



T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



YÜKSEK LİSANS TEZİ

TERSANELERDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU ve BİR
TERSANE UYGULAMASI

Pamela ATES

DANIŞMAN
Dr. Öğr. Üyesi Güldem ELMAS

Deniz Ulaştırma ve İşletme Mühendisliği Anabilim Dalı

Deniz Ulaştırma ve İşletme Mühendisliği (F.B.E), Tezli Yüksek Lisans Programı

Haziran, 2024

TEZ KABUL VE ONAYI

Pamela ATES tarafından, **Dr. Öğr. Üyesi Güldem ELMAS** danışmanlığında hazırlanan "**TERSANELERDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU ve BİR TERSANE UYGULAMASI**" başlıklı bu çalışma, jürimiz tarafından **.06.2024** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

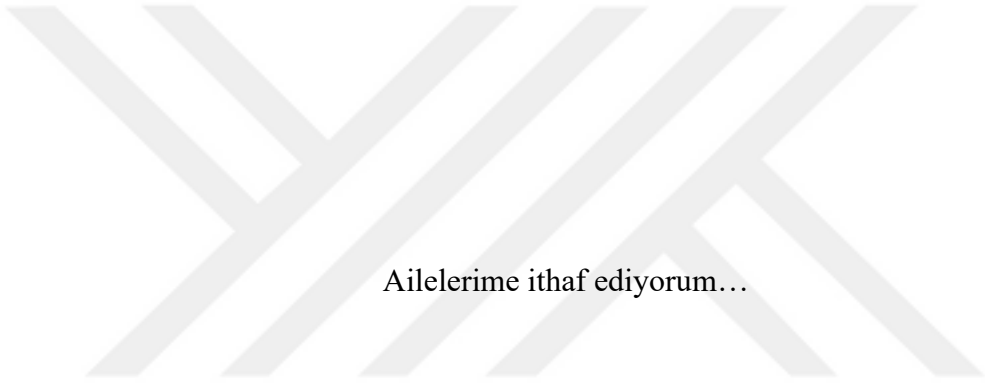
Tez Jürisi

	İmza	Sonuç
DANIŞMAN	Dr. Öğr. Üyesi Güldem ELMAS İstanbul Cerrahpaşa Üniversitesi Anabilim Dalı	<input checked="" type="checkbox"/> Kabul <input type="checkbox"/> Ret
ÜYE	Prof. Dr. Xxxx XXXX Üniversitesi Anabilim Dalı	<input type="checkbox"/> Kabul <input type="checkbox"/> Ret
ÜYE	Prof. Dr. Xxxx XXXX Üniversitesi Anabilim Dalı	<input type="checkbox"/> Kabul <input type="checkbox"/> Ret
ÜYE	Prof. Dr. Xxxx XXXX Üniversitesi Anabilim Dalı	<input type="checkbox"/> Kabul <input type="checkbox"/> Ret
ÜYE	Prof. Dr. Xxxx XXXX Üniversitesi Anabilim Dalı	<input type="checkbox"/> Kabul <input type="checkbox"/> Ret

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve bilimsel etik kuralları içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını ve her türlü hukuki sorumluluğu aldığımı kabul ederim.

Pamela ATES



Ailelerime ithaf ediyorum...

BÜTÇE DESTEKLERİ

TERSANELERDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU ve BİR TERSANE UYGULAMASI

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.



TEŐEKKÜR

Aileme, danıőmanıma, hocalarıma, denizcilik meslektaőlarıma ve Hicri Ercili tersane ailesine (özellikle Barıő Samut Beye ve Esra Nur İőőan Hanıma) teőekkür etmek isterim.

Haziran 2024

Pamela ATES



İÇİNDEKİLER

TEZ KABUL VE ONAYI	ii
BEYAN	iii
BÜTÇE DESTEKLERİ	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ŞEKİL LİSTESİ	x
TABLO LİSTESİ	xi
SİMGE VE KISALTMA LİSTESİ	xii
ÖZET	xiv
ABSTRACT	xv
1. GİRİŞ	1
1.1. GEMİ İNŞAA SANAYİ.....	2
1.2. TÜRKİYEDE VE DÜNYADA TERSANECİLİK	4
1.3. TERSANEDE YAPILAN FAALİYETLER	5
1.4. TERSANEDE GEMİ İNŞAASI	7
1.5. TERSANEDE YAT İNŞAASI	9
1.6. TERSANEDE BAKIM ONARIM İŞLEMLERİ.....	10
1.7. GEMİ İNŞAATINDA TEDARİK ZİNCİR YÖNETİMİ.....	12
1.8. TERSANEDE MALZEME TAŞIMACILIĞI	14
1.9. DOK VE ALAN OPTİMİZASYONU	14
1.10. GEMİ İNŞAATINDA İŞ GÜCÜ MAALİYETLER	16
1.11. DENİZ YOLU LOJİSTİĞİ	17
2. KAVRAMSAL ÇERÇEVE	19
2.1. GENETİK ALGORİTMA	19
2.2. TEMEL KAVRAMLAR	20
2.2.1. Genler	20
2.2.2. Kromozomlar	20
2.2.3. Popülasyonlar	21

2.2.4. Uygunluk Fonksiyonu	21
2.2.5. Elitizm	22
2.3. GENETİK ALGORİTMANIN BASİT BİR ŞEMASI	23
2.4. GENETİK ALGORİTMANIN ÇÖZÜM TÜRLERİ.....	25
2.4.1. Kodlama Türleri	25
2.4.2. Başlangıç popülasyonun Oluşması.....	26
2.4.3. Değerlendirme	26
2.4.4. Cross Over İşlemi.....	27
2.4.5. Cross Over Türleri.....	27
2.4.6. Mutasyon İşlemleri.....	28
3. YÖNTEM	29
3.1. GENETİK ALGORİTMA UYGULAMA ALANLARI	29
3.2. PYTHON DİLİ	30
3.2.1. Genetik Algoritma Python Kodu	31
3.2.2. Python’da Genetik Algoritma ile Knapsack Problemi.....	33
3.3. TERSANEDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU	36
3.3.1. RFID, Barkodlama ve Blok Zinciri.....	39
3.3.2. Endüstriyel Robotlar/Cobotlar	41
3.3.3. 3D Metal Printing.....	43
3.3.4. Automated Guided Vehicle (AGV).....	43
3.3.5. Digital Twin	44
3.3.6. Virtual Reality (Virtual Reality)	45
3.3.7. Eğitim Uygulamaları	45
3.3.8. Darboğazları / Şıkışma noktaları İncelenmesi	46
4. BULGULAR	47
4.1. HİCRİ ERCİLİ TERSANESİ	47
4.2. YENİ İNŞAA PROJELERİ	49
4.3. GENETİK ALGORİTMA (KNAPSACK METODU) İLE TERSANENİN DİJİTALEŞMESİ VE GELİŞTİRİLMESİ	52
4.3.1. Birinci Aşama.....	52
4.3.2. İkinci Aşama	62
5. TARTIŞMA.....	72
6. SONUÇ VE ÖNERİLER	75
KAYNAKLAR.....	77
EKLER	82

İNTİHAL RAPORU İLK SAYFASI	83
ETİK KURUL İZİN YAZISI	84
KURUM İZİN YAZILARI.....	85
PATENT HAKKI İZİNİ	86
ÖZGEÇMİŞ	87



ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: Genetik Algoritmanın basit şeması	24
Şekil 3.1: AGV'lerin Maaliyeti	43
Şekil 3.2: AGV'lerin Maaliyeti	44
Şekil 3.3: Digital Twinning Uygulama Maaliyetleri	45
Şekil 4.1: Hicri Ercili Tersanesinde bulunan tesisler	47
Şekil 4.2: Hicri Ercili Tersanenin Sertifikaları	48
Şekil 4.3: Çözümlerin Python'da Yazılması	52
Şekil 4.4: Parametrelerin Kodlanması	53
Şekil 4.5: Başlangıç Popülasyonu ve Uygunluk Fonksiyonunun Python Dilinde Kodu.....	54
Şekil 4.6: Fonksiyon Seçimi, Crossover Fonksiyonu ve Mutasyon Fonksiyonu.....	55
Şekil 4.7: Genetik Algoritmanın Uygulanması	58
Şekil 4.8: Ana İşlev Fonksiyonunun Kodu	60
Şekil 4.9: Kombinasyon Kodu Eklenmesi.....	63
Şekil 4.10: En iyi üç çözüm için Genetik Algoritma	68
Şekil 4.11: En iyi üç çözüm için Ana Fonksiyon.....	69

TABLO LİSTESİ

	Sayfa No
Tablo 2.1: Genler	20
Tablo 2.2: Kromozom	21
Tablo 2.3 Popülasyon	21
Tablo 2.4: İkili Kodlama	25
Tablo 2.5: Cross Over	27
Tablo 4.1: Olabilecek Çözümlerin Yazılması	51
Tablo 4.2: Olabilecek Çözümlerin Değerleri ve Maaliyeti	52
Tablo 4.3: Kombinasyonlar	66-67

SİMGE VE KISALTMA LİSTESİ

Simgeler Açıklamalar

P : Permütasyon

Kisaltmalar Açıklamalar

3D : 3 Dimensional (Üç Boyutlu)

ABD : Amerika Birleşmiş Milletleri

ADR : European Agreement Concerning the International Carriage of Dangerous Goods by Road (Tehlikeli Malların Karayolu ile Uluslararası Taşımacılığına ilişkin Avrupa Anlaşması)

AGV : Automated Guided Vehicle (Otomatik Yönlendirilen Araçlar)

AIS : Automatic Identification System (Otomatik Tanımlama Sistemi)

CPM : Critical Path Method (Kritik Yol Metodu)

ECDIS : Electronic Chart Display and Information System (Elektronik Harita Gösterim ve Bilgi Sistemi)

ERP : Enterprise Resource Planning (Kurumsal Kaynak Planlama)

GA : Genetik Algoritma

IMO : International Maritime Organization (Dünya Denizcilik Örgütü)

ISSC : International Ship and Offshore Structures Congress (Uluslararası Gemi ve Açık Deniz Yapıları Kongresi)

LNG : Liquefied Natural Gas (Sıvılaştırılmış Doğal Gaz)

OECD : Organization for Economic Cooperation and Development (Ekonomi İşbirliği ve Kalkınma Örgütü)

OPEC : Organization of the Petroleum Exporting Countries (Petrol İhraç Eden Ülkeler Örgütü)

RFID : Radio-Frequency Identification (Radyo Frekansı Tanımlama)

TSE : Türk Standart Enstitüsü

UNCTAD : United Nations Conference on Trade and Development (Birleşmiş Milletler Ticaret ve Kalkınma Konferansı)

VDR : Voyage Data Recorder (Gemi Seyir Kayıt Cihazı)

VR : Virtual Reality (Sanal Gerçeklik)



ÖZET

[YÜKSEK LİSANS TEZİ]

[TERSANELERDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU ve BİR TERSANE UYGULAMASI]

[Pamela ATES]

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Deniz Ulaştırma ve İşletme Mühendisliği Anabilim Dalı

Deniz Ulaştırma ve İşletme Mühendisliği (F.B.E), Tezli Yüksek Lisans Programı

[Danışman : Dr. Öğr. Üyesi Güldem ELMAS]

[Tersanelerde lojistik akışları optimize etmek, düzen optimizasyonu, teknoloji entegrasyonu, süreç standartlaştırma ve iletişimin geliştirilmesini içermektedir. Bu uygulamaları hayata geçirerek, tersaneler operasyonları geliştirebilir, maliyetleri azaltabilir ve verimliliği artırabilir.]

Haziran 2024 , [.....] sayfa.

Anahtar kelimeler: [Tersane,teknoloji,optimizasyon]

ABSTRACT

[M.Sc. THESIS]

**[OPTIMIZATION OF LOGISTICS FLOWS IN SHIPYARDS AND AN APPLICATION
IN ONE SHIPYARD]**

[Pamela ATES]

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Marine Transportation Management Engineering

Maritime Transportation&Management Engineering

[Supervisor : Assist. Prof. Dr. Güldem ELMAS]

[Optimizing logistic flows in a shipyard involves layout optimization, technology integration, process standardization, and enhanced communication. By implementing these, shipyards can improve operations, reduce costs, and enhance efficiency.]

July 2024, [.....] pages.

Keywords: [Shipyards,technology,optimization]

1. GİRİŞ

Storch'un ve diğerlerin (1995) belirttiği gibi gemi inşası, gemilerin inşa edilmesidir ve tersane, gemilerin inşa edildiği yerdir. Gemi inşası, temel yapı malzemelerinin yanı sıra çeşitli imal edilmiş bileşenler kullanan bir inşaat endüstrisidir. Bu nedenle süreç, hem inşaat hem de imalatın birçok özelliğine sahiptir. Gemi inşası, gerekli tesislerin mevcut olduğu belirli bir lokasyonda, çeşitli becerilere sahip birçok işçinin, belirlenmiş bir organizasyon yapısı içinde çalışmasını gerektirmektedir.

Tekrardan Storch ve diğerlerine (1995) göre, gemi inşası, müşterilere ürünler üreten bir endüstridir. Çoğu durumda, ürün sipariş üzerine inşa edilmektedir ve alıcının özel gereksinimlerine göre özelleştirilmektedir. Bu, benzer bir gemi serisinin inşa edildiği durumlarda bile geçerlidir. Tüm süreç, ilgili müşteriye bağlı olarak biraz farklılık gösterebilmektedir, ancak genellikle bir dizi belirli aşamaları içermektedir.

Storch ve diğerlerine (1995) göre, bu aşamalar, bireysel bileşenlerin ilk üretiminden, bu elemanların alt montajlar ve modüller halinde birleştirilmesine ve nihai gemi montajına kadar uzanmaktadır. Tersaneler, gemi inşası ve denize indirilmesi için gerekli altyapıyı sağlayan kızaklar, kuru havuzlar ve büyük vinçler gibi özel üretim tesisleriyle donatılmıştır. Üretimin ileri aşamalarında, büyük ve ağır ara ürünler, üniteler, bloklar ve modüller taşınıp monte edilerek geminin nihai yapısı oluşturulmaktadır. Bu süreçlerin yönetimi, gemi inşaat sanayinin karmaşıklığını ve gereksinimlerini ortaya koymaktadır.

Gemi inşa endüstrisi zaman alıcı bir sektördür ve sadece tek başında yürememektedir. Aynı zamanda başka endüstrilerle ve yan sanayi ile bağlıdır. Gemi inşa sektörü, ekonomik refahı yüksek ülkelerden, doğal kaynaklar gücü, yüksek iş gücü, gemi satın alma şirketlerden ve iç rekabetten gibi benzer faktörlerden belirlenmektedir. Deniz ekonomisi, global ticarete elzem bir rol oynamaktadır. Gemi inşası, balıkçılık, deniz kaynaklarının kullanımı, deniz taşımacılığı, limanlar ve kıyı turizmi gibi sektörler, bu ekonominin temelini oluşturmaktadır. Petrol, gaz, mineraller ve yenilenebilir enerji de dahil olmak üzere deniz kaynaklarının kullanımı, enerji taleplerini karşılamak ve ekonomik kalkınmayı teşvik etmek için önemlidir. Ayrıca, limanlar da tedarik zincirinde bir düğüm oluşturmaktadır. Son olarak, sigorta, brokerlik, hukuk, acente ve eğitim gibi denizcilik hizmetleri, endüstrinin işleyişi için temel destek sağlamaktadır. Bu

sektörlerin bir araya gelmesiyle sürdürülebilir bir deniz ekonomisi oluşmaktadır ve sürekli büyüme sağlanmaktadır.

Lee ve diğ. (2014) belirttiği gibi gemi inşa endüstrisinde, lojistik zincirinin işlevselliği iyi iş sonuçlarına ulaşmada kilit rol oynamaktadır. Daha iyi iş kararları almak ve işletmeye potansiyel olarak zarar verebilecek hataları en aza indirmek amacıyla yöneticiler, çeşitli destek araçları kullanmaktadır. Bilgi ve iletişim teknolojisinin gelişimi, kurumsal yönetim yöntemlerinde büyük bir ilerleme sağlamıştır.

Gemi inşaatında müşteri, maddi destek, teknik özellikler, kalite ve teslimat takvimi konularında taleplerde bulunur ve tedarikçilerle iş birliği yapmaktadır. Amaçları maliyeti düşürmek ve kaliteyi korumaktır. Başarı için tedarik zincirinin optimize edilmesi gerekmektedir. Bu optimizasyon, malzeme temini, üretim, kalite güvencesi ve teslimat süreçlerinin iyileştirilmesini içermektedir. Böylece gemi inşaat endüstrisi, müşteri beklentilerine uygun, zamanında ve maliyet açısından kaliteli gemiler sunabilmektedir.

Tedarik zinciri ve üretim operasyonlarının etkinliği performans göstergesidir. Gemi inşaatında, müşteri memnuniyeti malzeme kalitesine bağlıdır. Tedarik durumu ve teslimat lojistiği, planlama esnekliğini belirlemektedir. İş lojistiği verimliliğini artırmak için stratejik planlama, tedarik zinciri, tedarik durumu ve üretim operasyonlarının entegre edilmesini gerektirmektedir.

1.1. GEMİ İNŞAA SANAYİ

Colton ve Huntzinger (2002) tarafından dediği gibi gemiler, II. Dünya Savaşı'nda özellikle kuvvetlerin ve ekipmanların Atlantik üzerinden taşınmasında önemli bir rol oynamıştır. Savaştan sonra, ABD ve Birleşik Krallık büyük deniz ve ticaret filolarına ve güçlü gemi yapım endüstrilerine sahipti. 1956 Süveyş Kanalı'nın kapanması, petrol güzergahlarının Ümit Burnu'ndan dolaşması nedeniyle tanker sıkıntısına neden olarak tanker talebini artırmıştır. 1957'den 1973'e kadar gemi yapım endüstrisi, daha büyük tankernlere doğru bir kayma ve Japonya'nın önde gelen bir gemi yapımcısı olarak ortaya çıkmasıyla önemli değişiklikler yaşamıştır. 1973 OPEC petrol ambargosu, petrol talebinin çökmesine ve gemi yapımında bir düşüşe neden olmuştu. Bu düşüş 1987'de dibe vurmuştu ve 1988'de siparişlerde, 1990'da ise teslimatlarda bir toparlanma başlamıştır (Colton ve Huntzinger, 2002).

UNCTAD verilerine göre deniz ticareti, küresel ticaret hacminin %80'den fazlasını deniz yollarıyla gerçekleştirildiğini, ticaretin yanı sıra güvenlik ve savunma alanlarında gemi inşasının kritik rolünü vurgulayarak büyük bir ekonomik öneme sahiptir. Denizcilik endüstrisinin geleceği, ekonomik koşullar ve düzenleyici çerçevelerden etkilenmektedir. Dönemsel ekonomik durgunluklar yeni gemi inşaat projelerine olan talebi düşürürken, çevresel ve enerji verimliliği standartlarını içeren düzenlemeler yeni inşaat talebini artırabilir. Tahminler, artan deniz ticareti, devam eden ekonomik ilerlemeler, artan enerji tüketimi ve çevre dostu gemi ve hizmetlere olan talep gibi faktörlerle desteklenen küresel gemi inşaat pazarının yakın gelecekte genişleyeceğini öngörmektedir. Sonuç olarak, tersaneler genellikle kısıtlı zaman dilimlerinde gemiler inşa etmeye ve mevcut filoların kapasitelerini artırmaya ve eski gemileri yenileyerek yük taşıma yeteneklerini güçlendirmeye odaklanmaktadır.

Bazı tersaneler üretim süreçlerini optimize etmek için robotik otomasyonu benimsemelerine rağmen, her geminin inşasının karmaşıklığı nedeniyle kısmi otomasyonun uygulanmasında belirli zorluklarla karşılaşmaktadır. Bu nedenle, işgücünün yetkinliği ve verimliliği, gemi inşası faaliyetlerinin kalitesini ve maliyet etkinliğini sağlamak için hayati öneme sahiptir.

Gemi inşaa faaliyetleri tersanelerde hem kapalı hem de açık alanlarda gerçekleştirilmektedir. Kapalı alanlarda kesme, şekillendirme, montaj ve büyük imalat gibi işler yapılırken, açık hava aşamaları teçhizatlandırma, boyama, ön montaj, son montaj, test ve gemi teslimatını içermektedir. Kaynak, gemi inşaatında baskın üretim tekniği olarak öne çıkar ve belirli üretim gereksinimlerini karşılamak için özel prosesler ve kaynak yöntemleri kullanılmaktadır.

Gemi inşaatında farklı amaçlar için çelik dereceleri, demir dışı alaşımlar ve alüminyum gibi çeşitli malzemeler kaynaklanmaktadır. Kaynak tüketim malzemelerinin ve ekipmanlarının etkinliği, üretim ortamlarında yüksek kaliteli kaynakların ve işletme verimliliğinin sağlanmasında hayati öneme sahiptir. ECORYS (2009) verilerine göre Asya tersanelerinin Avrupa rakiplerine üstünlüğü iki ana faktöre dayanmaktadır. Birincisi, işgücü maliyetleri Avrupa, Japonya ve Güney Kore'de nispeten benzer olsa da Çin'de önemli ölçüde daha yüksektir. İkincisi, gemi inşaatının ana hammadde olan çelik fiyatları, küresel çelik üretimi ve tüketiminden etkilenmektedir ve bu alanda Asya ülkeleri, özellikle Çin, baskındır. Bu durum, Avrupa tersanelerini daha yüksek hammadde maliyetleriyle başa çıkmak zorunda bırakmaktadır. Bu zorlu ortamla başa çıkmak için Avrupa gemi inşaat endüstrisi, yüksek katma değerli özelliklerle teknolojik olarak gelişmiş ve karmaşık gemilerin inşasına öncelik vererek net bir uzmanlaşma stratejisi benimsemiştir.

1.2. TÜRKİYEDE VE DÜNYADA TERSANECİLİK

OECD (2021) verilere göre 2000'li yıllarda Türkiye'nin gemi inşaat sektörü, güçlü ihracat, yerel bankalardan dinamik finansman ve sürdürülen ekonomik istikrar gibi birkaç önemli faktörün etkisiyle dikkate değer bir genişleme yaşamıştır. Başlangıçta daha küçük ölçekli petrol ve kimyasal tankeri inşaatına odaklanan Türk tersaneleri, 2008 Küresel Finansal Krizi'ne yanıt olarak gemi tamiri, hurdalama ve alternatif yakıtlarla çalışan gemilerin geliştirilmesi gibi özel alanlara stratejik bir kayış sergilemiştir. Bu piyasa dinamiklerine uyumlu adaptif yaklaşım, Türk gemi yapımcılarının özellikle Avrupa müşterilerinin belirli ihtiyaçlarını karşılayarak küresel gemi siparişlerindeki düşüş dönemini başarıyla atlattıklarını sağlamıştır (OECD, 2021).

Tekrardan OECD (2021) verilere göre 2019'a gelindiğinde, Türkiye küresel gemi inşaatında 11. sıraya yükselerek, balıkçı teknelerinden tankerlere, römorkörlerden feribotlara ve LNG, hibrit veya elektrik tahrik sistemleri kullanan gemilere kadar geniş bir gemi yelpazesini tamamlama kabiliyetini göstermiştir. Ayrıca, Türkiye gemi tamiri, bakımı ve geri dönüşüm operasyonlarında da kritik bir rol oynamıştır, denizcilik endüstrisindeki konumunu daha da sağlamlaştırmıştır. Türk gemi inşaat sektörü, 2008 kriz sonrasında istihdamda bir toparlanma yaşamıştır ancak 2020'de COVID-19 pandemisi nedeniyle gemi siparişlerindeki yavaşlama gibi yeni zorluklarla karşılaşmıştır (OECD, 2021).

Türkiye'nin coğrafi ve stratejik konumu, hem bölgesel hem de küresel düzeyde büyük bir öneme sahiptir. Üç tarafı denizlerle çevrili olan Türkiye, Asya ve Avrupa kıtalarını birleştiren kritik bir köprü görevi görmektedir. Boğazlar (İstanbul ve Çanakkale) dünya deniz ticaretinin önemli geçiş noktaları olup, Karadeniz'i Akdeniz'e ve dolayısıyla dünya ticaretini bağlamaktadır.

Deniz Ticaret Odasının paylaştığı raporunda Türkiye dünya genelinde beşinci sırada yer aldığı gemi inşaatı ülkesi olmaktadır (Clarkson Research Services, 03/2023). Tersaneler, yerel düzenlemelerdeki tesis tanımına göre, Mart 2022 itibarıyla 84'e yükselirken, 2002'de sadece 37 idi (Ulaştırma ve Altyapı Bakanlığı, 03/2023). Aynı tarihte inşaat halindeki tersane sayısı 9 ve tersane yatırım alanı olarak tanımlanan 13 alan bulunmaktadır. Küresel Ekonomik Krizle birlikte Covid-19 pandemisi, gemi inşaat sektörünü de birçok diğer sektör gibi olumsuz etkilemiştir. Sipariş defterlerindeki azalma hem istihdamda hem de yeni yatırımlarda düşüşe neden olmuştur, bu yüzden çoğu tersane modernizasyon projelerini iptal etmiştir veya ertelemiştir. 2022 yılında toplam 55.798 ton DWT kapasitesinde 17 gemi teslim edilmiştir

(Clarkson Research Services, 03/2023). Türkiye'deki faal tersanelerin bazıları modernizasyon ve genişleme çalışmalarına devam etmektedir. Ancak, küresel ekonomik kriz nedeniyle, bazı tersaneler bankalar tarafından uygulanan yaptırımlar nedeniyle modernizasyon veya genişleme projelerini askıya almakta veya iptal etmektedir (Deniz Ticaret Odası, 03/2023).

1.3. TERSANEDE YAPILAN FAALİYETLER

Gemi İnşası:

Gemi inşası, gemilerin tasarımı, inşası, donatılması ve test edilmesi gibi çeşitli aşamaları içeren karmaşık bir süreçtir.

- Tasarım: Gemi İnşa mühendisleri, geminin boyutu, kararlılığı, tahrik sistemi ve amaçlanan kullanım gibi faktörleri göz önünde bulundurarak geminin tasarımı üzerinde çalışmaktalar.
- İnşaat: Gemi inşası, çelik levhaların, kaynak yapmanın ve diğer imalat tekniklerinin kullanılmasıyla geminin gövdesinin ve üst yapısının montajını içermektedir.
- Donatma: Temel yapı tamamlandıktan sonra, gemi tahrik sistemleri, elektrik sistemleri, navigasyon ekipmanları, yaşam mahalleri ve diğer temel bileşenlerle donatılmaktadır.
- Test: Teslimattan önce, gemi performans standartlarını ve güvenlik yönetmeliklerini karşıladığından emin olmak için deniz denemelerine tabi tutulmaktadır.

Gemi Bakımı ve Onarımı:

- Tersaneler, mevcut gemilerin işlevsel durumda kalmasını sağlamak ve ömürlerini uzatmak için bakım ve onarım hizmetleri sunulmaktadır.
- Rutin Bakım: Bu, temizlik, boyama, makine yağlama ve ekipmanın aşınma belirtileri açısından kontrol edilmesi gibi görevleri içermektedir.
- Onarım: Tersaneler, kazalardan, korozyondan veya mekanik arızalardan kaynaklanan hasarların onarımı ile ilgilenmektedir. Bu, kaynak yapma, hasar görmüş parçaların değiştirilmesi gibi işleri içerebilmektedir.
- Yenileme: Gemi sahipleri, gemilerini yeni teknolojilerle yükseltmek, verimliliği artırmak veya düzenleyici gereksinimleri karşılamak için yeniden düzenleme seçeneğini tercih edebilmektedir.

Deniz Mühendisliği Hizmetleri:

Deniz mühendisliği hizmetleri, gemi tasarımı, inşası ve işletmesiyle ilgili çeşitli teknik uzmanlık alanlarını kapsamaktadır.

- **Tasarım Danışmanlığı:** Tersaneler, gemi tasarımı, yapısal analizler ve uluslararası düzenlemelere uygunluk konularında müşterilere danışmanlık hizmetleri sunmaktalar.
- **Kararlılık Analizi:** Bir geminin denizde güvenliği için kararlılığının sağlanması önemlidir. Deniz mühendisleri, bir geminin devrilme direncini belirlemek için kararlılık analizleri yapmaktalar.

Gemi Donanımı ve Ekipmanı Temini:

Tersaneler, gemi işletimi ve güvenliği için gerekli çeşitli donanım ve ekipmanı temin eder ve kurmaktadır.

- **Tahrik Sistemleri:** Bunlar, gemi tahrik ve manevra için gerekli motorlar, pervaneler, şaftlar, iticeler, jeneratör ve seperatör içermektedir.
- **Navigasyon Ekipmanı:** Tersaneler, güvenli bir seyir sağlamak için radar sistemleri, GPS, ECDIS, AIS, VDR, otomatik pilot ve haberleşme cihazlarının tedarik ve montajını yapmaktalar.
- **Güvenlik Donanımı:** Can kurtaranlar, can salı, yangın söndürme sistemleri ve acil sinyal cihazları (Epirb ve Sart gibi), deniz güvenliği için temel öneme sahiptir ve tersaneler tarafından sağlanmaktadır.

Denizcilik Araştırma ve Geliştirme Faaliyetleri:

Bazı tersaneler, gemi tasarımı, performansı ve çevresel sürdürülebilirliği geliştirmek için araştırma ve geliştirme faaliyetlerine katılmaktadır.

- **Alternatif Yakıtlar:** Biyoyakıtlar (very low sulfur fuel oil) gibi alternatif yakıtlara yönelik araştırmalar, emisyonları azaltmayı ve gemi sevkياتında enerji verimliliğini artırmayı amaçlamaktadır.
- **Yeşil Teknolojiler:** Tersaneler, egzoz gazı temizleme sistemleri ve balast suyu arıtma sistemleri gibi çevre dostu teknolojileri araştırır ve geliştirmektedir.

Gemilerin Sınıflandırılması ve Belgelendirilmesi:

Gemi sınıflandırma kuruluşları, gemilerin uluslararası güvenlik, kalite ve çevre koruma standartlarını karşıladığından emin olmak için önemli bir rol oynamaktadır.

- Sınıflandırma Denetimleri: Sınıflandırma kuruluşlarından bağımsız denetçiler (Bayrak devleti, klas veya sigorta kuruluşları gibi), teknik standartlara uygunluğu doğrulamak için inşaat sürecinin her aşamasında denetimler yapmaktadır.
- Belgelendirme: Bir gemi tamamlandığında, sınıflandırma topluluğu tarafından sertifikasyon sürecine girmektedir ve sınıf sertifikaları, güvenlik sertifikaları ve kirlilik önleme sertifikaları gibi gerekli belgeleri almaktadır.

Bu faaliyetler, küresel ticareti, taşımacılığı ve deniz operasyonlarını destekleyerek denizcilik endüstrisinin işleyişine ve gelişimine katkıda bulunmaktadır.

1.4. TERSANEDE GEMİ İNŞAASI

Gemi inşa edenler, fiyat belirlerken maliyetler ile piyasa fiyatı arasında denge kurmak zorundadır; genellikle piyasa fiyatı referans alınmaktadır ancak maliyetler de göz önünde bulundurulmaktadır (Shetelig, 2013). Gemi inşa etmek çok yoğun ve iyi bir analiz isteyen bir prosedürdür. Gemi inşa endüstrisi tek bir endüstriden oluşmamaktadır. Yan sanayiye, tedarikçilere, taşeronlara ve lojistik şirketlere de bağlı aynı zamanda. Bu nedenle piyasa araştırmak elzemdir çünkü ona göre bir gemi inşa edilecektir.

Britannica (2024, <https://www.britannica.com/technology/ship-construction>, Erişim tarihi: 01.06.2024) paylaştığı makalede, gemi inşa mühendisi, belirli bir kargo ağırlığını veya yolcu sayısını belirli bir hızda taşıyabilen bir gemi tasarlamakla görevli olduğunu ve ticaret gereksinimlerini dikkate aldığını vurgulanmıştır. Yük türü ve yoğunluğu, geminin kapasitesini etkilediğini belirlemiştir. Ayrıca ölü ağırlık, kargo, yakıt ve tüketim malzemelerinin ağırlığını, hafif ağırlık ise makine ve ekipman dahil gövdenin ağırlığını kapsadığını belirlemiştir. Mühendis, geminin yer değiştirme miktarının ölü ağırlık ve hafif ağırlık tonajlarının toplamına eşit olduğundan emin olacak boyutları seçmektedir. Geminin şekli, istenen hıza uygun olmalı ve serbest borda kuralları tarafından düzenlenen çekme, geminin derinliğini belirlemektedir.

Bireysel bileşenlerin üretiminden alt montajlar ve modüller halinde birleştirilmesine ve nihai gemi montajına kadar tüm aşamaları kapsamaktadır. Tersaneler, gemi inşası ve denize indirilmesi için gerekli altyapıya sahip olup, büyük vinçler kullanarak çeşitli gemi bileşenlerini taşımaktadır ve monte etmektedir. Bu süreçlerin yönetimi, gemi inşaat sanayiinin karmaşıklığını vurgulamaktadır.

Britannica'nın (2024, <https://www.britannica.com/technology/ship-construction>, Erişim tarihi: 01.06.2024) paylaştığı gemi inşa süreci aşağıdaki gibi özetilmiştir ve açıklanmıştır :

1. Tersane Organizasyonu: Tersaneler, belirli zaman çizelgeleri ve bütçeler dahilinde gemiler teslim etmektedir. Çeşitli yetenekli işgücü ve taşeron yönetimi gerektirmektedir.
2. Layout: Çelik levhalar ve bölümler, prefabrikasyon atölyesine taşınarak alt montajlar oluşturulmaktadır ve daha sonra büyük prefabrik birimler halinde birleştirilmektedir.
3. Planlama: Gemilerin zamanında tamamlanması için titiz planlama gerektirmektedir. CPM, projeleri grafik olarak temsil etmektedir ve her görevin zaman tahminlerini sağlamaktadır.
4. Çizgi Planı ve Düzenleme: Geleneksel çizgiler planı, geminin 1/48 ölçekli çizimidir. Bilgisayarlar, ön çizgi planlarını düzenleyerek verimliliği artırmaktadır.
5. Üretim ve Montaj: Büyük gövdeler, önceden monte edilerek daha verimli bir yapım süreci sağlamaktadır.
6. Denize İndirme: Gemilerin kıçtan denize bırakılması, düzgün bir kayma sağlamak için ayrıntılı hesaplamalar gerektirmektedir.
7. Donatım: Denize indirildikten sonra gemiler, ana makine, yardımcı sistemler ve diğer ekipmanların montajını içeren donatım havuzlarına taşınmaktadır.
8. Denemeler: Gemi tamamlanmaya yaklaştığında çeşitli testler yapılmaktadır. Karada ve denizde yapılan testler, geminin performansını değerlendirmektedir.

Bu aşamalar, gemi inşa sürecinin kapsamlı bir planını oluşturmaktadır ve gemi yapımının karmaşıklığını ortaya koymaktadır.

Gemi inşasının beceriye dayalı bir zanaattan bilim destekli bir disipline evrimi, modern gemilerin verimliliğini, güvenliğini ve performansını önemli ölçüde artırmıştır. Britannica'dan kapsamlı bir makalede açıklandığı gibi, gemi inşa mühendisleri ve tersanelerin rolleri bu karmaşık süreçte hayati öneme sahiptir ve titiz planlama, ileri teknoloji ve çeşitli zanaat ve

bölümler arasındaki koordineli çabalar gerektirmektedir (https://www.britannica.com/technology/ship-construction, Erişim tarihi: 01.06.2024).

1.5. TERSANEDE YAT İNŞAASI

Yatçılık, 17. yüzyıla kadar uzanan zengin bir tarihsel dokuya sahipti çünkü bu dönemde Hollandalı tüccarlar, "jachts" olarak adlandırdıkları çevik gemilerin inşasında öncü rol oynamışlardır ve bunu rekreasyon amaçları için gerçekleştirmişlerdir. Zamanla, yat ustalığı önemli ölçüde evrim geçirmiş ve malzemeler ile tekniklerdeki yenilikler endüstrinin manzarasını şekillendirmiştir (ISSC Committee V.8: Sailing Yacht Design, 2009). 1800'lerin sonlarında ahşap ve geleneksel gemi yapım yöntemleri kullanılarak başlayan yat yapımı, sofistike tasarım ve mühendislik alanına ilerlemiştir. 1970'lerde epoksi reçinelerin kullanılmaya başlanması ahşap yapıların dayanıklılığını artırmıştır, 1930'larda çelik çerçevelerle ahşap kaplamaların bir araya getirilmesi, transatlantik seyahatler için uygun sağlam çelik gövdeli gemilerin ortaya çıkmasına olanak sağlamıştır. 1950'lerde fiberglass takviyeli polysterin devreye girmesiyle seri üretim çağı başlamış ve hafif malzemelerin kullanılması gibi devrim niteliğindeki yeniliklerle desteklenmiştir. Direk yapımı da benzer bir evrim geçirmiş, 1930'larda ahşaptan alüminyum alaşımlara ve daha sonra 1980'lerde karbon fiber kompozitlere geçilerek ağırlık, rijitlik ve performansta önemli iyileştirmeler sağlanmıştır. (ISSC Committee V.8: Sailing Yacht Design, 2009)

Yatların bayraklarına, hizmet amacına ve boyutlarına bağlı olarak yasal düzenlemeler değişmektedir. Aşağıdaki gibi listelenmiştir ve açıklanmıştır (ISSC Committee V.8: Sailing Yacht Design, 2012):

- Özel Yatlar: Zorunlu gereksinimler oldukça hafiftir; çoğu bayrak için, bir kayıt denetimi ve yetkilendirilmiş bir denetleyici tarafından gerçekleştirilen tonaj ölçümü geçerli standartlardır.
- Ticari Yatlar: Hemen hemen tüm büyük Bayrak İdareleri, ticari yatların belirli bir büyük yat güvenlik koduna göre sertifikalandırılmasını gerektirmektedir.
- Yolcu Yatları: Son beş yılda yatların ortalama boyutunun dramatik bir şekilde arttığı göz önüne alındığında, "12 yolcu" üst sınırının kısıtlayıcı olduğu açıktır. Bu nedenle, yatların 36 yolcuyu taşımasına değiştirilmiştir.

- Rekreatyoneel Tekneler: Bu kategori, 24 metrenin altında bir uzunluęa sahip tüm tekneleri içermektedir.

Gemi inşası gibi yat inşası da hemen hemen aynı süreçlerden geçmektedir. Yat inşası, gemiden daha hızlıdır ve nispeten daha kolaydır. Ancak yat inşası, amaca göre değişmektedir. Aynı şekilde amaca göre, daha çok estetiğine dikkat edilmektedir. Genel olarak, yat inşası süreçleri aşağıdaki gibidir :

- Tasarım ve Planlama: Yat inşası, tasarımcılar ve mühendisler tarafından başlamaktadır. Müşterinin isteklerine ve ihtiyaçlarına göre tasarımlar oluşturulmaktadır ve teknik çizimler hazırlanmaktadır.
- Malzeme Seçimi ve Hazırlığı: İnşaat için gerekli olan malzemeler titizlikle seçilmektedir ve hazırlanmaktadır. Bu genellikle çelik, alüminyum veya fiberglas gibi malzemeleri içermektedir.
- Gövde İnşası: İnşaat, genellikle yatın gövdesiyle başlamaktadır. Gövde, seçilen malzemeye göre ya kaynaklanarak ya da kalıplar kullanılarak şekillendirilmektedir.
- Güverte ve İç Döşeme İşleri: Gövde tamamlandıktan sonra, güverte ve iç mekan döşeme işleri başlamaktadır. Bu aşamada, güverte kaplamaları, kabin mobilyaları, elektrik tesisatı gibi detaylar eklenmektedir.
- Deneme Seferleri ve Son Kontroller: Yat tamamlandıktan sonra deneme seferleri yapılmaktadır. Bu seferlerde yatın denizdeki performansı ve davranışı test edilmektedir. Son kontroller ve kalite kontrol önlemleri yapılmaktadır.
- Teslimat ve Müşteri Eğitimi: Yat, müşteriye teslim edilirken, yatın işleyişi ve bakımı hakkında gerekli eğitim verilmektedir.

1.6. TERSANEDE BAKIM ONARIM İŞLEMLERİ

Tersanede bakım ve onarım işlemleri, gemilerin güvenli ve verimli bir şekilde çalışmasını sağlamak amacıyla gerçekleştirilen kapsamlı ve sistemli faaliyetlerdir. Tersanede, tipik olarak işin %75'i rutin gemi bakımı içermektedir ve kalan %25'i hasar onarımı ve gemi dönüştürme işlerinden oluşmaktadır (Mackenzie, 2004). Bu süreç, genellikle ayrıntılı bir planlama ve hazırlık aşaması ile başlamaktadır. İlk olarak, geminin ihtiyaç duyduğu bakım ve onarım gereksinimlerini belirlemek için detaylı bir inceleme ve değerlendirme yapılmaktadır. Bu değerlendirme sonucunda gerekli malzemeler, ekipman ve iş gücü planlanarak işlerin zaman

çizelgesi oluşturulmaktadır. Bakım ve onarım işlemleri için gemi kuru havuza alınarak su ile teması kesilmektedir ve böylece geminin alt kısımlarına erişim sağlanmaktadır. Alternatif olarak, kızaklama yöntemi kullanılarak geminin su seviyesinin altındaki bölgelerinde bakım yapılmaktadır. Kuru havuzlama ve kızaklama işlemleri, geminin alt gövdesine erişim ve çalışma kolaylığı sağlamaktadır.

Temizlik ve yüzey hazırlığı aşamasında, geminin yüzeyleri kumlama yöntemi ile temizlenmektedir ve paslanmaya karşı koruma sağlamak amacıyla boyanmaktadır. Bu aşamada ayrıca, geminin gövdesine yapışmış deniz organizmaları ve kirler temizlenmektedir. Geminin dış kısmının boya yapılmaktadır ve draft sayılarının ve Plimsol markalarının güncellenmesi yapılmaktadır. Ayrıca, ihtiyaca bağlı double bottom, tank top ve koferdam tankların kontrolleri ve boyaması yapılmaktadır. Kuru yük gemileri için, tersanede ambar kapakların bakımı ve kontrolleri yapılmaktadır. Kapak tipine göre kapaklar sökülmemektedir ya da zincirleri değiştirilmektedir. Geminin türüne, yaşına ve ihtiyacına göre aynı zamanda tutya değişimi yapılmaktadır. Mekanik ve elektrik bakımında, ana motorlar, yardımcı motorlar ve diğer makine ekipmanları kontrol edilir, gerekli bakım ve onarımlar yapılmaktadır. Geminin elektrik sistemleri de detaylı bir şekilde incelenmektedir, arızalı veya eskimiş parçalar yenilenmektedir.

Borulama ve valf bakımında, geminin yakıt, su ve havalandırma boruları kontrol edilerek gerektiğinde değiştirilmektedir. Valfler sökülüp temizlenir, sızdırmazlık testleri yapılarak tekrar monte edilmektedir. Güverte ve yapısal bakım aşamasında ise güverte yüzeyleri onarılmaktadır, gerekirse yeni kaplamalar yapılmaktadır. Gövde, perde ve diğer yapısal elemanlarda çatlak veya hasar varsa kaynak işleriyle onarılmaktadır.

Bakım ve onarım işlemlerinin sonunda, tüm sistemler ve ekipman çalıştırılarak fonksiyon testleri yapılmaktadır. Boru hatları ve tanklar gibi kritik bileşenler sızdırmazlık ve basınç testlerine tabi tutulmaktadır. Yapılan tüm işlemler detaylı olarak raporlanmaktadır ve gerekli denetimler yapılarak geminin işletmeye uygunluğu sertifikalandırılmaktadır. Bu süreç boyunca ulusal ve uluslararası denizcilik standartlarına uyulması son derece önemlidir.

Ek olarak, geminin denetleme zamanına göre, emniyet ekipmanları ve araçları kontrolleri, bakımları ve testleri yapılmaktadır. Gemi türüne göre ve kurtarma araçlarına göre, bazı testleri emniyet açısından tersanede yapılması zorunludur. Böylece, geminin denizde güvenli ve emniyetli bir şekilde operasyonlarına devam etmesi sağlanmaktadır.

1.7. GEMİ İNŞAATINDA TEDARİK ZİNCİR YÖNETİMİ

Pujawan (2005) tarafından yapılan açıklamaya göre, tedarik zinciri, bir ürünü tüketicilere veya son kullanıcılara ulaştırmak için bir araya gelen bazı şirketlerin veya kuruluşların ağıdır. Şirket genellikle tedarikçileri, üreticileri, distribütörleri, mağazaları veya perakende satış noktalarını ve lojistik hizmet şirketi gibi destekleyici şirketleri içermektedir. Bir tedarik zinciri, ortaklar ağıdır ve temel bir hammaddenin (yukarı akış) son kullanıcılar tarafından değer verilen bir nihai ürüne (aşağı akış) dönüştürüldüğü ve her aşamada iadelerin yönetildiği bir ağıdır. Tedarik zinciri, işbirliği içinde çalışarak başlangıçta temel bir malzeme olan bir ürünü müşterilere değerli bir son ürüne dönüştüren ortakların ağıdır.

Islam ve diğerlerin (2017), iç endüstri süreçlerinin dış tedarik zinciri ağlarıyla entegrasyonunun verimliliği artırmak için önemini vurgulamaktadır. Bu entegrasyon, bilgi paylaşımı, ilişki kurma ve işbirliği ile sağlanabilmektedir. Tedarik zinciri entegrasyonunu destekleyen temel unsurlar işbirliği, ortaklık, bilgi paylaşımı ve teknolojik yenilik olarak belirlenmiştir. Bireysel yönetim süreçlerinden entegre yönetim zincirlerine geçiş önemlidir. Bilgi şeffaflığı, endüstri, tedarikçiler ve alıcılar arasında sorunsuz iletişimi sağlamaktadır. Ortaklar arasındaki güven, rekabetçi piyasalarda başarı için hayati öneme sahiptir. Gemi inşaat endüstrisi, tedarik zinciri modernizasyonu ve teknoloji entegrasyonundan faydalanabilmektedir.

Shipbuilding Partners (2023, <https://www.linkedin.com/pulse/strategies-optimize-shipbuilding-supply-chains-thedrydok/>, Erişim Tarihi : 01.06.2024.) tarafından yazılan makalede Avrupadaki gemi inşa endüstrinin, tedarik zinciri yönetiminde karşılaştığı zorlukları aşağıdaki gibi listelenmiştir ve açıklanmıştır :

- **Malzeme Çeşitliliğinin Tedariki:** Avrupa gemi yapımı genellikle özel alaçımlardan yüksek kaliteli iç dekorasyon için ahşaba kadar geniş bir malzeme yelpazesine ihtiyaç duymaktadır. Bu malzemelerin sürekli bir tedariki, bazen çeşitli ülkelerden gelmesi gereken bir süreci içermektedir ve bu da lojistik olarak karmaşık bir durumu beraberinde getirebilmektedir (Shipbuilding Partners, 2023, Erişim Tarihi : 01.06.2024).
- **Lojistik Zorluklar:** Malzemelerin ve parçaların Avrupa sınırları içinde taşınması, sıkı gümrük prosedürleri, değişen yönetmelikler ve özel taşımacılık gereksinimleri gibi lojistik zorluklarla karşı karşıya bırakabilmektedir (Shipbuilding Partners, 2023, Ziyaret tarihi : 01.06.2024).

- **Düzenleyici Uyum Gereksinimleri:** Avrupa gemi yapım endüstrisi, sıkı çevresel, güvenlik ve kalite yönetmelikleri altında çalışmaktadır. Bu standartlara uyum sağlamak, tedarik zinciri boyunca titiz bir belgelendirme ve uyum gerektirmektedir (Shipbuilding Partners, 2023, Erişim Tarihi : 01.06.2024).
- **Kaynakların Tahsisi:** Proje sürelerine yetişmek için etkili kaynak yönetimi kritiktir. Belirli bölgelerdeki nitelikli işgücüne olan talep yüksek olabilmektedir, bu da bazı aşamalarda darboğazlara neden olabilmektedir (Shipbuilding Partners, 2023, Erişim Tarihi : 01.06.2024).
- **Tedarik Zinciri Kesintileri:** Doğal afetler, jeopolitik gerilimler ve küresel ekonomik dalgalanmalar gibi faktörler, tedarik zincirlerinde aksamalara neden olabilmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).
- **Bütçe Kısıtlamaları:** Maliyet yönetimi sürekli bir mücadele olabilmektedir. Bir alanda fazla harcama yapılması, diğer alanlarda bütçe kısıtlamalarına neden olabilmektedir ve bu da projenin başarısını tehlikeye atabilmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).
- **Mevsimsel Değişimler:** Bazı gemi tiplerinin talebi mevsimlik olarak değişebilmektedir, bu da üretim planlamasını karmaşık hale getirebilmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).
- **Kalite Kontrolü:** Gemi yapımında yüksek kalite standartlarının korunması hayati önem taşımaktadır. Alt standart malzemeler veya parçalar, maliyetli yeniden işleme ve gecikmelere yol açabilmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).
- **Teknolojik Entegrasyon:** Geleneksel gemi yapım bölgelerinde dijital araçların ve teknolojilerin kabul edilmesi zor olabilmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).
- **Çevresel Endişeler:** Sürdürülebilirlik, gemi yapımında giderek daha önemli hale gelmektedir (Shipbuilding Partners, 2023, Erişim tarihi : 01.06.2024).

Bu zorlukların üstesinden gelmek için, gemi yapım endüstrisinin tedarik zinciri yönetiminde aktif ve stratejik bir yaklaşım benimsemesi gerekmektedir. Gemi inşa tedarik zinciri boyunca, tedarikçiler, taşeronlar, tersane işçileri ve düzenleyici kurumlar dahil tüm paydaşlar arasında etkili koordinasyon ve iletişim esastır.

1.8. TERSANEDE MALZEME TAŞIMACILIĞI

Tersane içindeki malzeme akış yönetimi, denizcilik endüstrisinde üretim verimliliğini optimize etmek için kritik öneme sahip çok yönlü bir çabadır. Muther ve Hales (2015) , malzeme akışının analizinin, ilgili süreçlerin gerekli adımlarında malzemelerin en etkili hareket dizilerinin belirlenmesini ve bu hareketlerin yoğunluğunu veya büyüklüğünü içerdiğini belirtmektedir. Etkili bir akış, malzemelerin sürecin gerekli adımlarından sürekli olarak ilerleyerek tamamlanmaya doğru ilerlediği ve aşırı sapmalar veya geriye doğru hareketler olmaksızın (ters akış) ilerlediği anlamına gelmektedir.

Verimli malzeme akışı, gemi yapım sürecinin her aşamasında doğru malzemelerin doğru yerde ve zamanda bulunmasını sağlamak için hayati öneme sahiptir. Bunun için, farklı departmanlar ve tedarikçiler arasında dikkatli planlama, lojistik yönetimi ve koordinasyon gereklidir. Malzeme akışının optimize edilmesi, gecikmeleri en aza indirmektedir, envanter maliyetlerini azaltmaktadır ve genel üretim verimliliğini artırmaktadır.

Muther ve Hales'e (2015) göre, bazı hareketlerin diğerlerinden daha zor veya maliyetli olduğu durumlarda, bunlar malzeme akışı analizine yansıtılmalıdır. Taşıma yöntemleri belirlenmiş veya biliniyorsa, her bir rotadaki malzeme akışını kolaylık veya zorlukla ilişkilendirilmiş her sınıf veya türdeki malzeme ve her taşıma cihazı tipini dikkate alarak "faktörize" etmek oldukça basit olabilmektedir. Tipik düşünceler arasında hız, güvenlik, risk, hareketi gerçekleştirmek için gereken kişi sayısı, üretimdeki aksama vb. bulunmaktadır.

Sonuç olarak, tersanelerde malzeme akışının etkili bir şekilde yönetilmesi, üretim verimliliğini optimize etmek için kritik öneme sahiptir. Gelişmiş teknolojiler ve stratejik optimizasyon teknikleri kullanılarak, bu zorlukların üstesinden gelmek ve operasyonel mükemmelliği sağlamak mümkündür.

1.9. DOK VE ALAN OPTİMİZASYONU

Caprace ve diğ. (2013), gemi tersanelerin daha fazla yapısal blok kullanarak (modüler inşaat stratejisi) eşzamanlı görev sayısını artırmak için tasarım yöntemlerini değiştirdiğini söylemektedir. Geleneksel olarak, tasarım kararlarının çoğu, deneyim ve tasarımcıların görüşlerine dayanarak alınıyordu. Bu kararlar üretim maliyetleri üzerinde güçlü bir etkiye sahip olmakla birlikte, daha sonraki yaşamı boyunca geminin performansını da etkilemektedir.

Büyük elemanların montajı, üretimi gerçekleştirmek için üretim atölyesinde gerekli alanı gerektirmektedir. Bloklar daha büyük ve ağır hale geldikçe, tersanede üretim alanı bir kısıtlama haline gelmektedir. En büyük bloklar, kaldırma ve taşıma sınırları nedeniyle üretilebilecekleri alanlarda sınırlıdır. Bu nedenle, bu alanlardaki alanın doğru planlanması, blokların yalnızca gerektiğinde ve gerektiği yerde taşındığından ve mevcut alanın verimli bir şekilde kullanıldığından emin olmak için önemlidir. Gereksiz hareketler, bloğa katma değeri olmayan maliyetlere yol açmaktadır. Ancak, üretim kısıtlamaları ve agresif inşaat programları nedeniyle bir bölgedeki blok sayısını maksimize etmek gereksiz hareketlere yol açabilirken, gereksiz hareketleri en aza indirme, mevcut alanın daha az verimli kullanılmasına neden olabilmektedir (Caprace ve diğ.,2013).

Güvenlik ve çevresel endişeler, tersane optimizasyon çabalarının ayrılmaz bir parçasıdır. Kapsamlı güvenlik protokolleri, ergonomik tasarım prensipleri ve sürekli eğitim programları, işçilerin refahını korumak ve kazaları ve yaralanmaları en aza indirmek için önemlidir. Cheng ve diğerlerine (2004) göre, bir kuru havuz genellikle bir su kütesinin kıyısına kazılmış bir havuzdan oluşmaktadır ve su tarafında su geçirmez bir kapak ile donatılmıştır ve gemilerin büyük onarımları ve revizyonları için kullanılmaktadır. Bir gemi havuza alınacağına, kuru havuz suyla doldurulur ve kapak açılmaktadır. Gemi getirildikten sonra, uygun bir şekilde konumlandırılır ve bağlanmaktadır, kapak kapatılmaktadır ve havuz pompalanarak kurutulmaktadır, bu da gemiyi yavaşça destekleyen kılavuz ve su tutma bloklarına oturtmaktadır. Bu kılavuz bloklarının konumlandırılması ve sertlik tahsisi, bir gemiyi havuza alırken önemli bir karardır çünkü kılavuz bloklarının yanlış konumlandırılması veya yanlış tahsis edilmesi, gereğinden fazla büyük blok reaksiyonlarına ve dolayısıyla hem havuza alınan gemiye hem de bloklara ciddi zararlar verebilmektedir. Havuz bloğunun başarısız olması, havuza alma programlarının bozulmasına ve gemi bakım süresinin uzamasına neden olabilmektedir. Herhangi bir başarısızlık, hatta can kaybına bile yol açabilmektedir.

Tersane işletmelerinde, iskele ve alan optimizasyonu büyük önem taşımaktadır. İskele optimizasyonu, gemilerin bakımı ve yapım süreçlerini verimli hale getirirken, alan optimizasyonu da tersane alanının etkin kullanımını sağlamaktadır. Bu, düzenli iş akışı, malzeme yönetimi ve lojistik süreçlerin optimize edilmesini içermektedir. İskele ve alan optimizasyonu, işletme verimliliğini artırırken maliyetleri düşürmektedir ve projelerin zamanında tamamlanmasını sağlamaktadır, böylece tersanenin rekabet gücünü artırmaktadır.

1.10. GEMİ İNŞAATINDA İŞ GÜCÜ MAALİYETLER

Wu (2019), 2008'den beri gemi piyasasının "soğuk bir kış" dönemine girdiğini belirtmektedir. Gemi piyasasındaki uzun vadeli durgunluğun düşündürücü olduğunu belirtmektedir. Son on yılda denizcilik endüstrisinin toplam maliyetinin arttığı, ancak faydaların genel olarak azaldığı gözlemlenmektedir. Bir gemi yapım endüstrisinin toplam maliyeti yaklaşık olarak dört ana noktaya bölünmektedir: işçilik maliyeti, hammadde maliyeti, destekleyici tesis maliyeti ve tasarım yönetimi maliyeti. Bunlar arasında, işçilik maliyeti toplam gemi maliyetinin yaklaşık %30'unu, hammadde maliyeti toplam maliyetin yaklaşık %25'ini, geminin çeşitli destekleyici tesisleri toplam maliyetin yaklaşık %40'ını ve tasarım yönetimi maliyeti ise toplam maliyetin yaklaşık %5'ini oluşturmaktadır (Wu, 2019).

Wu (2019), ayrıca gemi inşa sürecinde işçilik maliyetinin kontrolünün önemini ve anlamını vurgulamıştır. Gemi inşa sürecinde işçilik maliyeti, üretimin toplam maliyetinin önemli bir parçasıdır ve işletmenin nihai kârıyla ilişkilidir; ayrıca gemi inşa işletmesinin her bir çalışanın çıkarlarıyla da yakından ilişkilidir. Yüksek verim ve yüksek ücretli durumda, çalışanların hevesi buna uygun olarak artacaktır. İnsana odaklılık, bilimsel gelişme kavramının çekirdeğidir. Bir şirketin geleceği de her bir çalışanın desteğiyle desteklenmektedir. Bu nedenle, işletmelerin canlı ve sağlıklı bir şekilde gelişimini sağlamak için işçilik maliyetlerinin makul ve etkili bir şekilde kontrol edilmesi gerekli bir önlemdir.

Wu (2019), tersane işçilik maliyet kontrolü için iyileştirme önlemleri, işe alım politikalarının optimize edilmesi, işçilik maliyet kontrol sistemlerinin kurulması, üretim bağlantılarının koordinasyonu ve akıllı trendlere uyum sağlama konusunda bahsetmiştir. İşe alım politikalarının ve maaş teşvik sistemlerinin iyileştirilmesi, uygun maliyetli işgücünün çekilmesinin gerekliliğini vurgulamıştır. Ayrıca, işçilik maliyet kontrol sistemlerinin bütçe yönetimi temelinde kurulması gerektiği ve üretim bağlantılarının yakın koordinasyonu ve ödül-ceza mekanizmasının uygulanması gerektiği belirtilmektedir. Bu önlemler, işgücü maliyetlerini kontrol etme amacını taşıırken, akıllı sistemlere uyum sağlamayı ve işletmelerin rekabet gücünü artırmayı amaçlamaktadır. Özetle, tersane endüstrisindeki gelişme trendlerini takip etmek, yenilik yapmak ve yapay zekayı işletmelerde uygulamak önemli olduğunu söylemiştir.

Sonuç olarak, gemi inşaat endüstrisinin sürdürülebilirliği, maliyet yönetiminin ötesinde daha geniş bir perspektife dayanmalıdır. Teknolojik yeniliklere, işbirliğine ve stratejik yönetim

uygulamalarına odaklanarak, endüstri, gelecekteki zorluklarla başa çıkma ve uzun vadeli başarıyı sağlama konusunda daha güçlü bir konuma gelebilmektedir.

1.11. DENİZ YOLU LOJİSTİĞİ

Denizcilik endüstrisi, bir ülkenin ekonomisini ve yaşam kalitesini iyileştirmeye destekleyen uluslararası ticaretin çekirdeğidir. UNCTAD verilerine göre Dünya ticaretinin %80'ini deniz taşımacılığı gerçekleştirdiği için denizcilik sektörü, küresel ticaretin önemli bir parça olarak kabul edilmektedir. Denizcilik sektörü hassas ve değişiklik gösteren bir sektördür. Ayrıca pandemiler, savaşlar, kazalar ve siyaset denizcilik sektörüne yön verebilen faktörler ama hiçbir zaman durduran faktörler değildir. Deniz taşımacılığı, diğer taşıma yöntemlerine kıyasla büyük miktarda malın uzun mesafelerde taşınması için önemli maliyet avantajları sunmaktadır. Urbanyi-Popiołek ve Klopott (2016) tarafından belirtildiği gibi, büyük miktarlarda kargonun gemiler aracılığıyla taşınması nedeniyle ölçek ekonomisinden kaynaklanan bir maliyet avantajına sahiptir. Ayrıca, gemilerin yüksek taşıma kapasitesi sayesinde, deniz taşımacılığı çeşitli kargo türlerini, toplu malları, konteynerleri ve büyük boyutlu ürünleri dahil olmak üzere çeşitli ihtiyaçlara uygun ve esnek hale getirmektedir.

Ancak, maliyet etkinliğine rağmen, deniz taşımacılığı teslimat sürelerini ve genel lojistik verimliliğini etkileyebilecek zorluklarla karşı karşıyadır. Limanlardaki uzun süren prosedürler, gümrük işlemleri, kargo denetimleri ve yanaşma düzenlemeleri gibi, kargo işleme ve gemi dönüş sürelerinde gecikmelere neden olmaktadır. Bu gecikmeler sadece transit süreleri uzatmakla kalmaz, aynı zamanda nakliye şirketleri için operasyonel maliyetleri artırmaktadır ve zamanında teslimatlara bağlı işletmelerin tedarik zinciri programlarını etkilemektedir.

Deniz taşımacılığı, uzun mesafeli taşımacılık için idealdir. Diğer taşımacılık türlerine göre daha ucuzdur ve çevre dostudur. Gemiler, çevre kirliliği açısından sürekli ve devamlı kontrollerden geçmektedir. Yeni çıkan teknolojilere ve regülasyonlara uyumak zorundadır. Sert ve sıkı regülasyonlardan dolayı, uyulmamsı durumunda cezaları çok yüksektir. Ayrıca yüksek hacimli ve ağırlıklı kargo için en kolay taşımacılık formudur. Deniz taşımacılığı, ticaretin temelidir. Simbiyotik bir şekilde bağlıdır. Bu nedenle de, inşa edilen gemiler coğrafik kanallara göre inşa edilmektedir. En basit örnekleri, Suveyş ve Panama kanalıdır. Bu kanallara göre, Suezmax ve Panamax gemi tipleri inşa edilmektedir. Bu kanallardan geçmek, mesafeyi kısaltarak ve zamanı düşerek deniz taşımacılığını kolaylaştırmaktadır.

Song ve Panayides'e (2015) göre denizcilik lojistiđi deęeri, denizcilik lojistiđi sistemi üzerinden yaratılan deęerdir ve denizcilik lojistiđi operatörleri hizmetlerini en verimli ve etkili şekilde sunarak bu deęeri maksimize edebileceđini vurgulamıştır. Bu, hem bireysel denizcilik operatörlerinin hem de genel lojistik sisteminin performansını artırabileceđi için, denizcilik operatörlerinin ulaşmak istediđi en önemli stratejik hedeflerden biri haline gelmiştir.

Deniz lojistiđini anlayabilmek için deniz taşımacılıđını da tanımak gerekmektedir. Deniz taşımacılıđı da çok renkli ve farklı bir taşımacılık türüdür. Gemi tipine, kargoya ve hizmete bađlı deęişmektedir.

Deniz taşımacılıđı endüstriyel, tramp ve lineer taşımacılıđı olarak üç ana kategoriye ayrılabilir ve aşıđıdaki gibi açıklanmıştır (Stopford, 2009):

- Endüstriyel Taşımacılık: Büyük miktarda hammadde, ara ürün veya nihai ürünlerin taşınmasıdır.
- Tramp Taşımacılık: Gemilerin belirli bir tarife veya düzenli rota takip etmeden çalıştığı esnek bir deniz taşımacılıđı türüdür (Stopford, 2009).
- Liner Taşımacılıđı ya da Hat hizmetleri: Gemiler, her bir mal için sabit fiyatlarla yük taşıyan ve düzenli bir tarifeli hizmet sunan limanlar arasında çalışmaktadır (Stopford, 2009).

Denizcilik lojistiđi, sadece deniz taşımacılıđına odaklanmakla kalmaz, aynı zamanda mal ve bilgi akışının etkin ve verimli bir şekilde sağlanmasını sağlayan lojistik süreçleri de içermektedir. Bu süreçler, depolama, ambalajlama, tamirat, yükleme ve boşaltma gibi lojistik faaliyetleri kapsamaktadır. Denizcilik lojistiđi, sadece deniz taşımacılıđının bir parçası olmakla kalmamalı, aynı zamanda küresel lojistik entegrasyon sisteminin merkezi bir bileşeni olarak diđer lojistik işlevlerle uyumlu olmalıdır.

Sonuç olarak, denizcilik lojistiđi, sadece deniz taşımacılıđı ile sınırlı kalmayan kapsamlı bir yaklaşım gerektirmektedir. Bu yaklaşım, tüm lojistik süreçlerin entegre edilmesini, verimliliđin artırılmasını, teknolojik gelişmelerini takip edilmesi ve müşteri memnuniyetinin sağlanmasını gerektirmektedir, böylece endüstri genelinde daha iyi bir performans elde edilmektedir.

2. KAVRAMSAL ÇERÇEVE

2.1. GENETİK ALGORİTMA

1970'lerde J. H. Holland Genetik Algoritmalar konusundaki öncü çalışmalarıyla tanınmıştır. Genetik Algoritmalar (GA'lar), doğal evrimin karmaşık mekanizmalarına dayanan uygulanabilir ve uyarlanabilir problem çözme metodolojileri olarak öne çıkmaktadır. Geçmişte "zor" veya hatta "imkansız" olarak kabul edilen sorunlar, artık hesaplama açısından bir sorun teşkil etmemektedir. Bu nedenle, geçmişte çıkmaza giren karmaşık ve çelişkili problemler, şimdi Genetik Algoritmalar (GA) ile çözülebilmektedir (Man ve diğ., 1996).

Goldberg'e (1989) göre genetik algoritmalar, doğal seçim ve doğal genetiğin mekanizmalarına dayanan arama algoritmalarıdır. Bu algoritmalar, insan arayışının bazı yenilikçi özellikleriyle yapılandırılmış fakat rastgele bilgi değişimini birleştirerek, string yapıların en iyileri arasında hayatta kalmayı sağlamaktadır. Her nesilde, en iyilerin bit ve parçalarını kullanarak yeni bir string seti oluşturulmaktadır ve zaman zaman yeni bir parça denenmektedir. Rastgele olmalarına rağmen, genetik algoritmalar basit bir rastgele yürüyüş değildir. Aksine, tarihsel bilgileri verimli bir şekilde kullanarak, iyileştirilmiş performans beklentisiyle yeni arama noktaları üzerinde spekülasyon yapmaktalar.

Genetik algoritmalar, doğanın uyarlanma süreçlerini taklit ederek, potansiyel çözümleri sürekli iyileştirmektedir ve kademeli evrimi yansıtmaktadır. Her yinelemede, çeşitli organizmalara benzer şekilde yeni bir potansiyel çözümler nesli oluşturulmaktadır. Bu çözümler, genetik bilgiyi içeren semboller dizileri olarak kodlanmıştır. Her çözüm, önceden belirlenmiş bir uygunluk işlevine göre değerlendirilmektedir, bu da doğal seçilimi andırmaktadır. Uygun olan bireyler seçilerek, çaprazlama ve mutasyon adımlarıyla çocuk çözümler üretilmektedir ve popülasyona çeşitlilik katılmaktadır. Seçim, üreme ve varyasyonun etkileşimiyle nesiller boyunca popülasyon, sofistifike ve güçlü çözümlere doğru evrimleşir. Bu süreç, tatmin edici bir çözüm bulunana veya durma kriteri karşılanana kadar devam etmektedir.

Tekrardan Goldberg'e (1989) göre genetik algoritmalar birçok farklı disiplinde zengin uygulamalara sahiptir. Genetik algoritmalar matematik, tıp, mühendislik ve siyaset bilimi gibi

çeşitli alanlarda kullanılmaktadır. Özünde, genetik algoritmalar, doğanın kendi zekasından ve düzeninden ilham alan akıllı stratejiler kullanarak çeşitli alanlarda esneklik ve etkinlikle en karmaşık zorluklarla başa çıkmak için hesaplamalı evrimin harikaları olarak durmaktadır. Bu yetenekleri, çeşitli alanlarda, optimizasyon ve arama alanından makine öğrenimine kadar uygulama alanı bulmaktadır.

2.2. TEMEL KAVRAMLAR

Genetik algoritma, doğal seleksiyon ve genetikle ilgili kavramları taklit eden bir optimizasyon algoritmasıdır. Bir problemin en iyi çözümünü bulmak için bu algoritma, popülasyonlar üzerinde iteratif bir şekilde çalışır. Temel kavramlar arasında "bireyler" (çözüm adayları), "popülasyon" (çözüm adaylarının toplamı), "genler" (bireylerin özelliklerini belirleyen parametreler), "mutasyon" (genetik materyalin rastgele değiştirilmesi) ve "çaprazlama" (bireyler arasında genetik materyalin değiştirilmesi) bulunmaktadır. Bu algoritma, problemin çözümünü bulmak için bireyler arasında uygunluk (fitness) değerlendirmesi yaparak, en uygun çözüme doğru evrimleşmektedir. Genetik algoritmanın gücü, büyük ve karmaşık arama alanlarında bile etkili çözümler bulabilmesidir.

2.2.1. Genler

"Gen" terimi Genetik Algoritmanın temeli ve başlangıcıdır. Genetik algoritmaların alanında, bu varlıklar verilen bir problemin potansiyel ya da olabilecek çözümlerini temsil etmektedir. Her bir varlığın önerilen çözümü, genellikle kromozom olarak adlandırılan bir dizi sembol içinde kapsamaktadır. Bu semboller, çözüm için gerekli çeşitli özellikleri içermektedir. Bu kromozom yapısı çerçevesinde, her belirgin eleman veya konum bir genden ibarettir. Özünde, genetik algoritmaların alanında, bir gen bir varlığın çözümünün temelini oluşturan belirli bir öğeyi veya özelliği kapsamaktadır.

Aşağıdaki gibi, bir kare bir geni temsil etmektedir.

1	2	3	4	8	...	2	3	5	...
---	---	---	---	---	-----	---	---	---	-----

Tablo 2.1 Genler

2.2.2. Kromozomlar

Biyolojide belirtildiği gibi de kromozomlar bir dizi genlerden oluşmaktadır. Aynı şekilde, GA kromozomları genetik bilgi depolamaktadır, düzenli gen dizileri aracılığıyla olası

çözümleri kodlamaktadır. Her gen, çözümle ilgili belirli bir özellik veya karakteristik temsil etmektedir ve kromozomun yapısını ve içeriğini şekillendirmektedir.

2	5	3	7	8	...	6	8	1	...
---	---	---	---	---	-----	---	---	---	-----

Tablo 2.2 Kromozom

2.2.3. Popülasyonlar

Popülasyonlar, kromozomların koleksiyonudur ya da gruplamasıdır. Gupta'nın (2021) açıklamasına göre, algoritma bir çözüm popülasyonu ile başlamaktadır (kromozomlar tarafından temsil edilmektedir). Yeni bir popülasyon oluşturmak amacıyla çözümler aranmaktadır ve uygulanmaktadır. Bu süreç, yeni popülasyonun mevcut olandan daha üstün olmasını sağlama arzusuyla yönlendirilmektedir.

Genetik algoritmada "popülasyon", bir optimizasyon problemi için benzersiz bir kombinasyonu temsil eden bireylerin toplamını içermektedir. Başlangıçta rastgele oluşturulan veya önceden belirlenmiş çözümlerle başlatılmaktadır ve seçim, çaprazlama ve mutasyon gibi süreçlerle evrim geçirerek potansiyel olarak optimal çözümlere ulaşmaktadır.

2	5	3	7	9	...	3	8	1	...
---	---	---	---	---	-----	---	---	---	-----

5	4	3	2	8	...	7	8	9	...
---	---	---	---	---	-----	---	---	---	-----

2	5	3	7	9	...	6	8	1	...
---	---	---	---	---	-----	---	---	---	-----

Tablo 2.3 Popülasyon

2.2.4. Uygunluk Fonksiyonu

Genetik algoritmanın uygunluk fonksiyonu, optimizasyon çabalarının temelini oluşturmaktadır. Bu fonksiyon, bireysel çözümlerin etkinliğini değerlendirmektedir ve uygunluk puanları atmaktadır, böylece seçim sürecini yönlendirmektedir. Nesiller boyunca popülasyonu daha optimal çözümlere doğru itmektedir, evrimsel süreci yönetmektedir. Uygunluk fonksiyonunun dikkatli kullanımı, problem alanını doğru yansıtarak çözüm alanının keşfi ve kullanımı arasında denge sağlamaktadır.

Busetti (2007), her çözülecek problem için bir uygunluk fonksiyonunun geliştirilmesi gerektiğini belirtmektedir. Belirli bir kromozom verildiğinde, uygunluk fonksiyonu tek bir sayısal "uygunluk" veya "hak / başarı değeri" döndürmektedir. Bu değer, o kromozomun temsil ettiği bireyin "kullanışlılığı" veya "yeteneği" ile orantılı olmalıdır. Birçok problem için, özellikle fonksiyon optimizasyonunda, uygunluk fonksiyonunun sadece fonksiyonun değerini ölçmesi yeterlidir.

Uygunluk fonksiyonunun her problem için özel olarak tasarlanması gerektiği anlaşılmaktadır. Bu fonksiyon, belirli bir kromozomun performansını değerlendirerek tek bir sayısal değer üretmektedir. Bu değer, kromozomun temsil ettiği bireyin problem çözmedeki etkinliğini yansıtmalıdır. Uygunluk fonksiyonunun doğru bir şekilde tasarlanması, algoritmanın başarısını ve problemin etkin bir şekilde çözülmesini sağlamaktadır.

Kinney ve Angeline (1994), popülasyonun evriminin tüm "sınır koşullarını" ve uygunluk fonksiyonundaki ince hataları "acımasızca kullanacağını" vurgulamaktadır ve bunu tespit etmenin tek yolu algoritmayı çalıştırmak ve ortaya çıkan kromozomları incelemektir. Özetle, uygunluk fonksiyonu, genetik algoritmayı etkili çözümlerin keşfi için yönlendiren bir rehber ışık olarak işlev görmektedir, çeşitli gerçek dünya problemleri üzerinde etkili çözümler bulmaya doğru genetik algoritmayı yönlendirmektedir.

2.2.5. Elitizm

Elitizm, genetik algoritmanın çerçevesinde, en iyi performans gösteren bireylerin bir alt kümesini nesiller arasında koruyan bir stratejidir. Bu yaklaşım, genetik işlemler sırasında oluşabilecek potansiyel kayıplara karşın en iyi çözümlerin devamını sağlamaktadır. Temel olarak, elitizm, en yüksek uygunluk puanlarına sahip bir grup bireyin belirlenmesini içermektedir. Coley (1999) belirttiği gibi en uygun birey diğerlerinden çok daha, çok daha uygun değilse bazen seçilmeyebilir. Seçilmemek, ölmek anlamına gelmektedir.

Özetle, genetik algoritma içinde elitizm, bugüne kadar ortaya çıkarılan en iyi çözümleri koruma mekanizması olarak hizmet etmektedir ve böylece algoritmanın verimliliğini ve dayanıklılığını güçlendirmektedir. Bu, kurulu en iyi çözümleri kullanma ve optimizasyon arayışını sürdürmek için bilinmeyen bölgelere doğru keşiflerde bulunma arasında hassas bir dengeyi temsil etmektedir.

2.3. GENETİK ALGORİTMANIN BASİT BİR ŞEMASI

Gupta (2021), genetik algoritmaların nasıl çalıştığına dair basit bir şekilde aşağıdaki gibi adımları listelenmiştir ve açıklanmıştır:

1. Başlangıç:

Bir başlangıç popülasyonu oluşturulmaktadır. Bu popülasyon genellikle rastgele oluşturulmaktadır ve birkaç bireyden binlerce bireye kadar değişebilmektedir.

2. Değerlendirme:

Sonra test edilmektedir. Popülasyondaki her bireye bir "uygunluk" sıralaması verilmektedir. Egzersizin önemi, ideal kriterlerimize ne kadar iyi uyduğuna göre ölçülmektedir. Bu özellikler, "daha hızlı algoritmalar daha iyidir" gibi basit olabileceği gibi, "daha dayanıklı malzemeler tercih edilmektedir, ancak çok ağır olmamalıdır" gibi daha karmaşık da olabilmektedir.

3. Seçim:

Popülasyonun fiziksel sağlığını iyileştirmeye devam edilmektedir. Bunun için filtreleme yardımıyla, popülasyondaki zayıf tasarımları kaldırılmaktadır ve en iyi bireyleri bırakılmaktadır. Birçok seçim yaklaşımı vardır, ancak temel fikir aynıdır: Daha uygun bireylerin bir sonraki nesil için seçilme olasılığını artırmak.

4. Üreme:

Seçilen bireyler, genetik kesişme ve mutasyon gibi genetik operatörler aracılığıyla bir sonraki neslin çocukları oluşturmak için kullanılmaktadır.

Çaprazlama: Seçilen bireylerin bileşenlerini birleştirerek çaprazlama sırasında yeni bireyler oluşturulmaktadır. Bu, doğada cinsiyetin nasıl davrandığına benzemektedir. Amaç, bu özellikleri iki veya daha fazla kişiden birleştirerek, her iki ebeveynden en iyi özellikleri miras alacak daha "uygun" bir nesil yaratmaktır.

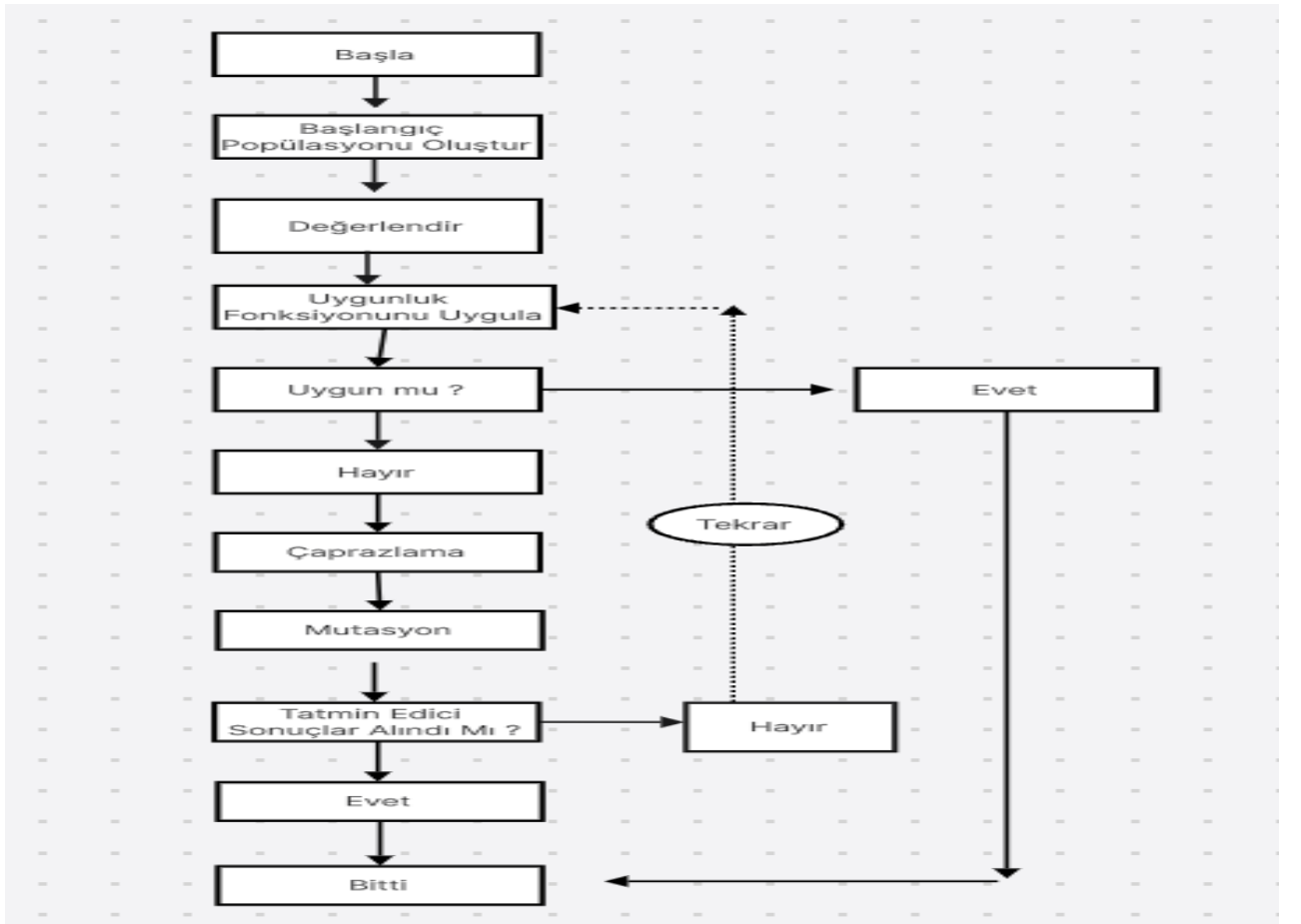
Mutasyon: Popülasyonun genetiğine bir miktar rastlantısallık eklemek gerekmektedir, aksi halde herhangi bir olası cevap başlangıç popülasyona dahil olmaktadır. Mutasyon genellikle bir kişinin genetik kodunda rastgele küçük değişiklikler yaparak çalışmaktadır.

5. Ve tekrar:

Bir sonraki nesili ele aldıktan sonra, süreci tekrarlamak için ikinci aşamaya geri dönülmektedir ve sonlandırma durumuna ulaşana kadar işlemi tekrarlanmaktadır.

6. Sonlandırma:

Genetik algoritmanın bir çözüm arayışını çeşitli nedenlerle sonlandırılmaktadır. Amaç, algoritmanın tatmin edici bir çözüm bulduğunda ve belirli minimum gereksinimleri karşıladığında, sonlandırmaktır.



Şekil 2.1 Genetik Algoritmanın basit şeması

2.4. GENETİK ALGORİTMANIN ÇÖZÜM TÜRLERİ

2.4.1. Kodlama Türleri

Genetik algoritmalarda (GA'ler), kodlama süreci bir problemin potansiyel çözümlerini belirli bir formatta temsil etmeyi içermektedir. Kodlama seçimi, GA'nın çözüm uzayını ne kadar etkili bir şekilde keşfettiğini ve optimal veya yaklaşık optimal çözümlere nasıl yakınsadığını büyük ölçüde etkilemektedir. Genetik Algoritmaya Giriş (Deepa & Sivanandam, 2008) kitabından alınan yaygın genetik algoritma kodlama türleri aşağıdaki gibi listelenmiştir ve açıklanmıştır:

- İkili Kodlama: Bu yöntem çözümleri ikili rakamların (0'lar ve 1'ler) dizisi olarak temsil etmektedir. Çözümlerin ikili dizeler olarak temsil edilebildiği problemler için uygundur.

Kromozom 1	1000 1111
Kromozom 2	1111 0001

Tablo 2.4 İkili Kodlama

- Değer Kodlama: Değer kodlamasında, her kromozom belirli bir değerler dizisini temsil etmektedir ve bu değerler, problem bağlamında çeşitli tiplerde olabilir. Bu kodlama, özel problemler için en iyi sonuçları üretebilmektedir, ancak bazen probleme özgü yeni genetik operatörler geliştirmek gerekebilmektedir.
- Permütasyon Kodlama: Her kromozom, bir dizideki sayıyı temsil eden sayı dizisidir. Genetik işlem tamamlandıktan sonra bazen düzeltmeler yapılması gerekmektedir. Permütasyon kodlamasında, her kromozom bir dizideki sayıyı temsil eden tamsayı/gerçek değerler dizisidir.
- Hekzadesimal Kodlama : (0-9, A-F) numaraları kullanarak kodlanmaktadır. Örneğin: Kromozom 1 = 3DBA ; Kromozom 2 = 9CE6
- Oktal Kodlama : (0-7) numaraları kullanarak kodlanmaktadır. Örneğin: Kromozom 1 = 05234561
- Ağaç Kodlama : Bu kodlama, genetik programlama için program ifadelerinin evrimini sağlamak amacıyla kullanılmaktadır. Her kromozom, bir programlama dilinin fonksiyonları ve komutları gibi bazı nesnelere ağacıdır.

Uygun bir kodlama türünün seçimi, problem özellikleri, çözüm uzayı yapısı ve hesaplama düşünceleri gibi faktörlere bağlıdır. Bir verilen problemin genetik algoritmalar bağlamında en etkili kodlama şemasını belirlemek için deney ve analiz hayati öneme sahiptir.

2.4.2. Başlangıç popülasyonun Oluşması

Genetik Algoritmalar (GA'lar) içinde, popülasyonun başlatılması, verilen problem için bir spektrum olası çözümlerin düşünüldüğü kritik bir aşamayı işaret etmektedir. Bu ilk topluluk, genetik algoritmanın iyileştirme ve gelişme sürecinin iteratif yolculuğu için temeli atmaktadır. Shyalika (2019, <https://medium.datadriveninvestor.com/population-initialization-in-genetic-algorithms-ddb037da6773>, Erişim Tarihi : 02.06.2024) yazdığı makalede, başlangıç popülasyonu oluşturmak için iki metodoloji tiplerinden bahsetmektedir. Shyalika'ya (2019, Erişim Tarihi : 02.06.2024) göre metodoloji tipleri aşağıdaki gibi listelenmiştir ve açıklanmıştır:

- Rastgele Başlatma: Başlangıç popülasyonunu tamamen rastgele çözümlerle doldurulmaktadır.
- Sezgisel Başlatma: Başlangıç popülasyonunu, problem için bilinen bir sezgisel yöntemi kullanarak doldurulmaktadır.

Başlangıç popülasyonunun seçimi, problem alanı, içsel özellikleri, mevcut veriler ve hesaplama kapasiteleri gibi bir dizi faktöre bağlıdır. Genellikle Genetik Algoritmaların içindeki belirli bir problem için en uygun yaklaşımı belirlemek için deneysel keşif ve analiz gereklidir.

2.4.3. Değerlendirme

Busetti (2007), değerlendirme fonksiyonu veya hedef fonksiyonunun, belirli bir parametre setine göre performans ölçüsü sağladığını belirtmektedir. Uygunluk fonksiyonu, bu performans ölçüsünü üreme fırsatlarının dağılımına dönüştürmektedir. Bir parametre setini temsil eden bir dizgenin değerlendirilmesi, diğer herhangi bir dizgenin değerlendirilmesinden bağımsızdır.

Değerlendirme fonksiyonu, algoritmanın etkinliğini analiz etmek için kullanılmaktadır. Uygunluk fonksiyonu ise, daha yüksek performans gösteren parametre setlerine öncelik vermektedir, böylece evrimsel süreci yönlendirmektedir ve genel verimliliği artırmaktadır. Her bir dizgenin değerlendirilmesi, bağımsızdır ve bu algoritmanın doğruluğunu artırmaktadır.

2.4.4. Cross Over İşlemi

Genetik algoritmalarında, çaprazlama bireyler arasında genetik bir alışveriş gibidir. Doğada ebeveynlerden çocuklara özelliklerin aktarılması gibi, çaprazlama iki ebeveyn çözümün genetik bilgisini birleştirerek yeni, potansiyel olarak daha iyi çözümler oluşturmaktadır. Farklı ebeveynlerin en iyi özelliklerini karıştırarak ve eşleştirerek, gelişmiş özelliklere sahip yavruları ya da çocukları üretmek gibidir.

Kromozom	Eski Hali	Yeni Hali
Kromozom 1	1111 (0001)	1111 (0011)
Kromozom 2	1100 (0011)	1100 (0001)

Tablo 2.5 CrossOver

2.4.5. Cross Over Türleri

Genetik algoritmalar, doğal seçim süreçlerinden esinlenerek yeni çözümler oluşturmak için çaprazlama işlemini kullanmaktadır. Çaprazlama, ebeveyn çözümlerden gelen genetik bilgiyi birleştirerek, potansiyel olarak daha iyi çözümler üretmektedir. Bu işlem, bireyler arasında genetik bilgi alışverişini yaparak yeni nesillerin oluşmasını sağlamaktadır. Çaprazlamanın temel amacı, çözüm alanında çeşitlilik sağlamak ve potansiyel olarak daha iyi çözümleri keşfetmek için genetik materyali karıştırmaktır. Bu nedenle, farklı çaprazlama teknikleri farklı stratejiler kullanmaktadır. Kora ve Yadyapalli (2017) tarafından paylaşılan değerlendirme makalede cross over türlerini aşağıdaki gibi listelenmiştir ve kısaca açıklanmıştır:

- Tek Nokta Çaprazlama: Bir çaprazlama noktası rastgele seçilmektedir ve bu noktadan sonraki bitler ebeveynler arasında değiştirilmektedir
- N Nokta Çaprazlama: Tek nokta çaprazlamaya benzemektedir, ancak birden fazla çaprazlama noktası kullanılmaktadır.
- Düzgün Çaprazlama: Genler, bir ikili maske kullanılarak bir ebeveynin herhangi birinden kopyalanmaktadır, böylece her iki ebeveynin genleri karışmaktadır.
- Üç Ebeveynli Çaprazlama: Üç ebeveyn rastgele seçilmektedir, genler eşleşirse ilk iki ebeveynin genlerinden, aksi halde üçüncü ebeveynin genlerinden seçilmektedir.

- Aritmetik Çaprazlama: Gerçel değer kodlaması için kullanılmaktadır, iki ebeveyn kromozomun değerleri lineer olarak birleştirilmektedir.
- Kısmen Eşleştirilmiş Çaprazlama (PMX): PMX, mutlak konumlara saygı gösterme eğilimindedir.
- Sıralama Çaprazlama (OX): OX, bağıl pozisyonlara saygı duyma eğilimindedir.
- Döngü Çaprazlama (CX): Permutasyon kodlaması için kullanılmaktadır, her bir gen, diğer ebeveynle birlikte bir döngü oluşturacak şekilde birleştirilmektedir.

2.4.6. Mutasyon İşlemleri

Genetik algoritmalar, doğanın evrimsel süreçlerinden esinlenerek karmaşık problemlerin çözümüne yönelik aday çözümleri temsil eden genetik yapılarla işlemektedir. Bu yapılar genellikle çözümün belirli yönlerini veya özelliklerini temsil eden bir dizi gen olarak düşünülmektedir.

Mutasyon, bu genetik yapıları değiştirerek genetik çeşitliliği artıran önemli bir mekanizmadır. Mitch (1999), bir genetik algoritmanın kendi mutasyon ve çaprazlama oranlarını bir arama süreci boyunca adapte etme yollarını bulmak, doğru dengeyi sağlama konusunda en umut verici perspektif olabileceğini belirtmiştir. Mutasyon süreci, her bir genin değerini belirli bir olasılıkla rastgele değiştirerek gerçekleşmektedir. Bu, mevcut çözümleri hafifçe değiştirerek yeni çözümler üretilmesine olanak tanımaktadır. Bu değişiklikler bazen daha iyi çözümlere yol açabilirken, diğer zamanlarda mevcut çözümlerin kalitesini düşürebilmektedir.

Mutasyon, genetik çeşitliliği artırmak için kullanılmaktadır. Ancak, aşırı mutasyon popülasyonu rastgele hale getirebilir ve iyi çözümleri kaybettirebilir. Tam tersi durumda ise, yetersiz mutasyon popülasyonun çeşitliliğini azaltabilir. Bu nedenle, mutasyon oranının dengeli bir şekilde ayarlanması önemlidir. Genellikle düşük bir mutasyon oranı tercih edilmektedir ve gerektiğinde artırılmaktadır. Bu, genetik algoritmanın hem keşif yapmasına hem de iyi çözümleri korumasına olanak sağlamaktadır.

3. YÖNTEM

3.1. GENETİK ALGORİTMA UYGULAMA ALANLARI

Genetik algoritmalar (GAs), çok çeşitli alanlardaki optimizasyon zorluklarıyla başa çıkmada uyumlulukları ve etkinlikleri nedeniyle geniş kapsamda övgü kazanmıştır. Kanade (2023, <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>, Erişim Tarihi : 02.06.2024), yazdığı makalede genetik algoritmaların ustalıklarını sergilediği birkaç önemli alanlardan kısaca listelemiştir ve açıklamıştır:

- Optimizasyon Problemleri: Genetik algoritmalar (GAs), seçim, çaprazlama ve mutasyon gibi genetik operatörleri kullanarak tekrarlayan evrim süreçleriyle geniş çözüm alanlarını keşfetmektedir, optimizasyon zorluklarını çözmeye üstün performans göstermektedir.
- Robotik Sistemler: Robotikte, genetik algoritmalar hareket planlaması, yol optimizasyonu, yol çizgisi üretimi ve kontrol sistemlerini hassas bir şekilde ayarlayarak, işletme verimliliğini ve hassasiyetini artırmaktadır.
- Finansal Modelleme: GAs, finansal modellemede portföy tahsisini, algoritmik ticareti ve risk yönetimini optimize etmektedir, karmaşık finansal sistemlere esnek bir yaklaşım sunmaktadır.
- Veri Analitiği ve Makine Öğrenimi: Genetik algoritmalar, veri analitiği ve makine öğrenimi alanında, özellik seçimi, parametre optimizasyonu, model iyileştirme ve hacimli veri kümeleri içinde desenlerin belirlenmesi gibi görevlerde vazgeçilmezdir, bu da eyleme geçirilebilir içgörüler çıkarmada yardımcı olmaktadır.
- Görüntü ve Sinyal İşleme: GAs, önemli bilgileri korurken görüntü ve sinyal işleme görevlerini optimize etmektedir, yeniden yapılandırma, gürültü giderme ve özellik çıkarma algoritmalarını iyileştirmektedir.

Kanade (2023, <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>, Erişim Tarihi : 02.06.2024), makalede Tesla'nın otonom sürüş görevleri hakkında da bahsetmiştir. Otonom sürüş teknolojilerinde genetik algoritma uygulanmıştır ve bu algoritmalar sinir ağlarını optimize etmektedir ve ayarlamaktadır. Makalede ayrıca

Amazon'un lojistik operasyon örneği değinmiştir. Amazon siparişleri karşılamayı ve lojistik operasyonlarını optimize etmek için genetik algoritma kullanmıştır. Bu şekilde, Amazon'un tedarik zincirini düzleştirir ve verimliliğini artırmaktadır (Kanade, 2023, <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>, Erişim Tarihi : 02.06.2024).

Ayrıca popüler ve duyulan başka uygulama alanları da aşağıdaki gibi eklenmiştir :

- Ulaşım ve Lojistik: Taşımacılık ve lojistik sektörü, araç rotalarını optimize etmek, filo yönetimini akılcılaştırmak, trafik sıkışıklığını hafifletmek ve verimli taşıma ağları tasarlamak için GA'ların gücünden yararlanır, bu da operasyonel etkinliği artırır ve maliyetleri azaltmaktadır.
- Yenilenebilir Enerji Optimizasyonu: GA'lar, rüzgâr türbinleri ve güneş panelleri gibi yenilenebilir enerji kaynaklarının yerleşimini ve işletimini optimize etmede kilit rol oynamaktadır, enerji üretimini en üst düzeye çıkarmaktadır ve yenilenebilir enerji sistemlerinin sürdürülebilirliğini güçlendirmektedir.
- Oyun Geliştirme ve Yaratma

Genetik algoritmalar, optimizasyon problemlerinden makine öğrenimi ve robotik alanına kadar geniş bir yelpazede kullanılmaktadır. Bu algoritmalar karmaşık, doğrusal olmayan ve çok modlu optimizasyon problemleriyle başa çıkabilirler ve minimum problem bilgisine ihtiyaç duymazlar. Ancak, büyük popülasyonlar ve karmaşık problemler için hesaplama açısından maliyetli olabilirler ve en iyi sonucu garanti etmezler. Parametre ve operatör seçimi performansı önemli ölçüde etkilemektedir. Ancak, tekrarlayan adımlarla genetik algoritmalar geniş bir problemler yelpazesinde optimal veya optimal yakın çözümlere doğru yakınsarlar. Sonuç olarak, genetik algoritmaların geniş bir yelpazede kullanılması, karmaşık optimizasyon zorluklarını aşmalarının ve yeni çözümler sunmalarının belirgin bir göstergesidir.

3.2. PYTHON DİLİ

Python, eski bir yazılım dilidir fakat sadece son senelerde daha çok duyulmaya başlamıştır. Python yüksek seviyede çalışan ve pratik bir programlama dilidir. Python en çok tercih edilen yazılım dillerin arasında çünkü Python dili basittir,doğaldır ve uyarlanabilir. Python dili,

kodlama alanında hem yeni başlayanlar hem de deneyimli geliştiriciler için ideal bir tercih olmaktadır. Python dilinin sözlüğü kısıtlıdır ve verbal olarak kullanılan diller aksine Python'da kelimelerin sadece bir anlamı vardır. Python'un yaygın kullanımı ve geniş standart kütüphanesi sayesinde birçok endüstri ve alana yayılmaktadır. Python, web geliştirme, veri bilimi, otomasyon, bilimsel hesaplama, oyun geliştirme, robotik, eğitim ve finansal analiz için kullanılmaktadır. Çeşitliliği ve zengin kütüphanesi sayesinde farklı sektörlerde popülerdir. Tekrarlayan görevlerden toplu işlem operasyonlarını yazmaya kadar, Python iş akışlarını kolaylaştırmaktadır ve verimliliği artırmaktadır.

Python, bilimsel hesaplamada güvenilir bir araç olarak hizmet vermektedir; Halvorsen'e (2019) göre NumPy ve SciPy gibi kütüphaneler, optimize edilmiş matematiksel algoritmalar ve fonksiyonlar sunarak sayısal analiz ve simülasyonları kolaylaştırmaktadır. Ek olarak Matplotlib ve Pandas da önemli Python paketlerdir.

Ayrıca, Python'un netliği ve erişilebilirliği, yeni başlayanlara programlama temellerini öğretmek için mükemmel bir dil olmasını sağlamaktadır. Eğitim kurumları geniş çapta, giriş seviyesi programlama kursları için Python'u birincil dil olarak kabul etmekte ve yetenekli programcıların yeni bir neslinin yetişmesine katkıda bulunmaktadır.

Özetlemek gerekirse, Python'un esnekliği, kullanıcı dostu yapısı ve geniş kütüphanesi, onu web geliştirme, veri bilimi, otomasyon ve bilimsel hesaplama gibi farklı alanlarda sağlam bir varlık haline getirmiştir. Kalıcı popülerliği ve kararlı topluluk desteği, onun yazılım geliştirme alanındaki yerini sağlamlaştırmaktadır.

3.2.1. Genetik Algoritma Python Kodu

Genetik algoritmalar, doğal seçim ve evrim sürecinden ilham alan ve genellikle optimizasyon ve arama problemlerine çözümler bulmak için kullanılan bir tür optimizasyon algoritmasıdır. Bu tezdeki tüm kodlar, genel bilgi kullanılarak ve Python kütüphaneleri ile yapay zeka araçlarının yardımıyla yazılmıştır ve tez problemine göre geliştirilmiştir. Python'da genetik bir algoritmanın basit bir örneğinin çevirisi:

```
[ import random
def generate_individual(length):
    return [random.randint(0, 1) for _ in range(length)]
```

```

def fitness(individual, target):
    return sum(1 for gene, target_gene in zip(individual, target) if gene == target_gene)

def crossover(parent1, parent2):
    crossover_point = random.randint(0, len(parent1) - 1)
    child1 = parent1[:crossover_point] + parent2[crossover_point:]
    child2 = parent2[:crossover_point] + parent1[crossover_point:]
    return child1, child2

def mutate(individual, mutation_rate):
    for i in range(len(individual)):
        if random.random() < mutation_rate:
            individual[i] = 1 - individual[i]

def genetic_algorithm(target, population_size, mutation_rate, generations):
    population = [generate_individual(len(target)) for _ in range(population_size)]
    for _ in range(generations):
        population = sorted(population, key=lambda x: fitness(x, target), reverse=True)
        if fitness(population[0], target) == len(target):
            return population[0]
        next_generation = population[:2] # Elitism: Keep the top 2 individuals
        while len(next_generation) < population_size:
            parent1, parent2 = random.choices(population[:10], k=2)
            child1, child2 = crossover(parent1, parent2)
            mutate(child1, mutation_rate)
            mutate(child2, mutation_rate)
            next_generation.extend([child1, child2])
        population = next_generation
    return None

target = [1, 0, 1, 0, 1, 0, 1, 0] # Target binary sequence to evolve towards

```

```

population_size = 100
mutation_rate = 0.01
generations = 1000
result = genetic_algorithm(target, population_size, mutation_rate, generations)
print("Result:", result)

```

Bu kod, hedef bir bit dizisine (target) doğru evrimleşen bir popülasyon oluşturmaktadır. Başlangıçta rastgele oluşturulan bireyler, genetik operatörler (çaprazlama ve mutasyon) ve elitizm kullanılarak bir sonraki nesil oluşturulmaktadır. Oluşturulan popülasyon, hedefe ulaşıldığında veya belirli bir nesil sayısı aşıldığında sona ermektedir.

3.2.2. Python'da Genetik Algoritma ile Knapsack Problemi

Knapsack problemi (çanta problemi), verilen bir ağırlık sınırlaması altında, öğelerin belirli bir değer toplamını maksimize etmeye çalışan bir optimizasyon problemidir. Genetik algoritma, bu tür bir problemi çözmek için etkili bir yaklaşım olabilmektedir. Python'da çanta problemi için genetik algoritma örneği:

```

[import random

class Item:

    def __init__(self, weight, value):

        self.weight = weight

        self.value = value

def generate_population(num_items, max_weight, population_size):

    population = []

    for _ in range(population_size):

        chromosome = [random.randint(0, 1) for _ in range(num_items)]

        population.append(chromosome)

    return population

def fitness(chromosome, items, max_weight):

    total_weight = 0

```

```

total_value = 0
for i in range(len(chromosome)):
    if chromosome[i] == 1:
        total_weight += items[i].weight
        total_value += items[i].value
if total_weight > max_weight:
    return 0
else:
    return total_value

def crossover(parent1, parent2):
    crossover_point = random.randint(1, len(parent1) - 1)
    child1 = parent1[:crossover_point] + parent2[crossover_point:]
    child2 = parent2[:crossover_point] + parent1[crossover_point:]
    return child1, child2

def mutate(chromosome, mutation_rate):
    for i in range(len(chromosome)):
        if random.random() < mutation_rate:
            chromosome[i] = 1 - chromosome[i]

def select_parents(population, items, max_weight):
    fitness_values = [fitness(chromosome, items, max_weight) for chromosome in population]
    total_fitness = sum(fitness_values)
    probabilities = [fit / total_fitness for fit in fitness_values]
    parent1 = random.choices(population, weights=probabilities)[0]
    parent2 = random.choices(population, weights=probabilities)[0]
    return parent1, parent2

def genetic_algorithm(items, max_weight, population_size, generations, mutation_rate):
    num_items = len(items)

```

```

population = generate_population(num_items, max_weight, population_size)
for _ in range(generations):
    next_generation = []
    for _ in range(population_size // 2):
        parent1, parent2 = select_parents(population, items, max_weight)
        child1, child2 = crossover(parent1, parent2)
        mutate(child1, mutation_rate)
        mutate(child2, mutation_rate)
        next_generation.extend([child1, child2])
    population = next_generation
    best_solution = max(population, key=lambda x: fitness(x, items, max_weight))
return best_solution

# Örnek kullanım
items = [Item(10, 50), Item(20, 100), Item(30, 150)] # Öğelerin ağırlıkları ve değerleri
max_weight = 50 # Çantanın maksimum taşıma kapasitesi
population_size = 100
generations = 100
mutation_rate = 0.03

best_solution = genetic_algorithm(items, max_weight, population_size, generations,
mutation_rate)

print("Best solution:", best_solution)

```

Bu kod, belirli bir ağırlık sınırlaması altında çanta problemi için genetik algoritmayı uygulamaktadır. İlgili fonksiyonlar, popülasyon oluşturma, uygunluk fonksiyonu hesaplama, çaprazlama, mutasyon ve ebeveyn seçimi gibi genetik algoritma adımlarını içermektedir. Kod, belirli bir nesil sayısı boyunca popülasyonu evrimleştirir ve en iyi çözümü bulmaktadır.

3.3. TERSANEDE LOJİSTİK SÜREÇLERİN OPTİMİZASYONU

Kim ve diğ., (2005) tasarımdan üretime kadar olan süreyi azaltmanın tek mümkün yolu bilgisayarların kullanımı olduğunu vurgulamıştır. Ayrıca, bilgisayarların yaygın kullanımı ve mevcut seviyesi çok yüksek olduğundan, simülasyon tabanlı bir gemi inşa süreci, tasarımdan işlem planlamaya ve üretime kadar olan geleneksel sıralı süreçle karşılaştırıldığında üretim süresini büyük ölçüde azaltabilmektedir. Teknolojiye ve dijitalleşmeye odaklanması tersanelerin ve lojistik süreçlerin optimizasyonuna ve gelişmelerine yol açacaktır.

Tersanelerdeki lojistik akışlarının optimizasyonuna yönelik tartışılan bazı çözümler, Spoehr, J. ve diğerlerinin (2021) robotik ve dijital tersane üzerine yazılarında belirtilmiştir.

Tersanede lojistik süreçlerin optimizasyonunun gerçekleşmesi amacıyla potansiyen olan çözümler aşağıdaki gibi listelenmiştir:

- a- RFID, barkodlama ve blok zinciri teknolojilerinin entegrasyonu, bir tersane operasyonlarında daha büyük bir görünürlük, verimlilik ve güven sağlayabilmektedir. Bu teknolojiler, gerçek zamanlı izleme, doğru veri yakalama ve güvenli işlem yönetimi sağlayarak, üretkenliği artırabilmektedir, maliyet tasarrufları sağlayabilmektedir ve müşteri memnuniyetine katkıda bulunabilmektedir. Ayrıca, bu teknolojileri kullanarak tersaneler, düzenleyici uyumluluk ve sürdürülebilirlik girişimleri gibi gelişen endüstri gereksinimlerine uyum sağlayabilmektedir.
- b- Üretimdeki robotlar ve otonom ajanlar, karmaşık süreçleri basitleştirme, üretim döngü sürelerini kısaltma, özelleştirme taleplerini karşılama, maliyetleri azaltma ve tedarik zinciri bağlantılarını iyileştirme yoluyla rekabet gücünü artırmaya uzun süredir yardımcı olmaktadır (Brettel ve diğ., 2014).

Rao (2023, <https://www.wevolver.com/article/cobot-vs-robot>, Erişim Tarihi: 02.06.2024.) tarafından paylaşılan makalede, cobotlar (collaborative robots), endüstriyel robotlardan farklı olarak, insanlarla birlikte çalışmak üzere tasarlanmış ve güvenlikleri öncelikli olduğunu belirtilmiştir. Daha esnek, adaptasyonu kolay ve daha kullanıcı dostudurlar. Endüstriyel robotlar ise genellikle daha spesifik görevlere odaklanır, daha yüksek hız ve taşıma kapasitesine sahiptirler. Ancak, kurulum ve entegrasyon süreçleri daha karmaşıktır ve genellikle daha pahalıdır.

Sonuç olarak, endüstriyel robotlar ve cobotlar, tersanelerdeki yapısal işler gibi zorlu ve tehlikeli işlerde insan işçilerin yerine geçerek üretkenliği artırırken güvenlik risklerini azaltmaktadır. Her iki teknoloji de lojistik akışları hızlandırmak ve düzenlemek gibi çeşitli uygulamalarda kullanılabilirlerdir.

- c- Metal 3D baskı, ya da diğer adıyla eklemeli üretim, gemi inşasında tasarım esnekliği, maliyet etkinliği ve tedarik zinciri dayanıklılığı sunarak devrim yaratmaktadır. Tersanelerde 3D baskı, hızlı prototipleme için kullanılarak tasarımların doğrulanması için ölçek modelleri ve bileşenlerin hızla üretilmesini ve geleneksel yöntemlerle elde edilemeyen karmaşık geometrilerin üretimini mümkün kılmaktadır. Bu teknoloji, stok ihtiyacını azaltarak ve acil tamiratlar için hemen parçaların üretimini sağlayarak gemi arıza süresini en aza indirmektedir.

Geleneksel imalatta, çelik plakalar ve kaynak teli gibi malzeme kaynaklarının temin edilmesi gerekmektedir. Bu kaynaklar, mevcut olması gereken çeşitli makineler aracılığıyla aktif olarak parça ve bileşenlere dönüştürülmektedir. Son olarak, montaj işlemi gerçekleştirilebilmektedir (Hengst, 1999).

Metal 3D baskının ana avantajları arasında maliyet ve zaman tasarrufu, malzeme israfının azaltılması, tasarım esnekliği, montaj süresinin azalması, tedarik zinciri dayanıklılığının artması ve sürdürülebilirliğin iyileştirilmesi bulunmaktadır.

- d- Otomatik yönlendirilen araç (AGV), zemindeki işaretleri veya kabloları takip eden ya da görme veya lazer teknolojisini kullanan bir mobil robottur. Bu araçlar genellikle endüstriyel uygulamalarda, üretim tesislerinde veya depolarda malzeme taşımak için kullanılmaktadır. 20. yüzyılın sonlarında otomatik yönlendirmeli araçların kullanımı genişlemiş ve artık yalnızca endüstriyel ortamlarla sınırlı kalmamıştır (Crawford 2011). Tersane lojistik akışlarını hızlandırmak ve düzenlemek için bir olası çözüm, AGV'lerin kullanımı olabilmektedir.
- e- Dijital İkiz kavramı, genellikle bir fiziksel varlığın özelliklerini ve karakteristiklerini doğru bir şekilde yansıtan dijital bir kopya olarak kavramsallaştırılmaktadır (Kritzinger ve diğ., 2018). Dijital ikizler, tersanelerde lojistik süreçleri optimize ederek malzeme stoklarını yönetmektedir, tedarik zinciri süreçlerini optimize etmektedir, iş gücü planlaması için bilgi sağlamaktadır, malzeme yükleme ve boşaltma işlemlerini optimize

etmektedir ve lojistik süreçleri gerçek zamanlı izlemektedir ve analiz etmektedir. Bu, tersanelerin daha verimli, esnek ve güvenilir olmasını sağlamaktadır, rekabet gücünü artırmaktadır ve işletme maliyetlerini düşürmektedir.

f- Dario Fedko (2021, <https://wear-studio.com/virtual-and-augmented-reality-in-logistics/>, Erişim Tarihi : 02.06.2024) tarafından yazılan makaleye göre tersanelerde Sanal Gerçeklik (VR) teknolojisi, lojistik akışları iyileştirmek için verimlilik, doğruluk ve güvenliği artırabilmektedir. Envanter yönetiminde VR, sanal depo alanları oluşturarak stok izleme ve düzenlemeyi optimize etmektedir. Malzeme taşıma ve nakliyede, işçiler sanal ortamda eğitim alarak daha yetkin hale gelir ve lojistik akışlar daha sorunsuz olmaktadır. VR, malzeme taşıma rotalarını optimize ederek seyahat mesafelerini kısaltır ve tıkanıklıkları azaltmaktadır. Ayrıca, VR tabanlı güvenlik eğitimleri, işçileri potansiyel tehlikeler ve güvenlik protokolleri konusunda eğitmektedir, bu da kazaları ve kesintileri azaltmaktadır. Böylece, tersane içi lojistik süreçler daha verimli ve maliyet etkin hale gelmektedir.

g- Tersanede taşeron çalışanların yeterliliği ve uyumluluğunun sağlanması, güvenlik standartlarının ve operasyonel verimliliğin korunması açısından son derece önemlidir. Yapılandırılmış bir yaklaşım ile eğitim kontrolleri, belirli beceri setlerini ve bilgi gereksinimlerini ele almak için oluşturulmuştur. Taşeronlarla işbirliği yapılarak, teknik beceriler, güvenlik protokolleri ve çevre düzenlemelerini içeren özel eğitim uygulamaları geliştirilmektedir. Bu uygulamalar, eğitim ve uygulamalı egzersizler de dahil olmak üzere çeşitli öğrenme stillerine uyum sağlamak için çeşitli eğitim yöntemlerinden yararlanmaktadır. Eğitim bitiminden sonra yapılan değerlendirmeler, çalışanların anlayışını ve yeterliliğini değerlendirir ve standartları karşılayanlar için sertifika verilebilir. Eğitim uygulamaları düzenli olarak gözden geçirilmesi ve güncellenmesi, değişen düzenlemeler ve endüstri en iyi uygulamalarıyla uyum sağlamayı sağlamaktadır. Eğitim faaliyetlerinin belgelenmesi, geri bildirim mekanizmalarıyla birlikte sürekli iyileşmeyi teşvik etmektedir.

h- Tersanenin düzenini kapsamlı bir şekilde inceleyerek, darboğazları, sıkışma noktalarını ve verimsizlikleri belirlemek için bir analiz yapılması gerekmektedir. Malzeme depolama alanlarını, işçi yollarını ve güzergahları değerlendirmesi lazım ve malzeme

ve işçi hareketlerindeki potansiyel engel noktalarını ve karmaşık alanları belirleyerek daha verimli ve organize bir planlama yapılmış olmaktadır.

3.3.1. RFID, Barkodlama ve Blok Zinciri

RFID (Radyo Frekans Tanımlama), barkodlama ve blok zincir teknolojilerini bir tersanede entegre etmek, işlemleri optimize etmek, tedarik zinciri görünürlüğünü artırmak ve izlenebilirliği iyileştirmek için oldukça etkili bir yol olabilmektedir. Pruyne ve diğ. (2010), RFID teknolojisinin gemi inşası faaliyetlerinde uygulanmasıyla üretkenliğin artırılmasının, dolayısıyla tersanelerin oldukça rekabetçi hale gelmesinin umut verici bir yol olabileceğini belirtmiştir. Bu gelişmiş izleme sistem teknolojisi ile envanter yönetimi ve lojistik işlemlerinde, RFID'nin gemi inşaat endüstrisinde iyi bir şekilde kullanılacağı öngörülmektedir.

RFID okuyucuları, belirli noktalara yerleştirilmektedir ve etiketlerin radyo frekanslarını kullanarak tanınmasını sağlamaktadır. Bu, varlıkların hareketini ve konumunu otomatik olarak izlemeyi mümkün kılmaktadır. Tersanede kullanılan hammaddelerin, işlenmiş parçaların ve bitmiş ürünlerin RFID etiketlenmesi, envanter yönetimini otomatikleştirir ve malzeme akışını izlemektedir. Örneğin, bir gemi parçasının üretim süreci boyunca RFID etiketi taşıması, işleme aşamalarını izlemeyi ve nihai montaj noktalarına kolayca yerleştirilmesini sağlamaktadır.

Barkodlama:

Barkodlar, parçalar, araçlar ve belgeler gibi öğelere basılabilmektedir veya yapıştırılabilmektedir. Barkod okuyucuları, barkodları tarayarak bilgileri okumaktadır ve otomatik olarak veritabanlarına kaydetmektedir. Barkodlama, envanter yönetimini, malzeme taşıma işlemlerini ve iş akışını izlemeyi kolaylaştırmaktadır. Her bir barkod, öğenin benzersiz tanımlayıcı bilgilerini içermektedir.

Blok Zinciri:

Blok zinciri, veri alışverişlerini ve işlemleri blok halinde kaydederek, değiştirilemez ve izlenebilir bir kayıt oluşturmaktadır. Czachorowski ve diğ. (2019) belirttiği gibi bu teknolojinin geniş bir uygulanabilirlik alanı bulunmaktadır; bu da tedarik zincirini daha verimli bir şekilde bağlamaya, zaman damgalı kanıtlanmış verilerin değişimini ve görünürlüğünü sağlamaya, endüstriyel işletim maliyetlerini araçlar aracılığıyla azaltmaya ve güvenliği artırmaya olanak tanımaktadır.

Bu teknolojilerin entegrasyonu, tersane operasyonlarını daha şeffaf, verimli ve güvenli hale getirebilmektedir. Bu teknolojilerin bir araya getirilmesi, gemi inşaatı sürecinde hataların azaltılmasına, üretkenliğin artırılmasına ve maliyetlerin düşürülmesine yardımcı olabilmektedir.

Bir tersane içinde RFID, barkod ve blok zincir sistemlerinin uygulanma maliyeti, tersanenin büyüklüğü, operasyonların karmaşıklığı ve seçilen özel teknolojilere bağlı olarak önemli ölçüde değişebilmektedir. Potansiyel maliyetlerin özeti aşağıdaki gibidir:

- **RFID Sistemi:** RFID etiketlerinin, okuyucuların, antenlerin ve ilişkili altyapının maliyeti değişebilmektedir. Genel olarak, RFID etiketleri, türlerine ve yeteneklerine bağlı olarak birkaç sentten birkaç dolara kadar değişebilmektedir. Okuyucuların ve antenlerin maliyeti, gereken menzil ve özelliklere bağlıdır. Orta büyüklükte bir tersane için, RFID teknolojisi için ilk yatırım birkaç bin ila on binlerce dolara kadar değişebilmektedir.
- **Barkod Sistemi:** Barkod sistemleri genellikle RFID sistemlerine göre daha az maliyetlidir. Maliyet, barkod tarayıcıları, yazıcılar, etiketler ve barkod verilerini üretmek ve yönetmek için yazılımı içermektedir. Toplam maliyet, uygulamanın ölçeği ve karmaşıklığına bağlı olarak birkaç yüz ila birkaç bin dolara kadar değişebilmektedir.
- **Blok Zinciri Entegrasyonu:** Blok zinciri teknolojisinin uygulanması, blok zinciri platformu lisanslarının, geliştirme kaynaklarının, akıllı sözleşme geliştirmenin ve sürekli desteklemenin maliyetlerini içermektedir. Orta büyüklükte bir tersane için maliyet on binlerce dolardan yüz binlere kadar değişebilmektedir.

Entegrasyon ve Uygulama Maliyetleri: Bireysel teknolojilerin maliyetlerinin yanı sıra, bu sistemlerin mevcut altyapıya entegre edilmesiyle ilgili masraflar da olacaktır. Bu, donanım kurulumunu, yazılım entegrasyonunu, personel eğitimini içermektedir.

Bakım ve Destek: Sürekli bakım ve destek maliyetleri de dikkate alınmalıdır. Bu, yazılım güncellemelerini, donanım bakımını, teknik desteği veya yazılım lisansları için tekrar eden ücretleri içermektedir.

3.3.2. Endüstriyel Robotlar/Cobotlar

Endüstriyel robotlar, verimlilikleri, hassasiyetleri ve çok yönlülükleri nedeniyle çeşitli alanlarda yoğun olarak kullanılmaktadır. Fairchild'ın (2021, <https://howtorobot.com/expert-insight/top-industries-using-robots>, Erişim Tarihi : 02.06.2024) yazdığı makalede, endüstriyel robotların başlıca uygulama alanları aşağıdaki gibi listelenmiştir ve açıklanmıştır:

- Otomotiv Endüstrisi: Otomotiv üretiminde endüstriyel robotlar, kaynak, boya, montaj, malzeme taşıma ve kalite kontrol gibi görevlerde yoğun bir şekilde kullanılmaktadır.
- Elektronik Endüstrisi: Elektronik imalatta, robotlar devre kartı montajı, lehimleme, paketleme ve test gibi görevlerde kullanılmaktadır.
- Metal İşleme Endüstrisi: Endüstriyel robotlar, havacılık, inşaat ve imalat gibi sektörlerde, işleme, kaynak, kesme, bükme ve metal parça taşıma gibi görevlerde kullanılmaktadır.
- Gıda ve İçecek Endüstrisi: Gıda işleme ve paketleme alanında robotlar, sıralama, alma ve yerleştirme, etiketleme ve paletleme gibi görevler için kullanılmaktadır.
- İlaç Endüstrisi: İlaç üretiminde endüstriyel robotlar, dozajlama, paketleme, etiketleme ve kalite kontrol gibi görevlerde önemli bir rol oynamaktadır.
- Lojistik ve Depolama: Robotlar, lojistik ve depolama alanında paletleme, sipariş toplama, sıralama ve envanter yönetimi gibi görevler için kullanılmaktadır.
- Sağlık Endüstrisi: Sağlık sektöründe, robotlar cerrahi prosedürlerde, rehabilitasyonda, ilaç dağıtımında ve hasta bakımında kullanılmaktadır.
- Plastik ve Kauçuk Endüstrisi: Endüstriyel robotlar, plastik ve kauçuk imalatında enjeksiyon kalıplama, şişirme kalıplama, ekstrüzyon ve montaj işlemleri için kullanılmaktadır.
- Tekstil Endüstrisi: Tekstil imalatında, robotlar dikiş, kesme, malzeme taşıma ve kalite kontrol gibi görevler için kullanılmaktadır.

Endüstriyel robotlar, tersanelerde gemi inşaatı ve bakımında çeşitli önemli görevler üstlenmektedir. Tersanelerde endüstriyel robotların bazı önemli kullanım alanları ve görevleri (Gupta ve diğ., 2017) :

- **Kaynak İşlemleri:** Tekrarlayan görevleri içeren ve benzer parçalarda gerçekleştirilen bir kaynak işlemi otomasyona uygun olabilmektedir. Robotlar, birden fazla eksenle kaynak içeren tekrarlayan görevler veya benzer parçalar için iyi çalışmaktadır. Otomatik kaynağın en belirgin avantajları hassasiyet ve verimliliktir. Robotlar, birden fazla eksenle kaynak içeren tekrarlayan görevlerde iyi çalışmaktadır. (Gupta ve diğ., 2017).
- **Boyama İşlemleri:** Robotların popüler ve verimli bir başka kullanım alanı ise spreylere boyama alanındadır. Bir robotun hareketinin tutarlılığı ve tekrarlanabilirliği, neredeyse mükemmel kaliteyi sağlarken aynı zamanda boya israfını önlemiştir. Sprey boyama uygulamaları, insan operatörü tehlikeli fakat beceri gerektiren bir işten kurtarıırken, aynı zamanda iş kalitesini, bir örnekliği artırarak ve maliyetleri düşürerek, robotik uygulamaların doğru kullanımını somutlaştırmaktadır. (Gupta ve diğ., 2017).
- **Kaldırma ve Taşıma:** Makine yükleme ve boşaltma, temel malzeme taşıma işlemlerinden daha karmaşık bir uygulamadır; bu uygulama için robot, hem manipülatif hem de taşıma yetenekleri sağlamaktadır. (Gupta ve diğ., 2017).
- **Yüzey Hazırlığı ve Temizleme:** Gemi yapılarının yüzeylerinin hazırlanması ve temizlenmesi, boyama ve kaplama işlemleri öncesinde önemlidir. Endüstriyel robotlar, yüksek basınçlı su jetleri veya özel fırçalarla gemi yüzeylerini temizleyebilmektedir.
- **Kesme ve Şekillendirme İşlemleri:** Robotlar, çekiç dövme işlemleri, hadde dövmeleri, sıcak şekillendirme presleri ve çekme tezgahı uygulamaları gibi birçok farklı dövme uygulamasında kullanılmaktadır (Gupta ve diğ., 2017).
- **Basınçlı döküm,** demir dışı metallerin kalıplara yüksek basınç altında zorlanarak istenilen şekillerde parçalar oluşturulmasını içermektedir. (Gupta ve diğ., 2017).

Endüstriyel robotlar, tersane ortamlarında güvenli bir şekilde çalışabilecek şekilde tasarlanmaktadır ve genellikle suya dayanıklıdır. Ayrıca, robotların gemi yapısının farklı alanlarına kolayca erişebilmesi için esnek montaj sistemleri kullanılmaktadır. Bu sayede, gemi

inşaat ve bakım süreçlerinde verimlilik artırılır ve insan işçilerin maruz kaldığı riskler azaltılmaktadır.

Endüstriyel robotların fiyatları, taşıma kapasitesi, hassasiyeti, hızı ve diğer özel gereksinimlere bağlı olarak önemli ölçüde değişmektedir. Ortalama olarak, endüstriyel robotlar on binlerden yüz binlere kadar fiyatlanabilir. İşbirlikçi robotlar (cobots) genellikle daha düşük fiyatlıdır ve toplam sahip olma maliyeti için bakım, eğitim ve entegrasyon gibi sürekli giderler de dikkate alınmalıdır.

3.3.3. 3D Metal Printing

3D metal baskı teknolojisinin tersanelerde uygulanması çeşitli maliyet faktörlerini içermektedir. Başlıca maliyet bileşenleri ekipman maliyetleri, malzeme maliyetleri, personel maliyetleri, işleme ve finisaj maliyetleri, yazılım ve tasarım araçları maliyetleri ve düzenleyici uyum maliyetleridir. Maliyetler, başlangıç yatırımı, tesis kurulumu, malzemeler, personel eğitimi, işçilik, ek işleme ekipmanları ve yazılım gereksinimlerine bağlı olarak değişir. Maaliyet, tersanenin büyüklüğüne, uygulama kapsamına ve yerel koşullara bağlıdır.

3.3.4. Automated Guided Vehicle (AGV)

Otomatik Rehberli Araçlar (AGV'ler), tersanelerde katma değerli bileşenler olarak ortaya çıkmıştır, artan verimlilik, güvenlik güçlendirme ve üretkenlik artışı gibi pek çok fayda sunmaktadır. Bu alan içinde AGV'ler, geniş tersane sahaları boyunca ağır malzemelerin, bileşenlerin ve ekipmanın sorunsuz taşınması için kullanılmaktadır.

Tekrarlayan görevler için AGV'lerin uygulanmasıyla, iş gücü daha zorlu rollerde daha iyi kullanılabilir (Ullrich, 2015). Manyetik navigasyon sistemine sahip jiroskop tabanlı AGV'ler, pozisyonlama hassasiyetinde ± 5 mm'ye kadar doğruluk sağlayabilmektedir (Ullrich,2015). Ayrıca, AGV'ler genel olarak personel maliyetini azaltabilmektedir ve malzemenin sadece taşınması için ayrılmış daha az personel olmasıyla da maliyeti düşürebilmektedir (Ullrich, 2015).

Tersanelerde AGV'lerin uygulanması, entegrasyonun karmaşıklığı ve ilişkili fiyat dinamikleri üzerinde önemli bir etkiye sahip olan bir dizi faktör tarafından etkilenmektedir. Genel olarak, bir AGV biriminin temin maliyeti, özellikle AGV'nin özellikleri ve gereksinimlerine bağlı olarak birkaç bin dolardan bir yüz bin dolara kadar değişebilir. Ancak, bu maliyetlerin ötesinde, özelleştirme, entegrasyon, bakım, personel eğitimi ve altyapı geliştirme gibi ek masraflar da

dikkate

alınmalıdır.

AGV'lerin Maaliyeti

AGV Tipi	Her Agv (Manyetik veya Optik)	Her Agv (Lazer veya doğal)	Agv Yazılım Sistemi	Program Kurulumu
AGC- Tek Yön yedekleme 1 Ton	14000 USD	25000 USD	10000 USD – 40000 USD Arasında	0 USD – 50000 USD Arasında
AGC- İki Yön yedekleme 1 Ton	23000 USD	30000 USD		
AGC- Dönen Yön yedekleme 1 Ton	18000 USD	27000 USD	Ortalama – 20000 USD	Ortalama- 20000 USD
AGC – Her yöne Dönen kaldırma 1 ton	-	35000 USD		
1 ton tugger AGV	20000 USD	28000 USD		
3-ton tugger AGV	30000 USD	30000 USD		
3-ton tugger AGV manüel traktör	45000 USD	54000 USD		

Şekil 3.1 AGV'lerin Maaliyeti (<https://www.agvnetwork.com/agv-cost-estimation-how-much-does-an-automated-guided-vehicle-cost> sitesinden alınmıştır ve çevrilmiştir, Erişim Tarihi : 02.06.2024)

AGV'lerin Maaliyeti

AGV Tipi	Her Agv (Manyetik veya Optik)	Her Agv (Lazer veya doğal)	Agv Yazılım Sistemi	Program Kurulumu
Palet Taşıyıcılar 1 ton, Kaldırma yüksekliği 2 m	-	70000 USD	60000 USD	60000 USD
Karşı Ağırlıklı 2 ton, Kaldırma yüksekliği 2 m	-	85000 USD		60000 USD
VNA 1 ton, Kaldırma yüksekliği 10 m	-	180000 USD		100000 USD

Şekil 3.2 AGV'lerin Maaliyeti (<https://www.agvnetwork.com/agv-cost-estimation-how-much-does-an-automated-guided-vehicle-cost> sitesinden alınmıştır ve çevrilmiştir, Erişim Tarihi : 02.06.2024)

3.3.5. Digital Twin

Tersanede bir Dijital İkiz uygulamanın maliyeti, tersane operasyonlarının karmaşıklığı, Dijital İkiz'de gereken detay ve işlevsellik düzeyi, mevcut sistemlerle entegrasyon ve gereken özelleştirme miktarı gibi çeşitli faktörlere bağlı olarak geniş çapta değişebilmektedir.

<u>Sirket Boyutu</u>	<u>Maaliyer</u>	<u>İş Gücü</u>
<u>Küçük Boyut Şirketi</u>	20000-45000 USD	50-200
<u>Orta Boyut Şirketi</u>	50000-75000 USD	200-1000
<u>Büyük Boyut Şirketi</u>	75000-90000 USD	1000-10000
<u>Digital Twin Dizaynın Maaliyeti</u>	10000-20000 USD	
<u>Yazılım Sistemin Maaliyeti</u>	35000-65000 USD	
<u>Yerleştirme Maaliyeti</u>	15000 USD	
<u>Entegrasyon Maaliyeti</u>	10000 USD	
<u>Bakım Maaliyeti</u>	1500-4000 USD	
<u>Proje Menajeri Maaliyeti</u>	25USD/Saatte	
<u>Testler</u>	5000 USD	

Şekil 3.3 Digital Twinning Uygulama Maaliyetleri (<https://risingmax.com/blog/digital-twin-development-cost> sitesinden alınmıştır ve çevrilmiştir, Erişim Tarihi : 02.06.2024)

3.3.6. Virtual Reality (Virtual Reality)

Sanal gerçeklik (VR) teknolojisinin tersaneye uygulanması çeşitli maliyetleri içermektedir :

- Donanım: VR başlıkları ve çevre birimlerinin maliyeti.
- Yazılım Geliştirme: Tersane süreçleri için özel VR yazılımı.
- Entegrasyon: Varolan sistemlere VR'nin entegrasyonu.
- Eğitim ve Destek: Personelin eğitimi ve teknik destek sağlanması.
- Altyapı
- İçerik Oluşturma: 3D modeller ve içerik geliştirme.
- Bakım: Sürekli bakım ve güncellemeler.
- Danışmanlık: Değerlendirme ve uygulama için uzmanlarla işbirliği yapılması.

3.3.7. Eğitim Uygulamaları

Eğitim gereksinimlerinin tutarlı bir şekilde uygulanmasıyla, bir güvenlik ve uyumluluk kültürü korunmaktadır ve tersane ortamında hem personeli hem de varlıkları korunmaktadır. Herhangi bir türden hoş olmayan durumları dışlamak ve proaktif bir yaklaşıma sahip olmak için iki taraf arasındaki karşılıklı güven ilişkisine ek olarak çok sıkı ve önleyici tedbirler almak son derece önemlidir. Hicri Ercili Tersanesi'nde de eğitimlere önem verilmektedir ve gerekli doğrulamalar yapılmaktadır.

Tersanelerde çalışanların eğitimini artırmak için geliştirilecek bir yazılım uygulaması, iş sağlığı ve acil durum bilgileri ile teknik bilgileri içerecektir. Bu uygulama, yeni işçilerin adaptasyonunu kolaylaştırırken temel eğitim metinlerini kontrol edebilme imkanı sağlayacaktır. Mobil uygulamaların kullanımı, eğitim metodolojilerini ve iletişim kanallarını geliştirerek operasyonel ortama etkin bir şekilde yanıt verme kapasitesini artıracaktır. Uygulamalar, görsel öğrenme kaynakları, anlık bildirimler ve sistem entegrasyonu ile sürekli öğrenmeyi teşvik edecek ve güvenlik standartlarını ve operasyonel verimliliği artıracaktır.

Tersane gibi ortamlarda eğitimler elzemdir. Eğitimlere yoğunlaşmak, gelecekte hata ve kaza payını azaltmak anlamına gelmektedir. Dolaylı ve dolaysız yoldan, bir hata veya bir kaza olduğunda iş süreçleri aksatılmaktadır. Böylelikle zamana ve maliyete yansıtılmaktadır. Eğitimli bir çalışan daha verimli çalışmaktadır. Bu şekilde tersaneler verimli çalışmaktadır.

3.3.8. Darboğazları / Şıkışma noktaları İncelenmesi

Tersanelerde verimlilik ve üretkenliği engelleyen darboğazlar, entegrasyon zorlukları, sınırlı rihtım kullanılabilirliği, verimsiz tesis düzenlemeleri ve düzenleyici uyum gecikmeleri gibi çeşitli sorunlardan kaynaklanmaktadır. Bu sorunlara çözüm olarak stratejik planlama, teknoloji ve altyapı yatırımları, işgücü eğitimi ve proje yönetimi iyileştirmeleri gerekmektedir. Darboğazların tespiti ve çözümü süreci, veri toplama, analiz, kök neden analizi ve çözüm geliştirme aşamalarını içermektedir ve birkaç haftadan birkaç aya kadar sürebilir. Bu süreç, verimliliği ve iş akışını artırmayı hedeflemektedir.

4. BULGULAR

4.1. HİCRİ ERCİLİ TERSANESİ

Hicri Ercili Şirketinden ve sitesinden (<https://www.hicriercili.com.tr/>, Erişim Tarihi : 02.03.2024) alınan bilgilere göre, 1941 yılında Bandırma'da doğan Hicri Ercili şirketi kurmuştur. 1980'lerin sonunda, Türkiye'nin küresel ekonomiye adapte olma sürecinde, Hicri Ercili'nin iş stratejileri bu değişimlere uyum sağlamıştır. Hicri Ercili Tersanesi ile alakalı tüm bilgiler Hicri Ercili şirketinden ve resmi sitesinden alınmıştır.

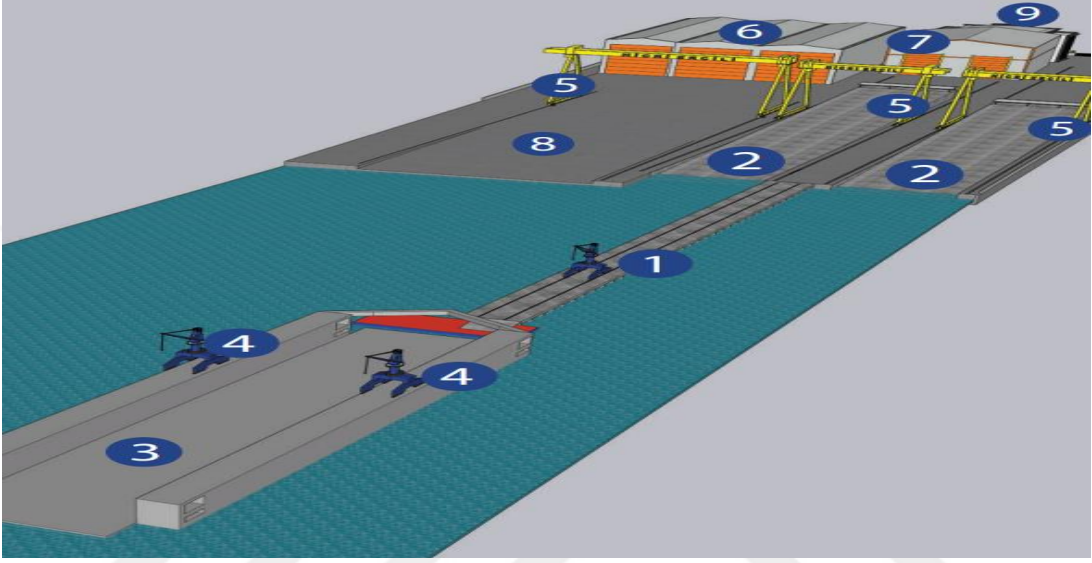
Oto yedek parça ve kimyasal madde taşımacılığı işini kurmuştur. Şirket, başlangıçta kimyasal madde taşımacılığı, yedek parça satışı ve petrol ofisi akaryakıt satış istasyonlarıyla büyümüştür. 2000'li yıllarda, özellikle sülfürik asit ithalatıyla birlikte büyüme hız kazanmıştır. Hicri Ercili, bugün kara ve kimyasal paslanmaz deniz tankerleriyle yurt içi ve yurt dışı taşımacılık faaliyetlerini yürütmektedir (<https://www.hicriercili.com.tr/>, Erişim Tarihi : 02.03.2024).

Şirketin, Türkiye'nin en büyük Alüminyum Sülfat üreticisi olması, üretim tesisleri ve TSE onaylı ADR'li Sülfürik Asit ve Amonyak tankı üretimi gibi inovasyonlarla sektördeki liderliğini pekiştirmiştir. Deniz taşımacılığına da odaklanarak gemi filosunu geliştirmiştir ve Yalova Altınova'da büyük bir tesis kurarak gemi inşa, tadilat ve bakım onarım faaliyetlerini sürdürmüştür (<https://www.hicriercili.com.tr/>, Erişim Tarihi : 02.03.2024).

Hicri Ercili Grup Şirketleri, 11 farklı lokasyonda faaliyet göstererek organik büyümeyi sürdürmektedir. 2015'te başlayan tersane faaliyetleri, 2017'de genişleyerek daha büyük bir alan ve kapasiteye ulaşmıştır. Bugün, müşterilere yeni inşaat, yat, özel tip tekneler, gemi tamir ve dönüşüm projeleri için hizmet veren çok boyutlu bir tersane haline gelmiştir. Tersanede bulunan tesisler aşağıdaki gibidir (<https://www.hicriercili.com.tr/>, Erişim Tarihi : 02.03.2024) :

1. Floating Dock (116m x 25 m)
2. 2 Slipway Kızak (120m x 30m ve 130m x 30m)
3. Yüzen Havuz (116 x 25 m, 4900T LC)
4. 2 adet 10T Dock Vinç

5. 4 Gantry Vinç
6. 3 Hangar
7. Atölyeler
8. Yeni İnşaa Kızak
9. Yönetim Binası ve Ofisler



Şekil 4.1 Hicri Ercili tersanesinde bulunan tesisler (Hicri Ercili tanıtım belgesinden alınmıştır)

Tersanede verilen hizmetler aşağıdaki gibidir (<https://www.hicriercili.com.tr/>, Erişim Tarihi : 02.03.2024) :

- Çelik İşleri
- Boru İşleri
- İşleme İşleri
- Makine ve Elektrik İşleri
- Güvenlik Ekipmanları
- Yük Alanları-Kapak Tamirleri ve Yükseltmeleri
- Dönüşüm Projeleri (BWT Sistemleri, Gaz Arıtma Sistemleri Kurulumları)
- Dönüşüm Projeleri (Boy Verme vb.)
- İtme Sistemi İşleri
- Yüzer Havuz İnşası

- Palamar ve Römorkör Üretimi

Hicri Ercili Tersanesi, müşterilerine çeşitli hizmetler sunarak gemilerinin bakımını ve modernizasyonunu gerçekleştirmektedir. Bu hizmetler arasında kuru havuz/kızaklar, tamir ve modernizasyon/dönüşüm projeleri bulunmaktadır. Tersane, müşterilerinin gemilerinin durumunu korumak için çelik işlerini kalite ve iş sağlığı güvenliği kurallarına uygun bir şekilde gerçekleştirmektedir. Ayrıca, boru hatları için önemli olan işleri için çözümler sunmaktadır ve yüzey hazırlığı ve boyama işlerini yapmaktadır. Tersane, mekanik ve elektrik işlerinde de bir ekibe sahiptir ve gemilerin çeşitli parçalarının bakımını yapmaktadır. Çevre dostu sistemlerin entegrasyonu ve dönüşüm projeleri gibi özel projelerde de hizmet sunmaktadır. Son olarak, gemilerin güvenli ve ekonomik seyahatler yapabilmesi için itme sistemlerinin bakımını ve modernizasyonunu gerçekleştirmektedir. Hicri Ercili Tersanesi yaklaşık bir yılda 3 gemi inşa etmektedir ve 100 geminin bakım-onarımı yapılmaktadır. Ortalama 150-200 kişi beyaz ve mavi yaka olarak çalışmaktadır ve taşeron firmaların çalışanlarıyla birlikte yaklaşık 1000 kişi tersanede çalışmaktadır (Hicri Ercili uzmanlarından alınan bilgiler).



Şekil 4.2 Hicri Ercili Tersanenin Sertifikaları (Hicri Ercili tanıtım belgesinden alınmıştır)

4.2. YENİ İNŞAA PROJELERİ

Hicri Ercili Tersanesi, denizcilik sektöründe geniş bir yelpazede hizmet sunmaktadır. Ticari gemilerden yüzen havuzlara ve yatlardan hizmet teknelerine kadar hizmet vermektedir. Yüzen

havuzlar, deniz araçlarının bakım ve onarımı için kritik öneme sahiptir. Ticari gemilerden çok çeşitli havuzlara kadar geniş bir yelpazede yüzen doklar inşa eden tersane, hizmetler sunmaktadır. Ayrıca, askeri gemiler ve diğer deniz araçları için de hizmet vermektedir ve bugüne kadar ürettiği 20'den fazla tekne ile kendi markasını oluşturmuştur. Yat inşaatı da Hicri Ercili Tersanesi'nin alanlarından biridir. Farklı boyutlarda, sınıflarda ve özelliklerdeki yat inşaatı ve tamir projelerini gerçekleştirmektedir. Farklı tonaj ve tiplerdeki projelerle denizlere imzasını atan tersane, geleceğin ihtiyaç duyulan verimli ve çevre dostu gemilerinin projelerini hazırlayarak gerekli çözümleri sunmaktadır. Tersane ve gemi inşa sektörü, sürekli teknolojik yeniliklere ayak uydurmak uzun ve masraflı bir süreçtir. Ancak, yeni teknolojilerin uygulanması sürecinde dikkatli olunmalıdır çünkü işlerin aksamaması için uygun bir adaptasyon süreci gerekmektedir. Bu bağlamda, Hicri Ercili Tersanesi gelecekteki operasyonlarını optimize etmek ve verimliliği artırmak için Genetik Algoritma gibi analiz araçlarını kullanma potansiyeline sahiptir.

Hicri Ercili, sadece Türkiye'de değil, aynı zamanda Hollanda'da da önemli bir tersane olmasının yanı sıra bir lojistik şirkettir. Pazarda zaten rekabetçi olması nedeniyle, tez amacını yerine getirdiği için 'Bir tersanede lojistik süreçlerin optimize edilmesi' tezi için uygun bir tersanedir. Teknoloji ve yapay zeka alanındaki gelişmelerin artması nedeniyle, tersaneleri dijitalleştirmek için önerilerde bulunmak elzemdir. Hicri Ercili Tersanesinde aynı zamanda ERP sistemi kurulmayı düşünülmektedir. ERP (Kurumsal Kaynak Planlama) sistemi, bir işletmenin farklı departmanları arasındaki iş süreçlerini tek bir sistemde birleştiren yazılım türüdür. Finans, insan kaynakları, tedarik zinciri yönetimi, üretim, satış ve pazarlama gibi ana iş alanlarını entegre ederek işletmenin verimliliğini artırmaktadır ve karar alma süreçlerini kolaylaştırmaktadır. Bunun dışında tersanelerde yapılan ziyaretlere göre ve uzman görüşmelere göre tersandeki lojistik akışları maksimize etmek için toplamda 8 adet olabilecek çözümler düşünülmüştür. Bu çözümlerin fiyatları hem türk hem de yabancı farklı şirketlerden ve firmalardan alınarak kabaca bir aritmetik ortalaması çıkartılmıştır. Ortalama maliyetleri aşağıdaki gibidir :

1. RFID, barkodlama Türkiye piyasasını göz önünde bulundurarak satıcılardan alınan ortalama maliyeti yaklaşık 45 000 USD olmaktadır.

2. Endüstriyel robotlar veya cobotlar için çeşitli şirketlerden alınan fiyatlar değişmektedir. Yüzey temizleme, kum veya su raspası ve boya işlemini yapan robot fiyatları 250 000 USD robotik kol başınadır.
3. 3D Metal Printing çeşitli şirketlerden araştırılan maliyet ortalama 500 000 dolardır. Bu large boyutundadır ve rahatlıkla tüm tip sacları için kullanılmaktadır.
4. AGV'lerin bir tersane ortamına entegre edilmesinin kapsamlı maliyet etkileri, projenin kapsamı, gerekli AGV miktarı, entegrasyon karmaşıklığı ve devam eden işletme gereksinimleri gibi faktörlere bağlıdır. Tersane işletmecilerinin ihtiyaçlarını dikkatlice değerlendirmesi ve AGV tedarikçileri veya entegratörlerle danışarak, özel durumlarına uygun özel maliyet tahminleri alması gerekmektedir. AGV'lerin piyasası yaklaşık AGV başına 80 000 dolardır. Nispeten ortalama bir tersaneye genellikle 2 adet AGV gerekmektedir. AGV başına da fiyatı ortalama 80 000 dolardır.
5. Orta büyüklükte bir tersane için bir Dijital İkiz geliştirilmesinin ve dağıtımının maliyeti, yaklaşık 60 000 Dolardır. Hicri Ercili Tersanesinde, taşeron şirketlerinden gelen personeli de kaydı alındığında yaklaşık 1000 kişi çalışmaktadır. Yani Orta Boyut bir tersane için yaklaşık 60 000 dolardır.
6. Yeni gelişen teknoloji düşünüldüğünde VR kullanımı ve uygulanması çok basit ve az zahmetli bir prosedürdür. Hicri Ercilinin alanını kaydı alındığında, VR kullanımı geniş bir alanı kapsamaktadır, bu nedenle fiyatı diğer şirketlerinden fark edebilmektedir. VR'ın tersanede uygulanması ortalama 10 000 dolar bir maliyete sebep olabilmektedir.
7. Eğitim uygulamalarının oluşturulması ve uygulanması çok zaman almadan yapılan bir işlemdir. Bu işlem için ortalama fiyat yaklaşık 5000 USD olacaktır.
8. Tersanede darboğazları tespit etmenin maliyeti, tersanenin büyüklüğü, operasyonların karmaşıklığı ve analiz için kullanılan yöntemler gibi faktörlere bağlı olarak önemli ölçüde değişebilmektedir.. Bu, veri toplama araçları, yazılımlar, danışmanların işe alınması ve personel zamanı gibi giderleri içerebilmektedir. Kısaca Dar Boğazların tespiti için gerekli mühendisler veya yapılacak analizlerin maliyeti 5000 dolardır.

4.3. GENETİK ALGORİTMA (KNAPSACK METODU) İLE TERSANENİN DİJİTALEŞMESİ VE GELİŞTİRİLMESİ

Mühendislik alanındaki geniş problem yelpazesi, diğer alanlarda olduğu gibi, farklı türde, farklı özelliklere ve ayarlamalara sahip algoritmaların kullanımını gerektirmektedir. İlk olarak rastgele bir popülasyon oluşturulmaktadır (problemin uygun çözümleri için).

Hicri Ercili tersanesi için birkaç çözüm sunulmaktadır ve bunlar hangi olduklarını açıklanmıştır. Genetik Algoritma ile bu problemin çözümün optimizasyonu aşağıdaki gibidir. Bilindiği gibi 0 veya 1 gibi bir karakter bir gen oluşmaktadır. Bir sürü genlerin birliği/birleşimi kromozomu oluşturmaktadır. Tüm bu kromozomların birliği ise bir popülasyonu oluşturmaktadır. Evolüsyonda ve genetikte gibi Fitness Çözümü uygulanacaktır yani bir popülasyonda en çok hayatta kalma şansları olanları seçilmektedir ya da kısaca probleme en iyi olabilecek ve kabul görececek çözümleri seçilecektir.

Python dilinde genetik algoritma uygulanacaktır. Birinci aşamada Python yazılımında, problemi çözebilmek için çanta (knapsack) kodu yazılacaktır. Çanta kodu genel kod olarak yazılacaktır. İkinci aşamada ise yine çanta problemin kodu kullanılarak kombinasyon ve en iyi üç çözümleri alabilmek için kod modifike edilecektir. Genetik algoritma ve ana fonksiyon kısımların kodu modifike edilecektir.

Bu tezdeki tüm kodlar, genel bilgi kullanılarak ve Python kütüphaneleri ile yapay zeka araçlarının yardımıyla yazılmıştır ve tez problemine göre geliştirilmiştir.

4.3.1. Birinci Aşama

Birinci Aşamada verilmiş olup çözümlerin maliyetini ve tersaneden değeri yazılmaktadır. Tersane ile yapılan uzman görüşlerine göre 1'den 10'a kadar bir bir tercih değerlendirilmesi yapılmıştır. Ayrıca tersaneden böyle bir dijitalleşme ve geliştirme için 200 000 USD bir bütçe tercih edilmiştir. Aşağıdaki tabloda bu çözümlerin değeri ve fiyatı gösterilmektedir.

1. Çözüm (İtem)	RFID,Barkodlama ve Blok Zinciri
2. Çözüm (İtem)	Robot/Cobot
3. Çözüm (İtem)	3D Printing

4. Çözüm (İtem)	AGV
5. Çözüm (İtem)	VR
6. Çözüm (İtem)	Eğitim Uygulamaları
7. Çözüm (İtem)	Dar Boğaz Tespiti
8. Çözüm (İtem)	Digital Twin

Tablo 4.1 Olabilecek çözümlerin yazılması

Çözümler	1	2	3	4	5	6	7	8
Maaliyet (Weight)	45000	250000	500000	160000	10000	5000	5000	60000
Değer (Value)	8	5	5	5	10	10	5	4

Tablo 4.2 Olabilecek çözümlerin değerleri ve maaliyetleri

Yukarıdaki tabloların Python Programlamasında çevirildiğinde aşağıdaki kodlama gibidir.

```
import random

# Define the items available for selection
items = [
    {"name": "item1", "weight": 45000, "value": 8},
    {"name": "item2", "weight": 250000, "value": 5},
    {"name": "item3", "weight": 500000, "value": 5},
    {"name": "item4", "weight": 160000, "value": 5},
    {"name": "item5", "weight": 10000, "value": 10},
    {"name": "item6", "weight": 5000, "value": 10},
    {"name": "item7", "weight": 5000, "value": 5},
    {"name": "item8", "weight": 60000, "value": 4},
]
```

Şekil 4.3: Çözümlerin Python'da Yazılması

Python programında orijinal bir kod yazılmıştır. Bu problemin çözülmesi için ilk olarak çözümlerin değerleri ve maliyetleri ,value ve weight şeklinde yazıldıktan sonra tersanenin bütçesi yazılmıştır. Bir sonraki adım ise Genetik Algoritmanın parametreleri yazılmasıdır. Popülasyonun sayısı , mutasyon oranı ve jenerasyonlar parametrelerin parçasıdır. Populasyonun içindeki kromozomlar 8 adet genden oluşturulmaktadır. Jenerasyon = 100" ifadesi, genetik algoritmanın çalışma sürecinde 100'üncü jenerasyonun oluşturulduğunu ifade etmektedir. Yani, bu noktada, başlangıç popülasyonundan başlayarak 100 iterasyon gerçekleştirilmiş ve yeni bireyler üretilmiştir. Her iterasyonda, mevcut popülasyondan seçilen bireyler çaprazlama ve mutasyon operatörleri kullanılarak yeni bireyler üretilmiş ve yeni jenerasyon oluşturulmuştur. Dolayısıyla, "jenerasyon = 100" ifadesi, genetik algoritmanın 100 iterasyon tamamlandığında olduğunu belirtmektedir.. Genetik algoritmalarda mutasyon oranı, her genin veya bitin üreme sürecinde mutasyona uğrama olasılığını belirtmektedir. Bu oran, çeşitliliği artırmak ve erken yakınsamayı önlemek için önemlidir. Genellikle küçük bir değer olarak belirlenmektedir ve optimal değeri problem alanına ve arama alanının özelliklerine bağlıdır. Mutasyon oranı problemde 0.1 olarak belirlenmiştir.

```
# Define the knapsack capacity
knapsack_capacity = 200000

# Genetic Algorithm Parameters
population_size = 8
mutation_rate = 0.1
generations = 100
```

Şekil 4.4 Parametrelerin Kodlanması

Bir sonraki adım kodlama sırasında başlangıç popülasyonun oluşturulması ve uygunluk fonksiyonun yazılmasıdır. Başlangıç popülasyonunu 0 ve 1 binar sistemi ile yazılmaktadır. '0' olan kodu bu genin olmadığını ve '1' olan kodu bu genin var olduğunu anlamına gelmektedir. Örneğin: 11110000 kromozomu RFIDi, barkodlamayı, blok zincirini, robot kullanımını, 3D yazıcıyı ve AGV'leri içermektedir. Geri kalan çözümler VR, eğitim uygulamaları, dijital ikizler ve dar boğaz tespitleri içermemektedir. Bu Python kodu, genetik algoritma çerçevesinde bir sırt çantası problemi çözmek için özelleştirilmiş bir uygunluk fonksiyonu tanımlamaktadır.

Fonksiyon, her bir ikili deęerin, karřılık gelen öęenin seęilip seęilmedięine göre temsil edildięi bir birey çözümleri temsili olarak almaktadır. Fonksiyon, her öęe için, temsil edilen bireyde seęilmişse deęerlerini ve aęırlıklarını biriktirmektedir. Seęilen öęelerin toplam aęırlığı sırt çantasının kapasitesini aşarsa, fonksiyon geçersiz bir çözümleri olduğunu belirten bir uygunluk deęeri olan 0 döndürmektedir. Aksi halde, seęilen öęelerin toplam deęerini bireyin uygunluk deęeri olarak döndürmektedir. Temelde, bu uygunluk fonksiyonu, sırt çantası probleminin potansiyel çözümlerinin uygunluklarını deęerlendirmektedir ve genetik algoritmayı, sırt çantasının kapasite kısıtlarına uyarak deęeri maksimize eden optimal çözümleri seęmeye yönlendirmektedir. Şekil 4.5'te gibi kod dilinde yazılmıştır.

```
# Generate initial population randomly
def generate_individual():
    return [random.randint(0, 1) for _ in range(len(items))]

def generate_population():
    return [generate_individual() for _ in range(population_size)]

# Fitness function
def fitness(individual):
    total_value = 0
    total_weight = 0
    for i in range(len(items)):
        if individual[i] == 1:
            total_value += items[i]["value"]
            total_weight += items[i]["weight"]
    if total_weight > knapsack_capacity:
        return 0
    else:
        return total_value
```

Şekil 4.5 Başlangıç Popülasyonu ve Uygunluk Fonksiyonunun Python Dilinde Kodu

Bu Python kodu, genetik algoritma içinde sıkça kullanılan üç fonksiyonu tanımlamaktadır:

Seçim Fonksiyonu (seçim): Bu fonksiyon, bir popülasyonu girdi olarak almaktadır ve fitness deęerlerine göre sıralamaktadır, ardından en yüksek uygunluk deęerine sahip olanların bulunduğu sıralı popülasyonun üst yarısını (en yüksek uygunluk deęerine sahip olanları) bir sonraki neslin ebeveynleri olarak seęmektedir.

Çaprazlama Fonksiyonu (çaprazlama): Bu fonksiyon, iki ebeveyn bireyi almaktadır ve çaprazlama işlemi yaparak iki çocuk birey üretmektedir. Rastgele olarak 1 ile öęelerin uzunluğu

eksi 1 arasında bir çaprazlama noktası seçmektedir. Ardından, çaprazlama noktasından önce ve sonra ebeveynlerin genetik materyalini birleştirerek iki yeni çocuk birey oluşturmaktadır.

Mutasyon Fonksiyonu (mutasyon): Bu fonksiyon, bir bireyi girdi olarak almaktadır ve mutasyon oranına göre genetik materyalini rastgele mutasyona uğratmaktadır. Bireyin temsilindeki her bit için, rastgele oluşturulan bir olasılık değerinin mutasyon oranından daha küçük olup olmadığını kontrol etmektedir. Eğer küçükse, biti tersine çevirir (yani, 0'ı 1'e veya 1'i 0'a değiştirmektedir). Son olarak, mutasyona uğramış bireyi döndürmektedir.

Bu üç fonksiyon bir araya gelerek, genetik algoritmanın çekirdek bileşenlerini oluşturmaktadır. Bu algoritma, çoklu nesiller boyunca aday çözümlerinin bir popülasyonunu evrimleştirerek optimizasyon problemlerini çözmek için kullanılır, örneğin sırt çantası problemi gibi. Bu üç fonksiyonların kodu Şekil 4.6'da gibi yazılmıştır.

```
# Selection function
def selection(population):
    sorted_population = sorted(population, key=lambda x: fitness(x), reverse=True)
    return sorted_population[:population_size//2]

# Crossover function
def crossover(parent1, parent2):
    crossover_point = random.randint(1, len(items) - 1)
    child1 = parent1[:crossover_point] + parent2[crossover_point:]
    child2 = parent2[:crossover_point] + parent1[crossover_point:]
    return child1, child2

# Mutation function
def mutation(individual):
    for i in range(len(individual)):
        if random.random() < mutation_rate:
            individual[i] = 1 - individual[i]
    return individual
```

Şekil 4.6 Fonksiyon Seçimi, Crossover Fonksiyonu ve Mutasyon Fonksiyonu

Bu kod bloğu, genetik algoritmanın ana işlevlerini içeren seçim, çaprazlama ve mutasyon işlevlerini açıklamaktadır ve ardından bu işlevlerin ana döngü içinde nasıl kullanıldığını aşağıdaki gibi açıklamaktadır :

Seçim Fonksiyonu (selection):

1. selection fonksiyonu, verilen bir popülasyonu uygunluk değerine göre sıralamaktadır ve en iyi bireylerin yarısını seçmektedir.

2. sorted_population deęişkeni, popülasyonu fitness fonksiyonuna göre azalan sırada sıralamaktadır.
3. sorted_population[:population_size//2] ifadesi, sıralı popülasyonun yarısını almaktadır. Bu, seçilen bireylerin sayısını popülasyon boyutunun yarısı olarak belirlemektedir.

Çaprazlama Fonksiyonu (crossover):

1. crossover fonksiyonu, iki ebeveyn arasında çaprazlama yaparak iki çocuk üretmektedir.
2. crossover_point deęişkeni, çaprazlama noktasını belirlemek için rastgele bir sayı seçmektedir.
3. child1 ve child2 deęişkenleri, çaprazlama noktasına göre ebeveynlerin parçalarını birleştirerek iki çocuk oluşturmaktadır.

Mutasyon Fonksiyonu (mutation):

1. mutation fonksiyonu, bir bireye mutasyon uygulamaktadır.
2. Her bir gen için, belirli bir olasılıkla mutasyon uygulanmaktadır. Bu, genlerin deęerini (0 veya 1) tersine çevirmektedir.

Ana Döngü İçinde Çözümlerin Sıralanması ve Yazdırılması:

1. all_solutions.sort(key=lambda x: fitness(x), reverse=True): Bu satır, bulunan tüm çözümleri fitness fonksiyonuna göre azalan sırada sıralamaktadır. key=lambda x: fitness(x) kısmı, fitness fonksiyonunu kullanarak her çözümün uygunluk deęerini hesaplamaktadır ve reverse=True ile sıralama tersine çevrilerek en yüksekten en düşüğe doğru yapılmaktadır.
2. for solution in all_solutions:: Bu döngü, sıralanmış çözümler listesi üzerinde dönmektedir. Her bir çözüm için aşağıdaki işlemler yapılmaktadır :
 - a. Çözümde seçilen elemanların toplam deęeri ve ağırlığı hesaplanmaktadır.
 - b. Çözüm ve hesaplanan toplam deęer ve ağırlık yazdırılmaktadır.

- c. Her çözüm arasında bir boş satır bırakılarak çıktı daha okunabilir hale getirilmektedir.

Bu Python kodunda, optimizasyon problemlerini çözmek için genetik algoritma kullanarak bir çözüm üretmektedir. Kodun her bir parçası şu işlevleri yerine getirmektedir:

- Genetik Algoritma Fonksiyonu (`genetic_algorithm`): Bu fonksiyon genetik algoritmayı yönetmektedir. İlk olarak, `generate_population` fonksiyonunu kullanarak aday çözümlerin başlangıç popülasyonunu oluşturmaktadır. Sonra, belirli bir nesil sayısı için, popülasyon üzerinde seçim, çaprazlama ve mutasyon işlemlerini tekrarlayarak çalışmaktadır. Her nesil için, mevcut popülasyondan en uygun bireyleri seçer ve onları bir sonraki neslin ebeveynleri olarak belirlemektedir. En sonunda, tüm nesiller boyunca bulunan en iyi bireyi seçer ve bu bireyi döndürmektedir.
- Ana Döngü: `genetic_algorithm` fonksiyonunun içinde, belirtilen nesil sayısı (`generations`) kadar bir döngü bulunmaktadır. Her bir iterasyonda:
 - Mevcut popülasyonun bir kısmını, uygunluk değerlerine göre seçim yaparak belirlemektedir (`selection` fonksiyonu kullanılmaktadır).
 - Çocuk bireyler üretilene kadar, çaprazlama ve mutasyon işlemleri uygulanarak yeni bireyler oluşturulmaktadır.
 - Seçilen bireylerle yeni oluşturulan çocuk bireyleri birleştirilerek yeni bir popülasyon oluşturulmaktadır.
 - Bu işlem belirtilen nesil sayısına ulaşıncaya kadar devam etmektedir.
- En İyi Bireyin Belirlenmesi: Tüm nesiller işlendikten sonra, fonksiyon, fitness değerine göre her bir bireyin uygunluk değerini hesaplayan ve en yüksek uygunluk değerine sahip bireyi bulan `max` fonksiyonunu kullanarak en iyi bireyi belirlemektedir.
- Dönüş: Fonksiyon, genetik algoritma tarafından belirlenen en iyi bireyi döndürmektedir. Bu, verilen probleme göre en iyi çözüm olarak kabul edilmektedir.

```

# Genetic Algorithm
def genetic_algorithm():
    population = generate_population()
    for _ in range(generations):
        selected_population = selection(population)
        children = []
        while len(children) < population_size - len(selected_population):
            parent1 = random.choice(selected_population)
            parent2 = random.choice(selected_population)
            child1, child2 = crossover(parent1, parent2)
            child1 = mutation(child1)
            child2 = mutation(child2)
            children.append(child1)
            children.append(child2)
        population = selected_population + children
    best_individual = max(population, key=lambda x: fitness(x))
    return best_individual

```

Şekil 4.7 Genetik Algoritmanın Uygulanması

Şekil 4.7’ de bulunan genetik algoritma kodunun her satırı aşağıdaki gibi açıklanmıştır :

1. `def genetic_algorithm():` Fonksiyon Tanımı: `genetic_algorithm` adlı fonksiyonu tanımlar. Bu fonksiyon genetik algoritmanın adımlarını uygulayacaktır.
2. Popülasyonun Başlatılması: `generate_population` fonksiyonunu çağırarak başlangıç popülasyonunu oluşturur. Bu popülasyondaki her birey, problemin olası bir çözümünü temsil etmektedir.
3. `for _ in range(generations):` Nesiller Döngüsü: Önceden belirlenmiş sayıda nesil için döngüyü başlatmaktadır. Her nesil, popülasyonun evrim geçirdiği bir iterasyonu temsil etmektedir.
4. `selected_population = selection(population)` Seçim: `selection` fonksiyonunu çağırarak, mevcut popülasyondan bireyleri seçmektedir. Seçim süreci genellikle daha yüksek uygunluğa sahip bireyleri tercih eder, böylece daha iyi çözümler genlerini bir sonraki nesile aktarma şansına sahip olmaktadır.
5. `children = []` Çocukların Başlatılması: Bu, mevcut nesilde üretilecek olan çocukları depolamak için boş bir liste başlatmaktadır.

6. `while len(children) < population_size - len(selected_population):` Çocuk Üretim Döngüsü: Çocuk sayısı, seçilen popülasyonun eksik kalan kısmını dolduracak kadar olana kadar dönmektedir. Bu, popülasyon büyüklüğünün nesiller boyunca sabit kalmasını sağlamaktadır.
7. `parent1 = random.choice(selected_population)/ parent2 = random.choice(selected_population)` Ebeveyn Seçimi (Ebeveyn 1 ya da 2): Seçilen popülasyondan rastgele bir ebeveyn seçmektedir.
8. `child1, child2 = crossover(parent1, parent2)` Çaprazlama: İki ebeveyni argüman olarak `crossover` fonksiyonunu çağırılmaktadır. Çaprazlama fonksiyonu, ebeveynlerin genlerini birleştirerek iki yeni yavru (çocuk) oluşturmaktadır.
9. `child1 = mutation(child1)` Mutasyon (Çocuk 1): İlk çocuğa rastgele mutasyonlar uygulamak için `mutation` fonksiyonunu çağırılmaktadır. Mutasyon, çocuğun genomunda küçük değişiklikler yaparak genetik çeşitliliği sağlamaktadır.
10. `child2 = mutation(child2)` Mutasyon (Çocuk 2): İkinci çocuğa rastgele mutasyonlar uygulamak için `mutation` fonksiyonunu çağırılmaktadır.
11. `children.append(child1)` ve `children.append(child2)` : Mutasyona uğramış çocuğu çocuklar listesine eklemektedir.
12. `population = selected_population + children` Popülasyonu Güncelleme: Seçilen popülasyon ile yeni üretilen çocukları birleştirerek bir sonraki nesil için yeni popülasyonu oluşturmaktadır. Bu, mevcut nesilden en iyi bireylerin ve yeni yavruların birlikte ilerlemesini sağlamaktadır.
13. `best_individual = max(population, key=lambda x: fitness(x))` En İyi Bireyi Bulma: Son popülasyondaki en yüksek uygunluğa sahip bireyi bulmaktadır. `Max` fonksiyonu, bireylerin uygunluğunu anahtar olarak kullanarak çalışmaktadır.
14. `return best_individual` En İyi Bireyi Döndürme: Tüm nesiller tamamlandıktan sonra bulunan en iyi bireyi döndürmektedir. Bu birey, genetik algoritmanın bulunduğu en iyi çözümdür.

Ana Fonksiyon

Bu Python kodu, ana bir işlev içermektedir. Bu işlev, genetik algoritmayı çağırır, en iyi çözümü almaktadır ve bu çözümün değerini ve ağırlığını hesaplamaktadır. İşlevin her bir parçası belirli görevleri gerçekleştirmektedir. İlk olarak, ana işlev bloğu (`__main__`) kodun başka bir dosyaya eklenmesi durumunda çalıştırılmasını sağlamaktadır. Kodun temel çalışma mantığı burada bulunmaktadır. `genetic_algorithm` fonksiyonu çağrılarak en iyi çözüm elde edilmektedir. Daha sonra, bu çözüm üzerinde bir döngü kullanılarak her bir öge için seçilip seçilmediği kontrol edilmektedir. Eğer öge seçilmişse, toplam değer ve toplam ağırlığa ilgili değerler eklenmektedir. Son olarak, en iyi çözüm, toplam değer ve toplam ağırlık ekrana yazdırılmaktadır. Bu kod parçası, genetik algoritmayı çağırarak ve en iyi çözümün değerini ve ağırlığını hesaplayarak belirli bir probleme çözüm üreten bir Python betiğini temsil etmektedir.

```
# Main function
if __name__ == "__main__":
    best_solution = genetic_algorithm()
    print("Best solution:", best_solution)
    total_value = 0
    total_weight = 0
    for i in range(len(items)):
        if best_solution[i] == 1:
            total_value += items[i]["value"]
            total_weight += items[i]["weight"]
    print("Total value:", total_value)
    print("Total weight:", total_weight)
```

Şekil 4.8 Ana İşlev Fonksiyonun Kodu

Şekil 4.8'deki Ana İşlev Fonksiyonunun her satırı aşağıdaki gibi açıklanmıştır:

1. `if __name__ == "__main__"` : Bu satır, betiğin ana program olarak çalıştırılıp çalıştırılmadığını kontrol etmektedir. Python'da bu deyim, modüller içe aktarıldığında veya doğrudan çalıştırıldığında kodun belirli bölümlerinin çalışmasını sağlamak veya engellemek için kullanılmaktadır. Betik doğrudan çalıştırılıyorsa, if bloğu içindeki kod çalıştırılmaktadır.
2. `best_solution = genetic_algorithm()` : Bu satır, `genetic_algorithm()` fonksiyonunu çağırır ve döndürülen değeri `best_solution` değişkenine atmaktadır. `genetic_algorithm()`

fonksiyonunun, verilen bir problem için en iyi çözümü döndürmesi beklenmektedir (genellikle bir liste veya dizi biçiminde).

3. `print("Best solution:", best_solution)` : Bu satır, genetik algoritma tarafından bulunan en iyi çözümü yazdırmaktadır. Çıktı, `genetic_algorithm()` fonksiyonunun döndürdüğü çözüm biçiminde olacaktır.
4. `total_value = 0 total_weight = 0` : Bu satırlar, `total_value` (toplam değer) ve `total_weight` (toplam ağırlık) değişkenlerini sıfır olarak başlatmaktadır. Bu değişkenler, en iyi çözüme dahil edilen öğelerin toplam değerini ve toplam ağırlığını biriktirmek için kullanılacaktır.
5. `for i in range(len(items))`: Bu satır, `items` listesinin indeksleri üzerinde dönen bir döngü başlatmaktadır. `Items` listesinin, öğelerin değerlerini ve ağırlıklarını temsil eden "value" (değer) ve "weight" (ağırlık) anahtarlarına sahip sözlükler içerdiği varsayılmaktadır.
6. `if best_solution[i] == 1` : Bu satır, `i` indeksindeki öğenin en iyi çözüme dahil edilip edilmediğini kontrol etmektedir. Genetik algoritmaların, genellikle sırt çantası (knapsack) problemi gibi problemler için çözümleri ikili bir liste olarak temsil ettiği durumda, 1 öğenin dahil edildiğini, 0 ise edilmediğini göstermektedir.
7. `total_value += items[i]["value"]` ve `total_weight += items[i]["weight"]` Eğer öğe en iyi çözüme dahil edilmişse (`best_solution[i] == 1`), bu satırlar öğenin değerini `total_value` değişkenine ve ağırlığını `total_weight` değişkenine eklemektedir.
8. `print("Total value:", total_value)` ve `print("Total weight:", total_weight)` Döngü tamamlandıktan sonra, bu satırlar en iyi çözüme dahil edilen öğelerin toplam değerini ve toplam ağırlığını yazdırmaktadır. Bu, çözümün değer ve ağırlık kriterleri açısından etkinliğini özetlemektedir.

4.3.2. İkinci Aşama

İkinci Aşamada kod yazıldıktan sonra çıkacak sonuçların yazılması ve değerlendirilmesidir. İkinci aşama da aynı zamanda üç aşamaya ayrılmaktadır.

1. Tüm olabilecek kombinasyonların kodu eklenmesi ve kombinasyonların yazılması

2. Tüm olabilecek kombinasyonlardan değer/maaliyet şeklinde yazılması
3. En uygun 3 seçeneklerin arasında en uygunun seçilmesi

Python dilinde kodlama sonrasında olabilecek tüm kombinasyonlar ve toplam değerleri ve maaliyetleri ile birlikte aşağıdaki gibi yazılmıştır. Çözümler, öğelerin toplam değerini maksimize etme ve toplam ağırlığı minimize etme arasında bir denge göstermektedir. Algoritma nesiller ilerledikçe, genellikle daha yüksek toplam değerlere sahip çözümlere yakınsar, ancak genellikle daha yüksek toplam ağırlıklarla birlikte gelmektedir. Kombinasyonlardaki farklılıklar, algoritmanın çözüm alanını keşfetmesini, verilen kısıtlamaları karşılayan en optimal öge kombinasyonunu aramasını yansıtmaktadır. Genel olarak, çıktı karmaşık optimizasyon problemleri için yaklaşık olarak optimal çözümler bulmak için büyük bir çözüm alanında etkili bir şekilde arama yapmak için genetik algoritmanın etkinliğini göstermektedir.

Bu kod, bir küme öğenin değerini maksimize etme ve ağırlık kısıtlamasına uygun olma problemine genetik algoritmayı kullanarak çözüm getirir. `genetic_algorithm()` fonksiyonunda algoritma birkaç aşamada gerçekleşmektedir : İlk olarak, `generate_population()` fonksiyonu aracılığıyla başlangıç çözümlerinin bir ilk popülasyonu oluşturularak başlatılmaktadır. Ardından, bir dizi yinelemeye, veya nesile, girilir ve bu süreçte seçim, çaprazlama ve mutasyon işlemleri gerçekleştirilerek yeni bir çözüm popülasyonu oluşturulmaktadır. Seçim sırasında, mevcut popülasyondan bireyler, genellikle uygunluklarına göre gelecek nesil için ebeveyn olarak seçilmektedir. Çaprazlama, seçilmiş ebeveyn çiftlerini birleştirerek bir çaprazlama işlevi kullanarak soy oluştururken, mutasyon, soyun genetik yapısına rastgele değişiklikler getirmektedir. Bu işlemlerin ardından, popülasyon, seçilmiş ebeveynler ve mutasyona uğramış soy birleştirilerek güncellenmektedir. Bu yinelemeler boyunca, karşılaşılan tüm çözümler, benzersizliği sağlamak için sonradan çiftleştirilen çözümler hariç tutularak `all_solutions` listesinde biriktirilir. `if __name__ == "__main__":` bloğunda, kod yalnızca betik ana program olduğunda, başka bir betik olarak içeri aktarılmadığında çalışmaktadır. Burada, genetik algoritma başlatılır, çözümler uygunluklarına göre sıralanır ve her bir çözüm, toplam değeri ve ağırlığı ile birlikte yazdırılmaktadır. Bu hesaplamalar, her çözümde seçilen öğelere dayanmaktadır.

```

# Genetic Algorithm
def genetic_algorithm():
    population = generate_population()
    all_solutions = [] # Store all solutions found
    for _ in range(generations):
        selected_population = selection(population)
        children = []
        while len(children) < population_size - len(selected_population):
            parent1 = random.choice(selected_population)
            parent2 = random.choice(selected_population)
            child1, child2 = crossover(parent1, parent2)
            child1 = mutation(child1)
            child2 = mutation(child2)
            children.append(child1)
            children.append(child2)
        population = selected_population + children
        all_solutions.extend(population) # Add current population to all solutions
    # Remove duplicates from all solutions
    all_solutions = [list(sol) for sol in set(tuple(sol) for sol in all_solutions)]
    return all_solutions

# Main function
if __name__ == "__main__":
    all_solutions = genetic_algorithm()
    all_solutions.sort(key=lambda x: fitness(x), reverse=True) # Sort solutions by
    fitness
    for solution in all_solutions:
        total_value = sum(items[i]["value"] for i in range(len(items)) if solution[i]
        == 1)
        total_weight = sum(items[i]["weight"] for i in range(len(items)) if
        solution[i] == 1)
        print("Solution:", solution)
        print("Total value:", total_value)
        print("Total weight:", total_weight)
        print()

```

Şekil 4.9 Kombinasyon Kodu Eklemesi

Şekil 4.9' daki Kombinasyon kodunun oluşturması için eklenen genetik algoritmanın her satırı aşağıdaki gibi açıklanmıştır :

1. `def genetic_algorithm()` : `genetic_algorithm` adlı bir fonksiyon tanımlamaktadır, bu fonksiyon genetik algoritmanın mantığını kapsamaktadır.
2. `population = generate_population()` : Başlangıç popülasyonunu `generate_population()` fonksiyonunu çağırarak başlatmaktadır. Bu fonksiyon potansiyel çözümler kümesi oluşturmaktadır.
3. `all_solutions = []` Algoritmanın yürütülmesi sırasında bulunan tüm benzersiz çözümleri saklamak için boş bir liste `all_solutions` oluşturmaktadır.

4. `for _ in range(generations)` : Belirli sayıda nesil boyunca döngü yapmaktadır. Her döngü, genetik algoritmanın bir neslini temsil etmektedir.
5. `selected_population = selection(population)`: Bir seçim yöntemi kullanarak mevcut popülasyondan bir alt küme seçmektedir, bu genellikle bireylerin uygunluğuna (fitness) göre seçim yapmaktadır.
6. `children = []` : Bu nesilde üretilen çocukları saklamak için boş bir liste `children` başlatmaktadır.
7. `while len(children) < population_size - len(selected_population)`: Çocukların sayısı, popülasyon boyutunu korumak için yeterli olana kadar çocuk üretmeye devam etmektedir. Bu, popülasyon boyutunun ebeveynlerle çocukların toplamı ile sabit kalmasını sağlamaktadır.
8. `parent1 = random.choice(selected_population)` Seçilen popülasyondan rastgele bir ebeveyn seçmektedir.
9. `parent2 = random.choice(selected_population)` Seçilen popülasyondan rastgele başka bir ebeveyn seçmektedir.
10. `child1, child2 = crossover(parent1, parent2)` İki seçilen ebeveyn arasında çaprazlama yaparak iki çocuk üretmektedir.
11. `child1 = mutation(child1)` İlk çocuğa mutasyon fonksiyonunu uygular, bu genetik çeşitlilik sağlamaktadır.
12. `child2 = mutation(child2)` İkinci çocuğa mutasyon fonksiyonunu uygular, bu genetik çeşitlilik sağlamaktadır.
13. `children.append(child1)` : İlk mutasyona uğramış çocuğu `children` listesine eklemektedir.
14. `children.append(child2)` İkinci mutasyona uğramış çocuğu `children` listesine eklemektedir.
15. `population = selected_population + children` : Popülasyonu, seçilen ebeveynler ve yeni üretilen çocuklarla güncellemektedir.

16. `all_solutions.extend(population)` : Mevcut popülasyonu, bulunan tüm çözümler listesine eklemektedir.
17. `all_solutions = [list(sol) for sol in set(tuple(sol) for sol in all_solutions)]` : `all_solutions` listesinden tekrar eden çözümleri çıkarmak için her çözümü bir tuple'a dönüştürerek (set işlemlerine izin vermek için), bir set oluşturur ve sonra her çözümü tekrar listeye dönüştürmektedir.
18. `return all_solutions` : Algoritma tarafından bulunan tüm benzersiz çözümleri döndürmektedir.

Şekil 4.9' daki Kombinasyon kodunun oluşturması için eklenen ana fonsiyonun her satırı aşağıdaki gibi açıklanmıştır :

1. `if __name__ == "__main__":` Bu satır, Python dosyasının doğrudan çalıştırılıp çalıştırılmadığını kontrol etmektedir. Dosya doğrudan çalıştırılıyorsa, aşağıdaki kod bloğu yürütülmektedir.
2. `all_solutions = genetic_algorithm()` : Genetik algoritmayı çalıştırarak, bulunan tüm çözümleri içeren bir liste dönmektedir.
3. `all_solutions.sort(key=lambda x: fitness(x), reverse=True)` : Bulunan tüm çözümleri fitness fonksiyonuna göre azalan sırada sıralamaktadır. `key=lambda x: fitness(x)` kısmı, fitness fonksiyonunu kullanarak her çözümün uygunluk değerini hesaplamaktadır ve `reverse=True` ile sıralama tersine çevrilerek en yüksekten en düşüğe doğru yapılmaktadır.
4. `for solution in all_solutions` : Sıralanmış çözümler listesi üzerinde döngü başlatmaktadır. Her bir çözüm için aşağıdaki işlemler yapılmaktadır.
5. `total_value = sum(items[i]["value"] for i in range(len(items)) if solution[i] == 1)` : `solution` listesindeki her 1 değeri için, `items` listesindeki ilgili elemanın (`items[i]`) değerini (`value`) toplamaktadır. Bu, çözümde seçilen elemanların toplam değerini hesaplamaktadır.
6. `total_weight = sum(items[i]["weight"] for i in range(len(items)) if solution[i] == 1)` : `solution` listesindeki her 1 değeri için, `items` listesindeki ilgili elemanın (`items[i]`)

ağırlığını (weight) toplamaktadır. Bu, çözümde seçilen elemanların toplam ağırlığını hesaplamaktadır.

7. print("Solution:", solution) : Mevcut çözümü yazdırmaktadır.
8. print("Total value:", total_value) : Mevcut çözümün toplam değerini yazdırmaktadır.
9. print("Total weight:", total_weight) : Mevcut çözümün toplam ağırlığını yazdırmaktadır.
10. print() : Çözümler arasında bir boş satır bırakır, bu sayede çıktı daha okunabilir olmaktadır.

Çözüm	Toplam Değer	Toplam Maliyet
[1, 1, 0, 1, 1, 1, 1, 1]	47	535000
[1, 1, 0, 1, 1, 1, 1, 0]	43	475000
[1, 1, 0, 0, 1, 1, 1, 1]	42	375000
[1, 1, 0, 1, 1, 1, 0, 1]	42	530000
[1, 0, 0, 1, 1, 1, 1, 1]	42	285000
[0, 1, 0, 1, 1, 1, 1, 1]	39	490000
[1, 0, 0, 1, 1, 1, 1, 0]	38	225000
[1, 1, 0, 0, 1, 1, 1, 0]	38	315000
[1, 1, 0, 1, 1, 1, 0, 1]	38	470000
[1, 1, 0, 1, 0, 1, 1, 1]	37	525000
[1, 0, 0, 1, 1, 1, 0, 1]	37	280000
[1, 1, 0, 1, 1, 0, 1, 1]	37	530000
[1, 1, 0, 0, 1, 1, 0, 1]	37	370000
[0, 1, 0, 1, 1, 1, 1, 0]	35	430000
[0, 1, 0, 1, 1, 1, 0, 1]	34	485000
[0, 1, 0, 0, 1, 1, 1, 1]	34	330000
[1, 1, 0, 1, 0, 1, 1, 0]	33	465000
[1, 1, 0, 1, 1, 0, 1, 0]	33	470000
[1, 1, 0, 0, 1, 1, 0, 0]	33	310000
[1, 0, 0, 1, 0, 1, 1, 1]	32	275000
[1, 0, 0, 0, 1, 1, 0, 1]	32	120000
[1, 1, 0, 1, 0, 1, 0, 1]	32	520000
[1, 0, 0, 1, 1, 0, 1, 1]	32	280000
[1, 1, 0, 0, 0, 1, 1, 1]	32	365000
[1, 1, 0, 1, 1, 0, 0, 1]	32	525000
[0, 1, 0, 1, 1, 0, 1, 1]	29	485000
[0, 1, 0, 1, 0, 1, 1, 1]	29	480000

Çözüm	Toplam Değer	Toplam Maliyet
[1, 1, 0, 0, 0, 1, 1, 0]	28	305000
[1, 0, 0, 1, 1, 0, 1, 0]	28	220000
[1, 1, 0, 1, 0, 0, 1, 1]	27	520000
[1, 1, 0, 0, 0, 1, 0, 1]	27	360000
[1, 0, 0, 0, 1, 0, 1, 1]	27	120000
[0, 1, 0, 0, 1, 0, 1, 1]	24	325000
[1, 0, 0, 0, 0, 1, 0, 1]	22	110000
[1, 1, 0, 0, 0, 0, 1, 1]	22	360000
[0, 1, 0, 0, 0, 1, 0, 1]	19	315000
[1, 0, 0, 0, 0, 0, 0, 1]	12	105000
[0, 0, 0, 0, 1, 0, 0, 0]	10	10000
[0, 0, 0, 0, 0, 0, 1, 1]	9	65000
[0, 1, 0, 0, 0, 0, 0, 1]	9	310000
[1, 1, 1, 1, 0, 1, 0, 1]	37	1020000
[1, 1, 1, 0, 0, 0, 1, 0]	23	800000
[1, 1, 1, 1, 1, 0, 1, 1]	42	1030000
[1, 1, 1, 1, 1, 1, 1, 1]	52	1035000
[1, 0, 1, 1, 1, 1, 0, 1]	42	780000
[0, 1, 1, 1, 1, 1, 1, 1]	44	990000
[1, 1, 1, 1, 1, 1, 1, 0]	48	975000
[1, 1, 1, 0, 1, 1, 1, 1]	47	875000
[1, 1, 1, 1, 0, 1, 1, 1]	42	1025000
[1, 1, 1, 1, 1, 1, 0, 1]	47	1030000
[0, 1, 1, 1, 0, 1, 1, 1]	34	980000
[1, 1, 1, 0, 0, 0, 1, 1]	27	860000
[1, 1, 1, 0, 0, 1, 1, 1]	37	865000
[1, 1, 1, 1, 1, 0, 0, 1]	37	1025000
[0, 1, 1, 1, 1, 1, 1, 0]	40	930000

Tablo 4.3 Kombinasyonlar

Python matematiksel fonksiyonlarla desteklenen bir yazılım dilidir. Permütasyon formülü sayesinde kombinasyonların oluşmaktadır. n elemanlı bir kümeden seçilen r elemanlı tekrarsız permütasyonların toplamı $n \geq r$ olması koşuluyla formülle gerçekleştirilmektedir. Permütasyon formülü aşağıdaki gibidir.

$$P(n, r) = \frac{n!}{(n-r)!}$$

Bir kromozomda 8 adet gen bulunmaktadır. Bu genler de 2 tür şeklinde yazılmaktadır. İkili kodlama şeklinde yazıldıkları için 1 ve 0 halinde yazılmaktadır. N, r'den daha küçük olmamak şartıyla $n = 8$ ve $r = 2$ 'dir. $P(8,2) = 56$. Toplamda kombinasyonların sayısı 56'dır.

Kombinasyonlardan bütçeye ve değere göre çıkan en iyi üç çözümleri alabilmek için kodlamada genetik algoritma ve ana fonksiyon kısmında değişiklikler yapılmaktadır. En iyi üç çözümleri alabilmek için aynı şekilde genetik algoritma ve ana fonksiyon kısmında kısmen değişecektir. Aşağıda Genetik Algoritma gösterilmiştir ve açıklanmıştır.

```
# Genetic Algorithm
1 usage
def genetic_algorithm():
    population = generate_population()
    all_solutions = [] # Store all solutions found
    for _ in range(generations):
        selected_population = selection(population)
        children = []
        while len(children) < population_size - len(selected_population):
            parent1 = random.choice(selected_population)
            parent2 = random.choice(selected_population)
            child1, child2 = crossover(parent1, parent2)
            child1 = mutation(child1)
            child2 = mutation(child2)
            children.append(child1)
            children.append(child2)
        population = selected_population + children
        all_solutions.extend(population) # Add current population to all solutions
    # Remove duplicates from all solutions
    all_solutions = [list(sol) for sol in set(tuple(sol) for sol in all_solutions)]
    return all_solutions
```

Şekil 4.10 Üç Çözüm için Genetik Algoritma

- Başlangıç: İlk çözümler `populasyon_uret()` fonksiyonunu kullanarak oluşturulmaktadır.
- İterasyonlar: Belirli bir jenerasyon sayısı boyunca döngü çalışmaktadır, her jenerasyonu işlemektedir.
- Seçme: Bir sonraki jenerasyon için popülasyonun bir kısmını seçme işlemi yapılmaktadır, `seçme()` fonksiyonunda belirli bir kriter kullanılmaktadır.

- Çaprazlama ve Mutasyon: Seçilmiş ebeveynlerden yeni çocuklar çaprazlama ve mutasyon operasyonlarıyla üretilmektedir.
- Populasyon Güncellemesi: Seçilmiş bireylerle yeni üretilen çocuklar birleştirilerek populasyon güncellenmektedir.
- Tekrarları Kaldırma: Çözümlerde tekrar olmaması için çözümler küme haline getirilmektedir ve tekrar listeye dönüştürülmektedir.

En iyi ve uygun üç çözümü alabilmek için Ana fonksiyonunun değişiklikleri aşağıdaki gibi gösterilmiştir ve açıklanmıştır.

```
# Main function
if __name__ == "__main__":
    all_solutions = genetic_algorithm()
    # Filter valid solutions and calculate their value and weight
    valid_solutions = []
    for solution in all_solutions:
        total_value = sum(items[i]["value"] for i in range(len(items)) if solution[i] == 1)
        total_weight = sum(items[i]["weight"] for i in range(len(items)) if solution[i] == 1)
        if total_weight <= knapsack_capacity and total_value > 0: # Ensure valid and valuable solutions
            weight_value_ratio = total_weight / total_value
            valid_solutions.append((solution, total_value, total_weight, weight_value_ratio))
    # Sort by fitness (value) and then by weight/value ratio
    valid_solutions.sort(key=lambda x: (-x[1], x[3]))
    # Get the top 3 solutions
    best_solutions = valid_solutions[:3]

    for solution, total_value, total_weight, weight_value_ratio in best_solutions:
        print("Solution:", solution)
        print("Total value:", total_value)
        print("Total weight:", total_weight)
        print("Weight/Value Ratio:", weight_value_ratio)
        print()
```

Şekil 4.11 Üç Çözüm için Ana Fonksiyon

- Genetik Algoritma Çalıştırma: `genetik_algoritma()` fonksiyonunu çağırarak genetik algoritmayı çalıştırmaktadır.
- Geçerli Çözümleri Filtreleme: Genetik algoritmadan üretilen tüm çözümler arasından geçerli olanları filtrelemektedir. Geçerli çözümler, çantanın kapasitesine sığacak ve pozitif toplam değere sahip olanlardır.

- Çözümleri Sıralama: Geçerli çözümleri önce toplam değerlerine (fitness) göre azalan düzende, sonra da ağırlık/değer oranına göre artan düzende sıralamaktadır.
- En İyi Çözümleri Seçme: Sıralı çözümlerden en iyi üç çözümünü seçmektedir.
- Sonuçları Yazdırma: En iyi çözümleri, toplam değerlerini, toplam ağırlıklarını ve ağırlık/değer oranlarını birlikte yazdırmaktadır.

Kombinasyonlardan bütçeye ve değere göre çıkan en iyi üç çözümler aşağıdaki gibidir :

1. Çözüm 1: (1, 0, 0, 0, 1, 1, 1, 0) Toplam Değer: 33 Toplam Maaliyet: 65000 Dolar
2. Çözüm 2: (1, 0, 0, 0, 1, 1, 1, 1) Toplam Değer: 37 Toplam Maaliyet: 125000 Dolar
3. Çözüm 3: (1, 0, 0, 0, 1, 1, 0, 1) Toplam Değer: 32 Toplam Maaliyet: 120000 Dolar

5. TARTIŞMA

Oluşturulan kod, sırt çantası probleminde genetik algoritma kullanarak çözüm getirmektedir. Bu problem, öğelerin toplam değerini maksimize ederken toplam ağırlığının belirli bir kapasiteyi aşmamasını sağlayacak en iyi kombinasyonu seçmeyi amaçlamaktadır. Her bir öğe, ad, ağırlık ve değer gibi özelliklerle tanımlanmaktadır. Algoritma, olası çözümlerin bir popülasyonunu başlatır; bu çözümler, her öğenin çantada olup olmadığını gösteren ikili listelerle temsil etmektedir. İterasyonlarla, algoritma her bir çözümün, içerdikleri öğelerin toplam değeri ve toplam ağırlığı üzerinden uygunluk değerini değerlendirmektedir. Ardından, gelecek nesil için ebeveyn olarak hizmet edecek en uygun bireyleri seçer; çaprazlama ve mutasyon işlemleri ise yeni nesil oluşturmaktadır. Bu yeni nesiller, ebeveynlerinden özellikleri devralırken popülasyona çeşitlilik kazandırmaktadır. Algoritma bu işlemi birkaç nesil boyunca sürdürür ve en iyi çözümü bulmaya çalışmaktadır. Sonuç olarak, algoritma tarafından bulunan en iyi çözüm, ilgili toplam değer ve toplam ağırlık bilgisiyle sunulmaktadır. Özetlemek gerekirse, birinci aşamadaki kod uygulandığında elde edilen sonucu çıkmaktadır fakat kombinasyonları görebilmek için genetik algoritma ana fonksiyon kısmında değişiklikler yapılmıştır. Yazılan kod, verilen kısıtlar içinde en iyi kombinasyonu elde etmek için öğelerin değerlerini ve ağırlıklarını dengelemek için genetik algoritmayı kullanmaktadır.

Oluşturulan kod içerisinde uygulanan genetik algoritmada, her bir çözümün uygunluk değeri, sırt çantasına dahil edilen öğelerin toplam değeri esas alınarak hesaplanırken, aynı zamanda toplam ağırlığın sırt çantası kapasitesini aşmaması sağlanmaktadır. En iyi üç çözümler ve her bir çözümün neden en iyi çözümler arasında olduğunu aşağıda incelenmiştir.

Çözüm 2 (1 0 0 0 1 1 1 1) : Üç çözüm arasında en yüksek toplam değere (37) sahiptir. Sırt çantasına, 1, 5, 6, 7 ve 8 numaralı öğeler dahil edilmiştir ve toplam ağırlığı 125.000'dir. Değeri en yüksek olması ve ağırlığın sırt çantası kapasitesi içinde olması sebebiyle, en iyi çözümlerden biri olarak kabul edilmiştir.

Çözüm 1 (1 0 0 0 1 1 1 0) : Toplam değeri 33'tür ve bu, Çözüm 2'den daha düşük ancak Çözüm 3'ten daha yüksektir. Ancak toplam ağırlığı (65.000) üç çözüm arasında en düşüktür. Çözüm 2'den daha düşük bir değere sahip olmasına rağmen, ağırlık-verimlilik oranı tercih edilebilir fakat değer bazında 52/80 maksimum değere kıyaslandığında çok yüksek bir fark çıkmaktadır.

Çözüm 3 (1 0 0 0 1 1 0 1): Toplam değeri 32'dir ve bu, üç çözüm arasında en düşüğüdür. Daha düşük bir değere sahip olmasına rağmen, toplam ağırlığı (120000) hala sırt çantası kapasitesi içindedir. Bu nedenle, değer bakımından Çözümler 1 ve 2 kadar optimal olmasa da, hala en iyi çözümler arasında kabul edilmektedir.

Chen ve diğ. (2020) yaptığı çalışmada, optimum çözümü kapasiteyi değiştirerek bulunduğunu belirtmiştir. Optimum çözümü için kapasiteyi 150, 450 ve 800 olarak değiştirilmiştir. Kıyaslama yapıldığında bütçe 200000, 400000 ve 600000 değişecek olursa, Python'da başka seçenekler gösterilecektir.

Örneğin bütçe 400000 dolar olarsa en iyi çözümler aşağıdaki gibi olacaktır:

- Çözüm 1 : [1, 0, 0, 1, 1, 1, 1, 1], Toplam değer: 42, Toplam ağırlık: 285000
- Çözüm 2 : [1, 1, 0, 0, 1, 1, 1, 1], Toplam değer: 42, Toplam ağırlık: 375000
- Çözüm 3 : [1, 1, 0, 0, 1, 1, 1, 0], Toplam değer: 38, Toplam ağırlık: 315000

Örneğin bütçe 600000 dolar olarsa en iyi çözümler aşağıdaki gibi olacaktır:

- Çözüm 1 : [1, 1, 0, 1, 1, 1, 1, 1], Toplam değer: 47, Toplam ağırlık: 535000
- Çözüm 2 : [1, 1, 0, 1, 1, 1, 1, 0], Toplam değer: 43, Toplam ağırlık: 475000
- Çözüm 3 : [1, 0, 0, 1, 1, 1, 1, 1], Toplam değer: 42, Toplam ağırlık: 285000

Chen ve diğ. (2020) uyguladığı gibi bütçe değiştirilirse, alınacak sonuçlar daha farklı ve çeşitlidir çünkü kapasite arttığı için daha fazla seçenek eklenmektedir. Python'a göre hem 400000 dolar hem de 600000 dolar kapasite için en iyi ve uygun çözümleri, birinci çözümlerdir. 400000 dolar kapasite için en iyi çözümünde toplam değer en yüksek olan ve toplam ağırlık/maliyet en düşük olan seçilmektedir. 600000 dolar kapasite için en iyi çözümünde ise değer, en yüksek olan seçilmektedir ama aynı zamanda toplam ağırlık/maaliyet de en yüksek olan seçilmektedir. Bu da değerler arasında önemli bir fark olduğu içindir. Ek olarak, 400000 dolar kapasite için 6 öğelerden oluşmaktadır. 600000 dolar kapasite için ise 7 öğelerden oluşmaktadır.

Özetlemek gerekirse, 200000 dolar kapasite için en iyi çözümler, dahil edilen öğelerin toplam değeri ile toplam ağırlık arasında bir denge sağlayarak belirlenmektedir ve ağırlığın sırt çantası kapasitesini aşmaması sağlanmaktadır. Çözüm 2, toplam değeri en yüksek olan ve ağırlık kısıtlamasını karşılayan çözümdür. Onu Çözüm 1 ve 3 izler; bu çözümler biraz daha düşük değere sahip olsalar da, ağırlık kriterlerini karşılamaktadır. Ağırlık kriterleri aşmadan karşılanırsa ve değerler arasında nispeten önemli bir fark varsa, o zaman seçim daha kolay olmaktadır. Değerlendirildiğinde aradaki değer farkı 4 puan olması, fazladan bir öğenin eklenmesi anlamına gelmektedir.



6. SONUÇ VE ÖNERİLER

Genetik algoritma tarafından belirlenen en iyi çözüm, 1, 0, 0, 0, 1, 1, 1, 1 olarak ikili kodla temsil edilen çözüm numarası 2'dir. Bu çözümü daha detaylı olarak analiz edilmiştir :

Dahil Edilen Öğeler: Çözüm, gemi tersanesinde lojistik akışları iyileştirmek ve optimize etmek için en faydalı görülen öneriler veya girişimler olan 1, 5, 6, 7 ve 8 numaralı öğeleri içermektedir.

Toplam Değer: Bu çözümün toplam değeri 37'dir ve üç önemli çözüm arasında en yüksektir. Bu, seçilen önerilerin, verimliliği artırma, maliyetleri azaltma veya tersane içindeki operasyonları geliştirme konusunda en yüksek toplam faydayı sunduğunu göstermektedir.

Toplam Ağırlık: Seçilen öğelerin toplam ağırlığı 125,000 dolardır. Bu çözüm önemli bir bütçe kısmını kullanmasına rağmen, 200,000 dolarlık ayrılmış bütçe sınırı içinde kalmaktadır. Bu, seçilen önerilerin mali açıdan uygun olduğunu ve mevcut kaynakları aşmadığını garanti etmektedir.

Uygulanabilirlik ve Uygunluk: Çözüm numarası 2, bütçe içinde kalarak en yüksek değeri sunması nedeniyle en uygun olarak kabul edilmektedir. Bu, faydaları maksimize etme ve maliyetleri etkin bir şekilde yönetme arasında dikkatli bir denge sağladığını göstermektedir.

Önerilen Girişimler: Çözümün ikili temsiline dayanarak, seçilen öneriler 1, 5, 6, 7 ve 8 numaralı öğeleri içermektedir. Bu, RFID, Barkodlama ve Blok Zinciri uygulama, Dijital İkiz teknolojisinden yararlanma, VR (Sanal Gerçeklik) eğitim amacıyla kullanma, personel eğitimlerini kontrol etme ve lojistik süreçlerdeki dar boğazları analiz etme gibi belirli projeler veya girişimlerle ilgili olabilmektedir.

Lojistik Akışları Üzerindeki Etki: Seçilen bu girişimlerin gemi tersanesindeki lojistik akışları optimize etmede önemli pozitif bir etkisi olması beklenmektedir. RFID, Barkodlama ve Blok Zinciri ve Dijital İkiz teknolojileri, veri yönetimini ve görselleştirmeyi iyileştirebilirken, VR eğitimi personel becerilerini ve verimliliğini artırabilmektedir. Ayrıca, eğitimleri kontrol etmek ve darboğazları analiz etmek, lojistik operasyonlardaki verimsizlikleri belirleme ve giderme konusunda yardımcı olabilmektedir.

Çözüm numarası 2, bu tez için en uyarlanabilir çözümleri içermektedir. Diğer seçenekler ise, AGV, 3D baskı ve robotlar kesinlikle verimlidir, ancak oldukça yeni teknolojilerdir. Bu çözümlerin uygulanması için hem zaman hem de para gerektirmektedir. Bu nedenlerden dolayı, tersane sahiplerinin bu çözümleri uygulama konusunda emin olmak istediklerini anlaşılabilmektedir. Bununla birlikte, tersanenin dijitalleştirilmesi yalnızca tersanenin değil, aynı zamanda ilgili tüm lojistik akışlarının da optimize edilmesine yol açmaktadır.

Sonuç olarak, 1, 5, 6, 7 ve 8 numaralı önerilerden oluşan çözüm numarası 2, gemi tersanesindeki lojistik akışları iyileştirmek ve optimize etmek için en uygun kombinasyonu temsil etmektedir. Bu öneriler, ayrılan bütçe içinde en yüksek değeri sunmaktadır ve lojistik zorluklarla başa çıkmak için hem uygulanabilirlik hem de etkinlik sağlamaktadır.



KAYNAKLAR

- Beck R., Boote D. ve diğ. 09-13 Eylül 2012. “ISSC Committee V.8: Sailing Yacht Design.” *18th International Ship & Offshore Structures Congress*, . Rostock, Germany,; Volume 3, s.324-325.
- Brettel M., Friederichsen N., Keller M., & Rosenberg, M. 2014. “How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0.” *Perspective International Journal of Information and Communication Engineering*, International journal of mechanical, industrial science and engineering, 8(1), s. 37–44. İkinci Kaynak: Manning, K., Jang, R., Rajagopalan A., Spoehr, J., Hordacre, A. L., Moretti, C., Howard, S., Yaron, P., - Australian Industrial Transformation Institute -Mart 2021, *Robotics and the Digital Shipyard*, s.1-21
- Buseti, F. 2007. “Genetic Algorithms Overview.” s. 1-13.
- Caprace J.D., Velarde M.G., Petcu C. & Rigo P. 2013 . “Optimization of shipyard allocation and scheduling using a heuristic algorithm.” *Journal of Marine Science and Technology*,s. 1-16.
- Chen, R.C., Saravananarajan, V.S., Dewi, C., Chen, L.S. 2020. “Solving unbounded knapsack problem using evolutionary algorithms with bound constrained strategy.” *International Journal of Applied Science and Engineering*, Chaoyang University of Technology, ISSN: 1727-7841 (Online) Vol.18(1) 2020205 s.1-11.
- Cheng Y.S., Au F.T.K., Tham L.G. & Zeng G.W. 2004. “Optimal and robust design of docking blocks with uncertainty.” *Engineering Structures* 26 (2004) s. 499–510, Department of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, PR China & Department of Civil Engineering, The University of Hong Kong, PR China.
- Clarkson Research Services. 03/2023. İkinci Kaynak: Deniz Ticaret Odası , *Denizcilik Sektör Raporu*, İstanbul (2023), bölüm 2.5, s. 153.
- Coley, D.A.,. 1999. “An Introduction to Genetic Algorithms for Scientists and Engineers.” University of Exeter, World Scientific Publishing Co. Pte. Ltd., ISBN 981-02-3602-6, bölüm 2.7, s.25.
- Colton, T., and L.V Huntzinger. Eylül 2002. “A brief history of shipbuilding in recent times.” The CNA Corporation, Virginia, s. 36.
- Conn, J.F.C. and Pounder, Cuthbert Coulson. 8 Apr. 2024. *Encyclopedia Britannica, Ship Construction*. Erişim Tarihi: 01.06.2024. <https://www.britannica.com/technology/ship-construction>.
- Crawford, A., 2011. “Industrial Robots.” 1st ed., The English Press, Delhi, ISBN: 978-93-81157-97-, s.25.

- Czachorowski, K.V., Kondratenko, Y., Marina, S. 2019. *“The Application of Blockchain Technology in the Maritime Industry.”* University College of Southeast Norway, Nord University Business School, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine, s.574.
- Deepa, S.N., Sivanandam, S.N. 2008. *An introduction to Genetic Algorithms.* Springer Verlag Berlin Heidelberg, ISBN 978-3-540-73189-4, s.425.
- ECORYS. 2009. *“Study on competitiveness of the European shipbuilding industry.”* Final report, s. 1-285, Brüksel.
- Fairchild, M. 12 Eylül 2021. *Top Industries Using Robots, howtorobot.* Erişim Tarihi : 02.06.2024. <https://howtorobot.com/expert-insight/top-industries-using-robots> .
- Fedko, D. 2021. *Virtual and Augmented Reality in Logistics.* Erişim Tarihi : 02.06.2024. <https://wear-studio.com/virtual-and-augmented-reality-in-logistics/> .
- Goldberg, D.E. Ocak 1989. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison – Wesley Publishing Company, INC., Ocak 1989, The University of Alabama, USA, ISBN: 0-201-15767-5, bölüm 1, s.1, & bölüm 4, s.89.
- Gupta, A.K., Arora, S.K., Westcott, J.R. 2017. *“Industrial Automation and Robotics.”* Mercury Learning and Information, Virginia, USA, s.1-560.
- Gupta, S.K. 2021. *“An Overview of Genetic Algorithms: A Structural Analysis.”* International Journal of Innovative Science and Research Technology Volume 6, Issue 5, May – 2021, Gopal Narayan Singh University, India, ISSN No:-2456-2165, s. 1305-1309 .
- Halvorsen, H.P. 2019. *“Python Programming.”* ISBN: 987-82-691106-4-7, s.20.
- Hengst, S. 1999. *Scheepsbouw Deel 1*, Delft University Press ; İkinci Kaynak : (Bergsma J., Pruyn J. & Zalm M.v.d., 2016 - *3D-Printing and the Maritime Construction Sector*, s. 437).
- Hicri Ercili Tersanesi, *Hicri Ercili Tersanesi*, Erişim Tarihi: 02.03.2024. <https://www.hicriercili.com.tr/>.
- Islam, Md.K., Shaikh, Md.A., Habib, Md. M. 2017. *“A Study of Supply Chain Management in Shipbuilding Industry.”* *ICBM 2017 - 1st International Conference on Business & Management.* Department of Shipping Administration, Bangabandhu Sheikh Mujibur Rahman Maritime University, & BRAC Business School, BRAC University, Bangladesh, s.414-419.
- Kanade, Vijay. 06 Eylül 2023. *What Are Genetic Algorithms? Working, Applications, and Examples; Spiceworks,* Erişim Tarihi: 02.06.2024. <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>.
- Kim H., Lee S.S., Park J. & Lee. J.G. 2005. *“A model for a simulation-based shipbuilding system in a shipyard manufacturing process.”* In *Applying digital manufacturing technology to ship production and the maritime environment.* International Journal of Computer Integrated Manufacturing, Vol. 18, No. 6, September 2005, s.427 – 441, Taylor and Francis Group.

- Kinnear, K. E. & Angeline P.J.,. 1994. “*Advances in Genetic Programming.*” Cambridge: MIT Press, s. 3-17.
- Kora P. & Yadlapalli P.,. Mart 2017. “*Crossover Operators in Genetic Algorithms: A Review.*” International Journal of Computer Applications (0975 – 8887) Volume 162 – No 10, s. 34-36.
- Kritzinger W., Matthias K., Traar G., Henjes J., and Sihm W. 2018. “*Digital Twin in manufacturing: A categorical literature review and classification.*” In: Ifac-PapersOnline 51.11, Vienna, Austria, s. 1016–1022.
- Lee, J. ve diğ. 2014. “*Quality Verification of Legacy Data of Manufacturing Information System to Improve Results of Shipyard Manufacturing Logistics Simulation.*” Journal of the Society of Naval Architects of Korea, South Korea, 51(6), s. 510–520.
- Mackenzie, L. 2004. *Inaugural Ship Superintendency Forum 2004, Lloyd’s List Events.* İkinci Kaynak:(Alhouli Y.,2011, *Development of Ship Maintenance Performance Measurement Framework to Assess the Decision Making Process to Optimise in Ship Maintenance Planning*, University of Manchester, PhD, Faculty of Engineering & Physical Sciences, UK, bölüm 2.8, s.47.
- Man K.F., Tang K.S. & Kwong S. 1996. “*Genetic Algorithms: Concepts and Applications.*” IEEE Transactions of Industrial Electronics Vol., 43, no. 5, s. 519-534.
- Mitch, M. 1999 5th printing. “*An introduction to Genetic Algorithms.*” Massachusetts Institute of Technology, USA, ISBN 0–262–13316–4 (HB), 0–262–63185–7(PB), 1996, s.1-162.
- Muther R. & Hales L. 2015. “*Systematic layout planning.*” Fourth edition 2015, MANAGEMENT & INDUSTRIAL RESEARCH PUBLICATIONS, USA.
- OECD, 9 Şubat 2021. “*Peer Review Of the Turkish Shipbuilding Industry.*” C/WP6(2020)11/Final, Executive Summary.
- Pastor A, *AGV Network*, Erişim Tarihi: 02.06.2024. <https://www.agvnetwork.com/agv-cost-estimation-how-much-does-an-automated-guided-vehicle-cost> .
- Pruyn, J., Martinez, J. C., & Molenaar, W. 2010. “*Can RFID be Successfully Used in Shipbuilding?*” Journal of Ship Production and Design, 26(2), s.111-116. İkinci Kaynak : (Mansor M.N., Ustadi M.N. & Shariffuddin M.A., *The Potential of Radio Frequency Identification (RFID) Technology Implementation in Malaysian Shipbuilding Industry*, Kuala Lumpur, Malaysia,2015, 31-36).
- Pujawan, I.N. 2005. “*Supply Chain Management.*” 1st ed. Surabaya: Penerbit Guna Widya, Accessed: Jul. 28, 2020. İkinci Kaynak: Tutuhatunewa A., Maitimu N.E. & Hukunala L.M., *Integration of SCOR Model and AHP to Measure the Shipyard Supply Chain Performance: A Case Study*, Faculty of Engineering, Pattimura Uni, Indonesia, Vol. 25, No. 2, December 2023, ISSN 1411-2485, doi:<https://doi.org/10.9744/jti.25.2.169-178>.
- Ravi Rao. 05 Temmuz2023. *Cobots vs Robots: Understanding the Key Differences and Applications*, WEVOLVER. Erişim Tarihi : 02.06.2024. <https://www.wevolver.com/article/cobot-vs-robot>.
- Rising Max. 11 Ocak 2024. *Rising Max*, Erişim Tarihi : 02.06.2024. <https://risingmax.com/blog/digital-twin-development-cost>.

- Shenoi, R.A., Beck, R., Boote, D. 16-21 August 2009. "ISSC Committee V.8: Sailing Yacht Design.", *17th International Ship & Offshore Structures Congress*, Seoul, Korea, Volume 2, s.489.
- Shetelig, H. 2013. "Shipbuilding Cost Estimation." Norwegian University of Science and Technology, Marine Technology, Norway, bölüm 3.1, s.7.
- Shipbuilding Partners. 15 Eylül 2023. *Shipbuilding Partners, Strategies to Optimize Shipbuilding Supply Chains*. Erişim Tarihi : 01.06.2024. <https://www.linkedin.com/pulse/strategies-optimize-shipbuilding-supply-chains-thehydok/>.
- Shyalika Ch., Engineer — *Big Data & Data Science, Population Initialization in Genetic Algorithms, An Insight to Genetic Algorithms - Part II* 28 Ocak 2019. Erişim Tarihi: 02.06.2024, <https://medium.datadriveninvestor.com/population-initialization-in-genetic-algorithms-ddb037da6773>.
- Song, D.W., Panayides, P.M. 2015. "Maritime Logistics, A guide to contemporary shipping and port management." Kogan Page Limited, ISBN: 978 0 7494 7268 9, bölüm 4,s.63.
- Spoehr, J., Jang, R., Manning, K., Rajagopalan, A., Moretti, C., Hordacre, AL., Howard, S., Yaron, P. & Worrall, L. 2020. *The Digital Shipyard: Opportunities and Challenges*. Adelaide: Australian Industrial Transformation Institute, Flinders University of South Australia, ISBN: 978-0-6488941-2-4, s.1-21
- Stopford, M. 2009. *Maritime Economics*. 3rd ed., Routledge, Oxon, UK, bölüm 1, s1-45.
- Storch, R.L., Hammon, C.P., Bunch, H.M., Moore, R.C. 1995. *Ship Production*. 2nd ed., The Society of Naval Architects & Marine Engineers, NJ, USA, bölüm 1, s.1-30.
- Ulaştırma ve Altyapı Bakanlığı, and Tersaneler ve Kıyı Yapıları Genel Müdürlüğü. 03/2023. İkinci Kaynak : Deniz Ticaret Odası , *Denizcilik Sektör Raporu*, İstanbul (2023), bölüm 2.5, s.147.
- Ullrich, G. 2015. "Automated Guided Vehicle Systems, A Primer With Practical Applications.", Berlin, Heidelberg, Springer Berlin Heidelberg, Imprint: Springer. İkinci Kaynak: Jabrane Z. & Borgemo E., Mayıs 2018, *Evaluating Usage of Automated Guided Vehicles with Respect to Warehouse Layout Changes*, Lund Uni, Department of Engineering Logistics.
- Ullrich, G. 2015 "Automated Guided Vehicle Systems: A Primer with Practical Applications, 2015." Berlin: Springer, s. 38–40, 2nd revised edition. İkinci Kaynak: Anand R., Vantagodi N.V., Shanbhag K.A. & Mahesh M.," *Automated Guided Vehicles by Permanent Magnet Synchronous Motor: Future of In-house Logistics*", Department of Electrical Engineering, PES University, Bangalore & Department of Electrical Engineering, PES Institute of Technology, Bangalore, India, 2019, s.152.
- UNCTAD, 2021, "Review of Maritime Transport." United Nations Publications, USA.
- Urbanyi-Popiołek, I., Klopott, M., 2016, "Container terminals and port city interface – a study of Gdynia and Gdańsk ports." Transportation Research Procedia, 2nd International Conference "Green Cities - Green Logistics for Greener Cities", 2-3 March 2016, Szczecin, Poland, 16; Gdynia Maritime University, Poland; s. 517–526.

Wu, H. 2019. *“Research on Labor Cost Control in Shipbuding Process.”* Jiangsu University of Science and Technology, China, Master of Accounting, s.325-329.



EKLER

Uyarı: Tez çalışmasında kullanılan hesaplamalar, tablolar, şekiller, veriler, kaynak kodları, formlar vb. materyal metin içinde akışı bölecek kadar çoksa bu bölümde ayrı başlıklar halinde ekleyebilirsiniz (Örn: Kaynak Kodlar, Anket Formları, Onam Formu Örneği başlıkları gibi). Ekler kısmına eklenmeyecek ve materyali olmayan tez çalışmaları için **EKLER** bölümü kaldırılabilir. Ekler bölümüne ait yazım kuralları ile ilgili tez yazım kılavuzu incelenmelidir. Bu uyarıyı okuduktan sonra siliniz!



İNTİHAL RAPORU İLK SAYFASI

Pamela Teranaj .pdf

ORJİNALLİK RAPORU

% **7**

BENZERLİK ENDEKSİ

% **6**

İNTERNET KAYNAKLARI

% **2**

YAYINLAR

% **6**

ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	% 3
2	Submitted to Southern Methodist University Öğrenci Ödevi	% 1
3	Submitted to University of Thessaly Öğrenci Ödevi	% 1
4	wenku.csdn.net İnternet Kaynağı	<% 1
5	www.kdnuggets.com İnternet Kaynağı	<% 1
6	Submitted to Purdue University Öğrenci Ödevi	<% 1
7	acikbilim.yok.gov.tr İnternet Kaynağı	<% 1
8	www.hicriercili.com İnternet Kaynağı	<% 1
9	Submitted to University College London Öğrenci Ödevi	<% 1

ETİK KURUL İZİN YAZISI

Uyarı: Canlı denekler üzerinde yapılan tüm arařtırmalar için Etik Kurul Belgesi alınması zorunludur.

- Etik Kurul izni gerekmektedir.
- Etik Kurul izni gerekmemektedir.

Pamela ATES



KURUM İZİNİ YAZILARI

Uyarı: Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

- Kurum izni gerekmektedir.
- Kurum izni gerekmemektedir.

Pamela ATES



PATENT HAKKI İZİNİ

Uyarı: Patent hakkı izni gerektirmeyen tez çalışmaları için bu bölüm ve başlık silinebilir!

