

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

**ÇOK AMAÇLI ESNEK ATÖLYE TİPİ ÇİZELGELEME
PROBLEMLERİNİN GELİŞTİRİLMİŞ PARÇACIK SÜRÜ
OPTİMİZASYONU İLE ÇÖZÜMÜNE YÖNELİK MODEL
ÖNERİLERİ**

SERKAN KAYA

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

DOKTORA TEZİ

ÇOK AMAÇLI ESNEK ATÖLYE TİPİ ÇİZELGELEME
PROBLEMLERİNİN GELİŞTİRİLMİŞ PARÇACIK SÜRÜ
OPTİMİZASYONU İLE ÇÖZÜMÜNE YÖNELİK MODEL
ÖNERİLERİ

SERKAN KAYA

Prof.Dr. Nilgün FİĞLALİ
Danışman, Kocaeli Üniv.

Doç.Dr. Orhan ENGİN
Jüri Üyesi, Selçuk Üniv.

Yrd.Doç.Dr. Gülşen AYDIN KESKİN
Jüri Üyesi, Kocaeli Üniv.

Yrd.Doç.Dr. Alparslan Serhat DEMİR
Jüri Üyesi, Sakarya Üniv.

Yrd.Doç.Dr. Kasım BAYNAL
Jüri Üyesi, Kocaeli Üniv.



Tezin Savunulduğu Tarih: 14.07.2014

ÖNSÖZ VE TEŞEKKÜR

Çok amaçlı optimizasyon problemleri son yıllarda oldukça ilgi görmektedir. Gerçek hayatta sıkça karşılaşılan üretim planlama, iş çizelgeleme, proje çizelgeleme, araç rotalama, yatırım planlama ve tesis planlama gibi alanlarda birçok başarılı uygulamaları görülmektedir.

Esnek atölye tipi çizelgeleme problemleri hem üretim yönetimi hem de kombinatoriyal optimizasyon alanlarında büyük öneme sahip problem türüdür. Bu tür problemleri yüksek hesaplama karmaşasından dolayı geleneksel optimizasyon yöntemleriyle çözmek, optimal sonuçlarını elde etmek oldukça zordur. Bu tür problemlerin çözümünde sıklıkla meta sezgisel yöntemlere başvurulmaktadır. Çizelgeleme alanının en zor problem sınıfından olan esnek atölye tipi çizelgeleme konusunun çok amaçlı optimizasyonuna yönelik geliştirilen iki modelin literatüre katkısı olması umuduyla,

Çalışmamın başından itibaren gösterdiği emek, destek, samimiyet ve ilgiyle çalışmanın gerçekleşmesi ve ilerlemesini sağlayan değerli danışman hocam Sayın Prof. Dr. Nilgün FIGLALI' ya

Çalışmam sırasında bana yol gösteren ve destekleyen hocam Sayın Prof. Dr. Alpaslan FIGLALI' ya

Çalışmam sırasında fikirleriyle bana yol gösteren ve destekleyen tez izleme komitesindeki hocalarım Sayın Doç. Dr. Orhan ENGİN ve Yrd. Doç. Dr. Gülşen Aydın Keskin'e,

Çalışmalarım sırasında yardımını hiçbir zaman esirgemeyen özellikle bilgisayar yazılımı aşamasında verdiği destekler için hocam Sayın Yrd. Doç. Dr. Nurettin BEŞLİ ve Endüstri Yük. Müh. Ahmet CİHAN'a

Çalışmam sırasında yardımını esirgemeyen, destek ve teşvikleri için bütün çalışma arkadaşlarıma,

Kariyerim boyunca desteğini hiçbir zaman esirgemeyen eşim Saniye KAYA ve her iki evladıma, sonsuz teşekkür ederim.

Temmuz – 2014

Serkan KAYA

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	vi
SİMGELER DİZİNİ VE KISALTMALAR	vii
ÖZET.....	viii
ABSTRACT.....	ix
GİRİŞ.....	1
1. ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMLERİ	7
1.1. Çizelgeleme	9
1.1.1. Çizelgeleme açısından üretim ortamları.....	10
1.1.2. Çizelgelemede kullanılan öncelik kuralları.....	13
1.2. EATÇ Problemleri Matematiksel Modeli	15
1.3. EATÇ Problemleri Literatür Taraması.....	20
2. OPTİMİZASYON PROBLEMLERİ VE ÇÖZÜM YÖNTEMLERİ	33
2.1. Meta Sezgisel Çözüm Yöntemleri.....	34
2.1.1. Genetik algoritma	35
2.1.2. Tavlama benzetimi	37
2.1.3. Tabu arama	39
2.1.4. Dağınık arama metodu	41
2.1.5. Doyumsuz algoritma	42
2.1.6. Karınca kolonileri algoritması.....	43
2.1.7. Yapay bağışıklık sistemi	44
2.1.8. Yapay sinir ağları	46
2.2. Parçacık Sürü Optimizasyonu	46
2.2.1. PSO terimleri ve parametreleri.....	51
2.2.2. Çeşitli PSO yapıları.....	56
2.2.2.1. İkili PSO	56
2.2.2.2. Yakınsama oranı geliştirmeleri	57
2.2.3. PSO literatür taraması	64
3. ÇOK AMAÇLI OPTİMİZASYON	66
3.1.Çok Amaçlı OptimizasyonYöntemleri.....	67
3.1.1. Sabit ağırlıklı amaç fonksiyonu.....	71
3.1.2. Değişken ağırlıklı amaç fonksiyonu.....	72
3.1.3. Vektör hesaplamalı genetik algoritma.....	72
3.1.4. Kuvvet pareto evrimsel algoritması	73
3.1.5. Kuvvet pareto evrimsel algoritması 2	75
3.2. Çok Amaçlı EATÇ Problemleri Literatür Taraması	78
4. GELİŞTİRİLMİŞ PARÇACIK SÜRÜ OPTİMİZASYONU	97
4.1. Çok Amaçlı EATÇ Problemleri Çözüm Yaklaşımları	98
4.1.1. Çok amaçlı PSO	99
4.1.2.Bütünleşik çok amaçlı PSO algoritması.....	104
4.1.3.Hiyerarşik çok amaçlı PSO algoritması	107

4.1.4.Yerel arama algoritması	109
4.1.5.Pareto optimal algoritması	114
4.1.6.Teslim tarihi belirleme	116
4.2. Deneysel Sonuçlar	120
4.2.1. PSO’ da kullanılan parametrelerin belirlenmesi	121
4.2.2. Deneysel sonuçlar	131
5. SONUÇ VE ÖNERİLER	147
KAYNAKLAR	151
EKLER.....	164
KİŞİSEL YAYIN VE ESERLER	170
ÖZGEÇMİŞ	172

ŞEKİLLER DİZİNİ

Şekil 1.1.	Esnek atölye tipi üretim ortamı	8
Şekil 1.2.	Makine ortamları arasındaki ilişkiler	12
Şekil 2.1.	Genetik algoritma adımları.....	37
Şekil 2.2.	Tavlama benzetimi algoritma adımları	39
Şekil 2.3.	Tabu aramaları algoritma adımları	40
Şekil 2.4.	Dağınık arama algoritma adımları.....	42
Şekil 2.5.	Paralel doyumsuz algoritma adımları.....	43
Şekil 2.6.	Karınca kolonileri algoritma adımları	44
Şekil 2.7.	Yapay bağışıklık sistemi algoritma adımları.....	45
Şekil 2.8.	Parçacık sürü optimizasyonu algoritma adımları	48
Şekil 2.9.	Parçacık sürü optimizasyonunda parçacık hareketi	51
Şekil 2.10.	Parçacık hızlarının iki boyutlu örneği	59
Şekil 3.1.	Pareto optimal küme.....	67
Şekil 3.2.	Pareto optimal cephe	69
Şekil 3.3.	(a) Başlangıç popülasyonu, (b) Çözüm kümesi	71
Şekil 3.4.	SPEA2 Ham puanları	76
Şekil 3.5.	SPEA2 Son puanları.....	77
Şekil 3.6.	(a). Arşivdeki yakın bireylerin silinmesi (b) Güncelleme.....	78
Şekil 3.7.	Çok amaçlı EATÇ problemleri yıllar bazında yapılmış çalışmalar	95
Şekil 3.8.	Çok amaçlı EATÇ problemleri amaç fonksiyonu bazında yapılmış çalışmalar.....	96
Şekil 3.9.	Çok amaçlı EATÇ problemlerinde en çok karşılaştırılan çalışmalar	96
Şekil 4.1.	Xp Vektör temsili	101
Şekil 4.2.	Xm Vektör temsili	102
Şekil 4.3.	Örnek problem çözümü.....	102
Şekil 4.4.	BÜÇAPSO Algoritması akış diyagramı.....	106
Şekil 4.5.	HİÇAPSO Algoritması akış diyagramı	108
Şekil 4.6.	YA Akış diyagramı	112
Şekil 4.7.	YA Algoritma uygulaması	113
Şekil 4.8.	YA Algoritması 1.alternatif uygulaması	113
Şekil 4.9.	YA Algoritması 2.alternatif uygulaması	114
Şekil 4.10.	Pareto optimal akış diyagramı.....	116
Şekil 4.11.	W_{max} Faktörü deney tasarım sonuçlarının grafiksel gösterimi.....	125
Şekil 4.12.	W_{min} Faktörü deney tasarımı sonuçlarının grafiksel gösterimi	126
Şekil 4.13.	c_1 Faktörü deney tasarımı sonuçlarının grafiksel gösterimi	128
Şekil 4.14.	c_2 Faktörü deney tasarımı sonuçlarının grafiksel gösterimi.....	130
Şekil 4.15.	4x5x12 Problemi BÜÇAPSO $C_{max}=12$, $W_t=32$, $W_m=8$ çözümü.....	133
Şekil 4.16.	4x5x12 Problemi HİÇAPSO $C_{max}=11$, $W_t=37$, $W_m=9$ çözümü	133
Şekil 4.17.	8x8x27 Problemi BÜÇAPSO $C_{max}=16$, $W_t=73$, $W_m=13$ çözümü.....	135
Şekil 4.18.	8x8x27 Problemi HİÇAPSO $C_{max}=15$, $W_t=75$, $W_m=12$ çözümü	135
Şekil 4.19.	10x7x29 Problemi BÜÇAPSO $C_{max}=12$, $W_t=60$, $W_m=12$ çözümü.....	137
Şekil 4.20.	10x7x29 Problemi HİÇAPSO $C_{max}=13$, $W_t=60$, $W_m=12$ çözümü	137
Şekil 4.21.	10x10x30 Problemi BÜÇAPSO $C_{max}=8$, $W_t=41$, $W_m=7$ çözümü.....	139

Şekil 4.22. 10x10x30 Problemi HİÇAPSO $C_{max}=7$, $W_t=42$, $W_m=6$ çözümü	139
Şekil 4.23. 15x10x56 Problemi BÜÇAPSO $C_{max}=14$, $W_t=91$, $W_m=13$ çözümü	141
Şekil 4.24. 15x10x56 Problemi HİÇAPSO $C_{max}=12$, $W_t=93$, $W_m=12$ çözümü	141
Şekil 4.25. 4x5x12 Problemi BÜÇAPSO ve HİÇAPSO $C_{max}=11$, $\sum C=34$, $\sum T=15$ çözümü	143
Şekil 4.26. 4x5x12 Problemi HİÇAPSO $C_{max}=12$, $\sum C=33$, $\sum T=15$ çözümü	143
Şekil 4.27. 8x8x27 Problemi BÜÇAPSO ve HİÇAPSO $C_{max}=17$, $\sum C=87$, $\sum T=35$ çözümü	144
Şekil 4.28. 10x7x29 Problemi HİÇAPSO $C_{max}=13$, $\sum C=83$, $\sum T=16$ çözümü	145
Şekil 4.29. 10x10x30 Problemi HİÇAPSO $C_{max}=7$, $\sum C=49$, $\sum T=2$ çözümü	145
Şekil 4.30. 15x10x56 Problemi HİÇAPSO $C_{max}=13$, $\sum C=126$, $\sum T=31$ çözümü ...	146

TABLÖLER DİZİNİ

Tablo 3.1. Çok amaçlı EATÇ problemlerinin çözümünde kullanılan yöntemler.....	86
Tablo 3.2. Çok amaçlı EATÇ problemlerinin çözümü için amaç fonksiyonu bazında yapılmış diğer çalışmalar	94
Tablo 4.1. Örnek problem işlem süreleri	100
Tablo 4.2. Örnek probleme ait hesaplanmış teslim tarihleri.....	120
Tablo 4.3. Kullanılan problem boyutları	121
Tablo 4.4. Parametrelere ait faktör ve düzeyleri	123
Tablo 4.5. Sabit tutulan parametrelere ait faktör ve düzeyleri	124
Tablo 4.6. 4x5x12 Problemi W_{max} faktörü için deney tasarımı sonuçları.....	125
Tablo 4.7. 4x5x12 Problemi W_{min} faktörü deney tasarımı sonuçları	127
Tablo 4.8. 4x5x12 problemi c_1 faktörü deney tasarımı sonuçları.....	128
Tablo 4.9. 4x5x12 problemi c_2 faktörü deney tasarımı sonuçları.....	129
Tablo 4.10. Belirlenen PSO parametre değerleri.....	130
Tablo 4.11. Çalışmada ele alınan amaç fonksiyonları.....	131
Tablo 4.12. 4x5x12 Problemine ait sonuçlar	132
Tablo 4.13. 8x8x27 Problemine ait sonuçlar	134
Tablo 4.14. 10x7x29 Problemine ait sonuçlar	136
Tablo 4.15. 10x10x30 Problemine ait sonuçlar	138
Tablo 4.16. 15x10x56 Problemine ait sonuçlar	140
Tablo 4.17. Yeni amaç fonksiyonu tüm problemlere ait sonuçlar.....	142

SİMGELER DİZİNİ VE KISALTMALAR

$\sum C$	: Toplam tamamlanma zamanı
$\sum T$	: Toplam gecikme
c_1	: Parçacık tecrübesi
c_2	: Sürünün tecrübesi
C_{max}	: Maksimum tamamlanma zamanı
d_i	: i. İşin teslim zamanı
G_{best}	: Global en iyi değer
J_i	: i. İş
m	: Toplam makine sayısı
M_j	: j. Makine
n	: Toplam iş sayısı
O_{ik}	: i İşinin k. operasyonu
P_{best}	: Yerel en iyi değer
P_{ikj}	: O_{ik} Operasyonunun j. makine üzerindeki işlem süresi
V_i^k	: k. iterasyonda i. parçacığın hız vektörü
W_j	: j. işin ağırlığı
W_m	: Maksimum İş Yüğü
W_t	: Toplam İş Yüğü
X_i^k	: k. iterasyonda i. parçacığın pozisyon vektörü

Kısaltmalar

ATÇ	: Atölye Tipi Çizelgeleme
BÜÇAPSO	: Bütünleşik Çok Amaçlı Parçacık Sürü Optimizasyonu
DA	: Dağınık Arama
DKA	: Değişken Komşuluk Arama
DSA	: Dal Sınır Algoritması
EA	: Evrimsel Algoritma
EATÇ	: Esnek Atölye Tipi Çizelgeleme
EKİZ	: En Kısa İşlem Zamanı
GA	: Genetik Algoritma
HİÇAPSO	: Hiyerarşik Çok Amaçlı Parçacık Sürü Optimizasyonu
KKA	: Karınca Kolonileri Algoritması
PSO	: Parçacık Sürü Optimizasyonu
SPEA	: Strength Pareto Evolutionary Algorithm (Kuvvet Pareto Evrimsel Algoritması)
SPEA2	: Strength Pareto Evolutionary Algorithm 2 (Kuvvet Pareto Evrimsel Algoritması 2)
TA	: Tabu Aramaları
TB	: Tavlama Benzetimi
YA	: Yerel Arama
YBS	: Yapay Bağışıklık Sistemi
YSA	: Yapay Sinir Ağları

ÇOK AMAÇLI ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMLERİNİN GELİŞTİRİLMİŞ PARÇACIK SÜRÜ OPTİMİZASYONU İLE ÇÖZÜMÜNE YÖNELİK MODEL ÖNERİLERİ

ÖZET

Modern üretim sistemlerinde, etkin üretim planlama faaliyetleri önemlidir. Esnek atölye tipi çizelgeleme (EATÇ) problemi, gerçek yaşam problemleri sınıfında yer alan, klasik atölye tipi çizelgeleme problemlerinin genişletilmiş halidir. EATÇ Problemlerinin çözümünde karşılaşılan en temel problemlerden biri; işlerin hangi sırayla işleneceği, diğeri ise; her işin işlenebileceği birden fazla aday makine olması nedeniyle, bu işlerin hangi makinelerde işleneceğidir. Bu tür problemlerin çözümü için literatürde hiyerarşik ve bütünlük olmak üzere 2 çözüm yaklaşımı vardır. Bütünlük yaklaşımında işlerin sıralanması ve makine atamaları eş zamanlı gerçekleştirilirken, hiyerarşik yaklaşımda sıralama ve atama işlemleri birbirinden bağımsız olarak gerçekleştirilmektedir. EATÇ Problemlerinin çok amaçlı optimize edilmesine yönelik literatürde sınırlı sayıda çalışma vardır. Bu çalışmada EATÇ problemlerinin çok amaçlı çözümüne yönelik bütünlük ve hiyerarşik olmak üzere 2 farklı çok amaçlı model önerisi yapılmıştır. Parçacık sürü optimizasyonu algoritmasına yerel arama sezgiselinin melezlenmesiyle elde edilen her iki model çok amaçlı hale getirilmiştir. Problem öncelikle literatürde en çok çalışılan amaç gruplarından, maksimum tamamlanma zamanı, toplam makine iş yükü ve maksimum makine iş yükü olmak üzere 3 amaçlı çözülmüş ve pareto sonuçlar alınmıştır. Elde edilen sonuçlar literatürde diğer yöntem sonuçlarıyla karşılaştırılarak önerilen modellerin etkinliği gösterilmiştir. Ayrıca, gecikme ölçütünün dâhil edildiği çok amaçlı EATÇ problemleri için sınırlı sayıda çalışma yapılmıştır. Bu nedenle, aynı problem seti, maksimum tamamlanma zamanı, toplam tamamlanma zamanı ve toplam gecikme olmak üzere yeni 3 amaçlı ölçütler için çözülmüştür. Bu çözümlere ait pareto optimal sonuçlar da sunulmuştur.

Anahtar kelimeler: Çok Amaçlı Optimizasyon, Esnek Atölye Tipi Çizelgeleme, Parçacık Sürü Optimizasyonu, Pareto, Yerel Arama.

MODEL PROPOSALS FOR THE SOLUTION OF SCHEDULING PROBLEMS OF MULTI OBJECTIVE FLEXIBLE JOB SHOP WITH IMPROVED PARTICLE SWARM OPTIMIZATION

ABSTRACT

Effective production planning activity is important in modern production systems. Flexible job shop scheduling problem is the extended version of classical job shop scheduling which takes part in real life problems class. One of the major problems in solving the flexible job shop scheduling problems is the order of the processes and the other one is the selection of the machine because one process can be completed by more than one machine. Two solution approaches take part in literature: hierarchical and integrated. In integrated approach ordering the processes and selection of machines is done simultaneously while in hierarchical approach ordering and selection are done independent of each other. There are limited numbers of works concerning multi objective optimization of flexible job shop scheduling problems. Aimed at solving flexible job shop scheduling problems in a multi objective way, two multi objective models were suggested in the study, one integrated and one hierarchical. Both models were rendered as multi-objective models by interbreeding particle swarm optimization algorithm to local search intuitional. Problem was firstly solved aiming at three objectives which belong to the most sought objective groups in the literature and include the makespan, the total workload of all machines and maximum workload of all machines. Also, pareto results were obtained. Results were compared to other method's results and the efficiency of the proposed methods. Also, there has been limited number of work concerning multi objective flexible job shop scheduling problems which include tardiness criteria. Therefore, same set of problems was solved according to makespan, total completion time and total tardiness. Pareto optimal is presented in results.

Keywords: Multi Objective Optimization, Flexible Job Shop Scheduling, Particle Swarm Optimization, Pareto, Local Search.

GİRİŞ

Günümüzün Pazar şartlarındaki tüketici taleplerinin değişkenliği, ürün ömrünün kısa olması, oluşan rekabetten dolayı maliyetleri düşürme gereksinimi, pazar payını koruma çabaları ve tüketici ihtiyaçlarına hızlı cevap verme ihtiyacı v.b. nedenler işletmeleri hızlı karar verme konusunda zorlamaktadır. İşletmeler bir yandan müşteri taleplerindeki hızlı değişimlerden dolayı sıfır stokla çalışmak isterken, diğer taraftan müşteri ihtiyaçlarına hızlı cevap verebilmek için yüksek seviyede elde stok bulundurmak isterler. Bu şekilde birbirleriyle çelişen amaçları aynı anda optimize etmeye çalışmak işletmeler için zorunluluk haline gelmiştir. Üretim ortamındaki belirsizliklerin artması ve amaçların birbiriyle çelişmesi işletmeler için çizelgeleme faaliyetlerinin önemini arttırmıştır. Çizelgeleme problemleri, işletmeler için önemli bir yeri olan ve araştırmacılar için önemini gün geçtikçe arttıran bir çalışma alanı haline gelmiştir.

Çizelgelemenin temel amacı, üretim kaynaklarının en verimli biçimde kullanılması, müşteri taleplerine anında cevap verilmesi, ürün teslimatlarında gecikmelerin yaşanmaması, yarı mamul stoklarının ve fazla mesai çalışmalarının azaltılması olarak düşünülebilir. Bir üretim sisteminde yarı mamul yığılmaları, tezgâhlar arasındaki iş dağılımının üretim akışını sağlayacak şekilde olması, kısacası işletme verimliliğinin artırılması çalışmaları çizelgeleme konusudur.

Çizelgeleme problemleri için bugüne kadar değişik çözüm yöntemleri geliştirilmiştir. Klasik çözüm yöntemleri olarak adlandırılan doğrusal programlama modeli, dal sınır algoritması (DSA) gibi yöntemler öncelikle literatüre kazandırılmıştır. Belirli bir süre sonra bu tür yöntemlerin çizelgeleme problemlerine etkin çözümler üretemedikleri gözlenmiş ve farklı çözüm yöntemleri arayışı başlamıştır. Bu süreçten sonra çizelgeleme problemlerinin çözümlerinde sezgisel yöntemlerin kullanıldığı görülmektedir. Sezgisel yöntemler klasik yöntemlere göre, daha basit hesaplamalar gerektirmesi ve daha kısa sürede çözümler üretilmesinden dolayı daha çok tercih edilir olmuştur.

İşletmelerin üretim ortamları, üretilen ürünün özellikleri, makine yerleşimleri, işletmede uygulanan üretim tiplerine v.b. nedenlere bağlı olarak farklılıklar gösterir. Bundan dolayı farklı üretim ortamlarına bağlı olarak farklı çizelgeleme problemleri ortaya çıkar. Çizelgeleme problemleri içerisinde basit yapısından dolayı tek makine problemleri ilk ele alınan problem türü olmuştur. Atölye tipi üretim ortamında her işin kendine ait rotası bulunmaktadır. Paralel makineli çizelgeleme problemlerinde, gelen işler makinelerden herhangi birinde işlem görebilmektedir. Esnek atölye tipi çizelgeleme (EATÇ) problemleri ise atölye tipi ve paralel makineli üretim ortamlarının birleştirilmiş halidir. Bu ortamların her birinde benzer makinelerin olduğu iş istasyonları yer almakta ve her işin kendi rotası bulunmaktadır. Atölye tipi çizelgeleme (ATÇ) problemlerinde, n iş sayısı, m makine sayısı olmak üzere $(n!)^m$ sayıda mümkün çözüm vardır [1]. Örneğin atölye tipi üretim ortamında 3 iş ve 3 makineden oluşan bir çizelgeleme problemi için $(3!)^3 = 216$ adet mümkün çizelge sayısı vardır. Aynı üretim ortamı için 5 iş 3 makineden oluşan bir çizelgeleme probleminde olası çizelge sayısı $(5!)^3 = 1.728.000$ adet olur. EATÇ problemi, klasik ATÇ probleminin bütün zorluklarını ve karmaşıklığını barındırmanın yanı sıra işlerin makinelere atanması sorunu da söz konusu olduğundan çok daha karmaşık bir problemdir. Gerçek yaşam çizelgeleme problemlerinde ise iş ve makine sayıları arttıkça çizelgeleme problemleri daha zor bir hal almaktadır. İş sayısındaki bir birimlik artış bile olası çizelge sayısında oldukça fazla bir artışa neden olmaktadır.

Çoğu mühendislik problemlerinin çözümünde meta sezgisel yöntemlerin kullanımı son yıllarda önemli bir oranda artmıştır. Matematiksel modelinin çıkarılamadığı, model kurmanın çok maliyetli olduğu optimizasyon problemlerine, meta sezgisel yöntemlerin kolaylıkla uygulanabilir olması, iyi bir hesaplama gücünün olması, bu yöntemlerin uygulamasının kolay olması, tatmin edici sonuçlar alınabilmesi ve bir problem için geliştirilen meta sezgiselin başka bir probleme de uygulanabilir olması gibi nedenlerden dolayı problem çözümünde tercih edilme sebepleri olmuştur.

Parçacık sürü optimizasyonu (PSO) popülasyon temelli bir meta sezgisel optimizasyon yöntemidir. Bazen tek başlarına hiçbir iş yapamayan varlıklar, toplu hareket ettiklerinde çok zekice davranışlar gösterebilmektedir. PSO yöntemi de bir topluluk içindeki bireylerin, en iyi bireyin davranışından ve kendi deneyimlerinden yararlanarak hareket ettikleri bir algoritmadır. Bu yöntemde problem için her bir

özüm, paracak olarak adlandırılır. PSO’ da her bir paracığın hızı, pozisyonu, elindeki bilgiyi diğerk paracıklarla değıştirme kabiliyeti, bir önceki en iyi pozisyonunu hatırlama kabiliyeti ve daha iyi bir karar vermek için bilgiyi kullanma kabiliyeti olan bir özüm sürecidir.

Bu tez alışması ile ok amaçlı EATÇ problemlerinin özümü için meta sezgisel yöntemlerden PSO tabanlı yöntemler geliştirilmiştir. Gerçek üretim sistemlerinde geniş bir uygulama alanına sahip EATÇ problemleri yine pratik hayatta uygulanabilirliği olduğı için ok amaçlı olarak ele alınmıştır. Literatürde yer alan alışmalara bakıldığında diğerk izelgeleme problemlerine göre EATÇ üzerine yapılan alışmalar oldukça kısıtlıdır. Bu problemler üzerine yapılan alışmalar genellikle tek amaçlı olarak özölmeye alışılmıştır. Bu tez alışmasında, ok amaçlı EATÇ problemlerinin meta sezgisel yöntemler ile özümüne yönelik iki model önerisi sunulmuştur. Modelde PSO algoritması tercih edilmiştir. PSO, optimizasyon problemlerine kolay uygulanabilir, yerel ve global arama arasında kurduğı denge ile istenen özömlere hızlı bir şekilde yakınsama özelliğine sahip bir algoritmadır. PSO’nun az sayıda parametreye ihtiyaç duyması algoritmanın diğerk bir üstün yanıdır. Ayrıca PSO’da paracak hareketi hem kendi en iyi hem de sürü içindeki en iyi değere göre ayarlandığından, iyi bir hafıza yeteneğine sahiptir. PSO algoritmasının sayılan bu avantajları ok amaçlı EATÇ problemlerinin özümünde tercih nedeni olmuştur. PSO ile elde edilen özömleri iyileştirmek ve daha kısa sürede sonuca ulaşabilmek için başlangı popölasyonu oluşturulurken en kısa işlem zamanı (EKİZ) öncelik kuralından yararlanılmıştır. Yerel arama (YA) yöntemleri ise kısaca meta sezgisel bir yöntem ile elde edilen özömler üzerinde alışılarak daha iyi özömlere ulaşabilme stratejilerini kapsar. PSO ile elde edilen özömleri daha da iyileştirmek için algoritmaya YA metodu adapte edilmiştir.

EATÇ Problemlerinin özümünde karşılaşılan en temel problemlerden birincisi; işlerin hangi sırayla işleneceğı, diğerk ise; her işin işlenebileceğı birden fazla aday makine olması nedeniyle, bu işlerin hangi makinelerde işleneceğidir. Bu iki temel alt problemin nasıl özöldüğüne bağılı olarak özüm yaklaşımları, hiyerarşik yaklaşım ve bütünleşik yaklaşım olmak üzere ikiye ayrılır. EATÇ Problemlerinin özümü için geliştirilen iki algoritmanın birincisi hiyerarşik özüm yaklaşımı, diğerk bütünleşik özüm yaklaşımıdır. Hiyerarşik özüm yaklaşımında, işlerin makine atamaları ve

işlenme sıralarının belirlenmesi birbirinden bağımsız olarak ayrı ayrı gerçekleştirilmektedir. Makine atamaları ve işlem sıralamasında birbirine müdahale olmamaktadır. Bu yaklaşımı literatürde ilk defa kullanan Brandimarte [2] olmuştur. Bütünleşik çözüm yaklaşımında işlerin sıralanması ve makine atamaları birbirinden bağımsız değildir ve eş zamanlı gerçekleştirilmektedir. Öncelikle işlerin sıralanması gerçekleştirilmekte hemen sonra bu işleri işleyecek makineler belirlenmektedir. Bu çalışmada her iki yaklaşımı kullanan iki algoritma geliştirilmiştir.

Gerçek yaşam üretim sistemlerinde amaçlar farklı olabilir. Bu amaçlar, işi zamanında teslim etmek, ara stok seviyesini düşürmek, işlerin sistemde kalış süresini azaltmak, makine ve işçiyi daha verimli kullanmak, işi istenilen standartlarda yapmak, makine hazırlık sürelerini azaltmak, işçi maliyetlerini azaltmak şeklinde sıralanabilir. İşletmelerin birimleri arasındaki amaçların farklılıkları yüzünden karar vericiler genellikle bu amaçları gerçekleştirmekte zorlanmaktadırlar. Çizelgeleme ölçütü, çizelgeleme problemlerinin amacını ifade eder. Tek amaçlı çizelgelemede, çizelgeleme performansı tek yönüyle ele alınır. Literatürde en çok tek amaçlı çizelgeleme problemleri ile çalışılmıştır. Diğer taraftan birden çok amacın dikkate alındığı çizelgeleme problemlerine olan ilgi zamanla artmıştır. Ancak çok amaçlı çizelgeleme problemlerini çözmek, tek amaçlı çizelgeleme problemlerine göre daha zordur. Amaçlar arasında genelde çelişki vardır. Bir amaç en iyilenmeye çalışılırken diğer bir amaç muhtemelen kötüleşmektedir.

Çok amaçlı problemlerin aynı anda optimize edilmesi çalışmalarında genelde karar verici tarafından amaçlara belirli öncelik değerleri verilerek bu amaçlar tek fonksiyon altında toplanmaya çalışılmıştır. Son yıllardaki gelişmeler, karar verici için birden fazla amacın en iyilenmesini sağlayacak, birden fazla çözümün elde edilip bu çözümler arasından seçim yapabilme ihtiyacı doğmuştur. Literatürde birden çok çözümün elde edilmesi de iki şekilde olmuştur. Bunlardan birincisi; her bir çözümde bir amaç için farklı bir çözüm elde ederek çok amaçlı problemlerde birden fazla çözüm elde edilme yoluna gidilmiştir. İkincisi ise yine her bir çözümde, diğer bir ifadeyle her bir aramada birden fazla amaç için çözüm elde etmek şeklinde olmuştur. İkinci yöntem çizelgeleme problemlerine pareto optimizasyon yönteminin uygulanması ile elde edilmiştir. Çok amaçlı problemlerin çözümünde pareto

optimizasyonu tekniğini kullanarak çeşitli çözüm kümesi elde etme çalışmaları diğer çalışma yöntemlerine göre oldukça azdır.

Bir üretim sisteminde, sipariş çevrim hızının arttırılması oldukça önemlidir. Sipariş çevrim hızının artması ile üretimin daha erken bitirilmesi, yeni siparişlerin üretime daha çabuk alınması mümkün olmaktadır. Aynı zamanda yarı mamul stoklarında da azalma sağlamaktadır. Üretim sistemlerindeki bu amaçlar çizelgelemede toplam tamamlanma zamanı ve maksimum tamamlanma zamanı ölçütleri olarak ifade edilmektedir. İşlerin gecikmesi ile de müşteri memnuniyetlerinde azalma ve işlerin kaybedilmesine sebep olabilmektedir. Tam zamanında üretim yapılan bir sistemde işlerin gecikmesi, montajın gecikmesine dolayısı ile ekstra maliyetlere sebep olabilmektedir. Bu durumda çizelgeleme gecikme ölçütü olarak ifade edilmektedir.

Bu çalışmada çok amaçlı EATÇ problemleri ele alınmıştır. İlk olarak çok amaçlı EATÇ problemlerinin literatürde en çok ele alınan maksimum tamamlanma zamanı (C_{max}), toplam makine iş yükü (W_t) ve maksimum iş yüküne sahip makinenin iş yükü (W_m) olarak 3 amaçlı ele alınmıştır. Hiyerarşik ve bütünleşik çok amaçlı PSO algoritmalarıyla elde edilen çözümler literatürde diğer yazarların çalışmalarıyla karşılaştırılmıştır. EATÇ Problemleri incelendiğinde gecikme ölçütünü dikkate alan çalışmalar oldukça azdır. Özellikle çok amaçlı EATÇ problemlerinde çok daha sınırlı sayıdadır. Bu çalışmada ayrıca çok amaçlı EATÇ problemlerinde daha önceden ele alınmamış olan maksimum tamamlanma zamanı (C_{max}), toplam tamamlanma zamanı ($\sum C$) ve toplam gecikme ($\sum T$) olmak üzere 3 ayrı amaç fonksiyonu ele alınmıştır. Ele alınan amaç fonksiyonlarına ait pareto optimal sonuçlar sunulmuştur.

Çalışmanın 1. bölümünde EATÇ problemlerine giriş yapılmıştır. Genel hatlarıyla EATÇ problemi tanımlandıktan sonra çizelgeleme konusu, çizelgeleme açısından üretim ortamları ve öncelik kuralları genel olarak incelenmiştir. EATÇ problemleri matematiksel modeli sunulmuştur. Bölümün sonunda literatürde tek amaçlı EATÇ problemleri üzerine yapılmış çalışmalar ayrıntılı bir şekilde incelenmiştir.

Çalışmanın 2. bölümünde optimizasyon problemlerine giriş yapılarak bu tür problemlerin çözümünde kullanılan meta sezgisel yöntemler ile ilgili genel bilgiler verilmiştir. Ayrıca çalışmada sunulan her iki algoritmaya temel teşkil eden PSO algoritması hakkında ayrıntılı inceleme yapılmıştır. PSO algoritmasının

optimizasyon problemlerinde kullanımı, çeşitli PSO yapıları ve PSO ile ilgili literatürde yapılmış çalışmalara yer verilmiştir.

Çalışmanın 3. bölümünde çok amaçlı optimizasyon konusu ele alınmıştır. Çok amaçlı optimizasyon problemlerinin çözümünde kullanılan yöntemler, bu yöntemlerden özellikle pareto optimal kavramı ayrıntılı sunulmuş ve Zitzler [3] tarafından sunulan kuvvet pareto evrimsel algoritması 2 (SPEA2) detaylandırılmıştır. Daha sonra çok amaçlı EATÇ problemleri üzerine literatürde yapılmış çalışmalar ayrıntılı olarak incelenmiştir. Çok amaçlı yapılmış literatür çalışmaları amaç fonksiyonları göz önünde bulundurularak incelenmiştir. Bölümün sonunda EATÇ problemleri üzerine literatürde yapılmış çalışmalar ile ilgili elde edilen bulgular sunulmuştur.

4. Bölümde çok amaçlı EATÇ problemlerinin çözümü için geliştirilen iki yöntem sunulmuştur. Öncelikle çok amaçlı PSO yapısının hiyerarşik ve bütünleşik çözüm yaklaşımlarına uygulanması anlatılmıştır. Geliştirilen her iki algoritmaya melezlenen YA algoritması sunulmuştur. Daha sonra geliştirilen her iki yönteme uyarlanan pareto optimal algoritması verildikten sonra, ele alınan problemlerin teslim tarihlerinin literatüre uygun olarak nasıl hesaplandığı anlatılmıştır. Ayrıca deneysel sonuçlara yer verilmiştir. Çok amaçlı EATÇ problemlerinin literatürde yer alan problem boyutları sunulmuştur. Çalışmada önerilen iki algoritmanın parametrelerinin belirlenmesi için deney tasarımı yapılmıştır. Her iki algoritma ile elde edilen sonuçlar literatürde diğer yazarların sonuçları ile karşılaştırılarak önerilen her iki algoritmanın etkinliği ölçülmüştür.

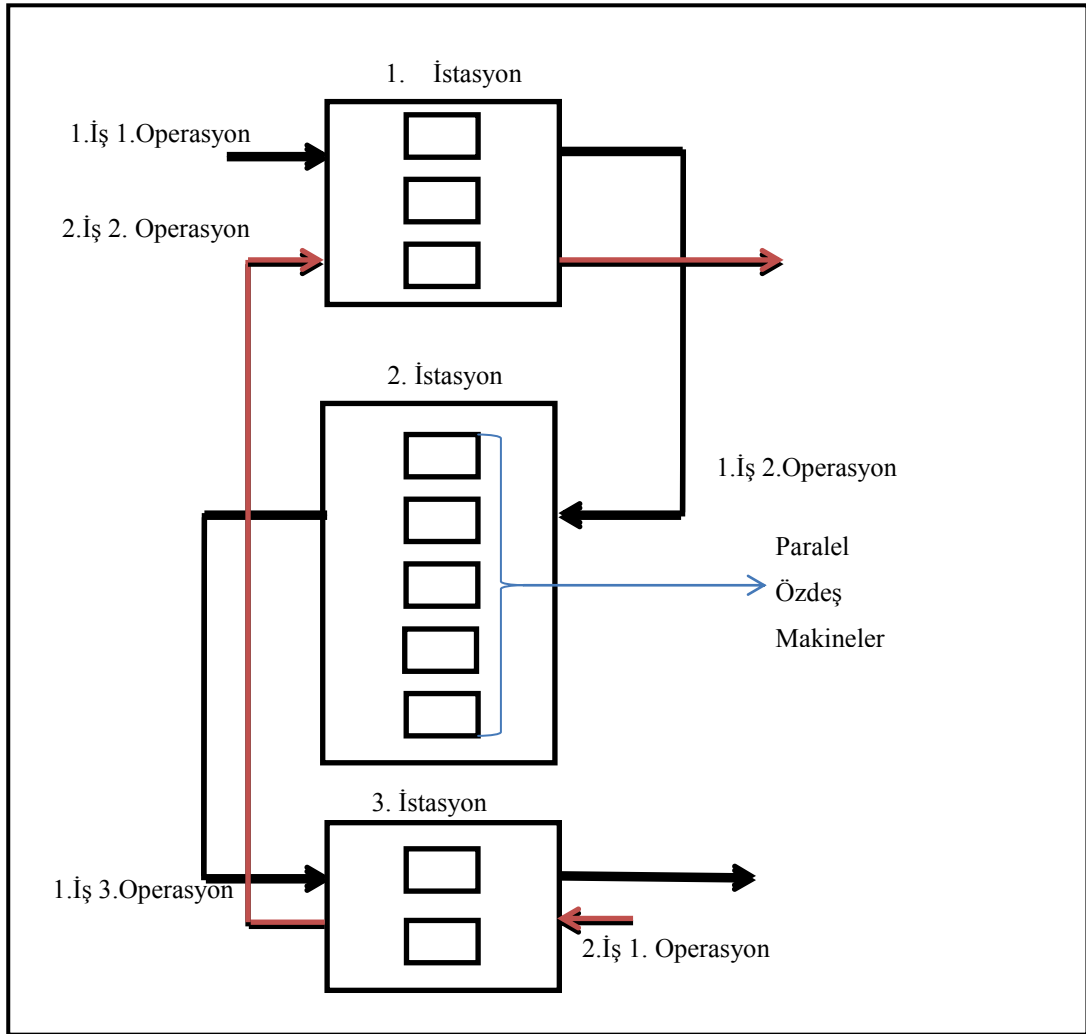
Çalışmanın 5. bölümündeki sonuçlar ve öneriler kısmında, yapılan çalışma değerlendirilmiş ve ileriye yönelik önerilerde bulunulmuştur. Literatürde yer alan problem setlerine ait işlem süreleri ile 4. bölümde hesaplanan bu problemlere ait teslim zamanlarına ait veriler EK-A'da sunulmuştur.

1. ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMLERİ

Son yıllarda teknolojide yaşanan gelişmeler, farklı işlemleri yapabilme kabiliyetine sahip makinelerin işletmelerde üretim hatlarında yer almasını sağlamıştır. Klasik çizelgeleme problemlerinin varsayımlarıyla, günümüz üretim ortamlarının ihtiyaçlarını karşılayacak çizelgeleme problemlerini çözmek artık mümkün değildir. İşletmeler ve akademisyenler, gelişen bu teknolojik makine özelliklerini yansıtabilecek yeni çizelgeleme modellerinden EATÇ modelini sunmuşlar ve bunların çözümleri için çeşitli algoritmalar geliştirmişlerdir. Bu modelde her bir makine farklı zamanlarda farklı işleri işleyebilme özelliğine sahiptir. Yani her bir işin her bir operasyonu farklı sürelerde işi işleyecek makine grubundaki herhangi bir makine üzerinde işlem görebilir. Bu model bünyesinde iki alt problemi barındırmaktadır. Birincisi, işlerin hangi makinelere yükleneceği, ikincisi ise makinelere yüklenen işlerin hangi sırada işleneceğidir. Literatürde bu problem türünün çözümü için, hiyerarşik ve bütünlük olmak üzere iki tür yaklaşım önerilmiştir. Hiyerarşik yaklaşımda, yükleme ve sıralama problemleri ayrı ayrı çözülmektedir. Bütünlük yaklaşımda ise, yükleme ve sıralama problemleri eş zamanlı olarak çözülmektedir.

Klasik ATÇ problemlerinde her bir operasyonun işlem göreceği makine tektir. EATÇ Problemlerinde her bir operasyon belirlenmiş bir makine grubu içerisinde yer alan makinelerden herhangi birinde işlem görebilir. Yani her operasyon için alternatif makine seçimi vardır. Operasyonların tüm makine grubunda işlenme durumu söz konusu ise tam esneklik, tüm makine grubunda işlenme söz konusu değil ise kısmi esneklik söz konusudur. EATÇ Problemleri, belirlenmiş performans kriterini optimize etmek üzere; operasyonların makinelere atanması ve makinelere atanan operasyonların işlenmek üzere sıralanması problemlerinin birleşiminden oluşmuştur. Klasik ATÇ problemlerinden farklı olarak, operasyonların makine grubundaki herhangi bir makineye atanması problemini de içerdiğinden daha karmaşık bir problem türüdür.

EATÇ Problemi, ATÇ ile paralel makine çizelgeleme problemlerinin genelleştirilmiş halidir. Şekil 1.1 'de esnek atölye tipi üretim ortamı verilmiştir. Bu tür üretim ortamlarında, paralel makinelerden oluşan iş istasyonları vardır. Bu iş istasyonlarında işlerin işlenebileceği birden fazla birbirine paralel özdeş makine setleri bulunmaktadır. İşlerin rotaları birbirinden farklıdır ve bu işler iş istasyonlarında bulunan herhangi bir makinede işlenebilirler. Şekil 1.1' de 3 operasyondan oluşan 1 no'lu iş ilk olarak 1. iş istasyonundaki herhangi bir makinede 1. operasyonu işlem gördükten sonra, 2. iş istasyonunda bulunan herhangi bir makinede 2. operasyonu işlem görür. Son olarak 3. operasyon 3. iş istasyonunda bulunan makinelerden herhangi birinde işlem görerek tamamlanır. Aynı şekilde 2 operasyondan oluşan 2 no'lu iş ise öncelikle 3. iş istasyonunda işlenmiş sonra 1. iş istasyonunda bulunan herhangi bir makinede işlenerek tamamlanmıştır.



Şekil 1.1. Esnek atölye tipi üretim ortamı

1.1. Çizelgeleme

Pinedo [4] çizelgelemeyi, belirli bir zaman sürecinde kaynakların işlere tahsis edilmesi ve bir veya daha fazla amacın optimizasyonunu hedefleyen, üretim ve hizmet sektörlerinin birçoğunda düzenli olarak kullanılan bir karar verme süreci olarak tanımlamıştır. Çizelgeleme genel olarak, bir işletmede belirli bir dönem aralığında yapılacak işlerin sıralarının ve sürelerinin belirlenmesi ve bu işlerin yapılabilmesi için kaynakların tahsis edilerek optimum üretim hedeflerini yakalama çalışmaları olarak tanımlanabilir.

Çizelgeleme, üretim planlamaya göre daha ayrıntılıdır. Çizelgeleme, en yakın zamandaki üretim amaçlarına ulaşabilmek için o anki koşulları göz önüne alarak ayrıntılı bir yol ortaya koyar. Üretimde, çizelgeleme fonksiyonu fabrikanın diğer karar alma fonksiyonları ile etkileşim içindedir. Birçok işletme tarafından kullanılan malzeme ihtiyaç planlaması sistemi bu fonksiyonlara bir örnektir. Çizelge oluşturulduktan sonra, gerekli tüm hammadde ve kaynakların belirlenen zamanda kullanılabilir olması şarttır. Tüm işlerin hazırlanma tarihleri; üretim planlama, çizelgeleme sistemi ve malzeme ihtiyaç planlaması sistemince ortaklaşa belirlenir.

Çizelgeleme, bir ya da daha fazla performans ölçütünü optimize etme amacı ile kısıtlı kaynakları faaliyetlere dağıtmaya çalışır. Duruma bağlı olarak, kaynaklar ve faaliyetler birçok farklı durumda ele alınabilir. Kaynaklar montaj hattındaki makineler, bilgisayar sistemindeki merkezi işlem birimi, havaalanlarındaki pist, araba tamirhanesindeki mekanizma v.b. olabilir. Faaliyetler ise üretim sürecindeki birçok işlem, bilgisayar programının uygulanması, havaalanındaki uçakların kalkış ve inişleri, araba tamirhanesindeki araba tamirleri v.b. olabilir. Ayrıca optimize edilecek birçok performans ölçütü bulunmaktadır. Örneğin, amaç en büyük tamamlanma zamanının en küçüklenmesi ya da geciken iş sayısının en küçüklenmesi olabilir [5].

Çizelgeleme problemlerinde genel olarak üç amaç en iyilenmeye çalışılır. Bunlardan birincisi, işlerin teslim tarihidir. Amaç, müşteri siparişlerinin zamanında teslim edilmesi veya geç kalan işin tespit edilmesidir. İkinci amaç, bir işin üretim sistemi içerisinde geçirdiği süreyi en aza indirmektir. Üçüncü temel amaç ise, işletme bünyesinde olan işgücü, makine gibi kaynakların kullanım oranlarının en üst düzeye

çıkarılmasıdır. Bu üç temel amacı aynı anda en iyilemeye çalışmak, tek amacı en iyilemeye çalışmaktan daha zor ve karmaşıktır. Bu üç amaç birbirleriyle çelişebilirler. Örneğin, teslimat gecikmelerini en aza indirebilmek için kapasite arttırımına gitmek kaynak kullanım oranlarını azaltır. Üretim sistemlerinde belirlenen amaç fonksiyonlarına göre oluşturulan iyi bir çizelgeleme sistemi ile işletmeler, yarı mamul ve envanter seviyelerini düşürebilmekte, verimlilikte artış sağlayabilmektedir.

1.1.1. Çizelgeleme açısından üretim ortamları

Çizelgeleme problemleri sınıflandırılırken temel alınan kriterlerden en önemlisi üretim ortamının türüdür. Günümüzde farklı yapı ve özelliklere sahip üretim ortamlarına rastlanmaktadır. Çizelgeleme probleminde ele alınan üretim ortamının türü, problemin kısıt ve kaynakların özelliklerini etkileyecektir. Üretim ortamının çeşitliliğine göre çizelgeleme problemleri de çeşitlilik göstermektedir. Çizelgeleme problemi, tek makineli birden fazla işin sıralandığı problemlerden, çok makineli karmaşık atölye yapısının olduğu boyutlara kadar farklılıklar gösterebilmektedir. Akış tipi üretim ile atölye tipi üretim tipleri üretim sistemlerinin iki uç noktalarını oluşturmaktadır. Diğer üretim tipleri bu iki uç nokta arasında çeşitli bileşimlerden oluşmaktadır.

Gerçek yaşamdaki üretim sistemlerinde bulunan müşteri talepleri, pazar gerekleri, tesis imkânları, ürünün bileşenleri, makine özellikleri v.b. durumlardan dolayı, çizelgeleme ortamlarını net olarak birbirinden ayırmak ve sınıflandırmak zordur. Gerçek yaşamda, literatürde yer alan üretim ortamlarının birebir eşdeğeri yerine birkaçının birleşiminden oluşmuş çizelgeleme ortamları vardır. Literatürde yapılan çalışmalara ve gerçek yaşam uygulamalarına bakıldığında en çok karşılaşılan çizelgeleme ortamları aşağıdaki gibidir.

Tek makineli çizelgeleme ortamı, sistemde sadece bir makinenin bulunduğu durumu ifade eder ve diğer çizelgeleme problemlerine göre en kolay problem sınıfı olarak tanımlanır. Tek makineli ortamda kullanılan yöntemler aynı zamanda daha karmaşık problemlerin çözümüne de ışık tutar. Literatürde tek makineli çizelgeleme problemleri, çeşitli özel koşullar ve kısıtlar altında farklı amaç fonksiyonları için çözülürler.

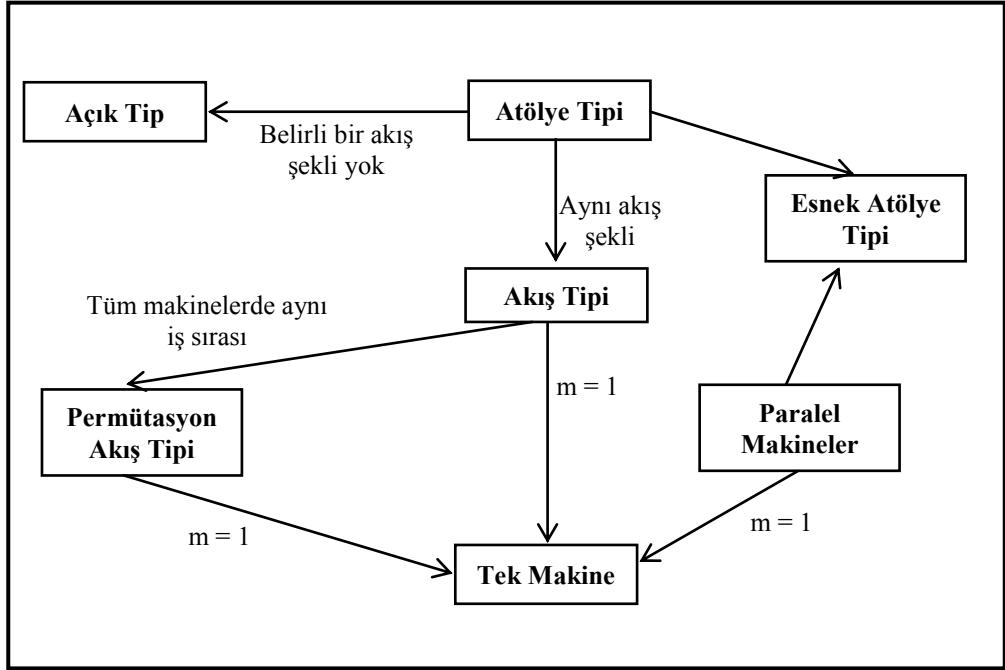
Tek makine çizelgeleme yapısı, n sayıda işin bir makine üzerinde çizelgelendiği durumdur. Her iş aynı makine üzerinde işlem görmektedir. Bu durumda n sayıda iş olmak üzere, tek makine çizelgeleme ortamında $n!$ adet farklı sıralama alternatifi bulunmaktadır.

İşlerin sistemde mevcut olan m adet makine grubundan herhangi birinde yapılabileceği çizelgeleme problemleri paralel makineli çizelgeleme problemlerini oluşturur. Paralel makineli çizelgeleme problemleri aslında tek makine çizelgeleme problemlerinin genelleştirilmiş halidir. Paralel makinelerde işler her bir makinede değişik sürelerde işlenebilir. Bazı paralel makine çizelgeleme problemlerinde işlerin tamamı tüm makine setlerinde işlem görebilirken, bazı problemlerde işler m adet makinenin alt seti olan daha az makine grubunda işlem görebilir.

Akış tipi çizelgeleme ortamı, birbirine seri bağlı m adet makinenin bulunduğu ve her işin aynı rotayı izleyerek makinelerin her birinde işlem gördüğü çizelgeleme ortamıdır. Makinede tamamlanan bir iş sisteme seri olarak bağlı diğer bir makinede işlem görür. Bu sistemlerde tek bir ürün veya birbirine benzeyen ürün grubu vardır. İş istasyonları, ürünün üretimi için gereken işlem sıralarına göre oluşturulmuştur. Özel amaçlı makineler, üretim akışına göre sıralanmıştır ve iş akışı düzgündür. Düzgün iş akışı yarı mamul yığılmalarını azaltmakta dolayısıyla çizelgeleme çalışmaları büyük ölçüde kolay olmaktadır.

ATÇ ortamlarında n sayıda iş m adet makine üzerinde işlem görür ve her işin kendine ait önceden belirlenmiş bir rotası vardır. Bazı işler bazı makinelerde birden fazla işlem görebilirken bazı işler ise bazı makinelerde işlem görmeyebilir. ATÇ probleminde amaç her bir makinede işlerin işlem sıralarının bulunmasıdır. Her bir makinede işlerin $n!$ sayıda farklı sıralama alternatifi olduğu düşünüldüğünde sistemdeki m adet makine için toplamda $(n!)^m$ sayıda alternatif vardır [1]. Bu nedenle az sayıda iş ve makinelerin bulunduğu ATÇ problemlerinin çözümü bile zordur.

Atölye tipi üretim ortamında ürün çeşitliliği çoktur. Makineler genel amaçlı olup, kullanım oranı yüksektir. İşler üretim aşamasında yavaş yavaş geçer dolayısıyla üretim içi ara stoklar fazladır. Genel amaçlı makineleri kullanabilecek vasıflı eleman ihtiyacı vardır. Atölye tipi ortamlarda üretim çizelgeleme faaliyetleri zor ve pahalıdır. Makine ortamları arasındaki ilişki Şekil 1.2' de özetlenmiştir.



Şekil 1.2. Makine ortamları arasındaki ilişkiler

Bu bölümde buraya kadar çizelgeleme çeşitleri üretim ortamlarına göre sınıflandırılmıştır. Çizelgeleme problemleri literatürde statik ve dinamik çizelgeleme olmak üzere iki sınıfa da ayrılmaktadır.

Çizelgeleme periyodu boyunca iş sayısının belirli olması, atölye ortamında hiçbir belirsizlik durumunun olmaması statik çizelgeleme olarak tanımlanır. Diğer bir ifadeyle, çizelgelenecek işler zamanla değişiklik göstermiyorsa statik durum söz konusudur. Çizelgeleme ortamında herhangi bir değişiklik olmaması durumundan dolayı statik modeller dinamik modellere göre daha kolay kontrol edilebilir bir yapıdadır. Bundan dolayı çizelgeleme alanında daha çok statik durumlar ele alınmıştır.

Statik çizelgeleme olarak bilinen geleneksel çizelgeleme yaklaşımı, problemleri çözebilse de genelde optimal olmayan sonuç verir ve gerçekçi olmayan kısıtlar içerdiğinden pratiğe geçirilmesi, gerçek hayatta uygulanması zordur. Bunun nedeni çok sayıda ürün ve sürece sahip gerçek üretim sistemlerinin kompleks ve dinamik yapıda olmalarıdır. Bu dinamiklik, yeni ürünlerin gelmesi, sırada olan işlerin ertelenmesi, bazı işlerin zaman içinde daha çok ya da daha az önemli hale gelmeleri, teknolojik donanımda arızanın meydana gelmesi, sevkiyattaki gecikmelerden dolayı

kaynakların geçici olarak mevcut olmaması, hammaddelerin tükenmesi, çalışanların hastalanması gibi nedenlerden kaynaklanabilir[6].

Dinamik çizelgelemede işlem süreleri değişkenlik gösterir, sistemde tahmin edilemeyen olayların gerçekleşme olasılığı vardır. Dinamik sistemlerde arızalanmalar, beklenmeyen acil işler, kalite problemleri, malzeme tedariki gibi beklenmeyen durumlar ortaya çıkabilir. Bu tür durumlar en etkin çözümü bir anda etkin olmayan bir çözüm haline getirebilir.

Dinamik ortamda, çizelgeleme periyodu boyunca iş sayılarında artışlar olabilir. Bu durumda atölye ortamı dinamik olarak adlandırılır. Gerçek yaşamda çizelgeleme problemleri stokastik ve dinamiktir. Ancak çoğu problem, statik ve deterministik olarak tasarlanır. Bunun nedeni, çeşitli varsayımlarla basitleştirilen statik ve deterministik modellerin iyi anlaşılması, gerçek zamanlı modellerin ortaya konmasının ilk adımını oluşturmasıdır. Gerçek hayatta üretim sistemlerinin çoğu, kaçınılmayan ve beklenmedik olayların meydana gelerek planlanan çizelgelerde değişikliğe neden olduğu dinamik çevrelerde faaliyet gösterdiklerinden çizelgeleme problemlerini statik olarak ele almak gerçekçi olmayacaktır. Dinamik, değişken çevrelerde faaliyet gösteren işletmelere etkin ve faydalı çizelgeler sunabilmek için çizelgelerin dinamik olarak ele alınması gerekmektedir [7].

1.1.2. Çizelgelemede kullanılan öncelik kuralları

Çizelgeleme problemlerinde sıralama yapılırken göz önünde bulundurulan basit kurallar öncelik kuralları olarak adlandırılırlar. Bu kurallara göre işlerin makinede sıralanması yapılır [8]. Öncelik kuralları çizelgeleme problemlerine kolaylıkla uygulanabilmeleri ve uzun hesaplama süresi gerektirmediğinden dolayı tercih edilirler. Küçük boyutlu problemler için tatmin edici sonuçlar vermelerine rağmen, problem boyutu büyüdükçe bu kurallar iyi sonuç vermezler. Bu kuralların avantajı, anlaşılması ve uygulanmasının kolay olmasıdır. Dezavantajı ise, en iyi çözümü garanti etmemeleridir. Farklı çizelgeleme amaçları için farklı kurallar geliştirilmiştir. Bunlardan bazıları aşağıda verilmiştir.

İlk gelen ilk çıkar: Bu kurala göre serbest kalan ilk iş, makineye yüklenecek ilk iştir. Diğer bir ifadeyle, işler geliş sıralarına göre makinelere yüklenmeleri yapılır.

Maksimum akış zamanı ve işlerin bekleme sürelerini minimize etme amaçlı problemlerde tercih edilirler.

Son gelen ilk çıkar: Sisteme son gelen işin makineye ilk sırada yükleneceği kuraldır. Bu kuralda son işten başlayarak ilk çıkan işe kadar sıralama yapılır.

En kısa işlem süresi: Bu kuralda, her işe ait işlemler içerisinde en kısa işlem zamanına sahip olan işlem seçilir ve makineye ilk yüklenir. Makineye yüklenen iş, diğer işler arasından çıkartılarak, kalan işlemler için aynı adımlar takip edilir. Ortalama akış zamanı amaçlı çizelgeleme problemlerinde tercih edilirler.

Ağırlıklandırılmış en kısa işlem süresi önce kuralı: Bu kuralda işler karar verici tarafından önem derecelerine göre ağırlıklandırılırlar. Ağırlık değeri en yüksek olan iş ilk gönderilme kuralına dayanır. Ağırlıklı tamamlanma zamanını en küçükmek için tercih edilirler.

En uzun işlem süresi: En uzun işlem süresine sahip olan işin makineye ilk yüklendiği kuraldır. Paralel makine çizelgeleme problemlerinde toplam işlem süresini minimize etme amaçlı problemlerde daha çok tercih edilirler.

En erken teslim süresi: Bir makine boşaldığı zaman en erken teslim süresine sahip olan işin gönderildiği kuraldır. Bu kuralda işler teslim tarihi en erkenden başlanarak sıralanır ve bu sırada makine yüklemeleri yapılır. Maksimum gecikmeyi en aza indirme problemlerinde tercih edilirler.

En az kalan işlem sayısı: Bu kurala göre her iş için, işlem sayısı en az olan iş ilk olarak makineye yüklemesi yapılır. Öncelikle her işe ait toplam işlem sayısı tespit edilir. Her işe ait işlem sayısı küçükten büyüğe doğru sıralanır. En az sayıda işleme sahip olan işin ilk operasyonu makineye yüklemesi yapılır.

En çok kalan işlem sayısı: Yüklenecek işler arasında en çok işleme sahip olan işin ilk yüklendiği kuraldır. En az kalan işlem sayısı kuralının bir nevi tersidir. Kapasite kullanım oranlarının yükseltilmeye çalışıldığı problemlerde tercih edilirler.

En kısa kalan işlem süresi: Herhangi bir zaman diliminde her bir işin kalan toplam işlem zamanlarından en kısa süreye sahip olanın makineye yüklendiği kuraldır.

Tamamlanma zamanı amaç fonksiyonu çizelgeleme problemlerinde tercih edilirler. Bu kuralda öncelikle bütün işlerin toplam işlem zamanları tespit edilir. İlk sıralamada aslında en kısa işlem zamanı kuralına göre sıralanır. İlk işlem sıralandıktan sonra kalan işler içinden küçükten büyüğe doğru sıralama yapılarak işlemler sıralanır.

En uzun kalan işlem süresi: Sistemde kalan işlerin, işlem sürelerinin büyükten küçüğe sıralanarak çizelgelemenin oluşturulduğu kuraldır. Kalan işlem süresi en uzun işlemler ilk önce bitirilerek kapasite kullanım oranı arttırılmaya çalışılır.

Rassal seçim: Çizelgelenecek işler kümesinden rastsal olarak seçimin yapıldığı kuraldır. Diğer bir uygulama şekli, çizelgelenecek iş sayısı kadar rassal üretilen sayıların küçükten büyüğe doğru sıralanarak karşılık gelen işin çizelgelenmesi şeklinde gerçekleştirilebilir.

1.2. EATÇ Problemleri Matematiksel Modeli

Atölye tipi üretim ortamında, her işin önceden belirlenmiş işlem rotası vardır. İşler belirlenmiş rotalardaki makinelerde işlem görürler. Paralel makine üretim ortamında, birbirine paralel şekilde bağlı m adet makine vardır. İşlem görecektir iş bu paralel makinelerden herhangi birinde işlenebilir. Esnek atölye tipi üretim, atölye tipi ve paralel makine üretim tiplerinin birleştirilmiş halidir. Esnek atölye tipi üretimde işlerin rotaları birbirlerinden farklıdır ve her iş en az bir operasyondan oluşmaktadır. Bu operasyonlar, birbirine paralel makine setlerinden herhangi birinde işlem görürler. Her bir makinedeki her bir operasyonun işlem süresi birbirinden farklıdır.

Genel olarak EATÇ Problemleri; toplam ve kısmi esnekliğe sahip olmak üzere iki gruba ayrılırlar. Toplam esneklikte her bir operasyon tüm makineler üzerinde işleme tabi tutulabilirler. Kısmi esneklik durumunda ise işlerin bir kısmı sadece bir grup makine üzerinde işleme tabi tutulabilirler. Kacem ve diğ. [9-11], aynı makine ve operasyon sayısı için kısmi EATÇ problemlerinin çözümünün toplam EATÇ problemleri çözümüne göre daha zor olduğunu belirtmişlerdir.

EATÇ Problemi, operasyonların aday makinelerden herhangi birisine atanması ve bu makine grubunda işlerin sıralanması alt problemlerinden oluşmaktadır. Probleme ait matematiksel model aşağıda sunulmuştur. Model için, Fattahi [12] tarafından sunulan

karışık tam sayılı doğrusal programlama modeli, Eren [13] tarafından sunulan tam sayılı doğrusal programlama çalışmalarından faydalanılmıştır. Matematiksel modelle ilgili varsayımlar, parametreler ve değişkenler ile ilgili gerekli açıklamalar yapıldıktan sonra model sunulmuştur. EATÇ Probleminde varsayımlar ve başlangıç koşulları aşağıdaki gibidir:

- Problemde her bir iş için proses alternatifler söz konusudur.
- Her bir prodesteki operasyonlar ve bunların sıraları belirlidir.
- Her bir iş farklı operasyonlardan oluşabilir.
- Her bir operasyon için en az bir tane olmak üzere birden fazla makine alternatifi vardır.
- Her bir operasyonun bütün makine alternatiflerindeki işlem süreleri farklıdır ve önceden belirlidir.
- İşlere ait operasyonlar tüm makine grubunda işlenebiliyorsa tam esneklik, aksi halde kısmi esneklik söz konusudur.
- Bir operasyonun bir makine üzerinde işlenemeyeceği durumda, o makine için ilgili operasyonun işlem zamanı çok büyük bir değer atanır.
- Tüm makineler başlangıçta boşta.
- Tüm işler başlangıçta işlenmeye hazırdır.
- Bir makine aynı anda sadece bir operasyon işleyebilir.
- Her operasyon kesintiye uğramadan tamamlanmak zorundadır.
- Operasyonların hazırlık zamanları makinelerden bağımsızdır ve işlem zamanının içindedir.
- Makineler arası taşıma süreleri ihmal edilir.

Parametreler:

n: Toplam iş sayısı

m: Toplam makine sayısı

i: İşlerin indisi $i = 1, 2, \dots, n$;

j: Makine indisi, $j = 1, 2, \dots, m$;

k : Operasyonların indisi $k = 1, 2, \dots, k_i$
 h : Öncelik indisi $h = 1, 2, \dots, h_i$
 J_i : i . iş
 k_i : J_i işine ait toplam operasyonların sayısı
 O_{ik} : i işinin k . operasyonu
 M_j : j . makine
 P_{ikj} : O_{ik} operasyonunun j . makine üzerindeki işlem süresi
 t_{ik} : O_{ik} operasyonunun başlama zamanı
 d_i : i . işin teslim tarihi
 T_{mjh} : h . önceliğe sahip j . makinenin başlama zamanı
 U : m boyutundaki makine seti
 U_{ik} : O_{ik} operasyonu için kullanılabilecek makine seti
 P_{sik} : O_{ik} operasyonun makine seçildikten sonraki işlem süresi
 C_{ik} : O_{ik} operasyonunun tamamlanma zamanı
 L_i : i işinin gecikme süresi
 T_i : i işinin geç bitmesi
 $C_{\max}$: En son biten operasyonunun tamamlanma zamanı
 $\sum C$: Toplam tamamlanma zamanı
 $\sum T$: Toplam gecikme
 W_j : M_j makinesinin iş yükü (toplam işlem süresi)
 W_t : M_m makine setinin toplam iş yükü (toplam işlem süresi)

a_{ijk} : O_{ik} operasyonunu işleyebilen U_{ijk} makine seti

Karar Değişkenleri:

$$a_{ijk} = \begin{cases} 1 & \text{Oik işi için j makinesi seçilecekse} \\ 0 & \text{Aksi halde} \end{cases}$$

$$y_{jik} = \begin{cases} 1 & \text{Oik operasyonu için j makinesi seçilecekse} \\ 0 & \text{Aksi halde} \end{cases}$$

$$X_{jikh} = \begin{cases} 1 & \text{Oik operasyonun l sırada işlenmesi için j makinesi seçilecekse} \\ 0 & \text{Aksi halde} \end{cases}$$

EATÇ Problemi modeli aşağıdaki gibidir:

Amaç Fonksiyonları:

$$\min C_{\max} = \max_i \max_k \{C_{ik}\} \quad (1.1)$$

$$\min W_M = \max_j \{W_j\} \quad (1.2)$$

$$\min W_T = \sum_{j=1}^m W_j \quad (1.3)$$

$$\min \sum C = \sum_{i=1}^n C_i \quad (1.4)$$

$$\min \sum T = \sum_{i=1}^n T_i \quad (1.5)$$

Kısıtlar:

$$\sum_j y_{jik} \cdot P_{jik} = P_{sik} \quad i=1,2,\dots,n; k=1,2,\dots,k_j \quad (1.6)$$

$$t_{ik} + P_{sik} \leq t_{i,k+1} \quad i=1,2,\dots,n; h=1,2,\dots,h_i-1 \quad (1.7)$$

$$T_{m_{j,h}} + P_{sik} \cdot x_{jikh} \leq T_{m_{j,h+1}} \quad (1.8)$$

$$T_{m_{j,h}} \leq t_{i,k} + (1 - x_{j,i,k,h}) \cdot L \quad (1.9)$$

$$T_{m_{j,h}} + (1 - x_{j,i,k,h}) \cdot L \quad (1.10)$$

$$Y_{j,i,k} \leq a_{j,i,k} \quad (1.11)$$

$$\sum_k \sum_h x_{j,i,k,h} = 1 \quad (1.12)$$

$$\sum_j y_{j,i,k} = 1 \quad (1.13)$$

$$\sum_h x_{j,i,k,h} = y_{j,i,k} \quad (1.14)$$

$$T_i \geq C_i - d_i \quad (1.15)$$

$$t_{i,k} \geq 0 \quad (1.16)$$

$$p_{s_{i,k}} \geq 0 \quad (1.17)$$

$$T_{m_{j,h}} \geq 0 \quad (1.18)$$

$$x_{j,i,k,h} \in \{0,1\} \quad (1.19)$$

$$y_{j,i,k} \in \{0,1\} \quad (1.20)$$

$$j=1,2,\dots,m; i=1,2,\dots,n; k=1,2,\dots,k_j : h=1,2,\dots,h_i$$

Amaç fonksiyonu olarak; Denklem (1.1) $C_{\max}$, son işin sistemi terk ettiği süreyi, Denklem (1.2) W_m , en yüklü durumdaki makinenin iş yükünü vermektedir. Denklem (1.3) ise W_t bütün makinelerdeki toplam iş yükünü hesaplar. Denklem (1.2) ve (1.3) göz önüne alındığında amaç, tüm makinelerin iş yükünü dengelemektir. Denklem (1.4) $\sum C$, tüm işlerin toplam tamamlanma zamanını, Denklem (1.5) $\sum T$, sistemde teslim tarihinden sonra biten işlerin toplam gecikmesini vermektedir.

Denklem (1.6), O_{ik} operasyonunun işlenmek üzere seçilen k makinesindeki işlem süresini belirleyen kısıttır. Denklem (1.7) her işe ait operasyonlar arasındaki öncelik kısıttır. Denklem (1.8) her makineye aynı anda sadece bir operasyon atanacağını belirleyen kısıttır. Denklem (1.9), O_{ik} operasyonunun atandığı k makinesi boşa çıktığında işleme tabi tutulacağını belirler. Denklem (1.10), O_{ik} operasyonunun kendinden önce gelen aynı işe ait operasyonun $O_{i,k-1}$ tamamlandıktan sonra işleme tabi tutulacağını belirler. Denklem (1.11) operasyonun işlenebilecek makineler tarafından işlenmesini sağlarken, Denklem (1.12) operasyonun makineye atanmasını ve işlerin makine üzerindeki sırasını belirlemektedir. Denklem (1.13) operasyonun bir makinede işleneceğini sınırlayan kısıttır. Denklem (1.14) her işe ait operasyonların öncelik kısıtın da işlenmesini sağlar. Denklem (1.15) i . işin gecikmesini ifade eder.

1.3. EATÇ Problemleri Literatür Taraması

Bir konu üzerinde literatür taramasının amacı, araştırma konusuyla ilgili o ana kadar yapılmış çalışmaları belirlemek ve incelemektir. Bu inceleme araştırmacının, konuyu daha iyi anlamasının yanı sıra daha önceden yapılmış bir çalışmayı tekrarlamasından korur. Literatür taraması araştırmacıya, konu ile ilgili temel sađlar ve araştırma konusunun eksik kalan kısımlarının tespit edilmesinde vazgeçilmez bir gerekliliktir. Bu bölümde EATÇ problemlerinin tek amaç üzerine literatürde yapılmış çalışmalar incelenmiştir.

Meto [14], çalışmasında EATÇ problemlerinin çözümünde daha hızlı ve iyi sonuçlar alan bir meta-sezgisel geliştirmiştir. Geliştirilen algoritma, amaç olarak $C_{\max}$ 'ı alan, hibrid bir meta sezgisel algoritmadır. Yazarın algoritması, problem rassal deđişim sezgiseli, genetik algoritma (GA) ve lokal iyileştirme aşaması olmak üzere üç temel aşamadan oluşmaktadır. Problemde rassal deđişim sezgiseli, GA'da kaliteli ilk nesil çözümlerinin bulunması için kullanılmıştır. İlk nesil oluşturulurken rassal deđişim sezgiseli ile operasyonların işlenme zamanları üzerinde eksi ve artı yönde yüzdesel deđişiklikler yapılarak seçtiđi öncelik bazlı kurallar yardımıyla çizelgeler oluşturmuştur. GA ile iyileştirilen çözüm nesillerini son aşamada lokal olarak daha da iyileştirmeye çalışmıştır. Yazar, algoritmasını literatürdeki test problemleri üzerinde denemiş ve etkin bir şekilde çalıştığını göstermiştir.

Roshanaei [15], EATÇ problemi matematiksel modelini sunmuştur. Yazar çalışmasında, üç sıralama ve bir pozisyon tabanlı tam sayılı doğrusal programlama modeli (MILPs) geliştirmiştir. Büyük boyutlu problemlerin çözümü için yapay bağışıklık sistemi (YBS) ve tavlama benzetimi (TB) algoritmalarından melez bir algoritma (AISA) geliştirmiştir. Amaç fonksiyonu olarak $C_{\max}$ 'ı ele almıştır. Yazar her iki çalışmasını da literatürde yer alan problem setleriyle karşılaştırmış ve etkin sonuçlar aldığını göstermiştir.

Ak [16] savunma sanayiine üretim yapan, esnek atölye tipi üretim ortamına sahip bir üretim tesisinde $C_{\max}$ 'ı en küçükleyen çizelgeleme problemini çalışmıştır. Parçaların üretiminde CNC tezgâhlar, frezeler, matkaplar, broşlar v.b. makineler kullanılmakta ve toplamda 23 makine bulunmaktadır. Yazar, problemi GA çözüm yöntemine adapte ederek çözmüştür. Çalışmada ele alınan 3 parçanın mevcut üretim

zamanından daha kısa üretim zamanına sahip olan çizelge elde edilerek firmanın üretim zamanlarında iyileştirme yapılmıştır. Elde edilen çizelgeden, makine boş beklemleri en aza indirilecek şekilde makine sayılarında düzenlemeye gidilmiş, bu duruma göre yeni bir fabrika yerleşim düzeni hazırlanmıştır. Yeni fabrika yerleşim düzenine göre makineler arası parça taşınmalarının en az olması sağlanarak, zaman kayıpları minimize edilmiştir.

Kafile aktarımı, ürün kfilesinin küçük partilere (aktarma kfilelerine) bölerek kfilenin tamamının bir aşamadaki işlemi tamamlamadan bir sonraki aşamaya gönderilmesi yoluyla C_{max} 'ın minimize edilmesini amaçlayan bir yaklaşımdır. Demir [17] çalışmasında, EATÇ problemine dayanan ve kafile aktarımı stratejisinin uygulandığı, otomotiv yan sanayinde faaliyet gösteren bir firmanın pres atölyesinin çizelgelenmesi problemini ele almıştır. Bu doğrultuda problemin önce matematiksel modelini oluşturmuştur. Önerilen model, literatürde bu problem için önerilen bir başka modelle karşılaştırılmış ve daha iyi sonuçlar alındığını göstermiştir. Ancak problemin NP-zor yapısı nedeni ile ancak küçük boyutlu problemler için sonuç alabilmiştir. Büyük boyutlu problemlerin çözümü için bir GA geliştirmiştir. Geliştirilen bu algoritmayla, literatürde sıkça kullanılan test problemlerini çözmüş ve sonuçları daha önce önerilen algoritmalarla karşılaştırarak şimdiye kadar elde edilen en iyi sonuçlara ulaştığını göstermiştir.

Yang ve diğ. [18], n sayıda işin, tek makine üzerinde bakım aktivitelerini göz önünde bulundurarak, C_{max} minimizasyonu amaçlı problemi çalışmışlardır. Makine, bakım için önceden tanımlanmış bir zaman diliminde durdurulmalıdır. Problem, sürdürülebilir ve sürdürülemez durumlar olmak üzere iki şekilde ele alınmıştır. Her iki durumun en iyi çözümü için üç özellik tanımlanmış ve en kısa işlem zamanı çözüm algoritmasının en iyi çözüm olduğu gösterilmiştir. Ayrıca en iyi çözümü bulmak için, bir dinamik programlama algoritması ve dal sınır algoritması (DSA) sunulmuştur.

Zhang ve diğ. [19], C_{max} minimize amaç fonksiyonlu EATÇ probleminin çözümü için etkin bir GA önermişlerdir. Önerdikleri GA, global ve yerel seçim yaparak, yüksek kalitede başlangıç popülasyonu oluşturmaktadır. Geliştirdikleri GA'yı

literatürde yer alan kıyaslama problem setleriyle test etmiş ve algoritmalarının EATÇ problemlerinin çözümü için etkin sonuçlar verdiğini göstermişlerdir.

Xing ve diğ. [20], EATÇ problemlerinin çözümü için birbirinden farklı ve bağımsız popülasyonlardan oluşan karınca kolonileri algoritması (KKA) ve GA tabanlı bir yaklaşım sunmuşlardır. Yazarlar bu çalışmalarında popülasyonların birbirleriyle etkileşim içinde olmasını sağlayacak bir mekanizma ile yeni popülasyonlar elde etmiş, elde edilen bu popülasyon hem GA hem de KKA'nda ilerletilerek optimal çözüm bulunmaya çalışmışlardır. Amaç fonksiyonu olarak $C_{\max}$ 'ın en küçüklenmesi olarak ele alınmıştır. Algoritmalarını literatürde yer alan kıyaslama problemleriyle test eden yazarlar, etkin sonuçlar elde etmişlerdir.

ATÇ Probleminin amacı, teknolojik kısıtlar altında işletmenin üretim çizelgelemesini optimize etmektir. Bu tür problemlerin çözümü için literatürdeki matematiksel ve algoritmik modeller ile sonuç elde etmek zorlaşmaktadır. EATÇ problemi, ATÇ problemine göre daha kompleks yapıya sahiptir. Alternatif sezgisel metotlar problemin kabul edilen kısıtlarının sadece bir kısmına çözümler üretebilir. Gutierrez ve Garcia-Magarino [21], EATÇ problemlerinde kullanılan sezgisel yöntemler üzerine araştırma yapmışlardır. Yazarlar öncelikle problemin optimal olmayan çözümlerini elde etmek için GA kullanmışlardır. Elde ettikleri optimal olmayan sonuçları iyileştirmek için yenileyici sezgiseller uygulamışlardır. Problem kısıtlarının farklı türleri için farklı sezgisel metotlar vardır. Yazarların yaklaşımı EATÇ problemine çözüm algoritmasının uyarlanabilirliğini geliştirmiştir. Yazarlar sayısal ölçümlerle kendi yaklaşımlarının mevcut diğer yaklaşımlara göre daha iyi sonuçlar verdiğini göstermişlerdir.

Amiri ve diğ. [22], $C_{\max}$ 'ı minimize amaçlı EATÇ probleminin çözümü için bir algoritma sunmuşlardır. Sundukları komşuluk yapısını esas alan algoritmaları, mevcut çizelge üzerindeki sıralı komşu işlerin yerlerini değiştirerek daha iyi sonuçları veren çizelgelemeyi bulmak şeklindedir. Çalışmalarında 2 tür komşuluk yapısını esas alan yöntem sunmuşlardır. Yazarlar kendi algoritmalarını literatürde yer alan 181 adet kıyaslama problem setiyle çözmüşler ve EATÇ problemlerinin en iyi bilinen çözümlerine tamamen yakın sonuçlar almışlardır.

De Giovanni ve Pezzella [23], dağıtık ve EATÇ probleminin çözümü için geliştirilmiş bir GA sunmuşlardır. Yazarlar geliştirdikleri GA için genlerin kodlanması, çaprazlama ve mutasyon işlemleri için yaklaşımlarının ayrıntılarını vermişlerdir. Algoritmanın çaprazlama ve mutasyon adımlarına YA algoritmasını adapte etmişlerdir. Amaç fonksiyonu olarak $C_{\max}$ 'ı ele almışlardır. Algoritmalarını literatürde yer alan kıyaslama problemleri ile değerlendirmiş ve kendi yöntemlerinin tatmin edici sonuçlar verdiğini göstermişlerdir.

YBS, canlılardaki bağışıklık sisteminin, hesaplama problemlerinde taklit edilmesi sonucu ortaya çıkmış sezgisel bir problem çözüm yaklaşımıdır. Aladağ [24], tekrar işlemeli (geri dönüşlü) EATÇ problemi için YBS algoritmasını kullanan bir çözüm yaklaşımı geliştirmiştir. Önerilen yaklaşımın başarılı sonuçlar verdiği literatürdeki test problemleri kullanılarak gösterilmiştir. Ayrıca önerilen yaklaşım ile büyük boyutlu bir gerçek yaşam problemi de başarıyla çözülmüştür.

Son yıllarda üretim teknolojilerindeki hızlı gelişmelerin sonucu olarak esnek üretim çizelgelemenin önemi, uygulamaya dayalı problemleri içerdiğinden dolayı artmıştır. Özgüven ve diğ. [25] sıralama ve yükleme alt problemleri ile alternatif süreç seçimli esnek süreç planlarına sahip alt problemlerini kapsayan ve NP-zor problem sınıfında yer alan iki tür EATÇ problemlerinin çözümü üzerine çalışmışlardır. Yazarlar, amaç fonksiyonu olarak $C_{\max}$ 'ı ele almışlardır. Çalışma iki adımda gerçekleştirilmiştir. Birinci adımda, EATÇ probleminin çözümü için karışık tam sayılı programlama modeli sunmuşlar ve literatürde bulunan diğer çözümler ile karşılaştırmışlardır. Çalışmalarının ikinci adımında, birinci adımda sundukları algoritmalarını alternatif süreç planlarına sahip EATÇ problemlerine yeniden uyarlayarak algoritmalarını test etmişlerdir. Yazarlar test ettikleri problem setlerinin çoğunluğunda karşılaştırdıkları algoritmalara göre daha iyi sonuçlar elde etmişlerdir.

Defersha ve Chen [26], sıra bağımlı hazırlık zamanlı EATÇ problemlerinin matematiksel modelini ve bu tür problemlerin çözümü için etkin sonuçlar veren paralel GA yaklaşımı sunmuşlardır. Amaç fonksiyonu olarak $C_{\max}$ 'ı almışlardır. Yazarların sundukları model aynı zamanda, EATÇ problemini makine gevşek zamanı ve gecikme şartları altında da incelemektedir. Yazarlar algoritmalarını orta ve büyük ölçekli problem setleriyle ölçmüşler ve etkin sonuçlar elde etmişlerdir.

Lei [27], bulanık işlem süreli EATÇ problemlerinin çözümü için bulanık C_{max} 'ı minimize etme amaçlı etkin bir GA sunmuştur. İş sıralamalarının ve makine atamalarının birbirinden bağımsız olarak gerçekleştirildiği ikili temsilli kodlama tanımlamıştır. Yazar kendi algoritmasının literatürdeki diğer algoritmalara göre daha iyi sonuçlar verdiğini göstermiştir.

Fahmy ve diğ. [30], EATÇ problemleri için yeni bir yaklaşım sunmuşlardır. Bu yaklaşımda, makine arızalanması, işlem sürelerindeki değişimler, acil işlerin devreye konulması, iptal edilen işlerin olması gibi gerçek yaşam varyasyonları göz önünde bulundurularak, serbest kilitlemeli reaktif bir yaklaşım (GDRS) algoritması geliştirmişlerdir. Yazarlar, algoritmalarını literatürde yer alan problem setleriyle karşılaştırmış ve etkin sonuçlar almışlardır.

Girish ve Jawahar [31], NP-zor problem sınıfı içerisinde yer alan, çoklu iş rotasına sahip ATÇ problemlerinin çözümü için KKA ve GA meta sezgisel yöntemlerini sunmuşlardır. C_{max} Performans ölçütünü alarak, işlerin makinelere optimum atanmasını sağlamak üzere iki yöntemi geliştirmişlerdir. Literatürde yer alan kıyaslama problem setleriyle çözüm yapmışlar ve kendi sundukları KKA'nın literatürde bulunan en iyi sonuçlara yakın veya eşit sonuçlar verdiğini göstermişlerdir.

Yazdani ve diğ. [32], EATÇ problemlerinde C_{max} 'ı minimize etme amaçlı çözüm için meta sezgisel yöntemler sınıfında yer alan TB algoritması geliştirmişlerdir. Algoritmalarına yerel tıkanmalardan kaçma özelliğine sahip çözüm uzayındaki aramaları geliştirmek için YA algoritmasını adapte etmişlerdir. Yazarlar algoritmalarını literatürde yer alan kıyaslama problemleri ile çözmüş ve sunulan yaklaşımın etkin sonuçlar verdiğini göstermişlerdir. Diğer bir çalışmalarında Yazdani ve diğ. [33], aynı problemin aynı amaç fonksiyonu için paralel değişken komşuluk arama (DKA) algoritması geliştirmişlerdir. Algoritmadaki paralellik yapı arama uzayında bağımsız aramaları çoğaltmak için kullanılmıştır. Yazarlar literatürdeki problem setleriyle algoritmalarının etkin sonuçlar verdiğini göstermişlerdir.

Süer [34], esnek atölyelere sahip siparişe üretim yapan firmalar için bir operasyonel (kısa vadeli) esneklik yönetimi yaklaşımı önermiştir. Bu yöntemde, sipariş inceleme ve sürme teknikleri ile tipik esnek imalat sistemi kararları birleştirilmiştir. Önerilen

yöntem, her üretim döneminde atölyeye farklı seviyelerde işlem ve rotalama esneklikleri tahsis edilmesiyle bir atölye ortamı hazırlamaktadır. Sipariş havuzunda bulunan ve gelmesi öngörülen parça tiplerinin çeşitliliği, hacmi ve kritikliği, esneklik tahsisi için ana girdileri oluşturmaktadır. Bir esneklik yönetimi politikası tanıtılmış ve uygun politikanın belirlenmesi matematiksel programlama ve benzetim modellemesinin ortak kullanımı yoluyla gerçekleştirilmiştir. Deneysel bir çalışma yapılarak, önerilen yöntemin kuramsal bir esnek atölye üzerindeki etkileri incelenmiştir. Sonuçlar, uygun bir politika izlenen dönemsel ve çevrimiçi esneklik yönetiminin, sipariş inceleme ve sürme teknikleri ile birleştirildiğinde talepteki belirsizlik ile başa çıkmak konusunda etkin bir araç olabileceğine işaret etmektedir.

Pezzella ve diğ. [29], EATÇ problemleri için bir GA yaklaşımı sunmuşlardır. Başlangıç popülasyonu ve yeni bireylerin üretilmesinde farklı stratejileri algoritmaya entegre etmişlerdir. Algoritma, bilinen en iyi algoritma olan tabu aramaları (TA) sonuçları ve diğer GA sonuçları ile karşılaştırılmış, GA sonuçlarına göre daha iyi sonuçlar alınmış, TA sonuçlarına yakın sonuçlar elde edilmiştir.

Xing ve diğ. [35], EATÇ problemlerinin çözümü için bilgi tabanlı KKA geliştirmişlerdir. Yazarların sundukları algoritma, bilgi modeli ile karınca kolonileri optimizasyon modelinin entegrasyonundan oluşmaktadır. Amaç fonksiyonu olarak C_{max} 'ı ele almışlardır. Yazarların sundukları algoritma literatürde yer alan kıyaslama problemlerine uygulanmıştır. Algoritma sonuçları, mevcut bazı yaklaşımlara göre daha iyi sonuçlar verdiğini göstermiştir.

Huang ve Liao [36], klasik ATÇ probleminin çözümü için TA ve KKA'nın birleştirilmesiyle oluşan melez bir algoritma sunmuşlardır. Amaç fonksiyonu olarak C_{max} 'ı ele almışlardır. Yazarlar sundukları algoritmayı literatürdeki kıyaslama problem setiyle analiz etmiş ve etkin sonuçlar almışlardır.

Chen ve diğ. [37], paralel makineli ve tekrar işlemeli durumlar için EATÇ problemleri üzerine çalışmışlardır. Yazarlar, iki tip EATÇ probleminin varlığından bahsetmişlerdir. Birinci tip problemlerde, işler alternatif işlem sıralamasına sahiptir ve her iş için alternatif özdeş veya özdeş olmayan makineler söz konusudur. Burada problem işlerin sıralanması ve makinelerdeki işlem sırasının belirlenmesidir. İkinci tip problemlerde, işler sadece belli bir işlem sıralamasında işlenebilmekte fakat her

işlem için özdeş veya özdeş olmayan makine alternatifleri vardır. Problem işleri kendi işlem sıralamalarını bozmadan makinelere atayabilmektir. Çizelgeleme, makine seçim modülü ve işlem çizelgeleme modülü olarak iki aşamada gerçekleşmektedir. Makine seçiminde çeşitli öncelik kuralları kullanılmış ve daha sonra işlerin sıralaması da öncelik kurallarına göre en iyi çizelgeyi sağlayacak şekilde yapılmıştır. Uygulama bir silah fabrikasında gerçekleştirilmiş ve geliştirilen en erken teslim zamanı, malzeme listesindeki en düşük seviyeli kod ve en uzun işlem süresinin kombinasyonuna dayanan yöntem oldukça başarılı olmuştur.

Lei ve Guo [38], bulanık C_{max} performans ölçütlü, bulanık EATÇ probleminin çözümü için iki popülasyonlu GA önermişlerdir. Yazarların algoritmaları çözümü göstermek için iki dizi ve optimal çizelgeyi bulmak için iki popülasyon kullanır. Her bir neslin popülasyonları içinde uygulanan çaprazlama ve mutasyon işlemleri, kromozomun sadece bir kısmına uygulanır ve bu popülasyonlar birleştirilmiş popülasyonlar içindeki en büyük performans ölçütü değerine sahip diğer bireylerin yarısı kullanılarak iyileştirmeler yapılır. Yazarlar algoritma performanslarını ölçmek için problem setleriyle çözüm yaparak, üç değişik algoritma ile karşılaştırmışlardır. Algoritmalarının etkin sonuçlar verdiğini göstermişlerdir.

Weng ve diğ. [39], tam zamanında üretim ile ilişkili olarak, teslim tarihli ve iş yükü kontrolü tabanlı, iki yeni kural geliştirmişlerdir. Yazarlar bu iki kurala geri besleme mekanizmasını da ilave etmişlerdir. Sundukları yöntemlerini esnek atölye ortamına uygulamış ve elde edilen sonuçlarla, kendi yöntemlerinin tam zamanında üretim yapan işletmelerin üretim planlama ve kontrol faaliyetleri için çok etkin sonuçlar verdiğini göstermişlerdir.

Zribi ve diğ. [40], EATÇ problemi için yeni bir hiyerarşik metot sunmuşlardır. Yaklaşımlarını yüksek esnekliğe sahip ATÇ problemine adapte etmişlerdir. Öncelikle yerel bir arama ve sezgisel bir yaklaşım olan DSA' nı sunmuşlar, daha sonra melez bir GA geliştirmişlerdir. Amaç fonksiyonunda C_{max} ' ı göz önünde bulundurmışlardır. Yazarlar, yaklaşımlarının iyi sonuçlar verdiğini göstermişlerdir.

EATÇ Problemi hem üretim yönetimi hem de kombinatoriyel optimizasyon alanlarında büyük öneme sahiptir. Hesaplama karmaşasından dolayı, geleneksel optimizasyon yöntemleri ile, orta büyüklükteki ve gerçek durum problemlerinde

optimal çözümü bulmak oldukça zordur. İki işten fazla gerçek durum probleminin çözümü için, bütünleşik ve hiyerarşik olmak üzere iki çeşit yaklaşım kullanılmaktadır. Fattahi ve diğ. [12], EATÇ probleminin çözümü için matematiksel model ve iki sezgisel yöntem sunmuştur. Matematiksel model, küçük boyutlu problemlerde, optimal çözümü bulmak için kullanılmıştır. EATÇ'nin gerçek yaşam problemlerini çözmek için, bütünleşik ve hiyerarşik olmak üzere iki tane sezgisel metot geliştirmişlerdir. Sunulan sezgisel yaklaşımlarda TB ve TA yöntemleri kullanılmıştır. İki algoritmaya bağlı olarak, 6 farklı melez arama yapısı sunmuşlardır. Matematiksel model için küçük boyutlu, diğer iki sezgisel yaklaşım için orta ve büyük boyuttaki problem setleriyle çözümler yapılmış ve yaklaşımlarının etkin sonuçlar verdiği gösterilmiştir.

Ho ve diğ. [41], EATÇ problemleri için evrimleşme ve öğrenme arasındaki etkileşimi sağlayan öğrenebilir genetik mimari algoritması (LEGA) önermişlerdir. Yazarların sunduğu algoritma seçim ve mutasyona dayalı klasik evrimsel algoritmaların tersine, önceki nesillerden öğrenme modülü yardımıyla bilgi toplamakta ve bu bilgiyi gelecek nesillerin çeşitliliğini ve kalitesini artırmada kullanılmaktadır. Algoritmanın performans ölçütü olarak C_{max} 'ın en küçüklenmesini ele almışlardır. Literatürde yer alan büyük boyutlu problem setleriyle yaptıkları çözümlerde, bu yaklaşımın klasik evrimsel yaklaşımlara göre daha başarılı olduğunu göstermişlerdir.

EATÇ gibi çoklu makinelerin etkin kullanımını sağlamak üzere yapılan çizelgeleme çalışmaları gerçek yaşam problemleri sınıfındandır. Optimal çözüm veren metotlar, NP-zor problem sınıfına giren EATÇ problemlerinin çözümünde uygulanamadığından dolayı, iyi sonuçları uygun zamanda elde edebilmek için sezgisel ve meta sezgisel yöntemler geliştirilmiştir. Liouane ve diğ. [42], esnek üretim sistemlerinde kullanılan EATÇ problemlerinin çözümü için iki yaklaşım sunmuşlardır. Bu yaklaşımlar YA metodunu içeren KKA ve TA'dır. Amaç fonksiyonu olarak C_{max} ele alınmıştır. Yazarlar önerdikleri algoritmaları, Brandimarte [2] ve Kacem ve diğ. [10] algoritmalarıyla karşılaştırmışlar ve etkin sonuçlar elde etmişlerdir.

Saidi-Mehrabad ve Fattahi [43], $C_{\max}$ 'ı minimize etme amaçlı EATÇ problemlerini çözmek için TA algoritması geliştirmişlerdir. Yazarların algoritması operasyonların en iyi sıralamada yapılmasını sağlayan ve bu operasyonların en iyi makinelere atanmasını sağlayan iki alt gruptan oluşmaktadır. Algoritma performansını ölçmek için kendi rassal ürettikleri problem setlerini kullanmışlardır. Kendi problem setlerini DSA ile çözerek algoritmalarının etkinliğini değerlendirmişlerdir. Yazarlar, algoritmalarının küçük ve orta boyuttaki problem setleri için kısa sürede iyi sonuçlar verdiğini göstermişlerdir.

Gomes ve diğ. [44], EATÇ problemi için yeni bir tam sayılı lineer programlama modeli sunmuşlardır. Model, paralel özdeş makine gruplarını ve makine hazırlık zamanlarını ihmal edilebilir olarak kabul etmektedir. Model, gerçek yaşam problemlerinde çözülmüş ve gerçek yaşam problemlerinde bu model ile daha iyi sonuçlar elde edilebileceği gösterilmiştir.

EATÇ problemi, her operasyonun bir grup makine üzerinde işlem gördüğü klasik ATÇ problemlerinin bir türüdür. Bu model gerçek üretim sistemlerinde geniş kullanıma sahip bir problem türüdür. Jansen ve diğ. [45], EATÇ probleminin çözümü için bir doğrusal yaklaşım sunmuşlardır. Bu yaklaşımları aynı zamanda m adet makine ve n adet işin sabit olduğu problemin öncelik versiyonu üzerinedir. Yazarlar yaklaşımlarını bazı problem setleriyle ölçmüş ve etkin sonuçlar almışlardır.

Zribi ve diğ. [46], çalışmalarında EATÇ problemleri için ilk aşamada atamaların sezgisel YA ile yapıldığı, algoritmanın ilerlemesinin ise GA yardımıyla sağlandığı optimizasyon tekniği geliştirmişler, sonuçları bazı problemlerin alt sınırları ile karşılaştırmışlar ve tatmin edici sonuçlara ulaşmışlardır.

Baykasoğlu [47], EATÇ problemlerinin çözümü için TB tabanlı bir meta sezgisel çözüm yaklaşımı sunmuştur. Yazarın sunduğu model, EATÇ probleminin süreç modelini daha basit hale getiriyor ve ATÇ probleminin var olan çözüm algoritmalarının kullanımını sağlıyor. Amaç fonksiyonu olarak $C_{\max}$ alınmıştır. Elde edilen sonuçlar bu tür kompleks problemlerin önerilen algoritma ile makul süre içinde etkin sonuçlar verdiğini göstermiştir.

Mastrolilli ve Gambardella [48] EATÇ problemlerinin çözümü için YA ve iki yeni komşuluk yapısı geliştirmişlerdir. $C_{\max}$ 'ın minimize edilmesi amaçlı ele aldıkları problemin çözümü için TA yöntemini sunmuşlardır. Yazarlar literatürde yer alan kıyaslama problemleri ile TA algoritmalarının daha hızlı ve etkin sonuçlar verdiğini göstermişlerdir.

Brandimarte [2], EATÇ problemi için hiyerarşik bir TA algoritması tanımlamıştır. Yazar, işlerin makinelere optimal atanması ve atanan işlerin makinelerde optimal sıralanmasını birbirinden bağımsız olarak düşünmüştür. Amaç fonksiyonu olarak, $C_{\max}$ ve toplam ağırlıklı gecikme düşünülmüştür. Problem setleri sonucunda elde edilen sonuçlar değerlendirilmiştir.

Zhang ve Wu [49], toplam ağırlıklı gecikme zamanının en küçüklenmesi amaçlı ATÇ problemini çalışmışlardır. Öncelikle problemin matematiksel modelini ve dual modelini sunmuşlardır. Bu problemin çözümü için, bir blok tabanlı komşuluk yapısı tanımlanmış ve bunun özellikleri sunulmuştur. Son olarak bu komşuluğun özelliklerini direkt olarak kullanan TB algoritması sunmuşlar ve yüksek kalitedeki çözümlere yakınsayan sonuçlar elde etmişlerdir.

Baykasoğlu ve Özbakır [50] farklı esneklik seviyelerine sahip ATÇ performansı üzerine dağıtım kurallarının etkilerini analiz etmişlerdir. Operasyonlar için dört farklı esneklik seviyesi tanımlanmıştır. Bu makineler sıfır, düşük, orta ve yüksek esnekliğe sahip olarak seçilmiştir. Çizelgeleme performans ölçütü olarak ortalama gecikme zamanı kabul edilmiş olup beş dağıtım kuralı değerlendirilmiştir. En kısa işlem zamanı, en erken teslim tarihi, en çok kalan iş zamanı, toplam kalan işlem zamanı ve minimum aylak zaman olmak üzere 5 dağıtım kuralı alınmıştır. Sunulan 5 kural için performans değişimleri, farklı esnekliğe sahip makineler üzerinde belirlenmiş ve istatistiksel analiz yapılmıştır. Dağıtım kurallarının etkilerinin detaylı analizi için üç farklı boyutta problem seti ile çalışılmıştır. Yazarlar tarafından sonuç olarak, atölye çizelgeleme esnekliği artarken atölye çizelgeleme performansının azaldığı gösterilmiştir.

Essafi ve diğ. [51], Toplam ağırlıklı gecikmeyi minimize amaçlı, teslim tarihli ve serbest zamanlı ATÇ problemi için YA'lı bir GA sunmuşlardır. Çalışmalarında GA parametreleri üzerine yoğunlaşmışlardır. Yazarlar kendi algoritmalarını literatürde

yer alan kıyaslama problem setleriyle çözmüşler ve algoritmalarının etkin sonuçlar verdiğini göstermişlerdir.

Kunadilok [52], toplam ağırlıklı gecikme zamanını minimize etme amaçlı, kompleks ATÇ problemi üzerine çalışmıştır. Kompleks ATÇ, bir veya daha fazla sayıda özdeş paralel makineden oluşan farklı iş merkezlerinde, işin akışı olarak tanımlanmaktadır. Süreç kısıtlarına sıfırdan farklı hazırlık zamanı ve bir dizi bağımlı hazırlık zamanları dâhildir. Yazar, kompleks atölye çizelgelemeyi; sınırsız ve sınırlı tampon kapasiteler olmak üzere iki sınıfta toplamıştır. Sınırsız tampon kapasiteli problemin çözümü için, karışık tamsayılı programlama, rastsal kodlama tabanlı bir GA ve TA olmak üzere üç yöntem sunmuştur. Sınırlı tampon kapasiteli kompleks ATÇ’de özellikle küçük boyutlu problemlerin çözümü için ise yine karışık tamsayılı programlama modeli geliştirmiştir. Son olarak, tampon kullanılabilirlik ve serbest iş planı kısıtları altında, atandıkları iş merkezleri boyunca iş akışını çizelgelemek için rastsal popülasyonlu GA sunmuştur. Yazar kendi algoritmalarını sezgisel yaklaşımlar ile karşılaştırmış ve sınırlı ve sınırsız tampon kapasiteli kompleks ATÇ problemleri için algoritmalarının etkin sonuçlar verdiğini göstermiştir.

Zribi ve diğ. [53] EATÇ problemlerinin çözümü için TA ve GA yöntemlerini melezleyerek yeni bir yaklaşım sunmuşlardır. Amaç fonksiyonu olarak tedarik zinciri ve üretim süreçlerinde önemli bir kriter olan toplam gecikme amacını en iyilemeye çalışmışlardır. Bu yaklaşımlarının ilk bölümü atama alt probleminin çözümünün yapıldığı TA, ikinci bölümü ise sıralama alt probleminin çözümünün gerçekleştirildiği GA tabanlı yaklaşımdan oluşmaktadır. Yazarlar, atama ve sıralama alt problemlerinin bütünleşik olarak çözülmesi ile daha iyi sonuçlar aldıklarını söylemişlerdir.

Büyükköprü [54], sıraya dayalı kurulum zamanlarının sistem performansları üzerindeki etkilerinin gözlenmesi üzerine iki sezgisel grubu karşılaştırmıştır. Bir grupta kurulum, çizelgeleme sürecinde göz önünde tutulurken diğer grupta göz ardı edilmiştir. Üzerinde durulan üretim yapısında kurulum süresi hem henüz işlenmiş olan hem de arkasından işlenecek olan işlere bağlı olan önemli bir unsurdur. Ayrıca her kalıp tipinden sadece tek adet olması, aynı işleme gereksinim duyan iki parçanın paralel, ayrı makinelerde işlenmesine engel oluşturmaktadır. Bu kısıtlar altında ve sık

ve uzun kurulum zamanlarının tehdidi altında kurulum zamanının düşürülmesine ve üretim zamanının etkin kullanımına katkısı olacak etkin bir çözüm sunmuştur. Akış zamanı, ortalama gecikme, gecikme oranı ve süreç içerisindeki parça sayısı gibi performans değişkenleri üzerinde kurulum zamanının ne gibi etkilerinin olduğunu gözlemlemek için deneysel dizayn uygulamıştır. Kurulum zamanının yukarıda sayılan değişkenler üzerindeki etkilerinin varlığına, tanımlanan çizelgeleme algoritmalarının kurulum zamanlarındaki değişimlere karşı gösterdiği tepkiye ve kurulum zamanlarının parça işleme zamanlarından ayrı olarak amaç fonksiyonunda ek bir kısıt olarak değerlendirilmesinin önemine vurgu yapmıştır.

Jang ve diğ. [55], çok düzeyli iş yapılarına sahip EATÇ problemi ile ilgilenmişlerdir. Yazarlar, karmaşık çizelgeleme problemi için, montaj hattında işlemlerin seviye ve sıralamasını makine üzerinde gösteren yeni bir gen tabanlı GA çözümü sunmuşlardır. Toplam gecikme zamanını minimize eden amaç fonksiyonlu GA çözümleri ile literatürde yer alan 40 adet problem setinde kendi çözüm performanslarının yüksek çıktığını göstermişlerdir.

Alvarez-Valdes ve diğ. [56], bazı özel karakteristiklerle esnek üretim sistemine sahip bir cam fabrikasında çizelgeleme sisteminin uygulamasına yönelik sezgisel bir yaklaşım sunmuşlardır. Sıcak sıvı camlar beklemeksizin bazı operasyonlara tabi tutulması gerekmektedir. Diğer yandan, kalifiye işçiler, manuel yapılan işleri bazı özel tezgâhlarda işlemektedir. Yazarların sunduğu algoritma, kısa süre içerisinde, belirli olmayan kısıtlamalar altında, teslim tarihi minimize amaçlı problemi çözme amaçlıdır.

EATÇ problemleri ek karmaşıklıklar getiren klasik ATÇ problemlerinin bir uzantısıdır. Wu ve Weng [57] esnek atölye ortamında erken ve geç teslim tarihi amaçlı sezgisel algoritma sunmuşlardır. Erken ve geç teslim amaçlı problem tam zamanında üretim felsefesi içerisinde ayrı bir öneme sahiptir. Yazarlar algoritmalarında yeni bir iş rota ve sıralaması sunmuşlardır. Yazarlar sundukları algoritmayı beş iş merkezi bulunan esnek atölye ortamında denemiş ve algoritmalarının mevcut olan diğer algoritmalar içinde iyi sonuçlar verdiğini göstermişlerdir.

Dosdoğru [58] çalışmasında, bir dinamik esnek atölye tipi üretim ortamında olurlu en iyi parça rotalarının üretilmesini sağlayan bir sistem geliştirmiştir. Bu amaçla, bir dinamik esnek atölye tipi üretim ortamında, sırasıyla hem her bir parça için olurlu en iyi proses planı hem de her bir operasyon için en iyi olurlu makinenin belirlenmesi gerekmektedir. Bu bağlamda, her bir parça için en iyi parça proses planı ve sonra belirlenen parça proses planına göre her bir parçanın her bir operasyonu için uygun makinenin seçilmesini sağlayan GA geliştirmiştir. Yazarın geliştirdiği algoritma çözüm metodolojisinin optimizasyon aşamasını oluşturmaktadır. Elde edilen makine-operasyon ikilileri, performanslarının tahmini amacıyla, kesikli olay sistem simülasyonu modeline beslenmiştir. Çalışmasının bu iki aşaması birbirini ardışık bir şekilde izlemektedir. Önerilen metodolojinin amacı tüm parçalar için ortalama akış süresini minimize eden çözümü bulmaktır. Yazarın sonuçları, amaç fonksiyonunun göz önüne alınan esneklik seviyesinin artan değerleri için iyileştirdiğini göstermiştir.

Buraya kadar incelenen tek amaçlı EATÇ problemleri amaç fonksiyonu bazında tarih sırasına göre en yeniden en eskiye doğru sıralanmıştır. İncelenen 46 çalışma içerisinde 36 tanesi C_{max} , 8 tanesi gecikme zamanı ölçütlü, 2 tanesi ise teslim zamanı ölçütlü olarak ele alınmıştır. Yapılan çalışmaların 33 tanesi statik durum için 13 tanesi dinamik durumlar için problemi ele almışlardır. Sunulan algoritmalar genelde literatürdeki problem setleriyle değerlendirilirken incelenen 46 çalışmadan sadece 7 tanesi bir işletmede gerçek yaşam problemlerine uygulanmıştır. Araştırmacılar tek amaç fonksiyonlu çalışmalarda, toplam tamamlanma zamanını en küçükmek üzere genelde Brandimarte (BRdata) [2] 10 adet problem seti, Hurink ve diğ. (HUdata) [59] 128 adet problem seti, Dauzere Peres ve Paulli (DPdata) [60] 18 adet problem setini kullanmışlardır. Tek amaçlı EATÇ problemlerinin çözümünde en çok GA meta sezgiselini tercih etmişlerdir. Çok amaçlı EATÇ problemlerinin çözümünde araştırmacılar genelde 2 meta sezgisel yöntemi melezleyerek çözümler elde ederken tek amaçlı EATÇ problemlerinin çözümleri için kullandıkları meta sezgisel yönteme özellikle YA algoritması başta olmak üzere klasik sıralama kurallarını adapte ettikleri gözlenmiştir. Araştırmacıların problemleri genelde statik durumlar için çözmeleri, gerçek yaşam problemleri sınıfından olan dinamik durumlar için az sayıda yapılan çalışmalar bir eksiklik olarak gözlenmiştir.

2. OPTİMİZASYON PROBLEMLERİ VE ÇÖZÜM YÖNTEMLERİ

Optimizasyon temel olarak, eldeki kısıtlı kaynakların optimum kullanımını gerçekleştirmektedir. Optimizasyon matematiksel olarak, bir amaç fonksiyonunu verilen tanım aralığında optimum yapan değerini bulmaktır. Yöneylem araştırması ve işletme yönünden, bir sistemi en düşük maliyetle en yüksek verimi elde etmek üzere düzenleme yapılması olarak tanımlanabilir [61]. Optimizasyon temelde modelleme ve çözümleme olarak iki alt bileşenden oluşur. Modelleme, gerçek hayatta karşılaşılan bir problemin matematiksel olarak ifade edilmesidir. Çözümleme ise modelin çeşitli yöntemler kullanılarak en iyi sonucunun alınmasıdır.

Model, bir sistemin değişen şartlar altında davranışlarını incelemek, kontrol etmek ve geleceği hakkında varsayımlarda bulunmak üzere elemanları arasındaki bağlantıları matematiksel olarak ifade etmektir. Eğer karar değişkenleri üzerinde herhangi bir kısıtlama yoksa kısıtsız model, en azından bir kısıtlama varsa kısıtlı modeller haline gelir. Genellikle gerçek hayat problemleri kısıtlı problemlerden oluşur. Ele alınan optimizasyon problemi, sadece bir dönem için çözülecekse statik model, birden fazla dönem göz önünde bulundurularak çözülecekse dinamik model haline gelir. Problemden tek amaç en iyilenmeye çalışılıyorsa tek amaçlı, birden fazla amaç aynı anda en iyilenmeye çalışılıyorsa çok amaçlı problem halini alır. Eğer optimizasyon problemindeki karar değişkenleri pozitif reel değerler alıyorsa, sürekli optimizasyon problemi söz konusu olur. Tüm karar değişkenleri tamsayı değerler alması gerekiyorsa kesikli optimizasyon problemi ortaya çıkar. Problemden bazı karar değişkenlerinin reel bazılarının tamsayı alması gerekiyorsa karışık kesikli optimizasyon problemi söz konusu olur. Karar değişkenlerinin kombinatoriyal seçenekleri söz konusu ise kombinatoriyal optimizasyon problemleri ortaya çıkar.

Dinamik modeller için kullanılan yaklaşım dinamik programlamadır. Eğer optimize edilecek birden fazla amaç varsa genellikle kullanılan yaklaşım hedef programlamadır. Modeldeki tüm fonksiyonların doğrusal olması durumunda sürekli optimizasyon problemleri doğrusal programlama yöntemi ile çözülür. Sürekli optimizasyon modelinde en azından bir fonksiyonun doğrusal olmaması

durumundaysa doğrusal olmayan programlama yöntemi kullanılır. Eğer kesikli optimizasyon problemlerinde karar değişkenleri herhangi bir tamsayı değer alıyorsa tamsayılı programlama yöntemi kullanılır. Kombinatoryal optimizasyon problemlerinin belirli bir boyuta kadar olan tamsayılı programlama yöntemi ile çözülürken, orta ve büyük boyutlu problemlerin sezgisel ve metasezgisel yöntemlerle çözülmesi gerekmektedir [61].

Sezgisel kelimesi eski Yunancadan “heuriskein” kelimesinin karşılığı olarak, problemleri çözmek için yeni yöntemler geliştirme anlamının karşılığıdır. Optimizasyon problemlerinde sezgisel, optimale yakın sonuçlar bulan, stratejilere dayalı, hızlı ve pratik bir çözüm olarak kabul edilmektedir. Problem optimum çözümü bulunamayacak kadar karmaşıksa, sezgisel yöntemler sezgiye veya bazı deneysel kayıtlara dayanan karar kuralları ile belirli sayıda adımdan sonra en iyi olmasa da tatminkar bir sonuç verirler.

Büyük boyutlu kombinatoryal optimizasyon problemlerine kesin yöntemlerle çözüm bulmak mümkün olmadığından bu tür problemlerin çözümünde sezgisel algoritmalar tercih edilmektedir. Sezgisel algoritmalar en iyi sonucu garanti etmemekle beraber, kısa sürede optimale yakın sonuçlar verirler.

Klasik sezgisel yöntemler genellikle probleme özgü bir yapıya sahiptirler. Bir problem türü için geliştirilen sezgisel bir yöntemin başka bir probleme türüne uygulanması zordur. Ancak son yıllarda problem özelliklerinden bağımsız, genel olarak tanımlanmış ve meta sezgisel olarak adlandırılan yöntemlere ilgi artmıştır.

2.1. Meta Sezgisel Çözüm Yöntemleri

Meta sezgisel yöntemler zor ve karmaşık optimizasyon problemlerde kaliteli çözümlere ulaşabilmek için sezgisel yaklaşımları kullanan tasarımlar olarak da adlandırılabilir [103]. Çözüm uzayının büyüklüğü ya da optimizasyon modelindeki değişken ve kısıt sayılarının fazlalığı nedeniyle kesin sonuç alınan yöntemlerin kullanılmadığı durumlarda, uygulamaya elverişli olmayan problemlerin çözümünde, meta sezgisellerin kullanılması giderek yaygınlaşmaktadır. Kesin sonuçların alındığı yöntemlerden farklı olarak meta sezgiseller, optimum yerine optimuma yakın çözümler elde edilmesini sağlar [62].

Meta sezgisel yöntemlerin klasik sezgisel yöntemlere göre en büyük avantajı, meta sezgisel yöntemlerin temel adımlarının genel olarak tanımlanmış olması, probleme özgü tasarlanmamış olmaları ve her türlü optimizasyon problemlerine uygulanabilir olmalarıdır. Ancak meta sezgisel yöntemlerin sezgisel yöntemler gibi en iyi sonucu garanti etmemeleri meta sezgisel yöntemlerin en önemli dezavantajıdır.

Meta sezgisel yöntemlerin büyük bir grubunu popülasyon tabanlı algoritmalar oluşturmaktadır. Popülasyon tabanlı yöntemlerde probleme ait aday çözümleri temsil eden popülasyonlar oluşturulur ve bu popülasyonlar her adımda yonteme ait operatörler yardımıyla geliştirilerek daha iyi çözümler elde edilir. Popülasyon tabanlı algoritmaları kendi içinde gelişime dayalı ve sürü zekâ temelli algoritmalar olarak iki alt gruba ayırmak mümkündür. GA, genetik programlama, EA gelişime dayalı algoritmalar, PSO, KKA, TA, sürü zekâ temelli algoritmalarla örnek olarak verilebilirler.

PSO popülasyon tabanlı algoritmalar içinde sürü zeka temelli alt grubuna dahil bir optimizasyon tekniğidir. Optimizasyon problemlerine kolay uygulanabilen PSO, yerel ve global arama arasında kurduğu denge ile istenen çözümlere hızlı bir şekilde yakınsama özelliğine sahip olmasından dolayı tercih edilen yöntem olmuştur. PSO önceleri sürekli optimizasyon problemlerinin çözümü için kullanılırken sonradan yapılan çalışmalar PSO'nun ayrık problemlerde de başarılı sonuçlar verdiğini göstermiştir[63]. PSO'da parçacık hareketi hem kendi en iyi hemde sürü içindeki en iyi değere göre ayarlandığından, iyi bir hafıza yeteneğine sahiptir. Çok amaçlı EATÇ problemlerinin çözüm uzayının etkili bir şekilde taranabilmesi PSO algoritmasıyla daha verimli olacağı düşünülmektedir. PSO az sayıda parametreye ihtiyaç duyması algoritmanın diğer bir tercih edilme nedeni olmuştur. Aşağıda meta sezgisel optimizasyon yöntemlerinden en çok kullanılanlar ile ilgili genel bilgiler daha sonra EATÇ problemlerinin çözümünde kullanılan PSO algoritmasıyla ilgili detay bilgiler verilecektir.

2.1.1. Genetik algoritma

GA, doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemi olarak ifade edilmektedir. GA'ların, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. GA'nın

temel prensibi, her adımda bir önceki nesilden yeni bireyler oluşturarak amaç fonksiyonunun uygunluk derecesini artırmak ve sonuç olarak belli kısıtları sağlayacak şekilde amaç fonksiyonunu sağlayan en uygun değerini elde etmektir.

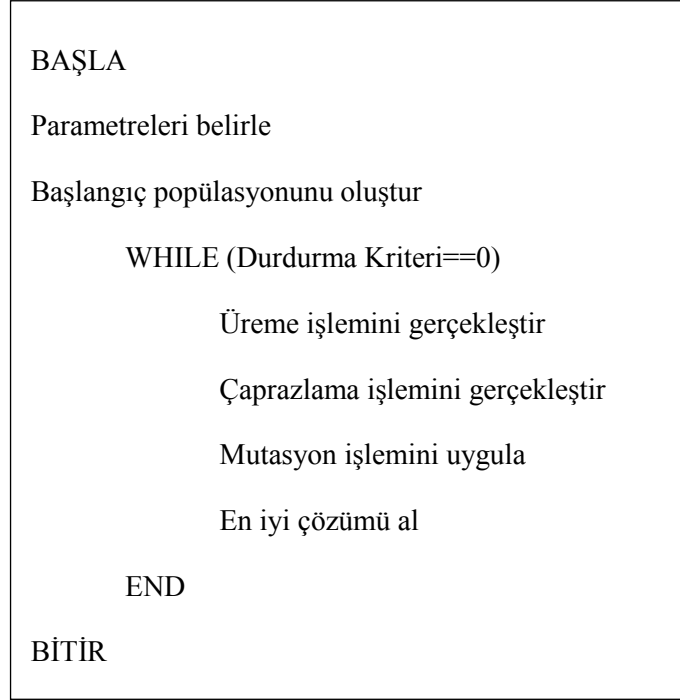
Geleneksel optimizasyon yöntemlerine göre farklılıkları olan GA, parametre kümesini değil kodlanmış biçimlerini kullanırlar. Olasılık kurallarına göre çalışan GA, yalnızca amaç fonksiyonuna gereksinim duyar. Çözüm uzayının tamamını değil belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa bir sürede çözüme ulaşırlar [64]. Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve böylelikle yerel en iyi çözümlere takılmamalarıdır.

GA'da kullanılan operatörler, mevcut popülasyon üzerine uygulanan işlemler olarak tanımlanabilir. Bu işlemlerin amacı daha iyi özelliklere sahip yeni nesiller üretmek ve aranan en iyi çözüm alanını genişletmektir. Farklı uygulamalarda farklı operatörler kullanılmakla birlikte genelde üreme, çaprazlama ve mutasyon olmak üzere üç standart operatör kullanılmaktadır.

Üreme, bireyleri seçme işleminden, seçilmiş bireyleri bir eşleme havuzuna kopyalama işleminden ve havuzda bireyleri çiftler halinde gruplara ayırma işleminden oluşur [65]. Üreme işlemi belli bir seçme kriterine göre bireylerin seçilip yeni kuşağın oluşturulması işlemidir. Seçme kriterleri uyumluluğu esas alarak birbirleriyle uyumlu olan bireyleri seçer. Daha sonra çaprazlama ve mutasyon uygulanacak olan bireylerden daha uyumlu yeni bireylerin ortaya çıkması olasıdır. Bireylerin tamamı uyumluluğa göre seçilebilir veya bir kısmı rasgele seçilerek yeni kuşağa aktarılabilir.

Çaprazlama operatörü GA'daki en önemli operatördür. Rastgele seçilen iki kromozomun yapıları kullanılarak yeni bir nesil oluşturulması esasına dayanır. Çaprazlama işlemi genel olarak ikili dizilerin parçalarının değiştirilmesi şeklinde gerçekleştirilir. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldığı için; tek noktalı çaprazlama, iki noktalı çaprazlama ve üniform çaprazlama gibi değişik çaprazlama yöntemleri kullanılmaktadır. Burada amaç, bir önceki neslin kromozom genlerinin yerini değiştirerek yeni nesil için kromozomlar üretmek ve böylece var olan uygunluk değeri daha yüksek olan kromozomlar elde etmektir.

GA'da çaprazlama işleminden sonra mutasyon işlemi gerçekleştirilir. Mutasyon oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek ve sonuca daha hızlı ulaşmak amacıyla yapılır. Mutasyon, bireyin kromozomunu oluşturan dizideki tek bir elemanın değerinin rastgele olarak değişmesidir. Mutasyon, çözümün alt optimal noktalara takılmasını önleyen ve çok düşük olasılık değeri ile uygulanan operatördür. Şekil 2.1 GA işlem adımlarını göstermektedir.



Şekil 2.1. Genetik algoritma adımları

EATÇ problemlerinin GA ile çözümünde Tamaki ve diğ. [66], Kacem ve diğ. [10], Tanev ve diğ. [67], Chan ve diğ. [68], Gao ve diğ. [67,68], Lin ve Jia-Zhen [71], Unachak [72], Wang ve diğ. [73], Prakash ve diğ. [74], Rabiee ve diğ. [75] çalışmalar yapmışlardır. Genel olarak yazarlar, GA ile çizelgeleme problemlerinde çok kısa süre içerisinde optimale yakın çözümler bulmuşlardır.

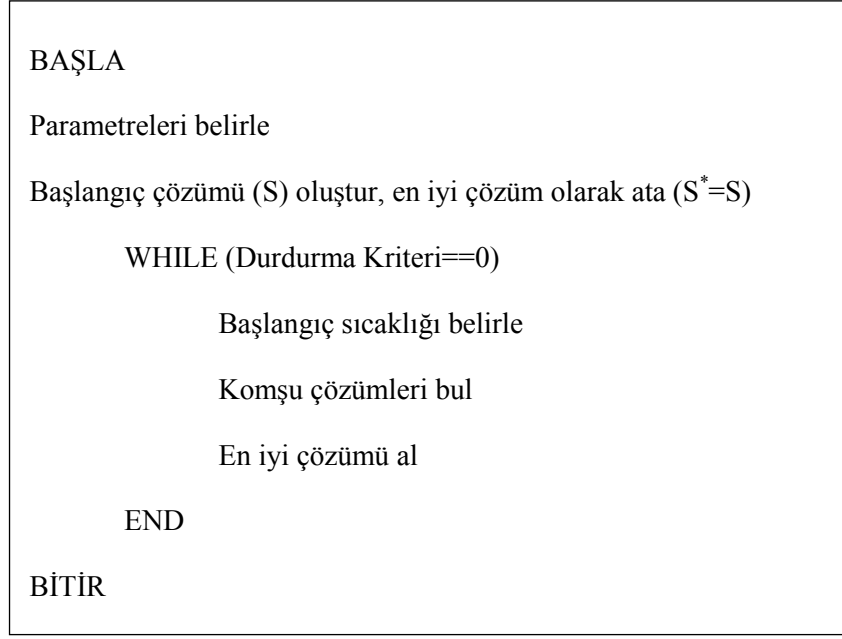
2.1.2. Tavlama benzetimi

TB yöntemi metallerin ısıtılıp soğutulmasıyla sertliğinin alınması işleminden esinlenen olasılıksal bir algoritmadır. Tavlama, metal bir malzemenin atomlarının serbestçe hareket edebileceği bir noktaya kadar sıcaklığının artırılması ve daha sonra atomların yeniden düzenlenmesi için sıcaklığın kademeli olarak düşürülmesidir. Burada en önemli işlem soğutma olayıdır. Eğer metal hızlı bir şekilde soğutulursa

metalin kristal yapısında düzensizlikler ve yapısal bozulmalar olur. Soğutma uygun şekilde yapılırsa kristal yapı çok düzenli olur. Metallerin ısıtma işlemi ile optimizasyon problemlerine çözüm arama arasındaki ilişki ilk defa Kirkpatrick ve diğ. [76] tarafından ortaya konulmuştur. Optimizasyon problemi ile ısıtma işlemi arasındaki ilişki, katının durumları optimizasyon probleminin muhtemel çözümlerine ve durumların enerjileri de çözümlere ait amaç fonksiyon değerlerine karşılık gelmektedir.

TB yerel komşuluk arama yöntemine benzer bir tekniktir. TB'nin ilk adımında başlangıç popülasyonu rastsal olarak oluşturulur ve başlangıç sıcaklığı ile soğuma süresi belirlenir. Bu yöntemde belirli bir başlangıç noktası seçilerek çözüm uzayında arama yapılır. TB çözüm uzayında arama yaparken elde edilen yeni çözüm değeri kötü çıksa bile yerel tıkanmalara takılmamak için uygun olmayan çözümlerinde kabul edildiği bir yöntemdir. TB'de problem performansını etkileyen diğer bir faktör komşuluk fonksiyonudur. Komşuluk genişliğini yüksek tutmak, daha iyi çözümler elde edilmesini sağlarken çözüm süresini de arttırabilir. Geçiş mekanizmasında kullanılan en basit yöntem rastgele seçim yapmaktır. Diğer bir yöntem; ilk iyi çözümlerin seçilip, bu çözümlerin üzerinde amaç fonksiyonunu en fazla etkileyen değişikliklerin yapılması da olabilir. Algoritmanın performansı, soğutma şartları ve komşuluk yapısına bağlıdır. Başlangıç çözümünün uygun bir çözüm olması ve amaç fonksiyonunun çözümün performansını etkin bir şekilde temsil etmesi de önem taşımaktadır. Şekil 2.2' de TB algoritmasına ait işlem prosedürü verilmiştir.

TB algoritmasının çok amaçlı optimizasyon problemlerinde kullanımı ilk olarak Serafini [77] tarafından yapılmıştır. Çok amaçlı TB olarak adlandırdığı algoritmada iki amaçlı optimizasyon problemini çözmek için hedef vektör yaklaşımını sunmuştur. Ray ve diğ. [78] çok amaçlı tasarım probleminin TB yöntemi ile çözümünde amaç fonksiyonunu ağırlıklandırarak çözmüşlerdir. Ruiz-Torres ve diğ. [79] iki amaçlı paralel makine çizelgeleme problemlerinin çözümü için pareto tabanlı TB algoritması sunmuşlardır. Çok amaçlı EATÇ problemlerinin çözümünde ise Xia ve Wu [28], TB ve PSO algoritmalarından melez bir algoritma sunmuşlardır. Ayrıca Fattahi [81] TB, Gholami ve Zandieh [82] TB ve GA'dan melez bir algoritma, Frutos ve diğ. [83] yine GA ve TB yöntemlerinden melez bir algoritma sunmuşlardır.



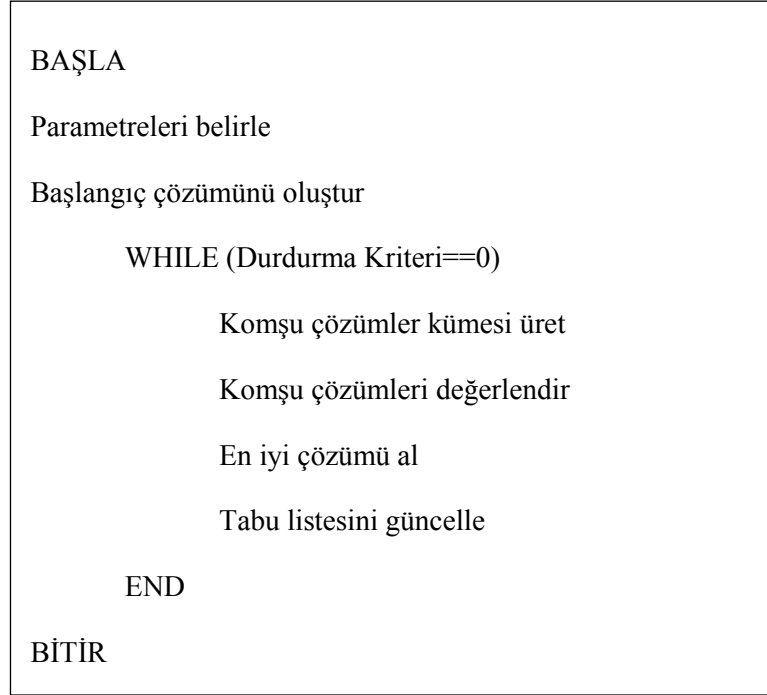
Şekil 2.2. Tavlama benzetimi algoritma adımları

2.1.3. Tabu arama

TA metodu, yerel tıkanmalara yakalanmamak için esnek hafıza yapılarını kullanan ve çözüm uzayını etkin bir şekilde tarayan, çizelgeleme problemlerinde çok kullanılan güçlü bir yöntemdir. İlk defa Glover [80] tarafından ortaya atılmıştır. TA Algoritması, bir başlangıç çözümü ile başlar ve çözümünden elde ettiği komşuları arasından performans kriterine göre en uygun komşuya doğru hareket eder. Burada dikkat edilmesi gereken en önemli nokta, yerel optimum noktalarda döngüye girerek global optimum noktayı yakalayamamasıdır. Buna engel olmak için araştırmacılar farklı stratejiler uygulamışlardır. TA’da bulunan çözümler tabu listesinde tutularak, aynı çözümlere geri dönmesi engellenmiş olur. Ancak bazı durumlarda daha önceden elde edilen çözümlere tekrar dönme ihtiyacı olabilir. Bu tür durumlar için uygun kriterler belirlenerek yasaklı listede olan çözümlere tekrar dönmesi sağlanabilir. Şekil 2.3 standart TA algoritması işlem adımlarını göstermektedir.

Çok amaçlı çizelgeleme problemlerinde TA yönteminin kullanımı Gandibleux ve diğ. [84] tarafından amaçlara ağırlıklar verilerek gerçekleştirilmiştir. Yazarlar, amaç fonksiyonundaki ağırlık değerlerini, amaç fonksiyonundaki iyileşme oranına göre değiştirmişlerdir. İki adet tabu listesi kullanmışlardır. Birincisi ulaşılmış çözümlere geri dönmeyi engelleyen normal tabu listesidir. Diğer ağırlık vektörünü kapsayan

listedir. Hansen [85] çok amaçlı sırt çantası probleminin çözümü için pareto tabanlı TA algoritması sunmuştur. Yazar, tabu listesinde baskın çözümlerin farklı alanlarını tarayan çözümleri elde tutmuştur. Her çözüm için bir ağırlık vektörü belirlemiştir.



Şekil 2.3. Tabu aramaları algoritma adımları

Hurink ve diğ. [59] ATÇ ortamında çok amaçlı makineler için TA algoritması geliştirmiştir. Dauzere Peres ve Paulli [60] çok işlemcili ATÇ problemlerinin çözümü için TA algoritması sunmuşlardır. Tek amaçlı EATÇ problemlerinin çözümünde; Pezzella ve diğ. [29] GA ve TA melezlenmiş bir algoritma, Kunadilok [52] TA algoritması, Liouane ve diğ. [42] TA algoritması, Brandimarte [2] hiyerarşik TA, Saidi-Mehrabad ve Fattahi [43] TA algoritması ve Zribi ve diğ. [53] TA ile GA' dan melezlenmiş bir algoritma sunmuşlardır. Çok amaçlı EATÇ problemlerinin çözümünde Li ve Pan [86] bakım kısıtı altında TA algoritmasını kimyasal reaksiyon algoritmasıyla melezleyerek yeni bir algoritma sunmuşlardır. Diğer bir çalışmada Li ve diğ. [87] etkin komşuluk yapısına sahip iki kural kombinasyonlu TA algoritması geliştirmişlerdir. Baykasoğlu ve diğ. [88] önceden tanımlanmış performans ölçütünü optimize etmek üzere alternatif işlem planlarını göz önünde bulunduran çok amaçlı TA algoritması sunmuşlardır. Özbakır [89] proses plan esnekliğine sahip çok amaçlı EATÇ problemi için çok amaçlı TA algoritması geliştirmiştir. Yazarın önerdiği algoritma, EATÇ problemi için, proses

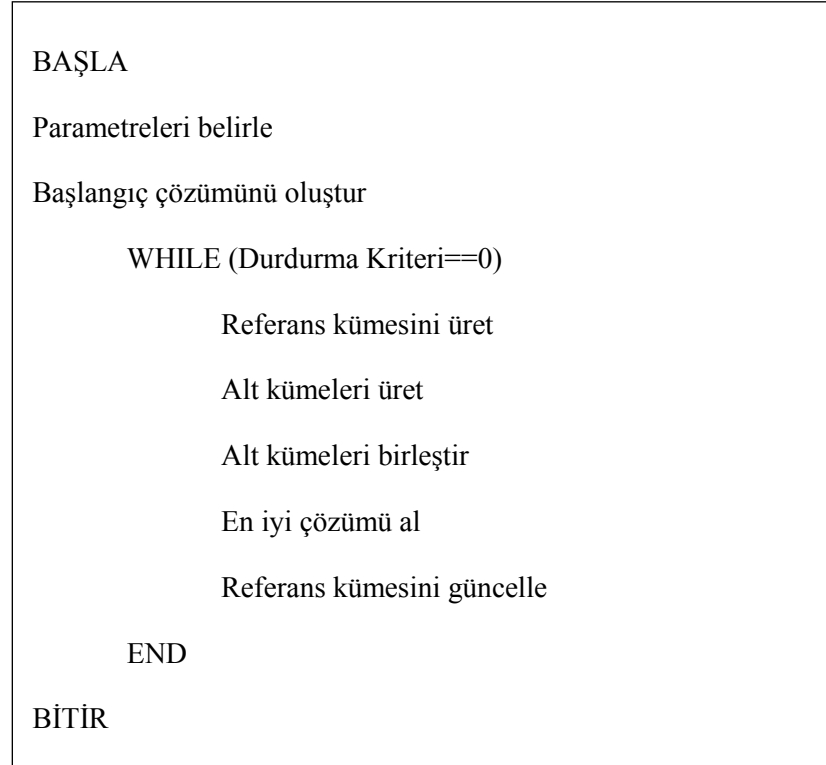
plan seçimi, operasyonların makinelere atanması ve bu makinelerde sıralanması alt problemlerine eş zamanlı olarak çözüm geliştirmektedir. Pareto optimizasyona dayalı çok amaçlı yaklaşım, çözüm sonucunda karar mekanizmasına, birden fazla uygun çözüm arasından seçim yapabilme imkânı sunmaktadır. Rahimi-Vahed ve diğ. [90], PSO ve TA tabanlı melez çok amaçlı algoritma geliştirmişlerdir.

2.1.4. Dağınık arama metodu

Dağınık arama (DA) metodu yeni çözümler elde edebilmek için referans küme adı verilen bir kümedeki çözümleri birleştirir. DA Metodunun diğer EA metotlarından temel farkı, çözümü seçme yoludur. DA metodunda sistematik olarak referans kümesinden iki ya da daha fazla çözüm seçilerek yeni sonuçlar üretilir.

DA metodu GA'ya benzer özelliklere sahiptir. Tamsayı vektörlerinden oluşan bir popülasyon üzerine kuruludur. Yeni popülasyonlar oluşturulurken, seçim, kombinasyon, tamsayı vektörü transformasyonu ve eleme metotları kullanılır. DA metodunda, GA'dan farklı olarak, yeni bir vektör oluşturulmasında ikiden fazla vektör eşleşebilir. GA'da kullanılan çaprazlama yerine çözüm setinin kombinasyonları, mutasyon yerine yeni oluşturulan vektörlerin onarılması veya değiştirilmesi prosedürü vardır. GA'da popülasyon içinden rastgele seçilen iki popülasyon üzerinden genetik üreme, çaprazlama işlemleri yapılırken, DA algoritmasında referans küme içerisinde sistematik biçimde seçilen iki veya daha fazla çözümden yeni çözümler elde edilmektedir. DA'da genellikle 20 veya daha az sayıda çözüm içeren referans küme kullanılır. Kombinasyon çeşitleri olarak, iki adet çözümün kombinasyonu üç adet çözümün kombinasyonu şeklinde yapılabilmektedir.

DA meta sezgiseli için temel prensipler; orijinal vektörlerde bulunmayan bilgilerin elde tutulması ve yeni vektörleri oluşturmak üzere kullanılacak vektörlerin seçiminde ve yeni vektörlerin oluşturulmasında yardımcı sezgisel metotların kullanılmasıdır. DA referans küme olarak adlandırılan bir çözüm kümesi içerisindeki çözümleri birleştirerek yeni çözümler oluşturur. Çözümleri birleştirmede temel mekanizma iki çözümün kombinasyonunu sağlayarak yeni çözümün elde edilmesidir. Bu birleşimde amaç çözümün farklılaştırılması ve iyileştirilmesine yardımcı olmaktır [91]. Şekil 2.4' te DA meta sezgiseline ait algoritma adımları verilmiştir.



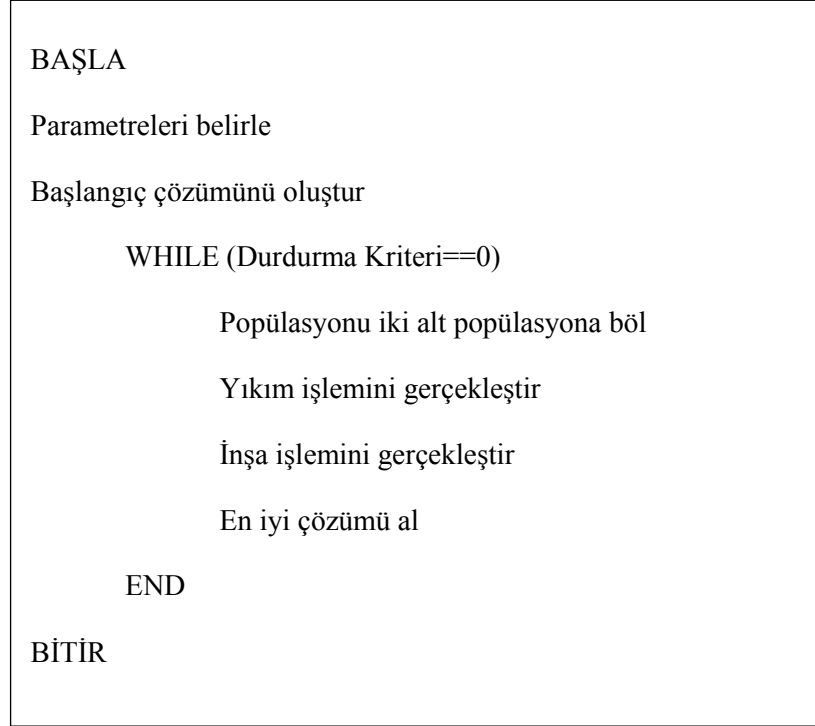
Şekil 2.4. Dağınık arama algoritma adımları

2.1.5. Doyumsuz algoritma

Doyumsuz algoritmalar, sadece şu an için elinde olan bilgiyle, bu bilginin gelecekte ne gibi etki doğuracağını tam düşünmeden, karar alan bir algoritmadır. İşlem adımlarında, şu an için mantıklı gözükten bir karar, belki de gelecek durumlar için daha kötü bir sonuç doğurabilecektir.

Doyumsuz Algoritma çalışırken çözüme küçük adımlarla yaklaşır ve her adım sonunda birden çok seçenek çıkabilir. Algoritma, her adımda bu seçeneklerden en iyi olanını seçer. Her zaman en iyiyi seçerek ilerlemek, sonuçta optimum çözümün elde edilmesini sağlayabilir. Ama bu her zaman için geçerli olmayabilir. Doyumsuz Algoritma, kombinasyonel optimizasyon problemleri için diğer sezgisel yaklaşımlardan daha hızlı bir yaklaşım metodudur. Doyumsuz Algoritma metodu genelde akış tipi problemleri çözmek için kullanılır. Akış tipi ile paralel makine problemlerinin bir araya gelmesinden oluşan esnek akış tipi problemleri çözmek için kullanılan Doyumsuz algoritma, paralel olmakta ve Paralel Doyumsuz Algoritma ortaya çıkmaktadır [92].

Doyumsuz Algoritma genellikle iki aşamada uygulanır. Bunlar; yıkım ve inşa aşamalarıdır. Yıkım aşaması süresince bazı işler, bulunan çözümlerden çıkartılır. İnşa aşamasında ise; önceden çıkartılmış işler, inşa yöntemleri kullanılarak iş sırasına tekrar alınır. Paralel doyumsuz algoritma adımları Şekil 2.5’ de verildiği gibidir.

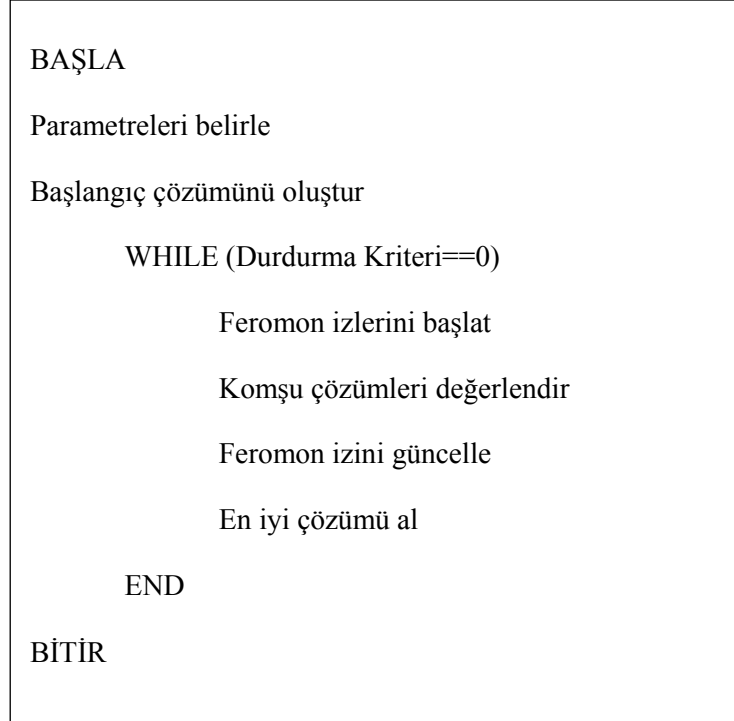


Şekil 2.5. Paralel doyumsuz algoritma adımları

2.1.6. Karınca kolonileri algoritması

Karınca kolonileri yöntemi, karıncaların doğal ortamlarda yiyecek toplama faaliyetlerinden esinlenerek geliştirilmiş, popülasyon tabanlı meta sezgisel bir optimizasyon tekniğidir. Karınca Sistemi ile arama uzayı içerisinde optimum sonucu arayan ajanlar arasında iletişim sağlanarak, arama esnasında yapılan iyi seçimlerden diğer ajanların da haberdar olması sağlanmış ve böylece çok ajanlı bir rastgele aramadan çok daha iyi sonuçlar elde edilmesi mümkün kılınmıştır. Birden fazla ajanın arama uzayında rastgele arama yapmasının ötesinde aralarındaki sinerjinin ortaya çıkması sağlanmıştır. Karınca sisteminin en önemli bileşenlerinden biri izlerin yönetimidir. Standart bir karınca sisteminde izler yeni çözümler oluşturmak için bir rehber olarak kullanılır. Bir çözüm elde edildiğinde önce tüm izler buharlaşmayı temsil etmek üzere zayıflatılır, sonra çözüm yolu üzerinde bulunan izler çözümün

kalitesine bağılı oranda güçlendirilir [62]. Şekil 2.6’ da KKA ait işlem adımları verilmiştir.



Şekil 2.6. Karınca kolonileri algoritma adımları

Çizelgeleme problemlerinin KKA ile çözümüne yönelik yapılan çalışmalarda, Bauer ve diğ. [93] tek makineli bir üretim sisteminde toplam gecikmeyi minimize etme amaçlı çalışmışlardır. Silva ve diğ. [94] paralel makine çizelgeleme problemi ve Rajendran ve Ziegler [95] akış tipi çizelgeleme problemlerinin çözümü için KKA kullanmışlardır. Çok amaçlı EATÇ problemlerinin çözümünde Li ve diğ. [96] melez KKA geliştirmişlerdir. Yine aynı problemin çözümü için Xing ve diğ. [35] tarafından çift katmanlı KKA literatüre kazandırılmıştır.

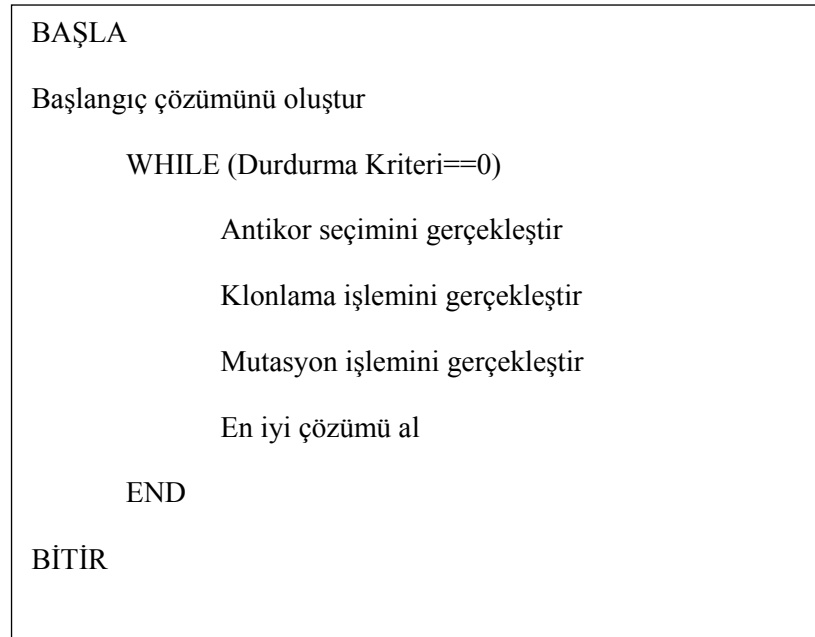
2.1.7. Yapay bağışıklık sistemi

Doğadaki tüm canlılar kendi vücudundan olmayan doku, hücre ve moleküllere karşı savunma mekanizmalarına sahiptirler. İnsan vücudunun hastalıklara karşı savunma mekanizmasını oluşturan bağışıklık sistemi, insan vücudunu her türlü zararlı ve yabancı maddelere karşı korur. Bağışıklık sistemi yabancı maddeler tarafından uyarıldığında derhal harekete geçer ve yabancı olarak algıladığı bu organizmalarla karşılaşınca, belirli hücreler yardımıyla bunlardan kurtulmak için derhal savaşıma

başlar. YBS, doğal sistemden esinlenerek ortaya konulmuş bazı model ve algoritmalarından oluşmaktadır. Bu modeller arasında yer alan bağışıklık ağ modeli, antikor ağ modeli ve özellikle klonal seçim algoritması, optimizasyon alanındaki problemin çözümlerinde başarıyla kullanılmaktadır.

Çizelgeleme problemlerinde YBS'nin uygulanma adımı, çizelgelenecek işler vücuda giren antijen, elde edilecek çözümlerde antikor olarak tanımlanır. Problemin amaç fonksiyonu, antikorum uyum değeridir. Popülasyon parametresi büyüklüğüne göre üretilen çözümler kümesi, antijenler için timüste oluşturulan hücre repertuarına karşılık gelmektedir. Bu hücre repertuarındaki antikorlar, mutasyon işlemine tabi tutularak daha iyi amaç fonksiyonu değerine sahip yeni antikorlar üretilmektedir. Bu aşamada, hücre repertuarında antijenlerle kötü uyum gösteren antikorlar, hücre repertuarından silinmekte yerlerine yeni antikorlar oluşturulmaktadır. Bu işlem karar verici tarafından belirlenen durma koşulu sağlanıncaya kadar devam eder.

YBS uygulamalarında, negatif seçim, klonal seçim v.b. mekanizmalar problemin özelliğine göre taklit edilerek, optimizasyon probleminin çözümü sağlanmaktadır. Bu mekanizmalardan en çok uygulanan olanları ise klonal seçim mekanizması ve negatif seçim mekanizmasıdır. YBS algoritmasının adımları Şekil 2.7' de verildiği gibidir.



Şekil 2.7. Yapay bağışıklık sistemi algoritma adımları

2.1.8. Yapay sinir ağıları

İnsan beyni öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yeteneklere sahip gizemli ve karmaşık bir organdır. Yapay sinir ağıları (YSA), insan beyninin işleyişini taklit ederek, zekice davranışların simülasyon ortamında gerçekleştirilmeye çalışıldığı yöntemdir. YSA, bir sisteme ilişkin tek veya çoklu parametrelere bağlı olarak tanımlanan veri girişleri ile çıkışları arasında ilişki kurabilme yeteneğine sahiptir.

YSA olay örneklerini incelemekte, genellemeler yapmakta, bilgiler toplamakta ve karşılaştığı diğer olaylar hakkında önceden elde ettiği bilgileri kullanarak karar verme mantığına dayanmaktadır. YSA, biyolojik sinir ağlarının işleyiş yöntemidir. Sinir sisteminde bulunan sinir hücreleri, nöronlar içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağ oluştururlar. Oluşturulan bu ağlar, öğrenme, hafızaya alma ve veriler arasında ilişki çıkarma özelliğine sahiptirler. Bu ağlar, öğrenme yöntemlerine göre birkaç gruba ayrılır. Literatürde en çok bilineni eğitici öğrenmedir. Bu yöntem, giriş ile çıkış arasındaki ilişkinin belirlenmesinin önemli olduğu problemlerde kullanılmaktadır. Optimizasyon problemlerinde ise çıkış değerinin ağ yapısının amaç fonksiyonunda yaptığı etkiye göre belirlendiği, öğreticisiz öğrenmenin yapıldığı sinir ağları kullanılmaktadır.

Sinir ağı içerisinde üç değişik türde nöron bulunur. Girdi nöronları, dış ortamdan kodlanmış bilgileri alır. Çıktı nöronları, probleme cevap niteliğinde olan kodlanmış bilgiyi dış ortama gönderir. Saklı nöronlar ise girdi bilgilerini işleyerek çıktı bilgilerini oluşturur. Sinir ağlarında en fazla kullanılan model, hiyerarşik katmanlarda gruplanmış nöronlardan oluşan çok katlı algılama modelidir.

2.2. Parçacık Sürü Optimizasyonu

Doğadaki hayvan sürülerinin davranışları insanların ilgisini çekmiştir. Kuş sürülerinin havada uçuşması, karıncaların yuvalarına yem taşıması, balık sürülerinin denizde yüzmeleri, arıların bal yapması şeklinde örnekler çoğaltılabilir. Araştırmacılar son yıllarda yaptıkları çalışmalarda, bu sürü davranışlarının nasıl modellenebileceği ve sürü bireylerinin aralarındaki iletişimin nasıl yapıldığının çözümlenmesi üzerine durmuşlardır. Bu çalışmalar, insanların günlük hayatta

karşılaştıkları problemlere çözüm üretme çabalarına ilham kaynağı olmuştur. İnsanların karşılaştıkları problemlere çözüm üretmek için, sürülerden esinlenerek kullandıkları popüler algoritmalarından bir tanesi PSO algoritmasıdır. 1995 yılında James Kennedy ve Russel Eberhart [97] tarafından önerilmiş bu yöntem aslında sosyal yaşamın basitleştirilmiş bir benzetimidir. Yöntemin benzerlerine olan üstünlükleri ve gelişime açık olmasından dolayı son yıllarda optimizasyon problemlerinin çözümünde kullanılmasını sağlamıştır.

PSO kuş sürülerinin yiyecek araması ve yiyecek bulduktan sonra birbirlerinden yararlanarak sürü halinde yiyeceğe yönelmelerini konu alan modelledir. Kuşlar yiyecek ararken aralarındaki sosyal iletişim ağı ile konumlarına göre yer değiştirirler. Burada en önemli nokta yiyeceği bulan kuşun diğer kuşlara rehberlik etmesi ve sürü üyeleri arasındaki bilgi paylaşımıdır. Kuşların yerini bilmedikleri bir yiyeceği aramaları, optimizasyon problemlerinde çözüm aramaya modellenmiştir. Bu modelde; parçacık olarak adlandırılan her bir kuş bir çözümü temsil etmektedir. Kuşun her iterasyonda yer değiştirmesi değişim, kuş sürüsündeki tüm kuşlar ise popülasyon olarak ifade edilmektedir.

PSO doğrusal olmayan problemlerin çözümü için tasarlanmıştır. Çok parametrelili ve çok değişkenli optimizasyon problemlerine çözüm bulmak için kullanılmaktadır. PSO, GA'lar gibi evrimsel hesaplama teknikleriyle birçok benzerlik gösterir. Sistem rasgele çözümler içeren bir popülasyonla başlatılır ve nesilleri güncelleyerek en optimum çözümü araştırır. PSO' da parçacık olarak adlandırılan olası muhtemel çözümler, o andaki optimum parçacığı izleyerek problem uzayında dolaşırlar. PSO'nun klasik optimizasyon tekniklerinden en önemli farklılığı türev bilgisine ihtiyaç duymamasıdır. Bu özellik birçok problemin çözümü için gerekli olan karmaşık işlem yükünün hafifletilmesini sağlamaktadır. PSO'yu uygulamak, algoritmasında ayarlanması gereken parametre sayısının az olması sebebiyle oldukça basittir. PSO; fonksiyon optimizasyonu, bulanık sistem kontrolü, yapay sinir ağı eğitimi gibi birçok alanda başarıyla uygulanabilmektedir [98].

PSO'da genel çözüm süreci Şekil 2.8' de verilmiştir. İlk adım olarak kullanılacak parametreler belirlenir. İkinci adımda başlangıç popülasyon değerleri rastsal olarak belirlenir ve her bir parçacık için önceden tanımlı amaç fonksiyonu hesaplanır.

Başlangıç adımındaki her bir çözüm aynı zamanda o parçacık için yerel en iyi değerlerdir. Tüm sürü içindeki en iyi amaç fonksiyonu değerine sahip olan parçacık ise global en iyi değer olarak atanır. Global en iyi değer global komşu olarak, yerel en iyi değerler ise, yerel komşular olarak hafıza da tutulur. Genelde başlangıç çözümünde başlangıç hızları rastgele belirlenmektedir. Başlangıç çözümünden elde edilen hız değerlerinden yararlanılarak her parçacığa ait yeni pozisyonlar belirlenir. Yeni pozisyonlara göre her parçacık için, amaç fonksiyonu değeri yeniden hesaplanır ve hızlar güncellenir. Her parçacık için o iterasyona kadar bulunan en iyi amaç fonksiyonu değerini veren çözüm yerel en iyi değer olarak hafızada tutulur. Tüm parçacıklar için elde edilen yerel en iyi değerler, yerel en iyi komşular kümesi oluşturulur. Her iterasyonda bulunan global en iyi değerler karşılaştırılarak en iyi amaç fonksiyonu değerine sahip parçacık çözümü ise global en iyi değer olarak atanır. Algoritma durdurma kriteri sağlanana kadar devam ettirilir.

```
BAŞLA
Parametreleri belirle
Başlangıç çözümünü oluştur
    WHILE (Durdurma kriteri==0)
        Amaç fonksiyon değerini bul
        Yerel en iyiyi bul
        Global en iyiyi bul
        Hız vektörünü güncelle
        Pozisyon vektörünü güncelle
        En iyi çözümü al
    END
BİTİR
```

Şekil 2.8. Parçacık sürü optimizasyonu algoritma adımları

d Adet parametre ve n adet parçacıktan oluşan popülasyon matrisi aşağıdaki gibi ifade edilir:

$$X = \begin{bmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nd} \end{bmatrix} \quad (2.1)$$

(2.1)'de verilen matriste i. parçacık X_i^k pozisyon vektörü olarak tanımlanır ve Denklem (2.2)'deki gibi ifade edilir;

$$X_i^k = [x_{i1}^k, x_{i2}^k, x_{i3}^k, \dots, x_{ij}^k] \quad (2.2)$$

Burada, x_{i1}^k , k. iterasyonda i. parçacığın pozisyon vektöründeki birinci elemanını ifade eder.

i. parçacığın hızı; V_i^k ile ifade edilir. Burada yer alan v_{i1}^k , k. iterasyonda i. parçacığın hız vektöründeki birinci elemanını ifade eder.

$$V_i^k = [v_{i1}^k, v_{i2}^k, v_{i3}^k, \dots, v_{ij}^k] \quad (2.3)$$

En iyi uygunluk değerini veren i. parçacığın pozisyonu, P_{ibest} şeklinde gösterilir ve Denklem (2.4)'de olduğu gibi ifade edilir. Burada p_{i1} , i.yerel en iyi vektörünün 1. değerini göstermektedir. Tüm parçacıklar için global en iyi değer vektörü, G_{best} şeklinde Denklem (2.5)'deki gibi ifade edilir. Burada g_1 global en iyi değer vektörünün 1. değerini gösterir.

$$P_{ibest} = [p_{i1}, p_{i2}, p_{i3}, \dots, p_{ij}] \quad (2.4)$$

$$G_{best} = [g_1, g_2, g_3, \dots, g_j] \quad (2.5)$$

PSO'da genelde başlangıç adımı ve başlangıç hız vektörleri rastsal olarak başlatılır. İlk iterasyonda her parçacığa ait konumlar aynı zamanda yerel en iyi değer (P_{best}), sürü içindeki en iyi konuma sahip parçacık ise global en iyi değer (G_{best}) alınarak, bir sonraki iterasyonda parçacıkların konum vektörleri aşağıdaki denklemlere göre güncellenir;

$$V_i^{k+1} = W \times V_i^k + C_1 \times \text{Rand}_1^k \times (P_{best}^k - x_i^k) + C_2 \times \text{Rand}_2^k \times (G_{best}^k - x_i^k) \quad (2.6)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2.7)$$

Denklem (2.6)' da her bir parçacığa ait hız vektörü, Denklem (2.7)'de yeni konum vektörünün hesaplanması gösterilmiştir. Yeni hız vektörleri ve konum vektörleri

durdurma kriteri sağlanıncaya kadar devam ettirilir. Denklem (2.6)'da gösterilen c_1 ve c_2 sabitleri öğrenme faktörleri olarak tanımlanır. c_1 sabiti kendi tecrübelerine göre en iyi pozisyona hareket etmesini sağlar. c_2 sabiti ise, sürüdeki diğer parçacıkların tecrübesine göre parçacıkların en iyi hareket etmesini sağlar. Bu iki faktörün küçük değer seçilmesi parçacıkların hedef bölgeye doğru hareket etmeden, uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresini arttırma dezavantajı vardır. Yüksek değerler seçilmesi ise, parçacıkların hedefe ulaşmalarını hızlandırırken, hedef bölgenin dışına taşınmasına neden olabilir. Araştırmacılar, yaptıkları çalışmalarda $c_1=c_2=2$ olarak alınmasının iyi sonuçlar verdiğini belirtmişlerdir. Denklem (2.6)'daki $Rand_1$ ve $Rand_2$ ise, $[0,1]$ arasında düzgün dağılımlı rasgele sayılardır. Denklem (2.7) parçacığın bir sonraki iterasyonda gideceği konumu vermektedir.

PSO' nun her bir iterasyonunda iyi çözümler hakkındaki bilginin sürünün tamamına yayılması ile sürüdeki tüm parçacıkların arama uzayındaki daha iyi alanlara yönelmesi sağlanır. Şekil 2.9' da gösterildiği gibi yeni mevcut iterasyonda elde edilen P_{best} ve G_{best} 'e göre hız vektörü güncellenmekte ve parçacığın bir sonraki iterasyonda daha iyi sonuç alabileceği bir konuma geçmesi sağlanmaktadır. Şekil 2.9' da gösterilen notasyonlar aşağıdaki gibi tanımlanmıştır;

X_i^t : i. parçacığın t. iterasyondaki konumu,

V_i^t : i. parçacığın t. iterasyondaki hızı,

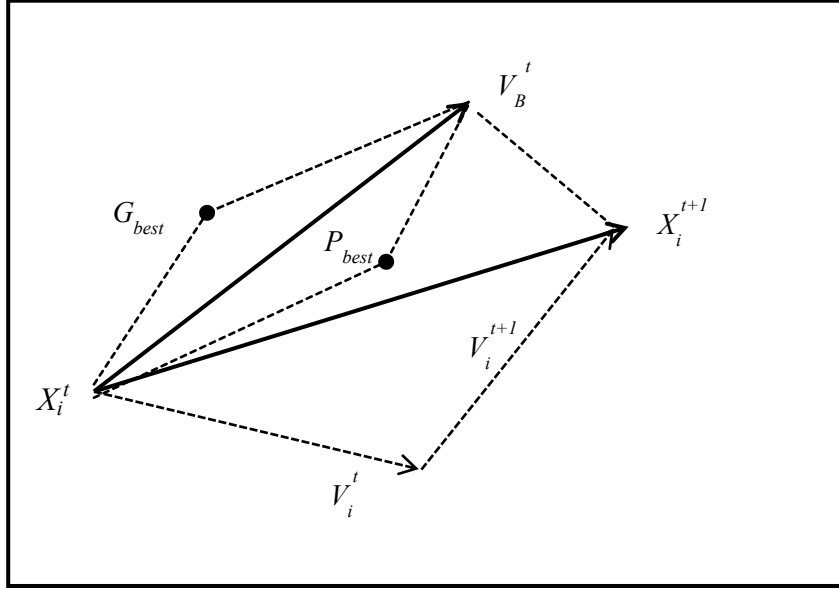
G_{best} : Mevcut iterasyona kadar sürüdeki en iyi konuma sahip parçacığın konumu,

P_{best} : Mevcut iterasyona kadar i. parçacığın en iyi konumu,

V_B^t : t. iterasyonda G_{best} ve P_{best} değerlerinin bileşkesi,

V_i^{t+1} : i. parçacığın (t+1) iterasyondaki yeni hızı,

X_i^{t+1} : i. parçacığın (t+1). iterasyondaki yeni konumu.



Şekil 2.9. Parçacık sürü optimizasyonunda parçacık hareketi

2.2.1. PSO terimleri ve parametreleri

PSO’da kullanılan çeşitli terimler ve parametreler bu başlık altında incelenecektir.

Parçacık vektörü: Sürüde yer alan her bir kuş parçacık olarak tanımlanmaktadır. Her parçacığın kendine ait pozisyon vektörü ve hız bileşeni vardır. Pozisyon vektörü X_i^k ile gösterilir. Problem boyutu j adet parçacıktan meydana gelsin. Denklem (2.8) başlangıç iterasyonunda her bir parçacık için konumların belirlenmesini Denklem (2.9) başlangıç iterasyonunda her bir parçacık için hız vektörünü vermektedir.

$$X_i = X_{\min} + (X_{\max} - X_{\min}) \times \text{random} () \quad (2.8)$$

$$V_i = V_{\max} \times \text{random} () \quad (2.9)$$

PSO’ da her bir parçacığın yeni konumu problem için yeni bir çözüm önerisidir. Parçacık hareketlerinin çözüm uzayının dışına taşmaması için belirli kısıtlamalar getirilmesi gerekir. Her bir parçacığın vektör değerlerinin $[X_{\min}, X_{\max}]$ olmak üzere, Denklem (2.10) bu kısıtlamaları vermektedir.

$$f(x) = \begin{cases} X_i = X_{\max}, & X_i > X_{\max} \\ X_i = X_{\min}, & X_i < X_{\min} \end{cases} \quad (2.10)$$

Populasyon büyüklüğü (Parçacık sayısı): Sürüde yer alan kuş sayısı, parçacıkların tamamına verilen addır. Populasyon büyüklüğü çözümlenecek probleme göre

değişiklikler gösterir. Popülasyon sayısının büyük seçilmesi daha hızlı sonuç elde etme imkânı sağlayacaktır.

Algoritmada kullanılan parçacık sayısı algoritmanın arama uzayında daha etkin bir arama yapmasına imkân tanır. Parçacık sayısının artması, çözüm uzayında daha az iterasyon ile optimum çözüme ulaşılmasını sağlar. Ancak çözüme ulaşma süresinde artışa neden olabilir[97].

Hız Vektörü: PSO’ da her bir parçacığın kendine ait hızı vardır. Her parçacık için tanımlanan bu hız sayesinde, parçacığın mevcut konumu ile hedeflenen konum arasında çözümler aranır. Parçacık hızının doğru bir şekilde belirlenmemiş olması, çözüm uzayının yetersiz araştırılmasına ve yerel minimumda takılmaya neden olabilir.

Hız, parçacığın bir sonraki pozisyonunu bulmada en önemli faktördür. Hız değeri pozitif ve negatif değerler olarak parçacıkların çözüm uzayında çok yönlü hareket etmelerini sağlar. Hız değerlerinin kontrol edilmediği durumlarda parçacıkların çözüm alanının dışına çıkma veya uygun olmayan bölgelerde arama yapma riskleri vardır. Bunu önlemek için hız değerine V_{max} gibi kısıtlayıcı bir limit konulmuştur. V_{max} değerinin belirlenmesi problemlere göre değişiklik gösterir. Bazı araştırmacılar $V_{max} = X_{max}$ olarak almışlardır[99]. Bazı uygulamalarda ise Denklem (2.11)’ de verildiği gibi V_{max} değeri arama uzayındaki parçacığın konum vektörünün her bir boyutunun alabileceği en büyük değer ile en küçük değer farkının [10,20] aralığındaki bir değerle çarpımından elde edilmektedir[100]. Bir parçacığın hız değeri limitini aşması durumunda Denklem (2.12)’ deki gibi bir kısıtlama getirilir.

$$V_{max} = (X_{max} - X_{min}) \times \%R, R \in [10,20] \quad (2.11)$$

$$f(x) = \begin{cases} V_i = V_{max}, & V_i > V_{max} \\ V_i = -V_{min}, & V_i < -V_{min} \end{cases} \quad (2.12)$$

Atalet Ağırlığı: Atalet ağırlığı, bir önceki iterasyona ait hız vektörünün mevcut iterasyona ait hız vektörü üzerindeki etkisini kontrol eder ve w şeklinde ifade edilir. Atalet ağırlığının parçacıkta global ve yerel araştırma kabiliyeti arasında değişim etkisi vardır. Eğer atalet ağırlığı yüksek bir değer alınırsa parçacık global bir şekilde

arama gerçekleştirir. Bu değerin küçük alınması ise yerel arama yapılmasını sağlar[96].

Atalet ağırlığı parçacığın çözüm uzayındaki hareket yönünün kontrol altında tutulmasını sağlar. Literatürde atalet ağırlığı bazen hız vektörüyle birlikte bazen de hız vektörü olmadan kullanıldığı görülmektedir. Atalet ağırlığının büyük bir değer seçilmesi global aramada iyileşmeyi, küçük seçilmesi ise yerel aramada iyileşmeyi sağlar.

Parçacığın Tecrübesi (c_1) ve Sürünün Tecrübesi (c_2) (Öğrenme Faktörleri): Algoritmanın performansını etkileyen en önemli parametrelerden iki tanesidir. Parçacığın tecrübesi c_1 olarak ifade edilir ve sosyal parametre olarak da adlandırılır. Parçacığın önceki iterasyonlarda elde ettiği en iyi sonuçların, bir sonraki iterasyona etkisidir. Sürünün tecrübesi c_2 olarak ifade edilir ve kavramsal parametre olarak da adlandırılır. Sürüde önceki iterasyonlarda en iyi sonuca sahip parçacığın sonuçlarının bir sonraki iterasyona etkisidir. Hız vektörü ve bu iki parametrenin birleşmesi ile parçacığın çözüm uzayında hareket etmesi sağlanır.

c_1 ve c_2 parametrelerinin büyük değerler seçilmesi, parçacıkların birbirlerinden uzaklaşıp ayrılmalarına sebep olurken, küçük değerler seçilmesi ise parçacık hareketlerinin kısıtlanmasına ve çözüm uzayının yeterince taranamamasına neden olur. Bu parametre değerleri problem türüne göre değişiklik göstermesine rağmen genelde yazarların $c_1=c_2=2$ olarak aldıkları görülmektedir. Ancak yazarların c_1 ve c_2 parametrelerinin birbirlerinden farklı değerler aldıkları da görülmektedir. $c=c_1 + c_2$ durumunda c ne kadar büyük değer alınırsa, algoritmanın optimum değer etrafında yaptığı salınım miktarı artmaktadır.

Yerel En İyi Değer (P_{best}) ve Global En İyi Değer (G_{best}): PSO yönteminde iki tür en iyi değer kavramı vardır. Birincisi, bireyin o ana kadar elde ettiği en iyi değeri ifade eden, P_{best} , diğeri popülasyonun o ana kadar elde ettiği, global en iyi değeri ifade eden G_{best} 'tir. PSO'da her iterasyonda elde edilen en iyi değerler hafızada tutulur.

Bir parçacığın iterasyonlar boyunca kendi geçmişindeki değerlere bakarak bulduğu en iyi değerine, kişisel en iyi veya yerel en iyi değer denir. Yerel en iyi değer aynı zamanda parçacığın geçmiş tecrübelerini gösteren bir değerdir.

P_{best} , bir parçacığın bulunduğu iterasyona kadar elde ettiği en iyi amaç fonksiyonu değerine sahip parçacıktır. Sürüde bulunan parçacık sayısı kadar P_{best} bulunur. Yerel en iyi komşular, her iterasyonda yerel en iyi değerin sahip olduğu pozisyon vektörüdür. Her bir iterasyonda parçacık bulduğu değeri geçmişte bulduğu en iyi değerle kıyaslar. Eğer yeni konum değeri önceden bulduğu en iyi konum değerine göre daha iyi ise yeni konum değeri yerel en iyi değer olur.

Bir parçacığın komşuları, sürü içerisindeki belirli sayıda parçacıktan oluşmaktadır. Böylece tüm sürü içerisindeki en iyi parçacık ile birlikte, parçacıklar için değişiklik gösterebilecek alt sürülerin en iyi parçacığına göre hız güncellemesi gerçekleştirilir. Daha yavaş sonuca ulaşmasına rağmen yerel optimuma takılma riski daha azdır.

G_{best} Popülasyon içindeki tüm parçacıklar arasından elde edilen en iyi amaç fonksiyonu değerine sahip parçacıktır. Global en iyi komşular, her iterasyonda bulunan global en iyi değere sahip parçacığın pozisyon değerleridir. Bir çözüm uzayında en fazla iterasyon sayısı kadar G_{best} vardır.

Sürü içerisindeki bütün parçacıklar birbirinin komşusudur. Genelde ilk iterasyonda rastgele ya da sezgisel olarak dağıtılmış olan parçacıklar, tüm sürü ile haberleşirler ve en iyi uygunluk değerine sahip olan parçacık G_{best} olarak seçilir. Sürüdeki diğer parçacıklar ise hareket yönlerini G_{best} 'e göre yeniden belirlerler ve bir sonraki iterasyonda bu bilgi ışığında hareket ederler. Bu hareket sonucunda parçacıkların, arama uzayının en iyi kısmına doğru yönlendiği gözlemlenir ve hızla sonuca ulaşacağı düşünülebilir. Ancak optimum nokta G_{best} değerinin yakınında değilse, sürüdeki diğer parçacıkların G_{best} etrafında toplanacağı için sürünün yerel optimum noktasında takılma ihtimali vardır.

Uygunluk Değeri: Parçacıkların her iterasyonda bulundukları pozisyona göre hesaplanır. Optimizasyon problemlerindeki amaç fonksiyonu bu yöntemde uygunluk değeri olarak ifade edilmektedir. Uygunluk değeri aynı zamanda çözümün kalitesini belirler. P_{best} ve G_{best} değerleri en iyi uygunluk değerine sahip parçacıklardan seçilir.

Durdurma Kriteri: Diğer yöntemlerde olduğu gibi PSO yönteminde de belirlenen iterasyon sayısına ulaşıncaya veya belirlenen amaç fonksiyon değeri elde edilinceye kadar döngü çalıştırılır. Algoritmanın sonlandırılma kriteridir.

Sonlandırma kriteri çözülen problem tipine bağlıdır. Genellikle, algoritma sabit sayıda fonksiyon değerlendirmesine (sabit iterasyon sayısına) kadar ya da belirlenen bir hata sınırına ulaşıncaya kadar çalıştırılır. Literatürde 13 çeşit sonlandırma kriteri aşağıda isimleri verilen 6 sınıfta toplanabilir [101]:

1. Referans kriteri: Sürünün belirli bir yüzdesi bir optimuma yakınsamışsa sonlandırılır.
2. Tükenme tabanlı kriter: Belirli nesil sayısı, amaç fonksiyon değerlendirme sayısı ya da merkezi işlem birimi zamanı sonlandırma için kriter sayılır.
3. İyileşme tabanlı kriter: Eğer küçük ilerlemeler elde ediliyorsa algoritma sonlandırılabilir. Bunun da farklı versiyonları vardır:
 - 3.a. En iyi amaç fonksiyon değeri, belli nesil sayısı boyunca belirli bir eşiğin altındaysa sonlandırılır.
 - 3.b. Ortalama amaç fonksiyon değeri belli nesil sayısı boyunca belirli bir eşiğin altındaysa sonlandırılır.
 - 3.c. Belirli bir nesil boyunca herhangi bir komşulukta bir ilerleme olmuyorsa sonlandırılır.
4. Hareket tabanlı kriter: Bireylerin hareketine bağlı sonlandırma kriteridir ve iki versiyonu bulunmaktadır:
 - 4.a. Ortalama amaç fonksiyon değerine bağlı olarak (amaç uzayı) popülasyondaki hareket, belirli nesil sayısı için belirli bir eşiğin altındaysa sonlandırılır.
 - 4.b. Pozisyonlara bağlı olarak (parametre uzayı) popülasyondaki hareket, belirli nesil sayısı için belirli bir eşiğin altındaysa sonlandırılır.
5. Dağılım tabanlı kriter: Bireyler birbirine çok yakınsa yakınsamaya ulaşılmıştır. Bu kriterin de, ilk üçü parametre uzayında sonuncusu da amaç uzayında uygulanan dört versiyonu bulunmaktadır:
 - 5.a. Her vektörden en iyi popülasyon vektörüne olan uzaklık, belirli bir eşiğin altındaysa sonlandırılır.

5.b. Popülasyonun en iyilerinin belirli bir yüzdesinin en iyi popülasyon vektörüne olan uzaklığının, belirli bir eşiğin altında olup olmadığı test edilir. Bir önceki versiyon tüm popülasyon üyelerinin yakınsamasını bekler. Ancak optimum değer zaten bulunmuşsa fazladan hesapsal işlem yapılacaktır. Bu yöntem diğerine göre daha iyi olabilir.

5.c. Vektörlerin standart sapması belirli bir eşiğin altındaysa sonlandırılır.

5.d. En iyi ve en kötü amaç fonksiyon değeri arasındaki fark belirli bir eşiğin altındaysa sonlandırılır.

6. Birleşik kriter: Bu kriter de iki şekilde uygulanabilir:

6.a. Eğer belirli bir nesil boyunca ortalama iyileşme, belirli bir eşiğin altındaysa maksimum uzaklığın belirli eşiğin altında olup olmadığı test edilir.

6.b. En iyi ve en kötü amaç fonksiyon değerleri arasındaki fark, belirli bir eşiğin altındaysa en iyi bireylerin belirli bir yüzdesinin en iyi çözüm olan maksimum uzaklığının belirli bir değerin altında olup olmadığı test edilir (en az belirli bir yüzdedeki bireyler uygun bölgede olmalıdır).

2.2.2. Çeşitli PSO yapıları

Kennedy ve Eberhart [97] tarafından ilk defa önerilmiş PSO yönteminde parçacıkların önceki hızları üzerinde bir kontrol mekanizması yoktu. Algoritmada görülen eksikliklerin giderilebilmesi için araştırmacılar PSO algoritmasında geliştirmeler yapmışlardır. Aşağıda öncelikle PSO ikili versiyonu daha sonra da farklılaştırılmış PSO yapıları anlatılacaktır.

2.2.2.1. İkili PSO

Parçacık sürü optimizasyonu ikili versiyonu Kennedy ve Eberhart [102] tarafından geliştirilmiştir. Algoritmanın ikili versiyonu, doğasında ikili kodlu olan problemleri temsil etmesi yönünden kullanışlıdır. Bu ikili versiyon x_i ve y_i bileşen değerlerini $\{0,1\}$ kümesindeki elemanlardan alınacak şekilde kısıtlar. Hız değeri de $[0,1]$ aralığında sınırlandırılır ve olasılık olarak ele alınır. Sınırlandırma sigmoid fonksiyonu kullanılarak Denklem (2.13)' de olduğu şekilde yapılır.

$$\text{Sig}(x) = \frac{1}{1+\exp(-x)} \quad (2.13)$$

İkili sürüde kullanılan hız vektörü için güncelleme Denklem (2.14) şeklinde olur. Buradaki hız vektörü klasik PSO'daki hız vektöründen farklı değildir.

$$V_i^{k+1} = V_i^k + C_1 \times \text{Rand}_1^k \times (Pbest_i^k - x_i^k) + C_2 \times \text{Rand}_2^k \times (Gbest^k - x_i^k) \quad (2.14)$$

Genel pozisyon güncelleme denklemi yerine Denklem (2.15)' deki gibi yeni bir olasılıksal güncelleme denklemi kullanılır.

$$X_i^{k+1} = \begin{cases} 0 & \text{if } r_{3j} \geq \text{sig}(v_{ij}(k+1)) \\ 1 & \text{if } r_{3j} < \text{sig}(v_{ij}(k+1)) \end{cases} \quad (2.15)$$

Burada $r_{3j} \sim U(0,1)$ düzenli olasılıksal bir değişkendir. Kennedy ve diğ. [103] V_{ij} değerinin ± 4 sınırında alınmasını önermişlerdir. Diğer bir çalışmalarında Kennedy ve Eberhart [102] V_{ij} için daha büyük eşik ± 6 olarak önermişlerdir.

Aynı vektörde hem ikili hem de sürekli değerleri kullanmak mümkündür. Bu versiyon melez sürü [103] olarak adlandırılmıştır. Aynı zamanda ikili-sürekli sürü olarak da kullanılmaktadır. Son zamanlardaki çalışmalar, PSO'nun basitçe ilgili nicelikleri gerekli olduğunda ayırıklaştırarak gelişi güzel ayırık temsilleri de içermesi için kabiliyetlerini genişletmiştir.

2.2.2.2. Yakınsama oranı geliştirmeleri

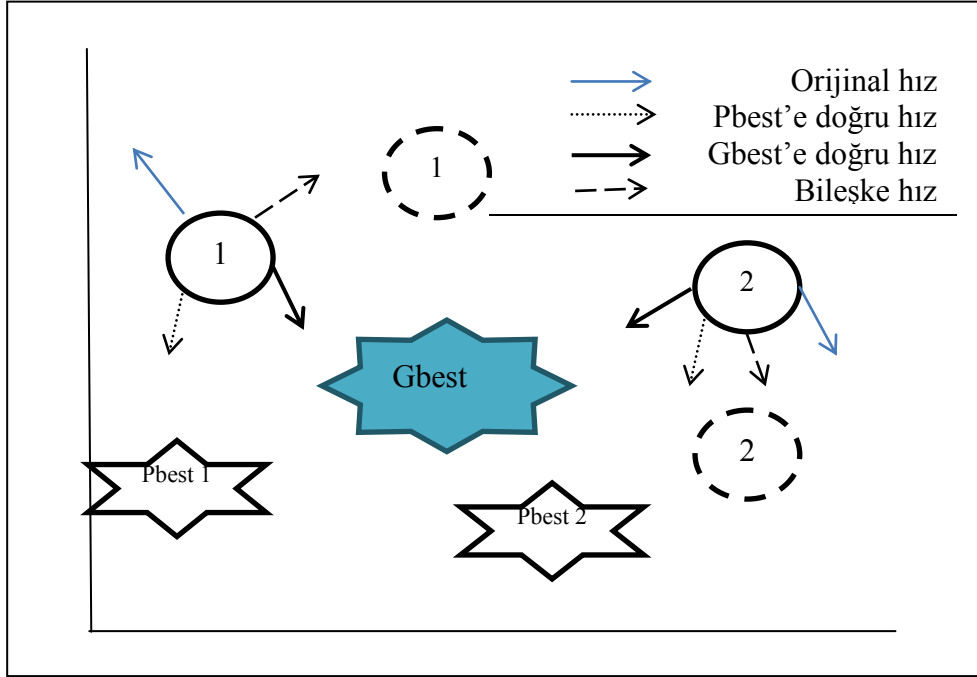
Kennedy ve Eberhart [97] tarafından ilk defa önerilmiş PSO yönteminde parçacıkların önceki hızları üzerinde bir kontrol mekanizması olmaması, algoritmanın optimum noktaları atlamasına, yerel tıkanmalara sebep olabilmekteydi. PSO yakınsama parametrelerini geliştirmek için literatürde birçok yöntem geliştirilmiştir. Bu önerilen geliştirmeler daha çok algoritmanın yapısını değiştirmeden her iterasyonda güncelleme yapılırken kullanılan denklemlerin geliştirilmesi üzerine olmuştur. Shi ve Eberhart [99] bu eksikliği gidermek için atalet ağırlığı yaklaşımını sunmuşlardır. Daha sonra Clerc [104] tarafından daraltma faktörü ve Ratnaweera ve diğ. [105] sürü içerisinde daha etkin YA yapabilmek ve global en iyiye ulaşabilmek için zamana bağlı değişken hızlandırma katsayıları yaklaşımını sunmuşlardır.

Atalet Ağırlığı : Denklem (2.6)' da verilen klasik PSO'da $W \times V_i^k$ kısmı parçacığın, bir önceki hız değerinin belirli bir değer ile ağırlıklaştırılmasıyla hareket etmesi sağlanmaktadır. $w=0$ alınması durumunda, parçacıkların hareketi, sadece mevcut konumu ile parçacığın o ana kadar en iyi konumu ve sürüdeki en iyi konum değeri bilgileri ışığında olacaktır. Bu durumda global en iyi konumda olan parçacık, kendisinden daha iyi bir değere sahip parçacık çıkana kadar sürekli yerinde kalacaktır. Algoritmada çözüm uzayının iterasyonlar boyunca küçüldüğü bir yapı haline gelecektir. Parçacıklar P_{best} ile G_{best} değerleri arasında kalma eğiliminde olacaktır. Bu şekilde etkin bir aramanın olmasından bahsedilemez. Denklem (2.6)' da verilen $W \times V_i^k$ kısım ile parçacıklar hareketlerini çözüm uzayında geniş bir alanda yapacaktır.

Kennedy ve Eberhart [97] tarafından geliştirilen ilk PSO versiyonunda atalet ağırlığı kullanılmamıştır. Atalet ağırlığı, Shi ve Eberhart [99] tarafından geliştirilen PSO'nun sonraki versiyonuna eklenmiştir. Yazarlar, atalet ağırlığını kullanarak parçacığın bir sonraki iterasyondaki hızını, bir önceki hızının etkisi altına almışlardır. Denklem (2.16) başlangıçta büyük değer alan atalet ağırlığının iterasyonlar ilerledikçe daha az değerler almasını sağlayan formülasyonu vermektedir. Bu şekilde, ilk iterasyonlarda parçacıklar daha genel arama yaparken, ilerleyen iterasyonlarda arama işleminin daha ayrıntılı ve bölgesel yapması sağlanmaktadır. Bazı araştırmacılar atalet ağırlığını [0.4,0.9] arasında almışlardır. Bu parametre yerel ve global arama arasında bir denge unsuru olarak ortaya çıkmaktadır. Atalet ağırlığının değeri pozitif sabit, doğrusal veya doğrusal olmayan fonksiyon olabilmektedir.

$$W = W_{max} - \frac{(W_{max} - W_{min})}{iter_{max}} \times iter \quad (2.16)$$

Denklem (2.16)' da verilen W_{max} ve W_{min} sırasıyla atalet ağırlığının başlangıç ve bitiş değerlerini, iter ifadesi şuan sürünün bulunduğu iterasyon sayısını, iter max ise algoritmanın çalıştırılacağı iterasyon sayısını temsil etmektedir. Şekil 2.10 sürüdeki 2 parçacık için sonraki iterasyonda hızlanmalarının iki boyutlu örneğidir.



Şekil 2.10. Parçacık hızlarının iki boyutlu örneği [101]

Daraltma Parametresi: PSO’ da her bir iterasyonda parçacıkların daha az güncelleştirmeyle yakınsamayı sağlamak için daraltma katsayısı yaklaşımı Clerc ve Kennedy [106] tarafından sunulmuştur. Bu yaklaşım ile algoritmada, maksimum arama hızına gerek kalmamakla birlikte arama yakınsaması matematiksel olarak sağlanmaktadır. Parçacık en iyi pozisyona yakın bir konumda olduğunda parçacığın titreşim genliği azalmakta ve istenen bölgede arama yapması sağlanmaktadır. Zhou ve diğ. [107] standart PSO’ya göre bu yaklaşımın daha iyi sonuçlar verdiğini göstermişlerdir. Clerc ve Kennedy [106] tarafından sunulan yaklaşım Denklem (2.17) ve (2.18)’ de verildiği gibidir.

$$V_i^{k+1} = W \times [V_i^k + C_1 \times \text{Rand}_1^k \times (Pbest_i^k - x_i^k) + C_2 \times \text{Rand}_2^k \times (Gbest^k - x_i^k)] \quad (2.17)$$

$$K = \frac{2}{2 - \phi - \sqrt{\phi^2 - 4\phi}} \quad (2.18)$$

$$\phi = c_1 + c_2, \phi > 4 \quad (2.19)$$

Hızlandırma katsayıları: PSO’da temel hedef, algoritmanın ilk adımlarından itibaren parçacıkların yerel optimumlara takılmadan tüm arama uzayında çözüm aramasını sağlamaktır. Algoritma belirli bir iterasyona ulaştıktan sonra parçacıkların global en iyiye yönelerek etkin sonuçlara ulaşması hedeflenir. Algoritmada hızlandırma kriteri

olarak ifade edilen c_1 ve c_2 parametreleri yerel ve globalda etkin sonuçlara ulaşmayı sağlayacak en önemli etkenlerdendir. Bu nedenle her iki parametrenin etkin kontrolü algoritmada alınacak sonuçlar bakımından önem arz etmektedir.

Ratnaweera ve diğ. [105] parçacıkların çözüm uzayında etkin hareket etmelerini sağlamak için bir çalışma yapmışlardır. Algoritmada her iterasyon ilerledikçe parçacığın tecrübesi olan c_1 değerini azaltarak, sürünün tecrübesi olan c_2 değerini arttırmışlardır. Algoritmada parçacıkların yerel en iyilere takılmaması, çözüm uzayında daha etkin bir alanı tarayabilmeleri için başlangıçta c_1 parametresi yüksek, c_2 parametresi düşük değer alınmıştır. İterasyon sonuna doğru c_1 parametresi düşürülerek ve c_2 parametresi ise artırılarak, global en iyi değere ulaşması amaçlanmıştır. Bu çalışmanın matematiksel ifadesi Denklem (2.20) ve (2.21)' de görüldüğü gibidir. Burada c_{1i} ve c_{1f} başlangıç, c_{2i} ve c_{2f} bitiş sabitleri olarak ifade edilmektedir.

$$c_1 = (c_{1f} - c_{1i}) \times \frac{\text{iter}}{\text{iter}_{\max}} + c_{1i} \quad (2.20)$$

$$c_2 = (c_{2f} - c_{2i}) \times \frac{\text{iter}}{\text{iter}_{\max}} + c_{2i} \quad (2.21)$$

Yakınsama garantili PSO: PSO'da her bir parçacık, kendi ve sürü içerisindeki en iyi konum vektörlerinden türetilen bir yöne doğru hareket etmektedir. Algoritmada parçacığın en iyi pozisyonu sürünün en iyi pozisyonuna doğru hareket eder. Sürü içindeki en iyi pozisyon vektörünün çözüm uzayında en iyi nokta olduğu garantisi olmadığından dolayı algoritmanın yerel en iyilerde takılıp, en iyi çözümlere ulaşamamasına, kısır döngü içerisine girmesine neden olabilir.

Standart PSO'da sürü içerisindeki G_{best} 'e sahip parçacığın yeni iterasyonda hızı güncellenirken sadece atalet ağırlık parametresi ve hızına bağlı olarak güncelleme işlemi gerçekleşir. Bu durumda global en iyi hız vektörüne sahip parçacık, olduğu noktada kalır ve sürü içerisindeki diğer parçacıklar bu noktaya doğru yakınsar. Eğer bu yakınsama durumu çözüm uzayındaki en iyi noktaya varmadan başlarsa, sürü erken yakınsamış olur. Bu durumda algoritmanın elde edeceği sonuçlar iyi çıkmayacaktır. Bu durumu önlemek için Vandanbergh ve Engelbrecht [108], yakınsama garantili PSO olarak adlandırdığı, global en iyi parçacık için yeni bir hız

güncelleme denklemi önermiştir. Her iterasyonda sadece G_{best} değeri, Denklem (2.22)' de verilen formülasyona göre güncellenir.

$$V_i^{k+1} = \omega x V_i^k + (Pbest_i^k - x_i^k) + k^k x r_j \quad (2.22)$$

Burada r_j , $[-1,1]$ arasında rastgele sayılardan oluşan vektördür. k^k ölçekleme faktörüdür. İlk değer olarak $k^0 = 1$ verilir. İterasyon ilerledikçe k ölçekleme faktörü Denklem (2.23)' te verildiği gibi hesaplanır.

$$k^{(k+1)} = \begin{cases} 2k^k & \text{başarı_sayısı} > s_c \\ 0,5k^k & \text{başarısızlık_sayısı} > f_c \\ k^k & \text{diğer} \end{cases} \quad (2.23)$$

s_c ve f_c ayarlanabilen eşik parametreleridir. Global en iyi değere sahip parçacık, önceki iterasyona göre daha iyi bir değer bulursa başarı_sayısı değeri arttırılırken başarısızlık_sayısı değeri sıfırlanır. Eğer global en iyi değere sahip olan parçacık, önceki iterasyona göre daha kötü bir değer bulursa, başarısızlık_sayısı değeri arttırılırken başarı_sayısı değeri sıfırlanır. Bu güncelleme sayesinde global en iyi değere sahip parçacık, sadece azalan hız vektörüne bağımlı kalmayıp en iyi parçacığın çevresine yönlendirilerek, en iyiyi bulana kadar devam ettirilir. Ancak bu yöntem, arama uzayında tek en iyi değer bulunduğ problemlerde iyi çalışmasına rağmen, birden fazla en iyi değer bulunduğ problemlerde yerel en iyiye takılmasından dolayı kötü sonuç verebilmektedir.

Uygunluk – Uzaklık Oranlı PSO: Peram ve diğ. [109] hız güncelleme denklemine, her parçacığın komşuluğunda daha iyi bir kişisel en iyi pozisyona sahip diğ bir parçacığa doğru hareket etmesini sağlayacak şekilde Uygunluk Uzaklık Oranlı PSO önerisi geliştirmişlerdir. Standart PSO'da parçacıklar, yerel ve global en iyi vektörlere doğru konumlarını güncellerler. Herhangi bir parçacık, mevcut iterasyonda iyi bir değer elde etse de bir sonraki iterasyonda elde ettiği en iyi değeri kaybetmesi mümkün olabilmektedir. Peram ve diğ.'nin [109] çalışmalarında, parçacığın her bir hız vektörü güncellenirken, kendisine yakın pozisyonadaki daha iyi uygunluk değerine sahip diğ bir vektörün tecrübesinden yararlanması sağlanmaktadır. Bu durumda hız güncelleme formülü Denklem (2.24)'de verildiği gibi olur.

$$V_i^{k+1} = W \times V_i^k + c_1 \times \text{Rand}_1^k \times (P_{\text{best}_i}^k - x_i^k) + c_2 \times \text{Rand}_2^k \times (G_{\text{best}}^k - x_i^k) + c_3 \times \text{Rand}_3^k \times (P_{\text{best}_n}^k - x_i^k) \quad (2.24)$$

Yukarıda verilen Denklem (2.24)' de $P_{\text{best}_n}^k$ parçacığa amaç fonksiyon değeri – uzaklık oranını en yüksek değerle sağlayan n. parçacığın yerel tecrübesidir. Parçacıklar arasındaki oran Denklem (2.25)' de verildiği gibi hesaplanır.

$$\underbrace{\text{MAX}}_{n \text{ parçacık}} \left\{ \frac{\text{Fitness}(P_{\text{best}_n}) - \text{Fitness}(p_i)}{|P_{\text{best}_n} - p_i|} \right\} \quad (2.25)$$

Tam bilgilendirilmiş PSO: Standart PSO'da parçacıklar yeni konum vektörlerini, P_{best} ve G_{best} 'e göre günceller. Mendes ve diğ. [110] parçacığın hızını güncellerken sadece P_{best} ve G_{best} pozisyonlarından yararlanmak yerine o parçacığın tüm komşularının tecrübelerinden yararlanmayı önermişlerdir. Hızı yenilenecek parçacık için, komşularının amaç fonksiyon değeri ve komşuluk genişliğine göre ağırlık katsayı değeri hesaplanır. Yazarların geliştirdiği bu yaklaşım Clerc ve Kennedy [106] tarafından geliştirilen yaklaşımın genelleştirilmiş halidir. Clerc ve Kennedy [106] tarafından sunulan yaklaşımda c katsayısı P_{best} ve G_{best} vektörlerine eşit şekilde paylaştırılırken Mendes ve diğ. [110] yaklaşımlarında ise komşusu olan tüm parçacıklara eşit bölünerek dağıtılmaktadır. Bu durumda Denklem (2.6)' daki hız formülü Denklem (2.26)' daki şekle gelir. Burada $W(s)$, s. parçacığın ağırlık katsayısını, P_{best_s} , s. parçacığın yerel tecrübesini, N ise s. parçacığın komşu parçacıklar kümesini gösterir.

$$V_i^{k+1} = \chi \left[V_{ij}^k + \sum_{s \in N} c_s W_{(s)} U_s(0,1) \times (P_{\text{best}_{s,j}}^k - p_{i,j}^k) \right] \quad (2.26)$$

Birleştirilmiş PSO: Standart PSO'da yerel P_{best} ve global G_{best} en iyi değerler her iterasyonda formülasyonda ayrı şekilde hesaba katılarak, parçacıkların yeni hız vektörleri bulunur. Parsopoulos ve Vrahatis [111] çalışmalarında, hız vektörünün güncelleme işleminde P_{best} ve G_{best} konum vektörleri birleştirilir. Denklem (2.27)' de verilen G vektörü, parçacığın G_{best} vektörüne göre yeni hız değerini, Denklem (2.28)' de verilen L vektörü ise, parçacığın P_{best} vektörüne göre yeni hız değerini gösterir. $G_{\text{best}_{p_i}}$, p_i komşuluğundaki en iyi vektördür. Pozisyon güncellenirken (0,1) arasındaki rastgele sayının çarpımından elde edilen sonuç Denklem (2.29),

parçacığın bulunduğu iterasyondaki pozisyona eklenerek yeni pozisyon değeri üretilir. (Denklem (2.30)).

$$G_i^{k+1} = \chi [V_{ij}^k + c_1 \times \text{Rand}_1 \times (P_{best_{i,j}} - X_i^j) + c_2 \times \text{Rand}_2 \times (G_{best_j} - X_i^j)] \quad (2.27)$$

$$L_i^{k+1} = \chi [V_{ij}^k + c_1 \times \text{Rand}_1 \times (P_{best_{i,j}} - X_i^j) + c_2 \times \text{Rand}_2 \times (G_{best_{p_{i,j}}} - X_i^j)] \quad (2.28)$$

$$U_i^{k+1} = u G_i^{k+1} + (1-u) L_i^{k+1} \quad (2.29)$$

$$P_i^{k+1} = p_i^k + U_i^{k+1} \quad (2.30)$$

Komşuluk Topolojileri: PSO’da komşuluk çalışmaları algoritmanın literatüre girdiği günden beri başlamıştır. Literatürde test edilen ilk komşuluk topolojisi, değişken sayıda rastsal yer değiştirilmiş bağlantılı orijinal halka topolojisidir. Ayrıca bütün parçacıkların bir merkeze bağlandığı ancak birbirlerine direkt olarak bağlanmadığı bir tekerlek topolojisi de düşünülmüştür. Bunların dışında, gelişigüzel bağlantılı topoloji ve yıldız topoloji de önerilmiştir. Yıldız topoloji, tamamen bağlantılı bir sürüyü temsil ettiğinde aslında bir global en iyi modeldir. Kennedy ve diğ. [103] çok sayıda yerel minimuma sahip bir problemin çözümünde yıldız topoloji gibi çok bağlantılı topolojilerin optimum sonucu bulmada zorluk yaşayacağını söylemişlerdir. Kennedy ve Mendes [112] parçacıkların dört komşu parçacıkla bir ızgara ağı ile bağlandığı Von Neuman topolojisi önermişlerdir. Yazarlar, kompleks problemlerin daha küçük komşuluk kullanılarak, basit problemlerin ise geniş komşuluk kullanılarak daha iyi sonuçlar alınabileceğini göstermişlerdir.

Kennedy ve diğ. [103] tarafından sunulan deneysel sonuçlar, topolojinin algoritmanın performansını önemli derecede etkilediğini ancak optimum topolojinin özel probleme bağlı olduğunu göstermiştir. Örneğin, tekerlek topolojisi çok lokal optimumlu bir fonksiyona uygulandığında en iyi sonucu vermiştir. Kennedy ve diğ. [103] bunun topoloji boyunca bilginin daha yavaş yayılmasından kaynaklanabileceğini ve böylece algoritmanın çoklu lokal minimum karşısında daha sağlam olacağını ifade etmiştir. Ancak tek modlu fonksiyonlarda, yıldız topoloji (global model), birbirine daha az bağlı topolojilere göre bilginin daha hızlı yayılmasından dolayı daha iyi sonuçlar vermiştir [101].

2.2.3. PSO literatür taraması

PSO ilk defa Kennedy ve Eberhart [97] tarafından sunulmuştur. Daha sonra Kennedy ve Eberhart [102] tarafından geliştirilen algoritma Shi ve Eberhart [98,112] ve Kennedy ve diğ. [103] tarafından daha da geliştirilmiştir. Yazarlar tarafından kuş sürülerinin yiyecek aramaları ve yiyeceği bulduktan sonra sürünün yiyeceğe yönelmesi modellenmiştir. Buradaki en önemli nokta yiyecek kaynağını bulan kuşun sürüdeki diğer kuşlara rehberlik etmesi ve sürü içindeki bilgi paylaşımıdır.

Kennedy ve Eberhart [97] çalışmalarında doğrusal olmayan fonksiyonların optimizasyonu ve network eğitimi uygulamalarının yapıldığı PSO algoritmasını sunmuşlardır. Global ve yerel en iyi değer kavramlarını, PSO ile yapay zeka ve GA arasındaki ilişkileri açıklamışlardır. Shi ve Eberhart [99] klasik PSO algoritmasına, yakınsama üzerinde büyük etkisi olan atalet ağırlığı parametresini eklemişlerdir. Yazarlar çalışmalarında atalet ağırlığı parametresinin alabileceği değerleri karşılaştırmalı bir şekilde vermişler ve problem çözümünde uygun parametre değerlerinin verilmesiyle daha etkin sonuçlar alındığını göstermişlerdir.

Şevkli ve Yenisey [114] ATÇ problemlerinin çözümü için bir PSO modeli, DKA algoritması ile birlikte geliştirmişlerdir. Yazarların oluşturduğu model, tamamlanma zamanı performans ölçütüne göre literatürde yer alan bazı zor test problemleri üzerindeki sonuçlar incelenmiş ve iyi sonuçlar veren diğer sezgisel yöntemlerin sonuçlarıyla karşılaştırılmıştır. Genel olarak önerilen modelin diğer yöntemlere göre daha iyi veya eşdeğer seviyede sonuçlar verdiğini göstermişlerdir.

Taşgetiren ve diğ. [115] permutasyon akış tipi çizelgeleme problemlerinin çözümü için PSO algoritmasına DKA metodunu melezleyerek yeni bir algoritma sunmuşlardır. Amaç fonksiyonu olarak, çevrim zamanı ve max gecikmelerin minimize edilmesi olarak düşünülmüştür. Yazarlar, algoritmalarını literatürdeki test problemleriyle denemişler ve etkin sonuçlar aldıklarını göstermişlerdir.

Çok amaçlı optimizasyon problemlerinin PSO ile çözülmesinde yapılan çalışmalar arasında Ray ve Liew [116], pareto baskınlık kavramını kullanan PSO algoritmasını GA ile melezleyerek çalışma yapmışlardır. Yazarlar, algoritmalarında pareto optimal sonuçların çeşitliliğini sağlamak için kalabalıklık yöntemini kullanmışlardır. Pulido

ve Coello Coello [117] bir parçacığın hareket yönünün belirlenmesi için pareto baskınlık kavramını kullanmışlardır ve karar uzayında daha iyi bir birey dağılımı elde edebilmek için kümeleme algoritmalarıyla sürüleri alt topluluklara bölmüşlerdir. Srinivasan ve Seow [118], PSO algoritmasını EA ile melezleyerek (PS-EA) çok amaçlı bir yöntem geliştirmişlerdir. Yazarlar algoritmalarında, elit bireyleri saklamışlar ve pareto yönteminde birleştirme işlemini uygulamamışlardır. Raquel ve Naval [119] ise çok amaçlı PSO algoritmalarında küresel en iyinin seçilmesi ve pareto arşivdeki bireylerin silinmesi için kalabalıklık mesafesi kavramını kullanmışlardır.

Çok amaçlı EATÇ problemlerinin çözümünde Moslehi ve Mahnam [120], PSO ve YA algoritmalarını melezleyerek yeni bir algoritma sunmuşlardır. Zhang ve diğ. [121] aynı tür problem için PSO ile TA algoritmalarını melezlemişlerdir. Yine EATÇ problemlerinin çözümünde Jia ve diğ. [122] PSO algoritmasını kullanmışlardır. Xia ve Wu [28] PSO algoritmasına TB algoritmasını adapte ederek yeni bir algoritma geliştirmişlerdir. Buraya kadar anılan EATÇ problemlerinin çözümünde araştırmacılar, C_{max} , W_t ve W_m olmak üzere üç amacı eş zamanlı optimize etmeye çalışmışlardır. Bu çalışmaların dışında EATÇ problemlerinin çözümünde diğer amaç fonksiyonları bazında Grobler ve diğ. [123], Liu ve diğ. [2,88], PSO algoritması ve Rahimi-Vahed ve diğ. [90], PSO algoritmasına TA algoritmasını adapte ederek yeni yöntemler sunmuşlardır.

3. ÇOK AMAÇLI OPTİMİZASYON

Günümüz gerçek yaşam problemlerinde birden fazla amaca sahip optimizasyon problemleriyle karşılaşılır. Çok amaçlı optimizasyon problemlerini çözmek tek amaçlı optimizasyon problemlerine göre daha zordur. Çok amaçlı optimizasyon problemlerinde özellikle birbiriyle çelişen amaçlar olması durumunda problemin zorluk derecesi daha da artmaktadır. Amaçlardan biri maksimize edilmeye çalışırken diğer bir amacın minimize edilmeye çalışılması problemin karmaşıklığını daha da arttırmaktadır.

Performans kriterinde birden fazla amacın sistematik ve eş zamanlı olarak optimize edilmesi çok amaçlı optimizasyon olarak adlandırılır. Çok amaçlı optimizasyon problemlerinde, karar probleminin modeli kurulurken amaç fonksiyonunun oluşturulması zor olabilir. Karar problemlerinin bir çoğunda çözümün kalitesini değerlendirmek için birden fazla kriter söz konusudur. Bu kriterleri tek bir amaç fonksiyonunda toplamak her zaman olanaklı olmayabilir. Birden fazla kriterin söz konusu olduğu, özellikle bu kriterlerin birbirleriyle çeliştiği problemlerde farklı çözüm alternatifleri söz konusu olur.

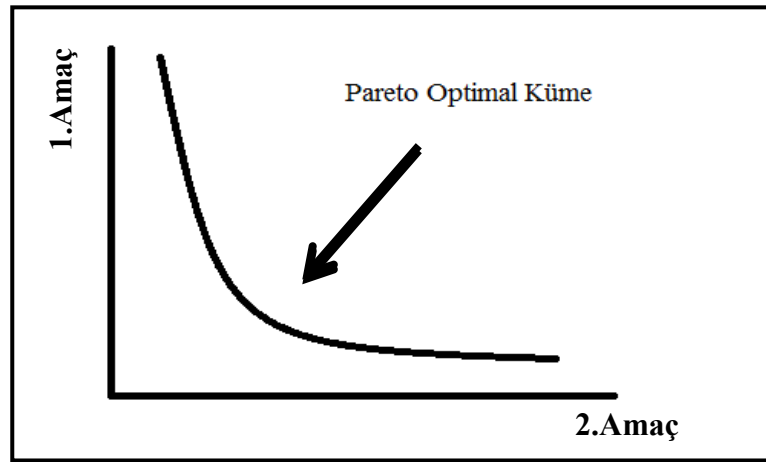
Çok amaçlı optimizasyon problemlerinin çözümünde kullanılan klasik yöntemlerden bir tanesi, performans kriterinde amaçların birleştirilmesidir. Farklı amaçlar, bir değerde birleştirilerek çok amaçlı problem, tek amaçlı problem haline getirilerek çözüm yapılır. Tek performans altında toplamak için en çok kullanılan yöntem ise, her bir amacın ağırlıklandırılarak tek amaç fonksiyonu haline getirilmesidir.

Çok amaçlı problemlerin çözümünde kullanılan ikinci yöntem olarak, amaçların her seferinde değiştirilerek çözümler elde edilmesidir. Her çözümde amaç fonksiyonu değiştirilerek, kalan diğer amaçların kısıt haline getirilmesi ile tek amaçlı problem gibi çözülmesidir. Amaçlar arasındaki tercih sırasının doğru belirlenmesi, çözüm sürecini etkileyeceğinden dolayı önemlidir.

Çok amaçlı optimizasyon problemlerinde kullanılan üçüncü yöntem olarak pareto optimal yöntemidir. Burada, tüm amaçları içeren bir vektör ve çözümler arasında tercih yapmayı sağlayan baskınlık kavramı ortaya çıkmaktadır.

3.1.Çok Amaçlı OptimizasyonYöntemleri

Çok amaçlı optimizasyon problemleri birden fazla amacın eş zamanlı en iyilenmesidir. Çok amaçlı optimizasyonda amaç fonksiyonları genellikle istenen bir sonucun farklı bir özelliğini dikkate alır. Çoğu zaman bu amaçlar, tüm fonksiyonları eş zamanlı olarak optimize eden tek bir sonucun bulunmadığı bir çelişki içerisindedirler. Bundan dolayı bir optimal çözümler kümesi söz konusu olur. Bu küme pareto optimalite kavramında, pareto optimal küme olarak adlandırılır. İki amaçlı optimizasyon problemine ait pareto optimal kümesi Şekil 3.1’ de gösterilmiştir.



Şekil 3.1. Pareto optimal küme

Çok amaçlı optimizasyon problemlerinde çözüm bir vektördür. Bundan dolayı çok amaçlı optimizasyon problemi aşağıdaki gibi bir vektör minimizasyonu olarak da ifade edilebilir:

$$\text{Min: } \{z_1(x), z_2(x), \dots, z_N(x)\} \quad (3.1)$$

Kısıtlar:

$$f_i(x) = z_i(x) \quad i=1,2,\dots,N \quad (3.2)$$

$$g_j(x) \leq 0 \quad j=1,2,\dots,J \quad (3.3)$$

$$h_k(x)=0 \quad k=1,2,\dots,K \quad (3.4)$$

Denklem (3.1) çözüm alanı, Denklem (3.2) $F(x)$ amaç fonksiyonu vektörü, x karar değişkenleri, Denklem (3.3) $g_j(x)$ eşitsizlik ve Denklem (3.4) $h_k(x)$ eşitlik kısıtlarını ifade etmektedir. Tek amaçlı optimizasyon problemlerinde tek amaç fonksiyonu değeri vardır. Çok amaçlı optimizasyon problemlerinde ise, tek amaçlı optimizasyondan farklı olarak; tek amaç fonksiyonu yerine amaç fonksiyonu vektörünün olmasıdır. Bu sebepten dolayı çok amaçlı optimizasyon problemleri bazen vektör optimizasyonu olarak da adlandırılmaktadır. Ayrıca amaç fonksiyonu ve kısıtların tiplerine göre, çok amaçlı doğrusal programlama modeli, çok amaçlı doğrusal olmayan programlama modeli, çok amaçlı tam sayılı programlama modeli şeklinde sınıflandırmalarda yapılabilmektedir. En iyi sonuç vektörünün belirlenmesi, vektörlerin sıralanarak seçilmesini kapsayan bir küme teorisi problemidir. Optimal çok amaçlı çözümler, pareto optimal olarak adlandırılan bu kümeler olarak tanımlanır.

Çok amaçlı optimizasyon problemlerinde tek amaçlı optimizasyon problemlerinde olduğu gibi tek global bir çözüm olmadığından, amaç fonksiyonlarına uygun olarak, bir çözümler kümesinin belirlenmesi gerekir. Bu durumda optimal bir noktanın belirlenmesi için, pareto optimalde baskın ve basılgın kavramlarının ortaya konulması gerekir.

Baskınlık: Çok amaçlı fonksiyonun bir vektörü $F(x^*) \in Z$, eğer; $F(x) \leq F(x^*)$ iken en az bir amaç için $F_i(x) < F_i(x^*)$ şartını sağlayan bir diğer $F(x) \in Z$ vektörü yok ise $F(x^*)$ vektörü baskındır.

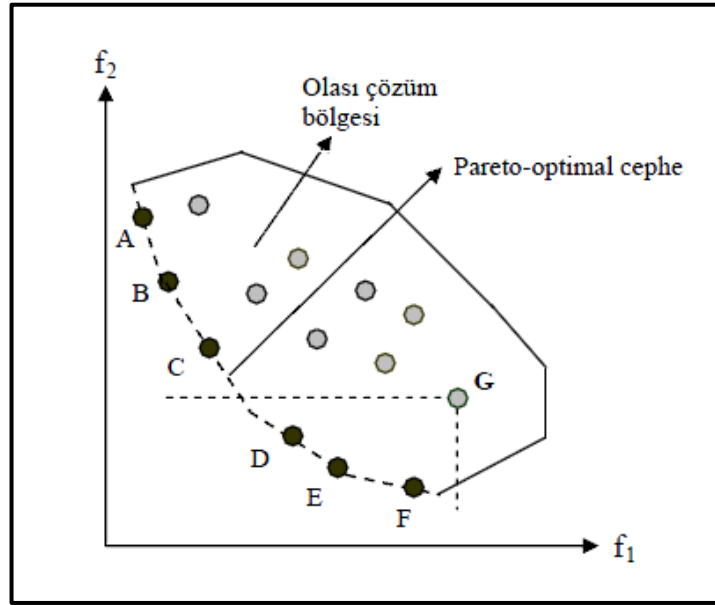
Basılgınlık: Çok amaçlı fonksiyonun bir vektörü $F(x^*) \in Z$, eğer; $F(x) \leq F(x^*)$ iken en az bir amaç için $F_i(x) < F_i(x^*)$ şartını sağlayan bir diğer $F(x) \in Z$ vektörü var ise $F(x^*)$ vektörü basılgındır.

Eğer Z çözüm alanında diğer noktalar bir çözüm noktasına baskın değilse bu çözüm noktası basılgın çözüm olarak adlandırılır. Diğer çözümler tarafından baskılanmamış olan tüm çözümler pareto optimal çözümlerdir. Dolayısıyla, bir x^* çözümü için eğer amaç fonksiyonlarının herhangi birisi diğer amaç fonksiyonunu bozmadan iyileştirilemiyorsa x^* pareto optimaldir.

Örneğin; U ve V iki amaçlı minimizasyon probleminin iki farklı çözüm vektörü olsun:

- Eğer $V_i < U_i$, $i=1,2,\dots,j$ ise V vektörü U vektörüne baskındır.
- Eğer $V_i \leq U_i$, $i=1,2,\dots,j$ ve en az bir i için $V_i < U_i$, ise V vektörü U vektörüne baskındır.
- Eğer $V_i > U_i$, $i=1,2,\dots,j$ ise V vektörü U vektörüne göre basılgındır.
- Eğer V_i vektörü U_i vektörüne baskın değil ve/veya U_i vektörü V_i vektörüne baskın değilse U_i ve V_i vektörleri karşılaştırılmaz.

Şekil 3.2 bireyler arasındaki baskınlık ilişkisini göstermektedir. İki amaçlı minimizasyon probleminde; A, B, C, D, E , ve F noktaları pareto optimal bireyleri veya baskın çözümleri, bu bireylerden oluşan $P^*=\{A,B,C,D,E,F\}$ pareto optimal kümeyi ve pareto optimal cepheyi göstermektedir. G noktası basılgın çözümdür.



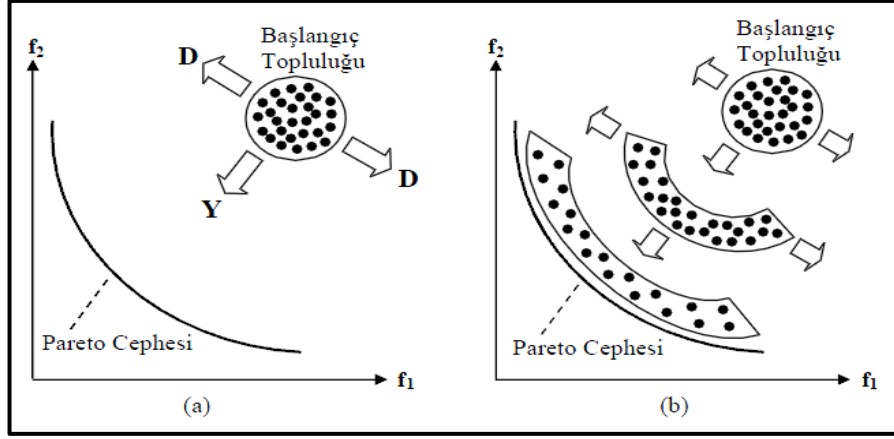
Şekil 3.2. Pareto optimal cephe [125]

Pareto optimal çözümlerde baskın ve basılgın çözümler arasındaki fark, çözüm prosedürü için önem arz etmektedir. Bir çözüm, diğer bir çözümden bütün kriterler açısından daha kötü ise bu çözüm kuvvetle basılgındır. Eğer bir çözüm bazı kriterler açısından diğer çözümden kötü, ancak en az bir kriter için diğer çözüm kadar iyi ise bu çözüm basılgındır. Eğer bir çözüm diğer bir çözümden bütün kriterler açısından daha iyi ise bu çözüm kuvvetle baskın bir çözümdür. Çok amaçlı optimizasyon

problemlerinde kuvvetle baskın çözümleri elde etmek, baskın çözümleri elde etmekten daha zordur. Bazı kombinatoriyal problemlerinin çözüm uzayında kuvvetle baskın çözümlere ulaşmak çok zordur. Bu tür durumlarda, kuvvetle baskın çözümler yerine, baskın çözümlere ulaşmak daha kolay olmaktadır. Özellikle birbirleriyle çelişen çok amaçlı optimizasyon problemlerinde bu tür durumlara daha çok rastlanmaktadır.

Pareto optimal cephe, çok amaçlı çözüm uzayında bulunan, diğer çözümler tarafından baskılanmamış yani baskın çözümleri kapsar. Pareto optimizasyon, pareto optimal cephenin ve buna en yakın komşuların bulunmasını amaçlar. Çok amaçlı optimizasyon problemlerinde pareto optimizasyonun kullanılmasının sebebi; problemlerin, sadece bir tek global optimuma sahip olmaması ve amaçlarla ilgili tercihlere arama prosesinden önce karar verilememesidir. Pareto optimizasyonda amaçlara ilişkin tercihler değişse bile, karar mekanizmasının elinde birden fazla alternatif olması seçimi kolaylaştırmaktadır. Çok amaçlı optimizasyonda göz önüne alınması gereken bir diğer durum ise bir problemde birbirleri ile çelişen amaçların yer almasıdır. Ancak çelişen amaçlara rağmen, pareto'ya dayalı modern sezgisel yöntemler, bütün kriterler için ideal çözümler bulabilmektedirler [89].

Tek amaçlı optimizasyon problemlerinde bir tek en iyi sonuç vardır ve çözümler en iyiden en kötüye doğru sıralanabilir. Çok amaçlı optimizasyon problemlerinde ise, bir tek en iyi çözüm yerine pareto optimal çözümler olarak adlandırılan, birden fazla en iyi çözüm vardır. Çok amaçlı optimizasyon problemleri çözüm kümesinde bir amaca göre iyi olan çözüm, diğer amaca göre kötü olabilir. Çok amaçlı optimizasyon problemlerinin asıl amacı, pareto cephesini bulmak veya yaklaşmak ve bu cephe üzerinde düzgün bir dağılım sağlayarak, karar vericiye alternatif karar seçenekleri sunmaktır. Bu amaçlar, Şekil 3.3'de çok amaçlı minimizasyon problemi için gösterilmektedir. Şekil 3.3a'da D düzgün dağılım veya çeşitliliği, Y yakınsamayı temsil etmek üzere, çözüm kümesinin düzgün dağılımının arttırılabilmesi için başlangıç popülasyonu D yönünde hareket ettirilmelidir. Pareto cephesine yakınsama için başlangıç popülasyonunun Y yönünde hareket ettirilmesi gerekir. Problemde düzgün dağılım ve yakınsama arasındaki denge doğru olarak ayarlandığında Şekil 3.3b'de görüldüğü gibi iyi bir çözüm kümesi elde edilir.



Şekil 3.3. (a) Başlangıç popülasyonu, (b) Çözüm kümesi [125]

Çok amaçlı optimizasyon problemlerinde birden fazla çözüm söz konusudur. Bu tür problemlerin çözümünde tek amaçlı problemlerin çözümlerinde kullanılan algoritmaları kullanmak bazen çözüm uzayının yeteri kadar taranamaması, iyi sonuçlar alınamaması gibi sonuçlara yol açabilmektedir. Araştırmacılar, çok amaçlı problemlerin çözümlerinde etkin sonuçlara ulaşabilmek için çözüm uzayının tamamını kapsayacak şekilde yöntemler geliştirmişler ve bunları çözüm algoritmalarına adapte etmişlerdir. Örneğin, çok amaçlı optimizasyon probleminde amaçlar tek bir amaçta birleştirilir. Bu tür yöntemlere, sabit ağırlıklı amaç işlevi ve değişken ağırlıklı amaç işlevi örnek olarak verilebilir. Bu tür yöntemler pareto tabanlı olmayan yöntemlerdir. Aşağıda çok amaçlı optimizasyon yöntemlerinden en çok kullanılanları hakkında bilgi verilecektir.

3.1.1. Sabit ağırlıklı amaç fonksiyonu

Sabit ağırlıklı amaç fonksiyonu, birden fazla amacı, tek amaç altında birleştirir ve tek amaçlı problem haline getirir. Amaçlara verilen farklı ağırlık değerleri, farklı pareto çözümlerin elde edilmesini sağlar. Birden fazla pareto optimal çözümün bulunabilmesi için, çözüm algoritmasının birden fazla çalıştırılması gerekir. Sabit ağırlıklı amaç fonksiyonu yöntemi, Denklem (3.5)' de görüldüğü gibi tek amaç fonksiyonu altında birleştirilir. Denklem (3.6)' da verilen $w_1, w_2, \dots, w_n$, n adet amaç için ağırlıklardır ve bu ağırlıkların toplamı 1 olmak zorundadır.

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_n f_n(x) \quad (3.5)$$

$$\sum_{i=1}^n w_i = 1 \quad (3.6)$$

Sabit ağırlıklar (w_i) ayarlanarak, çözüm algoritmasının, çözüm uzayını arama yönü ayarlanabilir. Sabit ağırlıklı amaç fonksiyonu yönteminde bir veya birden fazla amacın ağırlık değeri yüksek verilerek, arama uzayında belirli bir yönde hareket etmesi sağlanabilir. Burada karar verici en önemli gördüğü amacın ağırlık değerini daha yüksek tutarak, elde etmek istediği sonuçlara ulaşacak şekilde arama uzayının taranmasını sağlayabilir.

3.1.2. Değişken ağırlıklı amaç fonksiyonu

Değişken ağırlıklı amaç fonksiyonu yönteminde, rastgele seçilmiş ağırlık değerleri kullanılır. Sabit ağırlıklı amaç fonksiyonu yöntemi arama uzayında tek bir yönde arama yaptığından pareto çözüm kümesini bulmak için yeterli değildir. Çünkü çok amaçlı optimizasyon problemlerinde, çözüm uzayında farklı arama yönlerine ihtiyaç vardır. Bu eksikliği gidermek, çok amaçlı problemlerde farklı arama yönlerini gerçekleştirebilmek için değişken ağırlıklı amaç fonksiyonu yöntemi geliştirilmiştir.

Değişken ağırlıklı amaç fonksiyonu yönteminde yukarıda verilen Denklem (3.5) ve (3.6)'ya ek olarak aşağıda verilen Denklem (3.7) şartı da sağlanmalıdır. Bu yöntemde ağırlıklar sabit olmadığından çözüm uzayında tek yönde arama yerine farklı yönlerde arama yapar.

$$w_i = \frac{rasgele_i}{rasgele_1 + rasgele_2 + \dots + rasgele_n}, i = 1, 2, \dots, n \quad (3.7)$$

3.1.3. Vektör hesaplamalı genetik algoritma

Vektör hesaplamalı GA, bir sonraki popülasyonun alt kümesinin bir tek amaç tarafından seçilmesi ve gelecek popülasyonun oluşturulması için tüm alt kümelerin birlikte kullanılmasıdır. Bu şekilde gelecek popülasyon her amacın en iyi bireylerinden oluşmaktadır. Yöntemde n adet amaç varsa, algoritmanın bulunduğu iterasyondan n adet alt topluluk seçilir. Her alt topluluk, bir amaç fonksiyonunun en iyi değerlerinden oluşur ve her bir alt topluluk bir sonraki iterasyonun yalnızca bir bölümünü oluşturur. Yeni iterasyon, bir önceki iterasyonda oluşturulan alt kümelerin birleşimidir.

Vektör hesaplamalı GA, Denklem (3.8) ve (3.11) arasında verildiği gibi, amaç fonksiyonlarının doğrusal birleşimi gibi çalışır. Mevcut iterasyondan alt topluluklar oluşturulurken verilen denklemlerdeki eşitlikler kullanılır. $F(x)$ fonksiyonu, her alt topluluk seçimi için tek bir amaç tarafından belirlenir.

$$F(x) = 1f_1(x) + 0f_2(x) + \dots + 0f_n(x) \quad (3.8)$$

$$F(x) = 0f_1(x) + 1f_2(x) + \dots + 0f_n(x) \quad (3.9)$$

$$F(x) = 0f_1(x) + 0f_2(x) + \dots + 1f_{n-1}(x) + 0f_n(x) \quad (3.10)$$

$$F(x) = 0f_1(x) + 0f_2(x) + \dots + 0f_{n-1}(x) + 1f_n(x) \quad (3.11)$$

Vektör hesaplamalı GA pareto tabanlı bir yaklaşım olmayıp, pareto çözüm üreten bir yöntemdir [125]. Belirli bir amaç fonksiyonu tarafından seçilen her bir alt topluluk, daha sonra birleştirilir ve çözüm için kullanılan optimizasyon yönteminin adımlarına uygulanır.

3.1.4. Kuvvet pareto evrimsel algoritması

Zitzler [3] tarafından sunulan pareto kuvvet evrimsel algoritması (SPEA), farklı bir kümede (arşiv veya ikincil topluluk) saklanan baskın çözümlerin sayısını, çözüm kümesinin karakteristiğini bozmadan kümeleme yöntemini kullanarak azaltan bir yöntemdir. SPEA algoritmasının özellikleri;

- Mevcut iterasyona kadar elde edilmiş tüm çözümler arasından baskın çözümleri depolar.
- Depolanan bireylerin sayısını azaltmak için kümeleme tekniğini kullanır.
- Sayısal uygunluk değerlerini atamak için pareto üstünlük kavramını kullanır.

SPEA dört açıdan diğer yöntemlere göre farklılık gösterir [3]. Bu farklılıklar;

- Yukarıda bahsedilen, baskın çözüm deposunu, kümeleme tekniğini ve pareto üstünlük kavramını tek algortmada birleştirir.
- Popülasyondaki bir bireyin üstünlüğünü, harici depoda kümelenmiş çözümlere göre belirler.
- Harici kümedeki tüm çözümleri seçime katar.

- Popülasyondaki çeşitliliği korumak için yeni bir pareto temelli hücreleme metodu önerir.

SPEA Algoritmasında pareto optimal çözümlerin bulunması öncelikle pareto bireylerin puanlanması ile başlar. Pareto bireyin baskın olduğu bireylerin sayısı n , pareto olmayan bireylerin toplam sayısı N olarak ifade edilsin. Pareto bireyin puanı, baskın olduğu bireylerin sayısı ile orantılıdır. Denklem (3.12) bir pareto bireyin puanının hesaplanmasını vermektedir. Pareto bireylerin puanları aynı zamanda güçlerine de eşittir. Bu durumda, iki pareto birey aynı sayıda bireye baskın olması durumunda puanları eşit olur. Pareto olmayan bir bireyin puanı, Denklem (3.13)' de görüldüğü gibi, o bireye baskın olan tüm pareto bireylerin toplamına 1 eklenmesiyle hesaplanır.

$$s_i = \frac{n}{N+1} \quad (3.12)$$

$$f_i = 1 + \sum s_i \quad (3.13)$$

SPEA yönteminde her bir adımda bulunan baskın (pareto) bireyler, arşiv olarak adlandırılan ikinci bir toplulukta saklanır. Arşiv boyutu sabittir. Her adımda yeni bulunan baskın bireyler, arşivdeki bireyler ile karşılaştırılır ve baskın olan bireyler arşivde saklanmaya devam eder. İlerleyen iterasyonlarda baskın bireylerin sayısında artış olması durumunda, arşiv boyutu kadar birey arşivde saklanır. Eğer arşiv boyutu aşılsa, kümeleme yöntemi kullanılarak, çeşitliliği sağlamak üzere, kalabalık bölgede bulunan baskın çözümlerden eleme yapılır.

Kümeleme işleminde ilk olarak, birbirine yakın olan bireyler kümelenir. Her kümeden o kümeyi temsil eden bir birey seçilir ve geriye kalan bireyler kümeden silinir. SPEA'da kümeleme çözüm uzayında yapılmaktadır ancak parametre uzayında da yapılabilir.

Bazı problemlerde Pareto-Optimal küme aşırı derece geniş ve hatta sonsuz sayıda çözümden oluşabilir. Bununla beraber bu durum karar verici açısından kabul edilebilir sınırı aştığında, bulunan bastırılmamış çözümlerin çoğu yararsız hale gelir. Ayrıca harici kümenin boyutu SPEA'nın çalışmasını da etkiler. Çok fazla harici birey seçim baskısını düşürür ve aramayı yavaşlatır [3].

3.1.5. Kuvvet pareto evrimsel algoritması 2

Kuvvet pareto evrimsel algoritması 2 (SPEA2), Zitzler ve diğ. [126] tarafından, SPEA yöntemine göre daha iyi bir puanlama mekanizması, yoğunluk tahmin tekniği ve iyileştirilmiş bir arşiv yönetimi sağlamak için geliştirilmiştir.

SPEA yöntemi puanlama algoritmasında aynı puana sahip bireyler arasında rastsal seçim söz konusuydu. SPEA2 yönteminde arşiv üyeleri tarafından bastırılan bireylerin puanlarının aynı olmasını önlemek için hem baskı altına alındığı bireyler hem de baskın olduğu bireyler dikkate alınmaktadır. Her bir birey için hem arşivdeki ($\bar{P}_t$) hem de ana topluluktaki (P_t) her i bireyine, onu bastıran bireylerin sayısını gösteren bir güç değeri $S(i)$ Denklem (3.14)'deki gibi atanır.

$$S(i) = |\{j \mid j \in P_t + \bar{P}_t \wedge i \succ j\}| \quad (3.14)$$

Burada $+$ sembolü kümelerin birleşimini, $\succ$ sembolü ise pareto baskınlık ilişkisini göstermektedir. Bir i bireyinin ham puanı $R(i)$, S değerlerine bağlı olarak Denklem (3.15)'deki gibi hesaplanır.

$$R(i) = \sum_{j \in P_t + \bar{P}_t \wedge i \succ j} S(j) \quad (3.15)$$

$R(i)$ Değeri, bireyi hem arşivde hem de ana toplulukta bastıran bireyler kullanılarak belirlenir. Bu değerin küçük olması, bireyin pareto birey olduğunu gösterir. $R(i)$ değerinin büyük olması ise, o bireyin diğer bireyler tarafından bastırıldığını gösterir. Aynı puan değerlerine sahip bireylerin ayırt edilebilmesi için en yakın komşuluk yönteminden uyarlanan k yoğunluk tekniği uyarlanmıştır. Yazarlar genelde k . en yakın komşuya olan mesafenin tersini yoğunluk bilgisi olarak kullanmışlardır [125]. N , topluluk boyutu ve $\bar{N}$ ise arşiv boyutunu, σ_i^k k . elemana olan mesafeyi ifade etsin. Bu durumda, k yoğunluk bilgisi Denklem (3.16) ve i bireyinin yoğunluğu $D(i)$ Denklem (3.17)'de verildiği gibi hesaplanır.

$$k = \sqrt{N + \bar{N}} \quad (3.16)$$

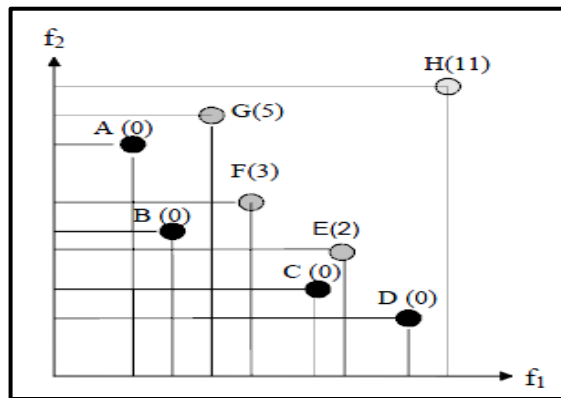
$$D(i) = \frac{1}{\sigma_i^{k+2}} \quad (3.17)$$

Denklem (3.17)'de yoğunluk değeri $D(i) < 1$ olması için paydaya 2 eklenmektedir. Bir i bireyinin son puanı $F(i)$, Denklem (3.18)'de verildiği gibi, ham puanı ($R(i)$) ile yoğunluk bilgisinin ($D(i)$) toplanmasıyla elde edilir.

$$F(i) = R(i) + D(i) \quad (3.18)$$

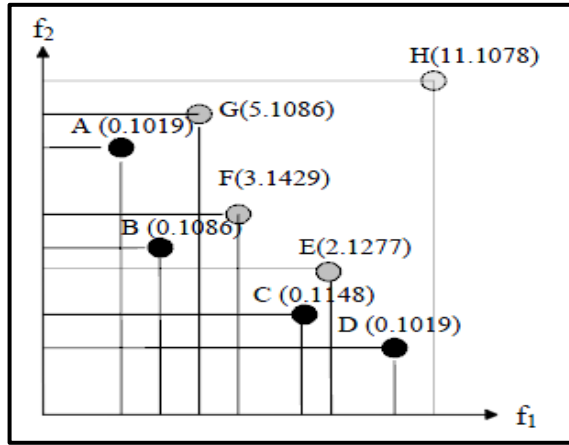
Şekil 3.4 örnek bir topluluğun SPEA2 yöntemine göre puanlanmasını göstermektedir. İlk olarak her bireye ait güç değerleri hesaplanır. Bir bireyin gücü, bu bireyin baskın olduğu bireylerin toplam sayısı ile belirlenir. A bireyi, G ve H bireylerine baskın olduğu için, A bireyinin gücü $S(A)=2$ olarak alınır. B Bireyi, F,G ve H bireylerine baskın olduğu için, B bireyinin gücü $S(B)=3$ olur. Aynı şekilde H bireyi, hiçbir bireye baskın olmadığı için $S(H)=0$ alınır. Bu durumda her bireye ait güç değerleri: $S(A)=2$, $S(B)=3$, $S(C)=2$, $S(D)=1$, $S(E)=1$, $S(F)=1$, $S(G)=1$, $S(H)=0$ şeklinde olur.

Her bireye ait güç değerleri belirlendikten sonra her bireye ait ham puanlar hesaplanır. Bir bireyin ham puanı, ona baskın olan bireylerin güçleri toplamıdır. A bireyinin ham puanı, hiçbir birey tarafından bastırılmadığı için $R(A)=0$ olur. Aynı şekilde B, C ve D bireyleri de hiçbir birey tarafından bastırılmadığı için ham puanları 0 olur. E bireyi, C bireyi tarafından bastırıldığı için C bireyinin $S(C)=2$ değeri, E bireyinin ham puanı $R(E)=2$ olarak hesaplanır. H bireyi diğer tüm bireyler tarafından bastırıldığı için $R(H)=S(A)+S(B)+S(C)+S(D)+S(E)+S(F)+S(G)=2+3+2+1+1+1+1+0=11$ değerini alır. Bu durumda tüm bireylere ait ham puanlar; $R(A)=0$, $R(B)=0$, $R(C)=0$, $R(D)=0$, $R(E)=2$, $R(F)=3$, $R(G)=5$, $R(H)=11$ şeklinde hesaplanır.



Şekil 3.4. SPEA2 Ham puanları [125]

Son aşamada, her bireye ait son puanlar, yoğunluk bilgisi değerleri ile ham puanların toplamından elde edilir. Şekil 3.4’de her bireye ait ham puanlar gösterilmiştir. Burada A,B,C ve D bireyleri aynı puana sahiptir ve bireylerin en çok istenilenden en az istenilene doğru sıralaması: A,B,C,D $\succ$ E $\succ$ F $\succ$ G $\succ$ H şeklinde olur. Şekil 3.5 ise SPEA2 tarafından atanan son puanları göstermektedir. Arşiv boyutu =2 ve k=3,1623 olarak hesaplanan son puan durumlarına göre, bireylerin en çok istenilenden en az istenilene doğru sıralaması ise; A, D $\succ$ B $\succ$ C $\succ$ E $\succ$ F $\succ$ G $\succ$ H şeklinde olur.



Şekil 3.5. SPEA2 Son puanları [125]

SPEA’ da kullanılan sınıflandırma tekniği, bastırılmamış kümenin karakteristiğini bozmadan küçültebilmesine rağmen, atılan bireyleri kaybedebilir. Bununla beraber, iyi bir bastırılmamış çözümler dağılımı elde etmek için bu çözümlerin arşivde tutulması daha iyidir. Bu durum SPEA2’de gerçekleştirilmektedir [126].

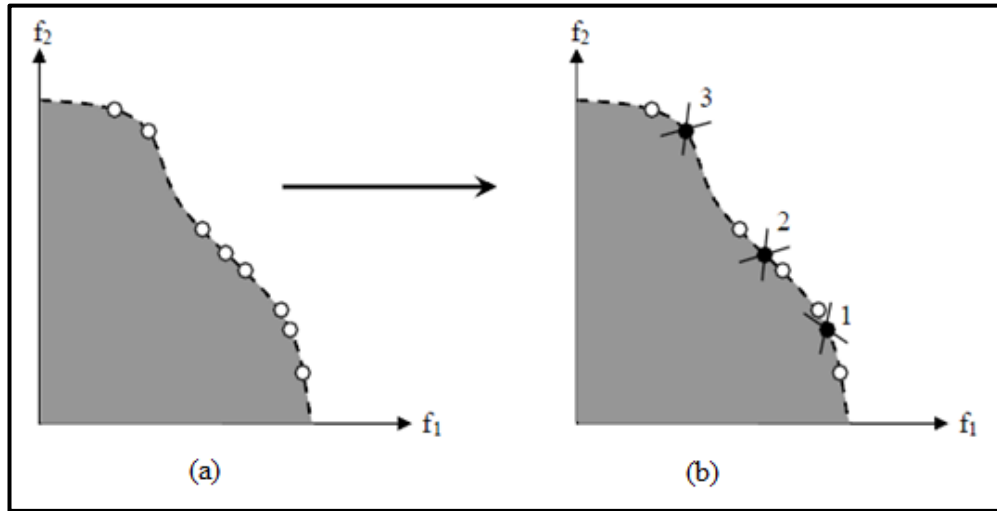
Arşiv yönetiminde ilk olarak, arşivdeki veya ana topluluktaki tüm pareto bireyler, bir sonraki iterasyonun arşivine kopyalanırlar. P ana topluluk, $\bar{P}$ arşiv, N topluluk boyutu ve $\bar{N}$ arşiv boyutu olmak üzere bir sonraki iterasyonda arşive kopyalanan pareto birey sayısı Denklem (3.19)’ da verildiği gibi gerçekleştirilir.

$$\bar{P}_{t+1} = \{i \mid i \in P_t + \bar{P}_t \wedge \text{Puan}(i) < 1\} \quad (3.19)$$

Arşive kopyalanan pareto birey sayısı, daha önceden belirlenen arşiv boyutuna ulaşırsa ($|\bar{P}_{t+1}| = \bar{N}$) seçme mekanizmasına geçilebilir. Arşive kopyalanan pareto birey sayısı arşiv boyutuna ulaşamazsa ($|\bar{P}_{t+1}| < \bar{N}$), bir önceki iterasyondaki pareto olmayan bireylerden en iyi ($\bar{N} - |\bar{P}_{t+1}|$) adedi gelecek arşive kopyalanır. Bu kopyalama

işlemi için, arşiv ve ana topluluk birleştirilir ($\bar{P}_t + P_t$) ve bireyler puan değerlerine göre sıralanır. Sıralanmış listeye göre, puanı 1'den büyük veya eşit olan bireyler yeni arşive ($\bar{P}_{t+1}$) kopyalanır.

Eğer pareto bireylerin sayısı arşiv boyutundan büyük olursa, bu durumda arşivdeki bireylerden bazıları arşiv boyutuna ulaşıncaya kadar silinirler. Silme işleminde bireylerin birbirlerine olan mesafeleri göz önünde bulundurulur. Bir bireye en yakın olan birey silinir. Bir bireye en az mesafeye sahip birden fazla birey olduğu durumda ise, en yakın 2.birey silinir. Şekil 3.6 arşiv güncellemesinin grafiksel gösterimidir. Arşiv boyutu ($\bar{N} = 5$) olarak kabul edilirse, Şekil 3.6a birbirine yakın bireylerin silinmesi, Şekil 3.6b ise güncelleme sonucu elde edilen yeni pareto bireyleri göstermektedir.



Şekil 3.6. (a). Arşivdeki yakın bireylerin silinmesi (b) Güncelleme [125]

3.2. Çok Amaçlı EATÇ Problemleri Literatür Taraması

Bu bölümde çok amaçlı EATÇ problemleri üzerine literatürde yapılmış çalışmalar detaylı bir şekilde incelenmiştir. Bölüm 1.3'te EATÇ problemlerinde tek amaç üzerine yapılmış çalışmalar incelenmişti. Bu bölümün başında öncelikle C_{max} , W_t ve W_m olmak üzere 3 amaçlı çalışmalar tarih sırasına göre sonrada diğer amaç fonksiyonları bazında yapılmış çalışmalar verilmiştir. Bölümün sonunda literatür çalışmalarından çıkarılan sonuçlar sunulmuştur.

Chiang ve Lin [127], EATÇ probleminin çözümü için etkin genetik operatörü ve populasyon çeşitliliği olan evrimsel bir algoritma geliştirmişler ve pareto sonuçlarını incelemişlerdir. Algoritmalarının ana özelliğinin, popülasyon büyüklüğü ve iterasyon olmak üzere sadece 2 parametreye ihtiyacı olduğunu belirtmişlerdir. Yazarlar algoritmalarını literatürde yer alan problem setleriyle ölçmüşler ve her problem seti için %70 oranında daha etkin sonuçlar aldıklarını göstermişlerdir.

Li ve diğ. [128], çok amaçlı EATÇ problemi için sıçrayan kurbağa algoritması geliştirmişlerdir. Başlangıç popülasyonuna kurallar ilave etmişlerdir. Ele aldıkları problemi çok amaçlı düşündüklerinden algoritmalarını da çok amaçlı hale getirmişler ve YA algoritmasını ilave etmişlerdir. Yazarlar algoritmalarını Kacem ve diğ. 'den [9,10] aldıkları 3 adet problem, Brandimarte'den [2] aldıkları 10 adet problem setleriyle değerlendirmişler ve literatürdeki mevcut çözümlere göre etkin sonuçlar aldıklarını söylemişlerdir.

Li ve Pan [86] bakım kısıtı altında çok amaçlı EATÇ problemlerinin çözümü için etkin ayırık kimyasal reaksiyon algoritması sunmuşlardır. Algoritmalarındaki her bir çözüm bir kimyasal molekül olarak temsil edilmiştir. Dört adet temel reaksiyon çaprazlama fonksiyonu tanımlamışlardır. Yazarlar algoritmalarını YA tabanlı TA yöntemiyle melezlemişlerdir. Algoritmadaki YA'yı daha da geliştirebilmek için çeşitli komşuluk yaklaşımı geliştirmişlerdir. Yazarlar algoritmalarını literatürde yer alan çalışmalar ile ölçmüşler ve algoritmalarının etkin sonuçlar verdiğini göstermişlerdir.

Wang ve diğ. [129] çok amaçlı EATÇ problemlerinin çözümü için pareto tabanlı yapay arı kolonileri algoritması geliştirmişlerdir. Başlangıç çözümlerinde çeşitliliği arttırmak için çeşitli stratejilerden oluşan bir kombinasyon adapte etmişlerdir. Daha sonra değişik komşuluk yapıları tanımlamışlardır. Algoritmalarına üçüncü olarak çözümleri geliştirebilmek için çaprazlama operatörü eklemişlerdir. Algoritmalarını literatürde yer alan kıyaslama problemleri ile ölçmüşler ve etkin sonuçlar verdiğini göstermişlerdir.

Davarzani ve diğ. [130] çok amaçlı YBS algoritması geliştirmişlerdir. Sundukları algoritma popülasyonun çeşitliliğini ve yakınsama hızını arttırmaktadır. Yazarlar,

literatürde yer alan problem setleriyle algoritmalarını test etmişler ve etkin sonuçlar almışlardır.

Xiong ve diğ. [131] melez çok amaçlı evrimsel algoritma (EA) geliştirmişlerdir. Algoritmaya, GA kromozomu ve operatörleri dâhil edilmiştir. Ayrıca EA'ya YA algoritmasını melezlemişlerdir. Pareto optimal sonuçlarının literatürde diğer yazarların algoritma sonuçlarına göre daha iyi sonuçlar verdiğini göstermişlerdir.

Moslehi ve Mahnam [120] YA algoritması ve PSO tabanlı melez bir yaklaşım sunmuşlardır. Çok amaçlı PSO algoritması, öncelik tabanlı EATÇ problemine uygulanmıştır. Yazarlar sundukları algoritmayı, Pareto optimal ve toplam ağırlıklı çok amaçlı diğer algoritmalar ile karşılaştırmışlardır. Kendi algoritmalarının diğer algoritmalara göre tatmin edici sonuçlar verdiğini göstermişlerdir.

Li ve diğ. [96] melez pareto tabanlı ayrık yapay arı kolonileri algoritması sunmuşlardır. Yazarlar melez algoritmalarında etkin bir çaprazlama operatörü geliştirmişlerdir. Şimdiye kadar bulunan etkin olmayan çözümleri hafızadan atmak ve çözüm aşamasında işlem süresini azaltmak için dışsal pareto geliştirmişlerdir. Algoritmanın arama kabiliyetini geliştirebilmek için çeşitli YA yaklaşımları sunmuşlardır. Yazarlar algoritmalarını literatürde yer alan kıyaslama problemleriyle test etmiş ve diğer algoritmalara göre daha etkin sonuçlar elde etmişlerdir.

EATÇ Problemi, gerçek yaşam çizelgeleme problemlerine daha yakın ve ATÇ problemlerinin uzantısıdır. Wang ve diğ. [73], EATÇ problemlerinin çözümü için bağışıklık ve entropi tabanlı çok amaçlı GA yaklaşımı sunmuşlardır. Özel kromozom yapısı için etkin çaprazlama ve mutasyon operatörleri geliştirmişlerdir. Yazarlar algoritmalarını mevcut diğer algoritmalar ile karşılaştırmış ve etkin sonuçlar elde etmişlerdir.

Bagheri ve diğ. [132] yapay bağışıklık tabanlı bütünleşik bir algoritma sunmuşlardır. Algoritmanın başlangıç popülasyonuna çeşitli stratejiler adapte etmişlerdir. Aynı zamanda yeni bireyler oluşturmak için farklı mutasyon operatörleri tanımlamışlardır. Yazarlar algoritmalarını literatürdeki problem setleriyle çözmüş ve algoritmalarının etkin sonuçlar verdiğini göstermişlerdir.

Unachak [72], çalışmasında EATÇ problemini gerçek yaşam üretim problemlerine uyarlamış ve bu problemin çözümünde GA kullanmıştır. Tam zamanında üretim felsefesinde işlerin makinelere atanması ve çizelgeleme kurallarını uyarlayarak, sunduğu GA yaklaşımıyla esnekliği başarabileceğini vurgulamıştır. Bu uyarlamalı gösterimleri hem statik hem de dinamik ortamda test edilmiştir. Statik EATÇ problemlerini literatürde yer alan kıyaslama problemleri ile çözmüş ve elde edilen sonuçları diğer algoritma sonuçlarıyla karşılaştırmıştır. Diğer yaklaşımların çoğu, EATÇ problemlerini tek amaçlı olarak optimize ederken, yazarın yaklaşımı çok amaçlı EATÇ problemi için sürekli olarak tatmin edici sonuçlar vermiştir. Makine arızalanması gibi kısıtların olduğu dinamik model için yazar, kendi algoritmasını literatürdeki, sağa kaydırmalı yeniden çizelgeleme ve ön çizelgeleme olmak üzere iki tane kıyaslama algoritmasıyla karşılaştırmıştır. Yazar, kayıp zamanı minimize etmek performans ölçütlü dinamik EATÇ problemleri çözümü için sunduğu algorithmada etkin sonuçlar elde etmiştir.

Li ve diğ. [87] etkin melez bir TA algoritması sunmuşlardır. İki uyarlamalı kural kombinasyonlu etkin komşuluk yapısına sahip TA algoritması tanımlanmıştır. Yazarlar algoritmalarını kıyaslama problem setleriyle çözmüşlerdir. Literatürde yer alan, Kacem ve diğ. [10], Xia ve Wu [28] ve Zhang ve diğ. [121] tarafından sunulan algoritmalar ile karşılaştırmışlar ve daha etkin sonuçlar almışlardır.

Rajkumar ve diğ. [133] bakım kısıtı altında çok amaçlı EATÇ problemlerinin çözümü için açgözlü rastsal uyarlamalı arama algoritması geliştirmişlerdir. Kendi yöntemlerini literatürde yer alan problem setleriyle çözmüş ve etkin sonuçlar almışlardır.

Üretim çizelgeleme problemlerinin çoğunluğu makinelerin sürekli olarak kullanılabilir olduğu varsayımı altında çalışılmıştır. Ancak gerçek yaşam problemlerinde makinelerin koruyucu bakım gibi faaliyetlerinden dolayı çalışmaya ara vermek zorunda oldukları durumlar vardır. Wang ve Yu [134] bakım kısıtı altında EATÇ probleminin çözümü için sezgisel bir yaklaşım sunmuşlardır. Yazarlar algoritmalarını literatürdeki problem setleriyle çözmüş ve algoritmalarının memnun edici sonuçlar verdiğini göstermişlerdir.

Zhang ve diğ. [121] çok amaçlı EATÇ problemlerinin çözümü için PSO ve TB yöntemlerinin kombinasyonu olan melez bir yaklaşım sunmuşlardır. Yazarlar yaklaşımlarının özellikle büyük boyutlu problem setleri için etkin sonuçlar verdiğini göstermişlerdir.

Xing ve diğ. [135] çok amaçlı EATÇ probleminin çözümü için bir simülasyon model önermişlerdir. Diğer bir çalışmalarında, Xing ve diğ. [136] aynı problemin çözümü için etkin bir YA metodu sunmuşlardır. Yazarlar, yaptıkları her iki çalışmalarını da literatürde yer alan kıyaslama problemleriyle test etmişler ve etkin sonuçlar almışlardır. Bu iki çalışmalarından bir yıl önce Xing ve diğ. [35], aynı problemin çözümü için çift katmanlı KKA sunmuşlardır. Problemin çözümü için iki farklı algoritma geliştirmişlerdir. Birinci algoritma, işlerin makinelere en uygun şekilde atanmasını sağlamaktadır. İkinci algoritmaları ise, her makineye atanan işlerin en uygun sıralamasını tespit etmektedir. Aynı amaç fonksiyonunun alındığı üç çalışmada da yazarlar sayısal sonuçlarla, kendi algoritmalarının literatürdeki diğer algoritmalara göre daha etkin sonuçlar verdiğini göstermişlerdir.

Gao ve diğ. [69] EATÇ problemlerinin çözümü için melez GA sunmuşlardır. Yazarlar, çözümleri temsil etmek üzere iki vektör kullanmışlardır. Gelişmiş çaprazlama ve mutasyon operatörleri, problemin karakteristikleri ve özel kromozom yapısına adapte etmek için kullanılmıştır. GA'da bireyler arasındaki arama yeteneğini geliştirmek amacıyla, iki adet YA prosedürü tanımlanmıştır. En erken ve en geç zamanlı operasyonların belirli zaman aralığında atamasını bulabilmek için etkin bir metot geliştirilmiştir. Algoritma literatürde yer alan kıyaslama problemlerine uygulanmış ve algoritmalarının etkin sonuçlar verdiği gözlenmiştir.

EATÇ Problemleri, gerçek yaşam çizelgeleme problemlerine en yakın özelliği ile klasik ATÇ probleminin uzantısıdır. Gao ve diğ. [70] EATÇ problemi için yenilikçi YA prosedürü içeren melez GA önermişlerdir. Yazarlar, önerdikleri algoritmayı, Kacem ve diğ. [10], Xia ve Wu [28], ve Zhang ve Gen [137] çözümleri ile karşılaştırmışlardır. Yazarlar karşılaştırdıkları yöntemlere göre daha iyi sonuçlar almışlardır.

Ho ve Tay [138] güdümlü YA ve EA yöntemlerinin birleştirilmesiyle oluşturdukları etkin bir yaklaşım sunmuşlardır. Komşu çözümleri bulmak için kullanılan rastgele

YA yerine, Pareto optimal çözümleri için yakınsama sürecini hızlandıran güdümlü YA algoritmalarını geliştirmişlerdir. Aynı zamanda, çok amaçlı çözümlerin alt sınırını bulmak için DSA önermişlerdir. Deneysel sonuçlar, algoritmalarının çok amaçlı pareto optimal çözümlerinin karşılaştırdıkları algoritmalara göre daha etkin sonuçlar verdiğini göstermiştir.

Shi-Jin ve diğ. [139] çok amaçlı EATÇ problemlerinin çözümü için sezgisel bir yöntem sunmuşlardır. Üç amaçlı bu problemin her bir amacı için bir ağırlık değeri tanımlamışlar, çözüm için DSA' dan esinlenerek sezgisel yöntem sunmuşlardır. Yazarlar, kendi algoritmalarını literatürde yer alan diğer çözümlerle karşılaştırmışlar ve kendi yöntemlerinin daha etkin sonuçlar verdiğini göstermişlerdir.

Jia ve diğ. [122] çok amaçlı EATÇ problemlerinin çözümü için, yerel optimum tıkanmalarından kaçma özelliğine sahip ve geliştirilmiş global arama kabiliyetini kullanabilen PSO yöntemini sunmuşlardır. Yazarlar yeni yaklaşımlarında performansı arttırmak için iki düzenleme yapmışlardır. Öncelikle, PSO parametreleri, etkili kullanabilme ve araştırma yapabilme için geliştirilmiştir. Yerel tıkanmalardan kaçınmak için YA algoritması adapte edilmiştir. Sunulan algoritma kıyaslama problem setleriyle test edilmiş ve etkin sonuçlar verdiğini gösterilmiştir.

PSO kuşların uçuş alışkanlıklarını ve bilgi alışverişlerini taklit eden evrimsel bir optimizasyon tekniğidir. Bu yöntem lokal ve yerel aramayı bir arada kullanır. TB, çizelgeleme ve sıralama problemlerinde çeşitli durumlar için etkili sonuçlar verdiğini ispatlanmış diğer bir yöntemdir. Xia ve Wu [28] çok amaçlı EATÇ problemi için, PSO ve TB metotlarının melez bir yaklaşımını sunmuşlardır. Yazarlar önerdikleri algoritmayı, Kacem ve diğ. [10] çözümleri ile karşılaştırmışlar ve algoritmalarının etkin sonuçlar veren uygulanabilir bir yaklaşım olduğunu göstermişlerdir.

Zhang ve Gen [137] toplam ve kısmi esnekliklere sahip EATÇ problemlerinin çözümü için GA sunmuşlardır. Yazarlar, büyük boyutlu problem setleri için sundukları GA'nın etkin sonuçlar verdiğini göstermişlerdir.

Özbakır [89] EATÇ problemine proses plan esnekliğini entegre ederek, çok objektifli EATÇ problemine modern sezgisel optimizasyon teknikleri ile çözüm yaklaşımı geliştirmiştir. EATÇ Problemlerinin modellenmesi için dil teorisi kapsamında yer

alan bağlam bağımsız gramer yapısı kullanmıştır. Bu yaklaşım ile problemin karmaşık yapısal özelliklerini, bağlam bağımsız gramer tanımlamasına dayalı basit bir gösterimle ifade etmiştir. Yazar çalışmasında proses plan esnekliği içeren ve gramer modeli oluşturulan EATÇ problemine, çok objektifli TA'ya dayalı optimizasyon tekniği ile çözüm algoritması geliştirmiştir. Önerilen algoritma, EATÇ problemi için; proses plan seçimi, operasyonların makinelerle atanması ve bu makinelerde sıralanması alt problemlerine eş zamanlı olarak çözüm geliştirmektedir. Pareto optimizasyona dayalı çok objektifli yaklaşım, çözüm sonucunda karar mekanizmasına, birden fazla uygun çözüm arasından seçim yapabilme imkanı sunmaktadır. Yazar ayrıca çalışmasında, EATÇ problemleri için ürün teslim tarihlerinin gerçekçi olarak belirlenmesine yönelik yeni bir yaklaşım ortaya koymuştur. Bu yaklaşımla, iş sisteminin özelliklerini, yapısı gereği öğrenilebilir ve yansıtılabilir niteliğine sahip olan genetik programlama yöntemi kullanılarak, çizelgeleme sistemine özel teslim tarihi belirleme denklemleri oluşturmuştur. Önerilen yaklaşım, YSA ve regresyon analizleri ile karşılaştırılmış ve etkinliği ortaya konulmuştur. Yazar çalışmasının sonunda, EATÇ problemleri için geliştirilen genel çözüm algoritması ile türetilen hipotetik problemler üzerinde deneysel çalışma ve analizler yapmıştır. Deneysel çalışmalar sonucunda, EATÇ performansı üzerinde etkili olan faktörler ve bu faktörlerin en iyi düzeylerini belirlemiştir. Sıralama kuralları ile makine esneklik düzeyleri arasındaki ilişkiyi, seçilen bir performans kriterine etkileri açısından irdeleyerek analiz etmiştir. Ayrıca literatürde yer alan çok objektifli EATÇ problemleri çözülerek önerilen yaklaşımın geçerliliğini ortaya koymuştur.

Kacem ve diğ. [9,10], kısmi ve toplam esneklik altında EATÇ problemlerinin çözümü için iki yeni yaklaşım sunmuşlardır. Birinci yaklaşımları, ideal bir atama modeli, ikinci yaklaşımları ise, evrimsel bir GA yaklaşımıdır. Diğer bir çalışmada Kacem ve diğ. [11], EATÇ problemlerinin çözümü için, evrimsel algoritmalar ve melez bulanık mantık tabanlı bir Pareto yaklaşımı sunmuşlardır. Bu melez yaklaşım, bulanık mantık bilgi gösterimi ve evrimsel algoritmaların uyarlama yeteneğini kullanmaktadır. Sonuçlar, yazarların önerdiği algoritmaların kısa işlem zamanı içerisinde tatmin edici sonuçlar verdiğini göstermiştir.

Buraya kadar incelenen çalışmaların tümü, C_{max} , W_t ve W_m olmak üzere 3 amaç fonksiyonu bazında yapılmış çalışmalardır. Bu çalışmalar en yeni çalışmadan başlanarak verilmiştir. Toplu olarak bu çalışmalar Tablo 3.1’ de görülmektedir. Tablo 3.1’ in birinci sütununda çalışmayı yapan yazarlar, ikinci sütunda problemin çözümü için kullandıkları algoritma, üçüncü sütunda çalışmalarında kullandıkları problem boyutları, dördüncü sütunda yazarların kendi algoritmalarını karşılaştırdıkları diğer yazarların çalışmaları, beşinci sütunda çalışmalarının yayınlanma tarihi ve son sütunda ise çalışmanın yayımlandığı dergi görülmektedir. Üçüncü sütunda yer alan problem boyutlarını gösterim şeklinde ilk değer iş sayısını, ikinci değer makine üçünce değer ise operasyon sayısını temsil etmektedir.

Tablo 3.1 incelendiğinde yazarlar çok amaçlı EATÇ problemlerinin çözümü için meta sezgisel yöntemler ile çözüm yaparken sadece bir algoritma kullanma yerine iki meta sezgisel yöntemi melezleyerek veya kullandıkları meta sezgisel yönteme sıralama kurallarını adapte ederek algoritmalarını geliştirdikleri görülmektedir. Melez algoritmalar özellikle çok amaçlı problemlerde üç amaç gerçekleştirilmeye çalışıldığında kullanıldığı görülmektedir. İki amaçlı problemi göz önünde bulunduran yazarlar sadece bir meta sezgisel yöntem kullanarak çözümler elde ettikleri de görülmektedir.

Çok amaçlı EATÇ problemlerinin çözümünde yazarlar kullandıkları meta sezgisel yönteme genelde YA algoritmasını adapte etmişlerdir. Buraya kadar üç amaçlı EATÇ problemlerinin çözümünde incelenen 28 çalışma içerisinde sadece bir yazarın dinamik durumları çalıştığı, bir yazarın hem dinamik hemde statik durumları diğer 26 yazarın ise sadece statik durumları çalıştığı gözlenmiştir. Tüm çalışmalar literatürdeki problem setleriyle karşılaştırılmış ancak herhangi bir gerçek yaşam problemi çözülmemiştir. Çok amaçlı problemlerin çözümünde yazarlar genelde amaçları ağırlıklandırarak tek amaç fonksiyonu gibi çözmekle beraber az sayıda pareto optimal çözümleri sunan çalışmalarda yapılmıştır.

Tablo 3.1. Çok amaçlı EATÇ problemlerinin çözümünde kullanılan yöntemler

Yazarlar	Kullanılan Algoritma	Kullanılan Problemler	Karşılaştırılan Çalışmalar	Yayın Yılı	Dergi
Chiang ve Lin [127]	Evrimsel Algoritma	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56 BRdata [48]	Ho ve Tay.[138] Xing ve diğ. [135] Li ve diğ. [87] Wang ve diğ.[73] Li ve diğ.[96] Bagheri ve diğ.[132]	2012	Int. Journal of Production Economics
Li ve diğ. [128]	Siçrayan Kurbağa Alg.	4x5x12 10x7x29 15x10x56 BRdata [48]	Kacem ve diğ. [10] Xia ve Wu [28] Ho ve Tay [138] Zhang ve diğ. [121] Xing ve diğ. [65,66] Wang ve diğ. [73] Bagheri ve diğ.[132]	2012	Applied Mathematics and Comp.
Li ve Pan [86]	Kimyasal Reaksiyon Alg. + Tabu Aramaları + Yerel arama	4x5x12 15x10x56 8x8x27 10x7x29 10x10x30	Xia ve Wu [28] Zhang ve diğ. [121] Xing ve diğ [136] Li ve diğ. [87] Moslehi ve Mahnam[120]	2012	Applied Soft Computing
Wang ve diğ. [129]	Arı kolonileri Alg. + Yerel arama	4x5x12 15x10x56 8x8x27 10x7x29 10x10x30	Kacem ve diğ. [10] Xia ve Wu [28] Zhang ve diğ.[121] Bagheri ve diğ.[132] Li ve diğ. [96]	2012	Int. Journal Advanced Manufacturing Technology
Davarzani ve diğ. [130]	Yapay bağıklık sistemi	8x8x27 10x10x30 15x10x56 BRdata [48]	Kacem ve diğ.[10] Xia ve Wu [28]	2012	Int. Journal of Hybrid Information Technology
Xiong ve diğ. [131]	Evrimsel Alg. + Yerel arama	4x5x12 15x10x56 8x8x27 10x7x29 10x10x30	Kacem ve diğ. [10] Xia ve Wu. [28] Gao ve diğ. [70] Ho ve Tay [138] Wang ve diğ. [73] Zhang ve diğ. [121] Bagheri ve diğ [132]	2012	Mathematical Problems in Eng.
Moslehi ve Mahnam [120]	Parçacık sürü Opt. + Yerel Arama Alg. (MOPSO+LS)	8x8x27 10x10x30 15x10x56 4x5x12 10x7x29 10x10x30 15x10x56	Kacem ve diğ.[10] Xia ve Wu [28] Gao ve diğ.[69] Zhang ve diğ. [121] Xing ve diğ. [135] Xing ve diğ. [136]	2011	International Journal of Production Economics
Li ve diğ. [96]	Melez Karınca Kolonileri Alg. (P-DABC)	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ.[10] Xia ve Wu [28] Zhang ve diğ. [121]	2011	International Journal Advanced Manufact Techn.
Wang ve diğ. [73]	Genetik Algoritma (MOGA)	4x5x12 8x8x27 10x10x30 15x10x56	Xia ve Wu[28] Zhang ve diğ[121] Xing ve diğ.[136]	2010	International Journal Adv Man. Techn.

Tablo 3.1. (Devam) Çok amaçlı EATÇ problemlerinin çözümünde kullanılan yöntemler

Yazarlar	Kullanılan Algoritma	Kullanılan Problemler	Karşılaştırılan Çalışmalar	Yayın Yılı	Dergi
Bagheri ve diğ. [132]	Yapay Bağışıklık Algoritması (AIA)	8x8x27 10x10x30 15x10x56 BRdata [48] Hurink ve diğ. [59]	Kacem ve diğ. [10] Xia ve Wu [28] Zhang ve Gen [137]	2010	Future Generation Computer Systems
Unachak [72]	Genetik Algoritma	8x8x27 10x10x30 15x10x56	Lower Bounds	2010	Ph. D. Thesis
Li ve diğ. [87]	Tabu Aramaları (HTSA)	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28] Zhang ve diğ. [121] Xing ve diğ. [136]	2010	Computers & Industrial Engineering
Rajkumar ve diğ. [133]	Açgözlü rastsal uyarlamalı arama algoritması (GRASP)	4x5x12 8x8x27 10x10x30 15x10x56	Gao ve diğ. [70]	2010	International Journal of Production Research
Wang ve Yu [134]	Süzülmüş Işın Arama Sezgiseli (FBS)	8x8x27 10x10x30	Gao ve diğ. [70]	2010	Computers & Industrial Eng.
Zhang ve diğ. [121]	Parçacık sürü Opt + Tabu Aramaları (PSO+TS)	4x5x12 8x8x27 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2009	Computers & Industrial Eng.
Xing ve diğ. [135]	Simulasyon Model	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2009	Applied Soft Computing
Xing ve diğ. [136]	Yerel arama Metodu	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2009	Journal Intelligent Manufacturing
Xing ve diğ. [35]	Çift Katmanlı Karınca Kolonileri	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2008	New Generation Computing
Gao ve diğ. [69]	Genetik Alg. + Değişken Komşuluk Arama (GA+VND)	8x8x27 10x10x30 15x10x56	Kacem ve diğ. [10] Zhang ve Gen [137] Xia ve Wu [28]	2008	Computers & Operations Research
Ho ve Tay. [138]	Evrimsel Alg. + Yerel Arama Alg. (MOEA-GLS)	4x5x12 8x8x27 10x7x29 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2008	IEEE Transactions On Systems, Man, And Cybernetics

Tablo 3.1. (Devam) Çok amaçlı EATÇ problemlerinin çözümünde kullanılan yöntemler

Yazarlar	Kullanılan Algoritma	Kullanılan Problemler	Karşılaştırılan Çalışmalar	Yayın Yılı	Dergi
Shi-Jin ve diğ. [139]	Süzülmüş Işın Arama Sezgiseli (HFBS)	4x5x12 10x10x30 15x10x56	Kacem ve diğ. [10] Xia ve Wu [28]	2008	Int. Journal of Production Research
Jia ve diğ. [122]	Parçacık Sürü Opt.	8x8x27 10x10x30	Kacem ve diğ. [10]	2007	Int. Conference on Grey Systems
Gao ve diğ. [70]	Genetik Alg. + Yerel Arama Alg.	8x8x27 10x10x30 15x10x56	Kacem ve diğ [10] Zhang ve Gen [137] Xia ve Wu[28]	2007	Computers & Industrial Engineering
Xia ve Wu [28]	Parçacık Sürü Opt +Tavlama Benzetimi (PSO+SA)	8x8x27 10x10x30 15x10x56	Kacem ve diğ.[10]	2005	Computers & Industrial Engineering
Zhang ve Gen [137]	Genetik Alg.	3x4x8 8x8x27 10x10x30	Heuristic (SPT) Klasik GA Kacem ve diğ.[10]	2005	Complexity International
Özbakır [89]	TA+ Genetik Prog.	4x5x12 8x8x27 10x10x30 15x10x56	Kacem ve diğ [10]	2004	Ph. D. Thesis
Kacem ve diğ. [10]	Genetik Algoritma + Yerini Belirleme Yaklaşımı (AL+CGA)	3x4 x8 8x8 x27 10x10x30	Lower Bounds	2002	IEEE transactions on Systems, Man, and, Cybernetics
Kacem ve diğ. [11]	Evrimsel Algoritma + Bulanık Mantık (FL+EA)	4x5x12 10x7x29 10x10x30 15x10x56	Lower Bounds	2002	Mathematics and Computers in Simulation

Bundan sonraki incelenecek çalışmalar yine çok amaçlı olmak üzere değişik amaç fonksiyonları kullanılarak yapılmış çalışmalardır. Bu çalışmalar farklı amaç fonksiyonları olduğundan sıralama, çalışma tarihi gözetilmeden amaç fonksiyonu bazında birbirine yakın çalışmalar arka arkaya gelecek şekilde verilmiştir. Bu çalışmalarda toplu olarak Tablo 3.2 'de görülmektedir.

Vilcot ve Billaut [140] çok kısıtlı EATÇ problemi üzerine çalışmışlardır. Amaç fonksiyonu olarak; C_{max} ve maksimum gecikmenin en küçüklenmesi olmak üzere iki amaçlı düşünülmüştür. Problemin çözümü için GA ve TA tabanlı iki çözüm yaklaşımı sunmuşlardır. Sunulan her iki yaklaşım literatürde kıyaslama problem setleriyle analiz edilmiş ve algoritmalarının etkin sonuçlar verdiği gösterilmiştir.

Tamaki ve diğ. [66] EATÇ problemlerinin karma tam sayılı doğrusal programlama yaklaşımlarıyla çözülebilmesi için bir model geliştirmişler ve aynı zamanda bu

modelin büyük boyutlu problemlerde çok fazla hesaplama zamanı gerektireceğinden GA geliştirmişlerdir. Yazarlar amaç fonksiyonu olarak $C_{\max}$ ve toplam gecikmenin minimize edilmesini ele almışlardır. Algoritmalarından kısa süre içerisinde optimale yakın iyi sonuçlar elde ettiklerini göstermişlerdir.

Baykasoğlu ve diğ. [88] çok amaçlı EATÇ problemlerinin çözümü için TA algoritması ve çözüm yaklaşımı sunmuşlardır. Problemi, önceden tanımlanmış performans ölçütünü optimize etmek için rotalama ve sıralama işlemlerini bulmak üzere ele almışlardır. Problem kapsamı, her kısım için alternatif işlem planları hesaba katılarak genişletilmiştir. Alternatif işlem planları ve makinelerin olasılıksal seçimi dikkate alınmıştır. Yazarlar, problemin çözümü için çok amaçlı TA ve dağıtım kural tabanlı sezgisel yöntemin temel prensiplerini sunmuşlardır. Amaç fonksiyonu olarak, $C_{\max}$, toplam gecikme ve yük dengesi olmak üzere üç amaçlı ele almışlardır. Yazarlar önerdikleri modelin EATÇ problemlerinin çözümünde daha hızlı ve etkin sonuç verdiğini göstermişlerdir.

Bütünleşik süreç planlama ve çizelgeleme için literatürde geliştirilmiş birçok yaklaşımlar vardır. Baykasoğlu ve Özbakır [141], bütünleşik etkin süreç planlama ve çizelgeleme için çok amaçlı TA algoritmasını ve süreç planlama temsil kuralını kullanan yeni bir yaklaşım sunmuşlardır. Yazarlar, amaç fonksiyonu olarak toplam akış zamanı ve toplam işleme maliyetini göz önünde bulundurmuşlardır. Yaklaşımlarını literatürde yer alan kıyaslama problemleriyle test etmişler ve alternatif süreç planlarına sahip EATÇ problemlerinin çözümünde etkin sonuçlar aldıklarını göstermişlerdir.

Fattahi [81] çok amaçlı EATÇ problemi çözümü için Pareto tabanlı melez bir yaklaşım sunmuştur. Amaç fonksiyonu olarak $C_{\max}$ ve toplam ağırlıklı gecikmeyi minimize etmek üzere çok amaçlı hedeflemiştir. Problemin çözümü için etkin bir TB algoritması geliştirmiştir. Algoritmanın gerçek endüstri problemleri ve büyük boyutlu problemler için kolay uygulanabilir olduğu ve optimale yakın sonuçlar bulduğu gösterilmiştir.

Fattahi ve Fallahi [142], dinamik EATÇ probleminin çözümü için GA'ya dört kural melezleyerek yeni bir yöntem önermişlerdir. Amaç fonksiyonu olarak, $C_{\max}$ ve toplam sapma cezası ile başlama zamanlarındaki sapmanın en küçüklenmesi olarak

ele almışlardır. Yazarlar küçük, orta ve büyük boyutlu problem setleriyle kendi algoritmalarını DSA çözümleriyle karşılaştırmışlardır. DSA ile küçük boyutlu problemleri çözebilmiş, orta boyutlu problemler için uzun zaman diliminde problemin alt ve üst sınırlarını bulabilmiş, büyük boyutlu problem setlerini ise çözememişlerdir. Yazarlar kendi algoritmaları ile orta ve büyük boyutlu problem setlerinde optimale yakın sonuçlar aldıklarını göstermişlerdir.

Gholami ve Zandieh [82] dinamik EATÇ problemleri için TB ve GA yaklaşımlarını birleştirmişlerdir. Çalışmalarında, literatürdeki çalışmaların çoğunda göz önünde bulundurulmayan, makinelerin sürekli olarak hazır olamaması (bozulma, bakım vb. nedenlerle) veya işlerin kesintiye uğraması durumları gibi gerçek hayatta karşılaşılan durumlar da rastsal olarak modele dâhil edilmiştir. Yazarlar amaç fonksiyonu olarak, C_{max} ve ortalama gecikme zamanının en küçüklenmesi olarak ele almışlardır. Sundukları algoritmalarını çeşitli büyüklükteki problem setleriyle test etmişler ve makinelerin ortalama tamir süresinin ve makinelerin ortalama arızalanma seviyesinin çizelgeleme verimi üzerinde önemli etkiye sahip olduklarını tespit etmişlerdir.

Grobler ve diğ. [123] bağımlı hazırlık zamanlı, yedek kaynaklı ve makine arızalı çok amaçlı EATÇ problemlerinin çözümü için PSO sunmuşlardır. Amaç fonksiyonu olarak; C_{max} , erken/geç teslim ve kuyrukta bekleme süresi olmak üzere üç amaçlı ele almışlardır. Önerilen algoritma kıyaslama problem setleriyle çözülmüş ve akademik çalışmalara katkısının yanı sıra maliyetleri düşürme, kârlılığı artırma ve rekabet üstünlüğü sağlamaya yardımcı olduğu gösterilmiştir.

Jian-jun ve diğ. [143] çok amaçlı EATÇ problemlerinin çözümü için bağımsızlık sistemine dayalı bir algoritma geliştirmiştir. Bu çalışmada her antikorun en mükemmel bölgeleri belirlenerek, diğer antikora aşılacaktır. Yazarlar performans ölçütü olarak; C_{max} , akış zamanı, makine kullanım oranı, erken ve geç ceza maliyeti, öncelik bağlantılı işlerin derecesi, hazırlık miktarı, makine iş yükü şeklinde 8 tane amaç değerini farklı ağırlıklandırarak göz önünde bulundurmuşlardır. Algoritmanın uygulaması Çin'de havacılık endüstrisinde faaliyet gösteren bir tesiste test edilmiştir. Sonuçta, mükemmel bölgelerin belirlenerek aşılması yönteminin, algoritmanın yakınsama hızını arttırdığı tespit edilmiştir.

Lin ve Jia-Zhen [71] EATÇ problemleri içinde yer alan çelik boru üretim tesisinde üretim çizelgeleme problemini çalışmışlardır. Paralel makineler, kapasite ve hız kısıtları altında çalışmakta, makine bakım faaliyetleri orta düzeydedir. İşlerin optimum sıralanması ve her iş için esnek rotaların belirlenmesi için karışık tam sayılı programlama modeli önermişlerdir. Amaç fonksiyonu olarak, C_{max} , üretim aksamalarını ve makinelerin boş bekleme sürelerini minimize etmek üzere çok amaçlı olarak ele alınmıştır. NP-zor sınıfında gösterilen problemin çözümü için bir GA sunulmuştur. Geliştirilen sistem işletmede bir yıl başarıyla uygulanmıştır. Sistem manuel olarak hazırlanan çizelgelemeye göre ortalama %10 daha etkin sonuçlar vermiştir. Ayrıca boş süre azalmış ve çizelgeleme sonucunda etkinlik artmıştır.

Liu ve diğ. [1] çok amaçlı EATÇ problemlerinin çözümü için çoklu PSO yaklaşımı sunmuşlardır. Yazarlar çalışmalarının başında çok amaçlı EATÇ probleminin ana faktörleri ve matematiksel modelini sunmuşlardır. Çalışmalarının devamında, tanımlanan problemin çözümü için geliştirilen çoklu PSO'nu tanımlamış ve algoritmayı analiz etmişlerdir. Performans ölçütü olarak, akış zamanı ve C_{max} olmak üzere iki amaçlı ele almışlardır. Yazarlar özellikle büyük boyutlu problem setleri için sundukları yaklaşımın etkin sonuçlar verdiğini göstermişlerdir.

Liu ve diğ. [124] DKA ve PSO algoritmalarının bileşimi olan, değişken komşuluk parçacık sürü optimizasyonu olarak adlandırdıkları, melez bir meta sezgisel algoritma geliştirmişlerdir. Performans ölçütü olarak, akış zamanı ve tamamlanma zamanı olmak üzere iki amaçlı ele alınmıştır. Yazarlar kendi algoritmalarını test etmek için, literatürde yer alan kıyaslama problem setleriyle çözüm yapmışlar ve sunulan algoritmanın diğer algoritmalara göre daha iyi sonuçlar verdiğini göstermişlerdir.

Frutos ve diğ. [83] çok amaçlı EATÇ probleminin çözümü için; TB tabanlı YA prosedürü ile GA'dan oluşan memetic algoritma sunmuşlardır. Amaç fonksiyonu olarak, C_{max} ve toplam işleme maliyeti ele alınmıştır. Yazarlar algoritmalarını önce C_{max} olmak üzere tek amaçlı daha sonra da iki amaçlı çözümler elde ederek diğer algoritmalarla karşılaştırmışlar ve etkin sonuçlar elde etmişlerdir.

Rabiee ve diğ. [75] C_{max} ve toplam operasyon maliyeti minimize amaçlı kısmi EATÇ problemi üzerine çalışmışlardır. Çözüm için üç çeşit GA ve bir tane de evrimsel

strateji geliřtirmişlerdir. Sundukları dört algoritma içinde pareto sonuçları almışlardır. İki amaç fonksiyonu üzerine odaklanan yeni bir çözüm kromozomu oluşturmuşlardır. Algoritmalarını kendi ürettikleri problem seti ve literatürde yer alan problem setleriyle ölçmüşlerdir.

Wu [144] erken ve geç tamamlanmalı işlerin çizelgelenmesi ve teslim tarihini eş zamanlı optimize etmek amaçlı çok temsilcili iş yükü kontrol yaklaşımı sunmuştur. Tam zamanında üretim yapan işletmelerin önemli bileşenleri arasında yer alan erken ve geç tamamlanma zamanlı işler hem sanayide hem de akademik ortamda büyük ilgi görmektedir. Yazarın sunduğu yaklaşım, işin çok temsilciliye giriři, rotdanması, sıralanması ve bilgi geri beslemeyi sağlamaktadır. Yazar, iki yeni teslim tarihli kural geliřtirmiştir. Teslim tarihini ayarlamak için bir geri besleme mekanizması tanıtmıştır. Her iki kuralda pratikte kolaylıkla uygulanabilir yöntemlerdir. Uygulama sonucunda yazarın önerdiği yöntem sayesinde, işlerin akış zamanında %20,3 ve iş sürecinde %33,1 oranında iyileřtirme sağlanmıştır.

Prakash ve diğ. [74] esnek üretim sistemlerinde kompleks çizelgeleme probleminin çözümü için bilgi tabanlı GA olarak adlandırdıkları yeni bir yaklaşım sunmuşlardır. Esnek üretim sistemlerindeki kombinatoriyel karar verme problemleri için meta sezgisel yöntemler ve bu yöntemlerden GA kullanıldığı görülmektedir. Yazarlar yeni yaklaşımlarında, bilgi tabanlı geliřtirdikleri GA ile esnek üretim sistemlerinde biten iş sayısı ve ortalama akış süresi olmak üzere iki amaçlı problem çalışmışlardır. Kendi sonuçlarını literatürde yer alan diğeri GA çözüm sonuçları ile karşılaştırmışlar ve büyük boyuttaki problemlerin çözümünde kendi yöntemlerinin daha iyi sonuçlar verdiğini göstermişlerdir.

Tanev ve diğ. [67] gerçek yaşam EATÇ problemi olarak, plastik enjeksiyon makineleri üreten bir fabrikada müşteri siparişlerinin çizelgelenebilmesi için bir yaklaşım sunmuşlardır. Yazarların algoritmaları, GA ve öncelik dağıtım kurallarının birleşiminden oluşturulmuş melez evrimsel bir yaklaşımdır. Algoritmanın performansı, fabrika veri bankasında depolanmış 400 müşteri sipariři kullanılarak ölçülmüştür.

Chan ve diğ. [68] kaynak kısıtlı EATÇ problemleri için GA tabanlı bir yaklaşım sunmuşlardır. Amaç fonksiyonu olarak; tamamlanma zamanı ve makine aylak süresinin minimize edilmesi amaçlanmıştır.

Karışık model montaj hattı dengeleme, talep çeşitliliğinin çok olduğu üretim yönetimi alanında önemli strateji problemlerinden biridir. Rahimi-Vahed ve diğ. [90] çalışmalarında, toplam iş faydası, toplam üretim hızı değişimi ve toplam hazırlık zamanı olmak üzere çok amaçlı problemi çözmek için, PSO ve TA tabanlı melez çok amaçlı algoritma geliştirmişlerdir. Algoritma, literatürde yer alan üç tane çok amaçlı GA sonuçları ile karşılaştırılmış ve önerilen yöntemin büyük boyutlu problemler için önemli derecede daha iyi sonuçlar verdiği gösterilmiştir.

Tablo 3.2’de çok amaçlı EATÇ problemlerinde Tablo 3.1’in dışında kalan çalışmalar özetlenmiştir. İlgili tabloların birinci sütununda çalışmayı yapan yazarlar, ikinci sütunda problemde ele aldıkları amaç fonksiyonları, üçüncü sütunda yazarların çözüm için geliştirdikleri algoritma, dördüncü sütunda kullanılan problem boyutları, beşinci sütunda çalışmanın yılı ve altıncı sütunda ise çalışmanın yayımlandığı dergi görülmektedir. Tablo 3.2’de yazarlar özellikle iki amaç fonksiyonunu göz önünde bulundurduklarında Tablo 3.1’in aksine tek yöntemle de çözümler elde ettikleri görülmektedir. Değişik amaç fonksiyonları bazında incelenen 19 çalışmadan 12 tanesi statik 7 tanesi dinamik durumlar olmak üzere literatürdeki problem setleriyle algoritmaların verimliliği ölçülmüştür. 19 Çalışmanın sadece 3 tanesi gerçek yaşam problemleri olarak işletmelere uygulanmıştır.

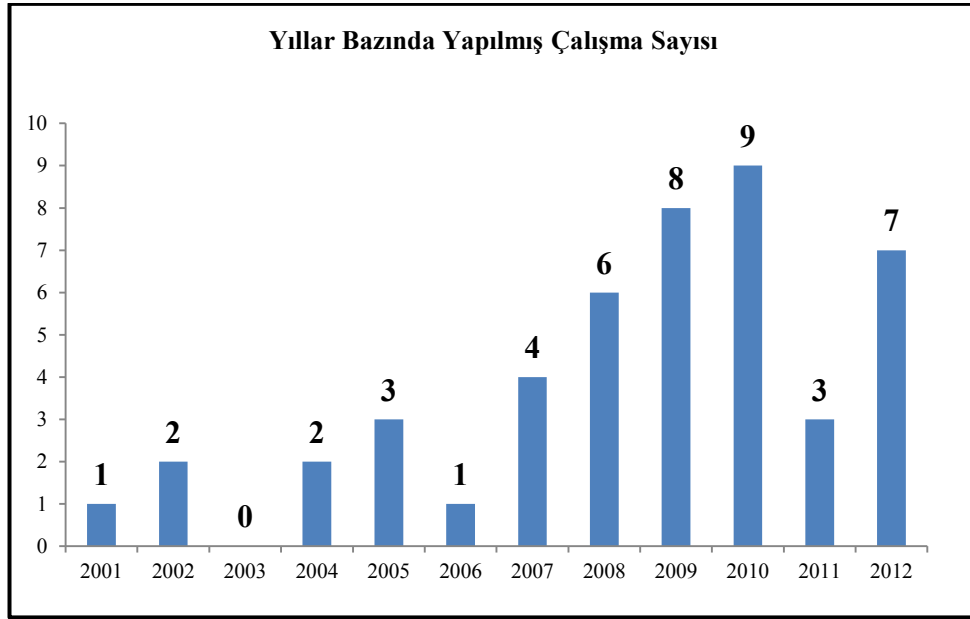
Çok amaçlı EATÇ problemleri literatür taraması başlığı altında incelenen toplam 47 çalışmanın 28 adedi C_{max} , W_t ve W_m amaçlı olmak üzere kalan 19 tanesi diğer amaç fonksiyonları bazında yapılmıştır. Çok amaçlı EATÇ problemlerinin meta sezgisel yöntemlerle çözümünde yazarların GA yöntemini daha çok tercih ettikleri gözlenmiştir. Çok amaçlı incelenen 47 adet çalışma içerisinde sadece 3 tanesi işletmelerde gerçek yaşam uygulamaları olarak yapılmıştır. Diğer çalışmalar literatürdeki problem setleriyle çözümü gerçekleştirilmiş teorik çalışmalardır.

Tablo 3.2. Çok amaçlı EATÇ problemlerinin çözümü için amaç fonksiyonu bazında yapılmış diğer çalışmalar

Yazarlar	Amaç Fonksiyonu	Kullanılan Algoritma	Kullanılan Problem Boyutları	Yayın Yılı	Dergi
Vilcot ve Billaut [140]	Makespan + Toplam Gecikme	(GA+TS)	5x10 5x15 5x20 10x10 10x15 10x20 10x30 15x15	2008	European Journal of Operational Research
Tamaki ve diğ. [66]	Makespan + Toplam Gecikme	GA	---	2001	IEEE International Conference
Baykasoğlu ve diğ. [88]	Makespan + Toplam Gecikme + Yük Dengesi	TA	---	2004	Journal of Intelligent Manufacturing
Baykasoğlu ve Özbakır [141]	Toplam Akış Zamanı + Toplam İşleme Maliyeti	TA		2009	Journal Intelligent Manufacturing
Fattahi [81]	Makespan + Toplam Ağırlıklı Gecikme	TB	---	2009	Proceedings Of World Academy Of Science
Gholami ve Zandieh [82]	Makespan + Ortalama Gecikme	TB+GA	BRdata [48]	2009	Journal of Intelligent Manufacturing
Grobler ve diğ. [123]	Makespan + E/G Teslim + Kuyruk zamanı	PSO	---	2010	Ann Oper ation Research
Jian-jun ve diğ. [143]	Makespan + Akış zamanı + Mak. kul. oranı + E/G Ceza + İş derecesi + Setup + Mak.iş yükü	YBS	---	2008	IEEE, Int. Conf. Man. Science & Engineering
Fattahi ve Fallahi [142]	Makespan + Toplam sapma ve başlama zamanlarındaki sapma	GA	---	2010	CIRP Journal of Manufacturing Science and Technology
Lin ve Jia-Zhen [71]	Makespan+Toplam Gecikme+toplam makine boş zamanı	GA+ Karışık Tamsayılı prog. Modeli	---	2009	Systems Engineering — Theory & Practice
Liu ve diğ. [1]	Makespan + Akış Zamanı	PSO	8x8 10x10 15x10	2009	Fundamenta Informaticae
Liu ve diğ. [124]	Makespan + Akış Zamanı	PSO+ Değişken Komşuluk Arama	8x8 10x10 15x10	2007	IEEE Transactions On Systems, Man, And Cbbernetics PART B.
Frutos ve diğ. [83]	Makespan + Topl İşleme Maliyeti	TB + GA	3x4 4x5 10x7 10x10 15x10	2010	Ann Operation Research
Rabiee ve diğ. [75]	Makespan +Toplam Operasyon Maliyeti	GA+ Evrimsel Alg.	BRdata [48]	2012	Int. Journal of Production Research

Tablo 3.2. (Devam) Çok amaçlı EATÇ problemlerinin çözümü için amaç fonksiyonu bazında yapılmış diğer çalışmalar

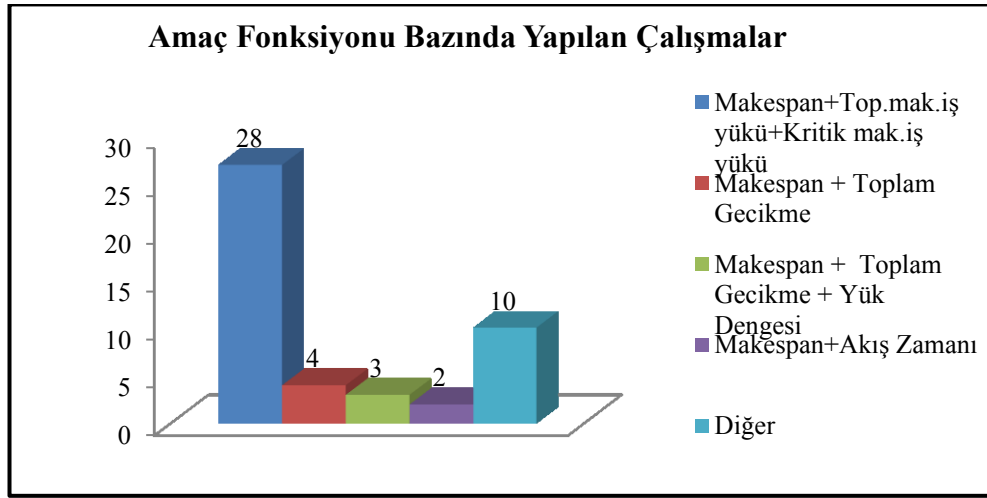
Yazarlar	Amaç Fonksiyonu	Kullanılan Algoritma	Kullanılma Problem Boyutları	Yayın Yılı	Dergi
Wu [144]	Akış Zam.+E /G Teslim	Çok Temsilcili İş yükü Kontrol Yaklaşımı	---	2005	Ph. D. Thesis
Prakash ve diğ. [74]	Ort.Akış Zamanı + Biten İş Sayısı	GA	--	2011	Expert Systems with Applications
Tanev ve diğ. [67]	Yüksek İş Hızı	GA+ Öncelik Kuralları	---	2004	Applied Soft Computing
Chan ve diğ. [68]	Makespan + Makine Aylak Zamanı	GA	---	2006	Int. Journal of Production Res.
Rahimi-Vahed ve diğ. [90]	toplam iş faydası + toplam üretim hızı + toplam hazırlık maliyeti	PSO+TA	--	2007	Engineering Optimization



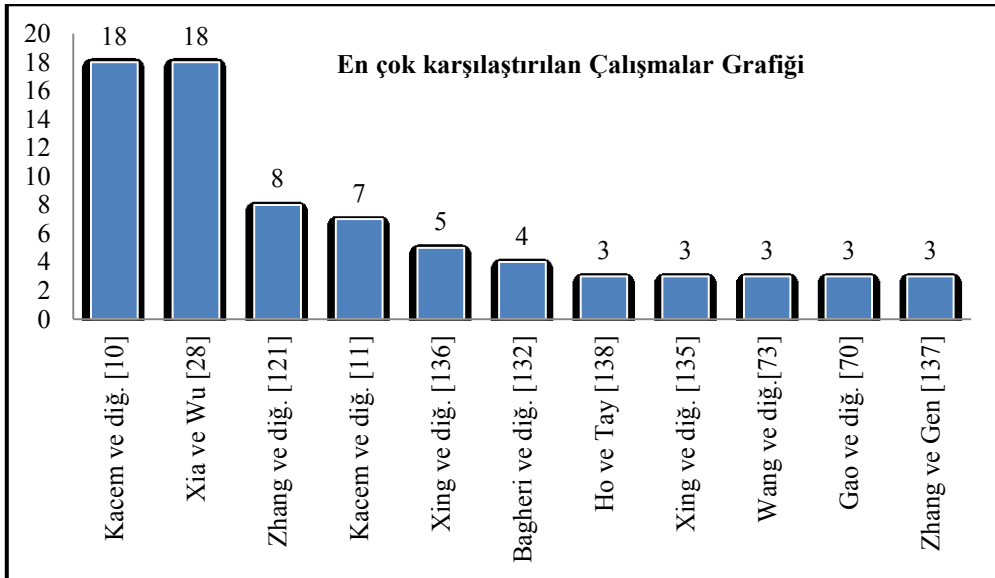
Şekil 3.7. Çok amaçlı EATÇ problemleri yıllar bazında yapılmış çalışmalar

Şekil 3.7 Çok amaçlı EATÇ problemleri üzerine incelenen toplam 47 makale içerisinde, yıllar bazında yapılan çalışmaları göstermektedir. Gerçek yaşam problemleri arasında yer alan EATÇ problemlerine olan ilgi zamanla artmıştır. Şekil 3.8 çok amaçlı EATÇ problemlerinde amaç fonksiyonu bazında yapılan çalışmaları göstermektedir. Araştırmacılar genelde üç amaç fonksiyonu olmak üzere C_{max} , W_t ve

W_m üzerinde daha çok yoğunlaşmışlardır. Yazarlar, Çok amaçlı çalışmalarında genelde amaçları ağırlıklandırarak, bazen de pareto optimal sonuçlar elde etmiş ve diğer çözüm sonuçlarıyla karşılaştırmışlardır. Şekil 3.9 Çok amaçlı EATÇ problemleri üzerine yapılan çalışmalarda yazarların kendi algoritma sonuçlarını literatürde diğer sonuçlardan hangisiyle en çok karşılaştırdıklarını göstermektedir. Grafikten görüleceği üzere, literatürde en çok karşılaştırılan çalışma Kacem ve diğ. [10] ve Xia ve Wu [28] sonuçlarıdır.



Şekil 3.8. Çok amaçlı EATÇ problemleri amaç fonksiyonu bazında yapılmış çalışmalar



Şekil 3.9. Çok amaçlı EATÇ problemlerinde en çok karşılaştırılan çalışmalar

4. GELİŞTİRİLMİŞ PARÇACIK SÜRÜ OPTİMİZASYONU

Bu çalışmada çok amaçlı EATÇ problemlerinin çözümü için hiyerarşik ve bütünleşik yaklaşımlı olmak üzere iki çeşit çözüm algoritması geliştirilmiştir. Hiyerarşik yaklaşımlı çok amaçlı PSO (HİÇAPSO) algoritması sıralama ve atama işlemlerini birbirinden bağımsız olarak gerçekleştirmektedir. Problemi çözme algoritması, sıralama ve atama problemleri olmak üzere 2 alt probleme ayrılmıştır. Öncelikle işlerin sıralanması gerçekleştirilmekte daha sonra da sıralanan bu işleri işlemek üzere atanacak makineler belirlenmektedir. HİÇAPSO çözüm yaklaşımında operasyon sıralamaları PSO algoritması ile operasyonların makinelere atanması işlemi ise YA algoritmasıyla gerçekleştirilmektedir. PSO çözüm algoritma adımları sadece işlerin sıralanma vektörü üzerinde uygulanmaktadır. Bütünleşik yaklaşımlı çok amaçlı PSO (BÜÇAPSO) algoritmasında ise operasyonların sıralama ve makine atamaları eş zamanlı olarak gerçekleştirilmektedir. Operasyonların işlenmesi için sıralama yapılırken aynı anda bu operasyonun işleneceği makine de belirlenmektedir. Atama ve sıralama işlemleri için öncelikle PSO algoritma adımları eş zamanlı uygulanmakta daha sonra yine atama ve sıralama işlemleri için YA algoritma adımları uygulanmaktadır. HİÇAPSO algoritmasında sıralama işlemine uygulanan PSO algoritma adımları BÜÇAPSO algoritmasında hem sıralama hem de atama işlemlerine uygulanmaktadır. HİÇAPSO algoritmasında sadece makine atama işlemlerine uygulanan YA algoritması BÜÇAPSO algoritmasında hem sıralama hem de atama işlemlerine uygulanmaktadır. PSO ve YA algoritmaları çözüm algoritmasına uygulanırken çok amaçlı hale getirilmiştir. PSO algoritmasında parçacık hareketleri P_{best} ve G_{best} konum vektörlerine göre hareket ettirilmektedir. Çok amaçlı problemimizin çözümünde pareto baskın çözümlere ulaşabilmek için parçacık hareketleri her amacı optimize edecek şekilde yeni konum vektörleri oluşturulmuştur. PSO'da uygulanan çok amaçlı yaklaşım aynı zamanda YA prosedürüne de uygulanmıştır. YA'da etkin komşu çözümler aranırken bir amaç yerine bütün amaçları optimize eden komşuluk yapıları araştırılmıştır. Aşağıda çok amaçlı EATÇ problemlerinin çözümü için önerdiğimiz HİÇAPSO ve BÜÇAPSO algoritmalarımız ayrıntılı bir şekilde sunulmuştur.

EATÇ Problemlerinin çok amaçlı optimizasyonu üzerine önerilen her iki algoritma öncelikle literatürde en çok çalışılan C_{max} , W_t ve W_m olmak üzere 3 amacın optimizasyonu olarak ele alınmıştır. Daha sonra EATÇ problemlerinde özellikle gecikmeleri optimize etmek üzere teslim tarihi göz önünde bulundurulmuştur. Bu nedenle 2. amaç grubu olarak da C_{max} , $\sum C$ ve $\sum T$ amaç fonksiyonları optimize edilmeye çalışılmıştır. Bu bölümde çok amaçlı EATÇ problemlerinin çok amaçlı PSO algoritması ile optimizasyonuna yönelik yapılan çalışma detaylandırılacaktır. Bölümün 2. kısmında sunulan çözüm yaklaşımlarına ait deneysel sonuçlar verilmiştir.

4.1. Çok Amaçlı EATÇ Problemleri Çözüm Yaklaşımları

Klasik ATÇ problemi, makine seti üzerinde işlerin sıralanması ile ilgilidir. EATÇ Problemi, ATÇ problemindeki bir operasyonun bir makine üzerinde işlenebilme kısıtının birden fazla makine üzerinde işlenebilme durumunun söz konusu olduğundan genişletilmiş halidir. EATÇ Probleminde her bir işin her bir operasyonu en az bir makine üzerinde işlenebilme durumu söz konusudur.

EATÇ Problemleri işlerin en iyi sırada işlenmesinin belirlenmesinin yanı sıra hangi işin hangi makinede işleneceğinin belirlenmesi alt problemlerini içermektedir. Literatürde bu iki alt problemin çözülmesine ilişkin, bütünlük ve hiyerarşik olmak üzere iki tür çözüm yaklaşımı geliştirilmiştir. Pezzella ve diğ. [29] bütünlük yaklaşımın diğer yaklaşıma göre zor olmasına rağmen, daha iyi sonuçlara ulaşılabildiğini belirtmişlerdir.

Hiyerarşik yaklaşımda işlerin makinelere atanması ve işleme sıralarının belirlenmesi ayrı ayrı gerçekleştirilmektedir. İşlerin sıralanması ayrı gerçekleştirilmekte, sonra sıralanan işlere makine atamaları yapılmakta daha sonra da bu iki alt problem birleştirilmektedir. Makine atamaları ve işlem sıralamasında birbirine müdahale olmamaktadır. Bu yaklaşımı ilk defa Brandimarte [2] sonra da Zribi ve diğ. [40] çalışmalarında kullanmışlardır. Bu yaklaşımın ana fikri, EATÇ problemlerinde rotalama ve çizelgeleme probleminin birbirinden ayrıştırılarak, problemin karmaşıklığının daha da azaltılabileceğidir.

Bütünleşik yaklaşımda işlerin sıralanması ve makine atamaları birbirinden bağımsız değildir ve eş zamanlı olarak gerçekleştirilmektedir. Çözüm sürecinde öncelikle işlerin sıralanması gerçekleştirilmekte, hemen ardından bu işleri işleyecek makineler belirlenmektedir. İşlerin sıralaması yapılırken bu işleri işleyecek makinelerde bellidir.

4.1.1. Çok amaçlı PSO

PSO, diğer evrimsel ve matematiksel temelli algoritmalara göre, fazla hafıza gerektirmeyen, etkili hesaplama kabiliyetine sahip, uygulaması kolay ve hızlı yakınsama özelliklerine sahip bir algoritmadır. PSO algoritmasında hedef, global ve yerel aramaların etkin uygulanarak, algoritmanın yerel optimumlarda takılmadan en iyi sonuca ulaşması amaçlanmaktadır.

Geliştirilen bu algoritma, daha önceden ortaya konmuş bazı yaklaşımlar temel alınarak yapılan iyileştirmelerle çözüm uzayının daha etkin taranması sağlanmıştır. Bu sayede daha yüksek kalitede çözümlere ulaşılmıştır.

PSO algoritması rastgele çözümler içeren bir popülasyonla başlatılmakta ve bu çözümler güncellenerek optimum çözüm araştırılmaktadır. Parçacık olarak adlandırılan aday çözümler, kuşların yiyecek ararken yiyeceğe en yakın kuşu takip etmeleri gibi, o andaki optimum parçacığı izleyerek problem uzayında dolaşmaktadırlar. Parçacık hareket ettiğinde, kendi koordinatlarını bir fonksiyona göndermekte, böylece parçacığın uygunluk değeri yani yiyeceğe ne kadar uzaklıkta olduğu ölçülmektedir. Bir parçacığa ait hız ve yön bilgisinin her seferinde nasıl değişeceği, kendi koordinatları ile komşu parçacıkların en iyi koordinatlarının birleşimi ile belirlenmektedir. Ajanların kendi tecrübelerine ve komşu ajanların tecrübelerine göre hareket etmeleriyle ajanlar arasında bilgi paylaşımı sağlanmaktadır.

Sürüdeki her bir parçacık arama uzayında bir çözümü temsil eder. Parçacıklar, ilk pozisyonlarını genelde rastgele belirler, daha sonra çözüm uzayında adım adım hareket eder. Parçacığın her iterasyonda hangi yönde ve hangi hızda hareket edeceği, kendi kişisel en iyi koordinatları ve sürü içindeki diğer parçacıkların o ana kadar en iyi koordinatlarının birleşiminden belirlenmektedir. Parçacık optimum

sonuca ulaşırken, bir sonraki adımını hem kendi en iyi pozisyonuna hem de sürüdeki en iyi pozisyona göre ayarlamaktadır.

HİÇAPSO ve BÜÇAPSO algoritmalarının çözüm adımları 3 iş 4 makine ve toplam 8 operasyondan oluşan tam esnekliğe sahip 3x4x8 problem üzerinde ayrıntılı verilmiştir. Problem verileri rastsal oluşturulmuştur. Probleme ait işlem süreleri Tablo 4.1’de verilmiştir.

Tablo 4.1. Örnek problem işlem süreleri

		M1	M2	M3	M4
J ₁	O ₁₁	4	7	3	2
	O ₁₂	3	1	3	5
	O ₁₃	4	5	6	7
J ₂	O ₂₁	1	3	4	6
	O ₂₂	4	5	3	2
	O ₂₃	6	2	1	3
J ₃	O ₃₁	8	3	4	5
	O ₃₂	4	3	6	2

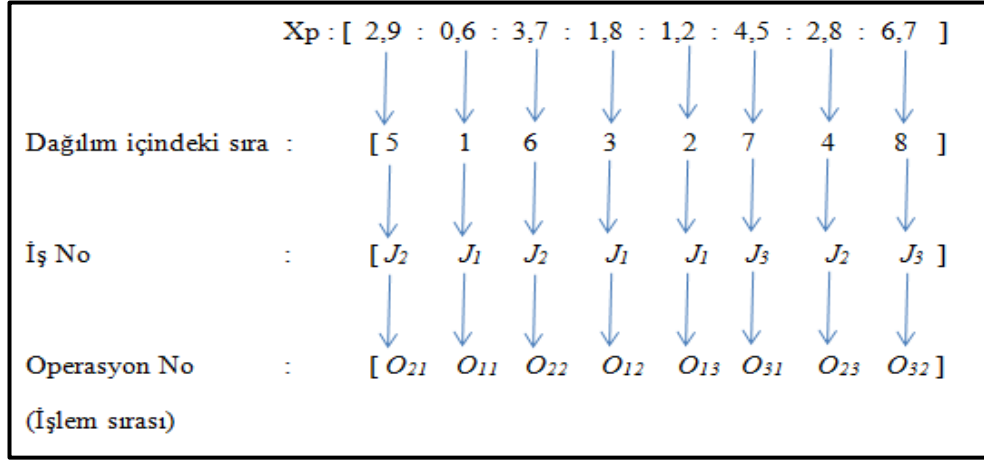
Pozisyon Vektörü: k. İterasyondaki i. parçacığın pozisyon vektörü X_i^k ile temsil edilir ve problem boyutu kadar eleman içerir. Denklem (4.1)’de verildiği gibi gösterilir.

$$X_i^k = [x_{i1}^k, x_{i2}^k, x_{i3}^k, \dots, x_{ij}^k] \quad (i=1,2,\dots,p) \quad (4.1)$$

Burada, x_{i1}^k , k. iterasyonda i. parçacığın pozisyon vektöründeki birinci elemanını, p, parçacık sayısını ifade eder. Burada problem boyutu n adet iş ve m adet makineden oluşan problem için, $j=nxm$ kadardır.

EATÇ Problemleri operasyonların hem makine üzerinde sıralanması hem de işlenmek üzere makinelere atanması alt problemlerinden oluştuğu için problemin çözümünde, pozisyon vektörü olarak her parçacık için 2 tane vektör oluşturulmaktadır. Bunlardan birincisi, Xp_i^k vektörü işlerin sıralama vektörüdür. Xp vektörü oluşturulurken D, operasyon sayısını temsil etmek üzere $[-D,D]$ aralığında rastsal üretilmektedir.

EATÇ Problemlerinin PSO ile çözümü gerçekleştirilirken oluşturulan 2. pozisyon vektörü ise, Xm_i^k makine atama vektörüdür. Xm vektörü oluşturulurken M makine sayısını temsil etmek üzere $[0,M]$ aralığında oluşturulur.



Şekil 4.1. X_p Vektör temsili

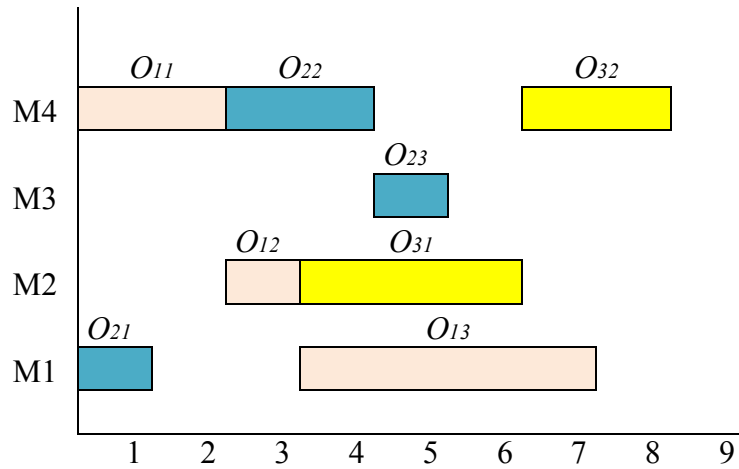
Başlangıç popülasyonunda 1. parçacık için X_p vektörünü Şekil 4.1’ deki gibi rastsal oluşturalım. Bu vektörün operasyon temsilleri için, X_p vektöründeki rakamlar küçükten büyüğe doğru sıralanır. Dağılım içindeki sıra satırı, rakamların küçükten büyüğe doğru sıralanışını göstermektedir. Vektörde en küçük rastsal rakam 0,6 olduğu için 1. sırayı, 1,2 rakamı 2. sırayı alır. Örnek problemde J_1 işinin 3 operasyonu olduğu için X_p vektöründeki ilk üç rakam J_1 işi olarak atanır. İş no satırı, işlerin sıralanmasını göstermektedir. Her bir işe ait operasyonlar arasında işlem öncelik kısıtı olduğundan dolayı X_p vektörünün J_1 işine ait ilk üç rakamı 1. operasyondan başlanarak atanır. Operasyon no satırı da operasyonların sıralanmış halini göstermektedir. Bu durumda X_p vektörüne ait işlem sırası $X_p = \{ O_{21}, O_{11}, O_{22}, O_{12}, O_{13}, O_{31}, O_{23}, O_{32} \}$ şeklinde olur.

Xm vektörü makine atamalarını temsil eden vektördür. Çok amaçlı PSO yönteminde çözüm yapılırken Xm vektörü $[0,M]$ aralığında operasyon sayısı kadar oluşturulmaktadır. X_p vektörü ile işlem sıraları tespit edilmişti. X_p Vektörü başlangıç popülasyonunda rastsal oluşturulurken, algoritmanın ilerleyen iterasyonlarında daha etkin sonuçlara ulaşabilmek için Xm vektörü EKİZ kuralına göre oluşturulmuştur. Örnek problem üzerinde Xm vektörünün oluşumu Şekil 4.2’ de verildiği gibidir. Makine atamaları EKİZ kuralına göre atandığı için her operasyon en kısa işlem süreli makinaya ataması yapılır.

İşlem Sırası	:	$[O_{21} \quad O_{11} \quad O_{22} \quad O_{12} \quad O_{13} \quad O_{31} \quad O_{23} \quad O_{32}]$
Atandığı Makine	:	$[M1 \quad M4 \quad M4 \quad M2 \quad M1 \quad M2 \quad M3 \quad M4]$
X_m vektörü	:	$[1 \quad 4 \quad 4 \quad 2 \quad 1 \quad 2 \quad 3 \quad 4]$

Şekil 4.2. X_m Vektör temsili

İşlem sıra ve makine atamaları yapılmış 1. iterasyonun 1. parçacığına ait çözüm $X_1^1 = \{O_{211}, O_{114}, O_{224}, O_{122}, O_{131}, O_{312}, O_{233}, O_{324}\}$ şeklinde olur. Buradaki çözüme göre, 2. işin 1. operasyonu 1 nolu makineye, 1. işin 1. operasyonu 4 nolu makineye, 2. işin 2. operasyonu 4 nolu makineye şeklinde devam eder. Problem çözümünde parçacık sayısı kadar X_p ve X_m vektörleri oluşturulur. Bu parçacığa ait çözümün Gantt şeması Şekil 4.3'te verilmiştir. Amaç fonksiyonu olarak $C_{\max}=8$, $W_t=16$ ve $W_m=6$ olur. Yeni amaç fonksiyon değerleri olarak; $C_{\max}=8$, $\sum C=20$ ve $\sum T=5$ olarak elde edilir.



Şekil 4.3. Örnek problem çözümü

Hız Vektörü: PSO parçacıklar arasındaki bilgi paylaşımını esas almaktadır. Her bir parçacık yeni pozisyonunu, önceki tecrübelerinden ve sürüdeki en iyi pozisyonu dikkate alarak bulur. Global komşular, her bir iterasyonda sürü içerisindeki tüm parçacıklar arasından en iyi uygunluk fonksiyonu değerini veren parçacığın değeridir. Yerel komşular ise, ilk iterasyonda her bir parçacığın pozisyon değeri olmak üzere, her iterasyonda parçacık sayısı kadar yerel en iyi değer ortaya çıkar. Her bir parçacık, yerel ve global komşu değerlerini kullanarak bir sonraki

iterasyonda yeni konumunu belirler. Durdurma kriteri ise genellikle istenen amaç fonksiyonu değerine veya iterasyon sayısına ulaşılmasıdır. Parçacık sürü optimizasyonunda önemli olan her bir parçacığın optimum sonuca ulaşması yerine sürü içerisindeki bir parçacığın optimum sonuca ulaşmasıdır.

Hız vektörü, parçacığın bir sonraki konumunu belirleyen parametrelerden biridir ve V_i^k ile ifade edilir. Sürü içerisindeki bir parçacığın hız vektörü Denklem (4.2)'de verildiği şekilde gösterilir. Burada v_{i1}^k , k. iterasyondaki i. parçacığa ait hız vektörünün 1. elemanını temsil etmektedir.

$$V_i^k = [v_{i1}^k, v_{i2}^k, v_{i3}^k, \dots, v_{i(nxm)}^k] \quad (4.2)$$

$$V_i^{k+1} = W \times V_i^k + c_1 \times \text{Rand}_1^k \times (P_{best_i}^k - x_i^k) + c_2 \times \text{Rand}_2^k \times (G_{best}^k - x_i^k) \quad (4.3)$$

$$W = W_{max} - \frac{W_{max} - W_{min}}{\text{iter}_{max}} \times \text{iter} \quad (4.4)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (4.5)$$

Denklem (4.3)'te bir sonraki iterasyon için yeni konum vektörünün hızının hesaplanması verilmiştir. Rand_1^k ve Rand_2^k her iterasyonda $[0,1]$ arasında düzgün dağılımlı rastsal değerlerdir. c_1 ve c_2 sabitleri öğrenme faktörleridir. c_1 sabiti parçacığın kendi en iyi pozisyonuna doğru hareket etmesini sağlar. c_2 sabiti sürüdeki diğer parçacıkların en iyi pozisyonuna göre parçacıkların hareket etmesini sağlar. Literatürde genelde $c_1=c_2=2$ alındığı görülmektedir. w Atalet ağırlığı, bir önceki iterasyona ait hız vektörünün mevcut iterasyondaki hız vektörü üzerindeki etkisini kontrol eden parametredir. Atalet ağırlığı, parçacığın yerel ve global araştırma kabiliyeti arasında değişim etkisi vardır. Atalet ağırlığının büyük bir değer seçilmesi global aramada iyileşmeyi, küçük bir değer seçilmesi ise yerel aramada iyileşmeyi sağlar. İterasyonlar boyunca atalet ağırlığı Denklem (4.4)'te verilen formülasyonla hesaplanmıştır. P_{best} bir parçacığın mevcut iterasyona kadar elde ettiği en iyi değeri ifade eder. G_{best} değeri ise, popülasyon içindeki tüm parçacıklar içinde en iyi değere sahip parçacığın aldığı değerdir. Bir çözüm uzayında en fazla iterasyon sayısı kadar G_{best} vardır. Ancak çok amaçlı problemlerin optimizasyonunda G_{best} sayısı iterasyon sayısını geçebilir. Bu durumda amaç fonksiyonları arasında en iyilenmesi düşünülen

amaç fonksiyonuna sahip olan G_{best} alınır. Denklem (4.5) parçacığın bir sonraki iterasyonda gideceği konumu vermektedir.

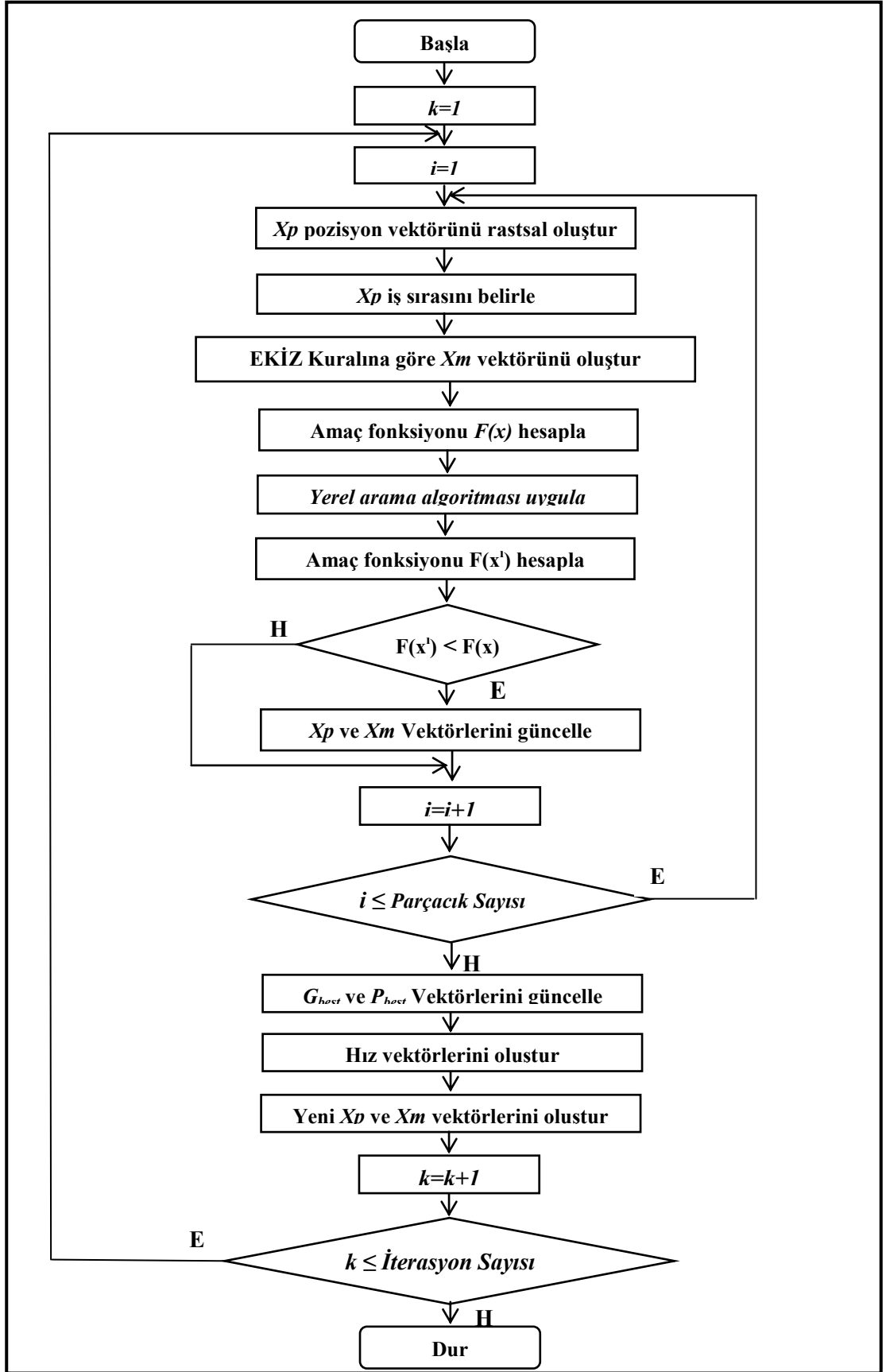
4.1.2.Bütünleşik çok amaçlı PSO algoritması

EATÇ Problemlerinin çözümü için sıralama kuralları gibi klasik yöntemler, TB, GA ve TA gibi çeşitli meta sezgisel yöntemler geliştirilmiştir. Yazarlar bu çalışmalarını hiyerarşik ve bütünleşik olmak üzere iki grupta birleştirmişlerdir. Hiyerarşik yaklaşımda problem zorluğunu azaltmak için problem sıralama ve atama alt problemlerine ayrılır. Tipik bir hiyerarşik çözüm yaklaşımında önce operasyonların makinelere ataması yapılır sonra da ATÇ probleminde olduğu gibi operasyonların sıralanması gerçekleştirilir. Brandimarte [2], Paulli [145], Barnes ve Chambers [146] hiyerarşik olarak EATÇ problemlerini çözmek için önce makine atamalarını gerçekleştirmiş sonrada problemi ATÇ problemi haline dönüştürerek TA yöntemini kullanmışlardır. Saidi-Mehrabad ve Fattahi [43], EATÇ problemine hiyerarşik yaklaşımlı TA algoritması geliştirmişlerdir. Yazarlar, önce operasyonları işleme kabiliyetine sahip makine grubu içinden uygun makinelere atamayı gerçekleştirmişler daha sonra da atanan işleri C_{max} 'ı minimize edecek şekilde sıralamayı gerçekleştirmişlerdir.

Literatürde bazı yazarlar, bütünleşik yaklaşımı, hiyerarşik yaklaşıma göre daha zor çözümü olan ancak genel olarak daha iyi sonuçlar veren çözüm yaklaşımı olarak tanımlamışlardır. Vaessens ve diğ. [147], Dauzere-Peres ve Paulli [60], Hurink ve diğ. [59], Mastrolilli ve Gambardella [48] TA algoritması ile EATÇ problemlerini bütünleşik çözüm yaklaşımı ile çözmüşlerdir. Chen ve diğ. [148], Jia ve diğ. [149], Ho ve Tay[150] ile Kacem ve diğ. [10] aynı problemin çözümü için bütünleşik GA yaklaşımlarını sunmuşlardır.

Çok amaçlı EATÇ problemlerinin çözümünde kullanılan BÜÇAPSO algoritması akış diyagramı Şekil 4.4'de verilmiştir. BÜÇAPSO algoritmasında, rastsal olarak oluşturulan X_p vektöründen sonra, EKİZ kuralına göre X_m vektörü oluşturulmaktadır. Başlangıç popülasyonunda elde edilen X_p ve X_m vektörlerine göre amaç fonksiyon değeri ($F(x)$) hesaplanmaktadır. Başlangıç popülasyonlarına YA algoritması uygulanmakta, daha sonra yeni amaç fonksiyon değeri ($F(x')$) tekrar hesaplanmaktadır. Elde edilen yeni ($F(x')$), başlangıç popülasyonunda elde edilen

($F(x)$) değerinden daha iyi ise YA algoritması ile elde edilen çözüme göre X_p ve X_m vektörleri güncellenmektedir. Karar verici tarafından belirlenen başlangıç popülasyon sayısı kadar bu işlem devam etmektedir. Bu aşamadan sonra, P_{best} ve G_{best} konum vektörleri belirlenmektedir. Algoritma çok amaçlı olarak tasarlandığından dolayı, algoritmanın tüm adımlarında amaç fonksiyon değerleri hesaplanırken, P_{best} ve G_{best} konum vektörleri belirlenirken Pareto optimal algoritması uygulanmaktadır. Bu algorithmada her amacın en iyi değere sahip olduğu vektörler P_{best} ve G_{best} vektörleri olarak saklanmaktadır. Parçacıkların her iterasyonda yeni hız ve konum vektörleri belirlendikten sonra yeni konum vektörlerine göre amaç fonksiyon değerleri tekrar hesaplanmaktadır. Daha sonra güncel konum vektörlerine tekrar YA algoritması uygulanmaktadır. Yeni vektörlere uygulanan YA algoritması ile daha iyi sonuç alınması durumunda konum vektörleri YA ile elde edilen çözüme göre tekrar güncellenmektedir. Karar verici tarafından belirlenen iterasyon sayısına ulaşıldığında algoritma durmaktadır.



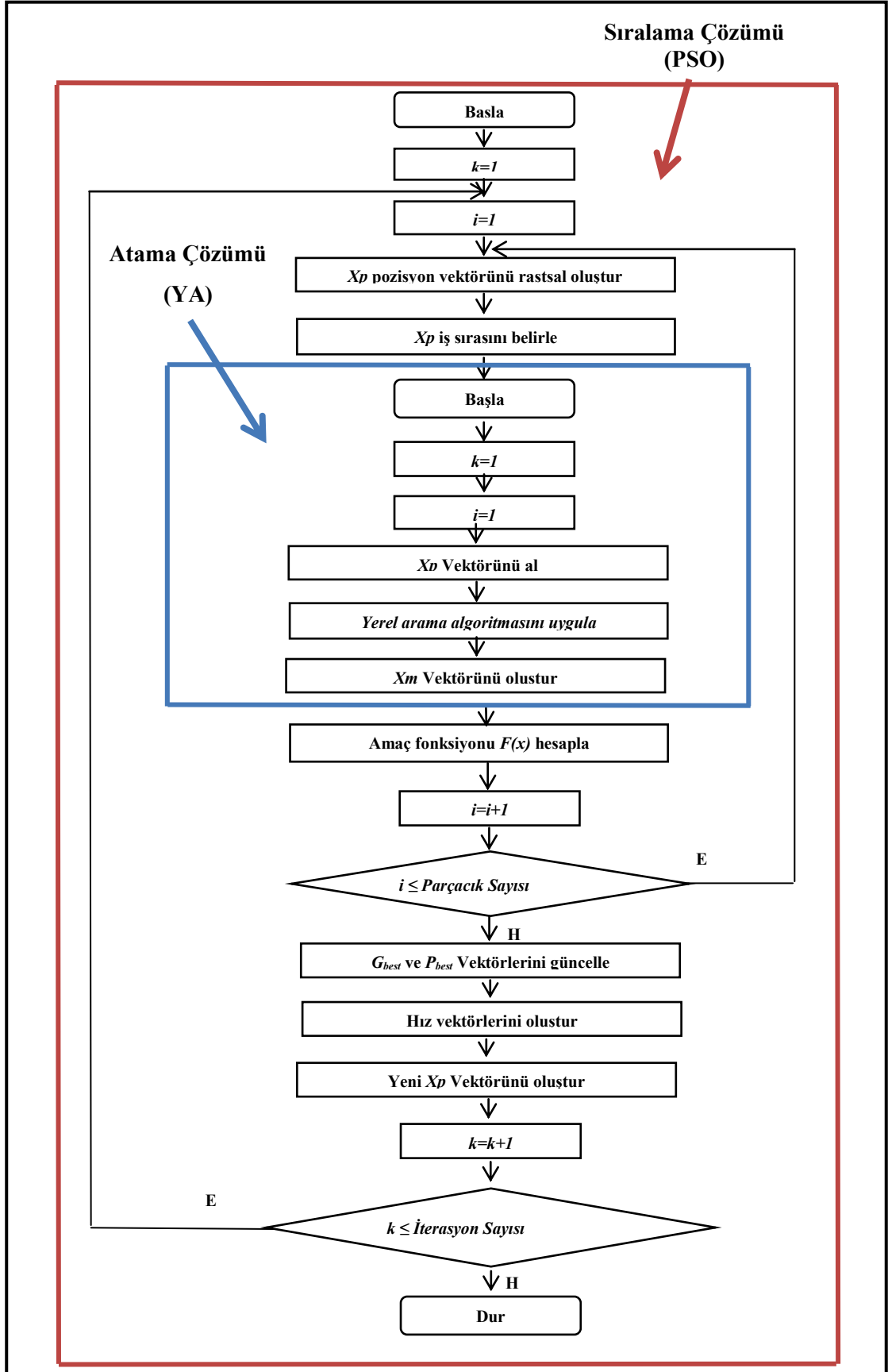
Şekil 4.4. BÜÇAPSO Algoritması akış diyagramı

4.1.3.Hiyerarşik çok amaçlı PSO algoritması

EATÇ Problemlerinin hiyerarşik çözümünde operasyonların makinelere ataması daha sonra bu makineler üzerinde operasyonların sıralanması gerçekleştirilir. Atama ve sıralama işlemleri birbirlerinden bağımsız olarak yapılır. Hiyerarşik yaklaşımın ana fikri problemin zorluk derecesini azaltmaktır. EATÇ Problemleri atama ve sıralama alt problemlerinden oluştuğu için hiyerarşik çözüm yaklaşımı bu problem türünün doğasından gelmektedir.

Fattahi ve diğ.[12] hiyerarşik çözüm yaklaşımında, atama problemini ana problem, operasyonların sıralanması problemini ise alt problem şeklinde ele alarak çözüm geliştirmiştir. Kendi çalışmamızda sıralama problemi ana problem, atama problemini ise alt problem olarak ele alınmıştır. Şekil 4.5'te EATÇ problemlerinin çözümü için geliştirilen HİÇAPSO algoritması akış diyagramı verilmiştir. Çok amaçlı optimizasyon problemi ele alındığından dolayı çözüm algoritması da çok amaçlı hale getirilmiştir. Üç amaçlı optimizasyon problemini ele aldığımızdan dolayı algoritma 1. amacı göz önünde bulundurarak çalışmaya başlamakta, istenen iterasyon sayısına ulaşıldığında 2. amaç için algoritma çalışmakta, aynı şekilde 3. amaç içinde aynı şartlar sağlandıktan sonra da bu üç amacın toplamını optimize edecek şekilde verilen iterasyon sayısı kadar algoritma çalışmaktadır.

HİÇAPSO çözüm algoritmasında operasyonların sıralama işlemleri PSO algoritmasıyla, makinelere atama işlemleri ise YA algoritmasıyla gerçekleştirilmiştir. PSO algoritmasında rastsal olarak oluşturulan başlangıç popülasyonları ile işlerin sıralanması (X_p vektörü) tespit edilmektedir. Belirlenen iş sırasına göre YA algoritmasıyla operasyonların makine atamaları gerçekleştirilmektedir. Bu aşamaya kadar yapılan işlemler birbirlerinden bağımsız olarak gerçekleştirilmektedir. PSO algoritmasıyla gerçekleştirilen sıralama işlemlerinde algoritma yapısında bulunan yeni hız ve konum vektörlerinin belirlenmesi, G_{best} ve P_{best} konum vektörlerinin belirlenmesi sadece X_p vektörü üzerinde gerçekleştirilmektedir. BÜÇAPSO algoritmasında hem sıralama hem de atama işlemlerine uygulanan PSO algoritma adımları HİÇAPSO algoritmasında sadece sıralama işlemine uygulanmaktadır.



Şekil 4.5. HİÇAPSO Algoritması akış diyagramı

4.1.4.Yerel arama algoritması

Meta sezgisel yöntemler, optimizasyon problemlerinde kısa sürede optimum veya optimuma yakın sonuçlar elde edebilmek için kullanılan yöntemlerdir. Bu yöntemler optimum çözümün elde edilmesini garanti etmemekle birlikte optimuma yakın sonuçların elde edilme olasılığının yüksek olduğu algoritmalarlardır. YA Algoritmaları ise meta sezgisel yöntemler ile elde edilen çözümleri daha da geliştirmeye yönelik stratejilerin olduğu algoritmadır.

Optimizasyon problemlerinde çözüm kümesi çok büyük bir küme olabilir. Bu durumda çözüm kümesi içerisinde bir alt küme seçilerek çözüm araştırılır. Böyle durumlarda ise YA algoritmaları kullanılır. YA Algoritması çözüm kümesinin alt kümesinde bir nokta belirler ve bu noktanın komşularını arar. Eğer aranan komşu çözümün uygunluk değeri, mevcut çözümün uygunluk değerinden daha iyi ise komşu çözüm mevcut çözüm olarak belirlenir.

Bir YA algoritmasının etkili olabilmesi için hem yeni arama uzaylarını keşfedebilmesi hem de bilinen arama uzayında olabilecek çözümleri gezebilmesi gerekmektedir. Bu iki ihtiyaç birbiri ile çelişmektedir. Bu sebeple iyi bir YA algoritması bu ikisini dengeleyebilmelidir. Sadece rastgele arama yapan bir algoritma yeni yerler keşfedebilirken, sadece tepe tırmanma algoritması yapan bir arama algoritması bilinen alanlarda gezmektedir.

YA Algoritmaları genellikle diğer algoritmalar tarafından bulunmuş bir başlangıç çözümle çalışmaya başlar ve daha iyi bir komşu çözüme taşınma yoluyla yinelemeli olarak çözümü geliştirir. Çözümü daha fazla geliştirebilecek hiçbir çözüm bulunamadığı takdirde algoritma bir yerel optimumda sonlanır[151].

Özel bir problemde bir YA algoritması elde etmek için başlangıç çözümü bulan bir algoritma ve bir komşuluk yapısına ihtiyaç vardır. Herhangi bir uygun çözümün tüm komşularını bulmak gerekmez, ancak tek bir tane iyileştirici komşu bulunması yeterlidir. NP-zor problemler için optimal çözüme ulaşmayı sağlayan bir komşuluk yapısı polinom zamanda bulunamaz. Bu durumda yaklaşık çözüm üreten YA algoritmaları araştırılır[152].

Komşuluk yapıları ile mevcut çözümde, problem türüne bağlı olarak belirli kurallar dâhilinde değişiklikler yapılarak yeni çözümler elde edilir. Çizelgeleme problemlerinde mevcut çizelgelerden yeni çizelgeler elde etmek için komşuluk yapıları kullanılır. Komşuluk yapısına göre farklı komşu çizelgeler elde edilir. Örneğin, bir makinede ardarda işlem gören işler ikişerli veya üçerli grup halinde yer değiştirilerek farklı komşuluk yapıları elde edilebilir. Literatürde yazarlar tarafından önerilen farklı komşuluk yapıları aşağıda incelenmiştir.

N1 Komşuluğu: Van Laarhoven ve diğ. [153] tarafından ATÇ problemlerinin çözümünde kullanılan N1 Komşuluk yapısına göre, aynı makinede işlem gören bitişik iki işin yer değiştirmesiyle yeni çizelgeler elde edilir. Bu komşuluk yapısıyla elde edilen yeni çözümün olurlu bir çözüm olması gerekir. N1 komşuluğunun komşuluk hacmi diğer yapılara göre daha büyüktür ve çözümü geliştirmeyen hareketleri de içerir. Örneğin aynı blokta, arada kalan işlemlerin yer değiştirmesi tamamlanma süresi üzerinde hiç bir zaman bir iyileşmeye neden olmamaktadır [154]. Bu yüzden, başka yazarlar tarafından farklı komşuluk yapıları geliştirilmiştir.

N2 Komşuluğu: Barnes ve Chambers [155] tarafından önerilen N2 komşuluğu, bir makine üzerinde yer alan çift işin yer değiştirmesi veya makine başında veya sonunda yer alan işlerin yer değiştirmesi şeklinde tanımlanmıştır. N2 komşuluk yapısında, değişmeye aday işlem çiftlerinin belirlenmesi için her makine gözden geçirilir. N2 komşuluğunda, bir blokta ara işlemlerin yer değiştirmesi önlenerek, iyileşmeye neden olmayan gereksiz değişimler yapılmaz ve daha az sayıda komşu değerlendirilerek zaman kaybı önlenir. Ancak bazı durumlarda N2 komşuluğu ile optimum çözüme ulaşılamayabilir.

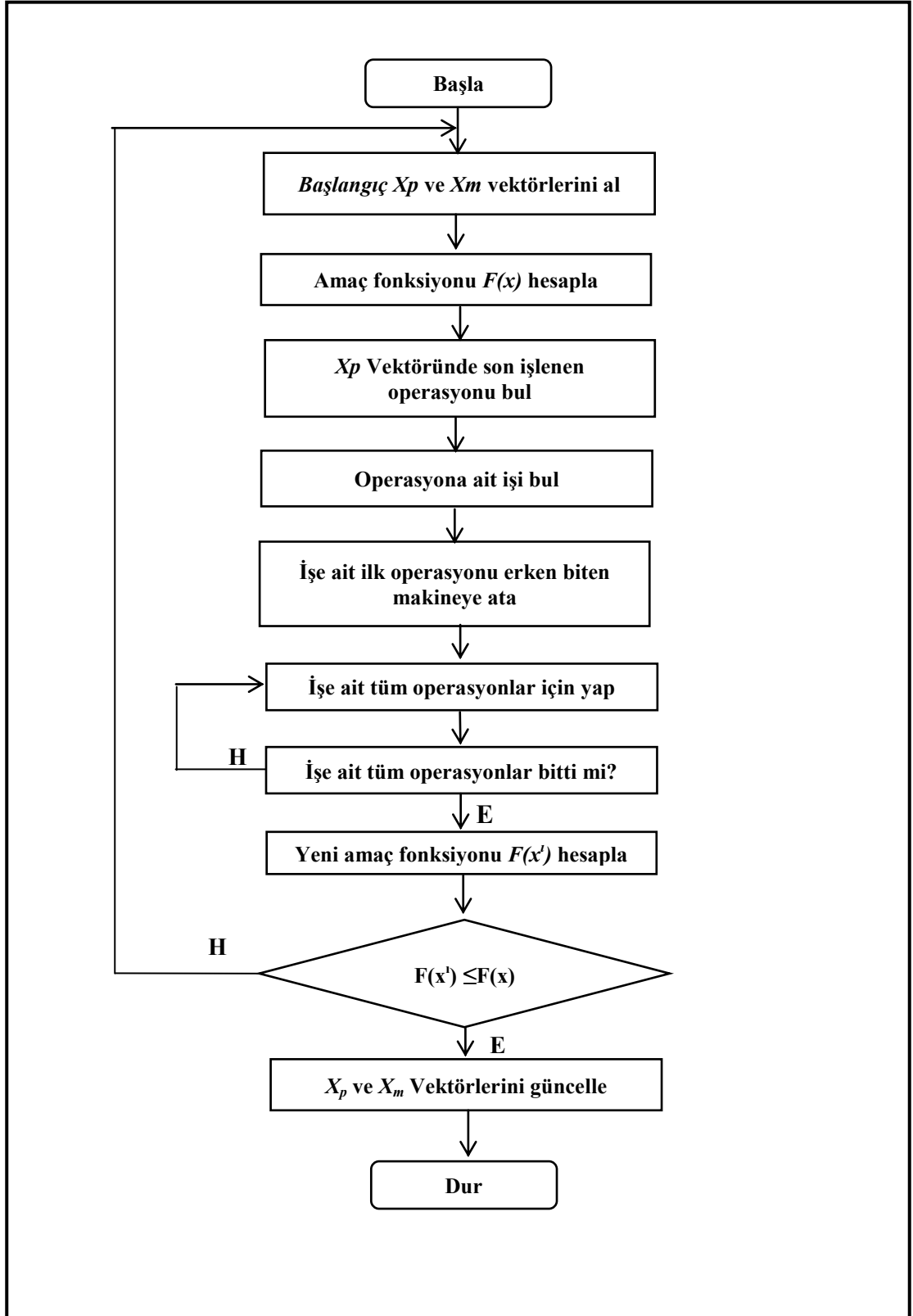
N3 Komşuluğu: Dell'Amico ve Trubian [156] tarafından geliştirilen N3 komşuluğu, N1 komşuluk yapısını temel alarak, aynı anda birden fazla yer değiştirmeyi kapsar. Aynı makine üzerinde işlem gören iki ardışık işlemin tüm olası değişimlerinin gerçekleştirilmesi esasına dayanır.

YA, çizelgeleme alanında özellikle EATÇ problemlerinde sıklıkla uygulanan sezgisel tekniklerdendir. YA algoritmasında, bir başlangıç çizelgesinden hareketle, bu çizelgedeki işlerin yerlerinin belirli kurallara göre değiştirilmesi ile yeni komşu çizelgeler elde edilir, bu süreç belli bir ölçüt sağlanıncaya kadar devam ettirilir ve bu

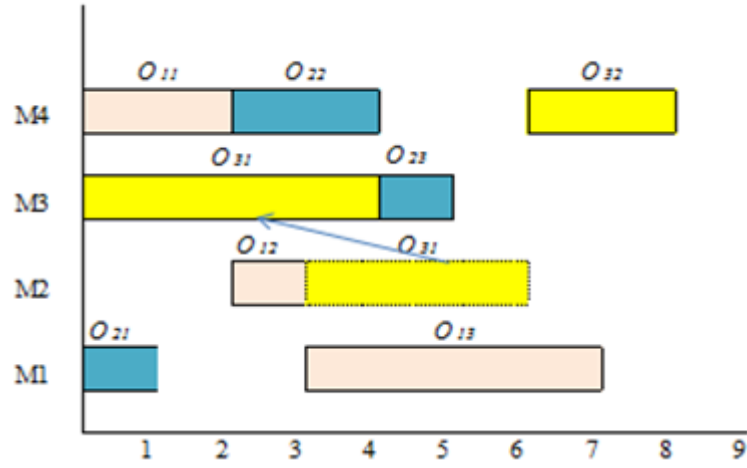
ölçüt sağlandığında algoritma sonlandırılır. Elde edilen yeni çizelgeler daha iyi uygunluk değerine sahip ise çözüm kümesi güncellenir.

Çok amaçlı EATÇ problemlerinin çözümü için geliştirilen algorithmada uygulanan YA algoritması Şekil 4.6’ da verilmiştir. BÜÇAPSO algoritmasıyla PSO’da oluşturulan sıralama (X_p) ve atama (X_m) vektörleriyle HİÇAPSO algoritmasında atama (X_m) vektörlerine YA algoritması uygulanmaktadır. Öncelikle son biten operasyon ve bu operasyona ait iş tespit edilmektedir. EATÇ Problemlerinin operasyonları arasında işlem kısıtı olduğundan dolayı işe ait ilk operasyondan başlanarak yeniden makine atamaları ve bu makineler üzerindeki sıralamalar yapılır. Çözüm algoritmalarımız çok amaçlı tasarlandığından dolayı YA algoritmasında da Pareto çözümler göz önünde bulundurulmaktadır. Bir işe ait operasyonların yeniden ataması ve sıralanması gerçekleştirilirken amaç fonksiyonu olarak C_{max} yanı sıra W_t ve W_m amaç fonksiyonları ile birlikte bu 3 amaç fonksiyonunun toplamının da optimizasyonu göz önünde bulundurulmaktadır.

Örnek problem üzerinde YA algoritmasının uygulanması aşağıdaki gibi gerçekleştirilir. Sıralama problemi için X_p vektörüne ait işlem sırası; $X_p = \{ O_{21}, O_{11}, O_{22}, O_{12}, O_{13}, O_{31}, O_{23}, O_{32} \}$, makine atama problemine ait X_m vektörü; $X_m = \{ M1, M4, M4, M2, M1, M2, M3, M4 \}$ ve 1.parçacığa ait çözüm $X_1^1 = \{ O_{211}, O_{114}, O_{224}, O_{122}, O_{131}, O_{312}, O_{233}, O_{324} \}$ şeklinde elde edilmişti. Bu çözüme ait Gantt şeması Şekil 4.3’te verilmiştir. Gantt şeması göz önüne alındığında en son biten operasyon O_{32} ’dir. Bu operasyonun bağlı olduğu J_3 işine ait ilk operasyon O_{31} belirlendikten sonra bu operasyonun daha erken işlenebileceği boş makine arayalım. O_{31} operasyonu normalde 2 no’lu makinede 3 birim zamanda işlemi biterken, 3 no’lu makinede 4 birim zamanda işlem görmesi gerekiyor. 3 No’lu makine 5.birim zamana kadar boşluğu vardır. Şekil 4.7’ de görüldüğü gibi O_{31} operasyonu 2 nolu makineden 3 nolu makineye kaydırılabilir.

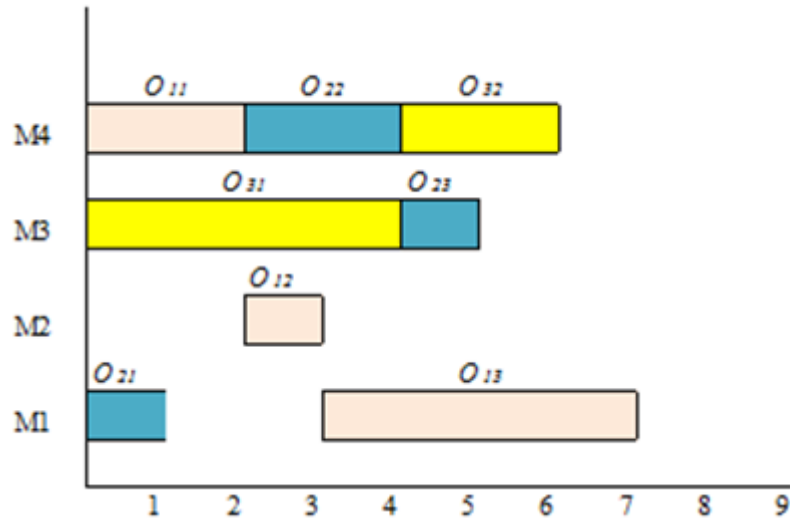


Şekil 4.6. YA Akış diyagramı



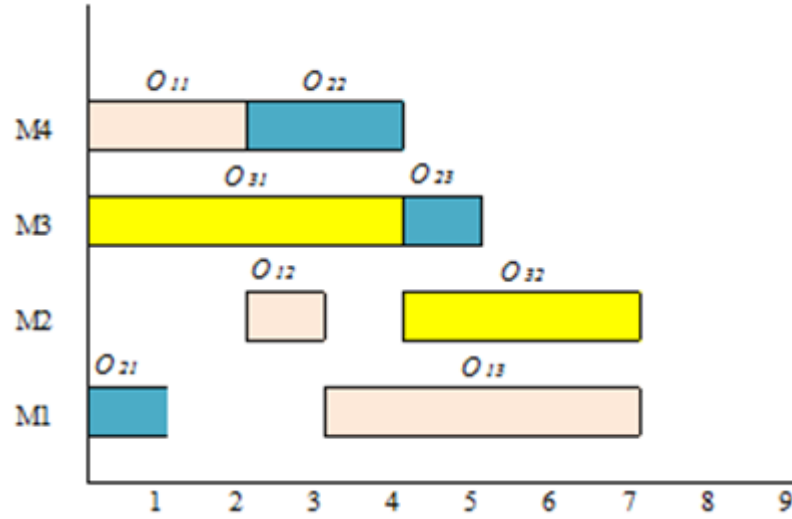
Şekil 4.7. YA Algoritma uygulaması

3 nolu işe ait 1. Operasyon (O_{31}), 3 nolu makineye atandıktan sonra, 2.operasyon (O_{32}) için iki alternatif söz konusu olmaktadır. Normalde 4 No'lu makinede 2 birim zaman işlem gören O_{32} operasyonu 2 birim zaman sola kaydırılarak operasyon tamamlanabilir. Bu durumda Şekil 4.8' te verilen Gantt şeması elde edilir. Amaç fonksiyonu, $C_{\max}=7$, $W_t=17$ ve $W_m=6$ olur. Yeni amaç fonksiyon değerleri olarak; $C_{\max}=7$, $\sum C=18$ ve $\sum T=3$ olarak elde edilir.



Şekil 4.8. YA Algoritması 1.alternatif uygulaması

Diğer alternatif O_{32} operasyonu 4 nolu makine yerine 2 nolu makinede 3 birim zaman işlem görür. Bu durumda Şekil 4.9' daki Gantt şeması elde edilir. Bu duruma göre amaç fonksiyonu, $C_{\max}=7$, $W_t=18$ ve $W_m=5$ şeklinde olur. Yeni amaç fonksiyon değerleri olarak; $C_{\max}=7$, $\sum C=19$ ve $\sum T=4$ olarak elde edilir.



Şekil 4.9. YA Algoritması 2.alternatif uygulaması

4.1.5.Pareto optimal algoritması

Çok amaçlı optimizasyon problemlerinde, bir çözümün tüm amaçlara göre en iyi olması çok zordur. Bu tür çözümler ideal çözüm olarak adlandırılır. Genellikle bir tek çözüm yerine çözüm kümesinden bahsedilir. Bu çözümlere Pareto (veya baskın) çözüm denilmektedir. Çok amaçlı optimizasyon teorisinin esas yapısı, basılgın çözümleri çözüm kümesinden eleyerek baskın olan iyi çözümlere ulaşmayı amaçlar.

Çok amaçlı optimizasyon problemlerinde karar vericinin sonuçlar içinden tercihini yapabilmesi için pareto baskınlık kavramı sunulmuştur. Elde edilen sonuçlar içerisinde bir çözüm kümesindeki bir amaç diğer çözüm kümesindeki aynı amaçtan daha iyi ve aynı çözüm kümelerindeki kalan diğer amaçlardan daha kötü değilse bu çözüm kümesi baskın çözüm olarak kabul edilir. Eğer herhangi bir çözüm kümesindeki bir amaç, diğer çözüm kümesindeki aynı amaçtan daha kötü ve diğer amaçlar diğer çözüm kümelerindeki aynı amaçlarla aynı ise bu çözüm kümesi basılgın çözüm olarak kabul edilir.

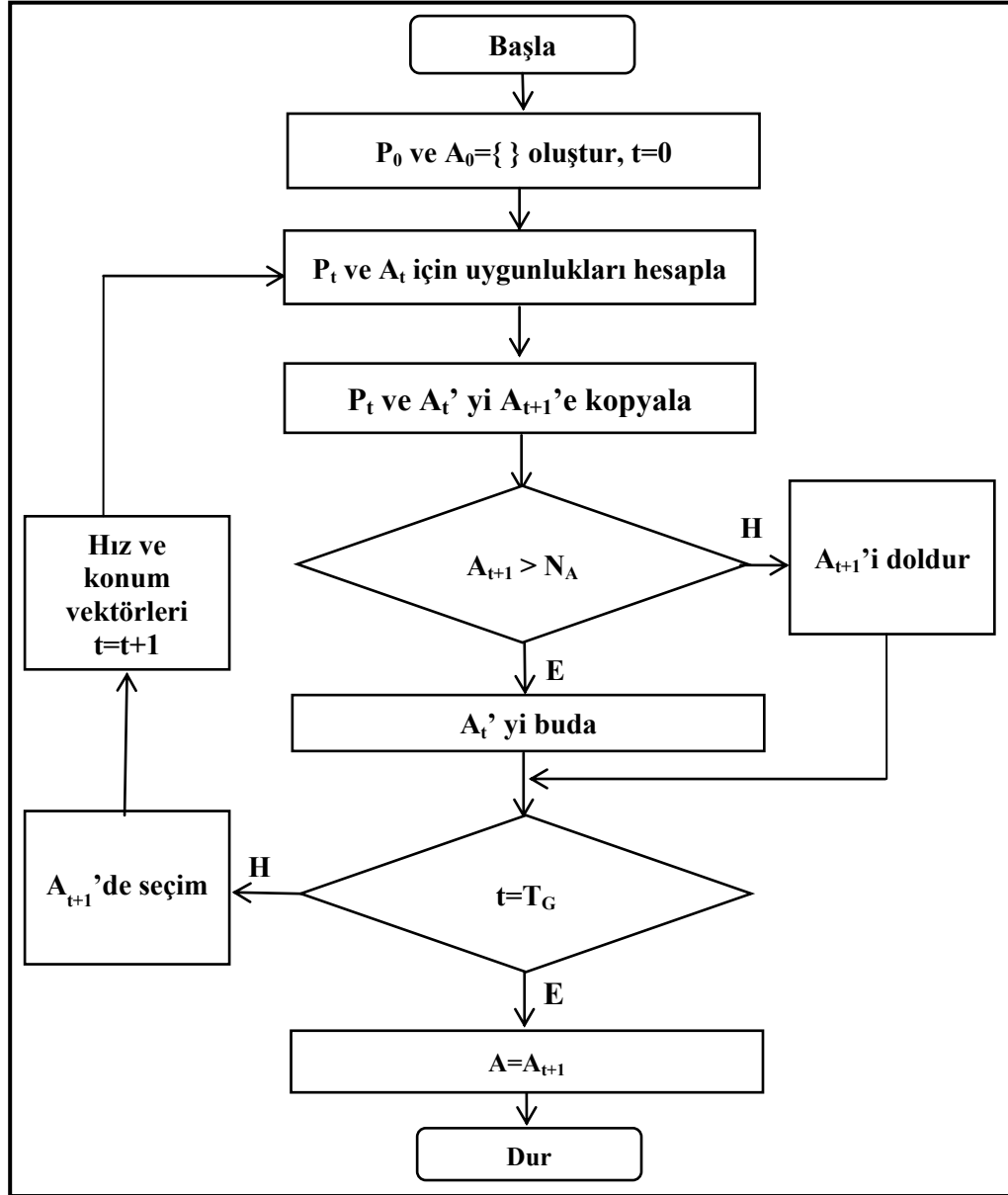
Yazarlar çok amaçlı problemleri genelde ağırlıklandırarak tek amaçlı problem şeklinde çözümler elde etmişlerdir. Bu durumda karar verici için alternatif çözümler olmamaktadır. Tek bir amaç haline getirilerek elde edilen optimal sonuç gerçek yaşam problemlerinde karar verici için gerçekçi veya uygulanabilir sonuç içermeyebilir. Bu nedenle karar verici için alternatif çözümlerin yer aldığı çözümler kümesi daha çok tercih edilir. Yine literatürde çok amaçlı problemlerin çözümünde

karar vericiye alternatif çözümler üreten sonuçların alındığı SPEA2 algoritması sıkça kullanılmaktadır.

Çok amaçlı optimizasyon için iki yaklaşım bulunmaktadır: klasik metotlar ve evrimsel metotlar. Klasik metotlar ayrı olan amaç fonksiyonlarını tek bir fonksiyona çevirirler. Örnek olarak ağırlıklandırılmış toplam metodu, ağırlıklandırılmış metrik metotlar, değer fonksiyon metodu ve hedef programlama metotları verilebilir. Bu metotların özgün problemi yetersiz bir şekilde modelleme, tek bir çözüm üretme ve tasarımcıya verilen seçenekleri sınırlandırma gibi dezavantajları vardır. Ancak, son birkaç yılda klasik olmayan, alışılmışın dışında arama algoritmaları ortaya çıkmıştır. Yakın zamanlarda yapılan çalışmalar, SPEA2'nin çok amaçlı optimizasyon problemlerinin çözülmesinde en etkili yöntem olduğunu ve genellikle diğerlerine üstünlük sağladığını göstermiştir [157].

Çok amaçlı EATÇ problemlerinin BÜÇAPSO ve HİÇAPSO algoritmalarıyla çözümünde pareto optimal çözümleri elde edebilmek için PSO algoritmasına uyarlanan SPEA2 algoritması kullanılmıştır. Çözüm sürecinde elde edilen pareto optimal sonuçlar için kullanılan SPEA2 algoritma akış diyagramı Şekil 4.10' da verildiği gibidir. Algoritma akış diyagramında N , popülasyon boyutunu, $\bar{N}$ harici kümenin maksimum boyutunu, T_G maksimum jenerasyon sayısını ve A baskın kümeyi ifade etmektedir.

Örnek problem üzerinde rastsal oluşturulan X_p sıralama vektörü ve EKİZ kuralına göre oluşturulan X_m atama vektörüne göre amaç fonksiyon değerleri, $C_{\max}=8$, $W_t=16$ ve $W_m=6$ şeklinde elde edilmişti. YA Algoritması 1. alternatif uygulamasına göre elde edilen amaç fonksiyon değerleri, $C_{\max}=7$, $W_t=17$ ve $W_m=6$ şeklinde gerçekleşir. YA Algoritması 2. alternatif uygulamasına göre elde edilen amaç fonksiyon değerleri ise, $C_{\max}=7$, $W_t=18$ ve $W_m=5$ şeklinde olur. Çok amaçlı optimizasyon problemlerinde amaçların ağırlıklandırılarak tek amaç fonksiyonu haline getirilmesi durumunda örnek problemde elde edilen 3 sonuç yerine tek bir sonuç söz konusu olurdu. Çok amaçlı problemlerin çözümünde kullanılan özellikle pareto optimal sonuçları veren algoritmalar örnek problemde de görüldüğü gibi karar verici tarafından tercih edilebilecek 3 alternatif sonuç vermiştir.



Şekil 4.10. Pareto optimal akış diyagramı

4.1.6. Teslim tarihi belirleme

Günümüz rekabet şartlarında işletmelerin, siparişleri en kısa zamanda tamamlama, minimum stokla çalışma ve doğru termin tarihleri belirlemek öncelikli hedefleri arasında yer almaktadır. Bunu gerçekleştirmek için de işletmelerin üretim seviyesinde iyi bir çizelgeleme ve teslim tarihi yönetimi ile mümkündür. Pratik hayatta işletmelerin en önemli hedefleri arasında bulunan bu amaçlar, son yıllarda çizelgeleme alanında çalışan araştırmacıların ilgisinin yoğunlaşmasına da sebep olmuştur.

Teslim tarihleri ya müşteri tarafından veya işletme tarafından belirlenebilir. Teslim tarihleri müşteri tarafından belirlendiğinde, işletme bu tarihlere uyabilmek için iş önceliklerini ve iş akışını belirlemek üzerine çalışır. Teslim tarihi, işletmeler tarafından belirlendiğinde ise genellikle atölyenin yoğunluğu, üretim sisteminin kapasitesi ve işin içeriği göz önünde bulundurulur. Teslim tarihi belirleme ile ilgili yapılan çalışmalarda birçok sezgisel, statik teslim tarihi belirleme kuralı geliştirilmiştir. Literatürde yer alan teslim tarihi belirleme kuralları; sistem içi ve sistem dışı olmak üzere iki farklı grupta incelenmiştir [158].

Sistem dışı teslim tarihi belirleme kuralları[89]:

d_i : i. işin teslim tarihi

k: Akış payı

e_i : i. iş için akış payı

Sabit akış payı (CON) : $d_i = r_i + k$ (4.6)

Her iş için rasgele akış payı (RAN) $d_i = r_i + e_i$ (4.7)

Bu iki yöntem işin özelliklerini, sistem içerisindeki diğer işleri, atölyenin durumunu göz ardı eder. Bundan dolayı teslim tarihi sapmaları genellikle yüksektir.

Sistem içi teslim tarihi belirleme kurallarına göre akış zamanları, işlem zamanları ve operasyon sayıları ile orantılıdır.

p_i : İşin toplam işlem zamanı

n_i : İşin operasyon sayısı

r_i : İşin sisteme geliş zamanı

Toplam iş kapsamı (TWK) $d_i = r_i + k \times p_i$ (4.8)

İşlem zamanlarına eşit boşluk (SLK) $d_i = r_i + p_i + k$ (4.9)

İşlem sayısı (NOP) $d_i = r_i + k \times n_i$ (4.10)

$$\text{İşlem zamanı} + \text{operasyon sayısı (PPW)} \quad d_i = r_i + p_i + k \times n_i \quad (4.11)$$

Teslim tarihinin belirlenmesi ile ilgili literatürde yapılmış çalışmalara bakıldığında, Ragatz ve Mabert'in [159] farklı teslim tarihi belirleme kurallarını doğrusal ve doğrusal olmayan regresyon analizi ile belirleyerek karşılaştırdıkları çalışmaları vardır. Chang [160] teslim tarihlerini ağırlıklandırarak YSA ile çözmüştür. Yazar, regresyon analizi ile elde edilen teslim tarihi değerlerine göre, daha uygun YSA ile elde edilen teslim tarihi değerlerini elde etmiştir. Baykasoğlu ve diğ. [161], EATÇ problemlerinde teslim tarihlerinin belirlenmesine ilişkin TA tabanlı yeni bir yaklaşım geliştirmişlerdir. Yazarlar, teslim tarihlerini önceden belirlemeden, karar değişkeni olarak ele almışlardır. Amaç fonksiyonu olarak, toplam erken tamamlanma ve geç tamamlanma olarak düşünmüşlerdir. TA yöntemi ile EATÇ sistemi için uygun teslim tarihleri elde etmişlerdir. Ayrıca, esneklik düzeyinin teslim tarihi üzerindeki etkilerini ortaya koymuşlardır. Baykasoğlu ve Özbakır [158] uygun teslim tarihi kurallarının belirlenmesi amacıyla, atölye durumunu ve işlerin özelliklerini göz önünde bulundurarak, her bir çizelgeleme problemi için, performansı optimize edecek şekilde genetik programlamayı kullanmışlardır. Yazarlar, genetik programlama ile elde edilen teslim tarihi belirleme kurallarını, literatürde yer alan klasik teslim tarihi belirleme kuralları ile performansa etkileri açısından karşılaştırarak önerdikleri yöntemlerin etkinliğini ortaya koymuşlardır. Özbakır [89], bir çizelgeleme sistemi için doğru teslim tarihlerinin belirlenmesine ilişkin yeni bir yaklaşım olarak genetik programlamanın doğrusal gösterim yöntemlerinden biri olan çoklu denklem programlamayı kullanmıştır. Teslim tarihi belirlemeye ilişkin elde edilen sonuçları, YSA ve regresyon analizi yöntemlerinden elde edilen sonuçlar ile karşılaştırmıştır. Yazar, genetik programlama yaklaşımının, regresyon analizinden ve YSA'dan daha iyi performans sergilediği sonucuna varmıştır.

Literatürde çok amaçlı EATÇ ile ilgili kısıtlı sayıda test problemleri vardır. Bu tez çalışmasında ele alınan test problemlerinde her bir iş operasyonlardan oluşmuş ve bu operasyonların işlenebilmesi için makine alternatifleri vardır. Her bir operasyonun, her bir makinede işlem süreleri mevcuttur. Ancak tez çalışmasında, bu test problemleri için ilk defa ele alınan amaç fonksiyonlarının hesaplanması için teslim süreleri mevcut değildir. Bu test problemlerine ait teslim süreleri belirlenirken literatürde Vilcot ve Billaut [140] ile Gholami ve Zandieh [82] çalışmalarında yer

alan teslim tarihi belirleme yöntemleri kullanılmıştır. Denklem (4.12)'de verilen formülasyonda her bir işin teslim zamanının hangi aralıkta olacağını, Denklem (4.13) ve (4.14) teslim tarihlerinin belirlenmesi için verilen formülasyonların ayrıntılarını vermektedir.

d_i : teslim zamanı

n_i : J_i işinin operasyon sayısı

$P_{i,j,k}$: $O_{i,j}$ operasyonunun m_k makinesindeki işlem zamanı

$n(F(O_{i,j}))$: $O_{i,j}$ operasyonunu işleyebilecek makine sayısı

$\bar{P}_{i,j}$: $O_{i,j}$ operasyonunun ortalama işlem zamanı

$T = 0,3$ (sabit parametre)

$R = 0,5$ (sabit parametre)

$$d_i \sim U \left[\mu_i x \left(1 - \frac{R}{2} \right) : \mu_i x \left(1 + \frac{R}{2} \right) \right] \quad (4.12)$$

$$\mu_i = \left(1 + \frac{T x n}{m} \right) x \sum_{j=1}^{n_i} \bar{P}_{i,j} \quad (4.13)$$

$$\bar{P}_{i,j} = \frac{\sum_{k=1}^{n(F(O_{i,j}))} P_{i,j,k}}{n(F(O_{i,j}))} \quad (4.14)$$

3 İş, 4 makine ve 8 operasyondan oluşan toplam esnekliğe sahip örnek probleme ait işlem süreleri Tablo 4.1' de verilmişti. Denklem (4.12), (4.13) ve (4.14)' de verilen formülasyonlara göre örnek probleme ait işlerin hesaplanan teslim tarihleri Tablo 4.2' de verilmiştir. Ortalama işlem süresi, her bir işe ait tüm operasyonların her bir makinede işlem süreleri göz önünde bulundurularak hesaplanmıştır. μ_i Değeri Denklem (4.13)'e göre hesaplanırken $\sum_{j=1}^{n_i} \bar{P}_{i,j}$ ifadesinde, ortalama işlem sürelerinin toplamını almak teslim tarihi değerlerini oldukça yüksek değerler çıkartmıştır. Literatürdeki problem verilerine uygun olarak teslim tarihleri elde edebilmek ve problem setlerinin zorluk derecesini biraz daha arttırabilmek için $\sum_{j=1}^{n_i} \bar{P}_{i,j}$ ifadesi yerine $\bar{P}_{i,j}$ değerleri alınmıştır. Her bir işe ait teslim tarihi, d_1 alt sınırı ile d_2 üst

sınırlarının ortalaması olarak alınmıştır. Literatürde yer alan problem setlerine ait işlem süreleri ve hesaplanan teslim tarihleri EK-A kısmında verilmiştir.

Tablo 4.2. Örnek probleme ait hesaplanmış teslim tarihleri

		M1	M2	M3	M4	Ortalama İşlem Süresi ($\bar{P}_{ij}$)	μ	d_1	d_2	d
J_1	O_{11}	4	7	3	2	4,17	5,17	4	6	5
	O_{12}	3	1	3	5					
	O_{13}	4	5	6	7					
J_2	O_{21}	1	3	4	6	3,33	4,13	3	5	4
	O_{22}	4	5	3	2					
	O_{23}	6	2	1	3					
J_3	O_{31}	8	3	4	5	4,88	6,05	5	8	6
	O_{32}	4	7	6	2					

4.2. Deneysel Sonuçlar

Çok amaçlı EATÇ problemlerinin çözümü için literatürde kullanılan test problemleri Tablo 4.3.' de verilmiştir. Problem gösterimi, probleme ait iş sayısı, makine sayısı ve operasyon sayısı şeklinde verilmiştir. Problem türü, ilgili problemin toplam esnekliğe veya kısmi esnekliğe sahip olduğunu, problem boyutu ise küçük, orta ve büyük boyut olmak üzere problemin hangi boyutta olduğunu göstermektedir. Tüm operasyonların tüm makineler üzerinde işlenebilir olduğu durumlar toplam EATÇ, operasyonların tüm makineler yerine belirli bir makine alt grubunda işlenebilir olduğu durumlar ise kısmi EATÇ problemleri olarak tanımlanmaktadır. Ayrıca literatür incelemelerinde araştırmacıların tek amaçlı (C_{max}) EATÇ problemlerinin çözümünde, özellikle Brandimarte [2] (BRdata) 10 adet problem seti olmak üzere, Barnes ve Chambers [146] (BC data) 21 adet problem, Dauzere Peres ve Paulli [60] (DPdata) 18 adet

problem ve Hurink ve diğ. [59] (HUdata) 128 adet problem setlerini kullandıkları görülmüştür.

Tablo 4.3. Kullanılan problem boyutları

Sıra No	Problem Gösterimi (nmxO)	Problem Türü	Problem Boyutu
1	4x5x12	Toplam esnek	Küçük
2	8x8x27	Kısmi esnek	Orta
3	10x7x29	Toplam esnek	Orta
4	10x10x30	Toplam esnek	Orta
5	15x10x56	Toplam esnek	Büyük

4.2.1. PSO’ da kullanılan parametrelerin belirlenmesi

Diğer optimizasyon yöntemlerine göre PSO yönteminde, parametre sayısının az olması, optimizasyon problemlerine kolay uygulanabiliyor olması yöntemin avantajları olarak söylenebilir. Ayrıca PSO yönteminde parçacıklar, hem kendi en iyi pozisyon değerlerini hem de popülasyondaki diğer komşuların en iyi pozisyon değerlerini elde ettüklerinden, iyi bir hafıza yeteneğine sahiptirler. PSO’da çözüm uzayında arama yapılırken en iyi pozisyona sahip parçacığın değerinden yararlanılır ve arama uzayında herhangi bir değişiklik olmaz. Arama uzayında herhangi bir değişikliğe gitmemek bu yöntemin yerel optimumlara takılmasını engeller.

PSO Algoritmasında yeni konum vektörleri bulunurken kullanılan atalet ağırlığı (w) parametresi, bir önceki iterasyondan hız ile ilişkilendirilmiş ölçekleme faktörü olarak tanımlanmaktadır. Klasik PSO’da hız güncelleme denklemi $w=1$ seçilerek elde edilebilir. Shi ve Eberhart [113], w değerinin $[0,1:4]$ aralığındaki etkilerini ve her iterasyonla değişen değerlerini incelemişlerdir. Yazarlar, $w \in [0,8:1,2]$ olarak seçmenin daha hızlı yakınsamayla sonuçlandığını, ancak $w>1,2$ seçmenin daha fazla hata ile yakınsadığını göstermişlerdir.

Atalet ağırlığı $w=0$ alınması durumunda, parçacıkların hareketi, sadece mevcut konumu ile parçacığın o ana kadar en iyi konumu ve sürüdeki en iyi konum değeri bilgileri ışığında olacaktır. Bu durumda global en iyi konumda olan parçacık, kendisinden daha iyi bir değere sahip parçacık çıkana kadar sürekli yerinde

kalacaktır. Algoritmada çözüm uzayının iterasyonlar boyunca küçüldüğü bir yapı haline gelecektir. Parçacıklar P_{best} ile G_{best} arasında kalma eğiliminde olacaktır. Bu şekilde etkin bir aramanın olmasından bahsedilemez. Atalet ağırlığı w 'nın etkisini görebilmek için $c_1=c_2=0$ olarak seçildiğini ve başlangıç hız vektörü değerinin ($V_i^0 \neq 0$) sıfır olmadığı varsayıldığında, $w>1$ değeri parçacığın değişmeyip olduğu gibi kalacağı maksimum hıza ivmelenmesine neden olacaktır. $w<1$ değeri, parçacığın hızı sıfıra ulaşmaya kadar yavaşlamasına neden olacaktır. $c_1, c_2 \neq 0$ olduğunda algoritmanın davranışını tahmin etmek daha zor olur. Shi ve Eberhart [113] çalışmalarındaki sonuçlar göz önünde bulundurulduğunda w değerlerinin 1'e yakın değerleri tercih edilir.

PSO'da w algoritmanın global ve yerel arama yeteneğini dengelemek için kullanılırlar. Atalet ağırlığının büyük bir değer seçilmesi global arama, küçük bir değer seçilmesi ise yerel arama yapılmasını kolaylaştırır. Atalet ağırlığı hesaplanırken özellikle kullanılan W_{max} ve W_{min} değerlerinin doğru seçilmesi yerel ve global arama arasındaki dengeyi sağlarken, algoritmanın daha az iterasyonla iyi sonuçlara ulaşmasını sağlar.

PSO'da algoritmanın performansını etkileyen en önemli parametrelerden iki tanesi de, parçacık tecrübesi (c_1) ve sürü tecrübesi (c_2) parametreleridir. c_1 Parametresi, parçacığın mevcut bulunduğu iterasyona kadar elde ettiği en iyi sonuçların bir sonraki iterasyona etkisi olarak tanımlanmaktadır. c_2 Parametresi ise, sürünün mevcut bulunduğu iterasyona kadar elde ettiği en iyi değerlerin bir sonraki iterasyona etkisi olarak tanımlanmaktadır.

c_1 ve c_2 parametrelerinin büyük değerler seçilmesi, parçacıkların birbirlerinden uzaklaşıp ayrılmasına sebep olurken, küçük değerler seçilmesi ise parçacık hareketlerinin kısıtlanmasına ve çözüm uzayının yeterince taranamamasına neden olur. Bu parametre değerleri problem türüne göre değişiklik göstermesine rağmen genelde yazarların $c_1=c_2=2$ olarak aldıkları görülmektedir. Ancak yazarların c_1 ve c_2 parametrelerinin birbirlerinden farklı değerlerini aldıkları da görülmektedir. $c=c_1+c_2$ durumunda c ne kadar büyük değer alırsa, algoritmanın optimum değer etrafında yaptığı salınım miktarı artmaktadır.

Çok amaçlı EATÇ problemlerinin, çok amaçlı PSO algoritmasıyla çözümünde kullanılan parametrelerin en iyi değerlerinin tespit edilebilmesi için parametrelere ait faktörler ve düzeyleri Tablo 4.4’ de verilmiştir.

Tablo 4.4. Parametrelere ait faktör ve düzeyleri

Faktörler	Faktör Açıklaması	Düzy Sayısı	Düzeyler
W_{max}	Atalet ağırlığı başlangıç değeri	21	0’dan 2’ ye kadar
W_{min}	Atalet ağırlığı bitiş değeri	21	0’dan 2’ ye kadar
c_1	Öğrenme Faktörü Parçacık Tecrübesi	21	0’dan 2’ ye kadar
c_2	Öğrenme Faktörü Sürü tecrübesi	21	0’dan 2’ ye kadar

Çok amaçlı EATÇ problemlerinin çözümünde geliştirilen her iki algoritma (HİÇAPSO ve BÜÇAPSO) için uygun parametreler tespit edilmiştir. Bu parametre değerleri Tablo 4.3’ de verilen 5 adet problem için ayrı ayrı belirlenmiştir. Her bir çözüm algoritması için her bir probleme ait parametreler belirlenirken, faktörlerden üçü sabit tutularak bir faktör için [0,2] aralığında algoritma çalıştırılmıştır. Her bir algoritma için amaç fonksiyonu aşağıda verilen 4 kriter esas alınmıştır.

- C_{max}
- W_t
- W_m
- Toplam = $C_{max}+W_t+W_m$

Geliştirilen algoritma çok amaçlı olduğu için her bir parametrenin alabileceği her bir değer için, her bir amaç fonksiyonu bazında her problem 3 defa çalıştırılmıştır. Probleme ait optimal sonuçlar olmadığı için her bir çalıştırmada, 4 amaç fonksiyonu bazında elde edilen en iyi değerler belirlenmiştir. Algoritmada parçacık sayısı =50, iterasyon sayısı = 100 olarak sabitlenmiştir.

Parametre belirleme adımları BÜÇAPSO algoritmasında 4x5x12 problemi için ayrıntılı verilmiştir. Öncelikle her bir parametreye ait bir değer belirlenirken 4 adet faktörden 3’ü sabit tutularak 1 faktör için [0,2] aralığında her 0,1 birimlik artışlarla algoritma her bir amaç fonksiyonu için 3 defa çalıştırılmaktadır. Sabit tutulan 3

faktör için her bir çalıştırmada ise faktörler 0,5 birimlik artışlarla sabitlenmektedir. Sabit tutulan parametrelere ait çalıştırma düzeni Tablo 4.5’ te verildiği gibidir.

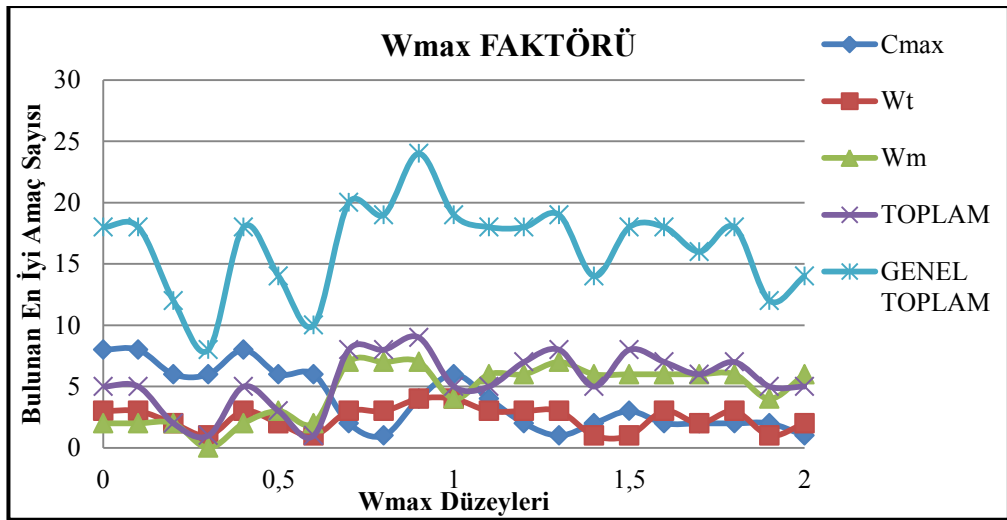
Tablo 4.5. Sabit tutulan parametrelere ait faktör ve düzeyleri

Faktörler	Faktör Açıklaması	Düzyey Sayısı	Düzeyler
W_{max}	Atalet ağırlığı başlangıç değeri	3	0,5 – 1 ve 1,5
W_{min}	Atalet ağırlığı bitiş değeri	3	0,5 – 1 ve 1,5
c_1	Öğrenme Faktörü Parçacık Tecrübesi	3	0,5 – 1 ve 1,5
c_2	Öğrenme Faktörü Sürü tecrübesi	3	0,5 – 1 ve 1,5

4x5x12 Problemine ait BÜÇAPSO algoritmasının en iyi W_{max} değeri belirlenirken $W_{min}=c_1=c_2=0,5$ ve $W_{max}=[0,2]$ aralığında 0,10 birimlik artış değerleri için algoritma 50 parçacık ve 100 iterasyonda çalıştırılarak, C_{max} , w_t , w_m ve toplam bazında en iyi değerleri veren düzeyler belirlenmiştir. Daha sonra $W_{min}=c_1=c_2=1,0$ ve $W_{max}=[0,2]$ aralığında 0,1 birimlik artış değerleri için aynı şartlarda algoritma çalıştırılarak aynı şekilde en iyi amaç fonksiyonu sonuçlarının alındığı düzeyler belirlenmiştir. 3. Denemede $W_{min}=c_1=c_2=1,5$ ve $W_{max}=[0,2]$ aralığında 0,1 birimlik artış değerleri için aynı şekilde en iyi amaç fonksiyonu değerlerine sahip düzeyler tespit edilmiştir. Buraya kadar çalıştırılan algoritma çok amaçlı algoritmamız için öncelikle C_{max} amaç fonksiyonu bazında çalıştırıldıktan sonra, W_t , W_m ve toplam amaç fonksiyonları bazında aynı şartlarda çalıştırılarak en iyi değerleri veren düzeyler tespit edilmektedir. Bu durumda W_{max} parametresinin (faktör) belirlenebilmesi için algoritma, 4 amaç fonksiyonu x 3deneme x 21 düzey = 252 defa çalıştırılmaktadır. Tablo 4.6 BÜÇAPSO algoritması W_{max} faktörü için her bir düzeye ait elde edilen en iyi amaç fonksiyon değeri sayılarını göstermektedir. Şekil 4.11’ de 4x5x12 problemi için W_{max} parametresinin 4 amaç fonksiyonu bazında en iyi değerleri bulma sayısının grafiksel gösterimidir. Tablo 4.6.’ da verilen değerlerin grafik olarak gösterimi Şekil 4.11’ de verilmiştir.

Tablo 4.6. 4x5x12 Problemi W_{max} faktörü için deney tasarımı sonuçları

DÜZEYLER	BULUNAN EN İYİ AMAÇ FONKSİYON SAYILARI				
W_{max}	C_{max}	W_t	W_m	TOPLAM	GENEL TOPLAM
2	1	2	6	5	14
1,9	2	1	4	5	12
1,8	2	3	6	7	18
1,7	2	2	6	6	16
1,6	2	3	6	7	18
1,5	3	1	6	8	18
1,4	2	1	6	5	14
1,3	1	3	7	8	19
1,2	2	3	6	7	18
1,1	4	3	6	5	18
1	6	4	4	5	19
0,9	4	4	7	9	24
0,8	1	3	7	8	19
0,7	2	3	7	8	20
0,6	6	1	2	1	10
0,5	6	2	3	3	14
0,4	8	3	2	5	18
0,3	6	1	0	1	8
0,2	6	2	2	2	12
0,1	8	3	2	5	18
0	8	3	2	5	18

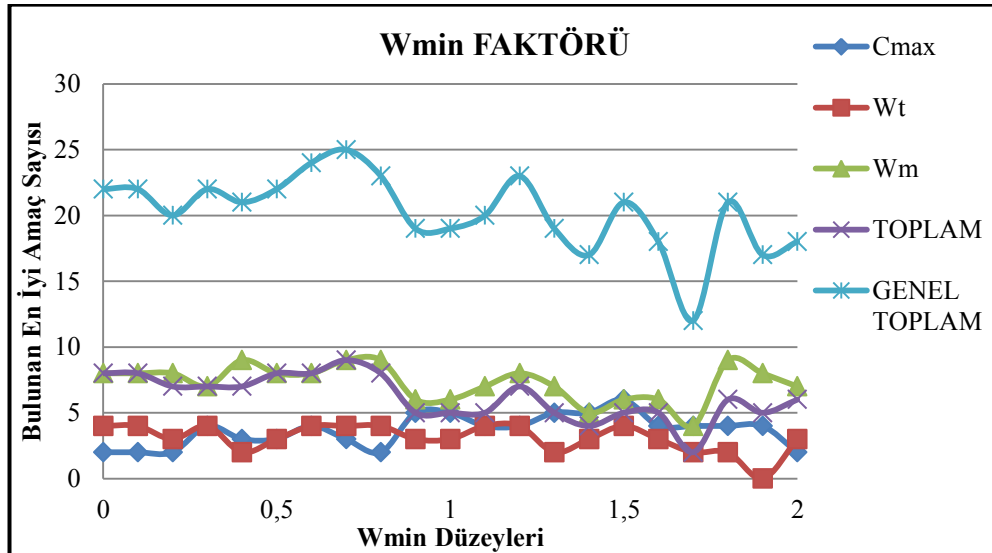


Şekil 4.11. W_{max} Faktörü deney tasarım sonuçlarının grafiksel gösterimi

Tablo 4.6’ da verilen değerlere göre, yapılan 252 deney içerisinde $W_{\max}=0,9$ değeri için 4 defa $C_{\max}$ amaç fonksiyonunda, 4 defa W_t , 7 defa W_m ve 9 defa toplam amaç fonksiyonlarında olmak üzere genel toplamda 24 defa en iyi değeri verdiği için BÜÇAPSO algoritmasına ait $4 \times 5 \times 12$ probleminin çözümü için $W_{\max}=0,9$ olarak belirlenmiştir. Buraya kadar uygulanan adımlar aynı şekilde $W_{\min}$, c_1 ve c_2 faktörleri için gerçekleştirilmektedir.

$W_{\min}$ faktörü en iyi parametre değerinin belirlenmesi için bir önceki adımda elde edilen $W_{\max}=0,9$ değeri sabitlenerek, $c_1=c_2=0,5$, $c_1=c_2=1,0$ ve $c_1=c_2=1,5$ düzeylerinde toplam 4 tane amaç fonksiyonu bazında $W_{\min}$ değeri için $[0,2]$ aralığında 0,1 birimlik artışlarla algoritma tekrar çalıştırılmaktadır. Bu adımda da $W_{\min}$ parametresinin belirlenmesi için algoritma 252 defa çalışmaktadır. Tablo 4.7 BÜÇAPSO algoritması $W_{\min}$ faktörü için her bir düzeye ait elde edilen en iyi amaç fonksiyon değeri sayılarını göstermektedir.

Tablo 4.7’ de elde edilen verilere göre, 252 deney içerisinde $W_{\min}=0,7$ değeri için algoritma en çok sayıda amaç fonksiyonu değerini bulduğundan dolayı, BÜÇAPSO algoritmasına ait $4 \times 5 \times 12$ probleminin çözümü için $W_{\min}=0,7$ olarak belirlenmiştir. Şekil 4.12’ de Tablo 4.7’ de verilen değerlerin grafiksel gösterimi sunulmuştur.



Şekil 4.12 $W_{\min}$ Faktörü deney tasarımı sonuçlarının grafiksel gösterimi

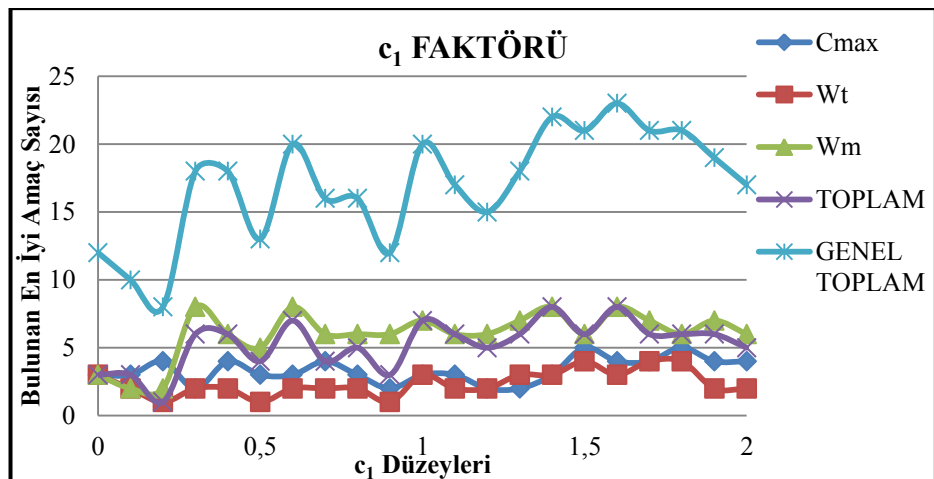
Tablo 4.7. 4x5x12 Problemi W_{min} faktörü deney tasarımı sonuçları

DÜZEYLER	BULUNAN EN İYİ AMAÇ FONKSİYON SAYILARI				
W_{min}	C_{max}	W_t	W_m	TOPLAM	GENEL TOPLAM
2	2	3	7	6	18
1,9	4	0	8	5	17
1,8	4	2	9	6	21
1,7	4	2	4	2	12
1,6	4	3	6	5	18
1,5	6	4	6	5	21
1,4	5	3	5	4	17
1,3	5	2	7	5	19
1,2	4	4	8	7	23
1,1	4	4	7	5	20
1	5	3	6	5	19
0,9	5	3	6	5	19
0,8	2	4	9	8	23
0,7	3	4	9	9	25
0,6	4	4	8	8	24
0,5	3	3	8	8	22
0,4	3	2	9	7	21
0,3	4	4	7	7	22
0,2	2	3	8	7	20
0,1	2	4	8	8	22
0	2	4	8	8	22

c_1 Faktörüne ait en iyi parametre değerinin belirlenebilmesi için bu adıma kadar elde edilen $W_{max}=0,9$ ve $W_{min}=0,7$ değerleri sabitlenerek, c_2 faktörüne ait 3 düzey ve c_1 faktörü için 21 düzeye göre algoritma 4 amaç fonksiyonu bazında aynı şekilde toplamda 252 deney yapılır. Tablo 4.8’ de elde edilen verilere göre, c_1 parametre değeri BÜÇAPSO algoritmasına ait 4x5x12 probleminin çözümü için $c_1=1,6$ olarak belirlenmiştir. Şekil 4.13 BÜÇAPSO algoritmasına ait 4x5x12 probleminin deney tasarımı çözümünde elde edilen c_1 faktörü sonuçlarının grafiksel gösterimidir. Şekil 4.13’ de görüldüğü gibi c_1 faktörü [1,4:1,8] aralığında diğer düzeylere göre amaç fonksiyonu bazında daha fazla sayıda iyi sonuç bulmuştur.

Tablo 4.8. 4x5x12 problemi c_l faktörü deney tasarımı sonuçları

DÜZEYLER	BULUNAN EN İYİ AMAÇ FONKSİYON SAYILARI				
c_l	C_{max}	W_t	W_m	TOPLAM	GENEL TOPLAM
2	4	2	6	5	17
1,9	4	2	7	6	19
1,8	5	4	6	6	21
1,7	4	4	7	6	21
1,6	4	3	8	8	23
1,5	5	4	6	6	21
1,4	3	3	8	8	22
1,3	2	3	7	6	18
1,2	2	2	6	5	15
1,1	3	2	6	6	17
1	3	3	7	7	20
0,9	2	1	6	3	12
0,8	3	2	6	5	16
0,7	4	2	6	4	16
0,6	3	2	8	7	20
0,5	3	1	5	4	13
0,4	4	2	6	6	18
0,3	2	2	8	6	18
0,2	4	1	2	1	8
0,1	3	2	2	3	10
0	3	3	3	3	12

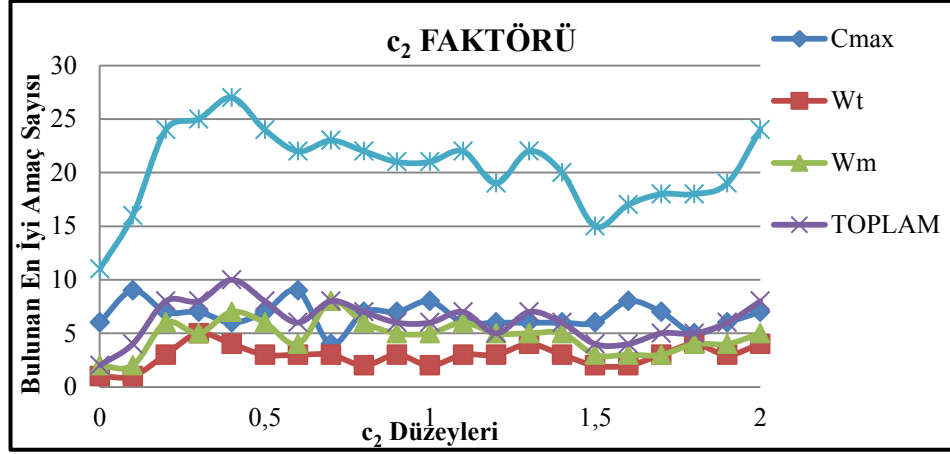


Şekil 4.13. c_1 Faktörü deney tasarımı sonuçlarının grafiksel gösterimi

Bu adıma kadar $W_{\max}=0,9$ ve $W_{\min}=0,7$ ve $c_1= 1,6$ şeklinde belirlenmiştir. Tespit edilen bu parametre değerlerine göre algoritma her bir amaç fonksiyonuna 3 deneme yapılmak üzere c_2 parametresinin $[0,2]$ aralığındaki değerleri için toplamda 4 amaç fonksiyonuna göre 252 deney tekrar yapılmıştır. Bu deneyler neticesinde c_2 parametresine ait sonuçlar Tablo 4.9’ da verilmiştir. Tablo 4.9’ da görüldüğü üzere c_2 faktörüne ait parametre değeri $c_2=0,4$ şeklinde belirlenmiştir. Şekil 4.14’ de $4 \times 5 \times 12$ probleminin BÜÇAPSO algoritması ile çözümünde kullanılacak c_2 faktörü için elde edilen deney tasarımı sonuçlarının grafiksel gösterimi verilmiştir. Şekil 4.14’de görüldüğü gibi c_2 faktörü $[0,2:0,5]$ aralığındaki düzeyler için daha fazla sayıda iyi sonuç bulurken $[0,6:2,0]$ aralığındaki düzeyler için gittikçe daha az sayıda iyi sonuç bulduğu görülmektedir.

Tablo 4.9. $4 \times 5 \times 12$ problemi c_2 faktörü deney tasarımı sonuçları

DÜZEYLER	BULUNAN EN İYİ AMAÇ FONKSİYON SAYILARI				
c_2	$C_{\max}$	W_t	W_m	TOPLAM	GENEL TOPLAM
2	7	4	5	8	24
1,9	6	3	4	6	19
1,8	5	4	4	5	18
1,7	7	3	3	5	18
1,6	8	2	3	4	17
1,5	6	2	3	4	15
1,4	6	3	5	6	20
1,3	6	4	5	7	22
1,2	6	3	5	5	19
1,1	6	3	6	7	22
1	8	2	5	6	21
0,9	7	3	5	6	21
0,8	7	2	6	7	22
0,7	4	3	8	8	23
0,6	9	3	4	6	22
0,5	7	3	6	8	24
0,4	6	4	7	10	27
0,3	7	5	5	8	25
0,2	7	3	6	8	24
0,1	9	1	2	4	16
0	6	1	2	2	11



Şekil 4.14. c_2 Faktörü deney tasarımı sonuçlarının grafiksel gösterimi

Çok amaçlı EATÇ problemlerinin çözümü için geliştirilen çok amaçlı BÜÇAPSO ve HİÇAPSO algoritmalarına ait PSO parametre değerleri Tablo 4.10’da toplu halde verilmiştir.

Tablo 4.10. Belirlenen PSO parametre değerleri

		PROBLEM SETLERİ				
		4x5x12	8x8x27	10x7x29	10x10x30	15x10x56
BÜÇAPSO	Parçacık Sayısı	50	50	50	50	50
	İterasyon Sayısı	50	50	50	50	50
	W_{max}	0,9	0,3	1,5	1,1	0,9
	W_{min}	0,7	1,8	2,0	1,1	1,3
	c_1	1,6	1,6	1,1	0,9	0,5
	c_2	0,4	1,0	1,2	1,6	0,3
HİÇAPSO	Parçacık Sayısı	50	50	50	50	50
	İterasyon Sayısı	50	50	50	50	50
	W_{max}	1,4	0,1	1,6	1,7	1,9
	W_{min}	1,6	1,3	2,0	2,0	0,3
	c_1	1,5	0,6	1,7	1,7	1,8
	c_2	0,6	1,2	1,5	0,2	2,0

4.2.2. Deneysel sonuçlar

Bu bölümde çok amaçlı EATÇ problemleri için geliştirilen çok amaçlı BÜÇAPSO ve HİÇAPSO algoritmalarının literatürde yer alan problem setleriyle etkinliği ölçülmüştür. Kullanılan problem setleri Tablo 4.3’ te verilmiştir. Test problemleri her bir operasyonun işlenebileceği makine alternatifleri arasında değişik işlem sürelerine sahiptir. Operasyonlar arasında öncelik kısıtı mevcuttur. Ele alınan 5 adet problem setinden 1 adet kısmi esnekliğe sahip 4 adet problem ise toplam esnekliğe sahiptir.

Çizelgeleme problemlerinde $\sum C$ ölçütü ile sipariş çevrim hızı artmakta ve yeni siparişlerin üretime daha erken alınması sağlandığı gibi yarı mamul stoklarında da azalma görülmektedir. Firmalar müşterilerin ihtiyaçlarını zamanında karşılamak veya en az gecikme ile müşterilerin isteklerine cevap vermek isteyebilir. Bu da $\sum T$ ölçütü ile tanımlanır. Çok amaçlı EATÇ problemleri gerçek yaşam problemleri sınıfından olduğundan son yıllarda en çok ilgi çeken konulardan olmuştur. Çok amaçlı çizelgeleme problemleri arasında gecikme ölçütünün dâhil edildiği problemler üzerinde oldukça sınırlı sayıda çalışma yapılmıştır.

Tablo 4.11. Çalışmada ele alınan amaç fonksiyonları

Grup	Amaç Fonksiyonu	Parametre	Formülasyon
Grup 1	Maksimum Tamamlanma Zamanı	C_{max}	$\min C_{max} = \max_i \max_k \{c_{ik}\}$
	Toplam İş Yüğü	W_t	$\min W_T = \sum_{j=1}^m W_j$
	Maksimum İş Yüğü	W_m	$\min W_M = \max_j \{W_j\}$
Grup 2	Maksimum Tamamlanma Zamanı	C_{max}	$\min C_{max} = \max_i \max_k \{c_{ik}\}$
	Toplam Tamamlanma Zamanı	$\sum C$	$\min \sum C = \sum_{i=1}^n C_i$
	Toplam Gecikme	$\sum T$	$\min \sum T = \sum_{i=1}^n T_i$

Çalışmada ele alınan amaç fonksiyonları Tablo 4.11’ de verilmiştir. Problem setleri, öncelikle literatürde en çok kullanılan C_{max} , W_t ve W_m olmak üzere 3 amaçlı olarak çözülmüştür. Elde edilen sonuçlar literatürde yer alan diğer çalışmalarla karşılaştırılmış ve BÜÇAPSO ve HİÇAPSO algoritmalarının etkinliği ölçülmüştür.

Daha sonra çok amaçlı EATÇ problemlerinin çözümü için literatürde eş zamanlı optimize edilmemiş 3 amaç fonksiyonu çözülmüştür. Bu amaç fonksiyonları, C_{max} , $\sum C$, ve $\sum T$ olarak ele alınmıştır. Yapılan literatür taramasında bu üç amaç fonksiyonunun çok amaçlı akış tipi çizelgeleme problemlerinde en çok çalışılan amaçlar olmasından ve EATÇ problemlerinde çalışılmamış olmasından dolayı tercih edilmiştir.

4x5x12 Problemi:

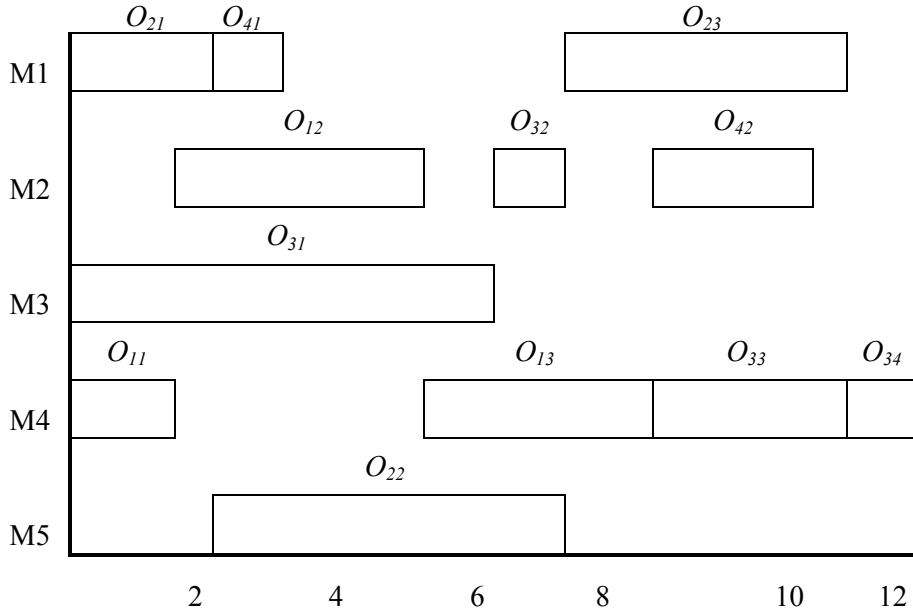
Çalışmada ilk ele alınan problem boyutu 4 iş, 5 makine ve 12 operasyondan oluşan toplam esnekliğe sahip problemidir. Probleme ait işlem süreleri ve teslim zamanları EK-A kısmında verilmiştir. HİÇAPSO ve BÜÇAPSO algoritmaları ile elde edilen sonuçlar Kacem ve diğ.[10] tarafından geliştirilen GA ve yerini belirleme algoritması (AL+CGA), Zhang ve diğ. [121] tarafından geliştirilen PSO ve TA algoritmalarının melezlenmesi ile elde ettikleri (PSO+TS), Wang ve diğ. [73] tarafından geliştirilen çok amaçlı GA (MOGA) ve Li ve diğ. [87] tarafından geliştirilen TA algoritması (HTSA) sonuçlarıyla karşılaştırılmıştır. Sonuçlar Tablo 4.12 'de verilmiştir.

Tablo 4.12. 4x5x12 Problemine ait sonuçlar

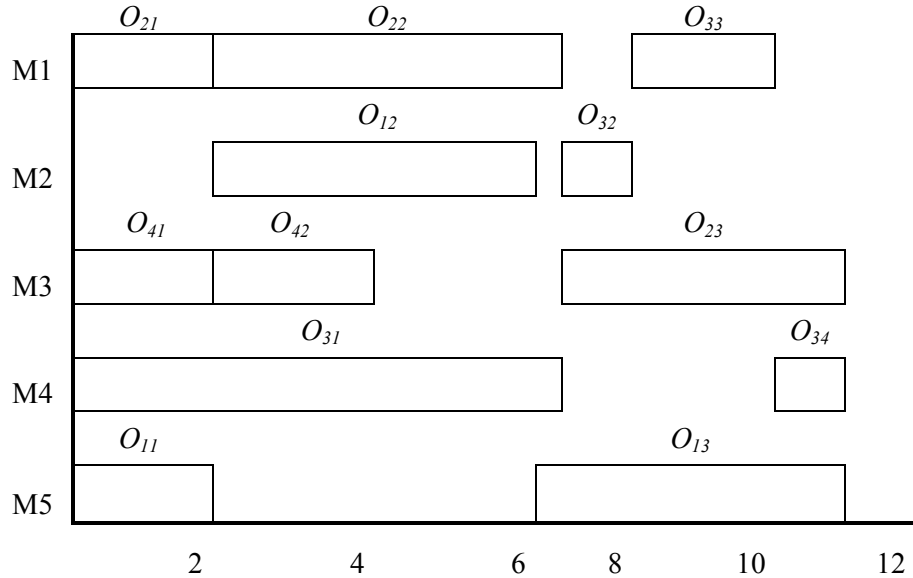
	HİÇAPSO			BÜÇAPSO		AL + CGA	PSO + TS	MOGA			HTSA	
C_{max}	11	12	11	12	11	16	11	11	11	12	11	12
W_t	32	32	37	32	32	34	32	32	34	32	32	32
W_m	10	9	9	8	10	10	10	10	9	8	10	8

4x5x12 Problem sonuçları göz önünde bulundurulduğunda HİÇAPSO ve BÜÇAPSO algoritmalarıyla elde edilen pareto sonuçların Kacem ve diğ. [10] tarafından geliştirilen (AL+CGA) algoritma sonuçlarına göre baskın sonuçlar verdiği görülmektedir. Aynı şekilde HİÇAPSO ve BÜÇAPSO algoritmaları ile Zhang ve diğ. [121] tarafından geliştirilen (PSO+TS) algoritma sonucuna göre daha iyi sonuçlar elde edilmiştir. BÜÇAPSO algoritmasıyla elde edilen sonuçlar Wang ve diğ. [73] tarafından geliştirilen MOGA algoritması ile aynı sonuçları bulmuştur.

BÜÇAPSO ve HİÇAPSO Algoritmaları, Li ve diğ. [87] tarafından geliştirilen HTSA algoritma sonuçları ile aynı sonuçları buldukları, ayrıca HİÇAPSO algoritmasıyla $C_{max}=11$, $W_t=37$ ve $W_m=9$ şeklinde bir baskın çözüm daha bulduğu gözlenmiştir. BÜÇAPSO algoritması ile elde edilen $C_{max}=12$, $W_t=32$ ve $W_m=8$ sonuçlarına ait Gantt şeması Şekil 4.15 'de HİÇAPSO algoritması ile elde edilen $C_{max}=11$, $W_t=37$ ve $W_m=9$ sonuçlarına ait Gantt şeması Şekil 4.16 'da verilmiştir.



Şekil 4.15. 4x5x12 Problemi BÜÇAPSO $C_{max}=12$, $W_t=32$, $W_m=8$ çözümü



Şekil 4.16. 4x5x12 Problemi HİÇAPSO $C_{max}=11$, $W_t=37$, $W_m=9$ çözümü

8x8x27 Problemi:

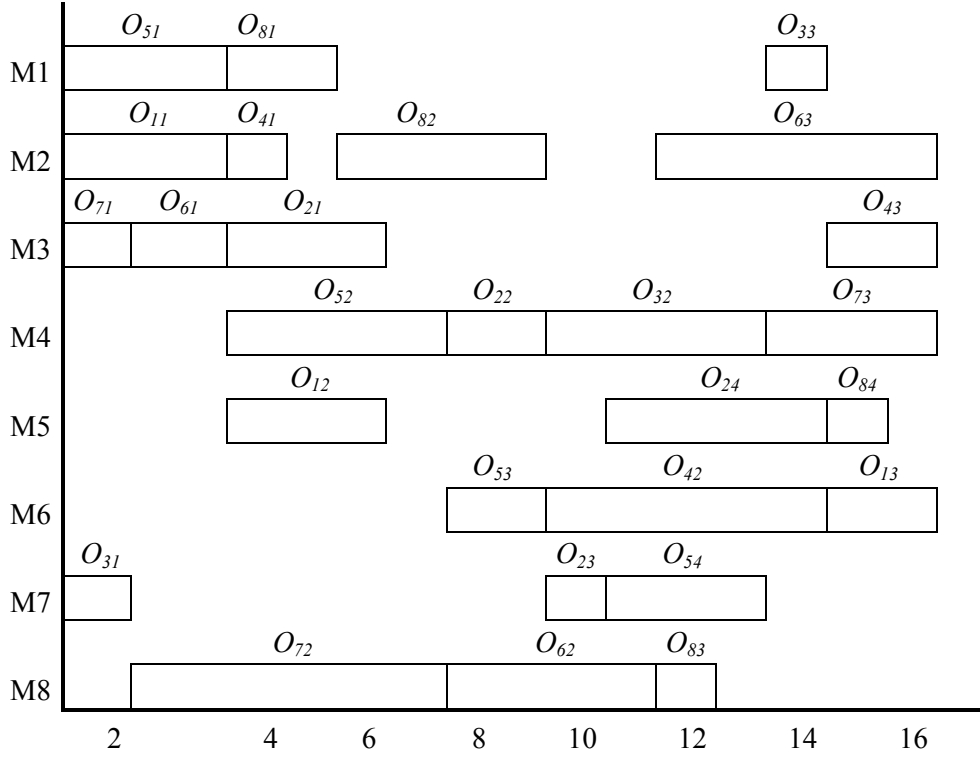
8 İş, 8 makine ve 27 operasyondan oluşan problem türü kısmi esnekliğe sahiptir. Operasyonlar 8 makinenin tümü yerine belirli bir makine alt grubunda işlem görebilmektedir. Her operasyon tüm makine grubunda işlenemediği için her iki çözüm algoritmasının kodlanması sırasında operasyonların işlenemeyeceği makine için işlem süresi 999 alınarak bu operasyonların bu makinelerle atanması engellenmiştir. Bu problem setine ait BÜÇAPSO ve HİÇAPSO sonuçları literatürde, Kacem ve diğ.[10] tarafından geliştirilen (AL+CGA), Zhang ve diğ. [121] tarafından geliştirilen (PSO+TS), Xia ve Wu [28] tarafından geliştirilen (PSO+SA), Bagheri ve diğ. [132] tarafından geliştirilen (AIA), Wang ve diğ. [73] tarafından geliştirilen (MOGA) ve Li ve diğ. [87] tarafından geliştirilen (HTSA) algoritmaları sonuçlarıyla karşılaştırılmıştır. Tablo 4.13 8x8x27 problemine ait sonuçları göstermektedir.

Tablo 4.13. 8x8x27 Problemine ait sonuçlar

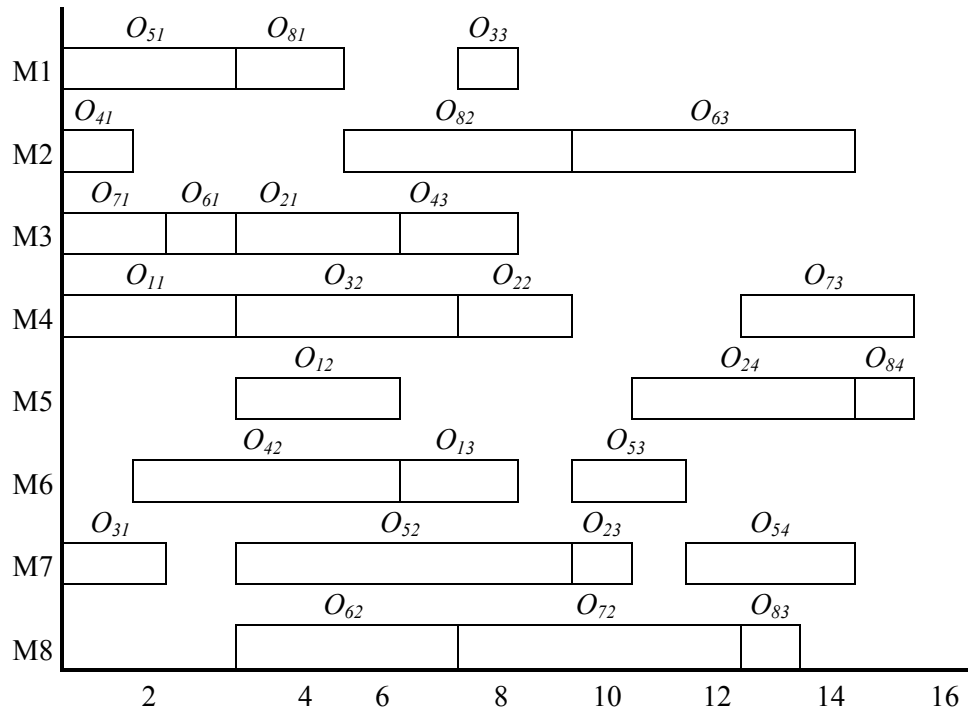
	BÜÇAPSO	HİÇAPSO	AL + CGA	PSO + TS	PSO+SA	AIA	MOGA	HTSA
C_{max}	16 16 15	15 17 14	15 16	14 15	15 16	14	15 15 16	14 15
W_t	73 75 75	75 73 79	79 75	77 75	75 73	77	81 75 73	77 75
W_m	13 12 14	12 13 14	13 13	12 12	12 13	12	11 12 13	12 12

Tablo 4.13’ de verilen sonuçlara göre BÜÇAPSO ve HİÇAPSO algoritmaları (AL+CGA) ve (AIA) algoritma sonuçlarına göre çok daha iyi sonuçlar verdiği görülmektedir. Aynı şekilde BÜÇAPSO algoritması ile $C_{max}=16$, $W_t=73$, $W_m=13$ ve HİÇAPSO algoritması ile $C_{max}=17$, $W_t=73$, $W_m=13$ şeklinde (PSO+TS) ve (HTSA) algoritma sonuçlarına göre birer baskın çözüm bulduğu görülmektedir. Ayrıca BÜÇAPSO ve HİÇAPSO algoritmalarımız (MOGA) algoritmasıyla elde edilen sonuçlarla aynı sonuçlara ulaşmakla beraber HİÇAPSO algoritması $C_{max}=14$, $W_t=79$, $W_m=14$ şeklinde bir baskın çözüm daha bulmuştur. BÜÇAPSO algoritması ile elde

edilen $C_{\max}=16$, $W_t=73$, $W_m=13$ çözümü Gantt şeması Şekil 4.17’ de, HİÇAPSO algoritması ile elde edilen $C_{\max}=15$, $W_t=75$, $W_m=12$ çözümü Gantt şeması Şekil 4.18’ de verilmiştir.



Şekil 4.17. 8x8x27 Problemi BÜÇAPSO $C_{\max}=16$, $W_t=73$, $W_m=13$ çözümü



Şekil 4.18. 8x8x27 Problemi HİÇAPSO $C_{\max}=15$, $W_t=75$, $W_m=12$ çözümü

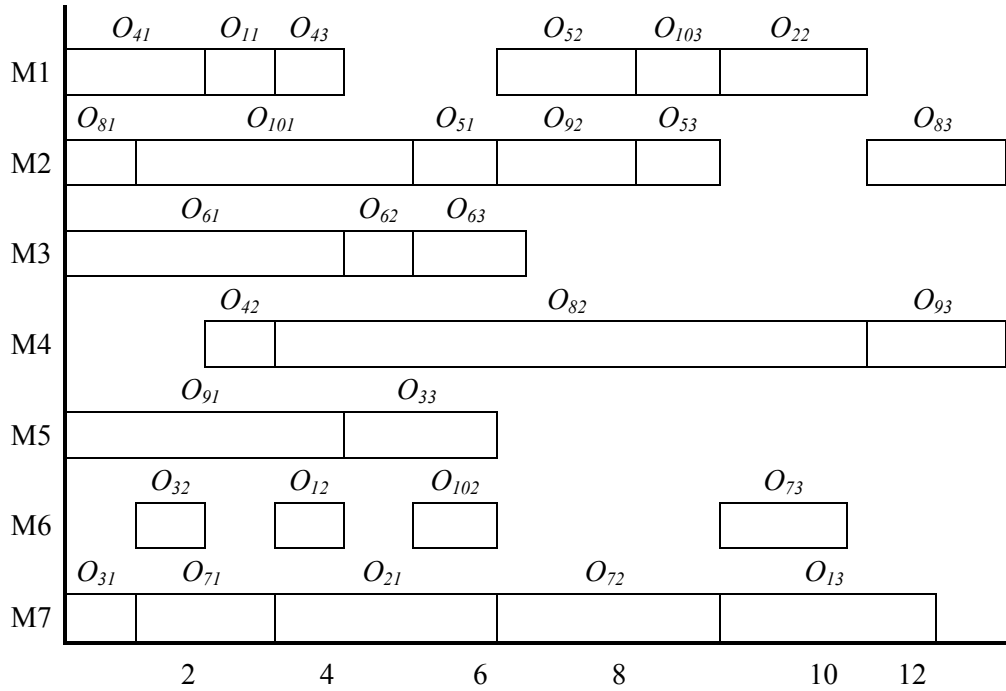
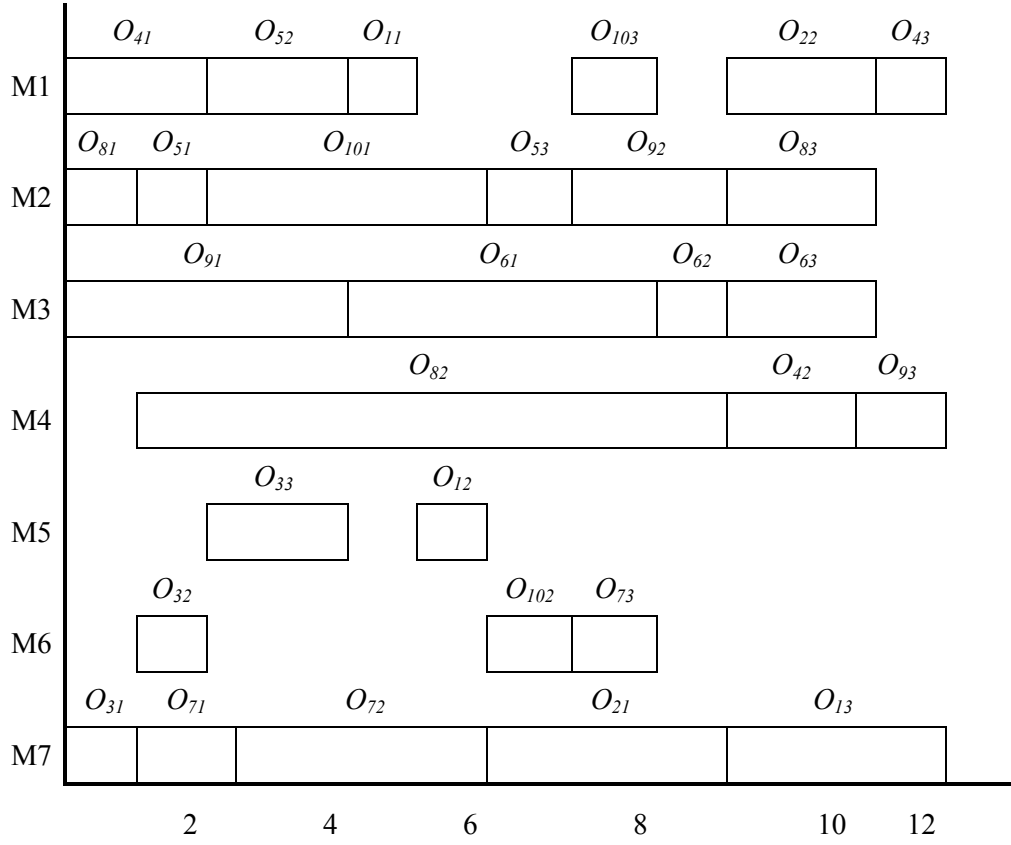
10x7x29 Problemi:

10 İş 7 makine ve 29 operasyondan oluşan toplam esnekliğe sahip problemin BÜÇAPSO ve HİÇAPSO sonuçları Tablo 4.14’ te verilmiştir. Bu sonuçlar, 8x8x27 problem setini çözen araştırmacılar içinde Li ve diğ. [87] 10x7x29 problem setini çözdüklerinden dolayı bu yazarlar tarafından geliştirilen (HTSA) algoritma sonuçlarıyla karşılaştırılmıştır. Ayrıca Li ve diğ. [96] tarafından geliştirilen melez KKA (P-DABC) sonuçlarıyla da karşılaştırılmıştır.

Tablo 4.14. 10x7x29 Problemine ait sonuçlar

	BÜÇAPSO	HİÇAPSO	(P-DABC)	HTSA
C_{max}	11 12 13	11 11 12 13	12 11 12	11 11
W_t	61 60 65	61 64 63 60	61 63 60	61 62
W_m	11 12 10	11 10 10 12	11 11 12	11 10

Tablo 4.14’ de verilen sonuçlara göre BÜÇAPSO ve HİÇAPSO algoritmaları, Li ve diğ. [96] tarafından geliştirilen melez KKA (P-DABC) sonuçlarına göre baskın çözümler bulmuştur. Ayrıca BÜÇAPSO algoritması (HTSA) algoritma sonuçlarına göre $C_{max}=12$, $W_t=60$, $W_m=12$ şeklinde bir baskın çözüm bulmuştur. HİÇAPSO algoritması ise (HTSA) algoritma sonuçlarına göre $C_{max}=13$, $W_t=60$, $W_m=12$ şeklinde yine baskın bir çözüm daha bulmuştur. BÜÇAPSO algoritması ile elde edilen $C_{max}=12$, $W_t=60$, $W_m=12$ sonucuna ait Gantt şeması Şekil 4.19’ da HİÇAPSO algoritması ile elde edilen $C_{max}=13$, $W_t=60$, $W_m=12$ sonucuna ait Gantt şeması ise Şekil 4.20’ de verilmiştir.



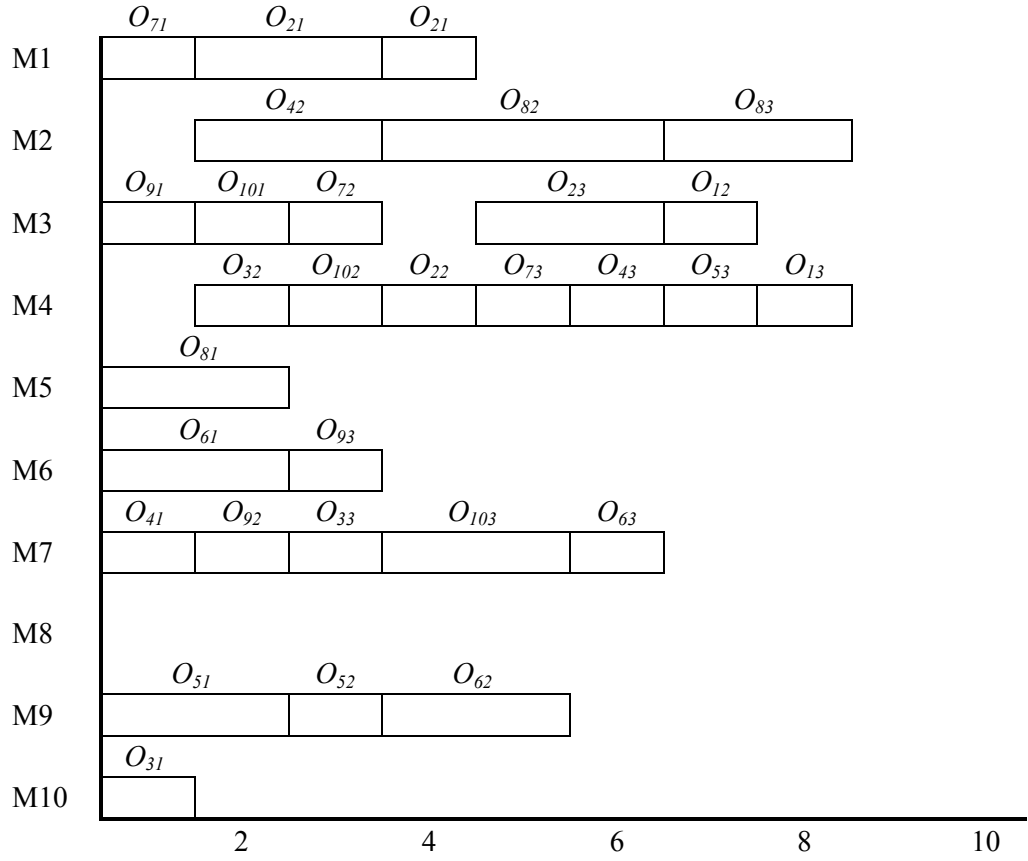
10x10x30 Problemi:

Çalışmamızda orta boyut problem sınıfının sonuncusu olan 10 iş 10 makine ve 30 operasyondan oluşan toplam esnekliğe sahip problemin BÜÇAPSO ve HİÇAPSO çözüm sonuçları Tablo 4.15’ te verilmiştir. Bu problem literatürde Kacem ve diğ. [10] tarafından geliştirilen (AL+CGA), Zhang ve diğ. [121] tarafından geliştirilen (PSO+TS), Xia ve Wu [28] tarafından geliştirilen (PSO+SA), Bagheri ve diğ. [132] tarafından geliştirilen (AIA) Wang ve diğ. [73] tarafından geliştirilen (MOGA), ve Li ve diğ. [87] tarafından geliştirilen (HTSA) algoritmaları sonuçları ile karşılaştırılmıştır.

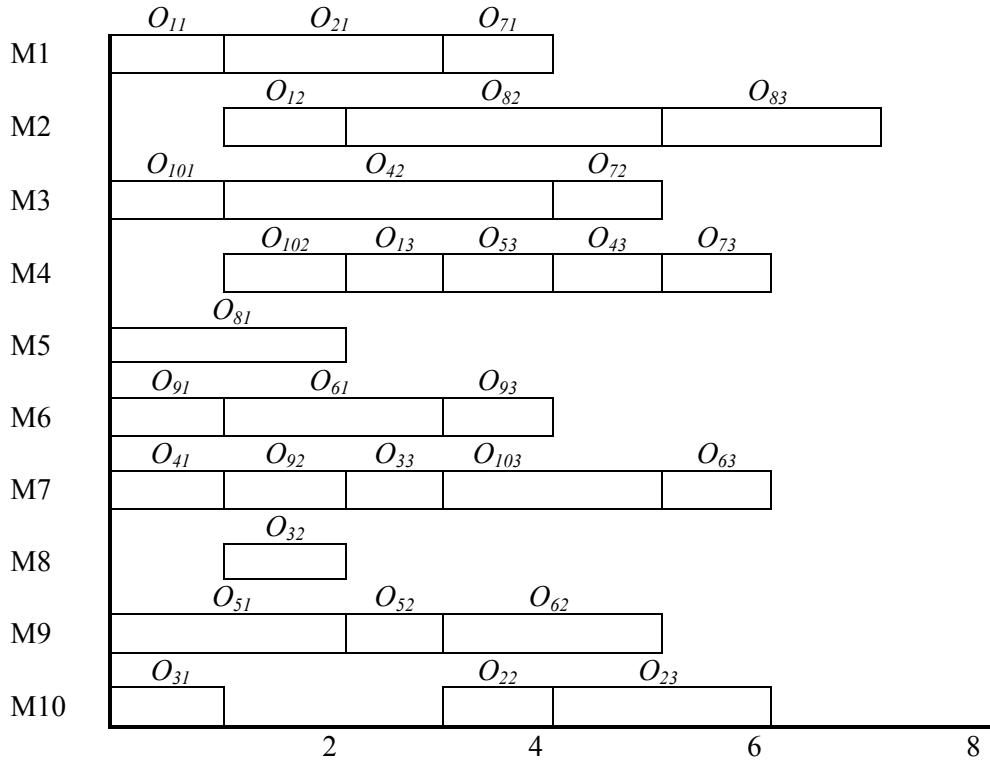
Tablo 4.15. 10x10x30 Problemine ait sonuçlar

	BÜÇAPSO					HİÇAPSO			AL + CGA	PSO + TS	PSO + SA	AIA	MOGA				HTSA		
C_{max}	8	7	8	8	7	8	7	8	7	7	7	7	8	7	8	7	7	7	8
W_t	42	43	41	44	45	42	42	41	45	43	44	43	42	42	41	45	43	42	42
W_m	6	6	7	5	5	5	6	7	5	6	6	5	5	6	7	5	5	6	5

Tablo 4.15’ te bulunan 10x10x30 problem sonuçlarına göre BÜÇAPSO ve HİÇAPSO algoritmalarımız (AL+CGA), (PSO+TS), (PSO+SA) ve (AIA) algoritmalarına göre baskın sonuçlar vermiştir. Her iki algoritmamız (MOGA) algoritma sonuçları ile aynı sonuçları vermiştir. Ayrıca HİÇAPSO algoritması (HTSA) algoritma sonuçlarının bulduğu iki sonucu da bulmakla beraber BÜÇAPSO ve HİÇAPSO algoritmaları $C_{max}=8$, $W_t=41$, $W_m=7$ şeklinde birer baskın sonuç daha vermiştir. BÜÇAPSO algoritmasıyla elde edilen $C_{max}=8$, $W_t=41$, $W_m=7$ çözümüne ait Gantt şeması Şekil 4.21’ de, HİÇAPSO algoritmasıyla elde edilen $C_{max}=7$, $W_t=42$, $W_m=6$ çözümüne ait Gantt şeması Şekil 4.22’ de verilmiştir.



Şekil 4.21. 10x10x30 Problemi BÜÇAPSO $C_{max}=8$, $W_t=41$, $W_m=7$ çözümü



Şekil 4.22. 10x10x30 Problemi HİÇAPSO $C_{max}=7$, $W_t=42$, $W_m=6$ çözümü

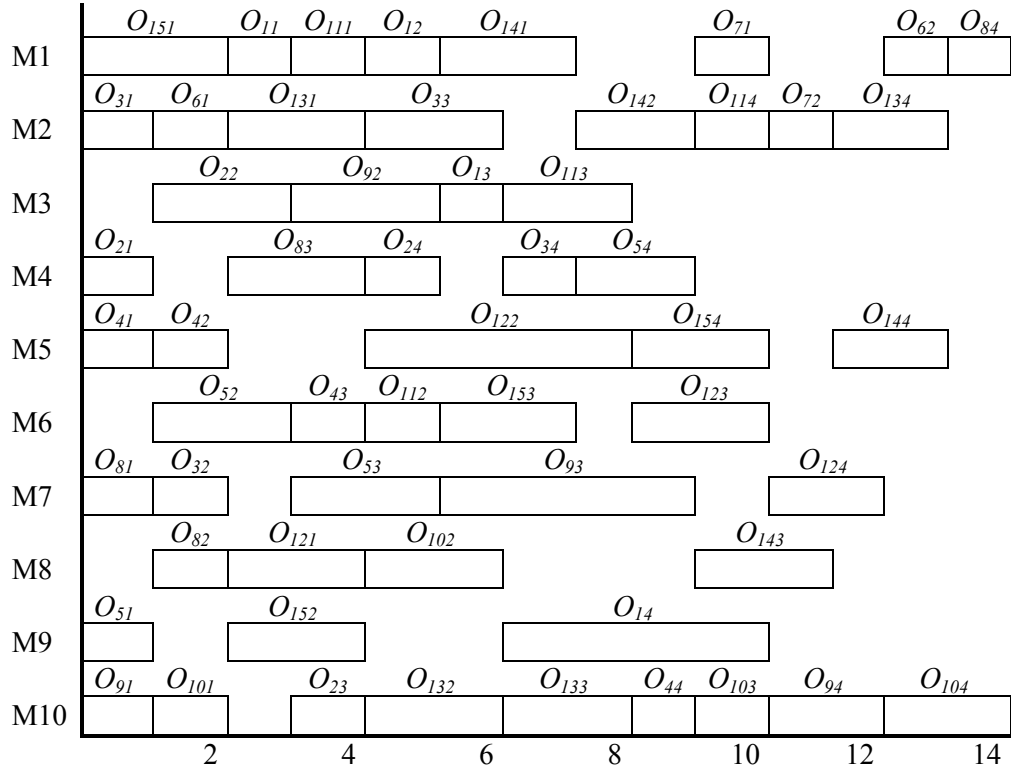
15x10x56 Problemi:

15 İş, 10 makine ve 56 operasyondan oluşan toplam esnekliğe sahip büyük boyut probleme ait BÜÇAPSO ve HİÇAPSO çözümleri Tablo 4.16’ da verilmiştir. Elde edilen sonuçlar, Kacem ve diğ. [10] tarafından geliştirilen (AL+CGA), Zhang ve diğ. [121] tarafından geliştirilen (PSO+TS), Xia ve Wu [28] tarafından geliştirilen (PSO+SA), Bagheri ve diğ. [132] tarafından geliştirilen (AIA) ve Wang ve diğ. [73] tarafından geliştirilen (MOGA) algoritmaları sonuçları ile karşılaştırılmıştır.

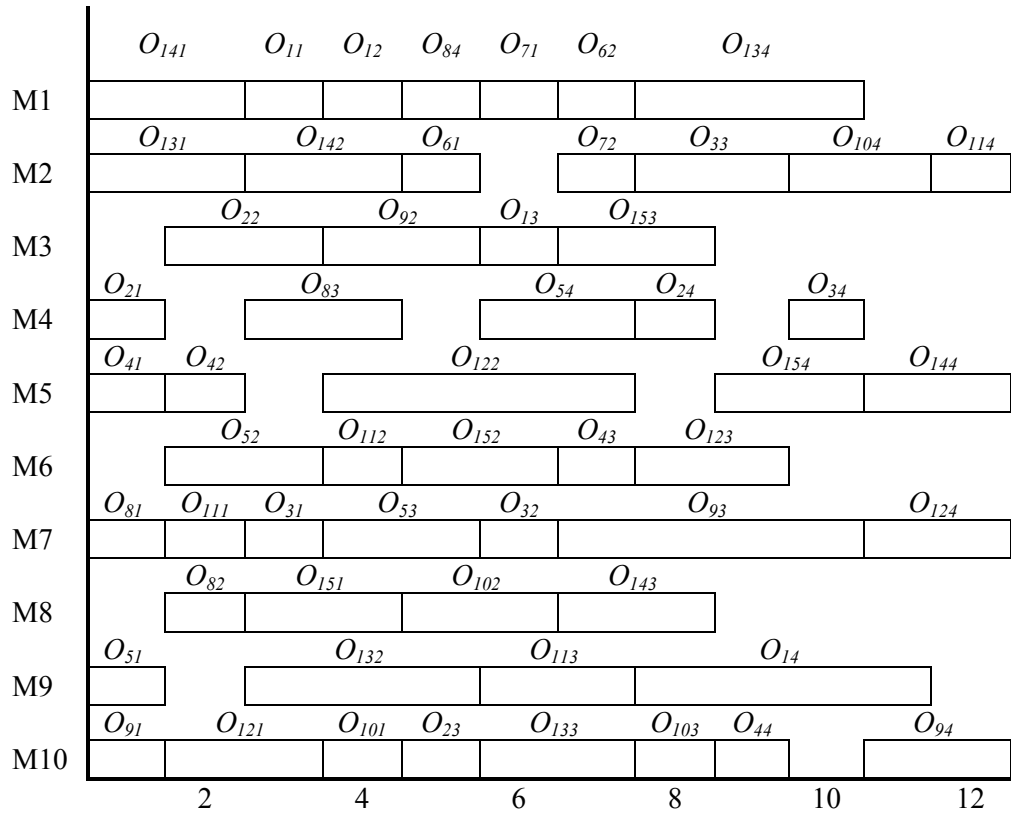
Tablo 4.16. 15x10x56 Problemine ait sonuçlar

	BÜÇAPSO	HİÇAPSO	AL + CGA	PSO + TS	PSO+ SA	AIA	MOGA
C_{max}	13 12 13 14 12 14 16	12 13 12 11 15	23 24	11	12	11	11 12 11
W_t	93 94 92 91 97 96 95	93 94 96 98 92	95 91	93	91	93	91 95 98
W_m	12 12 13 13 11 11 11	12 11 11 11 14	11 11	11	11	11	11 10 10

Tablo 4.16 göz önünde bulundurulduğunda BÜÇAPSO ve HİÇAPSO algoritmaları (AL+CGA) algoritma sonuçlarına göre baskın çözümler vermiştir. (PSO+TS) ve (AIA) algoritma sonuçlarına göre BÜÇAPSO algoritması $C_{max}=14$, $W_t=91$, $W_m=13$ ve $C_{max}=13$, $W_t=92$, $W_m=13$ şeklinde HİÇAPSO algoritması ise $C_{max}=15$, $W_t=92$, $W_m=14$ şeklinde baskın sonuçlar bulmuştur. Ayrıca HİÇAPSO algoritması (PSO+SA) algoritma sonuçlarına göre $C_{max}=11$, $W_t=98$, $W_m=11$ şeklinde bir baskın çözüm bulmuştur. Bu problem seti için (MOGA) algoritması BÜÇAPSO ve HİÇAPSO algoritma sonuçlarına göre daha iyi sonuçlar vermiştir. BÜÇAPSO algoritmasına ait $C_{max}=14$, $W_t=91$, $W_m=13$ çözümü Gantt şeması Şekil 4.23’ de HİÇAPSO algoritmasına ait $C_{max}=12$, $W_t=93$, $W_m=12$ çözümü Gantt şeması Şekil 4.24’ de verilmiştir.



Şekil 4.23. 15x10x56 Problemi BÜÇAPSO $C_{max}=14$, $W_t=91$, $W_m=13$ çözümü



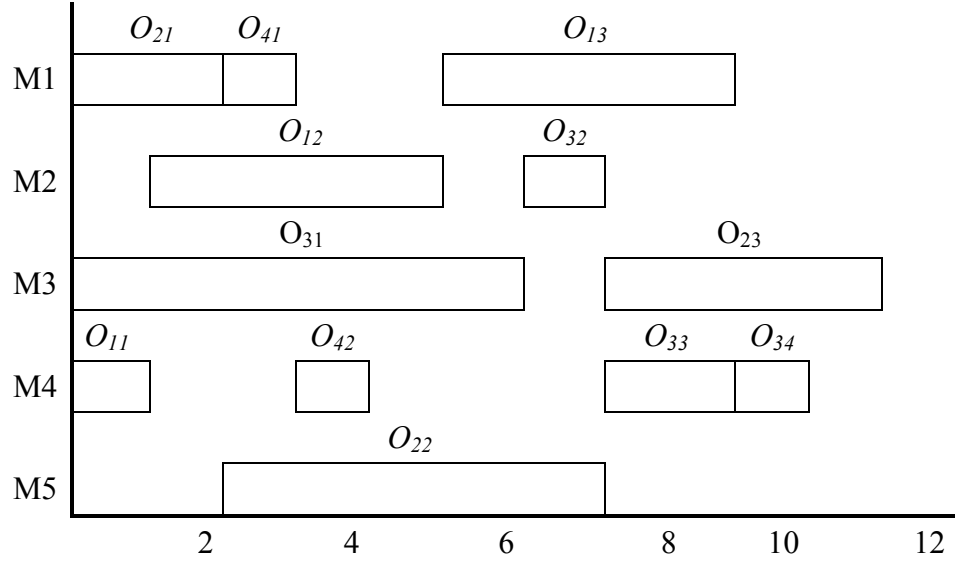
Şekil 4.24. 15x10x56 Problemi HİÇAPSO $C_{max}=12$, $W_t=93$, $W_m=12$ çözümü

Çalışmada ele alınan 2. grup amaç fonksiyonları; $C_{\max}$, $\sum C$ ve $\sum T$ 'den oluşmaktadır. Literatürde yer alan 1. grup amaç fonksiyonları için çözölen problem setleri 2. grup amaç fonksiyonları içinde çözülmüştür. BÜÇAPSO ve HİÇAPSO algoritmalarına ait pareto optimal çözümler toplu olarak Tablo 4.17' de verilmiştir. Ancak 2. grup amaç fonksiyonları için literatürde başka sonuç bulunmadığı için karşılaştırma yapılmamıştır. BÜÇAPSO ve HİÇAPSO algoritma sonuçları kendi aralarında değeriendirilmiştir.

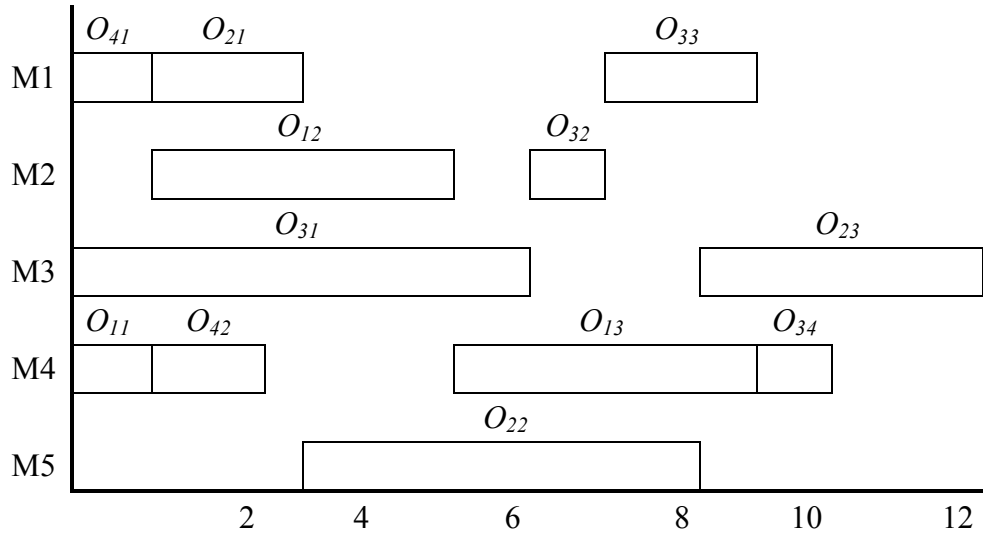
Tablo 4.17. Yeni amaç fonksiyonu tüm problemlere ait sonuçlar

Problem Boyutu	Amaç Fonk.	BÜÇAPSO				HİÇAPSO				
4x5x12	$C_{\max}$	11				11	12			
	$\sum C$	34				34	33			
	$\sum T$	15				15	15			
8x8x27	$C_{\max}$	17	16	15	14	17	14	16		
	$\sum C$	87	89	92	93	87	92	89		
	$\sum T$	35	37	40	41	35	40	37		
10x7x29	$C_{\max}$	13	11			13	15	12	11	12
	$\sum C$	84	88			83	82	84	86	87
	$\sum T$	20	20			16	18	20	20	19
10x10x30	$C_{\max}$	8	8	7		7				
	$\sum C$	51	50	52		49				
	$\sum T$	2	3	3		2				
15x10x56	$C_{\max}$	14	13	12		13	12			
	$\sum C$	135	136	146		126	132			
	$\sum T$	39	39	49		31	35			

Tablo 4.17.'de yer alan BÜÇAPSO ve HİÇAPSO sonuçları kendi aralarında değeriendirildiğinde 4x5x12 problemi için HİÇAPSO algoritması BÜÇAPSO algoritmasına göre daha iyi sonuç verdiğı görölmektedir. BÜÇAPSO algoritması ile elde edilen $C_{\max}=11$, $\sum C=34$, $\sum T=15$ çözümlüne ait Gantt şeması Şekil 4.25' te HİÇAPSO algoritması ile elde edilen $C_{\max}=12$, $\sum C=33$, $\sum T=15$ çözümlüne ait Gantt şeması Şekil 4.26' da verilmiştir.

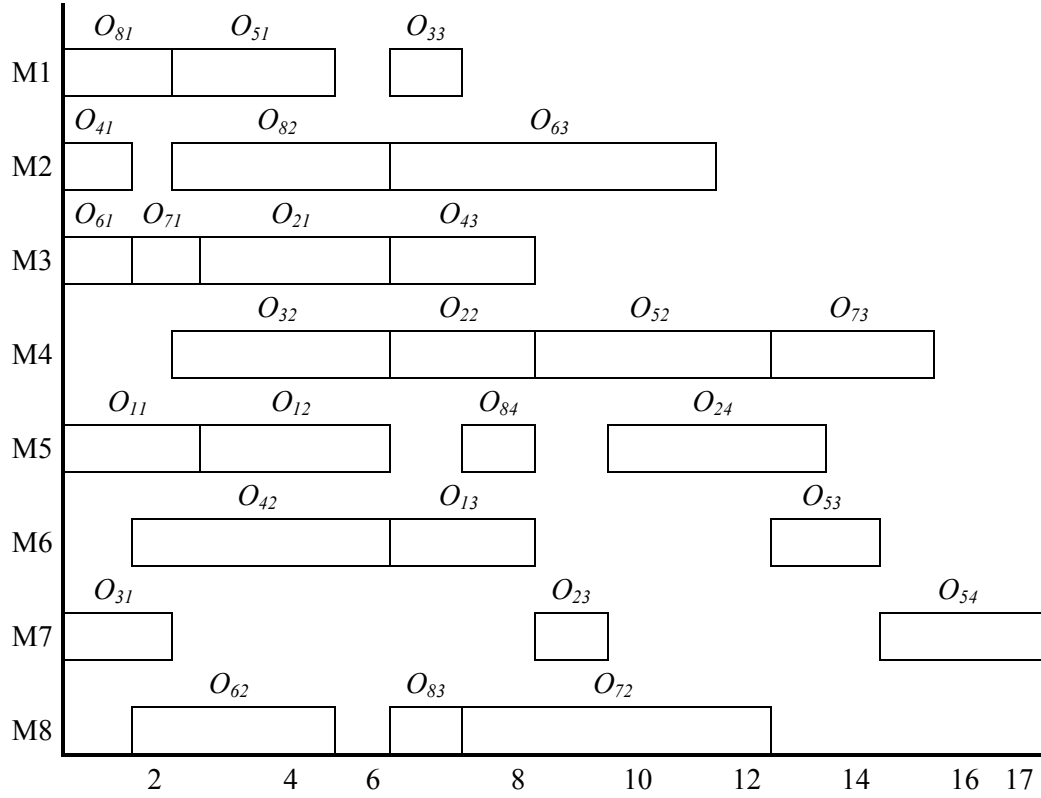


Şekil 4.25. 4x5x12 Problemi BÜÇAPSO ve HİÇAPSO $C_{max}=11$, $\sum C=34$, $\sum T=15$ çözümü



Şekil 4.26. 4x5x12 Problemi HİÇAPSO $C_{max}=12$, $\sum C=33$, $\sum T=15$ çözümü

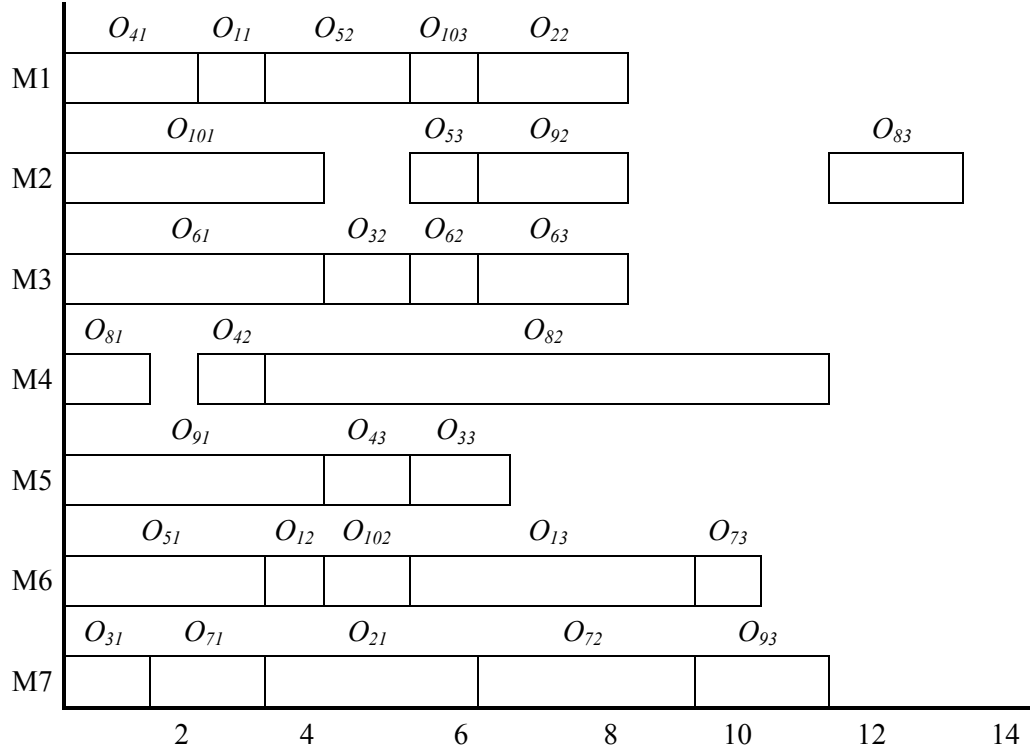
8 İş 8 makine 27 operasyondan oluşan kısmi esnekliğe sahip problemde BÜÇAPSO ve HİÇAPSO algoritmaları birbirine yakın sonuçlar bulmuşlardır. Bu problemde HİÇAPSO ve BÜÇAPSO algoritmaları 2 adet aynı sonucu bulmuştur. Ancak diğer sonuçlar incelendiğinde HİÇAPSO algoritmasının bulduğu $C_{max}=14$, $\sum C=92$ ve $\sum T=40$ sonucu BÜÇAPSO algoritmasının bulduğu $C_{max}=15$, $\sum C=92$ ve $\sum T=40$ ile $C_{max}=14$, $\sum C=93$ ve $\sum T=41$ sonuçlara göre baskın çözümdür. Her iki algoritmanın bulduğu $C_{max}=17$, $\sum C=87$ ve $\sum T=35$ sonucuna ait Gantt şeması Şekil 4.27’ de verilmiştir.



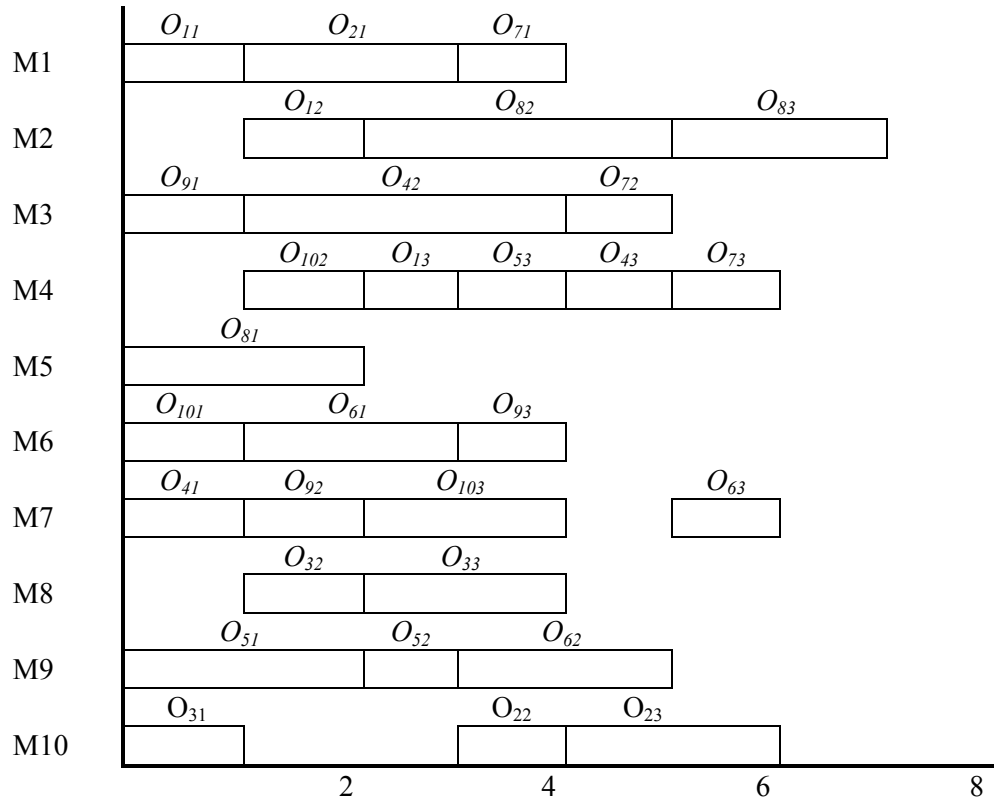
Şekil 4.27. 8x8x27 Problemi BÜÇAPSO ve HİÇAPSO $C_{max}=17$, $\sum C=87$, $\sum T=35$ çözümü

10 İş 7 makine ve 29 operasyondan oluşan tam esnekliğe sahip problem için BÜÇAPSO algoritmasının bulduğu $C_{max}=13$, $\sum C=84$ ve $\sum T=20$ sonucu HİÇAPSO algoritmasının bulduğu $C_{max}=13$, $\sum C=83$ ve $\sum T=16$ sonucuna göre basılgın çözümdür. Aynı şekilde BÜÇAPSO algoritmasının bulduğu $C_{max}=11$, $\sum C=88$ ve $\sum T=20$ sonucu HİÇAPSO algoritmasının bulduğu $C_{max}=11$, $\sum C=86$ ve $\sum T=20$ sonucuna göre basılgın çözümdür. 10x7x29 problemi için HİÇAPSO algoritması BÜÇAPSO algoritmasına göre daha iyi sonuçlar bulmuştur. HİÇAPSO Algoritması ile elde edilen $C_{max}=13$, $\sum C=83$ ve $\sum T=16$ sonucuna ait Gantt şeması Şekil 4.28’ de verilmiştir.

10 İş 10 makine ve 30 operasyondan oluşan toplam esnekliğe sahip problem için BÜÇAPSO algoritması 3 çeşit sonuç bulurken HİÇAPSO algoritması tek çözüm bulmuştur. Ancak HİÇAPSO algoritmasının bulduğu tek çözüm olan $C_{max}=7$, $\sum C=49$ ve $\sum T=2$ sonucu BÜÇAPSO algoritmasının bulduğu 3 sonuca göre baskın çözümdür. HİÇAPSO Algoritmasının bulduğu $C_{max}=7$, $\sum C=49$ ve $\sum T=2$ sonucuna ait Gantt şeması Şekil 4.29’ da verilmiştir.

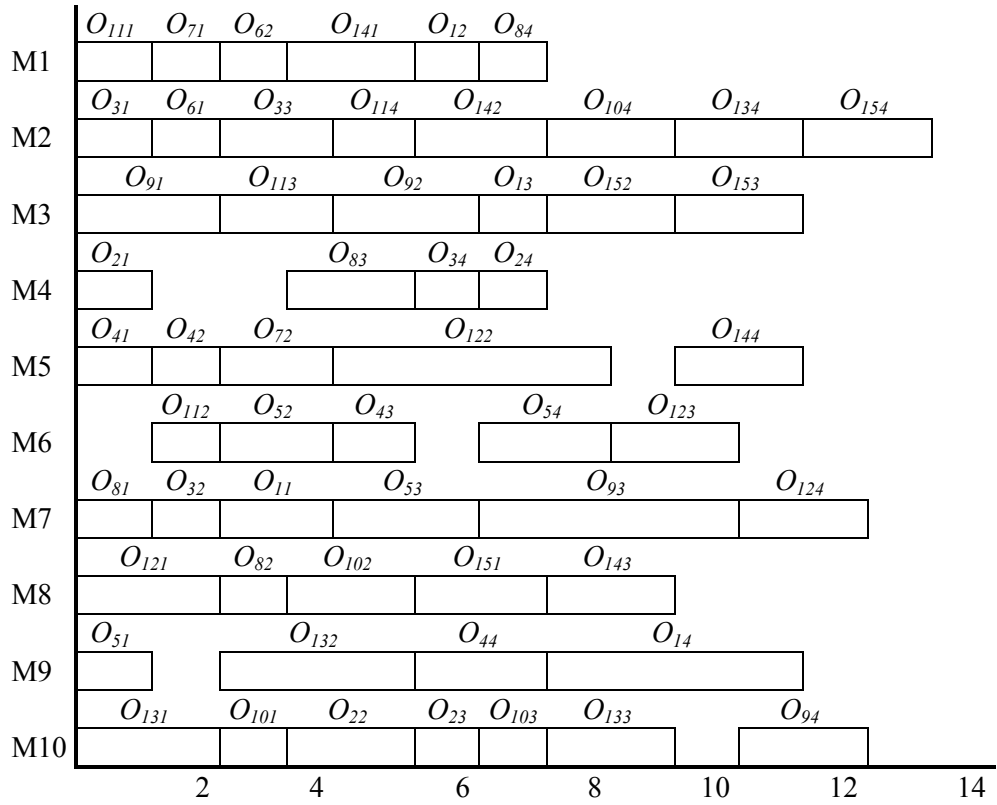


Şekil 4.28. 10x7x29 Problemi HİÇAPSO $C_{max}=13$, $\sum C=83$, $\sum T=16$ çözümü



Şekil 4.29. 10x10x30 Problemi HİÇAPSO $C_{max}=7$, $\sum C=49$, $\sum T=2$ çözümü

Çalışmada ele alınan büyük boyut problem setinden olan 15 iş 10 makine ve 56 operasyondan oluşan toplam esnekliğe sahip problem için BÜÇAPSO algoritması 3 baskın çözüm HİÇAPSO algoritması ise 2 baskın çözüm bulmuştur. Bu sonuçlar kendi aralarında değerlendirildiğinde, HİÇAPSO algoritması ile elde edilen $C_{max}=13$, $\sum C=126$ ve $\sum T=31$ sonucu BÜÇAPSO algoritmasıyla elde edilen $C_{max}=14$, $\sum C=135$ ve $\sum T=39$ ile $C_{max}=13$, $\sum C=136$ ve $\sum T=39$ sonuçlarına göre baskın çözümdür. Aynı şekilde HİÇAPSO algoritmasıyla elde edilen $C_{max}=12$, $\sum C=132$ ve $\sum T=35$ sonucu BÜÇAPSO algoritmasıyla elde edilen $C_{max}=12$, $\sum C=146$ ve $\sum T=49$ sonucuna göre baskın çözümdür. Bu problem setinde de HİÇAPSO algoritması BÜÇAPSO algoritmasına göre daha iyi sonuçlar bulmuştur. HİÇAPSO Algoritmasına ait $C_{max}=13$, $\sum C=126$ ve $\sum T=31$ sonucu Gantt şeması Şekil 4.30' da verilmiştir.



Şekil 4.30. 15x10x56 Problemi HİÇAPSO $C_{max}=13$, $\sum C=126$, $\sum T=31$ çözümü

5. SONUÇ VE ÖNERİLER

Bu çalışmada ele alınan EATÇ problemleri NP-zor sınıfındadır. Küçük boyutlu problemler için optimal sonuç bulunabilmektedir. Ancak orta ve büyük boyuttaki problemlerin optimal sonuçlarını bulmak oldukça zordur. Ayrıca gerçek üretim sistemlerinde genellikle daha büyük boyutlu problemlerle karşılaşmaktadır. Bu tür problemleri çözmek için etkin meta sezgisel yöntemlere ihtiyaç duyulur. Meta sezgisel yöntemler, özellikle zor kombinatorial problemlere uygulanan, en iyiye yakın sonuçlar veren pratik ve etkin yöntemlerdir. Problemlerin optimal çözümleri bulunamadığında veya optimal çözümleri bulmak için oldukça fazla bir zamana ihtiyaç olduğunda bu yöntemlere başvurulur. Bu çalışmada orta ve büyük boyutlu EATÇ problemlerini çözmek için çizelgeleme alanında başarıyla uygulanan ve kaliteli sonuçlar veren PSO algoritması kullanılmıştır.

PSO algoritmasının başlangıç çözümlerine sıralama kurallarından, EKİZ kuralı monte edilerek daha iyi amaç fonksiyonu değerine sahip parçacıklar elde edilmiştir. PSO Algoritmasının yerel optimumlara takılmasını engellemek ve global yakınsama hızını arttırmak için önerilen her iki modele YA sezgiseli monte edilmiştir. Çok amaçlı optimizasyon problemlerinde birden fazla çözüm alma arayışı literatürde pareto kavramını doğurmuştur. Pareto ile karar verici birden fazla iyi çözüm elde edebilmekte ve bu çözümler arasından tercihini yapabilmektedir. Bu nedenle çok amaçlı EATÇ problemlerinin çözümünde karar vericiye daha fazla karar alternatifi sunabilmek ve daha etkin sonuçlar alabilmek için geliştirilen modeller çok amaçlı hale getirilmiştir. Bu şekilde EATÇ problemlerinin çözümü için bütünsel ve hiyerarşik çözüm yaklaşımı olmak üzere çok amaçlı BÜÇAPSO ve HİÇAPSO algoritmaları literatüre kazandırılmıştır.

Gerçek yaşam optimizasyon problemleri genelde çok amaçlıdır. Bir işletmenin birimleri arasındaki amaçların farklılığı yüzünden karar vericiler bu amaçları gerçekleştirmekte zorlanabilmektedir. Literatürde yapılan çalışmaların büyük çoğunluğu tek amaçlı gerçekleştirilmeye çalışılan problemler olmuştur. Ancak çok amaçlı problemlerin çözümü tek amaçlı problemlerin çözümüne göre daha zor

olmaktadır. Amaçlar arasında genelde çelişkiler söz konusudur. Bir amaç en iyilenmeye çalışılırken diğer taraftan diğer amaç muhtemelen bozulabilmektedir. Bu çalışmada EATÇ problemlerinin çok amaçlı optimizasyonu ele alınmıştır. Çalışmada amaç fonksiyonu 2 ayrı grupta optimize edilmeye çalışılmıştır. Birinci grupta ele alınan performans ölçütleri, literatürde en çok kullanılan, C_{max} , W_m ve W_t ölçütleridir. Ele alınan 2. grup amaç fonksiyonları ise EATÇ problemlerinde şu ana kadar ele alınmamış olan C_{max} , $\sum C$ ve $\sum T$ ölçütleridir.

BÜÇAPSO ve HİÇAPSO algoritmalarının 1. grup amaç fonksiyonları için verimliliği literatürde yer alan bir adet küçük boyut, 3 adet orta boyut ve 1 adet büyük boyut olmak üzere toplam 5 adet problem setiyle ölçülmüştür. Literatüre sunulan her iki algoritma Kacem ve diğ. [10] tarafından geliştirilen (AL+CGA), Zhang ve diğ. [121] tarafından geliştirilen (PSO+TS), Wang ve diğ. [73] tarafından geliştirilen (MOGA), Li ve diğ. [87] tarafından geliştirilen (HTSA), Xia ve Wu [28] tarafından geliştirilen (PSO+SA), Bagheri ve diğ. [132] tarafından geliştirilen (AIA) ve Li ve diğ. [96] tarafından geliştirilen (P-DABC) algoritma sonuçlarıyla karşılaştırılmıştır. BÜÇAPSO ve HİÇAPSO algoritmaları (AL+CGA), (PSO+TS), (HTSA), (AIA), (P-DABC) ve (PSO+TS) algoritma sonuçlarına göre baskın çözümler vermekle beraber (MOGA) algoritma sonuçlarına göre genelde aynı çözümleri vermiştir. Her iki algoritmanın 2. grup amaç fonksiyon sonuçları da literatürde bundan sonra yazarların çalışma yaparak kendi sonuçlarını elde ettiğimiz sonuçlar ile kıyaslayarak algoritmalarının verimliliğini ölçmek üzere sunulmuştur.

EATÇ Problemleri operasyonların makinelere atanması alt problemi ile makinelere atanan bu operasyonların hangi sırada işleneceğinin belirlenmesi alt problemlerinden oluşmaktadır. Literatürde bu iki alt problemin çözümü için bütünleşik ve hiyerarşik olmak üzere iki tür yaklaşım bulunmaktadır. Bütünleşik yaklaşımda atama ve sıralama alt problemleri eş zamanlı çözülürken, hiyerarşik yaklaşımda iki alt problem birbirinden bağımsız olarak çözülmemektedir. Detaylı yapılan literatür taramasında EATÇ problemlerinin çözümü için hiyerarşik veya bütünleşik yaklaşımlardan hangisinin daha etkin sonuçlar verdiği üzerine çeşitli görüşler bulunmaktadır. Dauzere-Peres ve Paulli [60] ile Mastrolilli ve Gambardella [48] çalışmalarında, bütünleşik çözüm yaklaşımının daha iyi sonuçlar verdiğini belirtirken, Fattahi ve diğ. [12] hiyerarşik çözüm yaklaşımının daha iyi sonuçlar verdiğini söylemiştir.

Çalışmamızda EATÇ problemlerinin çözümü için hiyerarşik ve bütünlük olmak üzere iki adet çok amaçlı PSO algoritması geliştirilmiştir. Ancak hiyerarşik ve bütünlük çözüm yaklaşımlarından hangisinin daha etkin sonuçlar verdiği üzerine çözülen 5 adet problem seti sonuçlarına göre yorum yapmak doğru değildir. İki çözüm yaklaşımından hangisinin daha etkin sonuçlar verdiği çalışmaları için EATÇ problemlerinin literatürde yer alan diğer problem setleriyle çözülerek deney tasarımı çalışmaları sonucunda karar verilebilir.

Çizelgeleme problemleri için hazırlanan programlar, bir takım şartları yok saydıklarından gerçek uygulamaya pek yanaşamamaktadırlar. Bu da yapılan çizelgelerin tam anlamıyla gerçek yaşam problemlerine uygulanamamasına sebep olmaktadır. İşletmelerin dinamik koşullarından dolayı iş çizelgeleme problemlerine çözüm bulmaları daha zordur. Gerçek hayatta karşılaşılan çizelgeleme problemleri, üretim çevresinin dinamik şartlarının yanı sıra işletme hedeflerinin gerçekleştirmeye çalıştığı birden fazla amacın olması, bu amaçların birbirleriyle çatışması, işletmedeki çizelgeleme kısıtlarının çeşitliliğinden dolayı karmaşık olarak nitelendirilirler. Bu tür problemler NP-zor yapıya sahip kombinatoriyel optimizasyon problemlerindendir.

Detaylı yapılan literatür taraması sonucunda EATÇ problemlerinin çoğunluğunun literatürden alınan veriler olduğu görülmüştür. Tek ve çok amaçlı olmak üzere incelenen toplam 93 çalışma içerisinde 7 tanesi tek amaçlı 3 tanesi çok amaçlı olmak üzere toplam 10 tanesi gerçek yaşam problemlerine uygulanmış 83 çalışma ise literatürden alınan teorik problem verileri ile değerlendirilmiştir. Oysaki gerçek hayat çizelgeleme problemleri, literatüre göre çok daha karmaşıktır ve çözümleri daha zordur. Araştırmacılar çizelgeleme alanındaki problemlerini gerçek yaşam problemlerinden seçmeli, verileri gerçek hayata yönelik bir uygulama olması açısından, işletmelerdeki verileri alarak çözümler sunmalıdırlar.

Çok amaçlı EATÇ problemlerinde ele alınan C_{max} , W_m ve W_t amaç fonksiyonları en çok göz önünde bulundurulmuş amaçlar olmuştur. İncelenen toplam 47 çok amaçlı çalışma içerisinde 28 tanesi C_{max} , W_m ve W_t olmak üzere 19 tanesi de diğer amaç fonksiyonları bazında yapılmıştır. Günümüzde teknolojinin gelişmesiyle işletmelerin değişik amaçları ortaya çıkmıştır. Karar vericiler için yeni çıkan amaçlar doğrultusunda çok amaçlı EATÇ problemleri üzerinde çalışmalar çok azdır.

Araştırmacılar yeni amaç fonksiyonları için yeni algoritmalar ve çözümler gerçekleştirebilirler.

EATÇ Problemleri diğer çizelgeleme problemleri kadar geniş bir literatür alt yapısına sahip değildir. Bu problem türünün, ATÇ problemlerine göre daha karmaşık bir yapıya sahip olması ve bilgisayar hesaplama kapasitelerinin yeterli hızlara yeni yeni ulaşması v.b. nedenlerden dolayı literatürün kısıtlı kalmasına neden olmuştur. EATÇ problemlerinin değişik varyasyonları göz önünde bulundurularak çalışmalar yapılması önerilmektedir.

Literatürde yer alan statik teslim tarihi belirleme kuralları, atölye yoğunluğunu göz ardı ederek sadece işin özelliklerini dikkate aldıklarından daha gerçekçi sonuçlara ulaşamamaktadır. Dinamik teslim tarihi belirleme kuralları atölye durumunu, iş istasyonlarının yoğunluğu bakımından göz önünde bulundurularak işlerin akış paylarını belirler. EATÇ Problemleri, dinamik teslim tarihli olarak ele alınıp çözümler gerçekleştirilmesi yapılabilir.

Yine ayrıca ele alınan problemler, işin geliş zamanlarının birbirinden farklı olduğu dinamik durumlar için çok amaçlı EATÇ problemleri incelenebilir. Diğer taraftan öğrenme etkisinin de problemlerde dikkate alınması ile gerçek üretim sistemlerinde daha etkin ve kullanışlı çizelgelerin kurulabileceği ve bunun sisteme fayda sağlayabileceği düşünülmektedir.

Çok amaçlı hazırlık zamanlı EATÇ problemleri üzerinde çalışmalar yapılabilir. Hazırlık zamanlı olarak değişik çok amaçlı problemlerin incelenmesi ve ayrıca sıra bağımlı hazırlık zamanlı problemlerinde dikkate alınmasıyla çizelgeleme literatürüne önemli katkılar sağlayacağı düşünülmektedir.

Çok amaçlı EATÇ problemlerinin çözümünde TA, TB, GA gibi meta sezgisel yöntemlerin son dönemlerde sıkça kullanıldığı görülmektedir. Diğer taraftan yapay zekâ uygulamalarından uzman sistemlerin de bu alanda kullanılması ile karar vericiye uygun çözüm seçenekleri sunmada yararlı olabileceği düşünülmektedir.

KAYNAKLAR

- [1] Liu H., Abraham A., Wang Z., A Multi-swarm approach to multi-objective flexible job-shop scheduling problems, *Fundamental Informatics*, 2009, **95**, 465-489.
- [2] Brandimarte P., Routing and scheduling in a flexible job-shop by tabu search, *Annals of Operations Research*, 1993, **2**, 158-183.
- [3] Zitzler E., Evolutionary algorithms for multiobjective optimization: methods and applications, Ph. D. Thesis, Swiss Federal Institute of Technology (ETH), Computer Engineering and Networks Laboratory, Zurich, 1999, 13398.
- [4] Pinedo M. L., *Scheduling theory, algorithms, and systems*, 4th ed., Springer, New York, 2008.
- [5] Leung J., *Handbook of scheduling: algorithms, models, and performance analysis*, 4th ed., Chapman and Hall/CRC Computer and Information Science Series, New York, 2004.
- [6] Madejski J., Dynamic scheduling for agent based manufacturing systems, *Journal of Achievements in Materials and Manufacturing Engineering*, 2010, **40**, 66-69.
- [7] Kundakcı N., Üretim sistemlerinde dinamik iş çizelgeleme problemlerinin sezgisel yöntemlerle çözülmesi, Doktora Tezi, Pamukkale Üniversitesi, Sosyal Bilimler Enstitüsü, Denizli, 2013, 332597.
- [8] Cihanlı Ö., Hibrid akış tipi atölyede çizelgeleme, Yüksek Lisans Tezi, Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa, 2010, 283025.
- [9] Kacem I., Hammadi S., Borne P., Approach by localization and genetic manipulation algorithm for flexible job-shop scheduling problem, *International Conference on Systems, Man, and Cybernetics*, Tucson, USA, 7–10 October 2001.
- [10] Kacem I., Hammadi S., Borne P., Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems, *Transactions on Systems, Man, and Cybernetics—Part c: Applications and Reviews*, 2002, **32**, 1-13.
- [11] Kacem I., Hammadi S., Borne P., Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, 2002, **60**, 245-276.

- [12] Fattahi P., Mehrabad M. S., Jolai F., Mathematical modeling and heuristic approaches to flexible job shop scheduling problems, *Journal of Intelligent Manufacturing*, 2007, **18**, 331-342.
- [13] Eren T., Çok ölçütlü akış tipi çizelgeleme problemleri için çözüm yaklaşımları, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2004, 156037.
- [14] Meto B., Esnek atölye tipi çizelgeleme problemi için bir hibrid genetik algoritma yaklaşımı, Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2013, 334377.
- [15] Roshanaei V., Mathematical modelling and optimization of flexible job shops scheduling problem, M. Sc. Thesis, University of Windsor, Faculty of Graduate Studies through Industrial & Manufacturing Systems Engineering, Windsor, Canada, 2012, 77288.
- [16] Ak B., Seçilmiş bir endüstriyel tesiste üretim çizelgeleme problemi ve genetik algoritma yöntemi ile optimizasyon, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun, 2012, 306913.
- [17] Demir T., Esnek atölye tipi çizelgeleme problemleri için kafiye aktarımı stratejisi: otomotiv sektöründe bir uygulama, Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Erzurum, 2012, 312941.
- [18] Yang S., Ma Y., Xu D., Yang J., Minimizing total completion time on a single machine with a flexible maintenance activity, *Computers & Operations Research*, 2011, **38**, 755-770.
- [19] Zhang G., Gao L., Shi Y., An effective genetic algorithm for the flexible job-shop scheduling problem, *Expert Systems with Applications*, 2011, **38**, 3563-3573.
- [20] Xing L. N., Chen Y. W., Yang K. W., Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems, *Computational Optimization and Applications*, 2011, **48**, 139-155.
- [21] Gutierrez C., Garcia-Magarino I., Modular design of a hybrid genetic algorithm for a flexible job-shop scheduling problem, *Knowledge-Based Systems*, 2011, **24**, 102-112.
- [22] Amiri M., Zandieh M., Yazdani M., Bagheri A., A variable neighbourhood search algorithm for the flexible job-shop scheduling problem, *International Journal of Production Research*, 2010, **48**, 5671-5689.
- [23] De Giovanni L., Pezzella F., An Improved genetic algorithm for the distributed and flexible job-shop scheduling problem, *European Journal of Operational Research*, 2010, **20**, 395-408.

- [24] Aladağ A., Tekrar işlemeli esnek atölye tipi çizelgeleme problemi için yapay bağışıklık sistemi ile bir çözüm yaklaşımı, Yüksek Lisans Tezi, Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2010, 252658.
- [25] Özgüven C., Özbakır L., Yavuz Y., Mathematical models for job-shop scheduling problems with routing and process plan flexibility, *Applied Mathematical Modelling*, 2010, **34**, 1539-1548.
- [26] Defersha F. M., Chen M., A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups, *International Journal Advanced Manufacturing Technology*, 2010, **49**, 263-279.
- [27] Lei D., A genetic algorithm for flexible job shop scheduling with fuzzy processing time, *International Journal of Production Research*, 2010, **48**, 2995-3013.
- [28] Xia W., Wu Z., An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers & Industrial Engineering*, 2005, **48**, 409-425.
- [29] Pezzella F., Morganti G., Ciaschetti G., A genetic algorithm for the flexible job-shop scheduling problem, *Computers & Operations Research*, 2008, **35**, 3202-3212.
- [30] Fahmy S. A., Balakrishnan S., Elmekawy T. Y., A generic deadlock-free reactive scheduling approach, *International Journal of Production Research*, 2009, **47**, 5657-5676.
- [31] Girish B. S., Jawahar N., Scheduling job shop associated with multiple routings with genetic and ant colony heuristics, *International Journal of Production Research*, 2009, **47**, 3891-3917.
- [32] Yazdani M., Gholami M., Zandieh M., Mousakhani M., A Simulated annealing algorithm for flexible job-shop scheduling problem, *Journal of Applied Sciences*, 2009, **9**, 662-670.
- [33] Yazdani M., Amiri M., Zandieh M., Flexible job-shop scheduling with parallel variable neighborhood search algorithm, *Expert Systems with Applications*, 2010, **37**, 678-687.
- [34] Süer B. İ., Order-driven flexibility management in make-to-order companies with flexible shops, M. Sc. Thesis, Middle East Technical University, The Graduate School Of Natural And Applied Sciences, Ankara, 2009, 255442.
- [35] Xing L. N., Chen Y. W., Yang K. W., Double layer ACO algorithm for the multi-objective FJSSP, *New Generation Computing*, 2008, **26**, 313-327.
- [36] Huang K. L., Liao C. J., Ant colony optimization combined with taboo search for the job shop scheduling problem, *Computers & Operations Research*, 2008, **35**, 1030-1046.

- [37] Chen J., Chen K., Wu C., Chen W., A study of the flexible job shop scheduling problem with parallel machines and reentrant process, *Journal of Advanced Manufacturing Technology*, 2008, **39**, 344-354.
- [38] Lei D., Guo X. P., Solving fuzzy flexible job shop scheduling problems using genetic algorithm, *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kuming, China, 12-15 July 2008.
- [39] Weng M. X., Wu Z., Qi G., Zheng L., Multi-agent-based workload control for make to order manufacturing, *International Journal of Production Research*, 2008, **46**, 2197-2213.
- [40] Zribi N., Kacem I., El Kamel A., Borne P., Assignment and scheduling in flexible job-shops by hierarchical optimization, *Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, 2007, **37**, 652-662.
- [41] Ho N. B., Tay J. C., Lai E. M. K., An effective architecture for learning and evolving flexible job-shop schedules, *European Journal of Operational Research*, 2007, **179**, 316-333.
- [42] Liouane N., Saad I., Hammadi S., Borne P., Ant systems & local search optimization for flexible job shop scheduling production, *International Journal of Computers, Communications & Control*, 2007, **2**, 174-184.
- [43] Saidi-Mehrabad M., Fattahi P., Flexible job shop scheduling with tabu search algorithms, *International Journal Advanced Manufacturing Technology*, 2007, **32**, 563-570.
- [44] Gomes M. C., Barbosa-Povoa A. P., Novais A. Q., Optimal scheduling for flexible job shop operation, *International Journal of Production Research*, 2005, **43**, 2323-2353.
- [45] Jansen K., Mastrolilli M., Solis-oba R., Approximation algorithms for flexible job shop problems, *International Journal of Foundations of Computer Science*, 2005, **16**, 361-379.
- [46] Zribi N., Kacem I., El Kamel A., Borne P., Optimization by phases for the flexible job shop scheduling problem, *Computational Engineering in Systems Applications*, 2004, **3**, 1889-1895.
- [47] Baykasoglu A., Linguistic-based meta-heuristic optimization model for flexible job shop scheduling, *International Journal of Production Research*, 2002, **40**, 4523-4543.
- [48] Mastrolilli M., Gambardella L. M., Effective neighborhood functions for the flexible job shop problem, *Journal of Scheduling*, 2000, **3**, 3-20.
- [49] Zhang R., Wu C., A Simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective, *Computers & Operations Research*, 2011, **38**, 854-867.

- [50] Baykasoglu A., Özbakır L., Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system, *International Journal Production Economic*, 2010, **124**, 369-381.
- [51] Essafi I., Mati Y., Dauzere-Peres S., A genetic local search algorithm for minimizing total weighted tardiness in the job-shop scheduling problem, *Computers & Operations Research*, 2008, **35**, 2599-2616.
- [52] Kunadilok J., Heuristics for scheduling reentrant flexible job shops with sequence dependent setup times and limited buffer capacities, Ph. D. Thesis, Clemson University, College of Engineering and Science, Clemson, 2007, 3274324.
- [53] Zribi N., El Kamel A., Borne P., Total tardiness in a flexible job-shop, *Multiconference on Computational Engineering in Systems Applications*, Beijing, China, 4-6 October 2006.
- [54] Büyükköprü E., Flexible job shop scheduling via simulation under sequence dependent setup times, M. Sc. Thesis, Dokuz Eylül University, Graduate School Of Natural And Applied Sciences, İzmir, 2005, 202626.
- [55] Jang Y., Kım K., Jang S., Park J., Flexible job shop scheduling with multi-level job structures, *JSME International Journal Series*, 2003, **46**, 33-38.
- [56] Alvarez-Valdes R., Fuertes A., Tamarit J. M., Gimenez G., Ramos R., A heuristic to schedule flexible job-shop in a glass factory, *European Journal of Operational Research*, 2005, **165**, 525-534.
- [57] Wu Z., Weng M. X., Multiagent scheduling method with earliness and tardiness objectives in flexible job shops, *Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics*, 2005, **35**, 293-301.
- [58] Dosdoğru A. T., Dynamic flexible job shop scheduling with simulation optimization by using genetic algorithm, M. Sc. Thesis, University of Gaziantep, Graduate School Of Natural And Applied Sciences, Gaziantep, 2012, 318443.
- [59] Hurink E., Jurisch B., Thole M., Tabu search for the job shop scheduling problem with multi-purpose machine, *Operations Research Spektrum*, 1994, **15**, 205-215.
- [60] Dauzere-Peres S., Paulli J., An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search, *Annals of Operation Research*, 1997, **70**, 281-306.
- [61] Fırlalı A., Optimizasyona giriş, *Optimizasyon Seminerleri Dizisi*, Kocaeli, Türkiye, 23-27 Haziran 2008.
- [62] Aydın A., Metasezgisel yöntemlerle uçak çizelgeleme problemi optimizasyonu, Doktora Tezi, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, 2009, 231704.

- [63] Şevkli A. Z., En fazla kazanç sağlayan rota bulma problemlerinin sezgisel yöntemlerle incelenmesi, Doktora Tezi, Gebze Yüksek Teknoloji Enstitüsü, Mühendislik ve Fen Bilimleri Enstitüsü, Gebze, 2010, 271489.
- [64] Goldberg D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, 4th ed. Addison-Wesley Publishing Company, London, 1989.
- [65] Engin O., Fırlı A., Genetik Algoritmalarla akış tipi çizelgelemede üreme yöntemi optimizasyonu, *itü dergisi/d mühendislik*, 2002, **1**, 304-316.
- [66] Tamaki H., Ono T., Murao H., Kitamura S., Modeling and genetic solution of a class of flexible job shop scheduling problems, *International Emerging Technologies and Factory Automation Proceedings*, 2001, **1**, 7241-7248.
- [67] Tanev I. T., Uozumi T., Morotome Y., Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach, *Applied Soft Computing*, 2004, **5**, 87-100.
- [68] Chan F. T. S., Wong T. C., Chan L. Y., Flexible job-shop scheduling problem under resource constraints, *International Journal of Production Research*, 2006, **44**, 2071-2089.
- [69] Gao J., Sun L., Gen M., A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems, *Computers & Operations Research*, 2008, **35**, 2892-2907.
- [70] Gao J., Gen M., Sun L., Zhao X., A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems, *Computers & Industrial Engineering*, 2007, **53**, 149-162.
- [71] Lin L., Jia-Zhen H., Multi-Objective flexible job-shop scheduling problem in steel tubes production, *Systems Engineering*, 2009, **8**, 117-126.
- [72] Unachak P., An Adaptive representation for a genetic algorithm in solving flexible job-shop scheduling and rescheduling problems, Ph. D. Thesis, Michigan State University, College of Engineering, Michigan, 2010, 3435155.
- [73] Wang X., Gao L., Zhang C., Shao X., A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem, *International Journal Advanced Manufacturing Technology*, 2010, **51**, 757-767.
- [74] Prakash A., Chan F. T. S., Deshmukh S. G., FMS Scheduling with knowledge based genetic algorithm approach, *Expert Systems with Applications*, 2011, **38**, 3161-3171.
- [75] Rabiee M., Zandieh M., Ramezani P., Bi-objective partial flexible job shop scheduling problem: NSGA-II, NREGA, MOGA and PAES approaches, *International Journal of Production Research*, 2012, **24**, 7327-7342.

- [76] Kirkpatrick S., Gelatt C. D., Vecchi M. P., Optimization by simulated annealing, *New Science Series*, 1983, **220**, 671-680.
- [77] Serafin P., Simulated annealing for multiple objective optimization problems, *Expand and Enrich the Domains of Thinking and Application*, 1994, **1**, 283-292.
- [78] Ray T., Gokarn R., Sha O., A Global optimization model for ship design, *Computers in Industry*, 1995, **26**, 175-192.
- [79] Ruiz-Torres A. J., Ensore E. E., Barton R. R., Simulated annealing heuristics for the average flow-time and the number of tardy jobs bi-criteria identical parallel machine problem, *Computers and Industrial Engineering*, 1997, **33**, 257-260.
- [80] Glover L., *Tabu Search*, 2nd ed., Kluwer Academic Publishers, Boston, 1997.
- [81] Fattahi P., A Hybrid multi objective algorithm for flexible job shop scheduling, *Proceedings of World Academy of Science, Engineering and Technology*, 2009, **38**, 2070-2077.
- [82] Gholami M., Zandieh M., Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop, *Journal of Intelligent Manufacturing*, 2009, **20**, 481-498.
- [83] Frutos M., Olivera A. C., Tohme F., A memetic algorithm based on a NSGAII scheme for the flexible job-shop scheduling problem, *Annual Operation Research*, 2010, **181**, 745-765.
- [84] Gandibleux X., Mezdaoui A., Freville A., A Multiobjective tabu search procedure to solve combinatorial optimization problems, *Advances in Multiple Objective and Goal Programming in Economics and Mathematical Systems*, 1997, **6**, 291-300.
- [85] Hansen M., Tabu search for multi objective optimization: MOTS, *13th International Conference on Multiple Criteria Decision Making*, Cape Town, South Africa, 6-10 January 1997.
- [86] Li J., Pan Q., Chemical-reaction optimization for flexible job-shop scheduling problems with maintenance activity, *Applied Soft Computing*, 2012, **12**, 2896-2912.
- [87] Li J., Pan Q., Liang Y. C., An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems, *Computers & Industrial Engineering*, 2010, **59**, 647-662.
- [88] Baykasoglu A., Özbakır L., Sönmez A. İ., Using multiple objective tabu search and grammars to model and solve multi-objective flexible job shop scheduling problems, *Journal of Intelligent Manufacturing*, 2004, **15**, 777-785.

- [89] Özbakır L., Çok objektifli esnek atölye çizelgeleme problemlerinin sezgisel yöntemlerle modellenmesi: analizi ve çözümü, Doktora Tezi, Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü, Kayseri, 2004, 146511.
- [90] Rahimi-Vahed A. R., Mirghorbani S. M., Rabbani M., A hybrid multi-objective particle swarm algorithm for a mixed-model assembly line sequencing problem, *Engineering Optimization*, 2007, **39**, 877-898.
- [91] Oktay S., Engin O., Scatter search method for solving industrial problems: literature survey, *Sigma Journal of Engineering and Natural Sciences*, 2006, **3**, 144-156.
- [92] Öztürk R. E., Esnek akış tipi ve çok işlemcili esnek akış tipi çizelgeleme problemlerinin paralel doyumsuz algoritma ile çözümü, Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2007, 199739.
- [93] Bauer A., Bullnheimer B., Hartl R., Strauss C., An Ant colony optimization approach for the single machine total tardiness problem, *Proceedings in the Evolutionary Computation*, 1999, **8**, 125-141.
- [94] Silva C. A., Sousa J. M., Runkler T. A., Palm R., Sa da Costa J. M., Scheduling in manufacturing systems using the ant colonies optimization algorithm, *Proceedings of 5th Portuguese conference on automatic control*, Portekiz, 5-7 September 2002.
- [95] Rajendran C., Ziegler H., Two ant-colony algorithms for minimizing total flowtime in permutation flowshops, *Computers & Industrial Engineering*, 2005, **48**, 789-797.
- [96] Li J. Q., Pan Q. K., Gao K. Z., Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems, *International Journal Advanced Manufacturing Technology*, 2011, **10**, 1-11.
- [97] Kennedy J., Eberhart R. C., Particle swarm optimization, *International Neural Networks*, 1995, **95**, 1942-1948.
- [98] Engelbrecht A. P., *Fundamentals of Computational Swarm Intelligence*, 2nd ed., John Wiley and Sons, England, 2005.
- [99] Shi Y., Eberhart R., Empirical study of particle swarm optimization, *Evolutionary Computation*, 1999, **3**, 1945-1950.
- [100] Robinson J., Rahmat-Samii Y., Particle swarm optimization in electromagnetics, *Transactions on Antennas and Propagation*, 2004, **52**, 397-407.
- [101] Alataş B., Kaotik haritalı parçacık sürü optimizasyonu algoritmaları geliştirme, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ, 2007, 232067.

- [102] Kennedy J., Eberhart R. C., A Discrete binary version of the particle swarm algorithm, *Systems, Man, and Cybernetics*, 1997, **2**, 4104-4109.
- [103] Kennedy J., Eberhart R. C., Shi Y., *Swarm intelligence*, 4th ed., CA:Morgan Kaufmann Publishers, San Francisco, 2001.
- [104] Clerc M., The Swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Evolutionary Computation*, 1999, **1**, 1951-1957.
- [105] Ratnaweera A., Halgamuge S. K., Watson C., Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient, *Transactions Evolutionary Computation*, 2004, **8**, 240-255.
- [106] Clerc M., Kennedy J., The Particle swarm explosion stability and convergence in a multidimensional complex space, *Transaction Evolutionary Computation*, 2002, **6**, 58-73.
- [107] Zhou J., Duan Z., Li Y., Deng J., Yu D., PSO-Based neural network optimization and its utilization in a boring machine, *Journal of Materials Processing Technology*, 2006, **178**, 19-23.
- [108] Vandenberg F., Engelbrecht A., A new locally convergent particle swarm optimizer, *System, Man and Cybernetics*, 2002, **3**, 96-101.
- [109] Peram T., Vesramohanani K., Mohan C. K., Fitness- distance-ratio based particle swarm optimization, *Swarm Intelligence Symposium SIS2003*, Indiana, USA, 24-26 April 2003.
- [110] Mendes R., Kennedy J., Neves J., The fully informed particle swarm: Simpler, maybe better, *Transaction on Evolutionary Computation*, 2004, **8**, 204-210.
- [111] Parsopoulos K. E., Vrahatis M. N., UPSO-A unified particle swarm optimization scheme, *Lecture Series on Computational Sciences*, 2004, **1**, 868-873.
- [112] Kennedy J., Mendes R., Population structures and particle swarm performance, *Congress on Evolutionary Computation (CEC. 2002)*, Honolulu, Hawaii, May 2002.
- [113] Shi Y., Eberhart R. C., A modified particle swarm optimizer, *Proceedings of the International Conference on Evolutionary Computation*, Anchorage, USA, 4-9 May 1998.
- [114] Şevkli M., Yenisey M. M., Atölye tipi çizelgeleme problemleri için parçacık sürü optimizasyonu yöntemi, *itü dergisi/d mühendislik*, 2006, **5**, 58-68.
- [115] Taşgetiren M. F., Liang Y. C., Şevkli M., Gençyılmaz G., A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem, *European Journal of Operational Research*, 2007, **177**, 1930-1947.

- [116] Ray T., Liew K., A Swarm metaphor for multiobjective design optimization, *Engineering Optimization*, 2002, **34**, 141-153.
- [117] Pulido T. G., Coello Coello C. A., Using clustering techniques to improve the performance of a particle swarm optimizer, *GECCO 2004 Proceedings of the Genetic and Evolutionary Computation Conference*, Washington, USA, 26-30 June 2004.
- [118] Srinivasan D., Seow T. H., Particle swarm inspired evolutionary algorithm (PS-EA) for multicriteria optimization problems, Editors: Abraham A., Jain L., Goldberg R., *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*, 1st ed., Springer-Verlag, London, 147-165, 2005.
- [119] Raquel C. R., Naval P. C., An Effective use of crowding distance in multiobjective particle swarm optimization, *Genetic and Evolutionary Computation Conference (GECCO'2005)*, Washington, USA, 25-29 June 2005.
- [120] Moslehi G., Mahnam M., A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *International Journal Production Economics*, 2011, **129**, 14-22.
- [121] Zhang G., Shao X., Li P., Gao L., An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, *Computers & Industrial Engineering*, 2009, **56**, 1309-1318.
- [122] Jia Z., Chen H., Tang J., An Improved particle swarm optimization for multi-objective flexible job-shop scheduling problem, *International Conference on Grey Systems and Intelligent Services*, Nanjing, China, 18-20 November 2007.
- [123] Grobler J., Engelbrecht A. P., Kok S., Yadavalli S., Metaheuristics for the multi-objective FJSP with sequence-dependent set-up times, auxiliary resources and machine down time, *Annual Operation Research*, 2010, **180**, 165-196.
- [124] Liu H., Abraham A., Grosan C., A Novel variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems, *Digital Information Management*, 2007, **1**, 1-8.
- [125] Ergül E. U., Çok amaçlı genetik algoritmalar: temelleri ve uygulamaları, Doktora Tezi, Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun, 2010, 259241.
- [126] Zitzler E., Laumanns M., Thiele L., SPEA2: Improving the strength pareto evolutionary algorithm, *Swiss Federal Institute of Technology*, TIK-Report 103, 1-21, 2001.

- [127] Chiang T. C., Lin H. J., A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling, *International Journal of Production Economics*, 2012, **3**, 1-12.
- [128] Li J., Pan Q., Xie S., An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems, *Applied Mathematics and Computation*, 2012, **218**, 9353-9371.
- [129] Wang L., Zhou G., Xu Y., Liu M., An enhanced pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling, *International Journal Advanced Manufacturing Technology*, 2012, **60**, 1111-1123.
- [130] Davarzani Z., Akbarzadeh M., Khairdoost N., Multiobjective artificial immune algorithm for flexible job shop scheduling problem, *International Journal of Hybrid Information Technology*, 2012, **5**, 75-88.
- [131] Xiong J., Tan X., Yang K., Xing L., Chen Y., A Hybrid multiobjective evolutionary approach for flexible job-shop scheduling problems, *Mathematical Problems in Engineering*, 2012, **2012**, 1-27.
- [132] Bagheri A., Zandieh M., Mahdavi I., Yazdani M., An artificial immune algorithm for the flexible job-shop scheduling problem, *Future Generation Computer Systems*, 2010, **26**, 533-541.
- [133] Rajkumar M., Asokan P., Vamsikrishna V., A GRASP algorithm for flexible job-shop scheduling with maintenance constraints, *International Journal of Production Research*, 2010, **48**, 6821-6836.
- [134] Wang S., Yu J., An effective heuristic for flexible job-shop scheduling problem with maintenance activities, *Computers & Industrial Engineering*, 2010, **59**, 436-447.
- [135] Xing L. N., Chen Y. W., Yang K. W., Multi-objective flexible job shop Schedule: design and evaluation by simulation modeling, *Applied Soft Computing*, 2009, **9**, 362-376.
- [136] Xing L. N., Chen Y. W., Yang K. W., An efficient search method for multi-objective flexible job shop scheduling problems, *Journal Intelligent Manufacturing*, 2009, **20**, 283-293.
- [137] Zhang H., Gen M., Multistage-based genetic algorithm for flexible job-shop scheduling problem, *International Journal of Complexity*, 2005, **11**, 223-232.
- [138] Ho N. B., Tay J. C., Solving multiple-objective flexible job shop problems by evolution and local search, *Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, 2008, **38**, 674-685.
- [139] Shi-Jin W., Bing-Hai Z., Li-Feng X., A filtered-beam-search-based heuristic algorithm for flexible job-shop scheduling problem, *International Journal of Production Research*, 2008, **46**, 3027-3058.

- [140] Vilcot G., Billaut J. C., A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem, *European Journal of Operational Research*, 2008, **190**, 398-411.
- [141] Baykasoglu A., Özbakır L., A grammatical optimization approach for integrated process planning and scheduling, *Journal Intelligent Manufacturing*, 2009, **20**, 211-221.
- [142] Fattahi P., Fallahi A., Dynamic Scheduling in flexible job shop systems by considering simultaneously efficiency and stability, *CIRP Journal of Manufacturing Science and Technology*, 2010, **2**, 114-123.
- [143] Jian-jun Y., Xu-jun X., Fei Y., Study on multi-objective flexible production scheduling based on improved immune algorithm, *15th International Conference on Management Science & Engineering*, Long Beach, USA, 10-12 September 2008.
- [144] Wu Z., Multi-Agent workload control and flexible job shop scheduling, Ph. D. Thesis, University of South Florida, College of Engineering, South Florida, 2005, 3188440.
- [145] Paulli J., A hierarchical approach for the FMS scheduling problem, *European Journal of Operational Research*, 1995, **86**, 32-42.
- [146] Barnes J. W., Chambers J. B., Flexible job shop scheduling by tabu search, *University of Texas, Graduate program in operations research and industrial engineering*, Technical Report ORP 9609, 1-10, 1996.
- [147] Vaessens R. J. M., Aarts E. H. L., Lenstra J. K., Job shop scheduling by local search, *Inform Journal on Computing*, 1996, **8**(3), 302-317.
- [148] Chen H., Ihlow J., Lehmann C., A genetic algorithm for flexible Job-shop scheduling, *International Conference on Robotics and Automation*, Detroit, USA, 10-15 May 1999.
- [149] Jia H. Z., Nee A. Y. C., Fuh J. Y. H., Zhang Y. F., A modified genetic algorithm for distributed scheduling problems, *International Journal of Intelligent Manufacturing*, 2003, **14**, 351-362.
- [150] Ho N. B., Tay J. C., GENACE: an efficient cultural algorithm for solving the flexible job-shop problem, *Congress On Evolutionary Computation CEC2004*, Portland, USA, 19-23 June 2004.
- [151] Aarts E., Lenstra J. K., *Local Search in Combinatorial Optimization*, 4th ed., Wiley & Sons, England, 1997.
- [152] Güler A., Tamsayılı programlama problemleri için garanti değerli algoritmalar, Yüksek Lisans Tezi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, 2008, 215946.

- [153] Van Laarhoven P. J. M., Aarts E. H. L., Lenstra J. K., Job shop scheduling by simulated annealing, *Operational Research*, 1992, **40**, 113-125.
- [154] Zhang C. Y., Li P. G., Guan Z. L., Rao Y. Q., A Tabu search algorithm with a new neighborhood structure for the job shop scheduling problem, *Computers & Operations Research*, 2007, **34**, 3229-3242.
- [155] Barnes J. W., Chambers J. B., Solving the job shop scheduling problem using tabu search, *Transactions on Systems Man and Cybernetics*, 1995, **27**, 257-263.
- [156] Dell'Amico M., Trubian M., Applying tabu search to job-shop scheduling problem, *Annals of Operations Research*, 1993, **41**, 231-252.
- [157] Varlı A., Çok amaçlı ve çok konumlu aktarmalı taşıma problemlerinin genetik algoritma ile optimizasyonu, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007, 201376.
- [158] Baykasoğlu A., Özbakır L., Çizelgeleme problemleri için teslim tarihi belirlemede bir çözüm tekniği olarak genetik programlama, *III. Ulusal Üretim Araştırmaları Sempozyumu*, İstanbul, Türkiye, 19-20 Nisan 2003.
- [159] Ragatz G., Mabert V. A., A Simulation analysis of due date assignment rules, *Journal of Operations Management*, 1984, **5**, 27-39.
- [160] Chang F. R., A Study of factors affecting due-date predictability in a simulated dynamic job shop, *Journal of Manufacturing Systems*, 1997, **13**, 393-400.
- [161] Baykasoğlu A., Özbakır L., Sönmez A., Determining optimal due dates in earliness-tardiness flexible job shop scheduling, *Proceedings of 2nd international Conference on responsive manufacturing*, Gaziantep, Turkey, 26-28 June 2002.

EKLER

EK-A

Literatürde yer alan problem setlerine ait işlem süreleri ve hesaplanan teslim tarihleri EK-A kısmında verilmiştir.

Tablo A.1. 4x5x12 Problemine ait veriler

İşler	Operasyonlar	Makineler					d _i
		M1	M2	M3	M4	M5	
J ₁	O ₁₁	2	5	4	1	2	5
	O ₁₂	5	4	5	7	5	
	O ₁₃	4	5	5	4	5	
J ₂	O ₂₁	2	5	4	7	8	11
	O ₂₂	5	6	9	8	5	
	O ₂₃	4	5	4	54	5	
J ₃	O ₃₁	9	8	6	7	9	6
	O ₃₂	6	1	2	5	4	
	O ₃₃	2	5	4	2	4	
	O ₃₄	4	5	2	1	5	
J ₄	O ₄₁	1	5	2	4	12	4
	O ₄₂	5	1	2	1	2	

Tablo A.2. 8x8x27 Problemine ait verileri

İşler	Operasyonlar	Makineler								d _i
		M1	M2	M3	M4	M5	M6	M7	M8	
J ₁	O ₁₁	5	3	5	3	3	-	10	9	8
	O ₁₂	10	-	5	8	3	9	9	6	
	O ₁₃	-	10	-	5	6	2	4	5	
J ₂	O ₂₁	5	7	3	9	8	-	9	-	9
	O ₂₂	-	8	5	2	6	7	10	9	
	O ₂₃		10		5	6	4	1	7	
	O ₂₄	10	8	9	6	4	7	-	-	
J ₃	O ₃₁	10	-	-	7	6	5	2	4	8
	O ₃₂	-	10	6	4	8	9	10	-	
	O ₃₃	1	4	5	6	-	10	-	7	
J ₄	O ₄₁	3	1	6	5	9	7	8	4	9
	O ₄₂	12	11	7	8	10	5	6	9	
	O ₄₃	4	6	2	10	3	9	5	7	
J ₅	O ₅₁	3	6	7	8	9	-	10	-	9
	O ₅₂	10	-	7	4	9	8	6	-	
	O ₅₃	-	9	8	7	4	2	7	-	
	O ₅₄	11	9	-	6	7	5	3	6	
J ₆	O ₆₁	6	7	1	4	6	9	-	10	10
	O ₆₂	11		9	9	9	7	6	4	
	O ₆₃	10	5	9	10	11	-	10	-	
J ₇	O ₇₁	5	4	2	6	7	-	10	-	9
	O ₇₂	-	9	-	9	11	9	10	5	
	O ₇₃	-	8	9	3	8	6	-	10	
J ₈	O ₈₁	2	8	5	9	-	4	-	10	8
	O ₈₂	7	4	7	8	9	-	10	-	
	O ₈₃	9	9	-	8	5	6	7	1	
	O ₈₄	9	-	3	7	1	5	8	-	

Tablo A.3. 10x7x29 Problemine ait veriler

İşler	Operasyonlar	Makineler							d _i
		M1	M2	M3	M4	M5	M6	M7	
J ₁	O ₁₁	1	4	6	9	3	5	2	7
	O ₁₂	8	9	5	4	1	1	3	
	O ₁₃	4	8	10	4	11	4	3	
J ₂	O ₂₁	6	9	8	6	5	10	3	9
	O ₂₂	2	10	4	5	9	8	4	
J ₃	O ₃₁	15	4	8	4	8	7	1	9
	O ₃₂	9	6	1	10	7	1	6	
	O ₃₃	11	2	7	5	2	3	14	
J ₄	O ₄₁	2	8	5	8	9	4	3	7
	O ₄₂	5	3	8	1	9	3	6	
	O ₄₃	1	2	6	4	1	7	2	
J ₅	O ₅₁	7	1	8	5	4	3	9	7
	O ₅₂	2	4	5	10	6	4	9	
	O ₅₃	5	1	7	1	6	6	2	
J ₆	O ₆₁	8	7	4	56	9	8	4	12
	O ₆₂	5	14	1	9	6	5	8	
	O ₆₃	3	5	2	5	4	5	7	
J ₇	O ₇₁	5	6	3	6	5	15	2	8
	O ₇₂	6	5	4	9	5	4	3	
	O ₇₃	9	8	2	8	6	1	7	
J ₈	O ₈₁	6	1	4	1	10	4	3	9
	O ₈₂	11	13	9	8	9	10	8	
	O ₈₃	4	2	7	8	3	10	7	
J ₉	O ₉₁	12	5	4	5	4	5	5	14
	O ₉₂	4	2	15	99	4	7	3	
	O ₉₃	9	5	11	2	5	4	2	
J ₁₀	O ₁₀₁	9	4	13	10	7	6	8	10
	O ₁₀₂	4	3	25	3	8	1	2	
	O ₁₀₃	1	2	6	11	13	3	5	

Tablo A.4. 10x10x30 Problemine ait veriler

İşler	Operasyonlar	Makineler										d _i
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
J ₁	O ₁₁	1	4	6	9	3	5	2	8	9	5	7
	O ₁₂	4	1	1	3	4	8	10	4	11	4	
	O ₁₃	3	2	5	1	5	6	9	5	10	3	
J ₂	O ₂₁	2	10	4	5	9	8	4	15	8	4	8
	O ₂₂	4	8	7	1	9	6	1	10	7	1	
	O ₂₃	6	11	2	7	5	3	5	14	9	2	
J ₃	O ₃₁	8	5	8	9	4	3	5	3	8	1	6
	O ₃₂	9	3	6	1	2	6	4	1	7	2	
	O ₃₃	7	1	8	5	4	9	1	2	3	4	
J ₄	O ₄₁	5	10	6	4	9	5	1	7	1	6	7
	O ₄₂	4	2	3	8	7	4	6	9	8	4	
	O ₄₃	7	3	12	1	6	5	8	3	5	2	
J ₅	O ₅₁	7	10	4	5	6	3	5	15	2	6	8
	O ₅₂	5	6	3	9	8	2	8	6	1	7	
	O ₅₃	6	1	4	1	10	4	3	11	13	9	
J ₆	O ₆₁	8	9	10	8	4	2	7	8	3	10	8
	O ₆₂	7	3	12	5	4	3	6	9	2	15	
	O ₆₃	4	7	3	6	3	4	1	5	1	11	
J ₇	O ₇₁	1	7	8	3	4	9	4	13	10	7	7
	O ₇₂	3	8	1	2	3	6	11	2	13	3	
	O ₇₃	5	4	2	1	2	1	8	14	5	7	
J ₈	O ₈₁	5	7	11	3	2	9	8	5	12	8	9
	O ₈₂	8	3	10	7	5	13	4	6	8	4	
	O ₈₃	6	2	13	5	4	3	5	7	9	5	
J ₉	O ₉₁	3	9	1	3	8	1	6	7	5	4	7
	O ₉₂	4	6	2	5	7	3	1	9	6	7	
	O ₉₃	8	5	4	8	6	1	2	3	10	12	
J ₁₀	O ₁₀₁	4	3	1	6	7	1	2	6	20	6	8
	O ₁₀₂	3	1	8	1	9	4	1	4	17	15	
	O ₁₀₃	9	2	4	2	3	5	2	4	10	23	

Tablo A.5. 15x10x56 Problemine ait veriler

İşler	Operasyonlar	Makineler										d _i
		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
J ₁	O ₁₁	1	4	6	9	3	5	2	8	9	4	8
	O ₁₂	1	1	3	4	8	10	4	11	4	3	
	O ₁₃	2	5	1	5	6	9	5	10	3	2	
	O ₁₄	10	4	5	9	8	4	15	8	4	4	
J ₂	O ₂₁	4	8	7	1	9	6	1	10	7	1	8
	O ₂₂	6	11	2	7	5	3	5	14	9	2	
	O ₂₃	8	5	8	9	4	3	5	3	8	1	
	O ₂₄	9	3	6	1	2	6	4	1	7	2	
J ₃	O ₃₁	7	1	8	5	4	9	1	2	3	4	7
	O ₃₂	5	10	6	4	9	5	1	7	1	6	
	O ₃₃	4	2	3	8	7	4	6	9	8	4	
	O ₃₄	7	3	12	1	6	5	8	3	5	2	
J ₄	O ₄₁	6	2	5	4	1	2	3	6	5	4	8
	O ₄₂	8	5	7	4	1	2	36	5	8	5	
	O ₄₃	9	6	2	4	5	1	3	6	5	2	
	O ₄₄	11	4	5	6	2	7	5	4	2	1	
J ₅	O ₅₁	6	9	2	3	5	8	7	4	1	2	8
	O ₅₂	5	4	6	3	5	2	28	7	4	5	
	O ₅₃	6	2	4	3	6	5	2	4	7	9	
	O ₅₄	6	5	4	2	3	2	5	4	7	5	
J ₆	O ₆₁	4	1	3	2	6	9	8	5	4	2	7
	O ₆₂	1	3	6	5	4	7	5	4	6	5	
J ₇	O ₇₁	1	4	2	3	3	6	9	8	5	4	6
	O ₇₂	2	1	4	5	2	3	5	4	2	5	
J ₈	O ₈₁	2	3	6	2	5	4	1	5	8	7	9
	O ₈₂	4	5	6	2	3	5	4	1	2	5	
	O ₈₃	3	5	4	2	5	49	8	5	4	5	
	O ₈₄	1	2	36	5	2	3	6	4	11	2	
J ₉	O ₉₁	6	3	2	22	44	11	10	23	5	1	16
	O ₉₂	2	3	2	12	15	10	12	14	18	16	
	O ₉₃	20	17	12	5	9	6	4	7	5	6	
	O ₉₄	9	8	7	4	5	8	7	4	56	2	
J ₁₀	O ₁₀₁	5	8	7	4	56	3	2	5	4	1	8
	O ₁₀₂	2	5	6	9	8	5	4	2	5	4	
	O ₁₀₃	6	3	2	5	4	7	4	5	2	1	
	O ₁₀₄	3	2	5	6	5	8	7	4	5	2	
J ₁₁	O ₁₁₁	1	2	3	6	5	2	1	4	2	1	8
	O ₁₁₂	2	3	6	3	2	1	4	10	12	1	
	O ₁₁₃	3	6	2	5	8	4	6	3	2	5	
	O ₁₁₄	4	1	45	6	2	4	1	25	2	4	
J ₁₂	O ₁₂₁	9	8	5	6	3	6	5	2	4	2	13
	O ₁₂₂	5	8	9	5	4	75	63	6	5	21	
	O ₁₂₃	12	5	4	6	3	2	5	4	2	5	
	O ₁₂₄	8	7	9	5	6	3	2	5	8	4	
J ₁₃	O ₁₃₁	4	2	5	6	8	5	6	4	6	2	7
	O ₁₃₂	3	5	4	7	5	8	6	6	3	2	
	O ₁₃₃	5	4	5	8	5	4	6	5	4	2	
	O ₁₃₄	3	2	5	6	5	4	8	5	6	4	
J ₁₄	O ₁₄₁	2	3	5	4	6	5	4	85	4	5	10
	O ₁₄₂	6	2	4	5	8	6	5	4	2	6	
	O ₁₄₃	3	25	4	8	5	6	3	2	5	4	
	O ₁₄₄	8	5	6	4	2	3	6	8	5	4	
J ₁₅	O ₁₅₁	2	5	6	8	5	6	3	2	5	4	7
	O ₁₅₂	5	6	2	5	4	2	5	3	2	5	
	O ₁₅₃	4	5	2	3	5	2	8	4	7	5	
	O ₁₅₄	6	2	11	14	2	3	6	5	4	8	

KİŞİSEL YAYIN VE ESERLER

- [1] **Kaya S.**, Fırlalı N., Çok amaçlı esnek atölye tipi çizelgeleme problemleri, *Sigma Mühendislik ve Fen Bilimleri Dergisi*, 2013, **31**, 605-623.
- [2] **Kaya S.**, Akgöbek Ö., Engin O., Scatter search metaheuristic for solving the flow shop scheduling with earliness and tardiness penalties, *NWSA e-Journal of New World Science Academy*, 2011, **6**(1), 415- 420.
- [3] Akgöbek Ö., **Kaya S.**, Değirmenci Ü., Engin O., Parallel greedy metaheuristic algorithm for solving open shop scheduling problem, *NWSA e-Journal of New World Science Academy*, 2011, **6**(1), 421- 427.
- [4] Akgöbek Ö., **Kaya S.**, Knowledge discovery from data sets through data mining techniques: application to medical data mining, *NWSA e-Journal of New World Science Academy*, 2011, **6**(1), 237-245.
- [5] **Kaya S.**, Akgöbek Ö., Engin O., Erken ve geç cezalı permütasyon akış tipi çizelgeleme problemlerinin dağınık arama metasezgiseli ile çözümü, *Bilimde Modern Yöntemler Sempozyumu*, Diyarbakır, Türkiye, 14-16 Ekim 2010.
- [6] Akgöbek Ö., **Kaya S.**, Veri madenciliği teknikleri ile veri kümelerinden bilgi keşfi: medikal veri madenciliği uygulaması, *Bilimde Modern Yöntemler Sempozyumu*, Diyarbakır, Türkiye, 14-16 Ekim 2010.
- [7] Akgöbek Ö., **Kaya S.**, Değirmenci Ü., Engin O., Açık atölye tipi çizelgeleme problemlerinin paralel doyumsuz metasezgisel algoritma ile çözümü, *Bilimde Modern Yöntemler Sempozyumu*, Diyarbakır, Türkiye, 14-16 Ekim 2010.
- [8] **Kaya S.**, Engin O., Sabit iş çizelgeleme problemleri: literatür araştırması ve metasezgisel yöntemler için çözüm önerisi, *itü dergisi/d mühendislik*, 2009, **8**(1), 37-47.
- [9] Akgöbek Ö., **Kaya S.**, Doğrudan kural üreten endüktif öğrenme algoritmalarının karakteristik seçim yöntemleri ve yeni bir yaklaşım, *YA/EM 2006, Yöneylem Araştırması/Endüstri Mühendisliği XXVI. Ulusal Kongresi*, Kocaeli, Türkiye, 3-5 Temmuz 2006.
- [10] Akgöbek Ö., **Kaya S.**, Bir bilgi tabanlı ana üretim programlama modülünün tasarlanması, *YA/EM 2006, Yöneylem Araştırması/Endüstri Mühendisliği XXVI. Ulusal Kongresi*, Kocaeli, Türkiye, 3-5 Temmuz 2006.
- [11] **Kaya S.**, Akgöbek Ö., Bir tekstil işletmesinde başa baş noktasının belirlenmesi için bir uygulama, *GAP V. Mühendislik Kongresi*, Şanlıurfa, Türkiye, 26-28 Nisan 2006.

- [12] **Kaya S.**, Akgöbek Ö., Az gelişmiş bölgelerde üretim tesislerinin etkin kullanımı, *GAP V. Mühendislik Kongresi*, Şanlıurfa, Türkiye, 26-28 Nisan 2006.
- [13] Akgöbek Ö., **Kaya S.**, Talep tahminlerinin belirlenmesinde yapay sinir ağlarının kullanılması ve bir uygulama, *YA/EM2005-Yöneylem Araştırması Ve Endüstri Mühendisliği 25. Ulusal Kongresi*, İstanbul, Türkiye, 4-6 Temmuz 2005.

ÖZGEÇMİŞ

1974 Yılında Şanlıurfa'da doğdu. İlk ve orta öğrenimini Şanlıurfa'da tamamladı. 1992 Yılında Selçuk Üniversitesi Mühendislik Mimarlık Fakültesi, Endüstri Mühendisliği bölümüne girmeye hak kazandı. 1996 Yılında Endüstri Mühendisi olarak mezun oldu. 2007 Yılında Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim dalında yüksek lisans öğrenimini tamamladı. 2008 Yılından itibaren Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim dalında doktora eğitimine başladı. 2001 Yılından beri Harran Üniversitesi, Şanlıurfa Teknik Bilimler Meslek Yüksekokulu, Motorlu araçlar ve ulaştırma teknolojileri bölümünde öğretim görevliliği yanı sıra, Harran Üniversitesi Makine Fabrikasında, Fabrika sorumlu yardımcısı olarak çalışmaktadır.