

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

JEODEZİ VE JEOİFORMASYON MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

ELEKTRİKLİ SCOOTER KULLANIMINDA YOL AĞI
KISITLARININ PYTHON TABANLI DEĞERLENDİRİLMESİ
VE WEB TABANLI SUNUMU: İZMİT ÖRNEĞİ

AHMET ŞİRİN

KOCAELİ 2024

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

JEODEZİ VE JEOİFORMASYON MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

ELEKTRİKLİ SCOOTER KULLANIMINDA YOL AĞI
KISITLARININ PYTHON TABANLI DEĞERLENDİRİLMESİ
VE WEB TABANLI SUNUMU: İZMİT ÖRNEĞİ

AHMET ŞİRİN

Prof. Dr. Arzu ERENER

Danışman, Kocaeli Üniv.

.....

Prof. Dr. Ozan ARSLAN

Jüri Üyesi, Kocaeli Üniv.

.....

Doç. Dr. Kıvanç ERTUGAY

Jüri Üyesi, Akdeniz Üniv.

.....

Tezin Savunulduğu Tarih: 09.01.2024

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmada,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/Projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

■ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

□ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler tarafından no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Ahmet ŞİRİN

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.

Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan **“Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge”** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.

☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.

☒ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

Ahmet ŞİRİN

ÖNSÖZ VE TEŞEKKÜR

Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Jeodezi ve Jeoinformasyon Mühendisliği bölümünde yapılan tez çalışması kapsamında, gün geçtikçe hayatımızda daha fazla yer edinen, gelişen ve modernleşen elektrikli scooterlar için en uygun yol problemleri ile ilgili araştırmalar ve çalışmalar yapılmıştır. Çalışma içerisinde İzmit ilçesi sınırlarında, bir yol ağının nasıl oluşturulabildiğine, birden fazla faktörün bu yol ağı ile nasıl ilişkilendirilebildiğine, hangi durumlarda ve nasıl kullanılabildiklerine, sonuçlar üzerinde nasıl etkileri olduklarına ve web tabanlı harita uygulaması geliştirme süreci ile ilgili detaylı uygulamalar ve incelemeler yapılmıştır. Çalışmanın ana amacı, elektrikli scooterlar için yol ağı kısıtlarının incelenmesinin yanı sıra, bu konuda yapılacak diğer araştırmalar için de bir ön zemin oluşmasının sağlanmasıdır. Çeşitli uygulamalar ile farklı faktörlerin uygulamalarda nasıl sonuçlar verdiği incelenerek, bu çalışmanın ne gibi eklemelerle geliştirilebileceği ve ileriki süreçlerde nasıl çözümler oluşturulabileceği hakkında çeşitli saptamalar yapılmıştır.

Tez çalışması süreci boyunca bana yol gösteren, sahip olduğu kaynakları benimle koşulsuz paylaşan, ayrıca değerli zamanını bana ayırarak motive eden, manevi destek sağlayan tez danışmanım ve saygıdeğer hocam Prof. Dr. Arzu Erener'e sonsuz teşekkürlerimi sunarım.

Uygulama yöntemleri konusunda tavsiyelerini aldığım Piri1513 Coğrafi Bilgi Sistemleri şirketi yetkililerine teşekkür ederim.

Tez içerisinde önemli bir noktada gerekli olan bir veri kaynağını çalışmalarına dahil etmem için bana koşulsuz sunan Kocaeli Büyükşehir Belediyesi'ne teşekkür ederim. Bana maddi ve manevi desteklerini hiç esirgemeyen, daima yanımda olarak beni hayat boyu motive eden aileme de sonsuz teşekkürler.

Ocak – 2024

Ahmet ŞİRİN

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ	i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI	ii
ÖNSÖZ VE TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vi
TABLolar DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
1.1. Problemin Tanımı	3
1.2. Tezin Amacı	4
1.3. Literatür Taramaları	5
2. EN KISA YOL PROBLEMİ	8
2.1. En Kısa Yol Problemi Algoritmaları	9
2.1.1. Dijkstra Algoritması	10
2.1.2. Bellman-Ford Algoritması	10
2.1.3. A* Algoritması	11
2.1.4. Floyd-Warshall Algoritması	11
2.2. En Kısa Yol Problemi Türleri	12
2.2.1. Tek Kaynaklı En Kısa Yol Problemi	12
2.2.2. Tek Çift En Kısa Yol Problemi	12
2.2.3. Tek Hedef En Kısa Yol Problemi	12
2.2.4. Tüm Çiftler En Kısa Yol Problemi	13
2.3. En Uygun Yol Problemi	13
2.4. Coğrafi Bilgi Sistemleri	13
2.4.1. CBS Veri Yapıları	14
2.4.2. CBS'nin Kullanıldığı Alanlar	15
2.4.3. CBS'de Topoloji Kavramı	16
2.4.3.1. Hat-Düğüm (Arc-Node) Topolojisi	17
2.4.3.2. Poligon-Hat (Polygon-Arc) Topolojisi	17
2.4.3.3. Sol-Sağ (Left-Right) Topolojisi	18
2.4.4. En Uygun Yol Uygulamalarında CBS'nin Yeri	19
2.4.5. QGIS Yazılımı	19
2.4.6. OpenStreetMap	20
2.5. Python Programlama Dili	20
2.5.1. Python Kütüphaneleri	21
2.5.1.1. Pandas	21
2.5.1.2. Geopandas	21
2.5.1.3. NetworkX	21
2.5.1.4. OSMnx	21
2.5.2. Flask	22
2.6. JavaScript Programlama Dili	22
2.6.1. OpenLayers	23
2.7. API Entegrasyonu	23
2.7.1. Rest API	24

2.7.2. HTTP Metotları	24
2.8. Elektrikli Scooter Kullanımı	25
2.9. Elektrikli Scooterlar İçin En Uygun Yol Problemi ve Faktörleri	26
3. MATERYAL VE YÖNTEM	28
3.1. Metodoloji	28
3.2. Çalışma Alanı	30
3.3. Veri Kaynakları	31
3.3.1. Yol Ağının Oluşturulması	32
3.4. Yol Ağı ile Faktörlerin İlişkilendirilmesi	33
3.4.1. Yol Eğimi Faktörünün İlişkilendirilmesi	33
3.4.2. Yol Sınıfı Faktörünün İlişkilendirilmesi	41
3.4.3. Arazi Kullanımı Faktörünün İlişkilendirilmesi	42
3.5. Ağırlık İndeks Değerlerinin Hesaplanması	44
3.5.1. Yol Eğimi Ağırlık İndeks Değerlerinin Hesaplanması	45
3.5.2. Yol Sınıfı Ağırlık İndeks Değerlerinin Hesaplanması	46
3.5.3. Arazi Kullanımı Ağırlık İndeks Değerlerinin Hesaplanması	48
3.6. Uygulama Tiplerinin Belirlenmesi	50
3.6.1. Batarya Doluluğu Faktörünün Dikkate Alınması	51
3.6.2. Uygulama 1	53
3.6.3. Uygulama 2	53
3.6.4. Uygulama 3	54
3.6.5. Uygulama 4	54
3.6.6. Uygulama 5	55
3.6.7. Uygulama 6	55
3.6.8. Uygulama 7	55
3.7. Web Tabanlı Harita Uygulaması	55
3.7.1. Uygulama Arayüzü	56
4. UYGULAMA	62
4.1. Senaryoların Belirlenmesi	62
4.2. Senaryo 1	63
4.3. Senaryo 2	66
4.4. Senaryo 3	69
5. SONUÇLAR VE ÖNERİLER	72
KAYNAKLAR	78
EKLER	83
KİŞİSEL YAYIN VE ESERLER	94
ÖZGEÇMİŞ	95

ŞEKİLLER DİZİNİ

Şekil 2.1.	Düğümleeri arası ağırlıklı yönlendirilmiş örnek en kısa yol grafiğı	8
Şekil 2.2.	Königsberg köprüleri probleminin grafik teorisine uyarlanması	9
Şekil 2.3.	Dijkstra algoritması ile en kısa yol problemi çözüm örneğı	10
Şekil 2.4.	Bellman-Ford algoritmasının şematik prosedür örneğı	11
Şekil 2.5.	Coğrafi bilgi sistemleri katman yapısı örneğı	14
Şekil 2.6.	Raster ve vektör katmanlarının gösterimi örneğı	15
Şekil 2.7.	Örnek hat-düğüm topolojisi ve temsil listesi	17
Şekil 2.8.	Örnek poligon-hat topolojisi ve temsil listesi	18
Şekil 2.9.	Örnek sol-sağ topolojisi ve temsil listesi	18
Şekil 2.10.	En uygun yolun CBS ile görselleştirilmesi örneğı	19
Şekil 2.11.	QGIS yazılımı logosu	19
Şekil 2.12.	OSM logosu	20
Şekil 2.13.	Python programlama dili logosu	20
Şekil 2.14.	JavaScript programlama dili logosu	22
Şekil 2.15.	OpenLayers logosu	23
Şekil 2.16.	Örnek elektrikli scooter modeli	26
Şekil 3.1.	Metodoloji	29
Şekil 3.2.	Çalışma alanı	30
Şekil 3.2.	Yol ağı verisi, a-örnek bölge, b-tüm yol ağı	32
Şekil 3.3.	Yükseklik verisi ve İzmit ilçe sınırları (kırmızı)	34
Şekil 3.4.	İzmit İlçesi stilize edilmiş yükseklik verisi ve İzmit ilçe sınırı	34
Şekil 3.5.	Düğüm verileri ile yükseklik raster verisi	35
Şekil 3.6.	Yol eğimi hesabı	36
Şekil 3.7.	Yol eğimi verisi örnek görünümü	37
Şekil 3.8.	Yükseklik raster görüntüsü ile yol verileri	38
Şekil 3.9.	100 Metrekarelik yükseklik piksel orta noktaları	39
Şekil 3.10.	Yükseklik raster görüntüleri, a-100 Metrekare, b-1 metrekare	39
Şekil 3.11.	Yükseklik değerleri, a-eski, b-yeni	40
Şekil 3.12.	Mutlak eğim değerleri, a-eski, b-yeni	40
Şekil 3.13.	Örnek yol sınıfı verisi	41
Şekil 3.14.	İzmit bölgesi Google Satellite Görüntüsü	42
Şekil 3.15.	50000 ölçekli plan verisi	43
Şekil 3.16.	Plan verileri ve arazi kullanım grupları eşleştirmesi	44
Şekil 3.17.	Örnek yol eğim indeksi	46
Şekil 3.18.	Buffer işlemi örnek bölgesi, a-örnek yollar, b-buffer poligonları	49
Şekil 3.19.	Ağırlık indeks değerlerinin ilişkilendirildiğı örnek bir buffer	50
Şekil 3.20.	Uygulama arayüzü, a-ana ekran, b-temel butonlar	57
Şekil 3.21.	Başlangıç-hedef noktası bilgisi	57
Şekil 3.22.	Konuma git	58
Şekil 3.23.	Elektrikli scooter ayarları	58
Şekil 3.24.	En uygun yolların bulunması	59
Şekil 3.25.	Yakındaki düğümü getir servis sonucu	59
Şekil 3.26.	En uygun yolları getir servisi sonucu	60
Şekil 3.27.	Sonuç analizi bilgileri	61
Şekil 4.1.	Senaryo 1 başlığı uygulamaları harita gösterimi	63
Şekil 4.2.	Senaryo 1 Google Street View örneğı	64

Şekil 4.3.	Senaryo 2 başlığı uygulamaları harita gösterimi	66
Şekil 4.4.	Senaryo 2 batarya dolğunun bittiği örnek aralıklar	66
Şekil 4.5.	Senaryo 2 Google Street View örneği	67
Şekil 4.6.	Senaryo 3 başlığı uygulamaları harita gösterimi	69
Şekil 4.7.	Senaryo 3 başlığı eş yükseklik haritası ile örnek bölgesi	70
Şekil 4.8.	Senaryo 3 Google Street View örneği	70
Şekil A.1.	Ana kod dosyası	84
Şekil B.1.	Fonksiyon kod dosyası	86
Şekil C.1.	Uygulama kod dosyası	89



TABLÖLAR DİZİNİ

Tablo 3.1.	Veri kaynakları	31
Tablo 3.2.	Yol ağı kenar verisi tablosu (örnek sütun ve veriler)	33
Tablo 3.3.	Yükseklik değerli düğüm verisi (örnek sütun ve veriler)	35
Tablo 3.4.	Eğim değerli kenar verisi (örnek sütun ve veriler)	36
Tablo 3.5.	Arazi kullanımı sınıflandırma	48
Tablo 3.6.	Uygulamalar ve kullanılan faktörler	51
Tablo 4.1.	Uygulama temsili renkleri	62
Tablo 4.2.	Senaryo 1 genel bilgiler	64
Tablo 4.3.	Senaryo 1 maliyet değerleri detaylı karşılaştırması	65
Tablo 4.4.	Senaryo 2 genel bilgiler	68
Tablo 4.5.	Senaryo 2 maliyet değerleri detaylı karşılaştırması	68
Tablo 4.6.	Senaryo 3 genel bilgiler	71
Tablo 4.7.	Senaryo 3 maliyet değerleri detaylı karşılaştırması	71

SİMGELER VE KISALTMALAR DİZİNİ

B	: Batarya Doluluğu, (%)
E	: Eğitim Değeri, (%)
M	: Maliyet Değeri
NA	: Arazi Kullanımı Faktörü İndeks Değeri
NAB	: Arazi Kullanımı Buffer Verileri İndeks Değeri
NE	: Eğitim Faktörü Ağırlık İndeks Değeri
NS	: Yol Sınıfı Faktörü Ağırlık İndeks Değeri
U	: Yolun Uzunluğu, (m)
Δ	: Delta
∞	: Sonsuz

Kısaltmalar

API	: Application Programming Interface (Uygulama Programlama Arabirimi)
App	: Application (Uygulama)
CBS	: Coğrafi Bilgi Sistemleri
DEM	: Digital Elevation Model (Sayısal Yükseklik Modeli)
GeoJSON	: Geo JavaScriptObject Notation (Geometrik JavaScript Nesnesi Gösterimi)
HTTP	: Hypertext Transfer Protocol (Hiper Metin Transfer Protokolü)
JSON	: JavaScript Object Notation (JavaScript Nesnesi Gösterimi)
KBB	: Kocaeli Büyükşehir Belediyesi
OSM	: OpenStreetMap
TIN	: Triangulated Irregular Network (Üçgen Düzensiz Ağ)
XML	: Extensible Markup Language (Genişletilebilir İşaretleme Dili)

ELEKTRİKLİ SCOOTER KULLANIMINDA YOL AĞI KISITLARININ PYTHON TABANLI DEĞERLENDİRİLMESİ VE WEB TABANLI SUNUMU: İZMİR ÖRNEĞİ

ÖZET

Ulaşım araçları günlük hayatımızın büyük bir bölümünde yer almaktadır. Son dönemlerde şehir içi ulaşımlarda özgür hareket kolaylığı sağlaması, çevre dostu olması, düşük maliyetli olması ve araçlara göre daha ekonomik olması dolayısı ile scooterlar tercih edilir olmuştur. Ancak, araç trafiğinde kullanılmaya başlanan elektrikli scooterün güvenli kullanımı ve kuralları konusunda kamuoyunda soru işaretleri oluşmaya başlamıştır. Türkiye'de bu alandaki yasal düzenleme ve kuralların henüz yeterince standardize edilmediği görülmektedir. Bu soru işaretleri arasında güvenlik, konfor ve araç özellikleri elektrikli scooterların kullanımları sırasında hangi yolları seçmeleri gerektiği konusunda dikkat çekmektedir. Bu çalışma kapsamında Python programlama dili ve kütüphaneleri yardımıyla en uygun yol uygulamaları yapılmıştır. Bu kapsamda batarya doluluğu, yolun eğimi, yol sınıfı, yolun uzunluğu ve arazi kullanımı parametreleri dikkate alınmıştır. En uygun yol uygulamaları için farklı başlangıç ve hedef noktalarının bulunduğu senaryolar uygulanmış ve bu uygulamalarda Dijkstra Algoritması kullanılmıştır. Senaryoların uygulanmasında, analiz sonuçlarının tablosal olarak ve harita üzerindeki görselleştirmelerinde JavaScript ve python programlama dillerinden faydalanılarak oluşturulan web harita uygulamasından yararlanılmıştır. Uygulama arayüzü, tablosal değerler ve grafiksel gösterimlerle detaylı analiz yapılabilme olanağı sağlayan bir JavaScript-OpenLayers harita uygulamasıdır. Web arayüzü ve uygulama fonksiyonları arasındaki iletişim için Rest API kullanılmıştır. Sonuç olarak araştırmada kullanılan bu araç ve teknolojilerin bu alandaki araştırmalarda entegre olarak kullanılabileceği sonucu ortaya çıkmaktadır. Elektrikli scooterlar için yapılan çalışma en uygun alternatif güzergahların belirlenmesi ile zamanın daha verimli kullanılmasına katkıda bulunmakta olup, belirlenen kısıtlara yönelik önlemlerin alınmasına öncülük edecek olması bakımından da oldukça önemli sonuçlar üretmiştir.

Anahtar Sözcükler: Ağ Analizi, CBS, Elektrikli Scooter, En Kısa Yol, Python.

PYTHON-BASED EVALUATION AND WEB-BASED PRESENTATION OF ROAD NETWORK CONSTRAINTS IN ELECTRIC SCOOTER USE: IZMIT CASE

ABSTRACT

Vehicles hold a pivotal role in daily life, with scooters emerging as a preferred urban transportation choice due to their maneuverability, environmental friendliness, cost-effectiveness, and overall economic benefits. However, the integration of electric scooters into traffic has raised concerns regarding safety and regulations, particularly in Turkey, where standardization of legal rules remains insufficient. This study addresses these concerns, emphasizing the importance of selecting appropriate routes based on safety, comfort, and scooter features. Using the Python programming language and libraries, optimal route applications were developed, considering parameters like battery level, road incline, class, length, and terrain. Different scenarios, employing the Dijkstra Algorithm, were implemented for various starting and destination points. A web map application, created using JavaScript and Python, provided tabular results and visualizations, enhancing detailed analysis. The application interface, a JavaScript-OpenLayers map, facilitated comprehensive exploration through tabular values and graphical representations. Communication between the web interface and application functions utilized Rest API. Ultimately, the tools and technologies employed in this research can be integrated for broader use in the field. The study on electric scooters significantly contributes to time efficiency by identifying optimal alternative routes and implements measures to address identified constraints, marking a crucial step towards a more regulated and secure scooter transportation system.

Keywords: Network Analysis, GIS, Electric Scooter, Shortest Path, Python.

1. GİRİŞ

Elektrikli scooterlar, son yıllarda şehirlerde giderek artan bir popülerite kazanmıştır. Bu hızla yükselen ilgi, çevre dostu ulaşım seçeneklerine olan talebin artmasıyla birlikte ortaya çıkmıştır. Elektrikli scooterlar, benzinli motorlu alternatiflerine kıyasla daha çevre dostu ve sürdürülebilir bir ulaşım seçeneği sunmaktadır. Ayrıca, trafik sıkışıklığının ve otopark sorunlarının olduğu şehirlerde pratik ve hızlı bir ulaşım aracı olarak da tercih edilmektedir.

Gündelik yaşamda kullanılmaya başlanma süreci, elektrikli scooterların avantajlarından kaynaklanmaktadır. Birçok kullanıcı, kısa mesafelerde ve trafiği yoğun olan bölgelerde ulaşım ihtiyaçlarını karşılamak için elektrikli scooterları tercih etmektedir. Bu taşıtlar, hızlıca istenen noktaya ulaşma imkanı sağlayarak zaman tasarrufu yapmayı mümkün kılmaktadır. Ayrıca, elektrikli scooterların sessiz çalışması ve sıfır emisyon üretmesi, çevre kirliliğini azaltma potansiyeline sahiptir. Bununla birlikte, elektrikli scooterların gündelik yaşamda kullanılmasında bazı olumsuz durumlar da ortaya çıkabilmektedir. Özellikle, trafik düzenlemelerine uyum sağlama konusunda bazı zorluklar yaşanmaktadır. Elektrikli scooterların kaldırımlarda veya yaya bölgelerinde kullanılması, yaya güvenliğini tehlikeye atabilmektedir. Ayrıca, elektrikli scooter kullanıcılarının trafik kurallarına uymama eğilimi göstermesi, kazalara yol açabilecek riskleri beraberinde getirmektedir. Bu gibi sorunlar, elektrikli scooterların düzenlenmesi ve kullanımına yönelik daha etkin politikaların geliştirilmesi gerekliliğini ortaya koymaktadır.

En kısa yol uygulamaları, elektrikli scooter kullanıcıları için önemli bir araçtır. Bu uygulamalar, kullanıcılara hedef noktaya en hızlı ve verimli şekilde ulaşmalarına yardımcı olan rotalar sunar. Genellikle GPS tabanlı olan bu uygulamalar, kullanıcının konumunu belirler ve ardından çeşitli faktörleri dikkate alarak en uygun rotayı hesaplar. Bu uygulamalar, trafik durumunu, yol koşullarını ve hatta kullanıcının tercihlerini değerlendirerek en etkili rotayı belirlemek için karmaşık algoritmalar kullanır. Örneğin, trafik sıkışıklığı olan bölgelerden kaçınmak veya bisiklet yollarını tercih etmek gibi faktörler göz önünde bulundurulabilir. Ayrıca, elektrikli scooter kullanıcılarının seyahat mesafelerini ve hızlarını da hesaba katarak pil ömrünü en verimli şekilde kullanmalarına yardımcı olur. Kullanıcıların zamanı ve enerjisi tasarrufu yapmasına yardımcı olurken,

daha verimli ve etkili bir ulaşım deneyimi sunar.

En kısa yol uygulamaları, elektrikli scooter kullanıcıları için birkaç farklı platformda mevcuttur. Bazıları cep telefonlarına indirilebilir uygulamalar olarak sunulurken, bazıları web tabanlı arayüzlere sahiptir. Kullanıcılar, seyahat öncesi veya seyahat sırasında bu uygulamaları kullanarak en uygun rotayı belirleyebilir ve navigasyon desteği alabilir. Bu uygulamalar, elektrikli scooterların günlük yaşamda yaygınlaşmasında önemli bir rol oynar ve kullanıcıların bu çevre dostu taşıtlardan en iyi şekilde faydalanmalarını sağlar.

Ağ analizinde kullanılan ve birbiriyle ilişkili düğümlerden oluşan kenarlar aracılığıyla yapılan karar verme çalışmalarında, en kısa yol problemleri en popüler problemlerden biridir ve üzerine algoritmalar geliştirilir. Bu problemler günümüzde navigasyon cihazları, ağ sistemleri ve araç rotası hesaplamaları gibi birçok alanda karşımıza çıkar. Bu nedenle, bu problemlerdeki özel koşulları dikkate alan çözümler için çeşitli algoritmalar geliştirilmiştir. Bu algoritmalarından en popüler olanı, pozitif ağırlıklı kenarlarla çalışan Dijkstra Algoritmasıdır. Dijkstra algoritması, karmaşık bir ağ analizi problemine etkili bir çözüm sunarak, çeşitli alanlarda karşılaşılan En Kısa Yol problemlerini çözmekte büyük önem taşır. Bu algoritmanın kullanımı, günlük hayatta navigasyon cihazlarından ağ sistemlerine ve araç rotası hesaplamalarına kadar birçok alanda kullanıcıların zaman ve enerji tasarrufu yapmalarına yardımcı olur.

Bir yol ağı, birçok düğümün birbirlerine kenarlar aracılığıyla bağlanmasıyla oluşur. Bazı düğümler, merkeziyet durumundan dolayı ağ içinde daha önemli roller oynayabilir. Karmaşık yol ağları üzerinde geçmişten günümüze birçok algoritma geliştirilmiştir. Bu büyük çeşitlilik, uygulamanın farklı türlere ve amaçlara göre bölünmesini ve problemlerin çözümü için kullanılan sürenin dikkate alınmasını gerektirir. Veri boyutu ve veriler arasındaki dallanmalar gibi ek sorunlar, farklı optimizasyon yöntemlerinin takip edilmesine olanak tanır. Yol ağları, genellikle karmaşık yapılar içerir ve bu yapılar üzerindeki çeşitli problemlerin çözülmesi için farklı algoritmalara ihtiyaç duyulur. Örneğin, bazı algoritmalar, belirli bir hedefe en kısa yolu bulmayı amaçlarken, diğerleri ağ üzerindeki en merkezi düğümleri tanımlamak için kullanılır. Bu çeşitlilik, yol ağının büyüklüğü, veri boyutu ve özel gereksinimler gibi faktörlere bağlı olarak farklı optimizasyon yöntemlerinin kullanılmasını gerektirir. Ayrıca, veri boyutu ve verilerin kendi içerisindeki dallanmalar gibi ek problemler, yol ağı üzerindeki optimizasyon

yöntemlerinin belirlenmesinde etkili olabilir. Veri boyutu büyük olduğunda veya veriler arasında çok sayıda dallanma varsa, daha hızlı ve verimli algoritmalar kullanmak önemli hale gelir. Bu şekilde, yol ağı üzerindeki problemlerin çözümü için daha optimize edilmiş bir yaklaşım elde edilir.

Coğrafi Bilgi Sistemleri (CBS), mekâna dayalı bilgileri inceleyen ve analizler ile modellemeler gerçekleştirilen bir alanı kapsar. Bu sistemler içerisinde, çeşitli araştırmalar ve incelemeler için mekansal veriler kullanılmaktadır. Bu çalışmalar arasında, en uygun yer analizleri, afet risk araştırmaları, mekansal ilişkiler ve istatistikler gibi birçok konu yer alır. Bu çalışma kapsamında, yol ağları, yol geometrileri ve öznitelik bilgileri gibi coğrafi veri kaynakları, gün geçtikçe artan popülaritesiyle gözlemlenen elektrikli scooter gibi elektrikli araçların kullanımında karşılaşılan en kısa yol problemlerinin çözümünde kullanılmaktadır.

CBS'nin sunduğu yol ağı verileri ve analiz yöntemleri, elektrikli scooter gibi araçların en uygun ve kısa yolunun belirlenmesi gibi ihtiyaçları karşılamak için önemli bir kaynak oluşturur. Yol ağı verileri, yol geometrileri ve öznitelik bilgileri, elektrikli scooter kullanıcıları için hızlı, etkili ve enerji tasarruflu yol planlaması sağlamada yardımcı olur. Elektrikli scooterların yaygınlaşmasıyla birlikte, CBS'nin yol ağı verilerine dayalı analizlerinin önemi artmaktadır. Bu araştırma, elektrikli scooter kullanımıyla ilgili en kısa yol problemlerini çözmek için CBS'nin yol ağı verilerini ve analiz yöntemlerini etkili bir şekilde kullanarak, kullanıcıların daha verimli ve rahat bir yolculuk deneyimi yaşamalarına katkıda bulunmayı amaçlamaktadır.

1.1. Problemin Tanımı

Gündelik yaşamda yer edinmiş ve insanların farketmese de sık sık kendi çözümlerini oluşturduğu ya da çeşitli uygulamalar aracılığıyla çözümler edildiği en kısa yol problemleri, farklı çevresel parametrelerin de dikkate alınması sonucu en uygun yol problemlerine dönüşmekte. Farklı mesafelere sahip yollar arasında en kısa yol problemi çözümü için diğerlerine kıyasla daha kısa mesafe olan yolu seçmek doğru bir çözüm olsa da en uygun yol problemleri için bu durum daha karmaşıktır. Örneğin yol üzerindeki trafik yoğunluğu değerleri dikkate alındığında, trafik içerisinde geçecek bir araç için en kısa mesafeli yol, en uygun yol olmayabilmektedir. Bu gibi farklı parametreler, problem

içerisinde yolların taşıtsız veya hangi tip taşıtlarla kat edileceğine bağlı olmakla birlikte, parametrelerin en uygun yol problemi içerisindeki önemleri de çevresel faktörlere göre değişebilmektedir.

Teknolojinin gelişmesi ile elektrikli scooterlar ve elektrikli bisikletler gibi taşıtlar insanların gündelik hayatına yerleşmeye başlamıştır. Ana yolları kullanan motorlu taşıtların kullanımının gün geçtikçe hızla arttığı dönemlerde, insanlar trafik sıkıntısı çekmeyen veya bir eğlence aktivitesi olarak da kullanılabilen elektrikli scooterları bir alternatif olarak görmektedir. Bir araç için en uygun yol her zaman çözümü aranan bir problem olarak görülür. Bu nedenle trafiğe çıkabilen farklı türdeki bu yeni araçlar için de bu problem üzerine örnek araştırmaların ve uygulamaların yapılma gerekliliği ortaya çıkmıştır.

Bir elektrikli scooterın toplam kat edebileceği mesafe de mevcut taşıtın özelliklerine ve mevcut enerjisine bağlıdır. Yol özelliklerinin taşıtın maksimum alabileceği mesafe üzerindeki etkisi göz ardı edilememektedir. Kullanıcının hedef noktasına varıp varamayacağı da önemli bir husus haline gelmektedir. Bununla beraber henüz elektrikli scooter gibi yeni ve teknolojik araçların tam olarak gündelik hayata yerleşmemesi ve bu nedenle vatandaşların bu araçları trafikte görmeye alışık olmaması sebebiyle talihsiz kazalar yaşanmaktadır. Bu nedenle elektrikli scooterlar için en uygun yol tercihinde hangi yolların diğer yollara göre daha az tehlikeli olduğu da uygunluk bakımından önem arz etmektedir.

1.2. Tezin Amacı

Tezin amacı elektrikli scooter kullanıcıları için en uygun yol problemini çözen bir web harita uygulaması geliştirmektir. Bu uygulama, yolun uzunluğu, yolun eğimi, yolun sınıfı, arazi kullanımı ve batarya doluluğu gibi faktörleri dikkate alarak farklı kombinasyonlarla en uygun yolun belirlenmesini sağlamaktadır. Bu kombinasyonlar sayesinde faktörlerin en uygun yolun belirlenmesine nasıl etki ettiği incelenebilecektir. Farklı kombinasyonlarda yapılan uygulamalarla, yolun uzunluğu, yolun eğimi, yolun sınıfı ve arazi kullanımı en uygun yol üzerinde nasıl değişikliklere yol açtığı ve scooterın performansı üzerindeki etkisi incelenmekte ve analiz edilmektedir. Bu analizlerin, elektrikli scooter kullanıcılarına yolculuklarında faktörleri nasıl optimize

edebileceklerine dair değerli bilgiler sunması beklenmektedir.

Uygulama, elektrikli scooter kullanıcılarının yolculuklarında zaman tasarrufu yapabilmeleri ve enerji verimliliğini artırabilmeleri için önemli bir araçtır. Uygulama batarya doluluğunu da dikkate alarak, kullanıcının seyahat sırasında bataryanın hangi konumda tamamen tükenerek scooterın kullanılamaz hale geleceğini gösterir. Bu özellik, kullanıcının seyahat planlamasını optimize etmesine ve bataryayı verimli bir şekilde yönetmesine yardımcı olur. Kullanıcılar, uygulama sayesinde seyahatlerini bataryalarının kapasitesine göre planlayabilir ve yol üzerindeki şarj istasyonlarını belirleyebilir.

1.3. Literatür Taramaları

Literatürde CBS tabanlı en kısa yol algoritmaları oldukça yaygın iken, elektrikli scooterlar için yapılan çalışmalar son dönemlerde önem kazanmaya başlamıştır. Bu araştırmalar, sürücülerin rotalarını planlamalarına yardımcı olan çeşitli yöntemler ve algoritmalar içermektedir. Aynı zamanda CBS'nin bu alanda ve içinde mekansal bilginin bulunduğu diğer alanlarda kullanımı, en kısa yol uygulamaları gibi araştırmalar da yoğunluktadır. Bu literatür taramaları, mevcut araştırmalar arasından incelenen derlemeleri sunarak, en son bulguları ve gelecekteki araştırmalar için potansiyel fırsatları vurgulayacaktır.

(Cao ve diğ., 2021) Araştırmalarında Singapur Merkez bölgesindeki elektrikli scooterların kısa mesafeli toplu taşıma yolculukları üzerindeki hizmetlerinin etkinliği araştırılmıştır. Elde edilen sonuçlar ile toplu taşıma tarafından karşılanamayan taleplerin elektrikli scooterlar ile karşılanabileceği, ancak bu uygulamalar için öncelikle elektrikli scooterların nerelere konuşlandırılacağı ve özel operatörler ile kamu refahı arasında uygun bir dengeye nasıl ulaşacağı gibi konuların tartışılması gerektiği çıkarımlarına varılmıştır.

(Zuniga-Garcia ve diğ., 2020) Araştırmalarında elektrikli scooterların kentsel altyapıda kullanımı incelenmektedir. Araştırmacılar, kullanıcılardan toplanan izleme verilerini kullanarak, elektrikli scooterların yaya kaldırımları, bisiklet şeritleri ve yollar gibi farklı yolları nasıl kullandığını ve bu kullanımın hangi faktörlere bağlı olduğunu analiz etmektedirler. Araştırmanın sonuçlarına göre, e-scooter'lar genellikle yaya kaldırımlarında kullanılmaktadır, ancak bunun nedeni genellikle yetersiz bisiklet şeridi

altyapısıdır. Araştırmacılar, e-scooter'ların kentsel altyapıya entegrasyonu için daha iyi yönetmelikler ve planlama stratejileri önermektedirler.

(Badia ve Jenelius, 2021) Araştırmalarında, paylaşımlı elektrikli scooterların seyahat davranışı, sürdürülebilirlik, altyapı, güvenlik ve politikalarını incelemeyi amaçlamaktadır. Araştırmacılar, elektrikli scooterların kullanımının artmasıyla birlikte, sürdürülebilirliği, çevresel etkileri ve trafik kazaları ile ilgili endişelerin arttığını belirtmektedirler. Araştırma, elektrikli scooterların farklı ülkelerdeki kullanım örüntülerini, sürücülerin ve kullanıcıların davranışlarını, kaza istatistiklerini, elektrikli scooterların trafik akışına etkilerini ve politikaların yönetimini incelemiştir. Araştırmanın sonuçlarına göre, elektrikli scooterların sürdürülebilirliği ve çevresel etkileri, enerji kaynaklarının türüne ve scooter'ların kullanım sıklığına bağlıdır. Elektrikli scooterların altyapısı, bisiklet yolları, yaya yolları ve trafik akışı ile uyumlu hale getirilerek daha güvenli hale getirilebilir. Elektrikli scooterların kullanımı, kullanıcıların eğitimi ve uygun politikaların uygulanmasıyla daha güvenli hale getirilebilir.

(Taş ve Akay, 2020) Araştırmalarında çok kriterli karar verme yöntemlerinden olan Analitik Hiyerarşi Prosesinin (AHP) kullanılarak erişilebilir orman alanlarının itfaiye yoluyla değerlendirilmesinde coğrafi bilgi sistemleri kullanılmıştır. Bu araştırmada itfaiyelerin tanımlanan bir yol ağı üzerindeki ulaşımaları yol ağı analizleri ile incelenmiştir.

(Sarı ve Sen, 2017) Araştırmalarında en uygun yol algoritması kullanılarak karayolu güzergah seçimine yönelik tasarımlar incelenmiştir. Birden fazla ve çeşitli maliyetlerin en uygun yol/rota hesabı üzerindeki etkileri ve sonuç hesaplamaları analiz edilmiştir.

(Sert ve diğ., 2017) Araştırmalarında en uygun toplu taşıma politikasına temel oluşturacak ulaşım ağları CBS'nin mekansal ağ analizi teknikleri kullanılarak belirlenmiştir. Araştırmanın amacı, toplu taşıma, toplu taşıma hizmetlerinin tüm kent insanı için kullanılabilir olmasıdır. Bu durumun sağlanması için kent insanının özelliklerinin anlaşılması ve bu konuda çokça verinin bir arada kullanılmasının gerektiği saptanmıştır.

(Gökcan ve diğ., 2020) Araştırmalarında katı atıklar arasındaki en kısa yolların incelenmesi için yapılan araç rotalama problemi çözümü içerisinde Dijkstra

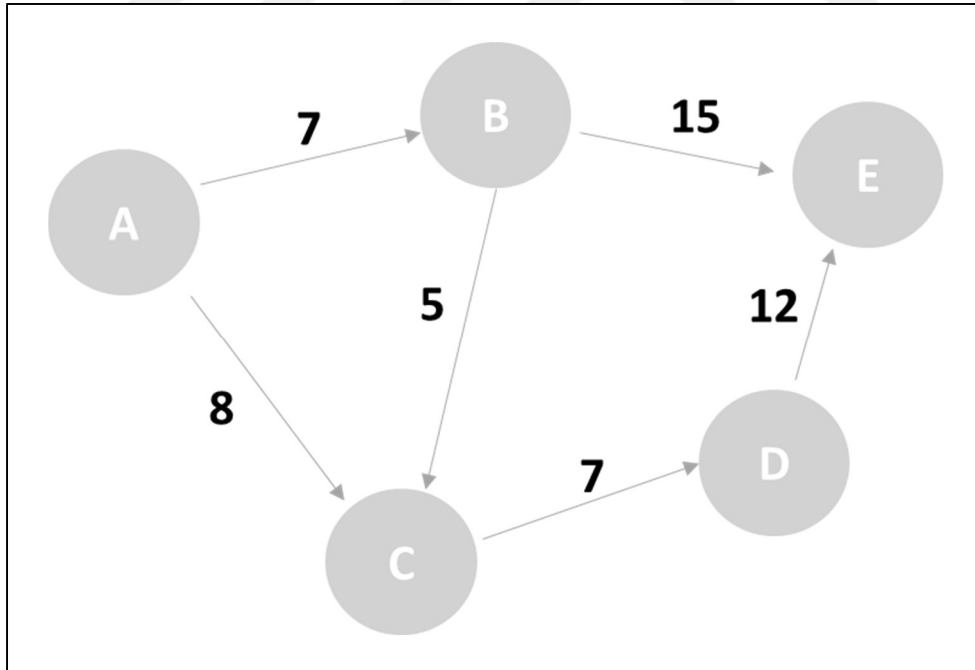
Algoritmasının kullanıldığı görülmektedir. Yol ağıları içerisindeki topolojik ilişkiler ile oluşturulan grafik modeli üzerinden yapılan en kısa yol uygulamalarında Dijkstra algoritmasının başarılı sonuçlar verdiği saptanmıştır.

(Plynnig ve diğ., 2016) Araştırmalarında bisikletler için uzunluk dışında farklı özniteliklerin de dikkate alınarak en uygun rotalar incelenmiştir. Öznitelikler en kısa yol algoritması içerisinde farklı maliyetler göz önüne alarak kullanılmış ve farklı kombinasyona sahip uygulamalar ile maliyetler üzerinde yapılan değişikliklerin bisiklet rotaları üzerindeki değişikliklerine dikkat çekilmiştir. Bu araştırma içerisinde farklı maliyetlerin aynı uygulamada nasıl kullanılabildikleri detaylıca işlenmiştir ve tez içerisinde de yoğunlukla bu araştırmanın yöntemlerinden faydalanılacaktır.

2. EN KISA YOL PROBLEMİ

En kısa yol problemi, bir ağ üzerinde iki nokta arasındaki en kısa yolun belirlenmesine odaklanan matematiksel bir problemidir. Bu problem, matematik, bilgisayar bilimi, fizik, mühendislik ve diğer birçok alanda yaygın olarak kullanılmaktadır (Güneri ve Durmuş, 2020; URL-1).

Problemin çözümü için genellikle grafik teorisi kullanılır. Grafikler, düğümler (node) ve kenarlardan (edge) oluşan matematiksel soyut yapılardır. En kısa yol problemlerinde ağ olarak da adlandırılabilirler. Düğümler birer nokta olarak, kenarlar ise bu noktalar arasındaki bağları tanımlamaktadır. Grafiğin yapısına bağlı olarak bu kenarlar çift yönlü, tek yönlü veya yönsüz olabilirler. En kısa yol hesaplamalarında kullanılmak üzere ağırlık adı verilen sayısal bir değere sahiptirler. Grafik teorisine göre iki nokta arasındaki en kısa yol, iki nokta arasındaki kenarların ağırlık değerleri toplamı en düşük değer olan kenarlar dizisidir (URL-2). Şekil 2.1. örneğinde görüldüğü üzere A düğümü ile E düğümü arasında kenar bağlantıları ve B, C ve D düğümleri mevcuttur.

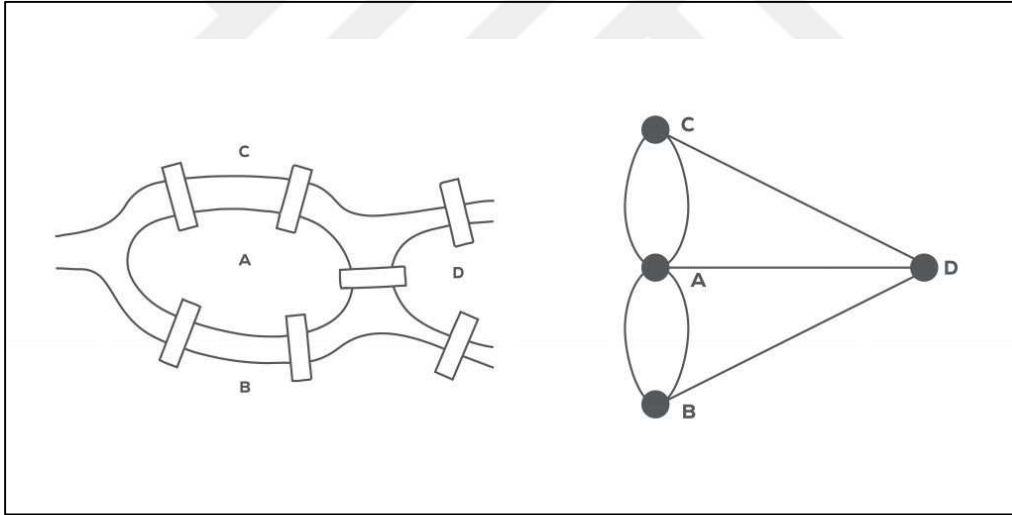


Şekil 2.1. Düğümleri arası ağırlıklı yönlendirilmiş örnek en kısa yol grafiği

Kenarlar tek yönlü olarak düğümleri birbirlerine bağlamaktadır. A düğümünden E düğümüne geçiş için izlenebilecek düğüm dizileri A-B-C-D-E, A-C-D-E ve A-B-E olmak

üzere 3 adettir ve kenar değerleri toplamaları sırasıyla 31, 27 ve 22'dir. En kısa yol hesaplamalarında kenar ağırlık değerleri toplamına literatürde maliyet adı verilmektedir. En kısa yol problemi çözümü için maliyeti en az olan dizi seçilmelidir. Bu bilgiler ışığında A-B-E düğüm dizisi A düğümünden E düğümüne yolların incelendiği bu grafik için en kısa yol anlamına gelmektedir (URL-3).

En kısa yol problemi, tarihte ilk kez 1736 yılında Euler tarafından matematiksel olarak formüle edilmiştir. Königsberg Köprüleri problemi adı verilen problemde, 7 köprü arasından her köprüyü yalnızca bir defa geçmek şartıyla en az sayıda yolculuk yapılan yolun seçilmesi gerekiyordu. Euler bu problemin çözümünde grafik teorisini kullanarak 4 düğüme ve 7 kenara sahip yönsüz bir grafik oluşturdu. Böyle bir yolculuğun yapılamayacağını matematiksel olarak kanıtlayarak çeşitli bulgular ortaya çıkardı (Gribkovskaia ve diğ., 2020). Şekil 2.2 üzerinden Königsberg Köprüleri probleminin grafik teorisine nasıl uyarlandığı incelenebilir.



Şekil 2.2. Königsberg köprüleri probleminin grafik teorisine uyarlanması (URL-4)

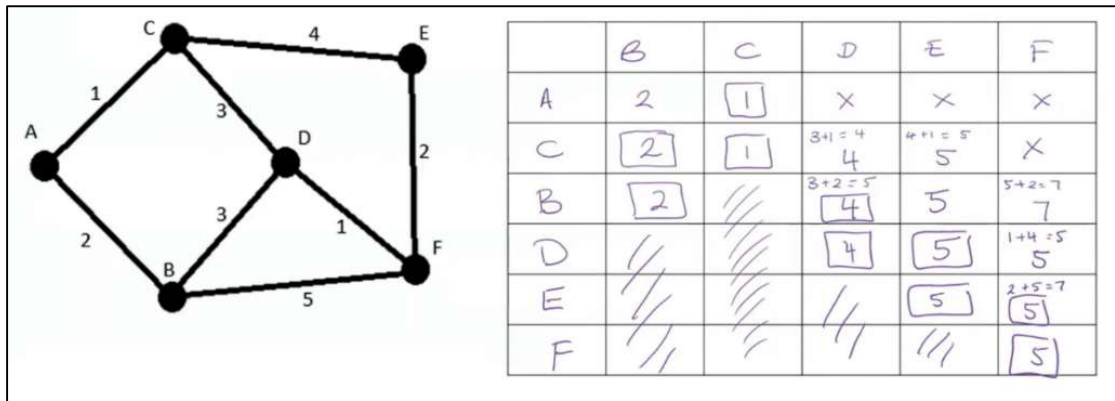
2.1. En Kısa Yol Problemi Algoritmaları

Zaman içinde en kısa yol probleminin çözümü için birçok algoritma geliştirilmiştir. Bunun nedeni ise her bir algoritmanın birbirlerine kıyasla çeşitli avantaj ve dezavantajlara sahip olmasıdır. Bu avantaj ve dezavantajlar genellikle en kısa yol problemi türlerine, algoritmaların kullanım amaçlarına, algoritmanın ilgili problem çözümündeki performansına ve ağ grafiğinin yön ilişkileriyle ilgili olabilmektedir (Elmasry ve Shokry, 2018; Tamimi, 2015; URL-1).

2.1.1. Dijkstra Algoritması

En kısa yol problemlerinde en sık kullanılan algoritmalarından biri olan Dijkstra algoritması bilim insanı Edsger Dijkstra tarafından 1956 yılında bir makalede ilk defa yayınlanmıştır. Temel prensip olarak başlangıç düğümünden herhangi bir düğüme olan en kısa yolu bulmak için kullanılır. Çok yönlü grafikler için kullanılabilmesiyle birlikte negatif ağırlıklı ağlarda çalışmamaktadır (Tamimi, 2015; Hofner ve Moller, 2012; Zhang ve diğ., 2013).

Algoritma için öncelikle bir başlangıç düğümü seçilir. Diğer tüm düğümlerin maliyet değerleri başlangıçta sonsuz olarak bir maliyet listesine eklenir. Başlangıç düğümünün komşu olan düğümlere mesafeleri hesaplanır ve maliyet listesi güncellenir. Maliyet listesinde en kısa maliyete sahip düğüm seçilerek bu düğüme geçiş yapılır. Aynı işlem bu düğüm için de tekrarlanarak eğer daha maliyetsiz bir yol bulunursa yeni maliyet kaydedilir. Bu işlem tüm düğümler üzerinde tekrarlanır. Her adımda en maliyetsiz yollar tercih edilir ve en kısa yol ağacı oluşturulur. En kısa yolların ve maliyetlerinin hesaplandığı bu ağaç başlangıçta seçilen düğümden diğer tüm düğümlere olan en kısa yolları ve maliyetleri tutacak şekilde düzenlenerek sonuç hesaplanır. Algoritma karmaşıklığı düğüm sayısı ile doğru orantılıdır (URL-5). Şekil 2.3 üzerinden Dijkstra algoritması ile en kısa yol problemi çözümünde sonuçların nasıl hesaplandığı incelenebilir.

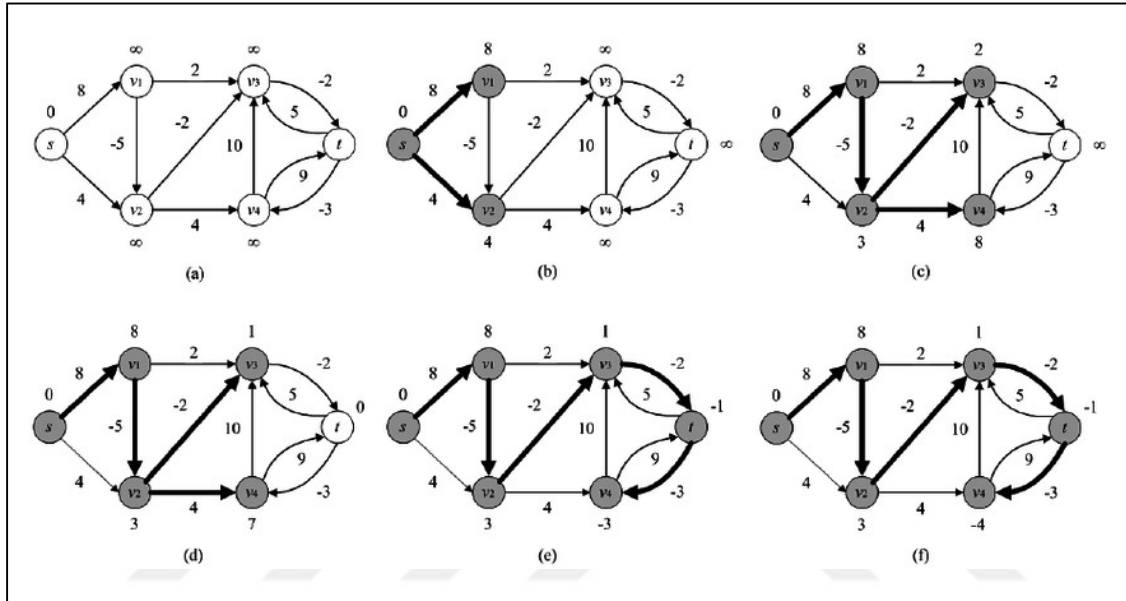


Şekil 2.3. Dijkstra algoritması ile en kısa yol problemi çözüm örneği (URL-6)

2.1.2. Bellman-Ford Algoritması

En kısa yol problemi çözümlerinde negatif ağırlıklı ağlarda da çalışabilen Bellman-Ford

algoritması, Dijkstra algoritmasından daha yetkin ancak verimliliği daha düşüktür. Başlangıçta hedef düğüme olan maliyetleri sonsuz kabul ederek tüm kenarları tek tek işler ve maliyetleri günceller. Dijkstra algoritmasından farklı olarak ağın yapısına bakarak, en maliyetsiz yolu bulana kadar yinelemeli olarak bir öncekinden daha iyi bir çözüm aramaktadır (Banerjee ve Kumar, 2022; URL-5). Şekil 2.4 üzerinden Bellman-Ford algoritmasının şematik prosedürüne örnek bir bakış yapılabilir.



Şekil 2.4. Bellman-Ford algoritmasının şematik prosedür örneği (URL-7)

2.1.3. A* Algoritması

A* (A star) algoritması, tek bir kaynaktan hedefe olan en kısa yolu bulmak için kullanılan bir arama algoritmasıdır. Algoritma öncelikle tahmini maliyet analizi yapar. Bunu yaparken “heuristik” fonksiyon kullanır. Bu yöntemin mantığı, tahmini en maliyetsiz yolların önceden hesaplanarak, daha sonra her düğümün gerçek maliyeti ile birlikte hesaplanmasıdır. Hedefe yakın durumlarda Dijkstra algoritmasından daha hızlı sonuçlar verirken, hedefe uzak durumlarda tahminleme fonksiyonlarından ötürü daha yavaş sonuç verecektir (Banerjee ve Kumar, 2022; URL-5).

2.1.4. Floyd-Warshall Algoritması

Floyd-Warshall algoritması, içerdiği döngüler aracılığıyla bir ağdaki tüm düğümlerin birbirleri ile arasındaki en kısa yolları analiz etmektedir. Ağın yönlü veya yönsüz olması

algoritmanın çalışmasına etki etmez. Çoğunlukla yoğun grafiklerde kullanışlıdır çünkü karmaşıklığı grafik sayısına bağlıdır (Banerjee ve Kumar, 2022).

2.2. En Kısa Yol Problemi Türleri

En kısa yol problemi türleri, başlangıç ve hedef düğümlerinin sayısı ve problemin amacıyla ilişkilidir. Problemin farklı senaryolarda nasıl ele alınacağına dair çeşitli yöntemler sağlar. En kısa yol problemleri dört grupta incelenmektedirler (Ekmen, 2020; Çölkesen, 2004).

2.2.1. Tek Kaynaklı En Kısa Yol Problemi

Tek kaynaklı en kısa yol problemi, bir başlangıç düğümünden diğer tüm düğümlere olan en kısa yolları bulmayı amaçlayan bir en kısa yol problemi türüdür. Gündelik hayatta nakliye işleri, seyahat planlaması gibi farklı uygulamalarda bu problem ile karşılaşılabilir. Dijkstra algoritması ve A* algoritması gibi farklı algoritmalarla çözülebilmektedir.

2.2.2. Tek Çift En Kısa Yol Problemi

Tek çift en kısa yol problemi, bir grafik üzerinde iki düğüm arasındaki en kısa yolun bulunmasını amaçlayan en kısa yol problemi türüdür. Bu problemin çözümünde iki düğümden biri başlangıç düğümü, diğeri hedef düğümü olarak tanımlanır ve yalnızca bu iki düğüm arasındaki en kısa yol hesaplanması amaçlanır. Gündelik hayatta belirli bir konumdan hedef başka bir konuma en kısa yolu bulmayı amaçlayan tüm uygulamalarda karşılaşılabilir. Dijkstra algoritması ve Floyd-Warshall algoritması gibi farklı algoritmalarla çözülebilmektedir. Ancak Floyd-Warshall algoritması tüm çiftler arasındaki en kısa yolları bulduğu için performans açısından pek tercih edilmeyebilir.

2.2.3. Tek Hedef En Kısa Yol Problemi

Tek kaynaklı en kısa yol probleminin bir benzeri olarak bir hedef düğümü için diğer tüm düğümlerden en kısa yolların hesaplandığı problem türüdür. Dijkstra Algoritması, Bellman-Ford algoritması gibi algoritmalarla çözülebilmektedir. Ancak bu algoritmaları kullanmadan önce ağırlık değerlerinin negatif olup olmadıklarına dikkat edilmesi gerekmektedir.

2.2.4. Tüm Çiftler En Kısa Yol Problemi

Bir grafik üzerindeki herhangi iki nokta arasındaki en kısa yol uzunluklarının hesaplanmasını amaçlayan en kısa yol problemi türüdür. Örneğin şehirler arası ulaşım ağındaki herhangi iki şehir arasındaki en kısa yolların hesaplanması gibi farklı uygulamalarda kullanılabilinmektedir. Bu problemde tüm düğümlerin birbirleri ile arasındaki en kısa yolların hesaplanmasının amaçlanmasından ötürü Floyd-Warshall algoritmasının kullanılması daha uygun olabilir.

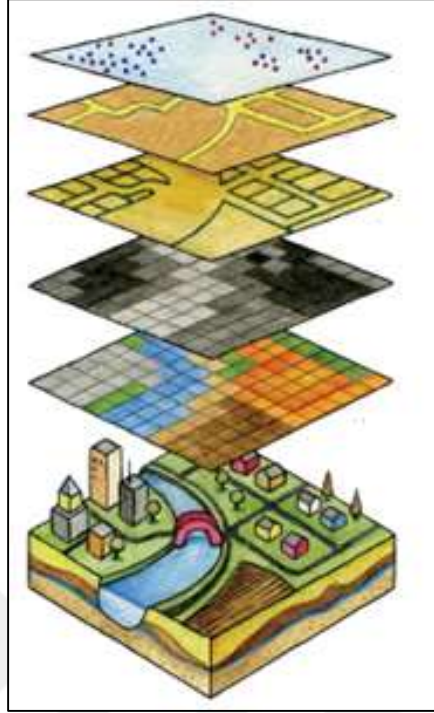
2.3. En Uygun Yol Problemi

En uygun yol problemi, bir ağdaki iki düğüm arasındaki en kısa yolun yanı sıra, farklı faktörlerin ve/veya kısıtlamaların da etkisiyle istenen şartlar altındaki en uygun yolun da hesaplanmasını amaçlayan en kısa yol problemi türüdür. En kısa yol problemlerinde genellikle tek bir kriter dikkate alınırken, en uygun yol problemlerinde maliyet hesaplamalarında farklı faktörler de devreye girmekte ve çözüm için optimizasyon işlemleri yapılmaktadır. Örneğin bir taşıt ile şehir içinde yapılacak seyahatte, iki konum arasında yalnızca uzaklık gibi mesafe değerleri dikkate alındığında en kısa yol bulunabilmektedir. Ancak trafik yoğunluğu, zaman gibi ek faktörlerin dikkate alınması ile problem sahibi için en kısa yol problemi artık en uygun yol problemine dönüşmektedir. Bu nedenle en uygun yol problemi, en kısa yol probleminden daha karmaşık ve çeşitlidir. Özellikle ulaşım ağı, telekomünikasyon, enerji yönetimi gibi birçok alanda kullanılmaktadır (Zhou ve Gao, 2019).

2.4. Coğrafi Bilgi Sistemleri

Coğrafi Bilgi Sistemleri (CBS) ya da ingilizce adıyla Geographic Information System (GIS), coğrafi verilerin toplanması, depolanması, işlenmesi, yönetimi, mekânsal analizi, sorgulaması ve sunulması fonksiyonlarını yerine getiren donanım, yazılım, personel, coğrafi veri ve yöntem bütünüdür (Fidan ve Ulvi, 2022). Akademisyenlere göre yer-referanslı bilgi sistemlerinin tamamını içeren ve bunları irdeleyen bilimsel bir kavram, uygulayıcılara göre harita bilgilerini elektronik ortamda sunabilen bilgisayar tabanlı bir araç, idarecilere göre de kurumsal organizasyonlara destek veren bir veri tabanı yönetim sistemidir. Şekil 2.5 üzerinden coğrafi bilgi sistemleri ile oluşturulabilecek bir katman

yapısı örneği incelenebilir.



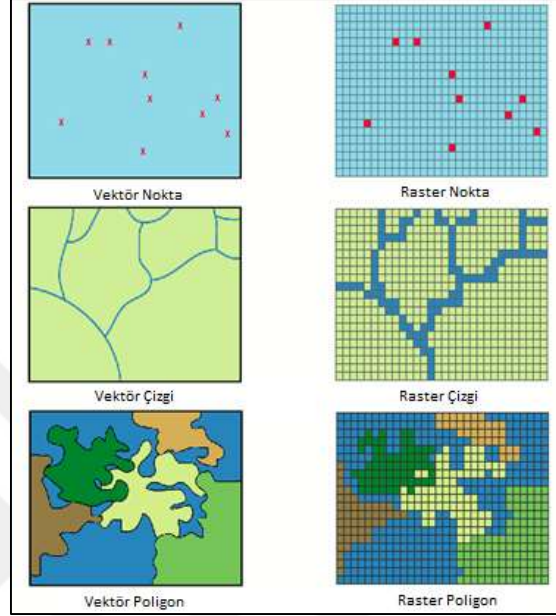
Şekil 2.5. Coğrafi bilgi sistemleri katman yapısı örneği (URL-8)

2.4.1. CBS Veri Yapıları

Coğrafi Bilgi Sistemleri içerisindeki verilerin birçok farklı formatı vardır. Veriler sözel veya grafik olarak görüntülenebilirler. Ancak temel olarak 3 tür veri tipi vardır (URL-9). Bunlar:

- Coğrafi olmayan öznitelik verileri: Bu veriler, yeryüzü veya bir referans sistem üzerinde herhangi bir konumu olmayan nesnelerin özelliklerini içeren verilerdir. Bu veriler grafik olarak görüntülenemezler.
- Vektör veriler: Harita üzerinde x ve y değerleri üzerinden koordinat tanımlanarak konumları tespit edilebilen verilerdir. Bu verilerin özniteliklerinde kendilerine has veya başka verilerle ilişkilendirilebilecek özellikler bulunabilir. Vektör veriler temelde üçe ayrılır. Bunlar Nokta verisi, Çizgi verisi ve Alan verisidir.
- Raster veriler: Birbirine bitişik (komşu) görüntü hücrelerinin (piksel) gruplanarak bir araya gelmesiyle oluşan verilere denir. Hücrenin boyutu ile haritanın çözünürlüğü birbiriyle ilişkilidir. Hücrelerin boyutları ne kadar büyük olursa harita çözünürlüğü de o kadar küçük olur.

Şekil 2.6 üzerinden vektör ve raster veri formatlarının karşılaştırmaları görülebilmektedir. Görüldüğü üzere raster veri formatında vektör veri formatındaki veri gösterimine nazaran verileri hücreler içerisinde tutarak göstermektedir. Vektör veri formatında ise verinin ham konumları görülebilmektedir.



Şekil 2.6. Raster ve vektör katmanlarının gösterimi örneği (URL-10)

CBS'nin birbirini tamamlayan 4 bileşeni vardır (URL-9). Bunlar:

- Donanım ve yazılım araçları,
- Coğrafi Veriler,
- Personel,
- Belirlenmiş bir problem için amaç.

Bu bileşenlerin her biri birbirleriyle ilişkilidir.

2.4.2. CBS'nin Kullanıldığı Alanlar

CBS genel itibar ile konum, şart ve durum, değişim, güzergâh, şekil ve modelleme vb. gibi konularla ilgilenebilmekte ve bu konulara çözümler getirebilmektedir. CBS kullanım alanları neredeyse sınırsız olmakla birlikte maddeler halinde örneklendirilmek istenirse;

- Elektrik/Gaz işletimi,
- Ticaret,

- Telekomünikasyon,
- Taşımacılık,
- Maden/Petrol arama,
- Boru hatları,
- Su ve atık su,
- Tıp/Sağlık hizmetleri,
- Hükümet faaliyetleri,
- Yerel yönetimler,
- Jeoloji/Yer bilimleri,
- Perakendecilik,
- Askeri/İstihbarat,
- Arazi kullanımı,
- Çevre yönetimi,
- İmar ve Kadastro,
- Ziraat/Tarım,
- Ormancılık,
- Risk yönetimi,
- Eğitim,
- Spor alanları

olarak gösterilebilir (URL-11).

CBS teknolojisi haritalamada, telekom ve ağ servislerinde, kaza ve sorunlu bölge analizlerinde, kentsel planlamada, ulaşım planlamasında, çevresel etki analizinde, tarımsal uygulamalarda, afet yönetimi ve önlenmesinde, heyelan tehlikesi bölgelemede, arazi kullanımı ve yüzey değişikliklerinde, navigasyonda, sel hasarı tahmininde, doğal kaynak yönetiminde, bankacılık sektöründe, toprak haritalamada, dijital vergilendirmede, yersel ölçmede, sulak alan haritacılığında, jeolojide, kömür madenlerinde yangın önlenmesinde, varlık yönetimi ve bakımında, planlama ve toplum yaşamını geliştirmede ve diğer bir çok alanda kullanılabilir (URL-11).

2.4.3. CBS’de Topoloji Kavramı

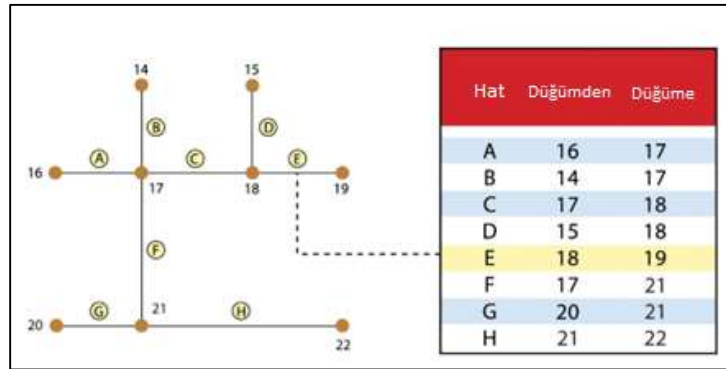
CBS’de topoloji, coğrafi verilerin ilişkilerini ve bağlantılarını tanımlayan bir konsepttir. Topolojik ilişkiler, coğrafi veriler arasındaki geometrik ilişkileri ve bağlantıları ifade

etmektedir. Coğrafi nesnelerin birbirleriyle olan etkileşimlerini, konumlarını ve bağlantılarını tanımlar. CBS’de temel olarak üç tür topoloji vardır. Bunlar nokta, alan ve ağ topolojileridir. Bu topoloji türleri yardımıyla en kısa yol analizleri, alan kesişimleri ve tampon oluşturma gibi işlemler gerçekleştirilebilir (URL-12; URL-13).

CBS’de topoloji 4 veri tablosunda saklanmaktadır. Bunlar nokta topolojisi, çizgi topolojisi, çokgen topolojisi ve koordinat tablosudur. Bunun yanında bir topoloji 3 şekilde oluşabilmektedir. Bunlar hat-düğüm topolojisi, poligon-hat topolojisi ve sol-sağ topolojisidir (URL-13).

2.4.3.1. Hat-Düğüm (Arc-Node) Topolojisi

Bağılılık topolojisi, CBS’de coğrafi verilerin bağlantılarını ve ilişkilerini temsil etmek için kullanılan bir topoloji türüdür. Hat-Düğüm topolojisi, çizgilerin (arcs) ve düğümlerin (nodes) birbirleriyle olan ilişkilerini tanımlamaktadır. Burada çizgiler (hatlar), bir ağ analizinde kenarları, CBS’de yolları, nehirleri, demiryolları gibi öğeleri temsil edebilmektedir. Düğümler ise çizgilerin birleştiği veya kesiştiği noktalardır. Çizgilerin başlangıç ve/veya bitiş noktalarını ifade etmektedir. Hat-Düğüm topolojisi bir topolojideki çizgilerin birbirleri arasındaki ilişkilerini ve geometrik bağlantılarını temsil etmektedir (Karaş ve Batuk, 2005). Şekil 2.7 üzerinden örnek hat-düğüm topolojisi ve temsil listesi incelenebilir.

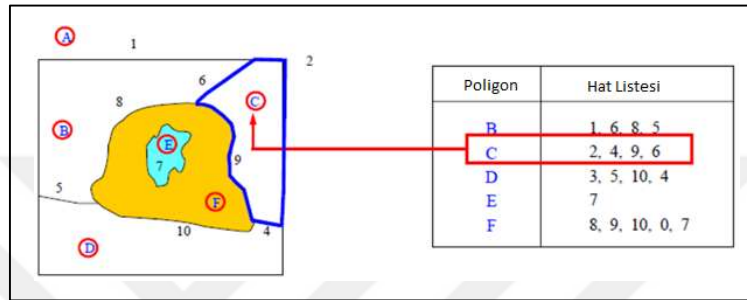


Şekil 2.7. Örnek hat-düğüm topolojisi ve temsil listesi (URL-14)

2.4.3.2. Poligon-Hat (Polygon-Arc) Topolojisi

Poligon-Hat topolojisi, CBS’de poligonlar ve bu poligonların aralarındaki çizgilerin (hatların) arasındaki topolojik ilişkileri temsil etmektedir. Bu topolojilerde poligonların

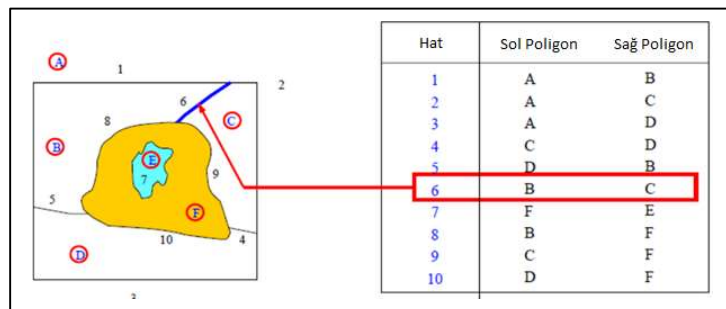
sınırları çizgilerden oluşmaktadır. Burada poligonlar CBS’de gölleri, parkları, parselleri temsil edebilir. Bir poligon bir veya daha fazla çizginin birleşmesiyle oluşmaktadır. Çizgiler ise bir veya daha fazla noktanın birleşmesiyle oluşur ve poligonların sınırlarını belirlemektedir. Bir poligona ait tüm hatlar numaralandırılarak temsil edilir. Bu topoloji modeli, örtüşme, içermeye, bitişiklik ve boşluk gibi poligonlar arasındaki ilişkileri tanımlamaktadır (Karaş ve Batuk, 2005; URL-14). Şekil 2.8 üzerinden örnek poligon-hat topolojisi ve temsil listesi incelenebilir.



Şekil 2.8. Örnek poligon-hat topolojisi ve temsil listesi (URL-14)

2.4.3.3. Sol-Sağ (Left-Right) Topolojisi

Sol-Sağ topolojisi CBS’de bir hat ağı veya yol ağı için kullanılan bir topooloji modelini ifade etmektedir. Bu modelde, çizgiler (hatlar) arasındaki topolojik ilişkileri bir çizginin sol ve sağ tarafına olan konumlarına dayalı olarak belirlemektedir. Sol ve sağ taraflar, çizginin yönünü temsil etmekle beraber, bir çizginin başlangıç noktasından bitiş noktasına doğru ilerlerkenki sol ve sağ bölgede kalan taraflardır. Sol-Sağ topolojisi, çizgiler arasındaki topolojik ilişkileri sol ve sağ tarafına olan konumlarıyla tanımlamaktadır. Bu ilişkiler, örneğin bir çizginin solunda veya sağındaki diğer çizgilerle olan kesişme, bitişiklik veya bağlantı durumlarını ifade eder (Karaş ve Batuk, 2005; URL-14). Şekil 2.9 üzerinden örnek sol-sağ topolojisi ve temsil listesi incelenebilir.



Şekil 2.9. Örnek sol-sağ topolojisi ve temsil listesi (URL-14)

2.4.4. En Uygun Yol Uygulamalarında CBS'nin Yeri

En uygun yol uygulamalarında CBS, yolculuk planlaması, trafik akışı, rota optimizasyonu ve taşımacılık gibi konularda büyük bir rol oynar. CBS, yol ağları ve trafik verilerinin etkili bir şekilde analiz edilmesine, işlenmesine ve görselleştirilmesine yardımcı olan bir araçtır. Elde edilen coğrafi verilerin analizi ve kullanımı, en uygun yol uygulamalarında önemli bir rol oynamaktadır (URL-15). Şekil 2.10 üzerinden en uygun yolun CBS ile görselleştirilmesi örneği incelenebilir.



Şekil 2.10. En uygun yolun CBS ile görselleştirilmesi örneği (URL-16)

2.4.5. QGIS Yazılımı

QGIS (Quantum GIS), açık kaynak kodlu bir CBS yazılımıdır. CBS verilerini düzenleme, haritalama ve analiz etme yeteneklerine sahiptir. Kullanıcıların CBS verilerini ve haritalarını görselleştirmelerine, yönetebilmelerine ve analiz etmelerine olanak tanır. Birçok farklı veri formatını desteklemekle birlikte açık kaynak kodlu olması nedeniyle ücretsiz olarak kullanılabilir (URL-17). Şekil 2.11 üzerinden QGIS yazılımının logosu incelenebilir.



Şekil 2.11. QGIS yazılımı logosu (URL-17)

2.4.6. OpenStreetMap

OpenStreetMap (OSM), dünyanın dört bir yanındaki insanların, yerlerini ve çevrelerindeki coğrafi özellikleri kaydederek oluşturdukları açık kaynaklı bir coğrafi bilgi sistemidir. Kullanıcılar, caddeler, yollar, binalar, parklar, nehirler ve hatta posta kutuları gibi pek çok farklı coğrafi özellikleri işaretleyebilir ve bu veriler daha sonra ücretsiz olarak kullanılacak haritalar, uygulamalar ve diğer coğrafi veri hizmetleri için kullanılabilir. OSM verileri, yüksek maliyetli CBS verilerine alternatif olarak kullanılabilir ve daha az gelişmiş bölgelerde de yüksek kaliteli coğrafi veriler sağlayabilir (URL-18). Şekil 2.12 üzerinden OSM logosu incelenebilir.



Şekil 2.12. OSM logosu (URL-19)

2.5. Python Programlama Dili

Nesne Tabanlı bir programlama dili olan Python, Guido van Rossum tarafından 1980 yılında ortaya çıkartılmıştır. Python, pek çok farklı kullanım alanı için ideal bir dil olarak kabul edilir. Günümüzde web uygulamaları, yapay zekâ çalışmaları, veritabanı yazılım programlama gibi birçok alanda popüler olarak kullanılmaktadır. Yüksek bir taşınabilirlik özelliğine sahiptir, yani bir kez yazılan kodun birçok farklı platformda çalışması mümkündür. Basit sözdizimleri sayesinde kolay kullanılabilmesinin yanında, kullanımının bu kadar yüksek olma nedenlerinden biri ise, içerisine yüklenebilen çok sayıda ve oldukça işlevsel kütüphanelere sahip olması denilebilir (URL-20). Şekil 2.13 üzerinden python programlama dili logosu incelenebilir.



Şekil 2.13. Python programlama dili logosu (URL-20)

2.5.1. Python Kütüphaneleri

Python, geniş bir standart kütüphaneye sahip olsa da üçüncü taraf kütüphaneleri de kullanarak programlama deneyimini geliştirmek mümkündür. Python kütüphaneleri, özellikle belirli kullanım amaçlarına yönelik önceden yazılmış fonksiyonlar ve sınıflar içeren modüllerdir. Bu fonksiyon ve sınıflar projelere entegre edilerek birer araç olarak kullanılabilmekte ve projelerdeki yoğun iş gücünü önemli ölçüde azaltabilmektedirler.

2.5.1.1. Pandas

Verilerin pratik olarak hızlı ve verimli bir şekilde kullanılmasına olanak sağlayan Pandas kütüphanesi günümüzde özellikle veri bilimi alanında oldukça sık kullanılmaktadır. Veri analizi için kullanılan bir kütüphanedir. Veri tabanlarından veri okuma, verileri düzenleme, filtreleme, gruplama ve toplama işlemlerini kolaylaştırır (Temiz, 2022).

2.5.1.2. Geopandas

Geopandas, coğrafi verileri temsil etmek için geometrik nesneler (örneğin noktalar, çizgiler ve çokgenler) kullanan GeoDataFrame adlı bir veri yapısı sağlar. GeoDataFrame'ler, standart Pandas DataFrame'lerine benzer şekilde kullanılabilir, ancak coğrafi işlevlerle birlikte çalışacak şekilde tasarlanmıştır. Bu kütüphane, Pandas veri analizi kütüphanesinin bir uzantısıdır (URL-21).

2.5.1.3. NetworkX

NetworkX, Python programlama dili için bir grafik teorisi kütüphanesidir. Bu kütüphane, ağların oluşturulması, analizi ve görselleştirilmesi için çeşitli araçlar sağlar. Yüksek düzeyde programlama yetenekleri sunar ve diğer Python kütüphaneleriyle kolayca entegre edilebilir. Özellikle, veri analizi ve makine öğrenimi gibi disiplinlerde, ağlarının modellenmesi, analizi ve görselleştirilmesi için kullanılabilir. (Chen ve Chen, 2020; URL-22).

2.5.1.4. OSMnx

OSMnx, OSM'nin sunmuş olduğu mekansal verileri kullanarak NetworkX kütüphanesi gibi ağ analizleri yapılabilmesini sağlayan bir python kütüphanesidir. OSM verilerinin

indirilmesine, yol-sokak verilerinin modellenmesine, ilişkilerinin oluşturulmasına, analiz edilmesine, görselleştirilmesine ve daha birçok mekansal analize olanak tanımaktadır. Ayrıca, açık kaynaklı olması ve geniş bir kullanıcı topluluğuna sahip olması, kütüphanenin gelişimine katkıda bulunmak için birçok olanak sunar (URL-23).

2.5.2. Flask

Flask, Python programlama dili için bir web uygulama çatısıdır. Web uygulamaları geliştirmek için birçok araç ve özellik içerir. Flask, hafif ve esnek yapısı ile diğer web çatılarına göre daha az kısıtlayıcıdır ve daha az kod yazarak web uygulamaları oluşturmanızı sağlar. Flask, URL yönlendirme, HTTP istek-cevap işleme, form verilerinin işlenmesi, HTML şablonlarının kullanımı gibi temel işlevleri içerir. Ayrıca Flask, birçok eklenti (plugin) ve özellik sunan geniş bir topluluğa sahiptir. Flask, birçok işlevi desteklemek için kullanılan üçüncü parti kütüphaneleri (library) kullanmaya izin verir (URL-24).

2.6. JavaScript Programlama Dili

JavaScript, web sayfalarını interaktif hale getirmek için kullanılan bir programlama dilidir. Bu dil sayesinde, web sayfalarında animasyonlar, kullanıcı etkileşimi, form doğrulama gibi dinamik özellikler eklenerek web deneyimi artırılır. JavaScript'in en önemli özelliklerinden biri, tarayıcı tarafında çalıştığı için sayfa yenilenmeden dinamik işlemler yapabilmesidir. Bu sayede, web sayfalarının hızı ve performansı artırılabilir. Ayrıca, JavaScript ile oluşturulan uygulamaların birçoğu modern tarayıcılarda desteklenen tüm platformlarda kullanılabilir (URL-25). Şekil 2.14 üzerinden JavaScript programlama dili logosu incelenebilir.



Şekil 2.14. JavaScript programlama dili logosu (URL-26)

2.6.1. OpenLayers

OpenLayers, web tabanlı haritalama uygulamaları oluşturmak için kullanılan bir JavaScript kütüphanesidir. Bu kütüphane, kullanıcıların etkileşimli haritalar oluşturmasına ve bu haritalarda veri görselleştirmeleri yapmasına olanak tanır. Açık kaynak kodlu bir projedir ve kullanıcılar tarafından geliştirilebilir. Bu sayede, haritalama uygulamalarının geliştirilmesi için gerekli olan birçok özellik ve araç, OpenLayers tarafından sağlanır. Kütüphane, birçok farklı harita sağlayıcısıyla uyumlu olarak çalışır ve çeşitli veri formatlarını destekler (URL-27). Şekil 2.15 üzerinden OpenLayers logosu incelenebilir.



Şekil 2.15. OpenLayers logosu (URL-26)

2.7. API Entegrasyonu

API entegrasyonları, istemciler ile sunucular arasındaki verilerin otomatik olarak güncelleyenmesine olanak sağlayan yazılım bileşenleridir. Web Uygulamaları günümüzde farklı servis sağlayıcılarından veya harici API'lerden veri alışverişi yapmak için API entegrasyonunu kullanır. API (Application Programming Interface), bir yazılımın başka bir yazılımla iletişim kurmasına olanak sağlayan bir ara yüz sağlar. Arka uç (Backend) servisleri, web uygulamasının veri alışverişini yönetmek ve farklı kaynaklardan veri almak veya veri göndermek için API entegrasyonunu kullanır. API'ler işletmelerin, şirketlerin, kurumların vb. kuruluşların iş operasyonlarının yönetilmesi ve düzenlenmesi konusunda yenilikçi çözümler geliştirilmesini sağlamaktadır (URL-28; URL-29).

Yaygın kullanılan API mimari türleri 4 adettir. Bunlar Restful, SOAP, WebSocket ve RPC API'leridir (URL-28).

- RPC API'leri: Uzaktan prosedür çağrısı (Remote Procedure Call) anlamına gelmektedir. Bu mimaride, bir istemci uzak sunucuda bulunan bir prosedürü çağırır ve sonuçları alır. Bu, bir istemcinin uzaktaki bir sunucudaki iş mantığını kullanmasına olanak tanır.
- SOAP API'leri: SOAP (Simple Object Access Protocol), xml tabanlı bir protokoldür ve mesajlaşma için karmaşık yapılar kullanmaktadır. Bu mimari, ağ üzerinde farklı sistemler arasında iletişimi sağlamak için genellikle Web Servisleri için kullanılır.
- WebSocket API'leri: Veri aktarımı için JSON nesneleri kullanır. Sunucu ve istemci (arayüz, kullanıcı vb.) arasında çift yönlü iletişimi desteklemektedir.
- Rest API'leri: REST (Representational State Transfer), web hizmetleri için popüler bir API mimari türüdür. Restful API'leri, kaynakları temsil eden URL'ler üzerinden http protokolü ile iletişim kurar. Bu mimari, kaynakların durumunu değiştirme veya almak için http metodlarını kullanır.

2.7.1. Rest API

Temsili Durum Aktarımı anlamına gelen (Representational State Transfer) REST, web hizmetlerinin temel prensiplerine dayanan bir mimari stildir. RESTful API, bu prensiplere uygun olarak tasarlanmış bir API'dir. Bu mimari stil, web kaynaklarını temsil eden URL'ler üzerinden iletişim kurmayı ve HTTP protokolünü kullanmayı vurgular. RESTful API, genellikle JSON veya XML gibi veri formatlarını kullanarak veri alışverişi yapar. RESTful API, HTTP metodlarını (GET, POST, PUT, DELETE vb.) kullanarak kaynaklarla etkileşime geçer. Her HTTP metodu, belirli bir eylemi gerçekleştirmek için kullanılır. Örneğin, GET metodu bir kaynağın alınması için, POST metodu yeni bir kaynağın oluşturulması için kullanılabilir (URL-28).

2.7.2. HTTP Metotları

HTTP (Hypertext Transfer Protocol) metotları, bir sunucudan kaynakları almak, kaynakları güncellemek, yeni kaynaklar oluşturmak veya kaynakları silmek gibi çeşitli işlemleri gerçekleştirmek için kullanılır. Bu metotlar, web tarayıcıları, web hizmetleri ve diğer HTTP istemcileri tarafından kullanılır (URL-30).

- GET: Belirtilen bir kaynağın alınması için kullanılır. Sunucudan bir kaynağın temsilini (veri) istemek için kullanılır. GET isteği, genellikle veri okuma işlemleri için kullanılır ve sunucu isteği karşılarak istemciye ilgili kaynağın temsili döndürülür.
- POST: Yeni bir kaynağın oluşturulması veya mevcut bir kaynağın işlenmesi için kullanılır. İstemci, sunucuya veri gönderir ve sunucu bu veriyi işler veya yeni bir kaynak oluşturur. POST isteği genellikle veri oluşturma, form gönderme veya veri yükleme işlemlerinde kullanılır.
- PUT: Belirli bir kaynağın güncellenmesi veya değiştirilmesi için kullanılır. İstemci, güncellenmiş kaynağı sunucuya gönderir. Eğer kaynak zaten varsa, güncellenir; eğer kaynak yoksa, sunucu tarafından yeni bir kaynak oluşturulur.
- DELETE: Belirli bir kaynağın silinmesi için kullanılır. Sunucuya, silinmek istenen kaynağın URL'sini göndererek kaynağın silinmesi talep edilir.
- PATCH: Belirli bir kaynağın kısmi güncellenmesi için kullanılır. İstemci, güncellenmiş kaynağın sadece değişen bölümlerini sunucuya gönderir. Diğer bir deyişle, kaynak üzerindeki belirli alanlar güncellenirken, diğer alanlar dokunulmaz kalır.
- HEAD: Belirtilen bir kaynağın başlık bilgilerinin alınması için kullanılır. HEAD isteği, GET isteği ile benzerdir, ancak sadece kaynağın başlık bilgilerini alır ve gerçek içeriği döndürmez. Bu istek, kaynağın varlığını doğrulamak veya başlık bilgilerini almak için kullanılabilir.
- OPTIONS: Bir sunucunun desteklediği HTTP metotlarını, kabul ettiği başlık alanlarını ve diğer ilgili bilgileri almak için kullanılır. OPTIONS isteği, sunucunun belirli bir kaynak üzerinde hangi işlemleri desteklediğini kontrol etmek için kullanılabilir.

2.8. Elektrikli Scooter Kullanımı

Elektrikli scooter, elektrik motoru tarafından çalıştırılan ve çoğunlukla şehir içi kullanım için tasarlanmış kişisel bir taşıt aracıdır. Genellikle iki tekerlekli ve bir bisiklete benzer şekildedir ancak bazı modeller üç tekerlekli olabilir (URL-31).

Dünyada iklim değişikliği ve küresel ısınma gibi çeşitli tehlikeler arttıkça yenilenebilir ve doğa dostu enerji kullanımının önemi de dikkat çekmeye başlamıştır. Gündelik yaşamda kullanılan taşıtlar atmosfere binlerce ton karbondioksit salınımı yapmaktadır.

Ulaşımında temiz elektrikli araç kullanımı da bu nedenle önemlidir. Ulaşım konusunda kolay kullanıma sahip olan elektrikli scooterların piyasaya çıkması ve popülerleşmesinin nedenlerinden biri de çevre dostu olmasıdır (URL-31).

Elektrikli scooterların maksimum alabilecekleri mesafe, motor özellikleri, batarya dolulukları, tekerler, ağırlık, yol kalitesi ve modelleriyle ilişkilidir. Dolaylı olarak sürtünme ilişkisinden de söz edilebilir. Farklı kaynaklardan elde edilen bilgilere göre ortalama modern bir elektrikli scooter tamamen dolu bir batarya ile tek seferde 5-50 kilometrelik bir mesafe gidebilmektedir (URL-31; URL-32; URL-33; URL-34; URL-35). Şekil 2.16 üzerinden örnek bir elektrikli scooter modeli çizimi incelenebilir.



Şekil 2.16. Örnek elektrikli scooter modeli (URL-36)

2.9. Elektrikli Scooterlar İçin En Uygun Yol Problemi ve Faktörleri

Elektrikli scooterlar şehir için ulaşım için pratik ve çevre dostu bir seçenek olsa da kullanıcıların trafik kurallarına uymaları ve güvenli seyahat etmek için dikkatli olmaları gerekmektedir. Aynı zamanda trafiğe çıkan diğer araçlarda olduğu gibi çeşitli en uygun yol faktörlerinin dikkate alınması gerekmektedir. Bu faktörlere bazı örnekler verilebilir:

- Yol Uzunluğu: En temel en uygun yol problemi faktörlerinden biri olan yol uzunluğu, en uygun yol seçimlerinde yolun mesafesini dikkate alan problemlerde etki dikkate alınmaktadır.
- Trafik Yoğunluğu: Şehir merkezlerinde trafik yoğunluğu oldukça yüksek olabilir ve bu, elektrikli scooter kullanıcılarının hızını ve ilerleyişini etkileyebilir.
- Yol Sınıfı: Yol durumu, yol kalitesi gibi anlamlara da gelebilecek bir faktör olan yol sınıfı, bozuk veya kalitesiz yollar gibi elektrikli scooter kullanıcıları için zorlu seyahat deneyimleri oluşturabilecek durumlar için göz önüne alınmalıdır.

- Yolun Eğimi: Elektrikli scooterların model ve özelliklerine bağlı olarak yüksek eğimli yollarda yukarı yönde aracın yolu tırmanamaması gibi durumlarla karşılaşılabilir. Aşağı yönde ise oluşabilecek yüksek hız gibi durumlardan ötürü kullanıcılar elektrikli scooterın hakimiyetini kaybedebilir ve çeşitli kazalara neden olabilir.
- Batarya Doluluğu: Elektrikli scooterların maksimum alabileceği mesafeler için dikkat edilmesi gereken bir özelliktir. Yol uzunluğu, yol sınıfı ve yolun eğimi gibi diğer faktörlerin de üzerinde etkisinin olduğu bir faktördür.
- Hava şartları: Yağmurlu havalarda elektrikli scooterlar için yağmur birikintisinin bulunduğu yollar sürüş kalitesini azaltmaktadır.
- Arazi kullanımı: Yolun bulunduğu bölgedeki arazinin yapısı, uygun yol arayışlarında kullanıcının isteği doğrultusunda önemli rol oynayabilmektedir. Örneğin bir elektrikli scooter kullanıcısı binaların yoğun olduğu yerleşim alanları yerine yeşilin bol olduğu park, bahçe gibi alanları tercih ederek seyahat veya eğlence amaçlı araç kullanıyor olabilir.

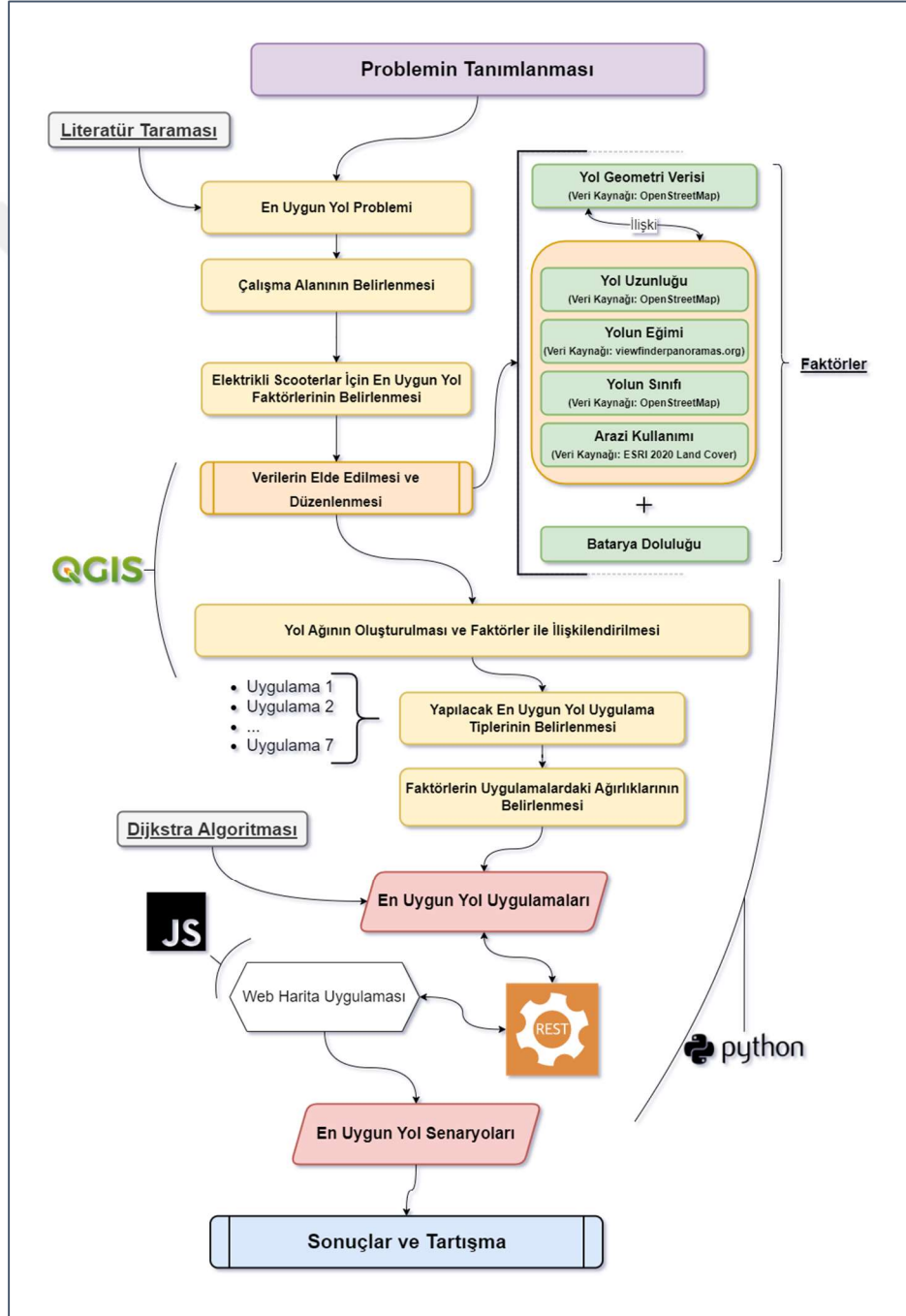
3. MATERYAL VE YÖNTEM

Tez kapsamı çerçevesinde öncelikle ilgili literatürler taranmış, yapılacak analizler ve uygulamalar için yöntem belirlenmiş, ardından ilgili veriler elde edilmiş ve bu veriler ile CBS uygulamalarının yapılabilmesi için hangi araçlardan ve yazılımlardan faydalanılacağı tespit edilmiştir. Bu bölümde, literatür taramaları, metodoloji, çalışma alanı, uygulama öncesi hazırlıklar ve tespitlere yer verilmiştir.

3.1. Metodoloji

Tez çalışması kapsamında yapılan literatür taramaları neticesinde Şekil 3.1’de görüldüğü gibi bir metodoloji planlaması yapılmıştır. Bu metodolojiye göre problemin tanımlanmasının ardından literatür taraması yapılarak ana problem ele alınmıştır. En uygun yol problemi tez çalışması içerisindeki ana problemdir. Bu problemin çözümü, elde edilecek mekansal bilgiler ve çevresel faktörlerin de kontrol edilebilmesi için sınırları belli bir çalışma alanı (İzmit) belirlenmiştir. Çalışma alanının belirlenmesinin ardından elektrikli scooterlar için en uygun yol faktörleri belirlenmiştir. Bu faktörler yol uzunluğu, yolun eğimi, yolun sınıfı, arazi kullanımı ve batarya doluluğudur. Faktörlerin seçiminde statik veri olmaları ve temel faktör olarak nitelendirilebilmeleri önem taşımaktadır. Bu faktörlerin birbirleriyle ilişkileri incelenmiş, hangi kaynaklardan ve nasıl elde edilecekleri planlanmıştır. Verilerin elde edilmesinin devamında python programlama dili ve QGIS CBS yazılımı yardımıyla gerekli düzenlemeler yapılarak yol ağı oluşturulmuştur. Bu yol ağının oluşturulmasının devamında yine aynı çalışma ortamlarında yol ağı elektrikli scooter faktörleri ile ilişkilendirilmiştir. Birden fazla faktör olmasının getirisi olarak, faktörlerin birbirleri ile yapabileceği farklı kombinasyonların analiz edilmesi için çeşitli uygulama tipleri belirlenmiştir. Bu uygulama tipleri içerisinde faktörlerin, en uygun yol algoritması içerisine farklı hesaplamalarla dahil edilerek ya da edilmeyerek sonuca etkileri incelenmiştir. Aynı zamanda bu uygulamalar içerisinde faktörlerin ağırlıkları belirlenmesi gerekmektedir. Uygulama tiplerinde faktörlerin ilgili uygulamanın sonucuna ne kadar etki ettiği önceden planlanmış ve en uygun yol uygulama senaryoları için gerekli ön şartlar tamamlanmıştır. Python programlama dili ve kütüphaneleri yardımıyla en uygun yol uygulamaları yapılmıştır. En uygun yol uygulamaları için farklı başlangıç ve hedef noktalarının bulunduğu senaryoların uygulanmış ve bu uygulamalarda Dijkstra Algoritması kullanılmıştır. Senaryoların

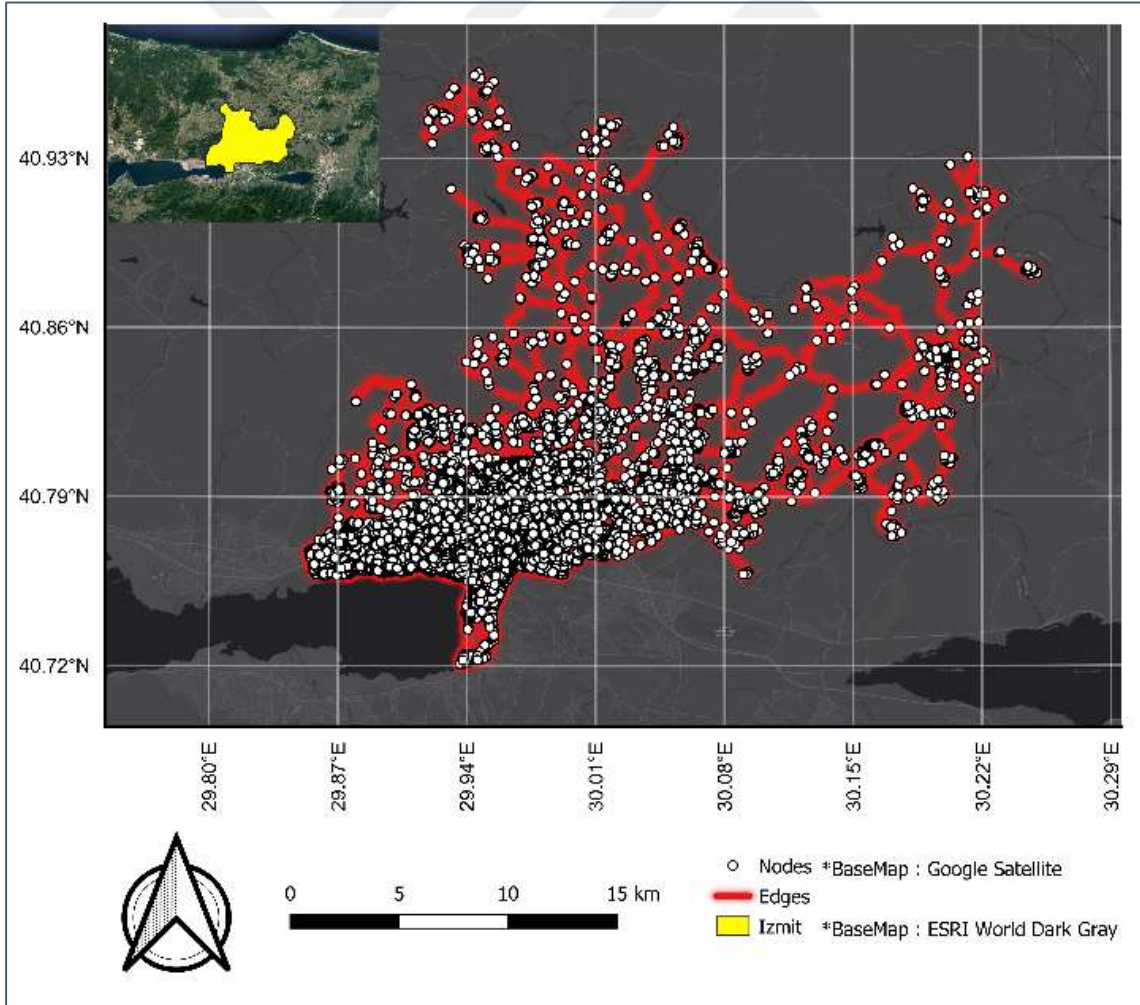
uygulanmasında, analiz sonuçlarının tablosal olarak ve harita üzerindeki görselleştirmelerinde JavaScript ve python programlama dillerinden faydalanılarak oluşturulan web harita uygulamasından yararlanılmıştır. Web arayüzü ve uygulama fonksiyonları arasındaki iletişim için Rest API kullanılmıştır. Son olarak analiz sonuçları değerlendirilmiş, yorumlanmış ve yeni tartışma konuları saptanmıştır. Şekil 3.1 üzerinden metodoloji incelenebilir.



Şekil 3.1. Metodoloji

3.2. Çalışma Alanı

Kocaeli İzmit ilçesi, Türkiye'nin Marmara Bölgesi'nde yer alan önemli bir sanayi merkezidir. Bölgedeki yoğun nüfus ve trafiğe ek olarak, İzmit'in eğimli bölgeleri, arazi kullanımı çeşitliliği, dar sokakları, yol ağlarının sadeliği ve merkezi konumu, elektrikli scooterlar için ve farklı tipte uygulamalar için ideal bir çalışma alanı sağlamaktadır. Ayrıca, İzmit'teki hızlı nüfus artışı ve toplu taşıma sistemlerinin bazı zamanlarda yetersiz kalması, elektrikli scooter kullanımının giderek artmasıyla sonuçlanabilir. Bu çalışma alanı, yerel yönetimler ve diğer ilgili kurumlar için faydalı olacak önemli bir araştırma konusu sağlamaktadır. İzmit bölgesinin merkezi koordinatları $40^{\circ} 45' 50''$ N (Kuzey), $29^{\circ} 56' 40''$ E (Doğu) olarak bilinmektedir. Şekil 3.2 üzerinden İzmit bölgesi için düğüm ve kenar bilgilerinin de bulunduğu çalışma alanı incelenebilir. Ayrıca bu şekil üzerinden bölgenin Google Satellite (Google uydu görüntüsü) üzerinden de konumu görülebilir.



Şekil 3.2. Çalışma alanı

3.3. Veri Kaynakları

Veri kaynakları, en kısa yol uygulamalarında doğrudan veya metodoloji içerisinde bahsi geçen 5 ayrı faktör için dolaylı yoldan kullanılmak için, öncelikle temin edilmeleri gerekmektedir. Hangi verilerin kaynak olarak kullanılabileceği konusunda (Plynnig, 2016) araştırması ve OSM'in ücretsiz olarak kullanıma sunduğu kaynaklar baz alınmıştır. Bu bilgiler ışığında Tablo 3.1 incelenerek veri kaynakları ve faktör ilişkileri incelenebilir. Batarya Doluluğu faktörü bir veri kaynağına sahip olmamasından ötürü Tablo 3.1'de yer almamıştır.

Tablo 3.1. Veri kaynakları

Veri	Format	İçerik	İlişkili Yol Ağı Faktörü	Kaynak
Yol Ağı Verisi	Vektör (Çizgi)	Yol geometrisi, yol uzunlukları ve sınıfları	Yol uzunluğu, yol sınıfı ve batarya doluluğu	OSM
Yükseklik (Rakım)	Raster	Bölgenin yükseklik değerleri	Yol eğimi	(URL-37)
Arazi Kullanımı	Vektör	Bölgedeki arazi kullanımı	Arazi Kullanımı	KBB

Yol ağı verisi, elektrikli scooterların kullanacakları yolların uzunluğuna, yolun sınıfına ve geometrisine dair bilgileri içerir. Bu bilgiler, elektrikli scooterların hızlı, güvenli ve konforlu bir şekilde seyahat edebilecekleri yolları belirlemeye yardımcı olmaktadır. Yol sınıfı verileri, elektrikli scooterların seyahat edeceği yollar için, trafiğe açık olan yollar mı yoksa ara sokaklar mı olduğu, yol türleri gibi dikkate alınması gereken durumları belirler.

Yükseklik verileri, elektrikli scooterların seyahat edecekleri yolların eğimlerinin ölçülmesine ve elektrikli scooterların yokuşlu yollarda daha az enerji harcamalarına olanak sağlayacak en uygun yolları belirlemeye yardımcı olmaktadır. Yükseklik verileri yol ağı içerisinde eğim verilerine dönüştürülerek kullanılmıştır.

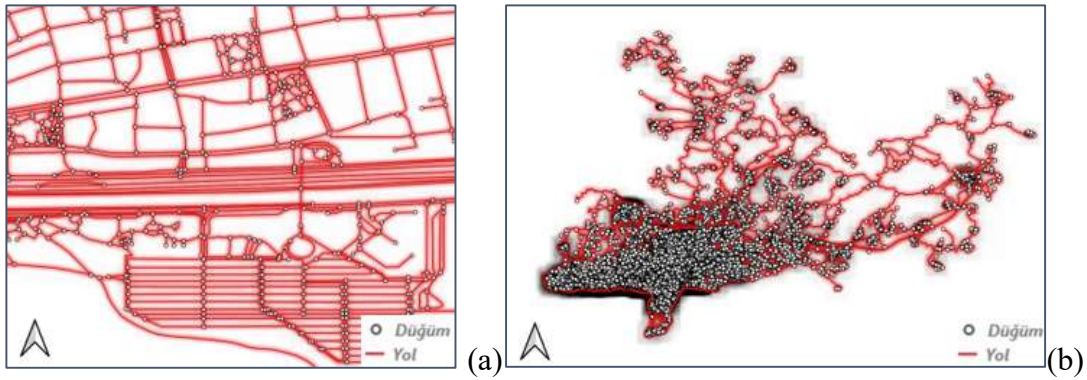
Arazi kullanımı verileri, elektrikli scooterların seyahat edecekleri alanların yüzey özelliklerini, insanlar tarafından nasıl kullanıldığını belirler ve elektrikli scooterların daha

güvenli ve verimli bir şekilde seyahat etmelerine olanak sağlar. Aynı zamanda kullanıcının isteğine bağlı olarak seyahat ve turizm gibi konfora dayalı uygun yol hesaplamalarında da önemli rol oynamaktadır.

3.3.1. Yol Ağı'nın Oluşturulması

En uygun yol problemi, iki nokta arasındaki en uygun yolun belirlenmesi ile ilgilidir. Bu problemi çözmek için, öncelikle bir yol ağı oluşturulması gerekmektedir. Yol ağı, yolların ve kavşakların birbirleriyle bağlantılarını gösteren bir grafik şeklindedir. Bu ağ, en uygun yol problemi çözülmek istenen bölgenin tamamını kapsamalıdır. En uygun yol probleminin kullanılacağı yol ağı oluşturulması için veri kaynağı olarak başlangıç ve bitiş düğümlerine sahip yolların olması ve bu yolların birbirleriyle topolojik ilişkilerinin bulunması gerekmektedir.

OSMnx kütüphanesi yardımıyla bölgesel (İzmit ilçesi) olarak sınırlandırılarak OSM tarafından sağlanan yol ağı verisi elde edilebilmektedir. Yol ağı verisinin elde edilmesi işleminin devamında CBS yazılımı yardımıyla veri görselleştirilmiş ve incelenmiştir. Şekil 3.2 üzerinden yol ağı verisi örnek bölge ile birlikte incelenebilir.



Şekil 3.2. Yol ağı verisi, a-örnek bölge, b-tüm yol ağı

Yine python kütüphaneleri ve CBS yazılımı aracılığıyla kenar verisi içerisindeki sütun bilgileri üzerinde manipülasyonlar yapılmıştır. Kenar veri katmanı çizgi verileri olmakla birlikte, düğüm katmanı nokta verileridir. Tablo 3.2 üzerinden de anlaşılabileceği üzere kenar veri katmanında başlangıç düğümü ve hedef düğüm adında düğümlere karşılık gelen iki sütun bulunmaktadır. Bu sütunlar aracılığıyla düğümlerin kenarlar aracılığıyla nasıl ilişkilendirildiği daha net anlaşılabilmektedir. Yol uzunlukları için “uzunluk” sütunu

ve yol kenarlarının temsil ettiği yolların sınıfları da “yol sınıfı” adlı sütun yardımıyla yol ağında kullanılmak üzere incelenebilmektedir. Tablo 3.2 içerisindeki örnek yol sınıfı bilgileri OSM tarafından etiketlenmiş orijinal adları (ingilizce) ile gösterilmiştir.

Tablo 3.2. Yol ağı kenar verisi tablosu (örnek sütun ve veriler)

Başlangıç Düğümü	Hedef Düğüm	OSM Id	Tek Yön	Ad	Yol Sınıfı	Uzunluk
29190470	29190476	[187508451, 187508524, 160626520, 187508443, 187508447]	1	Anadolu Otoyolu	motorway	498,124
443552589	641775269	83239149	0	Ömer Türkçakal Bulvarı	trunk	219,48
2164282807	2164282827	206422742	1	Zincirlikuyu Caddesi	primary	36,25
2264541873	2264541877	186835414	0	Anadolu Otoyolu	motorway	600,205
7920280690	9505958808	[187508432, 205802691, 4610213]	1	Anadolu Otoyolu	motorway	2263.018

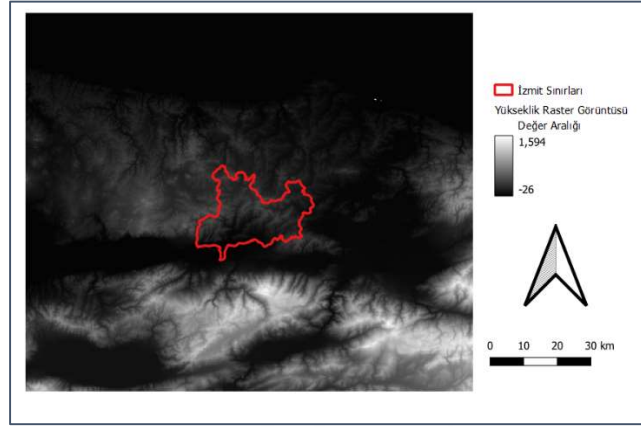
3.4. Yol Ağı ile Faktörlerin İlişkilendirilmesi

Belirlenen faktörler ile yol ağı arasında ilişkinin kurulması aşamasında her bir yola faktör değerleri atanacaktır. Bu değerlerin atanması işlemi için öncelikle faktörlerin yol verileriyle ilişkilendirilebilecek şekilde manipüle edilmesi ve şekillendirilmesi gerekmektedir. Faktörler, yol ağı ile ilişkilendirilmeleri konusunda farklı yöntemlerle topolojik ilişkiye dahil edilebilmektedir. Ancak her bir faktör ilişkilendirilmesi işlemi için ayrı ayrı inceleme yapılmalı ve ilişkilendirilme sonrasında değerlerin en kısa yol algoritması içerisinde kullanılmak üzere indeks değerlerine dönüştürülmesi gerekmektedir.

3.4.1. Yol Eğimi Faktörünün İlişkilendirilmesi

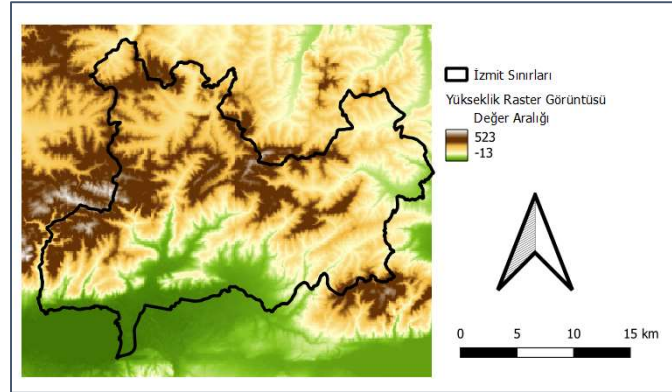
Yol eğimi faktörünün yol ağı ile ilişkilendirilmesi için öncelikle yükseklik verisinin elde edilmesi yükseklik değeri olarak yol ağı verisine dahil edilmesi gerekmektedir. Elde edilen yükseklik verisi Marmara bölgesinin bir kısmının bulunduğu bulunduğu 2 adet

raster verisinin birleşimidir. Şekil 3.3 üzerinden İzmit ilçesi sınırları ve Yükseklik verisinin (URL-37) kapladığı alan incelenebilir.



Şekil 3.3. Yükseklik verisi ve İzmit ilçe sınırları (kırmızı)

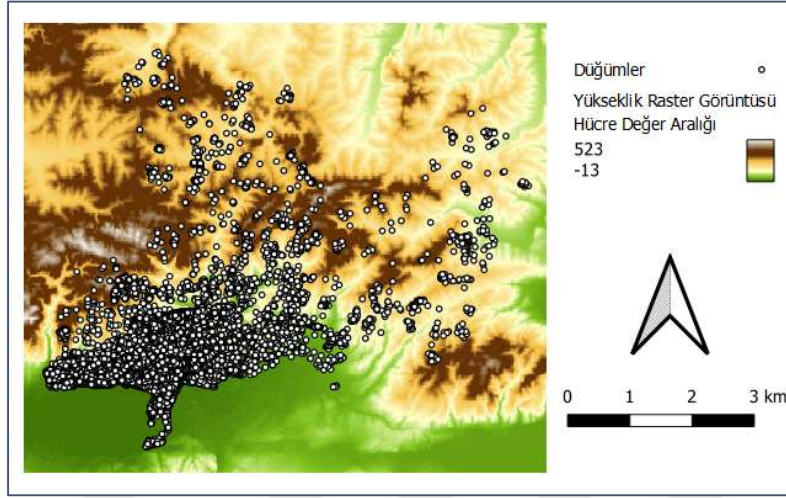
Bu işlem için elde edilmiş olan yükseklik raster verisinin İzmit ilçe sınırları ile kesişen alanlarının kırılması gerekmektedir. Bunun sebebi veri ilişkilendirme işlemlerinin daha hızlı yapılabilmesi, verideki gereksiz boyut büyüklüğünün ortadan kaldırılmasıdır. Bu işlem için QGIS yazılımı içerisinde bulunan Clip Raster by Extent aracı kullanılabilir. Şekil 3.4 üzerinden stilize edilmiş yükseklik verisi incelenebilir.



Şekil 3.4. İzmit İlçesi stilize edilmiş yükseklik verisi ve İzmit ilçe sınırı

Yol ağı verilerinin izmit ilçe sınırlarının dışarısında da olduğu ve yol ağının bütünlüğünün bozulmaması sebebi ile kırma işlemlerinde sınırlar için olduğundan daha büyük bir alan kullanılmıştır. Kırılan yükseklik raster verisi için maksimum 523 ve minimum 0 olacak şekilde bir semboloji çalışması yapılarak arazinin yapısı daha rahat incelenmiştir. Bir sonraki aşamada verinin yol ağı ile ilişkilendirme işlemlerine geçilmiştir. Bu işlem için

yol ağındaki düğüm noktalarına ihtiyaç duyulmaktadır. Şekil 3.5 üzerinden düğüm verileri yükseklik verisi ile birlikte incelenebilir.



Şekil 3.5. Düğüm verileri ile yükseklik raster verisi

QGIS yazılımı içerisinde bulunan Sample Raster Values aracı ile düğüm nokta verilerinin kesiştiği pikseller saptanarak, bu piksellerin değerleri düğüm verilerine aktarılabilmektedir. Yeni oluşan düğüm verisinin tablo yapısı Tablo 3.3 üzerinden incelenebilir.

Tablo 3.3. Yükseklik değerli düğüm verisi (örnek sütun ve veriler)

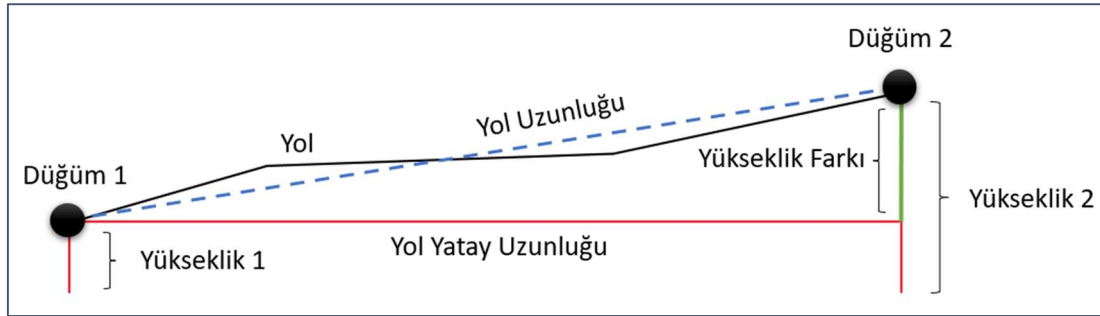
OSM Id	Y	X	Yükseklik
26553173	40,7837772	29,9310183	178
26553194	40,780746	29,924301	168
26553375	40,7805726	29,9780153	26
26553387	40,7832733	29,9754193	17
26553753	40,7657927	30,0239108	12

Elde edilen yükseklik verisinin yol kenar verileriyle ilişkilendirilmesi işlemi ile eğim faktörü oluşturulabilmektedir. Bu işlem için eğim hesabı kullanılabilir. Eğim hesabı iki yüksekliğin farkının, birbirleri ile arasındaki mesafeye oranıdır. Denklem (3.1) den anlaşılacağı üzere başlangıç ve bitiş düğümü belli bir yolun eğiminin ölçülebilmesi

için bu iki düğümün yükseklikleri ve yol kenarının uzunluk değeri yeterli olacaktır. İndeks işlemlerinde yüzde değeri olarak kullanılabilmesi için 100 ile çarpılması gerekmektedir;

$$Eğim (\%) = \frac{Yükseklik_2 - Yükseklik_1}{Yol\ uzunluğu} * 100 \quad (3.1)$$

Bu eşitlikteki parametrelerin ve eşitliklerin ve uygulamadaki görünüşünün daha iyi anlaşılabilmesi için Şekil 3.6 incelenebilir.



Şekil 3.6. Yol eğimi hesabı

Bu işlem için yol kenar verisi içerisinde bulunan düğüm bilgileri ile “length” sütun bilgisi kullanılarak eğim değerleri hesaplanmıştır. Eğim verisinin indeks hesaplamalarında mutlak değer içerisinde kullanılmasından ötürü yön bilgisi önemsiz olarak kabul edilmiştir. Bu sebeple düğümlerin bağlı oldukları yol kenarlarında başlangıç veya bitiş düğümü olup olmadıklarına dikkat edilmemiştir. Tablo 3.4 üzerinden eğim ve yükseklik değerli kenar verileri incelenebilir.

Tablo 3.4. Eğim değerli kenar verisi (örnek sütun ve veriler)

Başlangıç Düşümü	Hedef Düşüm	Yükseklik 1	Yükseklik 2	Uzunluk	Mutlak Eğim
2157861294	8015954084	9	9	63,45	0
2157861294	7681390201	11	9	145,09	1,381
2164282636	7577315341	11	6	104,78	4,773
2247610141	9428953439	29	17	63,73	18,843
2247610141	9397955487	16	17	81,07	1,237

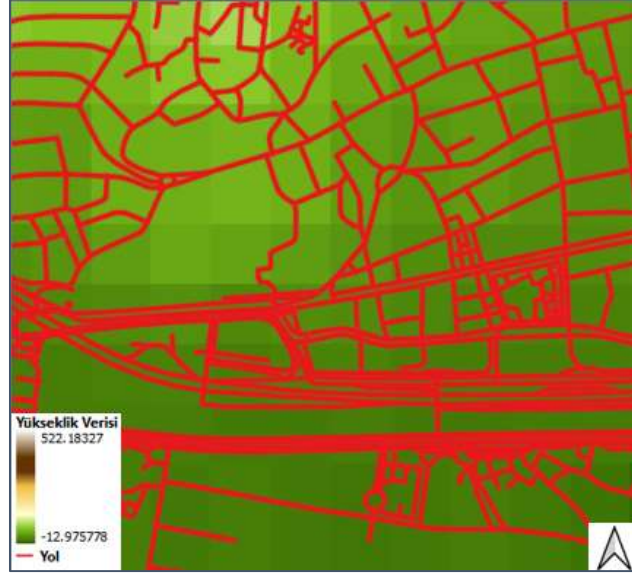
Yapılan işlemler sonucunda her yol kenarı için eğim değerleri hesaplanmıştır ancak yapılacak uygulama sonuçlarındaki doğruluğu etkileyecek bazı etkenler olduğu görülmüştür. Bunlar;

- Elde edilen yükseklik verisi bir dijital yükseklik modeli verisidir. Bölgedeki yapıların yükseklikleri yeri yer yüzeyinin şeklini (topoğrafya) yansıtmaktadır. Bu sebeple iki düğüm arasındaki yolun örneğin bir yaya köprüsü üzerinde olması durumunda, bu köprünün yüksekliği hesaplamalara dahil edilmemiştir. Elde edilen veri üzerinden bu hesaplamaların yapılamayacağı görülmüş ve uygulanan senaryolarda bu gibi durumların da göz önüne alınarak yorumların yapılması planlanmıştır.

37

değişiklikleri de yok sayılacaktır. Bu durum raster verisindeki çözünürlük etkenine benzese de bu konuda yolların uzunlukları da önem arz etmektedir. Yapılan incelemelerde yol uzunluklarının birden fazla pikseli aştığı bölgelerde yükseklik değişikliklerinin genellikle tek yönlü olduğu görülmüştür. Bunun anlamı, iki düğüm arasındaki yükseklik değişikliklerinin ya sürekli artan yönde ya da sürekli azalan yönde olduğudur. Bu sebeple bu etken üzerinden oluşabilecek hatalı eğim hesaplamalarının göz ardı edilebileceği düşünülmüştür.

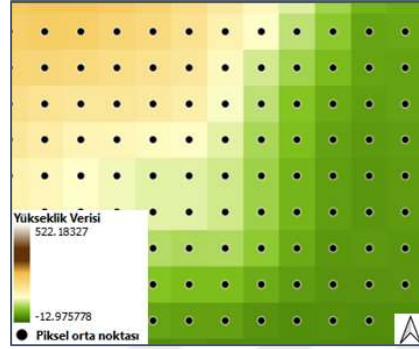
Yükseklik raster verisinin çözünürlüğü yaklaşık 100 metredir. Bunun anlamı yaklaşık 100 metrekairelik aynı piksel içerisine düşen iki düğüm arasındaki eğim farklarının yok sayılacağıdır. Bu durum anormal eğim değerlerini de ortaya çıkartmaktadır. Bir yolun başlangıç ve bitiş noktaları farklı piksellerde yer alıyor ise, yolun uzunluğunun 100 metreden aşağı olduğu durumlarda eğim değerleri dört tekerlekli normal bir taşıtın bile yukarı yönde çıkamayacağı eğim değerlerini oluşturmaktadır. Şekil 3.8 üzerinden 100 metrekairelik pikseller ve üzerlerinde bulunan yolların bulunduğu örnek bölge incelenebilir.



Şekil 3.8. Yükseklik raster görüntüsü ile yol verileri

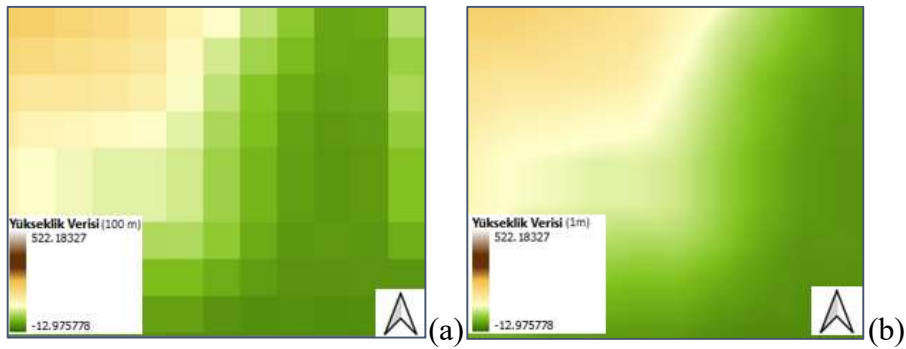
Bu durum için daha yüksek çözünürlükte yükseklik verisi elde edilmesi dışında farklı bir yöntem olarak raster görüntüsünün çözünürlüğünün yükseltilmesi düşünülmüştür. Bu yöntem için ise TIN enterpolasyonu kullanılacaktır. Bu enterpolasyon yöntemi, düzensiz aralıklı nokta verileriyle oluşturulmuş bir üçgen ağı kullanarak noktalar arasındaki

değerleri tahmin etmeye dayanır. Enterpolasyonun yapılabilmesi için öncelikle raster verisinin her bir pikselinin ortasına karşılık gelen bir nokta verileriyle bir vektör katmanı oluşturulmuştur. Şekil 3.9 üzerinden her 100 metrekairelik pikselin orta noktaları gözlemlenebilir.



Şekil 3.9. 100 Metrekarelik yükseklik piksel orta noktaları

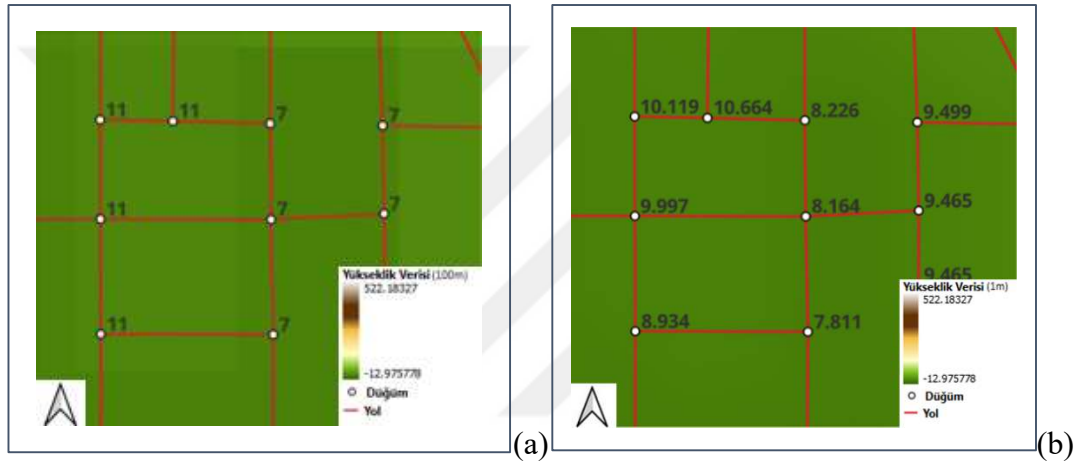
Oluşturulan nokta verileri arasındaki boşluklara tahmini değerler atanacaktır. Bu tahmini değerler ile oluşturulacak yeni yükseklik raster görüntüsünün piksellerinin kapladığı alanlar 1 metrekaire olarak tanımlanacaktır. Bu yöntem ile yükseklik verisi raster görüntüsü çözünürlüğü 100 metrekaireden 1 metrekaireye kadar yükseltilmiş olacaktır. Şekil 3.10 üzerinden 100 metrekairelik ve 1 metrekairelik yükseklik raster görüntüleri karşılaştırılabilir.



Şekil 3.10. Yükseklik raster görüntüleri, a-100 metrekaire, b-1 metrekaire

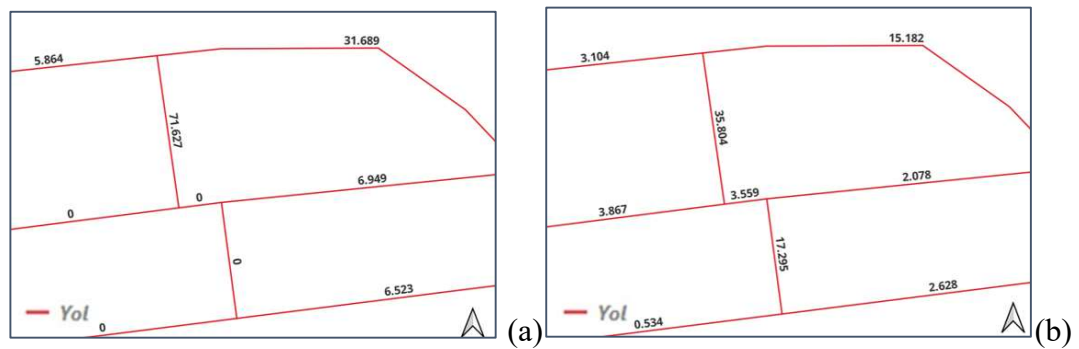
Bu yöntem her ne kadar veri değerleri olarak çözünürlüğü artırmış olsa da, elde edilen yeni yükseklik verisi yalnızca tahmini olarak hassasiyeti artırmaktadır. Başlangıçta herhangi 2 nokta arasındaki yükseklik değişimlerinin, aralarındaki uzaklık düşünüldüğünde gerçeği yansıtmayacak şekilde değerler aldığı gözlemlendi. Örneğin bir yolun başlangıç noktasının konumu bitişik iki pikselin birine, bitiş noktasının konumu da

öteki piksele denk geliyor ise bu yolun uzunluğunun 100 metreden az olduğu durumlarda anormal eğim değerleri oluşabiliyordu. Ancak yeni yükseklik verisinde bu iki piksel arasındaki her bir metreye diğer yükseklik noktaları ve değerleri ile yapılan üçgenlemeler sonucu tahmini yükseklik değerleri atanmıştır. Böylece örnekte bahsedilen iki noktanın yükseklik değerleri ve dolayısıyla eğim değerlerinin de bir önceki duruma göre daha doğru olduğu görülecektir. Şekil 3.11 üzerinden yolların başlangıç ve bitiş noktaları için iki raster farklı raster görüntüsünden nasıl yükseklik değerleri aldıkları incelenebilir. Eski yükseklik değerlerinin aynı piksellerde aynı değerleri aldığı, yeni yükseklik değerlerinin ise çözünürlüğün yüksek olması nedeniyle farklı değerler alabildiği görülmektedir.



Şekil 3.11. Yükseklik değerleri, a-eski, b-yeni

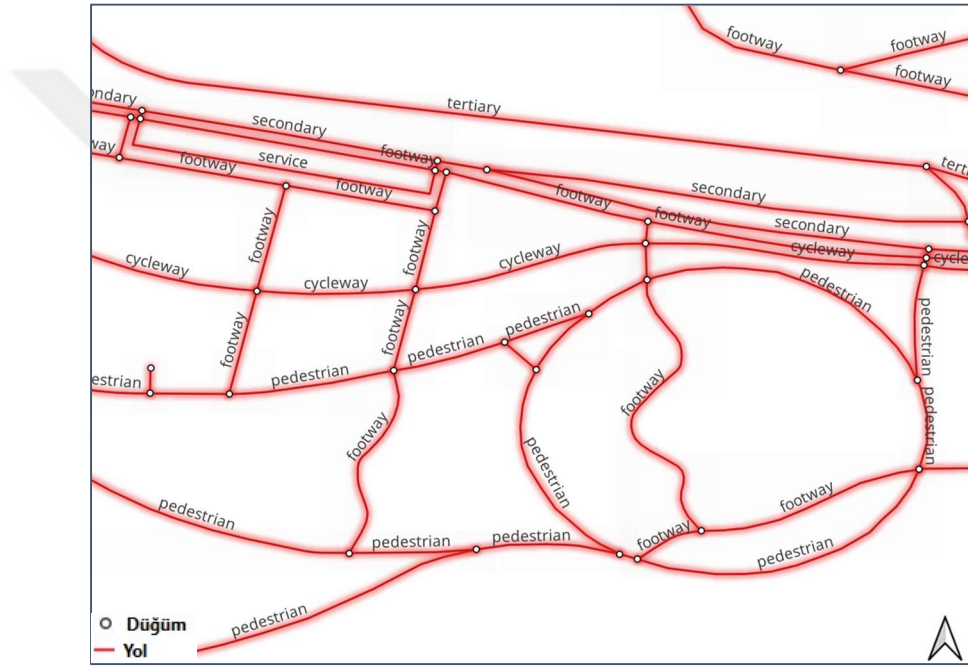
Hesaplanan eğim değerlerinin değişimi de Şekil 3.12 üzerinden incelenebilir. Yükseklik değerlerindeki değişimler sonrası yeni oluşan mutlak eğim değerlerinin daha düşük olduğu ve gerçeğe yakın değerler olabilecekleri görülmektedir. Hesaplanan bu yeni mutlak eğim değerleri uygulamalarda kullanılacaktır.



Şekil 3.12. Mutlak eğim değerleri, a-eski, b-yeni

3.4.2. Yol Sınıfı Faktörünün İlişkilendirilmesi

Yol sınıfı faktörü, OSMnx kütüphanesi aracılığıyla elde edilen OSM yol ağı içerisinde hazır olarak bulunmaktadır. Yol kenar verisi içerisinde “yol sınıfı” adlı sütun bilgileri yol sınıfı faktörü olarak kullanılacaktır. Bu bilgilerin her biri (URL-38) kaynağında da belirtildiği gibi bir etiket anlamı taşımaktadır. Bu sebeple yol sınıf bilgileri hesaplamalar içerisinde orijinal adları ile (ingilizce dilinde) kullanılmışlardır. Şekil 3.13 üzerinden bu yol sınıflarının yol verileri üzerindeki tanımlamaları örnek bölge üzerinden gözlemlenebilir.



Şekil 3.13. Örnek yol sınıfı verisi

Yol sınıfı verileri İngiliz Yol Sistemi'ne göre sınıflandırılmış verilerdir. Türkiye için aynı isimde birebir karşılıkları bulunmayabilir. Şehirler arası yollar, şehir içi yollar, yaya yolları, bisiklet yolları gibi araç ve yayaların kullandığı bütün yollar sınıflandırılıp haritada gösterilebilmektedir. (URL-38) kaynağı içerisinde “Yol Türleri” başlığı altında yol sınıflarının gruplandırılmalarına ve açıklamalarına detaylıca yer verilmiştir. En yaygın olarak kullanılan OSM highway etiketleri:

- motorway: Otoyollar için kullanılır.
- trunk: Ana yolları ifade eder.
- primary: Birinci derece yolları ifade eder.

- secondary: İkinci derece yolları ifade eder.
- tertiary: Üçüncü derece yolları ifade eder.
- residential: Yerleşim yollarını ifade eder.
- unclassified: Sınıflandırılmamış yolları ifade eder.
- service: Servis yollarını ifade eder.
- track: Arazi yollarını ifade eder.
- path: Yaya yollarını ifade eder.
- pedestrian: Sadece yaya trafiği için ayrılmış yolları ifade eder.
- cycleway: Bisiklet yollarını ifade eder.

Bu etiketler, yol özellikleri hakkında bilgi sağlar ve navigasyon, yolculuk planlama ve trafik analizi gibi amaçlar için kullanılabilir (URL-38).

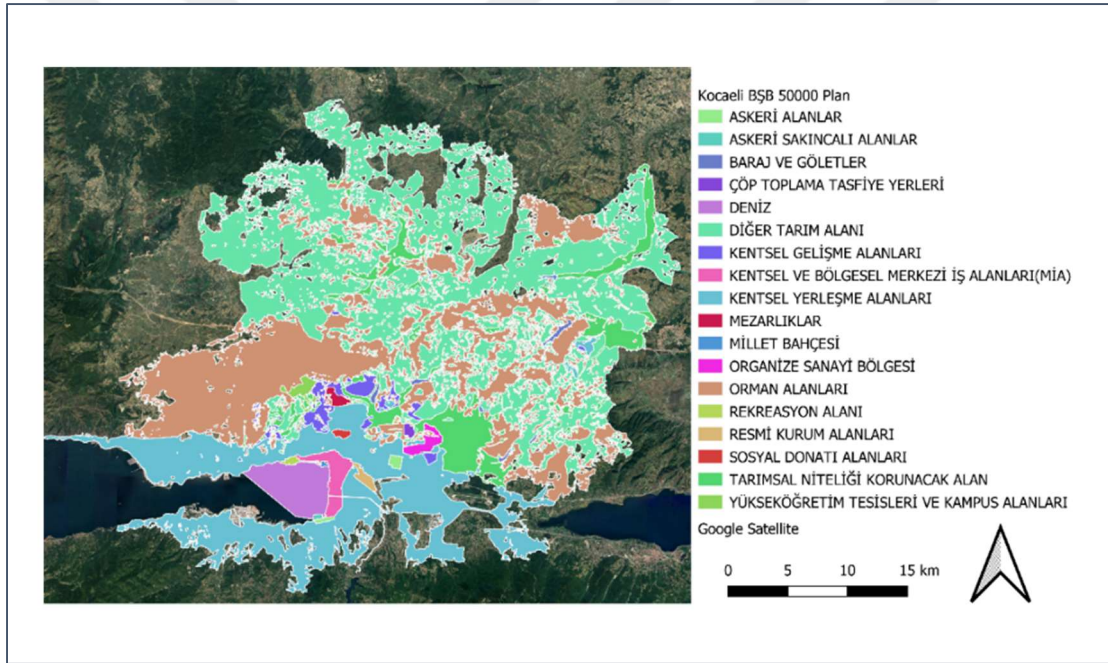
3.4.3. Arazi Kullanımı Faktörünün İlişkilendirilmesi

Arazi kullanımı faktörü, elektrikli scooter kullanıcılarının doğayla iç içe olan, eğlence ve seyahat amacıyla kullanılabilecek park, bahçe, sahil vb. bölgeleri öncelikli olarak gören bir faktör olarak kullanılacaktır. Bu faktörün ilişkilendirilmesi için öncelikle (URL-39) kaynağından elde edilen arazi örtüsü verisinin kullanılması düşünülmüştür. Veri tipi raster olan arazi örtüsü verisinin çözünürlüğü ve çeşitliliğinin uygulama için yeterli seviyede olamayacağı öngörülerek bu verinin kullanımından vazgeçilmiştir. Şekil 3.14'deki Google Satellite görüntüsü üzerinden İzmit bölgesinde birçok farklı arazi kullanımı olduğu anlaşılabılır.



Şekil 3.14. İzmit bölgesi Google Satellite Görüntüsü

Arazi kullanımı faktörü için bir başka yol olarak Kocaeli Büyükşehir Belediyesi veri tabanında bulunan 12.05.2023 tarihli güncel 50000 ölçekli plan verileri talep edilmesi uygun görülmüştür. Plan 50000 adlı vektör veri içeriği, bölgedeki yerleşim alanları, yeşil alanlar, rekreasyon alanları, park ve bahçeler, baraj ve göletler, yasak bölgeler, kampüs alanları gibi detaylı bilgilere sahiptir. Bu bilgilere ek olarak katman içerisine “Deniz” bilgisi poligon çizilerek eklenmiş ve elektrikli scooterlar için sahil şeritleri gibi önemli bir bilginin de ön hazırlığı yapılmıştır. Yalnızca yol ağı kenar ve düğüm verileriyle kesişen poligon verileri dikkate alınarak, gereksiz veriler kaldırılmıştır. Şekil 3.15 üzerinden 5000 ölçekli plan verisi ve lejant bilgileri ile İzmit bölgesi için ne gibi tanımlamalar olduğu incelenebilir.



Şekil 3.15. 50000 ölçekli plan verisi

5000 ölçekli plan verisi, yol verisi ile ilişkilendirilebilmesi için sadeleştirme amaçlı gruplanacak ve sayısal bir puan değerine karşılık gelecektir. Bu işlemler için (Plynnig, 2016) kaynağındaki gruplama çalışması örnek alınmıştır. Plan verisi için özel koşullar göz önüne alınarak arazi kullanım faktörü grupları ile plan bilgileri eşleştirilerek sadeleştirme yapılmıştır. Şekil 3.16 üzerinden plan verilerinin hangi arazi kullanım gruplarına dahil edildikleri incelenebilir. Endüstriyel alanlar ve resmi kurumlar gibi aynı puana sahip olacak bazı arazi kullanım grupları tek bir grup olarak tanımlanmıştır.

Arazi Kullanım Grupları	5000 Ölçekli Plan Verisi Bilgileri
Su	- Barajlar ve Göletler - Deniz
Yeşil Alan	- Millet Bahçesi - Orman Alanları - Rekreasyon Alanı
Tarım Alanı	- Diğer Tarım Alanları - Tarımsal Niteliği Korunacak Alan
Düşük Yoğunluklu Yerleşim Alanı	- Kentsel Gelişme Alanları - Sosyal Donatı Alanları
Orta Yoğunluklu Yerleşim Alanı	- Kentsel ve Bölgesel Merkezi İş Alanları - Yükseköğretim Tesisleri ve Kampüs Alanları
Yüksek Yoğunluklu Yerleşim Alanı	- Kentsel ve Bölgesel Merkezi İş Alanları - Kentsel Yerleşme Alanları
Endüstriyel Alanlar, Resmi Kurumlar ve Mezarlıklar	- Çöp Toplama ve Tasfiye Yerleri - Mezarlıklar - Organize Sanayi Bölgesi - Resmi Kurum Alanları
Askeri Alanlar	- Askeri Alanlar - Askeri Sakıncalı Alanlar

Şekil 3.16. Plan verileri ve arazi kullanım grupları eşleştirmesi

3.5. Ağırlık İndeks Değerlerinin Hesaplanması

Faktörleri doğrudan ağırlık olarak kullanmak yerine indeks değerlerine dönüştürmek, maliyet analizlerinde oldukça önemli bir adımdır. Bu işlem, farklı faktörlerin maliyet üzerindeki etkisini daha doğru bir şekilde yansıtabilir ve farklı faktörlerin önem derecesini belirleyebilir. Örneğin, yolun uzunluğu ve eğimi gibi faktörlerin maliyet üzerindeki etkisi farklı olabilir ve bu faktörlerin doğrudan ağırlık olarak maliyet hesabına katılması, gerçek maliyeti yansıtmayabilir.

İndeks değerleri, farklı faktörlerin maliyete olan katkısını daha hassas bir şekilde ölçebilir ve farklı faktörlerin önem derecesini belirleyebilir. Bu sayede, maliyet analizlerinde farklı faktörlerin katkısı doğru bir şekilde hesaplanabilir ve karşılaştırılabilir. Ayrıca, indeks değerleri sayesinde, farklı faktörlerin maliyet üzerindeki etkisini ayrı ayrı değerlendirerek, maliyeti en aza indirmek için hangi faktörlere öncelik verileceğine karar verilebilmektedir.

En kısa yol algoritmalarında faktörlerin hesaplamalara dahil edilebilmesi belli için sayısal

değerlere karşılık gelmeleri gerekmektedir. Bu nedenle faktörler kendi içlerinde gruplanabilir ve uygulama içerisinde kullanılmak üzere sayısal indeks değerlerine dönüştürülebilmektedir. Bu indeks değerleri en kısa yol algoritmaları içerisinde ağırlık indeks değeri olarak tanımlanabilir. Ağırlık indeks değerleri, faktörün en uygun yol hesabı içerisinde hangi özelliğinin uygulama sonucuna ne kadar etki edebileceğini belirler. Bu sebeple birden fazla faktör kullanılan en uygun yol uygulamalarında indeks değerlerinin diğer faktörler üzerindeki etkilerini de dikkate almak gerekecektir.

Tüm bu nedenlerle, faktörleri indeks değerlerine dönüştürmek, maliyet analizlerinin doğruluğunu ve hassasiyetini artırır. Bu sayede, daha doğru kararlar alınarak, maliyeti en aza indirmek mümkün olur.

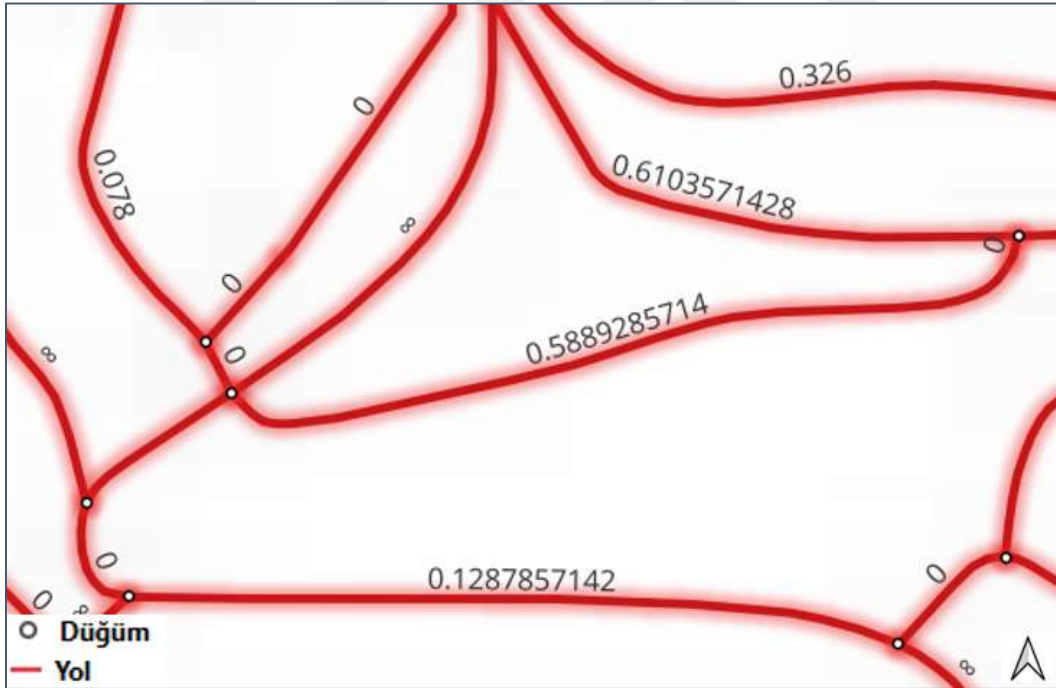
3.5.1. Yol Eğimi Ağırlık İndeks Değerlerinin Hesaplanması

Yol eğimi, elektrikli scooterların batarya doluluğuna, yokuş çıkabilme kapasitelerine ve tehlikeli sürüşlere doğrudan etkileyen bir faktördür. Yukarı yönde bir yokuşu tırmanabilmek için elektrikli scooterın enerjisi, kapasite özellikleri ve sürtünme kuvvetinin elektrikli scooter üzerindeki etkisi gibi birçok etken dikkate alınabilir. Bu nedenle eğim indeksi belirlenirken aracın tırmanabileceği maksimum bir eğim değeri olmak zorundadır. Maksimum eğim değeri için internet üzerinde yapılan incelemelerde ortalama bir elektrikli scooterın %10-20 arası bir eğimi tırmanabileceği görülmüştür. Bu değerler dış etkenlere göre yükselebilir veya alçalabilir (URL-40; URL-41; URL-42).

İndeks hesaplaması için yapılan literatür ve internet kaynağı incelemeleri sonrasında (3.2) Denklemi oluşturulmuştur. Bu eşitliğe göre eğer bir yolun mutlak eğim değeri 1'den az ise bu yolun eğim indeksi 0 olarak kabul edilecektir. Bunun anlamı bu yolların düz kabul edileceğidir. Eğer bir yolun mutlak eğimi 1 ila 15 arasında ise, eğim indeks değeri bir normalizasyon işlemi ile hesaplanacaktır. Buradan anlaşılacağı üzere elektrikli scooterların tırmanabileceği maksimum eğim değeri bu araştırma için 15 olarak kabul edilmiştir. Eğer bir yolun mutlak eğim değeri 15'in üzerinde ise bu yolun eğim indeksi sonsuz kabul edilecek, en kısa yol algoritması içerisinde tercih edilmeyen yol grubunda olacaktır. Mutlak eğim arttıkça indeks değeri, indeks değeri arttıkça maliyet değeri artacaktır (URL-44);

$$NE_i = \begin{cases} 0, & |E_i| < 1 \\ \frac{|E_i|-1}{15-1}, & 1 \leq |E_i| \leq 15 \\ \infty, & |E_i| > 15 \end{cases} \quad (3.2)$$

İzmit ilçesi için yapılan incelemelerde eğim değeri sonsuz olan yolların çoğunlukta olduğu bazı bölgeler görülmüştür. Bu bölgelerde yapılacak çalışmalarda, uygulamanın sonuç verebilmesi için indeks değeri sonsuz olan yolların da sayısal bir değere sahip olması düşünülmüştür. Bu sayısal değer uygulamada çok uçuk sayılabilecek bir değer olmasından ötürü, en kısa yol algoritması çalışması esnasında eğer hedef noktası ve başlangıç noktası arasındaki bütün rotalarda en az 1 tane sonsuz indeks değerine sahip yol varsa, en az sonsuz indeks değerine sahip olan rotaların tercih edilmesi planlanmıştır. Bu sayede uygulama sonucunda maliyet analizi sonsuz değer olarak gözükse bile en uygun olarak tercih edilebilecek yol üzerinde incelemeler yapılabilmesine olanak sağlanacaktır. Yol eğim indeks değerleri hesaplanmış örnek yol ağı görüntüsü Şekil 3.17 üzerinden incelenebilir.



Şekil 3.17. Örnek yol eğim indeksi

3.5.2. Yol Sınıfı Ağırlık İndeks Değerlerinin Hesaplanması

Yol sınıfı faktörünü indeks değerine dönüştürmek, farklı yol tiplerinin maliyete ekilerinin daha doğru bir şekilde karşılaştırılmasını sağlar. Yol sınıfı indeks değeri belirlenirken,

söz konusu uygulama için birbirleriyle aynı tipte kabul edilebilecek yol sınıfları aynı gruba dahil edilebilir. Bu sayede uygulamaların maliyet hesaplarında birden çok yol sınıfı karmaşasından kurtulmuş olunur.

Günümüzde İzmit ilçesinde elektrikli scooterlar için belirli özel kurallar bulunmadığı göz önüne alınırsa, yol sınıfı ağırlık indeks değerleri hesabında bu araştırma için 4 farklı yol sınıfı grubunun dikkate alınabileceği görülmüştür. Bunlar:

- Bisiklet yolları, yaya yolları vb. olarak elektrikli scooterlar için konfor bakımından en güvenli yollar,
- Şehir içi ara yollar gibi dört tekerli taşıtların da kullanılabildiği ancak yüksek hızlarda kullanılmadıkları ve sakin olarak nitelendirilebilecek yollar,
- Elektrikli scooterlar için tehlikeli olarak sayılabilecek, yüksek hızlı araç kullanımının yüksek olduğu ana yollar,
- Elektrikli scooter kullanımına kapalı otobanlar ve elektrikli scooterların kullanılamayacağı merdivenlerdir.

Bu grupların belirlenmesindeki ana düşünce, elektrikli scooterlar kullanıcılarının tehlike bakımından daha konforlu yol sınıflarını tercih etmeleridir. Ancak öte yandan bu araçların hiç kullanılamayacağı yol sınıflarının da göz önünde bulundurulması gerekmektedir. Bu nedenle yol sınıfı indeks değeri belirlenirken yolun elektrikli scooterlar üzerindeki tehlike durumuna dikkat edilerek, tehlike durumu arttıkça indeks değerinin de artması planlanmıştır. Elektrikli scooter kullanıcılarının kullanamayacağı yol sınıflarının indeks değerleri sonsuz olarak hesaplanacaktır. Ancak buradaki sonsuzluk ifadesi, tıpkı yol eğimi indeks değerleri hesaplanırken olduğu gibi uçuk bir sayısal değere karşılık gelmektedir. Bu sayede eğer en uygun yol hesaplamalarında elektrikli scooterların kullanamayacağı yolların da kullanıldığı durumlar varsa, bu yolların ve dolayısıyla yol ağının da incelenmesi sağlanacaktır. Yol sınıfı ağırlık indeks değerleri hesabı için (3.3) Denklemi oluşturulmuştur;

$$NS_i = \begin{cases} 1, & \text{Bisiklet Yolları, yaya yolları vb.} \\ 1.25, & \text{Şehir içi ara yollar vb.} \\ 1.5, & \text{Ana yollar, birinci yollar vb.} \\ \infty, & \text{Otoban ve merdivenler} \end{cases} \quad (3.3)$$

3.5.3. Arazi Kullanımı Ağırlık İndeks Değerlerinin Hesaplanması

Arazi kullanımı faktörü için oluşturulmuş olan arazi kullanımı gruplarının indeks değerlerine dönüştürülmesi için önce kendi içlerinde puanlanarak öncelik sınıflandırılması yapılmıştır. Bu puanlama sayesinde arazi kullanım bilgileri sayısal değerlere dönüştürülecektir. Normalizasyon yardımı ile arazi kullanımı indeksi 0 ve 1 değer aralığında kontrol edilebilir bir faktör olacaktır.

Arazi kullanımı gruplarının puanlanması işleminde 0-100 aralığında sayısal değerler kullanılacaktır. Burada işlemler için (Plynnig, 2016) çalışması incelenmiştir. (Plynnig, 2016) çalışmasından farklı olarak, arazi kullanım faktörünün kullanıldığı uygulamalarda maliyeti arttıracak olan gruplar için yüksek puanların verilmesi uygun görülmüştür. Maliyeti düşüren gruplar için de düşük puanlar verilecektir. Böylece yapılacak normalizasyon sonrası arazi kullanımı ağırlık indeks değerleri yükseldikçe maliyet de artacaktır. Arazi kullanımı sınıflandırmaları için yapılan gruplar ve puanlandırmalar Tablo 3.5 üzerinden detaylıca incelenebilir.

Tablo 3.5. Arazi kullanımı sınıflandırma

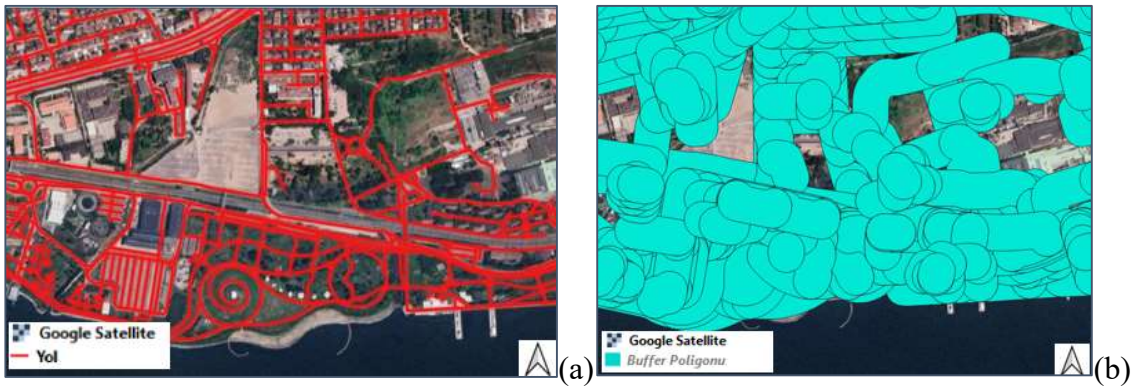
Arazi Kullanım Grubu	Puan
Su	0
Yeşil Alan	5
Tarım Alanı	20
Düşük Yoğunluklu Yerleşim Alanı	25
Orta Yoğunluklu Yerleşim Alanı	40
Yüksek Yoğunluklu Yerleşim Alanı	80
Endüstriyel Alanlar, Resmi Kurumlar ve Mezarlıklar	90
Askeri Alanlar	100

Yapılan puanlamaya göre kabalılığın, gürültünün ve seyahat amaçlı kullanılmayan bölgelerin maliyetleri yüksektir. Kabalılığın olmadığı ve yeşilin yoğun olduğu bölgelerin

maliyetleri düşüktür. Özellikle sahil şeridi gibi su bölgelerinin en çok tercih edilen alan olması belirlenmiştir. Verilen puanların en uygun yol uygulamalarında kullanılabilmesi için indeks değerlerine dönüştürülmesi gerekmektedir. İndeks değerlerinin daha rahat kontrol edilebilmesi ve uygulamalarda mantıklı sonuçlara varılabilmesi için min-max normalizasyonu yapılarak 0 ve 1 arasında bir değer aralığı uygun görülmüştür. Bu hesap için (3.4) Denklemi kullanılarak arazi kullanımı buffer verileri indeks değerleri hesaplanmıştır;

$$NAB_i = \frac{A_i - \min(AK)}{\max(AK) - \min(AK)} \quad (3.4)$$

Ağırlık indeks değerlerinin oluşturulması sonrasında yol ağı kenarlarının bu değerlerle ilişkilendirilmesi gerekmektedir. Örneğin bir yolun aynı anda birden fazla bölgeden geçiyor olması, deniz kenarında bulunduğu anlaşılabilmesi gibi çeşitli durumlar mevcuttur. İlk olarak yolların bazı bölgelerin yakınında bulunması durumu dikkate alınması nedeni ile yol ağı kenar verilerine buffer (tampon) işlemi uygulanacaktır. Buffer işlemi bir analitik işlem olarak kullanılır. Buffer, belirli bir nokta, çizgi veya alan etrafında belirli bir mesafe içeren bir tampon bölgesi oluşturmayı ifade eder. Bu işlem, bir konumun etrafındaki alanı veya yakınındaki diğer öznelikleri incelemek ve analiz etmek için kullanılır. Yol ağı kenar verileri çizgi verilerinden oluşmaktadır. Şekil 18 üzerinden buffer işleminin örnek bölge üzerinde ne gibi sonuçlar verdiği karşılaştırmalı olarak incelenebilir.



Şekil 3.18. Buffer işlemi örnek bölgesi, a-örnek yollar, b-buffer poligonları

Yol kenar ağlarına 50 metrelik buffer işleminin yapılmasının ardından buffer işlemi yapılmış yol ağı kenar verileri ile arazi kullanımı ağırlık indeks değerlerinin

ilişkilendirilmesi yapılacaktır. Bu adımdaki asıl amaç, bir yol buffer verisi ile kesişen arazi kullanımı poligon verilerinin tespit edilerek, kesişen poligonların arazi kullanımı ağırlık indeks değerlerinin ortalamasının alınması, daha sonrasında bu ortalama değerlerin yollar ile ilişkilendirilmesidir.



Şekil 3.19. Ağırlık indeks değerlerinin ilişkilendirildiği örnek bir buffer

Şekil 3.19 üzerinden yapılan işlemler için oluşturulan örnek incelenebilir. Bu örneğe göre, seçili bir yol için 50 metrelik oluşturulan buffer poligonu ile kesişen 3 adet arazi kullanımı poligonu verisi tespit edilmiştir. Bu verilerin grup, puan, indeks ve oranları da etiket olarak gösterilmiştir. Oran değerleri, arazi kullanımı poligonlarının seçili yol buffer poligonu içerisinde kapladığı alanı temsil etmektedir. Seçili yolun arazi kullanımı ağırlık indeksinin hesaplanması için ise (3.5) Denklemindeki ağırlıklı ortalama yöntemi kullanılmıştır;

$$NA_i = \frac{\sum_{i=1}^n (NAB_i * Geometri Alan_i)}{\sum_{i=1}^n Geometri Alan_i} \quad (3.5)$$

3.6. Uygulama Tiplerinin Belirlenmesi

Elektrikli scooterlar için uygun yol maliyet hesaplamaları yapılırken, farklı faktörlerin maliyet hesaplarına dahil edildiklerinde ve dahil edilmediklerinde ne gibi sonuçlarla karşılaşıldığı incelenecektir. Bu sebeple bir başlangıç noktası ve bir hedef noktası için

hesaplanacak en uygun yol uygulamaları birden fazla olacaktır. Her bir uygulamada farklı faktörlerin birbirleriyle oluşturdukları kombinasyonlar üzerinden kenar maliyetleri hesaplanarak en uygun yollar belirlenecektir.

Yapılacak uygulamalarda maliyet indeks değerlerinin hesaplamaları sabit tutulacaktır. Bunun anlamı, maliyet indeks değerlerinin hesaplanmasındaki değişimlerin incelenmeyeceğidir. Bunun yerine faktörlerin dahil oldukları uygulama sonuçları incelenerek yorumlanacaktır. Bu yorumlamalar üzerinden indeks değer hesaplamalarındaki değişimin önemine dikkat çekilebilir. Bu bilgiler ışığında belirlenen uygulama tipleri Tablo 3.6 üzerinden incelenebilir. Tablodan de anlaşılabilceği üzere yol uzunluğu değeri tüm uygulamalar için ana faktördür. Diğer faktörler yol uzunluğu değeriyle ilişkili olarak maliyete etki edecektir.

Tablo 3.6. Uygulamalar ve kullanılan faktörler

Uygulama	Yolun Uzunluğu Değeri (U_i)	Yolun Eğimi İndeks Değeri (NE_i)	Yolun Sınıfı İndeks Değeri (NS_i)	Arazi Kullanımı İndeks Değeri (NA_i)
Uygulama 1	✓	X	X	X
Uygulama 2	✓	✓	X	X
Uygulama 3	✓	✓	✓	X
Uygulama 4	✓	✓	✓	✓
Uygulama 5	✓	X	✓	X
Uygulama 6	✓	X	X	✓
Uygulama 7	✓	X	✓	✓

3.6.1. Batarya Doluluğu Faktörünün Dikkate Alınması

Elektrikli scooterların batarya doluluğu, özellikle uzun mesafeli ve eğimli yollarda seyahat ederken dikkat edilen önemli bir faktördür. Yolun uzunluğuna ve -varsa- yolun eğimine bağlı olarak, yolun uzunluğu ve -varsa- yolun eğim indeks değeri arttıkça batarya doluluğu azalacaktır. Önceden belirlenen maksimum eşik geçildiği takdirde en uygun olarak belirlenen yolun elektrikli scooterın bataryasının karşılamayacağı ve hedefe

varılamayacağı görülecektir.

Batarya doluluğu faktörü, bataryanın kalitesine, şarj seviyesine, ortamın sıcaklığı gibi çeşitli birçok faktöre bağlı olarak değişebilir. Yapılacak uygulamalarda bu gibi parametreler kullanılmayacağı için maksimum batarya doluluğu sabit bir değer olarak kabul edilecektir. Bu değer uygulamalarda opsiyonel olarak farklı yol uzunluklarına denk gelebilir. Elektrikli scooterın modeli gibi bilinmeyen durumlar olması nedeniyle web uygulaması içerisinde uygulama öncesi başlangıçta batarya doluluğu ve 100% dolu bir batarya doluluğunun karşılık geldiği katedilebilecek yol uzunluğu değiştirilebilir değerlerdir. Web uygulamasında tam dolu batarya standart olarak 20 kilometre uzunluklu bir yol ile eşdeğer olarak kabul edilecektir. Yani eğer bir elektrikli scooterın bataryası 100% dolu ise, elektrikli scooter aracı maksimum 20 kilometrelik bir yolu kat edebilecektir. Uygulamalarda iki ayrı batarya doluluğu hesabı yapılacaktır. Birincisi, batarya doluluğu hesabında yalnızca yol uzunluğunun dikkate alındığı yan değer. İkincisi ise batarya doluluğu katsayısı eğim-yol ilişkisine dahil edilerek, eğimin batarya doluluğu üzerindeki etkisi göz ardı edilmeyen ana değer. Uygulamalar içerisinde eğer bir maliyet hesabı içerisinde toplam yol uzunluğu ve -varsa- yolun eğimi faktörlerinin etkisiyle 10 kilometrelik bir yol kat edilmiş ise, bu eşiğin geçildiği düğüm işaretlenecektir. Tıpkı yol sınıfı indeksi ve -varsa- yol eğimi indeksinde olduğu gibi araştırmanın incelenebilmesi ve yorumlanabilmesi amacıyla, en uygun yol algoritması çalışmaya devam ederek hedef noktasına varacaktır. Ancak sonuç maliyet değerlerine batarya doluluğu eşiğinin geçildiği not edilecektir. Kabul değerleri ve eşik değerlerin daha iyi anlaşılabilmesi için (3.6) ve (3.7) Denklemleri incelenebilir;

$$B_{100\%} = 20 \text{ km} \quad (3.6)$$

$$B_{50\%} = 10 \text{ km} \quad (3.7)$$

Batarya doluluğunun tükenmesi konusunda yapılacak hesaplamalarda Uygulama 1 ve Uygulama 2'deki maliyet hesabı eşitlikleri dikkate alınabilir. Bu eşitliklerin nasıl oluşturulduğu ve maliyet hesabının nasıl yapıldığından Uygulama 2 başlığında bahsedilmiştir. Bu başlıklar incelendiğinde, yol eğimi faktörünün dikkate alınmadığı batarya doluluğu hesabı için (3.8) Denklemi ile algoritma içerisinde başlangıçtan itibaren n. kenar seçimi sonrası kalan batarya doluluğu hesaplanabilir. Batarya dolulukları

kilometre karşılıkları ile tutulmasından dolayı dört işleme dahil edilebilmektedir. (3.8) Denkleminden de anlaşılacağı üzere n. kenar seçimi sonrası kalan batarya doluluğu, başlangıçtaki batarya doluluğu ile o zamanki seçilen yolların uzunlukları toplamı farkına eşittir;

$$\Delta B_n = B_0 - \sum_0^n U_n \quad (3.8)$$

Yol uzunluğunun yanında yol eğiminin de dikkate alındığı batarya doluluğu hesabı için küçük bir ekleme gerekmektedir. Yolun eğim indeksinin de kullanıldığı bu hesaplamada, eğim indeksi yolun uzunluğu ile çarpılarak yine yolun uzunluğu ile toplanır ve batarya doluluğuna etki eder. Bunun anlamı ise, yol ağındaki herhangi bir x yol kenar eğimi 1 ise o yolun bataryaya doluluğunu azaltacak yöndeki etkisi 2 katına çıkacaktır. Bu yolun eğimi 0.5 ise batarya doluluğunu 1.5 kat daha fazla azaltacaktır. Bu hesap (3.9) Denklemi ile yapılacaktır. Bu hesaplamada yolun mutlak eğiminin eşik değeri geçmesi durumunda batarya doluluğu da sıfırlanmış olarak gözükecektir. Bu durum elektrikli scooterın yetenekleri açısından geçemeyeceği kesin bir yolu geçmesi anlamına gelmektedir. Ancak eğim faktörü eğer ilgili uygulamada dikkate alınmamışsa bu tip yolların tercih edilmesi olasıdır. Bu nedenle eğim değerinin eşiği geçtiği durumlarda eğim indeksi 1 olarak kabul edilecektir;

$$\Delta B_n = B_0 - \sum_0^n (U_n + U_n * NE_n) \quad (3.9)$$

3.6.2. Uygulama 1

Yalnızca yol uzunluğunun dikkate alındığı Uygulama 1 hesabında, en uygun yol ile en kısa yol aynı anlamı ifade etmektedir. Hesaplamalarda yalnızca yolun uzunluğu değeri kullanılmasından ötürü herhangi bir indeks değeri ile ilişki oluşmamaktadır. Bu uygulama, başlangıç noktasından hedef noktasına giden en kısa mesafeli yolun bulunması gerektiği durumlarda kullanılan en temel uygulamadır. (3.10) Denklemi ile yalnızca yol uzunluğu kullanılarak bulunan maliyet değeri hesabı incelenebilir;

$$M_1 = U_i \quad (3.10)$$

3.6.3. Uygulama 2

Yol uzunluğunun yanında yolun eğiminin de dikkate alındığı durumlarda Uygulama 2

kullanılacaktır. Uygulama 2 maliyet hesabında kullanılacak olan (3.11) Denklemine göre, eğer bir yolun eğim indeks değeri 1 ise, algoritma içerisinde o yolun kullanılması aynı mesafenin iki kez kat edileceği anlamına gelmektedir. Bu da elektrikli scooter kullanıcısı için yolun uygunluk bakımından tercih edilmesini zorlaştırmaktadır. Mutlak eğim değeri ile maliyet değeri doğru orantılıdır. Bu uygulama elektrikli scooter kullanıcısının engebeli yolları tercih etmek istemediği durumlarda istenilen en uygun yolu vermektedir;

$$M_2 = U_i + U_i * NE_i \quad (3.11)$$

3.6.4. Uygulama 3

Uygulama 1 ve Uygulama 2'ye ek olarak yolun güvenli ve sağlıklı kullanılmasına da önem veren Uygulama 3 içerisinde yolun sınıfı faktörü de dikkate alınmaktadır. Bu uygulamada hem engebeli yolların tercih edilmediği, hem de elektrikli scooter taşıtına daha çok hitap eden yolların tercih edildiği sonuçlar ortaya çıkmaktadır. Yolun sınıfı indeksi, maliyet hesaplamalarına tıpkı yolun eğiminde olduğu gibi yolun uzunluğuna bir katsayı çarpanı olarak dahil olmaktadır. İlgili yol kenarında yolun uzunluğu arttıkça o yolun sınıfının etkisi de aynı oranda artmaktadır. Bu uygulama, yapılan 7 uygulama içerisinde günümüzde en çok kullanıma ihtiyaç duyulan uygulama denilebilir. Uygulama 3 içerisindeki maliyet değeri hesabı için (3.12) Denklemi incelenebilir;

$$M_3 = U_i * NE_i + U_i * NS_i \quad (3.12)$$

3.6.5. Uygulama 4

Yol ağında bulunan tüm faktörlerin dahil edildiği Uygulama 4 içerisinde, ilk üç uygulamaya ek olarak arazi kullanımı da dikkate alınmıştır. Bu faktörün dahil edilmesi ile birlikte elektrikli scooter kullanıcısının doğa gezintisi talebine de yanıt verilebilmektedir. Bu uygulamaya göre bir kullanıcı için en uygun yol hesabında toplam mesafe, yollardaki eğim farklılıkları, yolların güvenliği, konforu ve doğayla iç içe olmasını dikkate alınmaktadır. Uygulama 4 içerisindeki maliyet değeri hesabı için (3.13) Denklemi incelenebilir;

$$M_4 = U_i * NE_i + U_i * NS_i + U_i * NA_i \quad (3.13)$$

3.6.6. Uygulama 5

Uygulama 5'te yolun sınıfı faktörü özellikle dikkate alınmaktadır. Elektrikli scooter kullanıcısı için eğer bir yol güvenli ve elektrikli scooter aracına uygun ise, o yol daha öncelikli yoldur. Bu uygulama elektrikli scooter kullanıcılarının yol güvenliğini ve konforu esas aldığı durumlar için elverişli sonuçlar vermektedir. Uygulama 5 içerisindeki maliyet değeri hesabı için (3.14) Denklemi incelenebilir;

$$M_5 = U_i * NS_i \quad (3.14)$$

3.6.7. Uygulama 6

Uygulama 6'da yolların bulundukları arazi kullanımı dikkate alınmaktadır. Bu uygulamada kullanıcının doğal arazi kullanımı bölgelerde gezinti yapabilmesi sağlanmış ve yolun eğimi, güvenliği gibi faktörler dikkate alınmamıştır. Uygulama 6 içerisindeki maliyet değeri hesabı (3.15) Denklemi ile incelenebilir;

$$M_6 = U_i * NA_i \quad (3.15)$$

3.6.8. Uygulama 7

Uygulama 7 maliyet değeri hesabında yolun eğimi faktörünün dikkate alınmamaktadır. Daha çok yol güvenliğine ve doğal çevreye önem veren bir uygulama olmakla birlikte, elektrikli scooter kullanıcıları için şehir içi keyifli sürüş sağlayan yolları tercih etmektedir. Uygulama 7 içerisindeki maliyet değeri hesabı için (3.16) Denklemi incelenebilir;

$$M_6 = U_i * NS_i + U_i * NA_i \quad (3.16)$$

3.7. Web Tabanlı Harita Uygulaması

Belirlenen uygulamaların sağlıklı sonuçlar verip vermediğinin incelenebilmesi ve sürekli olarak senaryo uygulamalarında kullanılabilmesi için bir web uygulaması geliştirilmiştir. Bu uygulamanın işlevi oldukça basittir. Kullanıcının belirlediği bir başlangıç ve bir hedef noktası seçimi sonrasında, uygula butonuna tıklaması ile bu başlangıç ve bitiş noktasının kullanıldığı 7 uygulama için en uygun yol sonuçlarını hem haritada görsel olarak

vermekte hem de tablosal ve grafiksel olarak kullanıcıya sunmaktadır.

Uygulamanın veri kaynağı için öncelikle QGIS ve python kütüphaneleri yardımıyla faktörleri dahil edilmiş şekilde oluşturulan İzmit ilçesi yol ağı hazırlanmıştır. Daha sonra python programlama dili ve kütüphaneleri (Pandas, Geopandas, NetworkX, OSMnx) yardımıyla ana ve yardımcı fonksiyonlar oluşturulmuştur. Ana fonksiyonlar en uygun yol uygulaması ve uygulama tiplerini yönetmektedir. Yardımcı fonksiyonlar ise ana fonksiyonların içerisindeki küçük eklemeleri yönetmektedir.

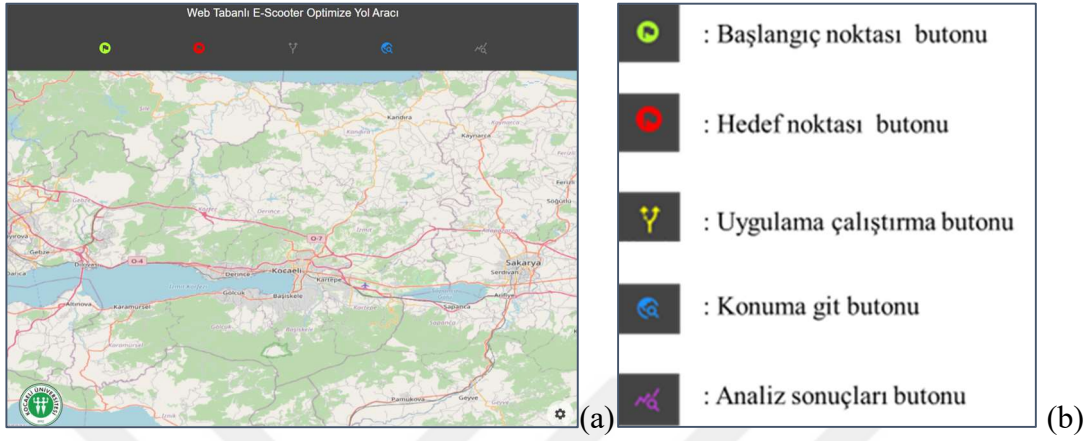
Web arayüzünün en kısa yol uygulamaları için gerekli olan isteklerine cevap verilebilmesi için arka uçta bir python kütüphanesi olan flask ile lokal olarak bir sunucu çalıştırılmıştır. Bu sunucu içerisinde çeşitli Rest API'ler oluşturulacaktır. API'lerin oluşturulma amacı web arayüzünden gönderilecek talepleri yönetmek ve bu talepler karşılığında istenilen sonuçları web arayüzüne cevap olarak dönmektir. Web uygulaması içerisinde uygula işlemi ile art arda çalışan, GET metodlu 2 servis oluşturulmuştur. Bunlar,

- Yakındaki Düğümü Getir Servisi: Verilen bir konuma yol ağındaki en yakın düğümü yakalayarak istemci (web arayüzü) tarafına cevap gönderen GET metodu kullanarak çalışan servis,
- En Uygun Yolları Getir Servisi: Parametre olarak başlangıç ve hedef düğümlerinin id değerlerini alarak çalışan, GET metodu kullanan bir servistir. Bu servis arka uçta uygulamanın gerektirdiği ana fonksiyonları kullanarak istemci tarafına tüm en uygun yol analizlerini ve istatistiksel değerleri cevap olarak gönderir.

3.7.1. Uygulama Arayüzü

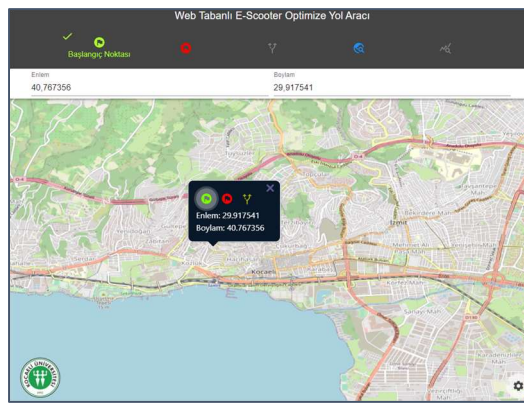
Uygulama arayüzü, elektrikli scooter kullanıcılarının 7 farklı uygulama için en uygun yol sonuçlarını hem haritada görsel olarak sunan hem de tablosal değerler ve grafiksel gösterimlerle detaylı analiz yapılabilmesine olanak sağlayan bir JavaScript-OpenLayers harita uygulamasıdır. Uygulama içerisinde kullanıcı için farklı yetenekler barındırılmakla birlikte basit ve açıklayıcı bir araç seti hazırlanmıştır. Farklı cihazlarda (mobil telefonlar, tabletler, dizüstü bilgisayarlar) ve ekran boyutlarında uyumlu bir şekilde çalışabilmekte ve kullanıcı deneyimini optimize edebilmektedir. Uygulama, kullanıcıların farklı

cihazlarda aynı içeriği sorunsuz bir şekilde görüntülerini ve etkileşime girmelerini sağlamaktadır. Şekil 3.20 üzerinden uygulamanın ana ekranı ve temel butonlar incelenebilir.



Şekil 3.20. Uygulama arayüzü, a-ana ekran,b-temel butonlar

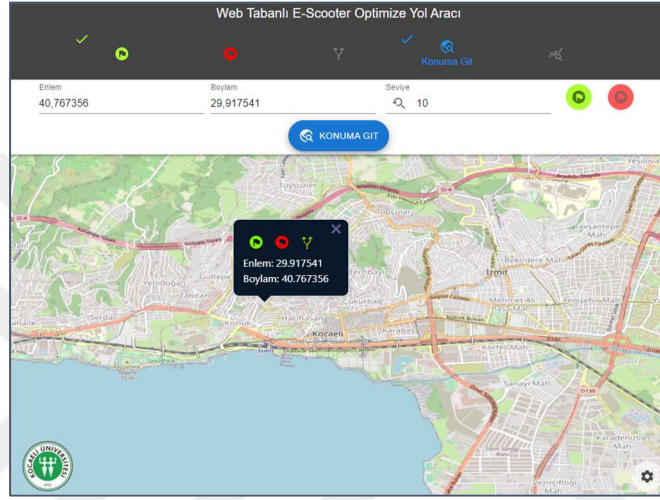
Uygulamada başlangıç ve hedef noktaları için harita üzerinden tıklama yöntemi veya enlem-boylam bilgisi girilerek yapılabilmektedir. Harita üzerinden tıklama işlemi sonrası ortaya çıkan pop-up adındaki kutucuk üzerinden başlangıç ve hedef noktalarını temsil eden yeşil ve kırmızı bayraklı butonlardan biri seçilerek, harita üzerinde tıklanan noktanın enlem-boylam değerleri seçili olan başlangıç veya hedef noktası bilgisi olarak kabul edilmektedir. Şekil 3.21 üzerinden ve hedef noktaları için tanımlamaların nasıl yapıldığı incelenebilir.



Şekil 3.21. Başlangıç-hedef noktası bilgisi

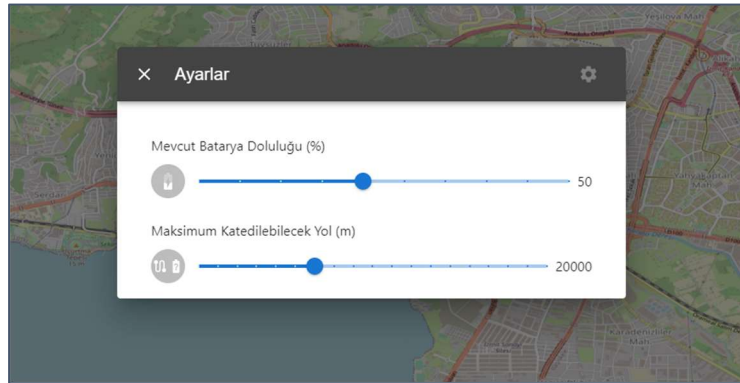
Uygulama içerisinde bulunan “Konuma Git” aracı ile kullanıcı tarafından girilen enlem-boylam noktalarına veya başlangıç-hedef noktalarına yaklaştırma yapılabilmektedir.

Kullanıcının isteğine bağılı olarak yakınlařtırma seviyesi deęiřtirilebilmektedir. Bu araçtaki Bařlangıç Noktası ve Hedef Noktası butonları eęer bilgileri mevcutsa kullanılabilir. Örneęin hedef noktası seçili deęil ise, “Konuma Git” aracındaki “Hedef Noktası” butonu kullanılmaz durumda olacaktır. řekil 3.22 üzerinden uygulamadaki “Konuma git” aracı gözlemlenebilir. Bu araç ile uygulama arayüzünde bařlangıç ve hedef noktasından ayrı olarak belirtilen konuma yakınlařtırma yeteneęi mevcuttur.



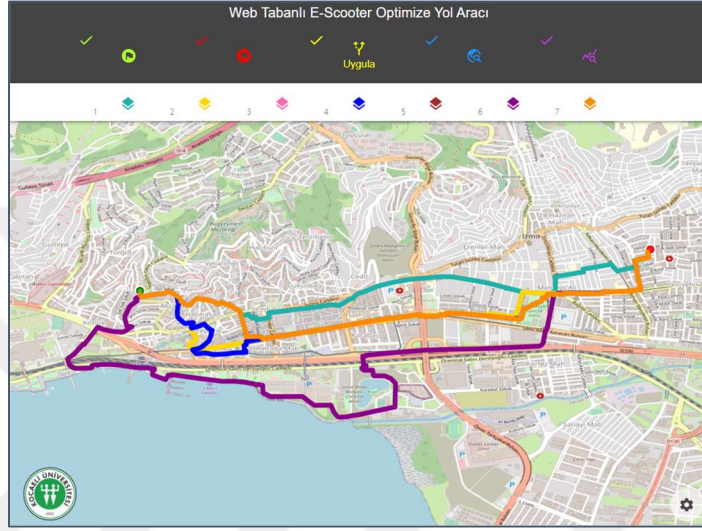
řekil 3.22. Konuma git

Uygulamayı kullanan kullanıcı, en uygun yolları getiren uygulamayı çalıřtırmadan önce elektrikli scooter aracı için mevcut batarya doluluęu ve maksimum kat edilebilecek yolu tanımlayabilmektedir. Uygulamada mevcut batarya doluluęu standart olarak bařlangıçta 50% ve maksimum kat edilebilecek yol bařlangıçta 20000 metre varsayılan olarak tanımlanmıřtır.



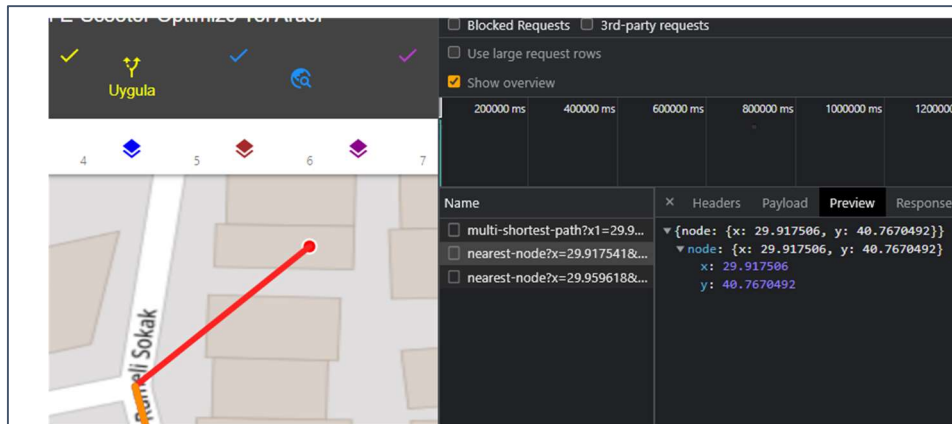
řekil 3.23. Elektrikli scooter ayarları

Uygulamanın ana fonksiyonunu barındıran “Uygula” aracı ile seçili olan başlangıç ve hedef noktaları arasındaki en uygun 7 yol bulunabilmektedir. Bu yollar hem uygulama bazlı numaralandırılarak hem de farklı renklerde katman olarak haritada gösterilmektedir. En uygun yolların haritada gösterilmesinin devamında kullanıcının isteğine bağlı olarak katman görünürlükleri kapatılıp açılabilir. Şekil 3.24 üzerinden 7 farklı uygulama için bulunan en uygun yollar ve bu yolların karşılık geldiği renkler incelenebilir.



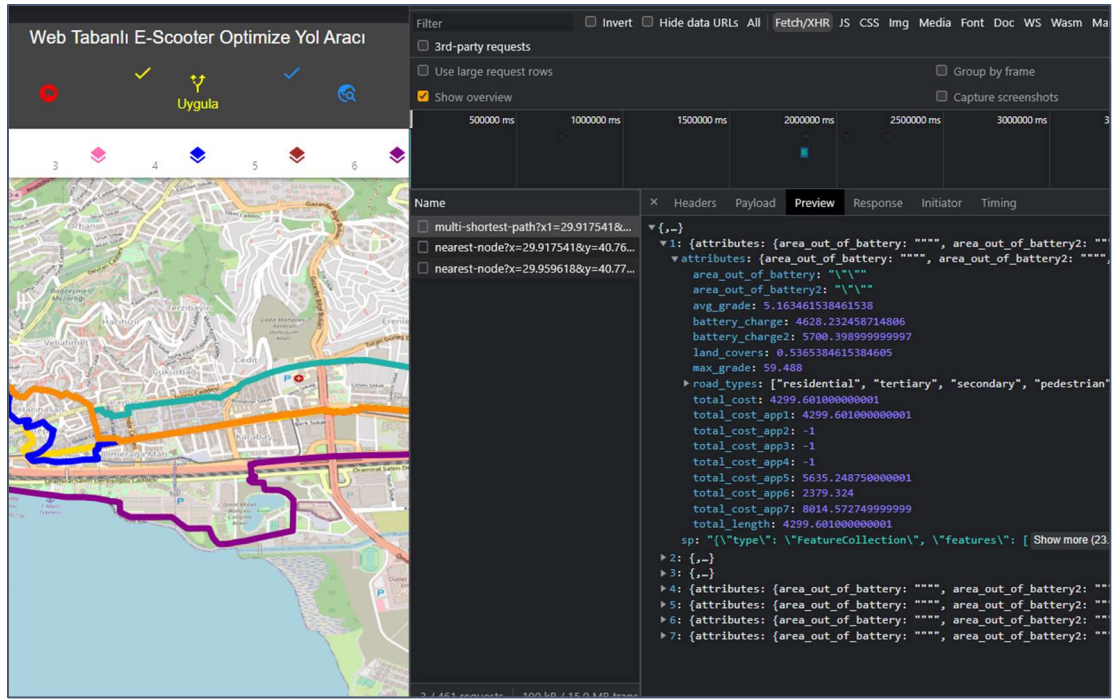
Şekil 3.24. En uygun yolların bulunması

Bu işlem için aynı zamanda backend servisleri kullanılmaktadır. İlk olarak başlangıç ve hedef noktaları için ayrı ayrı “Yakındaki Düğümü Getir” servisi çalışmaktadır. Bu servis sonrası gelen bilgi ile başlangıç ve hedef noktaları en yakındaki düğüme bağlanmaktadır. Bu değerler GET metodu ile yapılan sorgu sonrasında JSON veri formatında gelmektedir. Şekil 3.25 üzerinden yakındaki düğüm bilgilerini dönen API servisi incelenebilir.



Şekil 3.25. Yakındaki düğümü getir servisi sonucu

Yakındaki düğümlerin belirlenmesinin yanında en uygun yolları bulan “En Uygun Yolları Getir” servisi de çalışmaktadır. Bu servis için elektrikli scooter ayarları, başlangıç noktası konumu ve hedef noktası konumu parametre olarak verilerek analiz ve harita görseli için gerekli tüm veriler JSON formatında verilmektedir. Harita çizimi için gerekli olan bilgi ise metin veri tipine dönüştürülen GeoJSON formatında servis tarafından dönülmektedir. Şekil 3.26 üzerinden en uygun yolları getiren API servisinin sonucu incelenebilir.



Şekil 3.26. En uygun yolları getir servisi sonucu

“En Uygun Yolları Getir” servisi ile gelen bilgiler “Sonuçları Göster” aracı yardımıyla tablo gösterimi ve karşılaştırma grafiğinin bulunduğu “Sonuç Analizi” bölümü incelenebilmektedir. Tablo gösteriminde satırlar isteğe bağlı olarak harita gösterimindeki uygulama renkleriyle boyanarak daha rahat analiz edilebilir. Ayrıca karşılaştırma grafiği yardımı ile seçili bir bilgi farklı uygulamalar için karşılaştırılabilir. Sonuç Analizi bölümünde incelenebilen bilgiler sırasıyla “Uyg. Adı (Uygulama Adı)”, “Karşılaşılan Ortalama Eğim (%)”, “Kalan Batarya Doluluğu (m)”, “Kalan Batarya Doluluğu 2 (m)”, “Ort. Ar. Ör. İnd. (Ortalama Arazi Örtüsü İndeksi)”, “Ort. Yol Sınıfı İnd. (Ortalama Yol Sınıfı İndeksi)”, “Karş. Maks. Eğim (%) (Karşılaşılan Maksimum Eğim)”, “Uygulamanın Maliyeti”, “App (Uygulama) 1 Maliyeti”, “App 2 Maliyeti”, “App 3 Maliyeti”, “App 4 Maliyeti”, “App 5 Maliyeti”, “App 6 Maliyeti”, “App 7 Maliyeti” ve

“Toplam Uzunluk” bilgileridir. Şekil 3.27 üzerinden sonuç değerlerin uygulama arayüzünde grafiksel karşılaştırmaları ve tablosal değerleri incelenebilir. Grafik üzerinden uygulama sonuçlarının tüm değerleri karşılaştırılabilmektedir.



Şekil 3.27. Sonuç analizi bilgileri

4. UYGULAMA

Yapılacak uygulamalar için birbirinden farklı çeşitli sonuçların elde edileceği ön görülen 3 bölgede en uygun yol problemi uygulamalarının yapılması planlanmıştır. Bu uygulamalar 3 farklı senaryo içerisinde yapılacaktır. Senaryolar için web tabanlı harita uygulaması kullanılacak, veri görselleştirme çalışmalarında OSM altlık haritası üzerinde uygulamalar farklı renklerle temsil edilecektir. Tablo 4.1 üzerinden uygulamaların harita üzerindeki temsili renkleri incelenebilir.

Tablo 4.1. Uygulama temsili renkleri

Uygulama	Renk (Hex Kodu ile)
Uygulama 1	#20B2AA
Uygulama 2	#FFD700
Uygulama 3	#FF69B4
Uygulama 4	#0000FF
Uygulama 5	#A52A2A
Uygulama 6	#8B008B
Uygulama 7	#FF8C00

4.1. Senaryoların Belirlenmesi

Senaryolar belirlenirken farklı faktörlerin en yüksek verimde etkili olabileceği düşünülen bölgeler seçilmiştir. Bu bakımdan örneğin arazi kullanımı faktörünün dikkate alındığı uygulamalar ile dikkate alınmadığı uygulamaların farklarının görülebilmesi için sahil şeridi ve park bahçe gibi yolların tercih edilmesi olası olan başlangıç ve hedef noktaları seçilmiştir. Eğim faktörünün ve batarya durumu faktörlerinin dikkate alınacağı senaryoda ise eğim değerlerinin farklı oranlarda değişebileceği ve bu değerlerin değişmeyeceği yolların bulunduğu bölgeler de başlangıç ve hedef noktaları seçilmiştir. Senaryolarda bu gibi farklı tarzlarda seçim yapılmasının nedeni, en uygun yol uygulamalarının aynı senaryo içerisinde farklı sonuçlar verebildiğini ve bu sonuçların analizlerinin belirgin farklar ile yorumlanabilmesini sağlamaktır.

4.2. Senaryo 1

Uygulanacak Senaryoda 1 içerisinde 7 farklı uygulamalar arasında özellikle arazi kullanımı faktörünün ve yol sınıfı faktörünün uygulamalardaki en uygun yol seçimine nasıl etki ettiği konusunda inceleme yapılması planlanmıştır. Bu uygulama için başlangıç ve hedef noktaları arasında sahil, park bahçe, şehir merkezi gibi bölgelerin olması içerisinde arazi faktörünün bulunduğu uygulamalarda dikkate alınacak etkenler arasındadır. Aynı zamanda bu iki nokta arasında bisiklet yolları, şehir içi ara yollar gibi farklı sınıflarla karşılaşılan bölgeler de mevcuttur. Şekil 4.1 üzerinden senaryo 1 başlığı uygulamaları görülebilir. Farklı uygulamalar Tablo 4.1’de de belirtildiği üzere lejantlı olarak farklı renklerle temsil edilmektedir.



Şekil 4.1. Senaryo 1 başlığı uygulamaları harita gösterimi

Şekil 4.1 üzerinden de anlaşılacağı üzere özellikle arazi kullanımının dikkate alındığı Uygulama 6’da en uygun yol tercihleri için sahil şeritlerinin kullanıldığı görülebilir. Aynı zamanda yol sınıfının dikkate alındığı Uygulama 3,5 ve 7 için en uygun yol tercihlerinde yaya, bisiklet yolu gibi yolların yol sınıfı faktörünün öncelikli olarak dikkate aldığı yol sınıfları ile karşılaşıldığı kanısına varılabilir. Yalnız yol uzunluğu ve yol eğiminin dikkate alındığı Uygulama 1 ve 2 yol tercihlerinin ise herhangi bir arazi kullanımı ve yol sınıfı gibi faktörleri dikkate almaması nedeni ile hem şehir içi yolları hem de elektrikli scooter için daha az güvenli olan yol sınıflarının tercih edildiği görülebilmektedir. Aynı zamanda Şekil 4.2 üzerinden hem yeşil alanın hem de denizin iç içe bulunduğu bir bölgedeki yol fotoğrafı incelenerek daha doğru yorumlar yapılabilir.



Şekil 4.2. Senaryo 1 Google Street View örneği

Uygulanan senaryoda batarya doluluğu 100% ve 100%'lük bir batarya doluluğunda maksimum katedilebilecek yol uzunluğu değeri 5000 metre olarak tanımlanmıştır. Bu tanımlamalar sonrasında hiçbir uygulamada başlangıç noktasından hedef noktasına gidişte batarya doluluğunun yetmeyeceği bir durumla karşılaşılmamıştır. Uygulamaların genel analiz değerleri Tablo 4.2 üzerinden daha kolay bir şekilde incelenebilir.

Tablo 4.2. Senaryo 1 genel bilgiler

Uygulama Adı	Toplam Uzunluk	Ortalama Yol Sınıfı İndeksi	Ortalama Eğim İndeksi	Ortalama Arazi Örtüsü İndeksi	Ortalama Eğim (%)	Maks. Eğim (%)	Uygulama Maliyeti
1	3804,433	∞	0,252	0,367	4,138	14,696	3804,433
2	3804,433	∞	0,252	0,367	4,138	14,696	4512,791
3	4218,084	1,227	0,234	0,341	3,852	11,008	5809,571
4	4218,084	1,227	0,234	0,341	3,852	11,008	7038,688
5	4318,461	1,132	∞	0,472	5,116	23,306	4968,602
6	5148,208	∞	∞	0,209	4,25	25,544	924,376
7	4218,084	1,227	0,234	0,341	3,852	11,008	6287,658

Tablo 4.2 üzerinden edinilen bilgiye göre uygulamaların amaçları ve sonuç maliyetlerinin birbirleriyle uyum içinde olduğu söyleyebilir. Uygulama 1 ve 2 için toplam yol uzunluğu değerinin diğer tüm uygulamalardan daha az değer olması, Uygulama 1'in en kısa yol

uygulanması olması nedeniyle olması beklenen bir durumdur. Aynı zamanda eğim değerlerinin de en kısa yol üzerinde az olması nedeniyle Uygulama 2 için de aynı yollar tercih edilmiştir. Karşılaşılan ortalama ve maksimum eğimler dikkate alındığında Ortalama Arazi kullanımı indeksinin özellikle Uygulama 6’da en düşük değer olduğu dikkat çekmektedir. Aynı zamanda yol sınıfının dikkate alındığı Uygulama 3,4 ve 7’de ortalama yol sınıfı indeksinin aynı değerleri aldığı görülmektedir. Yol sınıfının dikkate alınmadığı bazı uygulamalarda (1,2 ve 6) ortalama yol sınıfı indeksinin sonsuz değer olduğu görülmektedir.

Uygulamaların birbirleriyle olan ilişkileri Tablo 4.3 üzerinden daha detaylı bir şekilde incelenebilmektedir. Bu Tabloya göre örneğin “Uygulama 1 için seçilen en uygun yol tercihlerinde, Uygulama 3 için olan faktörler dikkate alınsaydı ne olurdu?” gibi sorular cevaplanabilmektedir. Bu analiz değerleri tüm uygulamalar ve olasılıklar için geçerlidir. Bu Tabloda farklı uygulamaların normalde kullanılmadıkları yollarda kullanılmaları durumlarında, sonsuz maliyet değerlerine ulaşabildikleri görülebilmektedir. Örneğin karşılaşılan maksimum eğimi 15%’nin üzerinde olan Uygulama 5 ve 6’nın tercih ettiği yollarda eğer Uygulama 2 kullanılmış olsaydı eğim değerlerinin eşik değeri geçmesi nedeniyle en uygun yol algoritması maksimum maliyet değerine ulaşacaktı. Bu durum diğer uygulamalardaki faktörler ve farklı olasılıklar için de geçerlidir.

Tablo 4.3. Senaryo 1 maliyet değerleri detaylı karşılaştırması

Uygulama Adı	Uygulam 1 Maliyeti	Uygulama 2 Maliyeti	Uygulama 3 Maliyeti	Uygulama 4 Maliyeti	Uygulama 5 Maliyeti	Uygulama 6 Maliyeti	Uygulama 7 Maliyeti
1	3804,43	4512,79	∞	∞	∞	1326,77	∞
2	3804,43	4512,79	∞	∞	∞	1326,77	∞
3	4218,08	4969,11	5809,57	7038,69	5058,541	1229,117	6287,7
4	4218,08	4969,11	5809,57	7038,69	5058,541	1229,117	6287,7
5	4318,46	∞	∞	∞	4968,602	1996,319	6964,9
6	5148,21	∞	∞	∞	∞	924,376	∞
7	4218,08	4969,11	5809,57	7038,69	5058,541	1229,117	6287,7

4.3. Senaryo 2

Özellikle batarya doluluğu ve yol sınıfının dikkate alındığı Senaryo 2 uygulamalarında batarya doluluğunun sıfırı tükettiği durumlarla karşılaşılmıştır. Bu uygulamalar için elektrikli scooterın 100% dolu bataryalı ve 100% dolu bataryada maksimum 5000 metre yol katedebileceği tanımlanmıştır. Yol seçimlerinde bisiklet yolları ve şehir içi yollar arasında tercih farkları olduğu görülmekle birlikte arazi kullanımının dikkate alındığı uygulamalarda şehir merkezinin kullanılması yerine yeşil alanların ve sahil şeritlerinin kullanıldığı görülebilmektedir. Şekil 4.3 üzerinden senaryo 2 başlığı uygulama sonuçları harita üzerinden incelenebilir.



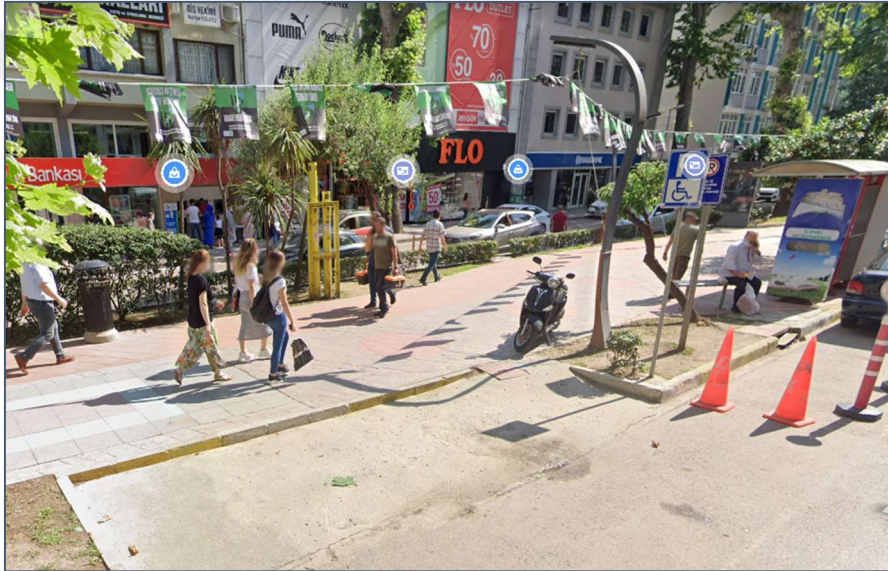
Şekil 4.3. Senaryo 2 başlığı uygulamaları harita gösterimi

Batarya doluluk değeri 0'ın altına düştüğü yol parçalarında, yol parçasının uzunluğu kadarınca yarıçaplı dairesel poligonlar atanarak, bu poligonların bulunduğu bölgelerde elektrikli scooterın enerjisinin tükeneceği bildirilmektedir. Bazı uygulamaların neredeyse aynı uzunluk ve eğim değişiminde bulunan yolları tercih etmesi nedeniyle batarya doluluğunun bitmiş olduğunu bildiren bölgelerdeki poligonlar üst üste binmektedir. Şekil 4.4 üzerinden senaryo 2 uygulamalarında batarya doluluğunun bittiği örnek aralıklar incelenebilir.



Şekil 4.4. Senaryo 2 batarya doluluğunun bittiği örnek aralıklar

Şekil 4.4 üzerinden de görülebileceği üzere aynı bölgelerde farklı uygulamaların batarya doluluklarının sıfırlanabileceği görülmektedir. Uygulamalar farklı yol tercihlerinde bulunmalarına rağmen yol uzunlukları ve yol eğimi değişimleri benzer değerlerde olabilir. Uygulama 7 için böyle bir durum söz konusu değildir. Diğer uygulamalara nazaran sahil şeridi ve yeşil alan gibi bölgelerin birincil öncelik olması nedeniyle yol tercihlerini diğer uygulamalardan çok daha farklı yapmıştır. Aynı zamanda yol sınıfı faktörünün dikkate alındığı uygulamalarda Şekil 4.5’de görünen yaya ve bisiklet yollarının tercih edildiği görülmektedir. Bu sebeple yol sınıfını faktörü dikkate alındıkça şehir içindeki bu yolların daha çok tercih edildiği görülebilir.



Şekil 4.5. Senaryo 2 Google Street View örneği

Uygulamaların genel analizlerinin bulunduğu Tablo 4.4’de tüm uygulamalar için batarya doluluğu değerlerinin 0’ın altına düştüğü görülmektedir. Uygulama 1 için kalan batarya doluluğu 1 (%) değeri beklendiği üzere en yüksek değerdir. Bunun nedeni batarya doluluğu 1 değeri ile Uygulama 1’in benzer formülle hesaplanmasıdır. Batarya doluluğu 2 (m) değerinin de Uygulama 2’de en yüksek olmasının nedeni aynıdır. Uygulama 6’da eğim değerinin 15%’yi geçtiği durumlarla karşılaşılmasından ötürü batarya doluluğu 2 (m) değeri hesaplanmamıştır. Aynı zamanda diğer değerlerin de uygulamaların belirgin özellikleriyle uyumlu olduğu söyleyebilir. Farklı uygulamalarda farklı kombinasyonlar kullanılması nedeniyle, yolun uzunluğu dışında birden fazla faktörün bulunduğu uygulamalarda faktörlerin belirgin özelliklerinin uygulamaya etki etmesi zorlaşmaktadır.

Tablo 4.4. Senaryo 2 genel bilgiler

Uyg. Adı	Toplam Uzunluk	Ort. Yol Sınıfı İnd.	Ort. Eğim İnd.	Ort. Ar. Örtüsü İnd.	Ort. Eğim (%)	Maks. Eğim (%)	Uygulama Maliyeti	Kalan Batarya Doluğu (%)	Kalan Batarya Doluğu 2 (m)
1	5496,118	1,346	0,202	0,586	3,384	11,098	5496,118	-9,922	-1606,33
2	5523,336	1,297	0,187	0,582	3,165	5,251	6497,37	-10,466	-1556,2
3	5888,799	1,024	0,17	0,597	2,883	5,251	6874,835	-17,775	-1889,89
4	6108,16	1,024	0,166	0,593	2,821	5,251	10285,125	-22,163	-2122,11
5	5771,574	1,031	0,202	0,585	3,4	5,251	5872,387	-15,431	-1937,43
6	7661,928	∞	∞	0,468	3,58	16,172	2874,209	-53,238	-
7	5771,574	1,031	0,202	0,585	3,4	5,251	9196,223	-15,431	-1937,43

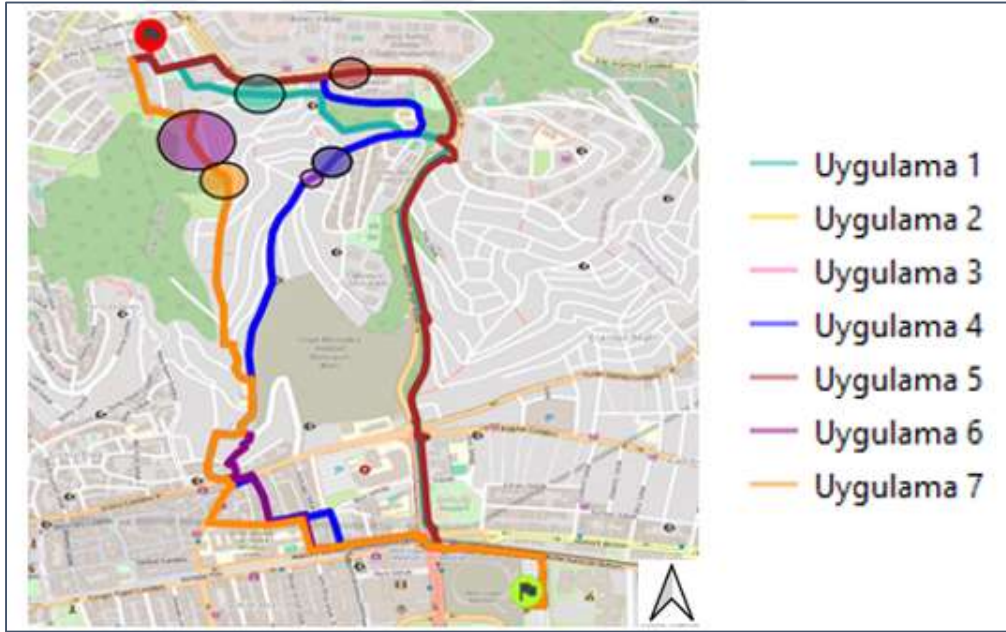
Uygulama maliyet değerlerinin birbirleriyle detaylı olarak karşılaştırılabildiği Tablo 4.5’de, Uygulama 6’da yol sınıfı indeksi ve eğim indeksinin sonsuz değerlere ulaştığı yolların diğer uygulamalara nasıl etki ettiği görülmektedir. Bu faktörlerin dikkate alındığı uygulamalarda Uygulama 6’nın kullanmış olduğu yollar uygulama maliyetini sonsuz değer yapmaktadır.

Tablo 4.5. Senaryo 2 maliyet değerleri detaylı karşılaştırması

Uygulama Adı	Uygulam 1 Maliyeti	Uygulama 2 Maliyeti	Uygulama 3 Maliyeti	Uygulama 4 Maliyeti	Uygulama 5 Maliyeti	Uygulama 6 Maliyeti	Uygulama 7 Maliyeti
1	5496,12	6552,78	8422,22	11671,5	7365,56	3249,273	10614,828
2	5523,34	6497,37	8077,08	11325,7	7103,04	3248,667	10351,71
3	5888,8	6778,12	6874,84	10245,6	5985,51	3370,726	9356,238
4	6108,16	6914,71	7052,59	10285,1	6246,05	3232,534	9478,579
5	5771,57	6912,63	7013,44	10337,3	5872,39	3323,836	9196,223
6	7661,93	∞	∞	∞	∞	2874,209	∞
7	5771,57	6912,63	7013,44	10337,3	5872,39	3323,836	9196,223

4.4. Senaryo 3

Senaryo 3 başlığı içerisinde özellikle yollardaki eğim değişiminin uygulamalara ve batarya doluluklarına etkisi incelenmektedir. Bu senaryo için birbirinden uzak, arasında eğimin yukarı ve aşağı yönde değişkenlik gösterdiği farklı yolların olduğu iki nokta seçilmiştir. Başlangıç noktasından bitiş noktasına gidişte eğime dikkat eden uygulamalar benzer benzer yolları, dikkat etmeyen uygulamalar benzer yolları tercih etmiştir. Elektrikli scooter batarya dolulukları 20% olarak belirlenmekle birlikte 7500% dolulukta kat edilebilecek maksimum yol uzunluğu 10000 metre olarak tanımlanmıştır. Bu bölgelerde yol sınıfı ve arazi kullanımı faktörlerinin çok etkili olmadığı düşünülmektedir. Başlangıç ve hedef noktaları arasında İzmit bölgesinde “Santral Yokuşu” adıyla bilinen eğimi yüksek bir yol bulunmaktadır. Eğim maliyet değişkeninin tercih edilmediği uygulamalarda bu yolun tercih edilmemesi beklenmektedir.



Şekil 4.6. Senaryo 3 başlığı uygulamaları harita gösterimi

Şekil 4.6 üzerinden de görülebildiği üzere tüm uygulamalarda elektrikli scooter batarya doluluğu 0'ı tüketmektedir. Bunun başlıca sebepleri batarya doluluğunun düşük olmasıyla birlikte tercih edilen yolların eğim değerlerinin yüksek oranlarda olmasıdır. Batarya doluluğunun tükenmesine neden olan bu uygulamada yükseklik değişikliklerinin fazla olduğu örnek bir bölgenin eş yükseklik eğrilerinin bulunduğu OSM altlık haritasındaki görünümü Şekil 4.7'deki gibidir.



Şekil 4.7. Senaryo 3 başlığı eş yükseklik haritası ile örnek bölgesi

Eğim değerlerinde İzmit ilçesinde şehir merkezinden kuzey yönüne gidildikçe yüksek oranda artışlar gözükmemektedir. Bu sebeple bu senaryoda diğer senaryolara nazaran çok daha yüksek eğim değerleri görülebilmektedir. Şekil 4.8’deki gibi eğimin yüksek olduğu bir yolu elektrikli scooterın eğim faktörünü dikkat edildiği uygulamalarda kullanabilmesi maliyeti sonsuz değere çıkarmaktadır.



Şekil 4.8. Senaryo 3 Google Street View örneği

Uygulamaların genel analizlerinin bulunduğu Tablo 4.6’da batarya doluluk değerlerine bakılarak eğim değerlerinin batarya üzerinde ne kadar büyük etkisi olduğu daha rahat anlaşılabilir. Aynı zamanda karşılaşılan maksimum ve ortalama eğim değerleri diğer senaryolara nazaran daha yüksek oranda değişmektedir.

Tablo 4.6. Senaryo 3 genel bilgiler

Uyg. Adı	Toplam Uzunluk	Ort. Yol Sınıfı İnd.	Ort. Eğim İnd.	Ort. Ar. Örtüsü İnd.	Ort. Eğim (%)	Maks. Eğim (%)	Uygulama Maliyeti	Kalan Batarya Doluğu (%)	Kalan Batarya Doluğu 2 (m)
1	2618,185	1,333	∞	0,629	7,042	29,093	2618,185	-74,545	-
2	3332,796	∞	0,289	0,557	4,672	12,302	4355,253	-122,186	-2795,97
3	3434,095	1,152	0,287	0,562	4,652	12,302	5060,401	-128,939	-2919,68
4	3383,155	1,174	0,286	0,555	4,623	12,302	7095,51	-125,543	-2850,73
5	2785,796	1,168	∞	0,645	6,517	29,093	3156,697	-85,719	-
6	2694,855	∞	∞	0,514	5,966	31,058	1585,282	-79,657	-
7	2796,154	1,154	∞	0,519	5,965	31,058	4926,584	-86,410	-

Uygulama maliyet değerlerinin birbirleriyle detaylı olarak karşılaştırılabildiği Tablo 4.7’de, eğim faktörünün dikkate alınmadığı uygulamaların yol tercihlerinin, eğim faktörü dikkate alındığında maliyeti sonsuz değere dönüştürdüğü görülmektedir. Bunların yanında bazı uygulamalarda merdiven gibi yol sınıflarının tercih edilmesi sebebi ile Uygulama 5 ve 7’de de maliyet değerlerinin sıfır ile karşılaşıldığı durumlar olmuştur.

Tablo 4.7. Senaryo 3 maliyet değerleri detaylı karşılaştırması

Uygulama Adı	Uygulam 1 Maliyeti	Uygulama 2 Maliyeti	Uygulama 3 Maliyeti	Uygulama 4 Maliyeti	Uygulama 5 Maliyeti	Uygulama 6 Maliyeti	Uygulama 7 Maliyeti
1	2618,19	∞	∞	∞	3563,28	1731,503	5294,78
2	3332,8	4355,25	∞	∞	∞	1998,31	∞
3	3434,1	4486,89	5060,4	7107,61	4007,6	2047,204	6054,806
4	3383,16	4419,95	5068,68	7095,51	4031,89	2026,828	6058,717
5	2785,8	∞	∞	∞	3156,7	1856,152	5012,849
6	2694,86	∞	∞	∞	∞	1585,282	∞
7	2796,15	∞	∞	∞	3292,41	1634,177	4926,584

5. SONUÇLAR VE ÖNERİLER

Tez araştırması içerisinde Elektrikli scooterlar için farklı faktörlerin çeşitli kombinasyonlara sahip en uygun yol uygulamalarında sonuçları nasıl etkilediği incelenmiştir. Bu uygulamaların yapılması için öncelikle faktörlerin ve yol ağı elde edilerek birbirleri ile ilişkilendirmeleri sağlanmıştır. Araştırma sonuçları CBS'nin faktörlerin ve yol ağının analizinde ve değerlendirilmesinde kullanılabileceğini göstermektedir. CBS, yol ağındaki farklı faktörlerin elektrikli scooter kullanımına olan etkisini belirlemek ve en uygun yolları bulmak için kullanılan veri analizi ve coğrafi işleme teknikleriyle entegre edilmiştir. CBS'nin sunduğu analiz yetenekleri sayesinde, bu faktörlerin kombinasyonları ve değişkenleri çeşitli senaryolarla test edilmiştir. Bu testler sonucunda, elektrikli scooter kullanıcılarının en uygun ve optimize edilmiş rotaları seçmelerine yardımcı olacak yol ağı kısıtları ve faktörler tanımlanmıştır.

OpenStreetMap, geniş bir veri topluluğuna dayanarak yol ağına ait detaylı bilgileri sağlamakta ve dünya genelindeki yol ağının kapsamlı bir temsilini sunmaktadır. Bu açık kaynak veri kaynağı, yol ağı analizleri ve coğrafi işlemler için zengin bir kaynak olarak hizmet vermektedir. Araştırma sonuçları OSM gibi açık kaynak verilerin yol ağı analizi için çeşitlilik ve topolojik ilişki konusunda güvenilir bir kaynak olduğunu doğrulamaktadır. Bu veri kaynağı, güncel bilgilere erişme avantajı sağladığından, elektrikli scooterlar için en uygun yolların belirlenmesi gibi uygulamalar için son derece değerlidir. OSM'nin çeşitlilik ve kapsamı, araştırmada kullanılan faktörlerin daha doğru bir şekilde değerlendirilmesini ve sonuçların güvenilir olmasını sağlamıştır. Bu tür açık kaynak veri kaynakları, güncel veri çeşitliliği ve geniş kapsamıyla araştırmacıların ve planlamacıların ulaşım sistemlerini daha etkin bir şekilde analiz etmesini ve optimize etmesini sağlar.

Araştırma sürecinde, Python programlama dili ve çeşitli kütüphaneleri kullanılarak açık kaynak verilerin kullanımını ve coğrafi veri analizi gerçekleştirilmiştir. Bu yaklaşımın, veri toplama, veri temizleme, analiz ve sonuçların sunumu gibi aşamalarda büyük ölçüde kolaylık görülmüştür. Python'un zengin ekosistemi, coğrafi verilerin işlenmesi ve analizinde geniş bir fonksiyonelite sunarak, açık kaynak verilerin etkili bir şekilde kullanılmasını sağlamıştır. Bu sayede, yol ağı analizi, faktörlerin etkisinin değerlendirilmesi ve en uygun yolların belirlenmesi gibi karmaşık işlemler kolayca

gerçekleştirilebildi. Bu araştırma üzerinden, Python programlama dilinin ve kütüphanelerinin açık kaynak verilerin kullanımında ve coğrafi veri analizinde önemli bir araç olduğu sonucu çıkarılabilir. Araştırmada geliştirilen programlama kodları sırasıyla ana kodlar, fonksiyon kodları ve uygulama kodları olarak Şekil A.1, Şekil B.1 ve Şekil C.1 üzerinden incelenebilir.

Araştırmada JavaScript programlama dili ve OpenLayers kütüphanesi kullanılarak, veri görselleştirme ve backend servisleriyle entegrasyon konularında çalışmalar gerçekleştirildi. JavaScript'in etkileşimli web uygulamaları geliştirmek için sunduğu zengin özellikler ve OpenLayers kütüphanesinin coğrafi verileri işlemek ve harita tabanlı uygulamalar oluşturmak için sağladığı yetenekler, araştırma sürecinin başarılı bir şekilde yürütülmesine katkı sağladı. Bu araçlar sayesinde, analiz sonuçlarının interaktif haritalarla görselleştirilmesi, verilerin gerçek zamanlı olarak güncellenmesi ve servislerle etkileşimli bir şekilde çalışabilmesi konusunda imkân sağlandı. Bu da araştırmanın etkileyciliğini ve kullanılabilirliğini artırmıştır.

Elektrikli scooter kullanıcıları için yapılan senaryo uygulamalarında uygulama tiplerinin istenilen sonuçlara varıp varamaması durumu ve sonuçların diğer uygulamalar için karşılıkları incelenmiştir. Yolun uzunluğu faktörü temel faktör olarak kullanılmasından ötürü, diğer faktörler yolun uzunluğu faktörü üzerinden türetilerek uygulamaya dahil edilmiştir. Batarya doluluğu faktörü ise yolun uzunluğu ve yolun eğiminin dikkate alınarak bataryanın hangi bölgelerde tükeneceğini gösteren bir faktördür. Yolun eğiminin dikkate alınmadığı ek bir batarya doluluğu değeri uygulama sonuçlarına eklenerek, eğimin batarya üzerinde ne kadar etkili olduğu daha da belirginleştirilmiştir.

Senaryo 1 başlığında özellikle arazinin kullanımı ve yolun sınıfı faktörlerinin önemi vurgulanmak istenmiştir. Bu faktörlerin dikkate alındığı uygulamalarda yolun sınıfı için genellikle yaya ve bisiklet yollarının, arazi kullanımı için genellikle yeşil alanların ve sahil bölgelerinde bulunan yolların daha az maliyetli olmasından ötürü tercih edilme olasılıkları daha yüksektir ve uygulama sonuçları da bu bekleneni vermektedir. Arazi kullanımı faktörünün yol uzunluğu faktörü ile tek başına etki ettiği Uygulama 6'nın tercih etmiş olduğu yollar genellikle sahil kesimlerini kullanmaktadır. Arazi sınıfı faktörünün yol uzunluğu faktörü ile tek başına etkiği Uygulama 5, yaya yolu ve bisiklet yollarını daha öncelikli olarak kullanarak diğer uygulamalardan kendini ayırabilmiştir. Bu

sonular Senaryo 1'in bařarılı sonular verdiđini gstermektedir.

Senaryo 2 bařlıđında zellikle batarya doluluđu ve yolun sınıfı faktrlerinin nemi vurgulanmak istenmiřtir. Seili iki nokta arasındaki mesafe Senaryo 1'e gre daha uzundur. Ayrıca bu iki nokta arasındaki yol sınıf eřitliliđi de yksektir. Arazi kullanımı verileri genellikle řehir ii blgelerden oluřsa da Uygulama 6 yine sahil kesimlerini tercih etmiřtir. Yol sınıfı faktrnn bulunduđu uygulamalarda ise řehir ii yaya ve bisiklet yollarının yođunluk olarak kullanıldıđı grlebilmektedir. Batarya doluluđu deđerlerinin benzer yol tercihlerini yapan uygulamalarda aynı blgede tkendiđi grlmektedir. Bu sonular Senaryo 2'in bařarılı sonular verdiđini gstermektedir.

Senaryo 3 bařlıđında zellikle batarya doluluđu ve yolun eđimi faktrlerinin nemi vurgulanmak istenmiřtir. İki nokta arasındaki ykseklilikler deđiřkendir. Bazı blgelerde eđim ařađı ynde, bazı blgelerde eđim yukarı ynde deđiřmektedir. Ancak elektrikli scooterlar iin frenleme mekanizması ve bayır yukarı gidiřlerin batarya doluluđu zerindeki etkileri eřit kabul edilmesinden dolayı eđimin yn nemsizdir. Mutlak eđim zerinden hesaplamalar yapılmıřtır. Bařlangıta 20% batarya doluluđu ile bařlanan uygulamalarda elektrikli scooterların hedefe ulařamadıđı ve tm uygulamalarda bataryaların tkendiđi tespit edilmiřtir. Diđer senaryolardan farklı olarak bu uygulamada yol eđimi faktrnn belirgin bir řekilde etki ettiđi grlmektedir. Bu sonular Senaryo 3'n bařarılı sonular verdiđini gsterse de yapılan uygulamalarda elektrikli scooterların teknik bilgileri hakkında daha detaylı alıřmalar yaparak daha hassas deđerlere ulařılabileceđi ngrlmřtr. Bunun en byk nedeni bataryaların hangi faktrlerin etkisiyle ne kadar deđiřebileceđi konusunda olması gerekenden daha fazla bilginin bu uygulamalarda kullanılabileceđidir.

Arařtırmada kullanılan faktrler aracılıđıyla farklı taleplere karřılık gelen uygulama tiplerine gre olumlu sonular elde edilmiřtir. Elektrikli scooterlar iin uygun yol uygulamalarının yapıldıđı bu arařtırmada, arařtırmanın faydaları temel olarak 2 nemli sonu zerinden yorumlanabilir:

- Enerji Verimliliđi: Yol uzunluđu, batarya doluluđu, yolun eđimi gibi faktrler dikkate alındıđında, elektrikli scooterların enerji verimliliđinin arttırılmasına ynelik stratejilerin belirlenmesi sađlanabilmektedir. Bu faktrlerin dikkate alınması ile

elektrikli scooter kullanıcılarının daha uzun mesafelerde daha az enerji tüketerek daha fazla yol katedebilmeleri sağlanabilir.

- Optimize Edilmiş Rotalar: Farklı uygulama tipleri dikkate alındığında, faktörlerin belli kombinasyonlarla birleşerek elektrikli scooter kullanıcılarına optimize edilmiş en uygun yollar, rotalar sunulabilmektedir. Bu özellik, kullanıcıların hangi tipte bir uygulama yapmak istemesi durumuna göre değişebilmektedir.

Yükselti değerleri haritası raster DEM verisi üzerinden elde edilmiştir. Ancak bu veri arazinin yüzeyi hakkında bilgi verse de insani yapılar hakkında gerekli bilgiyi vermediği için uygulamalarda bazı yanıltmacalar olabilir. Bu konudaki en belirgin örnek köprüler ve alt geçitlerdir. En uygun yol uygulamalarında bu konu için ek bir çalışma yapılmaması nedeniyle, araştırma içerisinde kullanılan bilgiler ışığında yapılacak çalışmalarda eğim faktörü konusunda hassas olmayan sonuçlar ortaya çıkabilmektedir. Bunun yanında bazı yaya köprüleri yol sınıfı olarak yaya yolu olarak gözükmekle birlikte, bazılarında asansör desteği de mevcuttur. Bu konunun araştırma içerisinde kullanılmaması sebebiyle herhangi bir çözüm üretilmemiştir. Özellikle büyük bölgeleri kapsayan poligon verilerinin olabildiğince detaylı ve raster verilerinin olabildiğince çözünürlüğünün yüksek ve çeşitli olması gerekmektedir. Bunun nedeni yapılacak senaryolar öncesi taslak çalışmalarda çözünürlük kaynaklı bazı yanıltıcı durumlar ile karşılaşılmasıdır. Örneğin yeşil alan olarak bilinen bölgelerin yerleşim yeri gibi gözükmesi seyahat için en uygun yolları arayan bir elektrikli scooter kullanıcısı için yanıltıcı bir özelliktir. Ancak yapılan düzeltmeler sonrası senaryo uygulamalarında bahsedilen durumlarla ilgili belirgin bir problemle karşılaşmadığı için bu uygulamalarda bu sorunlar söz konusu değildir.

Günümüzde yerel belediyelerin sunmakta olduğu bisiklet park noktaları gibi uygulamaların elektrikli scooterlar için de kullanılması durumunda farklı uygulama ve çalışmalarla daha farklı çeşitli araştırmaların yapılabileceği düşünülmektedir. Elektrikli scooterlar için günümüzde çeşitli ülkelerde yasal olarak bazı kurallar bulunsa da en uygun yolların incelendiği bu çalışmada İzmit ilçesinde (ve yakında bulunan diğer ilçelerde/illerde) herhangi bir kural bulunmadığı görülmüş ve eğer elektrikli scooter sürücüleri için bazı zorunlu kurallar getirilirse, bu tarz çalışmalar için de dikkate alınacak bir ön araştırma malzemesi oluşacaktır.

Araştırmanın uygulama ve faktörler bakımından daha da geliştirilebilmesi ve

iyileştirilebilmesi amacıyla bazı eklemeler yapılabilir. Bunlar:

- Ek Faktörlerin Eklenmesi: Mevcut faktörlerin yanı sıra, ek faktörlerin dikkate alınması araştırmanın kapsamını genişletebilir. Örneğin, hava durumu, trafik yoğunluğu veya yüzey kalitesi gibi faktörler, elektrikli scooter kullanıcılarının en uygun yolları seçmelerine yardımcı olabilir.
- Veri Hassasiyetlerinin Arttırılması: Veri hassasiyetinin artırılması, araştırmanın sonuçlarının daha güvenilir olmasını sağlar. Örneğin, yol uzunluğu verileri daha ayrıntılı ve gerçek zamanlı olarak elde edilebilir. Bunun için GPS verilerini veya daha yerel kurumlarda bulunan daha hassas yol ağı verileri kullanılabilir. Ancak bu durumda yol ağının oluşturulması için de ek çalışmalar gerekebilir. Öte yandan yükseklik verisi için Lidar verileri gibi nokta bulutu verileri kullanılabilir ancak veri bölgenin alanı genişledikçe veri boyutu yükseleceği için hesaplamalar daha çok zaman alabilir.
- Şarj İstasyonları ve Batarya Değişimi İstasyonları: Henüz bu konuda yerel kurumların ne kadar çalışma yaptığı tam olarak bilinmese de elektrikli scooterlar için bir şarj istasyonunun veya batarya değişimi istasyonunun bulunması ve bunun en uygun yol uygulamalarında faktör olarak kullanılması birçok konuda fayda sağlayabilir. Kullanıcılar hedef noktalarını bu istasyonlara göre belirleyebilir.

Bu eklemelerin bu konuda yapılacak araştırmalarda kullanılması, elektrikli scooter kullanıcıları için daha kapsamlı, kullanıcı odaklı ve optimize edilmiş bir yolculuk deneyimi sağlamada yardımcı olacaktır. Ayrıca, bu eklemelerin kullanılmasıyla ilgili verilerin toplanması ve analiz edilmesi, gelecekteki elektrikli scooterlar için en uygun yol uygulamalarının geliştirilmesine ve sürdürülebilir ulaşım çözümlerine katkıda bulunacaktır.

Araştırma aşamalarında karşılaşılan zorlukların başında veri kaynaklarının elde edilmesi gelmektedir. Ancak bu kaynakların nasıl elde edileceğinden ziyade, elde edildikten sonra araştırma aşamaları için ne derece uygun olabileceği bir zorluk oluşturmaktadır. Nitekim araştırma aşamalarında bazı deneme yanılma denemeleri sonucunda uygun verinin elde edilmediği farkedilerek bazı uygulama ve araştırma süreçlerinin tekrarlanması gerekmiştir. Örneğin topolojik yol ağının oluşturulması her ne kadar OSMnx kütüphanesi yardımı ile sağlanmış olsa da, araştırmanın başında OSM yol verileri ile topolojik ağın

oluşturulması için farklı metodlar denenmiştir. Bu farklı metodların araştırmanın ilk safhalarında olumlu sonuçlar verdiği görülse de tekrar tekrar kapsamlı düzeltmelerin yapılması gerekmiştir. Bu nedenle araştırmanın bu aşamasında daha uygun çözüm arayışları ile OSMnx kütüphanesinden faydalanılmıştır. Topolojik yol ağı verisinin oluşturulması dışında arazi kullanımı verisinin elde edilmesi başka bir örnek olarak verilebilir. Bu verinin ilk başta global ölçekte elde edilmesi ve çözünürlüğünün düşük olması nedeniyle araştırma içerisinde yapılan uygulamaların istenilen hassasiyette sonuçlar veremediği görülmüştür. Bu nedenle daha yerel ve çeşitliliği yüksek vektörel bir veri elde edilerek arazi kullanımı verisi faktörü olarak formatlanmıştır. Bir başka örnek olarak yükseklik verisinin yeteri kadar yüksek çözünürlüklü olmaması nedeniyle enterpolasyon yöntemi kullanılmış ve verinin hassasiyeti belli bir oranda artırılmıştır. Sonuç olarak verinin elde edilmesi ve uygulamalar için manipüle edilmesi işlemleri bu gibi çalışmalarda büyük önem arz etmektedir.

Araştırmanın beklenen sonuçlar göz önüne alındığında başarılı sonuçlar verdiği görülmektedir. Sonuçlar üzerinden yapılan incelemeler ve karşılaştırmalar ile araştırma sonuçlarının mantıklı sonuçlar olduğu incelenmiştir. Bu suretle CBS'nin, Dijkstra Algoritmasının, kullanılan CBS yazılımlarının, JavaScript programlama dilinin ve kütüphanelerinin, python programlama dilinin ve kütüphanelerinin bu alandaki araştırmalarda ve uygulamalarda kullanılabileceği sonucu ortaya çıkmaktadır. Uygulamalar için açık kaynaklar üzerinden elde edilebilen materyaller bu tarz çalışmalarda kullanılabilir. Ancak öncesinde bu materyallerin uygunluğunun detaylıca incelenmesi gerekmektedir.

KAYNAKLAR

- Badia, H., & Jenelius, E. (2021). Shared e-scooter micromobility: A review of travel behavior, sustainability, infrastructure, safety and policies. *Ön Baskı*. DOI: 10.13140/RG.2.2.19225.95841.
- Banerjee, A., & Kumar, P. K. (2022). Review of Shortest Path Algorithm. *International Journal of Computer Science and Mobile Computing*, 11(4), 1–8. DOI: 10.47760/IJCSMC.2022.V11I04.001.
- Cao, Z., Zhang, X., Chua, K., Yu, H., & Zhao, J. (2021). E-scooter Sharing to Serve Short-Distance Transit Trips: A Singapore case. *Transportation Research Part A: Policy and Practice*, 147(C), 177-196. DOI: 10.1016/j.tra.2021.03.004.
- Chen, Q., & Chen, F. (2020). The Hotspot Technologies and Cutting-edge Technologies of Organic Solar Cells. *International Journal of Engineering Science and Application*, 4(2). DOI: 10.1109/ICRERA47325.2019.8996522.
- Çöteli, R., Gökcan, A. O., & Tanyıldızı, E. (2020). The Shortest Road Algorithm Approach In Determining The Route Of Solid Waste Collection Vehicles: The Case Of Manisa 75. Yıl Neighborhood. *Turkish Journal of Science & Technology*, 15(1), 29-35.
- Çölkesen, R. (2004). *Veri Yapıları ve Algoritmalar* (3. baskı). İstanbul: Papatya Press Education.
- Ekmen, E. (2020). A Study on Performance Evaluation of Optimization Algorithms in the Shortest Path Problem. Yüksek Lisans Tezi, Ankara Yıldırım Beyazıt Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara.
- Elmasry, A., & Shokry, A. (2018). A New Algorithm for The Shortest-Path Problem. *Networks*, 74(1), 16–39. DOI: 10.1002/NET.21870
- Fidan, D., & Ulvi, A. (2022). Coğrafi Bilgi Sistemleri ve Çok Kriterli Karar Verme Yöntemleri ile Alternatif Havalimanı Konumlarının Belirlenmesi; Ankara Örneği. *Türkiye Coğrafi Bilgi Sistemleri Dergisi*, 4(2), 87-96. <https://doi.org/10.56130/tucbis.1203136>
- Gribkovskaia, I., Halskau, Ø., & Laporte, G. (2007). The bridges of Königsberg—A historical perspective. *Networks*, 49(3), 199–203. DOI: 10.1002/NET.20159
- Güneri, Ö. İ., & Durmuş, B. (2020). Classical and Heuristic Algorithms Used in Solving the Shortest Path Problem and Time Complexity. *Konuralp Journal of Mathematics*, 8(2) (2020) 279-283.
- Hofner, P., & Moller, B. (2012). Dijkstra, Floyd and Warshall meet Kleene. *Formal Aspects of computing*, 24, 459–476. DOI: 10.1007/s00165-012-0245-4.

- Karaş, İ. R., & Batuk, F. (2005). Coğrafi Bilgi Sistemlerinde Topoloji Kavramı. *TMMOB Harita ve Kadastro Mühendisleri Odası 10. Türkiye Harita Bilimsel ve Teknik Kurultayı*, Ankara, Türkiye, 28 Mart - 1 Nisan 2005.
- Plynnig, E. (2016). Route Planning Using Multiple Attributes: Finding Routes Other Than The Shortest for Bicycles. Lisans Tez Projesi, Stockholm, İsveç.
- Plynnig, E. (2016). Route Planning Using Multiple Attributes: Finding Routes Other Than The Shortest For Bicycles. Student Thesis, KTH, School of Architecture and the Built Environment, Urban Plannig and Environment, Sverige, Stockholm.
- Sarı, F., & Sen, M. (2017). Least Cost Path Algorithm Design for Highway Route Selection. *International Journal of Engineering and Geosciences*, 2(1), 1-8. DOI: 10.26833/ijeg.285770.
- Sert, E., Osmanlı, N., Eruc, R., & Uyan, M. (2017). Determination of Transportation Networks Based on the Optimal Public Transportation Policy Using Spatial and Network Analysis Methods: A Case of Konya, Turkey. *International Journal of Engineering and Geosciences*, 2(1), 27-34. DOI: 10.26833/ijeg.286034.
- Tamimi, A. A. (2015). Comparison Studies for Different Shortest Path Algorithms. *International Journal Of Computers Technology*, 14(8), 5979–5986. DOI: 10.24297/IJCT.V14I8.1857
- Taş, İ., & Akay, A. E. (2020). Using GIS Techniques for Assessment of Accessible Forest Lands by Firefighting Teams Considering Fire Risk Degrees. *European Journal of Forest Engineering*, 6(2), 87-95.
- TEMİZ, H. (2022). Recording Performances of Some File Types for Pandas Data. *Avrupa Bilim Ve Teknoloji Dergisi*, (36), 55-60. DOI:10.31590/ejosat.1103499.
- URL-1 : https://www.researchgate.net/publication/316736456_A_Survey_of_Shortest-Path_Algorithms, (Ziyaret tarihi: 09 Nisan 2023)
- URL-2 : https://www.researchgate.net/publication/310594546_A_Review_and_Evaluations_of_Shortest_Path_Algorithms, (Ziyaret tarihi: 09 Nisan 2023)
- URL-3 : https://www.researchgate.net/publication/283666925_Comparison_Studies_for_Different_Shortest_path_Algorithms, (Ziyaret tarihi: 09 Nisan 2023)
- URL-4 : <https://datateam.com.tr/graph-teorisi-nasil-ortaya-cikti-konigsbergin-yedi-koprusu>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-5 : https://www.academia.edu/40605005/Shortest_Path_Algorithms_Comparative_Study_and_Analysis, (Ziyaret tarihi: 11 Nisan 2023)
- URL-6 : https://www.youtube.com/watch?v=db_-e07jkjo, (Ziyaret tarihi: 11 Nisan 2023)

URL-7 : https://www.researchgate.net/publication/282135528_Networks?lo=1, (Ziyaret tarihi: 11 Nisan 2023)

URL-8 : https://saylordotorg.github.io/text_essentials-of-geographic-information-systems/s05-03-geographic-information-systems.html, (Ziyaret tarihi: 11 Nisan 2023)

URL-9 : https://vahaptecim.com.tr/e-kitaplar/CBS_kitap_.pdf, (Ziyaret tarihi: 11 Nisan 2023)

URL-10 : <https://gisinfo.hertfordshire.gov.uk/GISdata/vectorraster.htm>, (Ziyaret tarihi: 11 Nisan 2023)

URL-11 : <https://bancankara.com/cbs-kullanım-alanları/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-12 : <https://pro.arcgis.com/en/pro-app/latest/help/data/topologies/an-overview-of-topology-in-arcgis.htm#:~:text=Topology%20is%20the%20arrangement%20of,adjacent%20soil%20polygons%20share%20edges.>, (Ziyaret tarihi: 11 Nisan 2023)

URL-13 : <https://docplayer.biz.tr/3453173-Cbs-acısından-topolojinin-onemi-ve-alan-cizgi-geometrik-donusum-yontemleri.html>, (Ziyaret tarihi: 11 Nisan 2023)

URL-14 : https://abs.cu.edu.tr/Dokumanlar/2017/CSS119/878121946_veri_kaynak_analizi_final_ders_notu.pdf, (Ziyaret tarihi: 11 Nisan 2023)

URL-15 : https://www.academia.edu/44231680/Using_GIS_to_Determine_the_Shortest_Distance_in_the_Searching_Engines, (Ziyaret tarihi: 11 Nisan 2023)

URL-16 : <https://jerrychoices.cn/en/p/the-shortest-path/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-17 : <https://www.qgis.org/en/site/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-18 : www.openstreetmap.org/, (Ziyaret tarihi: 11 Nisan 2023)

URL-19 : <https://tr.m.wikipedia.org/> (Ziyaret tarihi: 11 Nisan 2023)

URL-20 : <https://docs.python.org/3/>, (Ziyaret tarihi: 14.02.2023)

URL-21 : <https://geopandas.org/>, (Ziyaret tarihi: 14.02.2023)

URL-22 : <https://networkx.org/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-23 : <https://osmnx.readthedocs.io/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-24 : <https://flask.palletsprojects.com/en/2.2.x/>, (Ziyaret tarihi: 11 Nisan 2023)

URL-25 : <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>, (Ziyaret tarihi: 11 Nisan 2023)

- URL-26 : <https://commons.wikimedia.org/>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-27 : <https://openlayers.org/>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-28 : <https://aws.amazon.com/tr/what-is/api/>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-29 : <https://bulutistan.com>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-30 : <https://developer.mozilla.org/en-US/docs/Web>, (Ziyaret tarihi: 11 Nisan 2023)
- URL-31 : https://www.academia.edu/31110565/All_Type_Of_Electric_Scooter_All_About_Electric_Scooter, (Ziyaret tarihi: 09 Nisan 2023)
- URL-32 : <https://orangebadge.co.uk/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-33 : <https://fancyapple.com/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-34 : <https://www.gadgetreview.com/how-far-does-an-electric-scooter-go>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-35 : <https://unagiscooters.com/scooter-articles/electric-scooters-range-vs-weight/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-36 : <https://media.istockphoto.com/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-37 : <http://viewfinderpanoramas.org/dem3.html>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-38 : wiki.openstreetmap.org, (Ziyaret tarihi: 09 Nisan 2023)
- URL-39 : <https://www.arcgis.com/apps/instant/media/index.html?appid=fc92d38533d440078f17678ebc20e8e2>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-40 : <https://www.mearth.com.au/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-41 : <https://rideaike.com/>, (Ziyaret tarihi: 09 Nisan 2023)
- URL-42 : <https://escooternerds.com/can-electric-scooters-go-up-hills/>, (Ziyaret tarihi: 09 Nisan 2023)
- Zhang, W., Chen, H., Jiang, C., & Zhu, L. (2013). Improvement and Experimental Evaluation of the Bellman-Ford Algorithm. In S. Editör (Ed.), *International Conference on Advanced Information and Communication Technology for Education (ICAICTE)* (s. 150-153). DOI:10.2991/icaicte.2013.29.
- Zhou, M., & Gao, N. (2019). Research on Optimal Path based on Dijkstra Algorithms. *Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019)* (pp. 884-892). Atlantis Press. DOI: 10.2991/icmeit-19.2019.141.

Zuniga-Garcia, N., Ruíz-Juri, N., Perrine, K., & Machemehl, R. (2021). E-scooters in urban infrastructure: Understanding sidewalk, bike lane, and roadway usage from trajectory data. *Case Studies on Transport Policy*, 9. DOI: 10.1016/j.cstp.2021.Nisan 004.





Ek-A

```
from flask import Flask, request
from flask_cors import CORS
import osmnx as ox
import networkx as nx
from lib.SPmodules import
create_shortest_path_edge_list,get_shortest_path_attributes
from create_graph import create_graph
app = Flask(__name__)
CORS(app)
G, edges = create_graph()
edges = edges.reset_index()
print("Graph oluşturuldu...")
@app.route("/nearest-node")
def nearest_node():
    x = request.args.get("x")
    y = request.args.get("y")
    node = ox.distance.nearest_nodes( G , float(x), float(y) , return_dist=False )
    data = {
        "node": {'x':G.nodes[node]['x'],'y':G.nodes[node]['y']}
    }
    return data,200
@app.route("/multi-shortest-path")
def multi_shortest_path():
    x1 = request.args.get("x1")
    y1 = request.args.get("y1")
    x2 = request.args.get("x2")
    y2 = request.args.get("y2")
    bd = float(request.args.get("bd"))
    byd = float(request.args.get("byd"))
    source_node = ox.distance.nearest_nodes( G , float(x1), float(y1) ,
return_dist=False )
    target_node = ox.distance.nearest_nodes( G , float(x2), float(y2) ,
return_dist=False )
    spApp1 = nx.shortest_path(G, source_node, target_node, weight="App1")
    spApp2 = nx.shortest_path(G, source_node, target_node, weight="App2")
    spApp3 = nx.shortest_path(G, source_node, target_node, weight="App3")
```

Şekil A.1. Ana kod dosyası

```

spApp4 = nx.shortest_path(G, source_node, target_node, weight="App4")
spApp5 = nx.shortest_path(G, source_node, target_node, weight="App5")
spApp6 = nx.shortest_path(G, source_node, target_node, weight="App6")
spApp7 = nx.shortest_path(G, source_node, target_node, weight="App7")
spApp1_lines = create_shortest_path_edge_list(spApp1,edges)
spApp2_lines = create_shortest_path_edge_list(spApp2,edges)
spApp3_lines = create_shortest_path_edge_list(spApp3,edges)
spApp4_lines = create_shortest_path_edge_list(spApp4,edges)
spApp5_lines = create_shortest_path_edge_list(spApp5,edges)
spApp6_lines = create_shortest_path_edge_list(spApp6,edges)
spApp7_lines = create_shortest_path_edge_list(spApp7,edges)
spApp1_attributes =
get_shortest_path_attributes(spApp1,edges,"App1",bd,byd)
spApp2_attributes =
get_shortest_path_attributes(spApp2,edges,"App2",bd,byd)
spApp3_attributes =
get_shortest_path_attributes(spApp3,edges,"App3",bd,byd)
spApp4_attributes =
get_shortest_path_attributes(spApp4,edges,"App4",bd,byd)
spApp5_attributes =
get_shortest_path_attributes(spApp5,edges,"App5",bd,byd)
spApp6_attributes =
get_shortest_path_attributes(spApp6,edges,"App6",bd,byd)
spApp7_attributes =
get_shortest_path_attributes(spApp7,edges,"App7",bd,byd)
data = {
    "1": {"sp" : spApp1_lines, "attributes" : spApp1_attributes},
    "2": {"sp" : spApp2_lines, "attributes" : spApp2_attributes},
    "3": {"sp" : spApp3_lines, "attributes" : spApp3_attributes},
    "4": {"sp" : spApp4_lines, "attributes" : spApp4_attributes},
    "5": {"sp" : spApp5_lines, "attributes" : spApp5_attributes},
    "6": {"sp" : spApp6_lines, "attributes" : spApp6_attributes},
    "7": {"sp" : spApp7_lines, "attributes" : spApp7_attributes},
}
return data, 200
if __name__ == "__main__":
    app.run(debug=False)

```

Şekil A.1. (Devam) Ana kod dosyası

Ek-B

```
import geopandas, pandas
import osmnx as ox
import networkx as nx
import osmnx
import numpy
sonsuz_deger = 999999999999999999
def create_graph():
    G = ox.graph_from_place(f'izmit,Kocaeli,Turkey', network_type =
    "all")
    nodes,edges = osmnx.utils_graph.graph_to_gdfs(G, nodes=True,
    edges=True, node_geometry=True, fill_edge_geometry=True)
    edges_wGrade =
    geopandas.read_file("izmit_work_qgis/edges_wSlope_3857.shp")
    edges_wGrade["grade_abs"] = abs(edges_wGrade["Slope_%"])
    edges_wGrade = edges_wGrade.set_index(["u","v","key"])
    edges_wGrade = edges_wGrade[["grade_abs"]]
    edges = edges.join(edges_wGrade, on=['u','v','key'])
    edges
    = edges.fillna(value={'grade_abs':float(edges["grade_abs"].mean())})
    def calc_grade_weight_index(grade_abs_val):
        if grade_abs_val<1:
            return 0
        elif grade_abs_val >= 1 and grade_abs_val <= 15:
            return (grade_abs_val-1)/(15-1)
        else:
            return sonsuz_deger
    edges["grade_weight"] = [calc_grade_weight_index(row.grade_abs) for
    index, row in edges.iterrows()]
    eslestirme_sozlugu = {
    "ASKERİ ALANLAR" : "Askeri Alanlar",
    "ASKERİ SAKINCALI ALANLAR": "Askeri Alanlar",
    "BARAJ VE GÖLETLER": "Su",
    "DENİZ": "Su",
    "DİĞER TARIM ALANI": "Tarım Alanı",
    "KENTSEL GELİŞME ALANLARI": "Düşük Yoğunluklu Yerleşim Alanı",
    "KENTSEL VE BÖLGESEL MERKEZİ İŞ ALANLARI(MİA)": "Orta Yoğunluklu
    Yerleşim Alanı",
    "KENTSEL YERLEŞME ALANLARI": "Yüksek Yoğunluklu Yerleşim Alanı",
    "MEZARLIKLAR": "Endüstriyel Alanlar, Resmi Kurumlar ve Mezarlıklar",
    "MİLLET BAĞÇESİ": "Yeşil Alan",
    "ORGANİZE SANAYİ BÖLGESİ": "Endüstriyel Alanlar, Resmi Kurumlar ve
    Mezarlıklar",
    "ORMAN ALANLARI": "Yeşil Alan",
    "REKREASYON ALANI": "Yeşil Alan",
```

Şekil B.1. Fonksiyon kod dosyası

```

"RESMÎ KURUM ALANLARI":"Endüstriyel Alanlar, Resmi Kurumlar ve
Mezarlıklar",
  "SOSYAL DONATI ALANLARI":"Düşük Yoğunluklu Yerleşim Alanı",
  "TARIMSAL NİTELİĞİ KORUNACAK ALAN":"Tarım Alanı",
  "YÜKSEKÖĞRETİM TESİSLERİ VE KAMPUS ALANLARI":"Orta Yoğunluklu
Yerleşim Alanı",
  "ÇÖP TOPLAMA TASFİYE YERLERİ":"Endüstriyel Alanlar, Resmi Kurumlar
ve Mezarlıklar",
  }
  puanlama_sozlugu = {
    "Su":0,
    "Yeşil Alan":5,
    "Tarım Alanı":20,
    "Düşük Yoğunluklu Yerleşim Alanı":25,
    "Orta Yoğunluklu Yerleşim Alanı":40,
    "Yüksek Yoğunluklu Yerleşim Alanı":80,
    "Endüstriyel Alanlar, Resmi Kurumlar ve Mezarlıklar":90,
    "Askeri Alanlar":100
  }
  arazi_kullanim =
geopandas.read_file("izmit_plan_50000/izmit_plan_50000_fix.shp")
  arazi_kullanim = arazi_kullanim.assign(grup=None)
  for ad in eslestirme_sozlugu:
    arazi_kullanim.loc[arazi_kullanim['ad'] == ad, 'grup'] =
eslestirme_sozlugu[ad]
  arazi_kullanim = arazi_kullanim.assign(puan=None)
  for grup in puanlama_sozlugu:
    arazi_kullanim.loc[arazi_kullanim['grup'] == grup, 'puan'] =
puanlama_sozlugu[grup]
  index_listesi = arazi_kullanim['puan'].values
  min_deger = numpy.min(index_listesi)
  max_deger = numpy.max(index_listesi)
  normalize_edilmis_degerler = (index_listesi - min_deger) /
(max_deger - min_deger)
  arazi_kullanim['agirlik_index'] = normalize_edilmis_degerler
  draft_edges_buffered = edges.copy()
  draft_edges_buffered = draft_edges_buffered.reset_index()
  draft_edges_buffered = draft_edges_buffered.to_crs(5254)
  draft_edges_buffered.geometry =
draft_edges_buffered.geometry.buffer(distance=50)
  intersection_gdf = geopandas.overlay(arazi_kullanim,
draft_edges_buffered, how='intersection')
  intersection_gdf['land_cover_weight'] = (intersection_gdf.area /
intersection_gdf.geometry.area) * intersection_gdf['agirlik_index']
  intersection_gdf =
intersection_gdf.groupby(["u", "v", "key"])['land_cover_weight'].mean().t
o_frame()
  intersection_gdf =
intersection_gdf.reset_index().set_index(["u", "v", "key"])

```

Şekil B.1. (Devam) Fonksiyon kod dosyası

```

intersection_gdf.groupby(["u","v","key"])['land_cover_weight'].mean().to_frame()
intersection_gdf =
intersection_gdf.reset_index().set_index(["u","v","key"])
edges = edges.join(intersection_gdf, on=['u','v','key'])
def calc_road_type_weight_index(highway):
    highway = str(highway)
    cycle_type = [
        "cycleway",
        "footway",
        "corridor",
        "pedestrian"
    ]
    intermediate_type = [
        "path",
        "residential",
        "tertiary",
        "tertiary_link",
        "unclassified"
    ]
    if "steps" in highway or "motorway" in highway:
        return sonsuz_deger
    for type in cycle_type:
        if type in highway:
            return 1
    for type in intermediate_type:
        if type in highway:
            return 1.25
    return 1.5
edges["road_type_weight"] =
[calc_road_type_weight_index(row["highway"]) for index, row in
edges.iterrows()]
edges["App1"] = edges["length"]
edges["App2"] = edges["length"] +
edges["length"]*edges["grade_weight"]
edges["App3"] = edges["length"]*edges["road_type_weight"] +
edges["length"]*edges["grade_weight"]
edges["App4"] = edges["length"]*edges["road_type_weight"] +
edges["length"]*edges["land_cover_weight"] +
edges["length"]*edges["grade_weight"]
edges["App5"] = edges["length"]*edges["road_type_weight"]
edges["App6"] = edges["length"]*edges["land_cover_weight"]
edges["App7"] = edges["length"]*edges["land_cover_weight"] +
edges["length"]*edges["road_type_weight"]
G = osmnx.utils_graph.graph_from_gdfs(nodes, edges)
return G, edges

```

Şekil B.1. (Devam) Fonksiyon kod dosyası

Ek-C

```
import geopandas, pandas
import osmnx as ox
import json
sonsuz_deger = 99999999999999999999
def create_shortest_path_edge_list(shortest_path,edges):
    edge_list = []
    edge_index = []
    for index in range(0,len(shortest_path)):
        if index+1<len(shortest_path):
            nodex = shortest_path[index]
            nodey = shortest_path[index+1]
            edge = edges[(edges["u"] == nodex) & (edges["v"] == nodey)]
            if len(edge)==0:
                edge = edges[(edges["v"] == nodex) & (edges["u"] == nodey)]
                if len(edge)==0:
                    continue
                edge_list.append(edge.iloc[0]["geometry"])
                edge_index.append(index)
            else:
                edge_list.append(edge.iloc[0]["geometry"])
                edge_index.append(index)
    df = pandas.DataFrame({'index_column': edge_index})
    gdf = geopandas.GeoDataFrame(df, geometry = edge_list)
    gdf = gdf.set_crs(4326, allow_override=True)
    geojson = gdf.to_json()
    return geojson
def get_shortest_path_attributes(shortest_path,edges,app,bd,byd):
    grade_meter = 0
    max_grade = 0
    all_grades = 0
    total_cost = 0
    total_length = 0
    road_types = []
    land_cover_meter = 0
    all_land_covers_index = 0
    total_cost_app1 = 0
    total_cost_app2 = 0
    total_cost_app3 = 0
    total_cost_app4 = 0
    total_cost_app5 = 0
    total_cost_app6 = 0
    total_cost_app7 = 0
    battery_charge = (bd*byd)/100
```

Şekil C.1. Uygulama kod dosyası

```

is_battery_charge_exhausted = False
area_out_of_battery_geojson = ""
battery_charge2 = (bd*byd)/100
is_battery_charge_exhausted2 = False
area_out_of_battery_geojson2 = ""
for index in range(0,len(shortest_path)):
    if index+1<len(shortest_path):
        nodex = shortest_path[index]
        nodey = shortest_path[index+1]
        edge = edges[(edges["u"] == nodex) & (edges["v"] == nodey)]
        if len(edge)==0:
            edge = edges[(edges["v"] == nodex) & (edges["u"] == nodey)]
            if len(edge)==0:
                continue
        total_length+=edge.iloc[0]["length"]
        if str(edge.iloc[0]["grade_abs"]) != "nan":
            all_grades+=int(edge.iloc[0]["grade_abs"])
            grade_meter+=1
        if str(edge.iloc[0]["land_cover_weight"]) != "nan":
            all_land_covers_index+=edge.iloc[0]["land_cover_weight"]
            land_cover_meter+=1
        total_cost+=edge.iloc[0][app]
        total_cost_app1+=edge.iloc[0]["App1"]
        total_cost_app2+=edge.iloc[0]["App2"]
        total_cost_app3+=edge.iloc[0]["App3"]
        total_cost_app4+=edge.iloc[0]["App4"]
        total_cost_app5+=edge.iloc[0]["App5"]
        total_cost_app6+=edge.iloc[0]["App6"]
        total_cost_app7+=edge.iloc[0]["App7"]
        if is_battery_charge_exhausted is not True:
            if edge.iloc[0]["grade_abs"]>15 :
                battery_charge = battery_charge - edge.iloc[0]["App1"] * 2
            else:
                battery_charge = battery_charge - edge.iloc[0]["App2"]
        if battery_charge <= 0:
            is_battery_charge_exhausted = True
            edge_metric = edge
            edge_metric = edge_metric.to_crs(5254)
            last_geometry = edge_metric.geometry
            last_geometry_length = last_geometry.length
            area_out_of_battery
last_geometry.centroid.buffer(last_geometry_length/2)
            area_out_of_battery_geojson
area_out_of_battery.to_crs(4326).iloc[0].__geo_interface__
        if is_battery_charge_exhausted2 is not True:
            battery_charge2 = battery_charge2 - edge.iloc[0]["App1"]

```

Şekil C.1. (Devam) Uygulama kod dosyası

```

last_geometry = edge_metric.geometry
    last_geometry_length = last_geometry.length
    area_out_of_battery =
last_geometry.centroid.buffer(last_geometry_length/2)
    area_out_of_battery_geojson2 =
area_out_of_battery.to_crs(4326).iloc[0].__geo_interface__
    if edge.iloc[0]["grade_abs"]>max_grade:
max_grade=edge.iloc[0]["grade_abs"]
    if edge.iloc[0]["highway"] not in road_types:
road_types.append(edge.iloc[0]["highway"])
    all_grades = all_grades/grade_meter
    all_land_covers_index = all_land_covers_index/land_cover_meter
    if total_cost_app1>sonsuz_deger:
        total_cost_app1 = -1
    if total_cost_app2>sonsuz_deger:
        total_cost_app2 = -1
    if total_cost_app2>sonsuz_deger:
        total_cost_app2 = -1
    if total_cost_app3>sonsuz_deger:
        total_cost_app3 = -1
    if total_cost_app4>sonsuz_deger:
        total_cost_app4 = -1
    if total_cost_app5>sonsuz_deger:
        total_cost_app5 = -1
    if total_cost_app6>sonsuz_deger:
        total_cost_app6 = -1
    if total_cost_app7>sonsuz_deger:
        total_cost_app7 = -1
    return {"max_grade":max_grade,
        "avg_grade":all_grades,
        "total_length":total_length,
        "road_types":road_types,
        "land_covers":all_land_covers_index,
        "total_cost":total_cost,
        "total_cost_app1":total_cost_app1,
        "total_cost_app2":total_cost_app2,
        "total_cost_app3":total_cost_app3,
        "total_cost_app4":total_cost_app4,
        "total_cost_app5":total_cost_app5,
        "total_cost_app6":total_cost_app6,
        "total_cost_app7":total_cost_app7,
        "battery_charge":battery_charge,
        "area_out_of_battery": json.dumps(area_out_of_battery_geojson),
        "battery_charge2":battery_charge2,
        "area_out_of_battery2": json.dumps(area_out_of_battery_geojson2)
    }

```

Şekil C.1. (Devam) Uygulama kod dosyası

```

def delete_edge_by_feature_name(deleted_features_names,G):
    deleted_features = [""]
    while len(deleted_features)!=0:
        deleted_features = [ (u,v) for u,v,k,d in G.edges(keys=True, data=True) if
d['highway'] in deleted_features_names]
        for u,v in deleted_features:
            G.remove_edge(u,v)
    return G
def add_elevation_values_to_graph(elevation_raster_path,G):
    ox.elevation.add_node_elevations_raster(G, elevation_raster_path, band=1,
cpus=1)
    G = ox.elevation.add_edge_grades( G , add_absolute=True , precision=3 )
    return G
def calc_grade_weight_index(grade_abs_val):
    if grade_abs_val<1:
        return 0
    elif grade_abs_val >= 1 and grade_abs_val <= 15:
        return (grade_abs_val-1)/(15-1)
    else:
        return 1
def add_grade_weight_column(edges,nodes,folder_head_name,network_type):
    nodes =
geopandas.read_file(f'{folder_head_name}_{network_type}_graph_shape\\node
s.shp")
edges.to_file(f'{folder_head_name}_{network_type}_graph_shape\\edges_for_g
rade.shp")
    edges_for_grade =
geopandas.read_file(f'{folder_head_name}_{network_type}_graph_shape\\edge
s_for_grade.shp")
    edges["grade_weight"] =
[calc_grade_weight_index(abs((float(nodes[nodes["osmid"]
row["u"]]["elevation"]) - float(nodes[nodes["osmid"]
row["v"]]["elevation"]))) * 100/row["length"])) for index, row in
edges_for_grade.iterrows())
    return edges
def calc_cost_and_get_graph(G,edges
,cost_type,folder_head_name,network_type):
    nodes, edges = ox.utils_graph.graph_to_gdfs(G)
    if cost_type == 0:
        edges =
add_grade_weight_column(edges,nodes,folder_head_name,network_type)
        edges["cost"] = edges["length"]
        print(edges.head(5))
    elif cost_type == 1:
        edges["cost"] = edges["length"]
    elif cost_type == 2:

```

Şekil C.1. (Devam) Uygulama kod dosyası

```
edges
add_grade_weight_column(edges,nodes,folder_head_name,network_type)
    edges["cost"] = edges["length"] + edges["length"]*edges["grade_weight"]
    G = ox.utils_graph.graph_from_gdfs(nodes, edges)
else:
    edges["cost"] = edges["length"]
return G
```

Şekil C.1. (Devam) Uygulama kod dosyası



KİŞİSEL YAYIN VE ESERLER

EKİZ, S., ERENER, A., & ŞİRİN, A. (2022). En uygun rüzgâr enerji santrali yerlerinin coğrafi bilgi sistemleri ile belirlenmesi: Kocaeli ili örneği. *Jeodezi Ve Jeoinformasyon Dergisi*, 9(1), 59-79. DOI: 10.9733/JGG.2022R0005.T.

ERENER, A., & ŞİRİN, A. (2024). Python-based evaluation of road network constraints for electric scooters and bicycles: Izmit Example. *International Journal of Engineering and Geosciences*, 9(1), 34-48. DOI: 10.26833/ijeg.1261677.



ÖZGEÇMİŞ

İlk, orta, lise ve üniversite öğrenimini Kocaeli’nde tamamladı. 2017 yılında girdiği Kocaeli Üniversitesi Mühendislik Fakültesi Harita Mühendisliği Bölümü’nden 2021 yılında Harita Mühendisi olarak mezun oldu. 2019-2020 yılları arasında Kocaeli Üniversitesi Harita Mühendisliği Kulübü yönetim kurulu başkanı olarak görev yaptı. 2021 yılı 18. Türkiye Harita Bilimsel ve Teknik Kurultayı’nda yürütme kurulu öğrenci temsilcisi olarak görev yaptı. 2021 Yılında Müdek görevlisi Öğrenci Değerlendirici olarak görev yaptı. 2021 yılında girdiği Anadolu Üniversitesi Açıköğretim Fakültesi Coğrafi Bilgi Sistemleri Bölümü’nden mezun oldu. 2021 Yılında girdiği Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Jeodezi ve Jeoinformasyon Mühendisliği Bölümü ve 2023 Yılında girdiği Anadolu Üniversitesi Açıköğretim Fakültesi Yönetim Bilişim Sistemleri Bölümü öğrencisi olarak öğrenimine devam etmektedir. Aynı zamanda 3 yıldır Coğrafi Bilgi Sistemleri Uzmanı olarak Piri1513 Coğrafi Bilgi Sistemleri şirketinde görev yapmaktadır.

