



**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**RPA İÇİN GEREKLİ MİNİMUM ROBOT SAYISININ BELİRLENMESİ
PROBLEMİNE ETKİLİ BİR ÇÖZÜM YÖNTEMİNİN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Anıl ÖZKAPAN

Akıllı Sistemler Mühendisliği Anabilim Dalı

Akıllı Sistemler Mühendisliği Programı

AĞUSTOS 2025

**T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**RPA İÇİN GEREKLİ MİNİMUM ROBOT SAYISININ BELİRLENMESİ
PROBLEMİNE ETKİLİ BİR ÇÖZÜM YÖNTEMİNİN GELİŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

**Anıl ÖZKAPAN
(22434998007)
ORCID: 0000-0002-6565-3534**

**Akıllı Sistemler Mühendisliği Anabilim Dalı
Akıllı Sistemler Mühendisliği Programı**

**Tez Danışmanı: Doç. Dr. Yunus DEMİR
ORCID: 0000-0003-3868-1860**

AĞUSTOS 2025

BTÜ, Lisansüstü Eğitim Enstitüsü'nün 22434998007 numaralı Yüksek Lisans Öğrencisi Anıl ÖZKAPAN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “RPA İÇİN GEREKLİ MİNİMUM ROBOT SAYISININ BELİRLENMESİ PROBLEMİNE ETKİLİ BİR ÇÖZÜM YÖNTEMİNİN GELİŞTİRİLMESİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Yunus DEMİR**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğretim Üyesi Hilal YILMAZ**
Bursa Teknik Üniversitesi

.....
Dr. Öğretim Üyesi İlker KARADAĞ
Sivas Cumhuriyet Üniversitesi

Savunma Tarihi **: 18.09.2025**

Teslim Tarihi **:**



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

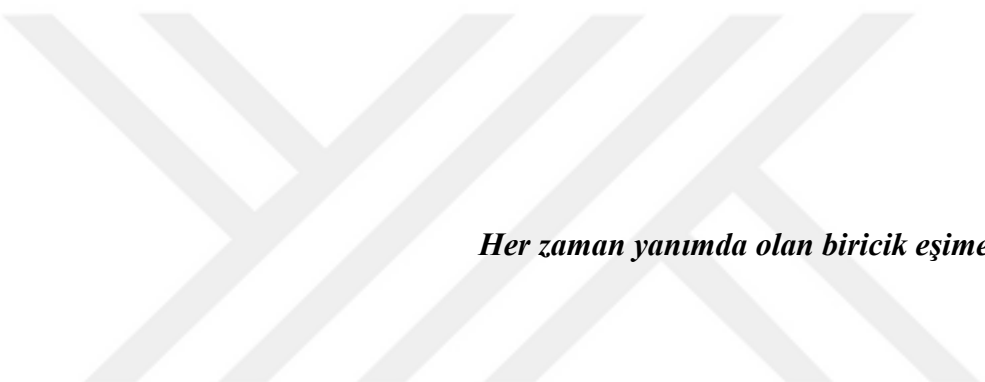
İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Adı Soyadı: Anıl ÖZKAPAN

İmzası:

(Faint, stylized signature watermark)



Her zaman yanımda olan biricik eşime ve aileme,

ÖNSÖZ

Bu tez çalışmasında, Robotic Process Automation (RPA) süreçlerinin verimli bir şekilde yürütülmesi için gerekli minimum robot sayısının belirlenmesi problemine yönelik etkili bir çözüm yöntemi geliştirilmiştir. Günümüzde birçok işletmenin dijital dönüşüm sürecinde kritik bir rol üstlenen RPA uygulamalarının doğru yapılandırılması, zaman ve kaynak yönetimi açısından büyük önem taşımaktadır. Bu doğrultuda yapılan çalışma hem akademik hem de sektörel açıdan fayda sağlamayı hedeflemektedir.

Tez süreci boyunca yaşadığım deneyimler hem mesleki gelişimime hem de kişisel olgunlaşmama katkı sağladı. Zaman zaman zorlayıcı olsa da karşılaştığım her zorluk, sürecin öğretici ve ilham verici yönünü daha fazla hissettirdi. Bu çalışmayı ortaya koyarken hem teknik bilgilerimi pekiştirme hem de analitik düşünme becerilerimi geliştirme fırsatı buldum.

Bu sürecin sağlıklı bir şekilde ilerlemesinde desteklerini esirgemeyen birçok kişiye teşekkür etmek isterim. Öncelikle, akademik birikimi ve yönlendirmeleriyle çalışmama değer katan tez danışmanım Doç. Dr. Yunus Demir'e teşekkür ederim. Ayrıca, manevi desteğiyle yanımda olan aileme ve iş yerimde bana anlayış gösteren tüm çalışma arkadaşlarıma minnettarım.

Ağustos 2025

Anıl ÖZKAPAN
(RPA & Yazılım Geliştirici)

İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ.....	vii
İÇİNDEKİLER	viii
KISALTMALAR	x
ÇİZELGE LİSTESİ.....	xii
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xiv
SUMMARY	xv
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	4
2.1 Robotik Süreç Otomasyonu (RPA): Dijital Dönüşüm Stratejileri, Kapsamlı Sektörel Uygulamaları ve Süreç Optimizasyonu Yaklaşımları	4
3. MATERYAL VE YÖNTEM.....	11
3.1 Robotik Süreç Otomasyonu	11
3.1.1 Robotik süreç otomasyonu: tanım ve kapsam.....	11
3.1.2 Robotik süreç otomasyonunun tarihsel gelişimi	12
3.1.3 Robotik süreç otomasyonu teknolojisinin işletmelerdeki yeri ve önemi ..	13
3.1.4 Robotik süreç otomasyonu uygulamaları için teknik gereksinimler.....	15
3.1.5 Robotik süreç otomasyonu adımları ve uygulama aşamaları.....	17
3.1.6 Süreçlerin seçim kriterleri ve uygunluk analizi	19
3.1.7 Robotik süreç otomasyonu uygulamalarında karşılaşılan zorluklar	22
3.1.8 Robotik süreç otomasyonu ve yapay zekâ etkileşimi	23
3.1.9 Robotik süreç otomasyonunun maliyeti ve maliyet azaltma stratejileri ...	25
3.1.10 Maliyet azaltma stratejileri.....	26
3.2 Kombinatorial Optimizasyon: Genel Bakış	27
3.2.1 Kombinatorial optimizasyon problemlerine örnekler	28
3.2.1.1 Sırt çantası problemi (knapsack problem).....	28
3.2.1.2 Çizelgeleme problemleri (scheduling problems)	28
3.2.1.3 Araç rotalama problemi (vehicle routing problem)	29
3.2.2 Kombinatorial optimizasyon ve çözüm yöntemleri	29
3.2.2.1 Kesin çözüm yöntemleri	29
3.2.2.2 Meta-sezgisel yöntemler	36
3.2.2.3 Hibrit yöntemler	37
3.3 İş Çizelgeleme (Job Scheduling Problem)	37
3.3.1 Tek makine çizelgeleme.....	38
3.3.2 Paralel makine çizelgeleme.....	39
3.3.3 Atölye tipi çizelgeleme	40
4. PROBLEM TANIMI	42
4.1 Robotik Süreç Otomasyonunda Minimum Robot Sayısı Probleminin Kapsamlı Tanımı ve Matematiksel Formülasyonu.....	42
4.1.1 Problemin karma tamsayılı modeli	44
4.1.2 Geliştirilen karma tamsayılı modelin koşumu ve değerlendirilmesi	50

5. SONUÇ VE ÖNERİLER.....	51
KAYNAKLAR	53
ÖZGEÇMİŞ.....	60



KISALTMALAR

ACO	: Ant Colony Optimization
AI	: Artificial Intelligence
ANCOPUR	: Analysing and Comparing Emerging Publications on RPA
API	: Application Programming Interface
B&B	: Branch and Bound
BPM	: İş Süreçleri Yönetimi (Business Process Management)
CRM	: Customer Relationship Management
DP	: Dinamik Programlama
ERP	: Enterprise Resource Planning
FTP	: File Transfer Protocol
GRASP	: Greedy Randomized Adaptive Search Procedure
GUI	: Graphical User Interface
HPM	: Human Resource Management
IA	: Intelligent Automation
ID	: Identification Number
ILP	: Tamsayılı Doğrusal Programlama
IoT	: Nesnelerin İnterneti
IP	: Integer Programming
IPA	: Intelligent Process Automation
IT	: Information Technologies
İK	: İnsan Kaynakları
JSSP	: Job Shop Scheduling Problem
KVKK	: Kişisel Verilerin Korunması Kanunu
LS	: Local Search
MIP	: Mixed Integer Programming
ML	: Machine Learning
NLP	: Natural Language Process
OCR	: Optik Karakter Tanıma
RPA	: Robotic Process Automation
RSO	: Robotik Süreç Otomasyonu

SA	: Simulated Annealing
SLR	: Sistematik Literatür İncelemesi
SMS	: Systematic Mapping Study
TA	: Threshold Accepting
TSP	: Traveling Salesperson Problem
VRP	: Vehicle Routing Problem
YZ	: Yapay Zekâ



ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1: RPA ve yapay zekâ arasındaki farklar.	24
Çizelge 4.1: Gerekli yazılım ve lisans bedelleri.	44
Çizelge 4.2: Açıklayıcı örnek işlem türleri.	48



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1: RPA pazar büyüklüğü	13
Şekil 3.2: Robotik süreç otomasyonu için şekil	16
Şekil 4.1: Bir işlemin zaman eksenindeki pozisyonu.	43
Şekil 4.2: Açıklayıcı örneğe ait gantt grafiği.	49



RPA İÇİN GEREKLİ MİNİMUM ROBOT SAYISININ BELİRLENMESİ PROBLEMİNE ETKİLİ BİR ÇÖZÜM YÖNTEMİNİN GELİŞTİRİLMESİ

ÖZET

Robotik Süreç Otomasyonu (RSO) veya tanınan ismiyle RPA (Robotic Process Automation) günümüzde birçok şirket tarafından iş süreçlerini otomatikleştirmek için kullanılan ve günden güne yaygınlaşan bir teknolojidir. Temelde bu teknoloji insanların bilgisayar üzerinde gerçekleştirdiği rutinleşmiş, tekrarlayan ve belirli kurallara tabi olan işlerini otomatikleştirmek için kullanılır. Bu sayede çalışanlar daha planlı görevlere odaklanırken Robotik Süreç Otomasyonu yazılım robotları (“ajanları, botları”) onlar yerine manuel işleri hızlı, kesintisiz ve hatasız şekilde gerçekleştirir. Robotlar çalışabilmek için bir ortama ihtiyaç duyarlar. Bu ortamlar birçok farklı kriter göz önünde bulundurularak seçilir. Kullanılan ortamlar yerel (on-premises) ortam, bulut (cloud) ortamı veya hibrit bir ortam olabilir. RPA robotları farklı çalışma ortamlarında kullanılabilirken, maliyetleri seçilen lisans türüne, robot sayısına veya ek hizmetlere bağlı olarak değişiklik gösterebilir. Şirketlerin rekabet ve maliyet baskısı altında minimum kaynak kullanarak maksimum çıktıyı elde etme ihtiyacı her alanda olduğu gibi RPA robotlarının sayısının optimize edilmesini de önemli hale getirmektedir. Bu tez kapsamında süreç sayısı, işlem süresi, öncelik seviyesi, yineleme düzeni vb. gibi faktörler göz önünde bulundurularak minimum robot lisansı ile maksimum verim elde edilmesi hedeflenmektedir. Tezin giriş bölümünde tezin amacından görevlerin en verimli şekilde yerine getirilmesi için RPA robotlarının sayısının ve kullanımının optimize edilmesi gerektiğinden bahsedilmiştir. Sonrasında olası çözümler arasında en uygun olanın bulunması, iş süreçlerinin en verimli şekilde yürütülmesi ve robot sayısının optimize edilmesi için kullanılacak kombinatoriyal optimizasyondan yaklaşımından bahsedilmiştir. İş çizelgeleme kapsamında yeni görevlerin dinamik bir biçimde atanması ve mevcut iş yükünün yeniden yapılandırılması amaçlanmaktadır. Geliştirilen karma tam sayılı programlama yaklaşımıyla robot görevleri arasında tam sayı kısıtlamaları ile atanma yapılması sağlanmaya çalışılmıştır. Sonuç olarak RPA teknolojisi doğru şekilde kullanıldığında operasyonel maliyetleri düşürmesi, hataları azaltması ve iş verimliliğini artırması beklenmektedir. Bu tezde minimum robot kaynağı kullanarak robotların uygun şekilde çizelgelenmesini sağlayarak hem maliyetlerin azaltılması hem verimliliğin artırılması beklenmektedir.

Anahtar kelimeler: RPA, İş çizelgeleme, Karma tamsayılı modelleme, Optimizasyon.

DEVELOPING AN EFFECTIVE SOLUTION METHOD FOR DETERMINING THE MINIMUM REQUIRED NUMBER OF ROBOTS FOR RPA

SUMMARY

Robotic Process Automation (RPA) is a technology increasingly adopted by many companies to automate business processes. Fundamentally, RPA automates routine, repetitive, and rule-based tasks traditionally performed by humans on computers. By implementing this technology, employees can shift their focus to more strategic tasks while RPA software robots (often referred to as "agents" or "bots") perform manual work rapidly, seamlessly, and without error. For robots to operate, they require an environment, which is chosen based on various criteria. These environments may be local (on-premises), cloud-based, or hybrid. Although RPA robots can operate in different environments, costs vary depending on the license type, the number of robots, and additional services utilized. In today's competitive and cost-sensitive market, companies are under pressure to maximize output with minimal resources, making the optimization of RPA robot numbers crucial. This thesis aims to achieve maximum efficiency with minimal robot licensing by considering factors such as the number of processes, processing time, priority level, and repetition frequency, among others. In the introductory chapter, the purpose of the thesis is discussed, emphasizing the need to optimize the number and deployment of RPA robots to ensure task efficiency. Following this, the study examines combinatorial optimization approaches to identify the most effective solution among possible alternatives for executing business processes and optimizing robot numbers. In the context of job scheduling, the goal is to dynamically assign new tasks and reconfigure existing workloads. It has been sought to enable the assignment among robot tasks with integer constraints through the developed mixed-integer programming approach. In conclusion, RPA technology, when implemented correctly, is expected to reduce operational costs, minimize errors, and improve work efficiency. This thesis aims to enable cost reduction and enhanced productivity by ensuring optimal scheduling of robots with minimal resource usage.

Keywords: RPA, Jop scheduling, Mixed-integer programming, Optimization.

1. GİRİŞ

Dijital dönüşüm çağında, işletmelerin karşılaştığı artan rekabet baskıları ve verimlilik talepleri, süreç otomasyonunu stratejik bir öncelik haline getirmiştir. Bu bağlamda, Robotik Süreç Otomasyonu (RPA), tekrarlayan ve kural tabanlı görevlerin insan müdahalesine gerek kalmadan yazılım robotları aracılığıyla gerçekleştirilmesini sağlayan yenilikçi bir otomasyon teknolojisi olarak dikkat çekmektedir. RPA, yalnızca zaman ve maliyet açısından önemli avantajlar sunmakla kalmayıp, aynı zamanda operasyonel hataları azaltarak süreçlerin kalitesini de belirgin şekilde iyileştirmektedir [1].

RPA sistemlerinin temel amacı, manuel olarak gerçekleştirilen standart operasyonları yazılım robotları aracılığıyla otomatikleştirerek, insan kaynağının daha stratejik ve katma değer yaratan işlere yönlendirilmesini sağlamaktır. RPA uygulamaları, bilgi sistemleri ile kullanıcı ara yüzleri (GUI) üzerinden etkileşime girerek; veri girişi, form doldurma, belge işleme, e-posta yönetimi gibi faaliyetleri, insan davranışlarını taklit eden bir yaklaşımla yerine getirmektedir. [2]. Bu kapsamda, robotlar insanların yerine çalışmakla kalmayıp, yorulmadan, hata yapmadan ve belirli görevlerde yüksek doğrulukla işlem gerçekleştirebilmektedir.

RPA teknolojisinin hızla yükselişi, yalnızca bireysel yazılım çözümleri geliştiren firmaların değil, aynı zamanda akademik çevrelerin de ilgisini çekmiştir. Bu bağlamda gerçekleştirilen sistematik literatür taramaları, RPA'nın işletmelerde geniş bir kullanım alanı bulmasına rağmen, konuya ilişkin bilimsel araştırmaların henüz sınırlı bir düzeyde kaldığını ortaya koymaktadır [3]. Bu durum, RPA teknolojisinin hem teorik çerçevesi hem de uygulama alanları açısından daha kapsamlı ve derinlemesine araştırmalara ihtiyaç duyduğunu göstermektedir.

RPA'nın uygulanabilirliğini belirleyen temel faktörlerden biri, süreçlerin yapısal ve tekrarlı nitelikte olmasıdır. Ancak, her sürecin otomatikleştirilmesi hem uygun hem de ekonomik olmayabilir. Bu nedenle, hangi süreçlerin RPA ile optimize edilebileceğinin belirlenmesi, uygulamanın başarısı açısından kritik bir öneme sahiptir. Ayrıca,

süreçlerin otomasyonu sırasında yazılım robotlarının sayısı da önemli bir maliyet faktörü olarak değerlendirilmelidir. Yazılım robotları genellikle lisans temelli olarak çalıştığından, kullanılacak minimum robot sayısının belirlenmesi, operasyonel verimlilik ve ekonomik sürdürülebilirlik açısından büyük bir öncelik taşımaktadır [4].

Bu çerçevede geliştirilen çözüm yöntemleri, çizelgeleme algoritmaları, sezgisel ve meta-sezgisel optimizasyon yaklaşımları ile tamsayılı doğrusal programlama modellemelerini içermektedir. Özellikle, farklı iş türlerinin işlem süreleri ve zaman dilimlerine göre robotlara atanması problemi, mevcut literatürde hem teorik hem de pratik açıdan detaylı bir şekilde ele alınmaktadır [5]. Bu yaklaşımlar, işlem maliyetlerinin minimize edilmesi, robot kaynaklarının etkin kullanımı ve işlem sürelerinin dengelenmesi gibi çok kriterli hedeflere odaklanmaktadır.

Sonuç olarak, RPA teknolojisi yalnızca süreç otomasyonu alanında değil, aynı zamanda kaynak planlaması ve optimizasyon problemlerinin çözümünde de yeni fırsatlar sunmaktadır. Bu potansiyelin etkin bir şekilde kullanılabilmesi için sistematik ve analitik yaklaşımlarla desteklenen yöntemlerin geliştirilmesi gerekmektedir. Bu bağlamda, yapılan çalışma, RPA'nın operasyonel verimliliğe katkı sağlarken, aynı zamanda yazılım robotlarının optimizasyonuna ilişkin önemli bir konuyu da gündeme getirmektedir.

Bu tez çalışmasının temel amacı, RPA teknolojisinin uygulama sürecinde karşılaşılan en önemli optimizasyon problemlerinden biri olan “minimum robot sayısının belirlenmesi” sorununa etkin ve uygulanabilir bir çözüm geliştirmektir. RPA uygulamalarında kullanılan yazılım robotlarının lisans, bakım ve yönetim maliyetleri göz önünde bulundurulduğunda, süreçlerin başarılı bir şekilde otomatikleştirilmesinin yanı sıra, bu sürecin ekonomik sürdürülebilirlik açısından da sağlanması büyük önem taşımaktadır. Bu bağlamda, RPA kapsamında gerekli robot sayısının optimum seviyede belirlenmesi hem operasyonel verimlilik hem de maliyet etkinliği açısından stratejik bir gereklilik olarak öne çıkmaktadır.

Literatürde yapılan çeşitli çalışmalar, RPA'nın farklı iş türleri için çizelgeleme problemleri bağlamında ele alındığını ve bu problemin çoğunlukla çok kriterli optimizasyon modelleriyle çözüldüğünü ortaya koymaktadır. Bu modeller, işlem süreleri, işlem türleri, işlem öncelikleri ve mevcut zaman dilimleri gibi farklı parametreleri dikkate alarak, işlem yükünü yazılım robotları arasında optimal bir

şekilde dağıtmayı amaçlamaktadır. Ancak, bu çalışmalarda da vurgulanan bir diğer önemli nokta, robot sayısının gereksiz bir şekilde fazla tahsis edilmesinin kaynak israfına yol açmakla birlikte, lisans maliyetlerini de artırdığıdır. Bu bağlamda, kullanılacak minimum robot sayısı ile tüm işlemlerin zamanında ve kesintisiz bir biçimde gerçekleştirilmesi, optimizasyon probleminin temel çözüm hedefini oluşturmaktadır.

Tez kapsamında ele alınan problem, belirli işlem tiplerinin farklı zaman dilimlerinde ve farklı işlem süreleriyle gerçekleştirilmesini içermektedir. Her bir işlem tipi, belirli bir süre boyunca işlenmek üzere bir robota atanmakta ve her robot, aynı anda yalnızca bir işlem gerçekleştirebilme kapasitesine sahiptir. Bu durum, problem yapısını klasik iş çizelgeleme problemleri ile benzer hale getirmektedir.

Bu tezde önerilen çözüm yöntemi, karma tamsayılı programlama (MIP) çerçevesinde geliştirilecektir. Gerçek dünya senaryolarına uygun olarak oluşturulan veri setleriyle gerçekleştirilen testler sonucunda, modelin minimum robot sayısını belirlemedeki başarısı detaylı bir şekilde incelenecektir.

Sonuç olarak, bu tez çalışmasıyla, literatürde sınırlı sayıda ele alınmış olan "minimum robot sayısı" belirleme problemini kapsamlı bir biçimde analiz eden hem teorik hem de pratik geçerliliği yüksek bir yöntem önerilmesi hedeflenmektedir. Bu yönüyle çalışma, RPA teknolojisinin uygulama verimliliğini artırmakla kalmayıp, aynı zamanda bilişim sistemleri ve optimizasyon mühendisliği alanlarında yeni araştırma perspektifleri de sunmayı amaçlamaktadır.

2. LİTERATÜR TARAMASI

Bu bölüm altında RPA alanındaki literatürü, uygulama yaklaşımlarını, faydalarını, karşılaşılan zorlukları ve gelecekteki araştırma boşluklarını kapsayan kapsamlı bir genel bakış sunulacaktır. Son olarak, RPA pazarının gelecekteki büyüme beklentileri ve tek makine çizelgeleme problemlerinin çözümüne yönelik matematiksel modeller ve algoritmalar gibi ilgili optimizasyon konuları da ele alınacaktır.

2.1 Robotik Süreç Otomasyonu (RPA): Dijital Dönüşüm Stratejileri, Kapsamlı Sektörel Uygulamaları ve Süreç Optimizasyonu Yaklaşımları

Operasyonel verimliliği artırmak, pazar erişimini genişletmek ve rekabet gücünü yükseltmek, işletmelerin temel stratejik hedefleri arasında yer almaktadır. Bu hedeflere ulaşmak için işletmeler, tarihsel süreçte değişen yöntem ve araçlar kullanmıştır. Günümüzde ise dijital çağın getirdiği yenilikler, işletmelerin bu amaçlara ulaşmasında belirleyici bir rol oynamaktadır. Özellikle dijital dönüşüm, işletmelerin dijital teknolojileri benimseyerek iş süreçlerini, ürünlerini ve hizmetlerini yeniden yapılandırmalarını sağlamaktadır [6,7]. Dijital dönüşüm, işletmelerin tüm fonksiyonlarında köklü değişiklikler yaparken, RPA bu dönüşümün özellikle ofis süreçlerine odaklanan bir boyutunu temsil etmektedir. RPA, tekrarlayan ve kural tabanlı iş süreçlerini otomatikleştirerek operasyonel verimliliği artırmakta, hata oranlarını azaltmakta ve çalışanların daha değerli görevlere odaklanmasına olanak tanımaktadır [8].

Ofis süreçlerinde daha çevik, verimli ve sürdürülebilir hale gelmesinde kritik bir rol oynayan RPA, son on yılda hem akademik araştırmacılar hem de endüstri uygulayıcıları tarafından kapsamlı bir şekilde incelenen ve üzerinde yoğun çalışmalar gerçekleştirilen bir disiplin haline gelmiştir. Bu teknoloji, yazılım botlarını kullanarak kural tabanlı iş süreçlerinin ve görevlerinin otomasyonunu sağlamaktadır [9]. Bu botlar bir bilgisayar yardımıyla, veri girilirken veya veriler düzenlenirken çalışanın ne yapacağını taklit ederler [10].

Robotik Süreç Otomasyonu (RPA), çeşitli sektörlerde önemli bir ivme kazanmış ve tekrarlayan görevleri otomatikleştirerek operasyonel verimliliği artırma potansiyeli sunmuştur. Örneğin bankacılık ve finans endüstrisinde RPA, muhasebe işlemleri [11], finansal portföy yönetimi [12], finansal belgelerin işlenmesi ve raporlama [13], risk değerlendirme süreçleri [14], gibi tekrarlayan görevlerin otomatikleştirilmesinde önemli bir rol oynamaktadır. Bu sayede, hata oranları azalmakta ve süreçlerin hızlanması sağlanmaktadır. Sağlık sektöründe RPA; sigorta süreçleri, tanısal görüntüleme analizi [15], acil durumlarda sağlık hizmetlerinde [16], önemli rol üstlenmektedir. Perakende sektöründe özellikle fiyatlandırma stratejileri [17], arka ofis süreçleri [18] ve envanter yönetiminde RPA çalışmaları karşımıza çıkmaktadır. Eğitim sektöründe RPA, öğretim süreçlerinin desteklenmesinde [19], öğrenci kaydı, devam takibi ve notlandırma gibi birçok idari süreçlerde [20], üniversite sınav sonuçlarının analiz sürecinde [21], e-öğrenme platformlarında (ör. Udemy, Coursera) kurs arama sürecini otomatikleştirilmesinde [22] ve burs başvuru süreçleri [23] gibi birçok başlık altında kullanılmaktadır. RPA, insan kaynakları (İK) ve işe alım alanında dönüşüm yapan bir teknoloji olarak ortaya çıkmıştır; süreçleri basitleştirerek operasyonel verimliliği artırmaktadır. İK analitiğinde veri görselleştirmenin önemini ve operasyonları kolaylaştırmada RPA'nın faydalarını tartışmaktadır [24]. RPA'nın aday taraması, mülakatların planlanması ve başvuranlarla iletişimin yönetilmesi gibi görevleri otomatikleştirebileceğini belirtmektedirler ve böylece RPA'ın işe alım sürecini hızlandırdığını vurgulamaktadırlar [25]. Çalışmada özellikle çalışan bilgilerini (maaş, pozisyon, performans değerlendirmeleri) toplama, saklama ve erişim gibi modüller üzerinde yoğunlaşmıştır. Makalede, RPA'nın insan kaynakları yönetimi süreçlerinden bordro yönetimi (maaş, prim, vergi gibi ödemelerin otomatik hesaplanması), işe alım ve oryantasyon (başvuru takibi, mülakat planlama, teklif mektupları) alanlarında nasıl kullanılabileceğini açıklamaktadır. Mhaske et al. [26] tarafından yapılan çalışmada ise PrimeBot adlı bir yazılım robotunun geliştirilmesini ve işe alım süreçlerinde (özgeçmiş analizi ve sıralama) nasıl kullanılacağı açıklanmaktadır.

Sigorta sektöründe RPA entegrasyonu, geleneksel olarak önemli zaman ve kaynak tüketen arka ofis süreçlerinin otomatikleştirilmesini sağlamıştır. Gelişmiş veri analitiği ve yapay zekâ (YZ) yeteneklerinin uygulanması ile RPA'nın sigorta sektöründeki rolü,

sadece görev otomasyonunun ötesine geçmektedir [27,28]. Bu gelişim kişiselleştirilmiş sigorta ürünleri ve hizmetleri sunmayı mümkün kılmaktadır [29].

Wiktorsson'un çalışması, sigortacılık sektöründe BPM ve akıllı otomasyonun entegrasyonu konusunda hem teorik hem de pratik katkılar sağlamaktadır [30]. Sigortacılık sektöründe iş süreçleri yönetimi (BPM) ve akıllı otomasyon uygulamalarını incelemektedir. Özellikle, Nova Claims adlı bir platformun geliştirilmesi ve etkileri üzerine odaklanmıştır. Nova Claims, sigorta hasar süreçlerini otomatikleştirerek iş akışlarını iyileştirmeyi, veri transferlerini kolaylaştırmayı ve hasar yönetimini standartlaştırmayı amaçlamaktadır. Çalışma, farklı paydaşların (brokerlar, sigorta şirketleri, müşteriler ve iç personel) bu platformdan algıladığı değerleri analiz etmiş ve akıllı otomasyonun BPM süreçlerine nasıl entegre edilebileceğini araştırmıştır [27]. Başka bir çalışmada Generative AI (Üretici Yapay Zekâ) ve RPA entegrasyonunun sigorta sektöründeki hasar süreçlerini nasıl optimize edebileceğini incelenmektedir. Çalışma, özellikle karmaşık ve nadir görülen hasar senaryolarını ele alarak operasyonel verimliliği, doğruluğu ve ölçeklenebilirliği artırmayı hedeflemektedir [28]. Diğer yapılan çalışmada hayat sigortası sektöründe otomasyon teknolojilerinin etkisini incelemekte ve bu teknolojilerin operasyonel süreçleri nasıl dönüştürdüğünü ele alınmaktadır. RPA, Yapay Zekâ (AI), Makine Öğrenimi (ML) ve Blockchain gibi teknolojilerin, sigorta sektöründe, hasar yönetimi, poliçe yönetimi ve müşteri hizmetleri gibi alanlardaki uygulamaları detaylandırılmıştır.

Bu alanda yapılmış birçok literatür taraması bulunmaktadır. Bhardwaj, RPA teknolojisinin iş operasyonlarındaki rolünü, çağdaş eğilimleri ve içgörülerini sistematik bir şekilde incelemektedir [31]. Yazarlar, 2016-2022 yılları arasında yayımlanmış 62 araştırma makalesinden elde edilen verilerle literatürü belirli temalar ve kategoriler altında sınıflandırmışlardır. Bu sınıflandırma RPA'nın temel ilkelerini, uygulama alanlarını, avantajlarını, zorluklarını ve gelecekteki potansiyelini içermektedir. Ayrıca, RPA'nın farklı sektörlerdeki uygulamalarına dair örnekler de sunulmuştur. RPA'nın iş süreçlerini dönüştürme potansiyelini vurgulamakta ve bu teknolojinin başarılı bir şekilde uygulanması için dikkat edilmesi gereken adımları detaylandırmaktadır. RPA'nın gelecekte daha akıllı ve bütünleşmiş çözümlerle iş dünyasında daha geniş bir yer bulacağı öngörülmektedir. Bu makale, RPA teknolojisinin tıbbi ve sağlık hizmetleri yönetiminde sürdürülebilir bir şekilde uygulanmasını inceleyen bir literatür

taramasıdır. Patrício et al. [32], RPA'nın ekonomik ve sosyal sürdürülebilirlik açısından faydalarını ele alırken çevresel etkilerle ilgili araştırmaların eksikliğini vurgulamaktadır. Costa et al. [33], RPA alanında yapılmış çalışmaları sentezlemek için Sistemik Literatür İncelemesi (SLR) yaklaşımını kullanmıştır. Yazarlar çalışmalarında başarılı RPA vakalarının uygulama yaklaşımlarını, gözlemlenen faydaları, işletmelerin karşılaştığı zorlukları ve mevcut literatürdeki araştırma boşluklarını belirlemiştir. Gajjar et al. [34], RPA'nın güvenliği üzerine sistematik bir literatür taraması yapmışlardır. Yazarlar, RPA'nın güvenlik risklerini ve bu risklerin nasıl yönetileceğini incelemişlerdir. Çalışmada, genel olarak güvenlik konularının önemi vurgulanmaktadır. Siderska et al. [35], RPA teknolojisinin Intelligent Process Automation (IPA)'ya dönüşüm sürecini, bu süreçte karşılaşılan zorlukları ve gelecekteki araştırma alanlarını ele alan bir literatür incelemesidir. Çalışma, RPA'nın yapay zekâ teknolojileriyle entegrasyonu sayesinde daha karmaşık ve bilişsel görevleri yerine getirebilen IPA'ya evrildiğini vurgulamaktadır. Çalışma, RPA'dan IPA'ya geçiş sürecini anlamak isteyen araştırmacılar ve uygulayıcılar için kapsamlı bir rehber niteliğindedir. Stravinskienė and Serafinas D. [36] süreç yönetimi ve RPA arasındaki ilişkiyi inceleyen bir sistematik literatür taraması sunmuşlardır. Çalışma, süreç yönetimi ve RPA arasındaki ilişkiyi yönetsel bir bakış açısıyla sistematik olarak analiz ederek, daha az araştırılmış alanları belirlemeyi ve gelecekteki araştırmalar için öneriler sunmayı amaçlıyor. Santos et al. [37], RPA'nın iş süreçlerindeki uygulamalarını anlamak ve başarılı bir şekilde uygulanabilmesi için bir rehber sunmak amacıyla yapılmış sistematik bir literatür taraması yapmıştır. Çalışma, RPA'nın faydalarını, zorluklarını ve uygulama kriterlerini detaylı bir şekilde ele alarak hem akademik hem de pratik katkılar sağlamaktadır. Çalışmada ayrıca RPA'nın iş süreçleri yönetimi ile ilişkisi, veri bilimiyle entegrasyonu ve gelecekteki fırsatları incelenmiştir [38]. Bir dizi önceden tanımlanmış araştırma sorusuna cevaplar şeklinde düzenlemiş, yapılandırılmış bir literatür incelemesi (*structured literature review*) sunmuşlardır. Çalışmada RPA'nın ne anlama geldiği, temel faydalar ve potansiyel yetenekler, RPA hazırlığı ve RPA'yı destekleyen metodolojiler ve teknolojiler temaları boyunca raporlanmıştır. Yazarlar çalışmalarında, şu anda RPA'nın temelini oluşturan teknikleri inceleyen araştırma literatürünün yetersiz olduğunu tespit etmiştir [39]. Hem akademik literatürü hem de endüstriyel çözümleri inceleyen bir sistematik haritalama çalışması (Systematic Mapping Study - SMS) yapmışlardır. Çalışmada RPA alanındaki durum ve boşlukları kapsamlı bir şekilde analiz edilmiştir [40]. RPA

literatürünü analiz etmek ve karşılaştırmak için ANCOPUR (ANalysing and COMparing emerging PUBlications on RPA) adlı bir çerçeve geliştirmiştir. Sundukları bu literatür çalışmasında yazarlar, RPA'nın tanımı, etkileri, uygulama alanları ve diğer teknolojilerle olan farklarını ele almışlardır. Ayrıca, RPA projelerinin uygulanmasını iyileştirmek için kullanılan yöntemler ve RPA'nın yapay zekâ ile nasıl birleştirilebileceği gibi konulara da odaklanılmıştır. Hirve et al. tarafından yapılan çalışmada [41] İnsan kaynakları yönetiminde dijital dönüşümün bir parçası olarak yapay zekâ, RPA ve Nesnelerin İnterneti (IoT) teknolojilerinin kullanımını incelemiştir. Çalışma, bu teknolojilerin insan kaynakları yönetimi üzerindeki etkilerini, faydalarını, zorluklarını ve gelecekteki potansiyel uygulamalarını ele almıştır.

Patrício et al. tarafından yapılan çalışmada [32] RPA teknolojisinin sürdürülebilir bir şekilde uygulanabilirliğini incelemekte ve kullanıcı gereksinimlerini dikkate alarak çok amaçlı bir matematiksel model geliştirmektedir. Çalışma, RPA'nın ekonomik, çevresel ve sosyal sürdürülebilirlik ilkelerine uygun olarak nasıl optimize edilebileceğini araştırmaktadır. Model, RPA projelerinin maliyet minimizasyonu, makespan (tamamlanma süresi) minimizasyonu ve ortalama iş yükü dengesi gibi hedeflere göre optimize edilmesini sağlar. Weighted Sum ve Tchebycheff yöntemleri kullanılarak farklı senaryolar için çözümler üretilmiştir. Bu çalışma, RPA'nın sürdürülebilir uygulaması için kapsamlı bir model sunarak ekonomik, çevresel ve sosyal faydalar sağlamaktadır. Çalışma, RPA'nın yalnızca verimlilik artırıcı bir araç değil, aynı zamanda sürdürülebilirlik hedeflerine ulaşmada stratejik bir araç olarak kullanılabileceğini göstermektedir.

Séguin et al. [4] RPA problemini ele alarak, finansal işlemleri yazılım robotlarına atamak için kullanılan bir yöntem sunmaktadır. Amaç, işlemleri en az sayıda yazılım robotu kullanarak tamamlamaktır. Her yazılım robotu bir lisans maliyeti gerektirdiğinden, robot sayısını minimize etmek önemlidir. Çalışma, özellikle finans sektöründe RPA'nın uygulanmasını ve optimizasyonunu incelemektedir.

Tremblay et al. tarafından yapılan çalışmada [5], RPA problemini çözmek için yeni bir yöntem sunmaktadır. Amaç, finansal işlemleri yazılım robotlarına minimum maliyetle atamaktır. Her robotun lisans maliyeti düşünüldüğünde amaç işlemleri en az sayıda robotla tamamlamayı hedefler. Problem, bir iki parçalı graflar (bipartite graf) olarak modellenmiştir. İşlem türleri ve zaman dilimleri düğümler olarak temsil

etmektedir. Grafin kenar ağırlıklarını belirlemek için bir elektrik devresi analojisi (Etkin Direnç) kullanılmıştır. Bu metrik, düğümler arasındaki "zorluk" seviyesini ölçmektedir.

Benkalai et al. tarafından yapılan çalışmada [42], RPA problemini çözmek için iki farklı yöntem sunmaktadır. Birinci yöntemde İki Aşamalı Tamsayılı Doğrusal Programlama geliştirilmiştir. İkinci yöntem olarak Problem, bir ağ akışı modeli olarak formüle edilmiştir. Amaç, finansal işlemleri yazılım robotlarına atayarak toplam robot sayısını minimize etmektir. Ford-Fulkerson Algoritması kullanılarak maksimum akış hesaplanır ve robot sayısı için bir alt sınır elde edilmiştir. Daha sonra, bu alt sınır bir tamsayılı programlama modeli ile optimize edilir. Bu yöntem, daha hızlı sonuçlar üretir ve dinamik ortamlara daha uygundur. Çalışma, bir finans kurumundan alınan gerçek bir veri seti ve sentetik test örnekleri üzerinde test edilmiştir. Finansal işlemler, işlem türüne göre farklı işlem süreleri, hacimler, piyasa saatleri ve son teslim tarihleri ile karakterize edilir.

RPA problemini ele alarak, finansal işlemleri yazılım robotlarına atamak için bir çözüm sunmaktadır [4]. Amaç, işlemleri en az sayıda robotla tamamlayarak maliyetleri minimize etmektir. Çalışma, dört farklı sezgisel algoritma ve bir tamsayılı doğrusal programlama modeli kullanarak problemi çözmeyi hedeflemiştir. Finansal işlemler, işlem türüne göre farklı işlem süreleri, hacimler ve piyasa saatleri ile karakterize edilmiştir. Sezgisel algoritmalar, robot sayısı için bir üst sınır hesaplar ve tamsayılı doğrusal programlama modeline başlangıç noktası sağlayarak çözüm süresini hızlandırır.

RPA problemini çözmek için bir tamsayılı doğrusal programlama modeli öneriyor [43]. Amaç, finansal işlemleri yazılım robotlarına atayarak toplam robot sayısını ve maliyetleri minimize etmektir. Çalışma, finans sektöründe RPA'nın uygulanabilirliğini inceleyerek, robotların görev planlamasını optimize eden iki aşamalı bir çözüm sunmaktadır. Finansal işlemler, işlem türüne göre farklı işlem süreleri, hacimler, piyasa saatleri ve son teslim tarihleri ile karakterize edilir. Amaç, robot sayısını ve robotların yeniden yapılandırma/startup maliyetlerini minimize etmektir.

Laalaoui and M'Hallah tarafından yapılan çalışmada [44] L_{max} minimizasyonu yaptıkları ortaya konan F arama sezgiselini, artımlı arama (*I-incremental-search*) ve çoklu yeniden başlatmalar (*R-multiple-restarts*) ile genişleterek oluşturdukları F_{IR}

ismini verdikleri bir sezgisel yöntem önermişlerdir. Önerilen yöntem 100.000 iş içeren büyük ölçekli problemleri bir dakikadan daha kısa sürede çözebilmiştir. Çalışmada ele alınan problem $1|r_j|L_{max}$ yapısında bir çizelgeleme problemidir.

Mallem et al. Tarafından yapılan çalışmada [45], tek makinede serbest bırakma tarihleri (release dates), son teslim tarihleri (deadlines) ve öncelik ilişkileri (precedence relations) olan işlerin çizelgelenmesiyle ilgilidir. Amaç, maksimum tamamlanma süresini (makespan, C_{max}) minimize etmektir. Bu problem, literatürde $1|prec, r_j, d_j|C_{max}$ olarak kodlanır. Çözüm için kesin çözüm yaklaşımlarından dinamik programlama önerilmiştir. Önerilen yöntem $1|prec, r_j|L_{max}$ problemleri için uygulanabilir ve her iki problemde de etkin sonuçlar elde ettiği gösterilmiştir.

Chu tarafından yapılan çalışmada [46], tek makine çizelgeleme problemi bağlamında, eşit olmayan serbest bırakma tarihleri (unequal release dates) ile işlerin toplam gecikmesini (total tardiness) minimize etmeyi amaçlayan bir çizelgeleme problemine ele almaktadır. $1|r_j, d_j|\sum_j T_j$. Problemi çözmek için bir Branch-and-Bound (Dallanma ve Sınırlandırma) algoritması geliştirmiştir. 230 işe kadar problemleri tam olarak çözebilen bir dal-sınır algoritması geliştirilmiştir.

Baptiste et al. yaptığı çalışmayla [47] $1|r_j, d_j|\sum_j T_j$ problemine dal-sınır yöntemi geliştirmiştir. Chu (1992) tarafından geliştirilen dal-sınır yöntemi ile karşılaştırıldığında daha güçlü alt sınırlar, gelişmiş dominance kuralları ve akıllı geri izleme gibi yeniliklerin ilave edildiği görülmektedir. Bu, algoritmanın hem hızını hem de doğruluğunu önemli ölçüde artırmış ve neticesinde algoritmayı 500 işe kadar olan problemleri çözebilecek hale getirmiştir.

3. MATERYAL VE YÖNTEM

3.1 Robotik Süreç Otomasyonu

Bu başlık altında, RPA konusuna dair genel bir açıklama ve temel bilgiler sunulacaktır.

3.1.1 Robotik süreç otomasyonu: tanım ve kapsam

RPA, günümüzde birçok işletme tarafından iş süreçlerini otomatikleştirmek amacıyla tercih edilen ve giderek yaygınlaşan bir teknoloji olarak karşımıza çıkmaktadır. RPA, dijital sistemler arasında etkileşimde bulunan, insan eylemlerini taklit eden robot yazılımlarını ifade etmektedir. Bu yazılımlar, yapay zekâ desteği kullanmakla birlikte, yapay zekâ ile karıştırılmamalıdır. Özellikle insanların bilgisayar üzerinde gerçekleştirdiği rutin, tekrarlayan ve belirli kurallar çerçevesinde şekillenen işlemlerin otomatikleştirilmesinde kullanılmaktadır. Bu teknoloji sayesinde çalışanlar, daha planlı ve stratejik görevlere odaklanabilmekte, bu sırada yazılım robotları (botlar veya ajanlar) ise manuel işlemleri hızlı, kesintisiz ve hatasız bir şekilde yerine getirmektedir. RPA'nın doğru bir şekilde uygulanması, işletmelere operasyonel maliyetlerin düşürülmesi, süreç hatalarının minimize edilmesi gibi avantajlar sağlayarak, organizasyonların daha stratejik hedeflere odaklanmalarına olanak tanımaktadır [6].

RPA teknolojisi, birden fazla temel bileşen ve teknolojiden oluşmakta olup, aynı zamanda çeşitli teknolojileri desteklemektedir. Bu özellik, RPA'nın gelişimi için uygun bir ortam sağlamaktadır. Makine öğrenmesi, yapay zekâ, optik karakter tanıma (OCR), görüntü işleme, API entegrasyonları ve bulut teknolojileri gibi birçok teknoloji, RPA'nın işleyişini destekler ve gelişimine katkı sağlar. Bu etkileşim, RPA'nın geniş bir yelpazede farklı iş süreçlerinin otomatikleştirilmesine olanak tanır [1].

Bu çerçevede, RPA aşağıda sıralanan her bir iş sürecini gerçekleştirebilme yeteneğine sahiptir ve sürekli gelişime açıktır.

- Dosyaların indirilmesi, kopyalanması, taşınması, oluşturulması ya da karşılaştırılması işlemlerinin gerçekleştirilebilmesi
- Web sitelerinde gezinebilme, veri girişinin yapılabilmesi ve mevcut verilerin okunabilmesi
- Uygulamalarla entegrasyon sağlanarak veri girişinin yapılması veya verilerin okunabilmesi
- PDF, Word, Excel gibi farklı dosya formatları üzerinde işlem gerçekleştirebilme
- E-postaların okunabilmesi, gönderilebilmesi ve e-posta içeriklerinden bilgi elde edilebilmesi
- Veri tabanlarına bağlantı kurulabilmesi
- Uzak masaüstü (Remote Desktop) bağlantılarının gerçekleştirilebilmesi
- FTP sunucularına erişim sağlanabilmesi
- Web servislerinden yararlanılabilmesi

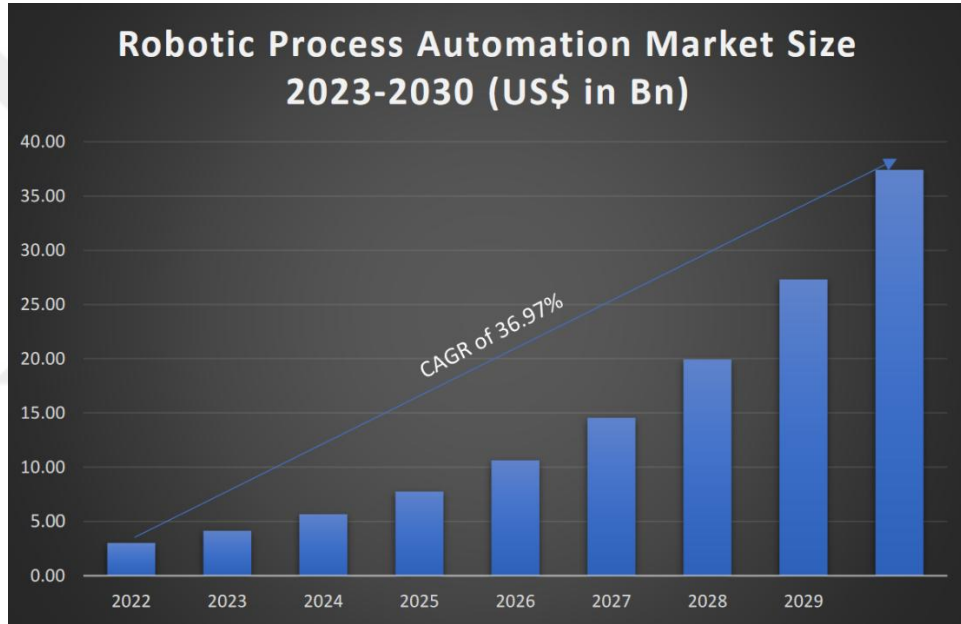
3.1.2 Robotik süreç otomasyonunun tarihsel gelişimi

RPA, dünya genelinde ve ülkemizde hâlâ birçok kişi tarafından yeterince tanınmamakla birlikte, temelleri 2000'li yılların başlarında atılmaya başlanmıştır. Ancak, RPA'nın gelişimini daha geniş bir çerçevede değerlendirdiğimizde, bu sürecin aslında makine ve otomasyon teknolojilerinin ortaya çıkışıyla başladığı söylenebilir. Makine öğrenmesinin temelleri, Arthur Samuel tarafından 1959 yılında kaleme alınan ve “*Some Studies in Machine Learning Using the Game of Checkers*” başlıklı makalede atılmıştır [48]. Makine öğrenmesinin keşfi, Doğal Dil İşleme (NLP) ve yapay zekâ teknolojilerinin gücünden yararlanarak, bilgisayarların metin ve ses verilerini anlamasını ve insan benzeri tepkiler vermesini mümkün kılmıştır. Bu gelişmeler, RPA teknolojisinin ileriye dönük evrimini şekillendirmiş ve potansiyelinin önünü açmıştır. RPA'nın gelişiminde, 1990'lı yıllar önemli bir dönüm noktası teşkil etmektedir. Bu dönemde, ekran kazıma (screen scraping) teknolojisi, RPA'nın öncüsü olarak ortaya çıkmıştır. Ekran kazıma, bir uygulamanın kullanıcı arayüzünden veri okuma ve yazma işlemleri gerçekleştirerek, eski yazılımlar ile etkileşim sağlamaya olanak tanıyan bir teknolojidir [1].

RPA teknolojisinin günümüzdeki biçimi, temelde 2000'li yıllarda şekillenmeye başlamıştır. İlk etapta, iş süreçlerinin otomatikleştirilmesi amacıyla makro ve script tabanlı çözümler geliştirilmiş, ancak bu çözümler sınırlı esneklik ve ölçeklenebilirlik

sunmuştur. Bu sınırlamaların aşılması, kullanıcı arayüzlerini taklit eden ve mevcut sistemlerle entegre olabilen yazılımsal robotların geliştirilmesiyle mümkün olmuştur. Sonuç olarak, RPA, finans, sigorta, sağlık ve telekomünikasyon gibi çeşitli sektörlerde uygulanabilir hale gelmiştir. Bu çok yönlü kullanım potansiyeli, RPA'nın benimsenme oranını arttırmış ve pazar büyüklüğünün hızla genişlemesine olanak tanımıştır.

Bu öngörü, 2023 yılında yayımlanan “*Robotic Process Automation Market Trend and Size*” başlıklı makale tarafından desteklenmektedir[49]. Bu çalışmaya göre, RPA pazarının büyüklüğü 2022 yılı itibarıyla 3,02 milyar ABD doları olarak belirlenmiştir. Sektördeki büyüme trendine paralel olarak, 2030 yılına gelindiğinde pazar büyüklüğünün 37,49 milyar ABD dolarına ulaşması öngörülmektedir.



Şekil 3.1: RPA pazar büyüklüğü [49].

3.1.3 Robotik süreç otomasyonu teknolojisinin işletmelerdeki yeri ve önemi

Günümüzde işletmeler, iş süreçlerini daha verimli bir şekilde yönetmek amacıyla çeşitli teknolojilerden yararlanmaktadır. Özellikle, popülerliğinin artmasıyla birlikte yapay zekâ uygulamaları ve iş süreçlerini kolaylaştırmaya yönelik birçok teknolojiyi iş süreçlerine entegre etme çabaları ön plana çıkmaktadır. Bu bağlamda RPA, hızla yaygınlaşan ve birçok işletme tarafından benimsenen önemli bir teknoloji haline gelmektedir. RPA, esnek yapısı ve kolay entegrasyon yeteneği sayesinde geniş uygulama alanları (örneğin, sağlık, eğitim, telekomünikasyon) ve her sektörde

verimlilik artışı sağlama potansiyeli ile teknolojinin gelecekte daha yaygın bir şekilde benimsenmesine olanak tanımaktadır [22].

RPA, iş süreçlerini iyileştirme, bilgi sızıntısı riskini azaltma, iç kontrol ve bilgi güvenliği yönetimini geliştirme potansiyeline sahiptir [50]. Ayrıca, maliyet tasarrufu, doğruluk, ölçeklenebilirlik ve uyumluluk gibi önemli kazanımlarıyla birçok sektör ve departmanda vazgeçilmez bir teknoloji haline gelmiştir [51]. RPA'nın başarılı bir şekilde uygulanabilmesi için doğru stratejilerin izlenmesi ve kurumsal hazırlık düzeyinin sağlanması gerektiği göz önünde bulundurulduğunda, bu teknoloji işletmeler tarafından sayısız kazanım sunduğu için tercih edilen bir çözüm haline gelmiştir [52]. RPA, organizasyonlar için maliyet tasarrufu, süreç kalitesinde iyileştirmeler ve çalışan memnuniyetinde artış sağlamaktadır. Bununla birlikte, bu faydalara ek olarak, RPA uygulamalarında çeşitli zorluklar da mevcut olabilir; örneğin, IT altyapısının kurulumu ve süreçlerin standardizasyonu gibi engeller söz konusu olabilir. Bu zorluklar göz önünde bulundurulduğunda, RPA'nın kullanımının, rutin işlerin otomatikleştirilmesi sayesinde çalışanların daha yaratıcı ve stratejik görevlere odaklanmalarını sağladığı gözlemlenmiştir [53]. Bu bağlamda, RPA'nın işletmelerdeki etkisi, iş gücü verimliliği, maliyet tasarrufu ve süreç iyileştirme gibi unsurlar üzerinden değerlendirilmektedir. RPA'nın işletmelerdeki etkisini ortaya koyan istatistiksel veriler, bu teknolojinin benimsenmesinin yaygınlığını ve sağladığı faydaları gözler önüne sermektedir. Örneğin, RPA uygulamalarının bankacılık sektöründe çalışan verimliliğini %30 oranında artırdığı ve maliyetleri %25 oranında düşürdüğü rapor edilmiştir [54].

RPA'nın kullanımıyla birlikte rutin işlerin otomatikleştirilmesi sayesinde çalışanların daha yaratıcı ve stratejik görevlere yönlendirildiği gözlemlenmiştir [13,14]. Bu veriler, RPA'nın işletmeler için sağladığı somut faydaları ortaya koymaktadır.

Literatürde RPA'nın güncel yaklaşımları arasında, yapay zekâ ve büyük veri analitiği ile entegrasyonu önemli bir yer tutmaktadır. RPA'nın, büyük veri analitiği ile birleştiğinde, işletmelere daha derinlemesine veri analizi yapma ve stratejik kararlar alma imkânı sunduğu belirtilmektedir [53]. Ayrıca, RPA'nın iç denetim süreçlerinde kullanımı da artmakta ve bu alanda yapılan çalışmalar, RPA'nın denetim süreçlerini nasıl dönüştürdüğünü ortaya koymaktadır [55]. RPA'nın, işletmelerin dijital dönüşüm süreçlerinde kritik bir rol oynadığı ve bu süreçlerin hızlanmasına katkı sağladığı da vurgulanmaktadır[13,14].

Sonuç olarak, RPA, işletmelerin verimliliklerini artırmak, maliyetlerini düşürmek ve süreçlerini optimize etmek için önemli bir araç olarak öne çıkmaktadır. Teorik çerçeve ve istatistiksel verilerle desteklenen RPA'nın, günümüzdeki önemi ve etkisi, işletmelerin dijital dönüşüm süreçlerinde kritik bir rol üstlenmektedir.

3.1.4 Robotik süreç otomasyonu uygulamaları için teknik gereksinimler

RPA, süreç otomasyonu, yapay zekâ ve makine öğrenimi gibi kavramlarla sıkı bir şekilde bağlantılıdır. RPA, tekrarlayan ve kural tabanlı iş süreçlerini otomatikleştiren yazılım robotları kullanarak verimlilik sağlamaktadır. Bu bağlamda, RPA, iş süreçlerinin yeniden tasarımı ve otomasyonu üzerine inşa edilmiştir. RPA'nın etkinliği, süreçlerin standartlaştırılması ve otomatikleştirilmesi ile doğrudan ilişkilidir. Bu süreçte, iş akışlarının analizi ve optimizasyonu önemli bir rol oynamaktadır [56]. RPA uygulamaları için gerekli olan teknik gereksinimler, yazılım ve donanım altyapısını içermektedir.

RPA sistemleri genellikle aşağıdaki bileşenleri içermektedir.

- *Altyapı Gereksinimleri*

Sunucu ve depolama: RPA süreçlerinin etkin bir şekilde çalışabilmesi için güvenilir ve ölçeklenebilir sunucu altyapısına ihtiyaç duyulmaktadır. Ajanların çalışacağı ortamların yüksek performanslı olması, işlem sürelerini ve uygulamadan elde edilen verimi doğrudan artırmaktadır.

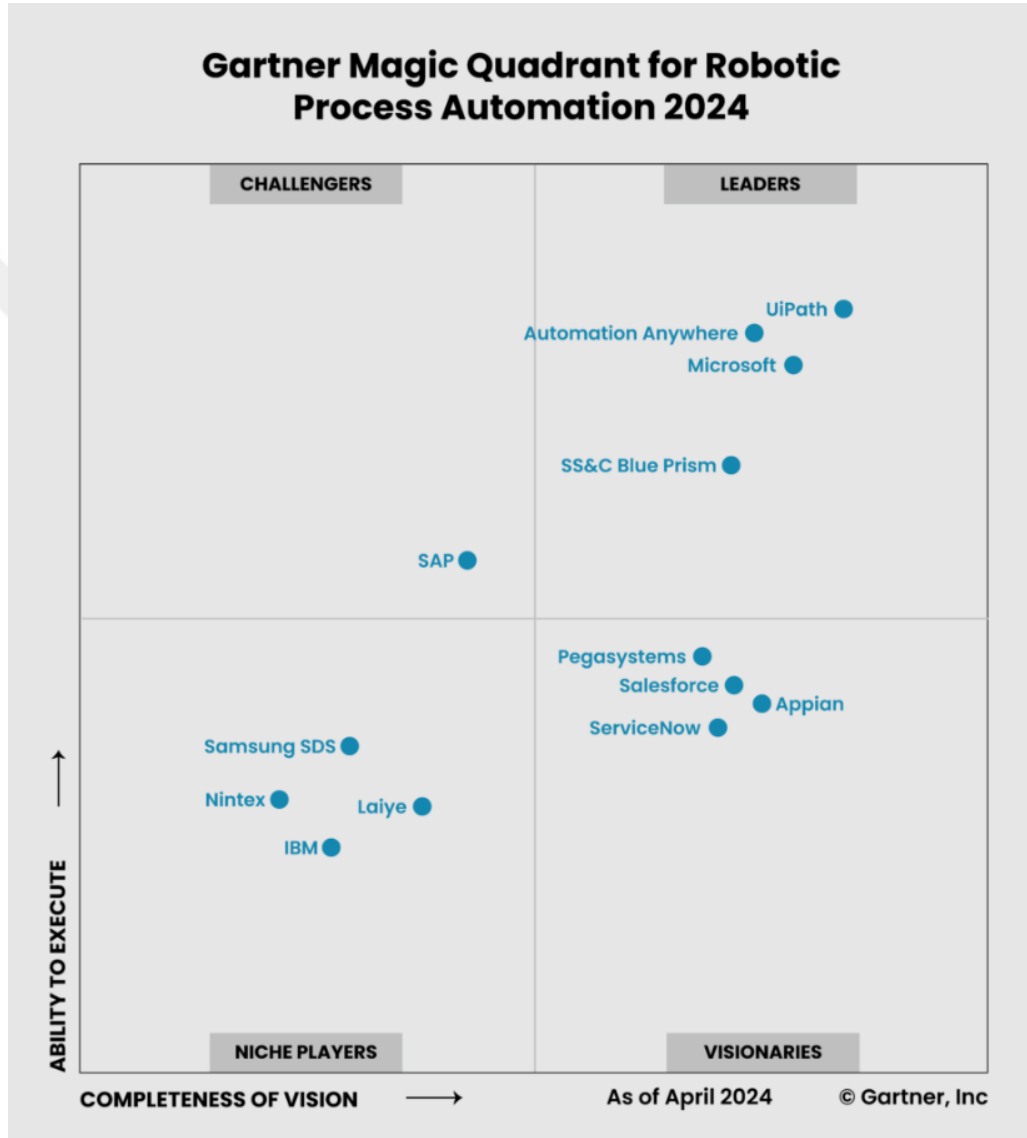
Veri tabanı: Süreçlerin etkin bir şekilde izlenmesi ve yönetilebilmesi için güvenli ve yüksek performanslı bir veri tabanı altyapısına ihtiyaç duyulmaktadır.

Network bağlantısı: Güçlü bir ağ bağlantısına sahip robot sunucuları, sistemlerle etkili bir şekilde iletişim kurabilmek için kritik bir öneme sahiptir. Yavaş ağ bağlantıları, robotların işlevselliğini olumsuz yönde etkileyebilir. RPA botlarının kesintisiz bir şekilde çalışabilmesi için güvenilir ve düşük gecikmeli bir ağ bağlantısının sağlanması gerekmektedir.

Yedekleme ve kurtarma araçları: Olası sistem arızalarına karşı kurtarma senaryoları oluşturulmalıdır. Ayrıca, bot logları ve işlenen veriler düzenli olarak yedeklenmelidir.

- *Yazılım Gereksinimleri*

Yazılım platformu: RPA uygulamaları, etkin bir şekilde çalışabilmek için bir platforma ihtiyaç duyar. Platformlar, lisanslama, güvenlik, güncellemeler ve entegrasyon yetenekleri açısından birbirlerinden farklılık gösterse de temel olarak aynı amaca hizmet etmektedir. RPA süreçleri için farklı platformlar tercih edilebilir. Örneğin, UiPath, Automation Anywhere ve Blue Prism gibi çeşitli platformlar bu alanda yaygın olarak kullanılmaktadır.



Şekil 3.2: Robotik süreç otomasyonu için şekil [80].

Uygulama entegrasyonları: RPA ajanlarıyla (botları) çalışacak uygulamaların desteklenmesi gerekmektedir. (ERP, CRM, web tarayıcıları gibi)

- *Güvenlik Gereksinimleri*

RPA süreçlerinde güvenlik, bazı güvenlik protokolleri (KVKK vb.) büyük önem taşır. Kullanıcıların verileri, şirket bilgileri, faturalar, ücretler gibi birçok bilgiye ulaşım sağlanabildiği için kullanıcı yetkileri denetlenmeli ve yetkisiz erişimler engellenmelidir. Erişimler için genellikle rollere dayalı erişim ve tek oturum açma (single sign on) kullanılır.

- *İş Süreçleri ve İş Akışı Gereksinimleri*

Süreçlerin belirlenmesi: Otomasyona uygun süreçlerin tespiti durumunda, bu süreçlerin belgelenmesi büyük bir önem taşır. Süreçler, geliştirme aşamasında ve öncesinde ayrıntılı bir şekilde belgelenmelidir.

İzlenebilirlik: RPA botları, çalışma sürecinde çeşitli hatalara açık olabilir. Sistem veya iş kuralı hataları gibi sorunlar, hata yönetimi (exception handling) mekanizmaları aracılığıyla otomasyon sırasında beklenmedik durumlarla karşılaşıldığında ele alınır. Hem bu tür durumlar hem de başarıyla tamamlanan senaryolarda, başarı oranları ve hata oranları gibi metriklerin izlenebileceği ve yönetilebileceği uygulamaların bulunması gerekmektedir.

- *Eğitim ve Kullanıcı Gereksinimleri*

RPA projelerinin uygulanabilirliğini artırmak için gerek son kullanıcılar gerekse proje ekipleri, eğitilmiş ve RPA farkındalığı yüksek bireylerden oluşmalıdır. Bu nedenle, ilgili kişilere yönelik eğitimlerin sağlanması elzemdir. Ayrıca, bu durum, genellikle düşünüldüğünden daha kritik bir öneme sahiptir. Son kullanıcıların RPA süreçlerine hâkim olmaları ve RPA'nın sınırlamalarını bilmeleri hem beklentilerin karşılanması hem de RPA süreçlerinin hatasız bir şekilde hayata geçirilmesi açısından büyük bir öneme sahiptir.

Sonuç olarak, RPA projelerinin başarıya ulaşabilmesi, teknik gereksinimlerin eksiksiz bir şekilde belirlenmesi ve uygulanmasına bağlıdır. Altyapı, yazılımlar, güvenlik, iş süreçlerindeki gereksinimler ve eğitim gibi faktörler, RPA ile ele alınmalı ve sistemlerin sürdürülebilirliği için her biri eksiksiz bir şekilde sağlanmalıdır.

3.1.5 Robotik süreç otomasyonu adımları ve uygulama aşamaları

RPA uygulamaları, manuel olarak yürütülen süreçlerin video kaydı ile belgelenmesini ve bu kayıtların test senaryoları olarak kullanılmasını içerir. Sonrasında, bu senaryolar üzerinden otomasyon süreçleri geliştirilir. RPA süreçlerinin geliştirilmesinde,

geleneksel yazılım geliştirme yöntemleri yerine çevik yaklaşımlar tercih edilmelidir. Geleneksel "şelale" yönteminin aksine, bu iteratif ve çevik yaklaşım, zaman ve maliyet tasarrufu sağlamakla birlikte hızlı prototiplemeye olanak tanır [57]. Genel olarak, RPA'nın başarılı bir şekilde uygulanabilmesi için beş temel adım bulunmaktadır. Bu adımlar, çalışılan firma, uygulama veya sürece bağlı olarak bazı farklılıklar gösterebilir. Süreç madenciliği ve bazı yapay zekâ uygulamaları entegrasyonu söz konusu olabilse de genel anlamda aşağıdaki adımlar izlenmektedir.

- *Süreç seçimi ve analizi:*

Otomasyona uygun süreçlerin seçilmesi, RPA'nın doğru bir şekilde uygulanabilmesi için büyük bir önem taşımaktadır. İyi bir analiz ve doğru bir şekilde belirlenmiş süreç, RPA süreçlerinin temel unsurlarıdır. Bu nedenle, seçilecek süreçlerin tekrarlanabilir, entegrasyonu kolay ve yüksek verimlilik potansiyeline sahip iş süreçleri olması gerekmektedir. Hackathonlar ve son kullanıcı tarafından gelen talepler, bu seçimde önemli bir rol oynamaktadır. Son yıllarda, RPA ve süreç madenciliği entegrasyonunun artan bir şekilde kullanılması dikkat çekmektedir. Süreç madenciliği, mevcut iş süreçlerinin analiz edilmesi ve otomasyona en uygun adayların belirlenmesi amacıyla kullanılan bir teknolojidir [58]. Konuyu özetlersek doğru seçilmiş bir süreçten sonra alınacak iyi bir analiz ile belgelendirme ve gelişim süreci başlar.

- *Süreç tasarımı:*

RPA süreçleri, yapılan analizlerin ardından teknik ve süreç dokümantasyonlarının hazırlanmasıyla başlar. Bu dokümanlar, gereksinimlerin, iş adımlarının, yetkilerin, testlerin, hata senaryolarının ve teknik tasarımların ayrıntılı bir şekilde açıklanmasını sağlar. Söz konusu adımın sonunda, üretilen dokümanların son kullanıcı tarafından onaylanması beklenir.

- *Robot geliştirme ve test:*

Onaylanan ve tasarımı tamamlanmış süreçler, geliştirmeye hazır hale gelir. Bu süreçler, bir framework kullanılarak geliştirilme aşamasına başlanır. Geliştirme süreci boyunca, çeşitli uygulama ve yazılımlar kullanılır. Süreçler tamamlandıktan sonra, testler gerçekleştirilir ve son kullanıcının onayına sunulur. Geliştirme işlemleri genellikle test sunucularında yapılmaktadır.

- *Dağıtım ve izleme:*

Robotlar, entegrasyonu kolay yazılımlar olsalar da test sunucularındaki süreçler ile canlı ortamlar arasında bazı farklar bulunabilir. Canlı ortama alınan robotların performansları, hypercare (bakım) süresi boyunca dikkatle izlenir ve gerektiğinde iyileştirmeler yapılır.

- *Sürekli iyileştirme:*

RPA süreçleri geliştirildikten ve çalışmaya başladıktan sonra bakım gerektiren süreçler haline gelir. Ayrıca, son kullanıcılar tarafından süreçlerde değişiklik talepleri de gelebilir. Bu nedenlerle, otomasyon süreçleri düzenli olarak gözden geçirilir ve sürekli olarak iyileştirilir.

Sonuç olarak, RPA, işletmelerin dijital dönüşüm süreçlerinde kritik bir rol oynamaktadır. Doğru bir şekilde uygulandığında, operasyonel verimliliği artırırken maliyetleri düşürür ve rekabet avantajı sağlar. Burada kritik olan unsur, doğru bir uygulama sürecinin izlenmesidir. Sürekli iyileştirme gerektiren bu süreçler, doğru bir şekilde analiz edilmeli, iyi eğitim almış ekipler tarafından tasarlanmalı ve sorunsuz ortamlarda çalıştırılmalıdır.

3.1.6 Süreçlerin seçim kriterleri ve uygunluk analizi

RPA süreçlerinin doğru planlanmadığı durumlarda sistemin verimliliğinin düştüğünü ve maliyetlerin arttığını savunmaktadır. Temel düşünce, RPA süreçlerinin doğru zamanlama algoritmalarıyla optimize edilmesinin hem iş yükü yönetiminde hem de süreçlerin tamamlanma hızında kayda değer iyileştirmeler sağlayabileceğidir [59].

RPA projeleri için uygun süreçlerin seçilmesi konusunda çeşitli yaklaşımlar bulunmaktadır. Ancak temel olarak, RPA projelerinde süreç seçiminde dikkate alınması gereken altı ana kriter bulunmaktadır: süreçlerin yürütme sıklığı, yürütme süresi, standardizasyon düzeyi, stabilite, hata oranı ve otomasyon potansiyeli. Bu göstergeler, sürecin verimli bir şekilde otomatikleştirilmesi için belirleyici faktörlerdir [60].

- *Süreçlerin Yürütme Sıklığı:*

RPA, rutin ve tekrarlayan işleri hızlı ve hatasız bir şekilde otomatikleştirir. Bu nedenle, otomasyona uygun süreçlerin yüksek bir sıklıkta gerçekleşmesi gerekmektedir. Örneğin, günlük, haftalık veya belirli periyotlarla sürekli olarak tekrar eden işlemler, RPA için ideal adaylardır. Eğer bir süreç nadiren tekrarlanıyorsa, otomasyonun

sağladığı fayda daha düşük olabilir, çünkü her tekrarında RPA sisteminin yeniden başlatılması ve yapılandırılması gerekebilir. Ancak bu durum kesin bir kısıtlama getirmez. Örneğin, yıllık periyotlarla gerçekleşen bir süreç, düzenli bakım gerektirmeyen ve RPA'ya uygun bir işlemse, otomasyonun faydası yine değerlendirilebilir.

- *Süreçlerin Yürütme Süresi:*

Sürecin tamamlanma süresi, RPA projelerinde önemli bir değerlendirme kriteri olarak öne çıkmaktadır. Kısa sürelerde tamamlanan ve zaman alıcı olmayan işlemler, genellikle RPA ile etkin bir şekilde otomatikleştirilebilmektedir. Buna karşın, uzun süre gerektiren ve karmaşık yapıya sahip süreçlerde, otomasyonun sağladığı verimlilik artışı daha sınırlı olabilir. Bu bağlamda, otomasyon uygulamadan önce süreçlerin yürütme sürelerinin titizlikle analiz edilmesi büyük bir önem taşımaktadır. Özellikle, kısa sürelerde tamamlanan fakat yüksek frekansta tekrarlanan işlemler, RPA'nın en verimli şekilde kullanılabildiği alanlar olarak dikkat çekmektedir.

- *Standardizasyon derecesi:*

RPA'nın etkin bir şekilde çalışabilmesi için, süreçlerin mümkün olduğunca standart hale getirilmesi büyük önem taşır. Süreçlerde sürekli değişiklikler yapılması veya aşırı esneklik bulunması, RPA'nın başarılı olma olasılığını azaltmaktadır. Zira robotlar, belirli kurallar ve prosedürler doğrultusunda işlemektedir. Dolayısıyla, bir sürecin değişkenlik göstermesi, robotların hata yapmasına veya yanlış adımlar atmasına neden olabilir. Bu nedenle, süreçlerin yüksek bir standardizasyon düzeyine sahip olması, kuralların net bir şekilde belirlenmesi ve süreç tasarımının istikrarlı olması gerekmektedir.

- *Stabilite:*

Sürecin stabilitesi, otomasyonun başarısı açısından kritik bir başka belirleyici faktördür. Eğer bir süreç sürekli olarak değişim gösteriyorsa, örneğin yazılım güncellemeleri, organizasyonel değişiklikler veya sistem yapılandırmalarındaki sık değişiklikler nedeniyle, bu tür bir sürecin otomatikleştirilmesi uzun vadede zorlu hale gelebilir. RPA'nın etkinliği için, sürecin stabil olması ve sıklıkla değişmemesi gerekmektedir. Stabil bir süreç, robotların düzgün çalışmasını sağlamakla birlikte, bakım ihtiyacını da minimize eder.

- *Hata Oranı:*

Sürecin hata oranı, otomasyonun başarısı açısından son derece önemli bir faktördür. Hata oranı yüksek olan süreçler, manuel olarak yürütüldüğünde zaman kaybına neden olabilir ve hatalı sonuçlar doğurabilir. Ancak, bu tür süreçler RPA ile otomatikleştirildiğinde, hata oranı genellikle azalır, çünkü robotlar sabırlıdır ve insan hatalarına eğilimli değildir. Hatalar çoğunlukla süreçlerin yanlış tasarlanmasından kaynaklanmaktadır; dolayısıyla, sürecin yüksek hata oranı, otomasyon açısından da bir risk teşkil edebilir. Hata oranı düşük ve adımları net bir şekilde belirlenmiş süreçler, RPA için en uygun adaylar olarak öne çıkmaktadır. Örneğin, sık değişiklikler içeren dosyalar, sürekli arayüz değişiklikleri yaşanan web siteleri veya okunması güç belgeler, hata riski taşıyan süreçlerdir. Bu tür süreçlerin otomasyon için değerlendirilmesi, başarılı bir RPA uygulaması için kritik öneme sahiptir.

- *Otomasyon Oranı:*

Sürecin daha önce ne kadar otomatikleştirildiği, RPA uygulamalarının etkinliği açısından önemli bir başka faktördür. Bazı süreçlerde çok sayıda manuel adım bulunabilir ve bu adımlar zaman içinde daha fazla otomasyon gerektirebilir. Eğer bir süreç zaten yüksek oranda otomatikleştirilmişse, ek otomasyonun sağladığı fayda sınırlı olabilir. Ancak tamamen manuel olarak yürütülen ve her adımda insan müdahalesine ihtiyaç duyan süreçlerde, RPA uygulaması büyük bir verimlilik artışı sağlayabilir. Yüksek manuel işlem gereksinimi taşıyan süreçler, RPA'nın etkin bir şekilde devreye girebileceği ve önemli verimlilik kazançları elde edilebilecek süreçlerdir.

Temelde, bu altı gösterge, RPA projelerinin başarısını önemli ölçüde etkileyen temel faktörlerdir. Süreçlerin doğru bir şekilde seçilmesi, yalnızca yatırımın geri dönüş süresini kısaltmakla kalmaz, aynı zamanda organizasyonel verimliliği de artırır. RPA'nın uygulanması, verimlilik, doğruluk ve hızda belirgin iyileşmeler sağlayabilir. Bu nedenle, doğru süreçlerin seçilmesi, RPA projelerinin başarılı bir şekilde yürütülmesindeki en önemli ilk adımdır. Ayrıca, RPA projelerinde süreç seçimi yapılırken, sürecin iş değeri, paydaşların beklentileri, sürecin etkileşimde bulunduğu diğer sistemler ve süreçler gibi faktörlerin de dikkate alınması faydalı olacaktır.

Bu faktörlerin doğru bir şekilde analiz edilmesi, başarılı bir otomasyon stratejisinin oluşturulmasında kritik bir rol oynamaktadır. "Neyin otomatikleştirilmesi gerektiği ve

neyin insan kontrolünde kalması gerektiği" sorusu, sürekli olarak değerlendirilmesi gereken önemli bir unsurdur. RPA, geleneksel "inside-out" yaklaşımı yerine "outside-in" yaklaşımını benimseyerek, mevcut bilgi sistemlerinin değiştirilmeden süreçlerin otomatikleştirilmesini mümkün kılmaktadır. RPA, bilgi sistemlerine dışarıdan müdahale ederek süreçlerin otomatikleştirilmesini sağlar; bu da klasik "inside-out" yaklaşımından farklı olarak, mevcut sistemlerin değiştirilmesini gereksiz kılar [61]. Bu yaklaşım, esneklik ve hızlı entegrasyon sağlarken, aynı zamanda şirketlerin mevcut altyapısına zarar verme riski olmadan yenilikçi çözümler geliştirme olanağı sunmaktadır. Ayrıca, organizasyonların eski sistemleriyle uyumlu bir şekilde çalışarak süreçlerde hız ve doğruluk artışı sağlanabilir. Ancak, doğru analiz yapılmadan gerçekleştirilen dışarıdan müdahale ile yapılan otomasyonun bazı sınırlamaları söz konusu olabilir. Süreçlerin karmaşıklığının ve sistemler arası etkileşimlerin doğru bir şekilde anlaşılması, otomasyonun başarısı için hayati öneme sahiptir.

3.1.7 Robotik süreç otomasyonu uygulamalarında karşılaşılan zorluklar

RPA uygulamalarının entegrasyon süreçlerinin karmaşıklığı, uygulama maliyetlerinin yüksek olmasına neden olabilir. RPA'nın başarılı bir şekilde uygulanabilmesi için, süreçlerin titizlikle analiz edilmesi ve uygun otomasyon araçlarının seçilmesi büyük önem taşır. Ancak, karşılaşılan zorluklar, bu potansiyelin sınırlanmasına yol açabilir [62]. RPA'nın uygulanması sırasında karşılaşılan en büyük zorluk, hangi görevlerin otomasyona uygun olduğunu doğru bir şekilde belirlemektir [63]. RPA uygulamalarının karşılaştığı zorluklar arasında, süreçlerin standardizasyonu, IT altyapısının kurulumu ve çalışanların iş kaybı korkusu önemli yer tutmaktadır. Örneğin, standartlaştırılmamış süreçlerin varlığı, RPA uygulamalarında engelleyici bir faktör olarak karşımıza çıkmaktadır. Bunun yanı sıra, çalışanların RPA nedeniyle iş kaybı endişesi, bu tür uygulamaların benimsenmesini zorlaştıran bir diğer önemli etkidir [64]. Ayrıca, RPA projelerinin başarısızlık oranı yüksek olabilmektedir ve bu başarısızlıkların temel nedeni, süreç odaklı bir stratejinin eksikliği olarak değerlendirilebilir. Değer odaklı RPA uygulamaları, yalnızca iş süreçlerinin verimliliğini artırmakla kalmaz, aynı zamanda riskleri de en aza indirmeye yardımcı olabilir [65].

RPA'nın entegrasyon süreçlerinin karmaşıklığı ve uygulama maliyetlerinin yüksekliği, özellikle başlangıç aşamalarında önemli engeller oluşturabilir. Ancak, doğru analizler

yapılır ve uygun otomasyon araçları seçilirse, bu zorluklar aşılabılır. RPA'nın başarılı bir şekilde uygulanabilmesi için, süreçlerin titizlikle analiz edilmesi, görevlerin doğru bir şekilde sınıflandırılması ve çalışanların bu değişikliklere uyum sağlaması gereklidir. Çalışanların iş kaybı korkusunun etkin bir şekilde yönetilmesi ve sürecin doğru bir şekilde standardize edilmesi, RPA uygulamalarının benimsenmesini artırarak sürecin verimli bir şekilde işlemesine katkı sağlar. Ayrıca, değer odaklı bir strateji benimsemek, RPA projelerinin başarısızlık oranlarını azaltmaya ve potansiyelini en üst düzeye çıkarmaya yardımcı olabilir.

Sonuç olarak, RPA teknolojisi, doğru planlama ve uygun stratejilerle önemli verimlilik artışları sağlayabilir. Süreçlerin standart hale getirilmesi, iş gücü endişelerinin giderilmesi ve doğru robot sayısının belirlenmesi, RPA uygulamalarının başarısını doğrudan etkileyen faktörlerdir. Bu bağlamda, "RPA için Gerekli Minimum Robot Sayısının Belirlenmesi" problemi, özellikle büyük organizasyonlar için kritik öneme sahiptir. Etkili bir çözüm yöntemi geliştirilmesi hem operasyonel verimliliği artıracak hem de kaynakların daha etkin kullanılmasını sağlayacaktır. Bu tür iyileştirmeler, RPA'nın potansiyelinin tam anlamıyla ortaya çıkmasına olanak tanır.

3.1.8 Robotik süreç otomasyonu ve yapay zekâ etkileşimi

Yapay Zekâ teknolojilerinin hızla gelişmesi ve çeşitli sistemlere entegrasyonu, birçok iş kolu ve teknolojiyi etkilemektedir. Kendisini yapay zekâ araçlarıyla entegre edemeyen kuruluşların, teknolojik yeniliklerden ve verimlilik artışlarından faydalanma konusunda geride kalacağı açıktır. Bu bağlamda, RPA, birçok yapay zekâ aracını içeren ve gelecekte bu araçlarla daha da iç içe geçmesi beklenen bir teknoloji olarak öne çıkmaktadır. RPA'nın, gelecekte yapay zekâ (AI) ile birleşerek Akıllı Süreç Otomasyonu (IPA) haline gelmesi ve bu gelişimin otomasyonun kapsamını daha da genişletmesi öngörülmektedir. Yapay zekâ ile entegre edilen RPA, zamanla Akıllı Süreç Otomasyonu (IPA) olarak evrimleşecek ve otomasyonun sınırlarını daha da genişletecektir [66]. RPA'nın AI ile entegrasyonu, daha karmaşık ve çok aşamalı görevlerin üstesinden gelinmesine olanak sağlar ve süreç otomasyonunun kapsamını genişletir. AI algoritmalarıyla birleştirilmesinin Endüstri 4.0 süreçlerinde önemli iyileştirmeler sağladığını göstermektedir. Örneğin, AI'nin RPA süreçlerine entegrasyonu, süreçlerin doğruluğunu artırmakta ve tahmin yapabilme yeteneklerini güçlendirmektedir.

Yapay Sinir Ağları, Doğal Dil İşleme (NLP) ve Metin Madenciliği gibi AI tekniklerinin RPA süreçlerinde bilgi çıkarma ve sınıflandırma için etkili olduğu bilinmektedir [67]. Bu noktada, bazı tekniklerin açıklanması çalışmaya fayda sağlayacaktır. RPA süreçlerinde, OCR (Optik Karakter Tanıma) teknolojisinin geliştirilmesinde derin öğrenme tabanlı sinir ağları kullanılabilir. Bu sayede, taranmış belgelerden bilgi çıkarma süreci otomatikleştirilebilir. Ayrıca, Doğal Dil İşleme (NLP) teknikleri, metin tabanlı veri analizi, anahtar kelime çıkarımı ve metin sınıflandırma gibi görevlerde RPA süreçlerinde etkin bir şekilde kullanılabilir. Özellikle e-postalar, sohbet kayıtları veya sosyal medya etkileşimleri üzerinden anlamlı verilerin elde edilmesini mümkün kılabilir.

Bu noktada, sıklıkla yapılan bir hata olan RPA'yı sadece bir yapay zekâ aracı olarak algılama yanılışına düşmemek önemlidir. RPA ve yapay zekâ, birlikte kullanılabilir ve kolayca entegre edilebilir. Hatta, yazıda da belirtildiği gibi, gelecekte RPA, yapay zekâ ile birleşerek tahmine dayalı işlemler gerçekleştiren bir Akıllı Süreç Otomasyonu (IPA) haline gelebilir. Ancak, en azından mevcut aşamada, RPA ve yapay zekâ arasındaki farkları Çizelge 3.1'deki gibi ele almak faydalı olacaktır.

Çizelge 3.1: RPA ve yapay zekâ arasındaki farklar.

Özellik	RPA	Yapay Zekâ
Temel Amaç	Tekrarlayan, kural tabanlı iş süreçlerini otomatikleştirir.	Karmaşık, öğrenmeye dayalı ve esnek süreçlerde karar alma ve problem çözme yeteneği gösterebilir.
Veri Türü	Yapılandırılmış verilerle çalışabilir. (Örneğin, tablolardaki sayılar vs.)	Yapılandırılmamış ve yapılandırılmış verilerle çalışabilir. (metin, ses, görüntü)
Öğrenme Yeteneği	Öğrenme yeteneği yoktur, önceden tanımlanmış kurallara dayalı çalışır.	Makine öğrenimi ve derin öğrenme ile öğrenme yeteneği vardır.
Esneklik	Çok esnek değildir, belirli bir işlem akışına sıkı sıkıya bağlı çalışabilir. Kurallar net olmalıdır.	Yüksek esneklik ve adaptasyon yeteneğine sahiptir. Kendi adaptasyonunu sağlayabilir.
İnsan Etkileşimi	İnsan etkileşimini taklit eder (örneğin, fare hareketleri, klavye girişi)	İnsan gibi düşünebilir.
Veri Giriş/Çıktı	Yapılandırılmış verilerle çalışır, belirli formatlar gerektirir. (Excel, ERP)	Hem yapılandırılmış hem de yapılandırılmamış verileri anlayabilir ve analiz edebilir. (metin, ses, görüntü)

Karar Alma	Sadece belirli kurallar ve algoritmalar ile sınırlıdır.	Verilere dayanarak dinamik ve tahmine dayalı kararlar alabilir, bazen insan benzeri kararlar verebilir.
Yapılandırma	Kullanıcılar tarafından manuel olarak yapılandırılır ve ayarlanır.	Otomatik öğrenme ve model eğitimi gerektirir.

Sonuç olarak RPA teknolojisi yapay zekâ ile karıştırılmamalıdır fakat entegrasyonu RPA için çok önemlidir. Ayrıca, AI destekli RPA araçları, organizasyonların operasyonel maliyetlerini %30-50 oranında azaltabilir. Bu bulgular, Endüstri 4.0'ın daha akıllı ve verimli bir hale gelmesine katkı sağlamaktadır ayrıca bu durum organizasyonların dijital dönüşüm sürecini hızlandırmaktadır [67]. Tüm bunlara ek olarak RPA, optimizasyon teknikleri ile birleştirildiğinde, optimizasyon teknikleri kullanılarak RPA süreçlerinin daha verimli hale getirilebileceği ve böylece daha az yazılım lisansı ile daha fazla işlemin tanımlanabileceği açıktır [68]. Bu konu tez çalışmasını da temelini oluşturacaktır. Buradan RPA'nın birçok farklı teknoloji ve disiplin ile iletişimde çalışabileceğini akılda tutmak gerekir.

3.1.9 Robotik süreç otomasyonunun maliyeti ve maliyet azaltma stratejileri

RPA'nın maliyet yapısını anlamak için, farklı maliyet bileşenlerinin analiz edilmesi gerekmektedir. Bu bileşenler hem başlangıç maliyetleri hem de operasyonel maliyetleri içerebilir.

- *Başlangıç Maliyetleri:*

RPA'nın uygulanması için gereken yazılım lisansları, eğitimler, altyapı yatırımları ve pilot projelerin başlatılması gibi başlangıç maliyetleri önemli bir yer tutmaktadır. Ayrıca, bu süreçlerin otomatikleştirilmesi için profesyonel hizmet sağlayıcılarının veya danışmanlık firmalarının ücretleri de dikkate alınması gereken önemli bir maliyet kalemidir. Başlangıç aşamasında bu faktörler, projeye dair ilk yatırım maliyetlerini oluşturacak ve işletmelerin bütçelerini etkileyebilecektir.

- *Operasyonel Maliyetler:*

RPA sistemleri, bakım, izleme ve güncelleme gereksinimlerini içerir. Bu maliyetler, zaman içinde yazılım robotlarının güncellenmesi, süreçlerin optimize edilmesi ve yeni iş süreçlerinin entegrasyonu gibi faktörlere bağlı olarak değişkenlik gösterebilir. Süreçlerin sürekli iyileştirilmesi ve teknolojiye uyum sağlanması gerektiği için, bu tür sürekli bakım ve güncelleme faaliyetleri ek maliyetler doğurabilir.

3.1.10 Maliyet azaltma stratejileri

RPA'nın en büyük avantajlarından biri, tekrarlayan görevlerin otomatikleştirilmesi sayesinde iş gücü maliyetlerini azaltmasıdır. Ayrıca, hata oranlarının düşürülmesi ve işlem sürelerinin kısaltılmasıyla verimlilik artar, bu da genel maliyetlerin daha verimli yönetilmesine olanak tanır [69]. Bazı raporlar, RPA teknolojilerinin kullanılmasıyla paylaşılan hizmetlerde işlem faaliyetlerinin operasyonel maliyetlerinde %30 ila %50 oranında azalma olduğunu göstermektedir [67].

RPA, işletmelerin maliyetlerini azaltmak ve operasyonel verimliliklerini artırmak adına önemli bir araçtır. Özellikle tekrarlayan ve kural tabanlı süreçlerde sağladığı faydalar, RPA'nın maliyet yönetimi üzerindeki etkisini güçlendirmektedir.

RPA'nın etkin bir şekilde uygulanabilmesi için gerekli robot sayısının belirlenmesi, genellikle önemli bir karar verme ve optimizasyon problemi olarak karşımıza çıkmaktadır. "RPA için Gerekli Minimum Robot Sayısının Belirlenmesi" sorusu, işletmelerin otomatikleştirilen süreçlerde verimliliği en üst düzeye çıkarmak adına kritik bir öneme sahiptir. Özellikle, robot sayısının doğru bir şekilde tahmin edilmesi, maliyetlerin optimize edilmesi, kaynakların verimli bir şekilde kullanılması ve süreçlerin hızlandırılması açısından hayati önem taşımaktadır.

Robot sayısının aşırı fazla olması, başlangıç maliyetlerini artırabileceği gibi; robot sayısının yetersiz olması da iş süreçlerinde tıkanıklıklara, gecikmelere ve verimlilik kayıplarına yol açabilmektedir. Bu bağlamda, gereksiz fazla robot kullanımı kaynak israfına sebep olurken, yetersiz robot sayısı işlem sürelerini uzatarak verimliliği düşürmektedir. Dolayısıyla, optimum robot sayısının belirlenmesi, maliyet analizi ve operasyonel verimlilik açısından kritik bir adım olarak karşımıza çıkmaktadır.

Bu dengeyi sağlarken, iş süreçlerinin çeşitliliği ve karmaşıklığı, iş yükündeki değişkenlik, zaman ve kaynak kısıtlamaları gibi faktörler önemli zorluklar yaratmaktadır. Bu unsurların detaylı bir şekilde analiz edilmesi, söz konusu durumu bir optimizasyon problemine dönüştürmekte ve farklı çözüm yöntemlerinin uygulanmasını gerektirmektedir [70]. Matematiksel modelleme, makine öğrenmesi, sezgisel algoritmalar ve simülasyon gibi yöntemler, bu problemin çözülmesinde etkili olabilir. Tezin ana konusu da bu bağlamdan şekillenecektir. Tezin amacı minimum robot sayısını belirleyerek robot maliyetlerini düşürmek ve süreçlerin maksimum verimle çalışmasını sağlamaktır.

Sonuç olarak, RPA'nın maliyet yapısının etkin bir biçimde yönetilmesi, işletmelere hem başlangıç maliyetlerini hem de operasyonel maliyetleri en aza indirerek verimliliği artırma imkânı sunmaktadır. Robot sayısının optimize edilmesi, maliyetlerin kontrol altında tutulması ve süreçlerin hızlandırılarak daha verimli hale getirilmesi açısından hayati bir öneme sahiptir. Doğru robot sayısının belirlenmesi, karmaşık bir optimizasyon problemi olup, bu süreç, iş yükü, kaynak kısıtlamaları ve süreçlerin karmaşıklığı gibi çeşitli faktörler dikkate alınarak farklı yöntemlerle çözümlenmelidir. Optimum robot sayısının bulunması, RPA'nın potansiyel faydalarının en üst düzeye çıkarılmasını sağlayarak, işletmelerin uzun vadede sürdürülebilir büyüme ve verimlilik elde etmelerine yardımcı olacaktır.

3.2 Kombinatorial Optimizasyon: Genel Bakış

Kombinatorial optimizasyon, belirli bir çözüm kümesinden (ya da çözüm uzayından) en iyi çözümü seçmeyi amaçlayan bir optimizasyon dalıdır. Bu alan, matematiksel modelleme ve algoritmalar kullanarak en iyi çözümün bulunmasını hedefler. Kombinatorial optimizasyon, özellikle büyük veri setleri ve karmaşık sistemlerin bulunduğu ortamlarda, çok sayıda olasılığı değerlendirmeyi gerektirdiğinden oldukça zorlu bir alandır [71].

Kombinatorial optimizasyon problemleri genellikle "NP-zor" problemler olarak sınıflandırılır. Yani, çözüm bulmak için harcanacak zaman, giriş büyüklüğü ile orantılı olarak hızla artar [72]. Bu tür problemler için genellikle tam çözümler elde etmek zor olduğundan, yaklaşık çözümler üreten algoritmalar kullanılabilir.

Optimizasyon, mevcut uygun çözüm alternatifleri arasından en iyisinin seçilmesi sürecidir. Bir probleme ait tüm potansiyel çözümler *çözüm uzayı* (S) olarak tanımlanırken, tüm kısıtları sağlayan alternatif çözümler bu uzayın bir alt kümesini oluşturarak *uygun (ya da olurlu) çözüm kümesi* (S') olarak adlandırılır ($S' \subseteq S$). En iyi çözüm (S^*), uygun çözüm kümesi içinde; bir *minimizasyon problemi* için $f(S^*) \leq f(s)$ veya bir *maksimizasyon problemi* için $f(S^*) \geq f(s)$ koşulunu sağlayan çözüm olarak tanımlanır [73].

Kombinatorik, sonlu ya da kesikli küme yapılarıyla ilgilenen bir matematik dalı olup, belirli nesnelerin kaç farklı biçimde birleştirilebileceğini araştırır [73]. Bu bağlamda, kombinatorial optimizasyon, belirli ölçütler doğrultusunda en uygun kombinasyonu

bulmayı amaçlayan ve genellikle kesikli yapıya sahip problemlerin çözümüne odaklanan yöntemlerin genel adıdır. Bu tür problemler, sınırlı ve çoğunlukla küçük boyutlu kümelerden seçilen, birbirinden farklı bileşenlerin en uygun kombinasyonunun belirlenmesini hedeflemektedir [73].

Bir kombinatoriyal problem, $P(\delta, f)$ şeklinde tanımlanabilir ve şu unsurları içerir:

- Değişken kümesi: $X = \{x_1, x_2, \dots, x_n\}$,
- Her değişken için tanımlı değer kümeleri: $D_1, D_2, \dots, D_n$,
- Değişkenler arasında tanımlı kısıtlar,
- Amaç fonksiyonu: $f: D_1 \times D_2 \times \dots \times D_n \rightarrow R^+$, ve bu fonksiyonun minimize edilmesi hedeflenir.

Tüm uygun çözümler,

$$\delta = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ tüm kısıtları sağlar}\}$$

şeklinde ifade edilir. Burada s , aday çözüm; δ ise uygun çözüm uzayını temsil eder [73].

3.2.1 Kombinatoriyal optimizasyon problemlerine örnekler

Literatürde sıklıkla karşılaşılan bazı temel kombinatoriyal optimizasyon problemleri aşağıda özetlenmiştir.

3.2.1.1 Sırt çantası problemi (knapsack problem)

Belirli ağırlık ve kârlılık değerlerine sahip bir dizi öge arasından, kapasitesi sınırlı bir çantaya yerleştirilebilecek en kârlı alt kümenin seçilmesini konu alır. Bu problem, paketleme problemleri içinde özel bir alt sınıf olarak yer almakta olup; yükleme, kesme, sermaye bütçeleme ve çizelgeleme gibi çeşitli uygulama alanlarında soyut bir model olarak kullanılmaktadır [73].

3.2.1.2 Çizelgeleme problemleri (scheduling problems)

Belirli işlerin veya operasyonların, sınırlı sayıda kaynak üzerinde belirli bir sırayla gerçekleştirilmesini gerektiren problemlerdir. Bu bağlamda, özellikle Atölye Tipi Çizelgeleme Problemi (Job Shop Scheduling Problem - JSSP), literatürde önemli bir yer tutmaktadır. Bu problemin detayları, ilerleyen bölümlerde ele alınacaktır [73].

3.2.1.3 Araç rotalama problemi (vehicle routing problem)

Bir merkezden (depo) çıkan araçların, belirli sayıda müşteriye hizmet verdikten sonra tekrar merkeze dönmelerini sağlayacak en kısa mesafe ya da en düşük maliyetli rotaların belirlenmesi problemidir. Lojistik ve dağıtım planlaması alanlarında yaygın olarak ele alınmaktadır [73].

3.2.2 Kombinatorial optimizasyon ve çözüm yöntemleri

Aşağıdaki temellerle birlikte değerlendirdiğimizde kombinatorial optimizasyon, genellikle belirli bir hedefi en iyi şekilde gerçekleştirebilmek için yapılan karar alma süreçlerini inceleyen bir matematiksel alandır. Birçok farklı endüstriyel ve teorik uygulamaya sahip olan bu alana, özellikle RPA gibi çok disiplinli bir teknolojiyle etkileşim kurabilen bir rol atanabilir. Bu rol, tezimizin ana fikrini oluşturmakta olup, kombinatorial optimizasyonun altında yatan temel prensip, çözüm alanında en uygun sonuca ulaşmak için çeşitli seçimler yapmaktır. Kombinatorial optimizasyon, genellikle bir dizi seçenek arasından en uygun seçeneği belirlemeye yönelik algoritmalar ve yöntemler geliştirmeyi hedefleyen bir alandır.

Kombinatorial optimizasyon problemleri için literatürde birçok farklı çözüm yöntemi geliştirilmiştir. Bu yöntemler, genel olarak iki ana grupta incelenmektedir: kesin çözüm algoritmaları ve yaklaşık yöntemler (sezgisel ve meta-sezgisel yaklaşımlar). Bu çalışmada, çeşitli çözüm yöntemlerinin karşılaştırılması yapılacaktır [73].

3.2.2.1 Kesin çözüm yöntemleri

Kombinatorial optimizasyon problemleri için ilk yaklaşım genellikle tam çözüm yöntemleri kullanmaktır. Bu tür yöntemler, çözüm uzayındaki her olasılığı değerlendirerek en iyi çözümü bulmayı amaçlar. Ancak, bu yöntemlerin dezavantajı, çözüm uzayının büyüklüğüne bağlı olarak çözüm süresinin çok uzun olabilmesidir. Bu nedenle, NP-zor problemler için bu tür yöntemler genellikle pratikte uygulanabilir değildir.

Kesin çözüm algoritmaları, bir problemin optimum çözümünü garanti eden yöntemlerdir. Ancak bu algoritmalar, kombinatorial yapıdaki problemleri çoğunlukla polinomial zaman içinde çözemezler; dolayısıyla çözüm süresi, problem boyutunun artmasıyla birlikte üstel derecede büyüme eğilimi gösterir [73]. Bu nedenle, özellikle

büyük boyutlu problemlerde, kesin yöntemler uygulamada verimsiz ya da kullanılamaz hâle gelebilmektedir [73].

Dal – Sınır Algoritması (branch & bound):

Dal-sınır, tam sayılı programlama problemlerinde sıklıkla kullanılan en önemli çözüm yöntemlerinden biridir. Bu yöntem, çözüm alanını bir ağacın dalları gibi düşünür ve her dalda, çözümün kısıtlamaları karşılayıp karşılamadığını kontrol eder. Dal-Sınır Algoritması, enumerasyon yöntemine benzer şekilde tüm olası çözümleri inceleyen ve en iyi çözümü bulmak amacıyla muhtemel tüm çözümleri araştıran bir tekniktir. Ancak bu yöntem, daha akıllıca ve ileri görüşlü bir yaklaşım benimser. Temel prensip, tüm uygun çözümlerin akıllıca sıralanarak bir sıralama ağacı içinde düzenlenmesine dayanır. Bu ağaçtaki her düğüm, bir kısmi çözümü ifade etmektedir.

Dal-sınır yöntemi şu şekilde çalışır:

- *Dallanma (branching):* Bu, problemi daha küçük alt problemlere ayıran yinelemeli bir süreçtir. Dallanma süreci, ayrık optimizasyon problemleri oluşturur ve her bir alt problem bir düğüm olarak değerlendirilir. Bir düğüm yalnızca daha iyi bir çözüm potansiyeli taşıyorsa dallandırılır (aktif düğüm).
- *Sınırlama (bounding):* Alt problemin tüm uygun çözümlerinin amaç değeri için bir alt sınır (L) hesaplanır. Aynı zamanda, problemin çözülmesiyle elde edilen mevcut en iyi amaç değeri bir üst sınır (U) olarak tutulur. Eğer bir alt problemin alt sınırı L, mevcut üst sınır U'dan büyük veya ona eşitse, bu alt problem daha iyi bir çözüm sağlayamayacaktır. Dolayısıyla, ilgili düğümden devam etmeye gerek kalmaz ve bu aşama budama (pruning) olarak adlandırılır.

Dal-sınır, her bir adımda çözüm alanını sınırlayarak hesaplama maliyetlerini azaltmayı amaçlar ve genellikle doğru bir çözüm bulana kadar devam eder [74,75].

Dinamik programlama:

Dinamik programlama (DP), genellikle n değişkenli bir problemin optimum çözümünü, problemi n aşamaya ayırarak ve her aşamada tek değişkenli bir alt problemi çözerek bulan bir yöntemdir. Bu yaklaşım, büyük ve karmaşık bir problemi daha küçük ve daha yönetilebilir parçalara ayırarak çözümde hesaplama avantajı sağlar. Çoğu uygulamada, dinamik programlama bir problemin çözümünü son aşamadan başlayarak geriye doğru çalışarak elde eder (backward recursion) [73].

Dinamik programlama, yapısal olarak parçala ve fethet (divide and conquer) yöntemine benzer bir strateji izler [73]. Ancak temel farkı, aynı alt problemin birden fazla kez ortaya çıkması durumunda, her seferinde yeniden çözüm yapmak yerine, her bir alt problemi sadece bir kez çözüp sonuçlarını bir tabloda saklamasıdır. Bu durum, işlemlerin çözülmesini hızlandırır. Dinamik programlamada her aşama, önceki aşamalarla sırasal bir ilişki içerir ve her bir aşamadaki çözüm, tek başına problemin çözümü olmasa da optimum çözümün bir parçası olan bilgi taşır.

Dinamik programlamanın altında yatan temel prensip, "optimalite prensibi"dir [73]. Bu prensip, her bir aşamada, önceki kararların bir etkisi olmaksızın, gelecekteki optimum kararları verme ilkesine dayanır. Dinamik programlama, aşamaları birbirine bağlayan tekrar eden denklemler kullanarak her aşamanın optimum sonucunun tüm problemin optimum çözümüne ulaşılmasını sağlar [75].

Tam Sayılı Programlama (integer programming):

Günümüzün rekabetçi iş ortamında, kuruluşlar verimliliği artırmak, maliyetleri düşürmek ve operasyonel süreçleri optimize etmek için sürekli olarak yenilikçi yaklaşımlar aramaktadır. Bu bağlamda, RPA tekrarlayan ve kural tabanlı görevleri otomatikleştirmek için kullanılan ve giderek önem kazanan bir teknoloji olarak ortaya çıkmıştır [4]. RPA'nın kullanımının ve öneminin artması, bu problemin çözümü için daha verimli yöntemlere olan ihtiyacı beraberinde getirmektedir [4]. Bu tez çalışmasının odak noktası olan RPA için gerekli minimum robot sayısının belirlenmesi problemi, birçok kuruluş veya organizasyonlar için kritik bir öneme sahiptir [74]. Bu bölümle birlikte, bu problemin çözümünde temel bir araç olarak kullanılabilecek olan Tam Sayılı Programlama (Integer Programming) kavramı derinlemesine incelenecektir.

Tam Sayılı Programlama (IP), bir problemi çözmek için kullanılan matematiksel optimizasyon tekniklerinin bir alt kümesidir. Bu yöntem, bir problemin çözümünün belirli kısıtlamaları sağlaması ve karar değişkenlerinden bazılarının ya da tamamının tam sayı değerlerine sahip olması gerektiği durumlarda kullanılır. IP, kaynak tahsisi, zaman çizelgeleme ve kombinatoriyal optimizasyon problemleri gibi gerçek dünya sorunlarında kritik öneme sahiptir. Bu tür problemler sıklıkla ikili (0-1) değişkenler veya kapasite, miktar ya da karar gibi gerçek dünya koşullarını modelleyen diğer tam sayı değerleri içerir [75].

Kombinatorial optimizasyon problemleri, genellikle Matematiksel Programlama (Mathematical Programming) yaklaşımları kullanılarak formüle edilir. Özellikle, Karma Tam Sayılı Lineer Programlama (Mixed Integer Linear Programming- MIP), ayrık karar değişkenleri içeren problemlerin, özellikle çizelgeleme problemleri gibi, modellenmesinde yaygın olarak başvurulmuş bir yöntemdir.

Matematiksel modellemenin, özellikle MIP'in, problemleri tanımlamak için güçlü bir araç olduğu ifade edilse de çoğu optimizasyon problemi için etkin bir çözüm algoritmasının henüz bulunmadığı ve yüksek hesaplama süresi nedeniyle klasik Operasyon Araştırması yöntemlerinin (MIP veya Dinamik Programlama gibi) çoğu zaman sınırlı kullanıma sahiptir. Özellikle JSSP gibi problemlerin NP-hard (zorlu) olması, bu tür formülasyonların büyük ölçekli gerçek dünya problemleri için doğrudan kesin çözümler üretmesinin pratik olmadığını ortaya koymaktadır [73]. Bu nedenle, matematiksel modelleme, problemin yapısal özelliklerini anlamak ve kesin çözümlerin zorluklarını görmek için önemli bir ilk adımdır.

Tam sayılı programlama, finans, imalat, lojistik ve telekomünikasyon gibi çeşitli sektörlerde geniş bir uygulama alanına sahiptir. Bu tezde, RPA süreçlerinin robotlara atanmasını optimize etmek için kullanılacaktır. Bu, robotların sayısını en aza indirirken, işlem süreleri, görev hacimleri ve piyasa çalışma saatleri gibi operasyonel kısıtlamalara uyulmasını sağlayacaktır [4].

Tam sayılı programlama türleri:

Tam sayılı programlama için birkaç tür öne çıkmaktadır. Aşağıda bunlarla ilgili detay verilecektir.

Saf tam sayılı programlama:

Bu model türünde, problemin tüm karar değişkenleri tam sayı değerleri almak zorundadır. Örneğin, üretilecek parça sayısı veya görevlendirilecek kişi sayısı gibi nicelikler sadece tam sayılarla ifade edilebildiğinde bu tür bir model kullanılır [73].

İkili tam sayılı programlama:

Saf tam sayılı programlamanın bir alt kümesidir ve değişkenler yalnızca 0 veya 1 ile sınırlıdır. Bu, görevlerin atanması veya zaman çizelgeleri gibi durumlar için kullanılır. Bu tür problemlerde, görevler ya atanır ya da atanmaz [75].

Karma tam sayılı programlama:

Bu bölüm altında MIP özelinde temel kavramlar ve uygulama alanlarına, çözüm yöntemleri ve modelleme teknikleriyle ilgili bir inceleme yapılacaktır. MIP tezimizde kullanacağımız modelin temelini oluşturmaktadır. Minimum robot sayısının belirlenmesi için MIP modeli uygulanmıştır.

MIP karar değişkenlerinin bir kısmının tam sayı, diğer kısmının ise sürekli değerler almasına izin verilen doğrusal optimizasyon problemlerini ifade eder. Bu yaklaşım hem sürekli hem de ayrık karar değişkenlerinin bulunduğu birçok gerçek yaşam problemine uygulanabilirliği nedeniyle operasyon araştırmaları ve endüstriyel mühendislik gibi alanlarda önemli bir yer edinmiştir [75]. MIP, üretim planlamasından çizelgeleme problemlerine, kaynak tahsisinden rota optimizasyonuna kadar geniş bir uygulama alanı sunar. Özellikle belirli kararların yalnızca belirli kesikli değerler alabileceği durumlarda (örneğin bir tesisin açılıp açılmaması, bir makinenin kullanılıp kullanılmaması vb.) bu tür modeller oldukça etkilidir.

Karma tam sayılı programlama ve problemin modellenmesi:

RPA sistemlerinin tasarımı ve uygulanması sürecinde karşılaşılan temel karar problemlerinden biri, iş yükünü karşılamak üzere kaç adet yazılım robotunun tahsis edilmesi gerektiğidir. Bu problem, sınırlı kaynakların, değişken iş yüklerinin ve zaman kısıtlarının dikkate alındığı bir çizelgeleme ve kaynak atama problemidir. Bu bağlamda, karma tamsayılı programlama (MIP), söz konusu problem yapısını matematiksel olarak modellemek için güçlü bir araçtır [77].

RPA sistemlerinde gerçekleştirilen görevlerin çoğu, belirli bir süre içinde tamamlanması gereken ve belirli kaynaklara (örneğin robotlara) ihtiyaç duyan işlerdir. Bu işler, zaman içinde değişen iş yüküne, görev önceliklerine ve bazen ardışıklık ilişkilerine sahiptir. Bu tür özellikler, MIP modelleriyle ifade edilebilecek biçimdedir.

- Görevler bir zaman aralığında başlatılmalı ve belirli bir sürede tamamlanmalıdır.
- Her görev, yalnızca bir robota atanmalıdır.
- Robotların günlük maksimum görev kapasitesi vardır.
- Belirli görevler diğerlerinden önce gerçekleştirilmek zorundadır (öncelik ilişkileri).
- Robot sayısı minimum olacak şekilde sistem çalışmalıdır.

Minimum robot sayısı problemi için örnek MIP modeli ve modelleme önerisi sunacak olursak. Karma tamsayılı programlama (MIP), belirli görevlerin belirli kaynaklara en uygun şekilde atanmasını gerektiren problemler için oldukça güçlü bir modelleme aracıdır. RPA kapsamında, sistemin işleyişini aksatmadan en az sayıda yazılım robotu ile görevlerin tamamlanmasını sağlama problemi, klasik bir kaynak atama ve çizelgeleme problemi olarak ele alınabilir. Bu problem yapısı, MIP kapsamında sistematik biçimde formüle edilebilir.

MIP modellerinde temel amaç, karar değişkenleri ile modelin temsil ettiği sistemi optimize etmektir. Minimum robot sayısının belirlenmesi amacıyla oluşturulacak modelde, iki tür değişken tanımı ön plana çıkar: birincisi, görevlerin robotlara atanmasını gösteren ikili değişkenler; ikincisi ise hangi robotların aktif olduğunu belirleyen ikili kaynak değişkenleridir. Böylece hem kaynak kullanımını minimize etmeye hem de sistemin görev taleplerini karşılamaya yönelik iki yönlü bir karar yapısı oluşturulur [75].

Bu tür modellerde kullanılan değişkenler, yalnızca atama ilişkilerini değil, aynı zamanda zamanlama kararlarını da yansıtabilir. Özellikle zaman indeksli modellere başvurulması, görevlerin başlama ve bitiş anlarının daha keskin biçimde tanımlanmasına olanak sağlar [78].

Minimum robot sayısının belirlenmesinde kullanılan tipik amaç fonksiyonu, aktif robot sayısının minimizasyonudur. Ancak bu amaç, belirli hizmet düzeyleri, işlem süreleri ve ardışık görev ilişkileri gibi birçok sistem kısıtına bağlı olarak şekillenir. Her bir görev yalnızca bir kez atanmalı, görevler çakışmadan planlanmalı, görevlerin işlem süreleri dikkate alınmalı ve sistemin mantıksal işleyişine uygun biçimde ardışıklık kısıtları tanımlanmalıdır [77].

Bu tür mantıksal ilişkiler, MIP modellerinde oldukça yaygın biçimde kullanılan ikili değişkenler aracılığıyla temsil edilebilir. Bu bağlamda tamsayılı programlamanın yalnızca niceliksel değil, aynı zamanda yapısal ve mantıksal kısıtlar açısından da oldukça esnek bir yöntem olması önemli bir yer tutar [75].

Modelin matematiksel formunun kurulması kadar, karar değişkenlerinin ve kısıtların doğru seçilmesi de model başarısında belirleyicidir. Tamsayılı optimizasyon problemlerinde daha iyi çözüm kalitesi elde etmek için problem yapısının mümkün olduğunca sadeleştirilmesi ve dönüştürülmesi gerektiğini önemlidir [76]. Bu

kapsamda, minimum robot sayısı problemi için modelin temel stratejik ilkelerle geliştirilmesi önerilir.

Minimum robot sayısının belirlenmesi problemi, yalnızca teorik değil, aynı zamanda endüstriyel ve ticari uygulamalarda da önemli bir yer tutmaktadır. RPA, iş gücü verimliliğini artırmayı hedefleyen yazılım robotlarının kullanılmasıyla gerçekleştirilir. İşlem süreleri ve kaynak kısıtlamaları gibi faktörler, RPA sisteminin verimli çalışmasını doğrudan etkileyebilir. Bu nedenle, minimum robot sayısının belirlenmesi hem maliyetlerin düşürülmesi hem de sistem verimliliğinin artırılması için kritik bir rol oynamaktadır.

Burada, özellikle RPA bağlamında inceleyecek olsak da Tam Sayılı Programlama yalnızca RPA ile sınırlı olmayıp, aynı zamanda birçok diğer optimizasyon probleminin çözümünde de yaygın olarak kullanılmaktadır. Taşıma ve lojistik problemleri, makine üretim hatları planlaması, personel yerleştirme ve tesis yeri seçimi gibi çeşitli endüstriyel uygulamalarda TSP yaygın bir şekilde kullanılmaktadır. Bu tür problemlerin çoğunda, belirli kaynakların sınırlı bir şekilde tahsis edilmesi ve kararların belirli kurallara dayalı olarak alınması gerekliliği ortaya çıkmaktadır. Örneğin, bir lojistik firmasında, araçların hangi güzergâhları izleyeceği ve hangi ürünlerin hangi araçlarla taşınacağına karar verirken, bu süreç Tam Sayılı Programlama ile modellenebilir. Bu bağlamda, her aracın taşıma kapasitesi, güzergâh uzunlukları ve zaman kısıtlamaları gibi çeşitli faktörler dikkate alınarak optimal bir çözüm arayışı geliştirilir [79].

Sonuçta bu tezde, RPA için gerekli minimum robot sayısının belirlenmesi problemine yönelik etkili bir çözüm yöntemi geliştirilmesi amaçlanmıştır. TSP yöntemleri, RPA'nın görev atama süreçlerini optimize etmek ve robot sayısını en aza indirmek için güçlü bir araç olarak kullanılmıştır. Bu bağlamda, robot sayısının minimize edilmesine yönelik olarak geliştirilen Tam Sayılı Programlama modelleri, çözüm süresi ve doğruluk açısından önemli avantajlar sunmaktadır.

Robotların görev ataması ve kaynakların verimli kullanımı, özellikle büyük veri setleri ve karmaşık kısıtlamalar altında çözülmesi zor olan problemler arasındadır. Ancak, kullanılan tam sayılı programlama teknikleri ve uygun optimizasyon algoritmaları sayesinde problem başarıyla modellenip çözülebilir. Bu çözüm, RPA uygulamaları

için robot sayısının azaltılması, işlem sürelerinin iyileştirilmesi ve genel verimliliğin artırılması gibi hedeflere ulaşılmasına yardımcı olacaktır.

Tam Sayılı Programlama, özellikle kesikli karar değişkenleri gerektiren problemler için ideal bir çözüm yöntemi sunmaktadır. Bu tezde kullanılan karma tam sayılı programlama modellemesi, robot sayısını ve robotların görev atamalarını optimize ederken, operasyonel kısıtlamalarla uyumlu çözümler sunmuştur. Bu yöntem, özellikle RPA gibi sistemlerde, robotların sadece belirli görevleri yerine getirmesi ve zaman dilimlerinde sınırlı kaynak kullanımı gereksinimlerini karşılaması gibi pratik gereksinimleri doğru bir şekilde modelleyecektir. Bu çalışmada elde edilen sonuçlar, tam sayılı programlamanın RPA problemleri için etkili bir çözüm sunduğunu gösterebilir. Tam Sayılı Programlama, robot atama süreçlerinde önemli bir rol oynayacak ve kaynakların verimli bir şekilde kullanılmasını sağlayacaktır. Gelecekteki çalışmalar için, daha karmaşık ve dinamik problemlere çözüm getirmek amacıyla tam sayılı programlamanın daha da geliştirilmesi ve yeni teknolojilerle entegrasyonunu hedeflemelidir.

3.2.2.2 Meta-sezgisel yöntemler

Büyük ve karmaşık kombinatorial optimizasyon problemleri için tam çözüm yöntemlerinin yetersiz kalması durumunda, meta-sezgisel algoritmalar devreye girer. Bu algoritmalar, daha hızlı çözümler elde etmek amacıyla çözüm uzayını kesirli bir şekilde tarar. Meta-sezgisel yöntemler, genellikle yerel arama stratejilerini benimseyerek, çözüm uzayındaki bölgesel optimumlara ulaşmayı amaçlar.

Büyük ölçekli IP problemlerinde, çözüm bulma süresi çok uzun olabileceği için, kesin çözüm garantisi veren yöntemler yerine sezgisel (yaklaşık) yöntemler sıklıkla tercih edilir. Sezgisel yöntemler, genellikle problemin bir kısmını çözerek "iyi" bir çözüm bulmayı hedefler, ancak bu çözümün optimal olup olmadığı garanti edilmez. Sezgiseller, çözüm süresini kısaltmak ve pratik çözümler sunmak için yaygın olarak kullanılır.

Yaklaşık yöntemler, kesin çözümü garanti etmemekle birlikte, kabul edilebilir bir süre içerisinde çözüm üretme kapasitesine sahiptir. Özellikle NP-zor problemlerin çözümünde, en iyi çözümün makul bir zaman diliminde bulunmasının güçlüğü, bu tür yöntemlere olan ihtiyacı artırmıştır. Geleneksel sezgisel yöntemler (heuristics) genellikle belirli bir probleme özgü olup, farklı problemlere uygulanması zor olan

yaklaşımlardır [73]. Ayrıca, bu yöntemler çoğunlukla tek bir iyi çözüm elde etmeyi hedefler. Bu sınırlamaları aşmak için, probleme bağlı olmayan arama stratejileri geliştirilmiş ve buna meta-sezgiseller (metaheuristics) denilmiştir [73].

Meta-sezgiseller, geniş bir yelpazede farklı problemlere uygulanabilen sezgisel metotları tanımlamak için kullanılan bir kavramlar bütünüdür. Başka bir deyişle, meta-sezgisel, sınırlı bir problem alanında yapılan değişikliklerle farklı optimizasyon problemlerine uygulanabilen genel bir algoritmik çerçeve olarak kabul edilebilir. Meta-sezgiseller, alt sezgisellerin operasyonlarını yönlendiren ve değiştiren yinelemeli bir ana süreçtir ve her iterasyonda tek bir çözüm ya da bir çözüm koleksiyonunu manipüle edebilir.

3.2.2.3 Hibrit yöntemler

Son yıllarda, çeşitli meta-sezgisel yöntemlerin entegrasyonu ile daha güçlü hibrit algoritmalar geliştirilmiştir. Bu hibrit yaklaşımlar, farklı çözüm tekniklerinin bir arada kullanılmasıyla çözüm kalitesinin iyileştirilmesini hedeflemektedir. Örneğin, genetik algoritmalar ile simüle tavlama algoritmalarının kombinasyonu, her iki yöntemin avantajlarından yararlanarak daha verimli ve etkili sonuçlar elde edilmesini mümkün kılmaktadır [81]. Ayrıca, makine öğrenimi ve yapay zekâ tekniklerinin kombinatoriyal optimizasyon problemleriyle entegrasyonu da günümüzde önemli bir araştırma alanı olarak öne çıkmaktadır.

3.3 İş Çizelgeleme (Job Scheduling Problem)

İş çizelgeleme problemleri, bir dizi işin (tasks) belirli kaynaklar üzerinde (örneğin makineler, iş gücü) nasıl dağıtılacağına dair en iyi düzenlemeyi bulmayı amaçlamaktadır [72]. Bu tür problemler, özellikle üretim ve hizmet sektörlerinde sıkça karşılaşılan optimizasyon problemleri arasında yer almaktadır. Çizelgeleme, yalnızca işlerin sırasını belirlemekle sınırlı kalmaz; aynı zamanda işlerin süreleri, kaynak kısıtlamaları, öncelik sıralamaları ve benzeri etmenler de dikkate alınarak çözülmesi gereken karmaşık bir süreçtir.

İş çizelgeleme problemi, bir dizi işin belirli kaynaklar altında bir zaman diliminde yapılması gerektiğinde, bu işleri en verimli şekilde sıralamak ve dağıtmak amacıyla ortaya çıkan bir kombinatoriyal optimizasyon problemidir. Bu çerçevede, belirli bir dizi iş emrinin ya da daha genel bir ifadeyle "işlerin", sınırlı sayıda makine üzerinde

hangi sırayla ve ne zaman işleneceğini belirleme süreci söz konusudur. Her bir iş, ardışık üretim adımlarından oluşan ve "operasyon" olarak adlandırılan bir dizi faaliyetten meydana gelir [72]. Bu operasyonların, önceden tanımlanan bir teknolojik sıralamaya (proses yönlendirmesi veya öncelik kısıtları) uygun olarak, belirli makinelerde tamamlanması gerekmektedir [73]. Çizelgeleme, birden çok kaynak ve işin yönetilmesi gerektiği durumlarda, genellikle zaman kısıtlamaları ve kaynak kısıtlamaları ile karşı karşıya kalır. İş çizelgeleme probleminin amacı, genellikle her işin yapılması için en uygun zaman ve kaynakları belirlemektir.

Çizelgeleme ihtiyacı, makinelerin sınırlı kapasiteye sahip olması nedeniyle ortaya çıkar. Belirli bir anda, mevcut operasyonların makine kapasitesini aşması durumunda, aynı makineyi kullanmak isteyen farklı operasyonlar arasında kaynak çekişmesi meydana gelir [73]. Bu çekişmeyi çözebilmek için operasyonların belirli bir sıraya konulması gereklidir. Çizelgelemenin nihai amacı, her bir makine üzerinde yapılacak operasyonların sırasını hem öncelik hem de kapasite kısıtlamalarını göz önünde bulundurarak belirlemektir.

İş çizelgeleme problemleri, farklı sınıflandırmalara tabi tutulabilir. Bu sınıflandırmalar arasında en yaygın olarak karşılaşılanlar; tek makine çizelgelemesi, paralel makinelerle çizelgeleme ve atölye tipi çizelgelemedir. Örneğin, tek makine çizelgeleme problemi, belirli bir makinede gerçekleştirilecek işlerin sıralanmasını içermekte iken, paralel makinelerle çizelgeleme, birden fazla makinenin ve bu makinelerde yer alacak işlerin yönetilmesini kapsamaktadır [82].

RPA bağlamında iş çizelgeleme, robotların iş süreçlerini otomatikleştirme sürecinde önemli bir rol oynamaktadır. Herhangi bir organizasyonel süreç, birbirinden bağımsız birçok iş ve alt işten meydana gelir bu süreçlerin otomatikleştirilmesi, ilgili işlerin doğru bir şekilde sıralanmasını ve kaynakların en verimli biçimde kullanılmasını gerektirir. RPA sistemlerinde iş çizelgeleme probleminin optimizasyonu, robotların iş akışlarını daha etkin bir şekilde yönlendirebilmesini sağlamak amacıyla kritik bir öneme sahiptir.

3.3.1 Tek makine çizelgeleme

Tek makine çizelgeleme, üretim ve operasyon yönetimi alanında karşılaşılan en temel ve yaygın optimizasyon problemlerinden birini temsil etmektedir. Bu problemde, belirli bir dizi iş emri veya kısaca "işlerin", sınırlı kapasiteye sahip tek bir makine

üzerinde hangi sırayla işleneceği belirlenmektedir. Her bir işin kendine özgü bir işlem süresi mevcut olup, makine yalnızca bir işi aynı anda işleyebilmektedir. İşlemeye başlanan bir işin kesintisiz olarak tamamlanması gerektiği varsayımı, bu tür modellerin temel varsayımını oluşturmaktadır [72].

Bu tür problemlerin amacı, belirli bir performans ölçütünü optimize eden bir işleme sırasının oluşturulmasıdır. Genel iş çizelgeleme problemlerinde olduğu gibi, burada da hedef genellikle zaman odaklıdır. Tek makine çizelgeleme problemleri, işlem süreleri ve diğer parametrelerin belirlilik (deterministik) veya belirsizlik (stokastik) içerip içermediğine göre farklı kategorilere ayrılabilir [72]. Deterministik modellerde her işin işlem süresi, teslim tarihi ve diğer tüm ilgili parametreler önceden kesin olarak bilinir. Amaç, belirli bir performans ölçütünü en iyi duruma getiren optimal bir çizelge oluşturmaktır. Stokastik modelde ise işlerin işlem süreleri veya diğer parametreler olasılık dağılımlarıyla tanımlanan rastgele değişkenlerdir. Bu durum, belirsizliğin yönetilmesini gerektirdiğinden problemleri daha karmaşık hale getirir. Amaç genellikle beklenen performans ölçütlerini (örn. beklenen yapım süresi) optimize etmektir.

Tek makine çizelgeleme, daha karmaşık çok makineli sistemler, örneğin atölye tipi çizelgeleme (Job Shop Scheduling) için geliştirilen algoritmalar ve sezgisel yaklaşımlar için önemli bir temel ve başlangıç noktası oluşturabilir. Bu tür basit yapıdaki problemlerin anlaşılması ve etkin çözümlerinin geliştirilmesi, endüstriyel operasyonlarda verimliliği artırmak ve kaynakları en verimli şekilde kullanmak açısından kritik bir öneme sahiptir.

3.3.2 Paralel makine çizelgeleme

Paralel makine çizelgeleme, birden fazla, eş zamanlı olarak çalışabilen makine üzerinde işlerin atanması ve sıralanmasıyla ilgili optimizasyon problemlerini tanımlar. Bu senaryoda, belirli bir dizi işin (jobs) birden fazla makineye nasıl dağıtılacağı ve her makine üzerinde hangi sırayla işleneceği konusunda kararlar aynı anda verilmektedir. [81]. Tek makine çizelgelemesinin aksine, burada işlerin birden fazla makineye yönlendirilmesi ve makinelerin kapasitelerinin etkin bir biçimde kullanılması gerekliliği, problemi daha karmaşık hale getirmektedir. Makineler, özelliklerine göre üç ana kategoriye ayrılabilir [81]:

- *Özdeş Paralel Makineler:* Tüm makineler aynı işlem hızlarına ve yeteneklere sahiptir. Bu durumda, bir işin herhangi bir makinedeki işlem süresi eşittir[81].
- *Tekdüze Paralel Makineler:* Makineler farklı işlem hızlarına sahip olup, bir makinedeki işlem süresi, o makinenin hızına orantılıdır (örneğin, daha hızlı bir makine, işi daha kısa sürede tamamlar) [81].
- *İlgisiz Paralel Makineler:* Makineler arasında işlem hızları ve yetenekleri tamamen farklıdır. Bir işin işlem süresi, yalnızca o işin ve makinenin özel kombinasyonuna bağlıdır. Örneğin, aynı işin farklı makinelerdeki işlem süreleri birbirinden tamamen bağımsız olabilir [81].

Paralel makine çizelgeleme problemleri de temelde deterministik ve stokastik olmak üzere iki ana kategoride incelenmektedir [72]. Deterministik paralel makine modellerinde, işlerin işlem süreleri gibi tüm önemli parametreler önceden kesin olarak bilinir. Bu modellerde amaç, belirli bir performans ölçütünü optimize eden en uygun çizelgeyi oluşturmaktır. Stokastik paralel makine modellerinde ise deterministik modellerden farklı olarak, işlerin işlem süreleri veya diğer parametrelerin rastgele değişkenler olduğu durumları ele almaktadır [72]. Yani, bu parametrelerin kesin değerleri yerine, olasılık dağılımları bilinmektedir. Bu durum, çizelgeleme problemlerine ek bir karmaşıklık katmanı ekler. Stokastik modellerde amaç genellikle beklenen performans ölçütlerini optimize etmektir. Sonuç olarak, paralel makine çizelgeleme, kaynak tahsisi ve sıralamanın birden fazla makine üzerinde eşzamanlı olarak gerçekleştirildiği karmaşık bir optimizasyon alanını temsil etmektedir. Problemin deterministik veya stokastik doğasına bağlı olarak, farklı yaklaşımlar ve algoritmalar kullanılabilir.

3.3.3 Atölye tipi çizelgeleme

Atölye tipi çizelgeleme problemleri, belirli bir dizi işin (jobs), farklı makineler üzerinde, her bir işin belirli bir işlem sırasına (rota) göre ve makinelerin sınırlı kapasiteleri çerçevesinde nasıl işleneceğini belirlemeyi amaçlamaktadır. Bu tür bir sistemde, temel özellikler ve kısıtlar şu şekilde özetlenebilir:

- *İşlem Süreleri:* Her işlem, belirli bir makinede sabit bir süre boyunca devam etmektedir.
- *Makine Kısıtı:* Her makine, bir seferde yalnızca bir işlemi işleyebilmektedir.

- *Kesintisizlik*: Bir işin bir makine üzerindeki işlemi başladığında, genellikle kesintisiz olarak tamamlanması beklenmektedir. Bu bağlamda, kaynaklarda "ön işlem yapılmadan işlenir" ifadesi de kullanılmaktadır.
- *İşlem Sırası (Rota)*: Belirli bir işe ait operasyonlar, önceden belirlenmiş bir sırayla (teknolojik önceliklere göre) işlenmektedir.
- *Makine Erişilebilirliği*: Her makine, sıfır zamandan itibaren sürekli olarak kullanılabilir durumdadır.

Bir örnekle açıklamak gerekirse, farklı ürünler üreten ve her ürünün belirli makinelerde, belirli bir sırayla işlenmesi gereken bir üretim hattı düşünülebilir. Her ürünün kendine özgü bir rotası ve her makinenin sınırlı bir kapasitesi olduğundan, işlerin makineler üzerinde en verimli şekilde nasıl sıralanacağı, karmaşık bir optimizasyon problemi oluşturur. Atölye tipi çizelgeleme, operasyon araştırmaları ve endüstri mühendisliği alanlarında hem teorik hem de pratik anlamda önemli bir araştırma konusu olup, özellikle NP-zor yapısı, araştırmacıları ve uygulayıcıları, sürekli olarak yeni ve daha verimli çözüm yaklaşımları geliştirmeye yönlendirmektedir.

4. PROBLEM TANIMI

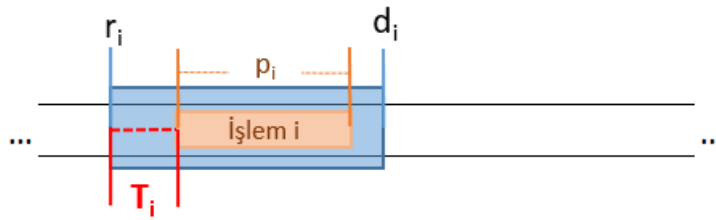
Bu bölümde, RPA için minimum robot sayısının belirlenmesi ve işlem atama/sıralama problemini ele alınmaktadır. Farklı özelliklere sahip (süre, yineleme, öncelik, yazılım gereksinimi vb.) rutin işlem türlerinin belirli kısıtlar altında robotlara nasıl atanacağını bu bölümde incelenmiştir. Amaç, toplam maliyeti (robot ve yazılım lisans maliyeti) ve işlemlerin başlangıç zamanlarındaki gecikmeyi en aza indirerek en uygun çözümü bulmaktır. Bu problem, “Karma Tam Sayılı Model” kullanılarak formüle edilmiş ve matematiksel kısıtlar ile detaylandırılmıştır.

4.1 Robotik Süreç Otomasyonunda Minimum Robot Sayısı Probleminin Kapsamlı Tanımı ve Matematiksel Formülasyonu

Bu çalışmada, kurallara dayalı rutin iş süreçlerini otomatikleştirmek üzere yazılım tabanlı bir çözüm olarak ortaya çıkan Robotik Süreç Otomasyonu (Robotic Process Automation - RPA) için belirli kısıtlar altında minimum robot sayısının belirlenmesi problemi ele alınmıştır. Burada robot, tekrarlanan görevleri yerine getirmek üzere hazırlanmış lisanslı bir yazılımdır. Robotlara pek çok rutin farklı işlem türü tanımlanabilmektedir. İşlem türleri farklı işlem süreleri, yineleme düzeni, işlenmesi için gerekli yazılım, öncül ardıl ilişkisi, öncelik seviyesi gibi özellikler taşımaktadır. Problemin daha rahat anlaşılabilmesi için açıklayıcı bir örnek oluşturulmuştur. Tablo 1’de, açıklayıcı örnekte kullanılan işlem türleri ve ile ilgili örnek veri seti sunulmuştur.

Tablo 1 ilk sütunda işlem türünün ID numarası verilmiştir. İkinci kolonda işlem türüne ait işlem süresi (p_i) sunulmuştur. İşlem süresi denilen kavram bir sürecin sürebileceği maksimum zaman olacağı varsayılmıştır (Normalde bu süre birçok nedene bağlı olarak değişkendir). İlgili işlem türüne ait tüm işlemlerin aynı sürede yürütüldüğü ve bu sürenin yenileme aralığını geçemeyeceği varsayılmaktadır (Yani bir saat sürece bir işlem için 15 dakikada bir başlatılması mümkün değildir). İşlemler atandıkları robotlarda işlenmek üzere kuyruğa alınır ve üçüncü kolonda belirlenen öncelik seviyesine (düşük, orta ve yüksek) göre sırayla işlenmelidir. Dördüncü kolonda verilen yineleme düzeni bir işlem türünün ne zaman başlaması gerektiği (serbest kalma

zamanı- r_i) ve ne aralıklarla yinleneceği bilgisini içerir. İlgili işlem türüne ait her yinelemeye işlem adı verilir. İşlemler, ikinci kolonda verilen serbest kalma zamanında oluşturulur. İşlemler mümkün olduğunca önceden belirlenen serbest kalma zamanında başlaması gerekmektedir. Bir işlemin başlama zamanında meydana gelecek gecikme ilgili işlemin önem seviyesine bağlı olarak ağırlıklandırılarak cezalandırılır. İşlemlerin başlama zamanlarındaki gecikme, cezalandırılarak yumuşak bir kısıt haline çevrilmiştir. Sağlanan bu esneklik ile her zaman bir uygun çözüm elde edilmesi amaçlanmıştır. Bu gecikme sınırsız değildir ve bir işlem en geç dokuzuncu kolonda belirtilen süre içerisinde tamamlanmalıdır. Aksi takdirde işlem ölür ki bunun gerçekleşmesi kabul edilmeyen sert bir kısıttır. Bu süre varsayılan olarak (ilgili hücre boş bırakılmış ise) ilgili işlem türü için iki işlem arasındaki süre kadardır. Bir işlem türünün son işlemi için teslim zamanı planlama ufkunun sonudur. Ancak kullanıcı bu süreyi kısaltabilmektedir. Özetle bir işlem, serbest kalma zamanı (r_i) ile tamamlanması gereken son zaman (d_i) arasında, özellikle serbest kalma zamanına yakın bir şekilde gerçekleştirilmelidir. Bir işlemin zaman eksenindeki pozisyonu ve ilgili kavramlar Şekil-4.1’de resmedilmiştir.



Şekil 4.1: Bir işlemin zaman eksenindeki pozisyonu.

İşlemin ölmesi durumunu şu şekilde açıklayabiliriz. Örneğin 15 dakika da bir üst yönetime iletilmesi gereken kritik bir raporlama işlem türünü ele alalım. Bu işlem 1 dakika sürsün. Bu işlemin başlama zamanı 14 dakikadan fazla gecikirse bu raporun bir anlamı olmayacaktır. Bu durum için işlemin ölmesi tabiri tercih edilmiştir. Bu çalışmadaki amaçlardan biri, toplam ağırlıklı ceza puanını minimize edecek şekilde işlemlerin robotlara atanması ve sıralanmasıdır. Her işlem önceden belirlenen işlem süresi sonunda tamamlandığı varsayılır. Yani teknik sebeplerden dolayı işlemimin durmadığı varsayılmaktadır. Her işlem türü en fazla bir robotta yürütülebileceği yani birden fazla robota ihtiyaç duyacak sıklıkta ve sürede işlemlere sahip olmadığı varsayılmaktadır. Ayrıca işlemler kesintisiz bir şekilde yürütülmekte, önalıma izin verilmemektedir. Beşinci kolonda işlem türleri için gerekli yazılımlar verilmiştir.

İşlemler bu yazılımları barındıran robotlara atanabilmektedir. Örnek vaka için hazırlanmış olan varsayımlara dayalı gerekli yazılımlar ve lisans bedelleri Çizelge 4.1’de verilmiştir.

Çizelge 4.1: Gerekli yazılım ve lisans bedelleri.

Gerekli Yazılımlar	Lisans Bedeli (TL/Yıl)
SAP/ERP bağlantı aracı	120000
Tarayıcı Otomasyon Modülü	60000
Veritabanı Bağlantısı (SQL Server, Oracle gibi bağlantı modülleri)	50000

Veri güvenliği gibi nedenlerden dolayı bazı departmanlara ait işlemler için özel robot(lar) tahsis edilmesi gerekmektedir. Bu robotlara ilgili işlem türü hariç başka bir işlem türü atanamamaktadır. Bu amaçla aynı robotta birleştirilmesi gereken işlemler altıncı kolonda verilen bölüm bilgisine göre yapılmaktadır. Bu şekilde belirli bölüm(ler)e ait işlem türleri aynı robot(lar)a atanması ve bu robot(lar)a bu bölüm(ler) haricinde başka bir işlem türünün atanmaması sağlanmış olur. Bu alan boş bırakılmış ise ilgili işlem türü için böyle bir sınırlamanın olmadığı anlamına gelmektedir. Bazı yazılımlar, aynı kullanıcı hesap bilgisi ile aynı anda farklı robotlardan işlem yapılmasına izin vermemektedir. Yedinci kolonda, eğer ilgili işlem türü kimlik doğrulama gerektiriyor ise, gerekli kullanıcı hesap bilgisi verilmiştir. Hesap bilgisi gerektiren işlemler farklı robotlarda da olsa aynı anda yapılmazlar. Sekizinci kolonda ilgili satırda tanımlanan işlem türünün varsa öncül işlem türü belirtilmektedir.

Planlama ufku bir hafta olarak belirlenmiştir. Yani işlem türlerinin ardışık işlemleri arası uzunluğun en fazla bir hafta olduğu varsayılmıştır.

Özetle işlem türlerinin robotlara atandığı bu çalışmada, minimum toplam maliyet (robot ve yazılım lisans maliyeti) ile işlemlerin mümkün olduğunca önceden planlanan serbest kalma zamanında başladığı (minimum toplam ağırlıklı gecikme) bir atama ve sıralama problemi ele alınmıştır.

4.1.1 Problemin karma tamsayılı modeli

Bu bölümde ele alınan karar probleminin karma tam sayılı modeli verilmiştir. Öncesinde modelde kullanılan indis/setler, parametreler ve karar değişkenleri tanımlanmıştır.

İndisler ve Setler

j, e : Robotlar ($j, e \in J$)

k, g : İşlem türü ($i, h \in I$)

i, f : İşlem ($i, f \in S_k$)

l : Kullanıcı hesabı ($l \in L$)

m : Bölüm ($m \in M$)

n : Lisanslı yazılımlar ($n \in N$)

Parametreler

Γ : Çok büyük bir sayı

p_i : i işlem türünün işlem süresi ($p_k = p_i \forall k \in i$)

r_i : i işinin başlaması gereken zaman (serbest kalma zamanı)

d_i : i işinin en geç tamamlanması gereken zaman

ϕ_i : Planlama ufku boyunca i işlem türüne ait işlem sayısı

ω_{in} : 1, i işlem türü için n yazılımı gerekli ise

0, Diğer durumlarda

c_n : n yazılımına ait lisanslama bedeli (bir robot için)

C : Robot kurulum maliyeti

pr_i : i işlem türünün öncelik seviyesi ($pr_k = pr_i \forall k \in i$)

τ_k : k işlem türüne ait işlemler kümesi $\{1, 2, \dots, \phi_k\}$

Θ_l : l kullanıcı hesabı gerektiren işlemler kümesi

η_m : m bölümüne ait işlem türü kümesi

Karar Değişkenleri

y_{kj} : 1, k işlem türü j robotuna atanmış ise

0, Diğer durumlarda

x_{ij} : 1, i işlemi j robotuna atanırsa

0, Diğer durumlarda

- z_{ifj} : 1, j robotuna atanmış iki işlemten i işlemi, f işleminden önce başlarsa
0, Diğer durumlarda
- v_{ifl} : 1, l kullanıcı adı gerektiren iki işlemten i işlemi, f işleminden önce başlarsa
0, Diğer durumlarda
- s_{ij} : i işleminin j robotunda başlama zamanı
- ss_{il} : l hesabı gerektiren i işleminin l başlama zamanı
- u_{nj} : 1, j robotuna n yazılımı yüklenmiş ise
0, Diğer durumlarda
- h_j : 1, j robotu kullanılmış ise
0, Diğer durumlarda
- α_{ifj} : Yardımcı değişken (0-1 değişkeni)
- T_i : i işleminin gecikme süresi

Karma Tam Sayılı Model

$$\text{Min. } Z_1 = \sum_n \sum_j c_n * u_{nj} + \sum_j h_j * C$$

$$\text{Min. } Z_2 = \sum_k t_k * pr_k$$

$$\begin{aligned} \sum_j y_{kj} &= 1 & \forall k & (1) \\ s_{ij} &\leq x_{ij} * \Gamma & \forall k, j & (2) \\ s_{fj} &\geq s_{ij} + p_i - (1 - z_{ifj}) * \Gamma & \forall i, f, \forall j; i \neq f & (3) \\ \sum_{g \in \eta_m} y_{gj} &\leq (1 - y_{kj}) * \Gamma & \forall k \in \eta_m, \forall j, m & (4) \\ \sum_{i \in \tau_k} x_{ij} &= \varphi_k * y_{kj} & \forall k, j & (5) \\ u_{nj} &\geq x_{ij} * \omega_{in} & \forall i, j, n & (6) \\ \sum_j s_{ij} &\geq r_i & \forall i & (7) \\ \sum_j s_{ij} &\leq d_i - p_i & \forall i & (8) \\ T_i &\geq \sum_j s_{ij} - d_i & \forall i & (9) \\ \sum_k y_{kj} &\leq h_j * \Gamma & \forall j & (10) \\ s_{ij} &\leq 10080 - p_i & \forall i, j & (11) \\ \alpha_{ifj} &\leq x_{ij} & \forall i, f, j; i \neq f & (12) \\ \alpha_{ifj} &\leq x_{fj} & \forall i, f, j; i \neq f & (13) \\ \alpha_{ifj} &\geq x_{ij} + x_{fj} - 1 & \forall i, f, j; i > f & (14) \\ z_{ifj} + z_{fij} &= \alpha_{ifj} & \forall i, f, j; i > f & (15) \\ \sum_j s_{ij} &= ss_{il} & \forall i, \in \Theta_l, \forall l & (16) \\ ss_{fl} &\geq ss_{il} + p_i - (1 - v_{ifl}) * \Gamma & \forall i, f \in \Theta_l \text{ and } i \neq f, \forall l & (17) \\ v_{ifl} + v_{fil} &= 1 & \forall i, f \in \Theta_l \text{ and } i \neq f, \forall l & (18) \end{aligned}$$

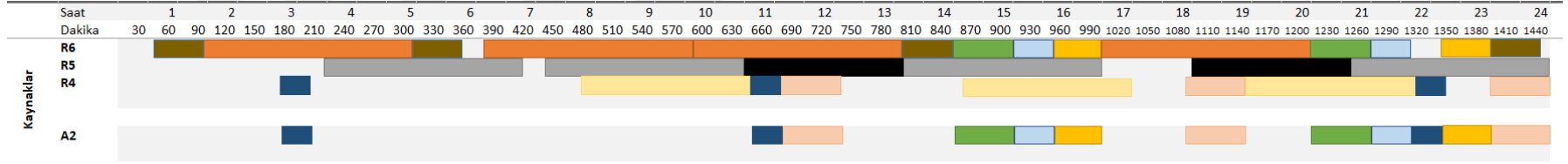
Birinci kısıtta, her k işlem türünün sadece bir j robotunda yürütülmesi sağlanmaktadır. Yani k işlem türüne ait tüm tekrarlı işlemler aynı robotta yapılması sağlayan y_{kj} 0-1 değişkeninin değeri belirlenmektedir. *İkinci kısıtta* eğer i işlemi j robotunda yürütülmeyecek ise s_{ij} değişkenine ait değer sıfırlanmaktadır. *Üçüncü kısıtta* aynı j robotuna atanmış i ve f işlemleri sıralanmaktadır. *Dördüncü kısıtta* m bölümüne ait işlem türlerinin, bu bölümde olmayan işlem türleri ile aynı robot(lar)a atanmaması sağlanmaktadır. *Beşinci kısıtta* bir k işlem türüne ait tüm tekrarlı işlemlerin ($i \in \tau_k$) robot(lar)a atanması sağlanmaktadır. *Altıncı kısıtta*, j robotuna n yazılımı gerektiren i işlem türüne ait bir işlem atanmış ise ilgili j robotuna n yazılımı kurulduğunu gösteren u_{nj} 0-1 değişkeninin değeri belirlenmektedir. *Yedinci kısıtta* her i işleminin, tayin edilen serbest kalma zamanından (r_i) önce başlamaması, *sekizinci kısıtta* ise en geç tamamlanması gereken zamanda (d_i) tamamlanması sağlanmaktadır. *Dokuzuncu kısıtta* her i işlemi için gecikme süresi (T_i) hesaplanmaktadır. *Onuncu kısıtta* bir j robotu en az bir k işlem türü tarafından kullanılmış ise h_k 0-1 değişkeninin değeri 1 olarak belirlenmektedir. Bu değişken amaçların biri olan toplam maliyet hesaplanırken, robot kurulum maliyetini ifade etmek için kullanılmıştır. *On birinci kısıtta* atama ve çizelgelemenin her bir robotun haftalık kapasitesini (7 gün*24saat*60dakika=10080 dakika) aşmayacak şekilde yapılması sağlanmaktadır. 12-15. kısıtlar, *on dokuzuncu* kısıtın doğrusallaştırılmış halidir.

$$z_{ifj} + z_{fij} = x_{ij} * x_{fj} \quad \forall i, f, j \quad (19)$$

16-18 kısıtları, l kullanıcı hesabı gerektiren i ve f işlemlerin farklı robotlara atanmış olsalar da aynı anda yürütülmemesi sağlanmamaktadır.

Çizelge 4.2: Açıklayıcı örnek işlem türleri.

İşlem Kodu	Tekrar Sayısı	Tekrar Saatleri	İşlem Süresi (dk)	Bölüm Tahsisi	Hesap Tipi	İşlem İçin Gerekli Yazılımlar		
						Yazılım-1	Yazılım-2	Yazılım-3
Process0	4	0:35 - 4:20 - 8:50 - 14:20	50			✓		
Process1	4	1:25 - 5:10 - 9:40 - 15:10	210					
Process2	4	3:25 - 7:10 - 11:40 - 17:10	200				✓	
Process3	2	15:15 - 21:15	50		Yönetici			
Process4	2	13:20 - 19:20	40		Yönetici			
Process5	2	11:50 - 17:55	60		Yönetici			
Process6	2	10:15 - 16:15	160					✓
Process7	3	2:45 - 6:00 - 12:00	30	Muhasebe	Yönetici			
Process8	3	11:5 - 14:20 - 20:20	60	Muhasebe	Yönetici	✓		
Process9	3	7:45 - 11:00 - 17:00	170	Muhasebe			✓	



Şekil 4.2: Açıklayıcı örneğe ait gantt grafiği.

4.1.2 Geliştirilen karma tamsayılı modelin kořumu ve değeriendirilmesi

Bu bölümde geliştirilen modelin doğruluęu ve geçerlilięi test edilmiřtir. Bunun için rastgele oluřturulmuř bir test problemi kullanılmıřtır. Oluřturulan test problemi 10 iřlem türünden oluřmaktadır. Bu iřlem türlerinin 24 saatlik bir zaman ufku içerisinde planlanması değeriendirilmiřtir. Test problemine iliřkin detaylar Çizelge 4.2’de sunulmuřtur. İlk kolonda iřlem türünün adı verilmiřtir. İkinci kolonda tekrar sayısı yani ilgili iřlem türüne ait iřlem sayısı verilmiřtir. İřlem sayısı bakımından değeriendirildięinde test problemi 29 iřten oluřmaktadır. Üçüncü kolonda iřlemlerin bařlaması geren saatler verilmiřtir. Dördüncü kolonda her iřleme ait iřlem süresi dakika cinsinden verilmiřtir. Beřinci kolonda, beřinci kısıtta ifade edilen durumu test etmek için üç iřlem türüne bölüm tanımlanmıřtır. Altıncı kolonda dördüncü kısıtı test etmek için bazı iřlem türlerine hesap tipi tanımlanmıřtır. Son üç kolonda ilgili iřlem türlerinin gerçekiřtirilebilmesi için robotta olması geren yazılımlar belirtilmiřtir.

Model GUROBI kütüphanesini kullanarak C# diliyle kodlanmıřtır. Problemler Core(TM) 2 Quad CPU, 2.66 GHz iřlemci and 4 GB RAM özelliklerine sahip bir bilgisayarda kořulmuřtur. Elde edilen sonuç Şekil 4.2’de verilen gantt diyagramı ile görselleřtirilmiřtir. Gantt diyagramında her iřlem Çizelge4.2’de ilgili iřlem türü için belirlenen renkte ifade edilmiřtir. Diyagramın solunda robot numarası verilmiřtir. Yani bu problem 3 robot gerektirecek şekilde çizelgelenmiřtir. A2 ile hesap tipi ifade edilmektedir. Yönetici hesap tipi gerektiren iřlemlerin aynı anda yapılmadıęı yani bir çakiřmanın olmadı gösterilmektedir (dördüncü kısıt). Muhasebe departmanına ait iřlem türleri (7, 8 ve 9) Robot 4’de atanmıř ve bu robota bu bölüm den olmayan bařka bir iřlem türü atanmamıřtır (beřinci kısıt). Gantt diyagramı incelendięinde dięer tüm kısıtların saęlandıęı görölmektedir.

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, RPA teknolojisinin uygulama sürecinde karşılaşılan temel optimizasyon problemlerinden biri olan minimum robot sayısının belirlenmesi sorununa etkili ve uygulanabilir bir çözüm geliştirilmesi amaçlanmıştır. Dijital dönüşüm çağında, işletmelerin artan rekabet koşulları ve verimlilik gereksinimleri doğrultusunda, tekrarlayan ve kural tabanlı işlemlerin insan müdahalesine gerek kalmadan yazılım robotları tarafından yürütülmesini sağlayan RPA, önemli bir otomasyon aracı hâline gelmiştir. Bu kapsamda RPA sistemlerinin etkin şekilde yapılandırılması, zaman ve kaynak yönetimi açısından kritik bir rol oynamaktadır.

Çalışma kapsamında, süreç sayısı, işlem süresi, öncelik derecesi ve yineleme sıklığı gibi çeşitli parametreler göz önünde bulundurularak, minimum sayıda robot lisansı kullanımıyla maksimum verim elde edilmesi hedeflenmiştir. Bu problem, belirli işlem türlerinin farklı zaman dilimlerinde ve işlem sürelerinde gerçekleştirilmesini içeren klasik bir iş çizelgeleme ve kaynak atama problemi şeklinde ele alınmıştır. Her bir işlem türünün belirli bir zaman boyunca yalnızca bir robot tarafından yürütülebildiği ve her robotun aynı anda yalnızca tek bir görevi yerine getirebildiği varsayılmıştır. Ayrıca, veri güvenliği, yazılım gereksinimleri ve departman bazlı robot tahsisi gibi operasyonel kısıtlar da modele entegre edilmiştir.

İşlemlerin, mümkün olduğunca önceden tanımlanan başlama zamanlarında başlatılması, belirli bir gecikme durumunda işlemlere önem düzeyine bağlı olarak ceza uygulanması ve bazı işlemlerin belirli bir sürede tamamlanması gerekliliği gibi esnek ve zorunlu kısıtlar dikkate alınarak bir çizelgeleme yapısı oluşturulmuştur.

Önerilen çözüm yaklaşımı, Karma Tam Sayılı Programlama (Mixed-Integer Programming, MIP) çerçevesinde geliştirilmiştir. MIP, karar değişkenlerinin bir kısmının tam sayı, diğerlerinin sürekli değerler alabildiği güçlü bir doğrusal optimizasyon modelleme yöntemidir. Bu yaklaşım sayesinde, yazılım robotlarının iş yüklerini üstlenmesiyle toplam robot sayısının ve buna bağlı maliyetlerin azaltılması hedeflenmiştir.

Bu çalışma, RPA teknolojisinin etkin bir şekilde kullanılması hâlinde operasyonel maliyetlerin düşürülebileceğini, hata oranlarının azaltılabileceğini ve iş süreçlerinin verimliliğinin artırılabilceğini ortaya koymaktadır. Robot kaynaklarının optimal biçimde çizelgelenmesi sayesinde hem maliyetler düşürülmekte hem de robot yetersizliğinden kaynaklanabilecek darboğazların önüne geçilmektedir. Böylece, organizasyonların kısıtlı kaynaklarla maksimum çıktı elde etme hedeflerine katkı sağlanmaktadır.

Gelecek Çalışmalar İçin Öneriler:

- *Daha Karmaşık ve Dinamik Problemlere Uyarlanabilirlik:* Mevcut modelin, RPA ortamlarında değişken iş yüklerini ve süreç dinamiklerini dikkate alacak biçimde olay tabanlı simülasyon ya da uyarlanabilir çizelgeleme yaklaşımlarıyla geliştirilmesi önerilmektedir.
- *Yapay Zekâ ve Makine Öğrenmesi Entegrasyonu:* Sistemlerin daha akıllı ve esnek hâle getirilmesi için yapay zekâ ve makine öğrenmesi tekniklerinin, özellikle derin öğrenme ve doğal dil işleme yöntemlerinin, modele entegre edilmesi potansiyel fayda sunmaktadır.
- *Çok Amaçlı Optimizasyon Yaklaşımları:* Yalnızca maliyet minimizasyonunu değil, aynı zamanda işlem süresi (makespan), kaynak kullanım dengesi gibi diğer performans kriterlerini de kapsayan çok amaçlı optimizasyon modelleri üzerine çalışmalar önerilmektedir.
- *Sektörel Farklılıkların Modellenmesi:* Geliştirilen yöntemin farklı sektörlerdeki (finans, sağlık, eğitim, perakende vb.) uygulama gerekliliklerine göre genişletilmesi, yöntemin uygulanabilirliğini artıracaktır.
- *Gelişmiş Sezgisel ve Meta-Sezgisel Yöntemler:* Gerçek dünyaya yönelik büyük ölçekli problemlerde daha hızlı ve etkin çözümler elde edebilmek için GRASP, ACO, genetik algoritmalar gibi meta-sezgisel yaklaşımların iyileştirilmesi ve hibrit modellerin geliştirilmesi önerilmektedir.

Sonuç olarak, bu tez çalışması yalnızca RPA teknolojisinin uygulama verimliliğini artırmakla kalmamakta, aynı zamanda optimizasyon mühendisliği ve bilişim sistemleri alanlarında yeni araştırma perspektifleri de sunmaktadır.

KAYNAKLAR

- [1] **Özdem, H., Bora, M. P.** (2022). Türkiye’de robotik süreç otomasyonu, *Bilgisayar Bilimleri ve Teknolojileri Dergisi*, 3 (1), 1-9. doi: 10.54047/bibted.1008340.
- [2] **Doruk, G.** (2022). *Robotik süreç otomasyonu için iş çizelgeleme algoritması*. (Yüksek lisans tezi). İstanbul Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.
- [3] **Ivančić, L., Suša Vugec, D., Bosilj Vukšić, V.** (2019). Robotic process automation: Systematic literature review. *Business Process Management: Blockchain and Central and Eastern Europe Forum: BPM 2019 Blockchain and CEE Forum*, Vienna, Austria, September 1–6, 2019, Proceedings 17 (s. 280-295). Springer International Publishing.
- [4] **Séguin, S., Tremblay, H., Benkalaï, I., Perron-Chouinard, D. E., Lebeuf, X.** (2021). Minimizing the number of robots required for a Robotic Process Automation (RPA) problem, *Procedia Computer Science*, 192, 2689-2698. doi: 10.1016/j.procs.2021.09.039.
- [5] **Tremblay, H., Séguin, S., Boily, L. A., Paul, V. D., Lalancette, S.** (2023). Robotic Process Automation (RPA) using a heuristic method and the effective resistance of a graph, *Procedia Computer Science*, 225, 3388-3394. doi: 10.1016/j.procs.2023.10.333.
- [6] **Hendrawan, S. A., Chatra, A., Iman, N., Hidayatullah, S., Suprayitno, D.** (2024). Digital transformation in MSMEs: Challenges and opportunities in technology management, *JİDT*, 6 (2), 141-149. doi: 10.60083/jidt.v6i2.551.
- [7] **Turkyılmaz, S.** (2024). Dijital dönüşüm: Tarihçesi, tanımı ve işletmeler üzerindeki etkisi. *Nişantaşı Üniversitesi Sosyal Bilimler Dergisi*, 12 (1), 276-297.
- [8] **Costa, D. A. D. S., Mamede, H. S., Mira Da Silva, M.** (2022). Robotic Process Automation (RPA) Adoption: A systematic literature review, *Engineering Management in Production and Services*, 14 (2), 1-12. doi: 10.2478/emj-2022-0012.
- [9] **Kregel, I., Koch, J., Plattfaut, R.** (2021). Beyond the hype: Robotic Process Automation’s public perception over time, *Journal of Organizational Computing and Electronic Commerce*, 31 (2), 130-150. doi: 10.1080/10919392.2021.1911586.
- [10] **Januszewski, A., Kujawski, J., Buchalska-Sugajska, N.** (2021). Benefits of and obstacles to RPA implementation in accounting firms, *Procedia Computer Science*, 192, 4672-4680. doi: 10.1016/j.procs.2021.09.245.

- [11] **Kaya, C. T., Turkyilmaz, M., Birol, B.** (2019). RPA teknolojilerinin muhasebe sistemleri üzerindeki etkisi, *Muhasebe ve Finansman Dergisi*, (82), 235-250. doi: 10.25095/mufad.536083.
- [12] **Ortiz, F. C. M., Costa, C. J.** (2020). RPA in finance: Supporting portfolio management: Applying a software robot in a portfolio optimization problem, *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*. Sevilla, Spain: IEEE. doi: 10.23919/CISTI49556.2020.9141155.
- [13] **Cho, S., Moon, J., Bae, J., Kang, J., Lee, S.** (2023). A framework for understanding unstructured financial documents using RPA and multimodal approach, *Electronics*, 12 (4), 939. doi: 10.3390/electronics12040939.
- [14] **Kothandapani, H. P.** Applications of robotic process automation in quantitative risk assessment in financial institutions. *International Journal of Business Intelligence and Big Data Analytics*, 6 (1), 40-52.
- [15] **Venigandla, K., Tatikonda, V. M.** (2021). Improving diagnostic imaging analysis with RPA and deep learning technologies, *Power System Technology*, 45(4), 237-249.
- [16] **Kedziora, D., Smolander, K.** (2022). Responding to Healthcare emergency outbreak of COVID-19 pandemic with Robotic Process Automation (RPA), *Program Adi: Hawaii International Conference on System Sciences*. doi: 10.24251/HICSS.2022.752.
- [17] **Venigandla, K., Vemuri, N., Thaneeru, N., Tatikonda, V. M.** (2023). Leveraging AI-Enhanced robotic process automation for retail pricing optimization: A comprehensive analysis, *JKLST*, 2 (2), 361-370. doi: 10.60087/jklst.vol2.n2.p370.
- [18] **Harmoko, H., Rahul, M., Axmann, B.** (2023). The assessment of robotic process automation projects with a portfolio analysis: First step - evaluation criteria identification and introduction of the portfolio concept, *Teh. Glas. (Online)*, 17 (2), 207-214. doi: 10.31803/tg-20230416193006.
- [19] **Lasso-Rodríguez, G., Gil-Herrera, R.** (2019). Robotic process automation applied to education: A new kind of robot teacher? *Program Adi: 12th Annual International Conference of Education, Research and Innovation*, Seville, Spain. doi: 10.21125/iceri.2019.0669.
- [20] **Johnson, F., Eide, S. S.** (2018). *An inquiry into robotic process automation implementation in institutions for higher education.* (Yüksek lisans tezi). Norwegian School of Economics, Bergen.
- [21] **Patil, S., Mane, V., Patil, P.** (2019). Social innovation in education system by using Robotic Process Automation (Rpa), *IJITEE*, 8 (1), 3757-3760, doi: 10.35940/ijitee.K2148.0981119
- [22] **Boban, D., Nakić, J.** (2024). Enhancing informal education with robotic process automation: A method for automating course search on e-learning platforms, *J. Commun. Softw. Syst. (Online)*, 20 (2), 215-225. doi: 10.24138/jcomss-2023-0178.

- [23] **Greca, S., Zymeri, D., Kosta, A.** (2022). The implementation of RPA in case of scholarship in University of Tirana during the Covid time. ICSNS XXI-2022, International Institute for Private- Commercial- and Competition Law.
- [24] **Madakam, S., Holmukhe, R. M., Jaiswal, D. K.** (2019). The future digital work force: Robotic Process Automation (RPA), *JISTEM USP*, 16. doi: 10.4301/S1807-1775201916001.
- [25] **Mohamed, S. A., Mahmoud, M. A., Mahdi, M. N., Mostafa, S. A.** Improving efficiency and effectiveness of robotic process automation in human resource management, *Sustainability*, 14 (7), 3920. doi: 10.3390/su14073920.
- [26] **Mhaske, H., Kulkarni, S., Menon, V., Nikam, P., Niras, B.** Development of PrimeBot as an assistant to HR in recruitment process using RPA, *International Journal of Engineering Research*, 8 (5), 97-99.
- [27] **Pingili, R.** (2024). The integration of generative ai in rpa for enhanced insurance claims processing. *Technology (JARET)*, 3 (2), 38-52.
- [28] **Peddammukkula, P. K.** (2024). The role and types of automation in the life insurance industry. *International Journal of Computer Engineering and Technology*, 15(5), 296-306.
- [29] **Adeoye, O. B., Okoye, Onyeka C. C., Ofodile, C., Odeyemi, O., Afua Addy, W., Ajayi-Nifise, A. O.** (2024). Integrating artificial intelligence in personalized insurance products: A pathway to enhanced customer engagement, *Int. J. Manag. Entrep. Res*, 6 (3), 502-511. doi: 10.51594/ijmer.v6i3.840.
- [30] **Wiktorsson, M. (2024).** *Transforming insurance claims management.* (Yüksek lisans tezi). KTH Industrial Engineering and Management, Stockholm.
- [31] **Bhardwaj, V.** (2023). A systematic review of robotic process automation in business operations: Contemporary trends and insights, *JISC*, 2 (3), 153-169. doi: 10.56578/jisc020304.
- [32] **Patrício, L., De Sousa Costa, C. R., Varela, L., Cruz-Cunha, M. M.** (2024). Literature review on the sustainable implementation of Robotic Process Automation (RPA) in medical and healthcare administrative services, *Procedia Computer Science*, 239, 166-176. doi: 10.1016/j.procs.2024.06.159.
- [33] **Costa, D. A. D. S., Mamede, H. S., Mira Da Silva, M.** (2022). Robotic Process Automation (RPA) adoption: A systematic literature review, *Engineering Management in Production and Services*, 14 (2), 1-12. doi: 10.2478/emj-2022-0012.
- [34] **Gajjar, N., Rathod, K., Jani, K.** (2022). A systematic literature review on Robotic Process Automation security. *arXiv*: arXiv:2212.05544. doi: 10.48550/arXiv.2212.05544.
- [35] **Siderska, J., Aunimo, L., Süße, T., Stamm, J. V., Kedziora, D., Aini, S. N. B. M.** (2023). Towards Intelligent Automation (IA): Literature review on the evolution of Robotic Process Automation (RPA), its challenges, and

- future trends, *Engineering Management in Production and Services*, 15 (4), 90-103. doi: 10.2478/emj-2023-0030.
- [36] **Stravinskienė, I., Serafinas, D.** (2021). Process management and robotic process automation: The insights from systematic literature review, *Management of Organizations: Systematic Research*, 85 (1), 87-106. doi: 10.1515/mosr-2021-0006.
- [37] **Santos, F., Pereira, R., Vasconcelos, J. B.** (2019). Toward robotic process automation implementation: An end-to-end perspective, *BPMJ*, 26 (2), 405-420. doi: 10.1108/BPMJ-12-2018-0380.
- [38] **Syed, R., ve ark.** (2020). Robotic process automation: Contemporary themes and challenges, *Computers in Industry*, 115, 103162. doi: 10.1016/j.compind.2019.103162.
- [39] **Enriquez, J. G., Jimenez-Ramirez, A., Dominguez-Mayo, F. J., Garcia-Garcia, J. A.** (2020). Robotic process automation: A scientific and industrial systematic mapping study, *IEEE Access*, 8, 39113-39129. doi: 10.1109/ACCESS.2020.2974934.
- [40] **Wewerka, J., Reichert, M.** (2020). Robotic process automation -- a systematic literature review and assessment framework. *arXiv*: arXiv:2012.11951. doi: 10.48550/arXiv.2012.11951
- [41] **Hirwe, S. B., Kashyap, A. K., Patle, B.** A review on involvement of AI, RPA and IoT in human resource management. *Abhivruddhi Journal*, 3 (1), 51-57.
- [42] **Benkalai, I., Seguin, S., Tremblay, H., Glangine, G.** (2020). Computing a lower bound for the solution of a Robotic Process Automation (RPA) problem using network flows, *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, Prague, Czech Republic: IEEE, Haz. doi: 10.1109/CoDIT49905.2020.9263933.
- [43] **Seguin, S., Benkalai, I., Tremblay, H., Lebeuf, X.** Combining heuristics and integer linear programming to compute a solution to the Robotic Process Automation (RPA) problem. *Les Cahiers du GERAD*, 711, 2440.
- [44] **Laalaoui, Y., M'Hallah, R.** (2022). A problem-independent search heuristic for single machine scheduling with release dates and deadlines, *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, Singapore, Singapore: IEEE, Ara. 2022. doi: 10.1109/SSCI51031.2022.10022172
- [45] **Mallem, M., Hanen, C., Munier-Kordon, A.** (2024). A New Structural Parameter on Single Machine Scheduling with Release Dates and Deadlines. A. Basu, A. R. Mahjoub ve J. J. Salazar González, (Ed.), *Combinatorial Optimization, Lecture Notes in Computer Science* (Cilt. 14594, s. 205-219). Cham: Springer Nature Switzerland. doi: 10.1007/978-3-031-60924-4_16
- [46] **Chu, C.** (1992). A branch and bound algorithm to minimize total tardiness with different release dates, *Naval Research Logistics*, 39 (2), 265-283. doi: 10.1002/1520-6750(199203)39:2<265::AID-NAV3220390209>3.0.CO;2-L.

- [47] **Baptiste, P., Carlier, J., Jouget, A.** (2004). A branch-and-bound procedure to minimize total tardiness on one machine with arbitrary release dates, *European Journal of Operational Research*, 158 (3), 595-608. doi: 10.1016/S0377-2217(03)00378-3.
- [48] **Samuel, A. L.** (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3 (3), 210-229.
- [49] **Yugandhara, R. Y.** (2023). Robotic process automation market trend and size report 2023.
- [50] **Maalla, A.** (2019). Development prospect and application feasibility analysis of robotic process automation, *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China: Aralık. doi: 10.1109/IAEAC47372.2019.8997983.
- [51] **Rizvi, A., Srivastava, N.** (2023). Exploring the potentials of robotic process automation: A review, *J. Infor. Electr. Electron. Eng.*, 4 (2), 1-12. doi: 10.54060/jieee.2023.100.
- [52] **Syed, R., Suriadi, S., Adams, M., Bandara, W., Leemans, S. J., Ouyang, C., ... Reijers, H. A.** (2020). Robotic Process Automation: Contemporary themes and challenges, *Computers in Industry*, 115, 103162. doi: 10.1016/j.compind.2019.103162.
- [53] **Börü, F., Zafer, M. R.** (2024). Dijitalleşme çağında robotların ve robotik süreç otomasyonunun ekonomik etkileri, *EPF Journal*, 9 (2), 396-424. doi: 10.30784/epfad.1450440.
- [54] **Yetiz, F., Turan, Y., Canpolat, İ.** (2021). Bankacılık sektöründe robotik süreç otomasyonu ve verimlilik ilişkisi: Bir banka örneği, *Verimlilik Dergisi*, (2), 65-80.
- [55] **Hacıhasanoğlu, T., İğde, M.** (2023). *Accounting and Auditing on the Axis of Current Developments*. Ozgur Press. doi: 10.58830/ozgur.pub77.
- [56] **Seçkiner, S., Atay, M., Eroğlu, Y.** (2021). Robotik süreç otomasyonlarının havacılık sektörü uygulamaları ve geleceği, *JAV*, 5 (2), 290-297. doi: 10.30518/jav.947025.
- [57] **Cewe, C., Koch, D., Mertens, R.** (2018). Minimal Effort Requirements Engineering for Robotic Process Automation with Test Driven Development and Screen Recording, E. Teniente ve M. Weidlich, (Ed.), *Business Process Management Workshops, Lecture Notes in Business Information Processing* (Cilt. 308, ss. 642-648). Cham: Springer International Publishing. doi: 10.1007/978-3-319-74030-0_51.
- [58] **Axmann, B., Harmoko, H.** (2022). Process & Software selection for Robotic Process Automation (RPA), *Teh. Glas. (Online)*, 16 (3), 412-419. doi: 10.31803/tg-20220417182552.
- [59] **Jokela, E.** (2018). How to improve software robots production efficiency with scheduling. (Yüksek lisans tezi). Tampere University of Technology.
- [60] **Wanner, J., Fischer, M., Janiesch, C., Hofmann, A., Imgrund, F., Geyer-Klingenberg, J.** Process selection in RPA projects – towards a quantifiable method of decision making. *Process Selection in RPA*

Projects, Fortieth International Conference on Information Systems, Munich 2019.

- [61] **Van Der Aalst, W. M. P., Bichler, M., Heinzl, A.** (2018). Robotic Process automation, *Bus Inf Syst Eng*, 60 (4), 269-272. doi: 10.1007/s12599-018-0542-4.
- [62] **Taulli, T.** (2020). *The Robotic Process Automation Handbook*. A Press.
- [63] **Choi, D., R'bigui, H., Cho, C.** (2021). Candidate digital tasks selection methodology for automation with robotic process automation, *Sustainability*, 13 (16), 8980. doi: 10.3390/su13168980.
- [64] **Thesis, M., Strömberg, K.** Robotic process automation of office work: Benefits, challenges and capability development.
- [65] **Kirchmer, M., Franz, P.** (2019). Value-Driven Robotic Process Automation (RPA): A Process-Led Approach to Fast Results at Minimal Risk. B. Shishkov, (Ed.), *Business Modeling and Software Design, Lecture Notes in Business Information Processing* (Cilt. 356, ss. 31-46). Cham: Springer International Publishing. doi: 10.1007/978-3-030-24854-3_3.
- [66] **Ghosh, G.** (2018). A review on robotic process automation- the future of business organizations, *International Journal of Computer Sciences and Engineering*, 6 (8), 475-477. doi: 10.26438/ijcse/v6i8.475477.
- [67] **Ribeiro, J., Lima, R., Eckhardt, T., Paiva, S.** (2021). Robotic Process Automation and artificial intelligence in industry 4.0 – a literature review, *Procedia Computer Science*, 181, 51-58. doi: 10.1016/j.procs.2021.01.104.
- [68] **Daptardar, S.** A review- the golden triangle of RPA, AI and digital transformation, *International Research Journal of Modernization in Engineering Technology and Science*, 3 (1), 887-891.
- [69] **Willcocks, D. L., Hindle, D. J., Lacity, D. M.** (2018). *How blue prism clients gain superior long-term business value*. SS&C Blue Prism.
- [70] **Doruk, G.** (2022). *Robotik süreç otomasyonu için iş çizelgeleme algoritması*. (Yüksek lisans tezi). İTÜ, İstanbul.
- [71] **Gendreau, M., Potvin, J. Y.** (2019). Handbook of Metaheuristics. *International Series in Operations Research & Management Science* (s. 37-56). Springer.
- [72] **Pinedo, M. L.** (2008). *Scheduling: Theory, Algorithms, and Systems*. New York: Springer.
- [73] **Zäpfel, G., Braune, R., Bögl, M.** (2010). *Metaheuristic Search Concepts: A Tutorial with Applications to Production and Logistics*. Springer.
- [74] **Seguin, S., Benkalai, I., Tremblay, H., Lebeuf, X.** (2020). Combining heuristics and integer linear programming to compute a solution to the Robotic Process Automation (RPA) problem. *Cybernetics and Systems*, 51 (1), 1-10.
- [75] **Cornuejols, G., Trick, M. A., Saltzman, M. J.** (1995). A tutorial on integer programming. <https://www.math.clemson.edu/~mjs/courses/mthsc.440/integer.pdf>, erişim tarihi 29.06.2024.

- [76] **Glover, F.** (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22 (4), 455-460.
- [77] **Floudas, C. A., Lin, X.** (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139, 131-162.
- [78] **Artigues, C., Koné, O., Lopez, P., Mongeau, M.** (2014). Mixed-integer Linear Programming Formulations. *Handbook on Project Management and Scheduling* (s. 17-41). Cham: Springer International Publishing.
- [79] **Wolsey, L. A., Nemhauser, G. L.** (1999). *Integer and Combinatorial Optimization*. John Wiley & Sons.
- [80] **Gartner Magic Quadrant for Robotic Process Automation**
<https://www.gartner.com/en/documents>
- [81] **Shafiei Nikabadi, M., Naderi, R.** (2016). A hybrid algorithm for unrelated parallel machines scheduling, *International Journal of Industrial Engineering Computations*, 681-702. doi: 10.5267/j.ijiec.2016.2.004.
- [82] **Bhatt, N., Chauhan, N. R.** (2015). Genetic algorithm applications on job shop scheduling problem: A review, *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, Faridabad, India: IEEE, Ekim. doi: 10.1109/ICSCTI.2015.7489556.

ÖZGEÇMİŞ

TARANMIŞ
VESİKALIK
FOTOĞRAF

Ad-SOYAD : Anıl ÖZKAPAN

ÖĞRENİM DURUMU :

- **Lisans** : 2021, Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Makine Mühendisliği