

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DİNAMİK BÜTÜNLEŞİK SÜREÇ PLANLAMA, ÇİZELGELEME
VE TESLİM TARİHİ BELİRLEME**

DOKTORA TEZİ

Caner ERDEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ
Tez Danışmanı : Dr. Öğr. Üyesi Halil İbrahim DEMİR

Haziran 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**DİNAMİK BÜTÜNLEŞİK SÜREÇ PLANLAMA, ÇİZELGELEME
VE TESLİM TARİHİ BELİRLEME**

DOKTORA TEZİ

Caner ERDEN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 25/06/2019 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.



Prof. Dr.
Ayhan DEMİRİZ
Jüri Başkanı



Doç. Dr.
Fatih TÜYSÜZ
Üye



Doç. Dr.
Özer UYGUN
Üye



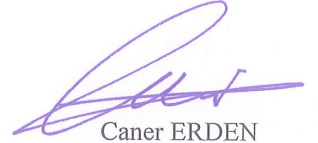
Doç. Dr.
Numan ÇELEBİ
Üye



Dr. Öğr. Üyesi
Halil İbrahim DEMİR
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.



Caner ERDEN

25.06.2019

TEŞEKKÜR

Tez çalışmam esnasında her zaman gece gündüz kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve destek olan Dr. Öğr. Üyesi Halil İbrahim DEMİR'e, öneri ve yorumlarıyla çalışmama destek olan hocalarım Doç. Dr. Özer UYGUN'a, Doç. Dr. Numan ÇELEBİ'ye, Endüstri Mühendisliği Bölümü ve Fen Bilimleri Enstitüsü'ndeki tüm çalışanlara ve özellikle çalışmalarım boyunca yardımlarını hiç esirgemeyen Dr. Abdullah Hulusi KÖKÇAM'a, süreç boyunca bana duygusal destek ve moral veren sevgili eşim Sumeyye ERDEN'e, kızıma, oğluma ve ayrıca hayatımın her evresinde bana destek ve örnek olan aileme teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ	vii
TABLolar LİSTESİ	ix
ÖZET	x
SUMMARY	xi
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
ELE ALINAN ÜRETİM FONKSİYONLARI VE BÜTÜNLEŞİK	
ÇALIŞMALAR.....	5
2.1. Süreç Planlama	5
2.2. Çizelgeleme	6
2.2.1. Dinamik çizelgeleme.....	10
2.3. Teslim Tarihi Belirleme	12
2.4. Bütünleşik Süreç Planlama ve Çizelgeleme (IPPS)	12
BÖLÜM 3.	
LİTERATÜR TARAMASI	15
3.1. Literatür Tarama Alanı ve Kapsamı	15
3.2. IPPS ile İlgili Literatür Taraması.....	17
3.2.1. Etmen sistem tabanlı yaklaşımlar.....	19
3.2.2. Algoritma tabanlı yaklaşımlar	20

3.3. SWDDA ile İlgili Literatür Taraması	21
3.4. Bütünleşik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme (IPPSDDA).....	23
3.5. Dinamik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme (DIPPSDDA).....	24
 BÖLÜM 4.	
DIPPSDDA PROBLEMİNİN MODELLENMESİ VE FORMÜLASYONU	27
4.1. Problem Tanımı	28
4.2. Problemin Modellenmesi	32
4.2.1. Sevk kuralları (dispatching rules)	34
4.2.2. Teslim tarihi belirleme kuralları	37
4.2.3. İşlerin geliş zamanlarının belirlenmesi	38
4.2.4. Rotaların belirlenmesi	39
4.2.5. Performans ölçütleri	40
4.2.6. Simülasyon çalışması	41
4.2.7. Örnek çalışma.....	42
 BÖLÜM 5.	
DIPPSDDA PROBLEMİNİN META-SEZGİSEL ALGORİTMALAR İLE ÇÖZÜMÜ	46
5.1. Genetik Algoritmalar (GA)	46
5.1.1. GA parametreleri	48
5.1.2. Kromozom yapısı	49
5.1.3. Seçilim ve tekrar üretim	50
5.1.4. Çaprazlama.....	51
5.1.5. Mutasyon.....	52
5.2. Tabu Algoritması (TA)	55
5.3. Tavlama Benzetimi (Simulating Annealing(SA))	56
5.4. Parçacık Sürü Optimizasyonu (PSO)	59
5.4.1. Modifiye edilmiş ICPSO (GA/ICPSO).....	64

5.4.2. Modifiye edilmiş kesikli PSO (Modified Discrete PSO - MDPSO)	65
5.5. Hibrit Algoritmalar (HA-GA/SA, GA/TA)	67
BÖLÜM 6.	
DENEYSEL SONUÇLAR VE KARŞILAŞTIRMALAR	69
6.1. Geliştirilen Bilgisayar Programı ve Kullanılan Kütüphaneler	69
6.2. Entegrasyon Seviyeleri	70
6.2.1. Birinci entegrasyon seviyesi (SIRO-RDM)	70
6.2.2. İkinci entegrasyon seviyesi (DSP-RDM)	71
6.2.3. Üçüncü entegrasyon seviyesi (SIRO-DD)	72
6.2.4. Dördüncü entegrasyon seviyesi (DSP-DD)	73
6.3. Atölyelere Ait Sonuçlar	75
6.3.1. Geliştirilen algoritmalara ait performans değerleri	81
6.3.2. En iyi çözümlere ait Gantt şemaları	83
BÖLÜM 7.	
SONUÇLAR VE TARTIŞMA	85
KAYNAKLAR	88
EKLER	104
ÖZGEÇMİŞ	133

SİMGELER VE KISALTMALAR LİSTESİ

$\bar{d}_j$	: Kesin Teslim Zamanı (Deadline)
$\sum w_j T_j$	: Toplam Ağırlıklı Gecikme
$\sum w_j U_j$	: Toplam Ağırlıklı Geciken İş Sayısı
$\sum w_j c_j$	: Toplam Ağırlıklı Tamamlanma Zamanı
μ_j	: j İşine Ait Dağılımın Ortalaması
s_j	: Hazır Zamanı (Release Date) = r_j
E_j	: Erken Tamamlanma Zamanı (Earliness)
L_j	: j İşinin Zamanında Bitirilmemesi (Lateness)
T_j	: j İşinin Gecikme Miktarı (Tardiness)
a_j	: j İşinin Sisteme Giriş Zamanı
c_{ij}	: j İşinin i Makinesinde Tamamlanması İçin Geçen Süre
c_j	: j İşinin Sistemde Bulunma Zamanı $C_j = f_j - a_j$
c_{max}	: Tamamlanma Zamanı (makespan)
d_j	: Müşteriye Taahhüt Edilen Teslim Zamanı (Due Date).
p_{ij}	: İşlem Zamanı: i. Makinedeki j işinin İşlem Görme Zamanı
r_j	: Serbest Kalma Zamanı
w_j	: j İşinin Önceliklendirilme Faktörü
σ_j	: j İşine Ait Dağılımın Varyansı
α	: Erken Bitirme Ceza Katsayısı, $\alpha > 0$
β	: Geç Bitirmenin Ceza Katsayısı, $\beta > 0$
m	: Makinelerin Sayısı
n	: İşlerin Sayısı
CDF	: Kümülatif Dağılım Fonksiyonu (Cumulative Distribution Function)
DD	: Teslim Tarihi Belirleme Kuralı (Due Date Assignment Rule)

DIPPSDDA	: Dinamik Bütünleşik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme
DSP	: Çizelgeleme Sevk Kuralı (Dispatching rule)
GA	: Genetik Algoritmalar
IDE	:Bütünleşik Geliştirici Ortamı (Integrated Development Environment)
IPPS	: Bütünleşik Süreç Planlama ve Çizelgeleme
IPPSDDA	: Bütünleşik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme
MDPSO	: Modifiye Edilmiş PSO
MICPSO	: Modifiye Edilmiş Kategorik PSO
NP-Zor	: Polinom Olmayan Zor Problem Sınıfı
PDF	: Olasılık Yoğunluk Fonksiyonu (Probability Density Function)
PSO	: Parçacık Sürü Optimizasyonu
SA	: Tavlama Benzetimi Algoritması
SWDDA	: Teslim Tarihi Atama ile Birlikte Çizelgeleme
TA	: Tabu Arama Algoritması
TZÜ	: Tam Zamanında Üretim

ŞEKİLLER LİSTESİ

Şekil 2.1. Bir Gantt şeması örneği.....	7
Şekil 2.2. Makine çizelgeleme problemi örneği.....	8
Şekil 2.3. Farklı rotalara sahip işler ve makineler gösterimi.....	10
Şekil 4.1. Örnek bir ağ gösterimi modellemesi.	34
Şekil 4.2. Alternatif rota sayısı için bir örnek gösterim.	39
Şekil 4.3. Simülasyon çalışmasının görselleştirilmesi	43
Şekil 4.4. Örnek çalışma sonucunda elde edilen Gantt şeması.	45
Şekil 5.1. Genetik algoritma için genel yalancı kod yapısı.	47
Şekil 5.2. Bireylerin(Kromozomların) gösterilişi.....	49
Şekil 5.3. Çaprazlama gösterimi.....	52
Şekil 5.4. Mutasyon gösterimi.....	53
Şekil 5.5. Genetik algoritma yalancı kodu.	54
Şekil 5.6. Dinamik olay çözümü için geliştirilen modelin akış şeması.....	54
Şekil 5.7. Tabu arama yalancı kodu.....	56
Şekil 5.8. Komşuluk arama algoritması akışı.....	57
Şekil 5.9. Tavlama benzetimi algoritması için yalancı kod.....	58
Şekil 5.10. Parçacık gösterimi	61
Şekil 5.11. MICPSO yalancı kodu	65
Şekil 5.12. MDPSO yalancı kodu.....	67
Şekil 6.1. İşlerin atölyelere geliş zamanları.....	76
Şekil 6.2. Birinci atölyedeki tüm algoritmalara ait iterasyon sonuçları.	77
Şekil 6.3. İkinci atölyedeki tüm algoritmalara ait iterasyon sonuçları.	77
Şekil 6.4. Üçüncü atölyedeki tüm algoritmalara ait iterasyon sonuçları.	77
Şekil 6.5. Dördüncü atölyedeki tüm algoritmalara ait iterasyon sonuçları.	77
Şekil 6.6. Beşinci atölyedeki tüm algoritmalara ait iterasyon sonuçları.	78
Şekil 6.7. Altıncı atölyedeki tüm algoritmalara ait iterasyon sonuçları.	78

Şekil 6.8. Yedinci atölyedeki tüm algoritmalara ait iterasyon sonuçları	78
Şekil 6.9. Sekizinci atölyedeki tüm algoritmalara ait iterasyon sonuçları	78
Şekil 6.10. Birinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.11. İkinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.12. Üçüncü atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.13. Dördüncü atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.14. Beşinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.15. Altıncı atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	82
Şekil 6.16. Yedinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	83
Şekil 6.17. Sekizinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları.....	83
Şekil 6.18. TA'dan gelen en iyi çözüme ait Gantt şeması ile gösterilen birinci atölye çizelgesi	84
Şekil 6.19. GA/TA'dan gelen en iyi çözüme ait Gantt şeması ile gösterilen ikinci atölye çizelgesi	84

TABLolar LİSTESİ

Tablo 3.1. IPPS konusunda yapılan çalışma sınıflandırması.....	18
Tablo 3.2. IPPS konusunda değişik veri tabanlarında şimdiye kadar yapılan çalışma sayıları.....	19
Tablo 4.1. Notasyonlar	28
Tablo 4.2. Atölyelere ait veriler.....	32
Tablo 4.3. Sevk kuralları ve açıklamaları.....	35
Tablo 4.4. Teslim tarihi belirleme kuralları	38
Tablo 4.5. Üstel dağılım fonksiyonları.....	39
Tablo 4.6. Mini atölye için üretilen işlerin operasyon süreleri ve rota bilgileri..	42
Tablo 4.7. Çizelgeleme sonuçları	44
Tablo 5.1. Birinci atölye için kullanılan GA parametreleri	48
Tablo 5.2. Jenerasyonlar arasındaki kromozom seçilme olasılığı farkları	51
Tablo 6.1. Birinci entegrasyon seviyesi için oluşturulan en iyi popülasyon	71
Tablo 6.2. İkinci entegrasyon seviyesi için oluşturulan en iyi popülasyon	72
Tablo 6.3. Üçüncü entegrasyon seviyesi için oluşturulan en iyi popülasyon.....	73
Tablo 6.4. Dördüncü entegrasyon seviyesi için oluşturulan en iyi popülasyon ..	74
Tablo 6.5. Sekizinci atölyeye ait entegrasyon sonuçları	75
Tablo 6.6. Atölyelere ait algoritma sonuçları	80

ÖZET

Anahtar Kelimeler: Süreç planlama, çizelgeleme, teslim tarihi belirleme, dinamik süreç planları, dinamik çizelgeleme

Süreç planlama, çizelgeleme ve teslim tarihi belirleme fonksiyonlarının bir arada çalıştırılması konusu literatürde güncel ve üzerinde çok sayıda çalışmalar olan bir konudur. Bu üç fonksiyonun entegrasyonu üzerine yapılan çalışmalar fabrikalarda daha etkili üretim süreçleri oluşturmaya yöneliktir. Fakat gerçek dünyadaki üretim süreçlerinde önceden tahmin edilemeyen değişiklikler, bozukluklar ya da aksaklıklar meydana gelebilmektedir. Bu aksaklıklar kimi zaman bir makine arızalanması, kimi zaman yeni bir siparişin acil olarak sürece dahil edilmesi gibi aksaklıklar olabilir. Fabrikalarda gerçekleştirilen ve optimum olduğu düşünülen üretim planlarının meydana gelebilecek aksaklıklar karşısında sürekli takip edilerek güncellenmesi gerekmektedir. Çünkü, tahmin edilemeyen süreç aksaklıkları optimum üretim planlarını bozabilir ve bunun sonucunda yapılan üretim planları ve çizelgeler daha az verimli hale gelebilir ya da çözümler olurlu bölgede olmayabilir. Bu nedenlerle, dinamik değişikliklere ve karmaşıklıklara karşı önlem alabilen ve uyum sağlayabilen üretim planları oluşturmak için yeni bir model geliştirilmesi önemli bir çalışma olacaktır. Dinamik olarak meydana gelen aksaklıkların sayısı ve yapısından dolayı kurulacak model karmaşık ve modelin çözümü zordur. Bu nedenle bu konuda literatürde fazla çalışma bulunmamaktadır. Bu tez çalışmasında dinamik olarak aksaklıkların dikkate alındığı süreç planlama, çizelgeleme ve teslim tarihi belirleme entegrasyon problemi çalışılmıştır. Modelin kurulmasında ve problemin çözümünde genetik algoritmalar, tavlama benzetimi algoritması, tabu arama algoritması ve parçacık sürü optimizasyonu algoritması ile hibrit algoritmaların sunduğu kombinasyonlar etkili bir şekilde kullanılmıştır.

DYNAMIC INTEGRATED PROCESS PLANNING, SCHEDULING AND DUE DATE ASSIGNMENT

SUMMARY

Keywords: Process planning, scheduling, due date assignment, dynamic process plans, dynamic scheduling

Integrating process planning, scheduling and due date assignment functions is a matter of current and many studies in the literature. The study on the integration of these three functions is aimed at creating more efficient production plans and schedules in manufacturing environments. However, in the real-world production processes unpredictable changes, disturbances and machine break downs may occur. These setbacks are sometimes a machine break down, sometimes including a new order as soon as the order is canceled may be glitch. The process plans that are best in the factories must be continuously monitored and updated in the face of difficulties that may occur. Because unpredictable process failures may disrupt best process plans and the resulting process plans, and schedules may be less efficient or may not be in the solutions area. For these reasons, developing a new model would be an important work to create process plans that can take precautions and adapt to dynamic changes and complexity. The model is complex, and the solution of the problem is difficult because of the number and structure of the difficulties that occur dynamically. Therefore, there is no study in this field in the literature. In this thesis study, the integration problem has been studied deciding the process planning, scheduling and due date assignment which has been taken into consideration dynamically. Genetic algorithms, simulated annealing algorithm, tabu search algorithm and particle swarm optimization algorithm with the combination of hybrid meta-heuristic algorithms will be used effectively in the solution of the problem.

BÖLÜM 1. GİRİŞ

Günümüz üretim şartlarında, kişiye özel, esnek ve küçük partiler ile üretim popüler hale gelmiştir ve bu tip üretimler gelecekte daha da fazla popüler hale gelecektir. Küçük partiler halinde üretim yaparken yalın üretimin bir gereği olan gerekli malzemeden gerektiği kadar üretim felsefesine de önem verilmesi ile üretimde ciddi oranlarda verimlilik artışları sağlanabilir. Tam zamanında üretim (TZÜ) felsefesine uygun üretim yapabilmek için üretim süreçlerinde her bir ürünün süreçlerinin izlenerek ve kontrol edilerek malzeme ihtiyaçlarının belirlenmesi ve müşteriye verilecek teslim zamanlarının ne geç ne de erken bir sürede olması gerekir. Bir girişimin başarılı olabilmesi üretim yapsın ya da yapmasın girişim sahiplerinin güvenilir olmasına sıkıca bağlıdır. Güvenirliğin ve müşteriye verilen teslim sürelerinde ürünlerin teslim edilmesinin müşteri üzerinde etkisi oldukça fazladır. Teslim zamanlarının verilen taahhütleri karşılamaşının müşteriye kazanma açısından önemi olduğu gibi şirketin kendisi için de prestij açısından önemi vardır. Teslim zamanlarının tutarlı olması ile şirket içerisinde daha kısa zamanlarda dağıtım imkanı sağlanmaktadır [1].

Bununla birlikte, esnek üretim yapan şirketlerin ürün çeşitliliğinin çok fazla olması nedeniyle atölyedeki üretim süreçlerinin planlanması ve atölye çizelgelenmesi seri üretime göre daha zor ve karmaşıktır. Çünkü üretimde farklı ürünlerin farklı yapıları, alternatif süreç rotaları ve işlem zamanları bulunmaktadır. Bununla birlikte, gerçek atölyelerde ürünler rastgele bir zamanda atölyeye gelirler ve çizelgelerin sürekli olarak gözden geçirilmesi ve yenilenmesi gerekmektedir. Örneğin bir makine boşa çıktığında çalışanlardan birinin hangi ürünün ileriki sırada işleneceğine karar vermesi gerekir. Makine önünde bekleyen operasyonların seçilmesi için hangi kuralın kullanılacağı kararı bir karar verme süreci sonucunda verilir. Üretimde hangi işin daha önce yapılacağına ilişkin kurallar vardır. Bu kurallara sevk kuralları denilir. Aynı şekilde,

işlerin teslim tarihlerinin belirlenmesinde kullanılan kurallara vardır. Bunlara teslim tarihi belirleme kuralları denilir. Üretim atölyelerindeki farklı durumlara göre farklı teslim tarihi ve sevk kuralları tercih edilebilir. Sevk ve teslim tarihi belirleme kurallarının birlikte çalıştırıldığı sistemler bütünleşik sistemler olarak adlandırılır. Bu kurallardan birisinin veya ikisinin birlikte kullanılması durumunda ortaya çıkacak iyileşmeler literatürde birçok çalışmanın konusu olmuştur.

Son zamanlarda bütünleşik süreç planlama ve çizelgeleme (Integrated Process Planning and Scheduling-IPPS) ve teslim tarihi belirleme ile çizelgeleme (Scheduling with Due-Date Assignment-SWDDA) konularında birçok çözüm yaklaşımı ortaya konmuştur. Özellikle IPPS ile alakalı literatürde oldukça fazla çalışma bulunmaktadır. Bütünleşik çalışmalar ile ilgili geniş literatür taramasına 2. Bölüm’de yer verilmiştir.

Genel olarak bütünleşik yaklaşımlarla süreç planlama ve çizelgelemede meydana gelebilecek sorunların önüne geçilmeye çalışılır. Bütünleşik fonksiyonlarla çalışarak daha verimli akış zamanları, işlem süreleri ve makine kapasitelerin ve üretim esnekliğinin çeşitlenmesine sahip olmak gibi bir atölye açısından hayati öneme sahip amaçlar hedeflenir. IPPS çözüm modelleri süreç planlama ve çizelgeleme fonksiyonlarının bütünleşik çalıştırılabileceğini ve çalıştırılması durumunda akış sürelerinin azalacağını göstermektedir. Ancak IPPS konusunda yapılan çalışmalar teslim tarihi belirleme fonksiyonundan bağımsız olarak yapılmış ve bu durumda literatürde yeni alan doğmuştur. Bu tez çalışmasında IPPS modellerine ek olarak teslim tarihi belirleme fonksiyonu da diğer iki fonksiyonla birlikte bütünleşik halde çalıştırılacaktır.

Çizelgeleme problemi literatürde deterministik olmayan polinom zor (NP-Zor) problem sınıfındadır. Problem küçük işletmeler için dahi çok büyük bir çözüm uzayına sahiptir. Çizelgeleme probleminin zor sınıfta yer alması ve bütünleşik olarak çözülecek süreç planlama ve teslim tarihi belirleme fonksiyonlarının farklı sayılabilecek parametrelere sahip olması nedenleriyle bütünleşik problemlerin kesin çözümüne ulaşmak mümkün değildir. Statik çizelgelemede hesaplamalı optimizasyon

teknikleri en iyi operasyon sıralamasını belirlemek için yeterli olabilirken dinamik çizelgelemede genellikle sezgisel yöntemler kullanılmak durumundadır [2]. Bu nedenle bu tez çalışmasında ele alınacak probleme en iyiye yakın çözümler bulabilecek çeşitli sezgisel yaklaşımlar denenmiştir. Ayrıca birden fazla entegrasyon seviyesi denenerek deneysel sonuçlar tezin sonuçlar bölümünde paylaşılmıştır. Bu sayede çözümü zor ve çözüm uzayı çok geniş olan bu tip problemler için en uygun çözüm yaklaşımını değerlendirme imkanı bulunmuştur. Çalışma sonucunda genetik algoritmalar, tavlama benzetimi, tabu arama, parçacık sürü optimizasyonu ve bu algoritmaların hibrit modelleri gibi çeşitli sezgisel yöntemler için performans kıyaslamaları ve üstünlük analizleri yapılmıştır.

Bu tez çalışmasında, teslim tarihi belirlenirken müşterinin önem katsayıları da dikkate alınmıştır. Ayrıca, her şirket için geçerli olan önemli müşterilere daha yakın teslim tarihleri verilerek şirketlerin daha iyi bir çizelgelemeye sahip olması sağlanmıştır. Şirketler müşteri önem sıralamasını kimi zaman işlem hacimlerine göre değerlendirirken kimi zaman da tamamen şirket içi dengeler gözetilecek şekilde sıralamayı gerçekleştirmektedir. Bu nedenle şirket için yapılacak bir önem sıralaması ile müşteriler arasında bir ayrım gözetilmiştir. Ayrım sonucunda şirket gözünde daha önemli olan müşterinin isteklerinin karşılanmasına daha fazla öncelik verilmiştir.

Literatüre bakıldığında süreç planlama, çizelgeleme ve teslim tarihi belirleme fonksiyonlarının bütünleşik olarak çalıştırılmasına pek değinilmediği görülecektir. Ancak Demir ve Taşkın [3] (2005), yaptığı doktora tezi çalışmasında ilk kez 3 fonksiyonun birlikte çalıştırılması ile ilgili bir tez çalışması yapmışlardır. Tez çalışmasında genetik algoritmaları kullanarak sonuca ulaşan Demir ve Taşkın, bu çalışmalar için “bütünleşik süreç planlama, çizelgeleme ve teslim tarihi belirleme (Integrated Process Planning, Scheduling and Due Date Assignment - IPPSDDA) kavramını kullanmışlardır.

Bu tez çalışmasında da 3 fonksiyonunun bütünleşik çalıştırılmasının fizibilitesine, avantajlarına, çözüm metotlarına ve sonuçlarına yeni yaklaşım açılarıyla tekrar

değinilmiş ve dinamik ortamda bu problemin çözümü meta sezgisel algoritmalar kullanılarak gerçekleştirilmiştir. Bu çalışmanın literatürdeki diğer çalışmalardan ayrıldığı nokta, çizelgelemeye girecek işlerin her birinin ayrı bir rotaya sahip olması ve geliş zamanlarının rastgele oluşudur. Ayrıca tezde literatürde yer alan önemli sevk kuralları ve teslim tarihi belirleme kuralları bir arada çalıştırılarak yine farklı bir yaklaşım ortaya koyulmuştur. İşler sisteme tek tek zaman içerisinde gelir ve değişik teslim zamanlarına sahip olur. Bir iş sisteme geldiğinde uygun bir teslim zamanının işe atanması gerekir. Ondan sonra diğer işlerin sıralanması sevk kurallarına göre ayarlanır. Sonuç olarak bu çalışmanın amacı meta sezgisel algoritmalar çözümleri oluşturarak süreç planlamanın ve çizelgelemenin daha verimli yapılmasını sağlamak ve bütünleşmiş problemin performansını artırmaktır. Yalın imalatın ve TZÜ bir gereği olarak bu çalışmada hem geç teslim edilen hem de erken teslim edilen işlere ve ayrıca işlere verilen teslim zamanlarının teslim tarihlerinin uzunluğuna ceza puanları verilerek erken ya da geç teslimlerin önüne geçilmeye çalışılmıştır.

Tezin ilerleyen bölümlerindeki ana hatlar şu şekildedir: Tezin ikinci bölümünde ele alınacak üretim fonksiyonlarının arka planı anlatılmıştır. Ayrıca bütünleşik çalışmaların özellikleri ve dinamik atölye çalışmalarının kapsamı bu bölümde açıklanmıştır. Üçüncü bölümde ise tezin konusunda yapılmış çalışmalar literatür taraması şeklinde verilmiştir. Dördüncü bölümde tezde ele alınan problemin özelliklerine, varsayımlarına, karmaşık yapısına değinilmiştir. Beşinci bölümde ise modellenen probleme ilişkin çözüm yöntemleri tartışılmış olup problem çözümü gerçekleştirilmiştir. Altıncı bölümde ise tezin çıktıları tartışılmış ve yapılan çalışmanın sonuç analizleri yedinci bölümde yapılmıştır.

BÖLÜM 2. ELE ALINAN ÜRETİM FONKSİYONLARI VE BÜTÜNLEŞİK ÇALIŞMALAR

Bu bölümde tezde bütünleşik hale getirilecek olan üretim fonksiyonlarının genel özellikleri ve yapıları ile literatürde bilinen bütünleşik çalışmaların unsurları ele alınmıştır.

2.1. Süreç Planlama

Üretim genel anlamda belirli teknik özelliklere sahip parçaları üretmek demektir, süreç planlama ise o teknik özellikleri karşılamak amacıyla yapılması gereken birtakım talimatları içeren sistem demektir. Süreç planlama modelleri işletmeler için stratejik öneme sahip modellerdir. Çünkü süreç planlama bir ürünün ekonomik ve tam olarak üretilmesi için yapılması gereken bir sistematik karar verme metodudur [4], [5], [6]. Süreç planlama fonksiyonunun esas amacı süreç planları oluşturma ve bu süreç planlarında ürünü üretmek için gerekli olan hammadde bilgileri, işlemler ve operasyonlarına yer vermektir. Süreç planlamadaki bir diğer amaç da üretimdeki karışıklıkların önüne geçmek, işlemlerin maliyetlerini azaltmak ve teslim zamanlarını minimize etmektir. Bu nedenle süreç planlama bir ürünün maliyetini etkileyen önemli bir fonksiyondur [7].

Süreç planlama görevini yapan bir kişi öncelikle ürün geometrisinden doğan kısıtlamaları düşünmek durumundadır. Ardından süreç planlamacı, gerekli özellikleri ürüne yansıtabilmek için uygun süreçlerin, makinelerin, malzeme ve ekipmanların ürün için hazır hale getirilmesi sağlamalıdır. Süreç planlama aşamasında, iş parçası seçimi, üretim süreci seçimi, süreç ekipmanı seçimi, makine ayarı seçimi, operasyon sıralaması, süreç parametreleri seçimi gibi görevler yerine getirilir.

Geleneksel üretim yapan atölyelerde bu görevler genellikle bir kişi tarafından bir not defteri kullanılarak yapılır. Ancak bu görev oldukça uzun zaman ve yüksek beceriler gerektirmekteydi. Yetenekli süreç planlamacılar dahi süreç planlarını uzun zaman aralıklarında tamamlayabilmekteydi. Bu nedenle otomatik oluşturulan süreç planlama çalışmaları gerçekleştirilmiştir. Bilgisayar destekli süreç planlamanın (Computer aided process planning)(CAPP) ortaya çıkması ile birlikte süreç planlama aşamasındaki planlar bilgisayar ortamında yapılmaya başlanmıştır [8].

2.2. Çizelgeleme

Genel anlamda işlerin ne zaman meydana geleceğinin planıdır. Çizelgelemenin sorusu “eğer her şey yolunda giderse bir sonraki aşamada neler yapılacaktır?” sorusudur. Bir öğrenci için yapılan ders çizelgelemesi ya da otobüs hareketleri için geliştirilen çizelgeler günlük hayatta herkesin incelediği çizelgelere basit örneklerdir. Çizelgelemeye günlük hayatımızın birçok yerinde karşılaştığımız gibi ürün ya da hizmet üreten şirketlerde de karşılaşabiliriz. Çizelgeleme bütün endüstrilerinde hayati öneme sahip bir karar verme sürecidir. Atölyelerde atölye kaynaklarının (makine, iş gücü, malzeme, ekipman) nasıl paylaştırılacağına kararı verilir. Şirketlerin pazarda daha iyi konumda yer alabilmesi için birçok yan etkenle birlikte çizelgelemelerini daha verimli halde gerçekleştirilmesi gerekmektedir. Ayrıca şirketler müşterilerine verdikleri teslim tarihi sürelerini ne geç ne de erken olmamak üzere planlama yapmak durumundadırlar. Yanlış bir planlama ya da çizelgeleme müşterilerin kaybına ya da ekonomik zararlara neden olabilir. Bu nedenle müşterilerini kaybetmek istemeyen şirketlerin üretim aktivitelerini ne zaman yapacaklarını ve ne zaman bitireceklerini daha önceden müşterilerine bildirmeleri beklenmektedir. Herhangi bir endüstrideki çizelgeleme yapısı yerine getirilmesi gereken görevlerden ve bu görevler için kullanılacak kaynaklardan oluşmaktadır. Çizelgelemede çıktı olarak amaç bazı kesin olmayan bilgiler ile birlikte verilen görevler (işler) ve kaynaklar (makineler) için zaman çizelgesi çıkarmaktır. Bu amacı yerine getirirken kaynakların etkin kullanılması da göz ardı edilmemesi gereken bir diğer amaçtır. Burada problem

genellikle bir karar verme süreci olarak ele alınır ve işlerin gerçekleşme zamanlarının gerçekte olacak zamanlardan ne erken ne de geç olarak gerçekleşmesi istenir.

Çizelgeleme sürecine başlamadan önce atölyedeki makinelerin ya da iş istasyonlarının hangi özellikte ve hangi kapasitede olduğunu bilmek gereklidir. Böylece hangi işlerin uygun bir şekilde hangi makinelerde işlenebileceği görülebilir. Bununla birlikte işlem görecekt işler açısından da birkaç bilginin daha çizelgelemeden önce bilinmesi gereklidir. Bunlar: kaynak kullanımı, işlem zamanı, en erken başlama ve bitiş zamanlarıdır.

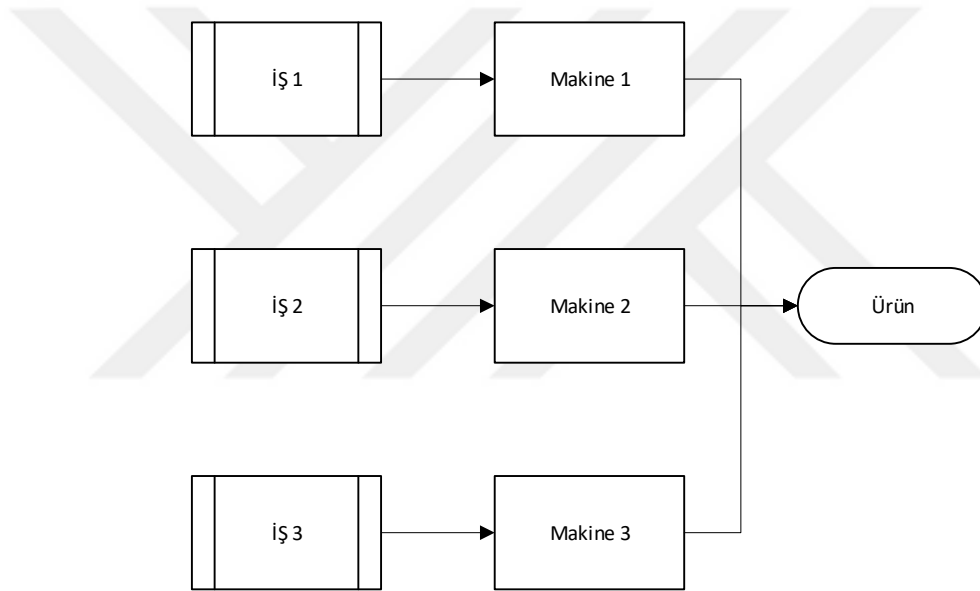
Çizelgeleme probleminin çözümü ilk olarak 1900'lü yılların başında Henry Gantt ve ark. [9] tarafından ortaya gerçekleştirilmiştir. Bu seneye rağmen çizelgelemenin literatüre geçmesi ve gerçek hayatta uygulamalarının ortaya çıkması birkaç yıl daha almıştır. Henry Gantt ve ark. tarafından geliştirilen Gantt şeması halen günümüzde en fazla kullanılan çizelgeleme modelidir. Gantt şemasının temelinde kaynaklara atanan işlerin zamanlarının ve bitiş zamanlarının gösterildiği bir format form vardır. Temel yapıdaki Gantt şeması Şekil 2.1.'de gösterilmiştir.



Şekil 2.1. Bir Gantt şeması örneği

Atölye çizelgelemede makine sayısına göre bir sınıflandırma yapmak mümkündür. Örneğin tek makineli çizelgelemede tek makine ve birden fazla iş bulunur. Bu çizelgeleme modeli daha çok sıralama problemi olarak tanımlanır. İşlerin başlama ve bitiş zamanlarına göre makinede işlem sıralaması oluşturulur. Öncelik kuralı olarak

teslim tarihi en yakın olan işin önceliklendirildiği kural “earliest due date” (EDD) kuralı olarak bilinir. Ayrıca ortalama bekleme süresini en aza indiren “Smallest process time (SPT)” öncelik kuralı da bu tip çizelgeleme için kullanılabilir. Tek makine probleminde aynı fonksiyona sahip başka bir makinenin de var olması durumunda paralel makine çizelgeleme problemi ortaya çıkar. Bu tip problemde işler boş olan makinede işlem görebilir. Makinelerin aynı fonksiyona sahip olması nedeniyle esnek bir çizelgeleme yapılması ve daha kısa çevrim zamanlarına ulaşılabilmesi mümkündür. Tek makine ve paralel makine çizelgeleme problemleri Şekil 2.2.’de gösterilmiştir.



Şekil 2.2. Makine çizelgeleme problemi örneği

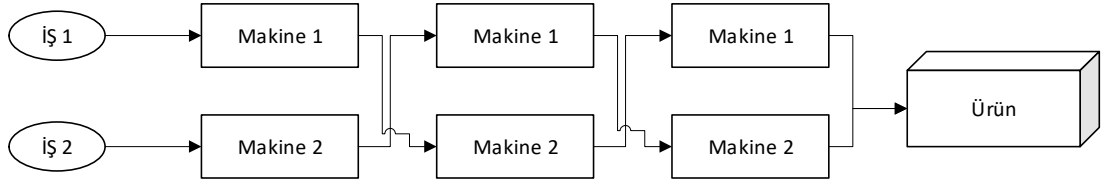
Tek makinenin ya da kaynağın bulunduğu ve çizelgelemenin özel bir hali olan bu modelin çözümü basittir. Buna rağmen tek makineli çizelgeleme probleminin çözümü önemlidir. Çünkü bu tip basit çözümlü çizelgeleme problemleri daha karmaşık çizelgeleme problemlerinin anlaşılması için bir ön ayak oluşturur. Tek makineli çizelgeleme problemi aşağıdaki şartlara/varsayımlara sahiptir [10].

- 0 zamanda işleme hazır n tane iş bulunmaktadır.
- Makineler tek seferde sadece tek bir işi yapabilmektedir.
- İşlerin hazırlık zamanları iş sıralamadan bağımsızdır ve işlem zamanlarına eklenmiş haldedir.
- İşler daha önceden bilinmektedir ve rastgele olmayan modellerdedir.
- Makineler sürekli olarak hazırdır, makine bozulmaları göz ardı edilmiştir.
- Makineler çalışma beklerken boşta duramazlar.
- Bir operasyon başladığında bir kesinti olmadan devam eder.

Tek makineli sıralama problemi yaklaşık 40 yıldır çözülmektedir. Bazı çözümler sayesinde problem oldukça verimli ve etkili sonuçlara ulaşmıştır. Fakat yine de tek makineli sıralama problemi genellikle NP-Zor sınıfında yer almaktadır. Bu modelin stokastik çözümü Baker tarafından 2014 yılında yapılmıştır. Baker işlem zamanlarını rastgele değişkenlerden oluşturmuş ve probleme etkili olacak bir çözüm üretmiştir [11].

Diğer bir çizelgeleme modeli akış tipi çizelgelemedir. Bu modelde birden fazla makinenin ve işin olduğu makine çizelgeleme problemi ele alınır. Bu problem tipinin önemli özelliği işlerin işlem göreceği makinelerin daha önceden belli olmasıdır. Yani örnek olarak 5 makineli akış tipi çizelgeleme örneği verilecek olursa 1. İş M1, M3, M5, M2, M4 makinelerinden sırasıyla geçerek işlem görüyorsa diğer işlerin de işlem görme sırası aynı kalacaktır. Akış tipi çizelgelemenin diğer bir özelliği ise çizelgeleme modelinin tekrar ziyaretlere genellikle izin vermemesidir. Atölye tipi çizelgelemede bu özellikten farklı olarak tekrar ziyaretlere izin verilmektedir. Ayrıca akış tipi çizelgelemede bütün işler bütün makineleri ziyaret etmek durumundadır.

Şekil 2.3.'te işlerin belirli bir rotada makineleri ziyaretleri gösterilmektedir.



Şekil 2.3. Farklı rotalara sahip işler ve makineler gösterimi

Bu tez çalışmasında ele alınacak olan çizelgeleme modeli atölye tipi çizelgeleme modelidir. Atölye çizelgeleme, m adet makineden işlem görmesi için n tane işin atanması problemidir. Çizelgeleme gerçekleştirilirken bir takım kapasite kısıtlarının karşılanması ve amaç fonksiyonunun en iyilenmesi hedeflenmektedir. Probleme ilişkin iki varsayımdan bahsedilebilir. Birinci varsayım, bir işin bir operasyonu bitmeden diğer operasyona başlanamaması ile ilgilidir. İkinci varsayım ise bir işin aynı anda sadece bir makinede işlem görmesidir. Atölye çizelgelemede mümkün çizelgelerin sayısı $(n!)^m$ 'dir. Mümkün çizelgelerin sayısı iş veya makine sayısının artması ile daha da artacaktır. McCormick ve Pinedo [12] (1995)'in ifadesiyle, eğer bir problemi makul süreler içerisinde çözebilecek bir algoritma yok ise bu tip problemler NP-Zor problemler olarak sınıflandırılır. Sadece atölye çizelgeleme problemi de NP-Zor problem sınıfına giren bir problemidir.

2.2.1. Dinamik çizelgeleme

Çizelgeleme dinamik ya da statik olabilir. Günümüzde birçok çizelgeleme modeli statik ortamda çalışmaktadır. Statik çizelgeleme atölyedeki koşulların çizelgelemenin sonuçlanacağı zamana kadar değişmediği varsayımı üzerine kurulur. Bu varsayım atölyede bazen olurlu olmayan çizelgelemelere neden olur. Gerçek bir atölye ortamında her zaman yeni bir olayla karşılaşılabilir. Bu olaylar çizelgelerde değişiklik yapılması ile sonuçlanmaktadır. Çizelgelerde değişiklik yapmayı zorunlu hale getiren olaylara cevap verebilen çizelgelerin oluşturulması için dinamik çizelgeleme konusu çalışılmaktadır. Dinamik çizelgelemede herhangi bir görev tam olarak mevcut olduğundan emin olmadıkça ilgili kaynağa atanmaz. Çizelgeleme sonucunda performans kriterine göre çizelgelerin verimliliği hesaplanır.

Çizelgeleme kuralları hangi görevin ya da kaynağın seçileceğine karar vermek üzere geliştirilmiştir. Bu kurallar sevk kuralları (dispatching rules) olarak bilinir. Sevk kuralları dinamik ya da statik yapıda olabilir. Dinamik yapıda olan kurallar zaman değiştikçe değişir.

Gerçek atölyelerde dinamiklik unsurları şu şekilde özetlenebilir:

- İşler $t=0$ anında çizelgelenmeye hazır değildir. İşlerin gelişleri herhangi bir t zamanında olabilir.
- İşlerin gelişleri arasındaki süreler değişkenlik gösterir. Gelişler arası süreler bir olasılık dağılımına uyabilir.
- Her işin öncelikleri aynı değildir, kimi işler müşterinin öneminden dolayı önceye ya da sonraya alınır. Bu nedenle işlere ağırlık atanarak işlerin çizelgelemeye sokulması gerekir.
- Çizelgelemenin herhangi bir zamanında makine arıza yapabilir. Makine bozulması (machine breakdown), bir makinenin sürekli kullanılabilir halde olmaması anlamına gelir. Bozulan makinenin onarımı için gerekli sürelerin belirlenmesi gerekebilir. Bu durumda makinenin arızalanması ve tamiri için geçen sürelerin çizelgelere dahil edilmesi gerekir.
- İşlerin işlem süreleri değişkenlik gösterebilir. Bu süreler ortalaması μ varyansı σ olan bir normal dağılıma uyabilir.
- Atölyeye bir an önce teslim edilmesi gereken acele işler gelebilir. Bu durumda atölyedeki mevcut çizelge değiştirilerek yeni çizelge hazırlanması gerekir.

Dinamik olarak meydana gelen aksaklıklar süreç planlarında ve çizelgelemede değişiklik yapma ihtiyacını doğurur. Yani yapılan optimuma yakın entegre planlar daha az verimli hale gelebilir. Ya da verilen planın olurlu olmama ihtimali de ortaya çıkabilir. Üretim planlarında yapılmak zorunda kalınan değişiklikler sonuç olarak tüm üretim verimliliğini etkileyebilecek boyuta gelebilir. Bu nedenlerle yapılan çalışmanın dinamik değişimleri de göz önünde bulundurarak dinamik atölyeler için güncellenmesi

bir gereklilik haline gelmiştir. Böylece, dinamik olarak düşünülen entegrasyon problemi üretim çevreleri açısından çok önemli bir çalışma olacaktır.

2.3. Teslim Tarihi Belirleme

Teslim tarihi belirleme probleminin esası, işlere teslim tarihi atanmasıdır. Teslim tarihi atanırken birçok durumda müşteriler ile görüşmeler yapılabilir [13],[14]. Uzak teslim tarihleri fiyat indirmeye, müşteri memnuniyetsizliğine hatta sonunda müşteri kaybına yol açabilir. Geleneksel üretim ortamında bir işin teslim tarihinden önce bitirilmesi beklenir. TZÜ ortamında işin tam teslim tarihinde bitirilmesi arzulanır. Çoğu durumlarda erken bitirmeler envanter maliyetinde bir artışa, geç bitirmeler ise müşteri memnuniyetsizliği, fiyat kırma ve en kötüsü müşteri kaybına yol açar. Ceza fonksiyonu için yaygın kanaat erken ve geç bitirmenin cezalandırılması gerektiği erken bitirmelerde daha çok doğrusal bir ceza fonksiyonu geç bitirmelerde ise karesel bir ceza fonksiyonu kullanılması gerektiği görüşü yaygındır.

2.4. Bütünleşik Süreç Planlama ve Çizelgeleme (IPPS)

IPPS problemlerindeki en büyük amaç proses planlama ve çizelgeleme aşamalarını birlikte ele alarak problemi ortak bir çözüme kavuşturmaktır. Süreç planlama ve çizelgeleme üretim atölyeleri için iki önemli üretim fonksiyonudur. Bu iki fonksiyon da zamanında üretim ve kaynakların maksimum kapasite kullanılması için çok büyük bir öneme sahiptir. Hem süreç planlaması hem de çizelgeleme fonksiyonları verimliliği ve kaynakların kullanımını doğrudan etkileyen bir noktadadır [5]. Rippin [15] tarafından tarif edildiği gibi süreç planlama ve çizelgeleme arasındaki farklar yapay olarak gösterilmesine rağmen, bu iki fonksiyon zamanlarının uzunluğu ve ilgili kararlar göz önüne alınarak birbirinden ayrılabilir. IPPS tanımı ilk olarak 1985 yılında Chrysolouris ve ark. [16] tarafından yapılmıştır.

Üretim yönetimi ve verimliliği için hayati öneme sahip bu 2 fonksiyonun birbirinden ayrı düşünülmesi beraberinde birçok sorunu getirmektedir. Süreç planları atölyedeki kaynakların sınırsız bir şekilde boş ve müsait olduğu varsayarlar. Bu yüzden süreç planları süreçler için en fazla önerilen kaynakları süreçlere atamaktadır. Bu da istenen kaynakları tekrar tekrar kullanmaları sorununa yol açar. Bunun sonucunda da gerçekçi olmayan süreç planları oluşturulur.

Birbirinden ayrı yapılan süreç planları ve çizelgelerde sabit süreç planları oluşturulur. Bu durumda çizelgelemeyi, her bir operasyonda sadece bir makine kullanmayla kısıtlar. Bu durumda çizelgelemenin muhtemel kullanacağı kaynaklar göz ardı edilmiş olur. Ayrıca süreç planları genellikle teknolojik ihtiyaçlar üzerine kurulurken çizelgeleme genellikle zamanlamayla ilgilenir. Bu bakımlardan süreç planlama ve çizelgeleme fonksiyonlarının bir arada çalıştırılması önemli birçok sorunu çözebilmektedir.

İş çizelgeleme veya çizelgeleme problemleri NP-Zor sınıfına ait problemlerdir. Dolayısıyla bütünlük problem de NP-Zor sınıfında olacaktır [17]. Süreç planlama ve çizelgelemenin bütünlük hale getirilmesi çok daha zor bir problemdir. Bu nedenle bu 2 fonksiyonun bir araya getirilmesinden önce bu 2 fonksiyonla ilgilenen üretim departmanlarının yeniden düzenlenmesi ve şekillendirilmesi gerekir. Çünkü bütünlük hale getirilecek problem birbirinden ayrı düzenlemeler ile çözülemez. Var olan süreç planları ve çizelgelerin yeniden ele alınması bu problemin çözümü için ilk adımı oluşturmaktadır. Kesin çözümlerin sadece küçük boyutlu problemler için geçerli olması nedeniyle problemin makul çözümü için sezgisel çözümler geliştirilmiştir.

Daha önce de bahsedildiği gibi süreç planlama ve çizelgelemenin bütünlük halde modellenmesine ilişkin literatürde birçok çalışma bulunmaktadır. Bu çalışmalar IPPS olarak anılmaktadır. IPPS ile ilgili çalışmaların derlendiği bir çalışma Phanden ve ark.

[18] tarafından 2011 yılında yapılmıştır. Phanden ve ark. bu derleme çalışması için 98 adet çalışmayı incelemiştir.

Literatüre baktığımızda ilk 2 fonksiyonun yani süreç planlama ve çizelgelemenin bütünleştirilmesinin ve son 2 fonksiyonun yani çizelgeleme ve teslim tarihi belirlemenin bütünleştirilmesinin çalışılmış konular olduğunu görebiliriz. Bu çalışmalara ilişkin bilgiler literatür özeti bölümünde verilmiş olup IPPS problemi ile ilgili yapılan çalışmalar derlenmiştir.

IPPS probleminin geliştirilmesiyle elde edilen kazanımları şu şekilde sıralayabiliriz [19]:

- Entegrasyon modelinin ana amacı süreç planlama ve çizelgeleme fonksiyonlarını bir arada çalışmasını sağlamaktır.
- Atölye koşullarını yansıtan ve üretim sisteminin dinamik yapısına yanıt veren proses planları oluşturmak
- Tam Zamanında Üretim (TZÜ) için bir zemin hazırlamak. Bu yaklaşım, süreç planlarının gerektiğinde hazırlanmasına olanak sağlar.
- Hem süreç planlaması hem de çizelgeleme fonksiyonlarının hedeflerini göz önünde bulundurarak, aynı anda ve en iyi planları elde etmek.
- Süreç planlarını hazırlama önceki yöntemin temel ilkesi, işlem planlama işlevi çizelgeleme işlevinin hiyerarşik olarak önündeki.
- Makine kullanımını geliştirerek ve işleme maliyetini ve zamanını azaltarak üretim sisteminin verimliliğini artırmak
- Birbirleriyle bağlantılı fonksiyonlar arasında entegrasyon geliştirmek, böylece üretimi tamamen entegre bir sisteme daha yakın hale getirmek.
- Bilgi daha iyi depolamak için tüm üretim fonksiyonlarının proses plan ve çizelgeleme bilgilerine daha kolay erişebilmesini sağlamak.

BÖLÜM 3. LİTERATÜR TARAMASI

Bu bölümde tez konusu alanında yapılmış bütünleşik üretim fonksiyonları ile ilgili yapılmış çalışmalar literatür özeti halinde verilmiştir. Problemlerin çözüm metotları ayrı ayrı incelenmiş olup çalışmalar problem çözümünde kullanılan yaklaşıma göre ayrılarak incelenmiştir.

3.1. Literatür Tarama Alanı ve Kapsamı

İşlerin teslim tarihi ile atölyede tamamlanma tarihlerinin olabildiğince yakın olması TZÜ felsefesinin temel ilkelerinden birisidir. Eğer şirketler verdikleri teslim tarihinde üretimlerini gerçekleştirebilirlerse daha gerçekçi planlar yapabilirler ve fabrikaları için optimum kapasite oranlarına sahip olabilirler. Bu sayede kaynaklar verimli bir şekilde kullanılmış olur. Bu amaçlara hizmet edebilmek için imalat fonksiyonlarının bütünleşik hale getirilmesi ve bütünleşik çalışmalarının avantajlarının ortaya konulması çalışmaları literatürde oldukça fazla sayıda araştırma ile incelenmiştir. Atölye tipi üretim yapan fabrikalarda müşteri memnuniyeti rekabetçi piyasa özelliklerinden dolayı günümüzde şirketlerin öne geçebilmeleri için oldukça belirleyici bir etkidir.

Bütün bu ihtiyaçlar neticesinde, şirketlerin bütünleşik bir şekilde süreç planlama, çizelgeleme ve teslim tarihi belirleme fonksiyonlarını gerçekleştirmek için matematiksel ya da sezgisel yöntemlerin adımlarını kullanabilir miyiz sorusu hasıl olmuştur. Özellikle Holland'ın [20] 1975 yılında yaptığı çalışmalar sonucunda ortaya çıkardığı genetik algoritmaların kullanımın yaygınlaşması ve kolaylaşması ile birlikte çizelgeleme gibi NP-Zor problemlerin çözümü üzerine araştırmacılar için yeni kapılar

açılmıştır. Bu bağlamda, 90'lı yıllarla birlikte süreç planlama, çizelgeleme ve teslim tarihi belirlemenin bütünleşik yapıları üzerine birçok akademik çalışma gerçekleştirilmiştir. 1990 yılında Nasr ve Elsayed [21] 2 algoritma geliştirerek akış süresini en aza indirmek için alternatif takım tezgâhlarını kullanmıştır. Üretim fonksiyonlarını bütünleşik hale getirme çalışmalarından birisini de Chen ve Khoshnevis geliştirmiştir. Chen ve Khoshnevis [22] süreç planlama ve çizelgeleme fonksiyonlarının bütünleştirmesi üzerine esnek süreç planlarıyla çalışmıştır. Morad ve Zalzal [23] ise genetik algoritmayı kullanarak üretimin toplam maliyetlerini minimize etmeyi hedeflemişlerdir.

2 fonksiyonun bir arada işlenebilmesi için ise Kim ve ark. [24] tarafından çalışılmıştır. Yapılan çalışmada araştırmacılar yapay zeka kullanarak arama teknikleri üzerine yeni bir yöntem geliştirdiler. Lim ve Zhang [25] ise dinamik süreç planlama ve dinamik üretim çizelgelemeyi bütünleşik hale getiren bir sistemi amaçladılar. Kumar ve Rajoiia [26] süreç planlama ve çizelgelemenin bütünleşik hale getirilmesinin bir çerçevesini ortaya koymuşlardır. Li ve ark. [27], bütünleşik süreç planlama ve çizelgeleme üzerine bir değerlendirme yayınlamışlardır.

Literatüre baktığımızda teslim tarihi belirlemenin çizelgeleme ile entegrasyonuna da pek çok çalışmada değinilmiş olduğunu görebiliriz. Diğer fonksiyonlarla aynı anda verilen teslim tarihleri amaç fonksiyonun da oldukça önemli ölçüde gelişmeye sebep olabilmektedir. Aynı zamanda teslim tarihi belirleme fonksiyonunun diğer fonksiyonlarla bütünleşik çalışmasının bugünün rekabetçi dünyasında artan bir ilgiye sahip olduğunu söyleyebiliriz. Gordon ve ark. [14] bir derleme çalışmasında çizelgeleme ve teslim tarihi belirlemenin performansından bahsetmiştir.

Yalnızca çizelgeleme problemi NP-Zor problem olduğundan entegre problem daha da zordur. Bu nedenle çok çok küçük problemler için kesin çözüm bulunabilmekte problem büyüdükçe kesin çözüm imkansız hale gelmektedir. Bu tezde bu zorluktan dolayı başarılı meta-sezgisel yöntemler kullanılmıştır. Ayrıca teslim tarihleri belirlenirken, müşterinin önemi literatürde hesaba pek katılmamıştır. Bu tez

çalışmasında önemli müşteriye erken teslim tarihi verme ve önemli işleri daha erken çizelgelemeyi hesaba katılmaktadır. Geleneksel olarak yalnızca gecikmeler cezalandırılırken, bu çalışmada hem gecikme hem erken tamamlanma hem de teslim tarihi süresi uygun sabit ve katsayılarla cezalandırılmaktadır. İmalat fonksiyonlarının bütünleşik hale getirilmesi ve bütünleşik çalışmalarının avantajlarının ortaya konulması çalışması doktora tezinde işlenecektir. Atölye tipi üretim yapan fabrikalarda müşteri memnuniyeti yukarıda anlattığımız özelliklerden dolayı günümüzde şirketlerin öne geçebilmeleri için oldukça belirleyici bir etkidir.

3.2. IPPS ile İlgili Literatür Taraması

IPPS problemi Guo [28] tarafından şöyle tanımlanmıştır:

“Verilen n adet işin alternatif üretim kaynaklarını kullanan operasyonlarla birlikte uygun makinelerin ve işlem sıralarının seçilerek makinelerde işlem görmesini öncelik sıralarını dikkate alarak çizelgelenmesini sağlamaktır. Bu çalışma yapılırken belirlenen optimizasyon kriteri de maksimize etmeye çalışılmaktadır.”

Bir diğer tanımda Chryssolouris [29] IPPS’yi bir karar verme yaklaşımı (Manufacturing Decision Making Approach-MADEMA) olarak tanımlamışlardır. IPPS probleminin çözümü için araştırmacılar ilk olarak alternatiflerin seçilmesi için bir karar matrisi oluşturdular. Ardından Sundara ve Fu [30] 1988 yılında çevrim zamanını minimuma indirmek için bir çizelgeleme yaklaşımı geliştirmişlerdir. Çalışmada araştırmacılar, grup teknolojisine dayanan bir bütünleşik bilgisayar destekli süreç planlama ve çizelgeleme otomasyon sistemi geliştirmişlerdir. Geliştirilen sistem Integrated Computer-Aided Proses Planning and Scheduling (ICAPPS) olarak literatüre geçmiştir [30]. Ayrıca Shrihari ve Geene [31] 1990 yılında CAPP olarak adlandırılan bir ilk örnek çözümü üretmiştir.

Süreç planlama ve çizelgeleme fonksiyonlarını bütünleşik çalıştırma fikri alternatif planlarla çalışmanın üretim çevrelerinde daha yüksek verimlilik elde ettiği sonucundan ortaya çıkmıştır. Wilhelm and Shin [32] 1985 yılında yaptıkları çalışmada alternatif operasyonların verimliliğinden bahsetmişlerdir. Bunun üzerine üretimdeki verimlilik artışını gören araştırmacılar bütünleşik process planları ve çizelgelemeler üzerinde çalışmalar yapmışlardır.

Süreç planlama ve çizelgelemeyi entegre edebilmek için en iyi yol iki fonksiyonu tek olarak ele almaktır. Süreç planlama ve çizelgeleme fonksiyonları tek başlarına NP-Zor problemlerdir ve bu iki problemin çözümü için gerçek zamanlarda optimal sonuçlar bulmak oldukça zordur [17]. Buna rağmen literatürde iki fonksiyonun entegrasyonu için bir çok çalışma bulunmaktadır. Bu çalışmaları etmen tabanlı ve algoritma tabanlı olarak ikiye ayırabiliriz. Tablo 3.1.'de yapılan sınıflandırma gösterilmiştir.

Tablo 3.1. IPPS konusunda yapılan çalışma sınıflandırması

	IPPS çalışmaları
Etmen Tabanlı Yaklaşımlar	[31]–[35]
Algoritma Tabanlı Yaklaşımlar	Evrimsel Algoritmalar
	[24], [39]–[41]
	Oyun Teorisi
	[42]–[44]
	Genetik Algoritmalar
	[23], [40], [45]–[59]
	Bal Arısı Optimizasyonu
	[60]–[62]
	Meta-Sezgisel Algoritmalar
	[63], [64]
	Parçacık Sürü Optimizasyonu
	[65]–[70]
	Tavlama Benzetimi
	[49], [71], [72]
	Hibrit Algoritmalar
	[42], [45], [60], [66], [73], [74]
	Karınca Kolonisi Algoritması
	[75]

IPPS literatürde oldukça fazla çalışılan bir konudur. Çünkü IPPS sayesinde üretimde verimlilik ve kapasite kullanım oranlarının artışı konularında önemli katkılar sağlanmaktadır. Birçok literatürdeki çalışmayla [70], [76]–[78] ve derleme makalesiyle [79]–[82] IPPS yaklaşımları incelenmiştir.

Şimdiye kadar değişik veri tabanlarında IPPS konusunda yapılan çalışma sayıları Tablo 3.2.'de verilmiştir.

Tablo 3.2. IPPS konusunda değişik veri tabanlarında şimdiye kadar yapılan çalışma sayıları.

Veri tabanı	Yıllar	IPPS
Scopus [83]	önce 2000	38
	2000-2010	132
	2010-	138
IEEE Explore [84]	önce 2000	32
	2000-2010	120
	2010-şimdi	122
ACM Digital Library [85]	önce 2000	6
	2000-2010	21
	2010-şimdi	47
ISI Web of Science[86]	önce 2000	16
	2000-2010	31
	2010-şimdi	44
ScienceDirect [87]	önce 2000	7
	2000-2010	11
	2010-şimdi	13
Toplam	önce 2000	99
	2000-2010	300
	2010-şimdi	364

3.2.1. Etmen sistem tabanlı yaklaşımlar

Etmen tabanlı yaklaşımlar IPPS için oldukça fazla sayıda kullanılmıştır. Gu ve ark. [35] bir parçaya ait rotaların ve çizelgelerin sözleşme tekliflerine göre bir araya geldiği bir çoklu etmen sistem tanımlamıştır. Lim ve Zhang [36] tarafından yapılan çalışmada IPPS için çoklu etmen sistem tabanlı bir çerçeve sunularak dinamik olarak imalat kaynaklarının bulunduğu bir platform gerçekleştirilmiştir. Yapılan çalışmada, Multi Agent Negotiation(MAN) adında bir düzeltme etmeni ve etmen sistem temelli (MAS) tabanlı bir Hibrit metot olan Hybrid-based Agent Negotiation (HAN) isminde etmenler geliştirilmiştir. Li ve ark. [81] tarafından yapılan çalışmada da bir etmen tabanlı yaklaşımla IPPS problemi çözülmüştür. Evrimsel algoritma temelli bir optimizasyon etmeni, etmenler arasındaki etkileşimi ve bilgi alışverişini yönetmek için kullanılmıştır. Leung ve ark. [75] yaptıkları çalışmada, Karınca Kolonisi Optimizasyonu(ACO) kullanılarak tamamlanma zamanını minimize etmek için IPPS

problemi çözülmeye çalışmışlardır. Zhang ve Wong [88] yaptıkları çalışmada dağıtık genetik algoritma (DGA) ve MAS kullanılarak IPPS problemine yeni bir yaklaşım eklemiştir. Bu yaklaşımın adına MAS-DGA denmiş ve yeni bir etmen tabanlı yapı alt popülasyonları değiştirmede, geleneksel GA'yı desteklemede kullanılmak üzere geliştirilmiştir.

3.2.2. Algoritma tabanlı yaklaşımlar

Algoritma tabanlı çözümler de literatürde oldukça fazladır. Kim ve ark. [24] tarafından yapılan çalışmada simbiyotik evrimsel algoritma ile IPPS problemi çözülmüştür. Li ve McMahon [71] tarafından yapılan çalışmada birleşik bir gösterim modeli ve tavlama benzetimi(SA) tabanlı bir gösterim modelini IPPS problemi için geliştirilmiştir. Zhao ve ark. [66] tarafından yapılan çalışmada bir hibrit parçacık sürü optimizasyonu (PSO) modeli ve bulanık mantık kullanılarak IPPS problemi çözülmüştür. Bu çalışmada alternatif makineleri rastgele seçmektense bir bulanık sistem çerçevesinde alternatif makinelerin seçilmesi gerektiği üzerinde durulmuştur. Li ve ark. [50] tarafından yapılan çalışmada aktif öğrenme genetik algoritmasını IPPS problemi çözümü için önerilmiştir. Li ve ark. [42] tarafından yapılan çalışmada oyun teorisi kullanarak GATS algoritmasıyla IPPS problemi çözülmeye çalışılmıştır. Lihong ve Shengping [51] tarafından yapılan çalışmada, geliştirilmiş genetik algoritma kullanılarak IPPS problemi çözülmüştür. Bu çalışmada süreç planları için yeni bir başlangıç belirleme yöntemi, yeni genetik gösterimler ve operatörler tanımlanmıştır. Li ve ark. [65] etkili bir güncellenmiş PSO algoritması ile IPPS problemini çözmeye çalışmıştır. Lian ve ark. [63] tarafından yapılan çalışmada yayılımcı bir algoritma kullanılarak IPPS problemi çözülmeye çalışılmıştır. Zhang ve ark. [89] tarafından yapılan çalışmada hibrit örnekleme stratejisi temelli bir çok amaçlı evrimsel algoritma geliştirilmiş ve geliştirilen algoritma HSS-MOEA olarak isimlendirilmiştir. Yu ve ark. [74] tarafından yapılan çalışmada IPPS problemi statik ve dinamik olarak iki evreye ayırmış ve GA/PSO temelli bir hibrit algoritma geliştirerek problemi çözülmeye çalışılmıştır. IPPS için son olarak Jin ve ark. [60] tarafından yapılan çalışmada hibrit arı kolonisi algoritması problem için kullanılmıştır.

3.3. SWDDA ile İlgili Literatür Taraması

Teslim tarihi atama problemi ile ilgili literatürde 3 çeşit teslim tarihi atama modeli bulunmaktadır.

- Ortak teslim tarihi atama: Her bir iş için önceden belirlenen ortak bir teslim tarihi atanması durumu.
- Süreç zamanlarını dikkate alarak yapılan teslim tarihi ataması: Slack due dates (SLK), total-work-content (TWK), Processing plus wait due dates (PPW) vb.
- Verilen değerlere göre işlere teslim zamanlarının atanması

Erken teslim ve geç teslim cezalandırmalarıyla tek makine çizelgeleme problemini ilk olarak Sidney [90] tarafından çalışılmıştır. Yapılan diğer bir çalışmada da tekrar aynı problem ele alınmış ve daha iyi sonuç üreten bir algoritma geliştirilmiştir [91]. Tek makine çizelgelemeleri için erken-geç tarih vermeler cezalandırıldığı diğer çalışmalar; [92]–[95].

Optimum teslim tarihinin belirlendiği çizelgeleme problemleri son 30 yılda önemli sayıda çalışmaya konu olmuştur. Teslim tarihi belirleme, şirketler için müşteri memnuniyeti açısından üzerinde durulması gereken bir konudur. Müşteriye verilen teslim tarihlerinin karşılanmaması durumunda şirketler çeşitli yaptırımlarla karşılaşabilirler. Teslim tarihi ile birlikte çizelgeleme konusunda yapılan çalışmaların bir kısmı tek makine çizelgeleme için yapılmışken bir kısmı da çok makine problemleri üzerinde yapılan çalışmalardan oluşmaktadır. Bu çalışmaların tamamında teslim edilecek işler için ortak teslim tarihleri belirlenmeye çalışılmıştır. Gordon ve ark. [14] tarafından yapılan çalışmada ortak teslim tarihinin ve çizelgeleme problemleri ile alakalı bir çerçeve çizilmiştir. Yapılan literatür çalışmasında tek makineli ve paralel makinelerin yer aldığı modeller incelenerek literatürdeki çalışmalar derlenmiştir.

Tek makineli problem için teslim tarihi ile birlikte yapılan çizelgeleme çalışmalarına literatürde sıklıkla karşılaşılabılır. Bunlardan bazıları; [96]–[104] şeklinde verilebilir.

Panwalkar ve ark. [105] yaptıkları çalışmada n tane iş ve tek makineden oluşan çizelgeleme problemi için ortak teslim tarihi belirleme problemini çalışmışlardır. Çalışma sonunda polinom temelli bir algoritma ile problem çözülmüştür. Chen [106] yaptığı çalışmada gecikmeyi ve erken teslim etmeyi en aza indirmek için her bir iş için ortak teslim tarihleri belirlemiştir. Bu çalışmada tek makineli çizelgeleme problemi incelenmiştir. Cheng ve ark. [107] yaptıkları çalışmada işlerin sıkışık süreç zamanları olduğu tek makine çizelgeleme problemi ve teslim tarihi atama problemini çözmeye çalışmışlardır. Cheng ve Kovalyov [108] yaptıkları çalışmada çizelgeleme, harmanlama (batching) ve teslim tarihi belirleme kararları birleştirilmişlerdir. Bu çalışmada işlerin n adet grupları olan tek makine çizelgeleme problemi çalışılmıştır. Bank ve Werner [109] yaptıkları çalışmada her bir iş için serbest kalma süresi ve işlem süresi belirlenmiş ve ortak teslim tarihi atanmıştır. Cheng [110] çalışmasında çizelgeleme problemi için optimal teslim tarihi bulmayı hedeflemiştir. Çalışma sonucunda sezgisel metodun problem için etkili çözümler bulduğu söylenmiştir.

Teslim tarihi belirleme ile birlikte çalıştırılan çizelgeleme problemi için literatürdeki pek çok çalışmanın tek makine çizelgeleme konusunda olduğu görülür. Birden fazla makinenin olduğu durumlarda, literatürde yine pek çok çalışmayla karşılaşılır. Sezgisel algoritmaların kolaylıkla kullanılmasıyla birlikte atölye tipi üretim gibi pek çok çok makine çizelgeleme problemi çözülebilmektedir. Çok makineli çizelgeleme problemleri için yayınlanmış bir literatür taraması bulunmaktadır. Lauff ve Werner [111] tarafından yapılan çalışmada belirli bir teslim tarihine sahip olan çok makineli çizelgeleme problemi çözümü için yayınlanan araştırmalara yer verilmiştir. Teslim tarihi ataması ile birlikte düşünülen çizelgeleme konusunda yapılan diğer çalışmalar; [14], [110], [112], [113].

3.4. Bütünleşik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme (IPPSDDA)

Geleneksel atölyelerde, süreç planları, çizelgelenmeler ve teslim tarihi atama fonksiyonları birbirinden ayrı ele alınmakta ve mevcut üretim kaynakları kullanılarak statik bir üretim planı oluşturmayı amaçlanmaktaydı. Bu da üretim verimliliğinde önemli sorunlara neden olmaktaydı. Modern üretim çevrelerinde ise, dinamik müşteri istekleri, daha fazla ürün çeşitliliği, yüksek global rekabet ve artırılmış otomasyon oldukça önemli bir noktadadır. Meta-sezgisel algoritmaların kullanımının yaygınlaşması, yeni teknoloji ve sistemlerin kolaylıkla kullanılabilir oluşu ile birlikte modern üretim çevrelerinde fonksiyonların bir arada ele alınıp çözülmesi üzerine çalışmalar yapılmıştır. Yapılan çalışmalar sonucunda, üretim fonksiyonlarının bir arada ele alınıp çözülmesinde üretim verimliliği anlamında ciddi kazanımlar olduğu görüldü. Bu nedenle araştırmacılar, süreç planlama ile çizelgelemeyi, çizelgeleme ile teslim tarihi belirlemeyi bir arada değerlendirmeyi düşündüler. Ancak üç fonksiyonun bir arada çalıştırılmasının çok daha iyi bir performansla sahip olacağı açıkken, bu konuda çok fazla çalışmanın olmadığını görmekteyiz. Süreç planlama, çizelgeleme ve teslim tarihi belirlemenin entegrasyonu üzerinde çalışılması gereken ve literatürde güncel bir konudur. Üretim açısından çok önemli olan bu üç fonksiyonunun bütünleşik çalıştırılması konusunda literatürde (indeksli dergilerde) çalışma bulunmamaktadır. Bütünleşik süreç planlama, çizelgeleme ve teslim tarihi belirleme (Integrated Process Planning, Scheduling and Due Date Assignment - IPPSDDA) sayesinde yine üretim maliyetlerinde azalma sağlanacağı ve daha sağlıklı üretim planlarının yapılabileceği söylenebilir. IPPSDDA konusu daha önce Demir ve ark. Tarafından çalışılmış ve problemin çözümü için çeşitli algoritmalar kullanılmıştır [114]–[117]. Bu çalışmalar ile IPPSDDA konusunun güncelliği sağlanmıştır. IPPS, SWDDA ve IPPSDDA konularında yapılan literatür çalışmalarına ve kısa özetlerine Bölüm 2.'de yer verilecektir.

Mevcut durumda IPPDSSA probleminin çözümüne ilişkin yeni yöntemler kullanılmaktadır. Bu tez çalışmasının asıl alanı olan dinamik ortamda çalıştırılacak

üretim fonksiyonlarının bir araya getirilmesi konusu daha önce çalışılmamış ve literatür için özgün bir konudur. Çizelgeleme problemi tek başına NP-Zor sınıfında yer alır. Hatta, atölye tipi üretim (job shop scheduling) problemleri sadece NP-zor sınıfında yer almaktadır. Ayrıca bu problemler çizelgeleme problemleri arasında en zoru olarak da gösterilir [118]. Dolayısıyla dinamik bütünleşik süreç planlama, çizelgeleme ve teslim tarihi belirleme (Dynamic Integrated Process Planning, Scheduling and Due Date Assignment–DIPPSDDA) problemi de çözümü zor olan ve çözüm uzayı çok geniş olan bir problemidir. Bu nedenle bu problemin çözümü için meta-sezgisel algoritmaların kullanılması zaruridir.

Belirtilen nedenlerle, bütünleşik halde çalıştırılan fonksiyonların daha verimli algoritmalarla çözülmesi günümüzde önem kazanmıştır. Gerçek atölyelerde meydana gelebilecek ani iç ve dış değişikliklere daha fazla karşılık verebilen üretimlerin uygulanması ve uygulamalarının iyileştirilmesi konuları araştırmacılar açısından güncel ve çalışmaya değer konulardır. Yapılacak olan çalışmalar dinamik ortamlarda çalışan atölyeler için uygulanabilir.

3.5. Dinamik Süreç Planlama, Çizelgeleme ve Teslim Tarihi Belirleme (DIPPSDDA)

Üç fonksiyon tek tek ele alındığında bile çözüm metotlarının çok karışık ve zor olduğu görülebilir. Sadece çizelgeleme problemi NP-Zor sınıfında yer almaktadır ve çözümü oldukça zaman almaktadır. Bu nedenle bu fonksiyonların entegrasyon problemlerini dinamik olarak incelemek literatürde çok nadir olarak ele alınmıştır. Birçok araştırmacı sadece çizelgeleme problemine odaklanarak dinamik atölye çizelgeleme problemini (job shop scheduling) (JSSP) ya da dinamik esnek atölye çizelgeleme(dynamic flexible job shop scheduling) (FJSP) çalışmışlardır. C. Zhang, Rao, and Li, 2008 yaptıkları çalışmada bir genetik algoritma kullanarak statik çizelgeleme problemini dinamik probleme çevirmiş ve geliştirilen genetik algoritma üzerine bir çizelgeleme stratejisi kurmuştur. Zandieh ve Adibi, 2010 yaptıkları çalışmada, JSP için değişken yakınlığı arama algoritması geliştirmiştir. Liping Zhang,

Gao, ve Li, 2013 [119] yaptıkları çalışmada hibrit genetik algoritma ve tabu arama temelli bir çizelgeleme stratejisini dinamik FJSP problemini çözmek için oluşturmuştur. Bu çalışmada çizelgeleme etkililiği ve çizelgeleme stabilizasyonu olmak üzere iki amaç belirlenmiş ve sistemin performansını iyileştirmek için birlikte çalıştırılmıştır.

Atölye çizelgeleme problemi ile ilgili güncel çalışma alanlarından birisi de dinamik çizelgeleme konusudur. Dinamik çizelgeleme problemlerinde işlerin rastgele zamanla atölyeye gelmesi, makinelerin arızalanması, acil işlerin olması ya da var olan işlerin iptal edilmesi gibi gerçek atölyelerde sıklıkla karşılaşılan durumlar ele alınır. Adibi ve ark. [120] yaptıkları çalışmada rastgele gelen işler ve makine arızaları gibi dinamik olaylara çözüm için bir komşu çözüm algoritması(VNS) kullanmışlardır. Dinamik atölye çizelgeleme problemi işlerin hazır zamanlarına göre deterministik ya da stokastik özellikler gösterebilir. Eğer işlerin hazır zamanları önceden biliniyorsa çizelgeleme deterministik, eğer zamanlar belirli bir dağılıma göre rastgele dağılıyorsa çizelgeleme stokastiktir denilebilir [121].

Dinamik çizelgeleme problemleriyle ilgili ilk literatür özetleri [122] ve [123] derleme çalışmalarında incelenebilir. Dominic ve ark. [124] yaptıkları çalışmada dinamik üretim atölyesinde çalışan 2 yeni sevk kuralı geliştirmişlerdir. Sevk kurallarının mevcut SPT, LIFO, LPT gibi kurallardan daha iyi sonuç verdiği ortaya konmuştur. Aydın ve Oztemel [125] yaptıkları çalışmada takviyeli öğrenme etmenlerini, uygun sevk kuralı seçimi için kullanarak dinamik atölye çizelgeleme problemi için çözüm bulmuşlardır. Li ve ark. [126] problem için genetik algoritma kullanarak bir yapay sinir ağı geliştirmişlerdir. Sha ve Liu [127] yaptıkları çalışmada veri madenciliği araçları işlerin dinamik koşullarına uyum sağlayacak bir veri madenciliği aracı geliştirmişlerdir. Zandieh ve Adibi [128] tarafından yapılan çalışmada en kısa ortalama işlem zamanı amaçlı çizelgeleme yöntemlerinin uygun parametrelerini tahmin etmek için YSA kullandılar. Zhang ve ark. [119] tarafından yapılan çalışmada ise dinamik esnek atölye çizelgeleme problemi için bir hibrit ve tabu arama algoritması geliştirilmiştir.

Dinamik bütünleşik proses planlama, çizelgeleme problemi ile ilgili çalışmalar ise sınırlı sayıdadır. Lin ve ark. [129] tarafından yapılan çalışmada atölyede meydana gelebilecek dinamik olaylardan kaçınmak için bir bütünleşik proses planlama ve çizelgeleme problemi ele alınmıştır. Xia ve ark. [130] tarafından yapılan çalışmada alternatif proses planlarının sağladığı yarar ile birlikte atölyede meydana gelen makine arızalanması ve işlerin rastgele atölyeye gelmesi ile ilgili problem komşu arama algoritması ile çözülmüştür. Bu konuda yapılan çalışmalar genellikle etmen tabanlı çözümlere dayanmaktadır. Wong ve ark. [37] tarafından yapılan çalışmada bir hibrit çoklu etmen tabanlı sistem önererek DIPPS problemini çözmüşlerdir.



BÖLÜM 4. DIPPSDDA PROBLEMİNİN MODELLENMESİ VE FORMÜLASYONU

Literatüre bakıldığında atölye çizelgeleme probleminin çoğunlukla statik olarak ele alındığı ve modellendiği görülebilir. Bu çalışmalarda atölyeye yeni iş gelmesi, makinelerin arızalanması vb. gibi dinamik unsurlar göz ardı edilmiştir. Çünkü atölye çizelgeleme problemi, tek başına NP-Zor sınıfında yer almakta ve çok geniş bir çözüm uzayına sahiptir. Sadece 6 makineli çizelgeleme problemlerinin çözümü dahi oldukça uzun hesaplamalar ile uğraşmamızı gerekmektedir [131]. Atölye tipi çizelgeleme probleminde her bir işin ayrı rotaları bulunur ve operasyonlar atölyedeki makinelerde işlem görecektir şekilde bir çizelgeleme yapılır. Çizelgeleme problemi iş karakteristiklerine göre şu şekilde 3 kategoriye ayrılabilirler.

- Deterministik Çizelgeleme: problemin işlerin geliş zamanları, teslim zamanları, rotaları ve makine durumları vb. gibi tüm elemanları deterministiktir ve deterministik olarak belirlenmiştir.
- Statik Çizelgeleme: Atölyeye gelen işlerin gelişler arası süreleri göz ardı edilmiştir. Tüm işlerin $t = 0$ anında hazır olduğu kabul edilir.
- Dinamik Çizelgeleme: Atölyeye iş gelişleri ve gelen iş sayıları zaman içerisinde değişkenlik göstermektedir.

Çizelgeleme probleminin çözümünde kaynakların verimli kullanımı, siparişlere zamanında ve hızlı yanıt verebilme, teslim zamanına göre erken ya da geç tamamlanmayı minimize etmek gibi amaçlar bulunmaktadır [10]. Bu amaçlara hizmet edebilmek adına, işlerin dinamik ortamda modellenebilmesi ve çizelgeleme probleminin dinamik ortamda çözülebilmesi gereklidir. Dinamik ortamda çalışan çizelgeleme modelleri gerçek hayattaki atölyelere göre daha uygun yapıdadır. Çünkü atölyelerde hazırlanan çizelgeler yeni işler geldiğinde yeniden gözden geçirilir ve

çizelgeleme işlemi tekrar yapılır. Bu nedenle çizelgelemenin dinamik yapıda olması gerekmektedir. Dinamik çizelgeleme problemi ile ilgili literatür çalışması ikinci bölümdeki çalışmaları vermiştir.

4.1. Problem Tanımı

Çizelgeleme problemlerinde sınırlı sayıda iş ve sınırlı sayıda makine olduğu varsayımı yapılır. İşlerin sayısı n ile gösterilir makinelerin sayısı ise m ile gösterilir. Genellikle, alt indis j işi gösterir. Benzer olarak alt indis i ise makineyi gösterir [132]. Problemde kullanılacak olan notasyon Tablo 4.1.'de verilmiştir.

Tablo 4.1. Notasyonlar

n	Atölyedeki iş sayısı
m	Atölyedeki makine sayısı
r	Atölyedeki her bir iş için rota sayısı
o	İşlere ait operasyon sayısı
d_j	İşin teslim zamanı
w_j	j . işin(müşterinin) ağırlığı
p_j	j . işin işlem süresini
p_{ij}	j . işin i . makinedeki işlem süresi
p_{av}	Ortalama işlem süresi
c_j	İşin tamamlanma zamanı
L_j	Gecikme süresi
T_j	Geç tamamlanma süresi
E_j	Erken tamamlanma süresi
α	Erken tamamlanma ceza katsayısı, $\alpha > 0$,
β	Geç tamamlanma ceza katsayısı, $\beta > 0$,
P_D	Teslim tarihi için hesaplanan ceza
P_E	Erken tamamlanma için hesaplanan ceza
P_T	Gecikme için hesaplanan ceza
P_j	j . iş için hesaplanan ceza
P_{total}	Toplam ceza
f_{min}	Amaç fonksiyonu
DD_{size}	Teslim tarihi kuralları sayısı
DR_{size}	Sevk kuralları sayısı
a_j	j . işin sisteme geliş zamanı

Çizelgeleme problemleri temel olarak 3 etmenden oluşmaktadır. Bu etmenler $\alpha < \beta < \gamma$ olarak gösterilirse α makineleri sembolize eder ve tek bir girdiye sahiptir. β ise süreç karakteristikleri ve kısıtlarla ilgili detayları tanımlar ve girdi içermeyebilir ya da tek girdili olabilir ya da birden fazla girdisi olabilir. γ ise problemin minimize edilecek olan amaç fonksiyonunu tanımlar ve genellikle tek girdi içerir [132]. Statik bir atölye çizelgeleme probleminde verilen performans kriterini optimize edecek şekilde n adet iş m adet makineye atanır [112]. Makine çevreleri genel olarak tek makineden başlar ve m adet makineye kadar çıkabilir. İşlerin rotalarına göre 3 farklı atölye tipi vardır. Bunlar;

- Açık atölye (open shop)
- Akış tipi atölye (flow shop)
- İş tipi atölye (job shop)

Gerçek bir üretim atölyesinde, bazı dinamik olaylar meydana gelebilir. Bu olaylar, makine arızalanması, yeni iş gelmesi ya da mevcut işlerden iptal edilen işlerin olması gibi olaylar olabilir. Dinamik olaylar atölyede çizelgelerin bozulmasına ve dolayısıyla çizelgelemenin yenilenmesine(rescheduling) neden olur. Bir dinamik olay meydana geldiğinde süreç planlarının ve çizelgelemenin meydana gelecek olaya yanıt verebilecek şekilde yeniden hazırlanması gerekmektedir.

Çizelgeleme problemlerindeki amaçlar genellikle işlerin tamamlanma zamanını minimize etmektir. Kimi zaman da amaç fonksiyonu ortalama akış süresini minimize etmek, makine yüklerinin dengelenmesi vb. şeklinde de olabilir. Bazı durumlarda amaç fonksiyonu teslim zamanlarının fonksiyonudur. Bu çalışmada amaç fonksiyonu olarak işlerin gecikmesi, erken tamamlanması ve verilen teslim zamanının optimize edilmesi olarak belirlenmiştir. İşin gecikmesi Denklem 4.1.'de gösterildiği gibi hesaplanır.

Gecikme ve erken tamamlanma sırasıyla aşağıdaki denklemdeki gibi hesaplanır:

$$T_j = \max(c_j - d_j, 0) \quad (4.1)$$

$$E_j = \max(d_j - c_j, 0) \quad (4.2)$$

Eğer iş verilen teslim tarihinden sonra tamamlanırsa, gecikme söz konusudur. Gecikme durumunda, beklendiği gibi erken tamamlama cezası 0'dır. Eğer iş, verilen son tarih tarihinden önce tamamlanırsa, erken tamamlanma söz konusudur ve gecikme cezası 0 olarak verilir. Erken ve geç tamamlanma tarihlerinin ceza değerleri sırasıyla Denklem 4.3, 4.4, 4.5'te gösterildiği gibi hesaplanır. Burada, denklem 6 Toplam ceza toplamını gösterir.

$$P_D = w_j \times \left(8 \times \left(\frac{D}{480} \right) \right) \quad (4.3)$$

$$P_E = w_j \times \left(5 + 4 \times \left(\frac{E}{480} \right) \right) \quad (4.4)$$

$$P_T = w_j \times \left(6 + 6 \times \left(\frac{T}{480} \right) \right) \quad (4.5)$$

$$P_{\text{total}} = P_D + P_E + P_T \quad (4.6)$$

Modelin amaç fonksiyonu erken ve geç tamamlanma ve verilen teslim tarihlerinin minimizasyonu olarak tanımlanır.

$$f_{\min} = \sum_{j=1}^n P_{\text{total}} \quad (4.7)$$

Burada gecikmenin pozitif çıkması durumunda j . iş geç tamamlanmış, negatif çıkması durumunda j . işi erken tamamlanmış demektir. Gecikme negatif olamayacağı için işin zamanında bitirilmemesi zamanının teslim zamanından farkını negatif olması engellenmeye çalışılmıştır. Denklem 4.4.'te erken tamamlanma hesabı gösterilmiştir. İşin erken bitirilmesi durumunda işin teslim zamanı ile tamamlanma zamanı arasındaki fark negatif olmayacaktır.

Denklem 4.7.'de gösterildiği gibi amaç fonksiyonu olarak erken ve geç tamamlanma sürelerinin toplamı alınmış ve daha iyi bir amaç fonksiyonu için erken tamamlama ve geç tamamlama için ağırlık puanı uygulanmıştır.

α erken tamamlanan iş için ağırlık puanını, β ise geç tamamlanan iş için ağırlık puanını göstermektedir. Denklem 4.4.'te cezalandırma değerlerinin hesaplanma formüller gösterilmiştir.

Bu çalışmada yapılan varsayımlar şu şekildedir;

- n adet iş ve işlerin birden fazla operasyonları bulunmaktadır.
- İşlerin geliş zamanları rastgeledir ve üstel dağılımı $\frac{1}{\mu}$ olan ortalamaya sahiptir.
- Her bir operasyonun işlem zamanı deterministik değildir ve normal olarak dağılmıştır.
- Her bir iş farklı ve kendi süreç rotasına sahiptir.
- Bir operasyon kendinden önce yapılması gereken bir operasyon varsa o operasyon yapılmadan yapılamaz.
- İşlerin her biri ayrı bir makinede yapılabilir, işler farklı makinelerde işlem görebilir.
- Yeni bir ürün sisteme geldiğinde uygun bir teslim zamanı işe atanır. Eğer bazı işler aynı anda aynı makine tarafından işlem görecektse, sevk kurallarına göre işlem görecektir.
- Her bir operasyon sadece bir makinede işlem görebilir, aynı şekilde bir makine sadece aynı anda sadece bir işlem yapabilir. Makinelerin arıza yapmayacakları varsayılmıştır.

4.2. Problemin Modellenmesi

DIPPSDDA problemi, atölye çizelgeleme problemine (JSSP) bir katkı olarak düşünülebilir. Genel olarak, JSSP'de, n adet işi belirtilen amaç fonksiyonunun optimizasyonu ile m adet makineye atanması problemi ele alınır. Bu modelde, her makinenin bir seferde sadece bir işi yapabileceği, bir seferde bir makinede bir işlem yapılabileceği, makinede işlem başladığında makinenin bozulmayacağı, acil bir işin sisteme gelmediği gibi bazı varsayımlar yapılır. Aksine, dinamik JSSP modelinde makineler bozulabilir veya acil işler atölyeye gelebilir. DIPPSDDA modeli dinamik bir JSSP probleminin bir çeşidi olduğundan, acil bir işin atölyeye gelebileceği varsayılmaktadır. Ayrıca, daha etkili çizelgeler oluşturmak için alternatif süreç planları da probleme dahil edilmiştir.

Bu tez çalışmasında 8 adet atölye tanımlanmış ve atölyelere ilişkin veriler Tablo 4.2.'de verilmiştir. Tabloya göre birinci atölyede 25 adet iş, 10 adet operasyon, 5 adet makine ve 5 adet rota bulunmaktadır. Birinci atölye için yapılacak olan iterasyon sayısı da genetik algoritma için 150 olarak belirlenmiştir. Probleme ilişkin girdi verileri Ek-1'de verilmiştir.

Tablo 4.2. Atölyelere ait veriler.

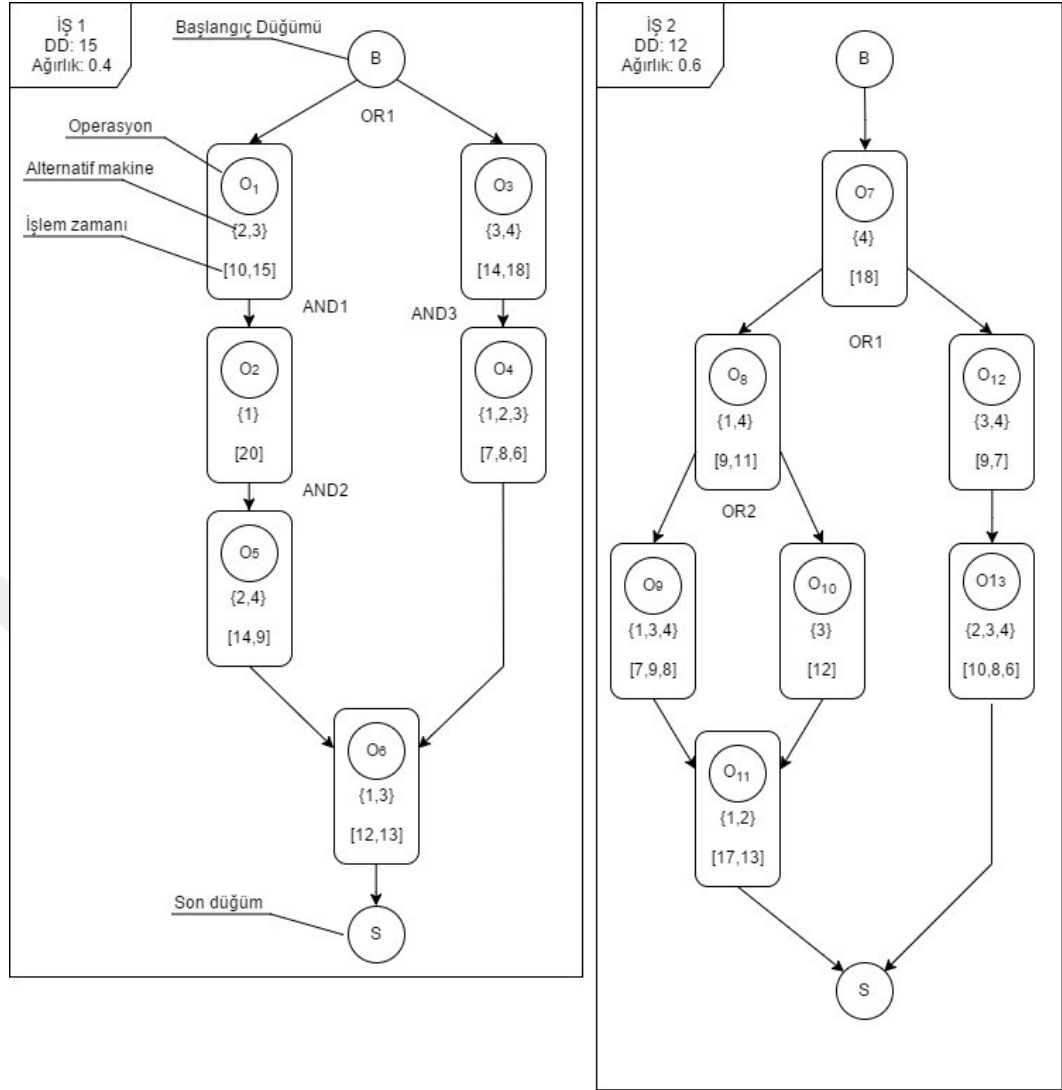
Atölyeler	1	2	3	4	5	6	7	8
İş sayısı	25	50	75	100	125	150	175	200
Operasyon sayısı	10	10	10	10	10	10	10	10
Makine sayısı	5	10	15	20	25	30	35	40
Rota sayısı	5	5	5	5	3	3	3	3
İterasyon Sayıları	150	150	100	100	75	75	50	50

Bütünleşik çalışacak DIPPSDDA sisteminin çözümünden 2 adet çıktı alınır. Birincisi her bir iş için ayrı ayrı üretilen süreç planları ve ikincisi de hangi işin hangi makinelerde ne zaman ve ne kadar işlem göreceğini gösteren amaç fonksiyonuna uygun çizelgelerdir. Dolayısıyla problemde bir adet seçme ve bir adet atama problemi bulunmaktadır. Bu iki alt problem için de çözüm bulunması durumunda uygun bir DIPPSDDA sistemi oluşturulmuş olur.

Proses planlamada 3 çeşit esneklikten söz edilebilir. Bunlar operasyon esnekliği, sıralama esnekliği ve süreç esneklikleridir. Operasyon esnekliğinde bir operasyonun alternatif makinelerde yapılabilmesinden bahsedilir. Sıralama esnekliğinde ise gerekli durumlarda operasyonların sıralarının değiştirilebilme olasılığıdır. Süreç esnekliği de benzer üretim özelliklerinin alternatif operasyonlarla ya da başka operasyon sıralamaları ile yapabilme olasılığıdır [133]–[135]. Bahsedilen esnekliklerin gösterimi için literatürde 3 adet grafik gösterimi bulunmaktadır. Bunlar; Petri-net [136], VE/VEYA (AND/OR) gösterimi ve ağ gösterimi modelleridir. Probleme uygun olması ve gösteriminin basit olması nedeniyle ağ gösterimi bu çalışma için seçilmiştir. Ağ gösterimine bu çalışma için teslim tarihleri ve işlerin ağırlıkları ilave edilmiştir. Ağ gösterimi atölyenin girdilerini göstermek için kullanılmıştır. Örnek bir ağ gösterimi modellemesi Şekil 4.1.'de verilmiştir. Çizelgelerin gösterimi için ise Gantt şeması kullanılmıştır. Gantt şemasında x ekseninde zaman gösterilirken y ekseninde sistemdeki makineler gösterilir. Makinelerin hangi zamanda hangi işi yapmakla meşgul olduğu Gantt şemasından okunabilir. Bu nedenle Gantt şeması kolay okunabilir ve fonksiyonel bir yapıdadır.

Ağ gösteriminin özellikleri aşağıda sıralanmıştır:

- Düğüm ve kenarlardan oluşur.
- Düğümler bir operasyon hakkındaki bilgileri tutar. Kenarlar ise düğümler arasındaki ilişkiyi saklar.
- Düğümün içerisinde operasyon numarası, operasyonun işlem görebileceği alternatif makineler ve o makinelerdeki işlem süreleri bulunur.
- Her bir çerçeve bir iş için hazırlanır. İşlerin teslim tarihleri de çerçevenin içerisine yazılır.
- Böylece işler için alternatif süreç planlar gösterilmiş olur.



Şekil 4.1. Örnek bir ağ gösterimi modellemesi.

4.2.1. Sevk kuralları (dispatching rules)

Makine boşa çıktığında sıradaki işin seçimi için bekleyen tüm işlere öncelik verilmesi ve makinede işlenecek işin belirlenmesi gerekmektedir. Yapılacak atama için tüm bekleyen işlerin 3 farklı özelliğine bakılabilir. Bunlar, işin özellikleri, makinenin özellikleri ve zaman özellikleri olabilir. Bu özelliklerden işin özelliklerine göre, işin ağırlıklarına göre, işlem zamanlarına göre ya da teslim zamanlarına göre atamalar yapılabilir. İkinci özellik olan makine özelliklerinde ise makinenin hızına, işlem görmeyi bekleyen iş sayısına ve bekleyen işlemlerin toplam sürelerine bakılarak

atamalar yapılabilir. Sevk kuralları arasında en temel olanları olarak en erken bitiş zamanı olan Earliest release date first (ERD), en erken teslim tarihi olan (Earliest due date first(EDD)) ya da en az artık atama (Minimum slack first(MS)) kuralları kullanılmaktadır. Sevk kuralları statik ve dinamik olmak üzere ikiye ayrılabilir. Statik sevk kuralları zamana bağlı değildir. Bu kurallar sadece iş ya da makine verilerine göre değişkenlik gösterirler. Dinamik kurallar ise zamana bağlıdır. Bu kurallarda zamanla değişkenlik gösteren hesaplamalar yapılır. .

Tablo 4.3. Sevk kuralları ve açıklamaları

Kural No	İsim	Açıklama	Hesaplama
0-1-2	WATC	Weighted Apparent Tardiness Cost	$I_i = \frac{w_i}{p_i} e^{\left(\frac{\max(\text{slack}, 0)}{K\bar{p}}\right)}$
3-4-5	ATC	Apparent Tardiness Cost	$I_i = \frac{1}{p_i} e^{\left(\frac{\max(\text{slack}, 0)}{K\bar{p}}\right)}$
6	WMS	Weighted Minimum Slack	$I_i = -(\text{slack})w_i$
7	MS	Minimum Slack	$I_i = -(\text{slack})$
8	WSPT	Weighted shortest process time	$I_i = \frac{w_i}{p_i}$
9	SPT	Shortest process time	$I_i = \frac{1}{p_i}$
10	WLPT	Weighted longest process time	$I_i = \frac{p_i}{w_i}$
11	LPT	Longest process time	$I_i = p_i$
12	WSOT	Weighted shortest operation time	$I_i = \frac{w_i}{p_{ij}}$
13	SOT	Shortest operation time	$I_i = \frac{1}{p_{ij}}$
14	WLOT	Weighted longest operation time	$I_i = \frac{p_{ij}}{w_i}$
15	LOT	Longest operation time	$I_i = p_{ij}$
16	EDD	Earliest due date	$I_i = \frac{1}{d_i}$
17	WEDD	Weighted Earliest due date	$I_i = \frac{w_i}{d_i}$
18	ERD	Earliest release date	$I_i = \frac{1}{a_i}$
19	WERD	Weighted earliest release date	$I_i = \frac{w_i}{a_i}$
20	SIRO	Service in random order	random
21	FIFO	First in first out	$I_i = \frac{1}{a_i}$
22	LIFO	Last in first out	$I_i = a_i$

I : Priority index, $\text{slack} = d_i - p_i - t$

Bir makine boşa çıktığında tüm kalan işler için index(I) değerleri hesaplanır ardından en yüksek index değerine sahip iş işlem için seçilir.

Sevk kuralları seri imalat rotaları için hangi işin hangi sırada yapılacağına karar verilen kurallardır. Şimdiye kadar literatürde değişik sevk kuralları çizelgeleme problemleri için kullanılmış ve sevk kurallarının avantajlı yönleri belirtilmiştir. En kısa teslim tarihine göre (Earliest Due Date (EDD)) sevk kuralı, kalan operasyonların toplam proses zamanına göre (smallest processing time (SPT), ilk giren ilk çıkar (FIFO), gibi sevk kuralları için çalışmalar yapılmıştır [137].

Bu sevk kurallarının performanslarının değerlendirildiği yüzlerce karşılaştırma çalışması yapılmıştır. Hangi sevk kuralının çizelgeleme probleminin global en iyi değerini bulduğuna dair kesin bir çalışma bulunmamaktadır. Bazı sevk kuralları bazı çizelgeleme probleminde çok iyi sonuçlar verirken bazı problemler için ise kötü sonuçlar vermektedir. Sevk kurallarının verimli çalışabilmesi problemin yapısına, işlemlerin kısıtlarına ve üretimin amaç fonksiyonuna göre değişiklik göstermektedir. Ancak birçok literatür çalışmasından çıkarılacak sonuç sevk kurallarının dinamik koşullara göre değişerek sisteme uygun hale gelmesiyle en iyi çözümlere ulaşılmakta olduğudur. Chibani ve Pierreval [138] 2009 yılında yaptıkları çalışmada problemin yapısına göre en iyi sevk kuralının hangisi olduğuna karar verecek bir yapay sinir ağı geliştirmişlerdir. Yapılan çalışma göstermiştir ki geliştirilen yapay sinir ağı otomatik şekilde verimli bir sevk kuralını belirleyebilmektedir.

Kaban ve ark. [139] yaptıkları çalışmada simülasyon kullanarak sevk kurallarının karşılaştırmasını yapmışlardır. Bu çalışmada 44 hibrit ya da tek sevk kuralını değerlendirmiştir. Çalışma sonucunda MTWR (Most Total Work Remaining) sevk kuralının en iyi sonuca ulaştığı belirtilmiştir. Genel olarak hibrit sevk kurallarının tekil sevk kurallarına göre daha başarılı sonuçlar verdiği söylenmiştir. Bu çalışma için otomotiv endüstrisinden bir atölye tipi üretim yapan bir üretim alanı seçilmiştir. Ayrıca, çalışmada sevk kurallarının çizelgelemenin global performansı için ne kadar önemli olduğu da belirtilmiştir [139].

4.2.2. Teslim tarihi belirleme kuralları

Önceden belirlenmiş teslim tarihi ile ilgili makine çizelgeleme problemi literatürde çalışılmış bir konudur. Ancak, teslim tarihi belirleme problemi başlı başına bir karar verme sürecidir ve üretim verimliliği açısından büyük öneme sahiptir. İş çizelgeleme ile birlikte yapılan teslim tarihi belirleme fonksiyonlarının avantajları birçok çalışmada gösterilmiştir [140]–[143]. Bu çalışmaların hepsinde ortak teslim tarihi belirleyerek çizelgeleme yapılmıştır. Ancak yalın üretimin gereği olarak ve günümüz dünyasının büyüyen esnek üretim ihtiyacı ile birlikte işler genellikle farklı teslim tarihlerine sahip olmaktadır.

The total work content (TWK) literatürde sıkça kullanılan teslim tarihi atama metotlarından birisidir. TWK aşağıda formülde gösterildiği şekilde hesaplanır [144]–[146].

$$d_i = a_i + k(p_i) \quad (4.8)$$

Tablo 4.4.'te tezde kullanılacak teslim tarihi belirleme kuralları açıklamaları ile birlikte verilmiştir. Tabloya göre RDM kuralı hariç tutulduğunda 18 adet kuralın çalışma için seçildiği görülebilir. TWK için 3 adet katsayı belirlenerek 3 adet kural oluşturulur. SLK için ortalama işlem süresi hesaplanır ve 3 adet katsayı ile ortalama işlem süresi çarpılarak teslim tarihi hesaplanır. NOPPT için 3 adet katsayı belirlenerek teslim tarihi hesaplanır.

Tablo 4.4. Teslim tarihi belirleme kuralları

Kural No	Kısaltma	Açıklama	Denklemler
0,1,2	SLK	Slack	$d_i = a_i + p_i + q_x$
3,4,5	WSLK	Weighted slack	$d_i = a_i + p_i + w_{1x}q_x$
6,7,8	TWK	Total work content	$d_i = a_i + k_x p_i$
9,10,11	WTWK	Weighted total work content	$d_i = a_i + w_{1x}k_x p_i$
12,13,14	NOPPT	Number of operations plus processing time	$d_i = a_i + p_i + 5k_x o_i$
15,16,17	WNOPPT	Weighted number of operations plus processing time	$d_i = a_i + p_i + 5w_{1x}k_x o_i$
18	RDM	Random-allowance due dates	$d_i = a_i + N \sim (3P_{av}, P_{av})$
19,20,21,22,23,24,25,26,27	PPW	Processing-time-plus-wait	$d_i = a_i + k_x p_i + q_x$
28,29,30,31,32,33,34,35,36	WPPW	Weighted processing-time-plus-wait	$d_i = a_i + w_{1x}k_x p_i + w_{2x}q_x$

4.2.3. İşlerin geliş zamanlarının belirlenmesi

Her bir iş sisteme ortalaması $1/\mu$ olan üstel dağılıma göre gelir. Aşağıda üstel dağılım için kullanılacak formül verilmiştir.

$$a_j = \frac{1}{\mu} = \frac{\mu_p \mu_g}{UM} \quad (4.9)$$

a_j : j işinin sisteme geliş zamanı

μ : işlerin geliş sıklığı

μ_p : operasyon başına ortalama işlem zamanı

μ_g : her bir iş için ortalama operasyon sayısı

U : Atölye kullanım oranı

M : atölyedeki makine sayısı

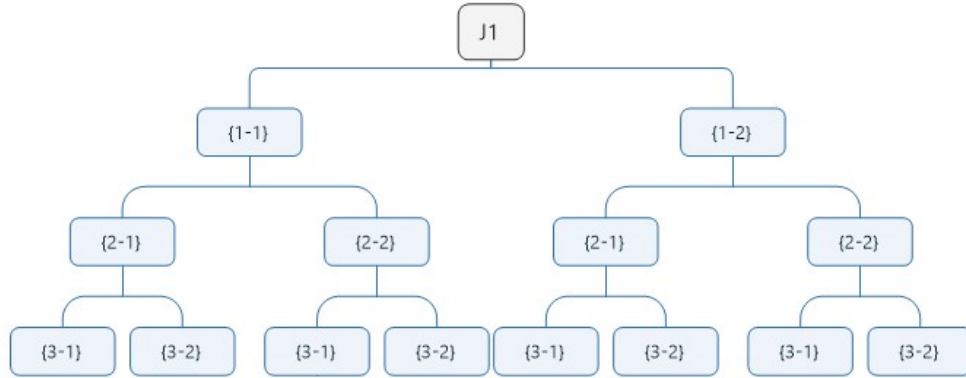
Üstel dağılımın olasılık dağılım fonksiyonu (PDF) ve kümülatif dağılım fonksiyonu (CDF)'si Tablo 4.5.'te verilmiştir.

Tablo 4.5. Üstel dağılım fonksiyonları

PDF	CDF
$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$	$F(x) = \begin{cases} 0 & x < 0 \\ 1 - e^{-\lambda x} & x \geq 0 \end{cases}$

4.2.4. Rotaların belirlenmesi

Rotaların belirlenebilmesi için tüm alternatiflerin değerlendirilmesi gerekmektedir. Alternatif rota sayıları iş sayısına ve rota sayısına göre değişmektedir. Alternatif rota sayısının hesaplanması için o^n formülü kullanılabilir. Örnek bir çalışma için iş sayısı $n = 3$ ve operasyon sayısı $r = 2$ olduğu durumda $2^3 = 8$ adet alternatif rota bulunmaktadır. Rota sayısı ve iş sayısı arttıkça alternatif rotaların sayısı da artacaktır. Şekil 4.2.'te 3 iş ve 2 rota için örnek bir alternatif rota sayıları şekli gösterilmektedir. Şekilde {iş-rota} notasyonu kullanılmıştır.



Şekil 4.2. Alternatif rota sayısı için bir örnek gösterim.

Örnek olarak {1-1}{2-1}{3-1} dalı alınırsa shorted processing time(SPT)'ye göre işlerin sıralanması işlemine geçilebilir. Çizelgelemenin birinci adımında şu işlem zamanları için sıralama işlemi yapılır: p_{11} - p_{21} - p_{32} . Bu adımda, işlem zamanları içerisinde birinci işin en düşük operasyon süresine sahip olan operasyon ilk önce çizelgelemeye girer.

4.2.5. Performans ölçütleri

Çizelgeleme problemlerinde genellikle ortalama işin bitiş süreleri, ortalama geciken iş sayısı ve en son bitecek işin bitiş süresi gibi performans ölçütleri amaç fonksiyonu olarak alınmıştır. Burada dinamik ortamda çalışan bu problem için minimum gecikme, erken tamamlanma ve işe verilen teslim tarihi cezaları amaç fonksiyonu olarak alınmıştır. Gecikme, işin bitiş zamanı ile teslim süresi arasındaki pozitif fark olarak tanımlanır. Ancak bu tez çalışmasında TZÜ'nün bir gereği olarak erken bitirme de gecikme olarak görülmüştür. Çünkü erken tamamlanma durumlarında da işletme tarafından karşılanması gereken bir takım maliyetler söz konusudur. Buna göre işlerin ağırlıkları ile birlikte erken tamamlanma, geç tamamlanma ve işlere verilecek teslim tarihleri ile birlikte performans ölçüsü tamamlanmış olur. Bu çalışma için belirlenecek olan teslim tarihleri ayrıca bir fonksiyon olarak ele alınacaktır.

Başlangıçta orijinal bir süreç planı ve zamanlaması göz önüne alındığında, atölye, sorunlar oluşana kadar çizelgeye göre yürütecektir. Çizelgelemede, uygun imalat kaynaklarını seçilerek ve operasyonları sıralanarak bir beklenmeyen durum olana kadar çizelgeleme işlemi devam edecektir. Ancak günümüz atölyelerinde muhakkak beklenmeyen, istenmeyen ya da bozukluk yaratacak diye tarif edebileceğimiz sıkıntılar ortaya çıkacaktır. Bu tip durumlar: makine arızalanması, acil siparişin çizelgelemeye dahil edilmek zorunda olunması vb. gibi durumlar olabilir. Bu tip durumların çözümü için, çizelgelemelerin önceden kestirilemeyen ve acilen çözülmesi gereken şartlara uyum sağlayacak şekilde geliştirilmesini zorunlu kılmaktadır. Meydana gelebilecek olumsuz durumlara göre kendini ayarlayabilen bir sistemin geliştirilmesi sayesinde, süreç planları belirlenen işlerin çizelgelenmesi ve teslim tarihlerinin belirlenmesi aşamalarında ortaya çıkabilecek verimsiz durumlar ortadan kaldırılabılır ve çizelgeleme hedeflerine ulaşılabilir.

Çizelgeleme aşamasında beklenmeyen bir olayla karşılaşılması durumunda yeni bir süreç planının oluşturulması ve çizelgelemenin oluşturulan yeni süreç planına göre yapılması gerekebilir. Bu durumda çizelgelemesi yapılan işler ile devam etmek ya da

yeniden çizelgeleme süreci devreye girer. Yeniden çizelgeleme için 2 varsayım vardır. Bunlardan birincisi, yeniden planlama ve çizelgeleme için gerekli sürenin ihmal edilmesidir. Yani, yeniden planlama ve çizelgeleme için herhangi bir süre harcanmaz ve yeniden çizelgeleme yapılır yapılmaz yeni süreç planlar ve çizelgeleme devreye girer. İkincisi, makine iş yaptığında dinamik bir durum meydana geldiğinde eğer makine yeniden çizelgeleme yapıldıktan sonra değişmezse işlem devam eder. Diğer durumda işlemin tekrar sisteme girmesi gerekecektir [130].

Yeni bir iş sisteme geldiğinde uygun bir teslim tarihi ataması işe yapılır. Teslim tarihi yeni gelen işe göre ve işin sahibi müşteriye göre ağırlık verilir. Daha sonra karar verilen sevk kuralına göre işin çizelgelemesi yapılır. İşlerin işlem zamanları normal dağılan bir seri olduğu için teslim tarihinden önce işin bitmesi kesin değildir. Fakat her iş için teslim tarihinden önce bitmesi hedeflenir. Ayrıca tam zamanında üretimin bir gereği olarak işlerin erken bitmemesi de hedeflenir. Bu noktada sisteme girişi erken olan işe daha yakın teslim tarihleri atamak yararlı olacaktır.

4.2.6. Simülasyon çalışması

Dinamik çizelgeleme öncelikle statik çizelgeleme olarak çalışır. Atölyeye yeni bir iş geldiğinde, iş listesi güncellenir ve problem dinamik çizelgelemeye dönüşür. Bu çalışmada da dinamik bir çizelgeleme problemi çalışılmıştır. Dinamik problemi çözmek için, sevk ve teslim tarihi kurallarının performansını doğrulamak üzere her bir atölye için ayrı ayrı kesikli olay simülasyonları oluşturulmuştur. Çalışmada simülasyonlar için daha etkili çözümler elde etmek amacıyla atölyeye gelen işlerin geliş zamanlarının üstel dağılıma, operasyon süreleri de normal dağılıma uygun dağıldığı varsayılmıştır. Kısaca, atölyedeki işler farklı rotalara sahiptir ve atölyedeki makineleri rotadaki sıraya göre ziyaret ederek üretim sürecini tamamlar. Atölyeye yeni bir iş geldiğinde, geliştirilen algoritma tarafından seçilen rotaya işlerin operasyonları göre makinenin işlem kuyruğuna girer. Makineler, bekleyen operasyonlardan geliştirilen algoritma tarafından üretilen sevk kuralı özelliğine göre tercih yapar. İşlerin teslim tarihleri hesaplandıktan sonra simülasyon çalıştırılır ve her bir iş için

tamamlanma zamanları simülasyondan çıktı olarak elde edilir. Atölyedeki tüm işler bitene kadar simülasyon çalıştırılır.

Simülasyon çalışmanın adımları aşağıdaki gibidir;

- Algoritma adımlarını çalıştırın ve çözüm için bir birey oluşturun.
- Yeni iş atölye konumuna geldiğinde, ağırlıklar, çalışma süreleri, operasyon önceliği ve süreleri ve iş rotasını belirleyin.
- İş için seçilen vade tarihi atama kuralına göre vade tarihini hesaplayın.
- İşin ilk operasyonu, atanacak makinenin kuyruğunu girer.
- Makine, seçili gönderme kuralına göre bekleyen operasyonlardan işlemi seçer.
- İşin son operasyonu için tamamlanma süresi, işin kalkış saatine göre belirlenir.
- İş için E/T ve son tarihler objektif işlevi kullanılarak hesaplanır.
- Çözüm algoritmayı kullanarak optimize edilmiştir.

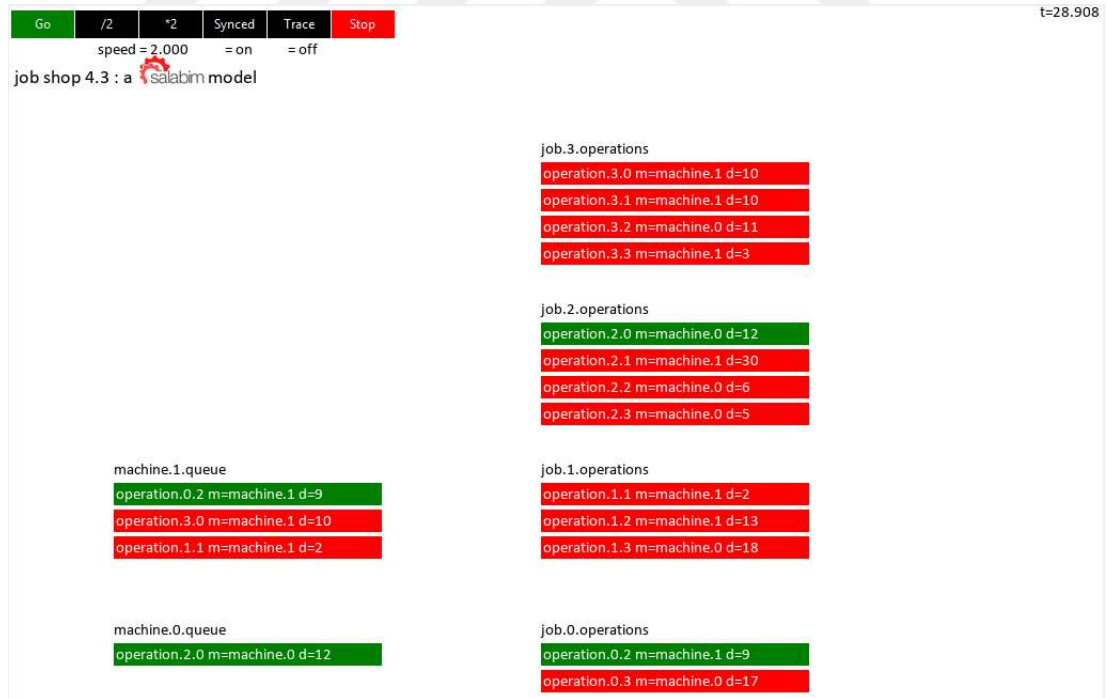
4.2.7. Örnek çalışma

Örnek vaka çalışması için kolay takip ve hesaplama nedeniyle bir mini atölye oluşturulmuştur. Mini atölyede 4 adet makine ve 2 adet iş bulunmaktadır İşlem zamanları ortalaması ve standart sapması olan normal dağılıma uygun olarak üretilen süreler olarak verilmektedir. Oluşturulan girdi verileri Tablo 4.6.'da gösterilmiştir.

Tablo 4.6. Mini atölye için üretilen işlerin operasyon süreleri ve rota bilgileri.

İş No	Rota No	Operasyon Süreleri				Makineler				Ağırlık	Geliş Zamanı
0	0	10	9	23	17	0	1	1	1	1	16
	1	19	2	27	1	0	1	0	0		
1	0	1	12	45	7	1	0	1	1	0.33	20
	1	1	8	10	13	0	1	1	0		
2	0	7	23	3	4	1	1	1	0	0.33	22
	1	23	17	5	13	0	0	1	1		
3	0	10	8	7	10	1	0	1	0	1	45
	1	23	4	13	6	0	1	1	1		

Simülasyon çalışması için Python programlama dilinde bir simülasyon paketi olan salabim paketi kullanılarak bir adet animasyon geliştirilmiş olup simülasyonun görselleştirilmesi sağlanmıştır. Şekil 4.3.'te animasyona ait ara yüz görülmektedir. Animasyon için de öncelikle mini atölye üretilmiş ve kullanılmıştır. Mini atölyedeki olayların görsel halleri geliştirilen program sayesinde gösterilmiştir. Şekilden atölyedeki makinelere ait kuyruklar ve işlerin operasyonları arasındaki kuyruklar görülebilir. Makine bekleyen operasyonlar arasından seçimi sevk kuralına göre yapar. İşlere ait operasyonlar, operasyonlara ait operasyon süreleri ve işlerin rota bilgileri ayrıca şekilde gösterilmiştir. İşin operasyonları arasında öncelik ilişkileri bulunmaktadır. Buna göre, önceki operasyon bitmeden işin diğer operasyonu işlem alınamaz. Kırmızı olan operasyonlar beklemekte yeşil olan operasyonlar ise işlem görmektedir. Animasyonun sağ üst köşesinde bir saat yardımıyla operasyon zamanları takip edilebilir durumdadır. Bir operasyonun işlemi bittikten sonra animasyondan çıkarılarak tüm operasyonlar bitene kadar animasyon çalıştırılmıştır.



Şekil 4.3. Simülasyon çalışmasının görselleştirilmesi

Animasyon sonunda simülasyonun adım adım sonuçları Ekler bölümünde Ek-2 olarak verilmiştir.

Tablo 4.7. Çizelgeleme sonuçları

Chromosome None Fitness 26.074416666666664 Route [0, 0, 1, 0]
Due Date Rule 11 Dispatchine Rule 8

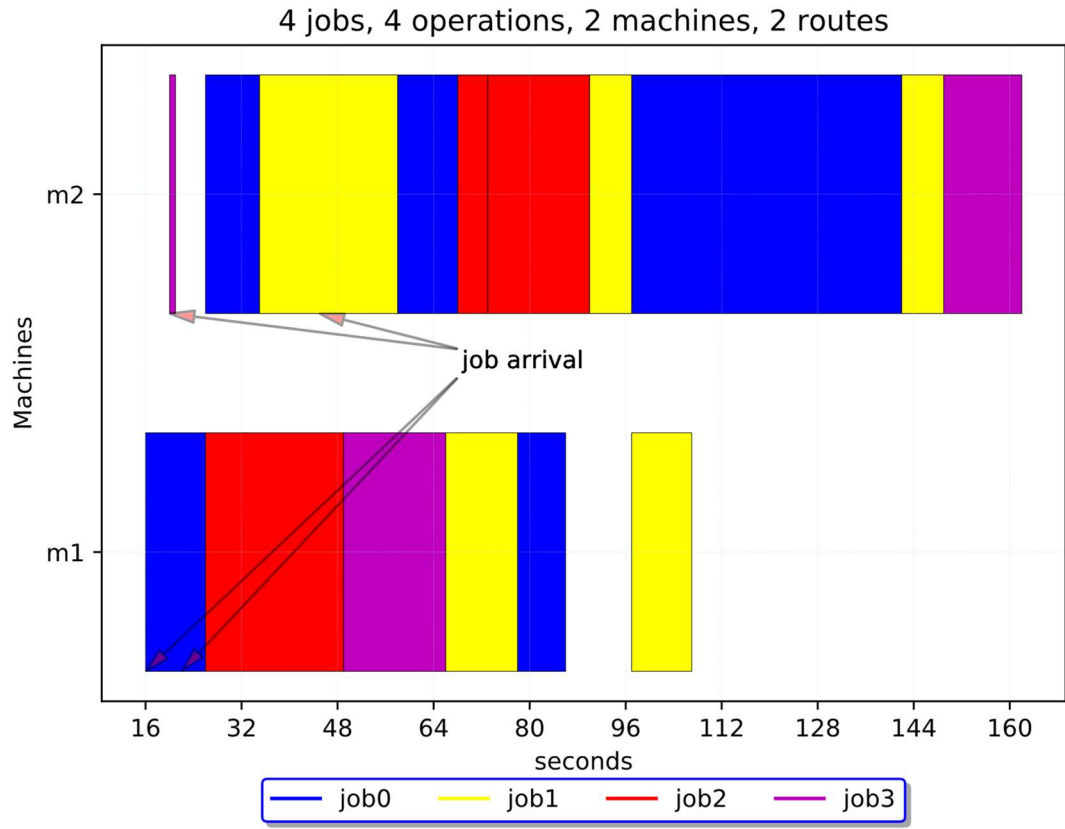
job 0 arrival time= 16 departure time= 90 due-date= 175.29999999999998 weight= 1.0
operation.0.0 m=machine0 d=10 start time= 16 finish time= 26
operation.0.1 m=machine1 d=9 start time= 26 finish time= 35
operation.0.2 m=machine1 d=23 start time= 35 finish time= 58
operation.0.3 m=machine1 d=17 start time= 73 finish time= 90

job 1 arrival time= 20 departure time= 149 due-date= 551.8181818181818 weight= 0.33
operation.1.0 m=machine1 d=1 start time= 20 finish time= 21
operation.1.1 m=machine0 d=12 start time= 66 finish time= 78
operation.1.2 m=machine1 d=45 start time= 97 finish time= 142
operation.1.3 m=machine1 d=7 start time= 142 finish time= 149

job 2 arrival time= 22 departure time= 162 due-date= 496.54545454545456 weight= 0.33
operation.2.0 m=machine0 d=23 start time= 26 finish time= 49
operation.2.1 m=machine0 d=17 start time= 49 finish time= 66
operation.2.2 m=machine1 d=5 start time= 68 finish time= 73
operation.2.3 m=machine1 d=13 start time= 149 finish time= 162

job 3 arrival time= 45 departure time= 107 due-date= 139.5 weight= 1.0
operation.3.0 m=machine1 d=10 start time= 58 finish time= 68
operation.3.1 m=machine0 d=8 start time= 78 finish time= 86
operation.3.2 m=machine1 d=7 start time= 90 finish time= 97
operation.3.3 m=machine0 d=10 start time= 97 finish time= 107

Geliştirilen algoritma sonucu üretilen sonuç çözümünde [11, 8, 0, 0, 1, 1] çözümü değerlendirilmiş olup performans değeri 26.07 olarak hesaplanmıştır ve çözüme ait Gantt şeması Şekil 4.4.'te verilmiştir.



Şekil 4.4. Örnek çalışma sonucunda elde edilen Gantt şeması.

BÖLÜM 5. DIPPSDDA PROBLEMİNİN META-SEZGİSEL ALGORİTMALAR İLE ÇÖZÜMÜ

Bu bölümde DIPPSDDA problemine uygun geliştirilen meta-sezgisel algoritmalarla ait metodolojiler anlatılmıştır. DIPPSDDA probleminin çözümü için kesin sonuç veren matematiksel modeller yetersiz kalmaktadır. Meta-sezgisel algoritmalar genellikle bu tip karmaşık NP-Zor sınıfında yer alan problemlerin çözümü için kullanılır. Meta sözcüğü genellikle yaklaşık algoritmalar anlamına gelmektedir. Ancak tüm algoritmalarda olduğu gibi sezgisel algoritmalarda da birtakım limitler bulunmaktadır. Problemin karmaşıklığı nedeniyle en iyi sonuç için üstel oranda artan bir çözüm uzayı ile çalışılması zorunlu bir hale gelmiştir. Bu nedenle kullanılacak sezgisel algoritmalarındaki iterasyon sayısı belirli bir sayıda tutulmuştur. Bu bölümde genetik algoritmaların, tabu aramanın, tavlama benzetimi algoritmasının ve parçacık sürü optimizasyonu algoritmasının genel özelliklerinden bahsedildikten sonra probleme ilişkin geliştirilen sezgisel ve bu sezgisellere ait hibrit algoritmalar kullanılarak çözümler önerilecek ve en iyiye yakın sonuçlar bulunmaya çalışılacaktır.

5.1. Genetik Algoritmalar (GA)

GA, temel prensipleri Holland tarafından geliştirilen popüler evrimsel algoritmalarından birisidir [147]. GA doğadaki evrimsel olayları taklit ederek optimizasyon problemlerini çözmeye çalışır ve 4 adımda çevrimini tamamlayan bir algoritmadır. Bu çevrimler, seçim, tekrar üretim, çaprazlama ve mutasyon operatörlerinden oluşmaktadır [148]. Genel GA akışı Şekil 5.1.'de gösterilmektedir. DIPPSDDA problemi öncelikle genetik algoritmalar kullanılarak modellenerek çözüm olarak GA'nın kromozom yapısı temel alınmıştır.

Genetik Algoritma	
Input	pop_size, probab_mut, probab_crx
Output	S_{best}
1:	BEGIN Initialize population
2:	Evaluate population
3:	S_{best} get best solution
4:	REPEAT
5:	Select parents
6:	Crossover
7:	Mutation
8:	Replace population
9:	WHILE stopping condition
10:	END

Şekil 5.1. Genetik algoritma için genel yalancı kod yapısı.

GA çok büyük çözüm uzayları etkili ve verimli bir şekilde ile çalışabilir. Bu çalışmadaki problem NP-Zor sınıfına girdiğinden dolayı meta-sezgisel algoritmalarından birinin kullanılması zorunludur. GA akışında problemlere ait çözümler seçim, mutasyon ve çaprazlama operatörleri kullanılarak iyileştirilmeye çalışılır. Bir GA akışı rastgele oluşturulan birey topluluğundan yani popülasyondan başlar. Popülasyon boyutu problemin yapısına göre değişmektedir. Popülasyonu oluşturan bireylere kromozom ya da genom da denilebilir. Kromozom problemin çözümü için bir gösterim şeklidir. İkinci aşamada, çaprazlama ve mutasyona girmek üzer başlangıç popülasyonundan bir kromozom kümesi seçilir. Seçilen kromozomların seçilme şansları kromozomun uygunluk değeri ile doğrudan bağlantılıdır. Kromozomlar stokastik olarak seçilerek yeni popülasyon için yeni kromozomlar(bireyler) üretilir. Seçim aşamasında kromozomların uygunluk değerleri seçilme olasılığı olarak alınabileceği gibi kromozomların uygunluk değerlerine göre sıralandığı ve ardından sıralama değerini de kromozom için seçilme olasılığı olarak alınabilir. Bir sonraki aşamada seçilen bireyler arasında çaprazlama ve mutasyon operasyonları gerçekleştirilir. Çaprazlama operasyonunda seçilen kromozomlar ebeveyn kromozomlar olarak adlandırılır ve ebeveyn kromozomlarından yeni çocuk bireyler üretilerek çözümlerin farklılaşması sağlanır. Ebeveyn bireyler üzerinden rastgele alınan noktalar üzerinde çaprazlama yapılarak 2 ebeveyn bireyden 2 yeni çocuk birey meydana getirilir. Mutasyon operatöründe çocuk bireyler üzerindeki rastgele oluşturulan noktalar farklı bir değer ile değiştirilir ve böylece bir iterasyondaki tüm bireyler üretilmiş olur. Yeni popülasyon başlangıçta belirlenen

popülasyon boyutuna ulaşınca kadar yeniden üretim süreci devam eder. Genellikle yeni popülasyonun uygunluk değerleri ortalamasının bir önceki popülasyonun ortalamasından düşük olması beklenir. Çünkü çaprazlama ve mutasyona girecek bireyler iyi uygunluk değerine sahip bireyler arasından seçilmektedir. Yeni popülasyon üretildikten sonra verilen durdurma kriteri (stopping criteria)'ne göre iterasyonlar devam ettirilir. GA uygulamalarında genellikle, bir çözümün minimum bir kriteri yerine getirmesi, belirli sayıda bir iterasyon sayısına ulaşılması, belirli bir hafıza kullanımına ulaşılması ya da bunların kombinasyonları şekilden durdurma kriterleri verilmektedir.

5.1.1. GA parametreleri

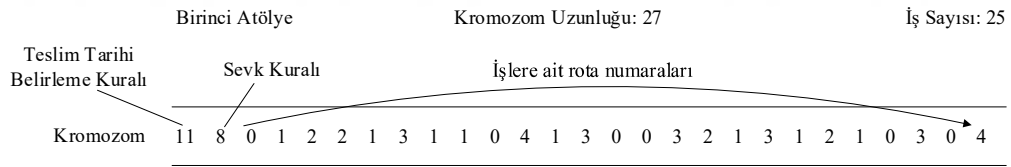
Çaprazlama ve mutasyon olasılıkları her bir iterasyonda çaprazlama ya da mutasyon olup olmayacağını belirleyen olasılıklardır. Eğer çaprazlama ya da mutasyon olmaz ise mevcut çözüm aynı haliyle bir sonraki çözüme geçer. Bunlar gibi, popülasyon boyutu her bir iterasyondaki popülasyonun sahip olacağı kromozom sayısını ifade eder. Eğer fazla ya da düşük sayıda popülasyon boyutu ile çalışılırsa algoritmada çeşitli verimsizlikler meydana gelebilir. Bu nedenle her bir parametrenin dengeli bir yapıda tutulması gerekir. GA için geliştirilen kromozomdaki her bir gen için ayrı seçilme olasılıkları tanımlanmıştır. Böylece dominant genler ile çalışılmıştır. Tablo 5.1.'de kullanılan parametreler gösterilmiştir.

Tablo 5.1. Birinci atölye için kullanılan GA parametreleri

Parametreler	Değerler
Popülasyon boyutu	10
İterasyon sayısı	400
Kromozom boyutu	27
Çaprazlama oranı	0.7
Mutasyon oranı	0.3
Gen seçilme olasılığı	0.5
Çaprazlama popülasyonu boyutu	6
Mutasyon popülasyonu boyutu	4
Çaprazlama noktası sayısı	2
Mutasyon noktası sayısı	1
Kromozom seçilme olasılıkları	[0.3, 0.2, 0.15, 0.12, 0.10, 0.07, 0.03, 0.02, 0.006, 0.004]
Gen seçilme olasılıkları	[0.25, 0.25] + [0.5]

5.1.2. Kromozom yapısı

Bu çalışmada oluşturulan GA modelinin başlangıç noktası daha önce Demir ve Taşkın [3] tarafından anlatılan modelle benzerlik göstermektedir. Çalışmada oluşturulan her bir kromozom problem için bir çözümü göstermektedir. Örnek bir kromozom görünümü Şekil 5.2.'de gösterilmiştir. Şekilde görülebileceği gibi her kromozomun yapısında yatay halde belirli sayıda gen hücresi bulunmaktadır. Bir kromozom teslim tarihi belirleme kuralı, sevk kuralı ve işlere atanacak rota bilgilerinden oluşur. Burada kromozomun içerisinde hangi işin hangi operasyonunun makinede işleme başlama zamanı ve işlemini bitiriş zamanı verilmemiştir. Çünkü bu bilgiler işlerin rota bilgileri içerisinde tutulmaktadır. Dolayısıyla bireydeki her bir genin belirlenmesi durumunda problemin çözümüne ulaşılmış olur. Çalışmadaki bir varsayım olan operasyonlar arasındaki öncelik ilişkileri gereği, işlere ait operasyonlardan önce bitmesi gereken operasyon bitmeden operasyonlar işleme başlayamazlar. Ayrıca yine varsayım olarak verilen bir makinenin arıza yapmaması kuralı da işletilecektir.



Şekil 5.2. Bireylerin(Kromozomların) gösterilişi.

Her bir iterasyonda eklenecek yeni kromozomlar için bir başlangıç popülasyonunun oluşturulması gerekmektedir. Dolayısıyla GA, verilen veriler içinde rastgele oluşturulan bir çözüm popülasyonu ile başlar. İlk aşamada sevk kuralı da rastgele olarak sıradaki işin seçilmesine yardımcı olur. Teslim tarihi atama kuralı, sevk kuralı ve işlere ait rota bilgileri rastgele verildikten sonra seçim operasyonuna geçilir.

5.1.3. Seçilim ve tekrar üretim

Rastgele oluşturulan başlangıç çözümündeki her bir kromozom performans ölçütüne göre değerlendirilir ve uygunluk değerleri hesaplanır. Burada, seçilecek kromozomların bir minimizasyon problemiyle çalıştığımız için en düşük uygunluk değerine sahip olması beklenir. Seçilim yapılırken “konuma göre seçme (ranking method)” yöntemi tercih edilmiştir. Bu yöntemle göre her bir kromozom popülasyonda yer aldığı konuma göre sonraki popülasyonlara aktarılma ihtimali taşımaktadır. Aynı zamanda, bir kromozomun popülasyonda en iyi uygunluk değerinden en kötü uygunluk değerine göre kromozomlar sıralandığında kaçınıcı sırada olduğu seçilme olasılığını belirlemektedir. Konuma göre seçme metodunun kullanılmasının nedeni, GA’daki son iterasyonlara doğru sıralamanın en yukarısındaki kromozom ile en aşağısındaki kromozom arasındaki farkın giderek azalmasıdır. Böylece her iterasyonda kromozomların sıralamadaki yere göre tercih edilmesi sağlanmış olur.

5.1.3.1. Kromozomların konumuna göre seçilmesi

Uygunluk değerine göre seçimde kromozomların popülasyondaki konumuna bakılmadan seçilme olasılıkları hesaplanır. Örneğin uygunluk değerine göre yapılan rulet seçimi yönteminde her kromozom için uygunluk değeri hesaplanır ve kromozomların uygunluk değerleri toplam uygunluk değerine bölünerek kromozomun seçilme olasılığı hesaplanır. Bu sayede yeniden seçimdeki rassallık sağlanmış olur. Rulet çemberi gibi metotlar çok fazla kullanılıyor olmasına rağmen bu çalışmada uygunluk değerine göre yapılan olasılık dağıtımlarının kullanılması problemde kullanılan kromozomların son iterasyonlara doğru uygunluk değerlerinin birbirine çok yaklaşması nedeniyle çok doğru olmayacaktır. Bu durumu bir örnek ile göstermemiz gerekirse; örneğin elimizde 10 kromozomdan oluşan bir başlangıç çözümü olsun. Başlangıç popülasyonunda çeşitlilik fazladır en iyi değer ile en kötü değer arasında önemli bir fark vardır. Bu durumda en iyi kromozomun seçilme olasılığı yüksek olacaktır. Birkaç jenerasyon sonra en iyi, en kötü ve ortalama uygunluk değerleri

arasındaki fark azalacaktır. Bu durumda en iyi ile en kötü arasındaki seçilme olasılığı azalacaktır. Bu örnek Tablo 5.2.'de gösterilmiştir.

Tablo 5.2. Jenerasyonlar arasındaki kromozom seçilme olasılığı farkları

	Kromozom	Uygunluk	Seçilme olasılığı	Konuma göre seçimde seçilme olasılığı
Başlangıç Popülasyonu	En iyi	189.423	0.1277	0.3
	Ortalama	241.53	0.10	0.10
	En kötü	380.76	0.0635	0.004
	Uygunluklar toplamı	2415.3		
Son jenerasyondaki popülasyon	En iyi	159.21	0.1011	0.3
	Ortalama	161.18	0.10	0.10
	En kötü	163.12	0.0987	0.004
	Uygunluklar toplamı	1611.8		

Burada görülen sonuç, son jenerasyonlara doğru en iyi ile en kötü uygunluk değerlerinin birbirine yaklaştığıdır. Son iterasyonda uygunluk değerleri birbirine neredeyse eşit durumdadır. Bu da en iyi ile en kötü arasında seçim yaparken aynı olasılıkları kullanmamıza neden olacaktır. Uygunluk değerine göre seçilimin yerine, kromozom konumuna göre sıralama ve olasılık ataması yapmak bu problem için daha doğru olacaktır. Konuma göre seçim metodunda kromozomlar bulundukları sıralamaya göre tercih edilme şanslarına sahip olur ve tüm jenerasyonlarda bu olasılıklar sabittir. Bu çalışmada 10 adet kromozom için tercih edilen olasılıklar [0.3, 0.2, 0.15, 0.12, 0.10, 0.07, 0.03, 0.02, 0.006, 0.004] şeklindedir.

5.1.4. Çaprazlama

2 ebeveynden 2 yeni birey oluşturmak için çaprazlama yapılır. Burada, çaprazlama için rastgele bir nokta belirlenir ve çaprazlama bu noktada yapılır. Rastgele nokta belirlenirken dominant genlerin daha fazla seçilmesi olasılığı dikkate alınır. Çalışmada ilk iki gen yani teslim tarihi belirleme kuralı geni ve sevk kuralı geni dominant gen olarak alınmıştır. Dominant genlerin seçilme olasılığı %50'dir ve diğer genlerin seçilmesi olasılıklarının toplamına eşittir. Bu operatördeki en önemli konulardan birisi de çaprazlama noktası sayısıdır. Küçük atölyeler için örneğin birinci atölye için 27

olan kromozom boyutu atölye büyüdükçe çok daha fazla bir boyuta çıkmaktadır. Bu nedenle her atölye için seçilecek çaprazlama noktası sayısı kromozom uzunluğuna bağlı şekilde hesaplanmıştır. $C_{size} = \lfloor N \times 0.1 \rfloor$ formülü ile atölyelere göre hesaplanan çaprazlama noktası sayısını gösterilmiştir.

Bu tez çalışmasında çaprazlama yöntemi olarak çok noktadan çaprazlama metodu tercih edilmiştir. Bu yöntemle göre öncelikle problemin çözümü için geliştirilen kromozom uzunluğuna göre çaprazlama noktası adedi seçilir. Ardından rastgele ve dominant genleri daha yüksek oranda seçmek üzere noktalar seçilir. Birinci ebeveynden gelen genler ikinci ebeveynden gelen genler ile yer değiştirilir. Böylece çaprazlama gerçekleştirilmiş olur. Şekil 5.3.'te çaprazlamanın nasıl yapıldığı gösterilmektedir.

	Birinci Atölye	Çaprazlama noktası sayısı: 2	Çaprazlama noktaları: [5 12]
Ebeveyn 1	11 8 0 1 2 2	1 3 1 1 0 4 1	3 0 0 3 2 1 3 1 2 1 0 3 0 4
		Çaprazlama noktaları	
Ebeveyn 2	10 10 1 4 1 3	4 0 3 3 2 1 1	3 3 4 0 3 0 2 2 2 2 0 0 4 4
Çocuk 1	11 8 0 1 2 2	4 0 3 3 2 1 1	3 0 0 3 2 1 3 1 2 1 0 3 0 4
Çocuk 2	10 10 1 4 1 3	1 3 1 1 0 4 1	3 3 4 0 3 0 2 2 2 2 0 0 4 4

Şekil 5.3. Çaprazlama gösterimi

5.1.5. Mutasyon

GA'daki ikinci önemli operatör mutasyon operatörüdür. Mutasyon doğada çok nadiren meydana gelir. Bu durum GA'da da kullanılmıştır. İlk yapılan GA uygulamalarında genellikle her bir genin mutasyona uğrama olasılığı 0.001 olarak alınmıştır. Mutasyon ile bir genin yapısının değiştirilmesine önceden tanımlı bir öncelik kuralını ihlal

etmeyecek şekilde izin verilir. Mutasyona uğrayacak genin seçimine rastgele karar verilir ve gen başka alternatif bir çözüm çizelgeleme ile yer değiştirilir. Mutasyona uğrayacak bireyin tercihinde çaprazlama operatöründe olduğu gibi tekrar dominant genlere daha fazla şans verecek şekilde ayarlamalar yapılır. Mutasyona girecek gen sayısı $M_{size} = \lfloor N \times 0.05 \rfloor$ formülü kullanılarak belirlenmiştir.

Mutasyon ile birlikte arama uzayındaki rastgeleliğe bir katkıda bulunulmuş olur. Böylece lokal minimum noktalarından kurtulmaya çalışılmış olur. Mutasyon örneği Şekil 5.4.'te verilmiştir.

	Birinci Atölye					Mutasyon Noktası Sayısı: 1										Mutasyon Noktası: [5]											
Ebeveyn	11	8	0	1	2	2	1	3	1	1	0	4	1	3	0	0	3	2	1	3	1	2	1	0	3	0	4
Çocuk	11	8	0	1	2	1	1	3	1	1	0	4	1	3	0	0	3	2	1	3	1	2	1	0	3	0	4

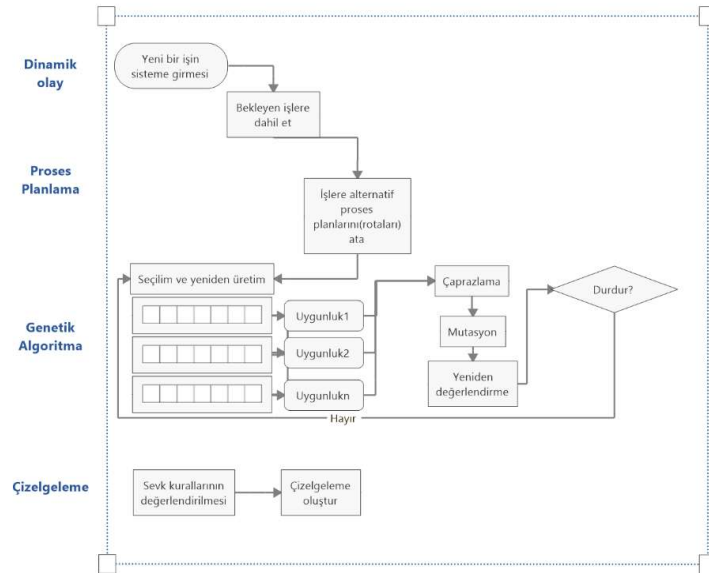
Şekil 5.4. Mutasyon gösterimi

Problemin çözümü için geliştirilen GA algoritmasının adımları Şekil 5.5.'teki gibi özetlenebilir. Problem çözülürken girdi olarak iş sayısı, makine sayısı, operasyon sayısı ve rota sayısı gibi değişkenler verilir. Ardından rastgele oluşturulan başlangıç çözümünde yer alan her bir kromozom için simülasyon çalıştırılır ve uygunluk değerleri hesaplanır. Hesaplanan değer kromozomun uygunluk değeri olarak atanır. Ardından popülasyondan seçilen kromozomlar üzerinde çaprazlama ve mutasyon operatörleri gerçekleştirilir. Mutasyondan sonra yeni popülasyon oluşturulur ve popülasyondaki kromozomlar uygunluk değerlerine göre sıralanır. Maksimum iterasyon sayısına ulaşılan kadar bu süreç devam eder.

Algoritma 1 Genetik algoritma	
1:	BEGIN Atölye Seçimi (n, m, o)
2:	$i = 0$
3:	başlangıç popülasyonu oluştur
4:	REPEAT
5:	$i += 1$
6:	konuma göre seçilim
7:	Çok noktalı çaprazlama
8:	mutasyon
9:	WHILE iteration < iter_size
10:	END

Şekil 5.5. Genetik algoritma yalancı kodu.

Şekil 5.6.'da ise yeni bir iş gelmesi durumunda yapılacak adımlar gösterilmektedir. Yeni bir iş atölyeye geldiğinde atölyede var olan işlerin arasına eklenir. Eklenen iş eğer diğer işlere göre daha yüksek öncelik oranına sahipse hemen çizelgelemeye alınır. Çizelgeleme aşamasında rota bilgisi algoritma tarafından üretilen çözümden alınır. İş atölyeye geldikten sonra çıkış süresine kadar gerekli hesaplamalar yapılır. Atölyedeki tüm işler geldiğinde ve algoritmadaki iterasyon sayısına ulaşıldığında simülasyon durdurulur.



Şekil 5.6. Dinamik olay çözümü için geliştirilen modelin akış şeması.

5.2. Tabu Algoritması (TA)

Lokal arama komşu çözümlerde dolaşarak verilen probleme ilişkin en iyiye yakın çözümler bulma konusunda geliştirilmiştir. Fakat lokal aramanın karşılaştığı en büyük problem lokal minimum ya da lokal maksimum noktalarıdır. Eğer komşu çözüm seçimleri stokastik bir şekilde yapılmazsa lokal minimum veya maksimumlara yakalanma olasılıkları ciddi derecede artar. Bu nedenle tavlama benzetimi algoritmasında kötü çözümlerin dahi seçilebilmesi için bir olasılık değeri belirlenir. Benzer amaçlarla geliştirilen bir diğer algoritma da tabu algoritmasıdır. TA algoritması Glover [149] tarafından geliştirilen ve karmaşık problemler için en iyiye yakın bir sonuç bulmayı meta sezgisel bir algoritmadır. TA lokal arama tabanlı bir başlangıç çözümü ile başlar. Olurlu bir çözüm bulunduğundan sonra, TA her bir iterasyonda daha iyi bir çözüm bulabilmek için komşuluk arama algoritmasını kullanır. Bu çalışmada komşuluk arama algoritması (variable neighborhood search) Şekil 5.8.'de gösterildiği şekilde tanımlanmıştır. TA, her bir iterasyonda bir önceki iterasyonda gerçekleştirdiği kötü çözümleri bir tabu listesine kaydeder. Sonra, maksimum iterasyon sayısına ulaşılan kadar tabu listesindeki kötü bireyleri seçmeyecek şekilde en iyiyi aramaya devam eder [150]. Bu çalışmada, tabu listesi belirli bir sayıda sabit olarak alınmıştır. Çünkü tabu listesinin çok uzun olması durumunda bilgisayar hafızasında yetersizlikler meydana gelmektedir. TA ve SA algoritmalarının çözümü için temel teşkil eden komşuluk arama algoritmasının adımları Şekil 5.8.'de verilmiştir.

Tabu Arama algoritması lokal optimuma yakalanmamak için her iterasyonda en iyi sonucu bir sonraki iterasyona aktarmaya çalışır. Bunun için bir tabu listesi oluşturulur ve tabu listesinde tutulan kötü çözümlerin bir sonraki harekette yer almaması sağlanmaya çalışılır. Tabu listeleri kronolojik bir hafıza yapısına sahiptir ve sürekli güncellenen dinamik bir hafıza sistemi kullanır. Tabu listeleri genel kullanımda istenmeyen çözümlerin tutulması amacıyla kullanılsa da iyi çözümleri tutulduğu bir liste olarak da kullanılabilir.

Algoritma 2 Tabu Arama

```

1:  BEGIN Atölye Seçimi ( $n, m, o$ )
2:  çözüm_x oluştur
3:  i = 0
4:  REPEAT
5:    IF NOT in tabu_list THEN
6:      i += 1
7:      komşu çözüm ile çözüm_y oluştur
8:      IF çözüm_y < çözüm_x THEN
9:        RETURN çözüm_y
10:     ELSE
11:       tabu_list'i güncelleştir
12:     RETURN çözüm_x
13:  WHILE iteration < iter_size
14:  END

```

Şekil 5.7. Tabu arama yalancı kodu

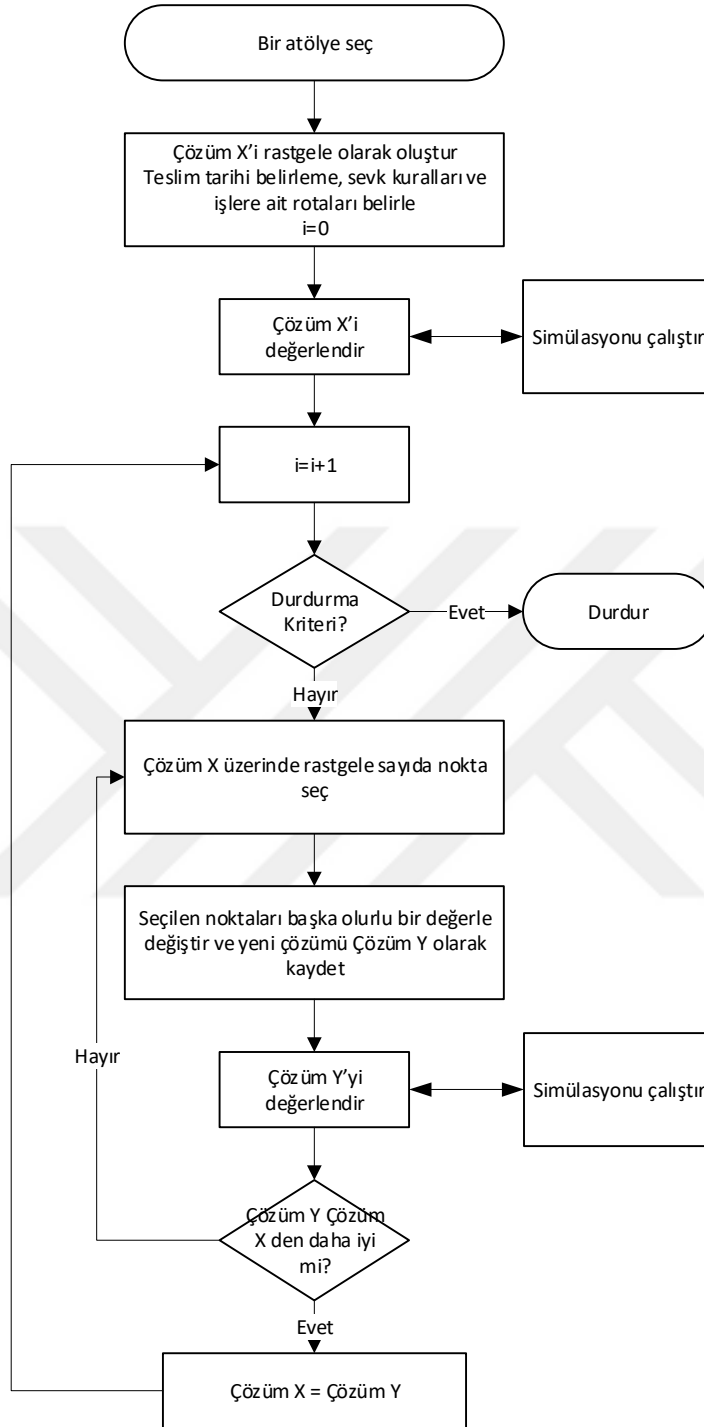
5.3. Tavlama Benzetimi (Simulating Annealing(SA))

SA büyük boyutlu optimizasyon problemlerinin çözümü için Kirkpatrick ve ark. [151] tarafından sıcak metallerin soğuma işlemi taklit edilerek geliştirilmiştir [152]. SA da TA gibi komşuluk arama algoritmasını temel alarak büyük arama uzaylarına sahip hesaplama problemlerinin çözümünde kullanılmıştır. SA'da lokal minimumdan kaçabilmek için bazen kötü sonuçların da kabul edilmesine şans verilir [153]. Bu durum SA'yı tırmanma algoritması(Hill-Climb Algorithm)'den ayıran en önemli etkidir. SA komşulukları uygunluk değerine göre değerlendirir ve en iyi komşuluklara geçiş yaparak bir iterasyonunu tamamlar. SA'da seçilecek çözümün olasılığı bir üstel kontrol fonksiyonu ile hesaplanan bir değere eşitlenir. Bu fonksiyon aşağıdaki denklemde gösterildiği gibi hesaplanır.

$$P_{\text{kabul}} = e^{\left(\frac{E_i - E_t}{k_B T}\right)} \quad (5.1)$$

Burada,

P_{kabul} : Kabul oranı, T : ısı banyosunun sıcaklığı, k_B : the Boltzmann sabiti, E_i : Mevcut durumun enerjisi, E_t : sonraki durumun enerjisi ni göstermektedir.



Şekil 5.8. Komşuluk arama algoritması akışı

 Algoritma 3 Tavlama benzetimi

```

1:  BEGIN Atölye Seçimi ( $n, m, o$ )
2:  çözüm_x oluştur
3:  i = 0
4:  REPEAT
5:    i += 1
6:    komşu çözüm ile çözüm_y oluştur
7:    IF çözüm_y < çözüm_x THEN
8:      RETURN çözüm_y
9:    ELSE IF
10:      $P_{kabul}$  hesapla
11:     IF  $P_{kabul}$  < RASSAL SAYI THEN
12:       RETURN çözüm_y
13:     ELSE
14:       RETURN çözüm_x
15:  WHILE iteration < iter_size
16:  END

```

Şekil 5.9. Tavlama benzetimi algoritması için yalancı kod.

P_{kabul} değeri T değerine bağlıdır. Sıcaklık yüksek olduğunda, fonksiyon değeri artar ve hareket kabul edilebilir. Ve ardından sıcaklık düşer, bu da olasılığı azaltacaktır. Bu yüzden en yüksek sıcaklık noktasında aramaya başlamak gereklidir. Algoritmada, sıcaklık yavaşça düşürülürken, arama işlemi her sıcaklık değerinden belirli sayıda hamle denemeden devam eder.

Soğutma sürecinde algoritmadaki tüm adımlar tamamlanınca sistem ısısının 0'a ulaşması beklenir. Bazı çalışmalarda ise sistem ısısını istenilen bir noktaya kadar soğutulur. Bu çalışmada mevcut sistemin ısısı her atölye için farklı olarak verilmiştir. Çünkü başlangıç ısısı sistemdeki tüm iterasyonların denenebilmesine müsaade edecek kadar yüksek değerde olmalıdır. Eğer başlangıç ısısı küçük bir değer verilirse algoritmanın verimli çalışması engellenmiş olur. Aynı şekilde başlangıç sıcaklığı eğer gerekenden daha yüksek verilirse, bu kez de algoritmadaki bazı çözümlerin değerlendirilmemesine neden olunur.

5.4. Parçacık Sürü Optimizasyonu (PSO)

Kompleks problemlerin çözümü için sayısız yöntem vardır. Bunlardan biri, optimizasyon problemlerini çözmede özelleştirilebilir özelliğinden dolayı sıklıkla kullanılan bir sezgisel algoritma olan PSO yöntemidir [154]. PSO, literatürde, özellikle IPPS problemlerinde yaygın olarak kullanılmaktadır. Örneğin, Ba et al., [155] üretim süresini en aza indirmek için PSO algoritmasını kullanarak bir multimodal program geliştirdiler. Petrovic et al. [156] IPPS problemlerinde arama alanını daha kapsamlı araştırmak ve yerel tercihlerden kaçınmak için PSO algoritması ve kaos teorisine dayanan bir yöntem kullanmaya odaklandılar. Yu et al. [157] iki farklı faz içeren statik ve dinamik IPPS problemini çözmek için GA ve PSO'ya dayalı bir hibrit algoritma geliştirildiler. Petrovic et al. [158] PSO ve kaos teorisi kullanımına dayalı esnek süreç planlarının optimizasyonu için yeni bir algoritma sundular. Wang et al. [159] ayrıca çok amaçlı IPPS problemini PSO kullanarak çözmeye çalıştılar. Bu çalışmalara ek olarak, birçok PSO çalışması literatüre dahil edilmiştir. Bununla birlikte, neredeyse tüm PSO çalışmaları sürekli değere sahip değişkenlere sahiptir. Bu, klasik PSO'nun kesikli optimizasyon modellerini çözmek için yetersiz olacağını gösterir. Çalışmalar sonucunda tam sayılı PSO (Integer PSO-IPSO), kesikli PSO (Discrete PSO-DPSO), ikili PSO, Veeramachaneni PSO, açı modülasyonlu PSO, Dağılım Parçacık Sürü Optimizasyonu (DEDPSO) gibi birçok PSO varyasyonu geliştirilmiştir. Bu eksikliği gidermek için kullanılan PSO varyasyonlarından biri, sürü içindeki partiküllerin değerlerinin olasılık dağılımları ile ifade edildiği bir PSO türü olan tam sayı ve kategorik PSO'dur (ICPSO). Bu çalışmada, teslim tarihi ve sevk kuralları ve işlerin rotaları kategorik veri özelliklerini gösterdiğinden problemi çözmek için ICPSO kullanılmıştır. Çoğu çizelgeleme çalışmasında, toplam çevrim zamanı amaç fonksiyonu olarak kullanılmıştır. Ancak, bu çalışmada, toplam erken tamamlama süresi, toplam gecikme süresi (E / T) ve her bir işin teslim tarihini en aza indirmeyi amaçlanmaktadır.

Önerilen ICPSO algoritması Strasser ve ark. [160] tarafından geliştirilen algoritma temel alınarak geliştirilmiş olsa da literatürdeki mevcut algoritmanın tüm adımları

problemi çözmek için kullanıldığında çözüm değerinin local minimuma takıldığı gözlenmiştir. Bunun üstesinden gelmek için ICPSO, çaprazlama ve mutasyon operatörlerinin eklenmesiyle modifiye edilmiştir. Böylece, geliştirilen algoritma çizelgeleme problemleri için daha kullanışlı hale getirilmiştir. Ek olarak, geliştirilen MICPSO, diğer PSO varyantları ile karşılaştırılmıştır. Çözüme ait pozisyon vektörü, her bir bölüm birbirinden farklı aralıklarla geçerli olduğundan 3 bölüme ayrılmıştır. Çözüm gösteriminin ilk bölümünde, teslim tarihi belirleme kuralı belirlenir, ikinci bölüm sevk kuralının belirlenmesi ve üçüncü bölümde her işin rotalarının olasılık dağılımları belirlenmektedir. Teslim tarihi atama kuralının konum vektörü X_{ddrule} , sevk kuralı konum vektörü ($X_{dsprule}$), ve tüm işlere ait rotaların konum vektörü (X_{routes}) olarak aşağıda gösterilmiştir.

$$X_{ddrule} = [D_{p,1,ddrule}, D_{p,2,ddrule}, \dots, D_{p,N,ddrule}] \quad (5.2)$$

$$X_{dsprule} = [D_{p,1,dsprule}, D_{p,2,dsprule}, \dots, D_{p,N,dsprule}] \quad (5.3)$$

$$\begin{aligned} X_{routes} &= [D_{p,1,routes}^{j=0}, D_{p,2,routes}^{j=0}, \dots, D_{p,N,routes}^{j=0}], [D_{p,1,routes}^{j=1}, D_{p,2,routes}^{j=1}, \dots, D_{p,N,routes}^{j=1}], \dots, \\ &[D_{p,1,routes}^{j=n}, D_{p,2,routes}^{j=n}, \dots, D_{p,N,routes}^{j=n}] \end{aligned} \quad (5.4)$$

Burada $D_{p,i,ddrule}$, $D_{p,i,dsprule}$ sırasıyla i. parçacığa ait teslim tarihi belirleme, sevk kuralı ve rotalara ait çözüm değerlerini göstermektedir. $D_{p,i,routes}^{j=k}$ k işi için rotalara ait olasılık dağılımını belirtir. Bu nedenle, p parçacığının konum vektörü, teslim tarihi atama, sevk kuralı ve tüm işlere ait rotalar olarak verilen toplam vektördür:

$$X_p = [X_{ddrule}, X_{dsprule}, X_{routes}] \quad (5.5)$$

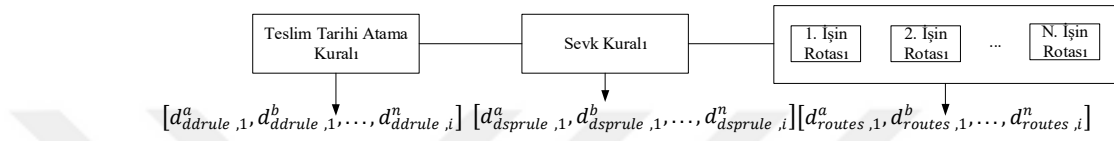
Teslim tarihi atama kuralı, sevk kuralı ve her işin rota olasılık vektörü aşağıda verilmiştir:

$$D_{p,i,ddrul} = [d_{ddrule,1}^a, d_{ddrule,1}^b, \dots, d_{ddrule,i}^n] \quad (5.6)$$

$$D_{p,i,dsrule} = [d_{dsrule,1}^a, d_{dsrule,1}^b, \dots, d_{dsrule,i}^n] \quad (5.7)$$

$$D_{p,1,routes} = [d_{routes,1}^a, d_{routes,1}^b, \dots, d_{routes,i}^n] \quad (5.8)$$

Kategorik verileri ICPSO'da kullanmak için kodlama yöntemi olarak bir parçacık gösterimi Şekil 5.10'daki gibi oluşturulmuştur. Bu göstergedeki tüm boyutlar problemi çözmek için bir kategorik veri oluşturur.



Şekil 5.10. Parçacık gösterimi

Burada $d_{ddrule,i}^a$, $d_{dsprule,i}^a$, $d_{routes,i}^a$ a boyutunun olasılık değişkenini belirtir. Parçacık hızı vektörleri şu şekilde hesaplanır:

$$V_{ddrule} = [\varphi_{ddrule,1}, \varphi_{ddrule,2}, \dots, \varphi_{ddrule,ddrule_{size}}] \quad (5.9)$$

$$V_{dsprule} = [\varphi_{dsprule,1}, \varphi_{dsprule,2}, \dots, \varphi_{dsprule,dsprule_{size}}] \quad (5.10)$$

$$V_{routes} = [\varphi_{routes,1}, \varphi_{routes,2}, \dots, \varphi_{routes,routes_{size}}] \quad (5.11)$$

Hız vektörlerine ait olma ihtimalinden hesaplanan koşullar aşağıdaki gibi verilmiştir:

$$\varphi_{ddrule,i} = [\psi_{ddrule,1}^a, \psi_{ddrule,1}^b, \dots, \psi_{ddrule,i}^{ddrule_{size}}] \quad (5.12)$$

$$\varphi_{dsprule,i} = [\psi_{dsprule,1}^a, \psi_{dsprule,1}^b, \dots, \psi_{dsprule,i}^{dsprule_{size}}] \quad (5.13)$$

$$\varphi_{routes,i} = [\psi_{routes,1}^a, \psi_{routes,1}^b, \dots, \psi_{routes,i}^{routes_{size}}] \quad (5.14)$$

Son olarak, her parçacığın son konum vektörünün denklemleri aşağıdaki gibi gösterilmiştir:

$$S_{ddrule} = [s_{ddrule,1}, s_{ddrule,2}, \dots, s_{ddrule,ddrule_size}] \quad (5.15)$$

$$S_{dsprule} = [s_{dsprule,1}, s_{dsprule,2}, \dots, s_{dsprule,dsprule_size}] \quad (5.16)$$

$$S_{routes} = [s_{routes,1}, s_{routes,2}, \dots, s_{routes,routes_size}] \quad (5.17)$$

Klasik PSO'daki hız denklemi, bu problem için özel olarak daha etkili bir çözüm bulabilmek için değiştirilmiştir. Parçacıklar her iterasyonda konum vektörlerini günceller. Teslim tarihi atama kuralının, sevk kuralının ve her işin rotalarının hız vektörü aşağıdaki gibi verilmiştir:

$$V_{p,ddrule} = \omega V_{ddrule} + c_1 r_1 (Pbest_{p,ddrule} + X_{p,ddrule}) + c_2 r_2 (gbest_{ddrule} + X_{p,ddrule}) \quad (5.18)$$

$$V_{p,dsprule} = \omega V_{dsprule} + c_1 r_1 (Pbest_{p,dsprule} + X_{p,dsprule}) + c_2 r_2 (gbest_{dsprule} + X_{p,dsprule}) \quad (5.19)$$

$$V_{p,routes} = \omega V_{routes} + c_1 r_1 (Pbest_{p,routes} + X_{p,routes}) + c_2 r_2 (gbest_{routes} + X_{p,routes}) \quad (5.20)$$

Burada,

X_i : pozisyon olasılıkları vektörü

$D_{p,1}$: X_i için olasılık dağılımı

N: Sürü boyutu

$k_{iter,i}$: I dükkanındaki iterasyon sayısı sayısı

ω : atalet oranı (inertia rate),

c_1 : biliş sabiti (cognition constant),

r_1 : [0,1] aralığında biliş rastgele sayısı

c_2 : sosyal sabit (social constant)

r_2 : [0,1] aralığında sosyal rastgele sayısı

$p_{g_{best}}$: Küresel en iyi parçacık

$p_{current}$: Mevcut Parçacık

p_{best} : Personal en iyi parçacık

$\mathcal{F}$: Uygunluk Fonksiyonu

Inf: Çok büyük bir sayı

Hız vektörlerinden oluşturulan yeni pozisyon vektörleri aşağıdaki gibi verilmiştir:

$$X_{p,ddrule} = X_{p,ddrule} + V_{p,ddrule} \quad (5.21)$$

$$X_{p,dsprule} = X_{p,dsprule} + V_{p,dsprule} \quad (5.22)$$

$$X_{p,routes} = X_{p,routes} + V_{p,routes} \quad (5.23)$$

Önerilen yöntem Algoritma 1'de sunulmuştur. Algoritmada ilk olarak, rastgele olasılıklar üretilir. Daha sonra, parçacığın teslim tarihi atama ve sevk kuralları ve her iş değerinin güzergahı oluşturulan olasılıklara göre belirlenir. Teslim tarihi atama ve sevk kuralları ve ilk çözümdeki her bir işin rotaları da parçacıkların en iyi kişisel değerleri olarak kaydedilir. Böylece, ilk sürü elde edilmiş olur. Sürüde en iyi uygunluk değerine sahip olan parçacığın uygunluk değeri, küresel en iyi değerler (en iyi) olarak kaydedilir. Aynı zamanda, en iyi parçacık, küresel en iyi parçacık olarak kaydedilir.

Sonraki iterasyonlarda, sürünün parçacıkları için olasılıklar PSO hızı ve konum denklemleri kullanılarak güncellenir. Teslim tarihi atama ve sevk kuralları ve atanmış rota değerlerine göre mevcut parçacık için uygunluk değeri hesaplanır. Eğer uygunluk değeri önceki iterasyondaki mevcut parçanın uygunluğundan daha iyiye, parçacığın kişisel en iyi değeri (p_{best}) güncellenir ve parçacık p_{best} parçacığı olarak kaydedilir. Daha sonra aynı işlem tüm parçacıklar için yapılır, diğerlerinden daha iyi bir uygunluk değerine sahip bir parçacık olup olmadığı incelenir. Daha iyi bir uygunluk değeri elde edilirse, en iyi parçacıklar güncellenir ve ilgili parçacık küresel en iyi parçacık olarak kaydedilir. Önerilen ICPSO'nun yalancı kodları Algoritma 1'de verilmiştir.

5.4.1. Modifiye edilmiş ICPSO (GA/ICPSO)

Bu algoritmada, klasik ICPSO'ya mutasyon ve çaprazlama operatörleri eklenmiştir. ICPSO ile en iyi olasılıkların yakalanmasına çalışılırken local minimuma takıldığından dolayı algoritmada modifikasyon ön görülmüştür. ICPSO V ve X vektörlerini kullanarak her iterasyonda problem için bir çözüm değeri oluşturur. Eğer daha iyi bir sonuç bulunamazsa, mevcut çözüm, en iyi personel ve en iyi küresel parçacık arasında GA'daki ile benzer bir mutasyon ve çaprazlama operatörleri uygulanır. Eğer literatürde belirtilen şekliyle ICPSO uygulanırsa, belli sayıda iterasyondan sonra iyileşme çok yavaş meydana gelmektedir. Bu noktada bazı parçacıklara mutasyon ve $pbest$ ile $gbest$ arasında çaprazlama uygulamak çözümü daha iyi hale getirmektedir. Mutasyon işleminde, seçilen noktalardaki değer, atalet(inertia) olarak belirlenen en iyi kişisel değerlerle değiştirilir. İlk çaprazlama işleminde, biliş(cognition) işlemi olarak mevcut parçacık, en iyi personel parçacık ile değerlerini değiştirir. İkinci çaprazlama işleminde sosyal (social component) unsur olarak, mevcut parçacık ile en iyi küresel parçacık, değerlerini değiştirir. Bu adımlar, iterasyon boyutuna ulaşıldığında durdurulur.

Apply the same initial steps with ICPSO and change steps after If $\mathcal{F}_{p_{current}} \leq \mathcal{F}_{p_{best}}$ line with below lines.

If $\mathcal{F}_{p_{current}} > \mathcal{F}_{p_{best}}$:

Apply mutation on the $p_{current}$ solution (the moment of inertia)

Perform a multi-point crossover between the p_{best} and the $p_{current}$ (cognitive component)

Perform a multi-point crossover between the g_{best} and the $p_{current}$ (social component)

Calculate fitness for $\mathcal{F}_{p_{current}}$

If $\mathcal{F}_{p_{current}} < \mathcal{F}_{p_{best}}$:

$p_{pbest} = p_{current}$

End if

If $\mathcal{F}_{p_{current}} < \mathcal{F}_{g_{best}}$:

$p_{gbest} = p_{current}$

End if

End if

Şekil 5.11. MICPSO yalancı kodu

5.4.2. Modifiye edilmiş kesikli PSO (Modified Discrete PSO - MDPSO)

Burada, Modifiye edilmiş kesikli PSO (MDPSO) [161], ICPSO'da kullanılan olasılık yapısının iyileştirilmesine yönelik problemi çözmek için uyarlanmıştır. Taşgetiren vd. [161] çalışmasında parçacıkların her geni için çaprazlama ve mutasyon olasılığı vermiştir. Örneğin, [161] no'lu referansta verilen çalışmaya göre her gen % 25 olasılıkla mutasyona uğrar ve daha sonra her gen tekrar % 25 olasılıkla p_{best} ile ve son olarak da % 25 olasılık ile ilişkili g_{best} ile çaprazlamaya uğrar.

Bizim MDPSO algoritmamızda, en büyük atölyede rastgele 50 gen seçerek parçacığın % 25'ini oluşturulur. Rastgele seçilen 50 gen, mutasyona uğrar ve başka bir olası değere değiştirilir (Atalet Momenti uygulandı). Ondan sonra tekrar rastgele 50 gen seçeriz ve bu genler ilişkili p_best değerlerine dönüşür (Bilişsel kısım uygulanır). Böylece, her parçacık ve ilişkili p_best değerleri çaprazlanmış olur. Son olarak, yine 50 gen seçeriz ve bu genlerin değerini, ilişkili g_best değerleriyle çaprazlarız (Sosyal kısım uygulanır). Böylece, her parçacık ile en iyi parçacık arasında çaprazlama uygulanmış olur. Bu durum da MDPSO'nun uygulanmasını daha pratik hale getirir.

Bir yinelemede ICPSO tarafından üretilen yeni çözüm değerleri kişisel en iyi olasılık değerlerinden daha iyi değilse, mevcut çözümün % 25 geninde atalet momenti olarak bir mutasyon işlemi uygulanır. Daha sonra, bilişsel bileşen olarak mevcut çözüm ile en iyi kişisel çözüm arasında % 25 gen sayısına çaprazlama işlemi uygulanır. Son olarak, mevcut çözüm ile sosyal bileşen olarak küresel en iyi çözüm arasında % 25 gen sayısına başka bir çaprazlama operasyonu uygulanarak algoritma iterasyon sayısı kadar tekrarlanır.

Apply the same initial steps with ICPSO and change steps after If $\mathcal{F}_{p_{current}} \leq \mathcal{F}_{p_{best}}$: line with below lines.

If $\mathcal{F}_{p_{current}} > \mathcal{F}_{p_{best}}$:

Apply mutation operation for 25% genes on $p_{current}$ (the moment of inertia)

Apply crossover operation for 25% genes on $p_{current}$ and p_{pbest} (*cognitive component*)

Apply crossover operation for 25% genes on $p_{current}$ and p_{gbest} (*social component*)

Calculate $\mathcal{F}_{p_{current}}$

If $\mathcal{F}_{p_{current}} < \mathcal{F}_{p_{best}}$:

$p_{pbest} = p_{current}$

End if

If $\mathcal{F}_{p_{current}} < \mathcal{F}_{g_{best}}$:

$p_{gbest} = p_{current}$

End if

End if

Şekil 5.12. MDPSO yalancı kodu

5.5. Hibrit Algoritmalar (HA-GA/SA, GA/TA)

HA ile hem genetik algoritmanın hem de tavlama benzetimi ve tabu arama algoritmalarının iyi noktalarına yer verilerek daha iyi sonuçlar üretilmeye çalışılmıştır. HA geliştirilirken kullanılan en önemli parametre iterasyon sayısıdır. İterasyon sayılarındaki eşit hesaplamalar sayesinde algoritmaların daha doğru karşılaştırmaları yapılabilir.

GA global çözüm uzayında daha iyi bir paralel arama yapma kapasitesine sahiptir. GA'nın daha önceki popülasyonlardan aldığı kullanışlı bilgiler bir sonraki popülasyonlara aktarılır. GA adımları basit ve güçlüdür. TA ve SA ise sadece bir çözüm üzerinde değerlendirme yapar. Bu nedenle global arama performansları GA'ya göre daha düşük kalır. Yapılan çalışmada öncelikle GA iterasyonları ardından üretilen en iyi çözüm SA ve TA için başlangıç çözümü olarak değerlendirilmiştir. Geliştirilen algorithmada öncelikle iterasyonlar GA'dan alınan çözüm ile başlar. GA tarafından üretilen ve GA iterasyon sayısının yarısı iterasyonundaki çözüm SA ve TA için başlangıç çözümü olarak alınır. SA ve TA algoritması belirlenen SA ve TA algoritması iterasyon sayısının yarısı kadar iterasyon ile devam eder. Eğer iterasyon sayısına ulaşırsa sonuç kaydedilir. Diğer prosedürler SA ve TA algoritmaları bölümünde anlatıldığı gibidir.

BÖLÜM 6. DENEYSEL SONUÇLAR VE KARŞILAŞTIRMALAR

Bu bölümde DIPPSDDA probleminin çözümüne ait GA, SA, TA ve GA/SA, GA/TA, ICPSO, MICPSO ve MDPSO algoritmaları tarafından üretilen çözüm sonuçları gösterilmiştir. Çalışmada öncelikle teslim tarihi belirleme ve sevk kurallarının olmadığı atölyeler ile sonuçlar elde edilmiş sonra elde edilen sonuçlar adım adım entegre edilen teslim tarihi ve sevk kuralları ile karşılaştırılmıştır. Bunun dışında ne yazık ki bu problemin çıktılarının başka problemler ve çözümler ile karşılaştırılma durumu yoktur. Çünkü literatürde bu probleme ilişkin standart girdi verileri bulunmamaktadır. Bu nedenle bu bölümde değişik meta sezgisel yaklaşımlar kullanılarak elde edilen sonuçlar ve değişik entegrasyon seviyeleri için karşılaştırmalar yapılmıştır.

6.1. Geliştirilen Bilgisayar Programı ve Kullanılan Kütüphaneler

DIPPSDDA problemini çözmek için Python programlama dilinde bir bilgisayar programı, Intel i5-6200U 2,30 GHz ve 8 GB RAM bellek olan bir bilgisayarda Jupyter/IPython IDE kullanarak geliştirilmiştir [162]. Geliştirilen program için kullanılan paket listesi: NumPy, pandas, copy, math, salabim, Matplotlib. NumPy kütüphanesi ile bilimsel hesaplamalar daha kolay halde yapılabilir. NumPy, kolay ve sezgisel çalışma yapmak için tasarlanmış hızlı, esnek ve birbirleriyle ilişkili veya alakalı veri yapıları sağlayan bir Python paketidir. Kısaca kullanılan paketlerden bahsetmek gerekirse: NumPy Python'da gerçek veri analizi yapmak için temel ve pratik bir üst düzey yapı oluşturmayı hedeflemektedir. Bu tez çalışmasında kullanılan diziler, matrisler ve olasılık hesaplamaları NumPy kütüphanesi kullanılarak hazırlanmıştır [163]. Pandas kütüphanesi veri analizleri ve veri bilimi için önemli kolaylıklar sağlamaktadır. Bu çalışmada verilerin tablolar halinde tutulması ve analizi

için pandas kütüphanesinden yararlanılmıştır [164]. Matplotlib platformlar arasında çeşitli sabit görüntüler ve interaktif ortamlar oluşturmaya yarayan kaliteli bir Python kütüphanesidir. Matplotlib ile Python betikleri, Python ve IPython kabukları, Jupyter Notebook, Web uygulama sunucuları ve dört grafik kullanıcı arabirimi olan Toolkits kullanılabilir. Matplotlib kütüphanesi ile tezdeki grafiklerin hazırlanması ve verilerin analizi çalışmaları yapılmıştır [165]. Tezin önemli kısımlarından olan simülasyon çalışması için salabim isimli kütüphane kullanılmıştır [166]. Salabim ile ayrıca geliştirilen animasyon daha önceki bölümde verilmiştir.

6.2. Entegrasyon Seviyeleri

Bu tez çalışmasında öncelikle, teslim tarihi belirleme ve sevk kuralları kullanmanın yararlı noktaları, 4 farklı düzeyde entegrasyon düzeyiyle belirlenmeye çalışılmıştır. Geliştirilen modelin verimliliğini ve kullanılabilir olduğunu test etmek için düşünülen entegrasyonların ilk düzeyinde, SIRO sevk kuralı ve RDM teslim tarih kuralı test edilmiştir. Yani, birinci entegrasyon seviyesinde hiçbir kural uygulanmamıştır. Entegrasyonun ikinci düzeyinde, sevk kuralları probleme dahil edilmiştir. Teslim tarihi tarih kuralı hala RDM olarak bırakılmıştır ve yalnızca sevk kuralını belirlemenin yararları denenmiştir. Entegrasyonun üçüncü düzeyinde, gönderme kuralı SIRO olarak bırakılmıştı ve son tarih kuralları uygulandı. Entegrasyonun dördüncü düzeyinde, sevk kurallarının ve teslim tarihi belirleme kurallarının birlikte çalıştırılması ile problemin performansında iyileşme olduğu gözlemlenmiştir.

6.2.1. Birinci entegrasyon seviyesi (SIRO-RDM)

Birinci entegrasyon seviyesi herhangi bir kuralın uygulanmadığı seviyedir. Bu seviyede teslim tarihi belirleme SIRO, sevk kuralı RDM olarak tercih edilir. Yani iki kural da rastgele sonuçlar üretecek şekilde yapılandırılır. Böylece teslim tarihi ve sevk kurallarının problem için meydana getireceği iyileştirmeler gösterilmiştir. Birinci entegrasyon seviyesinde birinci gen yani teslim tarihi belirleme kuralı geni ile ikinci

gen yani sevk kuralı geni sabit olarak ele alınmıştır. Her bir iterasyonda bu iki gen mutasyon, çaprazlama ve komşuluk aramaya girmez. Birinci entegrasyon seviyesi için üretilen ikinci atölyeye ilişkin son popülasyon tabloda verilmiştir. Tüm entegrasyon gösterimlerinde ikinci atölye verileri kullanılmıştır. Tüm atölyeler için entegrasyon karşılaştırmaları Ek-2’de verilmiştir. İkinci atölyede 50 iş, 10 operasyon, 15 makine ve 5 rota bulunmaktadır. Genetik algoritmada 400 iterasyon yapılarak sonuç elde edilmiştir. Tablo 6.1.’de teslim tarihi ve sevk kuralları için sabit 0 değeri kullanılmış olup bu değer tüm iterasyonlarda aynı olarak kalmıştır. 0 değeri teslim tarihi ve sevk kurallarının rastgele olduğu kurallardır.

Tablo 6.1. Birinci entegrasyon seviyesi için oluşturulan en iyi popülasyon

Population # 400 Best: 344.466 Avg: 345.64, Worst: 346.02	
Individual # 0 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 1, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 344.47
Individual # 1 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 3, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.54
Individual # 2 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 3, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 3, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.64
Individual # 3 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 3, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 1, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 3, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.65
Individual # 4 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 1, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 3, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.65
Individual # 5 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 3, 3, 2, 1, 3, 1, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.8
Individual # 6 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 1, 4, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.82
Individual # 7 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 1, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 1, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.85
Individual # 8 :	[0, 0, 4, 2, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 4, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 1, 3, 2, 1, 3, 1, 1, 4, 1, 0, 1, 2, 0] Fitness: 345.98
Individual # 9 :	[0, 0, 4, 2, 0, 0, 0, 0, 1, 1, 2, 2, 2, 4, 4, 2, 1, 2, 3, 2, 4, 3, 4, 3, 2, 0, 1, 2, 1, 1, 3, 1, 1, 4, 3, 0, 3, 2, 0, 4, 1, 3, 2, 1, 3, 0, 1, 4, 1, 0, 1, 2, 0] Fitness: 346.02
Execution Time: 00:21:02	

6.2.2. İkinci entegrasyon seviyesi (DSP-RDM)

İkinci entegrasyon seviyesinde ise teslim tarihi belirleme kuralı (RDM) rassal halde bırakılmış sevk kuralları(DSP) ise çalıştırılmıştır. Bu entegrasyon seviyesinde izlenmek istenen amaç fonksiyonundaki iyileşmelerdir. Tablo 6.2.’de birinci atölye için 400 iterasyon sonucunda üretilen popülasyon görülmektedir. Tabloya göre bu

entegrasyon seviyesinin birinci entegrasyon seviyesinden daha iyi sonuçlar verdiği söylenebilir.

Tablo 6.2. İkinci entegrasyon seviyesi için oluşturulan en iyi popülasyon

Population # 400 Best: 343.905 Avg: 344.21, Worst: 344.58	
Individual # 0 :	[0, 12, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 343.9
Individual # 1 :	[0, 0, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 0, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.01
Individual # 2 :	[0, 15, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 4, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.07
Individual # 3 :	[0, 7, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 4, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.08
Individual # 4 :	[0, 4, 4, 3, 0, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.09
Individual # 5 :	[0, 16, 4, 3, 0, 2, 0, 3, 0, 2, 3, 3, 0, 4, 0, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.19
Individual # 6 :	[0, 12, 4, 3, 0, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 4, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 1, 4, 1, 2, 1, 0] Fitness: 344.27
Individual # 7 :	[0, 6, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 2, 4, 1, 2, 1, 0] Fitness: 344.4
Individual # 8 :	[0, 13, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 0, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 1, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 2, 4, 1, 2, 1, 0] Fitness: 344.53
Individual # 9 :	[0, 2, 4, 3, 2, 2, 0, 3, 0, 2, 3, 3, 0, 4, 1, 3, 3, 3, 2, 0, 3, 3, 3, 2, 0, 4, 1, 1, 2, 4, 3, 1, 4, 0, 4, 3, 2, 1, 0, 3, 3, 1, 4, 3, 0, 1, 2, 4, 1, 2, 1, 0] Fitness: 344.58
Execution Time: 00:20:59	

6.2.3. Üçüncü entegrasyon seviyesi (SIRO-DD)

Üçüncü entegrasyon seviyesinde ikinci entegrasyon seviyesi ile benzer şekilde sadece sevk kuralı rassal (SIRO) bırakılmış, teslim tarihi belirleme fonksiyonu (DD) ise çalıştırılmıştır. Tablo 6.3.'ten ve bu entegrasyon seviyesinden çıkan sonuç yine entegrasyonun hiç entegrasyon olmadığı haldekinden daha iyi olduğu sonucudur.

Tablo 6.3. Üçüncü entegrasyon seviyesi için oluşturulan en iyi popülasyon

Population # 400 Best: 296.44 Avg: 375.81, Worst: 393.54	
Individual # 0 :	[11, 0, 0, 4, 3, 1, 0, 2, 4, 1, 1, 2, 3, 3, 0, 0, 4, 3, 4, 0, 0, 0, 1, 0, 3, 4, 3, 3, 0, 1, 1, 4, 0, 1, 0, 1, 4, 2, 1, 1, 2, 0, 3, 1, 0, 0, 1, 3, 2, 0, 0, 1] Fitness: 296.44
Individual # 1 :	[3, 0, 0, 3, 3, 2, 1, 4, 3, 3, 1, 4, 1, 4, 3, 2, 1, 2, 3, 4, 1, 4, 3, 2, 0, 2, 1, 3, 0, 2, 3, 1, 1, 4, 1, 3, 3, 2, 0, 0, 3, 3, 3, 1, 0, 4, 1, 2, 0, 1, 3, 3] Fitness: 360.27
Individual # 2 :	[10, 0, 4, 2, 0, 3, 1, 0, 1, 1, 1, 2, 1, 3, 3, 4, 0, 2, 2, 0, 0, 2, 4, 0, 4, 0, 1, 3, 1, 1, 0, 4, 1, 2, 4, 0, 1, 0, 1, 4, 0, 2, 1, 4, 3, 3, 2, 3, 1, 0, 3, 4] Fitness: 371.11
Individual # 3 :	[10, 0, 3, 2, 1, 0, 0, 2, 0, 4, 3, 0, 2, 1, 0, 4, 0, 3, 1, 1, 2, 2, 2, 4, 0, 4, 0, 2, 3, 0, 1, 2, 0, 1, 3, 3, 2, 4, 1, 1, 0, 2, 1, 3, 0, 3, 4, 0, 1, 0, 3, 1] Fitness: 371.71
Individual # 4 :	[0, 0, 1, 0, 1, 0, 1, 0, 1, 4, 3, 2, 1, 3, 3, 4, 0, 3, 2, 0, 3, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 1, 3, 1, 3, 2, 3, 1, 3, 3, 4] Fitness: 392.46
Individual # 5 :	[0, 0, 1, 0, 1, 0, 1, 0, 1, 4, 3, 2, 1, 3, 3, 4, 0, 3, 2, 0, 3, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 1, 3, 1, 3, 2, 3, 1, 2, 3, 4] Fitness: 392.78
Individual # 6 :	[0, 0, 1, 0, 1, 0, 1, 4, 1, 4, 3, 2, 3, 3, 3, 4, 0, 3, 2, 0, 4, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 1, 3, 1, 3, 2, 3, 1, 2, 3, 4] Fitness: 393.05
Individual # 7 :	[0, 0, 1, 0, 1, 0, 1, 0, 1, 4, 3, 2, 1, 3, 3, 4, 0, 3, 2, 0, 3, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 0, 3, 1, 3, 2, 0, 1, 2, 3, 4] Fitness: 393.29
Individual # 8 :	[0, 0, 1, 0, 1, 0, 1, 0, 1, 4, 3, 2, 1, 3, 3, 4, 0, 3, 2, 0, 3, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 0, 3, 1, 3, 2, 3, 1, 2, 3, 4] Fitness: 393.43
Individual # 9 :	[0, 0, 1, 0, 1, 0, 1, 4, 1, 4, 3, 2, 1, 3, 3, 4, 0, 3, 2, 0, 3, 3, 3, 2, 3, 2, 1, 3, 0, 2, 3, 1, 2, 4, 1, 3, 1, 0, 1, 4, 0, 2, 1, 3, 1, 3, 2, 3, 1, 3, 3, 4] Fitness: 393.54
Execution Time: 00:21:05	

6.2.4. Dördüncü entegrasyon seviyesi (DSP-DD)

Son entegrasyon seviyesinde tam entegrasyon denenmiştir. Yani hem teslim tarihi belirleme hem de sevk kuralları tamamen çalıştırılmış ve sonuçlar değerlendirilmiştir. Tablo 6.4.'ten de okunabilecek sonuçlara göre tam entegrasyon adım adım entegrasyonlardan daha iyi sonuç verdiği görülebilir. Buna göre yapılan entegrasyon çalışmalarının faydaları gösterilmiş olur.

Tablo 6.4. Dördüncü entegrasyon seviyesi için oluşturulan en iyi popülasyon

Population # 400 Best: 270.924 Avg: 271.29, Worst: 271.69	
Individual #0	[8, 1, 4, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 270.92
Individual #1	[8, 3, 4, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 270.92
Individual #2	[8, 6, 4, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 270.92
Individual #3	[8, 1, 3, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.38
Individual #4	[8, 6, 3, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.38
Individual #5	[8, 3, 3, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.38
Individual #6	[8, 1, 1, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.42
Individual #7	[8, 6, 1, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.42
Individual #8	[8, 3, 1, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 2, 3, 3] Fitness: 271.42
Individual #9	[8, 1, 3, 3, 2, 3, 1, 0, 3, 4, 4, 1, 0, 0, 0, 4, 0, 1, 1, 2, 4, 3, 1, 4, 1, 3, 3, 4, 3, 3, 2, 1, 3, 0, 0, 3, 2, 3, 2, 0, 4, 3, 4, 2, 3, 1, 4, 2, 3, 0, 3, 3] Fitness: 271.69

Genetik algoritmalar ile yapılan 400 iterasyon sonunda ikili entegrasyonun en iyi, 270.924 ortalama 271.29 ve en kötü 271.69 sonuçlarına ulaştığı görülür. Entegrasyonun olmadığı durumlarda en iyi 344.466, ortalama 345.64 ve en kötü 346.02 değerleri elde edilmişti. Buradaki farklar anlamlı derecede entegrasyonun faydalarını ortaya koymaktadır. Bu testlere göre, en iyi entegrasyon seviyesi tam entegrasyon sonucudur. Sadece sekizinci atölye için çözülen değişik entegrasyon seviyesi sonuçları Tablo 6.5.'te gösterilmiştir. Tam entegrasyon düzeyinin, Tablo 6.5.'te gösterildiği gibi en iyi performansa sahip olduğu gözlenmiştir.

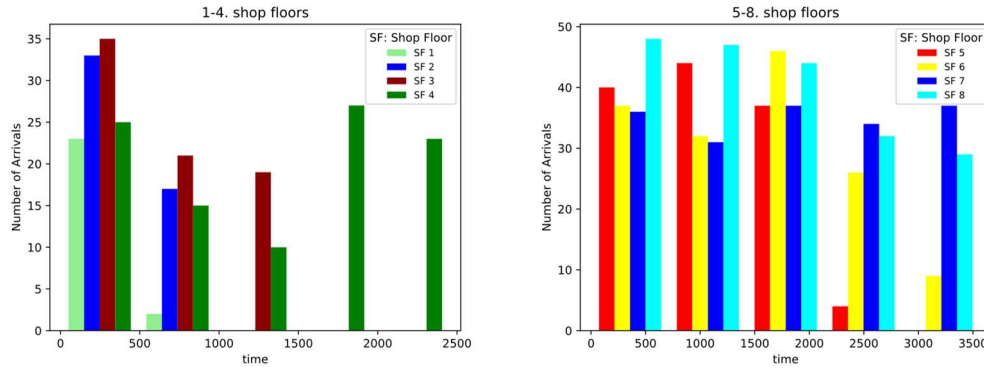
Tablo 6.5. Sekizinci atölyeye ait entegrasyon sonuçları

Algorithm	Atölye	Entegrasyon Düzeyi	En iyi	Ortalama	En kötü	CPU zamanları
ord	8	SIRO-RDM	1647,953	1665,6	1682,18	500
vns	8	SIRO-RDM	1670,781	1670,788	1672,396	330
sa	8	SIRO-RDM	1662,963	1665,285	1667,407	336
ta	8	SIRO-RDM	1540,97	1583,978	1662,53	333
ga	8	SIRO-RDM	1576,268	1578,43	1579,66	698
ha	8	SIRO-RDM	1596,393	1598,752	1600,792	683
ord	8	DSP-RDM	1641,615	1666,03	1695,3	500
vns	8	DSP-RDM	1524,838	1572,95	1665,591	325
sa	8	DSP-RDM	1648,103	1664,702	1683,021	335
ta	8	DSP-RDM	1520,669	1568,201	1654,376	332
ga	8	DSP-RDM	1583,77	1585,29	1586,08	698
ha	8	DSP-RDM	1592,137	1633,138	1666,773	698
ord	8	SIRO-DD	1135,059	8700,08	20388,91	500
vns	8	SIRO-DD	1647,398	1647,424	1651,048	168
sa	8	SIRO-DD	1641,088	1654,843	1674,169	171
ta	8	SIRO-DD	1635,279	1636,651	1649,231	202
ga	8	SIRO-DD	1135,059	1530,61	1623,52	350
ha	8	SIRO-DD	1643,088	1654,901	1667,067	295
ord	8	DSP-DD	1164,778	13141,17	60112,32	500
vns	8	DSP-DD	1012,305	1039,747	1122,021	341
sa	8	DSP-DD	1378,499	10455,41	20391,21	325
ta	8	DSP-DD	1104,217	9704,087	33388,61	315
ga	8	DSP-DD	1054,257	1054,51	1054,82	739
ha	8	DSP-DD	1054,057	1265,762	1433,251	563

Ord: Ordinary, vns: Variable neighbourhood search, ta: taboo algorithm, sa: simulated annealing, ga: genetic algorithm, ha: hybrid algorithms

6.3. Atölyelere Ait Sonuçlar

Dinamik atölyelerde işlerin zaman içerisinde stokastik olarak atölyeye geldiğinden bahsetmiştik. Şekil 6.1.'de histogram grafiğinde işlerin hangi atölyeye hangi zamanlarda geldiği gösterilmektedir.



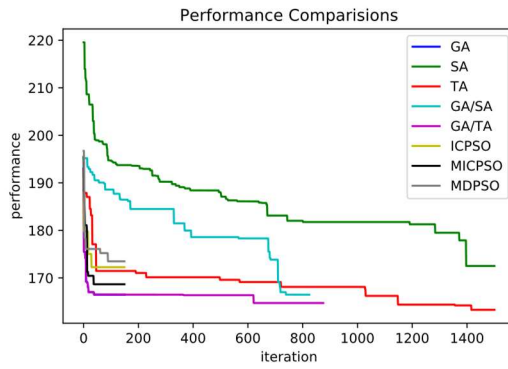
Şekil 6.1. İşlerin atölyelere geliş zamanları

Şekil 6.2.-6.9.'da algoritmalarla ait en iyi, ortalama ve en kötü değerleri gösterilmektedir. Her bir atölye için algoritmalar çalıştırılmış ve en iyi sonuçlar kaydedilerek grafiklerde verilmiştir. Algoritmaların daha iyi karşılaştırılabilmesi için iterasyon sayıları farklı değerlendirilmiştir. GA ve PSO'nun popülasyon tabanlı çalışmaları nedeniyle genetik algortmada her bir iterasyonda 10 adet birey üretilir. Ancak bireyler üzerinde çalışan tavlama benzetimi ve tabu arama algoritmalarında her iterasyonda sadece 1 yeni çözüm üretilir. Bu nedenle birey tabanlı çalışan algoritmaların daha fazla iterasyon ile çalıştırılmasının daha doğru olacağı düşünülmüştür. Tavlama benzetimi ve tabu arama algoritmalarına daha genetik algortmaya göre 10 kat daha fazla iterasyon sayısı verilmiştir. Hibrit algortmada ise öncelikle geniş uzayda daha etkili arama yapabilen genetik algortmalar çalıştırılmıştır. Ardından genetik algortmadan alınan en iyi birey ile tavlama benzetimi ve tabu arama algortması devam ettirilmiştir. Hibrit arama için önce genetik algortmanın yarısı kadar iterasyonla genetik algortma, yarısı kadar iterasyonla da tavlama benzetimi veya tabu arama algortması çalıştırılmıştır. Böylece daha iyi karşılaştırma yapılabilecek bir yapıya ulaşılmıştır.

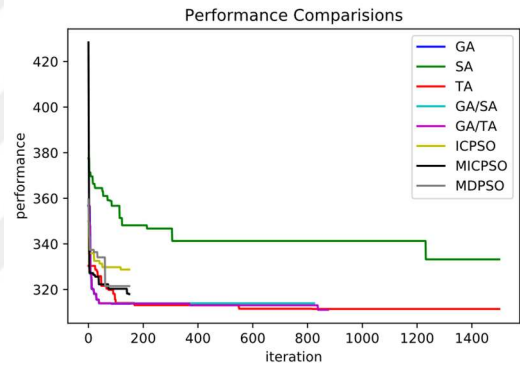
Elde edilen sonuçlara göre geliştirilen TA ve GA/TA algoritmalarının genellikle daha iyi sonuç verdiği gözlemlenebilir. Problemin yapısına uygun olarak en iyi sonucun kullanılarak bir sonraki adımda Gantt şeması çizilmesi çalışmanın sonuçlarından birisidir. Bu nedenle en iyi sonuç, ortalama ve en kötü sonuca göre önceliklidir. Birinci atölyede 25 adet iş ve 10 adet makine vardır. Diğer atölyelere göre küçük olan birinci

atölyedeki sonuçlar yanıltıcı olabilir. En karmaşık yapıya sahip olan yedinci ve sekizinci atölyede MICPSO algoritması en iyi sonucu vermiştir.

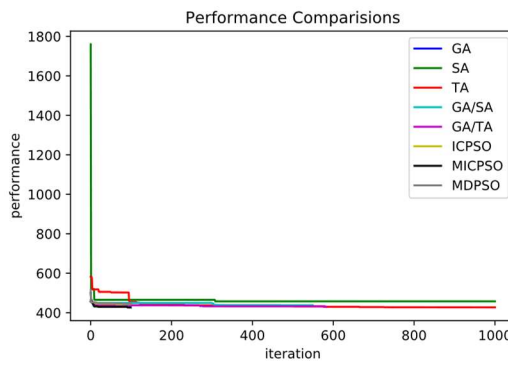
Birinci atölyede 25 adet iş, 10 adet operasyon, 5 adet makine ve 5 adet rota bulunmaktadır. Birinci atölye için yapılacak olan iterasyon sayısı da GA ve PSO için 150 olarak belirlenmiştir. TA ve SA için 1500 iterasyon denenmiştir. Hibrit algoritmalar için ise 75 adet GA ve 750 adet TA veya SA iterasyonları olmak üzere toplam 825 iterasyon denenmiştir. Deneyler sonucunda en iyi sonucu veren algoritmanın 163.28 uygunluk değeri ile TA olduğu sonucuna ulaşılmıştır.



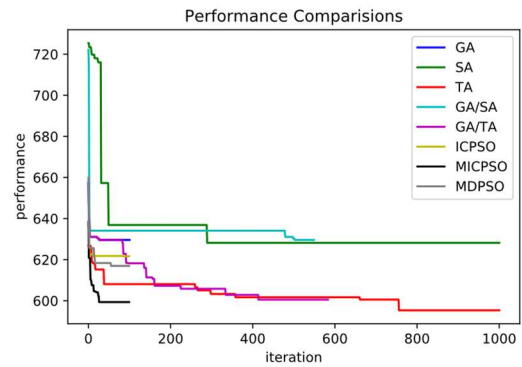
Şekil 6.2. Birinci atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



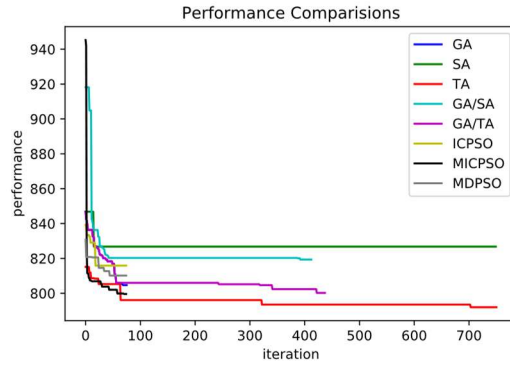
Şekil 6.3. İkinci atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



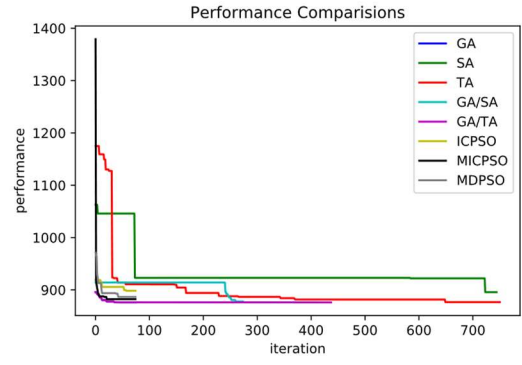
Şekil 6.4. Üçüncü atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



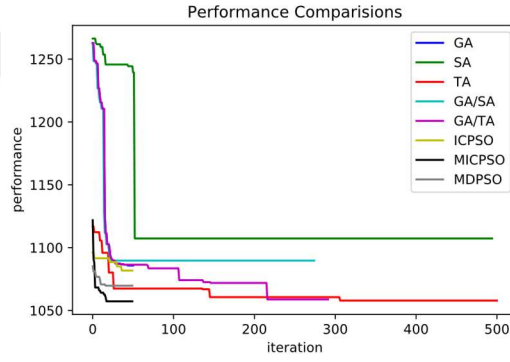
Şekil 6.5. Dördüncü atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



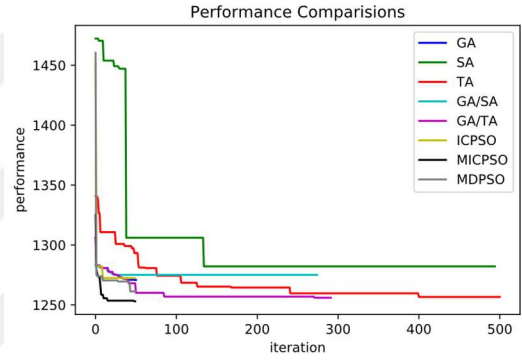
Şekil 6.6. Beşinci atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



Şekil 6.7. Altıncı atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



Şekil 6.8. Yedinci atölyedeki tüm algoritmalarla ait iterasyon sonuçları.



Şekil 6.9. Sekizinci atölyedeki tüm algoritmalarla ait iterasyon sonuçları

İkinci atölyede 50 adet iş, 10 adet operasyon, 5 adet makine ve 5 adet rota bulunmaktadır. İkinci atölye için yapılacak olan iterasyon sayısı da birinci atölyedeki gibi belirlenmiştir. TA-SA ve GA/TA-GA/SA için uygulanacak iterasyon sayıları birinci atölyede olduğu gibi oranlanmıştır. Deneyler sonucunda en iyi sonucu veren algoritmanın 311,09 uygunluk değeri ile GA/TA olduğu sonucuna ulaşılmıştır.

Üçüncü atölyede 75 adet iş, 10 adet operasyon, 10 adet makine ve 5 adet rota bulunmaktadır. İkinci atölye için yapılacak olan iterasyon sayısı da GA için 100 olarak belirlenmiştir. TA ve SA için ise 1000 iterasyon denenmiştir. HA için ise, 50 adet GA iterasyonu 500 adet ayrı ayrı SA ve TA iterasyonu gerçekleştirilmiştir. Deneyler sonucunda en iyi sonucu veren algoritmanın 426,43 uygunluk değeri ile MICPSO olduğu sonucuna ulaşılmıştır.

Dördüncü atölyede 100 adet iş, 10 adet operasyon, 10 adet makine ve 5 adet rota bulunmaktadır. Dördüncü atölye için yapılacak olan iterasyon sayısı da üçüncü atölyedeki gibi belirlenmiştir. Deneyler sonucunda en iyi sonucu veren algoritmanın 595,30 uygunluk değeri ile TA olduğu sonucuna ulaşılmıştır.

Beşinci atölyede 125 adet iş, 10 adet operasyon, 10 adet makine ve 5 adet rota bulunmaktadır. Atölye için yapılacak olan iterasyon sayısı da GA ve PSO için 75 olarak belirlenmiştir. TA ve SA için ise 750 iterasyon denenmiştir. Deneyler sonucunda en iyi sonucu veren algoritmanın 791,98 uygunluk değeri ile TA olduğu sonucuna ulaşılmıştır.

Altıncı atölyede 150 adet iş, 10 adet operasyon, 25 adet makine ve 5 adet rota bulunmaktadır. Altıncı atölyede beşinci atölyedeki ile aynı iterasyon sayıları kullanılmıştır. Deneyler sonucunda en iyi sonucu veren algoritmanın 876,11 uygunluk değeri ile GA olduğu sonucuna ulaşılmıştır.

Yedinci atölyede 150 adet iş, 10 adet operasyon, 35 adet makine ve 5 adet rota bulunmaktadır. Atölye için yapılacak olan iterasyon sayısı da GA için 50 olarak belirlenmiştir. TA ve SA için ise 500 iterasyon denenmiştir. Deneyler sonucunda en iyi sonucu veren algoritmanın 1057,22 uygunluk değeri ile MICPSO olduğu sonucuna ulaşılmıştır.

Sekizinci atölyede 200 adet iş, 10 adet operasyon, 40 adet makine ve 5 adet rota bulunmaktadır. Sekizinci atölyede yedinci atölyedeki aynı iterasyon sayıları kullanılmıştır. Deneyler sonucunda en iyi sonucu veren algoritmanın 1252,97 uygunluk değeri ile MICPSO olduğu sonucuna ulaşılmıştır. Toplu sonuçlar Tablo 6.6.'da verilmiştir.

Tablo 6.6. Atölyelere ait algoritma sonuçları

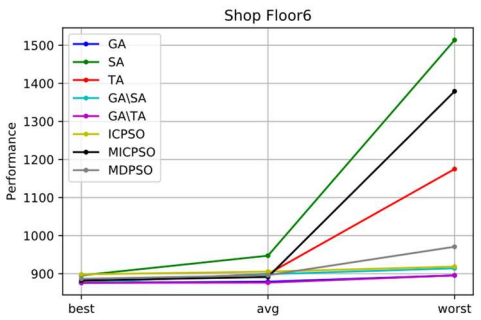
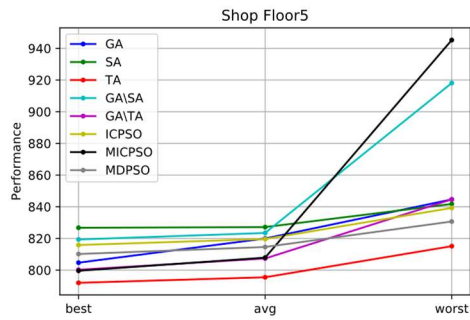
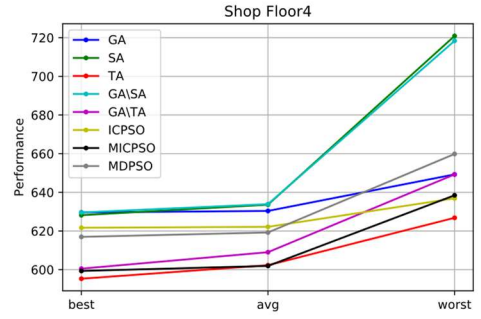
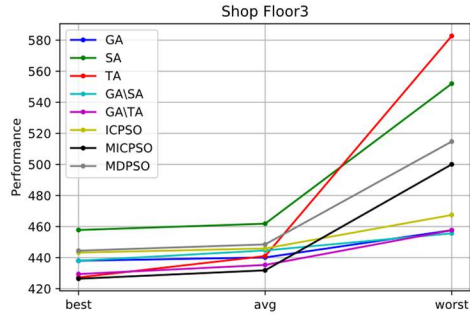
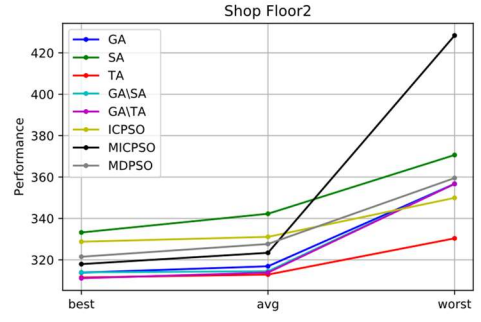
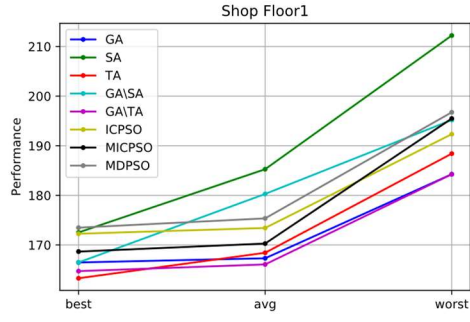
Atölye	Algoritma	En İyi	Ortalama	En Kötü
1	GA	166,475	167,3164	184,254
	SA	172,521	185,2701	212,2258
	TA	163,289	168,4264	188,41
	GA/SA	166,475	180,2847	195,192
	GA/TA	164,727	166,0796	184,254
	ICPSO	172,264	173,4218	192,331
	MICPSO	168,666	170,2855	195,488
	MDPSO	173,49	175,3604	196,749
2	GA	313,83	316,8764	356,699
	SA	333,217	342,2511	370,6496
	TA	311,464	312,8707	330,376
	GA/SA	313,96	314,4527	356,699
	GA/TA	311,091	313,831	356,699
	ICPSO	328,765	331,0876	349,942
	MICPSO	317,936	323,3914	428,403
	MDPSO	321,49	327,6459	359,509
3	GA	438,007	440,1038	457,6415
	SA	457,79	461,8413	552,0082
	TA	427,267	440,9922	582,719
	GA/SA	438,007	444,6151	455,6165
	GA/TA	429,44	435,2815	457,6415
	ICPSO	443,244	445,899	467,461
	MICPSO	426,43	431,8284	500,063
	MDPSO	444,479	448,5154	514,752
4	GA	629,593	630,3693	649,2625
	SA	628,165	633,5943	720,903
	TA	595,308	602,298	626,815
	GA/SA	629,593	633,8883	718,3735
	GA/TA	600,511	609,0249	649,2625
	ICPSO	621,665	622,1453	636,894
	MICPSO	599,335	601,926	638,465
	MDPSO	616,956	619,2189	659,843
5	GA	804,694	819,955	844,633
	SA	826,759	827,1593	841,7695
	TA	791,982	795,4791	815,122
	GA/SA	819,355	823,4518	918,126
	GA/TA	800,219	807,3426	844,633
	ICPSO	815,848	819,6427	839,248
	MICPSO	799,586	807,8806	945,263
	MDPSO	810,163	814,5955	830,692

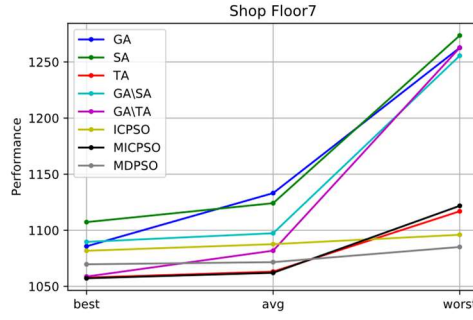
Tablo 6.6. (Devamı)

6	GA	876,11	879,136	895,637
	SA	895,637	947,348	1513,798
	TA	876,669	899,3633	1174,946
	GA/SA	876,248	898,8428	914,12
	GA/TA	876,11	876,6351	895,637
	ICPSO	898,304	905,57	918,741
	MICPSO	882,226	891,4095	1379,107
	MDPSO	886,583	896,0142	970,818
7	GA	1085,723	1133,128	1262,708
	SA	1107,268	1124,14	1273,625
	TA	1057,833	1063,134	1116,934
	GA/SA	1089,64	1097,34	1255,604
	GA/TA	1058,726	1081,815	1262,708
	ICPSO	1081,727	1087,62	1095,989
	MICPSO	1057,226	1062,082	1121,79
	MDPSO	1069,676	1071,594	1085,124
8	GA	1270,814	1276,444	1294,467
	SA	1282,144	1301,855	1467,114
	TA	1256,614	1267,453	1340,856
	GA/SA	1275,076	1275,49	1282,885
	GA/TA	1255,985	1260,485	1294,467
	ICPSO	1272,441	1277,589	1452,344
	MICPSO	1252,973	1257,878	1325,065
	MDPSO	1261,234	1273,238	1460,446

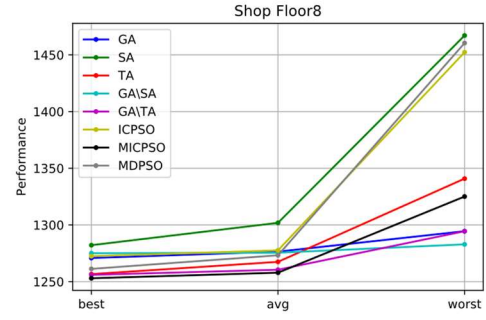
6.3.1. Geliştirilen algoritmalarla ait performans değerleri

Algoritmaların performans grafiklerinden sonra karşılaştırma yapabilmek için tüm algoritmalar için en iyi, en kötü ve ortalama uygunluk değerleri için ayrıca grafikler çalışmaya eklenmiştir. Buna göre en iyi sonuçlar arasındaki en iyi değere sahip olan algoritmanın en iyi sonuç verdiğini söyleyebiliriz. Şekilde tüm atölyelere ait algoritmaların ürettiği en iyi, en kötü ve ortalama performans değerlerinin grafikleri gösterilmiştir.





Şekil 6.16. Yedinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları

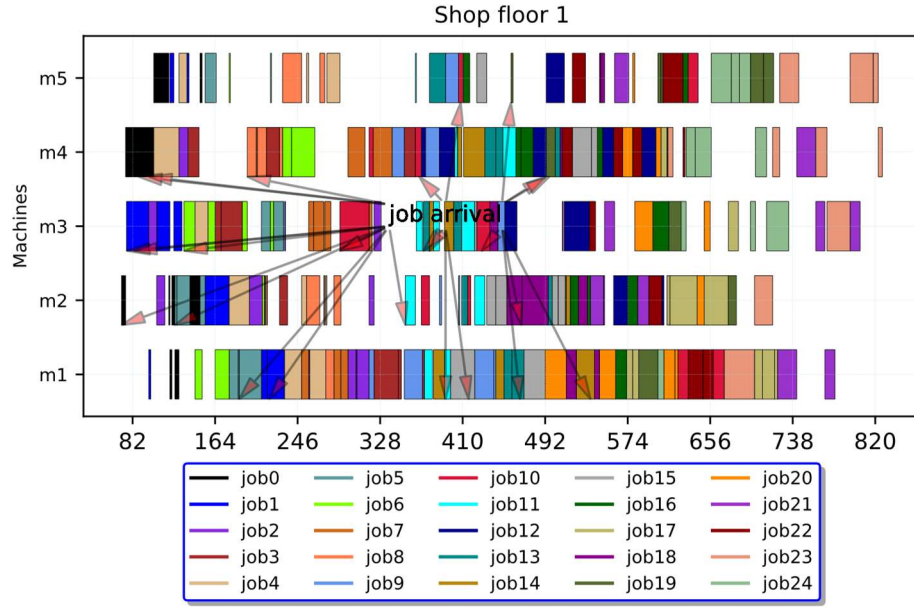


Şekil 6.17. Sekizinci atölyeye ait en iyi, ortalama ve en kötü çözümler için algoritma karşılaştırmaları

Elde edilen sonuçlara göre TA ve GA/TA algoritmalarının genel olarak daha iyi sonuçlar verdiği gözlemlenmiştir. Ayrıca, yedinci ve sekizinci yani en büyük 2 atölyede ise MICPSO'nun daha iyi sonuç verdiği görülmektedir. Ele alınan problemde büyük atölyelerde iyi sonuçlar vermesi nedeniyle MICPSO algoritmasının geliştirilmiş olması önem arz etmektedir. Çünkü gerçek atölyelerde çok daha fazla iş, makine, operasyon ve rota sayıları bulunmaktadır. Tezde ele alınan atölyelerden gerçek atölye problem verilerine en yakın olanlar büyük atölyelerdir.

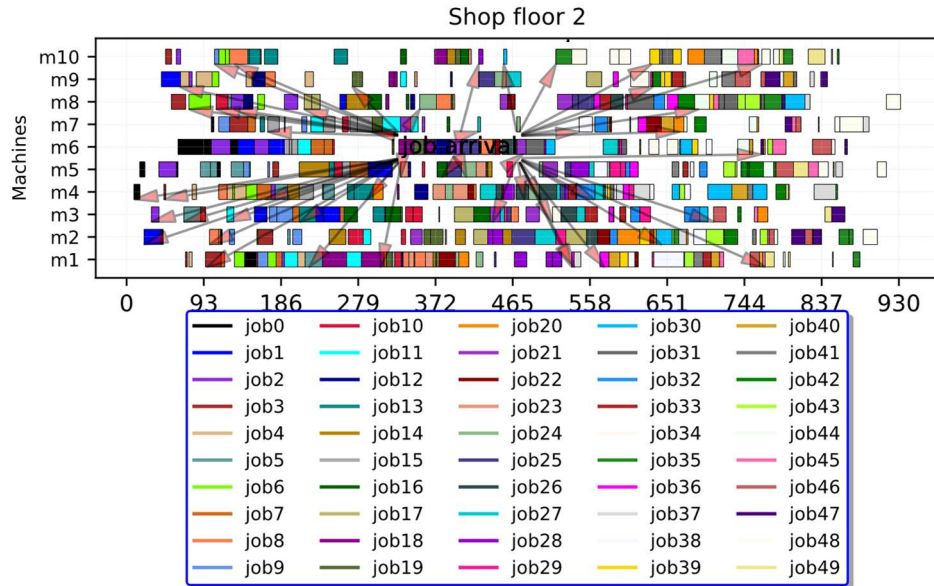
6.3.2. En iyi çözümlere ait Gantt şemaları

Çizelgeleme problemlerindeki çıktılar Gantt şeması ile gösterilebilir. Gantt şemasında hangi işin hangi makinede hangi sürelerde işlem göreceği bir şekil üzerinde gösterilir. DIPPSDDA probleminin analizleri sonucunda işlerin ne zaman atölyeye geldikleri ve atandıkları makineler de ayrıca Gantt şemasına eklenmiştir. Gantt şemasında her bir iş farklı bir renkle temsil edilmiştir. Ayrıca her bir iş parçası işe ait bir operasyonu göstermektedir. Operasyonlar arasında sıralama söz konusu olduğundan işe ait birinci dikdörtgen birinci operasyonu göstermektedir. Bir işin operasyonu o operasyondan önce yapılması gereken operasyonlar tamamlanmadan işlem görmeye başlayamaz. Gantt şemaları çizilirken algoritmalar sonucu üretilen en iyi çözümün bilgileri dikkate alınmıştır. Buna göre 1. Atölye için en iyi sonucu TA verdiği için Şekil 6.18.'deki çizim TA algoritmasından elde edilen çözüm kullanılarak yapılmıştır.



Şekil 6.18. TA'dan gelen en iyi çözüme ait Gantt şeması ile gösterilen birinci atölye çizelgesi

Şekil 6.19.'da ikinci atölyeye ilişkin Gantt şeması bulunmaktadır. İkinci atölye için en iyi sonucu GA/TA verdiğiinden dolayı Şekil 6.19.'daki çizim GA/TA algoritmasından elde edilen çözüm kullanılarak yapılmıştır



Şekil 6.19. GA/TA'dan gelen en iyi çözüme ait Gantt şeması ile gösterilen ikinci atölye çizelgesi

BÖLÜM 7. SONUÇLAR VE TARTIŞMA

Proses planlama ve çizelgeleme üzerine yapılan bütünleşik çalışmalara ilgi artarak devam etmektedir. Bu konudaki çalışmalar ile ilgili son yıllarda çok fazla sayıda yayın yapılmıştır ve yapılmaya devam etmektedir. Bu tez çalışmasında da ikili bütünleşik çalışmalara artı olarak üçlü üretim fonksiyonu bütünleşik çalışması ele alınmıştır. Çalışmada süreç planlama, çizelgeleme ve teslim tarihi belirleme üretim fonksiyonları bütünleşik hale getirilmiştir. Çalışmada ele alınan atölye işlerin gelişleri açısından dinamik bir yapıya sahiptir. Çizelgeleme yapılırken birden fazla olan rotalar dikkate alınarak çizelgelemeler yapılmıştır. Tezde ele alınan problem için DIPPSDDA kavramı kullanılmıştır. DIPPSDDA problemi genetik algoritmalar, tavlama benzetimi, tabu arama, parçacık sürü optimizasyonu ve hibrit arama gibi değişik meta-sezgisel algoritmalar kullanılarak çözülmüştür. Problem öncelikle genetik algoritmalar yapısında çözülmüştür. Probleme ilişkin çözüm değerleri genetik algortmada kullanılan kromozom yapısı ile gösterilmiştir. Ardından çözümler için diğer algoritmalar da denenerek en iyi sonuca ulaşılmaya çalışılmıştır.

Bu tez çalışmasının çıktılarından bir tanesi de teslim tarihi belirleme ve sevk kuralları ile tam entegrasyon halinde çalışmanın faydalarıdır. Tez için oluşturulan 8 adet farklı atölye için öncelikle herhangi bir kural olmadan çözümler oluşturulmuştur. Ardından adım adım kurallar eklenerek kurallar ile tam entegrasyon halde çalışmanın faydası gösterilmiştir. Buna göre hem teslim tarihi belirleme hem de sevk kurallarının birlikte çalışmasının daha iyi sonuçlar verdiği sonucundan hareketle algoritmalar geliştirilmeye başlanmıştır.

Tezde ayrıca genetik algortmada kullanılan mutasyon operatörü farklı şekillerde geliştirilerek bir komşu çözüm algoritması üretilmiştir. Komşu çözüm algoritması

tavlama benzetimi ve tabu arama algoritmaları için temel oluşturmaktadır. Komşu çözüm algoritması ile mevcut çözüme yakın ve mevcut çözümden daha iyi bir çözüm ile algoritma adımlarına eklenmiştir. Tavlama benzetimi bir sonraki adımda her zaman daha iyi çözümün seçilmesini garanti etmez. Yerel minimum noktalarından kaçmak için tavlama benzetiminde bazen daha kötü bireyler de çözüme dahil edilir. Ancak sonraki iterasyonlarda daha kötü bireylerin çözüme dahil edilmesi olasılığı azaltılır. Tabu arama algoritmasında literatürde genellikle kullanılan yapıdan farklı olarak her iterasyonda tabu listesinin belirli bir sayıda sabit tutulması sağlanmıştır. Daha sonra tavlama benzetimi ile genetik algoritmanın, tabu arama ile genetik algoritmanın, parçacık sürü optimizasyonu ile genetik algoritmanın bir arada çalıştığı hibrit algoritmalar ile problem çözülmüştür. Hibrit algoritma çözüm uzayını geniş tarayarak genetik algoritmalar ile çözüme başlar. Ardından genetik algoritmalar tarafından üretilen en iyi birey tavlama benzetimi algoritmasının başlangıç çözümünü oluşturur. Tavlama benzetimi algoritması ile devam ederek en iyi sonuca ulaşmaya çalışan hibrit algoritma için adil bir iterasyon sayısı belirlenerek sonuçların karşılaştırılması önceki bölümlerde verilmiştir.

Çalışmanın konularından birisi de kesikli olay zamanlı simülasyon çalışması olmuştur. Atölyede yeni bir olay olduğunda atölyenin belirli bilgileri güncellediği ve simülasyonuna devam ettiği bir sistem kurularak problemin dinamik yapıda olması sağlanmıştır. Geliştirilen simülasyon modeline göre, işler atölyeye geldiklerinde öncelikle işlere rota bilgileri, operasyon süreleri, ağırlık ataması yapılmaktadır. İşe ait teslim tarihi algoritma tarafından hesaplanarak simülasyon çalıştırılmıştır. Simülasyon sonucunda elde edilen işin atölyeden çıkış zamanı ile işin hesaplanan teslim tarihi arasındaki farktan yola çıkılarak amaç fonksiyonu hesaplanmıştır. Amaç fonksiyonu için hem erken tamamlanma hem geç tamamlanma hem de verilen teslim tarihinin süresi cezalandırılmıştır. Böylece atölyenin TZÜ felsefesi ile üretim yapması sağlanmaya çalışılmıştır.

Sonuç olarak bu tez çalışması sonunda beklenen amaç, dinamik ortamda çalışabilecek bir bütünleşik proses planlama, çizelgeleme ve teslim tarihi belirleme fonksiyonları

modelinin sezgisel yöntemler kullanılarak çözülebilmesini araştırmak ve analiz etmektir. Çalışmanın ana sonuçlarını aşağıdaki şekildeki gibi detaylandırabiliriz;

1. DIPPSDDA probleminin çözümüne yönelik model geliştirilmesi ve problem formülasyonunun belirlenmesi
2. Meta-sezgisel yöntemlerin DIPPSDDA problem için uygulanabilir olduğunun araştırılması ve problem için genetik algoritma yaklaşımlarının entegre edilmesi
3. Meta-sezgisel algoritmaların problem için bulduğu sonuçların örneğin Genetik Algoritma sonuçları ile kıyaslanması ve üstünlüklerinin belirlenmesi
4. DIPPSDDA probleminin literatüre kazandırılması ve problem ile ilgili farklı algoritmaların uygulanabilir olduğunun araştırılması

Aşağıda verilen üç çalışma, çalışmanın gelecekteki çalışmaları olarak verilebilir. Öncelikle, DIPPSDDA yapısı, operasyonların ve operasyon numaralarının esnekliğinin ya da süreçlerin ve sayıların esnekliğine entegre edilebilir. İkincisi, çalışma makine bozulması, işlerin iptal edilmesi, acil işlerin atölyeye gelmesi gibi dinamik olaylara da cevap verebilir hale getirilebilir. Son olarak, sorunun amaç fonksiyonu, erken tamamlanma, gecikme ve teslim tarihi belirleme cezalandırılmıştır. Buna ek olarak, her işin bekleme süreleri, makine dengesi oranları ya da maksimum tamamlanma zamanı gibi amaçlar da probleme ilave edilerek amaç programlama problemi çalışılabilir.

KAYNAKLAR

- [1] Philipoom, P. R., The choice of dispatching rules in a shop using internally set due-dates with quoted leadtime and tardiness costs, *International Journal of Production Research*, 38 (7), 1641-1655, 2000.
- [2] Rangsaritratamee, R., WG, F., MB, K., Dynamic rescheduling that simulataneously considers efficiency and stability, *Computer and Industrial Engineering*, 46, 1-15, 2004.
- [3] Demir, H. I., Taskin, H., Bütünleşik Proses Planlama, Çizelgeleme ve Teslim Tarihi Belirleme, Doktora Tezi, Sakarya Üniversitesi, Sakarya, 2005.
- [4] W. ElMaraghy, Integrated assembly planning and scheduling, 41, 11-14, 1992.
- [5] Yang, Y. N., Parsaei, H. R., Leep, H. R., Prototype of a feature-based multiple-alternative process planning system with scheduling verification, *Computers and Industrial Engineering*, 39 (1-2), 109-124, 2001.
- [6] Xu, X., Wang, L., Newman, S. T., Computer-aided process planning - A critical review of recent developments and future trends, *International Journal of Computer Integrated Manufacturing*, 24 (1), 1-31, 2011.
- [7] Morad, N., Zalzal, A. M. S., Genetic algorithms in integrated process planning and scheduling, *Journal of Intelligent Manufacturing*, 10 (2), 169–179, 1999.
- [8] Descotte, Y., IJCAI, J. L.-, 1981, undefined, GARI: A Problem Solver That Plans How to Machine Mechanical Parts., ijcai.org.
- [9] Gantt, H. L., A graphical daily balance in manufacture. 1903.
- [10] Baker, K. R., Trietsch, D., *Principles of Sequencing and Scheduling*. 2009.
- [11] Baker, K. R., Minimizing earliness and tardiness costs in stochastic scheduling, *European Journal of Operational Research*, 236 (2), 445-452, 2014.

- [12] McCormick, S. T., Pinedo, M. L., Scheduling n Independent Jobs on m Uniform Machines with both Flowtime and Makespan Objectives: A Parametric Analysis, *ORSA Journal of Computing*, 7, 63–77, 1995.
- [13] Morton, T., Pentico, D. W., Heuristic scheduling systems: with applications to production systems and project management, 3. John Wiley & Sons, 1993.
- [14] Gordon, V., Proth, J. M., Chu, C., A survey of the state-of-the-art of common due date assignment and scheduling research, *European Journal of Operational Research*, 139 (1), 1-25, 2002.
- [15] Rippin, D. W. T., Batch process systems engineering: A retrospective and prospective review, *Computers and Chemical Engineering*, 17, S1–S13, 1993.
- [16] Chryssolouris, G., Chan, S., Suh, N. P., An integrated approach to process planning and scheduling, *CIRP Annals-Manufacturing Technology*, 34 (1), 413–417, 1985.
- [17] Khoshnevis, B., Chen, Q. M., Integration of process planning and scheduling functions, *Journal of Intelligent Manufacturing*, 2 (3), 165-175, 1991.
- [18] Phanden, R. K., Jain, A., Verma, R., Review on integration of process planning and scheduling, *DAAAM International Scientific Book*, 593–619, 2011.
- [19] Zhang, H. C., Mallur, S., An integrated model of process planning and production scheduling, *International Journal of Computer Integrated Manufacturing*, 7 (6), 356-364, 1994.
- [20] Goldberg, D. E., Holland, J. H., Genetic Algorithms and Machine Learning, *Machine Learning*, 3 (2), 95-99, 1988.
- [21] Nasr, N., Elsayed, E. A., Job shop scheduling with alternative machines, *International Journal of Production Research*, 28 (9), 1595-1609, 1990.
- [22] Chen, Q., Khoshnevis, B., Scheduling with flexible process plans, *Production Planning and Control*, 4 (4), 333-343, 1993.
- [23] Morad, N., Zalzal, A. M. S., Genetic algorithms in integrated process planning and scheduling, *Journal of Intelligent Manufacturing*, 10 (2), 169–179, 1999.

- [24] Kim, Y. K., Park, K., Ko, J., A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling, *Computers & Operations Research*, 30 (8), 1151-1171, 2003.
- [25] Lim, M. K., Zhang, D. Z., An integrated agent-based approach for responsive control of manufacturing resources, *Computers & Industrial Engineering*, 46 (2), 221–232, 2004.
- [26] Kumar, M., Rajotia, S., Integration of scheduling with computer aided process planning, *Journal of Materials Processing Technology*, 138 (1-3), 297-300, 2003.
- [27] Li, X., Gao, L., Zhang, C., Shao, X., A review on Integrated Process Planning and Scheduling, *International Journal of Manufacturing Research*, 5 (2), 161, 2010.
- [28] Guo, Y. W., Li, W. D., Mileham, A. R., Owen, G. W., Applications of particle swarm optimisation in integrated process planning and scheduling, *Robotics and Computer-Integrated Manufacturing*, 25 (2), 280-288, 2009.
- [29] Chryssolouris, G., Chan, S., Suh, N. P., An integrated approach to process planning and scheduling, *CIRP Annals-Manufacturing Technology*, 34 (1), 413–417, 1985.
- [30] Sundaram, R. M., Fu, S., Process planning and scheduling—a method of integration for productivity improvement, *Computers & Industrial Engineering*, 15 (1), 296–301, 1988.
- [31] Srihari, K., Greene, T. J., MACRO-CAPP: a prototype CAPP system for an FMS, *The International Journal of Advanced Manufacturing Technology*, 5 (1), 34–51, 1990.
- [32] Wilhelm, W. E., Shin, H.-M., Effectiveness of alternate operations in a flexible manufacturing system, *International Journal of Production Research*, 23 (1), 65–79, 1985.
- [33] Li, X., Zhang, C., Gao, L., Li, W., Shao, X., An agent-based approach for integrated process planning and scheduling, *Expert Systems with Applications*, 37 (2), 1256-1264, 2010.
- [34] Wong, T. N., Leung, C. W., Mak, K. L., Fung, R. Y. K., Integrated process planning and scheduling/rescheduling-an agent-based approach, *International Journal of Production Research*, 44 (18-19), 3627–3655, 2006.

- [35] Gu, P., Balasubramanian, S., Norrie, D. H., Bidding-based process planning and scheduling in a multi-agent system, *Computers and Industrial Engineering*, 32 (2), 477–496, 1997.
- [36] Lim, M. K., Zhang, Z., A multi-agent based manufacturing control strategy for responsive manufacturing, *Journal of Materials Processing Technology*, 139 (1-3 SPEC), 379-384, 2003.
- [37] Wong, T. N., Leung, C. W., Mak, K. L., Fung, R. Y. K., An agent-based negotiation approach to integrate process planning and scheduling, *International Journal of Production Research*, 44 (7), 1331-1351, 2006.
- [38] Lim, M. K., Zhang, D. Z., An integrated agent-based approach for responsive control of manufacturing resources, *Computers & Industrial Engineering*, 46 (2), 221–232, 2004.
- [39] Li, X., Gao, L., Shao, X., Zhang, C., Wang, C., Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling, *Computers & Operations Research*, 37 (4), 656-667, 2010.
- [40] Moon, C., Kim, J., Hur, S., Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain, *Computers and Industrial Engineering*, 43 (1-2), 331-349, 2002.
- [41] Moon, C., Seo, Y., Evolutionary algorithm for advanced process planning and scheduling in a multi-plant, *Computers and Industrial Engineering*, 48 (2), 311-325, 2005.
- [42] Li, X., Gao, L., Li, W., Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling, *Expert Systems with Applications*, 39 (1), 288-297, 2012.
- [43] Li, W. D., Gao, L., Li, X. Y., Guo, Y., Game theory-based cooperation of process planning and scheduling, içinde *Proceedings of the 2008 12th International Conference on Computer Supported Cooperative Work in Design, CSCWD*, 2008, 2, 841-845.
- [44] Phanden, R. K., Jain, A., Verma, R., An approach for integration of process planning and scheduling, *International Journal of Computer Integrated Manufacturing*, 26 (4), 284–302, 2013.

- [45] Amin-Naseri, M. R., Afshari, A. J., A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints, *International Journal of Advanced Manufacturing Technology*, 59 (1-4), 273-287, 2012.
- [46] Bierwirth, C., Mattfeld, D. C., Production Scheduling and Rescheduling with Genetic Algorithms, *Evolutionary Computation*, 7 (1), 1-17, 1999.
- [47] S. Chan, Felix., Kumar, V., Tiwari, M., Optimizing the Performance of an Integrated Process Planning and Scheduling Problem: An AIS-FLC based Approach, 2006 IEEE Conference on Cybernetics and Intelligent Systems, 1-8, 2006.
- [48] Ip, W. H., Li, Y., Man, K. F., Tang, K. S., Multi-product planning and scheduling using genetic algorithm approach, *Computers & Industrial Engineering*, 38 (2), 283-296, 2000.
- [49] Lee, H., Kim, S. S., Integration of process planning and scheduling using simulation based genetic algorithms, *International Journal of Advanced Manufacturing Technology*, 18 (8), 586-590, 2001.
- [50] Li, X., Gao, L., Shao, X., An active learning genetic algorithm for integrated process planning and scheduling, *Expert Systems with Applications*, 39 (8), 6683-6691, 2012.
- [51] Lihong, Q., Shengping, L., An improved genetic algorithm for integrated process planning and scheduling, *International Journal of Advanced Manufacturing Technology*, 58 (5-8), 727-740, 2012.
- [52] Liu, M., Sun, Z., Yan, J., Kang, J., An adaptive annealing genetic algorithm for the job-shop planning and scheduling problem, *Expert Systems with Applications*, 38 (8), 9248–9255, 2011.
- [53] Park, B. J., Choi, H. R., A genetic algorithm for integration of process planning and scheduling in a job shop, içinde *Australasian Joint Conference on Artificial Intelligence*, 2006, 647–657.
- [54] Shao, X., Li, X., Gao, L., Zhang, C., Integration of process planning and scheduling—A modified genetic algorithm-based approach, *Computers & Operations Research*, 36 (6), 2082-2096, 2009.

- [55] Sugimura, N., Shrestha, R., Inoue, J., Integrated process planning and scheduling in holonic manufacturing systems-optimization based on shop time and machining cost, içinde Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2003, 2003-Janua, 36-41.
- [56] Yip-Hoi, D., Dutta, D., A genetic algorithm application for sequencing operations in process planning for parallel machining, IIE Transactions (Institute of Industrial Engineers), 28 (1), 55-68, 1996.
- [57] Zhang, F., Zhang, Y. F., Nee, a. Y. C., Using Genetic Algorithms in Process Planning for Job Shop Machining, IEEE Transactions, 1 (4), 278-289, 1997.
- [58] Zhang, L., Wong, T. N., Distributed genetic algorithm for integrated process planning and scheduling based on multi agent system, IFAC Proceedings Volumes (IFAC-PapersOnline), 46 (9), 760-765, 2013.
- [59] Sundaram, R. M., Fu, S., Process planning and scheduling—a method of integration for productivity improvement, Computers & Industrial Engineering, 15 (1), 296–301, 1988.
- [60] Jin, L., Zhang, C., Shao, X., An effective hybrid honey bee mating optimization algorithm for integrated process planning and scheduling problems, International Journal of Advanced Manufacturing Technology, 80 (5-8), 1253-1264, 2015.
- [61] Li, X., Li, W., Cai, X., He, F., A honey-bee mating optimization approach of collaborative process planning and scheduling for sustainable manufacturing, içinde Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2013, 465-470.
- [62] Li, X. X., Li, W. D., Cai, X. T., He, F. Z., A hybrid optimization approach for sustainable process planning and scheduling, Integrated Computer-Aided Engineering, 22 (4), 311-326, 2015.
- [63] Lian, K., Zhang, C., Gao, L., Li, X., Integrated process planning and scheduling using an imperialist competitive algorithm, International Journal of Production Research, 50 (15), 4326-4343, 2012.
- [64] Lian, K., Zhang, C., Shao, X., Gao, L., Optimization of process planning with various flexibilities using an imperialist competitive algorithm, International Journal of Advanced Manufacturing Technology, 59 (5-8), 815-828, 2012.

- [65] Li, X., Gao, L., Wen, X., Application of an efficient modified particle swarm optimization algorithm for process planning, *International Journal of Advanced Manufacturing Technology*, 67 (5-8), 1355-1369, 2013.
- [66] Zhao, F., Hong, Y., Yu, D., Yang, Y., Zhang, Q., A hybrid particle swarm optimisation algorithm and fuzzy logic for process planning and production scheduling integration in holonic manufacturing systems, *International Journal of Computer Integrated Manufacturing*, 23 (1), 20-39, 2010.
- [67] Guo, Y. W., Li, W. D., Mileham, A. R., Owen, G. W., Applications of particle swarm optimisation in integrated process planning and scheduling, *Robotics and Computer-Integrated Manufacturing*, 25 (2), 280-288, 2009.
- [68] Tseng, Y.-J., Yu, F.-Y., Huang, F.-Y., A green assembly sequence planning model with a closed-loop assembly and disassembly sequence planning using a particle swarm optimization method, *The International Journal of Advanced Manufacturing Technology*, 57 (9-12), 1183-1197, 2011.
- [69] Zhu, H., Ye, W., Bei, G., A particle swarm optimization for integrated process planning and scheduling, içinde *Computer-Aided Industrial Design & Conceptual Design*, 2009. CAID & CD 2009. IEEE 10th International Conference on, 2009, 1070-1074.
- [70] Guo, Y. W., Li, W. D., Mileham, A. R., Owen, G. W., Optimisation of integrated process planning and scheduling using a particle swarm optimisation approach, *International Journal of Production Research*, 47 (14), 3775-3796, 2009.
- [71] Li, W. D., McMahon, C. A., A simulated annealing-based optimization approach for integrated process planning and scheduling, *International Journal of Computer Integrated Manufacturing*, 20 (1), 80-95, 2007.
- [72] Palmer, G. J., A simulated annealing approach to integrated production scheduling, *Journal of Intelligent Manufacturing*, 7 (3), 163–176, 1996.
- [73] Ausaf, M. F., Gao, L., Li, X., Al, A., A priority-based heuristic algorithm (PBHA) for optimizing integrated process planning and scheduling problem, *Cogent Engineering*, 2 (1), 2015.
- [74] Yu, M., Zhang, Y., Chen, K., Zhang, D., Integration of process planning and scheduling using a hybrid GA/PSO algorithm, *The International Journal of Advanced Manufacturing Technology*, 78 (1-4), 583-592, 2015.

- [75] Leung, C. W., Wong, T. N., Mak, K.-L. L., Fung, R. Y. K., Integrated process planning and scheduling by an agent-based ant colony optimization, *Computers and Industrial Engineering*, 59 (1), 166-180, 2010.
- [76] Baykasoğlu, A., Özbakır, L., A grammatical optimization approach for integrated process planning and scheduling, *Journal of Intelligent Manufacturing*, 20 (2), 211–221, 2009.
- [77] Jain, A., Jain, P. K., Singh, I. P., An integrated scheme for process planning and scheduling in FMS, *International Journal of Advanced Manufacturing Technology*, 30 (11-12), 1111-1118, 2006.
- [78] Kumar, M., Rajotia, S., Integration of process planning and scheduling in a job shop environment, *The International Journal of Advanced Manufacturing Technology*, 28 (1-2), 109-116, 2006.
- [79] Tan, W., Khoshnevis, B., Integration of process planning and scheduling—a review, *Journal of Intelligent Manufacturing*, 11 (1), 51–63, 2000.
- [80] Phanden, R. K., Jain, A., Verma, R., Integration of process planning and scheduling: A state-of-the-art review, *International Journal of Computer Integrated Manufacturing*, 24 (6), 517-534, 2011.
- [81] Li, X., Gao, L., Zhang, C., Shao, X., A review on Integrated Process Planning and Scheduling, *International Journal of Manufacturing Research*, 5 (2), 161, 2010.
- [82] Gen, M., Lin, L., Zhang, H., Evolutionary techniques for optimization problems in integrated manufacturing system: State-of-the-art-survey, *Computers and Industrial Engineering*, 56 (3), 779-808, 2009.
- [83] Scopus - Document search results. <https://www.scopus.com/>. Erişim Tarihi: 04.12.2017.
- [84] IEEE Xplore - Search Results, 2014. http://bibliotecavirtual.ups.edu.ec:2065/search/searchresult.jsp?refinements%3D4291944246%26queryText%3DElectric+System+Harmonic+Response&addRange=2010_2015_Publication_Year&pageNumber=1&resultAction=REFINE. Erişim Tarihi: 04.12.2017.
- [85] IPPS Results of ACM. [http://dl.acm.org/results.cfm?query=\(%252B%22ipps%22\)&within=owners.owner=HOSTED&filtered=&dte=&bfr=](http://dl.acm.org/results.cfm?query=(%252B%22ipps%22)&within=owners.owner=HOSTED&filtered=&dte=&bfr=). Erişim Tarihi: 24.12.2016.

- [86] Web of Science [v.5.23.2] - Web of Science Core Collection. .
- [87] Integrated process planning and scheduling-ScienceDirect. .
- [88] Zhang, L., Wong, T. N., An object-coding genetic algorithm for integrated process planning and scheduling, *European Journal of Operational Research*, 244 (2), 434-444, 2015.
- [89] Zhang, W., Gen, M., Jo, J., Hybrid sampling strategy-based multiobjective evolutionary algorithm for process planning and scheduling problem, *Journal of Intelligent Manufacturing*, 25 (5), 881-897, 2014.
- [90] Sidney, J. B., Optimal Single-Machine Scheduling with Earliness and Tardiness Penalties, *Operations Research*, 25 (1), 62-69, 1977.
- [91] Seidmann, A., Panwalkar, S. S., Smith, M. L., Optimal assignment of due-dates for a single processor scheduling problem, *International Journal of Production Research*, 19 (4), 393-399, 1981.
- [92] Ibaraki, T., Nakamura, Y., A dynamic programming method for single machine scheduling, *European Journal of Operational Research*, 76 (1), 72-82, 1994.
- [93] Ow, P. S., Morton, T. E., The single machine early/tardy problem, *Management Science*, 35 (2), 177-191, 1989.
- [94] Gupta, S. K., Sen, T., Minimizing a quadratic function of job lateness on a single machine, *Engineering Costs and Production Economics*, 7 (3), 187-194, 1983.
- [95] Lee, C.-Y., Choi, J. Y., A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights, *Computers & Operations Research*, 22 (8), 857-869, 1995.
- [96] Biskup, D., Jahnke, H., Common due date assignment for scheduling on a single machine with jointly reducible processing times, *International Journal of Production Economics*, 69 (3), 317-322, 2001.
- [97] Cheng, T. C. E., Chen, Z. L., Shakhlevich, N. V., Common due date assignment and scheduling with ready times, *Computers and Operations Research*, 29 (14), 1957-1967, 2002.

- [98] Gordon, V., Kubiak, W., Single machine scheduling with release and due date assignment to minimize the weighted number of late jobs, *Information Processing Letters*, 68 (3), 153-159, 1998.
- [99] Gordon, V. S., Strusevich, V. A., Single machine scheduling and due date assignment with positionally dependent processing times, *European Journal of Operational Research*, 198 (1), 57-62, 2009.
- [100] Nearchou, A. C., A differential evolution approach for the common due date early/tardy job scheduling problem, *Computers and Operations Research*, 35 (4), 1329-1343, 2008.
- [101] Ng Daniel, C. T. D., Cheng, T. C. E., Kovalyov, M. Y., Lam, S. S., Single machine scheduling with a variable common due date and resource-dependent processing times, *Computers and Operations Research*, 30 (8), 1173-1185, 2003.
- [102] Qi, X., Yu, G., Bard, J. F., Single machine scheduling with assignable due dates, *Discrete Applied Mathematics*, 122 (1-3), 211-233, 2002.
- [103] Wang, J.-B., Single machine scheduling with common due date and controllable processing times, *Applied Mathematics and Computation*, 174 (2), 1245-1254, 2006.
- [104] Ying, K.-C., Minimizing earliness-tardiness penalties for common due date single-machine scheduling problems by a recovering beam search algorithm, *Computers & Industrial Engineering*, 55 (2), 494-502, 2008.
- [105] Panwalkar, S. S., Smith, M. L., Seidmann, a., Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem, *Operations Research*, 30 (2), 391-399, 1982.
- [106] Chen, Z.-L., Scheduling and common due date assignment with earliness-tardiness penalties and batch delivery costs, *European Journal of Operational Research*, 93 (1), 49-60, 1996.
- [107] Cheng, T. C. E., Oğuz, C., Qi, X. D., Due-date assignment and single machine scheduling with compressible processing times, *International Journal of Production Economics*, 43 (2), 107-113, 1996.
- [108] Cheng, T. C. E., Kovalyov, M. Y., Batch scheduling and common due-date assignment on a single machine, *Discrete Applied Mathematics*, 70 (3), 231-245, 1996.

- [109] Bank, J., Werner, F., Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties, *Mathematical and computer modelling*, 33 (4), 363–383, 2001.
- [110] Cheng, T. C. E., A heuristic for common due-date assignment and job scheduling on parallel machines, *Journal of the Operational Research Society*, 40 (12), 1129-1135, 1989.
- [111] Lauff, V., Werner, F., Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey, *Mathematical and Computer Modelling*, 40 (5-6), 637-655, 2004.
- [112] Baker, K. R., Scudder, G. D., Sequencing with Earliness and Tardiness Penalties: A Review, *Operations Research*, 38 (1), 22-36, 1990.
- [113] Cheng, T. C. E., Gupta, M. C., Survey of scheduling research involving due date determination decisions, *European Journal of Operational Research*, 38 (2), 156-166, 1989.
- [114] Demir, H. I., Erden, C., Goksu, A., Uygun, O., Solving Process Planning, ATC Scheduling and Due-date Assignment Problems Concurrently Using Genetic Algorithm for Weighted Customers, içinde 4th International Symposium on Innovative Technologies in Engineering and Science (ISITES2016) 3-5 Nov 2016 Alanya/Antalya - Turkey, 2016, 1759-1770.
- [115] Demir, H. I., Erden, C., Ipek, M., Uygun, O., Solving Weighted Number of Operation Plus Processing Time Due-Date Assignment, Weighted Scheduling and Process Planning Integration Problem Using Genetic and Simulated Annealing Search Methods, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 11 (1), 73–81, 2017.
- [116] Demir, H. I., Erden, C., Solving process planning and weighted scheduling with WNOPPT weighted due-date assignment problem using some pure and hybrid meta-heuristics, *Sakarya University Journal of Science*, 21 (2), 1-1, 2017.
- [117] Demir, H. I., Cakar, T., Ibrahim Cil, Dugenci, M., Erden, C., Integrating Process Planning, WMS Dispatching, and WPPW Weighted Due Date Assignment Using a Genetic Algorithm, *International Journal of Computer and Information Engineering*, 10 (7), 1290–1298, 2016.
- [118] Zhao, L., Kops, L., An integrated CAPP and scheduling system, *NAMRC Proceedings of Society of Manufacturing*, 1987.

- [119] Zhang, L., Gao, L., Li, X., A hybrid genetic algorithm and tabu search for a multi-objective dynamic job shop scheduling problem, *International Journal of Production Research*, 51 (12), 3516-3531, 2013.
- [120] Adibi, M. A., Zandieh, M., Amiri, M., Multi-objective scheduling of dynamic job shop using variable neighborhood search, *Expert Systems with Applications*, 37 (1), 282–287, 2010.
- [121] Lin, S.-C., Goodman, E. D., Punch III, W. F., A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problem., içinde *ICGA*, 1997, 481–488.
- [122] Ramasesh, R., Dynamic job shop scheduling: a survey of simulation research, *Omega*, 18 (1), 43–57, 1990.
- [123] Allahverdi, A., Scheduling in stochastic flowshops with independent setup, processing and removal times, *Computers & Operations Research*, 24 (10), 955–960, 1997.
- [124] Dominic, P. D., Kaliyamoorthy, S., Kumar, M. S., Efficient dispatching rules for dynamic job shop scheduling, *The International Journal of Advanced Manufacturing Technology*, 24 (1-2), 70–75, 2004.
- [125] Aydin, M. E., Öztemel, E., Dynamic job-shop scheduling using reinforcement learning agents, *Robotics and Autonomous Systems*, 33 (2), 169-178, 2000.
- [126] Li, S., Wu, Z., Pang, X., Job shop scheduling in real-time cases, *International Journal of Advanced Manufacturing Technology*, 26 (7-8), 870-875, 2005.
- [127] Sha, D. Y., Liu, C. H., Using data mining for due date assignment in a dynamic job shop environment, *International Journal of Advanced Manufacturing Technology*, 25 (11-12), 1164-1174, 2005.
- [128] Zandieh, M., Adibi, M. A., Dynamic job shop scheduling using variable neighbourhood search, *International Journal of Production Research*, 48 (8), 2449-2458, 2010.
- [129] Lin, C.-W. R., Chen, H.-Y., Xiao, Q.-S., Dynamic Integrated Process Planning and Scheduling, *Journal of the Chinese Institute of Industrial Engineers*, 18 (2), 21–32, 2001.

- [130] Xia, H., Li, X., Gao, L., A hybrid genetic algorithm with variable neighborhood search for dynamic integrated process planning and scheduling, *Computers & Industrial Engineering*, 102, 99-112, 2016.
- [131] Ramasesh, R., Dynamic job shop scheduling: A survey of simulation research, *Omega*, 18 (1), 43-57, 1990.
- [132] Pinedo, M. L., *Scheduling*. 2008.
- [133] Hutchinson, G. K., Pflughoeft, K. A., Flexible process plans: Their value in flexible automation systems, *International Journal of Production Research*, 32 (3), 707-719, 1994.
- [134] Saygin, C., Kilic, S. E., Integrating flexible process plans with scheduling in flexible manufacturing systems, *International Journal of Advanced Manufacturing Technology*, 15 (4), 268-280, 1999.
- [135] Benjaafar, S., Ramakrishnan, R., Modeling, Measurement, and Evaluation of Sequencing, Flexibility in Manufacturing Systems, *International Journal of Production Research*, 34 (5). 1195-1220, 1996.
- [136] Lee, K., engineering, M. J.-C. & industrial, 1994, undefined, Petri net application in flexible process planning, Elsevier.
- [137] Chiang, T. C., Fu, L. C., Using dispatching rules for job shop scheduling with due-date based objectives, *International Journal of Production Research*, 45, 3245-3262, 2007.
- [138] Mouelhi-Chibani, W., Pierreval, H., Training a neural network to select dispatching rules in real time, *Computers and Industrial Engineering*, 58 (2), 249-256, 2010.
- [139] Kaban, A. K., Othman, Z., Rohmah, D. S., Comparison of dispatching rules in job-shop Schedulingproblem Usingsimulation: A case study, *International Journal of Simulation Modelling*, 11 (3), 129-140, 2012.
- [140] Kanet, J. J., Minimizing Variation of Flow Time in Single Machine Systems, *Management Science*, 27 (12), 1453-1459, 1981.
- [141] Hall, N. G., Scheduling problems with generalized due dates, *IIE Transactions (Institute of Industrial Engineers)*, 18 (2), 220-222, 1986.

- [142] Hall, N., Single-and multiple-processor models for minimizing completion time variance, Naval Research Logistics (NRL), 1986.
- [143] Szwarc, W., Single-machine scheduling to minimize absolute deviation of completion times from a common due date, Naval Research Logistics (NRL), 36 (5), 663-673, 1989.
- [144] Baker, K. R., Sequencing Rules and Due-Date Assignments in a Job Shop, Management Science, 30 (9), 1093-1104, 1984.
- [145] Yu, X., Ram, B., Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups, International Journal of Production Research, 44 (22), 4793-4813, 2006.
- [146] Vinod, V., Sridharan, R., Dynamic job shop scheduling with sequence-dependent setup times: Simulation modelling and analysis, International Journal of Advanced Manufacturing Technology, 36 (3), 355-372, 2008.
- [147] Holland, J. H., Genetic algorithms, Scientific american, 267 (1), 66–73, 1992.
- [148] Velusamy Subramaniam, Scheduling of Manufacturing Systems Based on Extreme Value Theory and Genetic Algorithms, MIT, 1995.
- [149] Glover, F., Tabu search—part I, ORSA Journal on computing, 1 (3), 190–206, 1989.
- [150] Gholipour, Y., Shahbazi, M. M., Behnia, A., An improved version of Inverse Distance Weighting metamodel assisted Harmony Search algorithm for truss design optimization, Latin American Journal of Solids and Structures, 10 (2), 283-300, 2013.
- [151] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., Optimization by simulated annealing., Science (New York, N.Y.), 220 (4598), 671-80, 1983.
- [152] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., Equation of State Calculations by Fast Computing Machines, The Journal of Chemical Physics, 21 (6), 1087-1092, 1953.
- [153] Kirkpatrick, S., Optimization by simulated annealing: Quantitative studies, Journal of Statistical Physics, 34 (5-6), 975-986, 1984.

- [154] Eberhart, R., Kennedy, J., A new optimizer using particle swarm theory, içinde MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, 39–43.
- [155] Ba, L. vd., A Mathematical Model for Multiworkshop IPPS Problem in Batch Production, Mathematical Problems in Engineering, 2018, 1-16, 2018.
- [156] Petrović, M., Vuković, N., Mitić, M., Miljković, Z., Integration of process planning and scheduling using chaotic particle swarm optimization algorithm, Expert Systems with Applications, 64, 569-588, 2016.
- [157] Yu, M., Zhang, Y., Chen, K., Zhang, D., Integration of process planning and scheduling using a hybrid GA/PSO algorithm, The International Journal of Advanced Manufacturing Technology, 78 (1), 583-592, 2015.
- [158] Petrović, M., Mitić, M., Vuković, N., Miljković, Z., Chaotic particle swarm optimization algorithm for flexible process planning, The International Journal of Advanced Manufacturing Technology, 85 (9), 2535-2555, 2016.
- [159] Wang, Y. F., Zhang, Y. F., Fuh, J. Y. H., A PSO-based multi-objective optimization approach to the integration of process planning and scheduling, içinde IEEE ICCA 2010, 2010, 614-619.
- [160] Strasser, S., Goodman, R., Sheppard, J., Butcher, S., A new discrete particle swarm optimization algorithm, içinde Proceedings of the Genetic and Evolutionary Computation Conference 2016, 2016, 53–60.
- [161] Pan, Q.-K., Fatih Tasgetiren, M., Liang, Y.-C., A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, Computers & Operations Research, 35 (9), 2807-2839, 2008.
- [162] Ragan-Kelley, M., Perez, F., ... B. G.-A. F. M., 2014, undefined, The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication., adsabs.harvard.edu.
- [163] Oliphant, T., A guide to NumPy, 2006.
- [164] Science, W. M.-P. of the 9th P. in, 2010, undefined, Data structures for statistical computing in python, pdfs.semanticscholar.org.
- [165] Hunter, J. D., Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9 (3), 90-95, 2007.

- [166] Software, R. van der H.-T. J. of O. S., 2018, undefined, salabim: discrete event simulation and animation in Python, theoj.org.



EKLER

EK 1: Problemin Girdi Verileri

Çalışma için üretilen veriler, bu çalışma için özel olarak üretilmiştir, çünkü literatürde hem süreç planlama, hem zamanlama hem de zamanlama ile ilgili benzer veri bulunmamaktadır. Python programlama dilinde NumPy kütüphanesi kullanılarak üretilen mağaza katlarına ait veriler atölyelere göre ayrılmıştır ve '.txt' dosyası olarak kaydedilmiştir. Öncelikle, iş varış süreleri üssel dağılıma göre üretilmiş ve 'arrival_time_ ({shop floor number} .txt ') dosyasına kaydedilmiştir. İkinci olarak, işlerin her bir alternatif işlem planı için makine sıraları machine_numbers{shop floor number} .txt' dosyasına olduğu gibi kaydedilmiştir. İşlem süreleri ise operation durations {shop floor number} .txt' dosyasına kaydedilmiştir. Son olarak, ağırlıklar 'weights- {shop floor number} .txt' dosyasında verilmiştir. Veri dosyaları https://github.com/canererden/dippsdda_dataset adresinde açık kaynak olarak paylaşılmıştır.

EK 2: Mini Atölyeye Ait Simülasyon Sonuçları

```

163          main run          ends on no events left @ 163+
473 0.000 job0      current
474          job0 hold        scheduled for 16.000 @ 474+
473 0.000 job1      current
474          job1 hold        scheduled for 20.000 @ 474+
473 0.000 job2      current
474          job2 hold        scheduled for 22.000 @ 474+
473 0.000 job3      current
474          job3 hold        scheduled for 45.000 @ 474+
207 0.000 machine0  current
210          machine0 hold    scheduled for 0.000 @ 210+
207 0.000 machine1  current
210          machine1 hold    scheduled for 0.000 @ 210+
210+ 0.000 machine0  current
445          machine0 passivate
210+ 0.000 machine1  current
445          machine1 passivate
474+ 16.000 job0     current
475          job0             enter .jobs
483          operation.0.0 m=machine0 d=10 create data component
484          operation.0.0 m=machine0 d=10 enter job0.operations
483          operation.0.1 m=machine1 d=9 create data component
484          operation.0.1 m=machine1 d=9 enter job0.operations
483          operation.0.2 m=machine1 d=23 create data component
484          operation.0.2 m=machine1 d=23 enter job0.operations
483          operation.0.3 m=machine1 d=17 create data component
484          operation.0.3 m=machine1 d=17 enter job0.operations
488          operation.0.0 m=machine0 d=10 enter machine.0.queue
490          machine0 activate scheduled for 16.000 @ 445+
491          job0 passivate
445+ 16.000 machine0  current
210          machine0 hold    scheduled for 16.000 @ 210+
210+ 16.000 machine0  current
440          machine0 hold    scheduled for 26.000 @ 440+
474+ 20.000 job1     current
475          job1             enter .jobs
483          operation.1.0 m=machine1 d=1 create data component
484          operation.1.0 m=machine1 d=1 enter job1.operations
483          operation.1.1 m=machine0 d=12 create data component
484          operation.1.1 m=machine0 d=12 enter job1.operations
483          operation.1.2 m=machine1 d=45 create data component
484          operation.1.2 m=machine1 d=45 enter job1.operations
483          operation.1.3 m=machine1 d=7 create data component
484          operation.1.3 m=machine1 d=7 enter job1.operations
488          operation.1.0 m=machine1 d=1 enter machine.1.queue
490          machine1 activate scheduled for 20.000 @ 445+
491          job1 passivate
445+ 20.000 machine1  current
210          machine1 hold    scheduled for 20.000 @ 210+
210+ 20.000 machine1  current
440          machine1 hold    scheduled for 21.000 @ 440+
440+ 21.000 machine1  current
441          operation.1.0 m=machine1 d=1 leave job1.operations
441          operation.1.0 m=machine1 d=1 leave machine.1.queue

```

```

443      job1 activate          scheduled for 21.000 @ 491+
210      machine1 hold        scheduled for 21.000 @ 210+
491+ 21.000 job1             current
488      operation.1.1 m=machine0 d=12 enter machine.0.queue
491      job1 passivate
210+ 21.000 machine1         current
445      machine1 passivate
474+ 22.000 job2             current
475      job2                  enter .jobs
483      operation.2.0 m=machine0 d=23 create data component
484      operation.2.0 m=machine0 d=23 enter job2.operations
483      operation.2.1 m=machine0 d=17 create data component
484      operation.2.1 m=machine0 d=17 enter job2.operations
483      operation.2.2 m=machine1 d=5 create data component
484      operation.2.2 m=machine1 d=5 enter job2.operations
483      operation.2.3 m=machine1 d=13 create data component
484      operation.2.3 m=machine1 d=13 enter job2.operations
488      operation.2.0 m=machine0 d=23 enter machine.0.queue
491      job2 passivate
440+ 26.000 machine0         current
441      operation.0.0 m=machine0 d=10 leave job0.operations
441      operation.0.0 m=machine0 d=10 leave machine.0.queue
443      job0 activate          scheduled for 26.000 @ 491+
210      machine0 hold        scheduled for 26.000 @ 210+
491+ 26.000 job0             current
488      operation.0.1 m=machine1 d=9 enter machine.1.queue
490      machine1 activate      scheduled for 26.000 @ 445+
491      job0 passivate
210+ 26.000 machine0         current
440      machine0 hold          scheduled for 49.000 @ 440+
445+ 26.000 machine1         current
210      machine1 hold          scheduled for 26.000 @ 210+
210+ 26.000 machine1         current
440      machine1 hold          scheduled for 35.000 @ 440+
440+ 35.000 machine1         current
441      operation.0.1 m=machine1 d=9 leave job0.operations
441      operation.0.1 m=machine1 d=9 leave machine.1.queue
443      job0 activate          scheduled for 35.000 @ 491+
210      machine1 hold          scheduled for 35.000 @ 210+
491+ 35.000 job0             current
488      operation.0.2 m=machine1 d=23 enter machine.1.queue
491      job0 passivate
210+ 35.000 machine1         current
440      machine1 hold          scheduled for 58.000 @ 440+
474+ 45.000 job3             current
475      job3                  enter .jobs
483      operation.3.0 m=machine1 d=10 create data component
484      operation.3.0 m=machine1 d=10 enter job3.operations
483      operation.3.1 m=machine0 d=8 create data component
484      operation.3.1 m=machine0 d=8 enter job3.operations
483      operation.3.2 m=machine1 d=7 create data component
484      operation.3.2 m=machine1 d=7 enter job3.operations
483      operation.3.3 m=machine0 d=10 create data component
484      operation.3.3 m=machine0 d=10 enter job3.operations
488      operation.3.0 m=machine1 d=10 enter machine.1.queue
491      job3 passivate
440+ 49.000 machine0         current
441      operation.2.0 m=machine0 d=23 leave job2.operations

```

```

441          operation.2.0 m=machine0 d=23      leave machine.0.queue
443          job2 activate                        scheduled for 49.000 @ 491+
210          machine0 hold                      scheduled for 49.000 @ 210+
491+ 49.000 job2          current
488          operation.2.1 m=machine0 d=17      enter machine.0.queue
491          job2 passivate
210+ 49.000 machine0      current
440          machine0 hold                      scheduled for 66.000 @ 440+
440+ 58.000 machine1      current
441          operation.0.2 m=machine1 d=23      leave job0.operations
441          operation.0.2 m=machine1 d=23      leave machine.1.queue
443          job0 activate                      scheduled for 58.000 @ 491+
210          machine1 hold                    scheduled for 58.000 @ 210+
491+ 58.000 job0          current
488          operation.0.3 m=machine1 d=17      enter machine.1.queue
491          job0 passivate
210+ 58.000 machine1      current
440          machine1 hold                    scheduled for 68.000 @ 440+
440+ 66.000 machine0      current
441          operation.2.1 m=machine0 d=17      leave job2.operations
441          operation.2.1 m=machine0 d=17      leave machine.0.queue
443          job2 activate                      scheduled for 66.000 @ 491+
210          machine0 hold                    scheduled for 66.000 @ 210+
491+ 66.000 job2          current
488          operation.2.2 m=machine1 d=5       enter machine.1.queue
491          job2 passivate
210+ 66.000 machine0      current
440          machine0 hold                    scheduled for 78.000 @ 440+
440+ 68.000 machine1      current
441          operation.3.0 m=machine1 d=10      leave job3.operations
441          operation.3.0 m=machine1 d=10      leave machine.1.queue
443          job3 activate                      scheduled for 68.000 @ 491+
210          machine1 hold                    scheduled for 68.000 @ 210+
491+ 68.000 job3          current
488          operation.3.1 m=machine0 d=8       enter machine.0.queue
491          job3 passivate
210+ 68.000 machine1      current
440          machine1 hold                    scheduled for 73.000 @ 440+
440+ 73.000 machine1      current
441          operation.2.2 m=machine1 d=5       leave job2.operations
441          operation.2.2 m=machine1 d=5       leave machine.1.queue
443          job2 activate                      scheduled for 73.000 @ 491+
210          machine1 hold                    scheduled for 73.000 @ 210+
491+ 73.000 job2          current
488          operation.2.3 m=machine1 d=13      enter machine.1.queue
491          job2 passivate
210+ 73.000 machine1      current
440          machine1 hold                    scheduled for 90.000 @ 440+
440+ 78.000 machine0      current
441          operation.1.1 m=machine0 d=12      leave job1.operations
441          operation.1.1 m=machine0 d=12      leave machine.0.queue
443          job1 activate                      scheduled for 78.000 @ 491+
210          machine0 hold                    scheduled for 78.000 @ 210+
491+ 78.000 job1          current
488          operation.1.2 m=machine1 d=45      enter machine.1.queue
491          job1 passivate
210+ 78.000 machine0      current
440          machine0 hold                    scheduled for 86.000 @ 440+

```

```

440+ 86.000 machine0      current
441      operation.3.1 m=machine0 d=8      leave job3.operations
441      operation.3.1 m=machine0 d=8      leave machine.0.queue
443      job3 activate                      scheduled for 86.000 @ 491+
210      machine0 hold                      scheduled for 86.000 @ 210+
491+ 86.000 job3          current
488      operation.3.2 m=machine1 d=7      enter machine.1.queue
491      job3 passivate
210+ 86.000 machine0      current
445      machine0 passivate
440+ 90.000 machine1      current
441      operation.0.3 m=machine1 d=17      leave job0.operations
441      operation.0.3 m=machine1 d=17      leave machine.1.queue
443      job0 activate                      scheduled for 90.000 @ 491+
210      machine1 hold                      scheduled for 90.000 @ 210+
491+ 90.000 job0          current
492      job0                              leave .jobs
495+      job0 ended
210+ 90.000 machine1      current
440      machine1 hold                      scheduled for 97.000 @ 440+
440+ 97.000 machine1      current
441      operation.3.2 m=machine1 d=7      leave job3.operations
441      operation.3.2 m=machine1 d=7      leave machine.1.queue
443      job3 activate                      scheduled for 97.000 @ 491+
210      machine1 hold                      scheduled for 97.000 @ 210+
491+ 97.000 job3          current
488      operation.3.3 m=machine0 d=10      enter machine.0.queue
490      machine0 activate                  scheduled for 97.000 @ 445+
491      job3 passivate
210+ 97.000 machine1      current
440      machine1 hold                      scheduled for 142.000 @ 440+
445+ 97.000 machine0      current
210      machine0 hold                      scheduled for 97.000 @ 210+
210+ 97.000 machine0      current
440      machine0 hold                      scheduled for 107.000 @ 440+
440+ 107.000 machine0      current
441      operation.3.3 m=machine0 d=10      leave job3.operations
441      operation.3.3 m=machine0 d=10      leave machine.0.queue
443      job3 activate                      scheduled for 107.000 @ 491+
210      machine0 hold                      scheduled for 107.000 @ 210+
491+ 107.000 job3          current
492      job3                              leave .jobs
495+      job3 ended
210+ 107.000 machine0      current
445      machine0 passivate
440+ 142.000 machine1      current
441      operation.1.2 m=machine1 d=45      leave job1.operations
441      operation.1.2 m=machine1 d=45      leave machine.1.queue
443      job1 activate                      scheduled for 142.000 @ 491+
210      machine1 hold                      scheduled for 142.000 @ 210+
491+ 142.000 job1          current
488      operation.1.3 m=machine1 d=7      enter machine.1.queue
491      job1 passivate
210+ 142.000 machine1      current
440      machine1 hold                      scheduled for 149.000 @ 440+
440+ 149.000 machine1      current
441      operation.1.3 m=machine1 d=7      leave job1.operations
441      operation.1.3 m=machine1 d=7      leave machine.1.queue

```

```

443          job1 activate          scheduled for 149.000 @ 491+
210          machine1 hold          scheduled for 149.000 @ 210+
491+ 149.000 job1          current
492          job1                  leave .jobs
495+          job1 ended
210+ 149.000 machine1      current
440          machine1 hold          scheduled for 162.000 @ 440+
440+ 162.000 machine1      current
441          operation.2.3 m=machine1 d=13 leave job2.operations
441          operation.2.3 m=machine1 d=13 leave machine.1.queue
443          job2 activate          scheduled for 162.000 @ 491+
210          machine1 hold          scheduled for 162.000 @ 210+
491+ 162.000 job2          current
492          job2                  leave .jobs
495+          job2 ended
210+ 162.000 machine1      current
445          machine1 passivate
163+          run ended            no events left
163+ 162.000 main          current

```

EK 3: Tüm Atölyeler İçin Çalıştırılan Entegrasyon Seviyeleri Sonuçları

Algoritma	Atölye No	Seviye	En İyi	Ort.	En Kötü	CPU
ord	1	SIRO-RDM	189,799	197,84	208,15	500
vns	1	SIRO-RDM	187,932	188,2766	197,118	493
sa	1	SIRO-RDM	181,4986	190,2754	212,6426	477
ta	1	SIRO-RDM	171,5833	173,3065	191,2409	573
ga	1	SIRO-RDM	172,986	174,41	175,03	1141
ha	1	SIRO-RDM	169,6578	177,5656	199,645	760
ord	1	DSP-RDM	187,014	211,8	262,07	500
vns	1	DSP-RDM	188,0208	189,4525	233,6538	406
sa	1	DSP-RDM	183,2939	206,3335	257,5196	412
ta	1	DSP-RDM	170,7066	171,2485	189,5434	440
ga	1	DSP-RDM	173,293	174,59	175,21	977
ha	1	DSP-RDM	178,3833	207,0538	260,4671	864
ord	1	SIRO-DD	202,888	293,19	725,38	500
vns	1	SIRO-DD	231,7806	232,4019	249,0418	399
sa	1	SIRO-DD	211,4863	240,4759	284,5683	405
ta	1	SIRO-DD	200,9791	208,4916	253,6258	433
ga	1	SIRO-DD	202,888	207,96	209,15	971
ha	1	SIRO-DD	211,1215	241,5143	287,9275	858
ord	1	DSP-DD	189,423	241,53	380,76	500
vns	1	DSP-DD	129,3917	133,7719	306,5781	434
sa	1	DSP-DD	151,3109	225,6102	382,3516	438
ta	1	DSP-DD	143,5275	160,0798	563,7729	488
ga	1	DSP-DD	136,859	138,35	139,07	1067
ha	1	DSP-DD	153,3865	245,7309	388,2182	754

İkinci Atölye						
ord	2	SIRO-RDM	370,554	385,83	393,98	500
vns	2	SIRO-RDM	373,3582	373,8946	377,277	505
sa	2	SIRO-RDM	386,9038	392,4828	406,4526	539
ta	2	SIRO-RDM	344,4066	351,9891	386,7365	531
ga	2	SIRO-RDM	344,466	345,64	346,02	1109
ha	2	SIRO-RDM	343,3347	347,8325	355,7051	944
ord	2	DSP-RDM	373,668	383,31	393,01	500
vns	2	DSP-RDM	337,2194	340,865	380,6373	501
sa	2	DSP-RDM	362,1004	385,5406	410,7803	543
ta	2	DSP-RDM	333,2108	339,7133	386,5855	533
ga	2	DSP-RDM	343,905	344,21	344,58	1105
ha	2	DSP-RDM	339,2383	385,8559	410,1247	940
ord	2	SIRO-DD	296,44	1248,14	3101,02	500
vns	2	SIRO-DD	402,7972	402,9146	415,4128	499
sa	2	SIRO-DD	394,813	419,4379	451,9972	542
ta	2	SIRO-DD	385,5426	387,8251	406,832	540
ga	2	SIRO-DD	296,44	375,81	393,54	1119
ha	2	SIRO-DD	401,3369	421,8849	450,9009	936
ord	2	DSP-DD	356,209	1938,08	3946,46	500
vns	2	DSP-DD	229,8354	255,3888	350,7523	553
sa	2	DSP-DD	287,5247	408,7425	2669,37	447
ta	2	DSP-DD	242,4128	247,4087	308,5316	556
ga	2	DSP-DD	270,924	271,29	271,69	1365
ha	2	DSP-DD	253,4969	409,4754	530,5546	662

Üçüncü Atölye						
ord	3	SIRO-RDM	571,652	585,27	605,41	500
vns	3	SIRO-RDM	578,4146	578,9099	581,5755	364
sa	3	SIRO-RDM	580,6569	586,1046	594,4654	397
ta	3	SIRO-RDM	510,4405	527,1082	585,4372	470
ga	3	SIRO-RDM	532,417	533,24	533,69	847
ha	3	SIRO-RDM	528,7208	531,9395	534,8635	463
ord	3	DSP-RDM	576,47	582,81	595,42	500
vns	3	DSP-RDM	497,6071	512,0105	586,1976	363
sa	3	DSP-RDM	560,0077	582,2758	606,6129	399
ta	3	DSP-RDM	502,5885	512,9368	583,2066	475
ga	3	DSP-RDM	527,506	528,51	529,04	850
ha	3	DSP-RDM	534,2356	573,228	596,0202	493
ord	3	SIRO-DD	434,601	4359,4	8632,31	500
vns	3	SIRO-DD	613,874	614,1715	627,4107	361
sa	3	SIRO-DD	591,5926	611,285	637,5222	394
ta	3	SIRO-DD	568,0073	568,0073	568,0073	471
ga	3	SIRO-DD	434,601	558,31	582,3	842
ha	3	SIRO-DD	599,8599	616,0165	636,026	490
ord	3	DSP-DD	559,369	2325,73	8632,37	500
vns	3	DSP-DD	379,2853	485,8439	6796,463	536
sa	3	DSP-DD	404,1375	521,0067	634,7613	462
ta	3	DSP-DD	392,213	491,0176	2378,634	532
ga	3	DSP-DD	373,555	373,76	374	784
ha	3	DSP-DD	373,8042	499,8786	604,8769	756

Dördüncü Atölye						
ord	4	SIRO-RDM	823,229	837,68	852,31	500
vns	4	SIRO-RDM	840,5938	841,0214	844,8292	580
sa	4	SIRO-RDM	853,092	856,8732	860,3483	589
ta	4	SIRO-RDM	772,3274	783,4834	825,6438	549
ga	4	SIRO-RDM	753,854	755,24	756,17	1303
ha	4	SIRO-RDM	769,1896	773,5968	777,4301	1004
ord	4	DSP-RDM	814,704	847,06	859,76	500
vns	4	DSP-RDM	723,2963	754,4661	845,5741	616
sa	4	DSP-RDM	814,5078	838,9375	862,7135	657
ta	4	DSP-RDM	719,5623	741,9184	846,9321	657
ga	4	DSP-RDM	760,174	760,42	760,67	1468
ha	4	DSP-RDM	773,3078	838,4707	863,7491	931
ord	4	SIRO-DD	614,411	2089,46	10179,93	500
vns	4	SIRO-DD	596,5121	596,5121	596,5121	576
sa	4	SIRO-DD	852,8051	873,1058	902,515	596
ta	4	SIRO-DD	837,6314	847,8893	873,2369	553
ga	4	SIRO-DD	614,411	782,31	827,31	1301
ha	4	SIRO-DD	848,3897	870,5666	904,6807	999
ord	4	DSP-DD	620,126	3575,49	12810,82	500
vns	4	DSP-DD	499,3594	522,1481	612,2792	284
sa	4	DSP-DD	562,7229	2846,338	8624,635	314
ta	4	DSP-DD	516,7354	677,8267	5213,014	288
ga	4	DSP-DD	554,938	563,42	568,79	1297
ha	4	DSP-DD	569,1301	604,46	668,7747	442

Beşinci Atölye						
ord	5	SIRO-RDM	945,327	960,21	974,97	500
vns	5	SIRO-RDM	953,5925	953,8786	956,1518	463
sa	5	SIRO-RDM	965,0828	969,2953	972,0752	442
ta	5	SIRO-RDM	852,3661	873,5448	964,8169	463
ga	5	SIRO-RDM	883,782	885,74	886,47	974
ha	5	SIRO-RDM	908,2991	911,3061	914,2229	944
ord	5	DSP-RDM	954,809	965,39	978,24	500
vns	5	DSP-RDM	854,1197	875,9524	966,4621	462
sa	5	DSP-RDM	947,9131	964,5765	982,2068	445
ta	5	DSP-RDM	850,5939	873,5361	971,0688	460
ga	5	DSP-RDM	883,6	885,05	886,27	970
ha	5	DSP-RDM	896,4066	957,4785	978,1591	969
ord	5	SIRO-DD	638,855	9421,44	22518,54	500
vns	5	SIRO-DD	683,3516	683,3516	683,3516	458
sa	5	SIRO-DD	966,4362	978,0996	994,5991	444
ta	5	SIRO-DD	949,0905	956,3368	981,5518	461
ga	5	SIRO-DD	638,855	853,72	950,51	968
ha	5	SIRO-DD	968,8244	977,6688	991,8932	903
ord	5	DSP-DD	662,418	7333,71	22525,91	500
vns	5	DSP-DD	592,997	782,7602	943,2604	477
sa	5	DSP-DD	800,6292	1674,417	6363,492	439
ta	5	DSP-DD	577,5146	614,2834	678,471	439
ga	5	DSP-DD	616,003	617,75	618,65	969
ha	5	DSP-DD	611,3052	675,3411	904,3604	828

Altıncı Atölye						
ord	6	SIRO-RDM	1166,537	1177,97	1189,88	500
vns	6	SIRO-RDM	1174,252	1174,462	1175,927	509
sa	6	SIRO-RDM	1165,055	1166,931	1168,145	528
ta	6	SIRO-RDM	1062,489	1091,356	1180,096	593
ga	6	SIRO-RDM	1105,548	1106	1106,73	1105
ha	6	SIRO-RDM	1115,943	1117,524	1119,346	1189
ord	6	DSP-RDM	1166,17	1183,29	1200,36	500
vns	6	DSP-RDM	1050,455	1080,39	1172,263	511
sa	6	DSP-RDM	1159,591	1184,074	1207,987	532
ta	6	DSP-RDM	1052,902	1082,513	1191,217	603
ga	6	DSP-RDM	1096,869	1097,82	1098,84	1116
ha	6	DSP-RDM	1094,314	1170,79	1193,817	1197
ord	6	SIRO-DD	1096,185	10070,29	32526,86	500
vns	6	SIRO-DD	1193,557	1194,001	1204,65	506
sa	6	SIRO-DD	1175,348	1189,844	1205,527	528
ta	6	SIRO-DD	1021,733	1021,733	1021,733	597
ga	6	SIRO-DD	1096,185	1150,66	1157,4	1102
ha	6	SIRO-DD	1175,814	1188,678	1204,038	1063
ord	6	DSP-DD	765,902	8684,77	26905,19	500
vns	6	DSP-DD	696,2463	741,9177	920,1784	492
sa	6	DSP-DD	774,226	1078,359	1477,558	498
ta	6	DSP-DD	1137,453	5783,417	9204,254	562
ga	6	DSP-DD	756,235	756,32	756,42	1081
ha	6	DSP-DD	754,7635	784,1422	806,5042	993

Yedinci Atölye						
ord	7	SIRO-RDM	1351,213	1378,47	1404,49	500
vns	7	SIRO-RDM	1371,872	1371,884	1373,396	301
sa	7	SIRO-RDM	1381,321	1385,94	1390,53	290
ta	7	SIRO-RDM	1266,833	1311,455	1395,076	318
ga	7	SIRO-RDM	1294,028	1296,22	1297,52	607
ha	7	SIRO-RDM	1299,018	1302,571	1306,691	388
ord	7	DSP-RDM	1351,818	1375,61	1413,67	500
vns	7	DSP-RDM	1227,031	1284,628	1372,766	299
sa	7	DSP-RDM	1360,189	1379,657	1394,602	292
ta	7	DSP-RDM	1236,315	1288,963	1374,009	315
ga	7	DSP-RDM	1302,608	1304,38	1305,95	607
ha	7	DSP-RDM	1320,248	1347,215	1377,586	388
ord	7	DSP-DD	945,059	5234,9	29655,7	500
vns	7	DSP-DD	1311,53	1331,653	1363,483	301
sa	7	DSP-DD	1101,816	19116,27	29652,49	288
ta	7	DSP-DD	906,8463	10896,93	35985,14	300
ga	7	DSP-DD	886,771	889,42	891,35	647
ha	7	DSP-DD	886,9392	904,5843	930,8908	326

Sekizinci Atölye						
ord	8	SIRO-RDM	1647,953	1665,6	1682,18	500
vns	8	SIRO-RDM	1670,781	1670,788	1672,396	330
sa	8	SIRO-RDM	1662,963	1665,285	1667,407	336
ta	8	SIRO-RDM	1540,97	1583,978	1662,53	333
ga	8	SIRO-RDM	1576,268	1578,43	1579,66	698
ha	8	SIRO-RDM	1596,393	1598,752	1600,792	683
ord	8	DSP-RDM	1641,615	1666,03	1695,3	500
vns	8	DSP-RDM	1524,838	1572,95	1665,591	325
sa	8	DSP-RDM	1648,103	1664,702	1683,021	335
ta	8	DSP-RDM	1520,669	1568,201	1654,376	332
ga	8	DSP-RDM	1583,77	1585,29	1586,08	698
ha	8	DSP-RDM	1592,137	1633,138	1666,773	698
ord	8	SIRO-DD	1135,059	8700,08	20388,91	500
vns	8	SIRO-DD	1647,398	1647,424	1651,048	168
sa	8	SIRO-DD	1641,088	1654,843	1674,169	171
ta	8	SIRO-DD	1635,279	1636,651	1649,231	202
ga	8	SIRO-DD	1135,059	1530,61	1623,52	350
ha	8	SIRO-DD	1643,088	1654,901	1667,067	295
ord	8	DSP-DD	1164,778	13141,17	60112,32	500
vns	8	DSP-DD	1012,305	1039,747	1122,021	341
sa	8	DSP-DD	1378,499	10455,41	20391,21	325
ta	8	DSP-DD	1104,217	9704,087	33388,61	315
ga	8	DSP-DD	1054,257	1054,51	1054,82	739
ha	8	DSP-DD	1054,057	1265,762	1433,251	563

EK 4: Genetik Algoritma ile Üretilen Son Jenerasyonlar

Birinci atölyede elde edilen en iyi popülasyon	

Population # 150 Best: 166.475 Avg: 166.94, Worst: 167.64	

Individual #0 [3, 12, 0, 1, 2, 0, 3, 0, 1, 1, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.48
Individual #1 [3, 12, 0, 1, 2, 0, 3, 0, 1, 1, 2, 1, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.49
Individual #2 [3, 12, 0, 1, 2, 4, 3, 0, 1, 1, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.78
Individual #3 [3, 12, 0, 4, 0, 0, 3, 0, 1, 1, 2, 1, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.79
Individual #4 [3, 12, 0, 4, 0, 4, 3, 0, 1, 1, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.84
Individual #5 [3, 12, 0, 4, 2, 0, 3, 0, 1, 1, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 166.97
Individual #6 [3, 12, 0, 1, 2, 0, 2, 0, 3, 1, 2, 1, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 4, 4, 1, 1]	Fitness: 167.01
Individual #7 [3, 12, 0, 4, 0, 0, 3, 0, 1, 1, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 167.11
Individual #8 [3, 12, 0, 1, 2, 0, 3, 0, 1, 3, 2, 3, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 167.31
Individual #9 [3, 12, 0, 1, 0, 4, 3, 0, 3, 3, 2, 1, 1, 3, 4, 0, 2, 3, 3, 2, 2, 0, 0, 0, 3, 0, 1]	Fitness: 167.64

İkinci atölyede elde edilen en iyi popülasyon	

Population # 150 Best: 313.83 Avg: 315.07, Worst: 316.02	

Individual #0 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 2, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 313.83
Individual #1 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 2, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 1, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 313.96
Individual #2 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 2, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 3, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 314.19
Individual #3 [3, 17, 3, 3, 0, 2, 2, 4, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 4, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.28
Individual #4 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.37
Individual #5 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 3, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.43
Individual #6 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 4, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 1, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.51
Individual #7 [3, 17, 3, 3, 0, 2, 2, 4, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 4, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 3, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.54
Individual #8 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 4, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 4, 3, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 315.57
Individual #9 [3, 17, 3, 3, 0, 2, 2, 0, 4, 2, 1, 1, 3, 1, 0, 4, 0, 4, 0, 4, 3, 3, 2, 1, 0, 2, 4, 0, 3, 0, 2, 2, 2, 3, 3, 2, 0, 2, 4, 3, 1, 1, 0, 4, 4, 2, 1, 3, 4, 0, 1, 0]	Fitness: 316.02

Üçüncü atölyede elde edilen en iyi popülasyon

Population # 100 Best: 438.007 Avg: 438.53, Worst: 438.83

Individual #0 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 3, 4, 2, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.01
Individual #1 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 0, 3, 3, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.24
Individual #2 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 3, 4, 2, 0, 3, 2, 3, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.24
Individual #3 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 4, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 0, 3, 3, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.55
Individual #4 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 0, 4, 2, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.6
Individual #5 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 4, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 3, 4, 2, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.61
Individual #6 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 3, 3, 3, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.62
Individual #7 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 3, 3, 3, 0, 3, 2, 3, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.79
Individual #8 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 4, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 0, 4, 2, 0, 3, 2, 0, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.81
Individual #9 [0, 22, 3, 3, 0, 2, 0, 0, 2, 1, 4, 3, 2, 0, 1, 0, 1, 4, 4, 3, 3, 0, 3, 1, 2, 1, 1, 3, 4, 4, 3, 1, 2, 0, 0, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 2, 4, 2, 0, 3, 3, 0, 3, 2, 3, 3, 2, 0, 1, 4, 3, 3, 2, 1, 3, 0, 0, 2] Fitness: 438.83

Dördüncü atölyede elde edilen en iyi popülasyon

Population # 100 Best: 629.593 Avg: 630.61, Worst: 631.21

Individual #0 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 3, 2, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 629.59
Individual #1 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 3, 2, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 2, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 629.75
Individual #2 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 3, 2, 1, 3, 3, 0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 630.21
Individual #3 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 3, 2, 1, 3, 3, 0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 2, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 630.37
Individual #4 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 0, 2, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 630.76
Individual #5 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 0, 2, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 2, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 630.91
Individual #6 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 0, 3, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 631.06
Individual #7 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 0, 3, 1, 3, 3, 0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 631.13
Individual #8 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 3, 3, 3, 1, 3, 3, 0, 1, 2, 3, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 4, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 631.15
Individual #9 [3, 15, 4, 4, 4, 1, 2, 0, 0, 3, 2, 0, 0, 0, 2, 1, 4, 4, 3, 0, 4, 3, 4, 2, 1, 4, 3, 3, 1, 2, 0, 1, 1, 0, 1, 4, 4, 1, 3, 4, 0, 3, 0, 4, 4, 3, 1, 2, 0, 3, 2, 1, 2, 1, 4, 3, 0, 0, 0, 3, 4, 2, 0, 3, 1, 3, 3, 0, 1, 4, 2, 3, 4, 4, 4, 4, 4, 0, 4, 3, 3, 3, 1, 3, 4, 1, 4, 0, 1, 3, 1, 0, 2, 1, 1, 4, 4, 2, 1, 3, 1, 2] Fitness: 631.21

2, 2, 0, 2, 1, 1, 2, 1, 1, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 1, 0, 1, 0, 1, 0, 0, 2, 1, 0, 1, 0, 2, 0, 0, 0, 1, 1, 2, 0, 1, 2, 2, 0, 1, 0, 2, 2, 0, 0] Fitness: 1272.01
Individual #9 [3, 18, 1, 0, 1, 0, 0, 0, 0, 2, 0, 2, 1, 2, 0, 1, 2, 1, 0, 2, 2, 2, 2, 2, 1, 2, 0, 0, 1, 0, 1, 1, 0, 2, 2, 0, 2, 0, 1, 1, 1, 1, 1, 2, 1, 0, 1, 1, 2, 0, 2, 1, 2, 0, 2, 2, 0, 1, 2, 2, 2, 2, 1, 1, 0, 1, 2, 0, 1, 2, 2, 1, 2, 1, 0, 1, 2, 1, 0, 0, 2, 0, 0, 2, 0, 1, 0, 1, 2, 0, 0, 1, 2, 1, 0, 0, 1, 0, 0, 2, 0, 2, 1, 0, 2, 1, 2, 0, 2, 1, 2, 1, 1, 1, 0, 0, 1, 1, 1, 1, 2, 0, 0, 1, 0, 0, 2, 2, 0, 2, 1, 0, 1, 0, 1, 1, 2, 2, 0, 1, 2, 0, 1, 1, 1, 2, 2, 0, 1, 2, 2, 1, 2, 2, 0, 2, 1, 1, 2, 1, 1, 2, 0, 0, 2, 2, 0, 0, 2, 2, 0, 1, 0, 1, 0, 1, 0, 0, 2, 1, 0, 1, 0, 2, 0, 0, 0, 1, 1, 2, 0, 1, 2, 2, 0, 1, 0, 2, 2, 0, 1, 0, 2, 2, 0, 0] Fitness: 1272.08



EK 5: Birinci Atölyedeki Simülasyon Sonuçları

job	weight	arrival_time(s)	departure_time(s)	given_due-date(s)	earliness	tardiness	penalty_earliness	penalty_tardiness	penalty_due-dates	penalty_total
0	1,0	71,0	151,0	174,5	23,5	0,0	5,2	0,0	1,7	6,9
1	0,7	76,0	234,0	301,0	67,0	0,0	3,7	0,0	2,5	6,1
2	0,3	83,0	329,0	410,3	81,3	0,0	1,9	0,0	1,8	3,7
3	0,3	88,0	372,0	542,6	170,6	0,0	2,1	0,0	2,5	4,6
4	0,7	98,0	288,0	370,7	82,7	0,0	3,8	0,0	3,0	6,8
5	1,0	124,0	232,0	248,5	16,5	0,0	5,1	0,0	2,1	7,2
6	0,3	133,0	263,0	505,7	242,7	0,0	2,3	0,0	2,1	4,4
7	0,7	188,0	340,0	374,4	34,4	0,0	3,5	0,0	2,1	5,5
8	0,7	196,0	289,0	366,5	77,5	0,0	3,7	0,0	1,9	5,6
9	0,7	218,0	452,0	429,4	0,0	22,6	0,0	4,2	2,3	6,5
10	0,3	288,0	670,0	733,5	63,5	0,0	1,8	0,0	2,5	4,3
11	0,7	353,0	463,0	525,7	62,7	0,0	3,7	0,0	1,9	5,6
12	0,7	367,0	610,0	657,9	47,9	0,0	3,6	0,0	3,2	6,8
13	1,0	371,0	507,0	495,5	0,0	11,5	0,0	6,1	2,1	8,2
14	0,7	377,0	541,0	561,1	20,1	0,0	3,4	0,0	2,0	5,4
15	1,0	393,0	544,0	565,5	21,5	0,0	5,2	0,0	2,9	8,1
16	0,7	408,0	634,0	592,1	0,0	41,9	0,0	4,3	2,0	6,3
17	0,7	416,0	723,0	654,6	0,0	68,4	0,0	4,5	2,6	7,2
18	1,0	429,0	551,0	550,5	0,0	0,5	0,0	6,0	2,0	8,0
19	0,7	458,0	719,0	637,6	0,0	81,5	0,0	4,6	2,0	6,6
20	1,0	467,0	656,0	615,5	0,0	40,5	0,0	6,5	2,5	9,0
21	0,3	468,0	805,0	995,3	190,3	0,0	2,2	0,0	2,9	5,1
22	1,0	492,0	659,0	652,5	0,0	6,5	0,0	6,1	2,7	8,8
23	0,7	496,0	827,0	825,6	0,0	1,5	0,0	4,0	3,6	7,6
24	1,0	537,0	734,0	706,5	0,0	27,5	0,0	6,3	2,8	9,2

EK 6: Birinci Atölyedeki Çizelgeleme Sonuçları

job_no	operation_no	duration(s)	assigned_machine	start(s)	finish(s)
0	0	4	1	71	75
0	1	28	3	75	103
0	2	6	4	103	109
0	3	9	4	109	118
0	4	1	1	118	119
0	5	2	0	119	121
0	6	3	1	121	124
0	7	4	0	124	128
0	8	10	1	139	149
0	9	2	4	149	151
1	0	22	2	76	98
1	1	2	0	98	100
1	2	13	2	106	119
1	3	4	4	119	123
1	4	8	2	123	131
1	5	2	4	136	138
1	6	24	1	154	178
1	7	6	0	210	216
1	8	17	0	216	233
1	9	1	2	233	234
2	0	8	2	98	106
2	1	8	1	106	114
2	2	9	3	128	137
2	3	13	1	198	211
2	4	1	0	257	258
2	5	8	0	296	304
2	6	1	0	304	305
2	7	12	0	305	317
2	8	5	1	317	322
2	9	7	2	322	329
3	0	11	3	137	148
3	1	6	2	164	170
3	2	21	2	170	191
3	3	1	1	213	214
3	4	13	3	215	228
3	5	8	1	228	236
3	6	24	0	322	346
3	7	3	0	346	349
3	8	11	3	352	363
3	9	2	0	370	372

4	0	25	3	103	128
4	1	8	4	128	136
4	2	13	2	144	157
4	3	20	1	178	198
4	4	17	0	233	250
4	5	5	1	250	255
4	6	2	4	255	257
4	7	16	0	258	274
4	8	1	1	274	275
4	9	13	4	275	288
5	0	2	1	124	126
5	1	13	1	126	139
5	2	5	1	149	154
5	3	11	4	154	165
5	4	9	0	178	187
5	5	1	0	187	188
5	6	22	0	188	210
5	7	9	2	210	219
5	8	1	4	219	220
5	9	10	2	222	232
6	0	11	2	133	144
6	1	7	0	144	151
6	2	7	2	157	164
6	3	14	0	164	178
6	4	1	4	178	179
6	5	5	2	191	196
6	6	2	1	211	213
6	7	3	2	219	222
6	8	9	3	231	240
6	9	23	3	240	263
7	0	7	0	250	257
7	1	5	2	257	262
7	2	10	2	262	272
7	3	1	1	272	273
7	4	6	2	273	279
7	5	7	0	282	289
7	6	7	0	289	296
7	7	17	3	296	313
7	8	3	2	317	320
7	9	19	3	321	340
8	0	9	3	196	205
8	1	1	3	205	206
8	2	9	3	206	215

8	3	1	1	215	216
8	4	3	3	228	231
8	5	19	4	231	250
8	6	13	1	255	268
8	7	5	4	268	273
8	8	8	0	274	282
8	9	7	1	282	289
9	0	5	0	317	322
9	1	12	3	340	352
9	2	18	0	352	370
9	3	14	3	373	387
9	4	2	1	387	389
9	5	13	4	393	406
9	6	1	1	406	407
9	7	19	0	422	441
9	8	2	0	441	443
9	9	7	2	445	452
10	0	29	2	288	317
10	1	4	3	317	321
10	2	6	3	363	369
10	3	8	1	369	377
10	4	4	4	406	410
10	5	3	1	415	418
10	6	13	2	424	437
10	7	10	0	624	634
10	8	10	4	634	644
10	9	11	0	659	670
11	0	10	1	353	363
11	1	1	4	363	364
11	2	6	2	364	370
11	3	8	0	372	380
11	4	6	2	381	387
11	5	6	0	392	398
11	6	3	3	402	405
11	7	13	2	409	422
11	8	10	1	422	432
11	9	13	3	450	463
12	0	4	3	369	373
12	1	15	3	387	402
12	2	12	2	452	464
12	3	13	3	480	493
12	4	18	4	493	511
12	5	25	2	511	536

12	6	11	3	549	560
12	7	14	1	560	574
12	8	14	3	588	602
12	9	2	1	608	610
13	0	6	2	371	377
13	1	16	4	377	393
13	2	8	2	401	409
13	3	6	1	409	415
13	4	11	3	432	443
13	5	7	3	443	450
13	6	8	0	451	459
13	7	12	0	459	471
13	8	4	1	495	499
13	9	5	3	502	507
14	0	4	2	377	381
14	1	11	0	381	392
14	2	9	2	392	401
14	3	4	3	405	409
14	4	1	4	410	411
14	5	21	3	411	432
14	6	8	0	443	451
14	7	1	1	512	513
14	8	4	1	513	517
14	9	18	0	523	541
15	0	24	0	398	422
15	1	2	2	422	424
15	2	10	4	424	434
15	3	9	1	434	443
15	4	11	1	443	454
15	5	21	0	471	492
15	6	6	1	499	505
15	7	7	1	505	512
15	8	19	3	519	538
15	9	6	3	538	544
16	0	6	4	411	417
16	1	5	3	463	468
16	2	12	3	468	480
16	3	8	1	517	525
16	4	3	1	534	537
16	5	5	3	544	549
16	6	11	0	562	573
16	7	10	1	574	584
16	8	16	2	599	615

16	9	5	4	629	634
17	0	6	0	573	579
17	1	6	0	579	585
17	2	6	3	607	613
17	3	3	1	613	616
17	4	27	1	616	643
17	5	24	1	650	674
17	6	10	2	674	684
17	7	8	0	700	708
17	8	12	0	708	720
17	9	3	0	720	723
18	0	8	2	437	445
18	1	15	1	454	469
18	2	26	1	469	495
18	3	2	3	507	509
18	4	1	2	509	510
18	5	10	0	513	523
18	6	9	1	525	534
18	7	5	0	541	546
18	8	1	4	546	547
18	9	4	4	547	551
19	0	2	4	458	460
19	1	9	3	493	502
19	2	19	0	585	604
19	3	3	4	604	607
19	4	2	4	607	609
19	5	1	3	613	614
19	6	12	2	615	627
19	7	8	1	674	682
19	8	13	4	696	709
19	9	10	4	709	719
20	0	21	0	492	513
20	1	16	0	546	562
20	2	10	3	569	579
20	3	2	4	579	581
20	4	18	2	581	599
20	5	5	3	602	607
20	6	11	0	610	621
20	7	3	0	621	624
20	8	7	1	643	650
20	9	6	2	650	656
21	0	13	1	537	550
21	1	1	1	550	551

21	2	10	2	551	561
21	3	14	4	561	575
21	4	11	1	584	595
21	5	19	0	723	742
21	6	19	3	742	761
21	7	9	2	761	770
21	8	10	0	770	780
21	9	10	2	795	805
22	0	10	3	509	519
22	1	13	4	519	532
22	2	6	2	536	542
22	3	9	3	560	569
22	4	9	3	579	588
22	5	13	1	595	608
22	6	20	4	609	629
22	7	2	3	629	631
22	8	15	0	634	649
22	9	10	0	649	659
23	0	5	3	614	619
23	1	30	0	670	700
23	2	18	1	700	718
23	3	7	3	718	725
23	4	19	4	725	744
23	5	11	3	761	772
23	6	23	2	772	795
23	7	23	4	795	818
23	8	5	4	818	823
23	9	4	3	823	827
24	0	6	0	604	610
24	1	5	2	627	632
24	2	9	3	632	641
24	3	16	3	641	657
24	4	20	4	657	677
24	5	8	4	677	685
24	6	11	4	685	696
24	7	5	2	696	701
24	8	11	3	701	712
24	9	22	2	712	734

ÖZGEÇMİŞ

Caner Erden, 1989 yılında İstanbul Bakırköy’de doğdu. İlk, orta ve lise eğitimini İstanbul’da tamamladı. 2005 yılında Akşemsettin Anadolu Lisesi’nden mezun oldu. 2006 yılında başladığı İstanbul Ticaret Üniversitesi Endüstri Mühendisliği Bölümü’nü 2010 yılında bitirdi. Lisans mezuniyetinden sonra İstanbul Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü’nde yüksek lisans eğitimine başladı. 2012 yılında araştırma görevlisi olarak çalışmaya başladı ve yüksek lisans eğitimini 2013 yılında tamamladı. Aynı yıl, Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Bölümü’nde doktora eğitimine başladı. Halen Sakarya Üniversitesi Endüstri Mühendisliği Bölümü’nde araştırma görevlisi olarak görev yapmaktadır.