

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

SENSÖR YERLEŐTİRME PROBLEMİNİN
GENETİK ALGORİTMA VE PARÇACIK SÜRÜ
OPTİMİZASYONU İLE ÇÖZÜMÜ

Ali TOZAN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

GEBZE
2007

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

SENSÖR YERLEŐTİRME PROBLEMİNİN
GENETİK ALGORİTMA VE PARÇACIK SÜRÜ
OPTİMİZASYONU İLE ÇÖZÜMÜ

Ali TOZAN
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

TEZ DANIŐMANI
Yrd. Doç. Dr. Fatih Erdoğan SEVİLGİN

GEBZE
2007

 GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ	YÜKSEK LİSANS/DOKTORA JÜRİ ONAY FORMU
--	---

G.Y.T.E. Mühendislik ve Fen Bilimleri/Sosyal Bilimler Enstitüsü Yönetim Kurulu'nun __/__/__ tarih ve __/__/__ sayılı kararıyla oluşturulan jüri tarafından __/__/__ tarihinde tez savunma sınavı yapılan Ali TOZAN'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE (TEZ DANIŞMANI) : Yrd. Doç. Dr. Fatih Erdoğan SEVİLGİN

ÜYE : Prof. Dr. Ercan Öztemel

ÜYE : Doç. Dr. A. Arif ERGİN

ONAY

G.Y.T.E. MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ Yönetim Kurulu'nun __/__/__ tarih ve __/__/__ sayılı kararı.

İMZA/MÜHÜR

ÖZET

TEZİN BAŞLIĞI : Sensör Yerleştirme Probleminin Genetik Algoritma ve Parçacık Sürü Optimizasyonu İle Çözümü

YAZAR ADI : Ali TOZAN

Askeri ve sivil alanda, çeşitli sensörlerin, bir arazi üzerine, o araziye en etkin şekilde gözetlemesi için yerleştirilmesi oldukça önemlidir. Sensör yerleştirme probleminde bir arazi üzerine, belirli sayıda sensörün araziye maksimum kapsamı için konuşlandırılabilmesi en uygun noktaların bulunması hedeflenmektedir. Yapılan çalışmada arazi modellenirken sayısal yükseklik haritalarıyla gerçek yükseklik verileri kullanılmıştır. Sensör modeli olarak çok yönlü bir sensör modeli geliştirilmiş ve görülebilirlik hesaplamalarıyla arazinin sensörler tarafından ne oranda kapsandığı hesaplanmıştır.

En uygun sensör pozisyonlarının bulunması için literatürde optimizasyon problemleri için sıklıkla kullanılan Genetik Algoritma (GA) ve Parçacık Sürü Optimizasyonu (PSO) yöntemleri probleme uyarlanmıştır. Ayrıca bu yöntemleri daha da geliştirmek amacıyla, İndirgenmiş Değişken Komşuluk Arama Metodu, GA ve PSO yöntemlerinin iterasyonlarında uygulanmıştır. Problem algoritmalara uyarlanırken, çözüm kümesinin gösteriminde kullanılan sensör pozisyonları için reel sayılar kullanılmıştır. Haritalarda bulunmayan yükselti bilgileri doğrusal ara değerlendirme ile hesaplanmış ve problemin çözümü için reel sayı yaklaşımı getirilmiştir. Problemin GA ve PSO uyarlamalarında reel sayı yaklaşımının getirilmesi literatürdeki diğer araştırmalara göre farklılık göstermektedir.

Yapılan deneylerde elde edilen sonuçlara göre GA yönteminin PSO yöntemine göre daha iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca yerel arama yönteminin farklı arazi çeşitleri için PSO yöntemini iyileştirdiği fakat GA yöntemini iyileştirmediği görülmektedir.

SUMMARY

TITLE : Genetic Algorithm and Particle Swarm Optimization
Solution for Sensor Placement Problem

AUTHOR : Ali TOZAN

Surveillance of a terrain with various sensors and finding the most suitable positions of sensors is so important in military and public domains. “Given n observers and a terrain, where should we place them in order to maximize the overall coverage of terrain?” is the main question of the sensor placement problem. In this study, the terrain is modelled with digital elevation maps. Sensors are modelled as an omnidirectional observer and the visibility of terrain is computed using line of sight algorithms.

In order to find the most efficient sensor positions, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) methods are applied to the problem. Also Reduced Variable Neighborhood Search is applied as a local search routine in GA and PSO iterations. In problem representation, real valued numbers are used to represent sensor positions. Moreover, linear interpolation method is applied for missing elevation information in digital maps. Main difference of GA and PSO application for sensor placement problem from literature is applying real valued sensor position representation on the problem.

In experiments, GA is outperformed PSO method according to the results. Also local search improves PSO method for different kinds of terrains but does not improve GA.

TEŞEKKÜR

Bu çalışmanın tüm aşamalarında engin bilgi birikimini, deneyimini ve ilgisini hiçbir zaman esirgemeyen ve bana destek olan danışman hocam Yrd. Doç. Dr. Fatih Erdoğan Sevilgen'e, sayısal harita işleme ve sensör modelleme konularında destek olan ve yardımını esirgemeyen değerli arkadaşım Onur İnce'ye, paylaştığı fikirler ve yönlendirmeler ile çalışmamın şekillenmesinde katkısı bulunan TÜBİTAK-MAM Bilişim Teknolojileri Enstitüsü personelinden Prof. Dr. Ercan Öztemel, Ramazan Cengiz ve Dr. Abdülkerim Öncü'ye ve son olarak beni bu günlere getiren aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	İV
SUMMARY	V
TEŞEKKÜR	Vi
İÇİNDEKİLER DİZİNİ	Vii
SİMGELER VE KISALTMALAR DİZİNİ	İx
ŞEKİLLER DİZİNİ	X
ÇİZELGELER DİZİNİ	Xii
1. GİRİŞ	1
1.1. Problemin Tanımı ve Genel Bakış	1
1.2. Tez Çalışmasının Amaçları	1
1.3. Tez Çalışmasının Katkıları	2
1.4. Dokümana Genel Bir Bakış	2
2. LİTERATÜR ARAŞTIRMASI	4
3. META-SEZGİSEL ALGORİTMALAR	11
3.1. Optimizasyon Problemleri	11
3.2. Değişken Komşuluk Arama Metodu	12
3.2.1. Değişken Komşuluk İnişi	13
3.2.2. İndirgenmiş Değişken Komşuluk Arama Metodu	14
3.2.3. Değişken Komşuluk Arama Metodu	14
3.3. Genetik Algoritmalar	15
3.3.1. Basit Genetik Algoritma	16
3.4. Genetik Algoritmalarda Problemin Temsil Mekanizması	17
3.5. Başlangıç Popülasyonunun Oluşturulması	17
3.6. Uygunluk Değerinin Hesaplanması	17
3.7. Çaprazlama İşleminin Uygulanması	18
3.8. Mutasyon İşleminin Uygulanması	19
3.9. Tekrar Üreme	20
3.10. Parametrelerin Seçimi	20
3.11. Parçacık Sürü Optimizasyonu	21
3.12. PSO Çalışma Mekanizması	21

3.13.	PSO Parametre Kontrolü	23
4.	PROBLEMİN FORMÜLE EDİLMESİ VE KULLANILAN MODELLER	25
4.1.	Arazi Modeli	25
4.2.	Çizgisel Görülebilirlik	26
4.3.	Bresenham'ın Çizgisel Görülebilirlik Yöntemi	27
4.4.	Sensör Modeli	28
4.5.	Görülebilirlik Hesabı	29
4.6.	Problemin Matematiksel Modeli	30
5.	PROBLEMİN META-SEZGİSEL YÖNTEMLERE UYARLANMASI	31
5.1.	Genetik Algoritma Uyarlaması	31
5.2.	Kromozom Yapısı	31
5.3.	Çaprazlama Operatörü	32
5.4.	Mutasyon Operatörü	33
5.5.	Seçim Stratejisi	34
5.6.	PSO Uyarlaması	34
5.7.	Parçacık Yapısı	34
5.8.	Parametrelerin Seçimi	35
5.9.	RVNS Uyarlaması	36
5.10.	RVNS Çözüm Kümesi Gösterimi	36
5.11.	Komşuluk Fonksiyonları	36
5.12.	Parametrelerin Seçimi	37
6.	DENEYSEL ÇALIŞMALAR	38
6.1.	Parametrelerin Belirlenmesi	39
6.2.	Karşılaştırmalı Testler	48
7.	SONUÇLAR	56
8.	ÖNERİLER	58
	KAYNAKLAR	59
	ÖZGEÇMİŞ	61

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltma</u>	<u>Açıklama</u>
m	: Metre
km	: Kilometre
km ²	: Kilometrekare
PSO	: Parçacık Sürü Optimizasyonu
YPSO	: Yerel Aramalı Parçacık Sürü Optimizasyonu
GA	: Genetik Algoritma
YGA	: Yerel Aramalı Genetik Algoritma
DTED	: Digital Terrain Elevation Data
LOS	: Line of Sight
TIN	: Triangulated Irregular Network
VNS	: Variable Neighborhood Search
VND	: Variable Neighborhood Descent
RVNS	: Reduced Variable Neighborhood Search

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1 İki boyutlu ızgara noktalarından oluşmuş sensör alanı	5
2.2 Olasılıklı sensör tespit modeli	6
2.3 Sensör alanında bulunan engeller	6
2.4 Sensör görülebilirlik testi	7
2.5 TIN tabanlı arazi modeli	8
2.6 DTED formatındaki haritaya ait arazi modeli gösterimi	10
3.1 VND modeli	13
3.2 RVNS modeli	14
3.3 VNS modeli	15
3.4 Temel Genetik Algoritma adımları	16
3.5 Tek nokta çaprazlama sonucu oluşan kromozomlar	18
3.6 Üç nokta çaprazlama sonucu oluşan kromozomlar	19
3.7 İkili kodlamada mutasyon işlemine uğrayan bir kromozom	20
3.8 Temel PSO adımları	23
4.1 Bresenham çizgi algoritmasının gösterimi	27
4.2 Bresenham çizgi algoritmasının kod dizimi	28
4.3 Görülebilirlik hesabının yapılacağı alanın ızgara haline getirilmesi	29
5.1 Örnek kromozom yapısı.	32
5.2 Çaprazlama operatörü	32
5.3 $\alpha = 0.2$ ve $i = 1$ için örnek çaprazlama	33
5.4 Mutasyonu sonucunda oluşan yeni birey	33
5.5 Örnek bir PSO iterasyonu parçacık hız ve pozisyon hesaplaması	35
6.1 Deneylerde kullanılan alanlardan örnekler	38
6.2 Genetik Algoritma mutasyon oranı test sonuçlarının grafiksel gösterimi	40
6.3 Genetik Algoritma çaprazlama oranı test oranının grafiksel gösterimi	41
6.4 Genetik Algoritma popülasyon büyüklüğü testi gösterimi	43
6.5 PSO c_1 - c_2 parametrelerinin test sonuçlarının grafiksel gösterimi	44
6.6 PSO popülasyon büyüklüğü test sonuçlarının grafiksel gösterimi	45
6.7 YGA için yerel arama iterasyon sayısı test sonuçlarının grafiksel gösterimi	46

6.8 YPSO yerel arama iterasyon test sonuçlarının gösterimi	47
6.9 E037-N33 haritası üzerinde yapılan deneylerin sonuçlarının gösterimi	49
6.10 E037-N39 haritası üzerinde yapılan deneylerin sonuçlarının gösterimi	50
6.11 E037-N37 haritası üzerinde yapılan deneylerin sonuçlarının gösterimi	52

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
4.1 DTED formatındaki haritaların özellikleri	26
6.1 Tüm belirlenmiş alanlar üzerinde denenen örnek problemler	39
6.2 Genetik Algoritma mutasyon oranı test sonuçları	40
6.3 Genetik Algoritma çaprazlama oranı test sonuçları	41
6.4 Genetik Algoritma popülasyon büyüklüğü test sonuçları	42
6.5 Parçacık Sürü Optimizasyonu c_1 ve c_2 parametre test sonuçları	43
6.6 Parçacık Sürü Optimizasyonu popülasyon büyüklüğü test sonuçları	44
6.7 YGA için yerel arama iterasyon sayısı test sonuçları	46
6.8 YPSO yerel arama iterasyon test sonuçları	47
6.9 E037-N33 haritası üzerinde yapılan deneyler ve sonuçları	48
6.10 E037-N39 haritası üzerinde yapılan deneyler ve sonuçları	50
6.11 E037-N37 haritası üzerinde yapılan deneyler ve sonuçları	51
6.12 E037-N33 haritası kapsama haritası ve algoritma denemeleri	53
6.13 E037-N39 haritası kapsama haritası ve algoritma denemeleri	54
6.14 E037-N37 haritası kapsama haritası ve algoritma denemeleri	55

1. GİRİŞ

1.1. Problemin Tanımı ve Genel Bakış

Çeşitli sensörlerin, bir alan üzerine, o alanı en etkin şekilde gözetleyecek veya kapsayacak şekilde yerleştirilmesi oldukça önemlidir. Sensör yerleştirme probleminde “Bir alan üzerine belirli sayıda sensör, nerelere yerleştirilsin ki en fazla kapsama sağlanmış olsun?” sorusuna cevap aranmaktadır. GSM operatörleri için baz istasyonlarının sayısının azaltılması ve kapsama alanlarının artırılması, banka veya önemli yerleşim merkezlerinin güvenliğinin sağlanması için kameraların en uygun yerlere yerleştirilmesi, askeri senaryolarda bir arazinin etkin şekilde gözlenmesi ve takip altına alınabilmesi için termal kamera, radar gibi sensörlerin arazi üzerinde uygun yerlere yerleştirilmesi, temelinde sensör yerleştirme problemini barındıran pratik örneklerden birkaçıdır. Literatürde bu konu hakkında çeşitli çalışmalar yapılmış, çeşitli yöntemlerle çözümler aranmış ve bu çözümlerin sonuçları analiz edilmiştir. Bu tez çalışmasında, problemin çözümü için Genetik Algoritma ve Parçacık Sürü Optimizasyonu yöntemleri uygulanmış ve sonuçlar analiz edilmiştir.

1.2. Tez Çalışmasının Amaçları

Bu tezin genel amacı sensör yerleştirme problemini etkin sonuçlar bulacak bir şekilde çözmek ve bundan sonraki yapılacak çalışmalara bir altyapı hazırlamaktır. Bu tez kapsamında temel olarak ele alınan problem, askeri senaryolarda gerçekleştirilen gözetleme işleminin etkinleştirilmesidir. Bu amaç doğrultusunda yapılması hedeflenenler şunlardır:

- Literatürde bulunan benzer problemleri ortaya koymak ve bu problemlerin çözüm yöntemleri hakkında genel bir fikir edinmek,
- Problemin çözümü için gereken sensör ve arazi modellerini geliştirmek,
- Genetik Algoritmalar, Değişken Komşuluk Arama Metodu, Parçacık Sürü Optimizasyonu gibi tekniklerinin incelenmesi ve probleme uyarlanması,
- Bahsedilen yöntemlerin hangisinin bu problem için daha uygun olduğunun deneylerle araştırılması.

1.3. Tez Çalışmasının Katkıları

Tezin temel katkıları aşağıdaki gibi sıralanabilir:

- Bu konuda çalışmalar yapacak araştırmacılara kaynak sağlamak,
- Genetik Algoritmalar, Değişken Komşuluk Arama Metodu ve Parçacık Sürü Optimizasyonu gibi meta-sezgisel yöntemlere giriş yapılması ve bu yöntemlerin optimizasyon problemlerine uyarlanması,
- Sensör Yerleştirme Probleminin tanımı ve bileşenlerinin modellenmesi,
- Elde edilen çözümler ile pratik uygulama alanları için bir altyapı sunması.

1.4. Dokümana Genel Bir Bakış

Bu dokümanda öncelikle problemin teorik alt yapısı ile ilgili bilgiler verilecek ve ardından yapılan uygulamalar anlatılmaktadır. Son bölümlerde yapılan deneyler ve sonuçları değerlendirilmektedir.

Bölüm 2 kısmında konu ile alakalı yapılan literatür araştırması hakkında bilgi verilecek ve yapılan çalışmanın literatürdeki yeri ifade edilmektedir.

Bölüm 3'te kullanılan çözüm yöntemleri hakkında bilgilendirme yapılmaktadır. Bu yöntemler Genetik Algoritmalar, Parçacık Sürü Optimizasyonu ve İndirgenmiş Değişken Komşuluk Arama Metodu olup, bu yöntemler hakkında gerekli açıklamalar bu kısımda yapılmaktadır.

Bölüm 4'te Sensör Yerleştirme Problemi'nin nasıl modellendiği anlatılmaktadır. Görülebilirlik hesaplarının nasıl yapıldığı, arazinin ne şekilde temsil edildiği, sensör modelinin nasıl oluşturulduğu bu bölümde anlatılmış ve problem bir optimizasyon problemi şeklinde tanımlanmaktadır.

Bölüm 5'te Sensör Yerleştirme Problemi'nin Genetik Algoritma, Parçacık Sürü Optimizasyonu ve İndirgenmiş Değişken Komşuluk Arama Metoduna nasıl uyarlandığı anlatılmaktadır.

Bölüm 6’da önerilen çözümler üzerinde deneyler yapılmış ve elde edilen bulgular değerlendirilmiştir.

Bölüm 7 ve Bölüm 8’de yapılan çalışmanın sonuçları değerlendirilmekte ve ileride yapılabilecek çalışmalar hakkında önerilerde bulunmaktadır.

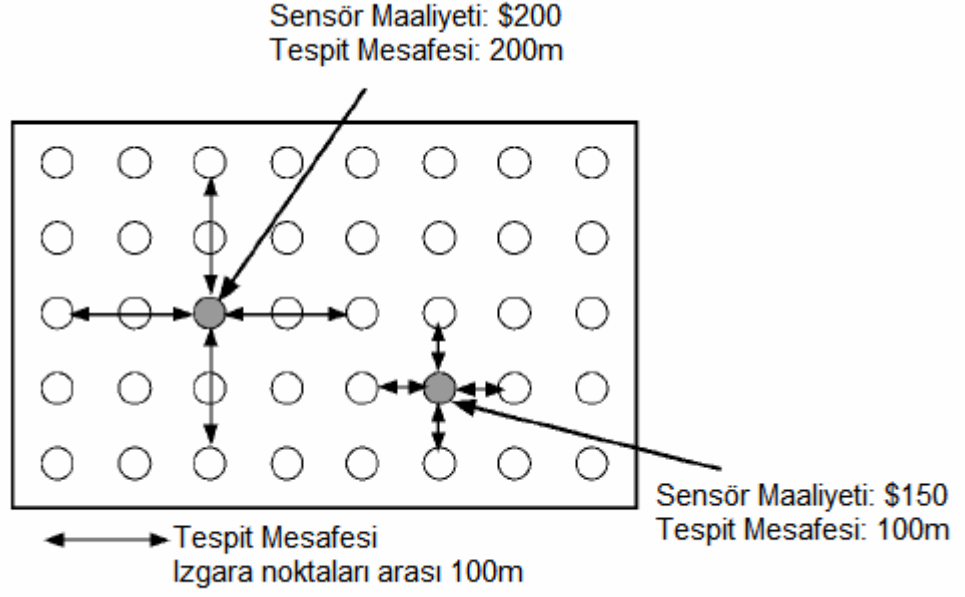
2. LİTERATÜR ARAŞTIRMASI

Sensör yerleştirme problemi çeşitli çalışmalarda ele alınmış ve farklı yaklaşımlarla probleme çözüm aranmıştır. Bu bölümde, literatürde bulunan çalışmalardan örnekler verilecek ve bu tez çalışmasında çözüm aranan problemin mevcut problemler arasındaki yeri tartışılacaktır.

Yapılan araştırmalarda ele alınana problem ve önerilen çözüm 3 temel konuda farklılıklar göstermektedir:

- Amaç Fonksiyonu: Sabit sayıda sensör ile bir alanın maksimum seviyede kapsanması veya en az sayıda sensör kullanılarak bir alanın belirli oranda kapsanması gibi farklı amaçlara göre optimizasyon yapılmaktadır.
- Sensör Modeli: Çok yönlü sensör modeli, sensörlerin menzili, bakış açısı gibi farklılıklar bulunmaktadır.
- Arazi Modeli: 2 boyutlu ve 3 boyutlu arazi modelleri şeklinde iki farklı alan modeli kullanılmıştır.

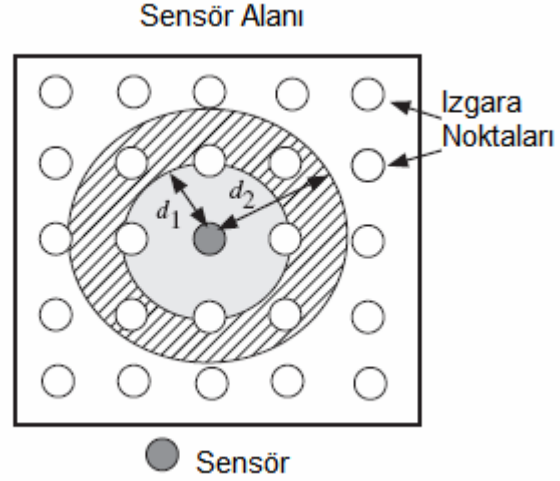
Chakrabarty ve arkadaşları [1] verilen ızgara noktalarından oluşmuş bir alanda etkin gözetleme ve hedef takibi yapabilmek için bir kapsama stratejisi önermiştir. Bu çalışmada sensörlerin yerleştirileceği alan şekil 2.1’de gösterildiği gibi ızgara (grid) noktalarından oluşan bir yapıdadır ve hedefler bu noktalardan herhangi biri üzerinde bulunabilirler. Bu yaklaşımda alan 2 boyutlu olarak tanımlanmış ve engellerden etkilenmediği kabul edilmiştir. Tanımlanan alanının tamamen kapsanması (coverage) için minimum maliyet ile sensörlerin yerleşimini gerçekleştiren bir tam sayı lineer programlama (Integer Lineer Programming) çözümü önerilmiştir.



Şekil 2.1 İki boyutlu ızgara noktalarından oluşmuş sensör alanı

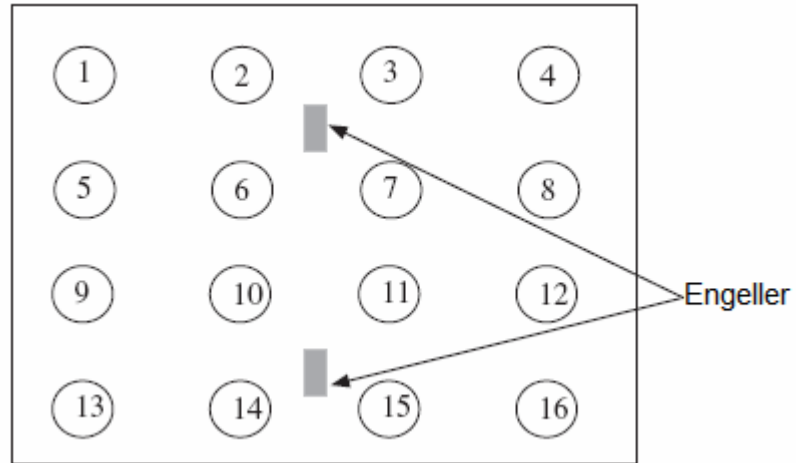
Dhillon ve Chakrabarty [2] sensörlerin bir alana etkin bir şekilde yerleştirilmesi için çalışmalar yapmışlardır. Bu optimizasyon probleminin çözümünde kullanılacak sensör sayısı ve kullanılan sensörlerin nerelere konuşlandırılacağı bulunmaya çalışılmıştır. Çalışmada ek olarak sensörlerin tespit edebilmelerine olasılıklı bir yaklaşım getirilmiştir. Geliştirilen algoritmalar kesin olmayan tespit yeteneği ve belirsiz sensör alanı özellikleri gibi kısıtları da göz önünde bulunduracak şekilde tasarlanmıştır.

Bu çalışmadan kullanılan sensör modelinin şekil 2.2’de gösterildiği gibi tespit yeteneği uzaklık ile azalan bir modeldir. d uzaklığındaki bir hedefin görülme ihtimali $e^{-\alpha d}$ şeklinde ifade edilmiştir. α parametresi sensörün tespit kalitesini ayarlayan parametredir [2].



Şekil 2.2 Olasılıklı sensör tespit modeli

Bir sensörün tespit olasılığını etkileyen bir başka etken ise şekil 2.3'te gösterilen sensör alanında bulunan engellerdir. Engeller, bir sensörün çizgisel görülebilirliğini doğrudan etkilemekte ve bir ızgara noktasında bulunan hedefin tespit edilme ihtimalini çok düşürmektedir.

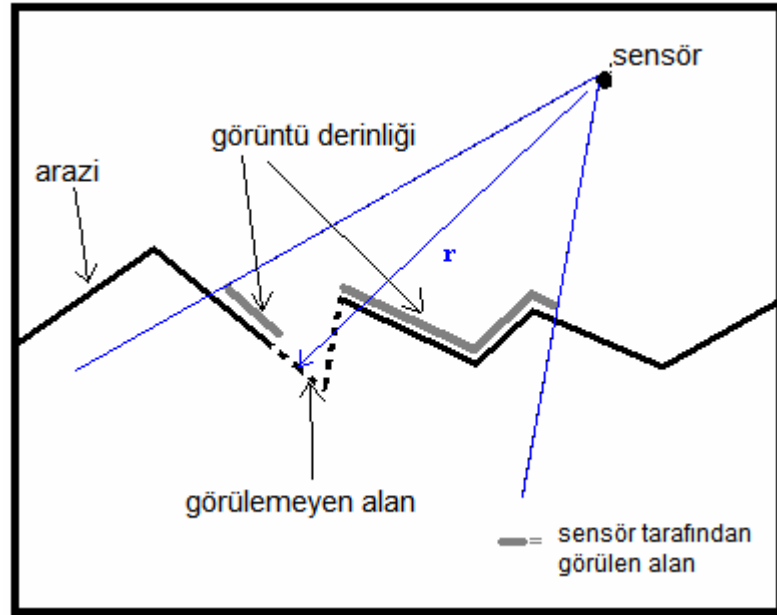


Şekil 2.3 Sensör alanında bulunan engeller

Can ve arkadaşları [3] yaptıkları çalışmada farklı tiplerdeki sensörlerin 3 Boyutlu (3D) bir arazide, istenilen düzeyde kapsamayı gerçekleştirecek şekilde yerleştirilmesini sağlayan bir yöntem geliştirmişlerdir. Bu çalışmada sensör modeli ileriye bakan kızıl ötesi (Forward Looking Infra-Red) kamera olarak tasarlanmıştır.

Geliştirilen sistemin temel amacı, tanımlı ortamda belirlenen bir alanın kısıtları sağlayacak şekilde güvenliğinin sağlanmasıdır.

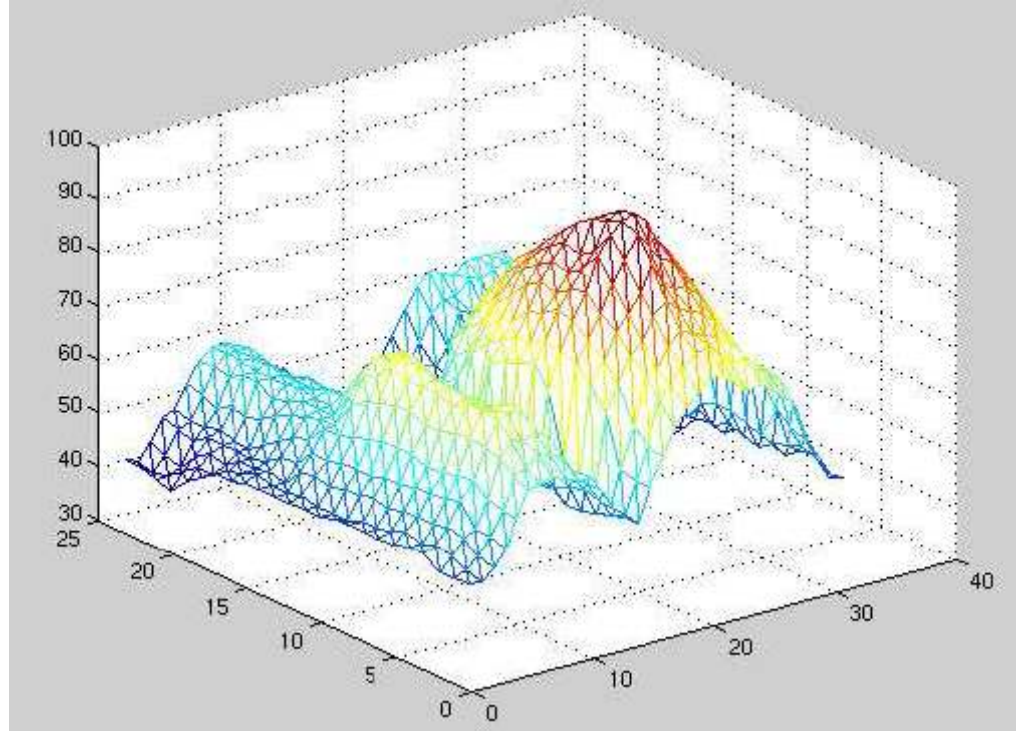
Bu çalışmada arazi modeli, 3D poligonlardan oluşan bir yapıdır. Her bir noktanın x , y , z koordinatları bulunmaktadır. Dolayısıyla bir noktadan başka bir noktayı görmek çizgisel görülebilirlik (Line of Sight) algoritmasıyla yapılmakta ve iki nokta arasında daha yüksek bir poligon bulunması durumunda görüş sağlanamamaktadır. Bu arazi yapısı aynı zamanda sensör modelinin davranışını da belirlemektedir. Şekil 2.4'te görüldüğü gibi bir sensörün doğrultusu ve bakış açısı bulunmaktadır. Bu açı içerisinde ve sensörün menzili dahilinde bulunan noktalar, eğer çizgisel görülebilirliği (LOS) sağlıyorsa görülmektedir. Bir arazinin görülebilirlik değeri, tüm sensörler tarafından görülebilen alanların toplamıdır. Bu problemin çözümünde genetik algoritmalar kullanılmıştır. Genetik Algoritma çözümünde kromozom yapısı sensörler bilgilerinden oluşan bir listedir. Sensör bilgisi olarak sensörün pozisyon bilgisi ve yerden yüksekliği tutulmaktadır. Ayrıca sensör bir hareketli platform üzerinde ise hareketli platformun başlangıç pozisyonu ve hareket yönü bilgileri de tutulmaktadır [3].



Şekil 2.4 Sensör görülebilirlik testi

Marengoni ve arkadaşları [4] çalışmalarında Müze Probleminin (Art Gallery Problem) bir türevi olan 3D versiyonuna bir çözüm yöntemi geliştirmişlerdir. Çalışmalarında problemlerini şu şekilde tanımlamışlardır: “Verilen 3D bir arazi

üzerine ne kadar ve nerelere sensör yerleştirelim ki tüm arazi kapsanmış olsun?”. Bu problemde arazi olarak DEM formatında sayısal yükseklik haritası kullanılmıştır. “Arazinin tümünün kapsanması” arazi üzerindeki tüm noktaların en az bir sensör tarafından görülmesidir. Arazi şekil 2.5’te gösterildiği gibi TIN (Triangulated Irregular Network) olarak temsil edilmiş ve bir noktadan bir başka noktanın görülebilirliği LOS algoritmalarıyla gerçekleştirilmiştir.



Şekil 2.5 TIN tabanlı arazi modeli

Çalışma sonucunda polinomsal zamanlı çalışan bir algoritma ile sensörler arazi üzerine yerleştirilmiş ve çözüme ulaşılmıştır.

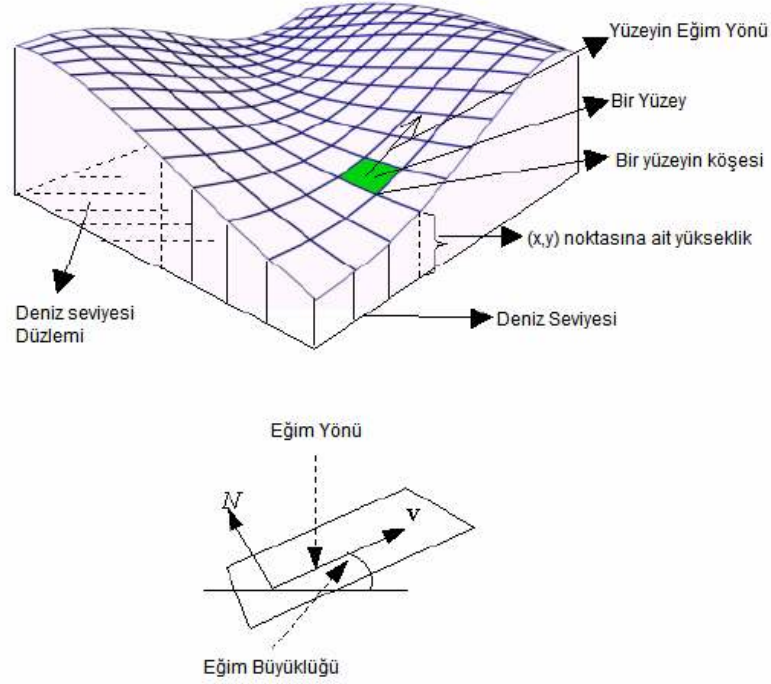
Marengoni ve arkadaşları [5] yukarıdaki çalışmalarını ilgili makalede geliştirmişler ve sonuç olarak tanımladıkları problemin daha farklı türevlerini araştırmaya açık yönler olarak belirtmişlerdir. Bu farklı versiyonlar şu şekilde tanımlanmıştır:

- “Verilen n adet sensörün, verilen bir arazi üzerine yerleştirilmesi sonucunda, arazinin tamamının kapsanması mümkün müdür? Eğer mümkün ise bu sensör pozisyonları nereleridir?”

- “Verilen n adet sensörü, verilen bir araziye, maksimum kapsamayı sağlayacak şekilde nerelere yerleştirebiliriz?”
- Problemin diğer bir versiyonunda sensörlerin konuşlanabileceği noktalara kısıt getirilebilmektedir. Arazi üzerinde düşman toprakları, deniz, göl vb. etmenlerin bulunmasından dolayı sensörlerin konuşlandırılabilmesi noktalara kısıt getirilebilir.
- Belirli sayıda sensör tarafından bir arazinin tamamı kapsanamamıştır. “Acaba arazinin tamamını kapsamak için ne kadar sensör nerelere yerleştirilmelidir?” sorusu ayrı bir problem olarak tanımlanmıştır.

Undeger ve arkadaşları [6] gözetleme işleminin askeri operasyonlardaki önemine dikkat çekmiş ve çalışmalarında geniş alanların ne kadar, hangi tipte sensör kullanarak ve bu sensörlerin nerelere konuşlandırılmasıyla en etkin sonuçları elde edebileceklerini araştırmışlardır. Sistemlerinde belirli görülebilirlik özelliklerine sahip gündüz kamerası, kızıl ötesi kamera ve radar gibi çeşitli sensör sistemleri modellenmiş ve görülebilirlik analizlerine imkan tanımışlardır. Bu problemin çözümü için bir genetik algoritma çözümü geliştirilmiştir. Kromozom yapısı sensör pozisyonlarından oluşan bir liste şeklinde tanımlanmıştır.

Bu çalışmada üç boyutlu (3D) arazi modeli kullanılmıştır. Arazi özellik olarak DTED formatında sayısal bir haritadır. Bu harita formatı gerçek dünyadaki yükseklik bilgisinin belirli örnekleme aralığıyla kaydedilmesi sonucunda ortaya çıkmıştır. Şekil 2.7’de görüldüğü gibi arazi modeli gerçek dünyadaki şekline yakın bir gösterime sahiptir.



Şekil 2.6 DTED formatındaki haritaya ait arazi modeli gösterimi

Ayrıca sensör yerleştirme alanı üzerinde her bir yüzeyin farklı derecede önemi ve sensörlerin bu yüzeye yerleştirilip yerleştirilemeyeceğinin bilgisi mevcuttur. Sensörler daha önce bahsedildiği gibi görülebilirlik hesaplarını LOS algoritmalarına göre yapmaktadır. Yapılan çalışmada sensörlerin zamandan bağımsız olarak gözetleme yapabileceği ve etrafını 360 derecelik bir açıyla görebileceği kabul edilmiştir.

Bu tez çalışmasında amaç fonksiyonu belirli sayıda sensör ile bir alanın maksimum seviyede kapsanmasıdır. Sensör modeli olarak belirli bir menzile sahip çok yönlü bir sensör modeli kullanılmış ve üç boyutlu bir arazi üzerinde görülebilirlik analizleri yapılmıştır.

3. META-SEZGİSEL ALGORİTMALAR

Pratikte pek çok optimizasyon problemi doğası gereği NP-Hard problemlerdir. Bilindiği gibi bu tür problemlerin çözümü için bilinen polinom zamanlı herhangi bir algoritma bulunmamaktadır. Dolayısıyla bu tür problemlerin çözümü için, optimal sonucu bulmayan fakat kabul edilebilir zaman içerisinde iyi sonuçlar bulan algoritmalarla çözümler önerilmiştir. Bu yaklaşım yerel arama ya da sezgisel yöntemlerin ortaya çıkmasını sağlamıştır. Sezgisel yöntemler, algoritma tasarımcı tarafından geliştirilen deneme yöntemleriyle sonuca ulaşmaya çalışan yöntemlerdir.

Meta-sezgiseller, problemleri sezgisel olarak çözmek için kullanılan genel çatılardır (framework). Meta-sezgisel algoritmalar temel olarak bir fikre veya örneklemeye (anology) dayanır ve çeşitli yönlerde genişletilerek yada melezleştirilerek geliştirilir. Bu yöntemlerin temel amacı geliştirilen yöntemlerin yerel optimum çözümlerinden kaçınacak metotların bulunmasıdır. Genellikle bu metotlar mevcut çözüm üzerinde kabul edilebilir değişiklikler yaparak farklı çözümler ararlar. Mevcut çözüm üzerinde yapılan bu değişiklikler yapıcı (constructive) yada yıkıcı (destructive) olabilmektedir. Örneğin Genetik Algoritmalarda seçilen iki çözüm arasında bileşenler yer değiştirilerek bilgi paylaşımı yapılır ve böylelikle daha iyi bireylerin oluşması beklenir ya da mevcut bir çözüm üzerindeki bir bileşenin değeri rasgele değiştirilerek yerel optimumdan kaçınılabılır. Sonuç olarak çıkan sezgisel yöntem kompleks hale gelerek pek çok parametreye bağlı hale gelir. Bu durum meta-sezgisel yöntemleri etkin kılar fakat aynı zamanda başarılı olma durumunu belirsiz hale getirir. Meta-sezgisel yöntemler genel olarak çözüm yaklaşımları sunarlar. Mevcut çözümün nasıl değiştirileceği Meta-sezgisel yöntemler belirli bir probleme ait çözüm sunmadığı için kullanıcı tarafından probleme uyarlanması gerekmektedir. Bu uyarlama kısımları sezgisel olarak inşa edilir ve deneylerle uyarlamanın performansı ölçülür.

3.1. Optimizasyon Problemleri

Verilen bir $f(x)$ fonksiyonu ve kısıtlar için, $f(x)$ fonksiyonunu minimum yapacak x değerini bulmak amacıyla bir optimizasyon problemi formül 3.1 de gösterildiği gibi tanımlanabilir.

$$x_{opt} = \arg \min \{f(x) \mid x \in X, X \subseteq S\} \quad (3.1)$$

S , X , x , f ve x_{opt} değerleri sırasıyla çözüm uzayı, olurlu bölge(feasible set), olurlu çözüm(feasible solution), gerçek değerli amaç fonksiyonu(real valued objective function) ve optimal çözüm olarak tanımlanmıştır. Eğer S sonlu fakat çok geniş ise problem birleşimsel (combinatorial) bir optimizasyon problemi olarak tanımlanmıştır. Eğer herhangi bir n tamsayısı için $S = R^n$ ise problem sürekli optimizasyon problemi olur. Çoğu optimizasyon problemi çözüm uzayının çok büyük olması ve kolay bir şekilde kısıtlamaması gibi sebeplerden dolayı NP-Hard'dır ve bu problemleri çözmek için sezgisel metotlar kullanmak gerekmektedir.

Problemin çözümün uzayının temsili problemin türüne göre farklı gösterim biçimlerinde olmalıdır. Örneğin gezgin satıcı probleminde (TSP) çözüm bir şehirden başlayarak tüm şehirlerden bir kere geçen, tüm şehirleri gezen ve tekrar başlangıç şehrine dönen bir şehir dizisidir. Başka bir örnek olarak optimize edilecek n boyutlu sürekli bir fonksiyonun çözüm temsili n 'lik bir reel sayı dizisidir.

3.2. Değişken Komşuluk Arama Metodu

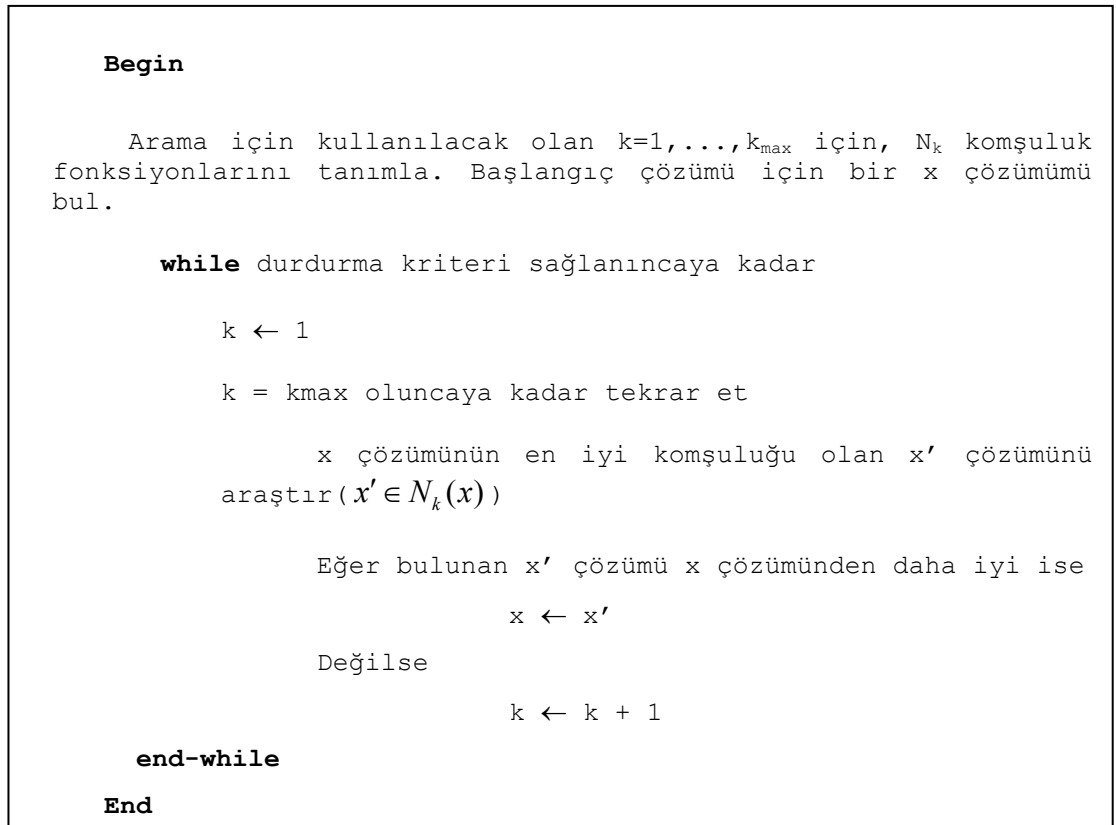
Değişken Komşuluk Arama Metodu (VNS - Variable Neighborhood Search) global optimizasyon problemlerinin çözümü için kullanımı son zamanlarda artan yerel arama tabanlı bir meta-sezgisel yöntemdir. VNS yönteminde temel fikir komşuluk yapısının arama sırasında sistematik bir şekilde değişmesidir [7]. Bu kısımda VNS yönteminin temel adımları ve türevleri anlatılacaktır.

N_k , ($k=1, \dots, k_{max}$) daha önceden belirlenmiş sonlu sayıda komşuluk yapıları ve $N_k(x)$, x çözümünün k 'inci komşuluğu olsun. Herhangi bir komşuluk S çözüm uzayında bir veya birden fazla metriğe göre belirlenmiş çözümler kümesidir. Eğer $x \in N_k(x') \subseteq X$ için $f(x) < f(x')$ şartını sağlayan bir x çözümü yok ise $x' \in X$ (1) numaralı denklemin k 'inci komşuluğa göre *yerel minimumudur*. Yerel arama tabanlı meta-sezgiseller ilk yerel minimumun bulunmasıyla başka yöntemlerle bu yerel minimumdan kurtulmaya çalışırlar.

Yerel minimumdan kurtulmak için VNS yöntemi komşuluğunu değiştirmektedir. Bu komşuluk değiştirme işlemi 3 şekilde yapılmaktadır. Bunlar sırasıyla deterministik (gerekirci), stokastik, deterministik ve stokastik yöntemler şeklinde sıralanabilir [7].

3.2.1. Değişken Komşuluk İnişi

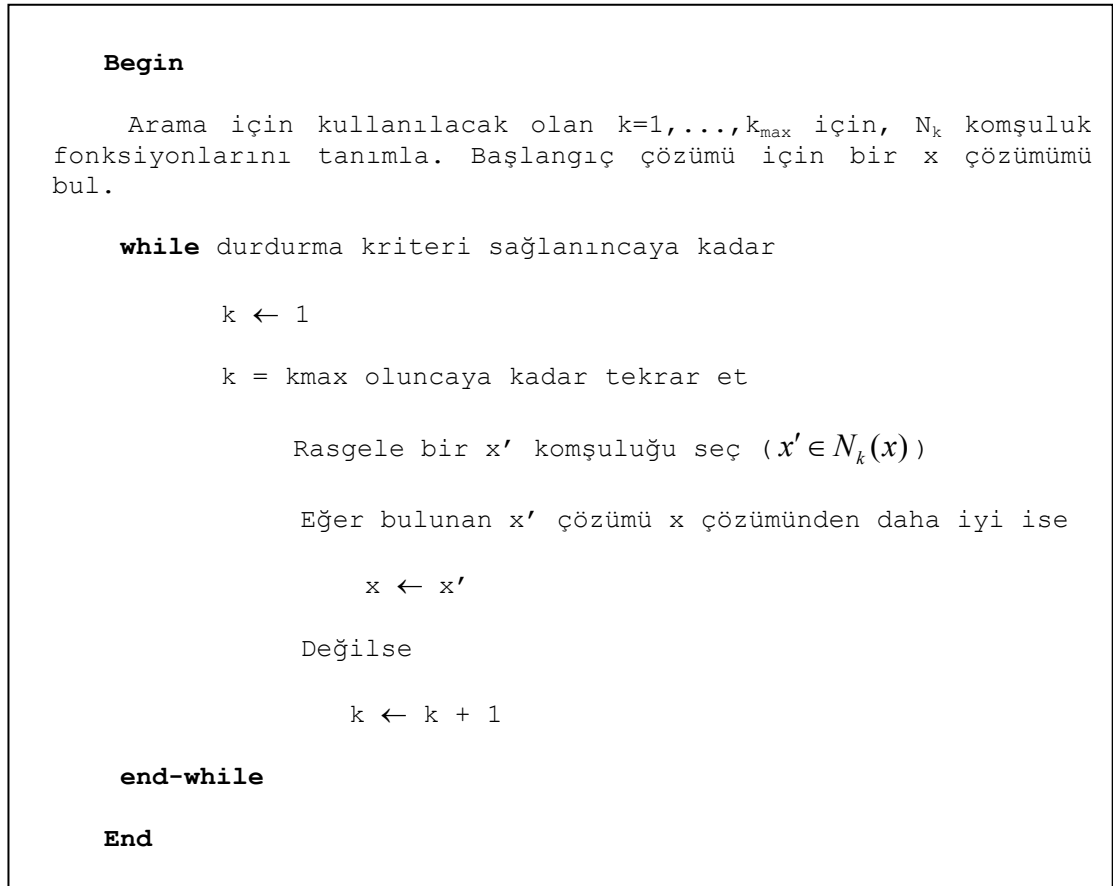
Değişken Komşuluk İnişi (VND - Variable Neighborhood Descent) komşuluk değişimlerinin deterministik olarak belirlendiği bir yöntemdir. Bir iterasyondaki komşu seçimi, komşuluk içerisindeki tüm komşuların aranması sonucunda en iyi bireyin seçilmesi şeklinde olmaktadır. Bu sebeple bu yöntem kullanılarak bulunabilecek en iyi sonucun tüm komşuluk yapılarına göre yerel minimum olması gerekmektedir. VND yönteminin temel adımları şekil 3.1’de gösterilmiştir.



Şekil 3.1 VND modeli

3.2.2. İndirgenmiş Değişken Komşuluk Arama Metodu

İndirgenmiş Değişken Komşuluk Arama Metodu (RVNS - Reduced Variable Neighborhood Search) VND metodunun aksine komşuluğun rasgele seçimine dayanır. RVNS metodu çok büyük problem örnekleri için yerel aramanın çok maliyetli olduğu durumlarda kullanışlıdır. Yapılan gözlemlere göre en uygun $k_{\max}$ değeri 2 olarak tavsiye edilmektedir. RVNS yönteminin temel adımları şekil 3.2’de gösterilmektedir.

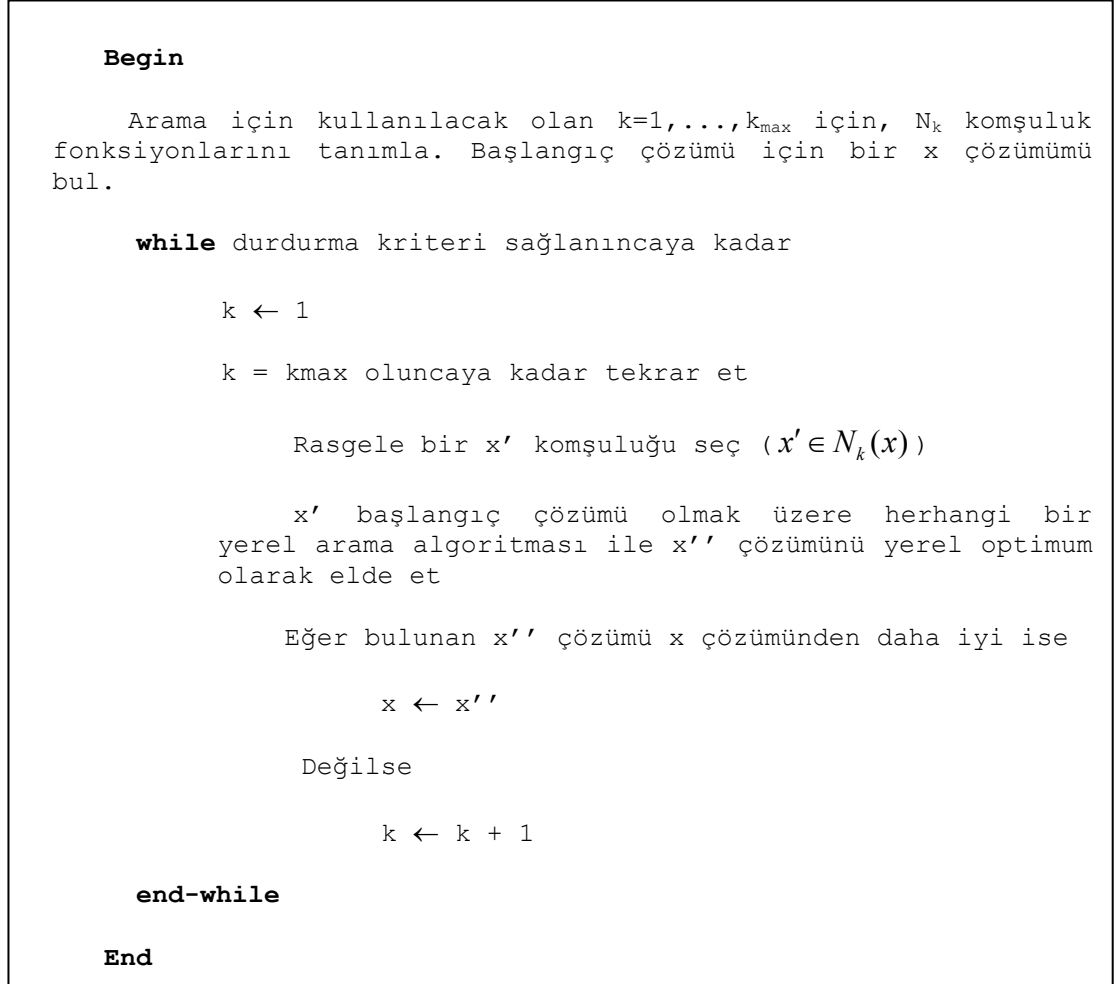


Şekil 3.2 RVNS modeli

3.2.3. Değişken Komşuluk Arama Metodu

Değişken Komşuluk Arama Metodu (VNS) deterministik ve olasılıklı komşuluk değiştirme yaklaşımını kullanır. Yöntemin temel basamakları şekil 3.3’te gösterilmiştir.

Böylelikle başarılı N_k komşuluğu ile daha detaylı arama yapılmış olunur. Yöntemin yerel arama mekanizması olarak VND, RVNS yada başka bir yerel arama algoritması seçilebilir.



Şekil 3.3 VNS modeli

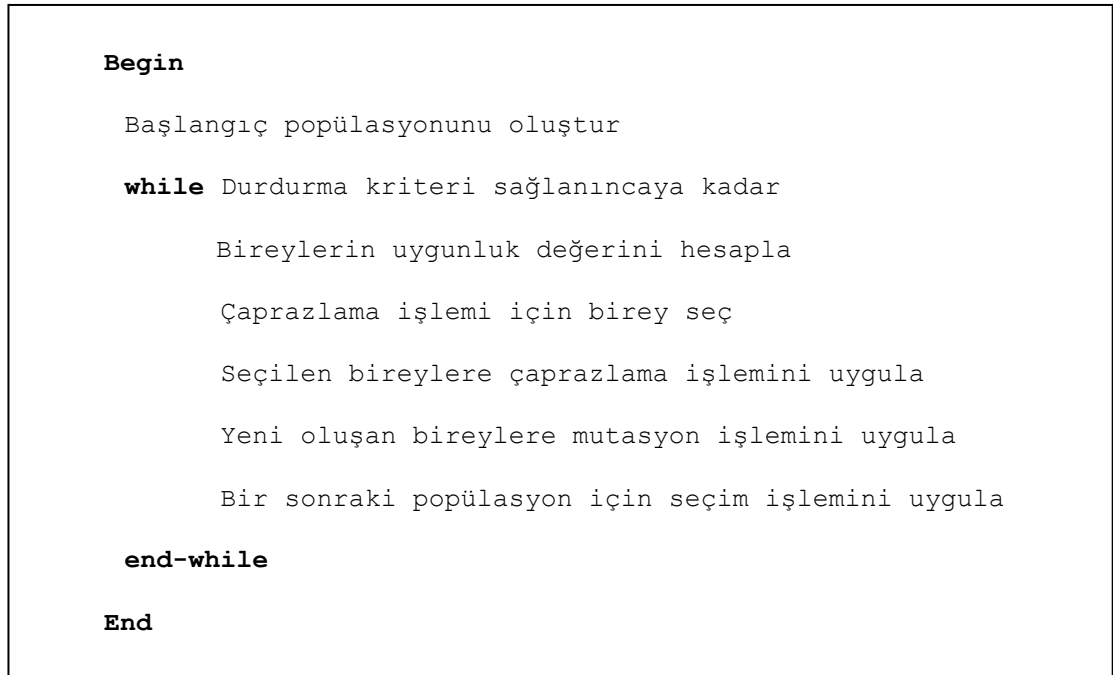
3.3. Genetik Algoritmalar

Genetik Algoritmalar, doğal seçim ilkelerine dayanan evrimsel bir arama ve optimizasyon yöntemidir. Temel ilkeleri John Holland(1975) tarafından ortaya atılan genetik algoritmalar pek çok probleme uygulanmış ve başarılı sonuçlar elde edilmiştir. Çözüm uzayının tamamını taramayıp sadece belirli bir kısmını taramaları ve bu alanda etkin arama yapmaları kısa sürede sonuca ulaşmayı sağlar [8]. Ayrık ve sürekli değişkenli fonksiyonları optimize edebilmeleri, çok fazla değişkene sahip

problemlere uyarlanabilmeleri, yerel optimum çözümünden kaçabilmeleri ve kolay paralelleştirilebilmeleri genetik algoritmaların en önemli avantajlarından birkaçıdır.

3.3.1. Basit Genetik Algoritma

Genetik algoritmalar, popülasyon temelli bir optimizasyon yöntemidir. Popülasyon belirli bir sayıdaki bireyi ve her bir birey bir aday çözümü temsil eder. Her bir çözüm daha önce belirtilen olurlu bölgeden (X) seçilmiş olurlu bir çözümdür (x). Problemin genetik algoritmaya uyarlanması için çözüm uzayının gösterimi, çaprazlama ve mutasyon operatörü, seçim stratejisi gibi temel adımların tanımlanması gerekmektedir. Genetik algoritmalar çözüme genelde rasgele seçilmiş bir popülasyon ile başlarlar. Belirli bir durdurma kriterine göre popülasyondan bireyler seçilir ve çaprazlama işlemine tabi tutulurlar. Bu işlemden sonra rasgele seçilen bireyler mutasyon işlemine tabi tutulur ve bu şekilde çözüm arayışı devam eder. Şekil 3.4'te basit bir genetik algoritmanın temel adımları gösterilmiştir.



Şekil 3.4 Temel Genetik Algoritma adımları

3.4. Genetik Algoritmelerde Problemin Temsil Mekanizması

Genetik algoritmelerde problemin gösterimi kromozom olarak adlandırılır. Her bir kromozom problemin çözüm uzayındaki (S) rasgele bir çözümü (x) temsil eder. En yaygın kromozom biçimi ikili bit dizisi şeklindedir. Bu yaklaşımda kromozom büyüklüğü probleme bağlıdır ve N adet değişkene sahip olan bir kromozom $[p_1, p_2, p_3, \dots, p_N]$ şeklinde gösterilmektedir. p değerleri ikili kodlamada 0 yada 1 değerini almaktadırlar. Problemlerin gösterim biçimi iki gösterime uygun olmak zorunda değildir. Problemi çözümü önce iki sistemde temsil edilebilecek şekilde dönüşüm gerçekleştirilir ve genetik algoritmanın bulduğu sonuç geri dönüştürülerek çözüm bulunmuş olunur.

3.5. Başlangıç Popülasyonunun Oluşturulması

Popülasyon belirli bir sayıda kromozom içeren bir kümedir. Genetik algoritmelerde popülasyon büyüklüğü (N_p) çözümün kalitesini değiştirebilecek etmenlerden biridir. Genel kullanım popülasyonun rasgele seçilmiş kromozomlardan oluşmasıdır. Böylelikle farklı kalitede çözümlerle çözüme başlanmış olunur ve yerel optimum değerlerinde kaçma ihtimali elde edilir. Öte yandan uygunluk değeri daha önceden iyi olduğu bilinen birkaç bireyin popülasyona dahil edilmesi erken sonuca varmaya önemli bir katkı sağlayacaktır.

3.6. Uygunluk Değerinin Hesaplanması

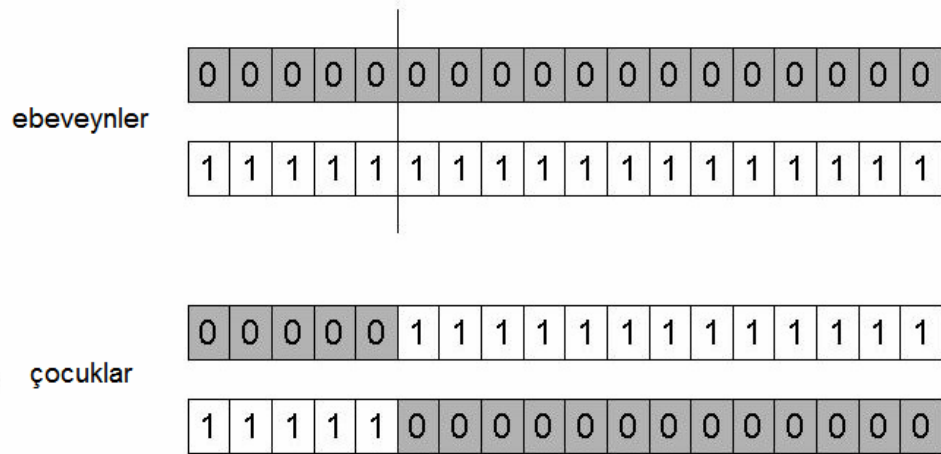
Genetik algoritmanın her iterasyonunda her x çözümü (kromozom) için uygunluk değeri olan $f(x)$ değeri hesaplanır. İyi bireylerin nesilden nesile aktarılması ve kaliteli bireylerin bilgilerini daha çok paylaşması için uygunluk değerine ihtiyaç vardır. İlerleyen bölümlerde görüleceği gibi tüm kromozom seçim işlemleri popülasyondaki her bir bireyin etkinliğine bağlı olarak yapılmaktadır. Bu durumda her bireyin uygunluk değerinin hesaplanmasını gerektirir.

3.7. Çaprazlama İşleminin Uygulanması

Bir popülasyondaki bireylerden gelecek nesil için daha iyi nitelikler içeren bireylerin oluşması için çaprazlama işlemi uygulanmaktadır. Popülasyon havuzundan seçilen iki kromozom melez yapıda bireyler oluşturmak hedefiyle çaprazlama işlemine girer ve yeni iki kromozom meydana gelir. Çaprazlama operatörünün hedefi iyi çözümlerin ortak bilgisinin kullanılarak daha iyi bireylerin meydana gelmesine olanak sağlamaktır. Seçilen bireylerin hepsine çaprazlanma uygulanmaz. Bu durum $[0,1]$ arasında bir değer alan çaprazlama oranı değişkeni ile kontrol edilir ve bu oranın 1 olması durumunda tüm seçilen kromozomlar çaprazlamaya tabi tutulur.

Yeni ortaya çıkan kromozomlar (çocuklar) ya doğrudan bir sonraki nesle aktarılır yada eski kromozomlar (ebeveyn) ile ortak bir havuza konarak yeni popülasyon için seçilmek üzere tutulurlar.

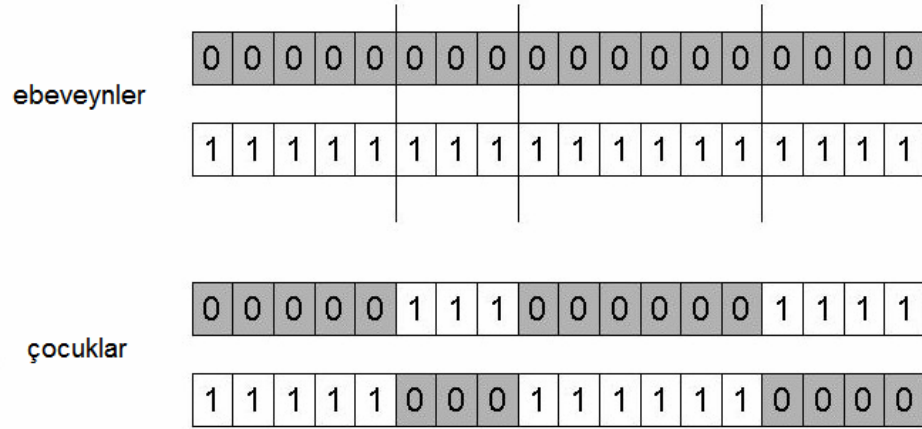
En basit çaprazlama operatörü tek nokta çaprazlamadır. En fazla değeri kromozom dizi büyüklüğü (N) olan rasgele bir c sayısı seçilir. Birinci çocuk için bu noktaya kadar birinci ebeveyninden gelen değişkenler atanır, kalan diğer değişkenler ikinci ebeveyninden alınır. İkinci çocuk için bu durum tam tersidir. Bu durum şekil 3.5'te gösterilmektedir.



Şekil 3.5 Tek nokta çaprazlama sonucu oluşan kromozomlar

Diğer bir çaprazlama operatörü n noktalı çaprazlamadır. Tek noktadakine benzer bir şekilde seçilen n adet kesim noktasına göre çocuklar meydana gelir.

Kesim noktalarının aralarında kalan genler bir çocuk için sırayla birinci ebeveyn ve ikinci ebeveyninden seçilir. Bu durum şekil 3.6’da gösterilmektedir.



Şekil 3.6 Üç nokta çaprazlama sonucu oluşan kromozomlar

3.8. Mutasyon İşleminin Uygulanması

Doğal yaşamdaki genetik mutasyondan esinlenerek ortaya çıkmış bir operatördür. Bu operatörün genetik algoritmanın bulduğu sonuçlara etkisi çok büyüktür. Mutasyon operatörünün uygulandığı kromozomda, kromozomun çeşitli değerleri değişime uğrar. Genetik algoritmada mevcut popülasyonda bulunan çözüm elemanlarından farklı yeni değerlerin çözüme dahil edilmesi ancak mutasyon ile mümkündür ve bu operatör olmaksızın optimum çözüme oluşmak mümkün olmayabilir. Bir kromozomun mutasyona uğrama ihtimali genetik algoritmada $[0,1]$ aralığında bir sayı olan mutasyon oranı ile kontrol edilir. Bu değer 1 olması durumunda tüm bireyler mutasyona uğratılmaktadır. Ancak kullanım olarak bu oran küçük değerler almaktadır.

Mutasyon işlemi bir kromozom yapısının nasıl olduğuna bağlı olduğu için çeşitli şekillerde uygulanabilmektedir. İkili kodlamaya sahip bir kromozomda, bir genin değerleri tersine dönüştürülür. Yani şekil 3.8’de gösterildiği gibi 0 olan değer 1, 1 olan değer 0 olarak atanır.

ebeveyn	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
çocuklar	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

Şekil 3.7 İkili kodlamada mutasyon işlemine uğrayan bir kromozom

3.9. Tekrar Üreme

Mevcut popülasyondan bir sonraki popülasyonun üretilmesine tekrar üreme denir. Bu işlemin yapılması için çeşitli seçim stratejileri bulunmaktadır. Bu operatörde temel amaç uygunluk değeri yüksek olan bireylerin hayatta kalmaları ve düşük nitelikli bireylerin zamanla azalarak kaybolmalarıdır. Bu işlem için çeşitli seçim teknikleri vardır.

Rulet tekerleği seçiminde popülasyondaki her bir bireyin tekerlek üzerinde uygunluk değeri oranında payı bulunmaktadır. Tekerleğin döndürülmesi sonucunda bir birey seçilir. Böylelikle uygunluğu yüksek bireylerin daha fazla seçilme ihtimali olur. Bir başka metot olan turnuva seçiminde rasgele seçilen n adet bireyden en iyisi seçilir. Böylelikle kötü bireylerin bir sonraki nesle aktarılma ihtimali ortadan kaldırılmış olunur. Ayrıca popülasyonda bulunan en iyi bireyler belirli oranda doğrudan bir sonraki nesle aktarılarak en iyi çözümler korunmuş olunur. Bu yönteme elitist seçim denilmektedir.

3.10. Parametrelerin Seçimi

Parametrelerin doğru seçilmesi, genetik algoritmaların iyi sonuç vermesi için önemli bir yere sahiptir. Tüm problemler için genel bir parametre seçiminden bahsetmek çok güçtür. Bu sebeple problem üzerinde çeşitli deneyler yaparak en uygun parametreleri bulmak gerekmektedir. Bir genetik algoritmada kontrol parametrelerini aşağıdaki şekilde sıralayabiliriz:

- Popülasyon büyüklüğü
- Çaprazlama oranı

- Mutasyon oranı
- Seçim stratejisi
- Kuşak aralığı (her kuşaktaki kromozom oranı)
- Elitist seçim oranı

Gerekli uyarlamanın yapılması ve doğru parametrelerin atanması sonucunda genetik algoritma ile bir optimizasyon probleminin iyi sonuçlar vererek çözümü mümkün olabilecektir.

3.11. Parçacık Sürü Optimizasyonu

Son yıllarda kullanım alanı giderek artan ve 1995 -1996 yıllarında Kennedy ve Eberhart tarafından ortaya atılan Parçacık Sürü Optimizasyonu (PSO) tekniği kuş ve balık sürülerinin hareketlerinden esinlenerek ortaya atılmış popülasyon temelli sezgisel bir yöntemdir [10].

PSO tekniği genetik algoritmalara benzer şekilde çalışan bir yöntemdir. Algoritma rasgele çözümlerden oluşmuş bir başlangıç popülasyonu (sürü) ile başlatılır ve çözüm uzayında optimum değer sürünün farklı yerlere hareket etmesiyle araştırılır. Ancak PSO yöntemi genetik algoritmalardan farklı olarak çaprazlama ve mutasyon operatörüne sahip değildir. PSO yönteminde çözümler parçacık (particle) olarak adlandırılır ve mevcut optimum değerler izlenerek çözüm uzayında arama yapılır. Bu durum doğal hayatta kuş sürülerini havada uçmasına benzetilmektedir.

3.12. PSO Çalışma Mekanizması

Yukarıda bahsedildiği gibi PSO algoritması kuşların uçuşundan esinlenerek ortaya atılmış bir yöntemdir. Senaryo olarak kuşların bir alanda rasgele yiyecek araması seçilebilir. Bu alanda sadece bir adet besin kaynağı bulunmaktadır ve bütün kuşlar bu kaynağın nerede olduğunu bilmemektedirler. Ancak kaynaktan ne kadar uzakta olduklarının bilgisi mevcuttur. Bu durumda kuşların en uygun hareket etme stratejisi tartışmaya değer bir konudur. Bu durum bir optimizasyon probleminin çözümünde nasıl kullanılabileceği PSO'nun çıkış noktasıdır. PSO'da her bir çözüm,

çözüm uzayında bir kuşu temsil eder ve parçacık olarak adlandırılır. Her bir parçacık bir uygunluk değeri ve uçuşlarını kontrol eden vektörel hıza sahiptir. Bir parçacığın bir sonraki pozisyonu sahip olduğu hızdan etkilenmektedir.

Yöntemin temel basamaklarından birisi başlangıç popülasyonunun oluşturulmasıdır. Genel olarak rasgele pozisyonlara yerleştirilmiş ve rasgele hızlarla başlatılmış bir sürü olarak tanımlanır. Daha sonraki adımlarda her bir parçacığın pozisyon bilgisi iki en iyi değere bakılarak güncellenir:

- her bir bireyin daha önce bulunmuş olduğu en iyi pozisyon bilgisi (pBest)
- tüm sürüde bulunan ve şimdiye kadar bulunmuş en iyi parçacığın pozisyon bilgisi (gBest)

Tüm parçacıklar hız vektörleriyle bu iki en iyi pozisyona ulaşmaya çalışırlar. Bir parçacığın hızı v , pozisyonu p olarak tanımlanırsa algoritmanın t . iterasyonu sonucunda yeni hız ve pozisyon bilgisi şu şekilde güncellenir:

$$v_{t+1} = wv_t + c_1r_1(p_{best} - p_t) + c_2r_2(g_{best} - p_t) \quad (3.2)$$

$$p_{t+1} = p_t + v_{t+1} \quad (3.3)$$

Yukarıdaki denklemlerde c_1 ve c_2 öğrenme faktörleri olmakla birlikte genel kullanım olarak 2 değerini alırlar. r_1 ve r_2 , $[0,1]$ aralığında düzgün dağılımlı rasgele sayılardır. w değeri ise atalet faktörü olarak adlandırılır ve zamanla değeri düşürülerek daha önceki hız bilgisinin etkisinin azaltılması, kullanılan yöntemlerden biridir. Şekil 3.9'da standart bir PSO algoritmasının komut dizimi görülmektedir:

Begin

Başlangıç parametrelerini ata

Başlangıç popülasyonunu oluştur

Her bir parçacık için

Uygunluk değerini hesapla

pBest'i bul

gBest'i bul

Do

Her bir parçacık için

(3.2)denklemine göre parçacık hızını hesapla

(3.3)denklemine göre parçacık pozisyonunu hesapla

Uygunluk değerini hesapla

pBest'i güncelle

gBest'i güncelle

w değerini güncelle

While Durdurma Kriteri**End****Şekil 3.8** Temel PSO adımları**3.13. PSO Parametre Kontrolü**

Bir optimizasyon probleminin PSO yöntemine uyarlanması için iki önemli tanımın yapılması gerekmektedir: çözümün gösterimi ve uygunluk fonksiyonu. PSO algoritmasının en büyük avantajlarında biri parçacıkların reel sayı olarak tanımlanmasıdır. Dolayısıyla sürekli fonksiyon optimizasyonu problemleri için hiçbir dönüşüm işlemi gerçekleştirmeye gerek yoktur.

PSO yönteminde çoğu meta-sezgisel algoritmada olan fazla parametre miktarı gibi bir dezavantaj bulunmamaktadır. Aşağıda bir PSO yöntemi için ayarlanması gereken parametreler anlatılmaktadır.

Başlangıç parçacık miktarı: Tipik olarak 20–40 arasında seçilen miktarda parçacıkla başlanılır. Bu miktar problemin büyüklüğüne göre daha düşük ya da yüksek miktarlardan seçilmelidir.

Parçacıkların boyutu: Bu değişken probleme bağlı bir değişkendir. Optimize edilecek fonksiyon kaç boyutlu ise parçacıkların boyutları da bu miktarda olacaktır.

Parçacıkların değer aralığı (range): Bu parametre de probleme bağlı bir parametredir. Optimize edilecek değişkenlerin değer aralığını belirtir.

Vmax: Bir parçacığın alabileceği maksimum hız aralık değeridir. Örneğin $[-5,5]$ aralığında hız değeri alabilen bir parçacığın V_{max} değeri 10'dur.

Öğrenme Faktörleri: c_1 ve c_2 genel olarak 2 seçilir. Ancak bu değerler $[0,4]$ aralığında olabilmektedir.

Durdurma Kriteri: Süre, maksimum sayıdaki iterasyon, belirli bir adım sayısınca gBest değerinin değişmemesi, belirli sayıda uygunluk fonksiyonunun hesaplanması gibi kriterler durdurma kriteri olarak seçilebilir.

Atalet Faktörü ve Azalma Katsayısı: 3.2 numaralı denklemde geçen ve eski hız bilgisinin yeni hız bilgisine etkisini ayarlayan parametredir (w). Bu parametrenin zamanla değeri küçülerek eski hız bilgisini etkisinin azaltılması azalma katsayısı (k) ile gerçekleşmektedir. w değeri aşağıdaki denkleme göre her iterasyonda güncellenir.

$$w_{t+1} = w_t \times k \quad (3.4)$$

4. PROBLEMİN FORMÜLE EDİLMESİ VE KULLANILAN MODELLER

Bu bölümde probleme girdi teşkil eden temel elemanların nasıl modellendiği anlatılmaktadır.

4.1. Arazi Modeli

DTED (Digital Terrain Elevation Data) formatı, arazi yükseklik verilerine ihtiyaç duyan çeşitli sistemler ve uygulamalar için standart bir harita formatıdır. Amerikan Milli Görüntüleme ve Haritalama Ajansı (NGA) tarafından geliştirilmiş standartları belirlenmiş bir harita formatıdır. DTED formatı 1970 yılından beri gerek askeri gerek sivil birçok uygulamada kullanılmış ve halen yaygın bir şekilde kullanılmaktadır. Bu uygulamalar genellikle görülebilirlik analizleri, arazi profili çıkartma, 3 boyutlu arazi gösterimi, görev/senaryo hazırlama, modelleme ve simülasyon programları gibi uygulamaları kapsamaktadır [14]. DTED formatı seviye 0, 1 ve 2 olarak sunulan sırasıyla yaklaşık 1000, 100 ve 30 metre çözünürlüğe sahip sayısal yükseklik haritalarından oluşmaktadır. Bu formatlara ait çözünürlük bilgileri tablo 4.1’de verilmiştir. Çok genel uygulamalara hitap eden seviye-0 tipindeki sayısal yükseklik haritaları ücretsiz olarak kullanıma sunulmuştur. Tez kapsamında yapılan çalışmalar, NGA’dan satın alınan DTED seviye-1 dosyaları üzerinde yapılmıştır. İlerleyen kısımlarda detayları verilen görülebilirlik hesabı vb. gibi işlemler, 2 boyutlu bir matris içerisinde saklanan, 100 metre çözünürlüklü DTED seviye-1 dosyalarından okunan yükseklik verileri üzerinde yapılmıştır. Bu verilere ait çözünürlük, görülebilirlik hesabının daha kesin sonuçlar verebilmesi için doğrusal ara değerlendirme yöntemi ile daha da arttırılmıştır.

Tablo 4.1 DTED formatındaki haritaların özellikleri

DTED Seviyesi	Örnekleme Aralığı	Yeryüzü Mesafesi	Satır x Sütun	Yerleşim Büyüklüğü
0	30 sec	~1000 m	120 x 120	1 x 1 derece
1	3.0 sec	~100 m	1200 x 1200	1 x 1 derece
2	1.0 sec	~30 m	3600 x 3600	1 x 1 derece

Verilen sayısal harita probleme girdi teşkil edecek araziyi temsil etmemektedir. Verilen sayısal harita üzerinde daha küçük alanlarda da problemin çözümü mümkündür. Probleme girdi teşkil edecek arazinin en, boy ve köşe koordinatları belirlenmiş olmalıdır. Ekran üzerinde bir dikdörtgenin sol üst köşesi ve dikdörtgenin en ve boy bilgisinin belirlenmesiyle arazi tanımlanmış olur. Harita formatının kısıtlamasıyla köşe bilgisi gerçek dünya koordinatlarında (enlem, boylam) tanımlanmalıdır. Örneğin bir arazi “E008-N37,100x150” şeklinde tanımlanırsa köşe koordinatı 8 derece doğu meridyeni, 37 derece kuzey paraleli üzerinde en 100 km, boy 150 km olarak tanımlanmış olunur.

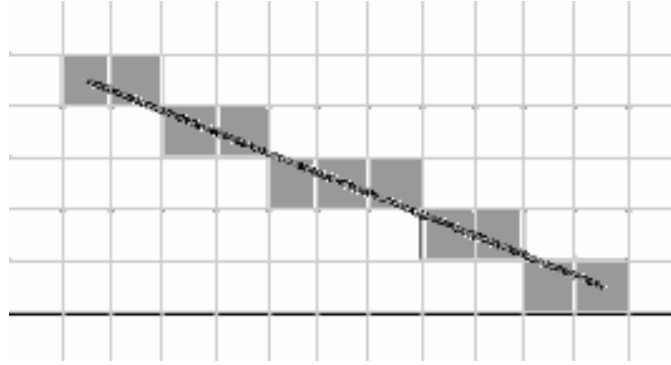
4.2. Çizgisel Görülebilirlik

Çizgisel görülebilirlik, bir yüzey üzerinde seçilen iki referans noktadan birinde bulunan bir gözlemci tarafından, hedefteki diğer noktanın görülüp görülemediğinin tespit edilmesidir. Ancak hedefte her zaman bir nokta bulunmayabilir, onun yerine bir alan veya bir nesne de olabilir. Bu durumda kullanılacak olan çizgisel görülebilirlik yöntemi hedefteki bir tek nokta yerine noktalar kümesini incelemek durumunda kalacaktır. Örneğin bir algoritmada, verilen bir arazi üzerinde referans iki noktanın birisinden diğerine gönderilen bir ışının, bu iki nokta arasında kalan yeryüzü yükseltilerini kesip kesmediğine bakılarak ve sonuç olarak görüyor veya görmüyor şeklinde bir cevap döndürülebilir. Diğer bir algoritmada ise gönderilen birkaç ışının, arazi yükseltilerine ek olarak, doğrultuları üzerinde bulunan ağaç, bina, otomobil vb. nesnelerin, görülmek istenen nesnenin veya alanın görülmesini ne kadar engellediği de hesap edilebilir ve sonuç olarak görülebilirlik belli bir yüzde ifadesi ile döndürebilir [11].

Bu çalışmada, literatürde Bresenham Çizgi Çizme (Bresenham's Line Drawing) algoritması olarak bilinen [13], bilgisayar grafiklerinde sıklıkla kullanılan, oldukça hızlı çalışan bir çizgi çizme algoritmasının üzerine bina edilmiş, ızgara tabanlı arazi verilerini kullanan Bresenham Çizgisel Görülebilirlik algoritması kullanılmıştır. Bu algoritma, tamsayı aritmetiğini kullandığı için oldukça hızlı çalışmaktadır.

4.3. Bresenham'ın Çizgisel Görülebilirlik Yöntemi

Bresenham'ın çizgi çizme algoritması, 2-boyutlu bir ızgara üzerinde verilen iki nokta arasındaki bir doğruyu, ızgara üzerindeki noktalar ile mümkün olduğunca düz bir çizgi şeklinde ifade etmeye yarayan bir algoritmadır (Şekil 4.1). Bilgisayar grafikleri alanında geliştirilen ilk algoritmalarından birisidir. Genellikle bilgisayar ekranına çizgi çizdirmek için kullanılır. Günümüzde halen grafik kartlarına ait grafik yonga setlerinde ve birçok grafik kütüphanesinde kullanılmaktadır. Çok sade bir algoritma olmasından dolayı sadece yazılımsal olarak değil, birçok grafik kartında bulunan grafik işlemcilerinde donanımsal olarak da desteklenmektedir.



Şekil 4.1 Bresenham çizgi algoritmasının gösterimi

Bresenham algoritması tam sayılarla (integer) toplama, çıkarma ve ikiyle çarpma işlemlerini içerir. İkiyle çarpma işlemi kaydırma (shift) operasyonu ile Assembler (makine dili derleyicisi) düzeyinde çok hızlı yapılabildiğinden, Bresenham algoritması oldukça hızlı bir algoritmadır. Şekil 4.2'de algoritmanın kod dizimi görülmektedir.

```

function line(x0, x1, y0, y1)

    int deltax := abs(x1 - x0)

    int deltay := abs(y1 - y0)

    real error := 0

    real deltaerr := deltay / deltax

    int y := y0

    for x from x0 to x1

        plot(x,y)

        error := error + deltaerr

        if error ≥ 0.5 then

            y := y + 1

            error := error - 1.0

```

Şekil 4.2 Bresenham çizgi algoritmasının kod dizimi

Bu yöntemin çizgisel görülebilirlik algoritmasına uyarlanmasında piksel değeri yerine koordinat bilgisi kullanılmaktadır. Algoritmanın koordinatlar üzerinden ilerlemesi ile o koordinat üzerinde yükseklik kontrolü yapılmakta ve böylelikle iki referans noktası arasında görülebilirlik hesabı yapılmaktadır.

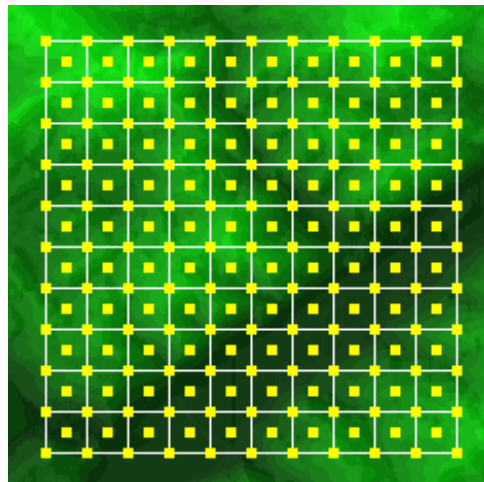
4.4. Sensör Modeli

Sensörler, genel olarak belirli bir algılama kapasitesi olan ve belirli bir menzilde dâhilinde bilgi toplayabilen cihazlardır. Problem çözümünde kullanılacak sensör modeli problemin gereksinimlerine göre farklılık gösterebilir (Termal, Elektromanyetik, Mekanik, Kimyasal, Akustik, Optik vb.). Bu çalışmada belirli bir menzile sahip, menzili içerisinde bulunan cisimleri çizgisel görülebilirlik analizi ile görebilen genel bir sensör modeli kullanılmıştır. Çizgisel görülebilirlik, bir yüzey üzerinde seçilen iki referans noktadan birinde bulunan bir gözlemci tarafından, bu iki nokta arasında kalan kısımların ne kadarının görülebileceğini bulmaya yarayan hesaplamalardır. Bir sensörün menzilinde bulunan herhangi bir nokta çizgisel görülebilirlik içerisinde ise bu noktanın sensör tarafından görüldüğü kabul edilir. Yani sensör menzili içerisinde bir cismin görülebilme ihtimali tüm uzaklıklar için

aynıdır. Sensörlerin görülebilirlik hesapları Bresenham Çizgisel Görülebilirlik algoritması ile yapılmıştır. Bir sensörün 360 derecelik açı ile etrafını görebildiği kabul edilmiştir. Ayrıca sensörlerin yerden belirli bir yükseklikte olduğu kabul edilmiştir.

4.5. Görülebilirlik Hesabı

Bu çalışmada probleme ait arazi, köşe koordinatları tanımlanmış olan dikdörtgen şeklinde bir arazidir. Tüm sensörlerin arazi üzerinde görülebilirliğinin hesaplanması iki aşamadan oluşur. Bu aşamanın ilkinde, tanımlanmış dikdörtgen araziye karşılık gelen bir görülebilirlik matrisi tanımlanır. Tanımlanan her bir sensörün bu alan içerisindeki diğer tüm noktaları görüp görmediği hesaplanır. Bu hesaplama sonucunda arazinin görülebilirlik matrisi elde edilmiş olur. İkinci aşamada ise arazinin hangi oranda görülebildiği hesaplanır. Ancak görülebilirlik matrisi üzerindeki her bir nokta için bulunan değerlerin teker teker toplanması uzun süreceği için arazi üzerinde belirli noktalar seçilerek örnekleme yapılır. Bu örnekleme işleminde arazi şekil 4.2’de görüldüğü gibi ızgara şeklinde ayrılmış hücrelerden oluşmaktadır. Görülebilirlik hesabında her bir hücrenin köşe noktaları ve merkezi olmak üzere 5 noktaya bakılır ve toplam görülen nokta oranında hücrenin görülebilirliği çıkartılır. Toplam alanın görülebilirliği her bir hücrenin görülebilirlik değerinin toplamıyla hesaplanır.



Şekil 4.3 Görülebilirlik hesabının yapılacağı alanın ızgara haline getirilmesi ve belirlenen noktalar

4.6. Problemin Matematiksel Modeli

Daha önce tanımlandığı gibi problemin çözümünden hedeflenen belirli sayıda sensörle belirli alanın azami derecede kapsanması sensör yerleştirme probleminin türevlerinden birisidir. Bu çalışmada problemin bu türevi üzerinde çalışmalar yapılmış ve verilen n adet sensörün hedef alan üzerinde maksimum kapsama hedefiyle en uygun noktalara konuşlandırılması hedeflenmiştir. Dolayısıyla etkinlik ifadesi alanın görülebilirlik değeri şeklinde tanımlanmalıdır. Daha önce ifade edildiği gibi bir alanın görülebilirlik değeri alan içerisindeki hücrelerin görülebilirlik değerinin toplamı şeklinde ifade edilmiştir. Bu şekilde kapsamanın tanımı yapılırsa i 'inci çözünürlük hücresindeki görülen nokta sayısı N_i , bir hücrenin alanı A_i ve optimizasyon yapılacak bölgenin toplam alanı T olarak tanımlanmış olsun. Böylece problemin çözümü için gereken amaç fonksiyonu aşağıdaki şekilde formüle edilmiştir:

$$f(x) = \frac{\left(\sum_i 0.2 N_i A_i \right)}{T} \times 100 \quad (4.1)$$

Bu formüle göre $f(x)$ değeri tanımlı alanın yüzde olarak hangi oranda görüldüğünü gösterir. 0.2 değeri ile ifade edilmek istenen bir hücre içindeki 5 noktadan her birinin görülebilirliğe etkisidir. $0.2N_i$ sonuç olarak $[0,1]$ aralığında bir değer almış olur. Bu çözüm modeli ile hedeflenen, probleme girdi olarak verilen alan ve sensör bilgileri ile sensörleri en uygun pozisyonlara atayarak, $f(x)$ değerinin alabileceği maksimum değerin bulunmasıdır.

5. PROBLEMİN META-SEZGİSEL YÖNTEMLERE UYARLANMASI

Tanımlanan sensör yerleştirme probleminin meta-sezgisel yöntemlerle çözümü için kullanılacak meta-sezgisel yöntemlere uyarlanacak şekilde modellenmesi gerekmektedir. Bu bölümde önce problemin Genetik Algoritmalar ile çözümünün modellenmesi ve ardından PSO yöntemine uyarlanması anlatılmaktadır. Ayrıca VNS yöntemi PSO ve GA çözümlerinin bir ara basamağı olarak yerel arama yapmak üzere kullanılmıştır. VNS yöntemlerinden RVNS yöntemi probleme uyarlanmış ve RVNS yöntemi Genetik Algoritma ve PSO yöntemlerinin en iyi çözüm değişimlerinde uygulanmıştır.

Tüm yöntemlerde enlem ve boylam bilgileri $[0,1]$ aralığında reel sayılar olarak tanımlanmıştır. Böylece çözüm koordinatlara bağlı olmaksızın yapılabilmektedir. Gerçek koordinatlara dönüştürme işlemleri uygunluk fonksiyonu hesaplama esnasında yapılmaktadır. Genel olarak tüm yöntemlerde olası bir çözüm probleme girdi teşkil eden sensörlerin pozisyon bilgilerinde oluşan bir listedir. Uygunluk hesabı yapmak için bu sensör pozisyon listesini ve alan bilgisini oluşturmak yeterlidir. Uygunluk değeri bir alanın oran olarak yüzde kaçının görüldüğünü ifade eder.

5.1. Genetik Algoritma Uyarlaması

Bir problemin Genetik Algoritmaya uyarlanması için çözüm kümesinin gösterimi (gen), çaprazlama ve mutasyon operatörlerinin tanımı, seçim stratejisinin nasıl olduğu ve algoritmanın parametrelerinin belirlenmesi gerekmektedir.

5.2. Kromozom Yapısı

Problem daha önce anlatıldığı gibi sürekli bir fonksiyonun optimizasyonu olduğu için gen yapısı reel değerli kromozom olarak tanımlanmıştır. Bir sensör için enlem bilgisi s_e ve boylam bilgisi s_b olarak ifade edilirse n adet sensörden oluşan bir problemde sensörlerin pozisyon bilgisi $[s_{e_1}, s_{b_1}, s_{e_2}, s_{b_2}, \dots, s_{e_n}, s_{b_n}]$ şeklinde ifade edilir.

$$[0.12, 0.8, 0.43, 0.9, 0.11, 0.28]$$

Şekil 5.1: Örnek kromozom yapısı.

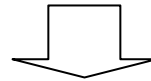
Şekil 5.1’de örnek bir gen yapısı görülmektedir. Bu gen yapısındaki çözümde 3 adet sensörün pozisyonu gösterilmektedir. Birinci sensör için enlem ve boylam 0.12, 0.8, ikinci sensör için 0.43, 0.9, üçüncü sensör için enlem ve boylam 0.11 ve 0.28 şeklinde ifade edilmiştir.

5.3. Çaprazlama Operatörü

Çaprazlama operatörü olarak tekil aritmetik çaprazlama kullanılmıştır. Bu yöntemde göre seçilen 2 adet ebeveyn ortak bilgi paylaşımıyla yeni iki adet çocuk oluşturmaktadırlar. Bilindiği gibi gen yapısı sensörlerin enlem ve boylam bilgilerinden oluşan bir dizisidir. Bu listede her bir gen çifti sırasıyla bir sensörün enlem ve boylam bilgisini içermektedir. Bu operatör ile rasgele seçilen bir sensörün enlem ve boylam bilgisi güncellenmektedir. Koordinat bilgisinin güncellenmesi için $[0,1]$ aralığında rasgele seçilen bir α değeri kullanılmaktadır. Üretilen α değerine göre i ’inci indeksteki bir sensörün seçilerek iki bireyin çaprazlanması sonucunda ortaya çıkan yeni bireyler şekil 5.2’de gösterilmiştir.

$$\text{Ebeveyn 1: } [x_{e_1}, x_{b_1}, x_{e_2}, x_{b_2}, \dots, x_{e_i}, x_{b_i}, x_{e_n}, x_{b_n}]$$

$$\text{Ebeveyn 2: } [y_{e_1}, y_{b_1}, y_{e_2}, y_{b_2}, \dots, y_{e_i}, y_{b_i}, y_{e_n}, y_{b_n}]$$



$$\text{Çocuk 1 : } [x_{e_1}, x_{b_1}, x_{e_2}, x_{b_2}, \dots, \alpha x_{e_i} + (1-\alpha)y_{e_i}, \alpha x_{b_i} + (1-\alpha)y_{b_i}, x_{e_n}, x_{b_n}]$$

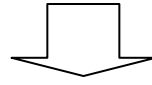
$$\text{Çocuk 2 : } [y_{e_1}, y_{b_1}, y_{e_2}, y_{b_2}, \dots, (1-\alpha)x_{e_i} + \alpha y_{e_i}, (1-\alpha)x_{b_i} + \alpha y_{b_i}, y_{e_n}, y_{b_n}]$$

Şekil 5.2 Çaprazlama operatörü

Şekil 5.3'te gösterilen örnekte $\alpha = 0.2$ ve $i = 1$ çaprazlamaya giren bireylerin sonucunda ortaya çıkan yeni bireyler görülmektedir. Bu operatöre göre yeni oluşan koordinat bilgisi seçilen iki sensörün arasında bir konumda olacak şekilde hesaplanmaktadır. α parametresi konumun ayarlanmasında belirleyici rol oynamaktadır.

Ebeveyn 1: [0.1,0.8,0.43,0.9,0.11,0.28]

Ebeveyn 2: [0.3,0.1,0.6,0.54,0.43,0.96]



Çocuk 1: [0.14,0.66,0.43,0.9,0.11,0.28]

Çocuk 2: [0.26,0.24,0.6,0.54,0.43,0.96]

Şekil 5.3 $\alpha = 0.2$ ve $i = 1$ için örnek çaprazlama

5.4. Mutasyon Operatörü

Mutasyon işlemi yapılacak olan bireyde rasgele seçilen i 'inci indeksteki bir sensörün enlem veya boylam bilgisi 5.1 numaralı formüle göre değiştirilmektedir. Bu formülde, $N(0,1)$ değeri standart normal dağılımlı bir sayıyı ifade etmektedir (0=ortalama, 1=varyans). Bir kromozom üzerindeki genlerde ne kadar mutasyon uygulanacağı mutasyon oranı ile belirlenmektedir.

$$p'_i = p_i + N(0,1) \quad 5.1$$

Seçilen Birey: [0.1,0.8,0.43,0.9,0.11,0.28]

Mutasyona Uğramış Hali: [0.1,0.803,0.43,0.891,0.11,0.28]

Şekil 5.4 Rasgele seçilmiş 2. ve 4. indekslerde bulunan genlerin mutasyonu sonucunda oluşan yeni birey

5.5. Seçim Stratejisi

Çaprazlama için seçilecek olan bireyler turnuva seçimi ile belirlenmiştir. Bu yaklaşıma göre popülasyonda bulunan bireylerden rasgele seçilen 2 bireyden daha iyi uygunluk değerine sahip olanı seçilmekte ve çaprazlama işlemine tabi tutulmaktadır.

Yeni popülasyondaki bireylerin oluşması için çaprazlama oranı ve elitist yaklaşım önemli rol oynamaktadır. Elitist yaklaşıma göre en iyi birey doğrudan bir sonraki nesle aktarılmaktadır. Çaprazlama oranının 1 değerinden küçük olması durumunda bazı bireyler çaprazlama işlemine tutulmadan doğrudan bir sonraki nesle aktarılma şansına sahiptir. Çaprazlama işlemine tabi tutulan bireylerden oluşan yeni bireyler doğrudan bir sonraki nesle aktarılmaktadır.

5.6. PSO Uyarlaması

PSO yöntemi doğası gereği reel sayılarla çalıştığı için bu probleme uyarlaması oldukça basittir. Bir parçacığın pozisyon ve hız bilgisinin tanımlanması ve parametrelerin tayin edilmesiyle problem yönetime uyarlanmış olacaktır.

5.7. Parçacık Yapısı

PSO yönteminde her bir parçacık sensör listesini pozisyon bilgisi olarak tutar. Bir sensör için enlem bilgisi s_e ve boylam bilgisi s_b olarak ifade edilirse n adet sensörden oluşan bir problemin PSO pozisyon bilgisi $[s_{e_1}, s_{b_1}, s_{e_2}, s_{b_2}, \dots, s_{e_n}, s_{b_n}]$ şeklinde ifade edilir. Algoritmanın her bir iterasyonu sonucunda sensörlerin koordinat bilgileri güncellenmiş olur. Bir parçacığın hız bilgisi başlangıç olarak $[0,1]$ aralığında rasgele sayılar olarak tanımlanmıştır. Parçacıkların pozisyon bilgisiyle genetik algoritma uyarlamasındaki kromozom yapısı bu haliyle benzerlik göstermektedir. Bir parçacığın hız bilgisi ise $[v_{e_1}, v_{b_1}, v_{e_2}, v_{b_2}, \dots, v_{e_n}, v_{b_n}]$ şeklinde ifade edilmektedir. Algoritma her iterasyonda pozisyon ve hız bilgisini 5.2 ve 5.3 numaralı denklemleri kullanarak güncellemektedir.

$$v_{t+1} = wv_t + c_1r_1(p_{best} - p_t) + c_2r_2(g_{best} - p_t) \quad (5.2)$$

$$p_{t+1} = p_t + v_{t+1} \quad (5.3)$$

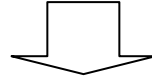
Pozisyon : [0.1,0.2,0.3,0.9,0.8,0.5]

Hız: [0.12,0.14,0.2,0.1,0.3,0.2]

pBest: [0.2,0.7,0.7,0.7,0.4,0.1]

gBest: [0.6,0.5,0.14,0.2,0.4,0.6]

c_1 :2, c_2 : 2, r_1 : 0.4, r_2 : 0.5, w : 0.3



Yeni Hız: [0.616,0.742,0.22,-0.83,-0.63,-0.16]

Yeni Pozisyon: [0.716,0.942,0.52,0.07,0.17,0.34]

Şekil 5.5 Örnek bir PSO iterasyonu parçacık hız ve pozisyon hesaplaması

5.8. Parametrelerin Seçimi

Problemin gösterim biçiminin tanımlanması sonucunda PSO algoritmasını uygulama için başlangıç popülasyonunun belirlenmesi ve algoritmanın parametrelerinin atanması gerekmektedir. Algoritma başlangıç popülasyonu olarak her bir parçacık için sensörlere alan içerisinde rasgele koordinatlar atanır. Her bir iterasyonda sensörlerin koordinat değerleri (5.2) ve (5.3) numaralı denklemler ile güncellenir ve böylece çözüm uzayında yeni yerler denenerek global optimum değeri araştırılmış olur. PSO algoritmasında atalet sabitinin zamanla düşürülmesi ile de daha iyi sonuçlar verdiği gözlemlenmiştir. w değerinin güncellenmesi aşağıdaki formüle göre yapılmaktadır.

$$w_{t+1} = w_t \times 0.95 \quad (5.4)$$

Başlangıç değeri olarak w değeri 1 olarak seçilir. c_1 ve c_2 değerleri 2 olarak seçilmiştir. Durdurma kriteri olarak belirli sayıda etkinlik fonksiyonu çağırılması kullanılmıştır.

PSO yönteminin bir başka versiyonunda yerel arama yöntemi kullanarak melez bir yöntem geliştirilmiştir. Yerel aramanın yapılıp yapılmayacağı bir parametre ile kontrol edilmektedir. Yerel arama sadece gBest değerinin değişiminde uygulanmaktadır. Ancak yerel aramaya tabi tutulan parçacığın pozisyon değerleri bir değişikliğe uğramadan popülasyonda varlığını sürdürür. Yerel arama yöntemi olarak RVNS yöntemi kullanılmıştır. Yerel aramaya tabi tutulan bir parçacığın hız vektörü rasgele güncellenir.

5.9. RVNS Uyarlaması

VNS yönteminin bir türevi olan RVNS yöntemi hızlı olması sebebiyle kullanılmıştır. Bir problemi RVNS yöntemine uyarlamak için çözüm kümesi gösterimi, komşuluk fonksiyonlarının tanımlanması ve parametrelerin atanması gerekmektedir.

5.10. RVNS Çözüm Kümesi Gösterimi

Bölüm 5.1.1 anlatılan gen yapısı ile RVNS çözüm kümesi aynı yapıdadır. N elemandan oluşan bir sensör listesi $[s_{e_1}, s_{b_1}, s_{e_2}, s_{b_2}, \dots, s_{e_i}, s_{b_i}, s_{e_n}, s_{b_n}]$ şeklinde ifade edilir.

5.11. Komşuluk Fonksiyonları

Komşuluk fonksiyonları yakın arama ve uzak arama komşuluk fonksiyonları olmak üzere iki çeşittir. RVNS algoritmasında pozisyon bilgisi yakın arama ile koordinatlar üzerinde çok küçük değişiklikler yapılarak değiştirilmiştir. Bir çözümde

rasgele seçilen bir sensörün koordinat değerleri aşağıdaki formüle göre hesaplanmaktadır:

$$p_{t+1} = p_t + k \quad (5.5)$$

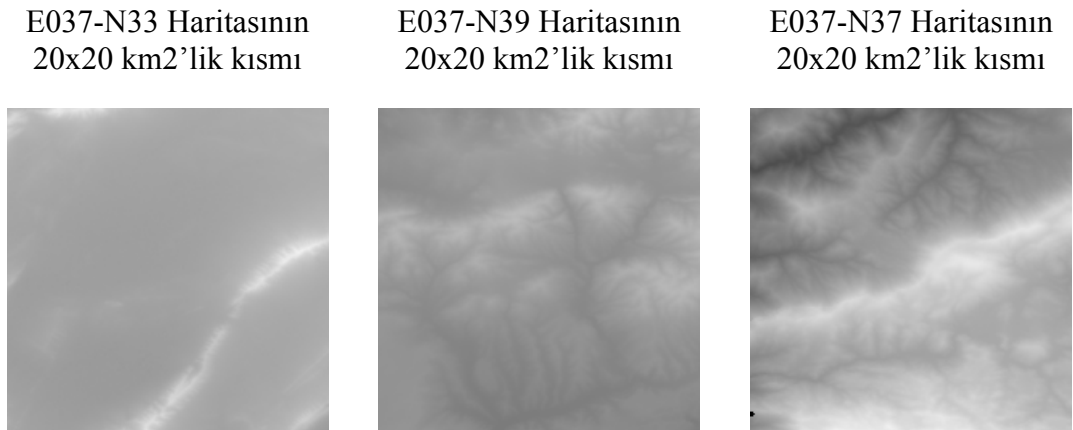
Yakın arama komşuluğu için k değeri [-0.00001, 0.00001] arasında düzgün dağılımlı bir sayıdır. Uzak arama komşuluk fonksiyonunda k aralığı biraz daha genişletilmiş ve [-0.0001, 0.0001] olarak tanımlanmıştır.

5.12. Parametrelerin Seçimi

RVNS yönteminin durdurma kriteri algoritmanın belirli sayıda iterasyon gerçekleştirmesidir. Yapılan testlerin sonuçları doğrultusunda Genetik algoritma ve Parçacık Sürü Optimizasyonu için ayrı ayrı seçilmiştir. Yakın aramadan uzak arama komşuluğuna geçiş için yakın aramada en az 4 kabul almayan çözüm denemesi gerekmektedir.

6. DENEYSEL ÇALIŞMALAR

Tüm deneysel çalışmalar AMD Turion 1.8 GHz işlemci ve 1.5 GB RAM'e sahip bir bilgisayarda yapılmıştır. Yazılımlar Sun Java JDK 1.5 dili kullanılarak kodlanmıştır. Örnek problemler için kullanılan haritalar DTED seviye-1 formatında bir haritalardır. Köşe noktaları E037-N33, E037-N39 ve E037-N37 olmak üzere farklı alanlar tanımlanarak kullanılmıştır. Bu araziler birbirine nispeten sırasıyla düz, engebeli ve çok engebeli olmak üzere farklı özelliklere sahiptir. Böylelikle algoritmaların farklı engebelik seviyesine sahip alanlardaki davranışları test edilmiştir. Deneylerde kullanılan alan ve sensör bilgileri tablo 6.1'de gösterildiği gibidir. Tabloda sensör sayısı ve kullanılan sensörlerin menzil bilgileri de bulunmaktadır. Ayrıca her bir sensörün yerden 3 metre yüksekte olduğu kabul edilmektedir. Alanlar kare şeklinde olup bir kenar uzunluğu tabloda belirtilmektedir. Her bir deney E037-N33, E037-N39 ve E037-N37 haritaları için ayrı ayrı gerçekleştirilmiştir. Deneyler hakkındaki bir diğer husus Deney1-Deney4 deneyleriyle Deney5-Deney8 arasındaki farkın bir sensörün eksik olmasıdır. Deney1-Deney4 deneyleri ise Deney9-Deney12 de bulunan sensörlerden bir eksik sensöre sahiptir. Aynı alan üzerine farklı sensör sayılarının kullanılmasını sebebi, sensör sayısının değişmesi sonucunda etkinliğin azalması ya da artmasının gözlemlenmesidir.



Şekil 6.1 Deneylerde kullanılan alanlardan örnekler

Tüm deneyler için testler 10'ar defa çalıştırılmış ve etkinliklerinin ortalamaları verilmiştir. Ayrıca bu verilere ait standart sapma (STD) değerleri de test sonuç

tablolarında yer almaktadır. Tüm yöntemler için algoritma durdurma kriteri etkinlik değerlerinin 7000 defa çağırılması olarak belirlenmiştir. Başlangıç popülasyonları belirlenirken, popülasyondaki çözümlerdeki sensör pozisyonları $[0,1]$ aralığında rasgele sayılar olarak tanımlanmıştır. PSO yönteminde başlangıç hızları aynı şekilde $[0,1]$ aralığında rasgele sayılardır.

Tablo 6.1 Tüm belirlenmiş alanlar üzerinde denenen örnek problemler

Deney Adı	Kenar Uzunluğu (km)	Alan (km ²)	Sensör Sayısı	Menzil (m)
Deney1	5	25	2	3
Deney2	10	100	4	4
Deney3	20	400	4	8
Deney4	30	900	8	9
Deney5	5	25	1	3
Deney6	10	100	3	4
Deney7	20	400	3	8
Deney8	30	900	7	9
Deney9	5	25	3	3
Deney10	10	100	5	4
Deney11	20	400	6	8
Deney12	30	900	9	9

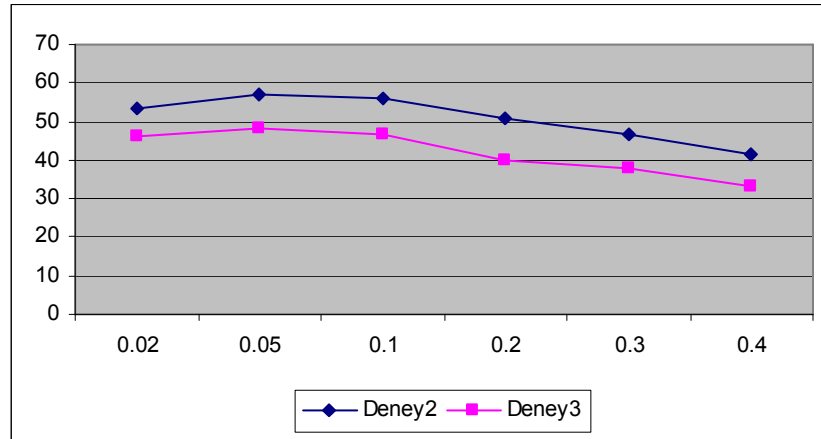
6.1. Parametrelerin Belirlenmesi

Algoritmaların parametrelerinin belirlenmesi için çeşitli testler yapılmış ve sonuçları analiz edilmiştir. Genetik algoritmanın mutasyon oranı, çaprazlama oranı ve popülasyon büyüklüğü parametreleri ile ilgili testler gerçekleştirilmiştir. Mutasyon oranı için yapılan testler tablo 6.2’de belirtilen oranlar için gerçekleştirilmiştir. Tablo 6.2’de gösterilen sonuçlar çaprazlama oranı 1 ve popülasyon büyüklüğü 30 olduğu durumlarda gerçekleştirilmiştir. Harita olarak E037-N37 haritası kullanılmıştır. Bu tabloda Standart Sapma (STD) ve etkinlik

değerleri gösterilmektedir. Sonuçlar analiz edildiğinde Test 2 ve Test 8 durumlarında en iyi etkinlik elde edilmiştir. Bu sebepten dolayı genetik algoritma için mutasyon oranı 0.05 olarak seçilmiştir.

Tablo 6.2 Genetik Algoritma mutasyon oranı test sonuçları

Deney No	Test No	Mutasyon Oranı	Etkinlik (%)	STD
Deney2	1	0.02	53,37	3,83
Deney2	2	0.05	57,19	2,50
Deney2	3	0.1	55,86	1,77
Deney2	4	0.2	50,94	2,93
Deney2	5	0.3	46,76	4,28
Deney2	6	0.4	41,57	4,94
Deney3	7	0.02	46,22	1,73
Deney3	8	0.05	48,42	3,14
Deney3	9	0.1	46,77	2,79
Deney3	10	0.2	39,84	3,49
Deney3	11	0.3	37,73	2,74
Deney3	12	0.4	33,31	2,02

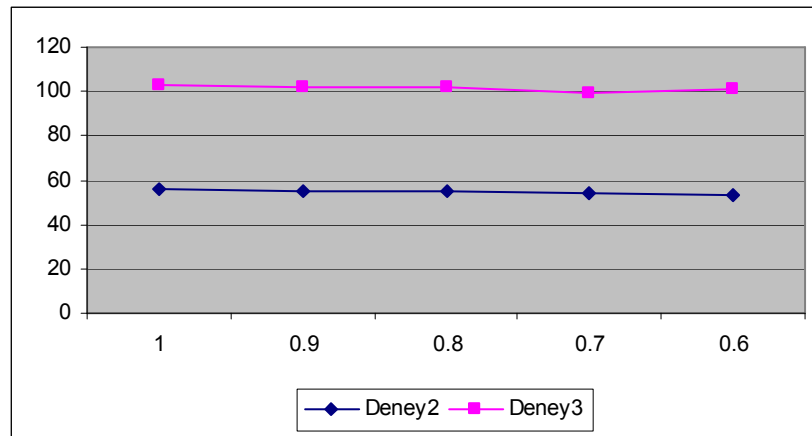


Şekil 6.2 Genetik Algoritma mutasyon oranı test sonuçlarının grafiksel gösterimi

Çaprazlama oranı için yapılan testler tablo 6.3'te gösterilmektedir. Çaprazlama parametresinin testlerinde mutasyon oranı 0.1 ve popülasyon büyüklüğü 30 olarak seçilmiştir. Harita olarak E037-N37 haritası kullanılmıştır. Tablo 6.3 incelendiğinde en etkin değerler Test 1 ve Test 6 ile elde edilmiştir. Bu durumda çaprazlama oranı 1 olarak seçilmiştir.

Tablo 6.3 Genetik Algoritma çaprazlama oranı test sonuçları

Deney No	Test No	Çaprazlama Oranı	Etkinlik	STD
Deney2	1	1	55,60	3,17
Deney2	2	0.9	55,04	2,33
Deney2	3	0.8	55,33	3,36
Deney2	4	0.7	54,17	3,63
Deney2	5	0.6	53,14	1,99
Deney3	6	1	47,30	2,95
Deney3	7	0.9	46,86	2,63
Deney3	8	0.8	46,42	2,65
Deney3	9	0.7	45,00	1,99
Deney3	10	0.6	47,60	2,38

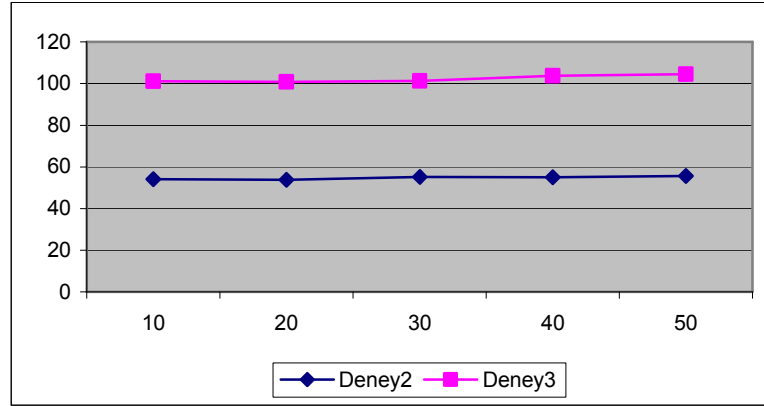


Şekil 6.3 Genetik Algoritma çaprazlama oranı test oranının grafiksel gösterimi

Popülasyon büyüklüğünün tespit edilmesi için kullanılan testler tablo 6.4'te gösterilmektedir. Bu tabloda gösterilen çalışmalarda mutasyon oranı 0.05, çaprazlama oranı 1 olarak seçilmiştir. Harita olarak E037-N37 haritası kullanılmıştır. Sonuçlar incelendiğinde Test 5 ve Test 10'da en etkin sonuçlar elde edildiği görülmektedir. Dolayısıyla Genetik Algoritma popülasyon büyüklüğü 50 olarak belirlenmiştir.

Tablo 6.4 Genetik Algoritma popülasyon büyüklüğü test sonuçları

Deney No	Test No	Popülasyon Büyüküğü	Etkinlik	STD
Deney2	1	10	54,11	3,26
Deney2	2	20	53,93	2,67
Deney2	3	30	55,21	2,53
Deney2	4	40	55,06	1,90
Deney2	5	50	55,74	2,63
Deney3	6	10	47,05	2,62
Deney3	7	20	46,92	3,03
Deney3	8	30	46,12	2,99
Deney3	9	40	48,74	3,14
Deney3	10	50	48,77	2,27

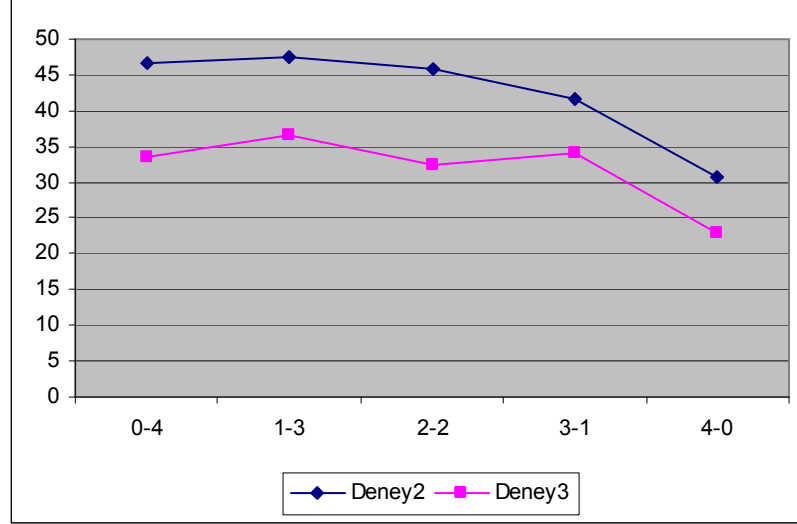


Şekil 6.4 Genetik Algoritma popülasyon büyüklüğü testi gösterimi

Parçacık Sürü Optimizasyonu yöntemi için c_1 , c_2 ve popülasyon büyüklüğü parametrelerinin belirlenmesi gerekmektedir. Parametrelerin belirlenmesinde kullanılan harita E037-N37 olarak belirlenmiştir. Tablo 6.5'te c_1 ve c_2 değerlerinin belirlenmesi için yapılan testler ve sonuçları bulunmaktadır. c_1 ve c_2 değerlerinin belirlenmesinde popülasyon değeri 30 seçilmiştir. Tablo 6.5'de görüldüğü gibi Test 2 ve Test 7'de en etkin değerler bulunmuş ve dolayısıyla parametre olarak c_1 parametresinin değeri 1, c_2 parametresinin değeri 3 olarak seçilmiştir.

Tablo 6.5 Parçacık Sürü Optimizasyonu c_1 ve c_2 parametre test sonuçları

Deney Adı	Test No	c_1	c_2	Etkinlik	STD
Deney2	1	0	4	46,59	5,91
Deney2	2	1	3	47,36	4,63
Deney2	3	2	2	45,69	3,84
Deney2	4	3	1	41,50	2,63
Deney2	5	4	0	30,82	4,62
Deney3	6	0	4	33,53	5,76
Deney3	7	1	3	36,60	5,41
Deney3	8	2	2	32,29	4,47
Deney3	9	3	1	34,03	4,90
Deney3	10	4	0	22,91	3,69

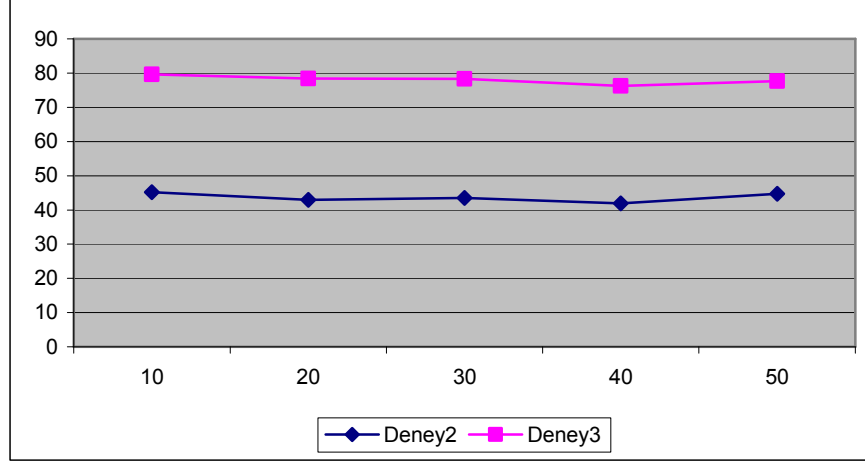


Şekil 6.5 PSO c_1 - c_2 parametrelerinin test sonuçlarının grafiksel gösterimi

Parçacık Sürü Optimizasyonu yöntemi için seçilecek parametrelerden biri de popülasyon büyüklüğüdür. Tablo 6.6’da bu parametrenin seçimi ile ilgili testler ve sonuçları bulunmaktadır. Bu sonuçlara göre popülasyon büyüklüğü 20 olarak seçilmiştir.

Tablo 6.6 Parçacık Sürü Optimizasyonu popülasyon büyüklüğü test sonuçları

Deney Adı	Test No	Popülasyon Büyüklüğü	Etkinlik	STD
Deney2	1	10	45,24	4,96
Deney2	2	20	43,00	3,93
Deney2	3	30	43,56	4,08
Deney2	4	40	41,98	4,69
Deney2	5	50	44,81	3,65
Deney3	6	10	34,41	7,00
Deney3	7	20	35,38	3,76
Deney3	8	30	34,73	4,23
Deney3	9	40	34,26	3,05
Deney3	10	50	32,81	4,66

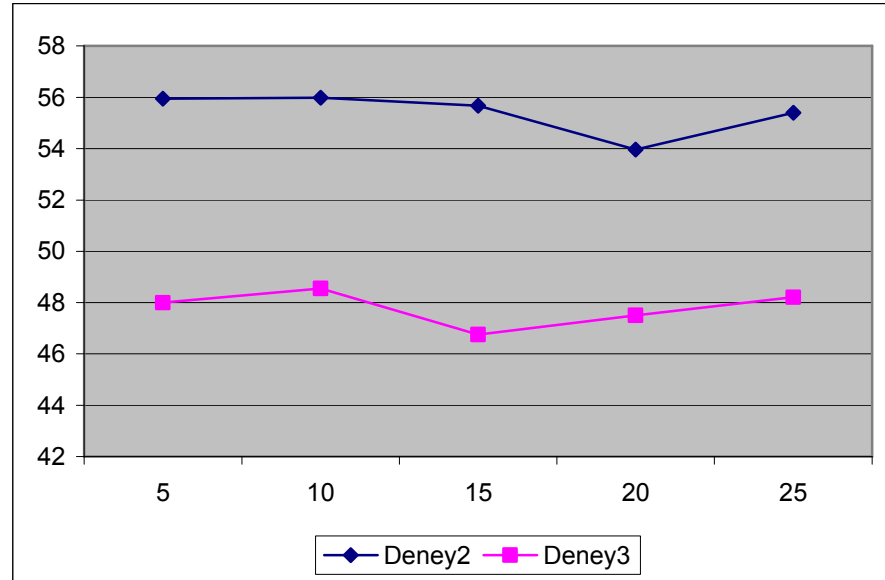


Şekil 6.6 PSO popülasyon büyüklüğü test sonuçlarının grafiksel gösterimi

Genetik Algoritma ve Parçacık Sürü Optimizasyonu yöntemlerinin RVNS yerel arama yöntemi ile değiştirilmesi sonucu iki farklı algoritma ortaya çıkmaktadır. Yerel Aramalı Genetik Algoritma (YGA) ve Yerel Aramalı Parçacık Sürü Optimizasyonu (YPSO) olarak adlandırılan bu yöntemlerde yerel arama esnasında ne kadar iterasyon yapılacağı ayrı bir parametre olarak ortaya çıkmaktadır. YGA çözümü için iterasyon miktarı için yapılan testlerin sonuçları tablo 6.7’de verilmiştir. YGA için seçilen çaprazlama oranı 1, mutasyon oranı 0.05 ve popülasyon büyüklüğü 50 olarak seçilmiştir. Test 2 ve Test 7’de görüldüğü gibi en etkin sonuçlar iterasyon miktarının 10 olarak seçilmesi gerekir.

Tablo 6.7 Yerel Aramalı Genetik Algoritma için yerel arama iterasyon sayısı test sonuçları

Deney Adı	Test No	İterasyon	Etkinlik	STD
Deney2	1	5	55,95	2,69
Deney2	2	10	55,98	2,84
Deney2	3	15	55,67	2,30
Deney2	4	20	53,96	1,92
Deney2	5	25	55,40	2,88
Deney3	6	5	48,00	1,78
Deney3	7	10	48,55	2,32
Deney3	8	15	46,75	2,44
Deney3	9	20	47,49	2,95
Deney3	10	25	48,21	2,32



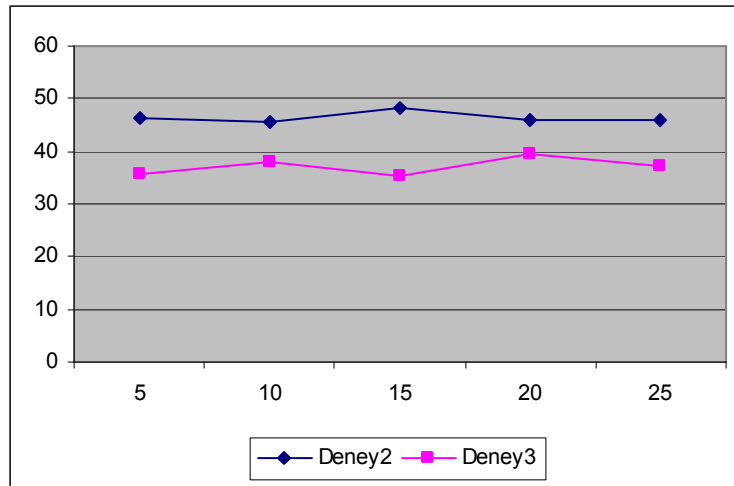
Şekil 6.7 Yerel Aramalı Genetik Algoritma için yerel arama iterasyon sayısı test sonuçlarının grafiksel gösterimi

Yerel Aramalı Parçacık Sürü Optimizasyonu (YPSO) için de yerel arama algoritması iterasyon sayısının belirlenmesi gerekmektedir. Tablo 6.8’de YPSO için

yapılan testler görülmektedir. YPSO için c_1 değeri 1, c_2 değeri 3 ve popülasyon büyüklüğü 20 olarak seçilmiştir. Yapılan testler incelendiğinde iterasyon miktarı 15 seçilmiştir.

Tablo 6.8 Yerel Aramalı Parçacık Sürü Optimizasyonu yerel arama iterasyon test sonuçları

Deney Adı	Test No	İterasyon	Etkinlik	STD
Deney2	1	5	46,41	4,82
Deney2	2	10	45,47	5,54
Deney2	3	15	48,40	4,66
Deney2	4	20	45,91	6,21
Deney2	5	25	45,97	6,79
Deney3	6	5	35,69	6,36
Deney3	7	10	37,96	4,05
Deney3	8	15	35,32	4,49
Deney3	9	20	39,64	4,36
Deney3	10	25	37,25	5,68



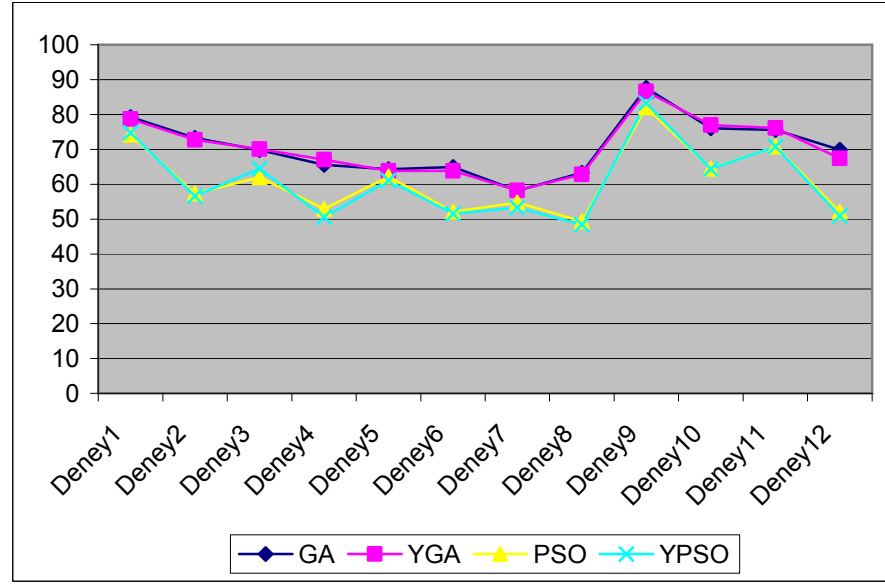
Şekil 6.8 Yerel Aramalı Parçacık Sürü Optimizasyonu yerel arama iterasyon test sonuçlarının grafiksel gösterimi

6.2. Karşılaştırmalı Testler

GA, YGA, PSO, YPSO yöntemleri için parametre değerlerinin atanması sonucunda karşılaştırmalı testler yapılmıştır. Tablo 6.9’da E037-N33 haritası üzerinde yapılan deneyler ve sonuçları gösterilmektedir. Bu tablodaki değerlere göre en iyi sonuçlar Genetik Algoritma (GA) yöntemi ve Yerel Aramalı Genetik Algoritma (GA) yöntemi ile bulunduğu görülmektedir. GA ve YGA yöntemlerinde çıkan değerler tüm test durumları için yakın değerde çıkmışlardır. Ayrıca yerel arama işleminin genel olarak PSO yöntemini iyileştirmediyi görmekteyiz. PSO ve YPSO etkinlik değerlerinin de genel olarak yakın çıkmış olduğunu görmekteyiz.

Tablo 6.9 E037-N33 haritası üzerinde yapılan deneyler ve sonuçları

Deney Adı	Test No	GA	YGA	PSO	YPSO
Deney1	1	79,28	78,69	74,20	74,77
Deney2	2	73,26	72,76	57,33	56,59
Deney3	3	69,86	70,12	62,01	64,45
Deney4	4	65,60	67,08	52,95	50,52
Deney5	5	64,35	63,88	62,17	61,09
Deney6	6	64,99	63,86	52,13	51,53
Deney7	7	58,18	58,23	54,73	53,30
Deney8	8	63,28	62,83	49,34	48,46
Deney9	9	87,63	86,62	82,00	83,28
Deney10	10	76,07	76,95	64,47	64,36
Deney11	11	75,61	76,12	70,82	70,79
Deney12	12	69,89	67,41	52,12	50,94

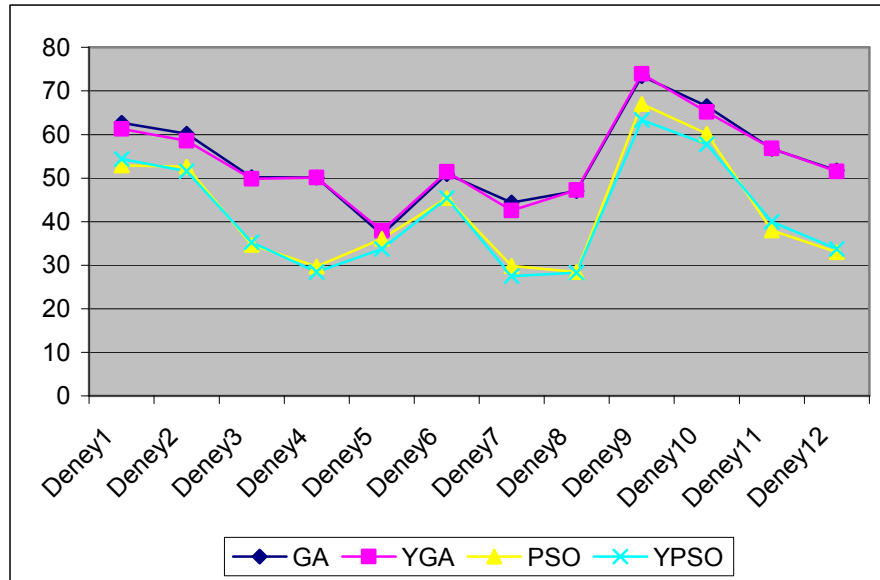


Şekil 6.9 E037-N33 haritası üzerinde yapılan deneylerin sonuçlarının grafiksel gösterimi

Tablo 6.10'da E037-N39 haritası üzerinde yapılan deneyler ve sonuçları gösterilmektedir. . Bu tablodaki değerlere göre en iyi sonuçlar Genetik Algoritma (GA) yöntemi ve Yerel Aramalı Genetik Algoritma (YGA) yöntemi ile bulunduğu görülmektedir. GA ve YGA yöntemlerinde çıkan değerler tüm test durumları için yakın değerlerde çıkmışlardır. Ayrıca yerel arama işleminin genel olarak PSO yöntemini iyileştirmediğini görmekteyiz. PSO ve YPSO etkinlik değerlerinin de genel olarak yakın çıkmış olduğunu görmekteyiz.

Tablo 6.10 E037-N39 haritası üzerinde yapılan deneyler ve sonuçları

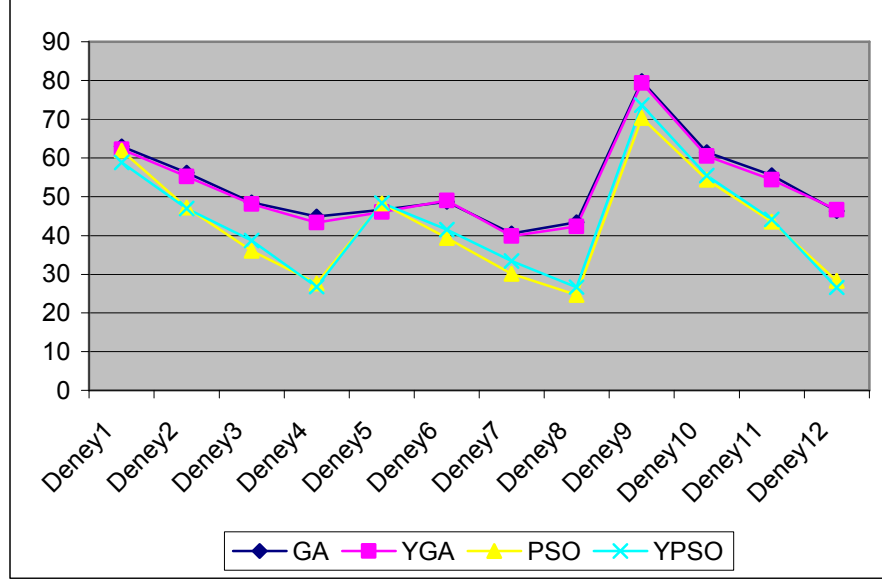
Deney Adı	Test No	GA	YGA	PSO	YPSO
Deney1	1	62,67	61,31	52,89	54,35
Deney2	2	60,21	58,60	52,60	51,60
Deney3	3	50,22	49,85	34,68	35,28
Deney4	4	50,10	50,12	29,76	28,48
Deney5	5	37,04	37,95	36,12	33,78
Deney6	6	50,92	51,48	45,43	45,44
Deney7	7	44,38	42,60	29,85	27,53
Deney8	8	47,08	47,30	28,51	28,31
Deney9	9	73,36	73,90	66,99	63,36
Deney10	10	66,51	65,18	60,15	57,76
Deney11	11	56,77	56,80	38,01	40,02
Deney12	12	51,72	51,53	33,01	33,77

**Şekil 6.10** E037-N39 haritası üzerinde yapılan deneylerin sonuçlarının grafiksel gösterimi

Tablo 6.11’de E037-N37 haritası üzerinde yapılan deneyler ve sonuçları gösterilmektedir. Bu tablodaki değerlere göre en iyi sonuçlar çoğunlukla Genetik Algoritma (GA) yöntemi ile bulunduğu görülmektedir. Ayrıca yerel arama işleminin bu harita için genel olarak GA yöntemini iyileştirmedeği fakat PSO yöntemini iyileştirdiği görülmektedir.

Tablo 6.11 E037-N37 haritası üzerinde yapılan deneyler ve sonuçları

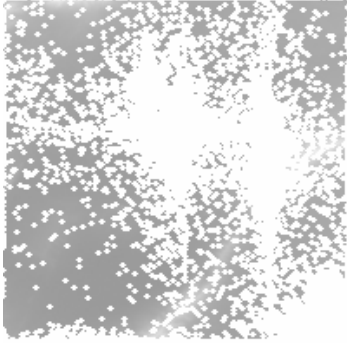
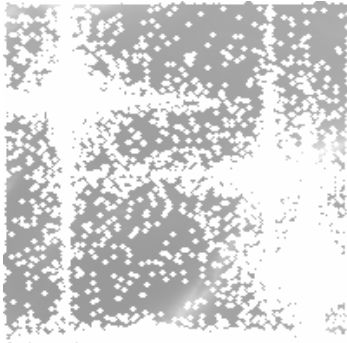
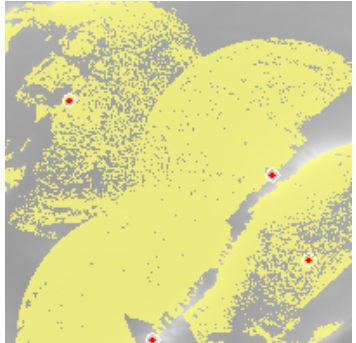
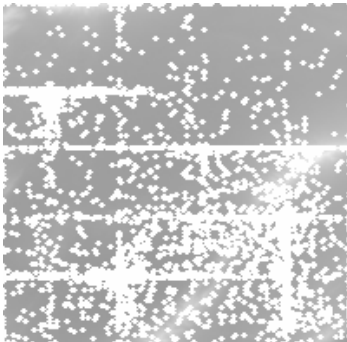
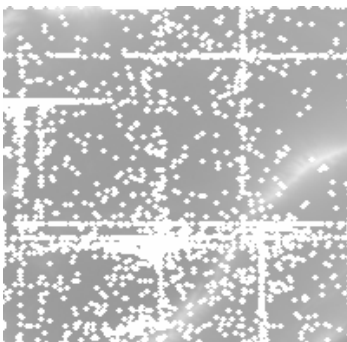
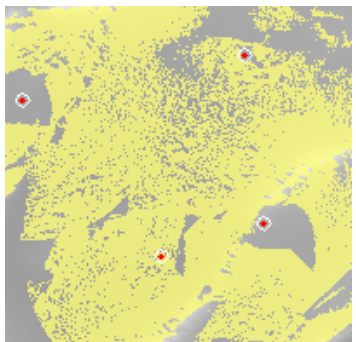
Deney Adı	Test No	GA	YGA	PSO	YPSO
Deney1	1	62,99	62,27	61,95	58,84
Deney2	2	56,19	55,24	47,19	46,99
Deney3	3	48,58	48,16	36,05	38,76
Deney4	4	44,94	43,36	27,62	26,78
Deney5	5	46,67	46,10	48,32	48,42
Deney6	6	48,75	49,05	39,39	41,50
Deney7	7	40,57	39,86	30,15	33,43
Deney8	8	43,44	42,34	24,57	26,63
Deney9	9	79,83	79,36	70,35	73,65
Deney10	10	61,53	60,54	54,45	55,45
Deney11	11	55,55	54,38	43,63	44,14
Deney12	12	46,36	46,64	28,33	26,57



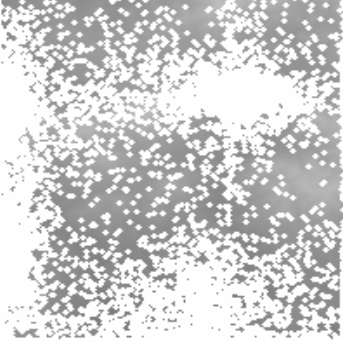
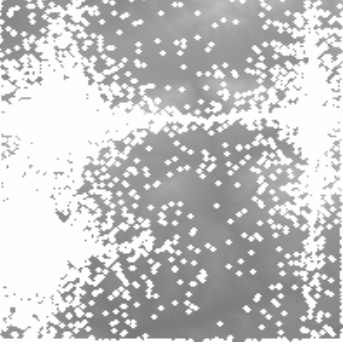
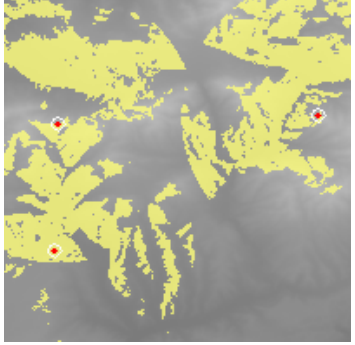
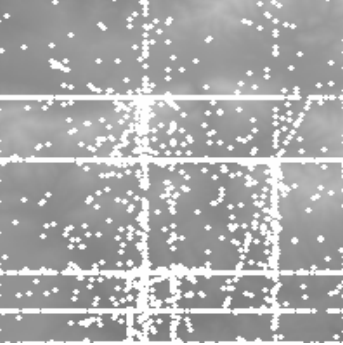
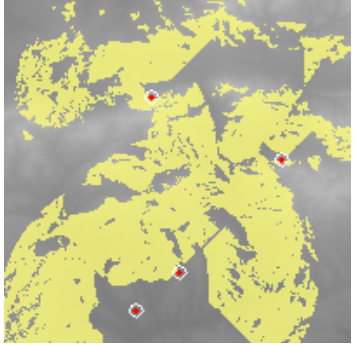
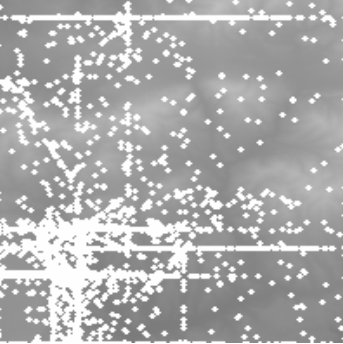
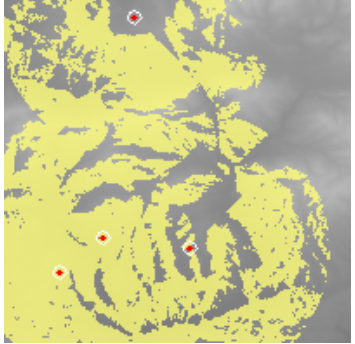
Şekil 6.11 E037-N37 haritası üzerinde yapılan deneylerin sonuçlarının grafiksel gösterimi

Aşağıdaki tablolarda bulunan şekiller, Deney 3 alanı için kapsama haritası ve algoritmaların iterasyonları esnasında denemiş olduklar sensör pozisyonlarını göstermektedir. Soldaki haritalarda beyaz noktalar ile belirtilen şekiller denenmiş olan sensör pozisyonlarını göstermektedir. PSO iterasyonlarında GA yöntemine göre daha fazla sensör pozisyonu denenmiş olduğu görülmektedir. Sağ taraftaki şekiller algoritmaların bulduğu sensör pozisyonları ile kapsanan alanı göstermektedir. Genel olarak GA ile bulunan sonuçların daha fazla alanı kapsadığı şekillerde görülmektedir.

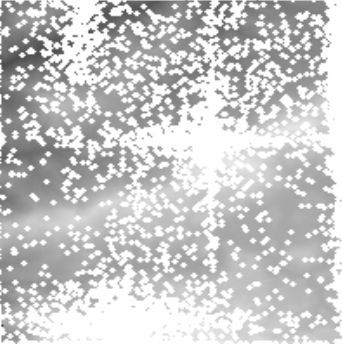
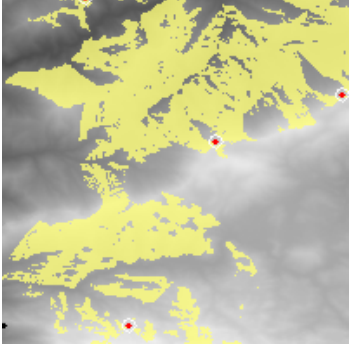
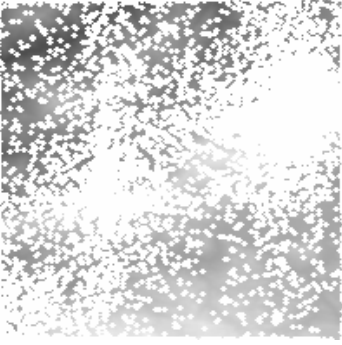
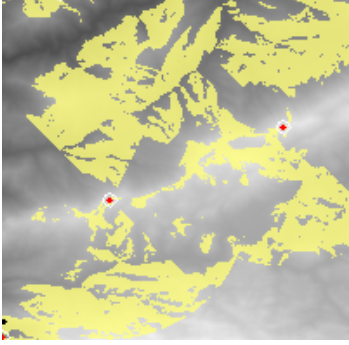
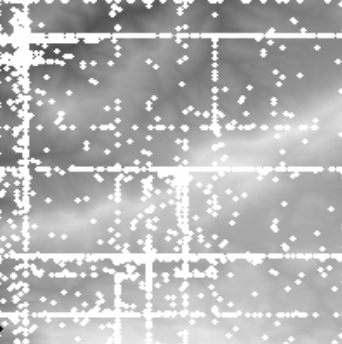
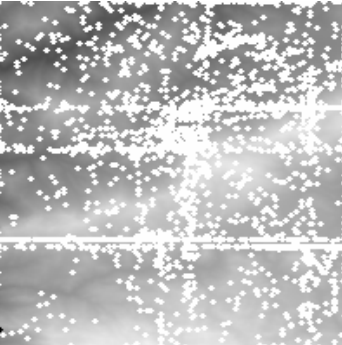
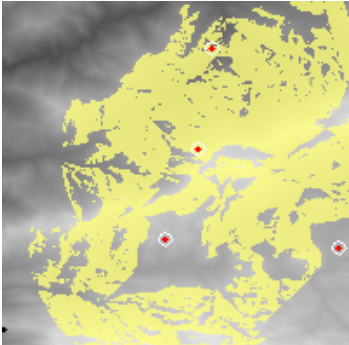
Tablo 6.12 E037-N33 haritası üzerinde Deney 3 testlerinden örnek bir kapsama haritası ve algoritmaların iterasyon esnasında denedikleri noktalar

Açıklama	Denenmiş Noktalar	Sensörlerin Kapsadığı Alan
Deney 3 – E037-N33 Haritası - PSO yöntemi		
Deney 3 – E037-N33 Haritası - YPSO		
Deney 3 – E037-N33 Haritası - GA		
Deney 3 – E037-N33 Haritası - YGA		

Tablo 6.13 E037-N39 haritası üzerinde Deney 3 testlerinden örnek bir kapsama haritası ve algoritmaların iterasyon esnasında denedikleri noktalar

Açıklama	Denenmiş Noktalar	Sensörlerin Kapsadığı Alan
Deney 3 – E037-N39 Haritası - PSO yöntemi		
Deney 3 – E037-N39 Haritası - YPSO		
Deney 3 – E037-N39 Haritası - GA		
Deney 3 – E037-N39 Haritası - YGA		

Tablo 6.14 E037-N33 haritası üzerinde Deney 3 testlerinden örnek bir kapsama haritası ve algoritmaların iterasyon esnasında denedikleri noktalar

Açıklama	Denenmiş Noktalar	Sensörlerin Kapsadığı Alan
Deney 3 – E037-N37 Haritası - PSO yöntemi		
Deney 3 – E037-N37 Haritası - YPSO		
Deney 3 – E037-N37 Haritası - GA		
Deney 3 – E037-N37 Haritası - YGA		

7. SONUÇLAR

Genel olarak testlere bakıldığında 3 farklı engebe seviyesine sahip alan üzerinde farklı büyüklükte ve farklı sensör sayıları ile testler yapılmıştır. Bu testlerde amaçlanan algoritmaların farklı alan tiplerindeki etkinlikleri ve aynı alan üzerinde sensör sayısındaki değişimin etkinliğe etkisinin incelenmesiydi.

Genetik Algoritma (GA) çözümünün davranışı incelendiğinde, $\alpha x_{ei} + (1 - \alpha)y_{ei}$ formülüne göre, seçilen iki kromozomun genlerinin bir α parametresi ile değiştirilmesi sonucunda yeni sensörün pozisyonu seçilen iki sensörün arasında bir konumda çıkması gerektiğini göstermektedir. GA ile arama yapılması sonucunda denenmiş noktalara bakılınca çizgisel şekiller görülmesi bu yaklaşımın bir neticesi olarak çıkmıştır. Ayrıca GA ile denenmiş sensör pozisyon miktarı PSO yöntemine göre oldukça azdır. GA çözümü ile elde edilen sonuçların neredeyse hepsi PSO ve YPSO yöntemlerine göre daha etkin değerlerdedir. E037-N33, E037-N39 haritaları ile yapılan deneylerde GA ile YGA çözümleri yakın değerlerde etkinlik sonuçları vermiştir. E037-N37 haritası ile yapılan testlerde GA çözümü diğer çözüm yöntemlerine göre daha etkin sonuçlar vermiştir. Yerel Arama algoritmasının GA algoritmasında belirgin bir performans artışı gösterdiği görülmemiştir.

Yerel Aramalı Genetik Algoritma (YGA) yönteminin davranışı da GA ile aynı olmakla birlikte algoritmanın belirli iterasyonlarında yerel arama uygulanması ile detaylı arama işleminin güçlendirilmesi hedeflenmiştir. Elde edilen sonuçlar PSO ve YPSO yöntemlerine göre daha iyi olmakla birlikte GA yöntemine göre daha iyi sonuçlar verdiğini söyleyemeyiz.

Parçacık Sürü Optimizasyonu (PSO) doğası gereği tüm arama alanında çok fazla sensör pozisyonu denendiğini görülmektedir. PSO yöntemi özellikle iyi sensör pozisyonları (gBest) çevresinde daha detaylı aramalar gerçekleştirmiştir. Bu yöntem GA yöntemine göre çok daha fazla yer denemesine karşılık daha iyi sonuçlar elde edememiştir.

Yerel Aramalı Parçacık Sürü Optimizasyonu (YPSO) ile detaylı arama mekanizması daha da geliştirilmiş ve E037-N37 haritasındaki bazı testlerde belirgin

bir performans artışı gözlemlenmiştir. Ancak diğer testlerde genel olarak YPSO ve PSO yakın değerlerde sonuçlar üretmiştir.

GA ve YGA yöntemlerinin PSO ve YPSO yöntemlerine göre standart sapma değerleri daha düşük olduğu belirlenmiştir. Bu sonuçlara göre GA ve YGA yöntemlerinin daha dayanıklı (robust) algoritmalar olduğunu söylenebilir. Ayrıca sensör sayılarının artmasıyla PSO ve YPSO yöntemleri daha kötü sonuçlar vermeye başlamıştır.

Sonuç olarak GA yöntemi hem etkinlik hesapları hem de standart sapma sonuçları yönüyle PSO yöntemine göre daha iyi sonuçlar bulmuştur ve Sensör Yerleştirme Probleminin kullanılan modelleme şekillerine göre çözümünün GA yöntemi ile yapılması daha uygun olacaktır.

8. ÖNERİLER

Yapılan çalışma ile sensör yerleştirme problemine etkin çözümler aranmıştır. Bu çalışma baz alınarak ileride yapılabilecek araştırma önerileri şu şekilde sıralanabilir:

- Bir alanın görülebilirlik hesabının daha hızlı yapılabilmesi için yöntemlerin araştırılması ve geliştirilmesi
- Farklı meta-sezgisel yöntemlerinin problemin çözümündeki etkinliklerinin denenmesi
- Problemin farklı versiyonlarına çözüm aranması
- Daha farklı sensör modellerinin problemin çözümüne etkilerinin araştırılması

Bundan sonraki çalışmalar yukarıda bahsedilen alanlar üzerinde yoğunlaştırılacak ve bu tez çalışmasının kapsamı daha da genişletilecektir.

KAYNAKLAR

- [1] K. Chakrabarty, S. S. Iyengar, H. Qi, E.C. Cho, "*Grid Coverage of Surveillance and Target location in Distributed Sensor Networks*" To appear in IEEE Transaction on Computers, 2002
- [2] S. S. Dhillon and K. Chakrabarty, "*Sensor placement for effective coverage and surveillance in distributed sensor networks*", Proc. IEEE Wireless Communications and Networking Conference, pp. 1609--1614, 2003
- [3] T. Can, V. Isler and Z. Ipekkan, "*Sensor Optimization In a Virtual Environment*", in Proceedings of the 9th Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL, 2000
- [4] M. Marengoni, B. Draper, A. Hanson and R. Sitaraman. "*Placing Observers to Cover a Polyhedral Terrain in Polynomial Time*", Proc., 1996
- [5] M. Marengoni, B.A. Draper, A. Hanson, R.A. Sitaraman, "*System To Place Observers On A Polyhedral Terrain In Polynomial Time*", Image and Vision Computing, vol.18, pp. 773-80, 1996
- [6] C. Undeger, M. Balci, S. Girgin, V. Koc, F. Polat, S. Bilir and Z. Ipekkan. "*Sensor Platform Optimization and Simulation for Surveillance of Large Scale Terrains*", Proceedings of I'ITSEC 2002 Conference on Modeling and Simulation, Orlando, Florida, 2002
- [7] Hansen, P., Mladenovic N. "*An introduction to variable neighborhood search, in Metaheuristics, advances and in local search paradigms for optimization*", i.S.V.e. al, Kluwer Academic Publishers: Dordrecht. 433-458, 1999
- [8] Goldberg D.E., "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley, USA, 1989
- [9] F. Herrera, M. Lozano, "*Tackling real-coded genetic algorithms: Operators and tools for the behavioural analysis*", Artificial Intelligence Review, 1998
- [10] R.C. Eberhard, J. Kennedy, "*New optimizer using Particle Swarm Theory*", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995
- [11] *Close Combat Tactical Trainer (CCTT)*, Version 8.4. Source Code, 2003
- [12] *Dismounted Infantry Semi-Automated Forces (DISAF)*, Version 9.4. Source Code, 2002

- [13] J. E. Bresenham, "*Algorithm for computer control of a digital plotter*", IBM Systems J., 4(1), pp.25-30, 1965
- [14] National Imagery and Mapping Agency. *Digital Terrain Elevation Data (DTED). Standards and Specifications Publications: MIL-PRF-89020A Amendment-1*, 27, 1999

ÖZGEÇMİŞ

Ali Tozan, 1982 yılında Denizli’de doğdu. İlkokul öğrenimini Mareşal Fevzi Çakmak İlkokulu’nda, Orta öğrenimini Atatürk Ortaokulu’nda tamamladı. Lise öğrenimini Kazım Kaynak Süper Lisesi’nde tamamladı ve 2000 yılında Marmara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği bölümüne girmeye hak kazandı. Buradaki öğrenimini 2005 yılında tamamladı ve aynı yıl Gebze Yüksek Teknoloji Enstitüsü’nde Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans öğrenimine başladı. Halen TÜBİTAK, Marmara Araştırma Merkezi, Bilişim Teknolojileri Enstitüsü’nde araştırma faaliyetlerini sürdürmektedir.