

Gezgin Kargo Problemi

Engin Yenice

Kocaeli Üniversitesi
Bilgisayar Mühendisliği

190201133
enginyenice2626@gmail.com

Özet—Kocaeli de bulunan bir nakliye firmasının en az maliyet ile ürünlerini bir veya birkaç (maksimum 10) şehire götürüp tekrar geri Kocaeli'ye gelmesi için en az maliyetli güzergahın bulunması ve bunun görüntülenmesi.

Anahtar Kelimeler—nakliye, mesafe, şehir, en kısa mesafe, arama

I. GİRİŞ

Gezgin Satıcı Probleminde amaç, bir satıcının bulunduğu şehirden başlayarak her şehre sadece bir kez uğradıktan sonra başladığı şehre dönebilmesi için en kısa yolun bulunmasıdır.

II. KULLANIM KOLAYLIĞI

A. En Maliyetsiz Yolun Bulunması

Bu uygulamayı kullanan kişi/kişiler bir ya da birkaç şehire gidip geri Kocaeli şehrine gelmek şartıyla en kısa yolu 5 yolu bulacaklardır. Bunlar arasından seçimler yaparak istedikleri güzergaha en maliyetsiz yoldan giderek hem zamandan hemde yakıttan tasarruf edeceklerdir.

III. PROJE SÜRECİ

A. Şehirlerin Listesinin Oluşturulması

İlk olarak şehirlerin komşuluklarının listesinin çıkartılması gerekiyordu. Bu bilgileri Programlama Laboratuvarı – I Bağlı Liste projesi ile önceden hazırlamıştık.

İkinci olarak komşulukları bulunan şehirlerin birbirleri arasındaki mesafelerin alınması gerekiyordu. Bunun için Karayolları Genel Müdürlüğü'nün şehirler arası mesafeler tablosu kullanıldı. Oluşan listesin örneği aşağıda belirtilmiştir.

26,ESKİŞEHİR,AFYONKARAHİSAR-144,KONYA-338,ANKARA-233,BOLU-291,BİLECİK-83,KUTAHYA-78

Örnekteki her bir virgül(,) ile ayrılan alan farklı bir değeri temsil etmektedir. Detaylıca açıklayalım:

- 26 : Ana şehrin plakası
- Eskişehir: Ana şehrin ismi
- Afyonkarahisar-144 : Şehrin komşusunun adı – Ana şehre olan uzaklığı (Devamındaki şehirlerde aynı şekilde kategorize edilmiştir.)

B. Listedeki Verilerin İşlenmesi ve Komşuluk Matrisi

Projenin verileri hazırlanmıştı ve bu verilerin işlenmesi gerekiyordu. Oluşturulan veriler oluşturulan ağaç yapısına göre uygulamaya aktarılacak şekilde oluşturuldu.

Grafları işlenerek komşuluk matrisi oluşturuldu.

C. En Kısa Mesafenin Oluşturulması

Bu konu altında bir çok deneme yapıldı. Dijkstra algoritması, A* algoritması, Karınca kolonisi algoritması, Sezgisel yöntemler. İçlerinden en rahat entegre edilebilen Dijkstra oldu ve bu yöntem üzerinde çalışmalara devam edildi.

İlk olarak bir şehirden başka tüm şehirlere Dijkstra yöntemi ile en kısa yol bulundu.

İkinci olarak Dijkstra'nın doğruluğu teyit edildikten sonra kullanıcı tarafından seçilen bir şehir listesi arasında en kısa mesafenin bulunacağı şekilde Dijkstra yapma işlemleri gerçekleştirildi. Burada Kocaeli – Seçilen Şehirler – Kocaeli turunu yapan en kısa güzergah bulunuyordu. Bunun ardından bu yol sayısını çoğaltmamız gerekiyordu. Aralarından aranan olmayan komşuları eleyerek yeni listeler oluşturuldu ve bu listeler üzerinde tekrar Dijkstra yapıldı.

Son olarak oluşan yollar belirli listelere aktarıldı ve kullanıcının yorumlayacağı şekilde ekranda gösterildi.

D. Form Tasarımının Yapılması

Form daki verileri 4 başlık altında ayrılarak programa eklendi.

1) Şehir Listesi ve Yol Bul

a) Gidilecek Yer

Kocaeli hariç tüm şehirler checkboxList içerisinde listelenmektedir.

b) Seçilen Yol Listesi

Gidilecek Yer listesindeki seçilen ve ardından Yol bul butonuna basıldıktan sonra seçilen şehirlerin listelendiği bir listBox'dır.

c) Bulunan Yol

Bulunan yolların listelendiği bir listBox'dır. Buradan tüm yolları veya tek tek yolları haritada çizilmek üzere seçebilirsiniz.

d) Yollar Arasındaki En Kısa Mesafe

Bulunan yollar arasındaki en kısa mesafesi yanına ekleneceği bir label.

e) RichTextBox

Bulunan en kısa 5 yolun uğrayacağı güzergahlar ile beraber ve mesafeleri ekleyerek listelendiği bir alandır. Örnek Çıktı:

KOCAELİ(41) - SAKARYA(54) - BİLECİK(11) - ESKİŞEHİR(26) - BİLECİK(11) - SAKARYA(54) - KOCAELİ(41)

----[TOPLAM MESAFE: 438]----

KOCAELİ(41) - SAKARYA(54) - BİLECİK(11) - ESKİŞEHİR(26) - KUTAHYA(43) - BİLECİK(11) - SAKARYA(54) - KOCAELİ(41)

----[TOPLAM MESAFE: 545]----

KOCAELİ(41) - SAKARYA(54) - BİLECİK(11) - ESKİŞEHİR(26) - BOLU(14) - SAKARYA(54) - KOCAELİ(41)

----[TOPLAM MESAFE: 661]----

KOCAELİ(41) - SAKARYA(54) - BİLECİK(11) - ESKİŞEHİR(26) - AFYONKARAHİSAR(3) - KUTAHYA(43) - BİLECİK(11) - SAKARYA(54) - KOCAELİ(41)

----[TOPLAM MESAFE: 711]----

KOCAELİ(41) - SAKARYA(54) - BİLECİK(11) - ESKİŞEHİR(26) - ANKARA(6) - BOLU(14) - SAKARYA(54) - KOCAELİ(41)

Kocaeli – Eskişehir Arası örnek bir RichTextBox çıktısı.

f) *Yol Bul Butonu*

Gidilecek yerlerin seçiminden sonra bu butona tıklanarak yolların bulunması sağlanmaktadır.

g) *Yolu Çiz Butonu*

Bulunan yol listesinden seçilen yolun haritaya çizilmesini sağlar.

2) *Harita Gösterimi*

Bu menüde sadece harita yer almaktadır. Yolu çiz seçeneği seçilerek bu ekrana gelindiğinde sizi ayrıca bir form karşılayacaktır.

a) *Yol Bilgileri Formu*

Seçilen yolun mesafesi güzergahı ve seçilen yol listelenmektedir.

3) *Komşuluk Matrisi*

Bu menüde komşuluk matrisinin richTextbox üzerinde gösterimi sağlanmıştır.

4) *Hakkında*

Proje hakkında küçük bilgilerin yazıldığı menü

E. *Çıktı Dosyasının Oluşturulması*

Bulunan yolların çıktı dosyasına aktarılması gerektiği için bir çıktı dosyası oluşturulmuştur. Çıktı dosyasının temel yapısı Form Tasarımları menüsündeki RichTextBox 'ın yapısıyla aynıdır.

Exe dosyasının bulunduğu klasörde eğer Outputs klasörü varsa onun içersine kayıt etmektedir. Eğer yoksa kendisi otomatik olarak Outputs klasörü açar ve içersine kayıt eder. Outputs klasörü içersine kayıtlar

YIL-AY-GÜN-SAAT-DAKİKA-SANİYE.TXT

şeklinde tutulmaktadır. Program açıldığında kapanana dek yapılan tüm işlemler aynı txt dosyası içersinde tutulmaktadır. Örnek ile açılacak olursak: Programı açıp 10 ayrı arama yaptınız ve kapattınız. Bu tek bir txt içerisinde tutulacaktır. 1 tane arama yapıp kapattınız. Buda tek bir txt içerisinde tutulacaktır.

IV. DOSYA YAPISI

Projenin kodlama kısmında dosya yapısı 2 ana klasörde toplanmıştır.

1) *Data Klasörü*

Bu klasörde genel olarak tutulması gereken veriler ve graf yapısının temeli tutulmaktadır.

a) *Ddosya.cs*

Dosya yolları bu dosya içerisinde tutulmaktadır.

b) *Dsehir.cs*

Şehirlerin tutulma şekli ve temel graf yapısı burada tutulmaktadır.

2) *Manager Klasörü*

a) *DosyaManager.cs*

Dosyadan veri çekme ve dosyaya veri yazma işlemleri için kullanılır.

b) *KomsulukMatrisiManager.cs*
Komşuluk matrisini oluşturur.

c) *SehirManager.cs*

Şehir bilgilerini, komşuluk bilgilerini, mesafelerini ve kordinatlarını ekleme işlemini yapar.

d) *YolBulucu.cs*

Yol bulma işlemini gerçekleştirir.

V. NASIL KULLANILIR

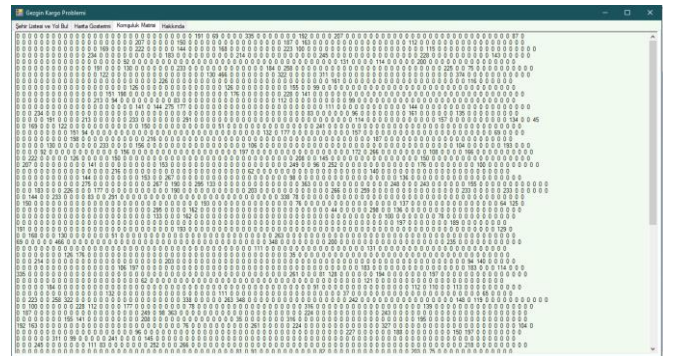
A. *Formların Genel Resimleri*



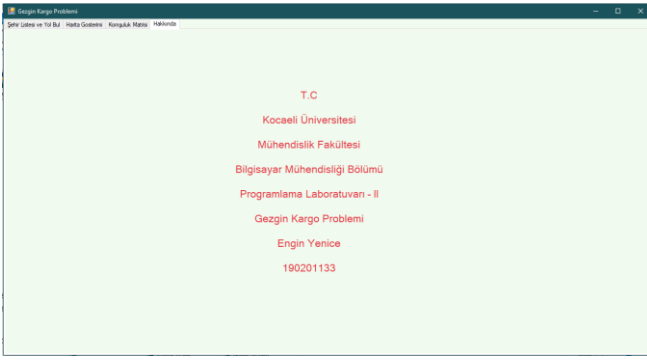
(Şehir Listesi ve Yol Bul)



(Harita Gösterimi)



(Komşuluk Matrisi)

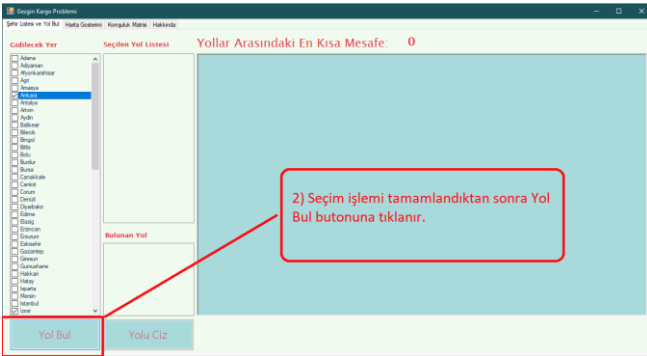


(Hakkında)

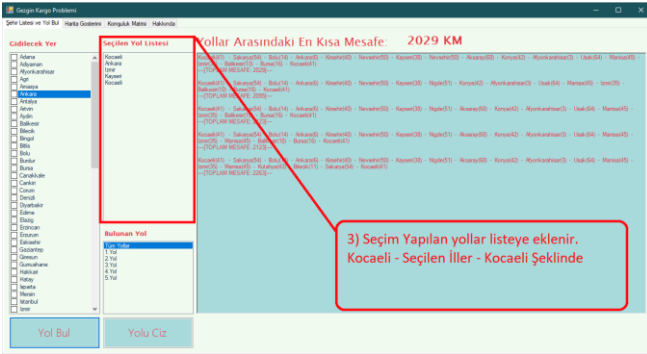
B. Adım Adım Kullanım



Adım 1) Gidilecek yer seçilir. Burada Ankara, Kayseri ve İzmir seçildi.

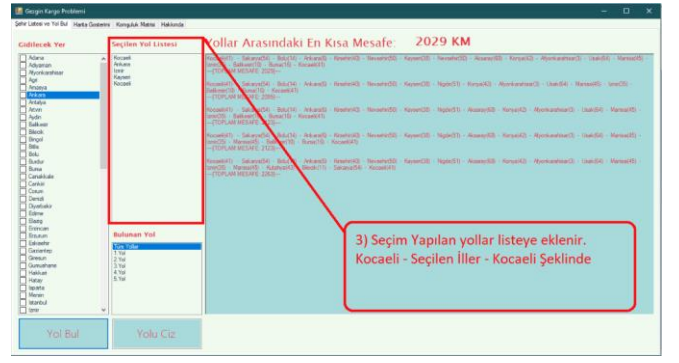


Adım 2) Seçim işlemi tamamlandıktan sonra Yol Bul butonuna tıklanır ve yollar bulunur.

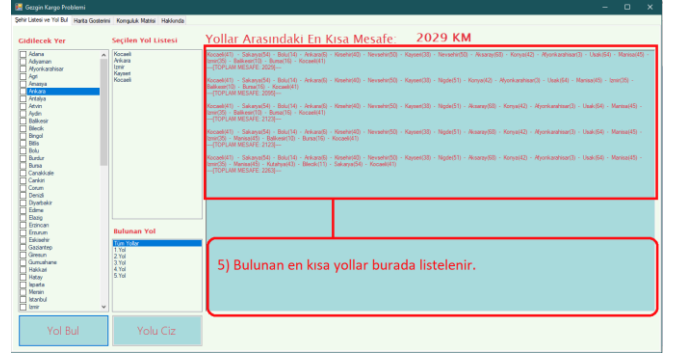


Adım 3) Seçim yapılan yollar bu listeye eklenir.

Ekleme Şekli: Kocaeli – Seçilen İller - Kocaeli



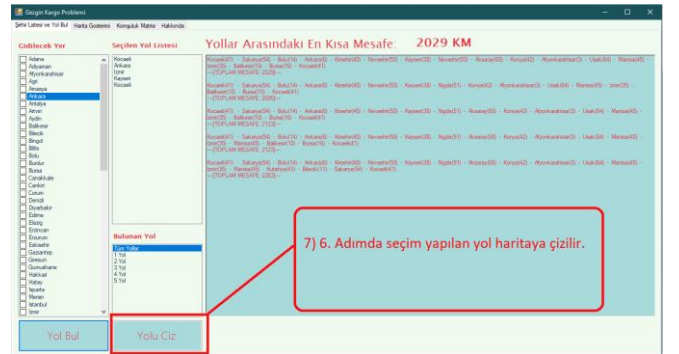
Adım 4) Bulunan yollar arasındaki en kısa mesafe buraya yazılır.



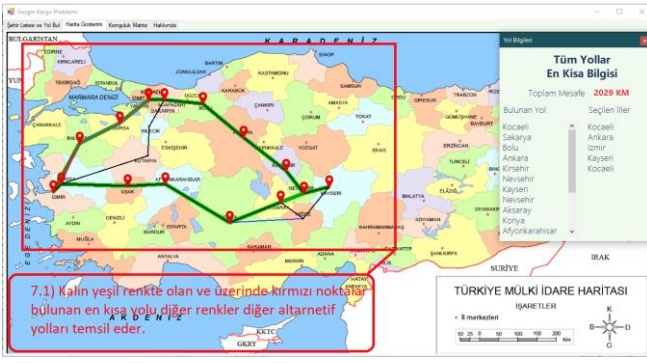
Adım 5) Bulunan en kısa yollar burada listelenir.



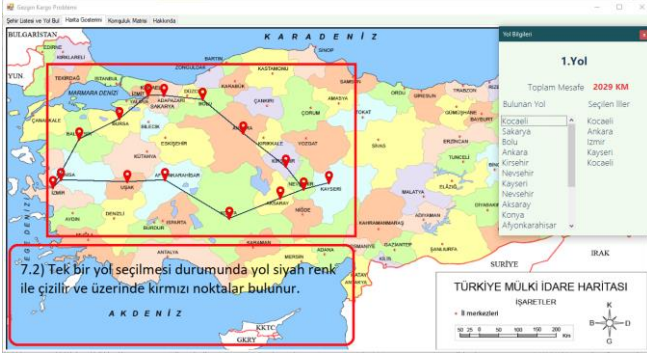
Adım 6) Bulunan yollar buraya aktarılarak seçim yapılması istenir.



Adım 7) 6. Adımda seçim yapılan yol haritaya çizilir.



Adım 7.1) Kalın yeşil renkte olan ve üzerinde kırmızı noktaları bulunan en kısa yolu temsil eder. Diğer renkler diğer yolları temsil etmektedir.



Adım 7.2) Tek bir yol seçilmesi durumunda yol siyah renk ile çizilir ve üzerinde kırmızı noktalar bulunur.

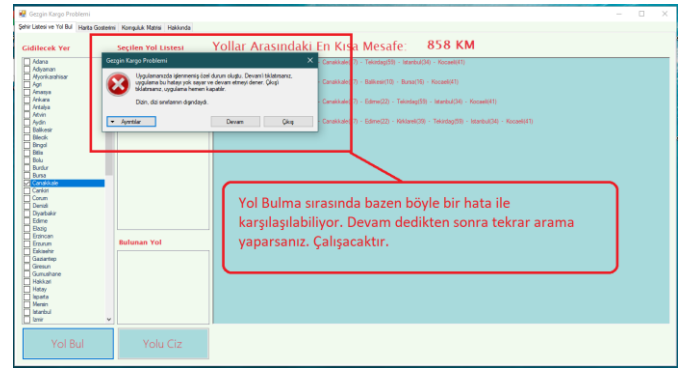


Adım 8) Yol çizildiğinde Yol Bilgisi adında bir form açılmaktadır. Bu form:

- Tüm Yollar seçilirse eğer aralarında en kısa yolun bilgileri vermektedir.
- Tek bir yol seçilirse eğer o yol ile ilgili bilgileri vermektedir.



Adım 9) Menüler arasında gezinme



BUG) Çok nadir durumlarda bu şekilde bir hata vermektedir. Hataya devam dedikten sonra tekrar yol bul diyerek işleme devam edebilirsiniz.

VI. KABA KOD

Kabakod.docx olarak kayıt edilmiştir.

I. KAYNAKÇA

- [1] <https://www.geeksforgeeks.org/> (Geek For Geeks)
- [2] <https://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Root/default.aspx> (Karayolları Genel Müdürlüğü)
- [3] <https://www.cografiyaegitimi.biz/> (Coğrafya Eğitimi)
- [4] <https://www.w3schools.com/cs/> (W3 Schools)
- [5] <http://bilgisayarkavramlari.sadievrenseker.com/> (Bilgisayar Kavramları)