KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

TAM ZAMANINDA ÜRETİM ORTAMINDA FABRİKA İÇİ DÖNGÜSEL TEDARİK SİSTEMİ İÇİN ÇAKIŞMASIZ ROTA OPTİMİZASYONU: MATEMATİKSEL MODEL VE MAT-SEZGİSEL ÇÖZÜM ÖNERİSİ

ALİ İHSAN BOYACI

KOCAELİ 2023

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

TAM ZAMANINDA ÜRETİM ORTAMINDA FABRİKA İÇİ DÖNGÜSEL TEDARİK SİSTEMİ İÇİN ÇAKIŞMASIZ ROTA OPTİMİZASYONU: MATEMATİKSEL MODEL VE MAT-SEZGİSEL ÇÖZÜM ÖNERİSİ

ALİ İHSAN BOYACI

Prof. Dr. Alpaslan FIĞLALI		
Danışman,	**Kocaeli Üniversitesi**	**.....**
Prof. Dr. Gülşen AKMAN		
Jüri Üyesi,	**Kocaeli Üniversitesi**	**.....**
Dr. Öğr. Üyesi Celal ÖZKALE		
Jüri Üyesi,	**Kocaeli Üniversitesi**	**.....**
Prof. Dr. İbrahim ÇİL		
Jüri Üyesi,	**Sakarya Üniversitesi**	**.....**
Doç. Dr. Mümtaz KARATAŞ		
Jüri Üyesi,	**Deniz Kuvvetleri Komutanlığı**	**.....**
Doç. Dr. Kasım BAYNAL		
Eş Danışman,	**Kocaeli Üniversitesi**	

Enstitü Yönetim Kurulunun 11/03/2020 tarih ve 2020/12 nolu toplantısında II. tez danışmanı olarak atanmıştır.

Tezin Savunulduğu Tarih: 20.10.2023

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/Projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

☒ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☐ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler ..... tarafından ..... no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Ali İhsan BOYACI

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.
Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.
Yükseköğretim kurulu tarafından yayınlanan ***"Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.

☒ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.

☐ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

Ali İhsan BOYACI

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında tam zamanında üretim ortamında, fabrika içerisinde sabit çevrim süreli döngüsel seferlerle parça tedariği yapan çekici trenlerin çakışmasız bir şekilde rotalanması, çizelgelenmesi ve yüklenmesi problemleri ele alınmaktadır.

Uzun süren çalışmalar sonucu ortaya çıkan bu tezin fikir aşamasından sonuçlanmasına kadar her noktasında desteklerini esirgemeyen danışmanım Prof. Dr. Alpaslan FIĞLALI'ya ve eş danışmanım Doç. Dr. Kasım BAYNAL'a sonsuz şükranlarımı sunarım. Ayrıca Tez İzleme Komitesinde bulunan ve bu teze çok önemli katkılar sunan çok değerli hocalarım Doç. Dr. Mümtaz KARATAŞ'a ve Dr. Öğr. Üyesi Celal ÖZKALE'ye teşekkür ederim.

Beni bu günlere getiren, her zaman olduğu gibi doktora eğitimim sırasında da manevi ve maddi desteklerini asla esirgemeyen, fedakâr ve çok kıymetli annem Nazike BOYACI'ya, babam Ekrem BOYACI'ya ve kardeşlerime sonsuz minnetlerimi sunarım. Ayrıca tez çalışmalarım boyunca her türlü fedakârlığı gösterip bana bütün kalbiyle destek olan biricik eşim Büşra BOYACI'ya en içten teşekkürlerimi sunarım.

Ekim – 2023 Ali İhsan BOYACI

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
İÇİNDEKİLER iv
ŞEKİLLER DİZİNİ v
TABLOLAR DİZİNİ vii
SİMGELER VE KISALTMALAR DİZİNİ ix
ÖZET xiii
ABSTRACT xiv
1. GİRİŞ 1
2. LİTERATÜR TARAMASI 9
2.1. Parça Besleme Politikaları 10
2.2. Süpermarket Sayısı ve Yerleşimi 15
2.3. Rotalama, Çizelgeleme ve Yükleme Problemleri 17
3. PROBLEM TANIMI 36
4. ÇÖZÜM YAKLAŞIMLARI 46
4.1. Birinci Çözüm Yaklaşımı: Karma Tam Sayılı Bir Model (Model-1) 46
4.2. İkinci Çözüm Yaklaşımı: İteratif Çözüm 59
4.2.1. Model-2 61
4.2.2. Model-3 62
4.2.3. İteratif Çözüm yaklaşımı 68
4.3. Üçüncü Çözüm Yaklaşımı: Mat-Sezgisel Algoritma 77
4.3.1. Başlangıç Birey Üretme 80
4.3.2. Çözüm Geliştirme 90
4.3.2.1. Komşuluk Operatörleri 93
4.3.2.2. Birey Seçimi 99
4.3.3. Çakışmasız Çözüm Arama 101
5. TEST PROBLEMLERİ 107
5.1. Test Problemlerinin Oluşturulması 107
5.2. Bilgisayar Koşum Sonuçlarının Analizi ve Tartışma 112
5.2.1. Küçük Boyutlu Problemler 114
5.2.2. Orta Boyutlu Problemler 119
5.2.3. Büyük Boyutlu Problemler 129
5.2.4. Çakışma Yoğunluğu Analizi 140
6. SONUÇLAR VE ÖNERİLER 143
KAYNAKLAR 150
EKLER 160
KİŞİSEL YAYIN VE ESERLER 165
ÖZGEÇMİŞ 167

ŞEKİLLER DİZİNİ

Şekil 3.1. Örnek bir fabrika krokisi 37
Şekil 3.2. Üretim hücresi 38
Şekil 3.3. Basitleştirilmiş fabrika krokisi görünümü 38
Şekil 3.4. Dar koridorda çekici trenlerin karşılıklı çakışması durumu 43
Şekil 3.5. Dar koridorda çekici trenlerin arka arkaya çakışması durumu 43
Şekil 3.6. Çekici trenler için örnek bir rotalama gösterimi 44
Şekil 4.1. Krokilerdeki köşe - başlangıç noktalarının ve koridorların numaralandırılması 47
Şekil 4.2. Dar bir yolda iki çekici trenin karşı karşıya gelmesi durumu 56
Şekil 4.3. Dar bir yolda iki çekici trenin arka arkaya gelmesi durumu 58
Şekil 4.4. Küçük boyutlu örnek bir kroki 60
Şekil 4.5. Örnek bir alt turlu rotalama 65
Şekil 4.6. Model-3 ile alt tursuz örnek bir rotalama 67
Şekil 4.7. Model-3 için alt tur eleme prosedürü 67
Şekil 4.8. İteratif çözüm yaklaşımının akış şeması 71
Şekil 4.9. Farklı koridorlardan gidilmesiyle oluşabilecek alternatif optimum rotalama örnekleri 72
Şekil 4.10. Aynı koridorların farklı yönlerde veya sıralarda gidilmesiyle oluşabilecek alternatif optimum rota örnekleri 73
Şekil 4.11. Çakışma kontrolü algoritmasının sözde kodu 74
Şekil 4.12. Model-2 ile elde edilmiş örnek bir optimum çözümde var olan çakışmalar 76
Şekil 4.13. Mat-sezgisel algoritmanın genel çerçevesi 79
Şekil 4.14. Örnek bir birey gösterimi 80
Şekil 4.15. Başlangıç birey oluşturma algoritmasının akış şeması 81
Şekil 4.16. Başlangıç birey oluşturma algoritmasının sözde kodu 82
Şekil 4.17. Koordinat eksenleri yerleştirilmiş bir fabrika krokisi 86
Şekil 4.18. Birinci görev atandıktan sonra Model 3 ile oluşturulan rota 88
Şekil 4.19. Çözüm geliştirme algoritmasının akış şeması 91
Şekil 4.20. Çözüm geliştirme algoritmasının sözde kodu 92
Şekil 4.21. Rotalar arası aktarma operatörünün akış şeması 94
Şekil 4.22. Rotalar arası aktarma operatörüne örnek bir durum 96
Şekil 4.23. Rotalar arası değiştirme operatörünün akış şeması 97
Şekil 4.24. Rotalar arası aktarma operatörüne örnek bir durum 98
Şekil 4.25. Çözüm uzayının temsili gösterimi 100
Şekil 4.26. Çakışmasız çözüm arama algoritmasının akış şeması 102
Şekil 4.27. Sıralı çakışmasız rotalama algoritmasının akış şeması 103
Şekil 4.28. Çakışmasız çözüm arama algoritmasının sözde kodu 104
Şekil 5.1. Koordinat eksenleri verilmiş yatay boyutlu dikdörtgen formlu kroki 110
Şekil 5.2. Test problemlerinin kodlanması 111
Şekil 5.3. Küçük boyutlu kroki kullanılarak üretilen test problemleri için AKKO değerleri 118
Şekil 5.4. Küçük boyutlu kroki kullanılarak üretilen test problemlerinin çözülmesi ile elde edilen kapasite kullanım oranları 119

Şekil 5.5. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri 123
Şekil 5.6. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri 124
Şekil 5.7. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri 128
Şekil 5.8. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri 129
Şekil 5.9. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri 132
Şekil 5.10. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri 133
Şekil 5.11. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri 135
Şekil 5.12. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri 136
Şekil 5.13. Orta ve büyük boyutlu problemlerin BÇDO iyileştirme oranları için Mann-Whitney U testi sonucu 139
Şekil 5.14. Orta boyutlu krokilerle üretilen test problemlerinde görülen çakışma sayıları 141
Şekil 5.15. Büyük boyutlu krokilerle üretilen test problemlerinde görülen çakışma sayıları 142
Şekil A.1. Küçük boyutlu kroki 161
Şekil A.2. Orta boyutlu - dikdörtgen formlu kroki 161
Şekil A.3. Orta boyutlu - kare formlu kroki 162
Şekil A.4. Büyük boyutlu - dikdörtgen formlu kroki 163
Şekil A.5. Büyük boyutlu - kare formlu kroki 164

TABLOLAR DİZİNİ

Tablo 2.1. Fabrika içi taşıyıcı araçların rotalanması çizelgelenmesi ve yüklenmesi ile ilgili literatür özeti 34
Tablo 4.1. Örnek bir görev listesi 48
Tablo 4.2. Çeşitli problemler için değişken sayıları dağılımları 61
Tablo 4.3. Geliştirilen modellerin özelliklerinin karşılaştırılması 68
Tablo 4.4. Modellerin farklı test problemlerinin çözümü için gerektirdiği değişken sayıları 69
Tablo 4.5. Modellerin farklı test problemlerinin çözümü için gerektirdiği kısıt sayıları 69
Tablo 4.6. Örnek bir görev-rota ataması 80
Tablo 4.7. Görev sıralama kriterleri kombinasyonları 83
Tablo 4.8. Örnek atanacak görevler listesi 87
Tablo 4.9. Birinci kombinasyon satırının ilk görev atama kriterlerine göre sıralanmış atanacak görevler listesi 88
Tablo 4.10. Birinci kombinasyon satırın diğer görevleri atama kriterlerine göre sıralanmış atanacak görevler listesi 89
Tablo 5.1. Test problemleri oluşturmak için belirlenen parametreler ve seviyeleri 107
Tablo 5.2. Koridorların alanları ve koridor sayıları 108
Tablo 5.3. Küçük kroki ile oluşturulan deneme problemlerinin parametrelerinin gösterimi 111
Tablo 5.4. Araç sayısına bağlı olarak CS araç böleni parametresinin aldığı değerler 112
Tablo 5.5. Mat-sezgisel algoritmanın parametre değerleri 113
Tablo 5.6. Küçük boyutlu kroki kullanılarak üretilen test problemlerinin çözüm sonuçları 115
Tablo 5.7. Küçük boyutlu kroki kullanılarak üretilen test problemleri için AKKO değerleri 117
Tablo 5.8. Küçük boyutlu kroki kullanılarak üretilen test problemleri için KKO değerleri 119
Tablo 5.9. Orta boyutlu dikdörtgen kroki ile üretilen test problemlerinin çözüm sonuçları 121
Tablo 5.10. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri 122
Tablo 5.11. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri 123
Tablo 5.12. Orta boyutlu kare kroki ile üretilen test problemlerinin çözüm sonuçları 125
Tablo 5.13. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri 127
Tablo 5.14. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri 128
Tablo 5.15. Büyük boyutlu dikdörtgen kroki ile üretilen test problemlerinin çözüm sonuçları 130
Tablo 5.16. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri 131

Tablo 5.17. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri 132
Tablo 5.18. Büyük boyutlu kare kroki ile üretilen test problemlerinin çözüm sonuçları 134
Tablo 5.19. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri 135
Tablo 5.20. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri 136
Tablo 5.21. Orta ve büyük boyutlu problemlerin BÇDO iyileştirme oranlarına ait veriler 137
Tablo 5.22. Normal dağılıma uygunluk testlerinin sonuçları 138
Tablo 5.23. Kroki tiplerine göre test problemlerinde görülen çakışma sayıları 140

SİMGELER VE KISALTMALAR DİZİNİ

h_{ij}^{ktfy} : Arka arkaya çakışma engelleme kısıtlarını modellemek için kullanılan yardımcı ikili değişken

$ACAKT_{(i,j)}$: (i,j) koridorunda aynı yönde çakışmanın olduğu hareket ikililerini içeren küme

AK_{ij}^{e} : e görevinin alma koridoru, (i,j) komşu köşe noktaları arasında ise 1 değerini alan parametre

ALT^{k} : k aracının esas alt turundaki köşe noktalarının esas alt tur dışındaki komşu köşe noktaları ile olan kenarlarını (koridorları) içeren küme

BK_{ij}^{e} : e görevinin bırakma koridoru, (i,j) komşu köşe noktaları arasında ise 1 değerini alan parametre

E_a : Atanacak görevler kümesi

F_{ij} : Alma veya bırakma koridoru (i,j) komşu köşe noktaları arasında olan görevler kümesi

$KCAKT_{(i,j)}$: (i,j) koridorunda karşılıklı çakışmanın olduğu hareket ikililerini içeren küme

M^{k} : k aracı için esas alt turun geçtiği köşe noktalarını içeren küme

PAA_e : e görevinin alma koridorunun alt köşe noktası

PAU_e : e görevinin alma koridorunun üst köşe noktası

PBA_e : e görevinin bırakma koridorunun alt köşe noktası

PBU_e : e görevinin bırakma koridorunun üst köşe noktası

V_{ij} : (i,j) koridorunda gidildikten sonra u dönüşü yapmadan gidilebilecek komşu köşe noktaları kümesi

b_i : i. birey

d_{ij} : $(i,j) \in A$ yolunun uzunluğu (metre)

l^{kt} : k aracının t hareketinde aldığı toplam dolu kutu miktarı

p_e^k : e görevinin k aracına atanıp atanmadığını belirleyen ikili değişken

q_e : e görevinin periyot başına kutu miktarı

r^{kt} : k aracının t hareketinde bıraktığı toplam dolu kutu miktarı

s_{ij}^{kt} : k aracının t hareketinde (i,j) köşe noktaları arasında hizmet verip vermediğini belirleyen ikili değişken

u_{ij}^{ktfy} : Karşılıklı çakışma engelleme kısıtlarını modellemek için kullanılan yardımcı ikili değişken

w^{kt} : k aracının t hareketinde bekleme süresi (saniye)

x_{ij}^{k} : k aracının (i,j) koridorunda $i-j$ yönünde kaç kere gittiğini ifade eden tam sayılı karar değişkeni

x_{ij}^{kt} : k aracının t hareketinde i noktasından j noktasına gidip gitmediğini belirleyen ikili değişken

z^{kt} : k aracının t hareketi sonunda köşe noktasından geçiş zamanı

MS : Maksimum geliştirme süresi

p : Önem değeri

Γ : Alma noktaları kümesi

Π : Tüm noktalar kümesi

Φ : Bırakma noktaları kümesi

A : Bütün koridorlar kümesi
$AÇS$: Alternatif çözümleri içeren küme
ABS : Alınan kutu başına birim işlem süresi (saniye)
$ACAK$: Aynı yönde çakışmanın olduğu (i, j) koridorlarını içeren küme
AS : Rasgele seçilmiş araç sıralamaları kümesi
BBS : Bırakılan kutu başına birim işlem süresi (saniye)
C : Çekici trenlerin kutu cinsinden kapasitesi (kutu)
CS : Çevrim süresi (saniye)
DY : Dar koridorlar kümesi
E : Görevler kümesi
EBS : Elitist yöntemle seçilecek birey sayısı
EIB : En iyi bireyler kümesi
GB : Geliştirme sonrası bireyler kümesi
GBK : Geliştirilecek başlangıç bireyler kümesi
GBS : Geliştirilecek birey sayısı parametresi
K : Araçlar kümesi
$KCAK$: Karşılıklı çakışmanın olduğu (i, j) koridorlarını içeren küme
KSK : Sıralama kriterlerinin kombinasyonlarını içeren küme
M : Büyük bir sayı
MNS : Maksimum nesil sayısı
$MSÇ$: Çakışmasız çözüm arama aşaması için maksimum süre.
N : Köşe noktaları kümesi
$NBK(gb, ns)$: gb başlangıç bireyinin, ns neslinde bulunan bireyler kümesi
$P(b_i)$: b_i bireyinin seçilme olasılığı
$RÇB$: Rulet çemberi ile seçilecek birey sayısı
$RASS$: Rasgele belirlenecek araç sıralamaları sayısı
RB : Tamamen rasgele seçilecek birey sayısı
$S(ks)$: ks. sıralama kriterleri kombinsayonu satırı
T : Hareketler kümesi (Planlama ufku)
YBK : Yeni bireyler kümesi
$açs$: Alternatif çözüm senaryosu indisi
as : Araç sıralaması indisi
e : Görev indisi
eib : En iyi bireyler indisi
f : Araç indisi
$f(b_i)$: b_i bireyinin çözüm değeri
gb : Geliştirilecek başlangıç bireyi indisi
i : Köşe noktası (düğüm) indisi
j : Köşe noktası (düğüm) indisi
k : Araç indisi
ks : Sıralama kriterinin kombinasyon sırası indisi
nib : Nesil içi birey sayacı
ns : Nesil sayacı
t : Hareket indisi
y : Hareket indisi
β : Arasında bırakma noktası olan koridorlar kümesi
θ : Arasında alma noktası olan koridorlar kümesi

ν : Çekici trenlerin hızı (metre/saniye)

Kısaltmalar

AGKA : Adaptif geniş Komşuluk Arama
AKKO : araç kapasitesi kullanım oranı
BÇDO : başlangıç çözüm değerleri ortalamasını
BD : Büyük Dikdörtgen
BK : Büyük Kare
CPU : Central Process Unit (Merkezi İşlem Birimi)
ÇÇD : Çakışmasız Çözüm Değeri
ÇD : Çözüm Değeri
ÇT : Çekici Tren
ÇYASÇAEA : Çift Yüzlü Ayna Sınır Teorisine Dayanan Çok Amaçlı Ayrıştırma Evrimsel Algoritması
DP : Dinamik Programlama
EÇÇD : En İyi Çakışmasız Çözüm Değeri
EÇD : En İyi Çözüm Değeri
EÇRÇP : En İyi Çakışmasız Rotalama ve Çizelgeleme Planı
EIBÇD : en iyi başlangıç çözüm değerini
EIÇD : en iyi çözüm değeri
EKBÇD : en kötü başlangıç çözüm değerini
FIFO : First In First Out (İlk Giren İlk Çıkar)
GA : Genetik Algoritma
GİA : Geri İzleme Algoritması
GKA : Geniş Komşuluk Arama
GKA : Geniş Komşuluk Araması
GÖA : Geniş Öncelikli Arama
HEBA : Hibrit Endokrin Bağışıklık Algoritması
K : Küçük
KA : Komşuluk Arama
KEÇD : Kalan Bireyler İçerisinden En İyi Olanın Çözüm Değeri
KKO : kapasite kullanım oranı
KKO : Karınca Kolonisi Optimizasyonu
MAYAK : Modifiye Edilmiş Ayrık Yapay Arı Kolonisi
MKKO : Memetik Karınca Kolonisi Optimizasyonu
MPSO : Modifiye Edilmiş Parçacık Sürü Optimizasyonu
NP : Non-polynomial (Polinamiyal Olmayan)
OD : Orta Dikdörtgen
OK : Orta Kare
PÖÇP : Probleme Özgü Çözüm Prosedürü
PSO : Parçacık Sürü Optimizasyonu
RAO : Rotalar arası aktarma operatörü
RDO : Rotalar arası değiştirme operatörü
SÇDRÇ : Sabit Çevrim Süreli Çakışmasız Döngüsel Rotalama ve Çizelgeleme
SÇRA : Sıralı Çakışmasız Rotalama Algoritmasının
Sig. : Significance (Önem değeri)
TA : Tabu Arama

TB	: Tavlama Benzetimi
TZKKAP	: Tam Zamanında Kapasite Kısıtlı Araç Rotalama Problemi
TZÜ	: Tam Zamanında Üretim
YA	: Yerel Arama

TAM ZAMANINDA ÜRETİM ORTAMINDA FABRİKA İÇİ DÖNGÜSEL TEDARİK SİSTEMİ İÇİN ÇAKIŞMASIZ ROTA OPTİMİZASYONU: MATEMATİKSEL MODEL VE MAT-SEZGİSEL ÇÖZÜM ÖNERİSİ

ÖZET

Tam zamanında üretim ortamında süreç içi envanter miktarının ve ara stokların en aza indirilmesi amacıyla çekici trenler kullanılarak iş istasyonlarına döngüsel seferlerle küçük partiler halinde ve yüksek frekansta parça tedariği yapılması oldukça yaygın bir uygulamadır. Ancak bu süreçte ortaya çıkabilecek gecikmeler veya aksamalar, iş istasyonlarında parça kıtlığı nedeniyle üretimin durmasına ve ciddi kayıpların oluşmasına sebep olacağı için fabrika içi tedarik sisteminin güvenilir ve robüst bir şekilde planlanması oldukça kritiktir. Bu bağlamda bu tez çalışmasında, özellikle hücresel yerleşim tipine sahip fabrikalarda çekici trenler için dar koridorlarda oluşabilecek çakışma durumları dikkate alınarak trafik kaynaklı beklemelerin ve gecikmelerin en aza indirildiği optimum rotaların ve çizelgelerin belirlenmesini hedefleyen Sabit Çevrim Süreli Çakışmasız Döngüsel Sefer Rotalama ve Çizelgeleme SSÇDRÇ problemi ele alınmıştır.

Problemin çözümüne yönelik olarak çakışma engelleme kısıtları içeren karma tam sayılı matematiksel bir programlama modeli geliştirilmiştir. Ancak bu model ile küçük sayılabilecek gerçek hayat problemleri dahi makul sürelerde çözülemediği için orta büyüklükteki problemleri optimum şekilde çözebilecek iteratif bir çözüm yaklaşımı önerilmiştir. Bu yaklaşımda, probleme özgü olarak geliştirilen karma tam sayılı matematiksel modeller kullanılarak çakışmalar iteratif olarak elenmektedir. Problemin NP-zor doğası nedeniyle büyük boyutlu gerçek hayat problemleri iteratif çözüm yaklaşımıyla da çözülemediği için problemin yapısı göz önünde bulundurularak üç aşamadan oluşan popülasyon ve komşuluk tabanlı özgün bir mat-sezgisel algoritma geliştirilmiştir. Önerilen çözüm yaklaşımlarının verimliliğini ve etkililiğini göstermek için gerçek hayat problemlerinden esinlenilerek farklı büyüklükte rassal olarak 60 adet test problemi oluşturulmuştur. Geliştirilen üç farklı çözüm yaklaşımının her biri ile test problemleri belirli bir süre çözülmüş ve elde edilen sonuçlar değerlendirilerek yönetimsel tavsiyeler sunulmuştur.

Anahtar Kelimeler: Çakışmasız Rotalama, Çekici Tren Rotalama ve Çizelgeleme, Döngüsel Sefer Rotalama, Fabrika İçi Lojistik.

CONFLICT-FREE ROUTE OPTIMIZATION FOR IN-PLANT MILK RUN SUPPLY SYSTEM IN JUST-IN-TIME PRODUCTION ENVIRONMENT: MATHEMATICAL MODEL AND MAT-HEURISTIC SOLUTION PROPOSAL

ABSTRACT

In order to minimize the amount of work-in-process inventory and intermediate stocks in a just-in-time production environment, it is a very common practice to supply parts in small batches and at high frequency with milk-run tours to the workstations using tow trains. It is very critical to plan the in-plant supply system in a reliable and efficient manner, since delays or disruptions that may arise in this process will cause production to stop and serious losses due to parts shortage in the workstations. In this context, in this thesis study, the problem of Fixed Cycle Duration Conflict-Free Milk-Run Routing and Scheduling (FCFMRS), which aims to determine optimum routes and schedules in which traffic-related waits and delays are minimized taking into account the conflict situations that may occur in narrow corridors for tow trains, especially in factories with cellular layout type, is studied.

To solve the problem, a mixed integer mathematical programming model that includes collision avoidance constraints has been developed. However, since even small real-life problems cannot be solved in a reasonable time with this model, an iterative solution approach that can solve medium-sized problems optimally has been proposed. In this approach, conflicts are eliminated iteratively by using mixed integer mathematical models developed specifically for the problem. Since large-scale real-life problems cannot be solved with the iterative solution approach due to the NP-hard nature of the problem, a unique mat-heuristic algorithm based on population and neighbourhood, consisting of three stages, has been developed considering the structure of the problem. In order to demonstrate the efficiency and effectiveness of the proposed solution approaches, 60 random test problems of different sizes were created, inspired by real-life problems. With each of the three different solution approaches developed, test problems were solved for a certain period of time and the results obtained were evaluated and managerial implications were presented.

Keywords: Conflict-Free Routing, Tow Train Routing and Scheduling, Milk-Run Routing, In-Plant Logistics.

1. GİRİŞ

Her ölçekten şirket günümüzün küresel rekabet ortamında varlığını sürdürebilmek için kaynaklarını daha etkin ve verimli kullanarak maliyetlerini azaltmak ve rekabet gücünü artırmak zorundadır. Bu durum, işletmelerin olabilecek tüm alanlarda maliyet iyileştirme çalışmalarına yönelmelerine neden olmaktadır (Gupta, 2011; Kumar, 2010). Endüstri bazında farklılıklar gösterse de lojistik faaliyetleri işletmeler için en önemli harcamalardan biri olarak ortaya çıkmaktadır (Atayah ve diğ., 2022). Bu açıdan bakıldığında lojistik faaliyetlerinde yapılacak her türlü maliyet azaltıcı ve verimlilik artırıcı çalışmanın birçok sektör için oldukça önemli rekabet avantajı kazandıracağını söylemek mümkündür (Tang ve Veelenturf, 2019).

Lojistik temel olarak, malzeme ve bilgi akışını tedarikçiler ve müşteriler arasında iki yönlü olarak sağlayan bütün faaliyetler olarak tanımlanabilir. Malzemelerin olabildiğince hızlı, sürekli, senkronize şekilde ve hiçbir noktada gereksiz bir envanter yığılması oluşmayacak biçimde transfer edilmesi oldukça önemlidir. Başarılı bir lojistik yönetimi sürecinde hammaddeler, ara mamuller ve tüm bileşenler zamanında, planlanan sırayla ve tam olarak ihtiyaç duyuldukları noktaya teslim edilmelidir (Crespo de Carvalho ve diğ., 2014). Envanter yığınlarının oluşması ve müşteri isteklerine yeterli hızda cevap verilememesi tedarik zinciri boyunca ciddi üretim kayıpları, maliyet artışları (elde bulundurma ve bulundurmama maliyeti vb.) ve nihayetinde rekabet avantajı kaybı olarak ortaya çıkacaktır (Harrison ve diğ., 2019).

Lojistik yönetiminde diğer bir anahtar nokta olan bilgi akışının, tüm tedarik zinciri boyunca iki yönlü ve kesintisiz şekilde gerçekleştirilmesi gerekmektedir. Nihai kullanıcıdan gelen satın alma talep bilgisi ile başlayan ve tedarik zinciri boyunca firmaların birbirine iletmeleri gereken parça ve ara mamul talep verileri ile tedarikçilerden perakendeciye kadar aşama aşama iletilen tedarik verilerinin doğru bir şekilde planlanıp kontrol edilebilmesi lojistik yönetiminin başarısı için kritik öneme sahiptir (Harrison ve diğ., 2019). Günümüz gelişmiş bilgi teknolojilerinin (BT) sağladığı olanakların da kullanılması ile tedarik zinciri boyunca entegre bir bilgi ve malzeme akışı oluşturmak zincirdeki bütün organizasyonlar için verimliliği ve karlılığı artırarak önemli bir rekabet avantajı sağlamaktadır (Cichosz ve diğ., 2020). BT'nin lojistik faaliyetlerinde kullanılmasının diğer bir avantajı günümüzde oldukça hızlı değişen müşteri ihtiyaçlarına

ve taleplerine tedarik zinciri boyunca çevik bir şekilde cevap verebilme yeteneğinin kazandırılabilmesidir (Gunasekaran ve diğ., 2017).

Bir ürünün hammadde ve/veya parçalarının tedarikçilerden alınıp üretiminin yapılması ve son kullanıcıya teslim edilmesine kadar bütün süreçlerinde rol oynayan lojistik faaliyetleri, tedarik lojistiği, fabrika içi lojistik ve dağıtım lojistiği olmak üzere üç ana başlığa ayrılabilir (Rushton ve diğ., 2006; Gülenç ve Karagöz, 2008). Literatürde iç lojistik, tesis lojistiği veya üretim lojistiği olarak da geçen fabrika içi lojistik, tedarikçilerden üretim tesisine kadar olan tedarik lojistiği ve üretim tesisinden son kullanıcıya kadar olan dağıtım lojistiği arasında yer alır. Genel olarak üretim tesisi içerisindeki bütün malzeme elleçleme, depolama ve transfer süreçlerini kapsayan fabrika içi lojistiğin temel görevi tesisin değer katan süreçleri için parça tedariğidir (Coelho ve diğ., 2021). İç lojistik kapsamındaki faaliyetler; mal kabul, depolama, sipariş toplama, çapraz sevkiyat, yükleme, hat besleme vb. olarak ifade edilebilir (Negri ve diğ., 2017).

Fabrika içi lojistik bir üretim tesisinin vazgeçilmez ve bütünleşik bir parçası olarak başarılı ya da başarısız olmasında oldukça önemli bir etkiye sahiptir (Coelho ve diğ., 2021). İyi planlanmış ve organize edilmiş iç lojistik sistemi, depolama ve envanter maliyetlerinin azaltılmasını, montaj veya iş istasyonlarının sorunsuz çalışmasını, katma değersiz faaliyetlerin ortadan kaldırılmasını, teslimat sürelerinin kısaltılmasını ve böylelikle genel üretim maliyetlerinin azaltılmasını sağlayarak şirketlere ciddi bir rekabet avantajı kazandırmaktadır (Efthymiou ve Ponis, 2019; Yener ve Yazgan, 2019; Usuk ve Selvi, 2019). Ayrıca başarılı bir iç lojistik yönetimi ile toplama, dağıtma, yükleme ve istifleme gibi süreçlerde kullanılan malzeme elleçleme ve taşıma ekipmanları ile işgücünün verimliliğinin de artırılması sonucunda daha esnek ve daha karlı bir üretim yapmak mümkün olabilmektedir (Thanou ve Matopoulos, 2021).

Standart olmayan ve iyi yönetilmeyen iç lojistik ise fabrika içinde katma değersiz faaliyetlerin artmasına, hat kenarlarında gereksiz envanter birikmelerine veya zamanında ve yeterli miktarda tedarik edilemeyen malzemeler nedeniyle istasyonların ve hatta bütün üretim sürecinin duraksamasına neden olarak maliyet artışına sebebiyet verebilmektedir (Küçükoğlu ve diğ., 2018). Üretimdeki bu tür aksamalar sonucunda nihai ürünlerin müşterilere sevkiyat zamanlarına uyulamaması sadece para değil güven kaybına da yol açmakta ve gelecekteki fırsatları yok etmektedir. Bu nedenle yüksek karlı ve güvenilir bir üretim sistemi için iç lojistiğin önemi oldukça yüksektir (Coelho ve diğ., 2021). Lojistik

faaliyetlerinin standardize edilmesi ve periyodik hale getirilmesi, maliyetinin azalmasını ve daha kolay yönetilmesini sağlamaktadır (Küçükoğlu ve diğ., 2018).

En önemli hedefi süreç içi envanterin ve stokların azaltılması yoluyla karlılığın artırılması olan Tam Zamanında Üretim (TZÜ) sistemi 1950'li yıllarda Japonya'da geliştirilmiş ve başarısı birçok uygulamada kanıtlanmıştır (Gupta, 2011). Belekoukias ve diğ. (2014)'e göre TZÜ, gerekli olan parça veya ürünleri, gerekli olduğu zaman ve gerekli olduğu kadar üretmeye dayalı bir üretim yönetimi şeklidir. Yalın üretim felsefesinin bir aracı olarak görülen TZÜ, üretimin her aşamasında israfları yok etmeyi ya da mümkün olduğunca azaltmayı önceleyen bir sistemdir. (Fullerton and McWatters 2001; García-Alcaraz ve Maldonado-Macías, 2016). Üretilen ürün ya da parça üzerinde müşteri gözünde herhangi bir katma değer oluşturmayan bütün işler olarak tanımlanan israfların arasında en önemlileri olarak gereksiz süreç içi envanter ve bitmiş ürün stokları, taşımalar, işçi hareketleri ve beklemeler sayılabilir (Htun ve diğ., 2019). Bu israfların ortadan kaldırılması veya en azından mümkün olduğunca azaltılması ile daha düşük maliyetli, daha esnek ve değişen müşteri taleplerine hızlı yanıt verebilen daha çevik bir üretim yaparak rekabet avantajı sağlamak mümkün olmaktadır.

TZÜ sistemi, istasyonlar arasında tek parça akışını en ideal durum olarak benimseyen ve işlenecek parçalarla üretim/montaj istasyonlarının birbirini bekletmediği pürüzsüz ve sürekli akışa sahip bir üretim ortamını ifade eder (Şeker, 2016; McLachlin 1997). Süreç içi envanter ve bitmiş ürün stoklarının minimuma çekilmesi hatta en ideal durumda yok edilmesi, elde bulundurma maliyetlerinin azaltılmasına ek olarak üretim sisteminin değişen taleplere daha hızlı cevap verebilmesi ve stoklama için kullanılacak alanların katma değerli faaliyetlere tahsis edilerek verimliliğin artırılabilmesi gibi birçok fayda sağlamaktadır (Belekoukias ve diğ., 2014; Kumar 2010).

Stok seviyelerinin minimum düzeye indirilmesi üretim veya montaj istasyonlarına küçük partiler halinde ve yüksek frekansta sürekli olarak parça tedariğini gerektirmektedir. Küçük partiler halinde sık ve sürekli parça tedariği; kalite, verimlilik, maliyet ve erken kusur tespiti gibi birçok avantaja sahip olsa da tedarik konusunda yaşanacak problemler veya gecikmelerin üretim sisteminin bir bölümünü veya tamamını sekteye uğratması gibi ciddi problemlere ve maddi kayıplara neden olabileceği açıktır (Kumar, 2010). Bu nedenle TZÜ sisteminde fabrika içi lojistik faaliyetleri klasik üretim sistemlerine göre çok daha önemli ve kritik bir konuma sahiptir (Htun ve diğ., 2019). TZÜ ortamında küçük

partiler halinde sürekli ve yüksek frekansta parça tedariği, genellikle çekici tren sistemleri kullanılarak döngüsel dağıtım seferleri ile yapılmaktadır (Emde ve Gendreau 2017).

TZÜ sisteminin etkili bir biçimde uygulanmasını sağlayan en önemli araçlardan birisi hücresel imalat yaklaşımıdır (Wang, 2015; García-Alcaraz ve Maldonado-Macías, 2016; Kumar, 2010). Grup teknolojisinin üretimde uygulaması olarak değerlendirilen hücresel imalat, benzer özelliklerine göre gruplanmış parça ailelerinin üretim hücrelerinde işlendiği bir imalat sistemidir (Reeb ve diğ., 2010; Negahban ve Smith, 2014). Üretim hücreleri farklı ve/veya aynı fonksiyonlara sahip makine ve teçhizat ile bir veya birden çok makinede çalışan birkaç işçiden (genellikle 2-10 arası) oluşan bağımsız ve genellikle küçük bir imalat alanıdır (Reeb ve diğ., 2010). Hücresel üretimde, parça ve makine ailelerinin oluşturulması, üretim hücresinin tasarımı, parça giriş çıkış noktalarının belirlenmesi ve malzeme akışının organize edilmesi gibi üzerinde durulması gereken birçok konu vardır (Raminfar ve diğ., 2013). Farklı tipte imalat hücresi tasarımları olsa da literatürde en çok kullanılan tasarımın verimliliği ve esnekliği artırdığı için U tipi tasarım olduğu görülmektedir (Şeker, 2016; URL-1).

Atölye tipi üretim sisteminin çok çeşitli ürünleri üretebilme esnekliğini ve akış tipi üretimin yüksek üretim kapasitesi ve etkili malzeme akışı avantajlarını bir arada sunan hücresel imalat sisteminin yapılan uygulama çalışmalarında birçok faydası raporlanmıştır (Tariq ve diğ., 2009; Raminfar ve diğ., 2013). Bunlar arasında en belirgin olarak; malzeme ve bilgi akışının kolaylaşması, malzeme elleçleme sürelerinin ve maliyetlerinin azalması, stok ve süreç içi envanterin azalması, hurda ve atık oranlarının azalması, teslim sürelerinin kısalması ve üretim alanının daha verimli kullanılması sayılabilir (Negahban ve Smith, 2014, Neto ve Filho, 2010; Shanker ve Vrat, 1999; Egilmez ve diğ., 2012; Murugan ve Selladurai, 2007). TZÜ sistemi ile birlikte yaygın olarak kullanılan ve gün geçtikçe çeşitli sektörlerdeki üretim çevrelerinde yaygınlaşmaya devam eden hücresel imalat için fabrika içi lojistik sistemi oldukça önemlidir (Azadeh ve diğ., 2010; Pitchuka ve diğ., 2006; Neto ve Filho, 2010). Birbirinden bağımsız olarak üretim yapan imalat hücrelerinin etkin ve verimli bir şekilde çalışabilmesi için her birine parça tedariği kusursuz şekilde yapılmalıdır.

Bu çalışmada TZÜ ortamında özellikle hücresel imalat sistemi ile üretim yapan tesislerde döngüsel seferlerle (ing: milk-run) üretim hücrelerine merkezi olmayan stok alanlarından malzeme tedariği sağlayan çekici trenler için çakışmasız rota optimizasyonu ele

alınmıştır. Bu noktadan sonra Sabit Çevrim Süreli Çakışmasız Döngüsel Rotalama ve Çizelgeleme (SÇDRÇ) problemi olarak anılacak yeni bir problem tanımlanmış ve çözüm yöntemleri geliştirilmiştir. TZÜ sisteminde hücre veya hat kenarı stokların minimum seviyede tutulabilmesi ve üretimin malzeme yetersizliği sebebiyle aksamaması için istasyonlara az miktarda ve yüksek sıklıkta parça tedariğinin sorunsuz ve güvenilir bir biçimde yapılması kritik öneme sahip olmaktadır (Hasani Goodarzi ve Zegordi, 2020; Staab ve diğ., 2016). Bu sebeple TZÜ sisteminin yaygın olarak kullanıldığı ve yüksek hacimli üretim yapan otomotiv, elektronik, mobilya ve beyaz eşya gibi sektörlerde süpermarket denilen merkezi olmayan stok alanlarından üretim hücrelerine veya montaj istasyonlarına çekici tren sistemleri kullanılarak döngüsel seferlerle malzeme tedariği oldukça yaygın bir uygulamadır (Mao ve diğ., 2020; Boysen ve diğ., 2015; Kılıç ve diğ., 2012; Satoglu ve Sahin, 2013; Chen ve diğ., 2023). Döngüsel seferler ile taşıyıcı araçların çevrimsel olarak belirli rotaları takip ederek birçok tüketici istasyonun malzemesini tek seferde tedarik etmesi sağlanmaktadır (Satoğlu ve Sipahioglu, 2018). Bu tip bir dağıtım doğrudan seferle yapılan dağıtımlara göre daha az maliyet, taşıyıcı araçların ve operatörlerin daha verimli çalışması ve yönetim kolaylığı gibi birçok avantaja sahiptir (Kluska ve Pawlewski, 2018; Satoglu ve Sahin, 2013). TZÜ sisteminde oldukça yaygın olan süpermarketler ise üretim hücrelerine çok yakın bir şekilde konumlandırılmış merkezi olmayan stok alanlarıdır. Bu yaklaşım ile üretim hücrelerine sık, hızlı ve sürekli malzeme tedariğinin daha az maliyetle yapılabilmesi mümkün olmakta ve böylelikle üretim hücreleri kenarındaki stoklar azaltılabilmektedir (Alnahhal ve Noche, 2015; Battini ve diğ., 2015; Battini ve diğ., 2013).

SÇDRÇ probleminde fabrika içindeki merkezi olmayan stok alanlarından üretim hücrelerine malzeme tedariği yapan çekici trenlerin sefer sürelerinin toplamının minimizasyonu hedeflenmektedir. Bu süreler içerisinde malzeme alma ve bırakmadan kaynaklanan süreler, yolda geçen süreler ve trafik sebebiyle oluşan bekleme süreleri bulunmaktadır. Böylelikle hem çekici trenlerin olabilecek en kısa yoldan giderek üretim hücrelerine malzeme beslemesi yapması, hem de trafik kaynaklı olabilecek bekleme sürelerinin en aza indirilmesi amaçlanmaktadır. SÇDRÇ probleminin çözümü ile trafik kilitlenmelerinin ve öngörülmeyen beklemelerin olmadığı güvenilir bir dağıtım sistemi elde etmek mümkün olmaktadır.

Bu çalışmada çakışmasız şekilde rotalanması, yüklenmesi ve çizelgelenmesi hedeflenen çekici trenler (ing: tow truck /tugger train); otomatik olarak çalışan veya bir operatör

tarafından kullanılan çekici bir lokomotif ile ona bağlı çeşitli tipteki römorklardan oluşan ve genellikle fabrika içi lojistik için kullanılan taşıyıcı araçlardır (González-Hernández ve diğ., 2022; Battini ve diğ., 2015). TZÜ yaklaşımını benimsemiş ve yüksek hacimli üretim yapan sektörlerde (otomotiv gibi) forkliftlere ve otomatik yönlendirmeli araçlara göre yüksek taşıma kapasitesi, kullanım kolaylığı, enerji tasarrufu ve çok yönlü olmaları gibi avantajları sebebiyle özellikle döngüsel seferlerle parça tedariğinde yaygın bir biçimde kullanılmaktadırlar (URL-2; Alnahhal ve Noche, 2013; Hanson ve Finnsgård, 2014; Gyulai ve diğ., 2013; Emde ve Gendreau 2017).

Literatürde fabrika içi döngüsel seferler için çevrim süresini sabit veya değişken olarak ele alan çalışmalar mevcuttur (Kılıç ve diğ., 2012). Ancak incelenen fabrikalardan elde edilen gözlemler ve yalın üretim felsefesi gereği sabit çevrim süreli döngüsel seferlerin, standardizasyon açısından çok daha iyi olduğu anlaşılmaktadır. Ayrıca çevrim süresinin sabit olması, çekici trenleri kullanan operatörler ve sistemi kontrol eden çalışanlar açısından yönetilmesinin çok daha kolay olması sebebiyle tercih edilmektedir (Duman, 2016; Alnahhal ve Noche, 2013). Hatta Bozer ve Ciemnoczolowski (2013) gibi bazı çalışmalar sabit çevrim süresini sistemin döngüsel sefer sistemi olması için zorunluluk olduğunu ifade etmektedirler. Bu çalışmada da SÇDRÇ problemi için döngüsel sefer sistemi sabit çevrim süreli olarak ele alınmıştır. Yani çekici trenlerin her biri aynı zamanda başlangıç noktasından hareket etmekte ve belirli bir çevrim süresi sonunda başlangıç noktalarına dönerek bir sonraki tur için beklemektedir.

SÇDRÇ problemi, çekici trenlerin birbirlerini bloklamalarını ve trafik kaynaklı plansız gecikmelerini engelleyecek bir şekilde rotalanmasını, yüklenmesini ve çizelgelenmesini içermektedir. Nitekim fabrika içinde trafik oluşması taşıyıcı araçların hem parça tedariğini zamanında yapamamasına hem de fabrika içinde karmaşaya sebebiyet vereceği için üretim süreçlerinin aksamasına neden olmaktadır. TZÜ ortamında çekici trenlerin döngüsel seferlerinin rotalanması ile ilgili literatür incelendiğinde, yapılan çalışmaların daha çok montaj hattı besleme konusunda olduğu dikkat çekmektedir. Montaj hatlarının kenarlarındaki yollar genellikle iki çekici trenin yan yana geçebileceği şekilde tasarlandığından trenlerin birbirlerini bloklamaları ya da bekletmeleri olası değildir. Montajlanan ürünler ya da modeller değişse bile montaj hatlarının genişleyerek yan taraflarında bulunan yolları çekici trenlerin ikisinin birden geçemeyeceği şekilde daraltması durumu söz konusu değildir. Bu nedenle montaj hattı besleme ile ilgili çalışmalarda trafik oluşması durumu rotalama ve çizelgeleme problemlerinin

kurgularında göz ardı edilmektedir (Mao ve diğ., 2020; Buyukozkan ve diğ., 2019; Emde ve Gendreau, 2017; Zhou and Peng, 2017; Peng ve diğ., 2020; Emde ve Schneider, 2018; Alnahhal ve Noche, 2013). Emde ve Boysen (2012a) gibi bazı çalışmalarda montaj hatlarına parça beslemesi yapan çekici trenlerin trafik karmaşası ihtimali olmaması için hizmet verdiği istasyonların ardışık olması şartı koşulduğu ve böylece koridorlarda trafik ve karmaşanın engellenmesinin hedeflendiği görülmektedir. Duman (2016) gibi bir kısım çalışmalarda ise koridorların tek yönlü olarak kullanılması varsayımı sunulmakta ve böylece çekici trenlerin karşılıklı olarak çakışmaması sağlanmaktadır. Ancak bu yaklaşımların bir kısmının hücresel imalat sistemi ile çalışan fabrikalara uygulanması pratik açıdan mümkün olmadığı, bir kısmının da etkili sonuçlar doğurmayacağı açıktır. Çünkü hücresel imalat ile çalışılan fabrikalarda montaj hatlarının aksine uzun bir hat yerine çok sayıdaki üretim hücresi arasında taşıyıcı araçların dolaşması gereken dikey ve yatay düzlemde çok daha fazla koridor bulunmaktadır. Ayrıca bu tür yaklaşımların çekici trenlerin kapasitelerinin yeterince etkin kullanılamamasına veya çözüm uzayının kısıtlanması sonucu daha iyi bir çözümden vazgeçilmesi anlamına gelmektedir.

Özellikle otomotiv, tüketici elektroniği ve beyaz eşya gibi sektörlerde hücresel imalat ile çalışılan fabrikalarda üretilen ürünlerin modellerinin ve tasarımlarının zamanla değişmesi sebebiyle üretim hücrelerinde imal edilen parçaların şekilleri, boyutları ve hatta üretim yöntemleri değiştiği için hücrelerin tasarımları değişmektedir. Bu şekilde tasarımı ve boyutları değişen hücreler sebebiyle, başlangıçta iki çekici trenin yan yana geçebileceği boyutta tasarlanmış olsa da zamanla fabrikadaki bazı koridorlar daralabilmekte ve iki çekici trenin yan yana geçemeyeceği duruma gelmektedir. Dar koridorlar olarak adlandırılan bu koridorlarda çekici trenlerin karşı karşıya gelip birbirlerini kilitlemesini veya duran bir trenin diğerlerini bloklayarak bekletmesini engelleyecek şekilde döngüsel rotaların belirlenmesi ve araçların çizelgelenmesi gerekmektedir. Literatür incelendiğinde hücresel imalat özelinde yapılan fabrika içi çekici trenler için döngüsel sefer rotalaması konusundaki çalışmaların neredeyse hiçbirinde çakışma durumlarının dikkate alınmadığı görülmektedir (Kılıç ve Durmuşoğlu, 2013; Gyulai ve diğ., 2013; Akıllıoğlu ve diğ., 2006). Bu çalışmada SÇDRÇ problemi ile yukarıda bahsedilen sebeplerle dar koridorlarda oluşabilecek trafik sıkışmalarını göz önüne alan bir problem tanımı ve çözümü sunulmaktadır. Böylelikle daha güvenilir, robüst ve sorunsuz bir malzeme tedarik süreci gerçekleştirilebilmesi hedeflenmektedir.

SÇDRÇ problemi için karma tam sayılı matematiksel bir model (Model-1) önerilmiştir. Probleminin kombinatoryal yapısı nedeniyle optimum çözümü bulmak orta boyutlu problemlerde bile polinom zamanda mümkün olmadığı için birinci modelin gevşetilmiş hali olan Model-2 ve üçüncü bir modelin (Model-3) birlikte kullanımı ile çakışmaların iteratif olarak engellendiği ikinci bir çözüm yaklaşımı önerilmiştir. Bu yaklaşımla orta boyutlu problemlere optimum çözümler bulunabilse de büyük boyutlu gerçek hayat problemleri için çözüm bulabilmek mümkün olmamaktadır. Bu nedenle probleme özgü popülasyon ve komşuluk tabanlı, üç aşamalı yeni bir mat-sezgisel algoritma geliştirilmiştir. Bu algoritma başlangıç çözüm oluşturma, çözüm geliştirme ve çakışmasız çözüm arama olmak üzere üç ana aşamadan oluşmaktadır. Önerilen bu üç çözüm yaklaşımı çeşitli parametrelere göre rasgele oluşturulan farklı büyüklükteki ve özellikteki test problemleri ile test edilmiştir.

Bu çalışmanın literatüre katkıları şu şekilde özetlenebilir;

1. TZÜ sisteminde hücresel imalat ile üretim yapan fabrikalarda döngüsel seferlerle malzeme tedariki yapan çekici trenlerin çakışmasız olarak rotalanması, yüklenmesi ve çizelgelenmesi için SÇDRÇ probleminin tanımlanması,
2. SÇDRÇ probleminin çözümü için karma tam sayılı matematiksel bir model (Model-1) geliştirilmesi (birinci çözüm yaklaşımı),
3. Orta boyutlu problemlerin optimum şekilde çözülebilmesi için Model-1'in gevşetilmiş hali olan Model-2'nin ve üçüncü bir modelin (Model-3) birlikte kullanılmasıyla iteratif bir çözüm yaklaşımının geliştirilmesi (ikinci çözüm yaklaşımı),
4. Büyük boyutlu gerçek hayat problemlerinin makul sürelerde çözümü için probleme özgün olarak popülasyon ve komşuluk tabanlı yeni bir mat-sezgisel algoritmanın önerilmesi (üçüncü çözüm yaklaşımı),
5. Çeşitli parametrelere göre rassal olarak oluşturulmuş test problemleri.

Çalışmanın bundan sonraki kısımları şu şekilde organize edilmiştir; ikinci bölümde TZÜ ortamında fabrika içi lojistik faaliyetlerine ilişkin literatür taraması sunulmaktadır. Üçüncü bölümde SÇDRÇ problemi tanımlanmakta, dördüncü bölümde problemin çözümü için geliştirilen matematiksel modeller ve mat-sezgisel algoritmadan oluşan çözüm yaklaşımları sunulmaktadır. Beşinci bölümde, geliştirilen çözüm yaklaşımlarını sınamak için oluşturulan test problemleri ve çözüm sonuçları verilerek yorumlanmakta ve tartışılmaktadır. Altıncı bölümde ise çalışmadan elde edilen bulgular değerlendirilmekte ve gelecek çalışmalara dair bilgiler verilmektedir.

2. LİTERATÜR TARAMASI

Temel rolü tesis içerisindeki değer katan süreçlere malzeme tedariği yapmak olan fabrika içi lojistik, bütün hammadde, ara mamul ve mamul elleçleme süreçlerini kapsamaktadır. Bunlar içerisinde tesise gelen malzemeleri veya tesis içerisinde işlenmiş yarı mamul ve mamulleri alma, stoklama, sipariş için toplama, sevk etme ve malzemelerin içerisinde taşındığı boş kutuları toplama bulunmaktadır (Coelho, 2021; Negri, 2017). Üretim süreçlerinin sorunsuz ve akıcı bir şekilde devam etmesinde kilit rolü olan iç lojistik, sıfır stok veya minimum seviyede stok ile üretim yapmayı hedefleyen TZÜ sistemi felsefesi ile çalışan şirketler için çok daha kritik bir öneme sahiptir.

Günümüzde TZÜ sisteminde çeşitli parçaların çok sayıdaki iş istasyonuna çevrimsel seferlerle tedarik edildiği fabrika içi lojistik sistemi sağladığı maliyet, kolay yönetilebilirlik ve çeviklik gibi avantajlar sebebiyle gittikçe daha cazip hale gelmektedir. Bu nedenle TZÜ ortamında iç lojistik faaliyetlerine ilişkin çeşitli alanlardan araştırmacılar ve uygulayıcılar tarafından yapılmış farklı kapsamda ve farklı noktalara odaklanan çok sayıda çalışma mevcuttur (Chen ve diğ., 2023).

Bu çalışmada özellikle TZÜ ortamında fabrika içinde üretim istasyonlarına malzeme tedariki ele alındığından bu alanda yapılan çalışmalar başta Web of Science, Scopus, Elsevier, Springer, Emerald, and IEEE Transactions veri tabanları olmak üzere birçok veri tabanı taranarak incelenmiştir. Fabrika içerisinde iş istasyonlarına parça tedariği ile ilgili çalışmalar, süpermarketlerin yerini ve sayısını belirlemeye yönelik olarak fabrika yerleşim planının düzenlenmesinden, farklı besleme stratejilerinin seçilmesine, değişik işlev ve fonksiyonlara sahip taşıma araçları içerisinden en uygun olanının seçilmesinden, bu araçların rotalanması, çizelgelenmesi ve yüklenmesine kadar çok çeşitli problemleri kapsamaktadır. Bu kapsamda yapılan çalışmalar genel olarak şu üç başlık altında gruplandırılabilir (Battini ve diğ., 2015):

1. İş istasyonları için farklı parça besleme politikaları/stratejileri arasında karar verilmesi,
2. Fabrika içerisinde süpermarketlerin yerlerinin ve sayısının belirlenmesi,
3. Çekici tren sistemlerinin veya kullanılan diğer taşıyıcı araçların rotalanması, çizelgelenmesi ve yüklenmesi.

2.1. Parça Besleme Politikaları

Parça besleme politikaları, üretim veya montaj istasyonlarına parçaların her birinin ne şekilde dağıtılacağının belirlenmesi ile ilgilidir. Bu konuda yaklaşık çeyrek asırdır yeni parça besleme politikaları önerme, politikalar arasında en verimli olanı seçme veya farklı politikaları bir araya getirerek hibrit besleme politikaları üretme üzerine araştırma yapan birçok çalışma olduğu görülmektedir. Schmid ve Limère (2019) parça besleme ile ilgili literatürü kapsamlı şekilde inceleyerek bir sınıflandırma yapısı önermişler ve literatürü sınıflandırmışlardır. İnceledikleri makalelerdeki parça besleme politikalarını genel olarak; hat yanı stoklama, kutulu tedarik, sıralama ve kit oluşturma (sabit veya hareketli) olmak üzere dört ana sınıf altında toplamışlardır. Ayrıca parçaların besleme politikalarına atanmasını ele alan ve besleme maliyetlerinin minimizasyonunu hedefleyen temel bir 0-1 tam sayılı matematiksel model sunmuşlardır.

Caputo ve Pelagagge (2011) karar vericilerin iş istasyonlarına malzeme tedariki için en uygun besleme politikasını değerlendirmelerine ve seçmelerine yardımcı olacak bir metodoloji önermeyi amaçlamışlardır. Çalışmalarında, sınıf tabanlı hibrit politikalar da dahil olmak üzere tam zamanında kanban tabanlı sürekli tedarik ve hat yanı stoklama olmak üzere üç temel politikaya dayandırmışlardır. Alternatif politikalar arasından seçim yaparken fabrika alanı kullanımı, süreç içi envanter kaynaklı maliyet, ilk yatırım maliyeti, işletme maliyeti kriterlerini göz önünde bulundurmuşlardır. Bir vaka çalışması ile sundukları karar verme aracını örneklendirmişlerdir.

Besleme politikalarının seçimi ile ilgili diğer bir çalışmada Limère ve diğ. (2012) montaj hattına tedarik edilecek her bir parçanın, kit oluşturma ve hat yanı stoklama olmak üzere iki olası besleme politikasından hangisine atanması gerektiğini belirlemek amacıyla karma tam sayılı bir matematiksel model önermişlerdir. Otomasyon sistemi düşünmedikleri için geliştirdikleri modelde ilk yatırım maliyetlerini ihmal ederek sadece, taşıma maliyetleri ile kit montaj işlemleri ve süpermarkette ikmal için çalışacak işçilerin ortalama yıllık maliyetlerini dikkate almışlardır. Modeli test etmek için Belçika'daki bir otomotiv şirketinden alınan vaka verilerini kullanılmışlar ve sonuç olarak bazı parçaların kitleme sistemi ile diğerlerinin ise montaj hattına toplu olarak beslendiği hibrit politikaların tercih edilmesinin daha avantajlı olduğunu vurgulamışlardır.

Sali and Sahin (2016) her bir parçanın hat yanı stoklama, kit oluşturma ve sıralama modlarından oluşan parça besleme politikalarından en verimli olana atanması için bir karar destek modeli oluşturmuşlardır. Önerdikleri modelde, montajdan önce parça hazırlama, sevk etme, parça toplama ve stoklamadan kaynaklanan ortalama toplam işletme maliyetini minimize etmeyi amaçlamışlardır. Otomobil üreticileri için araba koltukları monte eden birinci kademe bir tedarikçi fabrikasında yürüttükleri bir vaka çalışmasına dayanarak, optimizasyon modeli tarafından sağlanan çözüm kullanarak %10'a varan maliyet tasarruflarının elde edilebileceğini göstermişlerdir

Battini ve diğ. (2009) karma model montaj hatları için parça ve mamul yönetiminin optimizasyonuna yönelik yenilikçi ve bütünleşik bir yaklaşım sunmayı hedeflemişlerdir. Bu amaçla çalışmalarında, fabrika içinde toplam depolama maliyetlerini en aza indirebilmek için parçaların depolanmasının merkezi olması ya da olmaması ve doğru depolama ile besleme politikalarının belirlenmesi konularını ele almışlardır. Hat yanı stoklama, sabit kit kullanımı, hareketli kit kullanımı olmak üzere üç farklı besleme modu düşünerek tüm hat için tek bir besleme modu seçilmesini varsaymışlardır. Çok faktörlü bir analiz yaparak her bir parça için depolama, paketleme, alma ve taşıma sorunlarının çözümüne yönelik bir çerçeve geliştirmişlerdir. Endüstriyel bir uygulama ile tanımladıkları çerçevenin geçerliliğini ve kullanılabilirliğini göstermişlerdir.

Hua ve Johnson (2010) bir elektronik montaj şirketinde kit oluşturma işlemlerinin ve hat yanı stoklamaya potansiyel geçişin analizine dayalı olarak seçim kararını etkileyebilecek bir dizi araştırma konusunu tanımlamışlardır. Üretim hacmi, bileşen çeşitliliği ve boyutu faktörlerinin bu kararı vermede önemli faktörler olduğunu ifade etmişlerdir. Yazarlar nispeten geniş bileşen çeşitliliği olduğu durumlarda kit oluşturma daha avantajlı iken, ürünlerin benzer bileşenleri kullandığı durumlarda hat yanı stoklama politikasının muhtemelen en iyi seçenek olacağını vurgulamışlardır.

Hanson ve Brolin (2013) İsveç otomotiv montaj endüstrisindeki iki vaka çalışmasından elde ettikleri verilerden yararlanarak kit oluşturma sistemi ile sürekli tedariği (hat yanı stoklama) iş gücü tüketimi, ürün kalitesi, esneklik, envanter seviyeleri ve fabrika alanı kullanımı faktörleri açısından karşılaştırmışlardır. Kit oluşturma stratejisinde kitlerin hazırlanması için harcanan sürenin montaj hattında tasarruf edilen süreden fazla olduğunu ve kit hazırlama için kullanılan iş gücünün sürekli besleme stratejisinde kullanılandan

fazla olduğunu ancak kit stratejisinin ürün kalitesini iyileştirdiğini ifade etmişlerdir. Sürekli besleme stratejisinden kit oluşturmaya geçerken envanter seviyelerinin nispeten aynı kaldığını belirtmişlerdir. Kit stratejisinde montaj tesisindeki genel alan gereksinimlerinin (kit hazırlama alanı dahil) belirli bir donanım alanının getirilmesinin bir sonucu olarak artsa bile, kit oluşturma stratejisinin sunduğu en ilginç avantajlardan birinin montaj hattında yer açması olduğunu vurgulamışlardır.

Sali ve diğ., (2015) hat yanı stoklama, kit oluşturma ve sıralama olmak üzere üç farklı hat besleme politikası için hazırlama, sevk etme, stoklama ve hat kenarında alma süreçlerini göz önünde bulundurarak toplam ortalama maliyetleri formüle etmişlerdir. Çoklu senaryo analizi ile maliyet formülasyonunda kullanılan parametrelerin hat stoklama, kit oluşturma ve sıralama modlarının göreli performansında kritik bir rol oynadığını göstermişlerdir.

Faccio (2014) üretilen model çeşitliliğinin etkisini göz önünde bulundurarak montaj hattı besleme için hat yanı stoklama, kanban sürekli tedarik (kutulu tedarik) ve kit oluşturma politikalarının maliyet karşılaştırmasını yapmıştır. Bir endüstriyel vaka çalışması ve bir simülasyon çalışması aracılığıyla, en iyi besleme politikasını belirlemek için bir karar alma aracı sunmuştur.

Baller ve diğ. (2020) montaj hattına tedarik edilecek her bir parça için dokuz farklı hat besleme modu içerisinden en etkili olanını seçmek için karma tam sayılı matematiksel bir programlama modeli önermişlerdir. Toplam montaj maliyeti, süpermarkette hazırlık maliyeti, süpermarkete parçaları tedarik etme maliyeti, sevkiyat maliyeti ve ekipman maliyetinin birleşiminden oluşan toplam besleme maliyetini minimize etmeyi hedefledikleri modellerinde hat kenarında ve süpermarkette işgal edilen alan ve süpermarketteki değişken yürüme mesafeleri ile ilgili kısıtlara yer vermişlerdir.

Kilic ve Durmusoglu (2015) üretim hatlarına parça beslemesi ile ilgili bir literatür taraması sunmuşlardır. Çalışmalarında öncelikle parça besleme sistemlerini parçaların depolanması, taşınması ve parça besleme politikasını içeren bir sınıflandırma yapısı altında ortaya koymuşlardır. Ayrıca her bir ana bileşen için ilgili alt bileşenleri farklı sınıflandırma yapılarında başlıklara göre açıklamışlardır. Parça besleme sistemlerinin ana ve alt bileşenlerini belirleyip açıklandıktan sonra, parça besleme sistemlerinin üç ana bileşeninden biri olan parça besleme politikası üzerinde durmuşlardır. Parça besleme

politikaları literatürünü, hat yanı stoklama, kit oluşturma, kanban tabanlı besleme ve hibrit besleme olmak üzere dört bileşen altında analiz ederek ele almışlardır. Her bir bileşen için ilgili literatürü taramışlar ve önemli sayıda çalışmayı kısaca açıklamışlardır. Ayrıca besleme politikası türleri, amaçları, çözüm metodolojileri ve uygulama türleri ile ilgili ayrıntılı bir analiz sunmuşlardır.

Limère ve diğ. (2015) parçaların kit oluşturma ve hat yanı stoklama stratejilerden herhangi birine atanması için değişken operatör yürüme mesafelerini dikkate alan karma tam sayılı matematiksel bir karar modeli önermişlerdir. Ayrıca spesifik parça özelliklerinin bir parçanın kit stratejisine atanma şansını nasıl etkilediğini gösteren bir analiz sunmuşlardır. Çalışma sonucunda, kit oluşturma stratejisinin yürüme mesafelerini azaltma potansiyelinin onu çok sayıda donanıma sahip büyük iş istasyonları için bile daha cazip kıldığını göstermişlerdir.

Caputo ve diğ. (2015a) montaj hatlarına sürekli malzeme tedariki için tam zamanında teslimat ve hat yanı stoklama alternatiflerini karşılaştırmak amacıyla analitik planlama modelleri geliştirmişlerdir. Böylelikle üretim yöneticilerine ve tesis planlamacılarına alternatif sürekli tedarik politikalarını erken bir karar aşamasında değerlendirmek için nicel bir karar aracı sunmayı amaçlamışlardır. Çalışma sonucunda tam zamanında teslimat modelinin, hat yanı stoklama modeline göre karma model montaj hatlarına daha uygun olduğunu ifade etmişlerdir.

Karadayi Usta ve diğ. (2017) parça besleme sistemi seçimi için bir metodoloji geliştirmişlerdir. Çalışmalarında montaj hatlarında kullanılacak hibrit besleme sisteminin tasarımı için hiyerarşik kümeleme analizini önermişlerdir. Ek olarak, alternatif parça besleme sistemlerini karşılaştırmak ve değerlendirmek için ABC metodolojisinden yararlanmışlardır. Bu yeni birleşik metodolojiyi parça besleme sistemini tasarlamak ve karar vermek için önermişlerdir. Metodolojinin etkinliğini ve geçerliliğini göstermek için bir şirkette, en çok üretilen ürünü dikkate alarak vaka çalışması yapmışlardır. Sonuç olarak karışık modelli montaj sistemlerinde parçaların büyük ve pahalı olması nedeniyle sadece kit oluşturma sistemlerinin kullanılmasının sorunlara yol açabildiğini, bu nedenle hibrit besleme politikasının bu tür sistemler için maliyeti önemli ölçüde azaltarak daha iyi çözümler sağlayabildiğini ifade etmişlerdir.

Caputo ve Pelagagge (2011) yöneticilerin deterministik bir çevrede üretim hatlarına parça besleme konusunda çeşitli politikaları değerlendirmesine ve en uygun olanının belirlemesine yardımcı olacak bir metodoloji önermeyi amaçlamışlardır. Bu kapsamda her bir parça tipi için; kit oluşturma, tam zamanında kanban tabanlı sürekli tedarik ve hat yanı stoklamadan oluşan besleme politikalarından en etkili olana karar verebilmesi için bir yöntem önermişlerdir. Önerdikleri yöntemin uygulamasını örneklendirmek ve bir karar verme aracı olarak yeteneklerini göstermek için bir vaka çalışması sunmuşlardır.

Müllerklein ve diğ. (2022) operasyonel üretim çizelgesini, taktiksel parça besleme politikası seçme karar sürecine dahil etmişlerdir. Bu anlayışla önerdikleri karma tam sayılı matematiksel programlama modelinin amacı, her bir parçayı hat yanı stoklama, hareketli kit oluşturma veya sıralama politikalarından birine atayarak ve ürün karışımı senaryosu başına ideal bir çizelge tipi seçerek ilgili hat besleme maliyetlerini en aza indirmektir. Maliyet modellerini ise Sali ve Sahin (2016) tarafından önerilen parça besleme politikalarına ve Limère ve diğ. (2015) tarafından tanımlanan yürüme mesafesine dayandırmışlardır.

Zangaro ve diğ. (2021) parçaları hat yanı stoklama, sabit kit oluşturma ve sıralama (tesis içerisinde veya tedarikçide) olmak üzere üç farklı besleme stratejilerinden birine atamak için sınıflandırma ve regresyon ağacı (ing: classification and regression tree) algoritmasını kullanmışlardır. Modeli eğitmek, doğrulamak ve test etmek için sentetik olarak oluşturdukları verileri kullanmışlardır. Sınıflandırma ağacının hat besleme stratejisini ortalama %78,49'luk bir sınıflandırma doğruluğu ile tahmin edebildiğini ifade etmişlerdir.

Caputo ve diğ. (2015b) personel maliyeti, yatırım maliyeti (konteynerler, depolama rafları ve nakliye araçları), süreç içi envanter maliyeti ve yer işgal etme maliyetinden oluşan toplam maliyeti minimize etme amacıyla parça besleme politikası seçmeyi sağlayan bir matematiksel programlama modeli geliştirmişlerdir. Model ile her bir parçanın hat yanı stoklama, kit oluşturma ve tam zamanında teslimat politikalarından en uygun olanına atanmasını sağlamayı hedeflemişlerdir. Sonra yaptıkları diğer bir çalışmada Caputo ve diğ. (2018) bu üç farklı besleme politikasını karşılaştırmak için yönetimsel ve kalite maliyetlerini de dahil ederek daha ayrıntılı bir maliyet modeli sunmak için senaryo analizini kullanmışlardır. Bu çalışmalarında sabit bir günlük üretim

hacmini, ürün karmasını ve hacmini varsayarken, üretim sırasının rolünü ihmal etmişlerdir.

2.2. Süpermarket Sayısı ve Yerleşimi

TZÜ ortamında fabrika içi lojistik faaliyetlerine ilişkin literatürde karşılaşılan araştırma alanlarının bir diğeri tedarik süreçlerinde kullanılan süpermarketlerin sayısının ve yerleşim yerlerinin optimum şekilde belirlenmesidir. Çeşitli parçaların iş istasyonlarına sürekli beslenmesini gerektiren TZÜ yaklaşımında üretim veya montaj istasyonlarına yakın, dağınık stok alanları olarak süpermarketler taşıma maliyetlerini azaltma, koordinasyon yeteneğinin artırma ve esneklik sağlama gibi avantajları nedeniyle oldukça yaygın şekilde kullanıldığı görülmektedir (Uysal ve Özcan, 2019).

Bu konuda yapılan çalışmalardan biri olarak Emde ve Boysen (2012b) özellikle karma model montaj hatlarına besleme yapmak üzere merkezi olmayan fabrika içi lojistik alanları olarak bilinen süpermarket konseptinin genel olarak artılarını ve eksilerini tartışarak bunlarının optimal sayısının ve yerleşim yerlerinin belirlenmesi problemini ele almışlardır. Süpermarketlerin kurulması ve bakımının operasyonel ve ekonomik yönlerini tartışarak süpermarket konum problemini tanımlamışlar ve problemi en iyi şekilde çözmek için bir dinamik programlama modeli önerisi sunmuşlardır. Önerdikleri algoritmanın, çözüm süresinin polinomsal olarak sınırlı olduğu ve kapsamlı hesaplama testlerindeki tüm problemlerin sadece birkaç saniye CPU zamanında çözülebildiğini ifade ederek, gerçekçi boyuttaki örnekleri çözmek için uygun olduğunu göstermişlerdir. Yaptıkları sayısal çalışma sonucunda bir montaj hattının az sayıda, örneğin üç veya dört süpermarketten bile tedarik etmenin, yalnızca bir merkezi depolama alanına sahip bir kurulumdan çok daha üstün olduğunu ve bundan sonraki ek süpermarketlerin verimliliğinin ve kullanışlılığının hızla azaldığını ifade etmişlerdir.

Nourmohammadi ve diğ. (2018a) süpermarketler kullanılırken montaj hattı tasarımını birbiriyle ilişkili iki karar problemi olarak hem hat dengeleme hem de parça besleme açısından ele almışlardır. Amaç fonksiyonunda istasyon sayısı ve süpermarket sayısı ile ilgili toplam kurulum maliyetlerinin minimize edilmesi olan karma tam sayılı bir matematiksel model önermişlerdir. Bu model ile optimum süpermarket sayısını ve her birinin desteklemesi gereken istasyon adedini elde etmeyi amaçlamışlardır.

Battini ve diğ. (2010) sipariş üzerine montaj çerçevesinde fabrika içindeki depoların sayısı (merkezi, merkezi olmayan veya süpermarketler), konumu ve parça stoklama metodolojisi konusunda çalışmışlardır. Bir süpermarket deposunu ne zaman, nasıl ve nereye kurmanın uygun olduğunu tanımlamak için yenilikçi ve adım adım bir prosedür sunmuşlardır. Bu prosedürde montaj sistemlerinin tipi, sayısı, konumu, talep oranı, kullanılan bileşenlerin ortak olma derecesi, fabrika içi nakliye ekipmanları, nakliye maliyetleri, stoklama için alan maliyeti, envanter maliyetleri ve güvenlik stoğu boyutu gibi bir dizi etmeni göz önüne almışlardır.

Fathi ve diğ. (2020) süpermarket yer seçimi problemi için Tavlama Benzetimi algoritması önererek süpermarket maliyetlerini minimize etmeyi amaçlamışlardır. Toplam maliyeti, montaj hatları boyunca ilk kurulum ve parça sevkiyat maliyetlerini göz önüne alarak elde etmişlerdir. Önerdikleri algoritmanın verimliliğini, bir dizi test problemi çözerek test etmişlerdir. Ayrıca çalışmalarında iş yükü dengelemesinin süpermarket maliyeti üzerindeki etkisini de incelemişlerdir. Sonuç olarak, iş yükü dengelemenin toplam süpermarket maliyetini önemli ölçüde azalttığını, genel çevresel ve ekonomik sürdürülebilirliğe katkıda bulunduğunu ortaya koymuşlardır.

Zhou ve Tan, (2020) kullanım oranları ve kapasite kısıtlarını eşzamanlı düşünerek süpermarket yer seçimi problemini ele almışlardır. Problem için işletme ve sevkiyat maliyetlerinin toplamından oluşan toplam maliyeti minimize etme amacıyla karma tam sayılı matematiksel bir model geliştirmişlerdir. Ele aldıkları problem NP-tam bir problem olduğu için gerçek dünya ölçeğindeki problemleri kesin bir yaklaşımla çözmek mümkün olmadığından metasezgisel bir algoritma olan diferansiyel gelişim stratejisine sahip, "dağıtım algoritmasının kendine uyarlamalı bir tahmini" (ing: a self-adaptive estimation of distribution algorithm) adlı algoritmayı önermişlerdir. Son olarak bir dizi deneme problemi ile önerdikleri algoritmanın geçerli ve verimli olduğunu göstermişlerdir.

Uysal ve Özcan (2019), Emde ve Boysen (2012b) tarafından tanımlanan montaj hatlarını beslemek üzere kurulacak süpermarketlerin sayısının ve yerlerinin optimum şekilde belirlenmesi problemi için komşuluk yapısı içeren bir tavlama benzetimi algoritması geliştirilmişlerdir. Test problemleri ile yaptıkları denemeler sonucu kısa işlemci zamanlarında başarılı sonuçlar elde ettiklerini ifade etmişlerdir.

Süpermarketlerin yerini ve sayısını optimum şekilde belirlemeye yönelik bir diğer çalışma da Alnahhal ve Noche (2015) tarafından yapılmıştır. Taşıma ve envanter maliyetlerinin minimize edilmesi hedeflenen problemin karma tam sayılı matematiksel bir programlama modeli oluşturulmuş ve kısa sürede etkili sonuçlar elde edebilmek için özel kromozom gösterimi, iki özel eşleştirme ve iki özel mutasyon operatörü içeren gerçek genetik algoritma önerilmiştir. Test problemleri sonucunda genetik algoritmanın makul CPU süresinde optimal olanlara çok yakın veya aynı sonuçları verdiği ifade edilmiştir.

Nourmohammadi ve diğ. (2018b) ise iş istasyonlarının taleplerindeki değişkenliği dikkate alan stokastik bir süpermarket yerleşimi problemi modeli geliştirmişlerdir. Model ile sevkiyat, envanter ve kurulum maliyetlerini içeren toplam parça besleme maliyetini minimize etmeyi hedeflemişlerdir. Gerçek ve üretilmiş test problemleri ile modellerin geçerliliğini ve uygulanabilirliğini göstermişlerdir.

2.3. Rotalama, Çizelgeleme ve Yükleme Problemleri

Fabrika içerisinde iş istasyonlarına malzeme tedariki konusunda yapılan çalışmaların oldukça önemli bir miktarının bu çalışmanın da konusu olan taşıyıcı araçların rotalanması, çizelgelenmesi ve yüklenmesi problemleri üzerine olduğu ifade edilebilir (Chen ve diğ., 2023; Briand ve diğ., 2018). TZÜ ortamında süreç içi envanterlerin azaltılabilmesi ve üretimin akıcı bir şekilde devam edebilmesi için iş istasyonlarına küçük miktarlarda ve sık sık parça tedariği yapılması kritik öneme sahip bir ihtiyaç olarak ortaya çıkmaktadır. Bu ihtiyacın karşılanmasında bir çözüm olarak çekici tren sistemleri ile döngüsel malzeme dağıtım yaklaşımının literatürde oldukça yaygın olarak kullanıldığı görülmektedir (Emde ve Schneider 2018; Zhou ve Zhu, 2020; Zenker ve diğ., 2016; Vaidyanathan ve diğ., 1999). Genellikle elektrik enerjisi ile çalışan ve manuel olarak kontrol edilen çekici bir lokomotif ve ona bağlı farklı veya aynı yapıda olabilen bir dizi vagondan oluşan çekici trenler yüksek kapasiteleri ve farklı tipteki parçaları bir arada taşıyabilmeleri sebebiyle esnek ve verimli bir fabrika içi tedarik sistemine olanak tanımaktadırlar (Battini ve diğ., 2015; Lu ve diğ., 2019).

TZÜ ortamında çekici trenlerle parça tedariği konusunda yapılan çalışmalardan biri olarak Chen ve diğ. (2023) tam zamanında üretim ortamında toplanan verilerin entegre edilmesiyle karma model montaj hatları için proaktif bir fabrika içi parça tedariği sistemi

geliştirmeye odaklanmışlardır. Üretim süreçlerinde yaşanan aksama ve acil tedarik gereksinimi durumlarında çekici trenlerin yeniden rotalanması gerektiğinde hem taşıma maliyetlerini hem de hat kenarı envanter maliyetlerini bir arada minimize etmeyi hedefleyen bir parça besleme sistemi tasarlamayı amaçlamışlardır. Taşıyıcı araçların yeniden rotalanması probleminin çözümüne yönelik yerel arama sezgiselinin yanı sıra çıkarma ve ekleme operatörlerini özelleştirerek adaptif bir geniş komşuluk araması algoritması geliştirmişlerdir. Önerdikleri proaktif parça besleme sisteminin etkililiğini göstermek için araç koltukları üreten bir fabrikada gerçek verilerle bir vaka çalışması sunmuşlardır.

Diğer bir çalışmada Zenker ve diğ. (2016) çekici tren rotalama problemini iki önemli varsayım altında incelemişlerdir. Bu varsayımlar; (1) her tren önceden belirlenmiş atandığı bir güzergâhta çevrimsel turlar yapar ve (2) tedarik sağlanan tüm müşteriler doğrusal ya da eğrisel bir hat üzerinde bulunurlar ve depo bu hattın başında ya da sonunda yer alır. Bu varsayımlar ile yazarlar tarafından rotalama probleminin ele alınmadığı ifade edilebilir, sadece önceden belirlenmiş ve üzerinde müşteri olarak kabul edilebilecek talebi bilinen ve sabit olan belirli istasyonların bulunduğu güzergâhlara çekici trenlerin kaç tanesinin atanacağı belirlenmektedir. Bu varsayımlar altında matematiksel olarak modelledikleri probleme dinamik programlama yaklaşımı ile çözüm önermişlerdir.

Zhou ve Zhu (2020) karma model montaj hatları için çekici trenlerin çizelgelenmesi ve yüklenmesi problemlerini incelemişlerdir. Bu problemlerin çözümüne ilişkin olarak bütün iş istasyonları için toplam hat kenarı stokların minimize edilmesini hedefleyen bir fabrika içi parça tedarik modeli formüle etmişlerdir. Önerdikleri karma tam sayılı matematiksel model için süpermarketlerin ve iş istasyonlarının yerlerinin önceden belirli olduğunu ve her bir çekici trenin rotalarının ve sürüş mesafelerinin bilindiğini varsaymışlardır. Modelde trenlerin kalkış zamanlarını ve her turda her bir iş istasyonuna teslim edilecek malzeme miktarını belirlemeyi hedeflemişlerdir. NP-zor sınıfına girdiğini ifade ettikleri problem için komşuluk arama ve tavlama benzetiminden oluşan güçlü bir hibrit sezgisel algoritma önermişlerdir.

Lu ve diğ. (2019) uçak montaj hattı için malzeme tedarik problemini ele almışlardır. Otomobil montaj hatlarına göre aynı anda birçok işin bir arada yapıldığını ifade ettikleri uçak montaj işlemleri için hat kenarı stoklarında zamana bağlı olarak çok çeşitli malzemelerin bulunduğunu belirtmişlerdir. Bu nedenle hat kenarındaki malzeme

stoklarının karmasının belirlenmesi ve çekici trenlerin çizelgelenmesinin önemini vurgulamışlar ve hat kenarına malzeme teslimat sayısını en aza indirmek amacıyla entegre bir karar verme matematiksel modeli oluşturmuşlardır. Her işin malzemelerinin teslimat partisi, teslimat süresi ve depolama konumları hakkında birlikte kararlar almak için hibrit bir endokrin-bağışıklık (ing: endocrine-immune) algoritması önermişlerdir. Gerçek hayat verileri ve rastgele oluşturulmuş örneklerle önerdikleri algoritmanın etkinliğini ve verimliliğini göstermişlerdir.

Emde ve diğ. (2012) çekici trenlerle parça besleme sürecini dört temel karar probleminde tanımlayarak birbirinden ayırmışlardır: (1) üretim ortamındaki süpermarketler için yerleşim planlaması, (2) çekici trenlerinin rotalanması, (3) her bir çeki tren için bir zaman çizelgesinin belirlenmesi, (4) her tur için trenlerin yükünün belirlenmesi. Bu problemlerden dördüncüsü olan çekici trenlerin yüklenmesi problemini, iş istasyonlarının malzemesiz kalmasını engellerken hat yanı stoklarının minimizasyonunu hedefleyerek ve kısıtlı tren kapasitelerini göz önünde bulundurarak araştırmışlardır. Belirli bir üretim çizelgesine bağlı olarak çekici trenlerin önceden belirli istasyonları belirli rotaları takip ederek malzeme dağıtımı yaptığı varsayımlarını göz önünde bulundurmuşlar ve sadece trenlerin yüklenmesi problemini ele almışlardır. Problemde hem hat yanı envanter toplamını hem de herhangi bir istasyondaki maksimum envanteri en aza indirmeyi hedefleyerek her turda hangi istasyon için hangi kutuların yükleneceğini belirlemeyi amaçlamışlardır. Problemin çözümüne yönelik olarak kesin bir polinom-zaman çözüm prosedürü sunmuşlardır.

Rao ve diğ. (2013) çok modelli montaj hattı için parça besleme problemini ele aldıkları çalışmada, tek aracın her turunda sadece tek bir istasyonu ziyaret ettiği bir durumu incelemişlerdir. Her bir tur için hangi iş istasyonunun ziyaret edileceği, teslim edilecek parça miktarı ve turun başlangıç zamanı çalışmanın karar verilmesi gereken problemleri olarak belirlenmiştir. Bu kararlar alınırken seyahat ve elde bulundurma maliyetlerinin minimize edilmesini hedeflemişlerdir. Problemin çözümü için yazarlar geriye dönük bir izleme yaklaşımı geliştirmişler ancak sadece belirli büyüklükteki problemleri bu yaklaşımla çözebildikleri için büyük problemlerin çözümüne yönelik olarak genetik algoritma ile tavlama benzetimi algoritmasının birleşiminden oluşan hibrit bir algoritma önermişlerdir.

Fathi ve diğ. (2014a) Volkswagen'nin Navarra montaj fabrikasındaki bir vakadan hareket ettikleri çalışmada çekici trenlerin çizelgelenmesi ve yüklenmesi problemlerini birlikte ele almışlardır. Problem için tasarladıkları karma tam sayılı matematiksel modelde tur sayılarını ve hat kenarındaki envanter seviyelerini minimize etmeyi amaçlamışlardır. Ayrıca problemin zor doğasından dolayı yenilikçi bir tavlama benzetimi temelli sezgisel algoritma önermişlerdir. Önerilen algoritmanın performansını değerlendirmek ve matematiksel modeli doğrulamak için, üretilen bir dizi test problemi ve iki gerçek hayat örneği çözümünü sunmuşlardır.

Fathi ve diğ. (2014b), Fathi ve diğ. (2014a) tarafından tanımlanan montaj hattı parça besleme problemi için memetik karınca kolonisi optimizasyonu tabanlı yeni bir sezgisel algoritma önermişlerdir. Hem matematiksel modeli hem de önerdikleri memetik algoritmayı test etmek için bir otomobil üreticisinden aldıkları vaka verilerini ve bir dizi oluşturulmuş test problemini çözmüşlerdir. Sonuç olarak memetik algoritmanın kısa sürede yeterince iyi çözümler bulduğunu ortaya koymuşlardır.

Fathi ve Ghobakhloo (2020) karma modelli montaj hatlarına malzeme tedariği konusunda yaptıkları çalışmada malzeme taşıma ve malzeme tutma maliyetlerinin azaltılması yoluyla genel üretim maliyetlerini azaltmayı hedeflemişlerdir. Çekici trenlerin sadece önceden belirli olan yollarda ilerlediği varsayılan problemde sadece yükleme ve çizelgeleme problemlerini ele almışlardır. Problemin çözümü için karma tam sayılı matematiksel bir model ve probleme özel iki aşamalı bir sezgisel algoritma önermişlerdir. Önerilen sezgisel algoritmanın etkinliğini ve verimliliğini değerlendirmek için, bir dizi test problemini çözerek optimum sonuçlar ile karşılaştırmışlardır.

Zhou ve Peng (2019) otomotiv montaj hattına parça besleme sürecinde taşıyıcı araçları birbiri ile ilişkisiz paralel makineler olarak düşünerek çizelgelemeyi hedeflemişlerdir. Problem tanımlarında her bir makinenin her seferinde bir tek malzeme kutusunu taşıyarak montaj hattını beslediğini varsaymışlardır. Bu kapsamda stoksuzluk durumunu engelleyen kısıtlar altında hat kenarı envanter miktarını minimize etmek amacıyla malzeme kutularını taşıyıcı makinelere atayan ve taşıma sıralamasını belirleyen bir çizelgeleme yapmışlardır. Küçük ölçekli örnekleri çözebilmek için geri izleme tabanlı kesin bir algoritma geliştirmişlerdir. Gerçek hayat ölçeğindeki problemler için ise öğretme-öğrenme metasezgiselini bir ışın arama (ing: beam search) tekniği ile entegre ederek hibrit bir metasezgisel önermişlerdir. Sundukları deneysel sonuçlar, çizelgeleme

algoritmalarının önerilen ilişkisiz paralel makine çizelgelemeyi çözmede etkili ve verimli olduğunu göstermektedir.

Malzeme ile dolu kutuların montaj hatlarına tedarik edilmesi sürecini iş, taşımayı yapacak araçları ise paralel makineler olarak yorumlayan ve bu bakış açısıyla parça besleme sürecini çizelgelemeyi hedefleyen bir diğer çalışma ise Peng ve Zhou (2018) tarafından yapılmıştır. Yazarlar çoklu server çizelgeleme problemi olarak tanımladıkları problemde hat kenarı envanterlerin azaltılması amacıyla taşınması gereken her bir malzeme kutusunun server olarak isimlendirdikleri taşıyıcı araçlara atanmalarını ve taşıma sıralarının belirlenmesini elde etmeyi amaçlamışlardır. Tanımladıkları problemin optimum çözümünü elde edebilmek amacıyla karma tam sayılı matematiksel bir model önermişlerdir. Ayrıca problemin yüksek karmaşıklığını göz önünde bulundurarak, temel karınca kolonisi optimizasyonu ile hızlı bir yerel arama ve uyarlanmış genişlik öncelikli ağaç arama (ing: adaptive breadth-first tree search) yöntemini içeren yerel arama algoritması ile entegre ederek hibrit bir karınca kolonisi algoritması önermişlerdir.

Briand ve diğ. (2018) karma model montaj hatlarına parça besleme problemini, taşıyıcı araçların (çekici trenler) enerji tüketiminin minimizasyonunu hedefleyerek ele almışlardır. Problemde, süpermarketlerden iş istasyonlarına parçaların kutular içerisinde tedarik edildiğini ve her bir zaman periyodu için, üretim planına bağlı olarak, bir iş istasyonunun kutu cinsinden ifade edilen belirli bir parça talebi olduğunu belirtmişlerdir. Çizelgeleme, rotalama ve yükleme problemlerini bir arada ele aldıkları çalışmada, problemi; herhangi bir malzeme eksikliğini önlemek için, çekici tren tarafından her bir zaman periyodunda ziyaret edilecek iş istasyonlarının sırasını ve her durakta teslim edilecek kutu sayısını belirlemek olarak ifade etmişlerdir. Problem için iki orijinal karma tam sayılı doğrusal programlama formülasyonu önermişler ve enerji verimli tedariğin ekonomik verimlilikle çelişmediği koşulları gösteren hesaplamalı analizler ve deneyler sunmuşlardır.

Faccio ve diğ. (2013) bir İtalyan otomotiv üreticisindeki gerçek bir vaka çalışmasına dayanan bir montaj hattı besleme sisteminin simülasyon modelini sunmuşlardır. Kanban kontrollü bir üretim sistemi ve stokastik talebi göz önünde bulundurarak çekici trenlerinin çizelgelenmesi ve yüklenmesi problemleriyle baş etmeye çalışmışlardır. Çekici tren sayısı, hizmet seviyesi, tur sıklığı ve römorkör kapasitesi olmak üzere dört kritik parametreyi dikkate alarak farklı senaryoları analiz etmişlerdir. Altı üretim faktörü

bazında her bir senaryoya ilişkin çözümlerin performansını incelemiş ve her bir parametrenin çözüm üzerindeki etkisini araştırmışlardır.

Saez-Mas ve diğ. (2020) bir endüstriyel montaj tesisinin yeniden tasarlanması amacıyla yaptıkları çalışmada farklı iş istasyonlarını beslemek için bileşenlerin ve parçaların akışını optimize eden matematiksel bir formülasyon önermişlerdir. Geliştirdikleri matematiksel model ile tesis için elde ettikleri sonuçlar, tedarikçilerin alım rampalarına ve parçaların depolama alanlarına daha iyi atanmasıyla ve malzeme taşıma ekipmanlarının kapasitelerinin uygun şekilde seçilmesiyle toplam dağıtım süresinde neredeyse %33'lük bir tasarrufun sağlanabildiğini göstermiştir.

Kilic ve diğ. (2012) gerçek hayatta gözlemledikleri özellikle çekme sistemi ile çalışan ve yalın üretimi benimsemiş üretim tesislerinden ve literatürde var olan kısıtlı sayıdaki çalışmalardan yola çıkarak, fabrika içi döngüsel sefer rotalama problem türlerini sınıflandırmışlar ve açıklamışlardır. Açıkladıkları ana sınıflardan biri için model geliştirerek çalışmalarında sunmuşlardır. Ortaya koydukları matematiksel programlama modelinde döngüsel sefer rotalama problemi için gerekli araç sayısını ve toplam mesafenin en küçüklenmesini hedeflemişlerdir. Son olarak, gerçek uygulamadan esinlendikleri sayısal bir örnek ile geliştirdikleri modelin uygulanabilirliğini göstermişlerdir.

Sipahioğlu ve Altın (2019) otomatik yönlendirilmiş araçlar için fabrika içi döngüsel sefer rotaları oluşturmak ve bu rotaların çevrim sürelerini belirlemek amacıyla karma tam sayılı bir matematiksel model önermişlerdir. Geliştirdikleri model, montaj istasyonları için bölünmüş teslimatlara ve römork eklenerek araç kapasitelerinin artırılabilmesine izin vermektedir. GAMS yazılımını kullanarak kodladıkları modeli rassal olarak ürettikleri test problemleri ile test etmişlerdir. Önerdikleri modelin test problemleri için ortalama bir saniye civarında optimum sonucu elde ettiğini ifade etmişlerdir.

Akıllıoğlu ve diğ. (2006) dizel motorlar için enjektör üretimi yapan bir fabrikada malzemelerin fabrika içinde tekrarlı olarak döngüsel seferle dağıtılması için bir sistem tasarlamışlardır. Çalışmalarında öncelikle fabrikanın fiziksel koşullarına göre taşıyıcı araçların gidebileceği alternatif rotaları ve stok noktalarını belirlemişlerdir. Daha sonra stok noktalarını taşıyıcı araçlara atayan bir matematiksel model geliştirmişlerdir. Önerdikleri modelle malzeme stok maliyetini ve taşımadan kaynaklı maliyetleri en aza

indirmeyi hedeflemişlerdir. Önerdikleri malzeme dağıtım sistemi ile toplam taşıma ve malzeme stok maliyetlerini %66 oranında azaltabildiklerini ifade etmişlerdir.

Kılıç ve Durmuşoğlu (2013), çekme sistemli pürüzsüz ve tekrarlı olan yalın üretim sistemlerinde malzeme tedariki için döngüsel sefer tren sistemlerinin yaygın ve etkili şekilde kullanıldığını vurgulamışlardır. Bu doğrultuda yalın üretim sistemleri için önceden belirlenmiş rotalarda eşit çevrim zamanlarında periyodik olarak malzeme tedariki sağlayan döngüsel sefer trenleri için spesifik bir model geliştirmeyi amaçlamışlardır. İşlem yapılan mamul - ara mamul ve taşıma maliyetlerinin en küçüklenmesini amaçlayan karma tam sayılı doğrusal programlama modeli önermişlerdir. Geliştirdikleri model, değişken sayısı arttıkça problemin kombinatoryal yapısından dolayı optimum sonucu bulmakta zorlandığı için, probleme özgü yenilikçi bir sezgisel yaklaşım geliştirmişlerdir. Sayısal bir örnekle önerdikleri model ve sezgisel yaklaşımın uygulanabilirliğini göstermişlerdir.

Satoğlu ve Sipahioğlu (2018) fabrika içi döngüsel envanter rotalama problemi için iki aşamalı bir yaklaşım önermişlerdir. Bu yaklaşımın ilk aşamasında istasyonların rotalara atandığı ve rotaların servis periyodlarının belirlendiği karma tam sayılı matematiksel bir model çözülmektedir. İkinci aşamasında ise her bir rotada sıralamanın elde edilebilmesi için bir gezgin satıcı problemi çözülmektedir. Ek olarak çalışmalarında, istasyon sıralamalarının ve servis periyotlarının her bir rota için eşzamanlı olarak belirlendiği alternatif bir karma tam sayılı matematiksel model geliştirmişlerdir. Her iki atama temelli modelin de envanter rotalama problemi için çözüm süresini önemli ölçüde azalttığını ifade etmişlerdir. Orta boyutlu bir hipotetik veri setini her iki yaklaşımla da çözmüşler ve sonuç olarak önerdikleri iki aşamalı yaklaşımın alternatif atama temelli modelden çözüm süresi açısından daha iyi olduğunu ortaya koymuşlardır.

Mao ve diğ. (2020) Toyota fabrikasında kullanılan yeni bir depolama politikası olan ve maliyet tasarrufu üzerinde önemli bir etkiye sahip olan P-LANE modelini dikkate alarak otomobil parçaları toplama problemini incelemişlerdir. Bu yeni model için, üretimden ve iç lojistikten kaynaklanan toplam maliyeti en aza indirmek amacıyla otomobil parçalarının tedarikçilerden toplanma rotalarını ve toplanan her bir parçanın atanması gereken P-LANE hattını aynı anda belirleyebilen karma tam sayılı doğrusal programlama modeli önermişlerdir. Elde ettikleri bulgular, sıfır envanter modeliyle karşılaştırıldığında

döngüsel sefer sistemi içinde P-LANE kullanımının toplam maliyetleri önemli ölçüde azaltabildiğini ve aynı zamanda nakliye verimliliğini artırabildiğini göstermektedir.

Buyukozkan ve diğ. (2019) "çok-malzemeli çok-araçlı malzeme tedarik sistemi tasarım problemi" olarak tanımladıkları fabrika içi döngüsel sefer dağıtım problemini ele alarak çözüm geliştirmişlerdir. İlk olarak, fabrika içinde doğrudan dağıtılması gereken parçalar ile döngüsel sefer turları ile çevrimsel bir şekilde teslim edilebilecek olanlardan ayırt edilmesi gerektiğini ifade etmişlerdir. Döngüsel şekilde turlar yapacak araçların rotalarının belirlenmesi için öncelikle "tek araçlı döngüsel sefer matematiksel modeli" önermişlerdir. Daha sonra, çok araçlı fabrika içi bir döngüsel sefer sistemi tasarlamak için önerilen modeli yinelemeli olarak kullanan bir mat-sezgisel algoritma geliştirmişlerdir. Önerilen metodolojiyi gerçek bir çamaşır makinesi montaj fabrikasının döngüsel sefer sistemini tasarlayarak doğrulamışlardır.

Emde ve Gendreau (2017) tam zamanında üretim sisteminin hedeflerini göz önünde bulundurarak karışık modelli montaj hatlarına parça beslemek için genellikle çekici trenler olarak adlandırılan tesis içi taşıma araçlarının çizelgelenmesi problemini incelemişlerdir. İstasyonlarda duruş süreleri ve istasyon kapasiteleri olmayan özel durum polinom zamanında çözülebilse de, problemin güçlü anlamda NP-tam olduğunu göstermişlerdir. Problem için karma tam sayılı bir model ve dekompozisyon temelli bir sezgisel yöntem geliştirerek kendi oluşturdukları test problemleri ile sınamışlardır. Sonuç olarak, geliştirdikleri sezgisel yöntemin genel durum için optimuma çok yakın çözümler ürettiğini göstermişlerdir.

Zhou and Peng (2017) çalışmalarında tam zamanında üretim ortamında merkezi olmayan depodan montaj istasyonlarına parça tedariği problemini ele almışlardır. Problem için hat kenarındaki envanter seviyelerini en aza indirmeyi hedefleyen kombine bir rotalama ve yükleme dağıtım modeli formüle etmişlerdir. Küçük boyutlu problemleri çözebilmek için baskınlık özelliklerine dayalı olarak kesin çözüm üreten bir geri izleme (ing: back tracking) yaklaşımı önermişlerdir. Problemin NP-zor doğasını göz önünde bulundurarak, bir diferansiyel gelişim döngüsü ve genişlik öncelikli arama içeren verimli bir ABC tabanlı meta-sezgisel geliştirmişlerdir. Kesin çözüm üreten yöntemin, çalışma süresi sınırı içinde 25'ten az ürün modeline sahip örnekler için en uygun programları elde etme yeteneğine sahip olduğunu, değiştirilmiş ABC yaklaşımın da 100 ürün modeline kadar gerçek hayat ölçeğindeki örnekler için tatmin edici çözümler sunduğunu ifade etmişlerdir.

Emde ve Boysen (2012a) karma model montaj hatlarının istasyonlarına süpermarket olarak isimlendirilen merkezi olmayan şirket içi lojistik alanlarından parça tedarik sağlayan otomobil üreticileri için çekici tren rotalama ve çizelgeleme problemini incelemişlerdir. Problem kapsamında çekici trenlerin her birinin kendine tahsis edilecek ardışık montaj hattı parça besleme noktalarına karar verilmesini ve trenlerin turlarının başlangıç zamanlarına karar verilmesini hedeflemişlerdir. Her bir çekici tren kendi bölgesinde hareket edeceği için trafik oluşmayacağını ifade etmişlerdir. Polinom çalışma zamanında bu sorunu optimum şekilde çözmek için iç içe geçmiş bir dinamik programlama prosedürü sunmuşlar ve kapsamlı bir hesaplama yaparak çekici trenlerin sayısı ve kapasiteleri ile istasyonlarda tutulan envanter arasındaki ödünleşimi araştırmışlardır.

Kluska ve Pawlewski (2018) döngüsel sefer sistem tasarımı ve uygulama metodolojisinin temel yedi adımdan oluştuğunu ve bu adımlardan bir tanesinin de simülasyon tekniğinin kullanımı olduğunu ifade etmişlerdir. Döngüsel sefer sisteminin tasarımı ve uygulamasındaki ilk adımın ana üretim sürecini analiz etmek olduğunu, ikinci adımın tedarik yollarının üretim alanı düzeninde tasarlanması olduğunu, üçüncü adımının bilgi akışı ve yönetimi ile ilgili olduğunu, dördüncü adımın lojistik araçları için zaman çizelgesini planlamak olduğunu, beşinci adımın hibrit modelleme yaklaşımı kullanılarak doğrulamayı içerdiğini, altıncı adımın sistemin fabrikada uygulanması ve yedinci adımın ise sistemin bakımını ve iyileştirilmesini içerdiğini ifade etmişlerdir. Tanımladıkları metodolojinin uygulanmasına bir örnek olarak kesikli olay ve etmen tabanlı modellemeyi içeren hibrit bir simülasyon modelini FlexSim simülasyon yazılımı kullanılarak oluşturmuşlardır.

Gyulai ve diğ. (2013) büyük ölçekli imalathanelerde üretim ve/veya montaj istasyonlarının malzeme gereksinimlerini karşılamak için farklı taşıma sistemleri uygulandığını söyleyerek, taşıyıcı araçların sınırlı kapasitesi ve lojistik süreçlerin zaman tüketimi nedeniyle hatasız üretimi desteklemek için etkili araç rotalama yaklaşımlarının gerekliliğini ifade etmişlerdir. Bu doğrultuda fabrika içi döngüsel sefer rotalama probleminin çözümü için başlangıç çözüm üretme sezgiseli ve yerel arama metodundan oluşan yeni bir çözüm yaklaşımı önermişlerdir. Önerdikleri yaklaşımın yeteneklerini göstermek için bir yazılım prototipi geliştirmişler ve bir gerçek hayat problemi üzerinde test etmişlerdir.

Peng ve diğ. (2020) özellikle otomotiv sektöründe karma model montaj hatlarına parça tedariki için çekici tren çizelgeleme problemini ele almışlardır. Bu probleme ilişkin geliştirdikleri çözümle, istasyonların envanterine ve çekici trenlerin kapasitesine bağlı olarak, çekici trenlerin sayısına, her turun programına ve tur başına yüklenen parça tipine ilişkin kararları vermeyi hedeflemişledir. Bu amaçla araç kapasiteleri ve montaj hattının stok seviyeleri göz önünde bulundurularak en az sayıda çekici treninin kullanılması temel amacı ile bir karma tam sayılı doğrusal programlama modeli oluşturmuşlardır. Bu modelle asgari çekici tren sayısı temelinde, her bir turun programını, parça tedarik probleminin ikincil amacı olan montaj hattı envanterini en aza indirecek makul bir şekilde planlamayı amaçlamışlardır. Ek olarak modern otomobil üretiminin sürekliliğini ve karmaşıklığını dikkate alarak, büyük ölçekli problemlerin çözümü için kısa sürede tatmin edici sonuçlar elde edebilen sezgisel bir algoritma tasarlamışlardır. Ayrıca, bazı vakaları analiz ederek yaygın olarak kullanılan periyodik teslimat stratejisi ile tam zamanında teslimat stratejisini karşılaştırmışlar ve tam zamanında teslimat stratejisinin, envanter seviyesini azaltmada periyodik teslimat stratejisine göre daha fazla avantaja sahip olduğunu ortaya koymuşlardır

Emde ve Schneider (2018) tam zamanında üretim yapan montaj fabrikalarındaki veya atölyelerdeki istasyonlara parça beslemesi yapan fabrika içi taşıma araçlarının rotalanması problemini ele almışlardır. Bu problemde çekici tren olarak tanımlanan vagonlu ve belirli bir kapasiteye sahip taşıyıcı araçların ara vermeksizin döngüsel olarak istasyonlara parça tedariği sağlaması söz konusudur. İlgili problem için istasyonlarda verilen minimum servis frekanslarına bağlı kalarak, öncelikle araç sayısını ve daha sonra toplam rota süresini en aza indirmeyi hedeflemişlerdir. Bu amaçla karma tam sayılı bir matematiksel model geliştirmişlerdir. Ele aldıkları problem araç rotalama probleminin bir benzeri olarak NP-zor olduğu için gerçek hayat problemlerine çözüm üretebilmek adına yeni bir geniş komşuluk arama algoritması önermişlerdir. Gerçek hayat problemleri ile yaptıkları testler sonucunda geliştirdikleri algoritmanın oldukça iyi sonuçlar verdiğini göstermişlerdir.

Alnahhal ve Noche (2013) merkezi olmayan süpermarketlerden çekici tren sistemleri ile karma model montaj hatlarındaki istasyonlara malzeme sevkiyatını incelemişlerdir. Trenlerin rotalanması, çizelgelenmesi ve yüklenmesi problemlerini tren sayısını, yükleme ve rota uzunluklarındaki değişkenliği ve hat tarafındaki envanter tutma maliyetlerini en

aza indirgemek için birlikte ele almışlardır. Bu problemleri paralel olarak çözebilmek için analitik denklemler, Dinamik Programlama ve Karma Tamsayı Programlama yöntem ve tekniklerini kullanmışlardır. Kısa CPU zamanında elde ettikleri sonuçlar ile rotalar arasında zaman tamponu kullanımının olurlu bölge ve optimal çözüm üzerindeki etkisini göstermişlerdir. Ayrıca araç yüklemedeki değişkenliği azaltma hedefinin rotalama, çizelgeleme ve yükleme sonuçları üzerindeki etkisini de göstermişlerdir.

Satoglu ve Sahin (2013) merkezi depodan montaj hatlarına fabrika içi döngüsel malzeme tedarik sistemi tasarlamak için taşıyıcı araçların rotalarının ve döngüsel sefer çevrim süresinin belirlenmesini içeren matematiksel bir model ve sezgisel yaklaşım önermişlerdir. Önerdikleri matematiksel modelin amacı, toplam malzeme taşıma ve stok tutma maliyetlerinin minimizasyonudur. Geliştirdikleri sezgisel algoritma bir başlangıç periyot değerine dayalı olarak rotaları inşa etmeyi ve farklı periyot değerlerini dikkate alarak çözümü iyileştirmeyi amaçlamaktadır. Ayrıca sezgisel algoritma bütün rotaların eşit servis periyotlarına sahip olduğunu varsaymaktadır. En uygun çözüme, en az toplam malzeme taşıma ve süreç içi envanter/stok tutma maliyeti temelinde karar verilmektedir. Önerilen sezgisel yöntem bir TV montaj fabrikasında uygulanarak, gerçek uygulamalardaki faydası gösterilmiştir.

Fathi ve diğ. (2016) özellikle otomotiv sektöründe yaygın olarak kullanılan karma model montaj hatlarına merkezi olmayan depolardan parça tedariği için çok bileşenli çok araçlı döngüsel sefer tasarımı için çok amaçlı karma tam sayılı doğrusal programlama modeli ve modifiye edilmiş bir parçacık sürü optimizasyonu algoritması önermişlerdir. Problemlerinde çekici trenlerin rotalarının ve turların önceden belirlenmiş olduğunu varsaymışlar ve bu nedenle sadece çizelgeleme ve yükleme alt problemlerini ele almışlardır. Önerdikleri meta-sezgisel algoritmanın verimliliğini ve etkililiğini rastgele oluşturulmuş 48 deneme problemini ve Volkswagen -Navarra tarafından sağlanan gerçek bir problemini çözerek göstermişlerdir.

Golz ve diğ. (2012) merkezi bir depolama alanından karma modelli bir montaj hattının çeşitli istasyonlarına fabrika içi parça tedariki problemini ele almışlardır. Problemlerinde önceden belli olan montaj sıralamasına göre oldukça değişkenlik gösteren farklı özelliklerdeki otomobillerin montajı için gerekli olan bileşenleri tedarik edecek çekici trenlerin sürücülerin sayısının minimizasyonunu hedeflemişlerdir. Bu amaçla problemin çözümü için iki aşamalı sezgisel bir yaklaşım önermişlerdir. Yöntemlerinin ilk

aşamasında teslim edilecek siparişler, montaj sırasından ve malzeme listesinden türetilmektedir. İkinci aşamada, bu siparişler taşıyıcı araçların kapasite kısıtları, son teslim zamanları ve tur çizelgeleme kısıtları dikkate alınarak çekici tren sistemindeki turlara atanmaktadır. Çalışmada çekici trenlerin gidebileceği güzergahların önceden belirli olduğu varsayılmaktadır. Sundukları sayısal sonuçlar ile önerilen sezgisel yöntemin büyük boyutlu problem örneklerini bile kısa hesaplama süresinde çözdüğünü göstermişlerdir.

Vaidyanathan ve diğ. (1999) tam zamanında üretim sistemi altında çalışan üretim tesislerinde malzeme tedariki için araç rotalarının planlanması problemine ve çözümüne odaklanmışlardır. Problemde her taşıyıcı aracın, rota süresine göre belirlenen sabit talep oranlarına sahip istasyonlara üretim süreçlerinde ciddi bir değişiklik olmadığı sürece planlama ufku boyunca aynı çevrimsel rotalarda sürekli tedarik sağlayacağını varsaymışlardır. Tam Zamanında Kapasite Kısıtlı Araç Rotalama Problemi (TZKKAP) olarak isimlendirdikleri problem için klasik araç rotalama probleminin modifiye edilmiş hali olan doğrusal olmayan bir karma tam sayılı programlama modeli sunmuşlardır. Bunu, problemin çizelgeleme tarafını göz ardı ederek gerçekleştirmişlerdir. Geliştirdikleri modelde toplam yoldan ve malzeme alıp bırakmadan kaynaklanan toplam seyahat süresinin minimizasyonunu hedeflemişlerdir. Ayrıca, alt tur eleme ve talep karşılama kısıtlarını gevşeterek araç sayısı için bir alt sınır elde etmek için doğrusal bir model sunmuşlardır. Ek olarak, TZKKAP'yı çözmek için iki aşamalı bir sezgisel algoritma önermişlerdir. İlk aşamada, modifiye edilmiş bir en yakın komşu şeması yoluyla olurlu rotalar üretmişler; ikinci aşamada ise, bu rotaları 3-opt yerel arama aracılığıyla iyileştirmişlerdir. Sezgisel yöntemi, endüstri verilerinden türetilen 270 örneklik bir problem seti üzerinde test etmişlerdir, bu testlerde problem boyutunun 10 ila 71 istasyon arasında değişiklik gösterdiğini ifade etmişlerdir.

Fathi ve diğ. (2018) montaj hatlarına hammadde ve parça tedariğinin çizelgelenmesi konusunu ele almışlardır. Kullanılan araç sayısını en aza indirmeyi hedefledikleri bu problem için karma tam sayılı doğrusal bir programlama modeli önermişlerdir. Malzeme ve hammaddelerin taşıyıcı araçlar tarafından çevrim olarak isimlendirdikleri başlama zamanı önceden belirli, sabit ve düzenli aralıklarla yapıldığını varsaymışlardır. Ayrıca çalışmalarında taşıyıcı araçlar için rotaların önceden belli olduğunu varsayarak rotalama

yapmamış sadece çizelgeleme yapmışlardır. Önerdikleri modelin geçerliliğini birkaç farklı gerçek hayat problemini çözerek ve farklı senaryoları analiz ederek incelemişlerdir.

Zhou ve Shen (2018) karma model montaj hatlarında malzeme tedarik görevlerini yerine getirmek için enerji açısından verimli bir çizelgeleme yöntemi geliştirmeyi amaçlamışlardır. Enerji verimliliğini temel hedef olarak gördükleri çalışmalarında verimlilikle ilgili bir hedef olan toplam ağırlıklı gecikmeyi de minimize etmeyi amaçlayan çok amaçlı bir problem tanımlamışlardır. Problemlerinde malzeme taşıma aracı olarak bir grup otomatik yönlendirmeli araçtan oluşan otomatik bir malzeme taşıma sistemi olduğunu ve hangilerinin gerçekleşeceğine karar verecekleri birkaç alternatif otomatik yönlendirilmiş araç turu bulunduğunu ifade etmişlerdir. Problemin NP-zor olmasından dolayı aynı anda gecikmeyi ve toplam enerji tüketimini optimize etmeyi amaçlayan Tabu ile güçlendirilmiş Parçacık Sürü Optimizasyonu isimli bir algoritma geliştirmişlerdir. Önerdikleri algoritmanın etkinliğini göstermek için deneme problemleri tasarlayarak çözmüşlerdir.

Hwang (2004) üretim tesisi tasarımında kullanılmak üzere bir fabrika içi taşıyıcı rotalama problemi analiz etmiştir. Sabit bir tesis düzeninde önceden belirlenmiş malzeme akış yollarının göz önüne alındığı çalışmada üretim tesisi içinde minimum taşıma çabasıyla malzeme aktarmak için gereken minimum taşıyıcı sayısını belirlemeyi hedeflemiştir. Bu amaçla tam sayılı bir matematiksel model ve ayrıca sezgisel bir çözüm yaklaşımı geliştirmiştir. Oluşturduğu kapsamlı test problemleri ile algoritmalarını test etmiştir.

Staab ve diğ. (2016) fabrika içi döngüsel sefer sisteminin performansının iyileştirilmesini amaçlayan endüstriyel bir vaka çalışması sunmuşlardır. Çalışmalarında bir fabrika içi yerleşim planı içinde çok sayıdaki çekici trenin birbirlerini engellemeleri veya trafik oluşturma durumlarını inceleyerek, daha stabil ve dakik bir teslimat sağlayabilmek için simülasyon yaklaşımını kullanmışlardır. Çekici trenlerin sabit rotalarda sabit bir çizelgede ve aynı zaman periyotlarında hareket ettiğini varsaymışlar ancak bütün trenlerin aynı anda rotalarına başlamak yerine farklı zamanlarda başlayabileceğini ifade etmişlerdir. Simülasyon çalışmasının sonuçlarını ayrıntılı olarak sunmuşlar ve daha pürüzsüz dakik bir malzeme tedarik süreci için önerilerini geliştirmişlerdir.

Alizon ve diğ. (2009) parçaların depo alanlarına atanmasını optimize eden “depolarda malzeme ataması” olarak isimlendirdikleri bir metot sunarak depodan iş istasyonlarına

malzeme akışını optimize etmeyi amaçlamışlardır. Önerdikleri metot; maliyet değerlendirmelerinin birbirini izleyen çözümü, en kısa yol problemi ve genelleştirilmiş atama problemine dayanmaktadır ve şu üç aşamadan oluşmaktadır; (1) her parça için bir tedarik maliyetinin değerlendirilmesi, (2) bileşenlerin depolara en uygun şekilde atanması ve (3) taşıyıcı araç filosunun boyutlandırılması. Yöntemde bir parçanın tedarik maliyeti, tedarik döngüsünün seyahat süresine bağlıdır. Bu döngü, seyahat süresi ve koridorlardaki elleçleme yapan diğer araçları bekleme nedeniyle ortaya çıkan bekleme süresi ile tanımlanmıştır. Bu şekilde malzeme tedariki sırasında yükleme/boşaltma aşamalarında ortaya çıkan koridor ağındaki tıkanıklıklar hesaba katılmaktadır. Yazarlar tarafından her bir koridor için ortalama trafik kaynaklı bekleme süresinin yaklaşık bir değeri önerilmiştir. Önerilen yöntemi doğrulamak için Renault'nun Flins fabrikasına ilişkin bir örnek olay incelemesi detaylandırılmıştır.

Bocewicz ve diğ. (2019) fabrika içi döngüsel sefer rotalarının çakışmasız bir şekilde planlanması konusunda çalışmışlardır. Her bir çekici trenin takip edeceği istasyon sıralamalarını içeren rotalama ve çekici tren filosunun birbiri ile çakışmasız olarak ilerlemesine imkân verecek çizelgenin birlikte elde edilmesini amaçlamışlardır. Problemin kökünde toplamalı dağıtmalı ve zaman pencereli araç rotalama problemine dayandığını ve bu nedenle NP-zor yapıda olduğunu ifade etmişlerdir. Bu bağlamda, dağıtım ağı topolojisi kısıtlamalarının yanı sıra çeşitli müşteri talepleriyle çok dönemli, çok gezici ve çoklu parça akışlarının daha fazla entegrasyonuna izin veren bir kısıt programlama modeli sunmuşlardır. Ek olarak bir kısıt tatmin probleminin (ing: constraint satisfaction problem) özyinelemeli bir formülasyonunu önermişlerdir. Sağlanan bilgisayar deneyleri ile gerçek yaşam ölçeğindeki sistemlerde sunulan yaklaşımı kullanma olasılığını göstermişlerdir. Yazarlar çözüm yaklaşımlarında, toplama ya da dağıtma yapılacak iki istasyon arasındaki yolu en kısa yol algoritması ile elde etmişlerdir. Ancak gerçek hayatta çekici trenlerin u dönüşü yapamamasından dolayı araçların bir istasyona geliş yönünden farklı bir yöne gitmesi gerektiği gerçeğini dikkate almamışlardır. Ayrıca deneme problemlerinde en fazla 15 istasyon ve 4 çekici trene kadar denemeler gerçekleştirmişler ve 15 istasyon - 3 tren ile 15 istasyon - 4 tren kombinasyonlarında çözüme ulaşamamışlardır.

Duman (2016) fabrika içinde iş istasyonlarına döngüsel seferlerle malzeme tedariki için çekici trenlerin rotalanması ve çizelgelenmesi problemlerini çalışmıştır. Ele alınan

problemde her bir çekici trenin kroki üzerindeki bir koridordan en fazla bir kere geçerek bir Hamilton turu yapıp başlangıç noktasına geri döneceği varsayılmıştır. Koridorların iki tren yan yana geçemeyecek şekilde dar olduğunu ve bu nedenle arka arkaya çakışmaların engellenmesi gerekliliği vurgulanmıştır. Ayrıca koridorların tek yönlü olduğu ve bu yönlerin önceden belirlendiği ifade edilmiştir. Bu varsayımlar altında çekici lokomotiflerin ve lokomotiflere bağlanılacak kısıtlı kapasiteye sahip vagonların toplam maliyetini en küçükleyecek şekilde bir karma tam sayılı matematiksel model geliştirilmiştir. Problem boyutu büyüdükçe çözüm süresi çok fazla uzadığı için ayrıca yerel aramaya dayalı bir sezgisel algoritma önerilmiştir. Son olarak rasgele oluşturulan deneme problemleri ile önerilen matematiksel model ve sezgisel algoritma test edilmiştir. Çalışmada deneme problemleri sırasıyla 4, 7 ve 13 adet koridordan oluşan oldukça küçük üç adet kroki üzerinde oluşturulmuştur.

Duman (2016)'da ifade edilen her bir çekici trenin bir koridordan sadece bir kere geçmesi, uzun ve nispeten az sayıda koridorun bulunduğu montaj hatlarının aksine hücresel imalat ortamında malzeme dağıtımı için gerçekçi bir varsayım değildir. Bunun nedeni, hücresel imalat sisteminde çok fazla koridor olduğu için sadece Hamilton turları ile malzeme dağıtımının mümkün olmaması ya da çok verimsiz olmasıdır. Ayrıca çekici trenlerin her koridordan önceden belirlenmiş tek bir yönde geçiş yapabilmesi her ne kadar karşılıklı çakışma ihtimalini ortadan kaldırsa da çözüm uzayını daraltmakta ve daha iyi çözümlerin elde edilmesini engellemektedir.

Marialuisa ve diğ. (2023) bilişim teknolojisi tabanlı kanban ile yönetilen döngüsel sefer sisteminin dinamik optimizasyonu üzerinde çalışmışlardır. Fabrika içinde süpermarketlerden iş istasyonlarına giden en iyi yolu bulmak için karınca kolonisi optimizasyonundan yararlanmışlardır. Daha sonra elde edilen yol üzerinden parçaların teslimatı için optimum zamanları bulmak için ise genetik algoritmayı kullanmışlardır. Böylelikle, parça besleme için gerçekleştirilen her bir döngüsel sefer için rotaların dinamik olarak elde edilmesini sağlamışlardır. Sonuç olarak önerdikleri sistemle genel sistem performansından ve üretim parametrelerinde önemli iyileştirmeler elde ettiklerini vurgulamışlardır. Teslimat süresi ve malzeme taşıma süresinin sırasıyla %39 ve %48 oranında düşüş gösterdiğini ve süreç içi envanterin %22 oranında azaldığını ifade

etmişlerdir. Ancak araçların rotalanması konusunda herhangi bir şekilde çakışma ya da trafik oluşma durumuna değinmemişlerdir.

Zhou ve Zhao (2022) TZÜ ortamında karma modelli montaj hatları için iki seviyeli lojistik ağına dayalı, merkezi depo ile montaj hattına entegre süpermarketler arasındaki sürekli malzeme besleme sürecine odaklanan döngüsel sefer çizelgeleme problemi üzerinde çalışmışlardır. Problemlerinde aynı anda hem kullanılan elektrikli araç sayısını hem de elektrikli araçların maksimum seyahat mesafesini en aza indirmeyi hedeflemişlerdir. Bu amaçla çift yüzlü ayna sınır teorisine dayanan çok amaçlı ayrıştırma evrimsel algoritması (ing: multi-objective decomposition evolutionary algorithm based on double-faced mirror boundary theory) adı verilen akıllı bir optimizasyon yöntemi geliştirmişlerdir. Son olarak önerdikleri algoritmanın geçerliliğini çeşitli sayısal deneylerle kanıtlanmışlardır.

Francuz ve Bányai (2022) çalışmalarında fabrika içi lojistik operasyonlarının üretimin verimliliğini ve performansını etkilediğini vurgulamışlar ve bir otomotiv tedarikçisinin üretim tesisindeki döngüsel sefer rotalarının optimizasyonunu ele almışlardır. Tek araçlı ve çok araçlı fabrika içi döngüsel dağıtım sistemi için Excel çözücü tabanlı bir metodoloji geliştirmişlerdir. Optimizasyon sürecini ve uygulanabilirliğini bir örnek aracılığıyla göstermişlerdir. Çalışmalarında herhangi bir şekilde trafik sıkışıklığı ya da taşıyıcı araçların birbirlerini engellemeleri konusuna değinmemişlerdir.

Bu bölümde incelenen fabrika içi taşıyıcı araçların rotalanması, çizelgelenmesi ve yüklenmesi ile ilgili çalışmalar özet şekilde Tablo 2.1'de sunulmaktadır. Tabloda kullanılan kısaltmaların açıklamaları şu şekildedir;

GA: Genetik algoritma
PÖÇP: Probleme özgü çözüm prosedürü
GKA: Geniş komşuluk araması
TB: Tavlama benzetimi
HEBA: Hibrit endokrin bağışıklık algoritması
GKA: Geniş komşuluk arama
TA: Tabu arama
PSO: Parçacık sürü optimizasyonu

AGKA: Adaptif geniş komşuluk arama
KKO: Karınca kolonisi optimizasyonu
GÖA: Geniş öncelikli arama
KA: Komşuluk arama
MAYAK: Modifiye edilmiş ayrık yapay arı kolonisi
MPSO: Modifiye edilmiş parçacık sürü optimizasyonu
MKKO: Memetik karınca kolonisi optimizasyonu
YA: Yerel arama
ÇYASÇAEA: Çift yüzlü ayna sınır teorisine dayanan çok amaçlı ayrıştırma evrimsel algoritması
DP: Dinamik programlama
GİA: Geri izleme algoritması

Tablo 2.1 incelendiğinde, fabrika içindeki taşıyıcı araçların rotalanması, çizelgelenmesi ve yüklenmesi problemlerini ele alan çalışmaların oldukça az bir bölümünde çakışma durmalarının göz önünde bulundurulduğu açıkça görülmektedir. Bu az sayıdaki çalışmada da daha önce belirtildiği gibi, bir araç tarafından hizmet verilen iş istasyonlarının ardışık olası gerekliliği ya da koridorların önceden tek yön olarak belirlenmesi ve rotalama yaparken bu kurala uyulması gibi kısıtlamalar getirilmektedir. Ayrıca bu çalışmalarda, çekici trenlerin her koridordan sadece bir kere geçebilmesi veya u dönüşü yapamayacağı göz ardı edilerek iki istasyon arasındaki yolun en kısa yol algoritması ile elde edilmesi gibi yaklaşımlar da mevcuttur. Ancak bu yaklaşımların bir kısmı az sayıda koridor bulunan düz ve uzun hat şeklindeki montaj hatları için kullanışlı olsa da, fabrika içinde enine ve boyuna birçok koridorun bulunduğu daha dağınık olan hücresel imalat yerleşimleri için gerçekçi ve işe yarar olmaktan çok uzaktır. Bir kısmı ise çözüm uzayını kısıtlayarak daha iyi çözümlerden vazgeçilmesi anlamına gelmektedir. TZÜ felsefesinin gerektirdiği gibi güvenilir ve sorunsuz bir malzeme tedariği için, zamanla değişen koşullar, üretim teknikleri veya parçalar sonucunda fabrika içinde daralan koridorların sebep olduğu karşılıklı ve arka arkaya çakışmaların daha kapsamlı ve detaylı bir şekilde ele alınması gerekmektedir. Bu çalışmada tanımlanan SÇDRÇ problemi ve geliştirilen matematiksel modeller ile mat-sezgisel algoritma literatürdeki bu açığı dolduracağı değerlendirilmektedir.

Tablo 2.1. Fabrika içi taşıyıcı araçların rotalanması çizelgelenmesi ve yüklenmesi ile ilgili literatür özeti

Makale Künyesi	Çakışma kontrolü	Mat. Model	Problem(ler)			Çözüm Yöntem(ler)i			Amaç(lar)					
			Rotalama	Çizelgeleme	Yükleme	Kesin çözüm	Simülasyon	Yaklaşık Çözüm	Maliyet	Süre	Envanter	Mesafe	Trafik engelleme	Kaynak kullanımı
Chen ve diğ. (2023)		✓	✓	✓	✓			AGKA	✓					
Marialuisa ve diğ. (2023)		✓	✓	✓	✓			GA-KKO		✓				
Zhou ve Zhao (2022)		✓	✓	✓	✓			ÇYASÇAEA				✓		✓
Francuz ve Bányai (2022)			✓			✓				✓				
Saez-Mas ve diğ. (2020)		✓	✓	✓	✓	✓				✓				
Mao ve diğ. (2020)		✓	✓		✓	✓		GA	✓		?			
Peng ve diğ. (2020)		✓		✓	✓	✓		PÖÇP			✓			✓
Zhou ve Zhu (2020)		✓		✓	✓			GKA-TB			✓			
Fathi ve Ghobakhloo (2020)		✓		✓	✓	✓		PÖÇP	✓					
Sipahioğlu ve Altın (2019)		✓	✓			✓			✓					
Buyukozkan ve diğ. (2019)		✓	✓		✓			Mat-sezgisel	✓					
Lu ve diğ. (2019)		✓		✓	✓			HEBA	✓		✓			
Zhou ve Peng (2019)		✓		✓		GİA		PÖÇP			✓			
Bocewicz ve diğ. (2019)	✓	✓	✓	✓	✓	✓			✓				✓	
Satoğlu ve Sipahioğlu (2018)		✓	✓	✓	✓	✓			✓					
Kluska ve Pawlewski (2018)			✓	✓			✓		✓					
Emde ve Schneider (2018)		✓	✓			✓		GKA		✓				✓
Fathi ve diğ. (2018)		✓		✓	✓	✓								✓
Zhou ve Shen (2018)		✓		✓				TA-PSO	✓	✓				
Briand ve diğ. (2018)		✓	✓	✓	✓	✓			✓					
Peng ve Zhou (2018)		✓		✓		✓		KKO-GÖA			✓			
Emde ve Gendreau (2017)		✓		✓	✓	✓		TA-KA			✓			

Tablo 2.1. (Devam) Fabrika içi taşıyıcı araçların rotalanması çizelgelenmesi ve yüklenmesi ile ilgili literatür özeti

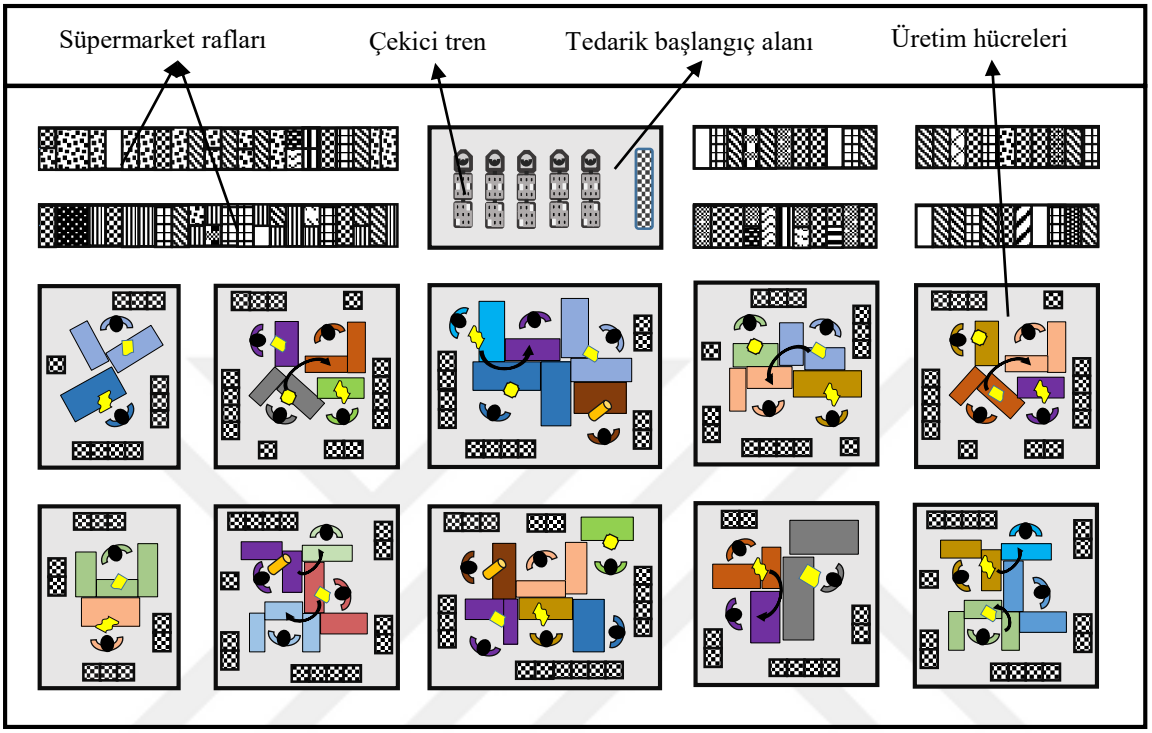
Zhou ve Peng (2017)		✓	✓	✓	✓	GİA		MAYAK			✓				
Fathi ve diğ. (2016)		✓		✓	✓	✓		MPSO			✓	✓			
Zenker ve diğ. (2016)		✓		✓	✓	DP			✓						✓
Staab ve diğ. (2016)	✓		✓	✓	✓		✓						✓		
Duman (2016)	✓	✓	✓	✓		✓		PÖÇP	✓				✓		
Fathi ve diğ. (2014a)		✓		✓	✓	✓		TB			✓	✓			
Fathi ve diğ. (2014b)		✓		✓	✓	✓		MKKO-PÖÇP			✓	✓			
Kılıç ve Durmuşoğlu (2013)		✓		✓	✓	✓		PÖÇP	✓						
Gyulai ve diğ. (2013)			✓		✓			YA				✓			
Alnahhal ve Noche (2013)		✓	✓	✓	✓	DP			✓			✓			
Satoglu ve Sahin (2013)		✓	✓		✓	✓		PÖÇP	✓						
Rao ve diğ. (2013)		✓		✓	✓	GİA		GA-TB	✓						
Faccio ve diğ. (2013)				✓	✓		✓								✓
Kilic ve diğ. (2012)		✓	✓		✓	✓						✓			✓
Emde ve Boysen (2012a)	✓		✓	✓		DP			✓						
Golz ve diğ. (2012)		✓		✓				PÖÇP							✓
Emde ve diğ. (2012)		✓			✓			PÖÇP			✓				
Akillioglu ve diğ. (2006)		✓		✓	✓	✓			✓						
Hwang (2004)		✓	✓			✓		PÖÇP							✓
Vaidyanathan ve diğ. (1999)		✓	✓		✓	✓		PÖÇP		✓					

3. PROBLEM TANIMI

TZÜ ortamında kritik bir öneme sahip olan fabrika içi lojistikte, döngüsel seferle parça tedariği, sağladığı maliyet, esneklik ve kolay yönetilebilme gibi avantajları nedeniyle oldukça yaygındır. Özellikle otomotiv, tüketici elektroniği, beyaz eşya ve mobilya gibi sektörlerde montaj hatlarına ve üretim hücrelerine parça beslenmesinde ve işlenmiş ürünlerin toplanarak depoya sevk edilmesinde etkin bir biçimde kullanılmaktadır (Küçükoğlu ve diğ., 2018; Kluska ve Pawlewski, 2018; Kılıç ve diğ., 2012; Emde ve Schneider 2018; Zheng ve diğ., 2020). Bu çalışmada otomotiv sektöründe faaliyet gösteren dünya çapındaki bir şirketin gerçek probleminden esinlenerek tanımlanan SÇDRÇ problemi, fabrika içinde döngüsel seferlerle parça tedariği yapan çekici trenlerin çakışmasız olarak rotalanması çizelgelenmesi ve yüklenmesi problemlerini eş zamanlı olarak ele almaktadır.

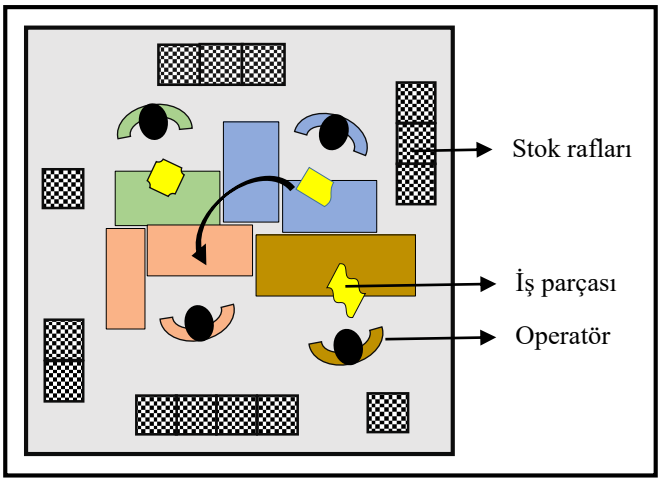
İncelenen problemin daha rahat anlaşılabilmesi için bahsedilen firmanın gerçek fabrika yerleşiminden esinlenilerek oluşturulmuş örnek bir fabrika krokisi Şekil 3.1'de gösterilmektedir. Hücresel imalat sisteminin kullanıldığı düzende, süpermarket alanlarının üretim hücrelerinin oldukça yakınına konumlandırıldığı görülmektedir. Üretim ortamlarına çok yakın bu tip dağınık stok alanlarına dayalı süpermarket konsepti, daha akıcı ve pürüzsüz bir imalat sürecini mümkün kılan küçük partiler halinde ve sürekli parça tedariğine imkan sağlaması nedeniyle son yıllarda birçok üretici tarafından sıklıkla tercih edilmektedir (Zhou ve Tan, 2020; Fathi ve diğ., 2020; Uysal ve Özcan, 2019; Alnahhal ve Noche, 2015). Böylelikle hem taşıma maliyetleri düşürülmekte hem de beklenmedik durumlara çok daha hızlı müdahale edilebilen esnek ve yönetilmesi kolay bir fabrika içi lojistiği mümkün olmaktadır (Jin ve diğ., 2021). Krokinin üst orta kısmında görülen tedarik başlangıç noktası, bütün sürecin yönetildiği ve rotasını erken tamamlayan çekici trenlerin bir sonraki çevrime kadar beklediği alandır. Genellikle yalın üretim felsefesi ve çekme sistemi ile çalışılan bu sistemlerde üretim hücrelerinden toplanan kanbanların sınıflandırılması ve sevk edilmesi gibi işlemler de bu alanda yapılmaktadır. Süpermarkette parçalar standart kutular içerisinde depolanmaktadır ve üretim hücrelerine kutu içerisinde tedarik yapılmaktadır. Her bir parçanın büyüklüğüne ve şekline göre kutu içi miktarı değişiklik göstermektedir. Bu çalışmada her bir parçaya olan ihtiyaç kutu cinsinden hesaplanarak değerlendirilmektedir. Süpermarket raflarında FIFO prensibi

uygulanmaktadır. Eğimli olarak tasarlanan raflarda yer çekimi ile parça dolu kutular sürekli öne hareket ettiği için kutu alma işlemleri süpermarket raflarının sadece tek tarafından yapılmaktadır.



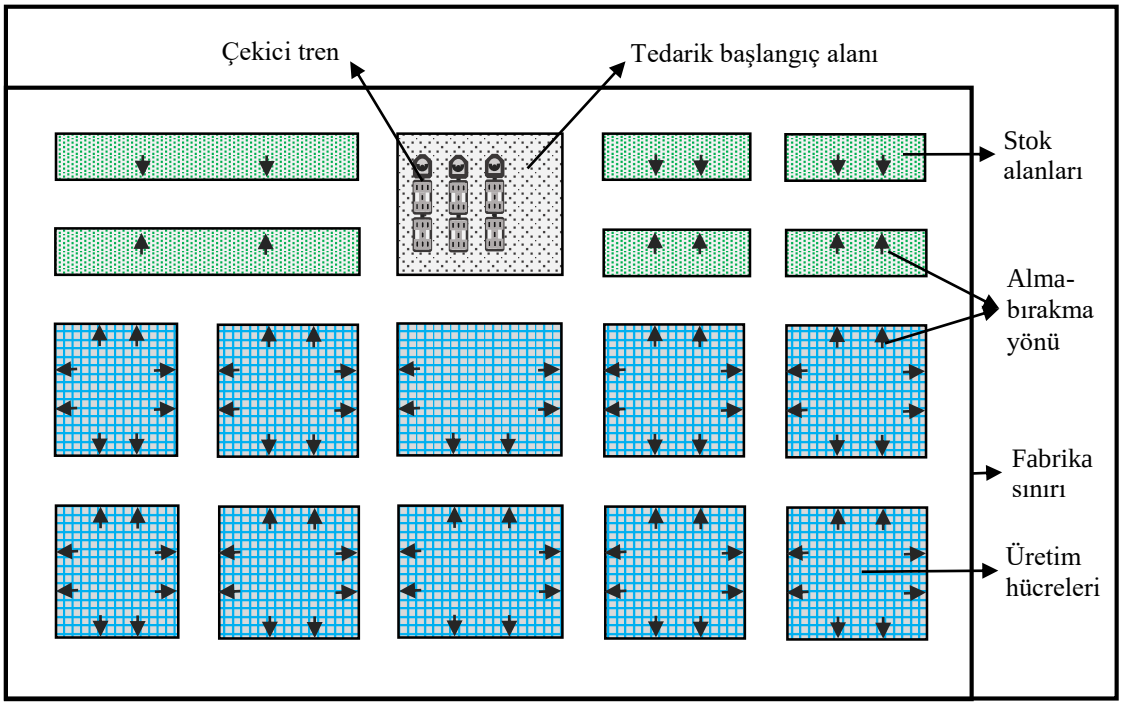
Şekil 3.1. Örnek bir fabrika krokisi

Krokinin alt iki satırında, bazıları küçük bazıları büyük olmak üzere 10 adet üretim hücresinin yer aldığı görülmektedir. Örnek bir üretim hücresi büyütülmüş şekilde Şekil 3.2'de gösterilmektedir. Fabrikanın diğer kısımlarından bağımsız olarak kendi içerisinde çalışan üretim hücrelerinin şekilde görüldüğü gibi kendi düzenleri, işleyiş şekilleri, makineleri, operatörleri, parça akış yolları ve işlem yapılan parçaların stoklandığı küçük stok alanları bulunmaktadır. Kutular içerisinde işlenmek üzere hazır bekleyen veya işlenmiş parçaları içeren stok rafları, hücrenin yapısına göre kenarda olacak şekilde çeşitli konumlarda bulunabilmektedir. Her bir raf komşu olduğu koridordan bir çekici tren operatörü tarafından dolu parça kutuları ile beslenmekte ve boş kutular da aynı operatör tarafından bu raflardan alınmaktadır. Bu şekilde üretim hücresinin parça beslemesi yapılmaktadır.



Şekil 3.2. Üretim hücresi

Şekil 3.1'de verilen fabrika yerleşim planı daha sade ve anlaşılabilir olması için Şekil 3.3'te sunulduğu şekilde basitleştirmiştir. Çalışmanın bunda sonraki bölümünde krokiler bu şekilde gösterilecektir. Üretim hücrelerindeki oklar ilgili hücreye ait besleme raflarının hangi koridora baktığını ifade etmektedir. Aynı şekilde stok alanlarındaki oklar ise ilgili süpermarketten kutuların hangi koridordan alınabileceğini göstermektedir. Gerçek fabrikalarda karşılıklı iki süpermarket rafları genellikle ortak bir koridora bakacak şekilde tasarlandığı için örnek krokide de bu şekilde yapılmıştır.



Şekil 3.3. Basitleştirilmiş fabrika krokisi görünümü

Her çevrim süresi başında tedarik başlangıç noktasında hareket eden çekici trenler, önceden belirlenmiş güzergah ve çizelgeye göre ilerleyerek kendilerine atanan görev listesine göre parçaları süpermarket raflarından kutulanmış şekilde alır ve teslim edilmesi gereken üretim hücresinin ilgili rafına dolu kutuları bırakarak boş kutuları toplarlar. Gerçek hayat uygulaması gözlemleri sonucunda ele alınan problemde üretim planının aylık periyotlar şeklinde belirlendiği ve bir ay boyunca değişmeden sabit kaldığı varsayılmaktadır. Bu nedenle $E = \{1, 2, \dots, ee\}$ ile ifade edilen görev kümesi bir ay boyunca sabit kalmakta ve kesin olarak bilinmektedir. Bir $e \in E$ görevi, belirli bir parçanın her çevrimde sabit bir kutu miktarının belirli bir süpermarket rafından alınarak önceden belirlenmiş bir üretim hücresi stok rafına teslim edilmesini içermektedir. Çekici tren operatörleri kendilerine atanan görevlere göre süpermarkette parça dolu kutuları vagonlarına yükleyip vagonlarındaki boş kutuları doldurulmak üzere süpermarket raflarına koymaktadırlar. Üretim hücrelerine geldiklerinde ise hücrelerin raflarındaki boş kutularla vagonlarındaki dolu kutuları değiştirerek parça beslemesi yapmaktadırlar. Dolayısıyla çekici trenlerin, kendilerine atanan görevlerin kutularını alacakları veya bırakacakları raflar hangi koridorlara bakıyorsa o koridorlardan geçerek parça dağıtımını yapmaları gerekmektedir.

Her çevrimde aynı rotaları ve çizelgeleri takip edecekleri için çekici trenlerin önce süpermarket raflarına veya üretim hücresi stok raflarına gitme zorunlulukları bulunmamaktadır. Yani önce süpermarket raflarını sonra üretim hücrelerini dolaşabilecekleri gibi, önce üretim hücrelerini sonra süpermarket raflarını da dolaşabilirler. Ya da öncelikle süpermarket raflarının bir kısmını gezdikten sonra üretim hücrelerini gezip sonra tekrar süpermarket raflarına dönebilirler. Parçaları stoklamak için kullanan kutular standart olduğu için kutu içi miktarlar değişse de operatörlerin dolu kutuları süpermarketten alma ve üretim hücresine bırakma süreleri kutu başına sabittir ve bilinmektedir. Aynı şekilde boş kutuların üretim hücrelerinden alınma ve süpermarkette bırakılma süreleri sabittir ve bilinmektedir. Çalışmada alma ve bırakma süreleri bir durakta elleçlenen kutu miktarı ile kutu başına işlem süresinin çarpılması ile elde edilmektedir.

Bu çalışmada $K = \{1,2, \dots kk\}$ kümesi ile ifade edilen çekici trenler, manuel olarak kontrol edilen ve elektrik ile çalışan çekici bir lokomotif ve ona bağlı iki adet vagondan oluşmaktadır. Çekici trenlerin taşıyabilecekleri yük miktarları, içindeki parçanın cinsi ya da ağırlığı ne olursa olsun standart kutu cinsinden ifade edilmektedir ve her bir çekici trenin eşit miktarda yük taşıyabilen, eşit hızlara sahip özdeş araçlar olduğu varsayılmaktadır. Bunun temel nedeni düz bir zeminde ve yeterince güçlü bir lokomotifle hareket eden trenlerin vagonlarının raf sınırlaması olmasıdır. İki vagon ve bir lokomotiften oluşan bu araçların uzunluğu sebebiyle dönüş çapları çok fazladır ve bu nedenle fabrika içerisinde herhangi bir noktada u dönüşü yapamamaktadırlar. Bu nedenle çekici trenlerin rotalarının belirlenmesinde özel olarak u dönüşünü engelleyecek kısıtlar modele dahil edilmiştir.

Fabrika içi döngüsel seferlerle parça tedariğini konu alan çalışmalara bakıldığında çevrim süresinin sabit veya değişken olarak kabul edildiği çeşitli uygulamaların olduğu görülmektedir. Her ne kadar çevrim süresinin sabit ya da değişken olması yöneticilerin inisiyatifinde olsa da TZÜ felsefesi ile çalışan ve yalın üretimi benimsemiş fabrikalarda yapılan gözlemler ve literatür taraması sonucu elde edilen bulgular sabit çevrim süresinin çok daha yaygın kullanıldığını ve daha fazla avantaja sahip olduğunu göstermektedir (Kılıç ve Durmuşoğlu, 2013). Bu avantajların başında özellikle operatörler ve süreç yöneticileri açısından standart ve sabit bir çevrim süresinin uygulama ve yönetim kolaylığı sağlaması gelmektedir. Bununla birlikte, bir sistemin döngüsel sefer sistemi olarak kabul edilebilmesi için sabit çevrim süresinin bir gereklilik olduğu da ifade edilebilir. (Bozer ve Ciemnoczolowski (2013). Bu nedenlerle bu çalışmada çevrim sürelerinin her periyot için sabit olduğu varsayılmıştır. Örneğin 30 dakikalık çevrim sürelerinin olduğu bir sistemde bütün çekici trenler 30 dakika arayla tedarik başlangıç noktasında birlikte hareket etmekte ve çevrim süresi sona ermeden turlarını tamamlayarak tedarik başlangıç noktasına geri dönmektedirler.

TZÜ ortamında döngüsel seferlerle kutular içerisinde az ve sık parça tedariğinin maliyet, süreç içi envanter, kolay yönetilebilirlik gibi sağladığı önemli avantajlar olsa da bu sistem; çekici trenler, fabrika içi koridorlar, trafik etmeni gibi birbiriyle ilişki içerisinde olan birçok unsur içerdiği için oldukça karmaşıktır ve iyi planlanması gerekmektedir

(Staab ve diğ., 2016). Aksi takdirde üretim hücrelerinin parça kıtlığı yaşayarak üretimin aksaması ve maddi kayıpların yaşanması ciddi bir olasılıktır. Genellikle düz ve uzun bir çizgi şeklinde olan montaj hatlarından farklı olarak bu çalışmada ele alınan üretim ortamının yerleşimi Şekil 3.3'te görüldüğü gibi iki boyutlu olarak daha dağınık ve karmaşıktır. Hücresel imalat yerleşiminin doğasından kaynaklanan bu tip fabrika krokilerinde özellikle TZÜ felsefesinin de getirdiği dakik olma gereksinimi nedeniyle taşıyıcı araçların rotalanması ve çizelgelenmesi problemlerinin daha zor olduğu ve bu sebeple daha dikkatli ele alınması gerektiği açıktır. Bunun nedenlerinden biri, üretim hücrelerinin dört tarafında da konumlandırılabilen malzeme stok raflarından dolayı hem yatay hem de dikey koridorlardan geçmesi gereken araçlar için rotalamanın çok daha karmaşık olmasıdır.

Diğer önemli bir nedeni ise, başlangıçta geniş tasarlansalar da zamanla meydana gelen değişiklikler sonucunda koridorların iki çekici trenin aynı anda geçemeyeceği şekilde daralabilmesidir. Özellikle günümüzde hızlı değişen müşteri tercihleri nedeniyle zamanla üretilen ürünlerin tasarımlarının değişmesine paralel olarak imal edilen parçaların şekil ve boyutlarındaki değişim, üretildikleri hücrelerin daha fazla alana gereksinim duymasına neden olabilmektedir. Ayrıca değişen üretim teknikleri ile yeni veya farklı makinelere olan gereksinim sonucu kullanılan teçhizatın değişmesi de, hücre yapısının büyümesine veya formunun değişmesine neden olabilmektedir. Üretim hücrelerinin veya süpermarket alanlarının boyut ve formlarındaki bu değişim ihtiyaçları, zamanla fabrika içerisindeki yatay veya dikey koridorların daralması ile sonuçlanmaktadır.

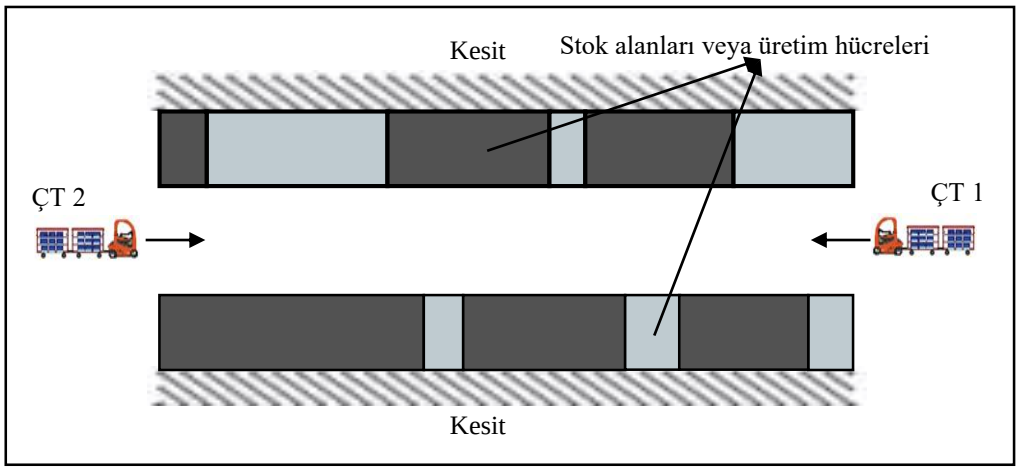
Üretilen ürünlerin tasarımları ve modelleri değişse bile montaj hatlarının genişlemesi ve hat kenarındaki besleme koridorlarının daralması genellikle söz konusu değildir. Bu sebeple montaj hatlarına parça besleme ile ilgili yapılan çalışmaların neredeyse hiçbirinde çekici trenlerin rotalanması ve çizelgelenmesine yönelik geliştirilen modeller ve algoritmalarda dar yollar nedeniyle oluşabilecek trafiğin dikkate alınmadığı görülmektedir (Zhou ve Zhu, 2020; Chen ve diğ., 2023; Lu ve diğ., 2019; Zhou ve Peng, 2019; Peng ve Zhou, 2018; Briand ve diğ., 2018). Emde ve Boysen (2012a) gibi bazı nadir çalışmalarda ise montaj hattı boyunca çekici trenlerin sadece önceden belirlenmiş tek bir yönde hareket etmesini zorunlu hale getirerek olası çakışmaların engellenmesi

hedeflenmektedir. Ancak hücresel yerleşimin olduğu fabrikalarda zamanla dar koridorların oluşması yukarıda ifade edilen sebeplerden dolayı oldukça sık karşılaşılan bir durumdur (Bocewicz ve diğ., 2019; Alizon ve diğ., 2009; Staab ve diğ., 2016). Bu nedenle bu tip yerleşime sahip fabrikalarda çekici trenlerin rotalanması ve çizelgelenmesi için geliştirilen modeller ve algoritmalarda trafik sıkışıklığı durumu dikkate alınması gereken önemli bir sorun teşkil etmektedir. Bu bağlamda çekici trenlerin en kısa yoldan gidecek şekilde rotalanması ve çizelgelenmesi, bu tip üretim ortamlarında çoğu zaman bloklamalara, trafik kaynaklı beklemelere, kargaşa ortamının oluşmasına ve üretim gecikmelerine sebep olabilmektedir (Staab ve diğ., 2016). Unutulmamalıdır ki TZÜ felsefesinde amaç sadece stok seviyelerini düşürmek değil aynı zamanda istenen ürünlerin istenen yerde ve istenen zamanda bulunmasıdır. Bunu gerçekleştirebilmek için dar yolların bulunduğu, iki boyutlu olarak geniş bir alana yayılmış üretim ortamlarında trafik gibi çevresel faktörlere karşı dirençli, iyi planlanmış güvenilir bir parça tedarik sistemi oldukça kritiktir.

Bu sebeplerle bu çalışmada tanımlanan SÇDRÇ problemi çekici trenlerin yüklenmesi, rotalanması ve çizelgelenmesi problemlerini fabrika içerisindeki dar koridorlarda araçların birbirlerini bloklamasını engelleyecek ve trafik kaynaklı beklemeleri en aza indirilecek şekilde ele almaktadır. Bu çalışma kapsamında yukarıda belirtilen nedenlerden dolayı oluşan dar koridorlarda, şu iki farklı çakışma durumu incelenmektedir.

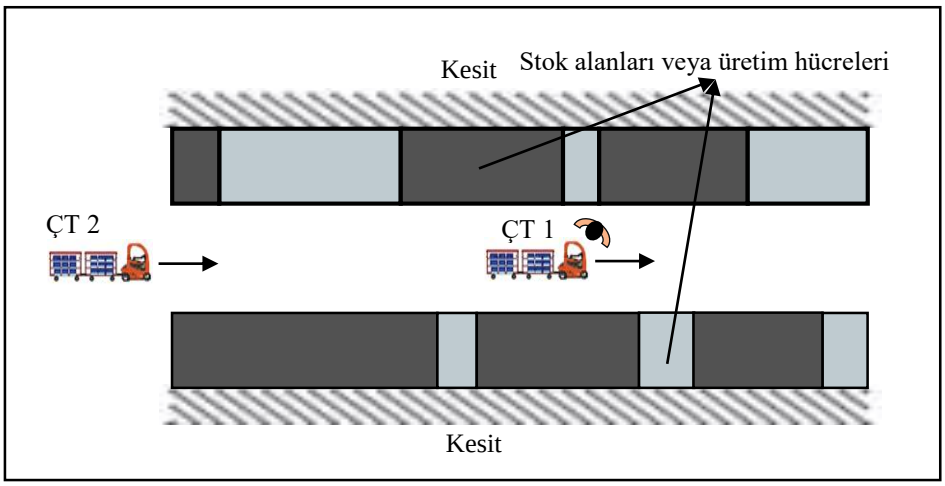
1. Farklı yönde giden çekici trenlerin birbirini bloklaması,
2. Aynı yönde giden çekici trenlerin birbirini bloklaması.

Birinci durum; kutu alma/bırakma işlemi yapsın ya da yapmasın iki çekici trenden birinin dar bir koridordan geçerken diğerinin karşı yönden gelmesi sonucu birbirlerini bloklamalarını ifade etmektedir. Daha iyi anlaşılması için Şekil 3.4'te gösterilen bu senaryoda, çakışma olmaması için çekici trenlerin aynı anda dar bir koridora karşılıklı olarak girmemesi gerekmektedir. Örneğin çekici tren 1 (ÇT 1) koridordan geçtikten sonra çekici tren 2 (ÇT 2) koridora girse ya da ÇT 2 koridordan geçtikten sonra ÇT 1 koridora girse bu çakışma olmayacaktır.



Şekil 3.4. Dar koridorda çekici trenlerin karşılıklı çakışması durumu

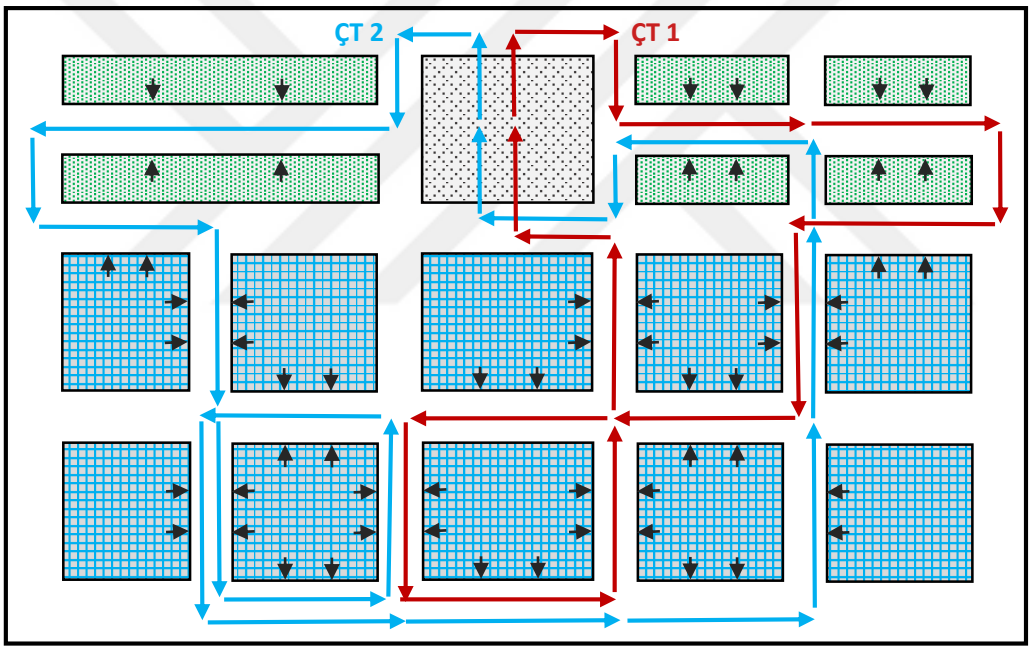
İkinci durum ise; bir çekici trenin dar bir koridora girip kutu alma/bırakma işlemi yapmak için durduğu sırada başka bir çekici trenin aynı yönde koridora girerek önündeki trenin işleminin bitmesini beklediği senaryodur. Bu tür bir çakışma, tren hızları aynı olduğu için sadece koridora erken giren çekici trenin sonra giren trene göre daha uzun süreli kutu alma/bırakma işlemi yapması gerekiyorsa ortaya çıkacaktır. Bu çakışma durumu Şekil 3.5'te görselleştirilmektedir. Şekilde görüldüğü gibi koridorda durup kutu alma/bırakma işlemi yapan ÇT 1 aynı yönden gelen ÇT 2'nin geçişini engellemektedir. Eğer koridora sonra giren trenin kutu alma/bırakma süresi, önce giren trenden daha fazla ise veya ilk giren tren, kutu alma/bırakma işlemi yapmayacaksa çakışma durumu oluşmayacaktır.



Şekil 3.5. Dar koridorda çekici trenlerin arka arkaya çakışması durumu

Bu çalışmada tanımlanan SÇDRÇ problemi E görev kümesinin tamamının gerçekleştirilmesi için çekici trenlerin başta çevrim süresi, kapasite, çakışma olmak üzere

bir dizi kısıt altında yüklenmesi, rotalanması ve çizelgelenmesi problemlerini birlikte ele almaktadır. SÇDRÇ problemi, toplamalı dağıtmalı ve zaman pencereli araç rotalama problemine benzese de bazı noktalarda farklılaşmakta ve kendine özgü varsayımları ve kısıtları bulunmaktadır. Örneğin Şekil 3.3'te verilen kroki üzerinde iki çekici tren ile yapılacak bir malzeme dağıtım sisteminde, trenlerin optimum rotaları Şekil 3.6'da sunulduğu gibi olabilmektedir. Klasik araç rotalama probleminde bir araç bir kenardan en fazla bir kere geçebilirken Şekil 3.6'da görüldüğü gibi bir çekici tren bir koridordan iki kere hatta bazı durumlarda üç kere geçmesi gerekebilmektedir. Diğer bir nokta ise çekici trenlerin uzun olmaları nedeniyle u dönüşü yapamamalarıdır. Klasik araç rotalama probleminden farklı olarak çekici trenler bir koridora girdikten sonra geri dönememekte ve o koridoru geçip farklı bir koridora yönelmeleri gerekmektedir.



Şekil 3.6. Çekici trenler için örnek bir rotalama gösterimi

Gerçek uygulamada gözlemlenen ve SÇDRÇ probleminin tanımında ve modellenmesinde dikkate alınan temel varsayımlar şu şekilde özetlenebilir:

- Fabrika yerleşimi (kroki) sabittir ve bir değişiklik yapılmamaktadır,
- Üretim planı aylık olarak bilinmektedir ve değişmemektedir,
- Tekrarlı ve istikrarlı bir üretim mevcuttur,
- Parça tedariği eşit çevrim sürelerine sahip sabit periyotlarla yapılmaktadır,

- Periyotların çevrim süreleri her çekici tren için aynıdır ve sabittir,
- Bütün çekici trenler periyot başında aynı anda tedarik başlangıç noktasından hareket ederler ve turunu erken bitiren trenler diğer periyot başlangıcına kadar beklerler.
- Çekici trenler turlarına tedarik başlangıç noktasından, önceden belirlenmiş bir yöne doğru hareket ederek başlarlar ve tur sonunda başlangıç alanına geldikleri yönün tersi yönden giriş yaparlar,
- Çekici trenlerin rota ve çizelgeleri her periyot için sabittir ve üretim planı değişmediği sürece değişmemektedir,
- Çekici trenlerin hızları sabittir ve bilinmektedir,
- Çekici trenlerin kapasiteleri kutu cinsinden ifade edilmektedir, sınırlıdır ve bilinmektedir,
- Kapasiteleri ve hızları aynı olan özdeş çekici trenler kullanılmaktadır,
- Turlar sırasında araç arızası meydana gelmemektedir,
- Görevler bölünemez yapıdadır, her görev mutlaka ve sadece bir çekici tren tarafından yapılmalıdır,
- Bütün parçalar, standart büyüklük ve şekildeki kutuların içerisinde stoklanmakta ve taşınmaktadır,
- Her hücrenin her stok noktasının talebi periyot başına kutu cinsinden bilinmektedir ve deterministiktir,
- Süpermarket raflarında parçalar kutulanmış şekilde yeterince mevcuttur ve kıtlık yaşanmamaktadır,
- İçindeki parçadan bağımsız olarak kutu başına alma ve bırakma süreleri bilinmektedir ve sabittir.

4. ÇÖZÜM YAKLAŞIMLARI

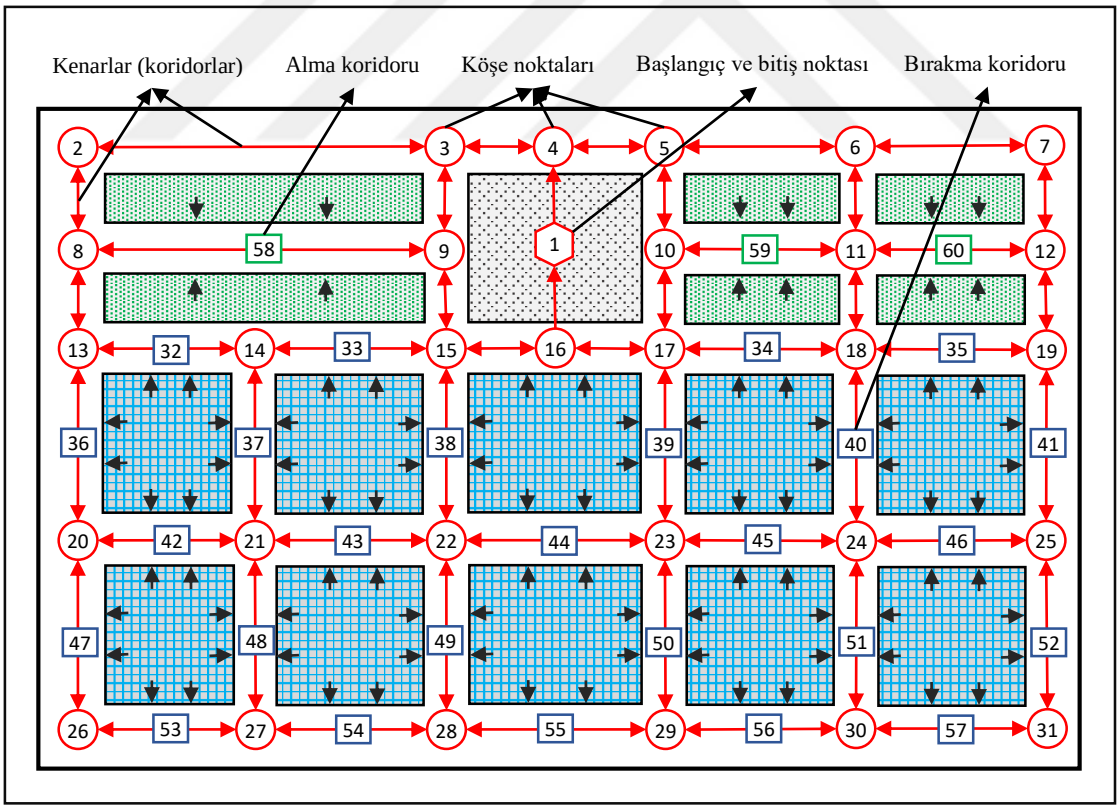
Gerçek hayat gözlemleri ve literatürden yola çıkarak günümüzde döngüsel seferlerle fabrika içinde malzeme dağıtımı yapan araçların rotalarının çoğunlukla üretim planlamacıları tarafından uzun süreler uğraşılarak elle çizildiği ifade edilebilir (Staab ve diğ., 2016). Taşıyıcı araçların dar koridorlarda birbirlerini bloklayarak bekletmelerini engelleyecek şekilde döngüsel seferlerin rotalanması ve çizelgelemesi oldukça fazla hesaplama gerektiren karmaşık bir süreçtir. Özellikle büyük boyutlu krokiler ile yüksek miktarda araç ve görev sayısı bu işi çok daha karmaşık bir hale getirmektedir. Bu nedenle çekici trenler için elle oluşturulan rotalarda ve çizelgelerde çakışma durumunun dikkate alınamadığı görülmektedir. Bunun sonucunda birçok dar koridorda çakışan çekici trenler nedeniyle üretim ortamında trafik kaynaklı beklemeler, karmaşa ve verimsizlik ortaya çıkmaktadır.

Bu çalışmada SÇDRÇ probleminin çözümüne yönelik olarak üç farklı karma tam sayılı matematiksel model ve her birinde bu matematiksel modellerin bir veya bir kaçının kullanıldığı üç farklı çözüm yaklaşımı geliştirilmiştir. Bu çözüm yaklaşımları ile TZÜ felsefesine uygun olarak istenilen parçanın istenilen üretim hücresinin rafında istenilen zamanda olmasını garanti edecek şekilde çekici trenler için trafiksiz, güvenilir ve robüst bir rotalama ve çizelgeleme planının elde edilmesi hedeflenmektedir.

4.1. Birinci Çözüm Yaklaşımı: Karma Tam Sayılı Bir Model (Model-1)

Bu bölümde SÇDRÇ probleminin çözümüne yönelik olarak karma tam sayılı bir doğrusal programlama modeli önerilmektedir. Model-1 olarak isimlendirilen bu model, çakışma ve diğer tüm kısıtları göz önünde bulundurarak çekici trenler için tam bir yükleme, rotalama ve çizelgeleme çözümü sunmaktadır. Problem Şekil 4.1.'de gösterildiği gibi $G = (N, A)$ yönlendirilmiş ağı üzerinde tanımlanmaktadır. Burada N, başlangıç ve bitiş noktası olan "1" dahil olmak üzere, şekilde 32'ye kadar olan kırmızı yuvarlaklar içerisine alınmış köşe noktaları kümesini, A ise X veya Y koordinatı aynı olan komşu iki köşe noktası arasındaki kenarlar (koridorlar) kümesini temsil etmektedir. Şekil 4.1'de görüldüğü üzere bütün kutu alma koridorlarına yeşil dikdörtgenler, kutu bırakma koridorlarına ise mavi dikdörtgenler içerisinde koridor numaraları verilmiştir. Ayrıca bütün koridorlarda iki yöne hareket etmek mümkünken başlangıç noktasına tek yönden giriş ve bu noktadan tek yöne çıkış yapılabildiği gösterilmektedir.

İlk olarak üretim hücresi ve süpermarket raflarının her birinin önüne bir köşe noktası konularak problemin bu noktalar üzerinden modellenmesi düşünülmüştür. Ancak yüksek miktarda nokta sayısının, değişken sayısını çok fazla artıracağı ve böylece modelin çözülmesi çok zor olacağı için bu düşünceden vazgeçilmiştir. Ayrıca çekici trenler u dönüşü yapamadığından dolayı bunun gereksiz olduğuna karar verilmiştir. Çünkü bir çekici tren bir koridora girdiğinde yüksek dönüş çapı nedeniyle u dönüşü yapamadığı için o koridoru sonuna kadar gitmek ve daha sonra geldiği koridordan başka bir koridora yönelmek zorundadır. Bu sebeple her koridorun ortasına sadece bir köşe noktası konularak problemin modellenmesi planlanmıştır. Daha sonra koridorların, köşe noktaları üzerinden ifade edilebileceği düşünülerek koridor ortalarındaki köşe numaraları da kaldırılmış ve değişken tanımlarında dikkate alınmayan, Şekil 4.1'de dikdörtgenler içerisinde gösterilen koridor numaraları verilmiştir. Nihai modelde değişkenler, Şekil 4.1'de görülen başlangıç noktası "1" ve yuvarlak içerisindeki köşe noktaları üzerinden tanımlanmıştır. Böylece değişken sayısının ciddi miktarda azaltılması sağlanmıştır.



Şekil 4.1. Krokilerdeki köşe - başlangıç noktalarının ve koridorların numaralandırılması

Bu bölümde verilen Model-1, her periyot başında başlangıç noktası olan "1"den hareket eden çekici trenlerin periyot sonunda geri dönene kadar adım adım hangi koridorlardan hangi sırayla geçeceklerini, koridorlardan hangi geçişleri sırasında ne kadar kutu alacaklarını veya bırakacaklarını veren ve her bir çekici trenin her köşe noktasından kaçıncı saniyede geçtiğini hesaplayarak karşılıklı ve arka arkaya çakışmaları engelleyen ayrıntılı bir yükleme, rotalama ve çizelgeleme çözümü sunmaktadır. Modelde $T = \{1,2,\dots,tt\}$ hareketler kümesini ifade etmektedir. Her hareket bir çekici trenin bir köşe noktasından diğer bir köşe noktasına gitmesini içermektedir. Örneğin bir trenin Şekil 4.1'de gösterilen kroki üzerinde "1" noktasından "4" noktasına gitmesi ya da "9" noktasından "8" noktasına gitmesi bir harekettir. Çekici trenler her harekette yalnızca bir köşe noktasından komşu diğer bir köşe noktasına gidebilirler.

$E = \{1,2,\dots,ee\}$ kümesi ile ifade edilen görevlerlerin her biri Tablo 4.1'de verilen örnek listede görüldüğü gibi, alma koridoru, bırakma koridoru ve kutu cinsinden periyot başına taşınacak miktar bilgilerini içermektedir. Bir görevin yapılabilmesi için atandığı çekici trenin her periyotta görevin alma koridoruna giderek ilgili raftan kutulanmış parçaları gerektiği miktarda alması ve bırakma koridorunda ilgili üretim hücresinin ilgili stok rafına teslim etmesi gerekmektedir. Görevler bölünemez olduğu için her görev mutlaka ve sadece bir çekici trene atanmalıdır.

Tablo 4.1. Örnek bir görev listesi

Görev Sıra No	Alma Koridoru	Bırakma Koridoru	Miktar/Periyot (kutu)
1	58	42	4
2	59	56	2
3	60	40	2
4	60	34	2
5	59	38	3

SÇDRÇ probleminin çözümüne yönelik olarak geliştirilen Model-1'de kullanılan kümeler, indisler, parametreler ve değişkenler aşağıda özetlenmektedir.

Kümeler:

$T = \{1,2,\dots,tt\}$: Hareketler kümesi (Planlama ufku)

$\Pi = \{1,2, \dots \pi\pi\}$: Tüm noktalar kümesi

N : Köşe noktaları kümesi ($N \subset \Pi$)

Φ : Bırakma noktaları kümesi ($\Phi \subset \Pi$)

Γ : Alma noktaları kümesi ($\Gamma \subset \Pi$), ($N \cup \Phi \cup \Gamma = \Pi$)

$A = \{(i,j) \mid i,j \in N, i \neq j, (i,j) komşu\ ikililer\}$: Bütün koridorlar kümesi

β : Arasında bırakma noktası olan koridorlar kümesi ($\beta \subset A$)

θ : Arasında alma noktası olan koridorlar kümesi ($\theta \subset A$)

DY : Dar koridorlar kümesi ($DY \subset A$)

$K = \{1,2, \dots kk\}$: Araçlar kümesi

$E = \{1,2, \dots, ee\}$: Görevler kümesi

F_{ij} : Alma veya bırakma koridoru (i,j) komşu köşe noktaları arasında olan görevler kümesi ($F_{ij} \subset E$)

İndisler:

i,j : Köşe noktaları (düğümler) ($i,j \in N$)

k,f : Araçlar ($k,f \in K$)

t,y : Hareketler ($t,y \in T$)

e : Görevler ($e \in E$)

Parametreler:

d_{ij} : $(i,j) \in A$ yolunun uzunluğu (metre)

v : Çekici trenlerin hızı (metre/saniye)

q_e : e görevinin periyot başına kutu miktarı

CS : Rotaların aşmaması gereken çevirim süresi (saniye)

C : Çekici trenlerin kutu cinsinden kapasitesi (kutu)

PBA_e : e görevinin bırakma koridorunun alt köşe noktası

PBU_e : e görevinin bırakma koridorunun üst köşe noktası

PAA_e : e görevinin alma koridorunun alt köşe noktası

PAU_e : e görevinin alma koridorunun üst köşe noktası

BK_{ij}^e : e görevinin bırakma koridoru, (i,j) komşu köşe noktaları arasında ise 1, değilse 0'dır

AK_{ij}^{e} : e görevinin alma koridoru, (i,j) komşu köşe noktaları arasında ise 1, değilse 0'dır

BBS : Bırakılan kutu başına birim işlem süresi (saniye)

ABS : Alınan kutu başına birim işlem süresi (saniye)

M : Büyük bir sayı

Değişkenler:

x_{ij}^{kt} : İkili değişken, k aracı t hareketinde i noktasından j noktasına gidiyorsa 1 değerini alır, aksi durumda 0 değerini alır, $\forall(i,j) \in A, \forall k \in K, \forall t \in T$

s_{ij}^{kt} : İkili değişken, k aracı t hareketinde (i,j) köşe noktaları arasında hizmet veriyorsa 1 değerini alır, aksi durumda 0 değerini alır, $\forall(i,j) \in \beta \cup \theta, \forall k \in K, \forall t \in T$

p_{e}^{k} : İkili değişken, e görevi k aracına atanırsa 1 değerini alır, aksi durumda 0 değerini alır, $\forall e \in E, \forall k \in K$

r^{kt} : k aracının t hareketinde bıraktığı toplam dolu kutu miktarı $\forall k \in K, t \in T$

l^{kt} : k aracının t hareketinde aldığı toplam dolu kutu miktarı $\forall k \in K, t \in T$

w^{kt} : k aracının t hareketinde bekleme süresi (saniye) $\forall k \in K, \forall t \in T$

z^{kt} : k aracının t hareketi sonunda köşe noktasından geçiş zamanı $\forall k \in K, \forall t \in T$

u_{ij}^{ktfy} : İkili değişken, karşılıklı çakışma engelleme kısıtlarını modellemek için kullanılan yardımcı değişken $(u_{ij}^{ktfy} \in \{0,1\})$

h_{ij}^{ktfy} : İkili değişken, arka arkaya çakışma engelleme kısıtlarını modellemek için kullanılan yardımcı değişken $(h_{ij}^{ktfy} \in \{0,1\})$

Oldukça kapsamlı olan Model-1 fazla sayıda kısıt denklemi içerdiği için, kısıtlar; yol-denge kısıtları, hizmet verme, kapasite ve çevrim süresi kısıtları, zaman hesaplama kısıtları, çakışma engelleme kısıtları ve değişken sınırlama kısıtları olmak üzere beş parça halinde verilecek ve anlatılacaktır. Modelin iki kısımdan oluşan amaç fonksiyonu Denklem (4.1)'de gösterilmektedir. İlk kısmı çekici trenlerin parça dağıtımı için yolda geçirdikleri süre toplamıdır. Denklemde görüldüğü gibi bu değer, bütün çekici trenlerin kat ettikleri toplam yol uzunluğunun hız ile çarpılmasıyla elde edilmektedir. İkinci kısmı ise çekici trenlerin bütün hareketleri boyunca bekleme yapmak zorunda kaldıkları

sürelerin toplamlarıdır. Amaç fonksiyonu bu iki değerin toplamının en küçüklenmesini içermektedir.

Bu yaklaşım ile model, çakışma olmaması için çekici trenlerin rotalarını değiştirmeye zorlamaktadır, ancak eğer rotalarını değiştirmeleri toplam süreyi yani yolu çok uzatacaksa bu durumda model çakışacak olan iki araçtan birinin dar koridora girmeden önce geniş bir koridorda kısa bir süre beklemesini seçebilmektedir. Böylelikle dar koridorlarda karşılıklı ve arka arkaya çakışma olmaksızın en kısa sürede parça dağıtımının gerçekleştirilmesi sağlanmış olacaktır.

$$minZ = \sum_{(i,j)\in A}\sum_{k\in K}\sum_{t\in T} x_{ij}^{kt} d_{ij} v \; + \sum_{k\in K}\sum_{t\in T} w^{kt} \tag{4.1}$$

Aşağıdaki kısıtlar altında:

Yol-denge kısıtları;

$$\sum_{\substack{(i,j)\in A \\ i\neq 1}} x_{ij}^{k[1]} = 0, \qquad \forall k \in K \tag{4.2}$$

$$\sum_{\substack{t\in T \\ t\neq 1}} \sum_{\substack{(i,j)\in A \\ i=1 \\ j\neq 1}} x_{ij}^{kt} = 0, \qquad \forall k \in K \tag{4.3}$$

$$\sum_{\substack{t\in T \\ t\neq 1}} \sum_{\substack{(i,j)\in A \\ j=1 \\ i\neq 1}} x_{ij}^{kt} = \sum_{\substack{(i,j)\in A \\ i=1 \\ j\neq 1}} x_{ij}^{k[1]}, \qquad \forall k \in K \tag{4.4}$$

$$\sum_{\substack{t\in T \\ t\neq 1}} x_{j[1]}^{kt} + x_{[1]j}^{k[1]} \leq 1, \qquad \forall k \in K, \forall j \in N - \{1\} \tag{4.5}$$

$$\sum_{i\in N} x_{ij}^{kt} - \sum_{i\in N} x_{ji}^{k(t+1)} = 0, \qquad \forall k \in K, \forall j \in N - \{1\}, \forall t \in T - \{tt\} \tag{4.6}$$

$$\sum_{(i,j)\in A} x_{ij}^{kt} \leq 1, \qquad \forall k \in K, \forall t \in T \tag{4.7}$$

$$x_{ij}^{kt} + x_{ji}^{k(t+1)} \leq 1, \qquad \forall (i,j) \in A, \forall k \in K, \forall t \in T - \{tt\} \tag{4.8}$$

Denklem (4.2) ve Denklem (4.3), çekici trenlerin sadece birinci hareketlerinde ve gerekiyorsa başlangıç düğümü olan "1" noktasından çıkış yapmasını sağlamaktadır. Bu denklemlerle herhangi bir trenin birinci hareketi dışında başlangıç noktasından diğer bir noktaya hareket etmesi engellenmektedir. Denklem (4.4), çevrim süresi başında başlangıç noktasından parça dağıtmak için çıkan trenlerin birinci hareketleri dışında herhangi bir hareketlerinde geri başlangıç düğümüne dönmesini garanti etmektedir. Denklem (4.5) ise trenlerin başlangıç noktasına, çıkış yaptıkları yönden girememelerini sağlamaktadır. Bu kısıtın modele dahil edilmesinin nedeni çekici trenlerin u dönüşü yapamamalarıdır. Denklem (4.6) her bir köşe noktası için geliş ve gidişi dengelemek için oluşturulmuştur. Buna göre t hareketi sonunda herhangi bir j köşe noktasına gelen bir tren $(t+1)$ hareketinde j noktasından ayrılmalıdır. Denklem (4.7) her çekici trenin her bir harekette yalnızca bir koridorda ilerleyebilmesini sınırlandırmaktadır. Denklem (4.8) ise u dönüşünü engelleme kısıtıdır. Buna göre çekici trenler $(t+1)$ hareketinde, t hareketinde gittiği bir koridorun tersi yönünde hareket edemezler.

Hizmet verme, kapasite ve çevrim süresi kısıtları;

$$s_{ij}^{kt} \leq x_{ij}^{kt}, \qquad \forall (i,j) \in \beta \cup \theta, \forall k \in K, \forall t \in T \tag{4.9}$$

$$\sum_{t\in T} s_{ij}^{kt} + s_{ji}^{kt} \leq 1, \qquad \forall (i,j) \in \beta \cup \theta, \forall k \in K \tag{4.10}$$

$$\sum_{k\in K} p_e^k = 1, \qquad \forall e \in E \tag{4.11}$$

$$\sum_{t\in T} (s_{PBA_e\, PBU_e}^{kt} + s_{PBU_e PBA_e}^{kt}) \geq p_e^k, \qquad \forall e \in E, \forall k \in K \tag{4.12}$$

$$\sum_{t\in T} (s_{PAA_e\, PAU_e}^{kt} + s_{PAU_e PAA_e}^{kt}) \geq p_e^k, \qquad \forall e \in E, \forall k \in K \tag{4.13}$$

$$\sum_{t\in T}\left(s_{ij}^{kt}+s_{ji}^{kt}\right)\le\sum_{e\in F_{ij}}p_e^k,\qquad \forall(i,j)\in\beta\cup\theta, i<j\ ,\forall k\in K \tag{4.14}$$

$$\sum_{e\in E}p_e^k\,q_e\le C,\qquad \forall k\in K \tag{4.15}$$

$$\sum_{t\in T}\sum_{(i,j)\in A}x_{ij}^{kt}d_{ij}+\sum_{e\in E}p_e^k\,q_e(BBS+ABS)+\sum_{t\in T}w^{kt}\ \le CS,\qquad \forall k\in K \tag{4.16}$$

$$r^{kt}=\sum_{e\in E}\left(p_e^k q_e BK_{ij}^e\right)s_{ij}^{kt},\qquad \forall(i,j)\in\beta,\ \ \forall k\in K,\forall t\in T \tag{4.17}$$

$$r^{kt}\le\sum_{(i,j)\in\beta}s_{ij}^{kt}\,M\ ,\qquad \forall k\in K,\forall t\in T \tag{4.18}$$

$$l^{kt}=\sum_{e\in E}(p_e^k\,q_e AK_{ij}^e)\,s_{ij}^{kt}\ ,\qquad \forall(i,j)\in\theta,\ \ \forall k\in K,\forall t\in T \tag{4.19}$$

$$l^{kt}\le\sum_{(i,j)\in\theta}s_{ij}^{kt}\,M,\qquad \forall k\in K,\forall t\in T \tag{4.20}$$

Denklem (4.9) eğer alma veya bırakma koridorunda hizmet veren (kutu alma-bırakma işlemi yapan) bir çekici trenin o koridordan geçmesini garanti etmektedir. Denklem (4.10) alma veya bırakma koridorlarına her trenin sadece tek bir hareketi sırasında hizmet verebilmesini sınırlandırmaktadır. Bu kısıtın eklenmesinin nedeni çekici trenlerin bir periyotta aynı koridora hizmet vermek için farklı hareketlerinde birçok kere durmasını engellemektir. Gerçek hayat uygulamalarına ait gözlemler, bu yaklaşımın çekici tren operatörleri için büyük bir uygulama kolaylığı sağladığını ve süreç verimliliğini artırdığını göstermektedir.

Denklem (4.11) her bir görevin mutlaka ve sadece bir çekici trene atanması gerektiğini ifade etmektedir. Denklem (4.12) ve (4.13) çekici trenlerin kendisine atanan görevlerin sırasıyla bırakma ve alma koridorlarına hizmet vermesini garanti etmektedir. Denklem (4.14) araçların hizmet vermediği koridorlar için hizmet verme değişkenini sınırlandırmaktadır. Daha açık bir ifadeyle, her bir çekici trenin kendisine atanan görevlerin gerektirdiği koridorlar haricindeki koridorlar için hizmet değişkeninin 0

değerini almasını garanti etmektedir. Denklem (4.15) çekici trenlerin kapasite kontrolünü, Denklem (4.16) ise çevrim süresi kontrolünü sağlamaktadır. Çekici trenler için yolda geçen süre, alma ve bırakma işlemleri sırasında geçen süre ve varsa bekleme süresi toplamı periyotlar için belirlenen sabit çevrim süresinden fazla olmamalıdır.

Denklem (4.17) ve (4.19) sırasıyla her bir koridorda araç başına toplam bırakılan ve alınan kutu miktarını hesaplayan kısıtları içermektedir. Denklem (4.18) ve (4.20) ise bir çekici tren tarafından hizmet verilmeyen koridorlar için sırasıyla bırakılan ve alınan kutu miktarını ifade eden r ve l değişkenlerinin 0 değerini almasını garanti etmektedir. Çekici trenlerin her bir koridorda geçirdikleri süreyi hesaplamak için alınan ve bırakılan kutu miktarlarının tam olarak bilinmesi gerekmektedir. Dört denklemden oluşan bu kısıt kümesi bu nedenle modele dahil edilmiştir. Ancak Denklem (4.17) ve (4.19)'da verilen kısıtlarda p ve s değişkenlerinin birbirleriyle çarpılması söz konusudur. Bu durum modelin doğrusallığını bozmaktadır. Doğrusal olmayan modellerin çözümü doğrusal modellere göre çok daha zor olduğu için bu kısıtlar büyük M kullanılarak doğrusallaştırılmışlardır.

$$r^{kt} \geq \sum_{e \in E} (p_e^k q_e BK_{ij}^e) - \left(1 - s_{ij}^{kt}\right)M\,, \quad \forall (i,j) \in \beta,\ \forall k \in K, \forall t \in T \tag{4.21}$$

$$r^{kt} \leq \sum_{e \in E} (p_e^k q_e BK_{ij}^e) + \left(1 - s_{ij}^{kt}\right)M\,, \quad \forall (i,j) \in \beta,\ \forall k \in K, \forall t \in T \tag{4.22}$$

$$l^{kt} \geq \sum_{e \in E} (p_e^k q_e AK_{ij}^e) - \left(1 - s_{ij}^{kt}\right)M\,, \quad \forall (i,j) \in \theta,\ \forall k \in K, \forall t \in T \tag{4.23}$$

$$l^{kt} \leq \sum_{e \in E} (p_e^k q_e AK_{ij}^e) + \left(1 - s_{ij}^{kt}\right)M\,, \quad \forall (i,j) \in \theta,\ \forall k \in K, \forall t \in T \tag{4.24}$$

Denklem (4.17)'de verilen kısıt, Denklem (4.21) ve (4.22)'de verilen iki kısıt denklemi ile, Denklem (4.19)'da verilen kısıt ise Denklem (4.23) ve (4.24)'te verilen denklemlerle doğrusal şekilde ifade edilmektedir. Görüldüğü gibi doğrusal denklemlerde değişkenlerin birbirleri ile çarpımı söz konusu değildir ve sadece s değişkeni 1 değerini aldığında denklemler geçerli olmaktadır.

Zaman hesaplama kısıtları:

$$z^{kt} = \left[d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt}\right]x_{ij}^{kt}, \quad \forall(i,j) \in A, i = 1, \forall j \in N - \{1\},$$
$$t = 1, \forall k \in K \tag{4.25}$$

$$z^{kt} = \left[z^{k(t-1)} + d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt}\right]x_{ij}^{kt}, \quad \forall(i,j) \in A,$$
$$\forall t \in T - \{1\}, \forall k \in K \tag{4.26}$$

$$z^{kt} \leq M \sum_{(i,j)\in A} x_{ij}^{kt}, \qquad \forall k \in K, \forall t \in T \tag{4.27}$$

$$w^{kt} \leq M \sum_{(i,j)\in A} x_{ij}^{kt}, \qquad \forall k \in K, \forall t \in T \tag{4.28}$$

Denklem (4.25) ve (4.26), çakışmasız bir rotalama ve çizelgeleme yapabilmek için her bir çekici trenin köşe noktalarından geçiş zamanlarını hesaplamak amacıyla oluşturulan kısıtlardır. Bunlardan ilki her aracın sadece ilk hareketi sonunda köşe noktasından geçiş zamanını hesaplamaktadır. İkincisi ise ikinci ve diğer hareketler sonunda köşe noktalarından geçiş zamanlarını belirlemektedir. Denklem (4.27) ve (4.28) sırasıyla geçiş zamanı ve bekleme süresi değişkenlerini sınırlandırmaktadırlar. Denklem (4.25) ve (4.26)'da verilen kısıtlar açıkça görüleceği üzere içerisinde x değişkeni ile z, r, l ve w değişkenlerinin çarpımını içerdiği için doğrusallığı bozmaktadırlar. Bu nedenle bu kısıtlar da büyük M kullanılarak aşağıda verildiği şekilde doğrusal kısıt denklemlerine dönüştürülmüşlerdir.

$$z^{kt} \geq d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt} - \left(1 - x_{ij}^{kt}\right)M, \quad \forall(i,j) \in A, i = 1,$$
$$\forall j \in N - \{1\}, t = 1, \forall k \in K \tag{4.29}$$

$$z^{kt} \leq d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt} + \left(1 - x_{ij}^{kt}\right)M, \quad \forall(i,j) \in A, i = 1,$$
$$\forall j \in N - \{1\}, t = 1, \forall k \in K \tag{4.30}$$

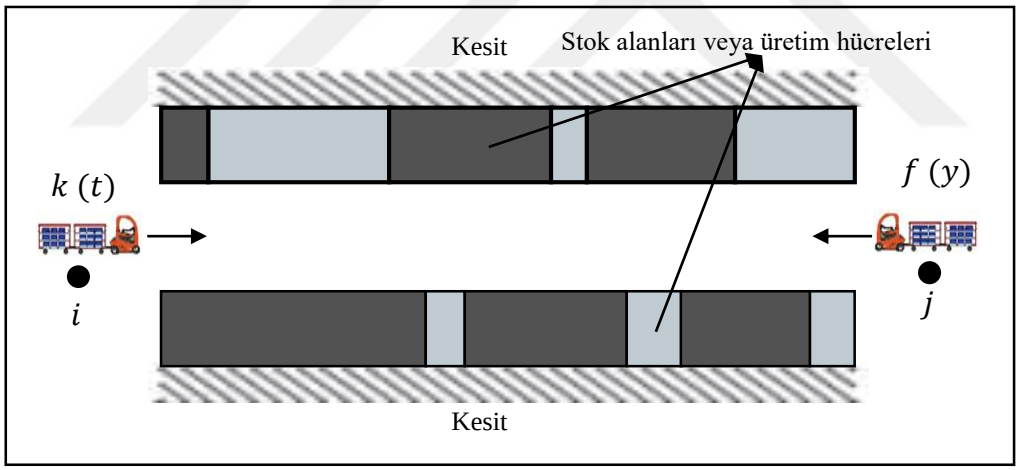
$$z^{kt} \geq z^{k(t-1)} + d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt} - \left(1 - x_{ij}^{kt}\right)M, \quad \forall(i,j) \in A,$$
$$\forall t \in T - \{1\}, \forall k \in K \tag{4.31}$$

$$z^{kt} \leq z^{k(t-1)} + d_{ij} + r^{kt}BBS + l^{kt}ABS + w^{kt} + \left(1 - x_{ij}^{kt}\right)M, \quad \forall(i,j) \in A,$$
$$\forall t \in T - \{1\}, \forall k \in K \tag{4.32}$$

Denklem (4.29) ve (4.30), Denklem (4.25)'teki zaman hesaplama kısıtının doğrusallaştırılmış halidir. Görüldüğü gibi bu denklem çifti, x değişkeni 1 değeri aldığında Denklem (4.25) ile verilen kısıta eşit olmaktadır. x değişkeninin 0 değerini alması halinde ise ikisi de geçersiz olmaktadır. Denklem (4.31) ve (4.32) ise Denklem (4.26)'nın doğrusal bir şekilde ifade edilmiş şeklidir. Aynı şekilde bu denklem çifti de x değişkeninin aldığı değere göre ya Denklem (4.26)'ya eşit ya da geçersiz olmaktadır.

Çakışma engelleme kısıtları;

Bu modelde çekici trenler için karşılıklı ve arka arkaya olmak üzere iki farklı çakışma durumu incelenmiştir. Bunlardan ilki dar bir koridorda araçların karşı karşıya gelmesi durumudur. Örneğin Şekil 4.2'de gösterildiği gibi k indisi ile temsil edilen trenin t hareketinde dar olan koridordan (i, j) yönüne, f indisi ile temsil edilen trenin ise y hareketinde (j, i) yönüne doğru geçmesi durumunda trenlerin karşılıklı olarak birbirlerini bloklaması ve trafik oluşturması söz konusu olabilir.



Şekil 4.2. Dar bir yolda iki çekici trenin karşı karşıya gelmesi durumu

Böyle bir senaryoda çakışma oluşmaması için şu iki durumdan birinin gerçekleşmesi gerekmektedir.

1. f aracı j noktasına, k aracı (i, j) koridorunu terk ettikten sonra vardığı, diğer bir değişle k aracı (i, j) koridorunu, f aracı j noktasına varmadan önce terk ettiği durum,
2. k aracı i noktasına, f aracı (j, i) koridorunu terk ettikten sonra vardığı, diğer bir değişle f aracı (j, i) koridorunu, k aracı i noktasına varmadan önce terk ettiği durum.

Bu iki durum sırasıyla şu eşitsizliklerle ifade edilebilir;

$$z^{k(t-1)} > z^{fy} \tag{4.33}$$

$$z^{kt} < z^{f(y-1)} \tag{4.34}$$

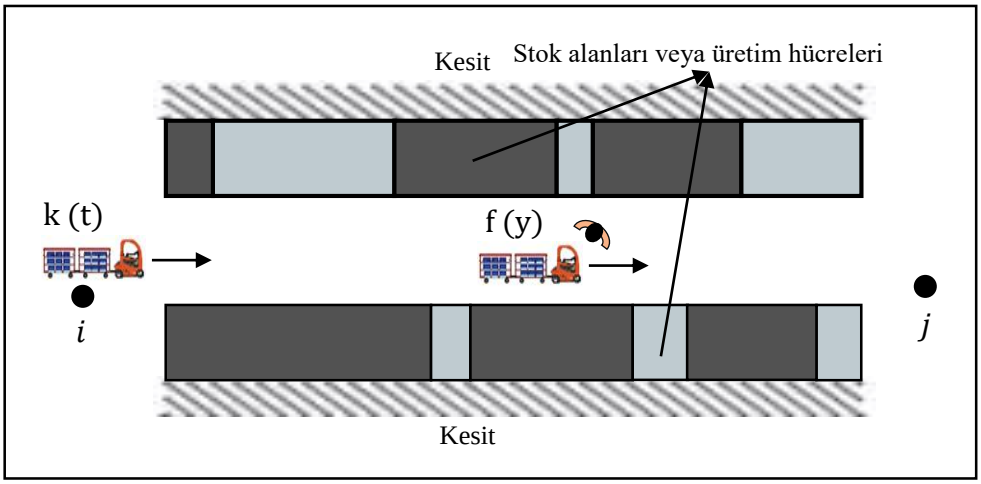
Çakışma durumun gerçekleşmesi için gerekli olan $x_{ij}^{kt} = 1$ ve $x_{ji}^{fy} = 1$ koşulları için denklemlerin sağ taraflarına sırasıyla $+M\left(x_{ij}^{kt} - 1\right) + M\left(x_{ji}^{fy} - 1\right)$ ve $-M\left(x_{ij}^{kt} - 1\right) - M\left(x_{ji}^{fy} - 1\right)$ ifadeleri eklenmelidir. Çünkü sadece bu koşullar gerçekleştiğinde çekici trenlerin dar bir koridorda karşılıklı olarak birbirini bloklamaları söz konusu olabilir. Ayrıca çakışma olmaması için yukarıda belirtilen iki durumdan sadece birinin sağlanması gerekmektedir. Bunun için kısıt denklemlerine sırasıyla $-M * (u_{ij}^{ktfy})$ ve $M * (1 - u_{ij}^{ktfy})$ ifadeleri eklenmelidir. Bu durumda karşılıklı çakışma engelleme kısıtları aşağıda verildiği gibi elde edilir;

$$z^{k(t-1)} > z^{fy} + M\left(x_{ij}^{kt} - 1\right) + M\left(x_{ji}^{fy} - 1\right) - M\left(u_{ij}^{ktfy}\right), \quad \forall (i,j) \in DY,$$
$$\forall t, y \in T - \{1\}, \forall k, f \in K, f > k \tag{4.35}$$

$$z^{kt} < z^{f(y-1)} - M\left(x_{ij}^{kt} - 1\right) - M\left(x_{ji}^{fy} - 1\right) + M\left(1 - u_{ij}^{ktfy}\right), \quad \forall (i,j) \in DY,$$
$$\forall t, y \in T - \{1\}, \forall k, f \in K, f > k \tag{4.36}$$

u_{ij}^{ktfy} ikili değişkeninin alacağı 0 veya 1 değerine göre bu kısıt çiftinden sadece birinin geçerli olacağı diğerinin geçersiz olacağı açıktır. Böylece dar koridorlarda çekici trenlerin karşılıklı olarak birbirini bloklamaları engellenecektir.

İkinci durum ise dar bir koridorda bir aracın, alma ya da bırakma yapmak için durmuş olan diğer bir araç tarafından bloklanmasıdır. Örneğin Şekil 4.3'te gösterildiği gibi dar bir koridorda f indisi ile temsil edilen çekici trenin y hareketinde (i,j) yönüne giderken kutu alma veya bırakma işlemi yapmak için koridorda durduğu sırada, k indisi ile temsil edilen diğer bir çekici trenin t hareketinde (i,j) yönünde koridora girmesi ve f aracının işlem süresinin bitmesini beklemek zorunda kalması ile arka arkaya çakışma durumu ortaya çıkacaktır.



Şekil 4.3. Dar bir yolda iki çekici trenin arka arkaya gelmesi durumu

Böyle bir senaryoda çakışma durumunun oluşmaması için şu iki durumdan bir tanesinin gerçekleşmesi gerekmektedir;

1. f treninin (i,j) koridoruna k treninden önce girip, k treninden önce çıktığı durum,
2. k treninin (i,j) koridoruna f treninden önce girip, f treninden önce çıktığı durum.

Bu durumlardan ilki şu iki eşitsizlik ile ifade edilebilir;

$$z^{k(t-1)} \leq z^{f(y-1)} \tag{4.37}$$

$$z^{kt} \leq z^{fy} \tag{4.38}$$

İkinci durum ise aşağıdaki iki eşitsizlik ile ifade edilebilir;

$$z^{f(y-1)} < z^{k(t-1)} \tag{4.39}$$

$$z^{fy} < z^{kt} \tag{4.40}$$

Bu tip bir çakışmanın olabilmesi için dar koridora aynı yönden giren iki çekici trenden en az bir tanesinin mutlaka kutu alma ya da bırakma işlemi yapıyor olması yani o koridorda hizmet veriyor olması gerekmektedir. Aksi takdirde tren hızları aynı olduğu için trenlerin birbirini bekletmesi söz konusu olmayacaktır. Bu nedenle yukarıda verilen denklemlerin sağ taraflarına $s_{ij}^{kt} = 1$ ve $x_{ij}^{fy} = 1$ koşulları için $+M\left(1 - s_{ij}^{kt}\right) + M\left(1 - x_{ij}^{fy}\right)$ ifadesi eklenmelidir. Ayrıca yukarıda belirtilen iki durumdan sadece bir tanesinin sağlanması

gerektiği için kısıtların sağ taraflarına sırasıyla $+M\left(h_{ij}^{ktfy}\right)$ ve $+M\left(1-h_{ij}^{ktfy}\right)$ ifadeleri eklenmelidir. Bu durumda arka arkaya çakışma kısıtları nihai olarak şu şekilde elde edilir;

$$z^{k(t-1)} \leq z^{f(y-1)} + M\left(1-s_{ij}^{kt}\right) + M\left(1-x_{ij}^{fy}\right) + M\left(h_{ij}^{ktfy}\right), \quad \forall(i,j) \in DY \cap (\beta \cup \theta), \forall t,y \in T-\{1\}, \forall k,f \in K, f \neq k \tag{4.41}$$

$$z^{kt} \leq z^{fy} + M\left(1-s_{ij}^{kt}\right) + M\left(1-x_{ij}^{fy}\right) + M\left(h_{ij}^{ktfy}\right), \quad \forall(i,j) \in DY \cap (\beta \cup \theta), \forall t,y \in T-\{1\}, \forall k,f \in K, f \neq k \tag{4.42}$$

$$z^{f(y-1)} < z^{k(t-1)} + M\left(1-s_{ij}^{kt}\right) + M\left(1-x_{ij}^{fy}\right) + M\left(1-h_{ij}^{ktfy}\right), \quad \forall(i,j) \in DY \cap (\beta \cup \theta), \forall t,y \in T-\{1\}, \forall k,f \in K, f \neq k \tag{4.43}$$

$$z^{fy} < z^{kt} + M\left(1-s_{ij}^{kt}\right) + M\left(1-x_{ij}^{fy}\right) + M\left(1-h_{ij}^{ktfy}\right), \quad \forall(i,j) \in DY \cap (\beta \cup \theta), \forall t,y \in T-\{1\}, \forall k,f \in K, f \neq k \tag{4.44}$$

h_{ij}^{ktfy} ikili değişkenlerinin alacağı 0 veya 1 değerine göre Denklem (4.41, 4.42) veya Denklem (4.43, 4.44) geçerli olacaktır. Böylelikle dar koridorlarda çekici trenlerin alma ve bırakma sırasında birbirini bloklamaları engellenecektir.

Son olarak değişkenlerin sınırlandırılması ile ilgili kısıtlar da Denklem (4.45)'te verilmektedir.

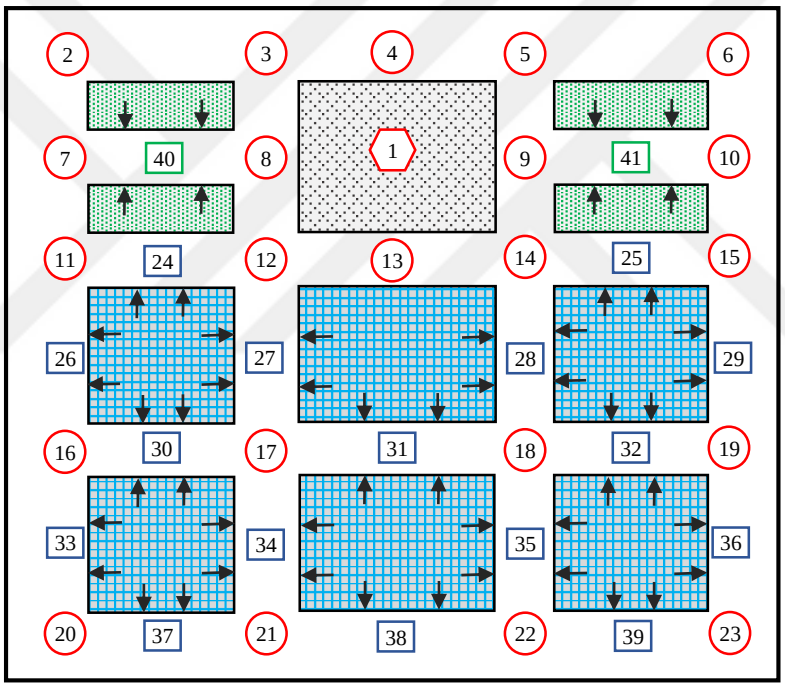
$$x_{ij}^{kt} \in \{0,1\}, s_{ij}^{kt} \in \{0,1\}, p_e^k \in \{0,1\}, r^{kt} \geq 0, l^{kt} \geq 0, w_{kt} \geq 0, z^{kt} \geq 0, h_{ij}^{ktfy} \in \{0,1\}, u_{ij}^{ktfy} \in \{0,1\} \tag{4.45}$$

4.2. İkinci Çözüm Yaklaşımı: İteratif Çözüm

SÇDRÇ probleminin çözümü için bir önceki bölümde geliştirilen Model-1, çok fazla ikili değişken içermesi ve problemin NP-zor doğası nedeniyle gerçek hayat problemlerine göre oldukça küçük boyutlu problemlerde bile optimum çözümü bulmak için kullanışsız

kalmaktadır. Örneğin Şekil 4.4'te verilen, 6 adet üretim hücresi ve 2 adet süpermarket alanına sahip küçük boyutlu bir kroki kullanılarak üretilen problemlerin çoğunda dahi optimum çözüm Model-1 ile ya çok uzun sürelerde elde edilmekte ya da hiç bulunamamaktadır.

Özellikle artan çekici tren sayısı, görev sayısı ve dar yol oranı gibi parametrelere bağlı olarak böyle küçük bir kroki üzerinde üretilen problemlere bile çözüm bulunamaması, kesin olmayan algoritmalara yönelmeden önce, en azından küçük ve orta boyutlu sayılabilecek krokiler ile üretilen problemlerin optimum çözümünü elde edebilecek kesin bir çözüm prosedürü arayışına sebep olmuştur.



Şekil 4.4. Küçük boyutlu örnek bir kroki

Bu nedenle bu çalışmada sezgisel, meta-sezgisel veya mat-sezgisel algoritmalara yönelmeden önce en azından belli bir boyuta kadar olan problemleri makul sürelerde optimum değeri bularak çözebilecek bir yaklaşım geliştirilmesi hedeflenmiştir. Model-1'in Bölüm 5.1'de açıklanan farklı büyüklükteki test problemlerinin çözümü için gerektirdiği değişken sayıları türlerine göre gruplandırılarak Tablo 4.2'de verilmiştir.

Tablo 4.2. Çeşitli problemler için değişken sayıları dağılımları

Problem kodu	Parametreler				Değişken sayıları		
	Kroki boyutu	Araç sayısı	Görev sayısı	Dar koridor oranı	İkili değişkenler		Genel değişkenler
					x, s ve p değişkenleri toplamı	h ve u değişkenleri toplamı	z, w, r ve l değişkenleri toplamı
K_2550_3	Küçük	3	50	25%	5.964	46.656	228
K_2550_4	Küçük	4	50	25%	7.544	83.232	288
K_5050_3	Küçük	3	50	50%	5.964	93.312	228
K_5050_4	Küçük	4	50	50%	7.544	166.464	288
OD_25100_5	Orta dikdörtgen	5	100	25%	15.500	259.920	400
OK_50100_4	Orta kare	4	100	50%	14.216	381.024	352
OK_50100_5	Orta kare	5	100	50%	16.985	576.000	420
BD_50200_7	Büyük dikdörtgen	7	200	50%	82.985	7.865.424	980

Tablo incelendiğinde açıkça görülmektedir ki, ikili değişkenlerin çok büyük bir kısmı trafik engelleme kısıtları için kullanılan h ve u değişkenlerinden oluşmaktadır. Dolayısıyla buradan yola çıkarak, Model-1, trafik engelleme kısıtlarını kaldırarak gevşetildiğinde çözülmesinin çok daha hızlı olacağı sonucuna varılabilir. Bu noktadan hareketle bu bölümde, Model-1'in zaman hesaplama ve trafik engelleme ile ilgili değişkenlerinin ve kısıtlarının kaldırılarak gevşetilmesiyle Model-2 oluşturulmuştur.

Küçük boyutlu problemlerde Model-2 ile çakışmalar göz ardı edildiği durumda makul sürelerde optimum çözüm elde edilse de, orta boyutlu problemlerde çözümün optimalliğini kanıtlamak oldukça uzun sürmekte yada hiç kanıtlanamamaktadır. Bu nedenle çakışma durumlarının göz ardı edildiği durumda optimum çözüm değerini orta boyutlu problemlerde de makul sürelerde elde edebilmek için Model-3 adında farklı bir karma tam sayılı doğrusal programlama modeli daha geliştirilmiştir. Bu bölümde Model-2 ve Model-3 ayrı ayrı başlıklar altında sunulduktan sonra iteratif çözüm yaklaşımı akış şeması verilerek detaylı bir şekilde anlatılacaktır.

4.2.1. Model-2

Model-2, Çakışmaları dikkate almayan ancak çekici trenler için adım adım hangi koridordan hangi sırayla gidileceğine ve hangi adımlarda hangi koridorlara alma bırakma

yapılacağına karar veren bir yükleme, rotalama ve çizelgeleme modelidir. Kısaca Model-1'in çakışma engelleme için gerekli olan kısıtların ve değişkenlerin çıkarılarak gevşetilmiş halidir. Dolayısıyla birinci modelden farklı olarak r, l, z, w, h, u değişkenlerini ve bunların hesaplanması ile ilgili kısıtları içermemektedir.

Model-2'nin amaç fonksiyonu Denklem (4.46)'da verilmektedir. Görüldüğü gibi bu amaç fonksiyonu sadece çekici trenlerin yolda geçirdikleri toplam sürenin, yani toplam yolun minimizasyonunu içermektedir. Bu modelde çakışmalar göz ardı edildiği için amaç fonksiyonu, Model-1'den farklı olarak bekleme sürelerini içermemektedir.

$$minZ = \sum_{(i,j)\in A} \sum_{k\in K} \sum_{t\in T} x_{ij}^{kt} d_{ij} \nu \tag{4.46}$$

Bölüm 4.1'de verilen Model-1'in kısıtlarından Denklem (4.2 - 4.15)'te verilen 14 adet kısıt, bu modelde de aynı şekilde kısıtları oluşturmaktadır. Bunlara ek olarak çevrim süresi kısıtı (Denklem (4.47)) ve değişken sınırlama kısıtları (Denklem (4.48)) aşağıda verildiği gibi revize edilmiştir;

$$\sum_{t\in T} \sum_{(i,j)\in A} x_{ij}^{kt} d_{ij} + \sum_{e\in E} p_e^k \, q_e (BBS + ABS) \leq CS, \qquad \forall k \in K \tag{4.47}$$

$$x_{ij}^{kt} \in \{0,1\}, s_{ij}^{kt} \in \{0,1\}, p_e^k \in \{0,1\} \tag{4.48}$$

4.2.2. Model-3

Burada sunulan Model-3 çekici trenlerin karşılıklı ve arka arkaya çakışmalarını dikkate almayan, adım adım hangi koridorlar hangi sırayla gideceklerini ve hangi hareketleri sırasında alma-bırakma işlemi yapacaklarını vermeyen ancak Model-2 ile aynı optimum sonucu çok daha kısa sürede elde edebilen bir modeldir. Bunun temel nedeni çok daha az tam sayılı değişken ve kısıt içermesi nedeniyle dal-kesme sürecinin kısalmasıdır. Bu modelin tanımlanmasının ve kullanılmasının temel amacı Model-2 için bir alt sınır belirlemektir. Böylelikle Model-2 ile çözüm optimalliğinin kanıtlanması için çok uzun süreler uğraşma gerekliliği ortadan kalkacaktır.

Model-1'de verilen kümeler, indisler, parametreler ve değişkenler dışında Model-3'te kullanılan ek notasyonlar aşağıda verilmektedir.

- V_{ij} : (i,j) koridorunda gidildikten sonra u dönüşü yapmadan gidilebilecek komşu köşe noktaları kümesi, $(i,j) \in A, V_{ij} \subset N$
- $x_{ij}^k \in \{0\} \cup \mathbb{Z}^+$: k aracının (i,j) koridorunda $i-j$ yönünde kaç kere gittiğini ifade eden tam sayılı karar değişkeni

Tam sayılı doğrusal bir programlama modeli olan Model-3 aşağıda sunulmaktadır.

$$minZ = \sum_{(i,j)\in A} \sum_{k\in K} x_{ij}^k d_{ij}\, v \tag{4.49}$$

Aşağıdaki kısıtlar altında:

$$\sum_{\substack{(i,j)\in A \\ i=1}} x_{ij}^k M \geq \sum_{\substack{(i,j)\in A \\ i\neq 1}} x_{ij}^k, \qquad \forall k \in K \tag{4.50}$$

$$\sum_{\substack{(i,j)\in A \\ i=1}} x_{ij}^k \leq 1, \qquad \forall k \in K \tag{4.51}$$

$$\sum_{i\in N} x_{ij}^k - \sum_{i\in N} x_{ji}^k = 0, \qquad \forall k \in K, \forall j \in N, (i,j) \in A \tag{4.52}$$

$$x_{[1]j}^k + x_{j[1]}^k \leq 1, \qquad \forall k \in K, \forall j \in N - \{1\} \tag{4.53}$$

$$x_{ij}^k - \sum_{v\in V_{ij}} x_{jv}^k \leq 0, \qquad \forall k \in K \tag{4.54}$$

$$\sum_{k\in K} p_e^k = 1, \qquad \forall e \in E \tag{4.55}$$

$$x_{PBA_e\, PBU_e}^k + x_{PBU_e PBA_e}^k \geq p_e^k, \qquad \forall e \in E, \forall k \in K \tag{4.56}$$

$$x_{PAA_e\, PAU_e}^k + x_{PAU_e PAA_e}^k \geq p_e^k, \qquad \forall e \in E, \forall k \in K \tag{4.57}$$

$$\sum_{e \in E} p_e^k q_e \leq C, \qquad \forall k \in K \tag{4.58}$$

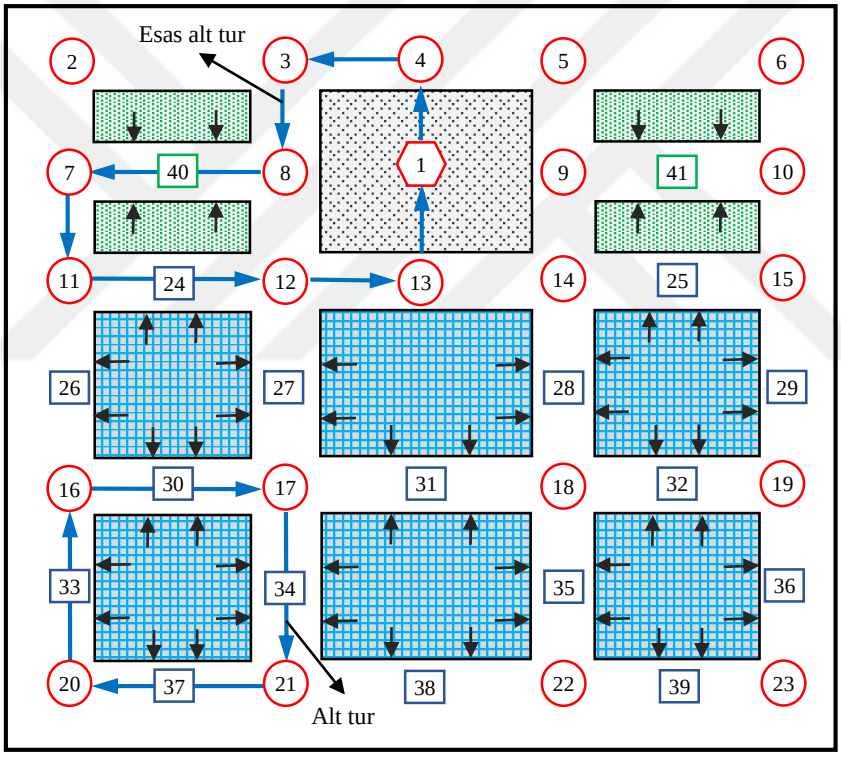
$$\sum_{(i,j) \in A} x_{ij}^k d_{ij} + \sum_{e \in E} p_e^k \, q_e (BBS + ABS) \leq CS, \qquad \forall k \in K \tag{4.59}$$

$$x_{ij}^{kt} \in \{0\} \cup \mathbb{Z}^+, p_e^k \in \{0,1\} \tag{4.60}$$

Modelin amaç fonksiyonu olan Denklem (4.49) bütün çekici trenler tarafından gidilen mesafenin toplamının minimizasyonunu ifade etmektedir. Denklem (4.50) çekici trenlerin eğer fabrika içinde herhangi bir koridordan geçeceklerse mutlaka başlangıç noktasından çıkmalarını garanti etmektedir. Bu kısıt sayesinde ihtiyaç duyulmayan trenlerin başlangıç noktasından hareket etmeme seçeneği bulunmaktadır. Yani sadece kendisine görev atanan ve bu görevlere göre belirli koridorlara gitmesi gereken trenler başlangıç noktasından çıkmak zorundadır. Denklem (4.51) ise başlangıç noktasından hareket eden bir çekici trenin tek bir yöne ve bir kere hareket ederek çıkmasını sağlamaktadır. Denklem (4.52) her bir tren için her köşe noktasına geliş ve gidişlerin toplamlarının eşit olmasını sağlamaktadır. Böylece dengeli bir akış sağlanmış olacaktır. Bu kısıt aynı zamanda başlangıç noktasından çıkan trenlerin geri başlangıç noktasına mutlaka giriş yapmasını da gerektirdiği için trenlerin dönüşünü garanti etmektedir.

Denklem (4.53) çekici trenlerin başlangıç noktasına çıkış yaptığı yönden girmemesini garanti etmektedir. Bu kısıtın eklenmesinin nedeni trenlerin u dönüşü yapamamasıdır. Bir çekici tren eğer başlangıç noktasına çıkış yaptığı yönden girerse bir sonraki tura başlaması mümkün olamayacaktır. Denklem (4.54) ise bir trenin geçtiği koridordan u dönüşü yaparak geri dönemeyeceği için başka bir koridora yönelmesini garanti etmektedir. Denklem (4.55) her bir görevin mutlaka ve sadece bir araca atanmasını sağlamaktadır. Denklem (4.56) ve (4.57) ise bir çekici trenin kendisine atanan bir görevin sırasıyla alma ve bırakma koridorlarından herhangi bir yöne en az bir kere geçmesini garanti etmektedir. Denklem (4.58) ve (4.59) sırasıyla çekici trenler için kapasite ve çevrim süresi kısıtlarını içermektedir. Son olarak Denklem (4.60) değişkenler için tanım kümelerini ifade etmektedir.

Model-3, Model-2'ye göre oldukça hızlı bir şekilde çözülüyor olsa da elde edilen çözümler alt turlar içerebilmektedir. Örneğin Şekil 4.5'te verilen kroki üzerinde tek bir çekici trenle iki görevin yapılması gerektiği bir senaryo olsun. Bu görevlerden biri 40 nolu alma koridorundan 24 nolu bırakma koridoruna, diğeri de 40 nolu alma koridorundan 37 nolu bırakma koridoruna belli miktar kutu taşıma içersin. Bu durumda Model-3 ile hesaplanan rotalama Şekil 4.5'teki gibi olmaktadır. Burada görüldüğü üzere kısıtlar tam olarak sağlanmakta ve iki alt tur oluşmaktadır. Bunun temel nedeni görev koridorlarının birbirinden uzakta olması ve bu şekilde alt turların kısıtlarda engellenmiyor olmasıdır. Dolayısıyla geçerli bir rotalama planı oluşturmak için bu alt turların kırılması gerekmektedir.



Şekil 4.5. Örnek bir alt turlu rotalama

Bu çalışmada gezgin satıcı ve araç rotalama problemlerinde alt turların iteratif olarak elenmesi yaklaşımından esinlenilerek Model-3 için iteratif bir alt tur kırma mekanizması geliştirilmiştir. Bu mekanizma, her araç için alt turların; "asıl alt tur" ve "diğer alt turlar" olarak ayırılmasına ve esas alt turların her iterasyonda genişlemeye zorlanmasına dayanmaktadır.

Esas alt tur; her aracın başlangıç noktasından çıkan ve geri başlangıç noktasına dönen alt turudur. İterasyonlar boyunca ek kısıtlar eklenerek her araç için esas alt turun genişletilmesi ve böylelikle belli bir iterasyon sonra araç için var olan diğer alt turlarla bir şekilde birleşmesi hedeflenmektedir.

k adet çekici trenin olduğu bir probleme ilişkin alt tur eleme kısıtlarını oluşturmak için şu iki adet kümenin tanımlanması gerekmektedir.

- M^k : k treni için esas alt turun geçtiği köşe noktalarını içeren küme ($M^k \subset N$)
- $ALT^k = \{(i,j) \mid i \in M^k,\ j \notin M^k,\ (i,j) \in A\}$: k aracının esas alt turundaki köşe noktalarının esas alt tur dışındaki komşu köşe noktaları ile olan kenarlarını (koridorları) içeren küme ($ALT^k \subset A$)

Alt tur eleme kısıtı her bir çekici tren için şu şekilde yazılabilir;

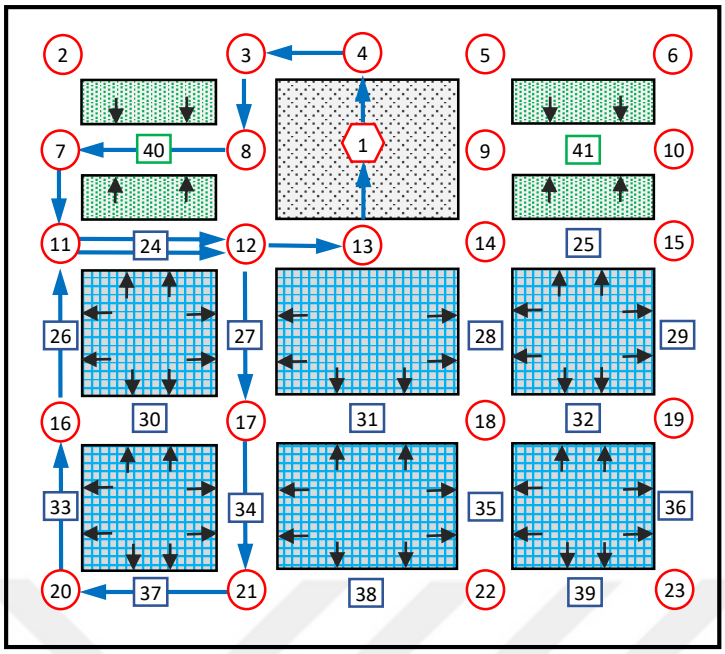
$$\sum_{(i,j)\in ALT^k} x_{ij}^k \geq 1 \qquad \forall k \in K \tag{4.61}$$

Bu kısıt denklemi, her bir aracın esas alt turunu, alt tur üzerindeki herhangi bir köşe noktasından ona komşu olan herhangi bir diğer köşe noktasına genişletmeyi sağlamaktadır. Örneğin Şekil 4.5'te verilen tek araçlı durum için alt tur kırma kısıtı şu şekilde olacaktır;

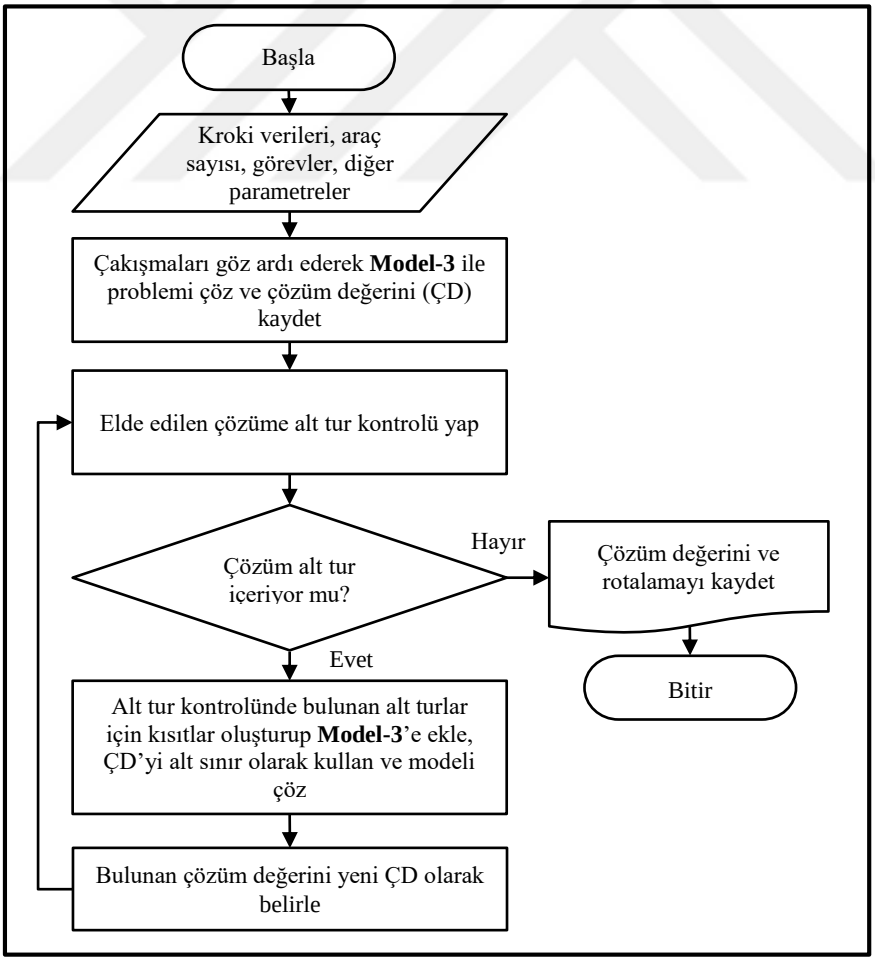
$$x_{4\,5}^1 + x_{3\,2}^1 + x_{8\,12}^1 + x_{7\,2}^1 + x_{11\,16}^1 + x_{12\,8}^1 + x_{12\,17}^1 + x_{13\,14}^1 \geq 1 \tag{4.62}$$

Bu şekilde kısıtlar iteratif olarak eklendikçe en sonunda Model-3, iki görev ve tek araç içeren bu örnek problem için rotalamayı Şekil 4.6'da gösterildiği gibi yapacaktır.

Burada anlatılan alt tur eleme süreci, akış şeması olarak Şekil 4.7'de gösterilmektedir. Şekilde görüldüğü gibi alt tursuz çözüm bulunana kadar her iterasyonda bir önceki iterasyonun optimum değeri alt sınır olarak kullanılmaktadır. Bu yaklaşım, çoğu iterasyonda çözüm optimalliğinin kanıtlanması sürecini ciddi oranda azaltarak alt tursuz çözüm bulma süresini kısaltmaktadır.



Şekil 4.6. Model-3 ile alt tursuz örnek bir rotalama



Şekil 4.7. Model-3 için alt tur eleme prosedürü

4.2.3. İteratif Çözüm yaklaşımı

Bu bölümde, belli bir büyüklüğe kadar olan SÇDRÇ problemlerini birinci çözüm yaklaşımına göre çok daha kısa sürede, optimum değer elde edilecek şekilde çözebilen iteratif bir çözüm yaklaşımı sunulacaktır. Bu çözüm prosedürü önceki bölümlerde tanımlanan üç matematiksel modele dayanmakta ve çeşitli aşamalarında bu modelleri kullanmaktadır. Bu nedenle modellerin daha iyi anlaşılabilmesi için öncelikle özellikleri ve aynı problem için gerektirdikleri değişken sayıları birbirleri ile kıyaslamalı olarak açıklanacaktır.

Tablo 4.3'te geliştirilen üç matematiksel modele ilişkin genel özellikler ve sundukları bilgiler verilmektedir. Tablo 4.4 ve Tablo 4.5'te ise sırasıyla Bölüm 5.1'de oluşturulan farklı büyüklükteki test problemleri için bu üç modelin gerektirdiği farklı tipteki değişken sayıları ve kısıt sayıları özetlemektedir. Model-1, SÇDRÇ problemi için çakışmaların engellendiği tam bir çözüm sunarken Model-2 çakışmaların dikkate alınmadığı yani dar yolların olmadığı varsayılan bir fabrika krokisi için optimum çözüm sunmaktadır. Her iki model de koridorlardan hangi yöne ve hangi sıra ile geçileceğini, hangi geçiş sırasında kutu alma veya bırakma işlemleri yapılacağını ayrıntılı şekilde veren bir çizelgeleme sunmaktadır. Aynı büyüklükteki problem için gerektirdikleri değişken sayılarına bakıldığında Model-1'in zaman hesaplama ve çakışma engelleme kısıtları nedeniyle içerdiği değişken sayısının (r, l, z, w, h, u değişkenleri) Model-2'ye göre çok daha fazla olduğu görülmektedir.

Tablo 4.3. Geliştirilen modellerin özelliklerinin karşılaştırılması

Özellikler	**Model 1**	**Model 2**	**Model 3**
Rotalama	✓	✓	✓
Yükleme	✓	✓	✓
Çizelgeleme	✓	✓	
Hangi koridordan toplamda kaç kere geçileceğini gösterme	✓	✓	✓
Adım adım gidilecek koridorları gösterme	✓	✓	
Hangi adımda hangi koridorda alma veya bırakma yapılacağını gösterme	✓	✓	
Çakışma engelleme	✓		
Çözüm hızı	Çok yavaş	Yavaş	Hızlı

Model-3 ise çakışmaların olmadığı varsayılan durum için optimum çözüm değerini Model-2'den çok daha hızlı bir şekilde bulabilmektedir. Bunun temel nedeni Tablo 4.4 ve Tablo 4.5'te görüldüğü gibi aynı problem için çok daha az tam sayılı değişken ve kısıt içermesidir. Ancak bunun bir ödünleşimi olarak, adım adım hangi koridorlardan hangi sırayla geçileceğini ve birden çok kez geçilen bir koridordan hangi geçiş sırasında kutu alma veya bırakma işlemi yapılacağı bilgilerini vermemektedir. Bu bilgiler çekici trenlerin çizelgelenmesi ve çakışıp çakışmadığının anlaşılabilmesi için gereklidir. Bu nedenle bu çalışmada Model-3, sadece alt sınır değerinin elde etmek ve çözüm prosedürünü hızlandırmak için kullanılmaktadır.

Tablo 4.4. Modellerin farklı test problemlerinin çözümü için gerektirdiği değişken sayıları

Problem kodu	**Model 1**			**Model 2**	**Model 3**	
	İkili değişkenler		**Genel değişkenler**	**İkili değişkenler**	**Tam sayılı değişkenler**	**İkili değişkenler**
	x, s ve p değişkenleri toplamı	**h ve u değişkenleri toplamı**	**z, w, r ve l değişkenleri toplamı**	**x, s ve p değişkenleri toplamı**	**x değişkenleri toplamı**	**p değişkenleri toplamı**
K_2550_3	5.964	46.656	228	5.964	198	150
K_2550_4	7.544	83.232	288	7.544	264	200
K_5050_3	5.964	93.312	228	5.964	198	150
K_5050_4	7.544	166.464	288	7.544	264	200
OD_25100_5	15.500	259.920	400	15.500	410	500
OK_50100_4	14.216	381.024	352	14.216	392	400
OK_50100_5	16.985	576.000	420	16.985	490	500
BD_50200_7	82.985	7.865.424	980	82.985	1533	1400

Tablo 4.5. Modellerin farklı test problemlerinin çözümü için gerektirdiği kısıt sayıları

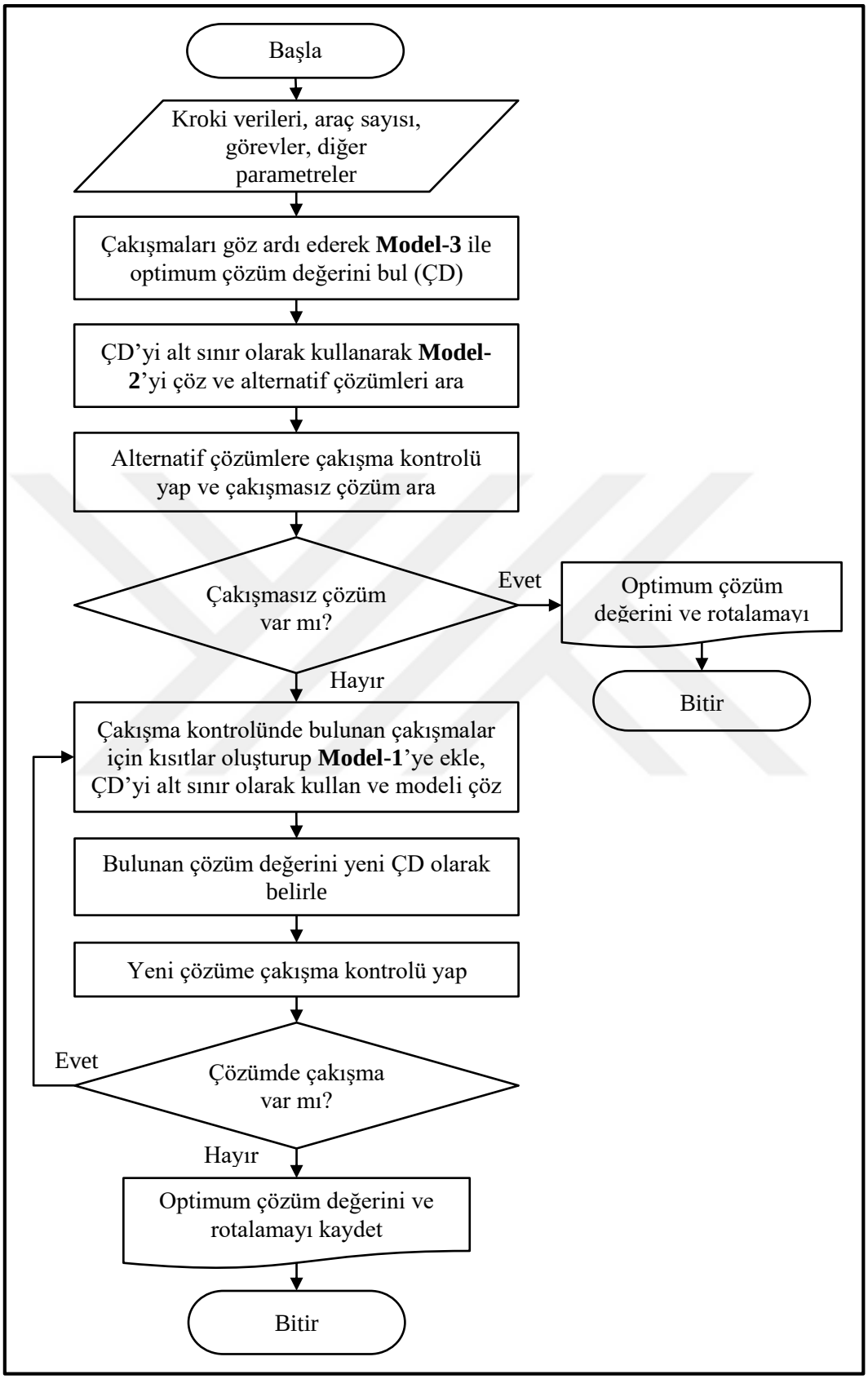
Problem kodu	**Kısıt sayıları**		
	Model 1	**Model 2**	**Model 3**
K_2550_3	175.331	8.238	680
K_2550_4	579.874	10.403	890
K_5050_3	330.851	8.238	680
K_5050_4	1.134.754	10.403	890
OD_25100_5	5.249.625	20.426	1825
OK_50100_4	2.587.384	20.097	1560
OK_50100_5	11.576.315	22.381	1925

Model-1'de zaman hesaplama ve çakışma kontrol kısıtlarının ve değişkenlerinin çok fazla olması ve bunların gevşetilerek daha hızlı çözüm elde edilebileceğinin düşünülmesi burada tanımlanan iteratif çözüm prosedürünün kilit noktasını oluşturmaktadır. Bütün dar koridorlar, bütün araçlar ve bütün hareketler için çakışma engelleme kısıtlarının yazıldığı Model-1, her ne kadar çakışmaların göz önünde bulundurulduğu durum için optimum çözümü garanti etse de çok fazla değişken ve kısıt içermektedir.

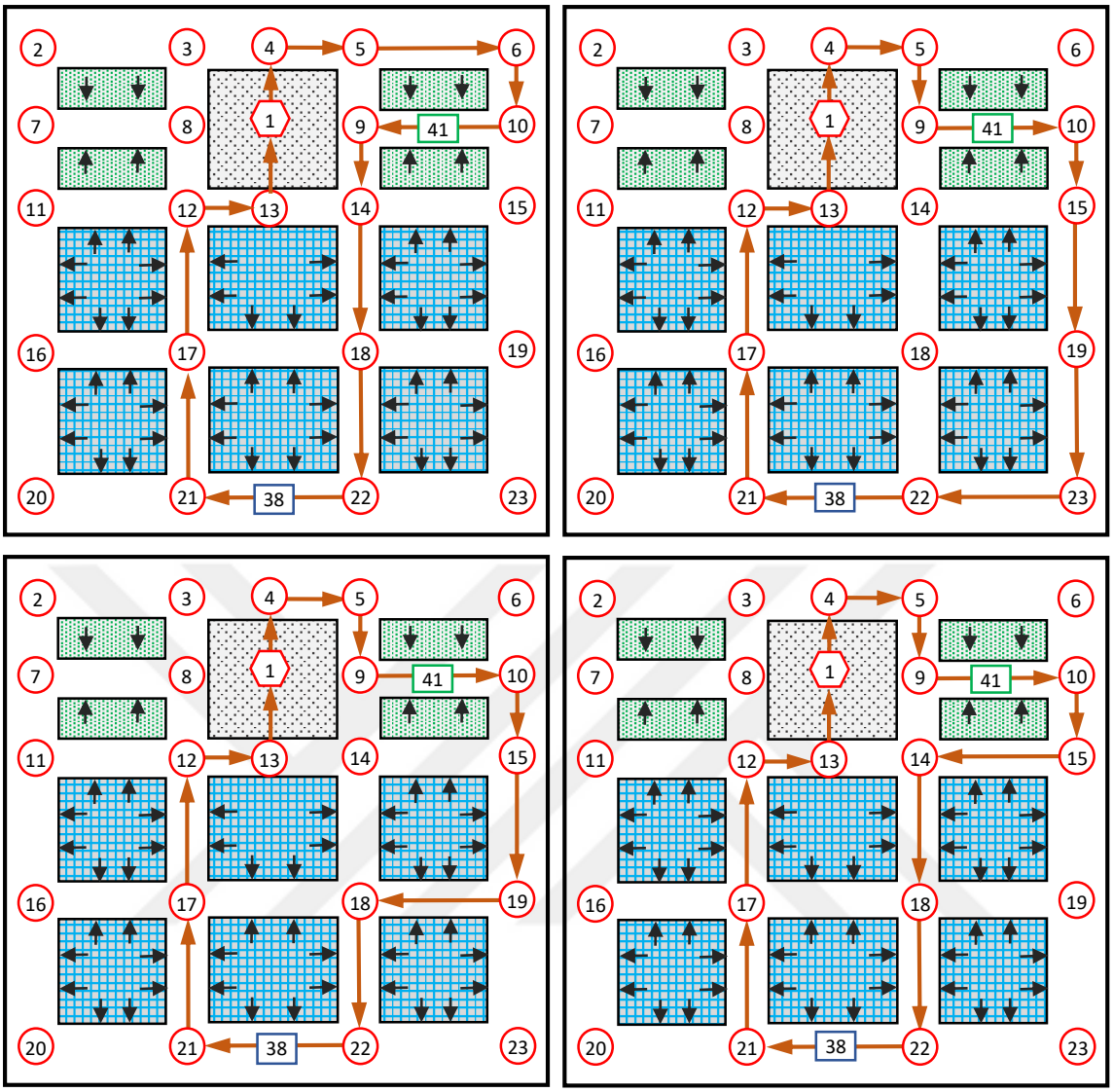
Diğer taraftan Model-2 ile elde edilecek bir çözümde sadece bazı koridorlarda bazı araçların belirli hareketlerinde çakışma olacaktır. Sadece çakışma olan bu durumlar için kısıt yazılarak Model-1'in çözülmesi ve elde edilen çözüme çakışma kontrolü yapılarak gerekirse bunun tekrarlanması süreci hızlandıracaktır. Bu düşünce ile geliştirilen iteratif çözüm yaklaşımının algoritması, akış şeması olarak Şekil 4.8'de verilmektedir.

Şekilde görüldüğü gibi öncelikle çakışmalar göz ardı edilerek problem Model-3 ile çözülmektedir. Yukarıda anlatıldığı şekilde Model-3; kapasite, çevrim süresi ve yol-denge kısıtları altında, görevlerin çekici trenlere optimum şekilde atanmasını ve trenlerin rotalanmasını içermektedir, ancak bir çizelgeleme planı sunmamaktadır. Bu nedenle burada kullanılmasının amacı sadece Model-2 için bir alt sınır belirlemektir. Böylelikle Model-2'nin çözümü sırasında çözüm optimalliğinin kanıtlanması için gereken çok uzun sürelerden tasarruf edilmektedir.

Alt sınırlı şekilde çok daha hızlı çözülen Model-2 ile belirli bir süre alternatif optimum çözümler aranmaktadır. Bu aşamada, hücresel imalat yerleşiminin bir sonucu olarak kroki üzerinde dikey ve yatay eksende paralel koridorların bulunması gibi bazı sebeplerle problemin büyüklüğüne bağlı olarak oldukça fazla sayıda alternatif optimum çözüm elde edilebilmektedir. Örneğin Şekil 4.9'da, 38 ve 41 nolu koridorlara hizmet vermesi gereken bir çekici tren için kısıtları sağlayan ve eşit yol uzunluğuna sahip dört farklı optimum rotalama seçeneği bulunmaktadır.



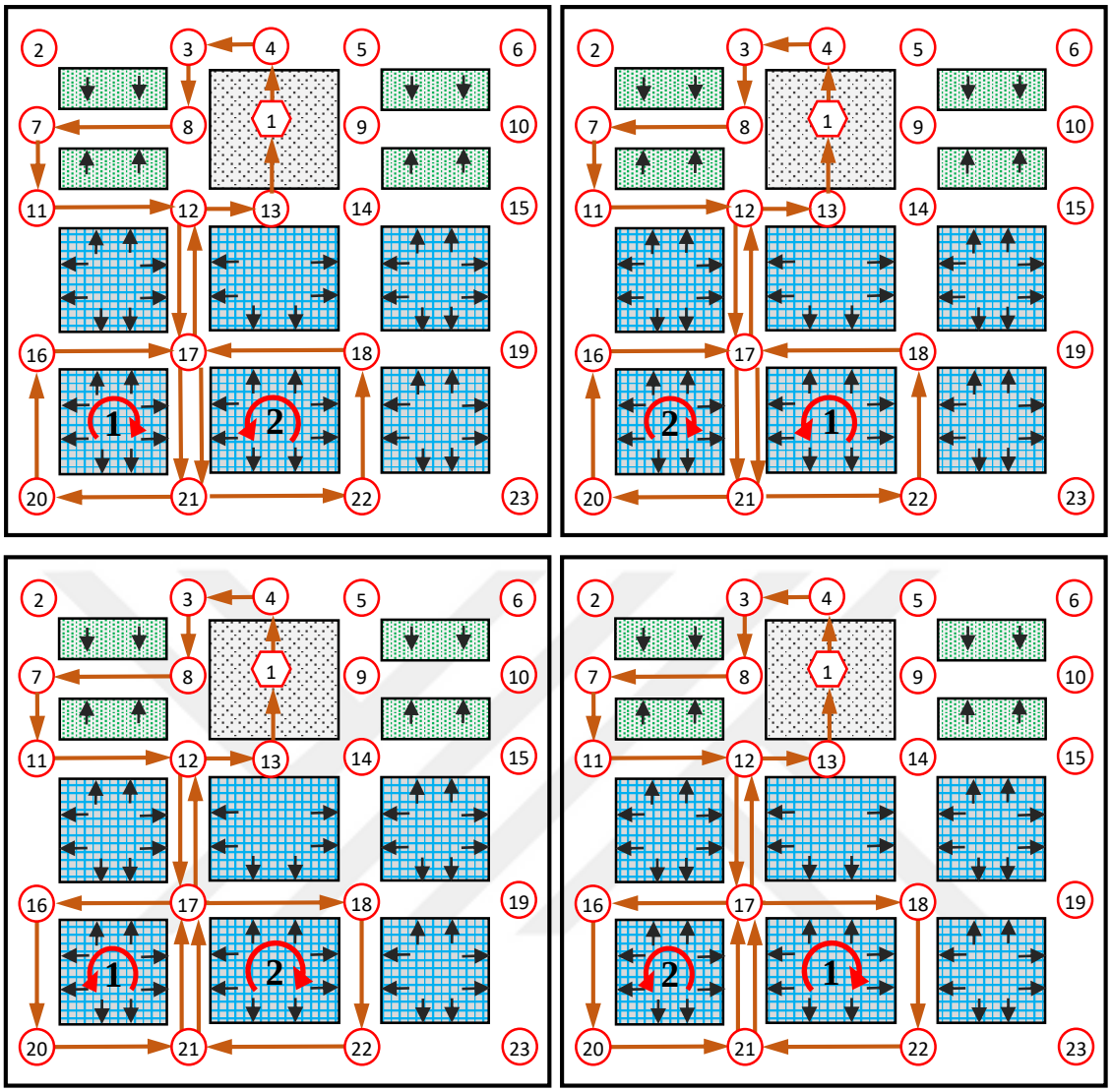
Şekil 4.8. İteratif çözüm yaklaşımının akış şeması



Şekil 4.9. Farklı koridorlardan gidilmesiyle oluşabilecek alternatif optimum rotalama örnekleri

Diğer bir örnek ise Şekil 4.10'da verilmektedir. Bir çekici tren aynı koridorlardan geçmesine rağmen sadece koridorların farklı sıralarda ve yönlerde gidilmesiyle dört farklı alternatif optimum rota elde etmek mümkündür. Bu rotalar sağ üstten sol alta doğru sırasıyla şu şekilde yazılabilir.

1. Rota: 1-4-3-8-7-11-12-17-21-20-16-17-21-22-18-17-12-13-1
2. Rota: 1-4-3-8-7-11-12-17-21-22-18-17-21-20-16-17-12-13-1
3. Rota: 1-4-3-8-7-11-12-17-16-20-21-17-18-22-21-17-12-13-1
4. Rota: 1-4-3-8-7-11-12-17-18-22-21-17-16-20-21-17-12-13-1



Şekil 4.10. Aynı koridorların farklı yönlerde veya sıralarda gidilmesiyle oluşabilecek alternatif optimum rota örnekleri

Alternatif optimum çözümlerin oluşmasının diğer bir nedeni ise birden çok kez geçilen bir koridora herhangi bir geçiş sırasında hizmet verilebilme seçeneğinin olmasıdır. Örneğin Şekil 4.10'daki rotalarda (12-17) ve (17-21) koridorlarından ikişer kere geçilmektedir. Her iki koridorda da kutu bırakma işlemi yapılacağını varsaydığımızda her bir koridorun kutu bırakma işlemi için ikişer hareket seçeneği olduğu için toplamda 4 farklı senaryo oluşmaktadır. Bu dört farklı senaryo Şekil 4.10'daki dört farklı rotanın hepsi içinde söz konusudur. Dolayısıyla tek bir çekici tren için 16 farklı senaryo elde edilmektedir. Şekildeki krokinin gerçek hayat problemine göre oldukça küçük bir kroki olduğu ve sadece tek rota üzerinden örneklerin incelendiği düşünüldüğünde fazlaca alternatif optimum çözüm elde edilebileceği rahatlıkla ifade edilebilir. Dolayısıyla elde

edilen bu çok sayıdaki alternatif optimum çözümlerden bazıları dar koridorlarda çekici trenlerin çakışmasını içeriyorken bazılarının tamamen çakışmasız olabileceği ifade edilebilir.

Çözüm yaklaşımının sonraki basamağında Model-2 ile elde edilen alternatif optimum çözümlere çakışma kontrolü yapılarak çakışmasız bir çizelgeleme planı olup olmadığının araştırılması gelmektedir. Eğer en azından bir tane çakışmasız çözüm bulunabilirse Model-1'in çözülmesine gerek kalmadan çakışmaların dikkate alındığı durum için de optimum bir rotalama ve çizelgeleme planı elde edilmiş olacaktır. Böylelikle çözüm kaydedilerek algoritma sonlandırılabilir. Çakışmasız çözüm bulunamadığı durumda ise Şekil 4.8'de görüldüğü gibi sadece çakışma kontrolünde bulunan çakışmalar için kısıtlar oluşturularak Model-1 çözülecektir.

```
Girdiler: Alternatif optimum çözümler
Çıktılar: Çakışmaların koridor ve hareket bilgileri
ACAK = ∅   // Arka arkaya çakışmaların olduğu koridorları içeren küme
ACAKT = ∅  // Arka arkaya çakışmaların olduğu hareket ikililerini içeren küme
KCAK = ∅   // Karşılıklı çakışmaların olduğu koridorları içeren küme
KCAKT = ∅  // Arka arkaya çakışmaların olduğu hareket ikililerini içeren küme
for cozum_indisi=1:alternatif_optimum _cozum_sayisi
  for dar_koridor_indisi=1:dar_koridor_sayisi
    for rota _1=1: rota_sayisi
      for rota _2=1: rota_sayisi;  rota _2 > rota _1
        for hareket_1=1: hareket_sayisi(rota_1)
          for hareket_2=1: hareket_sayisi(rota_2)
            if (koridora_giris_zamani(rota_1)> koridora _giris_zamani(rota_2) &  koridordan
              _cikis_zamani(rota_1) >  koridordan _cikis_zamani(rota_2)) ||
              (koridora_giris_zamani(rota_1)<  koridora _giris_zamani(rota_2) &  koridordan
              _cikis_zamani(rota_1) <  koridordan _cikis_zamani(rota_2))
            else
              ACAK ← ACAK ∪ {dar_koridor_indisi}
              ACAKT ← ACAKT ∪ {hareket_1, hareket_2}
            end
            if (koridora_giris_zamani(rota_1)> koridordan _cikis_zamani(rota_2)) ||
              (koridordan_cikis_zamani(rota_1)< koridora_giris_zamani(rota_2))
            else
              KCAK ← KCAK ∪ {dar_koridor_indisi}
              KCAKT ← KCAKT ∪ {hareket_1, hareket_2}
            end
          end
        end
      end
    end
  end
end
```

Şekil 4.11. Çakışma kontrolü algoritmasının sözde kodu

Alternatif optimum çözümlere uygulanacak çakışma kontrolü algoritmasının sözde kodu Şekil 4.11'de sunulmaktadır. Görüldüğü gibi her bir alternatif optimum çözüm için eğer

varsa karşılıklı ve arka arkaya çakışmaların hangi dar koridorda ve çekici trenlerin hangi hareketlerini yaparken meydana geldiği belirlenmektedir. Daha sonra Model-1'e sadece belirlenen bu mevcut çakışmalar için kısıtlar yazılarak çözüm yapılmaktadır. Çakışma engelleme kısıtları yazılırken yukarıda sözde kodu verilen algoritma ile elde edilen şu kümeler kullanılmaktadır;

$KCAK$: Karşılıklı çakışmanın olduğu (i,j) koridorlarını içeren küme, $(KCAK \subset DY)$

$KCAKT_{(i,j)}$: (i,j) koridorunda karşılıklı çakışmanın olduğu hareket ikililerini içeren küme $\left((i,j) \in KCAK,\ KCAKT_{(i,j)} \subset (T \times T)\right)$

$ACAK$: Aynı yönde çakışmanın olduğu (i,j) koridorlarını içeren küme, $(ACAK \subset DY)$

$ACAKT_{(i,j)}$: (i,j) koridorunda aynı yönde çakışmanın olduğu hareket ikililerini içeren küme $\left((i,j) \in ACAK,\ ACAKT_{(i,j)} \subset (T \times T)\right)$

Bu kümeler kullanılarak karşılıklı çakışmalar için Model-1'de Denklem (4.35 - 4.36) yerine yazılacak kısıtlar aşağıdaki denklemlerde verilmektedir.

$$z^{k(t-1)} > z^{fy} + M * \left(x_{ij}^{kt} - 1\right) + M * \left(x_{ji}^{fy} - 1\right) - M * \left(u_{ij}^{ktfy}\right),$$
$$(i,j) \in KCAK, (t,y) \in KCAKT_{(i,j)}, \forall k, f \in K, f \neq k \quad (4.64)$$

$$z^{kt} < z^{f(y-1)} - M * \left(x_{ij}^{kt} - 1\right) - M * \left(x_{ji}^{fy} - 1\right) + M * \left(1 - u_{ij}^{ktfy}\right), (i,j)$$
$$\in KCAK, \quad (t,y) \in KCAKT_{(i,j)}, \quad \forall k, f \in K, f \neq k \quad (4.65)$$

Aynı şekilde Denklem (4.41 – 4.44) yerine aynı yön çakışmaları için yazılacak kısıt denklemleri ise aşağıda verildiği gibidir.

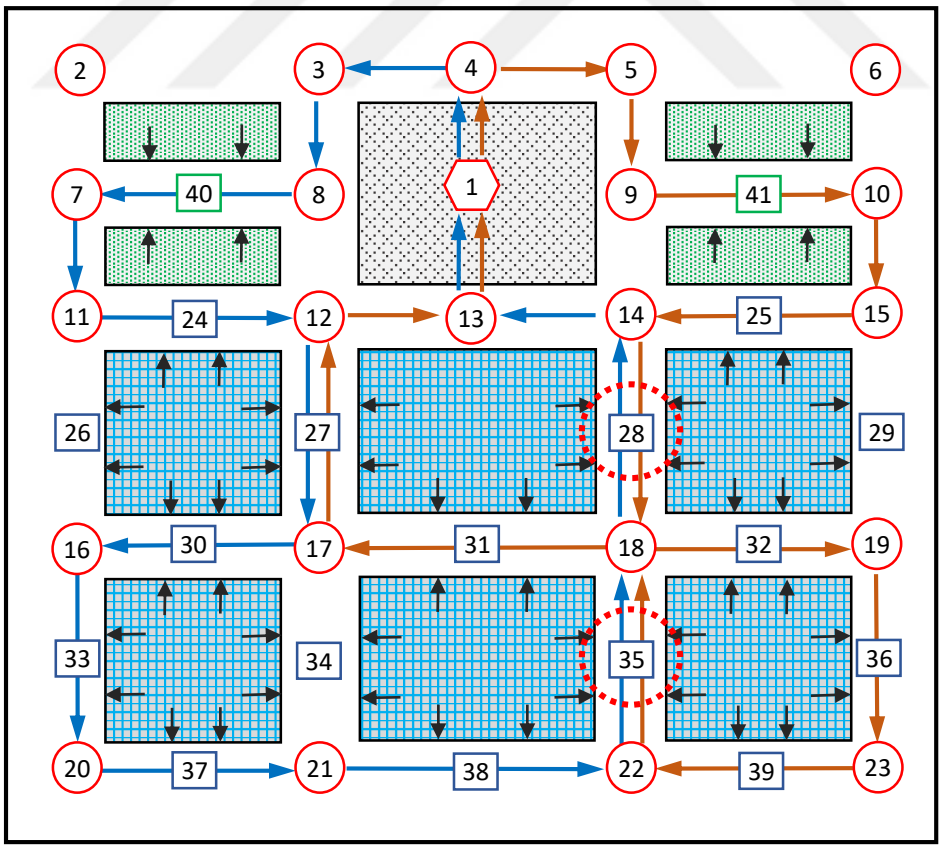
$$z^{k(t-1)} \leq z^{f(y-1)} + M * \left(1 - s_{ij}^{kt}\right) + M * \left(1 - x_{ij}^{fy}\right) + M * \left(h_{ij}^{ktfy}\right),$$
$$(i,j) \in ACAK, (t,y) \in ACAKT_{(i,j)}, \forall k, f \in K, f \neq k \quad (4.66)$$

$$z^{kt} \leq z^{fy} + M * \left(1 - s_{ij}^{kt}\right) + M * \left(1 - x_{ij}^{fy}\right) + M * \left(h_{ij}^{ktfy}\right),$$
$$(i,j) \in ACAK, (t,y) \in ACAKT_{(i,j)}, \forall k, f \in K, f \neq k \quad (4.67)$$

$$z^{f(y-1)} < z^{k(t-1)} + M * \left(1 - s_{ij}^{kt}\right) + M * \left(1 - x_{ij}^{fy}\right) + M * \left(1 - h_{ij}^{ktfy}\right),$$
$$(i,j) \in ACAK, (t,y) \in ACAKT_{(i,j)}, \forall k, f \in K, f \neq k \quad (4.68)$$

$$z^{fy} < z^{kt} + M * \left(1 - s_{ij}^{kt}\right) + M * \left(1 - x_{ij}^{fy}\right) + M * \left(1 - h_{ij}^{ktfy}\right),$$
$$(i,j) \in ACAK, (t,y) \in ACAKT_{(i,j)}, \forall k, f \in K, f \neq k \quad (4.69)$$

Bu duruma bir örnek olarak, iki çekici trenin bulunduğu bir problemde Model-2 ile çözüm yapıldıktan sonra elde edilen optimum çözümlerden bir tanesi Şekil 4.12'de verildiği gibi bulunmuş olsun. Bu çözümde çakışma kontrol algoritması ile sadece 28 ve 35. koridorlarda çakışma tespit edildiğini varsayalım. Bu durumda 28. koridorda herhangi iki çekici trenin 13. ve 7. hareketleri için karşılıklı çakışma kontrol kısıtları, 35. koridor için ise herhangi iki çekici trenin 12 ve 8. hareketleri için aynı yönde çakışma kontrol kısıtları yazılacaktır. Bu kısıtlar çekici trenlerin bütün olası ikili kombinasyonu için yazılacağından Model-1 ile yapılacak bir sonraki çözümde bu koridorlarda ilgili hareketler için çakışma olmayacaktır.



Şekil 4.12. Model-2 ile elde edilmiş örnek bir optimum çözümde var olan çakışmalar

Bu şekilde Model-1 ile çözüm yapıldıktan sonra elde edilen çözüme tekrar çakışma kontrolü uygulanacak eğer çakışma varsa bir sonraki iterasyonda aynı şekilde kısıtlar eklenerek tekrar çözüm yapılacaktır. Ayrıca her iterasyonda bir önceki iterasyonun çözüm değerinin alt sınır olarak kullanılması süreci oldukça hızlandırmaktadır. Böylelikle belirli bir boyuta kadar olan problemler için birinci çözüm yaklaşımına göre çok daha hızlı ve daha az kaynak kullanarak optimum çözüm elde edilebilmektedir

4.3. Üçüncü Çözüm Yaklaşımı: Mat-Sezgisel Algoritma

Döngüsel sefer rotalama ve çizelgeleme problemi, zaman pencereli ve kapasite kısıtlı araç rotalama problemi ile birçok yönden benzerlik göstermektedir (Gyulai ve diğ, 2013). Her iki problemde, sınırlı kapasiteye sahip, sınırlı bir araç filosunun önceden belirli bir ağ üzerinde zamanı dikkate alan farklı bir takım ek kısıtlarla rotalanmasını içermektedir. Bu nedenle birçok çalışmada fabrika içinde döngüsel seferlerle malzeme dağıtımı için taşıyıcı araçların rotalanması ve çizelgelenmesi problemlerinin, araç rotalama probleminin özel bir durumu olarak NP-zor problem sınıfında olduğu ispat ve ifade edilmiştir (Fathi ve diğ., 2016; Zhou ve Peng, 2017; Emde ve Schneider, 2018; Bocewicz ve diğ., 2019; Sipahioğlu ve Altın, 2019; Zhou ve Zhu, 2020). Benzer şekilde, bu çalışmada tanımlanan, sabit çevrim süresine sahip döngüsel seferlerle malzeme tedariği için çekici trenlerin çakışmasız olarak rotalanması, yüklenmesi ve çizelgelenmesi problemlerini eş zamanlı ele alan SÇDRÇ probleminin, araç rotalama probleminin özel bir durumu olarak NP-zor sınıfına ait olduğu ifade edilebilir (Levner ve diğ., 2010).

Küçük ve orta boyutlu olarak kabul edilebilecek problemler önceki bölümde geliştirilen çözüm yaklaşımı ile çözülebilse de SÇDRÇ probleminin NP-zor doğası sebebiyle, kroki boyutlarına, dar yol oranına, çekici tren sayısına ve görev sayısına bağlı olarak problem boyutu büyüdükçe iteratif çözüm yaklaşımı ile de makul sürelerde çözüm bulmak mümkün olmamaktadır. Bu nedenle bu bölümde gerçek hayat problemlerini etkili bir şekilde çözebilecek komşuluk ve popülasyon temelli bir mat-sezgisel algoritma önerilmektedir.

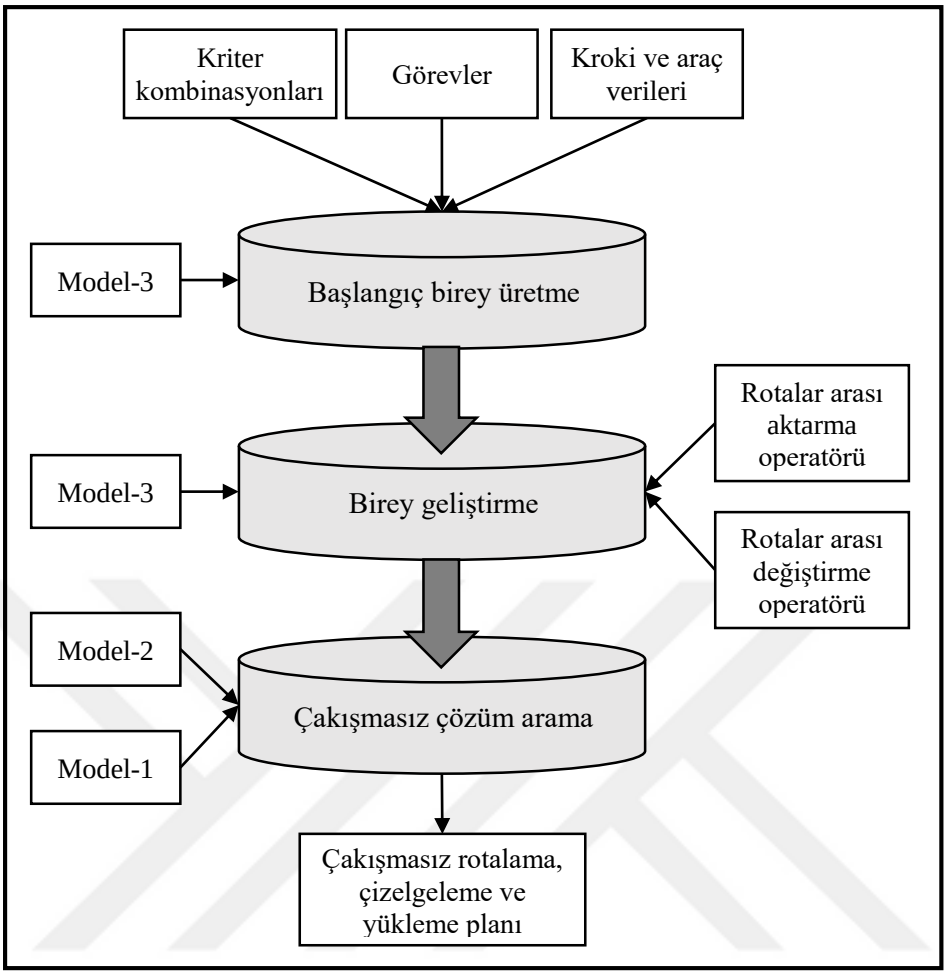
Mat-sezgisel algoritmalar, matematiksel programlama modelleri gibi kesin çözüm teknikleri ile sezgisel veya metasezgisel yöntemlerin birleşiminden oluşan çözüm

yaklaşımlarıdır (Boschetti ve Maniezzo, 2022). Genel olarak çözüm uzayındaki komşulukları araştırmak için matematiksel teknikleri kullanan mat-sezgisel algoritmaların çeşitli problem türlerinin çözümünde kullanımı bilgisayarların artan işlem gücü ile gittikçe yayınlaşmaktadır (Della Croce ve Salassa, 2014; Fonseca ve diğ., 2016; Lahyani ve diğ, 2019; Martinez-Sykora ve diğ., 2017). Özellikle araç rotalama problemlerine kısa sürede kaliteli çözümler üretmek için sıklıkla tercih edildikleri görülmektedir (Schermer ve diğ., 2019; Sartori ve Buriol, 2020; Queiroga ve diğ., 2021).

Büyük boyutlu SÇDRÇ problemlerinin makul sürelerde efektif bir şekilde çözülebilmesi için önerilen mat-sezgisel algoritmanın genel çerçevesi Şekil 4.13'te gösterilmektedir. Algoritma özetle, başlangıç çözüm üretme, çözüm geliştirme ve çakışmasız çözüm arama olmak üzere üç temel aşamadan oluşmaktadır.

İlk aşamada, çakışmalar göz ardı edilerek, görevlerin çekici trenlere çeşitli kriterlere göre atanması ile olabildiğince iyi çözüm değerine sahip olurlu başlangıç çözümler (bireyler) oluşturulmaktadır. İkinci aşamada ise ilk aşamada üretilen olurlu bireylerin her biri genetik algoritmanın mantığına benzer şekilde belirli bir popülasyon büyüklüğü ile nesiller boyunca probleme özgü olarak yapılandırılmış komşu oluşturma operatörleri ile geliştirilmekte ve daha iyi bireyler elde edilmektedir. Bu aşamada da daha hızlı ve verimli bir arama gerçekleştirmek için çakışma durumları göz ardı edilmektedir. Üçüncü aşamada ise önceki aşamada elde edilen, iyi çözüm değerlerine sahip bireyler kullanılarak çakışmasız çözüm aranmaktadır.

Şekilde görüldüğü gibi her üç aşamada da daha önceki bölümlerde geliştirilen matematiksel modeller kullanılmaktadır. İlk iki aşamada çakışma durumları göz ardı edilerek Model-3'ün hızlı çözüm üretebilme avantajından faydalanılmaktadır. Üçüncü aşamada ise birey bazında çakışmasız çözüm aranırken iteratif çözüm yaklaşımında olduğu gibi alt sınırlar kullanılarak Model-2 ve Model-1'den yararlanılmaktadır. Böylelikle büyük boyutlu problemlerin çözümü için daha hızlı ve verimli bir algoritma elde edilmektedir.



Şekil 4.13. Mat-sezgisel algoritmanın genel çerçevesi

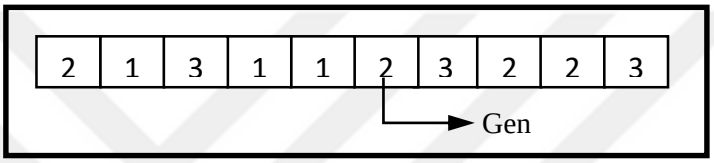
Algoritmanın başlıca girdileri arasında, köşe noktalarının ve koridor orta noktalarının koordinatları, mesafe matrisi, dar koridorlar gibi kroki bilgileri, taşıyıcı araçların sayısı ve kapasitesi ile ilgili bilgiler, görevlerin alma-bırakma koridorları ve miktarlarını içeren bilgiler ve başlangıç çözüm oluşturma sırasında kullanılacak kriter kombinasyon bilgileri bulunmaktadır. Bunlara ek olarak kullanılan diğer parametreler ilerleyen bölümlerde ayrıntılı olarak ifade edilecektir. Şekilde görüldüğü gibi algoritmanın çıktısı ise SÇDRÇ problemi için makul sürelerde optimum çözüme oldukça yakın veya optimum değerde çakışmasız rotalama, çizelgeleme ve yükleme planının elde edilmesidir.

Bu çalışmada birey yapısı her bir görevin atandığı rotaların numaralarının arka arkaya dizilmesi ile elde edilmektedir. Örneğin 10 görevin 3 araçla yapılacağı bir fabrika içi döngüsel malzeme dağıtım sistemini varsayalım ve her bir görevin atandığı rota Tablo 4.6'da görüldüğü gibi olsun.

Tablo 4.6. Örnek bir görev-rota ataması

Görev	1	2	3	4	5	6	7	8	9	10
Görevin atandığı rota	2	1	3	1	1	2	3	2	2	3

Bu durum için birey gösterimi Şekil 4.14'te verildiği gibi olacaktır. Burada her bir elemana gen adı verilmektedir. Bir bireyin uygunluk değerini elde etmek için öncelikle her bir araç kendisine atanan görevlere göre Model-3 ile rotalanmakta daha sonra bu rotaların süreleri toplanmaktadır. Elde edilen bu uygunluk değeri çakışmaların göz ardı edildiği durum için geçerlidir. Mat-sezgisel algoritmanın aşamaları ile ilgili detaylı bilgiler, sözde kodlar ve akış şemaları ile birlikte ilerleyen bölümlerde verilmektedir.

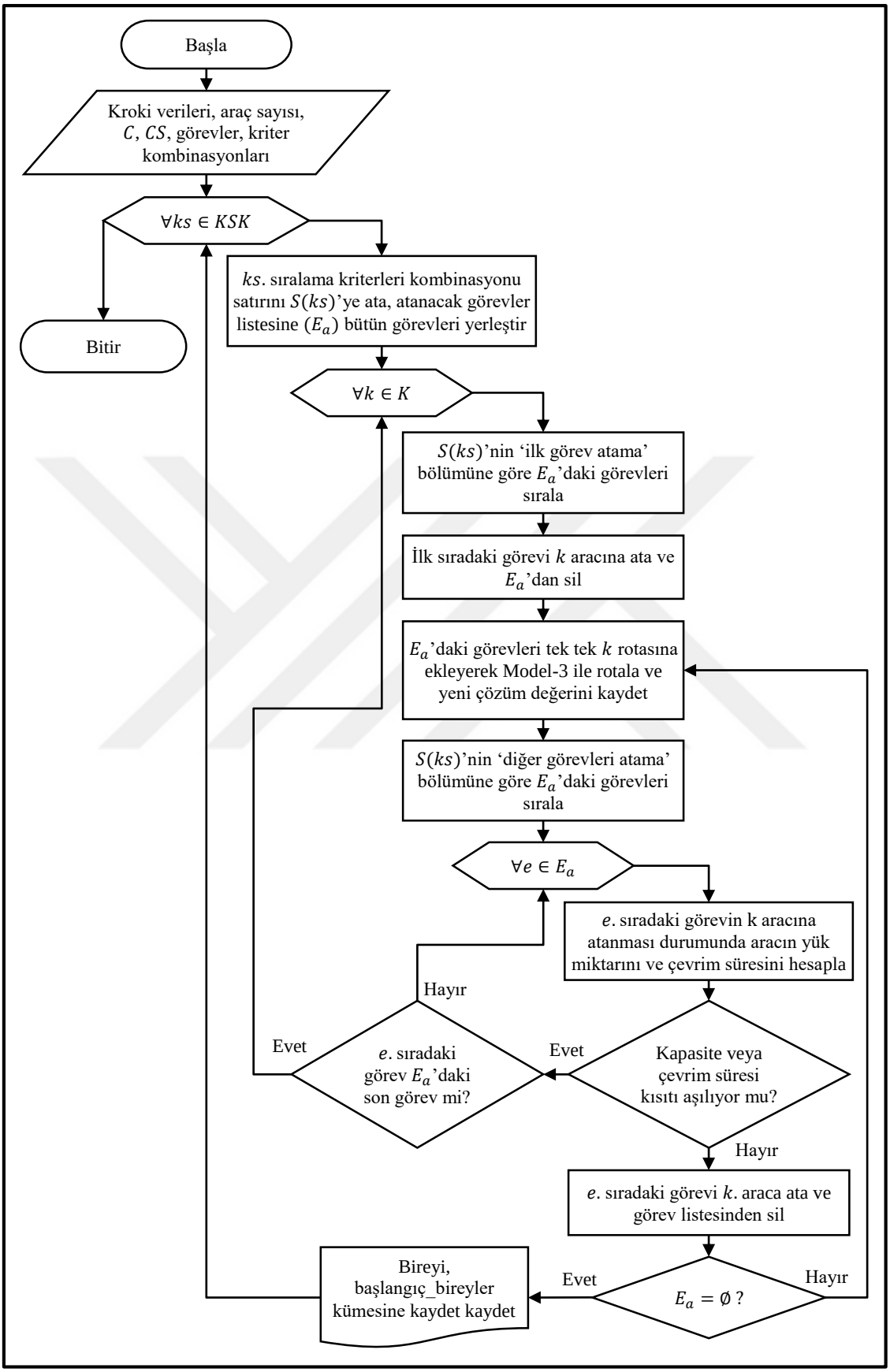


Şekil 4.14. Örnek bir birey gösterimi

4.3.1. Başlangıç Birey Üretme

Mat-sezgisel algoritmanın ilk aşamasında matematiksel model (Model-3) tabanlı bir birey oluşturucu prosedür yer almaktadır. İlk olarak başlangıç bireylerin rasgele oluşturulması düşünülse de bu yaklaşımla elde edilen bireylerin çok kötü çözüm değerlerine sahip oldukları ve mat-sezgisel algoritmanın performansını olumsuz yönde etkiledikleri görülmüştür. Bu nedenle bir mantık çerçevesinde Model-3 kullanılarak kaliteli başlangıç bireylerin üretilmesi ve böylelikle algoritmanın genel performansının artırılması düşünülmüştür. Bu amaçla oluşturulan başlangıç birey üretme algoritmasının akış şeması ve sözde kodu sırasıyla Şekil 4.15 ve Şekil 4.16'da verilmektedir. Akış şeması ve sözde kod yazılımında kullanılan kısaltmaların açıklamaları şu şekildedir;

- KSK : Sıralama kriterlerinin kombinasyonlarını içeren küme
- ks : Sıralama kriterinin kombinasyon sırası indisi
- $S(ks)$: ks. sıralama kriterleri kombinsayonu satırı
- K : Araçlar kümesi
- k : Araç indisi
- E_a : Atanacak görevler kümesi
- e : Görev sırası.



Şekil 4.15. Başlangıç birey oluşturma algoritmasının akış şeması

```
Girdiler: kriter_kombinasyonları, görev_listesi, araç_sayısı, C, CS, kroki_verileri
Çıktılar: başlangıç_bireyler
for ks = 1: kombinasyon_sayısı
S(ks) ← ks. kombinasyon satırı
E_a ← görev_listesi
  for k = 1: araç_sayısı
  E_a'yı S(ks)'nin "ilk_görev_atama" bölümüne göre sırala
  E_a'nın ilk sırasındaki görevi k. araca ata
  atanan görevi E_a'dan sil
  k. aracı Model 3 ile rotala
  atama_yapıldı_mı ← evet
    while n(E_a) > 0 & atama yapıldı mı == evet
    atama_yapıldı_mı ← hayır
    E_a'daki görevleri tek tek k. araca ekleyerek Model 3 ile rotalama yap ve çözüm değerini kaydet
    E_a'yı (ks)'nin "diğer_görevleri_atama" bölümüne göre sırala
    e ← 1 olarak belirle
      for e = 1: n(E_a)
        e. sırasındaki görevin k. araca atanması durumunda araç yükü ve CS değerini hesapla
        if kapasite ve CS aşılmıyor
          k. araca e. sırasındaki görevi ata
          atanan görevi E_a'dan sil
          atama yapıldı mı ← evet
          break    // for e = 1: n(E_a) döngüsünü sonlandırmak için
        end if
      end for e
    end while
  end for k
  if E_a == ∅
    bireyi, "başlangıç_bireyler"e kaydet
  else
    geçersiz birey
  end if
end for ks
```

Şekil 4.16. Başlangıç birey oluşturma algoritmasının sözde kodu

Sözde kod ve akış şemasından anlaşılabileceği üzere, başlangıç çözüm oluşturma prosedürü ile temel olarak yapılmak istenen şey; rotaların yolda geçen sürelerinin toplamı olabildiğince az olacak şekilde kaliteli başlangıç bireyler oluşturmaktır. Bu amaçla, görevlerin alma ve bırakma noktalarının krokinin hangi bölgesinde bulunduğu, görevlerin kutu cinsinden miktarları, yeni bir görevin bir araca eklendiğinde yolu ne kadar uzattığı gibi bir dizi görev sıralama kriterleri kullanılmaktadır. Bu kriterlerin yönü artan ya da azalan şekilde belirlenerek ve kriterler farklı sıralarda sıralanarak atanacak görevlerin öncelik sıralaması için sıralama kriterleri kombinasyonları oluşturulmaktadır. Tablo 4.7'de bu çalışmada kullanılan 48 adet sıralama kriterleri kombinasyonlarının tamamı verilmektedir.

Tablo 4.7. Görev sıralama kriterleri kombinasyonları

Sıra No	Rotaya ilk görevi atama				Rotaya diğer görevleri atama					
	Kriter 1	**Kriter 2**	**Kriter 3**	**Kriter 4**	**Kriter 1**	**Kriter 2**	**Kriter 3**	**Kriter 4**	**Kriter 5**	**Kriter 6**
1	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma Y (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (azalan)
2	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma Y (azalan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (azalan)
3	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma Y (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (azalan)
4	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma Y (azalan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (azalan)
5	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Rota sür. (artan)	Alma Y (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
6	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Rota sür. (artan)	Alma Y azalan	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
7	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma Y (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (azalan)
8	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma Y azalan	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (azalan)
9	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (artan)	Rota sür. (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (azalan)
10	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (azalan)	Rota sür. (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (azalan)
11	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (artan)	Rota sür. (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (azalan)
12	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (azalan)	Rota sür. (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (azalan)
13	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (artan)	Rota sür. (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
14	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (azalan)	Rota sür. (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
15	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (artan)	Rota sür. (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (azalan)
16	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (azalan)	Rota sür. (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (azalan)
17	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (azalan)
18	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (azalan)
19	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (azalan)
20	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (azalan)
21	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
22	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
23	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
24	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (azalan)
25	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma Y (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (artan)

Tablo 4.7. (Devam) Görev sıralama kriterleri kombinasyonları

26	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma Y (azalan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (artan)
27	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma Y (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (artan)
28	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma Y (azalan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (artan)
29	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Rota sür. (artan)	Alma Y (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (artan)
30	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Rota sür. (artan)	Alma Y (azalan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (artan)
31	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma Y (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (artan)
32	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma Y (azalan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (artan)
33	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (artan)	Rota sür. (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (artan)
34	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (azalan)	Rota sür. (artan)	Alma X (artan)	Alma kor. (gidilen)	Miktar (artan)
35	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (artan)	Rota sür. (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (artan)
36	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (azalan)	Rota sür. (artan)	Alma X (azalan)	Alma kor. (gidilen)	Miktar (artan)
37	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (artan)	Rota sür. (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (artan)
38	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (azalan)	Rota sür. (artan)	Bırakma X (artan)	Bırakma kor. (gidilen)	Miktar (artan)
39	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (artan)	Rota sür. (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (artan)
40	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (azalan)	Rota sür. (artan)	Bırakma X (azalan)	Bırakma kor. (gidilen)	Miktar (artan)
41	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (artan)	Alma X (artan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (artan)
42	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Miktar (azalan)	Bırakma X (artan)	Bırakma Y (azalan)	Alma X (artan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (artan)
43	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (artan)	Alma X (azalan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (artan)
44	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Miktar (azalan)	Bırakma X (azalan)	Bırakma Y (azalan)	Alma X (azalan)	Rota sür. (artan)	Alma kor. (gidilen)	Miktar (artan)
45	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (artan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (artan)
46	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Miktar (azalan)	Alma X (artan)	Alma Y (azalan)	Bırakma X (artan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (artan)
47	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (artan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (artan)
48	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Miktar (azalan)	Alma X (azalan)	Alma Y (azalan)	Bırakma X (azalan)	Rota sür. (artan)	Bırakma kor. (gidilen)	Miktar (artan)

Tabloda alan kısıtlı olduğu için kısa şekilde verilen ifadelerin açıklamaları şu şekildedir;

- Bırakma X (artan): Bırakma koridorunun orta noktasının X koordinatı daha küçük olan göreve öncelik verilmesi

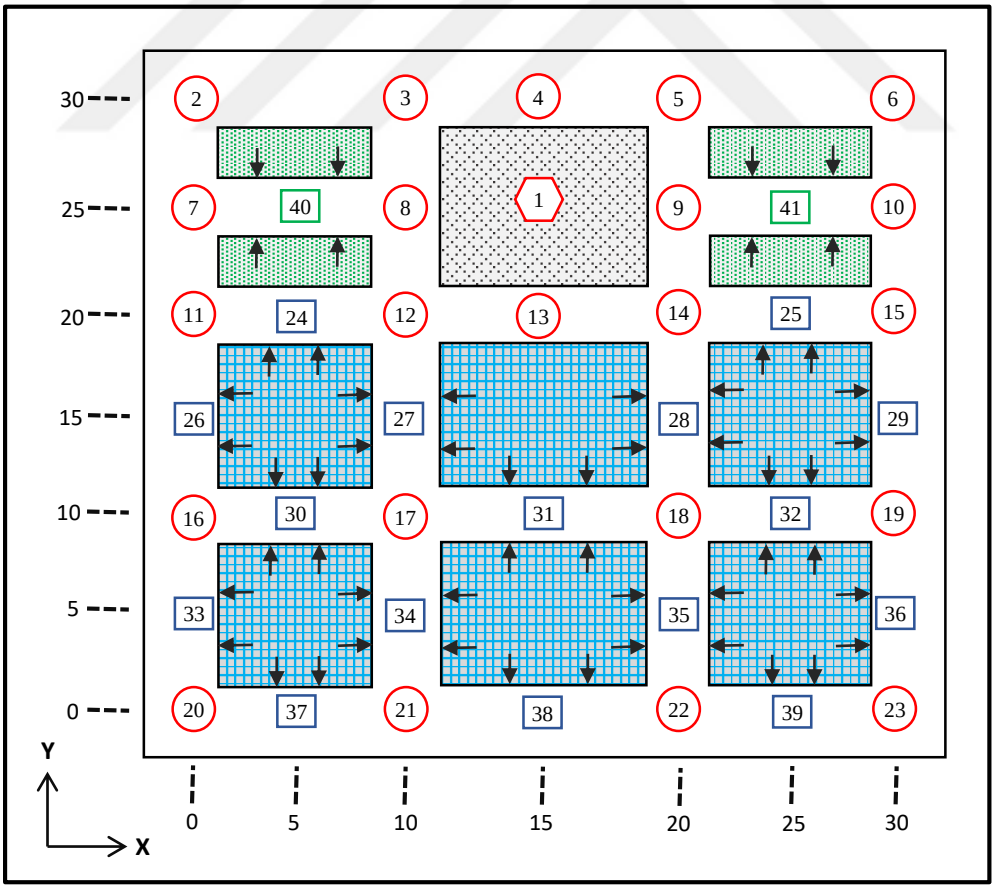
- Bırakma X (azalan): Bırakma koridorunun orta noktasının X koordinatı daha büyük olan göreve öncelik verilmesi
- Bırakma Y (artan): Bırakma koridorunun orta noktasının Y koordinatı daha küçük olan göreve öncelik verilmesi
- Bırakma Y (azalan): Bırakma koridorunun orta noktasının Y koordinatı daha büyük olan göreve öncelik verilmesi
- Alma X (artan): Alma koridorunun orta noktasının X koordinatı daha küçük olan göreve öncelik verilmesi
- Alma X (azalan): Alma koridorunun orta noktasının X koordinatı daha büyük olan göreve öncelik verilmesi
- Alma Y (artan): Alma koridorunun orta noktasının Y koordinatı daha küçük olan göreve öncelik verilmesi
- Alma Y (azalan): Alma koridorunun orta noktasının Y koordinatı daha büyük olan göreve öncelik verilmesi
- Rota sür. (artan): Rotaya eklenmesi durumunda aracın yolda geçirdiği süreyi en az artıran göreve öncelik verilmesi
- Miktar (artan): Kutu cinsinden miktarı daha az olan göreve öncelik verilmesi
- Miktar (azalan): Kutu cinsinden miktarı daha fazla olan göreve öncelik verilmesi
- Bırakma kor. (gidilen): İlgili araç tarafından bırakma koridoruna daha önce atanan görevler nedeniyle gidilen göreve öncelik verilmesi
- Alma kor. (gidilen): İlgili araç tarafından alma koridoruna daha önce atanan görevler nedeniyle gidilen göreve öncelik verilmesi

Tablodaki 48 sıradan her biri başlangıç birey oluşturulurken görevlerin araçlara atanma önceliğini belirlemek için nasıl sıralanacağını ifade etmektedir. Görüldüğü gibi her satırda, çekici trenlere ilk görevin atanması için kullanılacak önceliklendirme kriterleri ve diğer görevlerin atanması için kullanılacak önceliklendirme kriterleri ayrı ayrı verilmiştir. Örneğin birinci kriter kombinasyonu satırına göre bir araca ilk görev atanırken, atanmamış olan görevler sırasıyla; bırakma koridorunun orta noktasının X koordinatına göre artan sırada, bırakma koridorunun orta noktasının Y koordinatına göre artan sırada, alma koridorunun X koordinatına göre artan sırada ve kutu cinsinden miktarlarının büyüklüğüne göre azalan sırada sıralanmalıdırlar. Bu sıralama yapıldıktan sonra ilk

sıradaki görev araca atanmaktadır. İlk görev atandıktan sonra diğer görevlerin içinden hangisinin aynı araca atanacağına karar vermek için gene birinci satırda yer alan 6 kriter sırasıyla göz önünde bulundurulacaktır. Böylelikle bu süreç sözde kod ve akış şemasında görüldüğü gibi kapasite ve çevrim süresi kısıtlarının izin verdiği ölçüde devam edecektir.

Tabloda verilen 48 sıralama kriteri kombinasyon satırının her biri kullanılarak toplamda 48 adet başlangıç bireyinin oluşturulması hedeflenmektedir. Ancak sözde kod ve akış şemasında gösterildiği gibi bazı kombinasyonlarda kapasite ve çevrim süresi kısıtlarına uymadığı için görevlerin tamamı araçlara atanamamakta ve 48 den az sayıda birey oluşabilmektedir. Bazen de farklı iki kombinasyon aynı bireyin oluşmasıyla sonuçlanabilmektedir.

Başlangıç birey oluşturma sürecinin daha net anlaşılabilmesi için somut bir örnek olarak; Şekil 4.17'de koordinat eksenleri eklenmiş küçük bir kroki üzerinde Tablo 4.8'de verilen 10 adetlik görev listesinin çekici trenlere atanması durumunu ele alalım.



Şekil 4.17. Koordinat eksenleri yerleştirilmiş bir fabrika krokisi

İlk bireyi oluşturmak için görevler Tablo 4.7'nin birinci satırındaki sıralama kriterleri kombinasyonuna göre araçlara atanacaktır. Bunun için öncelikle görevler, kombinasyon satırının "rotaya ilk görev atama" kısmındaki 4 kriter baz alınarak sıralanmalıdır. Buna göre ilk olarak görevler bırakma koridorlarının orta noktasının X koordinatına göre artan sırada sıralanacaktır. X koordinat değeri aynı olan görevler ise kendi içerisinde ikinci kriter olan bırakma koridorunun orta noktasının Y koordinatına göre artan sırada sıralanacaktır. Bırakma koridorunun hem X hem de Y koordinatı aynı olan görevler ise üçüncü sıradaki kriter olan alma koridorunun orta noktasının X koordinatına göre artan sırada sıralanacaktır. Son olarak her üç kritere göre değerleri aynı olan görevler dördüncü kriter olan görev miktarına göre azalan sırada sıralanacaktır. Eğer bu dört kritere göre birden fazla görevin değerleri aynıysa, yani ilk sırada olmaları gerekiyorsa, içerisinden bir tanesi rasgele olarak ilk sıraya yerleşecektir. Bu şekilde Tablo 4.8'deki görevler Tablo 4.7'nin birinci satırındaki sıralama kombinasyonunun "rotaya ilk görev atama" kısmındaki kriterlere göre sıralandığında Tablo 4.9'da verilen sıralama elde edilmektedir. Tablonun ilk sırasındaki (yeşil ile işaretlenmiş) 7 numaralı görev ilk rotaya atanacak ve atanacak görevler listesinden silinecektir.

Tablo 4.8. Örnek atanacak görevler listesi

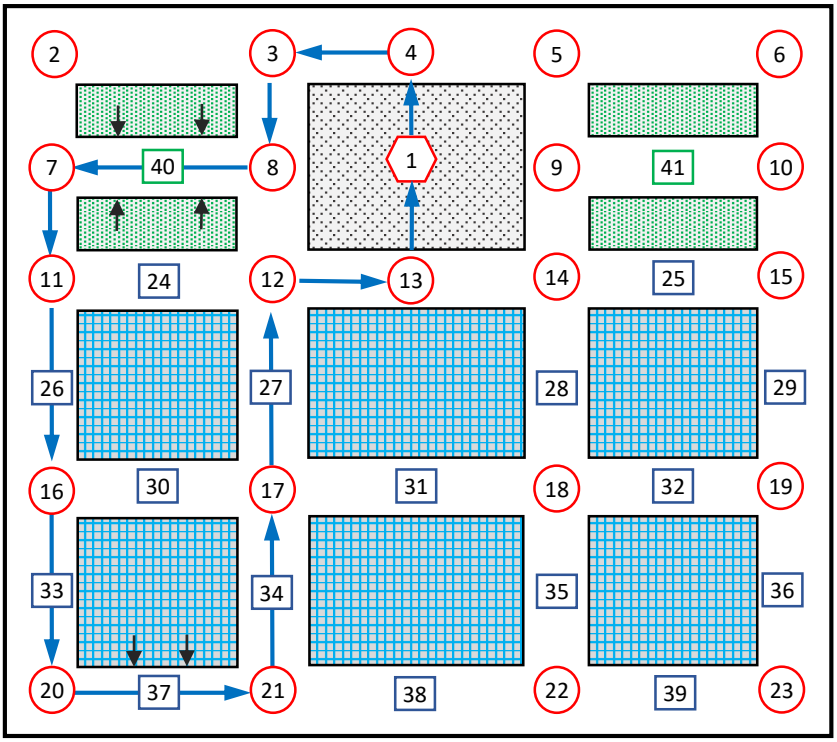
Görev no	Bırakma koridoru	Alma koridoru	Miktar
1	37	41	3
2	26	40	2
3	27	40	5
4	33	41	3
5	32	41	3
6	26	40	5
7	33	40	4
8	37	40	1
9	33	40	2
10	30	41	4

7 numaralı görevin atandığı araç, görevi yapabilmek için başlangıç noktasından çıkıp 33 ve 40 numaralı koridorlara gitmeli ve geri başlangıç noktasına dönmelidir. Bu şekilde Model-3 ile rotalama yapıldığında araç için alternatif optimum rotalardan bir tanesi Şekil 4.18'de görüldüğü gibi elde edilmektedir. Aracın yolda geçirdiği süre ise, mesafeler metre, araç hızı da 1 m/sn kabul edildiğinde 90 sn olmaktadır.

Tablo 4.9. Birinci kombinasyon satırının ilk görev atama kriterlerine göre sıralanmış atanacak görevler listesi

Görev no	Bırakma koridoru	Alma koridoru	Miktar	Bırakma X Kor.	Bırakma Y Kor.	Alma X Kor.	Alma Y Kor.
7	33	40	4	0	5	5	25
9	33	40	2	0	5	5	25
4	33	41	3	0	5	25	25
6	26	40	5	0	15	5	25
2	26	40	2	0	15	5	25
8	37	40	1	5	0	5	25
1	37	41	3	5	0	25	25
10	30	41	4	5	10	25	25
3	27	40	5	10	15	5	25
5	32	41	3	25	10	25	25

Rotaya ilk görev atandıktan sonra kalan görevler, ilgili kombinasyon satırının "rotaya diğer görevleri atama" kısmındaki kriterlere göre sıralanmalıdır. Verilen örnek görev listesinde kalan görevlere bu sıralama yapıldığında Tablo 4.10 elde edilmektedir.



Şekil 4.18. Birinci görev atandıktan sonra Model 3 ile oluşturulan rota

Tablo 4.10'da görüldüğü üzere kalan görevlerden her birinin ayrı ayrı eğer araca atanırsa toplam yolda geçen süreyi ne kadar uzatacağı Model-3 ile hesaplanarak "Rota sür. uzatma" sütununa yazılmıştır. Bazı görevlerin alma ve bırakma koridorları halihazırda alternatif optimum rotaların herhangi birinin güzergahında olduğundan rota süresini uzatmazken diğer bazı görevler aracın daha uzun yoldan gitmesini gerektirmektedir. Bu şekilde elde edilen yol süresi uzatma kriteri farklı kombinasyonlarda farklı önceliğe sahip olarak dikkate alınmaktadır.

Verilen örnekte sıralama sonucunda birinci sıradaki 9 numaralı görev, aracın kapasite ve çevrim süresi kısıtları aşılmıyorsa araca atanacak ve atanacak görevler listesinden silinecektir.

Tablo 4.10. Birinci kombinasyon satırın diğer görevleri atama kriterlerine göre sıralanmış atanacak görevler listesi

Görev no	Bırakma koridoru	Alma koridoru	Miktar	Bırakma X Kor.	Bırakma Y Kor.	Alma X Kor.	Alma Y Kor.	Rota sür. uzatma	Alma kor. gidildi mi?
9	33	40	2	0	5	5	25	0	1
6	26	40	5	0	15	5	25	0	1
2	26	40	2	0	15	5	25	0	1
4	33	41	3	0	5	25	25	+40	0
8	37	40	1	5	0	5	25	0	1
1	37	41	3	5	0	25	25	+40	0
10	30	41	4	5	10	25	25	+60	0
3	27	40	5	10	15	5	25	0	1
5	32	41	3	25	10	25	25	+40	0

Eğer sıralamanın birinci sırasındaki görev araca atandığında, araç kapasite veya çevrim süresi kısıtlarını aşıyorsa ikinci sıradaki göreve bakılacaktır. Bu şekilde kısıtları ihlal etmeyen ilk görev araca atanacak ve listeden silinecektir. Eğer kalan görevlerden hiçbiri araca atanamıyorsa, diğer araca geçilecek ve ilk atama için görevler yukarıda olduğu gibi yeniden sıralanacaktır.

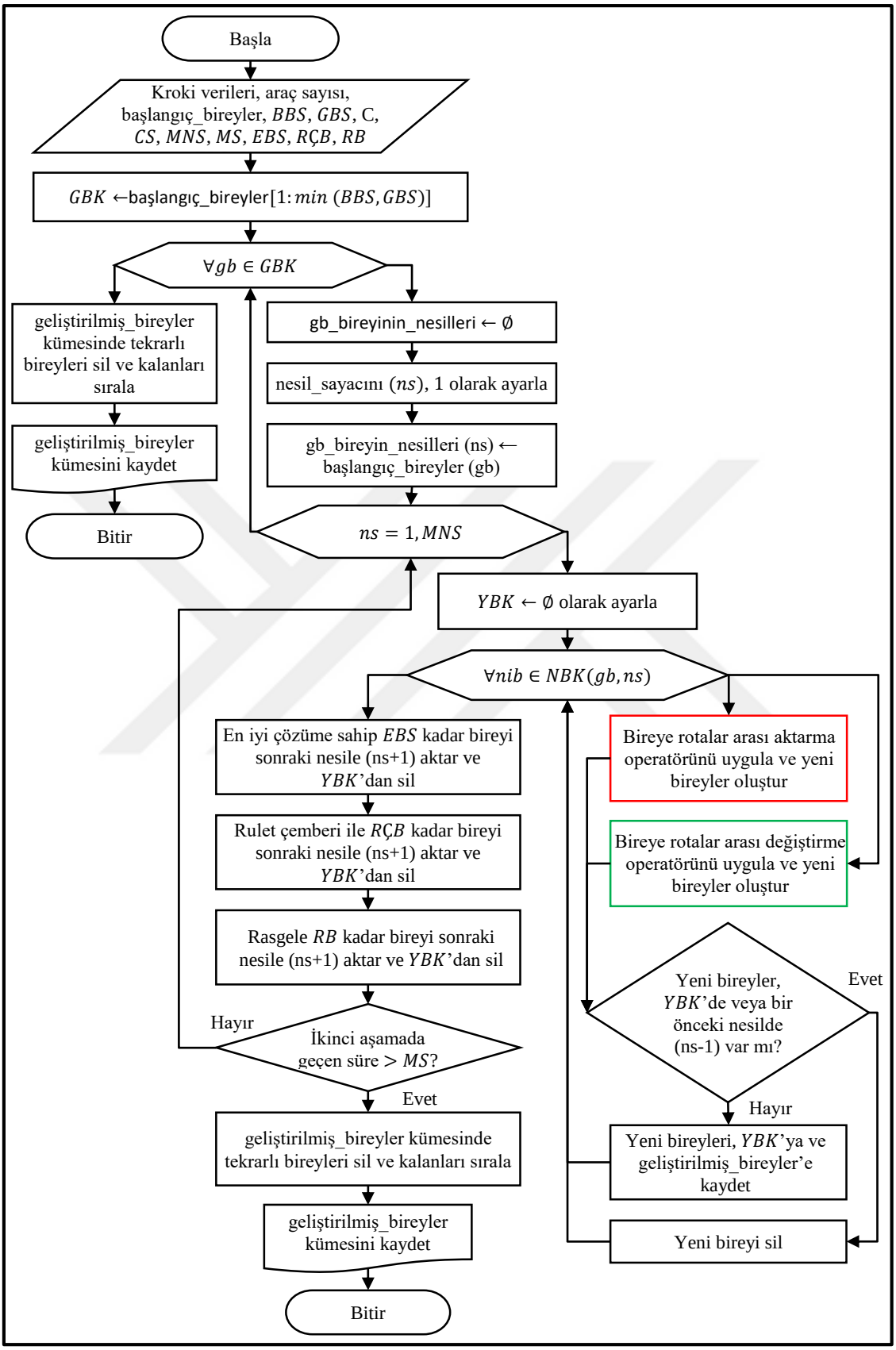
Bu şekilde görevlerin tamamı araçlara atanabilirse sağlam bir birey ortaya çıkacaktır. Aksi durumda açıkta atanamayan görev kalırsa, çözüm ifade etmeyen geçersiz bir birey elde edilmiş olacaktır. Böylelikle 48 kombinasyon satırı ile en fazla 48 farklı birey elde edilecek ve bu bireyler çözüm değerlerine göre ikinci aşamada kullanılacaklardır.

4.3.2. Çözüm Geliştirme

Mat-sezgisel algoritmanın ikinci aşaması olan çözüm geliştirme sürecinde başlangıç bireylerden yola çıkarak çözüm uzayının çeşitli bölgelerinin araştırılması ve optimum çözüme olabildiğince yakın çözüm bireylerinin elde edilmesi hedeflenmektedir.

Bu amaçla geliştirilen çözüm geliştirme prosedürünün akış şeması ve sözde kodu sırasıyla Şekil 4.19 ve Şekil 4.20'de verilmektedir. Akış şemasında ve sözde kodda kullanılan kısaltmaların açıklamaları şu şekildedir;

- ns : Nesil sayacı
- gb : Geliştirilecek başlangıç bireyi indisi
- GBK : Geliştirilecek başlangıç bireyler kümesi
- nib : Nesil içi birey sayacı
- $NBK(gb, ns)$: gb başlangıç bireyinin, ns neslinde bulunan bireyler kümesi
- GBS : Geliştirilecek birey sayısı parametresi
- BBS : İlk aşamada oluşan başlangıç bireylerin sayısı
- YBK : Yeni bireyler kümesi
- C : Araç kapasitesi
- CS : Çevrim süresi
- EBS : Elitist yöntemle seçilecek birey sayısı
- $RÇB$: Rulet çemberi ile seçilecek birey sayısı
- RB : Tamamen rasgele seçilecek birey sayısı
- MNS : Maksimum nesil sayısı
- MS : Maksimum geliştirme süresi
- RAO : Rotalar arası aktarma operatörü
- RDO : Rotalar arası değiştirme operatörü



Şekil 4.19. Çözüm geliştirme algoritmasının akış şeması

```
Girdiler: Kroki_verileri, araç_sayısı, başlangıç_bireyler, BBS, GBS, MNS, MS, C, CS, EBS, RÇB, RB
Çıktılar: geliştirilmiş_bireyler
bütün_bireyler ← başlangıç bireyler
for birey_indisi=1: min (BBS, GBS)
  nesil_sayacı ←1
  bireyin_nesilleri (nesil_sayacı) ← başlangıç_bireyler(birey_indisi)
  while nesil_sayacı < MNS
  YBK ← ∅
    for nesil_içi_birey_indisi= 1: nesildeki_birey_sayısı
    araştırılacak_birey ← bireyin_nesilleri (nesil_sayacı) (nesil_içi_birey_indisi)
      for k = 1: araç_sayısı      // araştırılacak_birey'den
        for k' = 1: araç_sayısı     // araştırılacak_birey'den
          if k ≠ k'
            for m = 1: k. rotanın alma-bırakma koridorları sayısı
              k. aracının m. koridorda alma yada bırakma işlemi olan bütün görevlerini k' aracına aktar
              if k' aracının yeni yük miktarı ≤ C
                k ve k' araçlarını Model 3 ile tekrar rotala
                if k' aracının çevrim süresi ≤ CS
                  yeni_birey'i oluştur
                  if yeni_birey, bireyin_nesilleri (nesil_sayacı-1) ve YBK'da yoksa
                    yeni bireyi ve çözüm değerini YBK'ya ve geliştirilmiş_bireyler'e kaydet
                  end if
                end if
              end if
            end for m
          end if
        end for k'
      end for k
                                                                                                  RAO
      for k = 1: araç_sayısı     // araştırılacak_birey'den
        for k' = 1: araç_sayısı     // araştırılacak_birey'den
          if k ≠ k'
            for m = 1: k. rotanın alma-bırakma koridorları sayısı
              for m' = 1: k'. rotanın alma-bırakma koridorları sayısı
                k aracının m. koridorda alma yada bırakma işlemi olan bütün görevlerini k' aracına aktar
                k' aracının m'. koridorda alma yada bırakma işlemi olan bütün görevlerini k aracına aktar
                if k aracının yeni yük miktarı ≤ C & k' aracının yeni yük miktarı ≤ C
                  k ve k' araçlarını Model 3 ile tekrar rotala
                  if k aracının çevrim süresi ≤ CS & k' aracının çevrim süresi ≤ CS
                    yeni_birey'i oluştur
                    if yeni_birey, bireyin_nesilleri (nesil_sayacı-1) ve YBK'da yoksa
                      yeni bireyi ve çözüm değerini YBK'ya ve geliştirilmiş_bireyler'e kaydet
                    end if
                  end if
                end if
              end for m'
            end for m
          end if
        end for k'
      end for k
                                                                                                  RDO
    end for nesil_içi_birey_indisi
    en iyi çözüme sahip EBS kadar bireyi sonraki nesile aktar ve YBK'dan sil
    rulet çemberi ile RÇB kadar bireyi sonraki nesile aktar ve YBK'dan sil
    rasgele RB kadar bireyi sonraki nesile aktar ve YBK'dan sil
    if birey geliştirme algortiması için geçen süre > MS
      "geliştirilmiş_bireyler" kümesinden tekrarlı bireyleri sil
      "geliştirilmiş_bireyler" kümesini çözüm değerlerine göre küçükten büyüğe sırala
      return     // algoritmayı sonlandırır
    end
    nesil_sayacı ← nesil_sayacı+1
  end while
end for birey_indisi
"geliştirilmiş_bireyler" kümesinden tekrarlı bireyleri sil
"geliştirilmiş_bireyler" kümesini çözüm değerlerine göre küçükten büyüğe sırala
```

Şekil 4.20. Çözüm geliştirme algoritmasının sözde kodu

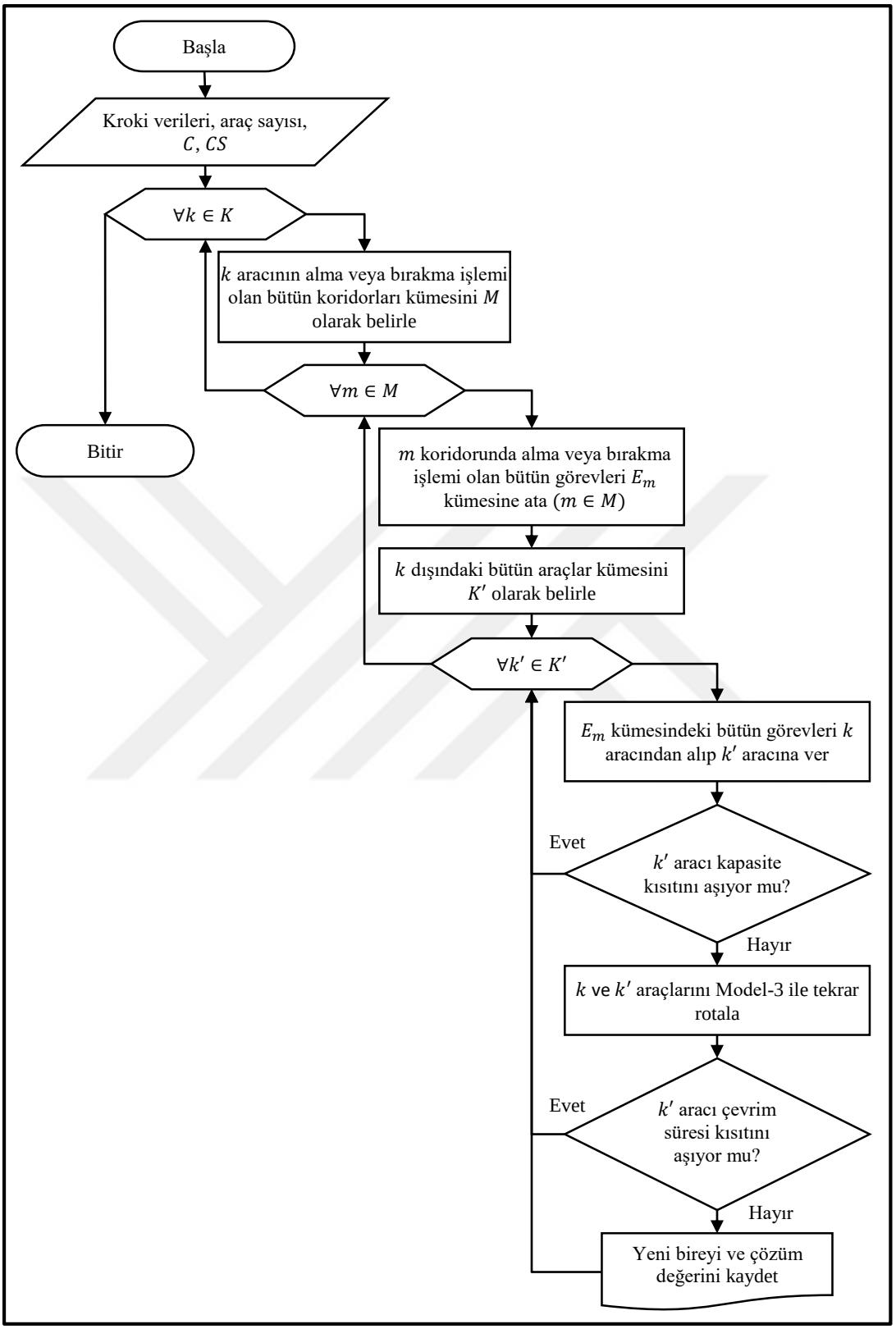
Akış şemasından ve sözde koddan anlaşılacağı üzere geliştirme aşamasında en fazla GBS parametresi kadar başlangıç bireyi kullanılmaktadır. Eğer birinci aşamada üretilen birey sayısı bu parametreden az ise, üretilmiş bütün başlangıç bireyler geliştirme için kullanılacaktır. Çözüm geliştirme aşamasında temel olarak yapılan şey, her bir başlangıç bireyinden ayrı ayrı nesiller boyunca yeni bireyler elde ederek birey kalitesinin nesilden nesile iyileşmesini sağlamaktır. Bu bağlamda algoritmanın genel akışı evrimsel algoritmalara benzemektedir ancak önemli bir nokta olarak bu çalışmada her bir başlangıç bireyi diğerlerinden bağımsız olarak nesiller boyunca yeni bireylere evirilmekte ve çözüm kalitesi nesiller boyunca iyileşmektedir. Böylelikle makul sürelerde optimuma oldukça yakın hatta optimum değerde çözümler elde edilebilmektedir.

Ancak mat-sezgiselin bu aşamasında da birinci aşamada olduğu gibi çekici trenlerin dar koridorlarda çakışma durumları dikkate alınmamaktadır. Çakışmalar göz ardı edilerek Model-3 ile olabildiğince hızlı ve etkili şekilde çok sayıda iyi çözüm değerine sahip bireylerin elde edilmesi ikinci aşamanın temel hedefidir. Çakışma durumları ise üçüncü aşamada ele alınmaktadır.

4.3.2.1. Komşuluk Operatörleri

Çözüm geliştirme algoritmasında çözüm uzayını olabildiğince kapsamlı bir şekilde taramak için güncel nesildeki her bir bireye rotalar arası aktarma operatörü (RAO) ve rotalar arası değiştirme operatörü (RDO) olmak üzere iki adet komşuluk operatörü uygulanmaktadır. Böylelikle yeni nesli oluşturacak bireyleri seçmek için yeni bireylerden oluşan bir havuz elde edilmektedir.

Komşuluk oluşturma operatörleri özellikle gezgin satıcı ve araç rotalama gibi problemlerin çözümüne yönelik olarak geliştirilen çeşitli sezgisel ve metasezgisel algoritmalarda mevcut bireylerden yeni bireyler elde etmek için sıklıkla kullanılmaktadır (Kramer ve diğ., 2015; Aydoğdu ve Özyörük, 2020; Subramanian ve diğ., 2010; Tan ve diğ., 2001). Bu çalışma kapsamında geliştirilen RAO ve RDO, mantık olarak araç rotalama problemlerinde kullanılan operatörlere benzeseler de tamamen SÇDRÇ probleminin yapısına göre özel olarak tasarlanmış operatörlerdir.



Şekil 4.21. Rotalar arası aktarma operatörünün akış şeması

Rotalar arası aktarma operatörünün akış şeması Şekil 4.21'de verilmektedir. Ayrıca Şekil 4.20'de verilen çözüm geliştirme algoritmasının sözde kodunda kırmızı kesikli çizgilerle dikdörtgen içine alınarak gösterilmektedir. Araç rotalama problemleri için geliştirilen aktarma operatörlerinde genellikle bir rotadan belirli bir sayıda müşteri diğer bir rotaya aktarılmaktadır. Bu çalışmada önerilen rotalar arası aktarma operatöründe ise Şekil 4.21'den ve Şekil 4.20'den anlaşılabileceği gibi bir çekici trenin belirli bir koridorunda alma ya da bırakma işlemi olan bütün görevlerinin diğer bir araca verilmesi söz konusudur. Bunun temel nedeni, herhangi bir aracın görev aktarma ile yolunun kısalmasının ancak bu şekilde mümkün olabilmesidir. Örneğin bir k aracının herhangi bir koridorda alma ya da bırakma işlemi olan görevlerinden bazılarının diğer bir araca verilip bazılarının kendisinde kaldığı durumda k aracının yolda geçirdiği sürenin azalması mümkün olmayacaktır.

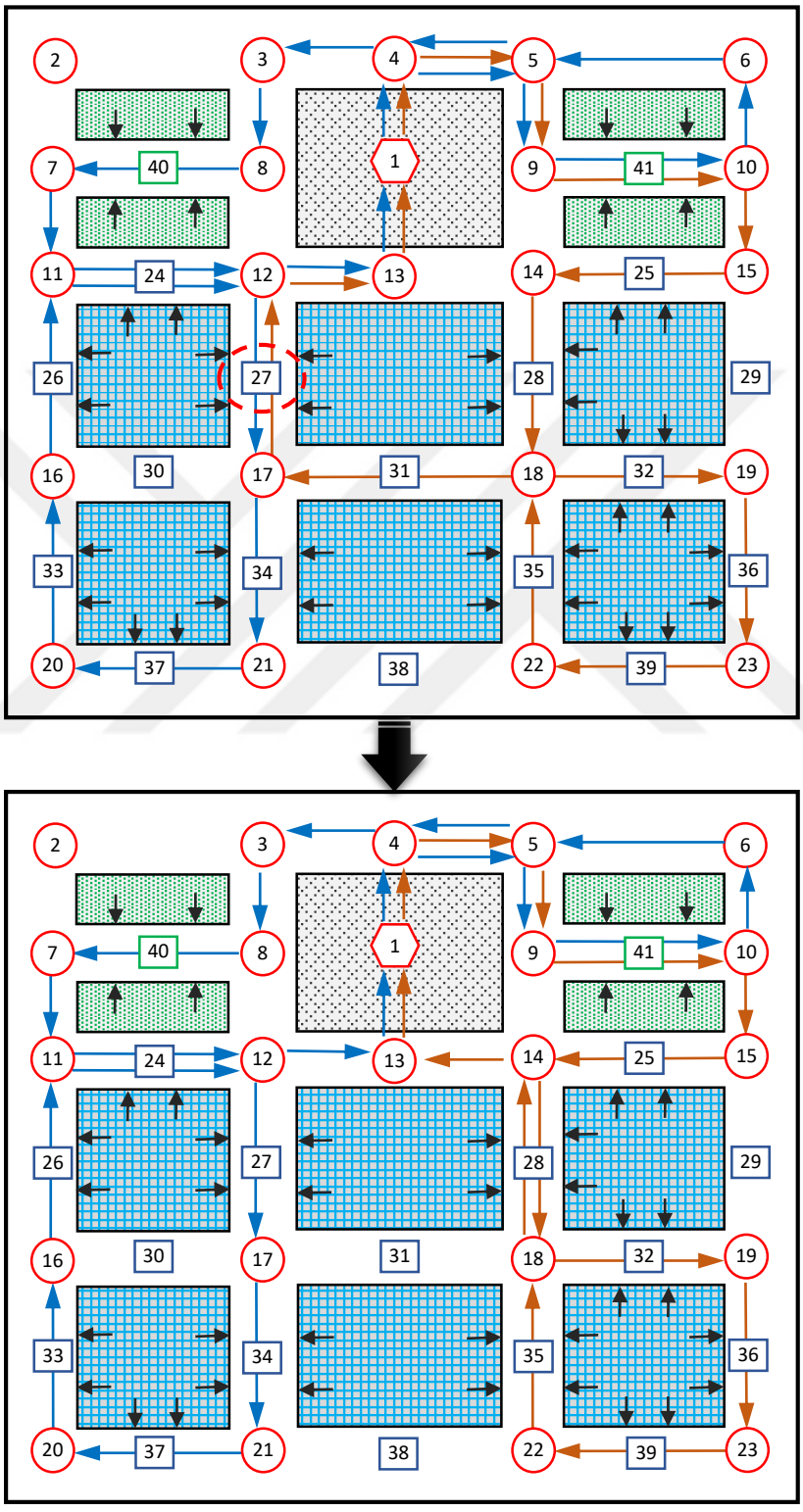
Akış şeması ve sözde kodda görüldüğü gibi, rotalar arası aktarma operatörü uygulanırken görevlerin aktarılacağı aracın kapasite ve çevrim süresi kısıtlarını ihlal edip etmediği kontrol edilmelidir. Bu iki kısıttan herhangi birinin ihlal edilmesi durumunda sağlıklı bir yeni bireyin oluşamayacağı açıktır.

Rotalar arası aktarma operatörünün daha iyi anlaşılabilmesi için Şekil 4.22'de bir örnek gösterilmektedir. Şeklin üst tarafında yer alan krokide iki aracın mevcut rotaları görünmektedir. Her iki araç da 27 numaralı koridorda bırakma işlemi yapmaktadır. Rotalar arası aktarma operatörünün uygulanması sonucunda alttaki krokide görüldüğü gibi, rotası kırmızı çizgilerle gösterilen araç 27 numaralı koridorda bırakma işlemi olan bütün görevlerini rotası mavi çizgilerle gösterilen araca vererek yol tasarrufu sağlamaktadır.

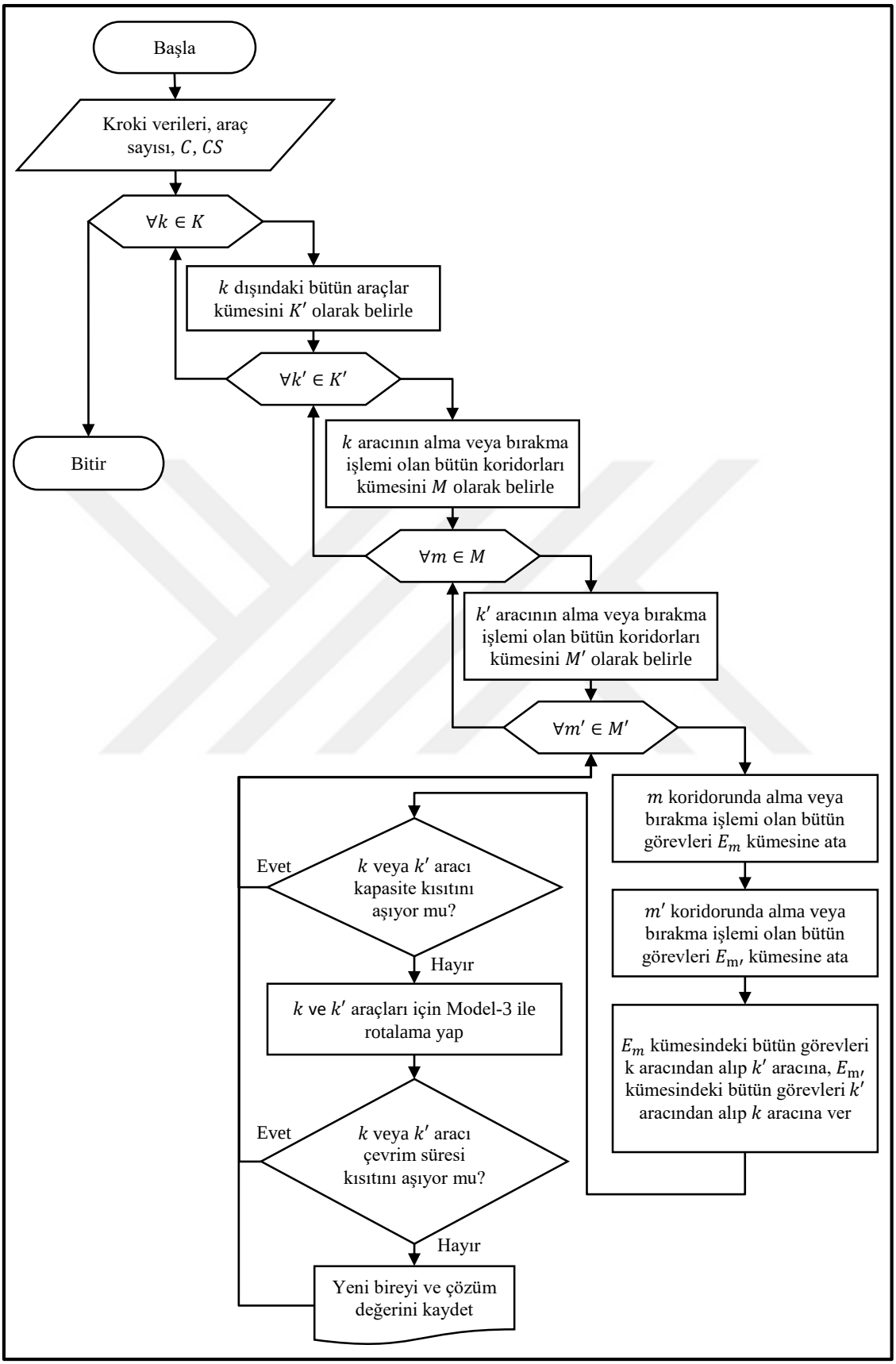
Rotalar arası değiştirme operatörü de aktarma operatörüne benzer bir mantıkla inşa edilmiştir. Akış şeması Şekil 4.23'te verilen ve Şekil 4.20'deki çözüm geliştirme algoritmasının sözde kodunda yeşil kesikli çizgilerle dikdörtgen içine alınarak gösterilen rotalar arası değiştirme operatörünün mantığı araç rotalama problemlerinde yaygın şekilde kullanılan değiştirme operatörlerine benzemektedir.

Genellikle araç rotalama problemlerinde iki aracın bir ya da birkaç müşterisi karşılıklı değiştirilerek komşu çözüm oluşturulmaktadır. Bu çalışmada önerilen RDO ise akış

şeması ve sözde koddan anlaşılabileceği üzere, iki farklı çekici trenin hizmet verdiği birer koridorlarındaki alma bırakma işlemi olan bütün görevlerin karşılıklı değiştirilmesi ile komşu bireyler oluşturmaktadır.

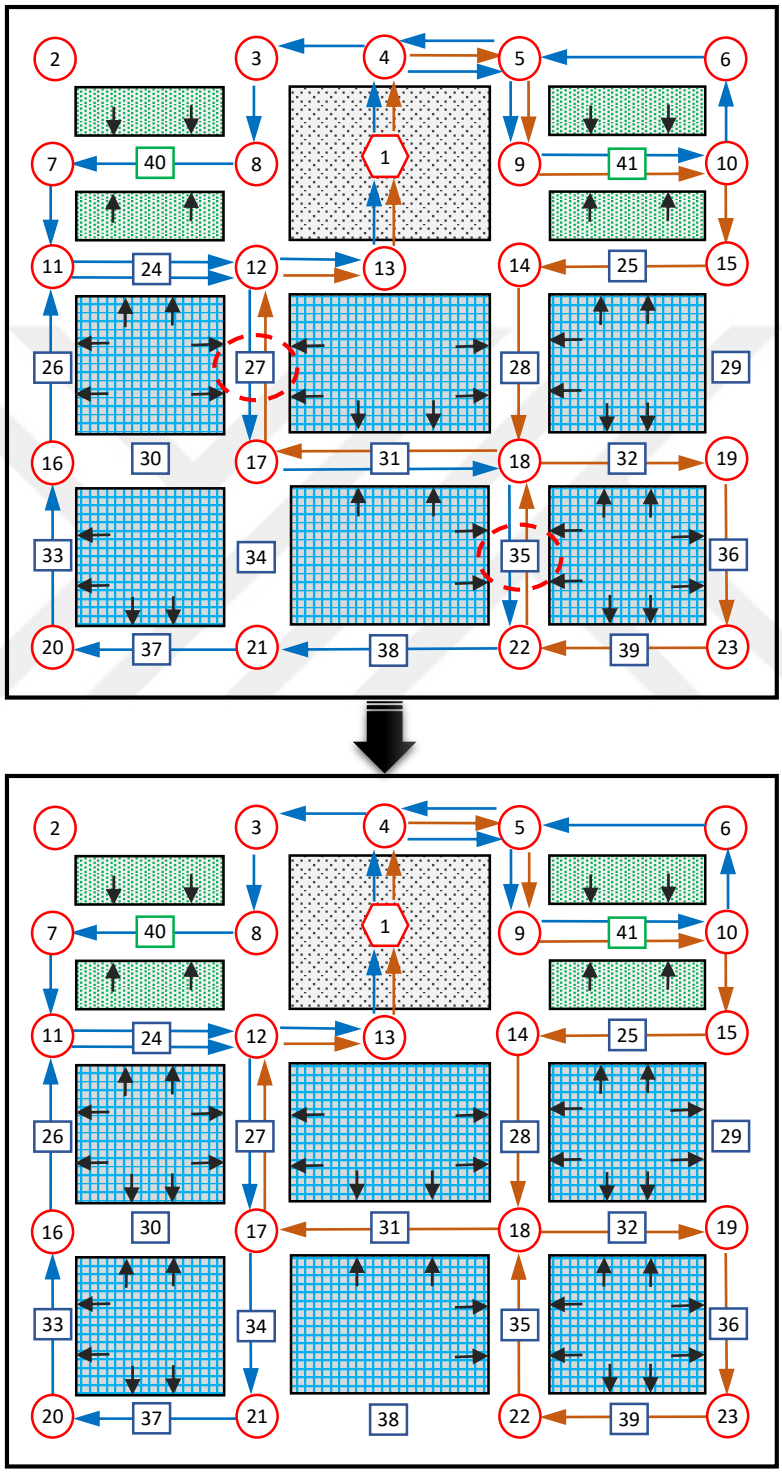


Şekil 4.22. Rotalar arası aktarma operatörüne örnek bir durum



Şekil 4.23. Rotalar arası değiştirme operatörünün akış şeması

Akış şeması ve sözde kodda görüldüğü gibi, rotalar arası değiştirme operatörü uygulanırken, karşılıklı görev değişimlerinden sonra her iki aracın da kapasite ve çevrim süresi kısıtlarını ihlal edip etmediği kontrol edilmelidir. Bu iki kısıttan herhangi birinin bir araç tarafından bile ihlal edilmesi durumunda sağlıklı bir yeni birey oluşmayacaktır.



Şekil 4.24. Rotalar arası aktarma operatörüne örnek bir durum

Şekil 4.24'te gösterilen durum rotalar arası değiştirme operatörünün işleyişine bir örnek olarak verilebilir. Şeklin üst tarafında bulunan krokide iki adet çekici trene ait mevcut durumda rotalar bulunmaktadır. Her iki araç da 27 ve 35 numaralı koridorlarda bırakma işlemi yapmaktadırlar. Rotalar arası değiştirme operatörü uygulanarak mavi rotanın 35 numaralı koridorunda bırakma işlemi olan bütün görevleri kırmızı rotaya, kırmızı rotanın da 27 numaralı koridorda bırakma işlemi olan bütün bireyleri mavi rotaya verilmektedir. Böylelikle şeklin alt tarafında bulunan krokide görüldüğü gibi mavi rotalı araç yol tasarrufu yapmaktadır. Kırmızı rotalı araç ise 31 nolu koridora hizmet vermesi gerektiği için yol tasarrufu yapamamıştır. Araçların toplam yolda geçen süresine bakıldığında ise tasarruf edildiği görülmektedir.

4.3.2.2. Birey Seçimi

Akış şeması Şekil 4.19'da, sözde kodu ise Şekil 4.20'de verilen çözüm geliştirme algoritmasının en önemli noktalarından bir tanesi de yeni nesile aktarılacak bireylerin seçilmesi sürecidir. Akış şemasında ve sözde kodda görüldüğü gibi komşuluk operatörleriyle elde edilen yeni bireyler YBK'ya kaydedilmekte ve sonraki nesile aktarılacak bireyler bu kümeden sırayla üç şekilde seçilmektedir. İlk olarak elitist yaklaşımla en iyi çözüm değerine sahip olan EBS kadar birey alınarak yeni nesile aktarılmakta ve YBK'dan silinmektedir. Daha sonra kalan bireyler arasından RÇB kadar birey rulet çemberi ile seçilerek sonraki nesile aktarılmakta ve YBK'dan silinmektedir. Son olarak da RB kadar birey, YBK'da kalan bireyler arasından tamamen rasgele seçilmekte ve yeni nesile aktarılmaktadır. Bu durumda algoritma boyunca sonraki nesillere aktarılacak toplam popülasyon büyüklüğü (PopB) şu denklemle ifade edilebilir.

$$PopB = EBS + RÇB + RB \tag{4.70}$$

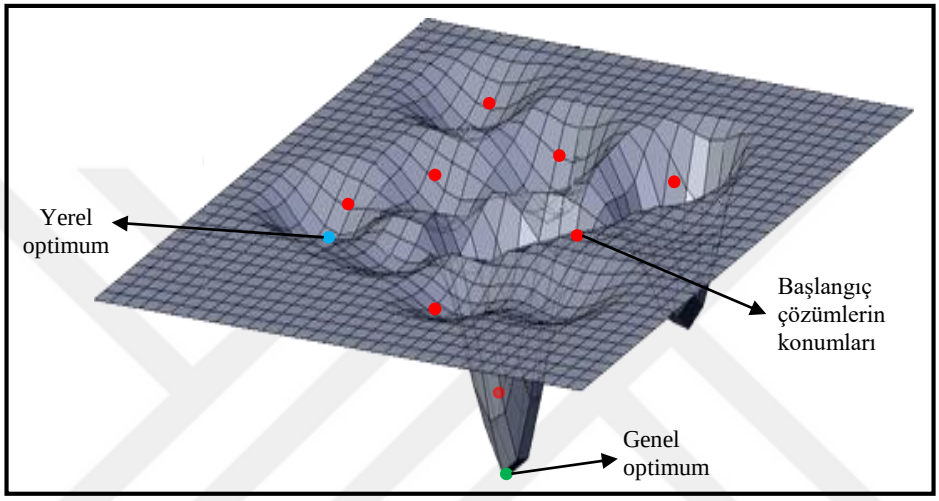
Bu çalışmada kullanılan rulet çemberi yöntemi, genetik algoritma ve diğer evrimsel algoritmalarda kullanılan klasik rulet çemberi seçim yöntemidir. Buna göre her bireyin çözüm değeri ne kadar iyiyse seçilme şansı da o kadar yüksek olmaktadır.

Daha formal olarak ifade etmek gerekirse; mevcut durumda YBK'yı oluşturan bireyler kümesi $YBK = \{b_1, b_2, b_3 \ \ldots \ b_n\}$ olarak kabul edilirse ve b_i bireyinin çözüm değeri $f(b_i)$

ile gösterilirse, amaç fonksiyonu minimizasyon yönünde olduğundan bu durumda b_i bireyinin seçilme olasılığı $\left(P(b_i)\right)$ aşağıda verilen denklem ile ifade edilebilir.

$$P(b_i) = \frac{(1/f(b_i))}{1/f(b_1)+1/f(b_2)+1/f(b_3)+\cdots+1/f(b_n)} \tag{4.71}$$

Rulet çemberi ile seçim hakkında daha detaylı bilgi için Cerf (2017) ve Maiti ve Maiti (2008) kaynaklarına başvurulabilir.



Şekil 4.25. Çözüm uzayının temsili gösterimi

SÇDRÇ probleminin çözüm uzayını görselleştirmek için Şekil 4.25'te temsili bir olurlu çözüm bölgesi verilmiştir. Şekil üzerinde yerel optimum, genel optimum ve başlangıç çözüm oluşturma algoritmasıyla elde edilen başlangıç çözümlerin örnek konumları gösterilmiştir. Bu bölümde anlatılan mat-sezgisel algoritma ile şekilde görüldüğü gibi çözüm bölgesinin çeşitli konumlarında bulunan başlangıç çözümlerden ayrı ayrı yola çıkarak genel optimum noktası aranmaktadır.

Algoritmanın çözüm geliştirme aşamasında kullanılan üç aşamalı ve sıralı birey seçim prosedürü ile yeni nesile aktarılacak bireyler seçilirken düşük uygunluk değerine sahip bireyler daha az seçilme olasılığına sahip olduğundan nesiller ilerledikçe birey kalitesi artmaktadır. Bununla birlikte rulet çemberi ve tamamen rasgele seçim yöntemleri sayesinde çözüm değeri kötü olan bireylerin az da olsa sonraki nesillere aktarılma şansı, Şekil 4.25'te görülen yerel optimum noktalara sıkışmanın engellenmesi ve daha geniş bir alanın taranması sağlanmaktadır.

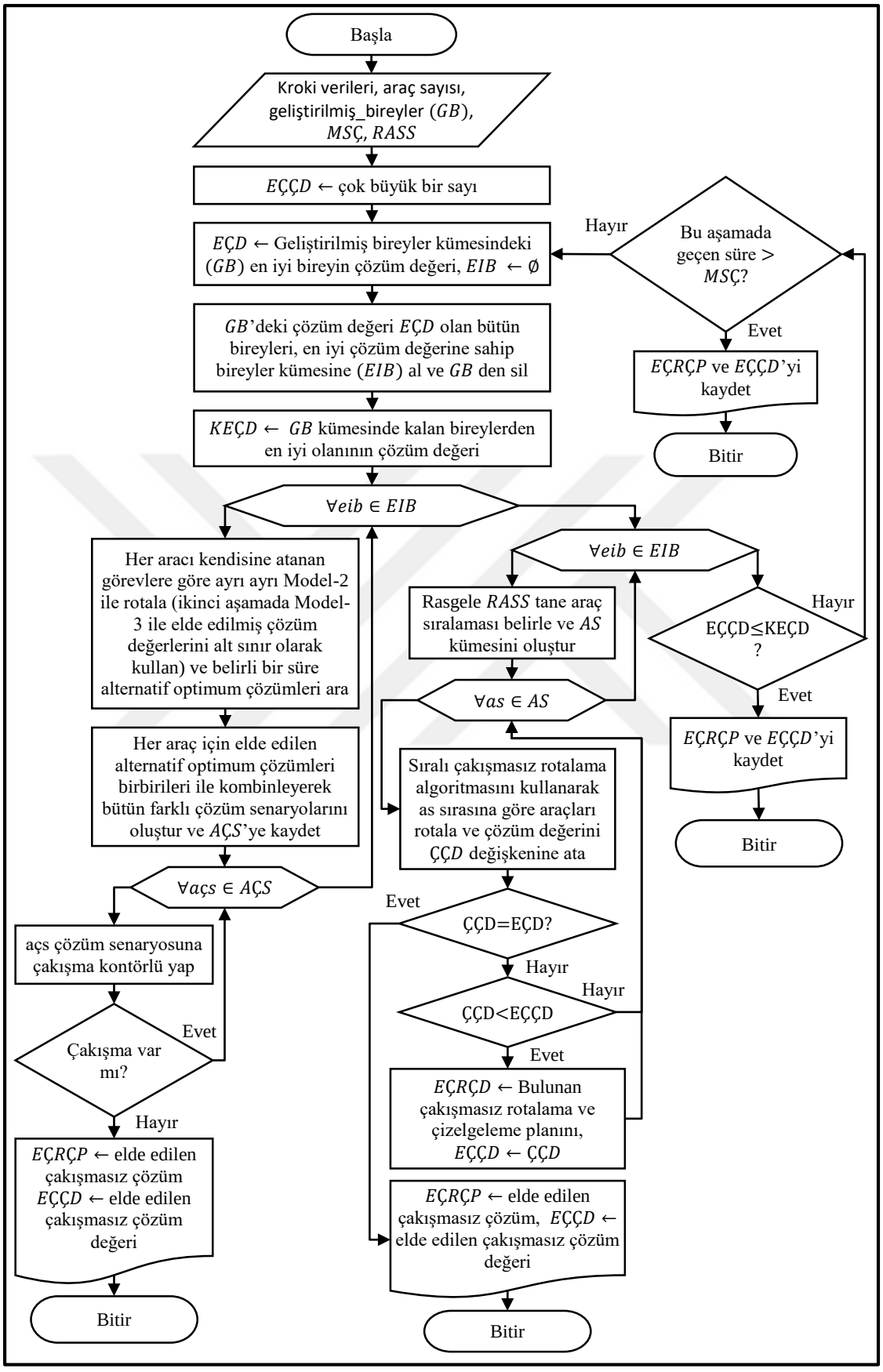
4.3.3. Çakışmasız Çözüm Arama

İlk iki aşama ile çakışmaların göz ardı edildiği durumda çözüm değeri olabildiğince iyi olan kaliteli bireyler elde edildikten sonra üçüncü aşamada bu bireylerin içerisinden mümkün olan en iyi çözüm değerine sahip çakışmasız bir rotalama ve çizelgeleme planı aranmaktadır. Bu amaçla oluşturulan çakışmasız çözüm arama algoritmasının akış şeması ve sözde kodu sırasıyla Şekil 4.26 ve Şekil 4.28'de verilmektedir.

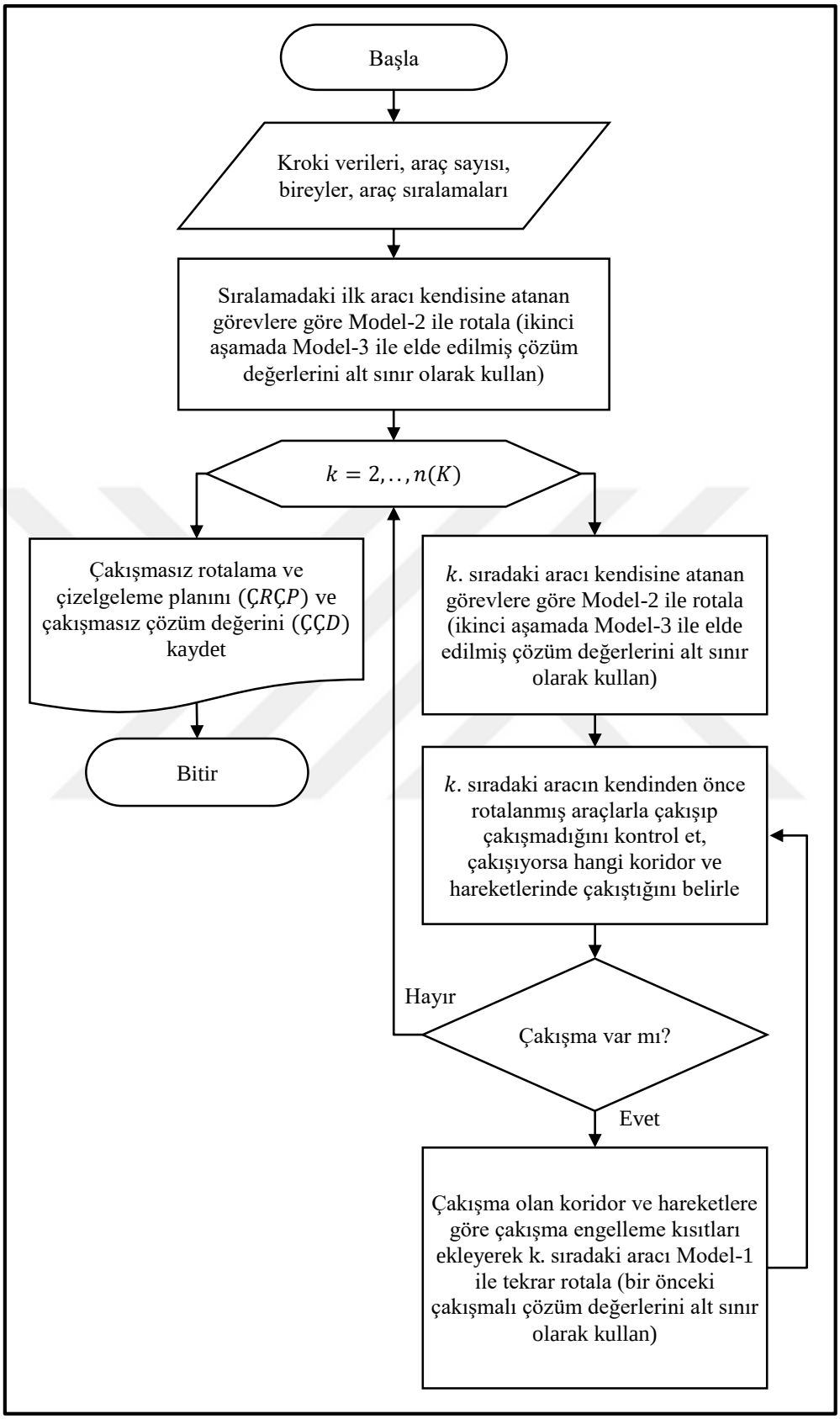
Akış şemasının çok büyümemesi ve daha rahat anlaşılabilmesi amacıyla sıralı çakışmasız rotalama algoritmasının (SÇRA) akış şeması ayrı olarak Şekil 4.27'de verilmiştir. SÇRA, sözde kod içerisinde çakışmasız çözüm arama algoritmasıyla bütünleşik olarak sunulduğu için ayrıca sözde kod oluşturulmamıştır.

Şekil 4.26, Şekil 4.27 ve Şekil 4.28'de kullanılan kısaltmaların açıklamaları şu şekildedir;

- $açs$: Alternatif çözüm senaryosu indisi
- $AÇS$: Alternatif çözümleri içeren küme
- as : Araç sıralaması indisi
- AS : Rasgele seçilmiş araç sıralamaları kümesi
- $RASS$: Rasgele belirlenecek araç sıralamaları sayısı
- GB : Geliştirme sonrası bireyler kümesi
- EIB : En iyi bireyler kümesi
- eib : En iyi bireyler indisi
- $EÇD$: En iyi çözüm değeri
- $KEÇD$: Kalan bireyler içerisinden en iyi olanın çözüm değeri
- $ÇÇD$: Çakışmasız çözüm değeri
- $EÇÇD$: En iyi çakışmasız çözüm değeri
- $EÇRÇP$: En iyi çakışmasız rotalama ve çizelgeleme planı
- $MSÇ$: Çakışmasız çözüm arama aşaması için maksimum süre.



Şekil 4.26. Çakışmasız çözüm arama algoritmasının akış şeması



Şekil 4.27. Sıralı çakışmasız rotalama algoritmasının akış şeması

```
Girdiler: Kroki_verileri, arac_sayısı, geliştirilmiş_bireyler (GB), MSÇ, RASS
Çıktılar: en iyi çakışmasız rotalama ve çizelgeleme planı (EÇRÇP), en iyi çakışmasız çözüm değeri (EÇÇD)
EÇÇD ← çok_büyük_bir_sayı
KEÇD ← 0
while EÇÇD > KEÇD
  çakışmasız_çözüm_bulundu_mu ← 0
  EÇD ← GB'nin en iyi bireyin çözüm değeri
  GB'deki EÇD çözüm değerine sahip bütün bireyleri EIB kümesine al ve GB'den sil
  KEÇD ← GB'de kalan bireylerin en iyi olanının çözüm değeri
  for eib = 1: n(EIB)
    for k = 1: araç_sayısı
      atanan görevlere göre k aracını Model 2 ile rotala (ikinci aşamada Model 3 ile elde edilmiş çözüm değerlerini
      alt sınır olarak kullan) ve alternatif optimum çözümleri ara
    end for k
    araçların alternatif optimum çözümlerini kombinleyerek farklı çözüm senaryolarını oluştur ve AÇS'ye kaydet
    for açs_indisi= 1: n(AÇS)
      çözüm senaryosuna çakışma kontrolü yap
      if çakışma yok
        EÇRÇP ← elde edilen çakışmasız çözüm, EÇÇD ← elde edilen çakışmasız çözüm değeri
        çakışmasız_çözüm_bulundu_mu ← 1
        return    // algoritmayı sonlandırır
      end if
    end for açs_indisi
  end for eib
  if çakışmasız_çözüm_bulundu_mu == 0
    for eib = 1: n(EIB)
      bütün olası araç sıralamaları içerisinden rasgele RASS tane sıralama belirle
      for as=1:RASS
        as. sıralamaya göre ilk aracı Model 2 ile rotala (ikinci aşamada Model 3 ile elde edilmiş çözüm değerlerini alt
        sınır olarak kullan)
        for k = 2: araç_sayısı
          as. araç sıralamasının k. sırasında bulunan aracı Model 2 ile rotala (ikinci aşamada Model 3 ile elde edilmiş
          çözüm değerlerini alt sınır olarak kullan)
          k. sıradaki aracın kendinden önce rotalanmış araçlarla çakışıp çakışmadığını kontrol et, çakışıyorsa hangi
          koridor ve hareketlerinde çakıştığını belirle
          if k. sıradaki araç önceden rotalanmış diğer araçlarla çakışıyorsa
            çakışma_var_mı ← 1
          end
          while çakışma var mı == 1
            çakışma_var_mı ← 0
            sadece çakışma olan koridor ve hareketler için kısıtlar yazarak k. sıradaki aracı Model 1 ile rotala (bir
            önceki çakışmalı çözüm değerlerini alt sınır olarak kullan)
            k. sıradaki araç için tekrar çakışma kontrolü yap, çakışıyorsa hangi koridor ve hareketlerinde çakıştığını
            belirle
            if çakışma varsa
              çakışma_var_mı=1
            end
          end while
        end for k
        ÇRÇP ← elde edilen çakışmasız çözüm, ÇÇD ← elde edilen çakışmasız çözüm değeri
        if ÇÇD = EÇD
          EÇRÇP ← elde edilen çakışmasız çözüm, EÇÇD ← ÇÇD
          return    // algoritmayı sonlandırır
        elseif ÇÇD < EÇÇD
          EÇRÇP ← ÇRÇP, EÇÇD ← ÇÇD
        end
      end for as
    end for eib
    if üçüncü aşamada geçen süre>MSÇ
      EÇRÇP ve EÇÇD'yi kaydet
      return    // algoritmayı sonlandırır
    end if
  end if
end while
```

Şekil 4.28. Çakışmasız çözüm arama algoritmasının sözde kodu

Sözde kod ve akış şemalarından anlaşılabileceği üzere, çakışmasız çözüm arama algoritmasında, ikinci aşamada elde edilen geliştirilmiş bireylerin en iyi çözüm değerine sahip olanlarından başlanarak üçüncü aşama için verilen süre (MSÇ) tamamlanana kadar veya daha iyi bir çakışmasız çözüm bulunamayacağı anlaşılana kadar çakışmasız rotalama ve çizelgeleme planı aranmaktadır.

Fabrika içerisinde aynı yol uzunluğuna sahip birçok farklı güzergah bulunabilmesi yada bir koridorda farklı geçişler sırasında hizmet verilebilmesi gibi durumlar nedeniyle SÇDRÇ problemlerinin genellikle birden fazla alternatif optimum çözüme sahip olabildiği Bölüm 4.2'de örneklerle birlikte açıklanmıştır. Mat-sezgisel algoritmanın üçüncü aşamasında çakışmasız çözüm planı ararken problemin bu özelliğinden faydalanılmaktadır. Sözde kod ve akış şemasından görülebileceği gibi öncelikle gelişmiş bireyler (GB) kümesinden en iyi çözüme sahip bireyler EIB kümesine alınmakta ve GB den silinmektedir. Ardından EIB kümesindeki bütün bireyler için her bir araç Model-2 ile rotalanmakta ve alternatif optimum çözümler bulunmaktadır. Daha sonra farklı araçların alternatif optimum rotalarının birbirleri ile kombinasyonları oluşturularak çok sayıda rotalama ve çizelgeleme senaryosu elde edilmektedir. Örneğin 3 araçlı bir fabrika içi tedarik sisteminde birinci araç için 10, ikinci araç için 4 ve üçüncü araç için 6 adet alternatif optimum rota olduğu düşünüldüğünde bu rotaların birbirleri ile kombinasyonları oluşturulduğunda toplamda 240 farklı alternatif rotalama ve çizelgeleme senaryosu elde edilecektir.

Daha sonra bu alternatif çözüm senaryolarına Şekil 4.11'de verilen algoritma ile çakışma kontrolü yapılarak çakışmasız bir plan olup olmadığı araştırılmaktadır. Eğer çakışmasız bir plan bulunabilirse, EIB ve GB kümeleri içerisinde daha iyi bir çözüm olamayacağı için algoritma sonlandırılmaktadır. Çünkü bir birey için çakışmasız çözüm aranırken, çakışmaların göz ardı edildiği durumda bulunan çözüm değerinden daha iyi bir çözüm değerinin elde edilmesi mümkün değildir. En fazla çakışmaların göz ardı edildiği durumdaki kadar iyi bir çözüm elde edilebilecektir.

Eğer EIB kümesindeki bireylerin hiçbirinde alternatif çözüm senaryoları ile çakışmasız çözüm bulunamazsa, sözde kod ve akış şemasında görüldüğü gibi EIB kümesindeki bütün bireyler için rasgele belirlenmiş $RASS$ adet araç sıralaması kullanılarak SÇRA ile çakışmasız çözümler elde edilecektir. Örneğin 4 araçlı bir fabrika içi dağıtım sisteminde

olası bütün araç sıralamaları $4! = 24$ adettir, RASS parametresinin 15 olması durumunda bu araç sıralamarından rasgele 15 tanesi seçilecek ve SÇRA bu sıralamalar için çalıştırılacaktır. Eğer RASS parametresi 30 olarak belirlenmişse bu durumda 24 adet sıralama kombinasyonunun tamamı alınacaktır. Birden çok araç sıralaması ile SÇRA'nın çalıştırılmasının nedeni, farklı sıralamalarda farklı çözüm değerlerinin elde edilebilmesidir. Araç sıralamalarının bir parametre ile sınırlandırılmasının nedeni ise artan araç sayıları için sıralama kombinasyonlarının çok fazla olması ve dolayısıyla bütün kombinasyonlar için çözüm bulmanın çok fazla zaman almasıdır.

SÇRA, Şekil 4.27'de görüldüğü gibi Model-2 ve Model-1'den yararlanılarak ilk aracın tamamen bağımsız olarak rotalanmasını ve ondan sonra gelen her bir aracın kendisinden önce rotalanmış araçlarla çakışma oluşturmayacak şekilde rotalanmasını içermektedir. Bunu yaparken olabildiğince hızlı olması için iteratif çözüm yaklaşımına benzer bir yaklaşım kullanılmaktadır. Her bir araç öncelikle Model-2 ile rotalanmakta ve diğer rotalanmış araçlarla çakışıp çakışmadığı, eğer çakışıyorsa hangi koridorlarda ve hangi hareketlerde çakıştığı tespit edilmektedir. Çakışma varsa, Model-1'e sadece çakışma olan koridorlar ve hareketler için kısıt eklenerek araç tekrar rotalanmakta ve tekrar çakışma olup olmadığı tekrar kontrol edilmektedir. Böylelikle her bir araç sıralaması için çakışmasız bir rotalama ve çizelgeleme planı ile çakışmasız çözüm değeri elde edilmektedir.

Bu şekilde bütün EIB kümesi için elde edilen en iyi çakışmasız çözüm değeri GB'de kalan bireylerin en iyisinin çözüm değerinden daha iyiyse algoritma sonlandırılmaktadır. Bunun nedeni, GB'deki bireylerden daha iyi bir çözüm değerinin elde edilemeyecek olmasıdır. Diğer taraftan EIB kümesi için elde edilen en iyi çakışmasız çözüm değeri GB'de kalan bireylerin en iyisinin çözüm değerinden daha kötüyse bu durumda bulunan en iyi çakışmasız rotalama ve çizelgeleme planı ile çakışmasız çözüm değeri saklanacak ve EIB kümesi silinecektir. Daha sonra GB'deki en iyi çözüme sahip bireyler tekrar EIB kümesine aktarılarak sonraki döngüye geçilecektir.

Böylece çakışmasız çözüm arama algoritmasının akış şeması ve sözde kodunda gösterildiği gibi daha iyi bir çözüm elde edilemeyeceğine karar verildiğinde ya da üçüncü aşama için belirlenen maksimum süre dolduğunda algoritma sonlandırılacak ve elde edilen en iyi çakışmasız çözüm planı ile çakışmasız çözüm değeri kaydedilecektir.

5. TEST PROBLEMLERİ

Çalışma kapsamında SÇDRÇ problemi için geliştirilen çözüm yaklaşımlarının performansları bir dizi test problemi çözülerek test edilmiştir. Literatürde SÇDRÇ probleminin yapısına uygun test problemleri bulunamadığı için bir sonraki bölümde anlatıldığı şekilde belirli kriterlere bağlı olarak bir dizi test problemi üretilmiştir. Her bir test problemi belirli bir süre üç yaklaşımla da çözülmüş ve sonuçlar birbirleri ile karşılaştırmalı olarak raporlanmıştır. Bu ana başlık altında ilerleyen bölümlerde ilk olarak test problemlerinin oluşturulması süreci ayrıntılı şekilde anlatılacak daha sonra bilgisayar koşum sonuçları ile elde edilen bulgular sunularak tartışılacaktır.

5.1. Test Problemlerinin Oluşturulması

SÇDRÇ problemi, hücresel imalat yerleşimine sahip ve TZÜ'yü benimsemiş üretim ortamlarında çekici trenlerin çakışmasız bir şekilde rotalanmasını, çizelgelenmesini ve yüklenmesini ele almaktadır. Literatürde özellikle bu şekilde bir üretim ortamı için oluşturulmuş test problemi bulunamadığı için bu çalışmada test problemleri gerçek hayat problemlerinden esinlenilerek bir dizi parametreye göre rassal olarak üretilmiştir. Problem boyutunu değiştirdiği düşünülen parametreler; kroki büyüklüğü ve yapısı, dar yol oranı, görev sayısı ve araç sayısı olarak belirlenmiştir. Tablo 5.1'de test problemleri üretilirken kroki tipine göre kullanılan parametrelerin seviyeleri verilmektedir.

Tablo 5.1. Test problemleri oluşturmak için belirlenen parametreler ve seviyeleri

Krokiler	**Parametreler**	**Seviyeler**		
		Seviye 1	**Seviye 2**	**Seviye 3**
Küçük (K)	Dar yol oranı	25%	50%	
	Görev sayısı	25	50	
	Araç sayısı	2	3	4
Orta dikdörtgen (OD)	Dar yol oranı	25%	50%	
	Görev sayısı	50	100	
	Araç sayısı	3	4	5
Orta kare (OK)	Dar yol oranı	25%	50%	
	Görev sayısı	50	100	
	Araç sayısı	3	4	5
Büyük dikdörtgen (BD)	Dar yol oranı	25%	50%	
	Görev sayısı	100	200	
	Araç sayısı	5	6	7
Büyük kare (BK)	Dar yol oranı	25%	50%	
	Görev sayısı	100	200	
	Araç sayısı	5	6	7

Kroki büyüklüğü ve yapısı parametresi için tabloda görüldüğü gibi, küçük boyutlu (K), orta boyutlu - dikdörtgen formlu (OD), orta boyutlu - kare formlu (OK), büyük boyutlu – dikdörtgen formlu (BD) ve büyük boyutlu - kare formlu (BK) olmak üzere 5 farklı kroki tasarlanmıştır. Oluşturulan krokiler Ek A bölümünde sırasıyla Şekil A.1, Şekil A.2, Şekil A.3, Şekil A.4 ve Şekil A.5'te verilmektedir. Üretim hücrelerinin ve süpermarket raflarının boyutları krokiler üzerinde metre cinsinden belirtilmiştir.

Gerçek hayat problemlerinden yola çıkarak, bütün krokilerde tedarik başlangıç alanından çekici trenlerin krokinin üst tarafına doğru çıkış yaptığı ve bunun tam tersi yönden gelerek başlangıç alanına giriş yaptığı varsayılmaktadır. Krokilerin büyüklükleri ile ilgili olarak kapladıkları alan ve içerdikleri koridor sayıları Tablo 5.2'de verilmektedir.

Tablo 5.2. Krokilerin alanları ve koridor sayıları

Kroki	**Kroki alanı (m^2)**	**Koridor sayısı**
Küçük	900	66
Orta dikdörtgen	1650	96
Orta kare	1800	98
Büyük dikdörtgen	6000	220
Büyük kare	5600	194

Hücresel imalat yerleşimine sahip olan gerçek bir üretim ortamının krokisine göre ufak sayılabilecek olan küçük boyutlu kroki, özellikle birinci çözüm yaklaşımı ile ikinci çözüm yaklaşımını karşılaştırmak için oluşturulmuştur. Orta ve büyük boyutlu krokiler ise gerçek fabrika boyutlarını temsil edecek şekilde tasarlanmıştır. Özellikle büyük boyutlu krokilerin gerçek hayat problemlerinde gözlemlenen fabrika krokilerinden daha büyük olacak şekilde oluşturulmasına dikkat edilmiştir. Orta ve büyük boyutta hem dikdörtgen hem de kare formlu krokilerin oluşturulmasının nedeni; özellikle mat-sezgisel algoritmanın bu şekilde farklı yapılardaki krokilerde de iyi çözümler üretip üretemediğini görmek içindir.

Dar yol oranı parametresi, tablodan anlaşılabileceği gibi bütün krokiler için 25% ve 50% olmak üzere iki seviyeli olarak belirlenmiştir. Gerçek hayat uygulamalarına dair yapılan gözlemler neticesinde zaman içerisinde üretilen ürünler ya da değişen üretim teknolojileri sonucunda fabrika içindeki koridorların en çok yarısı kadarının iki çekici trenin

geçemeyeceği şekilde daralabileceği görülmüş ve anlaşılmıştır. Bu nedenle bu çalışmada test problemleri geçekçi olması açısından 25% veya 50% dar yol oranına sahip olacak şekilde üretilmiştir. Her bir test probleminde hangi koridorların dar olacağı tamamen rassal olarak belirlenmiştir.

Görev sayısı parametresi ise kroki büyüklüğüne göre farklı sayılarda olacak şekilde iki seviyeli olarak belirlenmiştir. Tabloda görüldüğü gibi seviyeler; küçük boyutlu kroki kullanılarak üretilecek problemler için 25 ve 50 adet, orta boyutlu krokiler kullanılarak üretilecek problemler için 50 ve 100 adet, büyük boyutlu krokiler kullanılarak üretilecek problemler için ise 100 ve 200 adet olarak alınmıştır. Büyük krokiler için belirlenen 200 adetlik görev sayısının ise karşılaşılan gerçek hayat problemlerinden daha fazla olmasına dikkat edilmiştir.

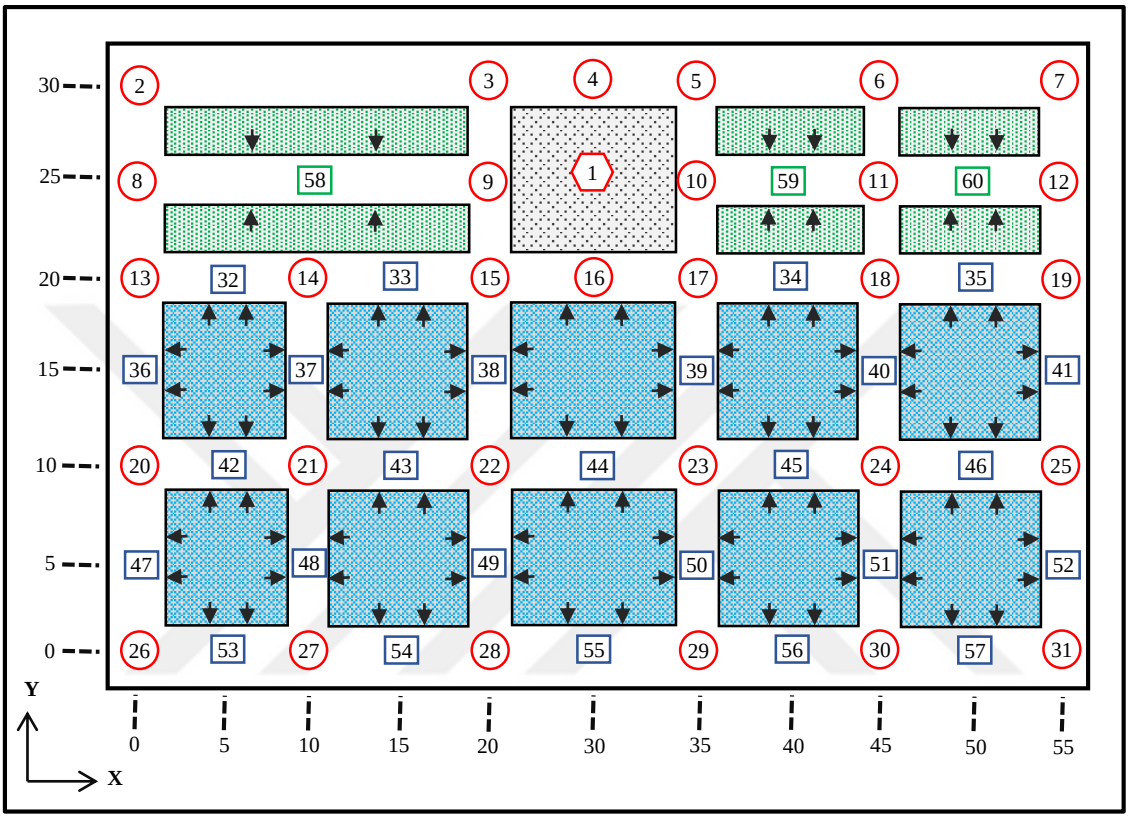
Görevler üretilirken öncelikle her bir görevin bırakma koridoru için imalat hücrelerinin kenarlarında bulunan koridorlardan bir tanesi tamamen rasgele olarak seçilmiştir. Daha sonra görevin alma koridoru belirlenirken; X koordinatı ilgili görev için belirlenen bırakma koridorunun X koordinatına en fazla MES parametresi kadar uzaklıkta olan süpermarket raflarının bulunduğu koridorlardan bir tanesi rasgele olarak seçilmiştir. MES parametresi aşağıda verilen denklem ile hesaplanmaktadır. $MES\ çarpanı$; orta boyutlu krokilerde 2, büyük boyutlu krokilerde ise 3 olarak belirlenmiştir.

$$MES = \frac{Krokinin\ X\ ekseninde\ (yatay\ olarak) uzunluğu}{MES\ çarpanı} \tag{5.1}$$

Görevlerin alma koridorlarının, bırakma koridorları göz önünde bulundurularak belirlenmesinin nedeni daha gerçekçi test problemleri elde etmektir. Örneğin Şekil F'de verilen orta büyüklükte ve yatay formlu kroki ele alındığında, MES parametresi 27,5 olarak elde edilmektedir. Bu kroki üzerinden üretilecek bir görevin bırakma koridoru 52 olarak seçildiği varsayılırsa görevin alma koridoru sadece 59 veya 60 numaralı koridorlardan biri olarak seçilebilecek, 58 numaralı koridor seçilemeyecektir.

Gerçek uygulamalara bakıldığında da planlamacılar tarafından herhangi bir görevin alma ve bırakma koridorlarının birbirine olabildiğince yakın tutulmaya çalışıldığı görülmektedir. Bunun temel nedeni fabrika içi taşıma maliyetlerinin olabildiğince azaltılmak istenmesidir. Bu nedenle Şekil 5.1'deki kroki üzerinde bırakma koridoru 52

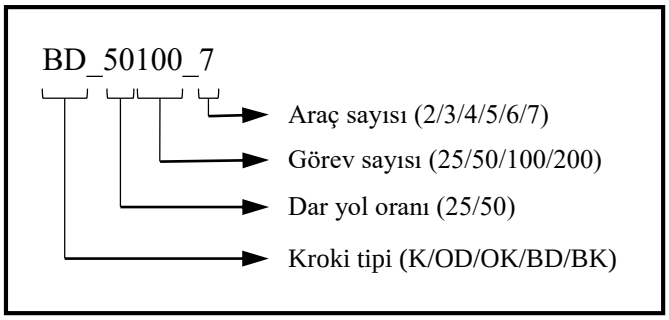
olan bir görevin alma koridorunun 58 olması beklenmeyecek bir durumdur. Bu düşünce özellikle büyük boyutlu krokilerde daha fazla ön plana çıkmaktadır. Bırakma ve alma koridorları belirlenen görevlerin kutu cinsinden miktarları ise [1 4] aralığında düzgün dağılıma göre rasgele üretilmiştir.



Şekil 5.1. Koordinat eksenleri verilmiş yatay boyutlu dikdörtgen formlu kroki

Test problemleri oluşturulurken dikkate alınan diğer bir parametre ise araç sayısıdır. Bu parametre Tablo 5.1'de görüldüğü gibi her bir kroki için farklı olacak şekilde üçer seviyeli olarak belirlenmiştir. Gerçek hayat problemlerinde en fazla 6 adet çekici trenin bir arada kullanıldığı gözlemlenmiştir. Bu nedenle, önerilen mat-sezgisel algoritmanın gerçek hayat problemlerini etkili bir şekilde çözüp çözemeyeceğini anlamak için bu çalışmada büyük boyutlu krokiler için 7 araca kadar test problemleri oluşturulmuştur.

Yukarıda anlatılan parametrelere bağlı olarak her bir kroki için 12 olmak üzere toplamda 60 adet test problemi üretilmiştir. Test problemleri Şekil 5.2'de gösterildiği gibi bir dizi harf ve sayılarla kodlanarak ifade edilmektedir.



Şekil 5.2. Test problemlerinin kodlanması

Küçük kroki kullanılarak oluşturulan test problemlerinin kodu ve bu problemlere ait parametreler örnek olarak Tablo 5.3'de verilmektedir.

Tablo 5.3. Küçük kroki ile oluşturulan deneme problemlerinin parametrelerinin gösterimi

Sıra no	Problem kodu	Kroki	Dar yol oranı	Görev sayısı	Araç sayısı
1	K_2525_2	Küçük	25%	25	2
2	K_2525_3	Küçük	25%	25	3
3	K_2525_4	Küçük	25%	25	4
4	K_2550_2	Küçük	25%	50	2
5	K_2550_3	Küçük	25%	50	3
6	K_2550_4	Küçük	25%	50	4
7	K_5025_2	Küçük	50%	25	2
8	K_5025_3	Küçük	50%	25	3
9	K_5025_4	Küçük	50%	25	4
10	K_5050_2	Küçük	50%	50	2
11	K_5050_3	Küçük	50%	50	3
12	K_5050_4	Küçük	50%	50	4

Test problemlerinin her biri için araç kapasiteleri Denklem 5.2'de verildiği gibi görevlerin çevrim başına kutu cinsinden miktarlarının toplamına 6 eklenip araç sayısına bölünmesi ile elde edilen sayının bir üst tamsayıya yuvarlanması ile hesaplanmıştır. Rasgele üretilen test problemlerinin her biri için araç sayısına ve toplam görev miktarlarına bağlı olarak araç kapasitelerinin bu şekilde standart bir formülle belirlenmesinin daha uygun olacağı düşünülmüştür.

$$C = \left\lceil \left(\left(\sum_{e \in E} p_e \right) + 6 \right) / K \right\rceil \tag{5.2}$$

Çevrim süresi ise hem kutu alma bırakma aşamalarında hem de yolda geçen süreyi içerdiği için Denklem 5.3'te görüldüğü gibi iki bileşenin toplamı olarak hesaplanmıştır. Denklemde, bütün görevler için toplam kutu alma bırakma süresinin araç sayısına bölünmesi ile çevrim süresinin ilk kısmı hesaplanmaktadır. İkinci kısmı ise krokideki bütün koridorların (tek yön olarak) uzunlukları toplamının $CS\ araç\ böleni$ parametresine bölünmesi ile hesaplanmaktadır.

$$CS = \left\lceil \frac{(\sum_{e \in E} p_e)}{K} \right\rceil * (BBS + ABS) + \frac{Bütün\ koridor\ uzunlukları\ toplamı}{CS\ araç\ böleni} \tag{5.3}$$

$CS\ araç\ böleni$ parametresi, araç sayısına bağlı olarak Tablo 5.4'de görüldüğü gibi değişmektedir.

Tablo 5.4. Araç sayısına bağlı olarak CS araç böleni parametresinin aldığı değerler

Araç sayısı	**CS araç böleni**
2	1,25
3	1,5
4	1,75
5	2
6	2,5
7	3

Çevrim süresinin her bir test problemi için verilen formülle standart bir şekilde belirlenmesinin daha sağlıklı sonuçlar vereceği düşünülmektedir.

5.2. Bilgisayar Koşum Sonuçlarının Analizi ve Tartışma

SÇDRÇ için geliştirilen modeller, iteratif çözüm yaklaşımı ve mat-sezgisel algoritma Matlab R2022b programı kullanılarak kodlanmış ve matematiksel modeller için çözücü olarak Gurobi 10.0.1 programından yararlanılmıştır. Deneme problemleri, Intel(R) Core(TM) i9-9900K 3,60GHz işlemci ve 64GB RAM belleğe sahip Windows 10 işletim sistemli bilgisayar ile çözülmüştür.

Üçüncü çözüm yaklaşımı olan mat-sezgisel algoritma rassal süreçler içerdiği için her bir test problemi bu çözüm yaklaşımı ile 5'er kez çözülmüş ve elde edilen en iyi çözüm değeri ile ortalama değerler ilerleyen sayfalarda tablolarda sunulmuştur. Bilgisayar koşumları için mat-sezgisel algoritmanın parametre değerleri Tablo 5.5'de verildiği şekilde ayarlanmıştır.

Çekici tren operatörlerinin birim kutu alma süresi 20 saniye, birim kutu bırakma süresi ise 15 saniye olarak alınmıştır. Çekici tren hızları 1 metre/saniye olarak belirlenmiştir. İteratif çözüm yaklaşımında Model-2 ile alternatif optimum çözümleri arama süresi, küçük boyutlu kroki ile üretilen test problemleri için 450 saniye, orta ve büyük boyutlu krokiler ile üretilen test problemleri için 5400 saniye olarak belirlenmiştir.

Tablo 5.5. Mat-sezgisel algoritmanın parametre değerleri

Mat-sezgisel algoritma parametreleri	**Parametre değerleri**
Geliştirilecek birey sayısı (GBS)	15
Maksimum nesil sayısı (MNS)	25
Maksimum geliştirme süresi (MS) (Saniye)	39600
Elitist yöntemle seçilecek birey sayısı (EBS)	3
Rulet çemberi ile seçilecek birey sayısı (RÇB)	17
Tamamen rasgele seçilecek birey sayısı (RB)	5
Çakışmasız çözüm arama süresi (MSÇ) (Saniye)	3600
Rasgele belirlenecek araç sıralamaları sayısı (RASS)	20

Test problemleri, üç çözüm yaklaşımının her biriyle maksimum 12 saat süreyle çözülmüştür. Gerçek hayat uygulamalarında, üretim planlarının genellikle aylık periyotlarla değiştiği ve böylece fabrika içi tedarik planlamalarının aylık olarak güncellendiği gözlemlenmiştir. Bu nedenle çözüm süresinin 12 saat olarak belirlenmesinin uygun olacağı düşünülmüştür. Buna paralel olarak Tablo 5.5'de görüldüğü gibi mat-sezgisel algoritmaya ikinci ve üçüncü aşamalar için sırasıyla 11 ve 1 saat olmak üzere toplamda en fazla 12 saatlik maksimum süre verilmiştir. Birinci aşama çok kısa sürdüğü için süre kısıtlaması konulmamıştır.

5.2.1. Küçük Boyutlu Problemler

Küçük boyutlu kroki kullanılarak üretilen 12 adet test probleminin bilgisayar koşum sonuçlarına ilişkin bilgiler Tablo 5.6'da sunulmaktadır. Tablo incelendiğinde problemlerin birinci çözüm yaklaşımı ile çözülebilmesi için gerekli olan sürelerin ikinci ve üçüncü çözüm yaklaşımlarına göre oldukça uzun oldukları görülmektedir. Öyle ki, birinci çözüm yaklaşımı için gereken ortalama süre iteratif çözüm yaklaşıma göre yaklaşık 50 kat, mat-sezgisel çözüm yaklaşımına göre ise yaklaşık 90 kat olarak ortaya çıkmaktadır.

Bununla birlikte K_5025_4 ve K_5050_4 kodlu test problemleri için birinci yaklaşımla 12 saatlik sürede çözüm optimalliğinin kanıtlanamadığı ve bulunan en iyi tam sayılı çözüm değeri ile alt sınır arasında %8 civarında büyük bir fark olduğu görülmektedir. Bu durum, artan araç sayısı, görev sayısı ve dar yol oranının Model-1'deki değişken ve kısıt sayısını dramatik olarak artırdığı için ortaya çıkmaktadır.

İteratif çözüm yaklaşımı ile elde edilen sonuçlar incelendiğinde, problemlerin tamamı için optimum çözüm bulunabildiği görülmektedir. "Çözüm / alt sınır bulunan model" sütununa bakıldığında, test problemlerinin çoğu için Model-2 ile elde edilen alternatif optimum çözümler içerisinden çakışmasız çözüm bulunabildiği, bu nedenle kısıt eklemeli Model-1 ile çözüm yapılmasına gerek kalmadığı anlaşılmaktadır.

Sadece K_5025_4 ve K_5050_4 kodlu problemler için Model-2 ile çakışmasız alternatif optimum çözüm bulunamadığı ve kısıt eklemeli Model-1 ile çözüm yapılarak optimum çakışmasız çözümün elde edilebildiği açık bir şekilde görülmektedir. Bu noktadan hareketle, söz konusu iki problem için çözüm süresinin, aynı görev ve araç sayısı bulunan ancak dar yol oranı %50 yerine %25 olan K_2525_4 ve K_2550_4 kodlu problemlerin çözüm süresine nispeten belirgin şekilde daha fazla olmasının nedeninin artan dar yol oranına bağlı olarak Model-2 ile çakışmasız çözüm bulabilme olasılığının azalması olarak değerlendirilebilir.

Tablo 5.6. Küçük boyutlu kroki kullanılarak üretilen test problemlerinin çözüm sonuçları

Problem kodu	Birinci çözüm yaklaşımı				İkinci çözüm yaklaşımı				Üçüncü çözüm yaklaşımı									
	Çözüm süresi	Alt sınır	En iyi tam sayılı çözüm	Fark (%)	Çözüm süresi	Alt sınır	Optimum çözüm değeri	Çözüm / alt sınır bulunan model	Ort. Çözüm süresi	En iyi başlangıç çözüm değeri ($EIBÇD$)	Başlangıç çözüm değerlerinin ortalaması ($BÇDO$)	En kötü başlangıç çözüm değeri ($EKBÇD$)	Ort. çözüm değeri	En iyi çözüm değeri ($EIÇD$)	($EKBÇD$) iyileştirme oranı	$BÇDO$ iyileştirme oranı (%)	$EIBÇD$ iyileştirme oranı (%)	$EIÇD$ - optimum çözüm değeri farkı (%)
K_2525_2	263	294,00	294	0	54	294	294	Model-2	52	308	321,2	356	294	294	21,09	9,25	4,76	**0**
K_2525_3	2454	364,00	364	0	431	364	364	Model-2	209	388	416,6	484	364	364	32,97	14,45	6,59	**0**
K_2525_4	40189	422,00	422	0	417	422	422	Model-2	245	460	476	480	423	423	13,48	12,53	8,75	**0,24**
K_2550_2	2982	316,00	316	0	270	316	316	Model-2	178	342	359,4	376	316	316	18,99	13,73	8,23	**0**
K_2550_3	30439	356,00	356	0	480	356	356	Model-2	325	388	411,9	436	361,6	356	22,47	15,70	8,99	**0**
K_2550_4	41897	416,00	416	0	495	416	416	Model-2	419	456	500,15	536	418,4	416	28,85	20,23	9,62	**0**
K_5025_2	343	294,00	294	0	54	294	294	Model-2	55	308	321,2	356	294	294	21,09	9,25	4,76	**0**
K_5025_3	9779	364,00	364	0	431	364	364	Model-2	238	388	416,6	484	366	366	32,24	13,83	6,01	**0,55**
K_5025_4	43200	386,06	423	8,73	974	423	423	Model-1	259	460	476	480	423	423	13,48	12,53	8,75	**0**
K_5050_2	11084	316,00	316	0	270	316	316	Model-2	180	342	359,4	376	316	316	18,99	13,73	8,23	**0**
K_5050_3	37866	356,00	356	0	480	356	356	Model-2	335,7	388	411,9	436	362,8	356	22,47	15,70	8,99	**0**
K_5050_4	43216	384,42	418	8,03	1270	418	418	Model-1	423	456	500,15	536	420	418	28,23	19,65	9,09	**0**
Ortalama	**21976**				**468,83**				**243,23**						**22,86**	**14,22**	**7,73**	**0,07**

Test problemlerinin çözüm sonuçlarına ilişkin olarak mat-sezgisel algoritmanın performansını ölçebilmek ve değerlendirebilmek için Tablo 5.6'da görüldüğü gibi bir takım göstergeler hesaplanmış ve sunulmuştur. Bunlardan bir tanesi mat-sezgisel algoritma ile elde edilen en iyi çözüm değeri ($EIÇD$) ve optimum çözüm değeri arasındaki oransal farktır. Bu fark şu şekilde hesaplanmaktadır:

$$EIÇD - Optimum \text{ çözüm değeri } farkı = \frac{(EIÇD - Optimum \text{ çözüm değeri})}{Optimum \text{ çözüm değeri}} \quad (5.4)$$

Bu gösterge, mat-sezgisel algoritma ile bulunan çözüm değerlerinin optimum çözüme ne kadar yaklaştığını ifade etmektedir. Tablo 5.6'da küçük boyutlu kroki ile oluşturulan test problemlerinde bu gösterge ile elde edilen sonuçlara bakıldığında, iteratif çözüm yaklaşımından daha kısa sürelerde bütün problemler için optimum veya optimuma çok yakın çözüm değerleri elde edildiği görülmektedir. Sadece iki problem için optimum değerden %0,24 ve %0,55'lik, çok küçük denilebilecek sapmalar mevcuttur.

Mat-sezgisel algoritmanın performansının değerlendirilmesine ilişkin olarak, en kötü başlangıç çözüm değerini ($EKBÇD$) iyileştirme oranı, başlangıç çözüm değerleri ortalamasını ($BÇDO$) iyileştirme oranı ve en iyi başlangıç çözüm değerini ($EIBÇD$) iyileştirme oranı olmak üzere üç gösterge daha belirlenmiş ve her bir test problemi için hesaplanmıştır. Bu göstergeler sırasıyla aşağıda verilen denklemler kullanılarak hesaplanmaktadır.

$$BÇDO \text{ iyileştirme oranı} = \frac{(BÇDO - EIÇD)}{EIÇD} \quad (5.5)$$

$$EIBÇD \text{ iyileştirme oranı} = \frac{(EIBÇD - EIÇD)}{EIÇD} \quad (5.6)$$

$$EKBÇD \text{ iyileştirme oranı} = \frac{(EKBÇD - EIÇD)}{EIÇD} \quad (5.7)$$

Bu göstergeler mat-sezgisel algoritmanın birinci aşamasında elde edilen bireylerin, rassal süreçler içeren ikinci ve üçüncü aşamasında ne kadar geliştirildiği hakkında bilgi vermektedir. Tablo 5.6'da küçük boyutlu kroki ile üretilen test problemleri için, $EKBÇD$ iyileştirme oranının %13,48 - %32,97 arasında, $BÇDO$ iyileştirme oranının %9,25 - %20,23 arasında ve $EIBÇD$ iyileştirme oranının %4,76 - %9,09 arasında olduğu

görülmektedir. 12 adet test problemi için bu göstergelerin ortalamaları sırasıyla %22,86, %14,22 ve %7,73 olarak elde edilmiştir. Bu değerler daha sonra farklı büyüklükteki test problemlerinde mat-sezgisel algoritmanın performansını analiz etmek için kullanılacaktır. Her bir test problemi için mat-sezgisel algoritma ile tekrarlı koşumlarda elde edilen çözüm değerinin ortalamasını ifade eden "Ort. çözüm değeri" sütununa bakıldığında ortalama değerlerin en iyi çözüm değerlerine eşit veya oldukça yakın olduğu görülmektedir.

Çalışma kapsamında her bir test problemi için elde edilen çakışmasız rotalama, çizelgeleme ve yükleme planına göre çekici trenler için araç kapasitesi kullanım oranları ($AKKO$) hesaplanmış ve değerlendirilmiştir. $AKKO$ hesaplamasında aşağıda verilen denklem kullanılmıştır.

$$AKKO = \frac{Her\ çevrimde\ araca\ yüklenen\ kutu\ miktarı}{Aracın\ kutu\ cinsinden\ kapasitesi} \tag{5.8}$$

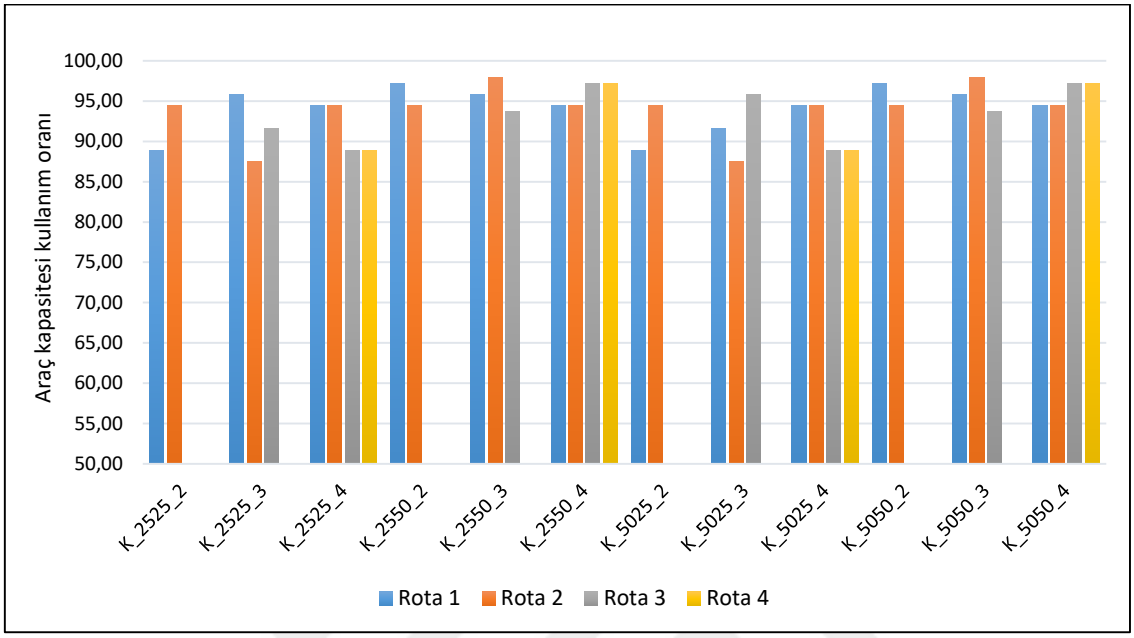
Küçük kroki kullanılarak oluşturulan test problemleri için AKKO değerleri Tablo 5.7'de verilmektedir.

Tablo 5.7. Küçük boyutlu kroki kullanılarak üretilen test problemleri için AKKO değerleri

Problem kodu	**Araç kapasitesi kullanım oranları (%)**			
	Rota 1	**Rota 2**	**Rota 3**	**Rota 4**
K_2525_2	88,89	94,44		
K_2525_3	95,83	87,50	91,67	
K_2525_4	94,44	94,44	88,89	88,89
K_2550_2	97,22	94,44		
K_2550_3	95,83	97,92	93,75	
K_2550_4	94,44	94,44	97,22	97,22
K_5025_2	88,89	94,44		
K_5025_3	91,67	87,50	95,83	
K_5025_4	94,44	94,44	88,89	88,89
K_5050_2	97,22	94,44		
K_5050_3	95,83	97,92	93,75	
K_5050_4	94,44	94,44	97,22	97,22

Tabloda verilen bilgiler, daha rahat anlaşılabilmesi için çubuk grafiği şeklinde Şekil 5.3'te görselleştirilmiştir. Tablo ve şekil incelendiğinde çekici trenlerin kapasite kullanım oranlarının genel olarak yüksek seviyede ve birbirlerine oldukça yakın olduğu dikkat

çekmektedir. Bu noktada test problemlerinde çekici tren kapasitelerinin Denklem (5.2) ile belirlenmesinin de etkisi olduğu söylenebilir.



Şekil 5.3. Küçük boyutlu kroki kullanılarak üretilen test problemleri için AKKO değerleri

Test problemleri için hesaplanan bir diğer gösterge ise kapasite kullanım oranı (KKO) değerleridir. Problem tanımı bölümünde anlatıldığı gibi çekici trenler her çevrim başında tedarik başlangıç noktasından hareket etmekte ve rotaları boyunca kendilerine atanan görevler için kutu alma bırakma işlemlerini yapıp geri tedarik başlangıç bölgesine dönerek bir sonraki çevrimin başlangıcına kadar beklemektedirler. KKO göstergesi çekici trenlerin ve operatörlerin çevrim süresinin ne kadarlık bir bölümünde aktif olarak çalıştığını ifade etmektedir ve Denklem (5.9) ile hesaplanmaktadır.

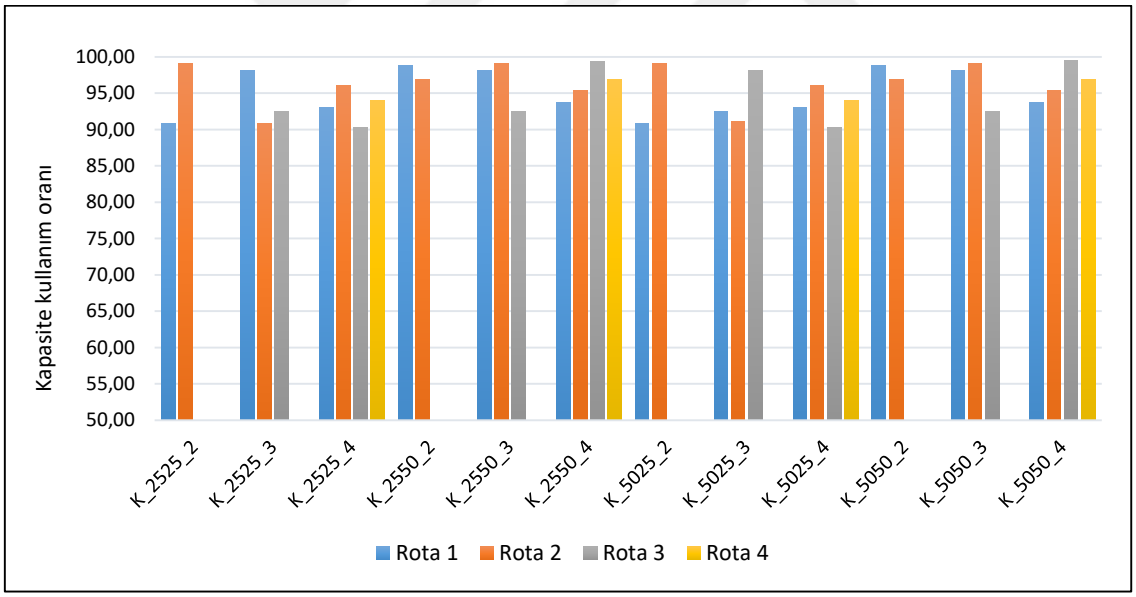
$$KKO = \frac{Çevrim\ başına\ aktif\ çalışma\ süresi}{Çevrim\ süresi} \tag{5.9}$$

Tablo 5.6'da çözüm değerleri verilen test problemleri için hesaplanan kapasite kullanım oranları Tablo 5.8'de verilmektedir. Ayrıca bu değerler çubuk grafiği şeklinde Şekil 5.4'te görselleştirilmektedir.

Tablo 5.8. Küçük boyutlu kroki kullanılarak üretilen test problemleri için KKO değerleri

Problem kodu	Kapasite kullanım oranları (%)			
	Rota 1	Rota 2	Rota 3	Rota 4
K_2525_2	90,88	99,05		
K_2525_3	98,21	90,84	92,42	
K_2525_4	93,06	96,07	90,33	94,02
K_2550_2	98,75	96,85		
K_2550_3	98,21	99,05	92,46	
K_2550_4	93,70	95,32	99,36	96,86
K_5025_2	90,88	99,05		
K_5025_3	92,42	91,05	98,21	
K_5025_4	93,06	96,07	90,33	94,02
K_5050_2	98,75	96,85		
K_5050_3	98,21	99,05	92,46	
K_5050_4	93,70	95,32	99,50	96,86

Tablo ve şekil incelendiğinde bütün test problemleri için KKO'nın %90'nın üzerinde olduğu görülmektedir.



Şekil 5.4. Küçük boyutlu kroki kullanılarak üretilen test problemlerinin çözülmesi ile elde edilen kapasite kullanım oranları

5.2.2. Orta Boyutlu Problemler

Orta boyutlu dikdörtgen formlu kroki kullanılarak oluşturulan 12 adet test probleminin üç çözüm yaklaşımı ile çözülmesi sonucu elde edilen değerler Tablo 5.9'da sunulmaktadır. Tablodan görüldüğü üzere, birinci çözüm yaklaşımı ile hiçbir test

problemine 12 saatlik süre içerisinde optimum çözüm bulunamamıştır. Sadece iki problem (OD_2550_3 ve OD_25100_3) için bulunan en iyi tam sayılı çözüm değerleri iteratif yaklaşımla elde edilen optimum çözüm değerlerine eşittir ancak bu problemlerde bile tam sayılı çözüm ile alt sınır değeri arasındaki farkların oldukça büyük olduğu (%13,45 ve %16,01) görülmektedir. Dolayısıyla bu problemlerde bile birinci çözüm yaklaşımı ile optimalliğin kanıtlanması oldukça uzun süreler gerektirecektir. Test problemlerinin bir kısmı için ise Model-1 ile hiç tam sayılı çözüm bulunamadığı görülmektedir. Tam sayılı çözüm bulunabilmiş olsa bile bu çözümün optimum olduğunun kanıtlanması çok uzun süreceğinden birinci çözüm yaklaşımının orta boyutlu - dikdörtgen şekilli krokiyle oluşturulan bu test problemleri için kullanışlı olmadığı rahatlıkla söylenebilir.

İkinci çözüm yaklaşımı ile elde edilen sonuçlara bakıldığında sadece bir problem (OD_50100_5) dışında bütün test problemleri için makul denilebilecek sürelerde optimum çözüm değerlerinin elde edilebildiği görülmektedir. OD_50100_5 problemi için ise alt sınırın 688 olarak elde edildiği ancak bu değerinin çakışmasız optimum çözüm olduğunun kanıtlanamadığı görülmektedir. Bu test probleminde Model-2 ile elde edilen alternatif optimum çözümler arasından çakışmasız bir çözüm bulunamadığı için kısıt eklemeli Model-1 ile çözüm aranmaya çalışılmış ancak belirlenen süre içerisinde kısıt eklemeli Model-1 çözümünü tamamlayamamıştır. Bu nedenle Model-2 ile elde edilen çözümün alt sınır olduğu, yani bulunabilecek çakışmasız optimum çözümün en çok bu çözüm değeri kadar iyi olabileceği ifade edilmiştir.

İkinci çözüm yaklaşımı ile elde edilen çözüm sürelerine bakıldığında genelde 2 saatlik bir sürenin altında optimum çözümün bulunabildiği görülmektedir. Ancak bazı problemler (OD_25100_5 ve OD_5050_5) için çözüm süresinin çok daha uzun olduğu göze çarpmaktadır. Bunun sebebine bakıldığında, OD_25100_5 problemi için Model-3 ile çözüm bulma aşamasında alt tur eleme sürecinin normalden daha uzun sürdüğü, OD_5050_5 problemi için ise Model-2 ile bulunan alternatif optimum çözümler içerisinden çakışmasız çözüm bulunamadığı için kısıt eklemeli Model-1 ile çözüm yapılmasının gerektiği ve bunun çözüm sürecini uzattığı anlaşılmaktadır.

Tablo 5.9. Orta boyutlu dikdörtgen kroki ile üretilen test problemlerinin çözüm sonuçları

Problem kodu	Birinci çözüm yaklaşımı				İkinci çözüm yaklaşımı				Üçüncü çözüm yaklaşımı									
	Çözüm süresi	Alt sınır	En iyi tam sayılı çözüm	Fark (%)	Çözüm süresi	Alt sınır	Optimum çözüm değeri	Çözüm / alt sınır bulunan model	Ort. Çözüm süresi	En iyi başlangıç çözüm değeri ($EIBÇD$)	Başlangıç çözüm değerlerinin ortalaması ($BÇDO$)	En kötü başlangıç çözüm değeri ($EKBÇD$)	Ort. çözüm değeri	En iyi çözüm değeri ($EIÇD$)	($EKBÇD$) iyileştirme oranı	$BÇDO$ iyileştirme oranı (%)	$EIBÇD$ iyileştirme oranı (%)	$EIÇD$ - optimum çözüm değeri farkı (%)
OD_2550_3	43200	450,07	520	13,45	5443	520	520	Model-2	2678	558	588,2	622	520	520	19,62	13,12	7,31	**0**
OD_2550_4	43200	496,34	600	17,27	5446	598	598	Model-2	3605	632	714,2	796	602	602	32,23	18,64	4,98	**0,67**
OD_2550_5	43200	546,58	670	18,42	5579	668	668	Model-2	5190	748	787,7	856	676	668	28,14	17,92	11,98	**0**
OD_25100_3	43200	460,24	548	16,01	5571	548	548	Model-2	2335	596	646,4	678	548	548	23,72	17,96	8,76	**0**
OD_25100_4	43200	496,30	686	27.65	5933	598	598	Model-2	3280	670	722	776	616,4	610	27,21	18,39	9,84	**2,01**
OD_25100_5	43200	546,82	696	21,43	41802	688	688	Model-2	4186	780	808,3	852	696	688	23,84	17,49	13,37	**0**
OD_5050_3	43200	459,05	522	12,06	5443	520	520	Model-2	2676	558	588,2	622	520	520	19,62	13,12	7,31	**0**
OD_5050_4	43200	474,17	686	30,87	5446	598	598	Model-2	3638	632	714,2	796	602	602	32,23	18,64	4,98	**0,67**
OD_5050_5	43200	529,20	-	-	31418	668	668	Model-1	5814	748	787,7	856	688	688	24,42	14,49	8,72	**2,99**
OD_50100_3	43200	442,27	-	-	5571	548	548	Model-2	2341	596	646,4	678	548	548	23,72	17,96	8,76	**0**
OD_50100_4	43200	492,70	-	-	5933	598	598	Model-2	3294	670	722	776	618	618	25,57	16,86	8,41	**3,34**
OD_50100_5	43200	531,68	-	-	43200	688	-	Model-2	4189	780	808,3	852	701,2	688	23,84	17,49	13,37	**0**
Ortalama	**43200**				**13898,7**				**3602,2**						**25,34**	**16,84**	**8,98**	**0,81**

Tablo 5.9'da üçüncü çözüm yaklaşımı ile elde edilen sonuçlar incelendiğinde, çoğu problem için optimum çözüm değerinin elde edilebildiği, diğer problemler için ise optimum çözümden çok küçük sapmalarla çözüm değerleri bulunabildiği anlaşılmaktadır. Tabloda sunulan 12 adet test problemi için optimum çözüm değerlerinden sapma ortalaması %0,81 olarak hesaplanmıştır. OD_50100_5 kodlu test problemi için mat-sezgisel algoritma ile bulunan en iyi çözüm değerinin iteratif çözüm yaklaşımı ile elde edilen alt sınır değerine eşit olduğu görülmektedir. Bu durumda ilgili problem için alt sınırdan daha küçük değere sahip bir çözüm bulunamayacağı için optimum değerin bulunmuş olduğu ifade edilebilir.

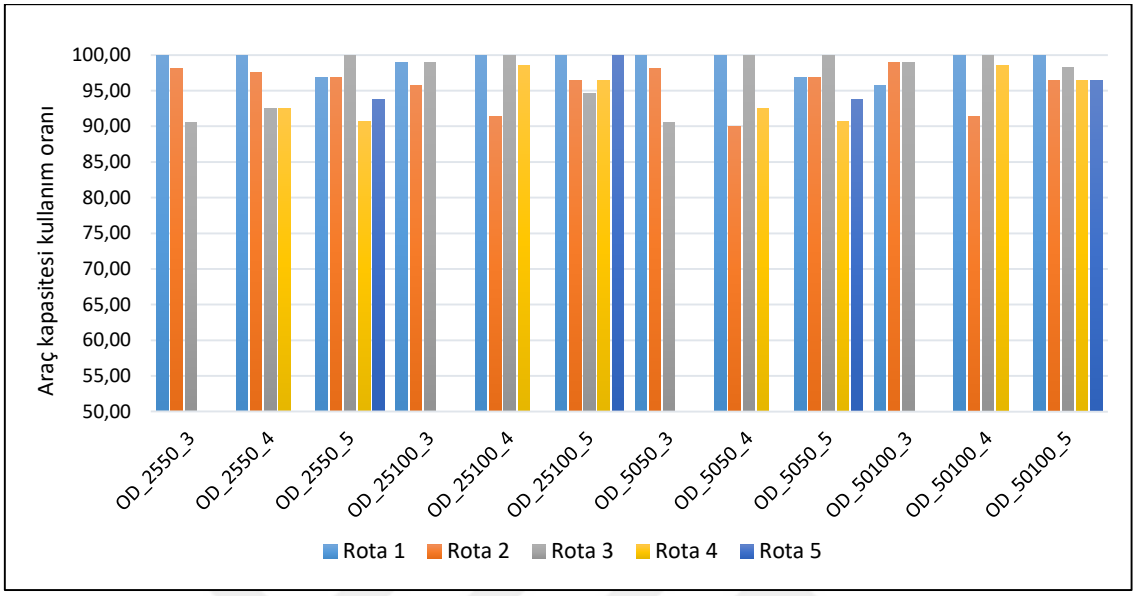
Çözüm süreleri açısından bakıldığında 12 adet test problemi için mat-sezgisel yaklaşımla elde edilen ortalama çözüm süresinin, iteratif çözüm sürelerinin ortalamasının üçte biri kadar olduğu anlaşılmaktadır. $EKBÇD, BÇDO$ ve $EIBÇD$ iyileştirme oranları ise sırasıyla %19,62 – %32,23, %13,12 - %18,64 ve %4,98 - %13,37 aralılarında oldukları görülmektedir. 12 adet test problemi için bu göstergelerin ortalamaları sırasıyla %25,34, %16,84 ve %8,98 olarak elde edilmiştir.

Tablo 5.9'da verilen test problemlerinin mat-sezgisel çözümleri için AKKO değerleri Tablo 5,10'da verilmektedir. Ayrıca bu değerler, Şekil 5.5'te çubuk grafiği şeklinde sunulmaktadır.

Tablo 5.10. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri

Problem kodu	**Araç kapasitesi kullanım oranları (%)**				
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**
OD_2550_3	100,00	98,11	90,57		
OD_2550_4	100,00	97,50	92,50	92,50	
OD_2550_5	96,88	96,88	100,00	90,63	93,75
OD_25100_3	98,92	95,70	98,92		
OD_25100_4	100,00	91,43	100,00	98,57	
OD_25100_5	100,00	96,43	94,64	96,43	100,00
OD_5050_3	100,00	98,11	90,57		
OD_5050_4	100,00	90,00	100,00	92,50	
OD_5050_5	96,88	96,88	100,00	90,63	93,75
OD_50100_3	95,70	98,92	98,92		
OD_50100_4	100,00	91,43	100,00	98,57	
OD_50100_5	100,00	96,43	98,21	96,43	96,43

Tablo ve grafikten araç kapasitesi kullanım oranlarının 12 adet test probleminin tamamında en az %90'ın üzerinde oldukça yüksek olduğu görülmektedir.



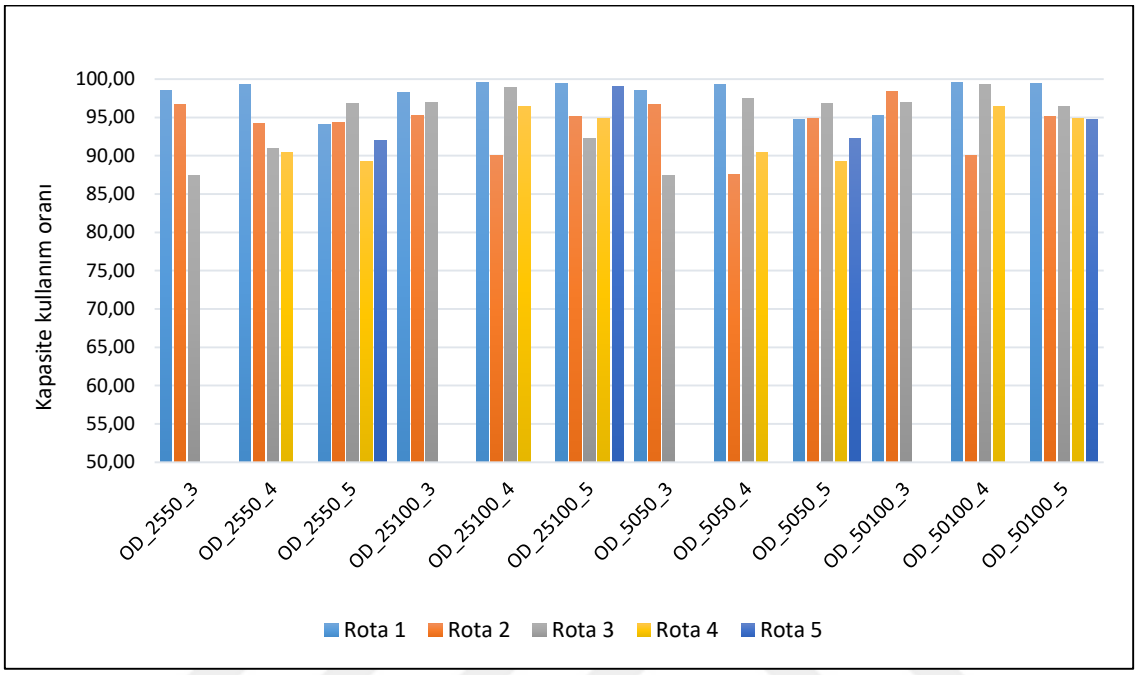
Şekil 5.5. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri

Orta boyutlu dikdörtgen formlu kroki kullanılarak oluşturulan test problemlerinin mat-sezgisel algoritma ile çözüm sonuçları baz alınarak hesaplanan kapasite kullanım oranları Tablo 5.11 ve Şekil 5.6'da sunulmaktadır.

Tablo 5.11. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri

Problem kodu	**Kapasite kullanım oranları (%)**				
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**
OD_2550_3	98,49	96,71	87,47		
OD_2550_4	99,23	94,14	90,87	90,37	
OD_2550_5	94,11	94,27	96,82	89,31	92,02
OD_25100_3	98,27	95,25	96,94		
OD_25100_4	99,54	90,07	98,93	96,47	
OD_25100_5	99,39	95,17	92,21	94,89	99,01
OD_5050_3	98,49	96,71	87,47		
OD_5050_4	99,23	87,54	97,47	90,37	
OD_5050_5	94,73	94,89	96,82	89,31	92,18
OD_50100_3	95,25	98,32	96,94		
OD_50100_4	99,54	90,07	99,24	96,47	
OD_50100_5	99,39	95,17	96,43	94,89	94,79

Tablodaki verilerden ve şekilde sunulan grafikten yola çıkarak en düşük kapasite kullanım oranının %87,47 olduğu ve iş yükünün çekici trenler arasında dengeli bir şekilde dağıldığı ifade edilebilir.



Şekil 5.6. Orta boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri

Orta boyutlu kare formlu kroki kullanılarak oluşturulan 12 adet test probleminin üç çözüm yaklaşımı ile çözülmesiyle elde edilen sonuçlar ve bu sonuçlara göre hesaplanan çeşitli göstergeler Tablo 5.12'de sunulmaktadır. Birinci çözüm yaklaşımı ile elde edilen sonuçlara bakıldığında çoğu test problemi için 12 saatlik sürede tam sayılı bir çözüm bile bulunamadığı, bulunanlarda ise iteratif yaklaşımla elde edilen optimum çözüme yaklaşılamadığı görülmektedir. Sadece en küçük boyutlu olan OK_2550_3 kodlu problem için optimum çözüm değeri bulunabilmiştir. Ancak bu problemde de çözümün optimalliğinin kanıtlanabilmesi için oldukça uzun süreler gerekeceği açıktır.

Orta boyutlu dikdörtgen formlu kroki ile üretilen test problemlerinde olduğu gibi orta boyutlu kare formlu kroki ile üretilen test problemlerinde de birinci çözüm yaklaşımının kullanışlı olmaktan oldukça uzak olduğu anlaşılmaktadır.

Tablo 5.12. Orta boyutlu kare kroki ile üretilen test problemlerinin çözüm sonuçları

Problem kodu	Birinci çözüm yaklaşımı				İkinci çözüm yaklaşımı				Üçüncü çözüm yaklaşımı									
	Çözüm süresi	Alt sınır	En iyi tam sayılı çözüm	Fark (%)	Çözüm süresi	Alt sınır	Optimum çözüm değeri	Çözüm / alt sınır bulunan model	Ort. Çözüm süresi	En iyi başlangıç çözüm değeri ($EIBÇD$)	Başlangıç çözüm değerlerinin ortalaması ($BÇDO$)	En kötü başlangıç çözüm değeri ($EKBÇD$)	Ort. çözüm değeri	En iyi çözüm değeri ($EIÇD$)	($EKBÇD$) iyileştirme oranı	$BÇDO$ iyileştirme oranı (%)	$EIBÇD$ iyileştirme oranı (%)	$EIÇD$ - optimum çözüm değeri farkı (%)
OK_2550_3	43200	455,20	524	13,11	5410	524	524	Model-2	3518	576	600,4	616	528	528	16,67	13,71	9,09	**0,76**
OK_2550_4	43200	493,80	604	18,24	5572	592	592	Model-2	3794	628	671,5	716	595,2	592	20,95	13,43	6,08	**0,00**
OK_2550_5	43200	514,09	752	31,63	12059	676	676	Model-2	4570	736	791,8	844	676	676	24,85	17,13	8,88	**0,00**
OK_25100_3	43200	449,00	-	-	5452	544	544	Model-2	3616	596	635,3	664	544	544	22,06	16,78	9,56	**0,00**
OK_25100_4	43200	483,05	-	-	5655	600	600	Model-2	4535	660	709,9	752	603,2	600	25,33	18,32	10,00	**0,00**
OK_25100_5	43200	524.08	920	43,00	21021	696	696	Model-2	6170	724	833,1	896	704	704	27,27	18,34	2,84	**1,15**
OK_5050_3	43200	436.95	-	-	5410	524	524	Model-2	3517	576	600,4	616	528	528	16,67	13,71	9,09	**0,76**
OK_5050_4	43200	484,43	702	30,99	5572	592	592	Model-2	3794	628	671,5	716	599,2	592	20,95	13,43	6,08	**0,00**
OK_5050_5	43200	508.82	-	-	43200	676	-	Model-2	5877	736	791,8	844	688	688	22,67	15,09	6,98	1,78
OK_50100_3	43200	438,00	-	-	43200	544	-	Model-2	3615	596	635,3	664	544	544	22,06	16,78	9,56	**0,00**
OK_50100_4	43200	471,50	-	-	26949	600	600	Model-1	4533	660	709,9	752	603,2	600	25,33	18,32	10,00	**0,00**
OK_50100_5	43200	524,48	-	-	43200	696	-	Model-2	6481	724	833,1	896	708	708	26,55	17,67	2,26	1,72
Ortalama	**43200**				**18558,3**				**4501,6**						**22,61**	**16,06**	**7,53**	**0,27**

İteratif çözüm yaklaşımı ile yapılan koşum sonuçları incelendiğinde, üç problem (OK_5050_5, OK_50100_3, OK_50100_5) dışındaki problemlere makul sürelerde optimum şekilde çözüm bulunabildiği görülmektedir. Bu üç probleme ait çözüm süreçleri incelendiğinde, Model-2 ile elde edilen alternatif optimum çözümler içerisinde çakışmasız çözüm bulunamadığı ve kısıt eklemeli Model-1'in de süre limiti içerisinde çözümü tamamlayamadığı görülmektedir. Bu tabloya bakıldığında, artan dar yol oranı ve araç sayısının çakışma olasılığını artırdığı ve böylece iteratif çözüm yaklaşımında Model-2 ile bulunan alternatif optimum çözümler arasından çakışmasız çözüm elde edilebilme olasılığını düşürdüğü ifade edilebilir.

OK_2550_5 ve OK_25100_5 kodlu problemlerin çözüm süresinin diğer problemlere göre nispeten daha uzun olması ise Model-3 ile alt turların iteratif şekilde elenmesi sürecinin daha uzun sürmesinden kaynaklanmaktadır. OD_25100_5 kodlu deneme probleminde de görülen bu durum, araç sayısı yükseldikçe, özellikle 5 ve üzerine çıktığında, Model-3 ile çözüm yapılırken çok sayıda araca ait alt turların kırılmasının gittikçe makul sürelerde yapılamayacak bir hale geleceğini ifade etmektedir.

Üçüncü çözüm yaklaşımı ile elde edilen sonuçlar incelendiğinde, test problemlerinin büyük çoğunluğunda optimum çözümün bulunabildiği görülmektedir. Diğer problemlerde ise optimuma oldukça yakın çözüm değerlerinin elde edilebildiği söylenebilir.

OK_50100_3 kodlu problemde ise mat-sezgisel algoritma ile bulunan en iyi çözüm değerinin iteratif çözüm yaklaşımı ile elde edilen alt sınıra eşit olduğu görülmektedir. Alt sınır değerinden daha iyi bir çözüm bulunamayacağı için bu problemin de optimum çözümünün bulunduğu sonucu çıkarılabilmektedir. Genel olarak bakıldığında ise iteratif çözüm yaklaşımı ile optimum çözüm değeri elde edilen test problemleri için mat-sezgisel algoritmanın optimum çözümden sapma ortalaması %0,27 gibi oldukça düşük bir değerdir.

En kötü başlangıç bireyi iyileştirme oranı, başlangıç bireyler ortalamasını iyileştirme oranı ve en iyi başlangıç bireyi iyileştirme oranı göstergelerine bakıldığında bunların

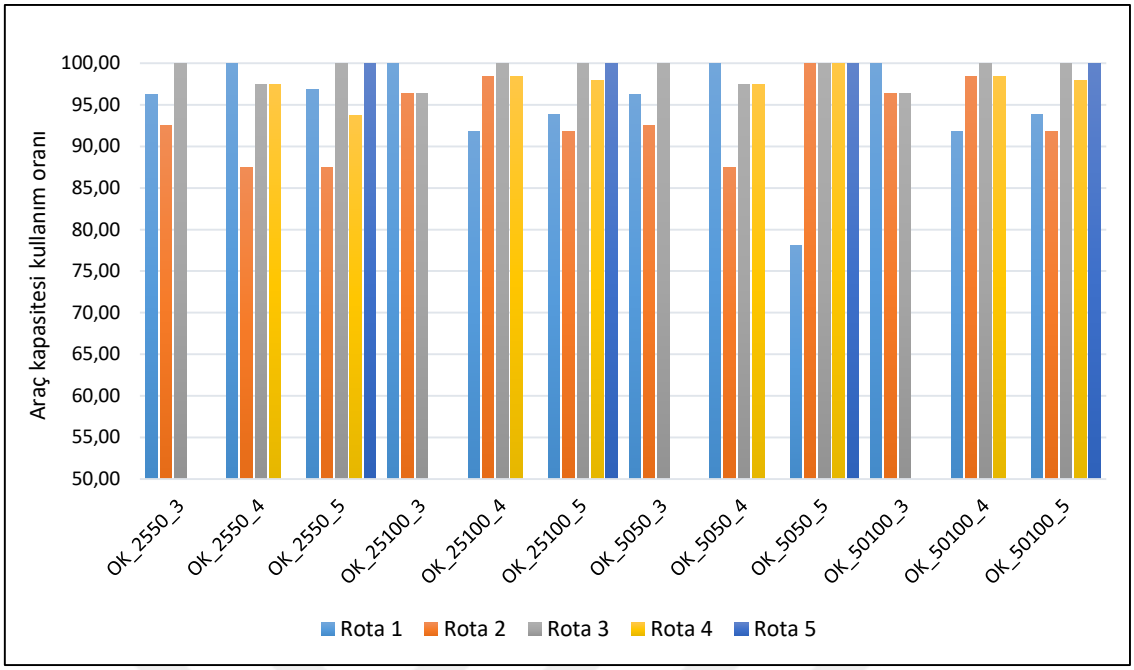
sırasıyla, %16,67 - %27,27, %13,43 - %18,34 ve %2,26 - %10,00 aralıklarında oldukları ve ortalamalarının %22,61, %16,06 ve %7,53 olduğu görülmektedir. Her bir test problemi için beş tekrarlı koşum sonucunda elde edilen ortalama çözüm değerlerine bakıldığında ise en iyi çözüm değerlerine oldukça yakın hatta birçok problemde eşit oldukları görülmektedir.

Orta boyutlu kare formlu kroki kullanılarak üretilen test problemlerinin mat-sezgisel algoritma ile çözülmesi sonucu elde edilen AKKO değerleri Tablo 5.13'te verilmektedir. Ayrıca bu değerler, çubuk grafiği şeklinde Şekil 5.7'de görselleştirilmekte ve sunulmaktadır.

Tablo 5.13. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri

Problem kodu	**Araç kapasitesi kullanım oranları (%)**				
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**
OK_2550_3	96,23	92,45	100,00		
OK_2550_4	100,00	87,50	97,50	97,50	
OK_2550_5	96,88	87,50	100,00	93,75	100,00
OK_25100_3	100,00	96,30	96,30		
OK_25100_4	91,80	98,36	100,00	98,36	
OK_25100_5	93,88	91,84	100,00	97,96	100,00
OK_5050_3	96,23	92,45	100,00		
OK_5050_4	100,00	87,50	97,50	97,50	
OK_5050_5	78,13	100,00	100,00	100,00	100,00
OK_50100_3	100,00	96,30	96,30		
OK_50100_4	91,80	98,36	100,00	98,36	
OK_50100_5	93,88	91,84	100,00	97,96	100,00

Tablo ve şekil incelendiğinde, sadece OK_5050_5 kodlu problemde araçlardan bir tanesinin kapasite kullanım oranının %78,13 olduğu ve diğer araçlara göre düşük kaldığı görülmektedir. Ancak diğer problemler için AKKO değerlerinin %87,5 ile %100 arasında dengeli bir biçimde gerçekleştiği görülmektedir.



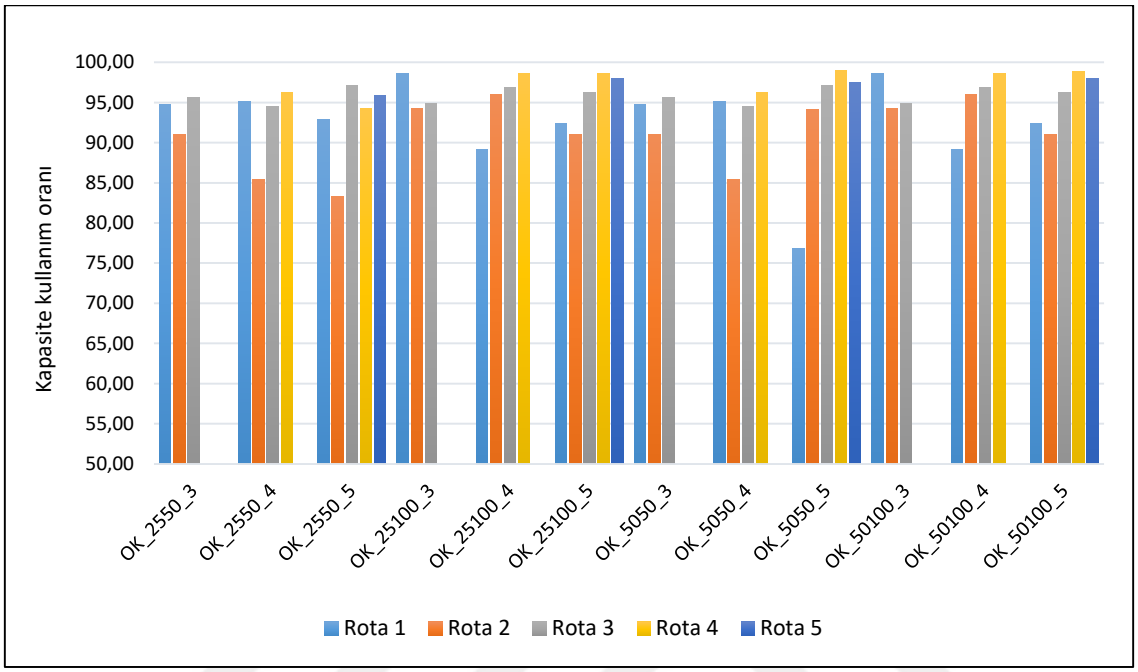
Şekil 5.7. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri

Çevrim süresinin ne kadarlık bir bölümünde çekici trenlerin ve operatörlerinin aktif olarak çalıştığını ifade eden KKO verileri OK problemleri için Tablo 5.14'te verilmiş ve çubuk grafiği şeklinde Şekil 5.8'de sunulmuştur.

Tablo 5.14. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri

Problem kodu	**Kapasite kullanım oranları (%)**				
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**
OK_2550_3	94,71	90,98	95,57		
OK_2550_4	95,15	85,48	94,47	96,22	
OK_2550_5	92,93	83,32	97,16	94,24	95,93
OK_25100_3	98,68	94,22	94,87		
OK_25100_4	89,19	96,04	96,85	98,60	
OK_25100_5	92,43	91,00	96,29	98,68	97,99
OK_5050_3	94,71	90,98	95,57		
OK_5050_4	95,15	85,48	94,47	96,22	
OK_5050_5	76,79	94,08	97,16	99,00	97,46
OK_50100_3	98,68	94,22	94,87		
OK_50100_4	89,19	96,04	96,85	98,60	
OK_50100_5	92,43	91,00	96,29	98,89	97,99

Veriler incelendiğinde, AKKO değeri de düşük çıkan OK_5050_5 kodlu test probleminde rota için KKO'nun da 76,79 olarak elde edildiği, bunun dışındaki problemler ve araçlar için genel olarak yüksek ve dengeli dağılıma sahip KKO'nın olduğu görülmektedir.



Şekil 5.8. Orta boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri

5.2.3. Büyük Boyutlu Problemler

Büyük boyutlu ve dikdörtgen formlu kroki kullanılarak üretilen test problemlerinin iteratif çözüm yöntemi ve mat-sezgisel algoritma ile elde edilen çözüm sonuçları ve bu sonuçlar kullanılarak hesaplanan çeşitli göstergeler Tablo 5.15'te verilmektedir. Büyük boyutlu krokilerle oluşturulan problemlerin birinci çözüm yaklaşımı yani Model-1 ile çözülmesi denenmiş ancak problemlerin büyüklüğünden dolayı modelin yüklenmesi aşamasında yetersiz bellek uyarısı alındığı için herhangi bir çözüm elde edilememiştir. Bu nedenle büyük boyutlu krokiler kullanılarak oluşturulan problemler için sadece ikinci ve üçüncü çözüm yaklaşımları ile elde edilen sonuçlar verilmiştir.

Tablo 5.15'te iteratif çözüm yaklaşımı ile elde edilen sonuçlara bakıldığında 12 saatlik süre sonunda hiçbir test problemi için geçerli tam sayılı bir çözüm bulunamadığı görülmektedir. Bunun nedeni artan araç, görev ve koridor sayısı nedeniyle Model-3 ile alt turların olmadığı olurlu bir çözümün bulunamamasıdır.

Tablo 5.15. Büyük boyutlu dikdörtgen kroki ile üretilen test problemlerinin çözüm sonuçları

Problem kodu	İkinci çözüm yaklaşımı				Üçüncü çözüm yaklaşımı									
	Çözüm süresi	Alt sınır	Optimum çözüm değeri	Çözüm / alt sınır bulunan model	Ort. Çözüm süresi	En iyi başlangıç çözüm değeri ($EIBÇD$)	Başlangıç çözüm değerlerinin ortalaması ($BÇDO$)	En kötü başlangıç çözüm değeri ($EKBÇD$)	Ort. çözüm değeri	En iyi çözüm değeri ($EIÇD$)	($EKBÇD$) iyileştirme oranı	$BÇDO$ iyileştirme oranı (%)	$EIBÇD$ iyileştirme oranı (%)	$EIÇD$ - optimum çözüm değeri farkı (%)
BD_25100_5	43200	1290,95	-	Model-3	40664	1538	1617,9	1690	1429,6	1428	18,35	13,30	7,70	-
BD_25100_6	43200	1348,31	-	Model-3	39916	1648	1751,6	1878	1516	1516	23,88	15,54	8,71	-
BD_25100_7	43200	1403,00	-	Model-3	41510	1758	1904,6	2028	1636	1636	23,96	16,42	7,46	-
BD_25200_5	43200	1297.14	-	Model-3	39962	1596	1670,8	1816	1484,8	1480	22,70	12,89	7,84	-
BD_25200_6	43200	1350.99	-	Model-3	39895	1746	1875,2	2016	1558	1558	29,40	20,36	12,07	-
BD_25200_7	43200	1315,44	-	Model-3	39843	1788	2009,7	2156	1680	1674	28,79	20,05	6,81	-
BD_50100_5	43200	1290,95	-	Model-3	40649	1538	1617,9	1690	1429,2	1428	18,35	13,30	7,70	-
BD_50100_6	43200	1348,31	-	Model-3	43392	1648	1751,6	1878	1526	1526	23,07	14,78	7,99	-
BD_50100_7	43200	1403,00	-	Model-3	43407	1758	1904,6	2028	1656	1656	22,46	15,01	6,16	-
BD_50200_5	43200	1297.14	-	Model-3	43331	1596	1670,8	1816	1490,8	1486	22,21	12,44	7,40	-
BD_50200_6	43200	1350.99	-	Model-3	40062	1746	1875,2	2016	1568	1568	28,57	19,59	11,35	-
BD_50200_7	43200	1315,44	-	Model-3	43357	1788	2009,7	2156	1695,6	1686	27,88	19,20	6,05	-
Ortalama	**43200**				**41332,33**						**24,13**	**16,07**	**8,10**	**-**

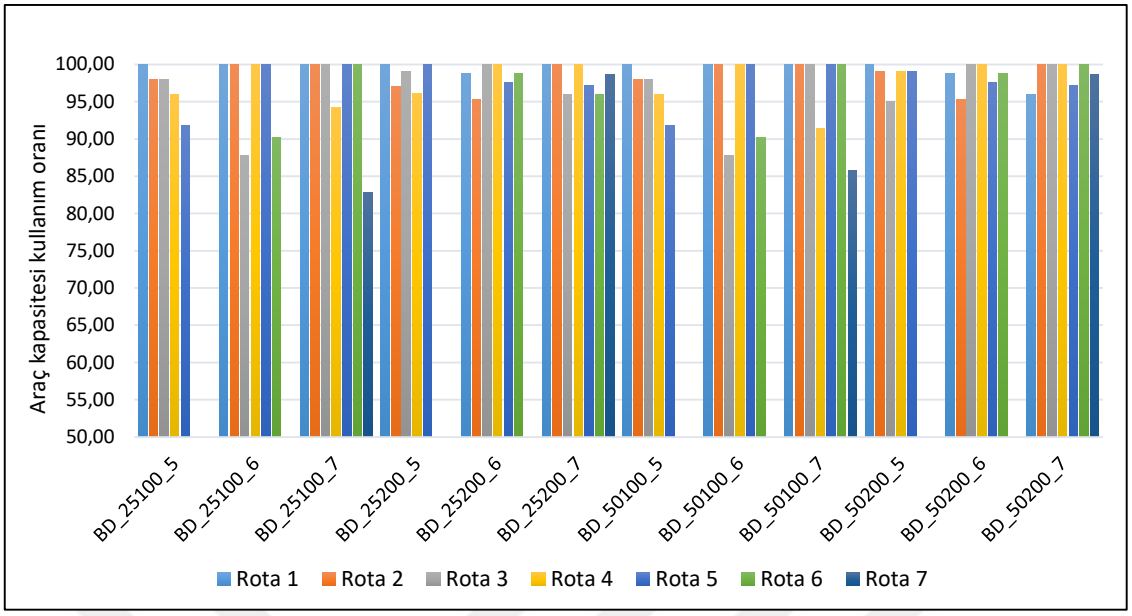
Hiç bir test problemi için optimum çözüm değeri bulunamadığı için mat-sezgisel algoritma ile elde edilen sonuçlar hakkında, sadece başlangıç bireylerin iyileştirme oranları üzerinden yorum yapılabilmektedir. Tablo 5.15'te görüldüğü gibi, en kötü başlangıç bireyi iyileştirme oranı, başlangıç bireyler ortalamasını iyileştirme oranı ve en iyi başlangıç bireyi iyileştirme oranı göstergeleri sırasıyla, %18,35 – %29,40, %12,44 - %20,36 ve %6,05 - %12,07 aralıklarında değerler almaktadır. Bu göstergelerin ortalama değerleri ise sırasıyla %24,13, %16,07 ve %8,10 olarak elde edilmiştir. Beş tekrarlı koşum sonucunda her bir test problemi için elde edilen ortalama çözüm değerlerine bakıldığında ise en iyi çözüm değerlerine oldukça yakın hatta birçok problemde eşit oldukları görülmektedir.

Tablo 5.15'te çözüm değerleri verilen test problemlerinin mat-sezgisel algoritma ile çözülmesi sonucu elde edilen AKKO değerleri Tablo 5.16'da verilmektedir. Ayrıca bu değerler Şekil 5.9'da çubuk grafiği şeklinde sunulmaktadır.

Tablo 5.16. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri

Problem kodu	**Araç kapasitesi kullanım oranları (%)**						
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**	**ÇT 6**	**ÇT 7**
BD_25100_5	100,00	97,96	97,96	95,92	91,84		
BD_25100_6	100,00	100,00	87,80	100,00	100,00	90,24	
BD_25100_7	100,00	100,00	100,00	94,29	100,00	100,00	82,86
BD_25200_5	100,00	97,06	99,02	96,08	100,00		
BD_25200_6	98,82	95,29	100,00	100,00	97,65	98,82	
BD_25200_7	100,00	100,00	95,89	100,00	97,26	95,89	98,63
BD_50100_5	100,00	97,96	97,96	95,92	91,84		
BD_50100_6	100,00	100,00	87,80	100,00	100,00	90,24	
BD_50100_7	100,00	100,00	100,00	91,43	100,00	100,00	85,71
BD_50200_5	100,00	99,02	95,10	99,02	99,02		
BD_50200_6	98,82	95,29	100,00	100,00	97,65	98,82	
BD_50200_7	95,89	100,00	100,00	100,00	97,26	100,00	98,63

Veriler ve grafik incelendiğinde BD_25100_7 kodlu problemde bir çekici trenin AKKO değerinin %82,86 ile diğer araçlara göre nispeten küçük kaldığı görülmektedir. Ancak genel duruma bakıldığında, AKKO oranlarının birbirlerine yakın olduğu yani çekici trenler arasında dengeli bir yük dağılımı olduğu söylenebilir.



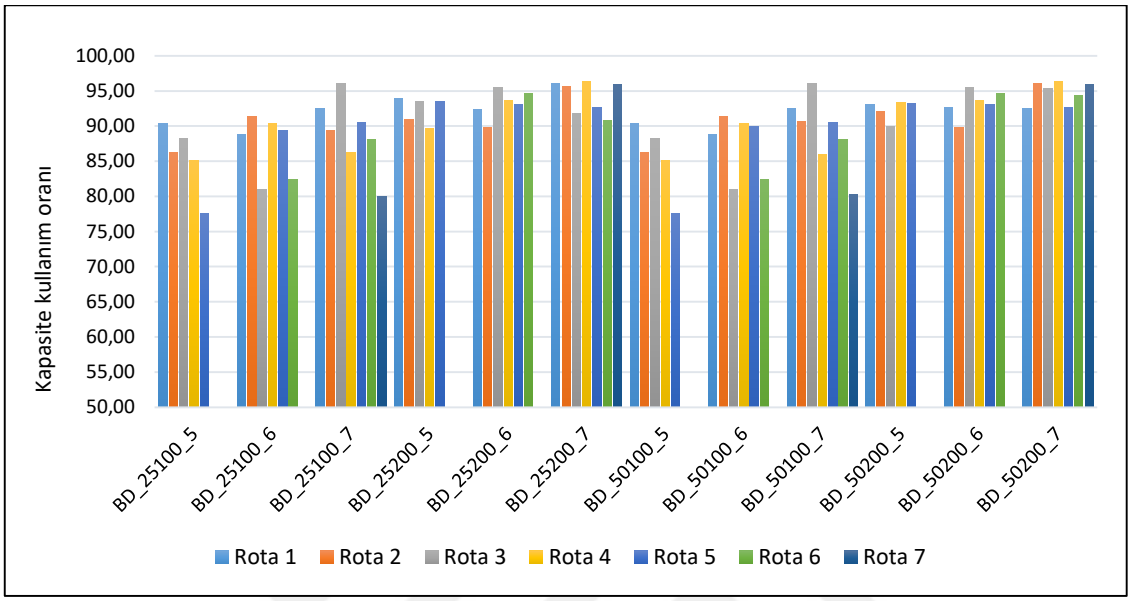
Şekil 5.9. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için AKKO değerleri

Büyük boyutlu dikdörtgen kroki kullanılarak oluşturulan 12 adet test probleminin mat-sezgisel algoritma ile çözüm sonuçlarına göre rotaların kapasite kullanım oranlarına ilişkin elde edilen veriler Tablo 5.17'de ve ayrıca grafik şeklinde Şekil 5.10'da verilmektedir.

Tablo 5.17. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri

Problem kodu	**Kapasite kullanım oranları (%)**						
	Rota 1	**Rota 2**	**Rota 3**	**Rota 4**	**Rota 5**	**Rota 6**	**Rota 7**
BD_25100_5	90,37	86,19	88,30	85,09	77,62		
BD_25100_6	88,82	91,38	80,98	90,42	89,36	82,42	
BD_25100_7	92,48	89,34	96,11	86,21	90,47	88,09	79,94
BD_25200_5	93,97	90,94	93,56	89,61	93,44		
BD_25200_6	92,39	89,85	95,46	93,70	93,12	94,73	
BD_25200_7	96,06	95,65	91,78	96,34	92,64	90,75	95,82
BD_50100_5	90,37	86,19	88,30	85,09	77,62		
BD_50100_6	88,82	91,38	80,98	90,42	89,89	82,42	
BD_50100_7	92,48	90,60	96,11	85,89	90,47	88,09	80,25
BD_50200_5	93,00	92,10	89,92	93,36	93,17		
BD_50200_6	92,68	89,85	95,46	93,70	93,12	94,73	
BD_50200_7	92,47	96,06	95,38	96,34	92,64	94,35	95,82

Veriler incelendiğinde bazı rotalar için kapasite kullanım oranları %77,62 seviyelerine kadar düşse de, tekil olarak düşünüldüğünde test problemlerinin her biri içerisinde iş yükü dağılımının dengeli olduğu ifade edilebilir.



Şekil 5.10. Büyük boyutlu dikdörtgen kroki kullanılarak üretilen test problemleri için KKO değerleri

Büyük boyutlu kare kroki kullanılarak oluşturulan test problemlerinin ikinci ve üçüncü çözüm yaklaşımlarıyla çözülmesi sonucu elde edilen sonuçlar Tablo 5.18'de verilmektedir. Büyük boyutlu dikdörtgen kroki ile üretilen test problemlerinde olduğu gibi bu test problemlerinde de, yüksek değişken ve kısıt sayısı nedeniyle iteratif çözüm yaklaşımıyla hiç bir problem için olurlu çözüm elde edilememiştir. Mat-sezgisel algoritma ile elde edilen sonuçlara bakıldığında ise, bulunan en iyi çözüm değerlerini kıyaslayacak optimum çözümler bulunamadığı için sadece başlangıç çözümleri iyileştirme oranları üzerinden değerlendirmeler yapmak mümkündür.

Tablo 5.18'de verildiği gibi, en kötü başlangıç bireyi iyileştirme oranı, başlangıç bireyler ortalamasını iyileştirme oranı ve en iyi başlangıç bireyi iyileştirme oranı göstergeleri sırasıyla, %18,15 – %36,44, %13,07 - %18,42 ve %5,43 - %10,37 aralıklarında değerler almaktadır. Bu göstergelerin ortalama değerlerinin ise sırasıyla %25,23, %15,26 ve %8,05 olduğu görülmektedir. Beş tekrarlı çözüm sonucunda her bir test problemi için elde edilen ortalama çözüm değerlerinin ise en iyi çözüm değerlerine oldukça yakın hatta birçok problemde eşit oldukları görülmektedir.

Tablo 5.18. Büyük boyutlu kare kroki ile üretilen test problemlerinin çözüm sonuçları

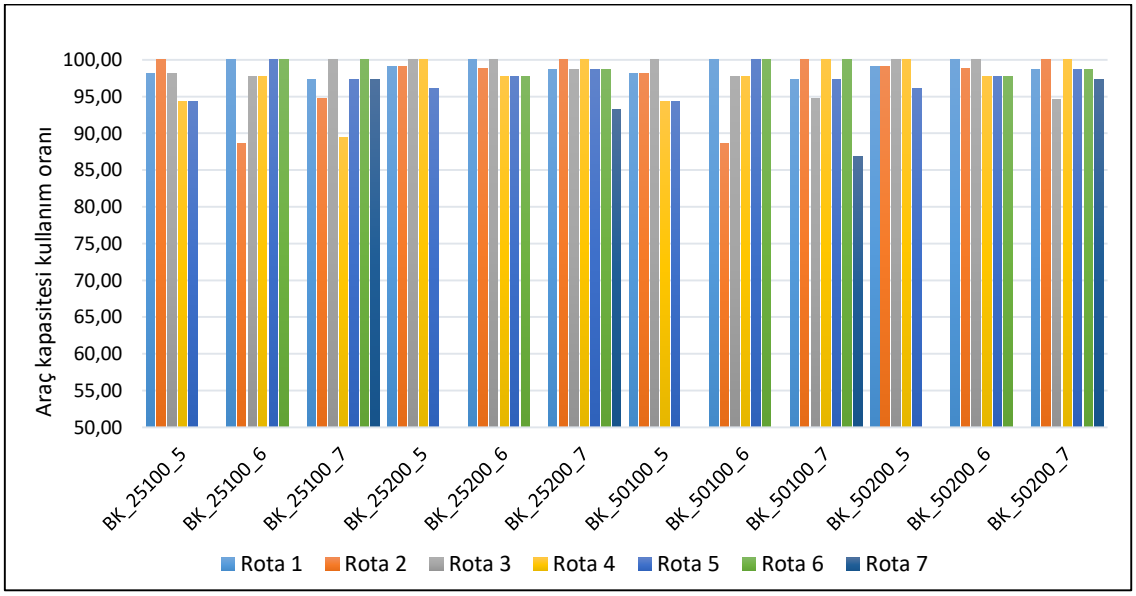
Problem kodu	İkinci çözüm yaklaşımı				Üçüncü çözüm yaklaşımı									
	Çözüm süresi	Alt sınır	Optimum çözüm değeri	Çözüm / alt sınır bulunan model	Ort. Çözüm süresi	En iyi başlangıç çözüm değeri ($EIBÇD$)	Başlangıç çözüm değerlerinin ortalaması ($BÇDO$)	En kötü başlangıç çözüm değeri ($EKBÇD$)	Ort. çözüm değeri	En iyi çözüm değeri ($EIÇD$)	($EKBÇD$) iyileştirme oranı	$BÇDO$ iyileştirme oranı (%)	$EIBÇD$ iyileştirme oranı (%)	$EIÇD$ - optimum çözüm değeri farkı (%)
BK_25100_5	43200	1323,07	-	Model-3	39943	1516	1573,9	1698	1380	1380	23,04	14,05	9,86	-
BK_25100_6	43200	1409,32	-	Model-3	39907	1682	1759,8	1836	1532	1524	20,47	15,47	10,37	-
BK_25100_7	43200	1457,83	-	Model-3	39925	1794	1916,2	2044	1682	1682	21,52	13,92	6,66	-
BK_25200_5	43200	1230,58	-	Model-3	43770	1530	1676,8	1932	1416	1416	36,44	18,42	8,05	-
BK_25200_6	43200	1437,65	-	Model-3	40203	1782	1917,5	2070	1631,6	1628	27,15	17,78	9,46	-
BK_25200_7	43200	1480,93	-	Model-3	40353	1902	2048	2246	1780	1780	26,18	15,06	6,85	-
BK_50100_5	43200	1323,07	-	Model-3	41891	1516	1573,9	1698	1392	1392	21,98	13,07	8,91	-
BK_50100_6	43200	1409,32	-	Model-3	42844	1682	1759,8	1836	1555,2	1554	18,15	13,24	8,24	-
BK_50100_7	43200	1457,83	-	Model-3	41384	1794	1916,2	2044	1688	1688	21,09	13,52	6,28	-
BK_50200_5	43200	1230,58	-	Model-3	44813	1530	1676,8	1932	1430	1430	35,10	17,26	6,99	-
BK_50200_6	43200	1437,65	-	Model-3	40208	1782	1917,5	2070	1635,2	1628	27,15	17,78	9,46	-
BK_50200_7	43200	1480,93	-	Model-3	43698	1902	2048	2246	1804	1804	24,50	13,53	5,43	-
Ortalama	**43200**				**41578,25**						**25,23**	**15,26**	**8,05**	**-**

Büyük boyutlu kare kroki kullanılarak oluşturulan test problemlerinin mat-sezgisel algoritma ile çözülmesi sonucu elde edilen AKKO değerleri Tablo 5.19'da verilmektedir. Ayrıca bu değerler Şekil 5.11'de grafik olarak sunulmaktadır.

Tablo 5.19. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri

Problem kodu	**Araç kapasitesi kullanım oranları (%)**						
	ÇT 1	**ÇT 2**	**ÇT 3**	**ÇT 4**	**ÇT 5**	**ÇT 6**	**ÇT 7**
BK_25100_5	98,11	100,00	98,11	94,34	94,34		
BK_25100_6	100,00	88,64	97,73	97,73	100,00	100,00	
BK_25100_7	97,37	94,74	100,00	89,47	97,37	100,00	97,37
BK_25200_5	99,03	99,03	100,00	100,00	96,12		
BK_25200_6	100,00	98,84	100,00	97,67	97,67	97,67	
BK_25200_7	98,65	100,00	98,65	100,00	98,65	98,65	93,24
BK_50100_5	98,11	98,11	100,00	94,34	94,34		
BK_50100_6	100,00	88,64	97,73	97,73	100,00	100,00	
BK_50100_7	97,37	100,00	94,74	100,00	97,37	100,00	86,84
BK_50200_5	99,03	99,03	100,00	100,00	96,12		
BK_50200_6	100,00	98,84	100,00	97,67	97,67	97,67	
BK_50200_7	98,65	100,00	94,59	100,00	98,65	98,65	97,30

Tablodaki veriler ve şekilde sunulan grafik incelendiğinde AKKO değerlerinin minimum %86,84 olduğu ve çekici trenler arasında büyük farklılıklar olmadığı anlaşılmaktadır.



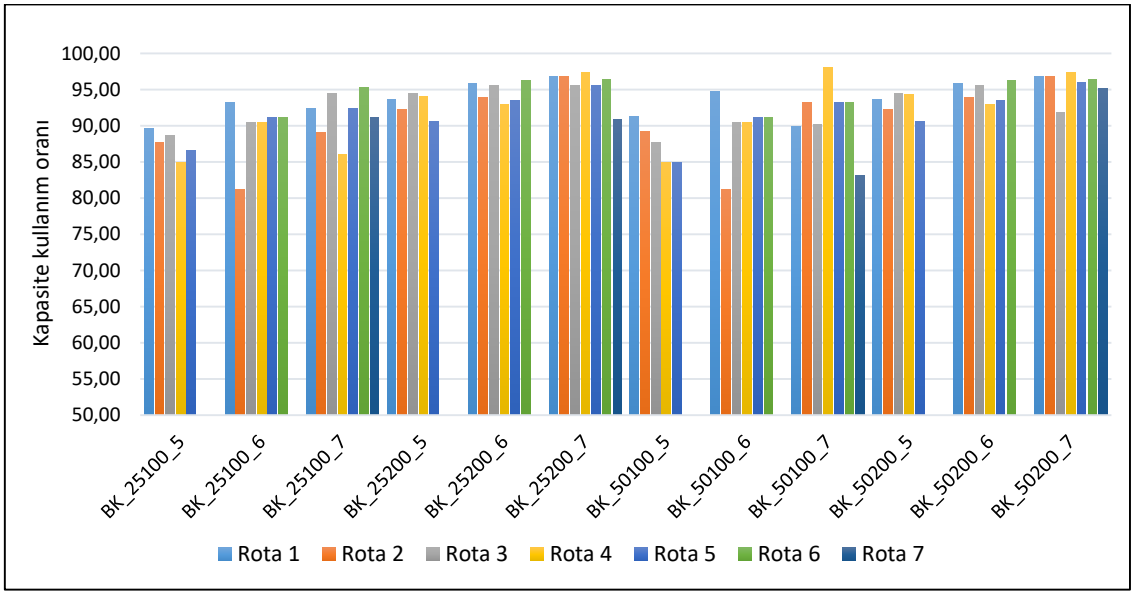
Şekil 5.11. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için AKKO değerleri

Büyük boyutlu kare kroki kullanılarak oluşturulan test problemlerinin mat-sezgisel algoritma ile çözüm sonuçlarına göre rotaların kapasite kullanım oranlarına ilişkin elde edilen veriler Tablo 5.20'de ve ayrıca grafik şeklinde Şekil 5.12'de verilmektedir.

Tablo 5.20. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri

Problem kodu	**Kapasite kullanım oranları (%)**						
	Rota 1	**Rota 2**	**Rota 3**	**Rota 4**	**Rota 5**	**Rota 6**	**Rota 7**
BK_25100_5	89,56	87,67	88,72	84,93	86,61		
BK_25100_6	93,20	81,19	90,39	90,39	91,15	91,15	
BK_25100_7	92,34	89,04	94,44	86,04	92,34	95,28	91,14
BK_25200_5	93,68	92,23	94,52	94,04	90,56		
BK_25200_6	95,82	93,87	95,53	92,90	93,49	96,23	
BK_25200_7	96,87	96,83	95,64	97,38	95,64	96,32	90,85
BK_50100_5	91,25	89,23	87,67	84,93	84,93		
BK_50100_6	94,73	81,19	90,39	90,39	91,15	91,15	
BK_50100_7	89,94	93,24	90,24	98,04	93,18	93,24	83,10
BK_50200_5	93,68	92,23	94,52	94,38	90,56		
BK_50200_6	95,82	93,87	95,53	92,90	93,49	96,23	
BK_50200_7	96,87	96,83	91,78	97,38	96,05	96,32	95,12

Tablodaki verilerden ve grafikten yola çıkarak, test problemlerinde KKO değerleri yaklaşık %80 ile %98 arasında olduğu ifade edilebilir.



Şekil 5.12. Büyük boyutlu kare kroki kullanılarak üretilen test problemleri için KKO değerleri

SÇDRÇ probleminin Np-Zor yapısı nedeniyle büyük boyutlu test problemlerinin optimum çözüm değerleri hiçbir şekilde elde edilememektedir. Dolayısıyla bu problemler için mat-sezgisel algoritma ile bulunan çözüm değerlerinin optimum değerler ile karşılaştırılarak kalitesinin değerlendirilmesi mümkün değildir. Diğer taraftan orta boyutlu dikdörtgen ve kare formlu krokiler ile oluşturulan test problemlerinin çoğu için iteratif çözüm yaklaşımıyla optimum çözüm değeri elde edilebilmiş ve mat-sezgisel algoritma ile bulunan çözüm değeri optimum değer ile kıyaslanarak yorumlanabilmiştir.

Bu noktadan hareketle bu çalışmada BÇDO iyileştirme oranları bakımından, optimum çözüm değeri bilinen orta boyutlu problemler ile büyük boyutlu problemler arasında anlamlı bir farklılık olup olmadığı istatistiksel olarak araştırılacaktır. Bunun sonucunda eğer BÇDO iyileştirme oranları açısından iki grup arasında anlamlı bir farklılık olmadığı tespit edilirse, orta boyutlu problemlerde optimum çözüme çok yakın değerler bulunabilen mat-sezgisel algoritmanın büyük boyutlu problemler için de optimum veya optimuma çok yakın değerler bulabildiğine dair bir kanıt elde edilmiş olacaktır.

Bu amaçla kullanılacak orta ve büyük boyutlu test problemlerine ait veriler Tablo 5.21'de verilmektedir. Görüldüğü gibi orta boyutlu problemlerden optimum çözüm değeri bulunamayan iki adet test problemi gruba dahil edilmemiştir.

Tablo 5.21. Orta ve büyük boyutlu problemlerin BÇDO iyileştirme oranlarına ait veriler

Orta boyutlu problemler				**Büyük boyutlu problemler**			
Problem kodu	**BÇDO iyileştirme oranı**	**Problem kodu**	**BÇDO iyileştirme oranı**	**Problem kodu**	**BÇDO iyileştirme oranı**	**Problem kodu**	**BÇDO iyileştirme oranı**
OD_2550_3	13,12	OK_2550_3	13,71	BD_25100_5	13,30	BK_25100_5	14,05
OD_2550_4	18,64	OK_2550_4	13,43	BD_25100_6	15,54	BK_25100_6	15,47
OD_2550_5	17,92	OK_2550_5	17,13	BD_25100_7	16,42	BK_25100_7	13,92
OD_25100_3	17,96	OK_25100_3	16,78	BD_25200_5	12,89	BK_25200_5	18,42
OD_25100_4	18,39	OK_25100_4	18,32	BD_25200_6	20,36	BK_25200_6	17,78
OD_25100_5	17,49	OK_25100_5	18,34	BD_25200_7	20,05	BK_25200_7	15,06
OD_5050_3	13,12	OK_5050_3	13,71	BD_50100_5	13,30	BK_50100_5	13,07
OD_5050_4	18,64	OK_5050_4	13,43	BD_50100_6	14,78	BK_50100_6	13,24
OD_5050_5	14,49	OK_50100_3	16,78	BD_50100_7	15,01	BK_50100_7	13,52
OD_50100_3	17,96	OK_50100_4	18,32	BD_50200_5	12,44	BK_50200_5	17,26
OD_50100_4	16,86			BD_50200_6	19,59	BK_50200_6	17,78
OD_50100_5	17,49			BD_50200_7	19,20	BK_50200_7	13,53

İki bağımsız grubun ortalamalarını kıyaslayarak aralarında anlamlı bir farklılık olup olmadığını araştırmak için en yaygın kullanılan istatistiksel yöntemlerden bir tanesi bağımsız örneklem t-testidir. Özellikle bu çalışmada olduğu gibi veri sayısının az olduğu durumlarda kullanılan t-testi, gruplardaki verilerin normal dağılıma uyması, grup varyanslarının homojenliği gibi önemli varsayımlara sahiptir. Bu nedenle öncelikle t-testinin en önemli varsayımlardan biri olan verilerin normal dağılıma uyması varsayımını test etmek için aşağıda verilen hipotezler oluşturulmuştur.

- H_0: BÇDO iyileştirme oranları değerlerinin dağılımı ile normal dağılım arasında fark yoktur.
- H_1: BÇDO iyileştirme oranları değerlerinin dağılımı ile normal dağılım arasında fark vardır.

Daha sonra bu hipotezleri sınamak için SPSS paket programı yardımıyla Kolmogorov-Smirnov ve Shapiro-Wilk testleri ile yapılmış ve sonuçlar Tablo 5.22'de verilmiştir.

Tablo 5.22. Normal dağılıma uygunluk testlerinin sonuçları

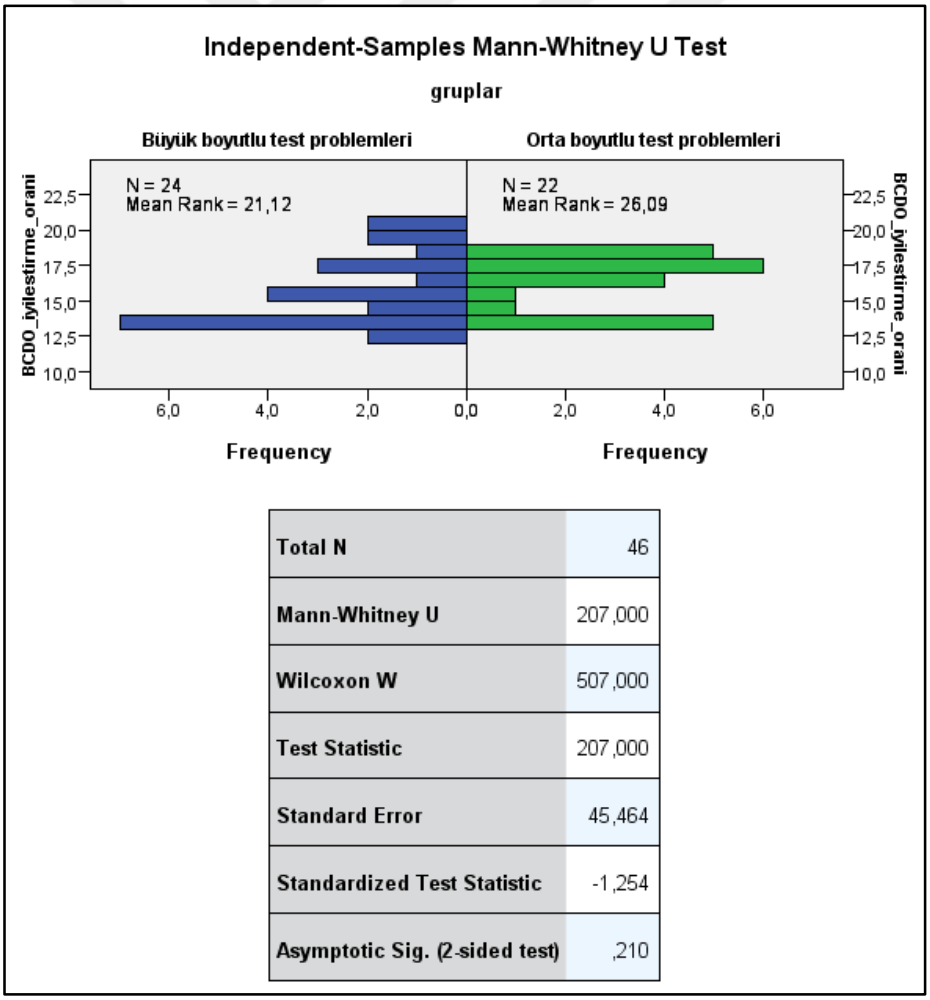
		Kolmogorov-Smirnov			Shapiro-Wilk		
		İstatistik	Serbestlik derecesi	p	İstatistik	Serbestlik derecesi	p
BÇDO iyileştirme oranı	Orta boyutlu test problemleri	0,243	22	0,002	0,809	22	0,001
	Büyük boyutlu test problemleri	0,155	24	0,140	0,902	24	0,024

Sonuç tablosunda orta boyutlu test problemleri için Kolmogorov-Smirnov ve Shapiro-Wilk testlerinin her ikisinde de p değerinin 0,05'den küçük olduğu görülmektedir. Dolayısıyla %5 hata payı ile değerlendirildiğinde H_0 hipotezinin reddedildiği yani orta boyutlu test problemlerinin BÇDO iyileştirme oranlarının normal dağılıma uymadığı kabul edilmektedir. Dolayısıyla bu veri grubu için bağımsız örneklem t-testinin anlamlı sonuçlar vermeyeceği, bunun yerine parametrik olmayan Mann-Whitney U testinin kullanılması gerektiği ortaya çıkmaktadır.

Bu nedenle, BÇDO iyileştirme oranları açısından gruplar arasından anlamlı bir farklılık olup olmadığını ifade eden hipotezler aşağıda verildiği gibi oluşturulmuş ve Mann-Whitney U testi ile test edilmiştir.

- H_0: BÇDO iyileştirme oranları açısından orta boyutlu krokiler kullanılarak oluşturulan test problemleri ile büyük boyutlu kroki kullanılarak oluşturulan test problemleri arasında anlamlı bir farklılık yoktur.
- H_1: BÇDO iyileştirme oranları açısından orta boyutlu krokiler kullanılarak oluşturulan test problemleri ile büyük boyutlu kroki kullanılarak oluşturulan test problemleri arasında anlamlı bir farklılık vardır.

SPSS paket programı yardımıyla elde edilen Mann-Whitney U testine ait sonuçlar Şekil 5.13'de verilmektedir.



Total N	46
Mann-Whitney U	207,000
Wilcoxon W	507,000
Test Statistic	207,000
Standard Error	45,464
Standardized Test Statistic	-1,254
Asymptotic Sig. (2-sided test)	,210

Şekil 5.13. Orta ve büyük boyutlu problemlerin BÇDO iyileştirme oranları için Mann-Whitney U testi sonucu

Şekilde Sig. (p) değerinin 0,05'den büyük olduğu görülmektedir. Bu nedenle H_0 hipotezinin kabul edildiği yani BÇDO iyileştirme oranları açısından orta boyutlu krokiler kullanılarak oluşturulan test problemleri ile büyük boyutlu krokiler kullanılarak oluşturulan test problemleri arasında anlamlı bir farklılık olmadığı görülmektedir. Bu test ile elde edilen sonuca dayalı olarak BÇDO iyileştirme oranları aynı olduğu için, mat-sezgisel algoritmanın büyük boyutlu problemlerde de orta boyutlu problemler gibi başlangıç çözümleri geliştirebildiği ve böylece optimum ya da optimuma çok yakın çözüm değerleri üretebildiği ifade edilebilmektedir.

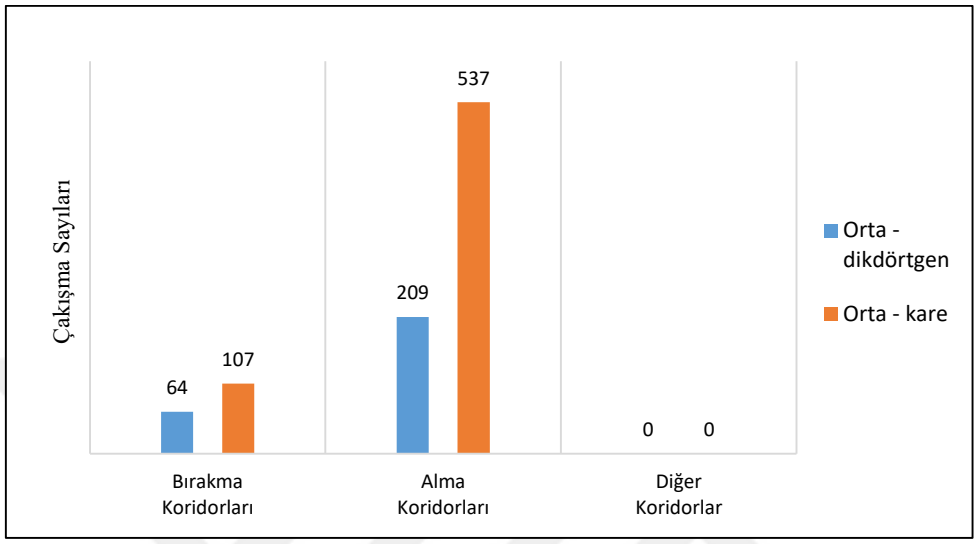
5.2.4. Çakışma Yoğunluğu Analizi

Test problemlerinin çözüm sonuçları değerlendirilirken, çekici trenlerin çakışma olasılığının yüksek olmasının hem iteratif yaklaşımla hem de mat-sezgisel algoritma ile çözüm arama sürecini zorlaştırdığı ve süreyi ciddi oranda olumsuz yönde etkilediği görülmüştür. Bu nedenle çakışmaların daha detaylı ele alınması gerektiği düşünülmüştür. Bu amaçla, test problemleri mat-sezgisel algoritma ile çözülürken üçüncü aşamada çakışmasız çözüm bulunana kadar incelenen bütün bireylerde tespit edilen çakışmalar, kroki şekilleri ve koridor tipleri bazında sınıflandırılarak Tablo 5.23'de sunulmuştur. Tabloda görüldüğü gibi koridorlar; bırakma, alma ve diğer olmak üzere üç tipte incelenmiştir. Bırakma koridorları; üretim hücrelerinin stok raflarına parça beslemesi yapılan koridorları, alma koridorları; süpermarket raflarından parça alınan koridorları ve diğer koridorlar ise bırakma ya da alma işlemi yapılmayan ve sadece geçiş için kullanılan koridorları ifade etmektedir.

Tablo 5.23. Kroki tiplerine göre test problemlerinde görülen çakışma sayıları

Kroki tipi	**Bırakma koridorları**	**Alma koridorları**	**Diğer koridorları**
Küçük	9	13	0
Orta - dikdörtgen	64	209	0
Orta - kare	107	537	0
Büyük - dikdörtgen	58688	169752	0
Büyük - kare	93022	224732	5096
Toplam	**151890**	**395243**	**5096**
Oran (%)	**27,50**	**71,57**	**0,92**

Tabloda sunulan verileri daha anlamlı bir şekilde ifade edebilmek için, orta boyutlu ve büyük boyutlu krokiler kullanılarak üretilen test problemlerinde karşılaşılan çakışmalar, kroki ve koridor tipleri bazında sırasıyla Şekil 5.14 ve Şekil 5.15'te sunulmaktadır.



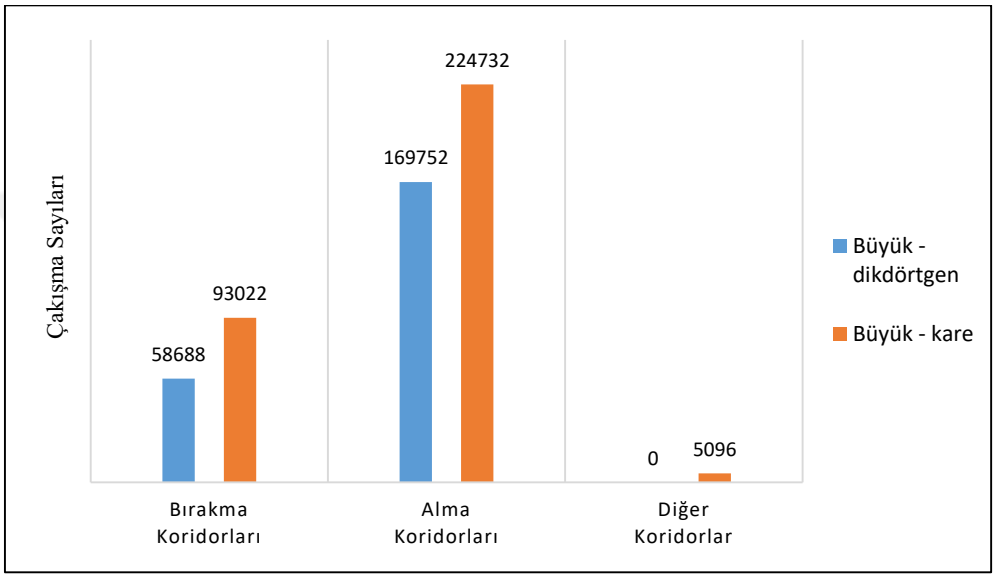
Şekil 5.14. Orta boyutlu krokilerle üretilen test problemlerinde görülen çakışma sayıları

Tablo 5.14'te oran olarak bakıldığında çakışmaların büyük bir çoğunluğunun (% 71,57) alma koridorlarında olduğu görülmektedir. Bunu % 27,50 ile bırakma koridorları ve son olarak çok küçük bir oran (% 0,92) ile diğer koridorlar takip etmektedir. Çekici trenler eşit hızlara sahip olduğu için kutu alma veya bırakma yapılmayan koridorlarda arka arkaya çakışma durumunun olması mümkün değildir. Bu koridorlarda çakışmalar, sadece karşılıklı iki trenin birbirlerini bloklamasıyla mümkün olabilmektedir.

Buna ek olarak genellikle çakışmalar, kutu alma ve bırakma sırasında bekleme yapması gereken çekici trenlerin diğer trenleri engellemesi ile ortaya çıkmaktadır. Alma koridorlarında tespit edilen çakışma miktarının bırakma koridorlarına göre çok daha yüksek olmasının, süpermarket raflarının üretim hücrelerine göre genellikle daha küçük bir alanda daha yoğun bir şekilde bulunmasından kaynaklandığı düşünülebilir.

Elde edilen bu sonuçlara göre, yeni bir fabrika tasarımı yaparken öncelikle süpermarket raflarının bulunduğu alma koridorlarının ve daha sonra mümkünse üretim hücrelerine parça beslemesinin yapıldığı bırakma koridorlarının iki çekici trenin geçebileceği şekilde tasarlanması, fabrika içerisinde daha güvenilir, sorunsuz ve rotalanması daha kolay bir taşıma sistemi sağlayacaktır.

Buna ek olarak, mevcut bir fabrikada zamanla değişen koşullar, üretim teknikleri veya parçalar nedeniyle hücre boyutları ve yapıları gibi yerleşim planını değiştirecek düzenlemeler yapılırken öncelikle alma koridorlarının ve daha sonra mümkünse yoğun olan bırakma koridorlarının daraltılmaması, fabrika içerisinde trafik kaynaklı karmaşaya, gecikmelere ve verim kaybına neden olacak bir lojistik sisteminden kaçınabilmeyi sağlayacaktır.



Şekil 5.15. Büyük boyutlu krokilerle üretilen test problemlerinde görülen çakışma sayıları

Sırasıyla orta ve büyük boyutlu krokiler için çakışma sayılarının kroki şekillerine göre ayrı ayrı gösterildiği Şekil 5.14 ve Şekil 5.15'teki grafikler incelendiğinde, kare formlu krokilerde dikdörtgen formlu krokilere göre daha fazla sayıda çakışma olduğu görülmektedir. Bu durum hem alma hem de bırakma koridorları için geçerlidir. Bunun sebebi, dikdörtgen şeklindeki bir fabrikaya nispeten kare şeklindeki bir alana kurulu fabrika içerisinde rotaların daha fazla iç içe geçmesi ve bunun bir sonucu olarak çekici trenlerin koridorları daha fazla ve yoğun şekilde ortak kullanma ihtiyacının ortaya çıkması olarak değerlendirilebilir. Bu verilerden hareketle, özellikle bu çalışmada ele alındığı şekilde bir üretim ortamına sahip yeni bir fabrika kurulması sürecinde kare biçimli bir alan yerine dikdörtgen şeklinde bir yerleşimin iç lojistik açısından bir avantaj oluşturacağını söylemek mümkündür.

6. SONUÇLAR VE ÖNERİLER

İlk olarak 1940'lı yıllarda Japon otomobil üreticisi Toyota'da geliştirilip uygulanan TZÜ sistemi, günümüzde dünya çapında başta otomobil, beyaz eşya, tüketici elektroniği ve mobilya olmak üzere çeşitli sektörlerde faaliyet gösteren irili ufaklı birçok üretici tarafından benimsenmiş ve yaygın olarak kullanılmaktadır. Nihai ve ara ürünlerin istenilen zamanda ve istenilen miktarda üretilmesi prensibine dayanan TZÜ felsefesi, süreç içi envanter ve stok seviyelerini minimum düzeye düşürerek üretim verimliliğini ve karlılığını artırmayı hedefleyen bir yaklaşımdır.

TZÜ sistemi ile imalat sürecinin, bekleme ve yığılmalar olmaksızın sürekli akışa sahip pürüzsüz bir işleyişe kavuşturulması amaçlanmaktadır. Bu şekilde bir üretimin mümkün olabilmesi için imalat hücrelerine veya iş istasyonlarına az miktarda, sürekli ve yüksek frekansta parça beslemesi yapılması gerekmektedir. Bu amaçla, süpermarket olarak adlandırılan fabrika içerisinde üretim alanlarına yakın, merkezi olmayan geçici stok noktalarından üretim hücrelerine veya iş istasyonlarına çekici trenler kullanılarak döngüsel seferlerle parça tedariği yaygın olarak tercih edilen bir yöntemdir. Bu şekilde yüksek taşıma kapasitelerine sahip çekici trenler ile birden çok istasyonun az miktarda olan parça ihtiyaçları tek bir seferde karşılanarak verimli ve esnek bir fabrika içi lojistik sağlanabilmektedir.

TZÜ felsefesi ile minimum seviyelerde stok tutulan, sürekli ve düzgün akışa sahip bir üretim sisteminin, israfların engellenerek maliyetlerin düşürülmesinin yanı sıra, beklenmedik durumlara hızlı cevap verebilme, çeviklik ve sistemde var olan sorunları kolay tespit edebilme gibi birçok avantajı bulunmaktadır. Ancak bununla birlikte, böyle bir üretim sisteminde, fabrika içi lojistikte yaşanabilecek ufak bir aksaklığın bile klasik üretim anlayışıyla çalışan fabrikalara kıyasla üretim sisteminin işleyişini çok ciddi bir biçimde etkileyeceği açıktır. Bu nedenle TZÜ felsefesi ile çalışılan üretim ortamlarında, parça veya hammadde kıtlığı sebebiyle üretimin aksamaması için çok iyi planlanmış güvenilir ve sorunsuz bir fabrika içi tedarik sistemi oldukça kritik bir öneme sahiptir.

Döngüsel seferlerle fabrika içi parça tedariği yapılan endüstriyel uygulamalar incelendiğinde, çekici trenlerin aynı koridorları ve yükleme-boşaltma alanlarını eş zamanlı olarak kullanmaları gerektiğinden birbirlerini bloklayarak trafik sıkışıklığına neden olduğu ve böylece stabil ve güvenilir bir lojistik sisteminin sağlanamadığı

görülmektedir. Özellikle hücresel imalat yerleşimine sahip fabrikalarda, düz ve uzun bir hat şeklinde olan montaj hatlarına göre yatay ve dikey olarak daha karmaşık ve dağınık şekilde birçok koridor bulunduğu için trafik kaynaklı beklemeler ve kilitlenmeler daha yoğun olmaktadır. Bununla birlikte gerçek hayat uygulamalarına ait gözlemler ve literatürden elde edilen bulgular günümüzde fabrika içi döngüsel tedarik rotalarının çoğunlukla üretim planlamacıları tarafından elle çizildiğini göstermektedir. Fabrika içi döngüsel seferlerin rotalanması ve çizelgelenmesi, araç kapasitesi, çevrim süresi gibi birçok kısıt içerdiği için elle planlanması oldukça karmaşık süreçlerdir. Buna ek olarak araçların çakışma durumlarını da dikkate alan bir rotalama ve çizelgeleme planını manuel olarak elde etmek fazlasıyla karmaşık ve vakit gerektiren bir süreç olduğu için pratikte çakışma durumları göz ardı edilmektedir. Ancak bu şekilde ortaya çıkan trafik kaynaklı kilitlenme ve beklemeler üretim ortamında karmaşaya gecikmelere ve verim kaybına sebep olmaktadır.

Bu bağlamda bu tez çalışmasında TZÜ ortamında süpermarket alanlarından üretim hücrelerine sabit çevrim süresine sahip döngüsel seferlerle parça beslemesi yapan çekici trenler için çakışma durumlarını göz önünde bulunduran SÇDRÇ problemi tanımlanmış ve çözüm yöntemleri geliştirilmiştir. SÇDRÇ problemi temel olarak, belirli bir miktar parçanın her periyotta belirli bir süpermarket rafından alınarak belirli bir üretim hücresinin stok rafına teslim edilmesini ifade eden görevlerin çekici trenlere atanmasını, trenlerin birbirleri ile çakışmayacak şekilde rotalanmasını ve çizelgelenmesini içermektedir. Problemde, fabrika içerisinde zamanla farklı sebeplerle meydana gelen değişiklikler nedeniyle iki çekici trenin yan yana geçemeyeceği şekilde daralmış koridorlarda trenlerin karşılıklı veya arka arkaya birbirlerini bloklamaları durumları detaylı şekilde ele alınmış ve incelenmiştir.

Çözüm için probleme özgü olarak üç adet matematiksel programlama modeli ve bu modellere dayanan üç farklı çözüm yaklaşımı geliştirilmiştir. İlk çözüm yaklaşımı, Model-1 olarak adlandırılan karma tam sayılı matematiksel bir modele dayanmaktadır. Bu model; çekici trenlerin dar koridorlarda arka arkaya ve karşılıklı çakışmalarını göz önünde bulundurarak SÇDRÇ problemi için optimum yükleme, rotalama ve çizelgeleme çözümü sunmaktadır. Yol-denge, kapasite, çevrim süresi ve çakışma engelleme gibi birçok kısıt içeren modelin amaç fonksiyonu çekici trenlerin yolda geçirdikleri sürenin ve bekleme sürelerinin toplamının minimize edilmesini içermektedir. Bu modelin

çözümüyle elde edilen planlama ile çakışma olmaksızın çekici trenler mümkün olan en kısa yoldan giderek, en verimli şekilde fabrika içerisindeki taşıma görevlerini yerine getirmektedirler.

Birinci çözüm yaklaşımı her ne kadar çekici trenler için optimum rotalama ve çizelgeleme planı sunsa da, çakışma engelleme için Model-1'in çok fazla sayıda tam sayılı değişken ve kısıt denklemi içermesi sebebiyle küçük boyutlu sayılabilecek gerçek hayat problemleri için bile makul sürelerde çözüm elde edilememektedir. Bu nedenle bu tez çalışmasında doğrudan sezgisel algoritmalara başvurulmadan önce en azından küçük ya da orta boyutlu sayılabilecek gerçek hayat problemlerini makul sürelerde optimum şekilde çözebilecek bir çözüm yaklaşımı geliştirilmiştir. İkinci çözüm yaklaşımı ya da iteratif çözüm yaklaşımı olarak isimlendirilen bu çözüm prosedüründe, Model-1'e ek olarak Model-1'in gevşetilmesiyle elde edilen Model-2 ve süreci hızlandırmak için ayrıca geliştirilen Model-3 adındaki üçüncü bir model kullanılmaktadır.

Bu çözüm prosedüründe, birinci çözüm yaklaşımında olduğu gibi bütün dar koridorlar ve bütün araçlar için çakışma engelleme kısıtları oluşturmak yerine öncelikle çakışmalar göz ardı edilerek Model-2 ile çözüm aranmakta ve alternatif optimum çözümler elde edilmektedir. Daha sonra alternatif optimum çözümlere çakışma kontrolü yapılmakta ve eğer çakışmasız çözüm bulunamazsa sadece çakışma olan koridorlar ve hareketler için çakışma engelleme kısıtları oluşturularak Model-1 çözülmektedir. Elde edilen çözüme tekrar çakışma kontrolü yapılarak eğer çakışmasız çözüm yoksa süreç tekrar edilmekte ve böylelikle iteratif olarak çakışmasız bir çözüm elde edilene kadar devam edilmektedir. Oldukça hızlı olan ancak detaylı bir çizelgeleme planı vermeyen Model-3 ise Model-2'ye alt sınır elde etmek için kullanılmaktadır. Böylelikle Model-2 ile çözüm sırasında çözümün optimalliğini ispatlamak için gereken çok büyük miktarda zamandan tasarruf edilerek süreç hızlandırılmaktadır.

İkinci çözüm yaklaşımıyla belirli bir boyuta kadar olan gerçek hayat problemleri optimum çözüm bulunabilecek şekilde makul sürelerde çözülebilse de problemin NP-zor doğası nedeniyle büyük boyutlu problemler çözülememektedir. Bu nedenle büyük boyutlu problemleri de etkili ve verimli bir şekilde çözebilmek için problemin yapısına özgün olarak komşuluk ve popülasyon temelli bir mat-sezgisel algoritma tasarlanmıştır. Üçüncü çözüm yaklaşımını oluşturan bu algoritma; başlangıç birey oluşturma, birey

geliştirme ve çakışmasız çözüm arama olmak üzere üç bölümden oluşmaktadır. Mat-sezgisel algoritmanın ilk aşamasında, çeşitli kriterlere göre görevlerin önceliklendirilerek sırayla çekici trenlere atanmasına dayanan bir başlangıç birey oluşturma prosedürü yer almaktadır. Bu şeklide oluşturulan başlangıç bireyler rasgele oluşturulan bireylere göre çok daha iyi çözüm değerlerine sahip olduğu için algoritmanın genel performansını ve verimliliğini artırmaktadır.

İkinci aşamada ise, belirli sayıdaki başlangıç bireyden her biri nesiller boyunca evrimsel bir şekilde geliştirilmektedir. Bu aşamada problemin yapısına özgü olarak tasarlanmış komşu oluşturma mekanizmaları ile her bir bireyden yeni bireyler türetilmekte ve üç seçim yönteminden oluşan karma bir seçim stratejisi kullanılarak yeni nesil popülasyon oluşturulmaktadır. Karma seçim stratejisi; elitist, rulet çemberi ve tamamen rasgele seçim yöntemlerini içermektedir. Böylelikle çözüm değeri daha iyi olan bireyin seçilme şansının daha yüksek olduğu ancak kötü çözüm değerine sahip bireylerin de rasgele olarak seçilebildiği bir mekanizma ortaya çıkmaktadır. Mat-sezgisel algoritmanın ilk iki aşamasında çakışma durumları göz ardı edilerek sadece Model-3 kullanılmaktadır. Bu şekilde daha az kaynak harcayarak daha hızlı ve verimli bir arama süreci sağlanmaktadır.

Üçüncü aşamada ise, ikinci aşamada elde edilen geliştirilmiş bireylerin en iyi çözüm değerine sahip olanından başlanarak çekici trenler için mümkün olan en iyi çözüm değerine sahip çakışmasız bir rotalama ve çizelgeleme planı aranmaktadır. Bu aşamada Model-2 ve Model-1 kullanılarak çekici trenler birbirlerine göre çakışma oluşturmayacak şekilde rotalanmakta ve çizelgelenmektedir.

Literatürde SÇDRÇ probleminin yapısına uygun test problemleri bulunmadığından bu çalışma kapsamında geliştirilen çözüm yöntemlerini test etmek için gerçek hayat problemlerinden esinlenerek rassal bir şekilde deneme problemleri türetilmiştir. Problem büyüklüğünü ve geliştirilen çözüm yaklaşımlarının performansını etkilediği düşünülen kroki büyüklüğü ve yapısı, görev sayısı, dar yol oranı ve araç sayısı parametrelerine bağlı olarak toplamda 60 adet test problemi oluşturulmuştur. Test problemlerinin her biri önerilen çözüm yaklaşımları ile 12 saatlik süre boyunca çözülmüş ve farklı büyüklükteki problemler için elde edilen sonuçlar detaylı bir şekilde sunularak tartışılmıştır.

Model-1'e dayalı birinci çözüm yaklaşımının, verilen süre içerisinde sadece küçük boyutlu kroki kullanılarak oluşturulan test problemlerinin en az değişken ve kısıt içeren

10 tanesine çözüm bulabildiği görülmektedir. Orta boyutlu krokiler ile oluşturulmuş test problemlerin de ise hiç bir şekilde optimalliği kanıtlanmış şekilde çözüm değerleri elde edilememiştir. Büyük boyutlu krokiler kullanılarak oluşturulan problemlerde Model-1, yetersiz bellek uyarısı nedeniyle yüklenememiştir. Bu sonuçlar ile birinci çözüm yaklaşımının küçük sayılabilecek gerçek hayat problemleri için bile kullanışsız olduğu ve makul sürelerde çözüm bulunamayacağı ifade edilebilir.

İkinci çözüm yaklaşımı olan iteratif çözüm prosedürü ile, küçük ve orta boyutlu krokilerle oluşturulan 36 adet test probleminin ikisi haricinde optimum çözüm değerleri birinci yaklaşıma göre oldukça hızlı bir şekilde elde edilmiştir. Büyük boyutlu problemler için çözüm veremeyen iteratif çözüm yaklaşımının orta büyüklükte sayılabilecek fabrika alanları için 6 araca kadar birçok problemi rahatlıkla makul sürelerde çözebileceği ifade edilebilir.

Test problemlerinin sonuçlarından hareketle araç sayısının artmasına paralel olarak iteratif çözüm yaklaşımının çözüm bulma süresinin uzadığı söylenebilir. Problemlerin çözüm aşamaları incelendiğinde bunun temel sebeplerinden bir tanesinin, araç sayısının artmasına paralel olarak Model-3 ile alt tursuz çözüm arama sürecinin uzaması olduğu görülmüştür. Ayrıca artan dar yol oranının, Model-2 ile elde edilen alternatif çözümler içerisinden çakışmasız optimum rotalama ve çizelgeleme planının elde edilmesi olasılığını düşürdüğü görülmektedir. Çakışma analizi ile özellikle alma koridorlarında oluşan çakışmaların bu sonuçlar üzerinde daha etkili olduğu bilgisi elde edilmiştir.

Büyük boyutlu gerçek hayat problemlerini makul sürelerde efektif bir biçimde çözebilmek için geliştirilen mat-sezgisel algoritma ile elde edilen sonuçlara bakıldığında, optimum çözüm değeri iteratif çözüm yaklaşımı ile bulunan küçük ve orta boyutlu problemlerin bir çoğunda optimum çözüm değerinin, diğerlerinde ise optimuma çok yakın değerlerin elde edildiği görülmektedir. Bu problemler için optimumdan sapma değerlerinin en büyüğü, %3,34, en küçüğü %0, ortalaması ise %0,6 olarak elde edilmiştir. Çözüm süreleri açısından bakıldığında ise bu problemler için mat-sezgisel algoritma ile elde edilen ortalama çözüm süresinin iteratif çözüm yaklaşımı ile edilenin üçte birinden daha az olduğu görülmektedir. Bu sonuçlardan hareketle mat-sezgisel algoritmanın küçük ve orta boyutlu problemler için oldukça verimli ve etkili olduğu açıktır.

Her bir test problemi için mat-sezgisel algoritma ile 5 tekrarlı koşum sonucunun ortalama değerleri incelendiğinde, çoğu test problemi için ortalama değer ile en iyi çözüm değerinin aynı olduğu, diğer problemlerde ise bu değerlerin birbirlerine oldukça yakın oldukları görülmektedir. Dolayısıyla algoritmanın tutarlı sonuçlar verdiği ifade edilebilmektedir.

Büyük boyutlu test problemlerinde ise üçüncü çözüm yaklaşımı ile elde edilen sonuçların kıyaslanabileceği optimum çözüm değerleri bilinmemektedir. Bu nedenle bütün test problemleri için mat-sezgisel algoritmanın çözüm performansını ifade eden çeşitli performans göstergeleri hesaplanmıştır. Bu göstergeler temel olarak, başlangıç çözüm bireylerinin algoritma sonunda ne kadar iyileştirildiği bilgisine dayanmaktadır. EKBÇD iyileştirme oranı, BÇDO iyileştirme oranı ve EIBÇD iyileştirme oranı bu çalışmada kullanılan performans göstergeleridir.

Mat-sezgisel algoritmanın bu üç performans göstergesinin her birinin, optimum değeri bilinen orta boyutlu test problemlerindeki ortalaması ile optimum değeri bilinmeyen büyük boyutlu test problemlerindeki ortalamasının birbirlerine oldukça yakın olduğu görülmüştür. Bu veriler, mat-sezgisel algoritmanın büyük boyutlu problemler için de, optimum veya optimuma çok yakın çözüm değerlerini bulabildiği sonucu çıkarmak için ipucu sunmaktadır. Ancak bu ortalama değerlerin yakın çıkması tesadüfilikten de kaynaklanabileceği için anlamlı olup olmadıkları istatistiksel olarak analiz edilmiş ve araştırılmıştır.

BÇDO iyileştirme oranı açısından, optimum çözüm değeri bilinen 22 adet orta boyutlu test problemi ile optimum değeri bilinmeyen 24 adet büyük boyutlu test problemi arasında anlamalı bir farklılık olup olmadığı Mann-Whitney U testi ile incelenmiştir. Test sonucunda BÇDO iyileştirme oranı açısından orta ve büyük boyutlu problemler arasında istatistiksel olarak anlamlı bir fark olmadığı sonucuna varılmıştır. Dolayısıyla mat-sezgisel algoritmanın orta boyutlu problemler için gösterdiği performansı büyük boyutlu problemler için de gösterdiği ve optimuma çok yakın çözüm değerleri elde edebildiği çıkarımını yapmak mümkündür.

Çalışma kapsamında mat-sezgisel çözüm sonuçlarına göre her bir test problemleri için araç kapasitesi kullanım oranı ve genel kapasite kullanım oranı hesaplanmıştır. Grafiklerle ayrıntılı olarak sunulan bu değerler incelendiğinde mat-sezgisel algoritma ile

çekici trenler için elde edilen yükleme, rotalama ve çizelgeleme çözümlerinde genel olarak iş yükü dağılımının dengeli olduğu görülmektedir.

Son olarak, çözüm yaklaşımlarının performansı üzerinde büyük bir etkisi olduğu görülen çakışma durumları detaylı olarak analiz edilmiş ve test problemleri çözülürken genel olarak hangi koridor tiplerinde ve hangi krokilerde çakışmaların daha yoğun olduğu araştırılmıştır. Bu analiz sonucunda alma koridorlarında bırakma koridorlarına göre çok daha fazla sayıda çakışmalara rastlandığı görülmüştür. Bu durumun, süpermarket raflarının imalat hücrelerine göre genellikle daha küçük bir alanda daha yoğun bir şekilde bulunmasından kaynaklandığı düşünülmektedir.

Bu noktadan hareketle, yeni bir fabrika tasarlarken tasarımcıların ve mühendislerin öncelikle süpermarket raflarının bulunduğu alma koridorlarını, daha sonra mümkünse üretim hücrelerinin etrafındaki bırakma koridorlarını iki çekici trenin yan yana geçebileceği şekilde geniş tutmalarının daha efektif ve sorunsuz bir fabrika içi lojistik sistemi için oldukça önemli olduğu ifade edilebilir. Diğer taraftan mevcut bir fabrikada çeşitli sebeplerle yerleşim düzeninin değişmesi gerektiği durumlarda öncelikle alma koridorlarının daha sonra da mümkünse bırakma koridorlarının iki çekici trenin yan yana geçemeyeceği şekilde daraltılmaması, fabrika içerisinde trafik kaynaklı gecikmelere ve verim kaybına neden olacak bir lojistik sisteminden kaçınabilmeyi sağlayacaktır.

Çakışma yoğunluğu analizinden elde edilen diğer bir bulgu ise, kare şekilli krokilerin çözümü sırasında, dikdörtgen krokilere göre hem orta hem de büyük boyutlu test problemlerinde daha fazla çakışma tespit edilmesidir. Bu durum, kare şekilli alanda çekici trenlerin daha fazla ortak görev koridorlarına sahip olmasından kaynaklanmaktadır. Bu noktada karar vericilere sunulabilecek tavsiye, yeni bir fabrika kurulumunda mümkünse dikdörtgen şekilli bir alan seçmeleri ve fabrika yerleşimlerini dikdörtgen bir formda oluşturmaları olacaktır.

Gelecek çalışmalarda, SÇDRÇ probleminin çözümüne yönelik olarak bu tezde tasarlanan mat-sezgisel algoritmanın geliştirilmesi konusunda farklı düşünceler öne sürülebilir. Örneğin mat-sezgisel algoritmanın parametrelerini daha iyi ayarlamak ve daha verimli sonuçlar elde etmek için deney tasarımı yöntemi kullanılabilir. Ayrıca bu çalışmada tanımlanan problemin çözümü için daha verimli algoritmalar ya da modeller geliştirilebilir.

KAYNAKLAR

Akillioglu, H., Baydogan, M.G., Bolatli, Y, Canbaz, D., Halici A., Sezgin, O., Ozdemirel, N.E., Turkcan, A. (2006). Dizel enjektör üretimi yapan bir şirket için fabrika içi çekme esasli tekrarli dağitim sistemi tasarımı. *Endüstri Mühendisliği Dergisi, 17*(3), 2-15.

Alizon, F., Dallery, Y., Essafi, I., Feillet, D. (2009). Optimizing material handling costs in an assembly workshop. *Int J Prod Res, 47*(14), 3853–3866.

Alnahhal, M., Noche, B. (2013). Efficient material flow in mixed model assembly lines. *SpringerPlus, 2*, 1-12.

Alnahhal, M., Noche, B. (2015). A genetic algorithm for supermarket location problem. *Assembly Automation, 35*(1), 122-127.

Atayah, O.F., Dhiaf, M.M., Najaf, K., Frederico, G.F. (2022). Impact of COVID-19 on financial performance of logistics firms: evidence from G-20 countries. *Journal of Global Operations and Strategic Sourcing, 15*(2), 172-196.

Aydoğdu, B., Özyörük, B. (2020). Dinamik eş zamanlı topla dağıt araç rotalama probleminin çözümü için matematiksel model ve sezgisel yaklaşım: Rassal iteratif yerel arama değişken komşu iniş algoritması. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 35*(2), 563-580.

Azadeh, A., Anvari, M., Ziaei, B., Sadeghi, K. (2010). An integrated fuzzy DEA–fuzzy C-means–simulation for optimization of operator allocation in cellular manufacturing systems. *The International Journal of Advanced Manufacturing Technology, 46*, 361-375.

Baller, R., Hage, S., Fontaine, P., Spinler, S. (2020). The assembly line feeding problem: An extended formulation with multiple line feeding policies and a case study. *International Journal of Production Economics, 222*, 107489.

Battini, D., Boysen, N., Emde, S. (2013). Just-in-Time supermarkets for part supply in the automobile industry. *Journal of Management Control, 24*, 209-217.

Battini, D., Faccio, M., Persona, A., Sgarbossa, F. (2009). Design of the optimal feeding policy in an assembly system. *International Journal of Production Economics, 121*(1), 233-254.

Battini, D., Faccio, M., Persona, A., Sgarbossa, F. (2010). Supermarket warehouses: stocking policies optimization in an assembly-to-order environment. *The International Journal of Advanced Manufacturing Technology, 50*, 775-788.

Battini, D., Gamberi, M., Persona, A., Sgarbossa, F. (2015). Part-feeding with supermarket in assembly systems: transportation mode selection model and multi-scenario analysis. *Assembly Automation, 35*(1), 149-159.

Belekoukias, I., Garza-Reyes, J.A., Kumar, V. (2014). The impact of lean methods and tools on the operational performance of manufacturing organisations. *International Journal of production research, 52*(18), 5346-5366.

Bocewicz, G., Nielsen, P., Zbigniew, B. (2019). Milk-run routing and scheduling subject to different pick-up/delivery profiles and congestion-avoidance constraints. *IFAC-PapersOnLine*, *52*(8), 313-320.

Boschetti, M.A., Maniezzo, V. (2022). Matheuristics: using mathematics for heuristic design. *4OR*, *20*(2), 173-208.

Boysen, N., Emde, S., Hoeck, M., Kauderer, M. (2015). Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research*, *242*(1), 107-120.

Bozer, Y. A., Ciemnoczolowski, D. D. (2013). Performance evaluation of small-batch container delivery systems used in lean manufacturing–Part 1: system stability and distribution of container starts. *International Journal of Production Research, 51*(2), 555-567.

Briand, C., He, Y., Ngueveu, S.U. (2018). Energy-efficient planning for supplying assembly lines with vehicles. *EURO Journal on Transportation and Logistics, 7*, 387-414.

Buyukozkan, K., Bal, A., Oksuz, M.K., Kapukaya, E.N., Satoglu, S.I. (2019). A Mathematical Model and a Matheuristic for In-Plant Milk-Run Systems Design and Application in White Goods Industry. Calisir, F., Akdag, H.C., Cevikcan, E., (Ed.), *Industrial Engineering in the Big Data Era* (1st ed.) (99-112), Switzerland: Springer.

Caputo, A.C., Pelagagge, P.M. (2011). A methodology for selecting assembly systems feding policy. *Industrial Management & Data Systems, 111*(1), 84–112.

Caputo, A.C., Pelagagge, P.M., Salini, P. (2015a). Planning models for continuous supply of parts in assembly systems. *Assembly Automation, 35*(1), 35–46.

Caputo, A.C., Pelagagge, P.M., Salini, P. (2015b). A decision model for selecting parts feeding policies in assembly lines. *Industrial Management & Data Systems, 115*(6), 974-1003.

Caputo, A.C., Pelagagge, P.M., Salini, P. (2018). Selection of assembly lines feeding policies based on parts features and scenario conditions. *International Journal of Production Research, 56*(3), 1208-1232.

Cerf, R. (2017). The quasispecies regime for the simple genetic algorithm with roulette wheel selection. *Advances in Applied Probability*, *49*(3), 903-926.

Chen, W., Rahman, H.F., Zhou, Q., Liu, S., Liu, H., Qi, E. (2023). Proactive in-house part-feeding for mixed-model assembly systems with dynamics. *Computers & Industrial Engineering, 178*, 109101.

Cichosz, M., Wallenburg, C.M., Knemeyer, A.M. (2020). Digital transformation at logistics service providers: barriers, success factors and leading practices. *The International Journal of Logistics Management, 31*(2), 209-238.

Coelho, F., Relvas, S., Barbosa-Póvoa, A.P. (2021). Simulation-based decision support tool for in-house logistics: the basis for a digital twin. *Computers & Industrial Engineering, 153*, 107094.

Crespo de Carvalho, J., Silva, J., O'Neill, H. (2014). Logistics and supply chain management: an area with a strategic service perspective. *American Journal of Industrial and Business Management, 4*, 24-30.

Della Croce, F., Salassa, F. (2014). A variable neighborhood search based matheuristic for nurse rostering problems. *Annals of Operations Research, 218*(1), 185-199.

Duman, N.O. (2016). Fabrika İçi Stok Rotalama Problemi İçin Filo Yapısı ve Çevrimsel Taşıma Çizelgelerinin Belirlenmesi. Yüksek Lisans Tezi, TOBB Ekomoni ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 436173.

Efthymiou, O.K., Ponis, S.T. (2019). Current status of industry 4.0 in material handling automation and in-house logistics. *International Journal of Industrial and Manufacturing Engineering, 13*(10), 1368-1372.

Egilmez, G., Süer, G.A., Huang, J. (2012). Stochastic cellular manufacturing system design subject to maximum acceptable risk level. *Computers & Industrial Engineering, 63*(4), 842-854.

Emde, S., Boysen, N. (2012a). Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *European Journal of Operational Research, 217*, 287–299.

Emde, S., Boysen, N. (2012b). Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *International Journal of Production Economics, 135*, 393–402.

Emde, S., Gendreau, M. (2017). Scheduling in-house transport vehicles to feed parts to automotive assembly lines. *European Journal of Operational Research, 260*(1), 255–267.

Emde, S., & Schneider, M. (2018). Just-in-time vehicle routing for in-house part feeding to assembly lines. *Transportation Science, 52*(3), 657-672.

Emde, S., Fliedner, M., Boysen, N. (2012). Optimally loading tow trains for JIT-supply of mixed-model assembly lines. *IIE Transactions, 44*, 121–135.

Faccio, M. (2014). The impact of production mix variations and models varieties on the parts-feeding policy selection in a JIT assembly system. *The International Journal of Advanced Manufacturing Technology, 72*, 543-560.

Faccio, M., Gamberi, M., Persona, A., Regattieri, A., Sgarbossa, F. (2013). Design and simulation of assembly line feeding systems in the automotive sector using supermarket, kanbans and tow trains: a general framework. *Journal of Management Control, 24*, 187-208.

Fathi, M., Ghobakhloo, M. (2020). Enabling mass customization and manufacturing sustainability in industry 4.0 context: A novel heuristic algorithm for in-plant material supply optimization. *Sustainability, 12*(16), 6669.

Fathi, M., Alvarez, M.J., Hassani Mehraban, F., Rodríguez, V. (2014a). A multi-objective optimization algorithm to solve the part feeding problem in mixed-model assembly lines. *Mathematical Problems in Engineering, 2014*, 654053.

Fathi, M., Nourmohammadi, A., Ghobakhloo, M., Yousefi, M. (2020). Production sustainability via supermarket location optimization in assembly lines. *Sustainability, 12*(11), 4728.

Fathi, M., Rodríguez, V., Alvarez, M.J. (2014b). A novel memetic ant colony optimization-based heuristic algorithm for solving the assembly line part feeding problem. *The International Journal of Advanced Manufacturing Technology, 75*(4), 629-643.

Fathi, M., Rodríguez, V., Fontes, D. B., Alvarez, M. J. (2016). A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines. *International Journal of Production Research, 54*(3), 878-893.

Fathi, M., Syberfeldt, A., Ghobakhloo, M., Eskandari, H. (2018). An optimization model for material supply scheduling at mixed-model assembly lines. *Procedia Cirp, 72*, 1258-1263.

Fonseca, G.H., Santos, H.G., Carrano, E.G. (2016). Integrating matheuristics and metaheuristics for timetabling. *Computers & Operations Research, 74*, 108-117.

Francuz, Á., Bányai, T. (2022). Optimisation of Milkrun routes in manufacturing systems in the automotive industry. *Cutting & Tools in Technological System, 96*, 32-41.

Fullerton, R.R., McWatters, C.S., Fawson, C. (2003). An examination of the relationships between JIT and financial performance. *Journal of Operations Management, 21*(4), 383-404.

García-Alcaraz, J.L., Maldonado-Macías, A.A. (2016). *Just-in-Time Elements and Benefits* (1st ed.). Switzerland: Springer International Publishing.

Golz, J., Gujjula, R., Günther, H.O., Rinderer, S. (2012). Part feeding at high-variant mixed-model assembly lines. *Flexible Services and Manufacturing Journal, 24*, 119–141.

González-Hernández, H.G., Reyes-Avendaño, J.A., Mora-Salinas, R.J., García-Rodríguez, A., Castillo-García, J.G., García-Toxqui, G. (2022). Automation of a tow-tractor for the autonomous delivery of materials in an industrial complex.

International Journal on Interactive Design and Manufacturing (IJIDeM), 16(2), 503-516.

Gunasekaran, A., Subramanian, N., Papadopoulos, T. (2017). Information technology for competitive advantage within logistics and supply chains: A review. *Transportation Research Part E: Logistics and Transportation Review, 99*, 14-33.

Gupta, A. (2011) A conceptual JIT model of service quality. *Int J Eng Sci Technol 3*(3), 2414–2427.

Gülenç, İ.F. Karagöz, B. (2008). E-Lojistik ve Türkiye'de E-Lojistik Uygulamaları. *Kocaeli Üniversitesi Sosyal Bilimler Dergisi, 15*, 73-91.

Gyulai, D., Pfeiffer, A., Sobottka, T., Váncza, J. (2013). Milkrun vehicle routing approach for shop-floor logistics. *Procedia Cirp, 7*, 127-132.

Hanson, R., Brolin, A. (2013). A comparison of kitting and continuous supply in in-plant materials supply. *International Journal of Production Research, 51*(4), 979-992.

Hanson, R., Finnsgård, C. (2014). Impact of unit load size on in-plant materials supply efficiency. *International Journal of Production Economics, 147*, 46-52.

Harrison, A., Skipworth, H., van Hoek, R.I., Aitken, J. (2019). *Logistics management and strategy* (1st ed.). United Kingdom: Pearson.

Hasani Goodarzi, A., Zegordi, S.H. (2020). Vehicle routing problem in a kanban controlled supply chain system considering cross-docking strategy. *Operational Research, 20*, 2397-2425.

Htun, A., Maw, T.T., Khaing, C. (2019). Lean manufacturing, just in time and Kanban of Toyota production system (TPS). *International Journal of Scientific Engineering and Technology Research, 8*(1), 469-474.

Hua, S.Y., Johnson, D.J. (2010). Research issues on factors influencing the choice of kitting versus line stocking. *International Journal of Production Research, 48*(3), 779-800.

Hwang, H.S. (2004). Heuristic transporter routing model for manufacturing facility design. *Comput Ind Eng 46*, 243–251.

Jin, W., He, Z., Peng, T. (2021). A JIT part supply scheduling strategy with electric transport device between central receiving store and supermarkets in the automobile industry considering energy. *eTransportation, 9*, 100126.

Karadayi Usta, S., Oksuz, M.K., Durmusoglu, M.B. (2017). Design methodology for a hybrid part feeding system in lean-based assembly lines. *Assembly Automation, 37*(1), 84–102.

Kılıç, H.S., Durmuşoğlu, M.B. (2013). A mathematical model and a heuristic approach for periodic material delivery in lean production environment. *The International Journal of Advanced Manufacturing Technology, 69*, 977-992.

Kılıç, H.S., Durmuşoğlu, M.B. (2015). Advances in assembly line parts feeding policies: A literature review. *Assembly Automation, 35*(1), 57–68.

Kılıç, H.S., Durmuşoğlu, M.B., Baskak, M. (2012). Classification and modeling for in-plant milk-run distribution systems. *The International Journal of Advanced Manufacturing Technology, 62*, 1135-1146.

Kluska, K., Pawlewski, P. (2018). The use of simulation in the design of Milk-Run intralogistics systems. *IFAC-PapersOnLine, 51*(11), 1428-1433.

Kramer, R., Subramanian, A., Vidal, T., Lucídio dos Anjos, F.C. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research, 243*(2), 523-539.

Kumar, V. (2010). JIT Based Quality Management: Concepts and Implications in Indian Context. *International Journal of Engineering Science and Technology 2*(1), 40–50.

Küçükoğlu, İ., Yağmahan, B., Çağlıyan, M.S., Yıldız, A., Aktokluk, D. (2018). İç lojistik sisteminde malzeme tedariği için geliştirilmiş matematiksel modelleme yaklaşımı: bir uygulama. *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi, 23*(4), 159-176.

Lahyani, R., Chebil, K., Khemakhem, M., Coelho, L.C. (2019). Matheuristics for solving the multiple knapsack problem with setup. *Computers & industrial engineering, 129*, 76-89.

Levner, E., Kats, V., De Pablo, D. A.L., Cheng, T.E. (2010). Complexity of cyclic scheduling problems: A state-of-the-art survey. *Computers & Industrial Engineering*, *59*(2), 352-361.

Limère, V., Landeghem, H.V., Goetschalckx, M., Aghezzaf, E.H., McGinnis, L.F. (2012). Optimising part feeding in the automotive assembly industry: deciding between kitting and line stocking. *International Journal of Production Research, 50*(15), 4046-4060.

Limère, V., Van Landeghem, H., Goetschalckx, M. (2015). A decision model for kitting and line stocking with variable operator walking distances. *Assembly Automation, 35*(1), 47-56.

Lu, Z., Zhu, H., Han, X., Hu, X. (2019). Integrated modelling and algorithm of material delivery and line-side storage for aircraft moving assembly lines. *International Journal of Production Research, 57*(18), 5842-5856.

Maiti, A.K., Maiti, M. (2008). Discounted multi-item inventory model via genetic algorithm with Roulette wheel selection, arithmetic crossover and uniform mutation

in constraints bounded domains. *International Journal of Computer Mathematics, 85*(9), 1341-1353.

Mao, Z., Huang, D., Fang, K., Wang, C., Lu, D. (2020). Milk-run routing problem with progress-lane in the collection of automobile parts. *Annals of Operations Research, 291*, 657-684.

Marialuisa, M., Matteo M.S., Muhammad, S. (2023). Optimizing milk-run system and IT-based Kanban with artificial intelligence: an empirical study on multi-lines assembly shop floor. *Production & Manufacturing Research, 11*(1), 2179123.

Martinez-Sykora, A., Alvarez-Valdés, R., Bennell, J.A., Ruiz, R., Tamarit, J.M. (2017). Matheuristics for the irregular bin packing problem with free rotations. *European Journal of Operational Research, 258*(2), 440-455.

McLachlin, R. (1997). Management initiatives and just-in-time manufacturing. *Journal of Operations management, 15*(4), 271-292.

Murugan, M., Selladurai, V. (2007). Optimization and implementation of cellular manufacturing system in a pump industry using three cell formation algorithms. *The International Journal of Advanced Manufacturing Technology, 35*, 135-149.

Müllerklein, D., Fontaine, P., Ostermeier, F. (2022). Integrated consideration of assembly line scheduling and feeding: A new model and case study from the automotive industry. *Computers & Industrial Engineering, 170*, 108288.

Negahban, A., Smith, J.S. (2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of manufacturing systems, 33*(2), 241-261.

Negri, E., Perotti, S., Fumagalli, L., Marchet, G., Garetti, M. (2017). Modelling internal logistics systems through ontologies. *Computers in Industry, 88*, 19–34.

Neto, A.R.P., Filho, E.V.G. (2010). A simulation-based evolutionary multiobjective approach to manufacturing cell formation. *Computers & Industrial Engineering, 59*(1), 64-74.

Nourmohammadi, A., Eskandari, H., Fathi, M., Bourani, M.R. (2018a). An integrated model for cost-oriented assembly line balancing and parts feeding with supermarkets. *Procedia CIRP, 72*, 381-385.

Nourmohammadi, A., Eskandari, H., Fathi, M., Aghdasi, M. (2018b). A mathematical model for supermarket location problem with stochastic station demands. *Procedia CIRP, 72*, 444-449.

Peng, T., Zhou, B. (2018). Scheduling multiple servers to facilitate just-in-time part-supply in automobile assembly lines. *Assembly Automation, 38*(3), 347-360.

Peng, Y., Zeng, T., Han, Y., Xia, B. (2020). Scheduling just-in-time transport vehicles to feed parts for mixed model assembly lines. *Discrete Dynamics in Nature and Society, 2020*, 2939272.

Pitchuka, L.N., Adil, G.K., Ananthakumar, U. (2006). Effect of conversion of functional layout to a cellular layout on the queue time performance: some new insights. *The International Journal of Advanced Manufacturing Technology, 31*, 594-601.

Queiroga, E., Sadykov, R., Uchoa, E. (2021). A POPMUSIC matheuristic for the capacitated vehicle routing problem. *Computers & Operations Research, 136*, 105475.

Raminfar, R., Zulkifli, N., Vasili, M., Sai Hong, T. (2013). An integrated model for production planning and cell formation in cellular manufacturing systems. *Journal of Applied Mathematics, 2013*, 487694.

Rao, Y.Q., Wang, M.C., Wang, K.P. and Wu, T.M. (2013). Scheduling a single vehicle in the just-in-time part supply for a mixed-model assembly line. *Computers & Operations Research, 40*(11) 2599-2610.

Reeb, J.E., Baker, E.S., Brunner, C.C., Funck, J.W., Reiter, W.F. (2010). Using simulation to select part-families for cell manufacturing. *International Wood Products Journal, 1*(1), 43-47.

Rushton, A. Croucher, P. Baker, P. (2006). *The Handbook of Logistics and Distribution Management* (1st ed.). London: Kogan Page.

Saez-Mas, A., Garcia-Sabater, J.P., Garcia-Sabater, J.J., Ruiz, A. (2020). Redesigning the in-plant supply logistics: a case study. *Computers & Industrial Engineering, 143*, 106422.

Sali, M., Sahin, E. (2016). Line feeding optimization for Just in Time assembly lines: An application to the automotive industry. *International Journal of Production Economics, 174*, 54-67.

Sali, M., Sahin, E., Patchong, A. (2015). An empirical assessment of the performances of three line feeding modes used in the automotive sector: line stocking vs. kitting vs. sequencing. *International Journal of Production Research, 53*(5), 1439-1459.

Sartori, C.S., Buriol, L.S. (2020). A study on the pickup and delivery problem with time windows: Matheuristics and new instances. *Computers & Operations Research, 124*, 105065.

Satoglu, S.I., Sahin, I.E. (2013). Design of a just-in-time periodic material supply system for the assembly lines and an application in electronics industry. *International Journal of Advanced Manufacturing Technology, 65*, 319–332.

Satoğlu, Ş.I., Sipahioğlu, A. (2018). An assignment based modelling approach for the inventory routing problem of material supply systems of the assembly lines. *Sigma Journal of Engineering and Natural Sciences, 36*(1), 161-177.

Schermer, D., Moeini, M., Wendt, O. (2019). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies, 106*, 166-204.

Schmid, N.A., Limère, V. (2019). A classification of tactical assembly line feeding problems. *International Journal of Production Research, 57*(24), 7586-7609.

Shanker, R., Vrat, P. (1999). Some design issues in cellular manufacturing using the fuzzy programming approach. *International Journal of Production Research, 37*(11), 2545-2563.

Sipahioğlu, A., Altın, İ. (2019). A mathematical model for in-plant Milk-Run routing. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 25*(9), 1050-1055.

Staab, T., Klenk, E., Galka, S., Günthner, W.A. (2016). Efficiency in in-plant milk-run systems—The influence of routing strategies on system utilization and process stability. *J. Simulation 10*(2):137–143.

Subramanian, A., Drummond, L.M., Bentes, C., Ochi, L.S., Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 37*(11), 1899-1911.

Şeker, A. (2016). Yalin üretim sisteminde kanban, tek parça akişi ve U tipi yerleştirme sistemleri. *The Journal of Academic Social Science Studies, 9*(50), 449-470.

Tan, K.C., Lee, L.H., Zhu, Q.L., Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial intelligence in Engineering, 15*(3), 281-295.

Tang, C.S., Veelenturf, L.P. (2019). The strategic role of logistics in the industry 4.0 era. *Transportation Research Part E: Logistics and Transportation Review, 129*, 1-11.

Tariq, A., Hussain, I., Ghafoor, A. (2009). A hybrid genetic algorithm for machine-part grouping. *Computers & Industrial Engineering, 56*(1), 347-356.

Thanou, E., Matopoulos, A. (2021). Improving efficiency of material flows in an automotive assembly plant: A case study. *CIRP Journal of Manufacturing Science and Technology, 35*, 959-967.

URL-1: https://www.lean.org/lexicon-terms/cell/, (Ziyaret Tarihi: 20 Temmuz 2023).

URL-2: https://www.flexqube.com/news/transition-from-forklifts-to-tugger-train-system/#:~:text=What%20Is%20A%20Tugger%20Train,operator%2C%20acting%20as%20a%20locomotive, (Ziyaret Tarihi: 15 Temmuz 2023).

Usuk, M.S., Selvi, İ.H. (2019). Fabrika içi lojistik sürecinde kablosuz acil parça istek sistemi otomasyonu: otomotiv fabrikası uygulaması. *Acta Infologica, 3*(1), 1-12.

Uysal, M., Özcan, U. (2019). Süpermarket Yerleşim Problemi İçin Tavlama Benzetimi Algoritması Yaklaşımı. *Karadeniz Fen Bilimleri Dergisi, 9*(1), 58-69.

Vaidyanathan, B.S. , Matson, J.O. , Miller, D.M., Matson, J.E. (1999). A capacitated vehicle routing problem for just-in-time delivery. *IIE Transactions, 31*, 1083–1092.

Wang, J.X. (2015). *Cellular Manufacturing: Mitigating Risk and Uncertainty* (1st ed.). US: CRC Press Taylor and Francis Group.

Yener, F., Yazgan, H.R. (2019). Optimal warehouse design: Literature review and case study application. *Computers & Industrial Engineering, 129*, 1–13.

Zangaro, F., Minner, S., Battini, D. (2021). A supervised machine learning approach for the optimisation of the assembly line feeding mode selection. *International Journal of Production Research, 59*(16), 4881-4902.

Zenker, M., Emde, S., Boysen, N. (2016). Cyclic inventory routing in a line-shaped network. *European Journal of Operational Research, 250*(1), 164-178.

Zheng, Y., Qiu, S., Shen, F., He, C. (2020). RFID-based material delivery method for mixed-model automobile assembly. *Computers & Industrial Engineering, 139*, 106023.

Zhou, B.H., Shen, C.Y. (2018). Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration. *Journal of Cleaner Production, 192*, 293-305.

Zhou, B.H., Tan, F. (2020). A self-adaptive estimation of distribution algorithm with differential evolution strategy for supermarket location problem. *Neural Computing and Applications, 32*, 5791-5804.

Zhou, B., Peng, T. (2017). Scheduling the in-house logistics distribution for automotive assembly lines with just-in-time principles. *Assembly Automation, 37*(1), 51–63.

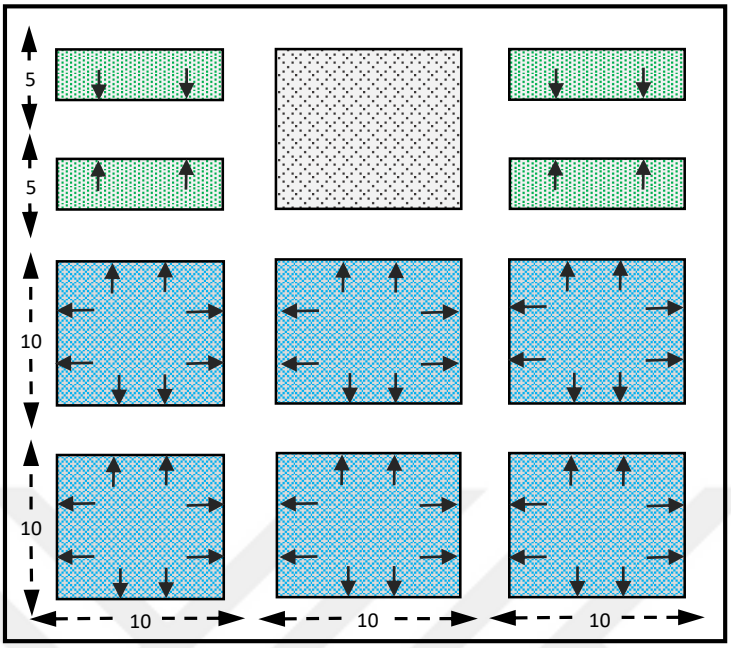
Zhou, B., Peng, T. (2019). Scheduling just-in-time part replenishment of the automobile assembly line with unrelated parallel machines. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233*(14), 5113-5130.

Zhou, B., Zhao, L. (2022). A multi-objective decomposition evolutionary algorithm based on the double-faced mirror boundary for a milk-run material feeding scheduling optimization problem. *Computers & Industrial Engineering, 171*, 108385.

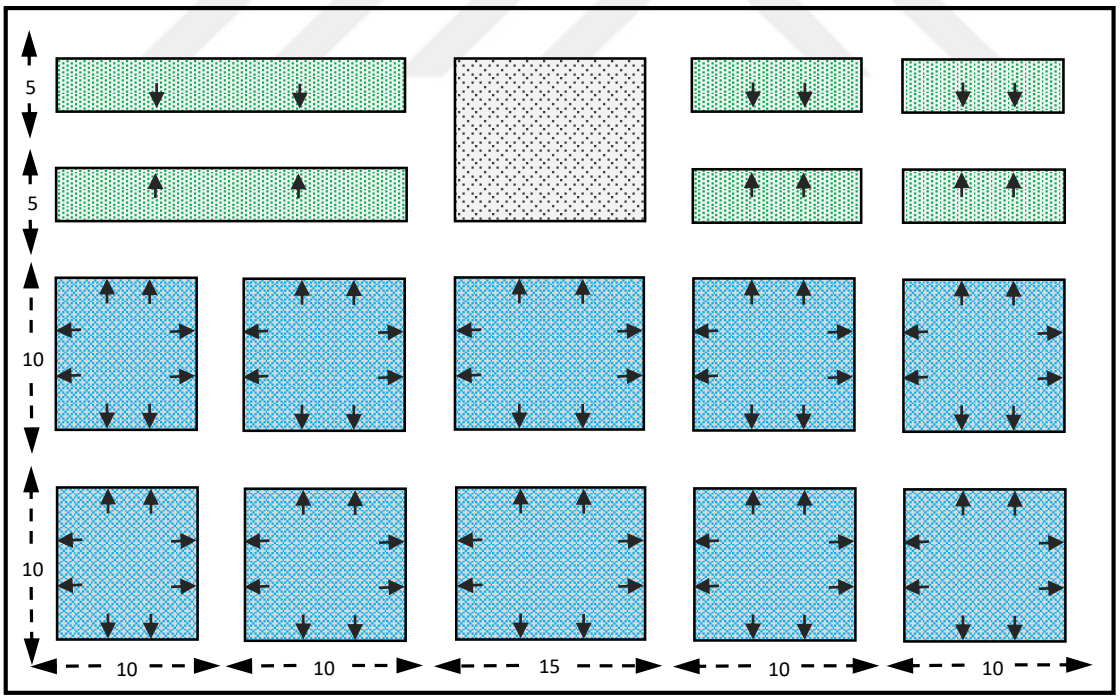
Zhou, B., Zhu, Z. (2020). Optimally scheduling and loading tow trains of in-plant milk-run delivery for mixed-model assembly lines. *Assembly Automation, 40*(3), 511-530.

EKLER

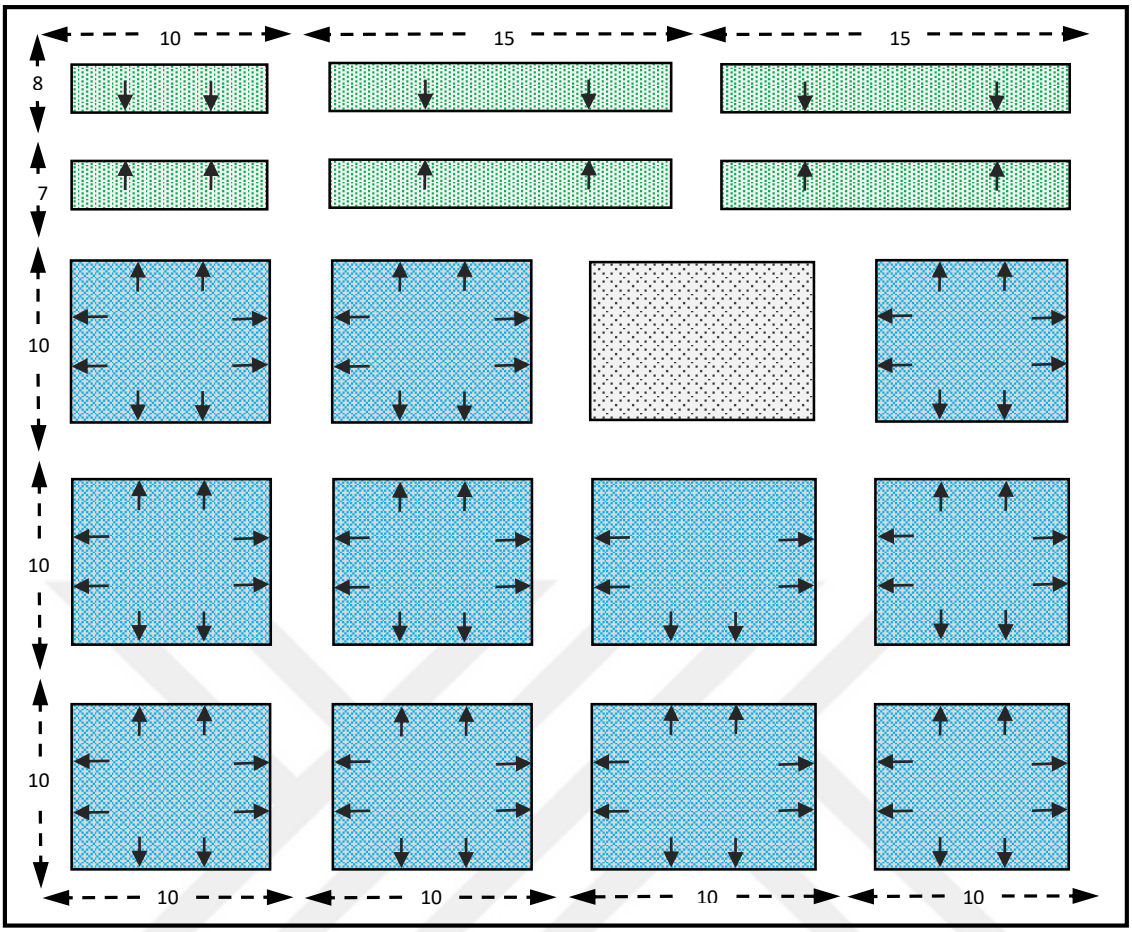
Ek-A



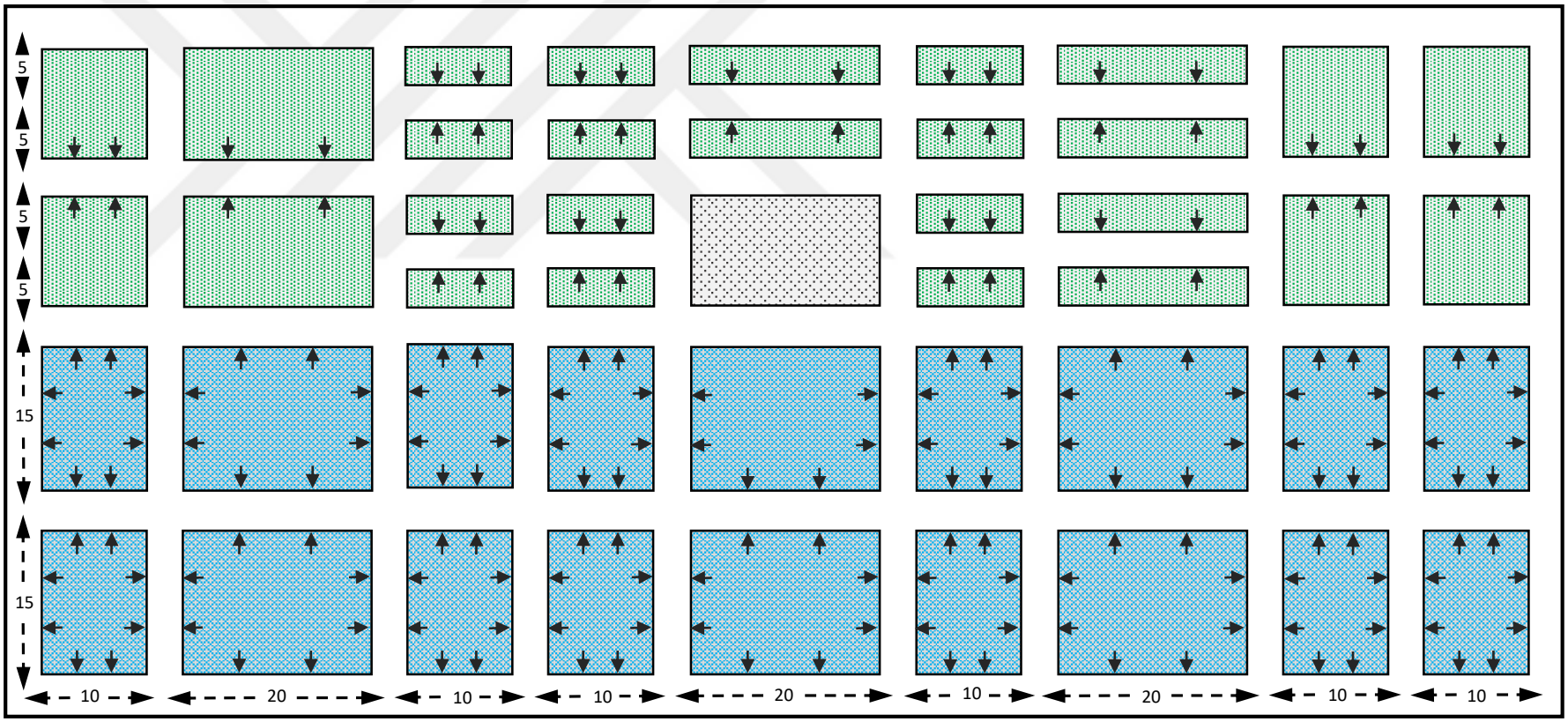
Şekil A.1. Küçük boyutlu kroki



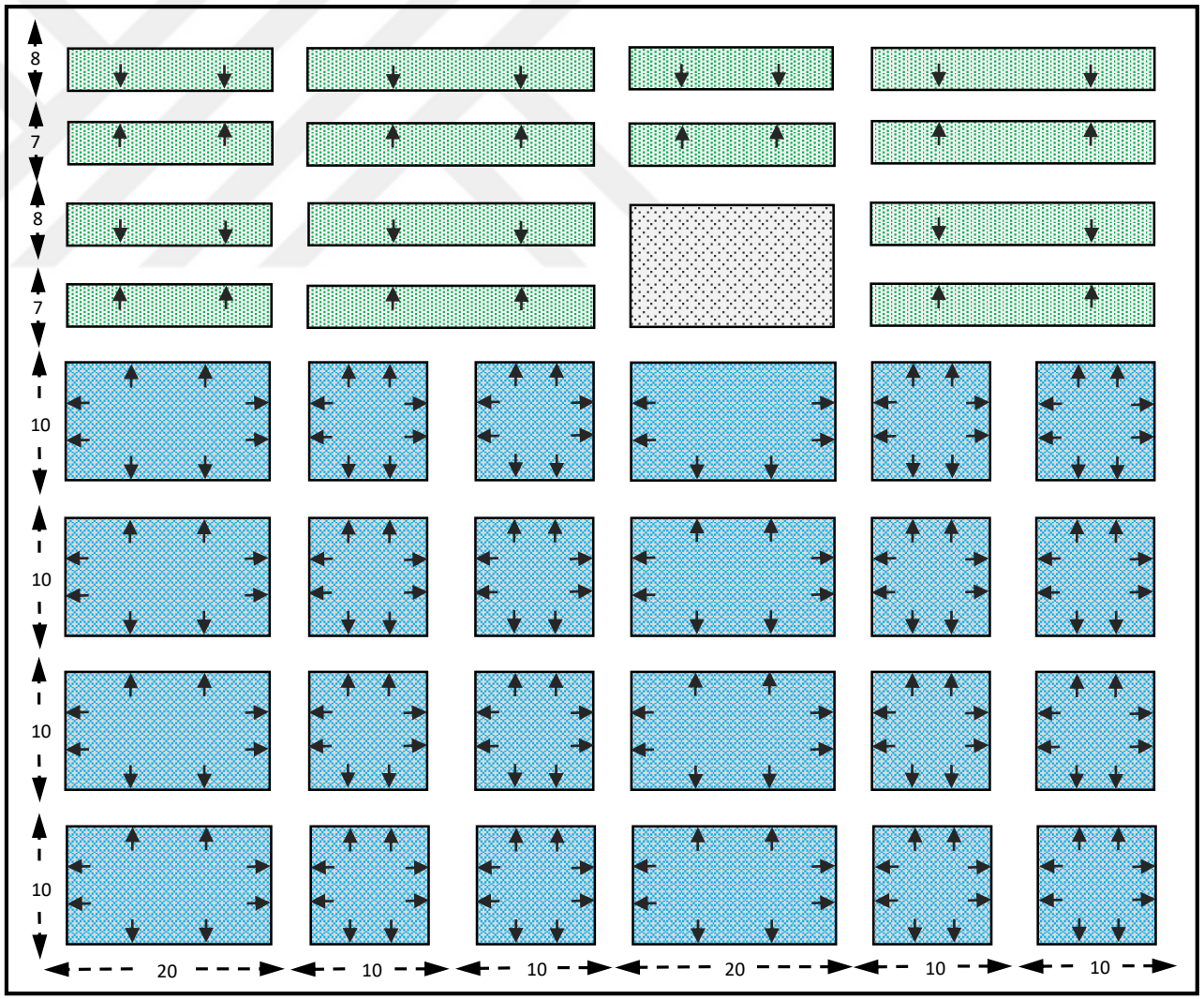
Şekil A.2. Orta boyutlu - dikdörtgen formlu kroki



Şekil A.3. Orta boyutlu - kare formlu kroki



Şekil A.4. Büyük boyutlu - dikdörtgen formlu kroki



Şekil A.5. Büyük boyutlu - kare formlu kroki

KİŞİSEL YAYIN VE ESERLER

Akman G., **Boyacı A. İ.**, Atay E. B., Sayar A. H. (2020). Yapay sinir ağları ile talep tahmini: otomobil satışları için YSA uygulaması. *International CEO Social sciences Congress*, Gorazhde, Bosna-Hersek, 18 - 20 Aralık 2020.

Akman G., Utku S., **Boyacı A. İ.** (2023). Evaluation of managerıal challenges for industry 4.0 application in lighting industry. *The 18 th International Conference of Constructive Design and Technological Optimization in Machine Building*, Bacau, Romanya, 11 - 13 Mayıs 2023.

Akman, G., **Boyacı, A. İ.**, & Kurnaz, S. (2022). Selectıng the suıtable e-commerce marketplace wıth neutrosophıc fuzzy ahp and edas methods from seller's perspectıve in the context of COVID-19. *International Journal of the Analytic Hierarchy Process*, *14*(3), 1-36.

Baynal K., **Boyacı A. İ.** (2018). Yalıtım malzemesi üretim sürecinde hata tespiti ve optimizasyonu. *8th International Conference of Strategic Research on Scientific Studies and Education,* Marmaris, Türkiye, 11 - 13 Mayıs 2018.

Boyacı, A. İ., Akman, G. (2018). Lojistik sektöründe kısıtlar teorisi düşünce süreçlerinin uygulanması: vaka çalışması. *Atlas Journal, 4(*10), 513-528.

Boyacı, A. İ., Baynal, K. (2019). Optimisation of elastomeric bearings' vulcanisation process using response surface methodology and desirability function approach. *Journal of Rubber Research, 22*(4), 187-193.

Boyacı, A. İ., Hatipoğlu, T., Balcı, E. (2017). Drilling process optimization by using fuzzy-based multi-response surface methodology. *Advances in Production Engineering & Management, 12*(2), 163-172.

Boyacı, A. İ., Hatipoğlu, T., Esen, H. (2014). Supplier Selection Using Fuzzy Linear Programming Technique for Multidimensional Analysis of Preference (LINMAP). *International Conference on Fuzzy Computation Theory and Applications*, Roma, İtalya, 22 - 24 Ekim 2014.

Boyaci, A. İ., Baynal, K., Özdin, Y. (2017). Sabun Üretim Prosesinin Cevap Yüzeyi Yöntemi ve İstenebilirlik Fonksiyonu Yaklaşımı ile Optimizasyonu. *5th International Symposium on Innovative Technologies in Engineering and Science* Baku, Azerbaijan, 29-30 Eylül 2017.

Çolak, M., **Boyacı, A. İ.** (2018). A Fuzzy Logic Based Green Performance Evaluation Model for Automotive Industry. *Avrupa Bilim ve Teknoloji Dergisi, 13,* 39-44.

Fığlalı A., **Boyacı A. İ**. (2019). Boru imalatı sürecinde sac kalınlıkları arası geçiş sıralamasının optimizasyonu için bir öneri. *International Marmara Sciences Congress (Autumn),* Kocaeli, Türkiye, 1 - 03 Kasım 2019.

Hatipoğlu T., **Boyacı A. İ**., Fığlalı A. (2021). İşgücü Çizelgeleme; Bir Havalimanı Giriş Kapısı için Senaryo Analizi Uygulaması. Özdemir, Y., (Ed.), *Mühendislik Alanında Akademik Çalışmalar* 37-50. İzmir: Duvar Yayınları.

Karabıçak, Ç., **Boyacı, A. İ.,** Akay, M. K., Özcan, B. (2016). Çok kriterli karar verme yöntemleri ve karayolu şantiye yeri seçimine ilişkin bir uygulama. *Kastamonu Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 13*(3), 106-121.

ÖZGEÇMİŞ

İlköğretim, ortaöğretim ve liseyi İstanbul'da tamamladı. 2007 yılında Kocaeli Üniversitesi Mühendislik Fakültesi, Endüstri Mühendisliği bölümüne başladı. 2012 yılında Endüstri Mühendisi olarak mezun oldu. 2013-2015 yılları arasında İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim dalında Yüksek Lisans öğrenimini tamamladı. 2015 yılında Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim dalında doktora eğitimine başladı. 2013 yılından beri Kocaeli Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği bölümünde Araştırma Görevlisi olarak görev yapmaktadır.