

**ATÖLYELERDE OTOMATİK KUMANDALI
ARAÇLARIN OPTİMUM SAYILARININ GENETİK
ALGORİTMA VE YAPAY SİNİR AĞLARI İLE
BELİRLENMESİ**

Onur Mesut ŞENARAS



**T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ATÖLYELERDE OTOMATİK KUMANDALI ARAÇLARIN OPTİMUM
SAYILARININ GENETİK ALGORİTMA VE YAPAY SİNİR AĞLARI İLE
BELİRLENMESİ**

**Onur Mesut ŞENARAS
0000-0002-4295-801X**

**Prof. Dr. Ferruh ÖZTÜRK
(Danışman)**

**DOKTORA TEZİ
OTOMOTİV MÜHENDİSLİĞİ ANABİLİM DALI**

**BURSA – 2023
Her Hakkı Saklıdır**

ÖZET

Doktora Tezi

ATÖLYELERDE OTOMATİK KUMANDALI ARAÇLARIN OPTİMUM SAYILARININ GENETİK ALGORİTMA VE YAPAY SİNİR AĞLARI İLE BELİRLENMESİ

Onur Mesut ŞENARAS

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Otomotiv Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ferruh ÖZTÜRK

Maliyetleri düşürmek için atölyelerde Otomatik Kumandalı Araçlar (OKA'lar) yaygın olarak kullanılmaktadır. Ancak, OKA kullanımı aynı zamanda bir sermaye yatırımı olduğu için, OKA projeleri ciddi analiz gerektirmektedir. Önerilen yaklaşım ile maliyetli ve zaman alıcı analizlere ihtiyaç duyulmadan OKA kullanım oranlarının artırılmasıyla gerekli OKA sayısının hesaplanması amaçlanmıştır. Bir veya daha fazla OKA havuz sisteminin kurulması ile OKA filusunun optimal büyüklüğünü belirlemek için matematiksel bir model geliştirilmiştir. Bu amaçla karma tamsayılı doğrusal programlama modeli kullanılmıştır. Artan OKA hatları sürecinde oluşan çözüm kısıtlamalarını gidermek için her OKA hattı ve park istasyonuna göre gerekli OKA miktarını hesaplayan bir algoritma geliştirilmiş, genetik algoritma uygulaması ile optimal gerekli OKA sayısı elde edilmiştir. Sistem parametrelerinin gerekli OKA sayısına etkisini belirlemek için deney tasarımı yapılmış ve gerekli OKA sayısını tahmin etmek için bir yapay sinir ağı modeli geliştirilmiştir. Geliştirilen benzetim modelleri ile elde edilen sonuçların üretim talebini karşıladığı gösterilmiştir. Sistem parametrelerinin değişimi ile değişen OKA ihtiyacı başarılı bir şekilde tahmin edilmiştir. Geliştirilen yaklaşım ile üretim süreçlerindeki değişikliklerin gerekli OKA sayısı üzerine etkisi analiz edilmiş ve OKA sayısının azaltılması için hangi parametrelerde değişiklik yapılmasına gereksinim olduğu maliyetli ve zaman alıcı analizlere ihtiyaç duyulmadan belirlenmiştir.

Anahtar Kelimeler: Otomatik Kumandalı Araç, Otomatik Kumandalı Araç Havuzu, Genetik Algoritma, Yapay Sinir Ağları.

2023, vii + 90 sayfa.

ABSTRACT

PhD Thesis

DETERMINATION OF OPTIMUM NUMBER OF AUTOMATED GUIDED VEHICLES USING GENETIC ALGORITHM AND ARTIFICIAL NEURAL NETWORKS IN WORKSHOPS

Onur Mesut ŞENARAS

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Automative Engineering

Supervisor: Prof. Dr. Ferruh ÖZTÜRK

Automated Guided Vehicles (AGVs) in workshops are widely used to reduce the costs. However, this is also a capital investment that requires a deep analysis of the system components. The present approach was introduced to minimize the AGV number by increasing AGV utilization through the AGV pool without costly computation and time-consuming analysis. A mathematical model was developed to determine the optimal size of the AGV fleet by establishing one or more AGV pools. Mixed-integer linear programming was employed. Subsequently, a heuristic genetic algorithm was developed to determine the minimum number of AGVs for cases, which may have limitations to solve with mathematical modelling with mixed integer programming. Design of experiment was employed and an artificial neural network model was developed to determine the fleet size for circuit combinations and parking stations. The results demonstrated that the proposed approach allowed the prediction of the optimal number of AGVs and the determination of the effects of the parametric changes required to satisfy manufacturing demand. This plays an essential role in the evaluating the processes and defining business strategy according to different production requirement conditions. It can be concluded that the present approach can be used for low cost AGV systems in workshops while satisfying the demand in manufacturing without costly computation and time-consuming analysis.

Key words: Automated guided vehicle, Automated guided vehicle pool, Genetic algorithm, Artificial neural networks.

2023, vii+ 90 pages.

TEŐEKKÜR

Öncelikle bana doktora yapma imkânı tanıyan, çalışmamda devamlı desteęi ve katkılarıyla çalışmanın sonuca ulaşmasında ve karşılaşılan güçlüklerin aşılmasında büyük emeęi olan danışmanım Sayın Prof. Dr. Ferruh Öztürk'e sonsuz teşekkürlerimi sunarım. Ayrıca değerli katkılarından dolayı Sayın Prof. Dr. Nursel Öztürk'e ve Sayın Dr. Öğr. Üyesi Erol Solmaz'a minnetlerimi sunarım.

Özel sektörde çalışırken doktora yapmak sadece yapan kişi için değil, o kişinin ailesi için de fedakârlık gerektirmesi sebebiyle eşim Arzu Eren Şenaras' a, kızlarım; Doęa Irmak'a ve Ada Defne'ye, kayınpederim rahmetli Ahmet Eren'e ve kayınvalidem Hüsniye Eren'e teşekkürlerimi sunarım.

Onur Mesut ŞENARAS
03/07/2023

İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
SİMGELER ve KISALTMALAR DİZİNİ	v
ŞEKİLLER.....	vi
ÇİZELGELER.....	vi
1. GİRİŞ	1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI	4
2.1. Otomatik Kumandalı Araçlar	4
2.2. OKA Sisteminin Bileşenleri.....	6
2.3. OKA Kontrol Seviyeleri	8
2.4. Kaynak Araştırması.....	9
3. MATERYAL VE YÖNTEM	15
3.1. Tam Sayılı Programlama	15
3.2. Genetik Algoritma.....	18
3.3. Yapay Sinir Ağları	21
3.4. Yüzey Yanıt Metodu	25
3.5. Kesikli Sistem Simülasyonu	27
4. BULGULAR	30
4.1. Problemin Tanımlanması	33
4.2. OKA Tur Zamanının Hesaplanması (Ts).....	38
4.3. OKA Kombinasyonları ve Park İstasyonları İçin Gerekli OKA Miktarının Hesaplanması	39
4.4. Karma Tamsayı Programlama ile Gerekli Optimum OKA Sayısının Hesaplanması:	45
4.5. Gerekli OKA Sayısının Optimizasyonu için Genetik Algoritma Modelinin Geliştirilmesi:.....	47
4.5. Yapay Sinir Ağları ile Gerekli Optimum OKA Sayısının Tahmin Edilmesi.....	61
4.6. Değişen Sistem Parametrelerinin Etkilerinin Analizi	64
5. TARTIŞMA ve SONUÇ	80
KAYNAKLAR	84
ÖZGEÇMİŞ	89

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
I	Park istasyonlarının kümesi
S	OKA hatlarının kümesi
J	OKA hatlarının kombinasyonun kümesi
B	Bir park istasyonuna atanmış toplam OKA hatlarının sayısı
Ω_s	s. OKA hattı için OKA hat kombinasyonu kümesi, $s \in S$
GP_{ij}	i. park istasyonu ve j. OKA kombinasyonu için gerekli olan OKA miktarı
UP_{ij}	Karma tamsayılı lineer program modeli için 2'li (binary) karar değişkeni
O	Taşıma emirleri Kümesi
o	Taşıma emri $o \in O$ olmak üzere
l	Tüketim postasına bir sonraki OKA gelmesi için gereken süre
a	İzin verilen hizmet süresi başlangıcı
b	İzin verilen hizmet süresi sonu
k	OKA numarası
t_s	Bir taşıma işleminin gerçekleştirilmesi için gereken süre
f_{ok}	k numaralı OKA'nın o emrine atandığı süre
e_{ok}	k numaralı OKA'nın o numaralı emirden kurtulduğu ve OKA'ya yeni bir taşıma emri atanabilecek an
Kisaltmalar	Açıklama
GA	Genetik Algoritma
GPS	Global Positioning System (Küresel Konumlandırma Sistemi)
Lidar	Light Detection And Ranging (Işık Algılama ve Menzil)
OKA	Otomatik Kumandalı Araçlar
RFID	Radio Frequency Identification (Radyo Frekansı Tanımlama)
YSA	Yapay Sinir Ağları
ÇLR	Çoklu Lineer Regresyon
YYM	Yüzey Yanıt Metodu

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1.	OKA türleri	5
Şekil 2.2.	OKA' ların tarihi.....	5
Şekil 2.3.	OKA sistemi bileşenleri.....	6
Şekil 2.4.	Lazer navigasyon yöntemi	7
Şekil 2.5.	Yerden enerji verilmesi ile OKA'nın yönlendirilmesi.....	8
Şekil 3.1.	Yapay sinir hücresi fonksiyonları	23
Şekil 3.2.	Aktivasyon fonksiyonu	24
Şekil 3.3.	YSA'ların yapısı	24
Şekil 3.4.	Simülasyon çalışmasının aşamaları	28
Şekil 4.1.	Önerilen yaklaşımın adımları.....	31
Şekil 4.2.	Taşıma arabası	34
Şekil 4.3.	Yerleşim planı.....	35
Şekil 4.4.	Havuz düzeni ile çalışma şekli.....	38
Şekil 4.5.	Önerilen algoritma	43
Şekil 4.6.	Akış çizgesi.....	44
Şekil 4.7.	Genetik algoritmasının sözde kodu.....	48
Şekil 4.8.	İterasyonlar süresinde elde edilen gerekli OKA miktarı.....	51
Şekil 4.9.	Witness ekran görüntüsü.....	53
Şekil 4.10.	Makina doluluk oranları.....	53
Şekil 4.11.	Senaryo 1 için Witness ekran görüntüsü.....	58
Şekil 4.12.	Senaryo 2 için Witness ekran görüntüsü.....	59
Şekil 4.13.	Senaryo 2 için YSA mimarisi	61
Şekil 4.14.	Senaryo 2 için YSA performans grafikleri	62
Şekil 4.15.	Mutlak yüzde hata grafiği	64
Şekil 4.16.	Deney setlerine ilişkin yerleşim planı.....	65
Şekil 4.17.	Standart etkilerin pareto diyagramı.....	69
Şekil 4.18.	Faktörlerin etki grafiği	70
Şekil 4.19.	Faktörlerin etkileşim grafiği	71
Şekil 4.20.	Faktörlerin yüzey yanıt grafiği.....	71
Şekil 4.21.	YSA mimarisi	72
Şekil 4.22.	Yapay sinir ağları Matlab model görüntüsü.....	73
Şekil 4.23.	Senaryo 1 için YSA performans grafikleri	74
Şekil 4.24.	YSA'nın yeni mimarisi	76
Şekil 4.25.	YSA'nın performans değerleri.....	77

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 4.1. OKA hatları sistem parametreleri	36
Çizelge 4.2. Park istasyonları ve diğer istasyonlar arası mesafe.....	37
Çizelge 4.3. OKA hatları ve oluşturulan kombinasyonlar	40
Çizelge 4.4. Zaman pencere yöntemi örneği.....	42
Çizelge 4.5. Park istasyonları OKA kombinasyonları ve gerekli OKA miktarları..	46
Çizelge 4.6. Kromozom örnekleri	48
Çizelge 4.7. Uygunluk fonksiyonunun hesaplanması	49
Çizelge 4.8. OKA sayısını sağlayan diğer kromozomlar	51
Çizelge 4.9. 12 OKA sayısını gerektiren kromozom değerleri.....	52
Çizelge 4.10. Makinaların doluluk oranları	54
Çizelge 4.11. Araç doluluk oranları	55
Çizelge 4.12. Deney seti.....	56
Çizelge 4.13. Deney seti sonuçları.....	57
Çizelge 4.14. Senaryo 1 için makine doluluk oranları	58
Çizelge 4.15. Senaryo 1 için OKA doluluk oranları	59
Çizelge 4.16. Senaryo 2 için makina doluluk oranları	60
Çizelge 4.17. Senaryo 2 için OKA doluluk oranları	60
Çizelge 4.18. YSA tahminleri ile GA sonuçlarının karşılaştırılması	63
Çizelge 4.19. Faktörler ve seviyeleri.....	66
Çizelge 4.20. Deney tasarımı tablosu.....	67
Çizelge 4.21. Varyans analizi tablosu	68
Çizelge 4.22. YSA girdi ve çıktı değerleri	75
Çizelge 4.23. YSA, YYM ve ÇLR modellerinin sonuçlarının karşılaştırılması.....	78

1. GİRİŞ

Günümüz pazar koşullarında, üretim süreçlerinde kaliteyi arttıracak aynı zamanda tedarik süresi ile maliyetleri azaltacak yeni yaklaşımların uygulanmasına gereksinim duyulmaktadır. Üretim performansını pratik, verimli, düşük maliyetli yöntemler kullanarak iyileştirmek için yeni yaklaşımlar gereklidir. Otomatik kumandalı araçların (OKA) kullanılması bu yaklaşımlar arasındadır ve maliyetleri azaltmak için yaygın bir şekilde kullanılmaktadır. OKA, otomobil üretim fabrikaları, yarı iletken imalat atölyeleri, esnek imalat sistemleri (FMS'ler) ve liman konteyner terminalleri dahil olmak üzere imalat ve hizmet endüstrilerinde sıklıkla kullanılmaktadır. OKA kullanımı, performans artırıcı bir yöntem olmak ile birlikte bir yatırımdır ve dikkatli bir şekilde analiz edilmelidir. Özellikle kullanılan OKA miktarının, OKA'ların çalışma koşulları, kurulum ve çalışma maliyetleri üzerinde önemli bir etkisi vardır (Lee ve Ventura, 2001; Van der Meer, 2000; Ventura ve Rieksts, 2009). Yüksek yatırım maliyetleri, şirketleri düşük maliyetli OKA çözümlerini benimsemeye zorlamaktadır. Bu tip OKA'lar üzerlerinde çoğunlukla yüksek kapasiteli bilgisayarlar yerine Programlanabilir Lojik Kontrolcü (PLC) veya gömülü devreler içerirler ve sabit güzergâhta çalışmaktadırlar. Genellikle bilimsel literatürde çokça ele alınmayan bu düşük maliyetli OKA'ların etkinliğini arttırmak, yatırım maliyetlerini daha ekonomik hale getirecektir. OKA hatlarının gruplandırılarak ve bu hatların en uygun park istasyonlarına atanması ile optimum miktarının elde edilmesi, bu çalışmada yapılan literatür taraması sonucu elde edilen bilgilere göre henüz detaylı olarak incelenmemiş bir konudur.

Tezin konusu, yeni çalışmalara gereksinim duyulan bu alanda düşük maliyetli (low-cost) OKA etkinliğini arttırmak ve bu sayede gerekli OKA miktarının azaltılması ile gerekli yatırım maliyetinin azaltılmasına yöneliktir. Düşük maliyetli OKA'ların, sadece bir hatta çalışmasıyla etkinliği azalmaktadır. Gereksiz OKA beklmelerinin önüne geçilmesi için yüksek maliyetli merkezi yönetim sistemleri yerine düşük maliyetli (desantralize) çözümlerle OKA hatları kombinasyonu ile gerekli OKA miktarı azaltılabilir. Merkezi yönetim sistemleri yüksek seviye otomasyon ve bunun karşılığında maliyet içerirken, merkezi yönetilmeyen bir çözüm uygun maliyetli bir PLC'den oluşur. Bu çözümün nasıl uygulanacağı tezin çalışma konusunu oluşturmaktadır. Yapılan tez çalışması ile gerekli

OKA sayısının en az olacak şekilde belirlenmesi ve optimum OKA sayısının bulunması sürecinin en az maliyet ve sürede yapılabilmesi için bir yaklaşım geliştirilmiştir.

Problemin tanımı yapılarak, ilk aşamada zaman pencere yaklaşımı ile matematiksel modelleme yapılmıştır. OKA havuz sistemlerinde hangi OKA hat kombinasyonunun hangi park istasyonuna atanmasıyla optimal sonuç elde edilmesiyle ilgili olarak öncelikle zaman pencere yaklaşımıyla karma tamsayılı model oluşturulmuştur. Küçük sistemler (3-4 OKA hattı) için sınanan yöntem mevcut ele alınan problem için hem modelin oluşturulması hem de oluşturulan modelin çözümü açısından uygun değildir. Basit OKA hatları ile optimum çözüm üreten matematiksel yöntemin, OKA hat sayısı ve park istasyonu sayısı artınca uygulanmasında zorluklar yaşanmaktadır. Bu nedenle, karmaşık yapıları ifade edilen, matematiksel modeli çözümünde zorluklar içeren problemlerde kullanılan sezgisel algoritmalarından en yaygın kullanılan Genetik Algoritma (GA) tabanlı bir çözüm geliştirilmiştir. Bu yaklaşımın uygulanması ile OKA sayısında %29 kazanç elde edilmiş ve oluşturulan kesikli olay benzetimi ile yöntem doğrulanması sağlanmıştır. Önerilen yaklaşım, farklı veri setleri ile test edilmiş ve sonuçlar benzetim modeliyle doğrulanmıştır. Sabit OKA parametreleriyle oluşturulan modelde, farklı parametrelerin gerekli OKA sayısına etkisi incelenemeye üzere önce yüzey yanıt yöntemi kullanılmıştır.. Daha sonra geliştirilen yapay sinir ağı modeliyle gerekli OKA sayısı tahmin edilmiştir. OKA hatlarının park istasyonlarına atanması ile gerekli OKA sayısı tahmini için ikinci yapay sinir ağı modeli geliştirilmiş ve gerekli OKA sayısı başarı ile gerçekleştirilmiştir. Dolayısıyla bu tez çalışmasında basit, karmaşık ve çok karmaşık OKA hatları için farklı yaklaşımlar ile düşük maliyetli OKA'lerden oluşan OKA havuzları kurulmasıyla OKA etkinliğinin artmasına yönelik gerekli OKA sayısının en aza indirgenmesi amaçlanmaktadır.

Yapay sinir ağları ve yüzey yanıt yöntemi ile sistem parametrelerinin gerekli OKA sayısı üzerindeki etkisi incelenmiş ve OKA sayısını azaltmak için öncelikle taşıma arabalarının kapasitelerinin artırılmasının önemi vurgulanmıştır. Taşıma kapasitesi artışının, OKA hızından, şarj süresinden ve mesafelerden daha fazla önemli olduğu belirtilmiştir.

Tez çalışmasının kısa özeti ve amaçları aşağıdaki şekilde tanımlanabilir: Maliyetleri düşürmek için atölyelerde OKA'lar yaygın olarak kullanılmaktadır. Ancak, OKA

kullanımı aynı zamanda bir sermaye yatırımı olduđu için, OKA projeleri ciddi analiz gerektirmektedir. Önerilen yaklaşım ile maliyetli ve zaman alıcı analizlere ihtiyaç duyulmadan OKA kullanım oranlarının artırılmasıyla gerekli OKA sayısının hesaplanması amaçlanmaktadır. Bir veya daha fazla OKA havuz sisteminin kurulması ile OKA filusunun optimal büyüklüğünü belirlemek için matematiksel bir model geliştirilmiştir. Bu amaçla karma tamsayılı doğrusal programlama modeli kullanılmıştır. Artan OKA hatları sürecinde oluşan çözüm kısıtlamalarını gidermek için her OKA hattı ve park istasyonuna göre gerekli OKA miktarını hesaplayan genetik algoritma uygulaması ile optimal gerekli OKA miktarı elde edilmiştir. Sistem parametrelerinin gerekli OKA sayısına etkisini belirlemek için deney tasarımı yapılmıştır. Maliyetli ve zaman alıcı analizlere gereksinim duyulmadan filo büyüklüğünü tahmin etmek için bir yapay sinir ağı modeli geliştirilmiştir. Geliştirilen benzetim modelleri ile elde edilen sonuçların üretim talebini karşıladığı gösterilmiştir. Ayrıca sistem parametrelerinin değişimi ile değişen OKA ihtiyacı başarılı bir şekilde tahmin edilmiştir. Geliştirilen yaklaşım ile üretim süreçlerindeki değişikliklerin gerekli OKA sayısı üzerine etkisi analiz edilmiş ve OKA sayısının azaltılması için hangi parametrelerde değişiklik yapılmasına gereksinim olduğu maliyetli ve zaman alıcı analizlere ihtiyaç duyulmadan belirlenmiştir.

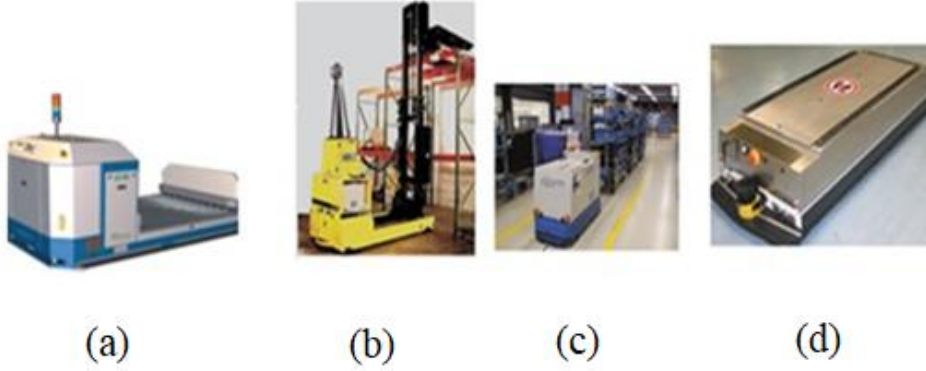
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Bu bölümde ilk olarak Otomatik Kumandalı Araçlar anlatılmış, sonrasında araştırmada ele alınan yöntemler incelenerek kaynak araştırması sonuçları ele alınmıştır.

2.1. Otomatik Kumandalı Araçlar

Malzeme taşıma, imalat sistemlerinin önemli bir parçasıdır. Ürüne değer katmayan bir faaliyet olarak görülen bu aktiviteler günümüzde otomatikleştirmeye çalışılarak işletmelerin performanslarının artırılması hedeflenmektedir. Taşıma maliyetlerinin akıllı çözümlerle otomatikleştirilmesi, işletmelerin genel giderlerini azaltacaktır. Bu uygulamalardan birisi de Otomatik Kumandalı Araç sistemlerinin kullanılmasıdır (Ferreira ve Gorch, 2016). Birçok sektörde OKA'ların kullanımı hızla artmaktadır. OKA'lar malzeme taşımada işgücü maliyetlerini ortadan kaldırmanın yanında aynı zamanda faaliyetlerini güvenlik aygıtlarıyla kesintisiz gerçekleştirdiği için iş güvenliğini de iyileştirmektedirler (Shivanand, 2006).

OKA'lar intralojistikte de kullanılmaktadır. Forklift OKA, Konveyörlü OKA, çekici OKA, yükün altına giren OKA bu alanda çalışmaktadır. Konveyörlü otomatik kumandalı araç Şekil 2.1.a'da, forklift otomatik kumandalı araç Şekil 2.1.b'de, çekici otomatik kumandalı araç Şekil 2.1.c'de, yük altına giren otomatik kumandalı araç Şekil 2.1.d'de gösterilmektedir.



Şekil 2.1. OKA türleri (Ullrich, 2015, s. 135-137, Shivanand, 2006, s. 61)

Günümüzde OKA'lar sadece malzeme taşımada etkili bir çözüm olarak görülmemekte ve son yıllarda OKA'lar aynı zamanda sıklıkla montaj hattı olarak da değerlendirilmektedir. Bu sayede esnek, düşük yatırım maliyetli bir üretim tesisi elde edilmiş olmaktadır. Otomotiv endüstrisinde sıkça bu tip uygulamalara rastlanmaktadır.

OKA sistemi, en az bir araç ve araca eşlik eden diğer çevresel bileşenler olmak üzere; şarj istasyonları, yükleme ve boşaltma istasyonları, merkezi kontrol bilgisayarları, kavşak yönetim uygulamalarından oluşmaktadır (Wu ve diğerleri, 2015; Colling ve diğerleri, 2019).

OKA'lar 1953'ten itibaren kullanılmaktadır. İlk OKA örnekleri çekicilerdir. 1960'larda gerçekleşen sensör ve mikrodenetleyicilerdeki hızlı gelişme OKA'ların kullanımını arttırmıştır. OKA'ların tarihsel gelişimi Şekil 2.2'de gösterilmektedir.

1.Dönem		2.Dönem		3.Dönem		4.Dönem	
1950	1970	1970	1980	1990	2000	2010	2020
*		*		*		*	
Yüksek maliyetli		Sensör teknolojisi gelişti.		Yapay Zeka		Düşük Maliyet	

Şekil 2.2. OKA'ların tarihi (Wu ve Wu, 2017)

1. Dönem: 1950'li yıllardan 1970'li yıllarını kapsayan birinci dönemde OKA'lar yeni tasarlanıp pazara girmiştir. Pazarda kendine çok yer bulamamıştır. Maliyetleri yüksek ve güvenilirlikleri azdır.

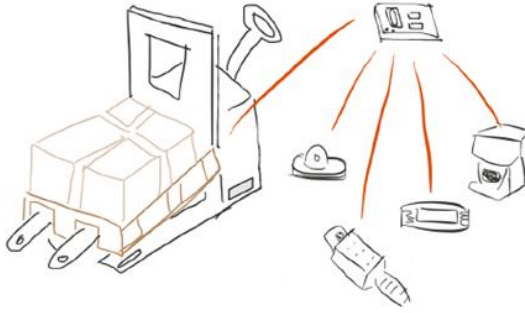
2. Dönem: 1970 ile 1990'lar arası yılları kapsayan bu dönemde OKA'lar otomotiv sektöründe kullanılmaya başlanmıştır. Sensörlerdeki gelişme 1970'lerden itibaren hızla artmıştır. Mikro denetleyiciler daha güvenilir hale gelmişlerdir.

3. Dönem: Artık OKA standartları oluşmuş ve birçok firma sektöre girmiştir. Temassız sensörler ve elektronik navigasyon daha çok kullanılmaya başlanmıştır. Yapay zeka uygulamaları kullanılmıştır.

4. Dönem: Birçok Asya firması piyasaya girmiştir ve OKA maliyetleri düşmeye devam etmektedir.

2.2. OKA Sisteminin Bileşenleri

OKA'lar genellikle yatay yönde malzeme taşımaktadırlar. Önceden belirlenmiş rotasında belirlenmiş görevleri gerçekleştirmektedir. Çok çeşitli türleri vardır. Bununla birlikte kendisini oluşturan bileşenlerin ortak özellikleri bulunmaktadır. OKA sistem bileşenleri Şekil 2.3'te gösterilmektedir.



Şekil 2.3. OKA sistemi bileşenleri (Turesson, 2016)

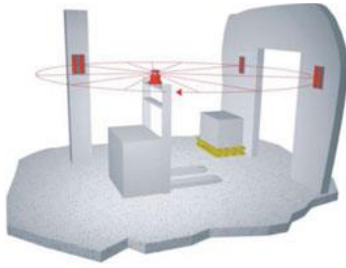
OKA aşağıdaki bileşenlerden oluşmaktadır (Turesson, 2016).

1. Tahrik sistemini oluşturan elektrik motorları
2. Tüm OKA'ya enerji sağlayan batarya
3. OKA kumanda sistemi
4. OKA navigasyon sistemi
5. Güvenlik sistemi
6. OKA ile yükü bağlayan sistemler

7.OKA'nın dışarıdan komut alması ve programlanması

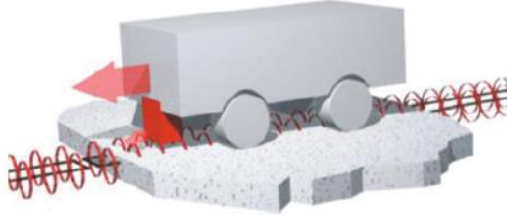
OKA'lar genellikle hangi yükleri taşımak için üretilmiş ise buna uygun motor seçilmiştir. Servo motor veya fırçasız DC motorlar sıklıkla kullanılır. Özellikle bataryanın doğru akım sunması DC motorların kullanılmasını arttırmıştır.

OKA yol bulma sistemi için farklı yöntemler mevcuttur. Kamera ile kontrol bu yöntemlerden biridir. Kamera kontrast farkını takip etmektedir. Bu yöntemin bir türevidir de yere barkod filmi yapıştırılmasıdır. Barkod filmini korumak boyadan daha zordur. Manyetik bant ile yol bulmada ise, OKA rotasına manyetik bant yapıştırılmaktadır. OKA'ya manyetik sensor takılması ile OKA'nın rotası belirlenir. GPS sisteminde ise uydulardan elde edilen bilgilerden yararlanılmaktadır. GPS navigasyonu, uydu sinyallerinin bloke edilmesi, karıştırılması ve çok yollu olması konumun doğruluğunu azalttığı için iç mekân kullanımı için benimsenmemiştir. LIDAR, mesafeleri ölçmek için laser pulsaları kullanılmaktadır. Bu yöntem Eşzamanlı Yerelleştirme ve Haritalama işlemi yapmaktadır. OKA'da bir bölgenin haritası oluşturulur ve etrafta dolaşırken haritadaki konumu belirlenir (Moshayedi ve diğerleri, 2019). Lazer navigasyon yöntemi Şekil 2.4'te gösterilmektedir.



Şekil 2.4. Lazer navigasyon yöntemi (Ullrich, 2015:105)

İndüktif kablo, araçların zeminde yerleştirilen indüktif hat ve kontrol noktalarını takip ederek konum belirlediği sistemlerdir. Esnekliğin azalması ve yüksek yatırım maliyeti bu yöntemin kullanımını sınırlamaktadır. Şekil 2.5'te yerden enerji verilmesi ile OKA'nın yönlendirilmesi gösterilmektedir.



Şekil 2.5. Yerden enerji verilmesi ile OKA'nın yönlendirilmesi (Ullrich, 2015, s. 102)

OKA'larda güvenlik sistemi; acil durdurma butonu, alan tarayıcı, güvenlik tamponları, çeşitli mekanik swicthler, OKA kornasından ve OKA rotasında oluşturulan sesli veya ışıklı uyarıcılardan oluşmaktadır.

OKA sistemlerinde önemli bir konu da OKA sistemlerinin enerji yönetimidir. Bataryanın hangi methodla şarj olmasının belirlenmesi gerekmektedir. Günümüzde var olan stratejiler aşağıdaki gibi sıralanmaktadır (Moshayedi ve diğerleri, 2019):

1. Manuel şarj
2. Otomatik hızlı şarj
3. Yerden sürekli şarj verilmesi

Öncelikle OKA'nın harcadığı enerji tespit edilmelidir: OKA'nın her çevrimde ne kadar enerji harcadığı eğer parkur mevcutsa OKA'nın ilgili turda çalıştırılması ile bulunabilir. Parkurun henüz oluşturulmadığı durumlarda yakın tahminler elde etmek gerekmektedir. OKA'nın özelliklerine (motor, tekerlek boyutu, dönüşüm oranı) ve taşıdığı yükün özelliğine göre harcanan enerji değişmektedir.

2.3. OKA Kontrol Seviyeleri

OKA'ların kontrolü hakkında üç düzeyde karar verilmesi gerekmektedir. Bunlar; stratejik düzey, taktiksel düzey ve operasyonel düzeydir.

1. Stratejik Düzey: Uzun dönemli kısıtlara, yerleşim kısıtlarına ve taşınan parçalara karar verilir.
2. Taktiksel Düzey: OKA sayısı belirlenmesi, şarj yöntemi gibi konular taktiksel seviyedeki kararlardır.

3. Operasyonel Düzey: OKA sıralama, OKA rotalama, çakışma önleme gibi konular bu seviyededir. Üç seviyede alınan kararlar diğer seviyedeki kararları etkilemektedir. Genellikle iç içe geçmiş yaklaşımlarla bazı seviyelerde kabuller yapılarak diğer seviyeler arasında bir en iyileme stratejisi izlenmektedir (Le-Anh ve diğerleri, 2006).

OKA sistemlerinin çizelgelenmesi sıklıkla çalışılan konulardan birisidir. Özellikle esnek imalat sistemlerinin çizelgelenmesinde sıklıkla ele alınmaktadır. Belirli bir zamanda en çok siparişe cevap verebilme gibi hedefler mevcuttur. OKA çizelgelenmesinin amacı, son teslim tarihleri, öncelik, vb. gibi belirli kısıtlamalar altında bir grup alım ve bırakma işine yönelik hedeflere ulaşmak için bir dizi OKA göndermektir. Sistem üretimini korurken OKA sayısını en aza indirmek veya tüm araçların toplam seyahat süresini en aza indirmek, konunun başlıca hedeflerini oluşturmaktadır (Aktürk ve Yılmaz, 1996).

Çizelgeleme kararı verildikten sonra, rotalama görevi, mevcut trafik durumuna göre her OKA için başlangıç noktasından hedefe uygun bir rotaya (Örneğin; en kısa mesafe yolu, en kısa zaman yolu veya minimum enerji yolu) ulaşmaktır. Özellikle birçok OKA'nın ve lojistik aracının kullanıldığı sistemlerde rotalama önemlidir. En kısa mesafe veya en kısa sürede OKA hattı tasarlanması için geliştirilen çeşitli algoritmalar mevcuttur. Bu algoritmaları ikiye ayırmak mümkündür. Statik rotalama için, en kısa yol hedefiyle bir i noktasından diğer j noktasına gidilecek ise en kısa yol bulunur ve her defasında bu yol kullanılır. Trafik koşulları ve diğer değişkenler göz ardı edilir. Dinamik rotalama için ise, sistem koşulları sabit i ve j noktaları arası gidişi, farklı yollarda gerçekleştirebilir. Bunun için gerçek zamanlı bir kontrol sistemi tüm yol durumlarını kontrol edip tıkanma gibi durumlarda OKA'ları farklı güzergâhlara yönlendirir. Endüstriyel hayatta OKA yolları çoğunlukla statik kabul edilir. OKA rotasına mühendislik, yerleşim, imalat, lojistik, iş güvenliği, bakım gibi birçok servisin katıldığı çalışma gruplarında karar verilir.

2.4. Kaynak Araştırması

Literatürde yer alan OKA kontrolü ile ilgili çalışmalardan bazıları aşağıdaki gibi özetlenebilir. İlk konu OKA'nın rotalanmasıdır ve OKA'nın A noktasından B noktasına gitmesini ele alır. Rota atamaları, OKA'ların sayısını etkiler. OKA trafiğinin yönetimi bir diğer önemli sorundur. Trafik yönetimi, OKA sistemini bloke edebilecek çarpışmaların ve kilitlenmelerin önlenmesini gerektirir. Trafik yönetim, merkezi bir kontrol sistemi

veya çeşitli yönlendirme yöntem ve teknolojileri ile yapılabilir. Filo büyüklüğü olarak adlandırılan üçüncü konu gerekli OKA sayısının belirlenmesidir. Sistem için gerekli OKA sayısı rota ve trafik faktörlerinin yanında başka faktörlerce de belirlenmektedir. Yükleme ve boşaltma istasyonlarının yeri bu faktörlere örnek olarak gösterilebilir. OKA'nın özellikleri, sistemin taşıma talepleri, gerekli OKA miktarını etkiler. Bu konu filo boyutlandırma olarak da bilinmektedir. Vis (2006)'e göre filo büyüklüğü, nakliye yükü talebi, zaman talebi, OKA kapasitesi, OKA hızı, yönlendirme, trafik yönetimi, OKA atama kriterleri, yükleme ve boşaltma istasyonlarının konumlarına göre yapılmalıdır. Son konu OKA kontrol problemi OKA atanması, programlama, yönlendirme ve park istasyonu seçimini içerir.

Bu problemlerin çözümü için matematiksel modellemeler, sezgisel yöntemler, ayrıştırma yöntemleri ve simülasyon yöntemine sıklıkla başvurulur.

Matematiksel yöntemler olarak Talbot (2003), OKA filo büyüklüğünü tahmin etmek için bir kuyruk modeli geliştirmiş; ancak model, hafif trafikte olması gereken OKA miktarından daha düşük çözümler üretmiştir. Matematiksel modeller genellikle diğer yöntemlerle birlikte de çokça kullanılmaktadır. Saidi-Mehrabad ve diğerleri (2015), OKA ile taşıma yapılan bir üretim tesisinde üretim süresini en aza indirmek ve aynı zamanda çarpışmaları önlemek için bir atölye çizelgeleme modelini karma tam sayılı programlama ve karınca kolonisi algoritması kullanarak çözmüştür. Modelinin hesaplama süresinin karınca kolonisi yöntemiyle azalacağı belirtilmiştir. Miyamoto ve diğerleri (2016), OKA sistemlerini atamak için tamsayı bir model kullanmıştır. Önce matematiksel çözüm üzerinde çalışılmış, ardından yerel arama ve rastgele arama yöntemleri test edilmiş ve yerel arama algoritmasının büyük ölçekli problem için daha iyi bir çözüm sağladığı raporlanmıştır.

Sezgisel yöntemler NP hard tipi problemlerin çözümünde sıklıkla kullanılmıştır. İnanç ve Şenaras (2019), OKA rotasını karınca kolonisi optimizasyonu ile analiz etmişlerdir ve ACO ile belirlenen rotanın kabul edilebilir bir süre içerisinde OKA verimliliği artırdığını göstermişlerdir. Bozorgi-Amiri ve diğerleri (2015), OKA hareketini en aza indirmek için bölünmüş dağıtım bir araç rotalama problemi önermiştir. Sezgisel ve matematiksel modelleri karşılaştırmışlardır. Ventura ve Lee (2003), yanıt süresini filo büyüklüğü

hesaplamasının bir faktörü olarak ele almışlardır. Tepki süresi, bir sonraki başlatmaya kadar olan boş seyahat süresidir. Tepki süresini en aza indirmeye çalışmışlardır. Katma değeri olmayan bu sürenin en aza indirilmesi ile toplam taşıma süresinin azaltıldığı gösterilmiştir. Ortalama yanıt süresini en aza indirmek için bir OKA sisteminde konumlandırılmış bekleme noktaları ile maksimum yanıt süresinin en aza indirilmesi Ventura ve diğerleri (2015) tarafından analiz edilmiştir. Tek bir OKA hattı analiz edilmiş ve matematiksel model ve genetik algoritma yaklaşımı ile karşılaştırılmıştır ve yazarlar hesaplama süresinin model boyutu ile arttığını ve GA'nın optimale yakın çözüm bulmada etkili olduğunu göstermişlerdir. Chaudhry ve diğerleri (2022), FMS makinaların ve OKA'ların eş zamanlı programlanması için Genetik Algoritma tabanlı bir çözüm geliştirmişlerdir. Çözüm kabul edilebilir doğrulukta makul bir zamanda gerçekleştirilmiştir. Moghaddam ve diğerleri (2012), belirsiz müşteri talebi ve dağıtım ortamları altındaki araç rotalama problemleri için gelişmiş bir parçacık sürüsü optimizasyon algoritması önermiştir. Yeni yöntemin çözüm kalitesini iyileştirdiği bildirilmiştir ve algoritmanın büyük ölçekli problemler için yeterli olduğu belirtilmiştir. Moghaddam ve diğerleri (2010) tarafından belirsiz talep altında sağlam (robust) araç yönlendirme optimizasyonu geliştirilmiştir. Model ile karşılanmayan talep azalmıştır. Fazlollahtabar (2016), OKA'ları ve makineleri kısa zaman dilimleri içinde atamak için bir programlayıcının kullanılmasını önermiş ve bu atama için sezgisel bir model geliştirmiştir. Önerilen çerçeve, OKA sevkiyatı için iki seviyeli hiyerarşik dinamik bir karar vermeyi ve minimum maliyet akışına dayalı bir sonraki istasyonun seçimi ile işleme programlayıcısını ve operasyon kontrolü için bir istasyon denetleyicisini içermektedir. Bulgular, ortalama akış süresini azaltarak sevk kuralının etkinliğini ortaya koymuştur. İleriki çalışmalarda yapay zekâ ve iki seviyeli programlamanın ınceleceđi belirtilmiştir. Lacomme ve diğerleri (2013), çevrimdışı (offline) olarak OKA ve makine çizelgelemeyi incelemiştir. Makine ve OKA'ları eş zamanlı olarak incelenmiştir. Ayrık bir grafiđe dayalı bir model oluşturulmuştur. Geliştirilen modelin çözümü için sezgisel bir algoritma önerilmiştir. Üretim süresini en aza indirilmesi amaçlanmıştır. Sonuçların karşılaştırılması, önerilen algoritmanın makineleri ve OKA'ları aynı anda programlamak için en uygun çözümleri sağlayabileceđini göstermiştir.

Simülasyon tekniđi OKA'larla ilgili problemlere ışık tutmak için sıklıkla başvuru olan bir yöntemdir. Valmiki ve diğerleri (2018), toplam seyahat süresini ve toplam maliyeti en

aza indirmeye dayalı olarak simülasyon tabanlı bir model geliştirmiştir. Simülasyon modeli toplam seyahat süresinin minimizasyonuna yönelik olarak OKA filo boyutunun hesaplanmasına dayanmaktadır. Simülasyon yönteminin, analitik yöntemle göre doğru bir filo boyutlandırma çözümü sağladığı bildirilmiştir. Başka bir yaklaşımda, Yifei ve diğerleri (2010), matematiksel modelle doğru sayıda OKA filo büyüklüğü atamanın zor olabileceğini ve tüm sistemi simüle etmenin zaman alabileceğini bildirmiştir. İlk aşamada gerekli OKA sayısını matematiksel model ile tahmin etmişlerdir ve çıktıyı simülasyon girdisi olarak kullandıkları için hesaplama süresini azaltmışlardır. Viharos diğerleri (2018), OKA'ları kontrol etme stratejisi için bir benzetim modeli önermişlerdir. Modelin amacı toplam operasyonel üretim süresini minimize etmektir. Chang ve diğerleri (2014), otomatik bir malzeme taşıma sisteminde çok amaçlı (minimum teslimat süresi ve maksimum parti teslimatı) araç filosu boyutlandırması için simülasyon optimizasyonu ve veri zarflama uygulamışlardır. Sayısal çalışma, önerilen yöntemin doğru çözümler sağladığını göstermiştir. Kim ve Park (2009), ortalama teslim süresini en aza indirmek için çeşitli araç atama kurallarını test etmiştir. Simülasyon tekniği ile kuralların sistem etkinliği üzerindeki etkisini göstermiştir. Şenaras (2019), OKA parametrelerini yüzey yanıt yöntemi ve simülasyon yöntemi ile analiz etmiştir. Çeşitli parametrelerin simülasyonundan sonra, gerekli OKA sayısını hesaplamıştır. Gerekli OKA sayısını en aza indirmek için optimal parametreler belirlenmiştir ve sonuçlar benzetim tekniği ile teyit edilmiştir.

Bir başka yöntem olan Yapay Sinir Ağları, tek bir OKA için en kısa rotayı belirlemek için kullanılmıştır (Soylu ve diğerleri, 2000). Çalışma minimum toplam boş seyahat süresini belirlemeyi amaçlamıştır. Hesaplama süresi için iyi bulgular rapor edilmiştir. Bununla birlikte, belirli sıralı modellerde, algoritma kabul edilebilir çözümler üretmiştir, ancak hiçbiri optimal veya optimale yakın değildir.

Ayrıştırma yöntemi NP hard problemlerin çözümüne katkı sağlamıştır ve OKA konusunda da kullanılmıştır. Nishi ve diğerleri (2007b), OKA'lar genelinde çarpışmaları veya kilitlenmeleri verimli bir şekilde çözmek için bir ayrıştırma yöntemi önermişlerdir. Nishi ve diğerleri (2011), eş zamanlı çizelgeleme ve çatışmasız yönlendirme problemlerini çözmek ve görev setlerinin toplam ağırlıklı gecikmesini en aza indirmek için iki seviyeli bir ayrıştırma algoritmasının kullanılmasını önermiştir. Lagrangian

gevşemesi, daha yüksek görev sayımlarında önemli bir sorun olan hesaplama süresini azaltmıştır. Nishi ve diğerleri (2007a), çoklu OKA'larda taşıma süresini en aza indirmek için hareket gecikme bozukluğuna sahip dağıtılmış bir yönlendirme yöntemi önermişlerdir, bu sayede OKA'lar arası çarpışma ve birbirini kitlemeyle sonuçlanan ceza puanlarını azaltmışlardır. Elde edilen bulgular, önerilen yöntemin gerçek ulaşım ortamlarında uygulanabileceğini göstermiştir.

Doğrusal programlama; hedefler, kısıtlamalar ve karar değişkenlerinden oluşmaktadır. En az bir karar değişkeni tamsayı ise, model karma tamsayılı programlama (MIP) haline gelmektedir ve matematiksel çözüme dayanan karma tamsayılı programlama optimum çözümü sağlamaktadır (Jia ve diğerleri 2022; Tonelli ve diğerleri 2013). Bununla birlikte matematiksel modeller, problemin karmaşıklığı arttıkça artan çözüm zamanı sebebiyle başka yöntemlerin kullanılmasını gerektirmektedir. Metasezgisel yöntemler bölgesel optimumlardan kaçınılması ve optimuma yakın sonuç elde edebilmesi sebebiyle sıklıkla kullanılmaktadır (Haupt ve diğerleri, 2004). Kendine has işlemcileri olan sezgisel algoritmalar doğada bulunan süreçlerden esinlenerek optimuma yakın sonuçlar elde etmektedirler. (Vob ve diğerleri, 1999). Bu yöntemlerden sıklıkla kullanılan Genetik Algoritma, 1975'te John Holland tarafından geliştirilmiştir. Sezgisel süreçlerin uygulanmasıyla başarılı sonuçlar elde edilmesi GA'nın birçok karmaşık problemde kullanılmasına sebep olmuştur ve sıklıkla baş vurulan bir yöntem olmuştur.

Bir yapay sinir ağı, beynin nörolojik olarak nasıl çalıştığının matematiksel bir modelidir. Beynin öğrenme sürecini taklit etmek için sinir hücreleri matematiksel olarak modellenmiştir. Katmanlarda yer alan nöron adı verilen birbirine bağlı öğelerle yapılandırılmıştır. Yapay nöronlar, yapay sinapslar aracılığıyla birbirine bağlanmaktadır ve öğrenme yoluyla bilgi edinilmektedir. İleri beslemeli sinir ağı, çok katmanlı sinir ağı olarak da adlandırılan bir yapay sinir ağı mimarisidir. İleri beslemeli bir ağ, bilgi veya sinyallerin girişten çıkışa yalnızca bir yönde gönderildiği bir ağıdır (Da Silva ve diğerleri 2017).

Yüzey Yanıt Yöntemi, Box ve arkadaşları tarafından 1950'li yıllarda geliştirilmiştir. Matematiksel modelin oluşturulması sonrası geliştirilen grafiksel perspektiften hareketle yöntem bu şekilde adlandırılmıştır. Deney tasarımıyla oluşturulan ampirik model,

istatistiksel olarak değerlendirilerek bağımlı değişkenin değişimi açıklanmaya çalışılmaktadır. Amaç önce incelenen sistemi tanımlamak ve sonuç olarak sistemin optimizasyonuna ulaşmaktır (Bezerra ve diğerleri, 2008).

Vis (2006), literatürdeki yöntemleri gözden geçirmiştir ve OKA sistemlerini tasarlamak ve kontrol etmek için geliştirilen modellerin belirli sorunların üstesinden geldiğini belirtmiştir. Mevcut yöntemlerin uzun hesaplama sürelerine sahip olduğunu ve özellikle tıkanıklık, kilitlenme gibi bazı problemlere odaklanıldığını bu sebeple yeni yaklaşımlara ihtiyaç duyulduğunu açıklamıştır. Matematiksel yöntemler yüksek hesaplama zamanı gerektirmesi araştırmaları diğer yöntemlere kaydırmıştır. Sezgisel yöntemlerin OKA konusunda kullanımı artarak devam etmektedir ve OKA kontrolü konusunda birkaç problemi aynı anda çözmeye elverişlidir. Simülasyon modelleri kullanımı artmasına rağmen; model oluşturma, tüm denemeleri gerçekleştirmek gibi sebeplerle çözüm süresini arttırmaktadır.

OKA sistemlerinin iyileştirilmesine ilişkin yapılan literatür araştırmasına göre, OKA sistemlerinin tasarımı ve kontrolü konusunda çeşitli çalışmalar yapılmış olsa da yapılan çalışmaların çoğunlukla bilgisayar destekli yüksek maliyetli sistemlere dayandığı görülmektedir. Düşük maliyetli sistemler için (PLC veya kart tabanlı ve şerit yönlendirmeli OKA'lar) optimum OKA filo büyüklüğünü hesaplayan ve sistem parametre değişikliklerinin filo büyüklüğü üzerindeki etkilerini inceleyen düşük maliyetli çözümlerden oluşan bir çalışma gözlenmemiştir. Bu çalışma, yapay zekâ tekniklerinin kullanımı ile akıllı düşük maliyetli sistemlerin kullanılması ile gerekli OKA sayısının minimizasyonuna imkân veren çözümler geliştirerek OKA sistemlerinin iyileştirilmesi konusunda literatüre katkı sağlamaktadır.

3. MATERYAL VE YÖNTEM

Bu çalışmada gereksinim duyulan en az OKA sayısını hesaplayan bir karma tam sayılı model geliştirilmiştir. Artan OKA hatları sürecinde oluşan zaman kısıtlarını gidermek için her OKA hat ve parkistasyonu kombinasyonuna göre gerekli OKA miktarını hesaplayan bir algoritma geliştirilmiş ve optimum GA uygulaması ile optimal gerekli OKA miktarı elde edilmiştir. Sistem parametrelerinin OKA sayısına etkisini belirlemek için deney tasarımı gerçekleştirilmiştir. Yüzey Yanıt Metodu ve YSA ile sistem parametre değişiminin OKA gereksinimi tahmin edilmiştir. En son olarak YSA ile OKA hat kombinasyonlarının park istasyonlarına göre gereksinim duyulan OKA miktarlarının tahmini yapılmıştır. Bu bölümde tez çalışmasında uygulanan yöntemler anlatılmıştır.

3.1. Tam Sayılı Programlama

Çalışmada matematiksel model 0-1 karma tamsayılı programlama ile ele alınmıştır. Doğrusal programlama uygulamalarının çok sayıda örneği olmasına rağmen, karar değişkenlerinin tamsayı olmayan değerler almasına izin veren bölünebilirlik varsayımı, daha çok uygulama yapılmasını sınırlamaktadır. Birçok uygulamalı problemlerde yer alan karar değişkenlerinin değeri sadece tamsayı olduğu zaman anlamlı olmaktadır. Bunun sebebi, girdi ve çıktıların bölünmezlik sorunu karar değişkenlerinin tamsayı değerli olmasını gerekli kılmaktadır. Tam sayılı programlamanın matematiksel modeli, karar değişkenlerinin tamsayı değerleri almak zorunda olduğu ek bir sınırlama ile doğrusal programlama modelidir (Hillier ve Lieberman, 2015/2019).

Tamsayılı programlama problemlerine geçmeden önce, bu tür problemlerin özelliklerine kısaca değinmekte yarar vardır. Amaç fonksiyonu en küçükleme yönünde olan genel bir doğrusal programlama modeli denklem 3.1 de amaç fonksiyonu ve denklem 3.2 ve denklem 3.3te kısıtlayıcılar olmak üzere aşağıdaki gibi yazılabilir (Öztürk, 2021).

Amaç fonksiyonu:

$$\text{En küçük } Z = \sum_{j=1}^n c_j x_j \quad (3.1)$$

Kısıtlayıcılar:

$$\sum_{j=1}^n a_{ij}x_j \geq b_i \quad i = 1, 2, \dots, k \quad (3.2)$$

ve

$$x_j \geq 0 \quad j = 1, 2, \dots, n \quad (3.3)$$

Modelde, x_j : deęişkenleri,

c_j : amaç katsayılarını,

b_i : kaynak kısıtlaması deęerlerini,

a_{ij} : teknoloji ya da kısıtlayıcı katsayılarını göstermektedir.

Tam sayı olması zorunlu olan karar deęişkenleri kısıtını eklendiğinde, tamsayı programlama problemlerini karar deęişkenlerinin aldıkları deęerlere göre üç sınıfta incelenebilir. Bunlar; arı tam sayılı programlama, karma tam sayılı programlama ve İkili (Binary) Tam Sayılı Programlamadır-

Arı tam sayılı programlama modelleri:

Bu tip modellerdeki bütün deęişkenlerin deęerlerinin tamsayı karakterinde olması koşulu bulunur. Bir tamsayı doğrusal programlama modeli denklem 3.4'te amaç fonksiyonu, denklem 3.5 ve denklem 3.6'da kısıtlar olmak üzere aşağıdaki gibi yazılabilir. Denklem 3.6'da verildięi gibi x_j , tamsayı deęerler almak zorunda olan deęişkenleri göstermektedir (Sezen, 2007, s. 115).

Amaç fonksiyonu:

$$\text{En küçük } Z = \sum_{j=1}^n c_j x_j \quad (3.4)$$

Kısıtlayıcılar:

$$\sum_{j=1}^n a_{ij}x_j \geq b_i \quad i = 1, 2, \dots, k \quad (3.5)$$

ve

$$x_j \geq 0 \text{ ve tamsayı } j = 1, 2, \dots, n \quad (3.6)$$

Karma tam sayılı programlama modelleri:

Bir modelde deęişkenlerin bir kısmının tamsayı türünde olma koşulu var ise de, bu deęişkenler dışındaki dięer deęişkenler tamsayı deęer alma koşulu ile sınırlandırılmazlar.

Diğer bir ifadeyle, n adet değişkeni olan bir modelde, bu değişkenlerden sadece k tanesinin tamsayı türünde olması istenmektedir ve $(n - k) \geq 1$ ise bu tip model “karma tam sayılı programlama modelleri” niteliği kazanmaktadır (Sezen, 2007, s. 115). Denklem 3.7’de amaç fonksiyonu, denklem 3.8;3.9 ve 3.10 kısıtlayıcılar gösterilmektedir.

Amaç fonksiyonu:

$$\text{En küçük } Z = \sum_{j=1}^n c_j x_j + m_j y_j \quad (3.7)$$

Kısıtlayıcılar:

$$\sum_{j=1}^n a_{ij} x_j + d_{ij} y_j \geq b_i \quad i = 1, 2, \dots, k \quad (3.8)$$

$$y_j \geq 0 \quad (3.9)$$

$$x_j \geq 0 \text{ ve tamsayı} \quad j = 1, 2, \dots, n \quad (3.10)$$

Burada;

y_j : tamsayı değerler alması zorunlu olmayan değişkenleri,

x_j : tamsayı değerler almak zorunda olan değişkenleri,

m_j : y_j değişkenlerin amaç katsayılarını,

d_{ij} : y_j değişkenler ile ilgili teknoloji katsayılarını

göstermektedir.

0 - 1 tam sayılı programlama modelleri:

Bazı karar durumlarında tek olası iki seçenek evet veya hayırdır. Örneğin belirli yerleşim yerine tesis kurulmalı mıdır? Belirli bir proje kabul edilmeli midir? Bu tür problemlerde sadece iki seçeneğin olduğu bu tür problemlerde, örneğin 0 ve 1 gibi iki değere sınırlanan karar değişkenleriyle böyle kararlar temsil edilebilir. Böylece j inci evet veya hayır kararı aşağıdaki gibi x_j ile gösterilebilir.

$$x_j = \begin{cases} 1 & \text{eğer } j \text{ inci karar evet ise} \\ 0 & \text{eğer } j \text{ inci karar hayır ise} \end{cases}$$

Bu tür değişkenler 0-1 değişkenleri veya ikili değişken olarak adlandırılmaktadır. Sadece ikili değişkenleri içeren tam sayılı programlama 0-1 tam sayılı programlama problemleri

veya ikili tam sayılı programlama problemleri olarak. Diğer bir ifadeyle, modelde yer alan değişkenlere 0 ve 1 değerleri dışında bir değer almamaları sınırlandırılması getirilmektedir (Sezen, 2007, s. 115). Denklem 3.11’de amaç fonksiyonu, denklem 3.12; denklem 3.13’te kısıtlayıcılar gösterilmektedir.

Amaç fonksiyonu:

$$\text{En küçük } Z = \sum_{j=1}^n c_j x_j \quad (3.11)$$

Kısıtlayıcılar:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad i = 1, 2, \dots, k \quad (3.12)$$

ve

$$x_j = 0 \text{ veya } 1 \quad j = 1, 2, \dots, n \quad (3.13)$$

3.2. Genetik Algoritma

Bu çalışmada OKA hatları için gerekli minimum OKA miktarının belirlenmesi için Genetik Algoritma kullanılmıştır. Günümüzde problemlerin matematiksel modelinin kurulması ve kesin sonuçlarının elde edilmesinin mümkün olmadığı durumlarla sıklıkla karşılaşmaktadır. Bazen model kurma, bazen de kurulan modelin hesaplanması da oldukça zaman alabilmektedir. Genetik Algoritma temel olarak optimal bir çözüm sağlamak için sezgisel bir problem çözme stratejisi olarak kullanılır.

Yöntemin sıklıkla uygulanmasında yöntemin basitliği ve esnekliğinin rolü vardır. Genetik algoritmanın operatörleri mutasyon, seçim ve çaprazlamadır. GA popülasyon olarak adlandırılan çözümler kümesi üzerinde çalışır. Başlangıçta rastlantısal oluşturulan bu popülasyonda bulunan her bir kromozom bir çözüm adayıdır ve bu kromozomların uygunluklarının belirlenmesi için uygunluk değeri tanımlanmıştır. Bir kromozomun uygunluk değeri ne kadar yüksekse o kromozomun popülasyonda kalma ihtimali o kadar yüksektir. Popülasyondaki kromozomlar çaprazlama işlemine girerek 2 kromozom arasındaki bazı bölgeler değiştirilir ve yeni bireyler oluşur. Yerel optimumdan kaçınmak için kromozomun bazı genleri rastlantısal olarak değiştirilir. Bu sayede bölgesel optimumlardan kaçınma sağlanır.

Genetik algoritmada her kromozom bir çözüm adayıdır. Kromozomlar genlerden oluşur. Gen bir bireyin belirli bir özelliğini oluşturan, kendi başına anlamlı en küçük genetik yapıdır.

GA doğadaki evrimi modelleyerek arama gerçekleştirir. Evrim çeşitliliklerin yönetilmesi ile o türe ait en uygun özellikler oluşur. Çeşitlilik, üreme ve mutasyon ile oluşur. GA da ise bu yapıya karşılık gelen operatörler bulunur; bunlar çoğalma, çaprazlama, mutasyondur.

Genetik Algoritma İşlemleri:

Genetik algoritma arama süresince evrimsel süreçlerden esinlenmiş kendine özgü bir çok işlemci kullanır (Katoch ve diğerleri 2021).

Genetik algoritma bir optimizasyon yöntemidir, aşağıdaki şekilde ifade edilebilir:

$$f: X \rightarrow R$$

$$f(x_0) = \max_{x \in X} f(x)$$

olmak üzere S bir dizi kümesi olsun. Bir c fonksiyonu aşağıdaki şekilde ifade edilebilir.

$$C: X \rightarrow S$$

$$x \rightarrow c(x)$$

kodlama fonksiyonu olarak adlandırılır (Bodenhofer, 2003).

Genetik algoritma kod uzayı ve çözüm uzayında çalışır. Her evrimsel algoritma kodlama fonksiyonunun belirlenmesi ile başlar (Eiben ve Smith, 2003). Kodlama yöntemi, olurlu çözümleri ve oluşacak tüm kromozomlara uygun olmalıdır.

Elektronik ve programcılıkta sıklıkla başvurulmuş ikili kodlamada aleller 0 veya 1 değerini alıp genetik algoritma başlangıçta bu kodlama yöntemini kullanmıştır. Çözüm uzayındaki değerler bu kodlamaya göre tasarlanması gerekmektedir (Katoch ve diğerleri, 2021). Bu çalışmada ikili kod sistemi zaman tabanlı olarak OKA hat kombinasyonları hesabındaki algoritmada kullanılmıştır. Bu yöntem yapılı en büyük eleştiri özellikle değer sayısal değerinin 1 artması için birçok alellin değer değişmesi gerektirir. Bu eleştirilere karşı gri kodlama ve tamsayı kodlama metotları da kullanılmaktadır (Mitchell, 1999; Eiben ve Smith, 2003).

0101010 010101 0101010 gibi bir ifadede 3 gen 20'ye yakın alel kullanılmıştır.

Genetik Algoritmada doğal seçilime yani en uygun kromozomların devamına imkân veren operatör, seçim operatördür. En uygun kromozomlar baskın (dominant) hale gelmektedir (Karaboğa, 2014, s. 78). Yalnız bu aşamada da rastgelelilik ve olasılık mevcuttur ve bu sayede bölgesel optimumlardan kaçınma imkânı sağlanır. Birçok seçim metodu olsa da temel yöntemler kısaca ele alınacaktır.

Rulet Çemberi Yöntemi:

En sık kullanılan seçim yöntemidir. Popülasyondaki kromozomların uygunluk fonksiyon değerleri hesaplanır. Popülasyondaki kromozomların toplam uygunluk değerinin kromozomların uygunluk değerine oranı ile kromozomların olasılık değeri elde edilir. Daha sonra bu değerlerden birikimli olasılık değerleri hesaplanır. Bu seçim sonucunda her kromozom 0 ile 1 arasında değerler alır ve popülasyon birey sayısı kadar rastlantısal sayı (0-1) oluşturulması ile bu sayılara karşılık gelen kromozomlar bir sonraki kuşağa aktarılır (Eiben ve Smith, 2003; Gen ve Cheng, 1997).

Sıralamalı Seçim Yöntemi:

Rulet Çemberi yönteminde eğer bir kromozom popülasyon içinde çok baskın ise çözüm bölgesel optimumda takılabilir, bu sorunu önlemek için sıralamalı seçim yöntemi geliştirilmiştir. Sıralama ve seçilme olasılığı arası ilişki farklı şekillerde olabilir: en çok kullanılan ise doğrusal veya üstel azalan şeklinde olasılık toplamı 1 olarak tasarlanır (Eiben ve Smith, 2003).

Turnuva Seçim Yöntemi:

Popülasyondan belirli miktar birey alınıp kendi aralarında en uygun birey bir sonraki popülasyona devam eder. Bu işlem popülasyon sayısı yeniden oluşuncaya kadar devam eder.

Elitizm:

Popülasyonun en uygun bireylerinin bir bölümünün hiçbir değişikliğe uğramadan (çaprazlama ve mutasyona) bir sonraki nesile aktarılmasıdır. Seçilen kromozomlar eşleştirme havuzuna alınarak çaprazlama işlemine tabi olurlar.

Eşleştirme havuzunda bulunan rastgele seçilen 2 kromozom belirli yerlerinden koparılıp kendi aralarında genlerin değişmesi sonucu ile gerçekleştirilir. İşlem sonucunda eski 2 kromozom yeni 2 kromozoma dönüştüğü için popülasyon büyüklüğü sabit kalır. En çok kullanılan çaprazlama ve en basit yöntem olarak tek noktadan çaprazlama kullanılır. Aşağıda görülen şekilde yeni 2 kromozom oluşur.

01010101 00110011 → 01010011 00110101

Sadece genlerin kromozomlar arasında değiştirilmesi tüm çözüm uzayının araştırılması için yeterli olmayabilir ve yerel optimum ile sonuçlanabilir. Bu sebeple kromozomların içindeki genler küçük bir olasılıkla mutasyona yani değişime uğrar.

01010101 → 01111111

Algoritmanın sonlanması için aşağıdaki yöntemlerden biri veya birkaçı kullanılabilir:

1. Belirlenen yeni kuşak sayısına ulaşılmışsa
 2. İstenen eşik değerine ulaşılmışsa
 3. Elde edilen değerlerde belirli bir nesil sonra daha iyiye gidilmemesi durumunda
- Bununla birlikte Genetik algoritmanın tek sefer çalıştırılmaması da önemlidir.

Genetik algoritmanın performansını üç faktör etki eder:

- Popülasyon büyüklüğü: Popülasyonun çok büyük belirlenmesi işlem süresini arttırırken, küçük popülasyon ise çözüm uzayının yeterli araştırılmamasına sebebiyet verir. Probleme göre değişkenlik gösterebilir (Taşkın ve Emel, 2009, s. 72).
- Çaprazlama oranı: $P(c)$ seçiminde düşüklük arama hızını azaltırken, yüksek seçilmesi ise oluşan bireylerin iyi araştırılmadan değiştirilmesi ile GA'nın performansını düşürür. Önerilen değerler 0,75 ile 0,95 arasında değişir.
- Mutasyon oranı: $P(m)$ genellikle 0,05 civarında tutulur ve çok arttırılması çözümün optimuma ulaşmasını engeller. Diğer taraftan aşırı küçük tutulması iraksamayı sınırlandıracağından bölgesel optimumda çözümün kalmasına sebebiyet verir.

3.3. Yapay Sinir Ağları

Çalışmada Yapay Sinir Ağları ile iki farklı uygulama gerçekleştirilmiştir. İlk uygulama OKA hatlarına atanan park istasyonuna göre gerekli OKA miktarının tahminidir. İkinci

uygulama ise sistem parametrelerinin deęişiminin OKA filo sayısına etkisinin tahminidir.

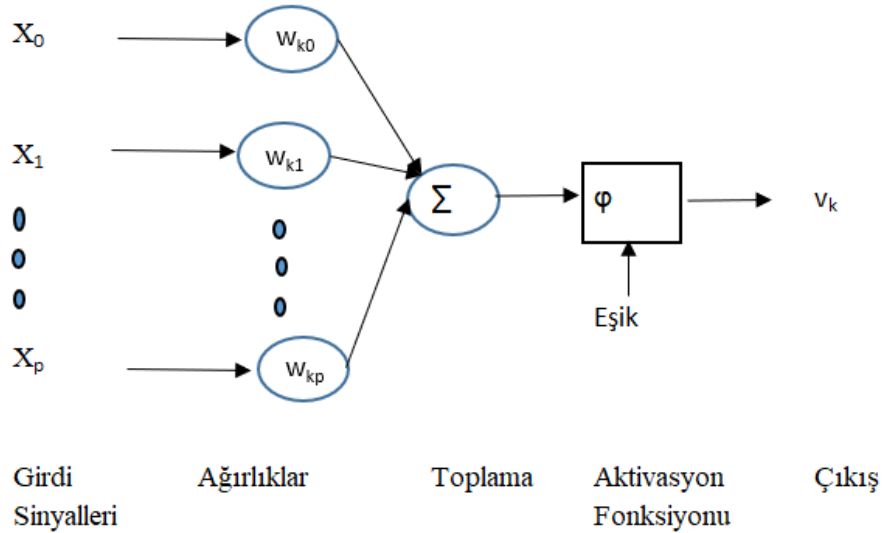
Biyolojik sinir aęlarının matematiksel modellenmesi sonucuyla yapay sinir aęları geliştirilmiştir. Bir nöron, bazı elektriksel ve kimyasal deęişimlerin yardımıyla bir nörondan dięerine bilgi işleyen özel bir biyolojik hücredir. Yapay Nöron ise temel olarak biyolojik nöronun mühendislik yaklaşımıdır. Birçok girişi ve bir çıkışı olan bir yapıya sahiptir. YSA, birbiriyle bağlantılı ve aynı zamanda katmanlı çok sayıda basit işleme elemanından oluşur.

Yapay sinir aęlarının yapısı ve işleyişi:

Biyolojik beyinde olduęu gibi yapay sinir aęları, birbirine paralel olarak bağlanmış, hiyerarşik olarak çalışabilen yapay sinir hücrelerinden oluşur.

Yapay sinir hücresi:

Yapay Sinir Hücrelerine nöron denir ve nöronların birleşmesi ile yapay sinir aęları oluşmaktadır. Yapay sinir hücresi fonksiyonları Şekil 3.1’te gösterilmektedir.



Şekil 3.1. Yapay sinir hücresi fonksiyonları (Dongare ve dięerleri, 2012: 191)

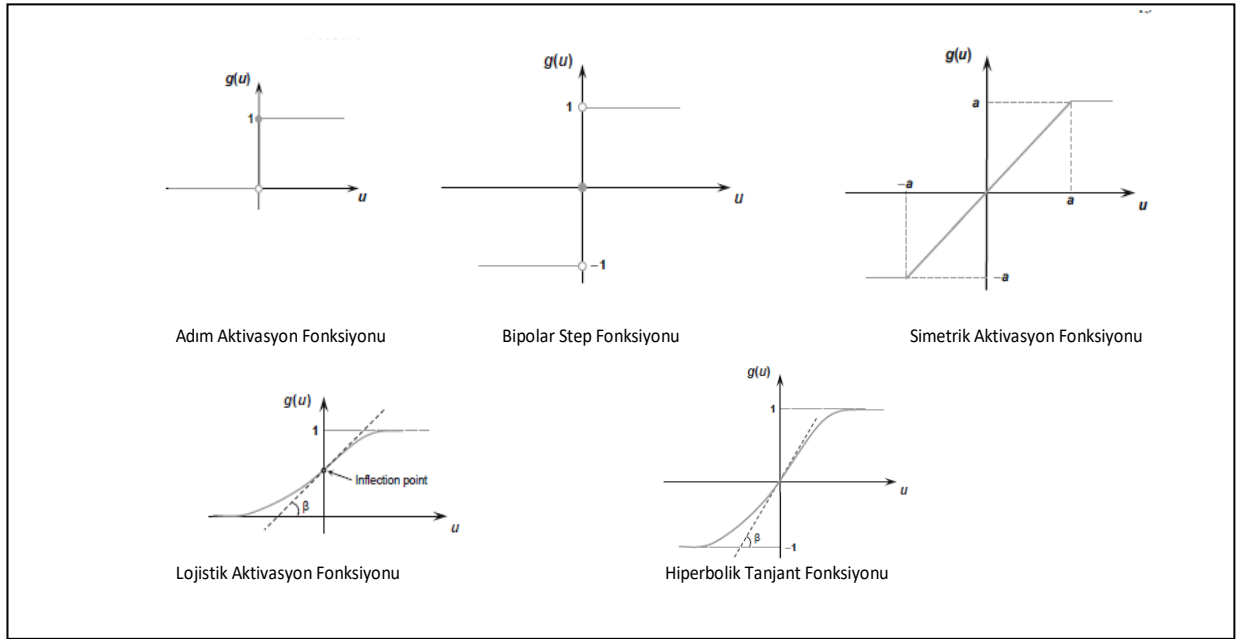
Nöronlar ağ üzerinde bağlantıları ile birbirlerine etki ederler. Her bağlantının kendine özgü bir ağırlığı vardır. Bu ağırlıklar öğrenme etabında oluşturulur aynı zamanda YSA'nın hafızalarını oluştururlar. Ağırlıklar, YSA'nın performansını belirlemektedir.

Denklem 3.14'te gösterildiği gibi nörona taşınan bilgiler, kendi ağırlıklar ile çarpılır ve net girdi, çarpılan değerlerin toplamı olarak ifade edilmektedir.

$$net = w_1x_1 + w_2x_2 + \dots + w_nx_n + x_0b = \sum_{i=1}^n w_i x_i + x_0b \quad (3.14)$$

$$y = f(net) \quad (3.15)$$

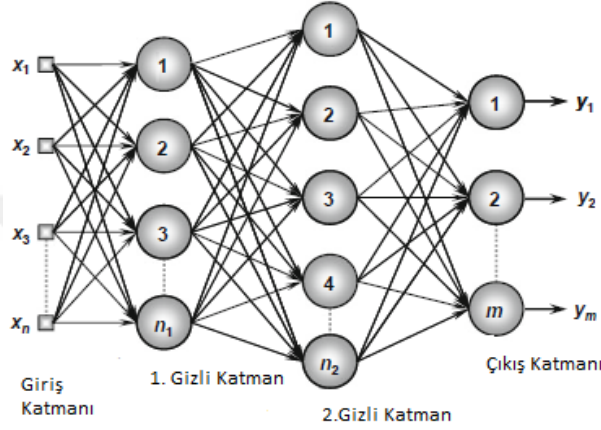
Gelen girdinin ne önemle değerlendirileceği ve hangi işaretle değerlendirileceği ağırlığına bağlıdır ve ağırlık pozitif, negatif ve sıfır olabilir. Ağırlık değişken veya sabit olabilir. Girdiler ve ağırlıklar daha sonra toplama fonksiyonu ile birleştirilerek net girdi elde edilir. Toplam fonksiyonu ile elde ettiği net girdiden nöronun çıktısını oluşturma fonksiyonuna aktivasyon denir ve denklem 3.15'te gösterilmektedir. Çeşitli aktivasyon fonksiyonu bulunmaktadır. Sigmoid, sinüs, lineer, hiperbolik fonksiyonlar sıklıkla kullanılan yöntemlerdir. Şekil 3.2 aktivasyon fonksiyonları ve ilgili matematiksel yazılımları gösterilmektedir. Aktivasyon ile nöron tek bir çıktı değeri oluşturur ve başka bir nörona iletilir.



Şekil 3.2. Aktivasyon fonksiyonu (Da Silva ve diğerleri, 2017)

Yapay sinir ağı yapıları:

YSA'ları nöronların bir araya gelmesiyle oluşur. Tek katmanlı ve çok katmanlı olarak ayrılırlar. Girdi katmanı, girdi bilgileri ara katmana iletir. Bu bilgilerin normalize edilmiş olması YSA performansını artırır. Ara katmanda birden çok katman bulunabilir. En son katman ise çıktı katmanıdır ve oluşturulan bilgi dışarıya aktarılmaktadır. Şekil 3.3.'te YSA'ların yapısı gösterilmektedir.



Şekil 3.3. YSA'ların yapısı (Da Silva ve diğerleri, 2017: 23)

YSA katman arası ilişkiye göre ileri ve geri beslemeli olarak iki grupta incelenmektedir. İleri beslemeli YSA'da her bir katmandaki nöronlar sadece bir önceki katmanın nöronları tarafından beslenirler. Geri beslemeli YSA'da ise en az bir nöron kendinden sonraki bir katmandan beslenmektedir.

Yapay sinir ağlarının eğitilmesi:

YSA'da öğrenme bir süreçtir. Öğrenme hızı, YSA'nın eğitim sürecini tamamlama hızı olarak açıklanır. Ağırlık değiştirme metodu seçilen öğrenme ile belirlenir. Öğrenme, denetimli öğrenme, denetimsiz öğrenme ve güçlendirilmiş öğrenme olarak üçe ayrılır. Denetimli öğrenmede, bir sistemin doğru şekilde çalışıp çalışmadığı veya istenen bir yanıtı belirtmek veya bir sistemin yanıtlarının kabul edilebilirliğini doğrulamak veya sistem performansındaki hata miktarını belirtmek için bir öğretmen mevcuttur. Girdiler ve bu girdilerden elde edilen çıktılar YSA'ya tanıtılır. Denetimsiz öğrenmede öğrenmenin, farklı örnek verileri veya ortamı inceleyen sistem tarafından, sezgisel olarak gerçekleştirilmesiyle yapılır. Bu metotta bir çıktı, girdi içindeki model kümelerine yanıt vermek üzere eğitilir. Sistem popülasyondaki farklılıkları tespit edip buna göre girdileri

ayırması beklenmektedir. Bir örneklemeden alıp sınıflandırılabilir bir durum bulunmamaktadır. Güçlendirilmiş öğrenmede ise bu durumun ortasıdır. Öğrenme, çevreden alınan geri dönüşlerle teyit edilir ve öğrenme süreci bir denge sağlanıncaya kadar devam etmektedir (Dongare, 2012; Öztemel, 2003).

3.4. Yüzey Yanıt Metodu

Sistem parametrelerinin değişiminin gerekli OKA sayısının tahmininde Yüzey Yanıt Metodu kullanılmıştır. OKA Deney tasarımında önemli bir konu olan Yüzey Yanıt Metodu (YYM), ilgilenilen bir yanıtın çeşitli değişkenlerden etkilendiği ve amacın bu yanıtı optimize etmek olduğudur (Montgomery, 2005).

$$y = f(x_1, x_2) + e \quad (3.11)$$

Denklem 3.11'de gösterildiği gibi bir model ele aldığı kabul edilsin. Ele alınan model iki faktörlüdür. X_1 ve X_2 değişkenler ve y onlara bağlı olarak değer almaktadır. e hata terimidir. Ortalaması 0 ve varyansı s^2 normal olarak dağıldığı kabul edilmektedir.

Oysa gerçek hayatta çoğu zaman denklem 3.11 gibi fonksiyon bilinmemektedir. Araştırmacı yaptığı denemelerle f fonksiyonunu basit bir modelden başlayarak oluşturmaya çalışır. Öncelikle polinom yapısında bir yaklaşım seçilir ve ilk olarak lineer bir model denenir (Lin ve Peterson, 2006). Denklem 3.12'de bu model ifade edilmiştir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \quad (3.12)$$

Yapılan deneylerden elde edilen veriler denklem 3.12 ile açıklanamazsa daha üst seviyeden bir polinom yapısındaki modele geçilir. Buradaki yaklaşım, Taylor serisi açılımıyla ifade edilmeye çalışılmasıdır. Denklem 3.13'te gösterildiği gibi ikinci dereceden bir model yazılabilir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \quad (3.13)$$

YYM'da genellikle denklem 3.13'te gösterilen modeller kullanılmaktadır. Faktör seviyeleri birbirinden bağımsız olmalıdır. Sabit katsayı değerleri En küçük Kareler yöntemi ile tahmin edilir. YYM analizi geliştirilen denklem 3.13 üzerine geliştirilir ve optimum değerlere ulaşılır.

YYM’de ilk hedef optimum değerleri bulmaktır. Diğer bir önemli nokta ise yanıt değerinin diğer faktörlere göre nasıl değiştiğini öngörmektir (Aslan ve Cebeci, 2007).

Yüzey yanıt grafikleri de bu konuda yardımcı olmaktadır.

1.seviyede yüzey yanıt için i yapılan deney sayısı, q bağımsız değişken sayısı, ε deneysel hata olmak üzere denklem 3.14 yazılabilir.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_q x_{iq} + \varepsilon_i \quad (i = 1, 2, \dots, N) \quad (3.14)$$

Denklem 3.14 bir çoklu regresyon modelidir. $\mathbf{y}=\mathbf{XB}+\mathbf{e}$ olarak matris formunda yazılabilir. y gözlem değeri, x bağımsız değişken seviyeleri, ε ise rastgele hata olarak tanımlanır.

Modelde eğrilik bulunması durumunda ikinci dereceden bir model uygulanmalıdır. İkinci derece model, birinci derece modelin tüm terimlerini içerir, ilave olarak quadratik terimlerde modelde bulunur. Pratikte 3. Seviyeden yukarı YYM uygulanmaz.

2. Seviye modelleri için geliştirilmiş tasarımlar bulunmakla birlikte genellikle CCD (central composite design) ve ortogonal central composite design kullanılmaktadır. İyi bir tasarım veriler ile uyumlu olmalı, maliyet olarak etkin olmalı (gereksiz deneye gerek duymamalı) ve tasarım yapılan aralıkta iyi bir varyans dağılımı oluşturmalıdır (Anderson ve diğerleri, 2009).

Modelin geçerliliğinin sınanması önemli bir etaptır. Oluşturulan YYM modelinin uygunluğunu kontrol etmek için önce hataların ortalamasının 0 ve s^2 standart sapmada olduğunun kontrol edilmesi gerekir. Residuel normal dağılım tablosu bu konuda fikir vermektedir.

ANOVA, modeldeki tüm ortalamaları aynı anda karşılaştırmaya imkân veren bir varyans analiz testidir (Mathews, 2004, s. 147). ANOVA çıktı verilerinin yorumlanmasını sağlar. Temel olarak, bir deneydeki toplam varyasyonun, kontrol edilen faktörlere ve üretilen hatalara atfedilebilen bileşenlere bölünmesinden oluşur. Modellerin istatistiksel önemi, ANOVA'nın p değerleri ve F değerleri ile değerlendirilir. ANOVA tablosunda p değeri, bir çalışmada gözlemlenen sonuçların tesadüfen meydana gelme olasılığıdır (0 ile 1 arasında değişir) (Bouzid ve diğerleri, 2015).

$p > 0.05$ ise parametre anlamlı değildir;

$p < 0.05$ ise parametre anlamlıdır.

Modelin yeterliliğini kontrol etmek için, hesaplanan F değerlerinin F-tablosundan elde edilen değerlerden büyük olması gerekir. Bu iki değer sonrası modelin istatistiksel olarak anlamlı olduğu söylenebilir. Modelin uyum ölçüsü iyiliği olarak R^2 değerleri kullanılır. R^2 [0,1] aralığındadır. R^2 değeri ne kadar 1'e yakın olursa, modelin açıklama gücü o kadar yüksektir. Genel olarak, R^2 , regresyon denklemiyle açıklanan y'nin y etrafındaki değişiminin yüzdesini ölçer. Bununla birlikte, modele bir değişken eklemek, bu değişken istatistiksel olarak anlamlı olsun ya da olmasın, her zaman R^2 'yi artırır. Bu nedenle, bazı araştırmacılar R^2 'yi kullanmak yerine düzeltilmiş R^2 kullanır. (Bouزيد ve diğerleri, 2015; Montgomery, 2005, s. 475).

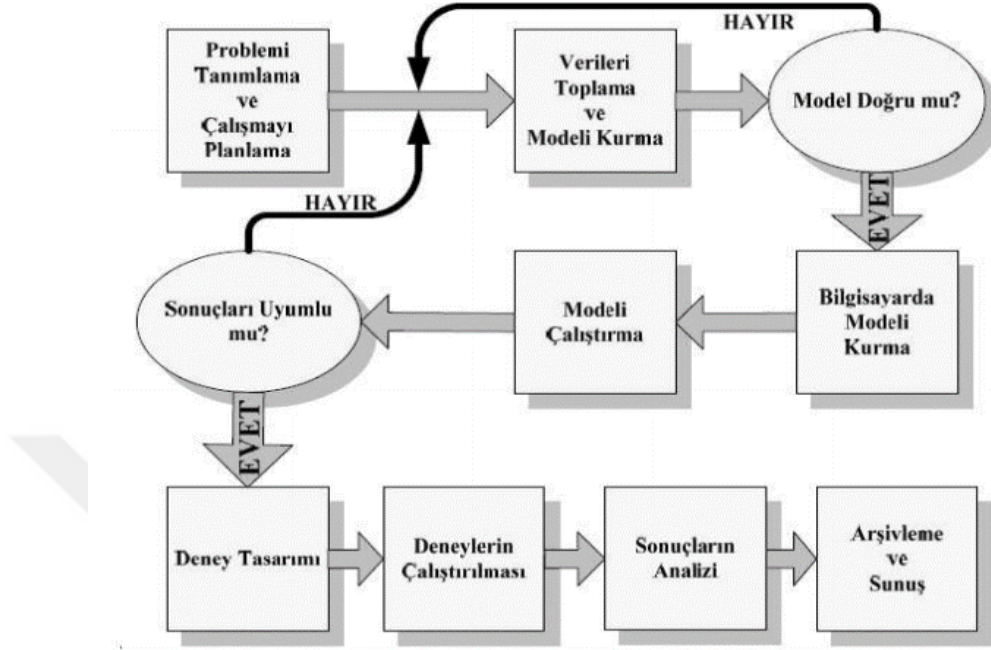
3.5. Kesikli Sistem Simülasyonu

Çalışmada elde edilen çözümlerin imalatta OKA kaynaklı bir duruş yaşanmadığının kontrolü amacıyla Simülasyon yöntemi kullanılmıştır. Bir sistemin zaman içinde gelişen davranışı, bir simülasyon modeli geliştirilerek incelenir. Bu model genellikle sistemin işleyişine ilişkin bir dizi varsayım şeklini alır. Bu varsayımlar, sistemin varlıkları veya ilgilenilen nesnelere arasındaki matematiksel, mantıksal ve sembolik ilişkilerde ifade edilir. Geliştirildikten ve doğrulandıktan sonra, gerçek dünya sistemi hakkında çok çeşitli "eğer öyleyse (What if)?" sorularını araştırmak için bir model kullanılabilir (Banks ve diğerleri, 2005, s. 3).

Simülasyon olay listesi, bir simülasyon çalışması sırasında meydana gelen farklı şeyleri takip etmenin bir yoludur (Law ve Kelton, 2000). Simülasyon çalışması sırasında meydana gelen ve sistemin durumunu etkileyebilecek her şey bir olay olarak tanımlanır. Basit bir simülasyondaki tipik olaylar, kuyruğa varlık varışlarını, varlıklar için hizmet sürelerinin başlangıcını ve varlıklar için hizmet sürelerinin bitişini içerir.

Bu olaylar, sistemdeki veya kuyruktaki varlık sayısını artırıp azaltabileceği veya kaynakların durumunu boşta ve meşgul arasında değiştirebileceği için sistemin durumunu değiştirir. Olay listesi, simülasyon saatindeki ilerlemelerle kontrol edilir. Temel simülasyon modelinde, simülasyon saati, olay listesindeki her olaya ayrı sıçramalarla ilerler. Bu tip modele kesikli olay simülasyonu denilmektedir (Chung, 2004, s. 10). Law

ve Kelton (2000) tarafından tanımlanan simülasyon çalışmasının aşamaları Şekil 3.4'te gösterilmektedir.



Şekil 3.4. Simülasyon çalışmasının aşamaları (Law ve Kelton, 2000, s. 107)

Kesikli olay simülasyon modeli, olayların meydana geldiği zamandaki bu kesikli noktalarda yalnızca durum değişkeninin değiştiği bir model olarak tanımlanabilir. Olaylar, aktivite süreleri ve gecikmelerin bir sonucu olarak ortaya çıkar. Varlıklar, mevcut bir kaynağı beklerken muhtemelen kuyruklara katılarak sistem kaynakları için rekabet edebilmektedir. Faaliyet ve gecikme süreleri, varlıkları belirli süreler boyunca tutmaktadır.

Kesikli olay simülasyonu, durum değişkenlerinin zamanın ayrı noktalarında anlık olarak değiştiği bir temsil yoluyla zaman içinde gelişen bir sistemin modellenmesiyle ilgilidir.

Simülasyon uygulamalarında, incelenen gerçek sistem kavramsal model ile temsil edilir. Kavramsal modelin ardından operasyonel model kodlanır. Operasyonel modelin gerçek sistemin doğru bir şekilde temsil edilmesi için geçerlilik ve doğrulama süreçlerinden geçirilir. Geçerlilik, kavramsal modelin bilgisayar uygulamasının doğru olup olmadığının belirlenmesidir. Operasyonel modelin, kavramsal modeli temsil edip etmediği araştırılır. Doğrulama ise kavramsal modelin gerçek sistemin yerine geçip geçemeyeceğinin

belirlenmesidir. Bu süreçlerden model geçene kadar sınırlıdır. Eğer operasyonel ve kavramsal modeller ve gerçek dünya sistemi arasında tutarsızlıklar varsa, kavramsal model gerçek sistemi daha iyi temsil etmesi için modifiye edilmelidir. Geçerlilik ve doğrulama süreçleri tekrarlanmalıdır.

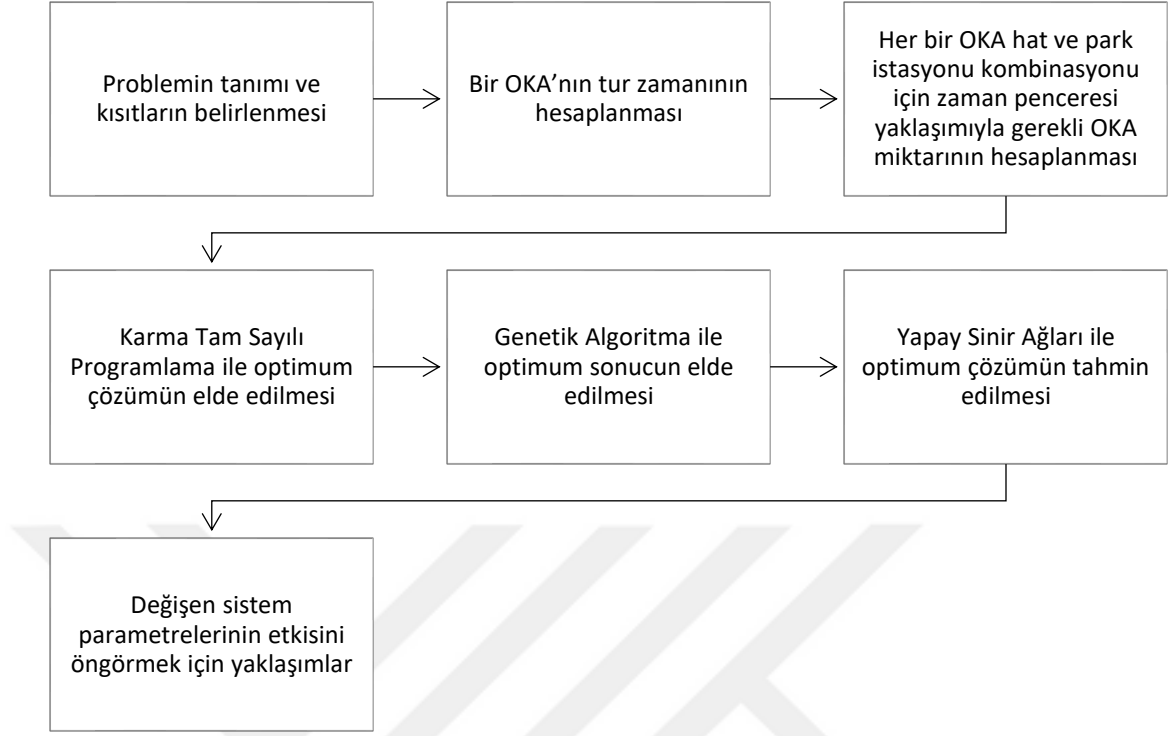
Kesikli olay simülasyonu kavramsal olarak elle yapılan hesaplamalarla yapılabilsede çoğu gerçek dünya sistemi için depolanması ve manipüle edilmesi gereken veri miktarı, kesikli olay simülasyonlarının bir dijital bilgisayarda yapılmasını zorunlu kılmaktadır (Law, 2015, s. 8). Bu çalışmada Witness Simülasyon Programı kullanılmıştır.



4. BULGULAR

OKA sistemleri, OKA'lar, merkezi kontrol bilgisayarlarını ve çevre birimlerini içermektedir. Yüksek yatırım maliyetleri nedeniyle, merkezi kontrol bilgisayarı yer almayan sistemler de kullanılmaktadır. OKA sistemi, araç ve çevresel cihazlara (hızlı şarj cihazları, radyo frekansı tanımlama (RFID) etiketleri ve iletişim cihazları gibi) evrilmiştir. Düşük maliyetli OKA'lar, yatırım tutarının düşüklüğü nedeniyle imalat atölyelerinde sıklıkla kullanılmaktadır. Bu yaklaşım, daha az esnek bir sistem oluşturulması ile; OKA, atanmış bir güzergahta sürekli çalışan basit ama etkin olmayan bir yapıya dönüşmektedir. Düşük maliyetli OKA'ların belirlenmiş tek bir yörüngede hareket etmesi beraberinde gereksiz imalat postalarını beklemeyi gerektirmektedir. Bu nedenle genellikle her OKA, bir hatta atanır ve taşıma arabası dolmadan OKA üretim postasını terk edememektedir. Bekleme olarak tanımlanan bu zaman dilimi OKA verimliliğini düşürürken aynı zamanda gerekli olan OKA sayısını arttırmaktadır.

Bir OKA'nın rotası ancak OKA program numarasının değiştirilmesi ile sağlanır. Program değişimi, belirli bir noktadan başlatılmalıdır. Merkezi olmayan, düşük maliyetli kontrol cihazlarının ilave edilmesi, program değişiminin otomatik şekilde yapılmasını sağlar ve bu sayede OKA havuzlarının kurulmasına imkân sağlanır. OKA havuzu kurulması ile OKA'ların etkinliği artırılır ve üretim talebi minimum sayıda OKA ile sağlanabilir. Bu tez çalışmasında OKA hatları gruplanarak ve OKA'lar en uygun OKA park istasyonlarına atanarak minimum OKA sayısının elde edilmesi amaçlanmıştır. Şekil 4.1 de önerilen yaklaşımın iş akışı gösterilmiştir.



Şekil 4.1. Önerilen yaklaşımın adımları

Bir atölyede gerekli optimum OKA sayılarının belirlenmesi için önerilen yaklaşımda, matematiksel model tasarımı, genetik algoritma ve yapay sinir ağları modeli kullanılarak yapılan çalışmanın işlem adımları aşağıda verilmiştir:

Adım 1: Problemin tanımı ve Çerçevesi (Bölüm 4.1): Düşük maliyetli OKA'ların, OKA havuzları kurulması ile gerekli minimum sayıda OKA miktarının elde edilmesi çalışmada ele alınan problemi oluşturmaktadır. Düşük maliyetli OKA'ların OKA havuzlarının oluşturulması ile kullanım oranlarının artırılması ve gerekli minimum OKA sayısının bulunması için öncelikle düşük maliyetli OKA'ların özellikleri ve çalıştığı ortam incelenmiştir. Bir otomotiv fabrikasında var olan mevcut OKA sisteminin incelenmesi ile problemin çerçevesi oluşturulmuştur. OKA'ların çalıştığı ortam, OKA'nın özellikleri sistem parametreleri (hız, şarj süresi, mesafeler) analiz edilmiştir. Mevcut sistemin çalışma biçimi sunulup istenen çalışma biçimi belirlenmiştir. Amaç OKA sayısının kullanım oranının artırılması olduğu için OKA'ların sahip olduğu özellikler ve çalıştığı atölye ortamının özelliklerinin tespiti problemin doğru şekilde modellenmesi için

önemlidir. Sistem parametrelerinin tespiti ile ilk önce OKA bir turu ne kadar sürede tamamladığının tespit edilmesi gerekmektedir. Bu nedenle bir sonraki işlem adımında tur zamanının hesaplanması yapılmıştır.

Adım 2: OKA'nın tur zamanının hesaplanması (Bölüm 4.2): Bu işlem adımında OKA'nın bir turu ne kadar sürede tamamladığı hesabı yapılmıştır. OKA hızı, postalar arası mesafe, şarj ve yükleme süreleri ve OKA arabası taşıma kapasitesi girdi olarak kullanılırken her bir turun kaç dakikada tamamlandığı hesaplanmaktadır. OKA'nın, bir taşıma emri geldiğinde OKA hat kombinasyonları ve atanacak park istasyonlarına göre üretim taşıma emirlerini gerçekleştirmek için kaç adet OKA'ya ihtiyaç olduğunun hesaplanması bir sonraki adımda ele alınmıştır.

Adım 3: Her bir OKA hat ve park istasyonu kombinasyonu için zaman penceresi yaklaşımıyla gerekli OKA miktarının hesaplanması (Bölüm 4.3): OKA tur sürelerinin girdi kabul edilmesiyle üretimde kesinti yaratmadan her OKA hat ve park istasyonu kombinasyonuna göre gerekli OKA sayısı hesaplanmıştır. Zaman penceresi yaklaşımı ile her taşıma emri izin verilen süre aralığında postaya taşınacaktır. İzin verilen hizmet süreci başlangıcı öncesi ve izin verilen hizmet süresi sonrası yapılan taşımalara izin verilmeyecektir. Bu yaklaşım ile problem zaman penceresi problem türüne dönüşmektedir. Önce basit bir örnek ele alınarak problem incelenmiş ve daha sonra karmaşık durumlar için geliştirilen algoritma sunulmuştur. Bu sayede her OKA hat kombinasyonuna ve atanmış her park istasyonuna göre gerekli OKA miktarı elde edilmiştir. Problem artık var olan hat kombinasyonları ve park istasyonlarına göre en uygun çözümün bulunmasıdır. Bir sonraki adım bu seçeneklerden en uygun olanın bulunmasıdır.

Adım 4: Karma Tam Sayılı Programlama ile optimum çözümün elde edilmesi (Bölüm 4.4): Hesaplanan OKA hat ve park istasyonu kombinasyonuna göre optimal çözümün karma tam sayılı programlama modelinin oluşturulması ve geliştirilen tam sayılı programın basit bir OKA sisteminde denenmesidir. Basit bir OKA sistemi ile önerilen yöntem incelenmiştir.

Adım 5: Genetik Algoritma ile optimum sonucun elde edilmesi (Bölüm 4.5): OKA sisteminde hat sayılarının artmaya başlamasıyla birlikte problemin, karma tamsayı

programlama modeliyle çözülmesi zorlaşmaktadır. Bu sebeple problem genetik algoritma yardımıyla ele alınmıştır. Elde edilen çözüm kesikli olay benzetim metoduyla modellenip GA ile elde edilen çözümün üretimde OKA kaynaklı bir duruş yaratıp yaratmadığı incelenmiştir. Geliştirilen model daha sonra oluşturulan bir veri setine uygulanmış, sonuçları yeniden kesikli olay benzetim tekniği ile kontrol edilmiştir. İncelenen OKA sistemleri çok karmaşık olduğunda, ele alınan OKA hatları sayıları çok arttığında GA optimizasyonu çözüm süresi artacaktır. Bu durum için GA'ya alternatif olabilecek hızlı tahmin yapmaya elverişli bir yöntem olarak Yapay Sinir Ağları modeli kurulması bir sonraki adımda ele alınmıştır.

Adım 6: Yapay Sinir Ağları ile optimum çözümün tahmin edilmesi (Bölüm 4.6): İncelenen OKA sistemleri bazen çok sayıda OKA hattı içerir ve hızlı çözümler gerektiği durumlarda gereken minimum sayıdaki OKA ihtiyacının tahmini için Yapay Sinir Ağları modeli geliştirilmiş ve tahmin sonuçları karşılaştırılmıştır. Ele alınan problem, çalışmanın bu bölümüne kadar OKA'ların sabit parametrelerle devam ettiği varsayımı altında gerçekleştirilmiştir. Sistem parametrelerinin zaman içinde değişebilir veya OKA sayılarının azaltılması istenebilir. Bu sebeple sistem parametrelerinin OKA sayısına etkisi bir sonraki adımda ele alınmıştır.

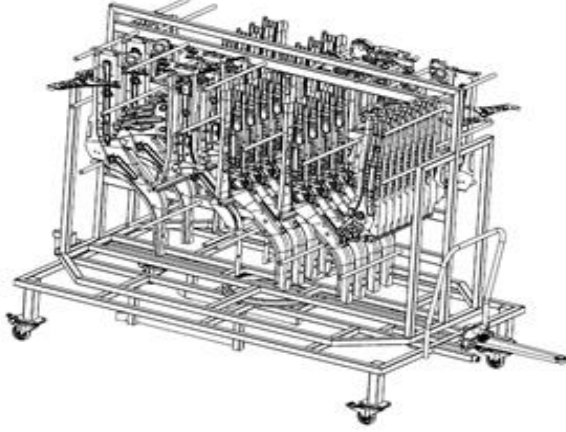
Adım 7: Değişen sistem parametrelerinin etkisini öngörmek için yaklaşımlar (Bölüm 4.7): Gerekli OKA sayısının hesaplanmasında sabit olarak ele alınan parametreler (OKA hızı, mesafeler, şarj süresi ve taşıma arabası kapasitesi) çeşitli sebeplerle değişebilir veya değişimi istenebilir. Sistem parametrelerinin OKA sayısı üzerindeki etkisi Yüzey Yanıt Yöntemi, Çoklu Doğrusal Regresyon ve YSA ile incelenmesiyle parametrelerinin etkisi analiz edilmiş ve gerekli OKA sayısı tahmini yapılmıştır.

4.1. Problemin Tanımlanması

Düşük maliyetli OKA sistemleri merkezi bir kontrolcü içermemektedir. OKA'lar manyetik bantları takip ederek rotalarını tamamlar ve atanmış bir güzergâhta (OKA hattında) hareket ederler.

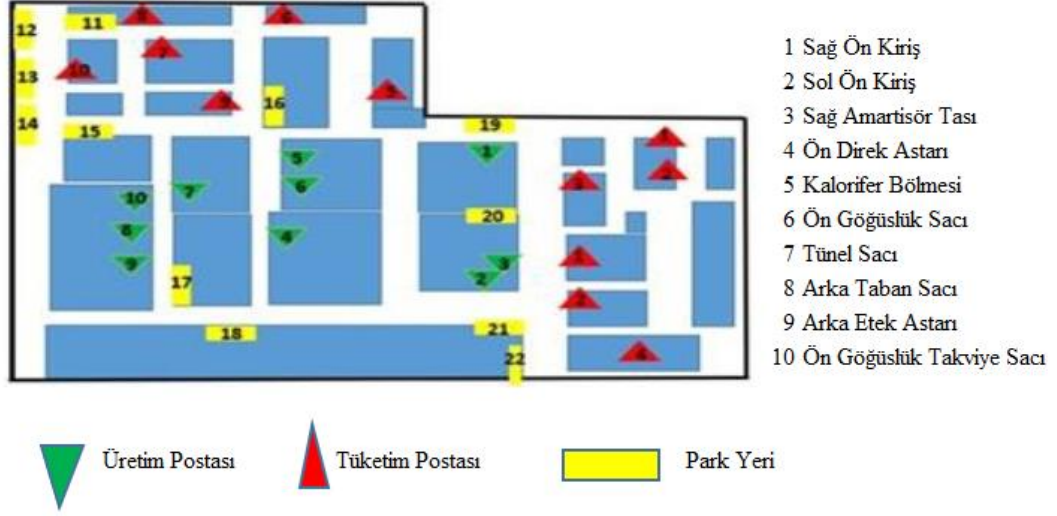
Üretim postalarında üretilen parçalar, taşıma arabalarına yüklenir. OKA üzerinde bulundurduğu kanca (hook) sistemi ile taşıma arabasının altına girer. OKA kanca sistemi

kanca pimini yukarı kaldırır ve OKA ile taşıma arabasının kenetlenmesi sağlanır. OKA taşıma arabasını tüketim postasına kadar götürür. Tüketim postasında, dolu taşıma arabasını bırakır. Bırakma işlemi kanca mekanizmasının pimi aşağıya indirmesi ile gerçekleştirilir. Taşıma arabası Şekil 4.2’de gösterilmektedir.



Şekil 4.2. Taşıma arabası

İncelenen sistemde 10 adet üretim postası, 10 adet tüketim postası mevcuttur. Her üretim postası ilgili tüketim postasını beslemektedir. Yerleşim sabit olup yerleşim değişiklikleri yüksek maliyet gerektirdiğinden yerleşimin yeniden değiştirilmesinin imkânı bulunmamaktadır. Üretim postaları, tüketim postaları Şekil 4.3’te gösterilmiştir.



Şekil 4.3. Yerleşim Planı

Mevcut durumda OKA'lar sadece bir OKA hattına atanmış durumdadır. Taşıma yapmadıkları durumlarda ya üretim postalarında ya da tüketim postalarında beklemektedir.

Atölyedeki yerleşim sıkıntısı sebebiyle OKA'lar için 12 adet park istasyonu tespit edilebilmiştir. Ayrıca her üretim postası da kendi park istasyonu olarak tanımlanarak toplamda 22 park istasyonu oluşturulmuştur. Bununla birlikte ilk 10 park istasyonu sadece kendi hatları için mümkündür.

Park istasyonlarının oluşumu için ilgili yerlere merkezi olmayan kontrol dolapları, besleme kabloları ve haberleşme modülleri eklenmelidir. Dolayısıyla her park istasyonu eklenmesi ek bir maliyet unsuru yaratmaktadır. Aynı zamanda park istasyonu alan da kullandığı için kısıtlı olan bir kaynağın kullanılması durumu da söz konusudur.

OKA'lar yerlere gömülen Radyo Frekansı Tanımlama (RFID) tagları ile hareketlerini değiştirirler. Hız, alan tarayıcısındaki kontrol edilen alan, durma ve diğer birçok parametre OKA programında ilgili tag numarası ile bütünleştirilmiş ve sıralanmıştır. Program adımı program step numarası ile adlandırılır. OKA yeni bir tagın üzerinden geçtiğinde step numarası 1 artar, o stepdeki tag numarası ile okuduğu tag numarası karşılaştırılır. Eğer uygunsa o stepteki komutlar gerçekleştirilir. Uygun değilse OKA

hataya geçer. Programda en son adımda program ilk baştaki adıma geri döner ve hep aynı program üzerinde ilerler. OKA'lar üzerinde 200 program bulunmaktadır. Her program bir güzergahı karşılamaktadır. OKA'nın program numarası elle veya haberleşme modülleri sayesinde değişebilmektedir. Kullanılan haberleşme modülleri infra red aracılığı ile OKA ve kontrol ünitesi arasında haberleşmeyi sağlamaktadır. Haberleşme mevcut durumda hızlı şarj üniteleri ve kavşak yönetim sistemlerinde kullanılmaktadır.

Üretim ve tüketim postaları arası mesafe, şarj süresi, taşıma arabalarının kapasitesi, Çizelge 4.1'de gösterilmektedir. OKA hatlarının mesafesi (m), şarj süresi (dakika), yükleme ve boşaltma süreleri ile OKA hattının toplam süreleri hesaplanmıştır. OKA hattının toplam mesafesi ortalama hıza bölünmesiyle ve diğer sürelerin bu zamana eklenmesi ile her hattın bir turu kaç dakikada tamamladığı hesaplanmaktadır.

Çizelge 4.1. OKA hatları sistem parametreleri

OKA HATLARI										
	1	2	3	4	5	6	7	8	9	10
Üretim ==>Tüketim (m)	128	133	51	103	75	25	74,5	69	139	49
Tüketim ==>Üretim (m)	67	41	66	119	60	78	172	167	56	169
Toplam Mesafe	195	174	117	222	135	103	246,5	236	195	218
Şarj Süresi (dak)	2	2	2	2	2	2	2	2	2	2
Yük.- Boş.Süresi (dak)	2	2	2	2	2	2	2	2	2	2
OKA toplam süre (dak)	13,75	12,7	9,85	15,1	10,75	9,15	16,325	15,8	13,75	14,9

Park istasyonları ve diğer istasyonlar arası mesafe Çizelge 4.2'de gösterilmektedir. Park istasyonları ile postalar arası mesafe ortalama OKA hızına bölünmesiyle OKA'nın park istasyonuna gidiş süresi hesaplanmıştır. Hız kabulleri: OKA'lar tüketim postası ile üretim postaları arası 20 m/dak ile ilerlerken; park istasyonlarına gidiş-gelişlerinde 15 m/dak ile ilerlemektedir. Mevcut durumda OKA'lar tek güzergâhta üretim ve tüketim postaları arasında çalışırken bekleme zamanları oldukça fazladır. OKA verimliliği düşmesi ile sonuçlanan bu durum gerekli OKA miktarını da arttırmaktadır.

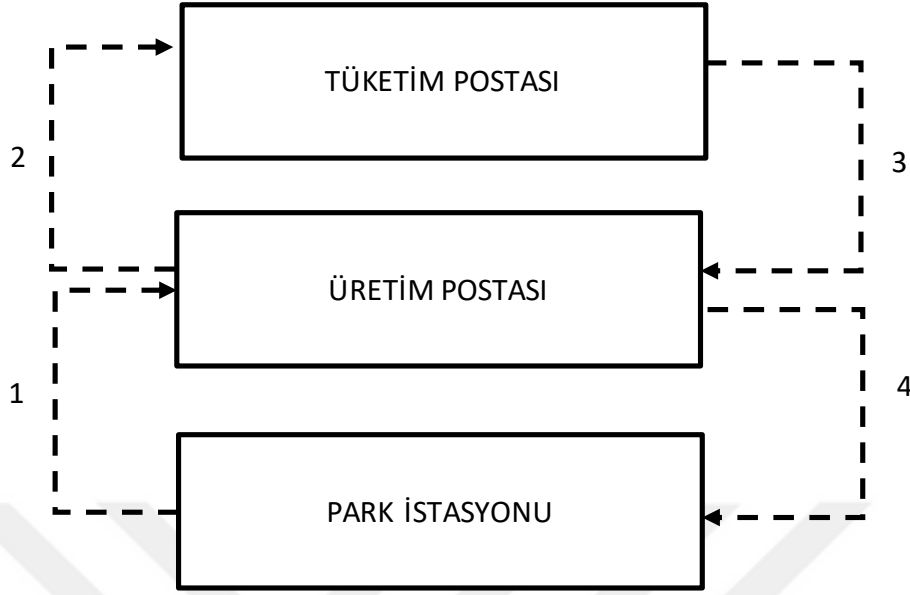
OKA havuz çalışma modeli Şekil 4.3'te gösterilmiştir. Tüketim postasındaki taşıma arabasındaki parçalar belirlenen seviyeye düştüğünde operatör onay butonuna basarak bir OKA taşıma emri oluşturmaktadır. Park istasyonunda bulunan OKA hareket ederek önce

üretim postasına gelir ve dolu taşıma arabasını alır. Dolu taşıma arabası tüketim postasına taşınır. OKA tüketim postasında dolu taşıma arabasını bırakır ve boş olan taşıma arabasını alır. OKA, boş taşıma arabasını üretim postasına getirir ve boş taşıma arabasını üretim postasına bıraktıktan sonra boş olarak park istasyonuna gider. Yeni bir OKA taşıma emrinin gelmesini bekler. Mevcut taşıma yöntemi ile karşılaştırıldığında ilk fark bu yöntemin çekme sistemine uygunluğudur.

Çizelge 4.2 Park istasyonları ve diğer istasyonlar arası mesafe (m)

		Üretim İstasyonu									
		1	2	3	4	5	6	7	8	9	10
Park istasyonları	11	296	326	287	410	114	140	92	80	136	58
	12	298	328	289	412	116	142	94	82	138	60
	13	296	326	287	410	114	140	92	80	136	58
	14	296	326	287	410	140	140	120	100	100	48
	15	236	266	227	350	140	140	120	100	100	48
	16	200	230	191	314	56	70	106	130	20	100
	17	184	154	250	238	250	290	230	300	184	172
	18	168	138	234	222	258	298	238	308	192	182
	19	106	136	97	220	160	160	199	269	116	196
	20	36	66	80	150	240	240	279	349	200	280
	21	62	32	128	88	294	294	333	403	256	336
	22	106	76	172	94	348	348	387	457	300	380

Akış tüketim postasının isteğine bağlı olduğu için üretim postasının itmesiyle değil, çekme sistemine geçilmesi aynı zamanda tüketim postasında gereksiz stoğu engeller. İkinci fark OKA'lar bekleme zamanında diğer postaların taşıma işlerini de gerçekleştirdiği için OKA verimliliği artmaktadır. Diğer yönden OKA'lar üretim ve tüketim postası haricinde park istasyonlarına da gittiği için kat ettikleri yol artmaktadır. Havuz düzeni ile çalışma şekli Şekil 4.4'te gösterilmiştir.



Şekil 4.4. Havuz düzeni ile çalışma şekli

4.2. OKA Tur Zamanının Hesaplanması (T_s)

OKA tur zamanı formül 4.1’de gösterildiği gibi iki bileşenden oluşmaktadır ve birinci taraf, üretim istasyonu ile tüketim istasyonu arasındaki transfer sürelerinden oluşmaktadır. Bu zaman dilimi Şekil 4.3’te ikinci hareket ve üçüncü hareket olarak gösterilmiştir. Bu süre D_2 ve D_3 mesafelerinin ortalama hıza bölünmesi, şarj süresi, yükleme ve boşaltma süresinin toplamıyla ifade edilmektedir ve denklem 4.2’deki gibi hesaplanır. Denklem 4.1’deki ikinci periyot ise park istasyonu ile üretim istasyonu arasındaki süredir. Park istasyonuna geliş ve gidiş mesafesinin ortalama OKA hızına bölümüne, ek şarj süresinin eklenmesi ile elde edilmektedir. Bu süre denklem 4.3’te ifade edilmiştir. Şekil 4.3’teki birinci hareket ve dördüncü hareket bu periyodu oluşturmaktadır.

T_{PCPS} : Üretim postasından tüketim postasına ve üretim postasına kadarki transfer süresi

T_{PAPAS} : Park istasyonundan üretim postasında ve üretim postasından park istasyonuna kadar gerçekleşen transfer süresi

D_2 : Üretim postasından tüketim postasına olan mesafe

D_3 : Tüketim postasından üretim postasına olan mesafe

D_1 : Park istasyonundan üretim postasına olan mesafe

D_4 : Üretim postasından park istasyonuna olan mesafe

T_{chr} : Şarj süresi

DT_{chr} : Eklenen şarj süresi

$T_{L\&UL}$:Yükleme ve boşaltma zamanı

$$T_s = T_{PCPS} + T_{PaPPaS} \quad (4.1)$$

$$T_{PCPS} = \frac{D_2 + D_3}{20} + T_{chr} + T_{L\&UL} \quad (4.2)$$

T_{PCPS} zarfında OKA'ların hızı 20 m/dak olarak kabul edilmiştir.

$$T_{PaPPaS} = \frac{D_1 + D_4}{15} + DT_{chr} \quad (4.3)$$

T_{PaPPaS} zarfında OKA 15m/dak hız ile hareket ettiği kabul edilmektedir.

4.3. OKA Kombinasyonları ve Park İstasyonları İçin Gerekli OKA Miktarının Hesaplanması

OKA hat kombinasyonları ve park istasyonlarına göre gerekli OKA sayısının hesabından önce notasyon hakkında bazı açıklamalar verilmiştir. Üç OKA hattı olarak üç üretim postası ve üç tüketim postasından oluşan bir örnek verilmiştir. Bu 3 hattın kendi içinde kombinasyonu 8 adettir. Olası kombinasyonlar Çizelge 4.3'te verilmiştir. Çizelge 4.3.'te 0; ilgili kombinasyonda o OKA hattının dahil olmadığını 1 ise; ilgili kombinasyonda o OKA hattının dahil olduğunu göstermektedir. 1 nolu kombinasyonda hiçbir OKA hattı kombinasyonu girmemektedir; 8 nolu kombinasyonda ise tüm OKA hatları kombinasyona girmektedir.

Çizelge 4.3. OKA hatları ve oluşturulan kombinasyonlar

		OKA Hatları		
		1	2	3
Kombinasyonlar	1	0	0	0
	2	0	0	1
	3	0	1	0
	4	0	1	1
	5	1	0	0
	6	1	0	1
	7	1	1	0
	8	1	1	1

Uygulanan problemde 10 OKA hattı mevcuttur. Bu uygulamada bir kombinasyon aşağıdaki gibidir:

[0 0 1 0 0 1 0 0 1 1]: 1., 2., 4., 5., 7., 8. no'lu OKA hatları bu kombinasyona girmediği buna karşın 3., 6., 9., 10. OKA hatlarının kombinasyona dahil olduğunu gösterir. Kombinasyon yazımını daha anlaşılır kılmak için aşağıda gösterilen notasyon ile yazım ve anlaşılabilirlik artırılmaya çalışılmıştır. Kombinasyondaki değerler 0 veya 1 değerini aldığı için bu yazım ikili kod olarak kabul edilmiş ve onluk düzene çevrilmiştir.

$[0 0 1 0 0 1 0 0 1 1]_2 = 147_{10}$ Bu sayede 147. OKA kombinasyonu : 1., 2., 4., 5., 7., 8. no'lu OKA hatları bu kombinasyona girmediği buna karşın 3., 6., 9., 10. OKA hatlarının kombinasyona dahil durumu ifade etmiştir.

Tüm OKA hat kombinasyonları ve park istasyonlarına göre gerekli OKA miktarının belirlenmesi özellikle OKA hat sayısı ve park istasyonları sayısı arttıkça karmaşık hale gelmektedir; bu nedenle her OKA kombinasyonu ve park istasyonu için OKA sayısını belirleyen bir algoritma geliştirilmiştir. Deterministik sipariş bazlı zaman pencere yöntemi kullanılmıştır. Zaman pencereleri, her sipariş için izin verilen hizmet süresinin (a) başlangıcını ve izin verilen hizmet süresinin (b) bitişi arasında taşıma yapılmasını gerektirmektedir. Her sipariş bu zaman aralıklarında yerine getirilmelidir. Bir OKA, parçaları izin verilen servis süresinin başlangıcından önce ve bitiminden sonra teslim edemez.

Değişkenlerin tanımı aşağıda verilmiştir:

O: Taşıma emirleri kümesi

o: Taşıma emri $o \in O$ olmak üzere

l: Tüketim postasına bir sonraki OKA gelmesi için gereken süre

a: izin verilen hizmet süresi başlangıcı

b: İzin verilen hizmet süresi sonu

k: OKA numarası

t_s : Bir taşıma işleminin gerçekleştirilmesi için gereken süre

f_{ok} k numaralı OKA'nın o emrine atandığı süre

e_{ok} k numaralı OKA'nın o numaralı emirden kurtulduğu zaman. Bu anda OKA'ya yeni bir taşıma emri atanabilir.

$$a_o = a_{o-1} + l \quad (4.4)$$

$$b_o = b_{o-1} + l \quad (4.5)$$

$$a_o \leq f_{ok} \leq b_k \quad (4.6)$$

$$e_{ok} = f_{ok} + t_s \quad (4.7)$$

Sistemdeki OKA'lar yukarıdaki kısıtları karşılamadığında, sistemdeki OKA miktarı 1 artırılır. Denklem 4.4 ve denklem 4.5'te hizmet sürelerinin başlangıç ve bitiş süreleri hesaplanmaktadır. OKA'ya atama bu zaman penceresi içinde olması kısıtı denklemler 4.6'da gösterilmiştir. Denklem 4.7'de OKA'nın yeni taşıma emri alabileceği an gösterilmiştir.

Yaklaşım, kısa bir uygulama ile açıklanabilir. Bir park istasyonu ve bir OKA hattı için, OKA'nın bir turunu tamamlama süresi elli dakikadır (t_s). Tüketim postasına her otuz dakikada bir OKA gelmesi gerekmektedir. İlk izin verilen hizmet süresi (a_0) onuncu dakikadır ve izin verilen son hizmet süresi (b_0) onikinci dakikadır. Bu bilgilere göre Çizelge 4.4. oluşturulmuştur.

Çizelge 4.4. Zaman pencere yöntemi örneği

o	a _o	b _o	f _{ok}	k	t _s	e _{ok}
1	10	12	10	1	50	60
2	40	42	40	2	50	90
3	70	72	70	1	50	120
4	100	102	100	2	50	140
5	130	132	130	1	50	180
6	160	162	160	2	50	210
7	190	192	190	1	50	240
8	220	222	220	2	50	270

Çizelge 4.4'e göre; OKA ilk taşıma emri onuncu dakikada gelir. İlk OKA sisteme atanır. OKA ilk emir ile atmışıncı dakikaya kadar doludur, ancak kırkıncı dakikada ikinci taşıma emri gelir ve sistemde bu emri gerçekleştirebilecek bir OKA olmadığı için sisteme ikinci bir OKA eklenir ve bu emir ikinci OKA ile gerçekleştirilir. İki OKA ile tüm taşıma emirleri gerçekleştirilir dolayısıyla gerekli OKA sayısı ikidir. Basit bir problem için bu zaman penceresi hesaplamasının elle veya basit bir Excel formunda gerçekleştirilebileceği açıktır, ancak on OKA hattı, yirmiiki park istasyonu için yardımcı bir algoritma önerilmiştir. Şekil 4.5 ve Şekil 4.6'da sözde (psuedo) kodları, algoritma ve akış şeması gösterilmektedir.

Algoritma

Gerekli OKA miktarını hesapla

WHILE durma şartı sağlanmadığında **DO**

Başla

FOR park istasyonu: = 1 **TO** 22 **DO**

FOR istasyon kombinasyonu: = 1 **TO** 1024 **DO**

 AGV matriks=ref agv matriks

 Atanan =0

 Kullanılan=0

FOR taşıma emri: = 1 **TO** 1053 **DO**

FOR üretim istasyonu indeksi: = 1 **TO** 10 **DO**

IF taşıma emri istasyonu == üretim istasyonu **THEN**

 min OKA bırakma süresini bul, ve miniOKA ve indeksini miniindex olarak tanımla

IF miniOKA<aj **THEN**

 Atanan=miniindex

 fk=a j

 ek=fk +ts

Else IF aj<miniOKA<bj **THEN**

 fk=miniOKA

 ek=fk +ts

Else IF bj<miniOKA **THEN**

 Kullanılan=Kullanılan+1

 Atanan=Kullanılan

 fk=aj

 ek=fk +ts

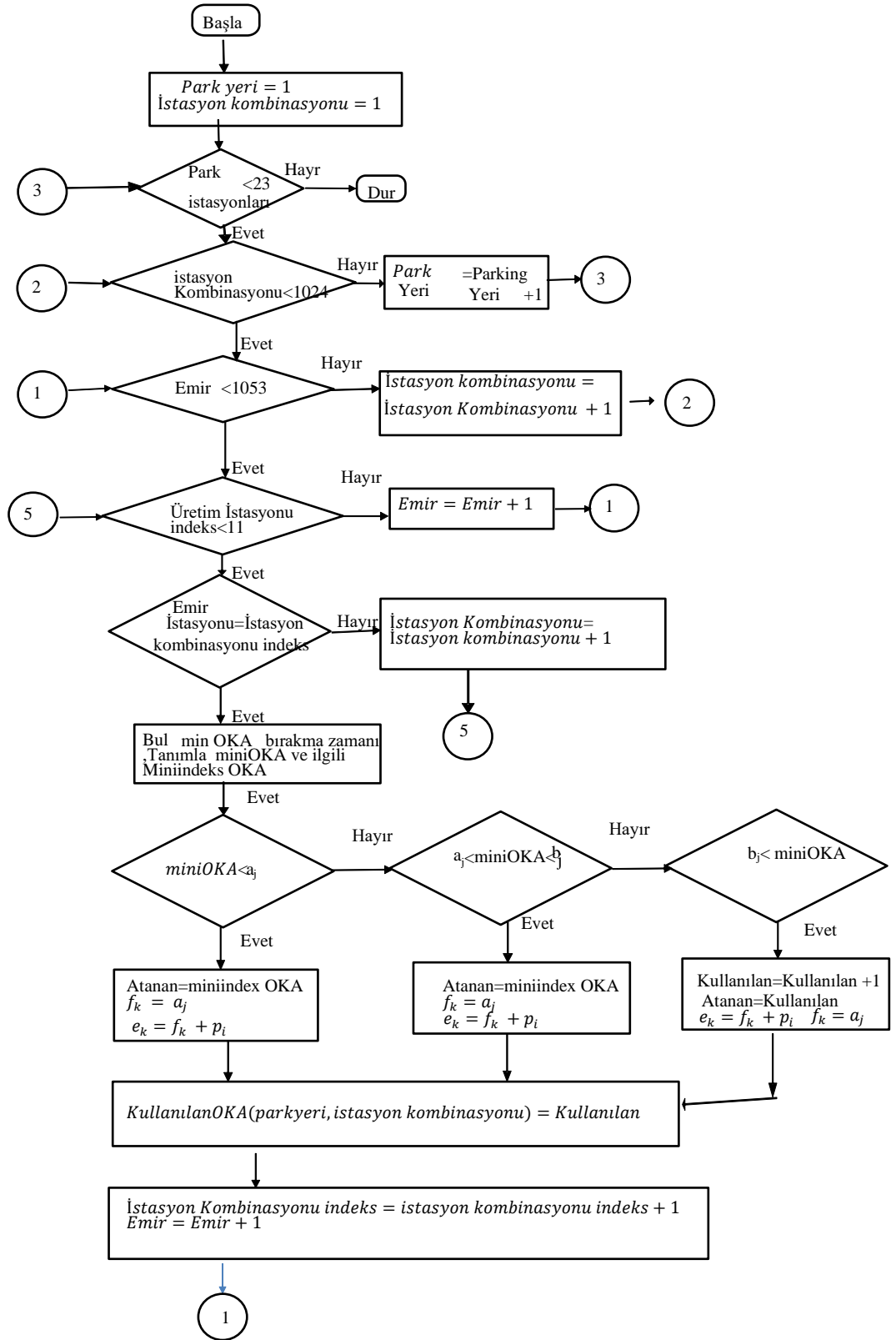
END

END

Kullanılan OKA(parking yeri,istasyon kombinasyonu)=Kullanılan

END

Şekil 4.5. Önerilen algoritma



Şekil 4.6. Akış çizelgesi

Önerilen algoritma, yukarıda anlatılan hesaplama yöntemine benzer şekilde çalışmaktadır. İlk olarak birinci siparişe bir OKA atanır ve algoritma OKA'nın serbest kalacağı süreyi hesaplar. Yeni bir sipariş ortaya çıktığında, algoritma zaman penceresi içinde atanabilecek herhangi bir OKA olup olmadığını araştırır. Boşta olan bir OKA varsa algoritma OKA'yı bu siparişe atar, boşta OKA bulunamazsa algoritma OKA sayısını arttırarak bu yeni OKA'yı siparişe atar ve siparişler var oldukça algoritma çalışır. Bu algoritma Matlab R17'de çalıştırılmakta ve tüm OKA hat ve park istasyonları için hesaplanmaktadır.

4.4. Karma Tamsayılı Programlama ile Gerekli Optimum OKA Sayısının Hesaplanması:

Bu çalışma deterministik OKA emirlerini sert zaman pencere yöntemi ile değerlendirerek OKA havuzları kurarak ve OKA hatlarını bu OKA havuzlarına atayarak gerekli OKA miktarını minimize etmeyi amaçlamaktadır.

I: Park istasyonlarının kümesi

S: OKA hatlarının kümesi

J: OKA hatlarının kombinasyonu kümesi

B: Bir park istasyonuna atanmış toplam OKA hatlarının sayısı

Ω_s : s . OKA hattı için OKA hat kombinasyonu kümesi, $s \in S$

GP_{ij} : i. park istasyonu ve j. OKA kombinasyonu için gerekli olan OKA miktarı

UP_{ij} : Karma tamsayılı lineer program modeli için 2'li (binary) karar değişkeni, eğer 1 i. park istasyonu ve j. OKA hat kombinasyonu atanmıştır; değilse değer 0 dır; $i \in I$ ve $j \in J$

B değeri aşağıdaki gibi hesaplanır:

$$B = 2^I$$

Yukarıda belirtilen gösterimler, parametreler ve karar değişkenleri kullanılarak problem aşağıdaki gibi formüle edilir.

Amaç fonksiyonu:

$$\text{Min } Z = \sum_{j \in J} \sum_{i \in I} GP_{ij} UP_{ij} \quad (4.8)$$

Kısıtlar :

$$\sum_{j \in \Omega_s} \sum_{i \in I} UP_{ij} = 1 \quad \forall s \in S \quad (4.9)$$

$$\sum_{j \in B} UP_{ij} = 1 \quad \forall i \in I \quad (4.10)$$

$$UP_{ij} = 0 \text{ veya } 1 \quad (4.11)$$

Amaç fonksiyonu (4.8), gerekli OKA sayısını en aza indirir. Kısıt (4.9), her OKA hattının bir park istasyonuna atanmasını sağlar. Benzer şekilde, kısıt (4.10), her park istasyonunun yalnızca bir istasyon hattı kombinasyonuna atanmasını sağlar. Bir OKA hattının durumunu açıklamak için ikili (binary) kod atanır (4.11).

Geliştirilen model iki OKA hattı ve iki park istasyonu olarak ele alınması ile park istasyonlarına ve OKA hat kombinasyonlarına göre gerekli OKA miktarı Çizelge 4.5'te gösterilmektedir.

Çizelge 4.5. Park istasyonu OKA kombinasyonları ve gerekli OKA miktarları

Park istasyonu	OKA Hat Kombinasyonu		Gerekli OKA miktarı	
1	[00]	0	GP ₁₀	0
1	[01]	1	GP ₁₁	5
1	[10]	2	GP ₁₂	2
1	[11]	3	GP ₁₃	2
2	[00]	0	GP ₂₀	0
2	[01]	1	GP ₂₁	6
2	[10]	2	GP ₂₂	3
2	[11]	3	GP ₂₃	4

Buna göre;

$$\text{Min } Z = \sum \sum UP_{ij} \times GP_{ij} \quad (4.12)$$

$$UP_{11} + UP_{13} + UP_{21} + UP_{23} = 1 \quad (4.13)$$

$$UP_{12} + UP_{13} + UP_{22} + UP_{23} = 1 \quad (4.14)$$

$$UP_{10} + UP_{11} + UP_{12} + UP_{13} = 1 \quad (4.15)$$

$$UP_{20} + UP_{21} + UP_{22} + UP_{23} = 1 \quad (4.16)$$

$$UP = 0 \text{ veya } 1 \quad (4.17)$$

Denklem 4.12’de amaç fonksiyonu, denklem 4.13-4.17’te kısıtlar gösterilmiştir. Matematiksel modelin çözümüyle $UP_{13} = 1$ ve $UP_{20} = 1$ sonuçları elde edilir. Böylece iki OKA hattı bir numaralı park istasyonuna atanır ve gerekli OKA sayısı beş olarak belirlenmiştir. Elde edilen çözüm optimum çözümdür, karma tamsayılı programlama optimum sonuca her zaman ulaşır ancak OKA hatları arttığında, denklemlerdeki kısıtlamalar katlanarak artar. Bu durum, matematiksel yöntemi uygunsuz kılmaktadır. Bu nedenle, artan OKA hatları ve park istasyonları için optimuma yakın sonuçlar elde edebilmek için yapay zekâ uygulamaları kullanılması amaçlanmıştır. Genetik algoritma ve yapay sinir ağı teknikleri kullanılarak algoritmalar geliştirilmiştir.

4.5. Gerekli OKA Sayısının Optimizasyonu için Genetik Algoritma Modelinin Geliştirilmesi:

Genetik algoritmalar (GA), doğadaki olayları taklit eden meta-sezgisel optimizasyon yöntemidir.

Holland (1992): genetik algoritmanın adımlarını aşağıdaki gibi önermiştir:

Adım 1 (Başlatma): Popülasyon oluşturun.

Adım 2 (Evrim): Her bireyin uygunluk değerini hesaplayın.

Adım 3 (Seçim): Bireyler arasından ebeveynleri seçin.

Adım 4 (Crossover): Crossover operatörünü ebeveynlere uygulayın.

Adım 5 (Mutasyon): Mutasyon operatörünü uygulayın.

Adım 6 (Sonlandırma): Sonlandırma kriteri karşılanana kadar Adım 2-5’i tekrarlayın.

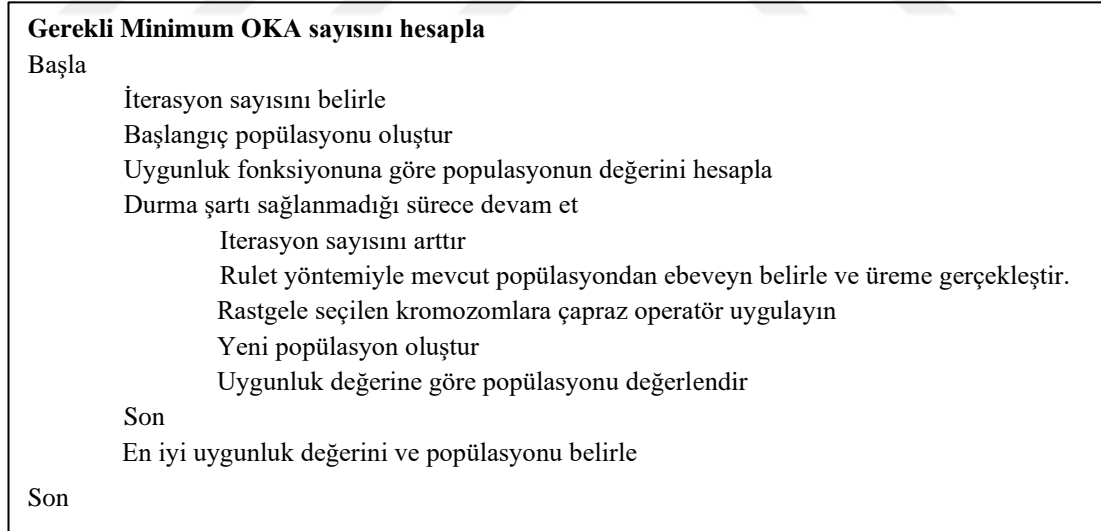
Genetik algoritmalar (GA), kromozom adı verilen kodlanmış değerlerle çalışır ve her bir kromozom bir çözüm adayıdır. Genetik operatörlere dayalı en uygun çözümü bulmaya çalışırken uygunluk fonksiyonu ile tüm kromozomlar analiz edilir ve kriterler sağlanana kadar algoritma çalışmaya devam eder (Ene ve diğerleri. 2016).

Algoritma performansı üzerindeki doğrudan etkileri nedeniyle, kodlama yönteminin seçimi GA için kritik bir karardır. Çalışmada gerçek değer kodlaması uygulanmıştır. Kromozom Örnekleri Çizelge 4.6’daki gibidir:

Çizelge 4.6 Kromozom örnekleri

	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8	Gen 9	Gen 10
Birey 1	12	17	10	9	13	4	9	9	11	12
Birey 2	10	4	9	20	18	15	15	8	15	7
Birey 3	22	21	19	22	20	10	22	15	21	9
Birey 4	9	3	1	9	17	15	18	13	12	15
Birey 5	13	4	4	9	8	12	16	11	6	7
Birey 6	18	13	15	5	5	22	1	20	4	6
Birey 7	2	19	20	6	20	19	4	4	6	5
Birey 8	11	17	22	22	4	2	15	20	2	10
Birey 9	11	10	14	14	7	13	14	6	20	18
Birey 10	20	18	4	11	17	15	5	14	4	22
Birey 11	20	8	12	16	16	20	22	12	20	9
Birey 12	3	1	13	4	12	4	15	10	21	14

Her birey 10 genden oluşmaktadır. Her bir gen ilgili OKA hattının hangi park istasyonuna atandığını göstermektedir. Birey bir için ilk OKA hattı 12 nolu park istasyonuna atanmış, iki nolu OKA hattı 17 nolu park istasyonuna atanmıştır. Mevcut çalışmada kullanılan GA algoritmasının sözde kodu Şekil 4.7'de sunulmaktadır.



Şekil 4.7. Genetik algoritmanın sözde kodu

Her kromozom S sayıda (OKA hat sayısı) genden oluşur. Genler, OKA hatlarının atandığı park istasyonunu gösterir. Başlatma aşamasında, her bir gen 22 park istasyonu arasında rastgele atanır. Uygunluk fonksiyonu, gerekli OKA sayısının toplamıdır. Bir kromozomun uygunluk değeri için aşağıdaki adımlarla hesaplanır:

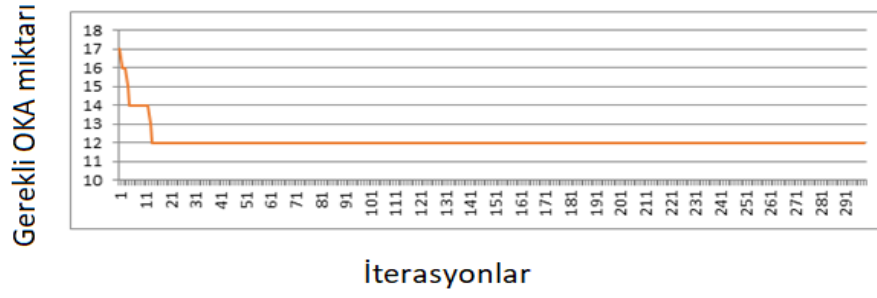
Çaprazlama operatörü, mevcut gen potansiyellerini araştırır. Popülasyon, sorunu çözmek için gereken olası tüm çözümleri içermeyebilir ve sonuçta elde edilen çözümler tatmin edici bir olmayabilir. Bu nedenle, mevcut gen havuzundan yeni bir kromozom üretebilen bir operatöre ihtiyaç vardır. Bu görev mutasyonla gerçekleştirilir. Basit genetik algoritmada mutasyon, kromozomdaki herhangi bir pozisyonun değerini rastgele değiştirir. İkili kodlayan kromozomda mutasyon, 1'in 0'a ve 0'ın 1'e dönüştürülmesi anlamına gelir. Dizilerdeki genler, çaprazlamada olduğu gibi belirli bir mutasyon oranı ile mutasyona uğrar (Sastry ve diğerleri, 2005). Mutasyon operatörü, popülasyonun herhangi bir yerel optimum noktaya yakınsamasını önleyebilir. Gerçek problemlerde, mutasyon oranları genellikle düşük tutulur. Bu seçimin nedeni, çaprazlama sonucunda elde edilen yüksek uygunluk değerlerine sahip dizileri kaybetmemek içindir. Mutasyonun amacı genetik çeşitliliği sağlamak veya korumaktır.

Çalışmada tek nokta mutasyon yöntemi uygulanmıştır. Rastgele sayı, kromozomun her geni için hesaplanır ve mutasyon olasılığı bu rasgele sayıdan büyük olduğunda mutasyon gerçekleştirilir. Mutasyon olasılığı, $P_m = 0,05$ 'te düşük tutulur ve mutasyona uğramış gen, hesaplanan ikinci bir rasgele sayıya kıyasla 1 artar veya azalır. Bu değer mod 23'e göre işlenir ve değerlendirilir. Örneğin gen 22 ise ve 1 artırılabırsa genin yeni değeri 1'dir. İterasyon sayısı 300 olarak kabul edilmiştir. Nüfus büyüklüğü 400 kromozomdur. $P_{crossover}$ parametresi 0,90 kabul edilmiştir.

Genetik Algoritma Matlab R17'de yazılan kod ile çalıştırılmıştır. Çalışma sonucunda 13 OKA ile biten denemelere devam edilmesi ile 12 OKA sonuçları da elde edilmiştir. Mevcut durumda kullanılan 17 OKA yerine 12 OKA elde edilmesi %29 luk bir iyileşmeye karşılık gelmektedir. 12 OKA gereksinimi sağlayan çözüm aşağıda gösterilmiştir.

1 20 15 22 5 15 7 8 20 11

Genetik algoritma çalışırken iterasyonlar boyunca elde edilen sonuçlar Şekil 4.8'de gösterilmiştir.



Şekil 4.8. İterasyonlar süresinde elde edilen gerekli OKA miktarı

Gerekli 12 OKA sayısını sağlayan diğer kromozomlar Çizelge 4.8’de gösterilmektedir.

Çizelge 4.8 OKA sayısını sağlayan diğer kromozomlar

1. hat	2. hat	3. hat	4. hat	5. hat	6. hat	7. hat	8. hat	9. hat	10. hat
1	22	22	21	16	14	7	8	18	14
1	22	15	21	16	14	7	8	14	14
20	22	17	22	5	12	7	8	20	11
21	20	22	15	16	13	7	8	21	15
21	20	19	22	16	12	7	8	12	13
1	2	17	22	16	16	7	8	13	12
1	2	19	17	5	11	7	8	16	16

Çizelge 4.8 incelendiğinde 12 OKA gereksinimi sağlayan çözümlerden arasından en uygunu seçmek gerekmektedir. Ayrıca OKA hatlarına atanan park istasyonlarının da bazı değerlere odaklanması istenmektedir. Bu sebeple bu çözümlere ilave bir kriter getirilmiştir. Hizmet süresi uygunluk fonksiyonuna aşağıdaki şekilde dahil edilmiştir.

$$\text{Uygunluk Fonksiyonu} = \text{Gerekli OKA sayısı} \times 10000 + \text{Hizmet Süresi} \div 1000$$

İterasyon sayısı 1000'e çıkarılarak genetik algoritma Matlab R17'de yeniden çalıştırılmıştır. Bu çalışmada 12 OKA sayısını gerektiren kromozom değerleri Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9. 12 OKA sayısını gerektiren kromozom değerleri

1.hat	2.hat	3.hat	4.hat	5.hat	6.hat	7.hat	8.hat	9.hat	10.hat
1	2	3	21	16	16	7	8	19	15
1	2	3	4	5	16	7	8	19	16
1	2	3	4	16	15	7	8	16	15
1	21	19	4	5	6	7	8	15	16
1	2	20	21	16	16	7	8	16	15
1	2	21	4	5	6	7	8	16	15
1	2	3	22	5	16	7	8	20	15
21	2	3	22	5	6	7	8	16	12

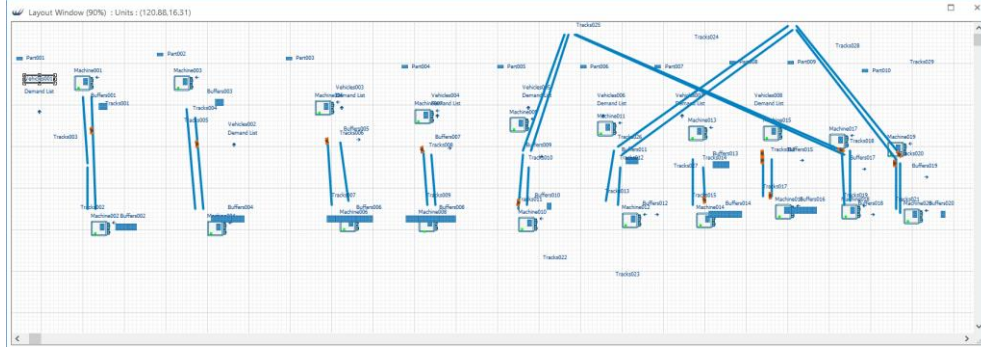
Önerilen algoritmanın birçok kez çalıştırılmasıyla elde edilen çözümler incelendiğinde artık OKA hatlarına atanan park istasyonları belli değerlere odaklanmıştır ve havuza girmeyen OKA hatları kendi üretim postalarına atanmış ve bu sonuç hizmet süresinin eklenmesiyle elde edilmiştir. Yukarıdaki yaklaşımlara göre, önerilen çözüm aşağıda gösterilmiştir:

1-2-3-4-16-15-7-8-16-15.

Bu çözümle gerekli OKA sayısında %29 luk bir iyileşme ile sağlanmıştır. Kazanılacak 5 OKA, ileride başka projelerde kullanılabilir. Bu çözüm geliştirilen kesikli olay simülasyonu ile Witness 22 b de denenmiş ve üretim isteklerinin bu çözüm ile karşılandığı doğrulanmıştır.

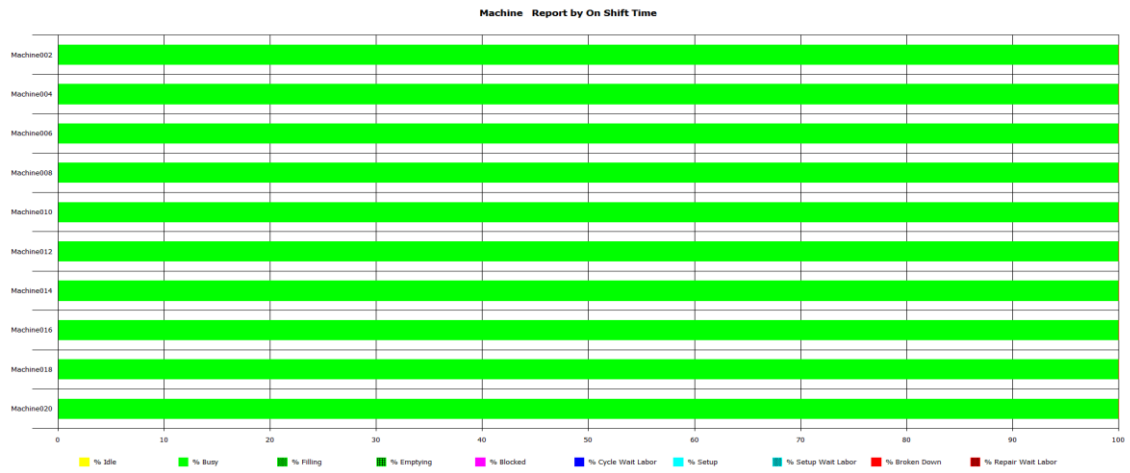
Benzetim What IF analizleri için uygun bir ortam sağlamaktadır. Benzetim modeli oluşturulduktan sonra sistem parametreleri değiştirilmesi ile birçok soruya cevap bulunabilir. Diğer bir konu da benzetim sayesinde model çalışma gruplarındakilere anlatılıp çalışma biçimiyle ilgili görüşleri alınıp gerekli değişiklikler ve senaryo analizleri

kolaylıkla gerçekleştirilebilir. Algoritmanın çalıştırılması ile elde edilen sonuçların üretimde kesinti yaratıp yaratmadığının kontrolü geliştirilen benzetim modeli ile yapılmıştır. Witness ekran görüntüsü Şekil 4.9'da gösterilmektedir.



Şekil 4.9. Witness ekran görüntüsü

OKA hızı, postalar arası mesafe, şarj süreleri model formülasyondaki belirtilen değerler ile benzetim modeli çalıştırılmıştır. 10 üretim ve 10 tüketim postası oluşturulmuştur. 5. ve 9. OKA hatları 16 nolu park istasyonuna atanmış; 6. ve 10. OKA hatları 15 nolu park istasyonuna atanmıştır. Öncelikli hedef OKA taşıma kaynaklı makine duruşları olmamasıdır. Model 100 dakika ısındırma periyodunda çalıştırılmıştır. Model 1100 dakika çalıştırılması sonucunda aşağıdaki sonuçlar elde edilmiştir Sistem için gerekli OKA miktarı algoritma ile belirlendiği gibi 12'dir. Tüm tüketim postalarındaki makineler %100 olarak çalışmıştır. Taşıma kaynaklı tesis duruşu olmadığını gösterir. Tüketim postalarının doluluk oranı Şekil 4.10'da gösterilmiştir.



Şekil 4.10. Makina doluluk oranları

Makinaların doluluk oranı Çizelge 4.10’da gösterilmiştir. Makina doluluk oranlarının %100 olması, makinaların hiç durmadığını dolayısıyla OKA kaynaklı duruş olmadığını göstermektedir.

Çizelge 4.10 Makinaların doluluk oranları

Makinalar	% Boşluk	% Doluluk	% Bloke Olmuş	% Setup	% Arıza
Tüketim Postası_1	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_2	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_3	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_4	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_5	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_6	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_7	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_8	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_9	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_10	0.000	100.000	0.000	0.000	0.000

2. gözlem OKA verimlilikleri ile ilgilidir. 8 nolu OKA hattı 3 OKA ile dönmektedir. Bununla birlikte araçların yaklaşık % 45 i boş görünmektedir. Bu hat 2 OKA ile dönme ihtimalini göstermektedir. Ama yapılan 2 OKA’lı denemelerde üretimde kopmalar gerçekleşmiştir. Araç doluluk oranları Çizelge 4.11’de gösterilmektedir.

Çizelge 4.11 Araç doluluk oranları

OKA	% Boş	% Transfer	% Yüklü	% Stop
OKA_001	16,8	43,8	32,0	7,5
OKA_002	10,3	49,0	33,3	7,5
OKA_003	10,9	75,7	8,4	5,0
OKA_004	13,2	72,0	11,3	3,5
OKA_005(1)	28,0	44,8	20,5	6,8
OKA_005(2)	20,8	54,0	19,7	5,6
OKA_006(1)	51,5	33,4	8,8	6,3
OKA_006(2)	60,8	26,7	6,2	6,3
OKA_007	47,7	23,4	20,5	8,4
OKA_008(1)	46,1	30,6	16,2	7,1
OKA_008(2)	45,7	30,7	16,5	7,1
OKA_008(3)	45,7	30,9	16,2	7,2

Geliştirilen modelin başka durumlarda da test edilmesi için aşağıda sunulan deneme seti yaratılmıştır. 3 OKA hattı ve 5 park istasyonu ile başlayan denemelerde OKA hat sayısı, taşıma arabalarının kapasitesi, hat çevrim süresi ve mesafeler değiştirilerek geliştirilen model uygulanmıştır. 9 farklı durum oluşturulmuştur. Deney Seti Çizelge 4.12’de gösterilmektedir.

Çizelge 4.12 Deney seti

Data No	Çev.Sr. (parç/saat)	OKA hattı	Taşıma Kapasite	a (mm)	b (mm)	Data No	Çev.Sr. (parç/saat)	OKA hattı	Taşıma Kapasite	a (mm)	b (mm)	Data No	Çev.Sr. (parç/saat)	OKA hattı	Taşıma Kapasite	a (mm)	b (mm)
1	60	1	30	20	22	2	30	1	30	20	22	3	80	1	30	20	22
		2	20	8	10			2	20	8	10			2	20	8	10
		3	10	4	6			3	10	4	6			3	10	4	6
4	60	1	30	20	22	5	60	1	30	20	22	6	60	1	30	20	22
		2	20	8	10			2	20	8	10			2	20	8	10
		3	10	4	6			3	10	4	6			3	10	4	6
		4	60	20	22			4	60	20	22			4	60	20	22
								5	80	30	32			5	80	30	32
														6	100	40	42
7	60	1	30	20	22	8	45	1	27	20	22	9	20	1	30	20	22
		2	20	8	10			2	13	8	10			2	20	8	10
		3	10	4	6			3	14	4	6			3	10	4	6
		4	60	20	22			4	23	20	22			4	60	20	22
		5	80	30	32									5	80	30	32
		6	100	40	42												
		7	120	60	62												

Yukarıdaki deney setinin geliştirilen yöntemle çalıştırılması sonucu Çizelge 4.13'te gösterilen sonuçlar elde edilmiştir.

Çizelge 4.13 Deney seti sonuçları

Deneme	Deney Seti Set	OKA Hatları	Park İstasyonları	Çevrim Süresi	Müsait Park İstasyonları	Gerekli OKA sayısı	OKA Hatları							Açıklama			
							1	2	3	4	5	6	7				
1	1	3	5	60	8,9	5	1	2	3								
2	2	3	5	30	8,9	3	1	2	3								
3	3	3	5	80	8,9	6	1	2	3								
4	4	4	6	60	8,9	6	1	2	3	4							
5	4	4	6	60	10,11	6	1	2	3	4							
6	4	4	6	60	12,13	6	1	2	3	4							
7	5	5	6	60	8	7	1	2	3	4	5						
8	5	5	7	60	8,9	7	1	2	3	4	5						
9	5	5	8	60	8,9,10	7	1	2	3	4	5						
10	6	6	9	60	8,9,10	8	1	2	3	4	5	6					
11	7	7	9	60	8,9	9	1	2	3	4	5	6	7				
12	7	7	10	60	8,9,10	9	1	2	3	4	5	6	7				
13	7	7	11	60	8,9,10,11	9	1	2	3	4	5	6	7				
14	7	7	12	60	8,9,10,11,12	8	1	2	3	4	12	12	12				
15	7	7	13	60	8,9,10,11,12,13	8	1	2	3	4	12	12	12				
16	7	7	13	60	8,9,10,11,12,13	8	1	2	3	4	13	13	13	a			
17	7	7	13	60	8,9,10,11,12,13	8	1	2	3	4	13	13	13	b			
18	7	7	13	60	8,9,10,11,12,13	8	1	2	3	11	5	11	11				
19	7	7	13	60	8,9,10,11,12,13	7	10	10	3	11	10	11	11	c			
20	8	4	6	45	8,9	5	6	2	6	6							c
21	9	5	8	20	8,9,10	5	1	2	3	4	5						

a:V 25m/dak b:V 25m/dak Mesafe/2 c:Mesafe/4

İlk 13 denemede OKA havuzu gerekli değildi. Tüm OKA hatları üretim istasyonunu park istasyonu olarak kullanmıştır. 14. denemede 12 numaralı park istasyonunda bir havuz oluşturulmuştur ve 5, 6 ve 7 OKA hatları bu havuzu kullanmaktadır. 19. denemede ilk havuz 10 nolu park istasyonunda olmak üzere iki havuz oluşturulmalı ve OKA hatları 1, 2 ve 5 bu havuza atanmalı ve ikinci havuz 11 nolu park istasyonunda oluşturulmalıdır. OKA hatları 4, 6 ve 7 ikinci havuza atanmalıdır. 3 no lu OKA hattı herhangi bir havuza atanmamıştır. Ardından Witness 22.b sistemi üzerinde simülasyon modeli çalıştırılarak OKA kullanımı dolayı üretim kaybı olup olmayacağı kontrol edilmiştir ve üretim kaybı olmadığı gösterilmiştir.

Simülasyon çalışması:

18 no'lu deney çalışması ele alınmıştır. Havuz sistemi olmaması durumunda gerekli OKA miktarı Şekil 4.10'da gösterildiği gibi 9 adet olmalıdır. Witness programında gerçekleştirilen model Şekil 4.11'de gösterilmektedir.



Şekil 4.6 Senaryo 1 için Witness ekran görüntüsü

7 üretim istasyonu ve 7 tüketim istasyonu bulunmaktadır. Her OKA hattı kendi üretim ve tüketim postaları arasında çalışmaktadır. 9 OKA ile çalışılması durumunda, tüketim postalarında bir duruş olmamaktadır. Bu durumda imalat kaynaklı bir kayıp yaşanmamaktadır. Senaryo 1 için makine doluluk oranları Çizelge 4.14'teki gibidir.

Çizelge 4.14 Senaryo 1 için makine doluluk oranları

Makinalar	% Boşluk	% Doluluk	% Bloke Olmuş	% Setup	% Arıza
Tüketim Postası_1	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_2	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_3	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_4	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_5	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_6	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_7	0.000	100.000	0.000	0.000	0.000

OKA'ların etkinlik durumu aşağıdaki tabloda gösterilmiştir. Toplam 9 adet OKA gereklidir. Senaryo 1 için OKA Doluluk Oranları Çizelge 4.15'teki gibidir.

Çizelge 4.15. Senaryo 1 için OKA doluluk oranları

OKA	% Boş	% Transfer	% Yüklü	% Durma
OKA_001	29	39	26	7
OKA_002 (1)	31	46	18	5
OKA_002 (2)	31	46	18	5
OKA_003 (1)	59	14	16	10
OKA_003 (2)	59	15	16	10
OKA_004	8	81	8	3
OKA_005	5	88	5	3
OKA_006	4	91	4	2
OKA_007	3	92	3	2

Havuz sistemine geçiş durumunda algoritmanın hesapladığı gerekli OKA sayısı 8 dir. OKA hattı 5, 6, 7; 13 numaralı park istasyonuna atanması ile modelin görünümü Şekil 4.12’de gösterilmektedir. Diğer OKA hatları kendi üretim istasyonunda beklemektedir.



Şekil 4.12. Senaryo 2 için Witness ekran görüntüsü

8 OKA ile tüketim postaları imalata kesintisiz devam edebilmektedir. Senaryo 2 için makina doluluk oranları Çizelge 4.16’daki gibidir. Tüketim istasyonlarında bekleme olmamıştır. Bu sonuç OKA taşıma sisteminin üretimi aksatmadan parçaların taşındığını göstermektedir.

Çizelge 4.16 Senaryo 2 için makina doluluk oranları

Makinalar	% Boşluk	% Doluluk	% Bloke Olmuş	% Setup	% Arıza
Tüketim Postası_1	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_2	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_3	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_4	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_5	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_6	0.000	100.000	0.000	0.000	0.000
Tüketim Postası_7	0.000	100.000	0.000	0.000	0.000

Senaryo 2 İçin OKA doluluk oranları Çizelge 4.17’de gösterilmektedir.

Çizelge 4.17 Senaryo 2 için OKA doluluk oranları

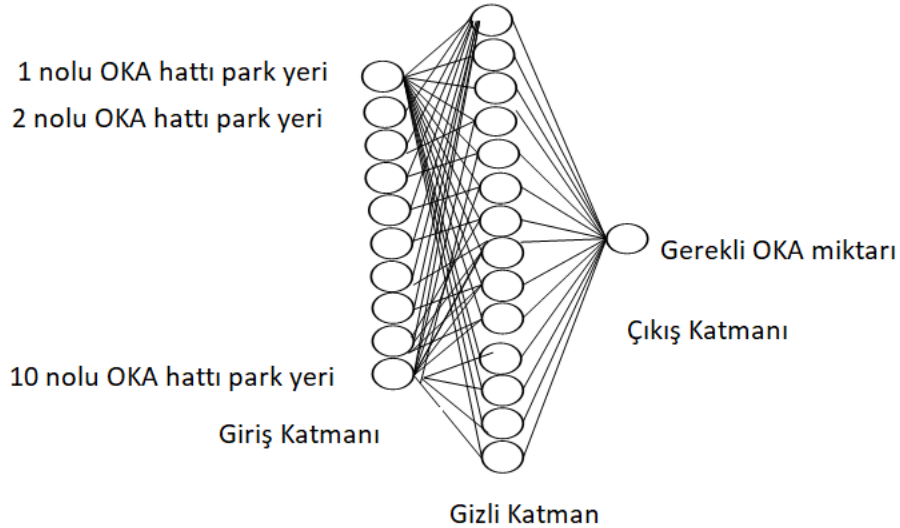
OKA	% Boş	% Transfer	% Yüklü	% Durma
OKA_001	29	39	26	7
OKA_002 (1)	31	46	18	5
OKA_002 (2)	31	46	18	5
OKA_003 (1)	59	14	16	10
OKA_003 (2)	59	15	16	10
OKA_004	8	81	8	3
OKA_005(1)	89	3	5	3
OKA_005(2)	88	3	6	3

GA sonuçlarına göre 8 OKA gereklidir. Son 2 OKA’nın etkinliği çok az olmak ile birlikte 1 OKA kalması durumunda imalat kesintiye uğramaktadır. Bu sebeple, OKA sayısı 8 olması gerekmektedir. Bu sistemde OKA havuz sistemine geçiş ile %11,1’lik bir iyileşme sağlanacağı benzetim tekniği ile gösterilmiştir.

4.5. Yapay Sinir Ağları ile Gerekli Optimum OKA Sayısının Tahmin Edilmesi

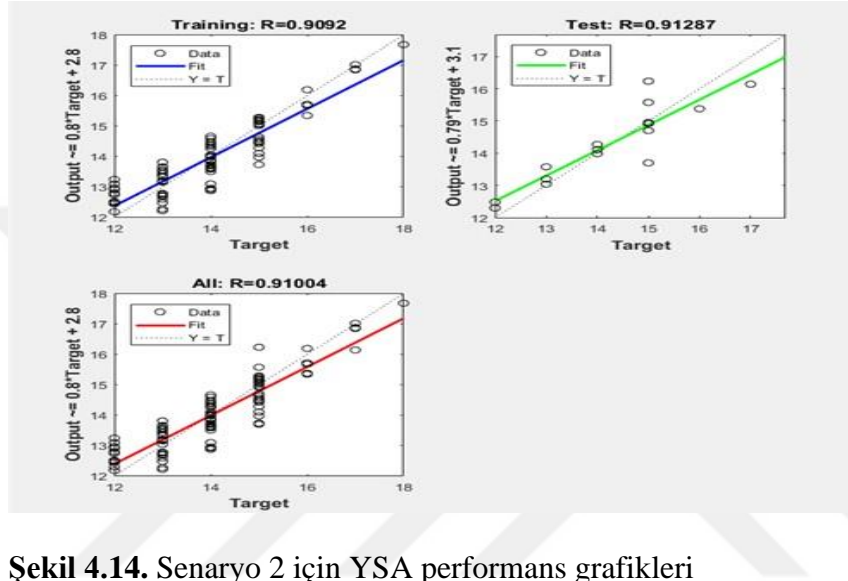
YSA, veri setlerinden öğrenebilmekte ve doğrusal olmayan tüm fonksiyonlara pratik olarak yaklaşabilmektedir. Bir sinir ağı, katmanlarda yer alan nöronlar olarak adlandırılan birbirine bağlı öğelerin veri odaklı bir modelidir. İleri beslemeli sinir ağı, en popüler sinir ağı mimarisidir. İleri beslemeli bir ağ, bilgi veya sinyallerin girişten çıkışa yalnızca bir yönde gönderildiği bir ağıdır. Geri yayılım yöntemi, keyfi kategori eşlemesini öğrenebildiği için yaygın olarak kullanılırken, diğer yöntemler gürültülü eğitim verileri veya sınırlı eğitim verileri durumunda YSA performansını geliştirebilmektedir.

OKA hat kombinasyonlarının park istasyonlarına göre gerekli OKA miktarının zaman pencere yaklaşımı için hesaplanmasında tüm durumların hesaplanması gerekmektedir. OKA hat sayısı 10 ve 22 park istasyonu için algoritma çalışmasında çözüm kabul edilebilir sürede elde edilebilmektedir. Bununla birlikte hat sayısının artması hesaplanması gerekli durum sayısını üstel olarak arttırmaktadır. Hesaplama süresini kısaltacak bir tahminleyici bu sorunun çözümüne katkı sağlayacaktır. Şekil 4.13'te gösterildiği gibi 10 OKA hattına atanan park istasyon numaralarını girdi olarak kabul ederek gerekli OKA miktarını hesaplayan bir yapay sinir ağı modeli geliştirilmiştir. 14 nöronlu gizli katmana sahip olan model, 10 girdi ve 1 çıkışa sahiptir.



Şekil 4.13. Senaryo için YSA mimarisi

Şekil 4.13' te çıktı olarak gerekli OKA miktarı tahmin edilmeye çalışılmıştır. Levenberg-Marquardt ve Bayesian Regularization'ın eğitim algoritmaları, sinir ağı modelini eğitmek için kullanılmıştır. Matlab 17R'de YSA modeli oluşturulmuştur. 110 rastgele seçilmiş veri ile çalıştırılan model, en iyi sonuçlar bayes düzeltilmesiyle %85 öğrenme oranıyla elde edilmiştir. Model R^2 değeri 0,91 olan doğruluk oranlarına sahiptir. Geliştirilen modelin tahminleme performansı Şekil 4.14'te gösterilmiştir.



Şekil 4.14. Senaryo 2 için YSA performans grafikleri

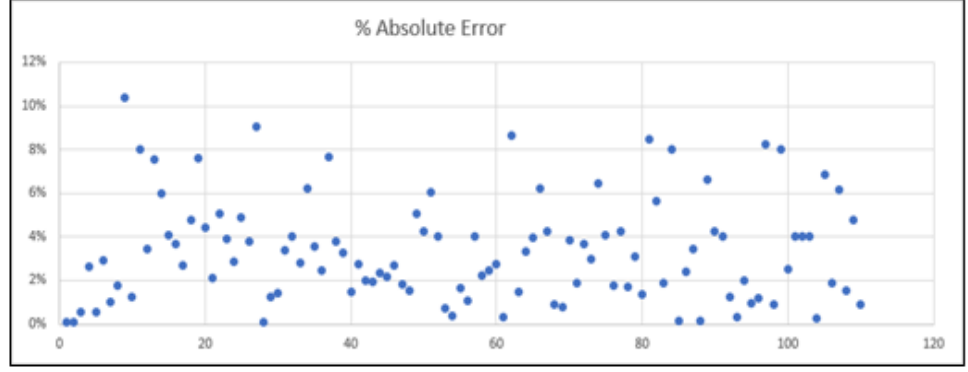
Yapay sinir ağları ile tahmin ve önerilen algoritma ile elde edilen değerler Şekil 4.14'te gösterilmektedir. Eğitimde $R:0,9092$ ve testte $R:0,91$ elde edilmiştir. Bu sonuçlar YSA'nın başarılı bir tahminleyici olduğunu göstermektedir. YSA çıktıları ile GA'nın hesapladığı değerlerin karşılaştırılması Çizelge 4.18'deki gibidir. GA ile hesaplanan OKA miktarları ile Yapay Sinir Ağları ile tahminlenen OKA miktarlarını karşılaştırmak için mutlak hata yüzdeleri hesaplanmıştır.

Çizelge 4.18. YSA tahminleri ile GA sonuçlarının karşılaştırılması

Veri Numarası	Gerekli OKA	YSA Tahmini	Veri Numarası	Gerekli OKA	YSA Tahmini	Veri Numarası	Gerekli OKA	YSA Tahmini	Veri Numarası	Gerekli OKA	YSA Tahmini	Veri Numarası	Gerekli OKA	YSA Tahmini
1	15	15,0	23	16	15,4	45	14	13,7	67	12	12,5	89	12	12,8
2	17	17,0	24	14	13,6	46	14	13,6	68	14	13,9	90	12	12,5
3	15	14,9	25	13	13,6	47	14	14,3	69	14	14,1	91	12	12,5
4	14	13,6	26	15	14,4	48	13	13,2	70	15	15,6	92	14	13,8
5	15	15,1	27	12	13,1	49	17	16,1	71	15	15,3	93	14	14,0
6	15	14,6	28	14	14,0	50	13	13,6	72	12	12,4	94	16	15,7
7	15	15,2	29	15	15,2	51	13	12,2	73	14	13,6	95	15	15,1
8	16	15,7	30	15	15,2	52	12	12,5	74	14	13,1	96	16	16,2
9	12	13,2	31	13	13,4	53	17	16,9	75	16	15,3	97	15	16,2
10	13	13,2	32	12	12,5	54	15	15,1	76	18	17,7	98	17	16,9
11	14	12,9	33	14	14,4	55	13	12,8	77	14	14,6	99	14	12,9
12	14	14,5	34	13	13,8	56	14	13,8	78	15	15,3	100	13	13,3
13	14	12,9	35	14	13,5	57	12	12,5	79	13	12,6	101	12	12,5
14	15	14,1	36	15	14,6	58	13	12,7	80	13	13,2	102	12	12,5
15	13	12,5	37	14	12,9	59	12	12,3	81	15	13,7	103	12	12,5
16	12	12,4	38	15	14,4	60	13	13,4	82	13	12,3	104	13	13,0
17	14	14,4	39	14	14,5	61	15	15,0	83	14	13,7	105	15	14,0
18	14	14,7	40	13	13,2	62	15	13,7	84	12	13,0	106	13	12,8
19	12	12,9	41	14	13,6	63	12	12,2	85	14	14,0	107	12	12,7
20	13	13,6	42	15	14,7	64	15	14,5	86	13	12,7	108	13	13,2
21	14	13,7	43	14	14,3	65	13	13,5	87	15	14,5	109	15	14,3
22	13	13,7	44	14	14,3	66	13	13,8	88	14	14,0	110	14	13,9

Mutlak yüzde hata grafiği Şekil 4.15'teki gibidir. Yatay eksen tahminleri; dikey eksen ise ilgili mutlak yüzde hatayı göstermektedir.

Mutlak Yüzde Hata



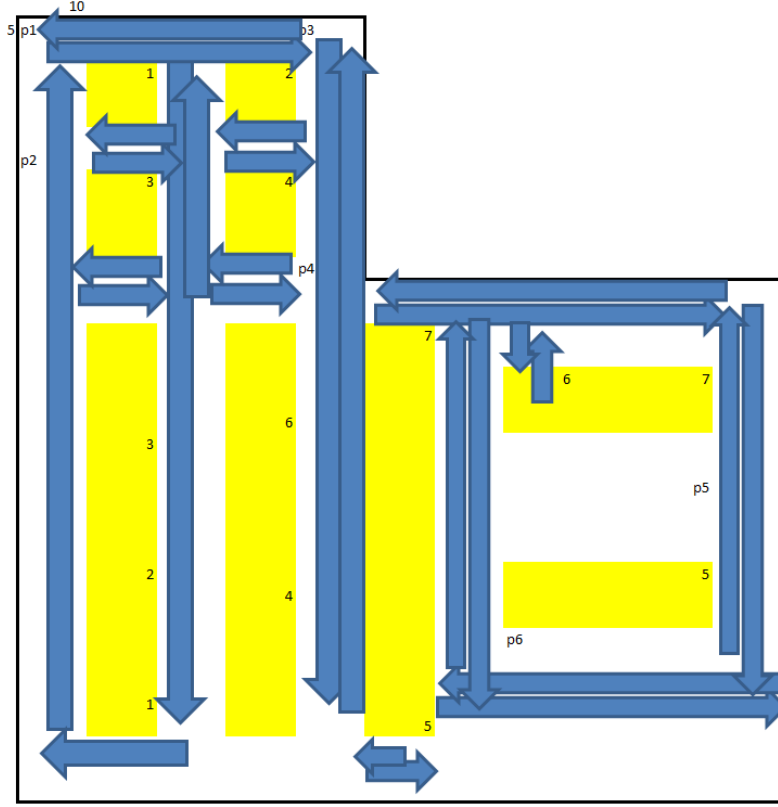
Denemeler

Şekil 4.15. Mutlak yüzde hata grafiği

Mutlak yüzde hata grafiği incelendiğinde sapmanın %2 ile %4 aralığında yoğunlaştığı gözlenmektedir. Bu tespit ile elde edilen 0,91 korelasyon katsayısı OKA hat kombinasyonuna atanan park istasyonlarının gerektirdiği OKA miktarının hesaplanmasında yapay sinir ağlarının güçlü bir tahminleyici olduğunu göstermektedir. Özellikle artan OKA hatlarında kullanılması hesaplama zamanında kazanımlar sağlaması beklenmektedir.

4.6. Değişen Sistem Parametrelerinin Etkilerinin Analizi

Geliştirilen model sabit sistem parametreleri kabulü altında çalışmaktadır. Sistem parametreleri kurulum sonrası değişirse gerekli OKA miktarının hesaplanması konusunda karar vericilere yol gösterecek bir yaklaşım gerekmektedir. Diğer taraftan gerekli OKA miktarını azaltmak için hangi parametrelerin değerinin değiştirilmesi sorusunun cevabı da aynı metot ile hesaplanabilmektedir. Analiz için deneme setindeki 17 no'lu örnek ele alınmış olup bu deneme için yerleşim planı Şekil 4.16'da gösterilmektedir.



Şekil 4.16. Deney setlerine ilişkin yerleşim planı

Sistemde 7 üretim istasyonu ve 13 park istasyonu bulunmaktadır. Sistem parametrelerinin gerekli OKA sayısı üzerine etkisi incelenmektedir. Sistem parametreleri olarak: ortalama Hız (m/dk), şarj ve diğer bekleme süreleri (Dakika), mesafe, taşıma arabasının kapasitesi belirlenmiştir. Her bir parametre için minimum, ortalama ve maksimum değerler Çizelge 4.19’da gösterilmektedir. Belirlenen 4 sistem değişkeni için yüzey yanıt deney tasarımı gerçekleştirilmiştir.

Box ve Behnken yöntemi, yanıt yüzeylerini oluşturmak amacıyla 2k faktöriyelerin tamamlanmamış blok tasarımlarının birleştirilmesiyle oluşturulmuştur. Tüm blok tasarımı yerine azaltılmış deneme sayıları ile verimli deney tasarım elde edilmiştir. Box Behnken yöntemi kullanılmış olup, 4 faktör, 1 tekrar olmak üzere 27 deneme gerçekleştirilmiştir (Montgomery, 2005).

Çizelge 4.19. Faktörler ve seviyeleri

	Hız	Şarj ve diğer bekleme süreleri	Mesafe	PS
Min	10	4	0.5	0.5
Ort	20	6	1	1
Max	30	8	1.5	1.5

Deneme değerlerinin önerilen yöntem için çalıştırılması ile elde edilen gerekli OKA sayıları Çizelge 4.20’de gösterilmektedir. Gerekli OKA sayısı GA çalıştırılması ile elde edilmiştir.

Yüzey Yanıt Metodu (YYM), fonksiyonu genelde bilinmeyen bir bağımsız değişkenin diğer bağımsız değişkenler ile değişiminin açıklanmasıdır. Polinom açılımından yararlanarak bir ilgilenilen bağımlı değişken için bir model geliştirilip, bağımlı değişkenin bağımsız değişkenlerine göre değerinin belirlenip, diğer bağımsız değişkenlere göre nasıl değiştiğini öngörmektir (Montgomery, 2005). Yüzey yanıt grafikleri de değişim öngörülerinde kullanılmaktadır.

Oluşturulan modelin geçerliliğinin kontrol edilmesi de önemli bir adımdır. Geliştirilen Yüzey Yanıt Metodunun Minitab 19’da oluşturulmuş ve Çizelge 4.21’de gösterilen Varyans Analizi Tablosu elde edilmiştir.

Çizelge 4.20. Deney tasarımı tablosu

	Hız	Şarj Süresi	Mesafe	Parça Sayısı	Gerekli OKA Sayısı
1	10	8	1	1	13
2	20	8	0.5	1	7
3	30	8	1	1	8
4	10	6	0.5	1	9
5	20	6	1.5	1.5	8
6	20	6	0.5	1.5	5
7	10	6	1	0.5	20
8	10	6	1	1.5	9
9	20	6	1	1	9
10	20	6	1.5	0.5	17
11	10	4	1	1	13
12	10	6	1.5	1	16
13	20	4	1	1.5	7
14	20	8	1.5	1	12
15	30	6	0.5	1	5
16	30	6	1	0.5	11
17	30	6	1	1.5	6
18	20	6	1	1	9
19	20	8	1	1.5	7
20	20	4	1	0.5	13
21	30	6	1.5	1	9
22	20	4	0.5	1	5
23	20	6	1	1	9
24	20	4	1.5	1	10
25	20	6	0.5	0.5	9
26	30	4	1	1	7
27	20	8	1	0.5	14

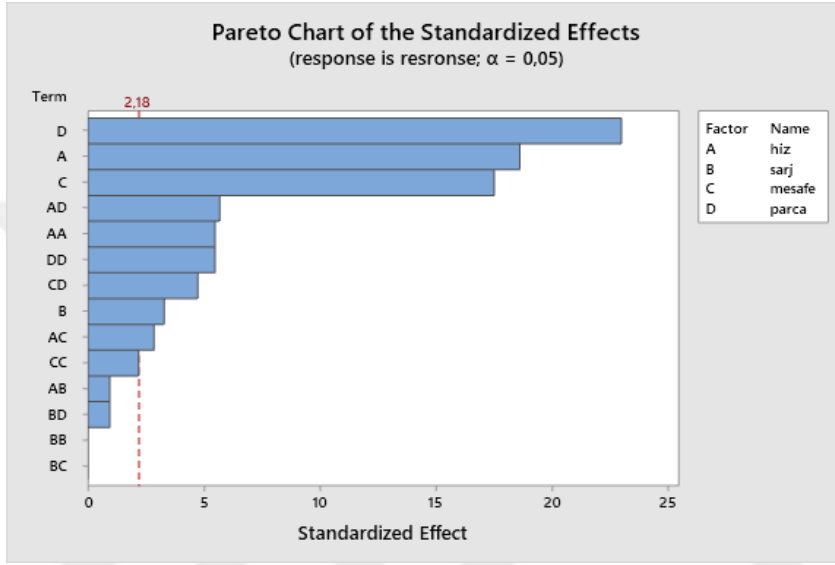
Çizelge 4.21.Varyans analizi tablosu

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Model	14	371,333	26,524	95,49	0,000
Linear	4	331,667	82,917	298,50	0,000
Hiz	1	96,333	96,333	346,80	0,000
Sarj	1	3,000	3,000	10,80	0,007
Mesafe	1	85,333	85,333	307,20	0,000
Parça	1	147,000	147,000	529,20	0,000
Square	4	21,667	5,417	19,50	0,000
hiz*hiz	1	8,333	8,333	30,00	0,000
sarj*sarj	1	0,000	0,000	0,00	1,000
mesafe*mesafe	1	1,333	1,333	4,80	0,049
parca*parça	1	8,333	8,333	30,00	0,000
2-Way Interaction	6	18,000	3,000	10,80	0,000
hiz*sarj	1	0,250	0,250	0,90	0,361
hiz*mesafe	1	2,250	2,250	8,10	0,015
hiz*parça	1	9,000	9,000	32,40	0,000
sarj*mesafe	1	0,000	0,000	0,00	1,000
sarj*parça	1	0,250	0,250	0,90	0,361
mesafe*parça	1	6,250	6,250	22,50	0,000
Error	12	3,333	0,278		
Lack-of-Fit	10	3,333	0,333	*	*
Pure Error	2	0,000	0,000		
Total	26	374,667			

Model, parametrelerin sistem üzerine etkisini başarılı şekilde açıklamaktadır. R^2 değeri 99,11 ve düzeltilmiş R^2 değeri 98,07 elde edilmiştir. P değerleri ayrıca 0,05'ten küçük elde edilmiştir. Yüzey Yanıt Metoduna göre gerekli OKA sayısı ikinci dereceden polinom şeklinde ifade edilebilir. Faktörler kodlanmamış olarak bağımlı OKA sayısı, bağımsız değişkenler olan hız, mesafe, şarj süresi ve taşıma kapasitesi olarak denklem 4.18'deki gibi ifade edilebilir (faktörler kodlanmamıştır):

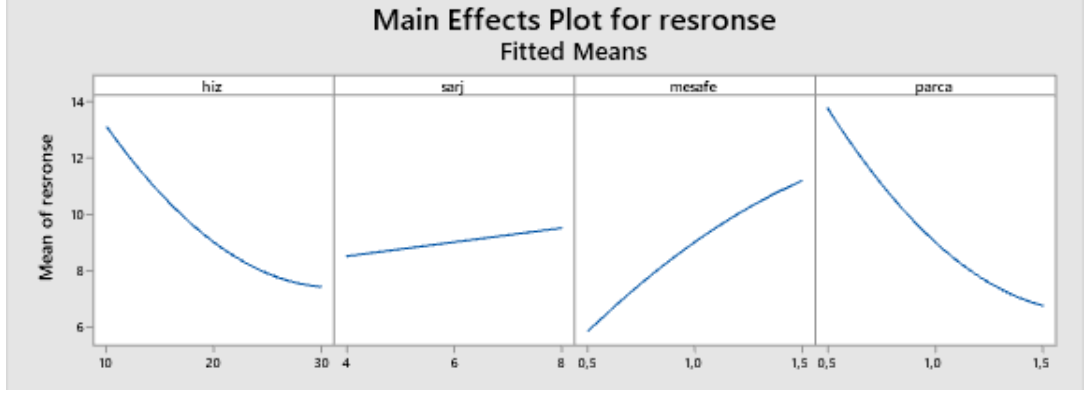
$$\begin{aligned}
\text{Gerekli} &= 20,83 - 1,008 \text{ hiz} + 0,250 \text{ sarj} + 17,33 \text{ mesafe} - 16,50 \text{ parca} \\
\text{OKA} &+ 0,01250 \text{ hiz} \times \text{hiz} - 0,0000 \text{ sarj} \times \text{sarj} - 2,000 \text{ mesafe} \times \text{mesafe} \\
\text{sayısı} &+ 5,000 \text{ parca} \times \text{parca} - 0,0125 \text{ hiz} \times \text{sarj} - 0,1500 \text{ hiz} \times \text{mesafe} \\
&+ 0,3000 \text{ hiz} \times \text{parca} + 0,000 \text{ sarj} \times \text{mesafe} - 0,250 \text{ sarj} \times \text{parca} \\
&- 5,00 \text{ mesafe} \times \text{parca} \quad (4.18)
\end{aligned}$$

Standart etkilerin pareto diyagramı Şekil 4.17’de gösterilmektedir.



Şekil 4.17. Standart etkilerin pareto diyagramı

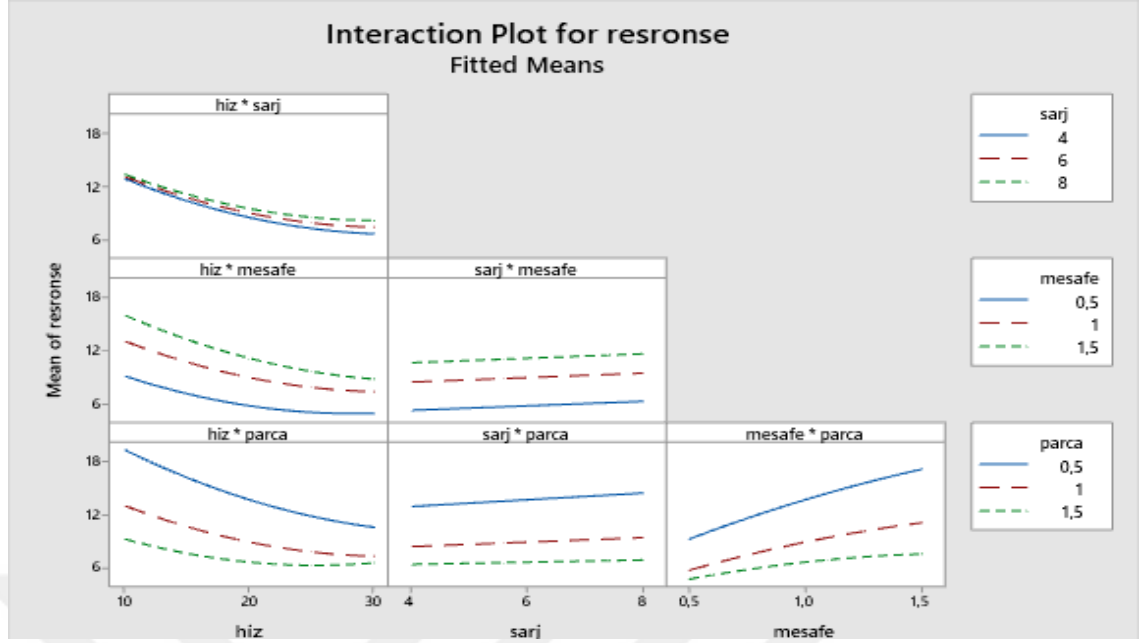
Standardize faktörlerin pareto tablosu incelendiğinde OKA sayısına en önemli etki taşıma arabalarının kapasitesi oluşturmaktadır. Hız ve mesafe 2. ve 3. önemli faktörlerdir. Faktörlerin etki grafiği Şekil 4.18’de gösterilmektedir. Grafiğin incelenmesi ile aşağıdaki sonuçlar çıkarılabilir: OKA hızının artışı ve taşınan parça adetinin artışı gerekli OKA sayısını azaltmaktadır. Şarj süresi ve mesafelerin artışı gerekli OKA sayısını arttırmaktadır. Şarj süresi ile gerekli OKA sayısı lineere yakın bir ilişki göstermektedir. Diğer parametrelerin etkisi hiperbolik fonksiyona yakın bir değişim göstermektedir.



Şekil 4.18. Faktörlerin etki grafiği

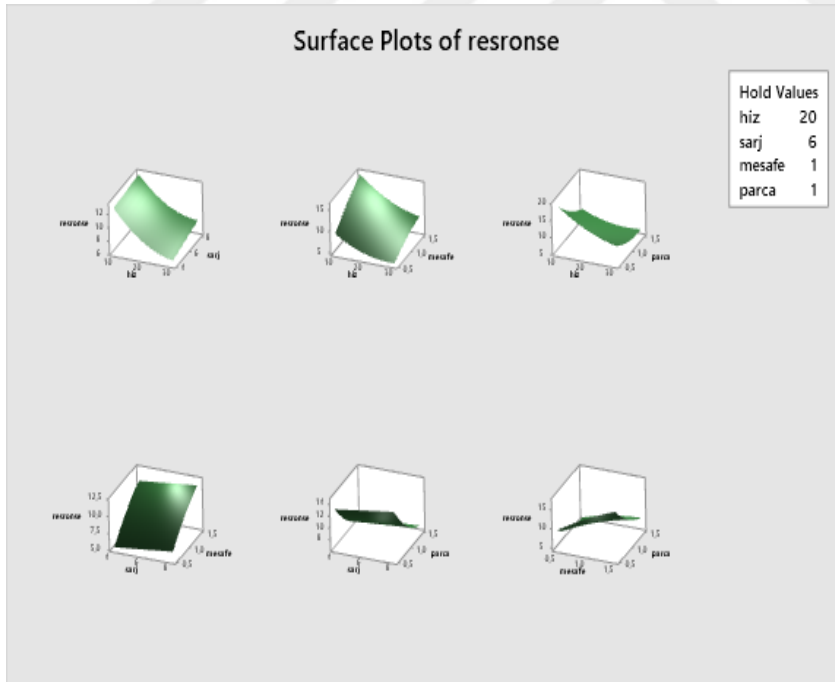
Özellikle OKA hızı genellikle her OKA projesinde tartışma konusu olmaktadır. OKA hattının belirli bir kısmında OKA hızı genellikle imalat ekipleri tarafından arttırılmak istenmekte, iş güvenlik yetkililerince azaltılmak istenmektedir. OKA'nın hızının arttırılması OKA çalışma güvenilirliğini olumsuz etkilemektedir. Hız, OKA sayısının belirlenmesinde en etkili parametrede değildir. En etkili faktör olan taşıma arabası kapasitesi, gerekli OKA sayısının azaltılmasında ele alınması gereken bir konudur. Taşıma arabasının ağırlığı ergonomik çerçeve içinde olmalı ve postalarda yerleşim kısıtı yaratmaması dikkate alınması gereken hususlardır. Bir diğer yaklaşım şarj sürelerinde yaşanmaktadır. Özellikle şarj süresinin kısaltılması çokça talep edilen bir konudur. Şarj süresi diğer faktörlere göre OKA sayısına etki eden en düşük önemdeki faktördür. OKA'nın yeterli şarj olamaması OKA bataryasının bitip sistem güvenilirliğini azaltacaktır. Mesafelerin ele alınması yüksek maliyet sebebiyle genelde düşünülmez ama ilk proje kurulumunda yerleşimin OKA projelerini de düşünülerek yapılması OKA'ların gereksiz hareketlerini ortadan kaldıracaktır.

İkili faktörlerin etkilerinin birlikte incelenmesi için interaksiyon diyagramları Şekil 4.19'da gösterilmiştir. Şarj×Parça ve Şarj ×mesafe diyagramlarındaki değişim doğrusal olarak gözlenmektedir, diğer etkiler hiperboliğe yakın bir değişim göstermektedir. Benzer sonuç yüzey yanıt grafiklerinde de gözlenmektedir. Faktörlerin etkileşim grafiği Şekil 4.19'da gösterilmektedir.



Şekil 4.19. Faktörlerin etkileşim grafiği

Faktörlerin yüzey yanıt grafiği Şekil 4.20’de gösterilmektedir.

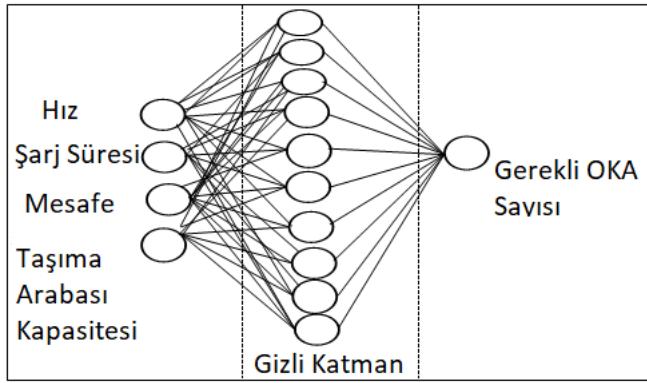


Şekil 4.20. Faktörlerin yüzey yanıt grafiği

Elde edilen model R^2 yaklaşık %99.11’dir. Düzeltilmiş R^2 ise %98’dir. Sonuçlar kabaca modelin gerçek verilere ne kadar iyi uyduğunu gösterir. Model varyasyonu başarılı bir

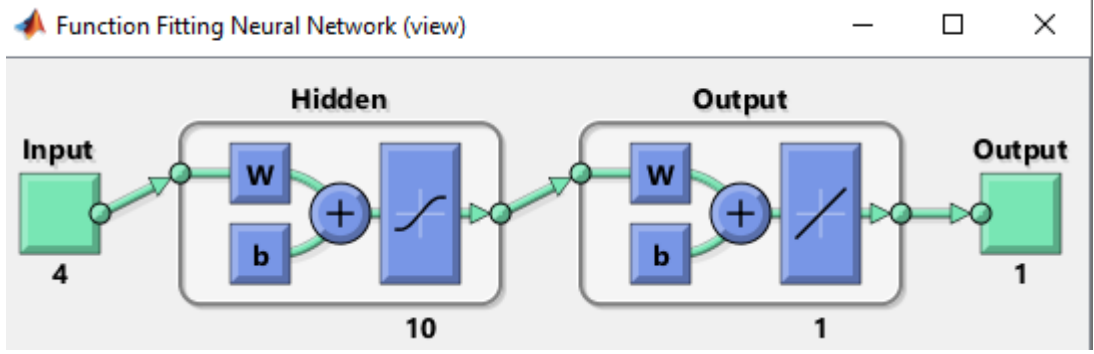
şekilde açıklayabilmektedir. Hız, şarj süresi, mesafeler ve taşıma kapasitesine belirtilen denklem ile gerekli OKA sayısı tahmin edilebilir; açıklanamayan kısımdaki varyasyon (rezidüel) çok düşüktür. Taşınan parça sayısının arttırılması ile en kısa yoldan gerekli OKA sayısı azaltılabilir. Taşınan parça sayısının arttırılması için taşıma arabalarının kapasiteleri arttırılmalıdır. Bu değişiklik için üretim ve tüketim postalarının elverişliliği, ergonomi, taşıma arabalarının modiflenmesinin maliyeti gibi diğer birçok etken göz önüne alınmalıdır.

Gerekli OKA sayısı, sistem tasarımının (parametrelerinin) değiştirilmesi ile azaltılabilir. Sistemde bir değişiklik olursa bu değişimin gerekli OKA miktarına etkisi geliştirilen model ile tahmin edilebilir. Çalışmada kullanılan YSA'nın yapısı Şekil 4.21'de belirtilmiştir.



Şekil 4.21. YSA mimarisi

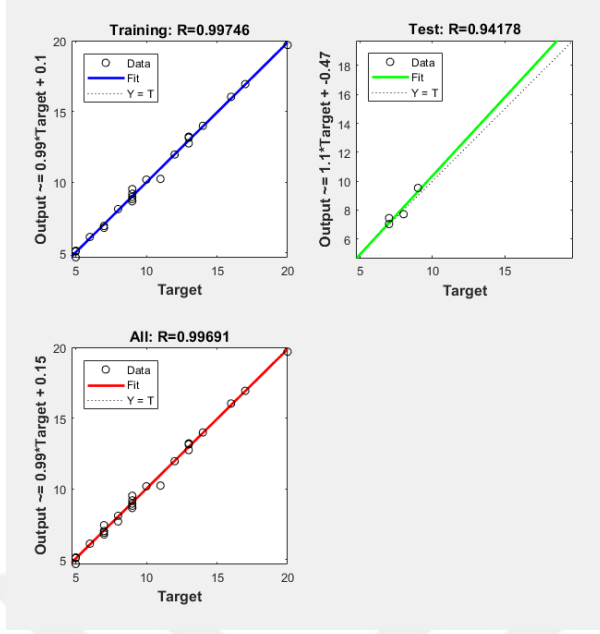
YSA yapısı üç katmanlı bir ağdan oluşmaktadır. Matlab'da, incelenen problem için uygun bir genelleştirilmiş ileri beslemeli YSA geliştirilmiştir. Girdiler hız, şarj süresi, mesafe ve taşınan parça sayısıdır ve çıkış olarak da gerekli OKA sayısı belirlenmiştir. En iyi YSA yapısını bulmak için Pearson korelasyon katsayısı (R) ve hataların karelerinin ortalaması (MSE) değerleri incelenir (Çallı ve diğerleri, 2022). Gizli katman 10 nöron içermektedir. Yapay Sinir Ağları Matlab Model Görüntüsü Şekil 4.22'de gösterilmektedir.



Şekil 4.22. Yapay sinir ağları Matlab model görüntüsü

Çapraz doğrulama, verileri belirtilen oranların alt kümelerine modeli değerlendirmek için kullanılan istatistiksel bir tekniktir. Bu araştırmada, verilerin test, doğrulama ve sinir ağı modeli eğitimi için kullanılan veriler olan alt kümelere bölünmesiyle çapraz doğrulama yöntemi kullanılmıştır. Veri seti küçük olduğu için bu teknik tercih edilmektedir. YSA modeli ve tahminleri, testin Pearson korelasyon katsayısı (R) 0,94'ten yüksek olduğunda ve daha iyi hataların karelerinin ortalaması (MSE) değerleri elde edildiğinde modelin geçerliliğinden söz edilir. R değerinin yanı sıra MSE'nin seçilmesinin nedeni, modelin aşırı öğrenmesini önlemek ve doğruluğu artırmaktır.

Veri setindeki verilerin %80'i eğitim, %15'i test ve %5'i doğrulama için kullanılmıştır. Eğitim için en iyi performans Bayes düzenlemesi ile gerçekleştirilmiş ve performans ortalama hata karesi (MSE) ve Pearson korelasyon katsayısı (R) değerleri açısından değerlendirilmiştir. Sinir ağı modeli için R değerleri, eğitim ve test için kaydedilir. Tüm R-değerleri, Şekil 4.22'de gösterildiği gibi hesaplanır. Sinir ağı modelini eğitmek için Bayes düzenleme geri yayılımı (BRB), Levenberg-Marquardt (LM) ve Esnek Geri Yayılım (RB) kullanılmıştır. Bayes Düzenlemesi, MSE= 0,02658 ile R=0,999 için daha iyi değerlere izin vermiştir. R değeri 1'e mümkün olduğunca yakın olmalıdır. Eğitim ve test R değerleri 0,99 ve 0,94 olarak hesaplanmıştır. Bu durumda, mevcut model, Şekil 4.23'te gösterildiği gibi oldukça iyi bir yaklaşım olan minimum R-değeri 0,94 ile doğruluk oranlarına sahiptir.



Şekil 4.23. Senaryo 1 için YSA performans grafikleri

Hız, şarj süresi, mesafe ve taşınan parça sayısı için sistem parametreleri, $R=0,99969$ ve $MSE 0,0265831$ gibi yüksek doğruluk oranlarıyla tahmin edilmiştir. Bu durumda YSA ile gerekli OKA sayıları Çizelge 4.22 gösterildiği gibi tahmin edilir. Genetik algoritma ile hesaplanan değerler ile yapay sinir ağı ile elde edilen sonuçlar, yapay sinir ağlarının sistem parametrelerinin değişimi sonucunda OKA sayısını tahmininde başarılı bir tahminleyici olduğunu gösterir. YSA girdi ve çıktı analizi Çizelge 4.22’de gösterilmektedir.

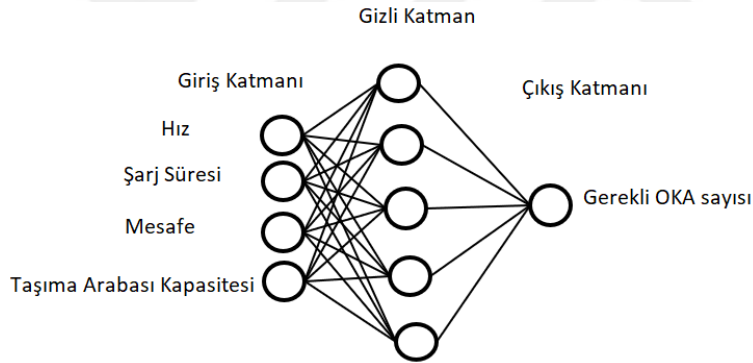
Çizelge 4.22. YSA girdi ve çıktı değerleri

	Hız (m/dak.)	Şarj Süresi (dak.)	Mesafe (oran)	Taşıma Kapasitesi (Oran)	Gerekli OKA sayısı	YSA Tahmini
1	10	8	1	1	13	13,16
2	20	8	0,5	1	7	7,03
3	30	8	1	1	8	7,71
4	10	6	0,5	1	9	8,97
5	20	6	1,5	1,5	8	8,11
6	20	6	0,5	1,5	5	4,70
7	10	6	1	0,5	20	19,73
8	10	6	1	1,5	9	8,66
9	20	6	1	1	9	9,53
10	20	6	1,5	0,5	17	16,95
11	10	4	1	1	13	12,76
12	10	6	1,5	1	16	16,06
13	20	4	1	1,5	7	6,79
14	20	8	1,5	1	12	11,98
15	30	6	0,5	1	5	5,11
16	30	6	1	0,5	11	10,25
17	30	6	1	1,5	6	6,14
18	20	6	1	1	9	9,53
19	20	8	1	1,5	7	6,92
20	20	4	1	0,5	13	13,24
21	30	6	1,5	1	9	8,82
22	20	4	0,5	1	5	5,17
23	20	6	1	1	9	9,53
24	20	4	1,5	1	10	10,20
25	20	6	0,5	0,5	9	9,20
26	30	4	1	1	7	7,45
27	20	8	1	0,5	14	14,01

Hesaplama zamanının kısalığı ve elde edilen başarılı tahminler yapay sinir ağlarının kullanımını arttırmaktadır.

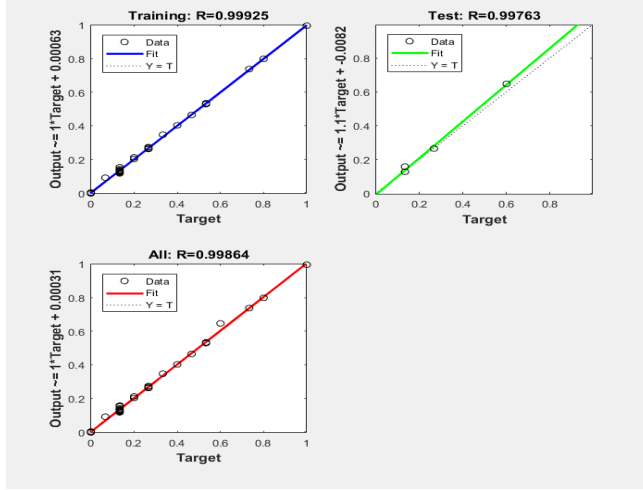
En son olarak veri sayısının artırılarak YSA modeli yeniden ele alınmıştır. Yaklaşım Çoklu Doğrusal Regresyon, Yüzey Yanıt Metodu ve YSA olarak karşılaştırılmıştır. Çoklu doğrusal regresyon modeli Matlab kullanılarak geliştirilmiştir.

Mevcut verilere 13 adet ek veri eklenmesiyle veri sayısı 40 olarak güncellenmiştir. Veri setinin artırılmasıyla incelenmesi yapılamamış olan bölgeler daha ayrıntılı incelenebilmektedir. YSA bu bölgelerde öğrenme sürecini daha etkin şekilde gerçekleştirilebilmektedir. Bu sayede YSA'nın tahmin yapabilme performansı artacaktır. YSA'nın yeni yapısı Şekil 4.24'te gösterilmiştir. Buna göre gizli katmandaki nöron sayısı 5'tir. Nöron sayısı kabulünde birçok nöron yapısıyla deneme yapılarak R^2 ve MSE değerleri en iyi yapılar seçilerek YSA mimarisi tamamlanmaktadır. Bu yapı itibariyle en iyi sonuçlar gizli katman sayısı 5 olduğunda tahmin yapabilme performansı en yüksek seviyeye çıkmaktadır. YSA modeli Matlab Programında oluşturulmuştur.



Şekil 4.24. YSA'nın yeni mimarisi

YSA giriş katmanı 4 nörona sahiptir. Bunlar: Hız, şarj süresi, mesafeler ve taşıma kapasitesidir. Gizli katman 5 nörona sahiptir. Çıkış katmanı ise tek bir nörona sahiptir: Gereklili OKA sayısı. Verilerin %80'i eğitim, %10'u test ve kalan %10'u da doğrulama için kullanılmıştır. Eğitim algoritması için en iyi sonuç Bayesian regularization backpropagation (BRB) ile elde edilmiştir. YSA modelin performansı Şekil 4.25'te gösterilmektedir. Pearson korelasyon katsayısı R oldukça yüksek elde edilmiştir. $R=0.998$ ve $MSE = 2.088 \times 10^{-5}$ elde edilmiştir. R değeri olabildiğince 1'e yakın olmalıdır.



Şekil 4.25. YSA'nın performans değerleri

Çoklu doğrusal regresyon, doğrusal ilişki ile birden fazla açıklayıcı değişken (x_1, x_2, \dots, x_p) ile yanıt değişkeninin (y) tahminlerini yapmak için verileri analiz etmek için kullanılan istatistiksel bir modelleme tekniğidir (Tranmer ve diğerleri,2020) Matlab kullanılarak çoklu doğrusal regresyon ile inceleme gerçekleştirilmiş ve denklem 4.19 elde edilmiştir (Y:Gerekli OKA Sayısı, A:Hız, B:Şarj Süresi, C:Mesafeler, D: Taşıma Kapasitesi; Y, A, B, C, D normalleştirilmiş formdadır):

$$Y = -0,1454 \times A + 0,336 \times B + 0,522 \times C - 0,2958 \times D \quad (4.19)$$

Yeni veri setine göre YSA, YYM ve ÇLR'un karşılaştırılmalı değerleri Çizelge 4.23'te gösterilmiştir.

Çizelge 4.23. YSA, YYM ve ÇLR modellerinin sonuçlarının karşılaştırılması

Deneme	Hız	Şarj Süresi	Mesafeler	Taşıma Kapasitesi	GA	YYM		ÇLR		ANN	
						Değer	MAPE	Değer	MAPE	Değer	MAPE
1	10	8	1	1.00	13	13.3	3%	11.7	10%	13.0	0.0%
2	20	8	0.5	1.00	7	7.3	5%	6.7	4%	7.0	0.1%
3	30	8	1	1.00	8	8.2	2%	9.6	19%	8.0	0.1%
4	10	6	0.5	1.00	9	9.9	10%	5.3	41%	9.0	0.0%
5	20	6	1.5	1.50	8	7.7	4%	9.8	23%	7.8	2.0%
6	20	6	0.5	1.50	5	6.3	27%	2.0	60%	5.0	0.1%
7	10	6	1	0.50	20	18.6	7%	11.4	43%	20.2	0.7%
8	10	6	1	1.50	9	10.1	12%	7.0	22%	9.0	0.0%
9	20	6	1	1.00	9	9.0	0%	8.1	10%	9.0	0.0%
10	20	6	1.5	0.50	17	15.7	8%	14.3	16%	17.0	0.1%
11	10	4	1	1.00	13	12.8	1%	6.7	48%	13.0	0.1%
12	10	6	1.5	1.00	16	15.2	5%	13.1	18%	16.0	0.0%
13	20	4	1	1.50	7	7.0	0%	3.4	52%	7.0	0.0%
14	20	8	1.5	1.00	12	10.7	11%	14.6	21%	12.0	0.0%
15	30	6	0.5	1.00	5	5.8	15%	3.1	38%	5.0	0.1%
16	30	6	1	0.50	11	9.9	10%	9.3	16%	11.0	0.0%
17	30	6	1	1.50	6	7.4	24%	4.8	20%	6.0	0.2%
18	20	6	1	1.00	9	9.0	0%	8.1	10%	9.0	0.0%
19	20	8	1	1.50	7	8.0	14%	8.4	20%	7.0	0.2%
20	20	4	1	0.50	13	12.5	4%	7.8	40%	13.0	0.0%
21	30	6	1.5	1.00	9	8.1	10%	11.0	22%	9.0	0.1%
22	20	4	0.5	1.00	5	5.8	17%	1.7	66%	5.0	0.0%
23	20	6	1	1.00	9	9.0	0%	8.1	10%	9.0	0.0%
24	20	4	1.5	1.00	10	10.2	2%	9.5	5%	10.0	0.0%
25	20	6	0.5	0.50	9	9.3	4%	6.4	29%	9.0	0.0%
26	30	4	1	1.00	7	6.7	5%	4.5	35%	7.0	0.1%
27	20	8	1	0.50	14	13.5	4%	12.9	8%	15.3	9.5%

Çizelge 4.23. YSA, YYM ve ÇLR modellerinin sonuçlarının karşılaştırılması (devam)

28	25	4	1.5	1.50	7	6.6	6%	6.8	3%	7.0	0.1%	
29	30	8	1	1.50	7	8.2	17%	7.3	5%	7.0	0.0%	
30	30	6	1.5	1.00	9	8.1	10%	11.0	22%	9.0	0.1%	
31	30	7	1.50	1.50	7	7.1	1%	10.0	43%	7.0	0.1%	
32	30	8	1	1.50	7	8.2	17%	7.3	5%	7.0	0.0%	
33	30	7	1.50	1.50	7	7.1	1%	10.0	43%	7.0	0.1%	
34	30	6	1.50	1.50	7	6.8	2%	8.7	25%	7.0	0.5%	
35	25	4	1.50	1.50	7	6.6	6%	6.8	3%	7.0	0.1%	
36	25	6	1.50	1.50	7	6.9	1%	9.3	33%	7.0	0.0%	
37	25	8	1	1.50	7	7.8	11%	7.9	13%	7.0	0.2%	
38	15	6	1.50	1.50	9	9.0	0%	10.4	15%	9.0	0.1%	
39	15	5	1.50	1.50	9	9.0	1%	9.1	1%	9.0	0.1%	
40	15	7	1.5	1.5	9	9.1	1%	11.6	29%	9.0	0.1%	
							7%	24%	0.4%			

Ortalama Mutlak Hata Yüzdelerine göre yapılan değerlendirmede en kötü sonuç ÇLR'nindir (%24); YYM'un değeri %7 ve YSA'nın değeri % 0,4 tür. Ayrıca YSA'nın $R=0,998$ ve $MSE=2.088 \times 10^{-5}$ elde edilmiştir. Bu sonuçlardan YSA'nın sistem parametrelerindeki değişimlere göre gerekli OKA sayısını hesaplamada diğer yöntemlere göre daha başarılı olduğu sonucuna varılabilir.

5. TARTIŞMA ve SONUÇ

OKA projeleri önemli yatırım maliyetleri gerektirmektedir. Firmalar maliyeti düşürmek için düşük maliyetli OKA kullanma eğilimindedir. Merkezi yönetim sisteminin olmaması durumunda sistemin verimliliği düşük olmaktadır. Bu nedenle bu çalışmada OKA verimliliklerini arttırmak için OKA havuzlarının kurulması ve üretimdeki talebin minimum OKA sayısı ile sağlanması amaçlanmıştır. Bu çalışmanın temel amacı, merkezi kontrol ünitesi olmayan OKA sistemleri için genetik algoritma (GA) kullanarak optimum OKA sayısının belirlenmesi ve OKA havuzları kurulması ve yapay sinir ağı (YSA) kullanarak sistem parametre değişiklikleri durumunda gerekli OKA sayısının zaman almayan, maliyetsiz bir şekilde tahmin edilmesidir.

PLC veya kart tabanlı ve şerit yönlendirmeli merkezi olmayan sistemlerin programlanmasında karşılaşılan temel sorun, çoğunlukla aşırı deneysel zaman, sınırlı veri ve karmaşıklık kaynaklanmaktadır. Değişkenler OKA hatları ile katlanarak artar ve OKA hatları için matematiksel modeller çözülemez problemlere dönüşür. Bu nedenle, çalışmada, artan OKA hat sayısında gerekli OKA sayılarının belirlenmesinde GA ve YSA, kullanılmıştır.

İlk aşamada, OKA havuzlu düşük maliyetli OKA'lar için optimum filo büyüklüğünü belirlemek amacıyla zaman pencereleri yöntemine dayalı karma tamsayılı programlama kullanılmıştır. Bu yaklaşım, az sayıdaki OKA hattı için başarılı bir şekilde optimum çözümler elde etmiştir, ancak yüksek hesaplama süresi, özellikle karmaşık sistemler söz konusu olduğunda modelin pratikliğini azaltmaktadır. Bu nedenle, kombinatoryal optimizasyonda etkili bir yöntem olan GA, bu sorunu gidermek için kullanılmıştır.

Sistem parametrelerinin OKA sayısı üzerindeki etkilerini tahmin etmek ve üretimde değişen koşullar altında OKA sayısını azaltma olasılığını kontrol etmek için YSA kullanılmıştır. Mevcut durumda YSA %0.4 , YYM %7 ve ÇDR %24 ile GA'ya kıyasla daha düşük hesaplama maliyeti, zaman ve düzeltme hataları ile iyi bir oranda ve karşılaştırmalı olarak düşük hesaplama süresi ile elde edilmiştir. Simülasyon tabanlı analiz için hesaplama süresi hem simülasyon modelinin oluşturulması hem de “what if” analizleri ile yüksek hesaplama zamanı gerektirmektedir. Sonuçlar, YSA'nın düşük hesaplama maliyeti ve zamanı ile atölyede kullanılmak üzere iyi bir korelasyona sahip

çözümler sağladığını göstermiştir. Özellikle üretim talebinin değişmesi durumunda minimum OKA filo büyüklüğüne ulaşmak için GA optimizasyon yaklaşımının veya simülasyon sürecinin tekrarlanmasına gerek olmadığı sonucuna varılabilir.

OKA sayısının uygulanan yaklaşım ile azaltılması nedeniyle üretimde herhangi bir kayba yol açmadığı simülasyon çalışmaları ile gösterilmiştir. Önerilen yaklaşımın, OKA sayısında %29 azalma ile 5 adet daha az OKA gerektiren bir filo boyutu sağladığı tespit edilmiştir. Üretim gereksinimlerinin değişmesi durumunda gerekli OKA sayısını tahmin etmek için sistem parametre değişikliklerinin etkileri de yapay sinir ağları (YSA) ile incelenmiştir. Sistem tasarımcısı, sistem tasarımını değiştirerek ve önerilen sinir ağı tabanlı algoritmayı kullanarak en etkili parametreyi bularak gerekli OKA sayılarını azaltabilir.

OKA sayısına etki eden en önemli faktörün taşıma arabası kapasitesi olduğu görülmüştür. Hız ve mesafe, gerekli OKA sayısının belirlenmesinde ikinci ve üçüncü etkili faktördür. Bu çalışmada OKA'nın hızı 20 m/dakika olarak kabul edilmiştir ve hızın artması OKA sistem güvenliğini ve OKA sistem güvenilirliğini olumsuz yönde etkilemektedir. Bu, süreçlerin değerlendirilmesinde ve farklı üretim gereksinim koşullarına göre işletme stratejisinin tanımlanmasında önemli bir rol oynar.

OKA sistemlerinin tasarımı ve kontrolü ile ilgili birçok çalışma yapılmış olsa da literatür araştırması bilimiz dahilinde var olan çalışmalar çoğunlukla bilgisayar destekli yüksek maliyetli OKA sistemlerine dayanmaktadır. Bu çalışma, düşük maliyetli (PLC veya kart tabanlı), merkezi yönetim sistemi bulunmayan OKA'lar için GA ve YSA kullanımıyla kullanım oranlarının artırılmasıyla araştırma alanına katkısı gösterilmiştir.

Yapay sinir ağları ve yüzey yanıt yöntemi ile sistem parametrelerinin gerekli OKA sayısı üzerindeki etkisi incelenmiş ve OKA sayısını azaltmak için öncelikle taşıma arabalarının kapasitelerinin artırılmasının önemi vurgulanmıştır. Taşıma kapasitesinin artışının, OKA hızı, şarj süresi ve mesafelerden daha önemli olduğu da belirlenmiştir. Taşıma arabası kapasitesinin artırılması durumunda posta düzeni, ergonomi ve iş güvenliği kısıtları da göz önüne alınmalıdır. Diğer yünden sistemde bir değişiklik olacaksa hızlıca etkisi hesap edilip gerekli OKA miktarında değişikliğe gidilebilir.

Çalışma endüstriyel bir problemde denenmesi ve yaratılan deney setinde test edilerek başarılı sonuçlar vermesi bu tip problemlerde karar vericilerin yatırım kararları öncesi bu metodolojiyi uygulamasının yararlı olacağı sonucunu çıkartır.

Havuz sistemi, OKA verimliliğini arttırdığı gibi bir başka avantajı da üretimde çekme sistemine elverişliliği ve tüketim postası parçaya ihtiyaç duydukça parça transferi yaparak postalar arası parça sayısını belirli bir düzeyde tutmasıdır.

Sistem parametrelerinin OKA sayıları üzerindeki etkisi YSA, YYM ve ÇLR ile tahmin edilmiştir. Elde edilen verilen YSA'nın tahminlerinin diğer iki yöntemden daha başarılı sonuçlar elde ettiğini göstermiştir.

Bu çalışmanın sonuçları aşağıda verilmiştir:

(i) Bu çalışmada, merkezi kontrol ünitesi olmayan OKA sistemleri için genetik algoritma (GA) kullanarak optimum OKA sayısının belirlenmesi ve OKA havuzları kurulması ve yapay sinir ağı (YSA) kullanarak sistem parametre değişiklikleri durumunda gerekli OKA sayısının zaman almayan, maliyetsiz bir şekilde tahmin edilmiştir.

(ii) Artan OKA hatları ile matematiksel modeller çözülemez problemlere dönüşür. Bu nedenle, çalışmada, artan OKA hat sayısı ile yöntem olarak GA ve YSA'nın OKA sayılarının belirlenmesi ile bir çözüm elde edilmiştir.

(iii) Bu çalışmada, sistem parametrelerinin OKA sayısı üzerindeki etkilerini tahmin etmek ve üretimde değişen koşullar altında OKA sayısını azaltma olasılığını kontrol etmek için YSA önerilmiştir. Mevcut durumda YSA %0.4, YYM %7 ve %24 ÇDR ile GA'ya kıyasla daha düşük hesaplama maliyeti, zaman ve düzeltme hataları ile iyi bir oranda ve karşılaştırmalı olarak düşük hesaplama süresi ile elde edilmiştir.

(iv) Sistem parametrelerinin değişmesi durumunda minimum OKA filo büyüklüğüne ulaşmak için GA optimizasyon yaklaşımının veya simülasyon sürecinin tekrarlanmasına gerek olmadığı sonucuna varılabilir. YSA'nın ÇLR ve YYM'na göre daha başarılı tahminler yaptığı saptanmıştır.

(v) Önerilen çalışmanın imalatta OKA yokluğu sebebiyle üretimde herhangi bir kayba yol açmadığı simülasyon çalışmaları ile gösterilmiştir. Önerilen yaklaşımın, OKA sayısında %29 azalma ile 5 daha az OKA gerektiren bir filo boyutu sağladığı tespit edilmiştir.

(v₁) Üretim gereksinimlerinin değişmesi durumunda gerekli OKA sayısını tahmin etmek için sistem parametre değişikliklerinin etkileri de yapay sinir ağları (YSA) ile incelenmiştir. Sistem tasarımcısı, sistem tasarımını değiştirerek ve önerilen sinir ağı tabanlı algoritmayı kullanarak en etkili parametreyi bularak gerekli OKA sayısını azaltılabileceği gösterilmiştir.

(v₁₁) OKA sayısına etki eden en önemli faktörün taşıma arabası kapasitesi olduğu görülmüştür. Hız ve mesafe, gerekli OKA sayısının belirlenmesinde ikinci ve üçüncü etkili faktör olduğu görülmüştür.

Bu çalışma aşağıda sıralanan alanlara katkı sağlamıştır:

(i) OKA sistemlerinin tasarımı ve kontrolü ile ilgili birçok çalışma yapılmış olsa da, var olan çalışmalar çoğunlukla yüksek maliyetli OKA sistemlerine dayanmaktadır. Bu çalışma, düşük maliyetli OKA'lar için GA ve YSA kullanımıyla kullanım oranlarının artırılmasıyla araştırma alanına katkı oluşturacağı gösterilmiştir.

(ii) Çalışma endüstriyel bir problemde denenmesi ve yaratılan deney setinde test edilerek başarılı sonuçlar vermesi bu tip problemlerde karar vericilerin yatırım kararları öncesi bu metodolojiyi uygulamasının yararlı olacağı sonucunu göstermiştir.

(iii) Sistem parametrelerinin OKA sayılarına olan etkisi YSA, ÇLR ve YYM ile analiz edilmiş bu sayede imalat ve mühendislik departmanlarında çalışanlar gerekli yeni OKA sayısını tahmin edebilir ve OKA miktarının azaltılması için hangi faktörleri değiştireceğinin tespit edilebilmesine katkısı olmuştur.

(iv) Özellikle artan rekabet, firmaları yatırımları en optimal şekilde yapmaya zorlamaktadır. Dolayısıyla yatırım kararları öncesi etütlerde bu yaklaşımlara başvurulması, firmanın yatırım maliyetlerinin azaltmasına katkısı olacaktır.

KAYNAKLAR

- Aktürk, M. S. ve Yılmaz, H. (1996). Scheduling of automated guided vehicles in a decisionmaking hierarchy, *International Journal of Production Research*, 32, 577±591.
- Anderson-Cook, C. M., Borror, C. M. ve Montgomery, D. C. (2009). Response surface design evaluation and comparison, *Journal of Statistical Planning and Inference*, 139 (2) , 629-641.
- Aslan, N. ve Cebeci, Y. (2007). Application of Box–Behnken design and response surface methodology for modeling of some Turkish coals, *Fuel*, 86(1-2), 90-97.
- Banks, J., Carson, J. S., Nelson, B. L. ve Nicol D. M. (2005). *Discrete-Event System Simulation*, Prentice-Hall. Fourth Edition.
- Bezerra, M.A, Santelli, R.E, Oliveira, E.P., Villar, L.S., Escaleira, L.A. (2008). Response surface methodology (RSM) as a tool for optimization in analytical chemistry. *Talanta*. 2008 Sep 15;76(5):965-77. doi: 10.1016/j.talanta.2008.05.019. *Epub* 2008 May 21. PMID: 18761143.
- Bodenhofer, U. (2003). *Genetic algorithms: theory and applications. Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg*, Winter, 2004.
- Bouzaïd, L., Yalçın, M. A., Chaoui, K., Mabrouki, T., ve Boulanouar, L. (2015). Mathematical modeling for turning on AISI 420 stainless steel using surface response methodology. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 229(1), 45-61.
- Bozorgi-Amiri, A., Mahmoodian, V. F., Elham ve Saffari, H. (2015). A new memetic algorithm for solving split delivery vehicle routing problem. *Management Science Letters*. 5. 1017-1022. 10.5267/j.msl.2015.8.010.
- Chang, K. H., Chang, A. L. ve Kuo, C. Y. (2014). A simulation-based framework for multi-objective vehicle fleet sizing of automated material handling systems: an empirical study, *Journal of Simulation*, 8(4), 271-280. 10.1057/jos.2014.6
- Chaudhry, I., Rafique, A., Elbadawi, I., Aichouni, M., Usman, M., Boujelbene, M., ve Boudjemline, A. (2022). Integrated scheduling of machines and automated guided vehicles (AGVs) in flexible job shop environment using genetic algorithms. *International Journal of Industrial Engineering Computations*, 13(3), 343-362.
- Chung, C. A. (2004). *Simulation Modeling Handbook A Practical Approach*, CRC Press, United States of America.
- Colling, D., Oehler, J., ve Furmans, K. (2019). Battery charging strategies for AGV systems. *Logistics Journal: Proceedings*, 2019(12).
- Çallı, M., Albak, E.I. ve Öztürk, F. (2022). Prediction and optimization of the design and process parameters of a hybrid DED product using artificial intelligence. *Applied Science*, 12(10), 5027. <https://doi.org/10.3390/app12105027>.
- Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., ve dos Reis Alves, S. F. (2017). *Artificial neural networks*. Cham: Springer International Publishing, 39.10.1007/978-3-319-43162-8
- Da Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B., ve dos Reis Alves, S. F. (2017). *Artificial neural networks*. Cham: Springer International Publishing, 39.
- Dongare, A. D., Kharde, R. R. ve Kachare, A. D. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1), 189-194.
- Eiben, A. E. ve Smith, J. E., (2003). *Introduction to Evolutionary Computing*, Springer.

- Ene S., Küçükoglu İ., Aksoy A. ve Oztürk N. (2016). A genetic algorithm for minimizing energy consumption in warehouses. *Energy* 114, 973-980.
- Fazlollahtabar, H. (2016). Parallel autonomous guided vehicle assembly line for a semi-continuous manufacturing system. *Assembly Automation*. Vol. 36 No. 3, pp. 262-273. 10.1108/AA-08-2015-065.
- Ferreira, T. ve Gorlach, I. A. (2016). Development of an automated guided vehicle controller using a model-based systems engineering approach, *South African journal of industrial engineering*, 27(2), 206-217.
- Gen, M. ve Cheng, R., (1997). *Genetic Algorithms and Engineering Design*, John Wiley and Sons, Inc.
- Han, W., Xu, J., Sun, Z., Liu, B., Zhang, K., Zhang, Z. ve Mei, X. (2022). Digital twin-based automated guided vehicle scheduling: a solution for its charging problems. *Applied Sciences*, 12. 3354. 10.3390/app12073354.
- Haupt, R. L. ve Haupt, S. E., (2004). *Practical Genetic Algorithms*, Second Edition, A John Wiley & Sons, Inc.
- Hillier F.S., Lieberman G.J., (2019). *Yöneylem Araştırmasına Giriş*, (Çebi F., Deveci Kocakoç İ., Çev. Editörleri), Atlas Akademi Basım Yayın Dağıtım, Ankara. (Orijinal çalışma basım tarihi 2015).
- İnanç, Ş. ve Şenaras, A. E. (2019). AGV Routing via Ant Colony Optimization Using C#. In *Optimization Using Evolutionary Algorithms and Metaheuristics* .pp. 23-31. CRC Press.
- Jia, L., Shi, L., Yao, J., Dai, X. ve Guo, G. (2022, November). Research on Intelligent Decision Method of Optimal Production Planning and AGV In-time Delivery in Mixed-Model Assembly Line. In *Journal of Physics: Conference Series* (Vol. 2363, No. 1,p. 012026). IOP Publishing.
- Karaboğa, D. (2014). *Yapay zeka optimizasyon algoritmaları*. Nobel Akademik Yayıncılık, 245.
- Katoch, S., Chauhan, S.S. ve Kumar, V. A review on genetic algorithm: past, present, and future. *Multimed Tools Appl* 80, 8091–8126 (2021). <https://doi.org/10.1007/s11042-020-10139-6>
- Kim, B. I. ve Park, J. (2009). Idle vehicle circulation policies in a semiconductor Fab. *Journal of Intelligent Manufacturing*, 20(6), 709-717. 10.1007/s10845-008-0159-4.
- Lacomme, P., Larabi, M., ve Tchernev, N. (2013). Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. *International Journal of Production Economics*, 143(1), 24-34. 10.1016/j.ijpe.2010.07.012
- Law, A.M. ve Kelton, W.D. (2000). *Simulation Modeling & Analysis*, McGrawHill, Third Edition.
- Law, A.M. (2015). *Simulation Modeling & Analysis*, McGrawHill, Fifth Edition.
- Le-Anh, T., ve De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171(1), 1-23.
- Lee, C., ve Ventura, J. A. (2001). Optimal dwell point location of automated guided vehicles to minimize mean response time in a loop layout. *International Journal of Production Research*, 39(17), 4013-4031. 10.1080/00207540110054605
- Lin, D. K., ve Peterson, J. J. (2006). Statistical inference for response surface optima. In *Response Surface Methodology and Related Topics* (pp. 65-88).
- Mathews, P. G. (2004). *Design of Experiments with MINITAB*. Quality Press.
- Matlab R17. <https://www.mathworks.com/products/matlab.html> (Accessed: 12.12. 2020).

Minitab19 <https://www.minitab.com/en-us/>

Mitchell, M., (1999). *An Introduction to Genetic Algorithms*, The MIT Press.

Miyamoto, T. ve Inoue, K. (2016). Local and random searches for dispatch and conflict-free routing problem of capacitated AGV systems, *Computers & Industrial Engineering*, 91, 1-9. 10.1016/j.cie.2015.10.017

Moghadam, B. F., Sadjadi, S. J. ve Seyedhosseini, S. M. (2010). An empirical analysis on robust vehicle routing problem: a case study on drug industry. *International Journal of Logistics Systems and Management*, 7(4), 507-518. 10.1504/IJLSM.2010.035635

Moghaddam, B. F., Ruiz, R. ve Sadjadi, S. J. (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering*, 62(1), 306-317. 10.1016/j.cie.2011.10.001.

Montgomery, D. C. (2005). *Design and Analysis of Experiments: Response surface method and designs*. New Jersey: John Wiley and Sons, Inc.

Moshayedi, A. J., Jinsong, L. ve Liao, L. (2019). OKA (automated guided vehicle) robot: Mission and obstacles in design and performance. *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering*, 12(4), 5-18

Mozol, Š., Krajčovič, M., Dulina, L., Mozolová, L. ve Oravec, M. (2022). Design of the System for the Analysis of Disinfection in Automated Guided Vehicle Utilisation. *Applied Sciences*. 12. 9644. 10.3390/app12199644.

Nishi, T., Morinaka S. ve Konishi M. (2007).a. A distributed routing method for AGVs under motion delay disturbance. *Robotics and Computer-Integrated Manufacturing*. 23 517–532 10.1016/j.rcim.2006.10.007

Nishi, T., Hiranaka Y., Inuiguchi, M., Grossmann, I.E. (2007).b. A Decomposition Method with Cut Generation for Simultaneous Production Scheduling and Routing for multiple AGVs. *2007 Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering* Scottsdale, AZ, USA. 10.1109/COASE.2007.4341680

Nishi T., Shimatani K. ve Inuiguchi, M. (2009). Decomposition of Petri nets and Lagrangian relaxation for solving routing problems for AGVs. *International Journal of Production Research*. 47:14, 3957-3977. 10.1080/00207540701846244

Nishi, T., Hiranaka, Y. ve Grossmann, I. E. (2011). A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. *Computers & Operations Research*, 38(5), 876-888. 10.1016/j.cor.2010.08.012

Öztemel, E. (2003). *Yapay sinir ağları*. PapatyaYayincılık, İstanbul.

Öztürk, A. (2021). *Yöneylem Araştırması*, Ekin Basım Yayım Dağıtım, Bursa.

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F. ve Mahmoodian, V. (2015). An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering*, 86, 2-13. 10.1016/j.cie.2015.01.003

Sastry, K., Goldberg, D. ve Kendall, G. (2005). Genetic Algorithms. In: Burke, E.K., Kendall, G. (eds) *Search Methodologies*. Springer, Boston, MA. https://doi.org/10.1007/0-387-28356-0_4

Sezen, H. K. (2007). *Yöneylem Araştırması*, Ekin Basım Yayım Dağıtım, Bursa.

Shivanand, H. K. (2006). *Flexible manufacturing system*. New Age International.

Shopova, E. G. ve Vaklieva-Bancheva, N. G. (2006). BASIC—A genetic algorithm for engineering problems solution. *Computers & chemical engineering*, 30(8), 1293-1309.

Soylu, M., Özdemirel, N. E. ve Kayaligil, S. (2000). A self-organizing neural network approach for the single AGV routing problem. *European Journal of Operational Research*, 121(1), 124-137. 10.1016/S0377-2217(99)00032-6

- Şenaras, A. E. (2019). Parameter optimization using the surface response technique in automated guided vehicles. In *Sustainable Engineering Products and Manufacturing Technologies*. pp. 187-197. Academic Press.
- Talbot, L. (2003). Design and performance analysis of multistation automated guided vehicle systems (Doctoral dissertation, UCL-Université Catholique de Louvain).
- Taşkın, Ç. ve Emel, G. G. (2009). *Sayısal yöntemlerde genetik algoritmalar*. Alfa Aktüel.
- Tonelli, F., Paolucci, M., Anghinolfi, D. ve Taticchi, P. (2013). Production planning of mixed-model assembly lines: A heuristic mixed integer programming based approach *Production Planning & Control* 24 (1) 110-27.
- Tranmer, M., Murphy, J., Elliot, M. ve Pampaka, M. (2020). *Multiple Linear Regression* (2nd Edition); Cathie Marsh Institute Working Paper 2020-01 <https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/2020-1-multiple-linear-regression.pdf>
- Tureson, J. (2016). *Development of a User Interface for Automated Vehicles Remote Control of an AGV* Gothenburg, Sweden
- Ullrich, G. (2015). *Automated Guided Vehicle Systems*. Springer. DOI 10.1007/978-3-662-44814-4_3 DIN 13312:2005-02 "Navigation – Terms.
- Valmiki, P., Reddy, A. S., Panchakarla, G., Kumar, K., Purohit, R. ve Suhane, A. (2018). A study on simulation methods for AGV fleet size estimation in a flexible manufacturing system. *Materials Today: Proceedings*, 5(2), 3994-3999. 10.1016/j.matpr.2017.11.658
- Van der Meer, J.R. (2000). Operational Control of Internal Transport (No. 1). ERIM Ph.D. Series Research in Management. Erasmus University Rotterdam. Retrieved from <http://hdl.handle.net/1765/859>
- Ventura, J. A., Pazhani, S. ve Mendoza, A. (2015). Finding optimal dwell points for automated guided vehicles in general guide-path layouts. *International Journal of Production Economics*, 170, 850-861. 10.1016/j.ijpe.2015.03.007
- Ventura, J. A. ve Lee, C. (2003). Optimally locating multiple dwell points in a single loop guide path system. *IIE Transactions*, 35(8), 727-737. 10.1080/07408170304346
- Ventura, J. A. ve Rieksts, B. Q. (2009). Optimal location of dwell points in a single loop AGV system with time restrictions on vehicle availability. *European Journal of Operational Research*, 192(1), 93-104. 10.1016/j.ejor.2007.09.014
- Viharos, A. B. ve Németh, I. (2018). Simulation and scheduling of AGV based robotic assembly systems. *IFAC-PapersOnLine*, 51(11), 1415-1420. 10.1016/j.ifacol.2018.08.317
- Vis, I. F. A. (2006). Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170(3), 677-709 10.1016/j.ejor.2004.09.020
- Vob, S., Martello, S., Osman, I. H., ve Roucairol C., (1999). *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publisher. Witness 22.b <https://www.lanner.com/en-us/technology/witness-simulation-software.html>
- Wu S. ve Wu Y. (2017). Development and Application Analysis of AGVs in *Modern Logistics Revista de la Facultad de Ingeniería U.C.V.*, Vol. 32, N°5, pp. 380-386.
- Yifei, T., Junruo, C., Meihong, L., Xianxi, L. ve Yali, F. (2010). An estimate and simulation approach to determining the automated guided vehicle fleet size in FMS. In 2010 3rd International Conference on Computer Science and Information Technology (Vol. 9, pp. 432-435). IEEE. 10.1109/ICCSIT.2010.5565147.

<https://www.maptoaster.com/maptoaster-topo-nz/articles/how-gps-works/how-gps-works.html> (Eriřim Tarihi: 16.09.2022)

