

İzmir Toplu Taşıma Rehberi

Hüseyin Akçekoce¹, Alp Buğra Çelik², Serdar Korhan Konaray¹,

Sinan Emre Yavuz¹, Hüseyin Akcan²

¹ İzmir Ekonomi Üniversitesi, Bilgisayar Mühendisliği Bölümü

² İzmir Ekonomi Üniversitesi, Yazılım Mühendisliği Bölümü

Özet: İzmir Toplu Taşıma Rehberi, kullanıcıların İzmir Büyükşehir Belediye sınırları içerisinde belirledikleri iki nokta arasında tüm toplu taşıma araçlarını (otobüs, vapur, metro) aktarma noktaları ile birlikte dahil ederek, varılacak noktaya mümkün olan en kısa sürede ulaşmaları için rehberlik eden, web ve mobil platformlarda hizmet veren bir yazılım projesidir. Uygulamada kullanıcıların bir noktadan ulaşmak istedikleri noktayı ya da adresi belirledikten sonra sırasıyla hangi durakları kullanmaları gerektiği, hat bilgileriyle birlikte harita üzerinden kullanıcılara sunulur. Kullanıcılar, isterlerse metro ve vapur rotalarını hesaplamaya dahil etmeyebilir, önerilen rotaya alternatif olarak başka rotalar var ise bunları da görüntüleyebilirler.

Anahtar Sözcükler: Yazılım mühendisliği, İzmir toplu taşıma rehberi, en kısa yol, mobil, ulaşım

Abstract: İzmir Mass Transit Guide is a web and mobile software project which guides for users to specify the proper and shortest way between two destinations in the boundary of İzmir Metropolitan Municipality with all public transportation vehicles (bus, metro, ferry) and transfer points. With the application, after specifying the starting and destination points, the most advantageous transportation route is displayed with expositional text or via map view. Preferably, metro and ferry routes can be eliminated from the outcome and if there exists, alternative routes can be displayed.

1. Giriş

Büyük şehirlerde yaşayan insanların ve ziyaret amacıyla gelmiş olan turistlerin karşılaştığı en büyük güçlüklerden biri ulaşımdır. Gerek akaryakıt masrafları, gerekse park yeri bulma sorunu bireylerin toplu taşıma araçlarını tercih etmesine sebebiyet vermekte, fakat bir şehre yeni taşınmış bir insanın ya da turistik gezi amacıyla gelen bir turistin bulunduğu şehirdeki toplu taşıma araçlarını etkin bir şekilde kullanabilmesi her zaman mümkün olamamaktadır. Her ne kadar bu konuda insanlara rehberlik eden haritalar olsa da haritadaki karmaşıklık ve detay insanların aradıklarını bulmasında pratik bir yöntem olmamakla beraber insanlara zaman kaybı da yaşatmaktadır. Bu haritaların birçoğu, toplu taşıma durakları ve güzergahlarıyla ilgili detaylı ve etkileşimli bir bilgi içermemektedir. Dolayısıyla, özellikle o şehre yabancı olan in-

sanlar bir noktadan diğer noktaya ulaşmak için doğru toplu taşıma duraklarını, araçlarını ve güzergahlarını kullanmakta sıkıntı yaşamaktadırlar. Ulaşım için doğru toplu taşıma araçlarının kullanılmaması da insanlar için zaman ve ekonomik kayıplara neden olabilmektedir.

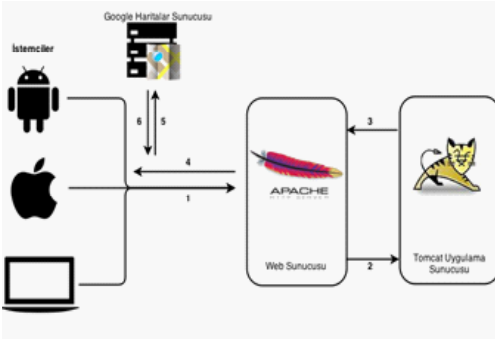
Bu sorunların bir nebze olsun önüne geçilebilir ve insanları toplu taşıma araçlarını daha fazla kullanmaya teşvik etmek amacı ile İzmir Toplu Taşıma Rehberi[1] adlı projeyi hayata geçirdik. Proje, şu an için yalnızca İzmir şehrini kapsamaktadır. Aydın Belediyesi ile de görüşmeler devam etmekte olup, projenin görme engellileri kapsayan versiyonunun da hayata geçirilmesi için çalışmalar devam etmektedir. İzmir Toplu Taşıma Rehberi Uygulaması, web üzerinden hizmet vermesinin yanı sıra, Google Play Store [2] ve Apple Store'da da kullanıcıların hizmetine açıktır.

Bildirinin ikinci bölümünde projenin tasarımı, yapılan analizler, kullanılan yöntemler, verilerin toplanması, çevrimiçi kullanıma hazır hale getirilmesi ve projeye ait teknik bilgilere yer verilmiş, proje boyunca sürdürülen çalışmalardan bahsedilmiş, bu çalışmalar sonucunda elde edilen sonuçlar açıklanmıştır. Üçüncü bölümde sonuç ve projenin uygulanabilirliğine yer verilmiştir.

2. Tasarım, Yöntem ve Prosedürler

Bu bölümde sistem altyapısı açıklanmış, uygulama geliştirme süreci boyunca yapılan araştırmalardan ve çalışmalardan bahsedilmiş, sistemin nasıl çalıştığı ile ilgili bilgilere ve kullanılan yöntemlere yer verilmiştir.

Tasarım : Şekil 1’de görüldüğü gibi istemimiz üç katmanlı istemci sunucu mimarisine sahiptir. Bu katmanlar istemci(web tarayıcısı, Ios ve Android işletim sistemli cihaz), web sunucusu ve uygulama sunucusudur (Tomcat).



Şekil - 1 : Sistem Tasarımı

a. İstemciler : İzmir Toplu Taşıma Rehberi için istemciler web tarayıcısı, Ios ve Android cihazlardır. İstemcilerin görevi web sunucusuna kullanıcıdan gelen istekleri göndermek ve gelen sonuçları gösteren arayüzü sağlamaktır.

b. Web Sunucusu : Web sunucusunun ana görevi web tarayıcısı için arayüz sağlamak ve istemcilerle uygulama sunucusu arasında iletişimi sağlamaktır. Sunucuda bulunan web servis, istek ve cevapları karşı tarafa iletmektedir.

c. Uygulama Sunucusu : Uygulama sunucusunun temel görevi İzmir’in toplu taşıma çizgesini tutmak ve gelen isteklere göre en kısa rotayı bulmaktır. İzmir çizgesinin haritasını saklamak için gerekli olan bellek küçük olduğu için, çizgeyi ana bellekte tutarak uygulama sunucusunun cevap süresinin kısalması sağlanmıştır. Uygulama sunucusu Windows Server 2008 R2 x64 (X5650 @ 2.67GHz 2.67GHz CPU, 2 GB RAM) işletim sistemi üzerinde kurulmuş olan Apache Tomcat 7.0.47 sunucusudur. Bu sunucuda, Java programlama diliyle kodlama yapılmıştır.

d. Google Haritalar Sunucusu : Google tarafından hizmete konulmuş ücretsiz çevrimiçi haritalama servisi olan Google Haritalar, uygulamamızda kullanıcılara başlangıç ve varış noktaları seçimi için harita arayüzü oluşturmakta, ayrıca sunucudan gelen en kısa yol sonuçlarının kullanıcı ekranında harita üzerinden gösterilmesini sağlamaktadır.

İstemci ve sunucular arasındaki veri trafiği Şekil 1’de gösterildiği gibi aşağıda açıklanan şekilde gerçekleşmektedir:

Aşama 1

Sistemin beklenen şekilde çalışması için kullanıcının sadece başlangıç ve bitiş adreslerini girdi olarak vermesi yeterlidir. Kullanıcı, bu adresleri metin olarak yazabileceği gibi harita üzerinden 2 nokta seçerek de belirleyebilir. Ayrıca kullanıcı, güzergah sonuçlarına metro ve vapuru dahil etmeyebilir.

Aşama 2

İstemci istekte bulunduğu anda, istek önce web sunucusuna iletilir. Web sunucusuna iletilen istek uygulama sunucusuna yönlendirir.

Aşama 3

Başlangıç ve bitiş adresleri alındıktan sonra uygulama sunucusu gelen isteğe göre adreslere en yakın durakları bulup, bu duraklar arasında aktarma noktaları da dahil olmak üzere en kısa ulaşım rotalarını hesaplar.

Aşama 4

Cevap, önce web sunucusuna iletilip, oradan istemciye iletilir. Talep ve cevaplar HTTP protokolü ile iletilmektedir. Kullanıcı girdileri servlet web adresine sorgu metni olarak eklenir. Sunucudan gelen cevap verisi ise istemcilere JSON formatında ulaşır. JSON formatının kullanılmasındaki amaç, alışverişi yapılan veri boyutunu minimumda tutmaktır. Ayrıca, bu veri formatı istemcilerde kolayca ayrıştırılabilmektedir.

Aşama 5

İstemciye ulaşan tüm durakların koordinat verileri, başlangıç ve varış noktası ile birlikte Google Haritalar Kullanıcı Arayüzü(Google Maps Api) kullanılarak Google Haritalar Sunucusu'na gönderilir. Google Kullanıcı Arayüzü, bu koordinatların harita üzerinde gösterilmesini ve bu koordinatlar arasındaki yolun harita üzerinden çizilmesini sağlar.

Aşama 6

Google Haritalar Sunucusundan döndürülen koordinat verileri ile duraklar ve arasındaki hatlar, web tarayıcısı ya da mobil cihazda Şekil 2'de görüldüğü gibi metin olarak ya da Şekil 3'de görüldüğü gibi harita üzerinden kullanıcıya gösterilir. Bu aşamanın tamamlanmasından sonra kullanıcı belirlediği noktalar arasındaki en kısa güzergahı alternatifleriyle birlikte görüntüleyebilmektedir.



Şekil - 2



Şekil - 3

Yöntem

Bu bölümde proje ile ilgili literatürde yapılan çalışmalar, gerçek verinin toplanması, ayıklanması, işlenmesi ve çevrimiçi kullanılması aşamaları açıklanmış, son bölümde elde edilen istatistiki veriler sunulmuştur.

a. Literatür Taraması

Bu süreçte uygulamanın geliştirileceği platformlar üzerine değerlendirmeler yapılmış, Phonegap, Rhodes, Titanium, MoSync altyapıları incelenmiş, bu altyapıları avantaj ve dezavantajları değerlendirilerek uygulama için en uygun olduğunu düşündüğümüz Phonegap altyapısının kullanılmasına karar verilmiştir. Ayrıca en kısa yol hesaplaması için kullanılması planlanan "Dijkstra en kısa yol algoritması"[3], "Best First Search (Hill Climbing) algoritması" [4], "A* arama algoritması" [5], alternatif güzergahlar için kullanılması planlanan Yen's algoritması[6] incelenip gerekli analizler yapılmıştır. Bu analizler sonucunda en kısa yol algoritması için Dijkstra en kısa yol algoritmasının mevcut probleme uyarlanmasına karar verilmiş, alternatif güzergahlar için Yen's algoritmasının problemin çözümü için beklenen sonucu vermeyeceği, alternatif çözümlere yönelinmesi gerektiği kararına varılmıştır.

Google Play Store ve Apple Store'da projemize benzer olduğunu düşündüğümüz "Nerde Bu Otobüs", "İstanbul Ulaşım", "SG Buses", "Live London Bus Tracker", "Edinburg Bus Tracker", "MTA Bus Tracker", "Google Transit" uygulamaları incelenmiş, bu uygulamaların İzmir Toplu Taşıma Projesi'ne olan benzerlik ve farklılıkları analiz edilmiştir.

Bu analizler sonucunda Google Transit dışında kullanıcılara tüm toplu taşıma araçlarıyla birlikte en kısa yol hesaplayan bir toplu taşıma uygulaması görülmemiştir. Google, bildiği üzere, "Google Haritalar" uygulaması ile 475'den fazla ülkede kullanıcılarına harita ve harita üzerinde yön bulma, yol tarifi alma, harita üzerinden mevcut bir konumu bulma gibi özellikler

sunmakta. Google, 2005 yılında Google Transit adını verdiği toplu taşıma uygulamasını piyasaya sürdü. Uygulama, ilk başta sadece Portland, Oregon gibi bölgeleri kapsamına karşın şu anda Amerika, Kanada, Asya, Avrupa, Afrika, Avustralya, Yeni Zelanda ve Hindistan'da birçok şehirde Google Haritalar'a entegre edilmiş bir şekilde faaliyet göstermekte. Google Transit Servisi, kullanıcının belirlediği 2 nokta arasında toplu taşıma rotası oluşturmakta. Servis, aynı zamanda toplu taşıma araçlarının kalkış ve varış saatlerini de içermekte ve kullanıcının belirlediği bu 2 nokta arasında ortalama ne kadar sürede alacağını hesaplamakta.

Google Transit birçok ülkede faaliyet göstermesine karşın bu ülkeler arasında Türkiye'yi barındırmamakta. Ayrıca yaptığımız araştırmalar sonucunda Türkiye'de aktarmalarla birlikte otobüsün yanında vapur ve metroyu da kapsayıp en kısa ulaşım yolları hesaplayan bir sistem bulunmamakta. Bu nedenle biz, uygulamamızda Google Transit Servisi'nden ilham alarak İzmir şehri için bir toplu taşıma rehberi uygulaması üretmeyi amaçladık.

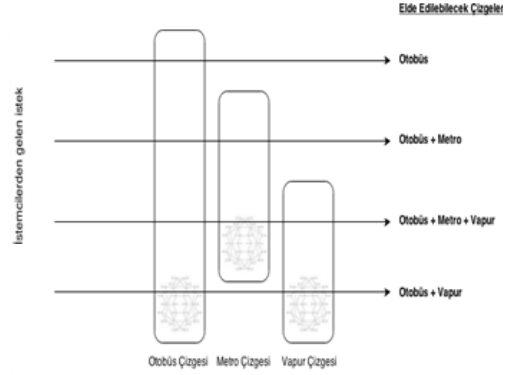
b. Verilerin Toplanması

Projemizde kullandığımız çizgeyi oluşturmak için çeşitli kaynaklardan sağladığımız verileri kullandık. Her hat için mevcut olan güzergah verisi üzerinden durak bilgilerini ve birbirine bağlı olan durakları belirledik. Birbirine bağlı her iki durak arası mesafeyi ve bu iki durak arasından geçen otobüs listesini belirledik. Bunun yanında çizge için yürüme bağlantılarını ekledik. Çizgede bulunan her durak için 400 metre yarıçapı içerisindeki tüm duraklara yürüme bağlantısı ekledik. Böylece kullanıcının gerektiği durumlarda hangi durağa yürümesi gerektiği en kısa yol hesaplamasına dahil edilmiş oldu.

c. Verilerin Ayıklanması ve İşlenmesi

Gelen veriler, uygulama sunucusunda gerekli ayıklama işlemlerinden sonra ASCII dosya olarak web sunucusuna kaydedilmektedir. Bu veriler otobüs, vapur ve metro güzergah ve duraklarını kapsayan üç katmanlı bir sistemle

tutulmaktadır. Şekil 4'de görüldüğü gibi kullanıcılar, vapur veya metroyu güzergah tercihlerine dahil etmeyebilirler.



Şekil - 4 : İstemci Güzergah Tercihleri ve Elde Edilebilir Çizgeler

Bu sürecin sonunda İzmir Büyükşehir Belediye sınırları içerisindeki 6474 otobüs durağı, 43 metro durağı ve 8 vapur iskelesinin koordinat bilgileri alınıp İzmir çizgesine eklenmiştir. Ayrıca, bu noktalar için ikili bağlantı güzergahları oluşturulmuştur. 6474 otobüs durağı için 7539 durak bağlantısı, 43 metro durağı için 82 durak bağlantısı, 8 vapur iskelesi için 14 durak bağlantısı ve tüm duraklar için yürüme bağlantılarıyla birlikte toplam 53542 sayıda bağlantı oluşturulup çizgeye eklenmiştir.

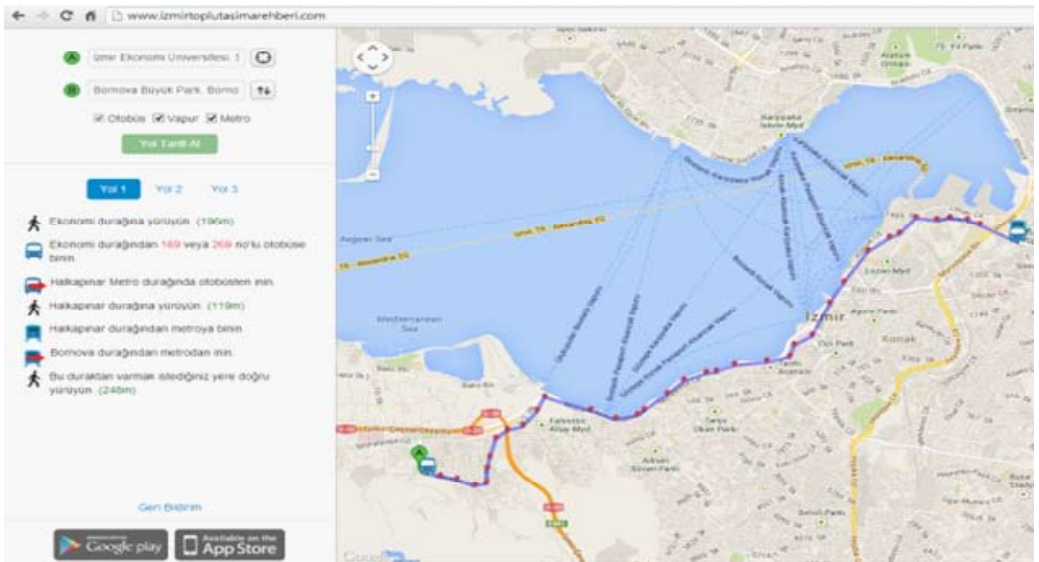
d. Verilerin Çevrimiçi Kullanılması

İstemcilerin yapmış olduğu her istekte sunucuya başlangıç noktası koordinatı, varış noktası koordinatı, metro ve vapur seçenekleri ile birlikte toplam 4 parametre gönderilir. Sunucuya istemcilerden gelen bu veriler doğrultusunda elde ettiğimiz çizge üzerinde Dijkstra en kısa yol algoritması çalıştırılır. Kullanılan Dijkstra algoritmasını kendimize göre uyarladık. Örneğin normal Dijkstra algoritmasında kısa olan bağlantı her zaman önceliğe sahiptir, fakat bizim durumumuz göz önüne alındığında, kullanıcıya çok aktarma yaptırmamak amacı ile en kısa bağlantı ile birlikte, otobüs için devam eden bağlantı da aynı zamanda önceliğe sahip olmaktadır. En kısa yolu hesapladıktan sonra

kullanıcılara alternatif güzergahlar da gösterilmek için K Shortest Path [7] problemini analiz ettik. Burada problem, kullanıcılara alternatif güzergahları mümkün olduğunca farklı olarak oluşturabilmektir. Bu problemin çözümü için edge disjoint algoritmaları [8] ile maksimum akış algoritmaları [9] incelenmiştir. Bu algoritmalar da ihtiyaç doğrultusunda modifiye edilip sistemimize uyarlanmıştır. K-Shortest Path problemi için üretilen çözümler bize birden fazla alternatif yol bilgisi bulmamızı sağlamaktadır. Yapılan hesaplama sonucunda bir ile üç adet arasında yol bilgisi durak listesi şeklinde elde edilir. Bulunan her yol bilgisi üzerindeki her iki durak arası bağlantılardan geçen otobüs listesi kesişimleri alınarak ilerlenir. Kesişen otobüs olduğu sürece geline noktaya kadar kesişim kümesindeki otobüslerden biri ile gelinebilir. Yol bilgisi üzerindeki herhangi iki bağlantı arasında eğer bir kesişim mevcut değil ise, seçili durak aktarma noktası olarak belirlenir. Kullanıcıya bu noktada bulunduğu otobüsten inip hangi otobüse binmesi gerektiği bilgisi gösterilir. Hesaplamalar tamamlandıktan sonra her yol bilgisi otobüs bilgisi ile birlikte işlemcilerle JSON veri formatında gönderilir. İstemci tarafında bu JSON verisi ayrıştırılarak

kullanıcının anlayacağı biçimde gösterilir. Bu iletişim HTTP protokolü ile sağlanır.

Bu süreç sonunda Dijkstra ve A* algoritmaları kullanılarak en kısa güzergahlar hesaplanmış, bu algoritmaların performansları karşılaştırılmıştır. Güzergaha göre değişmekle beraber A* algoritmasıyla ortalama olarak %60 performans artışı sağlanmıştır. Ancak bir kere hesaplandıktan sonra bir noktadan diğer tüm noktalara olan en kısa yolun bilinmesi ve bu sayede sık kullanılan rotaların ön belleğe kolayca kayıt edilebilmesi sebebiyle Dijkstra en kısa yol algoritması tercih edilmiştir. En kısa güzergah bulunduktan sonra bu rota üzerinden geçen otobüs, metro ve vapur güzergahları, aktarmalarla birlikte kullanıcıya en avantajlı olanı gösterecek şekilde hesaplanmıştır. Sonrasında en kısa güzergaha alternatif olabilecek iki güzergah daha hesaplanmıştır. Bu güzergahlar hesaplanırken edge disjoint algoritmaları ile max flow algoritmaları incelenmiş, bu algoritmalar mevcut sorunumuza çözüm olacak şekilde uyarlanmıştır. Ayrıca otobüs, metro ve vapur için üç ayrı çizge oluşturulup kullanıcının tercihine göre vapur ve metro çizgelerinin güzergah sonuçlarından elenmesi sağlanmıştır.



Şekil - 5 : İzmir Toplu Taşıma Rehberi Web Sitesi

e. Uygulama Sonuçları

İzmir Toplu Taşıma Rehberi Projesi planlanan şekilde hayata geçirilmiştir. Uygulamaya, www.izmirtoplutasimarehberi.com adresinden erişim mümkündür. Ayrıca uygulama, Google Play Store [10] ve Apple Store'da ücretsiz olarak mobil kullanıcılarının da erişimine açıktır.



Şekil - 6 : Web Sunucusu Gelen İstek Çizelgesi

10/11/2013 tarihi itibarıyla uygulama 2209 android cihaza, 3000'e yakın apple cihaza indirilmiştir. Şekil 6'da web sunucusuna yapılan istek sayısı(web ve mobil platformlardan gelen) tarih çizelgesi görülmektedir. 01/06/2013 ile 01/12/2013 tarihleri arasında toplanan verilere göre, web sunucusuna günde ortalama 95 istek gönderilmiş, standart sapma yaklaşık 46 olarak hesaplanmıştır.

Windows 7 işletim sistemli, Intel Celeron 2.0 GHZ işlemci hızı olan 4GB ram özellikli bir notebook bilgisayarda en kısa 3 güzergah, lokal sunucuda yapılan 20 deneme sonucunda ortalama olarak 0.8 saniyede hesaplanmıştır.

Gelen geri bildirimler sonucunda İzmir Toplu Taşıma Rehberi projesi için gerekli görülen yerlere düzenlemeler ve iyileştirmeler yapılmaktadır.

3. Sonuç

Teorik araştırmaların pratik uygulamaya dönüştürüldüğü İzmir Toplu Taşıma Rehberi projesi, İzmir Büyükşehir Belediye sınırları içerisindeki toplu taşıma araçlarının kullanımı konusunda insanlara yardımcı olmayı amaçla-

yan, web ve mobil platform destekli bir yazılım projesidir. Bu proje ile kullanıcılar belirledikleri iki nokta arasında tüm toplu taşıma araçlarıyla aktarma noktaları da dahil edilerek en avantajlı şekilde nasıl gidecekleri ile ilgili fikirler edinebilecek, toplu taşıma seçenekleri ile alternatif güzergahlar görüntüleyebilecek, bu güzergahlar ile ilgili bilgilere ulaşabileceklerdir. Toplu taşıma araçlarının durak ve güzergah bilgilerinin sağlanması halinde proje tüm şehirlere uyarlanabilecektir. Ayrıca proje, HTML5 ve CSS3 destekleyen tüm platformlarda hizmet verebilmektedir.

4. Kaynaklar

[1] Melihcan Türk, Gözde Kantarcı and Hüseyin Akcan, "Web-Based Mass Transit Guide", In Proceedings of the 4th International Student Conference on Advanced Science and Technology, Izmir, Turkey, (May 2010)

[2] <https://play.google.com/store>
<https://itunes.apple.com>

[3] Dijkstra, E. W. "A note on two problems in connexion with graphs". *Numerische Mathematik* 1: 269–271. doi:10.1007/BF01386390 (1959).

[4] Helmut Horacek, "A Best-First Search Algorithm for Generating Referring Expressions", *Uniyersitat des Saarlandes, FR 6.2 Informatik Postfach 151150, D-66041 Saarbdicken, Germany*

[5] Stuart Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach (3rd Edition)", (2009)

[6] Ernesto Q.V. Martin, Marta M.B. Pascoal, "A new implementation of Yen's ranking loopless paths algorithm", *Centro de Informática e Sistemas, Departamento de Mathematica Universidade de Coimbra, Apartado 3008, 3001-454 Coimbra, Portugal* (2002)

[7] David Eppstein “Finding the k Shortest Paths”, Department of Information and Computer Science University of California, Irvine, CA 92717 Tech. Report 94-26 (May 31,1994)

[8] Michael J. Dinneen, Yun-Bum Kim, and Radu Nicolescu, “Distributed, Parallel, and Cluster Computing” Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand, EPTCS 40, pp. 121-141, (2010)

[9] Eugene Lawler (2001). "4.5. Combinatorial Implications of Max-Flow Min-Cut Theorem, 4.6. Linear Programming Interpretation of Max-Flow Min-Cut Theorem". *Combinatorial Optimization: Networks and Matroids*. Dover. pp. 117–120. ISBN 0-486-41453-1.

[10] <https://play.google.com/store/apps/details?id=izmir.toplutasimarehberi>
[https://itunes.apple.com/gb/app/izmir -toplutas-ma-rehberi/id655839689](https://itunes.apple.com/gb/app/izmir-toplutas-ma-rehberi/id655839689)