

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**UNIFORM POPÜLASYON VE TÜREVE DAYALI
META-SEZGİSEL YENİ YÖNTEM GELİŞTİRME**

DOKTORA TEZİ

Ebubekir SEYYARER

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Ali KARCI

EYLÜL 2021

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**UNIFORM POPÜLASYON VE TÜREVE DAYALI
META-SEZGİSEL YENİ YÖNTEM GELİŞTİRME**

DOKTORA TEZİ

**Ebubekir SEYYARER
(23616190014)**

Bilgisayar Mühendisliği Anabilim Dalı

**Tez Danışmanı: Prof. Dr. Ali KARCI
Eş Danışman: Dr. Öğr. Üyesi Abdullah ATEŞ**

EYLÜL 2021

TEŐEKKÜR VE ÖNSÖZ

Lisans eğitimimden bu yana bana destek olan, öneri, bilgi ve tecrübesiyle yol gösteren kıymetli danışman hocam Sayın Prof. Dr. Ali KARCI'ya, doktora eğitimimde yol gösterici önerilerini ve yardımlarını esirgemeyen, bilgi birikimini sabır ve cömertlikle benimle paylaşan kıymetli eş danışman hocam Sayın Dr. Öğr. Üyesi Abdullah ATEŐ'e, çalışmalarına yaptığı yorumlar ve önerilerle hep katkı sunan arkadaşım Öğr. Gör. Dr. Faruk AYATA'ya, başta tez izleme komitelerimde bulunan jüri üyelerine ve diğer tüm Bilgisayar Mühendisliđi Anabilim Dalı öğretim üyelerine,

Tüm hayatım boyunca olduđu gibi bu çalışmalarım süresince de benden her türlü desteklerini esirgemeyen aileme, özellikle de bu süreçte çokça zamanlarından çaldığım ve ihmal ettiğim sevgili eşim Esra SEYYARER'e ve kızlarım Esma Ada ile Ela İkra'ya,

Őükranlarımı sunar, teşekkür ederim.

ONUR SÖZÜ

Doktora tezi olarak sunduđum “Uniform Popülasyon ve Türeve Dayalı Meta-Sezgisel Yeni Yöntem Geliştirme” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla dođrularım.

Ebubekir SEYYARER



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	vii
SEMBOLLER VE KISALTMALAR	viii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ.....	1
1.1 İlişkili Çalışmalar.....	9
1.1.1 Başlangıç popülasyonları üzerine yapılan çalışmalar.....	9
1.1.1.1 Stokastik yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar	10
1.1.1.2 Deterministik yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar.....	14
1.1.1.3 Sezgisel yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar	18
1.1.2 Optimizasyon algoritmaları üzerine yapılan çalışmalar	20
1.1.3 Hata fonksiyonları üzerine yapılan çalışmalar	21
1.1.4 Çok değişkenli lineer regresyon üzerine yapılan çalışmalar	22
1.1.5 İris veri seti üzerine yapılan çalışmalar	22
1.2 Tezin Amacı.....	23
1.3 Tezin Organizasyonu	23
2. BAŞLANGIÇ POPÜLASYONLARI	25
2.1 Stokastik Yöntemler (Stochastic Methods)	25
2.1.1 Sözcük rastgele sayı üreticileri (pseudo random number generation, PRNG) 25	
2.1.1.1 Orta kare yöntemi (middle square method, MSM).....	25
2.1.1.2 Doğrusal eşleşik üretici (linear congruential generator, LCG)	26
2.1.1.3 Karesel eşleşik üretici (quadratic congruential generator, QCG)	26
2.1.1.4 Gecikmeli Fibonacci üretici (lagged Fibonacci generator, LFG veya LFib)	27
2.1.1.5 Blum Blum Shub (BBS)	27
2.1.1.6 Geri beslemeli kaydırma yazmacı (feedback shift register, FSRG)	28
2.1.1.7 PRNG yöntemlerinin analizleri	28
2.1.2 Yarı rastgele sayı üreticileri (quasi random number generation, QRNG)	29
2.1.2.1 Van Der Corput dizisi	29
2.1.2.2 Halton dizisi	30
2.1.2.3 Faure dizisi.....	31
2.1.2.4 Sobol dizisi	33
2.1.2.5 Kronecker dizisi	34

2.1.2.6	Hammersley dizisi	34
2.1.2.7	QRNG yöntemlerinin analizleri	35
2.1.3	Gerçek rastgele sayı üreticileri (true random number generation, TRNG)....	35
2.1.3.1	Fiziksel olmayan TRNG'ler (Non-Physical, NPTRNG)	35
2.1.3.2	Fiziksel TRNG'ler (PTRNG).....	36
2.1.3.2.1	Gürültü (noise)	36
2.1.3.2.2	Kaos (chaos).....	36
2.1.3.2.3	Kuantum (quantum)	37
2.1.3.2.4	Serbest hareketli osilatör (free running oscilator, FRO)	37
2.1.3.3	TRNG yöntemlerinin analizleri	38
2.2	Deterministik Yöntemler	38
2.2.1	Düşük tutarsızlık dizileri (low discrepancy sequences)	38
2.2.2	Kaos (chaos)	38
2.2.2.1	Tek boyutlu haritalar (one-dimensional maps)	39
2.2.2.1.1	Lojistik harita (logistic map)	39
2.2.2.1.1	Çember harita (circle map).....	40
2.2.2.1.2	Gauss haritası (gauss map).....	40
2.2.2.1.3	Sinüs haritası (sine map)	41
2.2.2.1.4	Çadır haritası (tent map).....	41
2.2.2.2	İki boyutlu haritalar (two-dimensional maps)	42
2.2.2.2.1	Henon haritası (henon map)	42
2.2.2.2.2	Baker haritası (baker map)	42
2.2.2.2.3	Arnold Kedi haritası (Arnold Cat map).....	43
2.2.2.2.4	Zaslavskii haritası (Zaslavskii map).....	43
2.2.2.3	Kaos yöntemlerinin analizleri	44
2.3	Sezgisel Yöntemler (Heuristic Methods).....	44
2.3.1	Baskın olmayan sıralı genetik algoritma (non-dominated sorting genetic algorithm, NSGA)	44
2.3.1.1	NSGA-II.....	46
2.3.1.2	NSGA_III.....	46
2.3.2	Uyarlanabilir rastgelelik (adaptive randomness, AR)	47
2.3.3	Uyarlanabilir genetik algoritma ile başlatma (adaptive genetic algorithm Initialize, AGA).....	48
2.3.4	Muhalefet tabanlı başlangıç popülasyonu (opposite based initial population)	48
2.3.5	Farklı rastgelelik (distinct random)	49
2.3.6	Bilgi kazancı sıralaması (information gain ranking)	49
2.3.7	Açgözlü yaklaşım (greedy).....	50
2.3.8	Küçük rastgele (small random).....	50
2.3.9	Fidan gelişim algoritması (saplings growing up algorithm, SGuA).....	50

2.3.10	Karıştırılmış halton dizisi (scrambled halton sequence).....	52
2.3.11	Genetik çeşitlilikle uygulanabilir başlangıç popülasyonu (feasible initial population with genetic diversity).....	53
2.3.12	Sezgisel bir yöntemle başlangıç popülasyonu çeşitliliği (initial population diversity with a heuristic method).....	54
2.3.13	Sezgisel yöntemlerin analizleri.....	55
3.	TÜREV DENKLEMLERİ	56
3.1	SGD	56
3.2	Momentum.....	57
3.3	Adagrad.....	57
3.4	RMSprop.....	58
3.5	Adadelta	58
3.6	Adam.....	59
4.	GELİŞTİRİLEN YÖNTEM; DETERMİNİSTİK METASEZGİSEL ALGORİTMA.....	60
4.1	Problemin Formülasyonu.....	60
4.2	DeneySEL Çalışma	63
4.3	Kullanılan Amaç Fonksiyonları.....	67
5.	UYGULAMA SONUÇLARI.....	69
6.	SONUÇLAR	79
KAYNAKLAR.....		81
ÖZGEÇMİŞ		92

ÇİZELGELER DİZİNİ

Çizelge 2.1 : Orta kare yöntemi örneği.	25
Çizelge 2.2 : 3 tabanında 7 elemanlı Van Der Corput dizisi.....	30
Çizelge 2.3 : Van Der Corput ilk terimleri.	30
Çizelge 2.4 : 3 boyutlu 6 elemanlı Halton dizisi.	31
Çizelge 2.5 : 3 boyutlu Faure dizisi.	32
Çizelge 2.6 : 3 boyutlu Sobol dizisi (çekirdek=0).	34
Çizelge 3.1 : Dik iniş yöntemlerinin karşılaştırılması.....	56
Çizelge 4.1 : İris veri setinin genel görünümü.	62
Çizelge 4.2 : İris veri setinin çalışma görünümü.....	63
Çizelge 5.1 : N(başlangıç popülasyonu sayısı)=10 için sonuç tablosu.	70
Çizelge 5.2 : N(başlangıç popülasyonu sayısı)=50 için sonuç tablosu.	71
Çizelge 5.3 : N(başlangıç popülasyonu sayısı)=100 için sonuç tablosu.	72



ŞEKİLLER DİZİNİ

Şekil 1.1 : Optimizasyon problemlerinin sınıflandırılması.....	2
Şekil 1.2 : Algoritmaların sınıflandırılması.....	5
Şekil 1.3 : Başlangıç popülasyonu oluşturma yöntemleri.....	8
Şekil 2.1 : Örnek bir gürültü bazlı TRNG.....	36
Şekil 2.2 : Hızlı (soldaki) ve yavaş (sağdaki) FRO örnekleri.....	37
Şekil 2.3 : NSGA'nın akış diyagramı.....	45
Şekil 2.4 : NSGA-II'nin akış diyagramı.....	46
Şekil 2.5 : NSGA-III'ün akış diyagramı.....	47
Şekil 2.6 : Muhalefet tabanlı popülasyon başlangıcı.....	49
Şekil 2.7 : Standart halton dizisi.....	52
Şekil 2.8 : Karıştırılmış halton dizisi.....	52
Şekil 2.9 : Karıştırılmış Halton dizisi.....	53
Şekil 2.10 : Rastgele başlatılan popülasyon.....	53
Şekil 3.1 : Dik iniş yöntemlerin evrimsel haritası.....	56
Şekil 3.2 : Momentumsuz SGD.....	57
Şekil 3.3 : Momentumlu SGD.....	57
Şekil 4.1 : İris bitkisinin öznelilikleri.....	61
Şekil 4.2 : İris bitkisinin özneliliklerinin dağılımı.....	61
Şekil 4.3 : Çalışmanın tüm modülleri.....	64
Şekil 4.4 : Uygulamanın akış diyagramı.....	65
Şekil 5.1 : Dik iniş yöntemlerinin IAE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıkları.....	74
Şekil 5.2 : Dik iniş yöntemlerinin IAE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıkları.....	74
Şekil 5.3 : Dik iniş yöntemlerinin ITAE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıkları.....	75
Şekil 5.4 : Dik iniş yöntemlerinin ITAE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıkları.....	75
Şekil 5.5 : Dik iniş yöntemlerinin MSE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıkları.....	76
Şekil 5.6 : Dik iniş yöntemlerinin MSE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıkları.....	76
Şekil 5.7 : Dik iniş yöntemlerinin ISE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıkları.....	77
Şekil 5.8 : Dik iniş yöntemlerinin ISE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıkları.....	77

SEMBOLLER VE KISALTMALAR

MLR	: Çok Değişkenli Bir Lineer Regresyon (Multivariate Linear Regression)
SGD	: Stokastik Dik İniş (Stochastic Gradient Descent)
IAE	: Hatanın Mutlak Değerinin İntegrali (Integral of the absolute value of the error)
ITAE	: Zaman Ağırlıklı Mutlak Hatanın İntegrali (Integral of the time-weighted absolute error)
MSE	: Hata Karelerinin Ortalaması (Mean Square of the Error)
ISE	: Hata Karelerinin İntegrali (Integral of the square error)
MaOPs	: Çok Amaçlı Optimizasyon Problemleri (Multi Objective Optimization Problems)
SGuA	: Fidan Gelişim Algoritması (Saplings Growing up Algorithm)
CPU	: Merkezi İşlem Birimi (Central Process Unit)
TRNG	: Gerçek Rastgele Sayı Üretici (True Random Number Generator)
PRNG	: Sözde Rastgele Sayı Üretici (Pseudo Random Number Generator)
QRNG	: Yarı Rastgele Sayı Üretici (Quasi Random Number Generator)
FM	: Frekans Modülasyonu (Frequency modulation)
NSGA	: Baskın olmayan sıralı genetik algoritma (Non-Dominated Sorting Genetic Algorithm)

ÖZET

Doktora Tezi

UNIFORM POPÜLASYON VE TÜREVE DAYALI META-SEZGİSEL YENİ YÖNTEM GELİŞTİRME

Ebubekir SEYYARER

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

93 + x sayfa

2021

Danışman: Prof. Dr. Ali KARCI

Bir problemin en iyi sonucunu aramak ve bulmak için optimizasyon algoritmaları kullanılmaktadır. Kesin ve yaklaşık yöntemler (sezgisel ve meta-sezgisel) olarak ikiye ayrılan optimizasyon algoritmaları, tüm alanlarda kullanıldığı gözlemlenmektedir. Optimizasyon algoritmaları temelde beş bölümden (başlangıç popülasyonu oluşturma, uygunluk hesaplama, seçim, yeni nesil oluşturma ve karar verme) oluşmaktadır. Araştırmacılar ilerleyen teknolojilerle bu beş bölümde değişiklikler ve geliştirmeler yapmışlardır. En az çalışma başlangıç popülasyonu oluşturma bölümünde yapılmıştır. Günümüzde halen en çok rastgele başlatma yöntemi kullanılmaktadır. Bir standart başlatma yöntemi olarak kabul edildiği için araştırmacılar bu konu üzerinde çok araştırma yapmamışlardır. Son yıllarda popülasyon çeşitliliğini ve düzgün dağılımı arttırmak için bir çok yeni popülasyon başlatma yöntemi önerilmektedir.

Bu tez kapsamında, başlangıç popülasyonu oluşturma yöntemleri detaylı olarak incelenmektedir ve bu incelemenin sonucunda yeni bir kategorizasyon önerilmektedir. Ayrıca literatüre deterministik yeni bir başlangıç popülasyonu oluşturma yöntemi önerilmektedir. Bu yeni başlatma yöntemi ile başlatılan çok değişkenli bir lineer regresyon (MLR) modeli kullanarak iris veri setini temsil edecek lineer bir fonksiyon elde edilmektedir. Bu fonksiyondaki katsayıların optimum değerlerini bulmak için SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam optimizasyon algoritmaları kullanılmaktadır. Ayrıca amaç fonksiyonu olarak IAE, ITAE, MSE ve ISE hata fonksiyonlarından yararlanılmaktadır. Öncelikle alt ve üst sınırlar arasında deterministik ve stokastik başlatma yöntemleri ile uygulamaların başlangıç popülasyonları oluşturulmaktadır. Stokastik olarak başlatılan uygulama literatürdeki gibi birçok defa çalıştırılıp ortalama değerleri alınmaktadır. Buna karşılık deterministik olarak başlatılan uygulama ise bir kez çalıştırılmaktadır. Deterministik ve stokastik başlatılan uygulamalarda katsayılar ve adım sayıları birbirine yakın olduğu gözlemlenmektedir. Fakat deterministik olarak başlatılan uygulama ile zamansal kazanım elde edilmektedir. Elde edilen sonuçlar karşılaştırılarak analiz edilmektedir. Karşılaştırmalar sonucunda Adadelta optimizasyon algoritması ve MSE amaç fonksiyonuyla elde edilen lineer model en iyi performansı göstermektedir.

Anahtar Kelimeler: Deterministik Başlangıç Popülasyonları, Stokastik Başlangıç Popülasyonları, Çok Değişkenli Lineer Regresyon, Optimizasyon Algoritmaları

ABSTRACT

Phd. Thesis

INNOVATIVE META-HEURISTIC METHOD DEVELOPMENT BASED ON UNIFORM POPULATION AND DERIVATIVE

Ebubekir SEYYARER

Inonu University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

93 + x pages

2021

Supervisor: Prof. Dr. Ali KARCI

Optimization algorithms are used to search and find the best solution for a problem. Optimization algorithms, which are divided into two as exact and approximate methods (heuristic and metaheuristic), are observed to be used in all areas. Optimization algorithms basically consist of five parts (initial population generation, fitness computation, selection, creating a new generation and decision making). Researchers have made changes and improvements in these five sections with advancing technologies. The least work has been done in the initial population generation section. The random initiation method is still used the most today. Researchers have not done much research on this topic as it is considered a standard initiation method. In recent years, many new population initiation methods have been proposed to increase population diversity and uniform distribution.

Within the scope of the thesis, initial population generation methods are examined in detail and a new categorization is proposed as a result of this review. In addition, method of creating a deterministic new initial population is suggested to literature. a linear function to represent the iris data set was obtained by making use of the multivariate linear regression (MLR) model initiated with this new initialization method. SGD, Momentum, Adagrad, RMSProp, Adadelta and Adam optimization algorithms were used to find the optimum values of coefficients of this function. In addition, IAE, ITAE, MSE and ISE error functions were adopted as the objective function. First, initial populations of the methods were developed by using deterministic and stochastic initialization methods between upper and lower bounds. The method that was initialized stochastically was run several times as seen in literature and the mean values were calculated. On the other hand, the application that was initialized deterministic was only run once. According to deterministic and stochastic initialization Outputs, theta and iteration number were found to be close. However, temporal gain was achieved from the application that was initialized deterministic. Generated outputs were compared and analyzed. According to comparisons, the linear model obtained using the Adadelta optimization algorithm and the MSE objective function performed best.

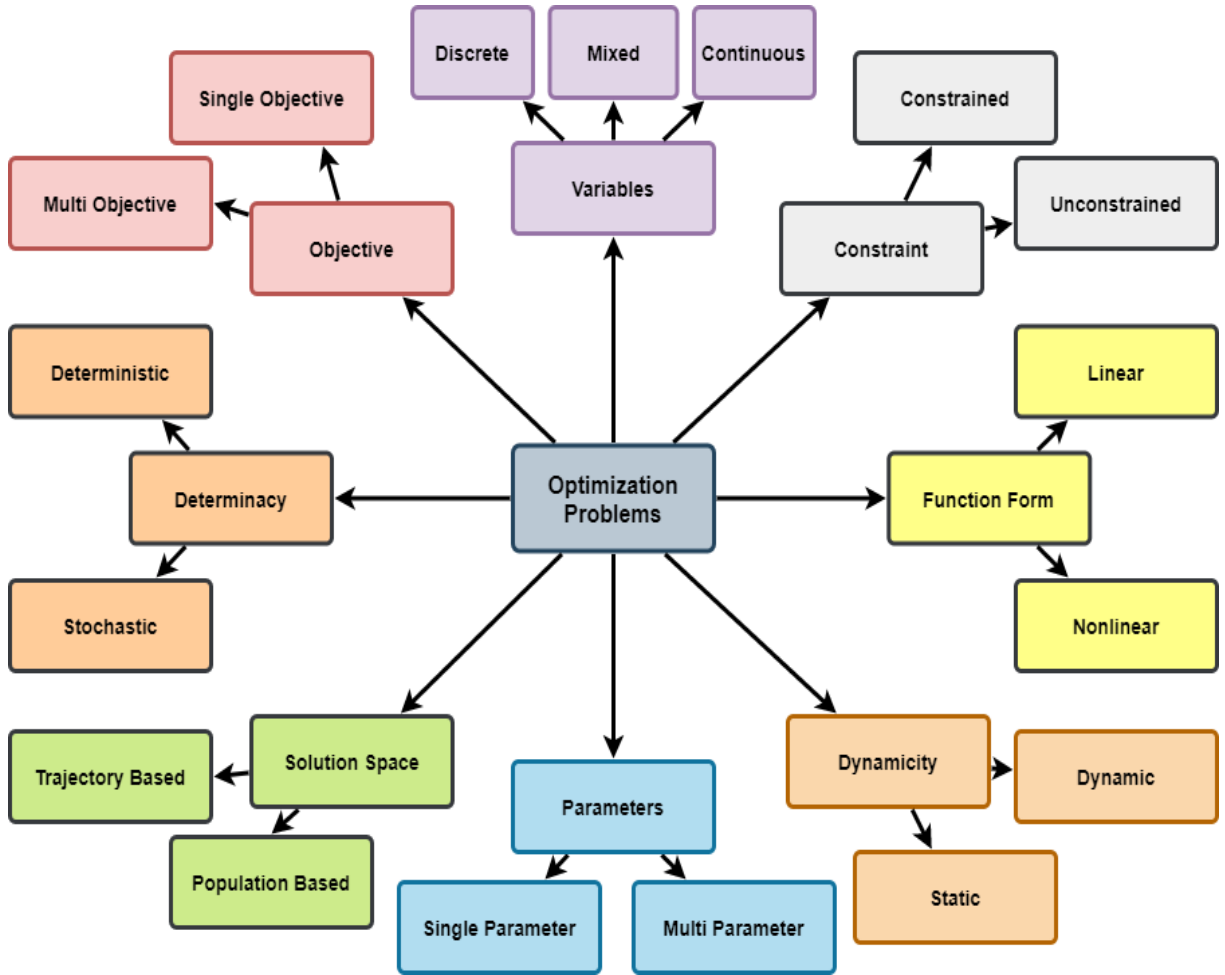
Keywords: Deterministic Initial Population, Stochastic Initial Population, Multivariate Linear Regression, Optimization Algorithms

1. GİRİŞ

Günümüzdeki problemlerin karmaşıklıkları ve boyutlarının artması sebebiyle analitik yöntemlerle bir çözüm elde etmek zorlaşmaktadır. NP (Nondeterministic Polynomial time)-hard olan bu optimizasyon problemlerin çözümünde analitik yöntemler yerine çözümler arasındaki en iyisini ve bu sonucu optimal sürede bulan sezgisel yaklaşımlar önerilmektedir. Bunlar genellikle doğadan esinlenerek ortaya konulan yaklaşımlardır. Optimizasyon temelli olan sezgisel yaklaşımların en popüler algoritmaları, Yapay Sinir Ağları (Dokeroglu ve diğ., 2019; Villarrubia ve diğ., 2018), Genetik Algoritma (Dokeroglu ve diğ., 2019; Pandey ve diğ., 2020; Villarrubia ve diğ., 2018), Parçacık Sürü Optimizasyonu (Dokeroglu ve diğ., 2019; Pandey ve diğ., 2020; Shihabudheen ve diğ., 2018; Villarrubia ve diğ., 2018) vb. algoritmalarıdır. Bu popüler yaklaşımların matematiksel modellemeleri karmaşık değildir. Karmaşık olmaması da büyük avantaj sağlamaktadır. Dezavantajları ise, bazı özel problemlerin çözümüne uygulanamamakta ya da uygulandığında ise zaman olarak çok fazla maliyetli olmalarıdır. Bu problemlerdeki maliyetleri aşmak için bazı özel optimizasyon yöntemleri de önerilmektedir. Bu yöntemlerin çözüm bulmada ve süre bakımından oldukça yüksek başarı oranları elde ettikleri gözlemlenmiştir. Fidan Gelişim Algoritması (Alatas, 2019), Lig Şampiyonluk Algoritması (Husseinzadeh Kashan ve diğ., 2019) vb. algoritmalar örnek gösterilebilir. Literatürde çok sayıda optimizasyon yöntemi bulunmaktadır, bazıları genel bazıları ise özel yaklaşımlardır (Akanksha Rai, 2019; Alpaydin, 2014; Deisenroth ve diğ., 2020; Garbade, 2018; Jason Brownlee, 2017; Kubat, 2017; Shalev-Shwartz ve Ben-David, 2013).

Optimizasyonun kelime anlamı eniyilemedir (Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Ponce-Ortega ve Hernández-Pérez, 2019; Zaki ve H. Zaied, 2019). En iyisini bulmak veya bulmaya çalışmaktır. Bir problemin en uygun çözümünün bulunmasına optimizasyon denilmektedir. Tüm alanlarda optimize edilecek problemler vardır. Yatırımcılar, üreticiler veya mühendisler yaptıkları işleri optimize etmek isterler. Örneğin yatırımcı aşırı riskten kaçınmak, üretici maksimum verimlilik ve mühendis ise en iyi performansı istemektedir. Aynı şekilde Doğa da, minimum enerji için optimizasyon yapmaktadır (Robinson, 2006). Aşırı riskten kaçınmak, maksimum verimlilik, en iyi performans ve minimum enerji gibi örnekler, optimizasyonda “amaçlar” olarak adlandırılmaktadır. Bir optimizasyon problemi, çok amaçlı da olabilmektedir. Amacı üçten fazla olan optimizasyon problemlerine, çok amaçlı optimizasyon problemleri (MaOPs) denilmektedir (Chong ve Zak, 2004). Optimizasyon, teoride veya gerçek hayattaki problemlerin amaçları doğrultusunda en iyi

sonucunu bulmak için ayarlama yapmaktır. Başka bir ifadeyle optimizasyon, sonuçları belli olan bir sistemin girdilerinin veya girdi değerlerinin nasıl ve ne olacağını belirlemektir (Türkay, 2013). Bu yüzden simülasyon ile karıştırılmaması gerekmektedir. Simülasyonlarda sistem girdileri bellidir. Bu girdiler girilerek sonuçlar üretilmektedir. Optimizasyon problemleri Şekil 1.1’de sınıflandırılmaktadır.



Şekil 1.1 : Optimizasyon problemlerinin sınıflandırılması.

Optimizasyon problemlerini çözmek için kullanılan yöntemler aşağıdaki gibi iki grupta kategorize edilmektedir (Chong ve Zak, 2004; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Ponce-Ortega ve Hernández-Pérez, 2019; Robinson, 2006; Zaki ve H. Zaid, 2019);

- Kesin (Tam) Metotlar (Analitik Yöntemler)
- Yaklaşık Metotlar
 - Sezgisel Yöntemler
 - Metasezgisel Yöntemler

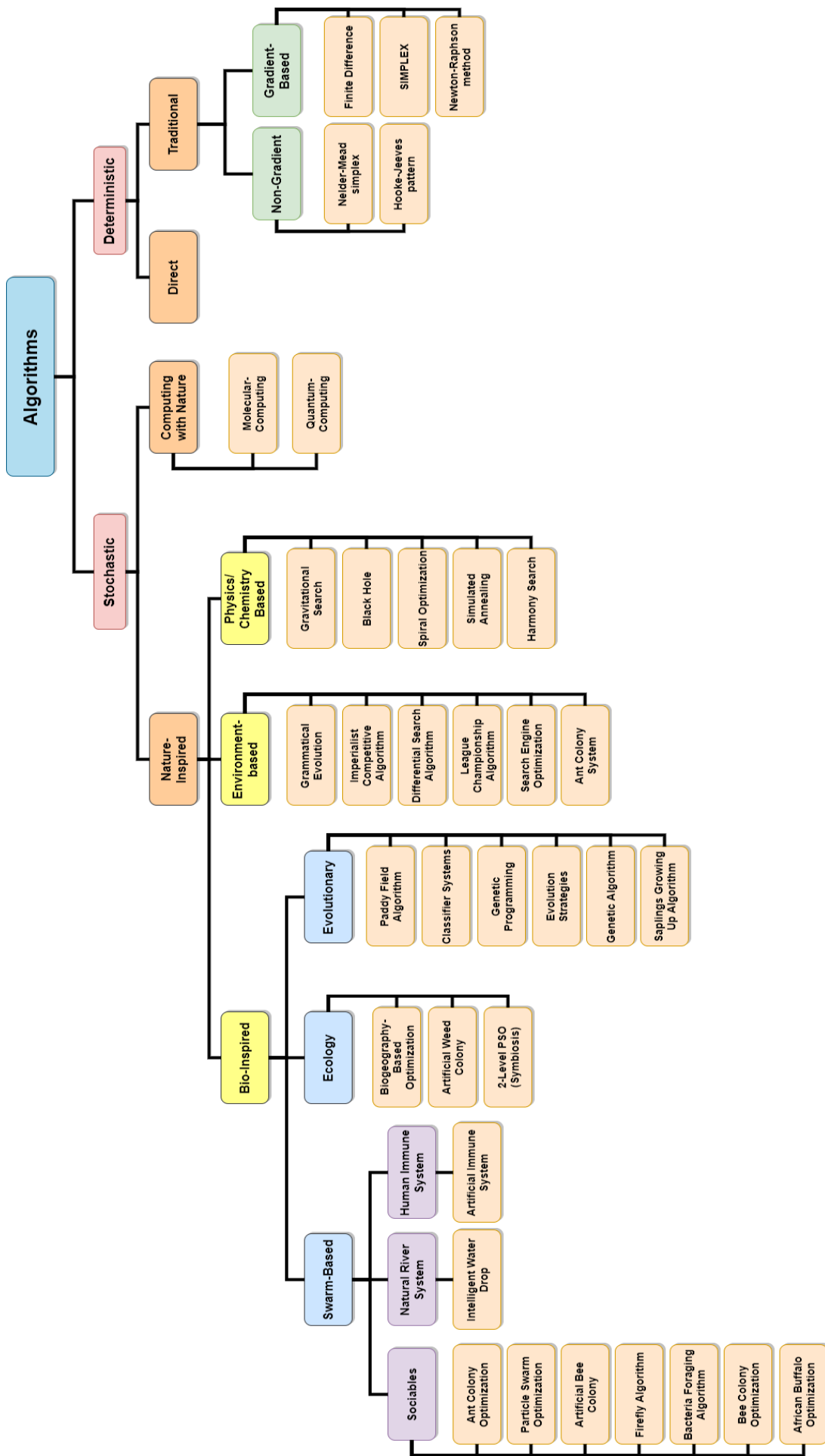
Kesin metotlar, tam metotlar olarak da adlandırılmaktadır. Sezgisel olmayan, deterministik ve sistematik optimizasyon algoritmalarının özel bir türüdür. Amaç fonksiyonlarıyla ilgili ön bilgiler yeterli ise, global optimumu bulmayı makul miktarda kaynak kullanarak garanti edebilmektedir. Bu kaynaklara, hesaplama süresi veya fonksiyon değerlendirmeleri gibi örnekler verilebilir. Kesin metotların probleme uygulanabilirliği varsa, sezgisel yöntemlerden daha güvenilir olmaktadır. Dahası, bu metotları, teoride kara kutu problemlerine uygulamak mümkündür. Çünkü sonlu sürede global optimumu bulma yeteneği mevcuttur. Ancak, pahalı ve yoğun bir arama gerekmektedir. Bu nedenle hesaplama süreleri üstel olmaktadır ve bazı uygulamalar dışında tercih edilmemektedirler (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019).

Yaklaşık yöntemlerin ilki olan sezgisel algoritmalar, optimizasyon problemleri için hedeflenen çözümleri, genellikle “deneme yanılma” yöntemiyle makul bir hesaplama süresi içinde bulmayı amaçlamaktadırlar. “sezgisel” kelimesi; “aramak” veya “bulmak” anlamlarına gelmektedir. Bu yöntemler, en uygun çözümü bulmayı garanti etmemektedirler. Kısmen iyi olan bir çözüm bulmaktadırlar ve genellikle yetersiz kalmaktadırlar. Sezgisel yöntemlere, yerel arama yöntemleri de denilmektedir. Parametre aramaları sezgisel yöntemlerle bulunmaktadır. Ayrıca zaman kısıtlarıyla yerel bölgelerde bulunan uygun çözümler, günümüzdeki birçok optimizasyon problemi için yeterli kabul edilmektedirler (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019).

Diğer bir yaklaşık yöntem olan metasezgisel algoritmalar, üst düzey sezgisel algoritmalar olarak bilinmektedir. “meta” kelimesi; “ötesi” veya “üst düzey” anlamlarına gelmektedir. Bu nedenle metasezgisel, sezgisel yöntemlerdeki deneme-yanılma süreçlerinin yanı sıra, çözümlerin üst düzey teknikler kullanarak bulunması anlamına gelmektedir. Başka bir ifadeyle, parametre aramalarında büyük alanın taranması için düşük seviyeli yöntemlerin birleştirilmesiyle elde edilen üst düzey teknikler olarak kabul edilmektedirler. “metasezgisel” kelimesi, Fidan Gelişim Algoritması (SGuA), Benzetimli Tavlama (SA), Genetik Algoritmalar (GA), Tabu Araması (TS), Karınca Kolonisi Optimizasyonu (ACO), Arı Algoritmaları (BA), Ateşböceği Algoritmaları (FA) ve Parçacık Sürü Optimizasyonu (PSO) gibi tüm modern üst düzey algoritmalarını ifade etmektedir. Bu yöntemler, klasik

sezgisel yöntemlere rehberlik etmektedirler (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaid, 2019). Optimizasyon Algoritmaları, Şekil 1.2'deki gibi sınıflandırılmaktadır.





Şekil 1.2 : Algoritmaların sınıflandırılması.

Genel olarak algoritma 1.1'deki yapı ve öğeler, tüm modern optimizasyon algoritmalarıyla eşleştirilebilir (Stork diğ., 2018).

Algoritma 1.1 : Optimizasyon Algoritmalarının Genel Sözde Kodu (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019)

- 1- **set** initial control paremeters
 - 2- **begin**
 - 3- $t=0$
 - 4- **initialize** candidate(s)
 - 5- **evaluate** initial candidate(s)
 - 6- **while** not termination-condition **do**
 - 7- $t=t+1$
 - 8- **variate** solutions to get new candidate(s)
 - 9- **evaluate** new candidate(s)
 - 10- **select** solution(s) for next iteration
 - 11- **optional**: update control parameters
 - 12- **end**
 - 13- **end**
-

Bir optimizasyon problemini çözmek için kullanılan algoritmalar, genelde beş adımdan oluşmaktadır (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019);

1. Başlangıç Popülasyonu Oluşturma
2. Uygunluk Hesaplama
3. Seçim
4. Yeni Nesil Oluşturma
5. Karar Verme

Başlangıç Popülasyonu Oluşturma; Çözüm uzayında bulunan bireylerden ilk çözüm adaylarının ve sayısının belirleme işlemidir. Birey sayısı ile ilgili bir kesinlik yoktur, probleme bağlı olarak değişmektedir (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019;

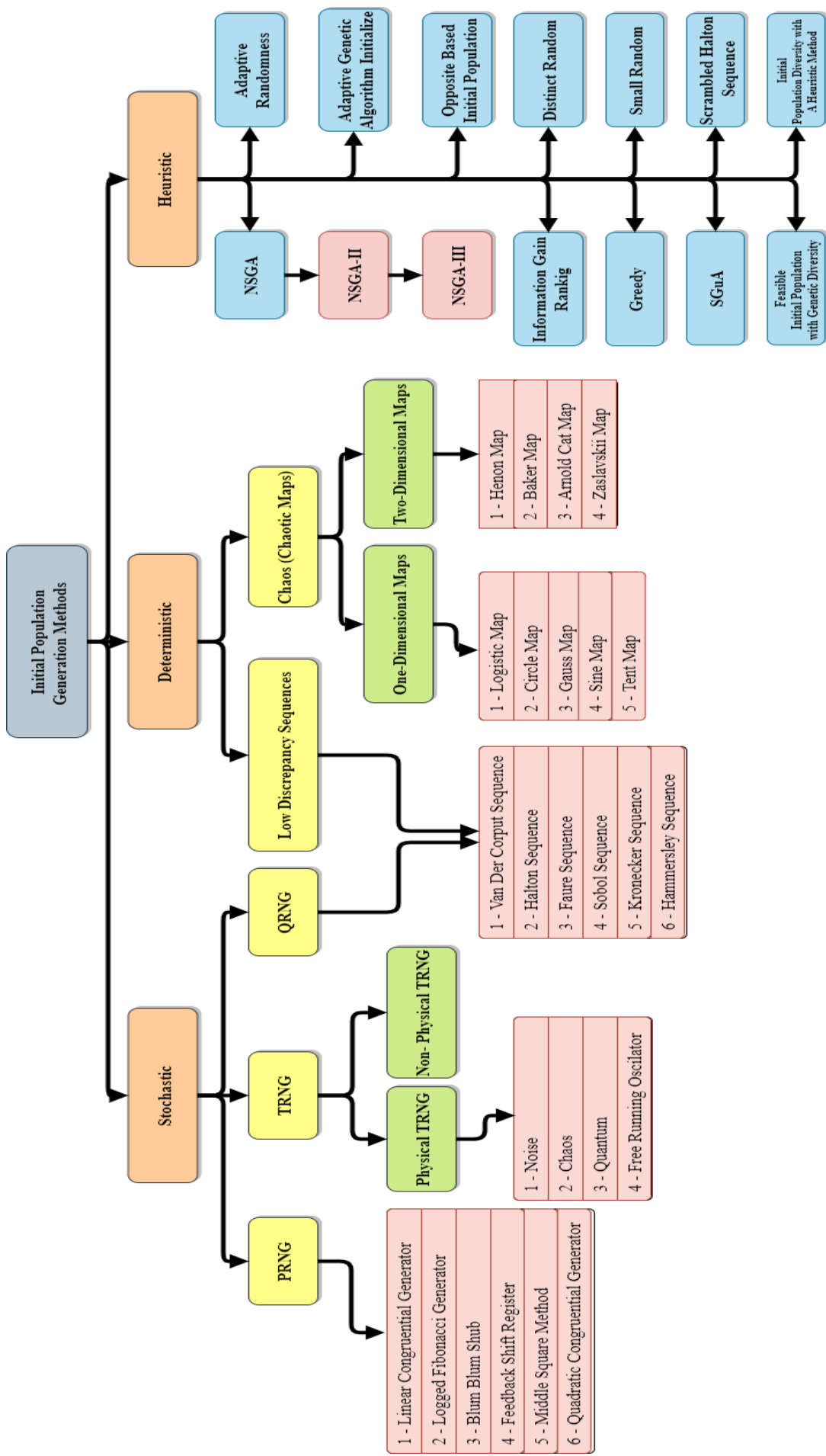
Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Pulat ve Kocakoç, 2017; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019).

Uygunluk Hesaplama; başlangıç popülasyonu adımıyla oluşturulan aday çözümlerin ne kadar iyi olduğunun belirleme işlemidir. Buna uygunluk değerinin hesaplanmasıyla karar verilmektedir (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Pulat ve Kocakoç, 2017; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019).

Seçim; uygunluk değerleri hesaplanan çözüm adayları arasında en yüksek uygunluğa sahip bireylerin seçilmesidir. Yeni Nesil Oluşturma; seçilen yüksek uygunluklu bireyler arasında bir takım işlemler yapılarak yeni bir neslin oluşturulmasıdır. Karar Verme; durdurma kriterine veya iterasyon sayısına göre önceki dört adımın çalıştırılma işlemidir (Collet ve Rennard, 2011; Dokeroglu ve diğ., 2019; Eröz ve Tanyildizi, 2018; Fausto ve diğ., 2020; Ponce-Ortega ve Hernández-Pérez, 2019; Pulat ve Kocakoç, 2017; Stork ve diğ., 2018; Türkay, 2013; Zaki ve Zaied, 2019). Problem çözümünde algoritmaların performanslarını iyileştirmek için, bu beş bölümde iyileştirmeler yapılmaktadır. Genellikle yeni nesil oluşturma adımı ile ilgili oldukça fazla çalışma bulunmaktadır. En az çalışma ise, başlangıç popülasyonu adımıyla yapılmıştır. Çünkü birçok çalışma rastgele başlatılmaktadır. Günümüzde bile en çok kabul gören yöntemdir (Kazimipour ve diğ., 2014).

Başlangıç popülasyonu ile yapılan çalışmaların amacı, rastgelelikten kısmen ya da tamamen kurtarmaktır. Çünkü rastgele yönteminde, çözüm uzayının tamamını kapsamama dezavantajı vardır. Çözüm uzayında homojen dağılım için çalışmalar yapılmıştır (Kazimipour ve diğ., 2014).

Bu tez kapsamında, başlangıç popülasyonu oluşturmada kullanılan yöntemler araştırılmıştır. Yapılan araştırmanın neticesinde Şekil 1.3'teki gibi, başlangıç popülasyonu oluşturma metotlarını üç ana başlık şeklinde ayırarak yeni bir kategorizasyon önerilmektedir.



Şekil 1.3 : Başlangıç popülasyonu oluşturma yöntemleri.

1.1 İlişkili Çalışmalar

1.1.1 Başlangıç popülasyonları üzerine yapılan çalışmalar

Kazimipour ve arkadaşları, 2014 tarihine kadar popülasyon başlatma teknikleri ile ilgili yapılan çalışmalar hakkında detaylı bir inceleme yapılmışlardır ve bu çalışmalarını toplaması açısından ilk olma özelliğine sahiptir. İncelenen çalışmalar temel alınarak başlatma teknikleri; rastgelelik, kompozisyonellik ve genellilik olmak üzere 3 ana kategoriye ayrılmıştır ve her bir ana kategori ise alt kategorilere ayrılmıştır (Kazimipour ve diğ., 2014).

Musrrat Ali ve arkadaşları, differential evolution (DE) algoritmasının başlangıç popülasyonunu iki ayrı yöntemle (Nonlinear Simplex DE (NSDE) ve quadratic interpolation DE (QIDE)) başlatarak temel DE ile yola devam etmektedirler. NSDE ve QIDE'nin, temel DE'ye göre CPU üzerindeki iyileştirme oranları sırasıyla %18 ve %14'tür (Ali ve diğ., 2013).

Kazemzadeh azad, çelik kafes boyutlandırma problemi için üç metasezgisel (adaptive dimensional search (ADS), modified big bang–big crunch (MBB-BC) and exponential big bang–big crunch (EBB-BC)) yöntem kullanmıştır. Uygulanabilir en hafif tasarımı elde etmek için optimizasyon yöntemlerinin ilk popülasyonu seçme işlemine odaklanmıştır. Optimizasyonların hesaplama sürelerinde iyileşme gözlemlenmiştir (Kazemzadeh Azad, 2018).

Zhang ve arkadaşları, Esnek Atölye Çizelgeleme Problemi (FJSP) için geçici üretim zamanını en aza indirmek istemişlerdir. Global Selection (GS), Local Selection (LS) ve Random Selection (RS) isimli üçlü bir karışım ile başlangıç popülasyonu oluşturulmaktadır. Etkin bir ilk popülasyonun başarı oranını pozitif yönde arttırdığı gözlemlenmiştir (G. Zhang ve diğ., 2011).

Tometzki ve arkadaşları, iki aşamalı stokastik karışık tamsayılı problemleri çözmeye kullanılan evrimsel algoritmalara üç tane başlatma yaklaşımı sunmaktadırlar. Birincisi klasik rastgele başlatma yöntemi, ikincisi Integer Relaxation (LP), üçüncüsü ise Second-Stage Integer Relaxation (2SLP) dir. Önerilen yöntemlerin başarılı olduğu gözlemlenmiştir (Tometzki ve Engell, 2011).

Dong ve arkadaşları, Çember tespit için yeni bir kaotik GA/PSO (Genetik Algoritma ve Parçacık Sürüsü Optimizasyonu) hibrit algoritması (CHA) önermektedirler. CHA'nın ilk popülasyonu için muhalefet temelli öğrenme (OBL) önerilmektedir. Yapılan çalışmaların

sonuçlarına bakıldığında, OBL'nin oldukça genel olduğu, algoritmaları hızlandırdığı ve çok çeşitli optimizasyon ve öğrenme algoritmalarında kullanılabileceği görülmektedir (Dong ve diğ., 2012).

1.1.1.1 Stokastik yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar

Budiman ve arkadaşları, n basamaklı sayıları (1.000.000.000'a kadar) iki saniyeden daha kısa sürede üretmek için orta kare yöntemi kullanmışlardır (Budiman ve diğ., 2020).

Rahimov ve arkadaşları, orta kare yöntemi ile üretilen sayıların benzersizliklerini arttırmak için kaotik haritalardan faydalanmaktadırlar. Sayıların benzersizliklerini ölçmek için kullanılan NIST testlerinde oldukça başarı elde edilmiştir (Rahimov ve diğ., 2011).

Mahmood ve arkadaşları, doğrusal eşlik üretici yöntemini kullanarak yeni bir simetrik anahtar şifreleme algoritması önermişlerdir. Önerilen yeni yöntem, yüksek güvenli sayılar üretmektedir fakat çok fazla hesaplama gerektirmektedir (Mahmood ve diğ., 2012).

Clawdia ve arkadaşları, kriptografinin bir uygulaması olan tek kullanımlık şifre (One Time Pad) üretmek için doğrusal eşlik üretici yönteminden faydalanmaktadırlar (Clawdia ve diğ., 2017).

Khairina ve arkadaşları, yüksek güvenli veri şifrelemek için RSA ve karesel eşlik üreticiden oluşan hibrid bir yeni yöntem önermişlerdir. Bu yeni yöntemle saldırılara karşı direnci %64'e çıkarmışlardır. Bu alanda oldukça iyi bir orandır (Khairina ve diğ., 2019).

Moussi ve arkadaşları, döviz kurları piyasasını takip etmek için kullanılan Asian Options Delta simülasyonu karesel eşlik üretici kullanarak gerçekleştirmişlerdir. Bu tür simülasyonlarda, lineer sayı üreticileri yerine ikinci dereceden sayı üreticilerinin etkili olduğu gözlemlenmiştir (Moussi ve diğ., 2013).

Srinivas Aluru, dağıtık bellekli paralel bilgisayarlarda gecikmeli fibonacci üretici yöntemi ile sayılar üretmiştir. Paralel olan işlemci sayıları ikinin kuvveti olduğunda LFG yönteminin başarılı olduğu gözlemlenmiştir (Aluru, 1997).

Delima ve arkadaşları, Nihilist Cipher düz metin şifreleme algoritması ve blum blum shub algoritmasını kullanarak yeni hibrid bir şifreleme algoritması önermişlerdir. Nihilist cipher'in anahtar eşleştirmedeki dezavantajını gidermek için eşleştirme sürecinde daha çok rastgeleliğe başvurmuşlardır. Bu rastgeleliği blum blum shub algoritması ile

sağlamışlardır. Önerilen yeni hibrid algoritmayla saldırılara karşı daha fazla tahmin edilemez bir yöntem elde etmişlerdir (Delima ve Arroyo, 2020).

Arroyo ve arkadaşları, Affine Cipher algoritması ve blum blum shub algoritması kullanarak yeni bir hibrid şifreleme algoritması önermişlerdir. Önerilen yeni yöntemle, şifrelenen özdeş metinlerin birbirlerine benzerliklerini ortadan kaldırmak amaçlanmaktadır (Arroyo ve Delima, 2020).

Kurniawan ve arkadaşları, Playfair Cipher algoritmasının dezavantajını gidermek için lineer geri beslemeli kaydırma yazmacından faydalanmaktadır. Playfair Cipher algoritmasının küçük harf ve sembollerin seçilememesini ortadan kaldırılması amaçlanmaktadır. Lineer geri beslemeli kaydırma yazmacı kullanarak şifrelemede kullanılan matris genişletilmiştir. Böylelikle küçük harfler ve sembollerin kullanılması sağlanmıştır (Kurniawan ve Priyatna, 2018).

Krishnapriya ve arkadaşları, iletilen imgelerin güvenliğini sağlamak için lineer geri beslemeli kaydırma yazmacıdan yararlanmıştır. Pikselleri satır ve sütun bazlı yeniden sıralamak için lineer geri beslemeli kaydırma yazmacı kullanılmıştır. Bu yeniden sıralamanın amacı, saldırganların sıralamadaki pikselleri tahmin edememesidir (Krishnapriya ve Smitha, 2017).

Todorov ve arkadaşları, hava kirliliği modellemesi oluştururken hava kirleticilerin duyarlılık analizlerinde Van Der Corput dizisi tabanlı bir optimizasyon kullanmışlardır. Hava kirliliği modellenmesinde çokça Sobol dizileri kullanılmaktadır. Bu çalışmada, hava kirliliği modellemesinde ilk defa Van Der Corput dizisi kullanılmıştır ve oldukça yüksek başarı sağlanmıştır (Todorov ve diğ., 2020).

Ohmer ve arkadaşları, yeraltı suları izleme ağlarının etkili tasarımı için Halton dizisinden faydalanmışlardır. Rastgele elde edilen sayılardan oluşturulan izleme ağları ile karşılaştırılması yapılmıştır. Halton dizisi ile oluşturulan eşit dağılımlı izleme ağından gelen verilerin doğruluğu oldukça yüksek çıkmaktadır (Ohmer ve diğ., 2019).

Kulkarni ve arkadaşları, kablosuz sensör ağlarının en büyük zorluklarından olan sensör düğümlerinin konumlandırılmasında Halton, Faure ve Sobol dizilerinden faydalanmışlardır. Düzenli sayılarla konumlandırılan sensörler, ağın kapsamını ve ömrünü arttırmaktadır. Kapsamı artan ağlarda kapsama boşlukları meydana gelmemektedir ve böylelikle enerji verimliliği sağlanmaktadır (Kulkarni ve diğ., 2018).

Naik, LCG ve Faure dizisini birleřtirerek yeni bir rastgele sayı üretici önermiřtir. Bu yeni üretçe ile üretilen sayıların rastgelelikleri istatistiksel testleri geçmektedir. Bu sayılarla Monte Carlo simülasyonlarında daha yüksek boyutlu problemler çözülebilir (Naik, 2020).

Mehrdoust ve arkadaşları, lineer homojen integral denklemlerin özdeğerlerini hesaplamak için Faure dizisi ve Monte Carlo yönteminden faydalanmışlardır. Çalışmadaki amaç, hesaplama süresini kısaltmaktır. Saf Monte Carlo yönteminden daha hızlı sonuçlar elde edilmiştir (Mehrdoust ve diğ., 2013).

Liu ve arkadaşları, stokastik hesaplama devrelerinde enerji verimliliğini arttırmak için Sobol dizinden faydalanmaktadır. Kullanılan bir çarpım devresinde, Sobol dizisinin hızlı sayı üretme yeteneğinden faydalanılarak daha kısa sürede sonuçlar elde edilmiştir. Yani, daha az enerji ile sonuçlar üretilmektedir (S. Liu ve Han, 2017).

Timotej Vesel, mühendislik, fizik ve ekonomi gibi alanlarda zor problemlerin çözümlerine ulaşmak için kullanılan Monte Carlo simülasyonunda Sobol dizi kullanmıştır. Bu tür simülasyonlar, problemi örnelemek için kullanılmaktadır. Sobol dizisinin doğrusal zamanda üretilmesinden dolayı simülasyonun sonuçlanması daha erken olmaktadır (Vesel, 2019).

Munteanu ve arkadaşları, diferansiyel dinamik mantık (dL) algoritmasının simülasyonunda Kronecker dizisinden faydalanmışlardır. Bu simülasyonun amacı, karaciğer tümörlerinin cerrahi operasyonlarında cerrah ile robot arasındaki iletişimi sağlayan hibrid sistemleri örnekmektir. Hibrid sistemler, sürekli hareketler (cerrah gibi) ile ayrık hareketler (robot gibi) arasında iletişimi sağlamaktadır. Robotun yapabileceği yörüngeler, değiştirilmiş Kronecker dizi ile çizilmektedir. Böylelikle damar hasarları ve kanama en aza indirgenmektedir (Munteanu ve diğ., 2018).

Girip ve arkadaşları, araçların yollarda maruz kaldıkları engellerin ve bu engeller yüzünden oluşabilecek salınımların değiştirilmiş Kronecker dizileri kullanarak simülasyonunu yapmışlardır. Daha düşük maliyetle sanal gerçeklik ortamının oluşturulması amaçlanmaktadır (Girip ve Munteanu, 2015).

Chen ve arkadaşları, ekonomisinin büyümesiyle artan sanayileşmenin sonucunda çevre sorunları ile karşılaşan Çin için çok amaçlı bir hibrid optimizasyon yöntemi önermişlerdir. Bu çevre sorunlarını gidermek için son yıllarda güneş ve rüzgâr enerji kaynaklarının kullanımını artmaktadır. Bu iki kaynağı kullanarak hibrid bir dağıtımli üretim sistemi

amaçlanmıştır. Fakat bu iki kaynağın istikrarsızlığı ve sürekli olmaması dezavantaj olarak görülmektedir. Bu çalışmada, bu iki kaynağın yanına yakıt hücresi de eklenmektedir. Önerilen üç kaynaklı dağıtımli sistemin tasarımında karar vericiler için önemli bir çözüm yolu sunulmaktadır. Hammersley dizisi kullanılarak oluşturulan çok amaçlı optimizasyon yönteminin amaçları elektrik verimliliği, sistemin maliyeti ve enerji tedarik güvenilirliği olarak tanımlanmaktadır. Bu amaçlar doğrultusunda problemin çözümü için önerilen optimizasyon algoritmasında başarılı sonuçlar elde edilmiştir (Chen ve., 2017).

Müller, bellek erişimini gürültü kaynağı olarak kullanıp CPU zaman değişimi tabanlı yeni bir fiziksel olmayan TRNG önermiştir. Bellek erişiminden bir zaman deltası elde edilmektedir. Yapılan çalışmada 64 bitlik gerçek rastgele sayılar üretilmektedir. Birçok istatistiksel testi başarıyla geçmiştir (Müller, 2014).

Lee ve arkadaşları, uygulanabilirliği yüksek ve ucuz bir TRNG geliştirmek için FM radyo sinyallerinden faydalanmaktadır. Radyo yayını yapan cihaza bağlı bir hoparlörden çıkan ses sinyalleri mikrofon yardımıyla kayıt edilmektedir. Kayıt edilen sinyaller, FM radyo sinyallerinin doğası gereği fazlaca çevre şartlarından etkilenmesi sebebiyle birçok gürültü barındırmaktadır. Bu gürültü kullanılarak üretilen sayılar, diğer gürültü bazlı TRNG'lerle üretilen sayılardan %431'e kadar daha yüksek entropili olmaktadır. Gerçek rastgele sayılar üretilmesi bakımından oldukça yüksek bir orandır (Lee ve Lee, 2019).

Siswanto ve arkadaşları, günümüzde popülaritesi oldukça yüksek olan nesnelerin interneti (IoT) alanında şifreleme işlemlerinde kullanılması için bir kuantum sayı üretici önermişlerdir. Önerilen yöntem fotonic-based'tir ve optik bileşen, analog-dijital elektronik sistem ve asenkron verici olmak üzere üç modülden oluşmaktadır. Oluşturulan elektronik sistem optik bileşenin ürettiği analog sinyallerini dijital rastgele verilere dönüştürmektedir. Üretilen sayıların rastgeleliği NIST istatistiksel testleri ile kanıtlanmıştır. Test sonuçlarında, üretilen sayıların %99 güvenli rastgele sayılar olduğu gözlemlenmiştir (Siswanto ve Rudiyanto, 2017).

Bird ve arkadaşları, başlangıç popülasyonu olarak rastgele sayıları girdi alan birçok makine öğrenme problemi için PRNG ve kuantum sayı üreticileri kullanmışlardır ve sonuçları karşılaştırmışlardır. PRNG ve kuantum sayı üreticilerinin kıyaslanması Accent ve EEG sınıflandırmalarında, MNIST ve CIFAR-10 sorunlarında yapılmıştır. Kuantum sayı üreticilerinin çok olmamakla birlikte PRNG'lerden üstün olduğu gözlemlenmiştir. Ayrıca

Rastgele Ağaç'dan farklı yeni Kuantum Rastgele Ağaç (Quantum Random Tree, QRT) modeli önerilmiştir (Bird ve diğ., 2020).

Marghescu ve arkadaşları, dört halka osilatörü kullanarak bir TRNG önermişlerdir. Önerilen yöntemin amacı, düşük maliyet ve hızlı sayı üretimidir. Önerilen sayı üreticinin güvenilirliğini ispatlamak için istilacı olmayan (non-invasive) iki ayrı set (elektromanyetik alan ve farklı sıcaklık aralıkları) ile test edilmiştir. Bu testler sonucunda önerilen yöntemin güvenilir ve kararlı olduğu kabul edilmektedir (Marghescu ve diğ., 2016).

Liu ve arkadaşları, akıllı kartlarda şifreleme işlemleri için düşük maliyetli ve düşük güç tüketimine sahip osilatör tabanlı yeni bir TRNG önermişlerdir. Dört yüzlü (tetrahedral) bir halka osilatörü kullanılarak yapılmıştır. Bu TRNG'den üretilen sayıların rastgeleliklerini arttırmak için devre sonuna bir işlemci eklenmiştir. Üretilen sayılarla yapılan şifrelemeler akıllı kartlara başarıyla uygulanmıştır. Ayrıca önerilen yöntem NIST ve Diehard testlerini de başarıyla geçmiştir (D. Liu ve diğ., 2016).

1.1.1.2 Deterministik yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar

Luo ve arkadaşları, Lojistik haritanın yüksek karmaşıklığa sahip olması için iki kademeli yöntem önermişlerdir. İlk kademesinde Lojistik haritanın parametre kontrollerini sağlamak ve karışıklığı arttırmak için yüksek karmaşıklığa sahip iki boyutlu Baker haritasından faydalanmaktadırlar. İkinci kademesinde ise Lojistik haritayı kullanmaktadırlar. Bu hibrid model ile yeni bir imge şifreleme algoritması da önerilmektedir. Önerilen imge şifreleme algoritması, birçok istatistiksel testi başarılı şekilde geçmektedir ve yüksek güvenli seviyeye sahip olmaktadır (Luo ve diğ., 2019).

Moysis ve arkadaşları, Lojistik haritanın karmaşıklığını arttırmak için değiştirilmiş bir versiyonunu önermişlerdir. Haritanın değerleri ile ondalık kısımlarının çarpılmasından elde edilen yeni değerler oldukça karmaşık olmaktadır. Önerilen yeni yöntem yüksek Lyapunov üslerin elde edilmesine olanak sağlar. Ayrıca 15 Nist testini geçmektedir ve tatmin edici anahtar aralığı sunmaktadır (Moysis ve diğ., 2020).

Nair ve arkadaşları, Dikgen Frekans Bölmeli Çoğullama (Orthogonal Frequency Division Multiplexing, OFDM) için iyi bir serpiştirici geliştirmek amacıyla Çember haritasından faydalanmaktadır. OFDM iletişim sistemleri kablosuz kanallar üzerinden veri iletimini yapmaktadırlar. Bu sistemlerin en büyük sorunları patlama hataları olmaktadır. Patlama hatalarını rastgele hatalar gibi gösterip etkisini azaltmak için serpiştiriciler kullanılmaktadır.

İyi bir serpiştiricinin sahip olması gereken iki temel özellik, yayılma etkisi ve rastgeleliktir. Çember haritasının kullanılma sebebi, serpiştiricinin rastgeleliliğini arttırmaktır (Nair ve Kiasaleh, 2016).

Kordov ve arkadaşları, seslerin iletiminde gerekli güvenliği sağlamak için yeni bir şifreleme algoritması önermişlerdir. Şifreleme algoritmasını oluştururken iki Çember haritadan faydalanmaktadır. Orijinal ses verisi ile bu Çember haritalarından elde edilen iki dizi XOR işleminden geçirilmektedir. Elde edilen şifreli dosya güvenlik testlerinde oldukça başarılı olmaktadır. Bu sebeple, Çember haritası ile oluşturulan yeni şifreleme algoritması kriptografik saldırılara karşı tavsiye edilmektedir (Kordov ve Bonchev, 2017).

Koyuncu, Parkinson hastalarının sınıflandırılması için Gauss haritasına dayalı kaotik bir parçacık sürü optimizasyon yöntemi önermektedir. Parkinson hastalarının sınıflandırılması için ses kayıtlarına bakılmaktadır. Önerilen yöntem iyi bilinen iki tane parkinson hasta veri setine uygulanmış ve çıkan sonuçlar başka yöntemlerin sonuçları ile karşılaştırılmıştır. Doğruluk, Eğri Altındaki Alan (Area Under Curve, AUC), Duyarlılık, Özgünlük, g-mean, Kesinlik ve F-Ölçümü gibi yedi değerlendirme kriterine göre kıyaslanmıştır. Önerilen yöntemin oldukça başarılı olduğu gözlemlenmektedir (Koyuncu, 2019).

Sahay ve arkadaşları, renkli görüntüler için yeni bir şifreleme algoritması önermişlerdir. Önerilen şifreleme algoritmasında Gauss haritası kullanılmaktadır. Renkli görüntünün pikselleri karıştırıldıktan sonra Gauss haritasından üretilen sayılarla XOR işleminden geçirilmektedir. Diğer kaotik haritalardan daha başarılı olduğu gözlemlenmektedir (Sahay ve Pradhan, 2018).

İrani ve arkadaşları, imge şifreleme ve karıştırma işlemleri için Sinüs haritasına dayalı 1-boyutlu yeni bir kaotik harita önermişlerdir. Sinüs ve diğer tek boyutlu haritaların sahip olduğu kısa anahtar alanları dezavantajlarını gidermek için bu yöntemi önermişlerdir. Sinüs haritasının doğası korunarak yeni kontrol parametreleri eklenilmiştir ve yeniden formülize edilmiştir. Böylelikle karmaşıklığı arttırılmıştır. Önerilen yöntem birçok uluslararası veri seti üzerinde test edilmiştir. Karıştırma hızının yüksek olması ve pikseller arası korelasyonun düşük olmasından dolayı şifreleme algoritmalarında etkin bir yöntemin önerildiği gözlemlenmektedir. Ayrıca kalite ölçüm yöntemlerinden elde edilen iyi sonuçlar yöntemin güvenilir olduğunu kanıtlamıştır (Yosefnezhad Irani ve diğ., 2019).

Usama ve arkadaşları, blok şifreleme algoritmalarında kullanılan Yer Değiştirme Kutuları (Substitution box, S-box) için sinüs tabanlı yeni bir yöntem önermişlerdir. S-box'lar

şifreleme algoritmalarında karıştırma işlemi yapmaktadırlar. S-box'ların en büyük problemi olan statik tabloları dinamik hale çevirmek için kaotik Sinüs haritasından faydalanmaktadırlar. Sistemin dinamikliği şifreleme algoritmalarında etkinlik sağlamıştır. Ayrıca önerilen yöntem SET test sistemi ile test edilmiştir ve kriptografik özellikleri iyi olduğu gözlemlenmiştir (Usama ve diğ., 2019).

Gao ve arkadaşları, Harris Hawk optimizasyon algoritmasının etkinliğini arttırmak için çadır haritasını kullanmışlardır. Tüm kıyaslama fonksiyonlarının çözümünde orijinal Harris Hawk optimizasyon algoritmasından daha iyi performans gösterdiği gözlemlenmiştir (Gao ve diğ., 2019).

Vishwas ve arkadaşları, imge şifrelemek için önerdikleri şifreleme algoritmasında Çadır haritasını kullanmışlardır. Önerilen yöntemde imgenin satır ve sütunlarını karıştırarak karmaşıklığı arttırmayı amaçlamışlardır. İmgelerde dört değişik şekilde (yalnızca x, yalnızca y, önce x sonra y, önce y sonra x koordinatına göre) karıştırma işlemi yapılarak şifreleme ve şifre çözme yöntemleri sunulmaktadır. Bu dört durum hesaplama karmaşıklığı getirmektedir. Bu da daha güvenilir şifrelerin oluşturulmasına olanak sağlamaktadır (Vishwas ve Kunte, 2020).

Roy ve arkadaşları, ses sinyallerinin şifrenmesi için Henon haritası kullanarak yeni bir şifreleme yöntemi önermişlerdir. Önerilen yöntem ön işleme ve şifreleme adımları olmak üzere iki aşamadan oluşmaktadır. İlk adımda ses sinyallerinin dijital verilere dönüşmesi sağlanmaktadır. İkinci adımda ise şifrelemede kullanılacak anahtar üretimi yapılmaktadır ve bu anahtarların gizlenmesi için yeni algoritmalar çalıştırılmaktadır. Anahtar alanları yüksek olduğu için saldırılara karşı güvenli olduğu vurgulanmaktadır. Korelasyon, entropi ve anahtar uzay analizi gibi çeşitli istatistiksel testlerle anahtarların güvenli olduğu kanıtlanmıştır (Roy ve Misra, 2017).

Meranza-Castillón ve arkadaşları, şifreleme algoritmaları için kaotik tabanlı bir PRNG önermişlerdir. Kaotik haritalardan Henon haritasının geliştirilmiş bir versiyonu kullanılmaktadır. Gelişmiş Henon versiyonu yüksek karmaşıklığa sahiptir. Önerilen PRNG'nin ilk adımında gelişmiş Henon ile başlangıç koşulları belirlenmektedir. İkinci adımda ilk adım n kez tekrar edilmektedir. Son adımda ise üretilen diziler (0,1)'den [0,255]'e dönüştürülmektedir. Yapılan çalışma hem yazılım (matlab) hem de donanım (Alanda Programlanabilir Kapı Dizileri, FPGA) olarak gerçekleştirilmiştir. İki uygulama da istatistiksel testleri başarılı bir şekilde geçmiştir (Meranza-Castillón ve diğ., 2019).

Liu ve arkadaşları, Baker haritası kullanarak yeni bir imge şifreleme algoritması önermişlerdir. Klasik kaotik haritalarla anahtar üreten şifreleme algoritmaları çok fazla öngörülemez değildir. Çünkü sabit bir başlangıç değeri ile üretilmektedir. Bu sebeple zamanla değişebilen (sabit olmayan) bir Baker haritası ile şifreleme algoritması oluşturulmuştur. Bu yöntem parametre değişkenli Baker haritası (PVBM) denilmektedir. PVBM ile üretilen sayı dizileri sabit değildir. Bu adımdan sonra karıştırma matrisi ve yer değiştirme matrisi olmak üzere iki tür matris oluşturmak için sabit olmayan sayı dizileri kullanılmaktadır. Bu iki matris kullanılarak imge şifreleme algoritması oluşturulmaktadır. Çeşitli istatistiksel testlerden başarı ile geçmiştir (Liu ve Miao, 2017).

Avaroğlu, Arnold Cat haritasını kullanarak yeni bir PRNG yöntemi önermiştir. Önerilen yöntem üç aşamadan oluşmaktadır. Birinci aşamada iki Arnold Cat haritası kullanarak sayı dizileri üretilmektedir. İkinci aşamada Arnold Cat haritalarından üretilen sayı dizileri ayrı ayrı birbirleri ile karşılaştırılmaktadır. Son aşamada ise birinci Arnold Cat haritasındaki bit 1 ise ikinci Arnold Cat haritasındaki bit çıkış olarak kabul edilmektedir, 0 ise boş geçilmektedir. Böylelikle örnekleme yapılmaktadır. Üretilen sayı dizilerin rastgelelikleri NIST test paketi, ölçek indeksi yöntemi, istatistiksel karmaşıklık ölçümleri ve otokorelasyon gibi istatistiksel analizlerle doğrulanmıştır (Avaroglu, 2017).

Istwal ve arkadaşları, 802.11a/g Kablosuz Sensör Ağ'larda (Wireless Sensor Network) iletilen verilerin güvenliğini arttırmak için Arnold Cat haritasını kullanarak yeni bir şifreleme algoritması önermişlerdir. Burada değiştirilmiş bir Arnold Cat haritası kullanılmıştır. Bu değiştirilmiş harita ile üretilen diziye orijinal veri dizi yerleştirilmektedir. Ayrıca karmaşıklığı arttırmak için L-Değer'de (depolama bölgesinin değerini temsil etmektedir) eklenmektedir. Son olarak bitsel xor uygulanmaktadır. Önerilen yöntemle kırılması çok zor olan ve karmaşıklığı yüksek şifreli veriler üretilmektedir. Orijinal veri ile şifrelenmiş veri arasındaki MSE (Mean Squared Error) değeri 0.0521 ve Tepe Sinyal-Gürültü Oran'ı (Peak Signal to Noise Ratio, PSNR) değeri ise 97.1184 olarak elde edilmiştir. Bu değerler mükemmel yakın değerlerdir (Istwal ve Verma, 2018).

Hamza ve arkadaşları, 2 boyutlu Zaslavskii haritasını kullanarak yeni bir imge şifreleme yöntemi önermişlerdir. İki aşamadan oluşan yöntemin istatistiksel test sonuçları (NPRC =% 99.61, UACI =% 33.47, entropi (CipherImage)=8 ve korelasyon katsayısı=0) oldukça başarılı olduğu gözlemlenmiştir. İlk aşamada Zaslavskii haritası ile şifreleme yönteminin anahtarları üretilmektedir. İkinci aşamada üretilen anahtarlara göre imgenin piksel değerleri

karıştırılmaktadır. İkinci aşamada permütasyon-difüzyon süreçleri de kullanılmıştır. Önerilen şifreleme yöntemi saldırılara karşı oldukça başarılı performans göstermektedir (Hamza ve Titouna, 2016).

Farsana ve arkadaşları, güvenli olmayan ortamlara aktarılması olası olan ses verileri için yeni bir şifreleme algoritması önermişlerdir. Önerilen yöntemde ilk olarak ayrık kosinüs dönüşümü ile veriler sıkıştırılmaktadır. Verilerin daha az anlaşılır olmasını sağlayan sıkıştırma işleminden sonra Zaslavskii haritası kullanılarak veriler gizlenmiştir. Son olarak karmaşıklığı arttırmak için Arnold Cat haritası da ilaveten uygulanmıştır. Korelasyon, Sinyal-Gürültü Oranı (SNR), anahtar boşluğu ve kaba kuvvet saldırısı gibi birçok analiz yapılmıştır. Yüksek boyutlu haritalarla kıyaslandığında önerilen yöntem hem basit hem de verimli bir yapıya sahip olduğu gözlemlenmektedir (Farsana ve Gopakumar, 2016).

1.1.1.3 Sezgisel yöntemlerle üretilen başlangıç popülasyonları ile ilgili çalışmalar

Saraç ve arkadaşları, permütasyon akış tipi çizelgeleme probleminin çözümünde yeni bir Baskın Olmayan Sıralı Genetik Algoritma-II yöntemi önermişlerdir. Toplam gecikme, toplam erken tamamlanma süresi ve son işin tamamlanma zamanı olmak üzere üç amaç fonksiyonunun minimizasyonu hedeflenmektedir. Önerilen yöntemin başarısını kanıtlamak için rastgele oluşturulan popülasyon ile de uygulama yapılmaktadır. Birçok test fonksiyonunda Baskın Olmayan Sıralı Genetik Algoritma-II yöntemi oldukça başarılı sonuçlar elde etmektedir (Saraç ve Bilgiçer, 2020).

Rahnamayan ve arkadaşları, diferansiyel evrim algoritması için iki farklı başlangıç popülasyon yöntemi uygulamış ve sonuçlar analiz edilmiştir. İlk yöntem klasik rastgele başlatma tekniğidir (DEr). İkincisi yöntem ise sezgisel bir yaklaşım olan muhalefet tabanlı öğrenme algoritması kullanılmıştır (DEo). Sonuçları test etmek için literatürde iyi bilinen 34 farklı kıyaslama fonksiyonu (sepher, Ackley, Rosenbrock vb.) kullanılmıştır. DEo ile başlatılan diferansiyel evrim algoritması 34 fonksiyonun 30'unda yüksek performans göstermektedir. Sonuç olarak rastgele başlatma yönteminin yerine muhalefet tabanlı başlatma yöntemi önerilmektedir (Rahnamayan ve diğ., 2007).

Pan ve arkadaşları, popülasyon temelli algoritmalar için yeni bir başlangıç popülasyonu oluşturma yöntemi önermişlerdir. Bu yöntemin adı Uyarlanabilir Rastgelelik'tir. Başlangıç popülasyonunun dağılımının eşit olmasını sağlayan bu metot aslında klasik rastgele yönteminin gelişmiş bir versiyonudur. Bu çalışmada önerilen yöntemin yanı sıra üç tane başka yöntem de (rastgele başlatma, muhalefet tabanlı başlatma ve genelleştirilmiş muhalefet tabanlı

başlatma) kullanılmıştır. Dört tane başlangıç popülasyonu oluşturma metodu Diferansiyel Evrim (DE) algoritmasında kullanılmıştır ve sonuçlar karşılaştırılmıştır. İyi bilinen 34 kıyaslama fonksiyonu ile test edilmiştir. Yakınsama hızı bakımından önerilen başlangıç yöntemi, diğer üç yöntemden %64.7 daha başarılı olmuştur. Çözüm kalitesi bakımından ise yine diğer üç yöntemden %70.5 daha başarılı olduğu gözlemlenmektedir (Pan ve diğ., 2014).

Cruz-Chávez ve arkadaşları, Araç Rotalama Problemi (ARP) çeşitlerinden zaman pencereci ARP için performans yükseltici bir çalışma gerçekleştirmişlerdir. Zaman kısıtlı olan bu ARP'nin performansını optimum seviyeye yükseltmek için uyarlanabilir başlangıç popülasyonu ile genetik çeşitlilik artırılmaktadır. 100 müşteri örneklili bir uygulama üzerinde yapılan bu çalışma Solomon'un 56 tane ölçütü kullanılarak test edilmektedir. 5 ayrı ekleme sezgiseli kullanılmıştır ve sonuçları analiz edilmektedir. Popülasyon temelli problemlerde çeşitliliği arttırmak, performansı yükselttiği gözlemlenmiştir. Ayrıca 4 ekleme sezgiselinin birleşiminden oluşan hibrid ekleme sezgiseli, diğer dördünden yüksek performans elde ettiği görülmektedir (Cruz-Chávez ve Martínez-Oropeza, 2016).

Pian ve arkadaşları, başlangıç popülasyonunun çeşitliliğini arttırmak ve bunun sonucunda kaliteli çözümler üretmek, yerel optimum noktalardan kaçınmak ve çözüme yakınsama zamanını azaltmak amacıyla sezgisel bir algoritma olan Yapay Arı Algoritmasının (ABC) klasik rastgele başlatma yöntemi yerine sezgisel bir yaklaşım önermişlerdir. Yapılan çalışmada klasik ABC'ye bir başka öneri de sunulmaktadır. Çapraz işleme dayalı komşuluk arama yöntemi adında iki farklı yöntem bir araya getirilmektedir. Bu hibrid komşuluk arama yöntemi ile global optimum noktasına yakınsama hızını yükseltmektedir. Sonuçlar en iyi bilinen altı test fonksiyonu ile kıyaslanmaktadır. Kıyaslama sonucunda popülasyon çeşitliliğinin ne denli yüksek performans gösterdiği gözlemlenmektedir (Pian ve diğ., 2018).

Deniz ve arkadaşları, 5 farklı başlangıç popülasyonu oluşturma tekniklerinin (Klasik Rastgelelik (Rnd), Farklı Rastgelelik (Dr), Bilgi Kazancı Sıralaması (IGR), Aç Gözlü Yaklaşım (Gr), Küçük Rastgelelik (Sr)) sonuçlara etkisini analiz etmişlerdir. Ayrıca başlangıç popülasyonu oluşturmada önce Çok Amaçlı Öğretim Öğrenme Tabanlı Optimizasyon (MTLBO-MD) ile kullanılan veri setlerinde özellik düşürülmesi yapılmıştır. Özellik seçimi adımıdaki amaç; gereksiz özellikleri eleyerek en iyi sonucu bulmak ve işlem süresini azaltmaktır. On iki adet veri seti boyutları dikkate alınarak üç kategoriye (küçük, orta ve büyük) ayrılmıştır. Ayrıca üç farklı (Lojistik Regresyon (LR), Destek Vektör Makineleri (SVM) ve Aşırı Öğrenme Makinesi (ELM)) algoritma ile sonuçlar

karşılaştırılmıştır. Küçük boyutlu verilerde tüm başlangıç popülasyonu oluşturma metotları (Rnd, Dr, IGR, Gr, Sr) makul sürede en iyi sonuçlar vermektedir. Verilerin boyutları büyüdükçe Rnd, Dr ve Sr yöntemlerinin başarısı azalmaktadır. IGR'nin performansı da Gr'nin performansından yüksektir. Sonuç olarak büyük ölçekli veri setlerinde IGR yöntemi önerilmektedir (Deniz ve Kiziloğlu, 2019).

Digehsara ve arkadaşları, PSO'nun başlangıç popülasyonunu oluşturmak için karıştırılmış Halton dizisi kullanmışlardır. Bu hibrid algoritmaya Halton-PSO adı verilmektedir. Çalışmada başlangıç popülasyonu oluşturduktan sonra standart PSO uygulanmıştır. 11 tane kıyaslama fonksiyonu kullanılmıştır ve 7 tane doğrusal olmayan mühendislik problemine (Dişli Zincir Tasarımı, Basınçlı Kap Tasarımı, Germe/Sıkıştırma Yay Tasarımı, I-Kirişin Dikey Sapmasının Minimizasyonu, Hız İndirgeyici Tasarım, Üç Çubuklu Kafes Tasarımı, Kaynaklı Kiriş Tasarımı) uygulanmıştır. Sonuçlar, diğer metasezgisel algoritmalar PSO (Kennedy ve Eberhart, 1995), HPSO-TVAC (Ratnaweera ve diğ., 2004), HEPSO (Mahmoodabadi ve diğ., 2014), WOA (Mirjalili ve Lewis, 2016), DA (Mirjalili, 2016), SCA (Mirjalili, 2016), GA (Coello, 2000), ALO (Mirjalili, 2015a), MFO (Mirjalili, 2015b) and MVO (Mirjalili ve diğ., 2016) ile kıyaslanmıştır. Diğer algoritmalara göre Halton-PSO'nun başarılı olduğu gözlemlenmiştir. Ancak PSO'ya uygulanan karıştırılmış Halton dizisi ile genel minimuma yakınsama veya lokal minimuma yakalanmama konularında herhangi bir başarı elde edilememiştir. Bunu çözmek için PSO'nun sömürü aşamasına Halton dizisi uygulanması önerilmektedir (Digehsara ve diğ., 2020).

Demir ve arkadaşları, DNA dizilerindeki motifleri bulmak için Fidan Gelişim Algoritması kullanmışlardır. TRANSFAC veri tabanından alınan dört farklı DNA dizisi (dm01g.fasta, dm01r.fasta, mus05r.fasta ve hm15r.fasta) kullanılmaktadır. Motif keşfi için daha önceden AlignACE, MEME, MEME3, MotifSampler, Consensus ve Weeder gibi çalışmalar yapılmıştır. Yapılan çalışmada elde edilen sonuçlar ile önceki çalışmalarda elde edilen sonuçların karşılaştırılması yapılmaktadır. Fidan Gelişim Algoritması'nın diğer yöntemlere göre başarılı olduğu gözlemlenmektedir (Demir ve diğ., 2011).

1.1.2 Optimizasyon algoritmaları üzerine yapılan çalışmalar

Vani ve arkadaşları, Hint Pines veri seti üzerinde Konvolüsyonel Sinir Ağı (CNN) yöntemini kullanarak sınıflandırma yapmışlardır. CNN'deki hatayı düşürmek için SGD, RMSProp, Adam, Adamax, Adagrad, Adadelta ve Nadam gibi optimizasyon yöntemleri

kullanılmaktadırlar. %99,58 başarı oranı ile Adamax yöntemi en iyi sonucu vermiştir (Vani ve Rao, 2019).

Rajakumaran ve arkadaşları, Servis Hizmet Reddi (Denial of Service, DoS) saldırılarının tespiti için çok değişkenli lineer regresyon kullanmışlardır ve MSE hata fonksiyonundan yararlanmışlardır. %3,3 olan hata oranını SGD optimizasyon yöntemi kullanarak %0,3'e düşürmüşlerdir (Rajakumaran ve diğ., 2020).

Dogo ve arkadaşları, 3 ayrı veri sete (Cats and Dogs, Fashion Mnist, Natural Images) CNN uygulamışlardır. Hata oranını azaltmak için 7 ayrı türeve dayalı optimizasyon algoritması (SGD, vSGD, SGDm, SGDm+n, RMSProp, Adam, AdaGrad, AdaDelta, Adamax, Nadam) kullanmışlardır. En iyi sonucu veren Nadam, en kötü sonucu veren ise Adadelata olmuştur (Dogo ve diğ., 2018).

Yazan ve arkadaşları, parametre güncelleme için 5 ayrı Dik İniş tabanlı optimizasyon yöntemi (Momentum, AdaGrad, RMSProp, AdaDelta, Adam) kullanmaktadır. Bu yöntemler Griewank fonksiyonunda test edilmiştir. Adadelata en başarılı yöntem olarak gözlemlenmiştir (Yazan ve Talu, 2017).

1.1.3 Hata fonksiyonları üzerine yapılan çalışmalar

Parnianifard ve arkadaşları, Oransal-İntegral-Türevsel (Proportional-Integral-Derivative, PID) kontrolörü ayarlama için kullanılan parametrelerin optimum değerlerini bulmak için yeni bir yöntem önermişlerdir. IAE, ITAE, ITSE (zaman ağırlıklı hata karelerinin toplamı) ve ISE amaç fonksiyonları kullanılmıştır. ITAE ve ITSE'nin sonuçları ile IAE ve ISE'nin sonuçlarının aynı olduğu gözlemlenmektedir (Parnianifard ve diğ., 2018).

Ansari ve arkadaşları, PID kontrolörünün parametrelerini bulmak için Karınca Koloni Optimizasyonu (Ant Colony Optimization) algoritmasını kullanmışlardır. Yapılan çalışmada 5 farklı amaç fonksiyonunun (IAE, ITAE, ITSE, MSE ve ISE) sonuçları analiz edilmiştir. ITAE'nin başarı oranının yüksek olduğu gözlemlenmektedir (Ansari ve diğ., 2014).

Amador-Angulo ve arkadaşları, bulanık kontrolörlerin üye fonksiyonlarındaki parametrelerin optimum değerlerini bulmak için hibrid bir yaklaşım önermektedirler. Uygulamada amaç fonksiyonu olarak ITAE, IAE, ISE, ITSE, RMSE (Ortalama Hata Kare Kökü) ve MSE hata fonksiyonları kullanılmaktadır. MSE'nin performansının yüksek olduğu gözlemlenmektedir (Amador-Angulo ve diğ., 2016).

1.1.4 Çok deęişkenli lineer regresyon üzerine yapılan çalışmalar

Kakchapati ve arkadaşları, Tayland'da halen büyük bir sağlık sorunu olan tüberkülozun hastalık, ölüm ve sakatlık oranlarındaki etkisini modellemek ve tahmin etmek için MLR yöntemi kullanmışlardır. Çalışmanın sonunda elde edilen faydalı bulgular sayesinde çeşitli sağlık programları geliştirilmektedir (Kakchapati ve Choonpradub, 2012).

Gustafsson ve arkadaşları, İsveç'te 24056 otopsi vakası değerlerine MLR uygulamışlardır. Vücut ağırlığı, vücut boyu, cinsiyeti ve yaş gibi parametreler ile bir model oluşturmak hedeflenmiştir. Hedeflenen model oluşturulmuş fakat başarı oranı düşük çıkmıştır. "Belirsiz parametreler ile MLR, düşük başarı oranı vermektedir" sonucuna varmaktadırlar (Gustafsson ve dię., 2017).

Nasri ve arkadaşları, LTE geniş kablosuz (wireless) ağlarında servis kalitesini etkileyen parametrelerin tahmini için MLR kullanmışlardır. Modelde, ortalama kullanıcı verimi, ortalama gecikme, ağ trafięi ve ortalama kanal kalitesini içeren bir öğrenme veri seti kullanmaktadırlar. Kullanıcı veriminin ağ trafięi ve ortalama kanal kalitesi ile arttığı, gecikme ile düştüğü gözlemlenmektedir (Nasri ve Hamdi, 2019).

1.1.5 İris veri seti üzerine yapılan çalışmalar

Patel ve arkadaşları, iris veri setini kullanarak 4 farklı algoritmanın (Karar Ağacı, Çok Katmanlı Algılayıcı, Naive Bayes ve Çok Sınıflı Sınıflandırma) sonuçlarını (TP-oran, Fp-oran, Kesinlik, Duyarlılık ve ROC parametresi) kıyaslamışlardır. WEKA programı ile uygulama gerçekleştirilmiştir. Çok Katmanlı Algılayıcı yönteminin tüm sonuç değerlerinde dięer yöntemlerden daha başarılı olduęu gözlemlenmektedir (Patel ve dię., 2014).

Shekhawat ve arkadaşları, Birliktelik Kural Çıkarımı (Association Rule Mining) ve Sınıflandırma Kural Madencilięi (Classification Rule Mining) yöntemlerini birleştirerek Sinir Aęı Birliktelik Sınıflandırma Sistemi (Neural Network Association Classification System) adında hibrid bir algoritma önermişlerdir. Bu algoritmayı kullanarak iris veri seti üzerinden başarı oranları analiz edilmiştir. Sonuç olarak 0.2 Öğrenme katsayısı, 0.7 momentum sabiti ve 2000 eğitim turu (epoch) ile başarı oranı 96.667 ile en yüksek seviyeye çıkmaktadır (Shekhawat ve Dhande, 2011).

Tiwari ve arkadaşları, iris veri seti üzerinde K-means ve K-medoid kümeleme algoritmalarını kullanarak zaman ve mekan karmaşıklık değerlerini kıyaslanmışlardır. İki algoritmanın analizi sonucunda, K-means yönteminin zaman olarak K-medoid yöntemine

göre az maliyetli olduğu görülmektedir. Fakat K-means yönteminin K-medoid yöntemine göre gürlütiye daha hassas olduğu gözlemlenmektedir (Tiwari ve Singh, 2012).

Demir ve arkadaşları, iris veri setine Fidan Gelişim Algoritması uygulayarak kümeleme işlemi yapmışlardır. Bu veri setine daha önce uygulanan algoritmalarda alınan başarı oranı %33 iken FGA'daki başarı oranı %79,33 olarak gözlemlenmektedir. Bu başarının sebebi, popülasyonun random yerine düzenli olarak üretilmesidir (Demir ve Karcı, 2007).

1.2 Tezin Amacı

Optimizasyon algoritmalarında en az çalışma yapılan alan başlangıç popülasyonu oluşturma yöntemleri olarak gözlemlenmektedir. Optimum noktaya yakınsama hızını önemli derecede yükseltmesi ve daha kaliteli çözümler elde edilebilmesi için başlangıç popülasyonu oluşturma yöntemleri konusuna daha çok odaklanması gerekmektedir. Araştırmacılar tarafından halen rastgele başlatma yöntemi tercih edilmektedir. Tercih edilen rastgele yöntemler deterministik bilgisayarlar ve formüllerle uygulandıkları için aslında tam olarak rastgele değillerdir. Bu sebeple sözde rastgele yöntemler olarak bilinmektedirler.

Tezin amacı, başlatma yöntemleri ile ilgili detaylı bir inceleme yapmak ve uniform bir başlatma yöntemi ile deterministik metasezgisel bir optimizasyon algoritması geliştirmektir. Bu doğrultuda tüm başlatma yöntemlerinin avantajları, dezavantajları ve kullanıldıkları alanlar ile ilgili detaylı bir araştırma yapılmaktadır. Bu araştırmanın sonucunda literatüre başlatma yöntemleri ile ilgili yeni bir kategorizasyon önerisi yapılmaktadır. Ayrıca düzgün dağılımlı yeni bir başlatma yöntemi elde edilmektedir. Bu yeni yöntem popülasyon tabanlı optimizasyon algoritmalarında kullanılabilir. Düzgün dağılımlı yeni yöntem ile türe ve dayalı optimizasyon algoritmaları hibridlenerek deterministik metasezgisel bir optimizasyon algoritması elde edilmektedir.

1.3 Tezin Organizasyonu

Tez çalışması altı bölümden oluşmaktadır. Birinci bölümde tez içerisindeki temel konularla ilgili literatürde yer alan ilişkili çalışmalara yer verilmektedir. Yapılan araştırmalar sonucunda başlangıç popülasyonları ile ilgili yeni kategorizasyon önerisi de bu bölümde yer verilmektedir. Ayrıca tezin amacı da bu bölümde ifade edilmektedir.

İkinci bölümde literatürde kullanılan tüm başlangıç popülasyon oluşturma yöntemlerinin detaylıca irdelenmesi sağlanmaktadır. Stokastik, deterministik ve sezgisel yöntemler olmak üzere üç ana başlık halinde anlatılmaktadır. Bu üç ana başlığın tüm alt başlıklarının

formülleri, kim tarafından önerildikleri ve analizleri de bu bölümde anlatılmaktadır. Ayrıca üç ana başlığın da analizleri yapılmaktadır.

Üçüncü bölümde türev denklemleri ile ilgili bilgilere yer verilmektedir. Literatürde yer alan en popüler türev denklemlerinin formülleri, kim tarafından önerildikleri ve özellik bakımından karşılaştırmaları anlatılmaktadır.

Dördüncü bölümde tez kapsamında geliştirilen yöntemin problem formülasyonu, deneysel çalışması ve kullanılan amaç fonksiyonları yer almaktadır. Geliştirilen deterministik metasezgisel algoritmanın başlangıç popülasyonu için önerilen yeni bir yöntemin sözde kodu verilmektedir. Kullanılan iris bitkisinin veri seti ile ilgili detaylı bilgiler yer almaktadır. Deneysel çalışma bölümünde uygulamanın tüm modülerinin görünümü, yöntemin akış diyagramı ve sözde kodu açıklanmaktadır. Ayrıca kullanılan hata fonksiyonlarının formüllerine ve detaylı bilgilerine yer verilmektedir.

Beşinci bölümde tez kapsamında geliştirilen deterministik metasezgisel algoritmanın farklı başlangıç popülasyonları ile başlatılarak elde edilen deneysel sonuçlarına yer verilmektedir. Deterministik ve stokastik başlatma ile elde edilen sonuçlar çizelgeler halinde verilmektedir. Bu çizelgelerde farklı türev denklemlerinden ve farklı amaç fonksiyonlarından elde edilen sonuçlar da mevcuttur. Ayrıca her hata fonksiyonuna ait tüm türev denklemlerinden alınan grafiksel sonuçlar da bu bölümde verilmektedir.

Altıncı bölümde tez çalışması ile ilgili sonuçlara ve yapılan önerilere yer verilmektedir. Önerilen yeni kategorizasyonun ve geliştirilen algoritmanın genel değerlendirilmesi ve algoritmanın uygulama alanları ile ilgili bilgiler bu bölümde yer almaktadır. Son olarak altıncı bölümü müteakiben kaynaklar verilmektedir.

2. BAŞLANGIÇ POPÜLASYONLARI

2.1 Stokastik Yöntemler (Stochastic Methods)

Rastgele sayılar üreten metotlardır. Bir bilgisayarın, bir insan gibi rastgele bir sayı üretmesi beklenmektedir, fakat bu deterministik bilgisayarlarla mümkün olmamaktadır. Aslında “insanlar yüzde yüz tahmin edilemeyen sayılar üretebilir” denilememektedir. Çünkü bir süre sonra aynı veya yakın ya da belli bir aralıktaki sayıları tekrar edebilmektedir. Harici bir bileşen yardımıyla deterministik olmayan (gerçek) rastgele sayılar üretilmektedir (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010).

2.1.1 Sözde rastgele sayı üreticileri (pseudo random number generation, PRNG)

Rastgele (random) kabul edilen fakat gerçek rastgele sayılar olmadığı için sözde rastgele (psudeo random) ismini almaktadır. Çünkü bilgisayar programlarında rastgele sayılar üreten komutlar, deterministik bir algoritma yardımıyla çalışmaktadır. Dolayısıyla gerçek rastgele sayıları üretmesi mümkün değildir. Ancak günümüzde PRNG’ler çok başarılıdır ve ürettikleri sayılar gerçek rastgele sayılarla ayırt edilememektedir. PRNG’ler, seçilen bir çekirdek değerinin üstüne çeşitli formüller uygulayarak yeni sayılar üretmektedir. Başlangıç değeri olarak kabul edilen bu çekirdek, çoğunlukla bilgisayar saatinden faydalanılarak üretilmektedir (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

2.1.1.1 Orta kare yöntemi (middle square method, MSM)

John Von Neumann tarafından 1946 yılında, bir sayı üretme yöntemi olarak literatüre kazandırılan bir yöntemdir. Çalışma mantığı Çizelge 2.1’de gösterilmektedir.

Çizelge 2.1 : Orta kare yöntemi örneği.

Sayı	Karesi	Sonraki Sayı
65 (Çekirdek)	04225 (ortadaki basamak sayısı=3)	422
422	0178084	780
780	0608400	84
...

İstenilen sayıda bu döngü tekrar edilmektedir (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos vd., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

Çok yavaş ve tatmin edici olmayan bir yöntem olan MSM, özellikle AES, RSA, IDEA ve GOST gibi kriptografik algoritmalarda kullanılmaktadır. Pozitif sayı üretiminde hızlı olmasına rağmen negatif sayı üretiminde hızı düşmektedir.

2.1.1.2 Doğrusal eşleşik üretici (linear congruential generator, LCG)

D.H.Lehmer tarafından 1949 yılında önerilen çok basit ve hızlı bir yöntemdir.

$$X_{n+1} = aX_n + c \pmod{m} \quad (2.1)$$

Burada, a ($0 < a < m$) ve c ($0 \leq c < m$) katsayılarıdır. m ($0 < m$) moduna göre bir sonraki sayı üretilmektedir. a , c ve m tamsayıdır, a ve m asal sayıdır. Ayrıca $0 \leq X_0$ (seed value) $< m$ olmalıdır (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

Basit ve hızlı bir yöntem olduğu için şifreleme algoritmalarında kullanılmaktadır. Özellikle dinamik şifrelerin oluşturulmasında oldukça başarılı bir yöntem olduğu gözlemlenmektedir. Şifreli mesajlaşma uygulamaları (whatsapp gibi) veya ücretli yayın yapan dijital platformlar (spor, film vb.) için LCG yönteminden faydalanılması daha uygun olabilir.

2.1.1.3 Karesel eşleşik üretici (quadratic congruential generator, QCG)

LCG yönteminden esinlenmektedir fakat lineer değildir. (2.2) ve (2.3)'teki gibi iki türü mevcuttur (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

1. Tür:

$$X_{n+1} = aX_n^2 + c \pmod{m} \quad (2.2)$$

2. Tür:

$$X_{n+1} = aX_n^2 + bX_n + c \pmod{m} \quad (2.3)$$

LCG'nin genelleştirilmiş ikinci dereceden bir türü de denebilir. LCG'den fazla çarpma yaptığı için nispeten daha yavaş sayı üretmektedir. Bu dezavantaj, shifting algoritması ile giderilebilir.

2.1.1.4 Gecikmeli Fibonacci üretici (lagged Fibonacci generator, LFG veya LFib)

Bu yöntemin fikri Fibonacci dizilerine dayanmaktadır. Fibonacci dizileri bu yöntemin özel bir halidir.

Fibonacci dizisinin genel formülü;

$$S_n = S_{n-1} + S_{n-2} \quad (2.4)$$

LFG'nin formülü ise;

$$S_n = S_{n-j} * S_{n-k} \pmod{m}, 0 < j < k \quad (2.5)$$

Burada * işareti, toplama, çıkarma, çarpma veya xor işlemi olabilir. m, genelde 2'nin kuvveti olarak alınmaktadır (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

Paralel çalışan yapılarda oldukça başarılı bir sayı üretme yöntemi olan LFG, başlangıç tohumu seçme dezavantajına sahiptir. Bu dezavantajı, bir tohum tablosu oluşturarak ortadan kaldırmak mümkündür. Bu tablolar her bir öge için tutulduğundan ekstra bellek gerekmektedir. LFG, paralel çalışan işlemcilerde ayrı ayrı tohumlarla birbirinden bağımsız başlatılarak sayı dizileri üretilebilir. Ayrı işlemcilerden sonuçlar aynı zamanda gelmemektedir ve gecikmeye sebep olmaktadır. Bu sebeple, kriptografiye uygun olmadığı söylenebilir.

2.1.1.5 Blum Blum Shub (BBS)

Lenore Blum, Manuel Blum ve Michael Shub tarafından 1986 yılında önerilen bir yöntemdir. (2.6)'daki formül ile bir sonraki sayı hesaplanmaktadır.

$$X_{n+1} = X_n^2 \pmod{m} \quad (2.6)$$

Burada $m=p*q$, p ve q büyük asal sayılardır (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

BBS, üstel bir sayı üretici olduğu için yüksek hesaplama karmaşıklığına sahiptir. Bu hesaplama karmaşıklığından faydalanarak, şifre çözme saldırılarına karşı yavaşlatmaya yönelik yaklaşımlarda kullanılabilir. Genelde düz metin şifreleme algoritmalarını daha fazla tahmin edilemez yapmak için BBS'den faydalanılmaktadır.

2.1.1.6 Geri beslemeli kaydırma yazmacı (feedback shift register, FSRG)

Tausworthe tarafından 1965 yılında önerilen lineer FSRG yönteminin 1973'te Lewis ve Payne tarafından değiştirilerek geliştirilmiş halidir. FSRG, mod 2'ye sahip çoklu özyinelemeli üreticinin özel bir formudur. O form (2.7)'de gösterilmektedir.

$$X_n = (a_1X_{n-1} + \dots + a_kX_{n-k}) \pmod{2} \quad (2.7)$$

Önceki k bitlerine göre 0 veya 1 ayrı bitler üretmektedir (Collet ve Rennard, 2011; Chong ve Zak, 2004; DiCarlo, 2012; Hassler, 2016; Mitsos ve diğ., 2008; Stork ve diğ., 2018; Venter, 2010; White, 1971; X.-S. Yang, 2010; X. Yang, 2010).

Lineer olma özelliğinden dolayı hesaplama karmaşıklığı düşük olmaktadır. Şifreleme algoritmalarında oldukça kullanılmaktadır. Hızlı bir şifreleme ihtiyacı olan uygulamalarda FSRG tercih edilebilir. Ancak düşük hesaplama karmaşıklığından dolayı saldırılara karşı zayıf kalmaktadır. Çünkü üretilen sayıların tahmin edilmesi kolay olmaktadır.

2.1.1.7 PRNG yöntemlerinin analizleri

Rastgele sayıların kullanılması zorunlu olan uygulamalarda PRNG'ler ucuz yöntemler olması sebebiyle tercih edilmektedir. Yazılımsal olarak sayı ürettikleri için ucuz yöntemlerdir. PRNG'ler, aslında gerçek rastgele sayılar üretmemektedirler fakat bu sayıların tahmin edilmesi mümkün değildir hatta imkânsız denebilir. Sayıların tahmin edilememesi, kriptografi gibi alanlarda oldukça ilgi çekmektedir. Şifreleme algoritmalarının saldırılara karşı bu tahmin edilemeyen sayıları kullanması gücüne güç katmaktadır. Kaba Kuvvet (Brute Force) tipli saldırılardan kaçınmak için hesaplama karmaşıklığı yüksek olan PRNG'ler tercih edilebilir. Karmaşıklığı yüksek olan bir PRNG'nin tahmin edilmesi imkânsız denebilecek kadar zordur.

2.1.2 Yarı rastgele sayı üreticileri (quasi random number generation, QRNG)

PRNG'ler, gerçekte rastgele olmayan sayılar ürettikleri için bazı uygulamalar için kullanışlı olmamaktadır. Çünkü üretilen sayılar, çözüm uzayının bazı bölgelerinde yoğunluk gösterebilir. Böylelikle çözüm uzayı, yüksek tutarsızlıkta bir dağılım olmaktadır ve bu istenmeyen bir durumdur. Bazı uygulamalar için çözüm uzayına eşit dağılan, düşük tutarsızlıkta sayılar üretmek gerekebilir. Bu sayılar, yarı rastgele sayılar olarak ifade edilmektedir. QRNG'ler, PRNG'lerden daha az rastgele sayılar üretmektedir. Ayrıca QRNG'ler tarafından üretilen sayı dizilerine düşük tutarsızlık dizileri denilmektedir (Collet ve Rennard, 2011; DiCarlo, 2012; Digehsara ve diğ., 2020; Dutang ve Wuertz, 2009; Kimura ve Matsumura, 2005; Morokoff ve Caflisch, 1994; Omran ve diğ., 2013; Uy ve diğ., 2007).

2.1.2.1 Van Der Corput dizisi

Van Der Corput dizisi, diğer tüm düşük tutarsızlık dizilerinin temelini oluşturmaktadır. Asal sayı tabanında oluşan tek boyutlu bir dizidir.

p ($p \geq 2$) asal sayı tabanında bir tamsayı (n) aşağıdaki formül ile hesaplanmaktadır.

$$n = (a_m \dots a_1 a_0)_p = \sum_{i=0}^m a_i p^i \quad (2.8)$$

Burada; a_i 'ler ($0 \leq a_i < p$) katsayılarıdır. m , p tabanına çevrilmiş haldeki basamak sayısının 1 eksiğidir. n tamsayısı p tabanına çevrildikten sonra, p tabanındaki sayı önce ters çevrilir sonra ondalık hale getirilmektedir. Elde edilen bu sayının $\phi_p(n)$ değeri (2.9) denklemi ile hesaplanmaktadır (Faure ve diğ., 2015; Gürsu ve diğ., 2018; Morokoff ve Caflisch, 1994; Pausinger, 2020). Çizelge 2.2'de üç tabanında yedi elemanlı Van Der Corput dizisi ve Çizelge 2.3'te Van Der Corput dizisinin ilk terimleri gösterilmektedir.

$$X_n = \phi_p(n) = (0.a_0 a_1 \dots a_m)_p = \sum_{i=0}^m \frac{a_i}{p^{i+1}} \quad (2.9)$$

Tek boyutlu en basit düşük tutarsızlık dizidir. Tek boyutlu problemlerin çözümünde tercih edilebilir.

Çizelge 2.2 : 3 tabanında 7 elemanlı Van Der Corput dizisi.

Taban 10	Taban 3	Taban 3 $\phi_3(i)$ formu	Taban 10 $\phi_3(i)$ formu
i=1	01	0.10	0.3333
2	02	0.20	0.6667
3	10	0.01	0.1111
4	11	0.11	0.4444
5	12	0.21	0.7778
6	20	0.02	0.2222
7	21	0.12	0.5556

Çizelge 2.3 : Van Der Corput ilk terimleri.

n	p tabanında n			$\phi_p(n)$		
	$p=2$	$p=3$	$p=5$	$p=2$	$p=3$	$p=5$
0	0	0	0	0	0	0
1	1	1	1	0.5	0.333	0.2
2	10	2	2	0.25	0.666	0.4
3	11	10	3	0.75	0.111	0.6
4	100	11	4	0.125	0.444	0.8
5	101	12	10	0.625	0.777	0.04
6	110	20	11	0.375	0.222	0.24
7	111	21	12	0.875	0.555	0.44
8	1000	22	13	0.0625	0.888	0.64

2.1.2.2 Halton dizisi

Tek boyutlu Van Der Corput dizisinin çok boyutlu halidir (Digehsara ve diğ., 2020; Dutang ve Wuertz, 2009; Gürsu ve diğ., 2018; Kimura ve Matsumura, 2005; Omran ve diğ., 2013; Uy ve diğ., 2007; Zeng, 2016).

$$X_n = (\phi_{p_1}(i), \dots, \phi_{p_n}(i)) \quad (2.10)$$

Burada; n problem boyutunu, i , dizinin eleman sayısını ve p problem boyutu kadar asal sayı tabanlarını göstermektedir. Çizelge 2.4'te 3 boyutlu 6 elemanlı Halton dizisi gösterilmektedir. Dizinin ilk boyutu 2 tabanında Van Der Corput dizisidir.

Çizelge 2.4 : 3 boyutlu 6 elemanlı Halton dizisi.

i	$p=2$	$p=3$	$p=5$
1	0.5000	0.3333	0.2000
2	0.2500	0.6667	0.4000
3	0.7500	0.1111	0.6000
4	0.1250	0.4444	0.8000
5	0.6250	0.7778	0.0400
6	0.3750	0.2222	0.2400

Çok boyutlu problemlerin çözümünde kullanmak için Van Der Corput dizisi çok boyutlu hale getirilmiştir. Halton dizisi olarak adlandırılan bu çok boyutlu dizi, birçok uygulamada başlangıç popülasyonu oluşumunda kullanılmaktadır. Çok yüksek boyutlu problemlerin çözümünde tavsiye edilmemektedir. Çünkü yakınsama hızı düşük olmaktadır. Fakat az boyutlu problemlerde diğer bütün düşük tutarsızlık dizilerine göre yakınsama hızı yüksektir.

2.1.2.3 Faure dizisi

Faure dizisi, Halton dizisi gibi çok boyutlu bir dizidir. Halton dizisinin genelleştirilmiş hali de denilmektedir. p asal sayı tabanındaki ilk boyutu ise, Van Der Corput dizisi ile eşittir. Halton dizisinden farklı olarak, her boyutta aynı p asal sayı tabanını kullanmaktadır (Dutang ve Wuertz, 2009; Gürsu ve diğ., 2018; Uy ve diğ., 2007).

$$c_j^i = \binom{i}{j} \text{mod}(p) = \frac{j!}{i! \cdot (j-i)!} \text{mod}(p) \quad (2.11)$$

$i=1,2,\dots,m$ olmak üzere, $a_i^k(n)$ (2.12) formülü ile hesaplanmaktadır.

$$a_i^k(n) = \sum_{j=0}^m c_j^i \cdot a_i^{k-1}(n) \text{ mod}(p) \quad (2.12)$$

Dizinin her bir boyutu ise (2.13) ile hesaplanmaktadır.

$$x_n^k = \phi_p^k(n) = \sum_{i=0}^m \frac{a_i^k(n)}{p^{i+1}} \quad 2 \leq k \leq s \quad (2.13)$$

Formülleri ile hesaplanmaktadır. Çizelge 2.5'te 3 tabanında ve 3 boyutlu bir Faure dizisi gösterilmektedir.

Çizelge 2.5 : 3 boyutlu Faure dizisi.

n	$a_i^1(n)$			x_n^1	x_n^2	x_n^3
	$a_0^1(n)$	$a_1^1(n)$	$a_2^1(n)$			
1	1	0	0	1/3	1/3	1/3
2	2	0	0	2/3	2/3	2/3
3	0	1	0	1/9	4/9	7/9
4	1	1	0	4/9	7/9	1/9
5	2	1	0	7/9	1/9	4/9
6	0	2	0	2/9	8/9	5/9
7	1	2	0	5/9	2/9	8/9
8	2	2	0	8/9	5/9	2/9
9	0	0	1	1/27	16/27	13/27
10	1	0	1	10/27	25/27	22/27
11	2	0	1	19/27	7/27	4/27

Faure dizileri de Halton dizileri gibi yüksek boyutlu problemlerde tercih edilmektedir. Halton dizisinin her boyutta farklı taban kullanması dezavantajını gidermek için önerilmiştir. Her boyutta aynı tabanı kullandığı için hesaplama karmaşıklığı düşük olmaktadır. Bu yüzden yüksek boyutlarda Halton dizinin yerine tercih edilebilir.

2.1.2.4 Sobol dizisi

Sobol dizileri, Van Der Corput dizisinden farklı bir yaklaşımla elde edilmiştir. Sobol dizileri $x_n = (x_{n,1}, \dots, x_{n,d})$ ($0 < x_n < 1$), uzunluğu ω olan bitlerin $(v_{i,j} (i = 1, \dots, \omega)(j = 1, \dots, d))$ ikili fonksiyonlarından üretilmektedir. Sobol dizisinde her bir boyut için, ilkel bir polinoma ihtiyaç vardır. Bu polinomun genel gösterimi (2.14)'te gösterilmektedir (Dutang ve Wuertz, 2009; Gürsu ve diğ., 2018; Omran ve diğ., 2013; Uy ve diğ., 2007).

$$p_j(x) = x^q + a_1x^{q-1} + \dots + a_{q-1}x + 1 \quad (2.14)$$

Sobol dizisilerini tanımlamak için, her bir boyutta ayrı ayrı ilkel polinomlardan seçilmesi gerekmektedir. Bazı ilkel polinomlar aşağıda gösterilmektedir.

$$1 + x, 1 + x + x^2, 1 + x + x^3, 1 + x^2 + x^3, 1 + x + x^4, 1 + x^3 + x^4, 1 + x^2 + x^5, \\ 1 + x^3 + x^5$$

Daha sonra tamsayılar $(M_{i,j})_i$ üzerinde (2.15)'teki gibi q-terimli tekrarlı ilişki tanımlanmaktadır.

$$M_{i,j} = 2a_1M_{i-1,j} \oplus 2^2a_2M_{i-2,j} \oplus \dots \oplus 2^{q-1}a_{q-1}M_{i-q+1,j} \\ \oplus 2^qa_qM_{i-q,j} \oplus M_{i-q} \quad (2.15)$$

Burada $i > q$ 'dur. Bu, yön numaralarının $v_{i,j} = M_{i,j}/2^i$ gibi hesaplanmasına izin vermektedir. Bu yineleme, sırasıyla $2, \dots, 2^q$ 'dan daha küçük olan rastgele tek tamsayılar kümesi $v_{1,j}2^\omega, \dots, v_j2^q\omega$ tarafından başlatılmaktadır. Son olarak Sobol dizisinin n th teriminin j th boyutu (2.16)'da tanımlanmaktadır.

$$x_{n,j} = b_1v_{1,j} \oplus b_2v_{2,j} \oplus \dots \oplus v_{\omega,j} \quad (2.16)$$

Burada b_k 'lar, n tamsayısının bitleridir ($n = \sum_{k=0}^{\omega-1} b_k2^k$) (Dutang ve Wuertz, 2009; Gürsu ve diğ., 2018; Omran ve diğ., 2013; Uy ve diğ., 2007). Çizelge 2.6'da 3 boyutlu bir Sobol dizisi gösterilmektedir.

Sobol dizisi oluşturulurken, 2 tabanında sayılar üretildiği için diğer tüm QRNG'lerden daha hızlı olmaktadır. 2 tabanında olduğu için işlem döngüsü hızlıca sonlanmaktadır. Yüksek boyutlarda kullanılması önerilmektedir. Çünkü her boyutta sayılar üretilirken tek taban kullanıldığı için sonuçların doğruluğunda daha fazla tutarlılık söz konusudur.

Çizelge 2.6 : 3 boyutlu Sobol dizisi (çekirdek=0).

	Boyut=1 (Taban=2)	Boyut=2 (Taban=2)	Boyut=3 (Taban=2)
n=1	1/2	1/2	1/2
n=2	3/4	1/4	3/4
n=3	1/4	3/4	1/4
n=4	3/8	3/8	5/8
n=5	7/8	7/8	1/8
n=6	5/8	1/8	3/8
n=7	1/8	5/8	7/8
n=8	3/16	5/16	5/16

2.1.2.5 Kronecker dizisi

Başka bir tür düşük tutarsızlık dizisi olan Kronecker dizisi, irrasyonel sayı ve kesirli kısım kullanmaktadır. Gerçek bir x 'in kesirli kısmı $\{x\} = x - |x|$ ile gösterilmektedir. Bu sonsuz dizi $(\{n\alpha\})_{n \leq 0}$, tutarsızlığı için bir sınıra sahiptir. (2.17)'de bu sınır hesaplanmaktadır.

$$D_n \leq C \frac{1 + \log n}{n} \quad (2.17)$$

Bu sonsuz diziye $(\{n\alpha\})_{n \leq 0}$ Kronecker dizisi denilmektedir (Dutang ve Wuertz, 2009).

Kronecker dizisi, ayrık olmayan (sürekli) problemlerin çözümünde tercih edilen bir QRNG'dir. Bu tür problemlerin çözüm simülasyonunda oldukça başarılı bir yöntemdir. Gerçek dünya problemlerinde daha çok tercih edilebilir.

2.1.2.6 Hammersley dizisi

Van Der Corput dizisinin n boyuta genişletilmiş halidir. Hammersley dizisi, (2.18)'de verilen denklem ile oluşturulmaktadır.

$$x_i = \left(i/N \phi_{p_1}(i), \dots, \phi_{p_{n-1}}(i) \right) \quad i = 1, 2, \dots, N \quad (2.18)$$

Hammersley dizisinin ilk boyuttaki elemanlar, i/N ile hesaplandığı için p asal sayı tabanında Halton dizisinden 1 eksik olmaktadır (Gürsu ve diğ., 2018; Wong ve diğ., 1997).

Halton dizisinin basit bir türü olan Hammersley dizisi mükemmel homojenlik ve yüksek hesaplama verimliliği sunmaktadır. Bu nedenle optimizasyon problemlerinde sıklıkla tercih edilmektedir.

2.1.2.7 QRNG yöntemlerinin analizleri

Aslında rastgele sayılar üreten yöntemler değildirler ancak stokastik sayılmaktadırlar. Genel olarak çözüm uzayını çeşitlendirmek için kullanılmaktadırlar. Yani çözüm uzayının tam kapsanması amaçlanmaktadır. PRNG'lerin bazı bölgelerde sıkışan başlangıç çözümlerine kıyasla çözüme ulaşmak daha az zaman almaktadır. Birçok problemde ve özellikle şifreleme algoritmalarında tercih edilmektedir. Problemin boyutuna göre başarı oranları değişmektedir. Düşük boyutlu problemler için tüm QRNG'lerin başarı oranları yüksek olmaktadır. Yüksek boyutlu problemler için Sobol dizisi tavsiye edilmektedir. Sobol dizisinden başka genelleştirilmiş Faure dizisi de yüksek boyutlu problemler için tavsiye edilebilir. Bu iki yöntemin uygulanması diğer QRNG'lere göre zor olmaktadır. Zor bir süreçlerinin olması şifreleme algoritmaları için avantaj olmaktadır. Çünkü tahmin edilmesi zorlaşmaktadır ve saldırılardan daha az etkilenmektedirler. Halton dizisi de yüksek boyutlu problemler olmamak şartıyla kullanım kolaylığından dolayı tavsiye edilebilir.

2.1.3 Gerçek rastgele sayı üreticileri (true random number generation, TRNG)

Herhangi bir matematiksel formülle elde edilmeyen, tahmin edilemeyen ve rastgele bir olay yardımıyla üretilen sayılardır. Bu rastgele olaylara, disk erişim süreçleri, sıcaklık, gürültü ve radyoaktif bozulma gibi örnekler verilebilir. Yapısı belirli olan bir algoritma ile üretilmezler (DiCarlo, 2012; Dutang ve Wuertz, 2009; Lee ve diğ., 2018; Petura, 2020; Tuna ve Fidan, 2018; Yu ve diğ., 2019).

2.1.3.1 Fiziksel olmayan TRNG'ler (Non-Physical, NPTRNG)

NPTRNG'ler, sayıları üretirken donanım yerine yazılımdan faydalanmaktadır. Donanım olaylarını, sistem saatini, CPU yürütme zaman gecikmesini gibi kaynaklar kullanılarak gerçek rastgele sayılar üretilmektedir. Ayrıca bir işletim sisteminde, kullanıcı ile sistem arasındaki etkileşimi kullanabilmektedir. Örneğin, kullanıcının klavyede bastığı tuş kombinasyonları kaynak olarak kullanılabilir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Petura, 2020; Yu ve diğ., 2019).

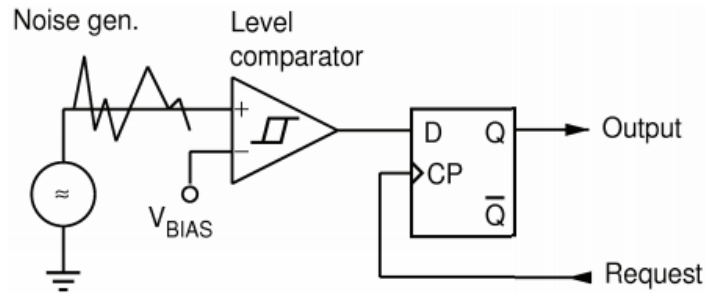
2.1.3.2 Fiziksel TRNG'ler (PTRNG)

PTRNG'ler, fiziksel cihazlardır. Donanımın gürültüsünden, rüzgârın hızından, gök gürültüsünün şiddetinden faydalanabilirler. Algoritma kullanmadan üretilen sayılardır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Petura, 2020; Yu ve diğ., 2019).

2.1.3.2.1 Gürültü (noise)

Fiziksel TRNG türü olan gürültü bazlı sayı üreteçlerindeki ana fikir, analog voltajın rastgeleliğinin periyodik olarak kayıt edilmesi ve bu kayıt edilen değerlerin belirli bir eşik değeriyle karşılaştırılmasıdır. Şekil 2.1'de örnek bir gürültü bazlı TRNG gösterilmektedir. Bu üreteç, eğer rastgele analog voltajın örneklemeleri eşik değerden yüksekse "1", değilse "0" değerini üretmektedir. Birçok gürültü bazlı TRNG'ler vardır. Johnson's effect, Zener noise (in semiconductor Zener diodes), inverse base-emitter breakdown in bipolar transistors, laser phase noise ve chaos noise gibi çeşitleri mevcuttur. Açıkça belirtmek gerekirse, gürültü bazlı TRNG'lerin uygulanabilirliği zordur, hatta aşağıdaki üç özellikten dolayı neredeyse imkânsız olmaktadır (Dutang ve Wuertz, 2009; Petura, 2020; Stipčević ve Koç, 2014).

- ✓ Gürültü kaynağının rastgeleliğinin kanıtlanması
- ✓ Sayısallaştırma veya örnekleme işleminin etkisi
- ✓ Belirleyici işlem sonrasının muhtemel kullanımı



Şekil 2.1 : Örnek bir gürültü bazlı TRNG.

Son yıllarda uygulanabilir gürültü bazlı TRNG'ler üzerinde çokça çalışma yapılmıştır. Gürültü kaynaklarının çok olmasından dolayı araştırmacıların ilgisini çekmektedir.

2.1.3.2.2 Kaos (chaos)

Stokastik bir sayı üretici için Kaos'un kullanılması felsefik bir problemdir. Çünkü Kaos deterministiktir. Rastgele bir sürecin deterministik bir işlemle giderilmesi anlamına gelmektedir. Kavramsal bir karmaşıklık mevcuttur. Bu yüzden Kaos üreteçleri deterministik

sayı üreticileri bölümü altında anlatılmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; S. Ergün ve Özoğuz, 2010; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

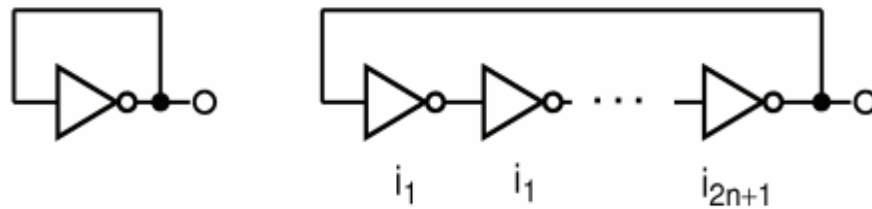
2.1.3.2.3 Kuantum (quantum)

Kuantum fiziğine göre herhangi bir gerçek rastgele sayı üretici “Kuantum” olarak adlandırılmaktadır. Yazı tura atmak gibi örnekler verilebilir. Aynı başlangıç koşullarından başlayıp ve aynı şekilde ölçüm yapıldığında farklı sonuçlar vermektedir, bu ancak Kuantum fiziği ile mümkün olmaktadır. Bu şekilde üretilen sayılar, istenilen gerçek rastgele sayılardır (Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2007; Petura, 2020; Stipčević ve Koç, 2014).

Kuantum sayı üreticileri, cihazların güvenliğine göre kategorilere ayrılmaktadır. Pratik, kendi kendini test eden ve yarı kendi kendini test eden sayı üreticileri olmak üzere üç kategoriden oluşmaktadır. Pratik kuantum sayı üretici güvenilir ve yüksek hızda rastgele sayılar üretmektedir. Kendi kendini test eden kuantum sayı üretici ile üretilen sayılar doğrulanabilmektedirler. Bu yöntemle üretilen sayılar sağlam rastgele sayılar olarak kabul edilmektedirler. Yarı kendi kendini test eden kuantum sayı üreticileri ise ilk iki türün arasında kabul edilen bir kategoridir. Rastgele sayı oluşturma hızı ile cihaz güvenilirliği arasında denge sağlamaktadır (Ma ve diğ., 2016).

2.1.3.2.4 Serbest hareketli osilatör (free running oscillator, FRO)

FRO’da rastgele sayılar üretmede kaynak olarak osilatörlerin faz gürültüsü kullanılmaktadır. Bir hızlı ve bir de yavaş olmak üzere en iki osilatör kullanılmalıdır. Şekil 2.2’de görüldüğü gibi bir tersleyicinin çıkışı girişine uygulandığında bir FRO meydana gelmektedir.



Şekil 2.2 : Hızlı (soldaki) ve yavaş (sağdaki) FRO örnekleri.

Giriş YÜKSEK konumdaysa çıkış DÜŞÜK konumdadır, giriş DÜŞÜK konumdaysa çıkış YÜKSEK konumdadır ve böylece devam etmektedir. Burada salınım meydana gelmektedir. FRO’larda rastgele sayı üretmenin temel ilkesi, hızlı sonuç veren bir FRO’nun çıktısının yavaş bir FRO tarafından örneklenmesidir. Hızlı ve yavaş FRO arasında faz titremesi

meydana gelmektedir. Bu faz titremesinin hem rastgele hem de büyük olması önemlidir. Aksi durumda tekrarlı ikili durum meydana gelmektedir ve rastgelelikten uzak olmaktadır (Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2007; Petura, 2020; Stipčević ve Koç, 2014; Tavas ve diğ., 2008).

TRNG'nin bir türü olan FRO'dan üretilen sayıların rastgelelikleri yetersiz sayılmaktadır. FRO'ların rastgelelik testlerini geçmesi için başka bir rastgelelik kaynağı eklenmelidir. Ek kaynak olarak direnç ısı gürültüsü gibi kaynaklar kullanılabilir.

2.1.3.3 TRNG yöntemlerinin analizleri

TRNG yöntemleri ile üretilen sayılar gerçek rastgele sayılardır. Tahmin edilemezler ve yüksek entropiye sahip olmaktadır. Hatta aynı şartlar oluşturulsa bile aynı sayıları üretmezler. Bu yüzden yüksek güvenli şifreleme algoritmalarında kullanılmaktadırlar. Tahmin edilememesi saldırganlar için istenmeyen bir durumdur. PRNG ve QRNG yöntemlerine göre pahalı sistemlerdir ve uygulanması zordur. Harici kaynaklar kullanarak sayılar üretmektedirler. Bu harici kaynaktan verileri toplamak için ek bir cihaza ihtiyaç duymaktadırlar. Bu yüzden sayı üretmeleri zor olmaktadır. Ayrıca TRNG'ler, aynı şartlarda farklı sayılar ürettikleri için rastgeleliğin doğrulanması süreci de diğer yöntemlere göre dezavantajlı olmaktadır. Bir diğer dezavantajı ise sayı üretiminde geçen zamandır. Diğer yöntemlere göre oldukça yavaş yöntemlerdir.

2.2 Deterministik Yöntemler

Deterministik yöntemler, aday çözümler oluştururken çözüm uzayının her bölgesinden örneklerin olmasını sağlamaktadır. Aslında çeşitliliği arttırmak için kullanılmaktadırlar. Rastgele yöntemlerin, çözüm uzayının bir bölgesine sıkışması ihtimalini ortadan kaldırmaktadırlar.

2.2.1 Düşük tutarsızlık dizileri (low discrepancy sequences)

Düşük tutarsızlık dizileri QRNG olarak da adlandırılmaktadır. Van Der Corput, Halton, Faure, Sobol, Kronecker ve Hammersly dizileri, QRNG başlığı altında anlatılmaktadır.

2.2.2 Kaos (chaos)

Uzun periyotlu ve iyi tekdüzelikli rastgele diziler oluşturmak, karmaşık fenomenleri kolayca simüle etmek, örnekleme, sayısal analiz, karar verme ve özellikle sezgisel optimizasyon için çok önemlidir. Kalitesi, istenen doğruluğu elde etmek için depolama ve hesaplama süresinin

azaltılmasını belirlemektedir. Oluşturulan bu tür diziler, bir uygulama için yeterince “rastgele” olabilir ancak bir başkası için yeterince rastgele olmayabilir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

Kaos deterministiktir, yani doğrusal olmayanda bulunan rastgele benzeri bir süreçtir (dinamik sistem, yani periyodik olmayan, yakınsak olmayan ve sınırlı). Dahası, başlangıç durumuna ve parametresine çok duyarlı bir bağımlılığa sahiptir. Kaosun doğası görünüşte rastgele ve tahmin edilemezdir ve aynı zamanda bir düzenlilik unsuruna sahiptir. Matematiksel olarak Kaos basit deterministik bir dinamik sistemin rastlantısallığıdır ve kaotik sistem rastlantısallığın kaynağı olarak düşünülebilir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

Kaotik bir harita, kaotik durumda çalışan ayrık zamanlı dinamik bir sistemdir.

$$x_{k+1} = f(x_k), \quad 0 < x_k < 1, \quad k = 0,1,2, \dots \quad (2.19)$$

Kaotik dizi, rastgele sayı dizisi olarak yayılı spektrum dizisi olarak kullanılabilir $\{x_k: k = 0,1,2, \dots\}$.

Kaotik dizilerin üretilmesi ve depolanmasının kolay ve hızlı olduğu kanıtlanmıştır, uzun dizilerin depolanmasına gerek yoktur. Çok uzun diziler için bile sadece birkaç fonksiyon (kaotik haritalar) ve birkaç parametre (başlangıç koşulları) gereklidir. Ek olarak, çok sayıda farklı dizi, basitçe başlangıç durumu değiştirilerek oluşturulabilir. Dahası, bu diziler deterministiktir ve tekrarlanabilir. Son zamanlarda rastgele diziler yerine kaotik diziler benimsenmiş ve güvenli iletişim ve doğrusal olmayan devreler, DNA hesaplama, görüntü işleme gibi alanlarda sıklıkla kullanılmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

2.2.2.1 Tek boyutlu haritalar (one-dimensional maps)

2.2.2.1.1 Lojistik harita (logistic map)

Biyolojik popülasyonun doğrusal olmayan dinamiklerinde ortaya çıkan ve kaotik davranışı gösteren, 1976 yılında önerilen en basit haritalardan biri olan lojistik haritanın denklemi

(2.20)'de gösterilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = aX_n(1 - X_n) \quad (2.20)$$

Burada X_n , n 'inci kaotik sayıdır ve n , iterasyon sayısını göstermektedir. $X_0 \in (0,1)$.

En iyi bilinen tek boyutlu kaotik haritadır. Tek boyutlu olmasından dolayı uygulanması kolaydır ve basit bir kaotik yapıya sahiptir. Basitliğinden dolayı saldırılara karşı oldukça savunmasız ve sınırlı aralıklarda sayılar üretmektedirler. Bu dezavantajları gidermek için son yıllarda birçok geliştirilmiş hali önerilmiştir. İlk yıllarda popülasyon oluşturma yöntemi olarak kullanılmasına rağmen sonrasında şifreleme yöntemlerinde tercih edilmektedir.

2.2.2.1.1 Çember harita (circle map)

1994 yılında Zheng tarafından önerilen Çember haritası (2.21)'de gösterilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün & Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = X_n + b - (a/2\pi) \sin(2\pi X_n) \text{ mod}(1) \quad (2.21)$$

Çember haritası da Lojistik harita gibi tek boyutlu bir haritadır. Lojistik haritanın kullanıldığı tüm alanlarda Çember haritası da kullanılabilir. Aralarındaki fark, Çember haritasının Lojistik haritadan biraz daha yüksek karmaşıklığa sahip olmasıdır. Bundan dolayı güvenlik konusunda daha çok ön plana çıkmaktadır. Çember haritasının başlangıcında yapılacak küçük bir değişiklikle birbirinden bağımsız birçok sayı dizisi elde edilmektedir.

2.2.2.1.2 Gauss haritası (gauss map)

1992 yılında Peitgen, Jurgens ve Saupe tarafından önerilen Gauss haritası, literatürde test amaçlı kullanılmaktadır. (2.22) ve (2.23)'te gösterilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = \begin{cases} 0, & X_n = 0 \\ 1/X_n \text{ mod}(1), & X_n = 0 \in (0,1) \end{cases} \quad (2.22)$$

$$1/X_n \bmod(1) = \frac{1}{X_n} - \left\lfloor \frac{1}{X_n} \right\rfloor \quad (2.23)$$

Gauss haritasında diğer kaotik haritalardan daha fazla parametre kullanıldığı için şifreleme algoritmaları için öncelikli olarak önerilmektedir. Parametrelerin fazla olması daha az tahmin edilmesi anlamına gelmektedir. Ayrıca şifreleme ve şifre çözme işlemlerinde gauss haritasının kullandığı süre, Lojistik haritadan daha az olmaktadır.

2.2.2.1.3 Sinüs haritası (sine map)

Sinüs haritası, tek boyutlu bir kaotik haritadır ve basit bir denkleme sahiptir. $[0, 1]$ aralığında sayı dizileri üretmektedir. Ayrıca tek modlu çizgeler olarak da adlandırılmaktadır. Sinüs haritası (2.24)'te tanımlanmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = 2.3(X_n)^{2 \sin(\pi X_n)} \quad (2.24)$$

Sinüs haritası, Lojistik ve Çadır haritaları gibi unimodal haritalardandır. Grafiğinde tek maksimum nokta olan haritalara unimodal haritalar denilmektedir. Formülleri farklı olmasına rağmen sinüs haritasının grafiği şaşırtıcı bir şekilde Lojistik harita ile özdeştir. Dolayısıyla Lojistik haritalara alternatif olarak kullanılabilir.

2.2.2.1.4 Çadır haritası (tent map)

1992 yılında Peitgen ve arkadaşları tarafından önerilen çadır haritası, lojistik haritaya benzemektedir. (2.25)'te gösterilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = \begin{cases} X_n/0.7, & X_n < 0.7 \\ 10/3X_n(1 - X_n), & otherwise \end{cases} \quad (2.25)$$

Unimodal haritalarından olan çadır haritası şifreleme ve optimizasyon algoritmalarında oldukça fazla tercih edilmektedir. Özellikle sürü tabanlı algoritmalarda kullanımı önerilmektedir. Daha önce yapılan çalışmalarda (kaotik gri kurt algoritması (Kohli ve Arora, 2018), kaotik balina optimizasyon algoritması (Kaur ve Arora, 2018) vb.) çadır haritasının kullanımı diğer haritalardan kullanımından daha fazla verim elde edildiği kanıtlanmıştır.

2.2.2.2 İki boyutlu haritalar (two-dimensional maps)

2.2.2.2.1 Henon haritası (henon map)

2003 yılında Cvitanovic`, Artuso, Mainieri ve Vattay tarafından önerilen Henon haritası, test amacıyla en sık kullanılan doğrusal olmayan 2 boyutlu bir haritadır. (2.26) ve (2.27)'de gösterilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$X_{n+1} = 1 - aX_n^2 + bY_n \quad (2.26)$$

$$Y_{n+1} = X_n \quad (2.27)$$

Bazen (2.28)'deki gibi temsil edilen 2 adımlı bir tekrarlama ilişkisi olarak yazılmaktadır.

$$X_{n+1} = 1 - aX_n^2 + bX_{n-1} \quad (2.28)$$

Henon haritası en basit iki boyutlu haritalardandır. Sabit noktalar ve kararlılık gibi bir sistemin kaotik dinamiklerini incelemek için tercih edilmektedir. Ayrıca gerçek dünya problemlerinin çözümünde oldukça kullanışlıdır (Wen, 2014).

2.2.2.2.2 Baker haritası (baker map)

Baker haritası imge şifrelemede çok kullanılan iki boyutlu kaotik haritadır. (2.29)'da tanımlanmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$B(x, y) = \begin{cases} (2x, 2y), & \text{for } 0 \leq x < 0.5, \\ (2 - 2x, 1 - \frac{y}{2}), & \text{for } 0.5 \leq x < 1. \end{cases} \quad (2.29)$$

(2.30)'da çadır haritasına benzer bir boyut benimsenmektedir.

$$x_{k+1} = \begin{cases} 2x_k, & \text{for } 0 \leq x_k < 0.5, \\ 2 - 2x_k, & \text{for } 0.5 \leq x_k < 1. \end{cases} \quad (2.30)$$

Bu harita (0, 1) aralığında kaotik diziler oluşturmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

Kaosun karakterize edilmesi fikri için oldukça yararlı bir yöntemdir. Hiperbolik (her noktanın daralma ve genişleme yönlerine sahip olması) çekiciler için önemli bir haritalamadır. Hiperbolik olması çeşitli özelliklerin matematiksel olarak kanıtlanmasında avantaj sağlamaktadır.

2.2.2.2.3 Arnold Kedi haritası (Arnold Cat map)

1960'larda Vladimir Arnold bir kedi görüntüsü örneği üzerinden haritanın etkilerini göstermektedir. Bu yüzden bu harita Arnold Cat Map olarak adlandırılmaktadır. (2.31)'deki gibi tanımlanmaktadır (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

$$\begin{aligned}x_{k+1} &= x_k + y_k \text{ mod}(1), \\y_{k+1} &= x_k + 2y_k \text{ mod}(1).\end{aligned}\tag{2.31}$$

$x_k \in (0,1)$ ve $y_k \in (0,1)$ olduğu açıkça görülmektedir (Lu vd., 2014).

Bir imgenin esnetildikten sonra kesilmesi ile elde edilen parçaların aynı birim kareye gelecek şekilde katlanması fikridir. İşlemlerin tekrarlanması ile imgenin hacmi korunmaktadır fakat oldukça karmaşık bir görüntü elde edilmektedir. Çok basit bir yöntemle karmaşık verilerin üretilmesi fikri rastgelelik için oldukça önemlidir. Fakat bu yöntemin tekrarlanma sayısı arttırıldığında orijinal görüntü ortaya çıkmaktadır. Bu durum yöntemin periyodik olduğu tartışmalarına yol açmaktadır.

2.2.2.2.4 Zaslavskii haritası (Zaslavskii map)

Zaslavskii haritası kaotik davranışa sahip ilginç bir dinamik sistemdir. Ayırıklaştırılmış denklem (2.32)'de verilmektedir.

$$\begin{aligned}x_{k+1} &= (x_k + v + ay_{k+1}) \text{ mod}(1), \\y_{k+1} &= \cos(2\pi x_k) + e^{-r} y_k.\end{aligned}\tag{2.32}$$

Zaslavskii haritası, $v = 400, r = 3$ ve $a = 12.6695$ için en büyük Lyapunov kuvvetine sahip bir haritadır (Pamuk, 2013). Bu durumda $y_{k+1} \in [-1.0512, 1.0512]$ olduğu hesaplanabilmektedir (DiCarlo, 2012; Dutang ve Wuertz, 2009; Ergün ve Özoğuz, 2010; Lu ve diğ., 2014; Ozer, 2010; Petura, 2020; Stipčević ve Koç, 2014; Yu ve diğ., 2019).

2.2.2.3 Kaos yöntemlerinin analizleri

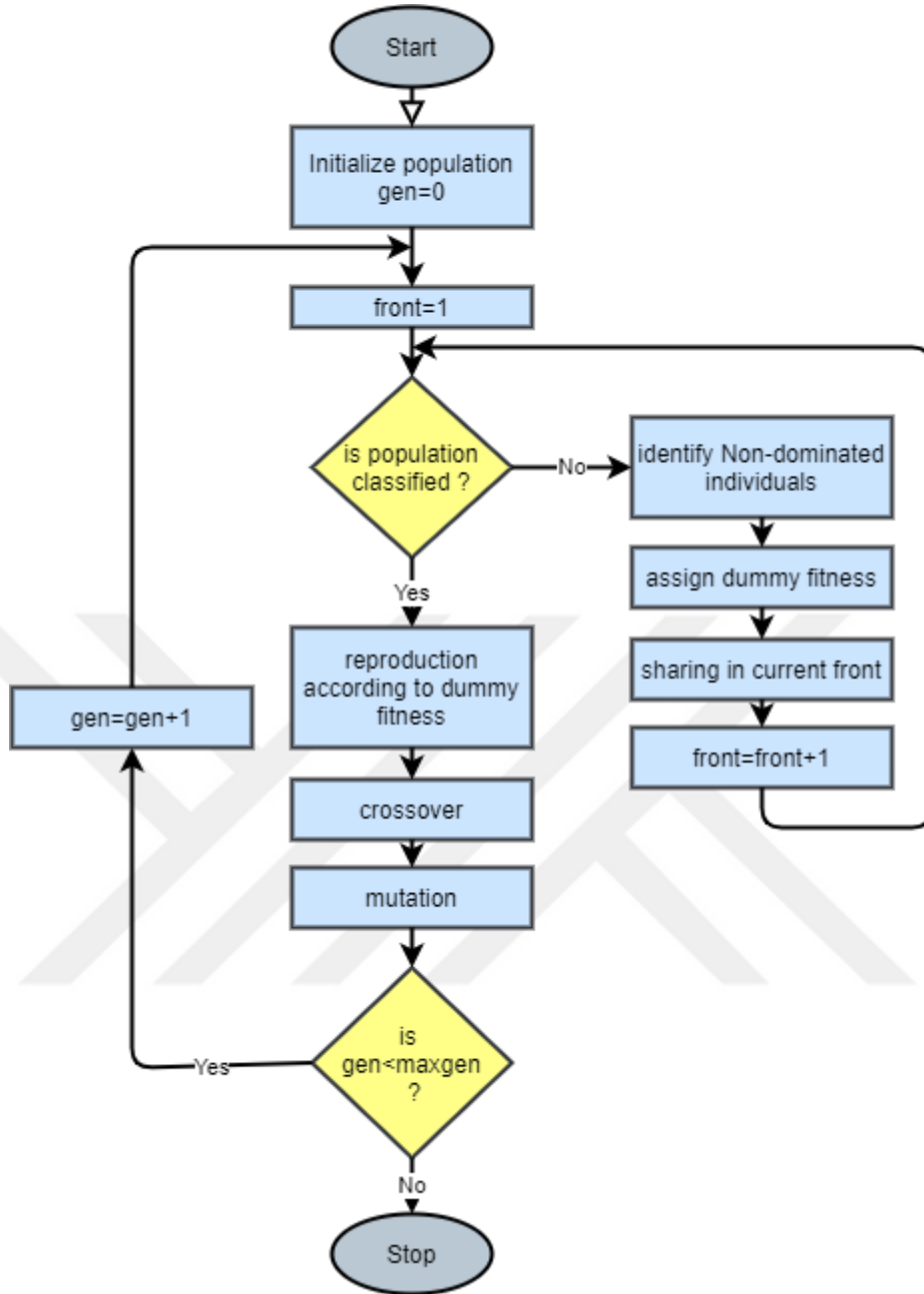
Kaotik dizilerin üretilmesi ve depolanmasının kolay ve hızlı olduğu kanıtlanmıştır, uzun dizilerin depolanmasına gerek yoktur. Çok uzun diziler için bile sadece birkaç fonksiyon (kaotik haritalar) ve birkaç parametre (başlangıç koşulları) gereklidir. Ek olarak, çok sayıda farklı dizi, basitçe başlangıç durumu değiştirilerek oluşturulabilir. Dahası, bu diziler deterministiktir ve tekrarlanabilir fakat yüksek karmaşıklığa sahiptir. Bir ve iki boyutlu türleri mevcuttur. İki boyutlu türleri değiştirilerek yüksek boyutlu kaotik haritalar elde edilmektedir. Bir boyutlu kaotik haritaların uygulanması kolaydır fakat sınırlı aralıklara sahip oldukları için saldırılara karşı zayıf olmaktadır. İki ve daha yüksek boyutlu kaotik haritaların yüksek karmaşıklığından dolayı tahmin edilebilirliği oldukça düşüktür. Tahmin edilemezlikleri yüksek güvenlik anlamına gelmektedir. Fakat bu yöntemlerin yüksek hesaplama maliyetleri mevcuttur ve uygulanması zor olmaktadır. Son zamanlarda rastgele diziler yerine kaotik diziler benimsenmiş ve güvenli iletişim ve doğrusal olmayan devreler, DNA hesaplama, görüntü işleme gibi alanlarda sıklıkla kullanılmaktadır.

2.3 Sezgisel Yöntemler (Heuristic Methods)

Sezgisel yöntemler, çözüm adaylarını oluştururken hem stokastik yöntemlerin hem de deterministik yöntemlerin avantajlı yönlerini kullanmaktadırlar. Bu yöntemler, hibridleme yaparak meydana gelmektedirler. Ayrıca bazı yöntemlerin iyileştirilmesiyle de oluşturulmaktadır.

2.3.1 Baskın olmayan sıralı genetik algoritma (non-dominated sorting genetic algorithm, NSGA)

Çok amaçlı problemlerin çözümü için kullanılan NSGA'lar, genetik algoritmanın bir uzantısıdır. 1994 yılında, Srinivas ve Deb tarafından önerilmiştir. Hem minimizasyon hem de maksimizasyon amaçlı işlemlerde kullanılmaktadır. NSGA, genetik algoritmanın temel yapısından sadece seçim operatörünün farklı olduğu bir yöntemdir. Diğer tüm genetik algoritma operatörlerini aynı şekilde içermektedir. NSGA'daki baskın olmayan sıralamanın amacı, bir sıralama seçim yöntemiyle iyi noktaları vurgulamak ve bir niche yöntemiyle de iyi noktaların sabit alt popülasyonlarını korumaktır. Şekil 2.3'te NSGA'nın akış diyagramı gösterilmektedir.



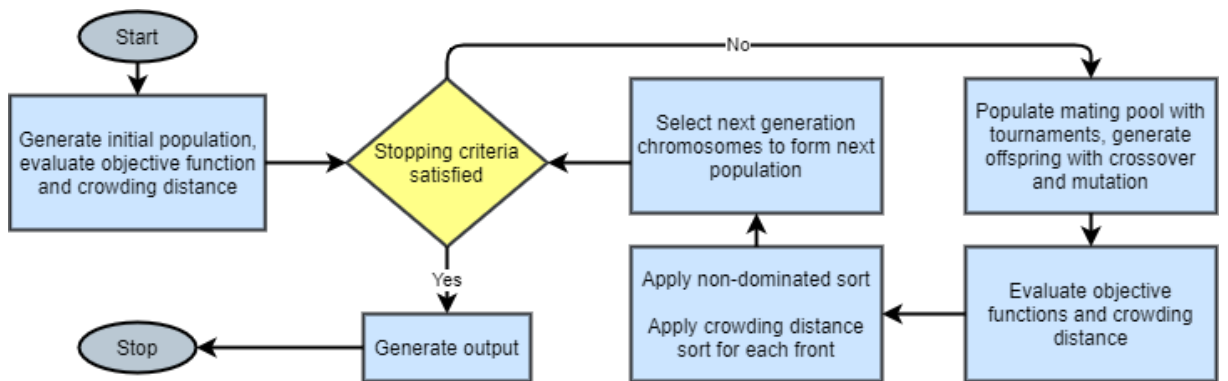
Şekil 2.3 : NSGA'nın akış diyagramı.

NSGA'da ilk olarak popülasyonun baskın olmayan bireyleri belirlenmektedir. Bu bireylere bir dummy fitness değeri atanmaktadır ve popülasyonun ilk baskın olmayan bireyleri olduğu varsayılmaktadır. Bu ilk baskın olmayan bireyler (popülasyonun ilk cephesi), popülasyondaki çeşitliliği korumak için atanan dummy fitness değerleri ile paylaşılmaktadır. Degrade fitness değerleri (bireylerin orijinal uygunluk değerlerinin etrafındaki bireylerin sayısına orantılı bir şekilde bölünmesi ile elde edilmektedir.) kullanılarak seçim işleminin yapılmasına paylaşım denilmektedir. Paylaşımın amacı, birden fazla optimal noktanın bir arada popülasyonda bulunmasını sağlamaktır. Bu işlemlerden sonra, ilk cephe haricindeki

bireyler içerisinde ikinci cepheyi belirlemek için aynı adımlar gerçekleştirilmektedir. Tüm bireylerin birkaç cephede şeklinde sınıflandırılmasıyla işlemler sona ermektedir. İkinci cephedeki bireylere atanan dummy fitness değerleri ilk cepheden düşük olmalıdır. Böylece en yüksek dummy fitness değerine sahip cephe, ilk cephe olmaktadır. Popülasyon seçilirken en çok birey ilk cepheden seçilmektedir. Bu da NSGA'nın amacına uygundur (Deb ve diğ., 2000, 2002; Gadhvi ve diğ., 2016; R. Kaur ve Kaur, 2020; Saraç ve Bilgiçer, 2020; Srinivas ve Deb, 1994; Wang, 2016; Yakıcı, 2018).

2.3.1.1 NSGA-II

NSGA'nın, üç eksikliğini gidermek için 2000 yılında Deb ve arkadaşları tarafından önerilmiştir. Bu üç eksiklik sırasıyla; baskın olmayan sıralamanın yüksek hesaplama karmaşıklığı ($O(mN^3)$), elitizm eksikliği ve Qshare paylaşım parametresini belirleme ihtiyacıdır. NSGA-II, hızlı ve elitist NSGA olarak da adlandırılmaktadır. NSGA'nın $O(mN^3)$ olan zaman karmaşıklığı $O(mN^2)$ 'ye düşürülmüştür. Bunu yapmak için ek depolama alanları kullanılmaktadır. Bu yüzden, NSGA'nın bellek karmaşıklığı $O(N)$ iken NSGA-II'nin $O(N^2)$ 'dir. NSGA-II'nin elitist özelliği ise, çaprazlama ve mutasyon adımlarından sonra bir sonraki neslin seçimindeki farklılıktır. Buradaki seçme işlemi, ilk popülasyonu oluşturmada kullanılan seçim yöntemi ile aynıdır. Son olarak, NSGA'daki Qshare parametresinin yerine kalabalık mesafesi kullanılmaktadır. Şekil 2.4'te NSGA-II'nin akış diyagramı gösterilmektedir (Deb ve diğ., 2000, 2002; Gadhvi ve diğ., 2016; R. Kaur ve Kaur, 2020; Saraç ve Bilgiçer, 2020; Srinivas ve Deb, 1994; Wang, 2016; Yakıcı, 2018).

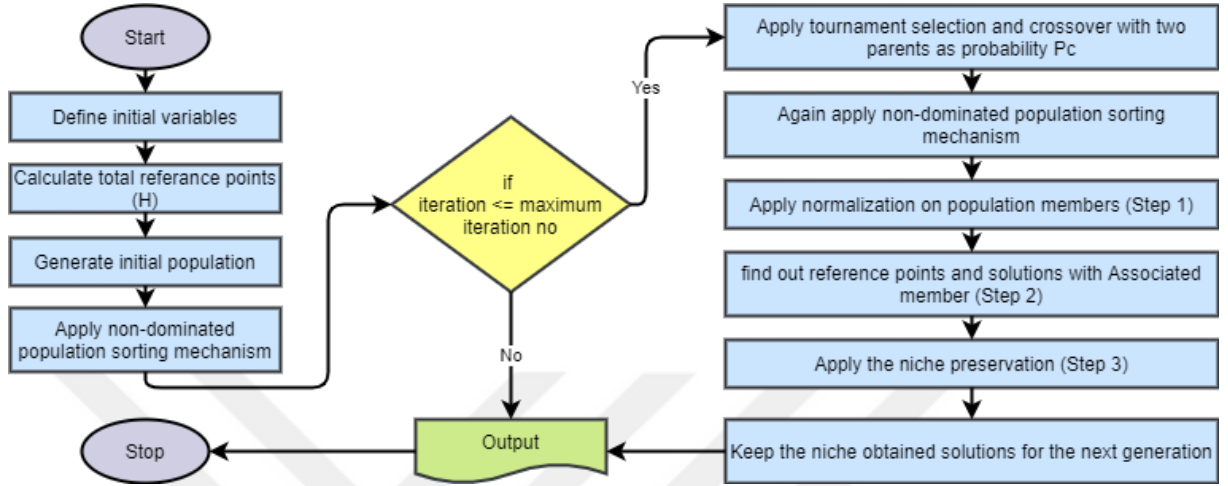


Şekil 2.4 : NSGA-II'nin akış diyagramı.

2.3.1.2 NSGA_III

NSGA-III, NSGA-II'nin seçim mekanizmasında önemli değişiklikler yapılarak 2013'te Deb ve Jain tarafından önerilmiştir. NSGA-III, daha iyi aday çözümler kümesi elde etmek için

referans noktası mekanizmasını kullanmaktadır. Referans noktası mekanizması, NSGA-II'deki kalabalık mesafesi işleminin yerine kullanılmaktadır. Şekil 2.5'te NSGA-III'ün akış diyagramı gösterilmektedir (Bhesdadiya ve diğ., 2016; Deb ve Jain, 2014; Liu ve diğ., 2019; Tavana ve diğ., 2016; C. Zhang ve diğ., 2019).



Şekil 2.5 : NSGA-III'ün akış diyagramı.

2.3.2 Uyarlanabilir rastgelelik (adaptive randomness, AR)

Klasik rastgele yöntemleri, başlangıç popülasyonun eşit dağıtımını yapamamaktadır. Elde edilen sayılar, klasik yöntemlerde de bir formülle üretilmektedir. Uyarlanabilir rastgelelik yönteminde, birbirine en uzak bireyler seçilmesi ile başlangıç popülasyonun eşit dağıtılması amaçlanmaktadır.

1. Başlangıç popülasyonu $P(ps) = \{X_1, X_2, \dots, X_{ps}\}$ olsun, ps popülasyon büyüklüğü ve X_i ($i = 1, 2, \dots, ps$) D -boyutlu bir uzayda aday çözümlerdir.
2. Kısmi başlangıç popülasyonu $PP (PP \subseteq P)$ tanımlanmaktadır.
3. Deneme bireyleri $ST(k) = \{Y_1, Y_2, \dots, Y_k\}$ tanımlanmaktadır. k deneme bireyleri kümesidir ve $ST \cap P = \emptyset$ 'dir.
4. Her deneme bireyi Y_i ($i = 1, 2, \dots, k$) D -boyutlu uzaydan rastgele seçilmektedir ve k önceden tanımlanmış sayıda deneme bireyleridir.

ST , arama uzayından rastgele seçilen ama PP 'ye eklenmemiş bireylerden oluşmaktadır. PP 'ye her birey ekleme işleminde önerilen AR yöntemi, çözüm uzayında k tane bireyden ST 'yi oluşturmaktadır. Sonrasında PP 'ye aktaracak bireyi, PP 'deki tüm bireylerden mesafe olarak en uzak olan bireyi ST 'den seçmektedir. Burada mesafe ölçümünü Öklid mesafe

formülü ile gerçekleştirmektedir. Bu işlem, popülasyon büyüklüğüne ulaşıncaya kadar devam etmektedir. Böylelikle *PP*, eşit dağılımlı bir başlangıç popülasyonu olarak elde edilmektedir (Pan ve diğ., 2014).

2.3.3 Uyarlanabilir genetik algoritma ile başlatma (adaptive genetic algorithm Initialize, AGA)

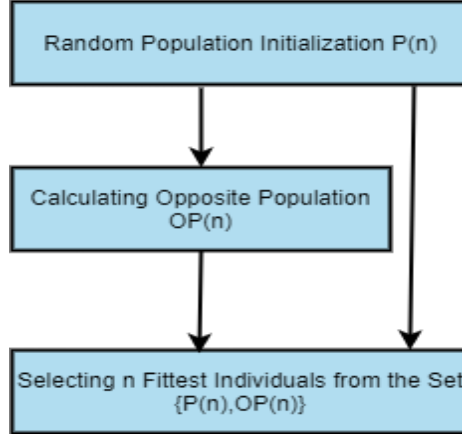
K-means algoritması için bir başlangıç popülasyonu oluşturma tekniği önerilmektedir. Düşük boyutlu veri setleri için geliştirilen teknik, kullanıcılar tarafından girilen küme sayısı girilme zorunluluğunu ortadan kaldırmaktadır. Yani otomatik k küme sayısını bulmaktadır. Ayrıca yakınsama hızı ve sonuçların kalitesinin yükseltilmesi için en iyi kromozom seçimi de amaçlanmaktadır. Bu yöntem, Genetik Algoritmasının (GA) geliştirilmiş bir versiyonu olan Uyarlanabilir Genetik Algoritma ile adlandırılmaktadır. Başka çalışmalarda GA, k-means algoritması için başlangıç popülasyonunu oluşturmada kullanılmıştır. Sonuçlar iyi olmasına rağmen karmaşıklığı yüksek çıkmaktadır. AGA, düşük karmaşıklığa sahiptir. AGA ile k-means birleşiminden oluşturulmuş yeni bir kümeleme algoritması olan SeedClust önerilmektedir (Zhou ve diğ., 2018).

SeedClust'ın içeriği;

- Geliştirilmiş k-means++ (k-means için basit ve hızlı bir tohum seçme yöntemidir) ve düşük karmaşıklığa sahip yoğunluk tahmin yönteminin birleştirilmesiyle ilk popülasyon üretilmektedir.
- SeedClust'ta nüfus çeşitliliğini arttırmak ve yerel optimuma takılmamak için AGA'nın kullanılması.
 - 1. adım: Uyarlanabilir crossover işlemi
 - 2. adım: Uyarlanabilir mutasyon işlemi
 - 3. adım: Elitizm işlemi
 - 4. adım: En iyi kromozomu elde etmek
- En iyi kromozomlar ile kümeleme işlemini yapan k-means kullanılarak kaliteli sonuçlar elde edilmektedir.

2.3.4 Muhalefet tabanlı başlangıç popülasyonu (opposite based initial population)

Muhalefet tabanlı öğrenme algoritması, Şekil 2.6'da gösterilmektedir.



Şekil 2.6 : Muhalefet tabanlı popülasyon başlangıcı.

Bu yöntem aşağıdaki gibi iki şekilde tanımlanmaktadır.

Tek boyutlar için, x ($x \in [a, b]$) gerçek sayı ve \check{x} ise karşıt sayı olarak (2.33)'teki denklemle hesaplanmaktadır (Dong ve diğ., 2012; Kazimipour ve diğ., 2014; Rahnamayan ve diğ., 2007; Xu ve diğ., 2014).

$$\check{x} = a + b - x \quad (2.33)$$

$a = -b$ için $\check{x} = -x$ ve $a = 0, b = 1$ için $\check{x} = 1 - x$ alınmaktadır.

Yüksek boyutlar için (2.34)'teki denklemle hesaplanmaktadır. $P(x_1, x_2, \dots, x_n)$, n-boyutlu uzayda bir nokta olsun. Burada $x_1, x_2, \dots, x_n \in R$ ve $x_i \in [a_i, b_i] \forall i \in \{1, 2, \dots, n\}$ 'dir. P 'nin karşıt noktaları $\check{P}(\check{x}_1, \check{x}_2, \dots, \check{x}_n)$ 'dür (Dong ve diğ., 2012; Kazimipour ve diğ., 2014; Rahnamayan ve diğ., 2007; Xu ve diğ., 2014).

$$\check{x}_i = a_i + b_i - x_i \quad (2.34)$$

2.3.5 Farklı rastgelelik (distinct random)

Çözüm adaylarının tümünü rastgele seçmek yerine, popülasyonun 10 katı kadar rastgele bireyler oluşturup içinden sezgisel bir yöntemle en belirgin başlangıç popülasyonu elde etmektir. Sezgisel yöntem olarak da Hamming mesafesi kullanılmaktadır (Deniz ve Kiziloz, 2019).

2.3.6 Bilgi kazancı sıralaması (information gain ranking)

En umut verici popülasyonu oluşturmak için bilgi kazancı yöntemi kullanılmaktadır. İlk birey en umut verici tek özellikli bireyler arasından seçilmektedir. Aynı şekilde ikinci birey

ise, en umut verici iki özellikli bireyler arasından seçilmektedir. Bu işlem popülasyon boyutuna ulaşıldığında bitmektedir (Deniz ve Kiziloz, 2019).

2.3.7 Açgözlü yaklaşım (greedy)

Özellik seçimi adımından sonra tek özelliği seçilmiş bireyler arasından doğruluğu en yüksek birey, popülasyonun ilk bireyi olarak seçilmektedir. Seçilen bireyin özelliğini barındıran ve iki özellikli bireyler arasından da yine doğruluğu en yüksek olan birey, popülasyonun ikinci bireyi olarak seçilmektedir. Bu işlem popülasyon boyutuna ulaşıldığında bitmektedir (Deniz ve Kiziloz, 2019).

2.3.8 Küçük rastgele (small random)

Rastgele yöntemler ile açgözlü yaklaşıma benzerlik göstermektedir. Popülasyonun İlk bireyi tek özellikli bireyler arasından rastgele seçilmektedir. İlk bireyin özelliğini barındıran iki özellikli bireyler arasından rastgele bir birey seçilmektedir ve bu popülasyonun ikinci bireyidir. Popülasyon boyutuna ulaşınca kadar bu işlem devam etmektedir (Deniz ve Kiziloz, 2019).

2.3.9 Fidan gelişim algoritması (saplings growing up algorithm, SGuA)

SGuA 2006 yılında Ali Karcı tarafından geliştirilmiştir. Önerilen yöntem isminden de anlaşıldığı gibi fidanların gelişimlerinden esinlenilerek ortaya atılan bir optimizasyon tekniğidir. Fidan ekimi, fidan büyümesi ve fidan eşleşmesine dayanan arama ve optimizasyon problemleri için sunulan yeni bir hesaplama yöntemidir. Fidanların ekilmesi, her yönden birbirine (batı, doğu, kuzey, güney) eşit uzunlukta bir mesafeye sahip olmalıdır ve bu yöntemin ilk basamağıdır. Fidan büyüme aşamasında fidanlar eşleşir, dallanır ve aşılır. Bu nedenle üç operatör söz konusudur. Bunlar eşleştirme, dallanma ve aşılama operatörleridir (Demir ve diğ., 2011; Gündoğan ve diğ., 2004; Karcı, 2004, 2007; Seyyarer ve diğ., 2018).

SGuA yöntemi aşağıdaki gibi açıklanabilir;

- Ekim Aşaması

Düzgün ekim örnekleme: Fidanlar uygun olan çözüm alanına eşit şekilde dağılmaktadır.

- Büyüme Aşaması

Bu aşama üç operatör içermektedir.

- ✓ Eşleşme operatörü; mevcut fidanları eşleştirerek yeni fidanlar oluşturmayı amaçlamaktadır. Eşleşme operatörü, iki fidan arasında bilgi alışverişinde bulunan global bir arama operatörüdür.
- ✓ Dallanma operatörü; mevcut dal pozisyonlarına bağlı olarak dal pozisyonunun belirlenmesi için olasılık yöntemini kullanarak mevcut fidanlardan yeni fidanlar üretmeyi amaçlamaktadır.
- ✓ Aşılama operatörü; mevcut fidanlardan benzer yeni fidanlar üretmeyi amaçlamaktadır. Aşılama operatörü, benzer fidanları kullanan bir arama operatörüdür.

Algoritma 2.1 : Başlangıç Popülasyonunu Oluşturmak (Demir ve diğ., 2011; Gündoğan ve diğ., 2004; Karci, 2004, 2007; Seyyarer ve diğ., 2018).

//G bir bahçe, I indis kümesi ve Ie genişletilmiş indis kümesi.

14- Dallanma değerlerinin alt sınırlarını içeren G[1] ve üst sınırlarını içeren G[2] gibi iki tane fidan oluştur.

15- Indis←3

16- k←2

17- **While** P doymamış **do**

ie, Ie'nin bir elemanı olsun, her ie bit değeri ile genişletilmiş ve bu bit değeri parçaya karşılık gelir.

i←1

While P doymamış ve özel bir k ($i \leq 2k-2$) değeri için tüm fidanlar üretilmemiş **do**

i bir k-bit sayı ve ie i'nin genişletilmiş değerine karşılık gelir.

i'nin her biti G[0] ve G[1]'in karşılık gelen kısmının

uzunluğuna kadar genişletilir.

For j←1 **to** n **do**

if ie'nin j. biti=1 **then** P[indis]'in j. dallanması=G[1]*r

else G[indis]'in j. dallanması=G[2]*r

r=random[0,1] ve reel sayıdır.

indis←indis+1

i←i+1

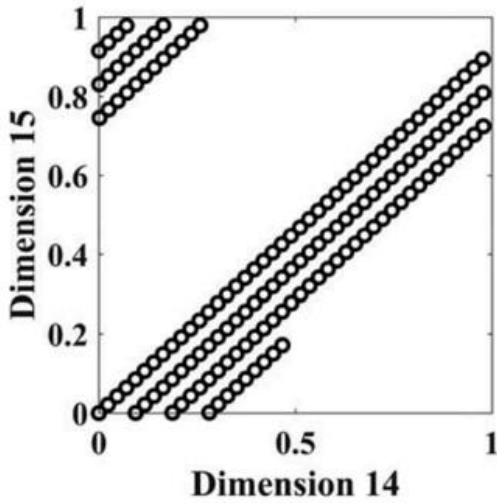
k←k+1

SGuA'nın başarılı olmasındaki en büyük etken başlangıç popülasyonunun düzenli olarak başlatılmasıdır. Bu başlangıç popülasyonu ile çözüm uzayından fazla uzaklaşmadan ve fazla zaman harcamadan en uygun çözüme ulaşılması hedeflenmektedir (Demir ve diğ., 2011; Gündoğan ve diğ., 2004; Karci, 2004, 2007; Seyyarer ve diğ., 2018).

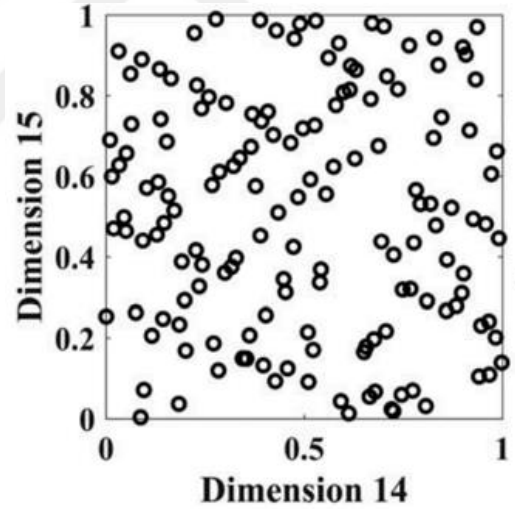
2.3.10 Karıştırılmış halton dizisi (scrambled halton sequence)

Halton dizisi, büyük boyutlarda dizi birleştirmesinde dezavantajlı olmaktadır. Büyük boyutlarda sırayla asal sayıların gelmesi ile zor korelasyonla karşılaşılması mümkündür. Bu sorun, parçacıkların çözüm uzayını tam olarak kaplamamasına neden olmaktadır. Sorunu gidermek için halton dizisi oluşturulurken (2.35)'teki denklem önerilmektedir (Digehsara ve diğ., 2020; Morokoff ve Caflisch, 1994; Zeng, 2016). Şekil 2.7'de standart halton dizisinin ve Şekil 2.8'de karıştırılmış halton dizisinin çözüm uzayı dağılımı gösterilmektedir.

$$\varphi_r(g) = \sum_{l=0}^L \sigma_r(b_l(g))r^{-l-1} \quad (2.35)$$

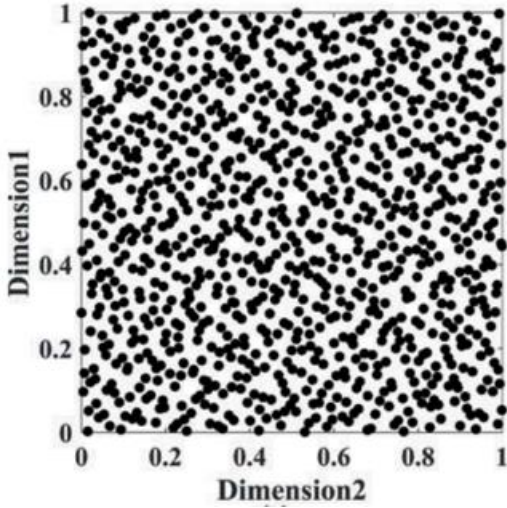


Şekil 2.7 : Standart halton dizisi.

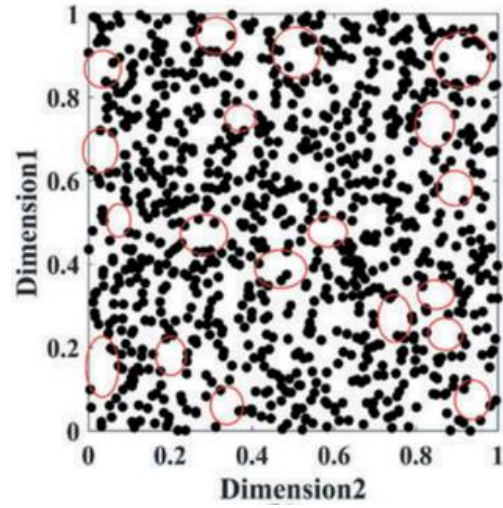


Şekil 2.8 : Karıştırılmış halton dizisi.

Karıştırılmış Halton dizisi standart Halton dizisinin geliştirilmiş bir versiyonudur. Çözüm uzayını kapsayan parçacıklar bakımından hem standart Halton dizinden hem de rastgele başlatmadan oldukça başarılı olduğu gözlemlenmiştir. Şekil 2.9'da karıştırılmış Halton dizisinin ve Şekil 2.10'da rastgele başlamanın çözüm uzayı dağılımı gösterilmektedir.



Şekil 2.9 : Karıştırılmış Halton dizisi.



Şekil 2.10 : Rastgele başlatılan popülasyon.

2.3.11 Genetik çeşitlilikle uygulanabilir başlangıç popülasyonu (feasible initial population with genetic diversity)

Popülasyon çeşitliliğindeki amaç, daha uygulanabilir çözümlerle başlangıç popülasyonunu oluşturmaktır. Çeşitliliği arttırmak için iki aşamadan oluşan bir algoritma önerilmektedir (Cruz-Chávez ve Martínez-Oropeza, 2016).

1. İlk Aşama:

Yarı uygulanabilir çözümler elde etmek için kullanılan aşamadır. Bu aşamada k-means kümeleme algoritmasının değiştirilmiş bir versiyonu kullanılmaktadır. K-means'in dezavantajları aşağıdaki değişikliklerle giderilmiştir.

- K küme sayısına gerek yoktur.
- Her centroid bir müşteriyi temsil etmektedir.
- Sadece bir kez öklid mesafesi hesaplanmaktadır.
- Centroidleri tekrar tekrar hesaplamaya gerek yoktur.
- Kriter olarak maksimum araç kapasitesi ve müşterilerin talebi dikkate alınmaktadır.

2. İkinci Aşama:

İlk aşamadan alınan yarı uygulanabilir çözüm popülasyonunun tam uygulanabilir hale gelmesi amaçlanmaktadır. Bu aşamada 4 ekleme sezgiseli (nearest route (NR), direct/swap insertion (D/S), swap insertion (S), insertions 1 by 1 and 2 by 2 (1-1 and 2-2)) ve bunların birleşiminden oluşan hibrid bir yaklaşım önerilmektedir.

- i. En Yakın Rota (NR): Başlangıçta rastgele atanmamış bir müşteri seçilmektedir, sonrasında en yakın rota aranmaktadır. Kısıtlamalar değerlendirildikten sonra kriterlere uygunsuz müşteri atanmaktadır. Değerlendirme kriterlerine uygun değilse başka bir rota seçilmektedir. Müşteri seçilene kadar işlem tekrar edilmektedir.
- ii. Doğrudan / Takas Ekleme (D / S): atanmamış bir müşteriyi değerlendirme kriterleri uygunluğuna göre en yakın rotaya eklemeye çalışmaktadır. Uygun değilse de müşteri eklenir ve başka bir rota rastgele seçilmektedir. İki rota arasında müşteri değişimi yapılmaktadır. Değerlendirme kriterlerine uygunluğu varsa başarılı işlem sayılmaktadır ve müşteri atanmaktadır. Kriterlere uygunluk yoksa her iki rotadan vazgeçilmektedir ve işlem yeni rotalar seçerek yoluna devam etmektedir. Müşteri ataması yapılana kadar işlem tekrar edilmektedir.
- iii. Takas Ekleme (S): rastgele bir rota ve atanmamış bir müşteri seçilmektedir. Değerlendirme kriterlerine bakılmaksızın müşteri rotaya eklenmektedir. Sonrasında düzenli olarak bir rota seçilmekte ve müşteri değişimi yapılmaktadır. Kriterlere uygunluğu oluşana kadar işlem tekrar edilmektedir.
- iv. 1'e 1 ve 2'ye 2 (1-1 ve 2-2) Takas: iki farklı değişim hareketiyle müşteri değişimi yapılmaktadır. İlk müşteri seçildikten sonra takas ekleme işlemi ile devam edilmektedir. Değerlendirme kriterlerine uygunluğu oluşmuşsa müşteri eklenmektedir, uygun değilse iki değişim yaparak süreç yeniden devam etmektedir. Müşteri eklenene kadar işlem sürdürülmektedir.

Önerilen iki aşamalı yöntemden sonra başlangıç popülasyonunun seçimi, Hamming mesafesi ile yapılmaktadır. Hamming mesafesi çeşitlilik için çok kullanılan bir yöntemdir (Cruz-Chávez ve Martínez-Oropeza, 2016).

2.3.12 Sezgisel bir yöntemle başlangıç popülasyonu çeşitliliği (initial population diversity with a heuristic method)

Yapay arı algoritmasında (ABC) klasik rastgele başlatma yöntemi yerine sezgisel bir yaklaşım ile ABC'nin performansının artırılması amaçlanmaktadır. Bu yaklaşım, başlangıç popülasyonundaki çeşitliliği artırarak kaliteli çözümler üretmektedir ve yerel optimum noktalardan kaçınmaktadır. Ayrıca çözüme yakınsama zamanını da azaltmaktadır (Pian ve diğ., 2018). Algoritma 2.2'de, bu yaklaşımın sözde kodu gösterilmektedir.

Algoritma 2.2 : Sezgisel Bir Yöntemle Popülasyon Çeşitliliği (Pian ve diğ., 2018).

- 1- Popülasyonu başlatın ve popülasyon sayısı NP'dir, $i=1, 2, \dots, NP$; $j=1, 2, \dots, D$.
- 2- $X(NP)$ başlangıç popülasyonunu denklem (2.36)'yı kullanarak oluşturun.

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}) \quad (2.36)$$

- 3- $OX(NP)$ başlangıç popülasyonunun tersini denklem (2.37)'yi kullanarak oluşturun.

$$ox_{i,j} = x_{min,j} + (x_{max,j} - x_{i,j}) \quad (2.37)$$

- 4- En iyi NP uygunluğu $\{X(NP) \cup OX(NP)\}$ içinden başlangıç popülasyonu olarak seçilir.
-

2.3.13 Sezgisel yöntemlerin analizleri

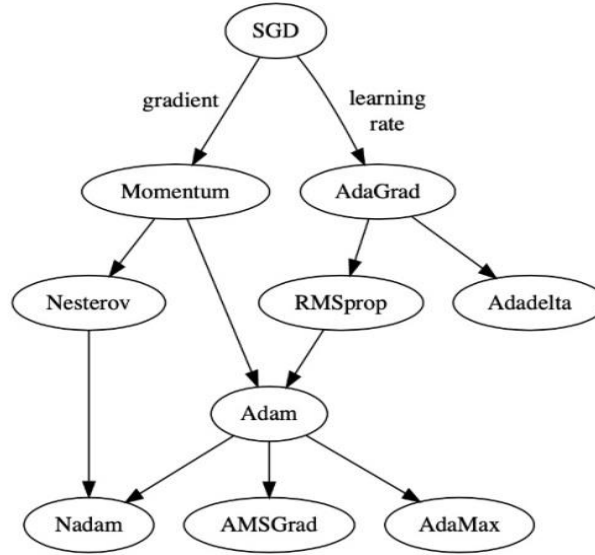
Bu yöntemlerin amacı, var olan yöntemlerin dezavantajlarını gidermek için eklentiler yapmaktır ya da birden fazla yöntemi birleştirerek daha etkin yöntemler geliştirmektir. Son yıllarda sezgisel yöntemlerle ilgilenen araştırmacıların sayısı oldukça fazladır. Çok başarılı sonuçlar elde edilmektedir. Fakat bazıları ise sadece özel bir çalışma ile sınırlı kalmaktadır. NSGA, SGuA ve Muhalefet Tabanlı Başlangıç Popülasyonu (Opposite Based Initial Population) yöntemleri oldukça popülerdir. GA'nın başlangıç popülasyonu oluşturma kısmı değiştirilerek elde edilen NSGA'lar çok amaçlı problemlerde oldukça başarılı performans göstermektedirler. Düzenli popülasyon oluşturmak için önerilen SGuA ve çözüm uzayında çeşitliliği arttırmak için önerilen Muhalefet Tabanlı Başlangıç Popülasyon yöntemleri de birçok çalışmada kullanılmaktadır.

3. TÜREV DENKLEMLERİ

Makine öğrenmesinde hata oranını en aza indirmek için sıklıkla kullanılan altı tane optimizasyon yöntemi vardır. Bunlar; SGD, Momentum, Adagrad, RMSProp, Adadelta ve Adam yöntemleridir. Bu yöntemlere Dik İniş (Gradient Descent) yöntemleri de denilmektedir. Çizelge 3.1’de dik iniş yöntemlerinin özellikleri karşılaştırılarak sunulmaktadır. Şekil 3.1’de ise dik iniş yöntemlerinin evrimsel haritası gösterilmektedir.

Çizelge 3.1 : Dik iniş yöntemlerinin karşılaştırılması.

Yöntem	Tarih	Öğrenme Katsayısı	Gradient
SGD	1951	√	√
Momentum	1964		√
Adam	2014	√	√
AdaGrad	2011	√	
RMSprop	2012	√	
Adadelta	2012	√	



Şekil 3.1 : Dik iniş yöntemlerin evrimsel haritası.

3.1 SGD

Literatürdeki birçok çalışmada Dik İniş ismi ile kullanılmaktadır. SGD, tüm türevler yerine rastgele olmak üzere bir kısım türevle ağırlıkları güncellemektedir. Mevcut türevi ($\partial L / \partial w_t$), öğrenme katsayısı (a) ile çarparak mevcut ağırlığı (w_t) güncellemektedir (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

$$w_{t+1} = w_t - a \frac{\partial L}{\partial w_t} \quad (3.1)$$

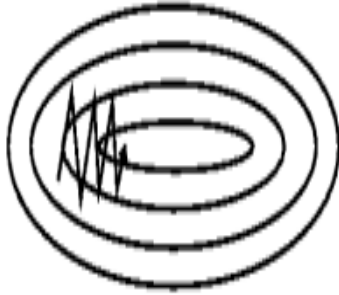
3.2 Momentum

SGD’de optimum nokta aranırken çok fazla salınım olmaktadır. Bu salınımları azaltmak ve dolayısıyla hedefe gitme hızını arttırmak için momentum yöntemi önerilmektedir. Bu yöntemde mevcut türevler yerine momentumlu türev kullanılmaktadır. Aslında momentum ismi biraz talihsizce olmuştur, bu yöntemin isminin tam karşılığı kontrollü inme olabilirdi (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

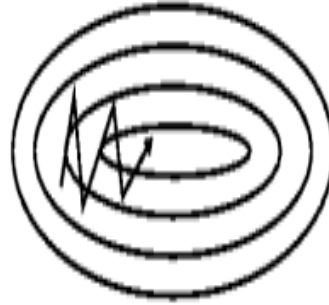
$$w_{t+1} = w_t - aV_t \quad (3.2)$$

$$V_t = \beta V_{t-1} + (1 - \beta) \frac{\partial L}{\partial w_t} \quad (3.3)$$

Burada V_t ’nin başlangıç değeri 0’dir. β , 0 ile 1 arasında olup genelde kullanılan değeri ise 0.9’dur ve geçmiş türevlerin işleme ne kadar katılacağını ayarlamak için kullanılmaktadır. Şekil 3.2’de momentumuz SGD’nin çözüme giderken çok fazla salınım yaptığı görülmektedir. Buna karşılık Şekil 3.3’te momentumlu SGD, daha az salınım yaparak çözüme ulaşmıştır (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).



Şekil 3.2 : Momentumsuz SGD.



Şekil 3.3 : Momentumlu SGD.

3.3 Adagrad

SGD ve Momentum yöntemlerindeki sabit öğrenme katsayısı problemini ortadan kaldırmak için önerilmiştir. Her adımda farklı bir öğrenme katsayısı ile işlem yapmak için, eşitlik 3.1’deki öğrenme katsayısını geçmiş türevlerin karelerinin kümülatif toplamının kareköküne bölmektedir (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (3.4)$$

$$S_t = S_{t-1} + \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (3.5)$$

Burada S başlangıçta 0, ϵ ise işlemi sıfıra bölme hatasından kurtarmak için genelde çok küçük bir sayı (10^{-7}) olarak alınmaktadır (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

3.4 RMSprop

Adagrad'da olduğu gibi sabit öğrenme katsayısı problemini çözmek için önerilmiştir. Aralarındaki fark ise, adagrad yöntemindeki türevlerin karelerini almak yerine momentumlu türevlerin karelerini almaktadır (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

$$w_{t+1} = w_t - \frac{a}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (3.6)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (3.7)$$

Burada S başlangıçta 0, $a=0.001$, $\beta=0.9$ ve ϵ ise 10^{-6} olarak tercih edilmektedir (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

3.5 Adadelta

Adadelta ve RMSProp yaklaşımları aynı yıl içerisinde farklı kişiler tarafından önerilmiştir. Her iki yöntem de, sabit öğrenme problemini çözmeye odaklanan adagrad versiyonlarıdır. Adadelta yönteminde, adagrad ve RMSProp yöntemlerinden farklı olarak öğrenme katsayısı seçme zorunluluğu yoktur. Öğrenme katsayısı yerine, geçerli ağırlıklar ile güncellenen ağırlıklar arasındaki farkı ifade eden delta değerlerinin karelerinin momentumlu toplamları kullanılmaktadır (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} - 1}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \quad (3.8)$$

$$D_t = \beta D_{t-1} + (1 - \beta) [\Delta w_t]^2 \quad (3.9)$$

$$S_t = \beta S_{t-1} + (1 - \beta) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (3.10)$$

$$\Delta w_t = w_t - w_{t-1} \quad (3.11)$$

Burada S ve D başlangıçta 0, $\beta=0.95$ ve ϵ ise 10^{-6} olarak tercih edilmektedir (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

3.6 Adam

Rmsprop ve momentum yöntemlerinin avantajlı yönlerinin birleştirilmesi ile önerilen dik iniş algoritmasıdır. Momentum yöntemindeki V ile rmsproptaki S 'i kullanmaktadır (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

$$w_{t+1} = w_t - \frac{a}{\sqrt{\hat{S}_t + \epsilon}} \cdot \hat{V}_t \quad (3.12)$$

$$\hat{V}_t = \frac{V_t}{1 - \beta_1^t} \quad (3.13)$$

$$\hat{S}_t = \frac{S_t}{1 - \beta_2^t} \quad (3.14)$$

$$V_t = \beta_1 V_{t-1} + (1 - \beta_1) \frac{\partial L}{\partial w_t} \quad (3.15)$$

$$S_t = \beta_2 S_{t-1} + (1 - \beta_2) \left[\frac{\partial L}{\partial w_t} \right]^2 \quad (3.16)$$

Burada S ve V başlangıç olarak 0, $a=0.001$, $\beta_1=0.9$, $\beta_2=0.999$ ve ϵ ise 10^{-8} olarak tercih edilmektedir (Karim, 2019; Lydia ve Francis, 2019; Ruder, 2016; Yazan ve Talu, 2017; Zeiler, 2012).

4. GELİŞTİRİLEN YÖNTEM; DETERMİNİSTİK METASEZGİSEL ALGORİTMA

4.1 Problemin Formülasyonu

Algoritma 4.1 ve 4.2’de stokastik ve deterministik popülasyon başlatma yöntemlerinin detayları sunulmuştur.

Algoritma 4.1 : Stokastik başlatma yöntemi.

```
def StochasticInitPopulation(LB,UB,X):
```

```
    for i in range(X.shape[1]):
```

```
        interval =(UB[i]-LB[i])
```

```
        for j in range(X.shape[0]-2):
```

```
            X[j+1,i]=LB[i]+random(0,1)*
```

```
interval
```

```
    return X
```

Algoritma 4.2 : Deterministik başlatma yöntemi.

```
def DeterministicInitPopulation (LB,UB,N,X):
```

```
    for i in range(X.shape[1]):
```

```
        interval =(UB[i]-LB[i])/(N-1)
```

```
        temp=0
```

```
        for j in range(X.shape[0]-2):
```

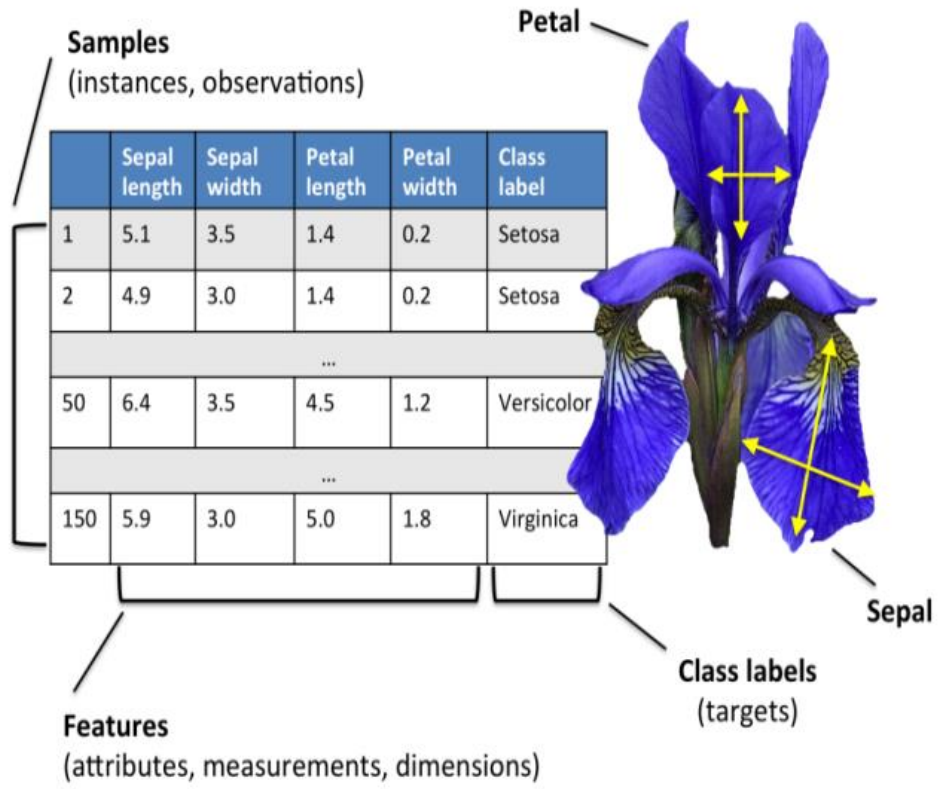
```
            temp=temp+ interval
```

```
            X[j+1,i] = LB[i]+temp
```

```
    return X
```

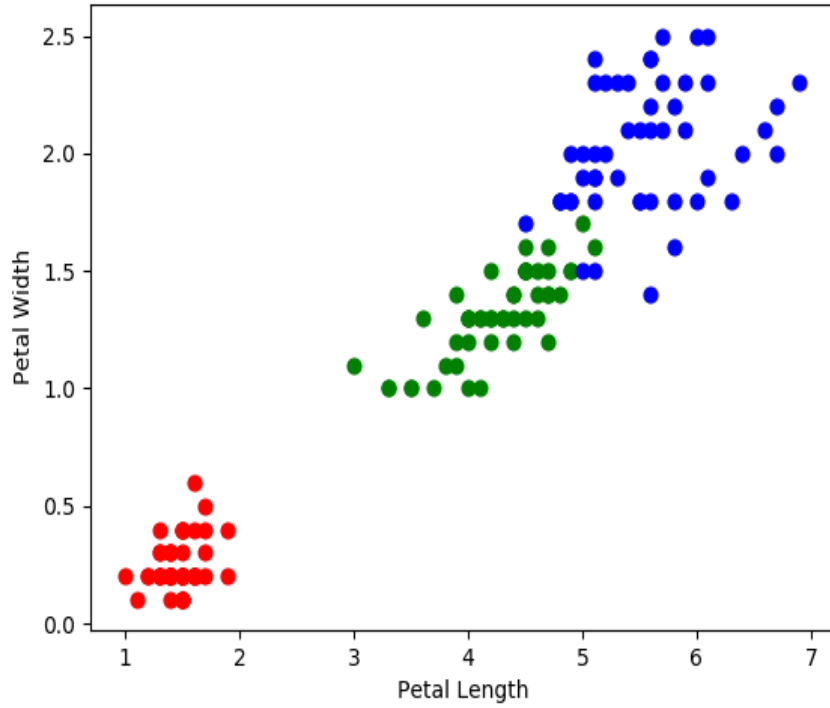
LB: Alt sınır, UB: Üst sınır, N: Popülasyon boyutu, X: Popülasyon matrisi

MLR’yi uyguladığımız iris data seti 1936’da R.A. Fisher tarafından oluşturulmuş, 1988’de Michael Marshall tarafından da kullanıma sunulmuştur. Literatürde örüntü tanıma ile ilgili belki de en iyi bilinen data settir. Üç sınıftan (iris setosa, iris virginica and iris versicolor) oluşmaktadır ve her sınıfın 4 tane özneliği (sepal length, sepal width, petal length, petal width) mevcuttur. Her bir sınıftan 50’şer olmak üzere toplamda 150 tane örnek bulunmaktadır (Dua ve Graff, 2019). Şekil 4.1’de iris bitkisinin tüm öznelikleri, Şekil 4.2’de ise petal length ve petal width özneliklerinin grafiksel gösterimi sunulmuştur.



Şekil 4.1 : İris bitkisinin öznelikleri.

IRIS DATA | Red - Iris-setosa, Green - Iris-versicolor, Blue - Iris-virginica



Şekil 4.2 : İris bitkisinin özneliklerinin dağılımı.

İris veri seti için (4.1)'deki denklem temel alınmaktadır.

$$Y = \theta_1 X_{petal_width} + \theta_2 X_{petal_length} \quad (4.1)$$

Y , X_{petal_width} ve X_{petal_length} değerleri iris veri setinden okunarak θ_1 ve θ_2 katsayıları tahmin edilmeye çalışılmaktadır. θ_1 ve θ_2 katsayıları, genellikle başlangıç olarak 0 veya 1 alınmaktadır ya da iki değer arasında rastgele üretilmektedirler. İşlemler sonucunda tahmin edilen katsayılar ile X giriş değerleri fonksiyonda yerine konularak bir Y' elde edilmektedir. Y gerçek değer ile Y' tahmin edilen değer arasındaki fark, hata miktarımızı oluşturmaktadır.

İris veri setinin genel görünümü ile çalışmadaki görünümü Çizelge 4.1 ve 4.2'te gösterilmektedir.

Çizelge 4.1 : İris veri setinin genel görünümü.

petal_length	petal_width	species
1,4	0,2	setosa
1,4	0,2	setosa
...
4,7	1,4	versicolor
4,5	1,5	versicolor
...
6,0	2,5	virginica
5,1	1,9	virginica
...

İris veri seti okunduktan sonra türlerinin işlem görebilmesi için sayılarla (setosa=1, versicolor=2, virginica=3) temsil edilmesi gerekmektedir.

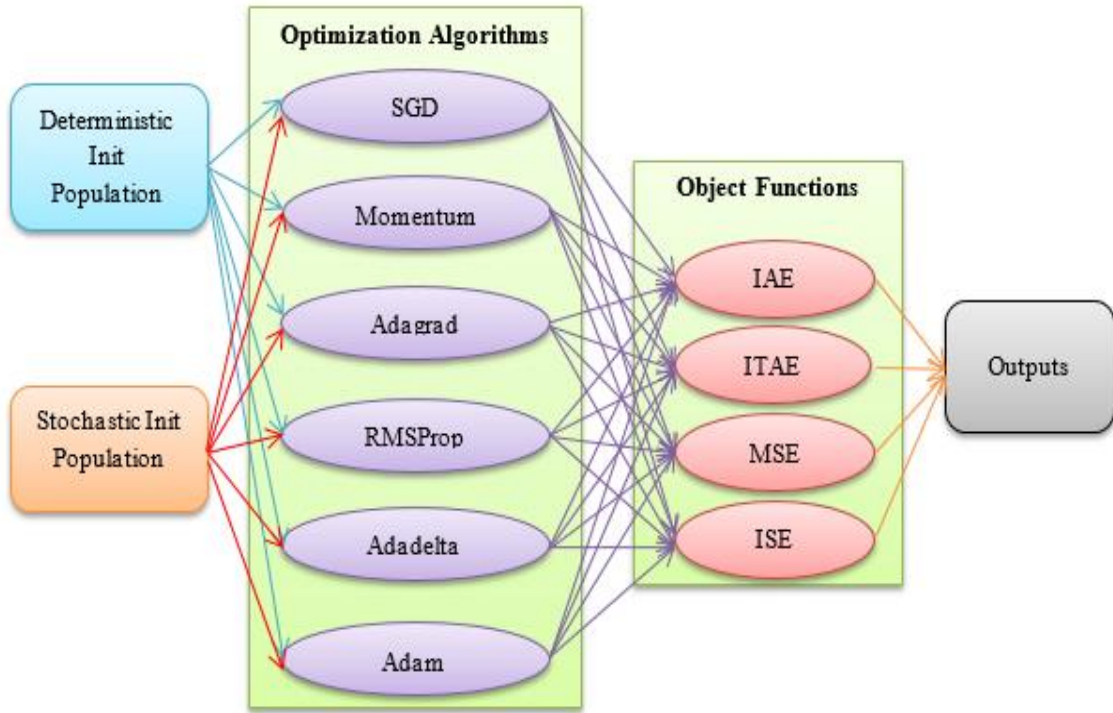
Çizelge 4.2 : İris veri setinin çalışma görünümü.

petal_length	petal_width	species
1,4	0,2	1
1,4	0,2	1
...
4,7	1,4	2
4,5	1,5	2
...
6,0	2,5	3
5,1	1,9	3
...

4.2 Deneysel Çalışma

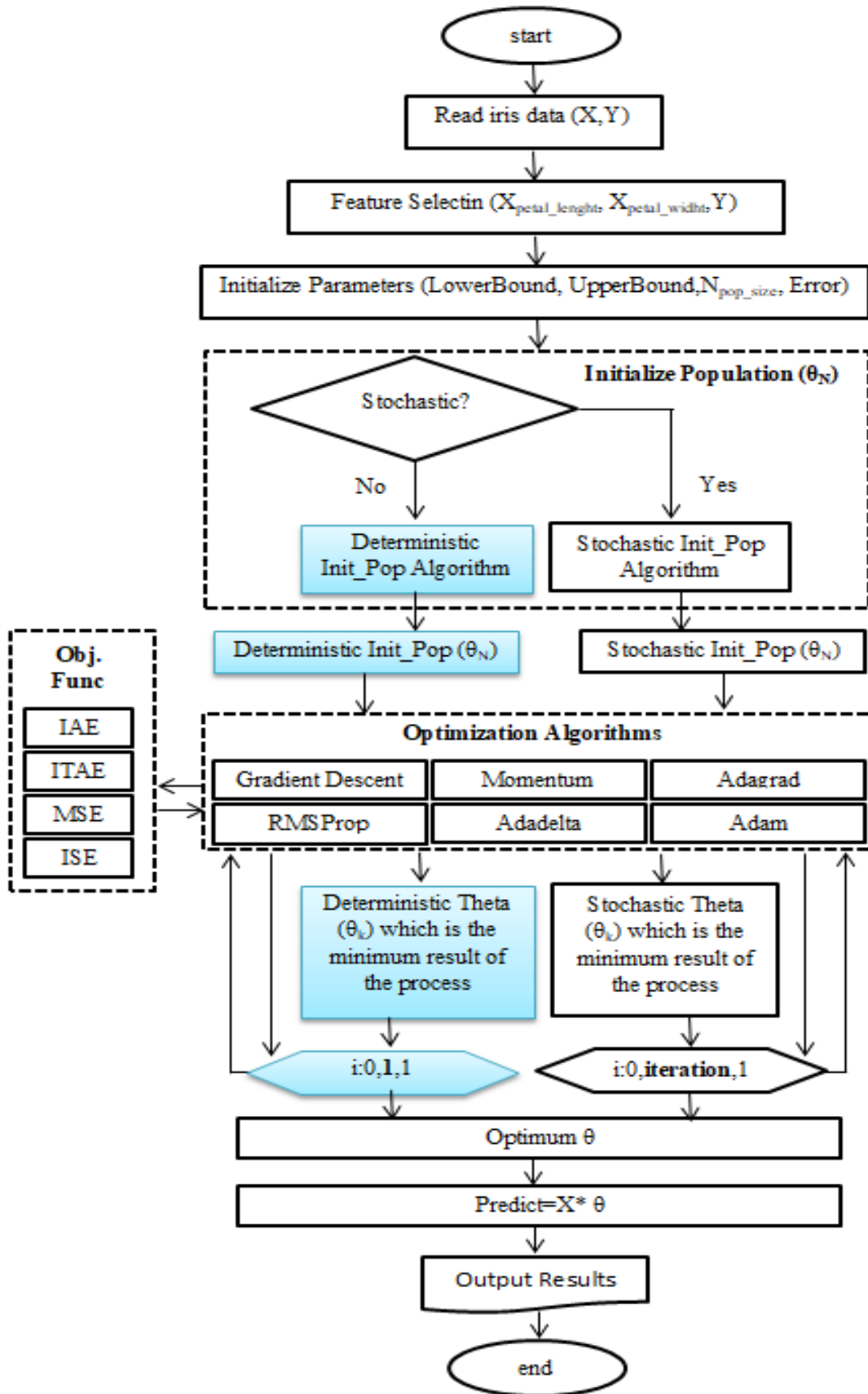
Yapılan uygulamanın tüm modülleri Şekil 4.3'te gösterilmektedir. 2 farklı başlangıç popülasyonu, 6 farklı optimizasyon algoritması ve 4 farklı amaç fonksiyonu ile sonuçlar üretilmektedir. Yani 2*6*4 tane sonuç karşılaştırılmaktadır ve sonuçlar analiz edilmektedir.

Çok değişkenli regresyon problemleri çözümünde, birçok giriş ve buna karşılık gelen çıkış değerleri kullanılarak bir fonksiyonun bulunması istenmektedir. Bu sebeple, regresyon işlemi sonucunda bulunan en uygun değerler, (4.1)'deki theta katsayılarının yerine konularak tüm giriş ve çıkış değerlerini temsil edecek fonksiyon bulunmaktadır. Bu katsayılar yerine uygun değerler bulmak için bir başlangıç değeri atanarak (genelde 0 ya da 1 verilir) süreç başlatılmaktadır. Ve bu başlangıç değeri ile dik iniş optimizasyon algoritmaları kullanılarak adım adım optimum theta katsayısı elde edilmektedir. Yani 1 tane theta değeri ile başlanmaktadır.



Şekil 4.3 : Çalışmanın tüm modülleri.

Bu çalışmada ise, stokastik ve deterministik birçok başlangıç katsayısı ile başlanmaktadır. Çünkü çözüme en kısa sürede gitmek istenmektedir. Bu başlangıç popülasyonundaki çözüm adaylarından sonuca en yakın bir aday çözüm ile başlamamız gerekmektedir. Sözde kodunda pre ile başlayan fonksiyonların (pre_SGD, pre_Momentum, pre_Adagrad, pre_RMSProp, pre_Adadelta, pre_Adam) amacı, başlangıç çözüm adayları arasındaki en uygun olan çözümü bulmaktır. Bütün çözüm adayları, tek tek optimizasyon ve amaç fonksiyonlarından birer kez işlendikten sonra en düşük sonucu veren çözüm adayı ile yola devam edilmektedir. Elde edilen ideal çözüm adayı ile regresyon adımları, hata toleransı 0.001 olana kadar devam ettirilmektedir. Regresyon işleminin sonucunda, tüm veri setini temsil edecek fonksiyonun katsayıları bulunmuş olacaktır. Şekil 4.4'te uygulamanın akış diyagramı ve algoritma 4.3'te sözde kodu verilmektedir.



Şekil 4.4 : Uygulamanın akış diyagramı.

Algoritma 4.3 : İris veri seti kullanarak MLR uygulamasının sözde kodu.

Input: N: Population Size, LB: Lower Bound, UB: Upper Bound

ϵ : error, X_{petal_width} , X_{petal_length} , Y

Output: iteration number, θ_1 , θ_2 , *Time*

1. For i in range (number of object functions = 4 (**IAE, ITAE, MSE, ISE**)):
 2. For j in range (number of work):

// Deterministik başlangıç için çalışma sayısı 1 ve stokastik başlangıç için 10'dur.
 3. Generate stochastic and deterministic a initial population
of N – dimensional solutions between LB and UB;
start_time;
 4. For k in range (N):
 5. **pre_SGD(i, k)** //Minimum theta'nın (θ_k) bulunması
 6. While $\epsilon < 0.001$
 7. *Optimum_* $\theta_{1,2}$, *iteration_number*, *time* = **SGD**(i, min_k)
 8. For k in range (N):
 9. **pre_Momentum(i, k)**
 10. While $\epsilon < 0.001$
 11. *Optimum_* $\theta_{1,2}$, *iteration_number*, *time* = **Momentum**(i, min_k)
 12. For k in range (N):
 13. **pre_Adagrad(i, k)**
 14. While $\epsilon < 0.001$
 15. *Optimum_* $\theta_{1,2}$, *iteration_number*, *time* = **Adagrad**(i, min_k)
 16. For k in range (N):
-

```

17.      pre_RMSPprop(i, k)
18.      While  $\epsilon < 0.001$ 
19.          Optimum_θ1,2, iteration_number, time = RMSPprop(i, min_k)
20.      For k in range (N):
21.          pre_Adadelta(i, k)
22.          While  $\epsilon < 0.001$ 
23.              Optimum_θ1,2, iteration_number, time = Adadelta(i, min_k)
24.          For k in range (N):
25.              pre_Adam(i, k)
26.              While  $\epsilon < 0.001$ 
27.                  Optimum_θ1,2, iteration_number, time = Adam(i, min_k)
28.          End For
29.  End For

```

4.3 Kullanılan Amaç Fonksiyonları

Makine öğrenme alanında Kayıp fonksiyon (Loss function), Hata fonksiyonu (Error function), Maliyet fonksiyonu (Cost function) ve Amaç fonksiyonu (Object function) kavramlarıyla sıkça karşılaşılmaktadır. Yapılan çalışmalarda genellikle aynı oldukları ve birbiri yerine kavramsal olarak kullanıldıkları görülmektedir. Fakat aralarında küçük de olsa fark bulunmaktadır (Allafi ve diğ., 2017; Ansari ve diğ., 2014; Kishnani ve diğ., 2014; Krishnan ve Karpagam, 2014).

Kayıp fonksiyon ve Hata fonksiyonu aynı kavramlardır. Genellikle tek veri örneği üzerinde tanımlanan fonksiyonlardır, hatanın karesini (square error) almak örnek olarak verilebilir (Allafi ve diğ., 2017; Ansari ve diğ., 2014; Kishnani ve diğ., 2014; Krishnan ve Karpagam, 2014).

Maliyet fonksiyonu, tüm veri seti üzerindeki toplam hata olarak tanımlanmaktadır. Hatanın karelerinin ortalaması (MSE) örnek olarak verilebilir (Allafi ve diğ., 2017; Ansari ve diğ., 2014; Kishnani ve diğ., 2014; Krishnan ve Karpagam, 2014).

Amaç fonksiyonu, eğitim esnasında optimize edilecek herhangi bir fonksiyon için kullanılan genel bir terimdir. Diğer bir deyişle maksimize veya minimize etmek istediğimiz fonksiyon, Amaç fonksiyonu olarak adlandırılmaktadır. Eğer fonksiyonu minimize etmek istiyorsak, bu fonksiyon aynı zamanda Amaç fonksiyonu olarak adlandırılmaktadır (Allafi ve diğ., 2017; Ansari ve diğ., 2014; Kishnani ve diğ., 2014; Krishnan ve Karpagam, 2014).

Özetlemek gerekirse, Hata fonksiyonu ve Kayıp fonksiyon Maliyet fonksiyonunun bir parçası, Maliyet fonksiyonu ise Amaç fonksiyonunun bir türü olarak tanımlayabiliriz.

Çalışmada kullanılan Amaç fonksiyonları;

$$IAE = \int_0^t |e(t)| dt \quad (4.2)$$

$$ITAE = \int_0^t t |e(t)| dt \quad (4.3)$$

$$MSE = \frac{1}{n} \int_0^t (e(t))^2 dt \quad (4.4)$$

$$ISE = \int_0^t (e(t))^2 dt \quad (4.5)$$

5. UYGULAMA SONUÇLARI

Başlangıç popülasyon sayıları farklı başlatılarak (N=10, N=50 ve N=100) çıkan sonuçlar Çizelge 5.1, 5.2 ve 5.3'te listelenmektedir.

Çizelge 5.1'de, başlangıç popülasyonu olarak 10 tane çözüm adayı ile başlatılan uygulamanın sonuçları karşılaştırılmıştır. Burada farklı başlangıç popülasyon tipleri, optimizasyon algoritmaları ve amaç fonksiyonları kullanarak yapılan uygulamaların sonuçları (θ_1 , θ_2 , iterasyon sayıları) listelenmiştir. Ayrıca tablonun en alt satırında her bir amaç fonksiyonunun tüm optimizasyon yöntemleri için geçen işlem zamanları verilmiştir. Deterministik başlatma yöntemi ile başlatılan uygulamada bütün amaç fonksiyonlarının işlem zamanları stokastik başlatma yöntemine göre çok düşük olduğu görülmektedir. Optimizasyon yöntemlerinden Adadelta, bütün amaç fonksiyonlarında en düşük iterasyon sayısı ile sonuçlara ulaşmaktadır. Adagrad yönteminde de ISE hariç diğer amaç fonksiyonlarında Adadelta ile birbirine yakın iterasyon sayıları ile sonuçlanmaktadır. Her iki başlatma yönteminde de adam optimizasyon algoritması en yüksek iterasyon sayılarına sahiptir. Amaç fonksiyonlarından MSE, bütün optimizasyon algoritmalarında en düşük iterasyon sayıları ile sonuçlara ulaşmaktadır.

Çizelge 5.2'de, 50 tane çözüm adayı içeren başlangıç popülasyonu ile başlatılan uygulamanın sonuçları listelenmiştir. Çizelge 5.1'de olduğu gibi deterministik yöntemle başlatılan uygulamaların işlem zamanları oldukça düşüktür. Adadelta en başarılı optimizasyon yöntemi ve MSE en başarılı amaç fonksiyonu olarak gözlemlenmiştir. Burada Adagrad'ın başarısı biraz düşmüştür.

Çizelge 5.3'te ise, 100 tane çözüm adayı ile başlatılan uygulamanın sonuçları listelenmiştir. Başarı oranları Çizelge 5.1 ve 5.2'ye yakın olduğu gözlemlenmiştir. Burada dikkat edilmesi gereken nokta, başlangıç popülasyon sayısı 100 olduğunda deterministik yöntemde işlem zamanlarının düşmesidir. Çünkü popülasyon sayısı yükseldikçe deterministik başlatma yönteminde, optimum çözüme daha yakın bir çözüm adayı ile uygulama başlatılmaktadır. Bu sebeple düşük işlem zamanları ile sonuçlara ulaşılmaktadır.

Çizelge 5.1 : N(başlangıç popülasyonu sayısı)=10 için sonuç tablosu.

Baş. Pop. Türleri Amaç Fonk. Opt. Alg. Çıktılar	Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)			
	IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE
01	0.4622060213	0.4624823941	0.4297202763	0.4614782471	0.4623812815	0.4623555949	0.4300724303	0.4614894980
02	0.1517512404	0.1517995926	0.1418751074	0.1516027184	0.1512339118	0.1521087809	0.1413715299	0.1515807489
iter_sayisi	276	283	131	261	281.5	287.2	132.2	265.4
01	0.4626477845	0.4628473200	0.4343439895	0.4618207401	0.4627478130	0.4626914288	0.4343783391	0.4617425651
02	0.1519929937	0.1520278621	0.1433727040	0.1518192524	0.1517067998	0.1525380662	0.1428163609	0.1519823895
iter_sayisi	236	241	120	220	241.5	246.2	120.7	223.9
01	0.0000121701	0.0000080874	0.0000080874	0.0002612904	0.0000141256	0.0000060233	0.0000067727	0.0002363687
02	0.0000122019	0.0000080961	0.0000080961	0.0002616019	0.0000131979	0.0000061062	0.0000053495	0.0002351421
iter_sayisi	5	2	2	543	9.8	4	2	515.8
01	0.4411992824	0.3869276024	0.3641362442	0.4223663903	0.4441376097	0.3870541841	0.3644675776	0.4280265633
02	0.2197256042	0.3773045799	0.3630577326	0.2740601046	0.2099684981	0.3769592366	0.36244448584	0.2575801922
iter_sayisi	604	434	390	546	702.9	433.8	388.6	568.2
01	0.0000001566	0.0000001566	0.0000001545	0.0000001566	0.0000001566	0.0000001566	0.0000001545	0.0000001566
02	0.0000001550	0.0000001550	0.0000001530	0.0000001550	0.0000001550	0.0000001550	0.0000001530	0.0000001550
iter_sayisi	3	3	2	3	3	3	2	3
01	0.4076068514	0.4071502772	0.3207650044	0.4072834342	0.4086150154	0.4078421021	0.3213391119	0.4075228035
02	0.2989511559	0.2987267915	0.2381594282	0.2987930603	0.2961195079	0.2970837585	0.2375826875	0.2976135932
iter_sayisi	1642	1621	637	1627	1745.1	1573.4	638.3	1761.8
zaman (sn)	0.22	0.21	0.1	0.25	3.33	3.26	1.34	4.03

Çizelge 5.2 : N(başlangıç popülasyonu sayısı)=50 için sonu tablosu.

Baş. Pop. Türleri Amaç Fonk. Opt. Alg. Çıktılar	Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)				
	IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE	
SGD	01	0.4623166167	0.4625967937	0.4300272708	0.4616345641	0.4621668085	0.4625397047	0.4300262615	0.4614728060
	02	0.1513230763	0.1513728850	0.1415018406	0.1511831362	0.1518450138	0.1515786806	0.1416158861	0.1516403951
	iter_sayisi	281	288	137	267	280.6	286.7	135.9	265.9
Momentum	01	0.4627221014	0.4629206297	0.4345649478	0.4618992588	0.4625553824	0.4628854577	0.4342853902	0.4617180067
	02	0.1518003868	0.1518349877	0.1432269814	0.1516278245	0.1522785550	0.1519750904	0.1431993942	0.1520390988
	iter_sayisi	242	247	126	226	240.9	245.7	124.5	224.4
Adagrad	01	0.0000442249	0.0000201104	-0.0000019629	0.0002933523	0.0000519890	0.0000270480	0.0000040082	0.0003040612
	02	0.0000434693	0.0000194965	-0.0000021752	0.0002923950	0.0000502632	0.0000274385	0.0000039891	0.0003039396
	iter_sayisi	25	10	2	563	24.7	9.3	2	563.4
RMSProp	01	0.4410738064	0.3871111222	0.3645063474	0.4221337271	0.4421857419	0.3869357842	0.3640832756	0.4311733018
	02	0.2200617486	0.3769018581	0.3630530122	0.2744203793	0.2175817818	0.3773450552	0.3628160551	0.2484023393
	iter_sayisi	599	430	386	541	680.5	425.9	384.1	564.3
Adadelta	01	0.0000001566	0.0000001566	0.0000001545	0.0000001566	0.0000001566	0.0000001566	0.0000001545	0.0000001566
	02	0.0000001550	0.0000001550	0.0000001530	0.0000001550	0.0000001550	0.0000001550	0.0000001530	0.0000001550
	iter_sayisi	3	3	2	3	3	3	2	3
Adam	01	0.4078948677	0.4078053039	0.3258893421	0.4077146710	0.4081278685	0.4074849926	0.2928250948	0.4073674981
	02	0.2982712544	0.2982279706	0.2410281619	0.2981838344	0.2977464763	0.2991245534	0.2172709282	0.2994012470
	iter_sayisi	1483	1479	573	1475	1464.3	1455.3	517.6	1442.6
	zaman (sn)	0.22	0.24	0.11	0.26	3.46	4.14	1.49	4.29

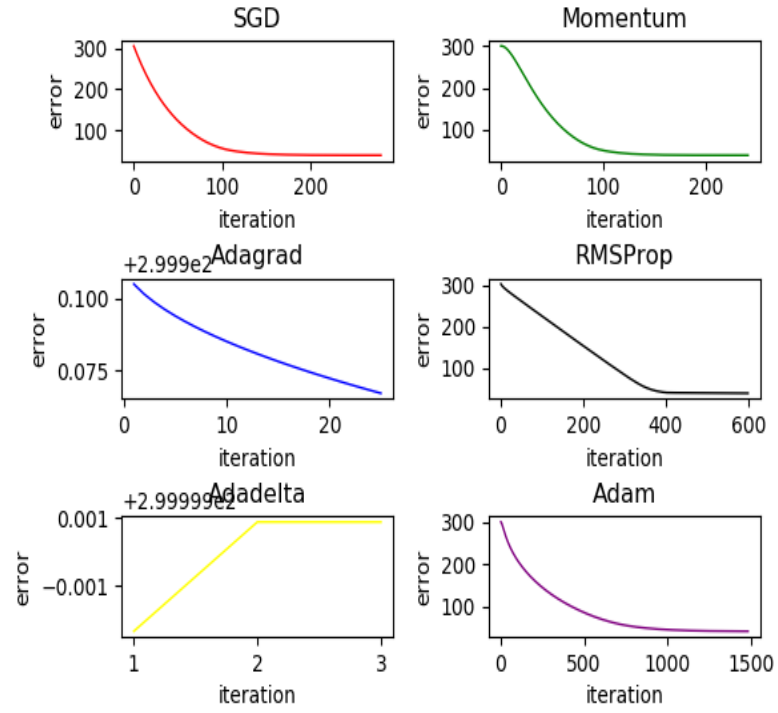
Çizelge 5.3 : N(başlangıç popülasyonu sayısı)=100 için sonuç tablosu.

Baş. Pop. Türleri Amaç Fonk. Opt. Alg. Çıktılar	Deterministik Başlangıç Popülasyonu				Stokastik Başlangıç Popülasyonu (10 çalışmanın ortalaması)			
	IAE	ITAE	MSE	ISE	IAE	ITAE	MSE	ISE
SGD								
01	0.4623264798	0.4626035964	0.4297443391	0.4615967019	0.4622245568	0.4625921858	0.4297919236	0.4616122328
02	0.1513736828	0.1514224563	0.1414619352	0.1512241217	0.1516526487	0.1514304828	0.1416483863	0.1512506115
iter_sayisi	281	288	136	266	281	286.6	135.4	266
Momentum								
01	0.4626977714	0.4629355465	0.4342532480	0.4618667616	0.4626056528	0.4629367345	0.3906698534	0.4618531872
02	0.1518181746	0.1518595360	0.1431500970	0.1516429414	0.1521084242	0.1518185307	0.1290040596	0.1516449267
iter_sayisi	241	247	125	225	241.3	245.5	111.9	224.4
Adagrad								
01	0.0000498493	0.0000258422	-0.0000004655	0.0002988242	0.0000473977	0.0000294838	0.0000065216	0.0003034246
02	0.0000493761	0.0000254232	-0.0000006573	0.0002982734	0.0000476333	0.0000296569	0.0000063738	0.0003013805
iter_sayisi	26	11	2	564	25.2	9.8	2	563.2
RMSProp								
01	0.4181153588	0.3870273112	0.3643242087	0.4219223773	0.4449118273	0.3869742931	0.3641696192	0.4245719501
02	0.2947513428	0.37707703	0.3630165739	0.2755960673	0.2091192967	0.3772550821	0.3628661613	0.2674355604
iter_sayisi	517	428	384	538	738.8	425.2	381.3	543.7
Adadelat								
01	0.0000001566	0.0000001566	0.0000001545	0.0000001566	0.0000001566	0.0000001566	0.0000001545	0.0000001566
02	0.0000001550	0.0000001550	0.0000001530	0.0000001550	0.0000001550	0.0000001550	0.0000001530	0.0000001550
iter_sayisi	3	3	2	3	3	3	2	3
Adam								
01	0.4077982136	0.4077982136	0.3267051778	0.4076618436	0.4075620668	0.4076196406	0.3267335606	0.4077484053
02	0.2985284930	0.2985284930	0.2419271803	0.2984624020	0.2990752248	0.2988847012	0.2421188853	0.2982401195
iter_sayisi	1454	1454	561	1448	1469.9	1444.3	558.7	1446.9
zaman (sn)	0.19	0.2	0.1	0.24	3.32	3.44	1.46	3.56

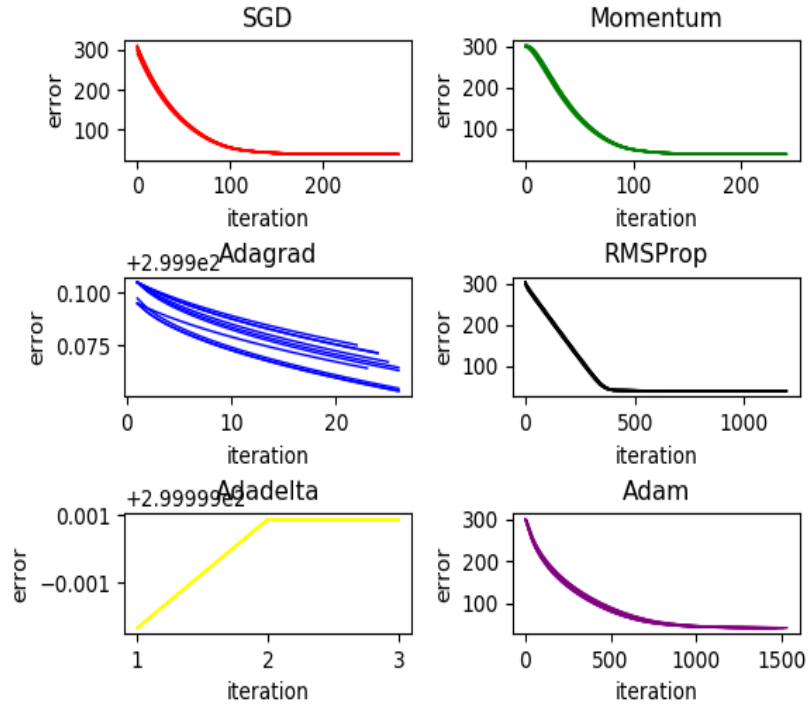
Sonuçlar incelendiğinde, deterministik başlatma yönteminin stokastik başlatma yönteminden zaman açısından oldukça avantajlı olduğu görülmektedir. Çünkü deterministik başlatma yönteminde uygulamayı bir kez çalıştırmak yeterli olmaktadır ve sadece bir kez çalışan uygulama ile başarılı sonuçlar elde edilebilir. Stokastik başlatma yönteminde ise başarılı sonuçların elde edilebilmesi için uygulamanın en az 10 kez çalıştırılarak ortalama sonuçlarının elde edilmesi gerekmektedir. Uygulamanın çalıştırılma sayısı 10'dan fazla da olabilir. Her iki yöntemde sonuçlar neredeyse aynı olmasına rağmen toplam geçen zaman olarak büyük fark vardır. Deterministik olarak başlatılan popülasyonda zamansal olarak çok yüksek bir kazanım elde edilmektedir.

Örneğin deterministik yöntemle başlatılan popülasyonda (N=50 için), IAE amaç fonksiyonu için tüm optimizasyon algoritmalarına uygulandığında geçen zaman 0.22 dk olmaktadır. Fakat aynı amaç fonksiyonu ve optimizasyon algoritmaları ile stokastik başlangıç popülasyonunda, 10 çalışma için geçen zaman miktarı 3.46 dk'dır.

Şekil 5.1'de deterministik yöntemle başlatılan (N=50 için) ve IAE amaç fonksiyonunu kullanan uygulamanın bütün optimizasyon algoritmalarındaki hata ve iterasyon sayıları grafiksel düzlemde gösterilmektedir. Şekil 5.2'de ise uygulama stokastik olarak başlatılmış ve IAE amaç fonksiyonu kullanılmaktadır. Burada, tüm optimizasyon algoritmaları için uygulamalar 10 defa çalıştırılarak sonuçlar grafiksel olarak sunulmaktadır. Yani Şekil 5.2'de uygulama 10 defa çalıştırıldığında oluşan sonuçlar karşılaştırmalı olarak çizilmektedir.

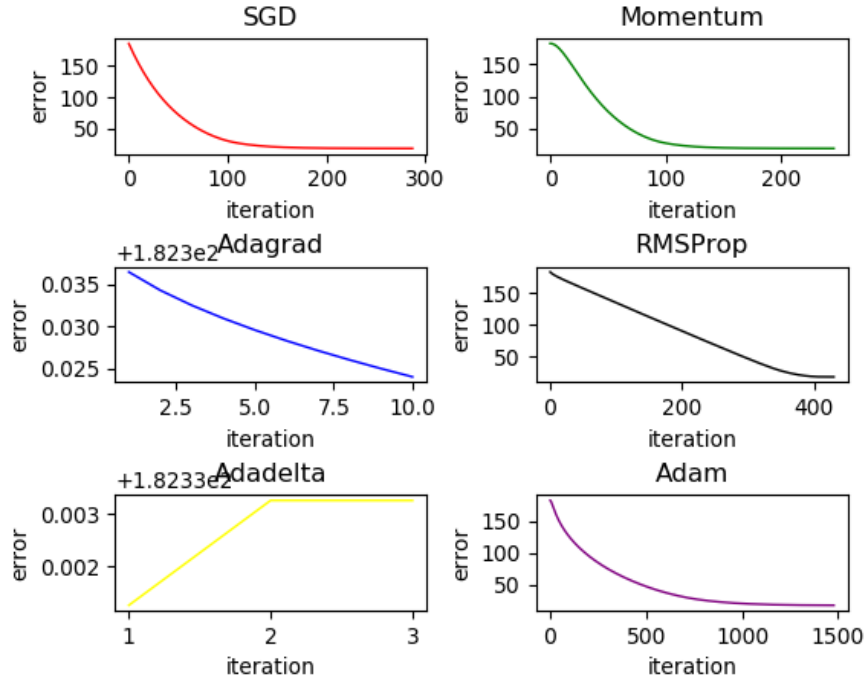


Şekil 5.1 : Dik iniş yöntemlerinin IAE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıları.

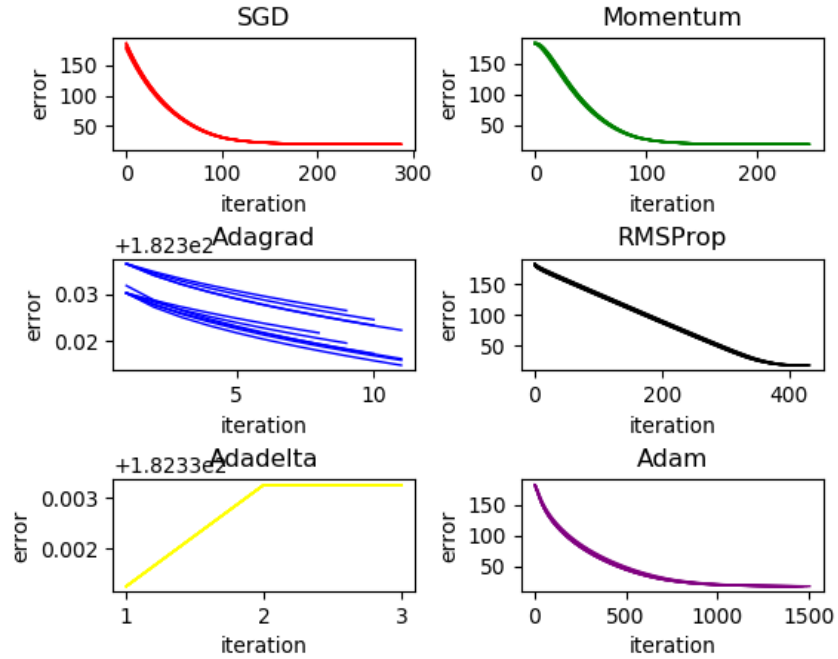


Şekil 5.2 : Dik iniş yöntemlerinin IAE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıları.

Şekil 5.2 ve 5.3'te, her iki başlatma yöntemi, ITAE amaç fonksiyonu ve farklı optimizasyon yöntemlerini kullanarak yapılan uygulamaların sonuçları grafiksel olarak gösterilmektedir.

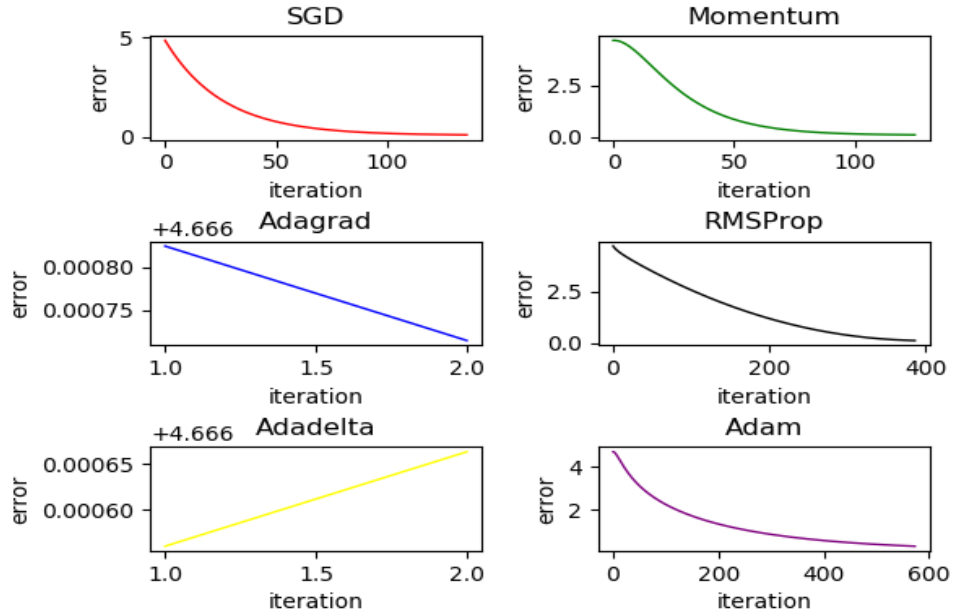


Şekil 5.3 : Dik iniş yöntemlerinin ITAE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıkları.

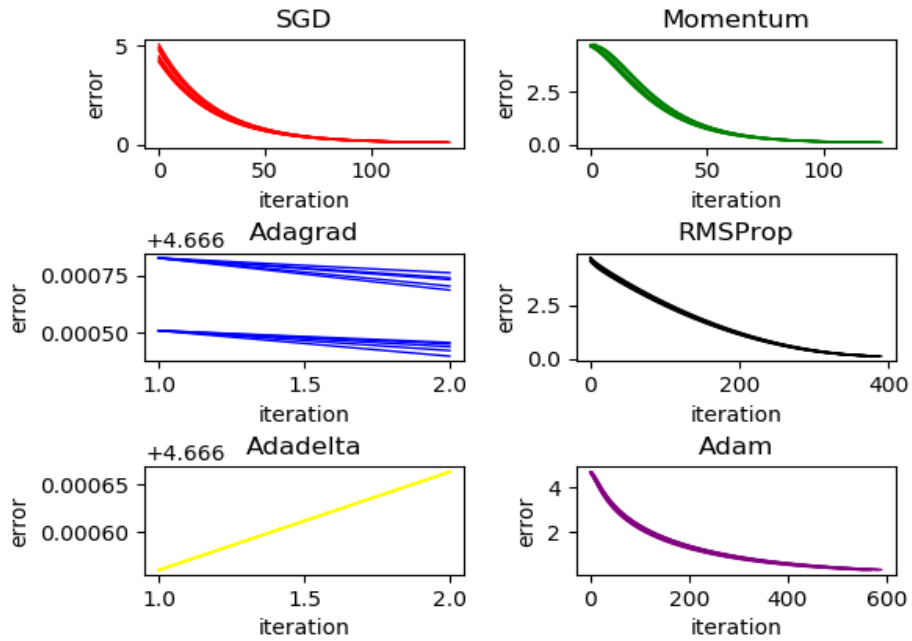


Şekil 5.4 : Dik iniş yöntemlerinin ITAE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıkları.

Şekil 5.5 ve 5.6'da, her iki başlatma yöntemi MSE amaç fonksiyonu ve farklı optimizasyon yöntemlerini kullanarak yapılan uygulamaların sonuçları grafiksel olarak gösterilmektedir. Her iki başlatma yöntemindeki tüm optimizasyon algoritmalarının sonuçlarına bakıldığında, en düşük iterasyon sayılarına MSE amaç fonksiyonu ile ulaşılmaktadır.

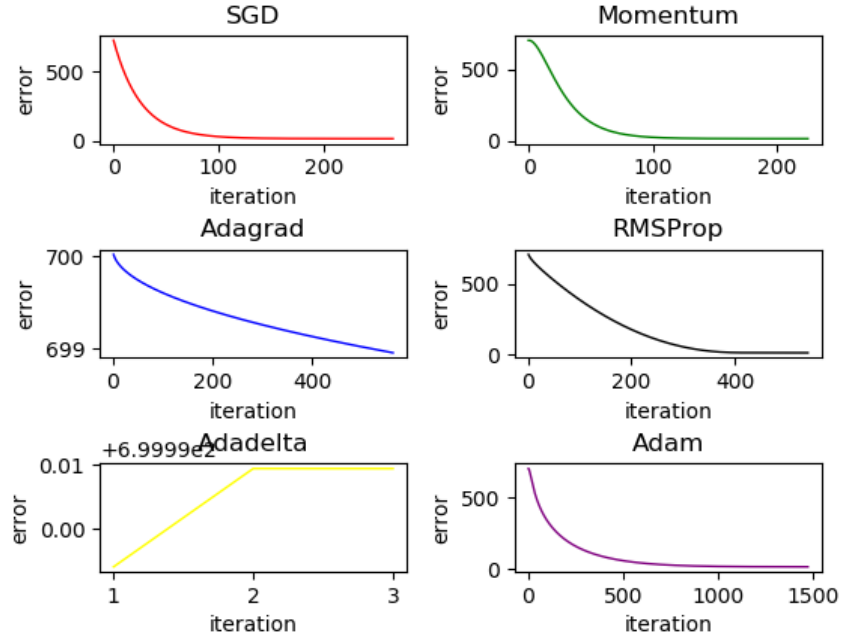


Şekil 5.5 : Dik iniş yöntemlerinin MSE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıları.

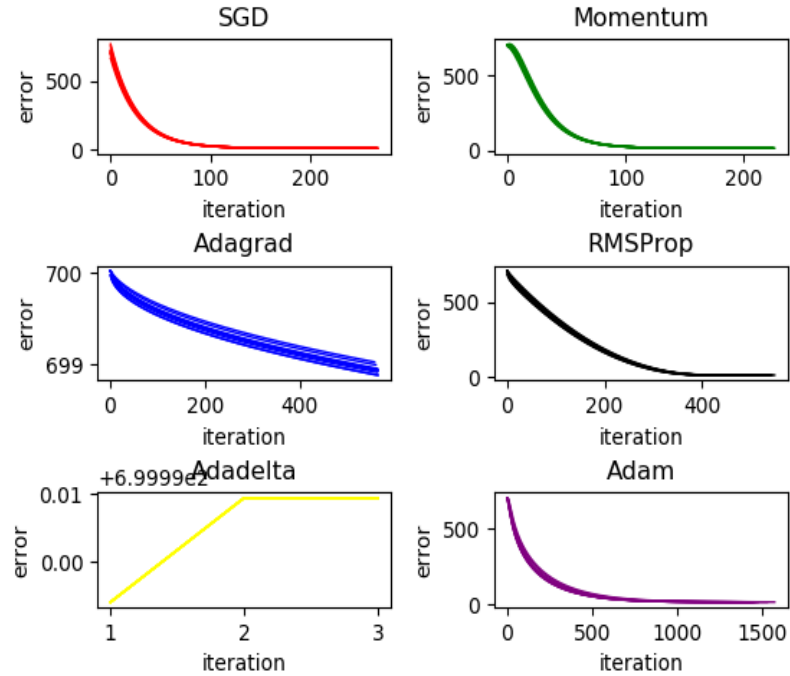


Şekil 5.6 : Dik iniş yöntemlerinin MSE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıları.

Şekil 5.7 ve 5.8’de, her iki başlatma yöntemi ISE amaç fonksiyonu ve farklı optimizasyon yöntemlerini kullanarak yapılan uygulamaların sonuçları grafiksel olarak sunulmaktadır.



Şekil 5.7 : Dik iniş yöntemlerinin ISE amaç fonksiyonunda deterministik başlangıç popülasyonu çıktıları.



Şekil 5.8 : Dik iniş yöntemlerinin ISE amaç fonksiyonunda stokastik başlangıç popülasyonu çıktıları.

Deterministik ve stokastik başlatma yöntemlerinin farklı amaç fonksiyonlarına göre tüm optimizasyon algoritmalarının sonuçları Şekil 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 ve 5.8'de gösterilmektedir. Deterministik başlatma yöntemi 1 kez, stokastik başlatma yöntemi ise 10 kez çalıştırılmaktadır ve aynı grafiksel düzlemde gösterilmektedir. Şekillerde de görüldüğü gibi, birbirine çok yakın sonuçlar elde edilmektedir.



6. SONUÇLAR

Np-zor olan problemlerin çözümünü matematiksel işlemlerle garanti etmek neredeyse imkânsızdır. Bunun yerine yaklaşık metotlar kullanılmaktadır. Yaklaşık metotların sonuçları garanti etme özelliği yoktur ama makul sürede en iyi sonuca yaklaşmaktadırlar. Optimizasyon algoritmaları kullanılarak bu sonuçlar elde edilmektedir. Günümüzde birçok optimizasyon algoritması mevcuttur. Bazıları genel problemlere, bazıları ise özel problemlere uygulanmaktadır. Bu algoritmaların başlangıç popülasyonu oluşturma adımı, çözümü bulmada oldukça etkili olmaktadır. Nüfus çeşitliliği artırılarak daha hızlı yakınsama gözlemlenebilir.

Bu makalede, başlangıç popülasyonu oluşturma teknikleri detaylı bir şekilde incelenmiştir ve bu alanda yeni bir kategorizasyon önerilmiştir. Yeni kategorizasyonda incelenen tüm yöntemler üç ana başlık altında kategorize edilmiştir. Bu ana başlıklar, Stokastik, Deterministik ve Sezgisel şeklinde adlandırılmıştır.

Stokastik metotlar PRNG, QRNG ve TRNG olmak üzere üç bölüm şeklinde incelenmiştir. PRNG'ler, rastgele sayılar için oldukça kullanışlı yöntemlerdir. QRNG'ler ve TRNG'ler ise, nüfus çeşitliliğini sağlamak için geliştirilen yöntemlerdir. Özellikle TRNG'ler, harici donanımlar kullanılarak gerçeğe en yakın rastgele sayılar üretmek için kullanılan yöntemlerdir.

Deterministik metotlar düşük tutarsızlık dizileri ve kaos yöntemleri olarak iki bölüm şeklinde incelenmiştir. Bu iki bölüm altında incelenen yöntemlerin amacı, çözüm adayları seçilirken daha geniş katılımı sağlamaktır. Yani seçilen çözüm adaylarının tüm bölgeleri kapsamalarına olanak sağlamaktadırlar.

Sezgisel metotlarda birçok yeni veya değiştirilmiş ya da hibridleme sonucunda oluşturulan yöntem incelenmiştir. Genelde var olan yöntemlerin geliştirilmesiyle meydana gelmektedirler.

Günümüzde sezgisel yöntemlerle üretilen sayılar kullanılarak oluşturulan başlangıç yöntemlerinin kullanımı artmaktadır. Bu sayılar tam gerçek rastgele sayılar değildir. Probleme özgü değiştirilebilirliği olduğu için tercih edilmektedir. Buna karşılık, gerçek rastgele sayılar ile başlangıç popülasyonu oluşturmak için TRNG'ler kullanılmaktadır. TRNG'lerin pahalı sistemler olması sebebiyle QRNG'ler ve PRNG'ler tercih edilmektedir.

Bu alanda arařtırmacıların halen en çok kullandığı yöntem olan PRNG'lerin nüfus çeřitliliđi için zayıf kaldığını öngörmekteyiz. Bunun için tüm çözüm bölgelerini kapsayacak yeni ve az maliyetli yöntemlere ihtiyaç olduğunu düşünüyöruz. Son yıllarda arařtırmacıların sezgisel modellere yoğunlaşmasını ümit verici olarak görmekteyiz.

Makine öğrenme yöntemlerinin tümünde olduğu gibi çok deđişkenli regresyon modelinde de başlangıç popülasyonu konusu oldukça önemlidir. Bu konuya yeni bir deterministik yaklaşımla katkı sunulmuştur. Önerilen yöntemin sonuçları ile geleneksel stokastik başlatma yönteminin sonuçları karşılaştırılmıştır. Önerilen deterministik yöntemle oluşturulan başlangıç popülasyonlu uygulamayı 1 kez çalıştırmak yeterli olmaktadır. Çünkü sabit deđerler ile başladığı için her çalışmanın sonuçları eşittir. Fakat stokastik başlangıç popülasyonlu uygulama 10 kez çalıştırılmış ortalama deđerleri alınmıştır. Stokastik başlatma yöntemlerinde sonuçların literatüre sunulabilmesi için birden fazla çalıştırılması gerekmektedir. Deterministik yöntemin sonuçları ile stokastik yöntemin ortalama sonuçları eşit olduğu gözlemlenmiştir. Ancak çalışma süreleri bakımından deterministik yöntemin yaklaşık %90 civarında bir iyileştirme yaptığı görölmektedir. Optimizasyon yöntemlerinden Adadelta, amaç fonksiyonlarından da MSE'nin yüksek başarı oranlarına sahip olduğu sonucuna varılmaktadır.

KAYNAKLAR

- Akanksha Rai.** (2019). Erişim: 17 Ocak 2020, *ML | Classification vs Regression*. <https://www.geeksforgeeks.org/ml-classification-vs-regression/>
- Alatas, B.** (2019). Sports inspired computational intelligence algorithms for global optimization. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-017-9587-x>
- Ali, M., Pant, M., Abraham, A.** (2013). Unconventional initialization methods for differential evolution. *Applied Mathematics and Computation*. <https://doi.org/10.1016/j.amc.2012.10.053>
- Allafi, N. I., Zayed, A. S., Daw, M. S., Ahmed, W. M.** (2017). Investigating the effect of GA based PID controller performance indices with application to stirred-Tank heater (jacket model). *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering, STA 2016 - Proceedings*. <https://doi.org/10.1109/STA.2016.7952015>
- Alpaydin, E.** (2014). Introduction to Machine Learning Ethem Alpaydin. *Introduction to Machine Learning, Third Edition*.
- Aluru, S.** (1997). Lagged Fibonacci Random Number Generators for Distributed Memory Parallel Computers. *Journal of Parallel and Distributed Computing*. <https://doi.org/10.1006/jpdc.1997.1363>
- Amador-Angulo, L., Mendoza, O., Castro, J. R., Rodríguez-Díaz, A., Melin, P., Castillo, O.** (2016). Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors (Switzerland)*. <https://doi.org/10.3390/s16091458>
- Ansari, A. Q., Ibraheem, Katiyar, S.** (2014). Application of ant colony algorithm for calculation and analysis of performance indices for adaptive control system. *Proceedings of the International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity, CIPECH 2014*. <https://doi.org/10.1109/CIPECH.2014.7019078>
- Arroyo, J. C. T., Delima, A. J. P.** (2020). An improved affine cipher using blum blum shub algorithm. *International Journal of Advanced Trends in Computer Science and Engineering*. <https://doi.org/10.30534/ijatcse/2020/133932020>
- Avaroglu, E.** (2017). Pseudorandom number generator based on Arnold cat map and statistical analysis. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(1), 633–643. <https://doi.org/10.3906/elk-1507-253>
- Bhesdadiya, R. H., Trivedi, I. N., Jangir, P., Jangir, N., Kumar, A.** (2016). An NSGA-III algorithm for solving multi-objective economic/environmental dispatch problem. *Cogent Engineering*. <https://doi.org/10.1080/23311916.2016.1269383>
- Bird, J. J., Ekárt, A., Faria, D. R.** (2020). On the effects of pseudorandom and quantum-random number generators in soft computing. *Soft Computing*. <https://doi.org/10.1007/s00500-019-04450-0>
- Budiman, A., Buulolo, E., Saputra, I.** (2020). *Middle Square Method Analysis of Number Pseudorandom Process*. 4(2), 35–38. <https://doi.org/10.30865/ijics.v4i2.1386>

- Chen, H., Yang, C., Deng, K., Zhou, N., Wu, H.** (2017). Multi-objective optimization of the hybrid wind/solar/fuel cell distributed generation system using Hammersley Sequence Sampling. *International Journal of Hydrogen Energy*. <https://doi.org/10.1016/j.ijhydene.2017.01.202>
- Chong, E. K., Zak, S. H.** (2004). An introduction to optimization. John Wiley & Sons.
- Clawdia, J., Khairina, N., Harahap, M. K.** (2017). Implementasi Algoritma Kriptografi One Time Pad (Otp) Dengan Dynamic Key Linear Congruential Generator (Lcg). *Konferensi Nasional Teknologi Informasi dan Komputer (KOMIK), I*, 12–14.
- Collet, P., Rennard, J.-P.** (2011). Stochastic Optimization Algorithms. *Intelligent Information Technologies*, 1121–1137. <https://doi.org/10.4018/978-1-59904-941-0.ch064>
- Cruz-Chávez, M. A., Martínez-Oropeza, A.** (2016). Feasible Initial Population with Genetic Diversity for a Population-Based Algorithm Applied to the Vehicle Routing Problem with Time Windows. *Mathematical Problems in Engineering*, 2016. <https://doi.org/10.1155/2016/3851520>
- Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.** (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/3-540-45356-3_83
- Deb, K., Jain, H.** (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/TEVC.2013.2281535>
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.** (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/4235.996017>
- Deisenroth, M. P., Faisal, A. A., Ong, C. S.** (2020). Mathematics for Machine Learning. İçinde *Mathematics for Machine Learning*. <https://doi.org/10.1017/9781108679930>
- Delima, A. J. P., Arroyo, J. C. T.** (2020). An enhanced nihilist cipher using blum blum shub algorithm. *International Journal of Advanced Trends in Computer Science and Engineering*. <https://doi.org/10.30534/ijatcse/2020/122932020>
- Demir, M., Karıcı, A.** (2007). Veri kümelemede fidan gelişim algoritmasının kullanılması. *ELEKTRİK-ELEKTRONİK-BİLGİSAYAR MÜHENDİSLİĞİ 12. ULUSAL KONGRESİ VE FUARI BİLDİRİLERİ*, 2007. https://www.emo.org.tr/etkinlikler/ulusal/etkinlik_bildirileri_detay.php?etkinlikkod=27&bilkod=2976
- Demir, M., Karıcı, A., Özdemir, M.** (2011). Fidan gelişim algoritması yardımı ile DNA motiflerinin keşfi. *Çankaya University Journal of Science and Engineering*, 8(1).
- Deniz, A., Kiziloz, H. E.** (2019). On initial population generation in feature subset selection. *Expert Systems with Applications*, 137, 11–21. <https://doi.org/10.1016/j.eswa.2019.06.063>
- DiCarlo, D.** (2012). Random Number Generation: Types and Techniques. *Senior Honors Theses*. <https://digitalcommons.liberty.edu/honors/308>

- Digehsara, P. A., Chegini, S. N., Bagheri, A., Roknsaraei, M. P.** (2020). An improved particle swarm optimization based on the reinforcement of the population initialization phase by scrambled Halton sequence. *Cogent Engineering*. <https://doi.org/10.1080/23311916.2020.1737383>
- Dogo, E. M., Afolabi, O. J., Nwulu, N. I., Twala, B., Aigbavboa, C. O.** (2018). A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks. *Proceedings of the International Conference on Computational Techniques, Electronics and Mechanical Systems, CTEMS 2018*. <https://doi.org/10.1109/CTEMS.2018.8769211>
- Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., Cosar, A.** (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering*, 137, 106040.
- Dong, N., Wu, C. H., Ip, W. H., Chen, Z. Q., Chan, C. Y., Yung, K. L.** (2012). An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection. *Computers and Mathematics with Applications*, 64(6), 1886–1902. <https://doi.org/10.1016/j.camwa.2012.03.040>
- Dua, D. and Graff, C.** (2019) UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA. <http://archive.ics.uci.edu/ml>.
- Dutang, C., Wuertz, D.** (2009). A note on random number generation. *Modern Electric Power Systems*.
- Ergün, S., Özoğuz, S.** (2010). Truly random number generators based on non-autonomous continuous-time chaos. *International Journal of Circuit Theory and Applications*. <https://doi.org/10.1002/cta.520>
- Ergün, S., Özoğuz, S.** (2007). Truly random number generators based on a non-autonomous chaotic oscillator. *AEU - International Journal of Electronics and Communications*. <https://doi.org/10.1016/j.aeue.2006.05.006>
- Eröz, E., Tanyildizi, E.** (2018). "Güncel Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırılması," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1-16, doi: 10.1109/IDAP.2018.8620764.
- Farsana, F. J., Gopakumar, K.** (2016). A Novel Approach for Speech Encryption: Zaslavsky Map as Pseudo Random Number Generator. *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2016.07.302>
- Faure, H., Kritzer, P., Pillichshammer, F.** (2015). From van der Corput to modern constructions of sequences for quasi-Monte Carlo rules. *Indagationes Mathematicae*. <https://doi.org/10.1016/j.indag.2015.09.001>
- Fausto, F., Reyna-Orta, A., Cuevas, E., Andrade, Á. G., Perez-Cisneros, M.** (2020). From ants to whales: metaheuristics for all tastes. *Artificial Intelligence Review*. <https://doi.org/10.1007/s10462-018-09676-2>
- Gadhvi, B., Savsani, V., Patel, V.** (2016). Multi-Objective Optimization of Vehicle Passive Suspension System Using NSGA-II, SPEA2 and PESA-II. *Procedia Technology*. <https://doi.org/10.1016/j.protcy.2016.03.038>
- Gao, Z. M., Zhao, J., Hu, Y. R., Chen, H. F.** (2019). The improved harris hawk optimization algorithm with the tent map. *2019 IEEE 3rd International Conference on*

Electronic Information Technology and Computer Engineering, EITCE 2019.
<https://doi.org/10.1109/EITCE47263.2019.9095091>

- Garbade, D. M. J.** (2018). *Regression Versus Classification Machine Learning: What's the Difference?* <https://medium.com/quick-code/regression-versus-classification-machine-learning-whats-the-difference-345c56dd15f7>
- Girip, I., Munteanu, L.** (2015). Modeling of non-stationary vibration signals based on the modified Kronecker sequences. *PAMM*. <https://doi.org/10.1002/pamm.201510088>
- Gündoğan, K. K., Alataş, B., Karci, A.** (2004). Mining classification rules by using genetic algorithms with non-random initial population and uniform operator. *Turkish Journal of Electrical Engineering and Computer Sciences*.
- Gürsu, B., İnce, M. C., Kent, S., Sevinç, A.** (2018). *Az Uyumsuz Dizilerle Geliştirilmiş Genetik Algoritmalar*. July 2012.
- Gustafsson, T., Eriksson, A., Wingren, C. J.** (2017). Multivariate linear regression modelling of lung weight in 24,056 Swedish medico-legal autopsy cases. *Journal of Forensic and Legal Medicine*. <https://doi.org/10.1016/j.jflm.2016.12.001>
- Hamza, R., Titouna, F.** (2016). A novel sensitive image encryption algorithm based on the Zaslavsky chaotic map. *Information Security Journal*, 25(4–6), 162–179. <https://doi.org/10.1080/19393555.2016.1212954>
- Hassler, U.** (2016). *Stochastic Processes and Calculus*. İçinde *Springer Verlag*. <http://link.springer.com/10.1007/978-3-319-23428-1>
- Husseinzadeh Kashan, A., Jalili, S., Karimiyan, S.** (2019). Premier league championship algorithm: A multi-population-based algorithm and its application on structural design optimization. İçinde *Studies in Computational Intelligence*. https://doi.org/10.1007/978-981-13-6569-0_11
- Istwal, Y., Verma, S. K.** (2018). Secured data transmission using improvised Arnold Cat map in WSN. *Proceedings - 2017 3rd International Conference on Advances in Computing, Communication and Automation (Fall), ICACCA 2017*. <https://doi.org/10.1109/ICACCAF.2017.8344665>
- Brownlee, J.** (2017). Difference between classification and regression in machine learning. *Machine Learning Mastery*, 25.
- Kakchapati, S., Choonpradub, C.** (2012). FORECASTING TUBERCULOSIS (TB) MORTALITY IN THAILAND USING MULTIVARIATE LINEAR REGRESSION. *Journal of Health Research*.
- Karci, A.** (2004). Novelty in the generation of initial population for genetic algorithms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-540-30133-2_35
- Karci, A.** (2007). Human being properties of saplings growing up algorithm. *ICCC 2007 - 5th IEEE International Conference on Computational Cybernetics, Proceedings*. <https://doi.org/10.1109/ICCCYB.2007.4402039>
- Karim, R. Bin.** (2019). *10 Gradient Descent Optimisation Algorithms + Cheat Sheet*. <https://www.kdnuggets.com/2019/06/gradient-descent-algorithms-cheat-sheet.html>

- Kaur, G., Arora, S.** (2018). Chaotic whale optimization algorithm. *Journal of Computational Design and Engineering*. <https://doi.org/10.1016/j.jcde.2017.12.006>
- Kaur, R., Kaur, S.** (2020). Non Dominated Sorting Genetic Algorithm Based Energy Aware Clustering Protocol. *International Journal of Scientific & Technology Research*, 9, 1853-1859.
- Kazemzadeh Azad, S.** (2018). Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. *Engineering Optimization*. <https://doi.org/10.1080/0305215X.2017.1284833>
- Kazimipour, B., Li, X., Qin, A. K.** (2014). A review of population initialization techniques for evolutionary algorithms. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, 2585–2592. <https://doi.org/10.1109/CEC.2014.6900618>
- Kazimipour, B., Omidvar, M. N., Li, X., Qin, A. K.** (2014). A novel hybridization of opposition-based learning and cooperative co-evolutionary for large-scale optimization. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, 2833–2840. <https://doi.org/10.1109/CEC.2014.6900639>
- Khairina, N., Harahap, M. K., Husein, A. M., Muhathir, Muliono, R.** (2019). Secure Data Encryption Through Combination of RSA Cryptography Random Key Algorithm and Quadratic Congruential Generator. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/1361/1/012027>
- Kimura, S., Matsumura, K.** (2005). Genetic algorithms using low-discrepancy sequences. *GECCO 2005 - Genetic and Evolutionary Computation Conference*. <https://doi.org/10.1145/1068009.1068225>
- Kishnani, M., Pareek, S., Gupta, R.** (2014). Optimal tuning of DC motor via simulated annealing. *2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014*. <https://doi.org/10.1109/ICAETR.2014.7012928>
- Kohli, M., Arora, S.** (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*. <https://doi.org/10.1016/j.jcde.2017.02.005>
- Kordov, K., Bonchev, L.** (2017). Using Circle Map for Audio Encryption Algorithm. *Mathematical and Software Engineering*.
- Koyuncu, H.** (2019). Parkinson's disease recognition using gauss map based chaotic particle swarm-neural network. *2019 International Conference on Engineering and Telecommunication, EnT 2019*. <https://doi.org/10.1109/EnT47717.2019.9030560>
- Krishnan, K., Karpagam, G.** (2014). Comparison of PID Controller Tuning Techniques for a FOPDT System. *International Journal of Current Engineering and Technology*.
- Krishnapriya, P. V, Smitha, S.** (2017). *Image Security Using Linear Feedback Shift Register*. 2(6), 2–5.
- Kubat, M.** (2017). An introduction to machine learning. Springer, Cham.
- Kulkarni, N., Prasad, N. R., Prasad, R.** (2018). A Novel Sensor Node Deployment using Low Discrepancy Sequences for WSN. *Wireless Personal Communications*. <https://doi.org/10.1007/s11277-017-5063-9>
- Kurniawan, D., Priyatna, B.** (2018). Pengamanan Data Berbasis Mobile Android Dengan

Penggabungan Linear Feedback Shift Register (Lfsr) Dan Modifikasi Matriks. *Jurnal TELEMATIKA MKOM*, 10(1), 42–46.

- Lee, K., Lee, M.** (2019). True random number generator (TRNG) utilizing FM radio signals for mobile and embedded devices in multi-access edge computing. *Sensors (Switzerland)*. <https://doi.org/10.3390/s19194130>
- Lee, K., Lee, S. Y., Seo, C., Yim, K.** (2018). TRNG (True Random Number Generator) Method Using Visible Spectrum for Secure Communication on 5G Network. *IEEE Access*, 6, 12838–12847. <https://doi.org/10.1109/ACCESS.2018.2799682>
- Liu, D., Liu, Z., Li, L., Zou, X.** (2016). A Low-Cost Low-Power Ring Oscillator-Based Truly Random Number Generator for Encryption on Smart Cards. *IEEE Transactions on Circuits and Systems II: Express Briefs*. <https://doi.org/10.1109/TCSII.2016.2530800>
- Liu, L., Miao, S.** (2017). An image encryption algorithm based on Baker map with varying parameter. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-016-3925-x>
- Liu, Q., Liu, X., Wu, J., Li, Y.** (2019). An improved NSGA-III algorithm using genetic K-means clustering algorithm. *IEEE Access*, 7, 185239–185249. <https://doi.org/10.1109/ACCESS.2019.2960531>
- Liu, S., Han, J.** (2017). Energy efficient stochastic computing with Sobol sequences. *Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017*. <https://doi.org/10.23919/DATE.2017.7927069>
- Lu, H., Wang, X., Fei, Z., Qiu, M.** (2014). The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms. *Mathematical Problems in Engineering, 2014*. <https://doi.org/10.1155/2014/924652>
- Luo, Y., Yu, J., Lai, W., Liu, L.** (2019). A novel chaotic image encryption algorithm based on improved baker map and logistic map. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-019-7453-3>
- Lydia, A. A., Francis, F. S.** (2019). Adagrad: An Optimizer for Stochastic Gradient Descent. *International Journal of Information and Computer Science*.
- Ma, X., Yuan, X., Cao, Z., Qi, B., Zhang, Z.** (2016). Quantum random number generation. *Nature Publishing Group, September 2015*, 1–9. <https://doi.org/10.1038/npjqi.2016.21>
- Mahmood, Z., L. Rana, J., Khare, A.** (2012). Symmetric Key Cryptography using Dynamic Key and Linear Congruential Generator (LCG). *International Journal of Computer Applications*. <https://doi.org/10.5120/7908-0973>
- Marghescu, A., Vasile, D. C., Svasta, P., Simion, E.** (2016). Personalized Ring Oscillator-based True Random Number Generator analysis using non-invasive attacks. *2016 IEEE 22nd International Symposium for Design and Technology in Electronic Packaging, SIITME 2016*. <https://doi.org/10.1109/SIITME.2016.7777253>
- Mehrdoust, F., Vajargah, B. F., Radmoghaddam, E.** (2013). A Robust and Accurate Quasi-Monte Carlo Algorithm for Estimating Eigenvalue of Homogeneous Integral Equations. *ISRN Computational Mathematics*. <https://doi.org/10.1155/2013/891029>
- Meranza-Castillón, M. O., Murillo-Escobar, M. A., López-Gutiérrez, R. M., Cruz-**

- Hernández, C.** (2019). Pseudorandom number generator based on enhanced Hénon map and its implementation. *AEU - International Journal of Electronics and Communications*. <https://doi.org/10.1016/j.aeue.2019.05.028>
- Mitsos, A., Chachuat, B., Barton, P. I.** (2008). Global optimization of algorithms. *AIChE Annual Meeting, Conference Proceedings*. <https://doi.org/10.1017/cbo9780511691881.010>
- Morokoff, W. J., Caflisch, R. E.** (1994). Quasi-Random Sequences and Their Discrepancies. *SIAM Journal on Scientific Computing*. <https://doi.org/10.1137/0915077>
- Moussi, A., Lidouh, A., Nqi, F. Z.** (2013). Simulation of asian options delta using a quadratic congruential pseudo-random numbers generator. *Applied Mathematical Sciences*. <https://doi.org/10.12988/ams.2013.36294>
- Moysis, L., Tutueva, A., Volos, C., Butusov, D., Munoz-Pacheco, J. M., Nistazakis, H.** (2020). A Two-Parameter Modified Logistic Map and Its Application to Random Bit Generation. *Symmetry*. <https://doi.org/10.3390/sym12050829>
- Müller, S.** (2014). *CPU Time Jitter Based Non-Physical True Random Number Generator*. 23–48.
- Munteanu, L., Ioan, R., Majercsik, L., Academy, R.** (2018). *ON THE COMPUTATION AND CONTROL OF A ROBOTIC SURGERY HYBRID SYSTEM*. Figure 4, 1–6.
- Naik, M. S.** (2020). *Generation of Random Numbers Using LCG and Faure Sequence*. 7(2), 840–843.
- Nair, A., Kiasaleh, K.** (2016). Chaotic circle map interleaver for OFDM. *2016 International Conference on Computing, Networking and Communications, ICCNC 2016*. <https://doi.org/10.1109/ICCNC.2016.7440578>
- Nasri, M., Hamdi, M.** (2019). LTE QoS Parameters Prediction Using Multivariate Linear Regression Algorithm. *Proceedings of the 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2019*. <https://doi.org/10.1109/ICIN.2019.8685914>
- Ohmer, M., Liesch, T., Goldscheider, N.** (2019). On the Optimal Spatial Design for Groundwater Level Monitoring Networks. *Water Resources Research*. <https://doi.org/10.1029/2019WR025728>
- Omran, M. G. H., Al-Sharhan, S., Salman, A., Clerc, M.** (2013). Studying the effect of using low-discrepancy sequences to initialize population-based optimization algorithms. *Computational Optimization and Applications*. <https://doi.org/10.1007/s10589-013-9559-2>
- Ozer, A. B.** (2010). CIDE: Chaotically Initialized Differential Evolution. *Expert Systems with Applications*, 37(6), 4632–4641. <https://doi.org/10.1016/j.eswa.2009.12.045>
- Pamuk, N.** (2013). Dinamik Sistemlerde Kaotik Zaman Dizilerinin Tespiti. *Balikesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 15(1), 77–91.
- Pan, W., Li, K., Wang, M., Wang, J., Jiang, B.** (2014). Adaptive randomness: A new population initialization method. *Mathematical Problems in Engineering*, 2014. <https://doi.org/10.1155/2014/975916>

- Pandey, M., Zakwan, M., Sharma, P. K., Ahmad, Z.** (2020). Multiple linear regression and genetic algorithm approaches to predict temporal scour depth near circular pier in non-cohesive sediment. *ISH Journal of Hydraulic Engineering*. <https://doi.org/10.1080/09715010.2018.1457455>
- Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., Ismail, M. I. S.** (2018). Design and Analysis of Computer Experiments Using Polynomial Regression and Latin Hypercube Sampling in Optimal Design of PID Controller. *Journal of Applied Research on Industrial Engineering*. <https://doi.org/10.22105/jarie.2018.141898.1051>
- Patel, K., Vala, J., Pandya, J.** (2014). COMPARISON OF VARIOUS CLASSIFICATION ALGORITHMS ON IRIS DATASETS USING WEKA. *International Journal of Advance Engineering and Research Development*. <https://doi.org/10.21090/ijaerd.0101001>
- Pausinger, F.** (2020). Greedy energy minimization can count in binary: point charges and the van der Corput sequence. *Annali di Matematica Pura ed Applicata*. <https://doi.org/10.1007/s10231-020-00990-7>
- Petura, O.** (2020). *True random number generators for cryptography : Design , securing and evaluation* Oto Petura To cite this version : HAL Id : tel-02895861.
- Pian, J., Wang, G., Li, B.** (2018). An Improved ABC Algorithm Based on Initial Population and Neighborhood Search. *IFAC-PapersOnLine*, 51(18), 251–256. <https://doi.org/10.1016/j.ifacol.2018.09.308>
- Ponce-Ortega, J. M., Hernández-Pérez, L. G.** (2019a). Optimization of Process Flowsheets through Metaheuristic Techniques. İçinde *Optimization of Process Flowsheets through Metaheuristic Techniques*. <https://doi.org/10.1007/978-3-319-91722-1>
- Pulat, M., Kocakoç, İ. D.** (2017). GEZGİN SATICI PROBLEMİNİN GENETİK ALGORİTMALARLA ÇÖZÜMÜNDE BAŞLANGIÇ POPÜLASYONUN BELİRLENMESİ. *JOEEP: Journal of Emerging Economics and Policy*.
- Rahimov, H., Babaie, M., Hassanabadi, H.** (2011). Improving Middle Square Method RNG Using Chaotic Map. *Applied Mathematics*. <https://doi.org/10.4236/am.2011.24062>
- Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A.** (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers and Mathematics with Applications*, 53(10), 1605–1614. <https://doi.org/10.1016/j.camwa.2006.07.013>
- Rajakumaran, G., Venkataraman, N., Mukkamala, R. R.** (2020). Denial of Service Attack Prediction Using Gradient Descent Algorithm. *SN Computer Science*. <https://doi.org/10.1007/s42979-019-0043-7>
- Robinson, S. M.** (2006a). Production Planning by Mixed Integer Programming. *Analysis*. Springer New York. <https://doi.org/10.1007/0-387-33477-7>
- Roy, A., Misra, A. P.** (2017). Audio signal encryption using chaotic Hénon map and lifting wavelet transforms. *European Physical Journal Plus*. <https://doi.org/10.1140/epjp/i2017-11808-x>
- Ruder, S.** (2016). *An overview of gradient descent optimization algorithms*. 1–14.

<http://arxiv.org/abs/1609.04747>

- Sahay, A., Pradhan, C.** (2018). Gauss iterated map based RGB image encryption approach. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*. <https://doi.org/10.1109/ICCSP.2017.8286437>
- Saraç, T., Bilgiçer, N.** (2020). ÇOK AMAÇLI PERMÜTASYON AKIŞ TİPİ ÇİZELGELEME PROBLEMİ İÇİN BİR NSGA-II ALGORİTMASI. *Endüstri Mühendisliği*. <https://doi.org/10.46465/endustrimuhendisligi.623030>
- Seyyarer, E., Hark, C., Ayata, F., Uçkan, T., Karci, A.** (2018). *Fidan Gelişim ve Lig Şampiyonluk Algoritmaları Kullanarak Aşırı-Akım Rölelerin Optimum Koordinasyonu* *Optimal coordination of over-current relays using saplings growing up and league championship algorithms*. 2, 1–14. <http://dergipark.gov.tr/download/article-file/564395>
- Shalev-Shwartz, S., Ben-David, S.** (2013). Understanding machine learning: From theory to algorithms. İçinde *Understanding Machine Learning: From Theory to Algorithms*. <https://doi.org/10.1017/CBO9781107298019>
- Shekhawat, M., Dhande, S. S.** (2011). Building an Iris Plant Data Classifier Using Neural Network Associative Classification. *International Journal of Advancements in Technology*.
- Shihabudheen, K. V., Mahesh, M., Pillai, G. N.** (2018). Particle swarm optimization based extreme learning neuro-fuzzy system for regression and classification. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2017.09.037>
- Siswanto, M., Rudiyanto, B.** (2017). Designing of quantum random number generator (QRNG) for security application. *Proceeding - 2017 3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech 2017*. <https://doi.org/10.1109/ICSITech.2017.8257124>
- Srinivas, N., Deb, K.** (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*. <https://doi.org/10.1162/evco.1994.2.3.221>
- Stipčević, M., Koç, Ç. K.** (2014). True Random Number Generators. İçinde *Open Problems in Mathematics and Computational Science*. https://doi.org/10.1007/978-3-319-10683-0_12
- Stork, J., Eiben, A. E., Bartz-Beielstein, T.** (2018). *A new Taxonomy of Continuous Global Optimization Algorithms*. <http://arxiv.org/abs/1808.08818>
- Tavana, M., Li, Z., Mobin, M., Komaki, M., Teymourian, E.** (2016). Multi-objective control chart design optimization using NSGA-III and MOPSO enhanced with DEA and TOPSIS. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2015.11.007>
- Tavas, V., Demirkol, A. Ş., Kılınc, S., Özoğuz, S., Toker, A., Zeki1, A.** (2008). *Tümleştirmeye Uygun Kaotik Osilatör Tabanlı Rastgele Sayı Üretici*. *April 2014*.
- Tiwari, M., Singh, R.** (2012). Comparative Investigation of K-Means and K-Medoid Algorithm on Iris Data. *International Journal of Engineering Research and Development*.

- Todorov, V., Ostromsky, T., Dimov, I., Fidanova, S.** (2020). Optimized Quasi-Monte Carlo Method Based on Low Discrepancy Sequences for Sensitivity Analysis in Air Pollution Modelling. *Communication Papers of the 2020 Federated Conference on Computer Science and Information Systems*. <https://doi.org/10.15439/2020f108>
- Tometzki, T., Engell, S.** (2011). Systematic initialization techniques for hybrid evolutionary algorithms for solving two-stage stochastic mixed-integer programs. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/TEVC.2010.2058121>
- Tuna, M., Fidan, C. B.** (2018). A Study on the importance of chaotic oscillators based on FPGA for true random number generating (TRNG) and chaotic systems. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(2), 413–423. <https://doi.org/10.17341/gazimmfd.416355>
- Türkay, M.** (2011). Optimizasyon Modelleri ve Çözüm Metodlari. URL: <http://home.ku.edu.tr/~mturkay/indr501/Optimizasyon.pdf>.
- Usama, M., Rehman, O., Memon, I., Rizvi, S.** (2019). An efficient construction of key-dependent substitution box based on chaotic sine map. *International Journal of Distributed Sensor Networks*. <https://doi.org/10.1177/1550147719895957>
- Uy, N. Q., Hoai, N. X., McKay, R. I., Tuan, P. M.** (2007). Initialising PSO with randomised low-discrepancy sequences: The comparative results. *2007 IEEE Congress on Evolutionary Computation, CEC 2007*. <https://doi.org/10.1109/CEC.2007.4424717>
- Vani, S., Rao, T. V. M.** (2019). An experimental approach towards the performance assessment of various optimizers on convolutional neural network. *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*. <https://doi.org/10.1109/ICOEI.2019.8862686>
- Venter, G.** (2010). Review of Optimization Techniques. *Encyclopedia of Aerospace Engineering, December 2010*. <https://doi.org/10.1002/9780470686652.eae495>
- Vesel, T.** (2019). Efficient Computation of Sobol' Quasi-Random Generator. *International Journal of Students' Research in Technology & Management*, 7(2), 01–04. <https://doi.org/10.18510/ijstrtm.2019.721>
- Villarrubia, G., De Paz, J. F., Chamoso, P., la Prieta, F. De.** (2018). Artificial neural networks used in optimization problems. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2017.04.075>
- Vishwas, C. G. M., Kunte, R. S.** (2020). An image cryptosystem based on tent map. *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Icssit*, 1069–1073. <https://doi.org/10.1109/ICSSIT48917.2020.9214291>
- Wang, R.** (2016). An Improved Nondominated Sorting Genetic Algorithm for Multiobjective Problem. *Mathematical Problems in Engineering*. <https://doi.org/10.1155/2016/1519542>
- Wen, H.** (2014). A review of the Hénon map and its physical interpretations. School of Physics Georgia Institute of Technology, Atlanta, GA, 30332-0430.
- White, R. C.** (1971). A survey of random methods for parameter optimization. *Simulation*, 17(5), 197–205. <https://doi.org/10.1177/003754977101700504>

- Wong, T.-T., Luk, W.-S., Heng, P.-A.** (1997). Sampling with Hammersley and Halton Points. *Journal of Graphics Tools*. <https://doi.org/10.1080/10867651.1997.10487471>
- Xu, Q., Wang, L., Wang, N., Hei, X., Zhao, L.** (2014). A review of opposition-based learning from 2005 to 2012. *Engineering Applications of Artificial Intelligence*, 29, 1–12. <https://doi.org/10.1016/j.engappai.2013.12.004>
- Yakıcı, E.** (2018). A multiobjective fleet location problem solved by adaptation of evolutionary algorithms NSGA-II and SMS-EMOA. *Pamukkale University Journal of Engineering Sciences*. <https://doi.org/10.5505/pajes.2017.20688>
- Yang, X.-S.** (2010). Engineering Optimization. İçinde *Engineering Optimization: An Introduction with Metaheuristic Applications*. John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470640425>
- Yang, X.** (2010). *Engineering Optimization and Metaheuristics Metaheuristics and Engineering Optimization Introduction Design Optimization Conventional Approach Metaheuristics Genetic Algorithms & Simulated Annealing Particle Swarm Optimization Firefly Algorithm & Cuckoo S.*
- Yazan, E., Talu, M. F.** (2017). Comparison of the stochastic gradient descent based optimization techniques. <https://doi.org/10.1109/idap.2017.8090299>
- Yosefnezhad Irani, B., Ayubi, P., Amani Jabalkandi, F., Yousefi Valandar, M., Jafari Barani, M.** (2019). Digital image scrambling based on a new one-dimensional coupled Sine map. *Nonlinear Dynamics*. <https://doi.org/10.1007/s11071-019-05157-5>
- Yu, F., Li, L., Tang, Q., Cai, S., Song, Y., Xu, Q.** (2019). A Survey on True Random Number Generators Based on Chaos. *Discrete Dynamics in Nature and Society*. <https://doi.org/10.1155/2019/2545123>
- Zaki, S., H. Zaied, A. E.-N.** (2019). Meta-heuristics Algorithms: A survey. *International Journal of Engineering Trends and Technology*, 67(5), 67–74. <https://doi.org/10.14445/22315381/IJETT-V67I5P210>
- Zeiler, M. D.** (2012). ADADELTA: An Adaptive Learning Rate Method. <http://arxiv.org/abs/1212.5701>
- Zeng, T.** (2016). Using Halton Sequences in Random Parameters Logit Models. *Journal of Statistical and Econometric Methods*, 5(1), 1792–6939. [http://www.scienpress.com/Upload/JSEM%2FVol 5_1_4.pdf](http://www.scienpress.com/Upload/JSEM%2FVol%205_1_4.pdf)
- Zhang, C., Peng, T., Li, C., Fu, W., Xia, X., Xue, X.** (2019). Multiobjective optimization of a fractional-order PID controller for pumped turbine governing system using an improved NSGA-III algorithm under multiworking conditions. *Complexity*. <https://doi.org/10.1155/2019/5826873>
- Zhang, G., Gao, L., Shi, Y.** (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2010.08.145>
- Zhou, X., Miao, F., Ma, H.** (2018). Genetic algorithm with an improved initial population technique for automatic clustering of low-dimensional data. *Information (Switzerland)*, 9(4), 1–22. <https://doi.org/10.3390/info9040101>

ÖZGEÇMİŞ

Ad-Soyad : **Ebubekir SEYYARER**

ÖĞRENİM DURUMU:

- **Lisans** : 2002-2006, Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek Lisans** : 2011-2016, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği
- **Doktora** : 2016-2021, İnönü Üniversite, Bilgisayar Mühendisliği A.B.D, Bilgisayar Bilimleri Bilim Dalı

MESLEKİ DENEYİM:

- 1 Yüzüncü Yıl Üniversitesi (2013-Devam ediyor)
Öğretim Görevlisi
- 2 Hakkâri Üniversitesi (2010-2013)
Öğretim Görevlisi
- 3 İvme Bilişim ve Danışmanlık (LOGO çözüm Ortağı, 17.03.2010- 01.07.2010)
Yazılım Uzmanı
- 4 İvme Bilişim ve Danışmanlık (LOGO çözüm Ortağı, 01.05.2008- 15.07.2009)
Yazılım Uzmanı
- 5 Compa Bilişim (Honeywell çözüm ortağı, 24.04.2007- 15.03.2008)
Akıllı Ev Projeleri Geliştirme (AR-GE)
- 6 ŞEMKA Bilgisayar ve Yazılım Ltd. Şti. (11.12.2006- 08.04.2007)
Yazılım Uzmanı
- 7 KALEDATA A.Ş.'de 2004-2005
Stajyer
- 8 BEKO ELEKTRONİK A.Ş.'de 2003-2004
Stajyer

DOKTORA TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **SEYYARER, E.**, Ayata, F., Uçkan, T., Karcı, A. (2020). Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması Ve Kıyaslanması. *Computer Science* , 5 (2) , 90-98 . Retrieved from <https://dergipark.org.tr/tr/pub/bbd/issue/57870/752132>
- **SEYYARER, E.**, Ayata, F., Uçkan, T., Hark, C., Karcı, A., (2019). Comparison of Optimization Algorithms Used in Deep Learning by Using Caltech 101 Data Set . International Conference on Data Science, Machine Learning and Statistics - 2019 (DMS-2019) (pp.18). Van, Turkey
- **SEYYARER, E.**, Uçkan, T., Hark, C., Ayata, F., İnan, M., Karcı, A. (2019, September). Applications and comparisons of optimization algorithms used in convolutional neural networks. In 2019 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-6). IEEE.
- **SEYYARER, E.**, Hark, C., Ayata, F., Uçkan, T., Karcı, A. (2018). Fidan Gelişim ve Lig Şampiyonluk Algoritmaları Kullanarak Aşırı-Akım Rölelerin Optimum Koordinasyonu . *Computer Science* , 3 (2) , 1-14 . Retrieved from <https://dergipark.org.tr/tr/pub/bbd/issue/40060/410645>
- **SEYYARER, A¹.**, Akdağ, O., Hark, C., Karcı, A., Yeroğlu, C. (2017, September). Overcurrent relay coordination of 154/34, 5 kV Hasaңcelebi substation by league championship algorithm. In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-6). IEEE.

¹ 26.03.2018 tarihinde 5490 sayılı kanunun geçici 8. maddesi uyarınca ismim “Abubekir” iken “Ebubekir” olarak deęişmiştir.