



OKAN ÜNİVERSİTESİ
MÜHENDİSLİK – MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME PROJESİ – I

RASTER GÖRÜNTÜLERİN VEKTÖRİZASYONUN'DA
GENETİK ALGORİTMA YAKLAŞIMI

Tez Danışmanı: Yrd. Doç. Dr. Erchan APTOULA

060202003, Taner EŞME

İstanbul, 2010

İÇİNDEKİLER

KISALTMA LİSTESİ	III
ŞEKİL LİSTESİ.....	IV
TABLO LİSTESİ.....	V
ÖNSÖZ	VI
ÖZET	VII
ABSTRACT.....	VIII
1 GİRİŞ.....	1
2 VEKTÖRİZASYON	2
2.1. VEKTÖR GÖRÜNTÜLERİN AVANTAJLARI	4
3 GENETİK ALGORİTMALAR (GA)	6
3.1. BİYOLOJİK ALT YAPI.....	7
3.2. BASİT GENETİK ALGORİTMA	8
3.3. GA BİLEŞENLERİ.....	8
3.3.1. Kodlama Tekniği (Encoding)	8
3.3.2. Başlangıç Nüfusunun Oluşturulması (Initial Population).....	10
3.3.3. Uygunluk Değerinin Hesaplanması (Fitness Function).....	10
3.3.4. Seçme İşleminin Yapılması (Selection).....	11
3.3.5. Çaprazlama Operatörünün Uygulanması (Crossover)	12
3.3.6. Mutasyon Operatörünün Uygulanması (Mutation).....	15
3.4. GA PARAMETRELERİ.....	17
3.4.1. Nüfus Büyüklüğü	17
3.4.2. Çaprazlama Olasılığı.....	17
3.4.3. Mutasyon Olasılığı.....	18
3.4.4. Maksimum Nesil Sayısı	18
4 SONUÇ.....	19
KAYNAKÇA.....	20
ÖZ GEÇMİŞ	22

KISALTMA LİSTESİ

DAR – Document Analysis and Recognition

OCR – Optical Character Recognition

GA – Genetik Algoritma

HT – Hough Transform

OZZ – Orthogonal Zig – Zag

CAD – Computer Aided Design

GIS – Geographic Information System

GP – Genetik Programlama

ŞEKİL LİSTESİ

Şekil 2.1 (a) Raster Görüntü (b) Vektör Görüntü

Şekil 2.2 Vektörizasyonun Üç Ana Bölümü

Şekil 3.3.4.1 Rulet Tekerı Seçimi için Kromozom dağılımı

Şekil 3.3.4.2 Rank Seçimi Yönteminden Önce

Şekil 3.3.4.3 Rank Seçimi Yönteminden Sonra

Şekil 3.3.5.1 Tek Noktalı Çaprazlama

Şekil 3.3.5.2 Çift Noktalı Çaprazlama

Şekil 3.3.5.3 Tek Biçimli Çaprazlama

Şekil 3.3.5.4 Aritmetik Çaprazlama

Şekil 3.3.5.5 Ağaç Çaprazlama

Şekil 3.3.6.1 Bit Ters Çevirme İşlemi

Şekil 3.3.6.2 İşlev veya Numara Değiştirme İşlemi

TABLO LİSTESİ

Tablo 3.2.1.1 İkili Kodlanmış Kromozom Örnekleri

Tablo 3.2.1.2 Permütasyon Kodlanmış Kromozom Örnekleri

Tablo 3.2.1.3 Değer Kodlanmış Kromozom Örnekleri

Tablo 3.2.1.4 Ağaç Kodlanmış Kromozom Örnekleri

Tablo 3.3.5.1 Tek Noktalı Çaprazlama Örneği

Tablo 3.3.5.2 Çift Noktalı Çaprazlama Örneği

Tablo 3.3.5.3 Tek Biçimli Çaprazlama Örneği

Tablo 3.3.5.4 Aritmetik Çaprazlama Örneği (AND)

Tablo 3.3.6.1 Bit Ters Çevirme İşlemi Örneği

Tablo 3.3.6.2 Sıra Değiştirme İşlemi Örneği

Tablo 3.3.6.3 Sayı Ekleme İşlemi Örneği

ÖNSÖZ

Büyük bir hızla gelişmeye devam eden bilişim teknolojisi ve bununla paralel olarak gelişen ve değişen yazılım geliştirme teknikleri, biz bilgisayar mühendislerini ve bu alanda çalışan diğer IT uzmanlarını ister istemez daha akıllı sistemlere ve bu sistemlerin geliştirilebilmesine, şimdilik kısmen de olsa imkân sağlayabilen yazılım geliştirme tekniklerine yönlendirmektedir. İşte bu kapsamda, genetik algoritmalar gibi yapay zekâ tekniklerinin kullanımı artacak ve gelişecektir.

Raster görüntülerin vektörizasyonu konusunda ve özellikle eskiden beri ilgimi çeken yapay zekâ konularından biri olan genetik algoritmalar; kodlama, seçim, çaprazlama, mutasyon gibi genetik algoritma operatörleri üzerine çalışmak benim için oldukça keyifli ve eğitici oldu.

Çalışmanın not için değil, öğrenmek için yapıldığını idrak etmemi sağladığı ve tez çalışmam süresince benden destek ve yardımlarını esirgemediği için tez danışmanın Sayın Yrd. Doç. Dr. Erchan Aptoula'a teşekkürü bir borç bilirim.

ÖZET

RASTER GÖRÜNTÜLERİN VEKTÖRÜZASYONUN'DA GENETİK ALGORİTMA YAKLAŞIMI

EŞME, Taner

Lisans Tezi, Mühendislik - Mimarlık Fakültesi

Tez Danışmanı: Yrd. Doç. Dr. Erchan, APTOULA

Ocak 2011, 19 Sayfa

Bu çalışmada, raster görüntülerin vektörizasyon sorununa değinilecek, bu bağlamda yapılan çalışmalar değerdendirilecek, aynı zamanda genetik algoritmalar ve genetik algoritmaların raster görüntülerin vektörizasyonu üzerinde nasıl kullanılabileceğı ve yapılan çalışmaların sonuçları ele alınacaktır. Bu çalışmalar kapsamında, Visual Studio 2010 platformu üzerinde C#.NET kullanılarak, genetik algoritmalar ile raster görüntülerin vektörsel olarak ifade edilmesini sağlayan bir uygulama geliştirilecektir. Geliştirilen bu uygulama ile değışik genetik algoritma parametreleri kullanılarak raster görüntülerin vektörizasyonu yapılabilecek ve farklı genetik algoritma parametreleri için elde edilen sonuçlar karşılaştırılmalı olarak sunulacaktır.

Anahtar Kelimeler: Genetik Algoritma, Vektörizasyon

ABSTRACT

**A GENETIC ALGORITHM BASED APPROACH TO
VECTORIZATION OF RASTER IMAGES**

EŞME, Taner

Undergraduate Thesis, Faculty of Architecture and Engineering

Supervisor: Asst. Prof. Dr. Erchan, APTOULA

January 2011, 19 Pages

In this study, the problem of vectorization of raster images, is addressed; also discussed genetic algorithms and how to be able to use genetic algorithms over of vectorization of raster images and the results of the studies done. In the extent of such studies, an application for vectorization of raster images is going to be implemented using genetic algorithm techniques on Visual Studio 2010 platform using C#.NET. Application to be implemented can perform vectorization of raster images with various parameters of genetic algorithm and the results obtained for different parameters of genetic algorithm are going to be represented comparatively.

Keywords: Genetic Algorithm, Vectorization

1 GİRİŞ

Günümüzde artık yapay zekânın ve yapay zekâ tekniklerinin uygulanma alanları giderek artmaktadır. Bu yapay zekâ tekniklerinden birisi de genetik algoritmalar (GA). Genetik algoritmalar; optimizasyon, otomatik programlama ve bilgi sistemleri, makine öğrenimi, ekonomik ve sosyal sistem modelleri gibi alanlarla birlikte finans, pazarlama, üretim gibi işletme alanlarında da uygulanabilmektedir [1].

Vektörizasyon, raster görüntülerden vektörlerin elde edilmesidir. Raster bir görüntü ızgara şeklinde birbirinden ayrık piksellerden oluşur ve her piksel kendine ait koordinat ve renk değerlerine sahiptir. En basit manada, böyle bir görüntü üzerinde uygulanan büyütme işlemi sonucunda görüntü kalitesi ile ters orantılı olarak kareleşme görülecektir. Bu da görüntünün büyütülen kısmının kalitesini düşürecek ve kullanılabilirliğini azaltacaktır. Ancak bir görüntünün vektör ifadesi; çizgi, eğri, poligon gibi temel geometrik şekillerden ve bu şekillerin koordinatlarını içeren bilgilerden oluşmaktadır [2]. Vektörel olarak ifade edilen bir görüntü üzerinde büyütme işlemi uygulanırsa yapılması gereken sadece görüntüyü ifade eden vektörleri daha büyük çizmek olacağından raster görüntülerin aksine her hangi bir kareleşme ya da bozulma görülmecektir. Bu bakımdan, vektörizasyon ve vektörizasyonun doğruluğu özellikle harita sektöründe büyük öneme sahiptir.

Bu çalışmada raster görüntülerin vektörizasyonu için genetik algoritmalar kullanılmıştır. Genetik algoritmaların kullanılması ile diğer vektörizasyon algoritmalarının uygulamak zorunda oldukları verilerin ikili formda sunumu (binarization) ve kenar izleme (Tracing) gibi adımlar uygulanmamaktadır. Böylece vektörizasyon algoritmalarının en büyük zayıflıklarından olan belli karakteristiğe sahip şekillere uygulanabilme sorunu ortadan kalkmakta ve farklı karakteristiğe sahip şekillerden oluşan karmaşık görüntülere de uygulanabilmektedir.

2. bölümde vektörizasyon konusu anlatılmış ve raster görüntülerin vektörizasyonu için kullanılan bazı yaklaşımlar ele alınacaktır.

3. bölümde genetik algoritmalar konusu ele alınmış, GA operatörleri ve GA parametreleri gibi GA kavramları anlatılacaktır.

4. bölümde ise genetik algoritmaların, raster görüntülerin vektörizasyonun da kullanımı konusu üzerinde durulacaktır..

2 VEKTÖRİZASYON

Vektörizasyon, raster bir görüntüden vektörlerin yani iki noktadan oluşan düz çizgi parçalarının elde edilmesidir [3]. Başka bir deyişle, raster bir görüntünün piksel gösterimini vektör gösterime dönüştürmek için görüntünün analiz edilmesidir [4].



Şekil 2.1 (a) Raster Görüntü (b) Vektör Görüntü [5]

Yukarıda raster bir görüntü ve aynı raster görüntünün vektörizasyonu gösterilmiş.

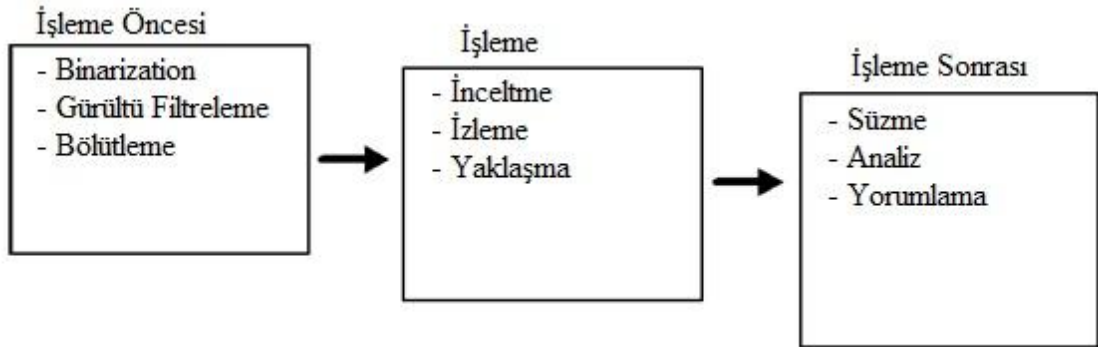
Raster görüntülerden elde edilen vektör; nokta, çizgi ya da poligon(çokgen) olarak depolanabilen verilerden oluşmaktadır. Noktalar (x, y) gibi koordinatlar kullanılarak, çizgiler ise (x_1, y_1) ve (x_2, y_2) gibi her bir çiftin düz bir çizgi parçası temsil ettiği nokta çiftleri kullanılarak depolanmaktadır. Yani bir çizgi $(x_1, y_1), (x_2, y_2), \dots(x_r, y_r)$ gibi koordinatlarla tanımlanabilir [5].

Vektörizasyon, harita sektöründe mevcut analog haritaların sayısallaştırılmasında[5] ve yüksek seviye obje tanıma işlemleri gibi DAR (Document Analysis and Recognition) alanlarında yaygın olarak kullanılmaktadır. Örneğin, OCR (Optical Character Recognition) vektörizasyonun kullanım alanı olarak gösterilebilir [3]. Böyle bir vektörizasyon, harita, şema ve mühendislik çizimleri gibi taranmış görüntülerin dönüşümü için daha uygundur [4].

Görüntü işleme tekniklerinin gelişmeye başladığı son 30 yıldır, pek çok vektörizasyon algoritması geliştirilmiştir [3]. Ancak bu algoritmaların raster görüntülerin vektörizasyon işlemleri için hala söz konusu olan kesinlik, sağlamlık ve değişmezlik problemleri vardır [6]. Raster görüntü üzerindeki vektöre dönüştürülecek şekillerin karakteristik özelliklerinin farklı olmasından, her farklı şekil için farklı bir algoritma gereksinimi söz konusudur [7].

Bu güne kadar geliştirilen ve uygulanan vektörizasyon algoritmalarını kabaca altı bölüme ayırabiliriz. Bunlar, Hough dönüşümü (Hough Transform-HT) tabanlı, inceltme tabanlı (Thinning-based), şekil izleme tabanlı (Contour-based), grafik yürütme tabanlı (Run-graph-based), ağ modeli tabanlı (Mesh pattern based) ve ayrık piksel tabanlı (Sparse pixel based) metotlardır [3]. HT tabanlı metodun en basit versiyonu çizgileri tarar fakat daha karmaşık şekiller için geliştirilebilir. İnceltme tabanlı metotlarda, piksel genişliği tek piksel olana kadar şekli oluşturan pikseller silinmektedir. Şekil tabanlı metotlarda, ilk önce şekiller izlenir ve çizgileri taramak için şekillerin benzerleri taranır. Merkez eksenler bu şekil çizgileri arasında oluşturulmaktadır. Grafik yürütme tabanlı metotlar da, run lenght encoding hesaplama için satır ya da sütunlarda raster görüntüleri araştırmaktadır. Yürütmeler grafik yapıları oluşturmak için analiz edilmektedir. Ayrık piksel tabanlı metot Dori tarafından geliştirilen Orthogonal Zig-Zag (OZZ) yönteminin geliştirilmiş halidir. Yöntemde tüm piksellere dokunmadan zig-zag çizilerek şeklin vektörizasyonu yapılmaktadır [7]. Ağ modeli tabanlı metot da ise, görüntü belirli bir ağ yapısına bölünerek bir kontrol haritası oluşturulur ve bu kontrol haritasının analizi yapılmaktadır [3].

Raster görüntülerin vektörizasyonu sırasında yapılan işlemleri üç ana bölüme ayırabiliriz. Bunlar; işleme öncesi, işleme ve işleme sonrası.



Şekil 2.2 Vektörizasyonun Üç Ana Bölümü [4]

İşleme öncesi aşama, raster görüntünün vektörizasyon için hazırlanma aşamasıdır. Bu aşamada renkleri gri değere çevirmek gibi vektörizasyonun doğruluğunu arttıracak işlemler yapılır [5]. Gri ölçekli görüntü ikilileştirilmeli (binarization işlemi), ikili görüntü ya da gri ölçekli görüntü, gürültü azaltmak için filtrelenmelidir. Gri ölçekli girdi görüntüsü iskeletleştirme (skeletonization) aşamasından önce ikilileştirilmelidir. İkileştirme işleminin amacı görüntüyü arka ve ön plan pikselleri olarak ayırmaktır. İkileştirme işlemi arkasından yapılan iskeletleştirme işleminin sonucunun kalitesi girdi olarak verilen ikili görüntünün kalitesine bağlıdır. Gri ölçekli görüntünün ikilileştirilmesi, bazı T eşik değeri ile görüntünün eşiklemesi ile yapılır. Eğer taranmış dokümanın homojen olarak aydınlatılmaması sonucu ya da dokümanın kendisindeki homojensizlikten dolayı böyle görüntülerde ikilileştirme eşiği (binarization threshold) seçmek çok zordur. Bu problem için ise, görüntüyü bloklara bölerek istatistiksel olarak

her blok için yerel bir eşik belirlemeye dayalı bir yöntem geliştirilmiştir. Tüm bunlara göre, işleme-öncesi aşamasında yapılacak işlemleri görüntünün tipi ve kalitesi belirler [4].

İşleme aşaması, vektörizasyon bu aşamada yapılır. Raster bir görüntüden vektörleri çıkarmak için, görüntünün hangi bölgesinde çizgi olduğunu ve nerede başlayıp nerede bittiğini bulmak gerekir [5]. Bu aşamanın çıktısı girdi görüntüsünün vektör ifadesidir [4]. Bu aşamada kullanılabilecek belli başlı yöntemlere yukarıda yer vermiştik. Bu metotların hepsinin artıları ve eksileri vardır. Bu yöntemlerden inceltme tabanlı gibi iskeletleştirmeye (skeletonization) dayalı metotlar iyi sonuçlar ortaya çıkarır, ancak bu metotlar gürültüye karşı hassastır. Şekil tabanlı metotlar ise, gürültüye karşı daha dayanıklı iken bu yöntemler de sezgisel ve karmaşık eşleştirme tasarılarına bel bağlar. Bu bağlamda, kullanılması gereken doğru metodu vektörizasyonu yapılacak olan raster görüntünün tipi (harita, çizim gibi) belirler [4]. Bu aşama da yapılan işlemler çoğunlukla kenar çıkarma ve görüntüdeki şekillerin iskelete yapılarının çıkarılması işlemlerini kapsar. Yukarıda da anlatılan yöntemler çoğunlukla kenar çıkarma işlemine dayanırlar, ancak her kenar çıkarma metodu her görüntü için kaliteli sonuç vermemektedir. Örneğin çoğunlukla çizgilerden oluşan mühendislik çizimleri gibi görüntülerde türev tabanlı kenar çıkarma yöntemi kaliteli sonuç vermektedir [5]. Kenar çıkarma için kullanılan çokgen yaklaşımı (polygonal approximation) yöntemi de harita görüntüsü gibi görüntülerin vektörizasyonunda iyi sonuçlar vermektedir [8]. Bunların dışında da MUSCLE adı verilen klasik yöntemlerden farklı olarak görüntünün değişik yönlerde analiz edilmesi yoluyla çizgilerin elde edilmesi temeline dayalı bir vektörizasyon yöntemi de vardır [9].

İşleme sonrası aşama da ise ana hedef vektör görüntünün analiz ve yorumlamasıdır. Bu aşamada vektör görüntüden gürültülerin temizlenmesi, nesne vektörizasyonu, nesne tanıma gibi işlemler yapılır [4].

2.1. VEKTÖR GÖRÜNTÜLERİN AVANTAJLARI

Raster görüntüler bir şeklin boyutu ya da geometrisi hakkında belirgin bir bilgi sağlayamazlar. Yani bir görüntünün vektörsel olarak gösteriminin raster gösterime göre pek çok avantajı vardır. Bu avantajlardan bazıları şöyle sıralanabilir [5].

- ✓ Vektör görüntülerin depolandıkları dosya boyutu raster görüntülerinkine göre daha küçüktür. Çünkü vektör görüntülerde sadece nokta koordinatları depolanır iken raster görüntüler de tüm piksellerin dijital değerleri depolanır.
- ✓ Bir görüntünün vektörsel olarak ifade edilmesi, Geographic Information System (GIS) ve Computer Aided Design (CAD) sistemlerinde geometrik şekillerin ifade edilebilmesi için iyi bir tercihtir. Çünkü vektör bir görüntüdeki şekillere ve eğrilere, ölçekleme gibi matematiksel dönüşümler kolaylıkla uygulanabilmektedir.

- ✓ Vektörizasyonun en önemli avantajlarında birisi de esnek veri işleyebilmesidir. Vektör görüntüler çözünürlük kaybı olmaksızın ölçeklenebilir.
- ✓ Vektör görüntülerde katmanlar oluşturulabilir ve farklı katmanlar görüntülenebilir. Sadece ilgilenilen katman gösterilerek görüntü daha okunaklı ve daha az karmaşık hale getirilebilir.
- ✓ Vektör görüntüleri oluşturan çizgi ve daire gibi şekillerin genişlikleri değiştirilebilir ve aynı zamanda 3D bir model oluşturulabilir.

Bu bölümde vektörizasyonun tanımına, vektörizasyon için kullanılan belli başlı yöntemlere ve bu yöntemlerin kapsadığı işlemlere değindik. Bu çalışmada gerçekleştirilecek olan ise, raster görüntülerin vektörizasyonun da Genetik Algoritmaların uygulanması olduğundan, öncelikle Genetik Algoritmalar nedir, hangi operatör ve parametrelere sahiptir, bu operatör ve parametrelerin kullanımı nasıldır gibi konuların ele alınması gerekir.

3 GENETİK ALGORİTMALAR (GA)

Genetik Algoritmalar (GA) C. Darwin'in evrim ilkelerini kullanarak problemlere çözüm getirmeye çalışan bir arama ve optimizasyon yöntemidir. GA, var olan olası bir çözüm kümesinden en iyi çözümü evrimleştirmeye çalışır.

GA'ın fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. Geleneksel optimizasyon yöntemlerine göre farklılıkları bulunan GA, parametre kümesini (çözüm uzayındaki olası çözümleri) değil kodlanmış biçimlerini kullanır. Olasılık kurallarına göre çalışan genetik algoritmalar yalnızca amaç fonksiyona gereksinim duyar. Çözüm uzayının hepsini değil belli bir kısmını taradıkları için etkin arama yaparak çok daha kısa sürede çözüme ulaşırlar. GA arama yapısından dolayı diğer yapay zekâ yöntemlerine göre daha üstündür [1].

GA'nın tarihçesine bakacak olursak, evrimsel hesaplama 1960'larda I. Rechenberg'in Evrimsel Stratejiler adlı çalışmasında tanıtılmıştır. Daha sonra fikri diğer araştırmacılar tarafından geliştirilmiştir [11]. GA'nın temel prensipleri 1970'li yılların başlarında John Holland tarafından ortaya atılmış ve J. Holland yaptığı çalışmalarını "Adaptation in Natural and Artificial Systems" adlı eserinde bir araya getirmiştir. GA çalışmaları 1980'lerin ortalarına kadar bilgisayar sistemlerinin işlem hızı gibi yeteneklerinin, GA çalışmaları için yeterli olmamasından dolayı, çoğunlukla teoride kalmıştır. 1992 yılına gelindiğinde, John Koza GA yardımı ile programları evrimleştirmek üzerine çalışmalar yapmıştır ve bu yönteme "Genetik Programlama(GP)" adını verdi. LISP dilinde programlara Ayırıştırma Ağaçları (Parse Tree) şeklinde ifade edildiği için LISP diliyle geliştirilmiştir [12]. Ayırıştırma Ağaçları olası çözüm uzayındaki her bir bireyin (kromozom) kodlanma biçimini ifade eder.

GA'nın kullanım amacı, determinist (gerekirci) olarak ifade edilemeyen, çok sayıda kısıt içeren ve hiçbir çözüm tekniği bulunmayan, örneğin NP-Hard (Non-deterministic Polynomial Time) olarak adlandırılan [10] problemlerin çözümlerini araştırmaktır.

GA kullanımına anlaşılır bir örnek vermek gerekirse şöyle açıklayabiliriz. Örneğin; dünyadaki en iyi ressamı bulmaya çalıştığımızı düşünelim. Bu problemin ilk akla gelen çözümü, dünya üzerindeki tüm insanların ressamlıklarına bakarak aralarından en iyi olanı bulmaya çalışmaktır. Ancak pratikte böyle bir şey elbette mümkün değildir. Bu probleme GA yaklaşımı ile çözüm getirmeye çalışırsak yapılması gereken dünyanın farklı bölgelerinden rastgele insanları seçerek bu insanlardan oluşan bir topluluk oluşturmak ve bu topluluktaki kişilerin her birisinin resim çizme yeteneklerine bakmaktır. Yani her bir bireye resim çizdiriyor ve nasıl resim çizdiklerine bakıyoruz (Tabi burada iyi resmin nasıl olacağının tanımlanması gerekir.). İyi resim çizenlere para verdiğimiz ve kötü çizenlere vermediğimizi düşünelim. Buna göre parası olan bireylerin evlenerek çoğalacağını ve parası olmayanların ise çoğalamayacağını düşünürsek iyi resim yapan kişiler çoğalacak ve bir sonraki nesilde daha iyi resim yapan

kişileri oluşturacağını varsayabiliriz. Daha sonra birinci nesil için uygulanan işlemleri bu yeni oluşan nesil için de uyguluyoruz ve daha sonraki nesil için ve daha sonraki için... Sonuç olarak her bir sonraki nesilde daha iyi resim çizen bireylerin oluşacağı düşünülürse belli bir nesilden sonra artık çok iyi bir ressam elde ediyoruz. Ancak elde edilen bu ressamın dünyadaki en iyi ressam olduğunun bir garantisi yok. Çünkü GA'nın en büyük problemlerinden birisi yerel bir en iyi çözüme takılabiliyor olmaları. Yani elde edilen en iyi ressam dünya üzerindeki en iyi ressam değil de belirli bir bölgedeki yerel en iyi ressam olabilir.

GA her zaman yerel en iyi çözümden kurtulmaya çalışır. Yerel en iyi çözüme takılmayı ise şöyle açıklayabiliriz. Dünyadaki en iyi ressamı bulma problemini düşünelim, her bir ressamın ressamlığını (problemin çözümü için uygunluğunu) topolojik olarak yani tepelerle ifade edersek, en iyi ressama ait tepe en yüksek olacaktır. Ancak biz ressamı seçerken rastgele bir seçim yapmıştık. Rastgele ressam seçme işlemini ise dağların üzerine paraşütçüleri bırakmak ve paraşütçülerin düştüğü yerdeki birey (ressam) seçiliyor olarak düşünürsek, bırakılan paraşütçülerden bazıları tepelerin üzerine bazıları ise ovalara düşecektir. Rastgele seçilen bu bireyler Genetik Algoritmamızın başlangıç nüfusunu oluşturacaktır. Bu yüzden rastgele seçilen bu bireyler arasında da bir eleme işlemi yapılarak Genetik Algoritmamız için yarar sağlayabilecek bireylerin elde edilmesi gerekir. Bu işlemi de daha anlaşılır şöyle ifade edebiliriz, belli bir yüksekliğin altındaki paraşütçüleri ayıların yediğini düşünürsek, böylece belli bir eşik değerinin altındaki bireyler bizim başlangıç nüfusumuzdan elenmiş olacak ve Genetik Algoritmamız için daha yararlı bireyler kalmış olacaktır. Tabi ki tüm bireylerinde ovalara düşme ve yok olma ihtimali söz konusudur. Ancak tüm bunların sonucunda elde edilen bireylerin hiç birisi bizim ulaşmak istediğimiz en iyi ressamın tepesi üzerinde olmayabilir. Yani GA mutlak en iyiyi bulmak yerine, yerel bir en iyiyi bulmaya çalışabilir. Çünkü GA işlemleri sonucunda oluşan daha iyi bireylerle sadece rastgele seçilmiş olan yerel en iyiye ulaşmaya çalışılır. Bunu önlemek için ise GA'da mutasyon işlemi yer alır. Mutasyon işlemi bireyden bağımsız olarak gerçekleştiği için GA'yı yerel en iyi çözümden kurtulmasını sağlayabilir.

3.1. BİYOLOJİK ALT YAPI

Yaşayan tüm organizmalar hücrelerden oluşur ve her hücrede aynı **kromozom** kümesi bulunur. Kromozomlar birer DNA dizisidir ve tüm organizmanın bir modelini temsil ederler. DNA ise **gen**lerden oluşur. Her bir gen bireyin göz rengi, boyu gibi bir özelliğini kodlar. Tüm genetik malzeme kümesine yani tüm kromozomlara genom denilir. Genom üzerindeki belli gen gruplarına genotip denilir. Genotipler doğumdan sonra gelişerek göz rengi, zekâ gibi organizmanın fiziksel ve karakteristik fenotiplerine dönüşür [11][12].

3.2. BASİT GENETİK ALGORİTMA

Bir genetik algoritma en basit haliyle aşağıdaki adımlardan oluşmaktadır [13].

```
{  
    Başlangıç Nüfusunu Belirle;  
    Uygunluk_Olcutunu_Denetle;  
    While Çıkış_Kriteri_Sağlanmadıkça  
    {  
        Üretim için seçim yap;  
        Çaprazla ve Mutasyona uğrat;  
        Nüfusu Evrimleştir;  
    }  
}
```

Yukarıda bir GA'nın basit yapısı gösterilmiştir. Bu işlemleri şöyle açıklayabiliriz [1]:

- ✓ Genellikle rastsal bir çözüm kümesi seçilir; başlangıç nüfusu olarak kabul edilir.
- ✓ Arama uzayındaki tüm mümkün çözümler kodlanır.
- ✓ Nüfusta ki her bir olası çözüm için bir uygunluk değeri hesaplanır, bulunan uygunluk değeri bireyin çözüm kalitesini gösterir.
- ✓ Bir grup birey belirli bir olasılık değerine göre rastsal olarak seçilip çoğalma işlemi gerçekleştirilir.
- ✓ Yeni bireyin uygunluk değeri hesaplanarak çaprazlama ve mutasyon işlemlerine tabi tutulur.
- ✓ Önceden belirlenen kuşak sayısı kadar yukarıdaki işlemler tekrar edilir.
- ✓ İterasyon belirlenen kuşak sayısına ulaşıncaya ya da belirtilen sonlandırma kriteri sağlanıncaya işlem sona erdirilir. Amaç fonksiyona göre en uygun birey seçilir.

3.3. GA BİLEŞENLERİ

3.3.1. Kodlama Tekniği (Encoding)

Kodlama, kromozomların genlerle ifade edilmesidir. Olası çözüm uzayındaki her bir birey temsil ettiği çözüm hakkında bilgi içermelidir [11]. Bireylerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır [1]. Bireylerin kodlanması için pek çok yöntem kullanılabilir. Bu yöntem çoğunlukla çözülen probleme bağlıdır. GA'nın hızlı ve güvenilir şekilde çalışabilmesi için kodlamanın doğru şekilde yapılması gerekir.

İkili (Binary) Kodlama: En çok kullanılan kodlama şekli ikili (binary) kodlamadır. Böyle kodlanan kromozomlar (bireyler) şu şekilde kodlanmaktadır.

Tablo 3.2.1.1 İkili Kodlanmış Kromozom Örnekleri

Kromozom 1	1001110101000111
Kromozom 2	1110110001110010

Her kromozom ikili karakter dizisi olarak ifade edilmektedir. Karakter dizisindeki her bit çözümün bir karakteristiğini temsil eder [11].

Permütasyon Kodlama: Bu kodlama tekniği, gezgin satıcı problemi ve çizelgeleme gibi düzenleme problemlerinde kullanılır [12].

Tablo 3.2.1.2 Permütasyon ile Kodlanmış Kromozom Örnekleri

Kromozom 1	9 6 1 9 4 5 2 6 8 3 1
Kromozom 2	4 7 1 3 7 8 1 4 6 1 7

Bununla beraber, gerçek sayı gibi karmaşık değerlerin kullanıldığı problemlerde doğrudan değer kodlama kullanılabilir. Bu tip problemler için ikili kodlama çok zor olmaktadır [12].

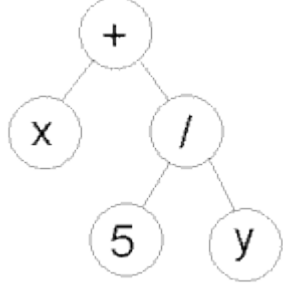
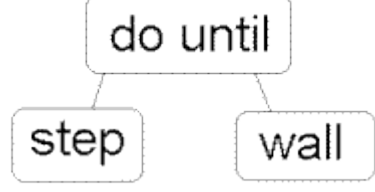
Değer kodlama: Her kromozom bazı değerlerin dizisi şeklindedir. Bu değerler problem ile bağlantılı olan gerçek sayılar, karakterler yada bazı karmaşık nesneler olabilir [12].

Tablo 3.2.1.3 Değer Kodlanmış Kromozom Örnekleri

Kromozom 1	2.3263 4,7155 0,7821 1,4562 6,5904
Kromozom 2	PAEMNDTANERLPDKSIOKRE
Kromozom 3	GERİ İLERİ SAĞA SOLA

Ağaç Kodlama: GP için kullanılan kodlama tekniğidir. Ağaç kodlama tekniği çoğunlukla program ve ifade evrimleştirmek için kullanılır. Bu kodlama tekniğinde her kromozom, programlama dillerindeki fonksiyon ve komutlar gibi bazı nesnelerden oluşan bir ağaçtır [12].

Tablo 3.2.1.4 Ağaç Kodlanmış Kromozom Örnekleri [12]

Kromozom 1	Kromozom 2
	
$(+ x (/ 5 y))$	$(do\ until\ step\ wall)$

Ağaç kodlama program evrimleştirmek için uygundur. LISP programlama dili, ağaç yapısına kolayca ayrıştırılabildiği için sıklıkla kullanılmaktadır [12].

3.3.2. Başlangıç Nüfusunun Oluşturulması (Initial Population)

GA'nın başlaması için öncelikle problemin olası çözümlerini ifade eden bir başlangıç toplumunun hazırlanması gerekir. Başlangıç çözüm grubu (nüfus) rastsal olarak seçilir. İkili kodlamanın kullanıldığı bir GA'da başlangıç nüfusunun oluşturulması için rastsal sayı üreticiler kullanılır. Rastsal sayı üreticisinin ürettiği değer 0,5'den küçük ise konum 0, değil ise 1 değerine ayarlanır. Birey sayısının kromozom uzunluğundan az olduğu problemlerde yazı-turayla da konum değerleri belirlenebilir [1].

3.3.3. Uygunluk Değerinin Hesaplanması (Fitness Function)

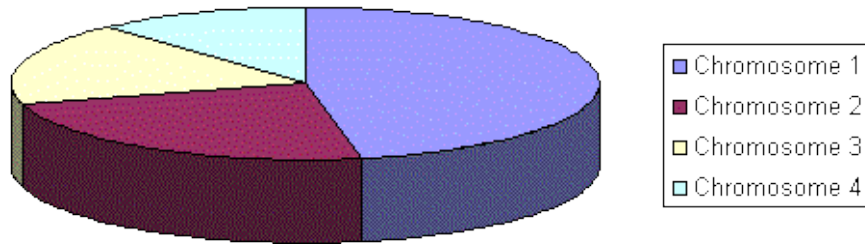
Evrin teorisine göre, doğa da güçlü olan bireyin hayatta kalma şansı daha fazladır. Bu ifade Genetik Algoritmalarda da geçerlidir. Burada sorulması gereken soru şudur: "Hangi bireyin daha güçlü (ya da problemin çözümü için daha uygun) olduğuna nasıl karar verebiliriz?". Bu sorunun cevabı uygunluk (Fitness) fonksiyonudur. Bir uygunluk fonksiyonu daha iyi bireyler için daha yüksek değer döndürmelidir [14].

Verilen bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiği çözümün kullanımı ile veya yeteneği ile orantılı olan sayısal bir uygunluk değeri verir. Bu bilgi her kuşakta daha uygun çözümlerin seçiminde yol göstermektedir. Bir çözümün uygunluk değeri ne kadar yüksekse, yaşama ve çoğalma şansı o kadar fazladır ve bir sonraki kuşakta temsil edilme oranı da o kadar yüksektir [1].

3.3.4. Seçme İşleminin Yapılması (Selection)

Yeni nüfus oluşturulurken hangi ferdin üreme için seçileceğine karar verme işlemidir. Seçme işlemi bireylerin uygunluk değerleri baz alınarak yapılır. Birçok seçim yöntemi vardır. Şimdi bu yöntemlerden bazılarını inceleyelim.

Rulet Tekerı Seçimi: Bireyler uygunluk değerlerine göre seçilirler, buna göre daha iyi bireyler daha fazla seçilme şansına sahiptir. Toplumdaki tüm kromozomların, yerinin boyutu uygunluk değeri ile orantılı olacak şekilde bir rulet tekerine yerleştirildiğini düşünelim. Buna göre daha uygun olan kromozom daha geniş yere sahip olacaktır [11].



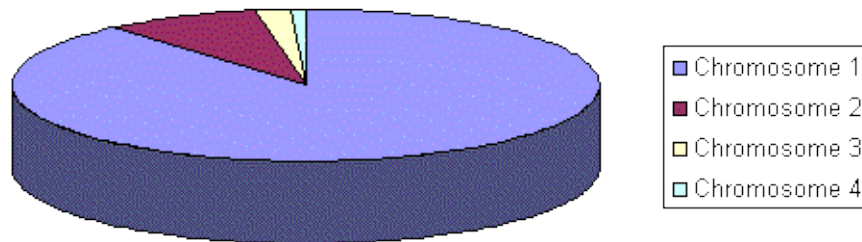
Şekil 3.3.4.1 Rulet Tekerı Seçimi için Kromozom dağılımı [12]

Böyle bir rulet tekeri üzerine bir bilye atılmakta ve bilyenin üzerinde durduğu kromozom seçilmektedir. Böylece daha fazla uzunluğa sahip olan kromozomların seçilme şansı artmaktadır. Bu teknikte uygunluk değerlerine göre kromozomlar bir rulet tekerine yerleştirilir. Seçim yapabilmek için rastgele bir sayı seçilir ve bu sayı hangi kromozom üzerine denk geliyorsa o kromozom bir sonraki adım için seçilir.

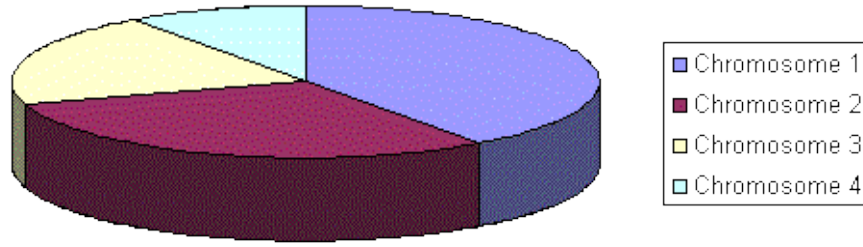
Rank (Sıralama) Seçimi: Rulet tekeri seçim tekniğinde eğer bir kromozomun uygunluk değeri tüm kromozomların uygunluk değerinin %90'ı gibi bir değere sahipse, bu durumda diğer kromozomların seçilme şansı çok azalacaktır [11]. Rank seçimi bunu önlemek için uygun bir yöntemdir.

Bu yöntemde ise; nüfustaki kromozomlar uygunluk değerlerine göre sıralanır. En kötü uygunluğa sahip olan 1, en yüksek uygunluğa sahip olan ise N (Nüfustaki birey sayısı) olacaktır [11].

Bir kromozomun uygunluk değerinin tüm kromozomların uygunluk değerinin %90'ı olduğu bir durum için kromozomların seçilme oranı şöyle olacaktır.



Şekil 3.3.4.2 Rank Seçimi Yönteminden Önce [12]



Şekil 3.3.4.3 Rank Seçimi Yönteminden Sonra [12]

Rank seçimi öncesi ve sonrası için her bir kromozomun seçilme olasılıkları yukarıda görülmektedir. Bu yöntem sayesinde, bir kromozomun uygunluk değerinin diğer tüm kromozomlarının uygunluk değerinin %90'ı olması gibi durumlarda da her kromozomun seçilme şansı vardır. Bu yöntemde de en iyi kromozomlar birbirinden çok farklı değillerdir [11].

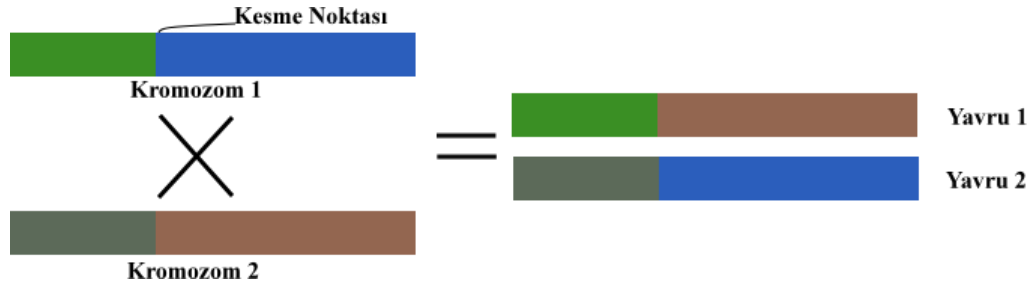
Kararlı Hal Seçimi: Bu nüfustan kromozom seçmek için bir teknik değildir. Bu yöntemde nüfustaki kromozomların büyük bit kısmı yeni nesle aktarılır. Bu yöntem şöyle çalışır. Nüfustan uygunluk değerleri yüksek olan kromozomlar seçilir ve yeni yavrular oluşturulur. Bu yavrular eski nüfustaki uygunluk değeri küçük olan bireylerin yerlerine konulur. Ancak nüfusun geri kalan kısmı değişmeden kalır [12].

Seçkinlik: Çaprazlama ya da mutasyon yöntemleri ile yeni nesil oluştururken, en iyi kromozomları kaybetme olasılığımız vardır. Seçkinlik en iyi kromozomların ya da bir kısmının önce kopyalanıp yeni nesle aktarıldığı yöntemdir. Seçkinlik, bulunan en iyi çözümün kaybolmasını önleyerek GA'nın başarımını arttırabilir [11].

3.3.5. Çaprazlama Operatörünün Uygulanması (Crossover)

Nüfustan uygunluk değerlerine göre seçilen, genlerle ifade edilen iki kromozomun birbiri ile gen alışverişinde bulunması işlemidir. Mevcut gen havuzunun potansiyelini arttırmak üzere, bir önceki kuşaktan daha iyi nitelikler içeren yeni kromozomlar yaratmak amacıyla çaprazlama operatörü kullanılmaktadır. Çaprazlama genellikle, verilen bir çaprazlama oranına eşit bir olasılıkla aile çeşitlerine (yani ebeveyn olarak seçilen iki kromozoma) uygulanmaktadır [1]. Kromozomun kodlanma tekniğine bağlı olarak çaprazlama yöntemi de değişiklik gösterir. Her kodlama tekniği için aynı çaprazlama yöntemi kullanılamaz. Şimdi başlıca kullanılabilecek çaprazlama yöntemlerini inceleyelim.

Tek Noktalı Çaprazlama: Her iki kromozom üzerinde de aynı olacak şekilde tek bir kesme noktası seçilerek yapılır. Kesme noktalarından itibaren her iki kromozomunda ilk kısımları yer değiştirilerek iki yavru elde edilir.



Şekil 3.3.5.1 Tek Noktalı Çaprazlama

Tablo 3.3.5.1 Tek Noktalı Çaprazlama Örneği

Kromozom 1	1001110 101000111
Kromozom 2	1110110 001110010
Yavru 1	1001110 001110010
Yavru 2	1110110 101000111

İki Noktalı Çaprazlama: Bu yöntemde iki kesme noktası seçilir ve kromozomların kesme noktalarının arasında kalan kısım bir kromozomdan, diğer kısım diğer kromozomdan alınarak yavrular oluşturulur.

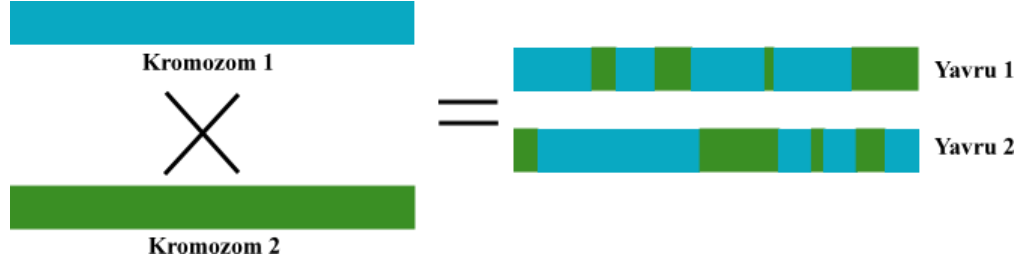


Şekil 3.3.5.2 Çift Noktalı Çaprazlama

Tablo 3.3.5.2 Çift Noktalı Çaprazlama Örneği

Kromozom 1	100 111010 1000111
Kromozom 2	111 011000 1110010
Yavru 1	100 011000 1000111
Yavru 2	111 111010 1110010

Tek Biçimli (Uniform) Çaprazlama: Bu yöntemde genler, ebeveyn kromozomlardan rastgele olarak seçilip kopyalanır.

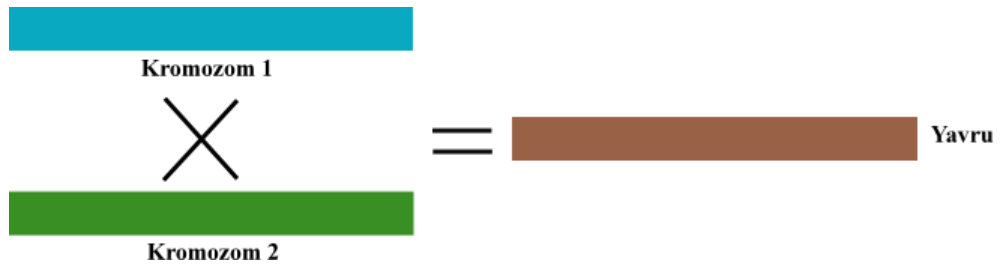


Şekil 3.3.5.3 Tek Biçimli Çaprazlama

Tablo 3.3.5.3 Tek Biçimli Çaprazlama Örneği

Kromozom 1	1001110101000111
Kromozom 2	1110110001110010
Yavru 1	1001110101110011
Yavru 2	1011110001000111

Aritmetik Çaprazlama: Yavrular, AND ve OR gibi mantık operatörlerinin ebeveynler arasında uygulanması ile elde edilir.

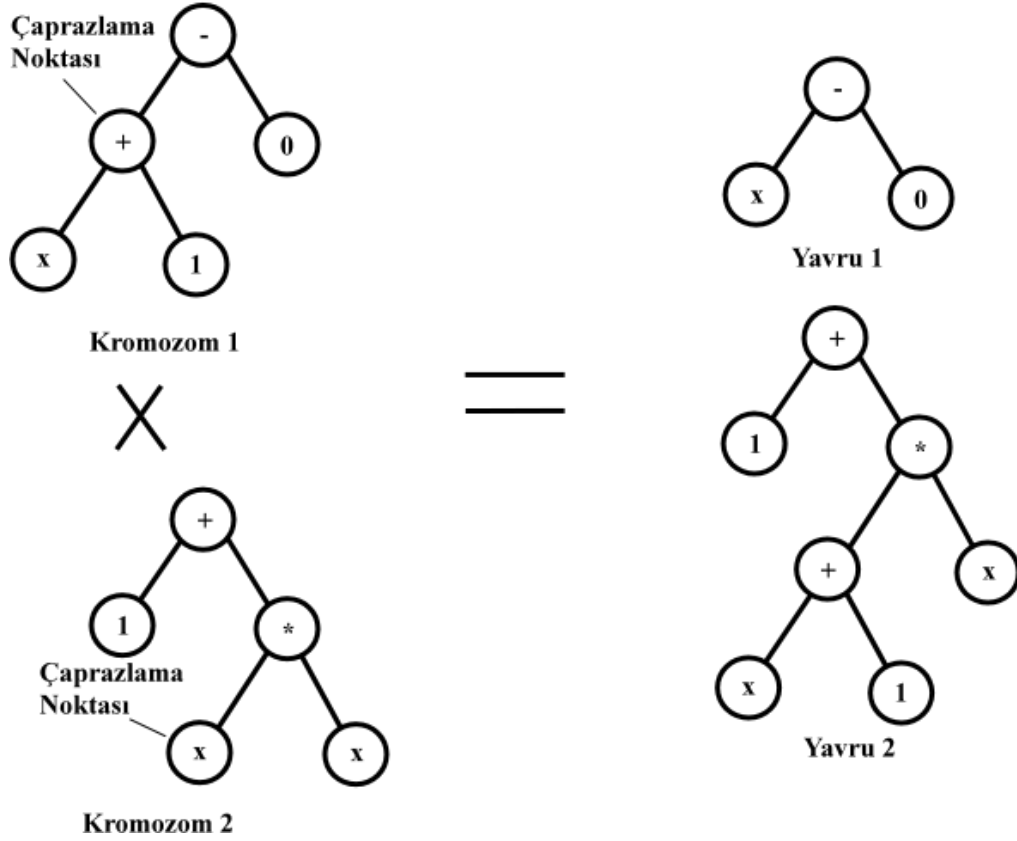


Şekil 3.3.5.4 Aritmetik Çaprazlama

Tablo 3.3.5.4 Aritmetik Çaprazlama Örneği (AND)

Kromozom 1	1001110101000111
Kromozom 2	1110110001110010
Yavru 1	1000110001000010

Yukarıdaki anlatılan çaprazlama yöntemlerinin dışında program evrimleştirmek için GP de kullanılan ağaç kodlama tekniğinde kullanılan çaprazlama yöntemi de vardır. Bu yöntemde her iki atada da bir kesme noktası seçilir. Ebeveynler bu noktalardan bölünerek ve bu noktaların altında kalan parçalar yer değiştirilerek yeni yavrular oluşturulur [11].



Şekil 3.3.5.5 Ağaç Çaprazlama

3.3.6. Mutasyon Operatörünün Uygulanması (Mutation)

Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat nüfus gerekli tüm kodlanmış bilgiyi içermez ise, çaprazlama tatmin edici bir çözüm üretmez. Bu yüzden mevcut kromozomlardan yeni bir kromozom üretmeye gerek vardır. Bunun için mutasyon operatörü kullanılır [1]. Bir kromozomdaki genlerin hepsi küçük bir bağımsız olasılık ile rastgele mutasyona tabidir [14]. Mutasyon genetik çeşitliliği sağlamak için kullanılır. Belli başlı mutasyon teknikleri şunlardır.

Bit Ters Çevirme: Bu yöntem ikili kodlama tekniğinin kullanıldığı metotlarda kullanılmaktadır.



Şekil 3.3.6.1 Bit Ters Çevirme İşlemi

Tablo 3.3.6.1 Bit Ters Çevirme İşlemi Örneği

Mutasyondan Önce	1001110101000111
Mutasyondan Sonra	1001110111000111

Sıra Değiştirme İşlemi: Bu yöntemde iki sayı seçilerek yer değiştirilir. Permütasyon kodlama tekniğinin kullanıldığı metotlarda kullanılmaktadır [11].

Tablo 3.3.6.2 Sıra Değiştirme İşlemi Örneği

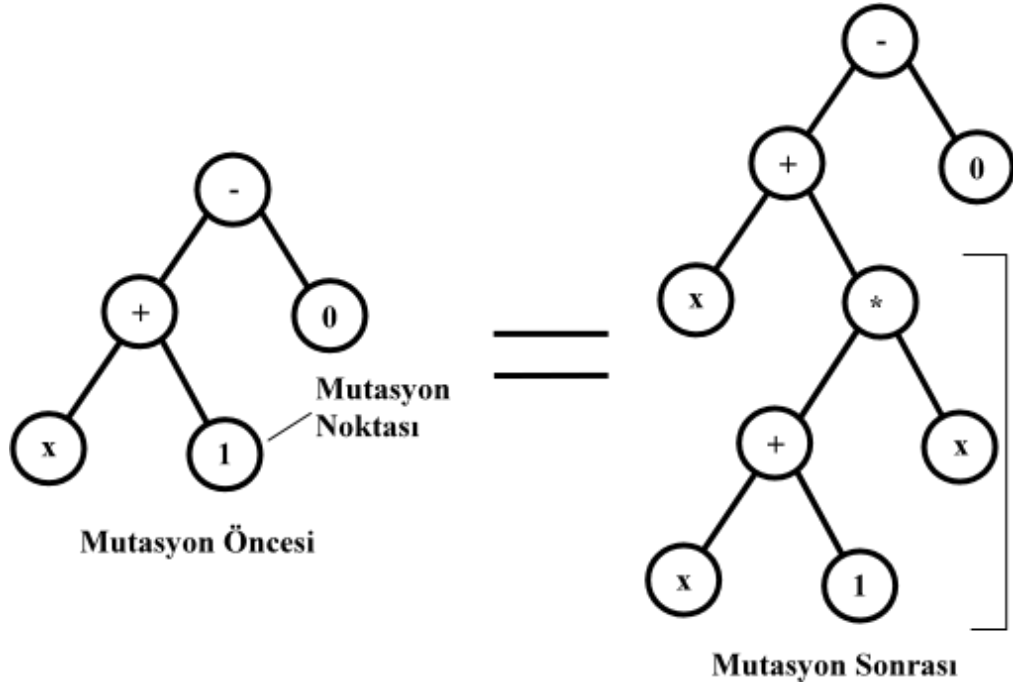
Mutasyondan Önce	9 6 1 9 4 5 2 6 8 3 1
Mutasyondan Sonra	9 9 1 6 4 5 2 6 8 3 1

Sayı Ekleme İşlemi: Değer kodlama tekniğinin kullanıldığı metotlarda kullanılır. Seçilen değere bir sayı eklenerek veya çıkarılarak yapılır [11].

Tablo 3.3.6.3 Sayı Ekleme İşlemi Örneği

Mutasyondan Önce	2.3263 4,7155 0,7821 1,4562 6,5904
Mutasyondan Sonra	2.3263 4,7155 1,5902 1,4562 6,5904

İşlev veya Numara Değiştirme: Bu yöntem ağaç kodlama tekniğinin kullanıldığı GP gibi yöntemlerde kullanılmaktadır. Seçilen mutasyon noktasının altında kalan ağaç silinerek yerine yeni bir ağaç geliştirilir.



Şekil 3.3.6.2 İşlev veya Numara Değiştirme İşlemi

3.4. GA PARAMETRELERİ

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır [1].

3.4.1. Nüfus Büyüklüğü

GA kullanıcısı tarafından verilen en önemli kararlardan biridir. Bu değer çok küçük olduğunda GA yerel bir optimuma takılabilir. Nüfus boyutunun çok büyük olması ise çözüme ulaşma zamanını uzatmaktadır. Bu konuda Goldberg 1985'te yalnızca kromozom uzunluğuna bağlı bir nüfus büyüklüğü hesaplama yöntemi önermiştir. Ayrıca Schaffer ve arkadaşları 1989'da çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20 – 30 arası bir nüfus büyüklüğünün iyi sonuçlar verdiğini belirtmiştir [1].

3.4.2. Çaprazlama Olasılığı

Çaprazlama olasılığı hangi sıklıkla çaprazlama yapacağımızı söyler. Eğer olasılık %0 ise, bu durumda yavrular ebeveynlerinin aynısı olacaktır. Eğer olasılık %100 ise, bu durumda tüm yavruların oluşumu sırasında çaprazlama gerçekleşecektir [12]. Çaprazlama olasılığının artması aynı zamanda iyi kromozomların bozulma olasılığını da arttırmaktadır [1].

3.4.3. Mutasyon Olasılığı

Kromozom parçalarının hangi sıklıkla mutasyona uğrayacağını göstermektedir. Eğer bu olasılık %0 ise yavrular sadece çaprazlama sonucunda oluşturulur ve GA yerel bir optimuma düşebilir. Eğer %100 ise bu durumda çaprazlama sonucunda oluşturulan yavrular tamamen değişecektir. Böyle bir olasılık da ise GA rastgele aramaya dönüşebilir [12]. Fakat bu aynı zamanda kayıp genetik malzemenin tekrar bulunmasına yardımcı olmaktadır [1]. Mutasyon olasılığı genelde 0.01 gibi düşük bir değer ile verilir.

3.4.4. Maksimum Nesil Sayısı

GA iterasyonunun kaçınıc nesle kadar süreceğini gösterir. Eğer aranan sonuç iterasyon nesil sayısına ulaşmıca kadar bulunamaz ise, nesil sayısına ulaşılınca nüfus içerisindeki en uygun kromozom (olası çözüm) GA'nın sonucu olarak döndürülecektir.

Bu bölümde GA'nın temel konuları ve kapsadığı kavramlar ele alınmıştır. GA'nın içerdiği operatörlere örnekler verilmiş ayrıntılı olarak ele alınmıştır. Bir önceki bölümde ise vektörizasyona konusu verilerek GA ve vektörizasyon konularında gerekli ön bilgiler verilmiştir. Yapılacak olan çalışmayla, 2. Bölümde anlatılan belli başlı vektörizasyon yöntemlerinin aksine görüntü üzerinden vektörleri elde etmek yerine, görüntü GA kullanılarak vektörlerle tekrar çizilmeye çalışılacaktır. Böylece bilinen vektörizasyon yöntemlerin de uygulanan kenar çıkarma, iskeletleştirme ve şekillerin belirlenebilmesi için uygulanan çokgen yaklaşımı gibi işlemler yapılmayacaktır. Bu sayede her farklı görüntü ya da görüntüdeki her farklı nesne için farklı bir algoritma gereksinimi sorunu çözümlenmektedir. Geliştirilecek olan uygulama her türlü görüntü ve görüntüdeki her türlü nesne için uygulanabilir olacaktır.

4 SONUÇ

Yapılacak olan uygulama ile raster görüntülerin vektörizasyonu için kullanılan yöntemlerin eksikleri GA yardımı ile çözümlenmeye çalışılacaktır. Böylece tek bir yöntem ile harita, şema gibi her türlü taranmış görüntünün vektörizasyonu yapılabilecektir. Aynı zamanda uygulama dâhilinde GA'nın farklı parametre değerleri ve operatörler için farklı teknikler seçilebilecek ve böylece farklı parametre ve farklı operatörler için vektörizasyon sonuçları ve aynı zamanda farklı parametre ve operatörlerin performansı da karşılaştırılabilecektir.

KAYNAKÇA

- [1] Emel, G. G., Taşkın, Ç. (2002) “*Genetik Algoritmalar ve Uygulama Alanları*”, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Vol.21, No.1, pp.129 – 152.
- [2] Koçer, B. ve Yıldız, F., (2006) “*Türev Tabanlı Kenar Çıkarma İle Tam Otomatik Vektörizasyon*”, Selçuk Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Vol.21, No.3 – 4, pp.97 – 106.
- [3] Liu, W., Dori, D., (1999) “*From Raster to Vektor: Extracting Visual Information From Line Drawing*”, Pattern Analysis & Application Vol.2, No.1, pp.10 – 21.
- [4] Kolesnikov, A., (2003) “*Efficient Algorithms For Vectorization and Polygonal Approximation*”, Doktora Tezi, Joensuu Üniversitesi Bilgisayar Bilimleri Bölümü.
- [5] Dharmaraj, G., (2005) “*Algorithms for Automatic Vectorization of Scanned Maps*”, Yüksek Lisans Tezi, Calgary Üniversitesi Geomatik Mühendisliği Bölümü.
- [6] Tombre, K., Tabbone, S., (2000) “*Vectorization in Graphic Recognition: To Thin or not to Thin*”, International Conference on Pattern Recognition – 2, Barcelona, İspanya.
- [7] Karabörk, H., Koçer, B., Bildirici, İ. Ö., Yıldız, F., (2005) “*Yapay Sinir Ağlarının Raster Görüntülerin Vektörel İfadesinde Kullanımı*”, 10. Türkiye Harita Bilimsel ve Teknik Kurultayı, Ankara.
- [8] Jimenez, J., Novalon, J. L., (1982) “*Some Experiments in Image Vectorization*”, IBM Journal of Research and Development, Vol.26, No.6, pp.724 – 734.
- [9] Karaş, İ. R., Bayram, B., Batuk, F., Acar, U., Uzar, M., (2006) “*Doğrusal Çizgilerden Oluşan Raster Görüntülerin Vektörizasyonu için Yeni Bir Yöntem ve 3 Boyutlu CBS’de Kullanılması*”, 1. Uzaktan Algılama – CBS Çalıştay ve Paneli, ISBN: 975 – 561 – 290 – 4.
- [10] Obitko, M., Slavik, P., (1999) “*Visualization of Genetic Algorithms in A Learning Environment*”, Spring Conference on Computer Graphics, SCCG’99, pp.101 – 106.
- [11] Kalaycı, T. E., (2006) “*Yapay Zekâ Teknikleri Kullanan Üç Boyutlu Grafik Yazılımları için ‘Extensible 3D’ (X3D) ile Bir Alt Yapı Oluşturulması ve Gerçekleştirimi*”, Yüksek Lisans Tezi, Ege Üniversitesi Bilgisayar Mühendisliği Bölümü.
- [12] Obitko, (1998) “*Introduction to Genetic Algorithm with Java Applets*”, <http://www.obitko.com/tutorials/genetic-algorithms/index.php> (08.01.2010).

- [13] Williams, W., (1995) “*Genetic algorithms: A Tutorial*”,
<http://web.mst.edu/~ercal/387/slides/GATutorial.ppt> (08.01.2010).
- [14] Russell, S. J., Norvig, P., (1995) “*Artificial Intelligence, A Modern Approach - Second Edition*”, Pearson Education International, USA.

ÖZ GEÇMİŞ

Taner EŞME

www.taneresme.com

Yenişehir Mah. Millet Cad. Yenipetekçiler

Sitesi A17 Pendik/İstanbul

tnr.esme@gmail.com, +90 544 629 77 51

EĞİTİM

2006 -	Okan Üniversitesi Mühendislik – Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü
2003 – 2006	Adıyaman Gölbaşı Lisesi

MESLEKİ İLGİ ALANLARI

Yapay Zekâ
Nesne Tabanlı Tasarım
Algoritma ve Veri Yapıları
Dijital Görüntü İşleme