**T.C.**
**ERCİYES ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİANABİLİM DALI**

# YAPAY ZEKA ALGORİTMALARI İLE KRİPTOANALİZ

**Hazırlayan**
**Arkan Kh Shakr SABONCHI**

**Danışman**
**Prof. Dr. Bahriye AKAY**

**Doktora Tezi**

**Ocak 2020**
**KAYSERİ**

**T.C.**
**ERCİYES ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİANABİLİM DALI**

# YAPAY ZEKA ALGORİTMALARI İLE KRİPTOANALİZ

**(Doktora Tezi)**

**Hazırlayan**
**Arkan Kh Shakr SABONCHI**

**Danışman**
**Prof. Dr. Bahriye AKAY**

**Bu çalışma; Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi Tarafından FDK-2016-7085 kodlu proje ile desteklenmiştir**

**Ocak 2020**
**KAYSERİ**

# BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Arkan Kh Shakr SABONCHI

## YÖNERGEYE UYGUNLUK

Yapay Zeka Algoritmaları ile Kriptoanaliz Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan
Arkan Kh Shakr SABONCHI

Danışman
Prof. Dr. Bahriye AKAY

Bilgisayar Mühendisliği ABD Başkanı
Prof. Dr. Veysel Aslantaş

Prof. Dr. Bahriye AKAY danışmanlığında **Arkan Kh Shakr SABONCHI** tarafından hazırlanan "**Yapay Zeka Algoritmaları ile Kriptoanaliz**" adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

27 / 01 / 2020

**JÜRİ:**

Danışman : Prof. Dr. Bahriye AKAY .....

Üye : Dr. Öğr. Üyesi Serkan Öztürk .....

Üye : Dr. Öğr. Üyesi İlker Dalkıran .....

Üye : Dr. Öğr. Üyesi Ali Gezer .....

Üye : Dr. Öğr. Üyesi Asuman Savaşçıhabeş .....

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 11/02/2020 tarih ve 2020/11-17 sayılı kararı ile onaylanmıştır.

11/02/2020

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

# ÖNSÖZ / TEŞEKKÜR

Doktora eğitimime başladığım ilk günden itibaren ve tez aşaması boyunca her konuda bana yol gösteren ve değerli vakitlerini ayırarak bana yardımlarını esirgemeyen, akademik kariyerimde başarılı olabilmem için örnek aldığım değerli tez danışmanım Prof. Dr. Bahriye AKAY'a teşekkür ederim. Ders dönemi boyunca katkılarından dolayı Prof. Dr. Derviş KARABOĞA'ya ve tez izleme komitesi üyeleri Dr. Öğr. Üyesi Serkan Öztürk'e ve Dr. Öğr. Üyesi İlker Dalkıran'a değerli bilgileri ve katkılarından dolayı teşekkürlerimi sunarım.

Öte yandan bu tez çalışmasına maddi destek veren Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne (Proje No: FDK-2016-7085) teşekkür ederim.

Doktora eğitimimin ilk gününden itibaren beni yalnız bırakmayan ve her daim varlıklarıyla bana ilham veren ancak bir sene önce aramızdan ayrılan rahmetli babam Khaleel SABONCHI başta olmak üzere ailemin tümüne minnettarım.

Akademik kariyerimin ilk gününden itibaren her an yanımda olan ve destek veren sevgili eşim Zainab Hashim OBAİD'e ve hayatımı güzelleştirdiği için oğlum Furkan SABONCHI'ya teşekkürlerimi sunarım.

Arkan Kh Shakr SABONCHI

Ocak 2020, KAYSERİ

**YAPAY ZEKA ALGORİTMALARI İLE KRİPTOANALİZ**

**Arkan Kh Shakr SABONCHI**

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü**
**Doktora Tezi, Ocak 2020**
**Danışman: Prof. Dr. Bahriye AKAY**

# ÖZET

Kriptoanaliz anahtara sahip olmadan okunamaz durumda olan metinlerin çözebilen bir metottur. Olası anahtarların grubu, alfabede bulunan bütün harflerin permütasyonlarından oluşabildiğinden bu permütasyonların sayısı oldukça fazladır. Bundan dolayı, Kaba Kuvvet (Brute Force) denilen bütün yöntemleri deneyerek çözüm arama yöntemi zaman açısından etkili bir yol olmaktan uzaktır. Bu nedenle optimal anahtarı bulmak amacı ile sezgisel yapay zeka algoritmaları tabanlı sistematik arama kullanılabilmektedir. Bu tezin amacı, Genetik Algoritma, Diferansiyel Gelişim, Parçacık Sürüsü Optimizasyon, Yapay Arı Kolonisi gibi yapay zeka optimizasyon algoritmalarının modern şifreleme yöntemlerinin de temeli olan Vigenére şifrelemeye uygulanabilirliğini incelemektir. Bu amaçla, farklı uzunluktaki anahtarlar İngilizce ve Türkçe dilindeki çeşitli metinler üzerinde denenmiştir. Vigenére şifrelemenin kriptoanalizinde temel Yapay Arı Koloni Algoritmasının bölgesel arama kabiliyetinin ve yakınsama hızının iyileştirilmesine gereksinim duyulmuştur. Bu nedenle, popülasyonu daha iyi bir alana yönlendirecek daha yüksek uygunluktaki besin kaynaklarını daha etkin kullanmak için Binom çaprazlama operatörü Yapay Arı Koloni Algoritmasına entegre edilmiştir. Önerilen bu yöntem ile elde edilen sonuçların çok rekabetçi olduğu ve karşılaştırılan diğer yöntemlerinden daha kararlı sonuçlar ürettiği gösterilmiştir.

**Anahtar Kelimeler:** Kriptoanaliz, Vigenére Şifreleme**,** Genetik algoritma, Parçacık Sürü Optimizasyonu Algoritması, Diferansiyel Gelişim algoritması, Yapay Arı Kolonisi Algoritması

**CRYPTOANALYSIS USING ARTIFICIAL INTELLIGENCE ALGORITHMS**

**Arkan Kh Shakr SABONCHI**

**Erciyes University, Graduate School of Natural and Applied Sciences**
**PhD Thesis, January 2020**
**Supervisor: Professor. Dr. Bahriye AKAY**

# ABSTRACT

Cryptanalysis is a method to break the unreadable cipher text without having the key. Beucase the group of potential keys is the group of all potential permutations of all letters (alphabet letter), there are many possible permutations. Therefore, Brute Force which tries all the permutations is not effective in terms of time complexity. For this reason, systematic search, introduced by metaheuristic algorithms, can be used to get the key involved in encoding. The aim of this thesis is to inspect the applicability of meta-heuristic algorithms such as Genetic Algorithm, Particle Swarm Optimization, Differential Evolution and Artificial Bee Colony Algorithms in cryptanalysis of Vigenére cipher which can be used as a basis in modern cryptography methods. In cryptanalysis of Vigenére cipher, an improvement in the local search capability and convergence speed of the Artificial Bee Colony Algorithm. Therefore, Binomial crossover operator that will guide the population to a better area with high quality sources is integrated into the Artificial Bee Colony Algorithm. It is shown that the proposed method produce very competitive results and produce more stable and robust solutions compared to the results of the methods used in the thesis for the analysis of the Vigenére cipher.

**Keywords:** Cryptanalysis, Vigenére cipher, Genetic Algorithm, Particle Swarm Optimization, Differential Evolution, Artificial Bee Colony Algorithm.

# İÇİNDEKİLER

## YAPAY ZEKA ALGORİTMALARI İLE KRİPTOANALİZ

# TABLOLAR LİSTESİ

# ŞEKİLLER LİSTESİ

# GİRİŞ

Günümüz dünyasında güvenli bir iletişim arzu edilmektedir. Yetkili olmayan bir şahıs gönderilen mesaja ulaşıp okuyamamalıdır. Bu tür bir güvenlik kriptoloji bilimi ile sağlanmaktadır. Kriptoloji ile, orijinal metin yerine, şifrelenmiş metine bir anahtar gönderilerken mesajın mahremliği, bütünlüğü ve ulaşılabilirliği sağlanmaktadır. Kriptoloji; kriptografi ve kriptoanaliz gibi birbirini tamamlayan iki tekniği kapsayan, mahrem bilginin korunması amacı ile çeşitli algoritmalar üreterek verileri kodlayıp çözme bilimidir [1]. Kriptoanaliz; anahtarla ilgili önbilgiye sahip olmadan ve iletişim halindeki tarafların izni olmadan şifrelenmiş bir metinden orijinal metni ve/veya anahtarı elde etme süreci ile ilgilidir [1]. Kriptoanaliz bilimi, şifreler üzerinde saldırılar oluşturmak için sürekli olarak yeni metotların geliştirilmesi üzerinde durmaktadır.

Yaygın olarak kullanılan modern şifreleme yöntemleri klasik yöntemleri temel taş olarak kullanmaktadırlar. Aslında karmaşık algoritmaların çoğu, yerine koyma ve yer değişim operatörleri ile şekillenir [2,3].

Klasik şifreleme sistemleri, monoalfabetik ve polialfabetik kripto sistemler olarak iki sınıfa ayrılır [4]. Monoalfabetik bir kripto sistemde her bir karakter, kendisine karşılık gelen sabit bir değişimle şifrelenmektedir. Basit yerine koyma (substitution), her bir harfin sabit olarak değiştiği monoalfabetik kripto sisteme bir örnektir. Polialfabetik şifrelemede, bir alfabe sabit bir yapı olmaksızın farklı alfabeler ile değiştirilebilmektedir. Vigenére şifreleme bunlardan en yaygın kullanılanıdır ve modern şifreleme yöntemlerinin de temel taşı olarak kullanılabilmektedir [2-4]. Vigenére şifreleme de asıl metindeki konumuna göre bir harfin birden çok harfle değiştirilebildiği bir polialfabetik şifreleme yöntemidir [5]. Saldırılarda kullanılan üç temel strateji mevcuttur: (i) sadece şifreli metin (ciphertext-only), (ii) bilinen ana metin (known-plaintext) ve (iii) seçilen ana metin (chosen-plaintext) [6]. Sadece şifreli metin saldırısında, kriptoanalist anahtarı sadece şifreli metinden yararlanarak elde etmek

zorundadır. Bilinen ana metin saldırısında, hem orijinal metin hem de şifreli metin bilinmektedir. Seçilen ana metin saldırısında ise, şifreli metni elde etmek için rasgele seçilen bir ana metin kullanmak gerekmektedir [6]. Simmons [4] asimetrik kripto sistem kavramını ortaya koymuştur. Açık anahtar şifreleme olarak adlandırdıkları yeni bir şifreleme tekniği sunmuşlardır. Bu da, seçilen şifreli metin saldırısı adında yeni bir saldırı türü ortaya çıkarmıştır.

**Tezin Amacı, Kapsamı ve Katkıları**

Olası anahtarların sayısı, 26 harfli alfabenin tüm permütasyonları kadar olduğundan monoalfabetik yerine koyma şifrelemenin uzayı eleman sayısı 26! kadardır, yani 403 trilyon (403*10^42) fazladır. Buna göre, saniyede 1 milyon anahtarı dahi deneyebilecek bile olsak, bu anahtarların tamamını elden geçirmek yaklaşık 12 trilyon yıl sürer. Yani kaba kuvvet metodu ile kriptoanaliz mümkün değildir [7]. Luke, kesin tam sonuç veren algoritmaları kullanmanın çok zahmetli olduğunu belirtmiştir [8]. Klasik şifrelemede, olası anahtarların grubu alfabedeki harflerin permütasyonlarının grubu olduğundan denenmesi gereken anahtar sayısı çok fazladır. Bu yüzden, kaybedilen zaman ve maliyet bir yana, kaba kuvvet ile kriptoanaliz hiç verimli değildir. Bu sebeplerden dolayı, optimal anahtarı bulmak için, optimizasyon algoritmalarından elde edilen bilgiye dayalı sistematik arama yöntemleri kullanılmıştır.

Yapay zeka optimizasyon algoritmaları, bilgisayar ile kabul edilebilir zamanlarda çözülemeyecek bu gibi sorunları kabul edilebilir zamanlarda optimal çözüme yaklaştırmak amacı ile olasılıksal ve yakınsak arama yöntemleri kullanır. Evrimsel algoritma, doğadan esinlenen optimizasyon algoritmalarından biridir. İlk olarak Rechenberg [9] ve Holland [10] tarafından ortaya atıldıktan kısa zaman sonra, günümüzde Genetik algoritma (GA) olarak bilinen hale bürünmüştür. Genetik algoritmalar doğal seleksiyon prensibine dayanan uyarlanabilir sezgisel arama algoritmalarıdır [10]. Parçacık Sürü Algoritmaları (PSO) ise Eberhart ve Kennedy [11] tarafından 1995 yılında geliştirilen kuş ve balık sürülerini modelleyen bir sezgisel optimizasyon yöntemidir. Price ve Storn [12] tarafından 1995 yılında geliştirilen Diferansiyel Gelişim (DE) algoritması popülasyon tabanlı bir evrimsel algoritmadır. Karaboğa [13] tarafından sunulan Yapay Arı Koloni (ABC) algoritması ise bal arılarının kaliteli yiyecek kaynağı arayışındaki topluluk davranışlarından esinlenilerek

geliştirilmiştir. Bu çalışmada, Vigenére şifreleme analizinde ABC, GA, PSO ve DE algoritmalarının Vigenére şifrelemede uygulanabilirliğininin araştırılması ve performans analizlerinin yapılması hedeflenmiştir. Standart ABC algoritmasının yerel arama kabiliyetini artırarak arama uzayında daha etkili arama yapabilmesi için çaprazlama yöntemine dayalı Binom çaprazlama operatörülü Yapay Arı Koloni Algoritması (BCABC) önerilmiştir.

Yapay zeka optimizasyon algoritmalarının kripto sistemlere uygulanmasında, algoritmayı araştırma uzayında yönlendirebilecek anahtar uygunluk değerinin hesaplanması gerekmektedir. Bu amaçla, anahtarın geçerliliğini değerlendirmede frekans temelli bir uygunluk (fitness) fonksiyonu kullanılır. Frekans analizi şifresi çözülmüş metindeki frekanslar ile kullanılan dilin literatüründeki frekansları karşılaştırır. Frekans analizi, çok düşük bir hesaplama maliyeti ile elimizdeki şifrelenmiş metin ile dile ait bütün sözcüklerin frekans uzayında karşılaştırılmasını sağlar. Ancak bu analizin anlamlı olması için kriptoanaliz uygulanan mesajın yeterli uzunlukta olması beklenmektedir [7,14].

**Araştırmanın Önemi**

Bu alanda Türkçe metinler üzerinde daha önce yapılan çalışma bulunmamaktadır. Algoritmaların kontrol parametre değerlerinin performans üzerinde etkisi olduğundan her bir algoritma için parametre analizinin yapılması önemlidir. Bu nedenle çalışma kapsamında ilk olarak elde edilen parametre değerleri çalışmanın devamında kullanılmıştır ve araştırmacıları için de tavsiye niteliğindedir. Çalışma farklı uzunluklardaki hem Türkçe ve İngilizce metinler üzerinde gerçekleştirildiği için yapılan işlemlerin dile bağlılığını da göstermektedir. Bu çalışmada dört farklı optimizasyon algoritmasının özelikleri incelenip olumlu taraflarını koruyarak olumsuz taraflarını gidermek üzere Binom çaprazlama operatörlü bir algoritma önerilmiştir. Böylece kriptoanaliz için etkin ve yeni bir algoritmanın ortaya koyulması daha sonraki çalışmalar için de bir altyapı oluşturacaktır.

**Tezin Organizasyonu**

Tezin organizasyonu şu şekildedir. Bölüm 1'de Kriptoloji konusu ile ilgili genel tanımlamalar yapılarak kriptografi ve kriptoanaliz yöntemleri tanıtılmaktadır.

Bölüm 2'de yapay zeka optimizasyon kavramı verilerek optimizasyon algoritmaları ve sürü zekasına dayalı algoritmalar ilgili temel tanımlamalar verilmektedir. Bölüm 3'te optimizasyon algoritmaları ile kriptoanaliz konusu ile ilgili literatür taraması verilecek, optimizasyon algoritmalarının bu alana uygulanması anlatılacaktır. Tez kapsamında önerilen Binom çaprazlama operatörlü Yapay Arı Kolonisi algoritması da bu bölümde anlatılacaktır. Bölüm 4'te optimizasyon algoritmalarının kriptoanaliz problemleri üzerinde test edilmesi esnasında kontrol parametrelerine olan duyarlılığı incelenecektir. GA, PSO, DE, ABC ve BCABC algoritmaları Vigenére şifrelemenin analizine uygulanacaktır ve metin uzunluğu ile anahtar uzunluğuna göre karşılaştırılacaklardır. BCABC algoritması kontrol algoritması olarak seçilerek GA, PSO, DE, ABC algoritmaları ile hipotez testi aracılığı ile istatistiksel kıyaslama yapılacaktır. En son bölümde ise çalışmada varılan sonuçlar tartışılacak ve gelecek çalışmalar hakkında bilgi verilecektir.

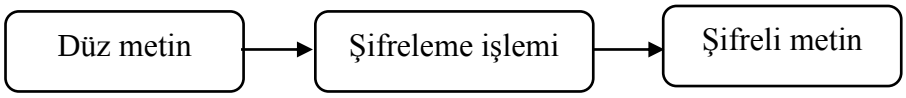
# 1. BÖLÜM

# KRİPTOANALİZ

Bu bölümde girişte söz konusu olan bazı teknik terimler ve söz konusu problem hakkında bilgiler verilmektedir.

## 1.1. Şifreleme

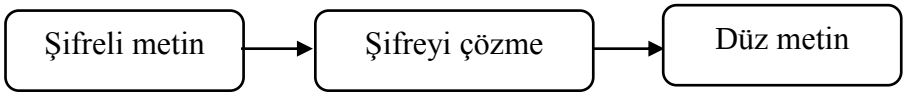
Yunancadaki *kripto (saklanmış/gizli)* ve *logos (söz, bilgi, bilim)* kelimelerinden türetilen kriptoloji bilimi ile özel bilgiler şifrelenmekte ve daha önce şifrelenmiş bilgiler çözümlenebilmektedir. [1-3]. Kriptoloji, kriptografi (şifreleme) ve kriptoanaliz olmak üzere bilimi iki alt bilim dalına ayrılır. Kriptografi, bilgi güvenliğinde gizlilik, kimlik denetimi, bütünlük gibi kavramların sağlanması amacıyla matematiksel yöntemlerin tümüne verilen addır. Bu yöntemlerin amacı, bilginin olduğu herhangi bir ortamda meydana gelebilecek saldırılardan bilginin kendisini, göndericisini ve alıcısını korumaktır [1-3]. Kriptografi, Şekil 1.1'de gösterildiği gibi mevcut bir metnin güvenliğini sağlamak için anahtar bilgi üreterek onu okunamaz hale getirir. Bu anahtar, metni şifrelemek veya şifrelenen metni tekrar düz metin haline getirmek için kullanılan harf, sayı, kelime, sembol, fonksiyon (veya bunların kombinasyonu) gibi bir veri parçası olabilir [1-3].




*Şekil 1.1.* Şifreleme işlemi

Kriptoanaliz, Şekil 1.2'de gösterildiği gibi şifreli durumda olan bir metinden düz metni ortaya çıkarmak amacıyla uygulanan tüm yöntemlere verilen genel addır. Kriptoanalizde, kullanılacak anahtar bilinmeden şifrelenmiş olan mesaj çözülmeye çalışılır. Basit şifre sistemlerini çözme işi, olası tüm alfabeleri denemek yerine, daha

kolay şekilde çözebilmek için matematik, istatistik ve dilbilim alanlarında yeterli bilgiye sahip olmak gerektirir [1-4,13,15-20].




*Şekil 1.2.* Şifre çözme işlemi

Klasik şifreleme sistemleri, monoalfabetik ve polialfabetik kripto sistemler olarak iki sınıfa ayrılır [4]. Monoalfabetik bir kripto sistemde her bir karakter, kendisine karşılık gelen sabit bir değişimle şifrelenmektedir. Polialfabetik şifrelemede, bir alfabe sabit bir yapı olmaksızın farklı alfabeler ile değiştirilebilmektedir.

### 1.1.1 Monoalfabetik Şifreleme

Monoalfabetik şifreleme ile açık metindeki orijinal karakterler kimliklerinden uzaklaşırlar, fakat bulundukları yer (konumları) değişmez. Yani açık metindeki karakterlerin yerleri sabittir, karakterleri şifrelemek için başka bir alfabenin karakterleri ya da sayılar kullanılır [21,22]. Düz ve şifreli metin harfleri arasında bire bir ilişki vardır. Düz metindeki her bir karakter şifreli metinde karşılığı tektir ve bu ilişki metin boyunca değişmez. A, $\{a_0, a_1, \dots, a_{n-1}\}$ *n* karakterli bir alfabe kümesi ve C, $\{f(a_0), f(a_1), \dots, f(a_{n-1})\}$ şeklinde *n* karakterli bir alfabe kümesi olsun. Düz metin harfleri D ve şifreli metin E(D) olarak temsil edilsin. *f*: A→C , A kümesi ile C kümesi arasında birebir bir fonksiyondur [7]. Buna göre düz metin $D = d_1\ d_2\ d_3 \dots d_n$ şifreleme işlemi ile $E(D) = f(d_1), f(d_2) \dots f(d_n)$ şekline gelir. Yani, her bir harf kendisine karşılık gelecek sabit bir kural ile şifrelenir.

### 1.1.2. Polialfabetik Şifreleme (Vigenére Şifreleme)

Bu yöntemde, düz metin içerisindeki her bir harfe karşılık gelen bir şifre alfabe vardır. Anahtar kelime, şifre alfabenin belirlenmesinde belirleyici unsurdur. Farklı olarak, oluşturulan her bir anahtara karşılık farklı şifreli metinler meydana gelir. Böylece tek alfabeli şifreleme yöntemlerinde olduğu gibi kolay bir şekilde çözülme durumu ortadan kalkmış olur. Bilim adamı Bliase de Vigenére'in ismi ile anılan yöntem poliafabetik şifreleme yöntemlerinden en popüler olanıdır. Vigenére şifrelemede Türkçe alfabe için Şekil 1.3'deki gibi bir Vigenére tablosu kullanılır. Başlangıçta Vigenére şifreleme yönteminde yalnızca *Vigenére Square* diye adlandırılan tablo kullanılmakta idi. Burada

harflerin yerleri şifreleme ve şifre çözme işlemlerinin her biri için de değişmemekteydi [23]. Şifrelemede kullanılmak üzere bir anahtar kelime seçilir ve düz metindeki karakter sayısı kadar tekrar edilir. Anahtar kelimenin harfleri Vigenére tablosunun sol sütununda bulunur. Benzer şekilde düz metnin karakterleri de üst satırda bulunur. Ardından tablo içinde bu iki harfin kesiştiği yerde bulunan karakter şifreli metinde karşılık gelen karakteri verir. Bu şekilde karakterlerin her biri eşleştirilip yerlerine konduğunda şifreli metnin tamamı elde edilmiş olur [1,6].

| | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z |
| B | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A |
| C | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B |
| Ç | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C |
| D | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç |
| E | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D |
| F | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E |
| G | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F |
| Ğ | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G |
| H | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ |
| I | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H |
| İ | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I |
| J | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ |
| K | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J |
| L | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K |
| M | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L |
| N | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M |
| O | O | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N |
| Ö | Ö | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O |
| P | P | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö |
| R | R | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P |
| S | S | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R |
| Ş | Ş | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S |
| T | T | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş |
| U | U | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T |
| Ü | Ü | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U |
| V | V | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü |
| Y | Y | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V |
| Z | Z | A | B | C | Ç | D | E | F | G | Ğ | H | I | İ | J | K | L | M | N | O | Ö | P | R | S | Ş | T | U | Ü | V | Y |

*Şekil 1.3.* Türkçe alfabe için oluşturulmuş Vigenére Tablosu

Vigenére şifreleme yönteminde, anahtar kelimede bulunan harflerin sayısal karşılıkları ele alınır. Bunlar düz metin içerisinde bulunan karakterlerin yer değiştirme miktarını verir. Bu işlem matematiksel olarak şöyle ifade edilebilir [5]. Anahtar kelime= ($a_1$ , $a_2$, … ,$a_i$) ve düz metin D= ($d_1$, $d_2$, … , $d_i$) olduğunda $f_i(d_i) = (d_i + a_i)$ mod *29* olur.

## 1.2. Saldırı Teknikleri / Kriptoanaliz Tekniklerinin Sınıflandırılması

Kriptoanaliz, kriptolu (şifreli) bir metinden yola çıkarak, bunu oluşturan düz metine ulaşabilmek amacıyla yapılan çalışma, uygulama ve analizlerin tümüne verilen isimdir. Bu çalışmaların her birine birer şifre çözme saldırısı veya atağı denir. Bu saldırı yöntemlerinden bazılarına aşağıda yer verilmektedir [1,6,24].

### 1.2.1. Sadece Şifreli Metin Saldırısı

Sadece şifreli metin saldırısı durumunda, kriptoanalist mesajın içeriği ile ilgili hiçbir bilgiye sahip değildir. Sadece şifreli metinden yola çıkarak ve birtakım tahminler yürüterek düz metini elde etmeyi amaçlar. Klasik saldırıların büyük bir kısmında harfin şifreli metin içerisindeki frekansı kullanılarak işlem yapılmaya çalışılır. Bu yöntem aslında şifrelenmiş yapıların kırılmasıyla ortaya çıkmıştır. Herhangi bir anahtar olmadığında, şifrelenmiş bir metni çözmek, ancak bütün şifre alfabelerin denenmesi sonucunda olabilir. Bu da teknik olarak olanak dışıdır. Bunu mümkün hale getirmek veya daha da kolaylaştırmak için, matematik, istatistik ve dilbilim alanlarından yararlanmak gerekir. Al-Kindi'nin *Kriptografik Mesajların Deşifresi* isimli eserine [17,18,25] göre, hangi dilde yazıldığı bilindikten sonra, şifreli bir mesajı çözmek için, o dilde yazılmış uzun bir metin kullanırız. Bu uzun metinde her harfin kullanım sıklığı (frekansı) hesaplanır. Böylece düz metinde en sık kullanılan harf şifreli metinde en sık kullanılan harfe denk geldiği kabul edilerek şifre çözülmeye çalışılır. Bu birebir eşleme sırasıyla tüm harflerle tekrar edilir. Bu işlem şifre metinde en çok kullanılan harften en az kullanılan harfe doğru yapıldığı zaman, kriptoanalizde başarılı olma olasılığı yükselir [26]. Neticede orijinal metindeki harfler, yani şifresi çözülmüş metin elde edilir. Bu kriptoanaliz tekniğine frekans analizi ismi verilmektedir. Frekans değeri küçük sapmalar gösterse de harflerin frekanslarının kendi aralarındaki öncelik sıraları genellikle fazla bir değişiklik göstermez [17,18,27]. Aynı alfabe kullanılsa da, dil değişikliği durumunda bu frekanslar değişir. Örneğin Latin alfabesini kullanan Türkçe ve İngilizce dilleri karşılaştırıldığında, Türkçe'de en çok kullanılan harfin A, İngilizce'de ise E olduğu görülür.

### 1.2.2. Bilinen Düz Metin Saldırısı

Bu saldırı türünde düz metnin tamamı veya bir kısmına, ayrıca şifreli metne sahip bulunulmaktadır. Eldeki tüm bu bilgiler kullanılarak metin bloklarını ya da çözülmemiş olan kısımları çözebilir. Bilinen düz metni elde etmek için, en yaygın olarak kullanılan yöntem bu şifrelenmiş bloklara karşılık lineer (doğrusal) bir kriptoanaliz metodu geliştirmektir. Burada amaç, şifreleme anahtarını elde etmektir [1,6,24].

### 1.2.3. Seçilmiş Düz Metin Saldırısı

Bu saldırı metodunda amaç şifrelemede kullanılmış olan anahtarı ortaya çıkarmaktır. Günümüzde RSA olarak bilinen kripto sistemler, seçilmiş düz metin saldırısına karşı savunmasızdır [1,6,24]. Kriptoanalistlerin şifreleme algoritmasını bilmesi durumunda, kaba kuvvet *(brute force)* ile saldırıya geçmesi beklenebilir. Bu saldırıda olası tüm anahtarları denedikten sonra sonuca ulaşmaya çalışır. Bu yöntemin oldukça zaman alması ve yüksek maliyetli olmasından dolayı istatistiksel araçlardan yararlanılabilir [1,6,24].

### 1.2.4. Seçilen Şifreli Metin Saldırısı

Bu tür saldırılarda, kriptoanalist eldeki şifreli bir metini ve bu şifreli metine karşılık gelen bir düz metini bulmaya çalışır. Bu saldırıda, özel bir cihaz sayesinde şifre çözme olayı gerçekleşir ve herhangi bir anahtara da ihtiyaç yoktur [1,6,24].

### 1.2.5. Uyarlanır Seçili Düz Metin / Şifreli Metin Saldırısı

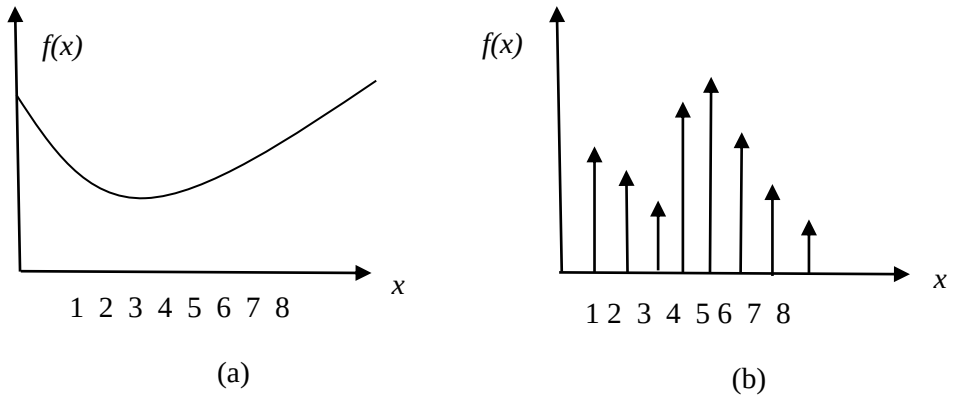
Bu saldırı türü ise, daha önce bahsettiğimiz seçilen düz metin ve seçilen şifreli metin saldırıları ile benzerlik gösterir. Buradaki temel fark, metinler rastgele seçilmek yerine, daha önceki deşifre işlemlerinde elde edilen bilgiler kullanılarak seçilir [1,6,24].

# 2. BÖLÜM

# YAPAY ZEKA OPTİMİZASYON ALGORİTMALARI

## 2.1. Optimizasyon

Optimizasyon probleminde amaç belli kısıtlar altında, belirlenen bir amaç fonksiyonunu en iyileyen tasarım değişkenlerinin değerlerine karar verilmesidir. Problemdeki tasarım değişkenleri kabul edilebilir bölge içinde tüm değerleri alabiliyorsa *sürekli optimizasyon problemi* (Şekil 2.1.a), sadece belirli değerler alabiliyorlarsa *ayrık optimizasyon problemi* (Şekil 2.1.b) olarak isimlendirilir.




*Şekil 2.1.* (a) Sürekli araştırma uzayı, (b) Ayrık araştırma uzayı.

Bir optimizasyon problemi Eşitlik (2.1) ile ifade edilebilir. Burada $f(x)$ minimize edilecek amaç fonksiyonu ve $x_1, x_2, \dots, x_n$de değeri belirlenmeye çalışılan tasarım parametreleri vektörüdür.

$$Min. f\ (x), x = (x_1, x_2, \dots., x_n) \tag{2.1}$$

Tasarım değişkenlerinin belirli değerler arasında kalabilmesini veya belirli değerleri alamamasını sağlayan probleme özgü eşitlik (Eşitlik (2.2)) veya eşitsizlik (Eşitlik (2.3))

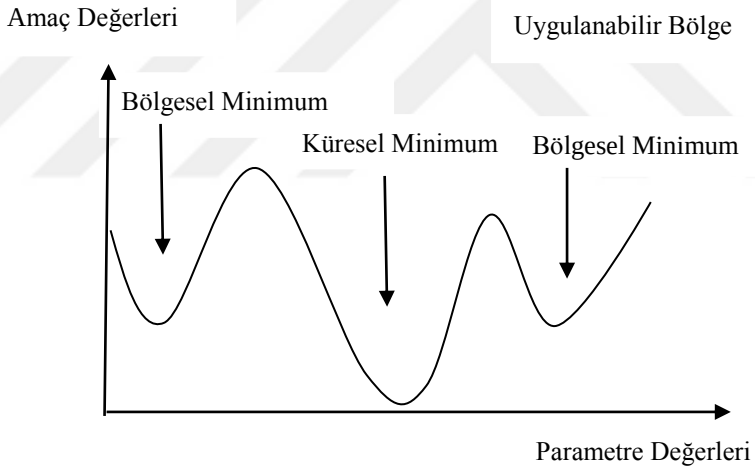
ifadelerine sınırlama adı verilir. Bu şekilde kısıtlara sahip problemlere *sınırlamalı optimizasyon* problemleri ismi verilir. Sınırlamaya sahip olmayan problemlere ise *sınırlamasız optimizasyon* denir. Sınırlamalı optimizasyon probleminde problemdeki bütün kısıtlarını sağlayan değerler kümesine *kabul edilebilir bölge* denir.

$$h_i(x) = 0, \quad i = 1, 2, 3, \dots, n \tag{2.2}$$

$$g_j(x) \leq 0, \quad j = 1, 2, 3, \dots, l \tag{2.3}$$

$$x \in R^n$$

Amaç fonksiyonunun olası çözümleri içinde belli bir komşulukta en iyi değeri veren nokta *bölgesel (yerel) optimum,* kabul edilebilir bölgedeki bütün değerlerden içinden en iyi çözüm ise *küresel (genel) optimum* (optimum çözüm) olarak tanımlanır (Şekil 2.2).




*Şekil 2.2*. Bölgesel minimum ve küresel minimum kavramları.

Optimizasyon probleminin uygulanabilir bölge içinde bir bölgesel optimum noktaya takılmasına ise *erken yakınsama* denir. Optimum noktada, gerek şartlar ve yeter şartlarda sağlanıyorsa o noktanın optimum çözüm noktası olduğu söylenebilir [28-30].

## 2.2. Optimizasyon Metotları

Optimizasyon yöntemleri, klasik teknikler ve modern sezgisel teknikler olmak üzere iki ana kategoride gruplandırılabilir (Şekil 2.3). Klasik optimizasyon teknikleri, sürekli ve türevlenebilir fonksiyonlar içeren optimizasyon problemlerinin çözümlerini bulmada etkili metotlardır. Buna karşın, genelde pratik uygulamalarda amaç fonksiyonu

türevlenebilir olmadığından, hatta birçok zaman sürekli bile olmadığından bu tekniklerin kullanılabileceği alan oldukça sınırlı kalmaktadır.

### 2.3. Modern Sezgisel Metotlar

Sezgisel metotlar sınırlı bir optimizasyon problem grubunun çözümüne göre değil çok çeşitli bir problem uzayına etkili bir şekilde çözüm arar. Sezgisel olarak adlandırılan algoritmalarda araştırma uzayı oldukça geniş olduğundan, çözümün bulunabilmesi için bazı kural ve çözüm stratejileri geliştirilir. Belirlenen hedefe ulaşmak için bazı kısıtlar tanımlanır ve bu kısıtlar içerisinden elde edilen çözümlerden en etkili olanı seçilmeye çalışılır. Klasik tekniklerin fonksiyonunun sürekli ve türevlenebilir olmasını gerektiren sınırlı kullanımından dolayı, sezgisel yöntemlerle ilgili çalışmalar büyük hız kazanmıştır. Bunun başlıca sebepleri aşağıdaki gibi sıralanabilir [31]:

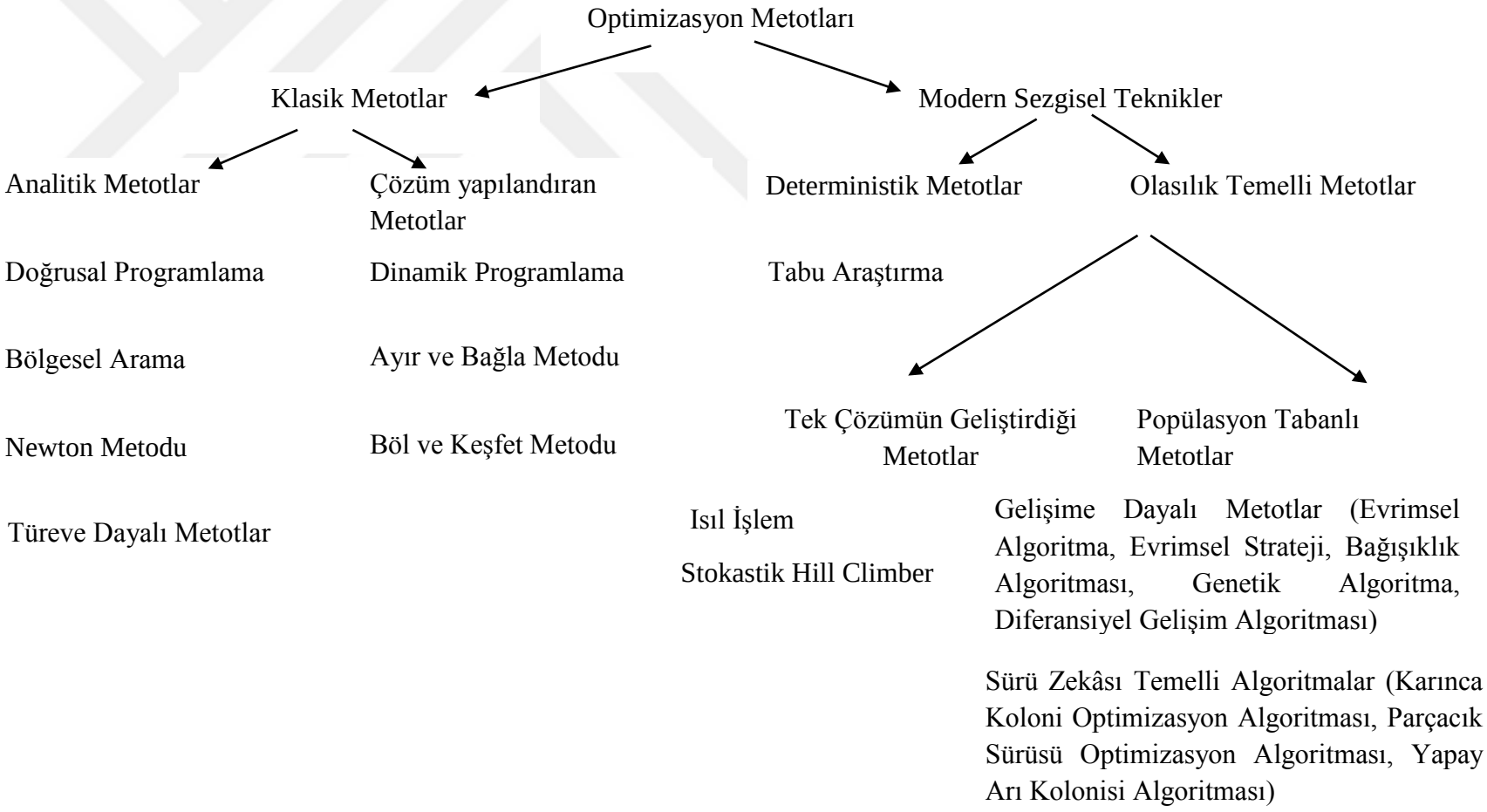
1. Bu yeni metotlar, esneklikleri sayesinde yapılarında fazla bir değişikliğe ihtiyaç duymaksızın birçok probleme uyarlanabilir.
2. Ana yapıları itibarı ile basittirler ve türev gibi ağır matematiksel işlemlere ihtiyaç duymazlar.
3. Kullanım kolaylıklarından, herhangi bir uzmanlık gerektirmeyen bir şekilde çok çeşitli problemin çözümüne uygulanabilirler.
4. Çoklu başlangıçlı bir yapıları vardır; yani aynı anda birçok farklı çözümden yola çıkarak çözüm uzayını tarayıp paralel hesaplamaya imkan verirler.
5. Çevrelerindeki değişimlere kolay ve hızlı bir şekilde adapte edilebilirler.
6. Sezgisel yöntemler kesin çözümü vermese bile, onu elde etmek için atılan adımlardan birisi olarak kullanılabilirler.

Genetik Algoritmalar, Evrimsel Programlama, Evrimsel Stratejiler, Genetik Programlama gibi türleri olan evrimsel hesaplama teknikleri en iyi olan hayatta kalır kuralı üzerine kurulmuştur. Evrimsel metotlar, tek bir başlangıç çözümü kullanmak yerine popülasyon ismi verilen bir çözüm topluluğu oluşturup araştırma uzayında arama yapar. Her adımda bazı genetik operatörler (kısıt veya kriterler) kullanılmasıyla yeni bireyler ve nesil oluşturulur [32].

Sürü zekası bireylerin veya organizmaların bir sorunu çözmek için merkezi bir kontrol olmaksızın birlikte yaşadıkları kolektif deneyim ve bilgiyi esas alır. Bir karıncanın geçtiği yollara feromon bırakarak,. diğer karıncaların yiyecek kaynağını veya yuvalarını bulmalarına yardımcı olması bu duruma örnek verilebilir. Bu tür varlıkların hayatlarını bir arada sürdürebilmeleri için yaptıkları iş bölümü, bilgi paylaşımı vb. gibi akıllı davranışları araştırmacıların ilgisini çekmiş ve gerçek dünya sorunlarını çeşitli sürülerdeki bu davranışlara dayanan yöntemlerle çözmeye çalışmışlardır. Karıncaların feromon bırakma davranışları Karınca Kolonisi Optimizasyon Algoritmasının, kuşların ve balıkların sürü halindeki sosyal davranışları Parçacık Sürü Optimizasyonunun ve bal arısı kolonilerinin davranışı da Yapay Arı Kolonisi Algoritmasının temelini oluşturur. Sürü zekasına dayalı metotlarda araştırma uzayının verimli bir şekilde keşfedilmesini sağlayacak, mevcut keşfedilmiş lokasyonlar civarında detaylı araştırma yapabilecek konum güncelleme teknikleri / kuralları bulunur [32].

**Gösterim**: Optimizasyon probleminin türüne bağlı olarak tasarım değişkenlerinin temsili için ikili kodlama, tamsayı kodlama, ağaç gösterimi, sürekli kodlama formların kullanılabilir. Bu kodlamalara amaç fonksiyonu öncesinde veya operatörler uygulanmadan önce çeşitli dönüşümler uygulanabilir [33].

**Popülasyon:** Birden fazla çözümden oluşan gruba popülasyon denir. Popülasyon, çeşitli kromozomlar, parçacıklar, yapay karınca ve arılar veya besin kaynaklarından oluşabilir. Başlangıç yoğunluğu oluşturmanın farklı yolları mevcuttur. Problemin başlangıçta çözümü hakkında genel bir fikre sahip olunması durumda, başlangıç popülasyonu bu çözümlerden oluşturulabilir. Bu şekilde, optimal çözümün aranmasında zamandan tasarruf edilmiş olur. Çözüm hakkında başlangıçta hiçbir fikir yoksa, başlangıç popülasyonu genelde rasgele bir algoritma kullanılarak oluşturulur. Popülasyon büyüklüğünün az olması, yetersiz örneklemeye (birey çeşitliliğine) sebep olacak ve araştırma yerel bir optimal noktaya sürüklenecektir. Popülasyon büyüklüğü çok büyük seçilirse, araştırma çok büyük bir zaman gerektirebilecektir. Bu nedenlerden, probleme uygun bir başlangıç popülasyon büyüklüğünün belirlenmesi önemlidir [34, 35]. Bir birey arama yaparken popülasyondaki başka bir bireyin bilgisinden faydalanabilir. Popülasyon yeni çözümler üretemez (durgunluk) olduğunda veya yerel minimumlara takıldığında bu sorunların üstesinden gelebilmek için algoritmalar çeşitli keşif ve farklılık mekanizmaları içermelidir [33].




*Şekil 2.3.* Optimizasyon metotlarının sınıflandırılması [36,37]

**Uygunluk Fonksiyonu:** Maliyet fonksiyonu (Uygunluk fonksiyonu), büyük bir arama alanı içinde en iyi çözümleri elde etmek için sezgisel bir algoritmada çok önemli bir rol oynamaktadır. Bir uygunluk fonksiyonunun amacı, belirli bir çözüm için anlamlı, ölçülebilir ve karşılaştırılabilir bir değer sağlamaktır [32]. Bu kıyas neticesinde araştırma uzayında aramanın ne tarafa kayacağı da belirlenir.

**Çözüm Seçme Mekanizması**: Araştırmayı sürdürmek için aday çözümlerin ya da ebeveyn çözümlerin belirlenmesi, yerel arama yapmak için komşu çözüm seçilmesi gerekir. Bunun için seçim mekanizmaları uygulanır. Bu seçim mekanizmaları, çözümün kalitesine bağlı olabilir veya rastgele olabilir [33].

**Değişim Operatörleri:** Mevcut bir çözümden daha iyi bir çözüm elde etmek amacıyla yeni bir çözüm üretmek için çaprazlama yöntemleri ve konum güncelleme ifadeleri kullanılabilir. Değişim operatörlerinde güncel en iyi çözüm, bireyin elde ettiği en iyi çözüm vb. değerler kullanılabilir. Bireylerin farklılaşmasını sağlamak için çeşitliliğe dayalı yöntemler de kullanılır [32].

**Hayatta Kalanı Seçme Mekanizması:** Doğal yaşamdaki seleksiyon gibi popülasyonunun gelişimi esnasında bazı bireyler hayatta kalırken bazı bireyler elenirler. Bu seçilim bireylerin kalite (uygunluk) değerlerine göre veya yaş değerine göre gerçekleşebilir [33].

**Durdurma Kriteri:** Uygun bir durdurma kriteri için iki koşul belirtilebilir. Eğer problem bilinen bir optimal değer seviyesine sahip ise, yani, hedef fonksiyonun bilinen bir optimum değerine ulaşıldığında algoritma durdurulabilir ($\epsilon > 0$). Ancak, sezgisel algoritmaların stokastik olduğu ve optimum bir erişim garantisi olmadığı için bu koşul hiç sağlanamayabilir. Bu durumda aşağıda listelenenler gibi ek bir koşul ya da koşullar dahil edilebilir [38,39]:

1. İzin verilen maksimum CPU zamanının dolması,
2. Toplam uygunluk hesaplamalarının belirtilen sayıya ulaşması,
3. Belirli bir zaman dilimi boyunca, uygunluk değerinin belirli bir eşik değeri altında seyretmesi,
4. Popülasyon çeşitliliği belirli bir eşiğin altında kalması.

## 2.4. Tez Çalışmasında Kullanılan Sezgisel Algoritmalar

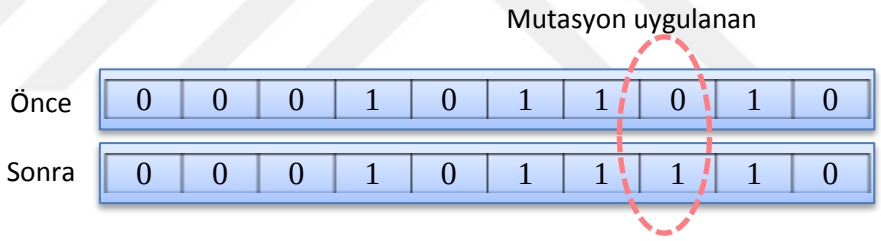
Bu tez çalışmasında Rechenberg [9] ve Holland [10] tarafından geliştirilen, Genetik Algoritma (GA), Kennedy ve Eberhart tarafından geliştirilen Parçacık Sürü Algoritması (PSO) [11], Storn tarafından geliştirilen Diferansiyel Gelişim Algoritması [12] ile Karaboga tarafından geliştirilen Yapay Arı Koloni Algoritması (ABC) [13] analizlerde kullanılmıştır. Takip eden alt bölümlerde algoritmaların kısa tanımalarına yer verilecektir.

### 2.4.1. Genetik Algoritma

Genetik algoritma, J. Holland tarafından [10] doğal seleksiyon prensibinden esinlenerek geliştirilen evrimsel hesaplama tekniklerinin bir parçasıdır. Genetik algoritmanın ana hatları ile temel adımları aşağıdaki gibidir:

1. Başlangıç parametrelerinin değerlerinin belirlenmesi (Karar değişkenlerinin sayısı, Değişkenlerin alt sınırı, Değişkenlerin üst sınırı, Azami iterasyon sayısı, Popülasyon büyüklüğü, Çaprazlama oranı, Mutasyon oranı)
2. Popülasyonun başlatılması: *n* bireyli bir popülasyon oluşturacak şekilde rasgele çözümlerin (kromozomlar) oluşturulması.
3. Değerlendirilme: Popülasyondaki her bir kromozomun uygunluk değerinin hesaplanması
4. **Tekrar et** (Herbir kromozom için)
   Aynı büyüklükte bir popülasyon oluşana kadar aşağıdaki adımları tekrar et:
   a. Doğal seçim: popülasyon içinden, daha yüksek uygunluk değere sahip kromozomların daha yüksek seçilme ihtimali olacak şekilde iki kromozomun (ebeveyn) seçilmesi
   b. Çaprazlama: [0,1] aralığında rasgele bir r değerinin seçilmesi, r <pc ise çaprazlama uygulanması
   **c.** Mutasyon: [0,1] aralığında rasgele bir r değerinin seçilmesi, r <pm ise iki rasgele seçilen noktada genlerin (karakter) değiştirilmesi.
5. Yeni oluşturulan kromozomun (çözümün), güncellenmesi (uygunluk değerinin hesaplanması) ve en iyi sonucu kaydedip bir sonraki popülasyona dahil edilmesi.
6. **Koşullar sağlanana kadar**

GA'da öncelikle bir başlangıç popülasyonu üretilir ve bu popülasyonun her üyesi kromozom olarak isimlendirilir. Bu kromozomlar elimizdeki çözümlerinden birini temsil eder. *İkili kodlama* ya da *gerçek değerli kodlama* ile temsil edilebilirler. Kromozomlar üreme adı verilen çeşitli metotlarla çoğaltılır ve geliştirilirler. Rulet çarkı yöntemi, turnuva yöntemi ve derecelendirmeli (ardışık) seçim gibi yöntemlerle seçilebilen iki ebeveyn kromozomun çaprazlama operasyonu veya mutasyon operasyonu sonucunda çocuk olarak adlandırılan yeni kromozomlar oluşturulurlar. Yüksek çaprazlama oranı, bireylerin çok fazla kopyalanmasına yol açacaktır. Düşük çaprazlama oranı, araştırmanın ilerlemesini ve bilgi aktarımını yavaşlatacaktır [40]. Mutasyon işlemi; temel olarak yeni oluşan kromozomların bazı genlerinde rastgele bir değişimdir. Şekil 2.4, ikili kodlama için basit bir mutasyon sürecini göstermektedir. Yüksek oranda mutasyon, araştırmaya aşırı bir rassallık kazandıracak ve ıraksamayı (çeşitlendirmeyi) hızlandıracaktır. Öte yandan, düşük bir mutasyon oranı ise, ıraksamayı yavaşlatır ve araştırma alanının tam olarak araştırılmasını engel olur [41].




*Şekil 2.4.* İkili kodlamada mutasyon uygulaması

Yeni üretilen çözümlerin uygunluk değerlerine bakıldıktan sonra, yeni ve eski nesil kromozomlardan bir çözüm grubu oluşur. Doğal seleksiyon olarak da bilinen doğal seçim; bir sonraki jenerasyonu oluşturmak için uygunluk değerlerine göre kromozomların seçilmesidir. Bu mekanizma, yüksek kaliteye sahip bireylerin hayatta kalma ilkesi üzerine yürümektedir. Bireyler bir nesilden diğerine geçerken kalite değerlerine göre seçilme şansına sahiptir. Eski neslin en iyi kromozomunun yeni nesildeki bir kromozomla değiştirilmesi ise elitizim olarak adlandırılır [42,43].

### 2.4.2. Parçacık Sürüsü Optimizasyon (PSO) Algoritması

1995 yılında Kennedy ve Eberhart [11] tarafından kuşların veya balık sürülerinin davranışlarından esinlenerek önerilen popülasyona dayalı bir optimizasyon algoritmasıdır. Algoritmada olası her çözüm parçacık olarak isimlendirilir. Bu

parçacıkların oluşturduğu popülasyon sürü adını alır ve birlikte hareket ederek çözümü arayan bir yapı oluşturur. Popülasyonun büyüklüğü bu çözümün bulunma zamanını ve arama katilesin doğrudan etkilemektedir. Metot esas olarak parçacıkların besin kaynağına doğru hareketine dayanmaktadır. Hareketin temeli hızdır ve parçacıkların hızının hesaplanmasında iki temel faktör vardır. Birinci faktör, bireyin yer aldığı sürüden etkilenmesini sağlayan sosyal bileşendir ve bu bileşen parçacığın sürü içinde en iyi çözüme sahip noktaya doğru hareket etmesine izin verir. İkinci faktör, bireyin kendi tecrübesidir ve sınırlı bellekte sonraki iterasyon için en iyi çözümü saklar. PSO algoritmasında popülasyon, sürüdeki bireyler olan parçacıkların konumlarına karşılık gelmektedir. Başlangıçtaki popülasyona rastgele parçacık konumlar atanır ve her bir parçacığın kendine özgü hız bilgisi vardır. Yine bu hız ve yön bilgisi, başlangıç popülasyonunun her bir parçacığı için rasgele üretilir [11, 44, 45]. PSO algoritmasının kod taslağı özet bir şekilde aşağıda verilmektedir.

1. Parametrelerin başlatılması (*w*: atalet ağırlığı, $C_1$ ve $C_2$: bilişsel ve sosyal hız katsayıları).
2. Eşitlik (2.4) ile her bir parçacık için rasgele bir konum üretilmesi ($x_i$):

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \tag{2.4}$$

3. **Tekrarla**
4. (Her bir parçacık için)
   - **a.** Uygunluk fonksiyonunun değerlendirmesi
   - **b.** Uygunluk değeri, parçacığın o ana kadarki en iyi uygunluk değerinden daha da iyiyse, en iyi konumun ($pb_i$) güncellenmesi
5. Sürü içinde en iyi uygunluk değere sahip parçacığın (*gb*) belirlenmesi
6. Her bir parçacık için:
   - **a.** (0,1) aralığında iki rasgele sayının ($r_1$ ve $r_2$) üretilmesi ve Eşitlik (2.5) ile hızın güncellenmesi:

$$v_i^{t+1} = w.v_i^t + C_1.r_1(pb_i - x_i^t) + C_2.r_2(gb - x_i^t) \tag{2.5}$$

     Burada $x_i^t$ , *i*. parçacığın *t*. iterasyonundaki konumu, $v_i^t$ ise *i*. parçacığın *t*. iterasyonundaki hızıdır.
   - **a.** Eşitlik (2.6) ile konumun güncellenmesi:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2.6}$$

7. **(Koşullar sağlanana kadar)**

Kennedy ve Eberhart tarafından önerilen PSO'nun atalet ağırlık faktörü bulunmamakta idi. 1998 yılında yayınlanan makalelerinde, Eberhart ve Shi [44] hız değiştirme formülünde bir parçacığın önceki hızının yeni hız üzerindeki etkisini kontrol eden bir etken olarak atalet ağırlığını önermişlerdir.

Hız belirleme denkleminde w hız değeri denkleminde çarpan ağırlık değeri olarak bulunduğundan, w> 1 olduğunda, parçacıkların hızları zaman içinde azami hıza yükselir ve ıraksama oluşarak parçacıkların daha iyi bölgelere doğru yönlendirilmesi zorlaşır. Eğer w <1 olursa, o zaman parçacıklar hızları 0 olana kadar yavaşlar ve sürünün araştırma yeteneği köreltilmiş olur [46]. Uygun şekilde ayarlandığında, atalet ağırlığı, yerel ve global aramayı dengeleyerek daha az sayıda iterasyon kullanılarak, optimal değerlerin elde edilmesine yardımcı olur [44].

Hızlanma katsayıları c1 ve c2, rasgele r1 ve r2 vektörleri ile birlikte parçacıkların bilişsel ve sosyal bileşenlerinin stokastik etkilerini kontrol eder. c1 ve c2 güven katsayıları olarak da adlandırılırlar. c1 parçacığın kişisel tecrübesine olan güveni, c2 ise sürünün kolektif tecrübesine olan güveni ifade eder [47].

Her bir iterasyonda, hızdaki değişimlerden dolayı parçacığın yörüngesi, kontrol edilemeyen bir biçimde, problem alanında sınırsız artış gösterebilir. Bu nedenle, parçacığın araştırmayı faydalı bir şekilde yapması için bu değişimlerin kontrol edilmesi gerekmektedir. Bu amaçla kullanılan geleneksel yöntem, $V_{max}$ sabitini aşağıda gösterildiği gibi sistem sabiti olarak kullanmaktır. Bununla birlikte, $V_{max}$ değerinin belirlenmesi problemden probleme değişebilir. Bazı uygulamalarda, $V_{max} = X_{max}$ olarak kullanılır [45].

### 2.4.3. Diferansiyel Gelişim (DE) Algoritması

Price ve Storn'un [12] 1995 yılında geliştirmiş oldukları Diferansiyel Gelişim Algoritması, özellikle sürekli optimizasyon problemlerinde etkin sonuçlar verebilen, işleyiş ve operatörleri bakımından evrimsel algoritma tabanlı sezgisel bir optimizasyon yöntemidir. Genetik Algoritmadaki çaprazlama, mutasyon ve seçim operatörleri Diferansiyel Gelişim Algoritmasında da kullanılmıştır [48,49]. Eski bireyle üretilen yeni bireyin fitnesi (uygunlukları) kıyaslanarak uygunluğu yüksek olan, yeni birey olarak bir sonraki jenerasyona aktarılmaktadır. Bu şekilde seçim operatörü de uygulanmış

olmaktadır. Diferansiyel Gelişim Algoritması genellikle sürekli değişkenlerle çalışmakla birlikte kesikli değişkenler çalışanları versiyonları da geliştirilmiştir [50,51]. Diferansiyel Gelişim Algoritmasının kod taslağı özet bir şekilde aşağıda verilmektedir.

1. Başlangıç parametreleri (Maksimum iterasyon sayısı, Popülasyon büyüklüğü, Çaprazlama oranı (CR), Ölçekleme faktörü(F))
2. Popülasyonun başlatılması: *n* bireyli bir popülasyon oluşturacak şekilde Eşitlik (2.4)'e göre rasgele çözümlerin oluşturulması.
3. Değerlendirilme: Popülasyondaki her bir kromozomun uygunluk değerinin hesaplanması
4. **Tekrar et** (Her bir birey için)
    a. Eşitlik (2.7) ile Mutasyon:

    $$x_i^{\text{mutasyon}} = x_{r1} + F(x_{r3} - x_{r2}) \quad (2.7)$$

    Burada ($r_1 \neq r_2 \neq r_3 \neq i$) rastgele seçilen komşu çözümlerdir.

    b. Eşitlik (2.8) ile Çaprazlama:

    $$x_i^{\text{yeni}} = \begin{cases} x_i^{\text{mutasyon}} & if\ j_{rand} \leq CR \\ x_i & diğer\ durumlarda \end{cases} \quad (2.8)$$

    c. Seçim operatörü: Bireylerin yeni jenerasyonda yer alma olasılıkları uygunluklarına bağlıdır. Karşılaştırılan bireylerden uygunluğu yüksek olan birey yeni jenerasyonun bireyi olarak atanmaktadır.
5. **Koşullar sağlanana kadar**

Her bir çözüm üretme işlemi neticesinde parametre değerleri araştırma uzayının alt ve üst değerlerinin ötesine geçerse sınır dışındaki değerler izin verilen aralığa çekilebilir veya değişkenin alt ya da üst sınır değeri alması sağlanabilir.

### 2.4.4. Yapay Arı Kolonisi (ABC) Algoritması

ABC algoritması Karaboğa tarafından, arıların doğada sergiledikleri besin arama hareketlerinden esinlenerek geliştirilmiş bir algoritmadır. ABC, arıların iş birliği içinde hareket ettikleri, popülasyona tabanlı, sezgisel bir optimizasyon tekniğidir. ABC algoritması, çok boyutlu optimizasyon problemlerinin çözümü için geliştirilmiş bir

optimizasyon tekniğidir [13,32]. Arı kolonisinin bulunduğu kovanın çevresi, optimizasyon probleminin arama uzayı olarak tanımlanır. Her çözüm arılar için bir yiyecek kaynağı lokasyonu, yiyecek kaynağının nektar miktarı ise çözüm değerinin uygunluk değerine karşılık gelmektedir. Bu nedenle ABC algoritması, en çok nektar içeren kaynağın konumunu bulmaya çalışarak, çözüm uzayı içinde mekândaki çözümlerden asgari veya azami değerleri veren yeri bulmaya çalışır [13,32,36].

ABC algoritmasında gerçek arıların yiyecek arama davranışında olduğu gibi üç çeşit arı bulunmaktadır.

**İşçi Arılar:** Hafızalarında bulunan lokasyondaki kaynağın nektarının çıkarılması ve kaynak hakkında bilgiyi kovanda içerisinde bekleyen arılara iletmekle sorumludurlar. Bu iletim işini dans ederek sağlarlar. Dans bilgisi, kaynağın kovana olan mesafesi, nektarın kalitesi, kaynağının yönü gibi bilgiler içermektedir.

**Gözcü Arı:** Besin kaynakları hakkında bilgi toplayan işçi arılar, toplanan bu bilgileri kovanda bekleyen gözcü arıcılara iletirler. Gözcü arılar işçi arıların dansını izleyerek tarif edilen kaynaklardan seçim yapan arılardır. Gözcü arılar olasılıksal olarak daha iyi kaynakları seçme eğilimindedir.

**Kâşif Arı:** Rastgele dolaşarak yeni besin kaynağı arayan arılardır. Ayrıca kaynağındaki nektar tükenen arılar da kaşif arı gibi davranmaya başlarlar.

Yapay Arı Kolonisi Algoritmasının kod taslağı şu şekildedir:

1. Başlangıç parametrelerinin ayarı: limit, yiyecek kaynağı sayısı, maksimum çevrim sayısı değerlerinin belirlenmesi
2. Eşitlik (2.4) ile başlangıç yiyecek kaynağı lokasyonlarının üretilmesi
3. Besin kaynaklarının kalitesinin Eşitlik (2.9) ile hesaplanması

$$fitness_i = \begin{cases} \frac{1}{1+f_i}, f_i \geq 0 \\ 1 + abs(f_i), f_i < 0 \end{cases} \quad (2.9)$$

4. **While** (Durdurma kriteri ile karşılaşılanına kadar) **do**
5. **Fo**r her işçi arı için
    - İşçi arının hafızasındaki kaynağın civarında Eşitlik (2.10) ile araştırma yaparak y bir besin kaynağının üretilmesi

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (2.10)$$

Burada $v_i$ yeni kaynağın konumu, $x_i$ eski kaynağın konumu, $x_k$rastgele seçilen bir komşu çözüm, $\emptyset_{ij}$ [-1,1] aralığında üretilen rasgele bir sayıdır.

- Üretilen besin kaynaklarının Eşitlik (2.9) ile kalitesinin hesaplanması
- Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması

**6. End**

- Eşitik (2.11) ile besin kaynaklarının seçilme ihtimalinin hesaplanması

$$p_i = \frac{fitness_i}{\sum_{i=1}^{foodsource} fitness_i} \quad (2.11)$$

7. **For** her gözcü arı için
   - Olasılık değerine bağlı olarak seçilen çözümün civarında Eşitlik (2.10) ile yeni çözümlerin üretilmesi
   - Üretilen besin kaynağının Eşitlik (2.9) ile kalitesinin hesaplanması
   - Kaynağın kalitesine bağlı olarak aç gözlü seçimin uygulanması

**8. End**

9. Popülasyon içinde o ana kadarki en iyi çözümün (güncellenmesi).
10. Tükenen yiyecek kaynağı varsa (sayac$_i$>*limit*):
    - Eşitlik (2.4) ile yeni rastgele bir besin kaynağının üretilmesi
    - Eşitlik (2.9) ile kaynağın kalitesinin hesaplanması
11. **End While**

Algoritmada her bir yiyecek kaynağı optimizasyon problemi için potansiyel bir çözüme karşılık gelir ve Eşitlik (2.4) ile kaynakların ilk değerleri üretilerek başlanır. Üretme işlemi, her parametrenin alt ve üst sınırları dahilinde kalacak şekilde rastgele değer üreterek gerçekleşir [13,32]. Eşitlik (2.10)'daki $x_{ij}$ ve $x_{kj}$ arasındaki mesafe azaldıkça, yani çözümler benzer olduğunda, parametredeki değişim miktarı azalacaktır. ABC algoritmasının her adımında kaynak ve komşu kaynak pozisyonu değerlerinin belirli sınır aralıkları içinde kalması gerekir ve bu sınırı aşmaması gerekir. Bu nedenle, bitişik kaynak için üretilen sınır değeri olan $v_{ij}$, önceden tanımlanan parametre sınırlarını

aşarsa, parametrenin alt veya üst sınır değerleri ile sınırlandırılmalıdır. Eşitlik (2.12), yeni kaynak pozisyon değerlerini optimizasyon problemi sınırları içinde tutmak için kullanılır [13,32,36].

$$v_{ij} = \begin{cases} x_j^{min}, v_{ij} < x_j^{min} \\ v_{ij}, x_j^{max} \leq v_{ij} \leq x_j^{max} \\ x_j^{max}, v_{ij} > x_j^{max} \end{cases} \quad (2.12)$$

$x_i$ ve $v_i$ arasında, miktarlarına (yani uygunluk değerlerine) göre aç gözlü bir seçim işlemi uygulanır. Yeni bulunan $v_i$ çözümü daha iyi ise işçi arı eski kaynağın yerini unutur (hafızasından siler) ve $v_i$ kaynağının yerini hafızaya alır. Öte yandan, işçi arı $x_i$ kaynağı ile ilgili geliştirememe sayacını günceller (sayac$_i$) ve bir artırır ve geliştirdiği durumlarda ise sayacını sıfırlar [13,32,36].

Gözcü arı biriminde her bir gözcü arı gideceği kaynağı seçmek için her bir kaynak için [0,1] aralığında rasgele üretilir. Bu sayı $p_i$'den küçükse, gözcü arı $i$'nci kaynağa gider. Kaynağın seçilme olasılığı, nektarın miktarı ile doğru orantılıdır. Nektarın miktarı ne kadar büyükse, gözcü arıcılar bu kaynağa daha yüksek olasılıkla gidecektir.

Tüm görevli ve gözcü arılar, kaynak arama işleminin bir iterasyonunu tamamladıktan sonra, çözüm sayacı (sayac$_i$) kontrol edilir. Bir kaynağın sayacı, *"limit"* olarak tanımlı bir kontrol parametresi olan belirli bir sınırın üzerine çıktığında, bu kaynağın (çözüm) tükendiğine işarettir ve bu kaynaktaki işçi arının kaynağını terk etmesine ve yeni besin kaynağı aramaya başlamasına karar verilir. İşçi arı yeni kaynaklar aramaya başladığında, kaşif bir arıya dönüşür ve Eşitlik (2.4) ile rastgele çözümler (besin) aramaya başlar.

### 2.4.5. Binom Çaprazlama Operatörlü Yapay Arı Kolonisi Algoritması (BCABC)

Sosyal öğrenme, sürü zekasının kolektif bilgi oluşumunun en önemli bileşenidir. ABC algoritmasında, bu olgu işçi arılar ve gözcü arıların seçim işlemleri ve komşu bölge arama işlemleri vasıtası ile gerçekleştirilmiştir. Ancak ABC algoritmasında kullanılan yerel arama metodu bazı problemler üzerinde yetersiz kalabilmektedir. Bunun sebepleri

1. Birincisi, boyut arttıkça bilgi değişikliği halen bir boyutta sınırlı kalmaktadır. Bu da algoritmanın yakınsama hızını yavaşlatabilmektedir.

2. İkincisi ise, komşu arı ve bilgisi değişecek boyut indeksi tamamen rasgele seçilmektedir. Bu durumda ise, popülasyonu daha iyi bir alana yönlendirebilecek olan ve daha yüksek uygunluk değere sahip besin kaynakları kullanılmamış olur.

Bu nedenlerle ABC algoritması ile çeşitli hibrit yaklaşımlar önerilerek bazı problem türleri üzerinde yakınsama hızının artırılması hedeflenmiştir. Örneğin, Yan, Xiaohui ve diğerleri [52] veri kümeleme için bir Hibrit Yapay Arı Koloni Algoritması (BCABC) algoritması sunmuşlardır. Li ve Li Bin [53] ise DE algoritmasının dezavantajının üstesinden gelmek ve aramanın genel (küresel) arama kabiliyetini kuvvetlendirmek için, DE'nin ve ABC'nin avantajlı yönlerini bir araya getirerek DE metodu destekli hibrit bir ABC algoritması geliştirmişlerdir. Kefayat ve diğerleri [54], dağılmış enerji kaynaklarının optimal ölçümleri ve lokasyonlarını bulmak amacıyla bir Karınca Koloni Algoritması (ACO) ile bir Yapay Arı Koloni Algoritmasının (ABC) karışımı olan ve ACO-ABC adını verdikleri bir hibrit konfigürasyonu sunmuşlardır. Zhang ve diğerleri [55] ise, toplam ağırlıklı gecikmeyi minimize ederek aynı problemi çözmek amacıyla karmaşıklığın büyüklüğünden dolayı yeni bir Yapay Arı Koloni (ABC) algoritması öne sürmüşlerdir. Feng Liu ve Yang Liu [56], permütasyonlu akış istasyonları planlama problemlerinin üretim sürelerini minimize etmek için ise bir Hibrit Ayrık Yapay Arı Koloni Algoritması sunmuşlardır. Yildiz [57], Yapay Arı Koloni Algoritması (ABC) ile Taguchi metodunun hibriti olan yeni bir optimizasyon yöntemi geliştirmiştir ve bunu da bir otomobil parçasının yapısal tasarımının optimizasyonu ve çok aletli öğütme optimizasyonu için kullanmıştır. Tuba [58], genetik algoritmasının (GA) çaprazlama ve mutasyon işlemlerine dayalı ABC algoritmasının bir modifikasyonunu tarif etmiştir. Pandey ve Kumar [59], Gezgin Satıcı Problemine uygulamak amacıyla ABC algoritmasını farklı türden kodlanmış çaprazlama işlemleri ile birlikte sunmuştur. Kumar ve diğerleri [60], Genetik Algoritmasının (GA) çaprazlama işlemini ABC algoritmasına entegre ederek, Çaprazlama Tabanlı ABC (CbABC) adını verdikleri GA ile birleştirilmiş yeni bir hibrit ABC algoritması tarif etmiştir. Brajevic [61], Çaprazlama Tabanlı ABC (CbABC) algoritmasını geliştirerek kısıtlı optimizasyon problemlerinde kullanılmak üzere yeni bir Yapay Arı Koloni Algoritması (ABC) tarifi vermiştir.

Bu tez çalışmasında, Binom çaprazlama operatörü [62] orijinal ABC algoritmasına dahil edilerek ayrık bir problem olan kriptoanalizde yakınsama performansını iyileştirecek

yeni bir ABC algoritması geliştirilmesi amaçlanmıştır. Elde edilen yeni metot Binom çaprazlama operatörü Yapay Arı Koloni Algoritması (BCABC) olarak adlandırılmıştır. İşçi arılar ve gözcü arılar arasına eklenen bu metot dışında kalan kısımlarda orijinal ABC algoritmasına sadık kalınmıştır. Binom çaprazlama işlemi ile, BCABC algoritmasında bilgi değişikliği kuvvetlendirilmiş olur. Ayrıca. BCABC algoritmasının kod taslağı aşağıda verilmiştir:

1. Başlangıç parametrelerinin ayarlanması (Koloni büyüklüğü, limit, maksimum çevrim sayısı, Çaprazlama oranı (CR))
2. Çözüm popülasyonunun Eşitlik (2.4) ile başlatılması ve popülasyonun değerlendirilmesi.
3. **Repeat** (Besin kaynağı bulundukça)
4. **For işçi arı aşaması**
    a. İşçi arılar için Eşitlik (2.10) ile yeni çözümlerin üretilmesi ve değerlendirilmesi.
    b. Açgözlü seçim işleminin yürütülmesi.
5. **End**
6. **Çaprazlama aşaması**
7. **For** (Her bir besin kaynağı için, ebeveyn (*i*) seçilerek)
    a. Besin kaynağı içinden *i*+1 ebeveynin rasgele seçilmesi
    b. Seçilen ebeveynlerin çaprazlanması ile yeni besin kaynağının *new(i)* üretilmesi
    c. Yeni besin kaynağı *new(i)*'nin değerlendirilmesi
    d. Açgözlü seçim işleminin yürütülmesi.
8. **End**
9. Besin kaynaklarının kalitelerine bağlı seçilme ihtimalinin Eşitlik (2.11) ile hesaplanması
10. **For gözcü arı aşaması**
    a. Olasılık değerine bağlı olarak yeni çözümlerin üretilmesi ve değerlendirilmesi.
    b. Açgözlü seçim işleminin yürütülmesi.
11. **End**
12. Popülasyon içinde o ana kadarki en iyi çözümün (güncellenmesi).

13. **Kaşif arı aşaması**
14. Kolonide tükenen kaynak varsa (sayac$_i$>*limit*):
    a. Rastgele bir besin kaynağının Eşitlik (2.4) ile üretilmesi
    b. Kaynağın kalitesinin hesaplanması
15. **Until** (koşullar sağlanana kadar)

Çaprazlama aşamasında kolonideki besin kaynaklarının her biri için, iki ebeveyn besin kaynağı seçilir. Bu ebeveynlerden ilki *i* ve ikincisi *i*'den farklı olarak rasgele seçilen *j* olmak üzere; bu iki ebeveyn çaprazlanır ve oluşan yeni nesil $i_{new}$ ile *i* karşılaştırılır. Burada açgözlü bir seçim uygulaması yapılır ve daha iyi olan popülasyonda tutulur. Ebeveyn sayısı eşitlenene kadar bu seçim devam ettirilir. Binom çaprazlamanın kod taslağı aşağıda verilmektedir [62].

**Binom çaprazlama**

1. Çaprazlama oranının girilmesi (CR);
2. Aynı boyutta iki ebeveynin seçilmesi (x,y);
3. (0,1) aralığında *k* sayısının rasgele üretilmesi;
4. *k*'nın uzunluğu = x'in uzunluğu;
5. **for** i=1: *k*'nın uzunluğu **do**
    a. eğer ki k ≤ CR ise
        i. Yeni birey *y*(*i*) olur;
    b. yoksa
        i. Yeni birey *x*(*i*) olur;
    **c. end**
6. **end**

Yeni nesil üretildikten sonra, orijinal besin kaynağı (*i*) ve yeni üretilen nesil ($i_{new}$)'ye bir açgözlü seçim uygulanır. Yeni neslin uygunluk değeri orijinal bireyden daha yüksek olursa, onun yerini alır ve bu besin kaynağı için işleyen sayaç sıfırlanır. Aksi halde, hafıza değişmeyecektir ve tıpkı işçi arı ve gözcü arılar aşamasında olduğu gibi sayaç değeri bir artırılır.

# 3. BÖLÜM

# OPTIMİZASYON ALGORİTMALARI İLE KRIPTOANALİZ

Çeşitli araştırmacılar doğadan esinlenilen algoritmalardan yararlanarak klasik kripto sistemlerin kriptoanalizlerinde belli ölçülerde başarı elde etmişlerdir. Spillman ve diğerleri [14] bir monoalfabetik yer değiştirmeli şifreleme sistemini çözmek için genetik algoritmadan faydalanmışlardır. Yine Clark [63] ise GA, TS ve SA algoritmalarını tanıtarak, bu metotları yer değiştirmeli şifreleme sistemlerinin kriptoanalizinde kullanmışlardır. Ayrıca Clark ve diğerleri [64] polialfabetik yer değiştirmeli şifrelemede GA kullanmayı önerirken bir ilki gerçekleştirmişlerdir. Daha sonra, Clark ve Dawson [65], [66]'te sunulan fikri daha da geliştirerek Vigenére şifrelemeye karşı saldırı gerçekleştirmek amacıyla bir paralel genetik algoritmayı temel alan yeni bir yöntem ortaya koymuşlardır. Dimovski ve Gligoroski [67] ise üç farklı sezgisel optimizasyon yöntemi (SA, GA ve TS) sunarak yerine koymalı şifreleme çözümünde kullanmışlardır. Verma ve diğerleri [68], monoalfabetik yer değiştirmeli bir şifreleme yönteminden bahsettikleri çalışmalarında ise, GA ve TS'ye dayalı bir kriptoanaliz uygulaması yaptıktan sonra bu yöntemlerin verimlerinin karşılaştırmasını yapmışlardır.

Song ve diğerleri [69] de yer değiştirmeli şifreleme yöntemin kriptoanalizini yürütmek amacıyla otomatik bir yöntem geliştirmişlerdir. Garg [70], GA, TS ve SA'yı temel alan bir metottan bahsettikleri makalelerinde ise yine yer değiştirmeli şifreleme yönteminin kriptoanalizi üzerine çalışmalar yürütmüşlerdir. Omran ve diğerleri [71] ise Vigenére şifrelenmiş metinlere saldırmak amacıyla bir GA uygulaması gerçekleştirmişlerdir. Bhateja ve Kumar [72] yeni bir uygunluk fonksiyonu ve seçkin GA algoritmalar kullanarak Vigenére şifrelemenin kriptoanalizini incelemek amacıyla bir yöntem geliştirmişlerdir. Öte yandan, Boryczka ve Dworak [73] de yer değiştirmeli şifreleme yöntemin kriptoanalizinin işlemesinin GA gibi evrimsel algoritmaların yardımıyla nasıl hızlandırılabileceğini göstermişlerdir. Yine Boryczka ve Dworak [74] bu kez GA gibi

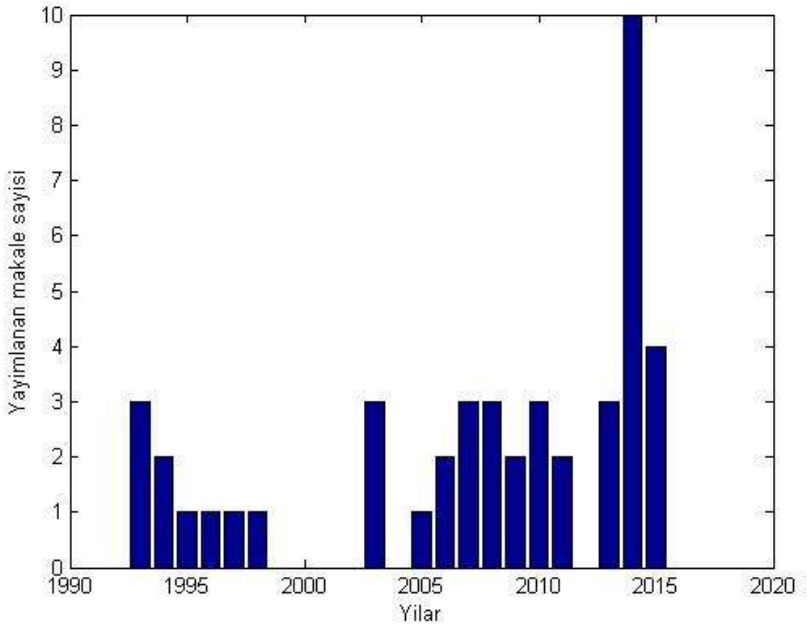
evrimsel algoritmaların karmaşık kriptoanaliz işleminin nasıl optimize edilebileceğini sunmuşlardır. Uddin ve Youssef [75], ACO kullanarak basit yer değiştirmeli şifrelemeye saldırmak amacıyla ACO yöntemini kullanmışlardır. Bhateja ve diğerleri [76] Vigenére şifrelemenin kriptoanalizinde bir CS algoritmasının varlığını araştırmışlardır. Luthra ve Pal ayrıca Li ve diğerleri [77, 38], monoalfabetik şifrelemenin kriptoanalizi için FA metodunun içerisine mutasyon ve çaprazlama işlemlerin gömerek farklı bir çalışma gerçekleştirmişlerdir. Vigenére şifrelemenin kriptoanalizi ayrık bir optimizasyon problemi olduğundan dolayı [76] ve GA ile PSO gibi yöntemler çok güçlü arama algoritmaları klasik şifrelemelerin çözümlemelerinde kullanılabilse de, bu tür teknikler anahtar 15'ten fazla karakter içermesi durumunda yetersiz kalmaktadır. Bu nedenle pratikte, gizli haberleşme için anahtar uzunluğu büyük seçilir (15 karakterden fazla).

Sezgisel algoritmaların çeşitli klasik şifreleme yöntemlerinin (monoalfabetik, polialfabetik, yerine koyma, Hill ve Vernem) kriptoanalizine uygulaması ile ilgili 44 adet makale incelenmiştir. Yüksek Öğretim Kurumu Ulusal Tez Veri tabanı, Springer, Elsevier, Scopus, IEEE Transaction, SCI ve SCI Expanded, EBSCO, Science Direct veri tabanlarında yapılan kaynak taramasında ulaşılabilen dokümanlar, veri tabanları ve ağ üzerinde ulaşılabilen dokümanlar ile kitap, dergi, tez, makale, bildiri, rapor gibi basılı materyaller ve dijital veriler incelenmiştir. Aynı şekilde Erciyes Üniversitesi Kütüphanesi'nin çevrimiçi katalog taramasına da başvurulmuştur. Internet üzerindeki farklı kaynaklardaki makale ve dokümanlar da incelenmiş ve bunlar destekleyici bilgi olarak kullanılmıştır. Bu makalelerin 30 tanesinde GA, 3 tanesinde SA, 3 tanesinde AC, 3 tanesinde CS, 2 tanesinde PSO, 1 tanesinde TS, 1 tanesinde DE ve 1 tanesinde de FA algoritmalarına başvurulmuştur. Tablo 3.1'de algoritmaların ve uygulamalarının kategorik görünümünü özetlemektedir. Bu tablodan, klasik şiflemenin birçok türüne yönelik saldırı gerçekleştirmek amacıyla daha çok stokastik optimizasyonlara dayalı kriptoanalizler yapılmıştır. Şekil 3.1 ise, yıllara göre stokastik optimizasyon algoritmalarını temel alan klasik şifreleme yöntemlerinin kriptoanalizleri üzerine yapılan çalışmaların sayılarını özetlemektedir. Burada, görülüyor ki, bu tür çalışmaların sayısı yıllara göre düşüş gösterse de literatürde birçok boşluğun olduğu söylenebilir. Bunlardan biri, klasik şifrelemeyi çözümlemek amacıyla Yapay Arı Koloni gibi modern optimizasyon algoritmalarının kullanılması olarak sayılabilir.

Tablo 3.1. Kriptoanaliz için optimizasyon yöntemleri tabanlı çalışmalar

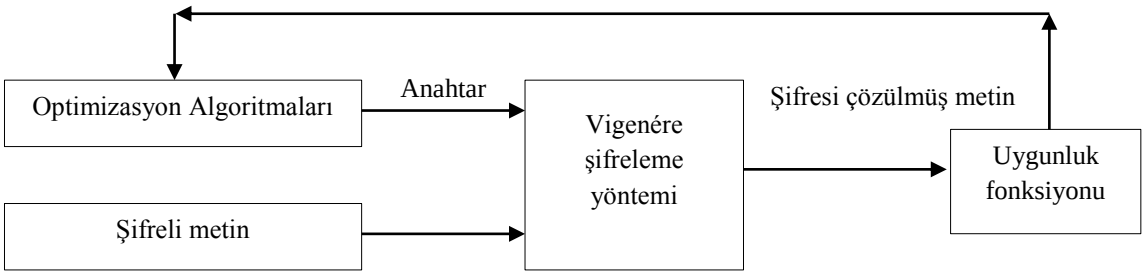
| **Algoritma** | **Yazar** | **Yayın başlığı** |
|---|---|---|
| Genetic Algorithm | Spillman et al. (1993) | Cryptanalysis of Simple Substitution Cipher [14] |
| | Matthews (1993) | GA in Cryptanalysis [62] |
| | Clark (1994) | Modern Optimisation Algorithms and Cryptanalysis [64] |
| | Lin and Kao (1995) | Cryptanalysis of vVernem Ciphers [78] |
| | Clark et al. (1996) | Cryptanalysis of Poly-alphabetic Ciphers [63] |
| | Clark and Dawson (1997) | A parallel GA for Cryptanalysis of the Poly-alphabetic Cipher [65] |
| | Clark and Dawson (1998) | Automated Cryptanalysis of Classical Ciphers [66] |
| | Grundlingh and Vuuren (2003) | Break a Simple Cryptographic Cipher [39] |
| | Dimovski and Gligoroski (2003) | Attacks on the Transposition Cipher [67] |
| | Li et al. (2005) | Heuristic Cryptanalysis of Classical and Modern Cipher [38] |
| | Toemeh and Arumugam (2007) | Breaking Transposition Cipher [79] |
| | Verma et al. (2007) | Attack on the Monoalphabetic Cipher [68] |
| | Toemeh and Arumugam (2008) | Searching Key Space of Poly-alphabetic Ciphers [80] |
| | Garg (2009) | A comparison and Cryptanalysis of Transposition Cipher [70] |
| | Muhajjar (2010) | Cryptanalysis Of Transposition Ciphers [81] |
| | Hausman and Erickson (2009) | A Dominant gene GA and Substitution Cipher [82] |
| | Omran et al. (2010) | Break a Monoalphabetic Substitution Cipher [71] |
| | Dureha and Kaur (2013) | Automate an Attack on Classical Ciphers [83] |
| | Heydari and Mohammad (2013) | Cryptanalysis of Transpositionciphers With Long Key Lengths And an Improved GA [84] |
| | Al-Khalid et al. (2013) | Break a Simple Transposition Cipher [85] |
| | Alkathiry and Al-Mogren (2014) | A Powerful GA to Crack a Transposition Cipher [86] |

| | | |
|---|---|---|
| | Saveetha et al. (2014) | Cryptography and Optimization Heuristics Techniques [87] |
| | Bhateja et al. (2014) | Analysis of Different Cryptosystems and Meta heuristic Techniques [88] |
| | Jadaun et al. (20014) | Deciphering of Transposition Ciphers [89] |
| | Sadeghzadeha and Taherbaghalb (2014) | A new Method and Comparison With TS and SA [90] |
| | Boryczka and Dworak (2014) | Genetic Transformation Techniques in Cryptanalysis [73] |
| | Boryczka and Dworak (2014) | Cryptanalysis of Transposition Cipher Using EA [74] |
| | Bhateja and kumar (201) | Genetic Algorithm With Elitism for Cryptanalysis of Vigenére Cipher [72] |
| | Al-Khalid and Al-Khfagi (2015) | Cryptanalysis of a Hill Cipher [91] |
| | Bergmann et al. (2015) | Cryptanalysis Using GA [92] |
| Simulated Annealing | Forsyth and Safavi-Naini (1993) | Automated Cryptanalysis of Substitution Cipher [93] |
| | Giddy and Safavi-Naini (1994) | Automated Cryptanalysis of Transposition Ciphers [94] |
| | Yang et al. (2008) | Cryptanalysis of Transposition Cipher [69] |
| Tabu Search | Verma et al. (2007) | Attack on the Monoalphabetic Cipher [68] |
| Particle Swarm | Uddin and Youssef (2006) | Cryptanalysis of Simple Substitution Cipher [95] |
| | Hameed and Hmood (2010) | Cryptanalysis of Transposition Cipher [96] |
| Differential Evolution | Wulandari et al. (2015) | Cryptanalysis of Transposition Cipher [97] |
| Ant Colony | Russell et al. (2003) | Breaking Transposition Cipher [98] |
| | Uddin and Youssef (2006) | Cryptanalysis of Simple Substitution Ciphers [75] |
| | Mekhaznia and Menai (2014) | Cryptanalysis of Classical Cipher [28] |
| Cuckoo Search | Sadiq and Kareem (2014) | Attacking Transposition Cipher Using Improved CS [29] |
| | Jain and Chaudhari (2015) | A New Heuristic Based on the CS for Cryptanalysis of Substitution Ciphers [30] |
| Firefly algorithms | Bhateja et al. (2015) | Cryptanalysis of Vigenére Cipher [76] |
| | Luthra and Pal (2011) | A hybrid FA and Cryptanalysis of a Mono-alphabetic Cipher [77] |




*Şekil 3.1.* Yıllara göre yayımlanan makale sayısı

Sezgisel algoritmalar, optimizasyon problemleri için en uygun çözümleri bulurken yönlendirilmiş arama ve rassal dağılımları kullanır. Luke [8], bir sorun için etkili olarak bilinen algoritmalar yoluyla soruna çözüm bulmanın çok zor hatta bazen imkansız olduğunu göstermiştir. Özellikle problemler sürekli olmadıklarında ve gürültülü olduklarında, sezgisel yaklaşımlar problemin yüzeyinde birçok varsayım yapan analitik yaklaşımlara tercih edilmektedir. Örneğin klasik şifrelemede, olası anahtarların grubu alfabedeki harflerin bütün permütasyonlarının grubu olduğundan, bu grubun ihtimaller sayısı oldukça yüksektir ve kaba kuvvet ile yapılan kriptoanaliz hesaplama karmaşıklığından dolayı verimli değildir. Sonuçta, optimal anahtarı elde etmek için önceki çözüm bilgileri ile rastgeleliğe dayalı arama algoritmaları kullanılır.




*Şekil 3.2.* Optimizasyon Algoritmaları ile Kriptoanaliz Blok Diyagramı

Sezgisel algoritmalarla kriptoanaliz yapılırken kullanılan sisteme ait blok diyagram Şekil 3.2'de gösterilmektedir.

Sezgisel algoritmalarla kriptoanalizde bir başlangıç çözüm popülasyonu oluşturulur. Her bir çözüm, şifreli metnin kriptoanalizi için önerilen bir anahtardır. Klasik kriptografik algoritmalarda şifreleme ve şifre çözme işlemi aynı anahtarla yapıldığından, çözüm (varsayılan anahtar) $N$ boyutlu bir vektördür. Başlangıç popülasyonu kullanılan dilin alfabesine bağlı olarak kabul edilebilir bir aralıkta üretilir. Bir vektördeki elemanların sayısı ($N$), anahtar elemanlarının sayısına eşit olmalıdır. Bundan sonra her bir çözüm (önerilen anahtar) maliyet fonksiyonunda değerlendirilir ve her bir anahtarın frekansına bağlı olarak önerilen anahtarın verimliliğini ölçer. Çeşitli seçim işlemleri ile daha iyi olanlar seçilir ve belirli sayıda iterasyon tamamlanana kadar üretim, değerlendirme ve seçim aşamaları devam eder.

Klasik kripto (şifrelenmiş) sistemlere saldırmak için sezgisel algoritmaların kullanımında, uygunluk (fitness) fonksiyonu adını verdiğimiz, bir anahtarın geçerliliğini değerlendiren bir metoda ihtiyaç vardır. Uygunluk fonksiyonları, frekans analizine bağlı olarak geçerliliği değerlendirebilirler. Frekans analizinin amacı ise şifresi çözülmüş metin ile İngilizce ve Türkçe literatürlerindeki frekansları karşılaştırmaktır. Başka bir deyişle, frekans analizi belli bir metin ile o metinde kullanılan dil arasında keskin bir ilişki kurar.

Vigenére şifrelemesine saldırmak amacıyla sezgisel bir algoritma kullanmak için, varsayılan anahtarın verimliliğini keşfetmek için bir fonksiyona ihtiyaç vardır. Daha önce Vigenére şifreleme ile ilgili yapılan çalışmalara göre, İngilizcede sadece tek ve ikili harflerle ilgili istatistiklerin bir önemi vardır [72,76]. Varsayılan anahtar (K) şifreli metnin şifresini çözmek için kullanılır. En sık kullanılan harflerin sıklıkları ve en sık

kullanılan ikililerin frekansları (İngilizce veya Türkçe) hesaplanır. İngilizce veya Türkçede, şifreleri çözülmüş metinlerin içerisindeki harflerin sayılarının frekanslarının mutlak farklarının toplamı hesaplanır. Ayrıca, güncel dilde buna karşılık gelen harflerin beklenen frekansları da hesaplanır. Benzer şekilde, şifreleri çözülmüş metinlerin içerisindeki ikililerin sayılarının frekanslarının mutlak farklarının toplamı ile güncel dilde buna karşılık gelen ikililerin beklenen frekansları hesaplanır. Belli bir K anahtarının uygunluk fonksiyonu Eşitlik (3.1)'e göre verilmektedir [72,76]:

$$f(K) = \lambda_1 . \sum_{i=1}^{unigram} |OFM(i) - EFM(i)| + \lambda_2 \sum_{i=1}^{bigram} |OFB(i) - EFB(i)| \quad (3.1)$$

Burada, OFM*(i)* ve EFM*(i)* sırasıyla, *i*. harfin gözlenen ve beklenen frekansları; OFB*(i)* ve EFB*(i)* ise *i*. ikilinin gözlenen ve beklenen frekanslarıdır $\lambda_1$ ve $\lambda_2$, sırasıyla harfler ve ikililere atanmış olan ağırlıklardır. $\lambda_1$ ve $\lambda_2$'ye en uygun değerlerin, farklı uzunluktaki şifreli metinler ve anahtarlar kullanılarak elde edilen doğru çözülmüş kelime sayısının yüzde değerine bağlı olarak elde edilebileceğini öne sürmüşler ve $\lambda_1$= 0.23 ve $\lambda_2$= 0.77 olması durumunda, yakınsama hızının azami olduğu gösterilmiştir[99].

Harflerin ve ikililerin optimal değerlerini bulmak içinse, Tablo 3.2 ve Tablo 3.3'de görülen istatistikler kullanılmaktadır [100]. Bu veriler İngilizce metinlerde bulunan yaklaşık 4.5 milyar karakterden üretilmiştir. Buna göre, 26 harften oluşan İngiliz alfabesi Tablo 3.4'te gösterildiği gibi beş gruba ayrılabilir.

Tablo 3.2. İngilizce 'de kullanılan harflerin kullanım sıklıkları (%)

| No | Harf | Frekans | No | Harf | Frekans | No | Harf | Frekans |
|---|---|---|---|---|---|---|---|---|
| 1 | A | 8.55 | 11 | K | 0.81 | 21 | U | 2.68 |
| 2 | B | 1.60 | 12 | L | 4.21 | 22 | V | 1.06 |
| 3 | C | 3.16 | 13 | M | 2.53 | 23 | W | 1.83 |
| 4 | D | 3.87 | 14 | N | 7.17 | 24 | X | 0.19 |
| 5 | E | 12.10 | 15 | O | 7.47 | 25 | Y | 1.72 |
| 6 | F | 2.18 | 16 | P | 2.07 | 26 | Z | 0.11 |
| 7 | G | 2.09 | 17 | Q | 0.10 | | | |
| 8 | H | 4.96 | 18 | R | 6.33 | | | |
| 9 | I | 7.33 | 19 | S | 6.73 | | | |
| 10 | J | 0.22 | 20 | T | 8.94 | | | |

Tablo 3.3. İngilizce ‘de kullanılan ikililerin kullanım sıklıkları (%)

| No | Bigram | Frekans | No | Bigram | Frekans | No | Bigram | Frekans |
|---|---|---|---|---|---|---|---|---|
| 1 | TH | 2.71 | 11 | EN | 1.13 | 21 | NG | 0.89 |
| 2 | HE | 2.33 | 12 | AT | 1.12 | 22 | AL | 0.88 |
| 3 | IN | 2.03 | 13 | ED | 1.08 | 23 | IT | 0.88 |
| 4 | ER | 1.78 | 14 | ND | 1.07 | 24 | AS | 0.87 |
| 5 | AN | 1.61 | 15 | TO | 1.07 | 25 | IS | 0.86 |
| 6 | RE | 1.41 | 16 | OR | 1.06 | 26 | HA | 0.83 |
| 7 | ES | 1.32 | 17 | EA | 1.00 | 27 | ET | 0.76 |
| 8 | ON | 1.32 | 18 | TI | 0.99 | 28 | SE | 0.73 |
| 9 | ST | 1.25 | 19 | AR | 0.98 | 29 | OU | 0.72 |
| 10 | NT | 1.17 | 20 | TE | 0.98 | 30 | OF | 0.71 |

Tablo 3.4. İngiliz alfabesi grubu

| No | Harfler | Olasılık aralığı |
|---|---|---|
| 1 | E | 12.10 |
| 2 | T, A, O, I, N, S, H, R | 8.55 – 6.33 |
| 3 | D, L | 3.87 – 4.21 |
| 4 | C, U, M, W, F, G, Y, P, B | 3.16 – 2.07 |
| 5 | V, K, J, X, Q, Z | 1.06-0 |

Türkçe ile ilgili istatistikler ise Tablo 3.5 ve Tablo 3.6’daki gibidir [101]. Daha önceden yapılmış olan çalışmaların sonucuna göre [101]. Buna göre, incelenen çalışmalarda harflerin kullanım sıklık değerlerinin bir metinden diğerine sayısal olarak küçük farklılıklar gösterebileceği, fakat sıralamada büyük bir farklılık olmayacağı söylenebilir. Tablo 3.5’te görüldüğü gibi Türkçe ‘de en çok kullanılan sesli harfler A, E, İ ve en çok kullanılan sessiz harfler N, R, L, K, D harfleridir. En az kullanılan harfler C, Ö, P, F, J harfleridir. Tabloda kullanım sıklıklarına göre en fazla olandan en aza doğru harflerin sıralanmış oldukları görülmektedir. Ancak yalnızca tek harf üzerinden frekans analizi yapmak tatmin eden sonuçlar vermeyebilir. Bu yüzden harf ikililerine de bakmak sonuçları geliştirmekte yardımcı olacaktır. Bu ikililer bigram olarak da anılmaktadır. Bu harf ikililerinin büyük çoğunluğunda sesli harfler sessiz harflerle birlikte görülür.

Tablo 3.5. Türkçe ‘de kullanılan harflerin kullanım sıklıkları (%)

| No | Harf | Frekans | No | Harf | Frekans | No | Harf | Frekans |
|---|---|---|---|---|---|---|---|---|
| 1 | A | 11.82 | 11 | Y | 3.42 | 21 | Ç | 1.19 |
| 2 | E | 9.00 | 12 | U | 3.29 | 22 | H | 1.11 |
| 3 | İ | 8.34 | 13 | T | 3.27 | 23 | Ğ | 1.07 |
| 4 | N | 7.29 | 14 | S | 3.03 | 24 | V | 1.00 |
| 5 | R | 6.98 | 15 | B | 2.76 | 25 | C | 0.97 |
| 6 | L | 6.07 | 16 | O | 2.47 | 26 | Ö | 0.86 |
| 7 | I | 5.12 | 17 | Ü | 1.97 | 27 | P | 0.84 |
| 8 | K | 4.7 | 18 | Ş | 1.83 | 28 | F | 0.43 |
| 9 | D | 4.63 | 19 | Z | 1.51 | 29 | J | 0.03 |
| 10 | M | 3.71 | 20 | G | 1.32 | | | |

Tablo 3.6. Türkçe ‘de kullanılan ikililerin kullanım sıklıkları (%)

| No | Harf | Frekans | No | Harf | Frekans | No | Harf | Frekans |
|---|---|---|---|---|---|---|---|---|
| 1 | AR | 0.02273 | 11 | İR | 0.01282 | 21 | İL | 0.00870 |
| 2 | LA | 0.02013 | 12 | Bİ | 0.01253 | 22 | Rİ | 0.00860 |
| 3 | AN | 0.01891 | 13 | KA | 0.01155 | 23 | ME | 0.00785 |
| 4 | ER | 0.01822 | 14 | YA | 0.01135 | 24 | Lİ | 0.00782 |
| 5 | İN | 0.01674 | 15 | MA | 0.01044 | 25 | OR | 0.00782 |
| 6 | LE | 0.01640 | 16 | Dİ | 0.01021 | 26 | NE | 0.00738 |
| 7 | DE | 0.01475 | 17 | ND | 0.00980 | 27 | RI | 0.00733 |
| 8 | EN | 0.01408 | 18 | RA | 0.00976 | 28 | BA | 0.00718 |
| 9 | IN | 0.01377 | 19 | AL | 0.00974 | 29 | Nİ | 0.00716 |
| 10 | DA | 0.01311 | 20 | AK | 0.00967 | 30 | EL | 0.00710 |

### 3.1. GA ile Vigenére Şifreleme Yönteminin Kriptoanalizi

Genel adımları daha önce verilen GA’nın Vigenére şifrelemenin kriptoanalizine uygulanması aşağıdaki adımlardan oluşmaktadır [102, 103]:

1. Şifreli metin, anahtar boyutu ve ilgili karakter frekansları belirlenir.
2. Başlangıç koşul parametreleri belirlenir. (Karar değişken sayısı, Değişkenlerin alt sınırı, Değişkenlerin üst sınırı, Azami iterasyon sayısı, Popülasyon boyutu (nPop), Çaprazlama oranı, Mutasyon oranı)
3. % Popülasyon dizisinin başlatılması
4. İngilizce için 0-25 ve Türkçe için 0-28 arasında rasgele sayılar seçerek N adet kromozom, başlangıç çözümünün Eşitlik (2.4) ile üretilmesi (varsayılan anahtar) (assumed key).

5. **for** i=1:nPop
    a. Şifre çözme anahtarı kullanarak mevcut çözümleri (varsayılan anahtar [i]) ile şifrenin çözülmesi.
    b. Deşifre edilen metnin uygunluk fonksiyon f değerinin bulunması (varsayılan anahtar [i])
    c. Azami uygunluk fonksiyon değere sahip kaynağın (BestSol) bulunması
6. **end**
7. % GA ana döngü
8. **for** it=1:MaxIt
9. % % Seçim olasılıkların hesaplanması
10. % Ebeveyn indislerin seçilmesi
11. Popülasyona iki kromozomun seçimi (varsayılan anahtar[i], varsayılan anahtar [j])
12. % Çaprazlama
13. **for** c=1:Npc
    a. İlk ebeveyn olarak c'nin seçilmesi
    b. İkinci ebeveynin i'den farklı olan j'nin rasgele seçilmesi
    c. Çaprazlama uygulayarak yeni varsayılan anahtar [i]'nin üretilmesi
    d. Deşifre anahtarı olarak yeni varsayılan anahtar [i]'yi kullanarak şifreli metnin çözülmesi
    e. Deşifre edilen metnin uygunluk değerinin hesaplanması: f(yeni varsayılan anahtar [i])
14. **end**
15. % Mutasyon
16. **for** m=1:Nm
    a. Ebeveyn (m)'nin seçilmesi
17. % Mutasyonun uygulanması
    a. Mutasyon uygulanarak yeni varsayılan anahtar [i]'nin üretilmesi
    b. Deşifre anahtarı olarak yeni varsayılan anahtar [i]'yi kullanarak şifreli metnin çözülmesi
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması: f(yeni varsayılan anahtar [i])
18. **end**

19. Bulunan en iyi sonucu güncelle
20. Bulunan en iyi maliyet değerini kaydet
21. end

### 3.2. PSO ile Vigenére Şifreleme Yönteminin Kriptoanalizi

PSO algoritmasının Vigenére şifreleme yönteminin kriptonanalizine uygulanması aşasıdaki adımlarla gerçekleşmektedir [102, 104]:

1. Şifreli metin, anahtar boyutu ve ilgili karakter frekansları belirlenir.
2. Başlangıç koşul parametreleri belirlenir. (Eylemsizlik ağırlığı w, c1 ve c2 - sırasıyla bilişsel ve sosyal güven katsayıları, popülasyon büyüklüğü, maksimum iterasyon sayısı)
3. İngilizce için 0-25 ve Türkçe için 0-28 arasında rasgele sayılar seçerek N adet parçacık başlangıç popülasyonun Eşitlik (2.4) ile üretilmesi (varsayılan anahtar) (assumed key).
4. **for** i=1:nPop
    a. Anahtar (varsayılan anahtar[i]) kullanarak metnin deşifre edilmesi.
    b. Deşifre edilen metnin uygunluk değerinin hesaplanması: f(varsayılan anahtar[i])
    c. Kişisel en iyi değerin güncellenmesi, pBestSol olarak kaydet
    d. Global en iyi değerin hesaplanması, gBestSol olarak kaydet
5. **end**
6. **% PSO Ana Döngü**
7. **for** it=1:MaxIt
8. **% Hızların güncellenmesi**
9. for i=1:Npop
    a. Eşitlik (2.5) ile Yeni parçacık hızı(i) hesaplanması;
10. **% Konumların güncellenmesi**
    a. Eşitlik (2.6)'ya göre konum güncellemesi (varsayılan anahtar[i]).
    b. Anahtar (varsayılan anahtar[i]) kullanarak metnin deşifre edilmesi.
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması: f(varsayılan anahtar[i]).
    d. Azami uygunluk değer sahip parçacığın (varsayılan anahtar[i]) bulunup pBestSol olarak kaydedilmesi

**11.** end

**12. % Kişisel en iyi değerlerin güncellenmesi**

13. end

**14. % Global en iyi değerlerin güncellenmesi**

15. End

### 3.3. DE ile Vigenére Şifreleme Yönteminin Kriptoanalizi

DE algoritmasının Vigenére şifreleme yönteminin kriptonanalizine uygulanması aşağıdaki adımlarla gerçekleşmektedir [102,105]:

1. Şifreli metin anahtar boyutu ve ilgili karakter frekansları belirlenir.
2. Başlangıç koşul parametreleri belirlenir (Maksimum iterasyon sayısı, popülasyon büyüklüğü, çaprazlama oranı ve ölçekleme faktörü).
3. İngilizce için 0-25 ve Türkçe için 0-28 arasında rasgele sayılar seçerek N adet kromozom kaynaklı başlangıç popülasyonun Eşitlik (2.4) ile üretilmesi (varsayılan anahtar).
4. **for i=1:nPop**
5. Şifre çözme anahtarı kullanılarak mevcut çözümleri (varsayılan anahtar[i] ile şifrenin çözülmesi.
6. Deşifre edilen metnin uygunluk fonksiyon f değerinin bulunması.
7. Maksimum uygunluk fonksiyon değere sahip kaynağı bulup BestSol olarak adlandırılması.
8. **end**
9. **% DE Ana Döngü**
10. **for it=1:MaxIt**
11. **for i=1:nPop**
    a. **% Mutasyon**
    b. Popülasyondaki mevcut çözümlerin (varsayılan anahtarın [i]) seçilmesi.
    c. Üç farklı kromozomun üretilmesi (r1≠r2≠r3≠i).
    d. Seçilen kromozomlardan ilk ikisinin farkının alınması (fark kromozomu [i]).
    e. Eşitlik (2.7) ile fark kromozom (fark kromozom [i]) * ölçekleme faktörü'nün uygulanması.

**f. %Çaprazlama**

g. Birinci ebeveyn olarak popülasyondaki çözümlerin (varsayılan anahtarın[i]) seçilmesi.

h. İkinci ebeveyn olarak fark kromozom (fark kromozom [i]) seçilmesi.

i. Uniform çaprazlama kullanılarak yeni kromozom (yeni kromozom [i]) Eşitlik (2.8)'e göre oluşturulması.

j. Çaprazlama uygulanarak yeni varsayılan anahtarın (yeni varsayılan anahtar[i]) üretilmesi.

k. Deşifre anahtarı olarak yeni varsayılan anahtar[i] kullanılarak şifreli metnin çözülmesi.

l. Deşifre edilen metnin uygunluk değerinin hesaplanması: f (yeni varsayılan anahtar[i]).

**m. % Seçim**

n. Daha iyi olan çözümün kurulması.

**12. end**

13. Bulunan en iyi çözüm için uygunluk değerinin kaydedilmesi.

**end**

### 3.4 ABC ile Vigenére Şifreleme Yönteminin Kriptoanalizi

ABC algoritmasının Vigenére şifreleme yönteminin kriptoanalizine uygulanması aşağıdaki adımlardan oluşmaktadır [102, 106, 107]:

1. Şifreli metin, anahtar boyutu ve gerekli karakter frekansların belirlenmesi.
2. Yapay arı koloni algoritmasının kontrol parametrelerinin başlangıç koşullarının ayarlanması: koloni boyutu (SN), limit parametresi (L), maksimum çevrim sayısı (MCN).
3. % Popülasyon dizisinin başlatılması
4. İngilizce için 0-25 ve Türkçe için 0-28 arasında arasında rasgele sayılar seçerek n adet besin kaynağı başlangıç popülasyonun Eşitlik (2.4)'de göre üretilmesi (varsayılan anahtar).
5. **% Başlangıç popülasyonunun oluşturulması**

6. **for** i=1:SN
    a. Şifre çözme anahtarı olarak kaynak [i] kullanarak metnin çözülmesi.
    b. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
7. End
8. En yüksek uygunluk değer sahip kaynağın bulunup BestSol olarak kaydedilmesi
9. **end**
10. **% ABC Ana Döngü**
11. **for** it=1:MCN
12. **% İşçi arılar**
13. **for** i=1:SN
    a. i. Çözüm civarında yeni yiyecek kaynağı pozisyonun Eşitlik (2.10) ile oluşturulması: Position(i)
    b. Şifre çözme olarak yeni konum Position(i) kullanarak metnin çözülmesi
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
    d. Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması
14. **end**
15. **% Eşitlik (2.9) ile Uygunluk değerlerinin ve Eşitlik (2.11) ile seçilme olasılıklarının hesaplanması**
16. **% Gözcü arılar**
17. **for** m=1:SN
    a. Olasılık değerine bağlı olarak bir çözüm seçilmesi ve Eşitlik (2.10) ile seçilen kaynak civarında yeni çözüm üretilmesi: Position(i)
    b. Şifre çözme olarak yeni konum Position (i) kullanarak metnin çözülmesi
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
    d. Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması
18. **end**
19. En yüksek uygunluk değerine sahip kaynağın bulunup BestSol olarak kaydedilmesi
20. **% Kaşif arılar**
21. **Eğer** tükenmiş kaynak varsa,
    a. Yeni yiyecek kaynağı pozisyonu oluşturulması: Position(i) Eşitlik (2.4);
    b. Şifre çözme olarak yeni konum Position(i) kullanarak metnin çözülmesi
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])

d. Terk etme sınır parametreleri sıfıra (0) eşitlenmesi (Kaynak tükenmesi);

**22. End**

### 3.5. BCABC ile Vigenére Şifreleme Yönteminin Kriptoanalizi

Yerel arama yeteneğini güçlendirmek için, standart yapay arı koloni algoritması ile çaprazlama işlemi entegre edilerek geliştirilen Binom çaprazlama operatörülü yapay arı koloni algoritmasının (BCABC) Vigenére kriptonanalizine uygulanması aşağıdaki gibidir [102, 106, 107]:

1. Şifreli metin, anahtar boyutu ve gerekli karakter frekansların belirlenmesi.
2. Yapay arı koloni algoritmasının kontrol parametrelerinin başlangıç koşullarının ayarlanması: koloni boyutu (SN), limit parametresi (L), maksimum çevrim sayısı (MCN).
3. % Popülasyon dizisinin başlatılması
4. İngilizce için 0-25 ve Türkçe için 0-28 arasında arasında rasgele sayılar seçerek n adet besin kaynağı başlangıç popülasyonun Eşitlik (2.4)'de göre üretilmesi (varsayılan anahtar).
5. **% Başlangıç popülasyonunun oluşturulması**
6. **for** i=1:SN
    a. Şifre çözme anahtarı olarak kaynak [i] kullanarak metnin çözülmesi.
    b. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
7. End
8. En yüksek uygunluk değer sahip kaynağın bulunup BestSol olarak kaydedilmesi
9. **end**
10. **% ABC Ana Döngü**
11. **for** it=1:MCN
12. **% İşçi arılar**
13. **for** i=1:SN
    a. i. Çözüm civarında yeni yiyecek kaynağı pozisyonun Eşitlik (2.10) ile oluşturulması: Position(i)
    b. Şifre çözme olarak yeni konum Position(i) kullanarak metnin çözülmesi

c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
d. Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması

**14. end**

15. En yüksek uygunluk değer sahip kaynağın bulunup BestSol olarak kaydedilmesi

**16. % Çaprazlama**

17. **for** i=1:SN
    a. Birinci ebeveyn olarak i seçilmesi;
    b. İkinci ebeveyn olarak i'den farklı olan j seçilmesi
    c. Binom çaprazlama kullanarak arı konumunun oluşturulması: Position(i)
    d. Yiyecek kaynağı pozisyonu (i) deşifre anahtarı olarak kullanarak metnin çözülmesi.
    e. Deşifre metnin uygunluk değerinin hesaplanması: f(kaynak[i])
    f. Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması

**18. end**

**19. %Eşitlik (2.9) ile Uygunluk değerlerinin ve Eşitlik (2.11) ile seçilme olasılıklarının hesaplanması**

**20. % Gözcü arılar**

21. **for** m=1:SN
    a. Olasılık değerine bağlı olarak bir çözüm seçilmesi ve Eşitlik (2.10) ile seçilen kaynak civarında yeni çözüm üretilmesi: Position(i)
    b. Şifre çözme olarak yeni konum Position (i) kullanarak metnin çözülmesi
    c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
    d. Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanması

**22. end**

23. En yüksek uygunluk değerine sahip kaynağın bulunup BestSol olarak kaydedilmesi

**24. % Kaşif arılar**

25. **Eğer** tükenmiş kaynak varsa,
    a. Yeni yiyecek kaynağı pozisyonu oluşturulması: Position(i) Eşitlik (2.4);

b. Şifre çözme olarak yeni konum Position(i) kullanarak metnin çözülmesi
c. Deşifre edilen metnin uygunluk değerinin hesaplanması, f(kaynak[i])
d. Terk etme sınır parametreleri sıfıra (0) eşitlenmesi (Kaynak tükenmesi);

**26. End**

### 3.6. BCABC ile Örnek Uygulama

**Kullanılan anahtar:** ERCİY

**Düz metin:**

okulmüdürümüzöğretmenlergünümünasebetiylebirkonuşmayapmamıricaettioldumolasıbutürkonuşmalardansıkılı

**Şifreli metin:**

Şbvüıwuyxrrkwqçüüüvascgxcwdyvrsrtnzıikfhıskxğşdvbıemcwıeçjxfgrgcönenmpreninmsvcrübpypyçcüyüucynmbjüe

Çaprazlama oranı (CR) = 0.2

Limit (L) = 3

İterasyon = 2

Başlangıç (x) popülasyon Eşitlik (2.4) ile üretilmiş 5 karakterli (anahtar boyutu) 4 anahtardan (popülasyon boyutu) oluşmaktadır. Algoritmanın bütün adımlarında anahtar karakterleri için karakter kodları kullanılmıştır ve tüm işlemler karakter koduna denk gelen sayılarla yapılmıştır, örneğin yerel aramada harf üretilmiyor, harfe denk gelen sayı üzerinden işlemler yapılmıştır.

| | | | | |
|---|---|---|---|---|
| S | Y | W | C | J |
| J | R | P | U | J |
| F | C | T | F | K |
| O | G | Q | M | E |

Oluşturulan anahtarlar şifreli metnin şifresini çözmek için tek tek kullanılır. Daha sonra Eşitlik (3.1) ile şifresi çözülmüş metinler için frekans analizi yapılır ve her anahtarın uygunluk değeri Eşitlik (2.9) ile elde edilir.

| | | | | | |
|---|---|---|---|---|---|
| S | Y | W | C | J | 10.76 |
| J | R | P | U | J | 16.12 |
| F | C | T | F | K | 14.61 |
| O | G | Q | M | E | 8.55 |

Daha sonra en yüksek uygunluk değerine sahip olan anahtar bulunup BestSol olarak kaydedilir.

| | | | | | |
|---|---|---|---|---|---|
| J | R | P | U | J | 16.12 |

**İşçi arı sayısı kadar devem et:**

Birinci anahtar için Eşitlik (2.10) ile yerel arama yaparak yeni çözümlerin üretilmesi. Rastgele seçilen komşunun k=4, $\Phi_{ij}$ = 0.8302 olduğunu kabul edelim. Uygunluk değerleri aşağıdaki gibi elde edilir.

| | | | | | | |
|---|---|---|---|---|---|---|
| *f(yeni anahtar)* | P | R | T | C | F | 14.81 |
| *f(eski anahtar)* | S | Y | W | C | J | 10.76 |

Yeni ve eski anahtarlar arasında aç gözlü seleksiyon yapılır yeni anahtar daha iyi olduğunda eski anahtar ile yer değiştirir, aksi takdirde limit değeri 1 artar. Bu durumda 10.76 <14.81 olduğu için yeni anahtar eski anahtar ile yer değişir ve limit değeri 0'a eşit olur.

İkinci anahtar için Eşitlik (2.10) ile yerel arama yaparak yeni çözümlerin üretilmesi. Rastgele seçilen komşu k=3, $\Phi_{ij}$ = 0.0761 olsun. Uygunluk değerleri aşağıdaki gibi elde edilir.

| | | | | | | |
|---|---|---|---|---|---|---|
| *f(yeni anahtar)* | G | G | M | O | J | 11.79 |
| *f(eski anahtar)* | J | R | P | U | J | 16.12 |

Bu durumda 11.79 <16.12 olduğu için yeni anahtar eski anahtar ile yer değiştiremez ve limit değeri 1 artar. İşçi arı aşaması sonunda elde edilen anahtar setimiz aşağıdaki gibi olacaktır.

| | | | | | |
|---|---|---|---|---|---|
| **P** | **R** | **T** | **C** | **F** | **14.81** |
| J | R | P | U | J | 16.12 |
| F | C | T | F | K | 14.61 |
| O | G | Q | M | E | 8.55 |

**Çaprazlama aşaması:**

Popülasyondaki tüm çözümler için iki ebeveyn seçilerek binom çaprazlama uygulanır. Birinci ebeveyn olarak i=1 indeksin seçilmesi, ikinci ebeveyn olarak ise rassal olarak i'den farklı olan y=2 indeksinin seçildiğini kabul edilelim. Elde edilen anahtar (z) için uygunluk değeri hesaplanır. Daha sonra eski anahtar (i) ile arasında aç gözlü seleksiyon yapılır. Bu durumda 11.79 <14.81 olduğu için yeni anahtar eski anahtar ile yer değiştirmez ve limit değeri 1 artar.

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | P | R | T | C | F | 14.81 |
| Y2 | J | R | P | U | J | 16.17 |
| **Z** | **P** | **R** | **P** | **C** | **J** | **11.79** |

i=2 için ikinci ebeveyn y=1 indeksin seçilmiş olsun. Elde edilen anahtar (z) için uygunluk değeri hesaplanır. Daha sonra eski anahtar (i) ile arasında aç gözlü seleksiyon yapılır. Bu durumda 14.18 <16.12 olduğu için yeni anahtar eski anahtar ile yer değiştirmez ve limit değeri 1 artar.

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | J | R | P | U | J | 16.17 |
| Y2 | P | R | T | C | F | 14.81 |
| **Z** | **J** | **R** | **T** | **U** | **J** | **14.18** |

i=3 için ikinci ebeveyn olarak y=4 indeksin seçilmiş olsun. Elde edilen anahtar (z) için uygunluk değeri hesaplanır. Daha sonra eski anahtar (i) ile arasında aç gözlü seleksiyon yapılır. Bu durumda 8.63 <14.18 olduğu için yeni anahtar eski anahtar ile yer değiştirmez ve limit değeri 1 artar.

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | F | C | T | F | K | 14.61 |
| Y2 | O | G | Q | M | E | 8.55 |
| Z | **F** | **G** | **Q** | **M** | **E** | **8.63** |

i=4 için ikinci ebeveyn y=1 indeksin seçilmiş olsun. Elde edilen anahtar (z) için uygunluk değeri hesaplanır. Daha sonra eski anahtar (i) ile arasında aç gözlü seleksiyon yapılır. Bu durumda 8.55 <13.40 olduğu için yeni anahtar eski anahtar ile yer değiştirir ve limit değeri 0'a eşit olur.

| | | | | | | |
|---|---|---|---|---|---|---|
| X1 | O | G | Q | M | E | 8.55 |
| Y2 | P | R | T | C | F | 14.81 |
| Z | **P** | **G** | **T** | **C** | **F** | **13.40** |

Çaprazlama sonunda elde edilen anahtar setimiz aşağıdaki gibi olacaktır.

| | | | | | |
|---|---|---|---|---|---|
| **P** | **R** | **T** | **C** | **F** | **14.81** |
| J | R | P | U | J | 16.12 |
| F | C | T | E | K | 14.61 |
| **P** | **G** | **T** | **C** | **F** | **13.40** |

Eşitlik (2.9) ile uygunluk değerleri ve Eşitlik (2.11) ile seçilme olasılıkları hesaplanır.

| **F** | **P** |
|---|---|
| 0.3660 | 0.2482 |
| 0.3348 | 0.2270 |
| 0.3711 | 0.2516 |
| 0.4027 | 0.2731 |

**Gözcü arı sayısı kadar devem et:**

Olasılık değerine bağlı olarak bir çözüm seçilmesi bu durum için i =1 ve Eşitlik (2.10) ile yeni çözüm üretir daha sonra uygunluk değeri hesaplanır.

| | | | | | |
|---|---|---|---|---|---|
| L | R | R | I | N | 16.22 |

Kaynakların kalitesine bağlı olarak aç gözlü seçimin uygulanır, bu durumda 14.81<16.22 olduğu için yeni anahtar ile eski anahtar yer değiştirir ve limit değeri 0'a eşit olur.

Yine olasılık değerine göre i=3 seçilir ve Eşitlik (2.10) ile yeni çözüm üretir ve uygunluk değeri hesaplanır.

H S R J F 17.19

Aç gözlü seçim uygulanır, bu durumda 14.16<17.19 olduğu için yeni anahtar ile eski anahtar yer değiştirir ve limit değeri 0'a eşit olur. Yeni anahtar setimiz aşağıdaki gibi olur.

**L R R I H 16.22**
J R P U J 16.12
**H S R J F 17.19**
**P G T C F 13.40**

Daha sonra en yüksek uygunluk değerine sahip olan anahtar bulunup BestSol olarak kaydedilir.

H S R J P 17.19

**Kaşif arı aşaması:**

Bir çevrim boyunca limit için elde edilen sonuçlar aşağıdaki gibidir.

0 2 0 0

Limit değeri 3 olduğundan dolayı tükenmiş kaynak olmamıştır eğer tükenmiş kaynak varsa, Eşitlik (2.4) ile yeni anahtarlar oluşturulur ve uygunluk değerleri elde edilir. Bu işlemeler koşulların sağlanmasına kadar devem eder.

# 4. BÖLÜM

# DENEYSEL ÇALIŞMALAR ve BULGULAR

Bu tez çalışmasında, donanımsal olarak 48 Adet Acer-Veritton İşlemci: Intel Core i7-2600K CPU @3.40GHZ x 4 RAM: 3.8GB Bellek, Ekran Kartı: NVIDIA Corporation GF [GeForce GT 430], Disk Alanı: 500GB ve Windows 7 Professional konfigürasyonuna sahip Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Bilgisayar laboratuvarı kullanılmıştır, yazılımsal olarak Matlab programı kullanılmıştır.

Tez kapsamında Türkçe ve İngilizce dil kullanımı durumları incelenmiştir ve bu diller için harf frekans analizi değerlendirmeleri yapılmıştır. Tez kapsamında 5 farklı uzunlukta örnek anahtar kullanılmıştır (5, 10, 15, 20, 25) aynı anahtarlar Türkçe ve İngilizce için 4 farklı uzunluktaki (250, 500, 750, 1000) şifreli metinleri çözmek için kullanılmıştır. Çalışma kapsamında açık ve şifreli metinlerin her ikisinin de alfabe bulunan harfler dışında karakter içermediği kabul edilmektedir. Yani İngilizce metinlerde 26, Türkçe metinlerde 29 karakter ile sınırlandırma söz konudur.

Vigenére şifrelemenin analizi amacıyla GA, PSO, DE, temel ABC algoritması ve BCABC algoritmaları kullanılmıştır.

## 4.1. Araştırma Modeli:

Olası anahtarlar kümesi alfabedeki tüm harflerin permütasonlarının kümesine denk geldiğinden, doğru anahtarı bulmak için kaba kuvvet (brute force) kullanarak yapılan kriptoanalizin hesaplama maliyeti verimli olmaktan çok uzaktır. Bu nedenle, optimal anahtarı bulmak için sezgisel algoritmaların ortaya çıkardığı sistematik arama kullanmak daha uygundur.

Çalışma kapsamında Vigenére şifrelemenin kriptoanalizi amacıyla GA, PSO, DE, ABC ve BCABC algoritmaları kullanılarak beş farklı deney uygulanmıştır. Bu deneylerin ilkinde, algoritmaların kontrol parametrelerine hassasiyetlerini göstermek ve çalışmaların devamında kullanılacak parametre değerlerini belirlemek amaçlanmıştır. İkinci deneyde, GA, PSO, DE, ABC ve BCABC algoritmalarının her biri ayrı olarak Vigenére şifrelemenin analizini yürütmek için uygulanmış ve her algoritma için metin uzunluğu ile anahtar uzunluğuna göre elde edilen en düşük ve en yüksek anahtar doğru karakter sayısı (ADKS) için ortalama ve standart sapma değerleri bulunmuştur. Ek olarak şifreli metin ile anahtarı için en düşük ve en yüksek uygunluk değerlerinin ortalama ve standart sapma değeri bulunmuştur. Üçüncü deneyde ise, sunulan algoritmanın güçlü ve zayıf yönlerini ortaya çıkarmak amacıyla birinci deneyde belirlenen en uygun kontrol parametre değerleri kullanılarak GA, PSO, DE, ABC ve BCABC algoritmaları ile bulunan ADKS ve uygunluk değerlerinin en iyi, ortalama ve standart sapma değerleri arasında bir karşılaştırma gerçekleştirilmiştir. Sonraki kısımda ise, BCABC algoritması kontrol algoritması olarak seçilerek ve GA, PSO, DE, ABC algoritmaları ile hipotez testi aracılığı ile istatistiksel kıyaslama yapılmıştır.

## 4.2. Deneysel Çalışmalar

Her bir deneyde, 5, 10, 15, 20 ve 25 gibi 5 farklı karakter uzunluğunda anahtarlar ile 250, 500, 750 ve 1000 karakterli gibi 4 farklı uzunlukta açık metinlerden oluşturulmuş şifreli metinler kullanılmıştır.

### 4.2.1. Deneysel Çalışma 1: Parametre Değerlerinin İncelenmesi

Algoritmaların performansı, kontrol parametrelerinin değerleri tarafından etkilediğinden, bu bölümde, sezgisel algoritmaların parametre hassasiyetleri incelenmiştir. Algoritmaların temel kontrol parametreleri popülasyon boyutu ve üretilecek maksimum yeni birey sayısıdır (değerlendirme sayısı). Sezgisel algoritmaların tamamında sonlandırma kriteri olarak 30000 değerlendirme sayısına erişilmiş olması kabul edilmiştir. Ayrıca, arama uzayı İngilizce metinler için [0, 25] ve Türkçe metinler için [0, 28] olarak sınırlandırılmıştır. Bütün algoritmalar için popülasyon boyutu 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri için analiz edilmiştir. Yine burada 5, 10, 15, 20 ve 25 uzunlukta anahtarlar ve Vigenére ile şifrelenmiş İngilizce metinlerden olmak üzere 250, 500, 750 ve 1000 karakter

uzunluğunda farklı örnekler kullanılmıştır. Algoritmalar her bir konfigürasyon için de rasgele örnekleme ile başlamak suretiyle 30 defa çalıştırılmıştır. Deney 1'de analiz edilmiş olan parametre aralıkları Tablo 4.1'de listelenmiştir.

Tablo 4.1. Deney 1'de denenen parametre değerleri.

| **Parametreler** | **GA** | **PSO** | **DE** | **ABC** | **BCABC** |
|---|---|---|---|---|---|
| Popülasyon boyutu (PS): | 10, 20, 30, 40, 50, 100, 150, 200, 250 | 10, 20, 30, 40, 50, 100, 150, 200, 250 | 10, 20, 30, 40, 50, 100, 150, 200, 250 | 10, 20, 30, 40, 50, 100, 150, 200, 250 | 10, 20, 30, 40, 50, 100, 150, 200, 250 |
| Çaprazlama oranı (CR): | 0.1, 0.25, 0.50, 0.75, 1 | | 0.1, 0.2, 0.5 | | 0.1, 0.2, 0.5 |
| Mutasyon oranı (MP): | 0.025, 0.050, 0.1, 0.20, 0.25, 0.50, 0.75, 1 | | | | |
| Sosyal bileşen (c1): | | 0.50, 0.75, 1, 1.25, 1.50, 1.75, 2 | | | |
| Kavramsal bileşen (c2): | | 0.50, 0.75, 1, 1.25, 1.50, 1.75, 2 | | | |
| Eylemsizlik ağırlığı (w): | | 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1 | | | |
| Ölçekleme faktörü (F) | | | 0.1, 0.25, 0.5, 0.75, 1 | | |
| Limit(L): | | | | 0.25, 0.50, 0.75, 1 | 0.25, 0.50, 0.75, 1 |

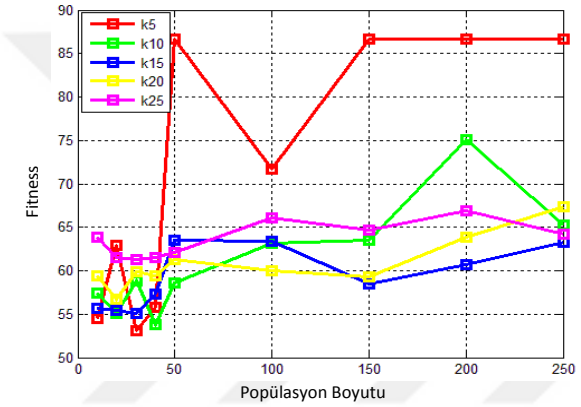
#### 4.2.1.1 GA'nın Kriptoanaliz Üzerinde Parametre Değerlerinin İncelenmesi

GA için daha önce belirtilen farklı uzunluktaki anahtar ve şifrelenmiş metinler kullanarak Tablo 4.1'de gösterildiği gibi popülasyon boyutu olarak 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri incelenmiştir. Dolayısıyla, sadece en iyi popülasyon boyutunu seçmek için 9× 4 × 5 = 180 farklı konfigürasyon denenmiştir. Çaprazlama oranı (CR) için 0.1, 0.25, 0.50, 0.75, 1 değerleri için çalışmalar tekrar edilmiştir. Çaprazlama için toplamda 5 × 4 × 5 = 100 farklı konfigürasyon test edilmiştir. Sekiz farklı mutasyon oranı (MP) (0.025, 0.050, 0.10, 0.20, 0.25, 0.50, 0.75, 1) için ise 8 × 4 × 5 = 160 farklı konfigürasyon test edilmiştir. Her bir konfigürasyon 30 kez koşulmuştur.

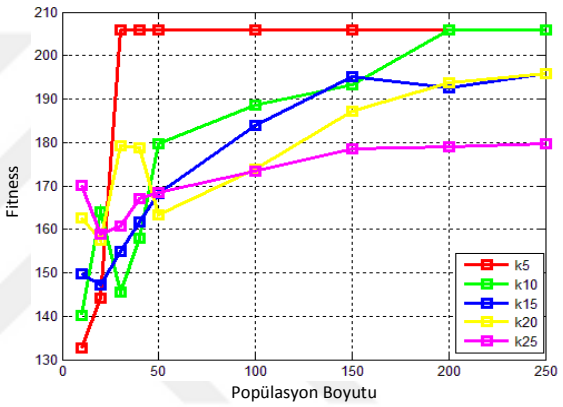
Algoritmanın her bir duruma ait 30 kez koşulmasında Eşitlik (3.1) ile hesaplanan uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri farklı popülasyon boyutları ve farklı uzunluktaki şifrelenmiş metinler için Şekil 4.1 ve Şekil 4.2'de

verilmiştir. Sonuçta, GA'nın farklı uzunluktaki Vigenére şifreli metinler için popülasyon boyutu aralığı 200-250 iken iyi sonuçlar ürettiği görülmüştür. Bunun dışındaki durumlarda ise uygunluk değerlerinin düşük seviyede kaldığı görülmüştür.

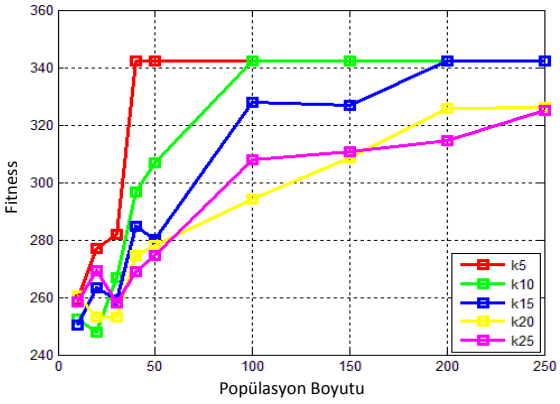
Şekil 4.3-4.6'da ise, GA'nın farklı çaprazlama ve mutasyon oranları ile elde edilen en iyi ve ortalama değerleri gösterilmektedir. En yüksek uygunluk değerleri CR 0.75 ve MP 0.75 olduğunda elde edilmiş. dışındaki durumlarda, algoritma performansının ya değişmediği ya da düştüğü görülmüştür.



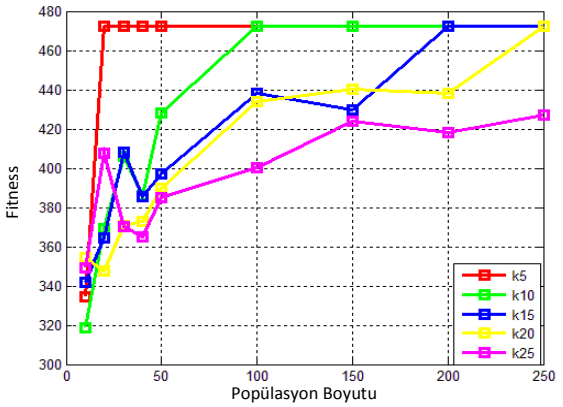

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

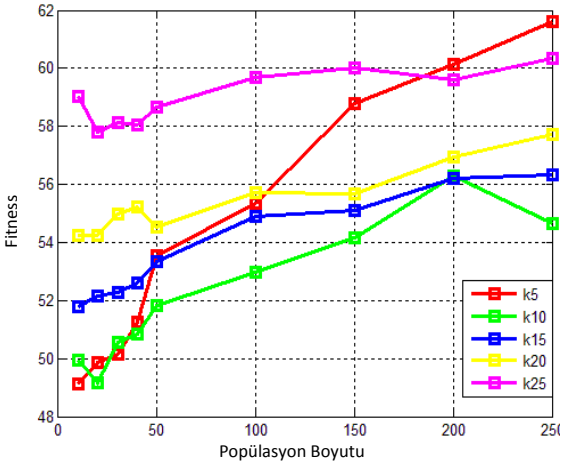



Şifrelenmiş metin uzunluğu = 750 karakter

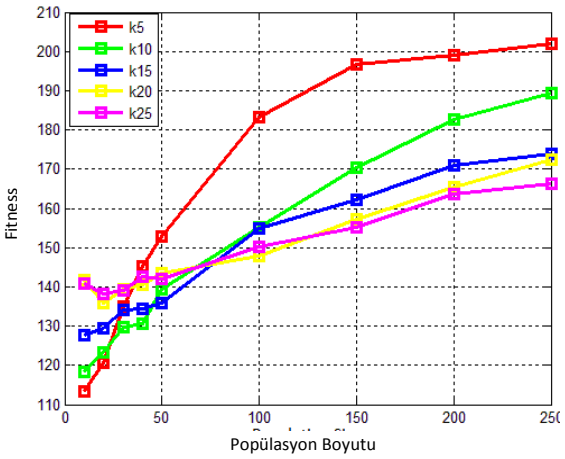



Şifrelenmiş metin uzunluğu = 1000 karakter

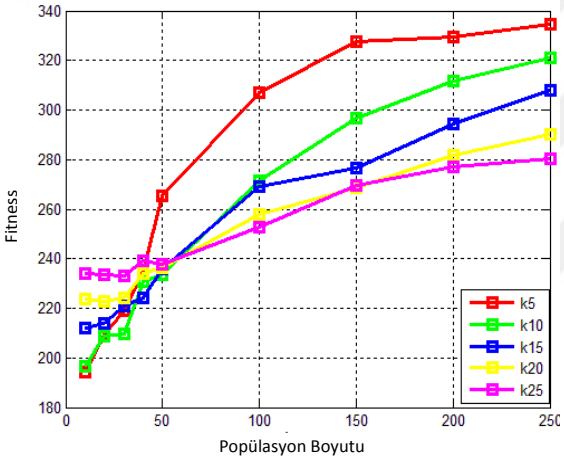
*Şekil 4.1.* GA algoritmasının farklı popülasyon boyutları kullanılarak ürettiği en iyi değerler



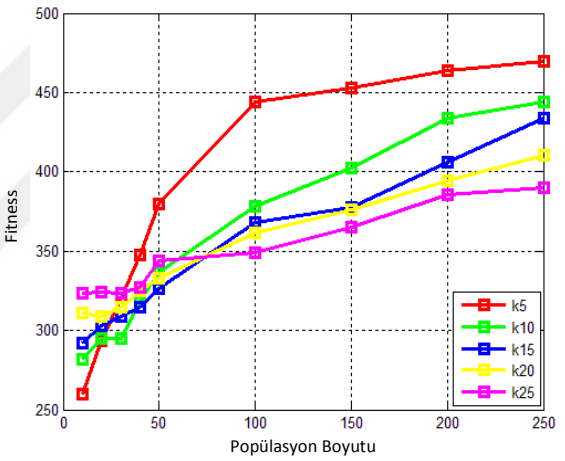

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

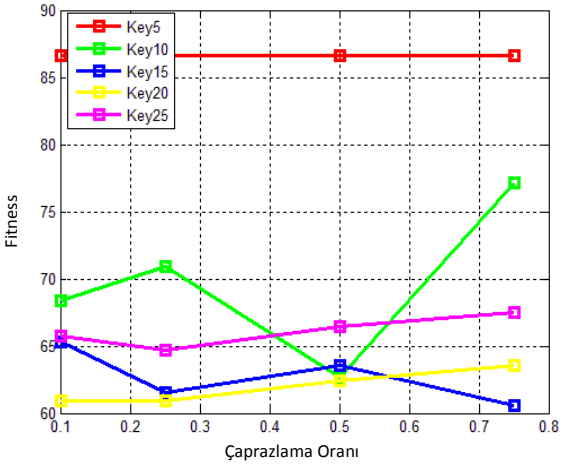



Şifrelenmiş metin uzunluğu = 750 karakter

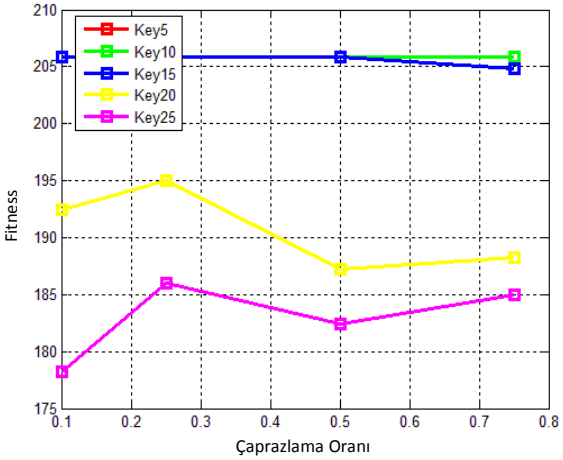



Şifrelenmiş metin uzunluğu = 1000 karakter

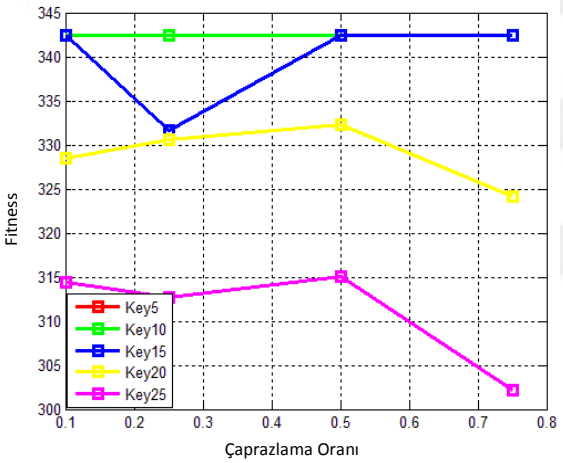
*Şekil 4.2.* GA algoritmasının farklı popülasyon boyutları kullanılarak ürettiği ortalama değerler



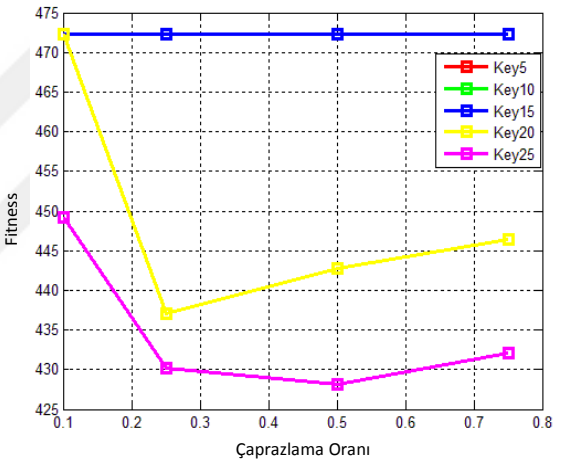

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

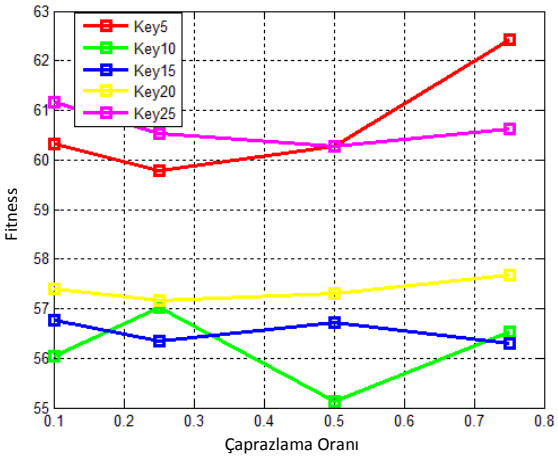



Şifrelenmiş metin uzunluğu = 750 karakter

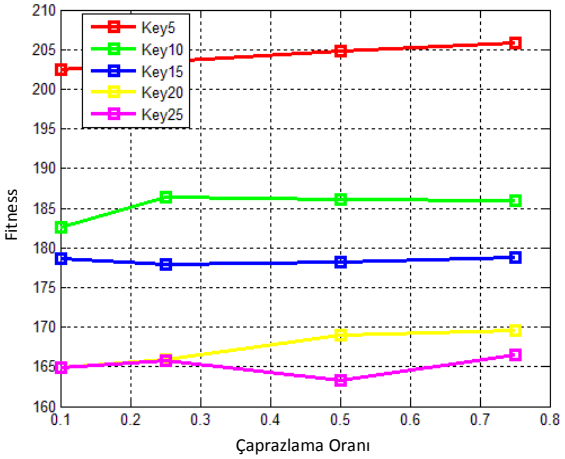



Şifrelenmiş metin uzunluğu = 1000 karakter

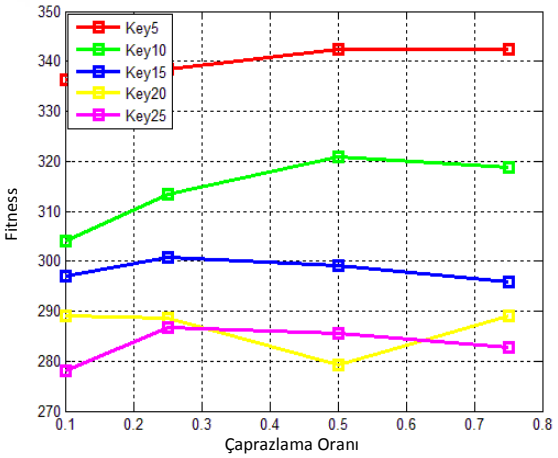
*Şekil 4.3.* GA algoritmasının farklı çaprazlama oranları kullanılarak ürettiği en iyi değerler.



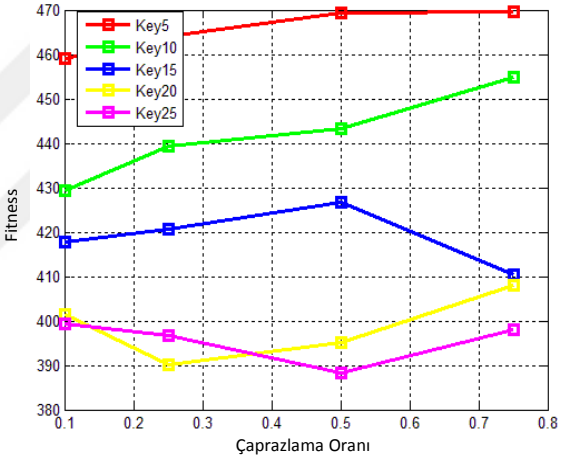

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

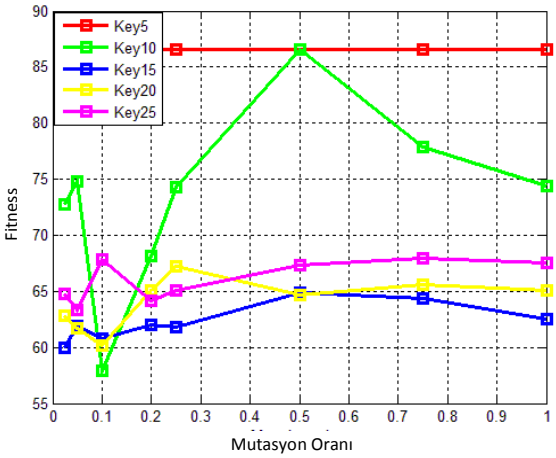



Şifrelenmiş metin uzunluğu = 750 karakter

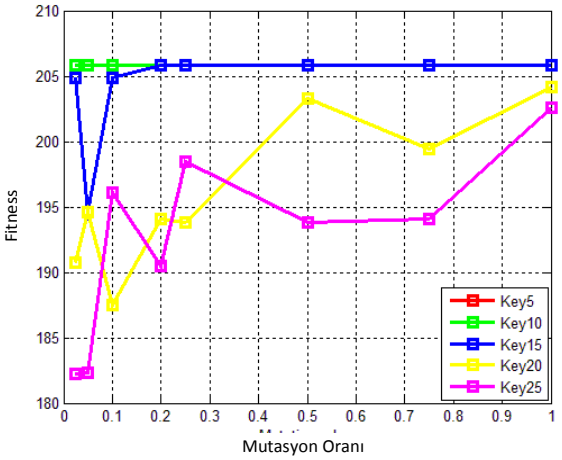



Şifrelenmiş metin uzunluğu = 1000 karakter

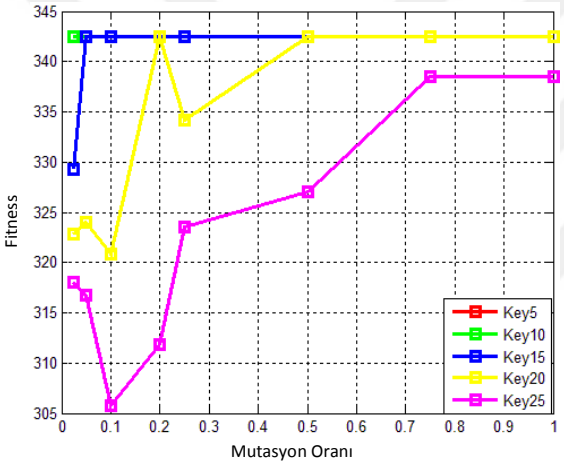
*Şekil 4.4.* GA algoritmasının farklı çaprazlama oranları kullanılarak ürettiği ortalama değerler.



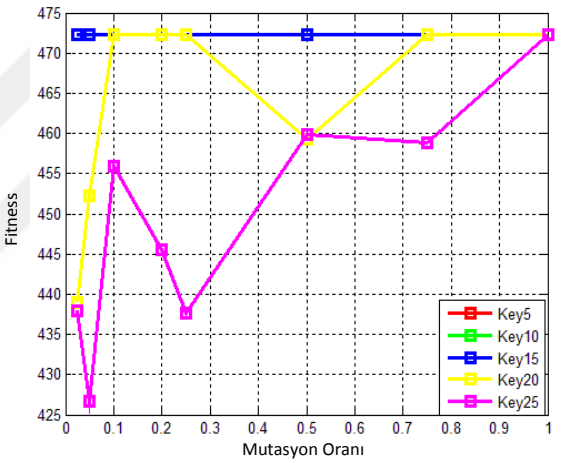

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

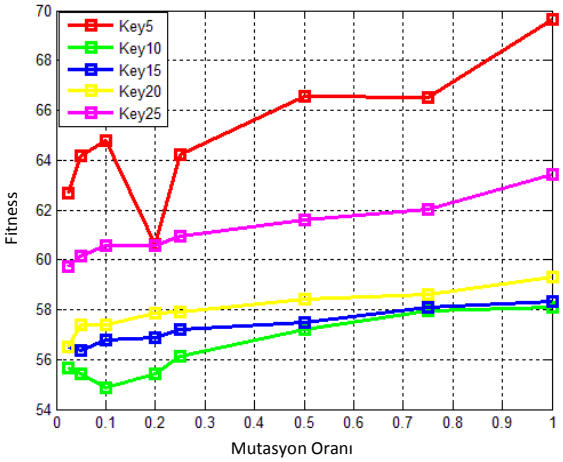



Şifrelenmiş metin uzunluğu = 750 karakter

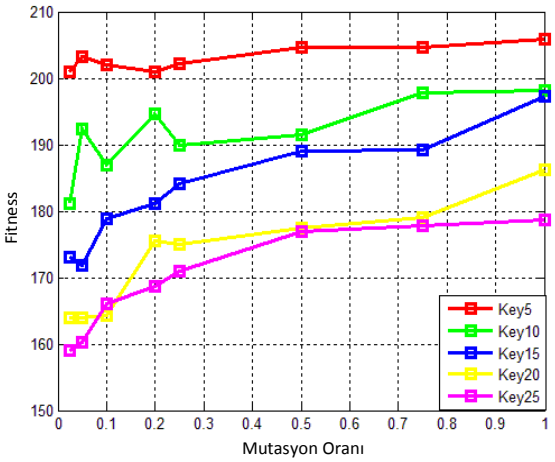



Şifrelenmiş metin uzunluğu = 1000 karakter

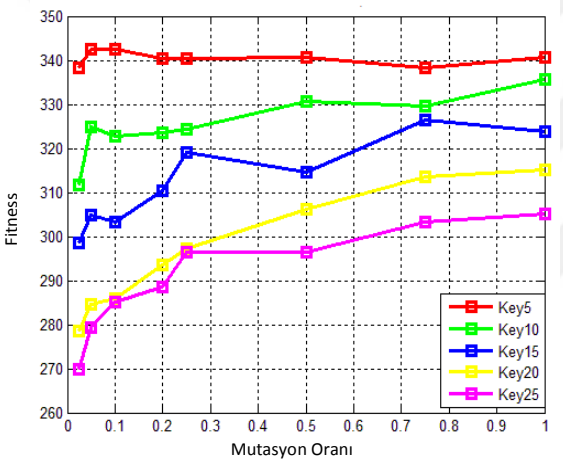
*Şekil 4.5.* GA algoritmasının farklı mutasyon oranları kullanılarak ürettiği en iyi değerler.



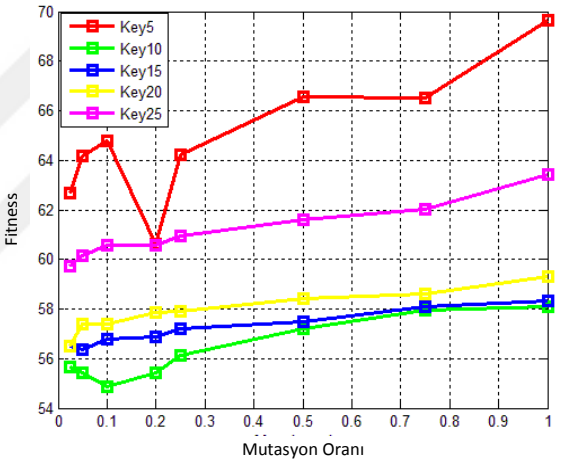

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter




Şifrelenmiş metin uzunluğu = 750 karakter




Şifrelenmiş metin uzunluğu = 1000 karakter

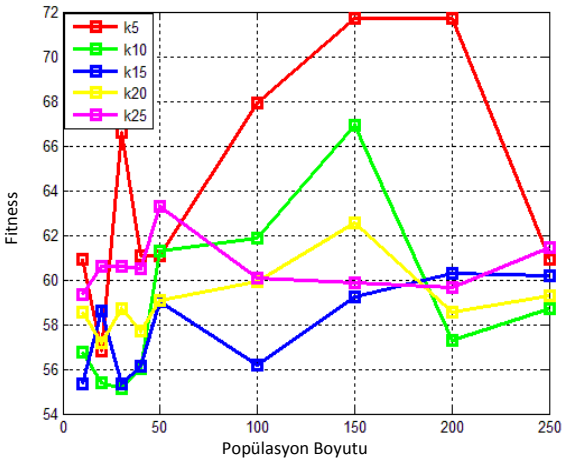
*Şekil 4.6.* GA algoritmasının farklı mutasyon oranları kullanılarak ürettiği ortalama değerler.

#### 4.2.1.2. PSO'nun Kriptoanaliz Üzerinde Parametre Değerlerinin İncelenmesi

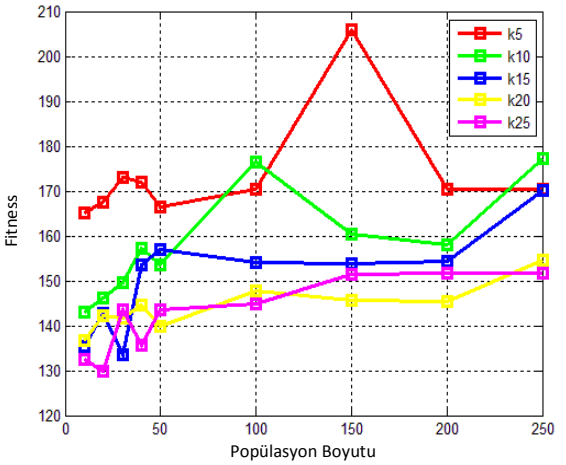
PSO algoritmasında, 5, 10, 15, 20 ve 25 uzunluğunda farklı anahtarlar ve 250, 500, 750 ve 1000 karakter uzunluğunda Vigenére ile şifrelenmiş metinler üzerinde popülasyon boyutu olarak 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri incelenmiştir. PSO algoritmasının ortak parametreler dışında kendine has kontrol parametrelerinden birisi sosyal bileşen (c1) için yedi farklı değer (0.50, 0.75, 1, 1.25, 1.50, 1.75, 2) yani 7 × 4 × 5 = 140 farklı konfigürasyon için çalışmalar tekrar edilirken bilişsel bileşen (c2) için de yedi farklı değer (0.50, 0.75, 1, 1.25, 1.50, 1.75, 2), yani 7 × 4 × 5 = 140 farklı konfigürasyon incelendi. Eylemsizlik ağırlığı (w) dokuz farklı değer (0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1) için, yani 9 × 4 × 5 = 180 farklı konfigürasyon için deneyler tekrar edildi. Toplamda 460 farklı konfigürasyon test edilmiştir. Her bir durum 30 kez koşulmuştur.

Algoritmanın her bir durumda 30 koşmasına ait Eşitlik (3.1) ile hesaplanan uygunluk fonksiyonun ortalama ve en iyi değerleri farklı popülasyon boyutları ve farklı uzunluktaki şifreli metinler için Şekil 4.7 ve Şekil 4.8'de verilmiştir. Sonuç olarak, popülasyon boyutu aralığı 200-250 iken en iyi sonuçların elde edildiği görülmüştür. Şifreli metnin uzunluğu 1000 karakter olduğunda bunu açık şekilde görmek mümkündür.

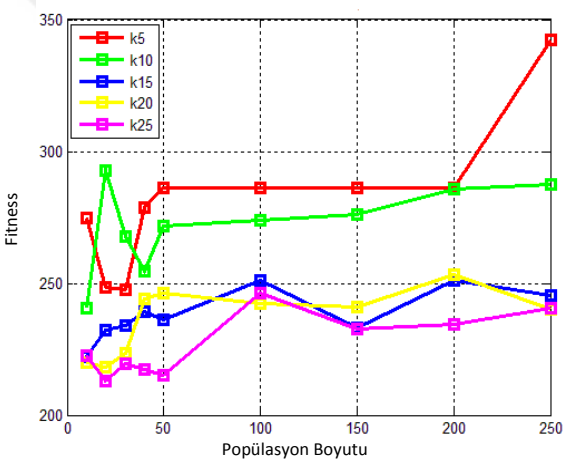
Şekil 4.9-4.14, PSO algoritmasının farklı c1, c2, ve w değerleri ile elde edilen en iyi değerleri ve ortalama değerleri grafiksel olarak göstermektedir. Burada, PSO'nun en yüksek uygunluk değerlerini c1 değerinin 1.5-2 aralığında, c2 değerinin 1.5-1.75 aralığında ve w için 0.6-0.7 aralığında olduğunda bulduğu görülmüştür. Bunun dışındaki durumlarda, algoritma performansının ya değişmediği ya da düştüğü görülmüştür.



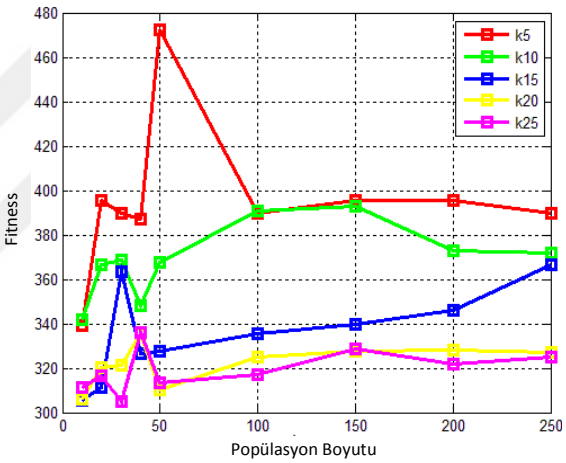

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

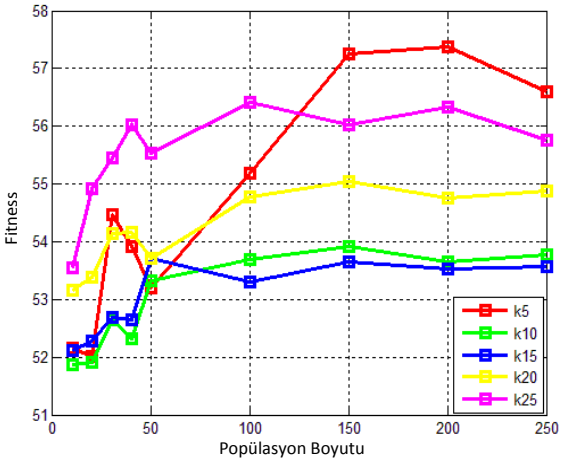



Şifrelenmiş metin uzunluğu = 750 karakter

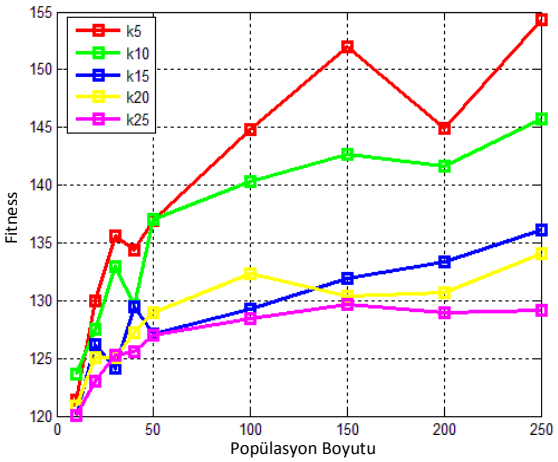



Şifrelenmiş metin uzunluğu = 1000 karakter

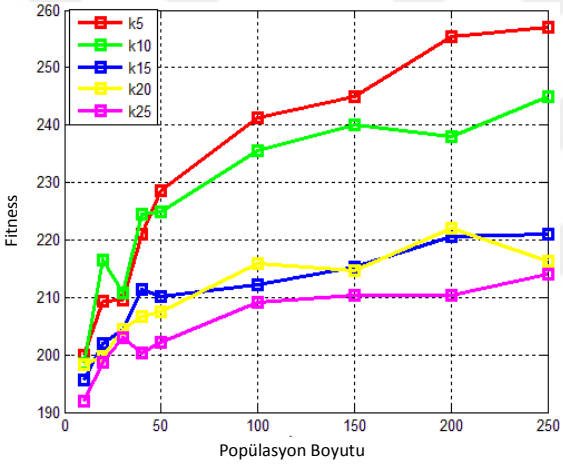
*Şekil 4.7.* PSO algoritmasının farklı popülasyon boyutları kullanılarak ürettiği en iyi değerler



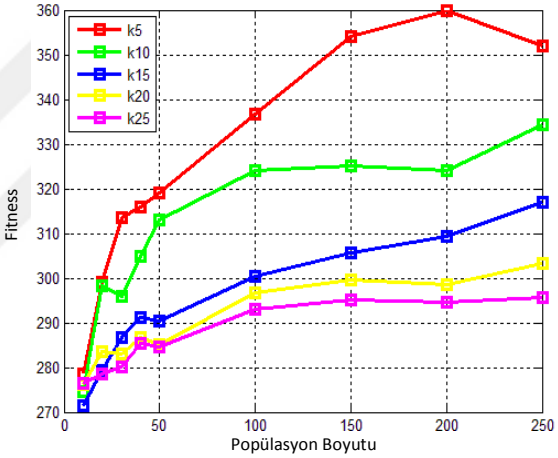

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

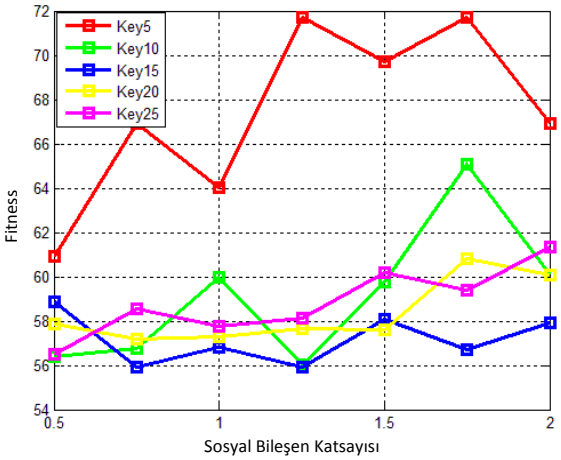



Şifrelenmiş metin uzunluğu = 750 karakter

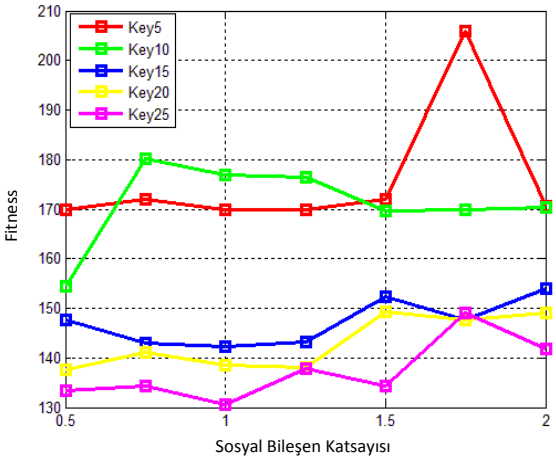



Şifrelenmiş metin uzunluğu = 1000 karakter

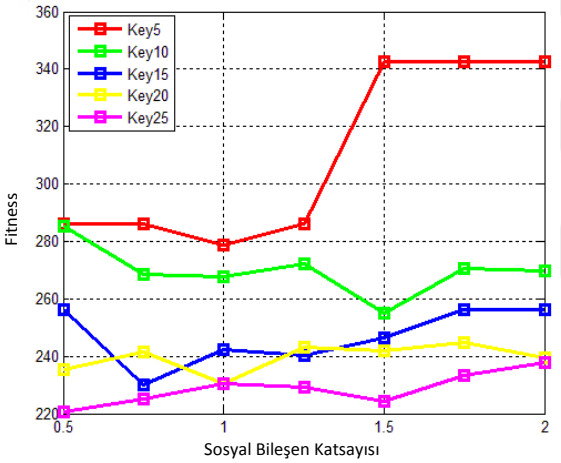
*Şekil 4.8.* PSO algoritmasının farklı popülasyon boyutları kullanılarak ürettiği ortalama değerler



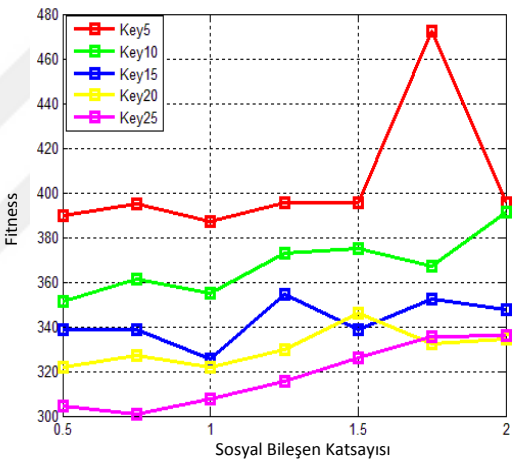

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

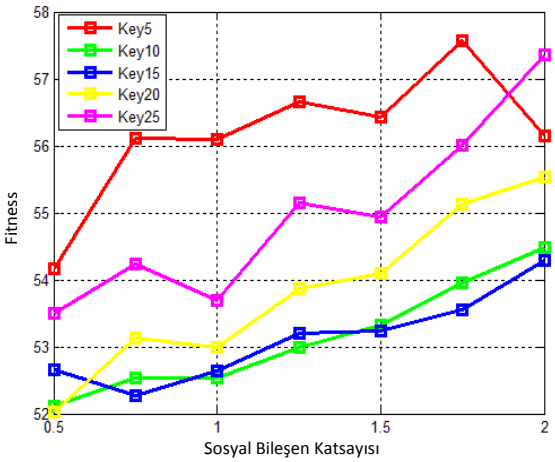



Şifrelenmiş metin uzunluğu = 750 karakter

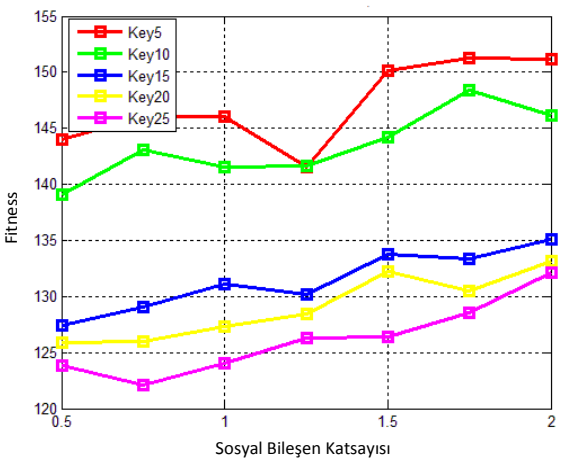



Şifrelenmiş metin uzunluğu = 1000 karakter

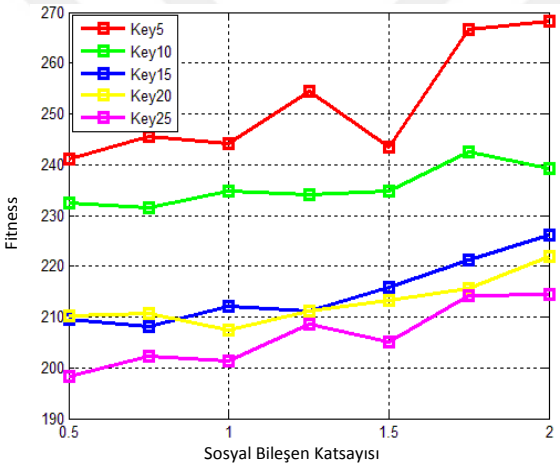
*Şekil 4.9.* PSO algoritmasının farklı c1 değerleri kullanılarak ürettiği en iyi değerler.



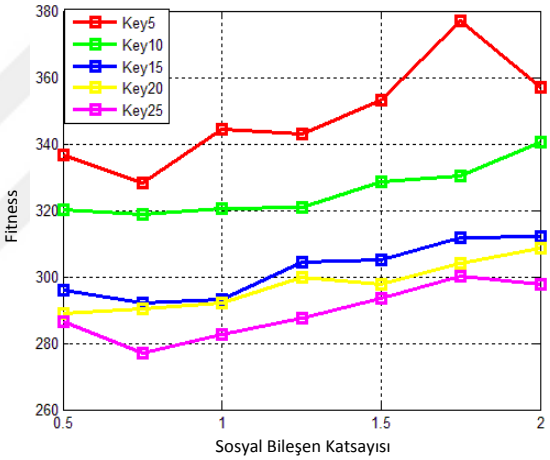

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

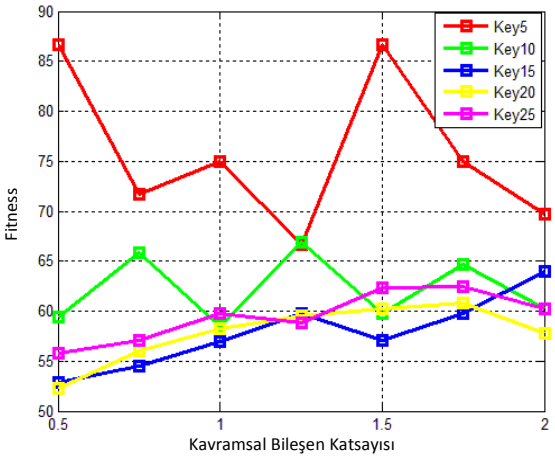



Şifrelenmiş metin uzunluğu = 750 karakter

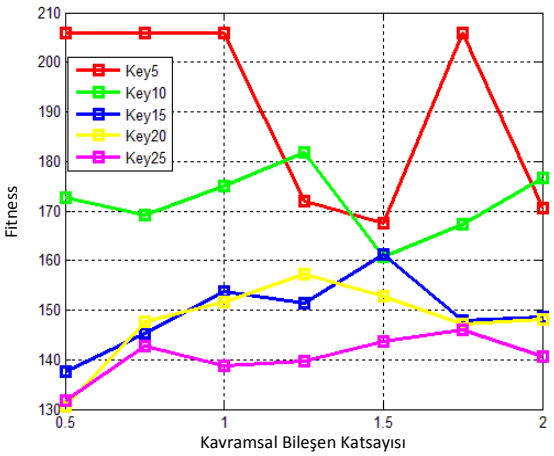



Şifrelenmiş metin uzunluğu = 1000 karakter

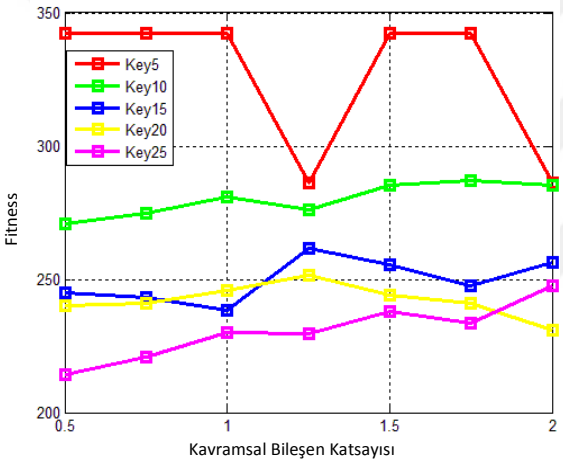
*Şekil 4.10.* PSO algoritmasının farklı c1 değerleri kullanılarak ürettiği ortalama değerler



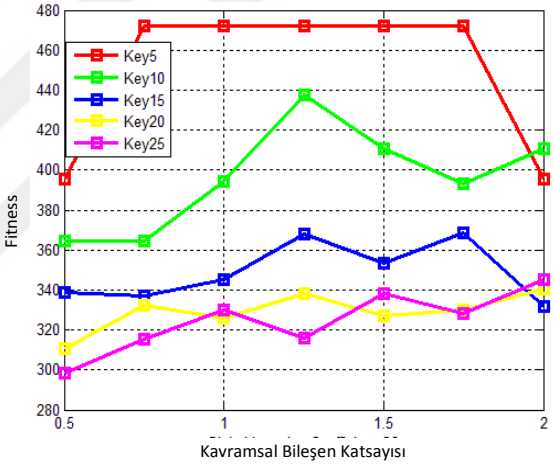

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

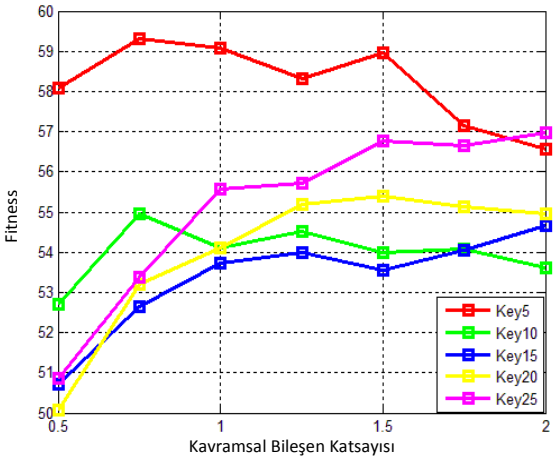



Şifrelenmiş metin uzunluğu = 750 karakter

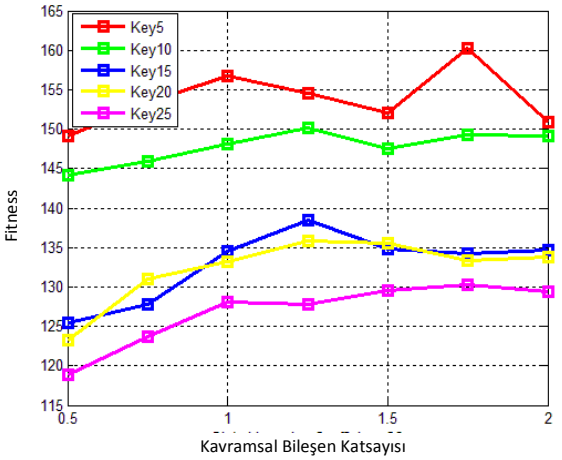



Şifrelenmiş metin uzunluğu = 1000 karakter

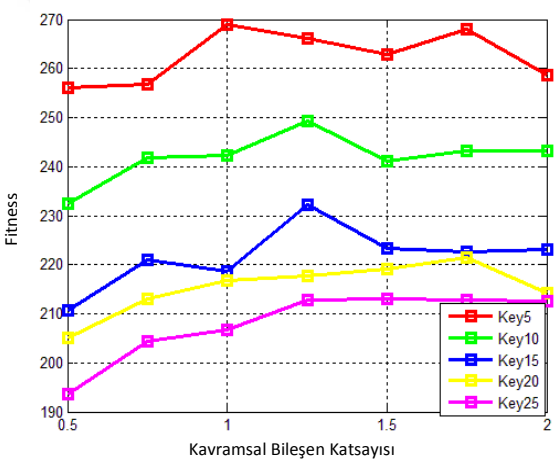
*Şekil 4.11.* PSO algoritmasının farklı c2 değerleri kullanılarak ürettiği en iyi değerler.



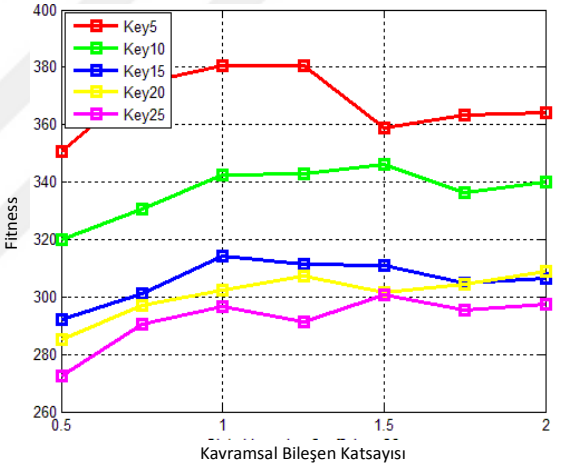

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

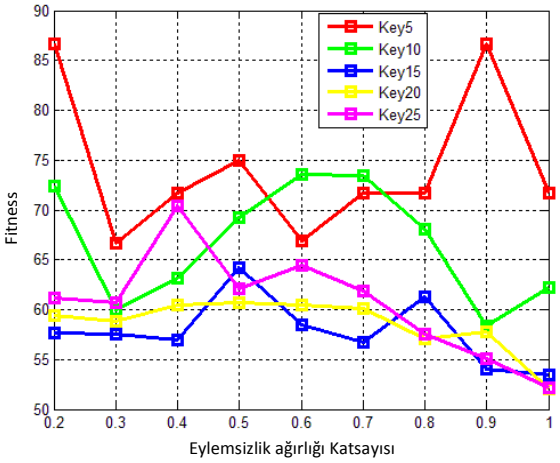



Şifrelenmiş metin uzunluğu = 750 karakter

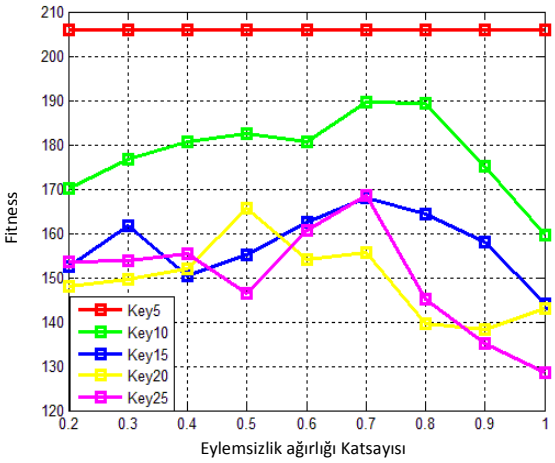



Şifrelenmiş metin uzunluğu = 1000 karakter

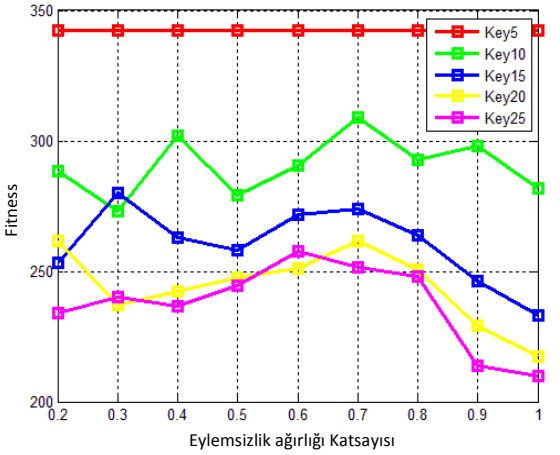
*Şekil 4.12.* PSO algoritmasının farklı c2 değerler kullanılarak ürettiği ortalama değerler



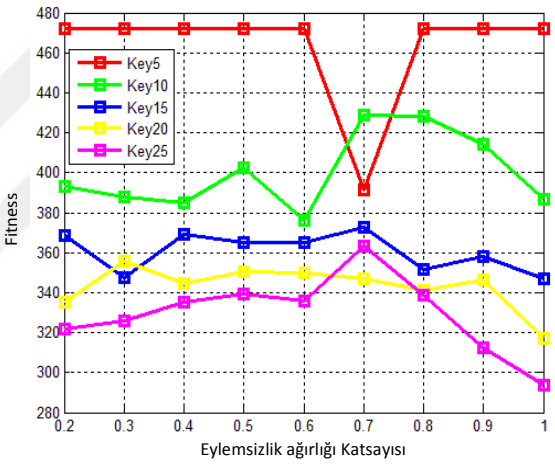

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

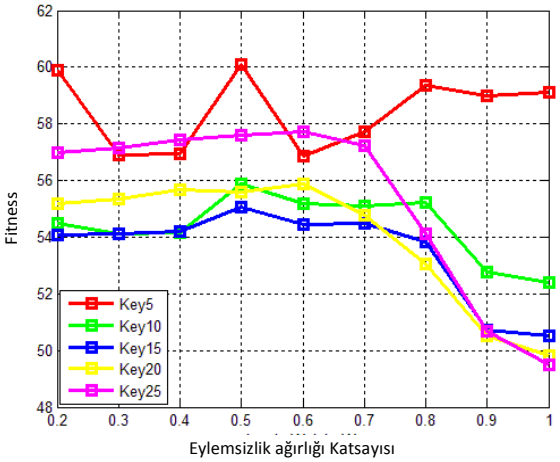



Şifrelenmiş metin uzunluğu = 750 karakter

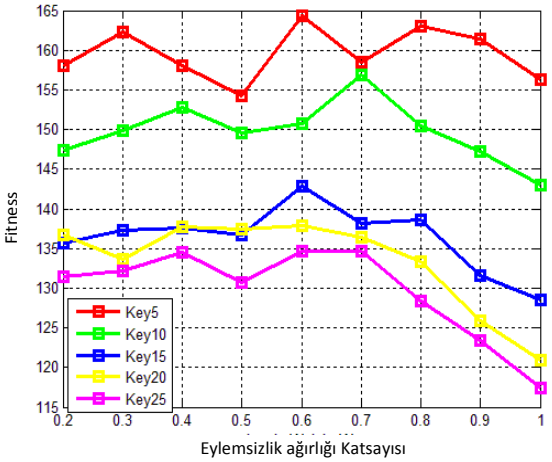



Şifrelenmiş metin uzunluğu = 1000 karakter

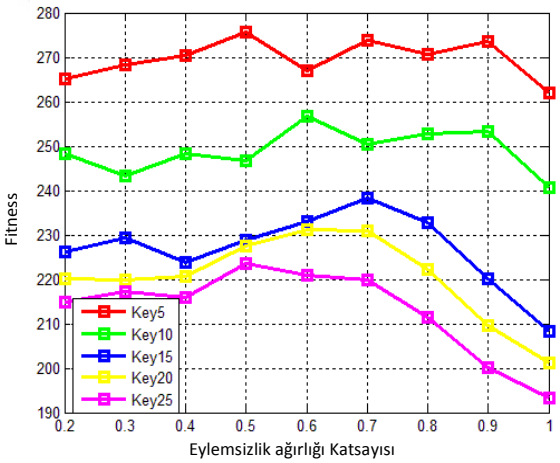
*Şekil 4.13.* PSO algoritmasının farklı w değerler kullanılarak ürettiği en iyi değerler.



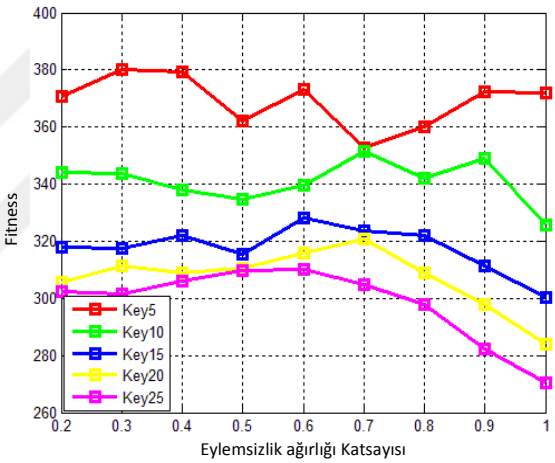

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter




Şifrelenmiş metin uzunluğu = 750 karakter



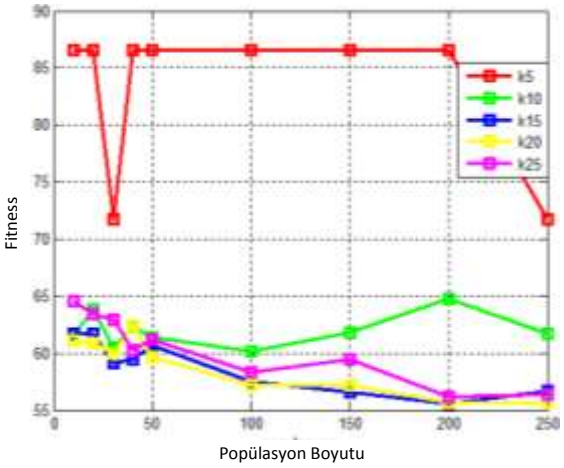

Şifrelenmiş metin uzunluğu = 1000 karakter

*Şekil 4.14.* PSO algoritmasının farklı w değerler kullanılarak ürettiği ortalama değerler

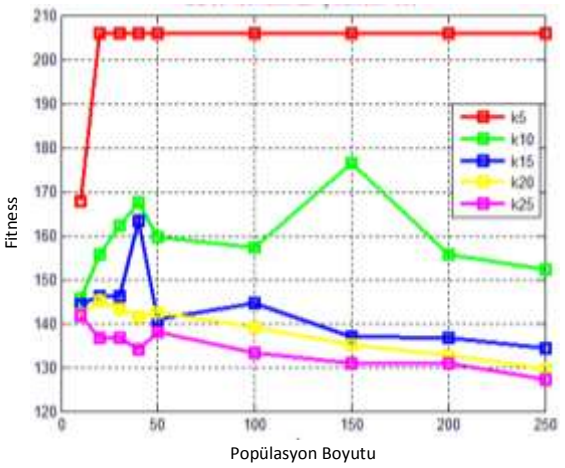
#### 4.2.1.3. DE'nin Kriptoanaliz Üzerinde Parametre Değerlerinin İncelenmesi

Daha önce belirtilen farklı uzunluktaki anahtar ve Vigenére ile şifrelenmiş metinler üzerinde 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri DE algoritmasının popülasyon boyutu (Tablo 5.1) olarak incelenmiştir. Dolayısıyla, sadece en iyi popülasyon boyutunu seçmek için 9× 4 × 5 = 180 farklı konfigürasyon denenmiştir. Çaprazlama oranı (CR) için 0.1, 0.25, 0.50 değerleri incelenmiştir. Çaprazlama için toplamda 3 × 4 × 5 = 60 farklı konfigürasyon test edilmiştir. Beş farklı ölçekleme faktörü (F) (0.1, 0.25, 0.5, 0.75, 1) için ise 5 × 4 × 5 = 100 farklı konfigürasyon test edilmiştir. Her bir konfigürasyon 30 kez koşulmuştur. Algoritmaların her bir duruma ait 30 kez koşulmasında Eşitlik (3.1) tarafından hesaplanan uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri farklı popülasyon boyutları ve farklı uzunluktaki şifrelenmiş metinler için Şekil 4.15 ve Şekil 4.16'da verilmiştir. Sonuçta, DE algoritmasının popülasyon boyutu 50 iken farklı uzunluktaki Vigenére şifreli metinler için iyi sonuçları ürettiği görülmüştür. Bunun dışındaki durumlarda, uygunluk değerlerinin düşük seviyede kaldığı görülmüştür.

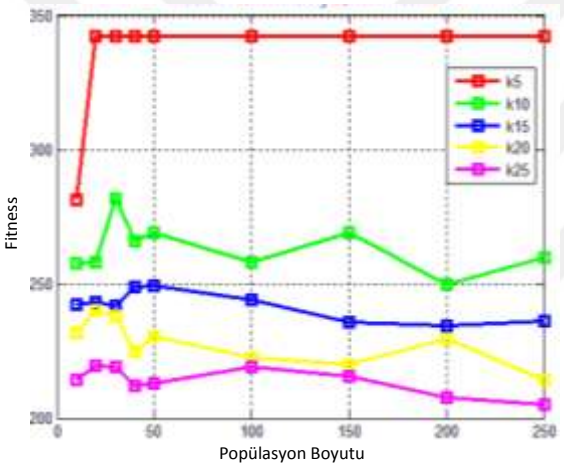
Şekil 4.17-4.20 ise, DE algoritmasının farklı çaprazlama ve ölçekleme faktörü oranları ile elde edilen en iyi ve ortalama değerleri göstermektedir. Burada, DE'nin çaprazlama oranı 0.2 olduğunda en yüksek uygunluk değerlerini ürettiği görülmüştür. Bunun dışındaki durumlarda, algoritmanın performansının ya değişmediği ya da düştüğü görülmüştür. Ölçekleme faktörü ise 1 olduğunda DE algoritmasının en yüksek uygunluk değerlerini ürettiği görülmüştür; onun dışındaki durumlarda, algoritma performansının ya değişmediği ya da düştüğü görülmüştür.



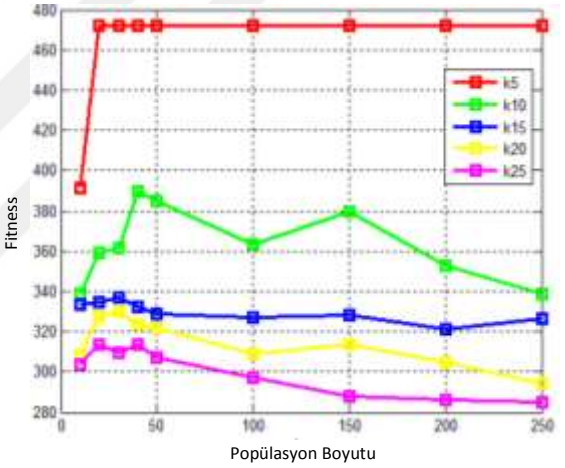

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

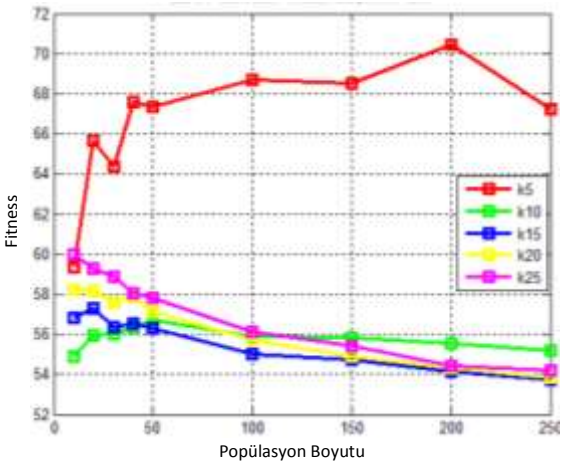



Şifrelenmiş metin uzunluğu = 750 karakter

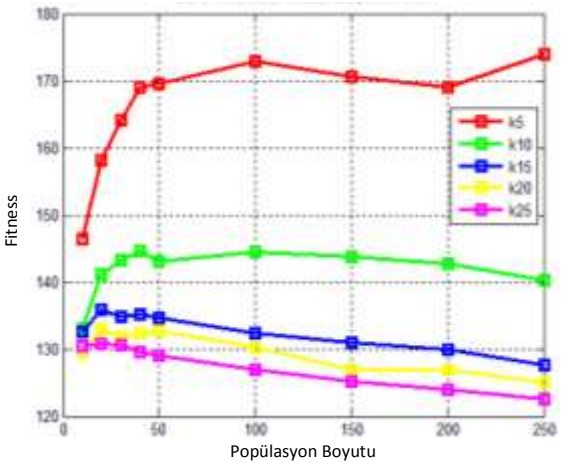



Şifrelenmiş metin uzunluğu = 1000 karakter

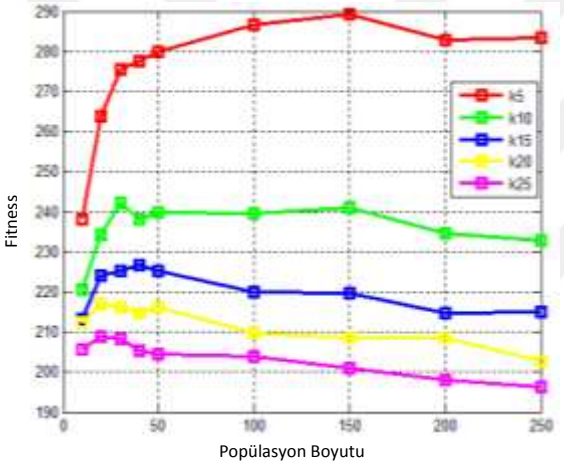
*Şekil 4.15.* DE algoritmasının farklı popülasyon boyutları kullanılarak ürettiği en iyi değerler



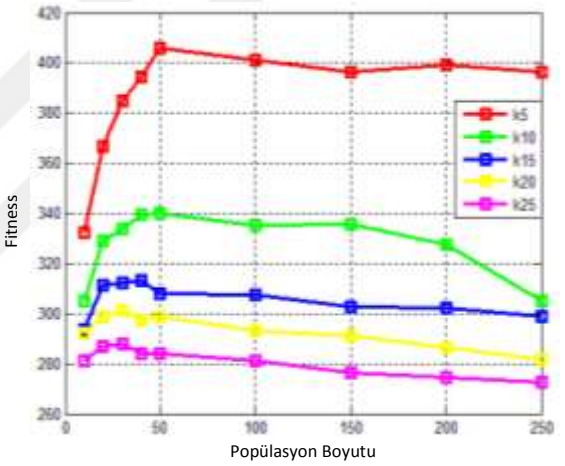

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

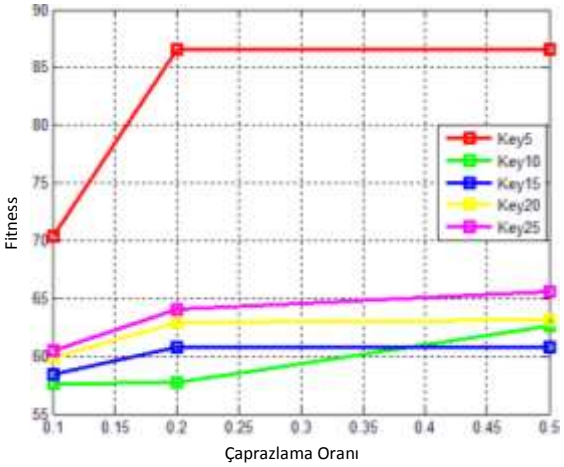



Şifrelenmiş metin uzunluğu = 750 karakter

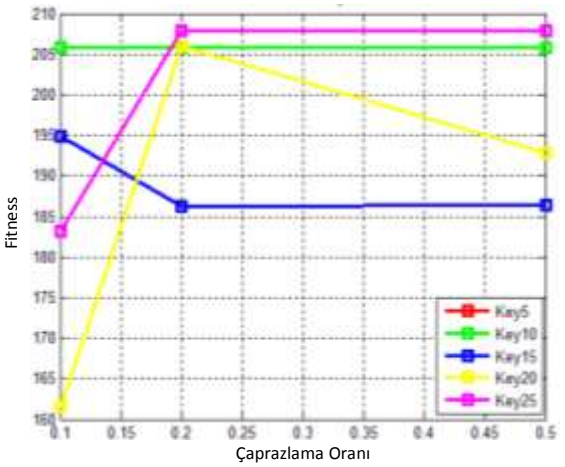



Şifrelenmiş metin uzunluğu = 1000 karakter

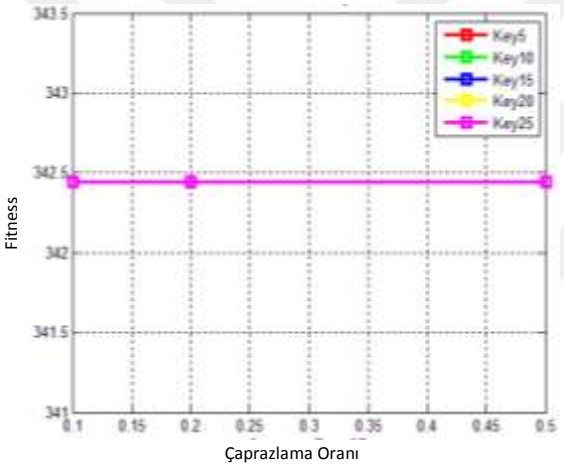
*Şekil 4.16.* DE algoritmasının farklı popülasyon boyutları kullanılarak ürettiği ortalama değerler



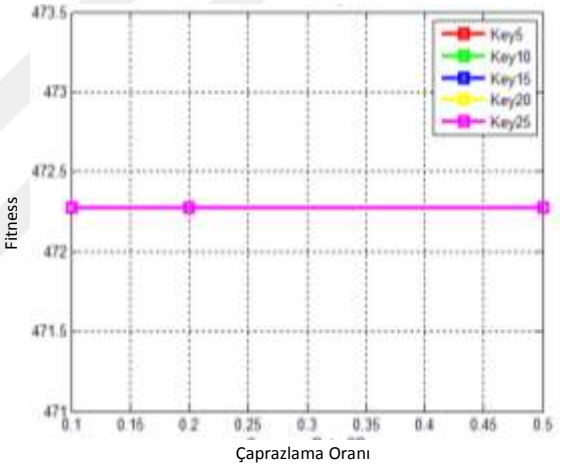

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

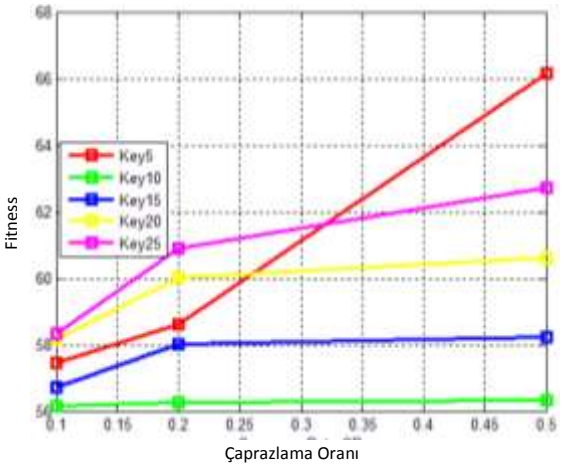



Şifrelenmiş metin uzunluğu = 750 karakter

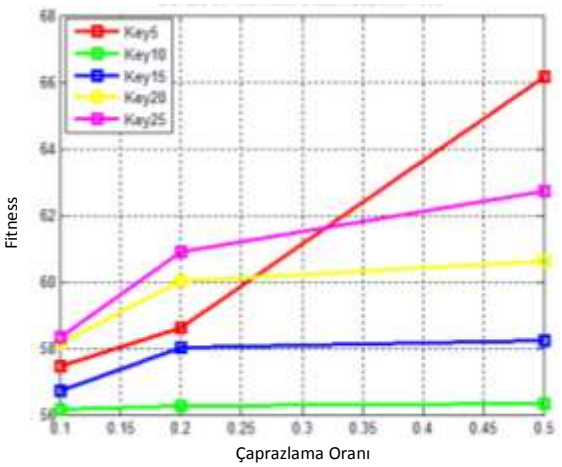



Şifrelenmiş metin uzunluğu = 1000 karakter

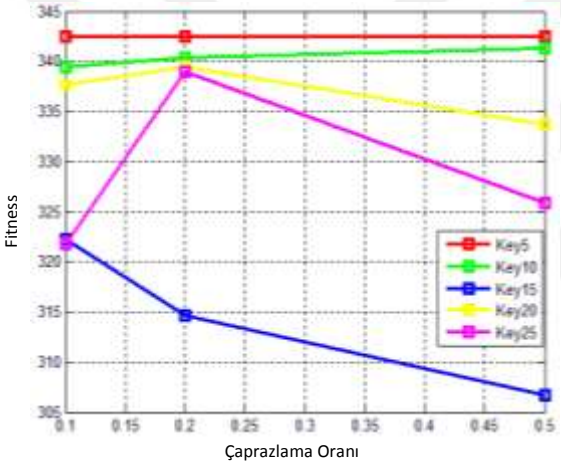
*Şekil 4.17.* DE algoritmasının farklı çaprazlama oranları kullanılarak ürettiği en iyi değerler.



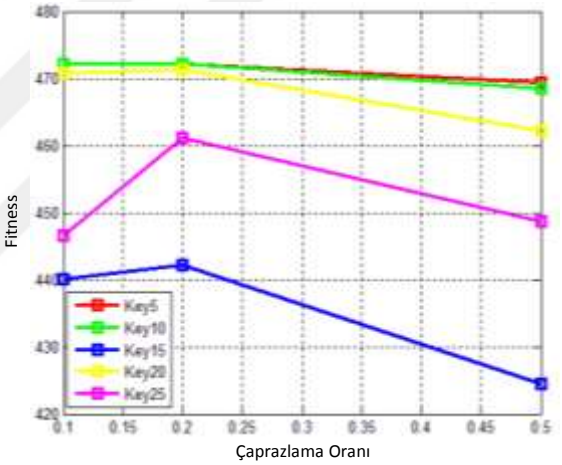

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

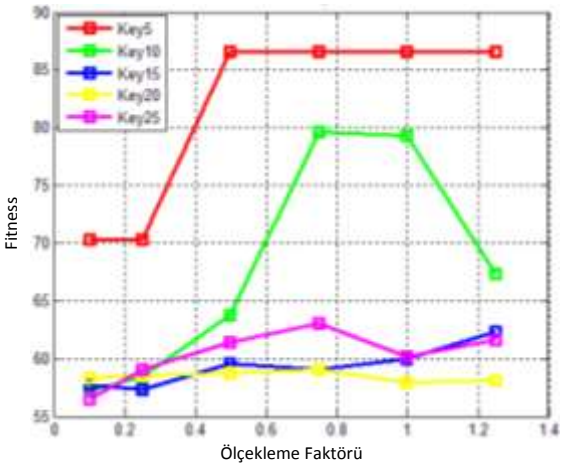



Şifrelenmiş metin uzunluğu = 750 karakter

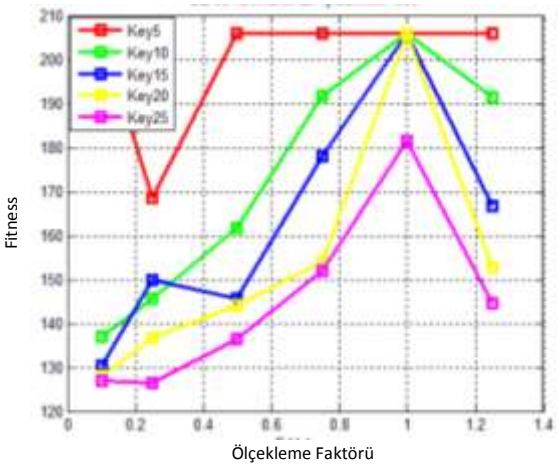



Şifrelenmiş metin uzunluğu = 1000 karakter

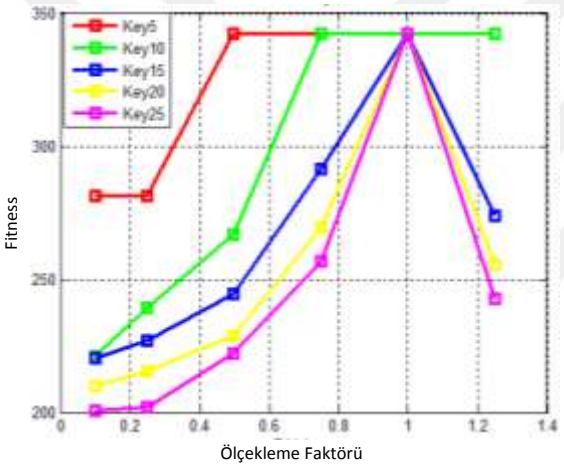
*Şekil 4.18.* DE algoritmasının farklı çaprazlama oranları kullanılarak ürettiği ortalama değerler.



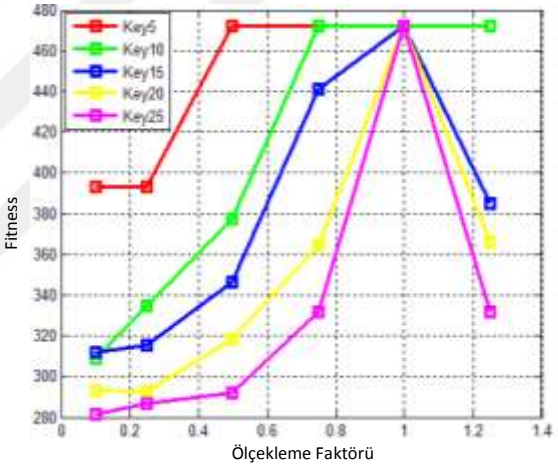

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

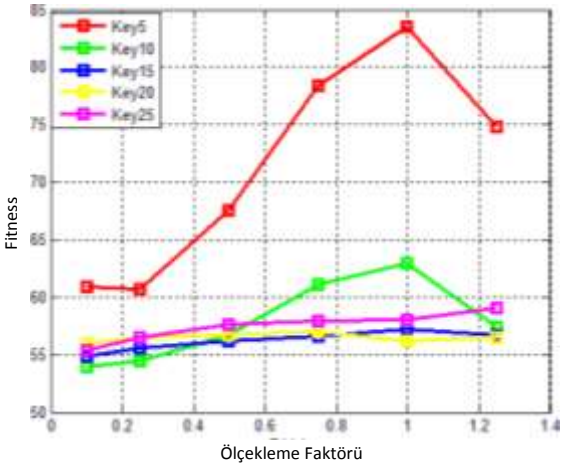



Şifrelenmiş metin uzunluğu = 750 karakter

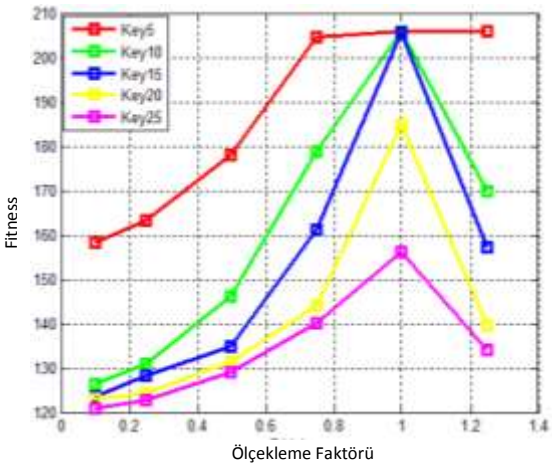



Şifrelenmiş metin uzunluğu = 1000 karakter

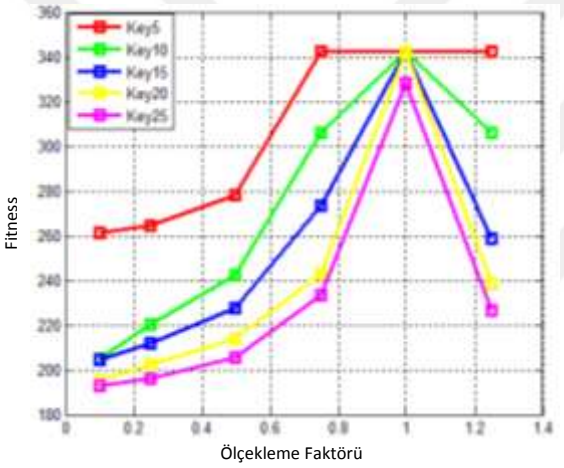
*Şekil 4.19.* DE algoritmasının farklı ölçekleme faktörü kullanılarak ürettiği en iyi değerler.



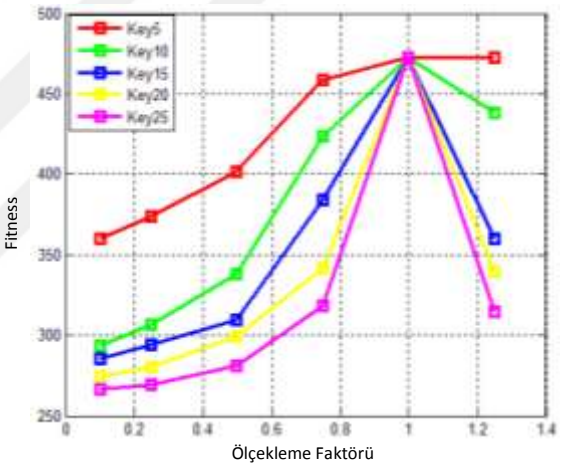

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter




Şifrelenmiş metin uzunluğu = 750 karakter




Şifrelenmiş metin uzunluğu = 1000 karakter

*Şekil 4.20.* DE algoritmasının farklı ölçekleme faktörü oranları kullanılarak ürettiği ortalama değerler.

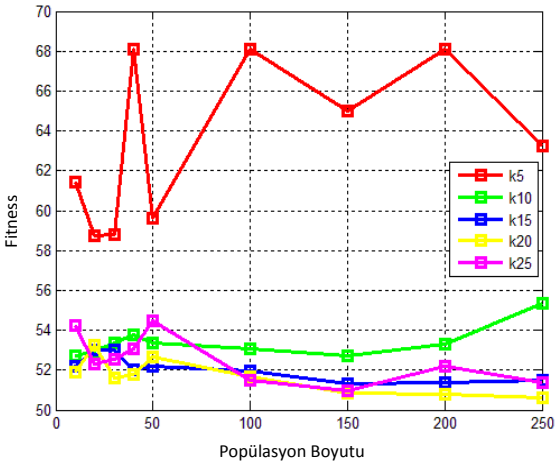
#### 4.2.1.4. ABC'nin Kriptoanaliz Üzerinde Parametre Değerlerinin İncelenmesi

Farklı uzunluktaki anahtarlar ve Vigenére ile şifrelenmiş metinler kullanarak deneylerde, ABC algoritması için, popülasyon boyutu olarak 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri ve limit çarpanı olarak 0.1, 0.25, 0.50, 0.75, 1 değerleri için çalışmalar tekrar edilmiştir. Popülasyon için $9 \times 4 \times 5 = 180$, limit için $5 \times 5 \times 4 = 100$ farklı konfigürasyon test edilmiştir. Her bir konfigürasyon için algoritma 30 kez koşulmuştur.

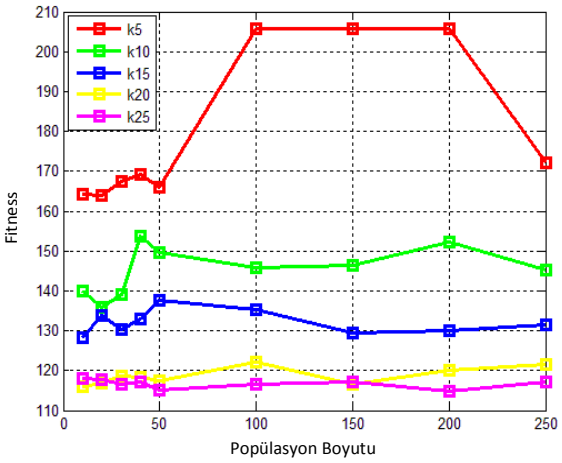
Algoritmanın 30 koşmasına ait Eşitlik (3.1) ile hesaplanan uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri Şekil 4.21 ve Şekil 4.22'de verilmiştir.

Sonuçta, ABC algoritmasının popülasyon boyutu aralığı 50-100 iken en iyi değerleri ürettiği görülmüştür. Bunun dışında, uygunluk değerlerinin değişiklik göstermediği veya düşük seviyede kaldığı görülmüştür.

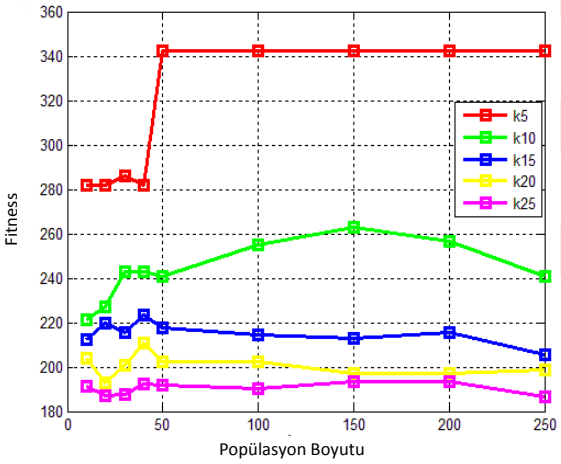
Farklı limit değerleri için ABC algoritmasının 30 koşmasına ait uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri Şekil 4.23 ve 4.24'te verilmiştir. ABC algoritmasının limit parametresinin aralığı 0.5-1 iken en iyi değerleri ürettiği görülmüştür. Bunun dışında, uygunluk değerlerinin değişiklik göstermediği veya düşük seviyede kaldığı görülmüştür.



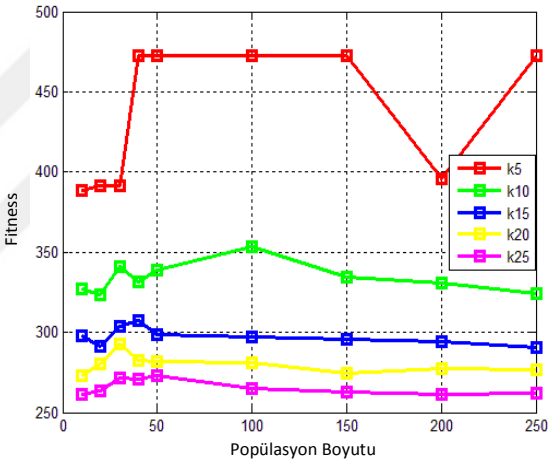

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

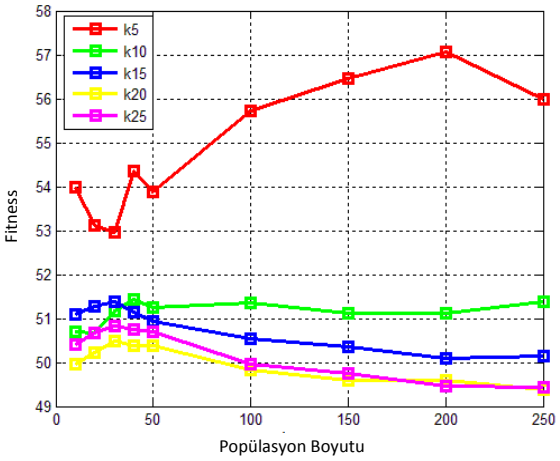



Şifrelenmiş metin uzunluğu = 750 karakter

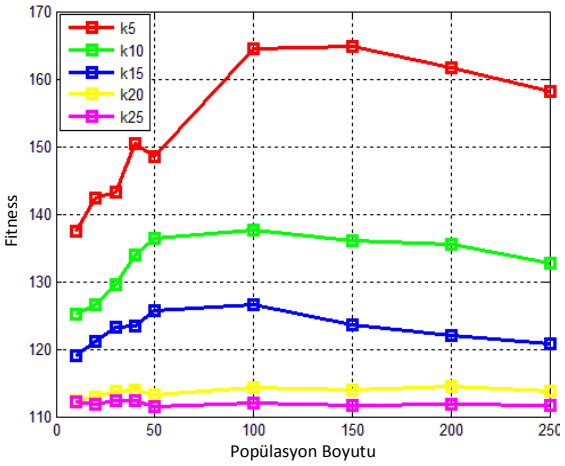



Şifrelenmiş metin uzunluğu = 1000 karakter

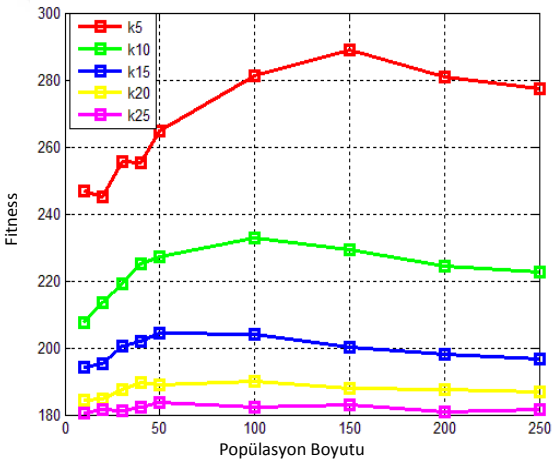
*Şekil 4.21.* ABC algoritmasının farklı popülasyon boyutları kullanılarak ürettiği en iyi değerler.



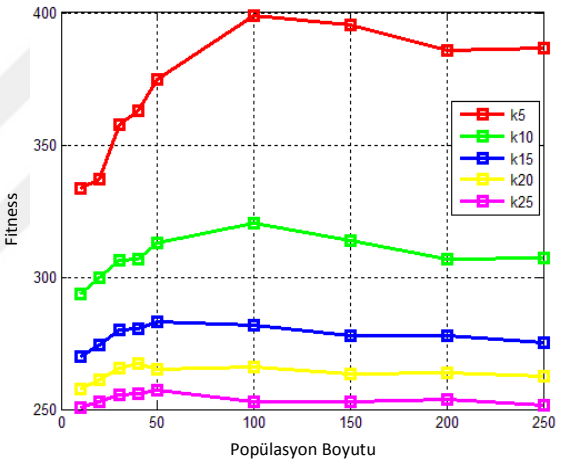

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

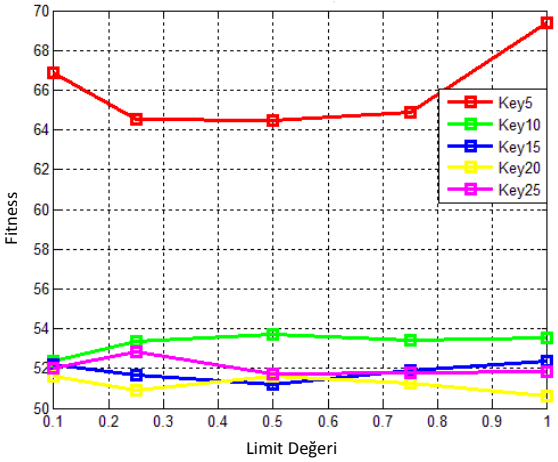



Şifrelenmiş metin uzunluğu = 750 karakter

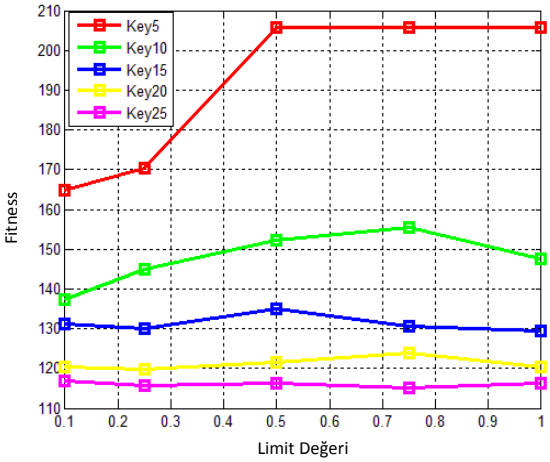



Şifrelenmiş metin uzunluğu = 1000 karakter

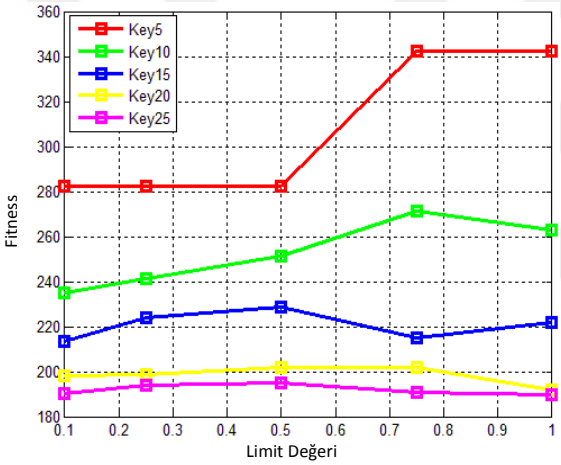
*Şekil 4.22.* ABC algoritmasının farklı popülasyon kullanılarak ürettiği ortalama değerler.



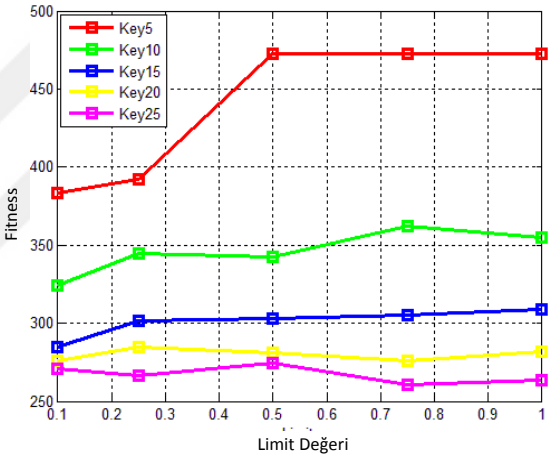

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

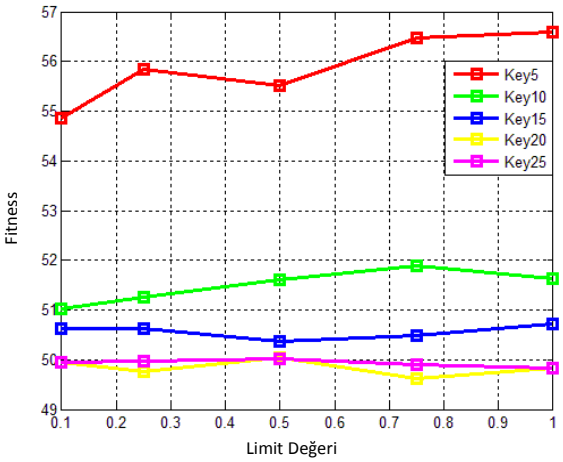



Şifrelenmiş metin uzunluğu = 750 karakter

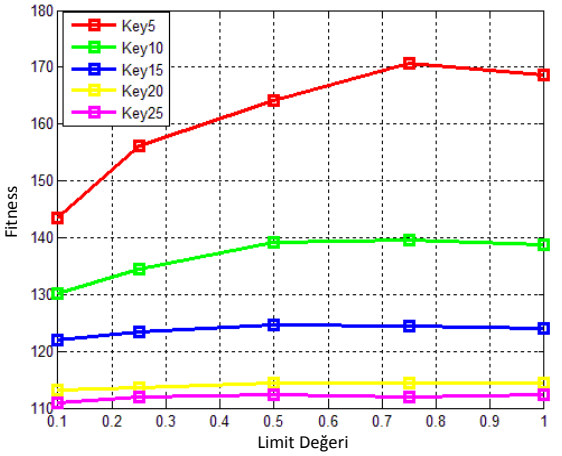



Şifrelenmiş metin uzunluğu = 1000 karakter

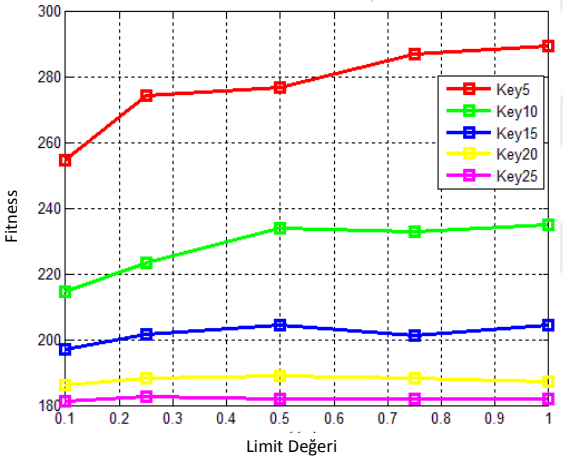
*Şekil 4.23.* ABC algoritmasının farklı limit değerleri kullanılarak ürettiği en iyi değerler.



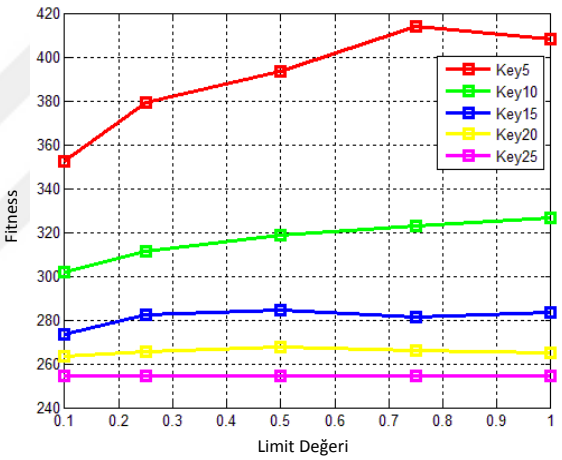

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter




Şifrelenmiş metin uzunluğu = 750 karakter




Şifrelenmiş metin uzunluğu = 1000 karakter

*Şekil 4.24.* ABC algoritmasının farklı limit değerleri kullanılarak ürettiği ortalama değerler.

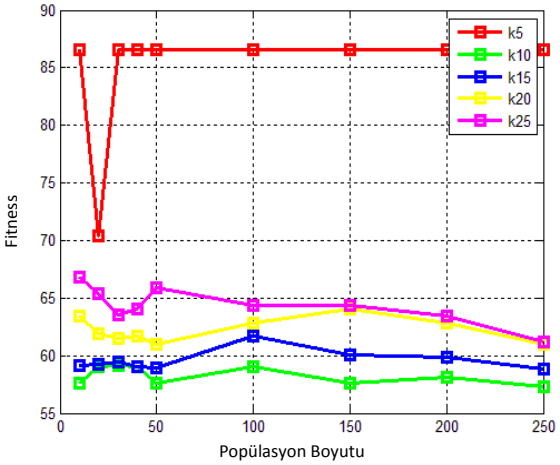
#### 4.2.1.5. BCABC'nin Kriptoanaliz Üzerinde Parametre Değerlerinin İncelenmesi

BCABC algoritmasında, popülasyon boyutu olarak 10, 20, 30, 40, 50, 100, 150, 200, ve 250 değerleri ve limit çarpanı olarak 0.1, 0.25, 0.50, 0.75, 1 değerleri için deneyler tekrar edilmiştir. Dolayısıyla, popülasyon için $9 \times 4 \times 5 = 180$, limit için $5 \times 4 \times 5 = 100$ farklı konfigürasyon test edilmiştir. Algoritma her durum için 30 kez çalıştırılmıştır. Çaprazlama oranı (CR) için 0.1, 0.25, 0.50 değerleri incelenmiştir. Çaprazlama için toplamda $3 \times 4 \times 5 = 60$ farklı konfigürasyon test edilmiştir.

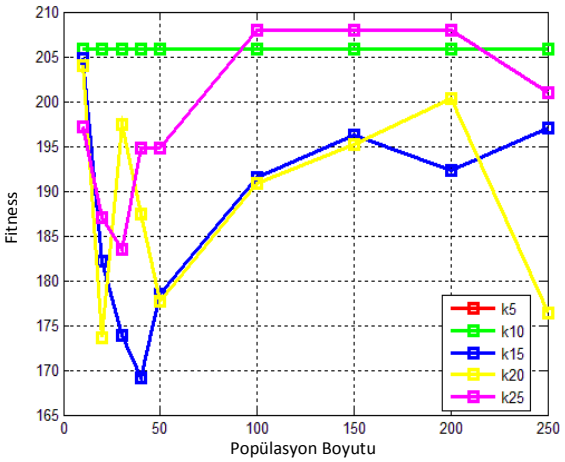
Algoritmanın farklı popülasyon boyutları için 30 koşmasında üretilen uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri Şekil 4.25 ve Şekil 4.26'da verilmiştir. BCABC algoritmasının popülasyon boyutu 150-200 aralığında iken en iyi değerleri ürettiği görülmüştür. Bunun dışında, uygunluk değerlerinin değişiklik göstermediği veya düşük seviyede kaldığı görülmüştür.

Farklı limit değerleri ile çalıştırılan 30'ar kez çalıştırılan BCABC algoritmasının uygunluk fonksiyonu değerlerinin ortalama ve en iyi değerleri Şekil 4.27-4.30'da verilmiştir. BCABC algoritmasının limit parametresi çarpanı [0.5-1] aralığında iken en uygun değerleri ürettiği görülmüştür. Bunun dışında, uygunluk değerlerinin değişiklik göstermediği veya düşük seviyede kaldığı görülmüştür.

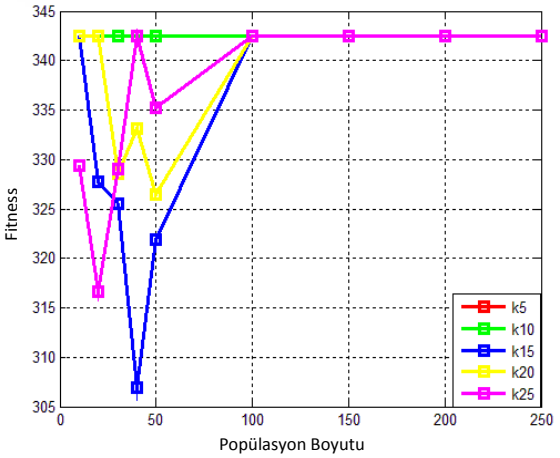
BCABC algoritmasının en yüksek uygunluk değerini çaprazlama oranının 0.2 olduğu durumda ürettiği görülmüştür. Bunun dışındaki durumlarda, algoritmanın performansının ya değişmediği ya da düştüğü görülmüştür.



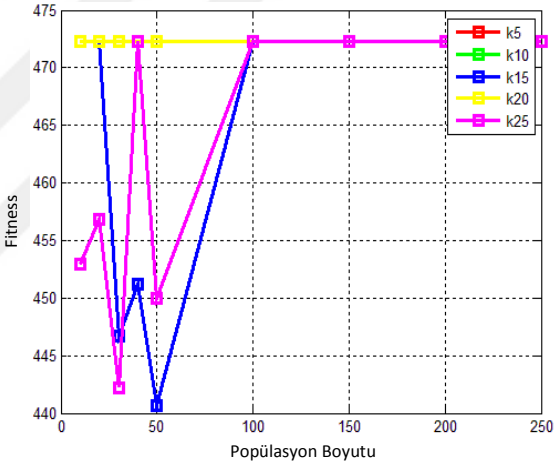

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

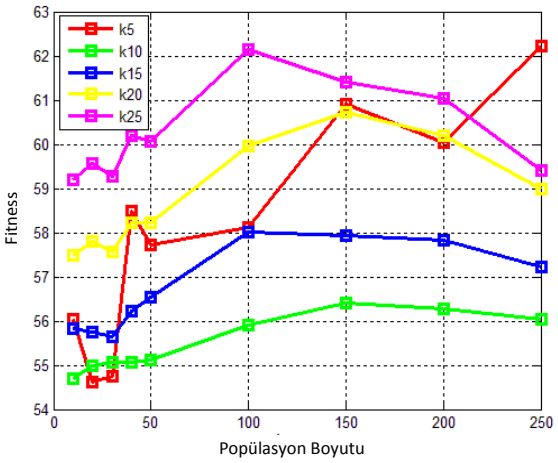



Şifrelenmiş metin uzunluğu = 750 karakter

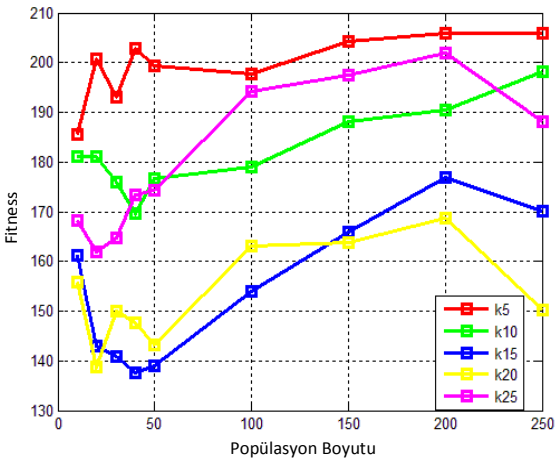



Şifrelenmiş metin uzunluğu = 1000 karakter

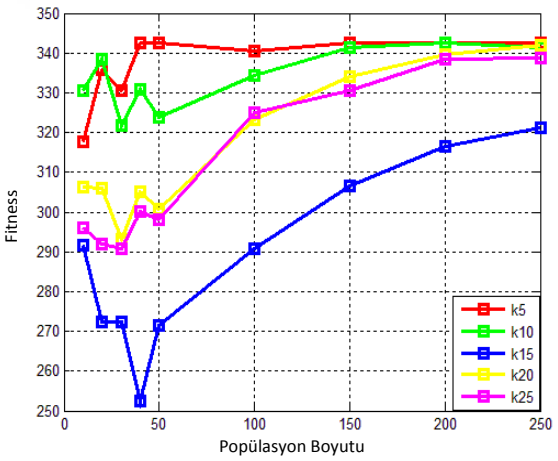
*Şekil 4.25.* BCABC algoritmasının farklı popülasyon boyutları kullanılarak ürettiği en iyi değerler.



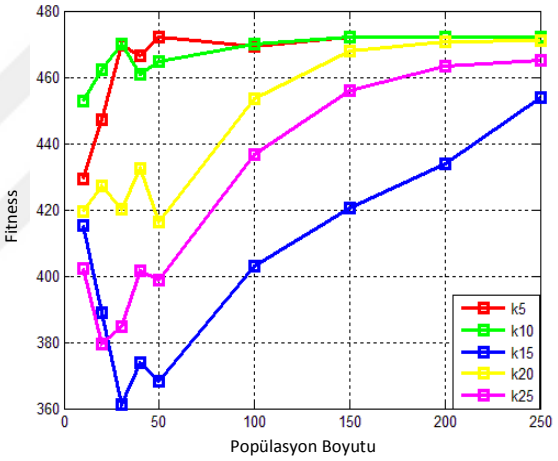

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

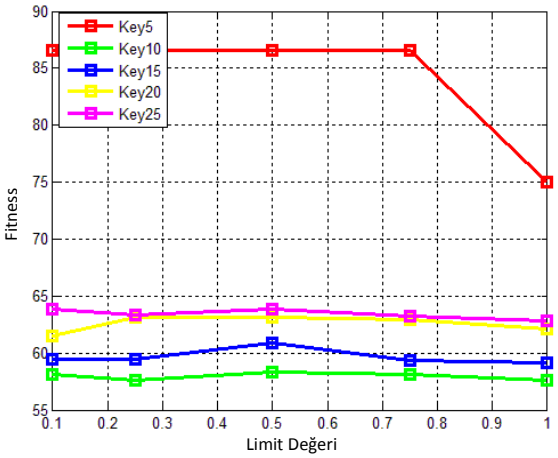



Şifrelenmiş metin uzunluğu = 750 karakter

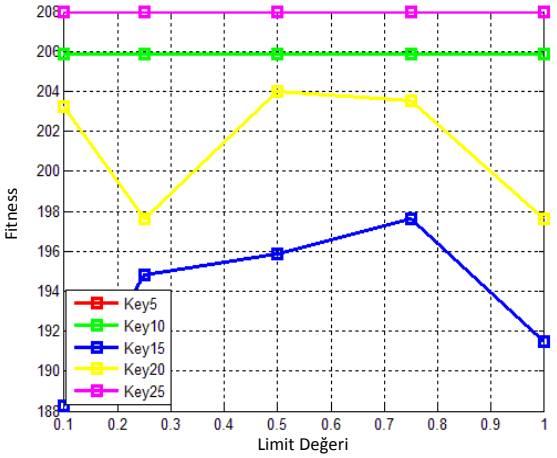



Şifrelenmiş metin uzunluğu = 1000 karakter

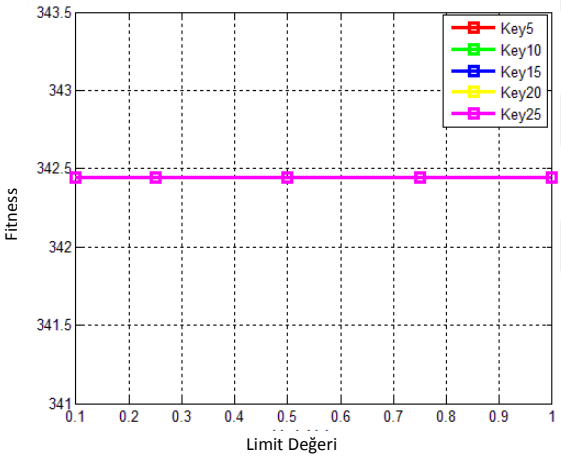
*Şekil 4.26.* BCABC algoritmasının farklı popülasyon boyutları kullanılarak ürettiği ortalama değerler.



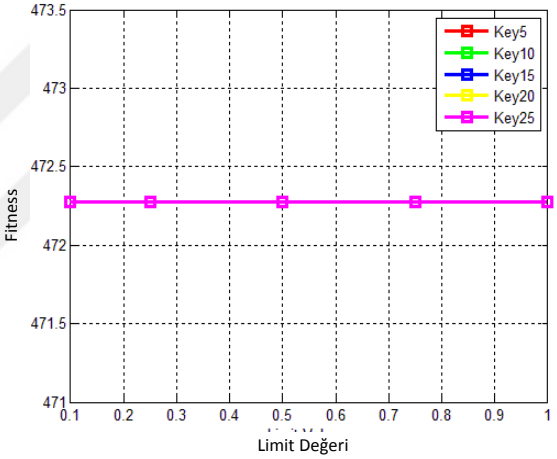

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

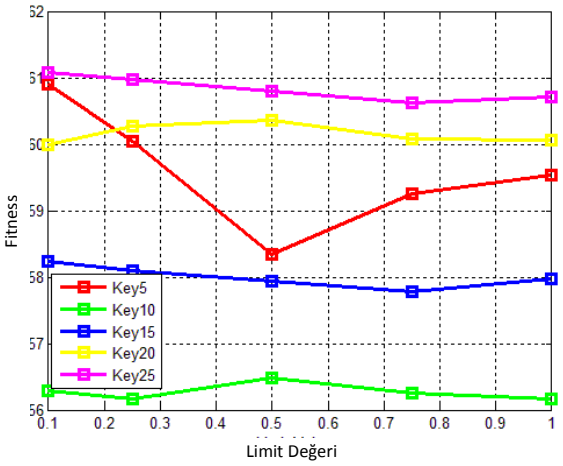



Şifrelenmiş metin uzunluğu = 750 karakter

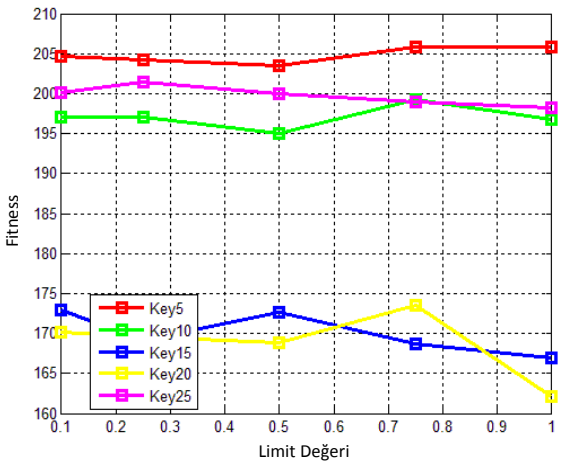



Şifrelenmiş metin uzunluğu = 1000 karakter

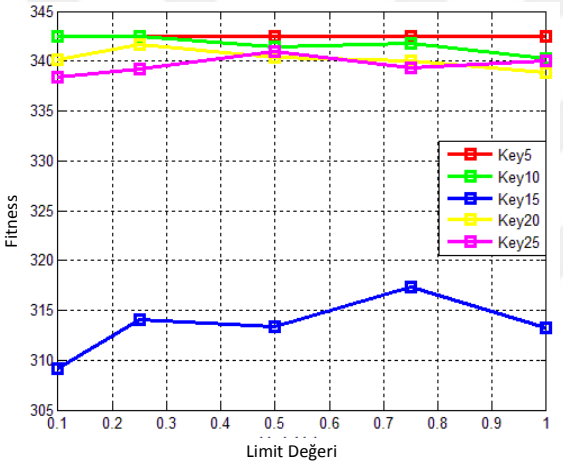
*Şekil 4.27.* BCABC algoritmasının farklı limit çarpanı parametreleri kullanılarak ürettiği en iyi değerler.



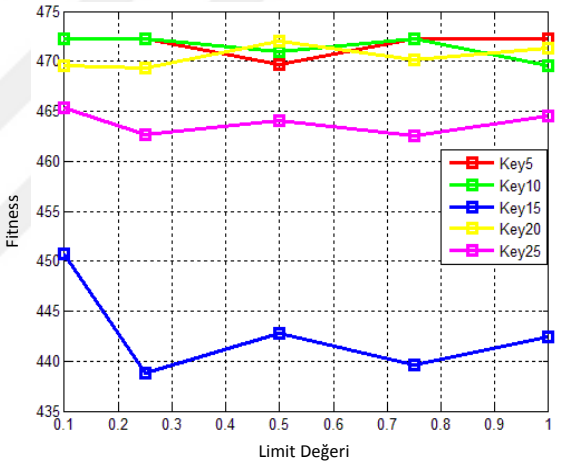

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

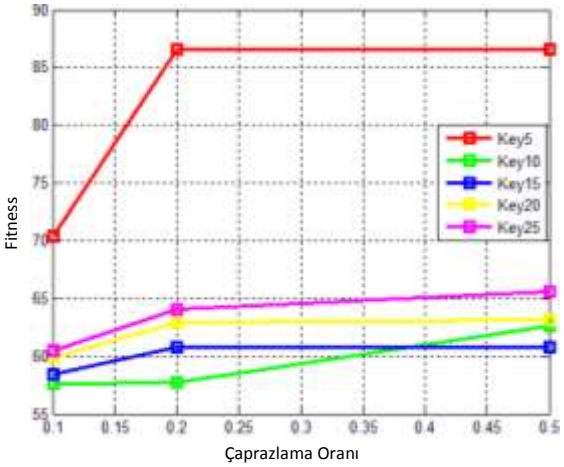



Şifrelenmiş metin uzunluğu = 750 karakter

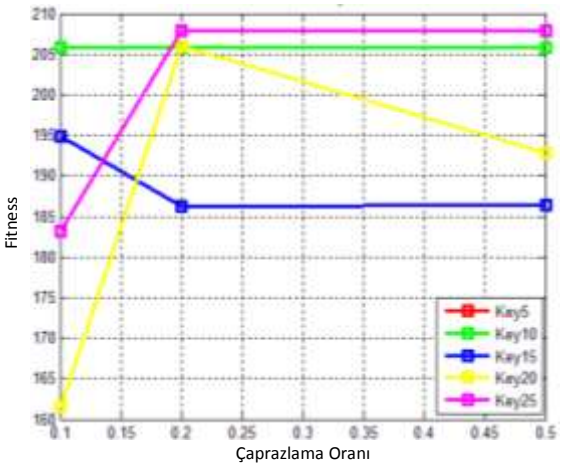



Şifrelenmiş metin uzunluğu = 1000 karakter

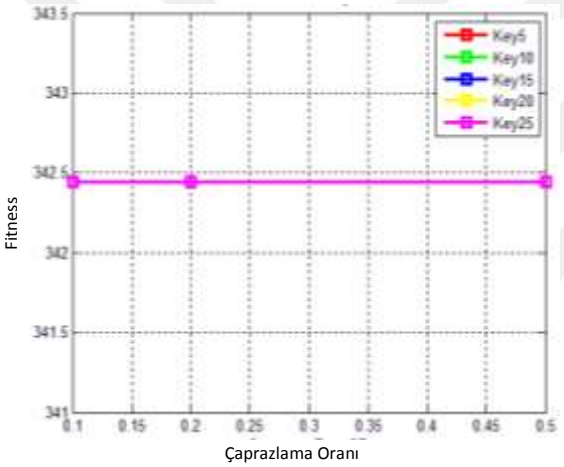
*Şekil 4.28.* BCABC algoritmasının farklı limit çarpanı parametreleri kullanılarak ürettiği ortalama değerler.



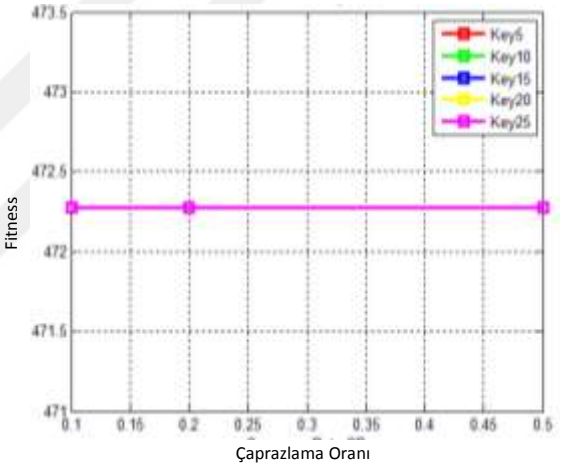

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter

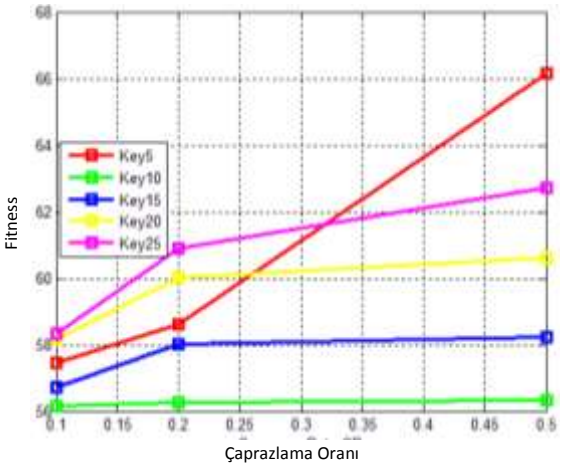



Şifrelenmiş metin uzunluğu = 750 karakter

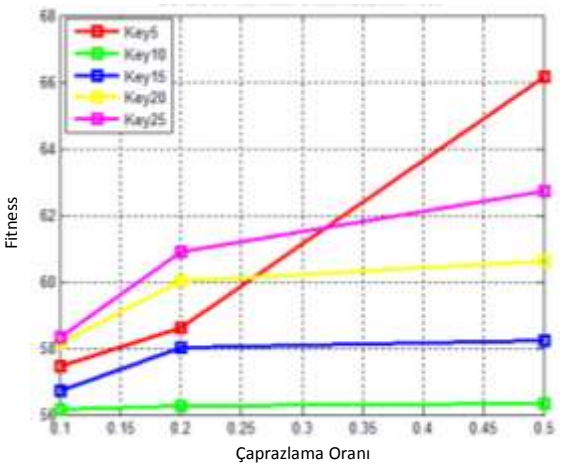



Şifrelenmiş metin uzunluğu = 1000 karakter

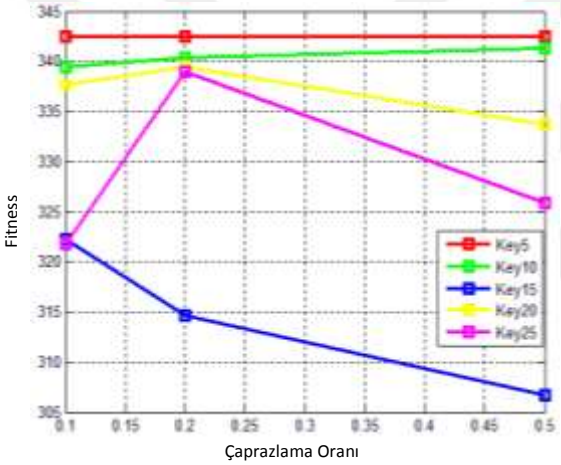
*Şekil 4.29.* BCABC algoritmasının farklı çaprazlama oranı kullanılarak ürettiği en iyi değerler.



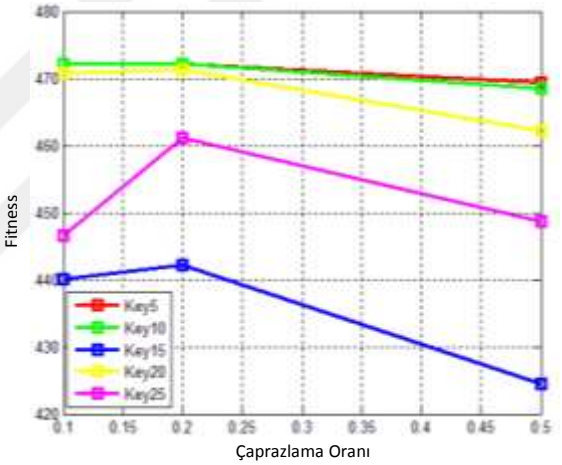

Şifrelenmiş metin uzunluğu =250 karakter




Şifrelenmiş metin uzunluğu = 500 karakter




Şifrelenmiş metin uzunluğu = 750 karakter




Şifrelenmiş metin uzunluğu = 1000 karakter

*Şekil 4.30.* BCABC algoritmasının farklı çaprazlama oranı kullanılarak ürettiği ortalama değerler.

Bütün deneyler sonucunda her bir algoritma için önerilen kontrol parametreleri değerleri Tablo 4.2'de yer almaktadır.

Tablo 4.2. Her bir algoritmanın kontrol parametreleri için önerilen değerler

| Parametreler | GA | PSO | DE | ABC | BCABC |
|---|---|---|---|---|---|
| Popülasyon boyutu (PS): | 250 | 250 | 50 | 100 | 200 |
| Çaprazlama oranı (CR): | 0.75 | | 0.2 | | 0.2 |
| Mutasyon oranı (MP): | 0.75 | | | | |
| Sosyal bileşen (c1): | | 2 | | | |
| Kavramsal bileşen (c2): | | 1.75 | | | |
| Eylemsizlik ağırlığı (w): | | 0.7 | | | |
| Limit (L): | | | | 0.75 | 0.1 |
| Ölçekleme faktörü (F): | | | 1 | | |
| Max generasyon: | 120 | 120 | 600 | 300 | 75 |

### 4.2.2. Deneysel Çalışma 2: Optimizasyon Tabanlı Kriptoanaliz

Bu çalışmada, Vigenére şifrelemenin kriptoanalizini yürütmek amacıyla farklı uzunlukta anahtarlarla Türkçe ve İngilizce metinler üzerinde beş farklı algoritma (GA, PSO, DE, ABC, BCABC) kullanılmıştır. Algoritmaların kontrol parametre değerleri olarak bir önceki çalışmada elde edilen uygun değerler kullanılmıştır. Yine her bir durum 30 kez koşulmuştur. Bu bölümde algoritmaların bu kontrol parametre değerleri ile koşulmasıyla her bir algoritma detaylı bir şekilde analiz edilecektir. Bütün algoritmalarda 5, 10, 15, 20 ve 25 uzunluğundaki farklı anahtarlar ve Türkçe ve İngilizce metinlerden oluşmuş olan 250, 500, 750 ve 1000 gibi farklı uzunlukta Vigenére ile şifrelenmiş metinler kullanılmıştır.

#### 4.2.2.1. GA Sonuçları

Deneyler sonucunda GA ile elde edilen anahtardaki doğru karakter sayısı (ADKS) ve uygunluk değerleri şifreli İngilizce metinler için Tablo 4.3'te ve Türkçe metinler için Tablo 4.4'te gösterilmiştir.

Şifreli İngilizce metinlerin uzunluğu 250'şer artacak şekilde 250'ten 1000'e kadar artacak şekilde ayarlanmıştır. Buradaki 250 karakterli metinde en düşük ADKS her bir anahtar için 0 (sıfır), en yüksek ADKS 15 karakterli anahtarda 11, ortalama ADKS ise 3.2333 değeriyle 5 karakterli anahtarda elde edilmiştir. Bunların en yüksek standart sapması ise 15 karakterli şifrede 2.4059, en düşük standart sapma ise 20 karakterli şifrede 0.3333 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 53.5698 ile 5 karakterli şifrede, en yüksek uygunluk değeri ise yine 5 karakterli şifrede elde edilmiştir. Bu da doğal olarak, 5 karakterli şifreye ait ve 11.7235 değerindeki en yüksek uygunluk standart sapmasını ortaya çıkarmıştır. Ortalama uygunluk değeri ise yine 5 karakterli şifrede 69.0764 olarak belirlenmiştir. Burada, 5 karakterli şifrenin yer yer başarılı sonuçlar vermesine rağmen oldukça dağınık bir görüntü çizdiği görülmüştür. Buna karşın, 25 karakterli şifrenin daha düzgün ve kararlı sonuçlar ortaya çıkardığı söylenebilir.

Uzunluğu 500 karaktere sahip şifreli metinde en düşük ADKS 5 karakterli anahtarda 4, en yüksek ADKS 15 karakterli anahtarda 25, ortalama ADKS ise 23 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.8929, en düşük standart sapma ise 5 karakterli şifrede 0.1826 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 156.3814 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise 5, 10, 15 ve 20 karakterli şifrelerde 205.8708 olarak elde edilmiştir. Burada ise, 15 karakterli şifreye ait ve 11.7197 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. Ortalama uygunluk değeri ise yine 5 karakterli şifreye ait olan 204.5317 olarak belirlenmiştir. Bu değerler neticesinde, uygunluk değerinin standart sapması da 9.7627 değerinde kaldığından, 25 karakterli şifrenin daha başarılı olduğu söylenebilir.

Benzer sonuçlar 750 karakterli metin için de görülmektedir. Burada en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 25, ortalama ADKS ise 20.4 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.4439, en düşük standart sapma ise 5 karakterli şifrede 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 269.3495 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise her bir şifrede 342.4439 olarak elde edilmiştir. Burada ise, yine 15

karakterli şifreye ait ve 23.3129 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. Ortalama uygunluk değeri ise en düşük olarak yine 25 karakterli şifreye ait olan 309.6918 olarak belirlenmiştir. Bu değerler neticesinde, uygunluk değerinin standart sapması 0 olmasına rağmen 5 karakterli şifrenin iyi sonuç vermediği ve 25 karakterli şifrenin diğerlerine göre daha başarılı olduğu söylenebilir.

Son olarak, 1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda 4, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 21.2 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.0578, en düşük standart sapma ise 5 karakterli şifrede 0.1826 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 387.3119 ile 15 karakterli şifrede, en yüksek uygunluk değeri ise her bir şifrede 472.2756 olarak elde edilmiştir. Burada ise, yine 15 karakterli şifreye ait ve 28.4480 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. En düşük ortalama uygunluk değeri ise 25 karakterli şifreye ait olan 429.0244 olarak belirlenmiştir. Bu değerler neticesinde, beklendiği gibi 25 karakterli anahtarın diğerlerine göre daha üstün performans gösterdiği görülmüştür.

Öte yandan, şifreli Türkçe metinlerin uzunluğu yine 250'şer artacak şekilde 250 ile 1000 karakter arasında değişmektedir. Buradaki 250 karakterli metinde en düşük ADKS 10 ve 20 karakterli anahtarlar için 0 (sıfır), en yüksek ADKS 15 ve 25 karakterli anahtarlarda 15, en yüksek ortalama ADKS ise 8.0667 değeriyle 25 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 25 karakterli şifrede 3.7040, en düşük standart sapma ise 5 karakterli şifrede 0.7184 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 58.6610 ile 10 karakterli şifrede, en yüksek uygunluk değeri ise 25 karakterli şifrede 89.5371 olarak elde edilmiştir. Ayrıca, en düşük uygunluk sapması ise 15 karakterli şifreye ait ve 5.28 değerinde ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 20 karakterli şifreye ait ve 73.3611 olarak belirlenmiştir. Burada, 20 ve 25 karakterli anahtarların diğerlerinden daha öne çıktığı görülmektedir.

Uzunluğu 500 karakterde olan şifreli metinde en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 24, en yüksek ortalama ADKS ise 18.9667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart

sapma yine 25 karakterli şifrede 2.8945, en düşük standart sapma ise 5 karakterli şifrede 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 171.8244 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise 5, 10 ve 15 karakterli şifrelerde 208.7812 olarak elde edilmiştir. Burada ise, 15 karakterli şifreye ait ve 9.4435 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. En düşük standart sapma ise 5 karakterli anahtara ait olan 0 olarak hesaplanmıştır. En yüksek ortalama uygunluk değeri ise yine 5 karakterli anahtarda 208.7812 ve en düşük ortalama uygunluk değeri ise 190.3536 olarak 25 karakterli anahtarda ortaya çıkmıştır. Bu değerler neticesinde, uygunluk değerinin standart sapması da 9.2062 değerinde olmasına rağmen, 25 karakterli anahtarın daha başarılı sonuçlar çıkardığı söylenebilir.

Bu durum 750 karakterli metinde de benzerlik göstermektedir. Burada en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 23, en yüksek ortalama ADKS ise 19.1333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.5695, en düşük standart sapma ise 5 karakterli şifrede 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 246.1968 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise 5, 10, 15 ve 20 karakterli anahtarlarda 332.4877 olarak elde edilmiştir. Burada ise, 25 karakterli anahtara ait ve 16.4095 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. Ortalama uygunluk değeri ise en düşük olarak yine 25 karakterli şifreye ait olan 298.0012 olarak belirlenmiştir. Bu değerler neticesinde, uygunluk değerinin standart sapması 0 olmasına rağmen 5 karakterli şifrenin iyi sonuç vermediği ve 25 karakterli şifrenin diğerlerine göre daha başarılı olduğu söylenebilir.

Ayrıca, 1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda yine 4, en yüksek ADKS 25 karakterli anahtarda yine 25, en yüksek ortalama ADKS ise 21.0667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.0998, en düşük standart sapma ise 5 karakterli şifrede 0.1826 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 368.2908 ile 15 karakterli şifrede, en yüksek uygunluk değeri ise her bir şifrede eşit olarak 459.2877 olarak elde edilmiştir. Burada ise, yine 15 karakterli şifreye ait ve 23.5861 değerindeki en yüksek uygunluk

standart sapma elde edilmiştir. En düşük ortalama uygunluk değeri ise 25 karakterli şifreye ait olan 421.7398 olarak belirlenmiştir. Bu değerler neticesinde, beklendiği gibi 25 karakterli anahtarın diğerlerine göre daha üstün performans gösterdiği görülmüştür.

Daha genel bir ifadeyle, anahtar boyutu arttıkça daha kısa şifreli metinleri çözmek için gerekli iterasyon sayısı artmaktadır. Şifreli metnin boyutu arttıkça anahtarın daha doğru sonuçlar verdiği görülmektedir. Ayrıca, doğru olarak elde edilen karakter sayısı da artmaktadır. Söz konusu karakter sayısındaki artış Türkçe metinlerinde daha net görülmektedir.

#### 4.2.2.2. PSO Algoritmasının Sonuçları

Deneyler sonucunda PSO ile elde edilen ADKS ve uygunluk değerleri şifreli İngilizce metinler için Tablo 4.5'de ve Türkçe metinler için Tablo 4.6'da gösterilmiştir.

Öncelikle, şifreli İngilizce metinlerin uzunluğu yine 250'şer artacak şekilde 250 ile 1000 karakter arasında değiştiğini belirttikten sonra, buradaki 250 karakterli metinde en düşük ADKS her bir anahtar için 0 (sıfır), en yüksek ADKS ise 5, 20 ve 25 karakterli anahtarlarda 5, en yüksek ortalama ADKS ise 1.3667 değeriyle 5 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise yine 5 karakterli anahtarda 1.5862, en düşük standart sapma ise 15 karakterli anahtarda 0.7739 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 50.4001 ile 10 karakterli şifrede, en yüksek uygunluk değeri ise 25 karakterli şifrede 52.4628 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması ise 15 karakterli şifreye ait ve 1.4838 değerinde hesaplanmıştır. En düşük ortalama uygunluk değeri ise yine 15 karakterli anahtara ait ve 54.2724 olarak belirlenmiştir. Burada, özellikle uygunluk değerlerine ait veriler karşılaştırıldığında 15 ve 25 karakterli anahtarların diğerlerine göre biraz daha öne çıktığı görülmüştür.

Uzunluğu 500 karakterde olan şifreli metinde en düşük ADKS 5 ve 20 karakterli anahtarlarda 1, en yüksek ADKS 25 karakterli anahtarda 13, en yüksek ortalama ADKS ise 7.6333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 25 karakterli anahtarda 3.0113, en düşük standart sapma ise 5 karakterli şifrede 1.0875 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 119.5057 ile 20 karakterli

anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 205.8708 olarak elde edilmiştir. Bu kez, 5 karakterli şifreye ait ve oldukça yüksek olan 24.6460 değerindeki en yüksek uygunluk standart sapmasına sahip olduğu fark edilmektedir. En düşük standart sapma ise 25 karakterli anahtara ait olan 8.6591 olarak hesaplanmıştır. En yüksek ortalama uygunluk değeri ise yine 5 karakterli anahtarda 156.8410 iken, en düşük ortalama uygunluk değeri ise 135.2078 olarak 20 karakterli anahtarda belirmiştir. Bu verilere göre, uygunluk değerinin standart sapması da 9.2062 değerinde olmasına rağmen, 25 karakterli anahtarın daha başarılı sonuçlar çıkardığı söylenebilir.

Benzer ifadeleri 750 karakterli metin için de ortaya koymak mümkündür. Burada en düşük ADKS 5 ve 25 karakterli anahtarlarda 2, en yüksek ADKS 15 ve 25 karakterli anahtarlarda 12, en yüksek ortalama ADKS ise 8.4333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli anahtarda 2.4023, en düşük standart sapma ise 5 karakterli şifrede 0.8172 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 192.3029 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise 10 karakterli anahtarda 220.2844 olarak elde edilmiştir. Burada ise, 5 karakterli anahtara ait ve oldukça yüksek olan 33.7318 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. Ortalama uygunluk değeri ise en düşük olarak yine 25 karakterli şifreye ait olan 216.987 olarak belirlenmiştir. Bu değerler neticesinde, uygunluk değeri standart sapması 11.685 elan ve en düşük olan 25 karakterli şifrenin diğerlerine göre daha başarılı olduğu söylenebilir.

Son olarak, 1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda bu kez 0, en yüksek ADKS 25 karakterli anahtarda 18, en yüksek ortalama ADKS ise 9.9333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 25 karakterli şifrede 2.6901, en düşük standart sapma ise 5 karakterli şifrede 0.9732 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 273.1470 ile 5 karakterli anahtarda, en yüksek uygunluk değeri ise yine 5 karakterli anahtarda 472.2756 olarak elde edilmiştir. Burada ise, yine 5 karakterli şifreye ait ve 48.8020 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. En düşük ortalama uygunluk değeri ise 25 karakterli şifreye ait olan 310.6931 olarak belirlenmiştir. Bu değerler

neticesinde, beklendiği gibi 25 karakterli anahtarın diğerlerine göre daha üstün performans gösterdiğini söylemek mümkündür.

Öte yandan, uzunluğu 250 ile 1000 karakter arasında değişen şifreli Türkçe metinlere bakılacak olursa, buradaki 250 karakterli metinde en düşük ADKS her bir uzunluktaki anahtarlar için 0 (sıfır), en yüksek ADKS 25 karakterli anahtara 9, en yüksek ortalama ADKS ise 3.2333 değeriyle 25 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 20 karakterli şifrede 2.3004, en düşük standart sapma ise 5 karakterli şifrede 0.9223 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 54.9033 ile 5 karakterli anahtarda, en yüksek uygunluk değeri ise yine 5 karakterli şifrede 87.6893 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması ise 25 karakterli şifreye ait ve 3.3676 değerinde ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 20 karakterli şifreye ait ve 62.4331 olarak belirlenmiştir. Burada, 15, 20 ve 25 karakterli anahtarların diğerlerine nazaran biraz daha öne çıktığı söylenebilir.

Uzunluğu 500 karakter olan şifreli metinde en düşük ADKS 5 karakterli anahtarda 2, en yüksek ADKS 25 karakterli anahtarda 12, en yüksek ortalama ADKS ise 7.2 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 25 karakterli şifrede 2.1560, en düşük standart sapma ise 5 karakterli anahtarda 0.6261 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 128.7282 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 208.7812 olarak elde edilmiştir. Burada ise, 5 karakterli şifreye ait ve 15.048 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. En düşük standart sapma ise 20 karakterli anahtarda 5.6208 ortaya çıkmıştır. En yüksek ortalama uygunluk değeri ise yine 5 karakterli anahtarda 168.7199 ve en düşük ortalama uygunluk değeri ise 153.1097 olarak 20 karakterli anahtarda görülmüştür. Bu değerler neticesinde, genel performansıyla 25 karakterli anahtarın diğerlerinden daha başarılı sonuçlar çıkardığı söylenebilir.

Şimdi de 750 karakterli metni inceleyince, en düşük ADKS 5 karakterli anahtarda 2, en yüksek ADKS 20 ve 25 karakterli anahtarlarda 12, en yüksek ortalama ADKS ise 8.4333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.0625, en düşük standart sapma ise 5 karakterli

şifrede 0.6814 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 204.2407 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise 5 karakterli anahtarlarda 265.9413 olduğu görülmektedir. Burada ise, 5 karakterli anahtara ait ve 24.9519 değerindeki en yüksek uygunluk standart sapmasını hemen görüyoruz. Ortalama uygunluk değeri ise en düşük olarak bu kez 20 karakterli anahtara ait olan 224.3153 olarak belirlenmiştir. Bu sonuçlara bakarak, düşük bir ADKS standart sapmasına sahip olan 5 karakterli şifrenin yetersiz bir sonuç verdiği ve 20 ile 25 karakterli anahtarların diğerlerine göre daha başarılı olduğu söylenebilir.

Son olarak ise, 1000 karakterli metin için en düşük ADKS 5 ve 15 karakterli anahtarlarda 2, en yüksek ADKS 25 karakterli anahtarda ise 13, en yüksek ortalama ADKS ise 8.5667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 2.2234, en düşük standart sapma ise 5 karakterli şifrede 0.7915 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 276.3453 ile 15 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 459.2877 olarak elde edilmiştir. Burada ise, yine 10 karakterli anahtarda 18.7291 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. En düşük ortalama uygunluk değeri ise 25 karakterli şifreye ait olan 303.2669 olarak belirlenmiştir. Bu sonuçlar değerlendirildiğinde, genel itibarı ile 10, 15 ve 25 karakterli anahtarların diğerlerine göre biraz daha üstün performans gösterdiği görülmüştür.

En genel ifadeyle anahtar boyutu arttıkça daha kısa şifreli metinleri çözmek için gerekli iterasyon sayısının arttığını söylemek mümkündür. Şifreli metnin boyutu arttıkça, anahtarın daha doğru sonuçlar verdiği görülmektedir. Ayrıca, doğru olarak elde edilen karakter sayısı da artmaktadır. Söz konusu karakter sayısındaki artış Türkçe metinlerinde daha net görülmektedir.

#### 4.2.2.3. DE Algoritmasının Sonuçları

Deneyler sonucunda DE ile elde edilen doğru karakter sayısı ve uygunluk değerleri şifreli İngilizce metinler için Tablo 4.7'de ve Türkçe metinler için Tablo 4.8'de gösterilmiştir.

Bu yöntem için, öncelikle uzunluğu 250 ile 1000 karakter arasında değişen şifreli İngilizce metinler incelenmiştir. İlk olarak 250 karakterli metinde en düşük ADKS her bir anahtar için 0 (sıfır), en yüksek ADKS ise 15 karakterli anahtarda 10, en yüksek ortalama ADKS ise 4.6667 değeriyle 5 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 10 karakterli anahtarda 3.4276, en düşük standart sapma ise 5 karakterli anahtarda 0.9589 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 55.0379 ile 15 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli şifrede 86.6142 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması ise 25 karakterli anahtarda 0.8186 değerinde hesaplanmıştır. En düşük ortalama uygunluk değeri ise yine 20 karakterli anahtara ait ve 56.9359 olarak belirlenmiştir. Burada, özellikle uygunluk değerlerine ait veriler karşılaştırıldığında 25 karakterli anahtarın daha kararlı sonuçlar verdiği, fakat 15 karakterli anahtarın daha başarılı olduğu söylenebilir.

Uzunluğu 500 karakter olan şifreli metne bakıldığında, en düşük ADKS 5 karakterli anahtar için 5, en yüksek ADKS 25 karakterli anahtarda 24, en yüksek ortalama ADKS ise 23.9 değeriyle ise 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 25 karakterli anahtarda 0.3051, en düşük standart sapma ise 5 karakterli şifrede 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 190.9625 ile 10 karakterli anahtarda, en yüksek uygunluk değeri ise 25 karakterli anahtarda 206.3602 olarak elde edilmiştir. Burada, 10 karakterli şifreye ait olan 3.7824 değerindeki en yüksek uygunluk standart sapması görülmektedir. En düşük standart sapma ise 5 karakterli anahtara ait olan 0 (sıfır) olarak hesaplanmıştır. En yüksek ortalama uygunluk değeri ise 25 karakterli anahtarda 207.7855 iken, en düşük ortalama uygunluk değeri ise 204.8769 olarak 10 karakterli anahtarda bulunmuştur. Bu verilere göre, uygunluk değerinin standart sapması da 0 olmasına rağmen, 5 karakterli anahtarın verimsiz olduğu ve 25 karakterli anahtarın en başarılı sonuçları ortaya çıkardığı söylenebilir.

750 karakterli metin değerlendirildiğinde, en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarlarda 25, en yüksek ortalama ADKS ise 24.9 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli anahtarda 0.3051, en düşük standart sapma ise 5, 10, 15 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise,

farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 334.7255 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlarda eşit olarak 342.4439 olarak elde edilmiştir. Burada ise, 25 karakterli anahtar üzerinde 1.6042 değerinde en yüksek uygunluk standart sapması elde edilirken, ortalama uygunluk değeri ise en düşük olarak yine 25 karakterli şifreye ait olan 341.966 olarak belirlenmiştir. Bu sonuçlar neticesinde, daha kısa olan anahtarların standart sapmaları 0 olarak ortaya çıkmış olsa da, uygunluk değeri standart sapması 1.6042 olan 25 karakterli şifrenin diğerlerine göre daha başarılı olduğu söylenebilir.

Son olarak, 1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda bu kez 5, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 24.8667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma ise 20 karakterli şifrede 0.4026, en düşük standart sapma ise 5 ve 10 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 440.0519 ile 20 karakterli anahtarda, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlarda eşit olarak 472.2756 olarak elde edilmiştir. Burada ise, 20 karakterli anahtara ait en yüksek uygunluk standart sapma 6.6081 olarak ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 25 karakterli şifreye ait olan 470.4147 olarak belirlenmiştir. Bu veriler değerlendirildiğinde, uygunluk değerleri birbirine yakın olmasına rağmen 25 karakterli anahtarın diğerlerine göre daha üstün performans gösterdiğini söylemek mümkündür.

Öte yandan, uzunluğu 250 ile 1000 karakter arasında değişen şifreli Türkçe metinlerinden oluşan verilere bakıldığında, buradaki 250 karakterli metinde en düşük ADKS 10 karakterli anahtarda 2, en yüksek ADKS ise 20 karakterli anahtarda 16, en yüksek ortalama ADKS ise 11 değeriyle 15 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 20 karakterli şifrede 2.6987, en düşük standart sapma ise 5 karakterli şifrede 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 64.4715 ile 20 karakterli anahtarda, en yüksek uygunluk değeri ise 15 karakterli anahtarda 88.4361 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması ise 5 karakterli şifreye ait ve yine 0 (sıfır) değerinde ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 25 karakterli anahtara ait ve 70.4414 olarak belirlenmiştir. Burada, 15 ve 20 karakterli anahtarların diğerlerinden biraz daha öne çıktığı söylenebilir.

Uzunluğu 500 karakter olan şifreli metne göz atılacak olursa, en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 24, en yüksek ortalama ADKS ise 24 değeriyle yine 25 karakterli anahtarda elde edildiği görülür. Bunlar içinde, en yüksek standart sapma ise 20 karakterli şifrede 0.1826, en düşük standart sapma ise 5, 10, 15 ve 25 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 204.6937 ile 20 karakterli anahtarda, en yüksek uygunluk değeri ise 25 karakterli anahtarda 209.5443 olarak elde edilmiştir. Burada ise, 20 karakterli anahtara ait ve 0.7463 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. En düşük standart sapma ise 5, 10, 15 ve 25 karakterli anahtarlarda 0 (sıfır) ortaya çıkmıştır. En yüksek ortalama uygunluk değeri ise yine 25 karakterli anahtarda 209.5443 ve en düşük ortalama uygunluk değeri ise 208.7812 olarak 5, 10, 15 karakterli anahtarlarda görülmüştür. Bu değerler neticesinde, diğerlerine kıyas edilerek daha yüksek bir uygunluk değerine sahip olmasına rağmen genel performansıyla 25 karakterli anahtarın diğerlerinden daha başarılı ve kararlı sonuçlar verdiği sonucuna varılabilir.

750 karakterli metin incelendiğinde, en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 24.9333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 0.3651, en düşük standart sapma ise 5, 10 ve 15 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 317.8508 ile 25 karakterli şifrede, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlar için eşit olarak 332.4877 olduğu görülmektedir. Burada ise, 20 karakterli anahtara ait ve 2.9814 değerindeki en yüksek uygunluk standart sapması belirmiştir. Ortalama uygunluk değeri ise en düşük olarak bu kez 20 karakterli anahtara ait olan 331.705 olarak belirlenmiştir. Bu sonuçlara bakarak, 0 (sıfır) değerinde ADKS standart sapmasına sahip olan 5, 10, 15 karakterli anahtarların yetersiz bir sonuç verdiği ve 25 karakterli anahtarın yine diğerlerine göre daha başarılı olduğu sonucuna varılabilir.

Son olarak ise, 1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda 4, en yüksek ADKS 25 karakterli anahtarda ise 25, en yüksek ortalama ADKS ise 24.9333 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli şifrede 0.2537, en düşük standart sapma ise 10, 15 ve 20 karakterli

anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 393.5308 ile 5 karakterli anahtarda, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlar için eşit olarak 459.2877 olarak elde edilmiştir. Burada ise, 5 karakterli anahtarda oldukça yüksek olan 12.0055 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. En düşük ortalama uygunluk değeri ise 5 karakterli anahtara ait olan 457.0958 olarak belirlenmiştir. Bu sonuçlar değerlendirildiğinde, genel performansı ile yine 25 karakterli anahtarın diğerlerinden biraz daha üstün performans gösterdiği görülmüştür.

Genel olarak, anahtar boyutu arttıkça daha kısa şifreli metinleri çözmek için gerekli iterasyon sayısı artmaktadır. Şifreli metnin boyutu arttıkça, anahtarın daha doğru sonuçlar verdiği görülmektedir. Ayrıca, doğru olarak elde edilen karakter sayısı da artmaktadır. Söz konusu karakter sayısındaki artış Türkçe metinlerinde daha net görülmektedir.

#### 4.2.2.4. ABC Algoritmasının Sonuçları

Deneyler sonucunda ABC ile elde edilen ADKS ve uygunluk değerleri şifreli İngilizce metinler için Tablo 4.9'da ve Türkçe metinler için Tablo 4.10'da gösterilmiştir.

Bu yöntem için de, yine uzunluğu 250 ile 1000 karakter arasında değişen şifreli İngilizce metinler incelenmiştir. İlk olarak 250 karakterli metinde en düşük ADKS her bir anahtar için 0 (sıfır), en yüksek ADKS ise 5 karakterli anahtarda 4, en yüksek ortalama ADKS ise 2.5333 değeriyle 5 karakterli anahtarda elde edilmiştir. Burada en yüksek ADKS standart sapması ise yine 5 karakterli anahtarda 0.8996, en düşük standart sapma ise 10 karakterli anahtarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 48.5052 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 64.0267 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması 20 karakterli anahtarda 0.4811 değerinde hesaplanmıştır. En düşük ortalama uygunluk değeri 20 karakterli anahtarda 49.8444 olarak belirlenmiştir. Burada, genel ADKS değerlerine bakıldığında 5 ve 25 karakterli anahtarların daha kararlı sonuçlar verdiği, fakat 15 karakterli anahtarın daha başarılı olduğu söylenebilir.

Uzunluğu 500 karakter olan şifreli metne bakıldığında, en düşük ADKS 20 ve 25 karakterli anahtarlarda 1, en yüksek ADKS 25 karakterli anahtarda 9, en yüksek ortalama ADKS ise 4.2667 değeriyle ise 10 ve 20 karakterli anahtarlarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 10 ve 20 karakterli anahtarlarda 4.2667, en düşük standart sapma ise 5 karakterli şifrede 0.4138 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 108.9989 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 205.8708 olarak elde edilmiştir. Burada, 5 karakterli şifreye ait olan 3.7824 değerindeki en yüksek uygunluk standart sapması göze çarpmaktadır. En düşük standart sapma ise 20 karakterli anahtarda 1.9824 olarak hesaplanmıştır. En yüksek ortalama uygunluk değeri ise 5 karakterli anahtarda 167.5250 iken, en düşük ortalama uygunluk değeri ise 113.3820 olarak 25 karakterli anahtarda bulunmuştur. Bu verilere göre, uygunluk değerinin standart sapması da düşük kalmasına rağmen, 5 karakterli anahtarın yetersiz kaldığı ve 25 karakterli anahtarın en başarılı sonuçları ortaya çıkardığı söylenebilir.

750 karakterli metin değerlendirildiğinde, en düşük ADKS 5, 15, 20 ve 25 karakterli anahtarlarda 3, en yüksek ADKS 10, 20 ve 25 karakterli anahtarlarda 7, en yüksek ortalama ADKS ise 24.9 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 25 karakterli anahtarda 0.3051, en düşük standart sapma ise 5, 10, 15 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 177.3535 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 342.4439 olarak elde edilmiştir. Burada ise, 5 karakterli anahtara ait 17.292 değerindeki en yüksek uygunluk standart sapması hesaplanırken, ortalama uygunluk değeri ise en düşük olarak yine 25 karakterli anahtara ait olan 181.5711 olarak belirlenmiştir. Bu sonuçlar neticesinde, anahtarların genel berberinin birbirine yakın bir performans gösterdiğini söylemek mümkündür.

1000 karakterli metin için en düşük ADKS 20 ve 25 karakterli anahtarlarda 3 olurken, en yüksek ADKS 20 karakterli anahtarda 8, en yüksek ortalama ADKS ise 5.5667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma ise 20 karakterli anahtarda 1.2452, en düşük standart sapma ise 5 karakterli anahtarda 0.3457 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı

uzunluktaki anahtarlar için en düşük uygunluk değeri 247.9509 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 472.2756 olarak elde edilmiştir. Burada ise, yine 5 karakterli anahtara ait en yüksek uygunluk standart sapma 5.3756 olarak ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 25 karakterli anahtara ait olan 254.6403 olarak belirlenmiştir. Bu veriler değerlendirildiğinde, genel performansı değerlendirildiğinde 25 karakterli anahtarın diğerlerine göre biraz daha fazla verimli olduğu söylenebilir.

Uzunluğu 250 ile 1000 karakter arasında değişen şifreli Türkçe metinlerden oluşan verilere bakıldığında, buradaki 250 karakterli metinde en düşük ADKS 10, 15, 20, 25 karakterli anahtarlarda 0 (sıfır), en yüksek ADKS ise her bir uzunluktaki anahtarlarda eşit olarak 5, en yüksek ortalama ADKS ise 3.9667 değeriyle 5 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 15 karakterli anahtarda 1.6333, en düşük standart sapma ise 5 karakterli şifrede 0.4138 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 49.7042 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 87.6893 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması ise 20 karakterli anahtarda 1.1948 değerinde ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 25 karakterli anahtara ait ve 51.7996 olarak belirlenmiştir. Burada, tüm anahtarlar benzer performans göstermiş olsa da 20 karakterli anahtarın biraz daha karalı görünmesinden dolayı diğerlerinden biraz daha öne çıktığını söylenebilir.

Uzunluğu 500 karakterde olan şifreli metinde ise, en düşük ADKS 20 ve 25 karakterli anahtarlarda 2, en yüksek ADKS 15, 20 ve 25 karakterli anahtarlarda 7, en yüksek ortalama ADKS ise 1.4320 değeriyle yine 25 karakterli anahtarda elde edildiği görülür. Bunlar içinde, en yüksek standart sapma ise 20 karakterli şifrede 0.1826, en düşük standart sapma ise 5, 10, 15 ve 25 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 114.3958 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 208.7812 olarak elde edilmiştir. Burada ise, 5 karakterli anahtara ait ve 14.1103 değerindeki oldukça yüksek olan uygunluk standart sapması göze çarpmaktadır. En düşük standart sapma ise 25 karakterli anahtarda 2.1761 değerinde ortaya çıkmıştır. En yüksek ortalama uygunluk değeri ise 5 karakterli anahtarda 194.2237 ve en düşük ortalama uygunluk değeri ise 118.7351 olarak 25

karakterli anahtarda görülmüştür. Bu değerler neticesinde, 15, 20 ve 25 karakterli anahtarların benzer performans göstererek diğerlerinden biraz daha başarılı ve kararlı sonuçlar verdiği değerlendirmesi yapılabilir.

750 karakterli metin incelendiğinde, en düşük ADKS 20 ve 25 karakterli anahtarlarda 3, en yüksek ADKS ise yine 20 ve 25 karakterli anahtarlarda 8, en yüksek ortalama ADKS ise 5.5333 değeriyle 15 karakterli anahtarda rastlanmıştır. Bunlar içinde, en yüksek standart sapma 20 karakterli anahtarda 1.2229, en düşük standart sapma ise 5 karakterli anahtarda 0.504 olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 184.1261 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 332.4877 olduğu görülmektedir. Burada ise, 5 karakterli anahtara ait ve 24.0569 değerindeki en yüksek uygunluk standart sapması kendini göstermektedir. Ortalama uygunluk değeri ise en düşük olarak 25 karakterli anahtara ait olan 187.5224 olarak belirlenmiştir. Bu sonuçlara bakarak, ADKS standart sapmaları göz önüne alındığında 5 ve 10 karakterli anahtarların biraz daha kararlı sonuç verdiği, yine de en iyi sonuçlar 20 ve 25 karakterli anahtarlarda elde edildiği görülmektedir.

1000 karakterli metin için en düşük ADKS 20 karakterli anahtarda 3, en yüksek ADKS 20 ve 25 karakterli anahtarlarda ise 8, en yüksek ortalama ADKS ise 5.5 değeriyle 20 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 20 karakterli anahtarda 1.2247, en düşük standart sapma ise 5 karakterli anahtarda 0.5074 olarak hesaplanmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 252.1442 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise 25 karakterli anahtarda 459.2877 olarak elde edilmiştir. Burada ise, 5 karakterli anahtarda çok yüksek olan 34.8339 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. En düşük ortalama uygunluk değeri ise 20 karakterli anahtara ait olan 260.0149 olarak belirlenmiştir. Bu sonuçlar değerlendirildiğinde, genel performansı ile 20 ve 25 karakterli anahtarların diğerlerinden biraz daha önde bir görüntü verdiği üstün performans gösterdiği görülmüştür.

Daha genel olarak, görüldü ki anahtar boyutu arttıkça daha kısa şifreli metinleri çözmek için gerekli iterasyon sayısı artmaktadır. Şifreli metnin boyutu arttıkça, anahtarın daha doğru sonuçlar verdiği görülmektedir. Ayrıca, doğru olarak elde edilen karakter sayısı da artmaktadır. Söz konusu karakter sayısındaki artış Türkçe metinlerinde daha net görünmektedir.

#### 4.2.2.5. BCABC Algoritmasının Sonuçları

Deneyler sonucunda BCABC ile elde edilen ADKS ve uygunluk değerleri şifreli İngilizce metinler için Tablo 4.11'de ve Türkçe metinler için Tablo 4.12'de gösterilmiştir.

BCABC yöntemini incelemek için yine uzunluğu 250 ile 1000 karakter arasında değişen şifreli İngilizce metinler kullanılmıştır. İlk olarak 250 karakterli metinde en düşük ADKS her bir anahtar için 0 (sıfır), en yüksek ADKS ise 5 karakterli anahtarda 5, en yüksek ortalama ADKS ise 1.7340 değeriyle 5 karakterli anahtarda elde edilmiştir. Burada en yüksek ADKS standart sapması ise yine 5 karakterli anahtarda 1.7340, en düşük standart sapma ise 10, 15 ve 20 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 53.9846 ile 10 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 86.6142 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapması 10 karakterli anahtarda 0.7367 değerinde hesaplanmıştır. En düşük ortalama uygunluk değeri yine 10 karakterli anahtarda 56.2596 olarak belirlenmiştir. Burada, genel ADKS değerlerine bakıldığında 5 karakterli anahtarın daha başarılı olduğu açıkça ortaya çıkmıştır.

Ardından uzunluğu 500 karakterde olan şifreli metne bakıldığında, en düşük ADKS 20 karakterli anahtarda 4, en yüksek ADKS 25 karakterli anahtarda 24, en yüksek ortalama ADKS ise 21.7667 değeriyle ise 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma ise 20 karakterli anahtarda 3.7516, en düşük standart sapma ise 5 karakterli anahtarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 126.8753 ile 20 karakterli anahtarda, en yüksek uygunluk değeri ise 25 karakterli anahtarda 207.9439 olarak elde edilmiştir. Burada, 20 karakterli şifreye ait olan 21.1167 değerindeki en yüksek uygunluk standart sapması kendini göstermektedir. En düşük

standart sapma ise 5 karakterli anahtarda 0 (sıfır) olarak hesaplanmıştır. En yüksek ortalama uygunluk değeri ise 5 karakterli anahtarda 205.8708 iken, en düşük ortalama uygunluk değeri ise 165.6999 olarak 20 karakterli anahtarda bulunmuştur. Bu veriler bize, uygunluk değerinin oldukça yüksek kalmasına rağmen, 25 karakterli anahtarın diğerlerinden daha verimli odluğu ve en başarılı sonuçları ortaya çıkardığı söylenebilir.

Sıradaki 750 karakterli metin değerlendirildiğinde, en düşük ADKS 5 karakterli anahtarlarda 5, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 24.5667 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma 15 karakterli anahtarda 1.3734, en düşük standart sapma ise 5 karakterli anahtarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 273.3619 ile 15 karakterli anahtarda, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarda eşit ve 342.4439 olarak elde edilmiştir. Burada ise, 15 karakterli anahtara ait 19.9249 değerindeki en yüksek uygunluk standart sapması hesaplanırken, buna rağmen ortalama uygunluk değeri ise en düşük olarak yine 15 karakterli anahtarda 314.6871 elde edilmiştir. Bu sonuçlar neticesinde, 0 (sıfır) standart sapmalarına sahip olan 5 karakterli anahtarın sabit olarak aynı sonucunu verdiğini görüyoruz. En başarılı sonuçların ise 25 karakterli anahtarda edildiği görülmektedir.

1000 karakterli metin için en düşük ADKS 5 karakterli anahtarlarda 5 olurken, en yüksek ADKS 25 karakterli anahtarda 25 ve en yüksek ortalama ADKS ise 24.2 değeriyle yine 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma ise 15 karakterli anahtarda 1.1943, en düşük standart sapma ise 5 ve 10 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 397.4719 ile 15 karakterli anahtarda, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlarda eşit ve 472.2756 olarak elde edilmiştir. Burada ise, 15 karakterli anahtara ait en yüksek uygunluk standart sapma 23.9817 olarak ortaya çıkmıştır. En düşük ortalama uygunluk değeri ise 25 karakterli anahtara ait olan 461.1489 olarak belirlenmiştir. Bu veriler değerlendirildiğinde, genel performanslar arasında 25 karakterli anahtarın yine diğerlerinden daha önde olduğu, 0 standart sapmasına sahip olan 5 ve 10 karakterli anahtarların kararlı görüntülerine rağmen yetersiz kaldıklarını söylenebilir.

Uzunluğu 250 ile 1000 karakter arasında değişen şifreli Türkçe metinlerden oluşan verilere bakıldığında, buradaki 250 karakterli metinde en düşük ADKS 10 ve 15 karakterli anahtarlarda 0 (sıfır), en yüksek ADKS ise 20 ve 25 karakterli anahtarlarda eşit olarak 13, en yüksek ortalama ADKS ise 7.9333 değeriyle 25 karakterli anahtarda elde edilmiştir. Burada en yüksek standart sapma ise 20 karakterli anahtarda 3.1878, en düşük standart sapma ise 5 karakterli anahtarda 0 (sıfır) olarak hesap edilmiştir. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 60.4729 ile 15 karakterli anahtarda, en yüksek uygunluk değeri ise 5 karakterli anahtarda 87.6893 olarak elde edilmiştir. Ayrıca, en düşük uygunluk standart sapma ise 5 karakterli anahtarda 0 (sıfır) değerini almıştır. En düşük ortalama uygunluk değeri ise 10 karakterli anahtara ait ve 64.5094 olarak belirlenmiştir. Bu defa, 20 ve 25 karakterli anahtarların diğerlerinden daha üstün bir performans gösterdiği açıkça ortaya çıkmıştır.

Uzunluğu 500 karakterde olan şifreli metinde ise, en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 23.7333 değeriyle yine 25 karakterli anahtarda elde edildiği görülür. Bunlar içinde, en yüksek standart sapma ise 20 karakterli anahtarda 0.8503, en düşük standart sapma ise 5 ve 15 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 192.6834 ile 10 karakterli anahtarda, en yüksek uygunluk değeri ise 15 karakterli anahtarda 208.7812 olarak elde edilmiştir. Burada ise, 10 karakterli anahtara ait ve 3.9313 değerindeki uygunluk standart sapması belirmiştir. En düşük standart sapma ise 5 ve 15 karakterli anahtarlarda 0 (sıfır) değerinde ortaya çıkmıştır. En yüksek ortalama uygunluk değeri ise 5 ve 15 karakterli anahtarlarda 208.7812 ve en düşük ortalama uygunluk değeri ise 207.1740 olarak 20 karakterli anahtarda görülmüştür. Bu değerler neticesinde, daha önce de olduğu gibi benzer şekilde 25 karakterli anahtarın daha başarılı ve kararlı sonuçlar verdiği değerlendirmesi yapılabilir.

Ardından 750 karakterli metin değerlendirildiğinde, en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS ise 25 karakterli anahtarlarda 25, en yüksek ortalama ADKS ise 24.7333 değeriyle 25 karakterli anahtarda rastlanmıştır. Bunlar içinde, en yüksek standart sapma yine 25 karakterli anahtarda 0.4498, en düşük standart sapma ise 5 ve 10 karakterli anahtarlarda 0 (sıfır) olarak ortaya çıkmıştır. Uygunluk değerlerine

bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 316.1677 ile 15 karakterli anahtarda, en yüksek uygunluk değer ise her bir uzunluktaki anahtarda eşi olarak 332.4877 değerinde ortaya çıkmıştır. Burada ise, 15 karakterli anahtara ait olan 2.9796 değeri en yüksek uygunluk standart sapması olmuştur. Ortalama uygunluk değeri ise en düşük olarak 25 karakterli anahtara ait olan 331.4456 olarak belirlenmiştir. Bu sonuçlara bakarak, ADKS değerlerine bakıldığında, en iyi sonuçların 25 karakterli anahtarda oluştuğu görülmüştür.

1000 karakterli metin için en düşük ADKS 5 karakterli anahtarda 5, en yüksek ADKS 25 karakterli anahtarda 25, en yüksek ortalama ADKS ise 24.8 değeriyle 25 karakterli anahtarda elde edilmiştir. Bunlar içinde, en yüksek standart sapma yine 25 karakterli anahtarda 0.4068, en düşük standart sapma ise 5 ve 10 karakterli anahtarlarda 0 (sıfır) olarak hesaplanmıştır. Uygunluk değerlerine bakıldığında ise, farklı uzunluktaki anahtarlar için en düşük uygunluk değeri 431.1108 ile 25 karakterli anahtarda, en yüksek uygunluk değeri ise her bir uzunluktaki anahtarlarda eşit olarak 459.2877 olarak elde edilmiştir. Burada ise, 20 karakterli anahtarda 6.8726 değerindeki en yüksek uygunluk standart sapması elde edilmiştir. En düşük ortalama uygunluk değeri ise 20 karakterli anahtara ait olan 456.7755 olarak belirlenmiştir. Bu sonuçlar değerlendirildiğinde, bir kez daha görülüyor ki genel performansı ile 25 karakterli anahtar diğerlerinden daha üstün performans göstermiştir.

Tablo 4.11 ve 4.12'ye göre, anahtar boyutu arttıkça daha kısa şifreli metinleri çözmek için gerekli iterasyon sayısı artmaktadır. Şifreli metnin boyutu arttıkça, anahtarın daha doğru sonuçlar verdiği görülmektedir. Ayrıca, doğru olarak elde edilen karakter sayısı da artmaktadır. Söz konusu karakter sayısındaki artış Türkçe metinlerinde daha net görünmektedir.

Tablo 4.3. Şifreli İngilizce metin ve anahtar boyutuna bağlı olarak GA sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **5** | **3.2333** | 1.8323 | **53.5698** | **86.6142** | **69.0764** | **11.7235** |
| | 10 | **0** | 6 | 0.5333 | 1.5916 | 53.6780 | 60.1863 | 56.1366 | 1.4144 |
| | 15 | **0** | **11** | 0.9333 | **2.4059** | 53.7704 | 74.0957 | 58.6584 | 4.1922 |
| | 20 | **0** | 7 | **0.3333** | 1.2954 | 55.9292 | 68.8373 | 59.0084 | 2.3885 |
| | 25 | **0** | 3 | 0.5000 | 0.9738 | 57.9293 | 69.6755 | 62.4770 | 2.1533 |
| 500 | 5 | **4** | **5** | 4.9667 | **0.1826** | 165.6981 | **205.8708** | **204.5317** | 7.3345 |
| | 10 | 8 | **10** | 9.4333 | 0.6261 | 164.8473 | **205.8708** | 196.8990 | 10.7718 |
| | 15 | 8 | **15** | 13.1000 | 1.6049 | 159.0927 | **205.8708** | 192.0746 | **11.7197** |
| | 20 | 12 | **20** | 14.7333 | 2.0833 | 162.9813 | **205.9255** | 182.0374 | 11.0484 |
| | 25 | 10 | 23 | 16.9000 | **2.8929** | **156.3814** | 199.4555 | 180.4832 | **9.7627** |
| 750 | 5 | **5** | **5** | **5.0000** | **0.0000** | **342.4439** | **342.4439** | **342.4439** | **0.0000** |
| | 10 | **8** | **10** | 9.6667 | 0.5467 | 305.3219 | **342.4439** | 334.4502 | 13.0033 |
| | 15 | **10** | **15** | 13.4333 | 1.5013 | 275.3371 | **342.4439** | 316.0985 | **23.3129** |
| | 20 | **14** | **20** | 16.9333 | 1.5960 | 281.6019 | **342.4439** | 310.2493 | 14.6019 |
| | 25 | **14** | **25** | **20.4000** | **2.4439** | **269.3495** | **342.4439** | **309.6918** | 16.8850 |
| 1000 | 5 | **4** | **5** | **4.9667** | **0.1826** | 395.6219 | **472.2756** | **469.7205** | **13.9950** |
| | 10 | 9 | **10** | 9.7667 | 0.4302 | **423.7219** | **472.2756** | 463.2532 | 16.9758 |
| | 15 | 11 | **15** | 13.4667 | 1.4320 | 387.3119 | **472.2756** | 438.6265 | 28.4480 |
| | 20 | 14 | **20** | 17.6667 | 1.5610 | 390.6119 | **472.2756** | 436.6399 | 21.7241 |
| | 25 | **18** | **25** | **21.2000** | **2.0578** | **393.9319** | **472.2756** | **429.0244** | **23.2646** |

Tablo 4.4. Şifreli Türkçe metin ve anahtar boyutuna bağlı olarak GA sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **2** | **5** | **4.6333** | **0.7184** | 59.6936 | 87.6893 | **83.9274** | 7.0960 |
| | 10 | **0** | **10** | 7.7667 | 2.3295 | **58.6613** | 87.6893 | 80.3693 | **7.5764** |
| | 15 | 1 | 13 | 7.8000 | 3.2632 | 64.4196 | **87.1034** | 77.4610 | **5.2800** |
| | 20 | **0** | 15 | 6.9000 | 3.5365 | 65.0445 | 88.3858 | **73.3611** | 5.5266 |
| | 25 | **2** | 15 | **8.0667** | **3.7040** | **66.5045** | **89.5371** | 78.3575 | 5.8534 |
| 500 | 5 | **5** | **5** | **5.0000** | **0.0000** | **208.7812** | **208.7812** | **208.7812** | **0.0000** |
| | 10 | **8** | **10** | **9.6333** | 0.5561 | 179.6737 | **208.7812** | 203.5036 | 8.1217 |
| | 15 | 11 | **15** | 13.9333 | 1.1427 | 174.7443 | **208.7812** | 199.9356 | **9.4435** |
| | 20 | **8** | 19 | 15.7000 | 2.4233 | 172.4416 | **204.7668** | 190.9227 | 8.0900 |
| | 25 | 14 | 24 | 18.9667 | **2.8945** | **171.8244** | 204.9300 | **190.3536** | 9.2062 |
| 750 | 5 | 5 | **5** | **5.0000** | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 10 | **8** | **10** | 9.7333 | 0.5208 | 290.9965 | **332.4877** | 326.5140 | 11.6262 |
| | 15 | **12** | **15** | 13.9000 | 1.0619 | 290.9108 | **332.4877** | 317.5800 | 13.8280 |
| | 20 | **12** | **20** | 17.2667 | 1.8182 | 265.7408 | **332.4877** | 309.0320 | 14.1485 |
| | 25 | **10** | 23 | **19.1333** | **2.5695** | **246.1968** | 328.1708 | 298.0012 | **16.4095** |
| 1000 | 5 | **4** | **5** | **4.9667** | **0.1826** | 390.4677 | **459.2877** | **456.9937** | **12.5648** |
| | 10 | **8** | **10** | 9.6333 | 0.6149 | **394.3877** | **459.2877** | 447.5076 | 19.2021 |
| | 15 | **11** | **15** | 13.9000 | 1.1552 | **368.2908** | **459.2877** | 437.1538 | **23.5861** |
| | 20 | **14** | **20** | 17.2667 | 1.8182 | 379.7608 | **459.2877** | 422.4303 | 20.9674 |
| | 25 | **17** | **25** | **21.0667** | **2.0998** | 382.3808 | **459.2877** | **421.7398** | 21.3322 |

Tablo 4.5. Şifreli İngilizce metin ve anahtar boyutuna bağlı olarak PSO sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **5** | **1.3667** | **1.5862** | 52.1522 | **86.6142** | **59.6482** | **7.5217** |
| | 10 | **0** | 4 | 0.5333 | 1.2521 | **50.4001** | 58.8240 | 54.5818 | 1.9792 |
| | 15 | **0** | 4 | **0.2333** | **0.7739** | 51.9975 | 57.8687 | **54.2724** | **1.4838** |
| | 20 | **0** | 5 | 0.6333 | 1.2452 | 52.4628 | 59.6172 | 55.8124 | 1.7471 |
| | 25 | **0** | 5 | 0.5667 | 1.1043 | 51.8429 | 60.2802 | 56.1749 | 2.2291 |
| 500 | 5 | **1** | **5** | **3.3000** | **1.0875** | 124.6192 | **205.8708** | **156.8410** | **24.6460** |
| | 10 | 2 | 7 | 5.1000 | 1.1552 | 124.0404 | 172.6860 | 149.5647 | 11.8037 |
| | 15 | 2 | 11 | 6.7333 | 2.0667 | 122.7424 | 173.2608 | 142.1190 | 10.6438 |
| | 20 | **1** | 11 | 6.3000 | 2.2614 | **119.5057** | 151.7895 | 135.2078 | 8.6860 |
| | 25 | 3 | 13 | **7.6333** | **3.0113** | 123.3494 | 153.8444 | 136.1634 | **8.6591** |
| 750 | 5 | **2** | **5** | **3.4333** | **0.8172** | 212.4944 | **342.4439** | **259.0414** | **33.7318** |
| | 10 | 3 | 9 | 5.5667 | 1.4547 | 220.2844 | 308.4972 | 251.1021 | 20.7271 |
| | 15 | 4 | 12 | 7.4000 | 2.0443 | 202.9892 | 281.3719 | 239.5816 | 21.0357 |
| | 20 | 5 | 11 | 7.6667 | 1.4933 | 203.8248 | 255.8072 | 223.6455 | 11.8842 |
| | 25 | **2** | 12 | **8.4333** | **2.4023** | **192.3029** | 241.7219 | **216.9870** | **11.6850** |
| 1000 | 5 | **0** | **5** | **3.5333** | **0.9732** | **273.1470** | 472.2756 | **369.1274** | **48.8020** |
| | 10 | 4 | 8 | 5.7333 | 1.1427 | 285.2263 | **399.2419** | 344.0369 | 23.0306 |
| | 15 | 4 | 10 | 7.1333 | 1.3578 | 287.5219 | 363.0519 | 323.3344 | 20.7343 |
| | 20 | 4 | 11 | 8.0000 | 1.9652 | 280.5701 | 346.6519 | 313.2915 | **19.0091** |
| | 25 | 4 | 18 | **9.9333** | **2.6901** | 274.7586 | 381.0719 | **310.6931** | 21.4984 |

Tablo 4.6. Şifreli Türkçe metin ve anahtar boyutuna bağlı olarak PSO sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **5** | 3.3333 | **0.9223** | **54.9033** | **87.6893** | **71.8616** | **7.3600** |
| | 10 | **0** | 7 | **3.6667** | 1.4223 | 56.6910 | 74.6070 | 64.9737 | 4.8416 |
| | 15 | **0** | 8 | **2.8667** | 2.1772 | 55.3411 | 70.0893 | 62.6000 | 3.6686 |
| | 20 | **0** | 8 | 3.1333 | **2.3004** | 56.6861 | 70.7564 | **62.4331** | 3.5329 |
| | 25 | **0** | 9 | 3.2333 | 1.7750 | 56.9423 | 71.5053 | 65.0669 | **3.3676** |
| 500 | 5 | **2** | **5** | **3.5667** | **0.6261** | 142.5457 | **208.7812** | **168.7199** | **15.0480** |
| | 10 | 3 | 7 | 5.1333 | 0.9732 | 140.8200 | 168.4371 | 154.5469 | 8.0682 |
| | 15 | 4 | 9 | 6.0000 | 1.2034 | 129.6609 | 157.5246 | 143.7117 | 7.5786 |
| | 20 | 3 | 10 | 6.4333 | 1.4782 | 129.6873 | 153.1097 | 143.3478 | **5.6208** |
| | 25 | 4 | 12 | **7.2000** | **2.1560** | **128.7282** | 154.6983 | **143.3216** | 5.8300 |
| 750 | 5 | **2** | **5** | **3.5333** | **0.6814** | 222.7208 | **332.4877** | **265.9413** | **24.9519** |
| | 10 | 3 | 7 | 5.2333 | 1.1943 | 204.9312 | 270.7537 | 242.3452 | 15.0759 |
| | 15 | 4 | 8 | 5.9333 | 1.3113 | 211.0540 | 252.7488 | 230.6005 | 11.4627 |
| | 20 | 4 | 12 | 7.3667 | 1.7317 | 205.4999 | 256.2708 | 224.3153 | 12.0862 |
| | 25 | 4 | 12 | **8.4333** | **2.0625** | **204.2407** | 247.2752 | **226.6884** | **10.1333** |
| 1000 | 5 | **2** | **5** | **3.8333** | **0.7915** | **298.0023** | **459.2877** | **381.9259** | **45.8663** |
| | 10 | 3 | 8 | 5.0667 | 1.0483 | 287.9268 | 388.0708 | 331.2852 | 18.7291 |
| | 15 | 2 | 10 | 6.5333 | 1.6344 | 276.3453 | 358.2808 | 316.5989 | 18.3243 |
| | 20 | 5 | 10 | 7.5333 | 1.2521 | 281.0108 | 330.9668 | 306.8192 | **13.1354** |
| | 25 | 3 | 13 | **8.5667** | 2.2234 | 278.3163 | 331.4208 | **303.2669** | 13.9922 |

Tablo 4.7. Şifreli İngilizce metin ve anahtar boyutuna bağlı olarak DE sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **5** | **4.6667** | **0.9589** | 61.0584 | **86.6142** | **82.6007** | **8.2841** |
| | 10 | **0** | 8 | 4.1000 | **3.4276** | 55.7505 | 72.3724 | 62.1395 | 5.5459 |
| | 15 | **0** | 10 | 2.0333 | 3.2851 | 55.7368 | 66.1111 | 58.1354 | 2.3933 |
| | 20 | **0** | 9 | 1.3667 | 2.4563 | **55.0379** | 61.3401 | 56.9359 | 1.4221 |
| | 25 | **0** | 5 | **1.2667** | 1.6386 | 56.5267 | 59.7120 | **57.8161** | **0.8186** |
| 500 | 5 | 5 | **5** | **5.0000** | **0.0000** | 205.8708 | 205.8708 | 205.8708 | **0.0000** |
| | 10 | 9 | **10** | 9.9333 | 0.2537 | **190.9625** | 205.8708 | **204.8769** | **3.7824** |
| | 15 | 14 | **15** | 14.9333 | 0.2537 | 197.4072 | 205.8708 | 205.5551 | 1.5498 |
| | 20 | **18** | 19 | 18.9667 | 0.1826 | 195.6625 | 205.9255 | 205.5551 | 1.8748 |
| | 25 | **23** | 24 | **23.9000** | **0.3051** | **206.3602** | **207.9439** | **207.7855** | 0.4832 |
| 750 | 5 | 5 | **5** | **5.0000** | **0.0000** | 342.4439 | **342.4439** | **342.4439** | **0.0000** |
| | 10 | 10 | **10** | 10.0000 | **0.0000** | 342.4439 | **342.4439** | **342.4439** | **0.0000** |
| | 15 | 15 | **15** | 15.0000 | **0.0000** | 342.4439 | **342.4439** | **342.4439** | **0.0000** |
| | 20 | **19** | **20** | 19.9667 | 0.1826 | 338.0419 | **342.4439** | **342.2972** | 0.8037 |
| | 25 | **24** | **25** | **24.9000** | **0.3051** | 334.7255 | **342.4439** | **341.9660** | **1.6042** |
| 1000 | 5 | 5 | **5** | **5.0000** | **0.0000** | **472.2756** | **472.2756** | **472.2756** | **0.0000** |
| | 10 | 10 | **10** | **10.0000** | **0.0000** | **472.2756** | **472.2756** | **472.2756** | **0.0000** |
| | 15 | 14 | **15** | 14.9667 | 0.1826 | 451.8462 | **472.2756** | **471.5946** | 3.7299 |
| | 20 | **18** | **20** | 19.9000 | **0.4026** | **440.0519** | **472.2756** | 470.6138 | 6.6081 |
| | 25 | 24 | **25** | **24.8667** | 0.3457 | 455.2919 | **472.2756** | **470.4147** | 4.9309 |

Tablo 4.8. Şifreli Türkçe metin ve anahtar boyutuna bağlı olarak DE sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | 5 | **5** | **5.0000** | **0.0000** | 87.6893 | 87.6893 | **87.6893** | **0.0000** |
| | 10 | 8 | **10** | 9.7667 | 0.5683 | 82.5173 | 87.6893 | 87.4076 | 1.0862 |
| | 15 | 6 | **15** | 11.0000 | 2.4635 | 77.4866 | **88.4361** | 86.3848 | 2.9394 |
| | 20 | 2 | 16 | 9.6000 | 2.6987 | **64.4715** | 86.8992 | 74.5462 | **4.7858** |
| | 25 | **3** | 12 | **8.1000** | **2.2491** | 65.5436 | 78.6273 | **70.4414** | 2.9952 |
| 500 | 5 | 5 | **5** | **5.0000** | **0.0000** | 208.7812 | 208.7812 | **208.7812** | **0.0000** |
| | 10 | 10 | **10** | **10.0000** | **0.0000** | 208.7812 | 208.7812 | **208.7812** | **0.0000** |
| | 15 | 15 | **15** | **15.0000** | **0.0000** | 208.7812 | 208.7812 | **208.7812** | **0.0000** |
| | 20 | **19** | **20** | 19.9667 | **0.1826** | **204.6937** | 208.7812 | 208.6450 | **0.7463** |
| | 25 | **24** | 24 | **24.0000** | **0.0000** | 209.5443 | **209.5443** | **209.5443** | **0.0000** |
| 750 | 5 | **5** | **5** | **5.0000** | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 10 | **10** | **10** | **10.0000** | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 15 | **15** | **15** | **15.0000** | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 20 | 19 | **20** | 19.9333 | 0.2537 | 320.2477 | **332.4877** | **331.7050** | 2.9814 |
| | 25 | 23 | **25** | 24.9333 | **0.3651** | **317.8508** | **332.4877** | 331.9998 | **2.6723** |
| 1000 | 5 | **4** | **5** | **4.9667** | 0.1826 | **393.5308** | **459.2877** | **457.0958** | **12.0055** |
| | 10 | 10 | **10** | **10.0000** | **0.0000** | 459.2877 | **459.2877** | **459.2877** | **0.0000** |
| | 15 | 15 | **15** | **15.0000** | **0.0000** | 459.2877 | **459.2877** | **459.2877** | **0.0000** |
| | 20 | 20 | **20** | **20.0000** | **0.0000** | 459.2877 | **459.2877** | **459.2877** | **0.0000** |
| | 25 | **24** | **25** | 24.9333 | **0.2537** | 444.4208 | **459.2877** | 458.2966 | 3.7719 |

Tablo 4.9. Şifreli İngilizce metin ve anahtar boyutuna bağlı olarak ABC sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **4** | **2.5333** | **0.8996** | 52.5114 | **64.0267** | **56.4020** | **2.7069** |
| | 10 | **0** | 0 | **0.0000** | **0.0000** | 50.5857 | 53.3889 | 51.5362 | 0.7267 |
| | 15 | **0** | 1 | 0.1000 | 0.3051 | 49.8574 | 53.0608 | 50.9561 | 0.8027 |
| | 20 | **0** | 1 | 0.0333 | 0.1826 | 49.0367 | 50.7284 | **49.8444** | **0.4811** |
| | 25 | **0** | 3 | 0.3667 | 0.7649 | **48.5052** | 51.7278 | 49.8536 | 0.9141 |
| 500 | 5 | 3 | **5** | 3.9667 | **0.4138** | 143.4218 | **205.8708** | **167.5250** | **12.4580** |
| | 10 | 3 | 5 | **4.2667** | 0.5208 | 130.4588 | 151.6807 | 138.8680 | 5.3378 |
| | 15 | 2 | 5 | **3.5333** | 0.9371 | 118.0250 | 129.3112 | 123.3281 | 3.2605 |
| | 20 | **1** | 6 | **4.2667** | 1.2576 | 110.1442 | 118.0029 | 114.1279 | **1.9824** |
| | 25 | **1** | 9 | 4.0667 | **1.5960** | **108.9989** | 127.9195 | **113.3820** | 3.6811 |
| 750 | 5 | 3 | **5** | **4.0333** | **0.3198** | 250.3419 | **342.4439** | **283.6307** | **17.2920** |
| | 10 | 4 | 7 | **4.9667** | 0.6687 | 219.5141 | 259.3519 | 234.3092 | 8.5797 |
| | 15 | 3 | 6 | 4.4333 | 0.6261 | 191.8620 | 225.1796 | 202.9465 | 6.7211 |
| | 20 | 3 | 7 | 4.7000 | 0.8769 | 181.6796 | 202.4107 | 188.8577 | 4.2668 |
| | 25 | **3** | 7 | 4.7333 | **0.9803** | **177.3535** | 189.7726 | **181.5711** | **3.3355** |
| 1000 | 5 | 4 | **5** | **4.1333** | **0.3457** | 380.3119 | **472.2756** | **400.8530** | **28.6436** |
| | 10 | 4 | 6 | 5.1000 | 0.6618 | 300.5131 | 359.7419 | 321.9401 | 14.3074 |
| | 15 | 4 | 7 | 4.6667 | 0.7581 | 268.1845 | 306.3119 | 282.6832 | 8.0079 |
| | 20 | 3 | 8 | 5.0333 | **1.2452** | 252.4580 | 282.1519 | 265.3535 | 7.0368 |
| | 25 | **3** | 7 | **5.5667** | 1.1943 | **247.9509** | 272.2407 | **254.6403** | **5.3756** |

Tablo 4.10. Şifreli Türkçe metin ve anahtar boyutuna bağlı olarak ABC sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | 3 | **5** | **3.9667** | **0.4138** | **66.6567** | 87.6893 | **75.4893** | **4.7468** |
| | 10 | **0** | 5 | 2.2000 | 1.4479 | 54.7390 | 62.8773 | 57.0546 | 2.0074 |
| | 15 | **0** | 5 | **1.5667** | **1.6333** | 51.2920 | 59.3545 | 53.6795 | 1.6610 |
| | 20 | **0** | 5 | 2.4667 | 1.4077 | 50.6885 | **55.7378** | 52.9640 | **1.1948** |
| | 25 | **0** | 5 | 1.6000 | 1.5222 | **49.7042** | 56.7146 | 51.7996 | 1.4357 |
| 500 | 5 | 4 | **5** | 4.4667 | **0.5074** | 174.7537 | **208.7812** | **194.2237** | **14.1103** |
| | 10 | 4 | 6 | 5.0667 | 0.5833 | 138.3754 | 159.5398 | 146.1941 | 5.4336 |
| | 15 | 4 | 7 | **5.1667** | 0.7915 | 121.9972 | 136.7473 | 127.1416 | 3.4820 |
| | 20 | 2 | 7 | **4.6000** | 1.1626 | 117.4816 | 129.8581 | 123.3861 | 2.9602 |
| | 25 | **2** | 7 | 4.4667 | **1.4320** | **114.3958** | 122.3029 | **118.7351** | **2.1761** |
| 750 | 5 | 4 | **5** | **4.4333** | **0.5040** | 282.2977 | **332.4877** | **305.5313** | **24.0569** |
| | 10 | 4 | 6 | 5.1000 | 0.6074 | 215.8774 | 250.6668 | 231.1823 | 8.3609 |
| | 15 | 4 | 7 | **5.5333** | 0.8193 | 195.0175 | 217.4108 | 204.5360 | 5.2829 |
| | 20 | 3 | 8 | 5.2333 | **1.2229** | 190.1940 | 217.3559 | 196.5783 | 5.7820 |
| | 25 | **3** | 8 | 4.6667 | 1.0933 | **184.1261** | 194.2468 | **187.5224** | **2.5900** |
| 1000 | 5 | 4 | **5** | **4.5333** | **0.5074** | 384.5208 | **459.2877** | **427.2798** | **34.8339** |
| | 10 | 4 | 7 | 5.2667 | 0.6397 | 297.7808 | 353.7408 | 320.6774 | 13.5416 |
| | 15 | 4 | 7 | 5.2333 | 0.8172 | 272.4056 | 296.9508 | 284.0453 | 6.6736 |
| | 20 | 3 | 8 | **5.5000** | **1.2247** | 258.9538 | 278.1714 | 267.7404 | **4.5597** |
| | 25 | **4** | 8 | 5.2667 | 0.9803 | **252.1442** | 277.3108 | **260.0149** | 5.0598 |

Tablo 4.11. Şifreli İngilizce metin ve anahtar boyutuna bağlı olarak BCABC sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | **0** | **5** | **1.6000** | **1.7340** | 54.1025 | **86.6142** | 58.6266 | 6.9267 |
| | 10 | **0** | 0 | **0.0000** | **0.0000** | **53.9846** | 57.6774 | **56.2596** | **0.7367** |
| | 15 | **0** | 0 | **0.0000** | **0.0000** | 56.8034 | 60.7787 | 58.0226 | 0.9434 |
| | 20 | **0** | 0 | **0.0000** | **0.0000** | 56.5081 | 62.9036 | 60.0210 | **1.4739** |
| | 25 | **0** | 1 | 0.2667 | 0.4498 | 58.1778 | 64.0077 | **60.9069** | 1.3047 |
| 500 | 5 | 5 | **5** | **5.0000** | **0.0000** | 205.8708 | 205.8708 | **205.8708** | **0.0000** |
| | 10 | 8 | **10** | 9.5333 | 0.6814 | 168.3519 | 205.8708 | 199.2021 | 10.1564 |
| | 15 | 6 | 12 | 9.6667 | 1.6259 | 135.6376 | 186.1697 | 168.8796 | 13.2430 |
| | 20 | **4** | 19 | 12.8333 | **3.7516** | **126.8753** | 205.9255 | **165.6999** | **21.1167** |
| | 25 | 19 | 24 | **21.7667** | 1.5687 | 181.6719 | **207.9439** | 198.6383 | 6.6743 |
| 750 | 5 | **5** | **5** | **5.0000** | **0.0000** | 342.4439 | **342.4439** | 342.4439 | **0.0000** |
| | 10 | 9 | **10** | 9.9333 | 0.2537 | 311.7009 | **342.4439** | **340.4158** | 7.7185 |
| | 15 | 10 | **15** | 13.1000 | **1.3734** | **273.3619** | **342.4439** | **314.6871** | **19.9249** |
| | 20 | 18 | **20** | 19.7667 | 0.5040 | 320.8149 | **342.4439** | 339.4815 | 6.3839 |
| | 25 | 23 | **25** | **24.5667** | 0.7739 | 316.0656 | **342.4439** | 338.9197 | 7.0481 |
| 1000 | 5 | 5 | **5** | **5.0000** | **0.0000** | 472.2756 | **472.2756** | **472.2756** | **0.0000** |
| | 10 | 10 | **10** | 10.0000 | **0.0000** | 472.2756 | **472.2756** | **472.2756** | **0.0000** |
| | 15 | 11 | **15** | 13.5667 | **1.1943** | **397.4719** | **472.2756** | **442.1973** | 23.9817 |
| | 20 | **19** | **20** | 19.9333 | 0.2537 | 456.8319 | **472.2756** | 471.3437 | 3.5673 |
| | 25 | 22 | **25** | **24.2000** | 0.9965 | 425.2619 | **472.2756** | 461.1489 | **14.2026** |

Tablo 4.12. Şifreli Türkçe metin ve anahtar boyutuna bağlı olarak BCABC sonuçları, ADKS: Anahtardan doğru olarak elde edilen karakter sayısı

| Şifreli metin uzunluğu | Anahtar uzunluğu | En düşük ADKS | En Yüksek ADKS | Ortalama ADKS | Std. Sapma ADKS | En düşük Uygunluk | En Yüksek Uygunluk | Ortalama Uygunluk | Uygunluk Std. Sapma |
|---|---|---|---|---|---|---|---|---|---|
| 250 | 5 | 5 | **5** | 5.0000 | **0.0000** | 87.6893 | **87.6893** | **87.6893** | **0.0000** |
| | 10 | **0** | 7 | **2.5667** | 2.6611 | **58.8406** | 81.0350 | **64.5094** | **6.0581** |
| | 15 | **0** | 9 | 3.4667 | 1.9070 | 60.4729 | 75.4298 | 67.5728 | 3.4528 |
| | 20 | 1 | 13 | 7.9000 | **3.1878** | 61.0676 | 83.5661 | 70.3603 | 5.6505 |
| | 25 | 3 | 13 | **7.9333** | 2.4202 | 64.3895 | 79.3419 | 71.1128 | 4.2436 |
| 500 | 5 | 5 | **5** | **5.0000** | **0.0000** | 208.7812 | 208.7812 | **208.7812** | **0.0000** |
| | 10 | **9** | **10** | 9.9333 | 0.2537 | **192.6834** | 208.7812 | 207.7491 | **3.9313** |
| | 15 | 15 | **15** | 15.0000 | **0.0000** | 208.7812 | 208.7812 | 208.7812 | **0.0000** |
| | 20 | 16 | **20** | 19.6333 | **0.8503** | 197.3783 | 208.7812 | **207.1740** | 3.2498 |
| | 25 | 22 | **25** | **23.7333** | 0.6915 | 200.2268 | **209.5443** | 208.5595 | 1.9624 |
| 750 | 5 | 5 | **5** | **5.0000** | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 10 | 10 | **10** | 10.0000 | **0.0000** | 332.4877 | **332.4877** | **332.4877** | **0.0000** |
| | 15 | **14** | **15** | 14.9667 | 0.1826 | **316.1677** | **332.4877** | 331.9437 | **2.9796** |
| | 20 | **19** | **20** | 19.9333 | 0.2537 | 321.7077 | **332.4877** | 331.9257 | 2.2260 |
| | 25 | **24** | **25** | **24.7333** | **0.4498** | 323.7808 | **332.4877** | **331.4456** | 2.3060 |
| 1000 | 5 | 5 | **5** | **5.0000** | **0.0000** | 459.2877 | **459.2877** | **459.2877** | **0.0000** |
| | 10 | 10 | **10** | 10.0000 | **0.0000** | 459.2877 | **459.2877** | **459.2877** | **0.0000** |
| | 15 | **14** | **15** | 14.9000 | 0.3051 | 437.1108 | **459.2877** | 457.3342 | 6.0000 |
| | 20 | **18** | **20** | 19.8333 | **0.4611** | **431.1108** | **459.2877** | **456.7755** | **6.8726** |
| | 25 | **24** | **25** | **24.8000** | 0.4068 | 444.9677 | **459.2877** | 457.7653 | 3.6525 |

### 4.2.3. Deneysel Çalışma 3: Algoritmaların Karşılaştırılması

Çalışmanın üçüncü kısmında Vigenére şifrelemenin kriptoanalizi üzerinde GA, PSO, DE, ABC ve BCABC algoritmalarının performansları kıyaslanarak daha uygun olan algoritmanın belirlenmesi amaçlanmıştır. Daha önceki deneylerde olduğu gibi kıyaslama çalışmalarında da anahtar uzunluğu olarak 5, 10, 15, 20 ve 25 ve her biri Vigenére ile şifrelenmiş, İngilizce ve Türkçe 250, 500, 750 ve 1000 karakter uzunluğunda dört farklı şifreli metin kullanılmıştır. Algoritmalar her bir durum için rasgele örnekleme ile başlamak suretiyle 30 defa çalıştırılmıştır.

Birinci deneyden elde edilen en iyi parametre değerlerine bağlı olarak, bahsedilen algoritmaların her biri için anahtarın doğru olarak kurtarılan en yüksek, en düşük ve ortalama karakter sayısı, anahtarın doğru olarak kurtarılan karakter sayısının standart sapması, en düşük, en yüksek ve ortalama uygunluk ile birlikte uygunluk değerinin standart sapması hesaplanmıştır.

Şifreli İngilizce metinler için, orijinal anahtardan elde edilen doğru karakter sayasını içeren sonuçlar Tablo 4.13'te, uygunluk değerlerine bağlı sonuçlar ise Tablo 4.14'te verilmiştir.

Öncelikle Tablo 4.13'e bakılacak olursa, doğru anahtara dayalı karakter sayısında en iyi değerler karşılaştırılması durumunda görülüyor ki şifreli metin 250 karakter uzunluğunda iken en iyi sonuçları GA ve DE algoritmaları verirken, metin uzunluğu 500, 750 ve 1000 karakter olduğunda ise GA, DE ve BCABC algoritmalarının sonuçları anahtar uzunluğuna yakınsamaktadır. Bu arada PSO algoritmasının performansı metin uzunluğu ile artsa da kıyaslandığında diğerlerinden biraz geride kalmaktadır. Ortalama değerler incelendiğinde ise durumda benzerlik vardır. Anahtar uzunluğu arttıkça genelde başarılı bir şekilde elde edilen karakter sayısı artış göstermektedir. Fakat burada ortalama değerlerin DE ve BCABC algoritmalarında daha istikrarlı ve başarılı sonuçlar verdiğini söyleyebiliriz. Bunların ardından en iyi ortalama ise GA algoritmasında elde edildiğini görmekteyiz. PSO ve ABC algoritmalarının sonuçları ise diğerlerinden ortalamada bir hayli geride kaldığını da ifade edebiliriz.

Ardından Tablo 4.14'e bakıldığında, algoritmaların uygulanması sonucunda elde edilen uygunluk değerlerini görmekteyiz. Bunlardaki en iyi değerler karşılaştırılması durumunda ise, 250 karakterli şifreli metinde GA, PSO, DE ve BCABC algoritmalarının benzer netice verdiğini görüyoruz. Burada ABC'nin daha düşük uygunluk değerlerine sahip olmuştur. 500 karakter olunca, GA, DE ve BCABC algoritmaları benzer davrandığını, PSO ve ABC'nin daha düşük değerler verdiğini görüyoruz. Bu durumun yine 750 ve 1000 karakterli şifreli metinlerde ise devam ettiğini görmekteyiz. Dahası, burada bu değerlerin her bir uzunluktaki anahtar için GA, DE ve BCABC algoritmalarında sabit kaldığını görmekteyiz: 750 karakter için 342.4439, 1000 karakter için 472.2756 değerleri elde edilmiştir. Bunlarda ise en düşük uygunluk değerleri ABC algoritmasında uygulamasında oluşmuştur.

Neticede anahtarın kısa boyutlu olması durumunda Vigenére şifreli metinler için PSO ve ABC algoritmalarının yeterince etkili olduğu sonucuna varılabilir. Daha uzun anahtar için (≥10), bu yöntemlerin yetersiz kaldığı görülmektedir. Anahtardaki karakter sayısı 15 veya daha az olduğunda, 250'den fazla karaktere sahip şifreli metinlerde DE, GA ve BCABC algoritmalarının karakterlerin büyük çoğunluğunu çözebildiği görülmektedir. Burada, şifreli metindeki karakter sayısı 250 olduğunda da, DE ve GA algoritmasının yine PSO, ABC ve BCABC'den daha yüksek performans gösterdiği de söylenebilir.

Deney esnasında, özellikle anahtar boyutu 25'ten az ve şifreli metindeki karakter sayısı 250'den fazla olduğunda ise DE ve BCABC'nin GA'dan daha etkili olduğu gözlemlenmiştir. Bazı durumlarda, anahtarın karakterlerinin tamamı doğru olarak çözülürken ve uygunluk değeri elde edilirken, sonuç değerlerin ortalamaları ve standart sapmaları yalnızca istatistiksel analiz sayesinde elde edilebilmiştir.

Şifreli Türkçe metinler için, orijinal anahtardan elde edilen doğru karakter sayısını içeren sonuçları Tablo 4.15'te, uygunluk değerlerine bağlı sonuçları ise Tablo 4.16'da verilmiştir.

Tablo 4.15'e bakılacak olursa, doğru anahtara dayalı karakter sayısında en iyi değerler karşılaştırılması durumunda görülüyor ki şifreli metin 250 karakter uzunluğunda iken en iyi sonuçları yine GA ve DE algoritmaları verirken, metin uzunluğu 500, 750 ve 1000 karakter olduğunda ise GA, DE ve BCABC algoritmalarının sonuçları anahtar uzunluğu ile çok yakın veya eşit değerler vermiştir. Burada PSO algoritmasının performansı metin

uzunluğu ile biraz artsa da, diğer algoritmaların performansının gerisinde kaldığını görmekteyiz. Ayrıca ABC algoritmasının metin uzunluğu nispetinde artamadığını ve oldukça düşük performans gösterdiğini söyleyebiliriz. Ortalama değerler incelendiğinde ise, benzer şekilde metin uzunluğu ile performans artışının genel olarak doğru orantılı olarak artış gösterdiğini görmekteyiz. Burada ortalama değerlerin DE ve BCABC algoritmalarında daha istikrarlı ve başarılı sonuçlar verdiğini söyleyebiliriz. Bu ikisine en yakın performans ise GA algoritmasında elde edilmiştir. PSO ve ABC algoritmalarının sonuçları ise anahtar uzunlukları ve şifreli metin uzunlukları küçük artışlar göstermesine rağmen, diğer metotların ortalamada oldukça geride kaldığını görmekteyiz.

Tablo 4.16'da ise, algoritmaların uygulanması sonucunda elde edilen uygunluk değerlerini görmekteyiz. Bunlardaki en iyi değerler karşılaştırıldığında, 250 karakterli şifreli metinde GA, DE ve BCABC algoritmalarının birbirlerine yakın netice verdiğini görüyoruz. Bunların ardından PSO algoritmasının uygunluk değerleri anahtar uzunluğu arttıkça biraz daha hızlı azaldığını görüyoruz. En düşük uygunluk değerlerinin işe ABC algoritmasında elde edildiğini görüyoruz. 500 karakter olunca, GA, DE ve BCABC algoritmaları birbirlerine yakın davrandığını, PSO ve ABC'nin daha düşük değerler verdiğini görüyoruz. Bu uzunlukta en düşük değerler ABC algoritmasında elde edilmiştir. Bu durumun yine 750 ve 1000 karakterli şifreli metinlerde ise devam ettiğini görmekteyiz. Baktığımız zaman, bu değerlerin her bir uzunluktaki anahtar için GA, DE ve BCABC algoritmalarında sabit kaldığını görmekteyiz: 750 karakter için 332.4877, 1000 karakter için 459.2877 değerleri elde edilmiştir. Bunlardan sonra en düşük uygunluk değerleri PSO algoritmasında, genelde en düşük uygunluk değerleri ise ABC algoritması uygulanmasında elde edilmiştir. Ortalama uygunluk değerleri de benzer sonuçlar ortaya koymaktadır. Ortalamada 250 karakterli şifreli metinde GA, DE ve BCABC algoritmalarının en yüksek uygunluk değerlerini verdiğini, daha sonra PSO'nun değerlerinin biraz daha düşük ve ABC'nin değerlerinin ise en düşük olduğunu görüyoruz. 500 karakterli metinde ise, en yüksek ortalama değerler DE ve BCABC 'de, bunlardan biraz daha düşük olan GA'nın değerleri de yakın seviyededir. PSO ve ABC'nin yine daha düşük değerler verdiğini, bunlardan ABC'nin değerlerinin anahtar uzunluğu arttıkça daha hızlı azaldığını görmekteyiz. Yine 750 ve 1000 karakterli metinlerde, GA, DE ve BCABC algoritmaları için ortalama uygunluk değerlerinin

anahtar uzunluğundan bağımsız hale geldiğini görüyoruz. Burada da yine ABC'nin en düşük ortalama uygunluk değerlerini verdiğini görmekteyiz.

Neticede daha kısa anahtarlar ile çözülen Vigenére şifreli metinler için PSO ve ABC algoritmaların yeterince etkili olduğu sonucuna varılabilir. Daha uzun anahtar için (≥10), bu yöntemlerin iyi sonuç vermediği görülmektedir. Anahtardaki karakter sayısı 15 veya daha az olduğunda, 250'den fazla karaktere sahip şifreli metinlerde DE, GA ve BCABC algoritmalarının karakterlerin büyük çoğunluğunu çözebildiği görülmektedir. Burada, şifreli metindeki karakter sayısı 250 olduğunda da, GA ve DE algoritmasının yine PSO, ABC ve BCABC'den daha yüksek performans gösterdiği de söylenebilir. Deney esnasında, özellikle anahtar boyutu 25'ten az ve şifreli metindeki karakter sayısı 250'den fazla olduğunda ise DE ve BCABC'nin GA'dan daha etkili olduğu gözlemlenmiştir. Bazı durumlarda, anahtarın karakterlerinin tamamı doğru olarak çözülürken ve uygunluk değeri elde edilirken, sonuç değerlerin ortalamaları ve standart sapmaları yalnızca istatistiksel analiz sayesinde elde edilebilmiştir.

İngilizce ve Türkçe de metin uzunluğu artığında gizli anahtarı bulma potansiyeli de orantılı olarak artmaktadır ve bu işleme paralel olarak iterasyon sayasını da aza düşürmektedir. Türkçe metinlerde doğru anahtar bulma oranı İngilizce metinlere göre daha yüksektir, şifrelenmiş metin uzunluğu çok kısa olduğunda bile (≤ 250 karakter). Kısa kelimeler daha kolay bulunabilir olduğundan dolayı, Türkçe dilindeki kısa kelimelerin (3-8 harfli) Türkçe metinlerde %60 oranında kullanılıyor oluşu gösterilmiştir. Dolayısıyla Türkçe metnin kriptoanalizi İngilizce metnine göre nispeten daha kolaydır [35]. Çaprazlama operatörü bu tür problemler için çok yararlı olduğundan dolayı, Türkçe ve İngilizce metinlerde DE, GA ve BCABC algoritmaları ABC ve PSO algoritmalarına göre daha iyi sonuçlar ürettiği görülmüştür.

Tablo 4.13. İngilizce metinler için bulunan doğru anahtar karakterine dayalı karşılaştırma

| Şifreli metin boyutu | Anahtar boyutu | En iyi değer | | | | | Ortalama değer | | | | | Standart sapma | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC |
| 250 | 5 | **5** | **5** | **5** | 4 | **5** | 3.2333 | 1.3667 | **4.6667** | 2.5333 | 1.6000 | 1.8323 | **1.5862** | 0.9589 | 2.5333 | 1.7340 |
| | 10 | 6 | 4 | **8** | 0 | 0 | 0.5333 | 0.5333 | **4.1000** | 0.0000 | 0.0000 | 1.5916 | 1.2521 | 3.4276 | **0.0000** | **0.0000** |
| | 15 | **11** | 4 | 10 | 1 | 0 | **0.9333** | 0.2333 | 2.0333 | 0.1000 | 0.0000 | 2.4059 | 0.7739 | 3.2851 | 0.1000 | **0.0000** |
| | 20 | 7 | 5 | **9** | 1 | 0 | 0.3333 | 0.6333 | **1.3667** | 0.0333 | 0.0000 | 1.2954 | 1.2452 | 2.4563 | 0.0333 | **0.0000** |
| | 25 | 3 | **5** | **5** | 3 | 1 | 0.5000 | 0.5667 | **1.2667** | 0.3667 | 0.2667 | 0.9738 | 1.1043 | 1.6386 | 0.3667 | 0.4498 |
| 500 | 5 | **5** | **5** | **5** | **5** | **5** | 4.9667 | 3.3000 | **5.0000** | 3.9667 | **5.0000** | 0.1826 | 1.0875 | **0.0000** | 3.9667 | **0.0000** |
| | 10 | **10** | 7 | **10** | 5 | **10** | 9.4333 | 5.1000 | **9.9333** | 4.2667 | 9.5333 | 0.6261 | 1.1552 | **0.2537** | 4.2667 | 0.6814 |
| | 15 | **15** | 11 | **15** | 5 | 12 | 13.1000 | 6.7333 | **14.9333** | 3.5333 | 9.6667 | 1.6049 | 2.0667 | **0.2537** | 3.5333 | 1.6259 |
| | 20 | **20** | 11 | 19 | 6 | 19 | 14.7333 | 6.3000 | **18.9667** | 4.2667 | 12.8333 | 2.0833 | 2.2614 | **0.1826** | 4.2667 | 3.7516 |
| | 25 | 23 | 13 | **24** | 9 | **24** | 16.9000 | 7.6333 | **23.9000** | 4.0667 | 21.7667 | 2.8929 | 3.0113 | **0.3051** | 4.0667 | 1.5687 |
| 750 | **5** | **5** | **5** | **5** | **5** | **5** | **5.0000** | 3.4333 | **5.0000** | 4.0333 | **5.0000** | **0.0000** | 0.8172 | **0.0000** | 4.0333 | **0.0000** |
| | **10** | **10** | **9** | **10** | 7 | **10** | 9.6667 | 5.5667 | **10.0000** | 4.9667 | 9.9333 | 0.5467 | 1.4547 | **0.0000** | 4.9667 | 0.2537 |
| | **15** | **15** | 12 | **15** | 6 | **15** | 13.4333 | 7.4000 | **15.0000** | 4.4333 | 13.1000 | 1.5013 | 2.0443 | **0.0000** | 4.4333 | 1.3734 |
| | **20** | **20** | 11 | **20** | 7 | **20** | 16.9333 | 7.6667 | **19.9667** | 4.7000 | 19.7667 | 1.5960 | 1.4933 | **0.1826** | 4.7000 | 0.5040 |
| | **25** | **25** | 12 | **25** | 7 | **25** | 20.4000 | 8.4333 | **24.9000** | 4.7333 | 24.5667 | 2.4439 | 2.4023 | **0.3051** | 4.7333 | 0.7739 |
| 1000 | **5** | **5** | **5** | **5** | **5** | **5** | 4.9667 | 3.5333 | **5.0000** | 4.1333 | **5.0000** | 0.1826 | 0.9732 | **0.0000** | 4.1333 | **0.0000** |
| | **10** | **10** | 8 | **10** | 6 | **10** | 9.7667 | 5.7333 | **10.0000** | 5.1000 | **10.0000** | 0.4302 | 1.1427 | **0.0000** | 5.1000 | **0.0000** |
| | **15** | **15** | 10 | **15** | 7 | **15** | 13.4667 | 7.1333 | **14.9667** | 4.6667 | 13.5667 | 1.4320 | 1.3578 | **0.1826** | 4.6667 | 1.1943 |
| | **20** | **20** | 11 | **20** | 8 | **20** | 17.6667 | 8.0000 | **19.9000** | 5.0333 | 19.9333 | 1.5610 | 1.9652 | 0.4026 | 5.0333 | **0.2537** |
| | **25** | **25** | 18 | **25** | 7 | **25** | 21.2000 | 9.9333 | **24.8667** | 5.5667 | 24.2000 | 2.0578 | 2.6901 | **0.3457** | 5.5667 | 0.9965 |

Tablo 4.14. İngilizce karakterler için uygunluk değerine dayalı karşılaştırma

| Şifreli metin boyutu | Anahtar boyutu | En iyi değer | | | | | Ortalama değer | | | | | Standart sapma | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC |
| 250 | 5 | **86.6142** | **86.6142** | **86.6142** | 64.0267 | **86.6142** | 69.0764 | 59.6482 | 82.6007 | 64.0267 | **86.6142** | 11.7235 | 7.5217 | 8.2841 | **2.7069** | 6.9267 |
| | 10 | 60.1863 | 58.8240 | **72.3724** | 53.3889 | 57.6774 | 56.1366 | 54.5818 | 62.1395 | 53.3889 | **57.6774** | 1.4144 | 1.9792 | 5.5459 | **0.7267** | 0.7367 |
| | 15 | **74.0957** | 57.8687 | 66.1111 | 53.0608 | 60.7787 | 58.6584 | 54.2724 | 58.1354 | 53.0608 | **60.7787** | 4.1922 | 1.4838 | 2.3933 | **0.8027** | 0.9434 |
| | 20 | **68.8373** | 59.6172 | 61.3401 | 50.7284 | 62.9036 | 59.0084 | 55.8124 | 56.9359 | 50.7284 | **62.9036** | 2.3885 | 1.7471 | 1.4221 | **0.4811** | 1.4739 |
| | 25 | **69.6755** | 60.2802 | 59.7120 | 51.7278 | 64.0077 | 62.4770 | 56.1749 | 57.8161 | 51.7278 | **64.0077** | 2.1533 | 2.2291 | **0.8186** | 0.9141 | 1.3047 |
| 500 | 5 | **205.8708** | **205.8708** | **205.8708** | **205.8708** | **205.8708** | 204.5317 | 156.8410 | 205.8708 | **205.8708** | **205.8708** | 7.3345 | 24.6460 | **0.0000** | 12.4580 | **0.0000** |
| | 10 | **205.8708** | 172.6860 | **205.8708** | 151.6807 | **205.8708** | 196.8990 | 149.5647 | 204.8769 | 151.6807 | **205.8708** | 10.7718 | 11.8037 | **3.7824** | 5.3378 | 10.1564 |
| | 15 | **205.8708** | 173.2608 | **205.8708** | 129.3112 | 186.1697 | **192.0746** | 142.1190 | 205.5551 | 129.3112 | 186.1697 | 11.7197 | 10.6438 | **1.5498** | 3.2605 | 13.2430 |
| | 20 | **205.9255** | 151.7895 | **205.9255** | 118.0029 | **205.9255** | 182.0374 | 135.2078 | **205.5551** | 118.0029 | **205.9255** | 11.0484 | 8.6860 | **1.8748** | 1.9824 | 21.1167 |
| | 25 | 199.4555 | 153.8444 | **207.9439** | 127.9195 | **207.9439** | 180.4832 | 136.1634 | **207.7855** | 127.9195 | **207.9439** | 9.7627 | 8.6591 | **0.4832** | 3.6811 | 6.6743 |
| 750 | 5 | **342.4439** | **342.4439** | **342.4439** | **342.4439** | **342.4439** | **342.4439** | **259.0414** | **342.4439** | **342.4439** | **342.4439** | **0.0000** | 33.7318 | **0.0000** | 17.2920 | **0.0000** |
| | 10 | **342.4439** | 308.4972 | **342.4439** | 259.3519 | **342.4439** | 334.4502 | 251.1021 | **342.4439** | 259.3519 | **342.4439** | 13.0033 | 20.7271 | **0.0000** | 8.5797 | 7.7185 |
| | 15 | **342.4439** | 281.3719 | **342.4439** | 225.1796 | **342.4439** | 316.0985 | 239.5816 | **342.4439** | 225.1796 | **342.4439** | 23.3129 | 21.0357 | **0.0000** | 6.7211 | 19.9249 |
| | 20 | **342.4439** | 255.8072 | **342.4439** | 202.4107 | **342.4439** | 310.2493 | 223.6455 | 342.2972 | 202.4107 | **342.4439** | 14.6019 | 11.8842 | **0.8037** | 4.2668 | 6.3839 |
| | 25 | **342.4439** | 241.7219 | **342.4439** | 189.7726 | **342.4439** | 309.6918 | 216.9870 | 341.9660 | 189.7726 | **342.4439** | 16.8850 | 11.6850 | **1.6042** | 3.3355 | 7.0481 |
| 1000 | 5 | **472.2756** | **472.2756** | **472.2756** | **472.2756** | **472.2756** | 469.7205 | 369.1274 | **472.2756** | **472.2756** | **472.2756** | 13.9950 | 48.8020 | **0.0000** | 28.6436 | **0.0000** |
| | 10 | **472.2756** | 399.2419 | **472.2756** | 359.7419 | **472.2756** | 463.2532 | 344.0369 | **472.2756** | 359.7419 | **472.2756** | 16.9758 | 23.0306 | **0.0000** | 14.3074 | **0.0000** |
| | 15 | **472.2756** | 363.0519 | **472.2756** | 306.3119 | **472.2756** | 438.6265 | 323.3344 | 471.5946 | 306.3119 | 472.2756 | 28.4480 | 20.7343 | **3.7299** | 8.0079 | 23.9817 |
| | 20 | **472.2756** | 346.6519 | **472.2756** | 282.1519 | **472.2756** | 436.6399 | 313.2915 | 470.6138 | 282.1519 | 472.2756 | 21.7241 | 19.0091 | 6.6081 | 7.0368 | **3.5673** |
| | 25 | **472.2756** | 381.0719 | **472.2756** | **272.2407** | **472.2756** | 429.0244 | 310.6931 | 470.4147 | 272.2407 | 472.2756 | 23.2646 | 21.4984 | **4.9309** | 5.3756 | 14.2026 |

Tablo 4.15.Türkçe metinler için bulunan doğru anahtar karakterine dayalı karşılaştırma

| Şifreli metin boyutu | Anahtar boyutu | En iyi değer | | | | | Ortalama değer | | | | | Standart sapma | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC |
| 250 | 5 | **5** | **5** | **5** | **5** | **5** | 4.6333 | 3.3333 | **5.0000** | 3.9667 | **5.0000** | 0.7184 | 0.9223 | **0.0000** | 0.4138 | **0.0000** |
| | 10 | **10** | 7 | **10** | 5 | 7 | 7.7667 | 3.6667 | **9.7667** | 2.2000 | 2.5667 | 2.3295 | 1.4223 | **0.5683** | 1.4479 | 2.6611 |
| | 15 | 13 | 8 | **15** | 5 | 9 | 7.8000 | 2.8667 | **11.0000** | 1.5667 | 3.4667 | 3.2632 | 2.1772 | 2.4635 | **1.6333** | 1.9070 |
| | 20 | 15 | 8 | **16** | 5 | 13 | 6.9000 | 3.1333 | **9.6000** | 2.4667 | 7.9000 | 3.5365 | 2.3004 | 2.6987 | **1.4077** | 3.1878 |
| | 25 | **15** | 9 | 12 | 5 | 13 | 8.0667 | 3.2333 | **8.1000** | 1.6000 | 7.9333 | 3.7040 | 1.7750 | 2.2491 | **1.5222** | 2.4202 |
| 500 | 5 | **5** | **5** | **5** | **5** | **5** | **5.0000** | 3.5667 | **5.0000** | 4.4667 | **5.0000** | **0.0000** | 0.6261 | **0.0000** | 0.5074 | **0.0000** |
| | 10 | **10** | 7 | **10** | 6 | **10** | 9.6333 | 5.1333 | **10.0000** | 5.0667 | 9.9333 | 0.5561 | 0.9732 | **0.0000** | 0.5833 | 0.2537 |
| | 15 | **15** | 9 | **15** | 7 | **15** | 13.9333 | 6.0000 | **15.0000** | 5.1667 | **15.0000** | 1.1427 | 1.2034 | **0.0000** | 0.7915 | **0.0000** |
| | 20 | 19 | 10 | **20** | 7 | **20** | 15.7000 | 6.4333 | **19.9667** | 4.6000 | 19.6333 | 2.4233 | 1.4782 | **0.1826** | 1.1626 | 0.8503 |
| | 25 | 24 | 12 | 24 | 7 | **25** | 18.9667 | 7.2000 | **24.0000** | 4.4667 | 23.7333 | 2.8945 | 2.1560 | **0.0000** | 1.4320 | 0.6915 |
| 750 | 5 | **5** | **5** | **5** | **5** | **5** | **5.0000** | 3.5333 | **5.0000** | 4.4333 | **5.0000** | 0.0000 | 0.6814 | **0.0000** | 0.5040 | **0.0000** |
| | 10 | **10** | 7 | **10** | 6 | **10** | 9.7333 | 5.2333 | **10.0000** | 5.1000 | **10.0000** | 0.5208 | 1.1943 | **0.0000** | 0.6074 | **0.0000** |
| | 15 | **15** | 8 | **15** | 7 | **15** | 13.9000 | 5.9333 | **15.0000** | 5.5333 | 14.9667 | 1.0619 | 1.3113 | **0.0000** | 0.8193 | 0.1826 |
| | 20 | **20** | 12 | **20** | 8 | **20** | 17.2667 | 7.3667 | **19.9333** | 5.2333 | **19.9333** | 1.8182 | 1.7317 | **0.2537** | 1.2229 | **0.2537** |
| | 25 | 23 | 12 | **25** | 8 | **25** | 19.1333 | 8.4333 | **24.9333** | 4.6667 | 24.7333 | 2.5695 | 2.0625 | **0.3651** | 1.0933 | 0.4498 |
| 1000 | 5 | **5** | **5** | **5** | **5** | **5** | 4.9667 | 3.8333 | 4.9667 | 4.5333 | **5.0000** | 0.1826 | 0.7915 | 0.1826 | 0.5074 | **0.0000** |
| | 10 | **10** | 8 | **10** | 7 | **10** | 9.6333 | 5.0667 | **10.0000** | 5.2667 | **10.0000** | 0.6149 | 1.0483 | **0.0000** | 0.6397 | **0.0000** |
| | 15 | **15** | 10 | **15** | 7 | **15** | 13.9000 | 6.5333 | **15.0000** | 5.2333 | 14.9000 | 1.1552 | 1.6344 | **0.0000** | 0.8172 | 0.3051 |
| | 20 | **20** | 10 | **20** | 8 | **20** | 17.2667 | 7.5333 | **20.0000** | 5.5000 | 19.8333 | 1.8182 | 1.2521 | **0.0000** | 1.2247 | 0.4611 |
| | 25 | **25** | 13 | **25** | 8 | **25** | 21.0667 | 8.5667 | **24.9333** | 5.2667 | 24.8000 | 2.0998 | 2.2234 | **0.2537** | 0.9803 | 0.4068 |

Tablo 4.16. Türkçe karakterler için uygunluk değerine dayalı karşılaştırma

| Şifreli metin boyutu | Anahtar boyutu | En iyi değer | | | | | Ortalama değer | | | | | Standart sapma | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC | GA | PSO | DE | ABC | BCABC |
| 250 | 5 | **87.6893** | **87.6893** | **87.6893** | **87.6893** | **87.6893** | 83.9274 | 71.8616 | **87.6893** | 75.4893 | **87.6893** | 7.0960 | 7.3600 | **0.0000** | 4.7468 | **0.0000** |
| | 10 | **87.6893** | 74.6070 | **87.6893** | 62.8773 | 81.0350 | **80.3693** | 64.9737 | 87.4076 | 57.0546 | 64.5094 | 7.5764 | 4.8416 | **1.0862** | 2.0074 | 6.0581 |
| | 15 | 87.1034 | 70.0893 | **88.4361** | 59.3545 | 75.4298 | 77.4610 | 62.6000 | **86.3848** | 53.6795 | 67.5728 | 5.2800 | 3.6686 | 2.9394 | **1.6610** | 3.4528 |
| | 20 | **88.3858** | 70.7564 | 86.8992 | 55.7378 | 83.5661 | 73.3611 | 62.4331 | **74.5462** | 52.9640 | 70.3603 | 5.5266 | 3.5329 | 4.7858 | **1.1948** | 5.6505 |
| | 25 | **89.5371** | 71.5053 | 78.6273 | 56.7146 | 79.3419 | **78.3575** | 65.0669 | 70.4414 | 51.7996 | 71.1128 | 5.8534 | 3.3676 | 2.9952 | **1.4357** | 4.2436 |
| 500 | 5 | **208.7812** | **208.7812** | **208.7812** | **208.7812** | **208.7812** | **208.7812** | 168.7199 | **208.7812** | 194.2237 | **208.7812** | **0.0000** | 15.0480 | **0.0000** | 14.1103 | **0.0000** |
| | 10 | **208.7812** | 168.4371 | **208.7812** | 159.5398 | **208.7812** | 203.5036 | 154.5469 | **208.7812** | 146.1941 | 207.7491 | 8.1217 | 8.0682 | **0.0000** | 5.4336 | 3.9313 |
| | 15 | **208.7812** | 157.5246 | **208.7812** | 136.7473 | **208.7812** | 199.9356 | 143.7117 | **208.7812** | 127.1416 | **208.7812** | 9.4435 | 7.5786 | **0.0000** | 3.4820 | **0.0000** |
| | 20 | 204.7668 | 153.1097 | **208.7812** | 129.8581 | **208.7812** | 190.9227 | 143.3478 | **208.6450** | 123.3861 | 207.1740 | 8.0900 | 5.6208 | **0.7463** | 2.9602 | 3.2498 |
| | 25 | 204.9300 | 154.6983 | **209.5443** | 122.3029 | **209.5443** | 190.3536 | 143.3216 | **209.5443** | 118.7351 | 208.5595 | 9.2062 | 5.8300 | **0.0000** | 2.1761 | 1.9624 |
| 750 | 5 | **332.4877** | **332.4877** | **332.4877** | **332.4877** | **332.4877** | **332.4877** | 265.9413 | **332.4877** | 305.5313 | **332.4877** | 0.0000 | 24.9519 | **0.0000** | 24.0569 | **0.0000** |
| | 10 | **332.4877** | 270.7537 | **332.4877** | 250.6668 | **332.4877** | 326.5140 | 242.3452 | **332.4877** | 231.1823 | **332.4877** | 11.6262 | 15.0759 | **0.0000** | 8.3609 | **0.0000** |
| | 15 | **332.4877** | 252.7488 | **332.4877** | 217.4108 | **332.4877** | 317.5800 | 230.6005 | **332.4877** | 204.5360 | 331.9437 | 13.8280 | 11.4627 | **0.0000** | 5.2829 | 2.9796 |
| | 20 | **332.4877** | 256.2708 | **332.4877** | 217.3559 | **332.4877** | 309.0320 | 224.3153 | **331.7050** | 196.5783 | **331.9257** | 14.1485 | 12.0862 | 2.9814 | 5.7820 | **2.2260** |
| | 25 | 328.1708 | 247.2752 | **332.4877** | 194.2468 | **332.4877** | 298.0012 | 226.6884 | **331.9998** | 187.5224 | **331.4456** | 16.4095 | 10.1333 | 2.6723 | 2.5900 | **2.3060** |
| 1000 | 5 | **459.2877** | **459.2877** | **459.2877** | **459.2877** | **459.2877** | 456.9937 | 381.9259 | 457.0958 | 427.2798 | **459.2877** | 12.5648 | 45.8663 | 12.0055 | 34.8339 | **0.0000** |
| | 10 | **459.2877** | 388.0708 | **459.2877** | 353.7408 | **459.2877** | 447.5076 | 331.2852 | **459.2877** | 320.6774 | **459.2877** | 19.2021 | 18.7291 | **0.0000** | 13.5416 | **0.0000** |
| | 15 | **459.2877** | 358.2808 | **459.2877** | 296.9508 | **459.2877** | 437.1538 | 316.5989 | **459.2877** | 284.0453 | 457.3342 | 23.5861 | 18.3243 | **0.0000** | 6.6736 | 6.0000 |
| | 20 | **459.2877** | 330.9668 | **459.2877** | 278.1714 | **459.2877** | 422.4303 | 306.8192 | **459.2877** | 267.7404 | 456.7755 | 20.9674 | 13.1354 | **0.0000** | 4.5597 | 6.8726 |
| | 25 | **459.2877** | 331.4208 | **459.2877** | 277.3108 | **459.2877** | 421.7398 | 303.2669 | **458.2966** | 260.0149 | 457.7653 | 21.3322 | 13.9922 | 3.7719 | 5.0598 | **3.6525** |

### 4.2.4. İstatistiksel Analiz

250, 500, 750, 1000 karakter boyutlarında Türkçe ve İngilizce şifreli metinler ile 5, 10, 15, 20 ve 25 karakterli anahtar kullanılması durumları çalışılmıştır. İncelenen 20 ayrı durum olduğundan ve her bir durum 30 kez koşulduğundan her algoritma 20 × 30 = 600 kez çalıştırılmıştır.

Karşılaştırmalar hipotez testi elde etmemizi sağlayan araçlar kullanılarak gerçekleştirilmiştir. 30 defa çalıştırma sonucu elde edilen uygunluk değerlerinin ortalaması normallik testlerinde başarısız sonuçlar verdiğinden, algoritmalar Mann Whitney U testi uygulaması kullanılmıştır.

Bu çalışmada elde edilen 30 koşma sonucunun normal dağılımlı bir yapıya sahip olmamasından dolayı istatistiksel kıyaslamada Mann Whitney U testi uygulanır, bu test sonucu p ve h değerleri ile gösterilir, P<0,05 ise anlamlı bir fark vardır, p>0,05 ise anlamlı bir fark yoktur şeklinde değerlendirilir, h değeri 0 olduğu zaman test sonuçları arasındaki pozitif ve negatif farklar toplamı birbirine eşittir. H değeri 1 olduğunda ise test sonuçları birbirine eşit değildir. Negatif farkların toplamı pozitif farkların toplamına göre çok küçüktür ya da büyüktür. Mann Whitney U testi ile yapılan istatistiksel analiz sonuçları Tablo 4.17 ve Tablo 4.18'de verilmektedir.

Burada İngilizce metinler için 20 × 4= 80 ve Türkçe metinler için yine 20 × 4= 80, yani toplamda 160 durum oluşmuştur.

İngilizce ve Türkçe metinler üzerinde yapılan deneylerde, BCABC algoritması kontrol algoritması olarak seçilmiş ve toplamda 160 durumun ortalama değerleri kullanılarak GA, PSO, DE ile ABC algoritmaları ile karşılaştırılmıştır ve sonuçlar sırasıyla Tablo 4.17 ve Tablo 4.18'te verilmiştir.

Tablo 4.17'ye bakıldığında İngilizce metin için, BCABC ve GA algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 15 karakterli anahtarda 0.53951, en küçük değer ise 0.0000375 kadar olmuştur. 500 karakterli metinde bu değer en fazla 10 karakterli anahtarda 0.36427 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 15 karakterli anahtarda 0.60815, 1000 karakterli metinde ise en büyük p değeri 15 karakterli anahtarda 0.63222 olmuştur.

H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 8'inde BCABC algoritmasının daha iyi sonuç gösterdiği, 8'inde eşit ve 4'ünde ise GA algoritmasının daha iyi olduğu sonucuna varılmıştır.

BCABC ve PSO algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 15 karakterli anahtarda 8.891e-10, en küçük değer ise 0.00085257 kadar olmuştur. 500 karakterli metinde bu değer en fazla 25 karakterli anahtarda 2.9972e-11 olmuştur, en küçük değer ise 1.0081e-08 kadar olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 25 karakterli anahtarda 9.3936e-12, en küçük değer ise 1.5524e-11 kadar olmuştur. 1000 karakterli metinde ise en büyük p değeri 5 karakterli anahtarda 5.5768e-11 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 19'unda BCABC algoritmasının daha iyi sonuç gösterdiği, 1'inde eşit sonucuna varılmıştır.

BCABC ve ABC algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 15, 20 ve 25 karakterli anahtarlarda 3.0199e-11, en küçük değer ise 0.42457 kadar olmuştur. 500 karakterli metinde bu değer en fazla 15 ve 20 karakterli anahtarlar 3.0104e-11 olmuştur, en küçük değer ise 10 karakterli anahtarda 1.4411e-11 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 25 karakterli anahtarda 9.3936e-12, en küçük değer ise 5 karakterli anahtarda 1.2428e-11 olmuştur. 1000 karakterli metinde ise en büyük p değeri 15 karakterli anahtarda 2.8628e-11 olmuştur, en küçük değer ise 10 karakterli anahtarda 1.2118e-12 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 19'unda BCABC algoritmasının daha iyi sonuç gösterdiği, 1'inde eşit çıkardığı sonucuna varılmıştır.

BCABC ve DE algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 10 karakterli anahtarda 0. 7.2438e-07, en küçük değer ise 0.083666 kadar olmuştur. 500 karakterli metinde bu değer en fazla 15 karakterli anahtarda 2.3565e-12 olmuştur, en küçük değer ise 10 karakterli anahtarda 0.0057909 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 15 karakterli anahtarda 6.2216e-10, en küçük değer ise 10 karakterli anahtarda 0.1608 olmuştur. 1000 karakterli metinde ise en büyük p değeri 15 karakterli anahtarda 4.598e-08 olmuştur, en küçük değer ise 25 karakterli anahtarda 0.0017171olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 8'inde BCABC algoritmasının daha iyi

sonuç gösterdiği, 7'sinde eşit ve 5'inde ise DE algoritmasının daha iyi çıkardığı sonucuna varılmıştır.

Tablo 4.17'de yer alan 80 durumum değerlerinden elde edilen p-değerleri ve h-değerleri karşılaştırıldığında, deneylerdeki 80 durumun 54'ünde BCABC algoritmasını daha iyi sonuç, 17'sinde eşit ve 9'unda ise daha kötü sonuç ortaya çıkardığı sonucuna varılmıştır.

Türkçe metinler üzerinde yapılan deneylerde, BCABC algoritması kontrol algoritması olarak seçilmiş ve toplamda 120 durumun ortalama değerleri kullanılarak GA, PSO, DE ile ABC algoritmaları ile karşılaştırılmıştır ve sonuçlar Tablo 5.18'de verilmiştir.

Tablo 4.18'e bakıldığında Türkçe metin için, BCABC ve GA algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 25 karakterli anahtarda 8.2919e-06, en küçük değer ise 0.002788 kadar olmuştur. 500 karakterli metinde bu değer en fazla 20 karakterli anahtarda 9.4444e-11 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 20 karakterli anahtarda 5.7202e-11, 1000 karakterli metinde ise en büyük p değeri 20 karakterli anahtarda 6.2276e-10 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 13'ünde BCABC algoritmasının daha iyi sonuç gösterdiği, 4'ünde eşit ve 3'ünde ise GA algoritmasının daha iyi olduğu sonucuna varılmıştır.

BCABC ve PSO algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 15 karakterli anahtarda 9.5139e-06, en küçük değer ise 0.42038 kadar olmuştur. 500 karakterli metinde bu değer en fazla 5 karakterli anahtarda 4.0172e-12 olmuştur, en küçük değer ise 1.2118e-12 kadar olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 25 karakterli anahtarda 9.3352e-12, en küçük değer ise 1.2108e-12 kadar olmuştur. 1000 karakterli metinde ise en büyük p değeri 25 karakterli anahtarda 6.4697e-12 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun hepsinde BCABC algoritmasının daha iyi sonuç gösterdiğine varılmıştır.

BCABC ve ABC algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 15, 20 ve 25 karakterli anahtarlarında 3.0199e-11, en küçük değer ise 1.581e-11 kadar olmuştur. 500 karakterli metinde bu değer en fazla 20 karakterli anahtarlar 7.8787e-12 olmuştur, en küçük değer ise 15 karakterli anahtarda

1.2118e-12 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 25 karakterli anahtarda 9.3352e-12, en küçük değer ise 10 karakterli anahtarda 1.2118e-12 olmuştur. 1000 karakterli metinde ise en büyük p değeri 25 karakterli anahtarda 6.4697e-12 olmuştur, en küçük değer ise 10 karakterli anahtarda 1.2118e-12 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 19'unda BCABC algoritmasının daha iyi sonuç gösterdiği, 1'inde eşit çıktığı sonucuna varılmıştır.

BCABC ve DE algoritmalarının p değerlerine bakıldığında, 250 karakterli metinde elde edilen en büyük değer 10 karakterli anahtarda 5.2039e-12, en küçük değer ise 0.64142 kadar olmuştur. 500 karakterli metinde bu değer en fazla 10 karakterli anahtarda 0.1608 olmuştur, en küçük değer ise 20 ve 25 karakterli anahtarlarda 0.022783 olmuştur. 750 karakterli metinde, elde edilen en büyük p değeri ise 20 karakterli anahtarda 0.9591, en küçük değer ise 25 karakterli anahtarda 0.01673 olmuştur. 1000 karakterli metinde ise en büyük p değeri 5 karakterli anahtarda 0.33371 olmuştur, en küçük değer ise 2 karakterli anahtarda 0.041926 olmuştur. H-değerleri karşılaştırıldığında, deneylerdeki 20 durumun 4'ünde BCABC algoritmasının daha iyi sonuç gösterdiği, 3'ünde eşit ve 13'ünde ise DE algoritmasının daha iyi çıkardığı sonucuna varılmıştır.

Tablo 4.18'de yer alan 80 durumum değerlerinden elde edilen p-değerleri ve h-değerleri karşılaştırıldığında, deneylerdeki 80 durumun 56'sında BCABC algoritmasını daha iyi sonuç, 17'sinde eşit ve 7'sinde ise daha kötü sonuç ortaya çıkardığı sonucuna varılmıştır.

Tablo 4.17. İngilizce karakterler için Mann Whitney U test karşılaştırma sonuçları

| Şifreli metin boyutu | Anahtar boyutu | p değerleri | | | | h değerleri | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | BCABC &GA | BCABC &PSO | BCABC &ABC | BCABC &DE | BCABC &GA | BCABC &PSO | BCABC &ABC | BCABC &DE |
| 250 | 5 | 3.5705e-05 | 0.73923 | 0.42457 | 1.9905e-05 | 1 – **GA** | 0 | 0 | 1 - **DE** |
| | 10 | 0.17129 | 0.00085257 | 2.9654e-11 | 7.2438e-07 | 0 | 1 - **BCABC** | 1 – **BCABC** | 1 - **DE** |
| | 15 | 0.53951 | 8.891e-10 | 3.0199e-11 | 0.083666 | 0 | 1 - **BCABC** | 1 - **BCABC** | 1 - **DE** |
| | 20 | 0.0050842 | 5.0723e-10 | 3.0199e-11 | 1.85e-08 | 1 - **BCABC** | 1 - **BCABC** | 1 – **BCABC** | 1 - **DE** |
| | 25 | 0.00074924 | 6.722e-10 | 3.0199e-11 | 2.3715e-10 | 1 - **GA** | 1- **BCABC** | 1 – **BCABC** | 1 - **DE** |
| 500 | 5 | 0.33371 | 1.9048e-10 | 1.6373e-11 | NaN | 0 | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 10 | 0.36427 | 2.1787e-11 | 1.4411e-11 | 0.0057909 | 0 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| | 15 | 5.9675e-08 | 1.0081e-08 | 3.0104e-11 | 2.3565e-12 | 1- **GA** | 1 - **BCABC** | 1 - **BCABC** | 1- **BCABC** |
| | 20 | 0.00103 | 1.7294e-07 | 3.0199e-11 | 1.262e-11 | 1- **GA** | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| | 25 | 4.9639e-09 | 2.9972e-11 | 2.9972e-11 | 1.0167e-09 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| 750 | 5 | NaN | 1.5524e-11 | 1.2428e-11 | NaN | 0 | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 10 | 0.023356 | 2.3638e-12 | 2.3638e-12 | 0.1608 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 15 | 0.60815 | 3.2846e-11 | 2.971e-11 | 6.2216e-10 | 0 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| | 20 | 6.5081e-10 | 6.4789e-12 | 6.4789e-12 | 0.039857 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| | 25 | 1.3054e-09 | 9.3936e-12 | 9.3936e-12 | 0.064119 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 0 |
| 1000 | 5 | 0.33371 | 5.5768e-11 | 1.8741e-10 | NaN | 0 | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 10 | 0.0055843 | 1.2118e-12 | 1.2118e-12 | NaN | 1- **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 15 | 0.63222 | 2.8628e-11 | 2.8628e-11 | 4.598e-08 | 0 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |
| | 20 | 2.3404e-10 | 2.3657e-12 | 2.3657e-12 | 0.9591 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 0 |
| | 25 | 5.9233e-07 | 2.1142e-11 | 2.1128e-11 | 0.0017171 | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** | 1 - **BCABC** |

Tablo 4.18. Türkçe karakterler için Mann Whitney U test karşılaştırma sonuçları

| Şifreli metin boyutu | Anahtar boyutu | p değerleri | | | | h değerleri | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | BCABC &GA | BCABC &PSO | BCABC &ABC | BCABC &DE | BCABC &GA | BCABC &PSO | BCABC &ABC | BCABC &DE |
| 250 | 5 | 0.002788 | 1.6052e-11 | 1.581e-11 | NaN | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 10 | 1.8375e-08 | 0.42038 | 2.6695e-09 | 5.2039e-12 | 1 – **GA** | 1 – **BCABC** | 0 | 1 - **DE** |
| | 15 | 1.698e-08 | 9.5139e-06 | 3.0199e-11 | 2.3967e-11 | 1 – **GA** | 1 – **BCABC** | 1 – **BCABC** | 1 - **DE** |
| | 20 | 0.067869 | 3.2555e-07 | 3.0199e-11 | 0.0053221 | 0 | 1 – **BCABC** | 1 – **BCABC** | 1 - **DE** |
| | 25 | 8.2919e-06 | 6.5261e-07 | 3.0199e-11 | 0.64142 | 1 – **GA** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| 500 | 5 | NaN | 4.0172e-12 | 5.2692e-06 | NaN | 0 | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 10 | 0.013684 | 2.3657e-12 | 2.3657e-12 | 0.1608 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 15 | 3.4318e-07 | 1.2118e-12 | 1.2118e-12 | NaN | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 20 | 9.4444e-11 | 7.8787e-12 | 7.8787e-12 | 0.022783 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 1 - **BCABC** |
| | 25 | 2.6571e-11 | 1.4325e-11 | 1.4325e-11 | 0.022783 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 1 - **BCABC** |
| 750 | 5 | NaN | 4.4739e-12 | 2.1691e-06 | NaN | 0 | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 10 | 0.0055822 | 1.2108e-12 | 1.2118e-12 | NaN | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 15 | 1.6387e-06 | 1.7203e-12 | 1.7203e-12 | 0.33371 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 20 | 5.7202e-11 | 2.3657e-12 | 2.3657e-12 | 0.9591 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 25 | 1.1507e-11 | 9.3352e-12 | 9.3352e-12 | 0.01673 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 1 - **BCABC** |
| 1000 | 5 | 0.33371 | 6.1042e-10 | 2.711e-05 | 0.33371 | 0 | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 10 | 0.0013702 | 1.2078e-12 | 1.2118e-12 | NaN | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 15 | 4.3414e-05 | 3.1602e-12 | 3.1602e-12 | 0.081523 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |
| | 20 | 6.2276e-10 | 4.111e-12 | 4.111e-12 | 0.041926 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 1 - **BCABC** |
| | 25 | 2.95e-10 | 6.4697e-12 | 6.4697e-12 | 0.18061 | 1 – **BCABC** | 1 – **BCABC** | 1 – **BCABC** | 0 |

# 5. BÖLÜM

# TARTIŞMA-SONUÇ ve ÖNERİLER

## 5.1. Sonuç ve Öneriler

Vigenére kriptosisteminin anahtar uzayı çok büyük olduğundan, gerçek zamanlı olarak kriptosistemin analizinde kaba kuvvet metodu ile problemi çözmekte zaman maliyeti bakımından verimsiz kalmaktadır. Anahtar boyutu çok küçük olmadığında, istatistiksel teknikler de yardımcı olamamaktadır. Bu çalışmanın amacı, BCABC algoritmasının klasik kriptosistemler üzerinde, özellikle Vigenére şifrelemede, bir kriptanaliz aracı olarak fizibilitesini incelemek etkili ve verimli bir algoritma ortaya koymaktır. Vigenére şifrelemenin analizini yapmak amacıyla GA, PSO, ABC, DE ve BCABC gibi çeşitli metotlar uygulanmıştır.

Bölüm 1 ‘de Kriptoloji konusu ile ilgili genel tanımlamalar yapılarak kriptografi ve kriptoanaliz yöntemleri tanıtılmıştır. Ayrıca kriptografi ile ilgili monoalfabetik ve polialfabetik algoritmalar tanıtılarak çeşitli saldırı teknikleri de belirtilmiştir. Bölüm 2’de yapay zeka optimizasyon kavramı tanıtılara optimizasyon algoritmaları konusu ile ilgili genel tanımlamalar yapılarak optimizasyon metotları anlatılmıştır. Ayrıca sürü zekasına dayalı algoritmalar verilmiştir. Bolüm 3’de optimizasyon algoritmaları ile kriptoanaliz konusu ile ilgili literatür taraması yapılmıştır literatürde birçok boşluğun olduğu görülmüştür. Bunlardan biri, klasik şifrelemeyi çözümlemek amacıyla Yapay Arı Koloni gibi modern optimizasyon algoritmasının bu alanda hiç kullanılmamış olduğu bilgisine varılmıştır. Ayrıca literatürde yer alan İngilizce ve Türkçe metinler için kullanılan frekansı analizi tablosu da taban alınılmış, sürü zekasına dayalı algoritmalar ile kriptoanaliz işlemi yapılmış güçlü yönlerinden faydalanarak zayıf yönlerini gidermek için bir çaprazlama operatörü entegre edilen bir algoritma geliştirilmiştir. Bolüm 4’de optimizasyon algoritmalarının kriptoanaliz problemleri üzerinde test

edilmesi esnasında kontrol parametrelerine olan duyarlılığı incelenmiştir. GA'nın popülasyon boyutu aralığı 200-250 ve çaprazlama oranının 0.50-0.75 aralığında olduğunda ve mutasyon oranı 0.50-0.75 aralığında iken farklı uzunluktaki Vigenére şifreli metinler için iyi sonuçlar ürettiği görülmüştür. PSO algoritmasının, popülasyon boyutu aralığı 200-250 iken, c1 değeri 1.5-2 aralığında ve c2 değeri 1.5-1.75 aralığında farklı uzunluktaki Vigenére şifreli metinler için iyi sonuçları ürettiği görülmüştür. DE, popülasyon boyutu 50, çaprazlama oranının 0.2 olduğunda ölçekleme faktörü 1 iken farklı uzunluktaki Vigenére şifreli metinler için iyi sonuçlar ürettiği görülmüştür. ABC algoritmasının popülasyon boyutu aralığı 50-100, limit parametresinin aralığı 0.5-1 iken en iyi değerleri ürettiği görülmüştür. BCABC'nin, popülasyon boyutu 150-200 aralığında, limit parametresi 0.5-1 aralığında ve çaprazlama oranı 0.2 iken iyi sonuçlar verdiği görülmüştür. Çalışmaların ikinci bölümünde yine sürü zekasına dayalı algoritmalarını kriptanaliz alanında performanslarını incelmek için kıyaslamalar yapılmış genel olarak değerlendirildiğinde, istatistiksel araçlar kullanılarak elde edilen sonuçlara göre, PSO ve ABC tekniklerinin küçük uzunluktaki anahtarlar ile Vigenére şifrelenmiş anahtarların tamamını başarılı bir şekilde geri getirebildiği, ancak anahtar boyutu arttığı zaman yetersiz kaldıkları görülmüştür. Ayrıca, BCABC'ye kıyasla GA ile anahtarın elde edilmesi daha zayıf olduğu anlaşılmıştır. Ancak BCABC ve DE algoritmaları, 250 karakterden büyük şifreli metinler için 25 karaktere kadar olan tüm anahtarın karakterlerini geri getirebildiği tespit edilmiştir. Burada GA, PSO ve ABC algoritmalarının yerel minimumlarda sıkışıp kaldığı ve BCABC, DE algoritmalarının ise global optimuma ulaşarak çözümü elde ettiği sonucuna varılmıştır. Neticede, deneysel sonuçlara göre, Vigenére şifrelemenin kriptanalizinin doğruluğu açısından ve istatistiksel karşılaştırma sonuçlarına göre BCABC algoritmasının DE, GA, PSO ve ABC algoritmalarına oranla daha üstün ve tatmin edici performans gösterdiğini ortaya koymuştur.

İstatistiksel analiz sonuçlarına göre ise İngilizce metinleri için yapılan 80 durumum değerlerinden elde edilen p-değerleri ve h-değerleri karşılaştırıldığında, deneylerdeki 80 durumun 54'ünde BCABC algoritmasını daha iyi sonuç, 17'sinde eşit ve 9'unda ise daha kötü sonuç ortaya çıkardığı sonucuna varılmıştır. Türkçe metinlerine bakıldığında ise 80 durumum değerlerinden elde edilen p-değerleri ve h-değerleri karşılaştırıldığında, deneylerdeki 80 durumun 56'sında BCABC algoritmasını daha iyi sonuç, 17'sinde eşit

ve 7'sinde ise daha kötü sonuç ortaya çıkardığı sonucuna varılmıştır. Neticede BCABC algoritması toplam 160 durumun 110 'da DE, GA, PSO ve ABC algoritmalarına göre daha iyi olduğuna varılmıştır.

Çaprazlama operatörü bu tür problemler için çok yararlı olduğundan dolayı, Türkçe ve İngilizce metinlerde DE, GA ve BCABC algoritmalarının ABC ve PSO algoritmalarına göre daha iyi sonuçlar ürettiği görülmüştür.

İngilizce ve Türkçe de metin uzunluğu artığında gizli anahtarı bulma potansiyeli de orantılı olarak artmaktadır ve bu işleme paralel olarak iterasyon sayasını da aza düşürmektedir. Türkçe metinlerde doğru anahtar bulma oranı İngilizce metinlere göre daha yüksektir, şifrelenmiş metin uzunluğu çok kısa olduğunda bile (≤ 250 karakter). Kısa kelimeler daha kolay bulunabilir olduğundan dolayı, Türkçe dilindeki kısa kelimelerin (3-8 harfli) Türkçe metinlerde %60 oranında kullanılıyor oluşu gösterilmiştir [101]. Dolayısıyla Türkçe metnin kriptoanalizi İngilizce metnine göre nispeten daha kolaydır.

### 5.2. Gelecek Çalışmalar

Bu çalışmanın ileriki döneminde sadece harflerden ibaret metinlerin yansıra, görüntü, video gibi biçimle dikkate alınacak inceleme yapılabilir. Modern şifreleme algoritmaları üzerinde de çalışmalar gerçekleştirebilir.

# KAYNAKÇA


1. Robling Denning, D. E. 1982. Cryptography and data security. Addison-Wesley Longman Publishing Co., Inc..
2. Al-Kadit, I.A. 1992. Origins of cryptology: The Arab contributions. **Cryptologia, 16**(2), 97-126.
3. Kahn, D. (1996). The Codebreakers: The comprehensive history of secret communication from ancient times to the internet. Simon and Schuster.
4. Simmons, G. J. 1979. Symmetric and asymmetric encryption. **ACM Computing Surveys (CSUR), 11**(4), 305-330.
5. Stallings W., 2006. Cryptography and Network Security, 4/E, Pearson Education India.
6. Diffie W., Hellman M., 1976. New directions in cryptography, **IEEE transactions on Information Theory, 22** (6):644-654.
7. Hilton R., 2012. Automated cryptanalysis of monoalphabetic substitution ciphers using stochastic optimization algorithms, Ph.D. thesis, Department of Computer Science and Engineering, University of Colorado Denver.
8. Luke, S. (2013). Essentials of metaheuristics (2). Raleigh: Lulu.
9. Rechenberg, I. (1965). Cybernetic solution path of an experimental problem. Royal Aircraft Establishment Library Translation 1122.
10. Holland, J. 1975. Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control and artificial intelligence.
11. Kennedy, J., & Eberhart, R.C. 1942. "Particle swarm optimization," in Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, Perth, Australia, IEEE Service Center, Piscataway, NJ, 1995, pp.
12. Storn, R., & Price, K. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization, 11**(4), 341-359.

13. Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization (Vol. 200, pp. 1-10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.

14. Spillman, R., Janssen, M., Nelson, B., & Kepner, M. 1993. Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers. **Cryptologia, 17**(1), 31-44.

15. Solso, R. L., Barbuto, P. F., & Juel, C.L. 1979. Bigram and trigram frequencies and versatilities in the English language. **Behavior Research Methods & Instrumentation, 11**(5), 475-484.

16. Trappe, W. 2006. Introduction to cryptography with coding theory. Pearson Education India.

17. Çimen, C., Akleylek, S., Akyıldız, E., 2008. "Şifrelerin Matematiği Kriptografi 6.Basım", ODTÜ Yayıncılık, Ankara, 183 s.

18. Canbek, G., Sağıroğlu, Ş., 2006. "Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri", Grafiker Yayınları, Ankara, 204 s.

19. Buluş, H.N., 2006. Temel Şifreleme Algoritmaları ve Kriptoanalizlerinin İncelenmesi, Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne,110 s.

20. Tuncal, T., 2008. Bilgisayar Güvenliği Üzerine Bir Araştırma ve Şifreleme-Deşifreleme Üzerine Uygulama, Yüksek Lisans Tezi, Maltepe Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 72 s.

21. Schneier, B. 1993. Applied cryptography: protocols, algorithms, and source code in *C*. john wiley & sons.

22. Sabonchi, A., Obaid, Z., Akay, B. 2016. Klasik Kriptoloji Yöntemlerinin Karşilaştirilmasi. **Engineering Sciences, 11**(4), 100-108.

23. Singh, Y.K. 2012. Generalization of Vigenere Cipher. **ARPN Journal of Engineering and Applied Sciences, 7**(1), 39-44.

24. Sakallı, M.T., 2006. Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi, Doktora Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne, 91 s.

25. Çeşmeci, M.Ü., 2009. Kriptoloji Tarihi , **UEKAE Dergisi, 1**(1): 20-31.

26. Arda, D., Buluş, E., Yerlikaya, T., 2005. Simetrik Kriptosistemlerden Çok Alfabeli Yerine Koyma Metodunun Türkiye Türkçesi'nin Yapısal Özelliklerini Kullanarak Kriptanalitik İncelenmesi, *Ağ ve Bilgi Güvenliği Ulusal Sempozyumu*-ABG2005, İstanbul.

27. Dalkiliç, M. E., & Dalkiliç, G. 2002. On the cryptographic patterns and frequencies in Turkish language. In *International Conference on Advances in Information Systems* (pp. 144-153). Springer, Berlin, Heidelberg.

28. Mekhaznia, T. and Menai, M.E.B. 2014. Cryptanalysis of classical ciphers with ant algorithms. **International Journal of Metaheuristics, 3**(3):175–198.

29. Sadiq, A. T., Ali, L., and Kareem, H. 2014. Attacking transposition cipher using improved cuckoo search. **Journal of Advanced Computer Science and Technology Research, 4**(1):22–32.

30. Jain, A. and Chaudhari, N.S. 2015. A new heuristic based on the cuckoo search for cryptanalysis of substitution ciphers. *In International Conference on Neural Information Processing*, p. 206–215.

31. Shang, Y., 1997. Global Search Methods for Solving Nonlinear Optimization Problems, University of Illinois at Urbana Champaign.

32. Karaboğa, D., 2011, Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, Genişletilmiş 2. Basım, Ankara.

33. Eiben, A. E., & Smith, J. E. 2003. Introduction to evolutionary computing (Vol. 53). Berlin: springer.

34. Grefenstette, J.J., 1986. Optimization of Control Parameters for Genetic Algorithms. IEEE Trans on Systems, **Man and Cybernetics SMC** 16:122-128.

35. De Jong, K.A. 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Thesis, Department of Computer and Communication Sciences. University of Michigan, Ann Arbor.

36. Akay B., 2009, Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi, Doktora Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri.

37. Vesterstrøm, J., & Riget, J. 2002. Particle swarms: Extensions for improved local, multi-modal, and dynamic search in numerical optimization. Master's Thesis, May, 113 p.

38. H. Y. Li, A. Samsudin, B. 2005. Belaton, Heuristic cryptanalysis of classical and modern ciphers, in: Networks, 2005. Jointly held with the 2005 *IEEE 7th Malaysia International Conference on Communication*., 2005 13th IEEE International Conference on, Vol. 2, IEEE,.

39. Grundlingh, W. and Van Vuuren, J. H. 2003. Using genetic algorithms to break a simple cryptographic cipher. Retrieved March, 31.

40. De Jong, K.A. 1975. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Thesis, Department of Computer and Communication Sciences. University of Michigan, Ann Arbor, p. 121.

41. Mitchell, M. 1998. An introduction to genetic algorithms. MIT press.

42. Mansfield, R.A., 1990. Genetic Algorithms, University of Wales College of Cardiff.

43. Baker, J.E. 1985. Adaptive Selection Methods for Genetic Algorithms, Proc. 1st Int. Conf. Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, 101-111.

44. Shi Y.H., Eberhart, R.C., 1998. A Modified Particle Swarm Optimizer, *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska.

45. Kennedy J., Eberhart R. C., 2001. Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco.

46. Belgin Ö., 2007. Haberleşme Şebekelerinin Tasarımında Sezgisel Yaklaşımlar: Değişken Komşu Arama, Kuş Sürüsü Optimizasyonu, Karınca Kolonisi Optimizasyonu, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

47. Abraham, A., Guo, H., & Liu, H. 2006. Swarm intelligence: foundations, perspectives and applications. In *Swarm intelligent systems* (pp. 3-25). Springer, Berlin, Heidelberg.

48. Michalewicz, Z., 1992, "Genetic Algorithms + Data Structure = Evolution Programs", A.B.D., Springer & Verlag.

49. Shiakolas, P. S., Koladiye, D., Kebrle, J., 2005, On The Optimum Synthesis of Six-Bar Linkages Using Differential Evolution and The Geometric Centroid of Precision Positions Technique, **Mechanism and Machine Theory, 40**, 319-335.

50. Lin, Y.C., Hwang, K.S., Wang, F.S., 2004, "A Mixed-Coding Scheme of Evolutionary Algorithms to Solve Mixed-Integer Nonlinear Programming Problems", **Computers and Mathematics with Applications, 47**, 1295-1307.

51. Karaboga, D, 2004. "Yapay Zeka Optimizasyonu Algoritmaları", İstanbul, Atlas Yayın Dagıtım.

52. Yan, X. Liu, Y. Y.-F. Liu, S.-Y. 2013. A hybrid discrete arti_cial bee colony algorithm for permutation owshop scheduling problem, **Applied Soft Computing, 13** (3), 1459-1463.

53. Zhu, W. Zou, L. Wang, 2012. A new approach for data clustering using hybrid arti_cial bee colony algorithm, **Neurocomputing 97:**241-250.

54. Li, Y., Wang, Y., & Li, B. 2013. A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. **International Journal of Electrical Power & Energy Systems, 52**, 25-33.

55. Kefayat, M., Ara, A. L., & Niaki, S. N. 2015. A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources. **Energy Conversion and Management, 92,** 149-161.

56. Zhang, R., Song, S., & Wu, C. 2013. A hybrid artificial bee colony algorithm for the job shop scheduling problem. **International Journal of Production Economics, 141**(1), 167-178.

57. Yildiz, A.R. 2013. A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing. **Applied Soft Computing, 13**(5), 2906-2912.

58. Milan T., 2013. Articial bee colony (abc) algorithm with crossover and mutation, **Appl. Soft Comput,** 687-697.

59. Pandey, S., & Kumar, S. 2013. Enhanced artificial bee colony algorithm and it's application to travelling salesman problem. **HCTL Open International Journal of Technology Innovations and Research, 2,** 137-146.

60. Kumar, S., Sharma, V. K., & Kumari, R. 2014. A novel hybrid crossover based artificial bee colony algorithm for optimization problem. **arXiv preprint arXiv:**1407.5574.

61. Brajevic, I. 2015. Crossover-based artificial bee colony algorithm for constrained optimization problems. **Neural Computing and Applications, 26**(7), 1587-1601.

62. Lin, C., Qing, A., & Feng, Q. 2011. A comparative study of crossover in differential evolution. **Journal of Heuristics, 17**(6), 675-703.

63. A. Clark, E. Dawson, H. 1996. Nieuwland, Cryptanalysis of polyalphabetic substitution ciphers using a parallel genetic algorithm, in: *Proceedings of IEEE International Symposium on Information and its Applications*, 17-20.

64. Clark A., 1994. Modern optimisation algorithms for cryptanalysis, in: Intelligent Information Systems, 1994. *Proceedings of the 1994 Second Australian and New Zealand Conference on*, *IEEE*, pp. 258-262.

65. Clark A., Dawson E., 1997. A parallel genetic algorithm for cryptanalysis of the polyalphabetic substitution cipher, **Cryptologia 21** (2):129-138.

66. Clark A., Dawson E. 1998., Optimisation heuristics for the automated cryptanalysis of classical ciphers, **Journal of Combinatorial Mathematics and Combinatorial Computing, 28:** 63-86.

67. Dimovski A., Gligoroski D., 2003. Attacks on the transposition ciphers using optimization heuristics, **Proceedings of ICEST,** 1-4.

68. Verma A., Dave M., Joshi R., 2007. Genetic algorithm and tabu search attack on the mono-alphabetic substitution cipher i adhoc networks, in: Journal of Computer science, Citeseer,.

69. Song J., Yang F., Wang M., Zhang H., 2008. Cryptanalysis of transposition cipher using simulated annealing genetic algorithm, in: International Symposium on Intelligence Computation and Applications, Springer, 795-802.

70. Garg P., 2009. Genetic algorithms, tabu search and simulated annealing: A comparison between three approaches for the cryptanalysis of transposition cipher., **Journal of Theoretical & Applied Information Technology 5** (4).

71. Omran S., Al-Khalid A., Al-Saady D., 2011. A cryptanalytic attack on vigenere cipher using genetic algorithm, in: Open Systems (ICOS), *2011 IEEE Conference on, IEEE,* 59-64.

72. Bhateja A., Kumar S., 2014. Genetic algorithm with elitism for cryptanalysis of vigenere cipher, in: Issues and Challenges in Intelligent Computing Techniques (ICICT), *2014 International Conference on, IEEE,* 373-377.

73. Boryczka U., Dworak K., 2014. Genetic transformation techniques in cryptanalysis, in: Asian Conference on Intelligent Information and Database Systems, Springer, 147-156.

74. Boryczka U., Dworak K., 2014.Cryptanalysis of transposition cipher using evolutionary algorithms, in: International Conference on Computational Collective Intelligence, Springer, 623-632.

75. Uddin M.F., Youssef A.M., 2006. An artificial life technique for the cryptanalysis of simple substitution ciphers, in: *Electrical and Computer Engineering, CECE'06. Canadian Conference on*, IEEE, 2006, 1582-1585.

76. Bhateja A.K., Bhateja A., Chaudhury S., Saxena P., 2015. Cryptanalysis of vigenere cipher using cuckoo search, **Applied Soft Computing, 26:**315-324.

77. Luthra J., Pal S.K., 2011. A hybrid rey algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher, in: Information and communication technologies (WICT), *2011 world congress on, IEEE,* pp. 202-206.

78. Lin, F.-T. and Kao, C.-Y. 1995. A genetic algorithm for ciphertext-only attack in cryptanalysis. In Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., *IEEE International Conference on*, 1: 650–654.

79. Toemeh, R. and Arumugam, S. 2007. Breaking transposition cipher with genetic algorithm. **Elektronika ir Elektrotechnika, 79**(7):75–78.

80. Toemeh, R. and Arumugam, S. 2008. Applying genetic algorithms for searching key-space of polyalphabetic substitution ciphers. **International Arab Journal of In-formation Technology (IAJIT), 5**(1).

81. Muhajjar, R.A. 2010. Use of genetic algorithm in the cryptanalysis of transposition ciphers. **Basrah Journal of Scienec, 28**(1):49–57.

82. Hausman, M. and Erickson, D. 2009. A dominant gene genetic algorithm for a substitution cipher in cryptography. CS591 at the University of Colorado.

83. Dureha, A. and Kaur, A. 2013. A generic genetic algorithm to automate an attack on classical ciphers. **International Journal of Computer Applications, 64**(12).

84. Heydari, M., Shabgahi, G.L., Heydari, M.M. 2013. Cryptanalysis of transposition ciphers with long key lengths using an improved genetic algorithm. **World Applied Sciences Journal, 21**(8):1194-1199.

85. Al-Khalid, A., Omran, S., and Hammood, D. A. 2013. Using genetic algorithms to break a simple transposition cipher. *In 6th International Conference on Information Technology ICIT*.

86. Alkathiry, O. and Al-Mogren, A. (2014). A powerful genetic algorithm to crack a transposition cipher. **International Journal of Future Computer and Communication, 3**(6):395.

87. Saveetha, P., Arumugam, S., and Kiruthikadevi, K. (2014). Cryptography and the optimization heuristics techniques. **International Journal, 4**(10).

88. Bhateja, A., Kumar, S., and Chaudhary, H. (2014). Analysis of different cryptosystems using meta-heuristic techniques. **Journal of Computer Science and Engineering, 1**(2):1931-1934.

89. Jadaun, A. S., Chaudhary, V., Sharma, L., & Singh, G. P. Deciphering of Transposition Ciphers u Deciphering of Transposition Ciphers using Genetic sing Genetic Algorithm Algorithm.

90. Sadeghzadeh, M. and Taherbaghal, M. 2014. A new method for decoding an encrypted text by genetic algorithms and its comparison with tabu search and simulated annealing. **Management Science Letters, 4**(2):213–220.

91. Al-Khalid, A. S. and Al-Khfagi, A. O. 2015. Cryptanalysis of a hill cipher using genetic algorithm. *In Computer Networks and Information Security (WSCNIS),* 2015 World Symposium on, pages 1–4.

92. Bergmann, K. P., Scheidler, R., and Jacob, C. 2015. Cryptanalysis using genetic algorithms. *In Proceedings of the 10th annual conference on Genetic and evolutionary computation,* p. 1099–1100.

93. Forsyth, W.S. and Safavi-Naini, R. 1993. Automated cryptanalysis of substitution ciphers. **Cryptologia, 17**(4):407–418.

94. Giddy, J. and Safavi-Naini, R. 1994. Automated cryptanalysis of transposition ciphers. **The Computer Journal, 37**(5):429–436.

95. Uddin, M.F. and Youssef, A.M. 2006. Cryptanalysis of simple substitution ciphers using particle swarm optimization. In Evolutionary Computation, 2006. CEC 2006. *IEEE Congress on*, pages 677–680.

96. Hameed, S. M. and Hmood, D.N. 2010. Particles swarm optimization for the cryptanalysis of transposition cipher. **Journal of Al-Nahrain University, 13**(4):211–215.

97. Wulandari, G. S., Rismawan, W., and Saadah, S. 2015. Differential evolution for the cryptanalysis of transposition cipher. In Information and Communication Technology (ICoICT), *2015 3rd International Conference on*, pages 45–48.

98. Russell, M. D., Clark, J. A., and Stepney, S. 2003. Making the most of two heuristics: Breaking transposition ciphers with ants. *In Evolutionary Computation, 2003.* CEC'03. The 2003 Congress on, 4, 2653–2658.

99. A. Dureha, A. Kaur, A generic genetic algorithm to automate an attack on classical ciphers, **International Journal of Computer Applications, 64**(12).

100. James, Practical cryptography, accessed = 2017-11-30 (2009-2012). URL **http://practicalcryptography.com/ciphers/**

101. Dalkılıç, M. E., Dalkılıç, G. 2002. On the Cryptographic Patterns and Frequencies in Turkish Language. *In International Conference on Advances in Information Systems* (Vol. 2457, ADVIS 2002. Lecture Notes in Computer Science, pp. 144-153). Berlin: Springer. doi:10.1007/3-540-36077-8_14

102. Math Works, “Implementation of Vigenere vigenere Encrypt and Decrypt”, Project, [Online] Copyright (c) 2015, Available: Yarpiz (**www.mathworks.com**).

103. Kalami S.M., “Implementation of Genetic Algorithms GA”, Project, [Online] Copyright (c) 2015, Available: Yarpiz (**www.yarpiz.com**).

104. Kalami S.M., “Implementation of Particle Swarm Optimization (PSO)”, Project, [Online] Copyright (c) 2015, Available: Yarpiz (www.yarpiz.com).

105. Kalami S.M., “Implementation of Differential Evolution (DE)”, Project, [Online] Copyright (c) 2015, Available: Yarpiz (www.yarpiz.com).

106. Kalami S.M., “Implementation of Artificial Bee Colony ABC”, Project, [Online] Copyright (c) 2015, Available: Yarpiz (www.yarpiz.com).

107. Karaboga D. and Akay B. “Implementation of Artificial Bee Colony ABC”, Project, [Online] Copyright © 2009, Erciyes University, Intelligent Systems Research Group, The Dept. of Computer Engineering. Available: (**www.abc.erciyes.edu.tr**).

# ÖZGEÇMİŞ

## KİŞİSEL BİLGİLER

**Adı Soyadı:** Arkan Kh Shakr SABONCHI

**Uyruğu:** Irak (IQ)

**Doğum Tarihi ve Yeri:** 05.05.1985 - Kerkük

**Medeni Durum:** Evli

**e-mail:** arkankhaleel@gmail.com

**Yazışma Adresi:** Tablakay Mah. Sayer Cad. Apt No 18. Daire No 18

Talas - Kayseri.

## EĞİTİM

| Derece | Kurum | Mezuniyet Tarihi |
|---|---|---|
| Yüksek Lisans | Çankaya Üniversitesi, Matematik Bilgisayar | 2013 |
| Lisans | Mosul Üniversitesi, Bilgisayar | 2007 |
| Lise | El Velit Erkek Lisesi, Kerkük / Irak | 2003 |

## İŞ DENEYİMLERİ

| Yıl | Kurum | Görev |
|---|---|---|
| 2007-2008 | Kerkük Teknik Enstitüsü | Bilgisayar Öğretmeni |
| 2007-2011 | Kerkük Genel Eğitim Müdürlüğü | Bilgisayar Öğretmeni |

## YABANCI DİL

İngilizce, Arapça

# YAYINLAR

1. Obaid, Z., Sabonchi, A., & AKAY, B., (2016). **Klasik Kriptoloji Yöntemlerinin Karşılaştırılması**. e-Journal of New World Sciences Academy , vol.11, 100-108.

2. Obaid, Z., Sabonchi, A., & AKAY, B., (2016). **Comparison of Classical Cryptography Methods**. In 2016 1th International Science Symposium Proceeding Book ISS2016 (pp. 01-02). Anadolu Kulübü, Büyükada-İstanbul.

3. Sabonchi, A. K. S., & Görür, A. K. (2017, May). Plagiarism detection in learning management system. In 2017 8th International Conference on Information Technology (ICIT) (pp. 495-500). IEEE, Amman, Ürdün.

4. Sabonchi, A. K. S., & AKAY, B., (2017). **Cryptanalysis using Artificial Bee Colony Algorithm Guided by Frequency based Fitness Value**. 1st International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2017 (pp.334-338). Tokat, Turkey

5. Sabonchi, A. K. S., & AKAY, B., (2017). **Cryptanalytic of Substitution Ciphers by Artificial Bee Colony Algorithm Guided by Statistics based Fitness Function**. 8th International Advanced Technologies Symposium (IATS'17) (pp. 3874- 3879). Elazığ, Turkey.

6. Sabonchi, A., & Obaid, Z., (2018). **Brain Tumor Detection in Medical Images Using Clustering and Morphological Operations Methods**. 2nd International Students Science Congress, 4-5 May, Izmir – Turkey

7. Sabonchi, A. K. S., & AKAY, B., (2018). **Yapay Zeka Algoritmaları ile Kriptoanaliz**. Boğaz'da Yapay Öğrenme, İsmail Arı Yaz Okulu 2018, BYÖYO 2018 Poster sunum. Boğaziçi Üniversitesi– Turkey

8. Sabonchi, A., & AKAY, B., (in press). **A Binomial Crossover Based Artificial Bee Colony Algorithm for Cryptanalysis of Polyalphabetic Cipher**. Technical Gazette, doi: 10.17559/TV-20190422225110, (SCIE İndekslerine Giren Dergi).

9. Sabonchi, A., & AKAY, B., (in press). **Cryptanalysis of Polyalphabetic cipher using Differential Evolution Algorithm**. Technical Gazette, doi: 10.17559/TV-20190314095054, (SCIE İndekslerine Giren Dergi).

10. Sabonchi, A., & AKAY, B., (under review). **A survey on the Metaheuristics for Cryptanalysis of Substitution and Transposition Ciphers**.