANG, Y.K., AHUJA, N., 1992, Gross Motion Planning - A Survey, *ACM Computing Surveys*, 24(3), 219-291.
- JIN, F.H., HONG, B.R., GAO, Q.J., 2002, Path Planning for Free-Flying Space Robot Using Ant Algorithm, *Jiqiren/Robot*, 24(6), 526- 529.

- KALINLI, A., KARABOĞA, N., KARABOĞA, D., 2001, A Modified Touring Ant Colony Optimisation Algorithm for Continuous Problems, *International Symposium on Computer and Information Sciences(ISCIS-XVI)*, Kasım 2001 Türkiye, 437-444.
- KARABOĞA, D., 2004, *Yapay Zeka Optimizasyon Algoritmaları*, Atlas Yayın Dağıtım, İstanbul, 9789756574279.
- KARABOĞA, N., GÜNEY, K., AKDAĞLI, A., KARABOĞA, D., 2006, Doğrusal Anten Dizilerinde Optimum Demet Şekillendirme Amacıyla Karınca Koloni Optimizasyon Algoritmasının Kullanılması, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(1-2), 66-74.
- KAVRAKİ, L.E., *Robotic Path Planning and Protein Modeling* [online], <http://cnx.org/content/m11457/latest/>, [23 Aralık 2010]
- KESKİNTÜRK, T., SÖYLER, H., 2006, Global Karınca Kolonisi Optimizasyonu, *Gazi Üni. Müh. Mim. Fak. Dergisi*, 21-4, 689-698.
- KHATİB, O., 1985, Real Time Obstacle Avoidance for Manipulators and Mobile Robots, *IEEE International Conference on Robotics and Automation*, 500-505.
- KHATİB, O., LİBRE, M., MAMPEY, R., 1978, Fonction Decision-Commande d'un Robot Manipulateur, Raport Scientifique 2/7156, France.
- KODİTSHEK, D.E., 1987, *Exact Robot Navigation by Means of Potential Fields: Some Topological Considerations*, Departmental Papers (ESE), Department of Electrical & Systems Engineering, University of Pennsylvania.
- KUFFNER J.J., LaValle, S.M., 2000, RRT-Connect: An Efficient Approach to Single-Query Path Planning, *Proceedings of the IEEE International Conference on Robotics and Automation*, 24-28 April USA, New York: IEEE.

- LATOMBE, J.C., 1991, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, 0792391292
- LAVALLE, S.M., KUFFNER J.J., 1999, Randomized Kinodynamic Planning, *Proceedings of the IEEE International Conference on Robotics and Automation*, 10-15 May USA, New York: IEEE, 473-479.
- LEE, J.W., CHOY, Y.I., SUGISAKA, M., LEE, J.J., 2010, Study of Novel Heterogeneous Ant Colony Optimization Algorithm for Global Path Planning, *International Symposium on Industrial Electronics* , 4-7 July 2010, IEEE, 1961-1966.
- LEE, J.W., KIM, J.J., 2009, Improved Ant Colony Optimization Algorithm by Path Crossover for Optimal Path Planning, *IEEE International Symposium on Industrial Electronics*, 5-8 July 2009 Seoul Korea, New York: IEEE, 1996-2000.
- LEE, J.W., KIM, J.J., CHOI, B.S., LEE, J.J., 2008, Improved Ant Colony Optimization Algorithm by Potential Field Concept for Optimal Path Planning, *IEEE International Conference on Humanoid Robots*, 1-3 December 2008 Daejeon, New York: IEEE, 662-667.
- LEE, J.W., LEE, J.J., 2010, Novel Ant Colony Optimization algorithm with Path Crossover and Heterogeneous Ants for Path Planning, *IEEE International Conference on Industrial Technology*, 14-17 March 2010 Vi a del Mar, New York: IEEE, 559-564.
- LEVEN, P., HUTCHINSON, S., 2001, Robust, Compact Representations for Real-Time Path Planning in Changing Environments, *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 29 Oct.-03 Nov. 2001 USA, New York: IEEE, 1483-1490.

- LĪU, G., LĪ, T., PENG, Y., HOU, X., 2005, The Ant Algorithm for Solving Robot Path Planning Problem , *Third International Conference on Information Technology and Applications*, 4-7 July 2005 Sydney, New York : IEEE, 25-27.
- LĪU, S., MAO, L., YU, J., 2006, Path Planning Based on Ant Colony Algorithm and Distributed Local Navigation for Multi-Robot Systems, *IEEE International Conference on Mechatronics and Automation*, 25- 28 June 2006 China, New York: IEEE, 1733-1739.
- LUMELSKY, V.J., STEPANOV, A.A., 1986, Dynamic Path Planning for a Mobile Automaton with Limited in Formation on the Environment, *IEEE Transactions on Automatic Control*, AC31(11), 1057-1063.
- LUO, D.L., WU, S.X., 2010, Ant Colony Optimization With Potential Field Heuristic For Robot Path Planning, *Systems Engineering and Electronics*, 6.
- LV, N., FENG, Z., 2006, Numerical Potential Field and Ant Colony Optimization Based Path Planning in Dynamic Environment, *The Sixth World Congress on Intelligent Control and Automation WCICA*, October 2006 Dalian, New York: IEEE, 8966-8970.
- MACIEJEWSKI, A., KLEIN, C.A., 1985, Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments, *International Journal of Robotics Research*, 4(3), 109-116.
- MASEHIAN, E., SEDĪGHĪZADEH, D., 2007, Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review, *World Academy of Science Engineering and Technology*, 23, 101-106.
- MATHUR, M., KARALE, S.B., PRIYE, S., JAYARAMAN V.K., KULKARNĪ, B.D., 2000, Ant Colony Approach to Continuous Function Optimization, *Ind. Eng. Chem. Res.*, 39(10), 3814–3822.

- MEI, H., TIAN, Y., ZU, L., 2006, A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment , *International Journal of Information Technology*, 12(3), 78-88.
- MİYAZAKI, F., ARIMOTO S., 1985, Sensory Feedback for Robot Manipulators, *Journal of Robotic Systems*, 2(1), 53–71.
- MOHAMMAD, H.A., KETABCHI, H., RASA, E., 2006, Elitist Continuous Ant Colony Optimization Algorithm: Application to Reservoir Operation Problems, *International Journal of Civil Engineerng*, 4(4), 274-285.
- MOHAMAD, M.M., DUNNIGAN M.W., TAYLOR, N.K., 2005, Ant Colony Robot Motion Planning, *The International Conference on Computer as a Tool EUROCON*, 21-24 November 2005 Belgrade, New York: IEEE, 213-216.
- MOHAMAD, M.M., DUNNIGAN M.W., TAYLOR, N.K., 2006, Articulated Robot Motion Planning Using Ant Colony Optimisation, *3rd International IEEE Conference on Intelligent Systems*, September 2006 London, New York: IEEE, 690-695.
- NAGIB, G., GHARIEB, W., 2004, Path Planning For a Mobile Robot Using Genetic Algorithms, *Accepted for Publication in International Conference on Electrical*, 5-7 September 2004 Egypt, New York: IEEE, 185-189.
- NILSSON, N.J., 1984, *Shakey The Robot*, Technical Note 323, Computer Science and Technology Division, SRI International.
- NOBORIO, H., NANIWA, T., ARIMOTO, S., 1990, A Quadtree-Based Path-Planning Algorithm for a Mobile Robot, *Journal of Robotic Systems*, 7(4), 555-574.
- OSHER, S., SETHIAN, J.A., 1988, Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, 79, 12-49.

- ÖKDİM, S., 2003, *Disk Planlama İçin Karınca Kolonisi Algoritmasına Dayalı Yeni Bir Metod*, Yüksek Lisans Tezi, Fen Bilimler Enstitüsü, Erciyes Üniversitesi.
- ÖZDEMİR, Y.S., 2008, *Karınca Kolonisi Algoritması ile Bilgisayar Ağlarının Topolojik En İyilenmesi*, Yüksek Lisans Tezi, Fen Bilimler Enstitüsü, Başkent Üniversitesi.
- ÖZFİRAT, M.K., *Robotik Sistemler ve Madencilikte Kullanımının Araştırılması* [online], <http://www.elektrik.gen.tr/icerik/robotik-sistemler-ve-madencilikte-kullaniminin-arastirilmesi>, [30 Aralık 2010]
- ÖZTÜRK, C., 2006, *Karınca ve Sürü Optimizasyon Yöntemlerinin İncelenmesi ve Yazılım Uygulamalarının Oluşturulması*, Yüksek Lisans Tezi, Fen Bilimler Enstitüsü, Marmara Üniversitesi.
- PAVLOV, V., VORONİN, A.N., 1984, The Method of Potential Functions for Coding Constraints of The External Space in An Intelligent Mobile Robot, *Soviet Automatic Control*, 17(6), 45–51.
- RİMON, E., KODİTSHEK, D.E., 1992, Exact Robot Navigation Using Artificial Potential Fields, *IEEE Transactions on Robotics and Automation*, 8(5), 501-518.
- SAMET, H., 1988, An Overview of Quadrees, Octrees and Related Hierarchical Data Structures, *NATO ASI Series*, F40, 51-68.
- SARİFF, N.B., BUNİYAMİN, N., 2010, Ant Colony System for Robot Path Planning in Global Static Environment, *9th WSEAS International Conference on System Science and Simulation in Engineering*, 4-6 October 2010, Japan, 192-197.
- SAUTER, J. A., MATTHEWS, R., PARUNAK, H. V. D., ve BRUECKNER, S., 2002, Evolving Adaptive Pheromone Path Planning Mechanisms, *Proceedings of the*

First International Conference on Autonomous Agents and Multi-Agent Systems, 2002 Bologna Italy, 434-440.

- SAVAŞ, K., 2007, *Kontrol Sistemleri için Matlab'te GUI Uygulamaları Tasarımı*, Lisans Tezi, Teknik Eğitim Fakültesi, Marmara Üniversitesi.
- SCHWARTZ, J., SHARİR, M., 1983, On the Piano Mover's Problem: I. The Case of a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers, *Communications on Pure and Applied Mathematics*, 36, 345-398.
- SHAOGANG, Z., MING, L., 2010, Path Planning of Inspection Robot Based on Ant Colony Optimization Algorithm, *International Conference on Electrical and Control Engineering*, 25-27 June 2010 Wuhan, New York: IEEE, 1474- 1477.
- SHI, C., BU, Y., LI, Z., TAN, J., 2007, Solving Path Planning Problem by an ACO-PSO Hybrid Algorithm, *International Conference on Intelligent Systems and Knowledge Engineering*, 15-16 October 2007 China, Paris: Atlantis-Press.
- SERİN, S., 2009, *Karınca Kolonisi Yaklaşımıyla Karayolu Üstyapı Rutin Bakım Çalışmalarının Planlanması*, Yüksek Lisans Tezi, Fen Bilimler Enstitüsü, Süleyman Demirel Üniversitesi.
- SETHIAN J.A., 1997, Level Set Methods: An Act of Violence, *American Scientist*, 85(3), 1-23.
- SLEUMER, N., GURMAN, N.T., 1999, *Exact Cell Decomposition of Arrangements Used for Path Planning in Robotics*, Technical Report, Institute of Theoretical Computer Science, Topuniversities.
- SOCHA, K., DORİGO, M., 2006, Ant Colony Optimization for Continuous Domains, *European Journal of Operational Research*, 185(2008), 1155–1173.

- SÖYLER, H., KESKİNTÜRK, T., 2007, Karınca Kolonisi Algoritması İle Gezen Satıcı Probleminin Çözümü , 8. *Türkiye Ekonometri ve İstatistik Kongresi*, 24-25 Mayıs 2007 Malatya.
- STİLMAN, B., 1993, Syntactic Hierarchy for Robotic Systems, *Integrated Computer-Aided Engineering*, 1(1), 57-81.
- STİLMAN, B., 1999, *Linguistic Geometry From Search to Construction*, Springer, US, 978-0-7923-7738-2.
- STİLMAN, B., 1995, A Linguistic Geometry for 3D Strategic Planning, *Proc. Of the Goddard Conference on Space Applications of Artificial Intelligence and Emerging Information*, May 1995 Greenbelt, 279-295.
- STİLMAN B., 2009, *Linguistic Geometry Tools LG-PACKAGE*[online], <http://www.stilman-strategies.com> [Ziyaret Tarihi: 07 Nisan 2011].
- STÜTZLE, T., HOOS, H.H., 1997, The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem, *IEEE International Conference on Evolutionary Computation*, 1997 Piscataway, NJ: IEEE Press, 309-314.
- TAN, G.Z., HE, H., SLOMAN, A., 2007, Ant Colony System Algorithm for Real-time Globally Optimal Path Planning of Mobile Robots, *ACTA AUTOMATICA SINICA*, 33(3), 279-285.
- UĞUR, A., AYDIN, D., 2006, Ant System Algoritmasının Java ile Görselleştirilmesi, *Akademik Bilişim*, 2006 Denizli, 572-576.
- VIEN, N.A., VIET, N.H., LEE, S.G., and CHUNG T.C., 2007, Obstacle Avoidance Path Planning for Mobile Robot Based on Ant-Q Reinforcement Learning Algorithm, *Lecture Notes in Computer Science*, 4491(1), 704-713.

- VIET, N.H., VIEN, N.A., LEE, S.G., CHUNG, T.C., 2008, Obstacle Avoidance Path Planning for Mobile Robot Based on Multi Colony Ant Algorithm, *First International Conference on Advances in Computer-Human Interaction*, 10-15 February 2008 France, New York: IEEE, 285-289.
- WANG, X., CYI, P.Y., CHEN, Y.Z., 2005, A New Method and Simulation for Path Planning Problem Based on Ant Colony Algorithm, *Computer Simulation*, 7.
- WANG H.J., XIONG, W., 2009, Research on Global Path Planning Based on Ant Colony Optimization for Auv, *Journal of Marine Science and Application*, 8, 58-64.
- WEN, Y., DENG, M., FAN, H., 2005, Algorithm for Low Altitude Penetration Aircraft Path Planning with Improved Ant Colony Algorithm, *Chinese Journal Of Aeronautics*, 18(4), 304-309.
- WODRÍCH, M., BÍLCHEV, G., 1997, *Cooperative Distributed Search: The Ants' Way*, *Control Cybernetics*, 26(3), 413-446.
- WU, C.Y., LIU, J.S., 2010, Level Set and Fat Fast Marching Method for Normal and Dynamic Path Planning of Pursuit-Evasion Problem, *Proceedings of the IEEE International Conference on Automation and Logistics*, 16-20 August Hong Kong and Macau, New York: IEEE, 267-272.
- WU, Y., SHI, L., LI, Q., CHEN, Y., 2010, Novel Path Planning of Robots Based on Bidirectional Ant Colony Algorithm, *Sixth International Conference on Natural Computation*, 10-12 Aug. 2010 Shandong, New York: IEEE, 2403- 2406.
- YANG, A., GAO, L., LUO, Y., 2010, An Improved Ant Colony System Algorithm for Optimal Path Planning Problem of Mobile Robots, *Second International Conference on Computer Modeling and Simulation*, 22-24 January 2010 Hainan, New York: IEEE, 526-530.

- YE, W., MA, D., FAN, H., 2006, Path Planning for Space Robot Based on The Self-Adaptive Ant Colony Algorithm, *1st International Symposium on Systems and Control in Aerospace and Astronautics*, 19-21 January 2006 Harbin, New York: IEEE, 30-33.
- YU, Y., GAO, H., WANG, D., 2008, Research on Path Planning for Mobile Robot Based on Ant Colony Algorithm in Dynamic Environment, *3rd IEEE Conference on Industrial Electronics and Applications*, 3-5 June 2008 Singapore, New York: IEEE, 497-499.
- ZHANG, J., CHEN, W., ZHONG, J., TAN, X. and LI, Y., 2006, Continuous function Optimization Using Hybrid Ant Colony Approach with Orthogonal Design scheme, *Lecture Notes in Computer Science*, 4247, 126-133.
- ZHANG, X., WU, M., PENG, J., JIANG F., 2009, A Rescue Robot Path Planning Based on Ant Colony Optimization Algorithm, *International Conference on Information Technology and Computer Science*, 25-26 July Kiev Ukraine, New York: IEEE, 180-183.
- ZHAO, J.P., LIU, J.G., GAO, X.W., CHEN, Y.Q., 2010, Research of Path Planning for Mobile Robot Based on Improved Ant Colony Optimization Algorithm, *2nd International Conference on Advanced Computer Control*, 27-29 March 2010 Shenyang, New York: IEEE, 241- 245.
- ZHU, D., LATOMBE, J.C., 1989, *Constraint reformulation and Graph Searching Techniques in Hierarchical Path Planning*, Technical Report, Dept. of Computer Science, Stanford University.
- ZHUANG, H.Z., DU, S.X., WU T.J., 2006, On-Line Real-Time Path Planning of Mobile Robots in Dynamic Uncertain Environment, *Journal of Zhejiang University SCIENCE A*, 7(4), 516-521.

ZOTOS, P., 2009, *Path Planning With The Humanoid Robot iCub*, Semester Project, Biologically Inspired Robotics Group(Bing), École Polytechnique Fédérale de Lausanne.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : SUVAYDAN, Fatih
Uyruğu : T.C.
Doğum tarihi ve yeri : 25.05.1984 - Manisa
Medeni hali : Bekar
Telefon : -
e-mail : fatih_suvaydan@hotmail.com.

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Düzce Üniversitesi / F.B.E. / Elektrik A.B.D.	2011
Lisans	A.İ.B.Ü. / T.E.F / Elektrik Öğretmenliği	2009
Ön Lisans	C.B.Ü. / Akhisar M.YO. / Elektrik Bölümü	2004
Lise	Manisa Endüstri Meslek Lisesi	2001

İş Deneyimi

Yıl	Yer	Görev
2001-2002	Ekinler Elektronik San. Ltd. Şti / Manisa	Kalite Kontrol
2009-2010	M.E.B. / Düzce	Ücretli Öğretmenlik

Yabancı Dil

İngilizce

Yayımlar

Hobiler

Bilgisayar teknolojileri, Basketbol ve Futbol maçları izlemek.