

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNDE
GENETİK ALGORİTMA UYGULAMASI**

Endüstri Müh. Mehmet Tufan KÖREZ

**FBE Endüstri Mühendisliği Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Yrd. Doç. Dr. Semih ÖNÜT

İSTANBUL, 2005

İÇİNDEKİLER

	Sayfa
KISALTMA LİSTESİ	iv
ŞEKİL LİSTESİ	v
ÇİZELGE LİSTESİ	vi
ÖNSÖZ	vii
ÖZET	viii
ABSTRACT	ix
1. GİRİŞ.....	1
2. AKIŞ TİPİ ÇİZELGELEME	3
2.1 Çizelgelemede Kullanılan Öncelik Kuralları ve Kabuller.....	5
2.2 Johnson Algoritması	6
2.3 Kombinatoriyel Optimizasyon	8
2.4 Tabu Araştırmaları.....	11
2.5 Tavlama Benzetimi.....	11
2.6 Karınca Kolonileri Optimizasyonu.....	14
2.7 Yapay Bağışıklık Sistemleri	15
3. GENETİK ALGORİTMALAR.....	17
3.1 Parametre Kodlama	19
3.1.1 Operasyona Dayalı Kodlama	21
3.1.2 İşe Dayalı Kodlama	23
3.1.3 Liste Önceliğine Dayalı Kodlama	24
3.1.4 İş Çifti İlişkisine Dayalı Kodlama	27
3.1.5 Öncelik Kuralına Dayalı Kodlama	28
3.1.6 Tamamlanma Zamanına Dayalı Kodlama	31
3.1.7 Makineye Dayalı Kodlama	31
3.1.8 Rassal Anahtarlı Kodlama	32
3.1.9 Karışık Kodlama.....	32
3.2 Genetik Algoritmalarda Kullanılan Operatörler.....	33
3.2.1 Üreme(Seçim) Operatörü	33
3.2.1.1 Makine Verimlerine Bağlı Rulet Çemberi Yöntemi.....	33
3.2.1.3 Yapay Seçim Yöntemi.....	35
3.2.1.4 Kısmi Yapay Seçim Yöntemi	35
3.2.2 Çaprazlama Operatörü.....	36
3.2.2.1 Pozisyona Dayalı Çaprazlama	36
3.2.2.2 Sıraya Dayalı Çaprazlama	37
3.2.2.3 Kısmi Planlı Çaprazlama(PMX).....	38
3.2.2.4 Dairesel Çaprazlama(CX).....	38
3.2.2.5 Doğrusal Sıralı Çaprazlama(LOX).....	39
3.2.2.6 Sıralı Çaprazlama(OX).....	40
3.2.2.7 Alt Değişimli Çaprazlama(SXX)	40

3.2.2.8	İş Tabanlı Sıralı Çaprazlama	41
3.2.3	Mutasyon Operatörü	41
3.3	Genetik Algoritma'nın Gelişimi ve Kullanım Alanları	44
3.3.1	Genetik Algoritma'nın Tarihsel Gelişimi	44
3.3.2	Çok Amaçlı Optimizasyon	44
3.3.3	Çizelgeleme Problemleri	46
4.	SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ	48
4.1	Sıralı Problemlerin Uygulama Temelleri	48
4.2	Sıralı Akış Tipi Çizelgeleme Problemi Çeşitleri	49
4.2.1	Genel Durum Problemi	49
4.2.2	Özel Durum Problemi	52
5.	SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ ÜZERİNE GENETİK ALGORİTMA UYGULAMASI	54
5.1	Uygulamanın Kapsam ve Yapısı	54
5.2	Uygulama İçin Kullanılan Model	57
5.2.1	Yapının Gösterimi	58
5.2.2	Başlangıç Popülasyonu ve Popülasyon Büyüklüğü	59
5.2.3	Popülasyondaki Bireylerin Uygunluk Değerleri	60
5.2.4	Bireylerin Seçimi	60
5.2.5	Genetik Operatörler	61
5.2.6	Tamamlanma Kriteri	61
5.3	Genel Durum Problemi Üzerine Uygulama	63
5.4	Özel Durum Problemi Üzerine Uygulama	65
5.5	Sonuçlar ve Değerlendirme	66
6.	SONUÇLAR ve ÖNERİLER	69
	KAYNAKLAR	71
	EKLER	74
	EK-A Genel Durum Problemleri İşlem Süreleri	75
	EK-B Özel Durum Problemleri İşlem Süreleri	80
	ÖZGEÇMİŞ	85

KISALTMA LİSTESİ

AS	Ant Systems (Karıncı Sistemleri)
CDS	Campbell Dudek Smith
CX	Cycle Crossover (Dairesel Çaprazlama)
FIFO	First In First Out (İlk Giren İlk Çıkar)
GA	Genetic Algorithm (Genetik Algoritma)
LOX	Linear Order Crossover (Doğrusal Sıralı Çaprazlama)
LPT	Longest Processing Time (Uzun İşlem Süreli)
LWR	Least Work Remaining (En Az İş Kalan)
MVRÇ	Makine Verimli Rulet çemberi
MWR	Most Work Remaining (En Çok İş Kalan)
NP	Non Polynomial Optimization (Polinomial Olmayan Optimizasyon)
OX	Order Crossover (Sıralı Çaprazlama)
PBX	Position Based Crossover (Pozisyon Tabanlı Çaprazlama)
SA	Simulated Annealing (Tavlama Benzetimi)
SPD	Smith, Panwalkar, Dudek
SPT	Shortest Processing Time (Kısa İşlem Süreli)
TS	Tabu Search (Tabu Araştırmaları)
TSP	Traveling Salesman Problem (Gezgin Satıcı Problemi)
VEGA	Vector Evaluated Genetic Algorithm (Vektör Değerli GA)

ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1 Johnson algoritması ile (2,4,5,3,1) numaralı işlerin gantt diyagramı	8
Şekil 3.1 Kod ve çözüm uzayı	19
Şekil 3.2 Kodlamada uygun olma ve yasaklı olma durumları	20
Şekil 3.3 Şifresi çözülmüş aktif çizelge	22
Şekil 3.4a İşe dayalı kodlama sistemi	23
Şekil 3.4b İşe dayalı kodlama sistemi	23
Şekil 3.4c İşe dayalı kodlama sistemi	24
Şekil 3.5a Liste önceliğine dayalı kodlama	25
Şekil 3.5b Liste önceliğine dayalı kodlama	25
Şekil 3.5c Liste önceliğine dayalı kodlama	26
Şekil 3.5d Liste önceliğine dayalı kodlama	26
Şekil 3.5e Liste önceliğine dayalı kodlama	26
Şekil 3.5f Liste önceliğine dayalı kodlama	27
Şekil 3.6a Öncelik kuralına dayalı kodlama	30
Şekil 3.6b Öncelik kuralına dayalı kodlama	30
Şekil 3.6c Öncelik kuralına dayalı kodlama	30
Şekil 3.6d Öncelik kuralına dayalı kodlama	31
Şekil 3.7 Pozisyona dayalı çaprazlama	37
Şekil 3.8 Sıraya dayalı çaprazlama	37
Şekil 3.9a Farklı mutasyon operatörleri	43
Şekil 3.9b Farklı mutasyon operatörleri	43
Şekil 3.9c Farklı mutasyon operatörleri	43
Şekil 3.9d Farklı mutasyon operatörleri	43
Şekil 4.1 Geliştirilen çözümün grafik gösterimi	51
Şekil 5.1 Temel genetik algoritma	56

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 Örnek 2.1 için işlem süreleri	7
Çizelge 2.2 Termodinamik simülasyonu ve kombinatoriyel optimizasyon benzerlikleri	12
Çizelge 3.1 İş sıralama problemi, 3-iş ve 3-makine örneği	22
Çizelge 3.2 Üç makine ve üç iş problemi	28
Çizelge 3.3 Seçilmiş öncelik kuralları	29
Çizelge 3.4 İkili sistemde dizi kodlama	33
Çizelge 3.5 Örnek 3.1 için varsayılan başlangıç popülasyonu	34
Çizelge 3.6 Örnek 3.1 için manuel rulet çemberi yöntemi	34
Çizelge 4.14-iş 4-makine problemi için sıralı matris	50
Çizelge 4. 2-iş sıralı matris problemi	52
Çizelge 4.3 Çizelge 4.2'deki problem için genel sıralama	52
Çizelge 4. Sıralı problem işlem zamanları	53
Çizelge 5.1 Genel durum problemi uygulama sonuçları	64
Çizelge 5.2 Genel durum problemi performans karşılaştırması	64
Çizelge 5.3 Özel durum problemi uygulama sonuçları	65
Çizelge 5.4 Özel durum problemi performans karşılaştırması	66
Çizelge A.1 Genel durum, 7-iş, 4-makine problemi için işlem zamanları	75
Çizelge A.2 Genel durum, 7-iş, 5-makine problemi için işlem zamanları	75
Çizelge A.3 Genel durum, 7-iş, 8-makine problemi için işlem zamanları	75
Çizelge A.4 Genel durum, 7-iş, 10-makine problemi için işlem zamanları	76
Çizelge A.5 Genel durum, 7-iş, 15-makine problemi için işlem zamanları	76
Çizelge A.6 Genel durum, 10-iş, 4-makine problemi için işlem zamanları	77
Çizelge A.7 Genel durum, 10-iş, 5-makine problemi için işlem zamanları	77
Çizelge A.8 Genel durum, 10-iş, 8-makine problemi için işlem zamanları	78
Çizelge A.9 Genel durum, 10-iş, 10-makine problemi için işlem zamanları	78
Çizelge A.10 Genel durum, 10-iş, 15-makine problemi için işlem zamanları	79
Çizelge B.1 Özel durum, 7-iş, 4-makine problemi için işlem zamanları	80
Çizelge B.2 Özel durum, 7-iş, 5-makine problemi için işlem zamanları	80
Çizelge B.3 Özel durum, 7-iş, 8-makine problemi için işlem zamanları	81
Çizelge B.4 Özel durum, 7-iş, 10-makine problemi için işlem zamanları	81
Çizelge B.5 Özel durum, 7-iş, 15-makine problemi için işlem zamanları	81
Çizelge B.6 Özel durum, 10-iş, 4-makine problemi için işlem zamanları	82
Çizelge B.7 Özel durum, 10-iş, 5-makine problemi için işlem zamanları	82
Çizelge B.8 Özel durum, 10-iş, 8-makine problemi için işlem zamanları	83
Çizelge B.9 Özel durum, 10-iş, 10-makine problemi için işlem zamanları	83
Çizelge B.10 Özel durum, 10-iş, 15-makine problemi için işlem zamanları	84

ÖNSÖZ

Son yıllarda teknolojideki hızlı gelişmeler sonucunda, geçmişte optimum çözümü bulunamayan NP kapsamındaki problemlerin, optimuma yakın çözümlerinin elde edilmesi sağlanmıştır.

Endüstrilerde, üretim planlamanın önemli bir bölümünü çizelgeleme problemleri oluşturmaktadır. Akış tipi çizelgeleme problemleri ise, bunların en sık karşılaşılan çeşididir. Tez kapsamında, akış tipi çizelgeleme problemlerinin bir alt gelişimi olan ‘sıralı akış tipi çizelgeleme problemleri’ ele alınmıştır.

Son yıllarda araştırmalarda sık kullanılan, stokastik arama özelliğine sahip güçlü bir sezgisel arama yöntemi olan Genetik Algoritma, sıralı akış tipi çizelgeleme problemlerinin çözümüne yeni bir yaklaşım getirmek amacı ile ele alınmıştır.

Bu çalışma kapsamında, bana bu konuyu ele alma şansı vererek beni yönlendiren ve çalışma boyunca yardımlarını esirgemeyen Sayın Yrd. Doç. Dr. Semih Önüt’e, sıralı akış tipi çizelgeleme problemleri konusunda materyalleri sağlayan Sayın Yrd. Doç. Dr. Serol Bulkan’a, başta genetik algoritma olmak üzere çalışma boyunca değerli fikirleriyle katkıda bulunan, Endüstri Yük. Müh. Bilgehan Bıyıklı’ya ve yardım eden herkese sonsuz teşekkürlerimi sunarım.

ÖZET

Tez kapsamında, NP problemleri grubunda olan akış tipi çizelgeleme problemlerinin bir alt gelişimi olan “sıralı akış tipi çizelgeleme problemleri” ele alınmış, problem için geliştirilen çözüm yöntemine alternatif olarak, Genetik Algoritma ile yeni bir çözüm yöntemi geliştirilmeye çalışılmıştır.

Çalışmanın ilk bölümünde, akış tipi çizelgeleme problemlerinin yapısı ve çizelgeleme problemlerinde kullanılan rassal arama metotlarından olan, tabu araştırmaları, tavlama benzetimi, karınca kolonileri ve yapay bağışıklık sistemlerinden bahsedilmiştir.

Genetik algoritmanın sıralı akış tipi çizelgeleme problemleri üzerindeki performansını değerlendirmek için yapılan uygulama, sıralı akış tipi çizelgeleme problemlerinin doğası gereği iki kısım olarak ele alınmıştır.

İlk kısımda, genel durum olarak adlandırılan, işlem zamanları matrisinde en büyük işlem zamanlarının ara makinelerden birinde oluşması durum ele alınmıştır. 10 farklı iş-makine kombinasyonu üzerine yapılan uygulamada, genetik algoritmanın Smith Panwalkar Dudek algoritmasına çok yakın sonuçlar verdiği, tavlama benzetimine kıyasla daha iyi olduğu gözlenmiştir.

İkinci kısım olan, en büyük işlem zamanlarının ilk yada son makinede oluşması durumu, özel durum problemlerinde ise tekrar genel durum probleminde ele alınan 10 farklı iş-makine kombinasyonu ele alınmış, burada da genetik algoritma Smith Panwalkar Dudek algoritmasına yakın sonuçlar vermiş, tavlama benzetimine göre daha iyi olduğu gözlenmiştir.

İki uygulama ele alındığında, genel durum problemleri için iyi bir çözümü Smith Panwalkar Dudek algoritmasına kıyasla çok daha kısa bir zamanda elde ettiğinden genetik algoritmayı kullanmak avantajlı olacak, ancak özel durum probleminde, optimum sonucu çok basit bir yöntemle, kısa zamanda ve az eforla garanti ettiğinden Smith Panwalkar Dudek algoritmasını tercih etmek daha doğru olacaktır.

Anahtar Kelimeler: Sıralı Akış Tipi Çizelgeleme, Genetik Algoritma

ABSTRACT

In this thesis, the problem of scheduling jobs in a flow-shop environment, which is a NP-complete problem is considered. Genetic algorithm is applied in order to find a new approach to the problem.

In the first part of the study, the structure of flow-shop problems and some related random search methods such as tabu search, simulated annealing, ant colonies and artificial immune systems are considered.

The application, which is made to determine the performance of genetic algorithm, on ordered flow-shop problems are considered as two different parts as the nature of the ordered type of flow-shop problems.

In the first part, the general case, which is the maximum processing time for every job occurred in intermediate machines, are considered. The application is made on 10 different job-machine combinations and as an outcome, genetic algorithm had close results, which were better than the results of simulated annealing, to Smith Panwalkar Dudek algorithm was observed.

In the second part, the maximum processing time for every job occurred on the first or last machine, which was the special case, are considered. Same job-machine combinations for general case are considered and again genetic algorithm had better result than simulated annealing where the results are close enough to Smith Panwalkar Dudek algorithm.

After two application, it was observed that in general case, genetic algorithm provides a reasonable and close optimal solution which less effort and time than Smith Panwalker and Dudek algorithm. But in the special case, Smith Panwalkar Dudek algorithm is much more simple and optimal solution guaranteed than genetic algorithm.

Keywords: Ordered Flow-Shop Problems, Genetic Algorithm

1. GİRİŞ

Yıllar içerisinde yapılan çalışmalar ile, çözümü zor olan optimizasyon problemlerinin- özellikle NP kapsamına giren optimizasyon problemlerinin- çözümünde çeşitli sezgisel arama metodları geliştirilmiştir. Genetik Algoritmalar(GA), Tavlama Benzetimi(SA), ve Tabu Araştırmaları(TS) bu metodlar içerisinde başarı ile uygulanan ve en popüler olan üç tanesidir. Genetik algoritmalar biyolojiden, tavlama benzetimi fizikten, tabu araştırmaları ise zeka tekniklerinden esinlenerek geliştirilmiş arama teknikleridir (Gen, 1996).

Genetik algoritma, ilk olarak 1975 yılında John Holland'ın "Doğal ve Yapay Sistemlerin Uyumu" isimli kitabı ile ortaya çıkmıştır. Genetik algoritmanın temel çalışma prensibi Darwin'in 'Doğal Seçim' ilkesine dayanır. Darwin, "Türlerin Kökeni" adlı yapıtında iki varsayımı ortaya atmıştır (Demirsoy, 1988):

1. Tüm organizmalar, gereğinden fazla yavru meydana getirme yeteneğine sahiptirler. Bununla beraber elemine edilenler ile popülasyonda denge sağlanmaktadır.
2. Bir tür içerisindeki bireyler, kalıtsal özellikleri bakımından farklıdırlar.

Genetik Algoritma doğadaki canlıların geçirdiği evrim sürecini ele alır. Amaç, doğal sistemlerin uyum sağlama özelliğini dikkate alarak, yapay sistemler tasarlamaktır. Genetik algoritmada tasarlanan yapay sistemde ele alınan en önemli faktör ise sağlamlıktır.

Yapay sistemler, doğal sistemler kadar sağlam olabilse, mevcut sistemler faaliyetlerini daha uzun zaman sürdürecekler ve pahalı olan yeniden tasarlama ve uyarlama işlemleri ortadan kalkacaktır.

Genetik algoritmalar konusundaki esas gelişim ise, John Holland'ın doktora öğrencisi David E. Goldberg tarafından 1985 yılında hazırlanan "Gaz Boru hatlarının Genetik Algoritma Kullanılarak Denetlenmesi" konulu tez ile sağlanmıştır. Bu ilk uygulamadan sonra Goldberg'in 1989 yılında yayımladığı "Makine Öğrenmesi, Arama ve Optimizasyon İçin Genetik Algoritma" adlı kitabı, genetik algoritmaya yeni bir boyut kazandırmış ve günümüzde dahi genetik algoritma konusunda en kapsamlı referans olma özelliğini korumuştur.

Öte yandan akış tipi çizelgeleme problemleri uzun yıllardır çeşitli araştırmacılara konu olmuş, ve hala içerdiği geniş çalışma olanakları ile de popüler bir alan olmaya devam etmektedir. Bu

konuda ilk alıřmalar Johnson tarafından yapılmıřtır. Ancak problem NP kapsamında olduėundan Palmer, Campbell, Dudek, Smith, Gupta gibi eřitli arařtırmacılar tarafından özümde sezgisel yöntemlere yönelinmiřtir.

Akıř tipi izelgeleme problemlerinin bir alt konusu olan ‘Sıralı Akıř Tipi izelgeleme Problemleri’ konusunda ilk alıřmalar Smith tarafından yapılmıř, ancak problemin tanımı ve kuralları Smith, Panwalkar ve Dudek tarafından 1975 yılında geliřtirilmiřtir.

Smith ve diėerleri geliřtirdikleri problem için iki farklı durum önermiřlerdir: ‘Genel Durum’ ve genel durumun bir geliřimi olan ‘Özel Durum’. Bu problemler için optimum sonu veren iki farklı yaklařım geliřtirmiřlerdir. Ancak yıllar içerisinde problem popüler olamamıř ve üzerine alıřma yapılmamıřtır.

Bu tez kapsamında, Sıralı Akıř Tipi izelgeleme Problemleri her iki durumu ile ele alınacak ve Genetik Algoritma kullanılarak yeni bir özüm geliřtirilmeye alıřılacaktır.

2. AKIŞ TİPİ ÇİZELGELEME

Pekçok üretim ve montaj ortamında bir dizi operasyonun bütün işler üzerinde yapılması gerekir. Bu, bütün işlerin aynı sırayla aynı operasyonlardan akmasını gerektirir. Bu durumdan doğan akış tipi çizelgeleme kavramı temelde, m makina ve n işin bulunduğu bir ortamda, her iş için m operasyonun bulunması ve bu operasyonların farklı makinalarda ve aynı sıra ile yapılmasıdır.

Akış tipi çizelgeleme problemlerinde amaç, toplam akış süresini ve en son işin tamamlanma zamanını minimize etmektir. Akış tipi çizelgeleme problemleri ile ilgili ilk çalışma Johnson (1954) tarafından yapılmıştır. Johnson iki makina n iş problemi için optimum çözüm veren bir algoritma geliştirmiştir. Daha sonraki çalışmalar $m > 2$ makina için yapılmıştır. Ancak bu tip problemler NP kapsamına girdiğinden, bu problemlerin çözümü için çeşitli sezgisel yöntemler geliştirilmiştir (Chen vd., 1995):

1.Palmer'in Eğim Dizisi Yöntemi(1965) : Bu yöntemde her işe bir dizin değeri tanımlanır, bu dizin değerine göre işler sıralanır. İşlem zamanları ilk makinelerde kısa olanlar öne, uzun olanlar sona gelecek şekilde eğim dizisi tamamlanır.

2.Campbell, Dudek, Smith Algoritması(1970) : Temelde Johnson algoritması kurallarına dayanan bu yöntemde, m makine n iş problemi öncelikle, $m-1$ adet n iş 2 makine problemine dönüştürülür. Sonrasında Johnson algoritması ile problemler tek tek çözülür. En iyi sonucu veren iş sırası uygun çözüm olarak kabul edilir.

3.Gupta Yöntemi(1971) : Gupta'nın yönteminde Palmer'a benzer bir eğim dizisi oluşturulur ancak farklı bir hesaplama ile dizin değerine ulaşılır.

4.Dannenbring Yöntemi(1972) : Dannenbring, Palmer ve Campbell, Dudek, Smith algoritmalarını incelemiş ve bunları avantajlı taraflarını birleştirerek yeni bir sezgisel yöntem geliştirmiştir. Bu yöntemde de Campbell, Dudek, Smith algoritmasında olduğu gibi m makine n iş problemi, $m-1$ adet 2 makine n iş problemine dönüştürülür. Ancak bu dönüşüm Palmer'in eğim dizisine göre yapılır.

5.Nawaz, Ensore ve Ham Yöntemi(1983) : Bu yöntem, işleri toplam işlem zamanlarına göre sıralayan bir algoritmadır. İşler tezgahdaki toplam işlem zamanlarına göre büyükten küçüğe doğru sıralanırlar.

6.Hundal ve Rajgopal Yöntemi(1988) : Palmer'in sezgisel yönteminin basit kurallarının iyileştirilmesi sonucu ortaya çıkmış bir yöntemdir.

7.Widmer ve Hertz Yöntemi(1989) : Widmer ve Hertz 'SPRINT' olarak adlandırılan iki aşamalı bir sezgisel yöntem geliştirmişlerdir. İlk aşamada, gezgin satıcı problemindeki(TSP) gibi analoji ile başlangıç çözümü belirlenir. İkinci aşamada ise bu çözüm 'Tabu araştırmaları' tekniği ile iyileştirilir.

8.Ho ve Chang Yöntemi(1991) : Bu yöntemde daha önce sezgisel metotlarla belirlenmiş işlerden, birbirini takip eden işler arasındaki boşlukların minimize edilmesine çalışılır.

Bu alandaki diğer çalışmalar ise,

Ignal ve Schrage, ilk olarak, dal-sınır tekniği ile akış tipi çizelgeleme problemlerini çözmeye çalışmışlardır. Hoogwen farklı bir dal sınır prosedürü geliştirmiştir. Bu çalışmalarda sadece bir tek kriter(C_{max} ; F veya F') göz önüne alınmıştır. Ancak karar verebilmek için çoğu zaman iki yada daha fazla kritere ihtiyaç vardır. Bu sebepten çok kriterli iki makine akış problemi gündeme gelmiştir. Selen, Hott ve Wilson çok kriterli iki makine akış problemi için tam sayılı programlama modeli geliştirmişlerdir. Bununla toplam akış süresi ve en son işin tamamlanma süresi minimize edilmeye çalışılmıştır (Chou ve Lee, 1999).

Genel olarak iş çizelgeleme problemlerinde kullanılan yöntemler aşağıdaki gibidir (Maturana vd., 1997) :

- 1.Matematik programlama,
- 2.Öncelik kurallarına göre sıralama,
- 3.Simülasyon tabanlı metotlar,
- 4.Yapay zeka tabanlı metotlar,
- 5.Sezgisel Yöntemler,
- 6.Çok temsilli paradigmlar(multi-agent paradigm),
- 7.Dağıtım yapay zeka teknikleri ile üretim çizelgeleme(DAI yazılımları),
- 8.Üretim düzgünleştirme ve çizelgeleme operatörü yönetimi.

Çizelgeleme optimizasyonunda, hesaplama süresi çok uzun olduğundan matematik programlama kullanmak uygun değildir. Esnekliğinden dolayı, simülasyon tabanlı metotlar son yıllarda daha çok tercih edilmektedir. Akış tipi çizelgelemede de, öncelik kurallı

simülasyon tabanlı çizelgeleme programlarının iyi performans verdiği belirlenmiştir (Maturana vd., 1997).

2.1 Çizelgelemede Kullanılan Öncelik Kuralları ve Kabuller

Literatür incelendiğinde çizelgeleme problemleri için on farklı kuralın kullanıldığı görülmektedir (Dorndorf ve Pesch, 1994). Bunlar:

1. Erken teslim tarihi,
2. İlk giren ilk çıkar(FIFO),
3. Kısa işlem zamanı(SPT),
4. Kalan işlem zamanı,
5. Toplam işlem zamanı,
6. Dinamik değişkenli,
7. Rassal öncelik kurallı,
8. Uzun işlem zamanlı(LPT)
9. Az operasyonu kalan öncelik kuralı,
10. Çok operasyonu kalan öncelik kuralıdır.

Akış tipi çizelgeleme probleminin çözümünde, şu kabuller yapılır (French, 1982):

1. Bir iş, farklı operasyonların birleşiminden meydana gelmesine rağmen, aynı işin iki operasyonu aynı anda işlenemez.
2. İşler için önceden rezervasyon yapılamaz, makinede operasyon yapılırken diğer operasyonun başlayabilmesi için bir öncekinin bitmiş olması gerekir.
3. Her iş m farklı operasyondan oluşur. Bir işin bir makinada ikinci bir defa işlem görmesine izin verilmez.
4. Operasyonların ertelenmesi yapılamaz, her işin süreci tamamlanmalıdır, eksik bırakılıp sonra tamamlanamaz.
5. Süre. işlem zamanları çizelgeden bağımsızdır. Burada iki durum kabul edilir. Birincisi, her bir işlemin hazırlık süresi çizelgeden bağımsızdır. İkincisi, işlerin makinalar arasındaki taşıma süreleri ihmal edilir.
6. Süreçler arasında envanter stoğuna müsadde edilebilir; iş, bir sonraki makinada işlenmek üzere bekleyebilir.

7. Her bir makina tipinden sadece bir tane mevcuttur, süreç içerisinde işin makina seçmesine müsaade edilmez. Bu kabul bazı durumlarda ortadan kaldırılabilir, belirli makinalar birden fazla eklenerek darboğaz ortadan kaldırılmaya çalışılabilir.

8. Makinaların, boş/aylak süreleri olabilir.

9. Hiçbir makina bir anda birden fazla operasyonu yerine getiremez.

10. Makinalar asla arızalanmazlar, çizelge periyodu boyunca makinalar hazır beklerler.

11. Teknolojik kısıtlar önceden bilinmektedir.

12. Olayda rassallık yoktur. Yani

- İş sayısı biliniyor ve sabittir.
- Makina sayısı biliniyor ve sabittir.
- İşlem zamanları biliniyor ve sabittir.
- İhtiyaç duyulacak diğer parametreler baştan beri bilinmektedir.

2.2 Johnson Algoritması

Birden fazla makinanın kullanıldığı çizelgeleme problemleri, akış tipi veya atölye tipi çizelgeleme problemleridir. Akış tipi çizelgelemede, teknoloji kısıtı, bütün işlerin aynı sıra ile makinalardan geçmesidir. Johnson algoritması, akış tipi çizelgeleme problemlerinde iki makina için ve özel şartlarda da üç makina için optimum çözüm sağlamaktadır.

$n/2/F/F_{\max}$ (n-iş, 2-makina, tamamlanma zamanı kriterli) probleminde amaç $F_{\max}=C_{\max}$ değerinin minimum yapmaktır. Bütün işlerin sıfır anında hazır olduğu kabul edilir. Buna göre çözümün adımları (French, 1981):

Adım 1: $k=1$ ve $p=n$ kabul et.

Adım 2: Sıralanmamış işlerin listesini oluştur, $iş=(J_1, J_2, J_3, \dots, J_n)$

Adım 3: Bütün işler için en küçük a_i ve b_i değerlerini bul.

Adım 4: J_i . işin makina birdeki işlem zamanı a_i en küçük ise,

- a. J_i . işi k . pozisyonda sıraya yerleştir
- b. J_i . işi, sıralanmamış işler çizelgesinden sil.
- c. k değerini, $k+1$ olarak arttır.
- d. Adım 6'ya git.

Adım 5: Eğer J_i . işin en küçük zamanı ikinci makinada ise yani b_i ise,

- a. J_i . işi 1. pozisyonda sıraya koy
- b. J_i . işi sıralanmış işler çizelgesinden sil.

- c. p değerini $p-1$ olarak azalt.
- d. Adım 6'ya git.

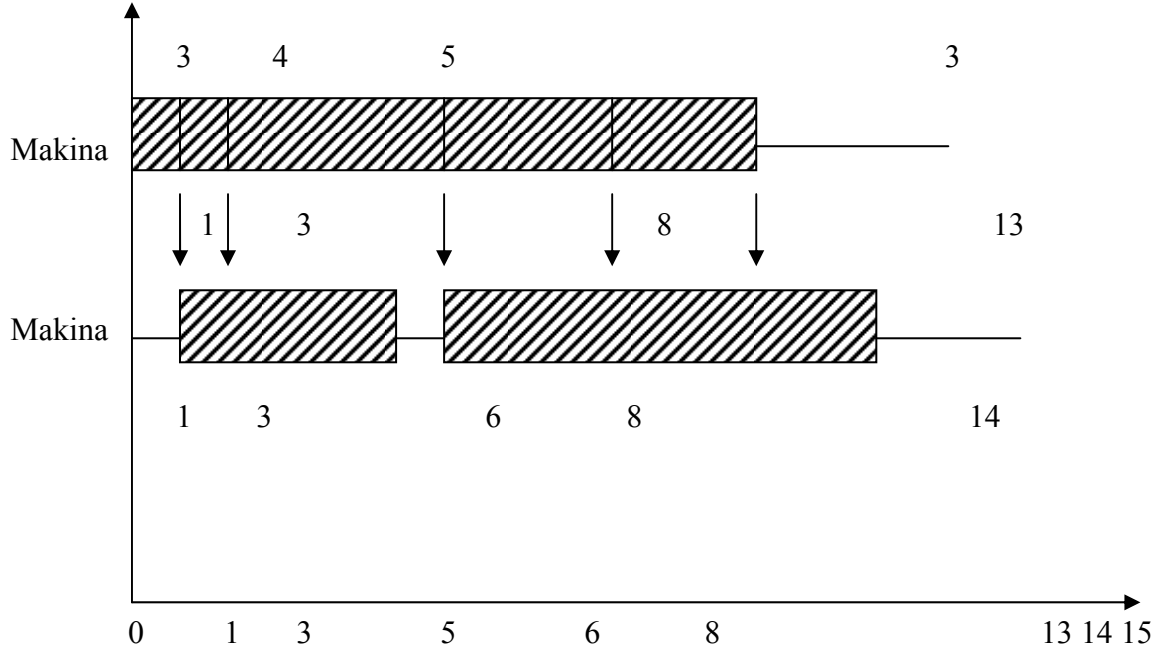
Adım 6: Eğer henüz çizelgelenmemiş iş mevcut ise 3. adıma git aksi halde dur.

Örnek 2.1 İki makinanın herbiri üzerinde beş iş için işlem süreleri Çizelge 2.1'de gösterilmiştir.

Çizelge 2.1 Örnek 2.1 için işlem süreleri

İş	Makine 1	Makine 2
1	4	3
2	1	2
3	5	4
4	2	3
5	5	6

Amaç minimum tamamlanma zamanlı çizelgeyi oluşturmaktır. Johnson algoritması kullanılarak makina 1 üzerinde iş 2 için $\min\{P_{ij}\} = 1$ olur. Bundan dolayı, dizilimde ilk iş hanesine 2. iş yerleşir. İkinci iş listeden çıkarıldıktan sonra, makina 1 üzerinde iş 4 için $\min_{i \neq 2,4}\{P_{ij}\} = 3$ olur. Bundan dolayı sıralamada ikinci iş olarak iş 4 yerleştirilir. Makina 2 üzerinde iş 1 için $\min_{i \neq 1,2,4}\{P_{ij}\} = 4$ oluşur. Bu değerden dolayı, dizilimde son iş hanesine 1 numaralı iş yerleştirilir. Daha sonra makina 2 üzerinde, 3. işin işlem zamanının kısa olduğu belirlenir. 3. iş dizilimde 4. olarak yerleştirilir. Geriye yalnızca 5. iş kalır ve bu da dizilimdeki tek boş yer olan 3. pozisyona yerleştirilir. Bu durumda minimum tamamlanma zamanlı sıralama $\{2, 4, 5, 3, 1\}$ olur. Bu örnek için gantt şeması çizildiğinde tamamlanma zamanı 21 birim olarak hesaplanır.



Şekil 2.1 Johnson algoritması ile (2,4,5,3,1) numaralı işlerin gantt diyagramı

2.3 Kombinatoriyel Optimizasyon

Optimizasyon problemleri, karar değişkenlerinin sürekli ve süreksiz olmasına göre iki kısma ayrılır. Karar değişkenleri süreksiz(kesikli) olanlara kombinatoriyel optimizasyon denir (Reeves, 1995a).

Problem çözümlerinde kullanılan algoritmalarda önemli olan, sonuca kısa sürede ulaşmalarıdır. Bir algoritmanın performansını ölçmede kullanılan en iyi kriter, sonucu bulana kadar harcadığı süredir. Fakat bu süre işlem hızı ve raporlama tekniğine bağlı olarak değişkenlik gösterir.

Pratikteki problemlerin çözümünde polinomiyal algoritmalar iyi performans gösterirler. Büyük algoritmalar genellikle üstel algoritma olarak adlandırılırlar. Üstel algoritmalarda polinomiyal olmayan artış hızları geçerli olur(2^n , $n!$, n^2 , $n^{\log n}$ gibi). Polinomiyal algoritmalar, üstel algoritmalara göre teknolojik gelişmelerden daha fazla faydalanırlar ve özel problemlerin çözümlerinde kapalılık özellikleri dolayısıyla polinomiyal algoritmalar birleştirilerek kullanılabilir.

Kombinatoriyel optimizasyon problemlerinin büyük çoğunluğu, “NP-Complete(Nonpolynomially Bounded)”, polinomiye zaman sınırı olmayan problemler sınıfına girerler. NP problemleri için polinomiye algoritmalar geliştirilememiştir. Bunun yerine NP kapsamına giren problemlerde, optimum çözüme ulaşmak yerine, en yakın çözümlere ulaşmak tercih edilir. Bu tip problemlerin kesin çözümleri çok zaman aldığından, yerel arama ve stokastik arama yöntemleri ile yaklaşık çözümler elde edilmeye çalışılır. Yerel arama yöntemlerinin, yerel optimum noktalarında takılıp kalmasını önlemek için stokastik arama yöntemleri geliştirilmiştir. Aşağıda kombinatoriyel optimizasyon problemleri ile ilgili bazı örnekler görülmektedir.

Örnek 2.2 Tahsis atama problemleri: n kişinin n adet göreve atanması.

1.personelin j. göreve atanmasının minimum maliyeti,

$$\min = \sum_{i=1}^n c_i \prod_i \dots\dots\dots \{\Pi_1, \Pi_2, \Pi_3, \dots, \Pi_n\} \quad (2.1)$$

Örnek 2.3 Sırt Çantası Problemi (0-1)

C kapasiteye sahip bir sırt çantasına, n adet malzeme yerleştirilecektir. i. malzemenin kazanç değeri V_i , kapsadığı birim alan c_i ise,

Amaç fonksiyonu,

$$\max = \sum_{i \in I} V_i \quad (2.2)$$

Ve kısıt kümesi,

$$\sum_{i \in I} c_i \leq C \quad (2.3)$$

Örnek 2.4 Araç Rotalama Problemi

m adet ile n adet noktaya dağıtım yapılacaktır. k. aracın kapasitesi c_k birim, i. noktanın ihtiyacı c_i birim, i ile j arasındaki mesafe d_{ij} kadardır. Hiçbir araç D birimden fazla taşıma yapamaz. Minimum maliyetli araç rotalama modeli,

$$\sum_{k=1}^m \sum_{i=0}^{n_k} d_{\Pi_{i,k} \Pi_{i+1,k}} \quad (2.4)$$

Talep kısıtı,

$$\sum_{i=1}^{n_k} c_{\Pi_{i,k}} \leq C_k \quad ; \quad k=1, \dots, m \quad (2.5)$$

Kapasites kısıtı,

$$\sum_{i=0}^{n_k} d_{\Pi_{i,k} \Pi_{i+1,k}} \leq D \quad ; \quad k=1, \dots, m \quad (2.6)$$

Tüm noktalara ulaşma kısıtı,

$$\sum_{k=1}^m n_k = n \quad (2.7)$$

Kombinatoriyel optimizasyon problemleri, lineer proglamlama modellerine dönüştürülerek çözülebilir. Örneğin, sırt çantası problemi 0-1 tamsayılı programlama problemine dönüştürülerek çözülebilir. Diğer taraftan, araç rotalama probleminde bu dönüşümün yapılması değişken ve kısıt sayısını arttıracığından oldukça zor olmaktadır.

Kombinatoriyel problemlerde, sonlu yada sayılabilir sonsuz bir kümeden, bir alt kümeye, nesneye, bir permutasyona ulaşılmaya çalışılır. Kombinatoriyel optimizasyon problemlerinin büyük bir kısmı sezgisel(heuristic) yöntemler ile çözülebilmektedir. Yani bu tür problemler için optimum değil optimuma yakın çözümler elde edilir (Reeves, 1995a).

Temelde NP problemlerinin çözümünde kullanılan metotlar;

- Lokal arama metotları
- Yapay sinir ağları
- Tabu araştırmaları
- Tavlama benzetimi
- Karınca kolonileri optimizasyonu
- Yapay bağışıklık sistemleri
- Genetik algoritmalarıdır.

2.4 Tabu Arařtırmaları

Glover, Modern Tabu arařtırmaları konusunda ilk alıřmaları yapan kiřidir (Reeves, 1995a). Lokal optimalliđin tıkanıđı noktalarından kaınmak ve yeni yonemlere yol gstermek iin geliřtirilmiř olan Tabu arařtırmaları, optimuma yakın zm veren sezgisel bir arama metodudur. Tabu arařtırmaları, hedefe kısıt ve tabu kullanarak belirli bir olasılıkla ulařır. Hedefe ulařma olasılıđını arttırmak iin kısıt ve tabu sayılarını arttırmak gerekir. Tabu arařtırmalarında algoritma adımları ařađıdaki řekildedir (Reeves, 1995a):

Adım 1: Bařlangı zmn rassal olarak se, bu zm iin ama deđerlerini hesapla;

Adım 2: Deđiřim operatrn se(SWAP veya INVERSE) bu operatrer gre pozisyonlar arasında deđiřimi gerekleřtir;

Adım 3: Yeni deđiřimin ama deđerlerini hesapla, bařlangı deđerini ile veya bir nceki deđer ile karřılařtır. Deđiřimin tabu olup olmadıđını belirle, eđer deđiřim kabul edilebilir ise tabu deđildir.

Adım 4: İterasyon sayısı kadar adım 2'ye giderek iřlemleri tekrarla, hesaplanan en iyi zm belirle.

Tabu listesi kabul edilebilir deđiřimleri saklamak, istenmeyen durumun ortadan kaldırılmasında kullanılır. Tabu arařtırmalarında temelde  ayrı strateji kullanılır. Bunlar:

1. Yasaklama Stratejisi, hangi hareketlerin tabu listesine girip girmeyeceđini belirleyen ve dngy nlemek iin belirli hareketleri yasaklama mekanizmasını oluřturan bir stratejidir.
2. Yeterlilik Kriteri, yeterli kořulları sađlayan zmlerin tabu durumundan ıkmasını sađlayan bir stratejidir.
3. Kısa Dnem Stratejisi, nceki iterasyonlardaki davranıřları depolayan bir stratejidir.

2.5 Tavlama Benzetimi

Tavlama Benzetimi(simulated Annealing) kombinatoriyal optimizasyon problemlerinin zmnde iyi performans gsteren sezgisel bir yntemdir. Tavlama benzetimi ilk olarak 1953 yılında, bir ısı banyosu iindeki taneler kmesinin denge dađılımını hesaplamak

amacıyla simülasyon tekniği ile birlikte Metropolis, Rosenbluth ve Teller tarafından kullanılmıştır (Reeves, 1995a).

Tavlama benzetimi, ayrık optimizasyon ile fiziksel tavlama sürecinin benzeşiminden ortaya çıkmış bir yöntemdir. Fiziksel tavlama süreci, ısı banyosundaki katı bir cismin düşük enerjilerini elde etmek için kullanılır. Katı bir cisim erime noktasına kadar ısıtılır daha sonra hızla soğutulmaya başlanırsa, moleküler yapısı soğutma oranına bağlı olarak değişir. Metropolis algoritmasının temel prensibi, soğutma sürecindeki enerji değişimine göre belirlenir.

Metropolis algoritmasının bulunmasından otuz yıl sonra, Kirkpatrick ve diğerleri bu algoritmayı bir optimizasyon probleminde uygun bir çözüm bulmak için kullanmışlardır. Temelde bu yöntem, lokal aramanın farklı bir versiyonudur (Reeves, 1995a).

Metropolis çalışmaları sırasında istatistiksel termodinamikten yararlanmıştır. Termodinamik yasalarına göre, t. durumdaki sıcaklığa bağlı olarak, enerjideki azalma olasılığı, δE ,

$$P(\delta E) = \exp(\delta E/kt) \quad (2.8)$$

formülü ile ifade edilir. Formüldeki k, Boltzman sabitidir. Metropolis simülasyonu kullanarak enerjideki fiziksel değişikliği hesaplamıştır.

Kirkpatrick ve diğerleri ile Cerny, Metropolis algoritmasının optimizasyon problemlerinde uygulanması ile ilgili Çizelge 2.2 deki benzerlikleri ortaya koymuşlardır (Reeves, 1995a).

Çizelge 2.2 Termodinamik simülasyonu ve kombinatoriyel optimizasyon benzerlikleri

Termodinamik Simülasyonu	Kombinatoriyel Optimizasyon
Sistem Durumu	Uygun Çözüm
Enerji	Maliyet
Durum Değişmesi	Komşuluk Aralığı Çözümü
Sıcaklık	Kontrol Parametreleri
Donma Noktası	Yaklaşık Çözüm

Malzemelerin çoğundaki enerji fonksiyonu, ayrık optimizasyon problemlerindeki maliyet fonksiyonlarına karşılık gelir. Optimizasyon problemlerinde, optimum çözüme yakın çok sayıda lokal minimum maliyet fonksiyonu değeri mevcuttur.

Tavlama Benzetimi algoritması çalışma prensibi olarak, Lokal arama metoduna benzemektedir. Lokal arama metodunun en büyük dezavantajı, global optimum yerine lokal optimum değerleri bulmasıdır. Tavlama benzetimi algoritması, bu durumu ortadan kaldırır. Kontrol parametrelerine(sıcaklık) bağlı olarak, komşuluk değerlerindeki maliyet fonksiyonu değişimini kabul eder. Algoritma adımları;

Adım 1: Rassal olarak yada seçilmiş bir çözüm kümesini başlangıç çözümü olarak kabul et.(S₀);

Adım 2: Başlangıç çözümüne en iyi çözüm olarak S ata. S* = S;

Adım 3: Başlangıç çözümünün maliyet fonksiyonunu hesapla. S : C(S);

Adım 4: Başlangıç sıcaklığını belirle. (T₀);

Adım 5: Başlangıç sıcaklığını T' değerine ata. T = T₀;

Adım 6: Durma kriteri gerçekleşmedi ise aşağıdaki işlemleri gerçekleştir.

a) Markov zinciri uzunluğunu belirle.

i) Mevcut S çizelgesinde rassal bir komşuluk aralığı, S', belirle.

ii) S' için C(S') maliyet fonksiyonunu hesapla.

iii) Bir önceki çizelge ile mevcut çizelgenin maliyet fonksiyonlarının farklarını al.

$$\Delta(C) = C(S') - C(S)$$

iv) Eğer $\Delta(C) \leq 0$ ise, b). adıma geri dön, S = S' kabul et.

$$C(S) < C(S^*) \text{ dan } S^* = S \text{ ata.}$$

v) $\Delta(C) > 0$ ise, adım 6 a) ya geri dön ve x gibi (0,1) aralığında bir rassal sayı üret. Eğer $X < e^{-\Delta(C)/T}$ ise S = S' ata.

b) T sıcaklığını azalt, adım 6 ya dön.

Adım 7: En iyi çizelgeyi, S*, oluştur ve dur.

Teorik olarak, Tavlama Benzetiminin çalışma prensibi, Markov Zincirlerinin modellenmesi ile elde edilir. Mevcut bir durumdan başka bir duruma hareketlenme olasılığı matris formları ile ve i. çözümden j. çözüme, sıcaklık sabitine bağlı olarak P_{ij} homojen Markov Zinciri olarak bilinir. Tavlama Benzetiminde sıcaklık sabit değildir, belirli bir iterasyona göre azaltılır. Sıcaklık parametresinin azaltılması, çözüme ulaşmada önemli bir parametredir. Pratikte iki farklı kontrol yöntemi kullanılır. Bunlardan biri, geometrik azalma fonksiyonudur.

$A(t) = a \cdot t$ ve burada, $a < 1$ dir. Literatürde $0,80 < a < 0,99$ arasında seçilir. Diğer bir yöntemde ise,

$$\alpha(t) = \frac{t}{(1 + \beta t)} \quad (2.9)$$

Bu ifadede β uygun küçük bir değerdir.

2.6 Karınca Kolonileri Optimizasyonu

Karınca sistemleri (Ant Systems) kombinatoriyel optimizasyon problemlerinin çözümünde doğal süreçlerin kullanıldığı sezgisel bir yöntemdir. 1991 yılında, Colomi, Dorigo, Maniezzo tarafından gezgin satıcı probleminin çözümünde başarı ile uygulanmıştır. Karınca Sistemleri, karınca kolonilerinin yiyecek toplama prensibine göre çalışır (Colomi vd., 1994). Karıncalar kör olduklarından, koloni halinde yiyecek toplamadaki en kısa yolu seçme mekanizmalarına göre algoritma oluşturulur. Karıncalar yiyecek toplarken geçtikleri yolun üzerine sıvı bir madde bırakırlar. Bu madde uçucu olup belirli bir süre sonra kaybolur. Belirli bir t zamanında belirli bir yoldan ne kadar çok karınca geçerse iz miktarı o kadar fazla olur. Daha sonra aynı yol üzerine gelen karıncalar kısa yolu iz miktarına bağlı olarak seçerler. Başka bir deyişle gidebileceği birden fazla yol olması durumunda iz miktarının fazla olduğu yolu tercih ederler. Karıncalar koloni halinde çalıştıklarından bu yöntem ile etkin sonuç elde etmektedirler.

Karınca sistemlerinin algoritmasının kullanımı için önce iz matrisi, $T = [\tau_{ij}]$, ve görünürlük matrisi, $H = [\eta_{ij}]$, tanımlanır.

Problem çözüm kümesi permutasyon tipi tanımlanır ise, $\Pi = [\Pi(1) \dots \Pi(n)]$, bu durum problemin n farklı girişi olduğunu gösterir. Çözümün ilk adımında, m adet karıncanın ağ üzerinde rassal olarak belirlenmiş bir pozisyonda olduğu ve bu karıncaların her birinin bulunduğu düğümden herhangi bir yere hareket etme serbestliğine sahip olduğu kabul edilir. Her bir karınca, Montekarlo prosedürüne göre geçiş olasılıklarına bağlı olarak hareket edeceği yönü belirler.

Karınca, i . düğümden j . düğüme hareket ettiğinde, j . düğüm karıncanın hafızasında tabu listesine kaydedilir. Bir sonraki adımda karınca, tabu listesinde bulunan bu yöne doğru hareket etmez.

Her bir karıncanın i. düğümünden j. düğüme, k. adımda geçme olasılığı,

$$P_{ij}^k(t) = \left[\frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j \in A_k} [\Pi_{ij}(t)]^\alpha [\eta_{ij}]^\beta} \right] \quad (2.10)$$

Burada, α ve β sezgisel parametrelerdir. $\Delta\tau_{ij}^k$, (i,j) çevrimi boyunca, k. karıncadan sonra bırakılan toplam iz miktarı aşağıdaki ifade ile gösterilmiştir.

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2.11)$$

Buradaki iz miktarı bir sonraki iterasyonda kullanılır. (t+1) durumundaki iz miktarı,

$$\tau_{ij}(t+1) = P.\tau_{ij}(t) + \Delta\tau_{ij} \quad (2.12)$$

P, kovaryans değerini, (1-P) izin buharlaşma miktarını ve (1-P) $\tau_{ij}(t)$, (i,j) düğümleri arasında, t ve (t+1) zaman diliminde buharlaşan iz miktarını gösterir.

Bu algoritma, Colorni, Drogo ve diğerleri tarafından 1994 yılında atölye tipi çizelgeleme problemlerinde başarı ile uygulanmıştır.

2.7 Yapay Bağışıklık Sistemleri

Doğal bağışıklık sistemleri, enfeksiyondan korunma ve enfeksiyonların ortadan kaldırılması işlemlerini gerçekleştirir. Bağışıklık sistemleri hücre ve moleküllerden oluşur. Bu sistemlerde, yabancı proteinlerin, “antijen”, tespit edilmesinde kullanılan iki dedektör mevcuttur, bunlar “T” ve “B” hücreleri olarak adlandırılır. Ayrıca antijenleri saran antikorlar mevcuttur.

Detektör hücreleri ile antijenler arasında, hücre yüzeyinde birtakım fiziksel ve kimyasal işlemler gerçekleşir. Her bir dedektör belirli bir antijeni tespit edebilir. Bir dedektör antijeni sardığında bu fiziksel ve kimyasal işlemler sayesinde çoğu zaman antijen elemine edilir. Antijenin elemine edilmesi, dedektörün içeriğine bağlıdır.

Bağışıklık sistemleri ile bir problem çözerken, süreç, içsel ve dışsal mesajlardan oluşur. İçsel mesajlar, hücre ve moleküller, dışsal mesajlar, bakteri, parazit ve virüs gibi yabancı

maddelerdir. Problem çözümlerinde, içsel ve dışsal mesajların ayırt edilmesi birçok nedenden ötürü oldukça zordur.

Forrest ve Hofmeyr, bilgisayar güvenlik sistemlerinde bağışıklık sistemleri ile ilgili çeşitli uygulamalar yapmışlardır. Bilgisayar güvenlik ve bağışıklık sistemleri benzer şekilde, dinamik, kompleks ve sürekli değişim gösteren bir yapıya sahiptirler (Forrest ve Hofmeyr, 2000).

Yapay bağışıklık sistemleri, birçok endüstriyel problemin çözümünde kullanılabilen yeni bir yapay zeka tekniğidir (Dasgupta ve Forrest, 1997).

Bağışıklık sistemleri, temelde öğrenme ve hafızada tutma tekniğini kullanırlar. Bir bağışıklık sistemi en az, 10^{16} farklı yabancı molekülü ortaya çıkartabilir. İnsan genomu, 10^5 farklı gen içerir (Forrest vd., 1993).

Genetik algoritmalarındaki doğal seçim işlemi, bağışıklık sistemlerine, bakterilerin farkına varılması prosedürü öğretilerek gerçekleştirilir (Forrest vd., 1993).

3. GENETİK ALGORİTMALAR

Genetik algoritma, rassal arama teknikleri kullanarak çözüm bulmaya çalışan parametre kodlama esasına dayanan bir arama tekniğidir (Goldberg, 1989).

Genetik algoritmayı diğer arama tekniklerinden farklı kılan özellikleri şöyledir (Goldberg, 1989):

- 1.Genetik algoritma, parametrelerin kendileri ile doğrudan ilgilenmez, parametre setlerinin kodları ile ilgilenir,
- 2.Genetik algoritmanın arama alanı, yığının veya popülasyonun tamamıdır; tek nokta veya noktalarda arama yapmaz,
- 3.Genetik algoritmalarda amaç fonksiyonu kullanılır, sapma değerleri veya diğer hata faktörleri kullanılmaz,
- 4.Genetik algoritmaların uygulanmasında kullanılan operatörler stokastik yöntemlere dayanır, deterministik yöntemler kullanılmaz.

Genetik algoritmalar, daima optimum olmasa da optimuma yakın sonuçları garanti eder. Bir problemin genetik algoritma yaklaşımı ile çözülmesinde kullanılan adımlar şunlardır (Croce, 1995):

- 1.Arama uzayındaki bütün muhtemel çözümler dizi olarak kodlanır. Bu diziyi oluşturan her bir elemana(kromozom) “gen” denir. Her bir dizi, arama uzayında belirli bir bölgeye tekabül eder.
- 2.Genellikle rassal olarak bir çözüm seti seçilir ve başlangıç popülasyonu olarak kabul edilir.
- 3.Her bir dizi için bir uygunluk değeri hesaplanır, bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
- 4.Bir grup dizi, belirli bir olasılık değerine göre rassal olarak seçilip üreme işlemi gerçekleştirilir.
- 5.Üreme işleminde, çeşitli genetik operatörler kullanılabilir.
- 6.Yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur.
- 7.Önceden belirlenen nesil sayısı boyunca yukarıdaki işlemler devam ettirilir.

8. İterasyon, nesil sayısına ulaşıncaya işlem bitirilir. Uygunluk değeri en yüksek olan dizi seçilir.

Genetik algoritma ile ilgili kullanılan temel kavramlar şöyledir:

Gen: Kalıtsal moleküllerde bulunan ve organizmanın karakterlerinin belirlenmesinde rol oynayan kalıtsal birimlere denir (Oraler, 1990). Yapay sistemlerde gen, kendi başına anlamlı bilgi taşıyan en küçük birim sayılır. Bu tezde ele alınan iş çizelgeleme başlığında her bir gen bir işi temsil eder.

Kromozom: Birden fazla genin bir araya gelerek oluşturduğu diziye denir. Kromozomlar olası alternatif çözümleri gösterir.

Plan-Katar(Schemata): Belirli pozisyonlarda uygun olan alt dizileri tanımlamak için kullanılan sayı katarlarına veya planlarına denir (Goldberg, 1989).

Örneğin, iki tabanlı(1,0) kodlama sisteminde beş adet genden oluşan bir kromozom dizisinde her bir genin yerine (0,1,*) gibi üç farklı kod kullanılabileceğinden,

$L = 5$ için, $3 \times 3 \times 3 \times 3 \times 3 = 243$ farklı alt dizi tanımlanabilir.

Genel olarak iki tabanlı kodlama dizisinde plan(katar) sayısı (3.1) ifadesinde gösterilmiştir (Goldberg, 1989).

$$\text{Plan Sayısı} = (k+1)^L \quad (3.1)$$

L = Dizideki rakam(gen) sayısı

K = Kodlama sistemindeki karakter sayısıdır.

Popülasyon: Kromozomlardan oluşan topluluğa denir. Genetik algoritma problemlerinde popülasyon, geçerli alternatif çözüm kümesidir. Popülasyon içersindeki birey sayısı genellikle sabit tutulur. Popülasyondaki kromozom sayısı arttıkça çözüme ulaşma süresi (iterasyon sayısı) azalır. Problemin büyüklüğüne bağlı olarak popülasyon sayısı (10 ile 100) arasında seçilebilir (Ghedjati, 1999).

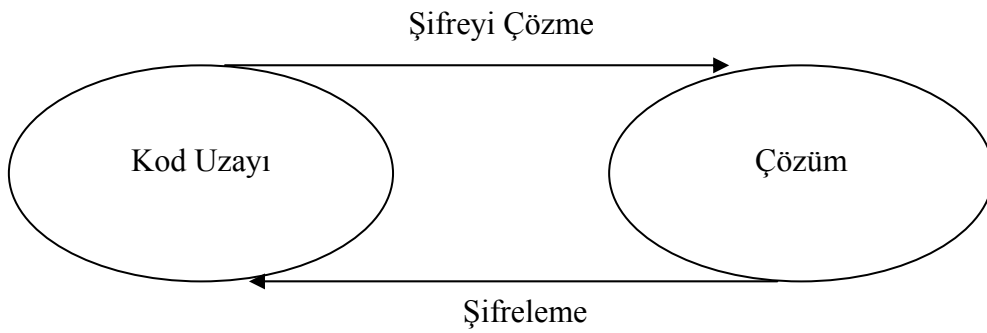
Genetik algoritmalarda dört ayı operatör kullanılır. Bunlar,

1. Parametre kodlama operatörü,

- 2.Üreme operatörü,
- 3.Çaprazlama operatörü,
- 4.Mutasyon operatörüdür.

3.1 Parametre Kodlama

Genetik algoritmaların en temel özelliği, çözüm uzayı ve kod uzayında çalışmasıdır. Kod ve çözüm uzayı arasındaki etkileşim aşağıda şekil 3.1’de gösterilmiştir.



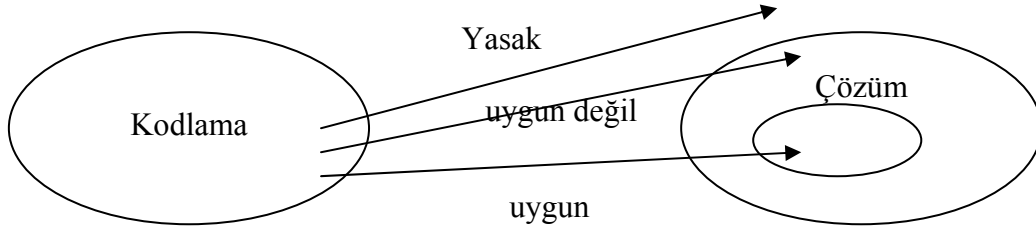
Şekil 3.1 Kod ve çözüm uzayı (Cheng vd., 1996)

Problemlerin başarılı bir şekilde çözümü için en kısa kodlama şekillerinin kullanılması gerekir (Goldberg, 1989).

Çözümün kromozomlara nasıl kodlanacağı genetik algoritma da anahtar niteliği taşır. Son on yıl boyunca değişik kodlama teknikleri geliştirilmiştir. Kromozomların kodlanması aşamasında dikkat edilmesi gereken üç önemli nokta vardır (Cheng vd., 1996). Bunlar;

- 1.Kromozomun uygunluğu(Feasibility) kontrol edilir,
- 2.Kromozomun belirtilen şartları(Constraints) sağlayıp sağlamadığı (Legality) kontrol edilir.

Kodlama haritasında, kromozomun tek olduğu kontrol edilir. Kromozomun uygunluğu ile şartları sağlayıp sağlamadığı işlemleri şekil 3.2’de belirtilmiştir (Cheng vd., 1996).



Şekil 3.2 Kodlamada uygun olma ve yasaklı olma durumları

Şartları sağlamayan bir kromozom, çözüm uzayındaki değerleri şifreleyemez ve evrime uğrayamaz. Bu tez kapsamında ele alınacak genetik algoritma modelinde akış tipi çizelgeleme problemi ile ilgilenileceğinden bu konuda geliştirilen kodlama yöntemleri anlatılacaktır.

İkili kodlama ile beraber kullanılan kodlama yöntemlerinden biri de harfli kodlamadır. Bu kodlamada her bir kromozoma bir harf verilir, kromozomdaki genler de ikili tabana göre kodlanır, eğer kromozom sayısı 26 dan fazla olursa artan kromozomlar için 1,2,3,... gibi rakamlar kullanılır (Goldberg, 1989).

Çok parametrelili kodlama, ikili kodlama tabanına göre kullanılır. Bu yöntemde uygunluk değeri en küçük olan kromozoma U_{\min} , en büyük olana da U_{\max} değerleri verilir ve kodlama U_1, U_2, \dots, U_n olarak yapılır (Min ve Cheng, 1999).

Akış tipi çizelgeleme problemleri için iki tip dizi kodlama yöntemi kullanılır (Cheng ve Gen, 1999). Bunlar;

- 1.Zayıf kodlama
- 2.Genel dizi kodlamadır.

Zayıf dizi kodlamada ayrıştırıcı semboller kullanılır. Genel dizi kodlamada ise birbirinin tekrarı olan (0-1) kullanılır. Zayıf kodlamada dizi içerisindeki tekrarlı genlere mücadele edilmez, her kodda en fazla bir tane bulunur.

John Holland tarafından geliştirilen ve yaygın olarak kullanılan plan son yıllarda çizelgeleme problemlerinde kullanılmamaktadır (Wang vd., 1999).

Çizelgeleme problemleri ile ilgili olarak son beş yılda, yukarıdaki yöntemlerden farklı olarak 9 farklı temsil yöntemi geliştirilmiştir (Cheng vd., 1996). Bunlar;

1. Operasyona dayalı kodlama,
2. İşe dayalı kodlama,
3. Liste önceliğine dayalı kodlama,
4. İş çifti ilişkisine dayalı kodlama,
5. Öncelik kuralına dayalı kodlama,
6. Tamamlanma zamanına dayalı kodlama,
7. Makinaya (iş yükleme grafiğine) dayalı kodlama,
8. Rassal anahtarlı kodlama,
9. Karışık kodlamadır.

Yukarıdaki kodlamalar iki temel yaklaşım içerisinde sınıflandırılabilir.

- Direkt yaklaşım
- En direkt yaklaşım

Operasyona dayalı kodlama, işe dayalı kodlama, iş çifti ilişkisine dayalı kodlama, tamamlanma zamanına dayalı kodlama, ve rassal anahtarlı kodlama direkt yaklaşım sınıfına aittir. En direkt yaklaşım olarak, tercih edilen dizine dayalı kodlama, öncelik kuralına dayalı kodlama yaklaşımları sayılabilir.

3.1.1 Operasyona Dayalı Kodlama

Bu tip kodlamada, her bir gen bir operasyona karşılık gelir. Her bir operasyon bir sayı ile temsil edilir. Gezgin satıcı probleminde olduğu gibi permütasyon kodlamaya benzer, yalnız öncelik kısıtları olduğu için permütasyon ile elde edilen bütün sonuçlar uygun bir çizelgeyi göstermez. m-makine n-iş probleminde kromozomlar $n \times m$ gen içerir, her iş tam olarak "m" defa kromozomda görünür ve her bir genin tekrarı, işin somut bir operasyonuna işaret etmez. Kod çözücü prosedürü ile bir kromozomdan bir çizelgelemenin şifresi çözülür. Bu tip kodlamada izlenen adımlar şöyledir;

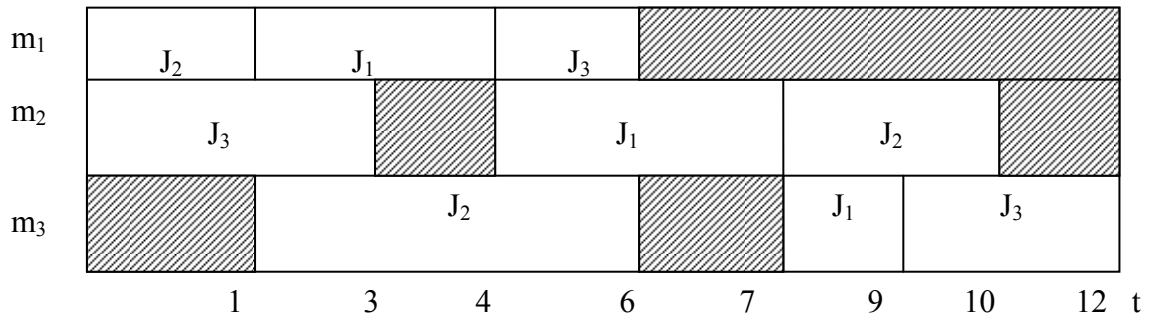
1. İlk olarak operasyon sıralarının oluşturduğu liste, kromozom şekline çevrilir,
2. Daha sonra sezgisel tabanlı bu listeden çizelge oluşturulur. Listedeki ilk operasyon, ilk çizelgelenir, daha sonra ikinci operasyon ve diğerleri devam eder. Uygun makine operasyon

gerekleri için mevcut en uygun işlem zamanı proses altındaki her operasyona ayrılır. Tüm operasyonlar çizelgelenene kadar işlem tekrar edilir (Cheng vd., 1996).

Çizelge 3.1’de verilen 3-iş ve 3-makine problemi için, [211122333] olarak verilen bir kromozom düşünülür, her iş üç operasyona sahip olduğundan dolayı, kromozomda her iş üç defa yer alır. Her gen bir operasyonu gösterir ve dizide işin oluşumu sırasına göre düzenlenebilir. m. makinede j işinin i. operasyonu O_{jim} ile gösterilir. Kromozom operasyonların öncelik ilişkilerine göre, $[O_{211}O_{111}O_{122}O_{133}O_{233}O_{232}O_{312}O_{321}O_{333}]$ şeklinde yazılır. Önceliğe sahip operasyon O_{211} olduğundan ilk olarak çizelgelenir. Daha sonra O_{111} ve diğerleri çizelgelenir. Şekil 3.3’de aktif çizelgeleme sonucu gösterilmiştir (Cheng vd., 1996).

Çizelge 3.1 İş sıralama problemi, 3-iş ve 3-makine örneği

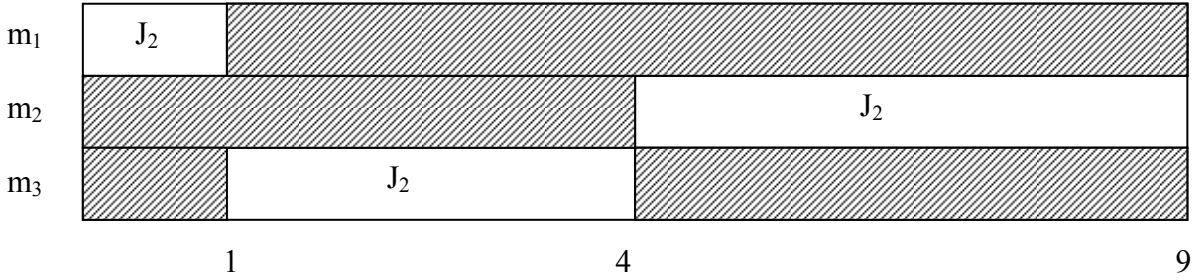
Operasyonlar			
	1	2	3
İşlem Zamanı			
J_1	3	3	2
J_2	1	5	3
J_3	3	2	3
Makine Dizilimi			
J_1	m_1	m_2	m_3
J_2	m_1	m_3	m_2
J_3	m_2	m_1	m_3



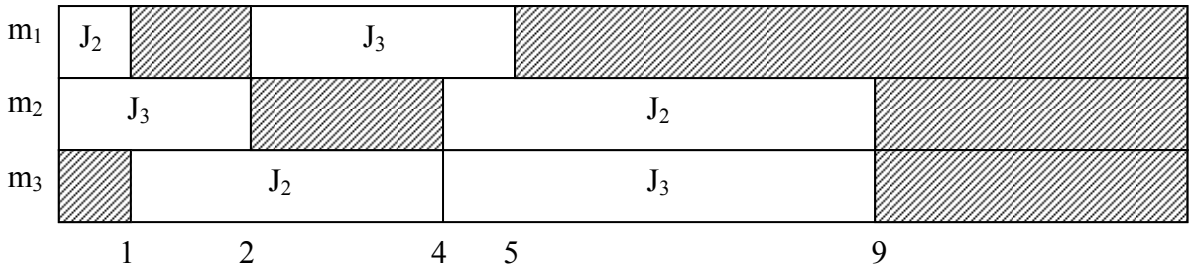
Şekil 3.3 Şifresi çözülmüş aktif çizelge

3.1.2 İşe Dayalı Kodlama

Bu kodlama, n iş listesinden oluşur ve işlerin dizilimine göre çizelge oluşturulur. İşlerin dizilimi verildiği için, listedeki ilk işin tüm operasyonları önce çizelgelenir, sonra listedeki ikinci işin tüm operasyonları çizelgelenir ve tüm operasyonlar çizelgeleninceye kadar işleme devam edilir. Kromozom dizisi $[2\ 3\ 1]$ olarak kabul edilir ise, işlem gördürülen ilk iş J_2 için öncelik k operasyonu $[m_1\ m_3\ m_2]$ ve her makine için işlem zamanının yerini tutan $[1\ 5\ 3]$ tür. Bu durumda ilk olarak J_2 işinin operasyonları (şekil 3.4(a)) daha sonra da J_3 işinin operasyonları çizelgelenir. Makineler içinde 3. işin operasyon önceliği $[m_2\ m_1\ m_3]$ ve her makinenin işlem zamanının yerini tutan $[3\ 2\ 3]$ tür. Şekil 3.4(b)'de gösterilen uygun işlem zamanında 3. işin operasyonlarının her biri çizelgelenir ve son olarak J_1 işinin operasyonları (şekil 3.4(c)) çizelgelenir (Cheng vd., 1996).

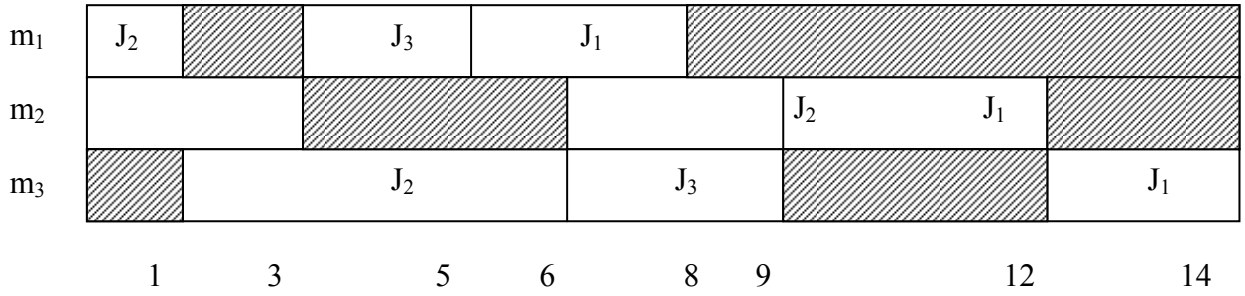


a. İlk olarak J_2 işi için çizelgeleme



b. J_3 işi için çizelgeleme

Şekil 3.4 İşe dayalı kodlama sistemi



c.J₁ işi için çizelgeleme
Şekil 3.4. İşe dayalı kodlama sistemi (Devamı)

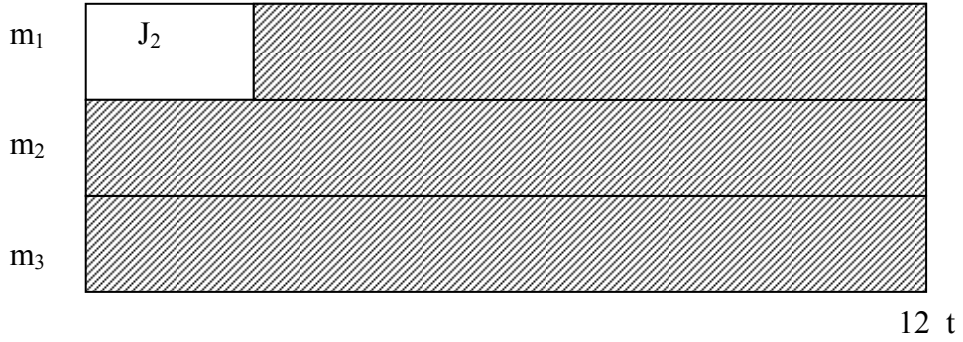
3.1.3 Liste Önceliğine Dayalı Kodlama

Bu tip kodlama, ilk olarak Davis tarafından çizelgeleme problemlerinde kullanılmıştır. Daha sonra Falkenauer ve Bouffouix detaylandırarak teslim tarihi ve geride kalan süreye göre, atölye tipi çizelgeleme problemlerinde kullanmışlardır (Cheng vd., 1996). Çizelgeleme problemi, n-iş m-makine için, m alt kromozomdan oluşturulur ve her bir alt kromozom dizisindeki semboller n iş sayısı uzunluğunda olur. Her bir sembol makinedeki ilgili bir operasyonu gösterir. Alt kromozomlar, makine üzerindeki operasyon dizilimlerini göstermez ve öncelik listesinde yer alır. Her makine kendine ait öncelik listesine sahiptir. Makine önünde, kuyruktaki beklemlerin durmunu analiz etmek için, simülasyonla kromozomlardan asıl çizelge çıkarılır ve çizelge düzenlemek için öncelik listeleri kullanmak gerekli ise, öncelik listesinde ilk görülen operasyon seçilir.

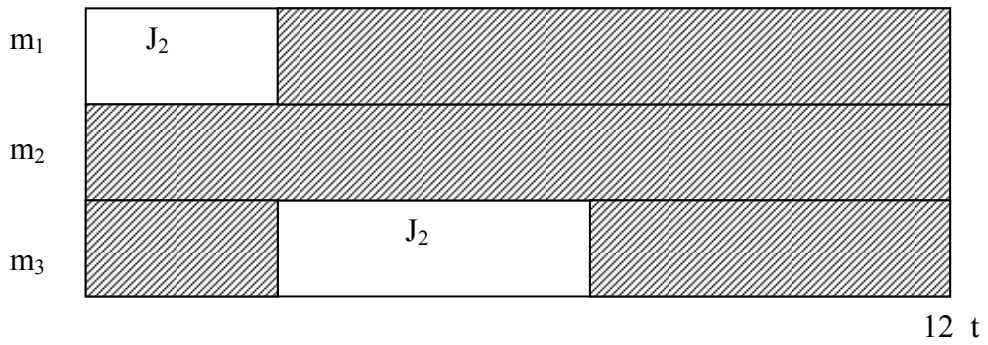
Kromozomdan asıl çizelgenin nasıl çıkartılacağı, çizelge 3.1'deki örnek üzerinden açıklanmıştır. Bir kromozomun [(2 3 1) (1 3 2) (2 1 3)] olduğu varsayılır ise, m₁ için ilk gen (2 3 1) tercih listesidir. Bu liste m₂ için (1 3 2) ve m₃ için de (2 1 3) dür. Operasyonlar, m₁ üzerinde J₂ işi, m₂ üzerinde J₁ işi ve m₃ üzerinde de J₂ işi çıkarabilir. Verilen öncelik kısıtlarına göre, şekil 3.5(a)'da gösterildiği gibi m₁ üzerinde J₂ çizelgelenir. m₃ üzerinde gelecek çizelgelenebilir operasyon J₂ dir (şekil 3.5(b)). Bu durumda geçerli tercih edilebilir operasyonlar m₁ üzerinde J₃ ve m₂ ve m₃ üzerinde J₁ dir. Geçerli zamanda tümü çizelgeleneemediğinden dolayı her listedeki ikinci tercih edilebilir operasyonlar belirlenmeye çalışılır. Şekil 3.5(c)'de m₁ üzerinde J₁ ve m₂ üzerinde J₃ çizelgelenebilir operasyonlardır. Şekil 3.5(d)'de gösterildiği gibi, m₂ üzerinde J₁ ve m₁ üzerinde J₃ gelecek çizelgelenebilir operasyonlardır. Bundan sonraki çizelgeleme operasyonları m₁ üzerinde J₁ ve m₃ üzerinde J₁

dir (Şekil 3.5(e)). Son olarak m_3 üzerinde J_3 gelecek operasyon olarak çizelgelenebilir (şekil 3.5(f)).

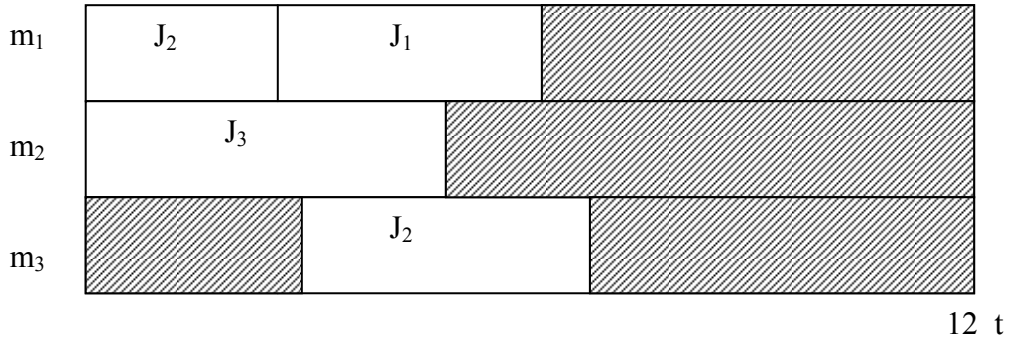
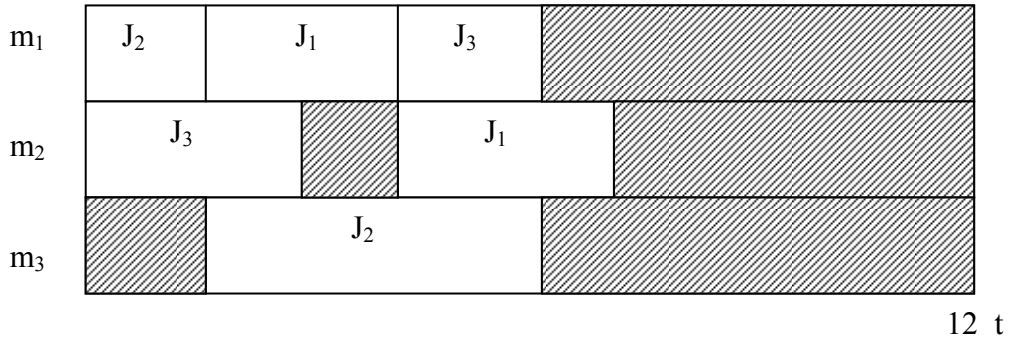
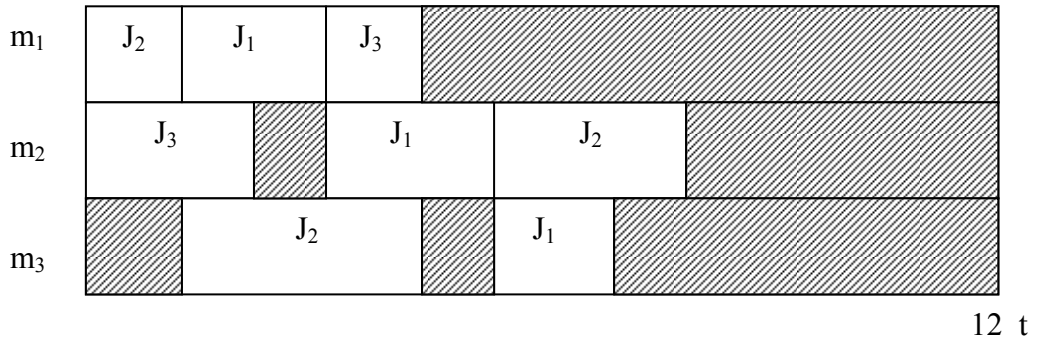
Bu işlemler sonucunda kromozomlardan çıkartılan çizelgeye göre en son işin tamamlanma zamanı 12 olarak bulunur (Cheng vd., 1996).

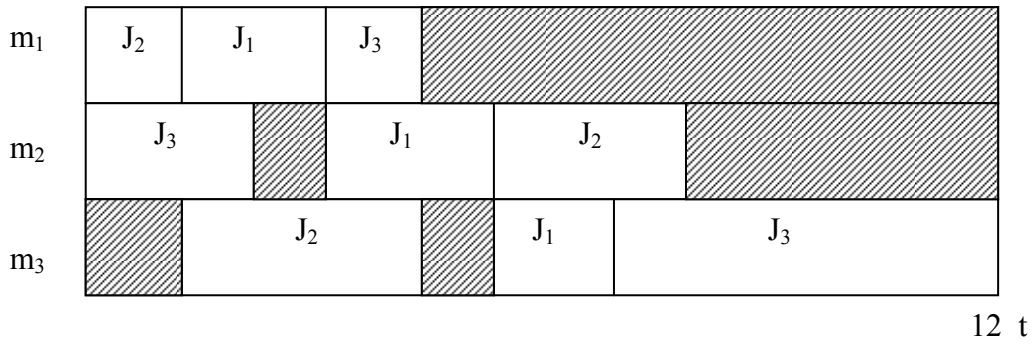


a. m_1 üzerinde iş 2 için çizelgeleme



b. m_3 üzerinde iş 2 için çizelgeleme
Şekil 3.5 Liste önceliğine dayalı kodlama

c. m₂ de iş 3 ve m₁ de iş 1 için çizelgelemed. m₂ de iş 1 ve m₁ de iş 3 için çizelgelemee. m₃ de iş 1 ve m₂ de iş 2 için çizelgeleme
Şekil 3.5 Liste önceliğine dayalı kodlama (devamı)



f.Sonuç çizelgesi
Şekil 3.5. Liste önceliğine dayalı kodlama (devamı)

3.1.4 İş Çifti İlişisine Dayalı Kodlama

Namaza ve Yamada çizelge kodlamak için ikilik düzendeki matrisi kullanmışlardır. Bu matris, makinelere uygun işlerin çift ilişkisi önceliğine göre düzenlenmesi ile elde edilmiş bir matristir (Cheng vd., 1996).

Bu kodlamanın açıklanmasında örnek olarak, 3-iş, 3-makine problemi ele alınmıştır. İşlerin operasyonlarının öncelik kısıtları ve operasyonları çizelge 3.2'de gösterilmiştir. İş çiftlerinin öncelik ilişkisini göstermek için ikilik düzende bir değişken tanımlanmıştır.

$$a_{ihk} = \begin{cases} 1, \text{ eğer } m \text{ makinede iş } j' \text{ ye göre iş } i' \text{ nin işlem görme önceliği var ise} \\ 0, \text{ diğer durumlarda} \end{cases}$$

(m_1, m_2, m_3) makinelerdeki iş çiftini, (j_1, j_2) öncelik ilişkisini gösterir. Verilen çizelgeye göre, $(x_{121} \ x_{122} \ x_{123}) = (0 \ 1 \ 0)$ elde edilir. İş ilişkisi (j_1, j_3) için (m_1, m_2, m_3) makineler üzerindeki öncelik ilişkisi $(x_{231} \ x_{233} \ x_{232}) = (1 \ 1 \ 0)$ dır. İlk iş i 'nin operasyonlarının dizilimi ile iş çiftleri için x_{ijm} de değişkenlerin dizilimi tutarlılığı muhafaza edilmelidir. Örneğin, iş çifti (j_2, j_3) için iş j_2 'nin operasyonlarının dizilimi $(1 \ 3 \ 2)$ dir.

Çizelge 3.2. Üç makine ve üç iş problemi

Operasyon Önceliği				Uygun Çizelge			
İş	Makine Dizilimi			Makine	İş Dizilimi		
J ₁	m ₁	m ₂	m ₃	m ₁	j ₂	j ₁	j ₃
J ₂	m ₁	m ₃	m ₂	m ₂	j ₃	j ₁	j ₂
J ₃	m ₂	m ₁	m ₃	m ₃	j ₂	j ₁	j ₃

Bu sonuçların özetlenmesinden, aşağıda uygun çizelge olarak ikilik düzende matris kodlaması elde edilir.

$$\begin{array}{l}
 (m_1, m_2, m_3) \text{ üzerinde } (j_1, j_2); \\
 (m_1, m_2, m_3) \text{ üzerinde } (j_1, j_3); \\
 (m_1, m_2, m_3) \text{ üzerinde } (j_2, j_3);
 \end{array}
 \quad
 \begin{pmatrix}
 x_{212} & x_{122} & x_{123} \\
 x_{131} & x_{132} & x_{133} \\
 x_{231} & x_{233} & x_{232}
 \end{pmatrix}
 =
 \begin{pmatrix}
 0 & 1 & 0 \\
 1 & 0 & 1 \\
 1 & 1 & 0
 \end{pmatrix}$$

İş çizelgeleme kodlamaları içerisinde en kompleks olanı, iş çifti ilişkisine dayanan kodlama yöntemidir.

3.1.5 Öncelik Kuralına Dayalı Kodlama

Öncelik kural tabanlı genetik algoritma, Dorndort ve Pesch tarafından geliştirilmiştir ve n-iş, m-makine problemi için, nxm giriş, (P₁, P₂,...,P_m) dizilimli bir kromozom seti kullanır.

Bu yöntemde göre, sıralamada önceliği olan işlere göre kromozom kodlaması yapılır. Genetik algoritma öncelik sırasına göre kromozomlar ile daha iyi bir sıra bulmada kullanılır. Seçilmiş öncelik kuralları çizelge 3.3'de gösterilmiştir (Cheng vd., 1996).

Çizelge 3.3. Seçilmiş öncelik kuralları

No	Kural	Tanım
1	SPT	En kısa işlem zamanlı operasyon seç
2	LPT	En uzun işlem zamanlı operasyon seç
3	MWR	En çok toplam işlem zamanı kalan iş için operasyon seç
4	LWR	En az toplam işlem zamanı kalan iş için operasyon seç

Örnek üzerinde çizelge 3.3’de verilen dört öncelik kuralı kullanılır ise, kromozom dizisi [1 2 2 1 4 4 2 1 3] olur. Bu dizide 1 SPT kuralını, 2 LPT kuralını, 3 MWR kuralını, 4 ise LWR öncelik kuralını tanımlar. İlk adımda, aşağıdakiler elde edilir,

$$S_1 = \{O_{111}, O_{211}, O_{312}\}$$

$$\Phi_1^* = \min [3 \ 1 \ 3] = 1$$

$$m^* = 1$$

$$C_1 = [O_{111}, O_{211}]$$

Makine m_1 için bu durumda O_{111} ve O_{211} operasyonları yarışır çünkü kromozomda verilen ilk gen 1’dir. Şekil 3.6(a)’da gösterildiği gibi m_1 üzerinde operasyon 211 çizelgelenir. Veri güncelleştirildikten sonra aşağıdaki elde edilir,

$$S_2 = [O_{111}, O_{223}, O_{312}]$$

$$\Phi_2^* = \min [4 \ 6 \ 3] = 3$$

$$m^* = 2$$

$$C_2 = [O_{321}]$$

Şekil 3.6(b)’de gösterildiği gibi m_2 üzerinde operasyon O_{312} çizelgelenir ve güncelleştirdikten sonra aşağıdakiler elde edilir,

$$S_3 = [O_{111}, O_{223}, O_{312}]$$

$$\Phi_3^* = \min [4 \ 6 \ 3] = 3$$

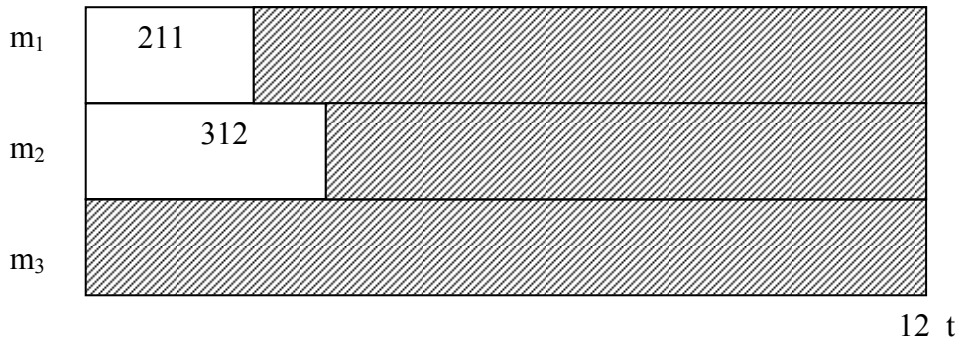
$$m^* = 1$$

$$C_3 = [O_{111}, O_{321}]$$

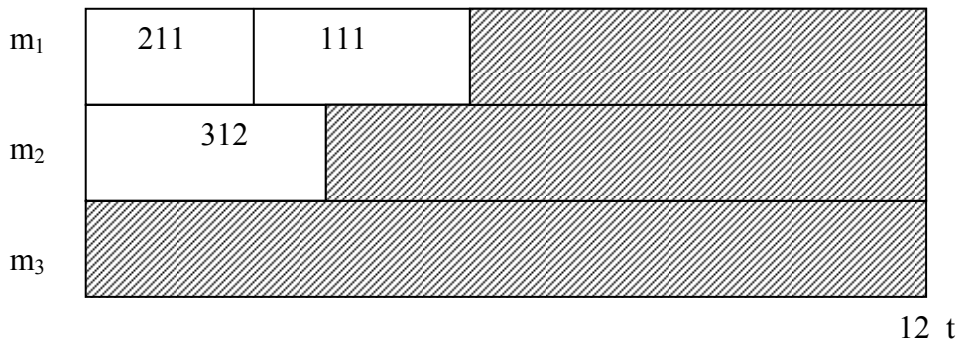
M_1 için O_{111} ve O_{321} operasyonları yarışır. Kromozomda verilen 3. gen 2 olduğundan dolayı, şekil 3.6(c)'de gösterildiği gibi m_1 üzerinde operasyon O_{111} çizelgelenir. Çizelge oluşturulana kadar bu adımlar tekrar edilir.



a. m_1 'de operasyon 211'in çizelgelenmesi



b. m_2 'de operasyon 312'nin çizelgelenmesi



c. m_1 'de operasyon 111'in çizelgelenmesi
Şekil 3.6. Öncelik kuralına dayanan kodlama

m_1	211	111	321			
m_2	312			122	232	
m_3		223			133	333

12 t

d.Sonlandırılmış çizelge
Şekil 3.6. Öncelik kuralına dayanan kodlama (devamı)

3.1.6 Tamamlanma Zamanına Dayalı Kodlama

Operasyonların tamamlanma zamanlarının sıralanmış listesi bir kromozomdur. Çizelge 3.1’de verilen örnek için aşağıdaki gibi bir kromozom oluşturulur.

$[c_{111} \ c_{122} \ c_{133} \ c_{211} \ c_{223} \ c_{232} \ c_{312} \ c_{321} \ c_{333}]$

Burada c_{jir} ’nin anlamı, r.makine üzerinde, j. işin k. operasyonu için tamamlanma zamanıdır (Cheng vd., 1996).

3.1.7 Makineye Dayalı Kodlama

Makineye dayalı genetik algoritmayı Dorndorf ve Pesch önermiştir. Makinelerin sırasına göre bir kromozom kodlanır ve sıralamaya dayalı olarak sezgisel darboğazı önleme ile bir çizelge oluşturulur.

Sezgisel darboğazı önleme, Adams tarafından önerilmiştir. Atölye tipi çizelgeleme problemleri için, bilinen sezgisel metodlar içerisinde en güçlü olanıdır. Makinelerin sıralanması birer birer yapılır. Henüz sıraya alınmamış makineler darboğaz olarak kabul edilir. Kromozomlar makine sıralama listesi oluşturur, genetik algoritma makine sıralarının bulunmasında kullanılır.

Sıralanmış makinelerin kümesi m ise kromozom, $[m_1, m_2, \dots, m_m]$ olur. Kromozomdan çizelgeleme prosedürü aşağıda açıklanmıştır (Cheng vd., 1996);

Adım 1: $M_0 \leftarrow \Phi, i \leftarrow 1$ ve kromozom $[m_1, m_2, \dots, m_m]$,

Adım 2: Makine m_i 'yi optimal bir şekilde sırala, küme $M_0 \leftarrow M_0 \cup \{M_i\}$ 'i güncelle,

Adım 3: Diğer düzenlenmiş sıralamaları muhafaza ederken, her kritik makine $m_i \leftarrow M_0$ 'ın dizilimini yeniden optimize et,

Adım 4: $i \leftarrow i+1$. Eğer $i > m$ ise, durdur; diğer durumlarda adım 2 ye geri dön.

3.1.8 Rassal Anahtarlı Kodlama

Rassal anahtarlı kodlama Bean tarafından geliştirilmiştir. Rassal sayı ile beraber, rassal anahtar kodlama bir çözümü şifreler. Çözümün şifresini çözmek için sıralı anahtar olarak bu değerler kullanılır ve n-iş, m-makine çizelgeleme problemi için her gen iki parçadan oluşturulur. Kümedeki tamsayı $[1, 2, \dots, m]$ ve $(0, 1)$ 'den rassal olarak oluşturulan bir parçasıdır. Rassal anahtarların tamsayı bölümü bu iş için makine düzenleme olarak yorumlanır.

Her makinede iş sıralama, kesirli kısımların sınıflandırılmasıyla sağlanır. Çizelge 3.1'de verilen örneğe göre, kromozom, $[1,34 \ 1,09 \ 1,88 \ 2,66 \ 2,91 \ 2,01 \ 3,23 \ 3,44]$ kabul edilir ise makine 1 için, anahtarların sınıflandırılması, iş diziliminin sıra sonuçlarını $2 \rightarrow 1 \rightarrow 3$ olarak çıkarır. Makine 2 için iş sırası; $3 \rightarrow 1 \rightarrow 2$; makine 3 için iş sırası $2 \rightarrow 1 \rightarrow 3$ olacaktır. Makine m'de iş j'yi O_{jm} gösterir ve aşağıdaki gibi operasyon sıralarının tam listesi içinde kromozom çevrilebilir, $[O_{21} \ O_{11} \ O_{31} \ O_{32} \ O_{12} \ O_{22} \ O_{23} \ O_{13} \ O_{33}]$ (Cheng vd., 1996).

3.1.9 Karışık Kodlama

Bir çizelgeleme probleminde sonsuz sayıda çizelge mevcut olduğundan bunlar üç temel sınıflandırmaya tabi tutulabilirler (Cheng vd., 1996);

1. Aktif Çizelge,
2. Yarı Aktif Çizelge,
3. Gecikmesiz Çizelgedir.

Optimal çizelge, aktif çizelge setlerinden oluşur, gecikmesiz çizelge aktif çizelgeden daha küçüktür ve optimum çözümü garanti etmez. Bütün kodlama teknikleri, iş çizelgeleme problemlerine aktif bir yapı kazandırır. Kodlama sistemlerinin farklı çizelgeleme problem

tiplerine göre avantaj ve dezavantajları vardır. Hangi tip problemlerde hangi tip kodlama sistemini kullanmanın avantajlı olacağı henüz tam olarak belirlenememiştir (Cheng vd., 1996).

3.2 Genetik Algoritmalarda Kullanılan Operatörler

Genetik algoritmalarda üç operatör kullanılır. Bunlar, üreme, çaprazlama ve mutasyon operatörleridir.

3.2.1 Üreme(Seçim) Operatörü

Bu prosedürde, bireysel diziler, amaç fonksiyonuna göre kopyalanır, gelecek nesilde daha iyi döl verebilecek bireyler seçilir. Üreme operatörü, yapay bir seleksiyondur (Goldberg, 1989).

3.2.1.1 Makine Verimlerine Bağlı Rulet Çemberi Yönetim

Bu yöntemde gen havuzunda yer alacak kromozomlar, ortalama makine verimlerine göre belirlenirler (Goldberg, 1989). Bu yöntem örnek 3.1’de açıklanmıştır.

Örnek 3.1 $f(x) = x^2$ öyle ki $[0 \leq x \leq 31]$, x belirlenen tamsayı aralığında değişiyor ise, $f(x)$ ’i maksimize edecek en uygun değeri genetik algoritma modeli ile belirleyiniz (Goldberg, 1989).

Geleneksel yöntemler kullanıldığında, böyle bir problem için, x yerine değerler denenerek maksimum $f(x)$ bulunmaya çalışılır.

Genetik algoritmada ise başlangıçta x parametresini uygun bir uzunlukta kodlamak gerekir. x parametresinin ikili sistme göre kodlanmış yapısı çizelge 3.4’de sunulmuştur.

Çizelge 3.4 İkili sistemde dizi kodlama

Dizi	Amaç Değeri
00000	0
11111	31

Bu yapıda kodlama beş genden oluşan bir yapı şeklinde tanımlanır. Bu problemin çözümü için toplam 32 adet kromozom mevcuttur. Bu kromozomlardan rassal olarak çizelge 3.5'dekilerin seçildiği varsayılır.

Çizelge 3.5 Örnek 3.1 için varsayılan başlangıç popülasyonu

No	Dizi	Amaç Değeri	Oranı(%)
1	01101	169	14,40
2	11000	576	49,20
3	01000	64	5,50
4	10011	361	30,90
TOPLAM		1170	100,0

Çizelge 3.5'de verilen değerlerin ortalama ve maksimum değerleri alınarak, rulet çemberine göre bir sonraki nesilde yer alacak kromozom sayılarının beklenen değerleri çizelge 3.6'da gösterilmiştir.

Çizelge 3.6 Örnek 3.1 için manuel rulet çemberi yöntemi

Dizi No	Başlangıç Popülasyonu	x Değeri	$F(x)=x^2$	Seçim Oranı	Beklenen Sayı(f_i/ort)	Rulet Çemberi Beklenen
1	01101	13	169	0,14	0,58	1
2	11000	24	576	0,49	1,97	2
3	01000	8	64	0,06	0,22	0
4	10011	19	0,31	0,31	1,23	1
TOPLAM			1170	1,00	4,00	4
ORTALAMA			293	0,25	1,00	1
MAKSİMUM			576	0,49	1,97	2

Çizelge 3.6'daki sonuçlara göre rulet çemberi yöntemi kullanıldığında, başlangıç popülasyonunda bulunan 1. ve 4. dizilerden birer tanesi fakat 2. diziden iki tanesi, yeni oluşturulan popülasyon içinde yer alır ve yeni oluşturulan popülasyonda 3. diziden hiç kromozom yer almaz.

3.2.1.2 Toplam Akış Zamanına Göre Rulet Çemberi Yöntemi

Bu yöntem rulet çemberinin özel durumudur. Amaç değeri yerine, toplam akış zamanı kullanılır. Maksimizasyon problemlerinin tersine, C_{\max} değeri küçük olanların bir sonraki popülasyona geçmelerine öncelik tanınır. Popülasyondaki her bireyin seçim olasılığı aşağıdaki prosedüre göre hesaplanır;

- a. Popülasyonda her birey için toplam akış zamanı hesaplanır,
- b. Popülasyondaki toplam maksimum akış zamanı (F_{\max}) bulunur,
- c. Her bireyin toplam akış zamanı ve toplam maksimum akış zamanı arasındaki farka eşit olan her bireyin uygunluk değeri hesaplanır,
- d. Uygunluk değerine dayanan her bireyin seçim olasılığı hesaplanır.

Bu değer, bireylerin uygunluk değerlerinin, popülasyondaki her bir bireyin uygunluk değerinin toplamına bölünmesine eşit olan her bireyin seçim olasılığıdır (Cheng vd., 1996).

3.2.1.3 Yapay Seçim Yöntemi

Bu yöntemde, son iki nesilden elde edilen kromozomlar, amaç fonksiyonuna göre büyükten küçüğe sıralanır. Bu sıralama içinde en iyi ve en kötü çözüm kromozomlarından, değişik oranlar alınır. Bir sonraki nesilde alınacak kromozomların ne kadarının iyi ve ne kadarının kötü olacağı genetik algoritma modelinde başlangıçta belirlenir. Bu oran bütün model boyunca sabit kalır.

3.2.1.4 Kısmi Yapay Seçim Yöntemi

Yapay seçim yöntemine benzer çalışır. Tek fark, en kötü sonuca sahip kromozomlar(en büyük akış zamanı değeri olan) tamamen ters çevrilerek(en son sırada yapılan iş ilk sıraya alınarak) gen havuzuna atılır. Bütün model boyunca bu seçim prosedürü sabit kalır. Yapay seçim yönteminde olduğu gibi, oran başlangıçta belirlenir.

3.2.2 Çaprazlama Operatörü

Genetik algoritmanın performansını etkileyen faktörlerden birisi de çaprazlama işlemidir. Doğal popülasyonlarda çaprazlama, organizmalar arasında yapılan döllenme çalışmaları olarak bilinir (Oraler, 1990). Rulet çemberi yöntemi ile yapay seçim sonucunda elde edilen yeni popülasyon dizisinde rassal olarak iki kromozom seçilir ve karşılıklı çaprazlama işlemine tabi tutulur. Çaprazlama işleminde, L dizi uzunluğu olmak üzere,

$1 \leq k \leq L-1$ aralığında üniform olarak k tamsayısı seçilir. Bu tamsayı değerine göre dizi çaprazlamaya tabi tutulur.

Başlangıç popülasyonunda A_1 ve A_2 gibi iki dizi rassal olarak seçilmiş olsun.

$$A_1 = 0 \ 1 \ 1 \ 0 \ 1$$

$$A_2 = 1 \ 1 \ 0 \ 0 \ 0$$

Bu örnekte dizi uzunluğu, $L=5$ dir. Bir ile beş arasında, $k=4$ seçildiğini varsayarsak, her iki dizinin de 4. genden sonra karşılıklı çaprazlama işlemine tabi tutulacağı kabul edilir. Bu durumda yeni kromozomlar aşağıdaki gibi olacaktır.

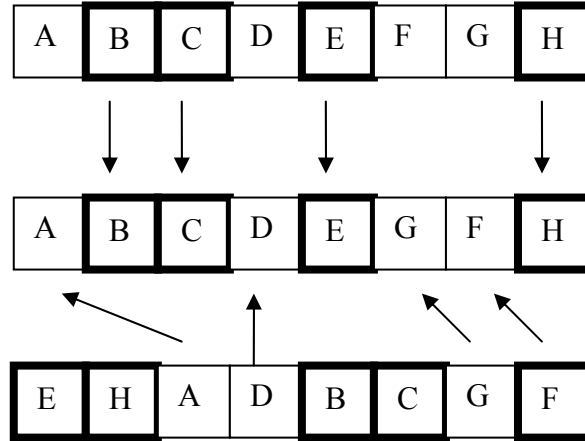
$$A_1' = 0 \ 1 \ 1 \ 0 \ 0$$

$$A_2' = 1 \ 1 \ 0 \ 0 \ 1$$

En basit çaprazlama yöntemi, yukarıdaki örnekte açıklanan bir noktalı çaprazlama yöntemidir. Ancak bir noktalı çaprazlama yapılabilmesi için her iki kromozomun da aynı gen uzunluğunda olması gerekir. İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve karşılıklı olarak pozisyonlar yer değiştirir (Croce vd., 1995). Akış tipi çizelgelerde genellikle bir ve iki noktalı çaprazlama yöntemi kullanılır (Cheng vd., 1999a).

3.2.2.1 Pozisyona Dayalı Çaprazlama

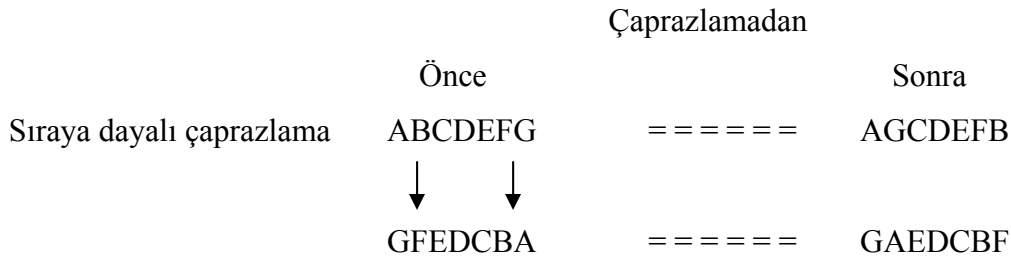
Şekil 3.7'de görüldüğü gibi rassal olarak seçilmiş pozisyondaki işler bir ebeveynden çocuğa kalıtsallaştırılır. Diğer işler, diğer ebeveynlerdeki görünümleri sırasıyla yerleştirilirler. Pozisyondaki sayılar, $[1, n]$ rassal tamsayılar olarak öncelikle düzenlenir, daha sonra bu pozisyonlar rassal olarak seçilir ve her pozisyonun çaprazlanma olasılığı %50 olarak kabul edilir (Murata vd., 1996a).



Şekil 3.7 Pozisyona dayalı çaprazlama

3.2.2.2 Sıraya Dayalı Çaprazlama

Sıraya dayalı çaprazlama ilk olarak Davis tarafından önerilmiştir (Cheng vd., 1999b). Bu yöntemde bir grup nokta rasgele seçilir. Birinci kromozomun seçilen noktalara karşılık gelen karakterleri aynen yerlerini korur. İkinci kromozomun seçilen noktalara ait karakterleri birinci kromozomun aynı noktalarındaki karakterlerin önüne getirilir. Geriye kalan boş pozisyonlara ikinci kromozomdan aktarılan yeni karakterler de göz önünde bulundurularak ilk kromozomun kullanılmayan karakterleri tarafından sıra ile (soldan sağa) yerleştirilerek yeni bir kromozom elde edilir. Bu tür çaprazlama, kromozomu oluşturan karakterlerin sayı ve sıralarının önem taşıdığı durumlarda kullanılır. bu tür çaprazlama işlemine ait bir örnek şekil 3.8’de verilmiştir.



Şekil 3.8. Sıraya dayalı çaprazlama

3.2.2.3 Kısmi Planlı Çaprazlama(PMX)

Goldberg tarafından geliştirilen bu çaprazlama ilk olarak gezgin satıcı probleminde kullanılmıştır (Goldberg, 1989). Bu yöntem örnek 3.2’de açıklanmıştır.

Örnek 3.2 Kısmi planlı çaprazlama operatörünün prosedürünü göstermek için 8 iş problemi oluşturulmuştur. Çaprazlama için seçilen ebeveyn yapılar A ve B olarak adlandırılmıştır. Yapıdaki elemanlar aşağıda gösterilmektedir.

$$A = 2 \ 8 \ 6 \ 4 \ 5 \ 7 \ 1 \ 3$$

$$B = 8 \ 7 \ 2 \ 1 \ 3 \ 4 \ 5 \ 6$$

A ve B’ye PMX operatörü uygulanır ise, ilk olarak A ve B’den ortak bir aralık rassal olarak seçilir. Daha sonra seçilmiş iki aralıktaki elemanların planları belirlenir. Bu örnekte, seçilmiş aralıklar arasındaki plan, 6-2, 4-1 ve 5-3’tür. İkinci olarak A ve B’deki iki aralık değiştirilir. Bu değişim sonucu aşağıdaki yapılar geçici sonucu gösterir.

$$\begin{array}{l} A = \quad 2 \quad 8 \quad | \quad 2 \quad 1 \quad 3 \quad | \quad 7 \quad 1 \quad 3 \\ B = \quad 8 \quad 7 \quad | \quad 6 \quad 4 \quad 5 \quad | \quad 4 \quad 6 \quad 5 \end{array}$$

Bir dizide aynı işten birden fazla olduğundan her iki yapının da uygun olmadığı gözlenir. Bundan dolayı, yeni yapıları uygun olmayan A ve B’de, yarleştirilemeyen daha önceki adımda belirlenen elementlerin planının değiştirilmesi gerekir. Bu örnekte A yapısının, 1, 7 ve 8. pozisyonlarında; 2, 1 ve 3 ile sırasıyla 6, 4 ve 5 yer değiştirir. Yeni yapı aşağıda gösterildiği gibi oluşturulur.

$$\begin{array}{l} A' = \quad 6 \quad 8 \quad | \quad 2 \quad 1 \quad 3 \quad | \quad 7 \quad 4 \quad 5 \\ B' = \quad 8 \quad 7 \quad | \quad 6 \quad 4 \quad 5 \quad | \quad 1 \quad 2 \quad 3 \end{array}$$

3.2.2.4 Dairesel Çaprazlama(CX)

Dairesel çaprazlama yöntemi Davis, Goldberg ve Linge tarafından geliştirilmiş bir yöntemdir. Yöntem aşağıdaki örnek üzerinde gösterilmiştir (Goldberg, 1989).

3.2.2.6 Sıralı Çaprazlama(OX)

Bu yöntem Davis, Goldberg ve Linge tarafından geliştirilmiştir. Yöntemin uygulanışı örnek 3.4’de gösterilmiştir (Goldberg, 1989).

Örnek 3.4 A ve B olarak kodlanmış iki kromozom aşağıda gösterilmektedir (Goldberg, 1989).

$$\begin{array}{l} A = 9 \ 8 \ 4 \mid 5 \ 6 \ 7 \ 1 \mid 3 \ 2 \ 10 \\ B = 8 \ 7 \ 1 \mid 2 \ 3 \ 10 \ 9 \mid 5 \ 4 \ 6 \end{array}$$

Sıralı çaprazlama yönteminde, 5, 6 ve 7 genleri yerine, 2, 3 ve 10 genleri atanır ve A kromozomunda daha önce 2, 3 ve 10 bulunan yerlere “H” yazılır. Buna göre aşağıdaki ifade elde edilir.

$$B = \mid 8 \ H \ 1 \ 2 \mid 3 \ 10 \ 9 \ H \ 4 \ H$$

Burada sola doğru kayma işlemi yapıldığında aşağıdaki ifade elde edilir.

$$B = 2 \ 3 \ 10 \ H \ H \ H \ 9 \ 4 \ 8 \ 1$$

H yerine dizide olmayan işler eklendiğinde yeni kromozomlar aşağıdaki gibi olur.

$$\begin{array}{l} A' = 5 \ 6 \ 7 \mid 2 \ 3 \ 10 \mid 1 \ 9 \ 8 \ 4 \\ B' = 2 \ 3 \ 10 \mid 5 \ 6 \ 7 \mid 9 \ 4 \ 8 \ 1 \end{array}$$

3.2.2.7 Alt Değişimli Çaprazlama(SXX)

Kobayashi, Brady ve Mühlenbein tarafından farklı zamanlarda yapılan çalışmalar ile gezgin satıcı problemi için geliştirilmiş bir çaprazlama yöntemidir. Bir iş sırası matrisinde, n-iş, m-makine probleminde nxm matris kodu yazılabilir. Bu koddaki her bir gen bir iş operasyonuna karşılık gelmektedir. Operasyon şeklinde kodlanan çizelgeleme problemlerinde iki işlem adımı ile SXX çaprazlama metodu kullanılır. Önerilen yöntemin işlem adımları aşağıdaki gibidir (Cheng vd., 1999b);

1. Her bir dizideki genler makine bazında tanımlanır,

2. Dizideki her bir geni makine olarak deęiřtirip yeni nesiller oluřturulur.

3.2.2.8 İř Tabanlı Sıralı aprazlama

One ve arkadaşları tarafından geliřtirilen bu yöntem, alt deęiřimli aprazlama metodunun bir varyantıdır. Bu metodun geliřtirilmesinin nedeni, iř izelgelemede bütn iřlerin mutlaka izelgede yer alması zorunluluęundan kaynaklanmaktadır. İř tabanlı sıralı aprazlama, iř sıra matrisinin kodlanması řeklindeyir. Yntem ařaęıdaki iřlem adımlarından oluřur (Cheng vd., 1999b);

1. Her bir makineden geen iř kmeleri, kromozomlar zerinde belirlenir,
2. Belirlenen iřler, ilk birey iin makineye baęlı olarak ebeveynlerden kopyalanır, aynı iřlemler ikinci birey iin gerekleřtirilir.
3. Seilmeyen iřleri de dikkate alarak ilk bireyin kromozomu, btn genleri ile oluřturulur. Benzer řekilde dięer bireyler iin de aynı iřlemler gerekleřtirilir.

3.2.3 Mutasyon Operatr

Genetik algoritmada nemli rol oynayan proseslerden biri de mutasyon operatrdr. Canlılarda gen rekombinasyonlarının dıřındaki dięer nedenlerle ve ani olarak meydana gelen kalıtsal deęiřimlere mutasyon denir. Doęal poplasyonlarda mutasyon iřlemi iře,

- 1.Kromozom yapısı deęiřmeleri,
- 2.Kromozom sayısı deęiřmeleri,
- 3.Gen yapısındaki fiziksel ve kimyasal deęiřimler gerekleřir (Kuru, 1987).

Yapay sistemlerde mutasyon iřlemi esnasında kromozomdaki gen sayısı deęiřmez, sabit kalır. Doęal poplasyonlarda mutasyon oranı olduka dřktr. Mutasyon frekansının byklę, genetik algoritmanın performansını etkilemektedir. rneęin, kromozom uzunluęu 20 gen olan bir dizide mutasyon oranı % 0,1 seildięinde, kromozom zerinde mutasyondan dolayı beklenen deęiřme olasılıęı,

$$20 \times 0,001 = 0,02$$

bulunur. Bu durumda yeni nesil oluşuncaya kadar mutasyon işlemi gerçekleştirilemez. Genetik algoritmalarda, popülasyonların mutasyon oranı genellikle düşük seçilir (Goldberg, 1989).

Mutasyon işlemi bir tek koromozom üzerinde yapılır. Mutasyon frekansına göre, mutasyona uğratılacak sayıdaki diziler popülasyondan rassal olarak seçilir ve belirlenen mutasyon yöntemine göre değişime uğratılır.

Son on yılda çok çeşitli mutasyon yöntemleri geliştirilmiştir. İş çizelgeleme ile ilgili yöntemler aşağıda açıklanmıştır (Murata vd., 1996b);

1. Ters Mutasyon :

Bir kromozomda rassal olarak iki pozisyon seçilir. Bu iki pozisyondaki alt diziler ters çevrilir.

2. Komşu İki İş Değiştirme :

Şekil 3.9(a)'da gösterildiği gibi rassal olarak seçilen iki komşu iş değiştirilebilir.

3. Keyfi İki İş Değiştirme :

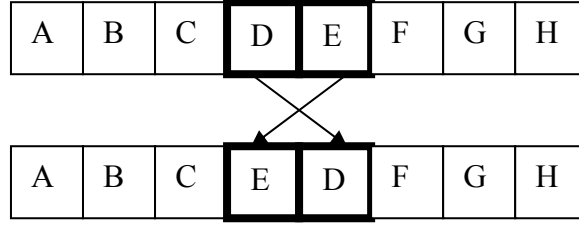
Şekil 3.9(b)'de gösterildiği gibi rassal olarak seçilen iki iş değiştirilebilir. Özel bir durum olarak, değiştirilebilen iki komşu işi bu mutasyon içerir.

4. Keyfi Üç İş Değiştirme :

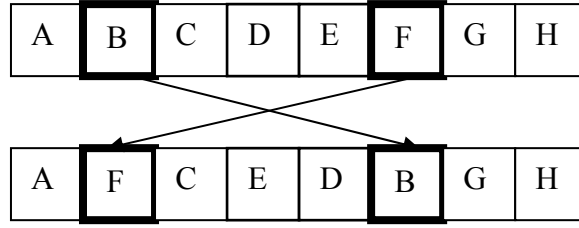
Şekil 3.9(c)'de gösterildiği gibi rassal olarak seçilen üç iş keyfi olarak değiştirilebilir.

5. Araya İş Ekleyerek Değiştirme :

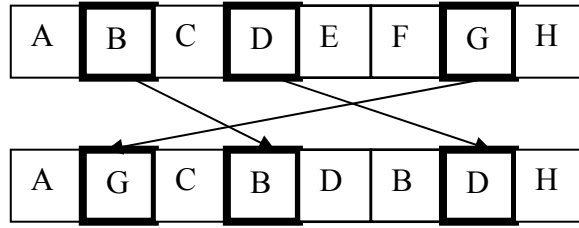
Bu mutasyon çeşidinde, şekil 3.9(d)'de gösteriliği gibi, bir operasyondaki bir iş kaydırılır ve diğer bir pozisyona konur. Kaydırma noktası rassal olarak seçilir. Bu yöntem komşu iki iş değiştirmenin özel bir durumudur. Keyfi üç iş değiştirme ile bir kesişime sahiptir.



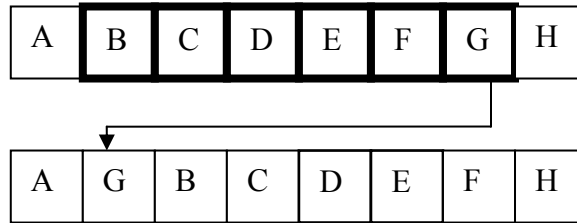
a. Komşu iki işi değiştirme



b. Keyfi iki iş değiştirme



c. Keyfi üç iş değiştirme



d. Araya iş ekleyerek değiştirme

Şekil 3.9. Farklı mutasyon operatörleri

3.3 Genetik Algoritma'nın Gelişimi ve Kullanım Alanları

3.3.1 Genetik Algoritma'nın Tarihsel Gelişimi

Genetik algoritma kavramından ilk bahseden ve konu ile ilgili ilk yayını yapan Bagley (1967) dir. Bagley ile aynı tarihte Rossenberg, biyolojik ve simülasyon esaslı bir çalışma yapmıştır (Goldberg, 1989).

De Jongs, 1975 yılında matematiksel fonksiyonları genetik algoritma ile çözmeye çalışmıştır. De Jong, fonksiyonların minimizasyonu için beş ayrı problem incelemiştir (Goldberg, 1989). Bunlar;

1. Sürekli ve sürekli olmayan fonksiyonlar,
2. Konveks ve konveks olmayan fonksiyonlar,
3. Tek ve çok değişkenli modeller,
4. Düşük ve yüksek dizili fonksiyonlar,
5. Deterministik ve stokastik problemlerdir.

David E. Goldberg'in 1985 yılındaki çalışmaları genetik algoritmanın gelişimini sağlamıştır. Bu çalışmada Goldberg'in amacı, doğalgaz borularındaki kayıpları, basınç oranını değiştirmek suretiyle minimize etmektir. Dinamik programlama yardımı ile Wong ve Larson tarafından basınç oranları hesaplanan bu problem, Goldberg tarafından genetik algoritma ile çözülmeye çalışılmıştır. Bu problemde amaç, kompresörlerin enerjilerini minimize etmektir (Goldberg, 1989).

3.3.2 Çok Amaçlı Optimizasyon

Bütün optimizasyon ve araştırma problemleri, mevcut durumu bir tek kriterle değerlendirir ve bu kriter amaç fonksiyonu olarak veya genetik algoritmalarda uygunluk fonksiyonu olarak adlandırılır. Bu yaklaşım birçok problemde başarı ile uygulanır ancak aynı anda birçok kriterin olduğu problemleri tek kriter haline getirmek oldukça zordur. Bu durumda çok amaçlı veya çok kriterli problemler ortaya çıkar (Goldberg, 1989).

Çok amaçlı modeller, tasarım, planlama modellerinde, günlük hayattaki karmaşık sistemlerin modellenmesinde, ulaşım problemlerinde, sermaye, bütçeleme problemleri gibi problemlerde kullanılmaktadır.

Genetik algoritma çok amaçlı optimizasyon problemlerinin çözümünde başarı ile uygulanmaktadır. Bu amaç doğrultusunda, genetik çok amaçlı optimizasyon modelleri geliştirilmiştir. Çok amaçlı problemlerin optimizasyonu konusunda ilk çalışmaları yapan Schaffer'dir (Goldberg, 1989). Fonseca ve Fleming, Horn, Tamaki, Kita ve Kabayashi bu çalışmaları devam ettirmişlerdir (Gen ve Cheng, 2000).

Genetik algoritma, birbirinden farklı çok amaçlı optimizasyon problemlerinde rahatlıkla kullanılabilir. Genetik algoritma, popülasyondan popülasyona global arama yaptığından, matematiğe ihtiyaç duymamaktadır. Bu sebeple oldukça kompleks olan problemler genetik algoritma ile çözülebilmektedir çünkü genetik algoritma bir çeşit meta sezgisel metottur (Gen ve Cheng, 2000).

Çok amaçlı optimizasyon problemlerinin genetik algoritma ile çözümünde ilk ortaya çıkan sorun, genetik algoritma'da kullanılan uygunluk değerinin nasıl belirleneceğidir. Uygunluk fonksiyonunun belirlenmesi ile ilgili son on yılda çeşitli yaklaşımlar geliştirilmiştir (Gen ve Cheng, 2000). Bunlar;

1. Vektör değeri yaklaşımı(VEGA)
2. Ağırlıklar toplamı yaklaşımı
3. Pareto tabanlı yaklaşım
4. Uzlaşma yaklaşımı
5. Amaç programlama yaklaşımıdır.

Çok amaçlı genetik algoritma optimizasyonlarında ilk geliştirilen yaklaşım, VEGA'dır. Bu yaklaşımda belirli bir ölçekteki uygunluk değeri yerine, her bir kromozom ile geliştirilen uygunluk değeri kullanılır.

Ağırlıklar toplamı yaklaşımında, her bir amaç fonksiyonuna bir ağırlık değeri verilir ve bu ağırlıkla amaç fonksiyonları tek bir fonksiyon haline getirilir. Sabit ağırlık yönteminde, nesil sayısı boyunca amaç denklemlerinin ağırlıkları değişmez. Murata, Ishibuchi ve Tanaka, rassal ağırlıklı bir yaklaşım önermişlerdir (Gen ve Cheng, 2000). Bu yaklaşımda, her nesilde rassal

olarak amaç fonksiyonları için yeni ağırlık değerleri oluşturulmaktadır. Pareto yaklaşımı Horn, Nafpliotis ve Goldberg tarafından geliştirilmiştir.

3.3.3 Çizelgeleme Problemleri

Çizelgeleme, iyi bilinen zor kombinatoriyal optimizasyon problemlerinden biridir. Son otuz yılda bu problemlerin çözümü için yeteri kadar araştırma yapılmış ve öncelik kurallı ve dal sınır tekniğine dayanan çeşitli yöntemler geliştirilmiştir. Bu tip problemler, NP türü problemler kapsamında olduğundan, bilgisayar hızlarının artması ile daha etkin çizelgeleme yöntemleri geliştirilmeye çalışılmıştır.

Çizelgeleme problemlerinde temel amaç, permutasyon tipi, her bir operasyonun herbir makinedeki yapılış sırasını belirleyerek minimum tamamlanma süresini elde etmektir.

Genetik algoritmaların çizelgeleme problemlerinde kullanımı iki farklı şekilde gerçekleşmiştir (Gen ve Cheng, 2000). Bunlar;

1. Genetik algoritma yardımı ile permütasyon tipi iş sırasını bulma,
2. Genetik algoritma ile bulunan permütasyon sırasının, bilinen sezgisel yöntemler ile karşılaştırılması.

Genetik algortmada optimuma yakın çözüm veren parametreler tam uygun olmadığından kısa sürede uygun çözüme ulaşılması için üç kategoride çalışmalar yapılmaktadır (Gen ve Cheng, 2000). Bunlar;

1. Genetik operatörlerin belirlenmesi ile ilgili çalışmalar
2. Genetik algoritma ile sezgisel özellikte operatör geliştirme
3. Melez (hybrid) genetik algoritma geliştirme.

Giffler ve Thompson algoritma tabanlı çaprazlama operatörü, genetik algoritma ile ilgili sezgisel özellikte geliştirilmiş bir operatördür. Bu çaprazlama yöntemi, Yamada ve Nakano tarafından geliştirilmiştir. Komşuluk araştırma tabanlı mutasyon yöntemi de bu kategoride geliştirilmiş bir yöntemdir.

Melez genetik algoritma kapsamında yapılan çalışmalarda, genetik algoritma ile lokal aramanın birleştirilmesi, genetik algoritma ile Giffler Thompson metodunun birleştirilmesi ve

genetik algoritma ile darboğaz giderme sezgisel yönteminin birleştirilmesi örnek olarak verilebilir.

Genetik algoritmalar ile ilgili ilk iş çizelgeleme çalışmasını Davis yapmıştır (Davis, 1985). Chen, Vempati ve Aljaber, tamamlanma zamanı kriterli akış tipi çizelgeleme problemleri için sezgisel tabanlı genetik algoritma kullanmışlardır. Akış tipi 4-makine, 7-iş, 5x10, 8x15, 10x25 problemlerini genetik algoritma ile çözmüş ve elde edilen tamamlanma zamanı değerlerini, mevcut sezgisel yöntemler ile karşılaştırarak (NEH, CDS) genetik algoritma'nın iyi performans verdiğini belirtmişlerdir (Chen vd., 1995).

Reeves, akış tipi çizelgeleme (n-iş; m-makine) problemlerini, genetik algoritma ile çözüp, elde edilen sonuçları komşuluk arama ve tavlama benzetimi ile karşılaştırmıştır. Çalışma sonucunda genetik algoritma'nın daha iyi performans verdiğini belirlenmiştir (Reeves, 1995b).

Murata, Ishibuchi, ve Tanaka, akış tipi çizelgeleme problemlerinin genetik algoritma çözüm değerlerini, diğer arama yöntemleri olan, yerel arama, tabu araştırmaları ve tavlama benzetimi yöntemleri ile karşılaştırmışlardır. Elde edilen sonuçlara göre, genetik algoritma biraz daha iyi sonuç vermiştir. Genetik algoritma'nın performansın artırılması için iki melez genetik algoritma önerilmiştir. Bunlar, genetik lokal arama ve genetik tavlama benzetimidir (Murata vd., 1996a).

Murata, Ishibuchi, ve Tanaka, diğer bir çalışmalarında, akış tipi çizelgeleme problemleri için, çok amaçlı bir genetik algoritma kullanmışlardır. Bu algoritmanın performansı iki amaca göre belirlenmiştir: tamamlanma zamanı ve toplam gecikmenin minimize edilmesi (Murata vd., 1996b).

Chen ve arkadaşları, akış tipi çizelgeleme problemlerinin çözümünde genetik algoritma kullanarak, toplam akış zamanını minimize etmeye çalışmışlardır. Genetik algoritma ile çözülen akış tipi problemlerde daha küçük akış zamanına ulaşılmıştır (Chen vd., 1996).

Jain ve Bagchi, akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm etkinliğinin artırılmasına yönelik olarak, Darwin ve Lamarck tabanlı genetik algoritmaların karşılaştırmasını yapmışlardır. Genetik algoritmalar'ın performansının artırılmasında öğrenen temelli genetik algoritma modellerini önermişler ve 15-makine, 49-iş problemini, %70 çaprazlama oranı ile genetik algoritma yardımı ile çözmüşlerdir (Jain ve Bogchi, 2000).

4. SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ

Araştırmalarda ele alınan pekçok çizelgeleme probleminde, belirli bir dağılım içerisinde rassal sayılar kullanılarak oluşturulmuş işlem zamanı matrisi kullanılır. Bu tip, verisi rassal olarak elde edilen problemler çözüm prosedürlerinin verim ve etkinliğini ölçmede kullanılırlar. Ancak herhangi bir dağılımdan rassal olarak elde edilen veriler kullanan araştırma problemleri gerçek hayatta karşılaştığımız problemleri anlamada sınırlı bir taban oluştururlar. Araştırmacılar çalışmalarında, teorik problemlerin, işlerin ve tesisin fiziksel karakteristikleriyle orantılı işlem zamanlarına sahip gerçek problemlerle karşılaştırıldıklarında, çözüm açısından çok zor yada çok kolay olmadıklarını gözlemlemişlerdir.

Akış tipi çizelgeleme problemleri içerisinde rassal olmayan bir çeşidi, “sıralı işlem zamanlı akış tipi çizelgeleme problemleridir” (Smith, 1968).

Smith, Panwalkar ve Dudek, n-iş, m-makine probleminin bir alt kategorisi olan “sıralı akış tipi çizelgeleme problemini” tanımlamışlardır (Smith vd., 1975). Bir akış tipi çizelgeleme problemi aşağıdaki iki özelliği sağlaması durumunda sıralı problem olarak adlandırılır:

- (1) Herhangi bir iş herhangi bir makinede aynı makine üzerindeki ikinci işten daha küçük işlem zamanına sahipse, birinci iş ilgili tüm makinelerde ikinci işten küçük yada eşit işlem zamanına sahiptir.
- (2) Herhangi bir iş, bir makinede j. küçük işlem zamanına sahipse, diğer tüm işler aynı makinede j. küçük işlem zamanına sahiptir. Başka bir deyişle, verilen bir iş için minimum işlem zamanına sahip bir makine, diğer tüm işler için de minimum işlem zamanına sahiptir.

4.1 Sıralı Problemlerin Uygulama Temelleri

Sıralı problemler ile ilgili uygulamalara çeşitli endüstrilerde rastlanmaktadır. İşlem zamanlarının işlere göre sıralanması pek çok faktörden kaynaklanır. Bunlardan bir tanesi, işlerin birim yada parça sayısıdır. Eğer her iş için operasyonların yapısı oldukça benzer ancak işler üretim sayılarında büyük farklar içeriyorsa, bütün işler için işlem zamanı birimleri bu farklılığı sıralı bir yapı oluşturarak yansıtır.

İşlem zamanlarını etkileyen bir diğer faktör ise işlerin karmaşıklığıdır. Bir iş her birimde çok basit işlemler gerektirirken başka bir iş zaman harcayıcı işlemler gerektirebilir. Bu durumda, işlerin kompleksliği tüm makinelerdeki işlerin işlem zamanları için ciddi bir etki oluşturur.

Benzer şekilde, işlem zamanlarının makinelere göre sıralanması makinelerdeki operasyonların farklı yapılarının sonucudur. Eğer bir makinedeki operasyon tüm işler için karmaşıksa, ancak başka bir makinedeki operasyon tüm işler için daha basit ise, ilk makinedeki birim işlem zamanları ikinci makineye oranla daha büyük olacaktır. Bu operasyon karmaşıklığı tüm işleri etkileyeceğinden sıralı işlem zamanı problemlerinin ikinci karakterisliğinin temel sebebidir (Smith vd., 1975).

4.2 Sıralı Akış Tipi Çizelgeleme Problemi Çeşitleri

Smith, Panwalkar ve Dudek sıralı akış tipi problemleri ele almaya başladıklarında pek çok problem üzerinde çalıştılar ve bu sıralı problemler üzerinde iki önemli karakteristik gözlemler yaptılar. Bunların ilki, eğer sıralı bir problemdeki maksimum işlem zamanı ilk (son) makinede meydana gelmişse, işlerin sıralı işlem zamanlarını azalan(artan) sıra ile sıralamak daima minimum tamamlanma zamanlı sıralamayı vermektedir. İkincisi ise, eğer sıralı bir problemdeki maksimum işlem zamanı ara makinelerden birinde gerçekleşmiş ise, sıralı işlem zamanları artan, kalan işlerin ise azalan sıralaması şeklinde takip ettiği bir sıralama alt kümesi daima minimum tamamlanma zamanlı sıralamayı vermektedir (Smith vd., 1976).

Bu gözlemlere dayanarak Smith ve diğerleri, sıralı akış tipi çizelgeleme için iki temel problem tipi tanımladılar ve bu problem tiplerinin çözümü için iki farklı algoritma önerdiler. Bu problemlerden ilki, genel durum olan tüm işler için sıralı işlem zamanı matrisinde işlem zamanı en büyük olan işin ara makinelerden birinde oluşması durumudur. İkinci problem ise genel durumun özel bir gelişimi olan tüm işler için maksimum işlem zamanı işin ilk yada son makinede oluşması durumudur (Smith vd., 1976).

4.2.1 Genel Durum Problemi

Sıralı akış tipi çizelgeleme problemlerinde, genel durum olarak tanımlanan problem çeşidinde, işler içerisinde maksimum işlem zamanı ara makinelerden birinde oluşmaktadır. Bu problem çeşidinin çözümü için önerilen algoritma temelde, 2^{n-1} sıralamanın oluşturulduğu

sınırlı deneme yanılma yönteminin bir çeşididir. Önerilen algoritma için işlem adımları aşağıdaki gibidir (Smith vd., 1976):

- Adım 1. İşleri işlem zamanı matrisindeki ilk makinenin işlem zamanlarına göre artan sıralama ile sıralayın. (işlem zamanlarındaki eşitlik durumunda, ikinci yada üçüncü makinedeki işlem zamanları dikkate alınır)
- Adım 2. Oluşturulan parçalı sıralama yapısı içinde boş en soldaki pozisyona sıralanmamış en düşük değerli işi yerleştirin. Aynı işi parçalı sıralama yapısı içerisinde boş en sağdaki pozisyona da yerleştirin.
- Adım 3. Adım 2'yi her sıralamaya ilk n-1 iş yerleşene kadar devam edin. Son olarak en yüksek değerli işi sıralamalardaki boş noktaya yerleştirin.
- Adım 4. Tamamlanma zamanı için 2^{n-1} sıralamayı değerlendirin. En kısa tamamlanma zamanını veren sıralama optimaldir.

Yukarıdaki işlem adımlarının açıklanması için örnek 4.1'de 4-iş, 4-makine problemi ele alınmıştır.

Örnek 4.1 4-iş, 4-makine problemine ait işlem zamanları çizelge 4.1'de gösterilmektedir. İşlem zamanlarını gösteren matris “sıralı matris” olarak adlandırılmaktadır (Smith vd., 1976).

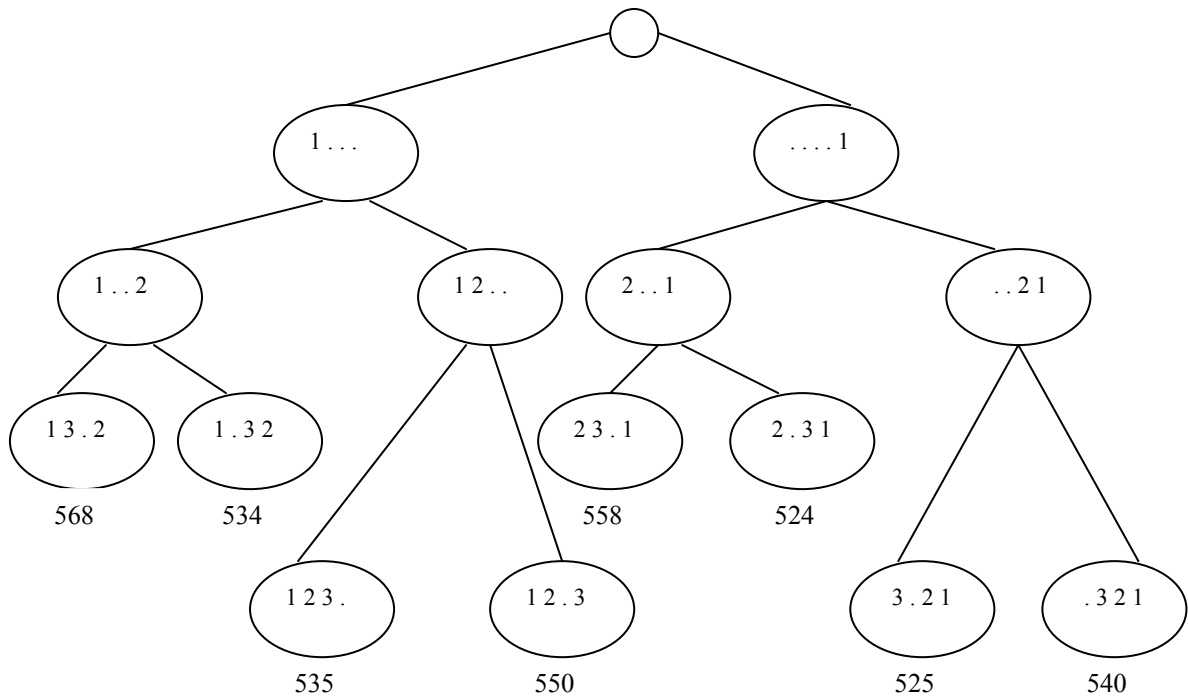
Çizelge 4.1 4-iş, 4-makine problemi için sıralı matris

İşler	Makineler			
	A	B	C	D
1	42	56	44	32
2	66	88	69	51
3	69	92	72	53
4	89	119	93	68

Örnek 4.1 ile ilgili her seviye için parçalı sıralama ve tamamlanmış sıralama şekil 4.1'de gösterilmektedir. Optimum sıralama [2 4 3 1] şeklinde 524 birim tamamlanma zamanı ile oluşmuştur.

Şekil 4.1'de gösterilen çözüm yöntemi, dal-sınır arama yönteminin bir benzeridir. Ancak çözüm içerisindeki pekçok nokta uygun üst sınır değerleri hesaplanarak elemine edilebilir.

Algoritma tarafından oluşturulan tüm sıralamalarında gözlemlenen bir karakteristik, en uzun işlem zamanına sahip işin yeri sabitlendiğinde, başlangıç işinden en uzun işlem zamanı olan işin bir solundaki işe kadar önceki tüm işler artan bir sıralama ile yerleşmektedir, şeklindedir. Benzer şekilde, en uzun işlem zamanlı işten sonraki işler de azalan sıralama ile yerleşirler. Bu karakteristiği sağlayan bir sıralama şekil 4.1’de görüldüğü gibi “piramit” şeklinde olacaktır. Genel durum problemi için önerilen çözüm yöntemi en az bir optimal çözüm vermektedir (Smith vd., 1976).



Şekil 4.1 Geliştirilen çözümün grafik gösterimi

Sıralı akış tipi çizelgeleme probleminin çözümünden elde edilen sıralama, örnek 4.1’de de görüldüğü gibi, yalnızca permutasyon sıralamasının sınırlandırıldığı durumlarda (işlerin tüm makineler için aynı sırayla sıralanması) optimaldir. Bu durum çizelge 4.2’de gösterilen 2-iş, 4-makine probleminde daha anlaşılır ifade edilebilmektedir. Bu problem, sıralı matris problemlerinin tüm koşullarını sağlamakta ve 29 birim tamamlanma zamanı ile [1 2] sıralamasında optimal permutasyon sıralaması elde edilmektedir. Ancak çizelge 4.3’de verilen genel sıralamada akış zamanı 28 olarak elde edilir. Bu problemin gösterdiği gibi, genel sıralamalar ile sıralı matris problemlerinde elde edilen optimal permutasyon sıralamasından daha küçük toplam akış zamanları elde edilebilir (Smith vd., 1976).

Çizelge 4.2 2-iş sıralı matris problemi

İşler	Makineler			
	A	B	C	D
1	8	6	4	7
2	5	2	1	4

Çizelge 4.3 Çizelge 4.2'deki problem için genel sıralama

Makine	Sıralama
A	1 2
B	1 2
C	2 1
D	2 1

4.2.2 Özel Durum Problemi

Sıralı akış tipi çizelgeleme problemlerinde ikinci problem çeşidi ise genel durumun özel bir gelişimi olan tüm işler için maksimum işlem zamanlı işin ilk yada son makinede oluşması durumudur. Bu problemin çözümünün gelişimi sırasında Dudek ve Teuton'un (Dudek, Teuton, 1964) genel varsayımları kullanılmıştır. Problemin optimizasyon kriteri olarak toplam akış zamanının minimizasyonu ele alınmıştır. Problemin çözüm adımları aşağıdaki gibidir (Smith vd., 1975).

Adım 1. İşleri herhangi belirli bir makinedeki işlem zamanlarına göre artan sıralama ile sıralayın.

Adım 2. Eğer en büyük işlem zamanları son makinede gerçekleşmişse, işleri birinci adımda elde edilen sıralamaya göre sıralayın. Eğer ilk makine en büyük işlem zamanlarına sahipse, işleri birinci adımda elde edilen sıralamanın tersine göre sıralayın. Eğer ilk ve son makine eşit işlem zamanlarına sahip ise, iki sıralama da optimum sonuç verecektir.

Yukarıdaki algoritmanın toplam akış zamanı için optimum sonuç verdiği Smith, Dudek ve Panwalkar tarafından ispatlanmıştır (Smith vd., 1975).

Çizelge 4.4'de gösterilen problem [4 2 1 3] sıralaması ile optimal sonucu vermektedir. Her iş için son makine maksimum işlem zamanlarına sahip olduğundan, adım 1'de elde edilen [4 2 1 3] sıralaması 155 birim toplam akış zamanı ile optimal sıralamayı vermektedir (Smith vd., 1975).

Çizelge 4.4 Sıralı problem işlem zamanları

İşler	Makine				Parça Sayıları
	A	B	C	D	
1	23	16	20	25	125
2	18	12	15	21	100
3	25	22	24	30	150
4	13	8	12	18	90

5. SIRALI AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ ÜZERİNE GENETİK ALGORİTMA UYGULAMASI

5.1 Uygulamanın Kapsam ve Yapısı

Sıralı akış tipi çizelgeleme problemleri üzerine genetik algoritma uygulaması kapsamında 1975 yılında Smith, Panwalkar ve Dudek tarafından geliştirilen, n-iş, m-makine probleminin bir alt kategorisi olan ‘Sıralı Akış Tipi Çizelgeleme Problemi’ kullanılmıştır. Uygulama problemleri Smith ve diğerleri tarafından tanımlanan aşağıdaki iki sıralı problem özelliğini sağlamaktadır.

- i. Eğer her i için, $P_{it} > P_{kt}$, $k \in N$ ve $t \in M$; ise,
Her $j \in M$ için, $P_{ij} \geq P_{kj}$ dir.
- ii. Eğer her $i \in N$ ve $r, j \in M$ için, $P_{ir} > P_{ij}$ ise,
Her $k \in N$ için, $P_{kr} \geq P_{kj}$ dir.

P_{ij} i . işin j . makinedeki işlem zamanını, N n işin kümesini, M ise m makinenin kümesini göstermektedir.

Uygulama kapsamında sıralı akış tipi çizelgeleme problemleri iki bölüm şeklinde ele alınmıştır. Smith ve diğerlerinin literatürde sıralı akış tipi çizelgeleme problemleri için tanımladıkları iki ayrı problem çeşidi de ayrı ayrı ele alınmıştır. Ele alınan ilk problem türü genel durum olarak tanımlanan, işler içerisinde maksimum işlem zamanının ara makinelerde olduğu durumdur. Bu problem çeşidinde probleme girdi teşkil eden işlem zamanlarını içeren sıra matrisinde, işler içerisinde en büyük işlem zamanlarının ilk ve son makinede değil ara makinelerden birinde oluşmaktadır. Smith ve diğerleri tarafından genel durum problemine dal-sınır algoritmasının benzeri sezgisel bir çözüm yöntemi önerilmiştir.

Uygulama kapsamında ele alınan ikinci problem çeşidi ise genel durumun bir özel çıkarımı olan, işler içerisinde en büyük işlem zamanlarının ilk yada son makinede oluşması durumudur. Bu problem çeşidi için Smith ve diğerleri uygulama açısından görece daha kolay olan işleri artan yada azalan biçimde sıralamaya dayalı bir çözüm yöntemi önermişlerdir.

Ayrıca uygulamada kullanılan sıralı akış tipi çizelgeleme problemleri, akış tipi çizelgeleme problemleri için genel olarak yapılan kabulleri de içermektedirler (Jain ve Bagchi, 2000);

1. Tezgaah hazırlık süreleri bilinmektedir ve işlem sürelerinin içerisinde yer almaktadır,

2. n adet iş, sıfırncı zamanda işlenmeye hazır olarak beklemektedir,
3. Her iş m operasyona ve her operasyon da farklı makinelere ihtiyaç duymaktadır,
4. Makinelerin önceden rezervasyon yapılmasına izin verilmez. Her işin ilk operasyonu önce başlar, ikinci operasyonun başlayabilmesi için, birincinin tamamlanması gerekir,
5. Çizelgeleme periyodu boyunca makineler sabittir, mevcut yerlerinde bulunurlar ve kapasiteleri aynıdır,
6. Bir iş aynı anda birden fazla makinede işlenemez,
7. Makine bir anda bir tek işi işleyebilir, birden fazla işi işleyemez,
8. Bir işe ait operasyon bölünemez, bir makinede tamamlanması gerekir.

Uygulamada çözüm yöntemi olarak genetik algoritma kullanılmıştır. Kullanılan genetik algoritma Goldberg tarafından geliştirilen Basit Genetik Algoritmanın(Simple Genetic Algorithm) sıralı akış tipi çizelgeleme problemleri için modifiye edilmiş durumudur.

Genetik algoritmanın iş çizelgelemede kullanılmasında iki temel yaklaşım benimsenmektedir (Cheng vd., 1999).

1. Genetik algoritma kullanılarak uygun permutasyonların geliştirilmesi,
2. Geliştirilen permütasyonlara uygun çözüm elde etmek için sezgisel metotların oluşturulması.

Farklı parametre seçiminin, akış tipi çizelgeleme probleminin genetik algoritma ile çözüm kalitesini etkileyeceği, genetik algoritmanın diğer uygulamalarında belirtilmiştir (Reeves, 1995a). Genetik algoritmanın birçok kontrol parametresi mevcuttur. Başlangıç popülasyonu, üreme yöntemi, çaprazlama ve mutasyon yöntemi ve çaprazlama ve mutasyon oranlarının belirlenmesi bu parametrelerin en önemlileridir. Genetik algoritmanın çalışma yöntemi şekil 5.1'de gösterilmiştir.

Uygulamada performans kriteri olarak, en son işin son makinedeki tamamlanma zamanı(makespan, C_{max}) kullanılmıştır (Chen vd., 1995). Uygulamada kullanılan problemler n-iş, m-makine olarak n/m/P/ C_{max} notasyonu ile gösterilir ve tamamlanma zamanı (5.1) ifadesindeki gibi hesaplanır (Reeves, 1995b). Buna göre, i. işin j. makinedeki işlem zamanı $P(i,j)$ ve iş sırası (J_1, J_2, \dots, J_n) ise,

$$C(J_1,1) = P(J_1,1)$$

$$C(J_i,1) = C(J_{i-1},1) + P(J_i,1) \quad i = 2, \dots, n$$

$$C(J_{1,j}) = C(J_{1,j-1}) + P(J_{1,j})$$

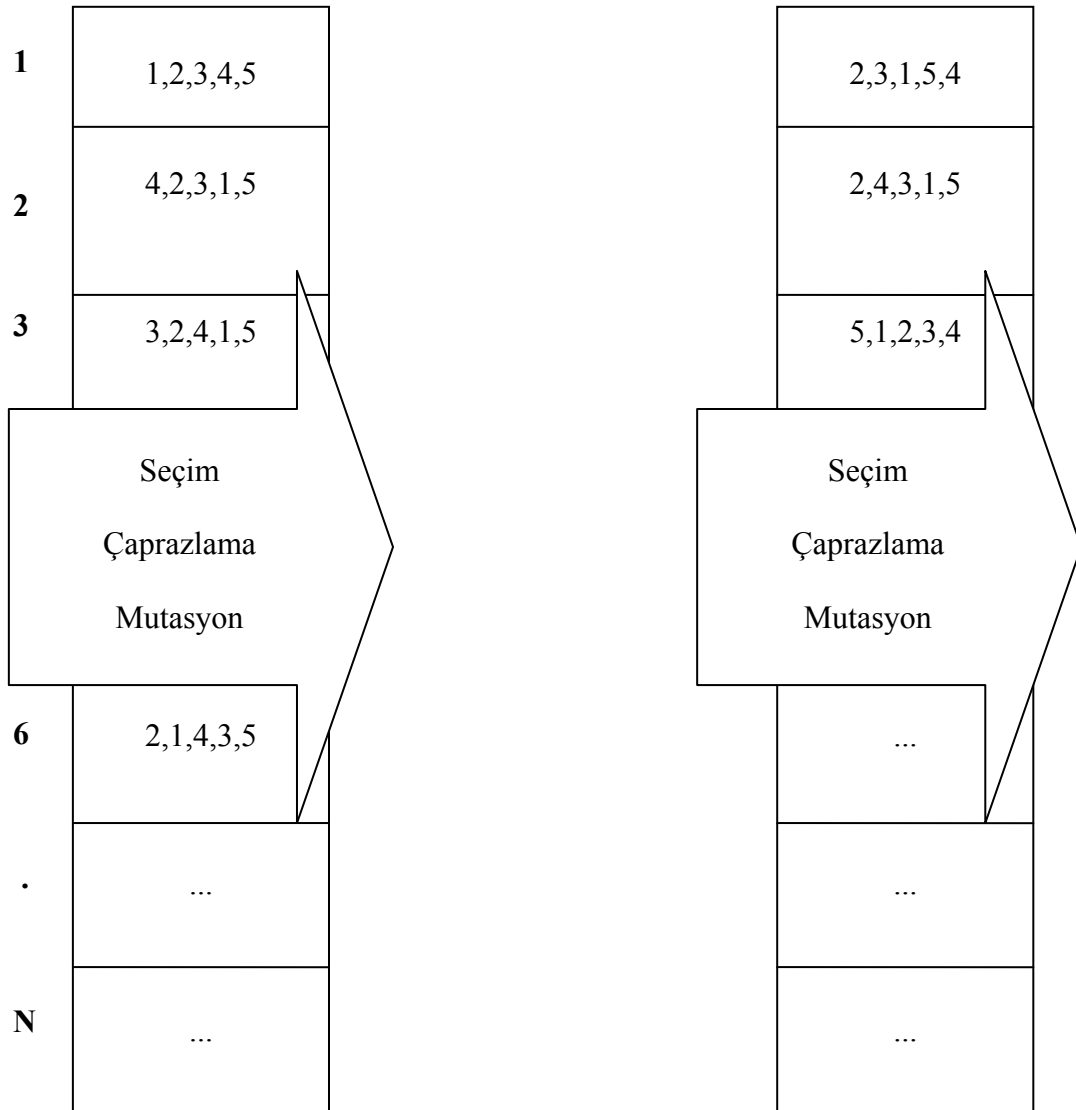
$$C(J_{i,j}) = \max (C(J_{i-1,j}), C(J_{i,j-1})) + P(J_{i,j})$$

$$C_{\max} = C(J_{n,m})$$

$$j = 2, \dots, m$$

$$i = 2, \dots, n ; j = 2, \dots, m$$

(5.1)



Şekil 5.1. Temel genetik algoritma

5.2 Uygulama İçin Kullanılan Model

Genetik algoritma ile model kurulmak istenildiğinde ele alınması gereken konu başlıkları,

- Yapının gösterimi(yada modelin kodlanması)
- Başlangıç popülasyonu ve popülasyon büyüklüğü
- Popülasyonlardaki bireylerin uygunluk değerleri
- Bireylerin seçimi
- Genetik operatörler
- Tamamlanma kriteri dir.

Uygulamada, genetik algoritmanın performansını değerlendirmek için karşılaştırma aracı olarak tavlama benzetimi seçilmiştir. Kullanılan tavlama benzetimi modeli, performansı olabildiğince gerçekçi yansıtması için, mümkün olduğu kadar parametrelerin etkisinden arındırılmaya çalışılmıştır. Uygulamada 1989 yılında Osman ve Potts tarafından önerilen tavlama benzetimi modeli kullanılmıştır. Bu modele göre rassal bir başlangıç çözümü, x , yaratılmakta ve bu çözümün komşu bir y çözümünde dönüşümü olasılığı olarak da aşağıdaki ifade kullanılmaktadır:

$$P(x \rightarrow y) = \min \left\{ 1, \exp\left(-\frac{f(y) - f(x)}{c_k}\right) \right\} \quad k = 1, 2, \dots, N \quad (5.2)$$

Burada, c_k sıcaklık olarak adlandırılan kontrol parametresi ve N ise toplam iterasyon sayısıdır. Tavlama benzetiminde kontrol parametresi büyük bir başlangıç değerinden küçük bir değere doğru azalır. Uygulamada c_k değeri için aşağıdaki ifade kullanılmıştır:

$$c_{k+1} = c_k / (1 + \beta c_k) \quad k = 1, 2, \dots, N-1 \quad (5.3)$$

Bu ifadede β pozitif bir tamsayıdır. Kullanılan β ve c_k değerleri Osman ve Potts'un önerdiği şekilde seçilmiştir.

$$c_1 = \sum_{i=1}^n \sum_{j=1}^m p_{ij} / (5mn)$$

$$c_k = 1$$

$$\beta = (c_1 - c_k) / ((k-1)c_1c_k) \quad (5.4)$$

Uygulamada tavlama benzetimi için 1200 iterasyon yapılmıştır. Uygulamada kullanılan tavlama benzetimi modelinin pseudo-code olarak gösterimi aşağıda verilmiştir.

```

x* = x = GenerateInitialSolution()
T = StartTemperature
i = 0
do

    Pick a neighbour y from the neighbourhood N(x) at random
    Δf = f(y) - f(x)
    if ( Δf < 0) x = y
    else

        r = random number in range [0,1]
        if (r < e-Δf/T) x = y

    i = i + 1
    T = AnnealingSchedule(T,i)
    if (f(x) < f(x*)) x* = x

while i <> '1200'

```

5.2.1 Yapının Gösterimi

Burada ilk aşama olan yapının gösterilmesi yada modelin kodlanması, tüm bu başlıklar içerisinde algoritmanın performansını etkileyen faktörlerin başında gelir. Yanlış kodlanmış bir yapı, hem algoritmayı çözümden uzaklaştıracak yada çözüme daha zor ulaşmasını sağlayacak, hem de çözüm için gereken işlem sürelerini arttıracaktır.

Permutasyon türü çizelgeleme problemleri için, temel kod yapısının işlerin dizilişi biçiminde olmasının problem üzerinde basitleştirici bir etkisi bulunmaktadır (Chen vd., 1996).

Bu uygulamada da yapı gösterimi olarak işlerin sırası seçilmiştir. Kromozom dizisi üzerinde bulunan her bir rakam bir işi temsil etmektedir. Örneğin 8 işe sahip bir akış tipi çizelgeleme probleminde, (1, 2, 3, 4, 5, 6, 7, 8) rakamları sıra ile işleri temsil etmektedir (Cleveland ve Smith, 1989).

Kullanılan bu kodlamanın bir diğer adı da, permütasyon tipi temsildir. Bu kodlama, literatürde en yaygın olarak kullanılan yöntemlerden biridir (Poon ve Carter, 1995).

Uygulamada kullanılan çizelgeleme problemlerinin işlem süreleri literatüre paralel biçimde uniform dağılıma göre [1-25] aralığında, ancak sıralı akış tipi çizelgeleme problemleri özelliklerine bağlı kalarak rassal olarak oluşturulmuştur.

5.2.2 Başlangıç Popülasyonu ve Popülasyon Büyüklüğü

Genetik algoritmanın verimini büyük oranda etkileyen bir diğer faktör de uygun bir başlangıç popülasyonunu ve popülasyon büyüklüğünün seçilmesidir. Genel olarak, başlangıç popülasyonu birkaç yol ile oluşturulabilir. Birçok genetik algoritma uygulamasında başlangıç popülasyonunun rassal olarak oluşturulduğu gözlenmektedir.

Başlangıç popülasyonunun farklı çözüm kümelerinden oluşturulması, aramanın etkinleşmesini sağlar. Başlangıç popülasyonunun uygun oluşturulması ortalama uygunluk değerini yükseltir ve hesaplama zamanını yada çözüme ulaşmak için ele alınması gereken nesil sayısını düşürür. Bu nedenle farklı prosedürlerle başlangıç popülasyonu oluşturmak, popülasyon ortalama uygunluk değerini yükseltecek ve çözüme ulaşma süresini kısaltacaktır. Bu çalışmada sonuçlardaki uniform yapının korunması için başlangıç popülasyonundaki bireylerin tamamı rassal olarak oluşturulmuştur (Chen vd., 1996).

Popülasyon büyüklüğü konusunda literatür incelendiğinde [10-90] aralığında kromozom üretildiği görülmektedir (Ghedjati, 1999). Başlangıç popülasyonu sayısı optimizasyonu konusunda çalışma yapan Gen ve Cheng, problemin çözüm süresi kriterine göre en iyi değerleri 40 başlangıç popülasyonu sayısı ile elde etmişlerdir (Gen ve Cheng, 2000). Ancak gerek literatürdeki diğer örnekler, gerekse de uygulamada seçilen n-iş ve m-makine sayıları incelendiğinde bu çalışma için başlangıç popülasyonu olarak 60 kromozom seçilmiştir.

5.2.3 Popülasyondaki Bireylerin Uygunluk Değerleri

Maksimizasyon problemlerinde performans ölçümü genellikle uygunluk fonksiyonu üzerinden yapılmaktadır. Ancak çizelgeleme problemlerinde amaç genellikle belirli bir performans kriterini minimize etmektir. Minimizasyon problemlerinde uygunluk fonksiyonunun hesaplanması, maksimizasyon problemlerine göre farklılık göstermektedir. Bu çalışmada uygunluk değerinin belirlenmesinde akış tipi çizelgeleme problemlerine uyumlu bir yapı kullanılmıştır.

Öncelikle popülasyondaki tüm bireyler için son işin son makinedeki tamamlanma zamanı(makespan) hesaplanmıştır. Burada en büyük tamamlanma zamanı CMAX ile gösterilmiştir. Popülasyondaki her bireyin tamamlanma zamanlarının CMAX ile olan farkı o bireyin uygunluk değerini vermektedir. Böylece daha düşük tamamlanma zamanına sahip bireylerin seçilme olasılığı yüksek olacaktır. Bu olasılık değeri aynı zamanda bireylerin yeni bireyler meydana getirmek için seçilmesinde de kullanılmaktadır.

5.2.4 Bireylerin Seçimi

Bireylerin seçiminde literatüre uygun şekilde, toplam akış zamanına göre rulet çemberi yöntemi kullanılmıştır (Cheng vd., 1996). Bireylerin uygunluk değerleri kullanılarak, her birey için uygunluk fonksiyonu hesaplanmış, bu fonksiyondan yararlanılarak da her birey için uygunluk değerine dayalı seçilme olasılığı belirlenmiştir.

Uygunluk fonksiyonunun notasyon olarak gösterimi (5.5) ifadesinde verilmiştir.

$S(t)$: t. jenerasyondaki popülasyon,

$s_i(t)$: $S(t)$ 'nin i. bireyi,

$C(s_i(t))$: $s_i(t)$ 'nin tamamlanma zamanı,

$CMAX(t)$: $\max\{C(s_i(t))\}$, her $s_i(t) \in S(t)$,

$f(s_i(t))$: $s_i(t)$ 'nin uygunluk değeri, $CMAX(t) - C(s_i(t))$,

$TOTFIT(t)$: $f(s_i(t))$ 'lerin toplamı, her $s_i(t) \in S(t)$

$$P(s_i(t)) = \frac{f(s_i(t))}{TOTFIT} \quad (5.5)$$

5.2.5 Genetik Operatörler

Genetik algoritmada literatür incelendiğinde üç temel genetik operatörden söz edilmektedir: üreme(seçim), çaprazlama ve mutasyon. Üreme operatörü, popülasyon oluşumu için kullanılan operatördür. Bu operatör bazı kaynaklar tarafından bir genetik operatör olarak kabul edilirken bazı kaynaklar tarafından da genetik operatör olarak kabul edilmemektedir. Bu çalışmanın uygulama aşamasında da üreme bir genetik operatör olarak kabul edilmemiş ve çalışmada çaprazlama ve mutasyon olmak üzere yalnızca iki operatör ele alınmıştır. Çaprazlama iki ebeveynden yeni bir bireyin oluşturulması işlemidir. Genetik algoritmada çeşitli çaprazlama yöntemleri kullanılmaktadır. Literatür incelendiğinde akış tipi çizelgeleme problemleri için genellikle bir yada iki noktalı çaprazlama yöntemi kullanıldığı görülür. Çaprazlama oranı konusunda da literatürde, “1.00, 0.95 ve 0.90” değerlerinin kullanıldığı görülmektedir. Uygulamada kullanılan Goldberg’ın basit genetik algoritmasında ise çaprazlama yapılıp yapılmayacağı bir paranın havaya atışı ile belirlenir. Paranın ‘yazı’ yüzü gelirse çaprazlama yapılır, ‘tura’ yüzü gelirse çaprazlama yapılmaz birey sonraki nesile değişmeden geçirilir olarak kabul edilir. Bu uygulamada literatürdeki değerlere de bağlı kalarak algoritmanın yapısını bozmamak açısından oran “1.00” olarak kabul edilmiş yani paranın hep ‘yazı’ geldiği varsayılmıştır.

Akış tipi çizelgeleme problemlerinde en iyi performans gösteren çaprazlama operatörünün sıraya dayalı çaprazlama olduğu belirlenmiştir. Bu uygulamada Goldberg’ın algoritmasına sadık kalınarak tek noktalı çaprazlama kullanılmıştır.

Mutasyon, tek birey üzerinde normal olmayan kalıtsal değişikliklerdir. Literatür incelendiğinde akış tipi problemler için, “0.01, 0.005 ve 0.00” mutasyon oranları için çalışmalar yapılmıştır. Kullanılan oran çalışmanın performansını büyük oranda etkilemektedir. Bu çalışmada Goldberg’ın algoritma yapısı göz önüne alınarak ve literatürdeki diğer çalışmaların performans değerleri incelenerek mutasyon oranının “0” olmasına karar verilmiştir. Uygulama kapsamında genetik operatör olarak yalnızca çaprazlama operatörü kullanılmıştır.

5.2.6 Tamamlanma Kriteri

Genetik algoritmalarda algoritmayı en iyi sonucunu bulana dek çalıştırmak yerine önceden belirlenmiş, belli bir düzeyin üzerinde iyi sonuç elde edinceye kadar çalıştırmak pek çok

arařtırmacı tarafından tercih edilen bir durumdur. Bu alıřmada olduđu gibi algoritmanın performansını deđerlendirmeye dayalı alıřmalarda iki temel tamamlanma kriteri kullanılmaktadır.

- Nesil Sayısı yada alıřma zamanı
- Önceden belirlenen sonu

Nesil sayısı kriterinde, algoritma önceden belirlenen nesil sayısı kadar yeni birey üretecek řekilde alıřtırılıp bu bireylerdeki iyileřme incelenir.

alıřma zamanı kriterinde ise algoritma önceden belirlenen CPU zamanı kadar alıřtırılıp benzer řekilde sonutaki iyileřme incelenerek performans deđerlendirmesi yapılır.

Önceden belirlenen sonu kriterinde ise algoritma alıřtırılmadan önce tamamlanma için bir eřik deđer atanır. Algoritmanın performansı bu eřik deđere kaç nesilde yada ne kadar alıřma zamanında ulařabildiđi göz önüne alınarak deđerlendirilir.

Bu alıřma kapsamında, literatürdeki benzer alıřmalar incelenmiř ve tamamlanma kriteri olarak 20 jenerasyon oluřumu belirlenmiřtir.

Uygulamada kullanılan genetik algoritma modelinin pseudo-code olarak gösterimi ařađıda verilmiřtir.

```
M:=60;
```

```
G:=20;
```

```
pop_no:=0;
```

```
gen_no:=0;
```

```
repeat
```

```
    pop_no:=pop_no+1;
```

```
    generate(random_sequence);
```

```
    evaluate(random_sequence);
```

```
until pop_no:=M;
```

```
repeat
```

```
    gen_no:=gen_no+1;
```

```
    select(parent_1) using fitness_func;
```

```
    select(parent_2) using fitness_func;
```

```

choose(crossover_point);
crossover;
evaluate(new_sequence);
select(old_sequence) from unfit_members;
delete(old_sequence) from population;
insert(new sequence) into population;
until gen_no:=G;
calculate(population_statistics);

```

5.3 Genel Durum Problemi Üzerine Uygulama

Sıralı akış tipi çizelgeleme problemi üzerine genetik algoritma uygulaması kapsamında öncelikle ele alınan, sıralı akış tipi çizelgeleme problemlerinde genel durum olarak adlandırılan, işler için en büyük işlem zamanlarının ara makinelerden birinde gerçekleşmesi durumudur.

Bu uygulama kapsamında, literatürdeki örnekler gözönünde bulundurularak, n-iş, m-makine şeklinde, 7x4, 7x5, 7x8, 7x10, 7x15, 10x4, 10x5, 10x8, 10x10, 10x15, 10 farklı iş-makine kombinasyonu ele alınmıştır.

Tüm problem çeşitleri için işlem zamanları sıralı akış tipi çizelgeleme problemleri kurallarına uygun şekilde [1-25] aralığında üniform olarak oluşturulmuştur. Her problem, öncelikle Smith, Panwalkar, Dudek algoritması ile çözümlenerek optimum sıralama ve tamamlanma zamanı bulunmuştur. Daha sonra genetik algoritma ile daha önce belirlenen kriterler ve 20 jenerasyon esasına göre çözüm yapılmıştır. Son olarak aynı problem tavlama benzetimi ile 1200 iterasyon olarak çözülmüştür.

Genel durum problemi üzerine yapılan uygulama için kullanılan problemlerin işlem zamanları EK-A'da verilmiştir.

İş makine kombinasyonlarının çözüm sonuçları çizelge 5.1'de görülmektedir. Elde edilen sonuçlara göre, genetik algoritma 10-iş, 5-makine probleminde, Smith, Panwalkar, Dudek algoritması ile aynı [61374528109] sıralamasını bulmuş, benzer biçimde 10-iş, 10-makine problemi için de Smith Panwalkar Dudek algoritması ile aynı tamamlanma zamanına sahip farklı bir sıralama olan [78361051924] sıralamasını elde etmiştir.

Çizelge 5.1 Genel durum problemi uygulama sonuçları

	SPD		GA		SA	
	Sıralama	TZ	Sıralama	TZ	Sıralama	TZ
7X4	5761324	145	5163724	149	2716345	153
7x5	2731465	140	4713265	141	5627314	144
7x8	7314265	223	7314625	225	3264157	234
7x10	6257134	225	4257136	226	2317546	231
7x15	7613452	332	5613472	334	7516342	343
10x4	10972564318	223	42613579810	225	10249761385	231
10x5	61374528109	231	61374528109	231	93471061582	247
10x8	95148107326	279	61021347859	281	94106175328	297
10x10	78361051942	306	78361051924	306	78491536210	337
10x15	43875911062	361	43810965127	366	61421075398	381

Çözüm sonuçlarının ikili performans değerlendirmesi çizelge 5.2’de görülmektedir. Çizelgede C_{SPD} Smith Panwalkar Dudek algoritması için son işin son makinedeki tamamlanma zamanı, C_{GA} genetik algoritma için son işin son makinedeki tamamlanma zamanı, C_{SA} ise tavlama benzetimi için son işin son makinedeki tamamlanma zamanıdır.

Çizelge 5.2 Genel durum problemi performans karşılaştırması

	C_{SPD}/C_{GA}	C_{SPD}/C_{SA}	C_{GA}/C_{SA}
7X4	97.32	94.77	97.39
7x5	99.29	97.22	97.92
7x8	99.11	95.30	96.15
7x10	99.56	97.40	97.84
7x15	99.40	96.79	97.38
10x4	99.11	96.54	97.40
10x5	100.00	93.52	93.52
10x8	99.29	93.94	94.61
10x10	100.00	90.80	90.80
10x15	98.63	94.75	96.06

Uygulama sonucunda genetik algoritma Smith Panwalkar Dudek algoritmasına göre ortalama % 99.17 performans göstermiştir. Benzer biçimde tavlama benzetiminin ortalama performansı ise % 95.10 olmuştur.

5.4 Özel Durum Problemi Üzerine Uygulama

Sıralı akış tipi çizelgeleme problemleri uygulaması kapsamında ikinci olarak ele alınan, özel durum problemidir. Bu durumda sıralı işlem zamanları matrisinde en büyük işlem zamanları ilk yada son makinede oluşmaktadır.

Bu uygulamada da genel durum probleminde olduğu gibi, 7x4, 7x5, 7x8, 7x10, 7x15, 10x4, 10x5, 10x8, 10x10, 10x15 şeklinde 10 farklı iş-makine kombinasyonu ele alınmıştır.

Tüm problemler için işlem zamanları genel durum probleminde olduğu gibi [1-25] aralığında üniform dağılıma göre, sıralı akış tipi çizelgeleme problemlerinin özellikleri dikkate alınarak oluşturulmuştur. İşlem zamanlarının oluşturulması sırasında üzerinde dikkatle durulan nokta, problemin özel durum problemi olarak tanımlanmasını da sağlayan kriter olan en büyük işlem zamanlarının ilk yada son makinede oluşmasına dikkat edilmesi olmuştur.

Özel durum problemi uygulaması için kullanılan işlem zamanları EK-B’de verilmiştir.

Özel durum uygulaması için iş makine kombinasyonlarının çözümleri çizelge 5.3’de görülmektedir. Uygulamada, genetik algoritma 7-iş, 4-makine probleminde Smith Panwalkar Dudek algoritması ile aynı sonucu veren bir sıralamaya ulaşılmıştır.

Çizelge 5.3 Özel durum problemi uygulama sonuçları

	SPD		GA		SA	
	Sıralama	TZ	Sıralama	TZ	Sıralama	TZ
7X4	6172435	142	6241357	142	5241736	146
7x5	3752146	225	3746152	237	4136275	236
7x8	2641735	236	2714365	243	7415326	277
7x10	7543126	255	7614532	258	7561324	264
7x15	7463215	433	7631246	434	5326147	483
10x4	43981102765	220	97361458210	230	72461510938	238
10x5	24106891537	220	26498105137	222	67145382109	234
10x8	57936241108	301	79563412108	303	61021347859	321
10x10	97451610328	356	74961531082	358	78369511042	377
10x15	58732964101	385	87526394101	388	73854629110	396

Çözüm sonuçlarının ikili performans değerlendirmeleri çizelge 5.4’de görülmektedir.

Çizelge 5.4 Özel durum problemi performans karşılaştırması

	CSPD/CGA	CSPD/CSA	CGA/CSA
7X4	100.00	97.26	97.26
7x5	94.94	95.34	100.42
7x8	97.12	85.20	87.73
7x10	98.84	96.59	97.73
7x15	99.77	89.65	89.86
10x4	95.65	92.44	96.64
10x5	99.91	94.87	94.02
10x8	99.34	93.77	94.39
10x10	99.44	94.43	94.96
10x15	99.23	97.22	97.98

Özel durum problemi üzerine yapılan 10 farklı uygulamada, genetik algoritma Smith Panwalkar Dudek algoritmasına kıyasla, ortalama %98.52 performans göstermiş, tavlama benzetiminin performansı ise ortalama %95.09 olmuştur.

5.5 Sonuçlar ve Değerlendirme

Sıralı akış tipi çizelgeleme problemlerinde genetik algoritma uygulaması kapsamında, 10 farklı iş-makine kombinasyonu için, genel durum problemi üzerine 10 ve özel durum problemi üzerine 10 adet olmak üzere 20 adet uygulama yapılmıştır. Uygulamada kriter olarak son işin son makinedeki tamamlanma zamanı ele alınmış ve karşılaştırma için tavlama benzetimi kullanılmıştır.

Genel durum problemi üzerine yapılan uygulamanın sonuçları incelendiğinde genetik algoritmanın, tavlama benzetiminden daha iyi sonuç verdiği görülmektedir. Ele alınan 10 farklı uygulama probleminin hiçbirinde, tavlama benzetimi genetik algoritmadan daha iyi sonuç elde edememiştir. Buna karşın genetik algoritma SPD algoritmasına yakın sonuçlar elde etmiş ve 10-iş, 5-makine kombinasyonunda SPD algoritması ile aynı tamamlanma zamanına ulaşmıştır.

Genel durum problemi uygulaması kapsamında, genetik algoritma SPD algoritmasına kıyasla % 99.17 performans göstermiştir. Tavlama benzetimi ile kıyaslandığında, performansı ortalama %104.32, olmuş başka bir deyişle, tavlama benzetimi uygulamada genetik algoritmanın performansına göre ortalama % 95.91 seviyesinde kalmıştır.

Uygulama sonuçlarında bir diğer dikkat çeken nokta ise, iş-makine kombinasyonlarında, iş ve makine sayıları arttıkça genetik algoritmanın performansın yükseldiği görülmektedir. Buna karşılık tavlama benzetiminin performansında iş-makine sayılarındaki artışa bağlı azalma gözlenmiştir.

Sıralı akış tipi çizelgeleme problemleri için Smith Panwalkar ve Dudek tarafından önerilen çözüm yöntemi optimum sonucu buluyor dahi olsa çözüm için, 2^{n-1} alternatifi denediğinden, belki iş sayısı küçük olduğu durumlarda dikkate değer olabilir. Ancak iş sayısının büyük olduğu durumlarda değerlendirilecek alternatif sayısı çok arttığından ve bu ciddi zaman kaybına sebep olacağından, optimum sonucu garanti etmeyen ancak yapılan uygulamada optimuma göre ortalama % 99.17 performans gösteren genetik algoritmayı kullanmak optimuma yakın bir çözümü çok daha kısa zaman ve eforla elde etmek açısından dikkate değer olacaktır.

Özel durum problemi üzerine yapılan uygulamada da, genel durum probleminde ele alınan iş-makine kombinasyonları ele alınmıştır. Yapılan 10 farklı uygulama probleminde tavlama benzetimi yalnızca bir defa 7x5 kombinasyonunda 236 birim tamamlanma zamanı ile genetik algoritmadan daha iyi sonuç elde etmiştir. Diğer 9 uygulamanın tümünde genetik algoritmanın performansı tavlama benzetimine kıyasla daha iyidir.

Uygulamada genetik algoritma, SPD algoritmasına kıyasla ortalama % 98.52 performans göstermiştir. Tavlama benzetimi ise SPD algoritmasına kıyasla ortalama % 93.67 performans göstermiştir. Genetik algoritma ile tavlama benzetimi kıyaslandığında tavlama benzetiminin genetik algoritmaya göre ortalama % 95.09 performans değerinde kaldığı gözükmemektedir.

Özel durum uygulamasında da genel durum uygulamasına benzer biçimde, iş-makine sayısı arttıkça genetik algoritmanın performansında artış gözlemlenmiştir. Bu süreçte tavlama benzetimi dalgalı bir performans izlemektedir.

Özel durum problemi için Smith Panwalkar ve Dudek tarafından önerilen çözüm yöntemi yapı olarak basit, kolay uygulanır, ve iyi sonucu garanti eden bir çözüm yöntemidir. Bu açıdan değerlendirildiğinde, özel durum problemlerinde her ne kadar genetik algoritma ortalama % 98.52 performans değerine ulaşmış ise de, daha iyi sonucu garanti eden ve çok kolay uygulanan Smith Panwalkar Dudek algoritmasını kullanmak daha avantajlı olacaktır.

Sonuçta, uygulamanın her iki kısmında bir bütün olarak ele alındığında genetik algoritma SPD algoritmasına kıyasla 20 uygulama probleminde ortalama % 98.84 performans göstermiştir. Burada tavlama benzetiminin performansı ise ortalama % 94.38 performans değerinde kalmıştır. Genetik algoritma özellikle genel durum problemi için, SPD algoritmasından çok daha kısa zamanda optimuma çok yakın (belki optimum) çözüm geliştirebildiğinden, bu problem çeşidi için kullanım açısından belirgin bir avantaj göstermiştir.

6. SONUÇLAR ve ÖNERİLER

Akış tipi çizelgeleme problemlerinde, 2 makine ve n iş probleminin Johnson algoritması ile optimum çözümü mümkündür. Makine sayısı bir arttırıldığında, Johnson algoritmasının özel şartları ile optimum sonuca ulaşılabilir. Ancak makine sayısının 3 den fazla olduğu durumlarda optimum çözüme ulaşmayı garanti eden bir algoritma henüz geliştirilememiştir. Bu nedenle akış tipi çizelgeleme problemlerinin çizelgenmesinde, sezgisel yöntemler kullanılmaktadır.

Sıralı akış tipi çizelgeleme problemleri, 1975 yılında Smith Panwalkar ve Dudek tarafından tanımlanmıştır. Akış tipi çizelgeleme problemlerinin alt kategorisi olan bu problem, problemin “sıralı” olarak tanımlanmasını sağlayan iki temel karakteristik içerir.

NP problemlerin çözümü için, literatürde üç grup yöntem kullanılmaktadır (Goldberg, 1989).

Bunlar:

1. Hesaplama tabanlı (matematiksel) yöntemler,
2. Sayısal yöntemler,
3. Rassal arama metotlarıdır.

Herhangi bir akış tipi çizelgeleme problemi için, hesaplama tabanlı çözüm metotları henüz mevcut değildir. Sayısal yöntemler ise, her problemde $n!$ farklı sıralama alternatifi olduğundan kullanılması zor ve zaman alıcıdır. Bu sebeplerle akış tipi çizelgeleme problemleri için en uygun metot rassal arama yöntemidir.

Genetik algoritma, rassal arama teknikleri kullanarak çözüm bulmaya çalışan, parametre kodlamaya dayalı bir arama yöntemidir. Literatürde genetik algoritmaların, akış tipi çizelgeleme problemlerinde başarı ile uygulandığını gösteren pekçok örnek bulunmaktadır.

Genetik algoritmanın, sıralı akış tipi çizelgeleme problemleri üzerindeki performansını ölçmek için bir dizi uygulama yapılmıştır. Uygulamada öncelikle, problem Smith Panwalkar ve Dudek tarafından tanımlandığı şekilde, genel durum problemi ve özel durum problemi olarak ikiye ayrılmış ve uygulama her problem tipi için ayrı gerçekleştirilmiştir.

Genel durum problemi üzerine yapılan uygulamada genetik algoritma performans olarak, diğer bir rassal arama tekniği olan tavlama benzetimi ile karşılaştırılmıştır. Problemin

optimum çözümünü veren SPD algoritması ile kıyaslandığında, genetik algoritma oldukça tatmin edici olarak değerlendirilebilecek ortalama % 99.17 performans değerine ulaşmıştır. Tavlama benzetimi ise, genetik algoritmaya göre % 95.91 değerinde kalmıştır. Genel durum problemi için SPD algoritması incelendiğinde, algoritmanın dal-sınır arama tekniğine benzer, 2^{n-1} alternatifi değerlendirmeyi baz alan bir yöntem olduğu görülmektedir. İş sayısı arttığında bu yöntemin uygulanırlığı oldukça azalmakta ve çözüme ulaşmak çok fazla zaman almaktadır. Bu bakımdan değerlendirildiğinde, optimum sonucu garanti etmese bile, genel durum problemlerinin çözümünde genetik algoritmanın tercih edilmesi avantaj sağlayacaktır.

Özel durum problemi üzerine yapılan uygulamada ise, genetik algoritma SPD algoritmasına kıyasla ortalama % 98.52 performans göstermiştir. Tavlama benzetiminin genetik algoritmaya göre performansı ise % 95.09 olmuştur. Bu problem çeşidi için SPD tarafından önerilen çözüm yöntemi basit ve etkilidir. Bu nedenle, özel durum olarak adlandırılan problem çeşidi ile karşılaşıldığında, Smith Panwalkar ve Dudek tarafından önerilen orjinal çözümü kullanmak daha etkili ve sonuç açısından garantili olacaktır.

Genetik algoritma, parametre kodlama esaslı olduğundan ileriki çalışmalarda performansını artırıcı araştırmalar yapılabilir. Özellikle başlangıç popülasyonunu rassal olarak seçmek yerine, daha elit bireylerin seçilimini sağlayacak özel bir yöntem geliştirmek, başlangıç popülasyonu sayısını optimize etmek, kullanılan çaprazlama oranı ve operatörünü optimize etmek, genetik algoritmanın performansını olumlu yönde etkileyecektir.

KAYNAKLAR

Arora, R. K., Rana, S. P., (1980), "Scheduling in a Semi-Ordered Flow Shop Without Intermediate Queues", *AIIE Transactions*, 12 (3): 264-272.

Blazewich, J., Ecker, K. H., Pesch, E., Schmidt, G., Weglarz, J., (1996), *Scheduling Computer and Manufacturing Processes*, Springer Verlag, Germany.

Chan, L. M. A., Mariuel, A., Levi, D. S., (1998), "Parallel Machine Scheduling, Linear Programming and Parameter List Scheduling Heuristics", *Operations Research*, 46 (5).

Chen, C. L., Neppalli, R. V., Aljaber, N., (1996), "Genetic Algorithms Applied to the Continuous Flow Shop Problem", *Computers and Industrial Engineering*, 30 (4): 919-929.

Chen, C. L., Vempati, V. S., Aljaber, N., (1995), "An Application of Genetic Algorithms for Flowshop Problems", *European Journal of Operational Research*, 80: 389-396.

Cheng, R., Gen, M., Tsujimura, Y., (1999a), "A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms: Part I, Hybrid Genetic Search Strategies", *Computers and Industrial Engineering*, 36: 343-364.

Cheng, R., Gen, M., Tsujimura, Y., (1999b), "A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms: Part II, Hybrid Genetic Search Strategies", *Computers and Industrial Engineering*, 37: 51-55.

Cheng, R., Gen, M., Tsujimura, Y., (1996), "A Tutorial Survey of Job Shop Scheduling Problems Using Genetic Algorithms-I, Representation", *Computers and Industrial Engineering*, 30 (4): 983-997.

Chou, F. D., Lee, C. E., (1999), "Two Machine Flowhop Scheduling with Bicriteria Problem", *Computers and Industrial Engineering*, 36: 549-564.

Croce, F. D., Tadei, R., Volta, G., (1995), "A Genetic Algorithm for the Job Shop Scheduling Problem", *Computers and Operations Research*, 22 (1).

Dasgupta, D., Forrest, S., (1997), "Artificial Immune Systems in Industrial Applications", Department of Computer Science University of Mexico.

Demirsoy, A., (1988), *Kalıtım ve Evrim*, Hacettepe Üniversitesi Fen Fakültesi Biyoloji Bölümü, Meteksan, Ankara.

Forrest, S., Hofmeyr, S. A., (2000), *John Holland Invisible Hand: An Artificial Immune System*, Department of Computer Science, University of Mexico.

Forrest, S., Javornik, B., Smith, R., Perelson, A. S., (1993), *Using Genetic Algorithm to Explore Pattern Recognition in the Immune System*, Department of Computer Science, University of Mexico.

French, S., (1981), *Sequencing and Scheduling: An Introduction to the Mathematics of Job Shop*, Ellis Horwood Press, England.

Gen, M., (1996), "Genetic Algorithms and Industrial Engineering", *Computers and Industrial Engineering*, 30 (4): 835-837.

Gen, M., Cheng, R., (2000), *Genetic Algorithms and Engineering Optimization*, John Wiley Sons Inc., USA.

Ghedjati, F., (1999), "Genetic Algorithms for Job Shop Scheduling Problem with Unrelated Parallel Constraints: Heuristic Mixing Method Machines and Precedence", *Computers and Industrial Engineering*, 37:39-42.

Gilkinson, J. C., Rabelo, L. C., Bush, B. O., (1995), "A Real-World Scheduling Problem Using Genetic Algorithms", *Computer and Language*, 29 (1): 177-181.

Goldberg, D. E., (1989), *Genetic Algorithms in Search Optimization and Machine Learning*, Addition Wesley Publishing Company, USA.

Kuru, M., (1987), *Moleküler Biyoloji*, Atatürk Üniversitesi Eğitim Fakültesi, Biyoloji Eğitimi Ders Notları, Erzurum.

Leestma, S., Nyhoff, L., (1993), *Turbo Pascal Programming and Problem Solving*, Prentice-Hall Inc., USA.

Maturana, F., Gu, P., Naumann, A., Norrie, D. H., (1997), "Object Oriented Job Shop Scheduling Using Genetic Algorithm", *Computers in Industry*, 32.

Murata, T., Ishibuchi, H., Tanaka, H., (1996a), "Genetic Algorithms for Flow Shop Scheduling Problems", *Computers and Industrial Engineering*, 30 (4): 1061-1071.

Murata, T., Ishibuchi, H., Tanaka, H., (1996b), "Multi-Objective Genetic Algorithms and Its Applications to Flow Shop Scheduling", *Computers and industrial Engineering*, 30 (4): 957-968.

Ogbu, F. A., Smith, D. K., (1989), "The Application of the Simalted Annealing Algorithm to the Solution of the $n/m/C_{\max}$ Flowshop Problem", *Computers Operations Research*, 17 (3): 243-253.

Ogbu, F. A., Smith, D. K., (1991), "Simulated Annealing for the Permutaion Flowhop Scheduling Problem", *Omega*, 19 (1): 64-67.

Oraler, G., (1990), *Genetik*, İstanbul Üniversitesi Fen Fakültesi, İstanbul.

Osman, I. H., Potts, C. N., (1989), "Simulated Annealing for Permutaion Flowshop Scheduling", *Omega*, 17 (6): 551-557.

Panwalkar, S. S., Khan, W. A., (1976), "An Ordered Flow Shop Sequencing Problem With Mean Completion Time Criterion", *International Journal of Production Research*, 14 (5): 631-635.

Panwalkar, S. S., Khan, W. A., (1977), "A Convex Property of an Ordered Flow Shop Sequencing Problem", *Naval Research Logistics Quarterly*, 24: 159-162.

- Panwalkar, S. S., Woollam, C. R., (1980), "Ordered Flow Shop Problems with No In-Process Waiting: Further Results", *Journal of Operations Research Society*, 31 (11).
- Pinedo, M., (1995), *Scheduling: Theory, Algorithms and Systems*, Prentice-Hall Inc., USA.
- Poon, P. W., Carter, J. N., (1995), "Genetic Algorithm Crossover for Ordering Applications", *Computers Operations Research*, 22 (1).
- Reeves, C. R., (1995a), *Modern Heuristic Techniques for Combinatorial Problems*, Mc Graw-Hill Book Company, UK.
- Reeves, C. R., (1995b), "Genetic Algorithms for Flowshop Sequencing", *Computers and Operations Research*, 22 (1):5-13.
- Smith, M. L., Panwalkar, S. S., Dudek, R. A., (1975), "Flowshop Sequencing Problem with Ordered Processing Time Matrices", *Management Science*, 21 (5): 544-549.
- Smith, M. L., Panwalkar, S. S., Dudek, R. A., (1976), "Flowshop Sequencing Problem with Ordered Processing Time Matrices: A General Case", *Naval Research Logistics Quarterly*, 23: 481-486.
- Wang, D., Gen, M., Cheng, R., (1999), "Scheduling Grouped Jobs on Single Machine With Genetic Algorithms", *Computers and Industrial Engineering*, 36: 309-324.

EKLER

- Ek-A Genel Durum Problemleri İşlem Süreleri
Ek-B Özel Durum Problemleri İşlem Süreleri

Ek-A Genel Durum Problemleri İşlem Süreleri

Uygulamada kullanılan tüm işlem zamanları, literatüre paralel biçimde [1-25] zaman aralığında üniform dağılıma göre tamsayı olarak seçilmiştir. İşlem zamanları için sıra matrislerinin oluşturulması sırasında, sıralı akış tipi çizelgeleme problemleri özellikleri dikkate alınmıştır.

Çizelge A.1 Genel durum 7-iş, 4-makine problemi için işlem zamanları

İş/Makine	1	2	3	4
1	14	25	18	17
2	5	15	13	9
3	11	21	16	15
4	5	11	10	7
5	3	8	7	6
6	16	23	21	18
7	8	18	15	13

7-iş, 4-makine probleminin SPD algoritması ile çözümünde (5761324) sıralaması 145 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (5163724) sıralaması 149 birim tamamlanma zamanı ve SA ile çözümünde ise (2716345) sıralaması 153 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.2 Genel durum 7-iş, 5-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5
1	19	14	21	14	12
2	13	12	18	10	8
3	16	14	16	12	12
4	13	11	18	9	7
5	5	3	9	2	2
6	7	6	11	6	5
7	15	14	17	13	9

7-iş, 5-makine probleminin SPD algoritması ile çözümünde (2731645) sıralaması 140 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (4713265) sıralaması 141 birim tamamlanma zamanı ve SA ile çözümünde ise (5627314) sıralaması 144 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.3 Genel durum 7-iş, 8-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8
1	14	8	21	18	10	13	19	7
2	19	12	24	19	13	15	23	11
3	9	6	12	11	7	7	12	5
4	16	9	19	16	10	14	18	5
5	11	4	17	15	9	11	15	2
6	21	14	22	22	17	19	22	10
7	9	4	15	13	5	7	13	3

7-iş, 8-makine probleminin SPD algoritması ile çözümünde (7314265) sıralaması 223 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde

Ek-A Genel Durum Problemleri İşlem Süreleri

(7314625) sıralaması 225 birim tamamlanma zamanı ve SA ile çözümünde ise (3264157) sıralaması 234 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.4 Genel durum 7-iş, 10-makine problemi için işlem zamanları

İş/ Makine	1	2	3	4	5	6	7	8	9	10
1	19	22	6	17	17	10	23	15	14	12
2	12	13	7	12	12	8	17	11	10	10
3	13	14	5	14	13	5	15	12	12	9
4	10	11	4	10	11	7	13	8	7	7
5	13	17	10	16	16	12	22	13	12	12
6	7	14	6	10	13	7	18	7	7	7
7	17	16	11	16	15	14	18	16	15	15

7-iş, 10-makine probleminin SPD algoritması ile çözümünde (6257134) sıralaması 225 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (4257136) sıralaması 226 birim tamamlanma zamanı ve SA ile çözümünde ise (2317546) sıralaması 231 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.5 Genel durum 7-iş, 15-makine problemi için işlem zamanları

İş/ Makine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	23	21	14	17	9	21	18	19	24	11	12	22	20	16	13
2	10	8	6	6	3	9	7	8	13	4	5	8	8	6	5
3	20	13	9	12	6	16	12	13	25	6	8	17	13	11	8
4	17	14	13	13	8	15	13	13	18	8	11	16	14	13	11
5	16	9	9	9	8	12	9	9	22	9	9	13	9	9	8
6	23	20	13	14	10	21	16	17	23	13	13	22	19	14	13
7	14	12	8	9	5	14	10	10	21	7	7	14	11	9	8

7-iş, 15-makine probleminin SPD algoritması ile çözümünde (7613452) sıralaması 332 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (5613472) sıralaması 334 birim tamamlanma zamanı ve SA ile çözümünde ise (7516342) sıralaması 343 birim tamamlanma zamanı ile elde edilmiştir.

Ek-A Genel Durum Problemleri İşlem Süreleri

Çizelge A.6 Genel durum 10-iş, 4-makine problemi için işlem zamanları

İş/Makine	1	2	3	4
1	11	10	19	16
2	18	12	22	21
3	18	10	23	22
4	15	12	19	16
5	16	13	21	19
6	14	9	17	15
7	17	14	18	18
8	11	6	15	13
9	12	8	19	15
10	9	5	14	11

10-iş, 4-makine probleminin SPD algoritması ile çözümünde (10972564318) sıralaması 223 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (42613579810) sıralaması 225 birim tamamlanma zamanı ve SA ile çözümünde ise (10249761385) sıralaması 231 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.7 Genel durum 10-iş, 5-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5
1	11	10	22	15	18
2	16	12	19	17	17
3	19	17	21	19	19
4	14	9	17	16	17
5	8	6	10	8	10
6	9	7	15	14	15
7	22	17	23	22	22
8	20	20	22	20	21
9	7	4	10	9	9
10	13	10	16	15	15

10-iş, 5-makine probleminin SPD algoritması ile çözümünde (61374528109) sıralaması 231 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (61374528109) sıralaması 231 birim tamamlanma zamanı ve SA ile çözümünde ise (93471061582) sıralaması 247 birim tamamlanma zamanı ile elde edilmiştir.

Ek-A Genel Durum Problemleri İşlem Süreleri

Çizelge A.8 Genel durum 10-iş, 8-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8
1	19	12	14	23	9	17	10	15
2	14	10	13	17	5	13	7	13
3	20	19	20	22	12	20	12	20
4	22	20	21	23	14	21	15	21
5	11	6	7	13	4	9	4	8
6	8	5	6	14	4	8	5	7
7	17	15	17	22	11	17	14	17
8	16	15	15	18	9	15	11	15
9	10	6	9	17	4	10	6	10
10	9	8	8	15	6	8	8	8

10-iş, 8-makine probleminin SPD algoritması ile çözümünde (95148107326) sıralaması 279 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (61021347859) sıralaması 281 birim tamamlanma zamanı ve SA ile çözümünde ise (94106175328) sıralaması 297 birim tamamlanma zamanı ile elde edilmiştir.

Çizelge A.9 Genel durum 10-iş, 10-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10
1	20	21	18	14	17	19	10	16	12	18
2	13	16	12	10	11	13	9	10	10	13
3	15	19	11	11	11	13	10	11	11	12
4	14	15	14	7	8	14	7	8	7	14
5	16	18	10	5	6	12	5	6	5	11
6	13	16	6	4	5	8	3	4	4	7
7	9	12	4	4	4	6	4	4	4	4
8	11	17	9	6	8	9	3	7	5	9
9	18	22	16	13	13	18	8	13	11	17
10	22	24	20	18	20	20	14	18	17	20

10-iş, 10-makine probleminin SPD algoritması ile çözümünde (78361051942) sıralaması 306 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (78361051924) sıralaması 306 birim tamamlanma zamanı ve SA ile çözümünde ise (78491536210) sıralaması 337 birim tamamlanma zamanı ile elde edilmiştir.

Ek-A Genel Durum Problemleri İşlem Süreleri

Çizelge A.10 Genel durum 10-iş, 15-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	22	20	15	19	14	23	19	13	21	10	22	18	13	17	12
2	23	18	11	17	8	25	18	5	25	5	23	12	7	11	4
3	11	9	6	8	5	16	9	1	13	1	15	6	2	6	1
4	10	9	5	9	5	15	9	5	13	2	15	8	5	7	2
5	19	15	9	11	7	21	14	4	21	3	20	11	6	10	3
6	16	14	7	12	4	19	13	3	18	3	19	11	3	11	3
7	15	12	8	11	6	19	12	1	17	1	16	10	5	8	1
8	12	10	7	10	7	16	10	4	14	2	14	7	7	7	4
9	20	16	11	12	9	21	13	7	20	4	21	11	7	11	5
10	17	14	7	12	4	20	14	4	19	2	17	11	4	10	3

10-iş, 15-makine probleminin SPD algoritması ile çözümünde (43875911062) sıralaması 361 birim tamamlanma zamanı ile optimum olarak bulunmuştur. Problemin GA ile çözümünde (43810965127) sıralaması 366 birim tamamlanma zamanı ve SA ile çözümünde ise (61421075398) sıralaması 381 birim tamamlanma zamanı ile elde edilmiştir.

Ek-B Özel Durum Problemleri İşlem Süreleri

Uygulamada kullanılan tüm işlem zamanları, literatüre paralel biçimde [1-25] zaman aralığında üniform dağılıma göre tamsayı olarak seçilmiştir. İşlem zamanları için sıra matrislerinin oluşturulması sırasında, sıralı akış tipi çizelgeleme problemleri özellikleri dikkate alınmış en büyük işlem zamanlarının ilk yada son makinede oluşması sağlanmıştır.

Çizelge B.1 Özel durum 7-iş, 4-makine problemi için işlem zamanları

İş/Makine	1	2	3	4
1	18	11	15	18
2	15	13	14	14
3	9	7	8	8
4	11	9	11	10
5	9	5	7	8
6	20	14	16	18
7	16	11	14	16

Özel durum problemlerinde 7-iş, 4-makine probleminin SPD algoritması ile çözümünde (6172635) sıralaması ile 142 birim tamamlanma zamanı elde edilmiştir. Problemin genetik algoritma ile çözümünde (6241357) sıralaması ile tekrar 142 birim tamamlanma zamanına ulaşılmıştır. Tavlama benzetimi ile çözümde ise (5241736) sıralaması ile 146 birim tamamlanma zamanı bulunmuştur.

Çizelge B.2 Özel durum 7-iş, 5-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5
1	20	10	13	21	25
2	19	16	17	20	22
3	13	8	11	16	18
4	23	15	16	24	24
5	17	11	13	19	21
6	24	22	22	24	25
7	16	14	14	18	20

7-iş, 5-makine probleminin SPD algoritması ile çözümünde (3752146) sıralaması ile 225 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (3746152) sıralaması ve 237 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (4136275) sıralaması ve 236 birim tamamlanma zamanı elde edilmiştir.

Ek-B Özel Durum Problemleri İşlem Süreleri

Çizelge B.3 Özel durum 7-iş, 8-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8
1	19	12	14	13	18	16	7	10
2	25	22	24	23	24	24	18	20
3	15	10	10	10	13	12	8	9
4	19	13	14	14	16	15	10	12
5	12	7	11	9	11	10	4	6
6	23	12	15	13	19	17	9	10
7	18	12	12	12	17	14	6	7

7-iş, 8-makine probleminin SPD algoritması ile çözümünde (2641735) sıralaması ile 236 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (2714365) sıralaması ve 243 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (7415326) sıralaması ve 277 birim tamamlanma zamanı elde edilmiştir.

Çizelge B.4 Özel durum 7-iş, 10-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10
1	18	10	12	18	9	13	15	8	16	11
2	12	7	9	10	5	10	10	5	10	8
3	19	11	14	15	9	14	15	6	15	13
4	20	12	15	17	11	17	17	9	17	14
5	21	14	17	18	10	17	18	7	18	16
6	9	4	6	9	3	8	9	3	9	5
7	25	18	19	20	14	19	20	10	20	18

7-iş, 10-makine probleminin SPD algoritması ile çözümünde (7543126) sıralaması ile 255 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (7614532) sıralaması ve 258 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (7561324) sıralaması ve 264 birim tamamlanma zamanı elde edilmiştir.

Çizelge B.5 Özel durum 7-iş, 15-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	20	19	13	22	18	7	12	21	9	14	15	11	13	16	24
2	18	17	14	18	17	6	13	18	9	16	17	8	15	17	23
3	17	17	15	19	17	8	9	19	8	15	18	9	116	18	19
4	13	11	9	14	12	4	6	13	5	12	13	6	10	13	14
5	23	20	19	23	21	9	13	23	10	20	21	11	20	21	24
6	14	14	12	16	14	5	7	15	5	13	14	5	12	14	19
7	11	9	7	13	9	4	5	12	4	8	9	4	8	9	17

7-iş, 15-makine probleminin SPD algoritması ile çözümünde (7463215) sıralaması ile 433 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (7631245) sıralaması ve 434 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (5326147) sıralaması ve 483 birim tamamlanma zamanı elde edilmiştir.

Ek-B Özel Durum Problemleri İşlem Süreleri

Çizelge B.6 Özel durum 10-iş, 4-makine problemi için işlem zamanları

İş/Makine	1	2	3	4
1	16	15	7	17
2	18	16	11	19
3	11	9	6	13
4	9	5	4	13
5	23	19	17	24
6	21	17	15	22
7	19	13	12	25
8	14	9	7	19
9	12	6	4	17
10	17	11	9	22

10-iş, 4-makine probleminin SPD algoritması ile çözümünde (43981102765) sıralaması ile 220 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (97361458210) sıralaması ve 230 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (72461510938) sıralaması ve 238 birim tamamlanma zamanı elde edilmiştir.

Çizelge B.7 Özel durum 10-iş, 5-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5
1	16	13	12	15	11
2	24	21	19	22	16
3	15	11	10	14	7
4	19	18	16	18	13
5	15	13	11	14	9
6	17	15	12	15	7
7	10	7	5	9	5
8	17	12	10	14	8
9	16	14	13	15	12
10	18	15	14	17	10

10-iş, 5-makine probleminin SPD algoritması ile çözümünde (24106891537) sıralaması ile 222 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (26498105137) sıralaması ve 222 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (67145382109) sıralaması ve 234 birim tamamlanma zamanı elde edilmiştir.

Ek-B Özel Durum Problemleri İşlem Süreleri

Çizelge B.8 Özel durum 10-iş, 8-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8
1	17	19	24	16	17	18	15	25
2	15	18	21	14	12	15	13	24
3	12	16	19	13	10	14	11	21
4	15	18	22	15	14	17	14	23
5	7	13	16	12	8	10	9	19
6	13	15	16	11	12	14	9	20
7	9	12	13	7	8	11	4	15
8	20	21	21	18	19	21	16	21
9	11	14	15	9	10	13	7	19
10	18	19	19	16	17	18	13	21

10-iş, 8-makine probleminin SPD algoritması ile çözümünde (57936241108) sıralaması ile 301 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (79563412108) sıralaması ve 303 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (61021347859) sıralaması ve 321 birim tamamlanma zamanı elde edilmiştir.

Çizelge B.9 Özel durum 10-iş, 10-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10
1	17	13	12	15	7	18	9	15	11	20
2	19	15	16	18	13	19	12	18	14	21
3	18	17	16	17	12	19	13	18	15	21
4	16	13	11	14	9	16	7	15	9	18
5	17	12	10	15	8	17	8	14	10	19
6	18	14	14	15	10	18	11	17	13	21
7	15	11	9	13	4	16	9	15	7	16
8	21	21	20	21	15	21	19	21	20	25
9	14	11	9	13	6	17	8	14	9	17
10	18	15	14	17	11	19	13	17	14	21

10-iş, 10-makine probleminin SPD algoritması ile çözümünde (97451610328) sıralaması ile 356 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (74961531082) sıralaması ve 358 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (78369511042) sıralaması ve 377 birim tamamlanma zamanı elde edilmiştir.

Ek-B Özel Durum Problemleri İşlem Süreleri

Çizelge B.10 Özel durum 10-iş, 15-makine problemi için işlem zamanları

İş/Makine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	17	19	16	21	17	18	14	18	13	22	10	17	15	7	24
2	13	15	13	18	14	14	11	15	9	20	7	14	12	4	21
3	12	16	11	17	13	14	9	14	7	18	6	13	10	5	20
4	15	18	14	18	15	15	13	17	12	16	11	15	13	6	21
5	7	13	6	16	8	10	6	11	4	19	4	9	6	4	22
6	15	17	13	18	16	17	10	17	8	21	7	17	12	2	21
7	11	13	10	17	13	13	7	13	3	18	3	13	9	1	19
8	9	12	8	16	11	12	4	12	3	18	2	12	7	2	21
9	13	16	12	21	15	16	9	16	8	22	5	16	11	5	23
10	16	19	14	22	17	18	11	19	10	23	8	18	14	7	25

10-iş, 15-makine probleminin SPD algoritması ile çözümünde (58732964101) sıralaması ile 385 birim tamamlanma zamanı elde edilmiştir. Genetik algoritma ile çözümde (87526394101) sıralaması ve 388 birim tamamlanma zamanı, tavlama benzetimi ile çözümde ise (73854629110) sıralaması ve 396 birim tamamlanma zamanı elde edilmiştir.

ÖZGEÇMİŞ

Doğum tarihi 18.09.1979

Doğum yeri İstanbul

Lise 1990-1997 Adile Mermerci Anadolu Lisesi

Lisans 1998-2002 Doğu Akdeniz Üniversitesi Mühendislik Fak.
Endüstri Mühendisliği Bölümü

Yüksek Lisans 2002-2005 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Endüstri Müh. Anabilim Dalı

Çalıştığı kurum(lar)

2004-2005 Stratech Stratejik Techizatlar Teknoloji Danışmanlık Sanayi ve Ticaret
Ltd. Şti.