

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ DOKTORA PROGRAMI**

**ARIZA VE TAMİR DURUMUNDA SİSTEM GÜVENİLİRLİĞİ:
GENETİK VE MEMETİK ALGORİTMALAR**

HAZIRLAYAN

MERVE UZUNER ŞAHİN

DOKTORA TEZİ

ANKARA – 2021

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
ENDÜSTRİ MÜHENDİSLİĞİ DOKTORA PROGRAMI**

**ARIZA VE TAMİR DURUMUNDA SİSTEM GÜVENİLİRLİĞİ:
GENETİK VE MEMETİK ALGORİTMALAR**

HAZIRLAYAN

MERVE UZUNER ŞAHİN

DOKTORA TEZİ

TEZ DANIŞMANI

PROF. DR. BERNA DENGİZ

EŞ TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ ORHAN DENGİZ

ANKARA – 2021

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Endüstri Mühendisliği Anabilim Dalı “Endüstri Mühendisliği” Doktora Programı çerçevesinde Merve Uzuner Şahin tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Doktora Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 04 / 08 / 2021

Tez Adı: Arıza ve Tamir Durumunda Sistem Güvenilirliği: Genetik ve Memetik Algoritmalar

Tez Jüri Üyeleri (Unvanı, Adı - Soyadı, Kurumu)		İmza
Prof. Dr. Berna Dengiz	Başkent Üniversitesi	
Prof. Dr. Fulya Altıparmak	Gazi Üniversitesi	
Prof. Dr. Çiğdem Alabaş Uslu	Marmara Üniversitesi	
Prof. Dr. Yusuf Tansel İç	Başkent Üniversitesi	
Dr. Öğr. Üyesi Burçin Çakır Erdener	Başkent Üniversitesi	

ONAY

Prof. Dr. Faruk Elaldı

Fen Bilimleri Enstitüsü Müdürü

Tarih : ... / ... /

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 10 / 08 / 2021

Öğrencinin Adı, Soyadı : Merve Uzuner Şahin

Öğrencinin Numarası : 21420206

Anabilim Dalı : Endüstri Mühendisliği

Programı : Endüstri Mühendisliği

Danışmanın Unvanı/Adı, Soyadı : Prof. Dr. Berna Dengiz

Dr. Öğr. Üyesi Orhan Dengiz

Tez Başlığı : Arıza ve Tamir Durumunda Sistem Güvenilirliği: Genetik ve Memetik Algoritmalar

Yukarıda başlığı belirtilen Doktora tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 99 sayfalık kısmına ilişkin, 10 / 08 / 2021 tarihinde şahsım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %5'dir. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:.....

ONAY

Tarih: ... / ... / 20...

Prof. Dr. Berna Dengiz

.....

TEŞEKKÜR

Yazar, bu çalışmanın gerçekleşmesinde katkılarından dolayı, aşağıda adı geçen kişi ve kuruluşlara içtenlikle teşekkür eder.

Sayın Prof. Dr. Berna Dengiz'e (tez danışmanı), başta bu süreçte gösterdiği tüm anlayış, sabır ve hoşgörüsü olmak üzere, çalışmanın sonuca ulaştırılmasında ve karşılaşılan güçlüklerin aşılmasında her zaman yardımcı ve yol gösterici olduğu için,

Sayın Dr. Orhan Dengiz'e, (eş tez danışmanı) deneyim ve bilgileri ile her zaman yol gösterici olduğu ve çalışmanın sonuca ulaştırılmasında sağladığı tüm katkılar için,

Sayın Prof. Dr. Fulya Altıparmak ve Sayın Prof. Dr. Yusuf Tansel İç'e (TİK üyeleri), tezimin her aşamasında desteklerini esirgemedikleri için,

Hayatımın her döneminde yanımda olan, maddi ve manevi her türlü desteği sonsuz sevgileri ile veren canım ailem, annem Gülten UZUNER, babam Hasan UZUNER ve kardeşim Meltem UZUNER'e,

Bu süreçte desteğini hep yanımda hissettiğim, hayatı değerli kılan biricik eşim Mustafa Taylan ŞAHİN'e,

Varlığıyla bana en büyük hediye, canım kızım Masal'ıma, içtenlikle teşekkürlerimi sunarım.

ÖZET

Merve UZUNER ŞAHİN

**ARIZA VE TAMİR DURUMUNDA SİSTEM GÜVENİLİRLİĞİ: GENETİK VE
MEMETİK ALGORİTMALAR**

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

2021

İleri teknolojiye sahip sistemlerin, örneğin iletişim, elektronik, uzay sistemleri gibi modern mühendislik sistemlerinin, giderek karmaşıklaşan yapısı ve işleyiş süreçleri, bu sistemlerin daha etkin bir şekilde değerlendirilmesinin önemini artırmıştır. Karmaşık yapıdaki bu tür sistemlerin tasarımı ve analizi ile ilgili çalışmalarda başarımlarının değerlendirilmesi için alan yazında yaygın olarak kullanılan ölçütlerinden birisi sistem güvenilirliğidir. Sistem güvenilirliği kavramı genel anlamda bir sistemin amacına uygun olarak çalışmasını ifade etmektedir. Problem türlerine bağlı olarak tanımlanan sistem kısıtları altında sistem güvenilirliğinin eniyilenmesi önemli ve her zaman araştırmacıların ilgisini çeken bir problemidir.

Alan yazında yaygın olarak kullanılan Yedek Bileşen Tahsis Problemi (YBTP), sistemde yer alan bileşenlere paralel olarak yerleştirilen, kullanıma hazır bileşenler ile daha yüksek güvenilirliğe sahip yeni sistemlerin tasarlanması olarak tanımlanabilir. YBTP NP-zor bir problemidir. YBTP ile ilgili çalışmaların çoğunda genellikle yalnızca arızalar dikkate alınırken, çok az sayıdaki çalışmada arıza ve tamir durumunun sabit arızalanma oranlarıyla dikkate alındığı görülmüştür. Oysa, gerçek hayat problemlerinde sistemi oluşturan bileşenler kullanımdan kaynaklanan yıpranmaya bağlı olarak artan arızalanma oranına sahiptir. Diğer yandan alan yazındaki mevcut YBTP çalışmalarında, hesaplama zorluğu nedeniyle, genellikle bileşenlerin arızalandıktan sonra tamir edilmediği ve dolayısıyla devre dışı kaldığı varsayımı kullanılmaktadır. Doğal olarak bu varsayım sistemlerin ilk tasarımlarında maliyeti artırmaktadır. Gerçek hayatta kullanılan sistemlerde ise, arızalanan bir bileşen sistemin güvenilirliği negatif yönde etkilerken, tamir edilen bileşen sistemin güvenilirliğini pozitif yönde etkilemektedir. Gerçek hayatta kullanılmakta olan bir sistemin

bu doğal yapısı dikkate alınarak yüksek güvenilirliğe sahip tasarımlar daha düşük maliyetle elde edilebilir. Ayrıca, YBTP’de güvenilirlik kısıtı olarak kullanılan, n ’den k ’lı (k -out-of- n) varsayımında “ k ” parametresi değişen servis sağlayıcı taleplerine bağlı olarak değişkenlik göstermelidir. Bu nedenle YBTP’de “ k ” parametresinin değişiminin de dikkate alındığı dönemsel talep değişimlerini karşılayacak sistem tasarımlarının oluşturulması önemlidir. Bu tez çalışmasında, artan arızalanma oranları ile ortaya çıkan bozulmaların, tamirlerin ve değişken “ k ” değerinin dikkate alındığı durumda YBTP’nin eniyilenmesi için Genetik Algoritma (GA) ve Memetik Algoritma (MA) olmak üzere iki metasezgisel algoritma önerilmiştir. Sistem güvenilirliği (amaç fonksiyonu) gerçekçi bir yaklaşımla olayların çıkışına dayalı olarak geliştirilen kesikli olaylı benzetim (KOB) modelinin kullanıldığı kesikli olaylı Monte Carlo benzetimi ile tahmin edilmiştir. KOB modelinin geçerliliği ve geliştirilen GA ve MA’nın etkinlikleri alan yazında yaygın olarak kullanılan test problemleri üzerinde gösterilmiştir. Sayısal analiz sonuçlarına göre, GA ve MA’nın kaliteli çözümler bulduğu; MA’nın GA’ya göre daha iyi sonuçlar verdiği gösterilmiştir. Böylece, arıza ve bozulmanın dikkate alındığı YBTP için daha yüksek güvenilirliğe sahip sistem tasarımlarının daha düşük maliyetle elde edilebildiği probleme özgü iki algoritma, GA ve MA, ilgili alan yazına kazandırılmıştır.

ANAHTAR KELİMELEER: Sistem Güvenilirliği, Eniyileme, Yedek Bileşen Tahsis Problemi, Artan Arızalanma Oranı, Tamir Edilebilir Bileşen, Kesikli Olay Benzetimi, Genetik Algoritma, Memetik Algoritma.

ABSTRACT

Merve UZUNER ŞAHİN

SYSTEM RELIABILITY CONSIDERING COMPONENT FAILURE AND REPAIRS: GENETIC AND MEMETIC ALGORITHMS

Başkent University Institute of Science

Department of Industrial Engineering

2021

Due to the increasing complexity of structures and functions of high-tech systems, such as modern engineering and service systems in the communication, electronics, and space technology fields, the need for effective evaluation tools for such systems has increased in parallel. In the analysis and design of complex systems, the system reliability is one of the performance criteria commonly used in the literature to evaluate performance. The concept of system reliability generally refers to the operation of a system in accordance with its purpose. The reliability optimization of a system with various problem-specific constraints is an important and current problem.

The Redundancy Allocation Problem (RAP), which is widely used in the literature, can be defined as the design of new systems with higher reliability using redundant components in a parallel arrangement. The RAP is an NP-hard problem. While most of the studies on RAP generally only consider failures, few studies are based on the assumption that failures and/or repairs occur at constant rates. However, in real-life problems, the components in the system have an increased failure rate due to wear and tear from regular use. Due to the computational difficulty in RAP studies, it is generally assumed that components are not repaired after they fail and thus become out of order. Without an opportunity to repair a failed part, the initial cost of the system increases. Under such circumstances, the system has to be designed with better components (i.e., highly reliable but expensive) to maintain the overall system reliability at the desired level.

In real-life applications, a component failure has a negative impact on system reliability. After the failed component is repaired and put back to work, the reliability of the system

increases. Structures designed with repair considerations are more realistic, and high reliability can be obtained at a lower cost by using this approach.

In addition, the “ k ” parameter in the metrics (k -out-of- n) is dynamically adjusted according to changing user requirements. The (k -out-of- n) metric is used as a reliability constraint in RAP, and the introduction of a dynamic “ k ” facilitates the design of adoptable systems to seasonal changes in demand.

In this thesis, two metaheuristic algorithms, Genetic Algorithm (GA) and Memetic Algorithm (MA) are proposed to optimize RAP with increasing failure rates, component repairs, and dynamic “ k ”. The system reliability (objective function value) in the RAP optimization is estimated with a discrete event Monte Carlo simulation with discrete event simulation (DES) model, which is developed according to the occurrence of events with a realistic approach. The validity of the DES model has been demonstrated on the test problems given in the literature. The effectiveness of the developed GA and MA is shown on the test problems that are widely used in the literature. According to the computational analysis, GA and MA found quality solutions. It has been shown that MA gives better results than GA. Thus, problem-specific GA and MA for RAP designs with higher reliability at lower cost, where failure and repair are taken into account, have been developed and brought to the relevant literature.

KEYWORDS: System Reliability, Optimization, Redundancy Allocation Problem, Increasing Failure Rate, repairable Component, Discrete Event Simulation, Genetic Algorithm, Memetic Algorithm.

İÇİNDEKİLER

TEŞEKKÜR	i
ÖZET	ii
ABSTRACT	iv
TABLOLAR LİSTESİ	viii
ŞEKİLLER LİSTESİ	ix
SİMGELER VE KISALTMALAR LİSTESİ	xi
1. GİRİŞ	1
2. SİSTEM GÜVENİLİRLİĞİ	7
2.1 Temel Kavramlar	7
2.1.1 Güvenilirlik	7
2.1.2 Başarısızlık fonksiyonu	7
2.1.3 Arıza-tamir süreci	8
2.2 Sistem Güvenilirliği Hesaplama Yöntemleri	9
2.2.1 Sistem yapı fonksiyonu	9
2.2.1.1 Seri sistemler	10
2.2.1.2 Paralel sistemler	10
2.2.1.3 Paralel-Seri sistemler	10
2.2.1.4 Seri-Paralel sistemler	11
2.2.1.5 Karma seri-paralel sistemler	11
2.2.1.6 Köprü türü sistemler	11
2.2.1.7 Ardışık n 'den k 'lı (k -out-of- n : F) sistemler	11
2.2.1.8 n 'den k 'lı (k -out-of- n : F) sistemler	11
2.2.1.9 n 'den k 'lı (k -out-of- n : G) sistemler	12
2.2.2 Sistem imzası	12
2.2.3 Yaşam imzası	12
2.2.4 Benzetim	13
2.3 Sistem Güvenilirliğinin Eniyilenmesi Yaklaşımları	14
2.4 Yedek Bileşen Tahsis Problemi Türleri ve Özellikleri	16
3. ALAN YAZIN TARAMASI	21
4. ARIZA VE TAMİRİN DİKKATE ALINDIĞI YEDEK BİLEŞEN TAHSİS PROBLEMİ İÇİN ÖNERİLEN SEZGİSEL ALGORİTMALAR	29
4.1 Problem Tanımı ve Varsayımlar	29
4.2 Geliştirilen Genetik Algoritma	31
4.2.1 Dizi Gösterimi	32
4.2.2 Başlangıç yığınının oluşturulması	33
4.2.3 Amaç fonksiyonu	33

4.2.3.1	Sistem güvenilirliğinin tahmini	33
4.2.3.2	KOB Modelinin doğrulanması ve geçerliliği	35
4.2.3.3	Kesikli Olaylı Monte Carlo Benzetiminin deneme sayısının belirlenmesi	37
4.2.4	Ceza fonksiyonu	38
4.2.5	Değerlendirme fonksiyonu	40
4.2.6	Seçim mekanizması ve genetik operatörler	40
4.3	Geliştirilen Memetik Algoritma	46
5.	SAYISAL ANALİZLER	50
5.1.	Test Problemleri	50
5.2.	GA Parametrelerinin Belirlenmesi	51
5.3.	GA'nın Doğrulanması ve Geçerliliği	53
5.4.	Başarım Ölçütleri	54
5.5.	Sayısal Sonuçların Değerlendirilmesi	55
5.5.1.	Sabit “ k ” değerine sahip sistemler için analiz sonuçları	55
5.5.1.1.	Tamir edilebilen bileşenlerin kullanılmasının sistem güvenilirliğine ve maliyetine etkisi	55
5.5.1.2.	GA ve MA algoritmalarının karşılaştırılması	59
5.5.2.	Değişken “ k ” değerine sahip sistemler için analiz sonuçları	64
6.	SONUÇ ve ÖNERİLER	68
	KAYNAKLAR	71
	EKLER	
	EK 1a: Örnek sistem tasarımları ve tahmini sistem güvenilirliği değerleri	
	EK 1b: Tahmini sistem güvenilirliği kesikli olaylı Monte Carlo benzetimi sonuçları	
	EK 2: Farklı problemler için tamir edilebilen bileşenlerin kullanımının sistemin kurulum maliyetine etkisi	
	EK 3: P_{kd2} ve P_{kd3} problemleri için oluşturulan değişken “ k ” değerleri	

TABLÖLER LİSTESİ

	Sayfa
Tablo 4.1. 25 test problemi için %95 güvenlik düzeyinde güven aralığı değerleri.....	36
Tablo 5.1. Test probleminin özellikleri.....	50
Tablo 5.2. Test probleminin özellikleri.....	51
Tablo 5.3. Deney tasarımı faktörleri ve düzeyleri.....	52
Tablo 5.4. Varyans analizi sonuçları.....	52
Tablo 5.5. Duncan test sonuçları.....	53
Tablo 5.6. GA Geçerliliği için sonuçlar.....	54
Tablo 5.7. Tamir edilebilen bileşen kullanmanın sistem güvenilirliği üzerindeki etkisi....	57
Tablo 5.8. Tamir edilebilen bileşen kullanmanın sistem maliyeti üzerindeki etkisi.....	58
Tablo 5.9. GA ve MA sonuçları.....	59
Tablo 5.10 (a-b). GA ve MA sistem tasarımları ve sonuçları.....	62
Tablo 5.11. P_{kd1} problemine ait değişken “k” değerleri.....	64
Tablo 5.12. Değişken ve sabit “k” değerleri için elde edilen sistem tasarımları.....	67

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1. Yıllara göre YBTP eniyilemesi alanında WoS tarafından taranan yayınların dağılımı.....	2
Şekil 1.2. YBTP çözümünde kullanılan yöntemleri dağılımı.....	3
Şekil 2.1. Başarısızlık fonksiyonunun zamanla değişimi [43]	8
Şekil 2.2. Tamir edilebilir bir bileşene ait durum uzayı diyagramı.....	9
Şekil 2.3. Sistem güvenilirliği eniyilemesi için kullanılan yaklaşımlar.....	15
Şekil 2.4. Yedekleme stratejileri seçenekleri.....	16
Şekil 2.5. Güvenilirlik-maliyet grafiği [52]	16
Şekil 2.6. YBTP'nin kullanıldığı farklı sistemlerin özellikleri.....	17
Şekil 2.7. YBTP çözüm yöntemleri.....	19
Şekil 4.1 Yedek bileşenli sistem tasarımı [53].....	29
Şekil 4.2. Genetik algoritmaların genel yapısı	32
Şekil 4.3. Örnek bir sistem tasarımı	33
Şekil 4.4. Sistem güvenilirliğinin tahmini için Kesikli Olaylı Monte Carlo benzetimi	35
Şekil 4.5. Örnek Sistem Tasarımı 1 (S1)	37
Şekil 4.6. Tahmini sistem güvenilirliği değerleri.....	37
Şekil 4.7. Tahmini Sistem Güvenilirliği (30 Tekrarın Ortalama Değerleri)	38
Şekil 4.8. Bir çözüm alanı/arama uzayı örneği [111].....	39
Şekil 4.9. Uniform Çaprazlamaya bir örnek.....	43
Şekil 4.10. Mutasyona bir örnek.....	44
Şekil 4.11. Durağan/Sabit durum GA sözde kodu.....	44
Şekil 4.12 Arıza ve tamirin dikkate alınarak yedekliğin kullanıldığı sistem güvenilirliğinin GA ile eniyilenmesi sözde kodu.....	45
Şekil 4.13. MA sözde kodu.....	47
Şekil 4.14 Arıza ve tamirin dikkate alınarak yedekliğin kullanıldığı sistem güvenilirliğinin MA ile eniyilenmesi sözde kodu.....	49
Şekil 5.1. P22 problemi için tamir edilmeyen bileşenlerin yer aldığı örnek sistem tasarımı.....	56
Şekil 5.2. P22 problemi için tamir edilen bileşenlerin yer aldığı örnek sistem tasarımı.....	56
Şekil 5.3. EBSG değerlerine göre algoritma sonuçlarının karşılaştırılması	60

Şekil 5.4. OSG değerlerine göre algoritma sonuçlarının karşılaştırılması.....	60
Şekil 5.5. Sabit k değeri için elde edilen sistem tasarımı.....	66
Şekil 5.6(a-c). Üç örnek probleme ait sistem tasarımı.....	66

SİMGELER VE KISALTMALAR LİSTESİ

R	sistem güvenilirliği
C	sistem maliyeti
W	sistem ağırlığı
R_{minimum}	karşılanması gereken minimum sistem güvenilirliği
x_{ij}	i'inci alt sistemde bulunan j'inci tür bileşenin sayısı
r_{ij}	i'inci alt sistemdeki j'inci bileşenin güvenilirliği
c_{ij}	i'inci alt sistemdeki j'inci bileşenin maliyeti
w_{ij}	i'inci alt sistemdeki j'inci bileşenin ağırlığı
m_i	i'inci alt sistemdeki alternatif bileşen sayısı (bileşen türü sayısı)
s	alt sistem sayısı
k_i	i'inci alt sistemde, arızalandığında sistemin arızalanmasına neden olan minimum bileşen sayısı
n_i	i'inci alt sistemde bulunan toplam bileşen sayısı
$n_{\text{max},i}$	i'inci alt sistemde toplamda bulunabilecek en fazla bileşen sayısı ($n_i \leq n_{\text{max},i} , \forall i$)

1. GİRİŞ

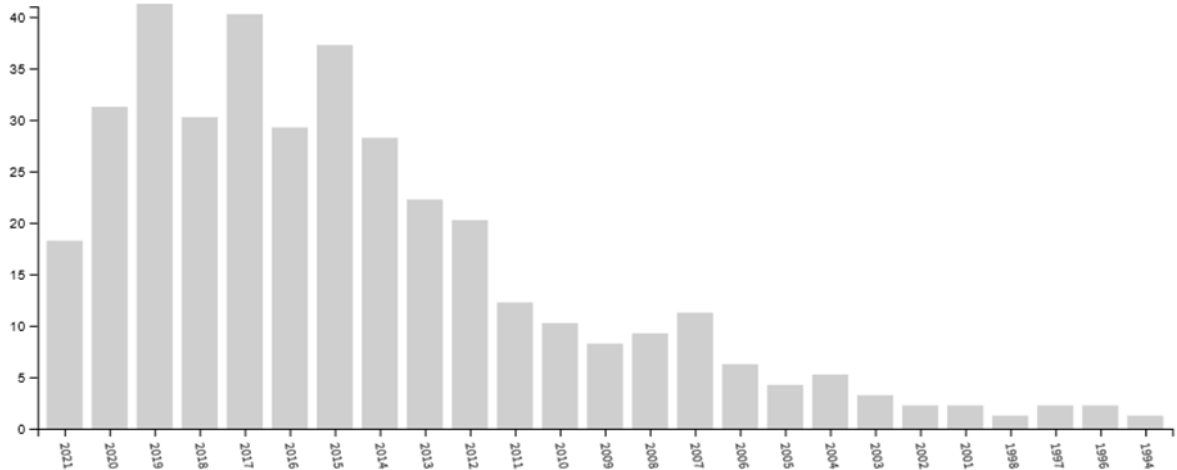
Günümüzde hızla gelişen teknoloji başta iletişim, elektronik, uzay sistemleri olmak üzere çoğu sektörde büyük değişimlerin yaşanmasına neden olmuştur. Modern mühendislik ve hizmet sistemlerinin giderek karmaşıklaşan yapısı ve ihtiyaç duyulan yüksek teknolojiye sahip endüstriyel sistemler ve süreçleri, sistemlerin daha etkin bir şekilde değerlendirilmesinin önemini artırmıştır. Karmaşık yapıdaki bu tür sistemlerin analizi ve tasarımlarıyla ilgili çalışmalarda başarımlarının değerlendirilmesi için alan yazında birçok farklı ölçüt kullanılmaktadır. Yaygın olarak kullanılan başarımlar ölçütlerinden birisi sistem güvenilirliğidir. Sistem güvenilirliği kavramı genel anlamda bir sistemin amacına uygun olarak çalışmasını ifade etmektedir.

Günümüz karmaşık sistem yapılarında, özellikle elektronik, iletişim, uydu ve bulut sistemlerinin tasarımında ve altyapılarının kurulumunda, çeşitli sistem kısıtları altında sistem güvenilirliğinin eniyilenmesi önemli ve her zaman güncelliğini koruyan bir problemdir. Günlük hayatımıza giren, çoğu çalışma alanında ve sosyal yaşamda kullandığımız uygulamaları direk etkileyen bu altyapıların tasarım problemleri güvenilirlik ölçütünü dikkate alan eniyileme problemi olarak tanımlanmaktadır. Bu nedenle bu problemin çözümü farklı disiplinlerde çalışan birçok araştırmacının yoğun ilgisini çeken bir araştırma alanıdır. Sistem güvenilirliğinin eniyilenmesi amacıyla kullanılan yöntemler dört ana başlık altında toplanabilir. Bu yöntemler; i) daha yüksek güvenilirliğe sahip bileşenler kullanmak, ii) yedek bileşen/alt sistem/sistem kullanmak, iii) ilk iki yöntemi bir arada kullanmak, iv) değiştirilebilir bileşenleri yeniden atamaktır [1]. İncelenen sistemin özelliğine bağlı olarak hangi yöntemin uygun olduğu belirlenmektedir.

Güvenilirlik tahsis problemi, çeşitli sistem kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımındaki bileşenlerin güvenilirlik değerleriyle ilgilenmektedir. Yedek tahsis problemi (YTP), çeşitli sistem kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımındaki yedek sayısı ve özelliği ile ilgilenmektedir. Bir diğer yöntem olan güvenilirlik ve yedek tahsis problemi ise hem bileşenlerin güvenilirlik değerleriyle hem de yedek sayısı ve özelliği ile ilgilenmektedir. Değiştirilebilir bileşenlerin yeniden atanması ise çeşitli sistem kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımındaki birbiriyle değiştirilebilir yani aynı işlevi yapabilen bileşenlerin uygun şekilde atanması ile ilgilenmektedir.

YTP, güvenilirlik eniyilemesi alanındaki yeni tasarım ve mevcut bir tasarımın iyileştirilmesi problemlerinde yaygın olarak kullanılan bir yöntemdir. YTP’de, genellikle maliyet, ağırlık gibi çeşitli sistem kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımındaki yedek sayısı ve özelliği ile ilgilenilmektedir. YTP ile ilgili çalışmalar, sistem güvenilirliğindeki en büyük artışın bileşen düzeyinde yedekleme yaklaşımının kullanımı ile sağlandığını göstermiştir [1]. Yedek Bileşen Tahsis Problemi (YBTP), sistemde yer alan bileşenlere paralel olarak yerleştirilen, kullanıma hazır bileşenler ile daha yüksek güvenilirliğe sahip yeni sistemin tasarlanması olarak tanımlanabilir.

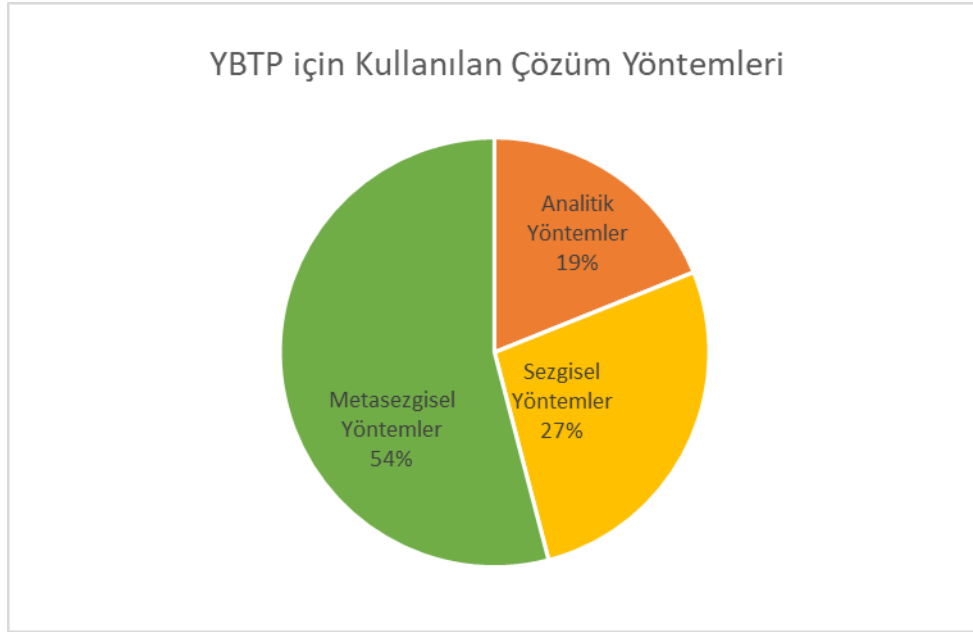
İletişim, elektronik, uzay sistemleri başta olmak üzere çoğu sektörde hızla gelişen teknoloji, modern mühendislik ve hizmet sistemlerinin giderek karmaşıklaşmasına sebep olmuştur. Bu tür karmaşık sistemlerde ihtiyaç duyulan yüksek teknolojiye sahip endüstriyel süreçlerin iyileştirilmesi amacıyla YBTP araştırmacılar tarafından her yıl artan bir hızla yaygın olarak çalışılan önemli bir konudur. Şekil 1.1, 1994 yılından günümüze dek YBTP ile ilgili yapılan çalışma sayılarının artışını göstermektedir.



Şekil 1.1. Yıllara göre YBTP eniyilemesi alanında WoS tarafından taranan yayınların dağılımı

Alan yazında sistem güvenilirliğini eniyilemek amacıyla YBTP tanımını kullanan birçok çalışma bulunmaktadır. Bu çalışmalar, analitik yaklaşım kullananlar ([2]–[5]) sezgisel ve metasezgisel yaklaşım kullananlar ([3], [6]–[10]) olmak üzere çözüm yöntemi açısından 2 grup altında incelenebilir. Yapılan ilk çalışmalarda küçük boyutlu sistemler için kesin çözüm sunan analitik yaklaşımlar kullanılmıştır. Belirli kısıtlar altında, alt sistemlerde bulunan bileşen sayısının ve alt sistemlerin sayısının artışına bağlı olarak (problemin boyutu büyüdükçe) olası alternatif sistem tasarımlarının sayısı üstel olarak artış gösterdiği için YBTP, NP-zor bir problemdir [11]. Bu nedenle, klasik eniyileme yöntemleri ile büyük boyutlu

problemlerin çözülmesi mümkün değildir. 1990'lı yılların sonlarından itibaren tek veya çok amaçlı YBTP problemlerin çözümünde sezgisel ve metasezgisel yöntemlerinden yararlanılmaktadır. Şekil 1.2 yapılan çalışmalarda kullanılan yöntemlerin dağılımını göstermektedir.



Şekil 1.2. YBTP çözümünde kullanılan yöntemleri dağılımı

YBTP çözümü için ilgili alan yazında yaygın olarak Genetik Algoritmalar (GA)'ın kullanıldığı, nadiren diğer algoritmaların da örneğin; Parçacık Sürüsü, Tavlama Benzetimi, Tabu Arama, Karınca Kolonisi ve Memetik Algoritmalar (MA) kullanıldığı görülmektedir.

Genetik algoritmalar, doğada gözlemlenen evrimsel mekanizmalara benzer mekanizmalar kullanarak çalışan eniyileme yöntemidir. Problem için olası pek çok çözümü içeren bir çözüm kümesini (yığın) kullanarak arama yaparlar. GA ilk olarak John Holland ve arkadaşlarının yaptığı çalışmalarda 1970'li yıllarda ortaya çıkmıştır [12]. Biyolojik sistemlerin gelişim sürecinin benzetimini yapan GA ve GA'nın yerel arama yaklaşımı ile iyileştirilmesi sonucunda elde edilen MA, bilinen analitik yöntemlerle çözilemeyen veya çözüm zamanı üstel artan problemlerde eniyi/eniyiye yakın çözümleri bulurlar. Alan yazın incelendiğinde, birçok farklı tür kombinatoryal eniyileme probleminin çözümünde örneğin iletişim ağlarının tasarımı ([13]–[20]), araç rotalama ([21]–[27]), proje çizelgeleme ([28]–[35]) gibi problemlerin çözümünde stokastik arama yöntemi olan GA ve MA'nın kullanıldığı görülmektedir.

YBTP, araştırmacılar tarafından kullanım alanındaki önem derecesine ve amacına göre bazen tek bazen de çok amaçlı olarak dikkate alınmıştır. Alan yazında var olan çalışmalarda

örneğin, maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımları ([10], [36]), ağırlık kısıtı altında en yüksek güvenilirliğe ve en küçük maliyete sahip sistem tasarımı ([37]) veya az sayıda çalışmada da en yüksek güvenilirliğe ve en küçük maliyet ve ağırlığa sahip sistem tasarımının ([38]) elde edildiği görülmektedir. Çok amaçlı eniyilemede aynı anda tüm amaçları eniyileyen baskın olmayan çözümlerin kümesi elde edilmektedir. ([39]).

Gerçek hayatta kullanılan sistemlerde, bir bileşenin arızalanması, tamir edilene kadar devre dışı kalması sistemin güvenilirliğini negatif yönde etkilerken, aynı bileşenin tamir edilerek çalışır duruma dönmesi sistemin güvenilirliğini pozitif yönde etkilemektedir. Yaşam ömrü boyunca sistemlerin gerçeğe uygun bu dinamik davranışları dikkate alınırsa güvenilirliğin en yüksek olacağı başlangıç tasarımları, yani hangi alt sistemde hangi tür bileşenden kaç adet yedek bulunmalı kararı gerçekçi bir yaklaşımla verilebilecektir. Böylece yüksek güvenilirliğe sahip tasarımlar daha düşük maliyetle elde edilebilecektir.

YBTP ile ilgili çalışmaların çoğunda, genellikle, yalnızca arızalar sabit arızalanma oranları kullanılarak dikkate alınmıştır. Çok az sayıdaki çalışmada ise arıza ve tamir durumu birlikte dikkate alınmış ancak yine sabit arızalanma ve tamir oranlarıyla çalışılmıştır. Busacca vd. [40], Zoulfaghari vd. [41], ve Hadipour vd. [42] bileşenlerin tamir edilebilir özellikte olduğu durumu dikkate almışlardır. Bu çalışmalarda sabit arızalanma ve tamir oranları dikkate alınarak, bileşen kullanılabilirliği analitik yöntemle hesaplanmış olup güvenilirlik hesabında bozulma ve tamirlerin dinamik etkisi ortaya konamamıştır. Bu çalışmalarda sistem bileşenlerinin arıza ve/veya tamir oranlarının sabit olduğu varsayımının kullanılmasının önemli bir nedeni sistem güvenilirliğinin analitik yöntemlerle kolay hesaplanabiliyor olmasıdır. Alan yazında bileşenlerin arızalanma ve tamir sürelerinin dikkate alınarak sistem güvenilirliğinin hesaplandığı diğer bir çalışma Lins ve Droguett'in [43] tarafından gerçekleştirilmiştir. Bozulma ve tamiri birlikte dikkate alan ve güvenilirlik tahmini için benzetim kullanan bu ilk çalışmada bileşenlerin sabit oranlarla arızalanıp sonra tamir edildiği varsayımı kullanılmıştır. Sistemin güvenilirlik değeri; daha gerçekçi bir yaklaşım olan en yakın olayın ortaya çıkışına göre değil, bölünmüş sabit zaman aralıklarına göre yapılan benzetim ile tahmin edilmiştir. Bu çalışmada önerilen algoritma sadece iki küçük örnek problem üzerinde denenmiştir, herhangi bir başarımlı kriterine göre analiz edilmemiştir.

Oysa, gerçek hayat problemlerinde sistemi oluşturan bileşenler kullanımdan kaynaklanan yıpranmaya bağlı olarak artan arızalanma oranlarına sahiptir. Öte yandan arızalanan bileşen tamir edilerek sistem içindeki görevini sürdürür. Sistemlerin bu doğal

yapısı dikkate alınarak yüksek güvenilirliğe sahip tasarımlar daha düşük maliyetle elde edilebilir. Ayrıca, YBTP’de güvenilirlik kısıtı olarak kullanılan, n ’den k ’lı (k -out-of- n) varsayımında “ k ” parametresi değişen kullanıcı taleplerine bağlı olarak değişkenlik gösterebilir. Bu nedenle YBTP’de “ k ” parametresinin değişiminin de dikkate alındığı dönemsel talep değişimlerini karşılayacak sistem tasarımlarının oluşturulması önemlidir.

Bu tez çalışmasında, artan arızalanma oranları ile ortaya çıkan bozulmaların, tamirlerin ve değişken “ k ” değerinin dikkate alındığı durumda YBTP’nin eniyilenmesi için Genetik Algoritma (GA) ve Memetik Algoritma (MA) olmak üzere iki metasezgisel algoritma önerilmiştir. Sistem güvenilirliği (amaç fonksiyonu) gerçekçi bir yaklaşımla olayların çıkışına dayalı olarak geliştirilen kesikli olaylı benzetim (KOB) modelinin kullanıldığı kesikli olaylı Monte Carlo benzetimi ile tahmin edilmiştir. KOB modelinin geçerliliği ve geliştirilen GA ve MA’nın etkinlikleri alan yazında yaygın olarak kullanılan test problemleri üzerinde gösterilmiştir. Sayısal analiz sonuçlarına göre, GA ve MA’nın kaliteli çözümler bulduğu; MA’nın GA’ya göre daha iyi sonuçlar verdiği gösterilmiştir. Böylece, arıza ve bozulmanın dikkate alındığı YBTP için daha yüksek güvenilirliğe sahip sistem tasarımlarının daha düşük maliyetle elde edilebildiği probleme özgü iki algoritma, GA ve MA, ilgili alan yazına kazandırılmıştır. Bu çalışmada güvenilirlik tahmininde kullanmak üzere geliştirilen ve önerilen KOB modeli “ k ” parametresinin değişimini dikkate alacak şekilde kurulmuştur. Böylece “ k ” açısından da dinamik bir KOB modeli ile dönemsel talep değişimlerine bağlı olarak sistem tasarımları yapılabilecektir.

Bu çalışma konusunun seçilmesindeki birinci motivasyon kaynağı, daha yüksek güvenilirliğe sahip sistem tasarımlarına duyulan gereksinimin, teknolojiye bağlı olarak birçok farklı alanda giderek artıyor olmasıdır. İkinci motivasyon kaynağı ise gerçek hayatta arızalanan bileşenler tamir edilerek sisteme kazandırıldığı halde problem varsayımlarında bu durumun genelde dikkate alınmıyor olması ve sistem tasarımlarının gerçek hayata uygun olan bu yaklaşımla yapılmasının sağlayacağı ekonomik katkının göz ardı ediliyor olmasıdır. Bu motivasyonlarla hareket ederek, daha yüksek güvenilirliğe sahip sistem tasarımlarının daha düşük maliyetle elde edilebileceği çalışmalar ve sonuçları bu tez kapsamında sunulmuştur.

Tezin diğer bölümleri şu şekilde düzenlenmiştir.

İkinci bölümde, sistem güvenilirliği ile ilgili temel kavramlar ve sistem güvenilirliği hesaplama yöntemleri ile sistem güvenilirliği eniyileme yöntemleri hakkında bilgi verilmiştir.

Üçüncü bölümde, tez konusu ile ilgili alan yazın taraması sonuçlarına yer verilmiştir.

Dördüncü bölümde, YBTP için maliyet ve ağırlık kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımını bulmak üzere geliştirilen GA ve MA ile ilgili çalışmalar detaylı bir şekilde verilmiştir.

Beşinci bölümde, geliştirilen algoritmalar için uygun parametre setinin seçiminde kullanılan deney tasarımı açıklanarak algoritmaların etkinliği incelenmiştir. Geliştirilen ve önerilen algoritmalar, alan yazında yaygın olarak incelenen problemler üzerinde, elde edilen çözümlerin kalitesi açısından karşılaştırılmıştır.

Tezin altıncı ve son bölümünde, tez kapsamında yapılan çalışmalara ilişkin genel bir değerlendirme yapılarak ileride yapılabilecek yeni çalışmalar önerilmiştir.

2. SİSTEM GÜVENİLİRLİĞİ

Bu bölümde öncelikle sistem güvenilirliği ile ilgili temel kavramlar açıklanarak sistem güvenilirliği hesaplama yöntemleri ve sistem özelliklerine bağlı olarak kullanım durumları incelenmiştir. Daha sonra daha yüksek güvenilirlik değerine sahip sistem tasarımlarının elde edilmesi amacıyla kullanılan eniyileme yöntemleri açıklanmıştır.

2.1 Temel Kavramlar

Modern mühendislik ve hizmet sistemlerinin analizi ve başarımlarının değerlendirilmesi amacıyla alan yazında yaygın olarak kullanılan başarımlar ölçütlerinden biri sistem güvenilirliğidir. Sistem güvenilirliği, bir sistemin amacına uygun olarak çalışmasını ifade eder. Sistem bileşenlerinin özellikleri, sayıları ve sistem içerisindeki konumları gibi birçok faktör sistem güvenilirliğini doğrudan etkilemektedir.

2.1.1 Güvenilirlik

Güvenilirlik, bir sistemin belirlenen çalışma koşulları altında, belirlenen zaman diliminde, kendisinden beklenen fonksiyonların yerine getirilmesi olasılığıdır [44]. Diğer bir deyişle, t rassal değişkeni arızalanma anını belirttiğinde, t anındaki güvenilirlik fonksiyonu, güvenilirliğin birikimli dağılım fonksiyonu olarak (2.1) numaralı eşitlikteki gibi gösterilebilir:

$$R(t) = P(t > T) \quad (2.1)$$

Bu durumda arızaların birikimli dağılım fonksiyonunu, $F(t)$,

$$F(t) = P(t \leq T) \quad (2.2)$$

olduğu bilindiğinden, arızaların birikimli dağılım fonksiyonu, $F(t)$, $R(t)$ 'nin tümleyicisidir.

$$R(t) + F(t) = 1 \quad (2.3)$$

Arızalanma zamanı t 'nin olasılık yoğunluk fonksiyonu $f(t)$ ise, güvenilirlik fonksiyonu, güvenilirlik birikimli dağılım fonksiyonu eşitlik (2.4)'te verilmiştir.

$$\begin{aligned} R(t) &= 1 - F(t) \\ &= 1 - \int_0^t f(s)ds \end{aligned} \quad (2.4)$$

2.1.2 Başarısızlık fonksiyonu

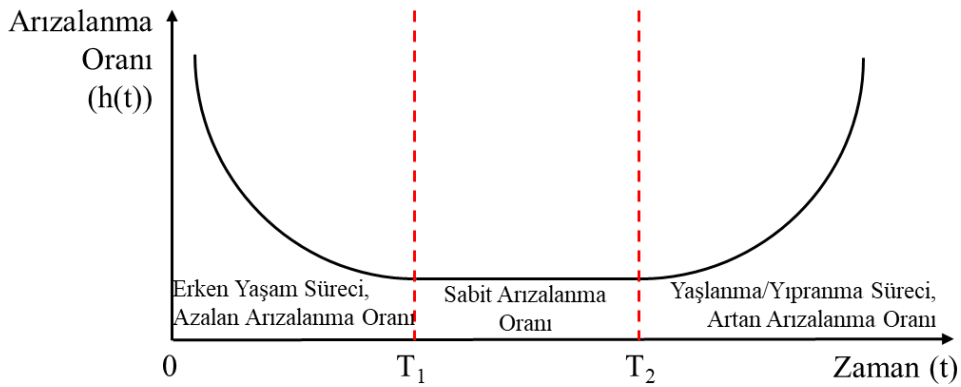
Sistem bileşenlerinin arızalanma eğilimleri, sistem güvenilirliği üzerinde önemli bir rol oynamaktadır. Arızalanma/Başarısızlık, belirli bir t zamanına kadar herhangi bir arızalanmanın olmadığı bilindiğinde $(t+dt)$ zaman aralığında bileşenin arızalanma olasılığıdır [45]. Arızalanma/Başarısızlık oranı olarak adlandırılan bu olasılık, dt zaman aralığının bir fonksiyonudur.

T , bir bileşenin arızalandığı zaman olmak üzere başarısızlık fonksiyonu eşitlik (2.5)'te verilmiştir.

$$h_T(t)dt = P(t < T \leq t + dt | T > t) = \frac{P(t < T \leq t + dt)}{P(T > t)} = \frac{f_T(t)dt}{R(t)} \quad (2.5)$$

Başarısızlık fonksiyonu, bir bileşenin ömrünü, genel olarak, üç farklı aşamada bir 'banyo küveti' eğrisi şeklinde gösterir (Şekil 2.1). Eğri üzerindeki her bir bölgenin uzunluğu, bileşenin özelliğine bağlı olarak değişiklik gösterir.

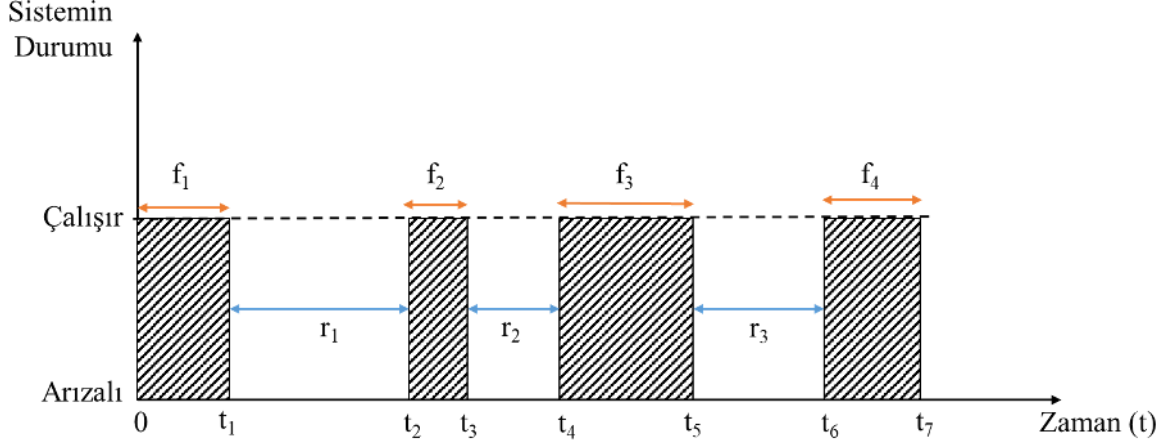
Eğrinin ilk bölgesi, bileşenlerin henüz yeni olduğu erken yaşam sürecini gösterir. Bu sürecin başlangıcında arızalanma oranı yüksektir, ancak hatalı aksam, kötü lehim gibi potansiyel arıza nedenleri giderilerek zaman içinde arızalanma olasılığı azaltılır. İkinci bölgede bileşenler sabit arızalanma oranına sahiptir; arızalar bileşenin kullanım süresinden bağımsız olarak rassal zamanlarda ortaya çıkar. Eğrinin üçüncü bölgesi, bileşenlerin kullanımdan kaynaklı yıpranmalarına bağlı olarak artan arızalanma oranına sahip olduğunu gösterir.



Şekil 2.1. Başarısızlık fonksiyonunun zamanla değişimi [44]

2.1.3 Arıza-tamir süreci

Tamir edilebilen bileşenler çalışma süreleri boyunca çalışma- arıza- çalışma döngüsü içerisinde devam ederler. Şekil 2.2'de yer alan modele göre sistemde yer alan bileşen ya çalışır durumdadır ya da arızalıdır. Burada $f_1, f_2, \dots$ ile gösterilen süreler bileşen arızalanmasına kadar geçen süreleri, $r_1, r_2, \dots$ ile gösterilen süreler ise bileşen tamir edilene kadar geçen süreleri ifade etmektedir.



Şekil 2.2. Tamir edilebilir bir bileşene ait durum uzayı diyagramı

2.2 Sistem Güvenilirliği Hesaplama Yöntemleri

Sistem güvenilirliğinin hesaplanması amacıyla mevcut alan yazında önerilmiş dört farklı yöntem vardır. Sistem özelliklerine bağlı olarak seçilen uygun yöntemle sistemin güvenilirlik değeri elde edilir. Böylece, güvenilirlik başarımları ölçütü dikkate alınarak daha yüksek güvenilirlik değerine sahip sistem tasarımlarının elde edilmesi sağlanabilmektedir.

2.2.1 Sistem yapı fonksiyonu

Geleneksel güvenilirlik hesaplama yöntemi olan *yapı fonksiyonunun* temel teorisi Birnbaum vd. tarafından oluşturulmuştur [46]. Yapı fonksiyonu, sistemin performansı ile sistem bileşenlerinin performansı arasındaki temel bağlantıyı gösteren ve sistemlerin performanslarını birbirleriyle kıyaslayabilmek amacıyla kullanılan bir kavramdır.

n bileşenden oluşan sistemde, x_i , i 'inci bileşenin durumunu gösterebilir. i 'inci bileşen çalışır durumda ise $x_i = 1$; arızalı ise $x_i = 0$ olmak üzere bileşenlerin durumlarını gösteren durum vektörü $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ 'dir. Bileşenlerin durumları ile sistem durumu arasındaki ilişki yapı fonksiyonu ile gösterilir. n bileşenden oluşan sistemin tüm olası durum vektörü uzayı $\{0, 1\}^n$ olduğu durumda sistemin durumu, yapı fonksiyonu $\phi: \{0, 1\}^n \rightarrow \{0, 1\}$ ile gösterilir. Sistem güvenilirliğinin yapı fonksiyonu ile ilişkisi eşitlik (2.6)'da verilmiştir.

$$R_{sistem} = P(\phi(x) = 1) = E(\phi(x)) \quad (2.6)$$

Gerçek hayat sistemlerinde çok sayıda bileşen bir arada çalışmaktadır. Sistem yapılandırması (tasarımı), tüm bileşenleri seri veya paralel bağlanmış şekilde basit bir yapıda

olabileceği gibi seri-paralel bağlı karma bir yapıya veya bir şebeke (ağ) tasarımına da sahip olabilir.

2.2.1.1 Seri sistemler

Birbirine seri bağlı n adet bağımsız bileşenden oluşan bu tür sistemlerde, bir bileşenin arızalanması, sistemin arızalanmasına neden olur. Seri sistemlerde sistem güvenilirliğinde en düşük güvenilirliğe sahip bileşenin etkisi en büyüktür. i 'inci bileşenin çalışma olasılığı R_i iken, sistemin güvenilirliği, bileşenlerin çalışma olasılığının çarpımıyla elde edilir. Sistemin yapı fonksiyonu eşitlik (2.7)'de; sistem güvenilirliği eşitlik (2.8)'de verilmiştir.

$$\phi(x) = x_1 \times x_2 \times \dots \times x_n = \prod_{i=1}^n x_i \quad (2.7)$$

$$R_{sistem} = R_1 \times R_2 \times \dots \times R_n = \prod_{i=1}^n R_i \quad (2.8)$$

2.2.1.2 Paralel sistemler

Birbirine paralel bağlı n adet bağımsız bileşenden oluşan bu tür sistemlerde, tüm bileşenlerin arızalanması durumunda sistem arızalanır. i 'inci bileşenin çalışma olasılığı R_i iken, sistemin güvenilirliği, bileşenlerin arızalı olma olasılıkları ile ilişkili olarak elde edilir. Sistemin yapı fonksiyonu eşitlik (2.9)'da; sistem güvenilirliği eşitlik (2.10)'da verilmiştir.

$$\phi(x) = 1 - [(1 - x_1) \times (1 - x_2) \times \dots \times (1 - x_n)] = 1 - \prod_{i=1}^n (1 - x_i) \quad (2.9)$$

$$R_{sistem} = 1 - [(1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)] = 1 - \prod_{i=1}^n (1 - R_i) \quad (2.10)$$

2.2.1.3 Paralel-Seri sistemler

m adet paralel yoldan oluşan ve her bir yolda birbirine seri bağlı n adet birbirinden bağımsız bileşenden oluşan bu tür sistemlerin güvenilirlik değeri, seri ve paralel sistemlerin yapı fonksiyonlarından yararlanılarak hesaplanır. i 'inci yoldaki j 'inci bileşen çalışır durumda ise $x_{ij}=1$; arızalı ise $x_{ij}=0$ olmak üzere, i 'inci yoldaki j 'inci bileşenin çalışma olasılığı R_{ij} olur. Sistemin yapı fonksiyonu eşitlik (2.11)'de; güvenilirlik formülü ise eşitlik (2.12)'de verilmiştir.

$$\phi(x) = 1 - \prod_{i=1}^m \left[1 - \prod_{j=1}^n x_{ij} \right] \quad (2.11)$$

$$R_{sistem} = 1 - \prod_{i=1}^m \left[1 - \prod_{j=1}^n R_{ij} \right] \quad (2.12)$$

2.2.1.4 Seri-Paralel sistemler

Seri bağılı n adet alt sistemden oluşan ve her bir alt sistemde paralel bağılı m adet birbirinden bağımsız bileşenden oluşan bu tür sistemlerin güvenilirlik değerleri, seri ve paralel sistemlerin yapı fonksiyonlarından yararlanılarak hesaplanır. i 'inci alt sistemdeki j 'inci bileşen çalışır durumda ise $x_{ij}=1$; arızalı ise $x_{ij}=0$ olmak üzere, i 'inci alt sistemdeki j 'inci bileşenin çalışma olasılığı R_{ij} olur. Sistemin yapı fonksiyonu eşitlik (2.13)'de, güvenilirlik formülü ise eşitlik (2.14)'te verilmiştir.

$$\phi(x) = \prod_{i=1}^n \left[1 - \prod_{j=1}^m (1 - x_{ij}) \right] \quad (2.13)$$

$$R_{sistem} = \prod_{i=1}^n \left[1 - \prod_{j=1}^m (1 - R_{ij}) \right] \quad (2.14)$$

2.2.1.5 Karma seri-paralel sistemler

Karma seri-paralel sistemlerin güvenilirlik değeri, seri ve paralel sistemlerin yapı fonksiyonlarından yararlanılarak sistem düzenine bağılı olarak hesaplanır.

2.2.1.6 Köprü türü sistemler

Bileşenlerin karmaşık bağlantı yapısına sahip olduğu köprü türü sistemlerin güvenilirlik değeri, seri ve paralel sistemlerin yapı fonksiyonlarından yararlanılarak sistem düzenine bağılı olarak hesaplanır.

2.2.1.7 Ardışık n 'den k 'lı (k -out-of- n : F) sistemler

n adet bağımsız bileşenden oluşan bu tür sistemlerde, en az k adet ardışık bileşenin arızalanması sistemin arızalanmasına neden olur. Bu tür sistemlerin çalışır durumda olması için en az $(n-k+1)$ adet ardışık bileşenin çalışır durumda olması gerekir.

2.2.1.8 n 'den k 'lı (k -out-of- n : F) sistemler

n adet bağımsız bileşenden oluşan bu tür sistemlerde, k veya daha çok sayıda bileşenin arızalanması sistemin arızalanmasına neden olur. Bu tür sistemlerin çalışır durumda olması için en az $(n-k+1)$ adet bileşenin çalışır durumda olması gerekir. Sistemin yapı fonksiyonu eşitlik (2.15)'te verilmiştir.

$$\phi(x) = \begin{cases} 1, & \sum_{i=1}^n x_i \geq n - k + 1 \\ 0, & \sum_{i=1}^n x_i \leq n - k \end{cases} \quad (2.15)$$

2.2.1.9 n 'den k 'lı (k -out-of- n : G) sistemler

n adet bağımsız bileşenden oluşan bu tür sistemlerde, k veya daha çok sayıda bileşenin çalışır durumda olması sistemin çalışır olmasını sağlar. Sistemin yapı fonksiyonu eşitlik (2.16)'da verilmiştir.

$$\phi(x) = \begin{cases} 1, & \sum_{i=1}^n x_i \geq k \\ 0, & \sum_{i=1}^n x_i \leq n - k + 1 \end{cases} \quad (2.16)$$

2.2.2 Sistem imzası

Alan yazında var olan çalışmalar incelendiğinde, sistem güvenilirliğinin çoğunlukla yapı fonksiyonu yöntemi kullanılarak hesaplandığı görülmektedir. Sistem güvenilirliğinin bileşenlerin durumuna ve sistem içerisindeki konumuna bağlı olarak hesaplandığı bu yöntem ile çok sayıda bileşenin bir arada çalıştığı gerçek hayatta karşılaşılan sistemlerin güvenilirliğinin hesaplanması sistemin boyutuna bağlı olarak zor bir problemdir. Bu nedenle son yıllarda, karmaşık yapıdaki bu tür sistemlerin güvenilirliklerini, bileşen konumundan bağımsız olarak hesaplayan yeni bir yöntem geliştirilmiştir. Samaniego tarafından tanımlanan *sistem imzası*, bileşenlerin konumundan bağımsız olarak oluşturulan sisteme özgü bir vektördür [47]. Sistem imzası, bileşenlerin homojen yani özdeş bağımsız dağıldığı (*i.i.d*) aynı arıza süresi dağılımına sahip olduğu varsayımı kullanılarak hesaplanır.

n bileşenli bir sistemde, bileşenlerin yaşam süreleri $X_1, X_2, \dots, X_n$ ve bu yaşam sürelerinin sıralı istatistikleri $X_{1:n}, X_{2:n}, \dots, X_{n:n}$ iken sistemin yaşam süresi T olsun. Dolayısıyla, sistemin arızalanma zamanı (veya T yaşam süresi), $X_{i:n}$ ile ilişkili olacaktır. F , $(0, \infty)$ aralığında sürekli bir fonksiyon iken, aynı F dağılımına sahip bağımsız bileşenlerden oluşan n boyutlu bir sistemin bileşenleri $X_1, X_2, \dots, X_n$ olmak üzere, i 'inci bileşeni $s_i = P(T = X_{i:n})$ olasılığına sahip olan $s = (s_1, s_2, \dots, s_n) \in [0, 1]^n$ vektörüne *sistem imzası* denir. Burada, $\sum_{i=1}^n s_i = 1$ 'dir. $X_1, X_2, \dots, X_n$, n adet bileşene sahip bir sistemin T yaşam süresi ve $s = (s_1, s_2, \dots, s_n) \in [0, 1]^n$ sistem imzası olmak üzere, t anındaki güvenilirliği eşitlik (2.17)'de verilen formül ile hesaplanır.

$$P(T > t) = \sum_{i=1}^n s_i \sum_{j=0}^{i-1} \binom{n}{j} \{F(t)\}^j \{\bar{F}(t)\}^{n-j} \quad (2.17)$$

2.2.3 Yaşam imzası

Son yıllarda giderek ilgi gösterilen diğer bir yöntem *yaşam imzası*'dır. Yaşam imzası sistem imzasından farklı olarak, homojen olmayan yani farklı türdeki bileşenlerin yer aldığı sistemlerin güvenilirlik hesabını kolaylaştırmaktadır. Bu yeni hesaplama yöntemi, birden fazla

türde bileşenden oluşan karmaşık sistemlerin güvenilirliğini hesaplamak amacıyla ilk kez 2012 yılında Coolen ve Coolen-Maturi tarafından tanımlanmıştır [48]. *Yaşam imzası*, bileşenlerin homojen olması zorunluluğunu ortadan kaldırarak gerçekçi bir yaklaşımla sistem güvenilirliğini hesaplayabilme olanağı sağlamaktadır.

$K \geq 2$ türde ve toplam m adet bileşenin bulunduğu sistemde m_k , her bir bileşen türüne ait toplam bileşen sayısı olmak üzere $\sum_{k=1}^K m_k = m$ 'dir ($k \in \{1, 2, \dots, K\}$). Bu şekilde birden fazla türde bileşen içeren sistemlerde yaşam imzası $\Phi(l_1, l_2, \dots, l_K)$ olmak üzere k türdeki bileşenlerden her birinden l_k tanesi çalışır durumdayken ($l_k = 0, 1, \dots, m_k$ ve $k = 1, 2, \dots, K$) sistemin yaşam imzası ve t anındaki güvenilirliği eşitlik (2.18) ve (2.19)'de verilmiştir.

$$\Phi(l_1, l_2, \dots, l_K) = \left[\prod_{k=1}^K \binom{m_k}{l_k}^{-1} \right] \times \sum_{x \in S_{l_1, l_2, \dots, l_K}} \phi(x) \quad (2.18)$$

$$P(T > t) = \sum_{l_1=0}^{m_1} \dots \sum_{l_K=0}^{m_K} \left[\Phi(l_1, \dots, l_K) \prod_{k=1}^K \left(\binom{m_k}{l_k} \{F_k(t)\}^{m_k - l_k} \{\bar{F}_k(t)\}^{l_k} \right) \right] \quad (2.19)$$

2.2.4 Benzetim

Günümüzde başarımlı ölçütü olarak güvenilirliğin kullanıldığı karmaşık ve yüksek maliyetli sistemlerin tasarımında, yüksek güvenilirlik değerine sahip tasarımlar araştırılırken bileşenler arızalandığında tamir edilebilme durumlarının da dikkate alınması gerçekçi bir yaklaşımdır ve önemli bir problemdir. Bu durumda; sistemin zamana göre işleyişi ve farklı koşullar altında sistem performansının nasıl etkilendiği, sistemde yer alan sürekli ve/veya kesikli süreçler kadar sistemin zamanla değişimi (dinamik) veya zamanla değişmeyen (statik) davranışları da incelenmelidir. Bu nedenle, özellikle iletişim, bulut sistemler, elektronik, uzay teknolojileri (uydu sistemleri vb.) ve enerji alanında tasarlanan sistemlerin güvenilirliğinin hesabında yukarıda belirtilen kriterleri sağlamak amacıyla benzetim tekniği kullanılabilir. Benzetim, karmaşık sistemlerin tasarımı ve analizinde kullanılan en güçlü modelleme tekniklerinden birisidir. Genel anlamda benzetim, zaman içinde sistemin işleyişinin taklididir. Benzetim, çeşitli koşullar altında sistemin tavrının gözlemlenebilmesi için, bu sistemin modellenmesi olarak da tanımlanabilir. Bir benzetim modeli geliştirildikten ve geçerliliği gösterildikten sonra gerçek sistem üzerinde yapılacak değişikliklerin etkilerini, yeni kurulacak bir sistemin performansını tahmin etmek için kullanılır [49].

Monte Carlo Benzetimi, (0-1) aralığında düzgün dağılan rassal sayılar kullanılarak, zaman faktörünün önemli olmadığı statik benzetim modellerinde kullanılır. Kesikli Olay

Benzetimi (KOB) ise durum deęiřkenlerinin zamanın belirli noktalarında deęiřtięi ve tamsayı deęerler aldıęı sistemlerde kullanılır. Benzetimdeki her bir tekrar aslında sistem sonuçlarını temsil eden bir deney olup her biri istatistiki anlamda bir örneklemdir.

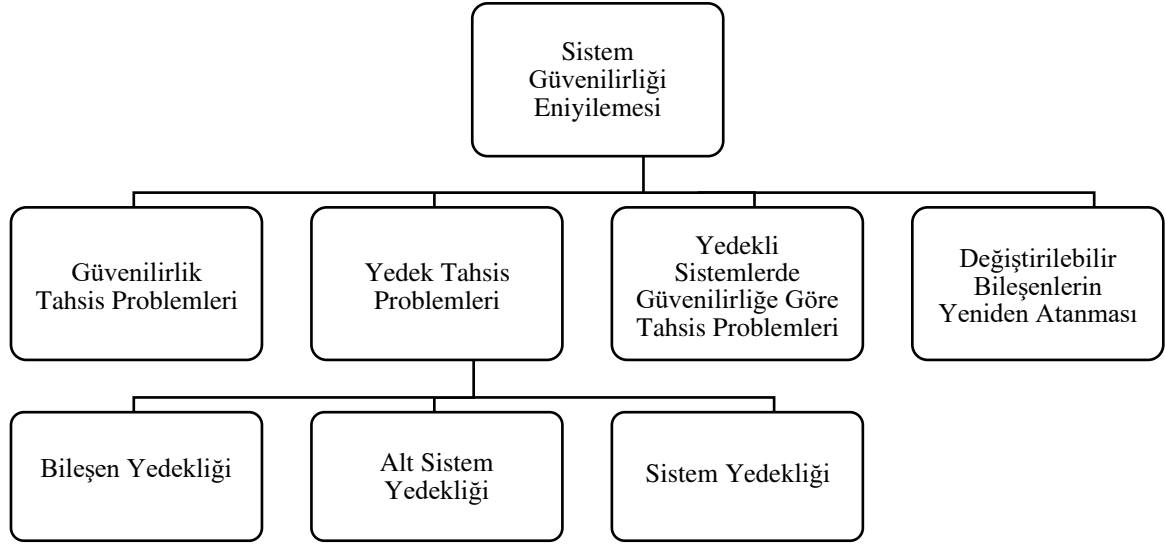
2.3 Sistem Güvenilirlięinin Eniyilenmesi Yaklařımları

Günümüzde, karmařık yapıdaki sistemlerin örneęin elektronik, haberleřme, uydu ve bulut sistemlerinin altyapılarının kurulumunda, çeřitli sistem kısıtları altında sistem güvenilirlięinin eniyilenmesi önemli ve güncel bir problemidir.

Güvenilirlięi artırmak amacıyla kullanılan yaklařımlar dört ana bařlık altında toplanabilir. Bu yaklařımlar; i) daha yüksek güvenilirlięe sahip bileřen kullanmak, ii) yedek bileřen/alt sistem/sistem kullanmak, iii) ilk iki yöntemi bir arada kullanmak, iv) deęiřtirilebilir bileřenleri yeniden atamaktır.

řekil 2.3'te sistem güvenilirlięini artırmak amacıyla kullanılabilecek yaklařımlar görölmektedir.

Güvenilirlik tahsis problemi, çeřitli sistem kısıtları altında en yüksek güvenilirlięe sahip sistem tasarımıdaki bileřenlerin güvenilirlik deęerleri ile ilgilenir. YTP, çeřitli sistem kısıtları altında en yüksek güvenilirlięe sahip sistem tasarımında, bileřenlere paralel olarak yerleřtirilen yedek sayısı ve özellięi ile ilgilenir. Bir dięer yöntem, güvenilirlik ve yedek tahsis problemi, hem bileřenlerin güvenilirlik deęerleri hem de yedek sayısı ve özellięi ile ilgilenir. Deęiřtirilebilir bileřenlerin yeniden atanması problemi ise çeřitli sistem kısıtları altında güvenilirlik deęerinin en yüksek olduęu sistem tasarımıdaki birbiriyle aynı iřlevi yapabilen bileřenlerin uygun řekilde atanması problemidir.



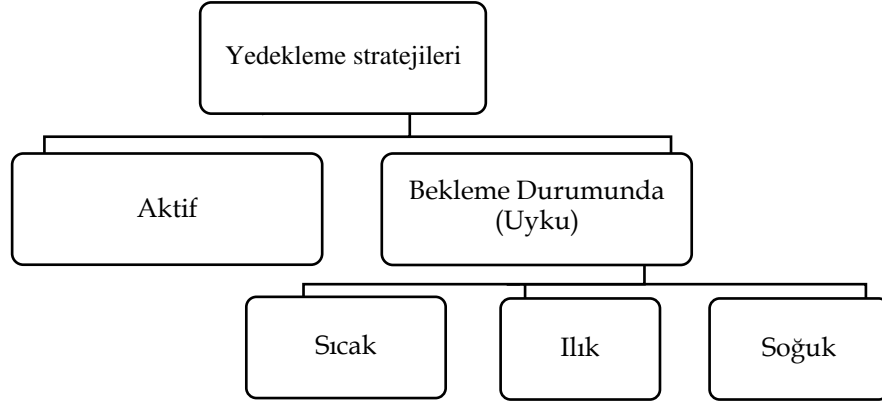
Şekil 2.3.Sistem güvenilirliği eniyilemesi için kullanılan yaklaşımlar

Karmaşık sistemlerin tasarımı ve iyileştirilmesi problemlerinde yaygın olarak kullanılan YTP’de, çeşitli sistem kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımındaki yedek sayısı ve özelliği ile ilgilenilmektedir. YTP ile ilgili çalışmalar, sistem güvenilirliğindeki en büyük artışın bileşen düzeyinde yedekleme yaklaşımının kullanımı ile sağlandığını göstermiştir [47]. YBTP, sistemde yer alan bileşenlere paralel olarak yerleştirilen, kullanıma hazır bileşenler ile daha yüksek güvenilirliğe sahip yeni sistemin tasarlanmasıdır.

İncelenen sistemin özellikleri ve kısıtlarına bağlı olarak yukarıdaki yaklaşımlardan birine dayalı tanımlanan probleme uygun çözüm bulmak için ise klasik eniyileme ([4], [50], sezgisel ([9], [51]) veya metasezgisel ([7], [37], [52]) eniyileme yöntemleri kullanılmaktadır.

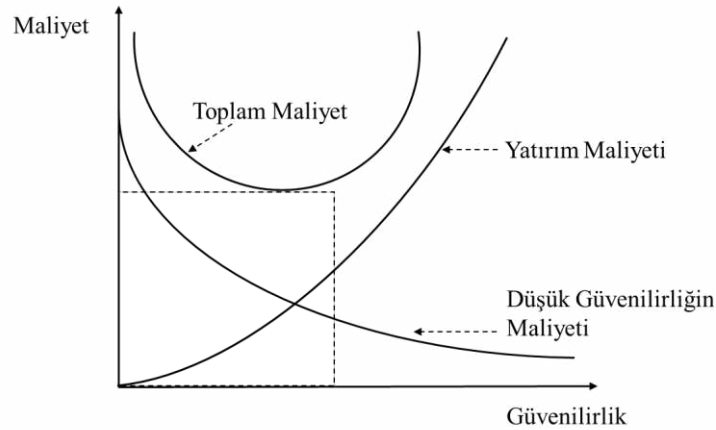
Çalışma kapsamında incelenen n ’den k ’lı sistemler (k -out-of- $n:F$) için YBTP’de, yedekleme stratejileri, yedeklerin ihtiyaç anında devreye girme hızları, bileşenin zamana bağlı yıpranma maliyeti, arıza ve tamir oranları gibi çeşitli faktörler dikkate alınmaktadır. Sistemde yer alan bileşenlere paralel olarak yerleştirilen, kullanıma hazır bileşenler *aktif yedek* olarak çalışır ve sistemin kesintisiz uzun süreler boyunca çalışması gerektiğinde kullanılmaktadır. Sistemde yer alan bileşenlere paralel olarak yerleştirilen ve çalışır durumdaki bileşenlerin arızalanmasıyla çalışmaya başlayan bileşenler *bekleme durumlu (uykuda) yedek* bileşenlerdir. Bu bileşenlerin devreye girme süreleri farklılık gösterebilir. Genellikle çalışmaya başlama süresi ile kaybedilen zamanın ihmal edilebilir olduğu durumlarda kullanılan bu bileşenler Şekil 2.4’te görüldüğü gibi üç grupta toplanabilir: i) *sıcak*, çalışmaya başlama süresinin çok

kısa olduğu, ii) *soğuk*, çalışmaya başlama süresinin uzun olduğu, iii) *ılık*, çalışmaya başlama süresinin ortalama bir süre olduğu bileşenlerdir.



Şekil 2.4.Yedekleme stratejileri seçenekleri

Yedek bileşenlerin kullanımı ile sistemde yaşanan arıza sıklığı azaltılarak daha yüksek güvenilirliğe sahip yeni sistemlerin elde edilmesi sağlanmaktadır. Yedek bileşen kullanımı nedeniyle sistem güvenilirliği yanı sıra maliyetlerde de artış gözlemlenecektir. Sistemin ilk kurulum maliyeti ve arızalardan kaynaklanan maliyetler iyi şekilde analiz edilerek sistem güvenilirliğini arttıran sistem tasarımı seçilmelidir. Şekil 2.5'te maliyet ve güvenilirliğin grafiği yer almaktadır [53].



Şekil 2.5. Güvenilirlik-maliyet grafiği [53]

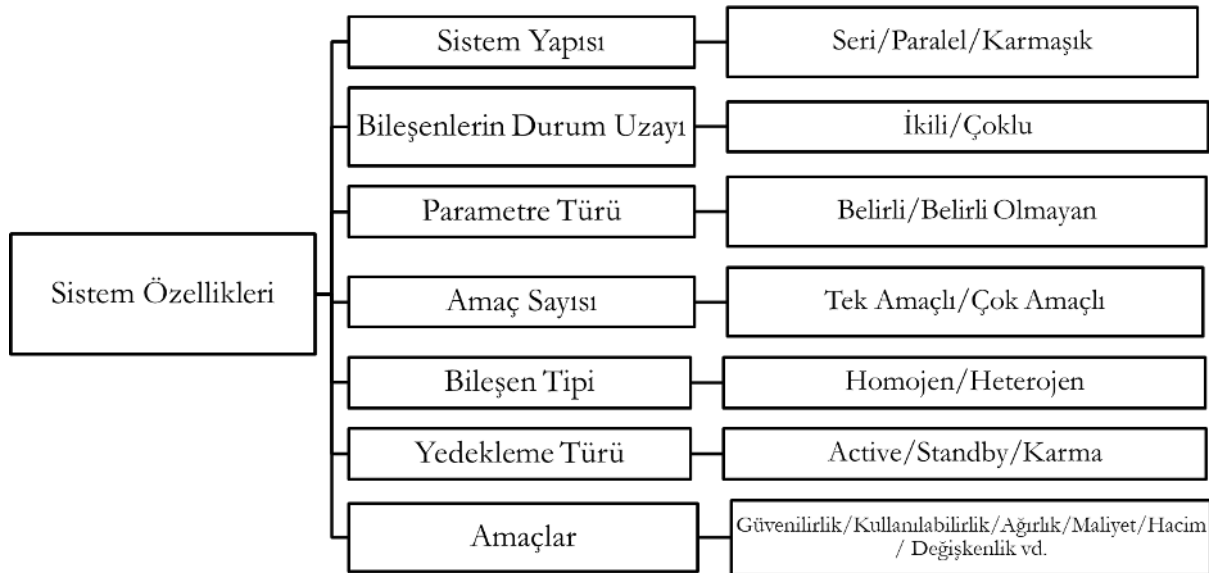
2.4 Yedek Bileşen Tahsis Problemi Türleri ve Özellikleri

YBTP, sistemde yer alan bileşenlere paralel olarak yerleştirilen, kullanıma hazır bileşenler ile daha yüksek güvenilirliğe sahip yeni sistemin tasarlanmasıdır. YBTP, farklı

sistem yapılandırması ve bileşen özellikleri için uygulanabilen, çözümü için farklı amaç fonksiyonlarını kullanabilen bir problem türüdür.

YBTP modelini oluşturmak için sistemlerin ve/veya sistem bileşenlerinin özellikleri incelenmektedir. Öncelikle, incelenen sistemin yapısı, tüm bileşenleri seri veya paralel bağlanmış şekilde basit bir yapıda olabileceği gibi seri-paralel bağlı karma bir yapıya veya bir şebeke (ağ) tasarımına da sahip olabilir. Diğer bir özellik, sistem bileşenlerinin bulunduğu durum uzayı ile ilişkilidir. Bileşenler çalışır veya arızalı durumda bulunabildikleri ikili; farklı seviyelerde çalışır veya arızalı durumda bulunabildikleri çoklu durum uzayına sahip olabilir. Bileşenlerin arızalanma ve/veya tamir verileri ile ilgili kesin ve doğru bilgi sahibi olunması da önemli bir husustur. Ayrıca, sistem içerisinde yer alan bileşenlerin aynı özelliklere sahip olması da YBTP için oluşturulan model üzerinde etkilidir. Sistemde yer alan bileşenlere paralel olarak yerleştirilen kullanıma hazır yedek bileşenlerin yedekleme stratejilerinin ne olduğu da incelenmelidir. Tüm bu bilgiler doğrultusunda, hangi değer(ler)in amaç fonksiyonunda dikkate alınacağı belirlenmelidir. Çalışmalarda yaygın olarak kullanılan amaç fonksiyonları güvenilirliği enbüyüklemek ve maliyeti enküçüklemek olsa da farklı amaç fonksiyonlarının dikkate alındığı çalışmalar da mevcuttur.

Şekil 2.6’da YBTP modelinin oluşturulması amacıyla kullanılabilecek sistem veya sistem bileşenleri özellikleri görülmektedir.



Şekil 2.6. YBTP’nin kullanıldığı farklı sistemlerin özellikleri

Alan yazında kullanılan bazı amaç fonksiyonları ve kapalı matematiksel model formları n ’den k ’lı (k -out-of- $n:F$) genel sistem yapısı için aşağıda verilmiştir [38], [54].

Problem 1. Güvenilirliği Enbüyüklemek

$$\sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \leq C$$

$$\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \leq W$$

$$\sum_{j=1}^{m_i} x_{ij} \geq (n_{\max,i} - k_i + 1) , \quad i = 1, 2, \dots, s$$

$$x_{ij} \in \{0, 1, 2, \dots\}$$

k.a.

$$enb R = \prod_{i=1}^s \prod_{j=1}^{m_i} \left[1 - (1 - r_{ij})^{x_{ij}} \right]$$

Problem 2. Maliyeti Enküçükmek

$$R = \prod_{i=1}^s \prod_{j=1}^{m_i} \left[1 - (1 - r_{ij})^{x_{ij}} \right] \geq R_{\min imum}$$

$$\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \leq W$$

$$\sum_{j=1}^{m_i} x_{ij} \geq (n_{\max,i} - k_i + 1) , \quad i = 1, 2, \dots, s$$

$$x_{ij} \in \{0, 1, 2, \dots\}$$

k.a.

$$enk C = \sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij}$$

Problem 3. Güvenilirliği Enbüyüklemek ve Maliyeti Enküçükmek

$$R = \prod_{i=1}^s \prod_{j=1}^{m_i} \left[1 - (1 - r_{ij})^{x_{ij}} \right] \geq R_{\min imum}$$

$$\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \leq W$$

$$\sum_{j=1}^{m_i} x_{ij} \geq (n_{\max,i} - k_i + 1) , \quad i = 1, 2, \dots, s$$

$$x_{ij} \in \{0, 1, 2, \dots\}$$

k.a.

$$enb R = \prod_{i=1}^s \prod_{j=1}^{m_i} \left[1 - (1 - r_{ij})^{x_{ij}} \right], \quad enk C = \sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij}$$

YBTP çözümü için kullanılan çözüm yöntemleri, sistem yapısının karmaşıklığı ve bileşen sayısına bağlı olarak değişkenlik göstermektedir. Basit yapıdaki az sayıda bileşenden oluşan sistemler için analitik yöntemler kullanılarak kesin çözümler elde edilebilirken elektronik, haberleşme, uydu ve bulut sistemleri gibi karmaşık yapıdaki çok sayıda bileşenden oluşan sistemler için, metasezgisel çözüm yöntemlerinden yararlanılır.

Şekil 2.7’de YBTP modelinin çözümü için kullanılan bazı yöntemler görülmektedir.



Şekil 2.7. YBTP Çözüm yöntemleri

YBTP alanında yapılan çalışmalar incelendiğinde, önceki yıllarda genellikle küçük boyutlu seri-paralel sistem yapıları üzerinde çalışıldığı ve aktif yedekleme stratejisinin dikkate alındığı görülmüştür. Sistem parametre değerlerinin bilindiği varsayımı altında genellikle güvenilirlik veya maliyeti eniyileyen sistem tasarımı kesin çözüm veren yöntemler kullanılarak araştırılmıştır. Oysa iletişim, elektronik ve bulut sistemleri başta olmak üzere günlük hayatta sıklıkla kullandığımız karmaşık yapıdaki bu modern mühendislik ve hizmet sistemlerinin de incelenmesi ve iyileştirilmesi gerektiği açıkça ortadadır. Bu nedenle son yıllarda çalışmalar bu tür karmaşık sistem yapıları üzerinde yoğunlaşmıştır. Son yıllarda farklı yedekleme stratejilerinin bir arada kullanılmasına izin verilen büyük boyutlu problemler sezgisel ve/veya metasezgisel algoritmalar kullanılarak çözülmüştür. Çünkü kesin çözüm sunan klasik eniyileme yöntemlerinin belirli büyüklükteki problemlere makul sürede cevap verebildiği, ancak problem boyutu büyüdükçe çözüme erişemediği veya çözüm için uzun süre beklenmesi gerektiği bilinmektedir [11]. 1960’lardan bu yana, bu tür zor eniyileme

problemlerinin çözümünde canlıları taklit eden yöntemlerin geliştirilmesi eğilimi başlamıştır. İnsanların doğal evrim sürecini taklit eden evrimsel algoritmalar olarak adlandırılan stokastik eniyileme yöntemleri geliştirilmiştir. Bu evrimsel algoritmalar YBTP gibi birçok farklı eniyileme problemine başarılı şekilde uygulanmıştır. YBTP alanında da 1990'lı yılların sonlarından itibaren sezgisel ve metasezgisel algoritmalar yaygın olarak kullanılmıştır. Sistemde bulunan yedek bileşenlerin güvenilirlik artışı sağlarken maliyet veya ağırlıkta yaratacağı artış nedeniyle çok amaçlı eniyileme problemi olarak çözüm araştıran çalışmalar artmıştır. Bazı çalışmalarda, çalışanlardan kaynaklı eksik veya yetersiz bilgi nedeniyle sistem parametre değerleri olasılıklı veya bulanık olarak dikkate alınmıştır.

3. ALAN YAZIN TARAMASI

Bu bölümde tezde ele alınan YBTP ile ilgili alan yazın taraması verilmiştir. Alan yazında sistem güvenilirliğini eniyilemek amacıyla YBTP'yi kullanan birçok çalışma bulunmaktadır. Yapılan ilk çalışmalarda doğal olarak dönemin ihtiyacına göre küçük boyutlu sistemler incelenmiş ve kesin çözüm sunan klasik eniyileme yöntemleri kullanılmıştır. Örneğin, Fyffe vd. [50], karmaşık yapıdaki bir YBTP'yi dinamik programlama kullanarak çözmüşlerdir. Nakagawa ve Miyazaki [4], maliyet ve ağırlık kısıtlarını bir vekil kısıtlama (surrogate constraints) ile kullanmışlardır.

Son 20 yıldır, gerçek hayat sistemlerinde karşılaşılan problemlerle ilgili çalışmalarda, alt sistemlerde bulunan bileşen sayısının ve alt sistemlerin sayısının artışına bağlı olarak olası alternatif sistem tasarımlarının sayısı üstel olarak artış göstermektedir ve bu nedenle YBTP NP-zor bir problemidir [11]. Bu tür zor problemlerin çözümü klasik eniyileme yöntemleri ile mümkün olmadığı için metasezgisel arama yöntemlerinden yararlanılmaktadır.

YBTP ile ilgili ilk çalışmalarda dinamik programlama, en iyi çözümü elde etmek için çoğu durumda uygulanabilir bir yöntemdir [55]. Bellman ve Dreyfuss [2], beş adet seri bağlı alt sistemin her birinde bulunacak bileşen sayılarının araştırıldığı problemde maliyet kısıtı altında en yüksek güvenilirliğe sahip sistem tasarımı için dinamik programlama kullanmışlardır. Benzer olarak Fyffe vd. [50], karmaşık yapıdaki bir YBTP'yi dinamik programlama kullanarak çözmüşlerdir. On dört adet seri bağlı her bir alt sistemde bulunması gereken bileşen sayılarının araştırıldığı problemde maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip tasarım elde edilmiştir. Bu problemde, her bir alt sistemde farklı güvenilirlik, maliyet ve ağırlık değerlerine sahip üç veya dört bileşen seçeneği bulunmaktadır. Birden fazla kısıtın dikkate alınmasını sağlamak amacıyla ağırlık kısıtı amaç fonksiyonu içinde bir Lagrange çarpanı ile kullanılmıştır. Lagrange çarpanları kullanmak yerine, Nakagawa ve Miyazaki [4], maliyet ve ağırlık kısıtlarını bir vekil kısıtlama (surrogate constraints) ile kullanmışlardır. Geliştirilen algoritmanın en iyi çözüme ulaşamayacağı durumlar için bir durdurma kriteri belirlenmiştir. Farklı ağırlık kısıtlarıyla genişletilen Fyffe ve ark.'nın [50] problemi 33 test problemi haline getirilmiş ve bu 33 problemin 30'unun en iyi çözümü elde edilmiştir. Ng ve Sancho [5], YBTP'nin en iyi çözümünü dinamik programlama ve "*depth-first search*" yaklaşımını birlikte kullanarak farklı boyutlardaki dört örnek sistem üzerinde göstermişlerdir.

Bu tür birden fazla kısıta sahip problemlerin çözümünde dinamik programlama yaklaşımının doğrudan kullanılamaması nedeniyle YBTP çözümü için doğrusal ve/veya tamsayılı programlama kullanılmıştır. Ghare ve Taylor [56], ikili karar değişkeni kullanarak bir tamsayı programlama modeli olarak ele aldıkları YBTP için maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistemi elde etmişlerdir. YBTP'yi bir sırt çantası problemi olarak formüle eden Bulfin ve Liu [51], dal-sınır algoritmasına dayanan iki yöntem ve sezgisel bir algoritma önermişlerdir. Bulfin ve Liu [51], Nakagawa ve Miyazaki [4]'ye benzer bir vekil kısıt yaklaşımı kullanmış ve önerdikleri sezgisel algoritma ile çözüm elde edilebileceğini küçük boyutlu problemler için göstermişlerdir. Coit ve Liu [57], aynı probleme her bir alt sistemde tek tür bileşenin bulunduğu ve en az k adet bileşenin çalışması gerektiği (n 'den k 'lı / k -out-of- n) durum için en iyi çözümü elde etmişlerdir. YBTP'nin çözümünde doğrusal ve/veya tamsayı programlamanın kullanıldığı diğer çalışmalara Misra ve Sharma [58], Gen vd. [59] ve Gen vd. [60] örnek verilebilir. Misra ve Sharma [58], Ghare ve Taylor'da [56] olduğu gibi tamsayı programlama kullanarak çözüme hızlı ulaşabilen bir model sunmuşlardır. Gen vd. [59] ve Gen vd. [60] çalışmalarında güvenilirlik, maliyet ve ağırlığı amaç fonksiyonunda dikkate alarak en iyi çözümü araştırmışlardır. Farklı boyutta örnek problemler için tamsayılı doğrusal programlama kullanarak çözümü araştıran diğer çalışmaların ([61]–[65]) yanı sıra Tekiner ve Coit [66], bileşenlerin güvenilirlik değerlerinin belirsiz olduğu durum için doğrusal tamsayılı programlama ile çözüm elde etmişlerdir.

Matematiksel olarak en iyi çözüme ulaşmayı amaçlayan bu tür modelleme yaklaşımlarında çözüm kolaylığı sağlanması amacıyla gerçek sistem tasarımlarında var olmayan çeşitli varsayımlar kullanılmıştır. Çok önemli ve etkili bu yöntemler güvenilirlik eniyilemesi araştırmalarının temelini oluşturmalarına rağmen elde edilen çözümler gerçek tasarım sorunlarını çözmekten uzaktır. Sistem güvenilirliği fonksiyonu doğrusal olmayan bir fonksiyondur ve YBTP için karma tamsayılı doğrusal olmayan programlama modellerinin de yer aldığı çalışmalar Tillman vd.'de [67], [68] yer almaktadır.

YBTP'ye uygulanan kesin çözüm sunan klasik eniyileme yöntemlerinin belirli büyüklükteki problemlere makul sürede cevap verebildiği, ancak problem boyutu büyüdükçe çözüme erişemediği veya çözüm için uzun süre beklenmesi gerektiği bilinmektedir [11]. Bu nedenle, sezgisel ve metasezgisel algoritmalar makul sürelerde kesin çözüm veya kesin çözüme yakın çözümler elde edebilmek amacıyla kullanılmıştır. Alan yazında 1990'lı yılların sonlarından itibaren yapılan çeşitli çalışmalarda YBTP çözümü için sezgisel ve metasezgisel algoritmalar kullanılmıştır.

GA, YBTP için en çok tercih edilen metasezgisel algoritmalarındandır. Coit ve Smith [3], [6], seri bağlı her bir alt sistemde bulunacak bileşen sayılarının araştırıldığı problemde her bir alt sistemde en az k adet bileşenin çalışması gerektiği (n 'den k 'lı / k -out-of- n) durum için GA kullanarak çözüm elde etmiştir. Coit ve Smith [69], *ceza fonksiyonu* kullanarak arama uzayına uygun olmayan çözümlerin bulunduğu alanın da dahil edilmesiyle daha etkin bir arama yapılmasını sağlamışlardır. Coit [70], bileşenlerin karma yedekleme stratejileriyle yedeklenmesine imkan sağlayan bir model önermiş ve GA kullanarak çözüm elde etmiştir. Tavakkoli-Moghadam vd. [71] ise Coit [70]'de çözümü araştırılan problemi çözmek için arama uzayını genişleten farklı bir GA önermişlerdir. Gholinezhad ve Hamadani [72], bileşenlerin karma soğuk yedekleme stratejileriyle yedeklenmesine imkan sağlayan karma bir strateji geliştirmiş ve problemi GA kullanarak çözmüşlerdir. Sistemdeki her bir alt sistemde farklı bileşenlerin bulunmasına izin veren ve bileşenlerin karma yedekleme stratejileriyle yedeklenmesine imkan sağlayan Feizabadi ve Jahromi [73], karma bir yedekleme stratejisi benimsemiş ve GA kullanarak çözüm elde etmişlerdir. Sadjadi ve Soltani [74], küçük boyutlu bir problem için geliştirdikleri sezgisel algoritma ve genetik algoritmayı elde edilen çözüm kalitesi ve elde edilme süresi bakımından karşılaştırmış ve sezgisel algoritmanın makul sürede daha iyi sonuçlar verebildiğini göstermişlerdir. Peiravi vd. [10], maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımı ve karma yedekleme stratejileriyle yedeklenmesine imkan sağlayan karma bir strateji geliştirmiş ve problemi Markov süreci ile GA kullanarak çözmüşlerdir.

Chambari vd. [9], karma yedekleme stratejilerinin kullanılabilirdiği en yüksek güvenilirliğe sahip sistem tasarımı Tavlama Benzetimi algoritması kullanarak elde etmişlerdir. Bulunan sonuçlar, Coit [70] ve Tavakkoli-Moghaddam vd. [71] tarafından elde edilen sonuçlarla karşılaştırılmıştır. Najafi vd. [75] ise bileşenlerin sadece soğuk yedekleme stratejisinin kullanılabilirdiği en yüksek güvenilirliğe sahip sistem tasarımı GA ve tavlama benzetimi algoritmasını bir arada kullanarak elde etmişlerdir. Kulturel-Konak vd. [7], üç farklı örnek problem için en yüksek güvenilirliğe sahip sistem tasarımı Tabu Arama algoritması kullanarak elde etmiş ve sonuçları Coit ve Smith [3]'da GA ile elde edilen sonuçlarla karşılaştırmışlardır.

Alan yazında kullanılan diğer metasezgisel algoritmalarından Karınca Kolonisi eniyilemesi ve parçacık sürüsü eniyilemesi, zorlu problemlerinin çözümü için genel arama tekniklerindendir. Liang ve Smith [36], maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımı Nakagawa ve Miyazaki [4]'de yer alan 33 örnek problem

için elde etmişlerdir. Çalışmada, ceza fonksiyonu kullanarak arama uzayına uygun olmayan çözümlerin bulunduğu alanın da dahil edilmesiyle daha etkin bir arama yapılması sağlanmıştır. Nahas vd. [52], önerilen bir yerel arama algoritması ile bir arada kullanılan Karınca Kolonisi algoritması, maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımını Nakagawa ve Miyazaki [4]'de yer alan 33 örnek problem için elde etmişlerdir. Elde edilen sonuçlar, alan yazında yer alan çözümlerle karşılaştırılmış ve 33 problemin 31'inde daha yüksek güvenilirliğe sahip sistem tasarımları elde edilmiştir.

Kong vd. [76], farklı yedekleme stratejilerinin kullanımına imkan sağlayan özel bir YBTP modeli oluşturmuş ve çözümü için bir Parçacık Sürüsü algoritması kullanmışlardır. Beji vd. [77], yerel bir arama algoritması ile bir arada çalışan Parçacık Sürüsü algoritması kullanarak, farklı boyutlardaki örnek problemler için maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe sahip sistem tasarımını elde etmişlerdir. Çalışmada, uyarlanabilir/adaptif ceza fonksiyonu kullanarak ihtiyaç duyulduğunda arama uzayının genişletilmesi sağlanmıştır.

Son yıllarda birçok farklı mühendislik probleminde kullanılan Yapay Arı Kolonisi algoritması da YBTP için kullanıldığı görülmüştür [78]. Safari ve Tavakkoli Moghaddam [79], Coit [70] tarafından sunulan modeli bir Memetik Algoritma kullanarak çözmüşlerdir. Diğer bir çalışma, Nahas ve ark. tarafından yapılan, farklı yedekleme stratejilerini inceleyen bir Memetik Algoritmanın kullanıldığı çalışmadır [80].

Buraya kadar verilen çalışmalarda tek amaçlı YBTP eniyilemesi dikkate alınmıştır. Oysa, günlük hayatta karşılaşılan problemlerin birden fazla amacı olabilmektedir ve bu amaçların çatışması nedeniyle tek amaçlı eniyileme algoritmaları etkili sonuçlar üretememektedir. Bu nedenle çok amaçlı optimizasyon algoritmaları da alan yazında çokça geliştirilmiştir.

Safari [81], en yüksek güvenilirliğe ve en düşük maliyete sahip sistem tasarımlarını YBTP'yi çok amaçlı bir problem olarak dikkate alarak baskın olmayan bir sıralama algoritması içeren Genetik Algoritma (NSGA-II) ile çözmüştür. Ayrıca, Jahromi ve Feizabadi [82], Feizabadi ve Jahromi [73]'de kurulan bu modeli güvenilirlik ve maliyetin amaç fonksiyonuna alınmasıyla çok amaçlı hale getirmiş ve NSGA-II ile çözüm elde etmişlerdir. Ardakan vd. [83], seri-paralel bağlı bir sistemde çok amaçlı ve karma yedeklemeye izin veren yeni bir modelin çözümünü baskın olmayan bir sıralama algoritması içeren Genetik Algoritma (NSGA-II) kullanarak elde etmişlerdir. Salazar vd. [8], alt sistemlerde farklı türde bileşenlerin bulunmasına imkan sağlayan, güvenilirlik ve maliyetin amaç fonksiyonuna alınmasıyla çok amaçlı hale getirmiş ve NSGA-II ile çözüm elde etmişlerdir. EbrahimNezhad vd. [84],

bileşenlerin satın alma ve kurulum maliyetlerini göz önüne alarak, maliyet ve ağırlık kısıtları altında en yüksek güvenilirliğe ve en yüksek net kâra sahip sistem tasarımı elde etmek amacıyla NSGA-II kullanmışlardır. Çalışmada, bileşenler farklı yedekleme stratejilerine sahip olabilmektedir. Ghorabae vd. [37], ağırlık kısıtları altında en yüksek güvenilirliğe ve en düşük maliyete sahip sistem tasarımı elde etmek amacıyla NSGA-II kullanmışlardır.

Shelokar vd. [85], en yüksek güvenilirliğe sahip sistem tasarımı elde etmek amacıyla karınca kolonisi algoritması kullanmışlardır. Ardından, en yüksek güvenilirliğe ve en düşük maliyete sahip sistem tasarımları çok amaçlı Karınca Kolonisi algoritması ile elde edilmiştir. Zhao vd. [86], belirlenen minimum güvenilirliğe sahip maliyet ve ağırlık kısıtları altında en düşük maliyet ve ağırlığa sahip sistem tasarımı elde etmek için çok amaçlı Karınca Kolonisi algoritmasını kullanmışlardır. Çalışmada, uyarlanabilir/adaptif ceza fonksiyonu kullanarak ihtiyaç duyulduğunda arama uzayının genişletilmesi sağlanmıştır.

Zhang vd. [87], çok amaçlı bir YBTP modelini oluşturmuştur ve problemi çözmek için Çıplak Kemik Parçacık Sürüsü (Bare-Bones) eniyilemesi ile duyarlılık tabanlı kümelemeyi birleştiren pratik bir yaklaşım önermişlerdir. Zaretalab vd. [88], çok amaçlı bir Parçacık Sürüsü ve Tavlama Benzetimine dayalı yeni bir algoritma geliştirmişlerdir. Valaei ve Behnamian [89], alt sistemlerde farklı türde bileşenlerin bulunabildiği ve bileşenlerin soğuk stratejiyle yedeklendiği bir sistem için, en yüksek güvenilirliğe ve en düşük satın alma maliyetine sahip sistem tasarımlarını Harmonik Arama algoritması kullanarak elde etmişlerdir. Çok amaçlı YBTP çözümü için metasezgisel algoritmaların kullanıldığı diğer çalışmalardan bazıları Dolatshahi-Zand ve Khalili-Damghani [90], Zhang ve Chen [91] ve Teimouri vd. [92] 'dir.

YBTP ile ilgili yapılan çalışmaların çoğunda, bütün bilgilerin belirli ve tam olduğu, dolayısıyla belirli (deterministik) bir ortamda çalışıldığı varsayımıyla çözüm aranmaktadır. Ancak, gerçek hayatta sistem güvenilirliği hesaplanırken veya en yüksek güvenilirliğe sahip sistem tasarımı araştırılırken bileşenlere ait güvenilirlik, arıza ve tamir oranları gibi bilgiler belirsiz olabilmektedir. Alan yazında var olan çalışmaların bir kısmında belirsizliğin bileşen güvenilirlik değerlerinde ortaya çıktığı düşünülmüştür ([66], [93]–[98]). Bir kısım alan yazında ise bileşen güvenilirliğin yanı sıra maliyet ve ağırlığın da belirsiz olduğu durumlar incelenmiştir ([99], [100]).

Yukarıda açıklanan çalışmalarda, bileşenlerin arızalandıktan sonra tamir edilmediği varsayımı kullanılmıştır. Oysa gerçek hayat problemlerinde arızalanan bileşenler tamir edildikten sonra sistem içinde görevini yeniden üstlenerek çalışmaya devam eder. Bu gerçekçi

yaklaşım dikkate alındığında yüksek güvenilirlik değerine sahip sistem tasarımları daha düşük maliyetle kurulabilmektedir. Juang vd. [101], tamir edilebilir özellikte bileşenlerin yer aldığı problemde en yüksek bileşen ekonomikliğine sahip sistem tasarımını GA kullanılarak elde etmişlerdir. Sistemin ekonomikliği, kullanılabilirliğinin toplam maliyete bölünmesiyle hesaplanır. Çalışmada, bileşenlerin kullanılabilirlik değerleri sabit arızalanma ve tamir oranları kullanılarak analitik olarak hesaplanmıştır. Jiansheng vd. [102], belirsiz arızalanma ve tamir oranlarını dikkate almış ve en yüksek kullanılabilirliğe ve en düşük maliyete sahip sistem tasarımlarını Yapay Arı Kolonisi Algoritması kullanarak elde etmişlerdir. Sabit arızalanma ve tamir oranları dikkate alındığı için bileşen güvenilirliği analitik olarak hesaplanmıştır. Çalışmada, belirsizlik altında çalışıldığı için kullanılabilirlik ve maliyet değerlerinin beklenen değeri ve varyansı incelenmiştir. Zoulfaghari vd. [41], bileşenlerin tamir edilebilir özellikte olduğu durumu dikkate almış ve sabit arızalanma ve tamir oranları kullanarak bileşen kullanılabilirliğini analitik olarak hesaplamışlardır. Çalışmada ağırlık ve hacim kısıtları altında en yüksek kullanılabilirliğe ve en düşük maliyete sahip sistem tasarımları için GA önerilmiştir. Kayedpour vd. [103], bileşenlerin farklı stratejilerle yedeklenebildiği problem için, ağırlık kısıtı altında en yüksek kullanılabilirliğe ve en düşük maliyete sahip sistem tasarımlarını NSGA-II algoritması ile elde etmişlerdir. Hadipour vd. [42], bileşenlerin tamir edilebilir özellikte olduğu durumu dikkate almış ve sabit arızalanma ve tamir oranlarını kullanmışlardır. Bileşenlerin farklı stratejilerle yedeklenebildiği bu çalışmada, ağırlık ve hacim kısıtları altında en yüksek minimum ortalama arızalanma süresine ve en düşük maliyete sahip sistem tasarımları çok amaçlı Su Akış algoritması, NSGA-II ve NREGA kullanılarak elde edilmiştir.

Tamir edilebilir bileşenlerin kullanıldığı sistem tasarımlarında sistem bileşenlerinin arıza ve/veya tamir oranlarının sabit olduğu varsayımının kullanılmasının önemli bir nedeni, sistem güvenilirliğinin analitik yöntemlerle hesaplanabiliyor olmasıdır. Artan arızalanma oranının kullanılması durumunda sistem güvenilirliği fonksiyonunun analitik yöntemlerle hesaplanması mümkün değildir. Guilani vd. [104] çalışmalarında, bileşen arızalanmalarının yıpranmaya bağlı olarak zamanla artış gösterdiği gerçeğine değinmişlerdir. Bileşen güvenilirlikleri Monte Carlo benzetimi kullanarak hesaplanmıştır. Ancak, bu çalışmada, bileşen güvenilirlikleri, bileşenlerin sadece arızalanma verilerine dayalı olarak hesaplanmıştır. Oysa, gerçek hayat sistemlerinde arızalanan bileşenler tamir edildikten sonra çalışmaya devam etmektedir.

Bosse vd. [105], internet teknolojileri için tamir kaynaklarının kısıtlı olduğu durumda, arıza ve tamir durumlarında bağımlılığı dikkate alarak internet teknolojileri için YBTP'yi tanımlamışlardır. Çalışmalarında, Petri Ağları ve Monte Carlo benzetimi ile sistem güvenilirliğini tahmin ederek GA ve Tabu Arama algoritması önermişlerdir. Ancak tanımladıkları kullanılabilirlik durum uzayında durumlar arası geçiş önerilerinin özel bilgiye dayalı olarak çıkarılması gerekliliği ve kullanımının pratik olmaması yöntemin tercih edilebilirliğini tartışmalı duruma sokmaktadır. İnternet teknolojileri ile ilişkili alt yapı sistemleri, örneğin uzaktan eğitimi, sağlık uygulamaları, milyonlarca kullanıcıyı etkilemektedir ve bu nedenle bu tür sistemlerde yeterli düzeyde kaynak bulundurulmalıdır. Diğer yandan sistem bileşenlerinin durumları arası geçişleri ile ilgili (yüzlerce bileşenin olduğu durumlar düşünüldüğünde) gerçekçi değerlerin elde edilmesinin de mümkün olamayacağı açıkça ortadadır. Yani bileşen bir durumdan diğer duruma geçtiğinde bundan sonraki bozulma zamanının nasıl etkileneceğinin belirlenmesi çok zor olup sadece varsayımlara dayalı olarak dikkate alınabilir. Ayrıca, NP zor olan problem daha da zorlaşmaktadır. Bu nedenle alan yazında bugüne dek kullanılan bozulmaların bağımsız olması varsayımı bu tez kapsamında da kullanılmaya devam edilmiştir.

Lins ve Droguett [43], arıza ve tamir işlemini dikkate alan çalışmalarında sistem güvenilirliğini tahmin etmek amacıyla kesikli olay benzetimi kullanmışlardır. Bu çalışmada eşit zaman aralıkları için sistemin çalışır/çalışmaz durumda olmasını dikkate alınarak Monte Carlo benzetimi ile güvenilirlik tahmin edilmiştir. Bileşenlerin arızalanma zamanı ve tamir süreleri genelleştirilmiş yenileme sürecine (Renewable Process) dayalı olarak elde edilir ve her bir zaman aralığının sonunda sistemin çalışır durumda olup olmadığı ikili karar ağacı diyagramı ile belirlenir. Çalışmada tüm zaman aralıkları için hesaplama yapılmaktadır.

Oysa, sistemin tüm çalışma zamanı boyunca, arıza ve tamir olaylarının ortaya çıkışına göre benzetim zamanının en yakın olay zamanına göre ilerletilmesi yaklaşımı sistemin durum değişkenleri güncellenerek sistemin gerçeğe uygun dinamik benzetim modellenmesi yapılabilir. Durum değişkenlerinin her değişiminde *k-out-of-n*'in sağlanmadığı durumla karşılaşırsa sistemin başarısız olduğu kararı verilerek (güvenilirlik=0) benzetimin ilgili tekrarı sonlandırılır ve sistem başlangıç anına döndürülerek diğer tekrara (örneğe) geçilebilir. Böylece KOB modeli ile (kesikli, dinamik, olasılıklı bir model) sistem gerçeğe uygun modellenirken, bu denli karmaşık sistemlerin sistem güvenilirliği tahmininde çalışma zamanı açısından makul sürede tahmin yapılabilecektir. Bu yaklaşım, alan yazındaki çalışmalardan farklı bir öneri olarak bu tez çalışmasının motivasyonlarından birisini oluşturmuştur.

Bu yapıda oluşturulan kesikli olaylı Monte Carlo benzetimi, sistem güvenilirliğinin tahmini için bu tez kapsamında modellenerek geliştirilmiştir. Kullanılan KOB modelinde dönemsel olarak değişebilecek müşteri taleplerini de karşılayabilmek amacıyla, dinamik *k-out-of-n* dikkate alınmıştır. Kesikli olaylı Monte Carlo benzetimi ~~KOB modeli~~ YBTP'nin çözümü için geliştirilecek sezgisel algoritmalarla etkileşimli olarak uygunluk (amaç) fonksiyonu olarak kullanılmıştır.

Bu tez çalışmasının diğer bir motivasyonu ise yukarıda belirtilen gerçekçi yaklaşım ile yüksek güvenilirliğe sahip tasarımların daha düşük maliyetlerle elde edilebilme düşüncesidir. Çünkü bozulan bileşenler gerçek hayatta uygulandığı şekilde devre dışı kalmayarak tamir edildikten sonra çalışmaya devam etmektedir.

Lins ve Droguett [43] çalışmalarında YBTP'nin çözümü için çok amaçlı GA önermişlerdir. GA'da başlangıç yığını rassal olarak oluşturulmuş, tek noktalı çaprazlama ve uniform mutasyon kullanılarak sonraki yığın oluşturulmuştur. Uygun olmayan çözümler algoritma işleyişi sırasında dikkate alınmamış ve uygunluk fonksiyonu olarak amaç fonksiyonu kullanılmıştır. Çalışmada, geliştirilen algoritmanın YBTP ile ilgili literatürde çözülmüş problemler üzerindeki başarısı ortaya konulmamış ve sadece iki küçük örnek problem çözümü verilmiştir. Dolayısıyla, çalışmalarında önerdikleri GA'nın YBTP alanında kullanılan test problemleri ve büyük boyutlu farklı problemler üzerindeki başarısı bilinmemektedir.

Bu tez kapsamında ele alınan YBTP elektronik, haberleşme, uydu ve bulut sistemlerinde karşılaşılan ve çok sayıda araştırmacının hala üzerinde çalıştığı önemli bir problem olup araştırmacıların daha etkili bir algoritma geliştirme çabası devam etmektedir. Dolayısıyla, çeşitli kısıtlar altında sistem güvenilirliğinin eniyilenmesi YBTP için önemli ve güncel bir problemidir.

Tezin en önemli motivasyonu alan yazın taraması sonucunda yukarıdaki şekilde oluşmuş ve bu ihtiyaç ile tez kapsamında etkili bir algoritmanın geliştirilmesi hedeflenmiştir.

Bu amaçla bu tez kapsamında geliştirilen GA ve MA maliyet ve ağırlık kısıtları altında yukarıda belirtilen kesikli olaylı Monte Carlo benzetimini amaç fonksiyonu olarak kullanarak en iyi sistem tasarımını elde etmek üzere çözüm yöntemleri olarak önerilmiştir. Geliştirilen KOB modelinin ve GA'nın doğrulanma ve geçerliliği alan yazında mevcut test problemleri üzerinde gösterildikten sonra çözüm kalitesi ve çözüm zamanı açısından karşılaştırılmış ve bileşenlerin arıza ve tamirlerinin dikkate alındığı YBTP için başarılı bir çözüm yöntemi olarak önerilmiştir.

4. ARIZA VE TAMİRİN DİKKATE ALINDIĞI YEDEK BİLEŞEN TAHSİS PROBLEMİ İÇİN ÖNERİLEN SEZGİSEL ALGORİTMALAR

YBTP'nin NP-zor problem olduğu Chern [11] tarafından ispat edilmiştir. Bu nedenle, maliyet ve ağırlık kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımının bulunması amacıyla alan yazın taramasında bir çok sezgisel ve metasezgisel algoritmalar geliştirilmiştir ([10], [36], [52], [77], [82], [84], [106]). Bu tezde ise alan yazında yok denecek kadar az ilgilenilmiş olan bileşen arızası ve tamirini dikkate alarak tanımlanan problemin gerçekçi varsayımlarla çözümü ele alınmış ve probleme özgü iki metasezgisel algoritma geliştirilerek önerilmiştir.

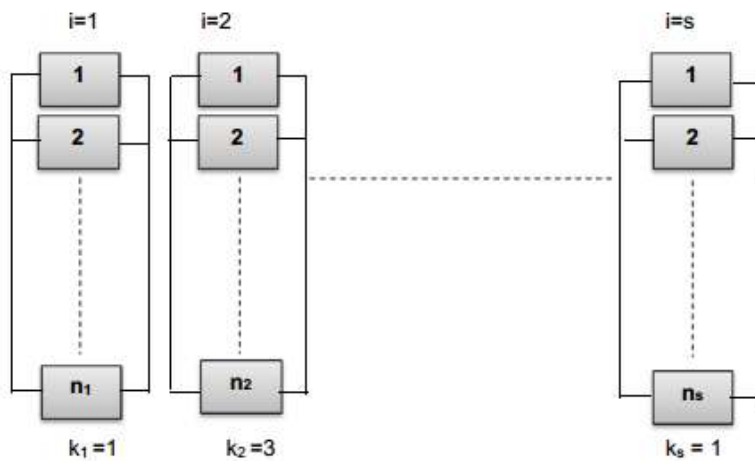
Algoritmalar sunulmadan önce, problem ve varsayımları hakkında kısa bir bilgi Bölüm 4.1'de verilmektedir.

4.1 Problem Tanımı ve Varsayımlar

YBTP, s adet seri bağlı her bir alt sistemde yer alan bileşenlere paralel olarak yerleştirilen ve kullanıma hazır bileşenler ile daha yüksek güvenilirliğe sahip yeni bir sistemin tasarlanması olarak tanımlanabilir.

Bu çalışmada, maliyet ve ağırlık kısıtları altında güvenilirlik değerinin en yüksek olduğu sistem tasarımının bulunması ile ilgilenilmektedir.

Örnek bir sistem yapısı Şekil 4.1'de gösterilmektedir.



Şekil 4.1 Yedek bileşenli sistem tasarımı [54]

Varsayımlar;

1. Her bir alt sistemde k_i veya daha çok sayıda bileşenin arızalanması sistemin arızalanmasına neden olur (n 'den k 'lı sistem/ k -out-of- n : F).
2. Bileşenlerin arızalanmaları birbirinden bağımsızdır.
3. Sistemde önleyici bakım planlaması yapılmamaktadır.
4. Bileşenler sistemde çalışır veya arızalı durumda bulunabilir.
5. *Bileşenler arızalandığında tamir edilmektedir.*
6. *Bileşenler tamir sonrası eski performansına sahip olmaktadır.*
7. *Tamir için sistemde yeterli sayıda görevli bulunmaktadır.*
8. *Bileşenlerin tamir edilme maliyeti ihmal edilmektedir.*
9. Sistem ağırlığı ve maliyeti, bileşen ağırlık ve maliyetlerinin doğrusal bir fonksiyonudur.

Bu varsayımlar kullanılarak yukarıda açıklanan problemin karar değişkeni ve parametreleri açıklanmış ve matematiksel modeli eşitlik 4.1'de verilmiştir.

Karar Değişkeni

x_{ij} i 'inci alt sistemde bulunan j 'inci tür bileşenin sayısı

Parametreler:

R sistem güvenilirliği

C sistem maliyeti

W sistem ağırlığı

r_{ij} i 'inci alt sistemde bulunan j 'inci tür bileşenin güvenilirliği

c_{ij} i 'inci alt sistemde bulunan j 'inci tür bileşenin maliyeti

w_{ij} i 'inci alt sistemde bulunan j 'inci tür bileşenin ağırlığı

m_i i 'inci alt sistemdeki alternatif bileşen sayısı (bileşen türü sayısı)

s alt sistem sayısı

k_i i 'inci alt sistemde, arızalandığında sistemin arızalanmasına neden olan minimum bileşen sayısı

n_i i 'inci alt sistemde bulunan toplam bileşen sayısı, $n_i = \sum_{j=1}^{m_i} x_{ij}$

$n_{max,i}$ i 'inci alt sistemde toplamda bulunabilecek en fazla bileşen sayısı ($n_i \leq n_{max,i} \forall i$)

Matematiksel Model:

$$\begin{aligned} \sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} &\leq C \\ \sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} &\leq W \\ \sum_{j=1}^{m_i} x_{ij} &\geq (n_{\max,i} - k_i + 1), \quad i = 1, 2, \dots, s \\ x_{ij} &\in \{0, 1, 2, \dots\} \\ k.a. \end{aligned} \tag{4.1}$$

$$enb R = \prod_{i=1}^s \prod_{j=1}^{m_i} \left[1 - (1 - r_{ij})^{x_{ij}} \right]$$

4.2 Geliştirilen Genetik Algoritma

Genetik Algoritmalar, doğada gözlemlenen evrimsel mekanizmalara benzer mekanizmaları taklit eden algoritmik bir yaklaşımdır. Darwin'in "en iyi olan yaşar (survival of the fittest)" prensibine dayalı olarak biyolojik sistemlerin gelişim sürecini taklit eden yapıdadır. Belirli bir maliyet fonksiyonuna göre her iterasyonda en iyi sonucu üreten kromozomun hayatta kalması prensibine dayanan bir arama yöntemi olarak da tanımlanabilir.

John Holland tarafından 1975 yılında geliştirilmiş ve önerilmiştir [12]. GA, problemin en iyi çözümünü bulma garantisi vermeyen, ancak problemlere makul süre içinde, kabul edilebilir en iyi çözümü sunan sezgisel bir yöntemdir. Eniyi çözümü bulmayı garanti etmeyen GA, kesin çözüm veren yöntemlerle çözülemeyen veya çözüm zamanı problemin büyüklüğü ile üstel artan problemlerde eniyiye çok yakın çözümler vermektedir. Genetik algoritmalar farklı çözümlerden oluşan bir yığın ile çalışır. Arama uzayında aynı anda birçok noktada arama yaparak eniyi/eniyiye yakın çözüme ulaşma olasılığını yükseltmektedir.

Başlangıçta sürekli doğrusal olmayan (non-linear) eniyileme problemlerine uygulanan GA, daha sonra yapılan çalışmalarda gezgin satıcı, karesel atama, yerleşim, atölye çizelgeleme, ders/sınav programı hazırlanması, topoloji tasarımı gibi kombinatoriyal eniyileme problemlerinde de başarıyla uygulanmıştır. Herhangi bir problemin çözümünde kullanılan basit bir GA'nın yapısı Sekil 4.2'deki algoritma ile açıklanabilir.

```

t=0;
Pt başlangıç yığını oluştur;
Pt 'yi değerlendir;
while not {bitis kosulu} do
    t=t+1;
    Pt-1'den Pt'yi seç; {seçim mekanizması}
    Pt'yi değişime ugrat; {çaprazlama ve mutasyon operatörü}
    Pt'yi değerlendir;
End While

```

Şekil 4.2. Genetik algoritmaların genel yapısı

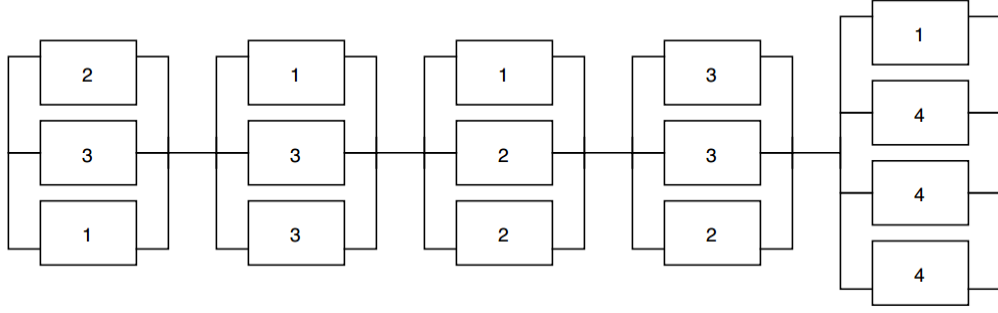
4.2.1 Dizi Gösterimi

GA' yı diğer sezgisel arama yöntemlerinden ayıran en önemli özellik, parametrenin kendisi yerine parametreleri temsil eden dizilerin kullanılmasıdır. Bu nedenle, herhangi bir problem için GA uygulanmasında öncelik, problemin arama uzayını en iyi temsil eden uygun bir dizi gösterimi yapısının seçimidir.

GA'da gösterim (kodlama) gerekli bilgiyi tutma ve çözümü dizi halinde sunma işlemidir. Dizi içerisinde yer alan her değer çözüme ilişkin bir bilgiyi saklamaktadır. Genetik algoritmada kullanılan temel gösterim biçimleri 0-1 (ikili sayı), tamsayı ve permütasyon gösterimlerdir. GA'da, kullanılan gösterim biçimlerinin etkin şekilde arama yapabilmeleri için oldukça önemlidir ve probleme uygun olarak seçilmelidir.

Bu tezde ele alınan problem için GA'da oluşturulan yığında her dizi, bir aday sistem tasarımını göstermektedir. Amaç, problemin kısıtlarını sağlayan en yüksek güvenilirliğe sahip sistem tasarımını elde etmektir. Her dizinin uzunluğu, alt sistem sayısı ve her bir alt sistemde bulunabilecek maksimum bileşen sayısının çarpımına ($s \times n$) eşittir. Dizideki her eleman ilgili konumda bulunan bileşen türünü göstermektedir. Her aday tasarım tamsayı kodlama ile gösterilmiştir [38]. Dizi içerisinde bileşen bulunmayan/kullanılmayan elemanlar "0" ile kodlanmıştır.

Örneğin, her bir alt sistemde en fazla 5 bileşen bulunmasına izin verilmiş olan ve $s = 5$ alt sistemden oluşan bir sistem için herhangi bir aday tasarım Şekil 4.3'te görüldüğü gibi olsun.



Şekil 4.3 Örnek bir sistem tasarımı

Bu sistem tasarımı [2,3,1,0,0, 1,3,3,0,0, 1,2,2,0,0, 3,3,2,0,0, 1,4,4,4,0] şeklinde kodlanmış bir dizi ile gösterilir.

4.2.2 Başlangıç yığınının oluşturulması

GA'yı diğer sezgisel arama algoritmalarından ayıran önemli bir özellik ise noktadan noktaya değil, noktaların oluşturduğu bir yığın içinde aramayı gerçekleştirmesidir [12]. Klasik eniyileme yöntemlerinin birçoğu çözüm uzayındaki tek bir noktaya uygulanır ve bu durum yerel eniyi tuzağına düşülmesi tehlikesini doğurur. GA ise, başlangıç yığını sayesinde çok yönlü arama yapabilir. Yığın yaklaşımı, yerel eniyiye düşme tehlikesini oldukça azaltır. Bu durum, her yeni nesilde görece iyi çözümlerin yeniden üretilmesi ve kötü çözümlerin ölmesi şeklinde sağlanır.

GA'nın ilk adımı başlangıç yığınının (nesil) oluşturulmasıdır. Alan yazında, başlangıç yığınının rassal ya da sezgisel yöntemlerle oluşturulduğu çalışmalar bulunmaktadır [38], [69], [81], [107]. Bu tezde ele alınan problem için geliştirilen GA'da başlangıç yığını rassal olarak oluşturulmuştur. Dizi içinde ilgili alanda bileşen olup olmadığı, varsa bileşenin türü rassal olarak belirlenmektedir. İlgili alanda bileşenin varlığı ve varsa türü eşit olasılıkla üretilmiştir.

4.2.3 Amaç fonksiyonu

4.2.3.1 Sistem güvenilirliğinin tahmini

Sistem güvenilirliğinin hesaplanması amacıyla kullanılan yöntemler Bölüm 2.2'de detaylı olarak verilmiştir. Önceki bölümlerde belirtildiği gibi, yüksek güvenilirlik değerine sahip tasarımlar araştırılırken bileşenler bozulduğunda tamir edilebilme durumlarının da dikkate alınması gerçekçi bir yaklaşım olup önemli bir problemidir. Bu durumda; sistemin zamana göre işleyişi ve sistemin durum değişkenlerinin farklı değerlerinde sistem başarımının nasıl etkilendiği, sistemde yer alan sürekli ve/veya kesikli süreçler kadar sistemin zamanla değişimi (dinamik) veya zamanla değişmeyen (statik) davranışları da incelenmelidir.

Bu çalışmada geliştirilen GA'da, amaç fonksiyonu olarak kullanılan sistem güvenilirliğinin tahmini Kesikli Olaylı Monte Carlo benzetimi ile elde edilmiştir. Sistem işleyişinin taklidi için kodlanan KOB modelinde olayların ortaya çıkışına dayalı yaklaşım kullanılırken benzetim zamanı en yakın olay zamanına göre ilerletilmiştir. Bileşen arızalanma ve tamir süreleri belirlenerek alt sistemlerin durumları sırasıyla kontrol edilmiştir. Sistem durumlarını dikkate alarak sistem güvenilirliğinin beklenen değeri Kesikli Olaylı Monte Carlo benzetimi ile tahmin edilmiştir.

KOB modeli, güvenilirlik kısıtı olarak kullanılan, n 'den k 'lı (k -out-of- $n:F$) varsayımında “ k ” parametresinin dönemsel talep değişimlerine bağlı olarak değişimi dikkate alacak şekilde kurulmuştur. Böylece, “ k ” açısından da dinamik bir KOB modeli kurularak değişen talep değerlerine karşı esnek sistem tasarımları oluşturulabilmektedir. Bu tezde geliştirilen Kesikli Olaylı Monte Carlo benzetiminin algoritması ve algoritmanın sözde kodu Şekil 4.4'te verilmiştir.

```

sayaç ← 0
For rep=1 to Rep
  Sistemde yer alan altsistem ve bileşenleri belirle
  For i=1 to s
    While not {  $t_{benzetim} > t_{son}$  } do
       $t_{benzetim} \leftarrow 0$ 
      j “eleman” 1 to  $n_i$ 
        Bileşen durumları ← çalışır
        Bileşenlerin arızalanma zamanlarını üret
       $t_{benzetim} \leftarrow t_{enyakin}$  (En erken arızalanan bileşenin zamanı)
      While not { altsistem arızalı } do
        Bileşen durumlarını güncelle ve en yakın olay
        (arızalanma/tamir) zamanlarını üret
         $t_{benzetim} \leftarrow t_{enyakin}$  (En yakın olay zamanına sahip
        bileşenin zamanı)
        break if {  $t_{benzetim} > t_{son}$  }
      End while
      if altsistem arızalı
         $sayaç \leftarrow sayaç + 1$ 
         $t_{benzetim} \leftarrow t_{son}$ 
         $i \leftarrow i + 1$ 
      End if
    End while
  End for
End for
Sistem güvenilirliği ←  $(Rep - sayaç) / Rep$  raporla

```

Şekil 4.4 Sistem güvenilirliğinin tahmini için Kesikli Olaylı Monte Carlo benzetimi

4.2.3.2 KOB Modelinin doğrulanması ve geçerliliği

KOB modelinin doğrulanması, problemin akış şeması ile benzetim modeli için yazılan program kodunun uygunluk kontrolü ile sağlanmıştır. KOB modelinin geçerliliği ise 25 test problemi üzerinde gösterilmiştir. İlgili alan yazında bulunan ve tamirlerin dikkate alınmadığı sistemlerden oluşan bu test problemlerinin analitik yöntemle hesaplanan güvenilirlikleri bilinmektedir[5], [50], [108]–[113].

Bu tez kapsamında önerilen ve geliştirilen KOB modeli seçilen problemler için, tamirlerin olmadığı varsayımı altında, çalıştırılarak güvenilirlik değerleri tahmin edilmiş ve %95 güvenlik düzeyinde güven aralıklarının hesaplanmış olan güvenilirlik değerlerini kapsadığı Tablo 4.1’de gösterilmiştir. Ayrıca tablo 4.1’de güven aralıkları tarafından kapsanan, analitik modellerle elde edilmiş güvenilirlik değerlerine hangi deneme sayılarıyla ulaşıldığı da yer almaktadır.

Tablo 4.1. 25 test problemi için %95 güvenlik düzeyinde güven aralığı değerleri

Örnek Sistem No	Analitik Sistem Güvenilirliği	Deneme Sayısı	%95 Güven Aralığı
1	0,7144	200	[0,7106, 0,7194]
2	0,9467	500	[0,9451, 0,9469]
3	0,9700	300	[0,9689, 0,9711]
4	0,9657	700	[0,9656, 0,9664]
5	0,9716	600	[0,9711, 0,9722]
6	0,9682	100	[0,9667, 0,9733]
7	0,9702	200	[0,9683, 0,9717]
8	0,9710	700	[0,9710, 0,9719]
9	0,9702	100	[0,9667, 0,9733]
10	0,9850	800	[0,9849, 0,9851]
11	0,9135	200	[0,9123, 0,9177]
12	0,9500	400	[0,9489, 0,9511]
13	0,8671	600	[0,8663, 0,8677]
14	0,9501	700	[0,9494, 0,9506]
15	0,9423	800	[0,9419, 0,9431]
16	0,9504	500	[0,9491, 0,9509]
17	0,9343	400	[0,9338, 0,9362]
18	0,9501	200	[0,9479, 0,9521]
19	0,6970	100	[0,6910, 0,7090]
20	0,9665	200	[0,9632, 0,9668]
21	0,9447	700	[0,9436, 0,9449]
22	0,6640	800	[0,6638, 0,6662]
23	0,7887	600	[0,7870, 0,7897]
24	0,9409	100	[0,9353, 0,9447]
25	0,8478	800	[0,8466, 0,8484]

Bu çalışmada, hem KOB modeli geçerliliğinde hem de güvenilirlik eniyilemesi için geliştirilen metasezgisel algoritmalar içinde kesikli olaylı Monte Carlo benzetimi ile güvenilirlik tahmininde gerçek hayat sistemlerini gerçekçi bir şekilde yansıtabilmek amacıyla bileşenlerin yıpranmaya bağlı olarak artan arızalanma oranına sahip olduğu varsayımı kullanılmıştır. Güvenilirlik alanında bileşen ömrü için yaygın olarak kullanılan dağılımlardan biri olan Weibull dağılım [114], bileşenler artan, azalan veya sabit arızalanma oranlarına sahip olduğunda kullanılabilir [104]. Weibull artan arızalanma fonksiyonu eşitlik (4.2)'de verilmiştir.

$$h(t) = \frac{B}{A} \left(\frac{t}{A} \right)^{B-1} ; A > 0, B > 0, t > 0 \quad (4.2)$$

A ölçek ve B şekil parametresini temsil ettiğinde güvenilirlik fonksiyonu eşitlik (4.3)'te verilmiştir.

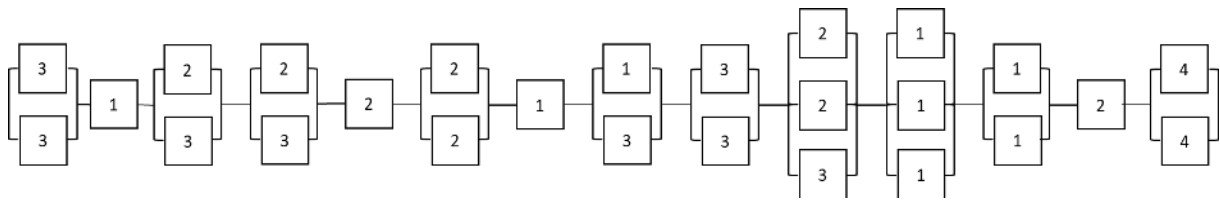
$$R(t) = e^{-\left(\frac{t}{A}\right)^B} \quad (4.3)$$

Bileşenler için Weibull dağılımının parametreleri güvenilirlik değerleri kullanılarak belirlenmiştir. [115].

4.2.3.3 Kesikli Olaylı Monte Carlo Benzetiminin deneme sayısının belirlenmesi

Arıza ve tamirin dikkate alındığı durumda YBTP güvenilirlik eniyilemesinde GA ve MA'nın başarımının incelenmesinde Tablo 4.1- 20 numarada verilen, alan yazında günümüze dek kullanılagelen Fyffe vd. [49]'nin 33 test problemi verileri kullanılmıştır. Bu problemler için Kesikli Olaylı Monte Carlo benzetimi ile ilgili genel bir deneme sayısının belirlenmesi amacıyla aşağıdaki analizler yapılmıştır:

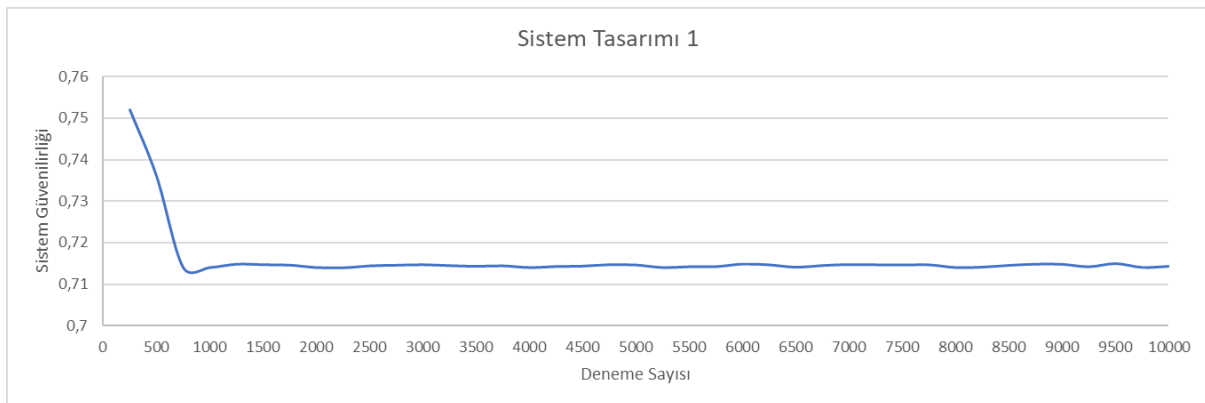
- i) Aşağıda verilen 3 sistem tasarımı Tablo 5.1'de verilen bileşen özellikleri kullanılarak rassal olarak oluşturulmuştur. Bu örnek sistem tasarımlarından biri Şekil 4.5'te görülmektedir. Sistem güvenilirlik değerleri ise analitik yöntemlerle hesaplanmıştır.



Şekil 4.5. Örnek Sistem Tasarımı 1 (S1)

- ii) İkinci adımda, aynı veriler kullanılarak $t=10$ zaman birimi için her bir bileşenin arıza dağılımı için Weibull dağılım parametreleri belirlenmiştir [115]. KOB modelinde bileşen arızalanma süreleri bu şekilde Weibull dağılımından üretilmiştir.
- iii) Kesikli Olaylı Monte Carlo benzetiminde yeterli deneme sayısının belirlenmesi amacıyla (sistem güvenilirliği değerinin tahmini için yeterli deneme sayısı) artan sayıda deneme gerçekleştirilerek sonuçlar elde edilmiştir.

Elde edilen sonuçlara göre analitik olarak hesaplanan sistem güvenilirliği 0,7145 olan ve şekil 4.5'te verilen S1'in sistem güvenilirliğinin tahmin değerinin değişimi şekil 4.6'da verilmiştir. Grafikte görüldüğü üzere, yaklaşık 1000 deneme sonrasında analitik çözüme yakınsama sağlanabilmektedir. Benzer şekilde üretilen S2 ve S3 sistemleri, ilgili analizler ve sonuçları EK-1a'da verilmiştir.

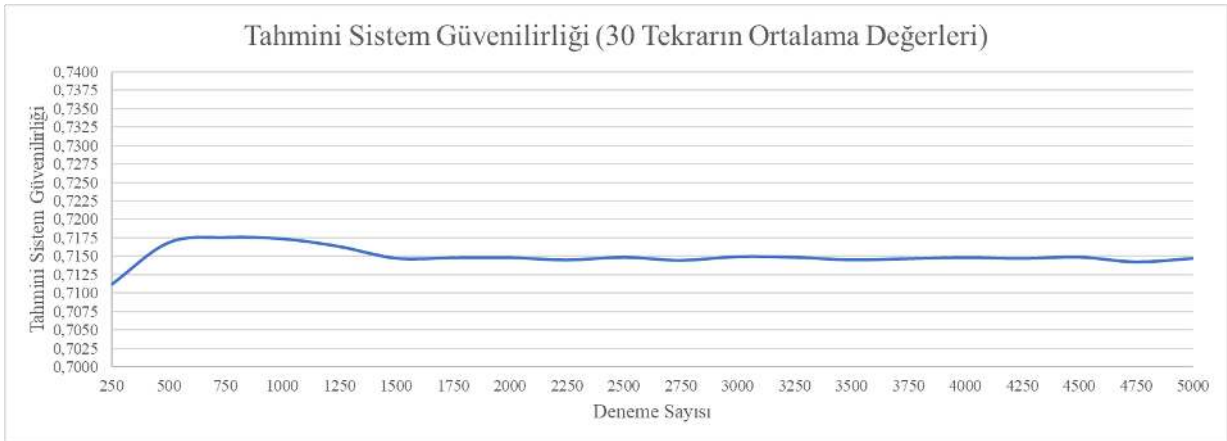


Şekil 4.6. Tahmini sistem güvenilirliği değerleri

Bu analize göre 3 farklı sistem tasarımı için farklı deneme sayılarında yakınsamanın sağlandığı görülmektedir. Bu tezde geliştirilen GA ve MA algoritmalarının başarımlarını analiz için alan yazında verilen Fyffe vd. [49] problemleri için Kesikli Olaylı Monte Carlo benzetiminde gereksinim duyulacak ortak deneme sayısı (genel yakınsamayı sağlayan ortak deneme sayısı) aşağıda açıklanan analizle belirlenmiştir.

Merkezi Limit Teoremi (MLT)'ne göre bu problemlerin farklı deneme sayıları için 30'ar tekrar yapılmış, elde edilen sistem güvenilirlikleri tahminlerinin ortalama tahmin değerleriyle analitik olarak hesaplanan sistem güvenilirliği değerleri karşılaştırılmıştır. Ortalama değerlerin değişimi incelenerek yeterli olabilecek deneme sayısı belirlenmiştir.

Kesikli Olaylı Monte Carlo benzetiminin deneme sayısı (çalışma uzunluğu) 250 ile 5000 deneme arasında dikkate alınmıştır. Elde edilen sonuçlar ve sapma miktarları Ek 1b'de, özet bilgi sunmak üzere örnek bir sistem tasarımı için çizilen grafik ise Şekil 4.7'de verilmiştir. Bu incelemeler sonucunda benzetim modeli ile sistem güvenilirliğinin tahmini için 1500 denemenin yeterli olduğu belirlenmiştir.

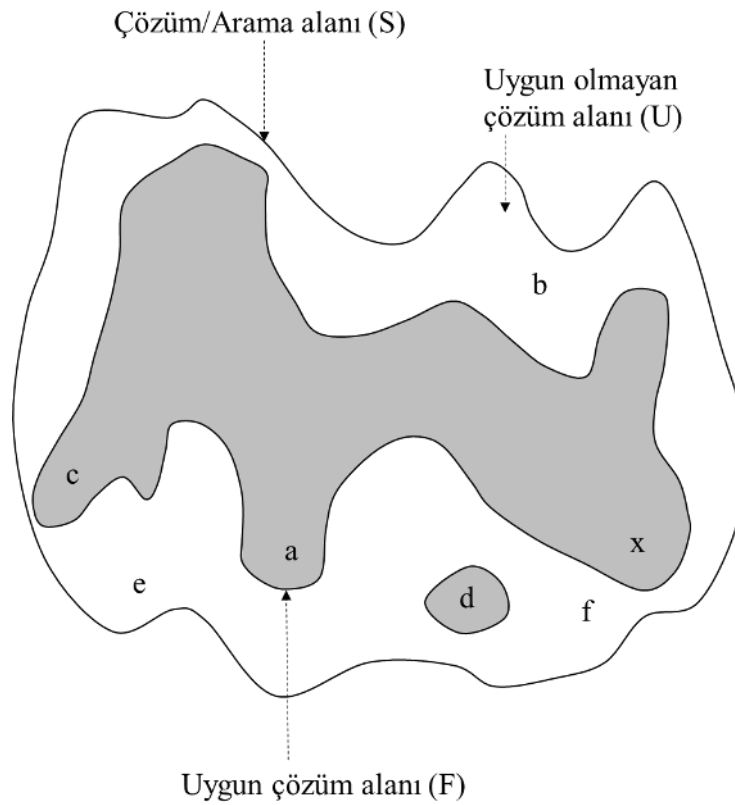


Şekil 4.7. Tahmini Sistem Güvenilirliği (30 Tekrarın Ortalama Değerleri)

4.2.4 Ceza fonksiyonu

Çok kısıtlı eniyileme problemlerinde, çözüm arama sırasında hareket mekanizmalarının uygulanması sonucunda kısıtları sağlamayan (uygun olmayan) çözümler elde edilebilmektedir. Bu durumda bulunan çözümler hemen elenerek değerlendirmeye alınmayabilir. Bu yaklaşıma reddetme yaklaşımı denir. Uygun olmayan çözümlerin dikkate alınmaması algoritmanın daha basit bir yapıda çalışmasını sağlar ancak bu yöntemin bazı sakıncaları vardır. Çok kısıtlı eniyileme problemlerinde, bulunabilecek az sayıdaki uygun çözümler için çok zaman harcamaktadır. Ayrıca, sadece uygun çözüm alanında bulunan çözümler araştırıldığı için daha iyi çözümler bulmayı da engellemektedir [116].

Bu tez kapsamında, uygun olmayan çözümlerin de dikkate alınmasıyla daha iyi çözümlerin elde edilmesini amaçlayan cezalandırma yönteminden yararlanılmıştır. Cezalandırma yöntemi, kısıtlı eniyileme problemlerinde yaygın olarak kullanılan basit ve kolay uygulanabilir bir yöntemdir. Bu yöntemde, GA ile elde edilen alternatif çözümler uygun çözüm alanının dışında kalıyorsa ceza miktarı hesaplanır ve amaç fonksiyonuna yansıtılır (enbüyükleme problemlerinde ceza miktarı amaç fonksiyonundan çıkartılır, enküçüklemeye problemlerinde ise ceza miktarı amaç fonksiyonuna eklenir). Şekil 4.8'deki örnek uygun çözüm alanı içerisinde c, a, x ve d elemanları uygun çözüm alanı içinde (F bölgesinde) kalırken b, e ve f elemanları uygun çözüm alanının dışında (U bölgesinde) kalmaktadır [117]. Böyle bir durumda uygun çözüm alanının dışında (U bölgesinde) kalan elemanlar cezalandırılacaktır.



Şekil 4.8. Bir çözüm alanı/arama uzayı örneği [117]

Sezgisel algoritmalarda, statik ya da dinamik cezalandırma yöntemi kullanılabilir [118]. Statik cezalandırma yönteminde, herhangi bir şekilde uygunluğu bozan çözümler için ceza değeri algoritma boyunca değişmezken; dinamik cezalandırma yönteminde arama boyunca jenerasyon sayısına bağlı olarak ceza değeri güncellenir. Adaptif cezalandırma yönteminde ise

herhangi bir şekilde uygunluğu bozan çözümler için ceza değeri arama boyunca yığından gelen bilgiler doğrultusunda güncellenir [116].

Bu tezde algoritmanın işleyişi sırasında uygun olmayan çözümlerin de dikkate alınmasını sağlamak amacıyla Feizabadi ve Jahromi'nin [73] önerdiği ceza fonksiyonu kullanılmıştır. Böylece uygun olmayan çözüm alanında da araştırma yapılarak arama uzayında çeşitliliğin sağlanması yoluyla daha iyi çözümler elde edilebilmektedir.

$g_1(x) \leq b_1$ ve $g_2(x) \leq b_2$ sistem kısıtlarını gösteriyor olsun. m_1 ve m_2 yeterince büyük sayılar iken ceza fonksiyonu eşitlik (4.4)'te verilmiştir. Bu sayede, sadece kısıtlar ihlal edildiğinde karşılık gelen ceza terimi sıfırdan daha büyük bir değer alır ve uygunluk fonksiyonunun değeri azalır.

$$f(t) = R(t) - m_1 \times Maks \left\{ 0, \frac{g_1(x)}{b_1} - 1 \right\} - m_2 \times Maks \left\{ 0, \frac{g_2(x)}{b_2} \right\}, \quad m_1, m_2 > 0 \quad (4.4)$$

Yapılan ön denemeler sonucunda ceza katsayıları, m_1 ve m_2 , sırasıyla 1.0 ve 1.5 olarak belirlenmiştir.

4.2.5 Değerlendirme fonksiyonu

Yığındaki dizilerin bir değerlendirme fonksiyonuyla uygunluk değerleri hesaplanır. Uygunluk değeri, bir sonraki yığını oluşturacak yeni aday çözümlerin elde edilmesi için mevcut yığından hangi aday tasarımların kullanılacağına belirlenmesinde rol oynamaktadır. GA'da kullanılan değerlendirme fonksiyonu, genellikle problemin amaç fonksiyonudur. GA'da kısıtları sağlamayan, yani uygun olmayan çözümler dikkate alınarak daha iyi çözümlere ulaşma şansı artırılabilir. Bu durumda değerlendirme fonksiyonu, amaç fonksiyonu ile ceza fonksiyonundan oluşur [118].

Bu çalışmada kullanılan değerlendirme fonksiyonu, problemin amaç fonksiyonu ile ceza fonksiyonundan oluşmaktadır. Amaç fonksiyonu, aday tasarımının sistem güvenilirliğini hesaplamaktadır ve kesikli olaylı Monte Carlo benzetimi ile tahmin edilir. Algoritmanın işleyişi sırasında kısıtları sağlamayan yani uygun olmayan çözümlerin de dikkate alınmasını sağlamak üzere Feizabadi ve Jahromi'nin [73] önerdiği ceza fonksiyonu kullanılmıştır. Böylece sonraki iterasyonlarda daha iyi çözümler elde edilebilmektedir.

4.2.6 Seçim mekanizması ve genetik operatörler

Başlangıç yığını oluşturulduktan sonra algoritmanın her iterasyonunda, yeni yığının dizileri, bir olasılıklı seçim süreci ile mevcut yığının dizileri arasından seçilir. Yüksek uygunluk değerine sahip diziler, yeni dizilerin (yeni çözümlerin) elde edilmesinde yüksek

olasılığa sahiptir [119]. Doğal yığınların uygunluğu, bireyin büyümesi ve çoğalmasında engellere karşı koyma yeteneği ile belirlenir. Bu doğal seçim, bir dizinin yaşaması yani sonraki yığına geçebilmesi ya da elenmesinde karar verici olarak değerlendirme fonksiyonunun kullanılmasıyla yapay olarak gerçekleştirilir.

GA herhangi bir sorunu çözmek için farklı şekillerde uygulanabilir. Geleneksel genetik algoritma yaklaşımına alternatif olarak sunulan durağan/sabit durum genetik algoritma yaklaşımı ile her iterasyonda yığının tamamı yerine küçük bir bölümü ($N-k$ kadarı) güncellenmektedir [120], [121]. Bu tez kapsamında, her iterasyonda az sayıda çocuk oluşturulan ve bu az sayıdaki çocuğun ebeveynlerle rekabet ettiği durağan GA kullanılmıştır. Durağan/Sabit durum GA, bireylerin uygunluğu ve değerlendirilmesinin hesaplama açısından pahalı olduğu durumlarda daha iyi bir seçim olarak bilinir [122]. Sistem güvenilirliğinin kesikli olaylı Monte Carlo benzetimi ile tahmin edildiği durumda makul sürelerde çözüme ulaşabilmek için durağan/sabit durum GA yaklaşımı kullanılmıştır.

GA'nın önemli adımlarından biri, algoritmanın performansı üzerinde etkili olan parametrelerin belirlenmesidir. Kontrol parametreleri olarak da adlandırılan bu parametrelerden bazıları *yığın genişliği*, *çaprazlama olasılığı*, *mutasyon olasılığı*, *kuşak aralığı* ve *seçim stratejisidir* [123], [124]. Alan yazında parametre seçimi ve adaptasyonu ile ilgili çok sayıda çalışma bulunmaktadır ve parametre değerlerinin en iyi değerleri, problem özelinde değişiklik göstermektedir.

Her iterasyonda oluşturulan yeni çocuk sayısının yığındaki birey sayısına oranına kuşak aralığı denir. Yüksek kuşak aralığı, her bir iterasyonda yığında çok sayıda bireyin değişmesine neden olur [124]. Durağan/Sabit durumlu GA yaklaşımının kullanıldığı çalışmalarda düşük kuşak aralığı kullanılması tercih edilir.

GA'da seçim mekanizmaları olarak Rulet Çemberi, Sıralama, Turnuva ve Denge Durumu Yöntemleri kullanılmaktadır. En basit seçim mekanizması rulet çemberi yöntemidir. Bu yöntemde çember, Y eşit aralığa bölünür. Çemberdeki i 'inci aralık yığındaki i 'inci diziyi temsil etmektedir ve bu aralığın genişliği, bu dizinin seçilme olasılığına eşittir. Bu durumda, çemberdeki aralık genişliklerinin toplamı 1'e eşittir. Seçim aşamasında çember, Y defa çevrilir. Her çevirmede 0-1 aralığında bir sayı üretilir. Üretilen sayının düştüğü aralıktaki dizi, yeni yığına kopyalanır [12]. Rulet çemberi yöntemi, basit bir yöntemdir. Ancak, yeni yığında her dizinin beklenen kopya sayısı ile gerçekleşen kopya sayısı arasında büyük bir fark olduğu için stokastik bir hataya sahiptir. Algoritmanın her tekrarında bu örnekleme hatası artmakta ve

teorik olarak tahmin edilenden çok daha farklı yönlerde aramaya devam edilmektedir. Bu olay algoritmanın zamansız yakınsamasına sebep olabilmektedir [125].

Sıralı Seçim Yöntemi, Baker tarafından önerilmiştir [126]. Bu yöntemde öncelikle yığındaki diziler uygunluk değerlerine göre en iyiden en kötüye doğru sıralanır. En iyi diziden başlamak üzere bir azalan fonksiyon yardımıyla dizilere kopya sayısı atanır. Kullanılan en genel atama fonksiyonu ise doğrusaldır. Bir fonksiyon yardımıyla atanan kopya sayıları yeni yığının oluşturulmasında kullanılır. Alan yazında, bir yığında seçim baskısını kontrol edebilmek için en iyi dizinin kopya sayısının 1 ve 2 arasında olması gerektiği belirtilmektedir.

Sıralı seçim mekanizmasının iki büyük avantajı vardır. Birincisi, Rulet çemberinde var olan örnekleme hatasının olmamasıdır. İkincisi ise, en küçükleme problemlerinde de uygunluk değerlerinin amaç fonksiyonu değerlerine eşit olarak alınabilmesidir.

Sıralı seçim yönteminin avantajlarına sahip olan Turnuva Yönteminde yığından rassal olarak bir grup dizi seçilir. Bu grup içindeki en iyi uygunluk değerine sahip dizi yeni yığına kopyalanır. Yığın genişliğine ulaşınca kadar bu işleme devam edilir. Genellikle grup genişliği ikidir. Ancak bu sayının artırılması da mümkündür [12], [127]. Tez kapsamında YBTP için geliştirilen GA'da grup genişliği iki olan Turnuva Yöntemi kullanılmıştır.

Denge Durumu Seçim Yönteminin işleyişi diğerlerinden biraz farklıdır. Anlatılan mekanizmalarda öncelikle mevcut yığından dizilerin seçimi ile yeni yığın oluşturulur. Oluşturulan bu yeni yığına genetik operatörler uygulanarak yeni diziler elde edilir. Elde edilen yeni diziler, kendilerini oluşturmakta kullanılan diziler ile yer değiştirerek yeni yığına alınır. Bu yöntemde ise, öncelikle doğrusal sıralı seçim mekanizması kullanılarak seçilen bir ya da iki bireye genetik operatörler uygulanır. Elde edilen yeni diziler mevcut yığındaki uygunluk değeri en düşük diziler ile yer değiştirerek yeni yığın oluşturulur.

Yukarıda açıklanan yöntemlerle arama, yüksek uygunluğa sahip bölgelere doğru yönlendirilir. Bu bölgelerdeki yeni çözüm noktalarına ulaşmada ise genetik operatörler kullanılır. Genetik operatörler, yığının genetik bilgilerini kullanarak yeni çözüm elde ederler. GA'da yeni çözümlerin elde edilmesi, çaprazlama ve mutasyon adı verilen genetik operatörler aracılığıyla gerçekleştirilir. Kullanılan dizi gösterimine uygun olarak seçilen bu operatörler GA'nın en önemli parçalarıdır ve algoritmanın performansı üzerinde önemli bir etkiye sahiptir.

Çaprazlama, farklı çözümler arasında bilgi değişimini sağlayarak arama uzayının benzer ancak araştırılmamış bölgelerine ulaşmayı sağlar [125]. Çaprazlama operatörü, belirli bir olasılıkla uygulandığı yığından rassal olarak seçilen iki dizinin belirli bölümlerini karşılıklı

olarak değiştirerek, arama uzayında yeni noktaları verecek yeni iki farklı diziyi elde eder [128]. Tez kapsamında geliştirilen GA’da tamsayı kodlama için alan yazında bilinen ve yaygın olarak kullanılan düzgün (uniform) çaprazlama operatörü kullanılmıştır. Düzgün çaprazlama operatörü ilk olarak Syswerda [129] tarafından önerilmiştir. Düzgün çaprazlama, bir ve iki noktalı çaprazlamaya kıyasla dizide daha büyük farklılıklar yaratabilir; yani arama uzayının farklı konumlarına daha erken geçebilme yeteneğine sahiptir [130]. Tez kapsamında düzgün çaprazlama operatörünün bir diğer kullanım nedeni de dizide her bir alt sistemde bulunabilecek bileşen türleri ve özelliklerinin farklı olmasıdır. Bu sayede, alt sistemlere kendi içlerinde çaprazlamaya uygulanabileceği için uygun olmayan dizilerin/çözümlerin oluşması engellenmiştir. Şekil 4.9’da verilen örnek, bir diziyeye düzgün çaprazlama operatörünün uygulanışını göstermektedir. Bu operatörün uygulanabilmesi için öncelikle uzunluğu, yığındaki dizilerin uzunluğuna eşit olan geçici dizi adıyla anılan ikili düzende bir dizi oluşturulur. Bu dizide, 0 ve 1 değerleri %50 olasılıkla rassal olarak elde edilir. Geçici dizideki “1” değerleri, Çocuk 1 yeni dizisinde Ebeveyn 1’den seçim yapılacağını, “0” elemanı ise Ebeveyn 2’den seçim yapılacağını gösterir. Çocuk 2’nin oluşumu ise “0” değeri ile Ebeveyn 1’den “1” değeri ile ise Ebeveyn 2’den seçim yapılacağını gösterir. Ebeveyn 1 ve Ebeveyn 2 dizilerinden bu operatörle yeni iki dizi, Çocuk 1 ve Çocuk 2, Şekil 8’deki örnekte görüldüğü gibi elde edilir [129].

	1	2	3	4	5	6	7	8	9	10
Ebeveyn 1	1	2	2	4	2	3	1	4	2	3
Ebeveyn 2	2	3	4	1	3	1	2	3	1	4
Geçici Dizi	1	0	0	1	1	0	1	0	0	1
Çocuk 1	1	3	4	4	2	1	1	3	1	3
Çocuk 2	2	2	2	1	3	3	2	4	2	4

Şekil 4.9. Uniform Çaprazlamaya bir örnek

Mutasyon operatörü, küçük bir olasılıkla bir dizi içerisindeki bir veya birkaç değeri rassal olarak değiştirerek yığında yeni dizilerin yani, arama uzayında farklı yeni çözüm noktalarının elde edilmesini sağlar [12]. Bu tezde dizi eleman değerleri, olası değerler arasından herhangi birine rassal olarak değiştirilerek yeni dizi elde edilir. Bu şekilde genlerin rassal olarak değiştirme işlemi, tamsayı kodlamanın kullanıldığı problemlerde yaygın olarak kullanılmaktadır [131]. Şekil 4.10’da verilen örnek, bir diziyeye mutasyon operatörünün uygulanışını göstermektedir. Çaprazlama sonrası elde edilen dizi içerisinde rassal olarak seçilen sıradaki bileşen, ilgili sırada bulunabilecek bileşen türleri içerisinde rassal olarak seçilen bir bileşen türüyle değiştirilir. Rassal olarak seçilen 3’üncü elemanı (4 numaralı

bileşen türü), ilgili konumda bulunabilecek 1,2,3 ve 4 numaralı bileşen türleri içerisinde rassal olarak seçilen 3 numaralı bileşen türüyle değiştirilir.

Çocuk	1	3	4	4	2	1	1	3	1	3
Çocuk Yeni	1	3	3	4	2	1	1	3	1	3

Şekil 4.10. Mutasyona bir örnek

Yapılan ön denemeler sonucunda algoritmanın 1000 iterasyon çalıştırılması durdurma koşulu olarak seçilmiştir. Geliştirilen durağan/sabit durum GA için oluşturulan sözde kodu Şekil 4.11’de verilmektedir.

```

N bireyden oluşan başlangıç yığını oluştur
For i= 1 to N (yığındaki tüm bireyler için)
    Bireylerin uygunluk değerini hesapla
End for
Uygunluk değerine göre büyükten küçüğe doğru sırala
While not {bitirme koşulu} do
    En iyi ilk “l” adet bireyi yeni yığına taşı (l: Doğrudan taşınan birey sayısı)
    For i= l to N (Kalan (N-l) bireyden oluşan kuşak aralığına ulaşana kadar oluşturulan
        yeni çocuk sayısı)
        İkili turnuva seçim mekanizması ile ebeveyn seç
        Düzgün çaprazlama ile çocuk oluştur
        Mutasyon ile çocukları değişime uğrat
        Çocukları yeni yığına ekle
    End for
    Bireylerin uygunluk değerini hesapla
    Uygunluk değerine göre büyükten küçüğe doğru sırala
End while

```

Şekil 4.11 Durağan/Sabit durum GA sözde kodu

Arıza ve tamirlerin dikkate alınarak sistem güvenilirliğinin kesikli olaylı Monte Carlo benzetimi ile tahmin edildiği ve yedekliğin kullanıldığı sistem güvenilirliğinin GA ile eniyilenmesi için oluşturulan sözde kodu Şekil 4.12’de verilmiştir.

```

N bireyden oluşan başlangıç yığının oluştur
For  $i = 1$  to N (yığındaki tüm bireyler için)
    sayaç  $\leftarrow 0$ 
    For  $rep = 1$  to Rep
        Sistemde yer alan altsistem ve bileşenleri belirle
        For  $i = 1$  to s
            While not {  $t_{benzetim} > t_{son}$  } do
                 $t_{benzetim} \leftarrow 0$ 
                 $j$  “eleman” 1 to  $n_i$ 
                    Bileşen durumları  $\leftarrow$  çalışır
                    Bileşenlerin arızalanma zamanlarını üret
                 $t_{benzetim} \leftarrow t_{enyakin}$  (En erken arızalanan bileşenin zamanı)
                While not {altsistem arızalı} do
                    Bileşen durumlarını güncelle ve en yakın olay
                    (arızalanma/tamir) zamanlarını üret
                     $t_{benzetim} \leftarrow t_{enyakin}$  (En yakın olay zamanına sahip
                    bileşenin zamanı)
                    break if { $t_{benzetim} > t_{son}$ }
                End while
                if altsistem arızalı
                    sayaç  $\leftarrow$  sayaç +1
                     $t_{benzetim} \leftarrow t_{son}$ 
                     $i \leftarrow i + 1$ 
                End if
            End while
        End for
    End for
    Sistem güvenilirliği  $\leftarrow (Rep - \text{sayaç}) / Rep$  raporla (Amaç Fonksiyonu Değeri)
    Bireylerin uygunluk değerini (=Sistem güvenilirliği-Ceza fonksiyonu değeri) hesapla
End for
Uygunluk değerine göre büyükten küçüğe doğru sırala
While not {bitirme koşulu} do
    En iyi ilk “ $l$ ” adet bireyi yeni yığına taşı ( $l$ : Doğrudan taşınan birey sayısı)
    For  $i = l$  to N (Kalan ( $N - l$ ) bireyden oluşan kuşak aralığına ulaşana kadar oluşturulan
        yeni çocuk sayısı)
        İkili turnuva seçim mekanizması ile ebeveyn seç
        Düzgün çaprazlama ile çocuk oluştur
        Mutasyon ile çocukları değişime uğrat
        Çocukları yeni yığına ekle
    End for
    Bireylerin uygunluk değerini hesapla
    Uygunluk değerine göre büyükten küçüğe doğru sırala
End while

```

Şekil 4.12 Arıza ve tamirin dikkate alınarak yedekliğin kullanıldığı sistem güvenilirliğinin GA ile eniyilenmesi sözde kodu

4.3 Geliştirilen Memetik Algoritma

Memetik algoritma, ilk olarak Norman ve Moscato tarafından 1989 yılında geleneksel GA ve TB algoritmasının bir arada kullanılmasıyla önerilmiştir [132]. Norman ve Moscato tarafından gezgin satıcı problemi için uygulanan bu yaklaşım günümüzde kullanılan birçok MA yapısı için temel oluşturmıştır. MA, genel olarak yerel arama algoritmaları ve genel/küresel arama algoritmalarının bir arada kullanılmasıyla oluşturulur. Yerel arama algoritmaları, araştırmayı yerel eniyi seviyeye veya daha önceden belirlenmiş bir seviyeye ulaştıran algoritmalarlardır. Bu algoritmalar, problemin eldeki bir çözümünü her yeni iterasyonla daha iyi bir çözüme götürmeyi amaçlar ve arama, iyileşmenin durduğu ana kadar devam eder. Yerel arama algoritmalarının eniyi çözüme ulaşması, başlangıç çözümünün küresel eniyinin bulunduğu bölgeye uzaklığına ve iterasyon sayısına bağlıdır. Genel arama algoritmalarında ise, yerel arama algoritmalarından farklı olarak yerel eniyi çözümlerden kaçmak için iyileşme yaratmayan bazı çözümlerin de dikkate alınması sağlanır. Alan yazın incelendiğinde, birçok farklı türde kombinatorial eniyileme probleminin çözümünde örneğin *iletişim ağlarının tasarımı* ([14], [17], [20]), *araç rotalama problemi* ([23], [25], [27]), *proje çizelgeleme problemi* ([28], [31]) gibi problemlerin çözümünde MA'nın kullanıldığı ve başarılı sonuçlar verdiği görülmektedir.

Bu çalışmada, Bölüm 4.2'de detayları verilen GA'yı iyileştirmek amacıyla, yani problemin tanımlı kısıtları altında daha yüksek güvenilirliğe sahip bir sistem tasarımı elde edebilmek amacıyla, yerel arama yaklaşımı olarak aşağıda açıklanan hareket mekanizması kullanılmıştır. GA'ya eklenen bu yerel arama yaklaşımı ile bir aday çözümün komşuları arasında kaçırılmış daha iyi bir komşunun yakalanması amaçlanmıştır. GA içerisinde çaprazlama ve mutasyon operatörleri ile elde edilen her yeni dizinin komşuluğu olarak bir dizi oluşturulur ve içlerinden daha yüksek güvenilirliğe sahip olan dizi yeni yığına alınır. Komşuluk yapısı olarak, yani yeni bir aday çözüm elde etme hareketi, her aday çözümün sistem tasarımı içindeki ilk alt sistemden başlayarak, bileşen bulunmayan boş bir konuma, kısıtları bozmadan en yüksek güvenilirliğe sahip bileşenin eklenmesiyle yapılmaktadır. Bu hareketle elde edilen dizi, mevcut dizinin komşusu olur. Yapılan ön denemeler sonucunda algoritmanın 1000 iterasyon çalıştırılması durdurma koşulu olarak seçilmiştir. MA'nın sözde kodu Şekil 4.13'te görülmektedir.

```

N bireyden oluşan başlangıç yığınınını oluştur
For i= 1 to N (yığındaki tüm bireyler için)
    Bireylerin uygunluk değerini hesapla
End for
Uygunluk değerine göre büyükten küçüğe doğru sırala
While not {bitirme koşulu} do
    En iyi ilk "l" adet bireyi yeni yığına taşı (l: Doğrudan taşınan birey sayısı)
    For i= l to N (Kalan (N-l) bireyden oluşan kuşak aralığına ulaşana kadar oluşturulan
        yeni çocuk sayısı)
        İkili turnuva seçim mekanizması ile ebeveyn seç
        Düzgün çaprazlama ile çocuk oluştur
        Mutasyon ile çocukları değişime uğrat
        While not {uygun değişim yok veya değişim yapıldı} do
            For j=1 to s (s: alt sistem sayısı)
                Sistem kısıtlarını bozmayan ve alt sistemde yer alabilecek en
                uzun Ortalama Arızalanma Zamanına sahip bileşeni bul
                if {bileşen bulunmayan boş alan var}
                    Bileşeni boş alana ekle
                    if {Daha yüksek uygunluk değerine sahip}
                        Yeni bireyi (çocuğun komşusu) yığına taşı
                    else
                        Mevcut bireyi (çocuğu) koru
                    End if
                else
                    uygun değişim yok
                    j ← j+1
                End if
            End for
        End while
    End for
    Bireylerin uygunluk değerini hesapla
    Uygunluk değerine göre büyükten küçüğe doğru sırala
End while

```

Şekil 4.13. MA sözde kodu

Arıza ve tamirlerin dikkate alınarak sistem güvenilirliğinin KOB modeli ile tahmin edildiği ve yedekliğin kullanıldığı sistem güvenilirliğinin MA ile eniyilenmesi için oluşturulan sözde kodu Şekil 4.14'te verilmiştir.

```

N bireyden oluşan başlangıç yığını oluştur
For  $i=1$  to  $N$  (yığındaki tüm bireyler için)
    sayaç  $\leftarrow 0$ 
    For  $rep=1$  to  $Rep$ 
        Sistemde yer alan altsistem ve bileşenleri belirle
        For  $i=1$  to  $s$ 
            While not {  $t_{benzetim} > t_{son}$  } do
                 $t_{benzetim} \leftarrow 0$ 
                 $j$  “eleman” 1 to  $n_i$ 
                    Bileşen durumları  $\leftarrow$  çalışır
                    Bileşenlerin arızalanma zamanlarını üret
                 $t_{benzetim} \leftarrow t_{enyakın}$  (En erken arızalanan bileşenin zamanı)
                While not {altsistem arızalı} do
                    Bileşen durumlarını güncelle ve en yakın olay
                    (arızalanma/tamir) zamanlarını üret
                     $t_{benzetim} \leftarrow t_{enyakın}$  (En yakın olay zamanına sahip
                    bileşenin zamanı)
                    break if { $t_{benzetim} > t_{son}$ }
                End while
                if altsistem arızalı
                    sayaç  $\leftarrow$  sayaç +1
                     $t_{benzetim} \leftarrow t_{son}$ 
                     $i \leftarrow i+1$ 
                End if
            End while
        End for
    End for
    Sistem güvenilirliği  $\leftarrow (Rep - \text{sayaç}) / Rep$  raporla (Amaç Fonksiyonu Değeri)
    Bireylerin uygunluk değerini (=Sistem güvenilirliği-Ceza fonksiyonu değeri) hesapla
End for
Uygunluk değerine göre büyükten küçüğe doğru sırala
While not {bitirme koşulu} do
    En iyi ilk “ $l$ ” adet bireyi yeni yığına taşı ( $l$ : Doğrudan taşınan birey sayısı)
    For  $i=1$  to  $N$  (Kalan ( $N-l$ ) bireyden oluşan kuşak aralığına ulaşana kadar oluşturulan
        yeni çocuk sayısı)
        İkili turnuva seçim mekanizması ile ebeveyn seç
        Düzgün çaprazlama ile çocuk oluştur
        Mutasyon ile çocukları değişime uğrat
        While not {uygun değişim yok veya değişim yapıldı} do
            For  $j=1$  to  $s$  ( $s$ : alt sistem sayısı)
                Sistem kısıtlarını bozmayan ve alt sistemde yer alabilecek en
                uzun Ortalama Arızalanma Zamanına sahip bileşeni bul
                if {bileşen bulunmayan boş alan var}
                    Bileşeni boş alana ekle
                    if {Daha yüksek uygunluk değerine sahip}
                        Yeni bireyi (çocuğun komşusu) yığına taşı
                    else
                        Mevcut bireyi (çocuğu) koru
                End if
            End for
        End for
    End for
End while

```

```

else
    uygun deęişim yok
     $j \leftarrow j+1$ 
End if
End for
End while
End for
Bireylerin uygunluk deęerini hesapla
Uygunluk deęerine göre büyükten küçükęe doęru sırala
End while
```

Şekil 4.14 Arıza ve tamirin dikkate alınarak yedeklięin kullanıldıęı sistem güvenilirlięinin MA ile eniyilenmesi sözde kodu

5. SAYISAL ANALİZLER

Bu bölümde, YTBP için önerilen algoritmaların etkinliğini incelemek amacıyla yapılan sayısal analizler verilmiştir. Önerilen metasezgisel algoritmalar; GA ve MA, Dev-C++ ortamında, C programlama dilinde kodlanmıştır. Denemeler, Intel Core i7 3 GHz hızında işlemciye, 32 GB RAM ara belleğe sahip Win10 Education işletim sistemi ile çalışan bir bilgisayarda gerçekleştirilmiştir. Tezde kullanılan test problemleri, algoritmanın parametre değerlerinin belirlenmesi ve algoritmaların başarımlarının karşılaştırılması detaylı olarak açıklanmıştır.

5.1. Test Problemleri

Tezde, geliştirilen metasezgisel algoritmaların etkinliğini incelemek için Fyffe vd. [50] tarafından oluşturulan ve Nakagawa ve Miyazaki [4] tarafından farklı ağırlık kısıtlarıyla genişletilen 33 test problemi kullanılmıştır.

Fyffe vd. [50] tarafından oluşturulan on dört adet seri bağlı her bir alt sistemde üç veya dört alternatif bileşen bulunan test problemi Tablo 5.1’de verilmiştir. Bu test problemlerinde maliyet sınırı 130 birim, ağırlık sınırı 170 birimdir. Bu problem, Nakagawa ve Miyazaki [4] tarafından ağırlık sınır değerleri 159, 160, ... , 190, 191 olacak şekilde değiştirilerek farklı ağırlık kısıtlarıyla 33 test problemine çıkarılmıştır.

Tablo 5.1. Test probleminin özellikleri

Alt sistem (i)	Bileşen Türü (j)											
	1			2			3			4		
	r_{ij}	c_{ij}	w_{ij}	r_{ij}	c_{ij}	w_{ij}	r_{ij}	c_{ij}	w_{ij}	r_{ij}	c_{ij}	w_{ij}
1	0,9	1	3	0,93	1	4	0,91	2	2	0,95	2	5
2	0,95	2	8	0,94	1	10	0,93	1	9	-	-	-
3	0,85	2	7	0,9	3	5	0,87	1	6	0,92	4	4
4	0,83	3	5	0,87	4	6	0,85	5	4	-	-	-
5	0,94	2	4	0,93	2	3	0,95	3	5	-	-	-
6	0,99	3	5	0,98	3	4	0,97	2	5	0,96	2	4
7	0,91	4	7	0,92	4	8	0,94	5	9	-	-	-
8	0,81	3	4	0,9	5	7	0,91	6	6	-	-	-
9	0,97	2	8	0,99	3	9	0,96	4	7	0,91	3	8
10	0,83	4	6	0,85	4	5	0,9	5	6	-	-	-
11	0,94	3	5	0,95	4	6	0,96	5	6	-	-	-
12	0,79	2	4	0,82	3	5	0,85	4	6	0,9	5	7
13	0,98	2	5	0,99	3	5	0,97	2	6	-	-	-
14	0,9	4	6	0,92	4	7	0,95	5	6	0,99	6	9

Alan yazında mevcut olan bu problemin bileşenlerinin belirli bir andaki bilinen güvenilirlik değerleri kullanılarak, bölüm 4.2.4'te açıklandığı şekilde, $t=10$ zaman birimi için her bir bileşenin arıza dağılımını tanımlayacak Weibull dağılım parametreleri belirlenmiştir [115]. Belirlenen t zaman değerinde bileşen güvenilirliklerini temsil eden Weibull dağılım parametreleri Tablo 5.2'de verilmiştir.

Tablo 5.2. Test probleminin özellikleri

Alt sistem (i)	Bileşen Türü (j)															
	1				2				3				4			
	A	B	c_{ij}	w_{ij}	A	B	c_{ij}	w_{ij}	A	B	c_{ij}	w_{ij}	A	B	c_{ij}	w_{ij}
1	85,27	1,05	1	3	89,00	1,20	1	4	85,55	1,10	2	2	132,34	1,15	2	5
2	169,24	1,05	2	8	112,42	1,15	1	10	121,62	1,05	1	9	-	-	-	-
3	45,45	1,20	2	7	70,77	1,15	3	5	60,03	1,10	1	6	106,55	1,05	4	4
4	40,56	1,20	3	5	55,53	1,15	4	6	52,16	1,10	5	4	-	-	-	-
5	112,42	1,15	2	4	121,62	1,05	2	3	118,84	1,20	3	5	-	-	-	-
6	546,05	1,15	3	5	347,17	1,10	3	4	208,21	1,15	2	5	210,36	1,05	2	4
7	71,54	1,20	4	7	86,74	1,15	4	8	141,56	1,05	5	9	-	-	-	-
8	36,61	1,20	3	4	70,77	1,15	5	7	85,55	1,10	6	6	-	-	-	-
9	208,21	1,15	2	8	799,25	1,05	3	9	143,74	1,20	4	7	77,93	1,15	3	8
10	49,54	1,05	4	6	45,45	1,20	4	5	77,35	1,10	5	6	-	-	-	-
11	141,56	1,05	3	5	132,34	1,15	4	6	143,74	1,20	5	6	-	-	-	-
12	35,13	1,15	2	4	43,50	1,10	3	5	56,43	1,05	4	6	70,77	1,15	5	7
13	347,17	1,10	2	5	546,05	1,15	3	5	278,02	1,05	2	6	-	-	-	-
14	77,35	1,10	4	6	79,27	1,20	4	7	169,24	1,05	5	6	654,94	1,10	6	9

5.2. GA Parametrelerinin Belirlenmesi

YBTP için probleme özgü geliştirilen ve önerilen GA'nın performansı, algorithmada kullanılan parametrelerin etkinliğine bağlıdır. GA'nın yığın genişliği (N), iterasyon sayısı, turnuva genişliği ve genin mutasyona uğrama olasılığı parametreleri ön denemelerle belirlenmiştir. Bu ön denemeler sonucunda, yığın genişliği (N)=100, iterasyon sayısı=1000, turnuva genişliği=2 ve genin mutasyona uğrama olasılığı=0,002 olarak alınmıştır. Diğer parametreler ise kuşak aralığı (çocuk oranı), çaprazlama ve mutasyon operatörleri Tablo 5.3'te verilen düzeyler kullanılarak istatistiki deney tasarımı ile elde edilmiştir. Bu amaçla 3^k faktöriyel tasarım uygulanmış ve her faktör kombinasyonu için 5'er deneme yapılmış ve toplamda 135 (=5x27) adet deneme gerçekleştirilmiştir. Varyans analizi (ANOVA) ile faktörlerin başarımlar üzerinde etkili olup olmadığı, DUNCAN çoklu aralık testi ile faktör düzeyleri arasında anlamlı bir farklılığın olup olmadığı test edilmiştir. SPSS paket programı

ile yapılan analizlerde başarımlı ölçütü olarak sistem güvenilirliğinin tahmin edilen değeri dikkate alınmıştır.

Tablo 5.3. Deney tasarımı faktörleri ve düzeyleri

Faktörler	Düzy		
	1	2	3
Kuşak Aralığı (Çocuk Oranı) (KA)	0,10	0,20	0,30
Çaprazlama olasılığı (ÇO)	0,75	0,85	0,95
Mutasyon olasılığı (MO)	0,35	0,65	0,95

Tablo 5.4'te varyans analizi sonuçları verilmiştir. Faktörlerin başarımlı ölçütü üzerinde anlamlı bir etkisinin olup olmadığının incelenmesinde faktörler ve etkileşimlerinin etkileri dikkate alınmıştır.

Tablo 5.4. Varyans analizi sonuçları

Kaynak	Serbestlik Derecesi	Kareler Toplamı	Ortalama Kare	F Değeri	P Değeri
Model	26	0,017 ^a	0,001	9,425	<0,001
Sabit	1	123,433	123,433	1749535,444	<0,001
KA	2	0,014	0,007	100,217	<0,001
ÇO	2	0,000	0,000	1,502	0,227
MO	2	0,001	0,000	6,414	0,002
KA*ÇO	4	0,000	7,98E-5	1,131	0,346
KA*MO	4	0,001	0,000	1,963	0,105
ÇO*MO	4	0,000	3,33E-5	0,472	0,756
KA*ÇO*MO	8	0,001	0,000	1,814	0,082
Hata	108	0,008	7,055E-5		
Toplam	135	123,457			
Düzeltilmiş Toplam	134	0,025			

$R^2=0,694$

^a $\alpha=0,05$ düzeyinde anlamlı

Tablo 5.4'te incelendiğinde, kuşak aralığı (çocuk oranı) ve mutasyon olasılığının $\alpha=0,05$ düzeyinde anlamlı iken, çaprazlama olasılığının performans ölçütü üzerinde anlamlı bir etkisinin olmadığı görülmektedir. Faktör etkileşimleri incelendiğinde hiçbir faktör etkileşiminin performans ölçütü üzerinde anlamlı bir etkiye sahip olmadığı görülmektedir.

Performans ölçütü üzerinde anlamlı etkiye sahip olan faktörlerin düzeyleri arasında anlamlı bir farklılığın olup olmadığının belirlenmesi için Duncan çoklu aralık testi yapılmıştır.

Duncan çoklu aralık testi sonuçları Tablo 5.5'te verilmiş ve faktörler bazında aşağıda incelenmiştir.

Tablo 5.5. Duncan test sonuçları

Faktörler	N	Grup Ortalaması	Duncan Grubu	Düzeyler
Kuşak Aralığı	45	0,9422	A	0,1
	45	0,9599	B	0,2
	45	0,9664	C	0,3
Mutasyon Olasılığı	45	0,9584	A	0,35
	45	0,9577	A	0,65
	45	0,9526	B	0,95

- a) Kuşak Aralığı: Seçilen tüm düzeyler arasında anlamlı bir farklılığın olduğu görülmektedir. Düzey ortalamaları incelendiğinde, kuşak aralığı yani üretilen çocuk sayısı arttıkça algoritma ile elde edilen çözümün, güvenilirlik değerinin tahmininin arttığı görülmektedir.
- b) Mutasyon Olasılığı: Bireylerin mutasyona uğrama olasılığı 0,95 olduğunda, elde edilen düzey ortalamasının diğer iki düzey ortalamasından anlamlı bir farklılığa sahip olduğu görülmektedir. Ancak, mutasyon olasılığı 0,35 ve 0,65 olduğunda bu düzeyler için arasında ortalamalar arası anlamlı bir farklılığın olmadığı, her iki düzey için de başarımlı ölçütünün olumlu yönde etkilendiği görülmektedir.

Bu sonuçlar doğrultusunda, bu çalışmada, kuşak aralığı=30, çaprazlama olasılığı=0,95 ve mutasyon olasılığı=0,35 olarak seçilmiştir.

5.3. GA'nın Doğrulanması ve Geçerliliği

Bu bölümde, GA'nın doğrulanması, problemin akış şeması ile GA için yazılan program kodunun uygunluk kontrolü ile yapılmıştır. GA'nın geçerliliği ise; tamirlerin olmadığı varsayımı altında çalıştırılan GA sonuçları ile analitik modellerle elde edilen sonuçlar karşılaştırılarak gösterilmiştir. Bu amaçla, alan yazında verilen 33 test problemi ile ilgili bilgiler kullanılmıştır. GA, 30 tekrar ile çalıştırılmış ve geçerliliğin sağlandığını gösteren sonuçlar Tablo 5.6'da verilmiştir.

Tablo 5.6. GA Geçerliliği için sonuçlar

Problem No	Analitik olarak Hesaplanan Sistem Güvenilirliği	GA ile Bulunan En Büyük Sistem Güvenilirliği	GA ile Bulunan Sistem Güvenilirliği Değerlerinin Ortalaması	Değişim Katsayısı
1	0,9868	0,98867	0,98222	0,002826
2	0,9864	0,98600	0,98073	0,0039
3	0,9859	0,98733	0,98127	0,00263
4	0,9854	0,98733	0,97987	0,004422
5	0,9847	0,98467	0,97851	0,004371
6	0,9842	0,98600	0,97844	0,004322
7	0,9835	0,98400	0,97691	0,003626
8	0,983	0,98733	0,97869	0,00446
9	0,9823	0,98333	0,97671	0,004072
10	0,9815	0,98400	0,97598	0,004536
11	0,981	0,98067	0,97491	0,003883
12	0,9803	0,97933	0,97347	0,003249
13	0,9795	0,98400	0,97251	0,004592
14	0,9784	0,97800	0,97140	0,004405
15	0,9776	0,97933	0,97018	0,004039
16	0,9767	0,97667	0,96907	0,004724
17	0,9754	0,97667	0,96882	0,004828
18	0,9749	0,97533	0,96820	0,003617
19	0,9738	0,97400	0,96631	0,004324
20	0,973	0,97533	0,96491	0,004833
21	0,9719	0,97733	0,96576	0,004866
22	0,9708	0,97600	0,96289	0,007719
23	0,9693	0,97133	0,96202	0,004775
24	0,9681	0,97067	0,96213	0,004612
25	0,9663	0,97333	0,95940	0,006056
26	0,965	0,97000	0,95691	0,006537
27	0,9637	0,97067	0,95576	0,005659
28	0,9624	0,96667	0,95551	0,005611
29	0,9606	0,96733	0,95478	0,006983
30	0,9592	0,96333	0,95249	0,007119
31	0,958	0,96067	0,95004	0,007194
32	0,9557	0,95733	0,94960	0,004385
33	0,9546	0,95933	0,94896	0,007783

5.4. Başarım Ölçütleri

YBTP için geliştirilen algoritmaların başarımını incelemek amacıyla beş farklı ölçüt kullanılmıştır. Bu ölçütlere ilişkin açıklamalar aşağıdaki gibidir;

Önerilen her iki algoritmanın analizi için tüm test problemlerinde;

- a) En Büyük Sistem Güvenilirliği (ESBG): İlgili test problemi için algoritmanın 15 kez çözdürülmesi ile elde edilen en yüksek sistem güvenilirliği tahmini değeri,
 - b) Ortalama Sistem Güvenilirliği (OSG): İlgili test problemi için algoritmanın 15 kez çözdürülmesi ile elde edilen sistem güvenilirliği tahmini değerlerinin ortalaması,
 - c) Değişim katsayısı (DK): İlgili test problemi için algoritmanın 15 kez çözdürülmesi ile elde edilen sistem güvenilirliği tahmini değerlerinin değişim katsayısı,
 - d) Ortalama iterasyon sayısı (OİS): İlgili test problemi için algoritmanın 15 kez çözdürülmesi ile elde edilen en iyi çözümlerin bulunduğu ortalama iterasyon sayısı,
 - e) Ortalama çözüm süresi (OÇS): İlgili test problemi için algoritmanın 15 kez çözdürülmesi ile elde edilen en iyi çözümlerin bulunduğu ortalama çözüm süresi,
- başarım ölçütleri olarak kullanılmıştır.

5.5. Sayısal Sonuçların Değerlendirilmesi

Bu bölümde, öncelikle sabit “ k ” değerine sahip sistemler için tamir edilebilir özellikteki bileşenlerin kullanılmasıyla daha düşük maliyetle daha yüksek güvenilirliğe sahip olan sistem tasarımları sunulmuştur. Ardından, YBTP için geliştirilen GA ve MA algoritmalarının performansını, belirlenen başarım ölçütlerine göre değerlendirmek amacıyla yapılan analizler sırasıyla yer almaktadır. Daha sonra, talep yoğunluğunun değişkenlik gösterdiği yani değişken “ k ” değerlerinin kullanıldığı durumda sistem tasarımlarında yaşanan değişiklikler incelenmiştir.

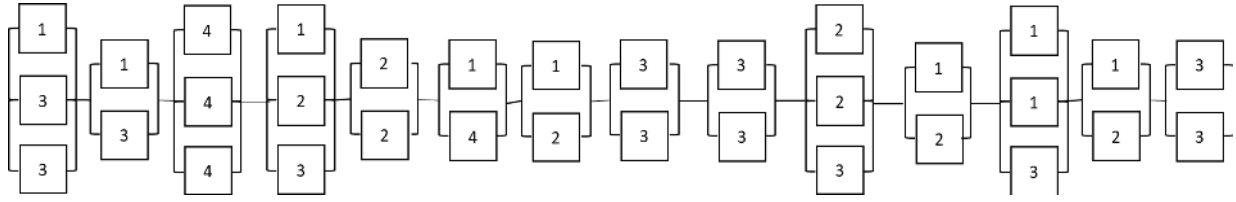
5.5.1. Sabit “ k ” değerine sahip sistemler için analiz sonuçları

Bu bölümde sabit “ k ” değerine sahip sistemlerde tamir edilebilir özellikteki bileşenlerin kullanılmasının sistem tasarımları üzerindeki etkisi incelenmiştir. Ayrıca, sabit “ k ” değerine sahip sistemlerde YBTP için geliştirilen GA ve MA algoritmaları ile elde edilen çözümler incelenerek algoritmaların performansları karşılaştırılmıştır.

5.5.1.1. Tamir edilebilen bileşenlerin kullanılmasının sistem güvenilirliğine ve maliyetine etkisi

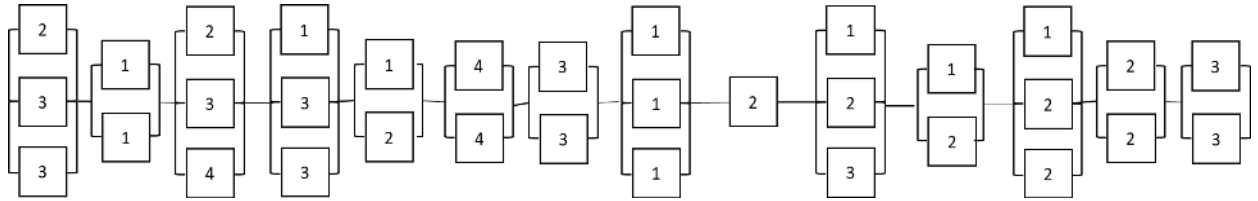
Gerçek hayatta kullanılan sistemlerde, bir bileşenin arızalanması sonrasında devre dışı kalması sistemin güvenilirliğini negatif yönde etkilerken aynı bileşenin tamir edilerek çalışır duruma dönmesi sistemin güvenilirliğini pozitif yönde etkileyecektir. Örnek olarak P22

Problemi için tamir edilen ve edilmeyen bileşenlerin bulunduğu örnek sistemler için tasarımlar ile güvenilirlik ve maliyet değerleri Şekil 5.1 ve şekil 5.2’de verilmiştir.



Şekil 5.1 P22 problemi için tamir edilmeyen bileşenlerin yer aldığı örnek sistem tasarımı

Şekil 5.1’de verilen tamir edilmeyen sistemin güvenilirlik değeri 0.9740; maliyeti 112 birim ve ağırlığı 170 birimdir.



Şekil 5.2 P22 problemi için tamir edilen bileşenlerin yer aldığı örnek sistem tasarımı

Şekil 5.2’de verilen tamir edilen sistemin güvenilirlik değeri 0.9867; maliyeti 104 birim ve ağırlığı 170 birimdir.

Tamirin dikkate alınmadığı durum için analitik olarak hesaplanabilen sistem güvenilirliği değerlerinin, aynı sistem için tamir edilebilir bileşen kullanıldığında elde edilen sistem güvenilirliği değerinin daha büyük olduğu görülmüştür. Tamir edilebilen bileşen kullanmanın sistem güvenilirliği (Tablo 5.7) ve maliyeti (Tablo 5.8) üzerindeki olumlu etkisini gösteren sonuçlar detaylı verilmiştir. Tablolardan genel olarak görüleceği üzere, her bir problem için bileşenlerin tamir edildiği durumda sistem güvenilirlik ortalama tahmin değerlerinin daha yüksek olduğu ve bu durum için aynı/yakın güvenilirlik değerine sahip sistem tasarımlarının yaklaşık ortalama %23 kadar düşük maliyetle kurulabildiği görülmektedir. Bir problem üzerinde örnek vermek gerekirse, P22 Problemi için tamir edilebilir ve edilemez özellikteki bileşenlerin kullanılmasıyla elde edilen ve aynı/yakın güvenilirlik değerine sahip olan iki farklı sistem tasarımının maliyetleri Ek 2’de verilmiştir. Bu tablo incelendiğinde tamir edilebilir bileşenlerden oluşan sistemin maliyetinin daha düşük olduğu açıkça görülmektedir. Burada tamir edilmeyen bileşenlerden oluşan sistem için en iyi çözümde 120 birimlik maliyetle güvenilirlik 0,9708 elde edilmiştir. Tamir edilebilir bileşenlerin kullanıldığı sistem için ise maliyet kısıtı azaltılarak farklı kısıt değerleri için

problem çözümü tekrarlanmış ve aynı güvenilirlik değerine sahip olan sistem tasarımının en düşük maliyetlisi araştırılmıştır. Tamir edilebilir bileşenlerin kullanıldığı sistem için en iyi çözüm (güvenilirlik değeri 0,9727 olan çözüm) 80 birimlik maliyetle elde edilebilmiştir. Böylece, bu problem için tamir edilebilir bileşenlerin kullanılmasıyla elde edilen sistem tasarımının, tamir edilemeyen bileşenler kullanıldığında aynı güvenilirlik değerine sahip olan sistem tasarımından yaklaşık %33 daha düşük maliyetle elde edilebildiği görülmüştür. Benzer şekilde detaylı analizler tüm problemler için yapılmış, tezde fazla yer tutmaması bakımından %50'si Ek 2'de sunulmuştur.

Elde edilen çözüm bileşenlerin tamir edilme durumunda elde edilmesi beklenen bir sonuç olup hedeflenen sonuca ulaşıldığı bu analizlerle gösterilmiştir.

Tablo 5.7. Tamir edilebilen bileşen kullanmanın sistem güvenilirliği üzerindeki etkisi

Problem No	Tamir edilemeyen bileşenlerle oluşturulan sistem güvenilirliği	Tamir edilebilen bileşenlerle oluşturulan sistem güvenilirliği	Tamir edilebilen bileşenlerle oluşturulan sistem güvenilirliği ortalaması
P1	0,9868	0,997333	0,9953332
P2	0,9864	0,996667	0,994755533
P3	0,9859	0,997333	0,994177667
P4	0,9854	0,997333	0,9951556
P5	0,9847	0,997333	0,994088867
P6	0,9842	0,996667	0,992400133
P7	0,9835	0,994667	0,9931112
P8	0,983	0,997333	0,993422267
P9	0,9823	0,995333	0,991688933
P10	0,9815	0,995333	0,991955467
P11	0,981	0,994	0,9894222
P12	0,9803	0,995333	0,991200067
P13	0,9795	0,992667	0,989733333
P14	0,9784	0,994	0,9890666
P15	0,9776	0,993333	0,988888933
P16	0,9767	0,993333	0,9884
P17	0,9754	0,994667	0,985333333
P18	0,9749	0,991333	0,984577867
P19	0,9738	0,991333	0,986622133
P20	0,973	0,992667	0,9858222
P21	0,9719	0,990667	0,9851112
P22	0,9708	0,988667	0,981689
P23	0,9693	0,988	0,9838666
P24	0,9681	0,988667	0,983866733
P25	0,9663	0,986667	0,979600067
P26	0,965	0,988667	0,979333333
P27	0,9637	0,987333	0,9802666

P28	0,9624	0,984	0,974533333
P29	0,9606	0,985333	0,9748888
P30	0,9592	0,981333	0,973999933
P31	0,958	0,982667	0,9721334
P32	0,9557	0,980667	0,9693778
P33	0,9546	0,982667	0,970666667

Tablo 5.8. Tamir edilebilen bileşen kullanmanın sistem maliyeti üzerindeki etkisi

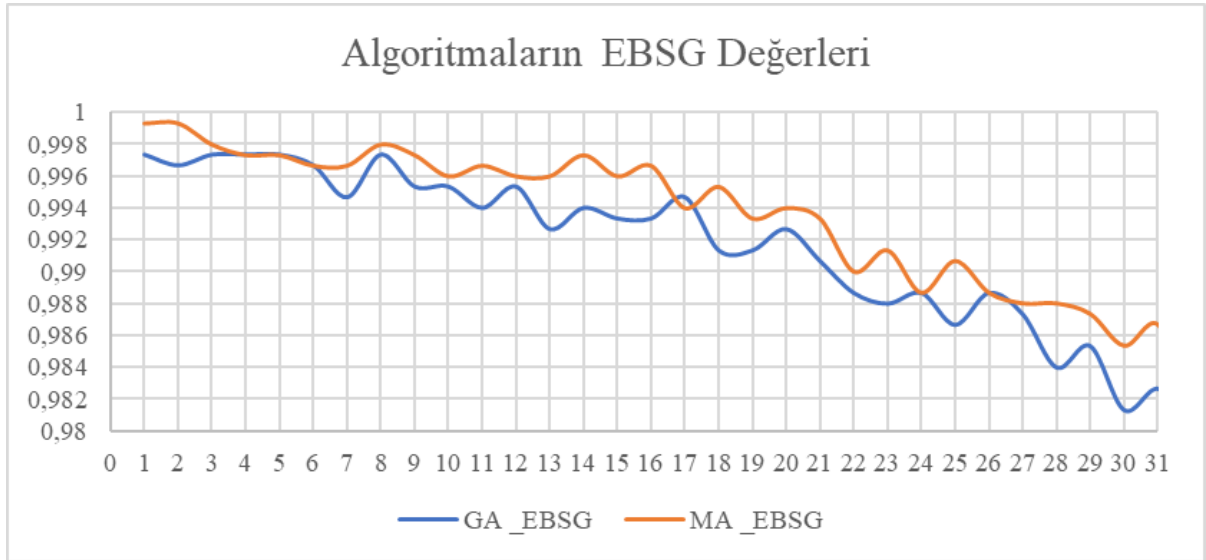
	Tamir edilemeyen bileşenlerle oluşturulan sistem		Tamir edilebilen bileşenlerle oluşturulan sistem	
Problem No	Sistem Güvenilirliği	Maliyet	Sistem Güvenilirliği	Maliyet
P1	0,9868	130	0,990667	90
P2	0,9864	129	0,988	97
P3	0,9859	130	0,988	97
P4	0,9854	130	0,988667	88
P5	0,9847	130	0,987333	87
P6	0,9842	129	0,988	100
P7	0,9835	130	0,990667	100
P8	0,983	130	0,990667	100
P9	0,9823	128	0,988	98
P10	0,9815	127	0,986	98
P11	0,981	125	0,988	100
P12	0,9803	128	0,984	100
P13	0,9795	126	0,984	100
P14	0,9784	125	0,978667	94
P15	0,9776	126	0,982667	97
P16	0,9767	125	0,982667	95
P17	0,9754	125	0,976667	94
P18	0,9749	123	0,978	95
P19	0,9738	122	0,980667	93
P20	0,973	123	0,971333	95
P21	0,9719	122	0,976	98
P22	0,9708	120	0,972667	80
P23	0,9693	121	0,978667	97
P24	0,9681	119	0,974	102
P25	0,9663	118	0,972	96
P26	0,965	116	0,98	94
P27	0,9637	117	0,981333	98
P28	0,9624	115	0,971333	89
P29	0,9606	114	0,978	102
P30	0,9592	115	0,962667	98
P31	0,958	113	0,971333	91
P32	0,9557	112	0,958	89
P33	0,9546	110	0,964	89

5.5.1.2. GA ve MA algoritmalarının karşılaştırılması

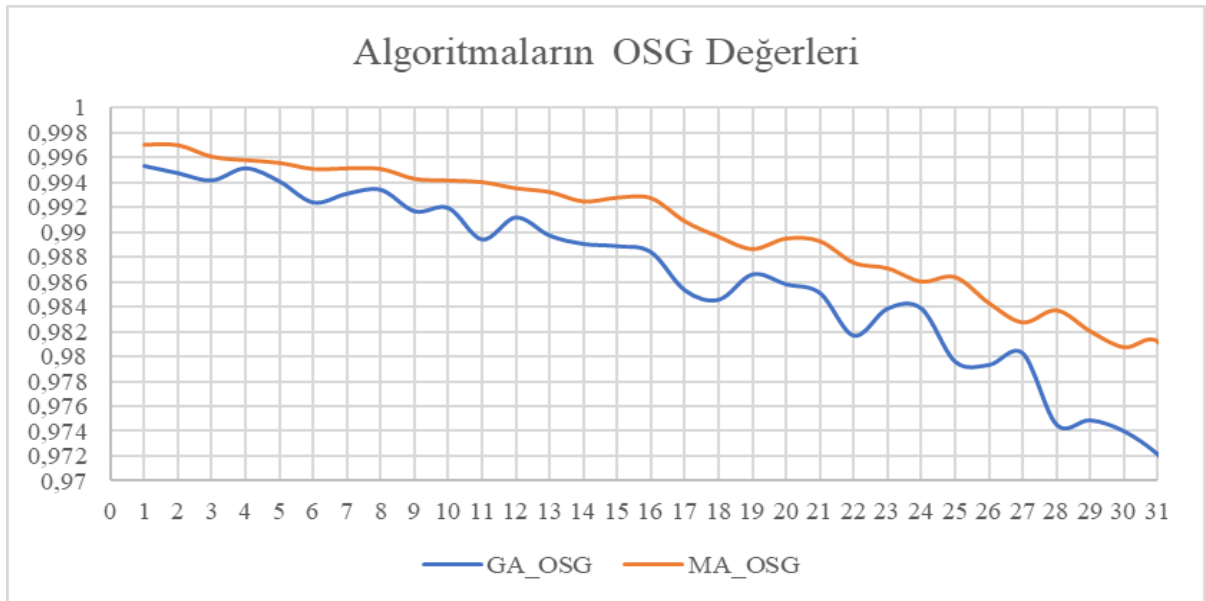
Bölüm 5.4'te belirlenen başarımlar ölçütlerine göre, önerilen GA ve MA algoritmaları ile elde edilen sonuçlar Tablo 5.9'da; bu veriler doğrultusunda oluşturulan grafik Şekil 5.1 ve Şekil 5.2'de verilmiştir. GA ve MA kullanılarak oluşturulan sistem tasarımları, maliyet ve ağırlık değerleri ile sistem güvenilirliği değerleri Tablo 5.10(a-b)'da yer almaktadır.

Tablo 5.9. GA ve MA sonuçları

Problem No	GA					MA				
	EBSG	OSG	DK	OİS	OÇS	EBSG	OSG	DK	OİS	OÇS
P1	0,997333	0,995333	0,001498	493,07	359,42	0,999333	0,997067	0,001207	499,53	728,27
P2	0,996667	0,994756	0,001157	556,40	315,30	0,999333	0,997022	0,001038	567,40	643,07
P3	0,997333	0,994178	0,003624	525,53	297,56	0,998	0,996089	0,000834	595,67	674,53
P4	0,997333	0,995156	0,001851	520,87	308,57	0,997333	0,995822	0,000928	564,87	669,27
P5	0,997333	0,994089	0,002866	590,33	292,75	0,997333	0,9956	0,00094	660,53	655,13
P6	0,996667	0,9924	0,002419	625,47	337,44	0,996667	0,995111	0,000865	599,13	646,47
P7	0,994667	0,993111	0,001733	492,80	280,12	0,996667	0,995155	0,000819	583,13	662,93
P8	0,997333	0,993422	0,002165	534,80	313,61	0,998	0,995111	0,001258	547,40	642,00
P9	0,995333	0,991689	0,002435	499,47	252,08	0,997333	0,994311	0,002236	659,20	665,40
P10	0,995333	0,991955	0,003032	647,60	379,72	0,996	0,994178	0,001254	541,60	635,13
P11	0,994	0,989422	0,003724	523,53	282,34	0,996667	0,994045	0,001279	591,27	637,73
P12	0,995333	0,9912	0,00389	533,40	269,37	0,996	0,993556	0,001493	612,73	618,87
P13	0,992667	0,989733	0,002619	552,33	306,04	0,996	0,993244	0,000945	615,60	682,20
P14	0,994	0,989067	0,003937	694,33	434,91	0,997333	0,992489	0,002131	503,53	630,80
P15	0,993333	0,988889	0,004153	566,87	270,59	0,996	0,9928	0,002315	673,67	643,13
P16	0,993333	0,9884	0,003736	566,80	277,42	0,996667	0,992756	0,001897	663,40	649,40
P17	0,994667	0,985333	0,00461	645,60	308,05	0,994	0,990889	0,001981	652,20	622,40
P18	0,991333	0,984578	0,00462	611,20	321,60	0,995333	0,989644	0,002968	569,33	599,13
P19	0,991333	0,986622	0,004456	536,33	298,21	0,993333	0,988667	0,00308	553,33	615,33
P20	0,992667	0,985822	0,003988	626,93	277,05	0,994	0,989511	0,002953	676,07	597,53
P21	0,990667	0,985111	0,003587	644,07	322,56	0,993333	0,989289	0,001717	611,20	612,20
P22	0,988667	0,981689	0,005897	598,60	284,45	0,99	0,987555	0,002361	657,27	624,67
P23	0,988	0,983867	0,003633	668,00	330,39	0,991333	0,987111	0,002548	585,73	579,40
P24	0,988667	0,983867	0,004059	641,73	311,79	0,988667	0,986045	0,00192	594,07	577,27
P25	0,986667	0,9796	0,004831	673,67	260,44	0,990667	0,9864	0,002119	750,20	580,07
P26	0,988667	0,979333	0,006239	645,20	300,81	0,988667	0,984311	0,002756	643,53	600,07
P27	0,987333	0,980267	0,004222	625,13	281,49	0,988	0,982756	0,002306	646,47	582,20
P28	0,984	0,974533	0,005954	586,93	319,02	0,988	0,983733	0,003093	514,53	559,33
P29	0,985333	0,974889	0,006709	574,67	287,78	0,987333	0,982044	0,002229	560,13	561,00
P30	0,981333	0,974	0,005457	580,87	278,57	0,985333	0,980756	0,002871	597,60	573,20
P31	0,982667	0,972133	0,006233	738,73	370,48	0,986667	0,9812	0,003476	638,60	640,53
P32	0,980667	0,969378	0,00816	585,07	264,05	0,982	0,976667	0,002656	597,13	539,00
P33	0,982667	0,970667	0,008327	779,60	398,12	0,982667	0,977556	0,003416	533,93	545,33
Genel Ortalama	0,991313	0,985591	0,004116	596,54	308,85	0,993151	0,989651	0,001997	601,82	621,00



Şekil 5.3. EBSG değerlerine göre algoritma sonuçlarının karşılaştırılması



Şekil 5.4. OSG değerlerine göre algoritma sonuçlarının karşılaştırılması

Her iki algoritma ile elde edilen OSG sonuçları karşılaştırıldığında, tüm test problemleri için MA'nın GA'dan daha yüksek güvenilirlik değerleri bulduğu görülmüştür. MA, EBSG açısından problemlerin %18.18'inde GA'nın başarımına eşdeğer, %78.78'inde ise daha iyi sonuçlar bulmuştur.

Her iki algoritma için OİS sonuçları incelendiğinde, test problemlerinin çoğunda MA'nın en iyi çözümü bulduğu OİS'nin GA'nın en iyi çözümü bulduğu OİS'den büyük olduğu görülmektedir. Bu MA'nın yapısı gereği beklenen bir durumdur. Bölüm 4.3'te açıklandığı gibi MA, GA algoritmasının yığın güncelleme adımında yerel komşu araması yaparak daha iyi bir çözüm ararken çözüm zamanı artmaktadır.

Her iki algoritma için OÇS sonuçları incelendiğinde, güvenilirlik değerindeki çok küçük bir artışın önemi bilindiğinden MA ile artan güvenilirlik artışı dikkate alındığında MA’da da makul sürelerde çözüm bulabildiği söylenebilir.

MA çözüm uzayında daha fazla noktada arama yaparak; daha iyi sonuçlara ulaşabildiği için GA’ya göre çözüm kalitesi bakımından daha iyi sonuçlar (yüksek güvenilirliğe sahip tasarımlar) elde edebilmektedir. Sistem güvenilirliği, günümüzün ileri teknolojilerinin kullanıldığı ve milyonlarca insana aynı anda kesintisiz hizmetin sunulduğu birçok sistem için çok kritik, değerli ve önemli bir ölçüttür. Bu nedenle OSG değerindeki %04’lük (0,989651-0,985591=0.004) artış çok önemli bir kazanımdır. İleri teknolojiye sahip ve milyonlarca kullanıcıya aynı anda hizmet sunan sistemler için sistemin tasarımı aşamasında problemin çözümü için GA’nın işlem zamanı ortalama 308.85 sn iken MA’nın 621.00 sn ile yaklaşık 2 katı zaman harcayarak daha iyi sonuca ulaşması katlanılabilir bir durumdur. Sonuç olarak, bu tezde YBTP için probleme özgü geliştirilen GA ve MA başarılı sonuçlar bulurken, MA’nın daha iyi sonuç buluyor olması nedeniyle tezde önerilen algoritma olarak öne çıkmaktadır.

Maliyet limiti=130			En İyi Sistem Tasarımı																
Problem No	Ağırlık Limiti	Algoritma	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
1	191	GA	0.995333	118	191	3 0 3 0 3	1 0 0 2 0	0 2 2 0 3	0 1 1 1	1 2 0 0 1	1 0 0 2 0	2 0 3	1 1 1 0 0	3 0 3	0 0 3	3 1 2 0 3	2 1 1 2 0	0 0 1 0 1	3 0 0 0 3
		MA	0.998	130	190	3 3 3 3 3	1 0 0 4 4	4 0 0 3 3	1 0 0 3	3 2 2 0 2	2 0 0 2 1	1 0 0 3	1 1 0 0 3	3 0 0 3	2 2 0 0 3	1 0 0 2 3	1 1 1 0 2	2 0 0 0 4	3 0 0 0
2	190	GA	0.997333	118	189	0 1 0 3 1	0 0 1 3 0	4 0 4 3 3	0 1 0 2 3	2 0 2 0 0	0 4 1 2 3	0 0 0 2 1	0 0 1 0 0	2 0 1 2 0	2 3 0 1 3	1 0 0 3 0	0 1 2 2 0	0 0 0 3 0	4 0 0
		MA	0.998667	118	190	3 2 1 0 0	1 1 0 0 4	2 0 0 3 3	2 0 0 3 3	2 0 0 2 2	1 0 0 4 2	0 0 0 3 3	0 1 0 0 2	1 0 0 0 3	3 3 0 0 3	2 0 0 4 1	1 0 0 1 1	0 0 0 4 3	0 0 0
3	189	GA	0.994667	118	188	1 1 3 0 0	4 1 0 1 0	4 2 0 4 0	0 1 2 3 2	3 0 2 0 4	0 1 0 3 1	0 0 0 1 0	1 0 3 1 0	1 0 0 2 0	2 0 2 0 0	3 0 2 0 4	0 1 2 0 4	0 0 0 1 0	1
		MA	0.995333	115	189	3 2 1 0 0	2 1 0 0 4	2 0 0 3 3	1 0 0 2 1	1 0 0 4 1	0 0 0 3 1	1 0 0 1 1	1 0 0 3 2	0 0 0 3 2	2 0 0 3 1	0 0 0 4 1	1 0 0 2 1	0 0 0 3 3	0 0 0
4	188	GA	0.994667	121	184	0 0 1 2 3	1 1 0 0 0	2 4 0 0 2	4 4 0 0 3	0 3 3 0 2	1 0 1 0 2	0 0 1 2 0	0 3 0 0 1	1 1 0 0 3	3 3 2 0 3	2 0 3 0 1	1 0 2 0 1	0 2 0 3 0	0 0 3
		MA	0.996667	121	188	3 3 1 0 0	1 1 0 0 4	2 0 0 3 3	1 0 0 2 1	1 0 0 3 2	2 0 0 3 1	1 0 0 1 1	1 0 0 3 3	0 0 0 3 2	2 0 0 3 1	0 0 0 3 2	1 0 0 1 1	0 0 0 4 1	0 0 0
5	187	GA	0.994	115	184	0 3 2 0 2	0 1 0 1 0	4 0 2 0 3	0 2 3 3 0	0 0 0 1 2	3 0 0 0 0	1 0 1 3 1	0 1 0 0 3	4 0 1 1 0	0 0 3 0 2	2 0 2 0 2	2 0 2 0 2	1 0 2 0 0	4 1 0
		MA	0.995333	120	185	3 3 1 0 0	1 1 0 0 4	2 0 0 3 3	1 0 0 2 1	1 0 0 4 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 4 2	1 0 0 0 3	0 0 0
6	186	GA	0.994667	118	186	3 3 0 3 0	1 0 2 0 4	3 2 0 0 3	3 0 0 0 3	1 0 0 4 1	0 0 0 4 1	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0
		MA	0.996	114	186	3 3 3 0 0	1 1 0 0 4	3 2 0 0 3	1 0 0 1 1	0 0 0 4 1	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0 3 0	0 0 0
7	185	GA	0.998	117	184	0 0 1 3 3	1 0 1 0 0	4 0 2 4 0	3 0 3 0 2	2 0 0 2 0	0 1 1 3 2	1 0 0 0 0	1 1 1 3 2	0 0 0 0 1	1 1 3 2 0	0 0 0 3 0	1 0 2 1 0	2 0 0 0 1	0 1 2
		MA	0.995333	121	184	3 3 3 0 0	1 1 0 0 4	2 0 0 3 3	2 0 0 3 3	2 0 0 3 2	1 0 0 4 2	0 0 0 3 1	0 0 0 3 1	0 0 0 2 1	0 0 0 2 2	2 0 0 3 3	0 0 0 3 3	1 0 0 2 1	0 0 0 4 3
8	184	GA	0.996	109	184	1 1 0 1 3	0 0 1 0 1	4 0 2 0 3	0 0 1 2 2	0 2 0 3 0	0 1 0 0 0	1 0 3 0 1	1 0 2 0 1	0 1 0 2 0	2 2 0 2 1	0 0 0 0 0	1 4 1 0 0	0 2 1 0 0	0 1 4
		MA	0.994667	112	182	3 1 1 0 0	1 1 0 0 4	4 0 0 3 2	1 0 0 3 2	1 0 0 4 2	0 0 0 3 0	0 0 0 3 0	0 0 0 3 1	0 0 0 3 2	0 0 0 3 2	2 0 0 2 1	0 0 0 2 1	0 0 0 2 1	0 0 0 3 0
9	183	GA	0.993333	122	182	0 3 3 0 3	0 1 1 0 0	4 4 0 0 0	0 3 3 2 0	2 0 2 0 2	2 0 1 1 0	0 0 3 3 1	0 0 3 0 1	2 0 0 2 0	3 0 3 0 1	2 0 2 0 3	0 2 0 4 0	2 1 0 1 0	0 4 0 0 3
		MA	0.995333	119	183	3 2 1 0 0	1 1 0 0 4	4 0 0 3 3	1 0 0 2 1	1 0 0 4 2	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 0	0 0 0 3 2	2 0 0 3 2	0 0 0 3 2	1 0 0 2 0	0 4 3 0 0
10	182	GA	0.994	122	182	0 3 3 2 0	1 0 0 0 3	4 4 0 0 0	0 3 2 0 3	2 0 2 0 4	2 0 0 0 2	1 0 0 0 2	1 0 0 0 3	1 0 0 3 1	0 0 0 3 2	2 0 2 0 0	0 0 3 1 3	1 0 3 0 0	2 0 4 0 0 3
		MA	0.996	117	180	3 3 3 0 0	1 1 0 0 4	2 0 0 3 3	1 0 0 2 2	2 0 0 4 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 2	1 0 0 3 1	0 0 0 1 1	1 0 0 2 1	0 0 0 4 3 0 0
11	181	GA	0.995333	115	179	3 3 1 0 0	1 1 0 0 4	4 0 0 4 1	1 0 0 3 0	2 2 0 1 0	2 0 4 3 3	0 0 0 1 0	1 0 3 0 1	1 3 0 0 0	2 0 2 0 2	2 3 3 0 0	0 0 1 0 1	2 0 2 0 2	0 3 0 4 0
		MA	0.995333	115	179	3 3 1 0 0	1 1 0 0 4	4 0 0 3 3	3 0 0 2 2	2 0 0 4 2	0 0 0 3 1	0 0 0 1 1	1 0 0 2 1	0 0 0 3 2	2 0 0 3 1	0 0 0 3 2	1 0 0 2 2	0 0 0 4 3	0 0 0
12	180	GA	0.994	117	180	0 3 3 0 3	0 0 1 0 4	4 0 3 2 0	0 0 0 2 2	1 0 0 0 2	0 2 2 0 0	0 3 0 1 1	0 0 3 0 0	3 0 1 0 3	0 2 0 3 0	1 0 3 2 0	2 0 1 0 1	3 0 3 0 1	3 0 0 3 0 0
		MA	0.994667	122	178	3 3 3 0 0	1 1 0 0 4	2 0 0 3 3	3 0 0 2 2	1 0 0 3 2	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 0	0 0 0 3 2	1 0 0 3 1	0 0 0 4 1	1 0 0 2 0	0 4 1 0 0
13	179	GA	0.993333	118	179	0 3 0 3 0	0 1 1 0 4	4 0 0 3 2	0 3 0 2 1	0 0 2 0 2	2 0 0 0 1	1 0 0 0 1	1 1 0 0 1	1 1 0 1 3	0 0 0 2 2	0 0 0 2 3	1 0 1 0 2	1 0 0 2 0	0 2 0 4
		MA	0.992	118	177	3 3 1 0 0	1 1 0 0 4	4 0 0 3 3	1 0 0 2 1	1 0 0 4 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 0	0 0 0 3 2	2 0 0 3 1	0 0 0 4 2	2 0 0 0 4	2 0 0
14	178	GA	0.987333	118	177	3 3 0 0 1	1 0 0 2 0	2 0 4 0 3	3 0 0 0 1	1 0 2 1 0	0 0 3 0 3	0 0 3 0 3	1 1 0 0 3	3 0 0 0 0	3 3 0 0 3	3 0 4 0 3	0 3 0 4 0	2 1 0 3 0	0 0
		MA	0.994	117	177	3 3 2 0 0	1 1 0 0 4	2 0 0 3 3	1 0 0 2 2	2 0 0 4 2	0 0 0 3 1	0 0 0 1 1	1 0 0 3 3	0 0 0 3 2	0 0 0 3 1	0 0 0 3 2	1 0 0 2 1	0 0 0 4 3	0 0 0
15	177	GA	0.99	110	177	3 0 0 3 1	0 1 0 2 0	0 2 0 1 0	3 1 1 2 0	2 0 2 0 2	0 0 3 0 2	0 0 3 0 2	0 1 3 0 0	1 0 3 0 0	1 0 3 0 2	3 3 1 0 0	0 0 2 0 1	1 1 0 1 0	0 0 3 0
		MA	0.995333	117	177	3 3 1 0 0	1 1 0 0 4	2 0 0 3 3	1 0 0 2 2	2 0 0 4 2	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 1	0 0 0 3 2	2 0 0 3 1	0 0 0 3 1	0 0 0 2 0	0 4 3 0 0

Tablo 5.10 (a). GA ve MA sistem tasarımları ve sonuçları

16	176	GA	0,992	107	176	2 2 0 0 1 1 1 0 0 0 1 2 0 4 4 0 3 3 0 1 1 0 0 2 2 1 1 0 0 0 0 4 2 1 1 3 0 0 0 1 1 0 1 1 0 1 0 0 3 0 1 0 3 0 1 0 3 0 0 2 2 1 3 0 0 0 1 1 1 0 0 1 2 3 0 0 0 1 0 3 0
		MA	0,995333	123	176	3 3 3 0 0 1 1 0 0 0 4 4 0 4 0 3 3 3 0 0 3 2 0 0 2 2 0 0 3 1 0 0 0 3 1 0 0 3 2 0 0 0 2 2 2 0 0 2 2 2 0 0 3 2 0 0 0 2 2 2 0 0 2 2 2 0 0 2 2 2 0 0 2 2 0 0 0 4 3 0 0
17	175	GA	0,992667	115	174	3 0 0 3 3 0 1 0 1 0 0 4 0 4 0 4 0 1 0 1 1 0 0 3 0 3 1 0 2 0 0 0 1 0 1 0 1 3 0 3 0 0 3 3 0 3 0 0 2 2 0 0 0 1 3 2 0 1 0 2 1 0 2 0 0 0 4 3 0
		MA	0,993333	114	175	3 3 3 0 0 1 1 0 0 0 4 4 2 0 0 3 2 1 0 0 2 2 1 0 0 4 1 0 0 0 3 1 0 0 0 3 1 1 0 0 3 3 0 0 3 2 2 0 0 1 1 0 0 0 2 1 1 0 0 2 2 0 0 0 3 2 0 0 0
18	174	GA	0,988667	109	174	0 2 3 3 0 1 0 0 1 0 4 0 4 0 0 3 0 1 3 0 3 1 0 0 3 0 3 0 3 0 0 1 0 1 1 0 1 0 2 3 0 0 0 0 3 2 1 1 3 0 0 0 2 0 3 1 0 2 0 2 0 0 0 3 2
		MA	0,994	118	174	3 3 3 0 0 1 1 0 0 0 4 4 0 4 0 3 3 1 0 0 3 2 2 0 0 2 2 0 0 3 1 0 0 0 1 1 0 0 3 3 0 0 0 3 2 2 0 0 3 2 2 0 0 3 2 0 0 2 1 1 0 0 2 1 0 0 4 1 0 0 0
19	173	GA	0,991333	112	171	0 0 3 2 3 1 1 0 0 0 4 0 4 0 4 0 3 3 0 3 0 0 2 2 2 2 0 0 1 0 0 0 1 0 0 2 0 1 0 0 2 3 0 0 3 0 2 2 2 1 0 0 0 2 0 1 1 1 0 0 0 3 0 0 4
		MA	0,991333	115	172	3 3 1 0 0 1 1 0 0 0 4 4 0 0 3 3 1 0 0 2 2 0 0 4 1 0 0 0 3 1 0 0 3 3 1 0 0 3 3 0 0 2 2 1 0 0 3 1 0 0 4 1 1 0 0 2 1 0 0 4 1 0 0 0
20	172	GA	0,987333	111	172	3 1 0 3 0 0 1 1 0 4 3 2 0 0 0 3 3 1 0 0 2 2 0 0 2 0 0 2 0 0 3 1 0 0 0 3 1 0 0 3 3 0 0 2 2 0 1 2 0 2 1 0 0 1 4 0 0 1 0 1 0 3 1 0
		MA	0,99	111	171	3 3 3 0 0 2 1 0 0 0 4 4 0 0 3 2 1 0 0 3 2 0 0 0 2 0 0 2 0 0 3 1 0 0 3 1 0 0 3 2 0 0 3 3 0 0 2 2 0 1 0 0 2 1 1 0 0 2 1 0 0 3 3 0 0
21	171	GA	0,986	111	171	3 2 0 0 3 0 1 1 0 0 0 4 0 4 0 2 0 3 3 1 0 2 0 1 0 4 1 0 0 1 2 0 0 3 0 0 3 0 3 0 0 3 1 0 2 3 0 0 3 0 1 0 2 2 0 1 2 0 0 0 0 4 3 0
		MA	0,989333	108	171	3 1 1 0 0 1 1 0 0 0 4 2 0 0 3 2 1 0 0 2 2 0 2 0 4 2 0 0 0 3 1 0 0 1 1 0 0 3 3 0 0 2 2 2 0 0 2 2 0 0 4 1 1 0 0 2 1 0 0 3 3 0 0
22	170	GA	0,986667	104	170	3 3 2 0 0 1 1 0 2 4 3 0 0 1 3 0 0 3 0 2 1 0 0 4 0 4 0 0 3 0 3 1 1 0 0 0 0 0 2 0 3 1 1 0 0 0 0 2 0 3 1 0 0 1 0 2 1 0 2 0 2 0 0 3 3
		MA	0,990667	112	170	3 3 1 0 0 1 1 0 0 0 4 4 2 0 0 3 3 1 0 0 2 2 0 0 2 1 0 0 2 1 0 0 0 2 1 0 0 1 1 0 0 3 3 0 0 3 2 2 0 0 3 2 0 0 1 1 1 0 0 2 2 0 0 4 1 0 0 0
23	169	GA	0,986	112	168	0 0 3 3 3 0 0 3 0 1 1 4 0 0 4 3 3 3 0 0 1 0 0 1 0 0 2 3 0 0 0 2 0 1 0 1 1 0 1 0 0 3 0 3 2 2 0 0 2 0 2 1 0 3 2 4 0 0 2 3 0 0 3 0
		MA	0,988667	109	169	3 3 3 0 0 2 1 0 0 0 4 4 0 0 3 3 2 0 0 2 1 0 0 4 2 0 0 0 3 1 0 0 1 1 0 0 2 3 0 0 3 2 2 0 0 3 1 0 0 2 2 2 0 0 1 1 0 0 3 3 0 0
24	168	GA	0,981333	109	168	0 0 0 4 3 0 0 3 0 1 1 4 0 0 4 0 1 3 3 0 0 1 2 0 0 2 0 1 0 2 3 0 0 1 0 0 3 1 0 0 1 4 0 0 3 0 0 3 0 3 0 3 0 3 1 0 0 1 2 0 0 0 2 3 0 0
		MA	0,987333	113	162	3 3 3 0 0 1 1 0 0 0 4 2 0 0 3 3 3 0 0 3 2 0 0 2 0 0 2 0 0 1 1 0 0 3 1 0 0 3 3 0 0 2 2 2 0 0 3 1 0 0 2 1 1 0 0 2 2 0 0 3 1 0 0
25	167	GA	0,985333	110	167	3 0 3 3 0 0 0 0 1 1 0 4 2 0 0 3 0 3 3 1 2 0 0 1 3 0 2 0 0 1 2 0 0 1 0 0 1 1 0 3 0 1 0 2 3 0 0 3 3 0 0 1 0 0 3 1 0 2 0 0 0 0 0 4
		MA	0,989333	112	167	3 3 3 0 0 1 1 0 0 0 4 4 0 0 3 3 1 0 0 2 1 0 0 2 0 0 2 0 0 3 1 0 0 0 3 1 1 0 0 3 1 0 0 3 2 1 0 0 3 1 0 0 2 1 2 0 0 2 1 0 0 3 3 0 0
26	166	GA	0,985333	106	165	3 2 3 0 0 0 1 1 0 0 0 2 4 1 3 3 0 0 2 2 0 0 0 4 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 3 0 0 2 2 0 3 0 1 0 0 3 0 2 2 0 2 0 2 0 1 0 3 0 0
		MA	0,989333	116	166	3 3 3 0 0 1 1 0 0 0 4 4 0 0 3 3 3 0 0 2 2 0 0 2 2 0 0 3 1 0 0 0 3 1 1 0 0 3 3 0 0 2 2 0 0 3 1 0 0 2 2 0 0 3 1 0 0 4 3 0 0
27	165	GA	0,984667	111	164	3 0 3 0 3 0 0 0 1 2 0 0 4 0 4 0 3 3 3 0 3 2 0 3 4 0 2 0 0 2 0 1 0 0 3 0 3 3 0 3 0 3 2 0 0 2 0 2 1 0 0 1 0 2 0 2 0 1 0 1 0 3 1 0 0
		MA	0,986667	108	165	3 3 3 0 0 1 1 0 0 0 4 4 0 0 3 3 1 0 0 2 2 0 0 4 2 0 0 3 1 0 0 0 3 1 1 0 0 2 2 0 0 3 3 0 0 1 1 0 0 2 1 0 0 2 2 0 0 3 3 0 0
28	164	GA	0,986667	112	164	0 3 0 3 3 0 3 1 0 0 0 4 0 4 0 3 1 3 0 0 0 3 0 2 0 1 0 2 0 0 1 0 0 1 1 1 0 3 0 3 0 0 2 0 0 2 3 1 0 0 0 1 1 0 0 1 0 0 2 0 2 3 0 0
		MA	0,987333	110	163	3 3 3 0 0 1 1 0 0 0 4 4 0 0 3 1 1 0 0 2 2 0 0 2 0 0 3 1 0 0 0 1 1 1 0 3 0 3 0 0 3 3 2 0 0 3 1 0 0 2 1 1 0 0 2 0 2 3 0 0
29	163	GA	0,979333	99	162	0 3 3 1 0 0 0 0 1 3 0 4 0 0 0 3 0 2 1 3 0 1 0 0 0 3 2 0 0 0 1 3 0 1 0 1 1 0 2 0 0 0 0 0 1 2 0 0 0 0 1 2 0 0 2 1 1 0 0 0 0 2 0 4 3 0 0
		MA	0,983333	108	163	3 2 1 0 0 1 1 0 0 0 4 2 0 0 3 3 3 0 0 2 1 0 0 4 2 0 0 3 1 0 0 3 1 0 0 2 0 0 3 1 0 0 3 1 0 0 2 0 0 3 2 2 0 0 3 2 0 0 2 1 1 0 0 2 0 0 3 0 0
30	162	GA	0,98	100	162	3 0 1 1 0 0 1 1 0 4 4 0 0 1 0 0 3 3 2 0 0 0 0 4 2 0 1 2 0 0 1 0 1 1 0 1 0 1 0 3 2 0 0 0 2 3 0 0 0 1 4 0 1 2 0 0 1 0 0 1 0 3
		MA	0,979333	102	161	4 3 0 0 3 1 0 0 0 4 0 0 3 3 2 0 0 2 1 0 0 2 1 0 0 3 1 0 0 0 1 1 0 0 3 3 0 0 3 2 0 0 2 1 0 0 2 1 1 0 0 2 1 1 0 0 2 0 0 3 0 0
31	161	GA	0,980667	106	160	3 0 1 0 3 0 1 0 2 0 0 4 4 3 3 0 3 0 0 1 2 0 4 0 0 3 1 0 0 1 0 3 1 0 0 2 0 2 0 3 0 0 1 0 0 1 0 0 1 2 0 0 2 2 3 0 0
		MA	0,987333	105	160	3 3 1 0 0 1 1 0 0 0 4 4 0 0 3 3 3 0 0 2 0 0 4 2 0 0 2 1 0 0 0 1 1 0 0 3 1 0 0 2 2 2 0 0 3 1 0 0 2 1 1 0 0 1 1 0 0 3 3 0 0
32	160	GA	0,972	100	159	3 1 1 0 0 3 0 1 0 0 4 4 0 0 0 1 2 0 3 0 0 2 0 1 0 2 0 0 2 0 3 1 0 0 0 1 0 3 0 0 0 2 1 0 0 2 1 2 1 0 0 0 1 1 0 0 2 0 0 0 3 0 0
		MA	0,977333	103	159	3 3 1 0 0 1 1 0 0 0 4 4 0 0 3 1 1 0 0 1 1 0 0 2 2 0 0 3 1 0 0 0 3 1 1 0 0 2 0 0 3 3 0 0 3 1 0 0 2 2 1 0 0 2 2 1 0 0 3 3 0 0
33	159	GA	0,976	102	159	0 3 0 3 4 1 1 0 0 0 4 2 0 0 1 3 0 0 1 0 0 2 2 0 4 1 0 0 0 1 1 0 0 1 0 3 1 0 0 0 2 0 3 0 0 3 0 0 1 3 2 1 0 2 0 2 0 0 2 0 3 0
		MA	0,976667	101	159	3 1 1 0 0 1 1 0 0 0 4 2 0 0 3 3 0 0 0 2 2 0 0 4 0 0 0 2 2 0 0 3 1 1 0 0 3 2 0 0 3 3 0 0 0 3 1 0 0 2 1 1 0 0 2 2 0 0 3 3 0 0

Tablo 5.10 (b). GA ve MA sistem tasarımları ve sonuçları

5.5.2. Değişken “k” değerine sahip sistemler için analiz sonuçları

İletişim, elektronik ve bulut gibi sistemlerde, dönemsel olarak yaşanan talep değişikliklerine karşı doğru öngöründe bulunmak önemlidir. Örneğin, anneler günü, sevgililer günü, kara cuma gibi özel günlerde online alışveriş sitelerinin kullanımında artış yaşanabilmektedir. Dünyayı etkileyen COVID-19 salgını ile başta eğitim olmak üzere online olarak uzaktan yürütülen süreçlerin artmasıyla birlikte bu tür sistemlerde yaşanan talep değişiklikleri eğer sistem tasarımları sırasında doğru öngörülerle dikkate alınmamış ise tüm kullanıcıları olumsuz yönde etkilemektedir.

Tezin bu bölümünde YBTP için önerilen MA’nın algoritmanın değişken “k” değerleri ile çalışması dikkate alınarak seçilen bir test problemi (P22) üzerinde analizler yapılmış ve sonuçlar sunulmuştur. Analizler için P22 probleminde değişiklik yapılarak, bir yıl boyunca (t=52 hafta) sosyal yaşamın içinde farklı özellikleri ve önemi olabilecek dönemler için servis sağlayıcıların talep ettiği “k” değerlerindeki değişimin dikkate alındığı farklı “k” değerlerine sahip 3 farklı örnek problem oluşturulmuştur. Oluşturulan bu P_{kd1} , P_{kd2} ve P_{kd3} problemlerinde maliyet sınırı 300 birim, ağırlık sınırı 400 birim olarak alınmıştır. Tablo 5.11’de ilk örnek probleme ait değişken “k” değerleri verilmiştir. Oluşturulan diğer örnek problemlere ait tablolar Ek 3’te verilmiştir. Bu yaklaşımla, servis sağlayıcının yılın belirli dönemlerinde ortaya çıkacak talep artışını önceden dikkate alarak her bir alt sistemde dönemsel olarak “k” değerinin ne olması gerektiğini öngörmesi ve uygun sistem tasarımını buna göre elde etmesi sağlanmaktadır.

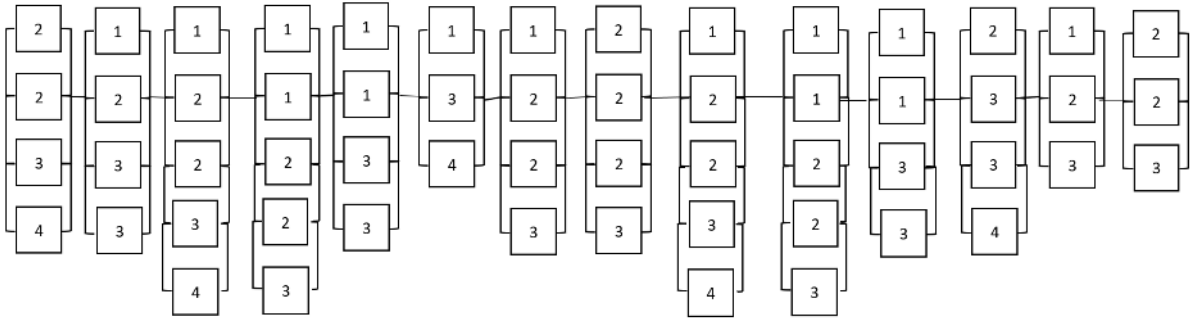
Bu bölümde YBTP için geliştirilen MA ile değişken ve sabit “k” değerleri için elde edilen sistem tasarımları ve bilgileri Tablo 5.12’de verilmiştir.

Tablo 5.11. P_{kd1} problemine ait değişken “k” değerleri

t		Alt Sistemlere ait Değişken "k" Değerleri													
Ay	Hafta	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7
2	5	6	4	7	4	7	4	6	4	7	4	7	4	6	4
2	6	6	4	7	4	7	4	6	4	7	4	7	4	6	4
2	7	6	4	7	4	7	4	6	4	7	4	7	4	6	4
2	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7
2	9	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	10	7	7	7	7	7	7	7	7	7	7	7	7	7	7

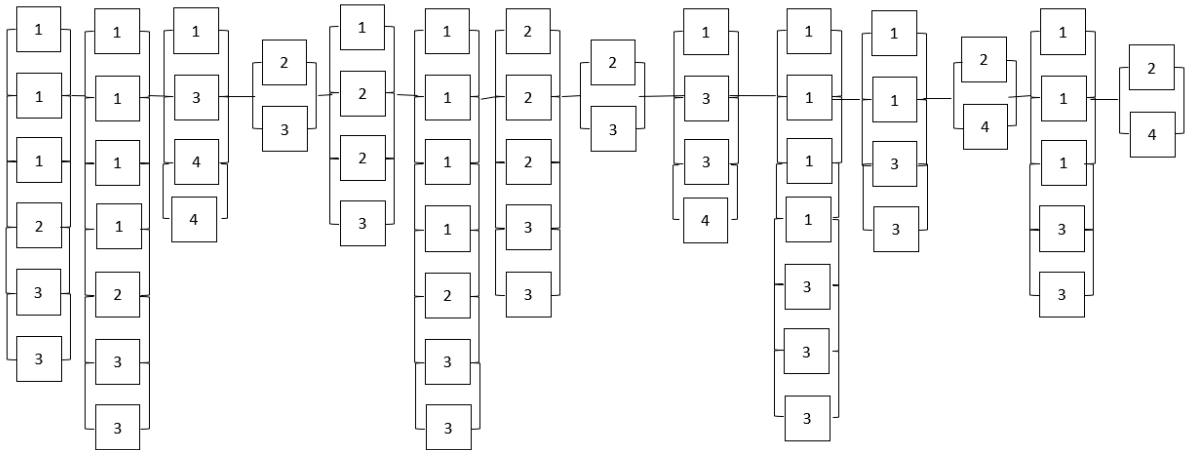
3	11	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	12	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	13	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	14	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	15	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	16	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	17	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	18	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	19	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	20	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	21	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	22	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	23	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	24	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	25	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	26	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	27	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	28	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	29	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	30	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	31	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	32	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	33	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	34	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	35	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	36	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	37	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	38	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	39	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	40	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	41	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	42	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	43	7	7	7	7	7	7	7	7	7	7	7	7	7	7
11	44	5	3	6	3	6	3	5	3	6	3	6	3	5	3
11	45	5	3	6	3	6	3	5	3	6	3	6	3	5	3
11	46	5	3	6	3	6	3	5	3	6	3	6	3	5	3
11	47	5	3	6	3	6	3	5	3	6	3	6	3	5	3
12	48	5	3	6	3	6	3	5	3	6	3	6	3	5	3
12	49	5	3	6	3	6	3	5	3	6	3	6	3	5	3
12	50	5	3	6	3	6	3	5	3	6	3	6	3	5	3
12	51	5	3	6	3	6	3	5	3	6	3	6	3	5	3
12	52	5	3	6	3	6	3	5	3	6	3	6	3	5	3

Sabit “ k ” değeri için elde edilen örnek sistem tasarımı (P_{ks}) Şekil 5.5’te verilmiştir. Bu sistemin güvenilirliği 0.99; maliyeti 186 birim ve ağırlığı 346 birimdir.

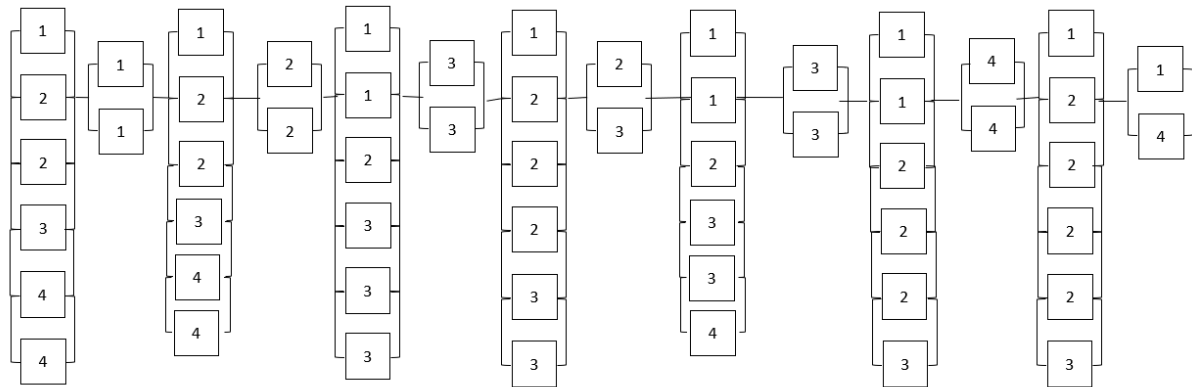


Şekil 5.5. P_{ks} problemi için (sabit k değeri) elde edilen sistem tasarımı

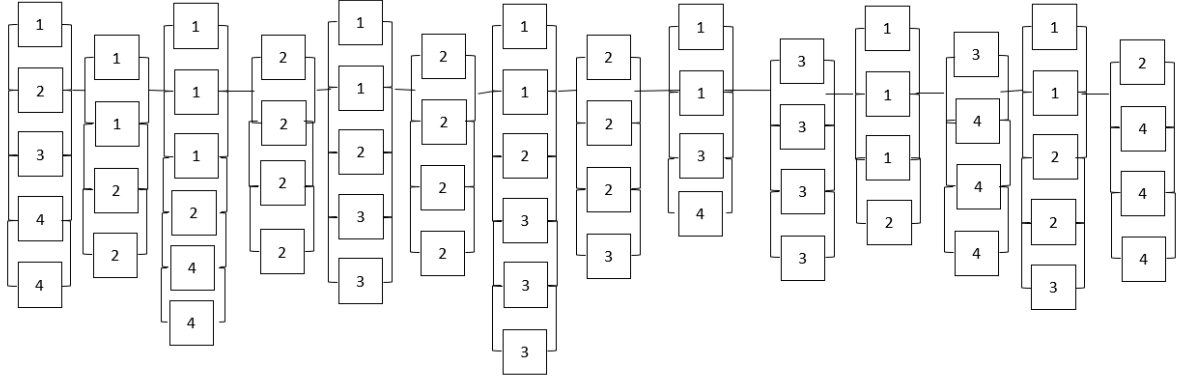
Değişken “ k ” değerleri için elde edilen örnek sistem tasarımları (P_{kd1} , P_{kd2} ve P_{kd3}) Şekil 5.6(a-c)’de sunulmuştur.



Şekil 5.6(a). P_{kd1} problemi için elde edilen sistem tasarımı



Şekil 5.6(b). P_{kd2} problemi için elde edilen sistem tasarımı



Şekil 5.6(c). P_{kd3} problemi için elde edilen sistem tasarımı

P_{ks} , P_{kd1} , P_{kd2} , P_{kd3} problemleri için elde edilen sistemin güvenilirlikleri,; maliyetleri ve ağırlıkları Tablo 5.12’de verilmiştir.

Tablo 5.12. Değişken ve sabit “ k ” değerleri için elde edilen sistem tasarımları

Problem	Güvenilirlik	Maliyet	Ağırlık	Altsistem 1								Altsistem 2								Altsistem 3								Altsistem 4							
P _{ks}	0,99	186	346	4	3	2	2	0	0	0	3	3	2	1	0	0	0	4	3	2	2	1	0	0	3	2	2	1	1	0	0				
P _{kd1}	0,984	188	361	3	3	2	1	1	1	0	3	3	2	1	1	1	1	4	4	3	1	0	0	0	3	2	0	0	0	0	0				
P _{kd2}	0,992	181	333	4	4	3	2	2	1	0	1	1	0	0	0	0	0	4	4	3	2	2	1	0	2	2	0	0	0	0	0				
P _{kd3}	0,909	216	389	4	4	3	0	1	2	0	2	2	1	1	0	0	0	4	4	2	1	1	1	0	2	2	2	2	0	0	0				

Problem	Altsistem 5							Altsistem 6							Altsistem 7							Altsistem 8							Altsistem 9						
P_{ks}	3	3	1	1	0	0	0	4	3	1	0	0	0	0	3	2	2	1	0	0	0	3	2	2	2	0	0	0	4	3	2	2	1	0	0
P_{kd1}	3	2	2	1	0	0	0	3	3	2	1	1	1	1	3	3	2	2	2	0	0	3	2	0	0	0	0	0	4	3	3	1	0	0	0
P_{kd2}	3	3	3	2	1	1	0	3	3	0	0	0	0	0	3	3	2	2	2	1	0	3	2	0	0	0	0	0	4	3	3	2	1	1	0
P_{kd3}	3	3	2	1	1	0	0	2	2	2	2	0	0	0	3	3	3	2	1	1	0	3	2	2	2	0	0	0	4	3	1	1	0	0	0

Problem	Altsistem 10							Altsistem 11							Altsistem 12							Altsistem 13							Altsistem 14						
P _{ks}	3	2	2	1	1	0	0	3	3	1	1	0	0	0	4	3	3	2	0	0	0	3	2	1	0	0	0	0	3	2	2	0	0	0	0
P _{kd1}	3	3	3	1	1	1	1	3	3	1	1	0	0	0	4	2	0	0	0	0	0	3	3	1	1	1	0	0	4	2	0	0	0	0	0
P _{kd2}	3	3	0	0	0	0	0	3	2	2	2	1	1	0	4	4	0	0	0	0	0	3	2	2	2	2	1	0	4	1	0	0	0	0	0
P _{kd3}	3	3	3	3	0	0	0	2	1	1	1	0	0	0	4	4	4	3	0	0	0	3	2	2	1	1	0	0	4	4	4	2	0	0	0

6. SONUÇ ve ÖNERİLER

Günümüzde başta iletişim, elektronik, uzay sistemleri olmak üzere çoğu sektörde teknolojiye yaşanan hızlı gelişmeler büyük değişimlerin yaşanmasına neden olmuştur. Modern mühendislik ve hizmet sistemlerinin karmaşık yapısı ve yüksek teknoloji endüstriyel süreçler sistemlerin daha etkin bir şekilde hizmet sunmasının önemini artırmıştır. Bu tür karmaşık sistemlerin analizi ve tasarımlarıyla ilgili çalışmalarda başarımlarının değerlendirilmesi için kullanılan ölçütlerden en kritik ve önemli olanı sistem güvenilirliğidir. Buna bağlı olarak, çeşitli sistem kısıtları altında sistem güvenilirliğinin eniyilenmesi güncel bir problem olarak önemini sürdürmektedir.

Sistem güvenilirliği eniyilenmesi, farklı disiplinlerde çalışan birçok araştırmacının yoğun ilgisini çekmektedir. Özellikle iletişim, elektronik, uydu ve farklı uzay sistemleri gibi teknoloji yoğun alanlarda yüksek güvenilirliğe duyulan ihtiyaç, araştırmaların sistemlerin gerçek özelliklerini yansıtmayan varsayımlar kullanmak yerine gerçekçi varsayımların kullanılmasını çok önemli hale getirmiştir.

YBTP’de enbüyük güvenilirlik değerine sahip olacak şekilde her bir alt sistemde yer alan bileşenlere paralel olarak yerleştirilecek bileşenlerin yeri, türü ve sayısının bulunması ile ilgilenilmektedir.

Gerçek hayatta kullanılan sistemlerde, arızalanan bileşenin tamir edilerek çalışır duruma dönmesi sistemin güvenilirliğini etkileyerek artıracaktır. Böylece yüksek güvenilirliğe sahip tasarımlar daha düşük maliyetle elde edilebilecektir. Mevcut çalışmalarda sistem bileşenlerinin arıza ve/veya tamir oranlarının sabit olduğu varsayımının kullanılmasının önemli bir nedeni sistem güvenilirliğinin analitik yöntemlerle kolay hesaplanabiliyor olmasıdır. Ancak, gerçek hayat problemlerinde sistemi oluşturan bileşenler kullanımdan kaynaklanan yıpranmaya bağlı olarak artan arızalanma oranına sahiptir.

Bu tez çalışmasında YBTP eniyilenmesinde dikkate alınan amaç fonksiyonu değeri, sistem güvenilirliği, gerçekçi bir yaklaşımla geliştirilen, olayların çıkışına dayalı ve benzetim zamanının en yakın olay zamanına göre ilerletildiği dinamik bir KOB modeli kurularak tahmin edilmiştir. Sistem tasarımı blok diyagramı ile gösterilerek n bileşenli m adet alt sistem için n ’den k ’lı sisteme (k -out-of- n) göre genelleştirilmiş bir KOB modeli önerilerek doğrulanmış ve geçerliliği alan yazındaki test problemleri üzerinde gösterilmiştir. Ayrıca geliştirilen GA ve MA algoritmalarının etkinliği ise alan yazında yaygın olarak incelenen test problemleri üzerinde detaylı olarak incelenmiştir. Her iki algoritmanın sonuçları

karşılaştırıldığında, tüm test problemleri için MA'nın GA'dan daha yüksek "ortalama sistem güvenilirliği" değerleri bulduğu görülmüştür. MA, problemlerin %18.18'inde GA ile eşdeğer; %78.78'inde ise daha yüksek "en büyük sistem güvenilirliği" değerleri bulmuştur. MA'nın en iyi çözümü bulduğu ortalama iterasyon sayısının GA'nın en iyi çözümü bulduğu ortalama iterasyon sayısından büyük olduğu görülmektedir. Bu MA'nın yapısı gereği beklenen bir durumdur çünkü MA, GA'nın yığın güncelleme adımıyla yerel komşu araması yaparak daha iyi bir çözüm aramaktadır. Güvenilirlik değerindeki çok küçük bir artışın önemi bilindiğinden MA çalışma süresinin katlanılabilir bir süre olduğu söylenebilir. Sonuç olarak, YBTP için probleme özgü geliştirilen GA ve MA başarılı sonuçlar bulurken, MA'nın daha iyi sonuç buluyor olması nedeniyle tezde önerilen algoritma olarak öne çıkmaktadır.

Bunlara ek olarak, YBTP'de, güvenilirlik tahmininde kullanmak üzere geliştirilen ve önerilen KOB modelinde güvenilirlik kısıtı olarak kullanılan, n 'den k 'lı (k -out-of- n) varsayımındaki " k " parametresi farklı dönemlerde değişkenlik gösterebileceği için KOB modeli geliştirilerek değişken " k " ya göre kurulmuş ve değişken " k " gereksinimlerinin karşılanabildiği sistem tasarımları test problemleri üzerinde gösterilmiştir.

Bu tezde sistem güvenilirliği eniyilenmesi kapsamında yapılan çalışmalar aşağıdaki gibi özetlenebilir:

- NP-zor bir problem olan YBTP için GA geliştirilmiştir.
- GA için en iyi parametre seti iki aşamalı olarak; i) ön denemeler ve ii) istatistiksel deney tasarımıyla belirlenmiştir.
- GA, yerel arama yaklaşımı ile karma yapıda kullanılarak MA geliştirilmiştir.
- Her iki algorithmada amaç ve ceza fonksiyonundan oluşan değerlendirme fonksiyonu kullanılmıştır.
- GA ve MA'da amaç fonksiyonu olarak sistem güvenilirliğini tahmin edecek KOB modeli kurulmuş ve kesikli olaylı Monte Carlo benzetimi ile sistem güvenilirliği tahmin edilmiştir.
- KOB modelinin geçerliliği, GA ve MA'nın etkinliği test problemleri üzerinde gösterilmiştir.
- Sayısal analizler sonucunda, GA ve MA'nın kaliteli çözümler bulduğu; MA'nın GA'ya göre daha iyi sonuçlar verdiği ortaya konmuştur.
- KOB modeli değişken " k " için geliştirilmiştir.
- " k " açısından dinamik bir KOB modelinin kullanıldığı MA ile dönemsel değişen " k "ya göre elde edilen sistem tasarımları elde edilmiştir.

Gelecek çalışmalar olarak;

- Tamir edilebilir bileşenli YBTP için çok amaçlı çözüm yaklaşımları dikkate alınabilir. Pareto optimal çözümler üzerine çalışılabilir.
- Metasezgisel algoritmaların başlangıç çözümleri üzerinde farklı önerilerle algoritmaların başarımı iyileştirilebilir.

KAYNAKLAR

- [1] W. Kuo and V. Rajendra Prasad, "An annotated overview of system-reliability optimization," *IEEE Trans. Reliab.*, vol. 49, no. 2, pp. 176–187, 2000, doi: 10.1109/24.877336.
- [2] R. Bellman and S. Dreyfus, "Dynamic Programming and the Reliability of Multicomponent Devices," *Oper. Res.*, vol. 6, no. 2, pp. 200–206, 1958, doi: 10.1287/opre.6.2.200.
- [3] D. W. Coit and A. E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm," *IEEE Trans. Reliab.*, vol. 45, no. 2, pp. 254–260, 263, 1996, doi: 10.1109/24.510811.
- [4] Y. Nakagawa and S. Miyazaki, "Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints," *IEEE Trans. Reliab.*, vol. 30, no. 2, pp. 175–180, 1981.
- [5] K. Y. Ng and N. G. F. Sancho, "A hybrid 'dynamic programming/depth-first search' algorithm, with an application to redundancy allocation," *IIE Trans. (Institute Ind. Eng.)*, vol. 33, no. 12, pp. 1047–1058, 2001, doi: 10.1080/07408170108936895.
- [6] D. W. Coit and A. E. Smith, "PENALTY GUIDED GENETIC SEARCH FOR RELIABILITY DESIGN OPTIMIZATION," vol. 30, no. 4, pp. 895–904, 1996.
- [7] S. Kulturel-Konak, A. E. Smith, and D. W. Coit, "Efficiently solving the redundancy allocation problem using tabu search," *IIE Trans. (Institute Ind. Eng.)*, vol. 35, no. 6, pp. 515–526, 2003, doi: 10.1080/07408170304422.
- [8] D. Salazar, C. M. Rocco, and B. J. Galván, "Optimization of constrained multiple-objective reliability problems using evolutionary algorithms," *Reliab. Eng. Syst. Saf.*, vol. 91, no. 9, pp. 1057–1070, 2006, doi: 10.1016/j.res.2005.11.040.
- [9] A. Chambari, A. A. Najafi, S. H. A. Rahmati, and A. Karimi, "An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies," *Reliab. Eng. Syst. Saf.*, vol. 119, pp. 158–164, 2013, doi: 10.1016/j.res.2013.05.016.
- [10] A. Peiravi, M. A. Ardakan, and E. Zio, "A new Markov-based model for reliability optimization problems with mixed redundancy strategy," *Reliab. Eng. Syst. Saf.*, vol. 201, no. April 2019, p. 106987, 2020, doi: 10.1016/j.res.2020.106987.
- [11] M. S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Oper. Res. Lett.*, vol. 11, no. 5, pp. 309–315, 1992, doi: 10.1016/0167-6377(92)90008-Q.
- [12] D. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Choice Reviews Online*, vol. 27, no. 02, pp. 27-0936-27-0936, 1989, doi: 10.5860/choice.27-0936.
- [13] F. Altıparmak and B. Dengiz, "A cross entropy approach to design of reliable networks," *Eur. J. Oper. Res.*, vol. 199, no. 2, pp. 542–552, 2009, doi: 10.1016/j.ejor.2008.11.022.
- [14] D. Arivudainambi and D. Rekha, "Heuristic approach for broadcast scheduling, problem in wireless mesh networks," *AEU - Int. J. Electron. Commun.*, vol. 68, no. 6, pp. 489–495, 2014, doi: 10.1016/j.aeue.2013.12.009.
- [15] M. Gen, R. Cheng, and S. S. Oren, "Network design techniques using adapted genetic algorithms," *Adv. Eng. Softw.*, vol. 32, no. 9, pp. 731–744, 2001, doi: 10.1016/S0965-9978(01)00007-2.
- [16] A. E. Jahromi and Z. B. Rad, "Optimal topological design of power communication

- networks using genetic algorithm,” *Sci. Iran.*, vol. 20, no. 3, pp. 945–957, 2013, doi: 10.1016/j.scient.2013.01.003.
- [17] A. Quintero and S. Pierre, “Evolutionary approach to optimize the assignment of cells to switches in personal communication networks,” *Comput. Commun.*, vol. 26, no. 9, pp. 927–938, 2003, doi: 10.1016/S0140-3664(02)00238-4.
 - [18] A. Robert singh, D. Devaraj, and R. Narmatha Banu, “Genetic algorithm-based optimisation of load-balanced routing for AMI with wireless mesh networks,” *Appl. Soft Comput. J.*, vol. 74, pp. 122–132, 2019, doi: 10.1016/j.asoc.2018.10.003.
 - [19] S. Salcedo-Sanz *et al.*, “Optimal switch location in mobile communication networks using hybrid genetic algorithms,” *Appl. Soft Comput. J.*, vol. 8, no. 4, pp. 1486–1497, 2008, doi: 10.1016/j.asoc.2007.10.021.
 - [20] K. Wu, J. Liu, and D. Chen, “Network reconstruction based on time series via memetic algorithm,” *Knowledge-Based Syst.*, vol. 164, pp. 404–425, 2019, doi: 10.1016/j.knosys.2018.11.009.
 - [21] S. Liu, K. Tang, and X. Yao, “Memetic Search for Vehicle Routing with Simultaneous Pickup-Delivery and Time Windows,” *Swarm Evol. Comput.*, p. 100927, 2020, doi: 10.1016/j.swevo.2021.100927.
 - [22] P. Sitek, J. Wikarek, K. Rutczyńska-Wdowiak, G. Bocewicz, and Z. Banaszak, “Optimization of capacitated vehicle routing problem with alternative delivery, pick-up and time windows: A modified hybrid approach,” *Neurocomputing*, vol. 423, pp. 670–678, 2021, doi: 10.1016/j.neucom.2020.02.126.
 - [23] M. D. A. Serna, C. A. S. Uran, J. A. Z. Cortes, and A. F. A. Benitez, “Vehicle Routing to Multiple Warehouses Using a Memetic Algorithm,” *Procedia - Soc. Behav. Sci.*, vol. 160, no. Cit, pp. 587–596, 2014, doi: 10.1016/j.sbspro.2014.12.172.
 - [24] G. Srivastava, A. Singh, and R. Mallipeddi, “NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows,” *Expert Syst. Appl.*, vol. 176, no. February, p. 114779, 2021, doi: 10.1016/j.eswa.2021.114779.
 - [25] N. R. Sabar, A. Bhaskar, E. Chung, A. Turky, and A. Song, “An Adaptive Memetic Approach for Heterogeneous Vehicle Routing Problems with two-dimensional loading constraints,” *Swarm Evol. Comput.*, vol. 58, no. November 2019, p. 100730, 2020, doi: 10.1016/j.swevo.2020.100730.
 - [26] A. T. Mulloorakam and N. Mathew Nidhiry, “Combined Objective Optimization for Vehicle Routing Using Genetic Algorithm,” *Mater. Today Proc.*, vol. 11, pp. 891–902, 2019, doi: 10.1016/j.matpr.2018.12.016.
 - [27] J. Long, Z. Sun, P. M. Pardalos, Y. Hong, S. Zhang, and C. Li, “A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem,” *Inf. Sci. (Ny)*, vol. 478, pp. 40–61, 2019, doi: 10.1016/j.ins.2018.11.006.
 - [28] C. Y. Li, J. Li, H. L. Chen, and A. A. Heidari, “Memetic Harris Hawks Optimization: Developments and perspectives on project scheduling and QoS-aware web service composition,” *Expert Syst. Appl.*, vol. 171, no. April 2020, p. 114529, 2021, doi: 10.1016/j.eswa.2020.114529.
 - [29] H. F. Rahman, R. K. Chakraborty, and M. J. Ryan, “Scheduling project with stochastic durations and time-varying resource requests: A metaheuristic approach,” *Comput. Ind. Eng.*, vol. 157, no. April, p. 107363, 2021, doi: 10.1016/j.cie.2021.107363.
 - [30] S. Asta, D. Karapetyan, A. Kheiri, E. Özcan, and A. J. Parkes, “Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem,” *Inf. Sci. (Ny)*, vol. 373, pp. 476–498, 2016, doi: 10.1016/j.ins.2016.09.010.
 - [31] H. F. Rahman, R. K. Chakraborty, and M. J. Ryan, “Memetic algorithm for solving

- resource constrained project scheduling problems,” *Autom. Constr.*, vol. 111, no. November 2019, p. 103052, 2020, doi: 10.1016/j.autcon.2019.103052.
- [32] R. L. Kadri and F. F. Boctor, “An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case,” *Eur. J. Oper. Res.*, vol. 265, no. 2, pp. 454–462, 2018, doi: 10.1016/j.ejor.2017.07.027.
- [33] X. N. Shen, L. L. Minku, N. Marturi, Y. N. Guo, and Y. Han, “A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling,” *Inf. Sci. (Ny)*, vol. 428, pp. 1–29, 2018, doi: 10.1016/j.ins.2017.10.041.
- [34] S. Chand, Q. Huynh, H. Singh, T. Ray, and M. Wagner, “On the use of genetic programming to evolve priority rules for resource constrained project scheduling problems,” *Inf. Sci. (Ny)*, vol. 432, pp. 146–163, 2018, doi: 10.1016/j.ins.2017.12.013.
- [35] A. Shirzadeh Chaleshtarti, S. Shadrokh, M. Khakifirooz, M. Fathi, and P. M. Pardalos, “A hybrid genetic and Lagrangian relaxation algorithm for resource-constrained project scheduling under nonrenewable resources,” *Appl. Soft Comput. J.*, vol. 94, p. 106482, 2020, doi: 10.1016/j.asoc.2020.106482.
- [36] Y.-C. Liang and A. E. Smith, “An ant colony optimization algorithm for selection problem,” *Appl. Mech. Mater.*, vol. 53, no. 3, pp. 417–423, 2004, doi: 10.4028/www.scientific.net/AMM.411-414.1939.
- [37] M. Keshavarz Ghorabae, M. Amiri, and P. Azimi, “Genetic algorithm for solving bi-objective redundancy allocation problem with k-out-of-n subsystems,” *Appl. Math. Model.*, vol. 39, no. 20, pp. 6396–6409, 2015, doi: 10.1016/j.apm.2015.01.070.
- [38] O. Dengiz, A. E. Smith, F. Altiparmak, and B. Dengiz, “A Pareto Fed Multi-objective Genetic Algorithm for the Redundancy Allocation Problem,” 2005.
- [39] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *EVOLUTIONARY MULTI-OBJECTIVE PROBLEMS Genetic Algorithms and Evolutionary Computation*. .
- [40] P. G. Busacca, M. Marseguerra, and E. Zio, “Multiobjective optimization by genetic algorithms: Application to safety systems,” *Reliab. Eng. Syst. Saf.*, vol. 72, no. 1, pp. 59–74, 2001, doi: 10.1016/S0951-8320(00)00109-5.
- [41] H. Zoufaghari, A. Zeinal Hamadani, and M. Abouei Ardakan, “Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components,” *ISA Trans.*, vol. 53, no. 1, pp. 17–24, 2014, doi: 10.1016/j.isatra.2013.08.002.
- [42] H. Hadipour, M. Amiri, and M. Sharifi, “Redundancy allocation in series-parallel systems under warm standby and active components in repairable subsystems,” *Reliab. Eng. Syst. Saf.*, vol. 192, no. January 2018, p. 106048, 2019, doi: 10.1016/j.ress.2018.01.007.
- [43] I. D. Lins and E. L. Drogue, “Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation,” *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 362–381, 2011, doi: 10.1016/j.simpat.2010.07.010.
- [44] E. A. Elsayed, *Reliability Engineering*. Addison Wesley Longman, 1996.
- [45] E. Zio, *An introduction to the basics of reliability and risk analysis*, vol. 1, no. 4. 2007.
- [46] Z. W. Birnbaum, J. D. Esary, and S. C. Saunders, “Multi-Component Systems and Structures and Their Reliability,” *Technometrics*, vol. 3, no. 1, pp. 55–77, 1961, doi: 10.1080/00401706.1961.10489927.
- [47] F. J. Samaniego, *System Signatures and their Applications in Engineering Reliability*, Vol. 110. Springer Science & Business Media, 2007.
- [48] F. P. A. Coolen and T. Coolen-Maturi, “Generalizing the signature to systems with multiple types of components,” *Adv. Intell. Soft Comput.*, vol. 170 AISC, pp. 115–130,

- 2012, doi: 10.1007/978-3-642-30662-4-8.
- [49] B. Dengiz, “Benzetim Ders Notları,” 2020. .
 - [50] D. E. Fyffe, W. W. Hines, and N. K. Lee, “System Reliability Allocation and a Computational Algorithm,” *IEEE Trans. Reliab.*, vol. R-17, no. 2, pp. 64–69, 1968, doi: 10.1109/TR.1968.5217517.
 - [51] R. L. Bulfin and C. Y. Liu, “Optimal Allocation of Redundant Components for Large Systems,” *IEEE Trans. Reliab.*, vol. R-34, no. 3, pp. 241–247, 1985, doi: 10.1109/TR.1985.5222133.
 - [52] N. Nahas, M. Nourelfath, and D. Ait-Kadi, “Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 2, pp. 211–222, 2007, doi: 10.1016/j.ress.2005.12.002.
 - [53] A. Özdemir, “Güvenilirlik analizi ve elektrik enerji sistemlerine uygulanması, ders notları,” 1996.
 - [54] F. A. Tillman, C.-L. Hwang, and W. Kuo, *Optimization of systems reliability*. M. Dekker, 1980.
 - [55] D. W. Coit and E. Zio, “The evolution of system reliability optimization,” *Reliab. Eng. Syst. Saf.*, vol. 192, no. May 2018, p. 106259, 2019, doi: 10.1016/j.ress.2018.09.008.
 - [56] P. M. Ghare and R. E. Taylor, “Optimal Redundancy for Reliability in Series Systems,” vol. 17, no. 5, pp. 838–847, 1969.
 - [57] D. W. Coit and J. Liu, “System reliability optimization with k-out-of-n subsystems,” *Int. J. Reliab. Qual. Saf. Eng.*, vol. 7, no. 2, pp. 129–142, 2000, doi: 10.1142/S0218539300000110.
 - [58] K. B. Misra and U. Sharma, “An efficient algorithm to solve integer-programming problems arising in system-reliability design,” *IEEE Trans. Reliab.*, vol. 40, no. 1, pp. 81–91, 1991, doi: 10.1109/24.75341.
 - [59] M. Gen, K. Ida, Y. Tsujimura, and C. E. Kim, “Large-scale 0-1 fuzzy goal programming and its application to reliability optimization problem,” *Comput. Ind. Eng.*, vol. 24, no. 4, pp. 539–549, 1993, doi: 10.1016/0360-8352(93)90196-5.
 - [60] M. Gen, K. Ida, and J. -U Lee, “A computational algorithm for solving 0-1 goal programming with GUB structures and its application for optimization problems in system reliability,” *Electron. Commun. Japan (Part III Fundam. Electron. Sci.)*, vol. 73, no. 12, pp. 88–96, 1990, doi: 10.1002/ecjc.4430731210.
 - [61] Y. C. Hsieh, “A two-phase linear programming approach for redundancy allocation problems,” *Yugosl. J. Oper. Res.*, vol. 12, no. 2, pp. 227–236, 2002, doi: 10.2298/YJOR0202227H.
 - [62] Y. C. Hsieh, “A linear approximation for redundant reliability problems with multiple component choices,” *Comput. Ind. Eng.*, vol. 44, no. 1, pp. 91–103, 2003, doi: 10.1016/S0360-8352(02)00186-9.
 - [63] J. E. Ramirez-Marquez, D. W. Coit, and A. Konak, “Redundancy allocation for series-parallel systems using a max-min approach,” *IIE Trans. (Institute Ind. Eng.)*, vol. 36, no. 9, pp. 891–898, 2004, doi: 10.1080/07408170490473097.
 - [64] A. Billionnet, “Redundancy allocation for series-parallel systems using integer linear programming,” *IEEE Trans. Reliab.*, vol. 57, no. 3, pp. 507–516, 2008, doi: 10.1109/TR.2008.927807.
 - [65] S. V. Amari and G. Dill, “Redundancy optimization problem with warm-standby redundancy,” *Proc. - Annu. Reliab. Maintainab. Symp.*, vol. c, pp. 0–5, 2010, doi: 10.1109/RAMS.2010.5448068.
 - [66] H. Tekiner-Mogulkoc and D. W. Coit, “System reliability optimization considering

- uncertainty: Minimization of the coefficient of variation for series-parallel systems,” *IEEE Trans. Reliab.*, vol. 60, no. 3, pp. 667–674, 2011, doi: 10.1109/TR.2011.2161029.
- [67] F. A. Tillman, C.-L. Hwang, and W. Kuo, “Optimization techniques for system reliability: a review,” *Reliab. Eng. Syst. Saf.*, vol. R-26, no. 3, pp. 148–155, 1977, doi: 10.1016/0951-8320(92)90033-H.
- [68] F. A. Tillman, C. L. Hwang, and W. Kuo, “Determining Component Reliability and Redundancy for Optimum System Reliability,” *IEEE Trans. Reliab.*, vol. R-26, no. 3, pp. 162–165, 1977, doi: 10.1109/TR.1977.5220102.
- [69] D. W. Coit and A. E. Smith, “Use of a Genetic Algorithm To Optimize a Combinatorial Reliability Design Problem,” *Proc. Third IIE Res. Conf.*, pp. 467–472, 1994.
- [70] D. W. Coit, “Maximization of system reliability with a choice of redundancy strategies,” *IIE Trans. (Institute Ind. Eng.)*, vol. 35, no. 6, pp. 535–543, 2003, doi: 10.1080/074081703004420.
- [71] R. Tavakkoli-Moghaddam, J. Safari, and F. Sassani, “Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm,” *Reliab. Eng. Syst. Saf.*, vol. 93, no. 4, pp. 550–556, 2008, doi: 10.1016/j.res.2007.02.009.
- [72] H. Gholinezhad and A. Zeinal Hamadani, “A new model for the redundancy allocation problem with component mixing and mixed redundancy strategy,” *Reliab. Eng. Syst. Saf.*, vol. 164, no. March, pp. 66–73, 2017, doi: 10.1016/j.res.2017.03.009.
- [73] M. Feizabadi and A. E. Jahromi, “A new model for reliability optimization of series-parallel systems with non-homogeneous components,” *Reliab. Eng. Syst. Saf.*, vol. 157, pp. 101–112, 2017, doi: 10.1016/j.res.2016.08.023.
- [74] S. J. Sadjadi and R. Soltani, “An efficient heuristic versus a robust hybrid meta-heuristic for general framework of serial-parallel redundancy problem,” *Reliab. Eng. Syst. Saf.*, vol. 94, no. 11, pp. 1703–1710, 2009, doi: 10.1016/j.res.2009.05.003.
- [75] A. A. Najafi, H. Karimi, A. Chambari, and F. Azimi, “Two metaheuristics for solving the reliability redundancy allocation problem to maximize mean time to failure of a series-parallel system,” *Sci. Iran.*, vol. 20, no. 3, pp. 832–838, 2013, doi: 10.1016/j.scient.2012.12.022.
- [76] X. Kong, L. Gao, H. Ouyang, and S. Li, “Solving the redundancy allocation problem with multiple strategy choices using a new simplified particle swarm optimization,” *Reliab. Eng. Syst. Saf.*, vol. 144, pp. 147–158, 2015, doi: 10.1016/j.res.2015.07.019.
- [77] N. Beji, B. Jarboui, M. Eddaly, and H. Chabchoub, “A hybrid particle swarm optimization algorithm for the redundancy allocation problem,” *J. Comput. Sci.*, vol. 1, no. 3, pp. 159–167, 2010, doi: 10.1016/j.jocs.2010.06.001.
- [78] H. Garg, M. Rani, and S. P. Sharma, “An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique,” *Comput. Oper. Res.*, vol. 40, no. 12, pp. 2961–2969, 2013, doi: 10.1016/j.cor.2013.07.014.
- [79] J. Safari and R. Tavakkoli-Moghaddam, “A redundancy allocation problem with the choice of redundancy strategies by a memetic algorithm,” *J. Ind. Eng. Int.*, vol. 6, no. 11, pp. 6–16, 2010.
- [80] N. Nahas, M. N. Darghouth, A. Q. Kara, and M. Nourelfath, “Non-linear threshold algorithm based solution for the redundancy allocation problem considering multiple redundancy strategies,” *J. Qual. Maint. Eng.*, vol. 25, no. 3, pp. 397–411, 2019, doi: 10.1108/JQME-05-2018-0041.

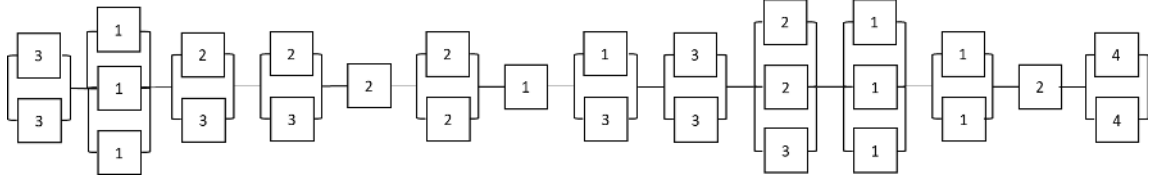
- [81] J. Safari, "Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies," *Reliab. Eng. Syst. Saf.*, vol. 108, pp. 10–20, 2012, doi: 10.1016/j.ress.2012.06.001.
- [82] A. Eshraghniaie Jahromi and M. Feizabadi, "Optimization of multi-objective redundancy allocation problem with non-homogeneous components," *Comput. Ind. Eng.*, vol. 108, pp. 111–123, 2017, doi: 10.1016/j.cie.2017.04.009.
- [83] M. A. Ardakan, A. Z. Hamadani, and M. Alinaghian, "Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy," *ISA Trans.*, vol. 55, pp. 116–128, 2015, doi: 10.1016/j.isatra.2014.10.002.
- [84] J. S. M. EbrahimNezhad, B. Maleki Vishkaei, H. R. Pasandideh, "Increasing the Reliability and the Profit in a Redundancy Allocation Problem," *Int. J. Appl. Oper. Res.*, vol. 1, no. 2, pp. 57–64, 2011.
- [85] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "Ant algorithm for single and multiobjective reliability optimization problems," *Qual. Reliab. Eng. Int.*, vol. 18, no. 6, pp. 497–514, 2002, doi: 10.1002/qre.499.
- [86] J. H. Zhao, Z. Liu, and M. T. Dao, "Reliability optimization using multiobjective ant colony system approaches," *Reliab. Eng. Syst. Saf.*, vol. 92, no. 1, pp. 109–120, 2007, doi: 10.1016/j.ress.2005.12.001.
- [87] E. Zhang, Y. Wu, and Q. Chen, "A practical approach for solving multi-objective reliability redundancy allocation problems using extended bare-bones particle swarm optimization," *Reliab. Eng. Syst. Saf.*, vol. 127, pp. 65–76, 2014, doi: 10.1016/j.ress.2014.03.006.
- [88] A. Zaretalab, V. Hajipour, M. Sharifi, and M. R. Shahriari, "A knowledge-based archive multi-objective simulated annealing algorithm to optimize series-parallel system with choice of redundancy strategies," *Comput. Ind. Eng.*, vol. 80, pp. 33–44, 2015, doi: 10.1016/j.cie.2014.11.008.
- [89] M. R. Valaei and J. Behnamian, "Allocation and sequencing in 1-out-of-N heterogeneous cold-standby systems: Multi-objective harmony search with dynamic parameters tuning," *Reliab. Eng. Syst. Saf.*, vol. 157, pp. 78–86, 2017, doi: 10.1016/j.ress.2016.08.022.
- [90] A. Dolatshahi-Zand and K. Khalili-Damghani, "Design of SCADA water resource management control center by a bi-objective redundancy allocation problem and particle swarm optimization," *Reliab. Eng. Syst. Saf.*, vol. 133, pp. 11–21, 2015, doi: 10.1016/j.ress.2014.07.020.
- [91] E. Zhang and Q. Chen, "Multi-objective reliability redundancy allocation in an interval environment using particle swarm optimization," *Reliab. Eng. Syst. Saf.*, vol. 145, pp. 83–92, 2016, doi: 10.1016/j.ress.2015.09.008.
- [92] M. Teimouri, A. Zaretalab, S. T. A. Niaki, and M. Sharifi, "An efficient memory-based electromagnetism-like mechanism for the redundancy allocation problem," *Appl. Soft Comput. J.*, vol. 38, pp. 423–436, 2016, doi: 10.1016/j.asoc.2015.09.043.
- [93] W. C. Yeh, "A MCS-RSM approach for network reliability to minimise the total cost," *Int. J. Adv. Manuf. Technol.*, vol. 22, no. 9–10, pp. 681–688, 2003, doi: 10.1007/s00170-003-1560-y.
- [94] D. W. Coit, T. Jin, and N. Wattanapongsakorn, "System Optimization With Component Reliability Estimation Uncertainty: A Multi-Criteria Approach," *IEEE Trans. Reliab.*, vol. 53, no. 3, pp. 369–380, Sep. 2004, doi: 10.1109/TR.2004.833312.
- [95] M. Marseguerra, E. Zio, L. Podofillini, and D. W. Coit, "Optimal Design of Reliable Network Systems in Presence of Uncertainty," *IEEE Trans. Reliab.*, vol. 54, no. 2, pp. 243–253, Jun. 2005, doi: 10.1109/TR.2005.847279.

- [96] V. S. S. Yadavalli, A. Malada, and V. Charles, "Reliability stochastic optimization for an n-stage series system with m chance constraints," *S. Afr. J. Sci.*, vol. 103, no. 11–12, pp. 502–504, 2007.
- [97] X. Li and X. Hu, "Some new stochastic comparisons for redundancy allocations in series and parallel systems," *Stat. Probab. Lett.*, vol. 78, no. 18, pp. 3388–3394, 2008, doi: 10.1016/j.spl.2008.07.023.
- [98] O. Dengiz, F. Altiparmak, B. Dengiz, and A. E. Smith, "A Genetic Algorithm for Fuzzy Optimization of the Multi-objective Redundancy Allocation Problem," 2005.
- [99] G. S. Mahapatra and T. K. Roy, "Optimal redundancy allocation in series-parallel system using generalized fuzzy number," *Tamsui Oxford J. Math. Sci.*, vol. 27, no. 1, pp. 1–20, 2011, doi: 10.6988/TOJIMS.201102.0001.
- [100] V. Ebrahimipour and M. Sheikhalishahi, "Application of multi-objective particle swarm optimization to solve a fuzzy multi-objective reliability redundancy allocation problem," *2011 IEEE Int. Syst. Conf. SysCon 2011 - Proc.*, pp. 326–333, 2011, doi: 10.1109/SYSCON.2011.5929085.
- [101] Y. S. Juang, S. S. Lin, and H. P. Kao, "A knowledge management system for series-parallel availability optimization and design," *Expert Syst. Appl.*, vol. 34, no. 1, pp. 181–193, 2008, doi: 10.1016/j.eswa.2006.08.023.
- [102] G. Jiansheng, W. Zutong, Z. Mingfa, and W. Ying, "Uncertain multiobjective redundancy allocation problem of repairable systems based on artificial bee colony algorithm," *Chinese J. Aeronaut.*, vol. 27, no. 6, pp. 1477–1487, 2014, doi: 10.1016/j.cja.2014.10.014.
- [103] F. Kayedpour, M. Amiri, M. Rafizadeh, and A. Shahryari Nia, "Multi-objective redundancy allocation problem for a system with repairable components considering instantaneous availability and strategy selection," *Reliab. Eng. Syst. Saf.*, vol. 160, no. October 2016, pp. 11–20, 2017, doi: 10.1016/j.ress.2016.10.009.
- [104] P. P. Guilani, P. Azimi, S. T. A. Niaki, and S. A. A. Niaki, "Redundancy allocation problem of a system with increasing failure rates of components based on Weibull distribution: A simulation-based optimization approach," *Reliab. Eng. Syst. Saf.*, vol. 152, pp. 187–196, 2016, doi: 10.1016/j.ress.2016.03.010.
- [105] S. Bosse, M. Splieth, and K. Turowski, "Multi-objective optimization of IT service availability and costs," *Reliab. Eng. Syst. Saf.*, vol. 147, pp. 142–155, 2016, doi: 10.1016/j.ress.2015.11.004.
- [106] J. Safari, "Multi-objective reliability optimization of series-parallel systems with a choice of redundancy strategies," *Reliab. Eng. Syst. Saf.*, vol. 108, pp. 10–20, 2012, doi: 10.1016/j.ress.2012.06.001.
- [107] S. Devi and D. Garg, "Hybrid genetic and particle swarm algorithm: redundancy allocation problem," *International Journal of Systems Assurance Engineering and Management*, vol. 11, no. 2, pp. 313–319, 2020, doi: 10.1007/s13198-019-00858-x.
- [108] O. Berman and N. Ashrafi, "Optimization models for reliability of modular software systems," *IEEE Trans. Softw. Eng.*, vol. 19, no. 11, pp. 1119–1123, 1993, doi: 10.1109/32.256858.
- [109] D. W. Coit, "OPTIMIZATION OF RELIABILITY DESIGN PROBLEMS CONSIDERING UNCERTAINTY IN COMPONENT RELIABILITY AND TIME-TO-FAILURE," 1996.
- [110] F. A. Tillman and J. M. Liittschwager, "Integer Programming Formulation of Constrained Reliability Problems," *Manage. Sci.*, vol. 13, no. 11, pp. 887–899, 1967, doi: 10.1287/mnsc.13.11.887.
- [111] R. Luus, "Optimization of System Reliability by a New Nonlinear Integer

- Programming Procedure,” *IEEE Trans. Reliab.*, vol. R-24, no. 1, pp. 14–16, 1975, doi: 10.1109/TR.1975.5215316.
- [112] W. Y. Yun, Y. M. Song, and H. G. Kim, “Multiple multi-level redundancy allocation in series systems,” *Reliab. Eng. Syst. Saf.*, vol. 92, no. 3, pp. 308–313, 2007, doi: 10.1016/j.ress.2006.04.006.
- [113] D. W. Coit and A. Konak, “Multiple Weighted Objectives Heuristic for the Redundancy Allocation Problem,” vol. 55, no. 3, pp. 551–558, 2006.
- [114] J. LI, Y. CHEN, Y. ZHANG, and H. HUANG, “Availability modeling for periodically inspection system with different lifetime and repair-time distribution,” *Chinese J. Aeronaut.*, vol. 32, no. 7, pp. 1667–1672, 2019, doi: 10.1016/j.cja.2019.03.025.
- [115] D. W. Coit and A. E. Smith, “Redundancy allocation to maximize a lower percentile of the system time-to-failure distribution,” *IEEE Trans. Reliab.*, vol. 47, no. 1, pp. 79–87, Mar. 1998, doi: 10.1109/24.690912.
- [116] Ö. Yeniay, “Mathematical and Computational Applications ,” *Math. Comput. Appl.*, vol. 10, no. 1, pp. 45–56, 2005.
- [117] Z. MICHAŁEWICZ, “A Survey of Constraint Handling Techniques in Evolutionary Computation Methods,” in *4th Annual Conference on Evolutionary Programming*, 1995, pp. 135–155.
- [118] A. Smith and D. W. Coit, “Penalty Functions,” in *Handbook of Evolutionary Computation*, 1995.
- [119] S. AUSTIN, “An introduction to genetic algorithms,” *AI Expert*, pp. 49–53, 1991.
- [120] S. Luke, *Essentials of Metaheuristics*. 2009.
- [121] A. Agapie and A. H. Wright, “Theoretical analysis of steady state genetic algorithms,” *Appl. Math.*, vol. 59, no. 5, pp. 509–525, 2014, doi: 10.1007/s10492-014-0069-z.
- [122] F. Altıparmak, M. Gen, L. Lin, and I. Karaoglan, “A steady-state genetic algorithm for multi-product supply chain network design,” *Comput. Ind. Eng.*, vol. 56, no. 2, pp. 521–537, 2009, doi: 10.1016/j.cie.2007.05.012.
- [123] B. Dengiz and F. Altıparmak, “Genetik Algoritmalar Genel Bir Giriş,” *Endüstri Mühendisliği Derg.*, vol. 9, no. 3, pp. 3–14, 1998.
- [124] G. G. Emel and Ç. Taşkın, “Genetik Algoritmalar ve Uygulama Alanları,” *Uludağ Üniversitesi İktisadi ve İdari Bilim. Fakültesi*, vol. 21, no. 1, pp. 129–152, 2002.
- [125] L. Booker, *Improving Search in Genetic Algorithms, Genetic Algorithms and Simulated Annealing*. Los Altos, CA: Morgan Kaufman Publishers, Inc., 1987.
- [126] J. BAKER, “Adaptive selection methods for genetic algorithms,” in *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 1985, pp. 101–111.
- [127] D. E. Goldberg and K. Deb, “A Comparative Analysis of Selection Schemes Used in Genetic Algorithms,” pp. 69–93, 1991, doi: 10.1016/b978-0-08-050684-5.50008-2.
- [128] Z. MICHAŁEWICZ, *Genetic Algorithms and Data Structures, Evolution Programs*, 3rd editio. Springer-Verlag Berlin Heidelberg, 1992.
- [129] G. SYSWERDA, “Uniform crossover in genetic algorithms,” 1989.
- [130] E. Falkenauer, “The worth of the uniform [uniform crossover],” 1999.
- [131] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, “A solution to the unit-commitment problem using integer-coded genetic algorithm,” *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1165–1172, 2004, doi: 10.1109/TPWRS.2003.821625.
- [132] M. Norman and P. Moscato, “A Competitive-Cooperative Approach to Complex Combinatorial Search,” *20TH toint Conf. informatics Oper. Res.*, pp. 15–29, 1991, [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.44.776&rep=rep1&type=pdf>

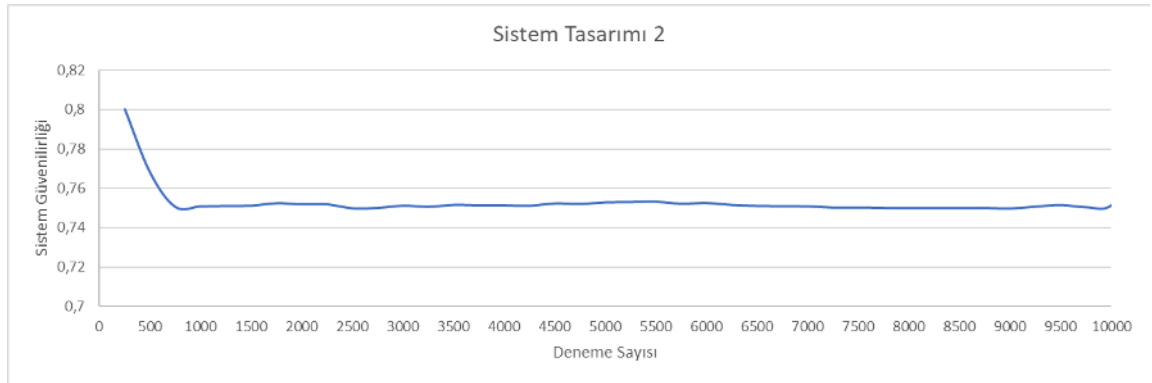
EK 1a: Örnek sistem tasarımları ve tahmini sistem güvenilirliği değerleri

Tablo 5.1’de verilen bileşen özellikleri kullanılarak rassal olarak oluşturulan diğer bir örnek sistem tasarımı Şekil 1’de görülmektedir. Sistem güvenilirlik değeri ise analitik yöntemlerle hesaplanmıştır.



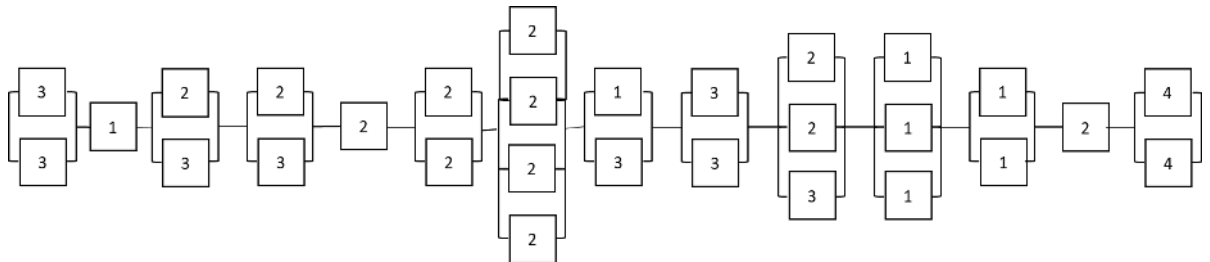
Şekil 1. Sistem Tasarımı 2 (S2)

Elde edilen sonuçlara göre analitik olarak hesaplanan sistem güvenilirliği 0,7520 olan ve şekil 1’de verilen S2’nin sistem güvenilirliğinin tahmin değerinin değişimi şekil 2’de verilmiştir. Grafikte görüldüğü üzere, yaklaşık 3000 deneme sonrasında analitik çözüme yakınsama sağlanabilmektedir.



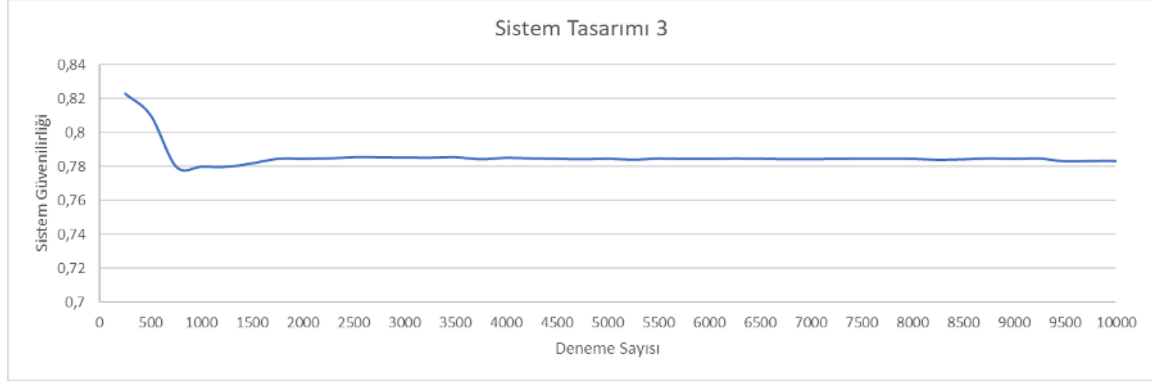
Şekil 2. Tahmini sistem güvenilirliği değerleri

Tablo 5.1’de verilen bileşen özellikleri kullanılarak rassal olarak oluşturulan diğer bir örnek sistem tasarımı Şekil 3’te görülmektedir. Sistem güvenilirlik değeri ise analitik yöntemlerle hesaplanmıştır.



Şekil 3. Örnek Sistem Tasarımı 3 (S3)

Elde edilen sonuçlara göre analitik olarak hesaplanan sistem güvenilirliği 0,7852 olan ve şekil 3'te verilen S3'ün sistem güvenilirliğinin tahmin değerinin değişimi şekil 4'te verilmiştir. Grafikte görüldüğü üzere, yaklaşık 2000 deneme sonrasında analitik çözüme yakınsama sağlanabilmektedir.



Şekil 4. Tahmini sistem güvenilirliği değerleri

EK 1b: Tahmini sistem güvenilirliği kesikli olaylı Monte Carlo benzetimi sonuçları

[illegible]

EK 2: Farklı problemler için tamir edilebilen bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

P1 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	P1	Güvenirlilik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
GA	Maliyet Limiti=130	0,996	121	182	3 3 0 3 0	1 1 0 0 0	4 4 0 0 4	2 2 0 3 0	1 0 0 1 1	0 0 1 0 3	1 2 0 0 0	0 4 4 2 0	4 0 3 0 0	2 2 3 0 0	1 4 0 0 4	0 4 0 4 1	0 2 0 2 0	4 0 0 0 2
	Maliyet Limiti=121	0,992667	116	190	0 4 0 1 1	0 0 3 0 1	0 2 0 4 3	2 0 1 0 3	0 0 0 1 4	0 0 0 4 1	0 0 0 1 1	0 2 1 0 2	1 3 0 0 0	0 0 3 3 3	0 0 2 2 0	0 4 0 4 4	1 0 0 2 0	0 0 0 4 2
	Maliyet Limiti=116	0,994667	113	191	3 0 0 1 4	1 0 0 0 2	3 4 4 0 0	0 0 1 1 1	4 4 0 1 0	3 0 1 0 0	1 1 0 0 0	2 0 2 4 0	3 0 4 0 0	0 0 2 3 2	2 0 0 0 1	0 4 0 4 1	2 0 0 0 1	0 4 4 0 0
	Maliyet Limiti=113	0,993333	110	191	0 0 1 3 3	0 1 0 0 2	0 0 1 3 2	3 1 3 0 0	4 1 1 0 0	0 1 0 0 4	0 2 0 3 0	0 2 4 4 0	0 0 4 3 0	0 0 3 2 1	0 0 0 4 1	0 4 0 1 1	1 0 1 0 0	0 4 1 0 0
	Maliyet Limiti=110	0,994000	110	190	3 1 0 0 2	1 0 0 1 0	0 2 4 3 0	3 0 3 2 0	1 0 4 4 0	4 0 0 4 0	0 3 0 2 0	4 0 2 4 0	0 4 4 0 0	3 2 0 0 3	0 0 4 0 1	4 1 4 4 0	0 0 1 0 1	0 0 1 0 2
	Maliyet Limiti=109	0,991333	108	191	0 4 0 1 2	1 3 0 0 0	0 2 4 3 0	0 1 2 2 0	1 0 4 0 1	3 1 0 0 0	1 0 0 1 0	0 4 2 0 4	0 0 0 1 4	0 3 3 0 0	2 0 4 0 0	4 4 0 4 0	0 2 0 2 0	0 4 0 0 4
	Maliyet Limiti=108	0,990667	108	188	3 0 3 1 0	3 1 0 0 0	0 2 1 2 0	1 0 3 2 0	0 0 4 4 1	0 0 4 4 0	0 2 0 2 0	0 0 4 4 2	0 3 0 4 0	0 3 0 2 3	0 1 0 4 0	4 4 0 4 0	0 0 1 0 2	1 0 1 0 0
	Maliyet Limiti=107	0,992667	106	190	2 0 3 1 0	0 0 1 0 2	3 0 0 1 4	0 1 0 1 3	0 0 1 2 0	0 0 1 3 0	0 0 3 0 0	4 4 4 0 0	0 0 4 0 4	1 1 0 0 1	2 0 4 0 0	0 4 3 0 1	2 0 0 0 2	0 0 2 0 2
	Maliyet Limiti=106	0,994667	106	191	3 3 0 0 1	2 0 0 2 0	4 1 3 0 0	3 1 0 0 3	0 4 0 0 2	0 4 0 0 3	0 0 1 1 1	4 0 0 4 4	0 0 0 4 4	2 0 0 3 2	4 4 4 0 0	4 0 1 4 0	0 0 1 0 2	2 0 2 0 0
	Maliyet Limiti=105	0,992667	105	189	0 1 3 2 0	0 0 0 1 3	2 3 0 0 3	2 0 1 3 0	0 0 4 1 0	0 4 0 0 4	1 0 0 0 3	2 4 0 2 0	4 1 0 0 0	0 0 2 2 2	4 0 0 2 0	4 0 0 3 4	1 0 0 0 3	4 2 0 0 0
	Maliyet Limiti=104	0,993333	104	189	0 3 3 1 0	0 1 0 0 1	0 3 1 0 2	1 0 1 0 3	4 0 0 4 0	0 1 1 0 0	1 2 0 0 0	0 0 2 4 4	0 4 0 0 4	1 0 2 0 1	0 4 1 0 4	0 0 3 4 4	0 1 0 3 0	0 0 0 4 4
	Maliyet Limiti=103	0,992667	103	191	2 3 0 3 0	0 0 1 1 0	0 1 3 0 2	2 1 2 0 0	0 1 1 0 4	4 0 0 4 0	3 0 0 0 1	1 0 0 4 1	0 1 0 0 4	2 2 3 0 0	0 4 0 4 0	0 0 4 4 4	0 2 0 0 3	2 0 0 1 0
	Maliyet Limiti=102	0,989333	102	191	0 0 2 3 2	0 2 0 0 2	0 4 0 3 4	1 3 0 1 0	0 1 0 0 4	4 0 0 4 0	0 0 2 3 0	0 2 0 0 1	0 4 0 4 0	0 2 1 0 3	4 0 4 0 4	4 0 1 0 4	0 1 0 0 1	4 0 0 0 1
	Maliyet Limiti=101	0,992667	101	191	3 0 3 2 3	0 0 2 1 0	3 0 0 3 2	3 2 1 0 0	0 0 1 0 1	0 0 4 0 4	3 0 2 0 0	0 4 4 0 4	4 0 0 0 1	0 1 1 0 2	0 0 4 4 0	4 1 0 3 1	0 3 0 0 0	0 1 4 0 0
	Maliyet Limiti=100	0,993333	98	189	2 0 1 0 1	2 0 0 0 2	0 0 4 4 3	1 0 1 1 0	0 4 2 0 4	0 4 0 0 4	0 0 0 1 3	0 4 4 4 0	0 4 4 0 0	1 2 0 0 2	4 0 0 4 0	4 4 0 1 0	1 2 0 0 0	0 0 4 0 2
	Maliyet Limiti=98	0,993333	97	191	3 2 2 0 3	2 0 0 0 1	0 3 0 3 2	0 0 1 1 2	4 0 4 1 0	4 0 0 4 0	0 3 3 0 0	4 0 4 0 4	3 0 4 0 0	0 2 1 1 0	0 0 4 4 0	0 4 0 4 4	0 0 0 3 1	0 2 0 0 1
	Maliyet Limiti=97	0,992	96	185	0 1 3 3 0	0 0 3 3 0	2 3 3 0 0	1 0 0 2 1	4 1 0 1 0	0 4 4 0 0	0 0 1 0 2	4 4 0 0 4	4 0 0 0 1	3 2 1 0 0	2 0 0 4 0	4 1 4 0 0	0 0 0 3 1	0 2 0 0 4
	Maliyet Limiti=96	0,991333	96	188	2 3 3 0 0	1 0 0 0 2	0 3 0 3 3	1 1 0 1 0	1 1 0 1 0	0 0 0 3 4	1 0 0 3 0	4 4 0 4 0	4 0 4 0 0	2 0 3 1 0	4 2 0 0 0	4 4 0 4 0	0 1 0 0 1	0 1 2 0 0
	Maliyet Limiti=95	0,991333	95	190	1 1 0 0 2	0 2 0 0 2	4 2 0 0 3	2 0 1 1 0	1 1 0 0 0	0 0 0 3 4	0 0 0 2 3	4 4 0 0 4	0 0 4 0 1	1 3 1 0 0	4 4 0 0 0	1 0 4 4 0	1 0 0 0 3	0 0 1 0 4
	Maliyet Limiti=94	0,990667	94	190	0 1 2 0 2	0 0 2 1 0	0 3 1 3 0	2 1 0 2 0	0 0 4 1 0	3 0 3 0 0	0 0 2 1 0	4 4 4 0 0	0 4 0 1 0	2 0 2 0 3	0 2 4 0 0	0 3 4 4 0	0 0 1 0 3	0 0 2 0 2
	Maliyet Limiti=93	0,989333	93	186	3 0 2 0 4	0 3 2 0 0	3 0 3 3 0	0 1 0 1 1	0 0 1 0 4	4 0 0 4 0	0 2 1 0 0	4 4 0 0 4	0 4 0 0 1	0 0 1 3 2	1 4 0 0 0	0 4 0 4 1	2 2 0 0 0	0 2 0 0 4
	Maliyet Limiti=92	0,989333	90	190	2 0 2 0 1	0 3 2 0 0	0 0 3 1 1	1 1 0 0 1	1 0 1 0 0	0 0 3 0 4	1 0 1 0 0	4 4 4 0 0	1 4 0 0 0	2 1 3 0 0	4 4 0 0 0	0 1 4 1 0	0 0 1 0 3	0 0 4 4 0
	Maliyet Limiti=90	0,990667	90	189	1 0 0 1 1	1 0 0 2 0	0 3 3 0 3	1 0 1 0 1	2 0 0 0 1	0 0 3 3 0	0 2 3 0 0	4 4 0 0 4	4 0 3 0 0	2 0 1 2 0	0 0 0 1 4	4 0 4 0 4	0 3 3 0 0	4 0 0 0 1
	Maliyet Limiti=89	0,981333	89	191	0 0 2 1 1	3 2 0 0 0	0 0 3 3 3	2 0 0 1 1	1 0 1 1 0	3 0 0 0 4	3 0 0 0 3	4 0 4 1 0	4 0 4 0 0	3 0 0 0 2	4 0 0 0 4	0 1 4 4 0	0 0 0 3 1	0 0 1 4 0

P3 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr3	Güvenilirlik	Maliyet	Ağırlık	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem	Altsistem
GA	Maliyet Limiti=130	0,995333	126	189	0 3 0 3 3	1 0 1 0 0	1 2 0 0 4	3 3 0 0 3	0 1 0 2 0	2 2 0 0 0	3 1 0 0 0	0 3 3 1 0	0 0 4 0 2	2 0 2 0 1	3 2 0 0 0	3 4 3 0 0	0 2 0 1 0	0 1 3 1 0
	Maliyet Limiti=126	0,994667	118	188	0 2 0 3 1	1 1 0 0 0	0 0 4 4 2	0 1 0 3 1	1 2 0 3 0	0 2 1 0 0	0 0 2 0 1	0 0 1 1 3	0 2 3 0 0	3 0 0 3 3	0 0 0 3 1	2 0 3 0 2	0 2 0 0 1	0 0 4 1 0
	Maliyet Limiti=118	0,995333	116	189	3 0 4 3 0	0 1 2 0 0	0 0 1 4 4	1 3 0 0 3	0 1 2 0 0	0 2 4 0 0	0 0 1 1 1	0 0 1 1 1	0 0 2 0 3	2 0 2 1 0	0 0 0 2 1	2 0 0 4 3	0 0 2 1 0	0 0 3 0 4
	Maliyet Limiti=116	0,989333	107	188	0 1 2 0 3	1 0 0 0 2	0 3 3 4 0	3 0 3 2 0	3 0 0 1 3	0 0 0 3 1	3 0 0 3 0	0 3 1 1 0	2 3 0 0 0	0 3 0 2 0	2 2 1 0 0	0 1 2 0 0	1 1 2 0 0	0 4 0 0 0
	Maliyet Limiti=107	0,993333	107	189	3 0 0 0 4	0 0 1 3 0	0 4 1 2 0	2 0 1 1 0	1 2 0 0 1	0 0 0 1 4	1 0 3 0 0	0 0 1 1 1	1 0 0 1 0	0 0 2 2 3	3 0 0 0 3	1 4 1 0 0	2 2 0 0 0	0 4 0 2 0
	Maliyet Limiti=106	0,986667	106	189	3 1 0 2 0	0 1 0 3 0	0 4 0 0 3	1 0 1 3 0	1 0 0 1 0	4 4 0 2 0	0 0 3 0 3	3 0 2 0 0	1 0 0 2 0	2 3 0 2 0	0 0 3 2 0	2 2 0 0 3	1 0 0 3 0	0 4 0 2 0
	Maliyet Limiti=105	0,992	105	189	2 0 4 0 0	0 2 3 0 0	0 2 0 3 4	0 3 0 2 1	0 0 0 3 2	2 0 0 0 1	4 1 0 3 0	0 1 0 1 1	0 0 2 1 0	1 3 0 0 2	2 0 0 0 2	0 0 2 3 1	0 2 2 0 0	0 4 0 0 3
	Maliyet Limiti=104	0,994	104	189	2 1 0 0 2	0 0 1 0 1	0 3 2 0 4	1 3 0 0 1	1 0 0 2 1	0 4 0 4 0	3 0 0 0 3	0 1 1 2 0	0 2 0 4 0	2 0 1 0 2	2 2 1 0 0	0 1 1 1 0	0 0 1 2 0	0 0 3 4 0
	Maliyet Limiti=103	0,990667	103	189	1 0 3 0 2	0 1 0 3 3	4 0 4 3 0	1 0 1 0 3	2 0 2 0 1	0 3 0 2 0	0 0 1 0 3	3 0 0 0 2	0 1 2 0 0	3 0 0 2 3	1 0 1 0 0	0 1 0 1 1	0 0 1 1 0	1 0 0 0 3
	Maliyet Limiti=102	0,986	102	185	1 0 2 0 1	0 2 0 0 2	0 3 3 4 0	0 0 3 3 3	3 0 0 0 2	2 0 0 1 0	1 0 2 0 0	1 0 1 1 0	0 2 0 0 2	0 2 3 1 0	2 1 0 0 0	2 0 0 4 2	0 3 2 0 0	4 0 0 0 0
	Maliyet Limiti=101	0,99	101	185	3 0 0 3 2	0 1 0 2 0	1 0 4 2 0	0 2 0 2 1	0 1 0 2 0	0 0 0 4 3	0 0 3 3 0	1 0 1 0 1	0 4 0 0 1	0 0 3 2 2	1 0 1 0 0	0 2 0 1 3	0 0 3 1 0	0 0 3 0 1
	Maliyet Limiti=100	0,991333	100	189	1 2 0 4 0	0 0 3 0 2	0 0 3 4 2	0 1 0 1 1	2 0 1 0 0	1 0 4 0 0	2 1 0 0 0	3 0 1 1 0	1 1 0 0 0	0 0 2 1 3	0 0 3 1 0	1 0 0 3 1	0 0 2 1 0	0 1 0 4 0
	Maliyet Limiti=99	0,991333	98	188	0 0 2 0 4	0 0 0 1 1	0 0 1 3 4	0 2 1 0 2	2 0 1 0 2	0 0 3 2 0	0 2 0 0 3	1 1 0 1 0	0 0 0 1 1	0 1 0 3 2	0 0 1 1 0	0 2 0 1 0	1 1 0 0 1	0 1 0 4 0
	Maliyet Limiti=98	0,992	98	189	0 0 4 1 1	0 1 0 2 0	3 3 3 0 0	0 0 1 1 1	0 2 0 0 2	2 0 0 4 0	0 2 1 0 0	1 0 2 1 0	0 0 3 0 2	3 0 2 0 2	2 0 0 0 2	2 1 0 0 2	0 0 1 0 2	0 4 0 0 1
	Maliyet Limiti=97	0,988	97	187	1 4 0 2 0	1 0 0 3 0	0 1 0 1 4	1 0 0 1 3	0 1 0 2 0	0 0 4 4 0	3 0 2 0 0	1 0 1 0 1	0 1 2 0 0	1 2 3 0 0	1 3 0 0 0	1 1 0 0 1	3 0 0 2 0	2 0 2 0 0
	Maliyet Limiti=96	0,984667	95	188	2 0 3 2 0	2 1 2 0 0	0 0 0 4 3	2 0 1 2 0	2 0 2 0 1	4 0 0 3 0	1 0 2 0 0	1 1 0 3 0	1 0 0 0 1	0 0 2 3 2	0 1 0 0 2	0 0 2 1 1	1 0 1 0 0	0 0 4 0 0

P5 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 5	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
GA	Maliyet Limiti=130	0,996	128	187	4 0 3 0 3	1 0 0 1 0	0 2 4 0 4	3 3 3 0 3	0 1 2 0 1	0 0 1 0 3	1 3 0 0 0	0 1 0 3 1	3 1 0 0 0	3 0 0 2 3	0 3 0 1 0	0 3 0 4 1	0 0 2 2 0	3 0 0 3 0
	Maliyet Limiti=128	0,996	122	185	1 0 0 4 3	1 0 0 3 0	4 0 0 4 4	3 0 3 3 0	0 2 3 2 0	0 2 0 3 0	1 0 0 0 1	3 1 3 0 0	0 2 0 0 4	2 2 0 2 0	3 0 0 0 3	1 1 0 0 3	0 0 2 2 0	0 4 0 2 0
	Maliyet Limiti=122	0,995333	115	187	0 3 1 0 3	0 1 0 0 3	2 0 4 4 0	0 3 3 0 2	0 0 3 0 3	0 4 0 1 0	3 0 0 3 0	2 0 1 0 1	0 3 2 0 0	2 0 0 3 1	0 0 0 1 1	1 0 0 4 1	0 0 3 2 0	0 0 1 0 4
	Maliyet Limiti=115	0,991333	113	184	0 0 1 1 3	0 0 2 0 1	0 4 0 0 4	1 3 0 0 1	0 2 2 0 2	1 0 3 0 0	0 1 1 0 0	0 1 2 0 1	2 3 0 0 0	1 2 0 0 2	1 1 0 0 0	1 4 0 0 4	2 0 0 2 0	3 3 1 0 0
	Maliyet Limiti=113	0,990667	111	187	0 1 2 1 0	0 3 0 3 0	2 2 0 4 0	0 3 3 0 3	1 1 2 0 0	0 3 0 0 1	0 0 0 2 2	0 2 0 0 3	0 0 0 3 2	0 2 3 1 0	0 3 0 1 0	2 0 0 2 1	0 0 0 1 1	0 0 3 0 4
	Maliyet Limiti=111	0,991333	111	184	0 3 0 1 3	0 0 3 2 0	0 4 2 4 0	2 0 3 0 3	1 1 0 2 0	0 0 1 0 4	0 1 0 0 0	1 1 1 0 0	0 0 2 0 3	0 0 2 0 2	1 0 2 0 3	0 0 2 1 0	2 0 0 0 3	3 0 0 0 3
	Maliyet Limiti=110	0,988	107	187	1 0 2 0 1	2 0 0 3 0	3 2 0 0 4	3 0 2 3 0	1 0 0 2 0	4 0 0 1 0	0 1 0 2 0	2 0 0 0 2	0 2 0 0 3	0 1 0 1 2	0 2 2 0 0	0 0 0 1 2	4 0 0 2 2	0 0 4 3 0
	Maliyet Limiti=107	0,989333	106	187	2 0 1 0 2	0 3 0 1 0	0 2 4 3 0	0 0 3 1 3	2 0 2 2 0	3 0 0 0 1	0 1 0 0 1	2 1 0 2 0	2 0 0 0 1	0 0 3 0 3	1 0 3 0 0	0 0 2 1 3	0 0 0 1 2	0 0 0 4 1
	Maliyet Limiti=106	0,989333	106	187	2 0 1 0 2	0 0 2 2 0	4 0 0 4 1	2 1 0 3 0	0 2 2 0 2	2 2 0 0 0	0 1 0 1 0	2 0 0 0 2	1 0 4 0 0	0 3 0 2 1	1 1 3 0 0	0 0 1 0 3	2 0 0 0 1	0 2 0 1 0
	Maliyet Limiti=105	0,992	101	187	2 1 2 0 0	3 1 0 0 0	0 4 2 2 0	0 3 1 0 1	0 2 1 0 1	4 2 0 0 0	2 0 0 0 1	0 0 1 2 1	0 0 1 0 2	1 0 0 2 2	0 0 0 1 1	0 0 1 2 1	0 1 0 1 0	2 4 0 0 0
	Maliyet Limiti=101	0,984667	100	186	0 0 4 1 0	0 0 3 1 0	0 0 2 1 3	0 2 0 2 1	0 1 0 2 0	3 2 0 0 0	0 2 0 0 2	0 2 0 3 0	4 2 0 0 0	2 2 0 2 0	0 3 0 1 0	4 0 0 1 1	1 0 3 0 0	0 3 3 0 0
	Maliyet Limiti=100	0,990667	100	187	2 1 0 0 3	0 0 2 2 0	3 4 0 3 0	0 3 2 0 3	0 2 0 0 2	0 0 0 2 3	0 0 1 0 2	0 1 2 0 1	2 0 1 0 2	0 0 1 1 0	0 2 3 1 0	1 0 0 1 2	1 0 0 1 0	2 0 4 0 0
	Maliyet Limiti=99	0,988	98	187	0 3 0 4 2	0 0 0 3 2	3 0 2 0 3	2 2 2 0 0	2 1 0 0 0	0 1 0 4 0	3 0 0 0 3	1 1 1 0 0	0 0 3 3 0	0 0 0 3 2	0 1 0 1 0	1 0 1 1 0	0 1 0 3 0	0 0 0 2
	Maliyet Limiti=98	0,986667	96	184	3 0 1 2 0	3 2 0 0 0	0 2 3 0 3	3 2 0 0 2	0 0 0 2 2	1 0 3 0 0	0 1 1 0 0	0 1 0 1 1	0 4 2 0 0	1 2 0 0 2	1 0 0 2 0	2 0 1 0 2	0 0 2 0 1	0 0 1 2 0
	Maliyet Limiti=96	0,987333	95	185	2 3 0 0 1	3 0 0 0 3	3 0 0 3 2	2 1 0 1 0	1 0 1 0 0	0 0 4 0 4	2 1 0 0 0	0 1 1 3 0	1 0 0 1 0	0 0 0 3 3	0 0 2 0 1	4 0 0 2 1	1 0 0 0 2	0 4 0 0 1
	Maliyet Limiti=95	0,987333	94	187	2 2 0 2 0	1 0 0 3 0	0 3 3 2 0	1 1 0 1 0	0 0 1 0 0	2 0 0 0 2	2 2 0 2 0	0 2 0 3 0	0 0 0 0 1	1 0 2 3 1	0 0 1 2 0	0 1 0 2 2	0 1 3 0 0	0 0 2 0 3
	Maliyet Limiti=94	0,985333	94	184	2 2 0 0 1	2 0 0 2 0	2 0 0 2 1	1 3 0 0 1	2 0 2 0 0	0 0 0 2 3	0 2 2 0 0	0 1 1 0 1	1 0 1 0 0	0 0 2 3 0	0 2 2 0 0	3 0 0 1 1	0 2 0 0 3	0 0 1 0 4
	Maliyet Limiti=93	0,986	93	187	0 0 0 4 1	3 3 0 0 0	0 3 0 3 1	2 0 1 0 2	0 1 0 0 2	0 0 4 4 0	0 1 3 0 0	0 0 1 1 1	0 1 0 4 0	2 0 2 3 0	0 0 0 1 0	1 0 1 0 3	1 0 3 0 1	0 0 4 3 0
	Maliyet Limiti=92	0,990667	91	185	0 3 1 0 1	2 2 0 0 0	0 0 3 3 3	2 0 1 0 1	2 0 0 0 3	0 0 4 0 3	1 3 0 0 0	0 0 1 1 1	0 0 2 0 4	2 0 0 2 2	0 0 0 1 1	1 2 0 0 1	0 3 0 1 0	3 0 0 0 3
	Maliyet Limiti=91	0,987333	90	186	0 1 3 0 2	2 2 0 0 0	3 3 0 1 0	1 1 0 1 0	2 0 0 0 2	0 0 0 2 3	2 0 3 0 0	0 1 0 1 1	3 0 2 0 0	0 0 2 1 2	0 1 0 0 1	0 0 2 1 1	3 0 3 0 0	0 0 1 1 0
	Maliyet Limiti=90	0,988667	88	185	0 4 1 1 0	0 2 0 2 0	0 3 3 0 2	1 1 1 0 0	0 0 0 1 1	0 4 3 0 0	1 0 0 1 0	0 1 0 1 1	2 0 0 0 1	0 1 2 0 2	2 0 0 1 0	1 0 1 0 0	1 1 0 0 1	0 2 0 0 3
	Maliyet Limiti=88	0,985333	88	186	0 0 0 2 4	0 2 0 0 2	3 0 3 0 1	0 0 1 1 2	0 0 2 2 0	4 0 1 0 0	1 0 2 0 0	1 0 0 1 1	0 0 2 0 1	0 0 2 2 1	1 0 1 0 0	0 1 2 2 0	0 1 0 0 1	1 0 2 0 0
	Maliyet Limiti=87	0,987333	87	182	2 0 0 1 1	0 0 1 0 3	3 0 1 3 0	1 1 0 1 0	2 1 0 0 0	3 0 1 0 0	1 0 0 0 1	0 1 1 1 0	1 0 1 0 0	0 1 0 1 3	1 0 0 1 0	0 0 1 1 0	1 0 0 1 0	0 0 1 3 0

P7 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 7	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
GA	Maliyet Limiti=130	0,994667	114	184	2 0 3 3 2 0 0 1 0 3	0 4 3 0 4 3 0 0 3 1	0 0 2 2 1 0 1 4 0 0	0 0 0 1 3 0 0 1 1 3	0 0 1 1 3 0 0 3 3 0 3	0 2 0 2 0 0 0 3 1	1 1 0 0 2 0 1 0 1 0	0 4 3 0 0						
	Maliyet Limiti=114	0,994	114	184	3 0 4 0 0 1 0 3 0 0	4 4 3 0 0 1 3 0 0 3	0 2 1 0 2 0 4 0 0 1	1 1 0 0 1 0 1 3 0 3	0 1 3 0 0 1 0 2 0 2	0 1 0 0 1 2 2 0 1 0	2 0 0 0 1 0 3 3 0 0							
	Maliyet Limiti=113	0,991333	109	181	0 1 1 3 0 1 0 0 0 2	0 0 0 4 4 1 3 0 0 3	2 1 2 0 0 0 0 0 0 1	4 3 0 0 3 0 2 0 1 0	1 0 0 2 0 1 0 2 0 3	2 0 0 1 3 0 1 0 2 1	0 0 2 2 0 0 4 1 0 0							
	Maliyet Limiti=109	0,992667	108	185	2 1 0 2 3 0 0 0 1 3	2 2 0 0 2 2 3 0 3 0	3 0 2 0 0 0 0 0 2 4	0 1 0 0 0 1 0 0 1 0	0 0 1 1 3 2 0 0	1 0 0 3 1 2 0 0	1 0 0 0 3 1 2 0	0 0 2 0 0 3 1 0 0 0 3						
	Maliyet Limiti=108	0,992667	107	184	3 3 0 2 0 0 0 0 1 3	0 4 0 4 3 3 3 0 0 2	2 2 0 0 1 0 1 4 0 0	0 1 1 0 0 0 0 1 3	1 2 1 0 0 0 1 0	2 0 2 0 0 1 0 1	1 0 3 1 0 1 3 0 0 0	0 0 0 4 2						
	Maliyet Limiti=107	0,992	104	182	0 1 1 0 3 0 3 1 0 0	4 0 2 0 0 0 0 1 1	3 2 2 0 2 0 2 4 0 0	0 0 0 2 3 0 3 1	0 1 2 0 0 2 0 0	2 2 0 1 2 0 1 0	0 2 0 1 0 2 0 3	3 0 0 0 4 1 0 0						
	Maliyet Limiti=105	0,988667	103	180	0 1 2 1 0 3 0 1 0 0	0 0 3 1 4 0 1 3	1 0 0 2 0 2 0 0 0 2	3 1 0 1 0 0 1 3	1 0 0 0 0 1 3 0	0 0 2 1 3 0 3 1	0 0 0 2 1 0 2 2	0 2 0 0 3 0 1 0 0						
	Maliyet Limiti=104	0,992	102	182	2 0 0 1 2 0 3 0 1 0	4 0 2 0 2 3 0 0	1 1 0 0 3 1 0 4	0 1 0 0 0 0 0 1	1 1 1 0 0 0 0 2	3 0 0 0 2 3 2 1	0 0 2 0 1 2 0 1	0 0 0 1 1 0 2 0	4 0 0					
	Maliyet Limiti=107	0,985333	100	182	0 1 3 3 3 0 3 1 0 0	0 0 3 3 2 0 1 3	0 2 2 0 1 0 0 0 3	0 4 3 0 1 0 0 1	0 1 0 1 1 0 0 0	2 2 0 3 0 3 1 3	0 0 0 1 0 2 3 0	3 0 2 0 0 0 0 4 0						
	Maliyet Limiti=106	0,990667	100	182	3 3 1 0 0 0 1 0 0 3	0 3 3 0 4 0 1 1	3 0 3 0 2 0 0 0 0 4	1 0 0 3 0 3 2 1	1 0 0 0 1 0 1 0	2 0 2 2 0 0 0 1	3 0 2 0 0 1 1 0	1 0 3 0 0 1 0 0 3						
Maliyet Limiti=105	0,980667	99	182	3 0 0 1 1 2 0 0 1 0	4 4 0 3 0 2 0 0	1 1 3 1 0 0 0 3	3 0 0 0 0 1 3 0	0 0 0 3 2 0 1 0	0 0 1 0 3 3 0 0	1 0 0 2 0 3 2 2	0 0 3 1 0 1 0 0 3 0							

P15 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 15	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
GA	Maliyet Limiti=130	0,992667	119	175	0 3 0 3 3 0 0 0 1 1	0 4 4 0 4 3 0 0 1 3	2 0 0 2 2 1 0 0 0 2	3 0 1 0 0 1 3 1 0 0	0 0 0 2 3 0 0 2 3 3	0 1 0 0 3 1 0 3 1 0	0 0 0 2 3 0 0 2 3 3	0 1 0 0 3 1 0 3 1 0	0 0 0 2 1 0 0 0 2 1	0 0 0 2 1 0 0 0 2 1	0 0 0 2 1 0 0 0 2 1	0 0 0 2 1 0 0 0 2 1	0 0 0 2 1 0 0 0 2 1	0 0 0 2 1 0 0 0 2 1
	Maliyet Limiti=119	0,992	108	176	2 3 3 0 0 0 0 0 2 0	1 4 2 0 0 4 3 1 0 0	3 2 0 2 0 0 0 2 0 0	2 0 2 0 0 0 2 0 0 1	2 0 0 1 2 0 1 1 0 0	3 0 0 0 3 0 0 1 1 0	0 3 0 0 1 1 0 0 1 2	0 2 0 2 0 0 3 0 1 0	1 0 1 0 2 1 0 0 0 1	1 0 0 1 1 4 0 3 0 0	0 0 0 1 1 4 0 3 0 0	0 0 0 1 1 4 0 3 0 0	0 0 0 1 1 4 0 3 0 0	0 0 0 1 1 4 0 3 0 0
	Maliyet Limiti=108	0,991333	108	176	1 3 1 0 0 0 3 0 1 0	4 4 0 0 4 2 2 1 0 0	1 1 0 0 0 2 0 2 0 0	1 2 0 0 0 1 1 3 0 0	0 0 0 3 2 0 2 3 0 0	1 0 1 0 1 1 0 0 1 1	0 0 0 3 2 0 2 3 0 0	1 0 1 0 1 1 0 0 1 1	0 1 0 0 1 0 0 3 3 0	0 1 0 0 1 0 0 3 3 0	0 1 0 0 1 0 0 3 3 0	0 1 0 0 1 0 0 3 3 0	0 1 0 0 1 0 0 3 3 0	0 1 0 0 1 0 0 3 3 0
	Maliyet Limiti=107	0,982	105	177	0 1 0 1 2 1 0 0 0 3	0 2 4 0 0 1 0 0 3 3	0 3 3 0 0 0 2 0 0 1	2 0 0 1 0 3 0 1 0 3	0 0 1 3 0 2 4 0 0 0	3 0 1 3 0 2 4 0 0 0	3 0 0 0 3 0 0 0 2 0	1 0 0 2 1 2 1 0 0 0	0 0 0 3 2 0 4 1 0 0	0 0 0 3 2 0 4 1 0 0	0 0 0 3 2 0 4 1 0 0	0 0 0 3 2 0 4 1 0 0	0 0 0 3 2 0 4 1 0 0	0 0 0 3 2 0 4 1 0 0
	Maliyet Limiti=105	0,982667	104	176	0 0 2 1 0 0 0 3 0 1	4 0 2 0 4 0 0 1 3 3	1 0 2 0 2 0 0 0 1 3	1 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3	0 0 0 3 0 0 0 0 1 3
	Maliyet Limiti=104	0,987333	99	175	0 3 3 1 0 1 2 0 0 0	3 2 4 0 0 3 0 2 3 0	3 2 0 0 0 4 4 0 0 0	2 0 0 1 0 0 2 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0
	Maliyet Limiti=99	0,985333	99	177	4 0 0 1 3 0 0 2 2 0	2 0 0 4 0 0 0 1 3 2	0 0 1 1 0 0 0 0 3 4	0 1 1 0 0 1 0 0 1 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0	0 0 0 3 4 0 1 1 0 0
	Maliyet Limiti=98	0,982667	97	175	1 1 0 0 3 2 0 2 0 0	0 4 0 0 3 0 3 1 1 0	2 0 2 2 0 0 0 1 0 1	0 0 1 0 0 1 0 2 1 0	1 0 1 0 2 0 1 0 3 0	3 0 0 2 0 1 0 0 1 1	0 0 4 0 1 0 2 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0	1 0 2 0 0 1 1 0 4 0

P16 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 16	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14																																																							
GA	Maliyet Limiti=130	0,994667	117	176	3	0	1	1	0	0	0	1	1	0	0	4	4	4	0	0	0	3	3	3	0	2	0	0	2	0	0	1	1	1	0	0	1	0	3	1	0	2	1	0	0	0	2	2	2	0	2	3	0	0	0	0	1	2	0	1	0	0	0	2	1	3	0	3	0	0			
	Maliyet Limiti=117	0,988	107	176	0	0	4	3	0	1	1	0	0	0	0	0	2	0	4	0	1	1	2	0	2	0	2	0	1	2	0	0	2	0	0	1	3	1	0	0	0	0	2	0	4	2	0	0	3	2	0	3	0	1	0	1	2	3	0	0	0	2	0	1	0	0	3	0	2	0			
	Maliyet Limiti=107	0,988	107	176	3	0	3	0	1	1	0	2	0	0	0	0	4	0	2	0	0	3	3	1	2	0	2	0	0	2	4	0	0	0	1	0	3	0	0	0	1	1	2	0	0	0	2	0	2	2	3	0	3	0	0	2	3	0	0	2	0	0	0	3	1	0	4	0	1	0			
	Maliyet Limiti=106	0,986667	105	176	1	1	1	0	0	0	1	1	0	0	4	0	3	0	4	3	0	3	1	0	0	0	0	1	1	0	2	0	1	0	0	0	3	1	0	1	1	3	0	0	0	2	2	0	3	3	0	0	0	2	0	0	1	0	1	0	3	0	0										
	Maliyet Limiti=105	0,984667	101	175	4	0	0	0	2	0	0	2	1	0	2	2	0	2	0	3	0	3	1	0	0	0	0	1	2	0	4	0	3	0	0	3	0	0	1	1	1	0	0	1	2	0	0	3	0	3	0	3	2	0	0	0	0	1	1	2	0	2	0	0	0	2	1	0	0	2	0		
	Maliyet Limiti=101	0,986	101	172	0	0	3	3	1	0	0	2	0	1	0	0	4	0	4	0	0	1	1	1	0	2	1	0	0	4	0	0	0	2	0	0	0	1	1	0	2	0	3	2	1	0	0	0	3	0	2	0	2	1	0	0	3	0	0	1	1	0	3	0	0	0	2	0	0	4	3		
	Maliyet Limiti=100	0,982667	95	174	1	2	0	3	0	1	3	0	0	0	0	2	4	0	0	1	1	0	0	1	1	0	0	2	0	4	0	1	0	0	3	0	0	0	2	0	1	1	0	3	0	0	2	1	0	1	2	0	0	1	0	0	0	4															
	Maliyet Limiti=99	0,976667	94	174	0	2	0	0	4	0	2	1	0	0	0	0	4	0	3	0	3	0	1	1	1	2	0	0	0	0	0	0	1	0	3	1	0	0	0	0	0	2	1	1	1	4	0	0	0	3	0	0	3	1	0	0	0	1	1	1	0	0	1	0	0	0	2	3	3	0	0	0	2
	Maliyet Limiti=98	0,972667	92	171	1	0	1	0	1	1	0	3	0	0	4	0	3	0	3	0	0	2	3	0	3	1	0	0	0	0	0	0	1	2	0	0	1	0	1	1	0	0	1	3	0	1	0	0	2	0	0	0	1	0	4	0	1	1	0	0	1	0	0	1	0	0	4	0	0				

P19 problemi için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 19	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14																																																								
GA	Maliyet Limiti=130	0,989333	100	173	0	1	1	0	3	0	3	0	0	3	0	4	0	2	0	0	3	3	1	1	0	0	0	2	0	0	2	0	3	2	0	0	2	0	1	1	1	0	0	0	2	0	3	1	2	0	2	0	0	2	0	1	0	1	2	0	0	0	1	0	4	0								
	Maliyet Limiti=100	0,984	98	173	1	0	1	2	0	0	1	0	0	1	2	2	0	0	0	0	0	1	3	1	0	0	2	3	0	0	0	1	4	0	0	0	2	0	2	0	1	1	3	0	0	3	2	0	0	2	0	3	0	0	0	1	2	0	0	0														
	Maliyet Limiti=98	0,975333	97	173	4	0	3	2	0	1	0	1	0	0	0	0	4	3	0	2	0	0	2	0	0	0	0	1	1	3	0	0	2	0	0	0	2	3	0	2	1	0	1	0	0	0	0	2	3	2	2	0	0	2	0	0	0	3	0	0	1	1	2	1	1	0	0	0	4	0	0	1		
	Maliyet Limiti=97	0,976	97	173	4	0	3	0	4	0	2	0	1	0	4	0	0	0	2	0	1	0	1	3	0	0	2	2	2	0	4	4	0	0	3	0	0	3	2	0	0	1	1	4	0	0	0	2	0	2	0	0	3	0	3	1	0	0	0	0	1	1	0	0	1	0	4	0	0	0				
	Maliyet Limiti=96	0,981333	94	172	0	0	3	2	1	0	1	0	2	0	4	0	0	4	0	0	3	0	2	1	2	0	1	0	0	0	4	0	3	0	0	0	2	1	0	0	1	0	1	1	0	0	0	1	1	0	0	0	1	4	0	2	1	0	0	1	3	0	0	0	0	4	0							
	Maliyet Limiti=94	0,976667	93	173	0	0	2	2	0	1	0	0	0	1	0	0	2	0	3	0	1	1	0	1	3	0	0	0	2	0	0	0	3	4	0	0	0	2	3	0	1	0	1	3	0	0	0	3	3	0	2	2	0	0	2	0	0	2	0	0	0	1	1	0	0	0	3	3	0	0	1	4	0	
	Maliyet Limiti=93	0,980667	93	170	0	1	1	0	2	3	0	1	0	0	0	4	0	2	0	1	1	0	1	0	0	1	0	0	1	0	2	0	3	0	1	2	0	0	0	1	1	0	1	0	1	0	3	0	0	2	3	0	2	0	1	0	0	0	1	3	0	2	1	0	1	2	0	0	0	4	0	0	0	0
	Maliyet Limiti=92	0,967333	91	170	1	0	1	3	0	0	2	0	0	2	4	0	0	0	3	0	1	0	1	3	2	0	3	0	0	3	0	0	3	0	0	0	2	0	1	0	0	2	1	0	2	0	0	0	0	1	1	1	0	0	0	0	3	2	0	1	2	0	0	1	0	1	0	0	2	1	1	0	0	0

P22 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	P22 Problemi	Güvenilirlik	Maliyet	Ağırlık	Alternatifler																																																																						
					Alternatif1				Alternatif2				Alternatif3				Alternatif4				Alternatif5				Alternatif6				Alternatif7				Alternatif8				Alternatif9				Alternatif10				Alternatif11				Alternatif12				Alternatif13				Alternatif14																		
GA	Maliyet Limiti=130	0,991333	108	170	0	0	3	3	3	0	0	0	0	1	1	0	3	0	4	2	0	3	3	3	0	0	0	4	0	1	1	4	0	0	0	0	2	0	0	1	4	0	4	0	4	0	0	3	0	1	2	0	3	0	3	4	0	0	0	4	0	0	4	1	4	1	0	1	0	0	0	0	0	2	2
	Maliyet Limiti=108	0,988	107	170	4	0	1	0	0	0	1	1	0	0	2	4	2	0	0	0	3	1	0	1	0	1	0	1	0	4	0	1	0	0	0	4	1	0	0	1	0	0	2	0	0	4	0	3	2	0	0	2	2	0	4	4	0	2	0	2	0	0	2	0	2	0	0	2							
	Maliyet Limiti=107	0,986667	107	168	0	1	3	1	0	0	0	1	0	1	0	0	4	0	4	3	3	0	0	2	1	1	4	0	0	4	0	0	0	1	3	0	0	0	2	4	0	2	4	0	0	0	1	0	0	0	2	2	0	0	4	4	0	0	4	4	0	0	4	4	0										
	Maliyet Limiti=106	0,986	105	169	1	1	0	0	3	1	1	0	0	0	0	4	2	0	0	3	1	0	0	3	0	1	0	1	0	1	1	3	0	0	0	4	0	0	4	4	3	0	0	3	0	2	2	0	0	0	0	4	4	3	0	4	0	1	0	1	0	3	0	0	1	0	4	0							
	Maliyet Limiti=105	0,989333	105	170	3	0	3	0	3	0	1	0	1	0	0	0	4	2	0	3	2	0	3	0	0	1	0	4	0	1	0	4	0	3	1	0	0	0	4	0	4	4	0	4	0	0	3	1	0	2	0	0	4	0	0	1	0	4	4	4	0	2	0	1	0	0	2	0	4	0					
	Maliyet Limiti=100	0,987333	100	170	2	0	0	3	2	0	0	3	0	1	4	2	0	0	0	0	0	1	3	3	0	0	0	0	1	1	1	1	0	0	0	0	1	3	0	0	4	4	0	0	4	0	1	0	0	4	1	3	0	2	0	0	4	4	4	0	0	0	2	0	1	0	1	0	2	0					
	Maliyet Limiti=95	0,983333	94	169	0	0	3	3	1	1	0	0	0	1	0	2	0	3	0	1	0	0	1	2	0	1	1	0	0	1	0	0	4	0	0	2	1	0	4	0	0	1	0	4	0	2	2	0	2	0	0	0	1	0	4	4	4	0	0	0	1	2	0	4	0	0	0	2							
	Maliyet Limiti=90	0,982667	90	170	1	0	3	3	0	0	0	2	0	3	0	3	0	0	4	0	1	1	0	2	1	1	0	0	0	0	4	0	0	4	0	1	0	0	3	4	0	4	0	4	1	0	0	0	0	0	2	2	1	0	0	0	1	0	4	4	0	4	0	0	2	1	0	0	0	0	4				
	Maliyet Limiti=89	0,979333	89	170	0	0	1	3	1	0	0	1	0	3	0	3	0	3	0	3	1	3	1	0	0	0	0	0	1	1	0	4	0	0	1	0	0	1	1	0	0	4	4	0	0	0	1	2	0	0	2	0	0	4	0	0	1	4	1	0	0	1	0	2	0	2	0	0	2						
	Maliyet Limiti=88	0,98	88	169	0	1	2	0	1	0	0	0	2	3	0	0	2	2	4	2	1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	4	0	1	1	0	0	4	0	1	1	0	0	0	3	1	0	0	0	0	0	0	4	0	0	0	4	0											
	Maliyet Limiti=87	0,978	86	169	4	2	0	0	0	0	2	3	0	0	2	0	2	0	0	0	1	1	0	1	0	0	0	0	4	4	4	0	4	0	0	0	2	0	1	0	0	4	0	0	3	1	0	0	2	2	0	4	0	0	4	0	4	4	0	0	0	0	0	0	0	0									
	Maliyet Limiti=86	0,977333	86	170	2	0	0	2	0	0	1	1	0	0	3	3	0	0	3	1	0	1	1	0	2	1	0	0	0	0	0	1	0	4	0	1	0	0	2	4	4	0	4	0	0	1	0	0	0	2	0	0	2	2	4	0	0	1	0	4	4	0	0	1	0	0	0	4	4						
	Maliyet Limiti=85	0,976	85	170	1	0	1	0	2	0	0	0	1	3	4	0	0	3	3	0	1	2	1	0	0	0	4	0	1	0	0	4	4	0	1	0	2	0	4	0	4	0	0	1	0	0	1	1	4	0	0	0	4	0	4	4	4	0	0	0	0	1	1	0	0	0	4	0							
	Maliyet Limiti=84	0,976	84	169	0	0	0	3	2	1	1	0	0	0	3	3	0	0	3	0	0	1	3	1	1	0	1	0	4	4	3	0	0	0	0	0	3	0	1	0	4	4	0	4	4	0	0	0	0	2	2	0	0	4	0	0	1	4	0	0	0	0	4	0	4	1	0	0	0	1	0	0	0	0	4
	Maliyet Limiti=83	0,975333	83	168	0	3	0	2	0	1	0	1	0	0	3	3	0	3	0	1	1	0	0	1	0	0	1	1	0	3	0	4	0	0	0	0	0	1	2	4	4	4	0	0	4	0	4	0	0	0	0	0	2	0	2	0	4	4	0	0	4	0	4	0	0	0	2	0	0	2	0				
Maliyet Limiti=82	0,973333	82	170	1	1	0	1	0	0	0	0	1	2	3	3	0	0	3	1	1	0	0	1	2	0	1	0	0	0	4	4	0	0	2	1	0	0	4	4	0	0	0	1	0	0	3	0	2	0	0	4	0	4	4	0	4	0	1	0	1	0	1	4	0	0	0	0	0							
Maliyet Limiti=81	0,972	81	169	0	2	2	1	0	0	0	0	3	0	2	0	2	0	3	3	0	1	1	0	1	2	0	4	0	0	0	1	0	4	0	0	2	2	0	4	0	0	4	4	0	0	0	1	0	0	2	3	0	0	4	0	4	4	0	4	0	1	4	0	1	0	0	0	0	0	0	0				
Maliyet Limiti=80	0,972667	80	170	3	0	0	1	0	1	0	2	0	0	0	3	3	3	0	1	0	0	1	2	0	1	0	0	4	0	0	0	4	3	2	1	0	0	0	4	0	0	4	4	0	4	0	4	0	0	3	0	2	0	4	0	0	1	0	4	0	1	0	3	0	0	0	0	0	4	0					
Maliyet Limiti=79	0,966	79	169	1	0	0	2	0	0	3	0	0	2	0	0	0	3	3	0	0	2	1	1	0	0	0	4	1	3	0	3	0	0	0	3	0	0	1	0	4	4	0	4	4	0	0	0	2	0	0	3	0	0	4	1	0	0	0	4	4	0	0	3	1	0	0	0	4	0	0	0	0			

P23 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 23	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14																																																									
GA	Maliyet Limiti=130	0,983333	103	167	3	0	1	3	0	1	0	2	0	0	0	4	0	0	2	0	0	3	1	2	0	0	2	2	2	4	0	0	4	0	0	0	0	1	1	0	3	1	1	0	0	0	2	0	0	2	0	3	3	0	1	1	0	0	0	0	1	2	0	3	0	0	0	2	1	0	0	1	3	0	0
	Maliyet Limiti=103	0,984667	103	169	3	3	3	0	0	0	0	3	1	0	0	0	4	0	2	3	0	2	0	1	1	0	0	0	1	0	0	2	0	4	1	0	0	1	0	1	1	0	1	1	0	0	3	0	0	1	1	0	0	3	2	0	2	0	1	0	0	2	2	0	0	0	2	3	0						
	Maliyet Limiti=102	0,982	101	168	3	0	3	0	3	2	0	0	1	0	0	4	4	0	0	0	1	1	0	2	0	2	2	0	2	3	0	3	0	0	2	0	1	0	0	1	3	1	0	0	0	0	2	0	0	0	2	1	2	0	0	0	3	3	1	1	1	0	0	1	1	0	0	0	0	0	2	3			
	Maliyet Limiti=101	0,974	101	168	1	0	1	0	3	1	0	0	0	2	0	0	4	4	2	3	1	0	0	3	0	0	1	2	2	4	0	0	3	0	0	0	2	0	3	0	0	1	0	3	0	0	0	0	2	0	0	0	3	3	1	0	0	0	1	1	2	4	0	0	0	3	0	2	0	1	0	0	1	0	
	Maliyet Limiti=100	0,976	100	168	0	2	1	0	0	0	1	0	0	3	4	0	0	0	4	0	3	0	1	1	0	2	0	1	0	0	0	4	4	0	0	3	3	0	0	0	1	0	3	3	0	1	0	1	0	0	2	3	0	0	0	3	0	0	3	2	1	3	0	0	2	0	0	0	1	4	0	0	0		
	Maliyet Limiti=99	0,972667	99	169	0	2	0	0	1	0	0	1	0	2	1	0	4	0	2	0	3	0	1	3	0	0	0	2	3	0	0	2	0	4	0	1	2	0	0	0	0	1	3	0	0	1	0	3	0	2	0	0	3	0	1	0	0	2	0	2	0	2	0	2	0	1	0	3	1	0	0	0			
	Maliyet Limiti=98	0,97	98	169	3	0	0	3	1	0	0	3	0	3	4	0	0	3	0	0	2	1	1	0	0	3	0	2	0	4	0	0	0	3	0	1	0	3	0	0	3	1	3	0	3	0	0	0	3	1	2	0	0	0	0	0	3	1	1	2	2	0	0	2	1	0	0	0	0	4	0	0			
	Maliyet Limiti=97	0,978667	97	169	2	0	0	0	3	0	0	3	1	0	2	0	0	4	0	1	0	1	2	0	0	0	2	1	0	4	2	0	0	0	2	0	2	0	0	0	3	1	1	0	2	0	0	0	0	0	2	2	1	0	1	3	0	0	4	0	1	0	1	0	0	2	0	3	2	0	2	0	0		

P25 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 25	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13
GA	Maliyet Limiti=130	0,983333	110	167	0 0 3 3 1 0 1 0 1 0 4 4 0 4 0 3 0 0 1 1 0 1 0 0 2 0 0 4 2 0 0 0 0 1 3 1 0 3 0 1 0 2 0 0 0 0 3 0 3 2 0 2 0 0 1 2 0 0 1 2 2 0 2 0 0												
	Maliyet Limiti=110	0,972667	104	164	3 3 0 0 3 0 0 1 0 3 0 0 4 3 0 3 3 1 0 0 2 2 0 0 0 0 0 2 3 0 2 0 0 1 0 1 0 0 3 0 3 0 0 1 0 3 2 0 2 0 3 0 0 3 0 1 1 4 0 0 0 0 0 2 0												
	Maliyet Limiti=104	0,98	103	167	0 1 0 3 3 0 0 2 0 1 0 0 3 0 4 3 2 0 0 3 2 0 2 0 0 0 0 1 0 2 0 0 1 3 0 1 0 3 0 3 0 0 3 0 2 0 0 1 0 0 2 0 1 1 0 2 2 0 0 1 0												
	Maliyet Limiti=103	0,974667	102	165	0 1 2 0 0 2 0 1 0 0 4 4 0 0 4 0 2 3 0 3 0 0 2 0 2 0 2 0 0 0 4 2 0 1 3 0 0 1 1 0 0 1 2 0 0 0 0 3 0 2 1 0 0 1 1 0 0 1 3 0 0 2 0 2 0 0 0												
	Maliyet Limiti=102	0,977333	100	166	0 3 0 3 3 0 3 0 0 1 0 4 0 0 4 1 3 1 0 0 1 1 0 0 0 1 3 0 0 0 3 0 0 1 0 1 0 0 3 1 0 3 0 0 0 3 1 2 0 0 0 0 1 2 2 0 0 1 2 0 3 0 2 0												
	Maliyet Limiti=100	0,969333	98	165	3 0 0 2 0 0 1 0 1 0 0 3 0 2 0 3 0 1 0 1 2 2 0 0 0 0 0 1 3 0 0 0 1 0 0 3 3 0 3 1 0 0 0 0 0 2 2 1 0 0 3 0 0 1 1 0 1 2 0 0 3 2 2 0 0 0												
	Maliyet Limiti=98	0,972	98	167	0 3 3 0 2 0 0 2 2 0 0 0 2 4 3 3 3 0 0 3 0 0 0 2 3 4 0 0 0 2 0 0 0 2 2 0 3 0 2 0 0 0 0 2 2 0 3 0 0 1 0 0 1 0 1 2 1 0 0 3 0 0 0 2												
	Maliyet Limiti=97	0,972	96	167	2 1 0 1 0 1 0 0 0 1 0 0 0 4 2 0 0 3 3 2 0 0 1 1 0 0 1 3 0 0 0 2 0 0 0 2 2 0 0 0 3 0 0 0 3 1 2 0 3 0 0 0 0 0 2 1 1 2 2 0 0 1 0 3 0 0 0												

P26 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 26	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14																																								
GA	Maliyet Limiti=130	0,974667	106	166	4	3	0	0	3	1	0	1	0	0	0	0	2	2	4	0	3	1	0	3	0	0	3	2	0	4	0	0	0	4	0	0	0	1	1	0	0	3	0	1	0	0	0	0	0	0	4							
	Maliyet Limiti=106	0,982667	104	166	3	0	3	3	0	1	0	0	0	1	1	2	0	2	0	3	0	0	3	3	0	2	2	0	0	0	0	2	0	1	3	0	2	2	2	0	0	0	1	0	0	2	1	1	1	0	0	0	0	1	0	3	0	
	Maliyet Limiti=104	0,980667	100	166	0	3	4	3	0	0	0	0	1	1	4	0	0	2	0	3	1	0	0	3	0	0	1	0	2	0	4	0	0	1	2	1	0	0	0	0	1	1	0	0	2	1	1	2	0	3	0	0	4	2	0	0	0	
	Maliyet Limiti=100	0,977333	99	166	0	3	0	3	3	1	0	0	0	1	0	3	0	0	4	1	3	1	0	0	0	1	0	0	1	0	3	4	0	0	3	0	0	0	3	1	3	0	1	0	0	2	2	3	0	3	0	1	0	0	3	1	0	0
	Maliyet Limiti=99	0,969333	98	166	1	0	0	1	3	1	2	0	0	0	3	4	0	4	0	0	2	3	0	3	2	0	0	2	0	3	0	2	0	0	2	0	0	0	1	0	0	1	0	0	0	4	1	1	0	0	0	3	4	0	0	0	2	
	Maliyet Limiti=98	0,966	97	163	3	0	3	1	0	1	0	0	0	2	0	0	4	2	0	0	0	0	1	3	0	0	1	2	0	1	0	0	1	0	3	0	1	0	0	0	2	0	0	0	0	2	0	4	0	0	0							
	Maliyet Limiti=97	0,982	96	164	3	0	0	3	3	2	0	0	1	0	0	3	4	0	2	0	0	3	0	3	0	1	0	2	0	0	2	0	4	0	0	0	1	1	0	1	0	0	1	1	2	0	0	1	1	0	0	0	1	0	0	0	3	3
	Maliyet Limiti=96	0,976667	95	166	0	2	3	0	0	0	0	1	1	0	0	1	2	0	3	0	1	3	0	1	0	2	0	0	3	2	0	1	0	0	0	1	3	0	1	1	0	1	0	0	0	2	0	0	0	0	3	0	0	3				
	Maliyet Limiti=95	0,965333	94	165	1	2	0	0	0	0	2	0	1	0	2	1	0	1	0	2	3	2	0	0	2	0	0	0	2	0	0	4	0	2	0	0	1	0	3	1	1	1	0	0	2	0	0	0	0	3	0	0	4	0	0			
	Maliyet Limiti=94	0,98	94	166	1	3	2	0	0	1	1	0	0	0	0	4	0	4	0	1	0	1	3	0	0	0	2	0	2	4	4	0	0	0	2	0	0	0	2	0	1	0	1	1	1	0	1	0	0	3	0	0	0	3	2	0		

P32 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 32	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14																																																								
GA	Maliyet Limiti=130	0,981333	103	160	3	1	0	0	3	0	1	0	1	0	0	0	4	4	0	3	0	0	1	3	0	0	0	1	2	2	2	0	0	0	0	1	0	0	2	1	0	1	1	0	0	0	2	0	0	3	2	0	3	0	0	0	1	1	0	0	1	0	4	1	2	0	2	0	0	0	0	0	2	1
	Maliyet Limiti=103	0,966667	102	159	2	3	2	0	3	0	0	1	1	0	4	4	0	0	0	3	0	3	0	2	0	0	0	1	1	0	0	2	2	0	2	0	0	0	1	0	0	3	3	0	0	1	0	2	0	0	0	2	0	3	1	2	0	0	0	1	0	0	0	4	0									
	Maliyet Limiti=102	0,969333	98	158	0	3	3	0	2	0	1	0	1	0	4	0	4	0	0	3	1	0	0	1	0	3	0	1	0	0	4	0	0	2	0	1	0	0	1	0	2	0	1	1	0	0	2	0	0	3	3	0	0	0	0	1	2	1	2	0	2	0	0	0	2	0	0	4	0	0	1			
	Maliyet Limiti=98	0,968667	98	160	3	0	2	0	0	0	1	0	2	0	0	4	2	0	0	1	0	3	0	2	0	2	2	0	0	2	0	0	0	4	0	3	0	0	2	2	0	0	0	3	0	3	0	0	0	1	1	2	1	0	0	2	0	0	0	2	3	0	0	0										
	Maliyet Limiti=97	0,969333	91	160	1	0	2	0	0	0	0	1	3	0	2	0	4	0	0	0	3	2	0	1	0	0	2	2	0	0	0	0	4	2	1	3	0	0	0	0	1	1	2	0	0	0	2	0	1	2	3	0	0	0	1	0	1	3																
	Maliyet Limiti=91	0,962667	91	156	3	0	0	3	1	0	3	1	0	0	0	0	0	2	4	0	0	1	3	0	1	0	0	2	0	0	0	0	4	2	2	0	2	0	0	1	3	0	1	0	0	0	1	0	3	3	0	0	2	1	0	0	1	0	2	0	2	0	0	0	1	1	0	0	4	0				
	Maliyet Limiti=90	0,957333	89	159	1	2	0	2	0	0	0	1	0	3	4	0	2	0	0	0	0	1	1	3	0	2	0	0	2	1	0	2	0	0	2	0	0	0	3	1	0	2	0	0	0	0	0	2	0	0	2	2	0	0	0	1	1	0	0	0	4	0	2	3	0	1	0	0	0	0	2	3		
	Maliyet Limiti=89	0,958	89	160	3	3	0	0	0	0	2	0	0	2	4	0	1	0	0	3	0	2	1	0	0	1	2	0	0	0	0	2	4	0	1	0	0	1	0	1	0	1	0	0	0	0	0	4	3	0	3	1	0	0	0	1	0	1	0	1	0	1	0	2	1	0	0	0	3	1	0	3	0	0
	Maliyet Limiti=88	0,953333	87	157	0	0	1	4	0	3	0	3	0	0	0	4	0	0	3	3	1	0	2	0	2	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4						

P33 için tamir edilebilir bileşenlerin kullanımının sistemin kurulum maliyetine etkisi

	Pr 33	Güvenilirlik	Maliyet	Ağırlık	Altsistem1	Altsistem2	Altsistem3	Altsistem4	Altsistem5	Altsistem6	Altsistem7	Altsistem8	Altsistem9	Altsistem10	Altsistem11	Altsistem12	Altsistem13	Altsistem14
GA	Maliyet Limiti=130	0,970667	102	159	3 0 3 0 0	1 3 0 0 0	2 0 0 0 4	0 3 0 3 1	0 0 0 2 2	3 0 0 0 2	3 0 0 1 0	1 0 0 1 3	0 2 0 0 0	3 0 0 0 3	0 0 0 2 2	2 0 4 0 1	1 0 0 1 0	0 3 0 0 3
	Maliyet Limiti=102	0,972	96	159	0 0 0 2 2	0 0 1 0 1	0 0 2 2 0	3 0 0 3 3	0 2 0 0 2	0 0 0 1 3	0 3 0 0 1	1 0 1 0 1	2 0 0 0 0	0 2 0 1 2	0 2 0 0 3	2 1 0 1 0	2 0 0 0 1	0 0 4 0 0
	Maliyet Limiti=96	0,974667	96	159	0 1 0 0 4	0 0 1 1 0	4 0 0 4 0	1 0 1 0 3	0 0 2 0 2	4 1 0 0 0	0 0 1 1 0	1 0 0 1 1	0 1 3 0 0	0 0 0 2 2	2 0 3 0 0	0 0 1 2 1	0 2 0 0 1	3 0 0 1 0
	Maliyet Limiti=95	0,97	95	159	0 2 3 0 0	1 0 0 1 0	0 0 4 2 0	1 0 0 1 2	0 0 2 2 0	0 3 0 0 3	0 2 0 2 0	0 0 3 0 3	0 0 3 3 0	2 0 3 0 0	0 0 0 3 1	1 0 0 2 1	2 1 0 0 0	0 0 0 4 0
	Maliyet Limiti=94	0,966667	94	159	0 1 0 1 0	0 0 1 2 0	4 0 0 2 0	1 3 0 0 2	0 0 2 0 1	4 0 0 0 4	0 1 0 0 2	1 0 1 3 0	2 0 0 0 0	3 0 0 0 2	0 2 0 1 0	2 0 0 2 1	0 0 1 2 0	0 3 0 0 3
	Maliyet Limiti=93	0,964	93	159	1 0 1 1 0	0 0 0 1 1	0 4 0 2 0	0 2 3 1 0	2 0 0 2 0	0 0 0 4 4	0 0 2 2 0	0 1 1 1 0	0 0 0 3 1	0 0 3 0 1	0 0 1 0 2	2 0 4 0 0	0 0 0 0 2	0 3 0 2 0
	Maliyet Limiti=92	0,966	92	159	2 2 0 0 0	0 0 1 1 0	4 0 0 2 0	1 0 0 1 1	2 3 0 0 0	0 0 2 0 0	4 1 0 0 1	0 1 0 3 2	0 0 0 0 2	2 0 0 0 3	1 0 0 2 0	1 2 0 0 2	0 0 2 1 0	0 4 0 0 0
	Maliyet Limiti=91	0,972667	91	159	0 2 0 3 0	3 1 0 0 0	2 0 1 0 4	1 3 0 3 0	0 0 0 2 2	4 4 0 0 0	1 3 0 0 0	0 1 1 0 1	0 0 0 0 2	0 2 0 3 0	0 1 0 0 1	1 1 1 0 0	0 0 2 1 0	2 0 0 0 1
	Maliyet Limiti=90	0,972	89	158	0 2 3 0 0	0 3 0 1 0	0 0 0 4 2	0 1 0 1 2	0 1 0 2 0	2 0 0 4 0	0 0 3 1 0	1 0 1 0 1	0 0 2 0 0	0 0 0 3 2	0 0 2 2 0	0 1 1 1 0	0 3 0 0 1	0 0 3 1 0
	Maliyet Limiti=89	0,964	89	156	3 0 2 1 0	2 0 0 0 2	0 0 4 4 0	3 1 0 0 1	0 0 2 0 2	0 0 0 4 4	0 0 1 1 0	0 0 1 3 0	0 2 0 0 0	2 0 0 3 0	0 0 0 1 2	0 0 2 1 1	1 0 0 1 0	0 2 3 0 0
	Maliyet Limiti=88	0,949333	87	159	0 1 0 0 4	0 3 0 1 0	0 3 0 0 2	0 3 1 0 3	0 0 3 0 1	3 0 0 0 4	1 0 0 1 0	0 2 0 1 0	3 0 0 1 0	0 2 0 1 0	0 0 1 0 0	0 2 0 4 0	0 0 1 2 0	0 0 0 0 4

EK 3: P_{kd2} ve P_{kd3} problemleri için oluşturulan değişken “k” değerleri

P_{kd2} problemine ait değişken “k” değerleri

t		Altsistemlere ait Değişken "k" Değerleri													
Ay	Hafta	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	2	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	3	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	4	7	7	7	7	7	7	7	7	7	7	7	7	7	7
2	5	4	3	7	3	7	3	4	3	7	3	7	3	4	3
2	6	4	3	7	3	7	3	4	3	7	3	7	3	4	3
2	7	4	3	7	3	7	3	4	3	7	3	7	3	4	3
2	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7
2	9	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	10	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	11	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	12	7	7	7	7	7	7	7	7	7	7	7	7	7	7
3	13	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	14	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	15	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	16	7	7	7	7	7	7	7	7	7	7	7	7	7	7
4	17	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	18	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	19	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	20	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	21	7	7	7	7	7	7	7	7	7	7	7	7	7	7
5	22	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	23	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	24	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	25	7	7	7	7	7	7	7	7	7	7	7	7	7	7
6	26	7	7	7	7	7	7	7	7	7	7	7	7	7	7
7	27	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	28	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	29	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	30	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	31	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	32	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	33	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	34	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	35	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	36	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	37	7	7	7	7	7	7	7	7	7	7	7	7	7	7
9	38	7	7	7	7	7	7	7	7	7	7	7	7	7	7

9	39	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	40	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	41	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	42	7	7	7	7	7	7	7	7	7	7	7	7	7	7
10	43	7	7	7	7	7	7	7	7	7	7	7	7	7	7
11	44	4	2	6	2	6	2	4	2	6	2	6	2	4	2
11	45	4	2	6	2	6	2	4	2	6	2	6	2	4	2
11	46	4	2	6	2	6	2	4	2	6	2	6	2	4	2
11	47	4	2	6	2	6	2	4	2	6	2	6	2	4	2
12	48	4	2	6	2	6	2	4	2	6	2	6	2	4	2
12	49	4	2	6	2	6	2	4	2	6	2	6	2	4	2
12	50	4	2	6	2	6	2	4	2	6	2	6	2	4	2
12	51	4	2	6	2	6	2	4	2	6	2	6	2	4	2
12	52	4	2	6	2	6	2	4	2	6	2	6	2	4	2

P_{kd3} problemine ait değişken “k” değerleri

t		Altsistemlere ait Değişken "k" Değerleri													
Ay	Hafta	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	6	6	6	6	6	6	6	6	6	6	6	6	6	6
1	2	6	6	6	6	6	6	6	6	6	6	6	6	6	6
1	3	6	6	6	6	6	6	6	6	6	6	6	6	6	6
1	4	6	6	6	6	6	6	6	6	6	6	6	6	6	6
2	5	5	4	6	4	6	4	5	4	6	4	6	4	5	4
2	6	5	4	6	4	6	4	5	4	6	4	6	4	5	4
2	7	5	4	6	4	6	4	5	4	6	4	6	4	5	4
2	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6
2	9	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	10	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	11	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	12	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	13	6	6	6	6	6	6	6	6	6	6	6	6	6	6
4	14	6	6	6	6	6	6	6	6	6	6	6	6	6	6
4	15	6	6	6	6	6	6	6	6	6	6	6	6	6	6
4	16	6	6	6	6	6	6	6	6	6	6	6	6	6	6
4	17	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	18	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	19	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	20	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	21	6	6	6	6	6	6	6	6	6	6	6	6	6	6
5	22	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	23	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	24	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	25	6	6	6	6	6	6	6	6	6	6	6	6	6	6
6	26	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	27	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	28	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	29	5	5	5	5	5	5	5	5	5	5	5	5	5	5
7	30	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	31	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	32	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	33	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	34	6	6	6	6	6	6	6	6	6	6	6	6	6	6
8	35	6	6	6	6	6	6	6	6	6	6	6	6	6	6
9	36	6	6	6	6	6	6	6	6	6	6	6	6	6	6
9	37	6	6	6	6	6	6	6	6	6	6	6	6	6	6
9	38	6	6	6	6	6	6	6	6	6	6	6	6	6	6
9	39	6	6	6	6	6	6	6	6	6	6	6	6	6	6

10	40	6	6	6	6	6	6	6	6	6	6	6	6	6	6
10	41	6	6	6	6	6	6	6	6	6	6	6	6	6	6
10	42	6	6	6	6	6	6	6	6	6	6	6	6	6	6
10	43	6	6	6	6	6	6	6	6	6	6	6	6	6	6
11	44	5	2	6	2	6	2	5	2	6	2	6	2	5	2
11	45	5	2	6	2	6	2	5	2	6	2	6	2	5	2
11	46	5	2	6	2	6	2	5	2	6	2	6	2	5	2
11	47	5	2	6	2	6	2	5	2	6	2	6	2	5	2
12	48	5	2	6	2	6	2	5	2	6	2	6	2	5	2
12	49	5	2	6	2	6	2	5	2	6	2	6	2	5	2
12	50	5	2	6	2	6	2	5	2	6	2	6	2	5	2
12	51	5	2	6	2	6	2	5	2	6	2	6	2	5	2
12	52	5	2	6	2	6	2	5	2	6	2	6	2	5	2