

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MEKÂNSAL KARAR DESTEK SİSTEMİ UYGULAMASI
OLARAK WEB TABANLI EMLAK DEĞER HARİTASI
TASARIMI VE GELİŞTİRİLMESİ**

Ayşe ERDOĞAN

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Galip AYDIN

AĞUSTOS-2011

**T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MEKÂNSAL KARAR DESTEK SİSTEMİ UYGULAMASI OLARAK
WEB TABANLI EMLAK DEĞER HARİTASI TASARIMI VE GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Ayşe ERDOĞAN

08129101

**Tezin Enstitüye Verildiği Tarih: 20 Temmuz 2011
Tezin Savunulduğu Tarih: 03 Ağustos 2011**

Tez Danışmanı: Yrd. Doç. Dr. Galip AYDIN (F.Ü)

Diğer Jüri Üyeleri: Doç. Dr. Mehmet KAYA (F.Ü)

Yrd. Doç. Dr. Mustafa TÜRK (F.Ü)

AĞUSTOS-2011

ÖNSÖZ

Bu tez çalışmasında, günümüzde bilişim sistemleri içerisinde yaygın bir kullanıma sahip olan mekânsal verilerin bir Karar Destek Sistemine entegrasyonu ile Elazığ ili Merkez ilçesi için Emlak Değer Haritası niteliğinde web tabanlı görsel bir uygulama gerçekleştirilmesi amaçlanmıştır.

Emlak değerlemede daha önce kullanılmamış bir yöntem izlenerek, başarılı sonuçlar elde edilmiştir. Ayrıca mekânsal verilerin bilgisayar ve web ortamına aktarılması sürecinde yeni gelişen teknolojilerden faydalanılmıştır.

Bu çalışmalarım süresince benden yardımlarını esirgemeyen saygıdeğer hocam Sayın Yrd. Doç. Dr. Galip AYDIN'A ve her zaman destekleriyle arkamda duran aileme katkılarından dolayı teşekkür ederim.

Ayşe ERDOĞAN
ELAZIĞ - 2011

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	ii
İÇİNDEKİLER	iii
ÖZET	vi
SUMMARY	vii
ŞEKİLLER LİSTESİ	viii
TABLolar LİSTESİ	x
SEMBOLLER LİSTESİ	xi
1. GİRİŞ	1
1.1. Sistem Mimarisi	2
2. COĞRAFI BİLGİ SİSTEMLERİ VE COĞRAFI İLİŞKİSEL VERİ TABANLARI... 5	5
2.1. Coğrafi Bilgi Sistemleri	5
2.1.1. Coğrafi Bilgi Sistemlerinde Kullanılan Veri Tipleri ve Özellikleri	7
2.1.1.1. Vektör Veriler	8
2.1.1.2. Raster Veriler	8
2.1.1.3. Vektör ve Raster Veriler Arasındaki Farklar	9
2.2. Coğrafi İlişkisel Veri Tabanları (The Geo-Relational Database)	10
2.2.1. MySQL Veri Tabanı	11
2.2.1.1. MySQL Tablo Tipleri	12
2.2.1.2. MySQL Control Center	12
2.2.2. PostgreSQL Veri Tabanı	13
2.2.3. PostGIS (GeoDatabase)	15
2.2.3.1. PostGIS'in Özellikleri	15
2.3. Emlak Değer Haritası Uygulamasında PostgreSQL ve PostGIS Kullanımı	19
3. OGC STANDARTLARI VE GEOSERVER	20
3.1. OGC Standartları	20
3.1.1. OpenGIS Standartları ve Özellikleri	20
3.1.1.1. OpenGIS Standartları	20
3.1.1.2. Özet Modeller	21
3.1.1.3. OpenGIS Referans Modeli (ORM)	21
3.2. GML (Geographic Markup Language)	21

3.3. WMS (Web Map Service)	22
3.3.1. WMS Fonksiyonları	24
3.3.1.1. GetCapabilities	25
3.3.1.2. GetMap	25
3.3.1.3. GetFeatureInfo	26
3.4. WMS – C	26
3.5. WFS (Web Feature Service)	26
3.6. WCS (Web Coverage Service)	28
3.7. GeoServer	29
3.7.1. GeoServer Kullanarak Uygulama Geliştirme	30
3.8. Emlak Değer Haritası Uygulamasında GeoServer Kullanımı	39
4. MEKÂNSAL KARAR DESTEK SİSTEMİ UYGULAMASI OLARAK EMLAK	
DEĞER HARİTASI GELİŞTİRİLMESİ	44
4.1. Karar Destek Sistemleri	44
4.1.1. Karar Destek Sistemlerinin Coğrafi Bilgi Sistemlerinde Kullanımı	45
4.1.1.1. Mekânsal Karar Destek Sistemlerinde İş Akışı	49
4.2. Emlak Değerlerinin Belirlenmesi	50
4.2.1 Emlak Değerlemede Kullanılan Yöntemler	50
4.2.1.1. Nominal Yöntem ile Emlak Değerleme	50
4.2.1.2. Puanlama Yöntemi ve Çoklu Karar Analizinin Emlak Değerlemede Kullanımı	52
4.2.1.3. Genetik Algoritmalar ile Emlak Değerleme	52
4.2.1.4. Hesapsal Zekâ ile Gerçek Emlak Kitlesel Değerinin Tahmini	53
4.3. Emlak Değerlemede Kullanılacak Yöntemin Belirlenmesi	54
4.4. Emlak Değerine Etki Eden Faktör ve Faktör Puanlarının Belirlenmesi	54
4.5. Emlak Değerine Etki Eden Faktör Ağırlıklarının Belirlenmesi	62
4.5.1. Sezgisel (Heuristik) Algoritmalar	62
4.5.2. Harmoni Arama Algoritması	64
4.5.2.1. Algoritmada Kullanılan Değişkenler	66
4.5.2.2. Harmoni Arama Algoritması İşlem Adımları	67
4.5.2.3. Harmoni Arama Algoritmasıyla Örnek Bir Uygulama	72
4.5.2.4. Harmoni Arama Algoritması ile Faktör Ağırlıklarının Hesaplanması	75
5. KULLANICI ARAYÜZ TASARIMINDA YARARLANILAN TEKNOLOJİLER. ..	80
5.1. JSP	80

5.1.1. JSP'nin Avantajları	80
5.2. JavaScript	81
5.3. Ext JS Kütüphanesi	82
5.3.1. Ext JS'in Avantajları	82
5.4. AJAX	82
5.4.1. AJAX'ın Avantajları	83
5.5. JSP, JavaScript, Ext JS ve AJAX Teknolojiler Kullanılarak Arayüz Tasarımı	83
6. SONUÇ	87
KAYNAKLAR	89
ÖZGEÇMİŞ	95

ÖZET

Emlak; eğitim ve sağlık imkânları, sosyal imkânlar ve iş fırsatlarına açılan bir kapıdır. Bu itibarla emlak seçimi de, insan hayatında önemli rolü olan bir karardır. Bu karar sürecinde, emlak alıcı ve satıcılarına, emlakçılara ve diğer ilgililere fikir vermek amacıyla, bu tez çalışmasında Elazığ ili Merkez ilçesi pilot bölge seçilerek, mekânsal verilere dayalı web tabanlı bir Karar Destek Sistemi uygulaması olarak Emlak Değer Haritası gerçekleştirilmiştir.

Çalışmada, mekânsal verileri depolama işi, coğrafi ilişkisel bir veri tabanı olan PostgreSQL ve eklentisi PostGIS kullanılarak gerçekleştirilmiştir. Mekânsal verileri modellemek ve görüntülemek içinse, harita ve coğrafi bilgi sunucu olarak GeoServer'dan faydalanılmıştır. Emlak değerleri, ağırlıklandırılmış faktör puanları yöntemine dayalı olarak hesaplanırken, coğrafi veri sorgulamaları Java'da gerçekleştirilmiş ve Harmoni Arama Algoritması Matlab'da yazılmıştır. Emlak değerlemede, kabul edilebilir hata oranlarına sahip, gerçeğe yakın sonuçlar elde edilmiştir. Hesaplanan emlak değerleri Java ve JavaScript teknolojilerinden faydalanılarak hazırlanan web tabanlı bir arayüz ile kullanıcıya sunulmuştur.

Sonuç olarak mekânsal verilerin karar alma sürecinde kullanılmasına olanak tanıyan, web tabanlı bir Mekânsal Karar Destek Sistemi geliştirilmiş ve sonuçlar online haritalar üzerinden sunulmuştur.

Anahtar Kelimeler: Emlak Değer Haritası, Coğrafi İlişkisel Veri Tabanları, Harita Sunucular, Sezgisel Optimizasyon Algoritmaları, Harmoni Arama Algoritması, Mekânsal Karar Destek Sistemleri, Web Tabanlı Uygulamalar

SUMMARY

Design and Development of a Web-Based Real Estate Value Map as Application of Spatial Decision Support System

Real estate is a gateway to education and health facilities, social facilities and business opportunities. Therefore, estate selection is a decision which plays an important role in human life. In this thesis, for the purpose of giving an idea to real estate buyers and sellers, estate agents and other interested persons in decision-making process, the downtown of the city of Elazığ was selected as pilot region and as an application of web-based Decision Support System for geo-spatial data Real Estate Value Map is implemented.

In this study, being a geo-relational database PostGreSQL and its plug-in PostGIS were used for storing spatial data. GeoServer which is a map server as well as a geographic information server was used for modeling and viewing spatial data. Estate values were calculated based on weighted factor score, geographic data queries were implemented in Java and Harmony Search Algorithm was written in Matlab. In the real estate valuation, results which were close to real values were obtained with acceptable error rates. Calculated real estate values were presented to users with a web-based interface using Java and JavaScript technologies.

As a result, a web-based Spatial Decision Support System which allows use of spatial data in decision-making process is developed and results are shown on online maps.

Key Words: Real Estate Value Map, Geo-Relational Databases, Map Servers, Heuristic Optimization Algorithms, Harmony Search Algorithm, Spatial Decision Support Systems, Web-Based Applications

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. Emlak Değer Haritası uygulamasının sistem mimarisi	3
Şekil 2.1. Mekânsal ve tablosal veriler arasındaki ilişki	5
Şekil 2.2. Gerçek dünya verilerinin vektör ve raster veriye dönüştürülmesi	9
Şekil 3.1. WMS sınıf diyagramları	24
Şekil 3.2. WMS (Web Map Service) 'de iş akışı	25
Şekil 3.3. WFS (Web Feature Service) sınıflar diyagramları	27
Şekil 3.4. WCS (Web Coverage Service - Kayıt/Katalog servisi) sınıf diyagramları	28
Şekil 3.5. Coğrafi Bilgi Sistemlerinde kullanılan temel servisler arasındaki iş akışı	29
Şekil 3.6. GeoServer çalışma ortamı	30
Şekil 3.7. GeoServer'da yeni çalışma uzayı (workspace) oluşturma ekranları	31
Şekil 3.8. GeoServer'da yeni saklama alanı (store) oluşturma ekranları	31
Şekil 3.9. GeoServer'da yeni katman (layer) oluşturma ekranları	32
Şekil 3.10. GeoServer'da oluşturulan yeni katman için harita sınırlarının belirlenmesi	32
Şekil 3.11. GeoServer'da oluşturulan yeni katman için stil seçimi	33
Şekil 3.12. GeoServer'da KML format ayarlarının düzenlenmesi	33
Şekil 3.13. GeoServer'da yeni katman grubu oluşturulması	34
Şekil 3.14. GeoServer'da SLD kodu ile yeni stil tanımlama	35
Şekil 3.15. Point şeklindeki verilerin GeoServer'da gösterimine bir örnek	37
Şekil 3.16. LineString şeklindeki verilerin GeoServer'da gösterimine bir örnek	37
Şekil 3.17. Poligon şeklindeki verilerin GeoServer'da gösterimine bir örnek	38
Şekil 3.18. Coğrafi veri katmanları	38
Şekil 3.19. Birden çok katmandan oluşan verilerin GeoServer'da gösterimine bir örnek	39
Şekil 3.20. Mahalleler katmanının GeoServer'daki görüntüsü	41
Şekil 3.21. GeoServer'da Emlak Değer Haritasına ait katman grubu	41
Şekil 3.22. GeoServer'da emlak değerlerinin görüntülenmesi	42
Şekil 3.23. GeoServer'da emlak değer aralıklarına göre binaların farklı renklerde görüntülenmesi (1)	42
Şekil 3.24. GeoServer'da emlak değer aralıklarına göre binaların farklı renklerde görüntülenmesi (2)	43
Şekil 4.1. Coğrafi Bilgi Sistemleri ve Karar Destek Sistemleri arasındaki ilişki	46

Şekil 4.2. Mekânsal Karar Destek Sistemi Mimarisi	47
Şekil 4.3. Karar destek sistemi ve CBS entegrasyonunun değerlendirilmesi	49
Şekil 4.4. Müzikte harmoniyi (uyumu) arama	64
Şekil 4.5. Müzikte Harmoni Arama yaklaşımına bir örnek	65
Şekil 4.6. Harmoni Arama Algoritması işlem adımları	67
Şekil 4.7. Müzik enstrümanının notaları sırasıyla çalması	68
Şekil 4.8. Tercih edilen tonların Harmoni Hafızasında saklanması	69
Şekil 4.9. x^+ veya x^- , $x \in$ Tercih edilen notalar	70
Şekil 4.10. Müzikte harmoniyi bozan durumlar	71
Şekil 4.11. $m=1$ ve iterasyon sayısı 1000 iken $f(x_i)$ grafiği	73
Şekil 4.12. $m=0.1$ ve iterasyon sayısı 3000 iken $f(x_i)$ grafiği	74
Şekil 4.13. $m=0.1$ ve iterasyon sayısı 3000 iken $f(x_i)$ grafiği	75
Şekil 4.14. Kabul edilebilir hata oranı %10 ve iterasyon sayısı 10.000 iken elde edilen grafik	78
Şekil 4.15. Kabul edilebilir hata oranı %5 ve iterasyon sayısı 20.000 iken elde edilen grafik	78
Şekil 4.16. Hesaplanan emlak değerlerinin, gerçek emlak değerlerinden yüzdesel farkını gösteren grafik	79
Şekil 5.1. Emlak Değer Haritası kullanıcı arayüzü (1)	85
Şekil 5.2. Emlak Değer Haritası kullanıcı arayüzü (2)	86

TABLÖLER LİSTESİ

	<u>Sayfa No</u>
Tablo 4.1. Bilgisayar sistemlerindeki optimizasyon problemi ile müzikte harmoniyi (uyumu) arama probleminin karşılaştırılması	65

SEMBOLLER LİSTESİ

FP	: Emlak değerlemede faktör puanı
FA	: Emlak değerlemede faktör ağırlığı
m	(1) : Emlak değerlemede birim arsa değeri
son_alan, alan,	
max_alan	: Emlak değeri için faktör puanlarının hesaplanmasında kullanılan değişkenler
x_1, x_2, x_3	: Müzikte Harmoni Arama için karar değişkenleri
HM	: Harmoni Hafızası
HMS	: Harmoni Hafızasının Boyutu
HMCRCR	: Harmoni Hafızasının Dikkate Alınma Oranı
RS	: Rastgele Seçim
PAR	: Ton/Düzey Ayarlama Oranı
m	(2) : PAR uygulamasında yardımcı değişken
f(.)	: Harmoni Arama amaç fonksiyonu
x_i	: Harmoni Aramada her bir karar değişkeni için mümkün değerler dizisi
i	: Harmoni Arama vektörünün karar değişken sayısı
X_{HMS}	: Harmoni Arama karar değişkeni olan x_i 'lerin birleşiminden oluşan çözüm vektörü
x'_i	: Harmoni Aramada yeni harmoni için karar değişkeni
X_i	: Harmoni Aramada X_{HMS} dışından rastgele seçilen karar değişkenleri
α	: Harmoni Aramada sürekli değerlerin hesaplanmasında kullanılan değişken
bw	: Harmoni Aramada devamlı değişkenler için keyfi bant genişliği mesafesi
u	: Harmoni Aramada sürekli değerlerin hesaplanmasında kullanılan dizi
x_{new}	: Harmoni Aramada oluşturulan yeni harmoni vektörü
x_{old}	: Harmoni Hafızasından atılacak harmoni vektörü
$y[,]$	: Emlak faktör puanları ve emlak değerlerinin tutulduğu matris
en_dogru[,]	: Her emlak için en iyi harmoni vektörlerinden oluşan matris

KISALTMALAR LİSTESİ

CBS	: Coğrafi Bilgi Sistemleri
VTYS	: Veri Tabanı Yönetim Sistemleri
SLD	: Styled Layer Descriptors
OGC	: Open Geospatial Consortium
ORM	: OpenGIS Reference Model
GML	: Geographic Markup Language
XML	: Extensible Markup Language
WMS	: Web Map Service
SRS	: Spatial Reference System
EPGS	: European Petroleum Survey Group
WMS – C	: Web Map Service – Cache
WFS	: Web Feature Service
UVDF	: Ulusal Veri Değişim Formatı
WCS	: Web Coverage Service
JSP	: Java Server Pages
AJAX	: Asenkron JavaScript ve XML

1. GİRİŞ

Bilişim teknolojilerindeki baş döndüren değişim ve gelişime paralel olarak, Coğrafi Bilgi Sistemleri (CBS) de yakın zamanda çok hızlı bir gelişim göstermiş, bir zamanlar sadece uzmanların ilgi alanına giren haritalar, artık modern insanın günlük hayatının vazgeçilmez bir parçası haline gelmiştir. Bilgisayar teknolojilerindeki gelişmelerin CBS alanına en önemli yansımalarından birinin; ilgili verilerin haritalarla dijital ortama aktararak toplumsal kullanıma açılması olduğu söylenebilir. Oysa yakın geçmişe kadar mekânsal verilerin depolanması, bilgisayar sistemleri için büyük kaynak kullanımını gerektirmekteydi. Coğrafi verileri kolay şekilde depolayabilme özelliğine sahip veri tabanları ya sadece bazı kurum veya laboratuvarlarda bulunmaktaydı, ya da araştırmacı ve son kullanıcıların ulaşabileceği kadar ucuz değildi. Fakat son yıllarda yazılım sektöründeki gelişmelere paralel olarak, mekânsal verilerin kodlanarak tutulmasına imkân tanıyan, böylelikle daha az kaynak kullanarak depolamayı gerçekleştirebilen, kullanımı kolay, hızlı ve güvenilir veri tabanları üretilmiştir. Bu sayede mekânsal verilerin toplanması, depolanması, işlenmesi, analiz edilmesi, modellenmesi ve görüntülenmesinde kullanılan CBS, bu tez çalışmasında olduğu gibi, günlük hayatımızda da sıkça kullanılan önemli bir kavram haline gelmiş bulunmaktadır [1].

Emlak; eğitim ve sağlık imkânları, sosyal imkânlar ve iş fırsatlarına açılan bir kapıdır. Buna paralel olarak, insanların nerede yaşadıkları, kim olduklarının bir yansıması olarak düşünülebilir. Aynı zamanda kim olabilecekleri üzerinde de etkilidir [2]. Bu derece önem arz eden bir konuda karar verirken, çok çeşitli faktörler bu karar üzerinde etkili olmaktadır.

Şehirler için emlak değerini etkileyen faktörler ve bunların yaşam alanı özelliği taşıyan emlağın değerine etkisi bu tez çalışmasının konusunu oluşturmaktadır. Çalışmalar sonucunda, mekânsal veriler için web tabanlı bir Karar Destek Sistemi uygulaması olarak Emlak Değer Haritası geliştirilmiştir. Sistem; emlak alıcı ve satıcılarına, emlakçılara ve tüm ilgililere, emlak değerine etki eden faktörler ve bunların etki etme yüzdelerine bağlı olarak ve online haritalar kullanılarak, şehrin seçilen herhangi bir bölgesi için emlak değerlerini hesaplamakta ve sonuçları kolay anlaşılır bir şekilde kullanıcıya sunmaktadır.

Geliştirilen Emlak Değer Haritası Sistemi, Elazığ ili Merkez ilçesinin mahalle sınırları içerisinde, **74419.525 km²** 'lik bir alan üzerinde bulunan 37462 adet binanın emlak değerlerini hesaplamak suretiyle oluşturulmuştur. Merkezde 36 adet mahalle bulunmakta olup, bunlardan bölünmüş parselde sahip 3 tanesi de ayrı kabul edilerek, 39 mahalle üzerinden çalışmalar yürütülmüştür. Çalışmada kullanılan mekânsal veriler; yerel harita-kadastro bürolarından ve

açık kaynak veri depolarından elde edilmiştir. Emlak değerini belirlemek için oluşturulan kriterler, Elazığ ili şartları göz önünde bulundurularak belirlenmiştir.

Çalışmada, mekânsal verileri depolama işi, coğrafi ilişkisel veri tabanları kullanılarak gerçekleştirilmiştir. Mekânsal verileri modellemek ve görüntülemek içinse, internet tabanlı harita sunuculardan faydalanılmıştır. Sistemin genelinde ve emlak değerlerini hesaplamada, veri tabanı sorgulamaları için, Java programlama dili kullanılmıştır. Harmoni Arama Algoritması ise Matlab kullanılarak gerçekleştirilmiştir. Sonuç olarak Coğrafi Bilgi Sisteminin ortaya koymuş olduğu ürün Karar Destek Sistemine entegre edilerek, mekânsal verilerin karar alma sürecinde kullanılmasına olanak tanınmıştır.

1.1. Sistem Mimarisi

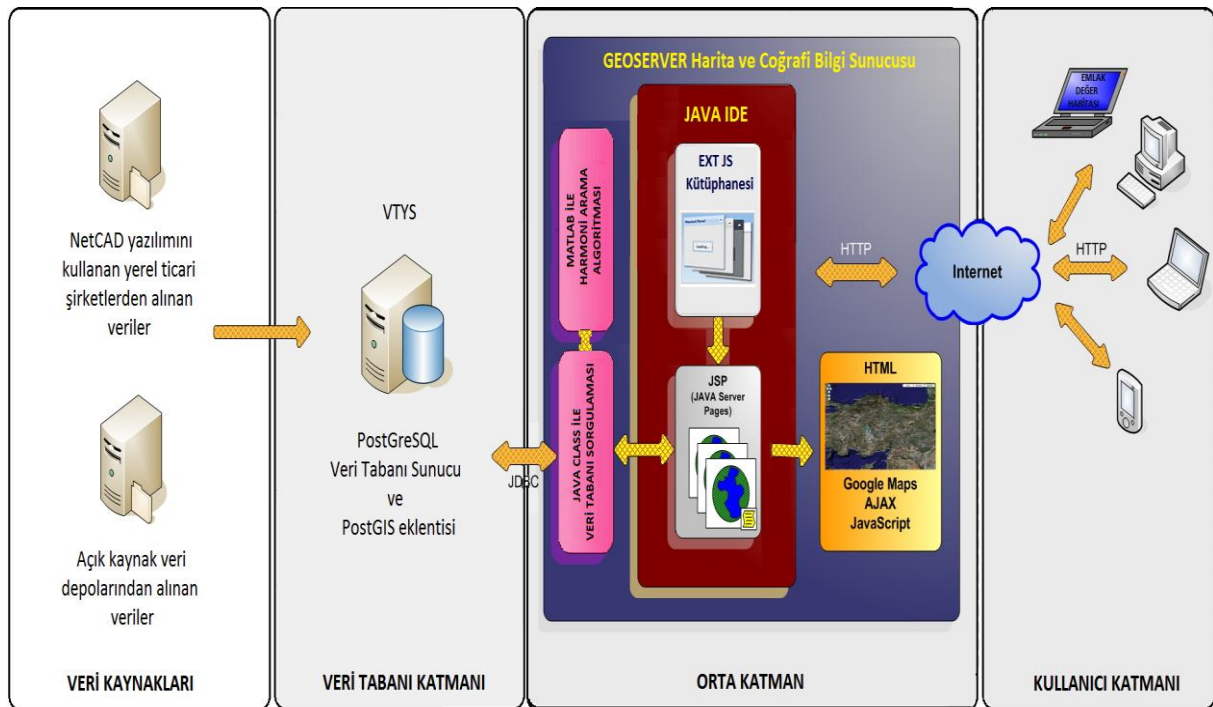
Geliştirilen Mekânsal Karar Destek Sistemi mimarisinin ana bileşenlerini şunlardır:

- Coğrafi Bilgi Sistemleri (CBS)
- Veri Tabanı Yönetim Sistemleri (VTYS) ve mekânsal veriler
- Harita Sunucuları
- Emlak Değerine etki eden faktörlerin puan ve ağırlıklarının hesaplanmasında kullanılan Java sınıfları ve Yapay Zekâ Optimizasyon Algoritmalarından biri olan Harmoni Arama Algoritması
- Web tabanlı kullanıcı arayüzü

Oluşturulan Mekânsal Karar Destek Sistemini yapılandıran anahtar teknolojilerinde başında; coğrafi ilişkisel bir veri tabanı olarak PostgreSQL [3] ve mekânsal verileri tutmaya ve coğrafi sorgulamalar yapmaya imkân tanıyan eklentisi PostGIS [4] gelmektedir. Bunun yanı sıra; mekânsal verilerin görüntülenmesinde, harita sunucu olarak GeoServer [5] önemli yer tutmaktadır. Mekânsal verilerin karar alma sürecinde kullanılması ve web tabanlı olarak sonuçların kullanıcıya sunulması sürecinde Java teknolojilerinden ve Matlab'dan faydalanılmıştır. Özellikle haritaların bir web uygulamasıyla kullanıcı ile buluşmasında Ext JS Kütüphanesi [6] önemli rol oynamıştır.

Mekânsal verilere dayalı web tabanlı bir Karar Destek Sisteminde kullanılacak teknolojilerin incelenmesinin ardından, bu bilgiler ışığında Emlak Değer Haritası uygulaması için sistem mimarisi aşağıdaki şekilde belirlenmiştir. Sistem temel olarak şu katmanlardan oluşmaktadır:

- Mekânsal verilerin saklandığı bir veri tabanı katmanı
- Java uygulamalarından yararlanarak, emlak değerine etki eden faktör puanlarının hesaplanmasını, Yapay Zekâ Optimizasyon Algoritmalarından Harmoni Arama Algoritmasını kullanarak Matlab’da emlak değerine etki eden faktör ağırlıklarının hesaplanmasını, bunların ışığında binaların emlak değerlerinin belirlenmesini, harita üzerinde mekânsal verilerin ve hesaplanmış olan emlak değerinin görüntülenmesini sağlayan ve web sunucusu üstünde çalışan bir orta katman
- Ve son olarak, kullanıcılara sistemle etkileşim olanağı tanıyan bir web arayüzü



Şekil 1.1. Emlak Değer Haritası uygulamasının sistem mimarisi

Emlak Değer Haritası uygulaması için şu teknolojilerden faydalanılmıştır: Coğrafi Bilgi Sistemi, Coğrafi İlişkisel Veri Tabanı olarak PostgreSQL ve PostGIS eklentisi, OGC Standartları ve bu standartlara dayanan WMS, WFS, WCS gibi servisler, harita sunucu (map server) olarak GeoServer, SLD (Styled Layer Descriptors - Katman Stil Tanımlayıcıları), Java bileşenlerinden JSP, JavaScript, Ext JS Kütüphanesi, Ajax, bunun yanı sıra Karar Destek Sistemi içerisinde Optimizasyon Algoritmalarından biri olan Harmoni Arama (Harmony Search) Algoritması.

Kullanılacak sistemler ve her bir katmanda çalışacak teknolojilerin incelenmesinin ardından, mekânsal verilere dayalı web tabanlı bir Karar Destek Sistemi uygulaması olarak Emlak Değer Haritası geliştirilmiştir.

Tez çalışmasının organizasyonu şu şekildedir:

II. Bölümde, Coğrafi Bilgi Sistemleri (CBS) ve mekânsal verileri depolamak için kullanılan Coğrafi İlişkisel Veri Tabanları anlatılmıştır. MySQL ve PostgreSQL veri tabanları üzerinde durulmuş, PostgreSQL veri tabanlarına coğrafi nitelik kazandıran PostGIS eklentisinden bahsedilmiş, örnek mekânsal veri sorgulamaları gösterilmiştir. Ayrıca, Emlak Değeri Haritası uygulaması dâhilinde tüm bu sistemlerin kullanımı anlatılmıştır.

III. Bölümde, OGC Standartları, bu standartlara dayanan WMS, WFS, WCS gibi servisler üzerinde durulmuştur. Ayrıca bu servisleri sağlayan bir harita ve coğrafi bilgi sunucusu olan GeoServer anlatılmış ve Geoserver ile örnek uygulamalar geliştirilmiştir. Son olarak, Emlak Değer Haritası uygulaması dâhilinde GeoServer'in kullanımı gösterilmiştir.

IV. Bölümde, Karar Destek Sistemleri üzerinde durularak, Karar Destek Sistemlerinin Coğrafi Bilgi Sistemleriyle birlikte kullanımına dair bilgiler sunulmuştur. Bu uygulama dâhilinde bir Emlak Değer Haritası oluşturmak için, emlak değerine etki eden faktörler literatür taramasıyla belirlenmiş ve bu faktörler için, faktör puanları ve faktör ağırlıkları hesaplanmıştır. Faktör puanlarının hesaplanması için kullanılan sorgulamalar ve faktör ağırlıklarının hesaplanmasında kullanılan Harmoni Arama Algoritması bu bölümde anlatılmıştır.

V. Bölümde, yukarıda bahsedilen yöntemlerle hesaplanan emlak değerinin kullanıcılara sunulmasında yararlanılan teknolojiler anlatılmıştır. JSP, JavaScript, Ext JS Kütüphanesi ve Ajax üzerinde durulmuş ve Emlak Değer Haritası uygulama arayüzü verilmiştir.

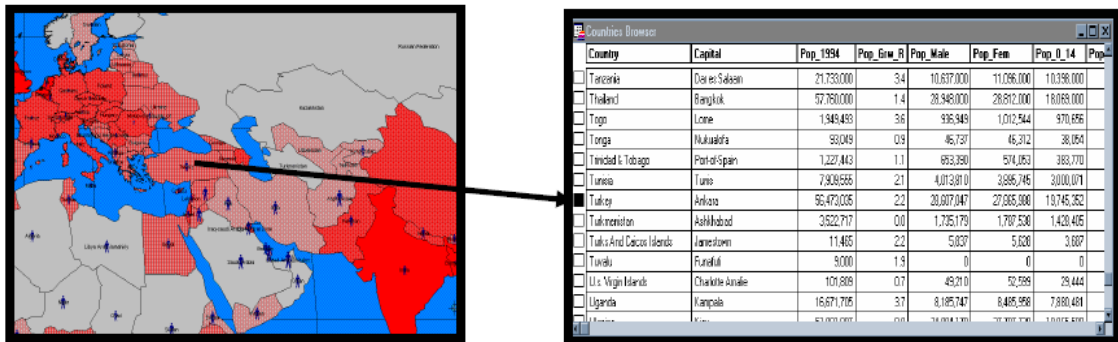
VI. Sonuç Bölümde ise, elde edilen grafiksel ve sayısal veriler ışığında, emlak değerine etki eden faktör ağırlıklarını hesaplamada Harmoni Arama Algoritmasının kullanılmasının, gerçeğe yakın değerler hesaplamak için uygun bir yaklaşım olduğu tespit edilmiştir. Bunun yanı sıra, harita sunucu olarak GeoServer'in kullanımının mekânsal verilerin görüntülenmesi için çok uygun olduğu ve Ext JS Kütüphanesi yardımıyla mekânsal verilere dayalı kullanışlı arayüzler elde etmenin mümkün olduğu sonucuna varılmıştır.

2. COĞRAFI BİLGİ SİSTEMLERİ VE COĞRAFI İLİŞKİSEL VERİ TABANLARI

2.1. Coğrafi Bilgi Sistemleri

Coğrafi Bilgi Sistemleri (CBS) (Geographic Information System), belli bir konumu ve biçimi olan nesnelere ait grafik ve grafik olmayan verilerin toplanması, depolanması, işlenmesi, analiz edilmesi, modellenmesi ve görüntülenmesini sağlayan donanım, yazılım, personel, veri ve işlemlerin bütünüdür [1]. CBS, karmaşık planlama ve yönetim sorunlarının çözümlenebilmesi amacıyla kullanılır. Coğrafi konumu belirlenmiş veriler üzerinde çalışır. Analiz etme ve karar alma süreçlerinde, çoğu bilgi mekânsal coğrafya ile ilişkilidir ve CBS, güçlü mekânsal veri işleme ve analitik yeteneğine sahip bir uzmandır [7].

Coğrafik veriler genel olarak, tablosal ve mekânsal veriler olmak üzere iki gruba ayrılır. Tablosal veriler, coğrafik nesnelerin özelliklerini ve konumlarını belirten verilerdir. Mekânsal veriler ise, dünya üzerinde yer alan nesnelerin (dağ, deniz, ova vs.) şekil ve konumlarını harita üzerinde gösteren verilerdir. Coğrafi Bilgi Sistemleri, mekânsal olmayan nitelik verilerinin yanı sıra, özellikle coğrafik referans verilerini birleştiren özel bir bilgi sistemi formudur. Ayrıca coğrafik ölçütleri yönetilen verinin doğal karakteristiğine dayalı, yapısal veri ve özetlenmiş niteliklerden oluşan hibrit bir bilgi sistemi olarak da düşünülebilir [8]. CBS, ilişkisel veri modeline dayalıdır. Böylece, coğrafi bilgi sistemleri aracılığıyla tablosal veriler ile mekânsal veriler arasında ilişki kurulmuş olur [9].



Şekil 2.1. Mekânsal ve tablosal veriler arasındaki ilişki [10].

Coğrafi Bilgi Sistemleri; haritaları tarama ve sayısallaştırma, bu süreç esnasında doğabilecek hataları düzeltme, sayısal verilerle harita verisi arasında bağlantıyı sağlama, gereksiz verileri temizleme, koordinat sistemleri ve projeksiyonlar arasında geçişi sağlama, dışarıdan veri girişi için çeşitli veri formatlarını destekleme, verilere sıkıştırılma ve hızlı erişimi olanağı sağlama, veri tipleri arasında aktarımı sağlama, gelişmiş veri girişine olanak tanıma, sorgulama, güncelleme ve raporlama işlemlerini gerçekleştirme, veri birleştirme ve gruplandırma, ara yüz desteği sağlama, nesnelerin koordinatlarını sorgulama, belirli sınırlar içinde kalan veriyi, bir nesnenin uzunluğunu, iki nokta arasındaki uzaklığı sorgulama, nesnelerin kapladıkları alanları sorgulama, nesneleri belirli özelliklerine ya da bir yere yakınlıklarına göre seçme, tematik haritalar oluşturma, haritalar üzerine açıklayıcı sembol veya yazılar ekleme, haritaları renklendirme ve verileri haritanın ölçeğine uygun olarak görüntüleme gibi işlemleri gerçekleştirebilen sistemlerdir [11].

Coğrafi Bilgi Sistemlerinin kullanım alanları yaygınlaştıkça, coğrafi verilere duyulan ihtiyaç da artmaktadır. Coğrafi Bilgi Sistemleri için veri sağlamada çeşitli yöntemler kullanılabilmektedir. Bunların başında Uzaktan Algılama teknikleri gelmektedir. Hava fotoğrafları, hava tarayıcıları ve uydu görüntüleri bu tekniğin temel veri kaynaklarıdır. Ayrıca klasik yöntemlerle elde edilen veriler de Coğrafi Bilgi Sistemi uygulamalarında rahatlıkla kullanılabilir.

Coğrafi Bilgi Sistemleri, coğrafi verilerin kullanıldığı her alana uygulanabilir. Bunun yanı sıra, yapılan araştırmalara göre Coğrafi Bilgi Sistemlerinin temel uygulama alanları şunlardır:

- Alt ve Üst Yapıların Oluşturulması: Doğal kaynakların verimli bir şekilde kullanılması amacı ile yeryüzünün üstünde ve altında dağılmış olan nesnelerin coğrafi konumlarının tespiti, sayımı ve dağılımlarının analiz edilmesi gibi konuları içerir. Örneğin; mülkiyete ait parsel kayıtlarının yapılması, altyapı ağının oluşturulması ve yönetilmesi gibi uygulama alanları vardır.
- Coğrafi Verilerin Üretimi veya Toplanması: Bir başka kullanım alanı; mekânsal veri tabanları oluşturmak amacıyla coğrafi verilerin, klasik haritacılık yöntemlerini ya da uzaktan erişim yöntemlerini kullanarak elde edilmesidir. Örneğin; mühendislik ile arazi ölçümleri, sayısal arazi ölçümleri, sayısal harita üretimi, elektronik kontrol, fiziksel ve kültürel verilerin uzaktan algılama yoluyla elde edilmesi gibi uygulamalarda kullanılır.
- Harita ve Plan Üretimi: Karmaşık ve güncel veriler üzerinde hızlıca işlem yapma olanağı sağlaması nedeniyle, planlama ve haritacılık sektörlerinde sıklıkla kullanılırlar.

Örneğin; planimetrik, topografik ve tematik haritaların oluşturulmasında tercih edilirler.

- Kaynak Tahsisi: Doğal ve yapay kaynakların, çeşitli kriterlere göre tahsisi için, sayı ve hareketlerin değerlendirilmesi amacıyla uygulanır. Pazarlama ve satış bölgesinin planlanması, hizmet ağı oluşturulması ve öğrenci yerleştirme gibi konularda Coğrafi Bilgi Sistemi uygulamaları kullanılır.
- Rota Seçimi ve Akış Optimizasyonu: Okul, posta ya da kargo araçlarının güzergâhlarının ve zamanlamaların yönetiminde kullanılır. Sağlık ve güvenlik ile ilgili konularda, acil hizmet araçlarının olay yerine gönderilmesi sırasında Coğrafi Bilgi Sistemlerinin kullanılması büyük yarar sağlamaktadır. Bunun yanı sıra, taksilerin ve tehlikeli madde taşıyan araçların güzergâhlarının belirlenmesinde de CBS uygulamaları kullanılabilmektedir.
- Tesis Yerlerinin Belirlenmesi: İtfaiye, hastane, sağlık ocağı, karakol, fabrika, alışveriş merkezi ve tehlikeli atık depoları gibi tesisler için en uygun yerin saptanmasında kullanılmaktadır.
- Yeraltı ve Yerüstü Kaynaklarının Değerlendirilmesi: Coğrafi Bilgi Sistemleri, doğal kaynakların tespiti, korunması ve verimli şekilde kullanımı için gereken değerlendirmelerin yapılmasına olanak tanır.
- Veri Madenciliği Uygulamaları: Tekrar eden olayları kaydedip, analiz ederek, durumu anlamak ve çözüm üretmek amacıyla kullanılır. Seçim veya reklâm kampanyası sonuçlarının izlenmesi, suç ve trafik kazası oranlarının takip edilmesi gibi alanlarda kullanılabilmektedir [12].

2.1.1. Coğrafi Bilgi Sistemlerinde Kullanılan Veri Tipleri ve Özellikleri

Coğrafik veriler, enlem-boylam şeklindeki coğrafi koordinatlardan veya adres, şehir adı, yol adı gibi referans verilerden oluşur. Bu coğrafik referanslar sayesinde nesneler, koordinatı bilinen uygun bir pozisyona yerleştirilebilirler. Böylece ticari merkezler, parseller, ormanlık alanlar ve yüzey şekillerinin analizleri, coğrafi konumlarına bağlı olarak yapılabilir. Coğrafi konum bilgisi, seçilecek veri modeline uygun şekilde ifade edilir. Coğrafi Bilgi Sistemleri, verilerini saklamak için temel olarak iki farklı model kullanırlar. Bunlar; ‘vektör’ modeli ve ‘raster’ modelidir [13].

2.1.1.1. Vektör Veriler

Nokta, çizgi ve poligon türlerinden oluşur ve sınırların tanımlanması mantığına dayanır.

- **Nokta veriler:** Elektrik direkleri, duraklar, hastaneler, okullar, spor salonları, alışveriş merkezleri gibi tek bir nesneyi belirten veriler, haritalar üzerinde tek bir nokta ile ifade edilebilmektedirler.
- **Çizgi veriler:** Elektrik hatları, telefon hatları, su ve kanalizasyon şebekeleri, yollar, akarsular gibi birçok farklı noktanın birleşmesi ile oluşan verilerdir.
- **Poligon veya alan veriler:** Şehirler, denizler, göller, ormanlık araziler gibi noktaların birleşmesi ile oluşan, çizgi verilerinden farklı olarak, belirli bir noktadan başlayıp tekrar aynı noktada son bulan verilerdir [14].

Vektör modelinde noktalar, çizgiler ve poligonlara ait coğrafik veriler, x ve y koordinatları belirtilerek saklanır. Sondaj çukuru gibi bir nokta verisinin coğrafik yeri, tek bir (x,y) koordinatıyla tanımlanabilir. Yollar ve akarsular gibi çizgisel veriler, (x,y) koordinatlarından oluşan noktalar dizisi şeklinde tanımlanır. Pazarlama ve satış bölgesi veya ülke sınırları gibi poligon verileri ise, (x,y) koordinatlarından oluşan ve başlangıç ve bitiş noktaları aynı olan noktalar dizisi şeklinde saklanabilir [13].

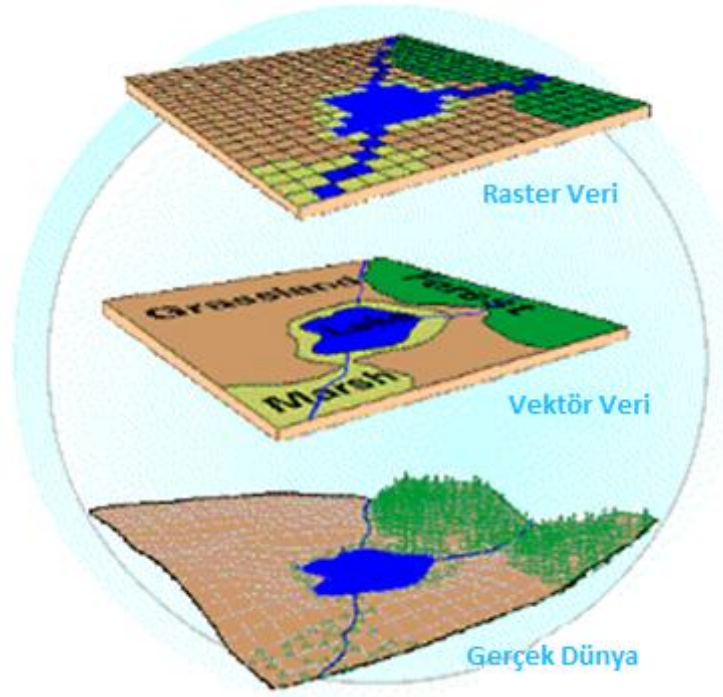
Vektör verileri, veri tabanlarında birbirleriyle ilişkilendirilmiş halde saklanırlar. Herhangi bir vektörün sağında ve solunda bulunan diğer vektör verileri de veri tabanında saklıdır. Vektör verilerini birbiriyle konumsal olarak ilişkilendirme ve birleştirme işlemlerine topoloji denilmektedir. Topoloji, kesişen çizgileri ya da üst üste binmiş noktaları veya poligonları tanıyarak, vektör verilerini analiz etme konusunda Coğrafi Bilgi Sistemi uygulamalarına yardımcı olmaktadır [15].

Vektör modeli, sabit özellikteki verileri tanımlarken kullanışlı olmasına rağmen, herhangi bir yere ulaşım masrafları gibi sürekli değişen değerler alan verileri tanımlarken pek kullanışlı olmamaktadır.

2.1.1.2. Raster Veriler

Raster modeli, vektör modelinin aksine sürekli özellikteki verileri modellemeye tercih edilir. Raster veriler, ızgara biçiminde hücrelerde saklanır [13]. Öncelikle görüntüler küçük parçalara ayrılır. Grid adı verilen ızgara şeklindeki her bir hücrede, bölgeye ait öznetelik

verilerinin o hücreye düşen değeri gösterilir. Bu hücrelerin her birinde yalnızca bir değer saklanır. Örneğin; yolları belirten bir raster veri dizininde, hücrenin aldığı 4 değeri o yolun bir otoyol olduğunu gösterebilir. Bir yolu belirten hücre sayısı yolun uzunluğu ile orantılı olabilmektedir. Fakat her bir hücrenin büyüklüğü birkaç metreyi ifade edebileceği gibi, birkaç kilometreyi de ifade edebilmektedir. Buna hücrenin çözünürlüğü denilmektedir. Aynı bölgeyi temsil etmede, çözünürlüğü yüksek olan haritalar düşük olanlara göre daha çok hücreye ihtiyaç duyarlar. Uydu görüntüleri, raster modeli kullanılarak analiz edilebilir.



Şekil 2.2. Gerçek dünya verilerinin vektör ve raster veriye dönüştürülmesi [16].

2.1.1.3. Vektör ve Raster Veriler Arasındaki Farklar

- Vektör veri, raster veriye göre daha hassastır.
- Raster modelde, boş hücreler de veri olarak saklanmaktadır. Bu nedenle raster veri, vektör veriye göre daha çok yer tutar.
- Raster modelde sayısal dosyanın büyüklüğü (byte cinsinden) harita yoğunluğu ile ilgili değildir. Haritanın kâğıt boyutuna bağlıdır. Vektör modelde ise, sadece koordinat bilgileri saklandığı için, raster verinin tersine sayısal dosyanın büyüklüğü (byte cinsinden) harita yoğunluğuna bağlıdır [17].

2.2. Coğrafi İlişkisel Veri Tabanları (The Geo-Relational Database)

Veri tabanları, birbiriyle ilişkili verileri tekrara yer vermeden, çok amaçlı kullanıma olanak sağlayacak şekilde depolayan yazılımlardır [18].

Veri tabanları; Hiyerarşik, İlişkisel, Ağ ve Nesne Yönelimli Veri Tabanları olmak üzere dört sınıfa ayrılır. Hiyerarşik Veri Tabanları, ağaç yapısına sahiptirler. Tabloları arasında child-parent ilişkisi bulunur. İlişkisel Veri Tabanları, tabloları arasında hiyerarşi bulunmayan, herhangi bir kısıtlama bulunmayan ve her tablonun birbiriyle ilişki içinde olabileceği veri tabanlarıdır. Ağ Veri Tabanları, hiyerarşik modelden gelişmiştir. Hiyerarşik Veri Tabanlarından farklı olarak, child'ların birden fazla atası olmasına müsaade ederler. Nesne Yönelimli Veri Tabanları ise, üç boyutlu kompleks veri tabanı tasarımına sahiptirler. İlişkisel Veri Tabanlarından farklı olarak, türlerin kullanılmasına gerek yoktur. Verileri tek parça halinde getirdikleri için, birden çok veri sorgulaması için performansları iyi değildir [19] .

Coğrafi Bilgi Sistemlerinde, veri tabanları çok önemli bir yere sahiptir. Coğrafi veriler, görüntü dosyalarından oluştuğu için çok büyük boyutludur. Coğrafi ilişkisel veri tabanları, Coğrafi Bilgi Sistemlerinin içerdiği mekânsal verileri saklamada kolaylık sağlamak amacıyla geliştirilmiştir. Aynı zamanda, Mekânsal Karar Destek Sistemi geliştirmenin de anahtar problemlerinden biri, heterojen veriyi (mekânsal ve mekânsal olmayan veriyi) ideal yollarla organize etmektir. Mekânsal veriler için destek sağlayan Coğrafi İlişkisel Veri Tabanları bu problemi kolayca çözmeye olanağı tanımaktadır [20].

Coğrafi ilişkisel veri tabanları, verileri coğrafik ilişkilerine göre tutarlar, yalnız bu durum ölçeklenebilirliklerini kısıtlamaktadır. Bu veri tabanlarındaki ikili veri yapısı, yedekleme ve yineleme gibi özelliklerinden tamamıyla yararlanılabilmesini engellemektedir. Buna rağmen coğrafi ilişkisel veri tabanlarının büyük boyutlu verileri desteklemeleri performanslarını dengelemektedir. Ayrıca coğrafi veriler için özel tanımlamalara imkân tanırırlar. Bu şekilde, tanımlanan verilerin karmaşık yapıları gizli tutulur. Örneğin; PostgreSQL veri tabanında geometry tipinde tanımlanan bir veri, çizgiye ya da poligona ait satırlarca süren (x,y) koordinat bilgilerinden oluşabilir. Bu coğrafi bilgiler, veri tabanında kodlanarak tutulur. Böylece karmaşık yapısı saklanmış olur [21].

MySQL ve PostgreSQL coğrafi ilişkisel veri tabanlarındandır.

2.2.1. MySQL Veri Tabanı

MySQL, açık kaynak kodlu, hızlı, güvenilir ve kullanımı kolay bir veri tabanı yönetim sistemidir.

- Çok kullanıcıdır ve çoklu iş parçacıkları bir arada yürütülür.
- Windows, UNIX ve OS/2 platformlarında kullanılabilir. Fakat Linux altında daha hızlı çalışabilmektedir.
- Ücretsiz dağıtılmakla birlikte, ticari lisans isteyenlere yönelik ücretli bir seçeneği de bulunmaktadır.
- Kaynak kodu açık olduğu gibi, ikilik kod halinde indirilebilen sürümleri de vardır.
- 7 milyon kayıt içeren, 10.000 tablodan oluşan, yaklaşık 100 GB kadar verinin MySQL veri tabanında tutulabildiği bilinmektedir. Bu özelliği sayesinde Coğrafi Bilgi Sistemleri yazılımları, mekânsal verilerini tutmak için rahatlıkla MySQL veri tabanını kullanabilmektedirler.
- Altı milyondan fazla sistemde kuruludur. Web sunucularında en çok kullanılan veri tabanı olduğu bilinmektedir.
- ODBC sürücülerinin bulunması, birçok kod geliştirme platformuyla birlikte rahatlıkla kullanılabilmesini sağlamaktadır. Ayrıca birçok web programlama dili ile birlikte kullanılabilir.
- Çok kullanıcı bir veri tabanı yönetim sistemi olmasından kaynaklanabilecek erişim problemlerini önlemeyi başarmıştır. Erişim için gerekli yetkilendirmeleri ve kısıtlamaları en sorunsuz şekilde sağlamaktadır.
- Transaction ve alt sorgu desteği bulunmaktadır.
- Ayrıca pek çok veri tabanı programcısı tarafından zorlayıcı olarak görülen veri tabanı yönetim sistemlerinin sunmuş olduğu veri tutarlılığını sağlama işini programcıya bırakmıştır.
- En önemlisi; coğrafi verileri tutmak için özel veri tiplerini içermektedir. Geometry tipiyle POINT, LINE, POLYGON gibi geometrik şekilleri ifade etmek için kullanılacak enlem, boylam (x,y) bilgilerini tutma imkânı sağlar.

2.2.1.1. MySQL Tablo Tipleri

MySQL’ de transaction tabloları ve atomik işlem tabloları olmak üzere iki farklı tip tablo yapısı bulunmaktadır. Transaction tabloları; InnoDB, BerkeleyDB’dir. Atomik işlem tabloları ise; MyISAM, HEAP, MERGE, ISAM’dır.

Her tablo için farklı tablo tipi seçilebildiği gibi, bir veri tabanı içinde iki ya da daha fazla tablo tipini bir arada kullanmak da mümkündür [22].

2.2.1.2. MySQL Control Center

MySQL sunucusunun gerçekleştirdiği tüm işlemler, MySQL Control Center yardımıyla yapılmaktadır. Örneğin; yeni veri tabanı oluşturma, var olan veri tabanlarına ait tabloların özellikleri düzenleyip, içeriklerini güncelleme, veri tabanlarını onarma, kullanıcılar için hesap ekleme gibi birçok işlemi yürütmektedir.

MySQL veri tabanında coğrafi veriler içeren bir tablo oluşturmak için kullanılabilecek örnek bir sorgu aşağıda gösterilmektedir.

```
CREATE TABLE `depremverileri`(  
  `geom` geometry default NULL,  
  `MM` text,  
  `mag` float,  
  `ddate` datetime default '0000-00-00 00:00:00',  
  `ref` text  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Coğrafi ilişkisel veri tabanlarında, mekânsal veriler farklı bir kodlama kullanılarak saklanır. Bu nedenle veri tabanına kaydedilmiş bir mekânsal verinin koordinat (enlem ve boylam) değerlerine sorgulamayla tekrar ulaşmak için ‘astext’ fonksiyonu kullanılır.

Coğrafi veri eklerken kullanılan geometry tipine ait nokta, çizgi ve poligon bilgileri aşağıda gösterilen formatlara uygun olmalıdır.

```
POINT (43.5 38.75)  
LINE (43.5 38.75, 44.3 38.98, 44.85 39.79)  
POLYGON (43.0 38.0, 44.3 38.98, 44.85 39.79, 43.0 38.0)
```

MySQL' in coğrafik ilişkisel bir veri tabanı olarak kullanımına ilişkin birkaç örnek coğrafi sorgulama aşağıda gösterilmektedir.

- Geometry adındaki tabloya coord isiminde POINT tipinde bir kolon ekleme ve ona mekânsal indeks (spatial index) oluşturma

```
ALTER TABLE GEOMETRY ADD coord POINT NOT NULL;  
CREATE SPATIAL INDEX coord ON GEOMETRY (coord);
```

- Geometry adındaki tablonun coord isimli sütununa Point tipinde veri ekleme

```
INSERT INTO GEOMETRY (coord) VALUES( GeomFromText( 'POINT(40 -100)' ));  
INSERT INTO GEOMETRY (coord) VALUES( GeomFromText( 'POINT(1 1)' ));
```

- Geometry tablosuna ait coord isimli sütunundaki verilerin x,y koordinatlarını alma

```
SELECT X(coord), Y(coord) FROM GEOMETRY
```

- Bir poligonla (dikdörtgenle) sınırlanmış bir alanın içinde kalan noktaları alma

```
SELECT MBRContains(  
GeomFromText('POLYGON((0 0,0 3,3 3,3 0,0 0))' ), coord)  
FROM GEOMETRY [23].
```

2.2.2. PostgreSQL Veri Tabanı

PostgreSQL de, MySQL gibi coğrafik ilişkisel bir veri tabanı yönetim sistemidir.

- PostgreSQL'in performansı yüksektir, güvenlidir ve geniş özelliklere sahiptir.
- UNIX ve UNIX benzeri işletim sistemlerinde çalışabilmesinin yanı sıra, NT çekirdekli tüm Windows sistemlerde de çalışabilir.
- PostgreSQL ücretsiz olarak indirilebilir ve kullanılabilir.
- Açık kaynak kodludur.
- Ticari ya da diğer açık kaynak kodlu veri tabanlarının sağladıkları özellikleri içerir.

- PostgreSQL'in güvenilirliği ispatlanmıştır. Geniş kullanıcı kitlesine sahip olması ve açık kaynak koduna herkes tarafından rahatlıkla erişilebilmesi nedeniyle, oluşabilecek hatalar çok kısa sürede kapatılabilmektedir. Beta sürümleri en az 1 ay kadar test edilmektedir.
- PostgreSQL'in performansı her yeni sürümle birlikte artmaktadır [24].
- Kaynak kodu açık olduğu gibi, ikilik kod halinde indirilebilen sürümleri de vardır.
- PostgreSQL'in de MySQL gibi ODBC sürücülere bulunmaktadır. Bunlar; PsqLODBC ve OpenLink ODBC'dir. ODBC sürücülerinin bulunması, birçok kod geliştirme platformuyla birlikte rahatlıkla kullanılabilmesini sağlamaktadır. Ayrıca birçok web programlama dili ile birlikte kullanılabilir.
- Çeşitli grafik arabirimlerine sahiptir.
- Standart SQL sorgu dilini destekler. SQL'in alt kümesi olan PostQuel dilini kullanır.
- POSTGRES'in zengin veri tiplerini ve güçlü veri modelini içeren bir yapıda sahiptir.
- Sınırsız veri tabanı büyüklüğüne olanak tanımaktadır (32 TB'lık veri tabanı bulunmaktadır) [25].
- En önemlisi ise; coğrafik verileri tutmak için özel veri tiplerini içermektedir. Geometry tipiyle nokta, çizgi, poligon gibi geometrik şekilleri ifade etmek için kullanılacak enlem, boylam bilgilerini tutma imkânı sağlar.

PostgreSQL veri tabanının içerdiği özellikler:

- Transactions
- Subselections
- Views
- Foreign key referential integrity
- Sophisticated Locking
- User-defined Types (Kullanıcı Tanımlı Tipler)
- Rules
- Inheritance (Kalıtım)
- Multi-version concurrency control
- HOT
- Point-In-Time-Recovery
- Warm Standby [24].

2.2.3. PostGIS (GeoDatabase)

PostgreSQL tarafından üretilen ve bu veri tabanı yönetim sistemine, coğrafi veri tabanı yeteneklerini kazandıran bir eklentidir. PostGIS, diğer açık kaynak kodlu Coğrafi Bilgi Sistemi araçlarına da standart mekânsal veri tabanı desteği sağlar [26]. Coğrafi Bilgi Sistemi yazılımları geliştirirken, mekânsal verileri tutmada rahatlık sağlayacak veri tabanı özelliklerini içinde barındırır. ESRI SDE ve Oracle Spatial ile aynı işi görebilecek bir yapıya sahiptir. OpenGIS [27] tarafından gelişimi sağlanmaktadır. Açık kaynak kodlu bir veri tabanı eklentisi olduğu için, birçok açık kaynak kodlu Coğrafi Bilgi Sistemi araç sağlayıcı PostGIS'i kullanmayı tercih etmektedir [21].

PostGIS, 'hafif geometri' ve 'indeks optimizasyonu' üzerine kurulu bir sistemdir. Bu sayede esas hafıza ve disk kullanımını düşürmeyi hedeflemektedir. Hafif geometri kullanımı ile sunucuların diskten RAM'e daha fazla veri aktarımı sağlanır. Bu da sorgu performanslarını oldukça artırır.

PostGIS'in ilk versiyonu 2001 yılında yayınlanmış olup GNU Genel Lisansına sahiptir. 2005 yılında ise ilk kararlı sürümü kullanıma sunulmuştur. 2006 yılında "Open Geospatial Consortium (OGC)" tarafından PostGIS'in SQL veri tabanı için temel özellikleri kabul edilmiştir [28].

2.2.3.1. PostGIS'in Özellikleri

- Hem mekânsal verileri hem de niteliksel verileri tutabilme özelliğine sahiptir.
- POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION gibi coğrafi veri tiplerini destekler.
- Uzunluk, çevre, alan gibi coğrafi ölçüleri belirlemek için mekânsal (spatial) verileri destekler.
- AsBinary() komutu ile binary verileri indeksler.
- AsText() komutu ile text verilerini indeksler.
- Veri analizi yapmak için çeşitli fonksiyonları içerir. Bunlardan bazıları; area(), length(), distance(), transform() 'dur.
- union(), difference(), symmetric difference() ve buffers() gibi GEOS tarafından sağlanan mekânsal operatörleri içermektedir.
- Vektörel verilerin yanı sıra raster verilerin saklanmasına da olanak tanımaktadır.

- Karmaşık SQL sorgularını çalıştırabilmenin yanında, yüksek hız ve yüksek performans ile sorgu çalıştırabilme özelliklerine sahiptir.
- center(geom) komutu ile geometry tipinde tanımlanmış POINT, LINESTRING ya da POLYGON şeklindeki verilerin merkez noktası bulunarak, odaklanma işlemlerinde kolaylık sağlanmış olur.

Birçok yazılım ürünü PostGIS'i veri tabanı arka planı olarak kullanmaktadır. Bunlar;

- **GeoServer** (www.geoserver.org)
- Feature Manipulation Engine (www.safe.com/fme/fme-technology)
- GRASS GIS (GPL) (<http://grass.fbk.eu>)
- Interoperability Extension from ESRI (www.esri.com/software/arcgis/extensions/datainteroperability)
- Manifold System (www.manifold.net)
- Mapnik (LGPL) (www.mapnik.org)
- MapDotNet Server (www.mapdotnet.com)
- MapServer (BSD) (www.mapserver.org)
- MapGuide (LGPL) (<http://mapguide.osgeo.org>)
- OpenJUMP (GPL) (www.openjump.org)
- Quantum GIS (GPL) (www.openjump.org)
- TerraLib (LGPL) (www.terralib.org)
- ArcGIS (ZigGIS, ArcSDE) (www.arcgis.com) [28].

Geometrik veri tipleri ile enlem ve boylam bilgileri tutulmaktadır. Coğrafi veri eklerken kullanılan geometry tipine ait nokta, çizgi, poligon, çoklu nokta, çoklu çizgi, çoklu poligon ve karışık geometrik şekil içeren bilgiler aşağıda gösterilen formatlara uygun olarak tanımlanmalıdır [29].

```
POINT(0 0)
LINESTRING(0 0,1 1,1 2)
POLYGON(0 0,4 0,4 4,0 4,0 0)
MULTIPOINT(0 0,1 2)
MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0)),((1 1,2 1,2 2,1 2,1 1)),((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
```

```
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))
```

PostgreSQL'in coğrafik ilişkisel bir veri tabanı olarak kullanımına ilişkin birkaç örnek coğrafi sorgulama aşağıda gösterilmektedir.

- Geometri kolonu içeren bir tablo oluşturma

```
CREATE TABLE binalar
( id character varying,
  sınıf character varying(50),
  geom geometry,
PRIMARY KEY(id));
```

- Var olan bir tabloya geom adında bir coğrafi veri kolonu ekleme

```
ALTER TABLE yollar ADD COLUMN geom GEOMETRY;
```

- yollar tablosuna ait geom sütununu belirtilen enlem, boylam değerlerine göre güncelleme

```
UPDATE yollar SET geom = makepoint(longitude, latitude);
```

- koyler isimli veri tabanına coğrafi ve coğrafi olmayan verilerden oluşan bir satır ekleme

```
INSERT INTO koyler VALUES('1','Yeniköy',GeomFromText('POINT(39.3
38.8)'));
```

- Kullanılmak istenen koordinat sistemini belirtmek için, 4326 değeri veri kayıt sorgusuna eklenebilir. (EPSG 4326 olarak kullanılan SRS, WGS84 koordinat sistemine karşılık gelmektedir.)

```
INSERT INTO köyler VALUES ('1','Yeniköy',GeomFromText('POINT(39.3
38.8)',4326));
```

- yollar isimli tabloya linestring şeklinde tanımlı geometrik veri içeren bir satır ekleme

```
INSERT INTO yollar VALUES ('1',GeomFromText('LINESTRING(39 38,38
37)',4326));
```

- Binalar tablosuna poligon olarak belirtilen alan verisini ekleme

```
INSERT INTO binalar VALUES ('1', 'SporSalonu',
GeomFromText('POLYGON((39.2 38.6,38.9 38.3,39.0 38.4,39.1 38.5,39.2
38.6))',4326));
```

Coğrafik ilişkisel bir veri tabanı olan PostgreSQL'in bir eklentisi olan PostGIS'in coğrafik sorgulamalarda kullandığı özel fonksiyonlar aşağıda gösterilmektedir. ('Geometry' tipi; bir noktayı, çizgiyi, poligonu, iki ya da üç boyutlu bir şekli içermektedir.)

Fonksiyonun Prototipi: Fonksiyon Adı (Parametreleri) : Geri Döndürdüğü Verinin Tipi

- Distance(geometry, geometry) : number (İki geometrik veri arasındaki uzaklığı verir.)
- Equals(geometry, geometry) : boolean (İki geometrik veri birbirine eşitse, True döndürür.)
- Disjoint(geometry, geometry) : boolean (İki geometrik veri birbirinden ayrıkça (kesişmiyorsa), True döndürür.)
- Intersects(geometry, geometry) : boolean (İki geometrik veri birbirinden ayrıkça (kesişmiyorsa), False döndürür.)
- Touches(geometry, geometry) : boolean (İki geometrik veri için en az bir nokta ortakça, fakat bu iki geometrik veri kesişmiyorsa, True döndürür.)
- Crosses(geometry, geometry) : boolean (İki geometrik verinin bazı (tümü değil) iç noktaları ortakça, True döndürür.)
- Overlaps(geometry, geometry) : boolean (İki geometrik veri kesişiyorsa, fakat biri diğerinin tamamını içermiyorsa, True döndürür.)
- Contains(geometry, geometry) : boolean (İki geometrik veriden ikincisi, ilkinin tamamen içinde ise, yani ilki ikincisini kapsıyorsa, True döndürür.)
- Within(geometry, geometry) : boolean (İki geometrik veriden ilki, ikincisinin tamamen içinde ise, yani ikinci veri ilkinin kapsıyorsa, True döndürür.)
- Length(geometry) : number (Bir geometrik verinin uzunluğunu verir.)

- Area(geometry) : number (Bir geometrik verinin (polygon, multipolygon gibi) alanını verir.)
- Centroid(geometry) : geometry (Bir geometrik verinin merkez koordinatlarını verir.)
- Boundary(geometry) : geometry (Bir geometrik verinin sınır noktalarının kombinasyonunu verir.) [30-31].

2.3. Emlak Değer Haritası Uygulamasında PostgreSQL ve PostGIS Kullanımı

Emlak Değer Haritası uygulaması için coğrafi ilişkisel bir veri tabanı olan PostgreSQL veri tabanı ve PostGIS eklentisi kullanılmıştır. PostGIS, PostgreSQL'e coğrafi özellikli verileri saklama imkânı sağlayan bir eklenti olup, PostgreSQL'de bir veritabanı oluşturulurken, template (şablon) özelliği template_postgis olarak seçildiğinde, coğrafi veri özelliklerini saklayan iki tablo (geometry_columns ve spatial_ref_sys) otomatik olarak oluşturulur. Bu özellik kullanılarak coğrafi nitelik taşıyan bir emlak veri tabanı oluşturulup, mahalleler, parseller, binalar ve yollar için veri tabanına data girişi yapılmıştır. Mahalleler, parseller ve binalar geometry tipinde POLYGON olarak tanımlanırken, yollar LINESTRING olarak tanımlanmıştır.

Ayrıca emlak veri tabanında, faktor_puani ve faktor_agirligi isminde iki önemli tablo oluşturulmuştur. Bu tablolardan faktor_puani, her bir bina için, emlak değerine etki edecek olan faktör puanlarının hesaplatılıp, kaydedilmesi amacıyla kullanılmıştır. faktor_agirligi tablosu ise, emlak değerine etki edecek olan bu faktörlerin etki etme yüzdelerine göre belirlenen faktör ağırlıklarını içermektedir.

Sonraki aşamalarda, yeni elde edilen veriler doğrultusunda noktalar adında bir tablo daha oluşturulmuş olup, bu tablo ile ulasim_imkan_yakinlik faktör puanını hesaplamada POINT şeklindeki verilerden de istifade edilmiştir.

Emlak değerine etki eden faktörlerin, etki etme yüzdelerini ifade eden faktör ağırlıkları, değeri bilinen 100 adet emlağın eğitim verisi olarak kullanılmasıyla Harmoni Arama (Harmony Search) Algoritması ile MATLAB' da hesaplanmıştır. Elde edilen en optimum faktör ağırlığı grubu seçilerek, veri tabanında kayıtlı tüm binaların emlak değerlerinin hesaplanmasında kullanılmıştır. Emlak değerleri, ilgili bina ile eşleşecek şekilde emlak_degeri sütununa kaydedilmiştir.

3. OGC STANDARTLARI VE GEOSERVER

3.1. OGC Standartları

Coğrafi Bilgi Sistemlerinde ve harita sunucularında ortak bir veri format belirlemek amacıyla birçok kullanıcının, kurum ve kuruluşun üye olduğu standart örgütüne ihtiyaç duyulmuştur. Bu amaçla OGC (Open Geospatial Consortium) standartları geliştirilmiştir [32]. OGC, hem OpenGIS çalışma çevresinin özet tanımlayıcılarını hem de veri modellerinin teknik uygulama detaylarını geliştirmeye ilgilenmektedir [33].

3.1.1. OpenGIS Standartları ve Özellikleri

Son yıllarda, veri ve uygulama katmanları üzerindeki birlikte çalışabilirlik sorununu çözmek amacıyla çeşitli organizasyonlar tarafından standartlar geliştirilmektedir. Standart gövdesi; coğrafi bilgi ve servislerin, tarafsızlığını ve her ağ, uygulama ya da platform karşısındaki uyumluluğunu sağlamayı hedefler [33].

OpenGIS Standartları ve Özellikleri, detaylı arabirim ve kodlamalardan oluşan teknik belgelerdir. Yazılım geliştiriciler bu dokümanları, ürün ve hizmetler içindeki arabirim ve kodlamalara destek oluşturmak için kullanırlar. Bu özellikler OGC'nin temel ürünüdür ve birlikte çalışabilirliğe dayalı sorunları ele almak için, üyeleri tarafından geliştirilmektedir. İdeal olarak, özellikler iki farklı yazılım mühendisinin çalışmalarına uygulanırken; ayrıca bir debug işlemi olmaksızın ortaya çıkan parçalar birleştirilir ve çalıştırılır.

Belgeleri herkes ücretsiz olarak kullanabilir.

3.1.1.1. OpenGIS Standartları

Uygulama standartları, özet modellerden farklıdır. Daha çok, teknik hedef kitlesi için yazılır ve yazılım bileşenleriyle ara yüz yapısı arasında detaylandırılır. Her biri, birbirinden habersiz, iki farklı yazılım mühendisi tarafından uygulandığında, sonuç bileşenlerinin her biri diğerine eklenip, birlikte çalıştırılırlarsa; bu ara yüz tanımlaması, uygulamalarda ayrıntı seviyesi olarak düşünülür.

3.1.1.2. Özet Modeller

OGC Teknik Komitesi, verilerin birlikte kullanılabilirliğine ve geometrik ve mekânsal veri (geospatial) teknolojisinin vizyonuna destek olarak, OpenGIS olarak isimlendirilen bir mimari geliştirmektedirler. Bu özet model en çok, OGC özelliklerinin geliştirilmesi için kavramsal bir temel oluşturur. Açık arayüzler ve protokoller yapılandırılırken özet modele başvurulur, böylece mekânsal veri işleme sistemlerinin farklı çeşit ve farklı markaları arasında birlikte çalışabilirlik sağlanır. Bu özet model, OpenGIS Uygulama Modelini geliştirmek için bir referans model sağlar.

3.1.1.3. OpenGIS Referans Modeli (ORM)

OpenGIS Referans Modeli (ORM), OGC'nin devam eden çalışmaları için bir çerçeve sağlar. ORM, OpenGIS tanımlama belgeleri arasındaki ilişkilere odaklanarak, OGC Standartlarının temelini oluşturur. OGC Standartının Temelleri, OGC Best Practices belgeleri gibi, onaylanmış OpenGIS Özet ve Uygulama Modellerinden oluşur. Best Practices belgeleri, OGC üyelerinden yetkili pozisyondakilere, çoğunlukla benimsenen modeller için sağlanan teknik bilgiler olarak verilir [34].

3.2. GML (Geographic Markup Language)

GML, mekânsal ve mekânsal olmayan verileri toplayan, depolayan, işleyen ve bu tür verileri içeren veri tabanlarının internet üzerinden birbirleriyle bilgi alışverişinde bulunmalarını kolaylaştıran XML (Extensible Markup Language) tabanlı bir işaretleme dilidir. Diğer bir deyişle, coğrafi bilgilerin mekânsal veya mekânsal olmayan özelliklerinin depolanması ve aktarımı için kullanılan bir çeşit kodlamadır [1].

GML' i daha iyi kavrayabilmek için kısaca XML'e bakmak gerekirse: XML, veri paylaşımında ortak bir dil sağlamak amacıyla geliştirilmiş bir standarttır. XML sayesinde web bütünleşmesi mümkün hale gelmiştir. Ayrıca, hem insanlar tarafından hem de bilgisayar sistemleri tarafından rahatça okunabilecek dokümanlar oluşturulması sağlanmıştır. Veri depolama ve veri alışverişini sağlayan bir ara formattır [35]. GML ise, XML' in Coğrafi Bilgi Sistemlerinde kullanılmak üzere düzenlenmiş şeklidir.

Yazılım geliştiriciler harita ya da metinden oluşan coğrafi verileri, GML formatına dönüştürüp, internet üzerinden veri aktarımını gerçekleştirebilirler. Aktarım tamamlandıktan sonra, coğrafi verileri tekrar diğer depolama formatlarına dönüştürebilirler [1].

3.3. WMS (Web Map Service)

İnternet ortamında harita yayınlanmasını sağlayan bir sunucudur. Çok yaygın kullanıma sahiptir. Bir URL yoluyla sorgu tanımlamalarını alır ve girilmiş olan tanımlamalara göre ürettiği haritayı istenilen resim formatında geri döndürür. Gelen harita resim formatında olduğu için, herhangi bir vektörel özelliğe sahip değildir. Bu sebeple yalnızca basit harita gösterimlerinde kullanılabilir.

URL’ de kullanılan sorgu, OGC Standartlarına dayanır. Bu şekilde OGC WMS standardına dayanan bir ürün, her yerde aynı sorgu yapısına sahip olur. Yalnız ürünün OGC WMS standartlarının hangi sürümüyle uyumlu olduğuna dikkat edilmelidir. Aşağıda WMS sorgusu için örnek bir URL ve sorguya karşılık harita servisten dönen sonuç görülmektedir.

URL=http://hostname/WMS?LAYERS=TurkiyeRaster&FORMAT=image%2Fjpeg&SR
S=epsg%3A4326&SERVICE=WMS&VERSION

Sorgu sonucu=1.1.1 & REQUEST = GetMap & BBOX = 86793, 475793, 547593,
936593 & WIDTH = 256 & HEIGHT = 256

Sorgu tanımlaması koyu renkle belirtilenlerden oluşur. Öncesinde kullanılan hostname ve WMS sözcüğü yayınlanmakta olan URL’ inin adresini belirtir.

Layers: Bir WMS üzerinde birden fazla harita katmanı bulunabilir. ‘Layers’ görüntülenmek istenen harita katmanlarının seçilmesinde kullanılır. Görüntülenmek istenen harita katmanlarının projeksiyon ve datum bilgilerinin aynı olmasına dikkat edilmelidir.

‘Layers’, WMS için en kritik bölümdür. Bir harita, bir esas katman ve üstünde birçok ek katmandan oluşabilir. Her harita <Layer> bileşeniyle/bileşenleriyle ifade edilir. Katmanlar, XML’de hiyerarşik şekilde iç içe bulunmaktadır. Esas katmanda tanımlanmış olan bazı özellikler, alt katmanlardan türetilmiştir. Türetilen bu özellikler ya alt katmanlarda yeniden tanımlanır ya da katmana olduğu gibi eklenir. Esas katman, alt katmanlarından birinin, birkaçının ya da hepsinin birden istenmesine izin verir.

Her katmanda en az bir <SRS> elemanı bulunmalıdır. Ana katman (baselayer), tüm yardımcı katmanlarda ortak olan SRS'lerin listesini barındırır.

Her katman bir <LatLonBoundingBox>' a sahip olmalıdır. <LatLonBoundingBox>, bir poligonu çevreleyen enlem ve boylamların ondalıklı gösterimidir. SRS'nin harita servisini destekleyip desteklemediğine bakılmaksızın LatLonBoundingBox sağlanmalıdır. Bunun amacı arama motoruna koordinat transferi yapmadan coğrafi aramaları kolaylaştırmaktır. BoundingBox ise, haritanın genişlik ve uzunluğunu belirtir.

Ayrıca görüntülenmek istenen haritaları, isteğe bağlı şekilde düzenlemek için <style> tanımlanmıştır.

Format: Map servisin istenilen haritayı hangi formatta döndüreceğini belirtmek için kullanılır. Alacağı değerler kullanılan harita sunucusuna göre değişir. En fazla PNG ve JPEG resim formatları tercih edilir.

SRS: SRS (Spatial Reference System), mekânsal referans sistemidir. Haritanın projeksiyon ve datum bilgilerini gösterir. En yaygın kullanımı, EPSG standardına dayanan kodlamalarıdır. EPSG standardına dayanan SRS kodları, birçok projeksiyon ve koordinat sistemine ait tanımlamaları içerir. Bu standartlar EPSG (European Petroleum Survey Group) tarafından belirlenmiştir. EPSG: 4326 olarak tanımlanan SRS, WGS84 koordinat sistemine karşılık gelir. Kullanılmak istenilen koordinat sistemine uygun SRS kodu, Spatial Reference sitesinden öğrenilebilir. İstenilen haritanın koordinat sistemi yazılımdan farklıysa (harita sunucusuna bağlı olarak) hata mesajı ya da reprojeeksiyona uğramış bir harita ekrana gelir.

Service: İstenilen servisin tipini belirtir. OGC tarafından tanımlanan en temel servisler WMS, WFS ve WCS 'dir. Bu servisler arasından istenileni seçmek için 'Service' tanımlaması kullanılır.

Version: Talep edilen servisin sürüm bilgisini gösterir. WMS kullanıcıları, sorgu tanımlamalarını kullandıkları sürüme göre şekillendirmek zorundadırlar.

Request: Talep edilen servisin kullanılmak istenilen fonksiyonunu gösterir. WMS' nin GetMap, GetCapabilities, GetFeatureInfo fonksiyonları bulunmaktadır. Request çağrısı ile talep edilen bir harita ise, GetMap fonksiyonu çağırılmalıdır. Bunun yanında, GetCapabilities

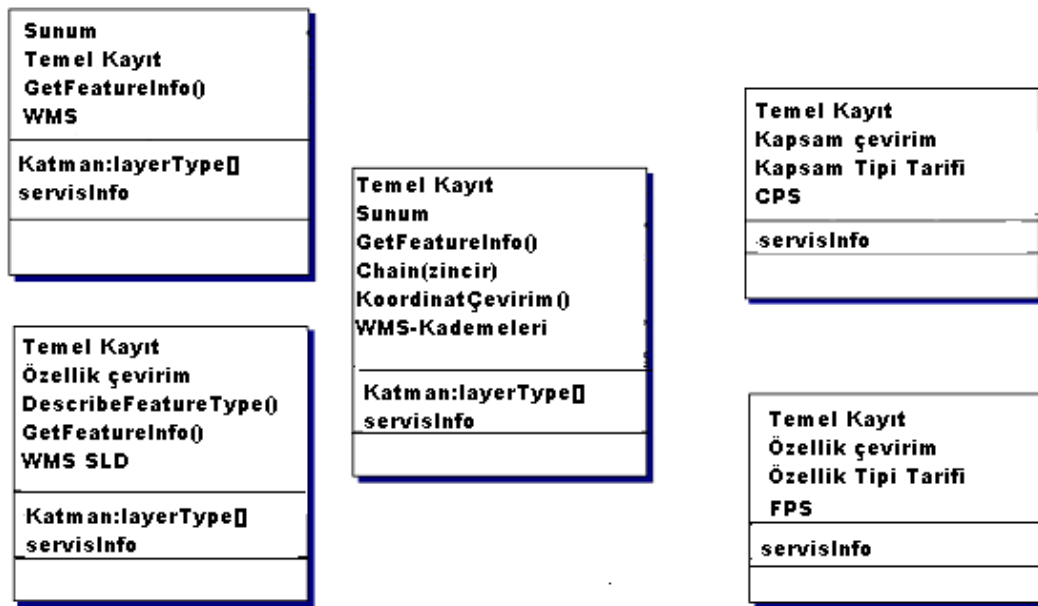
fonksiyonu ile kullanılan servisin özelliklerini içeren bir XML dosyası elde etmek mümkündür.

BBox (BoundingBox): Oluşturulmak istenen haritanın sınır değerlerini belirtir. Sırayla minimum X, minimum Y, maksimum X, maksimum Y koordinatlarından oluşur.

bbox = minx, miny, maxx, maxy

Width & Height: Oluşturulmak istenen haritanın çözünürlüğünü belirtir [36].

WMS (Web Map Service) 'nin kullandığı sınıflar ve bu sınıflara ait fonksiyonlar Şekil 3.1.'de gösterilmektedir. OGC standartlarına dayanan WMS, bu tür sınıf ve fonksiyonları kullanmaktadır.

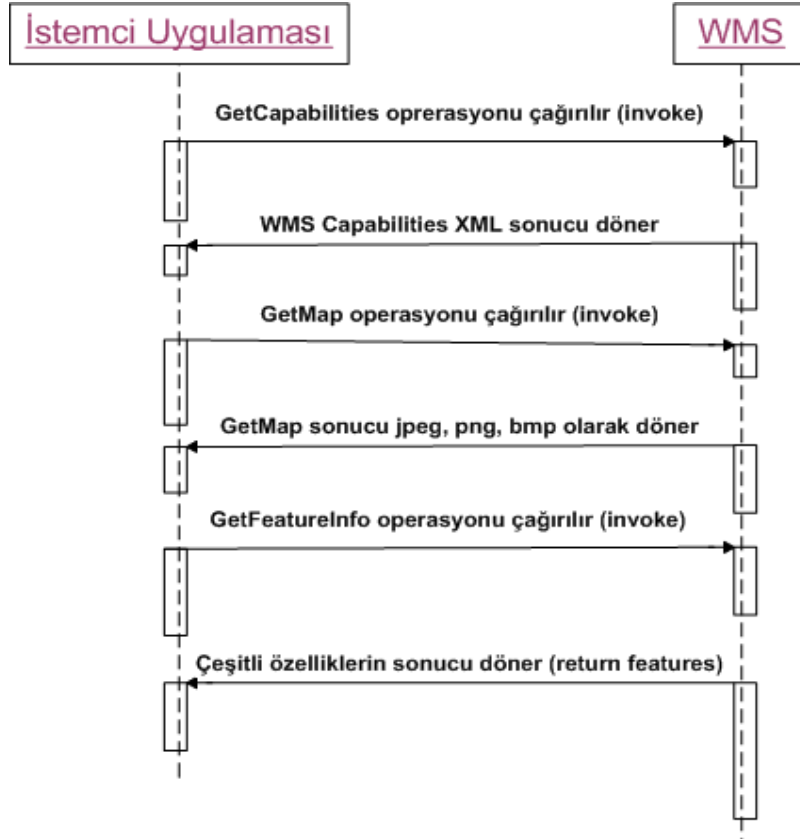


Şekil 3.1. WMS sınıf diyagramları [21].

3.3.1. WMS Fonksiyonları

Web harita servisi olan WMS üç önemli fonksiyona sahiptir. Bunlar; getCapabilities, getMap ve getFeatureInfo' dur. Bu fonksiyonlar ile WMS'nin HTTP üzerinde kullanımını düzenler.

Şekil 3.2.'de WMS fonksiyonları olan getCapabilities, getMap ve getFeatureInfo iş akışı üzerinde gösterilmiştir.



Şekil 3.2. WMS (Web Map Service) 'de iş akışı [21].

3.3.1.1. GetCapabilities

Servis kabiliyetlerini almak amacıyla kullanılan bir fonksiyondur. Kullanımı zorunludur. GetCapabilities isteğine cevap olarak servis hakkında genel bir bilgi ve haritalar hakkında XML şeklinde bilgi döndürülür.

3.3.1.2. GetMap

GetMap fonksiyonu da, kullanımı zorunlu olan bir fonksiyondur. Bu fonksiyon, grafik elemanlar ve resimli görüntüler için tanımlanmıştır. Harita üretmek için kullanılır. .jpeg formatlı resim döndürür.

3.3.1.3. GetFeatureInfo

GetFeatureInfo isteğe bağı bir fonksiyondur. ‘queryable=1’ olarak tanımlanmış katmanlar (layers) için kullanılır. Kullanıcılar diğer katmanlar için bu isteği yayınlamamalıdır. GetFeatureInfo fonksiyonu harita isteklerine cevap olarak gönderilen haritaların özellikleri hakkında bilgi döndürür. Ayrıca WMS ile birlikte kullanıcı tasarımlarına olanak tanımaktadır. GetFeatureInfo fonksiyonu harita hakkında daha fazla bilgi edinmek için kullanılırken, WMS protokolü bulunmadığı için açıklık getirilemeyen konulara da açıklık getirir. Örneğin; map servisin harita kullanıcılarından hangilerinin getMap istek parametrelerini görüntüleyebileceği konusunda bilgi verir. Ayrıca hangi görüntü öğelerinin istendiği, hangi tabakaların getirileceği ve bilginin hangi biçimde döndürüleceği bilgilerini sağlar. BBOX gibi isteklerde enlem boylam bilgileriyle birlikte, WMS pozisyonu hakkında ek bilgiler de döndürülür. Bu açıklanan davranışlar resim durumunda geçerlidir. ‘Grafik eleman durumunda GetFeatureInfo’nun anlamsallığı daha az tanımlanmıştır’ [21].

3.4. WMS – C

WMS-C (Web Map Service – Cache) , WMS ile aynı işlevi görür. Fakat farklı olarak, sorgu sonucu oluşturduğu haritayı hafızaya kaydeder. Yani cache mantığıyla çalışır. Haritaları hafızada tutma mekanizması sayesinde, özellikle çok büyük veri kümeleri üzerinde hızlı ve etkin çalışmayı sağlar. Ayrıca işlemci kaynakları da verimli şekilde kullanılmış olur. Bu durumun dezavantajı ise; işlemci kaynaklarından elde edilen kazanç, hafıza kullanımı için harcanır. Hafıza üzerinde saklanan verilerin miktarı arttıkça diske yazılmayı da gerektirir. Bunun sonucunda, bir süre sonra orijinal verinin onlarca katı hafızada tutuluyor olabilir.

3.5. WFS (Web Feature Service)

WFS, sorguda belirtilen tanımlamalara uygun olarak istenilen haritayı kullanıcıya döndürülür. Aslında WMS ile aynı fonksiyonu görmektedir. Aralarındaki en büyük farklılık şudur: WMS kullanıcıya resim formatında bir harita döndürürken, WFS vektör formatında bir GML belgesi döndürür. Daha önce bahsedildiği gibi GML, XML formatında kodlanmış ve içerisinde harita nesnelerinin verteks koordinatlarının bulunduğu bir dildir. OGC standartlarına dayandığı için birçok Coğrafi Bilgi Sistemi yazılımı tarafından okunabilir ve işlenebilir [36].

Şekil 3.3. WFS sınıflarını diyagramlarını ve temel WFS fonksiyonlarını göstermektedir.



Şekil 3.3. WFS (Web Feature Service) sınıflar diyagramları [21].

Aşağıda örnek bir GML belgesi gösterilmektedir. Bu belge bir poligon tanımlamasını içermektedir.

```
<gml:Polygon>
  <gml:outerBoundaryIs>
    <gml:LinearRing>
      <gml:coordinates>0,0 100,0 100,100 0,100 0,0</gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
```

GML dosyası, verilen koordinatları anlamlandırmaya yarayacak olan projeksiyon ve datum bilgilerini de içerir. Türkiye’de coğrafi nesneleri tanımlamaya olanak tanıyan Ulusal Veri Değişim Formatı (UVDF) kullanılabilmektedir. Buna bir örnek ise aşağıda görülmektedir.

```
<JeodezikNokta>
  <Sinif>YERKONTROLNOKTASI</Sinif>
  <ID>4</ID>
  <Kod>2404</Kod>
  <Koor>
    <Y>705526.068000</Y>
    <X>4229764.962400</X>
```



```

    <z>0</z>
  </Koor>
  <Aci>0</Aci>
</JeodezikNokta>'

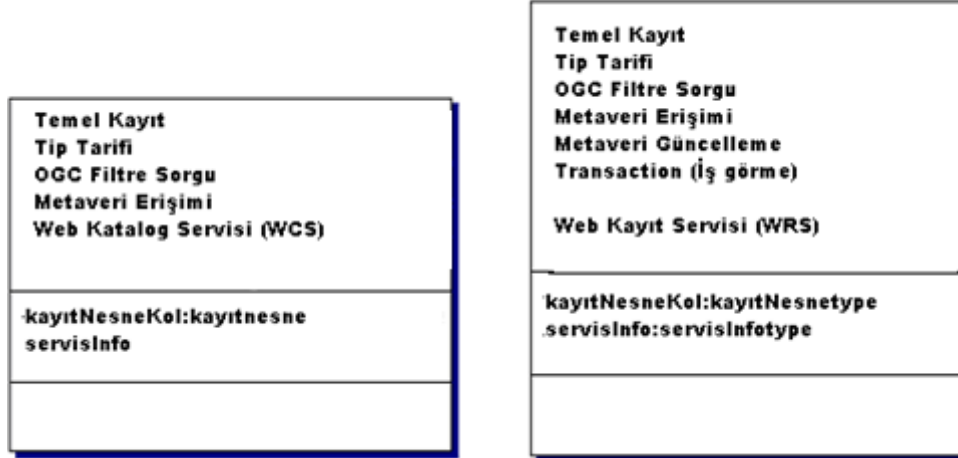
```

Sonuç olarak WFS (Web Feature Service), vektör tipinde GML formatına dayanan haritalar üretir.

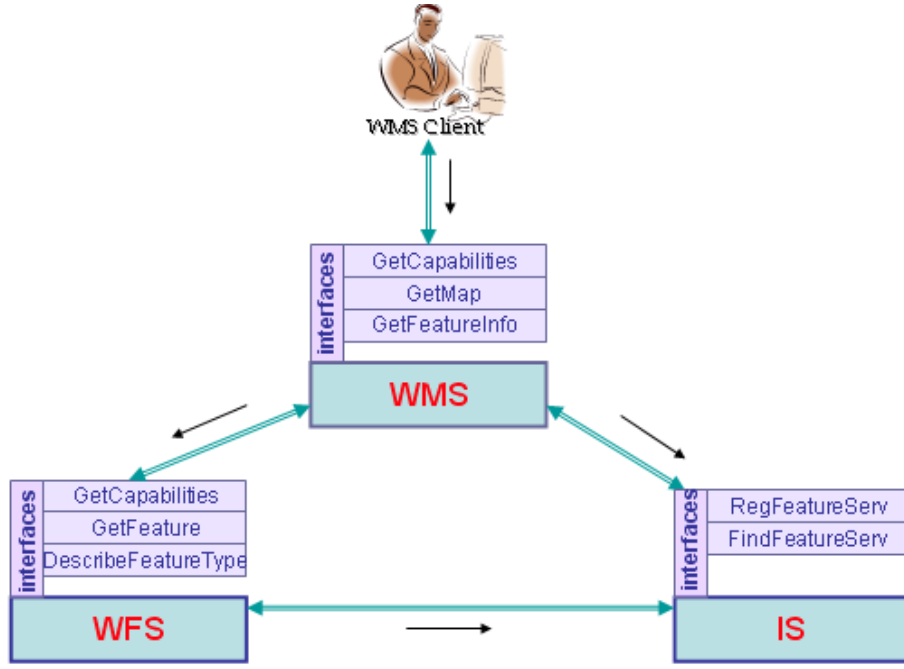
3.6. WCS (Web Coverage Service)

WCS çok yeni bir standart olup WMS' nin özelliklerini taşımaktadır. Aralarındaki tek fark; WMS normal resim formatında haritalar döndürürken, WCS coğrafi referanslı haritalar döndürür. Yani talep edilen harita JPEG veya PNG formatlarının yerine, GeoTiff veya Jpeg2000 gibi coğrafi referanslı resim klasörü şeklinde dönecektir. Yeni bir standart olması sebebiyle çoğu harita sunucusu tarafından desteklenmemektedir [36].

Şekil 3.4. WCS sınıf diyagramları ve WCS temel fonksiyonlarını göstermektedir.



Şekil 3.4. WCS (Web Coverage Service - Kayıt/Katalog servisi) sınıf diyagramları [21].



Şekil 3.5. Coğrafi Bilgi Sistemlerinde kullanılan temel servisler arasındaki iş akışı [21].

Şekil 3.5. ise, Coğrafi Bilgi Sistemlerinde kullanılan temel servisler (WMS, WFS, IS) arasındaki iş akışını göstermektedir.

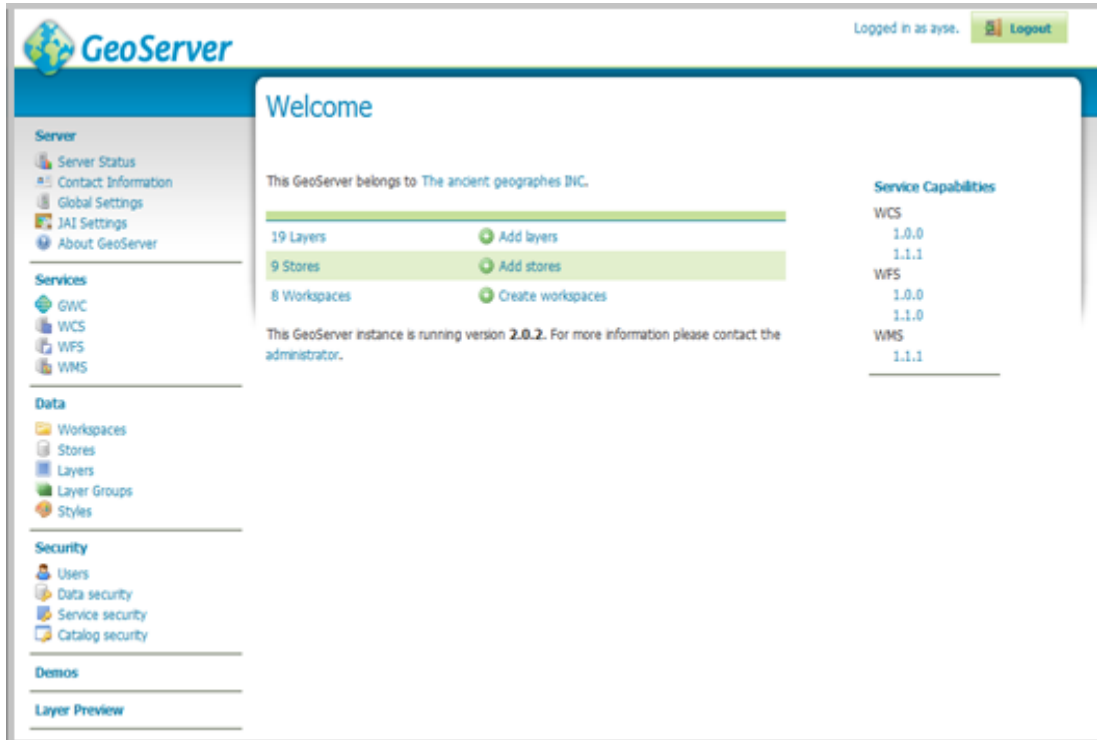
3.7. GeoServer

Açık kaynak kodlu yazılımlarda meydana gelen gelişmeler ve yeteneklerindeki kayda değer artış, kullanıcılar tarafından tercih sebebi olmuş ve bu yazılımlar ülkemizde de internet tabanlı coğrafi bilgi sistemi uygulamalarında yoğun olarak kullanılmaya başlamıştır. Açık kaynak kodlu yazılım olarak büyük oranda kullanıcı ve geliştirici kitlesine sahip olan GeoServer, kullanıcılara mekânsal verilerini düzenleme ve paylaşma imkânı sunan, Java dilinde yazılmış bir programdır. Büyük ölçekli mekânsal verileri OGC Standartlarında sunabilen ve birlikte çalışabilirlik ilkesi ile tasarlanmış bir programdır. GeoServer, OGC Web Feature Service (WFS) ve Web Coverage Service (WCS) standartlarının yanı sıra yüksek performans uyumluluğu garanti edilmiş Web Map Service (WMS)'in bir referans uygulamasıdır. GeoServer, 'konumsal web' kavramının çekirdek bileşenini oluşturmaktadır [37].

GeoServer, coğrafik verileri (çeşitli haritaları) görüntülemek için kullanılan bir harita sunucudur. Açık kaynak kodlu olup, ücretsiz dağıtılmaktadır [38].

GeoServer'ı çalıştırabilmek için, Java bileşenlerinden JDK'nın bilgisayarda kurulu olması şarttır. Ayrıca uygulama geliştirmek için her defasında GeoServer'ı başlatmak, bunun için de 'Start GeoServer' bileşenini çalıştırmak ve uygulama boyunca açık tutmak gerekmektedir. GeoServer çalışır durumdayken, <http://localhost:8080/Geoserver> adresinden çalışma ortamına bağlanılabilmektedir. Kullanıcı girişi yapıldıktan sonra kişisel sunucuya erişmek mümkün olmaktadır.

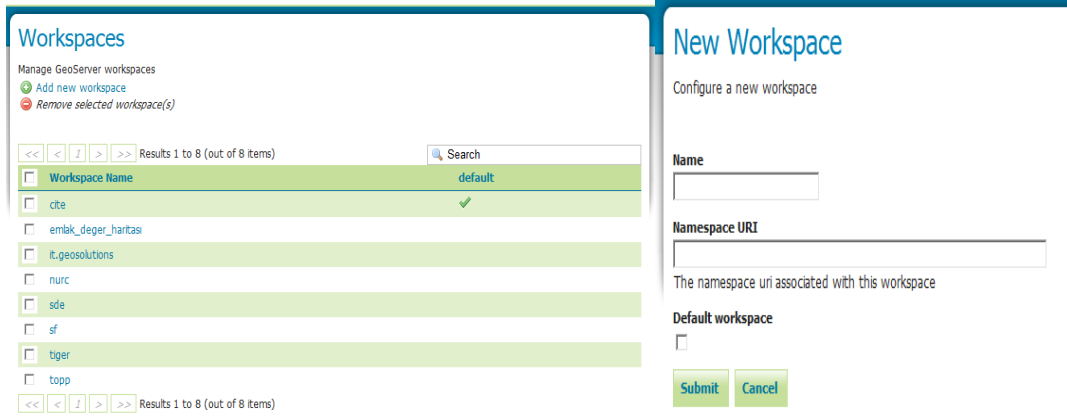
GeoServer çalışma ortamı Şekil 3.6.'da gösterilmektedir.



Şekil 3.6. GeoServer çalışma ortamı

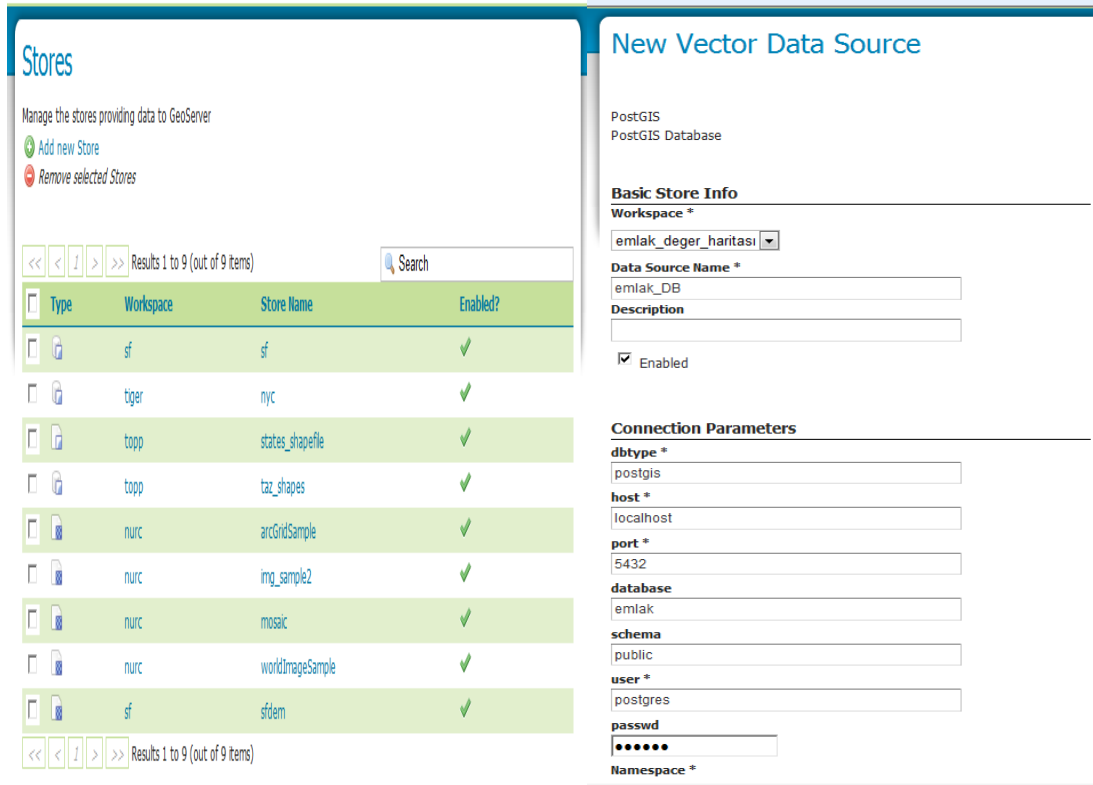
3.7.1. GeoServer Kullanarak Uygulama Geliştirme

GeoServer kullanan uygulamalar geliştirmek için, öncelikle yeni bir çalışma uzayı oluşturulur. Bunun için Şekil 3.7.'de görüldüğü gibi, Add new workspace seçeneği ile yeni bir çalışma uzayı (workspace) tanımlanır. Name ve Namespace URL (örneğin; www.firat.edu.tr) alanları doldurulur.



Şekil 3.7. GeoServer’da yeni çalışma uzayı (workspace) oluşturma ekranları

Ardından veri tabanına bağlanabilmek için, Şekil 3.8.’de görüldüğü gibi Add new Store seçeneği ile yeni bir saklama alanı oluşturulur.



Şekil 3.8. GeoServer’da yeni saklama alanı (store) oluşturma ekranları

Vektör Veri Kaynakları ve Raster Veri Kaynakları olmak üzere iki kısımdan oluşan seçenekler ekrana gelmektedir. Bu tez çalışmasında, PostgreSQL veri tabanı yönetim sistemi ve PostGIS eklentisi kullanıldığı için, Vector Data Sources başlığından PostGIS veri tabanı

seçilir. Yeni bir vektör veri kaynağı eklerken, önceden oluşturulmuş çalışma alanı (workspace) seçilir, veri kaynak adı belirlenir. Zorunlu alanlar olan veri tabanı tipi, host adı ve portu kontrol edilir. Veri tabanı ismi, kullanıcı adı ve şifresi girilerek yeni vektör veri kaynağı kaydedilir.

Bu işlemler yoluyla GeoServer ile PostGIS veri tabanına bağlanılmış olur. Ardından katman (layer) seçimi yapılır. Bu yolla ekli veri tabanlarından, GeoServer haritasında görüntülenmek istenen veri tabanı tablosu seçilir.

New Layer chooser

Add layer from: Choose One

- Choose One
- emlak_deger_haritasi:emlak_DB
- nurc:arcGridSample
- nurc:img_sample2
- nurc:mosaic
- nurc:worldImageSample
- sf:sf
- sf:sfdem
- tiger:nyc
- topp:states_shapefile
- topp:taz_shapes

New Layer chooser

Add layer from: emlak_deger_haritasi:emlak_DB

Here is a list of resources contained in the store 'emlak_DB'. Click on the layer you wish to configure

<< < | > >> Results 0 to 0 (out of 0 items) Search

Published	Layer name	
✓	binalar	Publish again
	binaline	Publish
	mahalleler	Publish
	parseller	Publish
	yollar	Publish

<< < | > >> Results 0 to 0 (out of 0 items)

Şekil 3.9. GeoServer’da yeni katman (layer) oluşturma ekranları

Şekil 3.9.’da da görüldüğü gibi, veri tabanı seçiminin ardından tablo seçimi de yapıldıktan sonra, Veri (Data) ekranında, veri tabanındaki veriye göre x ve y eksenleri için sınır noktaları (Bounding Box) hesaplatılır ve bu yolla haritanın sınırları belirlenmiş olur (Şekil 3.10.).

Bounding Boxes

Native Bounding Box

Min X	Min Y	Max X	Max Y
39,026	38,563	39,371	38,711

[Compute from data](#)

Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
39,026	38,563	39,371	38,711

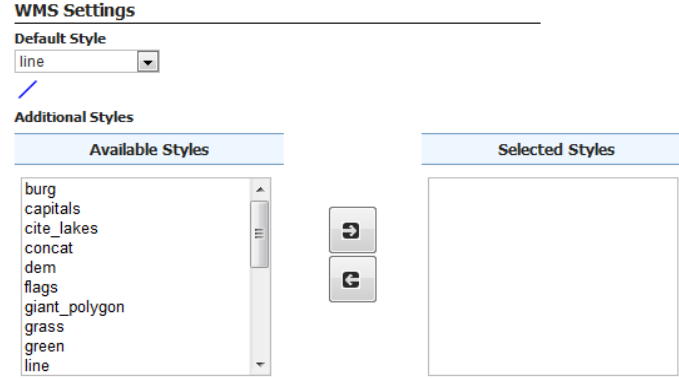
[Compute from native bounds](#)

Feature Type Details

Property	Type	Nilable	Min/Max Occurrences
isim	String	true	0/1
geom	LineString	true	0/1

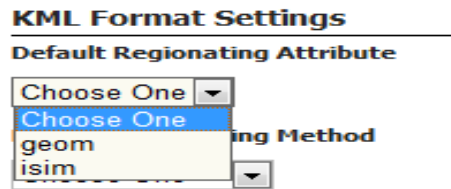
Şekil 3.10. GeoServer’da oluşturulan yeni katman için harita sınırlarının belirlenmesi

Şekil 3.10.'da da görüldüğü gibi veri tabanındaki veriye ait öncül anahtar (primary key) haricindeki nitelikler, GeoServer'in katman (layer) bilgileri içerisinde görüntülenmektedir.



Şekil 3.11. GeoServer'da oluşturulan yeni katman için stil seçimi

Ardından katmana ait Publishing ekranında, stil seçimi yapılır (Şekil 3.11.). Eğer GeoServer'ın varsayılan stilleri haricinde bir stil kullanılmak isteniyorsa, SLD kodlarıyla yeni bir stil de oluşturulabilir. Bu durumda, oluşturulan yeni stil de mevcut stiller içerisinde görüntülenecektir. Ayrıca herhangi seçim yapılmasa bile, GeoServer veriye uygun varsayılan (default) bir stili kendisi belirlemektedir.



Şekil 3.12. GeoServer'da KML format ayarlarının düzenlenmesi

Son olarak, KML format ayarlarından, varsayılan bölgesel nitelik belirlenerek katman oluşturma işlemi tamamlanabilmektedir. Burada dikkat edilmesi gereken husus, varsayılan bölgesel niteliğin, mekânsal olmayan nitelikler içinden seçilmesidir (Şekil 3.12.).

Ayrıca, oluşturulan bu katmanlar gruplanarak tek bir harita üzerinde de gösterilebilir. Bunun için bir katman grubu (Layer Group) oluşturmak gerekmektedir.

Layer group

Edit the contents of a layer groups

Name

Bounds

Min X	Min Y	Max X	Max Y
38,38	38,242	40,384	39,204

[EPSG:WGS 84...](#)

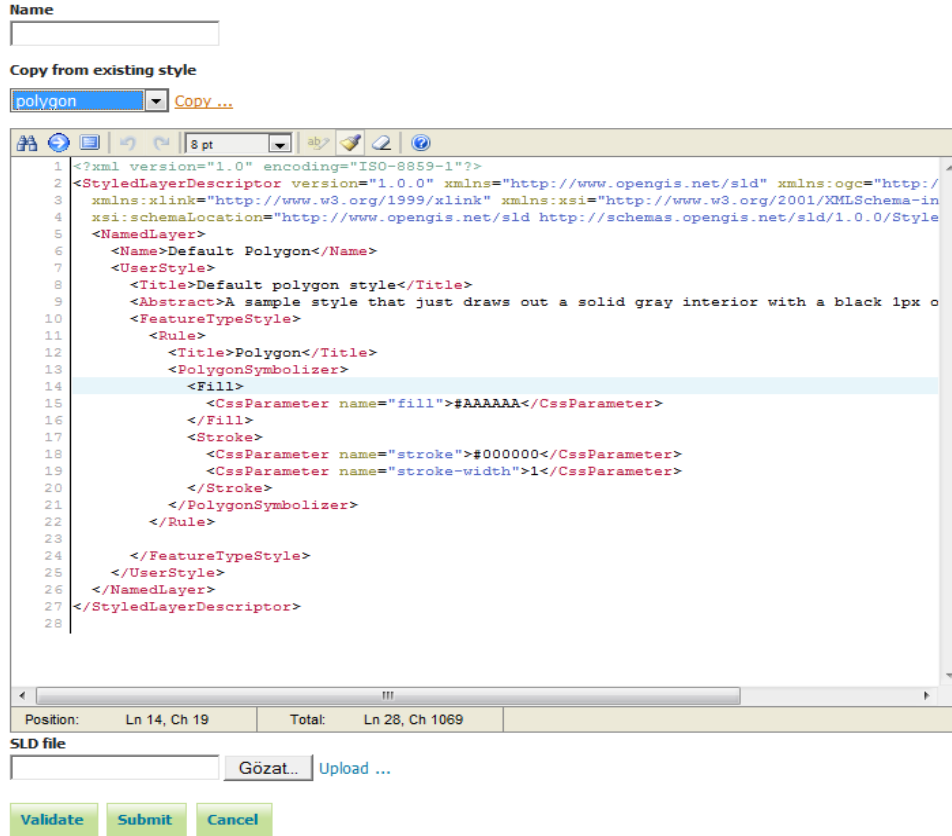
Layers

Layer	Default Style	Style	Remove	Position
yollar	☐	simple_roads	⊖	↓
mahalleler	☐	polygon	⊖	↑ ↓
parseller	☐	polygon	⊖	↑ ↓
binalar	☐	polygon	⊖	↑

Results 1 to 4 (out of 4 items)

Şekil 3.13. GeoServer’da yeni katman grubu oluşturulması

Şekil 3.13.’de görüldüğü gibi, harita üzerinde üst üste görüntülenecek mekânsal veriler katman grubu kullanılarak düzenlenir. En altta kalacak katmandan başlanmak üzere istenildiği kadar katman eklenebilir. Gerekirse eklenen bu katmanların haritadaki pozisyonları değiştirilebilir. Katmanlar eklendikten sonra, Generate Bounds ile görüntülenecek haritanın sınırları otomatik olarak alınır ve katman grubu (layer group) isimlendirildikten sonra yapılan değişiklikler kaydedilerek katman grubu oluşturma işlemi tamamlanır [39].



Şekil 3.14. GeoServer’da SLD kodu ile yeni stil tanımlama

Eğer harita katmanları için yeni stiller belirlenmek isteniyorsa, SLD (StyledLayerDescriptors) kodu kullanılabilir. SLD kodu yardımıyla, haritaları renklendirmek; harita üzerindeki belirli katmanların belirli seviyelerden sonra görünmesini sağlamak ya da bir zoom seviyesindeki şekil boyutunu belirlemek; nesnelere, veri tabanındaki diğer niteliklerine bağlı olarak etiket ya da semboller eklemek mümkün olabilmektedir [40]. Şekil 3.14.’de de görüldüğü gibi, stil oluşturma üç şekilde gerçekleştirilebilir. Bunlar; kod ekranında yeni kod oluşturmak, var olan stillerden kopyalama yapılarak üzerinde düzenleme yapmak ve dosyadan SLD dosyası (SLD file) eklemektir. Bu yollardan birini kullanarak oluşturulan yeni stil, isimlendirilerek kaydedilir ve artık katman (layer) oluşturma sırasında varsayılan stiller arasında görüntülenebilir.

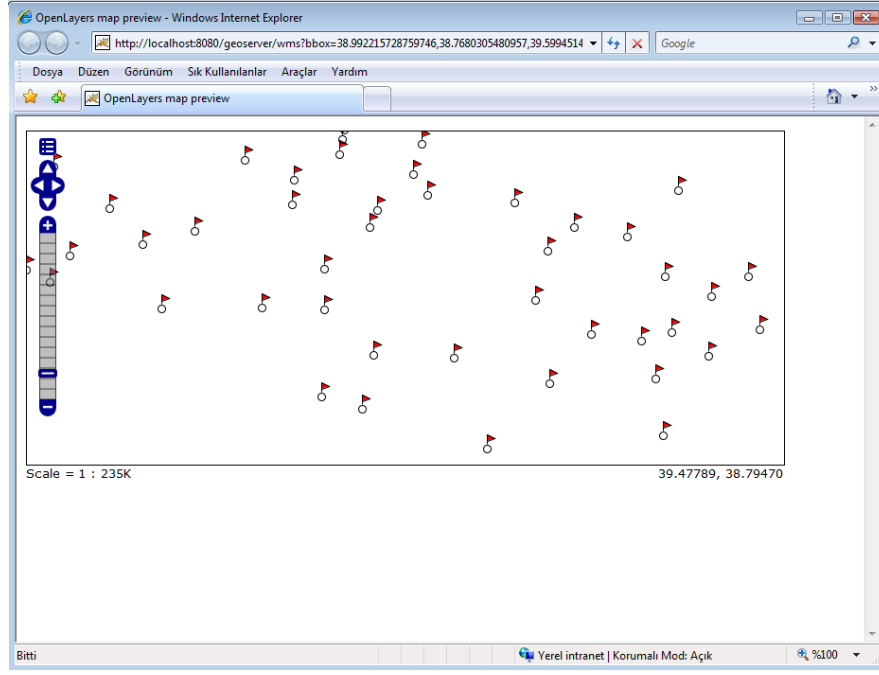
Aşağıda örnek bir SLD kodu görülmektedir. SLD kodu ile yeni bir stil tanımlanırken, ‘Rule’ etiketi içerisinde yeni stilin özellikleri belirtilir. Örnek kodda görüldüğü gibi, mekânsal verinin harita üzerinde farklı renklerde görüntülenmesi buradan düzenlenebilir. Ayrıca yine örnekte görüldüğü gibi, haritada mekânsal veri (örn; poligon) üzerinde, ona bağlı tablosal

verinin görüntülenmesi için, bu verinin veri tabanındaki ismi belirtilerek, ilgili font ayarlamalarının yapılmasıyla, mekânsal verinin etiketlenmesi mümkün olmaktadır.

```
<Rule>
  <Title>Polygon</Title>
  <PolygonSymbolizer>
    <Fill>
      <CssParameter name="fill">#cc3300</CssParameter>
    </Fill>
    <Stroke>
      <CssParameter name="stroke">#000000</CssParameter>
      <CssParameter name="stroke-width">1</CssParameter>
    </Stroke>
  </PolygonSymbolizer>
  <TextSymbolizer>
    <Label>
      <ogc:PropertyName>emlak_degeri</ogc:PropertyName>
    </Label>
    <Font>
      <CssParameter name="font-style">normal</CssParameter>
      <CssParameter name="font-weight">bold</CssParameter>
      <CssParameter name="font-family">Verdana</CssParameter>
      <CssParameter name="font-size">16</CssParameter>
    </Font>
  </TextSymbolizer>
</Rule>
```

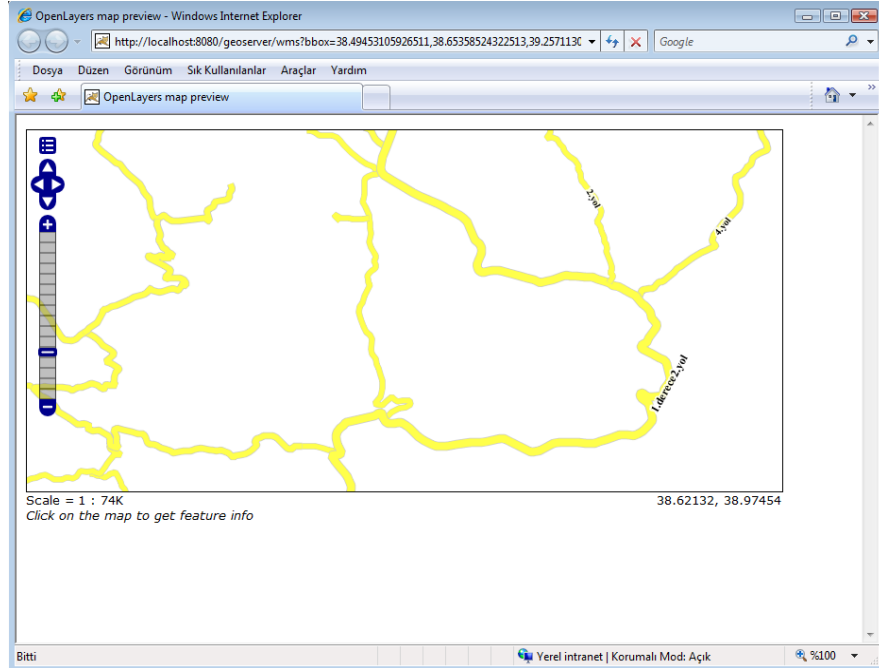
Sonuçta, oluşturulan haritalar, Layer Preview bölümünden bularak görüntülenebilir [39].

GeoServer’da oluşturulmuş örnek bir harita Şekil 3.15.’de gösterilmektedir. Bu örnekte, PostgreSQL veri tabanı kullanılarak kaydedilmiş POINT şeklindeki coğrafik veriler, GeoServer’ın otomatik stili olan bayrak (burg) şekli kullanılarak tanımlanmış ve görüntülenmiştir.



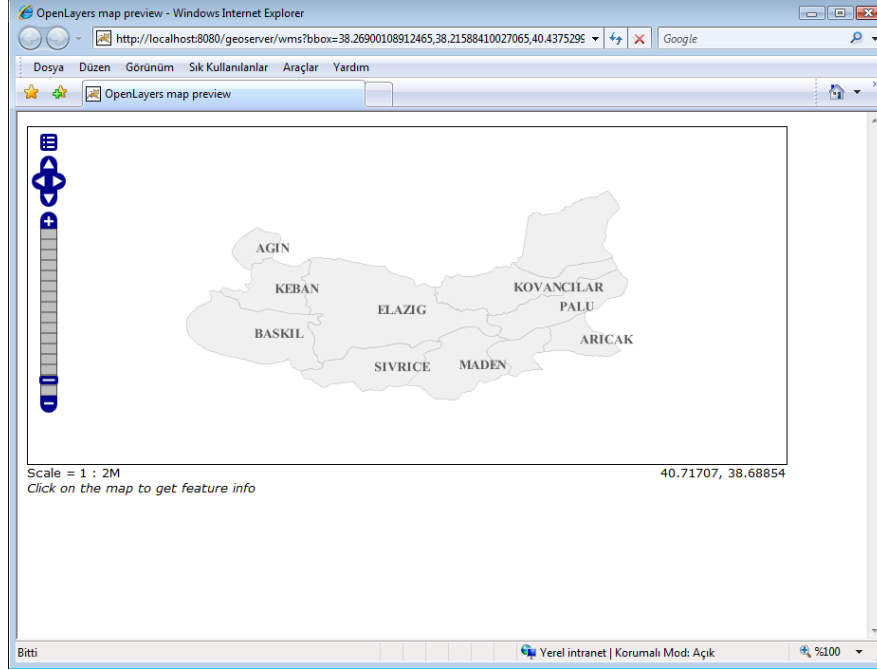
Şekil 3.15. Point şeklindeki verilerin GeoServer’da gösterimine bir örnek

Şekil 3.16.’da LineString tipindeki mekânsal veriler, SLD koduyla düzenlenmiş yeni bir stil kullanılarak görüntülenmektedir.

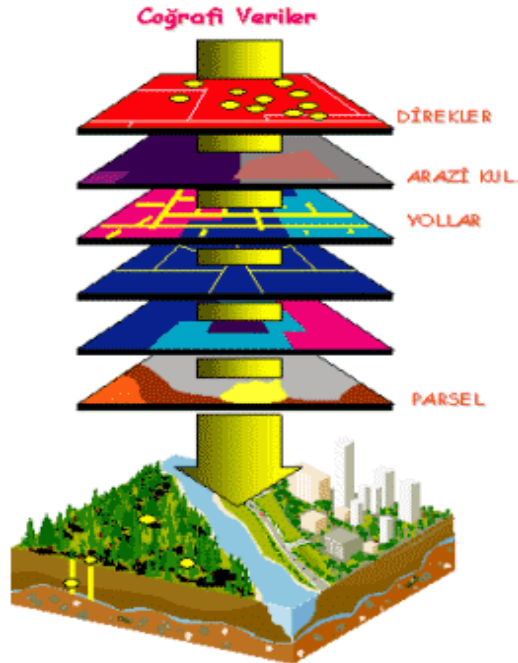


Şekil 3.16. LineString şeklindeki verilerin GeoServer’da gösterimine bir örnek

Şekil 3.17.'de, PostgreSQL veri tabanında poligon şeklinde tutulan veriler için GeoServer'ın sağladığı stiller haricinde, yeni bir stil oluşturulmuştur. Bu yeni stil ile poligon üzerinde etiketleme yapılmış ve harita üzerinde görüntülenmektedir.

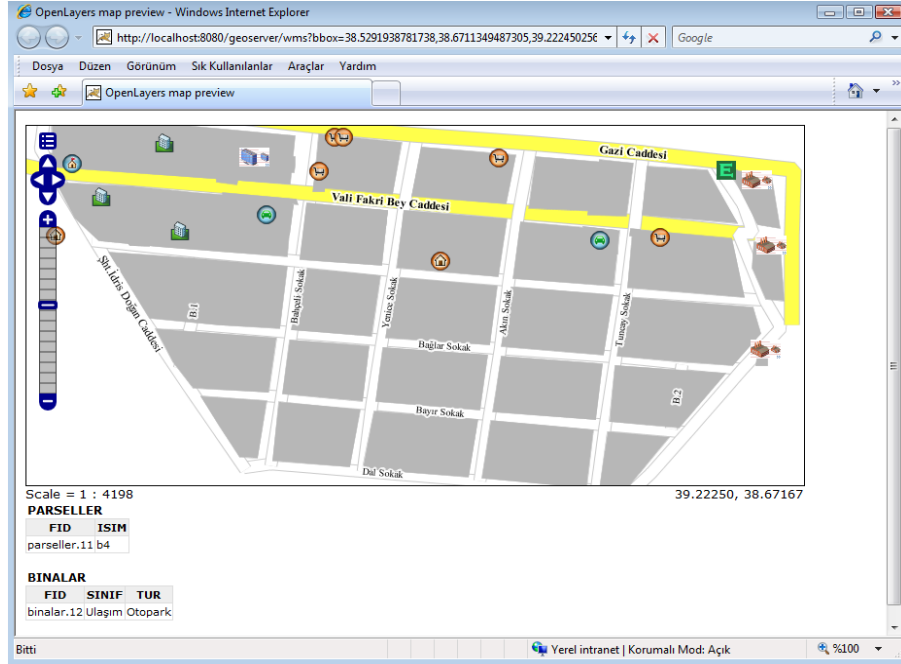


Şekil 3.17. Poligon şeklindeki verilerin GeoServer’da gösterimine bir örnek



Şekil 3.18. Coğrafi veri katmanları [21].

Şekil 3.18.'de birden çok coğrafi veri katmanının üst üste getirilmesi ile yeni bir harita oluşturulması gösterilmiştir. Şekil 3.19.'da ise GeoServer ile katman grubu oluşturularak, birçok katmanın bir arada kullanımı gösterilmektedir. Aşağıda görüldüğü gibi, GeoServer, harita üzerinde görüntülenene nesnelere tıklandığı zaman, nesnelere ait geometrik veriler haricindeki tablosal verilerin de görüntülenmesine imkân tanır.



Şekil 3.19. Birden çok katmandan oluşan verilerin GeoServer’da gösterimine bir örnek

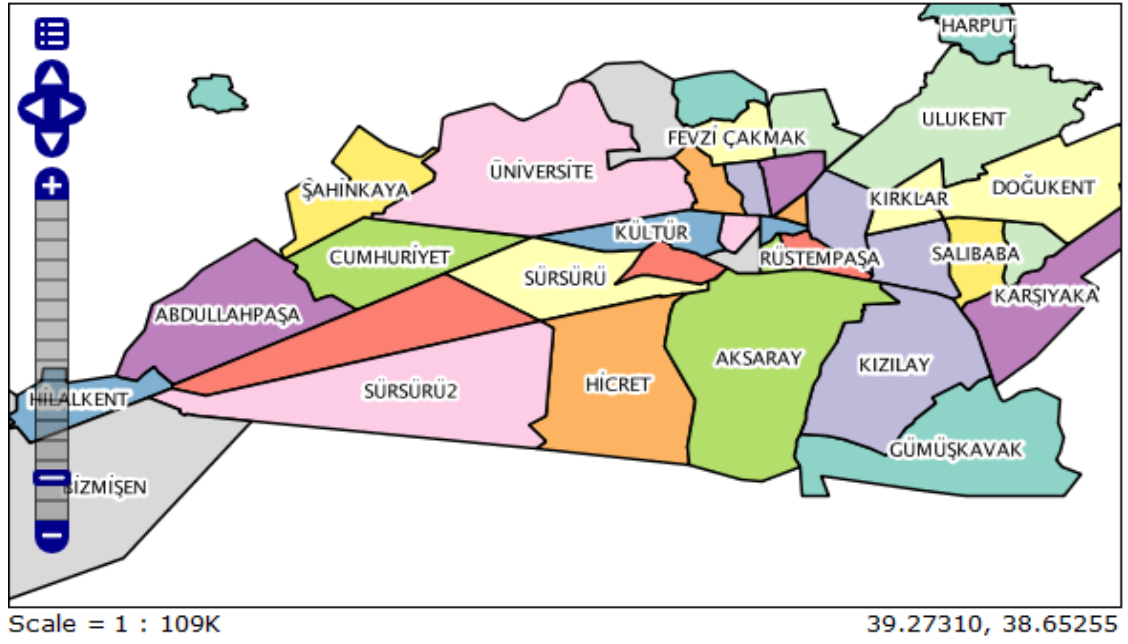
3.8. Emlak Değer Haritası Uygulamasında GeoServer Kullanımı

Emlak Değer Haritası uygulaması, mekânsal verilerin haritalar ile görüntülenmesi ve web üzerinden yayınlanmasını içermektedir. Bunu gerçekleştirebilmek için, bu tez çalışmasında GeoServer isimli harita sunucu (map server) kullanılmıştır. GeoServer harita sunucusu, tanıtılan veri tabanı tablolarına erişim sağlayarak, mekânsal ve mekânsal olmayan verilerin haritalar yardımıyla görüntülenmesine olanak tanır. Tanıtılan veri tabanı tablolarının koordinatları belli geometry tipindeki verilerini, coğrafik özellik taşıyan haritalara dönüştürürken; arzu edilirse, diğer nitelik verilerinin de harita üzerinde etiket şeklinde görüntülenmesine fırsat tanır. Bunun yanı sıra GeoServer üzerinde çalışırken, harita üzerinde geometrik bir veriye tıklandığında, ilişkili nitelik verilerinin görüntülenmesine de olanak tanır.

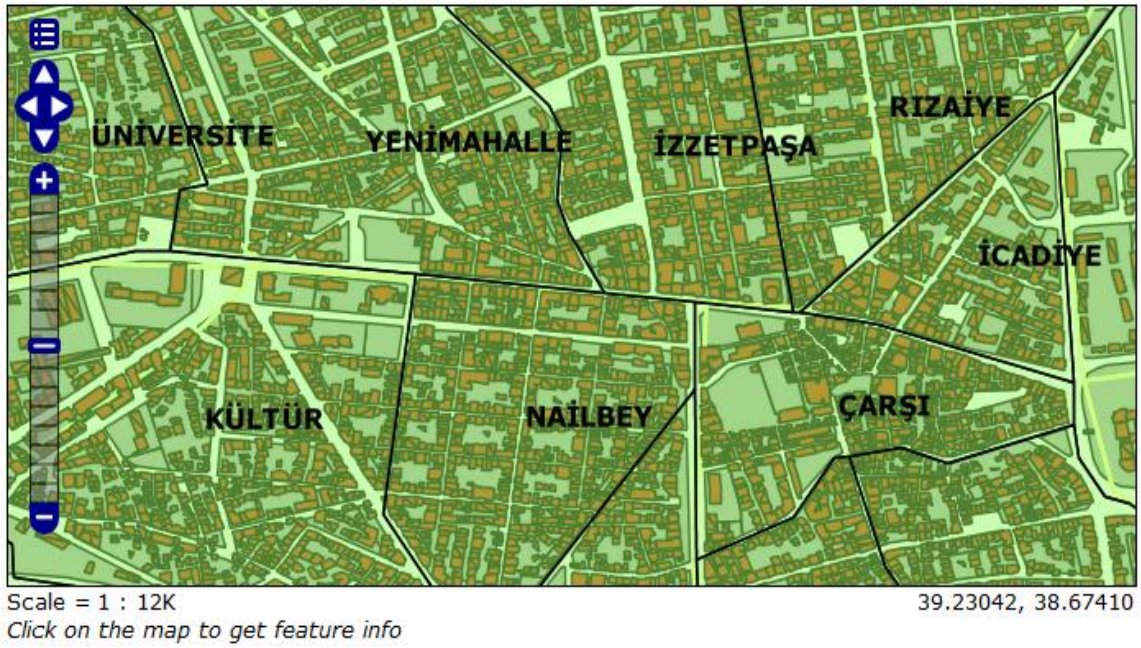
Emlak Değer Haritası uygulaması, GeoServer'ın yukarı da bahsedilen özellikleri kullanılarak gerçekleştirilmiştir. Öncelikle GeoServer'ın local sunucusuna bağlanılarak, emlak_deger_haritasi adında bir çalışma uzayı (workspace) oluşturulmuştur. Ardından, veri tabanına bağlanabilmek için bir saklama alanı (store) oluşturuldu. Bu tez çalışmasında, PostgreSQL Veri Tabanı ve PostGIS eklentisi kullanıldığı için, Vektör Veri Kaynakları ve Raster Veri Kaynakları arasından Vektör Veri Kaynakları (Vector Data Sources) başlığı altından PostGIS veri tabanı seçilmiştir. Önceden oluşturulmuş olan emlak_deger_haritasi çalışma uzayı seçilerek, veri tabanı ismi (emlak), kullanıcı adı ve şifre bilgileri doldurulmuş ve emlak_DB adında bir saklama alanı (store) oluşturulmuştur. Bu işlemler yoluyla, GeoServer ile PostGIS veri tabanına bağlanılmıştır. Sonrasında, emlak veri tabanından, GeoServer haritasında görüntülenmek istenen yol, mahalle, parsel, bina ve nokta verilerinin, veri tabanı tabloları seçilerek, her biri için tek tek katman (layer) oluşturulmuştur. Seçilen her tablonun içindeki geometry tipindeki verilere göre Bounding Box hesaplatılmış ve bu yolla haritanın sınırları belirlenmiştir. Ayrıca her katman için haritada görüntülenmek istenilen şekle uygun olarak stil seçimi yapılmıştır. Stiller genellikle GeoServer'ın sağladığı stiller haricinde SLD (StyledLayerDescriptor) kodu kullanılarak oluşturulmuştur. Oluşturulan katmanlar, emlak_harita adı altında katman grubu olarak birleştirilmiştir. Katman grubu oluşturulurken, parseller katmanı haritada en altta kalacak şekilde, sırasıyla; parseller, yollar, binalar, mahalleler şeklinde eklenmiştir. Mahalleler katmanının opacity özelliği 0.5 şeklinde ayarlanarak, transparan bir görüntü sağlanmıştır. Ayrıca istenirse, katman grubu oluşturulurken, katmanların varsayılan stilleri dışında yeni bir stil seçilmesi de mümkündür.

SLD kodu yardımıyla, emlak değer haritası üzerinde görüntülenen mekânsal veriler renklendirilmiş, harita üzerindeki belirli katmanların belirli seviyelerden sonra görünmesi sağlanmış, zoom seviyesine göre farklı şekil boyutları belirlenmiş; nesnelere, veri tabanındaki diğer niteliklerine bağlı olarak etiketler eklenmiştir. Ayrıca yine SLD kodu ile binaların elde edilen emlak değerleri harita üzerinde görüntülenmiş ve emlak değer aralıklarına göre binalar renklendirilmiştir.

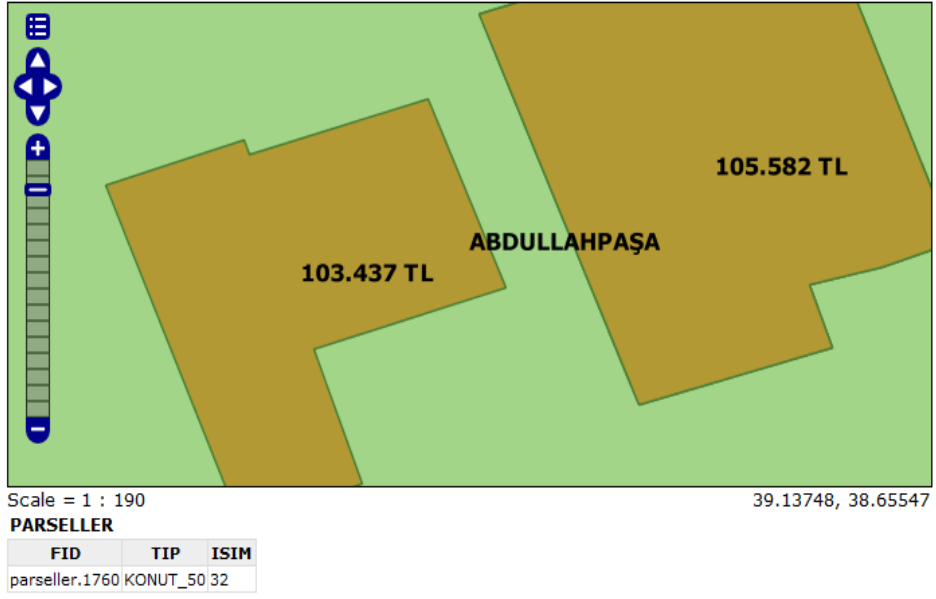
Emlak Değer Haritası uygulaması için GeoServer'da düzenlenen haritalardan bazıları Şekil 3.20 - 3.24.'de görülmektedir.



Şekil 3.20. Mahalleler katmanının GeoServer'daki görüntüsü



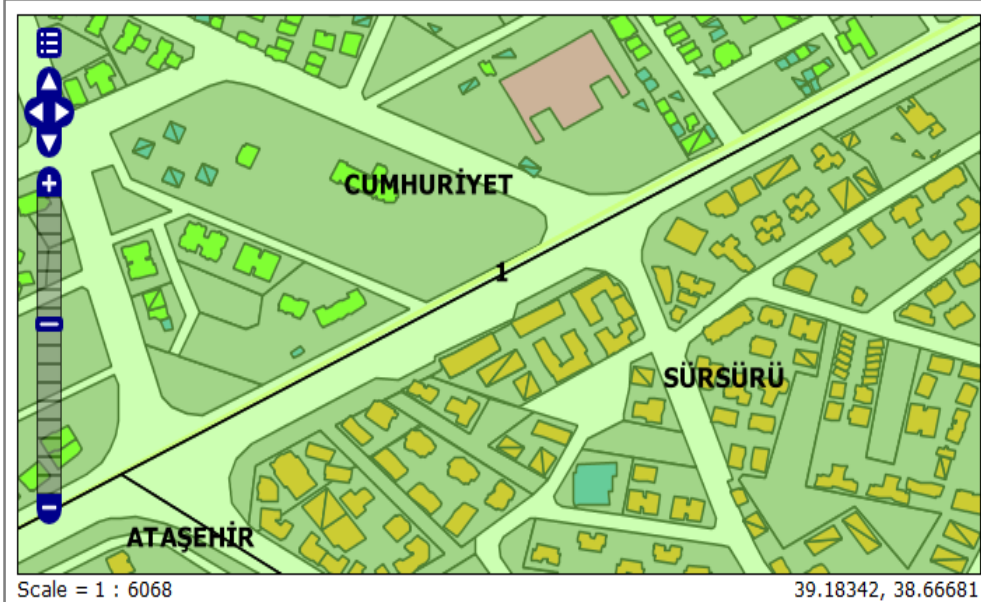
Şekil 3.21. GeoServer'da Emlak Değer Haritasına ait katman grubu



Şekil 3.22. GeoServer’da emlak değerlerinin görüntülenmesi



Şekil 3.23. GeoServer’da emlak değer aralıklarına göre binaların farklı renklerde görüntülenmesi (1)



Şekil 3.24. GeoServer’da emlak değeri aralıklarına göre binaların farklı renklerde görüntülenmesi (2)

4. MEKÂNSAL KARAR DESTEK SİSTEMİ UYGULAMASI OLARAK EMLAK DEĞER HARİTASI GELİŞTİRİLMESİ

4.1. Karar Destek Sistemleri

Karar Destek Sistemleri (Decision Support Systems), ‘Bir yöneticiye karar alma sürecinde yardımcı olmak için, kararların ve işlenen verinin prosedür kümesine dayanan bir model’ olarak tanımlanır [41]. Karar Destek Sistemleri, rutin olmayan karar alma işinde yöneticiyi destekleyen bilgisayar sistemleridir [2]. Karar alma sürecinde rol oynayabilecek şekilde verileri analiz ederler. Karar Destek Sistemleri, karmaşık problemleri çözebilmek için tasarlanmış olup, insan zekâsından, bilgi teknolojisinden ve bilgisayar yazılımlarından faydalanarak sonuç üretirler [42]. Bilgilerin toplanması, işlenmesi ve analizine odaklanan sistemlerdir. [43]. Karar Destek Sistemlerinde esas olan kararın alınması değil, kararın alınmasına destek olmaktır [44]. Bu yönüyle Yönetim Bilgi Sistemlerinden ayrılırlar. Karar Destek Sistemlerinde kullanılan tekniklerin başında Yapay Sinir Ağları, Uzman Sistemler, Genetik Algoritmalar (Sezgisel Optimizasyon Algoritmaları) ve Bulanık Mantık gelmektedir. Bu tez çalışmasında olduğu gibi, Karar Destek Sistemlerini otomatikleştirmek için yapay zekâ teknikleri (optimizasyon algoritmaları) kullanılabilir [42].

Karar Destek Sistemleri, karar alma faaliyetlerini destekleyen hesapsal bilgi sistemlerinin bir sınıfına aittir ve istatistiksel modeller, finansal modeller, optimizasyon modelleri ya da simülasyon modellerinin erişimini ve idaresini gerçekleştirmektedirler [45].

Karar Destek Sistemleri, interaktif bilgisayar tabanlı sistemlerdir [2]. Karar Destek Sistem Kuralları, temel olarak kullanıcılar, gereksinimler ve yazılımları içerir. Kullanıcılar, iki farklı şekilde yetkilendirilebilmektedirler. Ast kullanıcı, uygulama ile ilgili süreçler için belirli analizlere ve karar vericinin sonuçlarına ulaşabilecek şekilde yetkilendirilir. Üst kullanıcı ise, belirli iş ihtiyaçlarına dayalı mekânsal verileri, mekânsal analiz birimleri ve karar birimlerini bulmak, tamamlamak, sonuçlar almak ve yayınlamak üzere yetkilendirilebilmektedirler. Bunun yanı sıra; veri sağlayıcılar farklı alanlarda mekânsal veriler üretip yayınlamaları [43].

Karar Destek Sistemleri, değişik veri kaynaklarından bilgi toplar. Topladığı bu bilgiler içinden gereksiz verileri süzer ve geriye kalanları üzerinde düzenlemeler yapar. Ardından karar alma sürecine etki edecek fonksiyonu oluşturur. Veriler bu fonksiyona göre değerlendirilir. Değerlendirme sonuçları sunulur, karar vericiye karar verme sürecinde bilgisayar destekli bir sistem aracılığıyla yardım sağlanmış olur [1]. Karar Destek Sistemleri,

tekrarlı geliştirme işlemine ihtiyaç duyarlar. Bu işlemler Karar Destek Sisteminin değişimine ve gelişimine olanak tanır [46].

Karar Destek Sistemleri; tahmine ya da verilerin analizine dayanan yönetim bilimi tekniklerinden biri ile Yönetim Bilgi Sistemlerinin birleşmesinden oluşur. Yönetim Bilgi Sistemlerinde, yönetici harici olarak görev alır, yani sadece Yönetim Bilgi Sisteminin ürettiği sonucu uygular. Oysa Karar Destek Sistemlerinde yönetici doğrudan karar alma sürecinde etkilidir. Karar Destek Sisteminin ürettiği sonucu değerlendirirken, karar verme işini kendisi üstlenir. Sonuç olarak, Karar Destek Sistemlerinde, yönetici ile bilgi sistemi arasında bir etkileşim mevcuttur.

Karar Destek Sistemleri; diyalog yönetimi, model yönetimi ve veri yönetimi bölümlerinden oluşur. Diyalog yönetimi, kullanıcı arabiriminin en optimum şekilde düzenlenmesine dayanır. Model yönetimi, sistemin kullanım alanıyla ilgili iş modellerinin aktarıldığı ve yürütüldüğü bölümdür. Veri yönetimi ise, verilerin sisteme aktarılması ve depolanması işini yürüten bölümdür [42].

Karar Destek Sistemleri, karar verme sürecinin yanı sıra, fikir edinme ve alternatifleri gözden geçirmek amacıyla da kullanılabilir. Çoğunlukla yönetim ve stratejik planlama gibi konularda yararlanılmasına karşın, kullanım alanları her geçen gün artmaktadır.

4.1.1. Karar Destek Sistemlerinin Coğrafi Bilgi Sistemlerinde Kullanımı

Karar Destek Sistemleri, Coğrafi Bilgi Sistemlerinin de kullanıldığı karmaşık veriler ve analiz tekniklerini içerir [47]. Mekânsal Karar Alma Modelleri, çok kaynaklı mekânsal veri ve onun analiz sonuçlarına dayalı mekânsal ilişkili problemler üzerine dayandırılırlar [43].

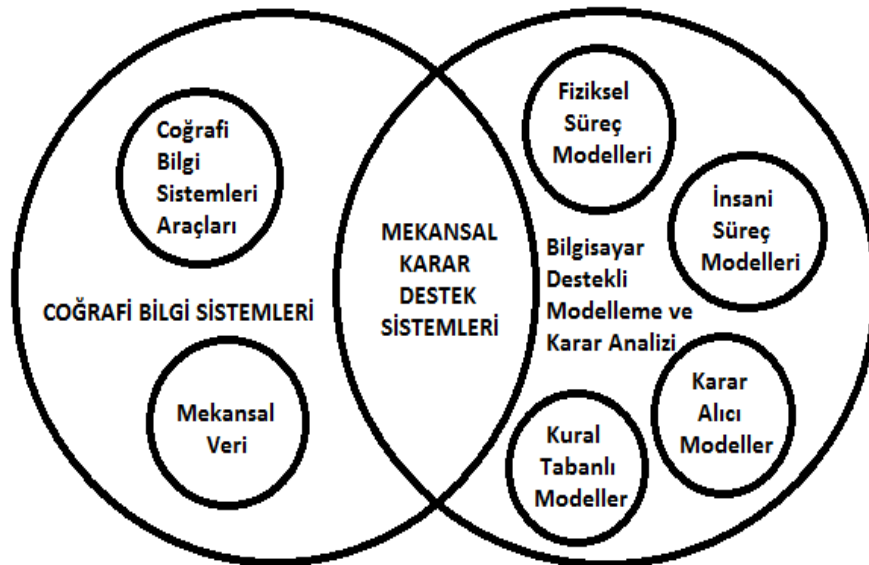
Bilgisayar teknolojisindeki gelişmeler ile birlikte Coğrafi Bilgi Sistemleri de hızla gelişmiş ve özellikle coğrafik verilerden oluşan sistemlerin karar destek mekanizmasında kullanılmaya başlanmıştır [48]. Bunun yanı sıra, web tabanlı teknolojiler ile üç boyutlu (3D) görselleştirme desteği geliştirilirken, Mekânsal Karar Destek Sistemlerinin kullanımında da belirgin bir artış olmuştur [49].

Bu gelişmeler paralelinde, Jonhson [50], ABD konut hareketlilik programı için Coğrafi Bilgi Sistemleri tabanlı bir Karar Destek Sistemi prototipi sunmuştur. Bunun yanı sıra, belirli bölgelerin yoğunluğunu değerlendirmek ve konut arama esnasında, mahalleler hakkında daha iyi kararlar alabilmek için Konut Seçim Programı ile mevcut müşterilere mekânsal bir Karar Destek Sistemi de sunmuştur [51].

Çeşitli alanlarda Karar Destek Sistemleri ve Coğrafi Bilgi Sistemleri birlikte kullanılmaktadır. Örneğin; mekânsal verilerin kullanıldığı, su ve doğal kaynakların yönetimi gibi alanlarda Coğrafi Bilgi Sistemleri karar mercilerine destek sağlamaktadır [47]. Bunun yanı sıra; zemin ve deprem durumuna göre riskli bölgelerin belirlenmesi, imar planlarının hazırlanması; telefon, elektrik, su ve kanalizasyon hatları için şebeke analizlerinin yapılması; otobüs, tren, tramvay ve metro hatları için gerekli değerlendirmelerin yapılması; okullar, acil yardım merkezleri, satış merkezleri vb. için uygun yerlerin seçilmesi gibi alanlarda Coğrafi Bilgi Sistemleri ve Karar Destek Sistemleri etkileşimli çalışarak başarılı sonuçlar üretebilirler [48].

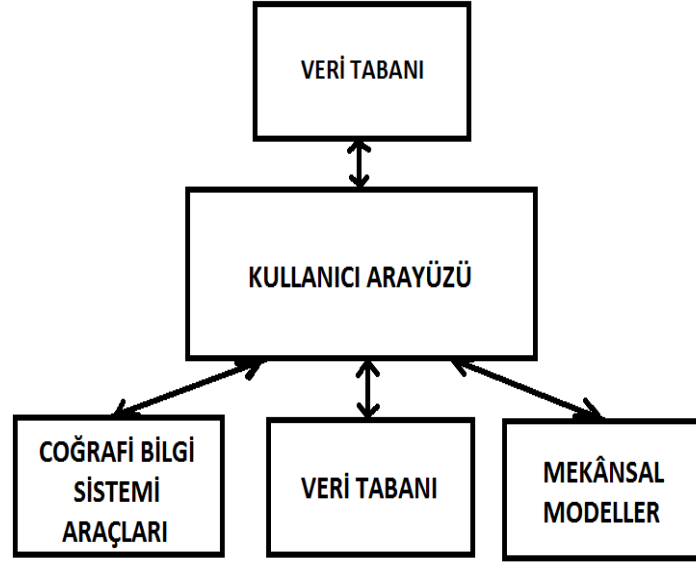
Bilindiği gibi Coğrafi Bilgi Sistemleri, mekânsal verilerin toplanması, depolanması, analizi, modellenmesi ve kullanıcıya sunumunu aşamalarından oluşmaktadır. Bu aşamalardan mekânsal verilerin analizi ve modellenmesi adımları karar mekanizmasına yardımcı olur [48]. Örneğin; coğrafi veriler üzerine yapılan çalışmalarda katman işlemleri, nokta, yüzey ve ağ analiz teknikleri karar vermeyi kolaylaştırıcı araçlardır.

Mekânsal Karar Destek Sistemleri ile problemleri tanımlamak, çözmek ve karar vermek için fiziksel süreç modellerini, insani süreç modellerini, kural tabanlı modelleri ve karar alma modellerini kullanarak, karar vericiye yardımcı olmak için alt sistemler tasarlanır [2].



Şekil 4.1. Coğrafi Bilgi Sistemleri ve Karar Destek Sistemleri arasındaki ilişki [2].

Coğrafi Bilgi Sistemleri, veri işleme ve mekânsal analiz için yetenekliyken, Karar Destek Sistemleri işlem destekli modelleme ve karar analizi için yeteneklidir. Şekilde görüldüğü gibi Mekânsal Karar Destek sistemleri tüm bu özellikleri içine alarak, mekânsal veri, veri işleme, mekânsal analiz ve modelleri kendinde toplar.



Şekil 4.2. Mekânsal Karar Destek Sistemi Mimarisi [2].

Şekilde görüldüğü gibi, Mekânsal Karar Destek Sistemi mimarisinde; veri tabanı, Coğrafi Bilgi Sistemi Araçları ve mekânsal modeller gibi genel bileşenler bulunmaktadır. Ve tüm bileşenler, kullanıcı arabirimi ile etkileşimdedir [2].

Coğrafi Bilgi Sistemleriyle birlikte kullanılan Karar Destek Sistemlerinde analizler iki yaklaşıma göre yapılır. Bunlar; niteliksel ve niceliksel yaklaşımlardır. Analizi yapan kişi tarafından kendi problemine uygun olan yaklaşım seçilir.

Karar Destek Sistemleri için iki yaklaşık yaygın olarak kullanılmaktadır.

- Niteliksel Yaklaşım: Numaralandırmalı veya adlandırmalı verilere sahip projelerde yaygın olarak kullanılır.
- Niceliksel Yaklaşım: Sürekli ve oransal karakterde verilere sahip projelerde tercih edilir.

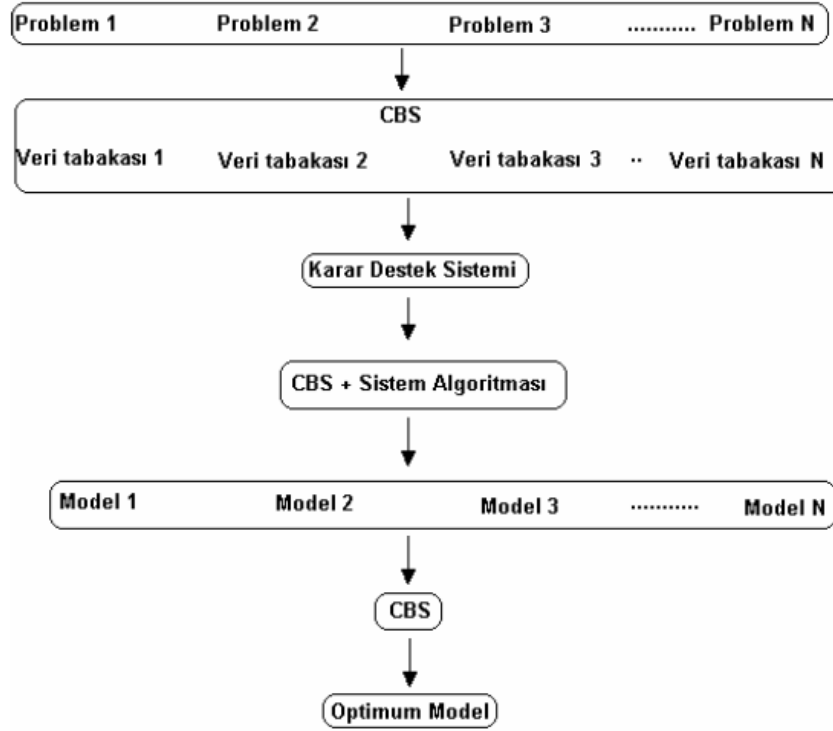
Niceliksel ve niteliksel yaklaşımın birbirine benzeyen yönleri vardır, fakat kullandıkları analitik işlemler açısından farklılık gösterirler [52].

Ayrıca, Coğrafi Bilgi Sistemlerine entegre nitelik katmanları, Karar Destek Sistemlerinin karar verici faktörlerini meydana getirir. Bu faktörler çeşitli yöntemlerle analiz edilebilir. Zhang ve arkadaşları [2], ekonomik konut için geliştirdikleri Mekânsal Karar Destek Sisteminde bu yöntemlerden Çoğullayıcı Analizi kullanmışlardır. Bu sayede konutların farklı şehir bölgelerine dağıtılabilmeleri için gerekli mekânsal istatistikler yapılabilmektedir.

Yu ve Liu [20] ise çalışmalarında, bir lojistik dağıtım firmasının, operasyonel iş akışını basitleştirmek ve optimum araç rotalama imkânı sağlamak için Mekansal Karar Destek Sistemine başvurulmuş ve başarılı sonuçlar elde edilmiştir. Araç rotalama problemi için sezgisel bir algoritma olan Karınca Koloni Algoritmasını, geliştirilen Mekânsal Karar Destek Sisteminde belirleyici algoritma olarak kullanmışlardır.

Coğrafi Bilgi Sistemlerinin veri kaynağı olan coğrafik referanslar, web sitelerindeki köprüler gibi esnek biçimde işlenebilirlerse, Coğrafi Bilgi Sistemlerinin (CBS) kullanım alanları daha da artacaktır. Böylece Coğrafi Bilgi Sistemleri, klasik manada harita üzerine simgeler yerleştiren bir sistem olmaktan çıkarak, açıklayıcı coğrafik veriler elde etmek için de kullanılabilir. Örneğin; sezgisel yöntemler ile elde edilen coğrafik referanslar kullanılarak, bir seyahat rehberi oluşturulabilir [1].

Bazı alanlarda sadece Coğrafi Bilgi Sistemi yazılımları ile analiz yapmak mümkündür. Fakat birden fazla kriterleri sağlayan en optimum çözümü bulmak için, iki sistemin entegre şekilde çalışması gerekmektedir. Örneğin; bir şehirdeki metro hatlarının ve belediye otobüs güzergahlarının belirlenmesi, okul, hastane, karakol, alışveriş merkezi vb. için uygun bölgelerin seçilmesi, zemin durumlarının göz önünde tutulması gibi çoklu seçenek içeren problemlerde Karar Destek Sistemlerinin yanı sıra, çoklu seçim yöntemleri de sisteme eklenmelidir [48].



Şekil 4.3. Karar destek sistemi ve CBS entegrasyonunun değerlendirilmesi [48].

Şekilde görüldüğü gibi, öncelikle problemler coğrafi koordinatlar ile ilişkilendirilerek Coğrafi Bilgi Sistemine işlenir. Problemlere özgü veri tabakaları oluşturulur. Her tabakanın ayrı ayrı modellenmesi yapılır. Sonrasında sorgulamalar yapılarak, her tabakanın tüm kriterlere göre değerlendirilmesi sağlanır. Ardından, her problem için optimum bir model ortaya konulur. Son olarak yerel optimumdan global optimum elde edilmeye çalışılır ve Coğrafi Bilgi Sistemi optimizasyon analizleriyle her problem için optimum modelden, genel optimum model elde edilir. Sonuç olarak birçok seçenek için en uygun çözüm bulunmuş olur.

4.1.1.1. Mekânsal Karar Destek Sistemlerinde İş Akışı

Mekânsal Karar Destek Sistemleri, öncelikle karar verici görev taleplerini analiz eder ve yanıtlar. Ardından belirli bir karar verici görev için uygun, hazır bir iş akış şablonu seçilir. Şablona dayalı Karar Destek Sistemleri, belirli görevler için yeni bir iş akışı oluşturur. Bu iş akışına uygun olarak gerekli mekânsal veriler bulunur ve açıklayıcı belgeler oluşturulur. Karar alma mekanizmasını gerçekleştirecek sistem yapılandırılır ve çalıştırılır. Karar verici sistemin yürütülmesi tamamlandığında sonuçlar kullanıcıya sunulur [43].

Mekânsal Karar Destek Sistemlerinin temel bileşeni mekânsal ve mekânsal olmayan verilerdir. Bu tez çalışması için mekânsal veriler; yollar, mahalleler, parseller, binalar, noktalar. Mekânsal olmayan veriler ise; binaların emlak değerine etki eden faktör puanları, faktör ağırlıkları ve sonuçta elde edilen emlak değerleridir.

4.2. Emlak Değerlerinin Belirlenmesi

Emlak değerinin belirlenmesi, hem gelişmekte olan ülkeler için hem de gelişmiş ülkeler için büyük önem arz eden bir konudur. Coğrafi bir varlık olan taşınmazların değerine çeşitli faktörler olumlu veya olumsuz etki eder. Bu etkilerin optimum şekilde belirlenmesinde Coğrafi Bilgi Sistemlerinin büyük rolü bulunmaktadır. Özellikle veri toplama aşamasında hız kazandırarak, birçok faktörün mekânsal veriler yoluyla belirlenmesine olanak tanıyıp, emlak değer haritalarının oluşturulmasına katkı sağlarlar [53].

4.2.1 Emlak Değerlemede Kullanılan Yöntemler

Taşınmaz değerlemesi iki grupta incelenebilir.

a) Klasik değerlendirme

- 1- Emsal Karşılaştırma Yöntemi
- 2- Maliyet Yöntemi
- 3- Gelir Yöntemi

b) Yeni gelişen değerlendirme

- 1- Stokastik Değerleme Yöntemi
 - Regresyon Yöntemi
 - Nominal Yöntem
 - Genetik Algoritmalar Yöntemi
- 2- Yapay Zekâ Yöntemi [54].

4.2.1.1. Nominal Yöntem ile Emlak Değerleme

Nişancı [53], raster tabanlı bir Emlak Değer Haritası oluştururken Nominal Değerleme Yöntemini kullanmıştır. Bu yöntemin işlem adımları aşağıda verilmiştir.

- 1- Uydu görüntüsünden, konumsal faktörler elde edilir.

- 2- Bu faktörler analiz edilir.
- 3- Parsellerle değerler ilişkilendirilerek her bir parsel için değer ataması yapılır.
- 4- Değer haritası ile kadastral harita karşılaştırılır.
- 5- Elde edilen değerler belli bir birim fiyatı ile çarpılarak emlak değeri bulunmuş olur.

Zeng [55] çalışmasında, emlak değerine etki eden faktörleri 3 grupta toplamıştır. Bunlar; çevresel faktörler, sosyal faktörler, kişisel faktörlerdir.

- a) Çevresel Faktörler: Eğim, bakı, vegetasyon (arazi kullanımı), parklar, doğal kaynaklar, nehir ve kıyılar, taşkın sahaları, çöp alanları, sağlık açısından tehlikeli bölgeler, gürültü, suç bölgesi gibi veriler çevresel faktörler altında sayılmıştır.
- b) Sosyal Faktörler: Alışveriş alanları, büyük marketler, sinemalar, tiyatrolar, eğlence yerleri, spor salonları, otobüs durağı, metro istasyonu gibi ulaşım imkanları, nüfus gibi faktörlerse, sosyal faktörler olarak sayılmaktadır.
- c) Kişisel Faktörler: Çevredekilerin ortalama geliri, çalışma alanları, konut yerleşimi, ev büyüklüğü, kalitesi, lüksü gibi faktörler de kişisel faktörler olarak değerlendirilmiştir [55].

Bunların yanı sıra; resmi makamlarca yapılmış kıymet takdirleri, maliyet hesabı, yıpranma payları, bedelin tespitinde etkili olan diğer objektif ölçütler (enflasyon, trafik emisyonları, taşınmaz aleyhine zeminde tesis bulunan irtifak hakları) da emlak değerine etki edebilecek ölçütlerdendir [56].

Birçok objektif ve subjektif faktörler tarafından etkilenen emlak fiyatı, piyasa fiyatına şekil verir. Piyasa fiyatı çok nadiren piyasa değerine eşittir, çünkü reel emlak piyasası, ideal bir piyasa değildir. Reel emlak piyasa fiyatı birçok subjektif faktörü yansıtır [57]. Örneğin; emlak değerine etki ettiği düşünülen tüm bu faktörlerin yanında; alıcı tercihleri de, kendi özel durumlarına bağlı olarak değiştiği için emlağın gerçek değeri üzerinde etkilidir. Fakat bu durumları göz önüne almak mümkün olamayacağından, oluşturulan emlak değer haritalarıyla gerçek emlak değerleri arasında belirli oranlarda sapma olması normaldir.

Bir parselde değer atanırken, yakınlık- uzaklık üzerinden değerlendirme yapılır. Örneğin; bir okula ya da üniversiteye olan uzaklık arttıkça değere etkisi (olumlu etkisi) azalır.

Ayrıca, okulun değere etkisi ile gürültünün değere etkisini aynı kabul etmek yanlış olur. Bu nedenle önce faktörler puanlanır, sonra faktörlerin ağırlığı belirlenir [55, 58, 59].

4.2.1.2. Puanlama Yöntemi ve Çoklu Karar Analizinin Emlak Değerlemede Kullanımı

Torun [60] yaptığı çalışmada; emlak değerini, faktör puanlarıyla faktör ağırlıklarının çarpımının faktör sayısına bölümü ile hesaplamıştır.

$$\text{Parsel değeri} = (\text{Eğim Puanı} \times 0.79 + \text{Parselin Şekli ve Büyüklüğü Puanı} \times 0.90 + \text{Kamu Hizmetlerinden ve Altyapı Tesislerinden Yararlanma Puanı} \times 0.96 + \text{Cephe Kullanımı Puanı} \times 0.88 + \text{Toprak Puanı} \times 1.00 + \text{Jeoloji Durumu Puanı} \times 1.00 + \text{Gürültü Kirliliği} \times 0.90 + \text{Manzara Bakı Puanı} \times 0.96 + \text{Mevcut İmar Planı} \times 0.98 + \text{Eğitim Merkezlerine, Sağlık Servislerine, Alışveriş Merkezine, Otopark Alanlarına, Ulaşım Sistemine Olan Uzaklık Puanı} \times 0.75) / 10 \quad (4.1)$$

Ayrıca Torun [60] çalışmasında, faktör ağırlıklarını hesaplamak için Çoklu Karar Verme Analizini kullanmıştır. Çoklu Karar Analizi, sonlu sayıda faktörün seçilme, sıralanma, sınıflandırma veya elenme amacıyla birbirleri ile çelişen, aynı ölçü birimini kullanmayan, nitel değerler alabilen, ağırlıklandırılmış çok sayıda kriter kullanılarak değerlendirilmesi işlemidir [61].

İşlem Adımları:

- 1- İlgili emlağa ait güncel veriler toplanır.
- 2- Etki faktörleri belirlenir.
- 3- Karşılaştırmada kullanılacak alternatiflere ait etki faktörlerinin değer, ağırlık, birimleri belirlenir.
- 4- Karar verme matrisi oluşturulur [60].

4.2.1.3. Genetik Algoritmalar ile Emlak Değerleme

Karadağ [54] ise çalışmasında, emlak değerini hesaplamak için Stokastik Değerleme Yöntemlerinden Genetik Algoritmalar Yöntemini kullanmıştır.

Stokastik Yöntemler istatistiki verilere dayanır. Elde edilen veriler arasında bir bağ yoktur. Fakat stokastik birliktelik olarak tanımlanan rastlantısal bir birliktelik vardır. Emlak değerine göre bu birliktelikten faydalanılarak oluşturulan sayısal bağıntılar, diğer emlakların değerini elde etmede kullanılmaktadır.

Genetik Algoritmalar ile emlak değeri, bir emlağa ait olan değerlerin rastgele seçimlilik esasıyla belirlenmesiyle, değeri aranan emlağa değerler üretilmesi yöntemine dayanmaktadır. Emlaklar kromozomları oluştururken, değere etki eden nitelikler de genleri oluşturmaktadır. Her niteliğin emlak değerine etkisi aynı olmayabilir. Emlak değerine farklı şekilde etki eden nitelikler, genlerin farklı türlerine karşılık gelmektedir. Emlak değerini tespit için benzer özellikte emsaller kullanılır. Bu emsal emlakların oluşturduğu topluluk da popülasyon olarak isimlendirilir.

İşlem Adımları:

- 1- Popülasyon oluşturulması
- 2- Popülasyondaki kromozomların uygunluk derecelerinin hesaplanması
- 3- Hedef fonksiyonunun oluşturulması
- 4- Çaprazlama ile popülasyondaki kromozomlar arasında rastgele birleşimin sağlanması
- 5- Mutasyon ile farklı kromozomların oluşturulması
- 6- Aranan gen dizilimine sahip olan emlakların oluşturulması ve elde edilen verilerin analizi
- 7- Sonuç üretilmesi [54].

4.2.1.4. Hesapsal Zekâ ile Gerçek Emlak Kitlesel Değerinin Tahmini

Kontrimas [57] çalışmasında, gerçek emlak kitle değerinin tahmininde Hesapsal Zekâyı kullanmıştır.

Gerçek emlağın kişisel değerlemeleri ve kitlesele değerlemeleri vardır. Kişisel değerlendirme, onun tüm kişisel karakterlerine göre belirlenen kesin nesne değeridir. Kitlesele değerlendirme, standart prosedürler ve istatistiksel testleri kullanarak verilen bir güne dair nitelik gruplarının sistematik değerlemesidir. Bu değerlendirme yöntemi, birçok benzerlikle birlikte, nesnelerin özelliklerine mahsustur. Gerçek emlağın kitlesele değerlemesine yaygın olarak reel emlak vergisini hesaplamak için başvurulur.

Emlak değerlendirme yöntemlerinin Klasik Değerleme ve Yeni Gelişen Değerleme olmak üzere iki grupta incelendiğinden daha önce bahsedilmiştir. Kitlesele Değerleme Modeli, emlak değerlemede yaygın olarak kullanılan Klasik Değerleme Yöntemlerinden Satış Karşılaştırma Yaklaşımı temellidir. Satış Karşılaştırma Yaklaşımına göre; değer, nesnenin piyasada satılan diğer nesnelerle karşılaştırılmasıyla belirlenir. Reel emlak nesneleri farklılıklara sahip olduğu için, değer farklılıklar göz önüne alınarak belirlenir. Değişik nesnelerin karakteristikleri

arasında %30-35'e kadar bir fark kabul edilebilir. Bu yöntem bölgeyi düzenlemek için çok uygundur. Piyasa fiyatının yansması olması, basit ve hızlı hesaplama olanağı bu yaklaşımın temel avantajlarından. Kitlesele değereleme modelleri yaygın olarak Satış Karşılaştırma Yaklaşımına dayanır. En küçük kareler yönteminin lineer gerilemesi, bu yaklaşımdaki modelleri oluşturmak için kullanılan en popüler tekniktir. Fakat Nöral Ağ, Desteklenmiş Vektör Makinesi ve Tahmin Komisyonu gibi diğer teknikler de kullanılabilir. Kontrimas [57] çalışmasında, kitlesele değereleme için tüm seçkin Hesapsal Zekâ tekniklerinin yararlılığını keşfetmeyi amaçlamıştır.

Litvanya'nın merkez kayıtları tarafından sağlanan gerçekçi veriler kullanarak gerçekleştirilen deneysel araştırmalar sonucu, Hesapsal Zekâ temelli tekniklerin performansı, merkez kayıtlarının resmi emlak modellerini kullanarak elde ettiklerinden gözle görülür derecede yüksek bulunmuştur. Bu da, ancak bir uzman yoluyla kaçınılabilen, reel emlak değere alanlarının pahalı iş adımları ve gereksiz zaman harcaması gibi dezavantajlarının önüne geçilebildiğini göstermiştir [57].

4.3. Emlak Değerelemede Kullanılacak Yöntemin Belirlenmesi

Emlak Değere Haritası uygulaması, emlak değereine etki eden faktör ağırlıkları ile ilgili faktör puanlarının çarpılarak, toplamalarının birim arsa değere kabul edilen bir katsayı ile çarpılması yöntemine dayandırılmıştır. Puanlama Yöntemi kullanarak faktör puanları hesaplanmış, Harmoni Arama Algoritması kullanarak da faktör ağırlıkları belirlenmiştir.

Burada; FP, faktör puanı; FA, faktör ağırlığı; m, birim arsa değere olmak üzere,

$$\text{Emlak Değere} = [(FP_1 * FA_1 + FP_2 * FA_2 + FP_3 * FA_3 + \dots + FP_n * FA_n)] * m \quad (4.2)$$

formülü kullanarak binaların emlak değereleleri hesaplanmıştır.

4.4. Emlak Değereine Etki Eden Faktör ve Faktör Puanlarının Belirlenmesi

- 1- **Mevki:** Bu kriter, binanın bulunduğu semt ya da mahallenin popülerliği esasına dayanmakta olup, bu tez çalışmasında arsanın bulunduğu mevkie (mahalleye) göre o arsa için bir puan verilmiştir. Bu puanlama elle yapılmamıştır. Her mahalle için belli bir puan oluşturulup, binanın içinde bulunduğu mahalle belirlenerek, ilgili mevki değere ataması yapılmıştır. Aşağıdaki sorguda da görüldüğü gibi PostGreSQL'e

PostGIS yazılımının eklenmesiyle coğrafik özelliklerin çeşitli komutlarla sorgulanması mümkün olmaktadır.

- `"SELECT binalar.bina_id FROM binalar,mahalleler
WHERE mahalleler.isim='" + mahalle_adi[i] + "'
AND ST_Within(binalar.geom, mahalleler.geom)"`

Burada ST_Within komutu ile iki geometrik veri olan, mahalleler.geom ve binalar.geom kullanılarak, mahalle koordinatları içerisinde kalan binaların ID'leri istenmektedir. Elde edilen bu veri, her bir binanın mahallesine göre mevki değeri atanmasında kullanılmaktadır.

Ayrıca binanın içinde bulunduğu mahalle bilgisi, binalar tablosuna da eklenmiş ve

- `DELETE FROM binalar WHERE mahalle is null`

sorgusu ile veri tabanından merkez mahalleler sınırları içerisinde bulunmayan binaların kayıtları silinmiştir. Tüm hesaplamalar sadece merkez mahalleler içerisinde bulunan binalar göz önüne alınarak yapılmıştır.

- 2- **Konum:** Binaların, konumları batıya yakın olanlarının daha değerli olduğu kabulüne uygun olarak; Elazığ Merkez ilçesinde bulunan binaların orta noktası bulunarak, merkeze 50 puan verilip, doğuya gidildikçe bu puan azaltılmış, batıya gidildikçe arttırılmıştır. Veri tabanında bulunan binalardan konumu itibariyle en batıdaki bina maksimum faktör puanına sahip olmuştur. Ve tüm kriterlerde olduğu gibi konum kriteri için de, faktör puanı maksimum 100 kabul edilerek, gerekli hesaplamalar yaptırılmıştır.

Bu hesaplamalar sırasında, bina poligonlarını birleştirip, tüm binaların merkez noktasını bulmak için PostGIS'in ST_Collect() ve Center() isimli iki fonksiyonu kullanılmıştır. ST_Collect() fonksiyonu geometry tipindeki verileri birleştirerek bir multi-geometry oluşturur. Center fonksiyonu ise oluşturulan geometry tipindeki verinin merkez noktasını bulmak için kullanılmaktadır.

- `SELECT Center(ST_Collect(geom)) FROM binalar`

Bu sorguyla, tüm binaların merkez noktasının x koordinatı alınarak, her bir binanın merkez noktasının x koordinatıyla arasındaki uzaklık hesaplanmıştır. Eğer bina merkezden doğuda kalıyorsa, uzaklık farkı negatif(-) değer alır ve buna bağlı olarak binanın değeri de 50'den küçük olur. Eğer bina konum itibariyle merkezden batıda ise, uzaklık farkı pozitif(+) değer alır ve binanın konum faktör puanı da 50'den büyük bir değer alır.

- 3- Okul Yakınlık:** Elazığ Merkezde bulunan ilköğretim, ortaöğretim, üniversite gibi okullara uzaklıkla ters orantılı olarak hesaplanmıştır. Örneğin; bir binanın üniversiteye olan uzaklığı arttıkça, ilgili faktör puanı düşer. Ayrıca üniversitenin emlak değerine etkisi, ilköğretim okulunun emlak değerine etkisi ile aynı olmayacağı için, faktör puanı hesaplanırken, 'ILKOGRETIM', 'ORTAOGRETIM', 'MESLEK_LISE', 'UNIVERSITE' şeklinde değerler almış olan parsel tipi niteliği göz önünde bulundurulmuştur.

```
• " SELECT parseller.tip, ST_DISTANCE(binalar.geom,parseller.geom)
  FROM parseller,binalar
 WHERE (parseller.tip='ILKOGRETIM'
 OR parseller.tip='ORTAOGRETIM'
 OR parseller.tip='MESLEK_LISE'
 OR parseller.tip='UNIVERSITE')
 AND binalar.bina_id=" + bina_id[i])
```

Yukarıdaki SQL komutunda da görüldüğü gibi parsel tipi okul belirten tüm parsellerin her bir bina ile arasındaki uzaklık, yine PostGIS fonksiyonlarından biri olan ST_DISTANCE ile hesaplatılmıştır. ST_DISTANCE komutu iki geometrik veri arasındaki uzaklığı hesaplatmak için kullanılmaktadır.

İlgili faktör puanlarının hesaplanmasından sonra, maksimum değere sahip bina 100 puan olacak şekilde faktör puanları yeniden belirlenmiştir.

- 4- Hastane Yakınlık:** Parseller tablosunda yer alan parsel tipi 'SAGLIK' olarak belirtilen verilerin, merkez mahalleler içinde bulunan her bir binaya olan uzaklığı ile ters orantılı olarak hesaplanmıştır.

- "SELECT ST_DISTANCE(binalar.geom,parseller.geom)*100
FROM parseller,binalar
WHERE parseller.tip='SAGLIK'
AND binalar.bina_id=" + bina_id[i]

Okul Yakınlık faktör puanının hesaplanmasında kullanıldığı gibi, PostGIS'in ST_DISTANCE fonksiyonu burada da kullanılmıştır.

5- Alışveriş Merkezlerine (AVM) Yakınlık: Burada alışveriş merkezi kavramı daha geniş düşünülmüştür. Elazığ ili için büyük alışveriş merkezi sayısının az olmasından dolayı, diğer ticari merkezlerin de emlak değerinde etkin bir faktör olduğu düşünülmüş ve ilgili faktör puanının hesaplanmasında bu durum göz önüne alınarak hareket edilmiştir.

Elazığ ili merkez mahalleleri içerisinde bulunan her bir binanın, parsel tipi 'TICARET' ya da 'PAZAR_ALANI' olan geometry tipindeki verilere olan uzaklığı bulunarak ve bu uzaklıkla ters orantılı olarak da faktör puanı hesaplatılmıştır.

- " SELECT parseller.tip, ST_DISTANCE(binalar.geom,parseller.geom)
FROM parseller,binalar
WHERE (parseller.tip='TICARET'
OR parseller.tip='PAZAR_ALANI')
AND binalar.bina_id=" + bina_id[i]

Ayrıca 'TICARET' tipindeki parsel ile 'PAZAR' tipindeki parselin faktör puanına etkisi de aynı değildir. Bu şart göz önünde tutularak hesaplamalar yapılmıştır.

6- Sosyal İmkânlara Yakınlık: Bu kriterin puanlanmasında elimizdeki veri tabanında bulunan ve sosyal olarak Elazığ halkına hitap edebilecek her unsur göz önüne alınmıştır. Bunlar; 'KULTUR', 'SPOR', 'OYUN_ALANI', 'TURİZM', 'KALE', 'FUAR_ALANI', 'PARK', 'CAMI', 'ORMAN', 'AGACLANDIRMA' tipindeki verilerdir.

- "SELECT ST_DISTANCE(binalar.geom,parseller.geom) FROM
parseller,binalar
WHERE (parseller.tip='KULTUR' OR parseller.tip='CAMI'
OR parseller.tip='SPOR' OR parseller.tip='OYUN_ALANI'

```

OR parseller.tip='TURIZM'      OR parseller.tip='KALE'
OR parseller.tip='FUAR_ALANI'  OR parseller.tip='PARK'
OR parseller.tip='ORMAN'      OR parseller.tip='AGACLANDIRMA')
AND binalar.bina_id=" + bina_id[i]

```

Diğer uzaklık hesaplamalarında olduğu gibi, burada da PostGIS'in ST_DISTANCE fonksiyonundan yararlanılmıştır ve tüm binalar için sosyal imkan yakınlık faktör puanı hesaplandıktan sonra, maksimum değere sahip bina 100 puan olacak şekilde faktör puanları yeniden belirlenmiştir.

- 7- İş İmkânlarına Yakınlık:** Bu faktör puanının hesaplanmasında da benzer mantıkla, her bir binanın, parsel tipi 'RESMI', 'TICARET', 'TOPTAN_TICARET', 'SANAYI', 'KUCUK_SANAYI', 'PAZAR_ALANI' ya da 'GUNU_BIRLIK' olan parsellere olan uzaklığı hesaplatılmış ve bu uzaklıkla ters orantılı olarak puanlama yapılmıştır.

```

• " SELECT parseller.tip, ST_DISTANCE(binalar.geom,parseller.geom)
FROM parseller,binalar
WHERE (parseller.tip='RESMI'
OR parseller.tip='TICARET' OR parseller.tip='TOPTAN_TICARET'
OR parseller.tip='SANAYI' OR parseller.tip='KUCUK_SANAYI'
OR parseller.tip='PAZAR_ALANI' OR parseller.tip='GUNU_BIRLIK')
AND binalar.bina_id=" + bina_id[i]

```

Ayrıca tüm bu parsel tipleri, faktör puanının hesaplanmasında aynı ağırlığa sahip değildir. Örneğin; bir binanın 'RESMI' tipindeki parselin uzaklığı, 'PAZAR_ALANI' tipindeki parselin uzaklığına eşit olsa bile, bu iki uzaklığın faktör puanını azaltıcı etkisi aynı değildir. Yani binanın 'RESMI' tipindeki parselin yakınlığı daha önemli olduğu için, uzak olması da faktör puanının daha düşük olmasına sebep olur.

- 8- Ulaşım İmkânlarına Yakınlık:** Modern kentlerde hızlı ulaşım her zaman ana taşımacılık yoludur. Özellikle orta ve düşük gelirli aileler için, şehrin hızlı ulaşım hatlarına yakın yaşamak hem seyahat maliyeti seviyesini korumak için, hem de daha kolay iş bulmak için önemlidir [2]. Yani ulaşım imkânlarına yakınlık emlak seçiminde belirleyici faktörlerden biridir. Bu kriter için binalar tablosundaki her bir binanın, parseller tablosundaki parsel tipi 'TERMINAL' ve 'DEMIRYOLU' olan merkezlere

uzaklığı ve noktalar tablosundaki 'OTOBUS_DURAGI' isimli ulaşım imkanlarına uzaklığı hesaplanarak, bu uzaklıkla ters orantılı olarak puanlama yapılmıştır.

- " SELECT parseller.tip, ST_DISTANCE(binalar.geom,parseller.geom)
FROM parseller,binalar
WHERE (parseller.tip='TERMINAL'
OR parseller.tip='DEMIRYOLU')
AND binalar.bina_id=" + bina_id[i]
- " SELECT ST_DISTANCE(noktalar.geom,binalar.geom)
FROM noktalar,binalar
WHERE noktalar.isim='OTOBUS DURAGI'
AND binalar.bina_id=" + bina_id[i]

Diğer uzaklık hesaplamalarında olduğu gibi, ilgili faktör puanlarının hesaplanmasından sonra, maksimum değere sahip bina 100 puan olacak şekilde faktör puanları yeniden belirlenmiştir.

'TERMINAL' tipindeki parsel ile 'DEMIRYOLU' tipindeki parselin faktör puanına etkisi aynı olmadığı gibi, 'OTOBUS DURAGI' isimli noktanın da faktör puanına etkisi bunlarla aynı değildir. Bu doğrultuda, uzaklıkları farklı katsayılarla çarpılarak hesaplama yapılmıştır.

- 9- Gürültüye Yakınlık:** Bu kriter yola yakınlıkla ters orantılı olarak hesaplanmıştır. Yani bir binanın yola olan uzaklığı arttıkça gürültüye yakınlığı azalır, dolayısıyla ilgili faktör puanı da artar.

- "SELECT binalar.bina_id, yollar.tip,
ST_DISTANCE(binalar.geom,yollar.geom)*100
FROM binalar,yollar,mahalleler
WHERE ST_Within(yollar.geom,mahalleler.geom) AND
binalar.bina_id<=10000
order by bina_id, mahalleler.isim, yollar.yol_id"

PostGIS'in ST_DISTANCE fonksiyonu binalarla yollar arasındaki uzaklığı hesaplamak için kullanılmıştır. Ayrıca ST_WITHIN fonksiyonu da, yalnızca

mahalleler sınırları içinde kalan yolların değerlendirilmeye alınması amacıyla kullanılmıştır.

Yaklaşık 38.000 binanın, her bir yola olan uzaklığını sorgulamanın tek bir seferde gerçekleştirilememesi sebebiyle, yukarıdaki sorguda bina_id'in 10.000'den küçük olması şartı eklenmiştir. Böylece tüm sorgulama dört seferde gerçekleştirilmiştir.

10-Merkezi Yollara Yakınlık: Gürültü Yakınlık kriterinde olduğu gibi, her bir binanın merkez mahalleler sınırları içerisinde kalan yollar ile uzaklığı hesaplanarak, gürültü yakınlık kriterinin tersi şekilde puanlaması yapılmıştır. Yani bir binanın yola uzaklığı arttıkça ilgili faktör puanı azalmıştır.

- "SELECT binalar.bina_id, yollar.tip,
ST_DISTANCE(binalar.geom,yollar.geom)*100
FROM binalar,yollar,mahalleler
WHERE ST_Within(yollar.geom,mahalleler.geom) AND
binalar.bina_id<=10000
order by bina_id, mahalleler.isim, yollar.yol_id"

Gürültü yakınlık faktöründe olduğu gibi, PostGIS'in ST_DISTANCE ve ST_WITHIN fonksiyonları kullanılmış ve aynı şekilde sorgulama dört seferde gerçekleştirilmiştir. İlgili faktör puanlarının hesaplanmasından sonra, maksimum değere sahip bina 100 puan olacak şekilde faktör puanları yeniden belirlenmiştir.

11- Parsel Şekli ve Büyüklüğü: Parsellerin şekil bakımından değerlendirilmesi, üzerinde bina yapmaya elverişli olup olmadığını anlaşılabilmesi için önemlidir. Fakat bu tez çalışmasında zaten binaların parsel değerleri hesaplanmakta olduğu için, şekil bakımından değerlendirme yapılmamıştır. Sadece binanın parsel büyüklüğü hesaplanarak değerlendirmeye alınmıştır.

- SELECT bina_id, ST_AREA(geom)*POWER(0.3048,2)*10^11 FROM binalar

Binanın parsel büyüklüğünü hesaplamak için yukardaki SQL kodundan yararlanılmış olup, PostGIS'in alan hesaplama fonksiyonu olan ST_AREA fonksiyonu kullanılmıştır. ST_AREA, *fit*² cinsinden sonuç ürettiğinden, *m*² cinsinden sonuç elde etmek için yukarıdaki dönüşüm uygulanmıştır. Tüm binaların parsel büyüklüğü

hesaplandıktan sonra, maksimum alana sahip bina 100 puan olacak şekilde faktör puanları;

$$\text{son_alan}[i] = \text{alan}[i] * 100 / \text{max_alan} \quad (4.3)$$

formülüne göre yeniden belirlenmiştir.

12-Zemin Durumu: Zemin durumu için de yine mahalleler bazında değerlendirme yapılmıştır. Mahallenin kayalık zeminde olması, daha sağlam yapıda ve depreme dayanıklı olduğunu gösterdiği için, bu durumdaki yerlerin ilgili faktör puanı yüksek tutulurken, eski dere yatağı gibi yumuşak zeminli bölgelerin depreme dayanıklı olmamaları nedeniyle, ilgili faktör puanları düşük tutulmuştur. Bu kurallar dikkate alınarak, her mahalle için belli bir faktör puan oluşturulduktan sonra, binanın içinde bulunduğu mahalle belirlenerek, ilgili zemin durumu puanı atanmıştır.

13-Eğim Durumu: Bu kriteri değerlendirebilecek veriler elde edilemediği için, ayrıca emlak değeri üzerindeki etkisinin de az olduğu düşünüldüğünden değerlendirme dışında tutulmuştur.

14-Deniz, Nehir Kıyısına Yakınlık: Elazığ il merkezi deniz, göl ya da nehir kıyısına sahip olmadığı için bu faktör puanı hesaplamalar dışında tutulmuştur.

15-Manzara Durumu: Elde bulunan veriler itibariyle bu kriteri değerlendirme imkânına sahip olunmadığından ‘manzara durumu’ faktörü değerlendirme dışında tutulmuştur.

16-Sağlık Açısından Tehlikeli Bölgelere Yakınlık: Elazığ ili Merkez mahalleleri içerisinde sağlık açısından tehlikeli kabul edilebilecek nükleer santral, bataklık, çöp toplama alanı gibi bölgeler bulunmadığı için bu kriter de değerlendirmeye alınmamıştır.

17-Kamu Hizmetlerinden İstifade Durumu: Bu kriter içeriğinde yer alan; yol, su, elektrik, kanalizasyon, telefon, doğalgaz gibi kamu hizmetlerinden faydalanma durumu göz önüne alındığında, Elazığ il merkezinde bu kriterin de değerlendirilmesine gerek görülmemiştir.

4.5. Emlak Değerine Etki Eden Faktör Ağırlıklarının Belirlenmesi

Emlak değerine etki eden faktör ağırlıklarının belirlenmesinde Harmoni Arama Algoritması kullanılmasına karar verilmiştir. Harmoni Arama Algoritması, Yapay Zekâ tekniklerine dayalı bir Sezgisel (Heuristik- Stokastik) Optimizasyon Algoritmasıdır.

Yapay Zekâ, insana özgü nitelikler olan akıl yürütme, anlam çıkarma, genelleme, geçmiş deneyimlerden öğrenme gibi zihinsel faaliyetleri kullanarak bir bilgisayarın arzu edilen işi yerine getirmesidir [62].

Optimizasyon, bir sistem veya tasarım için en iyi sonucu bulmak amacı ile yapılır. Aynı zamanda, minimum maliyet, minimum hata, maksimum kar ve maksimum fayda sonuçlarından birini ya da birkaçını sağlamak amacına yönelik olabilir.

Yapay Zekâ Optimizasyon Algoritmaları; bir sistemi en iyi sonucu bulmaya yönelik hazırlarken, matematiksel veya sezgisel (heuristik) tekniklerden faydalanır. Bu nedenle Optimizasyon Algoritmaları;

- Matematiksel Optimizasyon Algoritmaları
- Sezgisel Optimizasyon Algoritmaları

olmak üzere ikiye ayrılır.

4.5.1. Sezgisel (Heuristik) Algoritmalar

Problemlerin çözümü esnasında sezgisel yaklaşımların kullanıldığı algoritmalarlardır. Bu yaklaşımlar gerçek hayatta daima kullanılırlar. Örneğin; bir yerden başka bir yere giderken, takip edilen yolun nereye çıkacağını hiç bilmeden, yol ayrımlarında sezgilere dayalı olarak bir seçim yapmak bu tarz bir yaklaşımdır.

Sezgisel Algoritmalar, Genetik Algoritmalar da olduğu gibi esasen çalışmalarında kesinlik bulunmayan algoritmalarlardır. Optimum duruma ulaşmak için rastgele değerler üretirler [63]. Değerler rastlantısal olarak oluşturulduğundan, her zaman aynı performansı göstermezler ve her zaman aynı sonucu vermeyi garanti etmezler. Buna rağmen optimizasyon problemleri için çok kullanışlı yöntemlerdir [64]. Genellikle en iyiye yakın çözüme hızlı ve kolay bir şekilde ulaşırlar [65].

Non-linear algoritmalarından farklı olarak, sezgisel (stokastik) arama algoritmalarına, ayrık değişkenler içeren optimizasyon problemleri için başvurulur [63].

Sezgisel Algoritmaların uygulanan yöntemin doğruluğunu ispatlamaları beklenmemektedir. Sonuçta üretilen değerlerin tatmin edici doğrulukta olmaları ya da karmaşık problemleri basite indirgemeleri yeterli görülmektedir.

Bilgisayar bilimlerinde bir problemin çözümü genel olarak, zamandan kazanç ya da doğruluk amaçlarından birini gerçekleştirmeye yöneliktir. Birinci durumda, problemin çözümü için gerçekleştirilen adım sayısı (iterasyon sayısı) az olmalı ve hızlı gerçekleşmelidir. Fakat bu durumda doğru sonuca ulaşma garanti edilmemektedir. İkinci durumda ise, programın kabul edilebilir sonuçlar elde etmesi istenmektedir. Bu durumda da çözümün hızlı olması garanti edilmemektedir. Sezgisel Algoritmalarla genellikle bu iki amaçtan biri göz ardı edilmektedir. Ya da bu tez çalışmasında yapıldığı gibi iki durum için de makul sayılabilecek değerler sonlandırma şartı olarak eklenebilir [64].

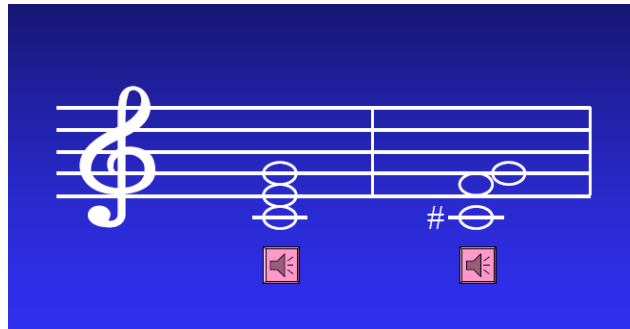
Sezgisel Algoritmalarından bazıları şunlardır:

- Genetik Algoritma (Genetic Algorithm)
- Diferansiyel Gelişim Algoritması (Differential Evolution Algorithm)
- Tabu Araştırma Algoritması (Tabu Search Algorithm)
- Yapay Bağışıklık Sistemi (Klonal Seçim Algoritması - Clonal Selection Algorithm)
- Karınca Koloni Algoritması (Ant Colony Algorithm)
- Arı Koloni Algoritması (Bee Colony Algorithm)
- Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
- Dağılım Tahmin Algoritması (Estimation of Distribution Algorithm)
- Parçacık Çarpışma Algoritması (Particle Collision Algorithm)
- Elektromanyetizma Algoritması (Electromagnetism-like Algorithm)
- Sıçrayan Kurbağa Algoritması (Leap-Frog Algorithm)
- A* Arama Algoritması (A* Search Algorithm)
- En İyi Öncelikli Arama Algoritması (Best First Search Algorithm)
- Açgözlü En İyi Öncelikli Arama Algoritması (Greedy Best First Search Algorithm)
- Tırmanış Arama Algoritması (Hill Climbing Algorithm)
- Fidan Gelişim Algoritması (Saplings Growing up Algorithm)
- Kültürel Algoritma (Cultural Algorithm)

4.5.2. Harmoni Arama Algoritması

Harmoni Arama Algoritması (Harmony Search Algorithm), Sezgisel (Heuristik) Optimizasyon Algoritmalarından biridir. Geem vd. [66-67] tarafından stokastik (sezgisel) arama teknikleri grubuna dâhil olarak geliştirilmiştir. Doğadan esinlenerek, müzik olgusundan ortaya çıkmıştır. En iyi harmoniyi (ahengi) arama düşüncesine dayandırılmaktadır.

Harmoni Arama Algoritması, var olan evrimsel ve sezgisel algoritmalarda olduğu gibi, doğal ya da davranışsal olay ve olgulara benzetilerek oluşturulan bir algoritmadır. Müzik geliştirme sürecinde aranan en iyi durumun (fantastik harmoni) estetik düşünceyle belirlenmesi gibi, Harmoni Arama Algoritmasına göre, optimizasyon sürecinde aranan en iyi durum (global optimum) da, amaç fonksiyonunun değeriyle belirlenir. Tıpkı, estetik ses kalitesinin çalma düzeyine (tonuna) göre gruplanmış enstrümanlar tarafından devamlı pratik yaparak geliştirilebileceği gibi, Harmoni Arama Algoritmasına göre, amaç fonksiyonunun değeri de devamlı iterasyonlarla geliştirilebilir.



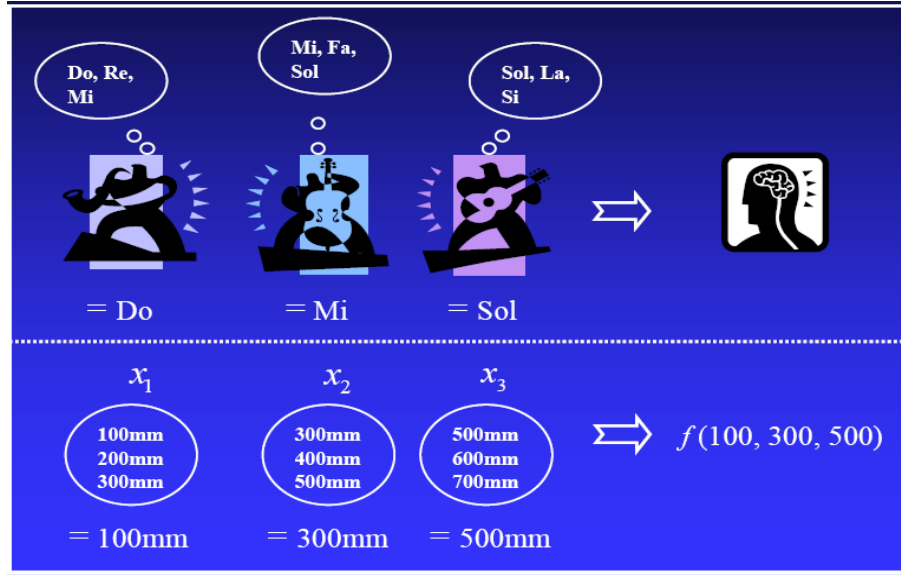
Şekil 4.4. Müzikte harmoniyi (uyumu) arama [68].

Daha çok, ahengi ve uyumu koruyucu bir algoritmadır. Benzerlikler, ahengi oluşturur. Yukarıdaki şekilde ilk nota iyi harmoniyi (ahengi), ikinci nota ise kötü harmoniyi (uyumsuzluğu) temsil eder. Müzik geliştirmedeki harmoniyi sağlama ile optimizasyon arasında büyük benzerlik vardır [68].

Aşağıdaki tabloda (Tablo 4.1.) bilgisayar sistemleri için bir optimizasyon problemi ile müzik için harmoni(ahenk) arama problemi karşılaştırılmaktadır.

Tablo 4.1. Bilgisayar sistemlerindeki optimizasyon problemi ile müzikte harmoniyi (uyumu) arama probleminin karşılaştırılması [66,68].

Karşılaştırma Faktörü	Optimizasyon Yöntemi	Performans Yöntemi
İşlem Birimi	Her bir iterasyon için	Her bir eylem için
En İyi Durum	Global Optimum	Fantastik(Mükemmel) Harmoni
Değerlendirme	Amaç Fonksiyonu	Estetik Standartlar
Diğer Değerlendirmeler	Değişkenlerin Değerleri	Çalgıların Tonları



Şekil 4.5. Müzikte Harmoni Arama yaklaşımına bir örnek [67].

Harmoni Arama yaklaşımına göre; her müzik geliştirici, her bir karar değeri (x_1 , x_2 , x_3) için uyumu göz önünde bulundurur. Sırasıyla her bir müzik aleti (Saksafon (Do,Re,Mi), Kontrabass (Mi,Fa,Sol), Gitar (Sol,La,Si)), her bir değişken değeri dizisiyle ($x_1=(100,200,300)$, $x_2=(300,400,500)$, $x_3=(500,600,700)$) uyumludur.

Şekil 4.5.'de görüldüğü gibi, eğer saksofonist Do'yu, kontrabass çalan Mi'yi, gitarist Sol'ü çalarsa, yeni bir harmoni (Do,Mi,Sol) oluşur ve eğer bu yeni harmoni var olandan daha iyiye, yeni harmoni saklanır. Aynı şekilde, optimizasyon sürecindeki yeni çözüm vektörü (100,300,500), amaç fonksiyonuna uygulanarak elde edilmiş olan önceki çözüm vektöründen

daha iyiye, harmoni hafızasında (harmony memory) korunur. Harmoni kalitesi devamlı pratikle arttığı gibi, aynı şekilde çözüm kalitesi de sürekli iterasyonlarla artar [67].

Harmoni Arama Algoritması, diğer optimizasyon tekniklerine göre problemin çözümü için daha az matematiksel ifadeye ihtiyaç duyar. Ayrıca diğer sezgisel arama yöntemlerine benzer şekilde, optimizasyon problemleri için aday çözümleri ayrık veya devamlı veri setlerinden rastgele seçer. Sonra bu çözümün uygun olup olmadığı kontrol edilir. Eğer uygun ise, HMCR (Harmoni Hafızasının Düşünülen Oranı), PAR (Düzey Ayarlama Oranı) ve RS (Rastgele Seçim) olmak üzere üç parametreye bağlı olarak seçilen yeni çözüm, her aday çözümün tutulduğu Harmoni Hafızasına yerleştirilir ve algoritma bu şekilde devam eder.

Matematiksel programlama teknikleriyle karşılaştırıldığında; Harmoni Arama Algoritmasının daha basit bir yaklaşım olduğu görülmektedir. Bu algoritma, karar değişkenleri için başlangıç değerlerine ihtiyaç duymaz. Ayrıca amaç fonksiyonu ve sınırlamalar için de türetici bilgiye ihtiyaç duymaz. Bu nedenle, karmaşık optimizasyon problemleri için kolay programlama sağlar [63].

Harmoni Arama Algoritması; müzik harmonisinde, sudoku problemlerinde, ağ sistemlerinde, şehir boru hatlarının kapasite dizaynında, kısa yol algoritmalarında, matematikte minimizasyon problemlerinde ve endüstride çeşitli alanlarda kullanılabilmektedir.

4.5.2.1. Algoritmada Kullanılan Değişkenler

HM (Harmony Memory): Harmoni Hafızası

HMS (Harmony Memory Size): Harmoni Hafızasının Boyutu

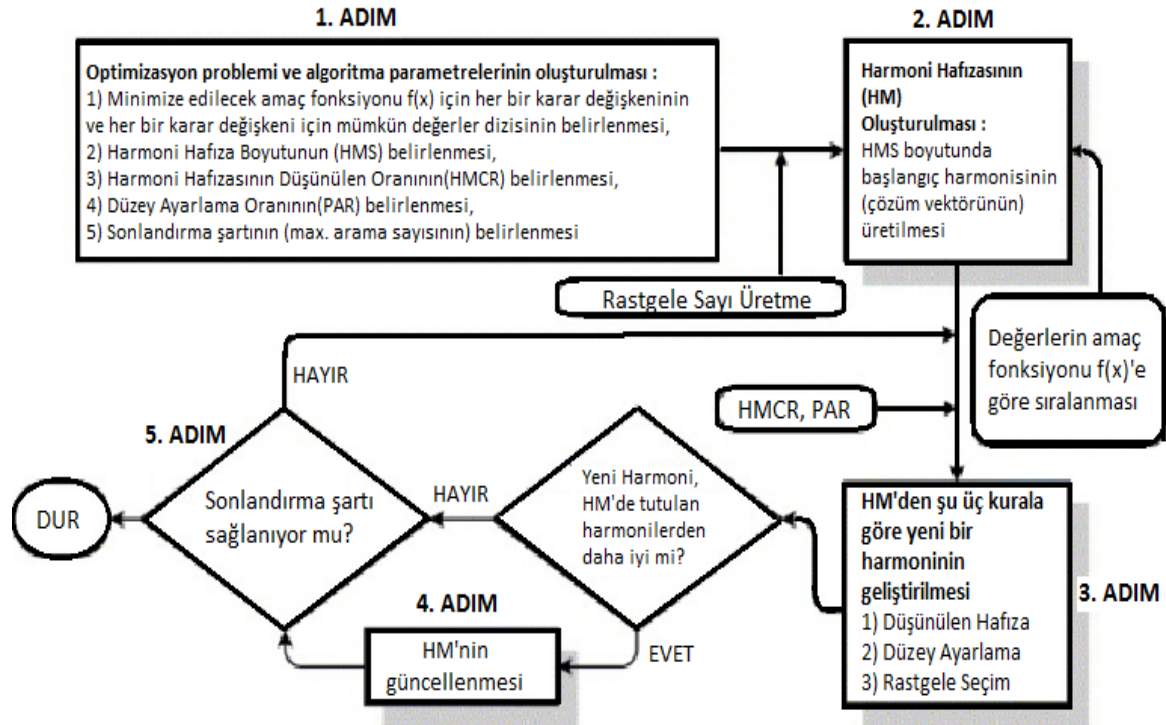
HMCR (Harmony Memory Considering Rate): Harmoni Hafızasının Düşünülen Oranı

RS (Random Selection): Rastgele Seçim

PAR (Pitch Adjustment Rate): Ton/Düzey Ayarlama Oranı

m: PAR uygulamasında yardımcı değişken

4.5.2.2. Harmoni Arama Algoritması İşlem Adımları



Şekil 4.6. Harmoni Arama Algoritması işlem adımları [67,69].

Şekil 4.6.'daki algoritma konseptine uygun olarak, Harmoni Arama (Harmony Search) Algoritması aşağıdaki beş adımdan oluşur.

- 1- Parametrelerin oluşturulması
- 2- Harmoni Hafızasının (Harmony Memory) oluşturulması
- 3- Yeni harmoni geliştirilmesi
- 4- Harmoni hafızasının güncellenmesi
- 5- Bitirme kriteri kontrolü [70].

1- Parametrelerin Oluşturulması: İlk adımda optimizasyon problemi aşağıdaki şekilde belirlenir.

$f(\cdot)$: Amaç fonksiyonu,

X_{HMS} : Her biri karar değişkeni olan x_i 'lerin birleşiminden oluşan çözüm vektörü,

x_i : Her bir karar değişkeni için mümkün değerler dizisi olmak üzere;

- Minimize ya da maksimize $f(x)$
- $x_i \in X_{HMS}$, $i=1,2,3,...,N$ parametreleri belirlenir.

2- Harmoni Hafızasının Oluşturulması: Harmoni Hafızası (HM), içinde çözüm vektörü bulunan iki boyutlu matristir. Rastgele oluşturulan çözüm vektörleriyle, Harmoni Hafızası büyüklüğünde oluşturulur.

$$\left[\begin{array}{ccccc|c} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 & f(x^2) \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \cdots & x_{N-1}^{HMS-1} & x_N^{HMS-1} & f(x^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_{N-1}^{HMS} & x_N^{HMS} & f(x^{HMS}) \end{array} \right] \quad (4.4)$$

3- Yeni Harmoni Belirlenmesi: Yeni harmoni vektörü üç kurala göre belirlenir.

- Rastgele Seçim (Random Selection)
- Harmoni Hafızasının Dikkate Alınma Oranı (Harmony Memory Consideration Rate)
- Ton/Düzey Ayarlama Oranı (Pitch Adjustment Rate)

a) Rastgele Seçim (Random Selection): Bir müzisyen, enstrümanı sırasıyla bir tonda çaldığı zaman (Örneğin; {Do,Re,Mi,Fa,Sol,La,Si} gibi (Şekil 4.7.)), x_i karar değişkeninin değeri X_{HMS} dizisi içinden rastgele seçilir.



Şekil 4.7. Müzik enstrümanının notaları sırasıyla çalması [64].

$x_i \in$ Çalınabilir Dizi (Playable Range) olmak üzere; Çalınabilir Dizi, x_i ' nin alabileceği tüm değerleri içeren değerler kümesidir ve $RS = (1 - HMCR)$ değerini alır.

b) Harmoni Hafızasının Dikkate Alınma Oranı (Harmony Memory Consideration

Rate): Bir müzisyen, hafızasındaki tercih edilen tonların dışında bir tonda çaldığı zaman (Örneğin; {Do,Mi,Do,Sol,Do} gibi (Şekil 4.8.)), Rastgele Seçim (Random Selection) sürecinin önceden tanımlandığı şekliyle, 1-HMCR oranında, yeni x_i karar değişkeninin değeri rastgele seçilirken; aynı şekilde HMCR (Harmoni Hafızasının Dikkate Alınma Oranı) ($0 \leq HMCR \leq 1$) oranında da, x_i karar değişkeninin değeri, Harmoni Hafızasında kayıtlı tonlar içerisinde seçilir. Yani HMCR, Harmoni Hafızasında bulunan çözüm vektörleri arasından seçim oranını; RS ise, random() fonksiyonu kullanılarak yeni çözüm vektörünü rastgele belirlenme oranını belirtir.

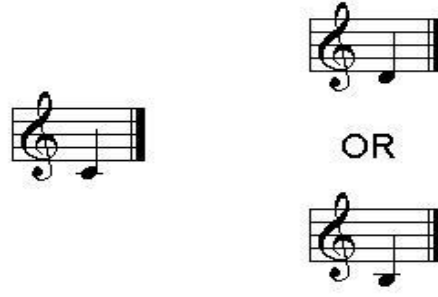


Şekil 4.8. Tercih edilen tonların Harmoni Hafızasında saklanması [64].

$$x'_i \leftarrow \begin{cases} x'_i \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{ise, } HMCR \\ x'_i \in X_i & \text{ise, } (1 - HMCR) \end{cases} \quad (4.5)$$

Hafızadaki değerler dikkate alınarak, uyumlu olduğu önceden tespit edilmiş değerler üzerinden işlem yapılır. ($0 < HMCR < 1$ 'dir.)

c) Ton/Düzey Ayarlama Oranı (Pitch Adjustment Rate): x notasının bir üstünde veya bir altındaki notalar, x notasına uyumlu olacağı için yeni harmoni elde etmede kullanılabilirler.



Şekil 4.9. $x+$ veya $x-$, $x \in$ Tercih edilen Notalar [64].

Optimizasyon algoritması içinde, PAR(Pitch Adjustment Rate), mümkün değerler dizisi içinden bir değeri komşu değere kaydırma işlemidir. Eğer (1,2,4,5,6,8) şeklinde altı tane olası değer varsa, (5) komşusu olan (4) ya da (6)'ya taşınabilir [63].

$$x'_i \leftarrow \begin{cases} x_i(k+m) & , \quad PAR \\ x'_i & , \quad (1- PAR) \end{cases} \quad (4.6)$$

Bu durum, Harmoni Hafızasındaki en uyumsuz vektörün rastgele seçilen değerine, $\pm m$ değerinin eklenmesi ile gerçekleştirilir. Ton ayarlama olasılığı varsa PAR, yoksa (1-PAR) kullanılır.

$0 < PAR < 1$ 'dir. (Tercihen $0.3 < PAR < 0.99$ aralığında alınır.)

$m = \dots -3, -2, -1, 0, 1, 2, 3 \dots$ değerlerini alabilir. (Tercihen $m = -1$ veya $m = +1$ alınır.)

Ayrık değerler için yeni harmoni oluşturulurken, aşağıdaki formül kullanılırken,

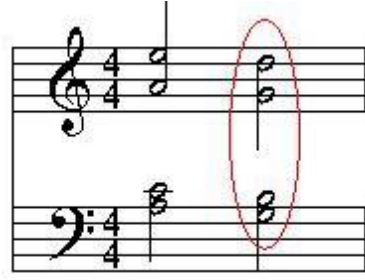
$$x'_i \leftarrow x_i(k+m) \quad , \quad HMCR \quad (4.7)$$

sürekli değerler için eski harmoni değerine, α değeri eklenir.

$$x'_i \leftarrow x'_i + \alpha, \quad (4.8)$$

Burada α , $bw \times u(-1, 1)$ değeridir. bw , devamlı değişkenler için keyfi bant genişliği mesafesidir ve u , -1 ve 1 arasında düzenli bir şekilde dağılır.

Harmoniyi Bozanı Dikkate Alma (Violated Harmony Consideration): Bu teknik, çalışmalarda sınırlayıcıları göz önünde tutmak için kullanılır. Yukarıda bahsedilen üç kural kullanılarak, yeni bir harmoni elde edildiğinde; kısıtlamaların(sınırlayıcıların) ihlal edilebileceği görülür. Bu durumda, tıpkı kural bozucu harmoninin Ludwig Van Beethoven gibi ünlü besteciler tarafından bestecilikte kullanıldığı gibi (Şekil 4.10.); yeni harmoni de, sınırlamalara uymasa bile, cezaya zorlayıcı olarak Harmoni Hafızasında korunabilir.



Şekil 4.10. Müzikte harmoniyi bozan durumlar [68].

- 4- **Harmoni Hafızasının Güncellenmesi:** Eğer Rastgele Seçim (RS), Harmoni Hafızasının Dikkate Alınma Oranı (HMCR) ve Düzey Ayarlama Oranı (PAR) gibi yöntemler ile oluşturulan yeni harmoni vektörü,

$$\mathbf{x}' = (x'_1, x'_2, \dots, x'_N) \quad (4.9)$$

Harmoni Hafızasındaki en kötü durumdan daha iyiye; bu kötü durum, Harmoni Hafızasından çıkartılır ve yeni harmoni, Harmoni Hafızasına eklenir.

- 5- **Bitirme Kriteri Kontrolü:** Eğer sonlandırma kriterine ulaşılmışsa, yürütme durdurulur. Aksi halde, 3. ve 4. adım tekrarlanır. Optimizasyon algoritmalarında sonlandırma şartı genellikle; ya belirli bir iterasyon sayısına ulaşması ya da belirli bir hata oranının altına düşmesi olarak belirlenir [67,69,71,72].

4.5.2.3. Harmoni Arama Algoritmasıyla Örnek Bir Uygulama

$f(x) = 100 (x_1^2 - x_2)^2 + (1 - x_1)^2$ ve $-2.048 < x_1, x_2 < 2.048$ olmak üzere;

$f(x)$ fonksiyonunu minimize eden x_1 ve x_2 değerlerini bulmak için Harmoni Arama Algoritması kullanılmıştır. Bu fonksiyon için; $f(x)$ 'in minimize olması demek, sonucun sıfıra yaklaşması demektir.

Harmoni Arama Algoritmasının işlem adımları dikkate alınarak, öncelikle Harmoni Arama parametreleri belirlenmiştir.

- HMS (Harmoni Hafıza Boyutu) = 100

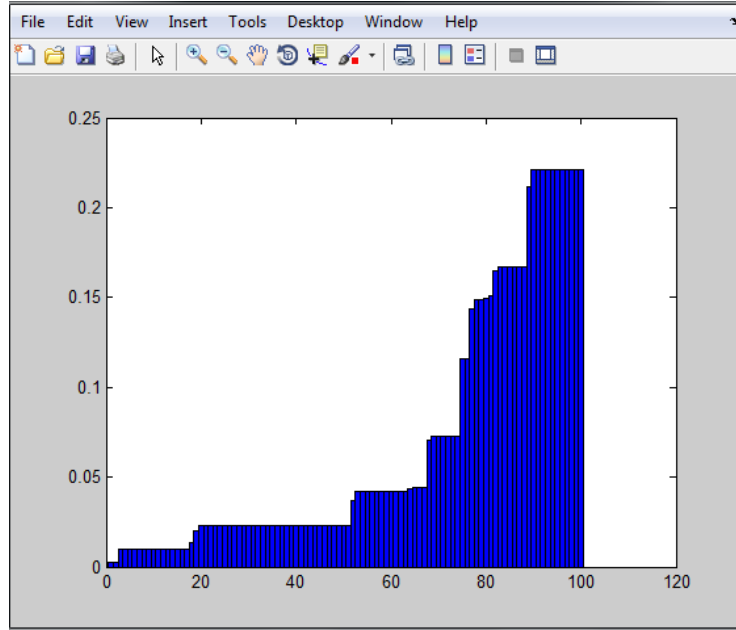
(Harmoni Hafızası, x_1 , x_2 ve $f(x)$ değerlerini tutmak için (100x3) 'lük bir matris olarak oluşturulmuştur.)

- $P_{\text{Random}} (\text{RS}) = 0.49$
- $P_{\text{Memory}} (\text{HMCR}) = 0.30$
- $P_{\text{Pitch}} (\text{PAR}) = 0.21$
- $m = 1$ ($m = +1$ ve $m = -1$)

İlk olarak Harmoni Hafızası -2.048 ve +2.048 arasında rastgele değerlerle oluşturulmuştur. Karar değişkenlerinin değerine göre amaç fonksiyonunun değeri hesaplanmıştır. Ardından Harmoni Hafızası, amaç fonksiyonunun aldığı değerlere göre küçükten büyüğe doğru sıralanmıştır. Bu problem bir minimizasyon problemi olduğu için, en büyük amaç fonksiyonu değerine sahip çözüm vektörü x_{old} (yani Harmoni Hafızasından çıkarılacak değer) olarak belirlenmiştir. Sonrasında Harmoni Hafızasından çıkarılacak değerın yerine, eklenecek yeni harmoniyi belirlemek için HMCR, PAR ve RS oranları kullanılmıştır. HMCR (Harmoni Hafızasının Dikkate Alınma Oranı), PAR (Düzey Ayarlama Oranı) ve RS (Rastgele Seçim) oranlarına göre, yeni harmoni (x_{new}) belirlenmiştir. HMCR oranında, Harmoni Hafızasında kayıtlı değerler arasından bir karar değişkeninin değeri x_{new} olarak belirlenir. Benzer şekilde, RS oranında, x_{new} için bir karar değişkeninin değeri rastgele (random) olarak belirlenir. Son olarak, PAR oranında, x_{new} için bir karar değişkeninin değeri, x_{old} değerine $\pm m$ eklenmesiyle belirlenir.

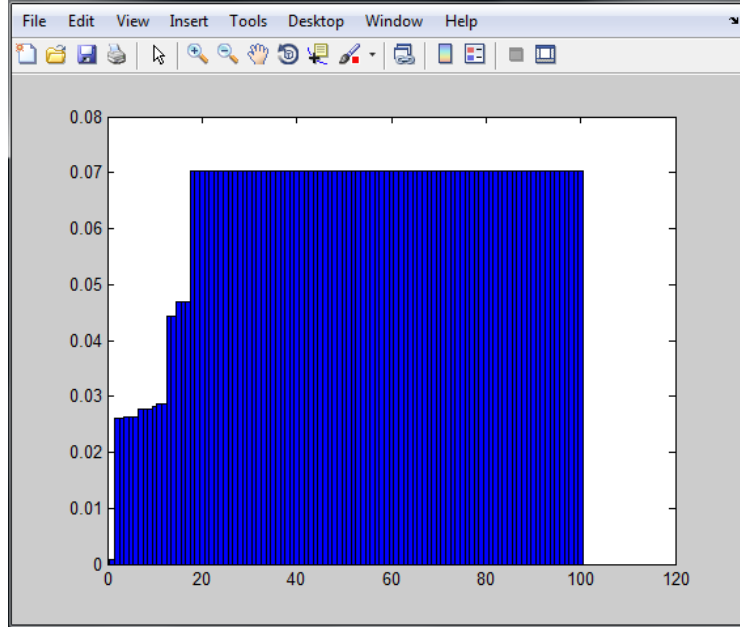
Elde edilen x_{new} karar değişkenlerinin $-2.048 < x_1, x_2 < 2.048$ aralığında olup olmadıkları kontrol edilir. Ardından bu değer seti ile $f(x)$ amaç fonksiyonunun değeri hesaplanır. x_{new} değeri x_{old} değerinden daha iyiye (bu problem için x_{new} değer vektörüyle hesaplanan $f(x)$ amaç fonksiyonunun değeri, x_{old} değer vektörüyle hesaplanan amaç fonksiyonunun değerinden daha küçükse) x_{old} 'un yerine Harmoni Hafızasına x_{new} eklenir ve Harmoni Hafızası (HM) amaç fonksiyonu değerlerine göre yeniden sıralanır.

Bu algoritma, 1000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0025 ile 0.2210 arasında değerler elde etmiştir (Şekil 4.11.). 2000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0054 ile 0.2329 arasında değerler elde etmiştir. 3000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0189 ile 0.2571 arasında değerler elde etmiştir. İterasyon sayısı arttıkça doğruluk oranının azaldığı gözlenmiştir.



Şekil 4.11. $m=1$ ve iterasyon sayısı 1000 iken $f(x_i)$ grafiği

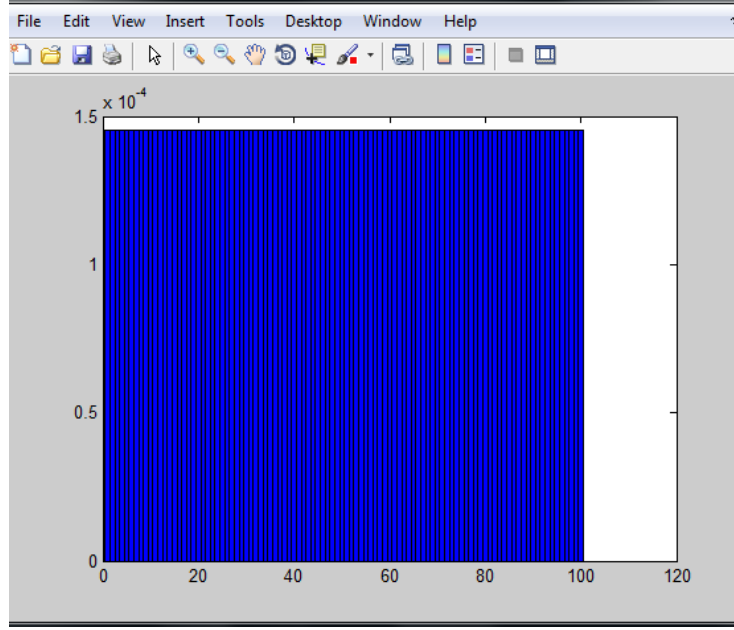
$m=0.1$ olarak değiştirildiğinde ise, algoritma 1000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0087 ile 0.5856 arasında değerler elde etmiştir. 2000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0068 ile 0.3812 arasında değerler elde etmiştir. 3000 iterasyonda $f(x)$ amaç fonksiyonu için 0.0009 ile 0.0704 arasında değerler elde etmiştir (Şekil 4.12.). İterasyon sayısı arttıkça doğruluk oranının da arttığı gözlenmiştir.



Şekil 4.12. $m=0.1$ ve iterasyon sayısı 3000 iken $f(x_i)$ grafiği

Program her çalıştığında sonuçların tutulduğu Harmoni Hafızasının birbirinden farklı değerlere sahip olduğu ve programın %100 doğrulukta çalışmadığı, doğruya yakın değerler ürettiği gözlemlenmiştir. Bu da sezgisel (heuristik) algoritmaların, sonuca ulaşmada rastlantıya dayalı yöntemler kullanmalarından, sonuçların programın her çalışmasında birbirinden farklı değerler almasına yol açtığını ve sezgisel algoritmaların, sonuçların %100 doğruluğunu garanti etmediklerini göstermektedir.

Şekil 4.13., $m=0.1$ iken ve iterasyon sayısı 3000 iken, elde edilen çözüm matrislerinden başka birinin değerler grafiğidir. Bu grafik, sezgisel (heuristik) algoritmaların, sonuca ulaşmada rastlantıya dayalı yöntemler kullanmalarından, sonuçların programın her çalışmasında birbirinden farklı değerler almasına bir örnektir.



Şekil 4.13. $m=0.1$ ve iterasyon sayısı 3000 iken $f(x_i)$ grafiği

Şekilde de görüldüğü gibi, Harmoni Arama Algoritması, heuristik özellik gösteren ve bu özellikler dahilinde optimizasyon problemlerinde genellikle iyi sonuçlar veren bir algoritmadır.

4.5.2.4. Harmoni Arama Algoritması ile Faktör Ağırlıklarının Hesaplanması

Emlak Değer Haritası oluşturmak için, emlak değerine etki eden mevki, konum, alışveriş merkezlerine yakınlık, sosyal imkânlar yakınlık, okullara yakınlık, hastanelere yakınlık, iş imkânlarına yakınlık, ulaşım imkânlarına yakınlık, gürültüye yakınlık, yola yakınlık, zemin durumu ve parsel büyüklüğü olmak üzere 12 faktör belirlenmiştir. Bunların her biri için faktör puanları ve faktör ağırlıkları hesaplanmıştır. Faktör puanları, yukarıda bahsedildiği gibi Java koduyla ve gerekli SQL sorgulamalarıyla veri tabanındaki veriler kullanılarak her bina için hesaplanırken; emlak değerine etki eden faktörlerin ağırlıkları ise, faktör puanları kullanılarak, sezgisel (heuristik) bir algoritma olan Harmoni Arama Algoritması ile hesaplanmıştır.

$$\text{Emlak Değeri} = \left[FP_1 \times FA_1 + FP_2 \times FA_2 + FP_3 \times FA_3 + \dots + FP_n \times FA_n \right] \times m \quad (4.2)$$

formülü, emlak değerinin hesaplanmasında esas fonksiyon olarak belirlenmiştir. Birim arsa değeri $m=1000$ olarak kabul edilmiştir.

Harmoni Arama Algoritmasının işlem adımları dikkate alınarak, öncelikle Harmoni Arama parametreleri belirlenmiştir.

- HMS (Harmoni Hafıza Boyutu) = 100

Harmoni Hafızası, $FA_1, FA_2, \dots, FA_{12}$ ve $f(x)$ değerlerini tutmak için $x[100 \times 13]$ 'lük bir matris olarak oluşturulmuştur. Değeri bilinen 100 emlak için ilgili faktör puanları $FP_1, FP_2, \dots, FP_{12}$ ve emlağın değeri de yine $y[100 \times 13]$ 'lük bir matriste tutulmuştur.

- $P_{\text{Random}}(\text{RS}) = 0.49$
- $P_{\text{Memory}}(\text{HMCR}) = 0.30$
- $P_{\text{Pitch}}(\text{PAR}) = 0.21$
- $m = 0,1$ ($m = +0,1$ ve $m = -0,1$)

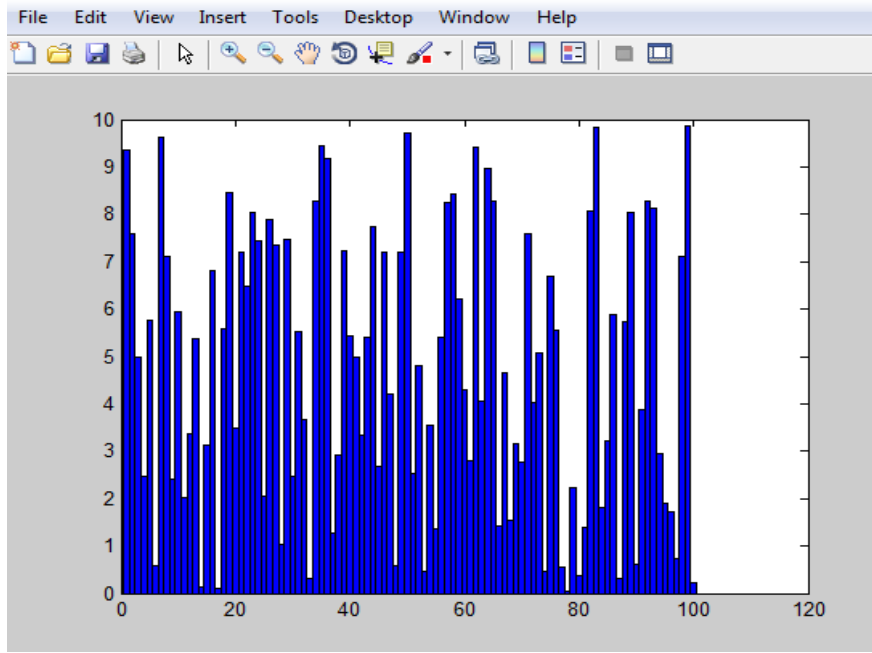
İlk olarak Harmoni Hafızası, 0-4 arasında rastgele değerlerle oluşturulmuştur. Harmoni Hafızasını oluşturan karar değişkenlerinin değerine göre, amaç fonksiyonunun değeri hesaplanmıştır. Ardından Harmoni Hafızası, amaç fonksiyonunun aldığı değerlere göre küçükten büyüğe doğru sıralanmıştır. y matrisinin son sütununda tutulan gerçek emlak değerlerine göre, sonuçtan en uzak amaç fonksiyonu değerine sahip çözüm vektörü x_{old} (yani Harmoni Hafızasından çıkarılacak değer) olarak belirlenmiştir. Sonrasında Harmoni Hafızasından çıkarılacak değer yerine, eklenecek yeni harmoniyi belirlemek için HMCR, PAR ve RS oranları kullanılmıştır. HMCR (Harmoni Hafızasının Dikkate Alınma Oranı), PAR (Düzye Ayarlama Oranı) ve RS (Rastgele Seçim) oranlarına göre, yeni harmoni (x_{new}) belirlenmiştir. HMCR oranında, Harmoni Hafızasında kayıtlı değerler arasından bir karar değişkeninin değeri x_{new} olarak belirlenir. Benzer şekilde, RS oranında, x_{new} için bir karar değişkeninin değeri rastgele (random) olarak belirlenir. Son olarak, PAR oranında, x_{new} için bir karar değişkeninin değeri, x_{old} değerine $\pm m$ eklenmesiyle belirlenir. Elde edilen x_{new} karar değişkenlerinin $0 < x_1, x_2, \dots, x_{12} < 4$ aralığında olup olmadıkları kontrol edilir. Ardından bu değer seti ile $f(x)$ amaç fonksiyonunun değeri hesaplanır. x_{new} değeri x_{old} değerinden daha iyiye (x_{new} değer vektörüyle hesaplanan $f(x)$ amaç fonksiyonunun değeri, x_{old} değer vektörüyle hesaplanan amaç fonksiyonunun değerinden daha çok gerçek emlak değerine yakınsa), x_{old} 'un yerine Harmoni Hafızasına x_{new} eklenir ve Harmoni Hafızası (HM) amaç fonksiyonu değerlerine göre yeniden sıralanır.

Bu algoritmanın sonlandırma şartı; ya hata oranının %10'un altına düşmesi ya da 10.000 iterasyon sayısına ulaşılması olarak belirlenmiştir.

Bu aşamaya kadar örnekteki Harmoni Arama Algoritmasında uygulanan yöntemle benzerlik göstermektedir. Yalnız örnekte tek bir denklemin çözümü yapılmaktayken, Emlak Değer Haritası uygulaması için 100 tane değeri bilinen emlak kullanıldığından, 100 farklı denklem oluşmaktadır. Bu problemin çözümü için şöyle bir yöntem benimsenmiştir: Her bir denklemin çözümü ile elde edilen (100x13)'lük matris içerisinde gerçek emlak değerine en yakın sonucu veren faktör ağırlıkları grubu en_dogru adında yeni bir matriste tutulmuştur. Böylece programın çalıştırılması sonucunda (100x13)'lük yeni bir matris elde edilmiş, bu matris alınarak, emlak_deger_hesapla() isimli yeni bir fonksiyonda kullanılmıştır. Bu fonksiyonda, her bir faktör ağırlığı grubu ile tüm emlak değerleri hesaplanmıştır ve tüm emlaklar için en iyi performansı gösteren faktör ağırlığı grubu, Emlak Değer Haritası uygulamasının faktör ağırlıkları olarak belirlenmiştir. Bu ağırlıklar, veri tabanında kayıtlı 37462 adet binanın emlak değerinin hesaplanmasında kullanılmıştır.

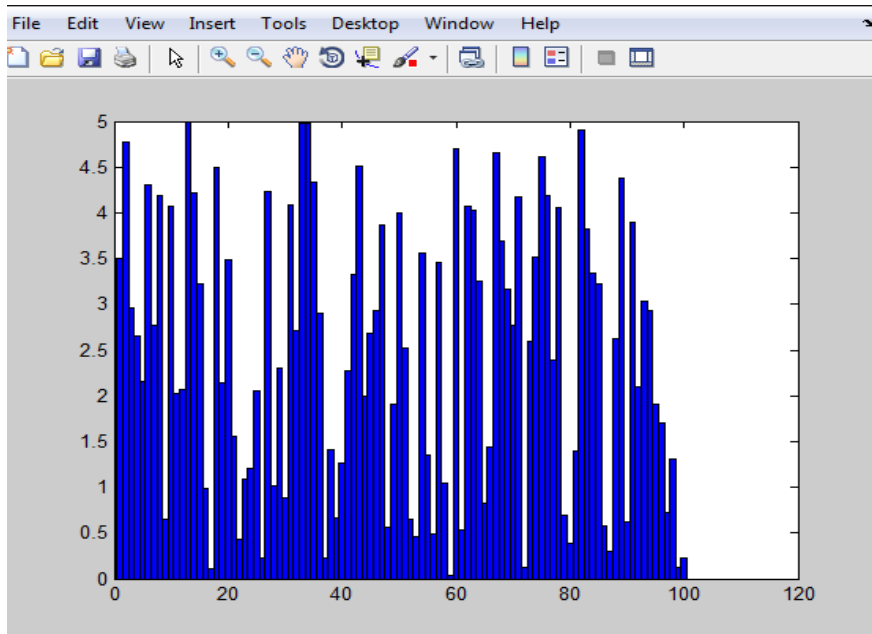
Tüm emlaklar için en az hataya sahip faktör ağırlığı setlerinden 2-3 veya 5 tanesi alınarak, ortalamalarıyla elde edilen değerler emlak değerinin hesaplanmasında kullanıldığında doğruluk oranında düşme gözlenmiştir. Bu nedenle ortalama bir ağırlık değeri bulmak yerine, en iyi faktör ağırlığı setinin alınması uygun görülmüştür.

Harmoni Arama Algoritmasının yürütülmesi sonucu elde edilen en_dogru matrisi %100 oranında istenen şartı (%10'dan az hatayı) sağlamaktadır. en_dogru matrisinin hata oranı Şekil 4.14.'de görülmektedir.



Şekil 4.14. Kabul edilebilir hata oranı %10 ve iterasyon sayısı 10.000 iken elde edilen grafik

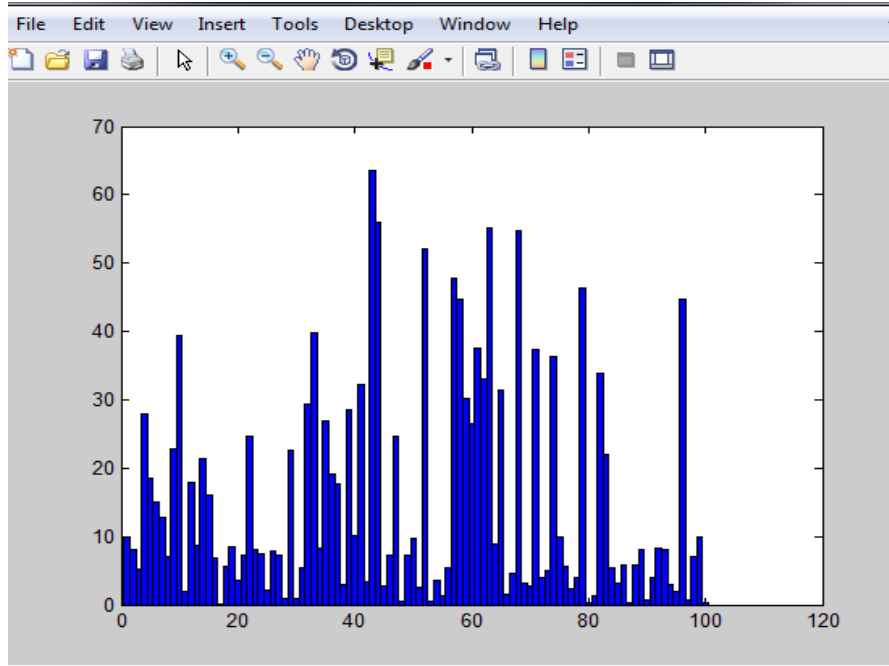
Hata oranı şartı %5 olarak belirlendiğinde ve iterasyon sayısı 20.000 olarak artırıldığında, yine %100 oranında (Şekil 4.15.) istenen şart sağlanmıştır. Bu da yazılan Harmoni Arama Algoritmasının yüksek performansla çalıştığını göstermektedir.



Şekil 4.15. Kabul edilebilir hata oranı %5 ve iterasyon sayısı 20.000 iken elde edilen grafik

Şekil 4.16.'da görüldüğü gibi, seçilen faktör ağırlığı seti 100 emlak içinde %63 oranında istenilen doğruluğa ulaşmıştır. Ayrıca ortalama hata oranı %15,1543 olarak hesaplanmıştır.

Kabul edilemez değer sayısı, bir kitlesel değerlendirme tekniğinin esas faydalılık parametresidir. Kontrimas'ın [57] çalışmasına göre; Litvanya'da bir nesnenin, kesin olarak %20'den daha çok reel değerinden farklı değerlendirilmesine, kabul edilemez değer denir. Bu da gösteriyor ki, bu tez çalışmasında elde edilen emlak değerlerinin gerçekliği kabul edilebilir sınırlar içerisinde ve iyi bir performansa sahiptir.



Şekil 4.16. Hesaplanan emlak değerlerinin, gerçek emlak değerlerinden yüzdesel farkını gösteren grafik

5. KULLANICI ARAYÜZ TASARIMINDA YARARLANILAN TEKNOLOJİLER

5.1. JSP

JSP (Java Server Pages) , dinamik web içeriği oluşturmak için kullanılan, sunucu ve platform bağımsız bir Java teknolojisidir. Web uygulamalarına basit ve hızlı bir yol sunar [73]. JSP, bir programlama dili değildir, bir web teknolojisidir. Web sayfalarını HTML, XML gibi dillerle oluşturur. HTML kodlarının arasına yazılır ve sunucu tarafındaki Java sınıfları (class) ile birlikte çalışır [74].

JSP, web uygulamaları geliştirmek ve dinamik içerik oluşturmak için kullanılan teknolojilerden en dikkat çekenidir. Diğer teknolojilerden farklı olarak çapraz platform ve çapraz uygulama sunucusu imkânlarını kullanmaktadır.

JSP sayfaları genellikle; statik HTML ve XML bileşenlerinden, özel JSP etiketlerinden ve opsiyonel olarak Scriptlet adı verilen Java programlama dili kod parçacıklarından oluşmaktadır.

JSP, spesifikasyonu Java Servlet API üzerine kurulmuş bir teknolojidir, fakat Servlet ve JSP arasında belirgin farklar bulunmaktadır. Uygulama geliştirmede programlama profesyonelliği gerektiren Servlet'lerin aksine, JSP daha geniş bir uygulama programcısı kitlesine hitap etmektedir. Sadece uygulama geliştiriciler tarafından değil, aynı zamanda web tasarımcıları tarafından da kullanılmaktadır. Böylelikle uygulama geliştirme yaşam döngüsünde doğrudan rol almaktadır. JSP teknolojisi ile sunulan içeriğin prezantasyon tarafından ayrılması hususu da, JSP'yi diğer teknolojilerden avantajlı hale getirmektedir. Çünkü JSP, Java Beans teknolojisi ve EJB teknolojisine dayalı yeniden kullanılabilir bileşenler içermektedir. Bunun yanı sıra, işlevsel JSP kodları yazıp test edebilmek için, TOMCAT uygulama sunucusu kullanılabilir [75].

5.1.1. JSP'nin Avantajları

JSP, statik ve dinamik içeriğin birbirinden ayrı tutulmasına imkân tanır. Bu yönüyle Servlet'e göre avantajlıdır. Servlet kullanımında, dinamik içeriğin oluşturulması Servlet'in kendi içindeki programlama mantığının bir parçasıdır. Bu dinamik içerik, kullanıcı arayüzünden sorumlu olan statik gösterim şablonlarına bağlıdır. Bundan dolayı kullanıcı arayüzü üzerinde yapılan çok ufak değişiklikler dahi, Servlet'in yeniden derlenmesini

gerektirmektedir. Dinamik ve statik içeriklerin birbirlerine bu şekilde bağlı olmaları, uygulama esnekliğine bir engel oluşturmaktadır. JSP ile dinamik içerik, statik içerikten JavaBean bileşenleri kullanılmak suretiyle ayrılmıştır. Dinamik içerik, daha sonra JSP sayfasında bazı özel script ve etiketler sayesinde oluşturulup kullanılabilir. Sonuçta, bir web tasarımcısı, sunum şablonu üzerinde herhangi bir değişiklik yaptığında, JSP sayfası otomatik olarak derlenmekte ve JSP motoru tarafından yine otomatik olarak web sunucusu içerisine konulmaktadır.

JSP, kodun bir kere yazılıp her platformda çalıştırılmasına imkân tanımaktadır. Platformlar ve web sunucuları arasında rahatlıkla taşınabilmekte ve herhangi bir değişikliğe ihtiyaç duymamaktadırlar.

JSP sayfası içerisindeki statik içeriğin herhangi bir düzende olma koşulu bulunmamaktadır. Böylece JSP, HTML/DHTML gibi standart tarayıcı düzenlerine, WML kullanan taşınabilir cihazlar ve cep telefonlarına veya XML kullanan diğer B2B uygulamalarına servis verebilmektedir.

JSP, Servlet'in soyutlaştırılmış halidir. Bu nedenle Servlet ile yapılabilen her şey, JSP ile de yapılabilir. Hatta JSP ile Servlet'e göre daha basit olarak gerçekleştirilebilmektedir [75].

5.2. JavaScript

JavaScript, web sayfalarında dinamik içerik sağlamak veya kullanıcı kontrolleri ekleyerek kullanıcıyla iletişim kurmak amacıyla kullanılan küçük kod parçaları olup, formları kontrol etmek, hesaplamalar yapmak gibi işlerde de kullanılırlar [76-77].

HTML sayfasının içine yerleştirilebilme özelliğine sahiptirler. Bunun yanında, tek başına .js uzantılı JavaScript kodları da yazılabilmektedir [77].

JavaScript elemanları kullanılarak kaliteli HTML sayfaları geliştirmek mümkün olmaktadır. Örneğin; kullanıcı hareketlerine anında yanıt verilebilir. Böylece kullanıcı ile karşılıklı etkileşim sağlanmış olur [78].

Şu anda HTML sayfalara gömülü yazılan scriptlerin çoğu JavaScript kullanılarak yazılmaktadır. Bu JavaScript kodları ile genellikle kullanıcı bilgilerinin doğru girilip girilmediğinin kontrolü yapılır. Ayrıca JavaScript ile kullanıcıya veri girişinde ve menü kullanımında kolaylık sağlanır [77].

5.3. Ext JS Kütüphanesi

Ext JS Kütüphanesi [6], AJAX [79], DHTML [80], DOM Scripting [81] gibi teknikler kullanılarak web uygulamaları geliştirilebilen istemci taraflı bir JavaScript kütüphanesidir. Ext JS, diğer sunucu taraflı (Php, Java, .Net) platformlara karşı çalışabilir. POST isteklerini işleyebilir ve yapısal veri döndürebilir. JSON ve XML formatlarındaki veriler Ext JS ile okunarak işlenebilir. Ext JS, jQuery ve Prototype ile birlikte de çalışabilir. Zengin internet uygulamaları geliştirilebilen güçlü bir kütüphanedir.

5.3.1. Ext JS' in Avantajları

- Tutarlı kodlarla basit, aynı zamanda görselliği iyi bir tasarım elde edilebilir.
- Ext JS nesne tabanlı konsepti destekler.
- Design pattern' i destekler.
- Plug-in özelliğini sağlar.
- Ext JS birçok bilinen web browser tarafından desteklenir [82].

5.4. AJAX

AJAX (Asenkron JavaScript ve XML), sayfanın yalnızca istenilen kısmının güncellenmesini sağlayan bir teknolojidir. AJAX, sayfanın güncellenmesi işini sunucuya istek göndermek ve sunucudan gelen cevabı işlemek şeklinde iki kısımda gerçekleştirir.

Sunucuya İstek Göndermek: İlk olarak “XML HTTP Request” bileşeni kullanılır. Daha sonra istek oluşturan nesneye verilecek cevabı işleyecek fonksiyon belirtilir. Bu, nesnenin ‘onreadystatechange’ özelliğine değer atanarak gerçekleştirilir. Bu adımdan sonra gerçek bir talepte bulunmak için open() ve send() metodları kullanılır. Open() metodunun ilk parametresi olan ‘GET’ veriyi almak için kullanılır. İkinci parametre ile URL adresi belirtilir. Üçüncü parametre (true) ise, isteğimizin eşzamanlı olup olmadığını gösterir (AJAX' ın asenkron olma özelliğiyle burada karşılaşılmaktadır.). send() metodu ise, sunucuya bilgilerin gönderilmesini sağlar.

Sunucudan Gelen Cevabı İşlemek: İsteği sunucuya doğru bir şekilde gönderdikten sonra yapılması gereken işlem, veriyi düzgün bir şekilde alabilmektir. Bunun için, önceki kısımda tanımlanan fonksiyona aşağıdaki kodlar yazılır.

```
if(http_istegi.readyState=4){  
    //yanıt alındı, her şey yolunda  
}  
else{  
    //henüz hazır değil  
}
```

Burada durum değerinin 4 olması, yanıt dönüşünün tamamlandığı anlamına gelir. Diğer durumlar:

- 0 - başlatılamadı
- 1 – yükleniyor
- 2 - yüklendi
- 3 - etkileşimli anlamlarına gelmektedir.

Bu adımdan sonra HTTP sunucu yanıtının durum kodu kontrol edilir. Sağlıklı bir yanıt gelmişse durum kodu 200 değerini alır. Aksi durumda (mesela 404 - sayfa bulunamadı) yanıtın ulaşmadığı anlaşılır.

İstek gönderilip yanıt alındıktan sonra, veriyi görüntülemek için `http_istegi.responseText` ya da `http_istegi.responseXML` fonksiyonları kullanılır.

Tüm bu işlemlerin bütünü bir AJAX uygulamasını meydana getirmektedir.

5.4.1. AJAX'ın Avantajları

- Ağ trafiğini azaltır.
- Kullanıcının gereksiz yere beklemesini önler.
- Önceden bilinen teknolojilerden oluşur.
- Web sayfaları için, Java ve Flash gibi karmaşık yapıdaki uygulamalardan daha pratik çözümler sunar [83].

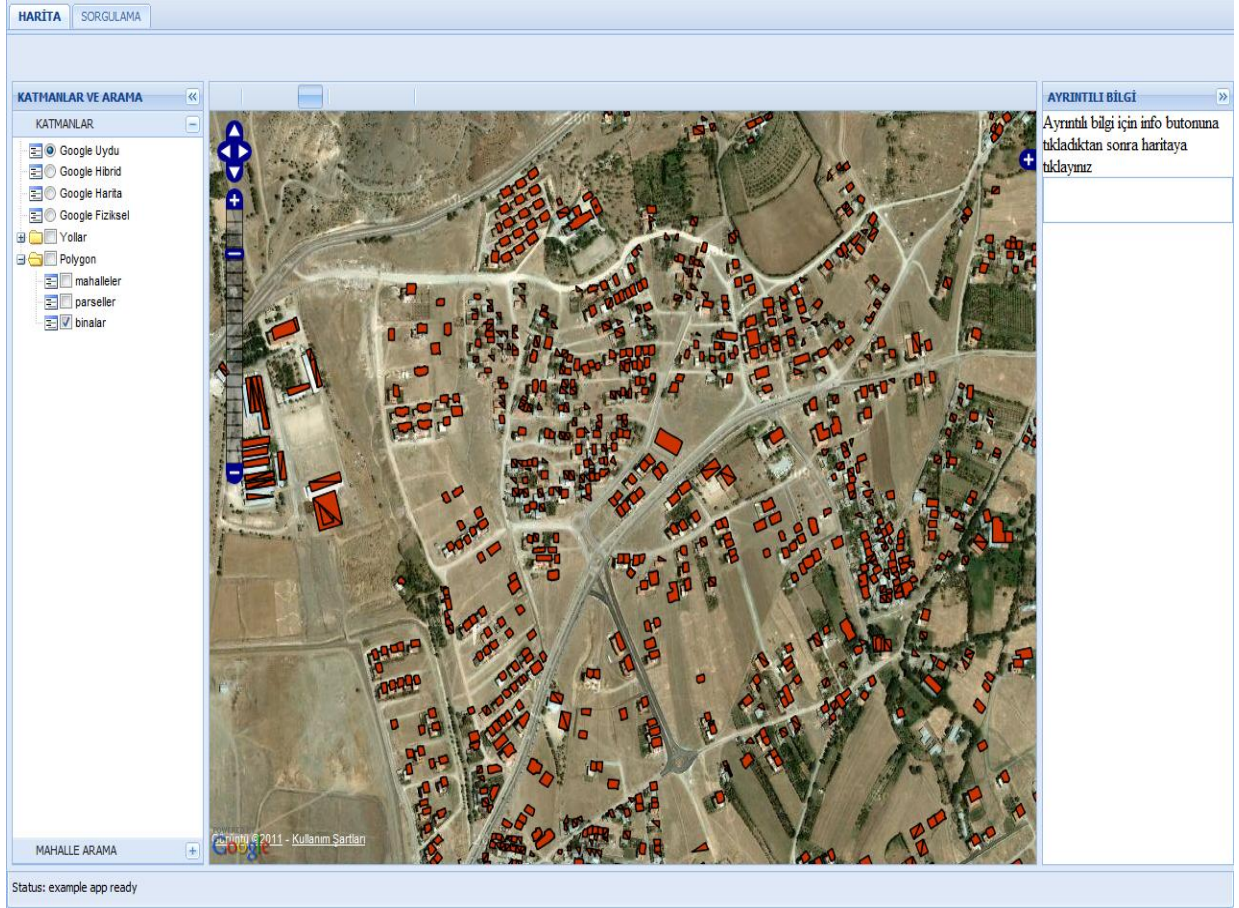
5.5. JSP, JavaScript, Ext JS ve AJAX Teknolojiler Kullanılarak Arayüz Tasarımı

Bu tez çalışmasıyla hesaplanan emlak değerleri, Java teknolojilerinden faydalanılarak, harita içerikli bir web arayüzü ile kullanıcıya sunulmuştur. Mekânsal Karar Destek Sistemleri uygulamaları için, heterojen verileri (mekânsal ve mekânsal olmayan verileri) işlemek ve doğrudan bu verileri kullanabilen birleşik bir arayüz sunmak önemlidir. Bu tez çalışmasında

bu arayüzü sağlamak için Ext JS Kütüphanesi kullanılmıştır. Ext JS Kütüphanesi yardımıyla hem haritalara ve hem de tablosal veriye dayalı bir kullanıcı arayüzü oluşturulmuştur.

Bu çalışmada, veri tabanında kayıtlı mekânsal verilerden oluşturulan GeoServer haritaları, JavaScript içerisinde OpenLayers [84] kullanımıyla web arayüzüne eklenmiştir. Bu verileri JSP ile web üzerinde görüntüleyebilmek için öncelikle projeksiyon ayarları düzenlenmiştir. Google haritalarını görüntüleyebilmek için EPSG:900913 projeksiyon ayarları, GeoServer’da tanımlı haritaları görüntüleyebilmek için ise EPSG:4326 projeksiyon ayarları kullanılmıştır. Haritanın sınır noktaları ve zoom seviyesi gibi bilgiler de yine OpenLayers kullanılarak belirlenmiştir ve harita üzerinde görüntülenmek istenen GeoServer katmanları tek tek eklenmiştir. GeoServer’da çekilen harita katmanlarının altında görüntülenmek üzere; Google Haritaları (Google Maps), Google Uydu Haritaları (Google Satellite Maps), Google Hibrit Haritaları (Google Hybrid Maps) ve Google Fiziksel Haritaları (Google Physical Maps) [85] da en alt katmanlara eklenmiştir. Sonuçta tüm emlak değer bilgileri, gerçek coğrafi harita üzerinden yansıtılmıştır. Bu şekilde, emlak değeri ile ilgili araştırma yapan kişiler tarafından tüm coğrafi etki faktörlerinin göz önünde bulundurulabilme olanağı sağlanmıştır.

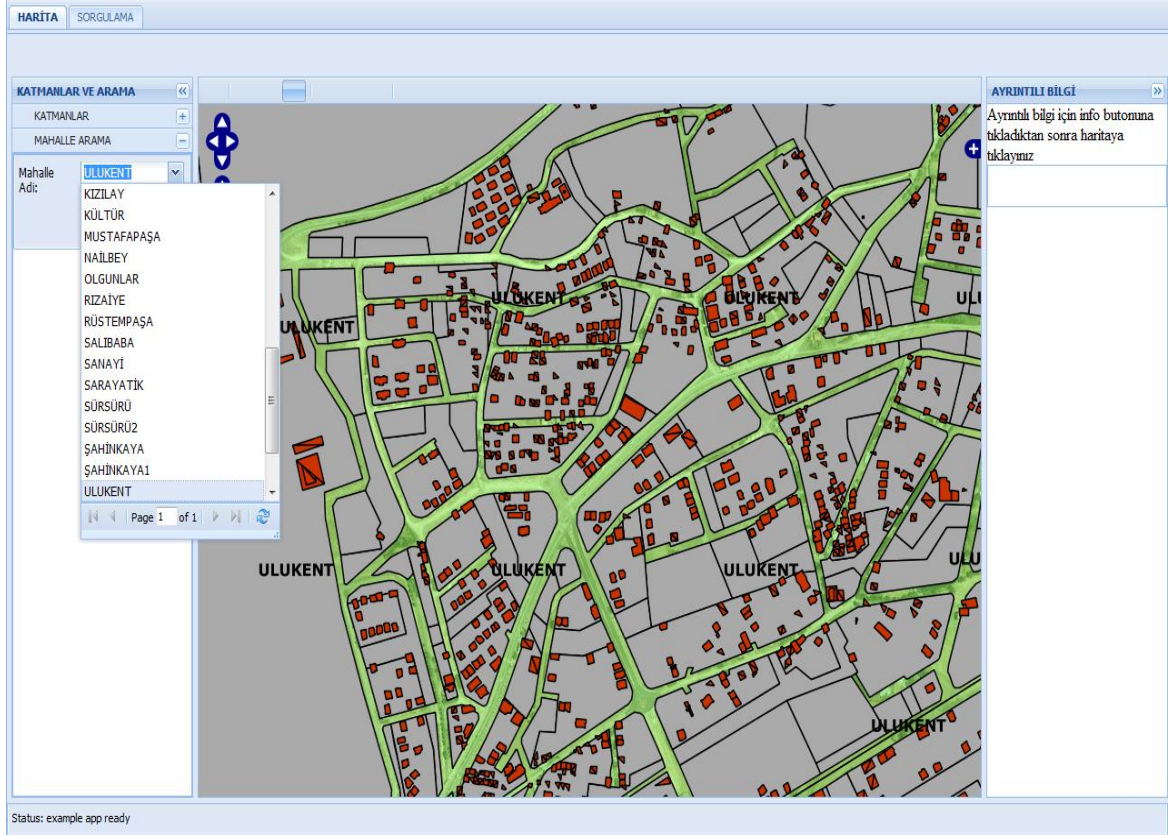
GeoServer’da katmanlar halinde tutulan mekânsal veriler, Ext JS Kütüphanesinden faydalanılarak, kullanıcı arayüzü Şekil 5.1.’de görüldüğü gibi istenen katmanın görüntülenip, istenen katmanın kaldırılmasına olanak tanıyan şekilde düzenlenmiştir.



Şekil 5.1. Emlak Değer Haritası kullanıcı arayüzü (1)

Ayrıca Ext JS Kütüphanesinin sunmuş olduğu harita araçlarından (tools) istifade edilmiştir. Bunun yanı sıra, AJAX istekleriyle JSP üzerinden Java sınıflarına erişim sağlanmış ve dinamik veri tabanı sorgulamalarına imkân tanınmıştır.

Uygulama içerisinde mahallelere göre arama imkânı sunulmaktadır. Seçilen mahallenin center() fonksiyonu ile merkez noktası hesaplanıp, Şekil 5.2.'de görüldüğü gibi belirlenen merkez noktasına otomatik olarak zoom yapılmaktadır.



Şekil 5.2. Emlak Değer Haritası kullanıcı arayüzü (2)

Bu tez çalışmasıyla hesaplanan emlak değerleri, Java teknolojilerinden faydalanılarak, harita içerikli bir web arayüzü ile kullanıcıya sunulmuştur.

6. SONUÇ

Bu tez çalışmasında mekânsal verilere dayalı web tabanlı bir Karar Destek Sistemi uygulaması olarak Emlak Değer Haritası oluşturulmuştur. Mekânsal veriler, yerel harita-kadastro bürolarından ve açık kaynak veri depolarından elde edilmiştir. Bu veriler coğrafi ilişkisel bir veri tabanı olan PostgreSQL ve ona coğrafi nitelik kazandıran eklentisi PostGIS kullanılarak depolanmıştır. PostGIS mekânsal verilerin daha az alan kullanarak saklanması için olanak tanır. Ayrıca coğrafi sorgulamalar için özel fonksiyonlar sağlar. Bu yönüyle mekânsal verilere dayalı Emlak Değer Haritası uygulaması için kullanışlı bir özellik göstermektedir.

Mekânsal verileri görüntülemek için, harita ve coğrafi bilgi sunucu olan GeoServer kullanılmıştır. GeoServer, OGC Standartlarına dayalı WMS, WFS, WCS sunucularını içinde barındırır. Emlak Değer Haritası uygulaması için veri tabanında kayıtlı mekânsal veriler (yollar, mahalleler, parseller, binalar) GeoServer'la haritalara dönüştürülmüştür. Aynı zamanda, GeoServer'ın desteklediği SLD kodu, harita üzerinde görüntülenen mekânsal verilerin renklendirilmesinde, belirli bir zoom seviyesinden sonra görünürlüklerinin sağlanmasında, haritanın zoom seviyesine göre veri boyutlarının belirlenmesinde ve mekânsal veriye bağlı nitelik verilerinin harita üzerinde etiketler şeklinde görüntülenmesinde kullanılmıştır. Sonuçta, GeoServer'ın Emlak Değer Haritası uygulaması için, mekânsal verilerin görüntülenmesinde uygun bir harita sunucu olduğu görülmüştür.

Emlak alıcı ve satıcılarına, emlakçılara ve diğer ilgililere karar alma süreçlerinde yardım sağlamak amacı ile Emlak Değer Haritası uygulaması içinde, Karar Destek Sistemlerine başvurulmuştur. Karar Destek Sistemi içerisindeki analiz süreci, kaba bilgiyi işlemeye ve emlak değerine etki eden kriterleri ve bu kriterlerin puan ve ağırlıklarını belirlemeye, böylelikle daha rasyonel (gerçekçi) emlak değerleri hesaplamaya imkân tanımıştır. Karar Destek Sistemi uygulamasında, değere etki eden faktörler belirlenerek, faktör puanları ile faktör ağırlıklarının çarpımına dayalı bir yöntem ile emlak değeri hesaplanmıştır. Faktör puanları her bina için veri tabanında kayıtlı mekânsal veri sorgulamalarıyla hesaplanmıştır. Faktör ağırlıkları ise, sezgisel bir algoritma olan Harmoni Algoritma Algoritması kullanılarak belirlenmiştir. Harmoni Arama Algoritmasıyla faktör ağırlıklarının belirlenmesinde, değeri bilinen 100 emlak kullanılmıştır. Bu emlaklar için, %63 oranında %10 altında hata ile sonuca ulaşılmıştır. Ortalama hata oranı %15.1543 olarak hesaplanmıştır.

Kontrimas ve Verikas'ın [57] çalışmalarında belirtilen 'kabul edilemez değer' göz önüne alındığında; bu tez çalışmasında emlak değerleri, kabul edilebilir hata oranlarıyla başarılı bir şekilde hesaplanmıştır.

Sonuç olarak elde edilen veriler, Java teknolojilerinden faydalanılarak, harita içerikli bir web arayüzü ile kullanıcıya sunulmuştur. GeoServer'da katmanlar halinde tutulan mekânsal veriler, Ext JS Kütüphanesinden faydalanılarak, kullanıcı arayüzünde istenen katmanın görüntülenip, istenen katmanın kaldırılmasına olanak tanıyan şekilde düzenlenmiştir. Ayrıca GeoServer ile düzenlenen harita katmanlarına ek olarak, Google'ın standart, uydu, hibrit ve fiziksel haritaları da kullanıcı arayüzüne eklenmiştir. Elazığ ili için oluşturulan Emlak Değer Haritasında, arama özelliği kullanılarak istenilen mahalleye zoom yapma ve binaların emlak değerlerini görüntüleme imkânı bulunmaktadır. Tüm bu özelliklerin sunulmasında, başta JSP olmak üzere, JavaScript, Ext JS Kütüphanesi, AJAX gibi teknolojilerden faydalanılmıştır.

Sonuç olarak, mekânsal veriler kullanılarak web tabanlı bir Karar Destek Sistemi uygulaması olarak Emlak Değer Haritası oluşturulmuştur. Coğrafi Bilgi Sistemleri ile Karar Destek Sistemlerinin entegre çalışması sonucunda, karar mercilerine karar alma sürecinde yardımcı olabilecek bir sistem geliştirilmiştir. Hesaplanan emlak değerleri göreceli gerçekliğe sahip olup, değeri belirleyici olarak kabul edilen emlaklara göre, diğer emlak değerleri de belirlenmiştir. Sistemin doğruluğu kabul edilebilir düzeyde olup, eğer istenirse z-filtre yöntemi kullanılarak, hata oranı fazla çıkan emlakların elenmesiyle, emlak değerine etki eden faktörlerin ağırlıkları bu elemeler sonucunda tekrar hesaplanıp, gerçek emlak değerlerine daha yakın sonuçlar elde etmek mümkün olabilir.

KAYNAKLAR

- [1] **Komesli, M.**, 2007. Coğrafi Bilgi Sistemlerinin OWL ve GML kullanılarak Anlamsal Web ile bütünleştirilmesi, *Doktora Tez Önerisi*, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.
- [2] **Zhang, Z., Li, J., Liu, Y. and Chen B.**, 2009. Application of GIS and Spatial Decision Support System for Affordable Housing, *4th International Conference on Computer Science & Education*, Nanning, 25-28 July, pp. 1110-1115.
- [3] www.postgresql.org, PostgreSQL, 15 Mart 2009.
- [4] postgis.refractory.net, PostGIS, 17 Mart 2009.
- [5] www.geoserver.org, GeoServer, 30 Mart 2009.
- [6] www.sencha.com/products/extjs, Ext JS, 25 Haziran 2009.
- [7] **Goodchild, M., Haining R. and Wise, S.**, 1992. Integrating GIS and spatial data analysis: problems and possibilities, *Geographical Information Systems*, 6(5), pp. 407-423.
- [8] **Church, R. L. and Murray, A.T.**, 2009. Business Site Selection, Location Analysis and GIS, pp. 20-22, *John Wiley & Sons*, New York.
- [9] <http://www.tagem.gov.tr/gis>, Tarla Bitkileri Merkez Araştırma Enstitüsü Coğrafi Bilgi Sistemleri ve Uzaktan Algılama Bölümü, 5 Haziran 2009.
- [10] **Doğruluk, Ö.**, 2007. Coğrafi Bilgi Sistemleri Kapsamında Batıkent Telekom Santral Şebeke Planlarının Sayısallaştırılması, Sorgulama ve Analizlerin Yapılması, *Yüksek Lisans Tezi*, Kahramanmaraş Sütçü İmam Üniversitesi Fen Bilimleri Enstitüsü, Kahramanmaraş.
- [11] http://www.gunduz.org/seminer/GIS/AKK_GIS_UygulamalarıNicolaiTufar22052005.pdf GIS Uygulamaları, 12 Mayıs 2009.
- [12] **İzgi, E. ve İnan, A.**, 1999. GIS (Coğrafi Bilgi Sistemi) Tabanlı Dağıtım Otomasyonu (1), *Kaynak Elektrik Dergisi*, **124**, s. 142-157.
- [13] <http://www.gislab.ktu.edu.tr/gisnedir/cbs.htm>, Coğrafi Bilgi Sistemleri (CBS) ArGe Laboratuvarı, 23 Mart 2009.
- [14] http://hydrology.rice.edu/bedient/Handouts/CEVE512_Lab2.ppt, Geographic Information Systems, 9 Mayıs 2009.

- [15] **Dempsey, C.**, Geodatabases Explored – Vector and Raster Data.
<http://gislounge.com/geodatabases-explored-vector-and-raster-data>, 21 Mart 2009.
- [16] **Maidment, D. R.**, Vector and raster data models (Introduction to Spatial Hydrology).
http://www.gishydro.umd.edu/documents/train_manual/old/vector-raster.pdf, 13 Mayıs 2009.
- [17] **Aydın, G.**, Coğrafi Bilgi Sistemleri (Geographic Information System: GIS).
<http://web.firat.edu.tr/bilmuh/gaydin/dersler/0809/bmu401/ppt/GIS.doc>, 28 Mayıs 2009.
- [18] **Çetin Z.**, Veri Tabanına Giriş.
www.fatih.edu.tr/~zcetin/0708/VT/dokuman/Veritabani1.pps, 06 Temmuz 2011.
- [19] <http://www.netogretim.com/dokumangoster.aspx?id=177&d=Veri-Taban%C4%B1-Modelleri>, Veri-Tabanı Modelleri, 02 Eylül 2009.
- [20] **Yu, X. and Liu, K.**, 2009. A Spatial Decision Support System for Large Scale Vehicle Routing, *ICMTMA '09. International Conference on Measuring Technology and Mechatronics Automation*, s. 444-449.
- [21] **Turan, F.**, 2006. Web Servisleri Tabanlı Coğrafi Bilgi Sistemleri, *Yüksek Lisans Tezi*, Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- [22] <http://tr.wikipedia.org/wiki/MySQL>, MySQL, 27 Mayıs 2009.
- [23] <http://maisonbisson.com/blog/post/12147/working-with-spatial-data-in-mysql>, Working With Spatial Data in MYSQL, 15 Nisan 2009.
- [24] <http://tr.wikipedia.org/wiki/PostgreSQL>, PostgreSQL, 19 Şubat 2009.
- [25] http://www.postgresql.org/docs/faqs.FAQ_turkish.html#1.1, PostgreSQL Türkiye, 12 Şubat 2009.
- [26] http://www.emo.org.tr/ekler/de6c74e347052d6_ek.pdf?dergi=2, EMO Ankara Şubesi Haber Bülteni, 3 Mayıs 2009.
- [27] www.opengis.com, OpenGIS, 15 Ağustos 2009.
- [28] <http://en.wikipedia.org/wiki/PostGIS>, PostGIS, 20 Mart 2009.
- [29] <http://postgis.refractory.net/documentation/manual-1.3/ch04.html#id2571510>, Using PostGIS, 8 Mart 2009.
- [30] http://en.wikipedia.org/wiki/Spatial_query, Spatial query, Nisan 2009.
- [31] <http://postgis.refractory.net/documentation/manual-1.3/ch06.html>, PostGIS Reference, 5 Nisan 2011.

- [32] **Dinçer, A., Seyrek, K. ve Öztürk, B.**, 2007. İnternet Tabanlı Coğrafi Bilgi Sistemi Uygulamalarında Uluslararası Standartlar, *Ulusal Coğrafi Bilgi Sistemleri Kongresi*, KTÜ, Trabzon, 30 Ekim –02 Kasım.
- [33] **Aydın G.**, 2007. Service Oriented Architecture for Geographic Information Systems Supporting Real Time Data Grids, *PhD Thesis*, Indiana University, Indiana, ABD.
- [34] <http://www.opengeospatial.org/standards>, OpenGeoSpatial(OGC), 24 Mayıs 2009.
- [35] <http://tr.wikipedia.org/wiki/XML>, XML, 3 Haziran 2009.
- [36] http://www.fourleafcontractor.com/index.php?option=com_content&view=article&id=16:ogc-servisleri&catid=11:map-servers&Itemid=40, Four Leaf Technology Contractor, 6 Haziran 2009.
- [37] **Sarı, F., Erdi, A. ve Kırtıloğlu, O. S.**, 2011. İnternet Tabanlı Coğrafi Bilgi Sistemi Uygulamalarında GeoServer, ArcGIS Server ve Google Map API Entegrasyonu, *TMMOB Harita ve Kadastro Mühendisleri Odası 13. Türkiye Harita Bilimsel ve Teknik Kurultayı*, Ankara, 18-22 Nisan.
- [38] <http://www.turkgis.com/forum/viewtopic.php?p=418&sid=caf4bd641409e31c342d94378aa4ca27>, Coğrafi Bilgi Sistemleri Portalı, 7 Mayıs 2009.
- [39] <http://geoserver.org/display/GEOSDOC/Users+Guide>, GeoServer User Guide, 5 Mart 2009.
- [40] <http://geoserver.org/display/GEOSDOC/1.3+Style+Your+Map>, GeoServer User Guide, 22 Mart 2009.
- [41] **Little, J. D. C.**, 1970. Models and managers: the concept of a decision calculus, *Management Science*, 16 (8), pp. 466-485.
- [42] <http://members.tripod.com/~Bagem/bagem/yz1.html>, Karar Destek Sistemleri, 24 Mayıs 2009.
- [43] **Tian-en, C., Li-ping, C., Yunbin, G. and Yanji, W.**, 2009. Spatial Decision Support System for Precision Farming Based on GIS Web Service, *International Forum on Information Technology and Applications*, Chengdu, 15-17 May, pp. 372 – 376.
- [44] <http://www.maximumbilgi.com/default.asp?sx=mkl&ID=10709>, Maximum Bilgi web sitesi, 12 Nisan 2009.
- [45] **Power, D. J.**, 2002. Decision Support Systems: Concepts and Resources for Managers, *Quorum Books*, Westport.

- [46] www.mis.boun.edu.tr/tanrikulu/YBS4.ppt, Karar Destek Sistemleri, 13 Mayıs 2011.
- [47] <http://www.turkgis.com/forum/viewtopic.php?t=115&sid=608051a1312d96d1863e65daa0730875>, TurkGIS Coğrafi Bilgi Sistemleri Portalı, 28 Mayıs 2009.
- [48] **Baykasoğlu, A., Dereli, T., Çağşır, H. ve Güllü H.**, 2004. Coğrafi Bilgi Sistemlerinin Mühendislik Karar Destek Sistemlerine Uygulanabilirliği Üzerine Bir Araştırma, *Yöneylem Araştırması/Endüstri Mühendisliği - XXIV Ulusal Kongresi*, Gaziantep – Adana, 15-18 Haziran.
- [49] **Barton, J., Plume, J. and Parolin, B.**, 2005. Public participation in a spatial decision support system for public housing, *Computers, Environment and Urban Systems*, 29(6), pp. 630-652.
- [50] **Johnson, M.P.**, 2001. A spatial decision support system prototype for housing mobility program planning, *Journal of Geographical Systems*, pp. 49-67.
- [51] **Johnson, M.P.**, 2005. Spatial decision support for assisted housing mobility counseling, *Decision Support Systems*, 41(1), pp.296-312.
- [52] **Erden, T. ve İpbüker, C.**, 2003. Karar Destek Sistemi Olarak Mekânsal Analiz ve CBS (Spatial Analysis For Decision Making and GIS), *Harita Dergisi*, 130, s. 1-11.
- [53] **Nişancı, R.**, 2003. Kentsel Taşınmaz Değer Haritalarının Oluşturulması, *9. Türkiye Harita Teknik ve Bilimsel Kurultayı*, Ankara, 31 Mart – 4 Nisan.
- [54] **Karadağ, H.**, Genetik Algoritma ile Taşınmaz Değerlemesi. <http://degerleme.blogcu.com/genetik-algoritma-ile-tasinmaz-degerlemesi/3426396>, 20 Ağustos 2010.
- [55] **Zeng, Q. T. and Zhou, Q.**, 2001. Optimal Spatial Decision Making Using GIS: A Prototype of A Real Estate Geographical Information System (REGIS), *International Journal Of Geographical Information Science*, 15(4), pp. 307-321.
- [56] **Timur S.**, 2009. Coğrafi Bilgi Sistemleri Destekli Taşınmaz Değer Haritalarının Oluşturulması: İstanbul İli Şişli İlçesi Örneği, *Yüksek Lisans Tezi*, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [57] **Kontrimas V. and Verikas, A.**, 2011. The Mass Appraisal of The Real Estate by Computational Intelligence, *Applied Soft Computing*, Amsterdam, The Netherlands, 11(1), pp. 443-448.

- [58] **Yomralıoğlu, T.**, 1993. A GIS - Based Land Readjustment System for Urban Development, *EGIS'93 IVth European Conference on Geographical Information Systems in Genoa*, Amsterdam, The Netherlands, 1, pp. 372-379.
- [59] **Din, A., Hoesli, M. and Bender, A.**, 2001. Environmental Variables and Real Estate Prices, *Urban Studies*, 38(11), pp. 1989-2000.
- [60] **Torun, M. K., Yanalak, M. ve Şeker, D.Z.**, 2009. Taşınmaz Değer Haritalarının Coğrafi Bilgi Sistemleri ile Üretilmesi, *TMMOB Harita ve Kadastro Mühendisleri Odası 12. Türkiye Harita Bilimsel ve Teknik Kurultayı*, Ankara, 11-15 Mayıs.
- [61] **Yoon, K. and Hwang, C. L.**, 1995. Multiple attribute decision making: An introduction, *Sage Publications*, Thousand Oaks, CA.
- [62] **Özkan, G. ve Yalpir, Ş.**, 2005. Taşınmaza Ekonomik Bakış ve Değerlendirmesi, *TMMOB Harita ve Kadastro Mühendisleri Odası 10. Türkiye Harita Bilimsel ve Teknik Kurultayı*, Ankara, 28 Mart - 1 Nisan.
- [63] **Erdal, F.**, 2007. Optimum Design of Grillage Systems Using Harmony Search Algorithm, *Master Thesis*, Graduate School of Natural and Applied Sciences of Middle East Technical University, Ankara.
- [64] **Şeker, Ş. E.**, Sezgisel Algoritmalar (Buluşsal Algoritmalar, Heuristic Algorithms). <http://www.bilgisayarkavramlari.com/2008/12/22/sezgisel-algoritmalar-bulussal-algoritmalar-heuristic-algorithms/>, 16 Haziran 2011.
- [65] http://tr.wikipedia.org/wiki/Sezgisel_algoritma, Sezgisel Algoritma, 17 Haziran 2011.
- [66] **Geem, Z.W., Kim J.H. and Loganathan G.V.**, 2001. A new heuristic optimization algorithm: harmony search, *Simulation*, 76(2), pp. 60-68.
- [67] **Lee, K. S. and Geem, Z. W.**, 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods Appl. Mech. Engrg*, 194, pp. 3902-3933.
- [68] **Geem, Z. W.**, Music-Inspired Optimization Algorithm: Harmony Search. http://ftp.ltu.edu.tw/~mhc/Special%20Topic/references/ppt/HS_Intro.pdf, 20 Mart 2009.
- [69] **Lee, K. S. and Geem, Z. W.**, 2004. A new structural optimization method based on the harmony search algorithm, *Computers and Structures*, 82, pp. 781-798.
- [70] **Gao K., Pan Q., Zuo F. and Duan J.**, 2010. A Harmony search algorithm for the no-wait flow shop optimization scheduling, *The 29th Chinese Control Conference*, Beijing, China, July 29-31, pp. 1723-1726.

- [71] **Mahdavi M., Fesanghary, M. and Damangir, E.,** 2007. An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation*, 188, pp. 1567–1579.
- [72] **Omran, M. G. H. and Mahdavi, M.,** 2008. Global-Best Harmony Search, *Applied Mathematics and Computation*, 198, pp. 643–656.
- [73] <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>, Java Server Pages Technology, 20 Haziran 2011.
- [74] http://tr.wikipedia.org/wiki/JavaServer_Pages, Java Server Pages, 14 Haziran 2011.
- [75] http://www.godoro.com/Divisions/Ehil/Mahzen/Java/JSPNedir/txt/html/document_What_IsJSP.html, JSP Nedir, 18 Haziran 2011.
- [76] <http://tr.wikipedia.org/wiki/JavaScript>, JavaScript, 24 Mayıs 2011.
- [77] www.bilgisayarogren.com/java1.doc, JavaScript Nedir, 14 Mayıs 2011.
- [78] <http://bid.ankara.edu.tr/yardim/www/javasturk/javascr.html>, JavaScript'e Giriş, 17 Mayıs 2011.
- [79] www.ajax.org, AJAX, 3 Haziran 2011.
- [80] www.dynamicdrive.com, DHTML, 5 Haziran 2011.
- [81] domscripting.com, DOM Scripting, 6 Haziran 2011.
- [82] **Topak, U.,** Ext JS Nedir. http://www.ugurtopak.com/Post/show_article.aspx?id=3, 23 Mart 2011.
- [83] **Aydın, H. ve Berktaş, Y.,** Web 2.0 Technologies (AJAX, RSS etc.). web.firat.edu.tr/bilmuh/gaydin/dersler/0809/bmu401/ppt/Web2.0.doc, 23 Haziran 2011.
- [84] www.openlayers.org, OpenLayers, 10 Nisan 2009.
- [85] www.maps.google.com, Google Maps, 20 Nisan 2009.

ÖZGEÇMİŞ

Ayşe ERDOĞAN, 1985 yılında Elazığ'da doğdu. Lise öğrenimini Elazığ Anadolu Lisesi'nde tamamladıktan sonra, 2004-2008 yılları arasında Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde eğitime devam ederek, Bilgisayar Mühendisi unvanını aldı. Aynı zamanda 2004-2009 yılları arasında Anadolu Üniversitesi İşletme Fakültesi İşletme Bölümü'nü okudu. Mart 2009 - Eylül 2009 tarihleri arasında Zemin Harita Müh. Müt. San. Tic. Ltd. Şti. 'de ve Eylül 2009 – Şubat 2010 tarihleri arasında Yonca CBS Bilişim Sistemleri Bilgisayar - Harita Müh. Şti' de Yazılım Geliştirici olarak çalıştı. Ağustos 2010 - Ocak 2011 tarihleri arasında İçişleri Bakanlığı Bilgi-İşlem Dairesi Başkanlığı'nda Programcı olarak görev yaptı. Ocak 2011 tarihinden itibaren ise, Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaya devam etmekte olan Erdoğan, iyi derecede İngilizce bilmektedir.