GENETİK ALGORİTMA VE PİKSELİZASYON YÖNTEMİ İLE MAYIN TARLASI OYUNUNUN ZORLUK SEVİYESİNİ BELİRLEME

***

DETERMINING THE DIFFICULTY LEVEL OF THE MINEFIELD GAME WITH GENETIC ALGORITHM AND PIXELIZATION METHOD

Cemal AKTÜRK*

Öz

Genetik algoritmalar, mühendislik, işletme vb alanlardaki sayısal problemlerin çözümünde arama algoritması olarak kullanılan etkili bir optimizasyon yöntemidir. Darwin'in evrim teorisine, yani içinde bulunulan koşullara daha iyi uyum sağlayan bireylerin hayatta kalması prensibine dayanır. Bireyleri oluşturan popülasyon,genetik algoritma operatörleri uygulanarak değişikliğe uğrar ve bunun sonucunda yeni popülasyonlar oluşur. Genetik algoritma kullanımında amaç, aranılan koşullara daha fazla uyum gösterecek bireylere ulaşmaktır. Yapılan çalışmada genetik algoritma ile pikselizasyon yöntemi birlikte kullanılarak mayın tarlası oyununun zorluk seviyesinin istenilen seviyede arttırılıp azaltılabileceği gösterilmiştir. Ayrıca genetik algoritma yönteminin gezgin satıcı, rota belirleme, iş akışı dengeleme gibi problemler dışında oyun tasarımında da kullanılabileceği gösterilerek literatüre katkı sağlamak amaçlanmıştır.

***Anahtar Kelimeler**: Genetik Algoritma, Mayın Tarlası, Oyun Tasarımı.*

Abstract

Genetic algorithms are an effective optimization method which is used as search algorithm for solving numerical problems in engineering, operation etc. fields. Darwin's theory of evolution is based on the principle of survival of individuals who better adhere to the circumstances in which they exist. The population of individuals is modified by applying genetic algorithm operators and as a result new populations are formed. The aim of using genetic algorithms is to reach individuals who will be more adaptable to the desired conditions. In the study, it was shown that the difficulty level of minefield game can be increased or decreased at the desired level by using genetic algorithm and pixelation method together. In addition, it is aimed to contribute to the literature by showing that genetic algorithm method can be used in game design besides traveling seller, route determination and work flow balancing.

***Keywords:** Genetic Algorithm, Minefield, Game Design.*

* Öğr.Gör. Kilis 7 Aralık Üniversitesi, TBMYO, Bilgisayar Teknolojileri Bölümü, cakturk@kilis.edu.tr
ORCID: 0000-0003-3764-3862

Geliş Tarihi / Received: 24.12.2018
Kabul Tarihi / Accepted: 27.12.2018

1. GİRİŞ

Genetik algoritma, karmaşık problemlerin çözümünde oldukça işlevsel olan bir yapay zekâ yöntemidir. Genetik algoritmanın temeli Darwin'in Evrim Teorisi'ne dayanmaktadır. Darwin'in teorisine göre her zaman en iyi uyumu gösterenler hayatta kalmaktadır. Buna doğal seçim de denmektedir (Darwin, 1859). Bu teoriye göre bir canlı topluluğunda daha iyi olanın hayatta kalma şansı daha yüksektir. Bu sebeple iyiler ve genetik özelliklerini kısmen ya da tamamen onlardan alan yavrular daha yüksek ihtimalle yeni nesiller oluşturacak ve genetik bilgilerini yeni nesillere aktarma şansı bulacaklardır (Arslanoğlu, 2006). Canlılarda süregelen bu genetik süreç 1975 yılında Holland tarafından bilgisayar ortamında denenmiştir (Holland, 1975). 1989 yılında ise Holland'ın öğrencisi Goldberg genetik algoritmanın pratikte kullanılabileceğini kanıtlayarak çalışmalarını bir kitap halinde yayınlamıştır (Goldberg, 1989).

Genetik algoritma kullanarak çözüm aranan problemin kendi doğasından kaynaklanan herhangi bir bilgiye ihtiyaç duyulmaz sadece problemin parametre değerlerine ve bu parametrelerin uygunluğunu kıyaslayabilmek için bir kodlama sistemine ihtiyaç duyulur (Daban ve Özdemir, 2004). Çünkü genetik algoritma geleneksel optimizasyon yöntemlerinden farklı olarak problemin parametre kümesini değil bu parametrelerin kodlanmış biçimlerini kullanır. Birçok karmaşık yapıdaki problemlere kolayca adapte edilebilir bir yöntemdir. Çünkü doğrusal yapıdaki algoritmalardaki kısıtlamaları yoktur. Problem ve parametreleri aranan uyum fonksiyonuna göre doğru bir şekilde kodlanabildiği taktirde, genetik algoritma diğer arama ve optimizasyon yöntemlerinden daha etkin kullanılır.

Genetik algoritmalar, parametrelerin kodlanması, başlangıç popülâsyonunun oluşturulması, çoğalma, uygunluk değerlerinin hesaplanması, seçim, çaprazlama ve mutasyon işlemlerinin gerçekleştirilmesinden meydana gelir (Emel ve Taşkın, 2002).

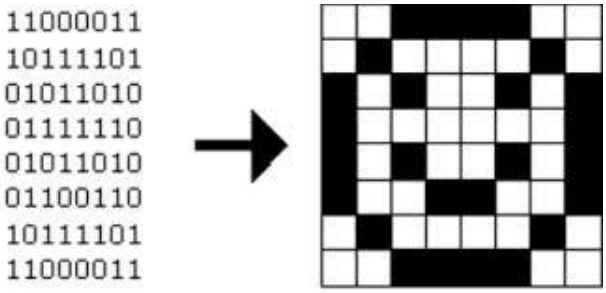
2. YÖNTEM

Çalışmada Microsoft Visua Studio ortamında C#.net platformu üzerinde genetik algoritma ve pikselizasyon yöntemleri kullanılarak mayın tarlası oyununun zorluk seviyesinin arttırılması için bir uygulama yazılımı geliştirilmiştir.

2.1. Kodlama ve Pikselizasyon Yöntemi

Bir problemin genetik algoritma ile çözümünün sağlanabilmesi için, problemin ilgili parametrelerinin algoritma içerisinde kullanılabilir bir bilgiye kodlanması gerekmektedir. Bu bilgi de genelde ikili sayı sistemini oluşturan 1 ve 0 şeklindedir. Kodlama aşamasında arama uzayındaki tüm mümkün çözümler bir dizi olarak kodlanır. Her bir olası çözüm bir bireyi oluşturur ve arama uzayı içerisindeki rastgele oluşturulan bireyler genetik algoritmanın kodlanmış ham verisi olan popülasyonunu oluşturmaktadır. Bir popülasyondaki her birey, iyileştirilmesi planlanan parametreleri içeren kodlanmış bilgiye sahip olmalıdır. Bu bilgilere genetik kavram olarak kromozom denmektedir. Kromozomların uzunluğu arama uzayını tanımlar ve karmaşık yapıların çözümü daha uzun kromozom bilgilerini gerektirir.

Çalışmada tasarlanacak mayın tarlası için zorlaştırılacak veya kolaylaştırılacak oyunun optimizasyonunun yapılabilmesi için mayınlı ve boş hücrelerin ikili sayı sistemi ile kodlanması gerekmektedir. Bu çalışmada 40 satır ve 25 sütundan oluşan piksel hücreleri tasarlanmıştır. Piksel değerlerine hazırlık aşamasında rastgele 1 ve 0 değerleri atanır. Mayınlı hücreler 1, boş hücreler ise 0 olarak ifade edilir. Oluşan yapıda toplam 1000 adet birim hücre vardır. Bu hücrelerin değerlerini saklamak için 40x25 elemanlı çok boyutlu dizi kullanılmıştır. Piksellerin yapısı ve değerleri ile ilgili örnek bir yapı Şekil 1'de gösterilmektedir. Şekil 1 incelendiğinde; beyaz pikseller (mayınlı hücre) 1, siyah pikseller (boş hücre) 0 değerine karşılık gelmektedir. Bu pikseller artık dijital olarak adreslenmiştir.



Şekil 1. Piksellerin İkilik Sistemde Haritalanması (Diaz ve Sigmund, 2010)

2.2. Mayın Tarlası Oyunu

İlk örnekleri 1960'lı yıllarda ortaya çıkmaya başlayan mayın tarlası oyununun günümüzde kullanılan versiyonu 1983 yılında yaygınlaşmıştır. Oyunun amacı mayına denk gelmeyecek şekilde olası tüm boş hücreleri tıklayarak ilgili seviyeyi tamamlamaktır (Tuncer ve ark., 2016). Şekil 2'de oyunun ekran görüntüsü verilmiştir. Oyunun zorluk seviyesini arttırmak için 1 ve 0 olarak tanımlanan piksel hücrelerindeki 1'lerin sayısını arttırmak gerekmektedir ve tıklanan mayın olmayan bir hücre ise bu hücre içerisinde etrafındaki komşu hücrelerdeki mayınların sayısı gösterilir. Yapılan çalışmadaki amaç, bir mayın tarlası oyunu geliştirme çalışması olmayıp, ilgili oyunun matematiksel mantığından yola çıkılarak tasarımına etki edecek zorluk seviyelerini hesaplayan bir genetik algoritması uygulaması geliştirmektir.

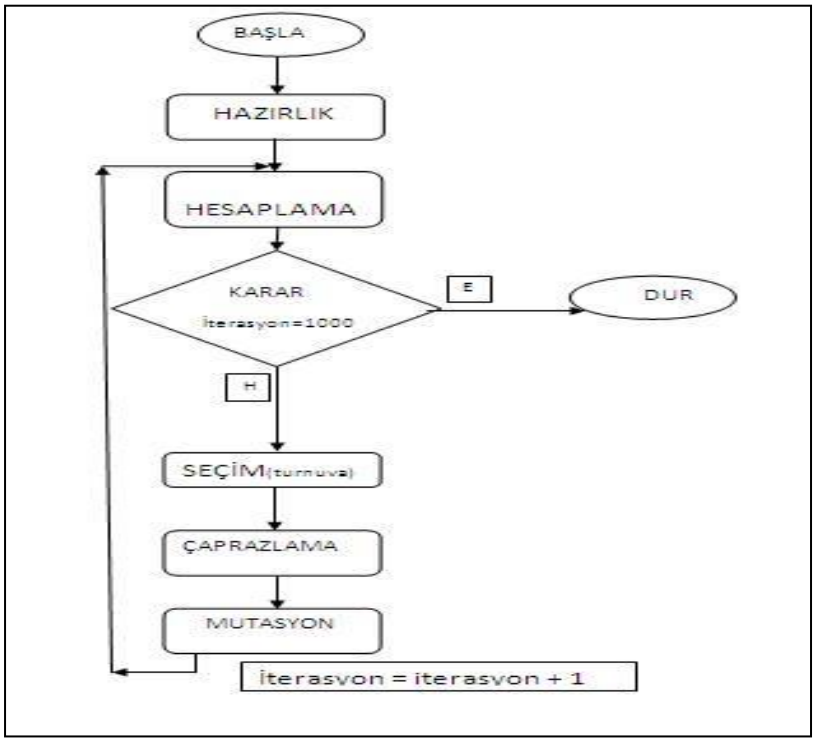
Şekil 2. Mayın Tarlası Oyunu Ekran Görüntüsü (www.rekoroyun.com)

2.3. Genetik Algoritma

Mayın tarlası oyununun zorluk seviyesi genetik algoritma kullanılarak arttırılmaktadır. Bu yöntemin seçilmesindeki amaç doğrusal veya doğrusal olmayan bir çok problemin çözümünde oldukça etkili bir yöntem olmasıdır (Altıparmak ve ark., 2006). Tasarlanacak mayın tarlası oyununun matematiksel ifadesi karmaşık olmadığından dolayı piksel yapısı ve karşılık gelen değerler genetik operasyonlara uğrayacak verileri sağlamaktadır. Piksel yapısındaki satır sayısı olarak kullanılan birey sayısı 40'tır. Her satır 25 hücreden oluştuğu için bireylerin kromozomları 25 gen uzunluğundadır. Çaprazlama işlemine tabi tutulacak bireyler turnuva yöntemi ile seçilmiştir (Ouedraogo, 2011). 40 satırlı yapıdan rastgele dört tanesi seçilir. Seçilen elemanlar kendi içerisinde 1 sayılarına göre karşılaştırılır. İki karşılaştırmanın kazananları ise çaprazlama işlemine tabi tutulur. Çaprazlama işlemi, 1 ile 24 arasındaki bir bitin rastgele seçilerek tek noktadan yapılmasıyla

uygulanır. Tablo 1'de çaprazlama işlemine örnek olarak seçilen nokta ve Tablo 2'de çaprazlama işlemi sonrası oluşturulan yavru kromozomların durumu gösterilmiştir. Bu çaprazlama işlemi için yalnız bir tane ayrım noktası kullanıldığından tek noktalı çaprazlama olarak isimlendirilmektedir (Arslanoğlu, 2006).

Ardından mutasyon işlemi için 40 bireyden 0.05 oranla rastgele seçim yapılır. Seçilen bireyin rassal bir biti mutasyona tabi tutularak 1 ise 0, 0 ise 1 yapılır. Mutasyon işlemi yapıldıktan sonra yavru bireydeki değişim Tablo 3'te gösterilmiştir. Bu işlemlerden sonra Şekil 3'te gösterilen algoritma yapısındaki gibi yeniden uygunluk değerleri kontrol edilir ve döngüye devam edilir. Döngünün iterasyon sayısı 1000 olduğu zaman durdurma kriteri koşulu sağlandığından işlemler tamamlanır. İyileştirmenin amaç fonksiyonu yani uygunluk kriteri olarak mayın tarlası piksel tasarımının her satırındaki 1'lerin sayısı kullanılmıştır. Oyunu zorlaştırmak için 1 sayısı fazla olanlar en uygun değere sahip kabul edilerek her satırdaki 1 değerlerinin sayısını arttırmak amaçlanır. Tersi istenirse oyunu kolaylaştırmak için 0'ların sayısını arttırmak amaçlanabilir.



Şekil 3. Genetik Algoritma Akış Diyagramı

Tablo 1. Çaprazlama işleminin uygulandığı kromozomlar

Kromozom1	1	1	0	1	1	0	1	0
Kromozom2	0	0	1	1	0	1	0	0

Tablo 2. Çaprazlama işleminden sonra oluşan yeni yavru kromozomlar

Yavru1	0	0	1	1	1	0	1	0
Yavru2	1	1	0	1	0	1	0	0

Tablo 3. Mutasyon işlemi sonrasında oluşan yeni kromozom

Yavru1	0	0	1	1	1	0	1	0
Mutasyondan Sonra	0	0	1	0	1	0	1	0

Başlangıç popülasyonunu oluştururken ve genetik operasyonlardan sonra yapılan uygunluk değeri hesaplamasının C# kodları Şekil 4'te gösterilmektedir. Şekil 4'teki kodlar incelenecek olursa 40 satır ve 25 sütundan oluşan tabloda her bir satırdaki 1 (mayınlı hücre) sayısı hesaplanır ve her satırın son bitine ilgili satırdaki toplam 1 sayısı, genetik algoritmanın yapacağı iyileştirmeyi takip etmek için kaydedilir.

```
for (int i = 0; i < 40; i++)
{
satir = "";
sayac = 0;
for (int j = 0; j < 25; j++)
{
if (stDizi[i, j] == 1)
{
sayac++;
}
}
stDizi[i, 25] = sayac;
}
```

Şekil 4. Uygunluk Değerinin Hesaplanarak 1'lerin Sayısının Bulunması

Şekil 5'te turnuva seçim operatörünün uygulanması için geliştirilen C# kodları gösterilmektedir. Öncelikle 40 satır içerisinden rastgele dört eleman seçilir ve seçilen bireyler ikişerli olarak uygunluk kriterlerine (1 sayısına) göre karşılaştırılır. Her karşılaştırmadan galip gelen iki birey sonraki işlem olan çaprazlama operatörüne tabi tutulacaktır.

```
int[,] secilenler = new int[4, 2];
int[,] kazananlar = new int[2, 2];
for (int i = 0; i < 4; i++)
{
rturnuva = r.Next(0, 40);
secilenler[i, 0] = rturnuva;
secilenler[i, 1] = stDizi[rturnuva, 25];
}
if (secilenler[0, 1] > secilenler[1, 1])  //1. karşılaştırma
{
kazananlar[0, 0] = secilenler[0, 0]; // kazananın indis değeri
kazananlar[0, 1] = secilenler[0, 1]; // kazananın 1 sayısı
}
else
{
kazananlar[0, 0] = secilenler[1, 0]; // kazananın indis değeri
kazananlar[0, 1] = secilenler[1, 1]; // kazananın 1 sayısı
}
if (secilenler[2, 1] > secilenler[3, 1])  //1. karşılaştırma
{
kazananlar[1, 0] = secilenler[2, 0]; // kazananın indis değeri
kazananlar[1, 1] = secilenler[2, 1]; // kazananın 1 sayısı
}
else
```

```
{
kazananlar[1, 0] = secilenler[3, 0]; // kazananın indis değeri
kazananlar[1, 1] = secilenler[3, 1]; // kazananın 1 sayısı
}
```

Şekil 5. Turnuva Seçim İşlemi ve Kazananların Belirlenmesi

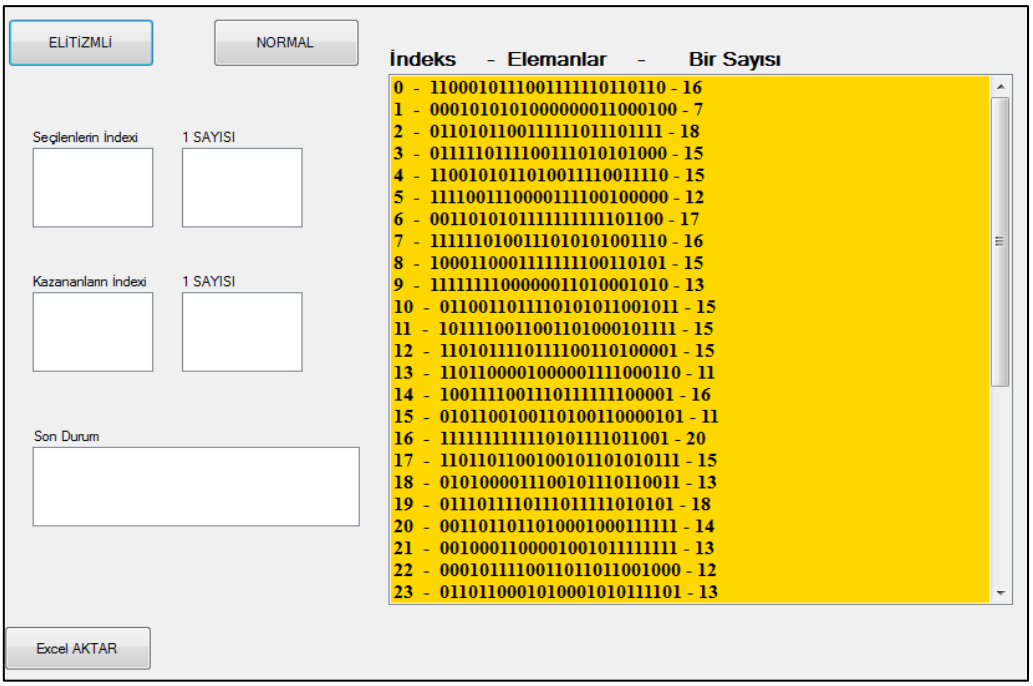
Şekil 6'da turnuvadan galip gelen bireyler için yapılan çaprazlama operatörünün ve sonrasında rastgele seçilen bir bireye ait yapılan mutasyon işlemlerinin C# kodları gösterilmektedir. Çaprazlama işlemi Şekil 6'dan incelendiğinde; öncelikle 1 ile 25 arasında rastgele bir çaprazlama noktası belirlenir. Daha sonra turnuvadan galip gelen ve çaprazlama işlemine tabi tutulan iki bireyin belirlenen çaprazlama noktasına kadar olan bit değerleri karşılıklı yer değiştirilerek çaprazlama işlemi tamamlanmış olur. Mutasyon işlemi için de 0 ile 40 arasında rastgele bir sayı oluşturularak öncelikle mutasyona tabi tutulacak birey belirlenir. Daha sonra kaçıncı bitin mutasyon işlemine tabi tutulacağını belirlemek için 0 ile 25 arasında tekrar rastgele sayı üretilerek bu bitin değeri 1 ise 0, 0 ise 1 yapılarak mutasyon işlemi tamamlanmış ve genetik çeşitlilik sağlanmış olur. Genetik algoritma operasyonları bitince tekrar tüm bireyler için Şekil 3'te gösterildiği gibi uygunluk kriteri hesaplanır ve durdurma kriteri olarak belirtilen iterasyon sayısı veya koşulun sağlanıp sağlanmadığı kontrol edilir. Eğer koşul sağlanmamışsa işlemlere tekrar döngü içerisinde devam edilir.

```
//ÇAPRAZLAMA
rcapraz = r.Next(1, 25);
for (int i = 0; i <= rcapraz; i++)
{
temp1 = stDizi[kazananlar[0, 0], i];
temp2 = stDizi[kazananlar[1, 0], i];
stDizi[kazananlar[0, 0], i] = temp2;
stDizi[kazananlar[1, 0], i] = temp1;
}

//MUTASYON
int rmutsecim = 0;
rmutsecim = r.Next(0, 40);
rmut = r.Next(0, 25);
if (stDizi[rmutsecim, rmut] == 1)
stDizi[rmutsecim, rmut] = 0;
else if (stDizi[rmutsecim, rmut] == 0)
stDizi[rmutsecim, rmut] = 1;
```

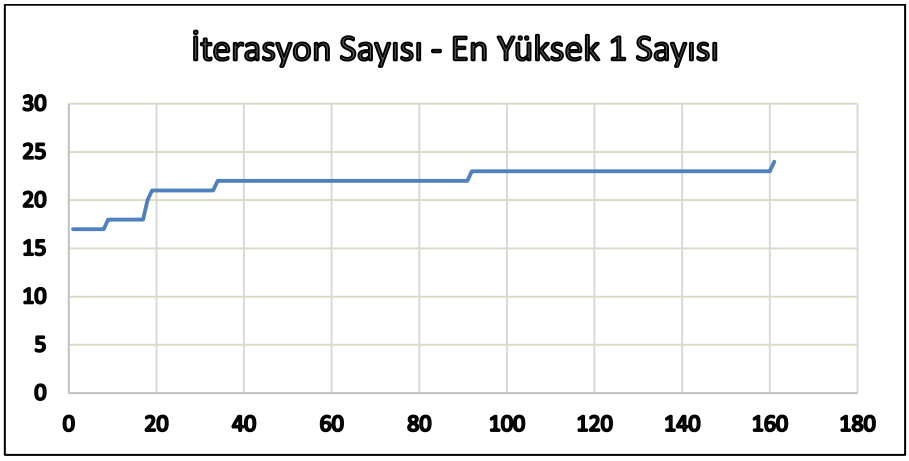
Şekil 6. Çaprazlama ve Mutasyon İşlemleri

Uygulama yazılımının ekran görüntüsü Şekil 7'de gösterilmektedir. Çalıştırılan örnek bir uygulama için genetik algoritma operasyonları sonrasında bireylerin son durumu sarı renkli listbox nesnesinde indis değeri- eleman değeri – bir sayısı şeklinde gösterilmektedir. Ayrıca iyileştirmeyi takip etmek için her iterasyon sonucunda en iyi uygunluk değerine sahip olan bireyin 1 sayısı da kayıt altına alınarak excel ortamına aktarılabilmektedir.



Şekil 7. Uygulama Yazılımı Ekran Görüntüsü

Şekil 8'de örnek olarak çalıştırılan uygulamadan elde edilen, her iterasyon sonrasındaki en iyi uygunluk değerine sahip bireyin 1 sayılarının genetik algoritma işleminde iterasyona göre değişimi grafik olarak gösterilmiştir. Grafikten anlaşılacağı üzere başlangıç durumunda en iyi uygunluk değerine sahip bireyin 17 adet 1 biti bulunmaktadır. 160 iterasyonun sonucunda birlerin sayısında iyileştirmeler yapılmış olup en iyi uygunluk değerine sahip olan bireyin 24 adet 1 biti bulunmaktadır.



Şekil 8. Örnek Bir Uygulamadaki Genetik Algoritma İyileştirmesi

3. SONUÇ

Genetik algoritma; içerisindeki seçim, çaprazlama ve mutasyon gibi operatörler sayesinde belirli bir veri grubunda (bireylerde) çeşitlilikler sağlamaktadır. Bu çeşitlilikler, daha iyi olanın hayatta kalma prensibine göre uygulandığında belirli bir koşulu sağlayanların hayatta kalacağı, diğerlerinin yok olacağı düşünülecek olursa etkili bir optimizasyon aracı olarak genetik algoritmanın tercih edilmesine imkan vermektedir.

Yapılan çalışmada genetik algoritma operasyonlarından kısaca bahsedilmiş olup, sonraki çalışmalara yol göstermesi amacıyla geliştirilen uygulamadaki genetik algoritma operatörlerinin kodları paylaşılmıştır. Ayrıca literatürde daha çok gezgin satıcı problemi (İlkuçar ve Çetinkaya, 2018), (Çakır ve Yılmaz, 2015), (Ertuğrul ve Özçil, 2016); çizelgeleme problemleri (Calp ve Akcayol, 2018), (Doğan ve Takçı, 2015); siber güvenlik (Özgür ve Erdem, 2018), elektromanyetik malzeme tasarımları (Çor ve Saka, 2018), (Alaybeyoğlu ve Ugranlı, 2018), (Aktürk ve ark., 2015), gibi çeşitli sayısal problemlerin çözümünde kullanılan genetik algoritmanın, matematiksel olarak ifade edilebilen oyunların senaryosunun tasarlanmasında da kullanılabileceği mayın tarlası örneği ile gösterilmiştir.

KAYNAKÇA

Aktürk, C., Karaaslan, M., Ozdemir, E., Ozkaner, V., Dincer, F., Bakir, M., & Ozer, Z. (2015). Chiral Metamaterial Design Using Optimized Pixelated İnclusions With Genetic Algorithm. Optical Engineering, 54(3), 035106.

Alaybeyoğlu, E., & Ugranlı, F. (2018, May). A New Approach For Electronic Design Automation Of Analog Building Blocks. In 2018 26th Signal Processing And Communications Applications Conference (SIU) (Pp. 1-4). IEEE.

Altiparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A Genetic Algorithm Approach For Multi-Objective Optimization Of Supply Chain Networks. Computers & İndustrial Engineering, 51(1), 196-215.

Arslanoğlu, Y., 2006. Genetic Algorithm For Personnel Assignment Problem With Multiple Objectives. Doctoral Dissertation, Mıddle East Technıcal University, Ankara.

Calp, M. H., & Akcayol, M. A. (2018). Optimization Of Project Scheduling Activities İn Dynamic CPM And PERT Networks Using Genetic Algorithms. Süleyman Demirel University Journal Of Natural And Applied Sciences (SDU J Nat Appl Sci), 22(2), 615-627.

Çakır, M., & Yılmaz, G. (2015, May). Traveling Salesman Problem Optimization With Parallel Genetic Algorithm. In Signal Processing And Communications Applications Conference (SIU), 2015 23th (Pp. 2557-2560). IEEE.

Çor, İ., & Saka, B. (2018, May). Analysis And Optimization Of Wideband Radomes. In 2018 26th Signal Processing And Communications Applications Conference (SIU) (Pp. 1-4). IEEE.

Daban, F.,& Ozdemir, E.. 2004. Eğitimde Verimliliği Artıran Ders Programlarının Hazırlanması İçin Genetik Algoritma Kullanımı. Journal Of Educational Sciences & Practices, 3(6).

Darwin, C. (1859). On The Origins Of Species By Means Of Natural Selection. London: Murray, 247, 1859.

Diaz, A. R., & Sigmund, O. (2010). A Topology Optimization Method For Design Of Negative Permeability Metamaterials. Structural And Multidisciplinary Optimization, 41(2), 163-177.

Doğan, N. Ö., & Takçı, E. (2015). Bir Tekstil İşletmesinde Simülasyon Yardımıyla Süreç İyileştirme. Ege Academic Review, 15(2).

Emel, G. G.,& Taşkın, Ç.. 2002. Genetik Algoritmalar Ve Uygulama Alanları. Uludağ Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi, Cilt XXI, Sayı 1, 2002, S. 129-152.

Ertuğrul, İ., & Özçil, A. (2016). Siyasi Parti Mitinglerinin Gezgin Satıcı Problemi Yaklaşımı İle Analizi. Siyaset, Ekonomi Ve Yönetim Araştırmaları Dergisi, 4(4).

Goldberg, D. E.. 1989. Genetic Algorithms İn Search, Optimization, And Machine Learning Reading Menlo Park: Addison-Wesley, Vol. 412.

Holland, J. H.. 1975. Adaptation İn Natural And Artificial Systems: An İntroductory Analysis With Applications To Biology, Control, And Artificial İntelligence. U Michigan Press.

Ilkucar, M., & Cetinkaya, A. (2018). Optımızatıon Of Local Travellıng Route Supported Wıth Mobıle Phone And Google Maps: Case Study Of Burdur. Journal Of Mehmet Akıf Ersoy Unıversıty Economıcs And Admınıstratıve Scıences Faculty, 5(1), 64-74.

Ouedraogo, R. O. (2011). Topology Optimization Of Metamaterials And Applications To RF Component Design. Doctoral Thesis. Michigan State University. Electrical Engineering.

Özgür, A., & Erdem, H. (2018). Saldırı Tespit Sistemlerinde Genetik Algoritma Kullanarak Nitelik Seçimi Ve Çoklu Sınıflandırıcı Füzyonu. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 33(1).

Tuncer, T., Avcı, D., & Avcı, E. (2016). İkili İmgeler İçin Mayın Tarlası Oyunu Tabanlı Yeni Bir Veri Gizleme Algoritması. Journal Of The Faculty Of Engineering & Architecture Of Gazi University, 31(4).