Marmara Fen Bilimleri Dergisi 2018, 1: 40-49
DOI: 10.7240/marufbd.364043

**RESEARCH ARTICLE** / ARAŞTIRMA MAKALESİ

# Ege Denizi Limanlarının Arasındaki En Kısa Yol Probleminin Çözümünde Genetik Algoritma Yaklaşımı

## A Genetic Algorithm Approach to Shortest Path Problem Among the Ports on the Aegean Sea

**Timur İNAN[1], Ahmet Fevzi BABA[2]**

*[1]İstanbul Arel Üniversitesi, Meslek Yüksekokulu, Bilgisayar Programcılığı Programı, 34295 Sefaköy – Küçükçekmece, İSTANBUL*
*[2]Marmara Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 34722, Kadıköy / İSTANBUL*

**Öz**

Genetik algoritma, evrimsel bir algoritma olup, en sık kullanılan problem çözümleme algoritmalarından biridir. En kısa yol bulma problemi ise denizcilik, havacılık, savunma ve yük taşımacılığı gibi önemli alanlarda çokça çalışılan bir konudur. Bu çalışmada Ege denizi kıyısında bulunan limanların arasındaki en kısa yolun bulunmasında genetik algoritma kullanılmıştır. Ege denizi üzerinde bulunan 61 liman ve bu limanlar arasında yapılacak olası seyir için rota belirlemede tanımlanan 604 adet düğüm en kısa yol probleminin genetik algoritma yardımıyla belirlenmesi amacıyla kullanılmıştır. Çalışmayı farklı yapan kısımlar; limanların ve düğümlerin gerçek harita üzerinde ve gerçek koordinatlara göre kayıt altına alınmış olması ve düğüm haritasının büyük ölçekli olmasıdır. Ege denizi üzerinde bulunan 61 adet liman arasında yapılabilecek olası seyir durumunda izlenilebilecek rotalar genetik algoritma yardımıyla hesaplanarak sonuçlar ortaya konulmuştur.

**Anahtar Kelimeler:** Genetik Algoritma, En Kısa Yol, Ege Denizi, Optimizasyon

**Abstract**

The genetic algorithm is an evolutionary algorithm that is one of the most frequently used problem-solving algorithms. As for the problem of finding the shortest route, it is a subject that is widely studied in important areas such as shipping, aviation, defence and freight forwarding. In this study a genetic algorithm was used to find the shortest paths between Greek and Turkish ports located on the Aegean Sea. Sixty one ports located on the Aegean Sea and 604 nodes that were identified for a possible course between these ports were used to determine the shortest path problem with the help of the genetic algorithm. The originality of this study is that the ports and nodes were identified according to real coordinates and a large scale node map was used. The routes that can be taken in the case of a possible course between the 61 ports located on the Aegean Sea were calculated with the help of the genetic algorithm and the results were put forward.

**Keywords:** Genetic Algorithm, Shortest path, Aegean Sea, Optimization

## I. GİRİŞ

Karınca kolonisi algoritması, genetik algoritma, parçacıklı sürü algoritması gibi üst sezgisel algoritmalar denizcilik, havacılık ve kara taşımacılığı gibi sektörel alanlarda en kısa yol problemlerinin çözümünde yaygın olarak kullanılmaktadır [1], [2], [3]. Bu yöntemler denizcilik alanında da filo optimizasyonu [4], rota optimizasyonu [5], çarpışma önleme [6] gibi alanlarda da çözüm olmuştur. Ege denizi bağlamında genetik algoritma kullanılarak yapılan rota optimizasyonu literatürde bulunamamıştır.

Bu çalışmada Ege denizi üzerinde bulunan Yunanistan ve Türkiye limanları temel alınmıştır. Ege denizi üzerinde bulunan 61 adet limanın arasında yapılabilecek olası seyir durumu için 604 adet düğüm belirlenmiştir. Rota belirlemede olası düğümlerin seçilerek en uygun rotanın bulunması genetik algoritma kullanılarak yapılmıştır.

Limanların gerçek koordinatları ve düğümlerin koordinatları için World Port Index, liman otoritelerinin web siteleri, Google Maps uygulaması ve liman bilgileri içeren web siteleri kullanılmıştır [7], [8], [9] , [10], [11]. Derinlikler için

**Sorumlu yazar/Corresponding Author:** Timur İNAN, Tel: 05376696627, e-posta: timurinan@arel.edu.tr
**Gönderilme/Submitted:** 08.12.2017, **Düzenleme/Revised:**08.03.2018, **Kabul/Accepted:** 22.03.2018.

General Bathymetric Chart of Oceans (GEBCO) kuruluşunun internet sitesinden talep edilen bilgiler kullanılmıştır. Haritalama için MATLAB programının Mapping Toolbox fonksiyonları kullanılmıştır [12]. Ege denizi haritası oluşturmak için Yunanistan kıyılarını ve Türkiye kıyılarını içeren shape file dosyaları kullanılmıştır. Haritanın oluşturulması için QGis programından yardım alınmıştır çünkü her iki ülkenin limanlarını barındıran ortak bir harita bulunamamıştır. İki shape file birleştirilerek ortaya yeni bir shape file çıkarılmıştır. Shape file kullanılmasının nedeni; shape file haritanın detaylı olarak tüm kıyıları alması ve internet bağlantısı gerektirmemesidir [13], [14], [15].

Genetik algoritmanın giriş değişkenlerinin problem üzerindeki etkisi incelenmiştir. Değişken birey sayısı ve iterasyon sayısı miktarlarında yapılan tecrübelerde elde edilen çözümler sonuç bölümünde grafiksel olarak gösterilerek genetik algoritmanın problemin çözümündeki performansı değerlendirilmiş ve başarılı sonuçlar elde edildiği görülmüştür.

### 1.1. Limanlar

Yunanistan ve Türkiye kıyılarında Ege denizinde seyir yapılarak ulaşılabilecek limanlar ve koordinatları kayıt altına alınarak harita üzerinde limanlar belirlenmiştir. Şekil 1'de limanlar ve düğüm haritasının bir kısmı görülmektedir.

Kayıt altına alınan limanların adları ve koordinat bilgileri Tablo 1'deki gibidir.

**Tablo 1.** Ege denizi limanları ve koordinatları

| Liman Adı | Enlem (Kuzey) | Boylam (Doğu) | Liman Adı | Enlem (Kuzey) | Boylam (Doğu) |
|---|---|---|---|---|---|
| Achladi | 38,9166 | 23,52183 | Miliana | 39,1740 | 23,2131 |
| Akra Kavonisi | 35,5145 | 23,64159 | Milos | 36,7211 | 24,4438 |
| Alexandroupoli | 40,8329 | 25,90073 | Mitilini | 39,1006 | 26,5848 |
| Aliağa | 38,8552 | 26,93722 | Navplio | 37,5636 | 22,7829 |
| Andros | 37,8445 | 24,94554 | Nemrut Limanı | 38,7919 | 26,8606 |
| Aspropirgos | 38,0370 | 23,56076 | Neon Karlovas | 37,7993 | 26,6855 |
| Ayios Nikolaos | 35,1940 | 25,74136 | Nisos Naxos | 37,1078 | 25,3641 |
| Ayvalık | 39,3217 | 26,63370 | Ormos Aliveriou | 38,3899 | 24,0450 |
| Bodrum | 37,0135 | 27,42766 | Ormos Mikro Vathi | 38,4343 | 23,5993 |
| Chalkis | 38,4430 | 23,60909 | Pachi Oil Terminal | 37,9725 | 23,3628 |
| Dhiavlos Steno | 38,4498 | 23,59945 | Perama | 37,9571 | 23,5663 |
| Dikili | 39,0806 | 26,86314 | Piraievs | 37,9344 | 23,6174 |
| Elevsis | 38,0384 | 23,55630 | Pithagorion | 37,6867 | 26,9507 |
| Gavrio | 37,8849 | 24,73322 | Plomarion | 38,9723 | 26,3694 |
| Güllük | 37,2407 | 27,58553 | Porthmos Evripou | 38,4618 | 23,5884 |
| Iraklion | 35,3613 | 25,17378 | Rethimnon | 35,3801 | 24,4851 |
| İzmir | 38,4370 | 27,11123 | Rodhos | 36,4548 | 28,2330 |
| Kalimnos | 36,9487 | 26,99567 | Samos | 37,7545 | 26,9660 |
| Kavala | 40,9211 | 24,40924 | Sitia | 35,2209 | 26,1298 |
| Khios | 38,3699 | 26,14638 | Soudha | 35,5318 | 24,2495 |
| Kuşadası | 37,8679 | 27,22846 | Spetses | 37,2695 | 23,1561 |
| Kymassi | 38,8135 | 23,52061 | Stilis | 38,9091 | 22,6192 |
| Lagos | 40,9497 | 25,06806 | Stratoni | 40,5130 | 23,8329 |
| Lakki | 37,1089 | 26,82484 | Thessaloniki | 40,6234 | 22,9295 |
| Larimna | 38,5731 | 23,29796 | Tsingeli | 39,1689 | 22,8536 |
| Lavrio | 37,7081 | 24,06613 | Volos | 39,3464 | 22,9525 |
| Limin Kos | 36,9136 | 27,30059 | Vrakhonisis Kallonis | 39,1010 | 26,5577 |
| Limin Sirou | 37,4352 | 24,94863 | Yerakini | 40,2643 | 23,4426 |
| Marmaris | 36,7451 | 28,36273 | Çanakkale | 40,0254 | 26,1766 |
| Megara Oil Terminal | 37,9770 | 23,35355 | Ekklisia Agios Georgios | 36,4279 | 23,2045 |
| Mikonos | 37,4499 | 25,3260 | | | |

### 1.2. Düğümler

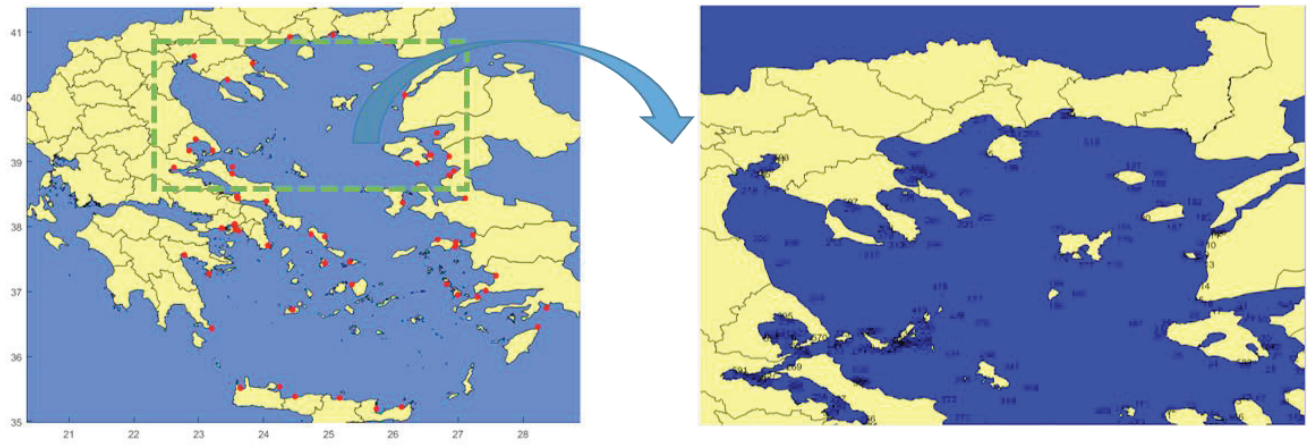
Limanlar arasındaki bağlantılar çoğu liman için direkt düz bir çizgi ile ifade edilemez, çünkü Ege denizi içerisinde birden çok irili ufaklı ada bulundurmaktadır. Düğüm haritası çıkarılarak bu düğümlerin koordinatlarının kayıt altına alınması, düğümlerin arasındaki deniz yolu bağlantılarının hesaplanıp hangi düğümlerin hangi limanlarla bağlantılı olduğunun hesaplanması gerekmektedir. Düğüm haritası çıkarılırken düğümler olası bir seyir durumunda ticari gemilerin rota planlamasında olduğu gibi en kısa yol olacak şekilde kara parçalarının burun kısımları hedef gözetilerek hesaplanmıştır. Elde edilen düğüm haritasının bir kısmı Şekil 1'de görülmektedir.

Çalışmamız, gemiler için dinamik olarak seyir planlaması yapabilen bir zeki sistemin parçasıdır. Rota planlamasının dinamik olarak güncellenebilmesi için zeki sistemin herhangi bir noktadan herhangi başka bir noktaya nasıl gidileceğini düğüm haritası üzerinden hesaplayabilmesi gerekmektedir. Düğüm haritasının çok detaylı olmasının nedeni budur. Düğüm haritasının çıkarılmasındaki asıl ihtiyaç, limanlar arasındaki en kısa yolu hesaplamaktan daha çok zeki sistemin karar değiştirmesi aşamasında tıpkı bir kaptan gibi düşünüp her ihtimali değerlendirip, bu ihtimallerin nasıl gerçekleşebileceği konusunda bilgiyi edinebileceği bir düğüm haritasının elde edilmesi zorunluğudur. Tablo 2'de düğüm numaraları ve düğümlerin koordinatları görülmektedir.

**Tablo 2.** Düğümlerin numaraları ve koordinatları

| Düğüm No. | Enlem (Kuzey) | Boylam (Doğu) | Düğüm No. | Enlem (Kuzey) | Boylam (Doğu) | Düğüm No. | Enlem (Kuzey) | Boylam (Doğu) |
|---|---|---|---|---|---|---|---|---|
| **1** | 39,9409 | 26,0538 | **41** | 38,6861 | 26,5077 | **81** | 37,1723 | 26,7507 |
| **2** | 39,9363 | 26,0932 | **42** | 38,6960 | 26,4051 | **82** | 37,2168 | 26,7514 |
| **3** | 39,8420 | 25,9578 | **43** | 38,6678 | 26,3287 | **83** | 37,0094 | 26,8659 |
| **4** | 39,8508 | 26,0548 | **44** | 38,4535 | 26,3068 | **84** | 36,9405 | 26,8878 |
| **5** | 39,8309 | 26,0848 | **45** | 38,3845 | 26,2639 | **85** | 36,8738 | 26,9104 |
| **6** | 39,7978 | 26,0843 | **46** | 38,3022 | 26,1949 | **86** | 36,7578 | 26,8956 |
| **7** | 39,7861 | 26,0488 | **47** | 38,2135 | 26,1189 | **87** | 36,6567 | 26,9702 |
| **8** | 39,8203 | 25,9662 | **48** | 38,1381 | 26,0220 | **88** | 36,7485 | 27,1243 |
| **9** | 39,8408 | 26,1285 | **49** | 38,1805 | 25,9020 | **89** | 36,8459 | 27,3716 |
| **10** | 39,9152 | 26,1421 | **50** | 38,2457 | 25,8441 | **90** | 36,6329 | 27,1257 |
| **11** | 39,9935 | 26,1701 | **51** | 38,5473 | 25,8081 | **91** | 36,5531 | 27,1253 |
| **12** | 40,0131 | 26,1927 | **52** | 38,5989 | 25,8501 | **92** | 36,5173 | 26,9637 |
| **13** | 39,7605 | 26,1303 | **53** | 38,6254 | 26,0030 | **93** | 36,4895 | 26,9821 |
| **14** | 39,5903 | 26,0940 | **54** | 38,5583 | 26,1699 | **94** | 36,5767 | 27,0537 |
| **15** | 39,4776 | 26,0500 | **55** | 38,4597 | 26,1649 | **95** | 36,6069 | 27,0237 |
| **16** | 39,4491 | 26,1160 | **56** | 38,3179 | 26,1739 | **96** | 36,4986 | 27,2575 |
| **17** | 39,3879 | 26,1849 | **57** | 38,1030 | 26,5418 | **97** | 36,4113 | 27,2749 |
| **18** | 39,3983 | 26,3446 | **58** | 37,9925 | 26,8342 | **98** | 36,3499 | 27,3973 |
| **19** | 39,3366 | 26,4320 | **59** | 37,7880 | 27,0981 | **99** | 36,4141 | 27,4954 |
| **20** | 39,1687 | 26,5522 | **60** | 37,0535 | 27,0719 | **100** | 36,4837 | 27,4019 |
| **21** | 39,0330 | 26,6396 | **61** | 37,6677 | 26,9388 | **101** | 36,5060 | 27,8754 |
| **22** | 38,9238 | 26,5898 | **62** | 37,6093 | 26,8580 | **102** | 36,5691 | 28,1340 |
| **23** | 38,9514 | 26,4051 | **63** | 37,6677 | 26,5679 | **103** | 36,7451 | 28,3627 |
| **24** | 38,9591 | 26,1613 | **64** | 37,7410 | 26,5323 | **104** | 36,4856 | 28,2345 |
| **25** | 39,0274 | 25,8922 | **65** | 37,8275 | 26,6393 | **105** | 36,3109 | 27,6848 |
| **26** | 39,1808 | 25,7473 | **66** | 37,8481 | 26,8176 | **106** | 36,2550 | 27,4873 |
| **27** | 39,2378 | 25,7555 | **67** | 37,3509 | 27,1480 | **107** | 36,2662 | 27,7391 |
| **…** | ... | ... | **...** | ... | ... | **...** | ... | ... |
| **602** | 35,5382 | 24,2472 | **603** | 35,4840 | 24,1780 | **604** | 35,6595 | 24,3384 |

**Şekil 1.** Ege limanları ve düğüm haritasının bir kısmı




Düğüm sayısı çok olduğundan tablo kısa tutulmuştur. Düğümler arasında deniz yolu ile bağlantının olup olmadığı, varsa aralarındaki mesafelerin ne olduğu ve aralarındaki açıların ne olduğu deniz seyrinin gerçekleştirilmesi açısından kritik öneme sahiptir.

### 1.2.1. Düğümler arasındaki açıların hesaplanması

Düğümlerin arasındaki bağlantıların hesaplanabilmesi için düğümlerin arasındaki bağlantının açısının kayıt altına alınması gereklidir. Açılar düğümler arasındaki gerçek deniz bağlantısının olup olmadığı konusunun anlaşılmasında kullanılmıştır ayrıca zeki sistemin rota belirlemesi aşamasında da kullanılacaktır. Açılar matrisi 604 düğümün birbirleri arasındaki açıyı tutacak şekilde oluşturulmuştur. Tablo 3'te düğümlerin arasındaki açı bilgilerini tutan matrisin bir kısmı görülmektedir.

### 1.2.2Düğümler arasındaki mesafelerin hesaplanması

Düğümlerin arasındaki bağlantıların hesaplanabilmesi için düğümlerin arasındaki mesafenin kayıt altına alınması gereklidir. Mesafeler matrisi 604 düğümün birbirleri arasındaki mesafeyi tutacak şekilde oluşturulmuştur. Tablo 4'te düğümler arasındaki mesafeleri kilometre cinsinden gösteren matrisin bir kısmı görülmektedir.

**Tablo 3.** Düğümler arasındaki açıları gösteren matris (açılar derece cinsindendir)

| **Düğüm No** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | … | **604** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 99 | 217 | 180 | 168 | 171 | 181 | 209 | 150 | 111 | … | 198 |
| **2** | 279 | 0 | 228 | 199 | 184 | 183 | 193 | 220 | 164 | 119 | … | 199 |
| **3** | 37 | 48 | 0 | 83 | 96 | 114 | 129 | 163 | 90 | 63 | … | 198 |
| **4** | 360 | 19 | 263 | 0 | 131 | 157 | 184 | 246 | 100 | 46 | … | 198 |
| **5** | 348 | 4 | 277 | 311 | 0 | 181 | 212 | 263 | 73 | 28 | … | 199 |
| **6** | 351 | 3 | 294 | 337 | 1 | 0 | 247 | 284 | 38 | 21 | … | 199 |
| **7** | 1 | 13 | 309 | 4 | 32 | 67 | 0 | 298 | 48 | 29 | … | 199 |
| **8** | 29 | 40 | 343 | 66 | 83 | 104 | 118 | 0 | 81 | 55 | … | 198 |
| **9** | 330 | 344 | 271 | 280 | 253 | 218 | 228 | 261 | 0 | 8 | … | 199 |
| **10** | 291 | 299 | 243 | 226 | 208 | 201 | 209 | 235 | 188 | 0 | … | 199 |
| … | … | … | … | … | … | … | … | … | … | … | 0 | 198 |
| **604** | 17 | 17 | 17 | 17 | 18 | 18 | 18 | 17 | 18 | 18 | … | 0 |

**Tablo 4.** Düğümler arasındaki mesafeleri gösteren matris (mesafeler kilometre cinsindendir)

| Düğüm No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | … | 604 |
|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 3,404 | 13,71 | 10,01 | 12,52 | 16,12 | 17,22 | … | 499,32 |
| **2** | 3,40 | 0 | 15,60 | 10,05 | 11,74 | 15,41 | 17,13 | … | 499,9 |
| **3** | 13,71 | 15,60 | 0 | 8,334 | 10,90 | 11,86 | 9,95 | … | 486,35 |
| **4** | 10,01 | 10,05 | 8,334 | 0 | 3,387 | 6,405 | 7,217 | … | 489,84 |
| **5** | 12,52 | 11,74 | 10,90 | 3,387 | 0 | 3,671 | 5,849 | … | 488,56 |
| **6** | 16,12 | 15,41 | 11,86 | 6,405 | 3,671 | 0 | 3,297 | … | 485,07 |
| **7** | 17,22 | 17,13 | 9,955 | 7,217 | 5,849 | 3,297 | 0 | … | 482,85 |
| … | … | … | … | … | … | … | … | 0 | … |
| **604** | 499,33 | 499,9 | 486,35 | 489,84 | 488,56 | 485,073 | 482,857 | … | 0 |

### 1.2.3. Düğümler arasındaki bağlantıların hesaplanması

Düğümlerin arasındaki bağlantıların hesaplanması aşamasında düğümler arasındaki mesafeler ve açılar hesaplanarak düğümlerin arasındaki yol üzerinde kara parçası olup olmadığı denetlenmiştir. Shape file dosyaları matrise çevrilerek belirtilen noktanın deniz üzerinde mi yoksa kara üzerinde mi olduğunun bulunması mümkündür. Bir düğüm ile diğer düğüm arasındaki mesafe ve açı biliniyorsa belli mesafe aralıklarıyla adım adım ilerleyerek belirtilen noktanın karaya mı yoksa denize mi geldiği 1,2,3,4 numaralı formüller yardımıyla hesaplanabilmektedir. Noktalar arasındaki mesafeler ve açılar kayıt altına alınarak her ikisi için de 604x604 büyüklüğünde iki adet matris oluşturulmuştur. Daha sonra bağlantıları kayıt altına alabilmek için yine 604x604 büyüklüğünde bağlantı matrisi oluşturulmuştur. Matriste bilgiler bağlantı var için 1, bağlantı yok için 0 olacak şekilde kaydedilmiştir.

Adım aralığı hesaplaması şöyle yapılmaktadır:

$$dx = km * \sin(açı(nokta1, nokta2)) \quad (1)$$

$$dy = km * \cos(açı(nokta1, nokta2)) \quad (2)$$

$$delta\ boylam = dx/(11320 * \cos(nokta1.enlem) \quad (3)$$

$$delta\ enlem = dy/110540 \quad (4)$$

dx: Boylam olarak istenilen açıda istenilen kilometre miktarında ilerlemek için gerekli yol miktarı.

dy: Enlem olarak istenilen açıda istenilen kilometre miktarında ilerlemek için gerekli yol miktarı.

km: İlerlemek istenilen yol miktarı (kilometre).

delta boylam= Boylam üzerinde ilerlemek istenilen yol miktarının derece cinsinden değeri.

delta enlem= Enlem üzerinde ilerlemek istenilen yol miktarının derece cinsinden değeri.

Adım aralığı hesaplaması sonrasında koordinat eğer kara parçasına denk geliyorsa bağlantı matrisine 0, tüm adımların bitmesinden sonra hiç kara parçasına denk gelinmediyse bağlantı matrisine 1 kaydedilmektedir.

Tablo5'te yapılan hesaplamalar sonucunda düğümlerin arasındaki bağlantıların var olup olmadığını gösteren matrisin bir kısmı görülmektedir.

**Tablo 5.** Düğümler arasındaki bağlantıyı gösteren matris

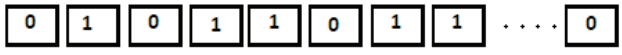
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | … | 604 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **8** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **9** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **10** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **11** | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | … | 0 |
| **…** | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| **604** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

## II. YÖNTEM

Çalışmada yöntem olarak genetik algoritma kullanılmıştır. Genetik algoritma ilk olarak J. Holland tarafından ortaya konulmuştur [16]. Genetik algoritma evrimsel bir algoritmadır yani problemin çözümünü canlıların hayatlarını devam ettirme yöntemlerini kullanarak çözüme ulaştırmaya çalışır.

### 2.1. Kromozom Yapısı

Problemin ortaya konulmasının ardından problemi çözmek için kullanılacak bireyler oluşturulur. Her birey kendi kromozom yapısına sahiptir. Şekil 2'de örnek bir kromozom yapısı görülmektedir.




**Şekil 2.** Genetik algoritmada kullanılan örnek kromozom yapısı

Kromozom yapısı 0 ve 1 lerden de oluşabilir, içinde string ifadeler de olabilir veyahut da bizim çalışmamızda olduğu gibi düğüm numaraları da olabilir. Çalışmamızda kromozom yapısı düğüm numaralarından oluşmaktadır.

### 2.2. Uygunluk Fonksiyonu

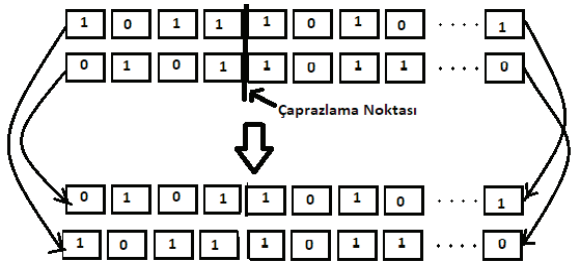
Her birey probleme uygun olarak kendi uygunluğunu kromozomu ile test etmek durumundadır. Uygunluk fonksiyonu probleme göre değişiklik gösterir. Uygunluk fonksiyonundan elde edilen sonuç ışığında bireylerin hayatlarına devam etmeleri veya devam etmemeleri konusunda bir karar elde edilir.

### 2.3. Seçim

Bireyler uygunluk fonksiyonundan elde ettikleri değer sonucunda bir sıralamaya tabi tutulurlar. En uygun değere sahip birey sıralamada en üstte, en kötü değere sahip birey ise sıralamada en altta tutulur. Bireyler arasından yeni nesil üretmek için seçim yapılması gerekir. Bu seçim çeşitli yöntemlerle yapılabilir. Rulet yöntemi, turnuva yöntemi veya sıralama gibi yöntemler seçim için kullanılan yöntemlerdir.

### 2.4. Çaprazlama

Uygunluk fonksiyonlarına göre sıralanan ve seçime giren bireyler kendi aralarında yeni bireyler üretmek amaçlı kullanılırlar bu işlemin ilk adımına çaprazlama denir. Çaprazlama için bireylerin kromozomları belli bir yerden kesilerek diğer bireyin kromozomunun parçası ile birleştirilir. Böylece yeni birey veya bireyler elde edilmiş olur. Birleşme noktası rasgele seçilen bir yer olabilir. Şekil 3'de bir çaprazlama örneği görülmektedir.




**Şekil 3.** Genetik algoritmada çaprazlama örneği

### 2.5. Mutasyon

Oluşturulan yeni bireylerin gen yapısında bazı yerlerin mutasyona uğraması, sonuca rasgele ulaşılabilmesi ve çeşitliliğin sağlanarak çözümün bir noktada takılmaması için önemlidir. Mutasyon bir veya birden çok gene uygulanabilir. Kaç genin mutasyona uğrayacağı problemin yapısına göre değişiklik gösterebilir.

### 2.6. Yeni Jenerasyonların Oluşturulması

Mutasyon sonucu yeni bireyler oluşturulur, yeni bireylerin oluşturulmasında eski bireylerden bazılarının devamı da sağlanabilir ya da tamamen yeni bireyler oluşturularak devam edilmek istenilebilir. Çeşitlilik ne kadar fazla ise o kadar doğru sonuca yaklaşma oranı artar, ancak çeşitliliğin fazla olması, algoritmanın işlemesi bakımından harcanan zaman yüzünden sonuca ulaşılması için harcanan süreyi yükseltir.

## III. GENETİK ALGORİTMANIN UYGULANMASI

### 3.1. Bireylerin Oluşturulması

Ege denizindeki düğüm sayısı bu problemde 604 adettir. Bireylerin kromozomlarındaki genler başlangıç düğümünden bitiş düğümüne kadar olan rasgele seçilmiş tekrar etmeyen düğümlerden oluşmaktadır. Başlangıç ve bitiş düğümleri ilk ve son genler olarak belirlenerek ilk genden başlayarak düğüme bağlı olan düğümlerden bir tanesi rasgele seçilmiş, bu işlem son gen yani bitiş düğümü elde edilene kadar devam ettirilmiştir. Şekil 4'te bireylerin oluşturulması aşaması görülmektedir.

### 3.2. Bireylerin, Uygunluk Fonksiyonlarına Göre Uygunluklarının Hesaplanması

Sayıları ve gen dizilimleri belli olan bireylerin her biri uygunluk fonksiyonuna göre değerlendirilirler. Çalışmadaki

```
bireydizisi tanımla;
  for i=1 to bireysayısı
   birinci geni başlangıç düğümü olarak ata;
   j=1;
        while rasgeledügüm!=bitişdüğümü
        rasgeledüğüm=rasgeledugumseç(1,604);
              if rasgeledüğüm!=başlangıçdüğümü&&rasgeledüğüm!=bitişdüğümü&& rasgeledüğüm
              bağlantılı mı bireydizisi(i,j-1)
        bireydizisi(i,j)=rasgeledügüm;
        j++;
              end if
        end while
  end for
```

**Şekil 4.** Bireylerin oluşturulması.

uygunluk fonksiyonu bireyin başlangıçtan hedefe ulaşmak için kullandığı düğümlerin arasındaki mesafe hesaplanmasına dayanmaktadır. En kısa mesafeyi bulan birey uygunluk fonksiyonuna göre en başarılı birey olarak kabul edilmektedir.

Bireyler en kısa mesafe bulandan en uzun mesafe bulana doğru sıraya konulmaktadır. Eski bireyler ve yeni bireyler aynı anda uygunluk yönünden kıyaslamaya tutulmaktadır, bu yöntemle yeni popülasyonda çeşitlilik hedeflenmekte ve jenerasyonun daha kötüye gitmesi engellenmektedir. Eski bireylerin içinde eğer yeni bireylerden daha uygun olanı varsa o birey hayatına devam etmektedir. Şekil 5'te algoritmada kullanılan uygunluk fonksiyonu görülmektedir.

```
başla;
bireyleri al;
mesafeler matrisini al;
skor=0;
dizi=dizinin 0 dan farklı elemanları;
        for t=1:to dizielemansayısı-1
        skor=skor+mesafeler(dizi(t),dizi(t+1));
        end for
bitir;
```

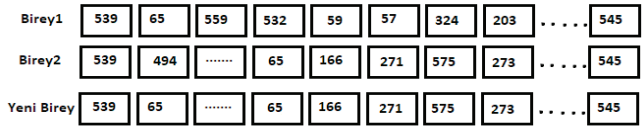
**Şekil 5.** Uygunluk fonksiyonu

### 3.3. Çaprazlama

Bireyler kendi aralarında sıralanma durumlarına göre çaprazlanmıştır. Çaprazlama işleminde bireyin kromozom yapısından rasgele bir geni seçilerek bu gene kadar olan gen dizisi kesilmiştir yani bireyin gen dizisi iki parçaya ayrılmıştır. Çaprazlama işlemine giren diğer bireyin de aynı şekilde rasgele bir geni seçilmiş ve seçilen genden iki parçaya ayrılmıştır. Birinci birey ile ikinci bireyden elde edilen genler çaprazlanarak yeni bireyler üretilmiştir.

### 3.4. Tekrar Eden Genlerin Kaldırılması

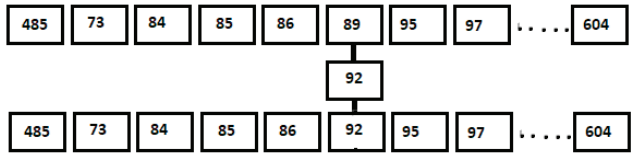
Mutasyon işlemi sonunda elde edilen gendeki düğüm, yerleştiği yerin öncesindeki veya sonrasındaki bir düğümle aynı olabilir. Şekil 6'da buna bir örnek görülmektedir. Yeni bireyde 65. Düğüme iki kere uğranılmaktadır. Çözümün aynı yolu dolaşmaması için tekrar eden düğümler arasındaki genler kesilerek tekrar birleştirilmiştir bu şekilde çözüme daha hızlı ulaşıldığı görülmüştür.




**Şekil 6.** Çaprazlama örneği

### 3.5. Mutasyon

Bireylerin gen dizilerinde çeşitliliği arttırmak ve çözümün bir noktada tıkanmasını engellemek amacıyla bir veya birkaç gen mutasyona uğratılır. Çalışmamızda bu mutasyon, içinde düğüm bilgisi bulunan genin uygun başka bir gen ile değiştirilmesi şeklinde yapılmıştır. Şekil 7'de mutasyon işlemine bir örnek görülmektedir. Mutasyona uğrayan gen yerine gelen genin içindeki düğüm hem kendisinden önceki hem de kendisinden sonraki düğüm ile bağlantılı olmalıdır. Bağlantılı olup olmadıkları daha önce bahsedilen bağlantı matrisi ile denetlenmektedir.



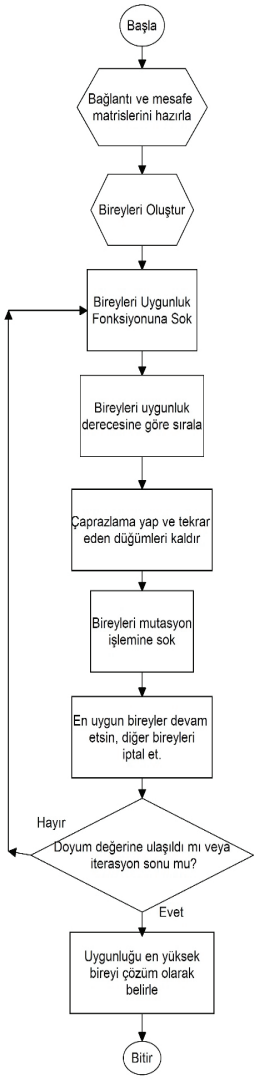

**Şekil 7.** Yeni bireylerin belirlenmesinde mutasyon işlemi

### 3.6. Seçim Yöntemi

Popülasyonun daha başarılı hale gelebilmesi için eski ve yeni bireylerin arasında bir kıyaslama yapılarak popülasyonun her denemede daha başarılı hale getirilmesi gerekir. Ancak popülasyonun hep en iyi bireylerden üretmesi çözümü bir noktada tıkayabilir. Bu sorunun önüne geçilebilmesi için popülasyonun başarılı bireylerden devam etmesi kadar başarısız bireylerin de yeni popülasyonda etkisi olması gerekir. Çalışmamızda bu amaçla "Random Selection" yöntemi kullanılmıştır ancak seçim yapılmadan önce yapılan sıralama ile en başarılı bireylerin seçimde üst sıralarda yer almaları sağlanmıştır.

### 3.7. Algoritma Genel Yapısı

Genetik algoritmanın probleme uygulanması aşamasında hangi adımların gerçekleştiği Şekil 8'de gösterilmiştir.



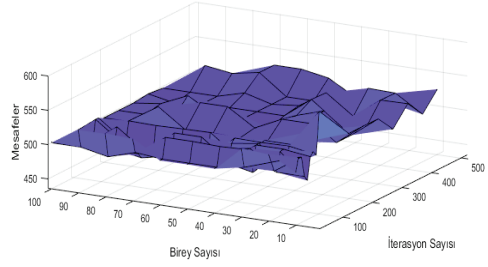

**Şekil 8.** Çalışmada kullanılan genetik algoritma yapısı.

## IV. SİMÜLASYON SONUÇLARI

Genetik algoritmanın hızını ve doğruluğunu etkileyen faktörler değiştirilerek elde edilen sonuçlar kayıt altına alınmıştır. Hıza ve doğruluğa etki eden faktörler birey sayısı ve iterasyon sayısıdır. Genetik algoritmanın sonlanması için bu çalışmada "stagnation" kullanılmıştır. En başarılı değerin 10 kez üst üste bulunması durumunda algoritma sonlanmaktadır. Hesaplamalar Windows işletim sistemi üzerinde MATLAB programı kullanılarak yapılmıştır. Hesaplamaların yapıldığı bilgisayar Intel Pentium i7, 3.6 Ghz hızında işlemciye sahiptir.

### 4.1. Değişken Birey Sayısı ve İterasyon Sayısı Miktarlarının Mesafeler Yönünden Karşılaştırılması

Birey sayısı 10 ile başlayarak 10 artımla 100'e kadar arttırılmış, buna karşılık iterasyon miktarı 10 ile başlayarak 10 artımla 200'e kadar, 200'den sonra ise 50 artımla 500'e kadar arttırılmıştır. Her birey sayısı ve iterasyon sayısı miktarı ile yapılan denemelerde elde edilen mesafe miktarı kilometre cinsinden sonuçlar Şekil 9'da gösterilmiştir.

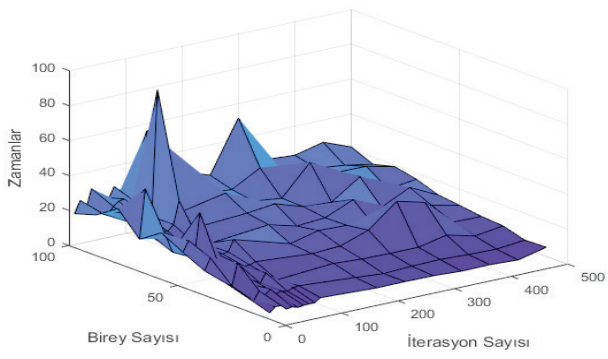



**Şekil 9.** Achladi limanı ile Aya Nikolas limanı arasında çeşitli birey sayısı ve iterasyon sayısı ile elde edilen mesafe miktarları

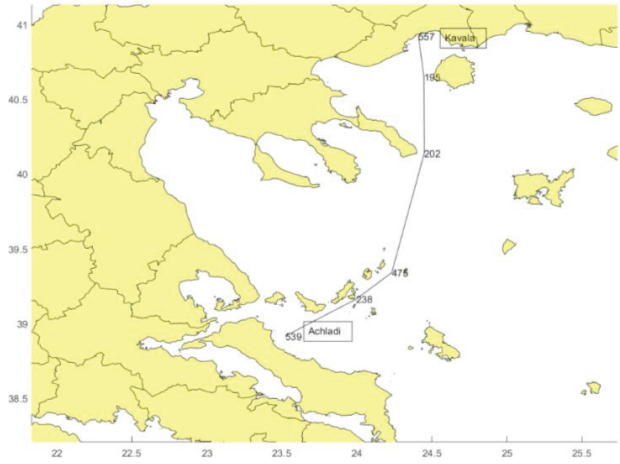
Şekil 9'da görüleceği üzere genetik algoritma birey sayısı ve iterasyon sayısına bağlı kalmaksızın tüm denemelerde aynı veya birbirine çok yaklaşık mesafe sonuçlarına erişmiştir.

### 4.2. Değişken Birey Sayısı ve İterasyon Sayısı Miktarlarının Zaman Yönünden Karşılaştırılması

Birey sayısı 10 ile başlayarak 10 artımla 100'e kadar arttırılmış, buna karşılık iterasyon miktarı 10 ile başlayarak 10 artımla 200'e kadar, 200'den sonra ise 50 artımla 500'e kadar arttırılmıştır. Her birey sayısı ve iterasyon sayısı miktarı ile yapılan denemelerde elde edilen zaman miktarları saniye cinsinden kayıt altına alınmıştır.




**Şekil 10.** Achladi limanı ile Aya Nikolas limanı arasında çeşitli birey sayısı ve iterasyon sayısı ile elde edilen zaman miktarları



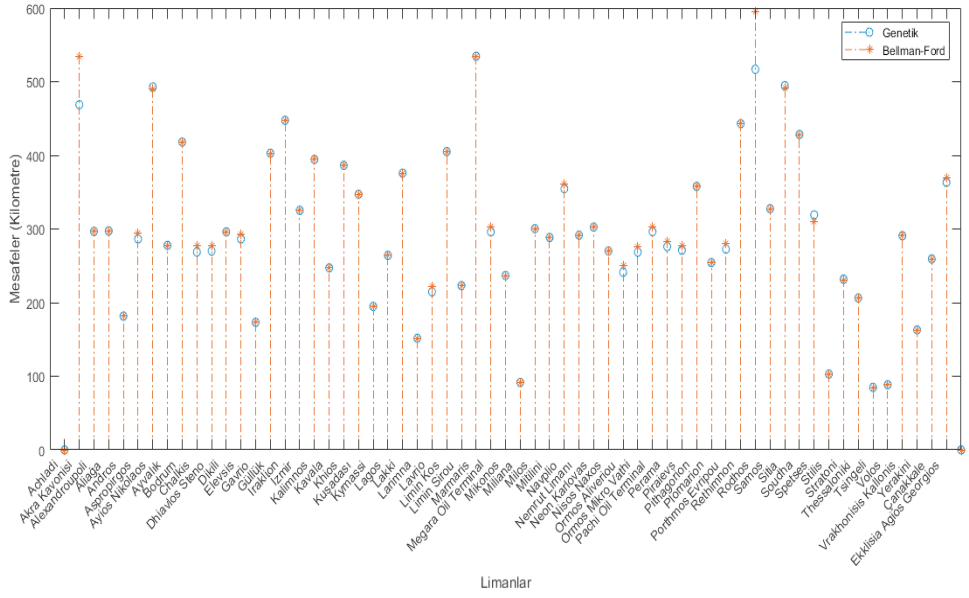

***Şekil 11***. Geliştirilen sistem ile yapılan bir hesaplama örneği

Genetik algoritmanın hesaplamayı sonlandırması için "stagnation" yöntemi kullanıldığından, hesaplamalar için harcanan zamanın incelenmesinde iterasyon sayısının harcanan zaman üzerinde bir etkisi olmadığı lakin birey sayısının az veya çok olmasına bağlı olarak hesaplama zamanının azaldığı veya arttığı Şekil 10'daki grafiğe bakarak gözlemlenmiştir. Genetik algoritma ile Achladi limanından Kavala limanına rota belirlemesi işlemi yapılmıştır. Algoritma 500 iterasyon ve 100 birey ile çalıştırılmıştır. Şekil 11'de bu uygulamaya ilişkin rota görülmektedir. Kavala limanına 539,238,475,202,195,557 numaralı düğümler takip edilerek sonuca 247 km'lik yol ulaşılmıştır.

## V. SONUÇ VE DEĞERLENDİRME

Bu çalışmada Ege denizi üzerinde bulunan Yunanistan ve Türkiye limanları temel alınmıştır. Ege denizi üzerinde bulunan 61 adet limanın arasında yapılabilecek olası seyir durumu için 604 adet düğüm belirlenmiştir. Rota belirlemede olası düğümlerin seçilerek en uygun rotanın bulunması genetik algoritma kullanılarak yapılmıştır.

Genetik algoritmanın değişkenleri olan birey sayısı ve iterasyon sayısının problemin çözümü üzerindeki etkisi incelenmiştir. Her bir liman için bu değişkenlerin en iyi değerleri bulunmuştur. Değişken birey sayısı ve iterasyon sayısı




**Şekil 12.** Genetik algoritma ile elde edilen en kısa mesafe sonuçları ile Bellman-Ford algoritması ile elde edilen en kısa mesafe sonuçlarının karşılaştırılması

miktarlarında yapılan tecrübelerde elde edilen çözümler sonuç bölümünde grafiksel olarak gösterilmiştir. Genetik algoritmanın problemin çözümündeki performansı değerlendirilmiş ve başarılı sonuçlar elde edildiği görülmüştür.

Yapılan literatür araştırmasında Ege denizi üzerinde genetik algoritmanın kullanıldığı bir örneğe rastlanmadığından çalışma bu yönüyle önceki yapılan çalışmalar ile benzerlik göstermemektedir.

En kısa yol sonuçlarının doğruluğunu ölçmek amacıyla düğüm haritası, yönlü grafik (directed graph) haline getirilerek Achladi limanı ile diğer limanlar arasındaki en kısa yollar Bellman-Ford algoritması [17], [18] kullanılarak MATLAB programı ile hesaplatılmış, elde edilen değerler ortaya konulan genetik algoritmanın elde ettiği değerler ile karşılaştırılmıştır. Sonuçlar Şekil 12'de görülmektedir. Ortaya konulan genetik algoritma Bellman-Ford algoritması ile aynı sonuçları bulmuştur.

Problemi çözmedeki hızlarına bakılınca ortaya konulan genetik algoritmanın Bellman-Ford algoritmasına göre daha yavaş olduğu gözlemlenmiştir. Ortaya konulmak istenen sistemin amacına bakıldığında algoritmanın çok hızlı sonuç vermesi zaten beklenmemektedir.

Çalışma, aslında seyir planlanması yapabilen zeki sistemin oluşturulması aşamasında kullanılması planlanan optimizasyon algoritmalarından biri olan genetik algoritmayı problemin çözümünde gösterdiği performans açısından değerlendiren bir çalışma olarak ortaya konulmuş, elde edilen sonuçlar ışığında ise genetik algoritmanın zeki sistemde kullanılabilirliğinin yüksek olduğu değerlendirilmiştir.

## Kaynakça


**[1]** K. S. T. a. S. K. K. F. Man, «Genetic Algorithms: Concepts and Applications,» *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* cilt 43, no. 5, pp. 519-534, 1996

**[2]** H. A. Y., «A genetic algorithm for finding the k shortest paths,» *Egyptian Informatics Journal,* pp. 75-79, 2010.

**[3]** K. M. Kumar R., «Exploring Genetic Algorithm for Shortest Path,» *Global Journal of Computer Science and Technology,* cilt 10, no. 11, pp. 8-12, 2010.

**[4]** «The ant colony system: optimization for the logistics of marine cargo in the Aegean,» *Journal of Statistics and Management Systems,* cilt 8, no. 1, pp. 1-11, 2005

**[5]** «Multi-criteria ACO-based Algorithm for Ship's Trajectory Planning,» *The International Journal on Marine Navigation and Safety of Sea Transportation ,* cilt 11, no. 1, pp. 31-36, 2017.

**[6]** «Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques,» *THE JOURNAL OF NAVIGATION ,* cilt 61, pp. 129-142, 2008.

**[7]** «World Port Index,» Erişim Tarihi: 10 12 2016. [Çevrimiçi]. https://msi.nga.mil/NGAPortal/MSI.portal?_nfpb=true&_pageLabel=msi_portal_page_62&pubCode=0015.

**[8]** «Ports.com,» Erişim Tarihi: 12 12 2016. [Çevrimiçi]. http://ports.com/sea/aegean-sea/.

**[9]** «http://www.worldportsource.com,» 13 12 2016. [Çevrimiçi]. http://www.worldportsource.com/waterways/Aegean_Sea_8.php.

**[10]** «Pireaus Port Authority S.A.,»Erişim Tarihi: 2 1 2017. [Çevrimiçi].http://www.olp.gr/en. [Erişildi: 2017].

**[11]** «PORT AUTHORITY S.A.,» Erişim Tarihi: 2 1 2017. [Çevrimiçi]. http://www.ola-sa.gr/en-us/home.aspx.

**[12]** Matlab, *R 2017a, Mapping Toolbox,* Natick, Massachusetts: Mathworks Inc., 2017.

**[13]** «GEBCO2014_22.2269_34.7694_27.9551_41.517_30Sec_ESRIASCII,» 2016.

**[14]** «QGis,» QGis Project, [Çevrimiçi]. http://www.qgis.org/tr/site/. [Erişildi: 25 12 2016].

**[15]** «British Ocenographic Data Centre,» Erişim Tarihi: 20 12 2016. [Çevrimiçi]. https://www.bodc.ac.uk/my_account/get_basket/E2DH5DEF85D870C4E008G86129F5G512/Skey-GEBCONETE/.

**[16]** J. Holland, Adaptation in Natural and Artificial Systems, Massachussetts: MIT Press, 1992.

**[17]** R. Bellman, «On a routing problem,» *Quarterly of Applied Mathe-matics,* pp. vol. 16, no. 1, pp. 87–90, 1958.

**[18]** L. R. Ford, «Network flow theory,» *DEFENSE DOCUMENTATION CENTER FOR SCIENTIFIC AND TECHNICAL INFORMATION,* p. 923, 1956.