

T.C.

EGE ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

SİSTEM GÜVENİLİRLİĞİNDE OPTİMİZASYON

Şefika Büşra MERT

Danışman: Dr. Öğr. Üyesi Özge ELMASTAŞ GÜLTEKİN

İstatistik Anabilim Dalı

İstatistik Yüksek Lisans Programı

Ocak, 2022

İzmir



EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**ETİK KURALLARA UYGUNLUK BEYANI**

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum “Sistem Güvenilirliğinde Optimizasyon” başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

21/01/2022

Şefika Büşra MERT



ÖZET

SİSTEM GÜVENİLİRLİĞİNDE OPTİMİZASYON

MERT, Şefika Büşra

Yüksek Lisans Tezi, İstatistik Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Özge ELMASTAŞ GÜLTEKİN

21 Ocak 2022, 66 Sayfa

Günümüzde sistemlerin karmaşık yapısı ve ilerleyen teknoloji sayesinde sistemlerin doğru çalışabilmesi ve kontrol edilebilir durumda olması güvenilirlik konusunu önemli bir hale getirmiştir. Sistemler birbiri ile etkileşim halinde bulunan alt bileşenlerden oluştuğu için sistemde bulunan bileşenlerin güvenilirlikleri de sistemin genel güvenilirliği üzerinde etkili olur. Sistem üzerinde olumlu etkiye sahip olan faktörler problemin amacına göre optimizasyon yöntemleri kullanılarak iyileştirilebilirken, olumsuz faktörlerin de en aza indirilmesi veya ortadan kaldırılması gerekir. Bu yöntemler kullanılırken hedeflenen amaca en yakın çözümleri bulabilmek için meta-sezgisel algoritmaların kullanımı tercih edilebilir. Gerçek hayattan esinlenerek oluşturulan bu algoritmalara Öğretme-Öğrenme Esaslı Optimizasyon algoritması (Teaching-Learning Based Optimization-TLBO), Genetik Algoritmalar (GA) örnek olarak gösterilebilir. Öğretme-Öğrenme Esaslı Optimizasyon (TLBO) algoritmaları, öğretmen-öğrenci ilişkilerini ve öğrencilerin kendi aralarındaki etkileşimlerinden yola çıkarak mevcut problemin iyileştirilmesi üzerine geliştirilen bir optimizasyon algoritmasıdır. Bu çalışmada doğrusal n 'den k çıkışlı ve doğrusal ardışık n 'den k çıkışlı F sistemlerin güvenilirliğinin optimize edilmesi için TLBO algoritması ele alınmış ve bununla birlikte sistem güvenilirliğinin en yüksek değerini veren bileşenlerin güvenilirliklerinin elde edilmesi amaçlanmıştır. Ayrıca söz konusu sistem için Genetik Algoritma ile de sonuçlar elde edilmiş olup, TLBO'dan elde edilen sonuçlar ile karşılaştırılmıştır.

Anahtar Kelimeler: Sistem Güvenilirliği Optimizasyonu, n 'den k Çıkışlı Sistemler, Öğretme-Öğrenme Esaslı Optimizasyon Algoritması, Genetik Algoritma.



ABSTRACT**OPTIMIZATION IN SYSTEM RELIABILITY**

MERT, Şefika Büşra

Master Thesis, Department of Statistics

Supervisor: Assist. Prof. Dr. Özge ELMASTAŞ GÜLTEKİN

21 Ocak 2022, 66 pages

Today, thanks to the complex structure of the systems and the advancing technology, the correct operation and controllability of the systems has made the issue of reliability important. Many applications have been carried out in engineering, from the proper functioning of the functions expected from the system to minimizing the errors in the system and the sub-components that make up the system. Since the systems consist of sub-components interacting with each other, the reliability of the components in the system also affects the overall reliability of the system. While the factors that have a positive effect on the system can be improved by using optimization methods according to the purpose of the problem, the negative factors should be minimized or eliminated. When using these methods, the use of meta-heuristic algorithms may be preferred to find the closest solutions to the targeted purpose. Teaching-Learning Based Optimization (TLBO), Genetic Algorithms (GA), can be given as examples to these algorithms inspired by real life. Teaching-Learning Based Optimization (TLBO) algorithms are an optimization algorithm developed to improve the current problem based on teacher-student relationships and interactions between students. In this study, TLBO algorithm is considered to optimize the reliability of a linear k -out-of- n : F systems and linear consecutive k -out-of- n : F system, and it is aimed to obtain the reliability of the components that give the highest value of system reliability. In addition, results were obtained with the Genetic Algorithm for the system in question and compared with the results obtained from TLBO.

Key Words: System Reliability Optimization, k -out-of- n Systems, Teaching-Learning Based Optimization Algorithm, Genetic Algorithm.



ÖNSÖZ

Bu tez çalışması, Ege Üniversitesi Fen Bilimleri Enstitüsü'ne bağlı İstatistik Anabilim Dalı'nda hazırlanan bir yüksek lisans tezidir. Günümüzde sistemlerin karmaşık yapısı ve ilerleyen teknoloji sayesinde sistemlerin doğru çalışabilmesi ve kontrol edilebilir durumda olması güvenilirlik konusunu önemli bir hale getirmiştir. Sistemden beklenen işlevlerin uygun bir şekilde çalışabilmesinden, sistemde ve sistemi oluşturan alt bileşenlerdeki hataların minimuma indirilmesine kadar mühendislikte pek çok uygulama gerçekleştirilmiştir. Sistem güvenilirliği hesaplanmasında karmaşık sistemlerin daha kolay çözümlenebilmesi için son zamanlarda yapay zekanın alt bir dalı olan meta-sezgisel yaklaşımların kullanımı kolaylık sağlamaktadır.

Bu çalışmada doğrusal n 'den k çıkışlı ve doğrusal ardışık n 'den k çıkışlı sistemlerin güvenilirliğini maksimize etmek ve en yüksek sistem güvenilirlik değerini veren bileşen güvenilirliklerini bulabilmek için meta-sezgisel yaklaşım gurubundan iki farklı algoritma kullanılıp sonuç karşılaştırmaları yapılmıştır.

İZMİR

21/01/2022

Şefika Büşra MERT

İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	vii
ÖNSÖZ.....	ix
İÇİNDEKİLER.....	xi
ŞEKİLLER DİZİNİ	xiii
TABLOLAR DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ.....	xvii
1.GİRİŞ.....	1
2. LİTERATÜR TARAMASI	4
3. GÜVENİLİRLİK VE SİSTEM GÜVENİLİRLİĞİ.....	6
3.1 Tamir Edilebilir ve Tamir Edilemez Sistem Türleri	7
3.2 Seri Sistemler	8
3.3 Paralel Sistemler.....	8
3.4 Karma Sistemler	9
3.5 n 'den k çıkışlı F ve G Sistemleri ($kn/$: $F-G$) ve Güvenilirlikleri	10
3.5.1 n 'den k çıkışlı Tamir Edilemez Sistemler	14
3.5.2 n 'den k çıkışlı Tamir Edilebilir Sistemler	15
3.6 Doğrusal Ardışık n 'den k çıkışlı F ve G Sistemleri ($DA/kn/$: $F-G$)	16
3.6.1 Doğrusal Ardışık n 'den k çıkışlı G Sistemleri ($DA/kn/$: G).....	16
3.6.2 Doğrusal Ardışık n 'den k çıkışlı F Sistemleri ($DA/kn/$: F)	17
4. SİSTEM GÜVENİLİRLİĞİNDE OPTİMİZASYON.....	19
4.1 Yapay Zekâ.....	19
4.2 Meta-Sezgisel Algoritmalar	20
4.2.1 Öğretme-Öğrenme Esaslı Optimizasyon Algoritması (TLBO).....	20
4.2.2 Genetik Algoritmalar (GA)	29

İÇİNDEKİLER(Devam)

Sayfa

5. n'den k ÇIKIŞLI SİSTEMLERDE GÜVENİLİRLİK OPTİMİZASYONU.....	44
5.1 Doğrusal 10'dan 2 çıkışlı F Sisteminin TLBO ve GA ile Optimizasyonları ve Karşılaştırılması.....	45
5.2 Doğrusal Ardışık 10'dan 2 çıkışlı F Sisteminin TLBO ve GA ile Optimizasyonları ve Karşılaştırılması	49
6. SONUÇ	56
KAYNAKLAR DİZİNİ	58
TEŞEKKÜR	64
ÖZGEÇMİŞ	65

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
Şekil 3. 1. Seri bağlı sistem yapısı	8
Şekil 3. 2. Paralel bağlı sistem yapısı.....	9
Şekil 3. 3. Karma sistem yapısı	10
Şekil 3. 4. 3'ten 2 çıkışlı G sistemi yapısı örneği	12
Şekil 3. 5. Doğrusal ardışık 4'ten 2 çıkışlı G sistemi örneği	16
Şekil 3. 6. Doğrusal ardışık 4'ten 2 çıkışlı F sistemi örneği	17
Şekil 4. 1. TLBO akış diyagramı	24
Şekil 4. 2. Denklem çözümleri	25
Şekil 4. 3. Matlab ile her iterasyon sonucundaki güvenilirlik değerleri	28
Şekil 4. 4. Genetik algoritma temel kavramlar	30
Şekil 4. 5. Genetik algoritma genel döngüsü	30
Şekil 4. 6. Rulet tekerleği seçimi	32
Şekil 4. 7. Turnuva seçimi	33
Şekil 4. 8. Tek nokta çaprazlama	33
Şekil 4. 9. İki nokta çaprazlama	33
Şekil 4. 10. Çok nokta çaprazlama	34
Şekil 4. 11. Uniform çaprazlama.....	34
Şekil 4. 12. Mutasyon operatörü	34
Şekil 4. 13. Genetik algoritma akış diyagramı	35
Şekil 4. 14. Permütasyon kodlama başlangıç kromozom	36
Şekil 4. 15. Permütasyon kodlamada tek nokta çaprazlama	36
Şekil 4. 16. Permütasyon kodlamada XO çaprazlama	37
Şekil 4. 17. Permütasyon kodlamada mutasyon	37
Şekil 4. 18. Birnbaum önem ölçüsüyle hesaplanan pozisyonlara bileşen atama ..	40
Şekil 4. 19. Birnbaum önem tabanlı genetik algoritma başlangıç popülasyonu ...	41
Şekil 4. 20. BIGA 20. iterasyon sonucundaki sistem güvenilirlikleri.....	42
Şekil 4. 21. GA 20. iterasyon sonucundaki sistem güvenilirlikleri.....	42
Şekil 4. 22. GGA başlangıç kromozom	43

ŞEKİLLER DİZİNİ (Devam)

Şekil

Sayfa

Şekil 5. 1. TLBO her iterasyon için en iyi sistem güvenilirlikleri grafiği (DF/10/2)	47
Şekil 5. 2. GA her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DF/10/2)	48
Şekil 5. 3. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (DF/10/2)	48
Şekil 5. 4. TLBO her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DAF/10/2)	50
Şekil 5. 5. GA her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DAF/10/2)	51
Şekil 5. 6. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (DAF/10/2)	52
Şekil 5. 7. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (50-50 DAF/10/2)	52
Şekil 5. 8. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (100-100 DAF/10/2)	53
Şekil 5. 9. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (50-50 DF/10/2)	54
Şekil 5. 10. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (100- 100 DF/10/2)	54

TABLolar DİZİNİTabloSayfa

Tablo 5. 1. TLBO ve GA yöntemleri için gerekli olan tanımlamalar 44

Tablo 5. 2. TLBO ve GA yöntemlerine göre sistem karşılaştırmaları 55





SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
x_i	i . Bileşen durumu
n	Bileşen sayısı
k	Sistemin başarılı ya da hatalı olması için gereken bileşen sayısı
x	Durum vektörü
ϕ	Sistem durumu
$\phi(x)$	Sistem yapı fonksiyonu
p_i	i . Bileşenin güvenilirliği
q_i	i . Bileşenin güvenilmezliği
R	Sistem güvenilirliği
Q	Sistem güvenilmezliği
$kn/: F$	n 'den k -çıkışlı F sistemi
$kn/: G$	n 'den k -çıkışlı G sistemi
$D/kn/: F$	Doğrusal n 'den k -çıkışlı F sistemi
$D/kn/: G$	Doğrusal n 'den k -çıkışlı G sistemi
$DA/kn/: F$	Doğrusal ardışık n 'den k -çıkışlı F sistemi
$DA/kn/: G$	Doğrusal ardışık n 'den k -çıkışlı G sistemi
$I_B^i(\underline{p})$	i . Bileşenin Birnbaum önem ölçüsü ile güvenilirliği
$\underline{X}$	n tane bileşenin durum vektörü
$\underline{p}$	n tane bileşenin güvenilirlik vektörü
$\phi(\underline{X})$	Sistem durumu
$R(\underline{p})$	Sistem güvenilirliği

SİMGELER VE KISALTMALAR DİZİNİ (Devam)

Kısaltmalar

PDF	Sistem Ömrünün Olasılık Yoğunluk Fonksiyonu
CDF	Sistem Ömrünün Dağılım Fonksiyonu
TLBO	Öğrenme-Öğretme Esaslı Optimizasyon Algoritması
GA	Genetik Algoritma
GGA	Gerçek Kodlu Genetik Algoritma
BI	Birnbaum Önem Ölçüsü
BIGA	Birnbaum Önem Tabanlı Genetik Algoritma
DF/10/2	Doğrusal 10'dan 2 Çıkışlı F Sistemi
DAF/10/2	Doğrusal Ardışık 10'dan 2 Çıkışlı F Sistemi
DAF/10/3	Doğrusal Ardışık 10'dan 3 Çıkışlı F Sistemi
DG/10/2	Doğrusal 10'dan 2 Çıkışlı G Sistemi
DAG/10/2	Doğrusal Ardışık 10'dan 2 Çıkışlı G Sistemi
DAG/10/3	Doğrusal Ardışık 10'dan 3 Çıkışlı G Sistemi

1.GİRİŞ

Herhangi bir sistemde, sistemden beklenen performansında belli kısıtlar altında ve belirli bir periyot boyunca bozulmadan çalışabilmesi güvenilirlik kavramını meydana getirir. Buna göre güvenilirlik, sistem için belirlenmiş sınırlar ve kısıtlar çerçevesinde, aynı sistemin başarılı bir şekilde çalışması olasılığı olarak tanımlanmıştır.

Güvenilirlik, bir bileşen, sistem veya ürünün sahip olması gereken en önemli unsur olarak karşımıza çıkmaktadır. Örneğin güvenilirliğe bir firmanın ürünü açısından baktığımızda, ürün ne kadar güvenilir ise firmanın da o kadar güvenilir olduğu düşünülmektedir. Öte yandan müşteri tarafından baktığımızda ise güvenilir ürünlerin müşteri memnuniyetini arttırdığı görülmektedir. Maliyet açısından düşünüldüğünde çoğu zaman maliyeti artırıcı bir unsur olarak gözükmeye rağmen yapılan güvenilirlik değerlendirmeleri sonucunda maliyet ve güvenilirlik arasında optimum bir düzey yakalanarak maksimum performans elde edilebilir (Ersalıcı, 2012).

Sistemin güvenilirliği tanımlanırken, yerine getirmesi gereken fonksiyon, bozulmayı oluşturan sebebin tanımıyla birlikte açıkça belirtilmelidir. Bir sistemin yaşam süresi ölçülmeden önce sistemden beklenen işlev ve neyin “bozulma” olduğu konusu çalışma koşullarına göre değişiklik göstereceğinden açıkça belirlenmelidir (Varan, 2007).

Sistemler birbirleri ile etkileşimde bulunan alt bileşenlerin oluşturduğu yapılardır. Alt bileşenlerdeki olumlu ya da olumsuz her türlü etken sistemin genel durumu üzerinde etkili olmaktadır. Temelde en basit olarak bilinen sistem türleri seri ve paralel yapıli sistemlerdir. Bu sistemlerin bir arada bulunduđu yapı da karmaşık sistemleri meydana getirir. Temelde basit ve anlaşılır sistemlerin çözümü ve hesaplamaya ilişkin ifadeleri de oldukça kolaydır. Ancak sistemler karmaşık yapıda olduğunda, sisteme ilişkin hesaplamalarda da zorluklar görülebilir. Karmaşık yapıli sistemlere, seri ve paralel sistemlerin özel durumları olarak ele alınan doğrusal n 'den k çıkışlı ve doğrusal ardışık n 'den k çıkışlı sistemler örnek olarak gösterilebilirler.

n 'den k çıkışlı sistemlerde sistem güvenilirliği, her bileşen aynı türdeyse veya aynı arıza sürelerine sahipse kolayca hesaplanabilir. Bir alt sistemdeki bileşenler aynı değilse, sistem (veya alt sistem) güvenilirliğinin tahmini daha zor hale gelir. Bu durumda, çok terimli bir dağılım düşünülmelidir. Bir alt sistemdeki bileşenler aynı olmadığında, bu genellikle bileşen karıştırma olarak adlandırılır. Bileşen karıştırmalı bir n 'den k çıkışlı sisteminin güvenilirliği için ise oluşturulacak matematiksel denklemler daha karmaşık bir hale gelecektir (Coit, 2000).

Sistem güvenilirliği hesaplamalarında optimizasyon yöntemlerinin kullanımı sıklıkla tercih edilir. Sistem güvenilirliği optimizasyonunda tasarımın amacı, maliyet, ağırlık gibi sistem gereksinimi olan kısıtları karşılayabilmek için sistem güvenilirliğini en üst düzeye çıkarmaktır. Uygulamalarda, sistem gereksinimlerini karşılayabilmek için sistem güvenilirliğini tasarlamak ve geliştirmek çok karmaşık sorunlara yol açabilir. Bu yüzden genelde bir tasarımın amacı, sisteme ilişkin kısıtları göz önünde bulundurarak sistem güvenilirliğini en üst düzeye çıkarmak için tasarım alternatiflerinden bileşen tipini ve yedek bileşen sayısını seçmektir. Her bir alt sistemin tasarım alternatifleri, aynı olmayan bileşenlerle elde edilebilir. Özdeş olmayan bileşen seçimleri, sorunun arama alanını büyük ölçüde genişletir. Bu nedenle, tasarım alternatiflerinin sayısı fazla olduğunda en uygun çözümü elde etmek zordur (Coit, 2011).

Sistem güvenilirliğinde optimizasyon süreci, üzerinde optimizasyon çalışması yapılacak sistem yapısına ve çeşitli kriterlere göre farklı şekillerde sınıflandırılabilir. Analitik optimizasyon yöntemleri, tüm çözüm kümesini tarayarak çözüme ulaşmayı amaçlarken, büyük boyutlu problemlerde sonuca ulaşmak çok uzun sürer. Bu yüzden çözüm kümesine sezgisel olarak yaklaşan ve en iyiye yakın bir çözüme ulaşmayı amaçlamayan sezgisel ve meta-sezgisel optimizasyon yöntemlerinin kullanılması daha avantajlıdır.

Bu tez çalışmasının amacı, karmaşık sistem yapıları (n 'den k çıkışlı sistemler) için sistem güvenilirliği optimizasyonunu gerçekleştirerek en iyi sistem güvenilirliğine ulaşmak ve en iyi sistem güvenilirliği değerini veren bileşen güvenilirliklerini bulmaktır. Aynı zamanda sistem güvenilirlik optimizasyonu hesaplamaları için gerçek hayattan esinlenerek tasarlanmış olan ve yapay zekanın alt dallarından biri olan meta-sezgisel yöntemlerden Öğrenme-Öğretme

Optimizasyon Algoritması (TLBO) ve Genetik Algoritma (GA) kullanılmış, her iki yöntem sonuçları karşılaştırılmıştır.

Bu çalışmanın ikinci bölümünde konuya ilişkin literatür taramasına yer verilmiştir.

Üçüncü bölümde, güvenilirlik ve sistem güvenilirliği ele alınarak konuya ilişkin temel bilgiler sunulmuş ve sistem türleri incelenmiştir.

Dördüncü bölümde, sistem güvenilirliğinde optimizasyon tanımlamasına yer verilmiş ve ardından sistemlerde güvenilirlik optimizasyonunu meta-sezgisel yöntemlerle gerçekleştirebilmek için Öğrenme-Öğretme Optimizasyon Algoritması (TLBO) ve Genetik Algoritma (GA) yöntemleri örneklerle açıklamalı olarak anlatılmıştır. Küçük boyutlu bileşenlerde öncelikle TLBO ve daha sonra GA yöntemleri ayrıntılı bir şekilde ele alınarak adım adım incelenmiş olup sonuçlarla ilgili yorumlara yer verilmiştir.

Çalışmanın beşinci bölümünde, doğrusal n 'den k çıkışlı F sistemi ile doğrusal ardışık n 'den k çıkışlı F sistemlerinin güvenilirlik optimizasyon uygulamasına yer verilmiş olup çalışma sonuçlarına ilişkin bulgular her iki yöntemle de karşılaştırılarak açıklanmıştır.

Altıncı bölümde çalışmanın genel sonuçlarına değinilerek öneride bulunulmuştur.

2. LİTERATÜR TARAMASI

Sistem güvenilirliği optimizasyonunu çözmenin birçok yolu vardır. Fyffe ve ark. (1968), bileşeni karıştırmadan n 'den k çıkışlı alt sistem güvenilirlik problemi için bir bileşen tahsis yöntemi önermiştir. Sistem maliyetini ve ağırlık kısıtlamalarını karşılarken sistem güvenilirliğini en üst düzeye çıkarmayı hedeflemişlerdir. Kuo ve Prasad (2000), sistem güvenilirliği optimizasyonuna açıklamalı bir genel bakış sunmuşlardır. Sistem güvenilirliğini en üst düzeye çıkarabilmek için bileşen güvenilirliğinin artırılması ve paralel olarak yedekli bileşenlerin sağlanmasından, sezgisel ve meta-sezgisel algoritmaların geliştirilmesine kadar birçok önemli konuya değinmişlerdir. Coit ve Liu (2000), n 'den k çıkışlı alt sistemlere sahip bir sistemin güvenilirliğinin optimizasyonunu ele almıştır. Du ve Chen (2002), verimli olasılık tasarımı için sıralı optimizasyon ve güvenilirlik değerlendirme yöntemi önermişlerdir. Youn ve ark. (2005), güvenilirliğe dayalı tasarım optimizasyonu için zenginleştirilmiş bir performans ölçümü sunmuşlardır. Onishi ve ark. (2007), bileşen türlerinin karıştırılmasıyla artıklık tahsis problemini çözmek için geliştirilmiş bir kısıtlama yöntemi önermiştir. Varan (2007), bir bilgisayar kasası modeli için sistem güvenilirliği analizi gerçekleştirmiştir. Toğan (2009), açık deniz yapılarının güvenilirliğine dayalı optimizasyon yöntemi uygulaması gerçekleştirmiştir. Ulucenk (2009), havacılık yapılarında güvenilirlik temelli tasarım optimizasyon tekniklerinin uygulanmasını sunmuştur. Kızılloluk ve Alataş (2012), sezgisel optimizasyon algoritmalarıyla ilgili güncel tanımlamalar sunmuşlar. Gökdere ve Gürcan (2016), mühendislik uygulamalarında kullanılan ardışık n 'den k çıkışlı sistemlerin güvenilirlik analizini bileşenlerin stres altındaki çalışma risklerini göz önünde bulundurarak hesaplamışlardır. Yine Gökdere ve Güral (2018), sistemdeki her bir bileşenin de önem değerlerini hesaplayarak genel sistem üzerindeki etkinliğini gösteren birnbaum önem tabanlı genetik algoritma ile doğrusal ardışık n 'den k çıkışlı sistemlerin güvenilirlik optimizasyonu üzerine bir çalışma sunmuştur. Rao ve ark. (2011), meta-sezgisel bir yöntem olan öğretme-öğrenme esaslı optimizasyon algoritması ile kısıtlı mekanik tasarım optimizasyon problemleri için yeni bir yöntem geliştirmişlerdir. Yine Rao ve ark. (2012), çalışmalarında öğretmen ve öğrenci aralarındaki etkileşimden esinlenerek tasarlanmış olan sürekli, doğrusal olmayan büyük ölçekli problemler için öğretme-öğrenme optimizasyon algoritması

kullanmışlardır. Soltani (2014), iki durumlu tamir edilemez sistemlerin güvenilirlik optimizasyonu için bir makale sunmuştur. Araştırmasında farklı bakış açılarından derlediği artıklık tahsisi, güvenilirlik tahsisi ve her ikisinin de bir arada bulunduğu üç tür güvenilirlik optimizasyon yöntemini incelemiştir.

Saleem (2017), seri ve paralel sistemlerin güvenilirlik optimizasyonu için Genetik Algoritma yöntemi ile istatistiksel analiz kullanarak maliyet ve ağırlık gibi kısıtlar altında sistem güvenilirliğini optimize etmeye çalışmıştır. Cao ve ark. (2017), belirsizliği dikkate alan çok amaçlı sistem güvenilirliği optimizasyon problemlerini çözmek için üç aşamalı bir yaklaşım sunmuşlardır. Belirsizlik sorununu çözmek için, her bir bileşenin deterministik güvenilirliğini belirleyerek artıklık tahsis problemine entropi tabanlı bir yaklaşım önerilmişlerdir. Wang ve ark. (2018), önem ölçüsüne dayalı bileşen iyileştirme maliyeti ile sistemlerin güvenilirlik optimizasyonuna dayalı bir çalışma gerçekleştirmiştir. Önem ölçüleriyle bakım maliyeti kısıtı altında sistemdeki en zayıf bileşenleri tanımlayarak, sistemin genel güvenilirliğini olumlu yönde etkilemeye çalışmıştır. Kumar ve ark. (2018), öğretim-öğrenme tabanlı optimizasyon algoritması kullanarak tekstil endüstrilerindeki kumaş terbiye sistemlerinin parametre optimizasyonu gerçekleştirmişlerdir. Coit ve Zio (2019), sistem güvenilirliği optimizasyonunun gelişimini, güvenilirlik optimizasyon problemlerini, artık tahsis problemlerini ve güvenilirlik tahsis problemlerini ele alan bir çalışma sunmuştur. Zhao ve ark. (2019), doğrusal ardışık n 'den k çıkışlı F sistemlerinin güvenilirlik optimizasyonu için birnbaum önem tabanlı kuantum genetik algoritmasını öne sürmüşlerdir. Demiralp (2020), ardışık n 'den k çıkışlı sistemlerin güvenilirliği üzerine sınır yaklaşımları sunmuştur. Yine Demiralp ve Güngör (2021), doğrusal ve dairesel ardışık k sistemleri için literatürde önerilen güvenilirlik sınırlarının karşılaştırmalarını sunmuşlardır.

3. GÜVENİLİRLİK VE SİSTEM GÜVENİLİRLİĞİ

Güvenilirlik, “zamanın bir fonksiyonu” olarak tanımlanabilir. Belirlenmiş bir zamanda, sistem bileşenlerinin belirli bir oranı bozulmadan çalışmaya devam eder. Sonuçta güvenilirlik çalışmaya devam eden oranı temsil eder (Rausand and Hoyland, 2004).

Bir birimin güvenilirlik fonksiyonu aşağıdaki gibi tanımlanır:

$$R(t) = 1 - F(t) = P(T > t) \quad t > 0$$

Burada $R(t)$ ile ifade edilen güvenilirlik fonksiyonudur. R güvenilirliği ve t ise zaman aralığını göstermektedir. Hata olasılığı $F(t)$ ise sistemin veya sisteme ait bir bileşenin zaman aralığının (t) aşılmasından önce bozulması olasılığıdır.

Sistem güvenilirliğinde ise güvenilirlik değerlendirmesi yapılacak sistem birçok eleman ve alt sistemden oluşabilir. Tüm bu elemanların ve alt sistemlerin de tek başlarına fonksiyonlarını yerine getirebilme ve başarısız olma olasılıkları vardır. Dolayısıyla bir bütün olarak sistemin güvenilirliği, kendisini meydana getiren tüm parçaları kapsamaktadır. Sistemin gerçek güvenilirlik değeri de tam olarak bilinemeyeceği için bu değere çok yakın olabilecek bir değer, olasılık hesaplamaları ve istatistiksel yöntemler ile elde edilebilir (Dündar, 2002).

Sistemi oluşturan etmenler, o sistemin belirlenebilen ve birbirine bağlı olan, sistemi çalışır duruma getiren ve aynı zamanda sisteme geçerlilik kazandıran somut elemanlardır (Dinç ve Abdioğlu, 2009).

Sistem güvenilirliğinde sistem, birbiri ile etkileşim halinde bulunan alt bileşenlerin oluşturduğu bir yapıdır. Her sistemde bütünü oluşturan alt bileşenler birbirlerini etkileyerek sistemin genel durumu üzerinde de etkili olurlar. Sistemde oluşan herhangi bir durumu anlayabilmek için onu oluşturan alt sistemleri veya bileşenleri ve bu sistemlerin birbirleriyle olan ilişkilerini de göz önünde bulundurmak gerekir (Özkılıç, 2016).

Sistem güvenilirliğinin hesaplanmasındaki bir diğer önemli öğelerinden biri ise sistem bileşenlerinin bozulmaları sonrası tamir edilebilir olup olmamasıdır. Çünkü güvenilirlik hesaplamalarındaki bozulma sürelerinin dağılımı bu özelliğe göre değişkenlik gösterebilir. Sistem bileşenleri bozulma sonrası tamir edilebilir

özellikte ise, bozulmalar arası süreler (Time Between Failure) dikkate alınırken, bozulma sonrası tamir edilemez özellikte ise bozulmaların gerçekleşme süreleri (Time To Failure) dikkate alınarak dağılım bulunur (Uzuner, 2015).

Sistemlerdeki süreç iyileştirmelerinde sistemin herhangi bir bölümünü geliştirmeye başlamadan önce genel sistem yapısı üzerinde etkili olabilecek alt sistemleri veya bileşenleri tanımlamak gerekir. Örneğin;

- Sistem performansının çok iyi olması, bu sistemin parçalarının da her birinin performansının iyi olduğu anlamına gelmemelidir.
- Sistemin performansını kısıtlayan faktörlerin belirlenmesi ve aralarında sebep-sonuç ilişkisi kurulması gerekir.
- Sistemi daha küçük parçalara ayırıp iyileştirerek sistemin genel performansı üzerinde olumlu bir etki yaratmak mümkündür.
- Sistemde zayıf olan bileşenin dışındaki herhangi bir başka bileşeni güçlendirmek için uğraşmanın, sistemin bütününe geliştirmeye bir katkı sağladığı söylenemez.
- Sistemin her aşamadaki başarısını en yüksek seviyede tutmak, sistemin genel performansının da en üst düzeyde olmasını sağlar (Özkılıç, 2016).

3.1 Tamir Edilebilir ve Tamir Edilemez Sistem Türleri

Güvenilirlik literatüründe sistemler genel olarak tamir edilebilir ve tamir edilemez sistemler olarak iki guruba ayrılır. Tamir edilemez sistemlerin bozulduktan sonra geri dönüşümleri yoktur. Bu yüzden sadece bir kez bozulurlar ve böyle sistemlerin bozulma zamanlarının dağılımı örnek olarak Weibull gibi bir yaşam modeli ile modellenenebilir. Tamir edilebilir sistemler ise bozulmadan sonra herhangi bir müdahale ile tekrar geri kazanılan sistemlerdir ve bozulma zamanlarının modellenmesi ile ele alınırlar (Bugatekin, 2017).

Tamir edilemez, sürekli devrede olmayan sistemler belirli bir göreve koşullanmış sistemlerdir. Bu sistemlerde, sistemi oluşturan herhangi bir bileşende arıza olması durumunda bile sistemin çalışmaya devam etmesi gerekir. Sistemde bu bileşen arızalı olarak kalmaya devam edebilir ya da tamir edilebiliyorsa tamir edilip gerekli durumlarda yenisi ile değiştirilebilir (Dündar, 2002).

Tamir edilebilir, sürekli işletimde olan sistemlerde ise sistemin arıza durumunun, çok sık ve uzun olmadığı sürece kabul edilebilir olması gerekir. Arıza durumunda arızalı sistem onarılabilir, yenisi ile değiştirilebilir ya da arıza giderilene kadar benzer bir bileşen bu arıza durumundaki bileşenin yerine geçebilir. Kısacası bu tarz sistemlerde çalışma, arıza ve onarım şeklinde bir döngüden bahsedilir (Ersalıcı, 2012).

3.2 Seri Sistemler

Seri yapıli sistemlerde bileşenler birbiri ardına sıralanır. Sistemin başarılı olabilmesi için tüm bileşenlerin işletimde olması gerekmektedir. Yalnızca bir tane bileşenin bile arızalanması durumunda sistemin çalışması söz konusu olamaz (Özdoğan ve Uzun, 2011).



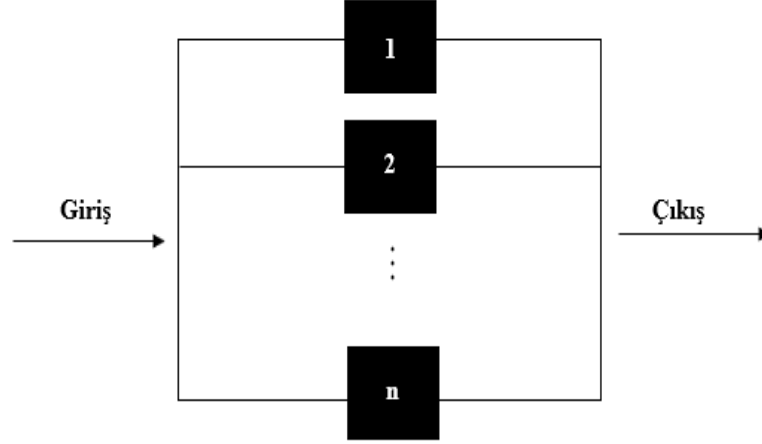
Şekil 3. 1. Seri bağıli sistem yapısı

Burada n elemanlı bir sistem için bileşenlerin güvenilirlikleri sırasıyla $R_1, R_2, \dots, R_n$ olsun. Bu durumda seri bağıli sistem için sistem güvenilirliği aşağıda verilen denklemdeki gibi olur ve R_i i'nci parçanın başarılı çalışması olasılığı olarak adlandırılır ise;

$$R_{SERİ} = R_1 R_2 \dots R_n = \prod_{i=1}^n R_i \quad (3.1)$$

3.3 Paralel Sistemler

Paralel sistemlerde bileşenler birbirlerine paralel olacak şekilde sıralanırlar. Bu tür sistemlerin çalışması için bir tane bileşenin çalışıyor durumda olabilmesi yeterli olacaktır.



Şekil 3. 2. Paralel bağlı sistem yapısı

Güvenilirlik açısından paralel bir sistemi oluşturan elemanların güvenilmezliklerinin çarpımı ise sistem güvenilmezliğini verir.

$$Q_{PARALEL} = Q_1 Q_2 \dots Q_n = \prod_{i=1}^n Q_i \quad (3.2)$$

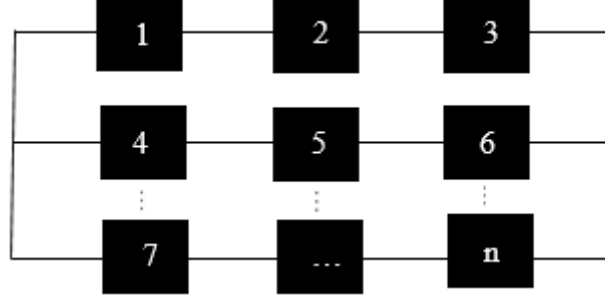
Sistem güvenilirliğine ait ifadesi ise aşağıdaki denklemdeki gibidir.

$$R_{PARALEL} = 1 - Q_{PARALEL} = 1 - (Q_1 Q_2 \dots Q_n) = 1 - [\prod_{i=1}^n Q_i] \quad (3.3)$$

Her iki sistem karşılaştırıldığında, seri sistemlerdeki güvenilirlik bu sistemdeki en zayıf olan tek bir bileşenin güvenilirliğinden bile daha küçük iken, paralel sistemin güvenilirliği ise en yüksek olan bileşenden bile daha yüksektir (Özdoğan ve Uzun, 2011).

3.4 Karma Sistemler

Karma sistemler, seri ve paralel sistemlerin bir arada bulunduğu yapılardır. Bu tür sistemlerde güvenilirlik hesaplamaları yaparken seri ve paralel sistemlerin güvenilirlik hesaplamalarından yararlanılır (Uzuner, 2015).



Şekil 3. 3. Karma sistem yapısı

3.5 n 'den k çıkışlı F ve G Sistemleri ($kn/: F - G$) ve Güvenilirlikleri

n 'den k çıkışlı F ve G sistemler temelde seri ve paralel olarak bilinen sistem türlerinin özel bir durumu olarak ele alınan daha karmaşık sistem yapılarıdır. n tane bağımsız ikili elemandan oluşan ve bu bileşenlerden en az k tanesi çalışır durumda görevini gerçekleştirebilen sistemlere n 'den k çıkışlı G sistemi denirken, en az k ya da daha fazla bileşenin başarısız olması durumunda görevini gerçekleştiremeyen sistemlere de n 'den k çıkışlı F sistemi denir. Her iki sistemin tanımlarına göre, bir n 'den k çıkışlı G sistemi, n 'den $(n - k + 1)$ çıkışlı F sisteminin eşdeğeri olurken, n 'den k çıkışlı F sistemi de n 'den $(n - k + 1)$ çıkışlı G sistemine eşdeğerdir. Bu, sistemlerin aynı bileşen güvenilirliklerine sahip olması durumunda ise, bir n 'den k çıkışlı G sisteminin güvenilirliğinin bir n 'den $(n - k + 1)$ çıkışlı F sisteminin güvenilirliğine eşit olduğu söylenebilir (Kuo ve Zuo, 2003; Levitin, 2005).

Seri sistemler n 'den k çıkışlı sistemlerde tüm unsurlarıyla çalışabilirse görevini gerçekleştireceği için n 'den n çıkışlı sistemlere karşılık gelirken, paralel sistemlerde en az bir tane bileşen başarıyla görevini gerçekleştirebilen sistem yapısına sahip olduğundan n 'den 1 çıkışlı sisteme karşılık gelecektir. Paralel sistemlerde, sistemin tamamen çalışabilmesi için en az bir bileşene ihtiyaç olduğundan geriye kalan $n - 1$ tane bileşenin gereksiz olduğu düşünülebilir. Ancak bu gereksiz bileşenlerde en az bir bileşenin başarı olasılığını arttırmak için sisteme dahil edilirler. Buna göre seri ve paralel sistemlerin, sistemin genel yapısı hakkında bilgi verdiği yapı fonksiyonları aşağıdaki gibidir (Demiralp, 2020; Levitin, 2005).

Seri Sistem:

$$\Phi(x) = \prod_{i=1}^n x_i = \min\{x_1, x_2, \dots, x_n\} \quad (3.4)$$

Paralel Sistem:

$$\Phi(x) = 1 - \prod_{i=1}^n (1 - x_i) = \max\{x_1, x_2, \dots, x_n\} \quad (3.5)$$

Burada x_i değişkeni $1 \leq i \leq n$ için i . bileşenin durumunu göstermek üzere aşağıdaki gibidir:

$$X_i = \begin{cases} 1, & i. \text{ bileşen başarılı ise} \\ 0, & i. \text{ bileşen başarısız ise} \end{cases}$$

Burada, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ durum vektörü, sistemi oluşturan bütün bileşenlerin durumlarını temsil etmektedir. Sistemin durumu, Φ ile gösterilmek üzere;

$$\Phi = \begin{cases} 1, & \text{Sistem başarılı} \\ 0, & \text{Sistem başarısız} \end{cases}$$

şeklinde ifade edilmektedir. Buna göre bütün bileşenlerin durumları biliniyorsa, sistemin genel durumu hakkında bilgi sahibi olunabilir. Ayrıca sistem durumu, bileşenlerin durumunun deterministik bir fonksiyonu olarak aşağıdaki gibi yazılır ve $\Phi(\mathbf{x})$ 'e, sistemin yapı fonksiyonu adı verilir (Kuo and Zuo, 2003).

$$\Phi(\mathbf{x}) = \Phi(x_1, x_2, \dots, x_n)$$

Sistemlerin belirlenen süre içinde yapması beklenen görevleri yerine getirip getiremediğini ölçme işleminde güvenilirlik kullanılmaktadır. Yukarıda verilen i . bileşenin durumunu gösteren x_i değişkeni için $p_i = P(x_i = 1) = E(x_i)$ ifadesi i . bileşenin güvenilirliğini gösterir ve $E(\cdot)$, bir rassal değişkenin beklenen değeridir.

$q_i = 1 - p_i$ ise i . bileşenin güvenilmezliğini gösterir. O zaman $\mathbf{x}$ ve $\mathbf{p} = (p_1, p_2, \dots, p_n)$ vektörleri için Φ sisteminin sırasıyla güvenilirliği ve güvenilmezliği aşağıdaki gibi tanımlanır (Kuo and Zuo, 2003).

$$R = P(\Phi(x) = 1) = E(\Phi(x)) \quad (3.6)$$

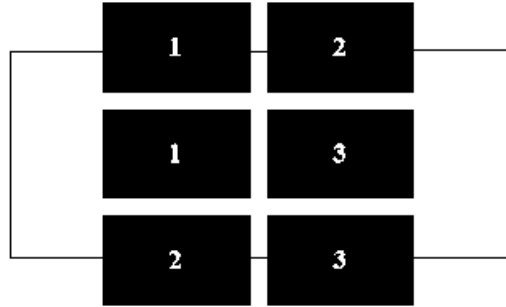
$$Q = P(\Phi(x) = 0) = 1 - R \quad (3.7)$$

n 'den k -çıkışlı: F ve G sistemlerin yapı fonksiyonları ise sırasıyla aşağıdaki gibi ifade edilir (Gürsoy, 2006; Demiralp, 2020).

$$\Phi(x) = \begin{cases} 1, \sum_{i=1}^n x_i > n - k \\ 0, \sum_{i=1}^n x_i \leq n - k \end{cases} \quad (3.8)$$

$$\Phi(x) = \begin{cases} 1, \sum_{i=1}^n x_i \geq k \\ 0, \sum_{i=1}^n x_i < k \end{cases} \quad (3.9)$$

n 'den k çıkışlı sistemler, birçok farklı teknik uygulamalarda karşımıza çıkabilmektedir. Basit bir örnek ele alacak olursak bir elektrik üretim sistemindeki jeneratörlerden en az üç bileşeninden yalnızca iki bileşeninin çalıştığı bir durumda bu sistemden beklenen görevin başarıyla sonuçlanması gerekir. Böyle bir sistemi açıklamak için aşağıdaki yapı örnek olarak gösterilebilir.



Şekil 3. 4. 3'ten 2 çıkışlı G sistemi yapısı örneği

$\Phi(x)$ yapı fonksiyonu ile tanımlanan bir sistemin güvenilirliği aşağıdaki gibi gösterilir.

$$R = E(\Phi(x)) = P\{\Phi(x) = 1\} \quad (3.10)$$

Burada E beklenen değeri ifade eder. Sistemi oluşturan elemanların birbirinden bağımsız olduğu varsayımı altında i . bileşenin güvenilirliği aşağıdaki gibidir:

$$E(x_i) = P\{x_i = 1\} = p \quad i = 1, 2, \dots, n \quad (3.11)$$

Bu durumda n bileşenden oluşan seri ve paralel sistemlerin güvenilirlikleri sırasıyla aşağıdaki gibidir:

$$R_S(p) = p^n \quad (3.12)$$

$$R_P(p) = 1 - (1 - p)^n \quad (3.13)$$

n 'den k çıkışlı bir sistemin güvenilirliği ise aşağıdaki gibidir:

$$R_{n:k}(p) = P\left\{\sum_{i=1}^n x_i \geq k\right\} = \sum_{j=k}^n \binom{n}{j} p^j (1-p)^{n-j} \quad (3.14)$$

Buradaki sonuçlar sistemi oluşturan elemanların birbirinden bağımsız ve aynı güvenilirliğe sahip olduğu durumlar için geçerlidir (Gürsoy, 2006).

Örnek: Şekil 3.4'te gösterilen 3'ten 2 çıkışlı bir jeneratör sisteminin güvenilirliğini hesaplanırsa aşağıdaki gibi olur:

Güvenilirlik fonksiyonu:

$$\begin{aligned} R_{3:2}(p) &= P\left\{\sum_{i=1}^3 x_i \geq 2\right\} \\ &= P\{X_1 = 1, X_2 = 1, X_3 = 0\} + P\{X_1 = 1, X_2 = 0, X_3 = 1\} \\ &\quad + P\{X_1 = 0, X_2 = 1, X_3 = 1\} + P\{X_1 = 1, X_2 = 1, X_3 = 1\} \\ &= P\{X_1 = 1\}P\{X_2 = 1\}P\{X_3 = 0\} + P\{X_1 = 1\}P\{X_2 = 0\}P\{X_3 = 1\} \\ &\quad + P\{X_1 = 0\}P\{X_2 = 1\}P\{X_3 = 1\} \\ &\quad + P\{X_1 = 1\}P\{X_2 = 1\}P\{X_3 = 1\} \\ &= pp(1-p) + p(1-p)p + (1-p)pp + ppp = 3p^2(1-p) + p^3 \end{aligned}$$

Sistemi oluşturan bileşenlerin birbirinden bağımsız ve farklı güvenilirlikte olduğu düşünülürse seri ve paralel sistemlerin yapı fonksiyonları da aşağıdaki gibi gösterilir.

$$E(x_i) = P\{x_i = 1\} = p_i \quad , \quad i = 1, 2, \dots, n$$

$$R_S(p_1, p_2, \dots, p_n) = p_1 \cdot p_2 \dots p_n = \prod_{i=1}^n p_i \quad (3.15)$$

$$\begin{aligned} R_P(p_1, p_2, \dots, p_n) &= 1 - [(1 - p_1)(1 - p_2) \dots (1 - p_n)] \\ &= 1 - \prod_{i=1}^n (1 - p_i) \end{aligned} \quad (3.16)$$

3.5.1 n 'den k çıkışlı Tamir Edilemez Sistemler

Herhangi bir sistemin güvenilirliği ve diğer performans ölçütleri zamanın işlevleridir. Gerçekte, görevi tamamlanana kadar bir sistemi onarmak bazen imkansızdır. Bu durumda sistemin güvenilirliği, zamanın azalan bir fonksiyonu şeklinde ifade edilir.

3.5.1.1 Bağımsız aynı tipte bileşenlere sahip olan sistemler

Bir n 'den k çıkışlı G sistemindeki bileşenler bağımsız ve özdeş dağılımlı olduğunda sistemin güvenilirlik işlevi şu şekilde ifade edilebilir:

$$R_{n:k:G}(t) = \sum_{j=k}^n \binom{n}{j} R(t)^j F(t)^{n-j} \quad (3.17)$$

Bu denklem, Denklem (3.14) 'te, p 'nin $R(t)$ ve $(1 - p)$ 'nin $F(t)$ ile değiştirilmesiyle oluşur. Benzer şekilde sistemin ömrünün dağılım fonksiyonu (CDF);

$$F_{n:k:G}(t) = 1 - R_S(t) = \sum_{j=0}^{k-1} \binom{n}{j} R(t)^j F(t)^{n-j} \quad (3.18)$$

Sistemin ömrünün olasılık yoğunluk fonksiyonu (PDF);

$$f_{n:k:G}(t) = \frac{d F_{n:k:G}(t)}{d t} = k \binom{n}{k} f(t) F(t)^{n-k} R(t)^{k-1} \quad (3.19)$$

Genellikle sistem kullanıldıkça sistemdeki bileşenler tek tek arızalanmaya başlayacaktır. $(n - k + 1)$ 'nci bileşen başarısız olduğunda sistem de başarısız

olacaktır. t_i , i 'nci bileşenin ömrünü belirtmek için kullanılırsa, sistem ömrü bu durumda $(n - k + 1)$ 'nci en küçük t_i 'ye eşit olur. Sistemin beklenen ömrü veya ortalama arızaya kadar geçen süre (Mean Time to Failure) ise aşağıdaki gibi gösterilir (Kaya, 2015).

$$MTTF_{n:k:G} = \int_0^{\infty} t f_s(t) dt = \int_0^{\infty} R_s(t) dt \quad (3.20)$$

3.5.1.2 Bağımsız farklı tipte bileşenlere sahip olan sistemler

Bileşenler aynı ömür dağılımlarına sahip olmadığında, n 'den k çıkışlı sistem güvenilirliği için bir ifade yazmak genellikle zordur. Basit durumlar için ise istenen ifadeleri türetmek mümkündür. Aşağıda n 'den k çıkışlı F sistemin yapı ve güvenilirlik fonksiyonu verilerek örneklendirilmiştir.

Örnek: 3'ten 2 çıkışlı bir F sistemi ele alalım. Sistemin yapı ve güvenilirlik fonksiyonu aşağıdaki gibidir.

$$\Phi(x) = (X_1 \cup X_2)(X_1 \cup X_3)(X_2 \cup X_3) \quad ; \quad 1 \leq i \leq n$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_1 + X_3 - X_1X_3)(X_2 + X_3 - X_2X_3)$$

$$R(p) = (p_1 + p_2 - p_1p_2)(p_1 + p_3 - p_1p_3)(p_2 + p_3 - p_2p_3)$$

Bileşen güvenilirlikleri sırasıyla $p_1 = 0.82, p_2 = 0.93, p_3 = 0.87$ iken sistem güvenilirliği aşağıdaki gibi hesaplanır.

$$R(p) = (0.82 + 0.93 - (0.82)(0.93))(0.82 + 0.87 - (0.82)(0.87))$$

$$(0.93 + 0.87 - (0.93)(0.87)) = 0.9555 \cong 0.96$$

3.5.2 n 'den k çıkışlı Tamir Edilebilir Sistemler

Bir n 'den k çıkışlı G sistemi çalıştırıldığında tüm n bileşen iyi durumdadır. Sistem kullanıldıkça bileşenler birbirini ardına arızalanacaktır. Çalışan bileşenlerin sayısı k 'nın altına düştüğünde veya arızalı bileşenlerin sayısı $n - k + 1$ 'e ulaştığında sistem başarısız olur. Arızalı bileşenleri onarmak için yeterli kaynak ayrılırsa, arızalı bileşenlerin sayısını çok daha uzun bir süre $n - k + 1$ 'in altında tutarak sistemin yaşam döngüsünü uzatmayı başarabiliriz. Herhangi bir anda arızalı bileşenlerin sayısı $n-k$ 'dan fazla olduğunda, sistem arızalanır ve yaşam döngüsü

sonlanmış olur. Birden fazla tamirci veya onarım tesisi olduğunda durumda birden fazla arızalı bileşenin aynı anda onarılabilirdiği birçok durum mümkündür. Onarım tesislerinin sayısını arttırdıkça, sistemin çalışma süresini de ilk arızasına kadar uzatabilme imkânı sağlanmış olur (Kuo and Zuo, 2003).

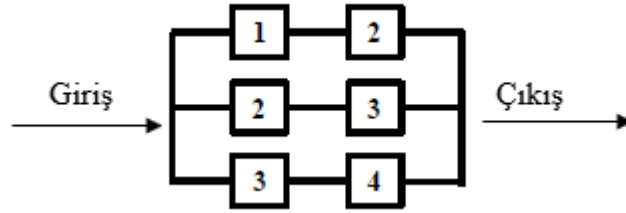
3.6 Doğrusal Ardışık n 'den k çıkışlı F ve G Sistemleri ($DA/kn/: F - G$)

Ardışık sistemler normal bir n 'den k çıkışlı sistemlerin şartlı durumları olarak ele alınırlar. Ardışık sistemlerde sadece ardışık bileşenlerin çalışmasına ya da arızalanmasına dayalı bir hesaplama yapılır (Kaya, 2015).

Doğrusal bir yol bozunca düzenlenmiş n tane bileşenden oluşurlar. n bileşenden ardışık olarak en az k ya da daha fazla bileşen başarısız olduğunda görevini gerçekleştiremeyen sistemlere doğrusal ardışık n 'den k çıkışlı F sistemi denirken, en az k ya da daha fazla bileşenin başarılı olduğunda görevini gerçekleştirebildiği sistemlere de doğrusal ardışık n 'den k çıkışlı G sistemleri denir.

3.6.1 Doğrusal Ardışık n 'den k çıkışlı G Sistemleri ($DA/kn/: G$)

Ardışık olarak en az k ya da daha fazla bileşen başarılı olursa sistemde başarılı olacaktır. Buna göre Şekil 3.5'te ardışık olarak dört bileşenden iki tanesinin başarıyla çalıştığında sisteminde başarılı olduğu bir G sistemi yapısı verilmiştir.



Şekil 3. 5. Doğrusal ardışık 4'ten 2 çıkışlı G sistemi örneği

Bu sistemin $\Phi(x)$ yapı fonksiyonu ve bu fonksiyondan elde edilen sistem güvenilirliği sırasıyla aşağıdaki gibidir:

$$\Phi(x) = (X_1 \cup X_2)(X_2 \cup X_3)(X_3 \cup X_4) \quad ; \quad 1 \leq i \leq n$$

$$\Phi(x) = 1 - (1 - X_1X_2)(1 - X_2X_3)(1 - X_3X_4)$$

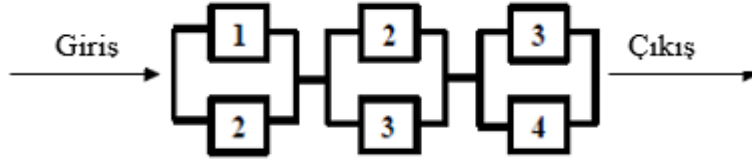
$$R(p) = 1 - (1 - p_1p_2)(1 - p_2p_3)(1 - p_3p_4)$$

Örneğin bileşen güvenilirlikleri sırasıyla $p_1 = 0.82, p_2 = 0.93, p_3 = 0.87, p_4 = 0.76$ iken sistem güvenilirliği aşağıdaki gibi olacaktır:

$$R(p) = 1 - (1 - (0.82)(0.93))(1 - (0.93)(0.87))(1 - (0.87)(0.76)) \\ = 0.9846$$

3.6.2 Doğrusal Ardışık n 'den k çıkışlı F Sistemleri ($DA/kn/: F$)

Ardışık olarak en az k ya da daha fazla bileşen başarısız olursa sistemde başarısız olacaktır. Buna göre Şekil 3.6'da ardışık olarak dört bileşenden iki tanesinin arızalı olduğunda sisteminde başarısız olduğu bir F sistemi yapısı verilmiştir.



Şekil 3. 6. Doğrusal ardışık 4'ten 2 çıkışlı F sistemi örneği

Bu sistemin $\Phi(x)$ yapı fonksiyonu ve bu fonksiyondan elde edilen sistem güvenilirliği sırasıyla aşağıdaki gibidir:

$$\Phi(x) = (X_1 \cup X_2)(X_2 \cup X_3)(X_3 \cup X_4) \quad ; \quad 1 \leq i \leq n$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_2 + X_3 - X_2X_3)(X_3 + X_4 - X_3X_4)$$

$$R(p) = (p_1 + p_2 - p_1p_2)(p_2 + p_3 - p_2p_3)(p_3 + p_4 - p_3p_4)$$

Örneğin bileşen güvenilirlikleri sırasıyla $p_1 = 0.82, p_2 = 0.93, p_3 = 0.87, p_4 = 0.76$ iken sistem güvenilirliği aşağıdaki gibi olacaktır.

$$R(p) = (0.82 + 0.93 - (0.82)(0.93))(0.93 + 0.87 - (0.93)(0.87)) \\ (0.87 + 0.76 - (0.87)(0.76)) = 0.9478 \cong 0.95$$

Doğrusal ardışık n 'den k çıkışlı F sistemleri $p_i, i = 1, 2, \dots, n$ sistemdeki her bir i 'nci bileşenin güvenilirliklerini temsil ederken, $p_{n+1} \equiv 1$ ve $k \leq n \leq 2k$ olması durumunda sistem güvenilirliği aşağıdaki şekilde hesaplanır (Zuo and Kuo, 1990).

$$R_{F;k;n;p_i} = 1 - \sum_{i=1}^{n-k+1} \left(p_{i+k} \prod_{j=i}^{i+k-1} (1 - p_j) \right) \quad (3.21)$$

Bileşen güvenilirlikleri sırasıyla $p_1 = 0.82, p_2 = 0.93, p_3 = 0.87, p_4 = 0.76$ iken Denklem (3.21) kullanılarak sistem güvenilirliği aşağıdaki gibi hesaplanır (Gökdere ve Gural, 2018).

$$\begin{aligned} R(p) &= 1 - (p_3(1 - p_1)(1 - p_2) + p_4(1 - p_2)(1 - p_3) + (1 - p_3)(1 - p_4)) \\ &= 1 - (0.87(1 - 0.82)(1 - 0.93) + 0.76(1 - 0.93)(1 - 0.87) \\ &\quad + (1 - 0.87)(1 - 0.76)) = 0.95 \end{aligned}$$

Doğrusal ardışık n 'den k çıkışlı F ve G sistemleri birbirlerinin duali olduğundan örnekteki F sistemini kullanılarak G sisteminin de güvenilirliğini hesaplayabiliriz (Kuo et al., 1990).

Doğrusal ardışık n 'den k çıkışlı G sisteminin güvenilirliği $R_{G;k;n;p_i} = 1 - R_{F;k;n;1-p_i}$ şeklinde ifade edilir.

$$\begin{aligned} R(p) &= (p_1 p_2 (1 - p_3) + p_2 p_3 (1 - p_4) + p_3 p_4) \\ &= (0.82)(0.93)(1 - 0.87) + (0.93)(0.87)(1 - 0.76) + (0.87)(0.76) = 0.95 \end{aligned}$$

4. SİSTEM GÜVENİLİRLİĞİNDE OPTİMİZASYON

Optimizasyon hemen her yerde kullanılan bir kavram olup, herhangi bir sistemde istenen işlevin performansını maksimum, maliyetini minimum yapmayı hedefler. Kısacası ilgilenilen problemin amacına göre alternatif çözüm yolları arasından en uygun olanın seçilmesi işlemidir.

Optimizasyon için birçok metot kullanılabilmektedir. Eğer kullanılan metot, belirli bir problemde her defa aynı sonucu veriyorsa, bu tür yöntemlere deterministik yöntemler denir. Bu yöntemler, genellikle en iyi tek çözümü bulmak için kullanılırlar. Deterministik yöntemler, bazı problemlerin çözümü için her zaman mümkün olmayıp, çok zaman ya da maliyet gerektirdiği için bu tür problemlerde genellikle deterministik olmayan yöntemlerin kullanılması tercih edilmektedir. Bu durumda olasılıksal modellerin kullanılması daha doğru olur. Örnek olarak markov zincirleri, kuyruk teorisi, karar analizi, simülasyon, tahmin modelleri, güvenilirlik analizi gibi yöntemler verilebilir (Akkoyunlu ve Engin, 2011).

Sistem güvenilirliği optimizasyonlarında ise optimizasyonun amacı, sistem güvenilirliğini maksimize etmeye çalışmak olabilir. Sistem üzerinde olumlu etkiye sahip olan faktörler problemin amacına göre optimizasyon yöntemleri kullanılarak iyileştirilebilirken, olumsuz faktörlerin de en aza indirilmesi veya ortadan kaldırılması beklenir. Bu yöntemler kullanılırken hedeflenen amaca en yakın çözümleri bulabilmek için son zamanlarda yapay zekanın alt dallarından biri olan meta-sezgisel algoritmaların kullanımı tercih edilmektedir. Gerçek hayattan esinlenerek oluşturulan bu algoritmalara Öğretme-Öğrenme Esaslı Optimizasyon algoritması (Teaching-Learning Based Optimization (TLBO)), Genetik Algoritmalar (GA), Karınca Koloni, Parçacık Sürü optimizasyon algoritmaları örnek olarak gösterilebilir.

4.1 Yapay Zekâ

Yapay zekâ, literatürde zeki sayısal sistemler yaklaşımı ile Bilgisayar Bilimlerinin alt bir dalı olarak kabul görmüş bir yapıdır. Bu kavram özünde insanların duygu, düşünce ve davranışlarını taklit etmeyi barındırmış ve daha sonra zaman içinde doğadaki her türlü oluşu mantıksal ve matematiksel birer kalıplara

sokmuştur. Yapay zekada optimizasyon kavramı oldukça önemli bir hale gelmiştir. Optimizasyon problemlerindeki tüm çözüm kümesi içerisinde hedefe en yakın çözüm çıktısını belirli kurallar ve kısıtlar dahilinde ve aynı zamanda tamamen gerçek hayattan esinlenerek tasarlanmış farklı sezgisel ve meta-sezgisel yöntemlerle sunmaktadır (Köse, 2017).

4.2 Meta-Sezgisel Algoritmalar

Günümüzde çoğu problemin çözümünde çeşitli optimizasyon yöntemleri kullanılır. Belli kısıtlar altında istenilen amacın elde edilmesi için optimizasyon problemlerinde matematiksel ve sezgisel yöntemler kullanılır. Matematiksel yöntemler, problemdeki tüm çözüm uzayını tarayarak hedefe ulaşırken, sezgisel yöntemler çözüm kümesine sezgisel olarak yaklaşıp en iyiye yakın bir sonuç bulmayı amaçlar. Sezgisel yöntemlerin etkili bir şekilde kullanılması sonucunda meta-sezgisel algoritmalar ortaya çıkmıştır. Matematiksel yöntemler ise bütün çözüm kümesini tarayarak zaman kısıtını ihlal edip maliyet artışına sebep olacağı için meta-sezgisel yöntemlerin kullanımı sıklıkla tercih edilir. Literatür incelemesi yaptığımızda bu tür algoritmalara öğretmen ve öğrenci ilişkilerinden ve öğrencilerin kendi aralarındaki etkileşimlerinden yararlanan Öğretme-Öğrenme Esaslı Optimizasyon Algoritması (TLBO), evrim teorisinden ilham alınarak tasarlanmış Genetik Algoritmalar (GA), Parçacık Sürü Optimizasyonu (PSO), Karınca Koloni Optimizasyonu (ACO), Benzetim Tavlama (SA), Harris şahinlerinin avcılık modelinden ve takım davranışlarından etkilenerek tasarlanmış Harris Hawks Optimizasyon Algoritması (HHO), Saten çardak kuşlarının birleşmesinden oluşturulan Satin Bowerbird Optimizasyon Algoritması (SBO), Kelebeklerin yiyecek bulma ve birleşme davranışlarını taklit eden Butterfly Optimizasyon Algoritması (BOA) gibi örnekler verilebilir (Çelik ve ark., 2019).

4.2.1 Öğretme-Öğrenme Esaslı Optimizasyon Algoritması (TLBO)

Öğretme-Öğrenme Esaslı Optimizasyon (TLBO) algoritmaları, ilk olarak 2011 yılında Rao ve arkadaşları tarafından öğretme ve öğrenme sürecinden ilham alınarak tasarlanmıştır. Bu algoritma, öğretmen-öğrenci ilişkilerini ve öğrencilerin kendi aralarındaki etkileşimlerini taklit ederek mevcut problemin iyileştirilmesi üzerine geliştirilen bir optimizasyon algoritmasıdır.

Öğretme-Öğrenme Esaslı Optimizasyon Algoritmasının çözüm uzayı bir sınıftaki öğretmen ve öğrencilerden oluşmaktadır. Bu algorithmada amaç ortalama öğrencilerin bilgi düzeylerini artırarak hedefe ulaşmaktır. Algorithmada öğretmen ve öğrenciler, öğretmen aşaması ve öğrenci aşaması olarak bilinen iki temel fazdan oluşurlar. TLBO algoritmasında çıktı, öğretmenin kalitesine bağlı olarak öğrencilerin sonuçları veya notları açısından değerlendirilir. Bu nedenle, bir öğretmen genellikle, öğrencilerin notları açısından daha iyi sonuçlar elde edebilmeleri için en bilgili kişi olarak kabul edilir. Ayrıca, öğrenciler kendi aralarındaki etkileşimden de öğrenirler ve bu da sonuçlarını iyileştirmeye yardımcı olur ve tüm popülasyonda en iyi çözüm öğretmen olarak kabul edilir (Kumar et al, 2021).

$$\text{Sınıf} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1D} \\ X_{21} & X_{22} & \dots & X_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{p_n1} & X_{p_n2} & \dots & X_{p_nD} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{bmatrix} \begin{matrix} 1. \text{öğrenci} \\ \vdots \\ n. \text{öğrenci} \end{matrix} \quad (4.1)$$

Algorithmaya göre öncelikli olarak başlangıç popülasyonumuzun olması gerekir. Bu popülasyon öğretmen ve öğrencilerden oluşur. Denklem (4.1)'de matristeki her satır öğrenci olarak adlandırılır ve R_i ; $1 \leq i \leq n$ problemin çözümüne ilişkin oluşturulan amaç fonksiyonundan elde edilmiş değerleri temsil eder. Aynı zamanda eğer problemin amacı, amaç fonksiyonunu maksimize etmeye çalışmaksa R_i matrisindeki en yüksek değer satırı yani öğrenci en iyi öğretmen M_{new} olarak kaydedilirken, problemin amacı, amaç fonksiyonunu minimize etmeye çalışmaksa R_i matrisindeki en düşük değer satırı en iyi öğretmen M_{new} olarak kaydedilip sonraki aşamalarda kullanılmak üzere muhafaza edilir. Ayrıca matristeki her bir sütunun yani tasarım değişkenlerinin p_i ortalama vektörü M_j oluşturularak algoritma içerisinde kullanılır. Tüm bu işlemler her iterasyonda yeniden hesaplanarak nihai sonuca ulaşana kadar devam eder.

Öğretmen Aşaması: Bu aşama öğrencilerin, öğretmenlerinden bilgi edindiği, algoritmanın ilk kısmını oluşturur (Kumar et al, 2021). Öğretmen aşamasında öğrenciler, öğretmenlerinden bilgiyi taklit ederek öğrenmektedirler. Burada öğretmenin amacı öğrencilere bilgi vermek ve sınıfının ortalama bilgi

düzeyini artırmaya çalışmaktır. Öğretmen sınıftaki en deneyimli ve en bilgili insandır dolayısıyla öğrenciler en fazla öğretmenin bilgi seviyesi kadar bilgiye erişebilirler.

$$Difference\ Mean = r_i(M_{new} - T_F M_j) \quad (4.2)$$

$$TF = round[1 + rand(0,1)\{2 - 1\}] \quad (4.3)$$

Denklem (4.2) mevcut ortalama ile yeni ortalama arasındaki farkı verir ve TF’de değiştirilecek ortalamanın değerine karar veren öğretim faktörüdür. Bu fark ortalamasına göre öğretmen aşamasında öğrencilerin ortalama bilgi düzeyleri aşağıdaki ifadeye göre her iterasyonda yeniden güncellenir.

$$p_{new,i} = p_{old,i} + Difference\ Mean \quad (4.4)$$

$$R(p_{new,i}) > R(p_{old,i}) \quad (4.5)$$

Eğer Denklem (4.5)’teki gibi ise güncelleme yap, değilse yapma.

Öğrenci Aşaması: Öğretmen aşamasından sonra, tüm en iyi fonksiyon sonuçları öğrenci aşamasında kullanılmak üzere kaydedilir. Bu kısımda, öğrenciler bilgiyi birbirleri ile tartışarak ve etkileşime girerek öğrenirler. Eğer bir öğrenci, diğer öğrencilerden daha bilgili ise bilgili olan öğrencinin yardımıyla yine diğer öğrencilerin ortalama bilgi düzeyi her iterasyonda güncellenerek istenilen amaca ulaşılması sağlanır.

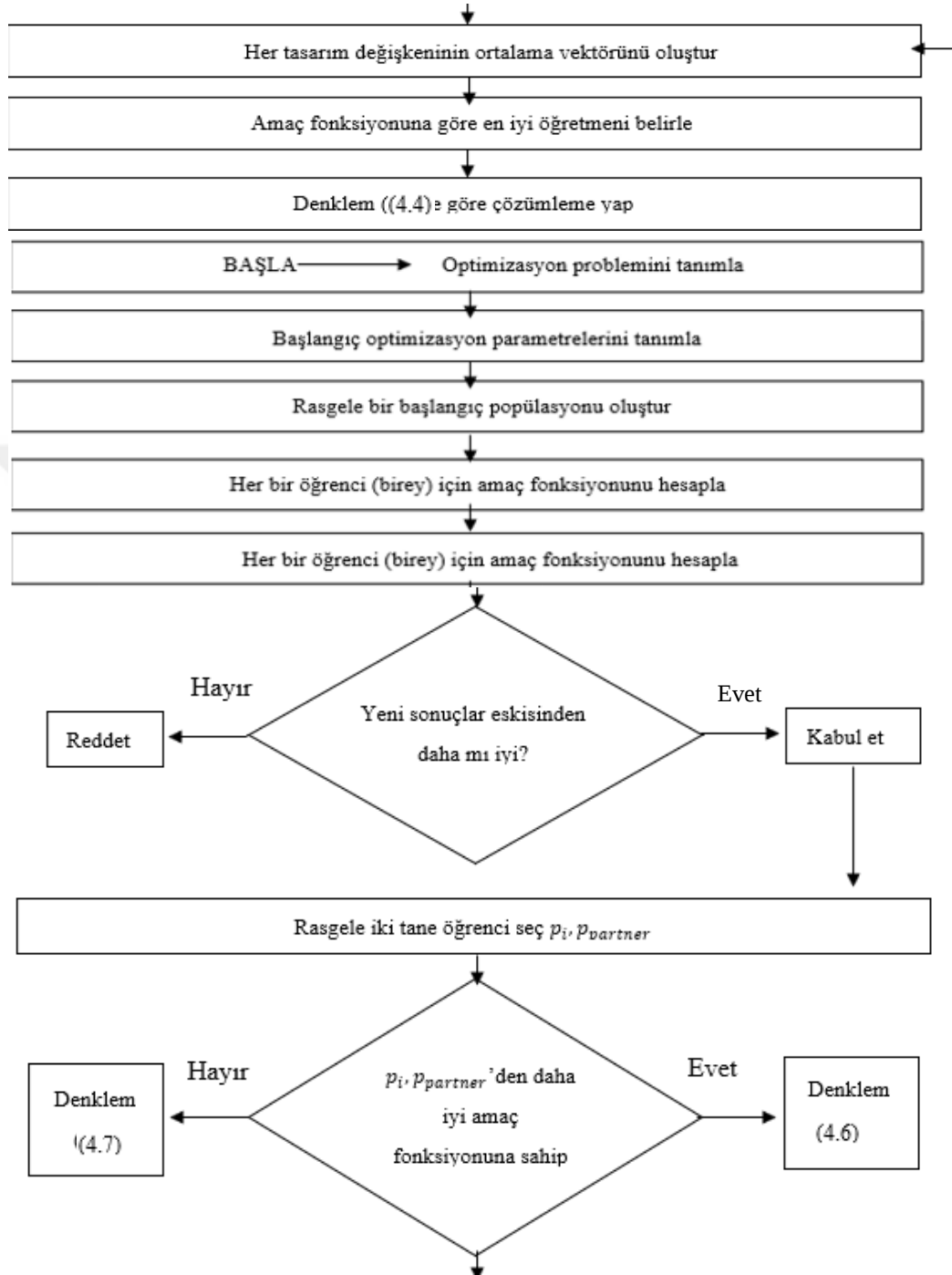
$$p_{new,i} = p_{old,i} + r_i(p_i - p_{partner}) \quad \text{Eğer } R(p_i) > R(p_{partner}) \quad (4.6)$$

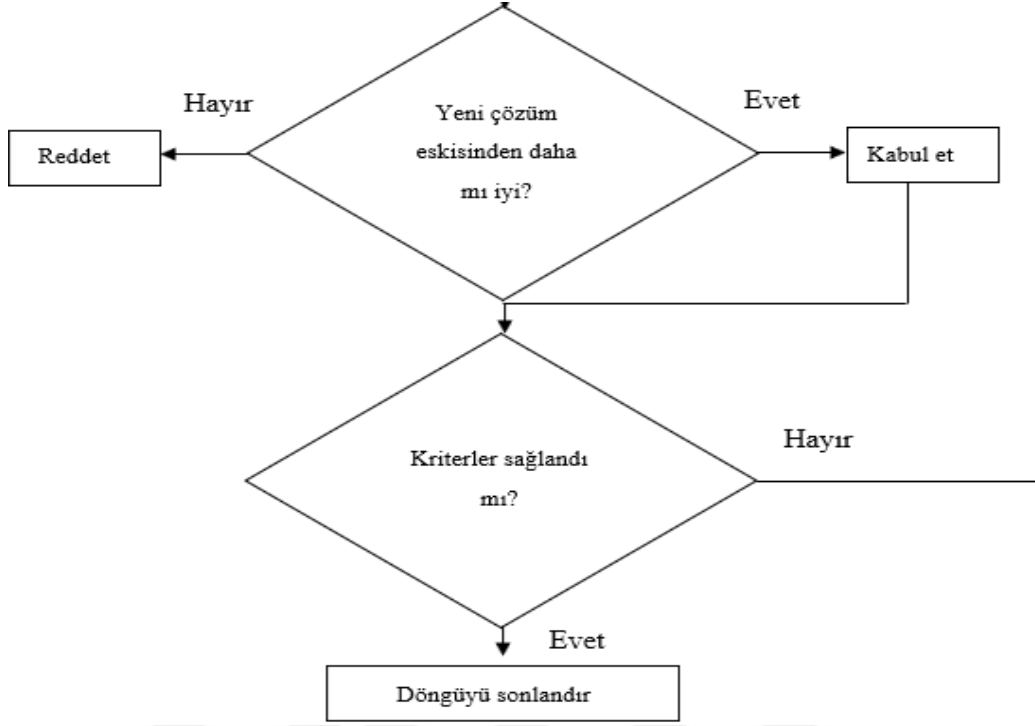
$$p_{new,i} = p_{old,i} - r_i(p_i - p_{partner}) \quad \text{Eğer } R(p_i) < R(p_{partner}) \quad (4.7)$$

$$R(p_{new,i}) > R(p_{old,i}) \quad (4.8)$$

Eğer Denklem (4.8)’deki gibi ise güncelleme yap, değilse yapma.

Öğretme-Öğrenme esaslı optimizasyon algoritması için işlem süreci aşağıdaki gibi ilerler.





Şekil 4. 1. TLBO akış diyagramı

4.2.1.1 Doğrusal ardışık 4'ten 2 çıkışlı F sisteminin TLBO algoritması ile sistem güvenilirliği

Doğrusal ardışık 4'ten 2 çıkışlı bir F sistemi ele alınsın. Sistemin güvenilirliği aşağıdaki gibidir.

$$\Phi(x) = (X_1 \cup X_2)(X_2 \cup X_3)(X_3 \cup X_4) \quad ; \quad 1 \leq i \leq n$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_2 + X_3 - X_2X_3)(X_3 + X_4 - X_3X_4)$$

$$R(p) = (p_1 + p_2 - p_1p_2)(p_2 + p_3 - p_2p_3)(p_3 + p_4 - p_3p_4)$$

Amaç $R(p)$ fonksiyonunun optimal güvenilirlik değerini ve bu optimal değeri veren bileşen güvenilirliklerini bulmaktır.

1.Adım: Başlangıç parametrelerini belirleyelim.

$$N=4, D(n)=4$$

$$\text{Alt Sınır}-\text{Üst Sınır} = \text{rand}(0,1)$$

$$\text{Amaç Fonksiyonu} = \max R(p)$$

$D(n)$ ile ifade edilen tasarım değişkenleridir. Değişkenler, 0 ile 1 arasında rasgele üretilmiş olan bileşen güvenilirliklerini temsil edecektir. Bu durumda

başlangıç popülasyonu 4x4'lük bir matristen oluşacaktır. Matrisin her satırı, öğrenciyi ifade eder. Matrisin her sütunu ise p_i , $1 \leq i \leq 4$ değişkenlerinden oluşacaktır.

$$\begin{bmatrix} p1 & p2 & p3 & p4 \\ 0,742494 & 0,370786 & 0,207092 & 0,951109 \\ 0,952521 & 0,523495 & 0,114473 & 0,003138 \\ 0,428517 & 0,068051 & 0,493814 & 0,434693 \\ 0,483397 & 0,707022 & 0,331470 & 0,426662 \end{bmatrix} = \begin{bmatrix} R(p) \\ 0,403623 \\ 0,066243 \\ 0,176259 \\ 0,420857 \end{bmatrix} \quad (4.9)$$

Her bir değişkenin ortalaması alınarak ortalama vektörü oluşturulur. $R(p)$, sistem güvenilirlik değeri en yüksek olan satır yani öğrenci, en iyi öğretmen olarak kaydedilir. Denklem (4.2) hesaplanır. Denklem (4.3)'teki TF değeri rasgele olarak seçilir.

M(pi)ort	0,651732	0,417339	0,286712	0,453900
Pteacher	0,483397	0,707022	0,331470	0,426662
ri	0,432786	0,752905	0,925760	0,851210
Di	-0,072853	0,218104	0,041435	-0,023186
TF	1			

Şekil 4. 2. Denklem çözümleri

2.Adım: Öğretmen aşamasındaki işlemler gerçekleştirilir. İşlemler öncelikle 1. öğrenci üzerinden başlatılır. Denklem (4.4) kullanılarak yeni bileşen değerleri hesaplanır. Burada işlemi gerçekleştirirken sınır kontrolü de yapılması gerekir.

$$\begin{array}{ccccc} p1 & p2 & p3 & p4 & R(p) \\ 0,669641 & 0,588890 & 0,248527 & 0,927923 & 0,564859 \end{array}$$

Hesaplanan yeni bileşen değerlerine baktığımızda alt ve üst sınırı ihlal eden herhangi bir değer yoktur. Aksi durumda alt ve üst sınırdan büyük değerler rasgele olarak tekrar üretilir ya da üst sınırdan büyük değerler üst sınır değerine eşit alınırken, alt sınırdan küçük değerlerde alt sınır değerine eşit alınarak hesaplama işlemleri gerçekleştirilir. Daha sonra Denklem (4.5)' i kullanarak matriste güncelleme yapılır. Güncelleme sonucunda aşağıdaki değerler elde edilmiştir. Öğretmen aşamasındaki sistem güvenilirlik değeri (0.564859), ilk çözüm

uzayındaki güvenilirlik değerinden (0.403623) daha yüksek olduğu için güncelleme sonucu da aşağıdaki gibi olmuştur.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,669641	0,588890	0,248527	0,927923	0,564859

3.Adım: Öğrenci aşamasındaki işlemler gerçekleştirilir. 1. öğrenci için öğretmen aşamasından sonra Denklem (4.6) uygulanır. Denklemdeki r_i değeri yine rasgele olarak 0-1 arasında üretilir. Sonuç aşağıdaki gibi olacaktır.

r_i	0,146719	0,678139	0,748355	0,647702
-------	----------	----------	----------	----------

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,705019	0,942091	0,064965	1,247389	1,144752

1. öğrenci için rasgele olarak 3. öğrenci partner olarak seçilmiş ve aralarında sistem güvenilirliği değerine göre kıyaslama yapılarak sonuçlar elde edilmiştir. Sınır kontrolü yapıldığında sınırları aşan değer (1.247389) görülmemektedir. Dolayısıyla üst sınırdan büyük olan bu değer için rasgele $[\text{rand}(0,1)]$ olarak tekrar üretilir ve hesaplama işlemleri gerçekleştirilir.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,705019	0,942091	0,064965	0,125228	0,169258

Sınır kontrolleri de sağlandıktan sonra Denklem (4.8)'i kullanarak yeni güncelleme yapılır. Öğrenci aşamasındaki sistem güvenilirlik değeri (0.169258), öğretmen aşamasındaki güvenilirlik değerinden (0.564859) daha düşük olduğu için güncelleme yapmaya gerek yoktur.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,669641	0,588890	0,248527	0,927923	0,564859

1. öğrenci için algoritma sonlanmıştır. 2. öğrenci içinde 2. adımdan itibaren işlemler gerçekleştirilir.

2.Adım: Öğretmen aşamasındaki işlemler gerçekleştirilir. İşlemler, 2. öğrenciden devam ederek başlatılır. Denklem (4.4)'ü kullanarak yeni bileşen değerleri hesaplanır. Burada işlemi gerçekleştirirken sınır kontrolü de yapılması gerekir.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,879668	0,450642	0,04162	-0,06971	-0,011141

Hesaplanan yeni bileşen değerlerine baktığımızda alt sınırı ihlal eden (-0.011141) değeri vardır. Bu yüzden değer rasgele [rand (0,1)] olarak tekrar üretilir ve hesaplama işlemleri gerçekleştirilir.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,879668	0,450642	0,04162	0,10055	0,061018

Daha sonra Denklem (4.5)' ü kullanarak matriste güncelleme yapılır. Güncelleme sonucunda aşağıdaki değerler elde edilmiştir. Öğretmen aşamasındaki sistem güvenilirlik değeri (0.061018), ilk çözüm uzayındaki güvenilirlik değerinden (0.066243) daha düşük olduğu için güncelleme sonucu da aşağıdaki gibi olmuştur.

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,952521	0,523495	0,114473	0,003138	0,066243

3.Adım: Öğrenci aşamasındaki işlemler gerçekleştirilir. 2. öğrenci için öğretmen aşamasından sonra Denklem (4.6) uygulanır. Denklemdeki r_i değerini yine rasgele olarak 0-1 arasında üretilir. Sonuç aşağıdaki gibi olacaktır.

$$r_i \quad 0,827457 \quad 0,003938 \quad 0,718825 \quad 0,005275$$

$p1$	$p2$	$p3$	$p4$	$R(p)$
0,564341	0,524218	0,270456	0,005372	0,142007

2. öğrenci için rasgele olarak 4. öğrenci partner olarak seçilmiş ve aralarında sistem güvenilirliği değerine göre kıyaslama yapılarak sonuçlar elde edilmiştir. Sınır kontrolü yapıldığında sınırları aşan değer görülmektedir. Sınır kontrolleri de sağlandıktan sonra Denklem (4.8) kullanılarak yeni güncelleme yapıp yapılmayacağına karar verilir. Öğrenci aşamasındaki sistem güvenilirlik değeri (0.142007), öğretmen aşamasındaki güvenilirlik değerinden (0.066243) daha büyük olduğu için güncelleme yapılır.

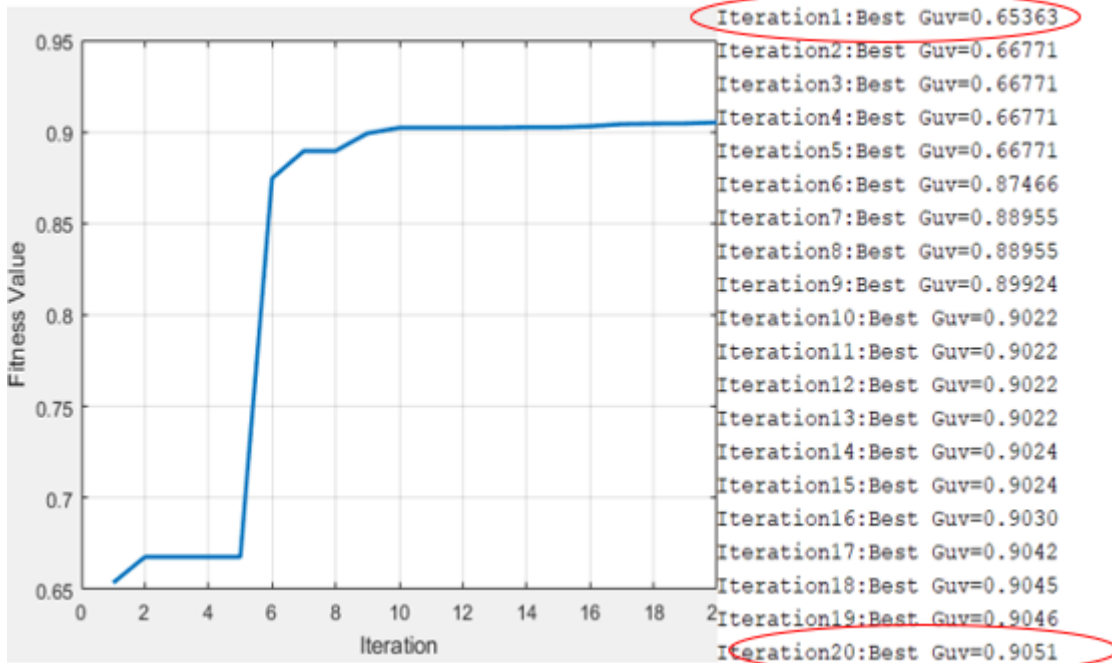
$p1$	$p2$	$p3$	$p4$	$R(p)$
0,564341	0,524218	0,270456	0,005372	0,142007

Tüm bu işlemler sırasıyla 3. ve 4. öğrenci için de yapıldığında son durumda (1. iterasyon sonucunda) matris aşağıdaki gibi bulunmuştur.

$$\begin{bmatrix} p1 & p2 & p3 & p4 \\ 0,669641 & 0,58889 & 0,248527 & 0,927923 \\ 0,564341 & 0,524218 & 0,270456 & 0,005372 \\ 0,355664 & 0,286155 & 0,535249 & 0,011634 \\ 0,410544 & 0,925126 & 0,372905 & 0,468097 \end{bmatrix} = \begin{bmatrix} R(p) \\ 0,564859 \\ 0,142007 \\ 0,195111 \\ 0,607122 \end{bmatrix} \quad (4.10)$$

Başlangıçta rasgele üretilmiş olan bileşen güvenilirlik matrisinde en yüksek sistem güvenilirliği değeri 0.420857 iken, 1. iterasyon sonunda sistem güvenilirlik değeri 0.607122 olarak bulunmuştur. Bu da 1. iterasyon sonundaki en iyi güvenilirlik değerini ifade eder. Ayrıca bu değeri veren bileşen güvenilirlik değerleri ise sırasıyla 0.410544, 0.925126, 0.372905 ve 0.468097 şeklinde bulunmuştur. İterasyon sayısı artırılarak, her iterasyon sonucundaki en iyi güvenilirlik değeri bulunabilir.

Bu örnek MATLAB R2016Bb programı ile de yapılmış, yol göstermesi açısından her iterasyon sonundaki en iyi güvenilirlik değeri grafiği sunulmuştur. MATLAB programında 1. iterasyon sonucunda en iyi güvenilirlik değeri 0.65363 olarak bulunurken iterasyon sayısı arttığında en iyi çözüme daha iyi ulaşarak 20. iterasyon sonunda 0.9051 sistem güvenilirlik değerine sahip olmuştur.



Şekil 4. 3. Matlab ile her iterasyon sonucundaki güvenilirlik değerleri

$$\begin{bmatrix} p1 & p2 & p3 & p4 \\ 0,742494 & 0,370786 & 0,207092 & 0,951109 \\ 0,952521 & 0,523495 & 0,114473 & 0,003138 \\ 0,428517 & 0,068051 & 0,493814 & 0,434693 \\ 0,483397 & 0,707022 & 0,331470 & 0,426662 \end{bmatrix} = \begin{bmatrix} R(p) \\ 0,403623 \\ 0,066243 \\ 0,176259 \\ 0,420857 \end{bmatrix} \quad (4.11)$$

Başlangıç popülasyonu manuel olarak hesaplanan örneklerle aynı alınmıştır. Her iterasyondaki en iyi sistem güvenilirlik değerleri Şekil 4.3'te gösterilmiştir. Denklem (4.12) ise 20. iterasyon sonundaki çözüm uzayı matrisi olup en yüksek sistem güvenilirlik değerini ve bu değeri veren bileşen güvenilirliklerini de göstermektedir.

$$\begin{bmatrix} p1 & p2 & p3 & p4 \\ 0,994990 & 0,808763 & 0,821855 & 0,641420 \\ 0,992794 & 0,809498 & 0,824382 & 0,641229 \\ 0,993171 & 0,809434 & 0,824236 & 0,641185 \\ 0,991060 & 0,810052 & 0,826257 & 0,641109 \end{bmatrix} = \begin{bmatrix} R(p) \\ 0,903363 \\ 0,904402 \\ 0,904372 \\ 0,905161 \end{bmatrix} \quad (4.12)$$

4.2.2 Genetik Algoritmalar (GA)

Genetik Algoritmalar ilk olarak 1975 yılında John Holland tarafından ortaya atılmıştır. Doğada gözlemlenen olaylar ve evrimsel teoriden ilham alınarak tasarlanmış bu algoritma, optimizasyon problemlerinde problemin amacına göre her iterasyonda en iyi sonucu üreterek bireylerin hayatta kalmasını sağlar. Kısacası iyi nesillerin yaşamlarını korurken, kötü nesillerin yok edilmesi ilkesine dayanır. Çözülme istenen problemin matematiksel olarak ifade edilmediği ve kesin çözümün olmadığı problemlerde muhtemel en iyi gerçek çözümü sunmak için kullanılarak avantaj sağlar.

Genetik Algoritma ile ilgili asıl gelişim John Holland'ın inşaat mühendisi olan doktora öğrencisinin (David E. Goldberg) "Gaz Boru Hatlarının Genetik Algoritma Kullanılarak Denetlenmesi" konulu doktora teziyle önemli bir boyut kazanmıştır. Daha sonra yapmış olduğu bu çalışmaya National Science Foundation tarafından genç araştırmacı ödülü verilmesi ile Goldberg "Makine Öğrenmesi,

Arama ve Optimizasyon için Genetik Algoritma” kitabını 1989 yılında yayınlarak bilime çok önemli bir katkı sağlamıştır (Ağalday, 2018).

Genetik Algoritmanın temelinde çözülmek istenen probleme göre bazı temel adımlar mevcuttur. Bunlar aşağıdaki gibi sıralanabilir: (Çakar, 2009)

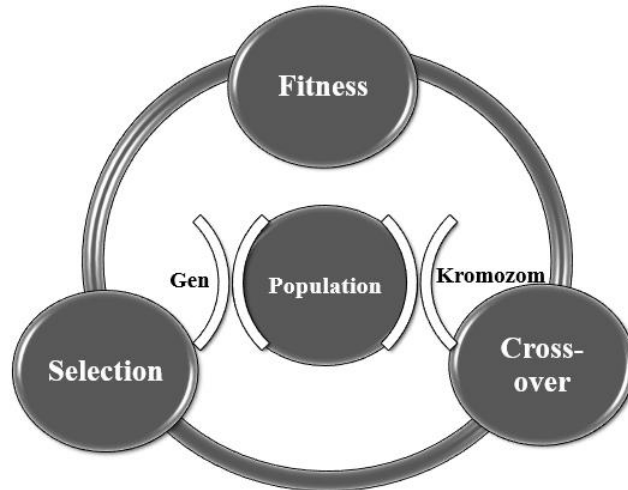
- Algoritmada kullanılacak çeşitli parametre değerleri.
- Problem için gerekli olan bir genetik gösterim.
- Problemin çözümüne başlamak için gerekli olan bir başlangıç popülasyonu.
- Çözümlerin amaca göre değerlendirileceği bir uygunluk fonksiyonu.
- Elde edilecek yeni ebeveynler üzerinde uygulanan genetik mekanizmalar.

Genetik Algoritmayı anlayabilmek için öncelikle aşağıdaki şekilde belirtilen bazı temel kavramlara hâkim olmak gerekir.



Şekil 4. 4. Genetik algoritma temel kavramlar

Çözülmek istenen probleme göre oluşturulmuş popülasyondaki her bir bileşen genleri ifade etmektedir. Genlerin bir araya gelerek oluşturduğu birtakım dizilere ise kromozom (birey) adı verilir. Genlerin ve kromozomların bir araya gelerek problemin çözümüne ilişkin oluşturduğu bütüne ise popülasyon adı verilmektedir.



Şekil 4. 5. Genetik algoritma genel döngüsü

Genetik Algoritmanın ilk aşamasında rasgele bir popülasyon oluşturularak uygunluk değerleri hesaplanır. Uygunluk değeri, problemin amacına göre oluşturulmuş bir uygunluk fonksiyonundan ya da amaç fonksiyonundan elde edilebilir. Daha sonra oluşturulan popülasyona temel genetik operatörler uygulanarak oluşmuş yeni nesil içinde uygunluk değerleri hesaplanır. Bu süreç gerekli sonlandırma kriteri sağlanana kadar devam eder (Taşkın ve Emel, 2009).

1. Başlangıç Popülasyonu: Alınması gereken popülasyon büyüklüğü problemlerin karmaşıklığına bağlıdır. Popülasyon büyüklüğünü çok büyük ya da çok küçük tutmak genel olarak algoritmanın performansının verimli çalışmasını engeller. Genetik algoritmada popülasyon büyüklüğünün çok büyük alınması istenilen amaca ulaşma etkinliğini artırırken daha fazla hesaplama maliyeti, hafıza ve zaman gerektireceğinden pratikte 100-300 arasında bir popülasyon büyüklüğünün seçilmesinin daha doğru olacağı literatürler arasında yerini alır (Ağalday, 2018; Çakar, 2009).
2. Uygunluk Değeri: Bu değer, problemin amacına göre oluşturulmuş bir uygunluk fonksiyonundan ya da problemin amaç fonksiyonundan elde edilebilir. Bu fonksiyon sayesinde popülasyondaki en iyi değer bir sonraki nesillere aktarımı sağlanır. Her iterasyonda hesaplanan bu yeni kromozomlar, eski kromozomların yerini alarak en iyi sonucuna ulaşılmasını sağlar (Ağalday, 2018):
3. Seçim İşlemi: Seçim işleminde ortama uygunluğu değerlendirilen popülasyondan yeni “iyi” olan ebeveynler (kromozomlar) seçilir. Bu ifadedeki iyi kelimesi problemin amacına göre değişkenlik gösterir (Satman, 2016). İlgilenilen problem amaç fonksiyonunu minimize etmekse eğer uygunluk fonksiyonundan elde edilen en düşük değerlere sahip kromozomlar seçilirken, fonksiyonunun maksimize edildiği bir durumda ise en yüksek uygunluk değerlerine sahip kromozomlar seçilerek sonraki adıma geçilir. Seçim işleminde çeşitli yöntemler vardır. Bu yöntemler aşağıdaki gibidir.
 - Rasgele Seçim: Bireyler tesadüfi olarak rasgele seçilirler.
 - Rulet Tekerleği Seçimi: Bu seçim yöntemi olasılığın klasik tanımına karşılık gelir. Bir kromozomun sonraki nesilde yer alma olasılığı, aynı kromozomun uygunluk değerinin popülasyondaki kromozomların toplam uygunluk değerine oranı şeklinde ifade edilir (Satman, 2016).

Örneğin, popülasyon büyüklüğünün 5 tane kromozom içerdiği ve her bir kromozomun uygunluk değerinin R_i olduğu bir rulet tekerleği seçim yönteminde kromozomların seçilme olasılığı her birinin uygunluk değerinin toplam uygunluk değerine oranıyla hesaplanabilmektedir.

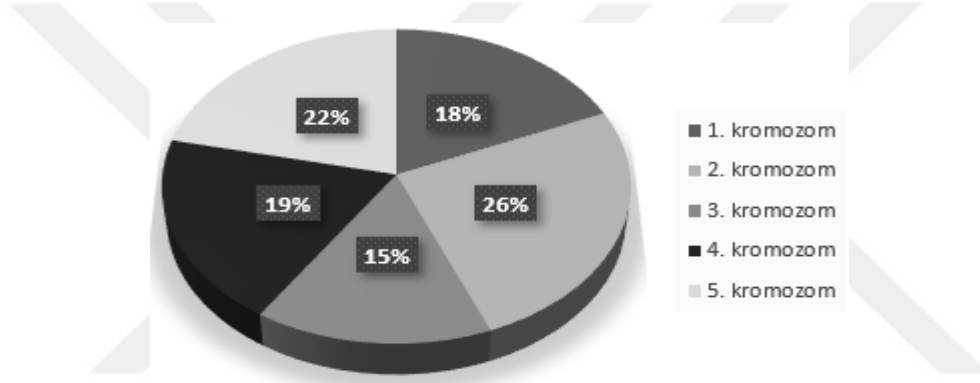
$$1. \text{ kromozom } R_1 = 0.18 \quad 0.18/0.98 = 0.1836$$

$$2. \text{ kromozom } R_2 = 0.25 \quad 0.25/0.98 = 0.2551$$

$$3. \text{ kromozom } R_3 = 0.15 \quad 0.15/0.98 = 0.1530$$

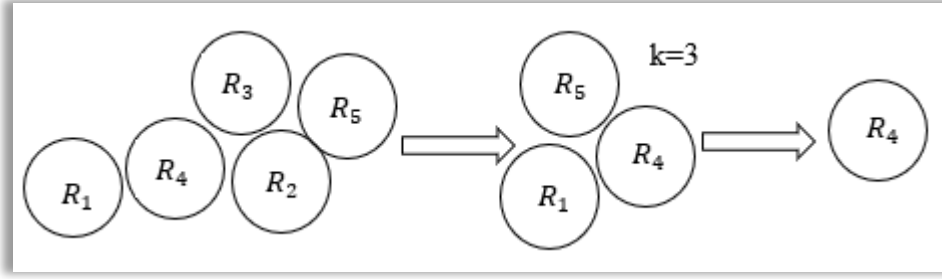
$$4. \text{ kromozom } R_4 = 0.19 \quad 0.19/0.98 = 0.1938$$

$$5. \text{ kromozom } R_5 = 0.21 \quad 0.21/0.98 = 0.2142$$



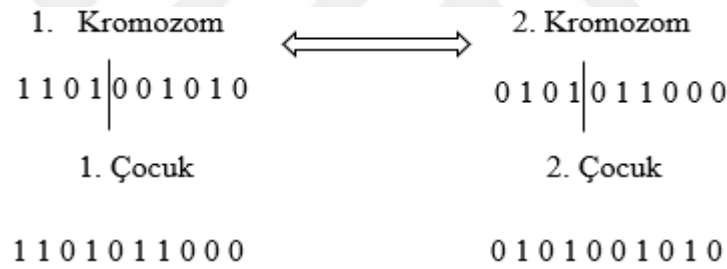
Şekil 4. 6. Rulet tekerleği seçimi

- Turnuva seçimi: Bu yöntemde popülasyondan rasgele olarak iki kromozom seçilir. Seçilen bu kromozomlar bir turnuvaya sokularak uygunluk değeri en iyi olan birey sonraki nesle aktarılmak üzere seçilir. Genelde turnuva iki birey arasında seçilir ancak isteğe bağlı olarak k adet seçilen bir sayıdaki bireyler arasında da uygulanabilir (Çakar, 2009). Rulet tekeri seçilimi ve turnuva seçilimi arasındaki fark, rulet seçimi yönteminde bireyler doğrudan uygunluk değerlerinden seçilme olasılığına sahip olduğundan bu durum popülasyondaki gen çeşitliliğinin azalmasına sebep olacaktır. Bu durumda verimli çalışmak isteyen algoritmanın performansını riske sokacaktır.



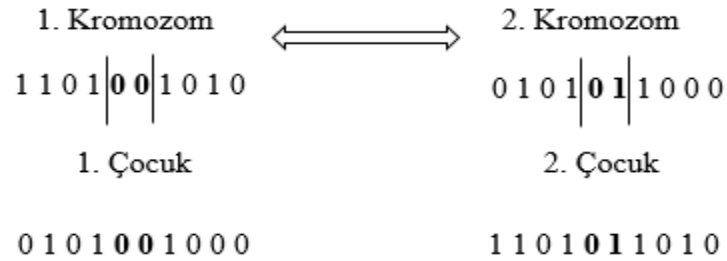
Şekil 4. 7. Turnuva seçimi

4. Çaprazlama İşlemi: Popülasyondan seçilmiş olan ebeveynler (kromozomlar) üzerinde gen karışımı uygulanır. Genetik algoritmanın türüne göre çeşitli çaprazlama teknikleri kullanılır. Yaygın olarak bilinen çaprazlama yöntemleri aşağıdaki gibi gösterilebilir.
- Tek Nokta Çaprazlama: Genetik algortmada kullanılan en basit yöntemdir. Kromozomdan rasgele bir kesme noktası alınır. İlk kesme noktasına kadar birinci kromozomdan, kesme noktasından sonra da ikinci kromozomdan kopyalama işlemi gerçekleştirilerek yeni bireyler üretilmiş olur.



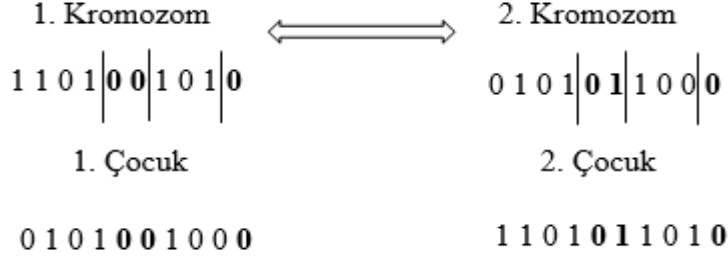
Şekil 4. 8. Tek nokta çaprazlama

- İki Nokta Çaprazlama: Kromozomlardan rasgele iki tane kesme noktası seçilir. Bu iki kesme noktası arasında kalan genler sabit tutularak etrafındaki genlerin değiştirilmesi ile yeni bireyler üretilir.



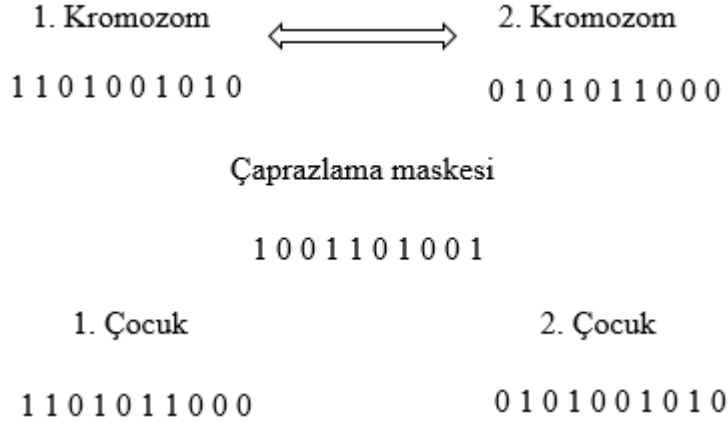
Şekil 4. 9. İki nokta çaprazlama

- Çok Nokta Çaprazlama: Kromozomlar daha fazla parçalara ayrılırlar. İki nokta çaprazlamanın daha gelişmiş versiyonudur (Bolat ve ark., 2004).



Şekil 4. 10. Çok nokta çaprazlama

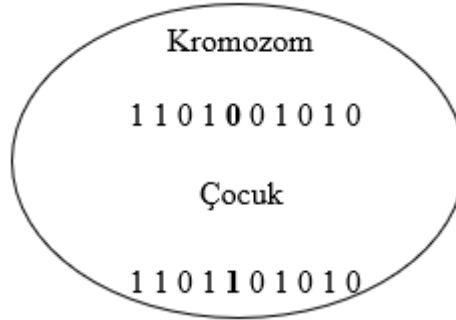
- Uniform Çaprazlama: Bu yöntemde gen karışımı yapabilmek için öncelikli olarak ikili (binary) rasgele oluşturulmuş bir çaprazlama maskesi elde edilir. Bu maskeye göre 1 değeri gördüğümüzde 1. çocuk için ilk değer 1. ebeveynden alınırken 0 değeri gördüğümüzde 2. ebeveynden alınır. Aynı durumun tersi 2. çocuk içinde uygulanır.



Şekil 4. 11. Uniform çaprazlama

5. Mutasyon İşlemi: Mutasyon elde edilen yeni çözümlerin önceki çözümü kopyalamasının önüne geçmek için kullanılır (Bolat ve ark., 2004).

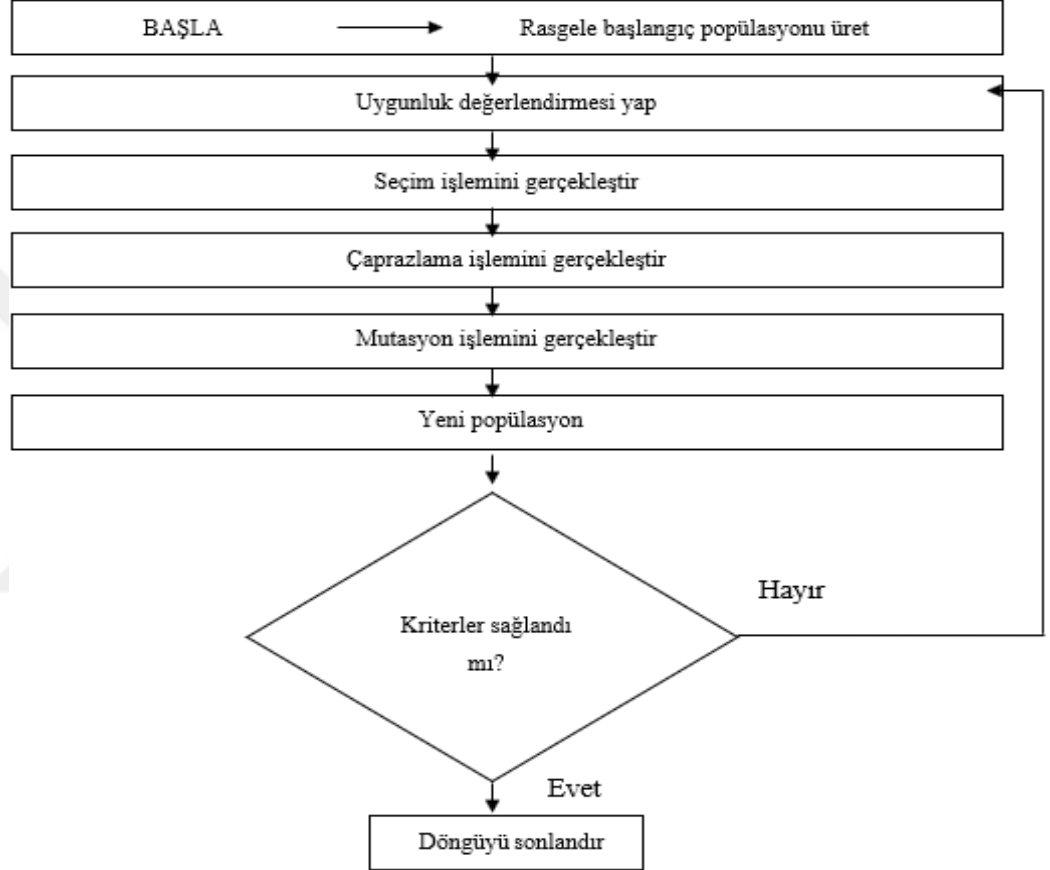
Çaprazlama işleminden sonra bazen aranılan gen popülasyonda bulunmayabilir. Böyle durumlarda da mutasyon mekanizması devreye girerek rasgele ihtimallerle, rasgele genleri, rasgele değerlere atar.



Şekil 4. 12. Mutasyon operatörü

Tüm bu işlemler sonucunda durdurma kriteri sağlanana kadar döngü devam eder. İstenilen amaca ulaşıldığında sonlandırma kriteri, problem çözümü için oluşturulmuş en iyi amaç fonksiyonu değerini, çözüm uzayında gerçek optimal cevaba en yakın ve en iyi çözüm çıktısı olarak sunar.

Genel olarak bir genetik algoritma akış süreci aşağıdaki şekildeki gibi gösterilebilir.



Şekil 4. 13. Genetik algoritma akış diyagramı

4.2.2.1 İkili kodlu genetik algoritma

İkili kodlu genetik algoritma 0 ve 1 (binary) bitlerin oluşturduğu bir yapıya sahiptir. Temelde basit ya da klasik genetik algoritma olarak bilinirler. Başlangıç popülasyonundaki her bir kromozom 0-1 dizilerinden oluşur. En yaygın kodlama biçimi olmasına rağmen çoğu problem için uygun bir yöntem değildir. Çünkü herhangi bir değişkenin alabileceği birden fazla değer olabilir. Bu durumda kodlamayı zorlaştırabilir (Taşkın ve Emel, 2009). İkili kodlama da çaprazlama ve mutasyon yöntemleri yukarıdaki örneklerde verilmiştir.

4.2.2.2 Permütasyon kodlu genetik algoritma

Bu kodlama yöntemi sıranın önemli olduğu durumlarda kullanılır. Literatürde sıklıkla gezgin satıcı problemlerinde ve bileşen atama problemlerinde karşımıza çıkmaktadır. Bu kodlama da diziler art arda sıralanan rakamlardan oluşurlar (Çakar, 2009). Optimizasyon probleminde hedeflenen amaç fonksiyonu değerinin en düşük olmasını sağlayan bileşen dizilimi ya da en yüksek olmasını sağlayan bileşen dizilimi söz konusu olduğunda kullanılabilir. Kısacası optimal veya minimal sıralamayı bulmak için kullanılırlar.

1. Kromozom	2. Kromozom
1 2 3 4 5 6 7 8 9 10	2 3 5 7 4 8 10 9 6 1

Şekil 4. 14. Permütasyon kodlama başlangıç kromozom

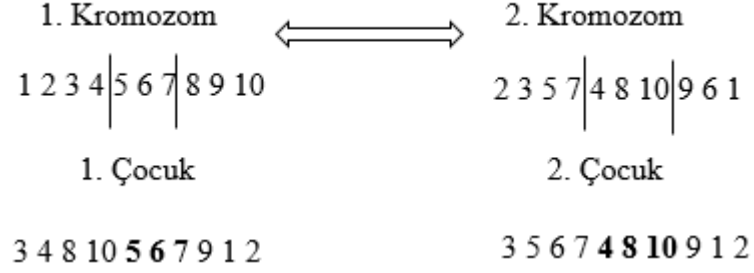
- Tek nokta çaprazlama: İkili kodlamadaki tek nokta çaprazlama ile aynı mantıkta çalışır. Art arda dizilmiş kromozomlardan rasgele bir kesme noktası seçilir. 1. ebeveynden belirlenmiş kesme noktasına kadar olan kısım 1. çocuk için aynı şekilde alınır. Daha sonra kesme noktasının diğer tarafı da 2. ebeveynden aynı rakamlar dizilimde tekrar etmeyecek şekilde kontrol edilerek alınır. Aynı yöntem 2. ebeveynden, 2. çocuk içinde uygulanarak iki yeni birey elde edilmiş olur.

1. Kromozom	2. Kromozom
1 2 3 4 5 6 7 8 9 10	2 3 5 7 4 8 10 9 6 1
1. Çocuk	2. Çocuk
1 2 3 4 8 10 9 6 5 7	2 3 5 7 6 8 9 10 1 4

Şekil 4. 15. Permütasyon kodlamada tek nokta çaprazlama

- OX (Sıralı Çaprazlama): Sistemdeki bileşenlerin sırası önemlidir. İki farklı rasgele kesme noktası seçilir. 1. ebeveynde kesme noktaları arasında kalan değerler aynen aynı noktalar arasında kalacak şekilde 1. çocuğa aktarılır. Daha sonra kalan genler de 2. ebeveynden yine tekrarsız değerlerin olması amacıyla

kontrol edilerek kromozom tamamlanır. Aynı durum 2. çocuk içinde gerçekleştirilir.



Şekil 4. 16. Permütasyon kodlamada XO çaprazlama

- Mutasyon: Kromozom dizisinde rasgele iki tane genin yeri değiştirilerek mutasyona uğrattılır. Şekil 4.17’de rasgele olarak 3. gen ile 9. gen yer değiştirilerek mutasyon gerçekleştirilmiştir.



Şekil 4. 17. Permütasyon kodlamada mutasyon

Genetik Algoritmada sıranın önemli olduğu durumlarda permütasyon kodlama yönteminin kullanılması daha uygun olur. Sistemler açısından ele alındığında sistemleri verimli hale getirmek ve aynı zamanda sistem güvenilirliğini en üst seviyeye çıkartmak için en uygun bileşen dizilişini bulmak gerekir. Bunun içinde bileşenlerin sistemin genel güvenilirliğine olumlu katkısı olabilmesi açısından yeni pozisyonlara atanmaları bileşen atama problemlerini ve çözümünü ortaya çıkarır. Bileşen atama problemleri ilk olarak Derman tarafından önerilmiştir. Burada amaç farklı güvenilirliklere sahip n tane bileşene n tane pozisyon verilen bir sistemde en uygun yer değiştirmeyi bulmaktır. Ancak bileşen atama problemi genellikle bileşenlerin dizilme biçimini verimli hale getirme problemidir ve karmaşık sistemler için uygun bir zaman içerisinde en uygun atamayı bulmak zordur. Sistemdeki zayıflıkları veya kritik bileşenleri tanımlamak ve bileşen

arızalarının etkisini ölçebilmek için önem ölçülerinin dikkate alınması gerekmektedir (Gökdere ve Gural, 2018).

Önem ölçüleri, bakım maliyeti kısıtlamasıyla sistem güvenilirliğinde en büyük gelişmeyi elde etmek için kritik bileşenleri seçme yeteneğine sahiptir. Sonunda bir güvenilirlik optimizasyon modeli oluşturulur ve güvenilirlik optimizasyon problemini verimli bir şekilde çözmek için önem ölçüsüne dayalı genetik algoritmalar geliştirilir. Güvenilirlik mühendisliğinde, önem ölçüleri genellikle en zayıf bileşenleri tanımlamak için kullanılır ve sistem güvenilirliği, bileşen güvenilirliğini artırarak iyileştirilebilir (Wang et al, 2018).

Önem ölçüleri kavramı, ilk olarak Birnbaum tarafından önerilmiştir, bu da bileşen güvenilirlik değişikliklerinin sistem güvenilirliğine katkısı anlamına gelir (Si et al, 2016).

Birnbaum (1969) güvenilirlik önemi ölçüsü (BI) bileşen güvenilirliğindeki değişikliklerin sistem güvenilirliği üzerindeki etkisini ifade eder. Matematiksel olarak, sistem güvenilirliğinin bileşen güvenilirliğine oranı şeklinde formüle edilir.

$$X_i = \begin{cases} 1, & i. \text{ bileşen başarılı ise} \\ 0, & i. \text{ bileşen başarısız ise} \end{cases}$$

Bileşen durum vektörü:

$$\underline{X} = (X_1, X_2, \dots, X_n)$$

i. bileşenin güvenilirliği:

$$p_i = P(X_i = 1)$$

Bileşen güvenilirlik vektörü:

$$\underline{p} = (p_1, p_2, \dots, p_n)$$

Sistem durumu:

$$\Phi(\underline{X}): 1(0) \text{ ise başarılı (başarısız)}$$

Sistem güvenilirliği:

$$R(\underline{p})$$

Birnbaum (1969)'un tanımına göre n bileşenli bir sistemde i 'nci bileşenin B-önemi ölçütü $I_B^i(\underline{p})$, sistemin başarılı ve başarısız olduğu durumdaki B-önem ölçütünün toplanmasıyla elde edilirken son durumda bileşen güvenilirliklerinin bağımsız olduğu varsayımı altında ise Denklem (4.15) gibi olur.

$$I_{BS}^i(1_i, \underline{p}) = P\{\Phi(\underline{X}) = 1/X_i = 1\} - P\{\Phi(\underline{X}) = 1\} \quad (4.13)$$

$$I_{BF}^i(0_i, \underline{p}) = P\{\Phi(\underline{X}) = 0/X_i = 0\} - P\{\Phi(\underline{X}) = 0\} \quad (4.14)$$

$$I_B^i(\underline{p}) = \frac{\sigma R(\underline{p})}{\sigma(p_i)} = R(1_i, \underline{p}) - R(0_i, \underline{p}) \quad (4.15)$$

Doğrusal ardışık 4'ten 2 çıkışlı F sistemi için Birnbaum önem değerine göre genetik algoritma ile sistem güvenilirliğinin hesaplanması

Doğrusal ardışık 4'ten 2 çıkışlı F sisteminin güvenilirlik fonksiyonu aşağıdaki gibidir:

$$R(p) = (p_1 + p_2 - p_1 p_2)(p_2 + p_3 - p_2 p_3)(p_3 + p_4 - p_3 p_4)$$

Bileşen güvenilirlikleri sırasıyla $p_1 = 0.96, p_2 = 0.93, p_3 = 0.87, p_4 = 0.76$ iken sistem güvenilirliği aşağıdaki gibi hesaplanır.

$$R(p) = (0.96 + 0.93 - (0.96)(0.93))(0.93 + 0.87 - (0.93)(0.87)) \\ (0.87 + 0.76 - (0.87)(0.76)) = \underline{0.9573}$$

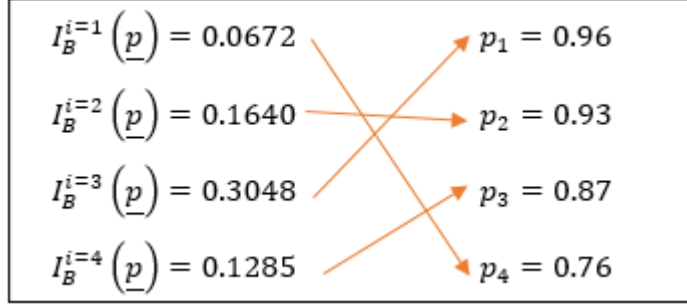
Birnbaum hesaplaması yapabilmek için Denklem (4.15) kullanılarak her bir değişkenin türevleri alınarak önem değerleri hesaplanır.

$$I_B^{i=1}(\underline{p}) = \frac{\sigma R(\underline{p})}{\sigma(p_1)} = -(p_2 - 1)(p_2 + p_3 - p_2 p_3)(p_3 + p_4 - p_3 p_4) = 0.0672$$

$$I_B^{i=2}(\underline{p}) = \frac{\sigma R(\underline{p})}{\sigma(p_2)} = -(p_3 - 1)(p_1 + p_2 - p_1 p_2)(p_3 + p_4 - p_3 p_4) \\ -(p_1 - 1)(p_2 + p_3 - p_2 p_3)(p_3 + p_4 - p_3 p_4) = 0.1640$$

$$I_B^{i=3}(\underline{p}) = \frac{\sigma R(\underline{p})}{\sigma(p_3)} = -(p_4 - 1)(p_1 + p_2 - p_1 p_2)(p_2 + p_3 - p_2 p_3) - (p_2 - 1)(p_1 + p_2 - p_1 p_2)(p_3 + p_4 - p_3 p_4) = 0.3048$$

$$I_B^{i=4}(\underline{p}) = \frac{\sigma R(\underline{p})}{\sigma(p_4)} = -(p_3 - 1)(p_1 + p_2 - p_1 p_2)(p_2 + p_3 - p_2 p_3) = 0.1285$$



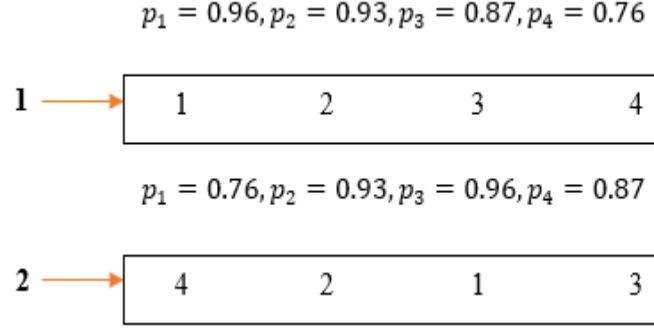
Şekil 4. 18. Birnbaum önem ölçüsüyle hesaplanan pozisyonlara bileşen atama

Şekil 4.18.'de 1. bileşenin Birnbaum önem değeri 0.0672 çıkmıştır. Sırasıyla 2., 3. ve 4. bileşenlerinde Birnbaum önem değerleri 0.1640, 0.3048 ve 0.1285 olarak hesaplanmıştır. Buna göre önem değeri en yüksek olan 3. pozisyona, bileşen güvenilirlik değeri en yüksek olan (0.96) 1. bileşen atanır. Daha sonra önem değeri en yüksek olan 2. pozisyona, bileşen güvenilirlik değeri geriye kalanlardan en yüksek olan (0.93) 2. bileşen atanır. Sonrasında 4. pozisyona 3. bileşen atanırken 1. pozisyona ise 4. bileşen atanacaktır. Atama işlemleri gerçekleştikten sonraki sistem güvenilirliği ise aşağıdaki gibi olacaktır.

$$R(p) = (0.76 + 0.87 - (0.76)(0.87))(0.87 + 0.93 - (0.87)(0.93)) \\ (0.93 + 0.96 - (0.93)(0.96)) = 0.9753$$

Sonuçlara bakıldığında başlangıçtaki sistem güvenilirliği 0.9573 iken, Birnbaum önem kavramıyla bileşen atama işlemleri gerçekleştirdikten sonra sistem güvenilirliği 0.9753'e yükselmiştir.

Birnbaum önem kavramı kullanılarak permütasyon kodlu genetik algoritma bileşen atama problemleri için tasarlanabilir.



Şekil 4. 19. Birnbaum önem tabanlı genetik algoritma başlangıç popülasyonu

Şekil 4.19.'da 1 ve 2 ile ifade edilen başlangıç popülasyondaki kromozom dizileridir. 2. kromozom görüldüğü üzere rassal olarak oluşturulmak yerine Birnbaum önem kavramından elde edilen bileşen dizilimi olarak popülasyona dahil edilmiştir. Yani, bir Genetik Algoritmada rasgele oluşturularak üretilen başlangıç popülasyonu içerisine daha iyi sonuç vermesi açısından Birnbaum kavramını kullanarak bulunmuş olan bileşen dizilimi dahil edilebilir. Daha sonra çaprazlama ve mutasyon işlemlerini permütasyon kodlu genetik algoritma başlığı altındaki yöntemlerden yararlanarak algoritma sürdürülür.

$$\begin{array}{c}
 \begin{bmatrix} p1 & p2 & p3 & p4 \\ 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \\ 2 & 1 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{bmatrix}
 \end{array}
 =
 \begin{array}{c}
 \begin{bmatrix} R(p) \\ 0,957296 \\ 0,975349 \\ 0,961064 \\ 0,968698 \end{bmatrix}
 \end{array}
 \quad (4.16)$$

Denklem (4.16)'da başlangıç popülasyonu 4 tane kromozomdan oluşmuş, üç tanesi rasgele ve bir tanesi de (2. satır-kromozom) Birnbaum önem değeriyle hesaplanarak elde edilmiştir. Genetik algoritma uygularken başlangıç popülasyonuna dahil edilen Birnbaum önem ölçüsü ile doğrusal ardışık 4'ten 2 çıkışlı F sisteminin güvenilirliği, Birnbaum önem ölçüsünün olmadığı ve Denklem (4.17)'de gösterildiği gibi dört kromozomunda rasgele üretildiği bir genetik algorithmaya göre daha yüksek sistem güvenilirlik değerini verdiği gözlemlenmiştir.

$$\begin{array}{c}
 \begin{bmatrix} p1 & p2 & p3 & p4 \\ 1 & 3 & 2 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \\ 2 & 1 & 4 & 3 \end{bmatrix}
 \end{array}
 =
 \begin{array}{c}
 \begin{bmatrix} R(p) \\ 0,969187 \\ 0,957296 \\ 0,9649 \\ 0,956813 \end{bmatrix}
 \end{array}
 \quad (4.17)$$

Denklem (4.16) iterasyon sayısı 20, çaprazlama oranı 0.8 ve mutasyon oranı 0.1 iken Birnbaum önem tabanlı genetik algoritmanın (BIGA) 20. iterasyon sonucundaki en yüksek sistem güvenilirlik değerini (0.975349) ifade ederken, Denklem (4.17) ise Birnbaum önem ölçüsünün olmadığı, iterasyon sayısı 20, çaprazlama oranı 0.8 ve mutasyon oranı 0.1 iken 20. iterasyon sonucundaki en yüksek sistem güvenilirlik değerini (0.969187) vermektedir. Problem çözümü için MATLAB R2016b programı kullanılmış ve her iki yöntem içinde 20. iterasyon sonundaki çözüm çıktıları sırasıyla Şekil 4.20. ve 4.21.'de verilmiştir.

	pozisyon	güvenilirlik
Kromozom	[4 2 1 3]	0,9753487
	[3 2 4 1]	0,9649001
	[3 4 1 2]	0,9568129
	[2 4 3 1]	0,947571

Şekil 4. 20. BIGA 20. iterasyon sonucundaki sistem güvenilirlikleri

	pozisyon	güvenilirlik
Kromozom	[1 3 2 4]	0,9691868
	[1 2 3 4]	0,957296
	[1 4 2 3]	0,9649001
	[2 1 4 3]	0,9568129

Şekil 4. 21. GA 20. iterasyon sonucundaki sistem güvenilirlikleri

Şekil 4.20.'ye bakıldığında Birnbaum önem değeriyle hesaplanmış genetik algoritmanın (BIGA), Şekil 4.21'de Birnbaum önem değerinin kullanılmadığı genetik algorithmaya göre daha yüksek sistem güvenilirlik değeri verdiği görülmüştür.

4.2.2.3 Gerçek kodlu genetik algoritma (GGA)

Gerçek kodlu genetik algoritmalar, çözüm uzayı sınırları içerisinde kromozomların olası çözümlerini sayısal değerlerle ifade ederler. Gerçek optimizasyon problemlerinde genlerin kendi değerleriyle birlikte kullanılması, genlerin ikili düzende tanımlanmasına göre daha yararlıdır. Çünkü kromozom

değerleri çok büyük ya da ondalıklı sayılardan oluşabilir. Bu durumda da kromozomları bitlerle ifade etmek oldukça zor ve karmaşık bir hal alabilir. Klasik genetik algorithmada kullanılan gen operatörleri GGA'da da kullanılır ancak temel farklılık uygulamada karşımıza çıkmaktadır (Bağış ve Özçelik, 2007).

GGA problemin çözümüne ilişkin sayısal değerlerden oluşan başlangıç bir popülasyonun belli sınırlar arasında üretilmesiyle başlar. Daha sonra belirlenen bir amaç fonksiyonundan elde edilen uygunluk değerleri değerlendirilir. En iyi ebeveynlerine sahip bireylere seçim işlemi uygulanır ve seçilmiş olan bireylere daha sonra çaprazlama işlemiyle gen karışımı ve mutasyon işlemlerinin ardından döngü tamamlanır.

1. Kromozom	2. Kromozom
0.25 0.32 0.58 0.49 0.76	0.23 0.62 0.57 0.33 0.71

Şekil 4. 22. GGA başlangıç kromozom

- Aritmetik Çaprazlama: Bu çaprazlamaya göre 1. ve 2. Ebeveynin (1.E ve 2.E) çaprazlanması sonucunda;

$$1. \text{ Çocuk (Kız): } K = \alpha 1.E + (1 - \alpha) 2.E$$

$$2. \text{ Çocuk (Erkek): } E = (1 - \alpha) 1.E + \alpha 2.E$$

şeklinde oluşturulur. α , Uniform (0,1) dağılımdan rasgele olarak seçilen bir değerdir (Satman, 2016). Şekil 4.16.'ya göre çaprazlama işlemi gerçekleştirilirse sonuç aşağıdaki gibi olur.

$$\alpha = 0.2 \text{ için}$$

$$1. \text{ Çocuk (Kız): } (0.23 \ 0.56 \ 0.57 \ 0.36 \ 0.72)$$

$$2. \text{ Çocuk (Erkek): } (0.24 \ 0.38 \ 0.57 \ 0.45 \ 0.75)$$

- Normal Dağılan Mutasyon: Normal dağılımdan yararlanarak gerçekleştirilen bir mutasyon türüdür. Aşağıdaki ifadeler kullanılarak hesaplanır.

$$c_i(\text{yeni}) = c_i(\text{eski}) + \sigma N(0,1)$$

$$\sigma = m(x_{\max} - x_{\min})$$

5. n 'den k ÇIKIŞLI SİSTEMLERDE GÜVENİLİRLİK OPTİMİZASYONU

Önceki bölümde, çalışmada ele alınan optimizasyon yöntemleri (TLBO ve GA) daha küçük boyutlu sistemler için incelenmiş ve hesaplamaları ayrıntılı olarak ele alınmıştır. Bu bölümde ise daha fazla bileşen için hesaplamalar gerçekleştirilmiş olup doğrusal n 'den k çıkışlı F sistemi ile doğrusal ardışık n 'den k çıkışlı F sisteminin güvenilirlik optimizasyonu için meta-sezgisel yöntemlerden olan Öğretme-Öğrenme Esaslı Optimizasyon Algoritması (TLBO) ile Genetik Algoritma (GA) yöntemleri kullanılmıştır. Bu sistemlerin yapı fonksiyonundan elde edilen güvenilirlik fonksiyonları, sistem güvenilirliği optimizasyonu için gerekli olan amaç fonksiyonu olarak belirlenmiştir. Amaç, sistem güvenilirliklerini optimum düzeye çıkarmak ve en yüksek sistem güvenilirlik değerini veren bileşen güvenilirliklerini bulmaktır. Bunun için her iki algoritma yöntemiyle MATLAB R2016b programı kullanılarak hesaplanan sonuçlar karşılaştırılmıştır. Çalışma için aynı zamanda Genetik Algoritma, Gerçek Kodlu Genetik Algoritma olarak tasarlanmıştır.

Tablo 5. 1. TLBO ve GA yöntemleri için gerekli olan tanımlamalar

TLBO	GGA
Amaç Fonksiyonu: $\max(R(p))$	Amaç Fonksiyonu: $\max(R(p))$
Popülasyon P_n : 10, 50, 100	Popülasyon P_n : 10, 50, 100
Değişken Sayısı D_n : 10	Değişken Sayısı D_n : 10
İterasyon Sayısı I_n : 20, 50, 100	İterasyon Sayısı I_n : 20, 50, 100
Değişken Aralığı: $0 \leq p_1, p_2, \dots, p_{10} \leq 1$	Değişken Aralığı: $0 \leq p_1, p_2, \dots, p_{10} \leq 1$
-	Çaprazlama Oranı: 0.7, 0.8, 0.9
-	Mutasyon Oranı: 0.1, 0.2
-	Çaprazlama Türü: Aritmetik Çaprazlama
-	Mutasyon Türü: Normal Dağılan Mutasyon
-	Seçim Yöntemi: Rastgele

Tablo 5.1'de TLBO ve GA yöntemlerinde kullanılmak üzere bazı tanımlar verilmiştir. Doğrusal 10'den 2 çıkışlı F sistemi ile doğrusal ardışık 10'den 2 çıkışlı F sisteminin karşılaştırılmaları sunulacaktır. Her iki sistem türü için de amaç fonksiyonları sistem güvenilirliğini maksimize etmektir. Başlangıç popülasyonu

için gerekli olan popülasyon sayısı ilk olarak 10, sonra 50 ve daha sonra 100 alınarak kıyaslanmak üzere aynı veri setinden oluşturulmuştur. Değişken sayıları her iki yöntem için de sabit olup 10 olarak alınmıştır. İterasyon sayıları da her iki algoritma için aynı olup sırasıyla 10, 20, 50 ve 100 olarak denemeler gerçekleştirilmiştir. Bileşen güvenilirlikleri söz konusu olduğundan alt ve üst limit değerleri 0 ile 1 arasındadır. Genetik Algoritma yönteminde, Öğretme-Öğrenme Esaslı Optimizasyon Algoritmasına ek olarak çaprazlama oranı 0.7, 0.8 ve 0.9 olarak alınmış, mutasyon oranı ise çaprazlama oranının aksine daha küçük değerlerden oluşturulmuştur. Çaprazlama türü, aritmetik çaprazlama ve mutasyon türü de normal dağılan mutasyon şeklinde sunulmuştur. Seçim yönteminde ise seçim işlemleri rasgele olarak yapılmıştır.

5.1 Doğrusal 10'dan 2 çıkışlı F Sisteminin TLBO ve GA ile Optimizasyonları ve Karşılaştırılması

TLBO ve GA yöntemleriyle doğrusal 10'dan 2 çıkışlı F sisteminin güvenilirlikleri hesaplanmıştır. Ele alınan sistem için rasgele üretilmiş olan başlangıç bileşen güvenilirlikleri her iki yöntem için de aynı alınmıştır. Sistemin amaç fonksiyonu ise R_p sistem güvenilirliği olarak verilmiştir.

$$\Phi(x) = (X_1 \cup X_2)(X_1 \cup X_3)(X_1 \cup X_4) \dots (X_9 \cup X_{10}) ; 1 \leq i \leq n$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_1 + X_3 - X_1X_3) \dots (X_9 + X_{10} - X_9X_{10})$$

$$R_p = (p_1 + p_2 - p_1p_2)(p_1 + p_3 - p_1p_3) \dots (p_9 + p_{10} - p_9p_{10})$$

Başlangıç bileşen güvenilirlikleri

$$\begin{bmatrix} p1 & p2 & p3 & p4 & p5 & p6 & p7 & p8 & p9 & p10 \\ 0,742494 & 0,370786 & 0,207092 & 0,951109 & 0,663568 & 0,951759 & 0,507471 & 0,149387 & 0,204428 & 0,991631 \\ 0,952521 & 0,523495 & 0,114473 & 0,003138 & 0,750896 & 0,323302 & 0,493768 & 0,506119 & 0,233175 & 0,722217 \\ 0,428517 & 0,068051 & 0,493814 & 0,434693 & 0,963063 & 0,467606 & 0,934832 & 0,750566 & 0,172260 & 0,162045 \\ 0,483397 & 0,707022 & 0,331470 & 0,426662 & 0,591464 & 0,549272 & 0,187015 & 0,047393 & 0,186121 & 0,200941 \\ 0,970599 & 0,229230 & 0,209536 & 0,229554 & 0,846114 & 0,844191 & 0,089506 & 0,125205 & 0,629511 & 0,173823 \\ 0,047987 & 0,803769 & 0,301624 & 0,266876 & 0,823950 & 0,679397 & 0,873601 & 0,770902 & 0,598692 & 0,985864 \\ 0,700809 & 0,444155 & 0,978896 & 0,809006 & 0,904185 & 0,267142 & 0,797172 & 0,646386 & 0,155901 & 0,747468 \\ 0,661753 & 0,558728 & 0,697705 & 0,304829 & 0,556989 & 0,505504 & 0,151534 & 0,649512 & 0,931877 & 0,381886 \\ 0,405882 & 0,244154 & 0,885121 & 0,517032 & 0,731639 & 0,738188 & 0,864595 & 0,411887 & 0,738064 & 0,482645 \\ 0,787814 & 0,529531 & 0,459164 & 0,718206 & 0,134419 & 0,921456 & 0,466229 & 0,714487 & 0,691321 & 0,905117 \end{bmatrix} = \begin{bmatrix} R(p) \\ 0,071139 \\ 0,009981 \\ 0,039062 \\ 0,002952 \\ 0,010521 \\ 0,236740 \\ 0,326738 \\ 0,120006 \\ 0,249649 \\ 0,288137 \end{bmatrix}$$

TLBO algoritması için her bir değişkenin ara işlemlerde kullanılmak üzere ortalama vektörü M_p hesaplanır. Daha sonra sistem güvenilirliği değerini en yüksek

veren satır (öğrenci) yine ara işlemlerde kullanılmak üzere en iyi öğretmen olarak kaydedilir.

Ortalama vektör ve en iyi öğretmen

M_p	0,618177	0,447892	0,467890	0,466110	0,696629	0,624782	0,536572	0,477184	0,454135	0,575364
$P_{teacher}$	0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468

Daha sonra öğretmen ve öğrenci aşamaları uyguladıktan sonra 1. iterasyon sonlanır. Bu işlemler MATLAB R2016b programı ile yapılmış ve sunulmuştur. 1. iterasyon sonundaki sistem ve bileşen güvenilirlik değerleri aşağıdaki gibidir.

TLBO yöntemi için Şekil 4.1.'deki TLBO akış diyagramındaki adımlar gerçekleştirildikten sonra 1. ve 20. iterasyon sonundaki bileşen ve sistem güvenilirlik değerleri aşağıdaki gibidir:

TLBO 1. iterasyon sonundaki bileşen ve sistem güvenilirliği (DF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,78325440	0,51355950	0,43380334	0,74163822	0,18765730	0,92450458	0,47037798	0,65763214	0,64233451	0,91382090	0,000991
0,78855030	0,51328709	0,43459306	0,67618502	0,28439053	0,81399443	0,49669364	0,64804687	0,55829347	0,90435236	0,000565
0,29696474	0,87680841	0,47111682	0,38744558	0,83331641	0,17537332	0,83088495	0,61666060	0,88365839	0,07716238	0,000008
0,56401711	0,60954658	0,57154676	0,56844180	0,70742667	0,44465331	0,41327177	0,26951003	0,17491504	0,40360286	0,000000
0,64612397	0,82339584	0,89721642	0,89721642	0,49296582	0,82339584	0,92080976	0,82339584	0,82339584	0,82339584	0,159222
0,89721642	0,82339584	0,89721642	0,08012141	0,89286475	0,63550700	0,88032345	0,82339584	0,49464307	0,82339584	0,031731
0,87345938	0,57658930	0,82339584	0,82339584	0,82339584	0,00363614	0,77668335	0,79694689	0,89721642	0,91450697	0,040532
0,90998780	0,34227804	0,85189587	0,24860860	0,36895126	0,73015371	0,89721642	0,67186734	0,96468955	0,58924747	0,003021
0,42306365	0,14435563	0,95042875	0,53747433	0,71560031	0,82696976	0,86991290	0,37958491	0,71509723	0,54660874	0,000462
0,85072560	0,43310786	0,66139560	0,47639375	0,25518803	0,82294763	0,68815925	0,69254076	0,83208810	0,74246458	0,002220

TLBO 20. iterasyon sonundaki bileşen ve sistem güvenilirliği (DF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,8637540	0,8990818	0,9989595	0,7468284	0,3803904	0,9912131	0,8708518	0,8937223	0,6416057	0,9695474	0,281128
0,9354769	0,8868574	0,9968370	0,8897116	0,3330747	0,9619404	0,8576992	0,9067976	0,6263224	0,9304035	0,321051
0,9362069	0,8778150	0,9996243	0,9076041	0,2485928	0,9998923	0,8554074	0,9121564	0,6187377	0,9704844	0,336277
0,9771442	0,8585052	0,9990705	0,9855996	0,2578761	0,9883175	0,8484129	0,9222366	0,6133513	0,9467854	0,375622
0,9782946	0,8581433	0,9969730	0,9740663	0,3450207	0,9586951	0,8498640	0,9176597	0,6204459	0,9137679	0,368294
0,9775361	0,8584594	0,9969867	0,9728349	0,3446771	0,9587807	0,8499898	0,9176373	0,6204805	0,9140354	0,367466
0,9404134	0,8822510	0,9969428	0,9019757	0,3373170	0,9600337	0,8565972	0,9089672	0,6257583	0,9260413	0,326358
0,9051059	0,8652437	0,9978422	0,8456347	0,3442190	0,9833098	0,8614377	0,9017091	0,6205515	0,9701052	0,304577
0,9400637	0,8766163	0,9985811	0,9108360	0,2890919	0,9826534	0,8553971	0,9118602	0,6213637	0,9502514	0,334644
0,9384948	0,8429982	0,9981943	0,9094224	0,3227049	0,9936089	0,8552818	0,9087414	0,6121701	0,9753502	0,340886

TLBO algoritmasında 1. iterasyon sonunda en iyi sistem güvenilirlik değeri 0.159222 iken, 20. iterasyon sonunda sistem güvenilirlik değeri 0.375622 değerine yükselmiştir. Şekil 5.1'de ise her iterasyon sonucundaki en iyi sistem güvenilirlik değerleri grafikte gösterilmiştir.



Şekil 5. 1. TLBO her iterasyon için en iyi sistem güvenilirlikleri grafiği (DF/10/2)

GA için başlangıç bileşen güvenilirlikleri yine aynı değerlerden olup Şekil 4.13'te verilen genetik algoritma akış şemasındaki adımlar uygulandıktan sonra 1. ve 20. iterasyon sonundaki bileşen ve sistem güvenilirlikleri aşağıdaki gibi bulunur:

GA 1. iterasyon sonucu bileşen ve sistem güvenilirliği (DF/10/2)

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	R(p)
0,732247	0,662904	0,695904	0,901689	0,895688	0,923036	0,666904	0,856904	0,896904	0,934729	0,22359289
0,736055	0,666904	0,890002	0,857425	0,670064	0,666904	0,694605	0,666904	0,666904	0,804369	0,03535803
0,776352	0,666904	0,817536	0,726887	0,666904	0,835103	0,673679	0,677449	0,666904	0,778281	0,03219992
0,666904	0,666904	0,827629	0,669040	0,666904	0,867180	0,694256	0,666904	0,716630	0,736235	0,02407894
0,712270	0,666904	0,666904	0,800324	0,666904	0,666904	0,666904	0,683423	0,666904	0,874303	0,01778118
0,702293	0,666904	0,666904	0,718176	0,666904	0,813504	0,666904	0,688441	0,727931	0,745550	0,01732495
0,747273	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,675557	0,895806	0,666904	0,01384623
0,666904	0,666904	0,666904	0,668228	0,717533	0,792463	0,666904	0,670970	0,712754	0,666904	0,01049706
0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,00503053
0,700809	0,444155	0,978896	0,809900	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,00180392

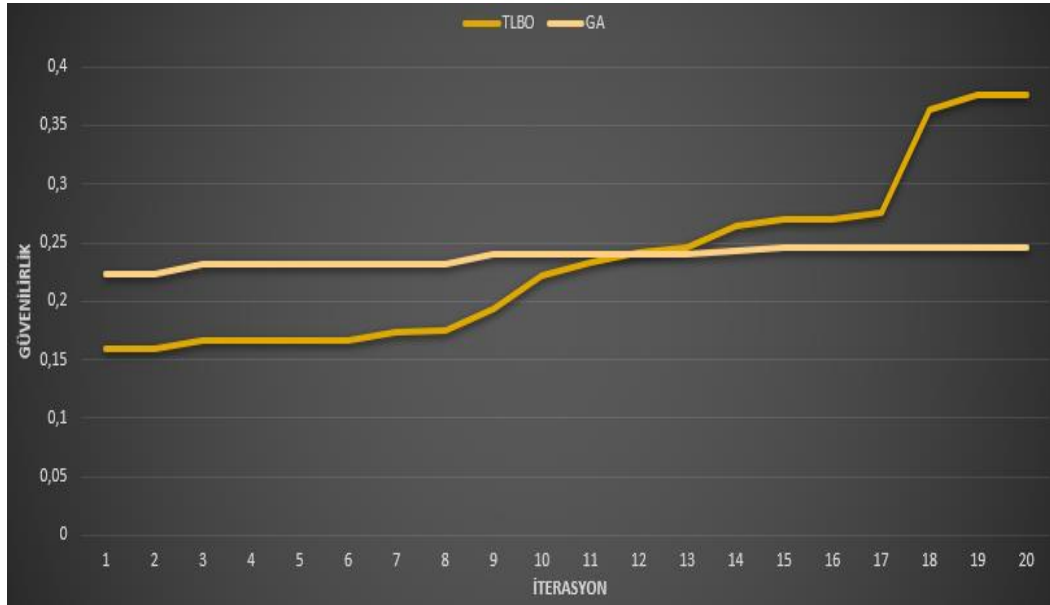
GA 20. iterasyon sonucu bileşen ve sistem güvenilirliği (DF/10/2)

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	R(p)
0,784425	0,784425	0,799167	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,245500
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,792950	0,784425	0,806279	0,897878	0,243040
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,792950	0,784425	0,806279	0,897878	0,243040
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,239703
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,239703
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,234885
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,784425	0,897878	0,231245
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,784425	0,897878	0,229068
0,784425	0,784425	0,784425	0,897878	0,784425	0,897878	0,784425	0,784425	0,784425	0,897878	0,223521
0,784425	0,799210	0,784425	0,784425	0,784425	0,784425	0,784425	0,795720	0,798301	0,784425	0,127351

GA'da 0.1 mutasyon oranı ve 0.7 çaprazlama oranına göre 1. iterasyon sonunda en iyi sistem güvenilirlik değeri 0.22359289 iken, 20. iterasyon sonunda sistem güvenilirlik değeri 0.245500 değerine yükselmiştir. Şekil 5.2.'de GA'da her iterasyon sonundaki en iyi sistem güvenilirlik değerleri grafikte verilmiştir.



Şekil 5. 2. GA her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DF/10/2)



Şekil 5. 3. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (DF/10/2)

Şekil 5.3.'e bakıldığında ise TLBO ve GA yöntemlerindeki en iyi sistem güvenilirlik değerleri tek bir grafikte sunulmuştur. Grafiğe bakıldığında TLBO yönteminin yine GA yöntemine göre 20. iterasyon sonunda daha yüksek sistem güvenilirlik değerine sahip olduğu görülmüştür.

5.2 Doğrusal Ardışık 10'dan 2 çıkışlı F Sisteminin TLBO ve GA ile Optimizasyonları ve Karşılaştırılması

TLBO ve GA yöntemleriyle doğrusal ardışık 10'dan 2 çıkışlı F sisteminin güvenilirlikleri hesaplanmıştır. Ele alınan sistem için rasgele üretilmiş olan başlangıç bileşen güvenilirlikleri her iki yöntem için de aynıdır. Sistemin amaç fonksiyonu ise R_p sistem güvenilirliği olarak verilmiştir.

$$\Phi(x) = (X_1 \cup X_2)(X_2 \cup X_3)(X_3 \cup X_4) \dots (X_9 \cup X_{10}) ; 1 \leq i \leq n$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_2 + X_3 - X_2X_3) \dots (X_9 + X_{10} - X_9X_{10})$$

$$R_p = (p_1 + p_2 - p_1p_2)(p_2 + p_3 - p_2p_3) \dots (p_9 + p_{10} - p_9p_{10})$$

TLBO yöntemi için Şekil 4.1.'deki TLBO akış diyagramındaki adımlar gerçekleştirildikten sonra 1. ve 20. iterasyon sonundaki bileşen ve sistem güvenilirlik değerleri aşağıdaki gibidir:

TLBO 1. iterasyon sonundaki bileşen ve sistem güvenilirlikleri (DAF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,395989	0,587513	0,261933	0,613574	0,746917	0,810139	0,694612	0,463048	0,397440	0,991273	0,169907
0,728335	0,550660	0,564153	0,235745	0,601391	0,463782	0,229901	0,616677	0,771883	0,459818	0,085219
0,493622	0,082789	0,784079	0,622043	0,847222	0,335506	0,868804	0,758009	0,460717	0,311755	0,161374
0,610658	0,582219	0,299620	0,375176	0,657981	0,626307	0,161545	0,067718	0,301939	0,193858	0,005230
0,838644	0,446022	0,389747	0,582319	0,332330	0,899970	0,361468	0,550617	0,674133	0,701756	0,155471
0,406973	0,837330	0,266815	0,271003	0,853460	0,707986	0,958950	0,791351	0,575728	0,869620	0,267127
0,737702	0,387656	0,869620	0,986543	0,869620	0,120162	0,869620	0,883195	0,096138	0,974023	0,518832
0,655755	0,559455	0,709736	0,311052	0,552989	0,509262	0,144474	0,652470	0,946290	0,374866	0,123629
0,417712	0,235786	0,913949	0,558687	0,752933	0,693851	0,892252	0,438164	0,695875	0,513895	0,263696
0,802034	0,428544	0,766904	0,929573	0,583253	0,360975	0,766396	0,889660	0,243773	0,923154	0,385272

TLBO algoritması için 20. iterasyon sonundaki sistem ve bileşen güvenilirlikleri aşağıdaki şekilde gösterilmiştir:

TLBO 20. iterasyon sonundaki bileşen ve sistem güvenilirlikleri (DAF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,804084	0,746183	0,554577	0,949365	0,330643	0,998546	0,310321	0,998808	0,644367	0,786635	0,733148
0,800786	0,746042	0,557883	0,949433	0,333503	0,993708	0,312876	0,998194	0,641494	0,788178	0,728104
0,803842	0,747055	0,554938	0,949838	0,329547	0,999467	0,309311	0,999630	0,644752	0,787076	0,735740
0,803283	0,745465	0,555756	0,949562	0,330512	0,998618	0,311029	0,998347	0,645231	0,786580	0,732922
0,803073	0,746660	0,555844	0,949403	0,331297	0,997002	0,310817	0,998916	0,643285	0,787399	0,732145
0,789860	0,748565	0,573241	0,953559	0,328499	0,991273	0,313809	0,999714	0,643071	0,795946	0,735435
0,804068	0,747138	0,554615	0,949883	0,329274	0,999931	0,309001	0,999773	0,644972	0,786963	0,736356
0,802096	0,745875	0,556965	0,949854	0,330906	0,997404	0,311332	0,998561	0,644346	0,787411	0,732502
0,803174	0,745898	0,555742	0,949720	0,330197	0,998864	0,310579	0,998734	0,645173	0,786782	0,733831
0,804812	0,747617	0,552812	0,948948	0,331453	0,997846	0,309317	0,999759	0,642442	0,786708	0,733003

TLBO algoritmasında 1. iterasyon sonunda en iyi sistem güvenilirlik değeri 0.518832 iken, 20. iterasyon sonunda sistem güvenilirlik değeri 0.736356 değerine yükselmiştir.



Şekil 5. 4. TLBO her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DAF/10/2)

Şekil 5.4'teki grafikte TLBO'da her iterasyon sonundaki en iyi sistem güvenilirlik değerleri verilmiştir. Görüldüğü gibi her iterasyon sonucunda sistem güvenilirlik değeri artarak en iyi sonucuna ulaşmıştır.

GA için ele alınan başlangıç bileşen güvenilirliklerine göre, sistem güvenilirliğinin en yüksek değeri, GA'da en iyi kromozom olarak kaydedilir. Daha sonra Şekil 4.13'te verilen genetik algoritma akış şemasındaki adımlar uygulandıktan sonra 1. ve 20. iterasyon sonundaki bileşen ve sistem güvenilirlikleri aşağıdaki gibidir:

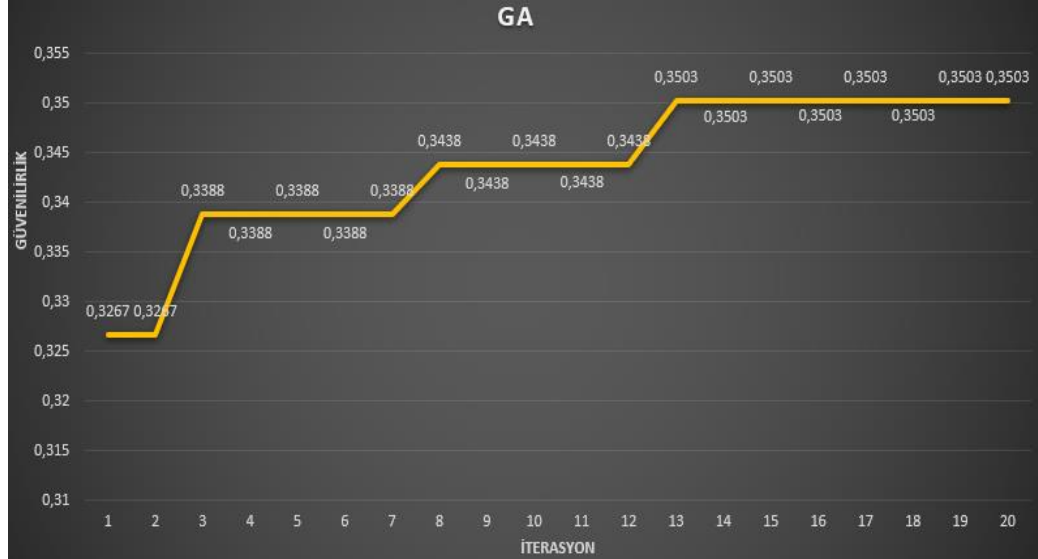
GA 1. iterasyon sonundaki sistem ve bileşen güvenilirlikleri (DAF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,326738
0,787814	0,529531	0,459164	0,718206	0,134419	0,921456	0,466229	0,714487	0,691321	0,905117	0,288138
0,405882	0,244154	0,885121	0,517032	0,731639	0,738188	0,864595	0,411887	0,738064	0,482645	0,249648
0,047987	0,803769	0,301624	0,266876	0,823950	0,679397	0,873601	0,770902	0,598669	0,985864	0,236740
0,699652	0,444155	0,699652	0,699652	0,699652	0,267142	0,699652	0,646386	0,155901	0,699652	0,163525
0,661753	0,558728	0,697705	0,304829	0,556989	0,505504	0,151534	0,649512	0,931877	0,381886	0,120006
0,742494	0,370786	0,207092	0,951109	0,663568	0,951759	0,507471	0,149387	0,204428	0,991631	0,071139
0,428517	0,068051	0,493814	0,434693	0,963063	0,467606	0,934832	0,750566	0,172260	0,162045	0,039061
0,970599	0,229230	0,209536	0,229554	0,846114	0,844191	0,089506	0,125205	0,629511	0,173823	0,010521
0,952521	0,523495	0,114473	0,003138	0,750896	0,323302	0,493768	0,506119	0,233175	0,722217	0,009981

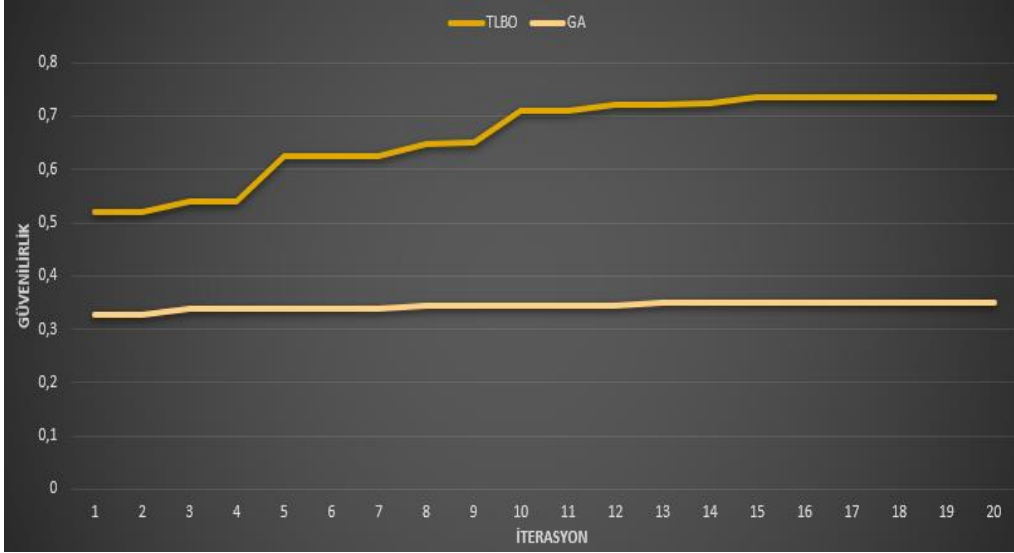
GA 20. iterasyon sonundaki sistem ve bileşen güvenilirlikleri (DAF/10/2)

$p1$	$p2$	$p3$	$p4$	$p5$	$p6$	$p7$	$p8$	$p9$	$p10$	$R(p)$
0,762958	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,747468	0,350311
0,700809	0,444155	0,800691	0,800691	0,800691	0,385267	0,797172	0,712574	0,385267	0,747468	0,343889
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,747468	0,338836
0,745881	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,697355	0,334498
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,663973	0,385267	0,697355	0,332262
0,700809	0,468345	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,747468	0,328040
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,697355	0,327725
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,697355	0,327725
0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,326738
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,697355	0,326479

GA'da 0.1 mutasyon oranı ve 0.7 çaprazlama oranına göre 1. iterasyon sonunda en iyi sistem güvenilirlik değeri 0.326738 iken, 20. iterasyon sonunda sistem güvenilirlik değeri 0.350311 değerine yükselmiştir. Sistem güvenilirlik değeri düşüken, bileşen güvenilirlik değerlerinin de daha düşük olduğu gözlemlenir. Şekil 5.5'de GA'da her iterasyon sonundaki en iyi sistem güvenilirlik değerleri grafikte verilmiştir. Şekil 5.6.'da ise TLBO ve GA yöntemlerindeki en iyi sistem güvenilirlik değerleri tek bir grafikte sunulmuştur. Grafiğe bakıldığında TLBO yönteminin, GA yöntemine göre 20. iterasyon sonunda daha yüksek sistem güvenilirlik değerine sahip olduğu görülmüştür.



Şekil 5. 5. GA her iterasyon sonucu en iyi sistem güvenilirlikleri grafiği (DAF/10/2)



Şekil 5. 6. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (DAF/10/2)

Popülasyon büyüklüğünü ve iterasyon sayısı 50'ye çıkarıldığında doğrusal ardışık 10'dan 2 çıkışlı F sisteminin TLBO ve GA yöntemiyle karşılaştırıldığı grafik aşağıdaki gibidir. GA yöntemi için çaprazlama ve mutasyon oranı sırasıyla 0.8 ve 0.2 alınmıştır.



Şekil 5. 7. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (50-50 DAF/10/2)

Şekil 5.7.'ye bakıldığında yine popülasyon büyüklüğü ve iterasyon sayısı artırıldığında hem TLBO yönteminde hem de GA yönteminde, popülasyon büyüklüğünün 10 ve iterasyon sayısının 20 olduğu duruma göre sistem güvenilirlik değerlerinin arttığı görülmektedir. TLBO yöntemi 1. iterasyonda 0.59078 değerinden, 50. iterasyon sonunda sistem güvenilirlik değerini 0.98706'ya çıkarmışken, GA yöntemi de 1. iterasyonda 0.3523 değerinden, sistem güvenilirliğini 0.56013'e kadar yükseltmiştir. 50. İterasyon sonunda TLBO

yönteminin, GA yöntemine göre daha yüksek sistem güvenilirlik değeri verdiği söylenebilir.

Popülasyon büyüklüğünün ve iterasyon sayısının 100 olduğu durumda doğrusal ardışık 10'dan 2 çıkışlı F sisteminin TLBO ve GA yöntemiyle karşılaştırıldığı grafik Şekil 5.8.'de verilmiştir. GA yöntemi için çaprazlama ve mutasyon oranı sırasıyla 0.9 ve 0.2 alınmıştır. Popülasyon büyüklüğü ve iterasyon sayısı artırıldığında hem TLBO yönteminde hem de GA yönteminde, popülasyon büyüklüğünün 10 ve 50 ve iterasyon sayısının 20 ve 50 olduğu duruma göre sistem güvenilirlik değerlerinin arttığını söyleyebiliriz. TLBO yöntemi 1. iterasyonda 0.5078 değerinden, 100. iterasyon sonunda sistem güvenilirlik değerini 0.99993'ya çıkarmışken, GA yöntemi de 1. iterasyonda 0.47826 değerinden, sistem güvenilirliğini 0.66388'e kadar yükseltmiştir. 100. iterasyon sonunda TLBO yönteminin, yine GA yöntemine göre daha yüksek sistem güvenilirlik değeri verdiği söylenebilir.



Şekil 5. 8. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (100-100 DAF/10/2)

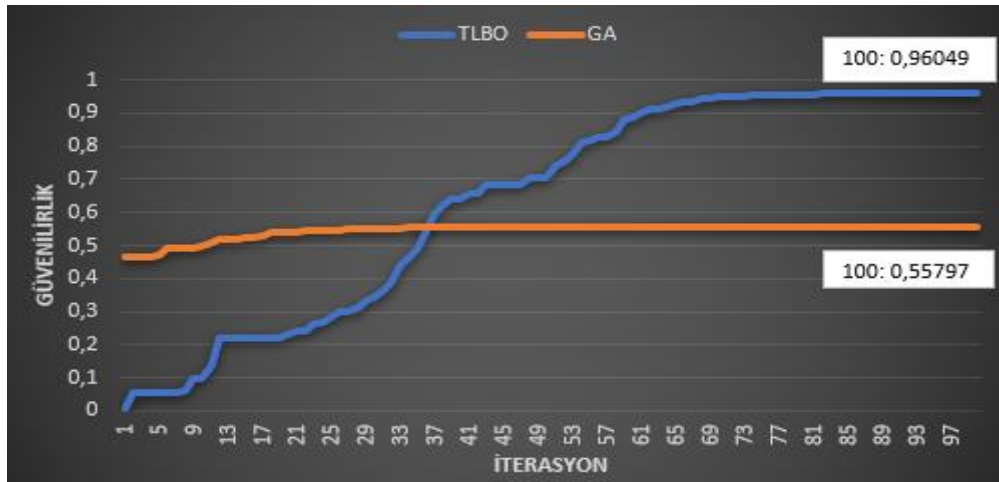
Popülasyon büyüklüğünün ve iterasyon sayısının 50 olduğu durumda doğrusal 10'dan 2 çıkışlı F sisteminin TLBO ve GA yöntemiyle karşılaştırıldığı grafik aşağıdaki gibidir. GA yöntemi için çaprazlama ve mutasyon oranı sırasıyla 0.8 ve 0.2 alınmıştır.



Şekil 5. 9. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (50-50 DF/10/2)

TLBO yöntemi 1. iterasyonda 0.091994 değerinden, 50. iterasyon sonunda sistem güvenilirlik değerini 0.89414'e çıkarmışken, GA yöntemi de 1. iterasyonda 0.23162 değerinden, sistem güvenilirliğini 0.39173'e kadar yükseltmiştir. 50. iterasyon sonunda TLBO yönteminin, GA yöntemine göre yine daha yüksek sistem güvenilirlik değeri verdiği söylenebilir. Aynı zamanda TLBO ve GA yöntemlerinin popülasyon büyüklüğü ve iterasyon sayısı 50 iken ardışık 10'dan 2 çıkışlı F sisteminin, doğrusal 10'dan 2 çıkışlı F sistemine göre daha yüksek sonuçlar verdiğini ve yöntemlere bakıldığında ise yine TLBO'nun daha iyi sonuçlar verdiğini görülmektedir.

Popülasyon büyüklüğünün ve iterasyon sayısının 100 olduğu durumda doğrusal 10'dan 2 çıkışlı F sisteminin TLBO ve GA yöntemiyle karşılaştırıldığı grafik aşağıdaki gibidir. GA yöntemi için çaprazlama ve mutasyon oranı sırasıyla 0.9 ve 0.2 olarak alınmıştır.



Şekil 5. 10. TLBO ve GA en iyi sistem güvenilirlik karşılaştırması grafiği (100-100 DF/10/2)

Şekil 5.10.'a bakıldığında TLBO yöntemi 1. iterasyonda 0.005873 değerinden, 100. iterasyon sonunda sistem güvenilirlik değerini 0.96049'a çıkarmışken, GA yöntemi de 1. iterasyonda 0.46566 değerinden, sistem güvenilirliğini 0.55797'e kadar yükseltmiştir. 100. iterasyon sonunda TLBO yönteminin, GA yöntemine göre yine daha yüksek sistem güvenilirlik değeri verdiği söylenebilir. Ayrıca popülasyon büyüklüğü ve iterasyon sayısı 100 iken TLBO ve GA yöntemlerinin genel olarak doğrusal ardışık 10'dan 2 çıkışlı F sisteminin, doğrusal 10'dan 2 çıkışlı F sistemine göre daha yüksek sonuçlar verdiğini ve yöntemlere bakıldığında ise yine TLBO'nun sistem güvenilirliği üzerinde daha fazla olumlu etkiye sahip olduğu söylenebilir.

Tablo 5. 2. TLBO ve GA yöntemlerine göre sistem karşılaştırmaları

	Popülasyon	İterasyon	Çaprazlama Oranı (GA)	Mutasyon Oranı (GA)	TLBO	GA
D/10/2/:F	10	20	0,7	0,1	0,375622	0,24550
	50	50	0,8	0,2	0,89414	0,39173
	100	100	0,9	0,2	0,96049	0,55797
D/10/2/:G	10	20	0,7	0,1	1,00000	1,00000
	50	50	0,8	0,2	1,00000	1,00000
	100	100	0,9	0,2	1,00000	1,00000
DA/10/2/:F	10	20	0,7	0,1	0,736356	0,350311
	50	50	0,8	0,2	0,98706	0,56013
	100	100	0,9	0,2	0,99993	0,66388
DA/10/2/:G	10	20	0,7	0,1	0,99995	0,99500
	50	50	0,8	0,2	1,00000	0,99911
	100	100	0,9	0,2	1,00000	0,99948
DA/10/3/:F	10	20	0,7	0,1	0,91405	0,79548
	50	50	0,8	0,2	0,99993	0,86531
	100	100	0,9	0,2	0,99999	0,96291
DA/10/3/:G	10	20	0,7	0,1	0,99985	0,97136
	50	50	0,8	0,2	0,99999	0,99227
	100	100	0,9	0,2	1,00000	0,99987

Tablo 5.2.'ye bakıldığında doğrusal 10'dan 2 çıkışlı F sistemi ve doğrusal ardışık 10'dan 2 çıkışlı F sisteminin farklı popülasyon büyüklüklerinin ve farklı iterasyon sayısı, çaprazlama oranı ve mutasyon oranlarındaki TLBO ve GA optimizasyon yöntemleri sonuç karşılaştırmaları verilmiştir. Aynı zamanda k değeri 3 olarak alındığında doğrusal ardışık 10'dan 3 çıkışlı F sisteminin farklı popülasyon büyüklükleri ve farklı iterasyon sayısı, çaprazlama oranı ve mutasyon oranlarındaki TLBO ve GA optimizasyon yöntemleri ile doğrusal ardışık 10'dan 2 çıkışlı F sisteminin karşılaştırmaları da sunulmuştur. Ayrıca doğrusal 10'dan 2 çıkışlı F sistemi ve doğrusal ardışık 10'dan 2 çıkışlı F sistemlerinin duali olan doğrusal 10'dan 2 çıkışlı G sistemi ve doğrusal ardışık 10'dan 2 çıkışlı G sistemleri içinde MATLAB program çıktıları tabloda sunulmuştur.

6. SONUÇ

Bu çalışma da doğrusal ardışık 10'dan 2 çıkışlı bir F sistemi ile doğrusal 10'dan 2 çıkışlı bir F sisteminin güvenilirlikleri verilen tanımlamalar altında maksimize edilmiş ve sistem güvenilirliğinin en yüksek değerini veren bileşen güvenilirlikleri TLBO ve GA yöntemleri ile hesaplanarak karşılaştırılmıştır. Farklı popülasyon büyüklükleri ve farklı iterasyon değeriyle birlikte hesaplanan sistem ve bileşen güvenilirliklerinin, GA yöntemine göre TLBO'da daha yüksek sistem güvenilirlik değerleri verdiği görülmüştür. Aynı zamanda GA yöntemi için literatürde yer alan çaprazlama ve mutasyon oranları da dikkate alınmıştır. Doğrusal 10'dan 2 çıkışlı F sistemiyle doğrusal ardışık 10'dan 2 çıkışlı F sistemlerinin her iki yöntemde de sonuçlarına bakıldığında doğrusal ardışık 10'dan 2 çıkışlı F sistemin daha yüksek sistem güvenilirlik değerleri verdiği görülmüştür. Bunun sebebi 10 taneden ardışık olarak iki tane bileşenin bozulması olasılığının, 10 tane bileşenden rasgele iki tanesinin bozulması olasılığından daha düşük olması şeklinde yorumlanabilir. Doğrusal ardışık 10'dan 2 çıkışlı F sistemi ile doğrusal ardışık 10'dan 3 çıkışlı bir F sistemi karşılaştırılması da yapılmıştır. Karşılaştırma sonucunda doğrusal ardışık 10'dan 3 çıkışlı F sisteminin güvenilirlik değerlerinin her iki yöntemde de doğrusal ardışık 10'dan 2 çıkışlı F sistemine göre daha yüksek olduğu gözlemlenmiştir. Bunun nedeninin üç tane bileşenin art arda bozulması durumunun, iki tane bileşenin art arda bozulması durumuna göre daha düşük olasılığa sahip olması şeklinde yorumlanabilir. Aynı zamanda doğrusal ardışık 10'dan 2 ve 3 çıkışlı bir G sistemi ile doğrusal 10'dan 2 çıkışlı bir G sistemi de her iki yöntemde hesaplanarak Tablo 5.2.'de sunulmuştur. G sistemlerinin, sistem güvenilirliklerine bakıldığında 1 ve 1'e çok yakın sonuçlar çıktığı görülmektedir. Bunun sebebi genel olarak k değeri ile ilişkilidir. G sistemleri için söz konusu olan en az k ya da daha fazla bileşen başarılı bir şekilde çalışırsa sistem de başarılı olur tanımından sistem de 10 tane bileşenden 2-3 tane bileşenin başarılı olması olasılığının çok yüksek olması şeklinde yorumlanabilir. Genel olarak yöntemlere baktığımızda ise TLBO yönteminin, GA'ya göre daha kısa sürede optimum değerlere ulaştığını ve genel sistem güvenilirliği üzerinde daha fazla olumlu bir etkiye sahip olduğunu söylemek mümkündür. Her iki yöntem de doğadan esinlenerek tasarlanmış meta-sezgisel yöntemler olup problemin çözümü için tüm çözüm kümesini taramaktan ziyade istenilen amaca uygun en iyi cevabı

vermelerine rağmen TLBO yönteminin, GA yöntemine göre çalışmayı incelediğimizde daha fazla avantajlı olduğunu söyleyebiliriz. Sebebi ise TLBO yönteminin, GA'ya göre daha anlaşılabilir ve daha kolay işlem basitliğine sahip olmasıdır. GA, diğer yönteme kıyasla çok daha fazla kısıta sahiptir ve göz önünde bulundurulması gereken birçok kriter içerir.

Sonuçta, seri ve paralel sistemlerin özel durumları olarak ele alınan doğrusal ardışık n 'den k çıkışlı sistemler ve doğrusal n 'den k çıkışlı sistemlerin güvenilirlik optimizasyonu problemlerinde, sistemlerin karmaşık bir yapıya sahip olmasından dolayı çözüm uzayındaki gerçek cevaba en yakın, muhtemel en iyi çözümün bulunmasında meta-sezgisel algoritmaların kullanımıyla zaman ve maliyet açısından daha fazla fayda sağladığı söylenebilir.

KAYNAKLAR DİZİNİ

- Ağalday, M.F.**, 2018, Genetik Algoritma ile Merkezi Sınavlarda Tek ve Çok Boyutlu Yakınlığa Göre En İyi Oturum Planının Oluşturulması, Yüksek Lisans Tezi, Bilgisayar Mühendisliği, 1-84s.
- Akkoyunlu, M.C. ve Engin, O.**, 2011, Kesikli harmoni arama algoritması ile optimizasyon problemlerinin çözümü: literatür araştırması, *S.Ü. Mühendislik ve Mimarlık Fakültesi Dergisi*, 26(4): 140-148 s.
- Amrutkar, K.P., & Kamalja, K.K.**, 2017, An Overview of Various Importance Measures of Reliability System, 2(3), 150-171p.
- Ardakan, M.A., & Dobani, A.R.**, 2019, RRAP-CM: A new reliability-redundancy allocation problem with heterogeneous components, *Reliability Engineering and System Safety*, 191, 1-11p.
- Artar, M.**, 2016, Öğretme-Öğrenme Esaslı Optimizasyon Yöntemi ile Uzay Kafes Kule Yapı Sisteminin Optimum Boyutlandırılması, Dicle Üniversitesi Mühendislik Fakültesi, 7(3), 471-480s.
- Bağış, A. ve Özçelik, Y.**, 2007, Gerçek Kodlu Genetik Algoritma Kullanarak Sistem Kimliklendirme, 12. Elektrik- Elektronik- Bilgisayar- Biyomedikal Mühendisliği Ulusal Kongresi, 1-6s.
- Bolat, B., Erol, K.O. ve İmrak, C.E.**, 2004, Genetic Algorithms in Engineering Applications and the Function of Operators, *Mühendislik ve Fen Bilimleri Dergisi*, 264-271s.
- Bugatekin, A.T.**, 2017, Homojen Olmayan Poisson Süreci ile Bir Makinenin Güvenilirliğinin Test Edilmesi, Fırat Üniversitesi, İstatistik Bölümü, 29(1), 207-210 s.
- Cao, R., Hou, W. and Gao, Y.**, 2017, An Entropy-Based Three-Stage Approach for Multi-Objective Systems Reliability Optimization Considering Uncertainty, *Engineering Optimization*, 50(9), 1453-1469p.

KAYNAKLAR DİZİNİ (Devam)

- Coit, D.W. And Liu, J.**, 2000, System Reliability Optimization with k -out-of- n Subsystems, *International Journal of Reliability*, 7(2), 129-142p.
- Coit, D.W.**, 2011, System Reliability Optimization with k -out-of- n Subsystems and Changing k , *Department of Industrial Engineering*, 1382-1387p.
- Coit, D.W., and Zio, E.**, 2019, The evolution of system reliability optimization, *Reliability Engineering and System Safety*, 192, 1-11p.
- Çakar. K.**, 2009, Genetik Algoritmalar Yardımıyla Acil Servis İstasyonu Yerleşiminin Optimizasyonu, Doktora Tezi, Sosyal Bilimler Enstitüsü, 1-125s.
- Çalışkan, F., Yüksel, H. ve Dayık, M.**, 2016, Genetik Algoritmaların Tasarım Sürecinde Kullanılması, *SDU Teknik Bilimler Dergisi*, 6(2), 21-27 s.
- Çavuş, F. ve Yanıkoğlu, E.**, 2003, Karmaşık Sistemlerin Monte Carlo Yöntemi ile Güvenilirlik Analizi, 7(3), 99-102 s.
- Çelik, Y., Yıldız, İ. ve Karadeniz, A.T.**, 2019, Son Üç Yılda Geliştirilen Meta-Sezgisel Algoritmalar Hakkında Kısa Bir İnceleme, *Avrupa Bilim ve Teknoloji Dergisi*, 463-477 s.
- Demiralp, A.**, 2020, n 'den ardıl k Çıkışlı Sistemlerin Güvenilirliği İçin Sınır Yaklaşımları, Doktora Tezi, Sosyal Bilimler Enstitüsü, 1-69 s.
- Demiralp, A. ve Güngör, M.**, 2021, Ardıl- k Sistemler için Önerilen Güvenilirlik Sınır Karşılaştırılması, *BEÜ Fen Bilimleri Dergisi*, 10(3), 877-890s.
- Du, X. and Chen, W.**, 2002, Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design, *Design Engineering Technical Conferences and Computer and Information in Engineering Conference*, 1-10p.
- Emel, G.G. ve Taşkın, Ç.**, 2009, Sayısal Yöntemlerde Genetik Algoritmalar, Alfa Aktüel Yayınları, 1, Bursa, 153s.

KAYNAKLAR DİZİNİ (Devam)

- Erdem, İ.**, 2012, Matematiksel İstatistik Problemler ve Çözümleri, 2 (22), Ankara, 454s.
- Erçalık, H.**, 2012, Elektrik Dağıtım Sistemlerinin Güvenilirlik Analizi, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, 3-4 s.
- Fyffe, D.E., Hines, W.W. and Lee, N.K.**, 1968, System Reliability Allocation and a Computational Algorithm, *IEEE Transactions on Reliability*, 17(2), 64-69p.
- Geo, W.J., Xing, B., and Marwala T.**, 2013, Teaching-Learning-based Optimization Approach for Enhancing Remain Facturability Pre-Evaluation System's Reliability, *IEEE Symposium on Swarm Intelligence (SIS)*, 235-239p.
- Gökdere, G. ve Güral, Y.**, 2016, Mühendislik Uygulamalarında Kullanılan Ardışık n'den k Çıkışlı Sistemlerin Güvenilirlik Analizi, *Mühendislik Bilimi Dergisi*, 461-467s.
- Gökdere, G. ve Güral, Y.**, 2018, Birnbaum önem tabanlı genetik algoritma ve doğrusal ardışık n -den k -çıkışlı sistemlerin optimizasyonunda uygulaması, *BEÜ Fen Bilimleri Dergisi* 7(2): 276-283 s.
- Gürsoy, E.**, 2006, n'den k Çıkışlı Sistemlerin Güvenilirliği (Reliability of k-out-of-n systems), Yüksek Lisans Tezi, Uygulamalı İstatistik Anabilim Dalı, 1-27 s
- Kaya, V.**, 2015, n'den k çıkışlı Tamir Edilemez Sistemlerin Güvenilirliği Üzerine, Yüksek Lisans Tezi, Sosyal Bilimler Enstitüsü, 1-74 s.
- Kızılok, S. ve Alataş, B.**, 2012, Sosyal Tabanlı Güncel Sezgisel Optimizasyon Algoritmaları, *C.Ü. İktisadi ve İdari Bilimler Dergisi*, 13(2), 1-18s.
- Köse, U.**, 2017, Yapay Zekâ Tabanlı Optimizasyon Algoritmalarının Geliştirilmesi, Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı, 1-185s.

KAYNAKLAR DİZİNİ (Devam)

- Kuo W., Zhang W., Zuo M.**, 1990, A consecutive- k -out-of- n : G system: The mirror image of a consecutive- k -out-of- n : F system, *IEEE Transactions on Reliability*, R-39 (2): 244-253p.
- Kuo, W. and Prasad, R.**, 2000, An Annotated Overview of System-Reliability Optimization, *IEEE*, 49(2), 176-185p.
- Kuo, W., & Zuo, M. J.**, 2003, Optimal Reliability Modeling: Principles and Applications, John Wiley & Sons, Inc.
- Kumar. R., Tewari. P.C. and Khanduja. D.**, 2018, Parameters Optimization of Fabric Finishing System of a Textile Industry Using Teaching-Learning-based Optimization Algorithm, *International Journal of Industrial Engineering Computations*, 221-234p.
- Levitin, G.**, 2005, The Universal Generating Function in Reliability Analysis and Optimization, The State University of New Jersey 96 Frelinghuysen Road Piscataway.
- Metas, A.**, 2000, Reliability Allocation and Optimization for Complex Systems, *Electrical and Electronics Engineers*, 1-6p.
- Onishi, J., Kimura, S., James, R.J.W. and Nakagawa, Y.**, 2007, Solving the Redundancy Allocation Problem With a Mix of Components Using the Improved Surrogate Constraint Method
- Öztürk, A.**, 2002, Gerçel Sayı Kodlamalı Genetik Algoritmanın Optimizasyonunda Kullanımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2-12 s.
- Özkılıç, Ö. 2016.** “Güvenilirlik Nedir? (Dependability/Reliability) (Reliability Centred Maintenance-RCM)”
<https://www.linkedin.com/pulse/g%C3%BCvenilirlik-nedirdependabilityreliability-centred-rcm-%C3%B6zkili%C3%A7> (Erişim Tarihi: 21 Aralık 2020)

KAYNAKLAR DİZİNİ (Devam)

- Paksoy, S. ve Uzun, A.**, 2008, Genetik Algoritma ile Kaynak Kısıtlı Proje Çizelgeleme, *Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi*, 17(2), 345-362s.
- Rao, R.V., Savsani, V.J. and Vakharia, D.P.**, 2012, Teaching–Learning-based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems, *Information Sciences*, 183, 1-15s.
- Saleem, E.A.**, 2017, Series-Parallel Systems Reliability Optimization Using Genetic Algorithm and Statistical Analysis, *International Journal of Mechanical and Mechatronics Engineering*, 11(5), 960-966p.
- Satman, M.H.**, 2016, Genetik Algoritmalar, Türkmen Kitabevi, İstanbul, 264s.
- Si, S., and Cai, Z., Sun, S. and Li, C.**, 2016, Optimization of linear consecutive-k-out-of-n system with a Birnbaum importance-based genetic algorithm, *Reliability Engineering and System Safety*, 152, 248-258p.
- Soltani, R.**, 2014, Reliability Optimization of Binary State Non-Repairable Systems: A State of The Art Survey, *International Journal of Industrial Engineering Computations*, 339-364p.
- Tan, R. ve Bora, Ş.**, 2017, Modelleme ve benzetim ortamında parametre optimizasyonu ve kullanılan teknikler, *Mühendislik Bilimleri ve Tasarım Dergisi*, 5(3): 685 – 697 s.
- Ulucenk, Ç.**, 2009, Implementation of Reliability Based Design Optimization Techniques for Aerospace Structures, Master Thesis, Computational Science and Engineering, 1-75p.
- Uzun, A. ve Özdoğan, A.**, 2011, Güvenilirlik Analizlerine Dayalı Önleyici Bakım Planlaması, 20(1), 303-320 s.
- Uzuner, M.**, 2015, Bir İşletmede Güvenilirlik Analizlerine Dayalı Sistem Tasarımı, Yüksek Lisans Tezi, Başkent Üniversitesi Fen Bilimleri Enstitüsü, 9-10 s.
- Varan, S.E.**, 2007, Sistem Güvenilirliği Analizi ve Bir Uygulama, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü, 1s.

KAYNAKLAR DİZİNİ (Devam)

- Youn, B.D., Choi, K.K. and Du, L.,** 2005, Enriched Performance Measure Approach for Reliability-Based Design Optimization, *AIAA Journal*, 43(4), 1-11p.
- Wang, N., Zhao, J., Jiang, Z. and Zhang, S.,** 2018, Reliability Optimization of Systems with Component Improvement Cost based on Importance Measure, *Mechanical Engineering*, 10(11), 1-15p.
- Zuo, M.J., and Kuo W.,**1990, Design and Performance Analysis Of Consecutive- k -out-of- n Structure, *Naval Research Logistics*, 37: 203-230p.
- Zhao, J.B., Zhang, S., Zhu, W.J. and Du, Li.,** 2019, Reliability Optimization of Lineer Consecutive k -out-of- n : F systems with Birnbaum Importance-based Quantum Genetic Algorithm, *Mechanical Engineering*, 11(4), 1-16p.

TEŞEKKÜR

Tez sürecim boyunca değerli görüşlerinden yararlandığım ve çok kıymetli bilgilerini her an benimle paylaşan danışmanım Sayın Dr. Öğretim Üyesi Özge ELMASTAŞ GÜLTEKİN'e, aynı zamanda değerli bilgi ve deneyimleriyle bu zamana kadar İstatistik Bilimine katkı sağlayan ve hiçbir zaman yardımlarını eksik etmeyen tüm bölüm hocalarıma da en içten duygularıyla teşekkürü bir borç bilirim.

Ayrıca eğitim ve öğretim hayatım süresince her türlü desteği ve imkânı bana sağlayan aileme, çalışmamın arka planındaki yazılım sürecinde kıymetli görüşlerinden faydalandığım arkadaşlarıma da en içten teşekkürlerimi sunarım.

21/01/2022

Şefika Büşra MERT

ÖZGEÇMİŞ

Şefika Büşra MERT, 2018 yılında Dokuz Eylül Üniversitesi Fen Fakültesi İstatistik Bölümünden başarıyla mezun olmuştur. 2019 yılında Ege Üniversitesi Fen Bilimleri Enstitüsü İstatistik Bölümünde tezli yüksek lisansa başlayıp yoğun bir çalışma süresi içerisinde tezini tamamlayabilmiştir.

10 Ocak 2022 tarihinde Aydem & Gediz Elektrik Perakende Satış A.Ş.'de, İç Kontrol Yöneticiliği biriminde, İç Kontrol Uzman Yardımcısı olarak göreve başlamıştır.

