

**GENETİK ALGORİTMA YÖNTEMİ İLE TOPLU ÜRETİM
PLANLAMA**

**YÜKSEK LİSANS TEZİ
Müh. Mehtap VURAL
(507021035)**

**Tezin Enstitüye Verildiği Tarih : 9 Mayıs 2005
Tezin Savunulduğu Tarih : 30 Mayıs 2005**

**Tez Danışmanı : Yrd.Doç.Dr. Ferhan ÇEBİ
Diğer Jüri Üyeleri Prof.Dr. Sıtkı GÖZLÜ (İ.T.Ü.)
Yrd.Doç.Dr. Murat BASKAK (İ.T.Ü.)**

MAYIS 2005

ÖNSÖZ

Günümüzün yoğun rekabet ortamı şirketlerin pazardaki değişimlere hızla cevap verebilecek bir esneklik kazanmalarını gerekli kılmakta ve bu esneklik hızla gelişen bilişim teknolojilerinin sunduğu olanaklar ve fırsatlarla sürekli desteklenmekte ve geliştirilmektedir.

Bilgisayar tabanlı problem çözme teknikleri işletmelerin birçok alanında karşılaştığı problemlerde olduğu gibi üretim sistemleriyle ilgili karşılaştığı karmaşık problemlerin çözümüne ilişkin daha hızlı, daha güvenilir ve daha objektif kararlarının alınmasında önemli katkılarda bulunmaktadır.

İşletmelerin üretim sistemlerine ilişkin stratejilerini oluşturabilmeleri için çözümler aradığı kombinatoriyal optimizasyon problemlerine uygulanabilen yöntemlerden biri olan genetik algoritmalar bu çalışmanın temelini oluşturmakta ve çalışmada genetik algoritmaların genel yapısı, işleyiş tarzı incelenerek algoritmanın toplu üretim planı problemlerine uygulanabilmesi ile ilgili bir bilgisayar programı geliştirilmektedir.

Bu çalışmam sırasında bilgi ve tecrübelerinden yararlandığım, ilgi ve desteğini gördüğüm değerli hocam Yrd. Doç Dr. Ferhan ÇEBİ'ye teşekkür ederim. Ayrıca çalışma sırasında yardımlarını esirgemeyen Beko Elektronik Şirketi Üretim Planlama Koordinatörü Mehmet ÇOKKEÇECİ'ye teşekkürü bir borç bilirim.

Çalışmanın yürütülmesi esnasında sağladığı yardım ve ilgiden dolayı Yusuf BAYRAMOĞLU'na ve programın geliştirilmesi sırasında bize moral veren Gamze ÖZKAN'a en derin teşekkürlerimi sunarım.

Son olarak çalışmalarım sırasında manevi desteği ile her zaman yanımda olan ve benim eğitimimin için tüm olanaklarını kullanan anneme, babama ve kardeşime en içten teşekkürlerimi sunarım.

Mayıs, 2005

MEHTAP VURAL

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
1.1. Evrimsel Algoritmalar	3
2. GENETİK ALGORİTMALAR	6
2.1. Genetik Algoritma Terminolojisi	8
2.2. Genetik Algoritmaların Matematiksel Analizi	10
2.3. Genetik Algoritmaların Güçlü ve Zayıf Yönleri	11
2.4. Genetik Algoritmanın Temel Çalışma Prensibi	14
3. GENETİK ALGORİTMALARDA KULLANILAN PARAMETRELER	18
3.1. Parametre Kodlama Operatörü	18
3.2. Başlangıç Popülasyonu	21
3.3. Üreme (Seçim) Operatörü	22
3.4. Çaprazlama Operatörü	25
3.5. Mutasyon Operatörü	30
3.6. Sonlandırma Operatörü	31
4. GENETİK ALGORİTMALARIN UYGULAMA ALANLARI	32
4.1. Çizelgeleme Problemleri	32
4.2. Hat Dengeleme Problemleri	34
4.3. Tesis Yerleşimi Problemleri	35
4.5. Üretim Planlama Problemleri	36
4.6. Makine Tasarımı Problemleri	36
5. TOPLU ÜRETİM PLANLAMA	37
5.1. Toplu Üretim Planlama Teknikleri	39
5.2. Toplu Üretim Planlama Yöntemleri	41
5.2.1. Doğrusal Karar Kuralı	42
5.2.2. Doğrusal Programlama Yöntemleri	43
5.2.3. Amaç Programlama Yöntemi	44
5.2.4. Grafik Yöntemi	45
5.2.5. Yönetim Katsayıları Yöntemi	45
5.2.6. Arama Karar Kuralı	46
5.2.7. Simülasyon Yaklaşımı	47
5.3. Neden Genetik Algoritma ile Toplu Üretim Planlama?	47

6. GATÜP BİLGİSAYAR PROGRAMI GELİŞTİRME SÜRECİ	51
6.1. Toplu Üretim Planlama Probleminin Tanımı	51
6.2. Toplu Üretim Planlama Probleminin Matematiksel Modeli	52
6.2.1. Rastsal Olarak Üretilen Parametreler	56
6.2.2. Türetilen Parametreler	56
6.3. Genetik Algoritma Modelinin Kurulması	58
6.3.1. Parametre Kodlama Operatörü	58
6.3.2. Başlangıç Popülasyonu Oluşturma	60
6.3.3. Seçim (Üreme) Operatörü	60
6.3.4. Çaprazlama Operatörü	62
6.3.5. Mutasyon Operatörü	63
6.3.6. Sonlandırma Operatörü	64
6.4. Bilgisayar Programının Hazırlanması	64
6.5. GATÜP Programı Problem Çözüm Performansının Değerlendirilmesi	65
6.5.1. Genetik Algoritma Operatörlerinin Çözüm Performansına Etkisi	68
6.6. GATÜP Programının Bir Şirkette Uygulaması	73
6.6.1. Şirket Tanıtımı	73
6.6.2. Uygulama ve Sonuçları	74
7. SONUÇ VE TARTIŞMA	78
KAYNAKLAR	83
EKLER	87
ÖZGEÇMİŞ	93

KISALTMALAR

CLP	: Constraint Logic Programming
CRAFT	: Computerized Relative Allocation of Facilities Technique
CX	: Dairesel Çaprazlama
ES	: Evrim Stratejileri
EP	: Evrimsel Programlama
GA	: Genetik Algoritma
GP	: Genetik Programlama
HGA	: Hibrid Genetik Algoritma
LOX	: Doğrusal Sıralı Çaprazlama
NP	: Non-polynomial
OBX	: Sıraya Dayalı Çaprazlama
OX	: Sıralı Çaprazlama
PBX	: Pozisyona Dayalı Çaprazlama
PMX	: Kısmi Planlı Çaprazlama
SA	: Simulated Annealing
TÜP	: Toplu Üretim Planlama

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1 : Doğal evrim sürecindeki terimlerin genetik algoritmalarındaki karşılıkları ve tanımları.....	9
Tablo 3.1 : Gri kodlama yöntemi.....	20
Tablo 6.1 : Üretim planının bir çözümünün genetik kodlanması.....	59
Tablo 6.2 : Toplu üretim planlama problem değerleri.....	66
Tablo 6.3 : Seçim yöntemine göre elde edilen minimum ve ortalama toplam maliyet değerleri.....	69
Tablo 6.4 : Çaprazlama olasılığına göre elde edilen minimum ve ortalama toplam maliyet değerleri.....	70
Tablo 6.5 : Mutasyon olasılığına göre elde edilen minimum ve ortalama toplam maliyet değerleri.....	72
Tablo 6.6 : 55 ekran televizyon grubuna ilişkin veriler.....	75

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Araştırma uzayının katarlara bölünmesi (Dianati ve diğ., 2003)...	10
Şekil 2.2 : Genetik algoritmaların çalışma prensibinin akış şeması.....	15
Şekil 2.3 : Genetik algoritmaların temel yapısı (Sakawa, 2002).....	17
Şekil 3.1 : Rulet seçim yöntemi.....	24
Şekil 3.2 : İki dizinin tek-nokta çaprazlaması.....	26
Şekil 3.3 : Üç-noktalı çaprazlama.....	27
Şekil 3.4 : Üniform çaprazlama.....	27
Şekil 3.5 : Kısmi planlı çaprazlama.....	28
Şekil 6.1 : İki nokta çaprazlama yöntemi.....	63
Şekil 6.2 : Karşılıklı değişim mutasyon yöntemi.....	63
Şekil 6.3 : GATÜP programının hesapladığı minimum maliyetli toplu üretim planı.....	67
Şekil 6.4 : Çaprazlama olasılığının etkisi.....	71
Şekil 6.5 : Mutasyon olasılığının etkisi.....	72
Şekil 6.6 : 55 ekran televizyon için GATÜP deneme sonuçları.....	76
Şekil 6.7 : 55 ekran televizyon grubu için minimum maliyetli toplu üretim planı.....	76
Şekil A.1 : GATÜP programının çalışma akış şeması.....	87
Şekil B.1 : GATÜP programı kullanıcı arayüzü.....	88

SEMBOL LİSTESİ

P_c	: Çaprazlama olasılığı
P_m	: Mutasyon olasılığı
M	: Bir katarın beklenen örnek sayısı
l	: Bir dizinin uzunluğu
l_d	: Bir dizinin tanımlanan uzunluğu
O	: Bir katarın derecesi
$f_H(t), f_i$	: Uygunluk fonksiyonu
$ortf(t)$	: Mevcut popülasyonun ortalama uygunluğu
H	: Entropi
S	: Dizi seti
X	: Optimizasyon probleminin araştırma uzayı
c	: Şifreleme fonksiyonu
$\hat{c}$	: Şifre çözme fonksiyonu
$p(i)$	: Bireyin seçim olasılığı
$e(i)$	: Bireyin beklenen döl sayısı
V_t	: Üretim hızı.
W_t	: İşgücü miktarı
I_t	: Envanter miktarı
FD_t	: Tahmini talep miktarı
a, b, c, K	: Yönetim katsayı sabitleridir
$g(t)$	: Amaç fonksiyonu

GENETİK ALGORİTMA YÖNTEMİ İLE TOPLU ÜRETİM PLANLAMA

ÖZET

Kombinatorial optimizasyon problemlerine uygulanabilen genetik algoritmalar (GA), doğal genetik seçim ve rastsal arama tekniklerini kullanarak çözüm bulmaya çalışan bir yöntem olarak eldeki çözümlerden hareketle daha iyi çözümler üreterek optimum veya optimuma yakın çözümlere ulaşmaya çalışır. Genetik algoritmalar kendi arama proseslerini oluşturması bakımından pek çok avantaja sahiptir. Bu nedenle üretim operasyonlarının planlanması ve tasarlanması sırasında oluşan problemlerin çözümü için ortak bir çözüm tekniği olan genetik algoritmalarla ilgili araştırmalar artmaktadır.

Basit bir genetik algoritmanın işlem adımları; bir gösterim metodu kullanarak uygun çözümlerin dizilere kodlanması, uygunluk fonksiyonunun değerlendirilmesi, genetik parametrelerin belirlenmesi, seçim stratejisi, genetik operatörler ve durdurma kriteri olarak sıralanabilir.

Bu çalışma temel olarak, son yıllarda optimizasyon problemlerinde yaygın olarak kullanılmakta olan genetik algoritmanın işleyiş prosedürlerini açıklayarak bir toplu üretim planlama probleminin çözümünde uygulamasını amaçlamaktadır. Toplu üretim planları, gelecek satış tahminlerinin karşılanabilmesi için firmanın gelecek kaynaklarının ve mamul envanter düzeylerinin belirlenmesini sağlar. Bu planlar, firmaların stratejik işletme amaçlarının kısa dönemli uygulanabilir operasyonel planlara dönüştürülmesini kolaylaştırır. Bu nedenle üretim sistemlerinin karar verme sürecinin önemli bir başlangıç noktasıdır.

Çalışmada literatürden elde edilen bir toplu üretim planı probleminin çözümüne yönelik bir genetik algoritma geliştirilmiştir. Bu algoritmanın işletmelerde kolay ve hızlı uygulanabilirliğini sağlayan ve farklı genetik operatörlerle çalışılmasına olanak veren bir genetik algoritma bilgisayar programı .Net programlama dilinde hazırlanmıştır. Genetik algoritmanın optimum veya optimuma yakın çözüme ulaşma performansını etkileyen; başlangıç popülasyonu, seçim yöntemi, çaprazlama ve mutasyon olasılıkları gibi parametrelerin uygun değerlerinin, bu genetik algoritma programı için literatürden elde edilen bir toplu üretim problemi değerleri üzerinde deneyler yaparak belirlenmesine çalışılmıştır. Böylece kullanılan modelin geçerliliği araştırılmıştır. Genetik algoritma programı elektronik sektöründe televizyon üreticisi olan büyük ölçekli bir firmada uygulanarak bu firmanın üretim planlama koordinatörü tarafından değerlendirilmiş ve firmalarda uygulanabilir bulunmuştur. Sonuç olarak bu çalışmada literatürde birçok uygulama kısıtlarına sahip toplu üretim planlama problemi çözüm yöntemlerinin pratikte şirketlerde çok fazla uygulama fırsatı bulamadıkları şu günlerde genetik algoritmaların toplu üretim planlama problemlerinde şirketlerde uygulama fırsatı elde edebileceklerine dair birtakım sonuçlara varılarak genetik algoritmanın çözüm performansını artıracak parametre setleri belirlenmiştir.

AGGREGATE PRODUCTION PLANNING WITH GENETIC ALGORITHMS

SUMMARY

Genetic algorithm (GA) which is applied to combinatorial problems is stochastic search algorithms based on natural genetic and selection. GA mimics the selective and evolutionary phenomena that occur in nature by the reproduction and recombination of highly fit chromosomes or solution strings within a string population to form better individuals and hence develops increasingly desirable generations. Genetic algorithm has its own advantages about setting up its search processes. Because of this, research has been carried out to investigate the use of genetic algorithm as a common solution technique for solving the range of problems that arise when designing and planning manufacturing operations. A basic genetic algorithm constitutes of encoding the feasible solutions into chromosomes using a representation method, evaluation of fitness function, determination of genetic parameters, selection strategy, genetic operators and termination criterion.

The main aim of this study is to describe genetic algorithm solution procedures that have been used to optimize a wide variety of complex problems. The study investigates the use of genetic algorithms for solving the aggregate planning problems. Aggregate planning takes expected sales demand and production capacity and translates this into future manufacturing plans for family of products. An aggregate plan is a statement of future resources and finished goods inventory levels that must be made available to enable a manufacturing organization to meet its forecast sales demand. The aggregate planning process must enable the strategic business objectives of a manufacturing organization to be quickly converted into short-term operational plans that are economical to implement.

This thesis firstly presents a genetic algorithm application for solving an aggregate production planning problem. According to this application, a genetic algorithm computer program is prepared by using .Net software language. This program enables firms to use genetic algorithm solution procedures easily for their aggregate plans. Furthermore, users can choice different genetic operators in the program. The performance of the algorithm depends on genetic parameters, such as number of initial population, reproduction operator, crossover rate and mutation rate. In the study, the set of genetic parameters is tried to be optimized to achieve optimal or suboptimal solutions of aggregate planning problems by using the computer program. Besides, the genetic algorithm program is applied in the firm that produces televisions in the electronic sector. Production planning coordinator of the firm evaluates the program and finds the program as applicable in industry.

As a conclusion, the thesis shows that genetic algorithm can be used to solve aggregate production planing problems in firms practically; although nowadays solution techniques of aggregate planning problems in the literature have some limitations for application in industry. And also the thesis gives the optimal parameter settings of the computer program.

1. GİRİŞ

Günümüzde kurumları ve işletmeleri saran iç ve dış ekonomik koşullar ve bunların giderek toplumsal ve sosyal yaşama yansması yaşamı her gün biraz daha zor hale getirmektedir. Lider endüstriyel ulusların pazarları büyötmek, sınırsızlaştırmak ve böylelikle kendi hareket alanlarını, pazarlarını genişletmek adına yürüttükleri stratejiler sonucu küreselleşen dünyamızda rekabet acımasız bir düzeye yükselmektedir. Sayısız rekabet faktörünün geçerli olduđu bu ortamda, eski stratejilerin geçerliliğini yitirmesiyle, yeni değerler çerçevesinde yürütölen rekabet yarışında başarıya ulaşmak için yeni stratejilere gerek duyulmaktadır. Bu stratejiler aslında işletmeler için değışken parametreler içeren birçok zor problemin çözümleri sonucu belirlenmektedir. Bu nedenle günümüzde gerçek koşullara uygun parametrelerle bu problemlerin hızlı, kolay ve rasyonel çözümleri, bu çözümlere ulaşma yöntemlerinin teknolojik gelişmelere uygunluğu giderek önem kazanmaktadır.

Üretim sistemlerinin dinamik ortamı nedeniyle üretime yönelik problemler genellikle oldukça karmaşık problemlerdir. Ayrıca, bu problemlere ait kararlar, zaman kısıtı altında alınan ve çok fazla bilgi kullanımını gerektiren kararlardır. Bu nedenlerden dolayı üretim sistemleriyle ilgili problemler genellikle NP-zor (Polinomsal olmayan (Non-Polynomial) problem sınıfı: üssel zaman karmaşıklık fonksiyonuna sahip olan algoritmaların bulunduđu bütün problemleri içeri) problem sınıfındadır. Bu tip problemlerin optimum veya optimuma yakın çözümlerinin elde edilebilmesi için birçok araştırma yöntemleri ve matematiksel modeller geliştirilmektedir. Günümüzde genetik algoritmalara karmaşık üretim sistemleri problemlerinde optimuma yakın çözümlerin elde edilmesi için kullanılan bir yöntem olarak ilgi artmaktadır.

Genetik algoritma (GA) rassal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayalı bir arama tekniğidir. Genetik algoritma, pek çok problem türü için uygun parametreler ile çalışıldığı taktirde optimuma yakın çözümler veren bir yöntemdir.

Genetik algoritmanın çalışma yöntemi Darwin'in doğal seçim prensibine dayanmaktadır. Genetik algoritma doğadaki canlıların geçirdiği evrim sürecini dikkate almaktadır. Amaç doğal sistemlerin uyum sağlama özelliğini dikkate alarak yapay sistemleri tasarlamaktır.

Teoride genetik algoritmalar en iyi çözüme ulaşmayı garanti edememesine rağmen pratikte kalitesiz olmayan çözümler elde edebilmekte ve bazı problemlerde en iyi çözümü bulmak mümkün olmaktadır. Bu nedenle genetik algoritma, bir çok kombinatoriyel optimizasyon problemi üzerinde başarılı bir şekilde uygulanmıştır.

Küresel pazarlarda rekabet edebilmek için; maliyetlerin azaltılması, kalitenin yükseltilmesi ve ürünlerin toplam üretim zamanlarının azaltılması gerekmektedir. Bu gerekliliklerin yerine getirilebilmesi için üretim sistemlerindeki karar verme süreçlerinin iyileştirilmesi ve geliştirilmesi gerekmektedir. Bir üretim sisteminde, karar vermeyi gerektiren faaliyetler hiyerarşik olarak üç seviyede oluşmaktadır. Bunlar; stratejik, taktik ve kontrol seviyesidir. Stratejik seviyede, üretim planlarının pazar taleplerini karşılaması istenir. Taktik seviyede, planlanan üretim çizelgesi, stok, makine kapasitesi, bakım planı ve işçi verimliliği gibi bazı atölye kısıtları ile koordine edilir. Kontrol seviyesinde ise, planlanan üretim çizelgelerinin uygulanmasını gerçekleştirmek için işlerin akışı sürekli olarak düzenlenir ve beklenmedik olaylarla bozulan çizelgeler derhal güncelleştirilir.

Genetik algoritmaların uygulama alanı bulduğu optimizasyon problemlerinden biri de toplu üretim planlama problemleridir. Üretim sistemlerinin karar verme süreçlerinde toplu üretim planlama problemleri önemli bir yer tutmaktadır.

İşletmelerde politika ve hedefler, stratejik planlar ile sağlanırken bunların somut kararlara indirgenmesi, daha kısa zaman dilimleri (genellikle 1 yıla kadar) için hazırlanan taktik planlarla sağlanmaktadır. Bu düzeydeki planlar, genellikle fonksiyonel yapıdadır ve toplu üretim planlama (TÜP) bunlardan birisidir. Toplu üretim planlama, firmanın uzun ve kısa vadeli problemleri arasında bağlantı kuran bir faaliyettir.

Bu çalışmanın temel amacı; son yıllarda optimizasyon problemlerinde yaygın olarak kullanılmakta olan genetik algoritmanın toplu üretim planlama problemlerinin çözümünde kullanılmasını açıklamaktır. Bu temel amaç çerçevesinde çalışma; genetik algoritma yönteminin geniş ölçüde açıklanmayı, genetik algoritma yönteminin ürünlere olan toplu talep değişimini orta dönemde karşılamak üzere üretim düzeyini değiştirme maliyetlerini minimize etme problemi olarak tanımlanan toplu üretim planlama problemlerinde uygulanmayı, bu uygulamaya dayalı bir bilgisayar programının geliştirilerek bu programın uygun parametrelerinin belirlenmesini ve programın sektörde faaliyet gösteren bir firmanın verileriyle uygulanabilirliğinin değerlendirilmesini hedeflemektedir.

Çalışmada öncelikli olarak ilk bölümde kısaca genetik algoritmaların bir parçası olduğu evrimsel hesaplama teknikleri kavramı anlatılmaktadır. İkinci ve üçüncü bölümlerde genetik algoritma kavramı, tarihçesi, uygulama aşamaları ve genetik algoritma parametreleri açıklanarak dördüncü bölümde üretim sistemlerinde genetik algoritmaların literatürdeki uygulama alanlarından bahsedilmektedir. Beşinci bölümde genetik algoritmaların uygulama alanlarından biri olan toplu üretim planlama ve toplu üretim planlama yöntemleri anlatılmaktadır. Bir sonraki bölümde ise toplu üretim planlama problemi için bir genetik algoritma geliştirilmiştir. Geliştirilen GA temelli toplu üretim planlama optimizasyon algoritması için .Net yazılım programıyla GATÜP (Genetik Algoritma ile Toplu Üretim Planlama) adında bir bilgisayar programı hazırlanmıştır. Optimum toplu üretim planı çalışması ile geliştirilen algoritmanın etkinliği ve geçerliliği incelenmektedir. Bu bölümde ayrıca GATÜP programının uygun genetik algoritma parametreleri hakkında araştırma yapılmıştır. Son bölümde GATÜP programının televizyon üreticisi büyük ölçekli bir firmada uygulanması anlatılmaktadır.

1.1. Evrimsel Algoritmalar

Evrimsel hesaplama tekniklerinin ilk izlerine; bilim adamlarının, 1950 ve 1960'larda evrimsel süreçleri mühendislik problemlerinin optimizasyonunda uygulanabilirliği ile ilgili fikirlerinde rastlanır (**Dianati ve diğ., 2003**).

Günümüzde evrimsel algoritmalar, optimizasyon, sistem tanımlama, veri madenciliği, makine öğrenmesi, tasarım problemleri ve birçok alandaki çeşitli kompleks problemlerin çözümünde doğadaki prensipleri temel edinmiş yenilikçi teknolojiler olarak başarıyla uygulanmaktadır. Evrimsel algoritmalar doğal evrimin temellerine

dayalı araştırma ve optimizasyon teknikleridir. Bu algoritmalar kompleks problemlerin çözümü için evrim sürecini taklit etmektedirler.

Evrimsel algoritmaların asıl önemli formları dört gruba ayrılmıştır **(Biethahn ve Nissen, 1995; Chipperfield, 1997)**;

- Genetik Algoritmalar (Genetic Algorithms- GA)
- Genetik Programlama (Genetic Programming - GP): Genetik algoritmanın farklı çözüm sunan bir varyantıdır. Genetik programlama, bir problemi çözmek için kullanılan bilgisayar programlarının otomatik olarak evrim geçirmesi veya gelişmesi için uygulanır.
- Evrim Stratejileri (Evolution Strategies - ES): Mühendislikte pratik optimizasyon problemlerin çözümü için geliştirilmiştir ve çoğunlukla Almanya'da kullanılmaktadır. Genetik algoritma yöntemine benzemektedir ancak bazı temel farklılıkları bulunmaktadır.
- Evrimsel Programlama (Evolutionary Programming - EP): Evrim stratejilerinin ana kollarından biridir. Evrimsel programlama çözüm sunumunda hiçbir kısıtlama içermez. Genellikle parametre optimizasyonunda uygulanır. Sadece mutasyon operatörü vardır. Çaprazlama veya değişim operatörlerini kullanmaz.

Bu yaklaşımlar, çözümün sunumunda, operasyon yapısında, şema seçiminde, uygulamada ve strateji parametrelerin belirlenmesinde bir takım farklılıklar göstermektedir.

Evrimsel hesaplama teknikleri daha sonra genetik algoritmalar ve evrim stratejileri olmak üzere iki bağımsız uygulamalara yönlendirilmiştir. Genetik algoritmalar, yapay zekanın gittikçe genişleyen bir dalı olarak evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadırlar **(Dianati ve diğ., 2003)**.

Genetik algoritmalar ve evrim stratejileri evrimsel algoritmaların en temel yöntemleridir. Genetik algoritmalar, John Holland ve öğrencilerinin önderliğinde Amerika'da geliştirilmiştir. Bu yöntemde, şifrelenmiş ve uygunluk bakımından değerlendirilmiş genotiplerin seçimi, çaprazlanması ve mutasyonu üzerinde durmaktadır. Çaprazlamanın üzerinde mutasyona göre daha fazla durulmaktadır. Mutasyon, genetik algoritmanın geri plan operatörüdür. Evrim stratejileri Ingo Rechenberg, Hans Paul Schwefe ve öğrencilerinin önderliğinde Almanya'da

geliştirilmiştir. Bu yöntemde, mutasyon üzerinde çaprazlamaya göre daha fazla durulmaktadır. Evrim stratejilerinde amaç fonksiyonu gerçek değerleri içerir kodlanmış değerleri içermez **(Whitley, 2001; Biethahn ve Nissen, 1995)**.

Çalışmanın ikinci bölümünde evrimsel algoritmaların temel yöntemlerinden biri olan genetik algoritmalar hakkında geniş çaplı bilgi verilecektir.

2. GENETİK ALGORİTMALAR

Mantıklı ve hızlı algoritmaların oluşturulamadığı çok sayıda problem sınıfı mevcuttur. Bu problemlerin büyük bir kısmını, uygulama alanı gittikçe artan optimizasyon problemleri oluşturmaktadır.

Zor optimizasyon problemlerinin çözümleri için kullanılan yöntemler, genellikle optimuma yakın sonuçlar veren algoritmalarlardır. Bir çok zor optimizasyon problemlerinin çözümünde optimum değeri garanti edemeyen, olasılığa dayalı algoritmalar kullanılmaktadır. Bu algoritmalar çeşitli çözümleri rasgele seçerek hata olasılığını istenen düzeyde azaltmaktadırlar. Küçük çözüm uzayları için klasik ayrıntılı (exhaustive) metotlar yeterli olabilmektedir. Ancak büyük çözüm uzayları için yapay zeka teknikleri kullanılmalıdır **(Michalewicz, 1992)**. Genetik algoritmalar da bu tekniklerden biridir.

Genetik algoritmalar Darwin'in en iyi olan hayatını sürdürür prensibine dayalı olarak doğadaki biyolojik sistemlerin gelişim süreçlerini taklit etmektedir. Doğal çevreye daha iyi uyum sağlayan bireylerin soylarını sürdürebilmesi ve üstün nitelikli anne ve babadan yine üstün nitelikli nesillerin yetişmesi, hatta yeni nesillerin öncekilerden daha üstün niteliklere sahip olabilmesi; eldeki kompleks bir problemin bilinen bazı çözümlerinin çaprazlanması ile daha iyi çözümler üretilebileceği fikrini doğurmuştur. Bu fikir temel alınarak genetik algoritmalar geliştirilmiştir.

Genetik algoritma, doğal biyolojik evrimi taklit eden stokastik (stochastic) bir global araştırma yöntemidir. Genetik algoritmalar potansiyel çözümleri içeren bir popülasyon üzerinde, daha iyi bir sonuç vermeye uygun olan çözümlerin yaşaması prensibine dayalı olarak çalışır. Her yeni nesilde, problem için uygunluk seviyelerine göre bireylerin seçimi ve çoğaltılması prosesleri ile en iyi sonuca olan yakınlık artar.

Genetik algoritma ilk olarak 1958’de Bremermann tarafından geliştirilmiştir. Bu kavramla ilgili ilk yayını yapan 1967 yılında Bagley’dir (**Dianati, 2003; Engin, 2001**). Ancak genetik algoritmalar üzerindeki ilk çalışmayı Michigan Üniversitesi’nde psikoloji ve bilgisayar bilimi uzmanı John Holland yapmıştır. Holland, evrimsel süreçten esinlenerek canlılarda oluşan genetik işlemleri bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Yaptığı çalışmalarla Katar-Plan (Schema) Teoreminin geliştirilmesine öncelik etmiştir. Bu teoremin detaylarını 1975 yılında yayınladığı “Doğal ve Yapay Sistemlerin Uyumunu” kitabında açıklamıştır. Bu teorem ile genetik algoritmalar, işletmelerin teorik iskeletlerine yerleştirilmeye çalışılmıştır. Teorem ile ilgili ilk gelişim Holland’ın doktora öğrencisi David E. Goldberg’in, 1985 yılında hazırlanmış olduğu gaz boru hatlarının genetik algoritmalar kullanılarak denetlenmesi konulu teziyle sağlanmıştır. Goldberg, 1989 yılında yayınladığı “Genetic Algorithm in Search Optimization and Machine Learning” kitabı ile genetik algoritmaların çeşitli konularda pratik kullanımlarının olduğunu göstermiştir (**Dianati, 2003; Sakawa, 2002; Biethahn ve Nissen, 1995; Eren, 2002**).

Günümüzde genetik algoritmalar birçok değişik formlar almıştır. Bazı durumlarda Holland’ın orijinal formülasyonuna çok az benzerlik gösterebilmektedir. Araştırmacılar farklı tiplerdeki kodlama, çaprazlama ve mutasyon operatörleri, özel amaçlı operatörler ve üreme (seçim) için farklı yöntemler kullanarak deneyler gerçekleştirmektedirler (**Forrest, 1996**). Ancak kullanılan tüm yöntemlerin ortak bir noktası bulunmaktadır; hepsi biyolojik evrimden ve Holland’ın orijinal genetik algoritmasından bazı esinlenmeler içermektedir.

Genetik algoritmalar kendi içerisinde farklı formlara dönüşse de temel yapısı itibarıyla birçok geleneksel araştırma ve optimizasyon yöntemlerinden farklılıklar göstermektedir. **Chipperfield (1997)**’a göre en önemli farklılıkları aşağıdaki gibidir;

- Genetik algoritmalar, popülasyon üzerinde araştırmasını paralel olarak yapar, tek noktada araştırma yapmaz.
- Genetik algoritmalar, olasılığa dayalı geçiş kurallarını kullanır, deterministik kurallar kullanmaz.
- Genetik algoritmalar, problemin parametre setinin kendisinin üzerinde değil şifrelenmiş hali üzerinde çalışır. Bu durum gerçek değerli bireylerin kullanıldığı problemlerde geçerli değildir.

- Genetik algoritmalar, türetilmiş bilgilere veya diğer yardımcı bilgilere ihtiyaç duymaz; sadece amaç fonksiyonu ve ilintili uygunluk seviyeleri araştırmanın yönünü etkiler.

2.1. Genetik Algoritma Terminolojisi

Genetik algoritmalar hem doğal genetik bilimini hem de bilgisayar bilimini temel alan bir tekniktir. Bu nedenle genetik algoritmaların açıklanması için kullanılan terimler bu iki bilimde kullanılan terimlerin bir karışımıdır. Genetik algoritmaların işleyiş süreçlerinin daha iyi anlaşılabilmesi için öncelikli olarak doğal genetik bilimde ve matematiksel programlama alanlarında kullanılan bu terimlerin aralarındaki ilişkiler açıklanmalıdır. Bu ilişkiler Tablo 2.1’de özetlenmiştir.

GA’lardaki terimler doğal genetik terimleridir. Popülasyondaki bireyler veya genotipler ayrıca kromozomlar veya dizilimler olarak da adlandırılmaktadır. Bunun nedeni, GA’da her birey doğadakinden farklı olarak yalnızca bir kromozomdan oluşmaktadır. Her genotip belirlenen problem için bir potansiyel bir çözümü temsil etmektedir (**Michalewicz, 1992**).

Tablo 2.1: Doğal Evrim Sürecindeki Terimlerin Genetik Algoritmalarındaki Karşılıkları ve Tanımları.

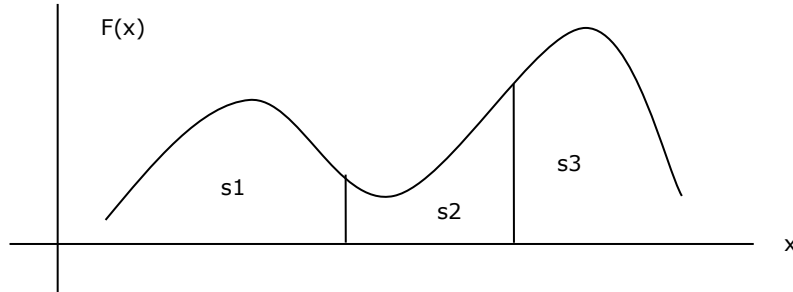
Doğal Süreçteki Terim	GA'daki Karşılığı	Tanımı
Gen (Bit)	Özellik	Kalıtsal yapıda bulunan ve organizmanın karakterini belirleyen kalıtsal yapılardır. Kromozomun ögesi olan ve her biri problemin bir parametresini kodlayan birimlerdir.
Kromozom	Dizi	Genlerin bir araya gelerek oluşturdukları dizidir. Kromozomlar, alternatif aday çözümlerini gösterirler.
Allel	Özelliğin değeri	Her bir genin alabileceği değerlerin kümesine o genin alleli denilir.
Lokus	Özelliğin pozisyonu	Dizi üzerinde problemin bir parametresinin bulunduğu pozisyon.
Popülasyon	Çözüm topluluğu	Bir kombinatorik problemin çok sayıdaki çözümleri veya bu çözümlerin sonlu bir alt kümesi.
Genotip	Yapı / Birey	GA'larda her biri probleme farklı bir çözüm adayı olan kromozomlardan bir popülasyon oluşturulmakta ve bu popülasyon evrimsel yöntemlerle değişikliğe uğratılmaktadır. Kromozom biçimindeki ve problemin bir aday çözümü olan popülasyon öğelerine genotip denilir.
Fenotip	Deşifre edilmiş çözüm	Genotipin çevre ile etkileşimi sonucu oluşan organizmadır. GA'larda ise çözüm alternatifi oluşturmak üzere deşifre edilmiş yapıdır.
Uygunluk fonksiyonu	Amaç fonksiyonu	Çevreye uyum gücünü gösteren başarı ölçütü. Bu fonksiyonun alacağı yüksek değerler problemdeki bireylerin daha iyi sonuç verdiği anlamına gelmektedir.

2.2. Genetik Algoritmaların Matematiksel Analizi

Birçok araştırmacı GA'nın matematiksel davranışını ve matematiksel temelini belirleyen teoriyi tanımlamak için çalışmıştır. Bu konuda yapılan en önemli çalışma Holland'ın Katar Teorisi'dir. Günümüzde bu teori hakkında tartışmaya yol açan birçok araştırma bulunmasına rağmen GA'ların matematiksel davranışını açıklamak için halen yaygın olarak bu teorem kullanılmaktadır.

Plan veya Katar (Schema): Bu kelime Yunanca bir kelimedir. '0', '1' ve '*' karakterlerini içeren dizi anlamındadır. Belirli pozisyonlardaki uygun olan alt dizileri tanımlamak için kullanılan sayı katarlarına veya planlarına denir (**Dianati ve diğ., 2003**). Katar, GA'nın analizinde kullanışlı bir kavramdır. Böylece araştırma uzayının farklı bölümlerindeki noktalar kategorize edilebilmektedir.

Şekil 2.1.'de amaç fonksiyonunun farklı bölümlerinin içinde bulunduğu araştırma uzayı 3 katar (s1, s2 ve s3) ile kategorize edilmiştir. Eğer kromozomlardan bir popülasyon yaratılırsa, kromozomlar s1, s2 ve s3'ün örneklerine sahip olacaklardır (**Dianati ve diğ., 2003**). Bir kromozom, katarda bulunan '*' sembollerinin uygun gen değeri almasıyla oluşturulur. Örneğin 1***** şeklinde 10-bit olan bir katar araştırma uzayının bir alt grubunu tanımlar ve bu katarda tüm kromozomlar 1 ile başlar. '*' sembolü içeren tüm kromozomlar uzayda bir alt uzayı gösterir ve uzayda bir nokta olarak adlandırılmaz. Bir katarın derecesi içerdiği toplam '*' olmayan bit sayısıdır. Örneğin, '*0*1*10*'ın derecesi 4'tür (**Whitley, 2001; Deb, 1996**).



Şekil 2.1: Araştırma Uzayının Katarlara Bölünmesi (**Dianati ve diğ., 2003**).

Katar Teoremi: Birbiriyle yarışan hiperalanları (hyperplanes) temsil eden katarlar, popülasyondaki kendi temsil (çözüm temsili) sayısını bu hiperalanlara ait bölümlerde bulunan kromozomların bağıl uygunluklarına göre düşürür veya yükseltir. Böylece ortalama çözümlerin üzerindeki çözümleri içeren araştırma uzayı bölgelerinde daha fazla deneme yapılır (**Whitley, 2001**). Geleneksel GA formunda sonraki nesil (2.1)'de verilen eşitsizliği sağlar (**Dianati ve diğ., 2003**);

$$M_k(t+1) \geq [f_k(t) / \text{ort}f(t)] M_k(t) \{1 - P_c[l_d/(l-1)]\} (1-P_m)^{O(k)} \quad (2.1)$$

P_c = çaprazlama olasılığı

P_m = mutasyon olasılığı

M_k = k katarının beklenen örnek sayısı

l = uzunluk

l_d = tanımlanan uzunluk

$O(k)$ = derecesi

$f_k(t)$ = k katarının ortalama uygunluğu

$\text{ort}f(t)$ = mevcut popülasyonun ortalama uygunluğu

Eğer katar yapısında yalnızca çaprazlama ve mutasyonun zararlı etkisi dikkate alınırsa yukarıdaki eşitsizlik eşitliğe dönüşür. Böylece sonraki nesillerde yüksek uygunluğa sahip katar örneklerinin sayısı eksponensiyel olarak yükselirken ve az uyumlu olanların sayısı eksponensiyel olarak azalacaktır (**Dianati ve diğ., 2003**).

Katar teoremi ile ilgili birçok eleştiri bulunmasına rağmen en çok gündeme gelen; bu teoremin geleceğe yönelik yalnızca bir nesil için uygulanabilir olmasıdır. Yani teorem tarafından sağlanan sınırlar yalnızca bir nesil için elde tutulabilmektedir. Gelecek nesillerde denemelerin nasıl tahsis edileceği hakkında bilgi vermemektedir.

2.3. Genetik Algoritmaların Güçlü ve Zayıf Yönleri

Genetik algoritma yaklaşımı öncelikli olarak problem çözme yöntemi olarak bilinmektedir. Fakat farklı biyolojik, sosyal ve kavramaya ilişkin sistemler gibi konularda çalışmak ve model oluşturmak için kullanılabilir. Genetik algoritmaların en yaygın uygulaması çok parametrelili fonksiyon optimizasyonudur

(**Marczyk, 2004**). Birçok problem, optimum bir sonuca ulaşmak için bazı girdi parametrelere bağlı karmaşık bir fonksiyon olarak formüle edilebilir. Ancak çok az durumda bu parametrelere bağlı gerçek bir optimum sonuç elde edilebilir. Birçok durumda gerçek optimum elde edilemez ve yalnızca yaklaşık değer veya mevcut en iyi değer biraz daha gelişmiş hali elde edilebilir. Genetik algoritma yöntemi, ikinci durumda iyi değerlerin bulunması için uygun bir yöntemdir. Ancak her yöntemde olabileceği gibi bu yöntemin de diğer yöntemlere göre bazı üstün ve zayıf yönleri mevcuttur.

Genetik algoritmaları geleneksel araştırma yöntemlerinden farklı kılan ve problem çözümündeki etkinliğini artıran bir takım özellikleri aşağıda tanımlanmıştır (**Buckles ve Petry, 1992; Marczyk, 2004**):

- GA'ların en önemli özelliklerinden birisi kesinlikle paraleldir. Diğer birçok algoritma seri özelliktedir; yani bir problemin çözümü sırasında çözüm uzayında belli bir sürede yalnızca bir yönü takip ederek sırayla çözümleri araştırır. Eğer bulunan çözüm optimum değil ise yapılan bütün çalışmayı bırakıp araştırmaya tekrar en başından başlamakta başka yapılacak bir şey yoktur. Ancak genetik algoritmalar çok fazla bireye sahiptir ve çözüm uzayında çok fazla yönde araştırmayı ilk seferde yapabilir. Arama alanı popülasyonun tamamıdır; tek nokta veya noktalarda arama yapmaz. Eğer her hangi bir yol çözümsüz duruma gelirse algoritma kolayca onu elimine ederek başka bir sonuca götüren yol üzerinde çalışmaya başlar. Böylece optimum çözümü bulabilmek için daha fazla şansa sahiptir.
- Paralelizm nedeniyle GA'lar ilk seferde birçok sayı katarını inceleyebilme özelliğine sahiptir. Bu nedenle GA'lar, mantıklı bir süre içerisinde çözüm sunamayacak kadar çok geniş potansiyel çözüm içerebilen bir çözüm uzayına sahip problemlerin çözümleri için uygun bir yöntemdir. Doğrusal olmayan problemlerin büyük bir kısmı bu problem türündedir ve gerçek hayat problemlerinin çoğu doğrusal değildir.
- GA'ların diğer bir güçlü özelliği ise sürekli olmayan, zamanla değişen ve birçok lokal optimum noktaya sahip kompleks uygunluk fonksiyonuna sahip problemlerin çözümünde iyi sonuçlar vermesidir. Çünkü birçok araştırma algoritmaları lokal optimumlar nedeniyle arama alanında en yüksek noktaya ulaştıklarını zannederek araştırma işlemini daha yüksek pikler olmasına rağmen sonlandırırlar.

- GA'lar aynı anda birçok parametreyi kendi amacı doğrultusunda kullanabilme özelliğine sahiptir. Gerçek hayatta problemler yalnızca bir parametrenin minimize veya maksimize edilmesi ile ilgilenmez. Aralarında ödün vererek sonuca ulaşılabacak çok amaçlı problemlerdir. GA'lar paralelizm özelliği nedeniyle bir problem için eşit iyi çözümleri (bir çözüm bir parametreyi optimum yaparken diğer bir çözüm başka bir parametreyi optimum yapar) aynı anda sunabilir. Böylece araştırmacı kendi amaçları doğrultusunda bu çözümlerden birini seçebilir.

- GA'lar problem hakkında bir şeyler bilmez, yalnızca problemin çözümleri üzerine yoğunlaşır. Parametrelerin kendileri ile direkt ilgilenmez, kodlarıyla ilgilenir. Problemin olası çözümlerini rasgele değiştirir ve uygunluk fonksiyonunu kullanarak değişimlerin bir gelişim yapıp yapmadığını değerlendirir. Olasılığa bağlı sonuçlar üreten operatörler kullandığı için GA'lar rassal algoritmalarıdır.

- GA'ların geçiş kuralları, örnekleme için stokastiktir, oysa diğer yöntemler deterministik geçiş kuralları kullanmaktadırlar. Bu yüzden çok çözümlü fonksiyonlar için genetik algoritmalar daha uygundur.

Genetik algoritmalar etkin ve güçlü problem çözüm stratejisi geliştirmesine karşın her problem için uygulanabilir değildir. Bazı belirlenmiş kısıtlamaları mevcuttur. Bu kısıtlamalar **(Marczyk, 2004)**;

- GA'ların oluşturulmasında önemli problemlerden biri problemin kodlanmasıdır. Çözümlerin belirtilmesinde kullanılan dil güçlü olmalıdır. Rassal değişiklikleri tolere edebilmeli ki böylece tehlikeli hatalar veya anlamsızlıklar sürekli oluşmasın.

- Uygunluk fonksiyonun yazılması aşamasında dikkatli davranılmalıdır. Eğer uygunluk fonksiyonu olması gerektiği gibi tanımlanamazsa GA, gerçek problem için bir çözüm bulamayabilir ve farklı bir problemi çözerek araştırmasını sonlandırır.

- Uygunluk fonksiyonunun iyi seçilmesinin yanı sıra GA parametrelerinin (popülasyon büyüklüğü, mutasyon ve çaprazlama oranı, seçim yöntemi) seçiminde de dikkatli davranılmalıdır. Eğer popülasyon büyüklüğü yeterli değil ise GA uygun çözümlerin bulunabilmesi için araştırma uzayında yeterli inceleme gerçekleştiremez. Eğer genetik değişim hızı çok yüksek veya seçim yöntemi zayıf ise faydalı katar dağıtılabılır ve popülasyon yanlış yıkıma girebilir.

- GA’larda bilinen diğer bir problem ise prematüre yakınsamanın oluşumudur. Eğer işlemlerin ilk aşamalarına diğer rakiplerinden çok daha uygun bir birey ortaya çıkarsa bu birey çok daha fazla çoğalır. Böylece algoritmanın global çözümü bulabilmesi için araştırma alanında yeterli inceleme yapmasını engelleyecek bireyler oluşturarak ve popülasyonun farklılığını kendi özelliklerine göre şekillendirerek algoritmanın lokal bir optimuma yakınsamasına neden olabilir. Bu özellikle az sayıda birey içeren popülasyonlar için geçerlidir. GA’larda bağımsız örnekler rassal olarak başlatılan büyük popülasyonlarla sağlanabilir.

Prematüre yakınsama bireyler arasındaki farklılığı çok erken ve hızlı bir şekilde yok ederek genetik yığılma problemine yol açar. Bu problemin giderilmesi için bazı araştırmacılar bazı önerilerde bulunmaktadır. Bunlardan biri popülasyon büyüklüğünün artırılmasıdır. Ancak uygun çözümlerin elde edilebilmesi için popülasyon büyüklüğünde belli bir sınırlama mevcuttur. Diğer çözüm önerisi popülasyondaki kromozomların entropisinin (H) ölçülmesidir. Entropi aşağıdaki formüle göre hesaplanır **(Onwubolu, 2002)**;

$$H = \{[\sum_{j=1}^c (n_{ij} / p)] * \log(n_{ij} / p)\} / \log(c) \quad (2.2)$$

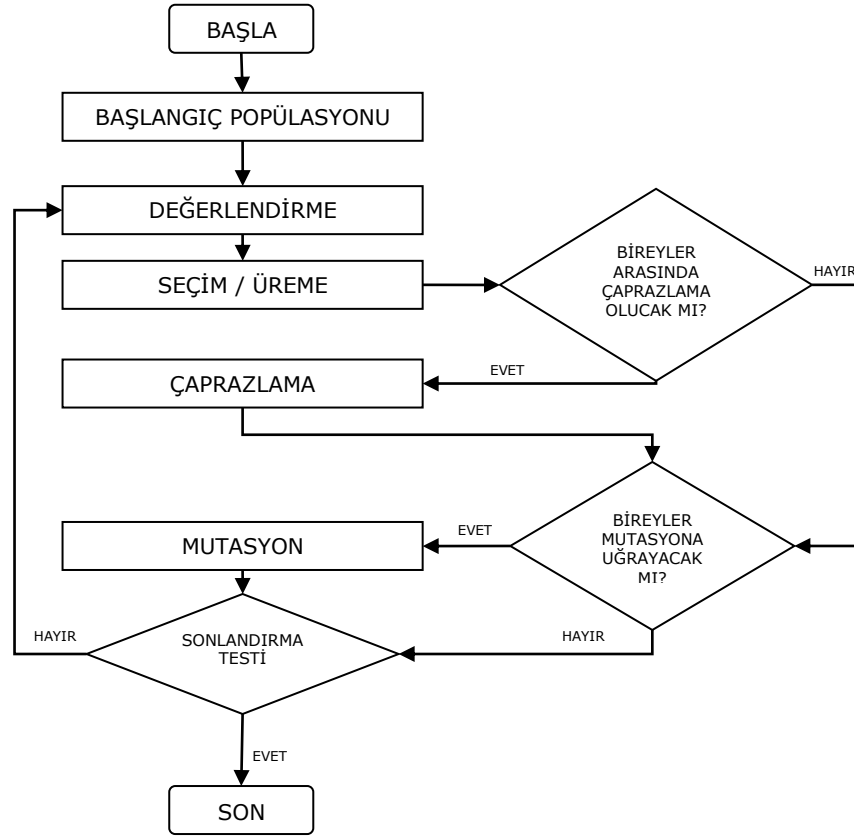
n_{ij} : dizi sayısını, i : alt üniteyi, j : planlanan periyodu, p : çözüm uzayının büyüklüğünü ve c : çözüm uzayının büyüklüğünü ifade eder.

2.4. Genetik Algoritmanın Temel Çalışma Prensibi

Basit genetik algoritmanın işleyiş mekanizması şaşırtıcı şekilde basittir. Yalnızca dizilimlerin kopyalanması ve kısmi değişimlerini içerir. Tam sayıların, karakterlerin vb. dizilimlerinin parçaları basit bir şekilde değiştirilir. Birçok pratik problemde iyi sonuçlar üreten basit bir genetik algoritmanın ana süreçlerinin akış şeması Şekil 2.2’de gösterilmektedir. Genetik algoritmalar Şekil 2.2’de de belirtildiği gibi basit operasyonlardan oluşmasına rağmen güçlü ve etkin optimizasyon tekniklerinden biridir.

Genetik algoritmanın temel prensibi, her adımda, bir önceki nesilden daha yüksek bir amaç değerine sahip olan dizinin, yeni nesil için bir veya daha fazla yeni birey yaratma yeteneğine sahip olduğudur.

Standart bir genetik algoritma, gen dizilimlerin popülasyonu üzerinden çalışır. Her bir birey karar değişkenlerinin tümünü şifreleyen (kodlayan) bir çözüm sunar. Temel bir genetik algoritmanın işlem basamakları;



Şekil 2.2: Genetik Algoritmaların Çalışma Prensiplerinin Akış Şeması.

Aşama 1: Başlangıç - Rastgele olarak başlangıç topluluğu (popülasyonu) yaratılmalı. Diğer bir ifadeyle n adet kromozom içeren popülasyonun oluşturulması (problemin uygun çözümleri) aşamasıdır. Eğer öncelikli baskın (domain) bilgi mevcut ise; bu bilgi, başlangıcı ilgililenen bölgeye doğru kısıtlamak veya başlangıç hüristiği (heuristic) uygulamak şeklinde kullanılabilir. Ancak çaprazlama işleminin etkin gerçekleşmesi için çözümler arasında yüksek derecede farklılığın olduğundan emin olunmalıdır (**Biethahn ve Nissen, 1995**).

Aşama 2: Değerlendirme - Değerlendirme aşamasında mevcut topluluk içerisindeki tüm çözümler için sabit bir uygunluk değeri hesaplanır. Bir bireyin sabit uygunluk değeri kendisiyle uyuşan karar fonksiyonu değerinin dönüştürülmesiyle genellikle hesaplanır. Ancak gerekli görüldüğünde bir çözümün kalitesi deneysel olarak veya

simülasyonla belirlenebilir. Maksimizasyon problemlerinde uygunluk değeri pozitif ve amaç fonksiyonu değeri ile benzer olmalıdır (**Biethahn ve Nissen, 1995**).

Genetik algoritmaların çözümünde ortaya çıkan sorunlardan biri uygunluk değerinin nasıl hesaplanacağıdır. Bununla ilgili birçok yaklaşım öne sürülmüştür. Bu yaklaşımlardan bir kısmı aşağıdaki gibidir (**Dianati ve diğ., 2003**);

- 1- Vektör Değerli Genetik Algoritma (Vector Evaluated Genetic Algorithm): İlk olarak Schaffer tarafından 1985 yılında öne sürülmüştür. Bu yaklaşımda üreme havuzu tek bir amaç uygunluk fonksiyonuna göre değerlendirilen bölümlere ayrılmaktadır. Bu algoritma bazı kısıtlar içeriyor olmasına rağmen referans olarak kullanılmaktadır.
- 2- Ağırlıklar Toplamı Yaklaşımı (Aggregation by Variable Objective Weighting): İlk olarak Hajela ve Lin tarafından 1992 yılında öne sürülmüştür. Bu yaklaşımda her kromozomun toplam uygunluğu, ağırlık değerleriyle derecelendirilmiş ve her bir uygunluk fonksiyonunun toplanmasıyla hesaplanmaktadır.
- 3- Pareto Tabanlı Yaklaşım (Niche Pareto Genetic Algorithm): İlk olarak Horn ve Nafpliotis tarafından 1993 yılında öne sürülmüştür. Bu yaklaşımda turnuva seçim yöntemi ile bireyler seçilmektedir.

Aşama 3: Seçim - Daha iyi çözümlerin elde edilebilmesi için sonraki nesle iyi genlerin geçme şansının kötü genlere göre artırılması için iki kromozomun uygunluğuna göre seçim uygulanır (**Biethahn ve Nissen, 1995**). Seçim operatörü dizilimlerin bulunduğu çözüm uzayına uygulanan ilk operatördür.

Aşama 4: Çaprazlama – Çaprazlama, GA’ların farklı çözüm elemanlarını karıştıran temel bir araştırma operatörüdür. En basit formu tek nokta çaprazlamadır. İlk olarak çaprazlama olasılığına - P_c bağlı olarak çiftleşen partnerlerin çaprazlaması gerçekleşecek mi gerçekleşmeyecek mi belirlenmelidir. Eğer gerçekleşmezse iki birey Aşama 5’e geçer. Gerçekleşir ise farklı şifrelenmiş karar değişkenleri arasındaki sınırları dikkate almadan iki dizilim üzerindeki ortak çaprazlama noktaları rasgele olarak belirlenerek çaprazlama gerçekleştirilir (**Biethahn ve Nissen, 1995**).

Aşama 5: Mutasyon – Mutasyon genetik algoritmaların arka plan operatörüdür. Bu aşamanın ana amacı kaybolan alellin (her bir genin alabileceği değerlerin kümesi) topluluk içerisine tekrar katılmasını sağlamaktır. Tüm topluluk içerisinde sabit bir gen değeri şeklindeki bazı genlerin prematüre birleşmeleri nedeniyle GA istemeden

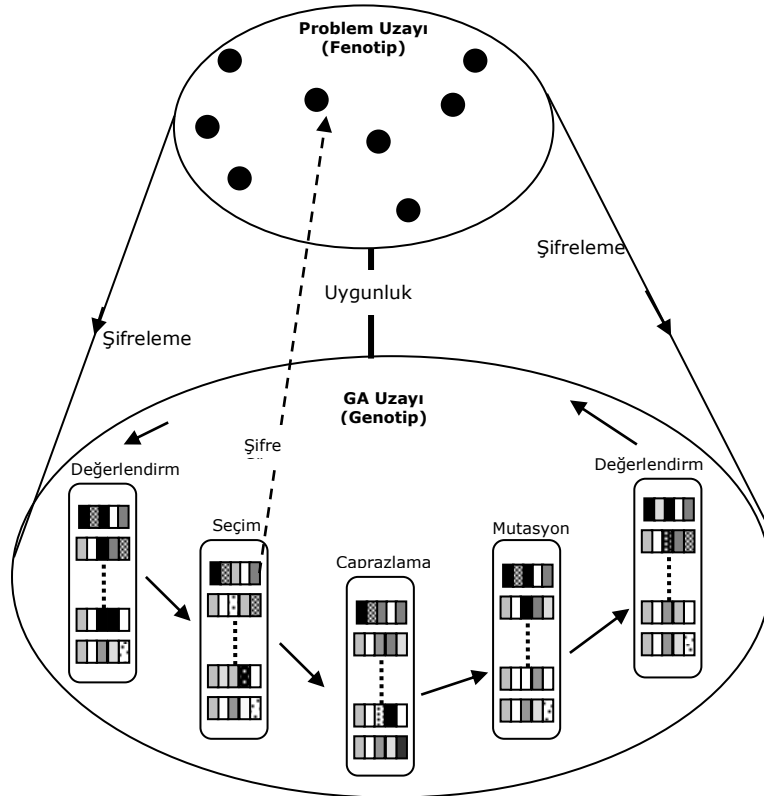
araştırma uzayının ilgilenilen diğer alanlarını eğer bu mekanizma olmazsa istisna edebilir. Bir bireyin her geni önceden belirlenmiş olasılıkla (P_m) mutasyona uğratılabilir (**Biethahn ve Nissen, 1995**).

Aşama 6: Ekleme – Üretim döngüsünün başlangıcında yeni popülasyon boştur. Ancak bu aşamada ebeveynlerden elde edilen yeni bireyler popülasyona eklenir.

Aşama 7: Tekrar – Yeni popülasyon tamamen dolana kadar 3’den 6’ya kadar olan aşamalar tekrar edilir.

Aşama 8: Test Etme – Eğer sonlanma kriteri gerçekleşmiş ise süreç durdurulmalıdır ve son popülasyonun en iyi uygunluk değerine sahip kromozomu çözüm olarak sunulmalıdır. GA ile bir problem çözümünde algoritmanın ne zaman sonlanacağına kullanıcı karar vermektedir. Sonlanma kriteri çözüm kalitesi (istenilen seviye gibi) ile ilgili veya kullanılan kaynaklarla (maksimum iterasyon sayısı gibi) ilgili olabilir. Yani sonucun yeterince iyi olması veya yakınsamanın sağlanması algoritmanın durması için kriter olarak kullanılabilir. Eğer kriter oluşmamışsa, süreç 2. aşamadan itibaren tekrarlanır (**Biethahn ve Nissen, 1995**).

Şekil 2.3’de genetik algoritmaların temel yapısı şekilsel olarak açıklanmaktadır.



Şekil 2.3: Genetik Algoritmaların Temel Yapısı (Sakawa, 2002).

3. GENETİK ALGORİTMALARDA KULLANILAN PARAMETRELER

Genetik algoritma, rasgele arama tekniklerini kullanarak çözüm bulmaya çalışan ve parametrelerin kodlanmasına dayalı bir arama algoritmasıdır **(Deb, 1996)**. Genetik algoritma, pek çok problem türü için uygun parametreler ile çalışıldığı takdirde optimuma yakın çözümler vermektedir. Yani genetik algoritmanın etkinliği kullanılan parametre değerlerine ve yöntemlerine bağlı olarak değişmektedir.

Genetik algoritmada altı ayrı parametre bulunmaktadır. Bunlar;

- Parametre kodlama operatörü
- Başlangıç popülasyonu
- Üreme / Seçim operatörü
- Çaprazlama operatörü
- Mutasyon operatörü
- Sonlandırma operatörü

Çalışmanın devamında bu parametreler sırasıyla açıklanmaktadır.

3.1. Parametre Kodlama Operatörü

Gerçek hayatta üreme ve adaptasyon genetik bilgi seviyesinde gerçekleşir. Dolayısıyla genetik algoritma, araştırma uzayındaki değerler üzerinden işlem yapmaz, o değerlerin bazı kodlanmış versiyonları (diziler) üzerinden işlem yapar. Genetik algoritmanın temel özelliği çözüm uzayı ve kod uzayında çalışmasıdır. S, bir dizi seti ve X, optimizasyon probleminin araştırma uzayı ise **(Bodenhofer,2003)**;

$$c: X \rightarrow S$$

$$x \rightarrow c(x)$$

c : şifreleme fonksiyonu olarak tanımlanır.

$$\hat{c}: S \rightarrow X$$

$$s \rightarrow \hat{c}(s)$$

$\hat{c}$: şifre çözme fonksiyonu olarak tanımlanır.

Şifreleme ve şifre çözme fonksiyonları gerçek problemin ihtiyaçlarına bağlı olarak belirtilmelidir.

Genetik algoritmada, karar değişkenlerinin n -boyutlu x vektörü, bir birey olarak n uzunluktaki s dizisi olarak aşağıdaki gibi temsil edilir;

$$\mathbf{x}: S = s_1 s_2 \dots s_j \dots s_n$$

s dizisi n tane gen içeren bir kromozom olarak ele alınır.

Fenotiplerin genotip olarak tanımlanması işlemine şifreleme, genotiplerin fenotip olarak tanımlanması işlemine şifre çözme denir. Gerçek sayılar, tam sayılar, harfler ve bazı semboller şifreleme için kullanılabileceği halde genellikle bireyler 0 ve 1'lerden oluşan iki tabanlı diziler şeklinde şifrelenmektedir (**Sakawa, 2002**).

Tam veya reel değerler bit dizileri olarak çeşitli yöntemlerle gösterilebilirler. Reel değerler ve tam sayılar standart iki tabanlı kodlama yöntemiyle temsil edilebilir. Ancak bu yöntem bazı dezavantajlara sahiptir. Örneğin 15 ve 16 ikili bit şeklinde 01111 ve 10000 olarak kodlanır. Bu değerler tam sayı uzayında komşudur fakat bit-uzayında komşu değildir. Bunun gibi geleneksel ikili bit gösteriminin bazı saklı eğilimlerinin giderilmesi için gri kodlama kullanılması gerektiği savunulmaktadır. Gri kodlamada, bu diziler sabit bir Hamming uzaklığı (iki komşu onluk düzenindeki sayı için ikili düzende farklı rakam sayısıdır) oluşturur. Bu nedenle bitişik tam sayıların temsilinde farklı kodlama sistemi kullanılarak Hamming uzayında komşu olan bit dizileriyle temsil edilebilir (**Chipperfield, 1997; Biethahn ve Nissen, 1995**). Tablo 3.1'de gri kodlama yönteminin uygulanış biçimi gösterilmektedir.

Tablo 3.1: Gri Kodlama Yöntemi (**Biethahn ve Nissen, 1995**).

Onluk Sayı	Standart İkili Kodu	Hamming Uzaklığı (İkili Kod)	Gri Kodu	Hamming Uzaklığı (Gri Kod)
0	000	-	000	1
1	001	1	001	1
2	010	2	011	1
3	011	1	010	1
4	100	3	110	1

İkili kodlama yöntemi genetik algoritmalarda yaygın olarak kullanılmasına rağmen alternatif kodlama stratejilerine ilgi artmaktadır. Bu yöntemlerden biri gerçek değer temsilidir. Bu yöntemde problem, tam sayı veya reel sayı şeklindeki çözümü doğal olarak temsil edilmektedir. Bu gerçek değerler de genetik algortmada kullanılmaktadır (**Chipperfield, 1997**). Çaprazlama ve mutasyon operatörleri çözümün gerçek değerlerine uygulanmaktadır.

Problem çözümlerinin kromozomlara nasıl şifreleneceği genetik algortmada anahtar niteliğinde bir konudur. Problemlerin başarılı bir şekilde çözümü için en kısa şifreleme yöntemlerinin kullanılması gerekmektedir. Birçok şifreleme yöntemi bulunmaktadır. Ancak kromozomların şifrlenmesi ve şifrelerin çözümü aşamasında üç noktaya dikkat edilmelidir (**Eren, 2002**). Bunlar;

1. Kromozomun uygunluğu kontrol edilmelidir.
2. Kromozomun belirlenen kısıtları sağlayıp sağlamadığı kontrol edilmelidir. Kısıtları sağlamayan bir kromozom araştırma uzayındaki değerleri şifreleyemez ve böyle bir kromozom evrime uğrayamaz.
3. Kodlama haritasında kromozomun tek olup olmadığı kontrol edilmelidir.

Optimizasyon problemleri genellikle birden fazla değişken içerdiği için bu tip problemlerde çok parametrelili kodlama yapılmaktadır. Her değişken önce iki tabanlı olarak şifrenir ve tüm kromozomlar bütün bir kromozom oluşturmak için birleştirilir (**Deb, 1996**).

N tane değişken için değişken vektörü $x = (x_1, x_2, \dots, x_N)$ aşağıdaki gibi şifrelenir (Deb, 1996);

$$\underbrace{1001\dots 01}_{x_1} \underbrace{0101\dots 00}_{x_2} \dots \underbrace{0010\dots 10}_{x_N}$$

Şifreleme yöntemi problemlerin çeşidine göre de farklılık gösterebilmektedir. Örneğin, çizelgeleme problemleri kompleks kodlama sistemleri kullanılmaktadırlar. Genellikle operasyonların sırasını gösteren permütasyonlar kullanılarak temsil edilirler. Çizelgeleme problemlerinde proseslerin bağlı sırası önemlidir (**Whitley, 2001**). Çizelgeleme problemlerinde dokuz çeşit kodlama yöntemi bulunmaktadır (**Cheng ve diğ., 1999**). Bunlar;

- a) Operasyona dayalı kodlama: Bu kodlama yönteminde her bir gen bir operasyonu temsil eder ve m-makine, n-iş probleminde kromozomlar $n \times m$ tane gen içerir. Her gen işin somut operasyonlarını göstermez ve bir iş tam olarak m kere kromozomda görünür.
- b) İşe dayalı kodlama
- c) Liste önceliğine dayalı kodlama
- d) İş çifti ilişkisine dayalı kodlama
- e) Öncelik kuralına dayalı kodlama
- f) Tamamlanma zamanına dayalı kodlama
- g) Makineye iş yükleme grafiğine göre kodlama
- h) Rassal anahtarlı kodlama
- ı) Karışık kodlama

3.2. Başlangıç Popülasyonu

GA yaklaşımının içerdiği parametrelerden birisi de popülasyon büyüklüğüdür. Bu parametre popülasyon içinde (yalnızca bir kuşakta) kaç adet kromozom yani birey olduğunu gösterir. Kromozomlardan oluşan topluluğa yani geçerli alternatif çözüm kümesine popülasyon denir. Popülasyondaki birey sayısı genelde sabit tutulur. Ancak popülasyondaki birey sayısının sabit tutulması gerektiğiyle ilgili genel bir kural yoktur.

Popülasyon büyüklüğü genetik algoritmanın tüm performansını ve verimliliğini etkiler. Genetik algoritmalar genellikle az sayılı popülasyonlarla çalıştıklarında iyi sonuçlar vermemektedirler. Çünkü popülasyon yeterli örnek büyüklüğünü içermemektedir. Eğer kromozom sayısı az olursa GA, çözüm aranan uzayın ancak bir kısmını gezebilmekte ve çaprazlama için fazla bir seçenek bulamamaktadır. Büyük popülasyonların çok sayıdaki hiperalanlardan temsiller içermeye ihtimali daha fazladır. Böylece GA, daha fazla bilgilendirilmiş araştırma yapabilir. Bu gösteriyor ki; büyük popülasyonlar optimum olmayan çözümlere erken yakınsama sorununu engellemektedir. Diğer yandan büyük popülasyonlar her nesil için daha fazla değerlendirme yapmayı gerektirmektedir. Bu durum, GA'nın kabul edilemeyen bir sürede sonuca ulaşmasına neden olabilir. Yapılan araştırmalar, belli bir noktadan sonra popülasyon sayısını artırmanın bir yararı olmadığını göstermektedir (**Grefenstette, 1992**).

Optimum popülasyon büyüklüğünün belirlenebilmesi için bazı araştırmalar yapılmıştır. **Odetayo (1993)**'ın GAPOLE adı verilen GA-tabanlı program ile yaptığı araştırma sonucunda 300'ün optimum bir popülasyon büyüklüğü olduğunu belirlemiş ve daha önceki araştırmalarda belirtilen 60-110'un optimum popülasyon büyüklüğü olduğu düşüncesinin doğru olmadığını, büyüklüğün farklı uygulamalarda farklı bir optimum değer alması gerekebileceğini savunmuştur. Ancak birçok araştırma bu sonucu doğrulamamaktadır.

3.3. Üreme (Seçim) Operatörü

Seçim operatörü dizilerin çözüm uzayına uygulanan ilk operatördür. Bu operatör, bir sonraki nesil için popülasyonda döl verecek bireylerin nasıl seçileceği ve kaç tane yeni birey yaratacakları ile ilgilidir (**Sakawa, 2002**). Bu aşamada her bir kromozom uygunluk fonksiyonuna göre kopyalanır ve gelecek nesilde daha iyi döl verecek kromozomlar seçilir. Üreme işlemi aslında bir yapay seçme işlemidir.

Basit genetik algoritmada, seçme işlemi bireylerin rasgele seçilerek yeni kuşağa aktarılmasıyla oluşturulur. Denge durumlu (Steady-State) genetik algoritmalarda ise genellikle bir veya iki kötü sonuç yer değiştirir ve yeni bireyler yalnızca mevcut popülasyondaki bireylerden yeterince farklı iseler popülasyona girebilirler. Bu yöntem uyumlu bireylerin uzun süre yaşamasına olanak sağlar ve farklılığı korur. Ancak denge durum yaklaşımı uygunluk fonksiyonu stokastik permütasyon içeren

problemlerin çözümü için uygun değildir. Çünkü şans eseri pozitif değerlendirmeye vasat bireyler popülasyonda kendilerini açıkça gösterebilirler. GA'nın, bireylerin serbestçe birleştikleri tek bir geniş popülasyonda sürdüğü varsayılır. Aslında bu varsayım taklit edilen doğadaki sürece uygun değildir. Bu nedenle doğadaki sürece daha yakın olan göç modeli (migration model) ve difüzyon veya komşuluk modeli (neighbourhood model) geliştirilmiştir. Göç modelinde tüm popülasyon birbirinden ayrılan alt popülasyonlar içerecek şekilde tasarlanır. Bu alt popülasyonlardan bazı bireyler karşılıklı değiştirilir. Difüzyon modeli tek bir popülasyon kavramını korumaktadır. Ancak tüm bireylerin komşularını tanımlar. Seçim bölgesel olarak gerçekleştirilir ve bireyler arasındaki tüm etkileşimler bu komşuluk tanımlarıyla sınırlandırılır. Yani bireylerin komşuları arasındaki etkileşimler pas geçilir. Bu iki model de standart yaklaşıma göre daha fazla farklılığı koruyabilmektedirler **(Biethahn ve Nissen, 1995)**.

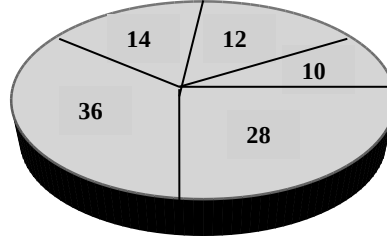
Genetik algoritmanın temel felsefesine göre iyi olan bireyler yaşamını sürdürmek için seçilmeli ve bu bireylerden yeni bireyler yaratılmalıdır. Uygulamada birçok seçim mekanizması bulunmaktadır. Literatürde bulunan ve çok sık kullanılan seçim mekanizmaları genelde altı ana sınıfa ayrılmıştır. Bunlar; 'Rassal Seçim', 'Rulet Seçimi', 'Beklenen Değer Seçimi', 'Sıralı Seçim', 'Turnuva Seçimi' ve 'Elitizm Seçimi'dir. Seçim işlemi temelde, popülasyon içinde uygunluk değerleri kötü olan bireyleri eleyip, elenenler yerine uygunlukları iyi olan bireylerin kopyalanarak çoğalması prensibine dayanır. Uygunluk değerleri iyi olan bireylerin kopyalanmasından sonra oluşan yeni popülasyon çaprazlama işlemine gönderilir.

1. Rassal Seçim: Seçim operatörü popülasyondaki kromozomlardan rasgele seçer.
2. Rulet Seçimi: Bu seçim yönteminde her bir dizi uygunluk fonksiyonuna (f) göre seçilerek kopyalanır. Uygunluk fonksiyonu geçerlilik, karlılık gibi ölçümü istenilen herhangi bir unsur olabilir **(Coley, 1999)**. Mevcut popülasyondaki bireyler rulet çemberinde uygunluklarına göre büyüklüğü ayarlanmış dilimler olarak Şekil 3.1'de gösterildiği gibi düşünülebilir. Uygunluğuna göre birey i'nin seçim olasılığı $p(i)$ aşağıdaki gibidir **(Onwubolu, 2002)**;

$$p(i) = f_i / \sum_{j=1}^n f_j \quad (3.1)$$

Her bir dizinin amaç değeri rulet çemberine yerleştirilir ve dizi sayısı kadar çember çevrilir. Her çevirme işlemine karşılık gelen kromozomlar gelecek nesil bireyini

yaratmak için seçilir. Daha büyük alana sahip bireyin seçilme şansı daha fazla olacaktır. Seçilen kromozomlar gen havuzunda biriktirilir ve aynı diziden gen havuzunda birden fazla mevcut olabilir.



Şekil 3.1: Rulet Seçim Yöntemi.

Rulet seçimi için oluşturulacak algoritma aşağıdaki gibidir (**Sakawa, 2002**);

Adım 1: N tane birey içeren popülasyonda t zamanındaki nesil için uygunluk değerlerini f_i ($i=1,...,N$) ve tüm popülasyonun toplam uygunluk değerini $f_{\text{toplam}} = \sum_{j=1}^N f_j$ hesapla.

Adım 2: $[0,1]$ aralığında rasgele bir sayı $\text{rand}()$ yarat ve $s = \text{rand}() \times f_{\text{toplam}}$ hesapla.

Adım 3: $\sum_{j=1}^k f_j \geq s$ şeklinde en az değerli k'yı elde et ve k. bireyi t+1 zamanındaki nesil için seç.

Adım 4: Seçilen birey sayısı N'e gelene kadar 2. ve 3. adımları tekrar et.

3. Beklenen Değer Seçimi: Rulet seçiminin stokastik hatalarını azaltmak için geliştirilmiş bir seçim yöntemidir (Sakawa, 1992). Bu yöntemde i. bireyin beklenen döl sayısı (3.2)'de verilen eşitliğe göre hesaplanır (**Onwubolu, 2002**);

$$e(i) = f_i / \sum_{j=1}^n (f_j/n) = f_i / f_{\text{ortalama}} \quad (3.2)$$

$e(i)$ 'nin tam sayı değeri kadar i. bireyden sonraki popülasyonda bulunur. $e(i)$ 'nin kesirli kısmı i. bireyin bir kopyasının daha diğer popülasyon'da bulunma olasılığını gösterir.

4. Sıralı Seçim: Rulet seçimi, uygunluk değerinin çok fazla değişkenlik gösterdiği problemlerde sorun oluşturabilmektedir. Örneğin en iyi kromozomun uygunluğu %90 ise diğer kromozomların seçilme şansı azalacaktır. Bunu önlemek için sıralı seçim kullanılabilir (**Kurt ve Semetay, 2001**). Sıralı seçimde mevcut popülasyondaki bireylerin uygunluk değerlerinin sıralamadaki yeri seçim olasılığını

belirlemektedir. Popülasyon beklenen değerlerine göre en iyiden en kötüye olucak şekilde sıralanır. Uygunlukların ölçeklendirilmesine gerek yoktur. Çünkü uygunluklar arasındaki mutlak farklılıklar göz ardı edilmektedir. Bu yöntemde, her bireyin seçim olasılığını belirleyecek doğrusal veya doğrusal olmayan sıralama yöntemleri kullanılmaktadır (**Sakawa, 2002**).

5. Turnuva Seçimi: Standart bir genetik algoritmada bir nesilden diğer nesle geçiş iki aşamalı bir süreçtir. Mevcut popülasyondan seçim yapılarak orta nesil (intermediate population) yaratılır. Orta nesle çaprazlama ve mutasyon uygulanarak gelecek popülasyon yaratılır. Orta nesli oluşturmak için rastlantısal olarak iki dizi seçilir ve orta nesle yerleştirilir. Her neslin iki kez örneklediği umulur. En iyi dizilimin her iki turnuvayı kazandığı kabul edilir ve orta popülasyonda iki kopyası elde edilir. Ortalama dizi bir tane kazanır, bir tane kaybeder ve orta popülasyonda bir kopyası bulunur. En kötü diziler iki turnuvayı da kaybeder ve tekrar üretilmez. Bu yöntem en iyi bireye yönelik 2 eğilimli doğrusal bir sıralama oluşturur. Eğer turnuva kazanan, orta popülasyonda $0,5 < p < 1$ olasılıkla yerleştirilirse eğilim 2'den küçük olur. Eğer turnuva sayısı 2'den büyükse kazanan deterministik olarak seçilir ve eğilim 2'den büyük olur (**Whitley, 2001**).

6. Elitizm Seçimi: Üreme, çaprazlama ve mutasyon işlemleri sonrasında kuşakta bulunan en iyi uygunluğa sahip bireyin sonraki kuşağa aktarılması garanti değildir. Ancak en uygun bireyin diğerlerinin uygunluğundan çok çok yüksek bir uygunluk değeri olması durumunda garanti edilebilir. En uygun bireyin seçilmemesi bu bireyin ölmesi anlamına gelmektedir. Bu durum, problemin en iyi çözümüne ulaşmak için yapılacak araştırma süresini uzatacak, algoritmayı yavaşlatacaktır. Bunu önlemek için çaprazlama ve mutasyon sonrası oluşan yeni kuşağa bir önceki kuşağın en iyi (elit) bireyi çaprazlama veya mutasyona uğramadan, yeni kuşaktaki herhangi bir birey veya en kötü birey ile değiştirilir (**Coley, 1999; Sakawa, 2002**).

3.4. Çaprazlama Operatörü

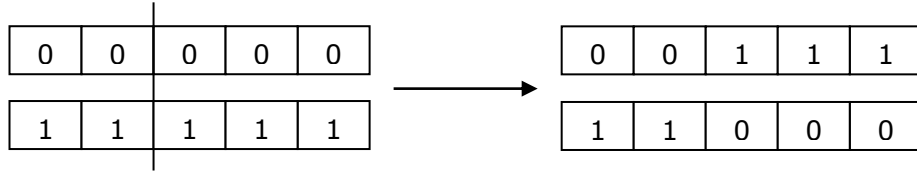
Çaprazlama genetik algoritmaların en önemli araştırma operatörüdür. Bu nedenle hala etkin çaprazlama yöntemlerinin bulunması için çalışmalar sürdürülmektedir. Çaprazlama, iki bireyin dizilerinden yeni bir nesil oluşturma sürecidir. Bu operatörün asıl rolü, popülasyondaki farklı bireylerdeki bilgi parçalarının kombine edilmesidir. Gelecek nesil için iki ebeveynin genetik materyalini tekrardan birleştirir.

Üreme işleminden elde edilen ve iyi bireylerden oluşmuş popülasyon çaprazlama işlemine tabi tutularak, bireyler arasında bir gen aktarımı oluşturulur. Böylece önceki popülasyona göre daha iyi bireylerden oluşmuş yeni bir popülasyon oluşması amaçlanır. Bu işlemde farklı çözümler arasında bilgi değişimi vasıtasıyla araştırma uzayının benzer fakat araştırılmamış bölgelerine ulaşım sağlanır **(Deb, 1996)**.

Çaprazlama operatörü tasarımında iki ana düşünce mevcuttur. Bunlar **(Engin, 2001)**;

- Çaprazlama işlemi ile diziler üzerinde çok az değişiklik yapmak ve bütün varyasyonlarda tek noktalı çaprazlama kullanmak.
- Çaprazlama işlemi ile diziler üzerinde büyük değişiklikler gerçekleştirerek farklı çaprazlama yöntemleri uygulamak.

Klasik genetik algoritmalarda genellikle bir noktalı çaprazlama operatörü kullanılır ve bu yöntem diğer çaprazlama yöntemlerinin öncüsüdür. İki ebeveyn seçildikten sonra dizilerinin parçaları Şekil 3.2’de gösterildiği gibi değiştirilir. Diziler üzerinde değişim noktası belirlenir ve bu noktanın sağında kalan tüm genler karşılıklı değiştirilir **(Biethahn ve Nissen, 1995)**.



Şekil 3.2: İki Dizinin Tek-nokta Çaprazlaması.

Bir noktalı çaprazlama işleminde ilk gen değiştirilemez ve bu genetik algoritmanın operasyon etkinliğini azaltabilir. Bu nedenle çok noktalı çaprazlama yöntemi geliştirilmiştir. N-nokta çaprazlama, bir-nokta çaprazlama şeklinin geliştirilmiş halidir **(Onwubolu, 2002)**. Şekil 3.3’de üç noktalı bir çaprazlama örneği gösterilmektedir. Uygun çaprazlama nokta sayısı dizilimin uzunluğuna bağlıdır.

Aile 1	1	1	1	1	1	1	1	1
Aile 2	0	0	0	0	0	0	0	0
Çocuk 1	1	1	0	0	0	0	1	1
Çocuk2	0	0	1	1	1	1	0	0

Şekil 3.3: Üç-noktalı Çaprazlama.

Genetik algoritmanın araştırma gücünü artırmak için üniform çaprazlama yöntemi geliştirilmiştir (Şekil 3.4). Bu yöntemde her çaprazlama için rastlantısal olarak yeni bir çaprazlama maskesi (crossover mask) yaratılır. Maskedeki sıfır (bir) değerini taşıyan genin bulunduğu pozisyondaki aile 1 (aile 2)'deki genin değeri çocuk 1'e geçer. Çaprazlama maskesinin tersi kullanılarak çocuk 2 yaratılır (**Biethahn ve Nissen, 1995; Sakawa, 2002**).

Aile 1	0	0	1	1	1	1
Aile 2	1	1	1	1	0	0
Maske	0	1	0	1	0	1
Maskenin Tersi	1	0	1	0	1	0
Çocuk 1	0	1	1	1	1	0
Çocuk2	1	0	1	1	0	1

Şekil 3.4: Üniform Çaprazlama.

Genetik algoritmalarda kullanılan diğer çaprazlama yöntemleri aşağıda açıklanmıştır.

1. Kısmi Planlı Çaprazlama (PMX): Goldberg tarafından geliştirilen bu çaprazlama ilk olarak gezgin satıcı probleminde kullanılmıştır. Bu yöntemde iki ayrı iş sırasında rasgele aralıklar belirlenir ve bu aralıkta yer alan işlerin yeri karşılıklı olarak değiştirilir (**Cheng ve diğ., 1999**). Bu yöntem Şekil 3.5'te gösterilen bir örnekle açıklanmaktadır.

Aile 1	2	8	6	4	5	7	1	3
Aile 2	8	7	2	1	3	4	6	5
Aile 1'	2	8	2	1	3	7	1	3
Aile 2'	8	7	6	4	5	4	6	5
Çocuk 1	2	8	6	1	3	7	4	5
Çocuk2	8	7	6	4	5	1	2	3

Şekil 3.5: Kısmi Planlı Çaprazlama.

Bu yöntemde aile dizilerinden ortak bir aralık rasgele seçilir ve bu aralıktaki gen değerleri karşılıklı değiştirilir. Dizide bir genin tekrarlı olarak yer alması yapının uygun olmadığı anlamına gelir. Bu nedenle geçici sonuç veren bu dizilerde uygunsuzlukların giderilmesi için karşılıklı değişim gerçekleştirilir.

2. Dairesel Çaprazlama (CX): Davis, Goldberg ve Linge tarafından geliştirilmiş bir yöntemdir. Bu yöntemde ilk kromozomdan en baştaki gen seçilir ve bu gen yeni diziye yerleştirilir. Bu gene karşılık gelen ikinci kromozomdaki gen belirlenir ve bu değer yeni kromozom üzerine yerleştirilerek dairesel bir şekilde bütün genler belirlenir (**Cheng ve diğ., 1999**).

3. Doğrusal Sıralı Çaprazlama (LOX): Falkenauer ve Bouffouix tarafından geliştirilmiştir. Dairesel çaprazlamanın bir çeşididir. Mevcut popülasyondan seçilen iki kromozom üzerinde rasgele iki alt dizi seçilir. P_1 dizisinden seçilen alt dizi kromozomdan kopar ve boş kalan pozisyonlar belirlenir. Aynı işlem P_2 dizisinde de gerçekleştirilir. Birinci alt dizi P_1 'e, ikinci alt dizi P_2 'ye yerleştirilir (**Cheng ve diğ., 1999**).

4. Sıralı Çaprazlama (OX): Bu yöntem Davis, Goldberg ve Linge tarafından geliştirilmiştir. Rasgele seçilen iki kromozom üzerinde rasgele iki ayrı kesim noktası belirlenir. Belirlenen kesim noktaları arasındaki gen sayısının her iki kromozomda da aynı olmasına dikkat edilir. Kesim noktaları arasındaki genler karşılıklı olarak yer değiştirir. Kromozomlarda gen tekrarları oluşursa bunlar yerine sıra ile soldan sağa doğru kromozomda bulunmayan genler yazılır **(Cheng ve diğ., 1999)**.

5. Pozisyona Dayalı Çaprazlama Yöntemi (PBX): Bu yöntem Syswerda tarafından geliştirilmiştir. Rassal olarak seçilmiş pozisyonadaki genler bir ebeveynden çocuğa kalıtsallaştırılır. Diğer genler diğer ebeveynde bulundukları sıra ile yerleştirilir. Öncelikle pozisyonadaki sayılar rassal tamsayılar şeklinde düzenlenir ve daha sonra bu pozisyonlar rasgele seçilir **(Cheng ve diğ., 1999)**.

6. Sıraya Dayalı Çaprazlama Yöntemi (OBX): Bu yöntemde çaprazlama noktaları rasgele seçilir. İlk kromozomda seçilen noktalara karşılık gelen genler aynen yerlerini korur. İkinci kromozomun seçilen noktalara ait karakterleri ilk kromozomun aynı noktalarındaki genlerin arkasına getirilir. Geriye kalan boş pozisyonlara ikinci kromozomdan aktarılan yeni genler sıra ile soldan sağa yerleştirilerek yeni bir kromozom elde edilir **(Engin, 2001)**.

Çaprazlama yapılacak bireyler ve bunların çaprazlama noktaları bilgisayar tarafından rasgele seçilir. Çaprazlama olasılığı (P_c) çaprazlamanın hangi sıklıkta yapılacağını ve popülasyon içerisinde çaprazlama işlemine tabi tutulacak birey sayısını belirtir. Eğer hiç çaprazlama yapılmaz ise (çaprazlama olasılığı %0) yeni bireyler eski bireylerin aynısı olur ama bu yeni kuşağın eskisiyle aynı olacağı anlamına gelmez. Eğer bu oran %100 olursa yeni bireyler tamamıyla çaprazlama ile elde edilir (Kurt ve Semetay, 2001). Her yeni nesilde $P_c \times$ (popülasyon büyüklüğü) kadar yapı çaprazlama işlemine tabi tutulur. Yüksek çaprazlama olasılığı popülasyonda yeni yapıların daha hızlı oluşmasını sağlar. Ancak bu olasılık çok yüksek ise yüksek performanslı yapılar seçim mekanizması iyileştirme gerçekleştirilmeden atılabilir. Eğer çaprazlama olasılığı çok düşükse, araştırma düşük inceleme hızından dolayı durgunlaşabilir **(Grefenstette, 1992)**.

3.5. Mutasyon Operatörü

Mutasyon, üreme ve çaprazlama işlemlerinin tamamlayıcı bir işlemdir. Genetik algoritmalarda mutasyon işlemi ile genellikle küçük bir olasılıkla (P_m) birey içindeki bir değeri değiştirerek popülasyon içinde yeni bireylerin oluşması sağlanır. Bu işlemin uygulama amacı temelde algoritmanın tıkanmasını önlemektir.

Mutasyon olasılığı (P_m) ise mutasyonun hangi sıklıkta yapılacağını ve işlemde değişikliğe uğrayacak birey sayısını belirtir. Mutasyon olmaz ise yeni birey çaprazlama veya kopyalama sonrasında olduğu gibi kalır. Eğer mutasyon olur ise yeni bireyin bir kısmı değiştirilmiş olur. Eğer bu oran %100 olursa kuşak içindeki bireyler tamamen değişir, %0 olursa hiç değişmeden kalır (**Kurt ve Semetay, 2001**).

Mutasyon oranı genel değerleri (0,01 – 0,001) aralığındadır. GA’da mutasyon oranı değerinin değiştirilmesi kullanışlılığını artırır. Özellikle popülasyonda farklılığın azalması nedeniyle çaprazlamanın etkinliğini kaybettiği optimizasyonun son aşamasında mutasyonun önemi artar ve yüksek mutasyon oranı kullanışlı olur (**Biethahn ve Nissen, 1995**). Seçim işleminden sonra yeni popülasyondaki tüm kromozomlarda bulunan her gen mutasyon oranına göre rasgele değişme ihtimaline sahiptir. Sonuç olarak $P_m \times (\text{popülasyon büyüklüğü}) \times (\text{kromozom uzunluğu})$ kadar mutasyon her nesil için gerçekleşir. Mutasyon oranının çok fazla düşük olması herhangi bir genin sonsuza kadar tek bir değerde kalarak popülasyonun sabit bir değer yakınsamasına neden olabilir. Çok yüksek mutasyon oranı ise araştırmayı tamamen rassal hale getirir (**Grefenstette, 1992**).

Kromozomların mutasyona uğratabilmek için farklı mutasyon yöntemleri kullanılabilir. Bunlar aşağıda açıklanmıştır (**Cheng ve diğ., 1999**);

1. Ters Mutasyon: Mutasyon operatörü seçilen bir genin değerini tersine çevirir (0 ise 1’e; 1 ise 0’a dönüştürür). Bu operatör yalnızca ikili kodlanmış genlerde uygulanabilir.
2. Ekleme Mutasyon: Herhangi bir gen rassal olarak seçilir ve rassal olarak bir pozisyona yerleştirilir.
3. Yerinden Çıkarma Mutasyon: Bir alt dizi rassal olarak seçilir ve rassal olarak seçilen bir pozisyona yerleştirilir. Eğer alt dizi yalnızca bir gen içeriyorsa ekleme mutasyon yöntemine dönüşür.

4. Karşılıklı Değişim Mutasyonu: İki pozisyon rassal olarak seçilir ve bu pozisyonlardaki genler karşılıklı olarak yer değiştirilir.
5. Taşınarak Mutasyon: Bir gen rassal olarak seçilir ve genin bulunduğu pozisyondan sağa veya sola doğru taşınır.

3.6. Sonlandırma Operatörü

Sonlandırma operatörü genetik algoritmanın araştırmaya devam mı edecek veya araştırmayı durduracak mı kararını verebilmesi için kullanılan bir işlemdir. Her nesil oluşumundan sonra genetik algoritma belirlenen sonlandırma kriterini kontrol ederek araştırmaya devam eder veya araştırmayı durdurur. Sonlandırma ile ilgili farklı yöntemler bulunmaktadır. Bunlar;

1. Nesil sayısı: Evrim süreci problem çözücü tarafından belirlenen maksimum nesil sayısına ulaştığında algoritma araştırma işlemini sonlandırır.
2. Evrim Süresi: Algoritmanın araştırma süresi problem çözücü tarafından belirlenen maksimum süreyi aştığında işlem sonlanır.
3. Uygunluk Eşiği: Minimizasyon problemlerinde mevcut popülasyondaki en uygun değeri taşıyan bireyin uygunluk değeri problem çözücü tarafından belirlenen uygunluk eşiğinden az ise, maksimizasyon problemlerinde mevcut popülasyondaki en uygun değeri taşıyan bireyin uygunluk değeri problem çözücü tarafından belirlenen uygunluk eşiğinden yüksek ise algoritma arama işlemini sonlandırır.
4. Popülasyon Yakınsaması: Evrim süreci popülasyon yakınsamaya başladığında sonlanır.
5. Gen Yakınsaması: Kromozomlardaki genlerin değeri problem çözücü tarafından belirlenen yüzde değeri kadar yakınsamaya başladığında evrim süreci sonlanır.

4. GENETİK ALGORİTİMLERİN UYGULAMA ALANLARI

Genetik algoritmalar yapay zekanın gittikçe genişleyen bir kolu olan evrimsel hesaplama tekniğinin bir parçasını oluşturmaktadır. Herhangi bir problemin genetik algoritma ile çözümü, problemi sanal olarak evrim sürecinden geçirerek sağlanmaktadır. Genetik algoritmalar geleneksel yöntemlerle çözümü zor veya imkansız olan problemlerin çözümünde kullanılabilmektedir.

Genetik algoritmalar optimum veya optimuma yakın çözümlerin tanımlanabilmesi için hızlı ve verimli bir araştırma prosedürüdür. Bu nedenle genetik algoritmalar, çok geniş kapsamlı ve farklı alanlardaki problemlerin çözümünün elde edilebilmesi için araştırmalara konu olmuştur (**Sakawa, 2002**). Ekonomi, psikoloji, biyoloji, tıp, bilgisayar bilimi, evrimsel programlama, makine öğrenmesi, veri tahmini ve yapay sinir ağları gibi birçok farklı konu genetik algoritmaların kullanıldığı alanlar için birer örnek oluşturmaktadır.

Üretim planlama ve kontrol alanında, üretim sistemlerindeki işlerin genellikle çizelgelenmesi problemleri üzerinde genetik algoritma uygulamasına geniş bir ilgi bulunmaktadır.

Bu bölümde, üretime yönelik farklı uygulama alanlarındaki problemlerin optimizasyonu konusunda literatürde yapılan genetik algoritma çalışmaları özetlenmiştir.

4.1. Çizelgeleme Problemleri

Çizelgeleme, iyi bilinen zor kombinatoriyel optimizasyon problemlerinden biridir. Bu tip problemler, NP türü problemler kapsamında olduğundan bilgisayarların hızlarının artması ile daha etkin çizelgeleme yöntemleri geliştirilmeye çalışılmıştır.

Permütasyon tipi çizelgeleme problemlerinde temel amaç, minimum tamamlanma süresini veren her bir operasyonun makinedeki yapılış sırasının elde edilmesidir.

Genetik algortimaların çizelgeleme problemlerinde kullanımı iki farklı şekilde gerçekleşmektedir; 1- genetik algoritma yardımıyla permütasyon tipi iş sırası bulma, 2- GA ile bulunan permütasyon sırasının diğer sezgisel yöntemlerle karşılaştırılması.

Reeves (1995), $n/m/P/C_{max}$, akış tipi çizelgeleme problemlerini genetik algortimalar ile çözerek elde ettiği sonuçları tavlama benzetimi (Simulated Annealing-SA) ve komşuluk aralığı tekniği (Neighbourhood Search) ile bulunan sonuçlarla karşılaştırarak GA'nın akış tipi çizelgeleme problemleri için daha iyi sonuç verdiğini göstermiştir.

Chen ve diğ. (1995) farklı ölçeklerde oluşturdukları akış tipi çizelgeleme problemlerini genetik algoritma ile çözmüşler ve elde ettikleri sonuçları Widmer ve Hertz'in sezgisel yöntemi (SPRINT) ve **Ho ve Chang (1991)**'in sezgisel yöntemi ile karşılaştırmışlardır. Genetik algoritmanın daha iyi performans gösterdiğini gözlemişlerdir. **Ho ve Chang (1991)** C_{max} ve ΣF kriterli iki makineli problem için CDS algoritmasına dayanan sezgisel bir yaklaşım sunmuşlardır.

Bolat ve diğ. (2005) son iki tanesi tekrarlı olan üç seri istasyon için oluşturdukları akış tipi çizelgeleme problemini dal ve sınır yöntemi, hibrid hōristik algoritma ve genetik algoritma ile çözmüşlerdir. Genetik algoritmanın kısa sürede her boyuttaki problemler için etkili sonuçlar verdiğini tespit etmişlerdir.

Genetik algortimalar çok amaçlı kombinatoriyel optimizasyon problemlerin çözümü konusunda büyük ilgi gören bir yöntem haline gelmiştir. Çok amaçlı çizelgeleme problemlerinin çözümünde de kullanılabilirliği araştırılmıştır. **Arroyo ve Armentano (2004)**, yaptıkları araştırmada; (i) en son işin tamamlanma zamanını ve maksimum pozitif gecikmeyi en azaltmak, (ii) en son işin tamamlanma zamanını ve toplam pozitif gecikmeyi en azaltmak gibi iki amaçlı akış tipi çizelgeleme problemleri için çok amaçlı lokal genetik algoritma geliştirmişlerdir. Literatürdeki farklı yöntemlerle karşılaştırdıklarında geliştirilen algoritma Pareto optimum çözümlere yakın sonuçlar vermiştir.

Farklı formülasyonlardaki NP çizelgeleme problemlerinin optimuma en yakın olası çözümlerinin belirlenebilmesi için klasik yapıdaki genetik algortimalar farklı yöntemlerle geliştirilerek hibrid genetik algortimalar oluşturulmuştur. Hibrid genetik

algoritmalar (HGA), klasik GA'ların prematüre yakınsama probleminin çözümü ve araştırma verimliliğini artırmak için birçok araştırmaya konu olmuşlardır. Genetik algoritmalarda kullanılan farklı genetik operatörlerin birleştirilmesi veya GA'nın farklı çözüm yöntemleriyle birleştirilmesi şeklinde iki farklı teknikle hibrid genetik algoritmalar geliştirilmektedir. **Gonzalez ve diğ. (1995)**, zorlamalı (no-wait) akış tipi çizelgeleme problemi için farklı genetik operatörlerinin bir arada kullanıldığı bir HGA geliştirmişlerdir. Aynı teknikle **Park ve diğ. (2003)**, atölye tipi çizelgeleme problemlerinin çözümü için yeni genetik operatörlerin ve seçim yönteminin tasarlandığı bir HGA sunmuşlardır. Farklı yöntemlerin birleştirilmesiyle oluşturulan HGA'lardan biri; **Yu ve Liang (2001)** tarafından bir çok kısıt içeren geliştirilmiş atölye tipi çizelgeleme problemi için geliştirilen yapay sinir ağları ve genetik algoritmayı içeren HGA'dır. Bir diğer örnek; **Zhou ve diğ. (2001)** tarafından geliştirilen $n/m/G/C_{\max}$ atölye tipi çizelgeleme problemleri kısa işlem zamanlı öncelik kuralının entegre edildiği ve komşuluk aralığı tekniğinin de adapte edildiği HGA'dır. Çok ürünlü tek akışlı çizelgeleme problemi için Spina ve diğ. (2003) Kısıtlama Mantıklı Programlama (Constraint Logic Programming – CLP) ile birleştirilmiş bir HGA sunmuşlardır. **Gonçaves ve diğ. (2004)**, önceliklerin genetik algoritma tarafından belirlendiği bir öncelik kural setine göre atölye tipi işler için çizelge oluşturan bir HGA tasarlamışlardır. Tüm araştırmalardaki HGA'lar klasik yöntemlerle kıyaslandığında en geçerli ve etkin çözümü sağlamışlardır.

Çizelgeleme periyodu boyunca atölye ortamına herhangi bir anda yeni iş girdileri olabilen durumlarda atölye ortamı dinamik olarak tanımlanır. Bu tip problemler gerçek iş koşullarına en yakın problemlerdir. **Chryssolouris ve Subramaniam (2001)** dinamik atölye tipi çizelgeleme problemlerinin çözümü için genetik algoritma kullanmışlar ve bazı öncelik kurallarına göre hazırlanan çizelgelerden daha iyi sonuç verdiğini saptamışlardır.

4.2. Hat Dengeleme Problemleri

Ürün oluşumu sırasında yapılması gereken işlerin iş istasyonlarına, kayıp süreleri en aza indirecek şekilde atanması olayına hat dengeleme denilmektedir. Hat dengeleme problemlerinin temel amacı, varolan kısıtlar altında işlem sayısının çok ve üretim hızının yüksel olması nedeniyle iş istasyonları arasında oluşabilecek işlem süresi farkları toplamının minimize edilmesidir. İşlemlerin üretim hızları, işlem sıraları ve

işlemlerin gruplanması gibi birçok kısıdın bulunmasına rağmen hat dengeleme problemleri kombinatoriyel problem türündedir. Çünkü işlerin istasyonlara atanmasında birçok yol bulunabilmektedir.

Liu ve Smith (2001), montaj hatlarında işlemlerin sırasını belirlemek için çok seviyeli genetik hat dengeleme planlayıcısı önermişlerdir. Önerdikleri yöntemin verilen herhangi bir ürün tasarımı için optimum veya optimuma yakın işlem sırasını hızlı bir şekilde verdiğini ispatlamışlardır.

Stockton ve diğ. (2004a) hat dengeleme problemleri için bir genetik algoritma geliştirmişlerdir. Genetik algoritma ile elde edilen sonuç konum ağırlıklı dengeleme tekniği ile aynı sonucu vermiştir. Ancak genetik algoritma aynı sonuçta birden fazla işlem sırasını çözüm olarak sunabildiği için planlamacılar içi bir esneklik sağlayabilmektedir.

4.3. Tesis Yerleşimi Problemleri

Tesis yerleşimi, ürün veya hizmet üretmek, kaynak kullanımında en yüksek etkinliği sağlamak için fiziksel tesisin en uygun yerini seçme işi olarak tanımlanabilir. Tesis yerleşimi problemleri, tesislerin zemin alanı ihtiyaçları ve yerleşim sınırları gibi kısıtları dikkate alarak toplam malzeme taşıma maliyetini minimize edecek şekilde m tesisi n yerleşime uygun bir şekilde atamayı amaçlar. Yerleşimdeki tesis ve alan sayısına bağlı olarak alternatif yerleşim sayısının üstel olarak artış göstermesinden dolayı tesis yerleşimi problemleri NP-zor problemler kategorisindedir.

Kulluk ve Türkbey (2004), yaptıkları çalışmada tesis yerleşim problemleri için genetik algoritmaları inceleyerek eşit alanlı büyük boyutlu tesis yerleşimi problemleri için bir genetik algoritma sunmuşlardır. Geliştirilen algoritmanın çözüm kalitesine bağlı performansının problemin zorluğuna göre değiştiğini bulmuşlardır. Problemin boyutu arttıkça çözüm kalitesi azalmaktadır.

Stockton ve diğ. (2004a), tesis yerleşimi için bir genetik algoritma sunarak bu algoritmanın performansını tesis yerleşimi problemlerinin çözümü için kullanılan Computerized Relative Allocation of Facilities Technique (CRAFT) yöntemiyle karşılaştırmışlardır. Genetik algoritma, CRAFT yönteminin sunduğu çözümden daha iyi bir sonuç vermiştir.

4.5. Üretim Planlama Problemleri

Üretim planlama problemlerinin genel amacı, belirlenen üretim hedeflerine ulaşabilmek ya da mevcut satış imkanlarından faydalanabilmek için kaynakların optimum kullanımını tespit etmektir.

Stockton ve diğ. (2004a), toplu üretim planlamada yöneticilerin optimum biçimde tespit etmek zorunda oldukları fason üretim, envanter düzeyi, geç teslim ve mesai gibi karar değişkenlerini optimize edecek bir genetik algoritma sunmuşlar ve bu algoritmanın geleneksel yöntemlerden daha iyi sonuç verdiğini bulmuşlardır. Ayrıca bu çalışmalarında malzeme ihtiyaç planlaması için de bir genetik algoritma geliştirmişlerdir. **Wang ve Wu (2002)** yaptıkları çalışmada çok periyotlu, çok ürünli ve çok kaynaklı üretim çizelgeleme problemleri için bir çözüm yöntemi geliştirmişlerdir. Bu yöntem içerisinde genetik algoritma kullanmışlardır.

Çizelgeleme yapılmadan önce yapılan geleneksel üretim planlama yöntemleri optimum olmayan çizelgelerin oluşmasına neden olabilmektedir. Bu nedenle çizelgeleme problemlerinin üretim planlama ile birleştirilmesi önemlidir. **Morad ve Zalzal (1997)** hücreli üretimde işlerin tamamlanma zamanını, toplam üretilen ıskarta miktarını ve toplam üretim maliyetini minimize etmeyi amaçlayan üretim planlama ve iş çizelgelemenin entegre edildiği problemin çözümünde genetik algoritma kullanmışlardır. Çizelgeleme öncesi proses planının oluşturulduğu geleneksel yöntemle kıyaslandığında daha iyi sonuçlar vermiştir.

4.6. Makine Tasarımı Problemleri

Bir makinenin veya makineyi oluşturan herhangi bir parçasının işlevini beklenen doğrultuda yapmasını ve ömrünü istenilen sürede tamamlamasını sağlamak için konsept tasarım alternatifleri arasından optimum olan seçilmelidir. Bunun gibi en iyi sonucun elde edilmesine yönelik tasarım işlemleri bir optimizasyon problemidir.

Klasik yöntemlerle çözümü, uzun zaman alan ve optimuma ulaşması sürecinde ıraksama veya lokal optimuma yakalanma problemi içeren makine tasarım problemleri için **Öztürk ve Yılmaz (2003)** genetik algoritma yaklaşımını açıklamışlardır. Helisel yay tasarımı için genetik algoritma ile uygun çözüm elde edilmiştir.

5. TOPLU ÜRETİM PLANLAMA

Toplu üretim planlama; talep tahminleri, envanter seviyeleri, sipariş miktarları, işgücü seviyeleri, üretim merkezlerinin kapasiteleri, malzeme temin edilebilirliği, üretim standartları, maliyet standartları ve yönetim politikaları gibi verilerden yararlanılarak uygun kaynakların bir araya getirilmesi sonucu envanter ve üretim maliyetlerinin beklenen değerini minimize eden üretim planının oluşturulması işlemidir.

Toplu üretim planlama yaklaşımı, mamul veya hizmetlere olan toplu talep değişimini orta dönemde karşılamak üzere, üretim düzeyini değiştirme maliyetlerini minimize etme yaklaşımı olarak da tanımlanabilir (**Setyan, 1994**). Toplu üretim planı, kullanılan üretim teknolojisine bağlı olarak genellikle 3 ay ile 18 aylık süreleri kapsayan kararları içermektedir. Üretim planlarının süresi, verilen kararların uygulanabilmesi için yeterli uzunlukta olmalıdır.

Firmaların ürünlerine olan talep zamana göre sabit ise üretimi planlama oldukça basitleşmektedir. Bu durumda planlama periyodunun başlangıcında belirlenen kaynak seviyeleri ile tüm süre içerisindeki talepler karşılanabilmektedir (**Tanyaş, 1982**). Ancak genellikle üretim sistemlerinin ürüne olan talep düzeyleri zamana göre değişmektedir.

Toplu üretim planlamanın temel amacı, tesisin ihtiyaç duyduğu mamulleri karşılayabilmek için üretim miktarı, işgücü seviyesi, envanter bulundurma şekli ve diğer kaynak stratejilerin toplam üretim maliyetlerini minimize edecek, yönetimin ve satışın hedeflerini karşılayacak şekilde kombine edilmesidir (**Onwubolu, 2003**).

Kısaca üretim planlamanın amacı, belirlenen üretim hedeflerine ulaşabilmek ya da mevcut satış imkanlarından faydalanabilmek için kaynakların optimum kullanımını tespit etmeye çalışmaktır. Planlamacının değişen çevre koşullarında yer alan tüm kaynakların kullanımını optimum yapacak şekilde araştırmak için zamanını

harcaması gerekmemektedir. Burada önemli olan konu ideal planın oluşturulması için zaman kaybetmektense yeterli bir planın oluşturulması için çalışılmalıdır.

Toplu üretim planlama prosesi bir karar verme prosesidir. Bu prosesin basamakları aşağıda açıklanmaktadır **(Onwubolu, 2003; Tanyaş, 1982)**:

a) Konu ile ilgili firma hedeflerinin ve planlama politikalarının (planlama periyodunun belirlenmesi gibi) tanımlanması: Toplu üretim planlama süreci organizasyonun amaçlarının ve kısıtlarının belirlenmesi ile başlamaktadır. Bu amaçlar ve kısıtlar dikkate alınmadan hazırlanan üretim planları planlama amacının dışında kalır ve organizasyonun performansını kısıtlar.

Toplu üretim planlama karar verme prosesi başlıca zaman içerisinde değişkenlik gösteren iki farklı bilgidен etkilenir. Bunlardan biri gerçek talebin önceden yapılan talep tahminlerine göre farklılıklar göstermesi ve diğeri gelecek dönem için alınan kararların oluşturduğu gerçek maliyetlerin, karar anında belirlenen veya tahmin edilen maliyetlerden farklı olma durumudur. Bu nedenle toplu üretim planlamada esas alınan periyot, talep ve maliyetlerin planlama dönemlerine göre güvenilir tahminlerin yapılabildiği süreyi içermelidir.

b) Belirlenen planlama periyodunun dönemleri için tahmini talep miktarlarının belirlenmesi: Talep tahminleri belirleme çalışması, geçmişte gerçekleşen satış değerlerine, çevre şartlarına ve tecrübeli yöneticilerin görüşlerine göre gelecekteki satış miktarlarının belirlenmesi işlemidir. Tahminler firma tarafından üretilecek tüm ürünler için ayrı ayrı yapılmalıdır. Değişkenliği az satış tahminlerinin yapılabileceği sistemlerin oluşturulması üretim planlama sürecindeki karışıklıkları azaltacaktır.

c) Belirlenen planlama periyodunun dönemleri için üretim karakteristiklerinin belirlenmesi: Toplu üretim planlamada rasyonel karar verilebilmesi için üretim sisteminin üretim kapasitesi, işgücü seviyesi ve envanter düzeyi gibi karakteristiklerinin belirlenmesi gerekir.

Bunların dışında üretim prosesinin özellikleri, darboğaz durumunda olabilecek iş istasyonları, üretim kapasitesinin talep değişimlerine karşı çabuk ayarlanabilme özelliği ve fason imalat kuruluşları olan ilişkilerin durumu gibi birçok bilginin üretim planlama sürecinde kullanılması gerekmektedir.

d) Maliyetlerin belirlenmesi veya tahmin edilmesi: Toplu üretim planlama probleminin çözümü için kullanılan stratejilerden kaynaklanabilecek maliyetler iki grupta toplanmaktadır:

i – Üretim düzeyini değiştirme maliyetleri: Üretim hızını (fazla mesai, atıl kapasite), işgücü seviyesini, fason üretim düzeyini artırma ve azaltmadaki maliyetleri kapsar.

ii – Envanter düzeyi değiştirme maliyetleri: Elde bulundurma maliyetlerini (kapital bağlama, sigorta, vergi, nakliye, ürünün muhafaza edilmesi) ve elde bulundurmama maliyetlerini (siparişlerin bekleme, satış kaybı, müşteriler üzerindeki olumsuz etki) kapsar.

e) Toplu üretim planlama probleminin çözümü: Bu problemi çözmek için geliştirilmiş yöntemlerin kullanılarak alternatif planların geliştirilmesi sürecidir.

f) Hedefleri karşılayan optimum planın seçilmesi.

5.1. Toplu Üretim Planlama Teknikleri

Toplu üretim planlama stokastik bir prosestir (**Kuşsan, 1991**). Planlamayı amaçladığı olaylar yakın gelecekte gerçekleşmektedir. Planın yapıldığı ve planlanan olayların gerçekleştiği süre içerisinde farklı bir takım olaylar gerçekleşebilmektedir.

Toplu üretim planlama ile ürünlere olan talep değişimini orta dönemde karşılamak üzere üretim düzeyini veya envanter düzeyini değiştirme tekniklerine yöneticiler karar vermektedirler. Bu çerçevede, değişen talep düzeylerinin karşılamak üzere bazı temel stratejiler uygulanmaktadır. Bu temel stratejiler toplu üretim planları için kullanılan genel stratejilerdir. Aslında, bu stratejiler her üretim sistemi için geçerli olmayabilmektedir. Bu nedenle üretim sisteminin tipine göre yeni stratejilerin oluşturulması mümkündür. Aşağıda değişken talep düzeylerini karşılamak üzere uygulanabilecek temel teknikler tanımlanmıştır:

a) İşgücü Büyüklüğünü Ayarlamak: Yeni işçi alma veya işçi çıkarma politikalarıyla işgücü seviyesi değişen talebin karşılanabileceği şekilde ayarlanmaktadır. Yeni işçiler işe alındığında seçme, eğitime ve düşük verimlilik, gerektiğinde yeni vardiya eklenmesi nedenleriyle maliyetleri yükseltmektedir.

İşçilerin işten çıkarılması durumunda firmanın sosyal imajının zedelenmesi, elde kalan işçilerdeki moralin bozulması nedeniyle verimlerinin düşmesi gibi elle tutulmayan maliyetlerinin yanı sıra işsizlik sigortası ve tazminat gibi işten çıkarmanın diğer hesaplanabilir maliyetleri yükselmektedir (**Erfan, 1994**).

b) Üretim Hızını Ayarlamak: Talep dalgalanmaları üretim hızının değiştirilmesiyle karşılanıyor ise, artışlar halinde firmanın üretim olanaklarını zorlayacağı için makinelerin hızla aşınması, bakım gereklilerinin yerine getirilememesi ve işçilere giderek artan fazla çalışma ücreti gibi fazla mesai maliyetleri yükselecektir; düşüşler halinde ise aylak işçilik maliyetleri yükselecektir.

c) Envanter Düzeyini Ayarlamak: Talep değişiminin stoklardan karşılanmasıdır. Yol açtığı maliyetler elde bulundurma ve elde bulundurmama maliyetleridir.

Envanterde bulunan hammadde, yardımcı madde veya ürünlere bağlanmış sermayenin maliyeti, muhafaza süresinde oluşabilecek bozulma maliyetleri, depolama ihtiyaçlarından oluşan maliyetler, sigortalama maliyetleri ve vergi maliyetleri elde bulundurmada dolayı katlanılması gerekli maliyetlerdir.

Envanter bulunmaması nedeniyle karşılanamayan müşteri siparişlerinin oluşturduğu kayıp kazançlar ve müşteri memnuniyetsizliğinden kaynaklanan müşterinin gelecek olumsuz tutumları gibi maliyetlere katlanılmaktadır.

Envanter düzeyini değiştirme ile ilgili kararlar, envanter düzeyinin üretim düzeyi ile satış düzeyi arasındaki farktan oluşmasından dolayı bağımlı bir değişkendir (**Tanyaş, 1982**).

d) Fason İmalat Düzeyini Ayarlamak: Ürünlerin farklı firmalarda üretilmesiyle talep değişimleri karşılanır.

e) Geç Teslim Etme Düzeyini Ayarlamak: Yüksek olan dönemlerde müşteri siparişlerinin biriktirilerek geç teslim edilmesi şeklinde talep değişiklikleri sağlanır. Bu durumda müşteri memnuniyetsizliği ve sözleşmelere dayalı ceza maliyetleri oluşacaktır. Bunun dışında müşterilerin taleplerinin rakip firmalardan karşılamalarından ve satış gelirlerinin azalmasından doğan maliyetlerde bu tekniğin uygulama sürecindedir.

f) Talep Miktarını Değiştirmek: Reklam, fiyat indirimi, hediyeler gibi çeşitli kampanyaları uygulayan pazarlama stratejileri ile dönemlik talep miktarları değiştirilebilmektedir.

Toplu üretim planlama, yönetim tarafından göz önüne alınması gereken kaynak kullanım seçeneklerinin sayısının artırmaktadır. Gelecek planlanan periyot içerisinde oluşacak talep dalgalanmalarının karşılanması için ne kadar envanter kullanılacağı, bu dalgalanmaları, sadece işgücü düzeyinde değişiklikler yapmak suretiyle karşılama imkanı, istikrarlı bir işgücü sağlanarak talepteki dalgalanmaların, iş saatlerinin veya üretim hızlarının değiştirerek karşılanabilirliği ya da fason kullanılarak karşılanması seçeneklerini oluşturur. Birçok durumda, bu tekniklerin birkaçının kombinasyonu yalnızca birinin kullanılmasından daha etkin olacaktır. Her stratejiye ilişkin oluşacak maliyetler vardır ve bu seçeneklerin akılcı bir bileşimi aranmalıdır.

5.2. Toplu Üretim Planlama Yöntemleri

Toplu üretim planlama şirketlerin insan ve ekipman kaynaklarını tahmin edilen müşteri taleplerini karşılamak üzere en iyi şekilde kullanmak için çalışır. Aynı zamanda aynı kaynakları kullanan tüm ürünlere olan sınırlı bir dönemdeki toplam talebi karşılamak üzere optimum üretim, envanter ve iş gücü seviyelerini belirler. Üretim planlama ve kontrol sistemlerinde toplu üretim planları aslında ana üretim çizelgeleri için bir kısıt niteliğindedir.

Toplu üretim planı hazırlamak için öncelikle ürünlerin gruplandırılması gerekmektedir. Gruplandırılmış ürünlerin toplu üretim planlarının oluşturulması için birçok yöntem kullanılabilir. Toplu üretim planlama problemlerinin çözümü için 1950 yılından beri çok sayıda yöntem geliştirilmesine rağmen endüstride bu yöntemler geniş kullanım alanları bulamamışlardır (**Nam ve Logendran, 1992**). **Nam ve Logendran (1992)**, literatürde uygulanan üretim planlama yöntemleri hakkında yaptıkları geniş çaplı araştırma sonuçlarına göre bu yöntemleri optimum veya optimumla yakın sonuç verebilme yeteneklerine göre sınıflandırmışlardır. Planların optimum olup olmadıkları planların uygulanması için gerekli olan maliyetlerle ölçülmektedir. Toplu üretim planlama yöntemleri;

A. Optimum çözüm yöntemleri

1. Doğrusal karar kuralı
2. Doğrusal programlama yöntemleri
3. Amaç programlama yöntemi
4. Grafik yöntemi

B. Optimuma yakın çözüm yöntemleri

1. Yönetim katsayıları yöntemi
2. Arama karar kuralı
3. Simülasyon yaklaşımı

Bu yöntemlerinde dışında toplu üretim planlama için höristik tekniği kullanılabilmektedir. **Gilbert ve Madan (1991)**, toplu üretim planlama problemleri için höristik tekniği kullanmışlardır.

5.2.1. Doğrusal Karar Kuralı

Doğrusal karar kuralı yöntemi, bütünleştirme seviyesindeki her bir planlama süreci boyunca stok, işgücü seviyeleri, üretim oranlarının planlanması kararlarını vermek için 1955’de Holt, Modigliani, Muth ve Simon tarafından geliştirilen bir kuadratik programlama yaklaşımıdır (**Holt ve diğ., 1960**). Yöntem ayrıca HMMS olarak da adlandırılmaktadır ve yöntemde taleplerin deterministik olmadığı varsayılır.

Doğrusal karar kuralı yönteminde optimum işgücü ve üretim düzeylerini tanımlamak için ikinci derece fonksiyonlar oluşturulmaktadır. Bunun yanı sıra işçi alma ve çıkarma maliyetleri arasındaki ilişkiler, mesai üretim, envanter bulundurma veya bulundurmama maliyetleri arasındaki ilişkiler oluşturulan maliyet eğrileriyle ortaya konulur. Bu ikinci derece fonksiyonların türevi doğrusal fonksiyon olduğundan genelleştirilmiş karar kuralları doğrusal bir yapıya sahip olmaktadır. Optimum karar kuralları ikinci derece maliyet fonksiyonlarını minimize ederken, toplu üretim planlama oranı ve işgücü seviyelerini de belirler.

Doğrusal karar kuralında eğrileri işçilerin çıkarılması veya alınması, mesai kullanılması, envanterlerin elde bulundurulması ve bulundurulmaması gibi nedenlerle yükselir. Bu maliyet eğrilerinin türevi alınarak işgücü maliyetleri ile ilişkilendirilir (**Holt ve diğ., 1960**). **Kamien ve Li (1990)**, doğrusal karar kuralı

yöntemine fason üretim bileşenin katarak geliştirmiştir. Ancak üçüncü karar kuralının yönteme eklenmesi yöntemin geliştirilme sürecini zorlaştırmış ve birçok duruma da uymamıştır.

Bu yöntemin en önemli eksikliği başlangıç varsayımı olan beklenen maliyetlerin kuadratik olmasıdır. Kuadratik fonksiyon düşük işgücü değişikliklerinin gerçek maliyetlerinin daha az tahmin edilmesine neden olabilir. Ayrıca aylak işgücü maliyetleri için iki misli etki oluşturacaktır. Diğer bir dezavantajı doğrusal karar kuralı formülasyonu planlanan periyot içerisinde bir dönemden diğerine geçişteki üretim hızı ve işgücü seviyesi değişim maliyetlerine yeterince önem vermemektedir. Üretim hızında veya envanter düzeylerinde kısıtlar uygulanmasına izin vermez. Ayrıca doğrusal karar kuralı değişkenler veya kısıtlar için tamsayılı değerleri bulunduramaz. Yöntem, talep tahminleri sürecinin doğasında bulunan hata miktarlarına karşı duyarlı değildir. Bu tip dezavantajlarına karşın doğrusal karar kuralı, optimize edici olması, iki karar kuralının bir kez türetildikten sonra uygulamasının basit olması, dinamik olması, gerçek endüstri koşullarını çok gerçekçi sergilediği için ve toplu üretim planlama tekniklerinden daha düşük maliyetli planlar oluşturabildiği için tercih edilebilmektedir (**Nam ve Logendran, 1992**).

5.2.2. Doğrusal Programlama Yöntemleri

Doğrusal programlama yöntemi yöneylem araştırması tekniklerinden doğrusal programlamanın hem simpleks hem de dağıtım modelleri kapsamında oluşturulmuştur. Bu yöntem, üretim ve depolama maliyetlerinin minimizasyonu ve talebin mevcut kapasite kısıtları içinde karşılanmasının sağlayacak şekilde üretken kaynak birimlerine tahsis edilmesi prensibi üzerine geliştirilmiştir (**Erfan, 1994**).

Toplu üretim planlama için kullanılan doğrusal programlama yöntemlerinden en çok bilinen Bowman'ın 1956'da geliştirdiği toplu üretim planlama problemi için doğrusal programlamanın transportasyon yöntemidir (**Tanyaş, 1982**).

Lineer toplu üretim modelleri tüm planlanan dönemlerde üretimle ilişkili değişkenlere ihtiyaç duymaktadır. Farklı periyot dönemlerinde gerçek üretim miktarı, periyottaki değişkenlerin toplamına eşit olmaktadır. Eşit olmadığı dönemlerde eşitsizlik durumundaki model kısıtları (örneğin; periyottaki mesai saat miktarı $\leq$ herhangi bir miktar, vb.) uygulanmaktadır.

Toplu üretim planlama için kullanılan doğrusal programlama yöntemlerinde talep miktarlarının deterministik olması, üretim maliyetlerinin ve üretim hızı değiştirme maliyetlerinin belirlenen dönemlerde lineer olarak artması, siparişin gecikmeli olarak karşılanması ve satışların kaydedilmesi halinde maliyet yükünün olmaması, geri siparişlerin genellikle olmaması, envanter düzeyinin planlanan dönemde sınırlandırılabilmesi gibi bazı kabuller mevcuttur. Özellikle taleplerin deterministik olması nedeniyle sektördeki planlamacılar gelecek tahminleri mutlak gerçek şekliyle tahmin edememektedirler. Ayrıca endüstride tüm maliyetlerin lineer olması söz konusu değildir. Bu yöntem işe alma ve işten çıkarma maliyetlerini dikkate almamaktadır **(Nam ve Logendran, 1992)**.

5.2.3. Amaç Programlama Yöntemi

Amaç programlama yöntemi mevcut iş çevresinde toplu üretim planları etkileyecek üretim yöneticilerinin birden fazla amaçlarının dikkate alınması için geliştirilmiştir. Yöntemin genel amacı bütün yönetim amaçlarını kısıtlar altında birleştirebilmektir.

Yöntem doğrusal programlamayla benzerlikler göstermektedir. Lee ve Moore tarafından 1974 yılında geliştirilmiştir **(Nam ve Logendran, 1992)**. Yöntemde öncelikle karar değişkenleri tanımlanır. Daha sonra yönetim amaçları belirlenerek önceliklik sırası yapılır. Model öncelik sırasına göre çözülür. Çözüm yöntemi iterasyona dayanır. En büyük öneme sahip amaç öncelikle gerçekleştirilir ve diğerine geçilir.

Bu yöntem kullanılarak kapasite, sevkıyat çizelgesi, sabit işgücü düzeyi, üretim, envanter ve mesai üretim ile ilgili amaçlar arasında ödün verilebilir. Yöntem maliyetlerle ilgili geniş çaplı bir çalışma yapılmasını gerektirir **(Nam ve Logendran, 1992)**.

Amaç programlama yöntemi de doğrusal programlama yöntemindeki gibi maliyetlerin doğrusal olduğu varsayımına sahiptir. Ancak yöntemin, yöneticilere amaçlarını formülasyon içerisine dahil etme imkanı vermesi, amaçların gerçekleşmesi için gerekli kaynak birleşimlerinin farklı çözümlerinin tanımlaması gibi güçlü yönleri mevcuttur. Doğrusal programlama modelinden daha zor formüle edilir **(Nam ve Logendran, 1992)**.

5.2.4. Grafik Yöntemi

Grafik yönteminde geliştirilen basit grafiklerle görsel olarak tahmini talep ihtiyaçlarıyla gelecek üretim kapasiteleri karşılaştırılabilir. Bu yöntemle oluşturulan planlar tahmini talep miktarları ile mevcut kapasitenin birkaç değişkene bağlı olarak karşılaştırılmasına olanak verir. Toplu üretim planı, garantili sonuç vermeyen ancak sınırlı hesaplamalarla ve deneme yanılma yöntemiyle elde edilir.

Grafik yöntemler, basit oldukları ve alternatif planların geniş bir planlama dönemi içinde görülmesini sağladıkları için kullanımı yaygın bir yöntemdir. Ancak grafik modelin durgun özelliği ve hiçbir şekilde optimizasyonu sağlamaması bu yöntemin eksiklikleridir **(Nam ve Logendran, 1992)**.

5.2.5. Yönetim Katsayıları Yöntemi

Bowman tarafından 1963 yılında geliştirilmiş bir modeldir **(Nam ve Logendran, 1992)**. Sezgisel bir yaklaşımdır. Bu yöntemde belirli şartlar altında yöneticinin karar verme davranışını gösteren katsayılar belirlenmektedir. Yönetimin katsayıları yöntemi, tecrübeli bir karar vericinin karar verme prosesindeki tüm ölçütlerden oldukça haberdar olduğu ve bu kararları tüm ölçütleri değerlendirerek verdiği varsayımına dayanır. Bu varsayımına göre alınan tüm kararlar optimume oldukça yakın olmaktadır.

Yönetim katsayıları yöntemi herhangi bir periyottaki üretim hızını aşağıdaki genel karar kuralına göre hesaplamaktadır **(Nam ve Logendran, 1992)**;

$$V_t = aW_{t-1} - bI_{t-1} + cFD_{t+1} + K \quad (5.1)$$

V_t : t. periyottaki üretim hızı.

W_{t-1} : önceki periyottaki işgücü miktarı

I_{t-1} : önceki periyottaki bitiş envanter miktarı

FD_{t+1} : sonraki dönemin tahmini talep miktarı

a, b, c ve K katsayı sabitleridir.

Yöntemde tüm katsayılar geçmişte yönetimin uyguladığı toplu üretim planı kararlarına çoklu regresyon analizi uygulanarak elde edilmektedir. Yöntem, yöneticilerin karar verme prosesini tekrar etmesinden dolayı insan davranışlarının

değişkenliğinden kaynaklanan yönetsel karar değişkenliklerini azaltır. Ancak regresyon modelinin belirli yönetici grubu tarafından verilen kararlara dayalı olmasından dolayı personel değişiklikleri modeli geçersiz kılmaktadır. Ayrıca kısa bir periyotta geçmiş kararların regresyonu yöntemi hatalı sonuçlara götürebilmektedir **(Nam ve Logendran, 1992)**.

5.2.6. Arama Karar Kuralı

Arama karar kuralı yöntemi 1967 yılında William H. Taubert tarafından geliştirilmiştir **(Tanyaş, 1982)**. Toplu üretim planlama probleminin çözümü için geliştirilen birçok yaklaşım problemin maliyet yapısının lineer, kuadratik veya başka formda olması gerektirmektedir. Bu nedenle araştırmacılar karmaşık endüstriyel yapının daha gerçekçi olarak değerlendirebilecekleri bir model arayışına gitmişlerdir **(Nam ve Logendran, 1992)**. Bu yöntemin temeli, amaç fonksiyonunun optimum değerinin bir bilgisayar arama prosedürü yardımıyla bulunması esasına dayanmaktadır.

Arama karar kuralı yönteminde bilgisayar doğrudan arama yöntemiyle maliyet ölçütü bir nokta değerlendirmekte, elde edilen sonuç önceki deneme sonuçlarıyla karşılaştırılmakta ve sezgisel yöntemler dizisine dayanan bir hareket tarzı izlemektedir. Bu şekilde yeni bir nokta saptanmakta ve fonksiyonun daha iyi bir değeri bulununcaya kadar veya bilgisayar işleyiş zaman sınırı aşıncaya kadar işlem tekrarlanmaktadır **(Erfan, 1994)**.

Bu yöntemin en önemli avantajı çözüm sürecinde periyodik olarak değişen kapasite düzeylerine bağlı olarak çok farklı maliyet fonksiyonlarına yer verebilmesidir. Bu nedenle bu yöntem değişen operasyonel koşullara adapte olabilmektedir. Ancak yöntemin uygulanabilmesi için bilgisayar temini gibi bazı maliyetlere katlanması, yöntemin uzman kişilerce kullanılabilmesi ve çözülebilen problemin karmaşıklık düzeyinin bilgisayar kapasitesiyle sınırlandırılması gibi bazı dezavantajları mevcuttur **(Nam ve Logendran, 1992)**.

5.2.7. Simülasyon Yaklaşımı

Simülasyon yaklaşımı, katı kuralları olan lineer veya kuadratik maliyet formüllerine sahip yöntemlere göre bir kademe daha gelişmiş bir toplu üretim planlama yöntemidir. Özellikle karmaşık maliyet yapılarında kullanılabilir (Nam ve Logendran, 1992).

Bu yöntemde tecrübeler veya tesisin mevcut durumuna göre başlangıç bir plan oluşturulmaktadır. Amaç fonksiyonu hiçbir kısıt içermez ve oluşturulan planın performansını belirler. İşgücü düzeyi, mesai, envanterler, fason üretim ve birçok teknikle ilgili değerler lokal minimum elde edilene kadar değiştirilir (Nam ve Logendran, 1992).

Bu yöntemin en önemli dezavantajı kısıtlı sayıda kural içerebilmektedir. Çünkü her kural ayrı bir simülasyon çalışması gerektirmektedir. Bu yöntemin uygulanması ve geliştirilmesi maliyetleri yüksek. Çözüm için kullanılan bilgisayar prosedürleri optimuma yakın sonuçlar veren yöntemlerin optimum sonuçlarına benzer şekilde sonuçlar vermeyi garanti etmez (Nam ve Logendran, 1992).

5.3. Neden Genetik Algoritma ile Toplu Üretim Planlama?

Stockton ve Quinn (1995)'e göre mevcut toplu üretim planlama yöntemleri tesislerde çok az uygulama alanı bulmaktadır. Çünkü toplu üretim planlama problemleri için geliştirilen birçok yöntem aşağıda belirtilen bir takım eksikliklere ve sınırlamalara sahiptir. Bunlar (Stockton ve Quinn, 1995);

1. Yöntemler durum bağımlıdır yani her yöntem belirli durumlarda uygulanabilmektedir. Bu yüzden farklı durumların planlaması için değişiklik yapmadan direkt kullanılamamaktadırlar. Bu durum, malzeme ve işgücü maliyetlerinin yükselmesi, pazardaki rekabet faaliyetlerinin artması gibi operasyon koşullarının değişmesi durumunda planlama problemi için kullanılan yöntemin sınırlı olarak kullanılabileceği veya tamamen işlerliğini yitireceği anlamına gelmektedir.

2. Yöntemlerin uygulanabilmesi için elde etmesi çok zor veya ölçülemeyen bilgilere

ihtiyaç duyulmaktadır. Bu bilgiler genellikle pratikte istenilen formatta hazır bulunmamaktadır. Yeni işçi eğitimi, işçi çıkarma maliyetleri ve buna bağlı verimlilik azalmasına bağlı maliyetlerin belirlenmesi ihtiyaç duyulan bu bilgilere örnek olarak gösterilebilir.

3. Toplu üretim planlama problemleri için geliştirilen yöntemlerin çoğu gerçek hayattaki problemler için optimum veya optimuma yakın sonuçlar verememektedir. Çünkü gerçek hayatta bu problemler kompleks planlama değişkenlerine sahiptir. Elde edilen planlar genellikle maliyet ilişkili optimum planlardır. Maliyetler dışındaki amaçların yöneticiler tarafından tekrar düşünülmesi gerekmektedir.

4. Mevcut yöntemler üretim yöneticilerinin anlayamayacağı karmaşık ve zor matematiksel prosedürlere sahiptir. Bu nedenle yöneticiler bu yöntemleri kullanmak istememektedirler.

Son yıllarda, toplu üretim planlama süreci aşamalarında yeni düzenlemelere gidilmesindeki en önemli amaç, hızla değişen ekonomik, sosyal ve diğer çevre şartlarına uyumlu, değişebilen, kolay geliştirilebilecek planlama yöntemlerinin tasarlanmasıdır. Planlamacılar bu gereksinimlere en etkin şekilde cevap vermek zorunda kaldığı için, toplu üretim planlama çözüm yöntemlerinde yeni arayışlar içine girilmiştir. Bu nedenle, toplu üretim planlama sürecinin önemli bir aşamasını oluşturan optimizasyon ve yeni çözüm yaklaşımları araştırma çalışmalarında ilgi odaklarından birisi olmuştur **(Stockton ve Quinn, 1995)**.

Literatürde çeşitli klasik optimizasyon yöntemleri yer almaktadır. Her yöntemin kendine özgü avantajları ve dezavantajları bulunmaktadır. Bir problemi çözmek için seçilen yöntem başka bir problemde iyi sonuç vermeyebilmektedir. Genetik algoritmanın en büyük avantajı da bu noktada ortaya çıkmaktadır. Genetik algoritmalar problemlerden bağımsız çalışırlar. Kullandıkları sadece problemin uygunluk fonksiyonudur. Bu nedenle, genetik algoritmaların toplu üretim planlama optimizasyon problemlerinde klasik yöntemlerin karşılaştıkları olumsuzlukları gidermek için kullanılabilirlikleri araştırılmaktadır.

Planlamacılar, içinde bulundukları koşullara uygun olarak en iyi çözüm veren bir yöntem seçme zorunluluğundan genetik algoritmalar ile kurtulabilmektedirler. Genetik algoritmalar tüm planlanan durumlar için kullanılabilir bir tekniktir. Talep

tahminleri, maliyetler, sipariş çevrimleri, planlama süresi ne olursa olsun aynı prosedürler kullanılarak problem çözülebilmektedir.

Planlamacıların optimum veya optimuma yakın çözümler elde edebilmesi için genetik algoritmalarda diğer yöntemlerdeki gibi kompleks prosedürler anlaması gerekmemektedir. Genetik algoritma yaklaşımı planlamacıları alternatif toplu üretim planlarının uygunluklarını belirleyen amaç fonksiyonlarını geliştirme yönüne odaklayacaktır. Basit prosedürleri planlamacıların anlamasını kolaylaştıracaktır.

Literatürde genetik algoritmaların diğer uygulama alanlarına bakıldığında bu yöntemin kompleks ve zor optimizasyon problemlerin çözümünde uygun sonuçlar verdiği görülmektedir. Toplu üretim planlama problemlerinde de problemin zorluğu ne kadar artarsa artsın optimum çözümü elde etmeyi sağlayan genetik algoritma prosedürlerinin etkinliği azalmayacaktır. Genetik algoritma prosedürlerinin etkinliği yalnızca genetik algoritma parametrelerine bağlı olarak değişmektedir. Toplu üretim planlama problemlerinde genetik algoritma parametreleri ile ilgili literatürde **Stockton ve Quinn (1995)**, **Stockton ve diğ. (2004a)** ve **Stockton ve diğ. (2004b)** çalışmalarından faydalanılabilmektedir. Gelecekte bu konuyla ilgili yapılacak diğer çalışmalar parametre seçimini kolaylaştıracaktır.

Mevcut problemlerin hiçbirisi gerçek dünya problemlerini yansıtamamaktadır. Çoğunlukla maliyetlerle ilgili amaçlarla ilgilenmekte diğer amaçları göz ardı etmektedirler. Genetik algoritma değerlendirme fonksiyonuna nicel veya nitel kriterler eklenebilmektedir ve toplu üretim planı için planlanacak değişkenler için bir sayı sınırı yoktur (**Stockton ve Quinn, 1995**).

Genetik algoritmaların toplu üretim planlamada kullanımının diğer bir avantajı tedarik zincirinde yer alan malzeme ihtiyaç planlama süreci ile birleşimi kolaylaştırmaktadır. Müşteri taleplerini karşılama prosesi hammaddelerin tedariki ile başlar, üretim ve diğer toplu operasyonlarda işlenmesi ile devam eder ve müşterilere ulaştırılması ile sonlanır. Ticari işletmelerde bu prosesler hiçbir eksiği olmaksızın birbirleriyle fonksiyonel sınırlarla bölünmüştür. Günümüzde tüm bu proseslerin planlanması için farklı algoritmalar kullanılmaktadır ve birleşimi zorlaşmaktadır (**Stockton ve Quinn, 1995**).

Genetik algoritma yaklaşımı farklı proseslerin planlama aktivitelerinin birleşerek bir proses olarak düşünülmesine olanak vermektedir. Bu nedenle toplu üretim planların ürün tiplerine ayrılması problemi ortadan kalkmaktadır. Genetik algoritmaların paralel arama sürecine sahip olması bu birleştirme sürecinin tasarlanmasına olanak vermektedir.

6. GATÜP BİLGİSAYAR PROGRAMI GELİŞTİRME SÜRECİ

Bu bölümde genetik algoritma prosedürlerini kullanarak çalışan toplu üretim planlama problemlerinin çözümü için kullanılacak bir bilgisayar programının geliştirme süreci, programının geçerliliğinin ve uygulanabilirliğinin araştırılması anlatılmaktadır.

6.1. Toplu Üretim Planlama Probleminin Tanımı

Bu çalışmada optimum veya optimuma yakın çözümü aranan toplu üretim planı probleminin genel karakteristikleri aşağıdaki gibidir;

1. Toplu üretim planlama için planlanan periyot 1 yıllık bir süredir. Başlangıç dönemi herhangi bir yılın Ocak ayının ilk günüdür, bitiş dönemi ise Aralık ayının son günüdür. Planlanan periyot 12 aylık dönemlere ayrılmıştır.
2. Toplu üretim planlama probleminde planlanan her dönem için tahmini talep miktarları, normal üretim miktarları, başlangıç dönemi başlangıç stok miktarı, her bir birim için mesai maliyeti, fason üretim maliyeti, geç teslim maliyeti ve elde bulundurma maliyeti bilinmektedir.
3. Toplu üretim planlama probleminde üretim hızını artıracak stratejilerden yalnızca fason üretim ve mesai üretim stratejilerinin uygulanacağı varsayılmaktadır. İşgücü düzeyi sabit kabul edilmekte ve problemde işgücü düzeyini etkileyecek işçi alma veya çıkarma gibi stratejiler uygulanmamaktadır. Tahmini talepleri envanterden karşılamak, taleplerin yüksek olduğu dönemlerde siparişleri biriktirerek müşteriye geç teslim etmek gibi stratejiler problem içerisinde bulunmaktadır.
4. Yalnızca bir ürün çeşidi veya ürün grubu için toplu üretim planı oluşturulması istenmektedir.

5. Firmanın normal üretim miktarı sabit olduğu için bu dönemlerdeki normal üretim maliyetlerinin hesaplaması kolaydır ve sabittir. Bu nedenle normal üretim maliyetleri problem içerisinde minimize edilecek maliyetler içerisinde dahil edilmemiştir.

Şirket politikaları şirket üst yöneticileri tarafından oluşturulmaktadır. Üretim planlama bölümü yöneticiler tarafından oluşturulmuş şirket politikalarını dikkate almak zorundadır. Bu çalışmada optimum veya optimuma yakın çözümü aranan toplu üretim problemi için şirket politikalarına bağlı kısıtlar aşağıda tanımlanmıştır:

1. Mesai ile üretilen ürünlerin her bir birimi için bir maliyete katlanması gerekmektedir. Bu maliyet normal üretim maliyetinin üstünde olacaktır. Bu nedenle mesai üretilebilecek ürün miktarı için bir üst sınır getirilmelidir.
2. Fason ile üretilen ürünlerin her bir birimi için bir maliyete katlanması gerekmektedir. Bu nedenle fason üretilebilecek ürün miktarı için bir üst sınır getirilmelidir.
3. Ürünlerin stokta bulundurulmaması sonucu müşteriye siparişlerin geç teslimi durumu oluşacaktır. Bur durumda her bir birimi için bir maliyete katlanması gerekmektedir. Bu nedenle geç teslim edilebilecek ürün miktarı için bir üst sınır getirilmelidir.
4. Planlanan periyodun dönem sonlarında elde kalan envanter miktarının her bir birimi için elde bulundurma maliyetine katlanması gerekmektedir. Bu nedenle son stok miktarları için bir üst sınır getirilmelidir.

Çalışmada kullanılan problem için maliyet hesaplamalarında maliyetlerin ürün miktarı ile doğrusal arttığı varsayılmaktadır. Örneğin, bir ürünün toplam fason üretim maliyeti toplam ürün miktarı ile bir birimin fason üretim maliyetinin çarpılması ile elde edilmektedir. Diğer mesai üretim maliyeti, elde bulundurma maliyeti ve geç teslim maliyetleri de benzer şekilde hesaplanmaktadır.

6.2. Toplu Üretim Planlama Probleminin Matematiksel Modeli

Toplu üretim planlama süreci içerisinde planlamacılar üst yönetimin belirlediği hedefleri dikkate almak zorundadırlar. Bu hedefleri mevcut toplu üretim planlama problemi çözüm yöntemi içerisinde çeşitli kısıtlar veya parametreler olarak tanımlayarak dahil etmelidirler. Aksi halde elde edilecek toplu üretim planları

organizasyonun genel hedefleri ile uyuşmayarak firma için performans ve verimlilik artırıcı bir unsur olmaktan çıkacaktır. Üretim planlama ve kontrol sürecinin başlangıç basamağı olan toplu üretim planının firma yönetimiyle uyuşmayacak sonuçlar vermesi bu sürecin diğer basamakları olan ana üretim çizelgeleme ve malzeme ihtiyaç planlama gibi basamakları da yanlış yönlendirecektir.

Çalışmada dikkate alınan toplu üretim planlama probleminin tanımlanması kısmında yöneticiler tarafından belirlenen fason üretim, mesai üretim, geç teslim ve dönem sonu envanteri miktarlarının belirli bir sayıyı aşmaması hedefi aşağıdaki formüllerle sağlanmıştır (**Onwubolu,2002**);

$$\text{Mesai üretim üst sınırı} = (\text{maksimum talep miktarı} - \text{maksimum normal üretim miktarı}) / \infty \quad (6.1)$$

$$\text{Fason üretim üst sınırı} = (\text{maksimum talep miktarı} - \text{maksimum normal üretim miktarı}) / \infty \quad (6.2)$$

$$\text{Geç teslim üst sınırı} = \text{mesai üretim sınırı} + \text{fason üretim sınırı} \quad (6.3)$$

$$\text{Son stok üst sınırı} = \text{başlangıç stoğu} + \text{geç teslim üst sınırı} \quad (6.4)$$

“ ∞ ” değeri mesai üretim üst sınırı ve fason üretim üst sınırı değerlerinin birbirinden farklı olabilmesini sağlayan ve bu değerlerin sabit olmamasını sağlayan bir değişkendir. **Onwubolu (2002)** tarafından önerildiği şekilde problemin çözümünde rassal olarak 2 ,3, 4 tamsayı değerlerinden birini almaktadır.

İşletme yöneticileri hedeflerinde tüm stratejiler için sabit birer üst limit belirleyebilirler. Ancak bu çalışmada üst limitler sabitlenmemiştir. Çünkü geliştirilecek bilgisayar programı çok farklı üretim ve tahmini talep değerlerine göre çözüm üretecektir. Sabit alınacak üst değer miktarları çok farklı üretim ve tahmini talep değerlerinde anlamsız kalabilir. Anlamsız üst sınırlar rassal bir teknik olan genetik algoritma çözümünü olumsuz etkileyecektir. Bu ihtimallin yok edilebilmesi için normal üretim miktarları ve tahmini taleplere bağlı olarak, stratejiler için akılcı üst sınırlar (6.1), (6.2), (6.3) ve (6.4)’de verilen formüllerden elde edilmektedir.

Toplu üretim planlama probleminin çözümü için oluşturulan matematiksel model aşağıda açıklanmıştır:

Modeldeki Değişkenler:

$x_{o(t)}$: ürünün planlama dönemindeki t. ay için mesai üretim miktarı.

$x_{s(t)}$: ürünün planlama dönemindeki t. ay fason üretim miktarı.

$x_{ld(t)}$: ürünün planlama dönemindeki t. ay geç teslim miktarı.

$x_{fs(t)}$: ürünün planlama dönemindeki t. ay bitiş stok miktarı.

$x_{os(t)}$: ürünün planlama dönemindeki t. ay başlangıç stok miktarı.

Modelde on iki aylık tek bir planlama dönemi ele alınmaktadır. Bu nedenle model içerisinde ele alınan ay sayısı ($t : 1,2,3,...,12$ ay indeksi) kadar her bir karar değişkeninden yer almaktadır. Bu değişkenlerin birimi üretilen ürüne göre adet, koli, litre, parça ... vb. olabilir.

Modeldeki Parametreler:

$x_{dem(t)}$: ürünün planlama dönemindeki t. ay için tahmini talep miktarı.

$x_{prod(t)}$: ürünün planlama dönemindeki t. ay için normal üretim miktarı.

$z(t)$: ürünün planlama dönemindeki t. ay için stok miktarı.

($t:1,2,...,12$)

Her ay için tahmini talep değerleri ve normal üretim miktarları problemin başlangıcında bilinmektedir. Bu parametrelerin birimi üretilen ürüne göre adet, koli, litre, parça ... vb. olabilir.

Z : Dönem içinde toplam üretim maliyeti (para birimi).

Problemde, toplam maliyeti minimize edecek $x_{o(t)}$, $x_{s(t)}$, $x_{ld(t)}$, $x_{fs(t)}$ ve $x_{os(t)}$ değerlerinin tespiti amaçlanmaktadır.

x_{ou} : ürünün planlama dönemindeki mesai üretim miktarının üst sınırı.

x_{su} : ürünün planlama dönemindeki fason üretim miktarının üst sınırı.

x_{lu} : ürünün planlama dönemindeki geç teslim miktarının üst sınırı.

x_{fu} : ürünün planlanan en son dönemin bitiş stok miktarının üst sınırı.

$$x_{ou} = [\text{maksimum } x_{dem(t)} - \text{maksimum } x_{prod(t)}] / \infty \quad (6.5)$$

$$x_{su} = [\text{maksimum } x_{dem(t)} - \text{maksimum } x_{prod(t)}] / \infty \quad (6.6)$$

$$x_{lu} = x_{ou} + x_{su} \quad (6.7)$$

$$x_{fu} = x_{os(t-1)} + x_{lu} \quad (6.8)$$

x_{oc} : Bir birim ürünün mesai üretim maliyeti (para birimi).

x_{sc} : Bir birim ürünün fason üretim maliyeti (para birimi).

x_{ldc} : Bir birim ürünün geç teslim maliyeti (para birimi).

x_{fc} : Bir birim ürünün elde bulundurma maliyeti (para birimi).

$P_{xo(t)}$: t. ayda mesai üretimin olup olmama olasılık değeri.

$P_{so(t)}$: t. ayda fason üretimin olup olmama olasılık değeri.

(t: 1,2,...12)

x_1, y_1, y_2 : Rassal üretilecek parametreler için kullanılan bazı ara parametreler.

$$x_1 = x_{oc} + x_{sc} \quad (6.9)$$

$y_1 = (1, x_{ou})$ arasında rassal bir sayı.

$y_2 = (1, x_{su})$ arasında rassal bir sayı.

$fact(t)$: Mesai üretim veya fason üretim olup olmayacağı stratejisinin belirleyen bir parametredir.

$$fact(t) = x_{dem(t)} - x_{prod(t)} - x_{fs(t-1)} \quad (6.10)$$

“fact” değişkeninin pozitif değer aldığı dönemler tahmini taleplerin normal üretim miktarı ve eldeki stok miktarlarıyla karşılanamayacağı anlamına gelmektedir. Rassal üretilen parametrelerin formülleri üzerinde fason üretim olasılığı ve mesai üretim olasılığı pozitif değer alacağı için bu parametrelere rassal olarak üretilen üretim miktarı değerleri verilir. “fact” değişkeninin negatif olması o dönem içerisindeki tahmini taleplerin firma tarafından normal üretim zamanında üretilen miktar ve elde bulundurduğu stoklarla karşılayabileceği anlama gelmektedir. Bu nedenle üretim

hızını değiştirecek mesai üretim veya fason üretim stratejilerine ihtiyaç duyulmayacaktır.

Çalışma için geliştirilen toplu üretim planının genetik algoritma ile çözümü sırasında kullanılan parametreler rassal olarak üretilen parametreler ve türetilen parametreler olmak üzere iki gruptur.

6.2.1. Rassal Olarak Üretilen Parametreler

Mesai ile üretilecek miktar ve fason olarak üretilecek miktar parametreleri rassal olarak aşağıdaki formüllere göre oluşturulur (Onwubolu, 2002).

$$P_{xo(t)} = 1 + \left[\frac{\text{random}(1, x_{ou})}{(\text{fact}_{(t)} * x_{ou})} \right] * \left[\frac{x_{sc}}{x_1} \right] \quad (6.11)$$

$$P_{xs(t)} = 1 + \left[\frac{\text{random}(1, x_{su})}{(\text{fact}_{(t)} * x_{su})} \right] * \left[\frac{x_{oc}}{x_1} \right] \quad (6.12)$$

$$x_{o(t)} = \begin{cases} \text{random}(0, y_1) & \text{eğer } P_{xo(t)} > 1 \\ 0 & \text{eğer } P_{xo(t)} \leq 1 \end{cases} \quad (6.11a)$$

$$x_{s(t)} = \begin{cases} \text{random}(0, y_2) & \text{eğer } P_{xs(t)} > 1 \\ 0 & \text{eğer } P_{xs(t)} \leq 1 \end{cases} \quad (6.12a)$$

6.2.2. Türetilen Parametreler

Geç teslim edilecek miktar, bitiş stok miktarı ve başlangıç stok miktarı gibi parametreler aşağıdaki formüllere göre rassal parametrelerden türetilir (Onwubolu, 2002).

$$Z(t) = X_{os}(t) + X_{prod}(t) + X_o(t) + X_s(t) - X_{dem}(t) \quad (6.13)$$

$$\left. \begin{array}{l} X_{ld}(t) = -Z(t) \\ X_{fs}(t) = 0 \\ X_{os}(t+1) = Z(t) \end{array} \right\} \text{ eğer } Z(t) < 0 \text{ ise} \quad (6.13a)$$

$$\left. \begin{array}{l} X_{ld}(t) = 0 \\ X_{fs}(t) = Z(t) \\ X_{os}(t+1) = Z(t) \end{array} \right\} \text{ eğer } Z(t) \geq 0 \text{ ise} \quad (6.13b)$$

Bu çalışmadaki toplu üretim planlama problemi, bir çeşit ürün grubuna olan toplu talep değişimini orta dönemde karşılamak üzere, fason üretim, mesai üretim stratejileriyle üretim düzeyini değiştirme ve envanter elde bulundurma veya elde bulundurmama maliyetlerini minimize etmeyi amaçlamaktadır.

Toplu üretim planlama problemi bir minimizasyon problemi olduğu için amaç fonksiyonu toplam maliyetin minimize edilmesi şeklinde aşağıdaki gibi formulüze edilmiştir:

$$\text{Min } Z = \left[\sum_{t=1}^{12} X_{o(t)} \right] * (X_{oc}) + \left[\sum_{t=1}^{12} X_{s(t)} \right] * (X_{os}) + \left[\sum_{t=1}^{12} X_{ld(t)} \right] * (X_{ldc}) + \left[\sum_{t=1}^{12} X_{fs(t)} \right] * (X_{fc}) \quad (6.14)$$

Probleme ait herhangi bir toplu üretim planı oluşturma basamakları aşağıdaki gibi özetlenmiştir;

Basamak 1: Fason üretim ve mesai üretim için üst sınırları tahmini talep miktarlarına ve normal üretim miktarlarına göre hesaplanır.

Basamak 2: Eşitlik (6.11) ve (6.12) kullanılarak her dönem için sırasıyla rassal olarak mesai üretim ve fason üretim miktarları hesaplanır.

Basamak 3: Eşitlik (6.13) kullanılarak her dönem için başlangıç stok miktarları, geç teslim miktarları ve bitiş stok miktarları hesaplanır.

6.3. Genetik Algoritma Modelinin Kurulması

Bu bölümde tanımlanan toplu üretim planlama probleminin çözümü için geliştirilen genetik algorithmada kullanılan kodlama, seçim, çaprazlama, mutasyon ve sonlandırma operatörlerinin detaylı açıklaması yapılacaktır.

6.3.1. Parametre Kodlama Operatörü

Genetik algoritmanın rasyonel bir sonuç vermesi için önemli operatörlerden biri kodlama operatörüdür. Kodlama operatörü, toplu üretim planlama problemini genetik kodlara çevirmek için kullanılmaktadır. Problemin her bir çözümü bir kromozom olarak ifade edilmelidir. Bir kromozomun genlerinde problemin karar değişkenleri kodlanmalıdır. Bu aşamada önemli olan oluşturulan kodların kolaylıkla yorumlanabilmesi ve skor fonksiyonunu kolayca ifade edebilmesidir.

Geleneksel genetik algorithmalarda kromozomlar 0 ve 1 değerleriyle ikili sistemde kodlanmaktadır. Ancak bu kodlama yöntemde ardışık bazı değerler tam sayı uzayında komşudur fakat bit-uzayında komşu değildir. Bunun gibi geleneksel ikili bit gösteriminin bazı saklı eğilimlerinin giderilmesi için alternatif kodlama stratejileri geliştirilmiştir. Bu yöntemlerden biri gerçek değer temsildir. Bu yöntemde problem, tam sayı şeklindeki çözümü doğal olarak temsil edilerek ve bu gerçek değerler de genetik algorithmada kullanılmaktadır (**Chipperfield, 1997**). Çaprazlama ve mutasyon operatörleri çözümün gerçek değerlerine uygulanmaktadır.

Çalışmada çözümü aranan toplu üretim planlama probleminde kodlama yöntemi olarak “gerçek değer temsili” kullanılmıştır. **Stockton ve Quinn (1995)**, toplu üretim planlama problemi ile ilgili geliştirdikleri genetik algorithmada ikili kodlama sistemini tercih etmişlerdir. Ancak geliştirdikleri modelde mesai üretim, fason üretim, geç teslim ve son stok miktarları değerlerinin hepsini kromozom içerisindeki gen değerlerine aktarmışlardır. Bu durum çözüm uzayını daha da büyütmüş ve algoritmanın çözüm süresini arttırmaktadır. Bu çalışmada kromozom içerisindeki genlerde yalnızca fason ve mesai üretim değerleri bulunmaktadır. İkili kodlama yönteminin kullanılması tüm kromozomların ilk oluşturulması veya çaprazlama, mutasyon sonrası yeniden oluşturulması durumlarında kromozomun uygunluk değerinin hesaplanması için bit değerlerinin tekrardan tam sayı değerlerine çevrilip formüllerde kullanılması gerekliliğini meydana getirecektir.

Bu şekildeki ilave bir iş yükü parametre değerlerinin gerçek değerlerinin kullanılmasıyla ortadan kaldırılmıştır.

Problemlerin başarılı bir şekilde çözümü için en kısa şifreleme yöntemlerinin kullanılması gerekmektedir. Bu nedenle yalnızca fason üretim miktarları ve mesai üretim miktarları genlere aktarılmıştır. Diğer geç teslim miktarları, başlangıç ve bitiş stok miktarlarının bit içerisine aktarılmasına gerek görülmemiştir çünkü bu değişkenler diğer değişkenlere bağlı olarak hesaplanabilmektedir. Yalnızca fason üretim ve mesai üretim miktarlarının elde edilmesiyle diğer değişkenler kolayca elde edilebilecektir. Bu şekilde bilgisayar hafızasında gereksiz bilgilerle işlem yapılması önlenmiştir. **Stockton ve Quinn (1995)**'nin geliştirdikleri ilk modelde tüm parametre değerleri kromozoma aktarılmaktadır. Ancak **Stockton ve diğ. (2004a)**'nın yaptıkları uygulamada tüm parametrelerin kromozoma aktarılmasının çözüm uzayını gereksiz yere büyüttüğü vurgulanmıştır ve bizim çalışmamızda uygulanan şekilde parametre değerleri genlere aktarılmıştır.

Toplu üretim planlamada fason üretim, mesai üretim, geç teslim ve stok miktarları parametreleri değerleri birbirinden tamamen bağımsız değerler almamaktadır. **Stockton ve Quinn (1995)**'nin araştırmasında bu parametreler hepsinin tamamen rassal değerler alması çözüm uzayının uygulanabilir planların yanında uygulanamayan planları da içermesine neden olacaktır. Bu durumda çözümlerin uygulanabilirliğinin araştırılması gerekmektedir. Ancak yaptığımız çalışmada rassal parametrelere bağlı olarak uygulanabilir planların oluşmasını sağlayan diğer parametreler formüllere göre hesaplanmaktadır. Bu durumda çözüm uzayı uygulanamayan planları içermemektedir.

Gerçek değer kodlama yöntemine göre bir kromozom (dizilim) planlanan periyottaki ay sayısının iki katı uzunluğuna sahip olacaktır. Çalışmadaki problemde bir dizilim 24 tane gen içermektedir.

Toplu üretim planlama probleminde rassal olarak üretilen mesai üretim ve fason üretim miktarları Tablo 6.1'deki gibi kodlanmaktadır.

Tablo 6.1: Üretim Planının Bir Çözümünün Genetik Kodlanması.

$X_0(1)$	$X_S(1)$	$X_0(2)$	$X_S(2)$	$X_0(3)$	$X_S(3)$	$X_0(4)$	$X_S(4)$	$X_0(5)$	$X_S(5)$	$X_0(6)$	$X_S(6)$	$X_0(7)$	$X_S(7)$	$X_0(8)$	$X_S(8)$	$X_0(9)$	$X_S(9)$	$X_0(10)$	$X_S(10)$	$X_0(11)$	$X_S(11)$	$X_0(12)$	$X_S(12)$
Ocak		Şubat		Mart		Nisan		Mayıs		Haziran		Temmuz		Ağustos		Eylül		Ekim		Kasım		Aralık	

Tablo 6.1’de gösterilen bir kromozom toplu üretim planının bir çözümünü göstermektedir.

6.3.2. Başlangıç Popülasyonu Oluşturma

Genetik algoritma yaklaşımında karar verilmesi gerekli diğer parametrelerden birisi de popülasyon büyüklüğüdür. Bu parametre çözüm uzayı büyüklüğünü belirler yani popülasyon içinde kaç adet kromozom yani birey olduğunu gösterir. Bu çalışmadaki genetik algortmada popülasyondaki birey sayısı her nesilde sabit tutulmuştur.

Popülasyon büyüklüğü genetik algoritmanın tüm performansını ve verimliliğini etkilemektedir. Optimum olmayan bir çözüme yakınsama ihtimalini azaltacak ve geçerli sürede uygun çözümü verebilecek bir popülasyon büyüklüğü belirlenmelidir. Çalışmada, 90,100 ve 110 olmak üzere üç farklı popülasyon büyüklüğü kullanılmıştır. **Stockton ve diğ. (2004a)**’nin yaptıkları araştırmada toplu üretim planlama problemleri için popülasyon büyüklüğünü 100 olarak belirlemişlerdir. **Grefenstette (1992)**’ye göre genetik algoritmalarda popülasyon büyüklüğü 10 ile 160 arasında 10’nun katları şeklinde olmalıdır. Literatürdeki bu bilgilere benzer şekilde **Stockton ve diğ. (2004a)**’nin uyguladığı büyüklük, bu büyüklükten daha az veya daha fazlası olması durumlarının da incelenebileceği şekilde popülasyon büyüklükleri seçilmiştir.

Genel olarak ilk popülasyon farklı yollarla oluşturabilmektedir. Genetik algoritmanın birçok uygulamalarında rassal olarak oluşturulmaktadır. İlk popülasyonun farklı çözüm kümelerinden oluşturulması daha etkili araştırmayı sağlamaktadır (**Onwubolu, 2002**). Bu çalışmada ilk popülasyon rassal olarak (6.11), (6.12) ve (6.13) eşitliklerine bağlı olarak oluşturulmuştur.

6.3.3. Seçim (üreme) Operatörü

Genetik algoritmanın yürütülebilmesinde düşünülmesi gerekli diğer bir faktör seçim olasılığıdır. Genel tanıma göre bir bireyin seçim olasılığı, popülasyonda bireyin performans ölçümüne göre belirlenmelidir (**Onwubolu, 2002**).

Toplu üretim planlama probleminin amaç fonksiyonu alternatif planların toplam maliyetini minimize eder. Bu nedenle popülasyondaki her bir bireyin uygunluk değeri bireyin toplam maliyeti ile ilişkilidir. Her bireyin uygunluk değeri bu bir minimizasyon problemi olduğu için (6.15)’de gösterildiği şekilde hesaplanır.

$$f(k) = \begin{cases} f_{\text{mak}} - g(k) & \text{eğer } g(k) < f_{\text{maks}} \\ 0 & \text{eğer } g(k) \geq f_{\text{maks}} \end{cases} \quad (6.15)$$

Bu formülde $g(k)$, k . kromozomun amaç fonksiyonu değeridir; $f(k)$, k . kromozomun uygunluk değeridir; f_{mak} popülasyon içerisinde en yüksek amaç fonksiyonu değeridir; k , kromozomları ifade eden bir indekstir ($k: 1,2,3,\dots,\text{popülasyon büyüklüğü}$).

Çözüm uzayı içerisindeki bireyler uygunluk değerlerine göre kıyaslanmaktadır. Yukarıdaki formüle göre en düşük maliyete sahip çözüm en yüksek uygunluk değerine sahip olmaktadır. Yüksek uygunluk değerine sahip bireylerin seçim olasılığı daha yüksektir.

Uygunluk değerinin dışında kıyaslama parametrelerinden biri bireyin beklenen değeridir. Bir bireyin beklenen değeri aşağıdaki formülle hesaplanır:

$$\text{Beklenen değer } [e(k)] = \frac{\text{k. kromozomun uygunluk değeri } [f(k)]}{\text{tüm kromozomların uygunluk değerlerinin ortalaması}} \quad (6.16)$$

Mevcut problemde seçim yöntemlerinden biri olarak, **Stockton ve Quinn (1995)**, **Stockton ve diğ. (2004a)**'nin araştırmalarında kullandıkları ve literatürde tüm problem alanları için en çok tercih edilen seçim yöntemlerinden biri olan rulet yöntemi tercih edilmiştir. **Sakawa (2002)**'ye bağlı olarak rulet yönteminin stokastik hatalarının azaltılması için önerilen diğer bir seçim yöntemi olan beklenen değer yöntemi ikinci seçim yöntemi olarak belirlenmiştir.

Her iki seçim yönteminde de bazı durumlarda popülasyondaki en iyi çözüm bir sonraki nesle aktarılamayabilmektedir. Bu olayın meydana gelmemesi için elitizm yöntemi seçim yöntemlerine ilave edilmiştir. Yani her iki seçim yöntemine ilave olarak programda elitizm yöntemi de uygulanabilmektedir. Bu şekilde seçim yöntemlerinin performansının artırılması umulmuştur.

6.3.4. Çaprazlama Operatörü

Çaprazlama operatörü genetik algoritmanın temel işlemcisidir. Farklı iki kromozomların genlerini değiştirerek daha iyi veya daha kötü çözümler yaratmaktadır. Genetik algoritmalarda iki çok kötü kromozom çaprazlanarak daha iyi kromozomlar elde edilebilir. Bu nedenle optimuma yakın çözümlerin elde edilmesinde çaprazlama operatörü önemli bir rol oynamaktadır.

Bu operatörün önemli bir bileşeni çaprazlama olasılığıdır. Çaprazlama olasılığı popülasyondaki kaç tane kromozomun çaprazlanacağı gösteren bir parametredir. **Grefenstette (1992)**'ye göre genetik algoritma ile ilgili yapılan çalışmalarda çaprazlama olasılığı 0,25 ile 1 arasında 0,05'lik artışlar şeklinde değişmektedir. Çaprazlama işlemi stokastik bir yapıya sahiptir. Bu nedenle çaprazlama olasılığındaki küçük artışlar çözüm üzerinde bu artış değerinden çok daha önemsiz bir etkiye sahip olacaktır. Bu etkinin belirgin olabilmesi için 0,05'lik artışlar olması gerekmektedir (**Stockton ve diğ., 2004b**). **Biethahn ve Nissen (1995)**'e göre çaprazlama olasılığının 0,6'dan büyük olması tavsiye edilmektedir. **Stockton ve Quinn (1995)** ve **Stockton ve diğ. (2004a)**'nin geliştirdikleri genetik algorithmada çaprazlama olasılığı olarak 0,6 - 0,65 - 0,7 ve 0,75 değerleri kullanılmıştır. Bu değerlerin literatürdeki diğer değerlere uygun olması nedeniyle ve değerler 0-1 aralığında aralık daraltmasına uygun olduğu için bu araştırmada kullanılmak üzere seçilmiştir.

Çaprazlama yöntemi olarak en basit tek nokta çaprazlama yöntemi bulunmaktadır. Bu yöntem etkin olmasına karşın çözümleri geliştirmek için daha iyi yöntemler araştırılmalıdır. Bilgisayar testleri çok noktalı çaprazlama yöntemlerinin tek noktalıdan daha iyi sonuçlar verdiği ispatlanmıştır (**Onwubolu, 2002**). Çalışmada oluşturulan kromozomların dizi uzunluğu 24'tür. Bu dizi uzunluğuna da bağlı olarak iki noktalı çaprazlama yöntemi tercih edilmiştir. **Stockton ve Quinn (1995)**'nin ilk araştırmasında tek noktalı çaprazlama yöntemi kullanılırken **Stockton ve diğ. (2004a)**'nin geliştirdikleri genetik algorithmada da iki noktalı çaprazlama yöntemi kullanılmıştır.

Bu yöntemde rasgele seçilen iki farklı çaprazlama noktası arasında kalan gen değerleri karşılıklı olarak yer değiştirmektedir. Bu yöntem Şekil 6.1'de açıklanmaktadır.

Seçilen İki Kromozom

1. çaprazlama noktası						2. çaprazlama noktası																	
X _o (1)	X _s (1)	X _o (2)	X _s (2)	X _o (3)	X _s (3)	X _o (4)	X _s (4)	X _o (5)	X _s (5)	X _o (6)	X _s (6)	X _o (7)	X _s (7)	X _o (8)	X _s (8)	X _o (9)	X _s (9)	X _o (10)	X _s (10)	X _o (11)	X _s (11)	X _o (12)	X _s (12)
Y _o (1)	Y _s (1)	Y _o (2)	Y _s (2)	Y _o (3)	Y _s (3)	Y _o (4)	Y _s (4)	Y _o (5)	Y _s (5)	Y _o (6)	Y _s (6)	Y _o (7)	Y _s (7)	Y _o (8)	Y _s (8)	Y _o (9)	Y _s (9)	Y _o (10)	Y _s (10)	Y _o (11)	Y _s (11)	Y _o (12)	Y _s (12)

Çaprazlama Sonucu Oluşan Yeni Kromozomlar

X _o (1)	X _s (1)	X _o (2)	X _s (2)	X _o (3)	X _s (3)	Y _o (4)	Y _s (4)	Y _o (5)	Y _s (5)	Y _o (6)	Y _s (6)	Y _o (7)	Y _s (7)	Y _o (8)	Y _s (8)	X _o (9)	X _s (9)	X _o (10)	X _s (10)	X _o (11)	X _s (11)	X _o (12)	X _s (12)
Y _o (1)	Y _s (1)	Y _o (2)	Y _s (2)	Y _o (3)	Y _s (3)	X _o (4)	X _s (4)	X _o (5)	X _s (5)	X _o (6)	X _s (6)	X _o (7)	X _s (7)	X _o (8)	X _s (8)	Y _o (9)	Y _s (9)	Y _o (10)	Y _s (10)	Y _o (11)	Y _s (11)	Y _o (12)	Y _s (12)

Şekil 6.1: İki Nokta Çaprazlama Yöntemi.

6.3.5. Mutasyon Operatörü

Mutasyon, genetik algoritmaların önemli diğer işlemcisidir. Genetik algoritmalarda mutasyon işlemi, kromozomlardaki gen değerlerini rasgele değiştirerek popülasyon içinde yeni bireylerin oluşması sağlanır. Bu işlemin uygulama amacı temelde algoritmanın tıkanmasını önlemektir.

Mutasyon olasılığı ise işlemde değişikliğe uğrayacak birey sayısını belirtir. Mutasyon olmaz ise yeni birey çaprazlama veya kopyalama sonrasında olduğu gibi kalır.

Mutasyon oranı genel değerleri (0,01 – 0,001) aralığındadır (**Biethahn ve Nissen, 1995**). **Stockton ve diğ. (2004a)**'nin geliştirdikleri genetik algoritmada mutasyon olasılığı olarak 0,0001 – 0,001 ve 0,005 değerleri kullanılmıştır. Bu değerlerin literatürdeki diğer değerlere uygun olması nedeniyle ve değerler 0-1 aralığında aralık daraltmasına uygun olduğu için bu araştırmada kullanılmak üzere seçilmiştir.

Mutasyon yöntemi olarak karşılıklı değişim yöntemi tercih edilmiştir. Seçilen iki kromozomdan rasgele seçilen iki gen karşılıklı değiştirilmektedir. Bu yöntem Şekil 6.2.'de açıklanmaktadır.

Seçilen İki Kromozom

X _o (1)	X _s (1)	X _o (2)	X _s (2)	X _o (3)	X _s (3)	X _o (4)	X _s (4)	X _o (5)	X _s (5)	X _o (6)	X _s (6)	X _o (7)	X _s (7)	X _o (8)	X _s (8)	X _o (9)	X _s (9)	X _o (10)	X _s (10)	X _o (11)	X _s (11)	X _o (12)	X _s (12)
Y _o (1)	Y _s (1)	Y _o (2)	Y _s (2)	Y _o (3)	Y _s (3)	Y _o (4)	Y _s (4)	Y _o (5)	Y _s (5)	Y _o (6)	Y _s (6)	Y _o (7)	Y _s (7)	Y _o (8)	Y _s (8)	Y _o (9)	Y _s (9)	Y _o (10)	Y _s (10)	Y _o (11)	Y _s (11)	Y _o (12)	Y _s (12)

Mutasyon Sonucu Oluşan Yeni Kromozomlar

X _o (1)	X _s (1)	X _o (2)	X _s (2)	X _o (3)	X _s (3)	Y _o (4)	X _s (4)	X _o (5)	X _s (5)	X _o (6)	X _s (6)	X _o (7)	X _s (7)	X _o (8)	X _s (8)	X _o (9)	X _s (9)	X _o (10)	X _s (10)	X _o (11)	X _s (11)	X _o (12)	X _s (12)
Y _o (1)	Y _s (1)	Y _o (2)	Y _s (2)	Y _o (3)	Y _s (3)	Y _o (4)	Y _s (4)	Y _o (5)	Y _s (5)	Y _o (6)	Y _s (6)	Y _o (7)	Y _s (7)	Y _o (8)	X _s (8)	Y _o (9)	Y _s (9)	Y _o (10)	Y _s (10)	Y _o (11)	Y _s (11)	Y _o (12)	Y _s (12)

Şekil 6.2: Karşılıklı Değişim Mutasyon Yöntemi.

6.3.6. Sonlandırma Operatörü

Genetik algoritma araştırmaya devam mı edeceği veya araştırmayı durduracak mı kararını sonlandırma kriterine göre belirler. Sonlandırma ile ilgili farklı yöntemler bulunmaktadır. Bu çalışmada sonlandırma kriteri olarak literatürdeki birçok çalışmada kullanılan nesil sayısı yöntemi tercih edilmiştir.

Toplu üretim planlama problemi için geliştirilen genetik algoritma için 400-500 arasında bir nesil oluşturulduktan sonra algoritmanın sonlandırılmasının uygun olduğu **Onwubolu (2002)** tarafından yapılan araştırmada bulunmuştur. Bu araştırmaya göre 500 nesille kadar toplam maliyetin düşürüldüğü daha yüksek nesil sayısında elde edilen çözümlerin maliyet değerlerinin çok farklı olmadığı saptanmıştır. Bu nedenle bu çalışmada sonlandırma işlemi 500 nesile ulaşıldığında gerçekleşmektedir.

6.4. Bilgisayar Programının Hazırlanması

Optimizasyon problemlerinin çözümünde genetik algoritma yaklaşımı genellikle bilgisayar ortamında çözümler sunabilmektedir. Bu yöntem NP-zor problemlerin çözümünde kullanıldığı için genellikle bilgisayar ortamı dışında çözüm sunamamaktadır. Literatürde, genetik algortimaların çözümü için GENESIS, GOAL, GALIP245, GALOPP32 ve GENOCOP gibi bilgisayar programlarına rastlanmıştır. Bunlar gibi genetik algoritmaları çözen hazır programların birçoğu yüksek maliyetlerde elde edilebilmektedir. Ayrıca GENESIS ve GOAL gibi birçok programda genetik algoritmanın çalışabilmesi için problemin mutlaka matematiksel modellere dönüştürülmesi, kısıtlarının ve amaç fonksiyonlarının matematiksel programlama tekniklerine uygun olarak modellenmesi gerekmektedir. Ayrıca bu programlar genellikle yalnızca bir değişkenin kodlandığı problemlere uygundur ve kodlama yöntemi olarak ikilik sistemi kullanmaktadır.

Literatürde birçok araştırmacı kendi problem alanına yönelik C++ ve Fortran yazılım dillerini kullanarak geliştirdikleri genetik algoritmaları çözmüşlerdir. Ancak bu yazılım dilleri günümüzde en gelişmiş yazılım dilleri arasında bulunan .Net'e göre yazılım olanakları daha kısıtlı dillerdir.

Bu çalışmada toplu üretim planlama problemi için software olarak .Net yazılım dili kullanılarak bir GATÜP (**Genetik Algoritma ile Toplu Üretim Planlama**) bilgisayar programı geliştirilmiştir. Hardware olarak .Net konusunda bilgili programcının sahip olduğu donanımlar yeterli olmuştur. Programın geliştirilmesi sürecinde .Net konusunda bilgili bilgisayar öğretmeni Yusuf BAYRAMOĞLU'ndan yardım alınmıştır. GATÜP programının akış şeması Ek A'da ve programın nasıl kullanıldığı EK B'de sunulmaktadır.

.Net yazılımının tercih edilme nedenleri aşağıda sıralanmıştır (**Bayramoğlu, 2005**):

- .NET Framework geliştiricilerin her türlü programlama dilini kullanabilmelerini ve herhangi bir programlama dilinde yazılmış olan uygulamaların birbirleriyle derin bir entegrasyon kurabilmesini sağlar, bu ise mevcut geliştirme becerilerinin yeniden eğitime gerek kalmaksızın geliştirilebilmesi imkanını tanır.
- .NET Framework, geliştiricilerin iş programları yazmanın mantığına odaklanmalarını sağlayan yüksek seviyede bileşenleştirilmiş, detaylardan arındırılmış bir tasarım kullanır. Bu nedenle daha az kodla bir program geliştirilebilir. Kısa sürede kullanıma hazır hale getirir, uygulama kurulumu ve yönetimi kolaydır ve çok daha gelişmiş bir performans sağlar.
- .NET Framework uygulamaları daha güvenilir hale getirmeye yönelik teknolojileri içermektedir. Örneğin bellek sızıntılarını önlemek için bellek, iş parçacıkları ve işlemler .NET Framework tarafından yönetilir. Ayrıca, .NET çalışan Web uygulamalarını denetler ve yöneticinin belirleyeceği aralıklarda bu uygulamaları yeniden başlatır.
- .NET Framework, tipik Web uygulamalarının performansını artırır.

GATÜP programında genetik algoritma prosedürlerinin uygulanabilmesi için kullanılan .NET kodları EK C'de tanımlanmıştır.

6.5. GATÜP Programı Problem Çözüm Performansının Değerlendirilmesi

GATÜP programının toplu üretim planı oluşturma performansının değerlendirilmesi için **Stockton ve Quinn (1995)**'nin, **Stockton ve diğ. (2004a)**'nin ve **Stockton ve diğ. (2004b)**'nin kullandıkları problem değerleri tercih edilmiştir. Bu değerler Tablo 6.2'de gösterilmektedir.

Bu problemin değerlerinin hangi kriterlere bağlı olarak seçildiği aşağıda belirtilmiştir;

- (a) planlanan periyot süresince çok değişken satış talepleri içermektedir.
- (b) bitişik dönemlerde satış taleplerinde ani olarak büyük değişimler oluşmaktadır.
- (c) planlanan periyot içerisinde tahmini talep miktarının normal üretim kapasitesinin üzerinde olduğu veya normal üretim kapasitesinin tahmini talep miktarının üzerinde olduğu tüm durumlar mevcuttur.
- (d) tahmini talep ve normal üretim kapasitesi arasında çok büyük farklılıklar oluşmaktadır.

Tablo 6.2: Toplu Üretim Planlama Problem Değerleri.

Planlanan Dönem	Tahmini Talep Değerleri (adet)	Normal Üretim Kapasitesi (adet)	Maliyetler [(£)/adet]
Ocak	1400	1000	Mesai üretim maliyeti : 10
Şubat	900	1000	Fason üretim maliyeti : 50
Mart	800	1000	Geç teslim maliyeti : 200
Nisan	1500	1000	Stoklama maliyeti : 100
Mayıs	1500	1000	
Haziran	800	800	
Temmuz	500	800	
Ağustos	1200	800	
Eylül	600	1000	
Ekim	2000	1000	
Kasım	500	1000	
Aralık	1500	900	

Stockton ve diğ. (2004a)'nın yaptığı çalışmadaki genetik algoritma operatörlerine uygun olarak GATÜP programı on kez çalıştırıldığında elde edilen minimum maliyet değeri Şekil 6.3'de verilmiştir. GATÜP programının elde ettiği minimum toplam

maliyet değeri £213130'dir. **Stockton ve diğ. (2004a)**'nın çalışmasında elde edilen minimum maliyet değeri £414900. **Stockton ve Quinn (1995)**'in çalışmasında ilk on bir ayın stok maliyeti olmaksızın elde ettikleri değer £58900'dür. GATÜP programının maliyet hesaplamasına benzer hale getirmek için ilk on bir ayın stok değeri ilave edilirse toplam maliyet değeri £550900'e yükselmektedir.

GATÜP programı literatürdeki bu çalışmalardan oldukça düşük toplam maliyetli bir toplu üretim planı çözüm olarak sunabilmiştir. Ancak burada şu nokta önemlidir; literatürdeki bu çalışmalarda katı yönetim hedefleri doğrultusunda fason üretim için 300 birim ve mesai üretim için 90 birim olarak sabit üst sınır değerleri belirlenmiştir. GATÜP programı bu sınır değerleri rassal olarak tahmini talep ve normal üretim miktarlarına bağlı olarak hesaplamaktadır. Buna rağmen fason üretim üst sınırı literatürdeki değerlere uygundur ancak mesai üretim için üst sınır 90 birim'in üzerindedir. Bu çözüm değerlerine göre yönetimin hedeflerinin toplu üretim planları çözümlerini nasıl etkilediği görülmektedir. Daha esnek yönetim hedefleri ile toplu üretim planı çözümlerinde optimuma daha yakın sonuçlar elde etmek mümkün olabilmektedir.

Genetik Algoritma ile Toplu Üretim Planlama

YARDIM

Çaprazlama Olasılığı: 0,65 Başlangıç Stok Miktarı: 0

Mutasyon Olasılığı: 0,005 Fason Üretim Birim Maliyeti: 50

Popülasyon Büyüklüğü: 100 Birim Başına Mesai Maliyeti: 10

Seçim Yöntemi: Rulet Yöntemi Geç Teslim Birim Maliyeti: 200

Elitizm Yöntemini Uygula ☒ Birim Stok Maliyeti: 100

	Üretim Miktarı	Talep Tahmini	MESAI	FASON ÜRETİM	GEÇ TESLİM	BAŞLANGIÇ STOK	SON STOK
OCAK	1000	1400	342	9	49	0	0
ŞUBAT	1000	900	0	0	0	-49	51
MART	1000	900	0	0	0	51	251
NISAN	1000	1500	180	95	0	251	26
MAYIS	1000	1500	372	78	24	26	0
HAZİRAN	800	800	20	0	4	-24	0
TEMMUZ	800	500	0	0	0	-4	296
AĞUSTOS	800	1200	66	66	0	296	28
EYLÜL	1000	600	0	0	0	28	428
EKİM	1000	2000	451	20	91	428	0
KASIM	1000	500	0	0	0	-91	409
ARALIK	900	1500	172	20	0	409	1

TOPLAM MALİYET 213130

HESAPLA

Şekil 6.3: GATÜP Programının Hesapladığı Minimum Maliyetli Toplu Üretim Planı.

Onwubolu (2002)'nin yaptığı çalışmada benzer toplu üretim planı probleminin genetik algoritma ile çözümü sonucunda minimum toplam maliyet £296010'dur. GATÜP programından elde edilen toplam maliyet değeri daha azdır. Ayrıca fason üretim ve mesai üretim için **Onwubolu (2002)** tarafından belirlenen 500'er birim üst sınır değerlerini GATÜP programının çözüm olarak sunduğu toplu üretim planı sağlamaktadır.

6.5.1. Genetik Algoritma Operatörlerinin Çözüm Performansına Etkisi

Toplu üretim planlama probleminde optimum veya optimuma yakın üretim planının elde edilebilmesi için genetik algoritma yaklaşımında kullanılan operatörlerin değerleri önemlidir. Çalışmanın bu bölümünde farklı operatör değerlerinin GATÜP programının çözüm performansını nasıl etkilediği araştırılacaktır. Bu araştırma sırasında Tablo 6.2.'de sunulan toplu üretim planı problemi üzerinde çalışılmıştır. Toplu üretim planlama problemlerinin rassal araştırma tekniklerinden biri olan genetik algoritma yaklaşımı ile çözümünde rassallıktan kaynaklanabilecek sapmaları engellemek amacıyla her incelenen parametre değerleri için GATÜP programı 10 kez çözülerek ortalama değerlerde de verilmiştir.

I - GATÜP programında kullanıcılara rulet yöntemi ve beklenen değer olmak üzere iki farklı seçim yöntemi sunulmaktadır. Bu seçim yöntemlerine kullanıcı isterse elitizm yöntemini ilave edebilmektedir. Bu seçim yöntemlerinin etkinliğini belirleyebilmek için **Stockton ve diğ. (2004b)**'nin çalışmasındakine benzer olarak genetik algoritma parametrelerinden çaprazlama olasılığı 0,75 ve mutasyon olasılığı 0,001 olarak sabit kabul edilmiştir. Seçim yöntemlerinde kromozomların seçim şansı, bulundukları popülasyondaki diğer kromozom değerlerine bağlı olarak oluşmaktadır. Bu nedenle seçim yöntemlerin popülasyon büyüklüğü değiştiğinde etkilerinin nasıl değişeceğinin belirlenebilmesi için popülasyon büyüklüğü sabit alınmamıştır.

Toplu üretim planlama problemi tüm seçim yöntemlerine ve popülasyon büyüklüklerine bağlı durumlar için 10 kez çözülmüştür ve her durum için 10 çözümden minimum değere sahip toplu üretim planı maliyet değerleri ve ortalama maliyet değerleri Tablo 6.3.'de sunulmuştur. Dört farklı seçim yöntemi, üç farklı popülasyon büyüklüğü için denendiğinden ve her problem 10 kez çözüldüğünden toplam 120 kez GATÜP programı ile deneme yapmak gerekmiştir.

Tablo 6.3. incelendiğinde popülasyon büyüklüğü 90 ve 110 için elitizm yöntemiyle birlikte rulet seçiminin uygulanmasıyla en düşük maliyetli toplu üretim planı elde edilmiştir ve popülasyon büyüklüğü 100 için elitizm yöntemiyle birlikte beklenen değer seçiminin uygulanmasıyla en düşük maliyetli toplu üretim planı elde edilmiştir. Tüm popülasyon büyüklükleri ve seçim yöntemleri için elde edilen ortalama maliyet değerleri de bu sonuçlarla benzerdir.

Tablo 6.3: Seçim Yöntemine Göre Elde Edilen Minimum ve Ortalama Toplam Maliyet Değerleri.

Seçim Yöntemi	Toplam Maliyet (£)					
	Popülasyon: 90		Popülasyon:100		Popülasyon:110	
	minimum	ortalama	minimum	ortalama	minimum	ortalama
Rulet Seçim	279760	294075	272240	279860	274310	279508
Rulet ve Elitizm	219000	231590	232630	247623	229930	228626
Beklenen Değer Seçimi	331030	333333	348430	345253	277810	325928
Beklenen Değer ve Elitizm	231370	247498	221690	232620	231530	233405

Bu sonuçlara göre seçim yöntemi ne olursa olsun elitizm yöntemi ile birlikte kullanılması genetik algoritmanın çözüm performansını artırmaktadır. **Stockton ve diğ. (2004b)**'nin yaptığı araştırmada da elitizm yöntemiyle birlikte uygulanan rulet yöntemi yalnızca rulet yönteminin uygulanmasından daha iyi çözümler sunmuştur.

Genel olarak rulet seçim yöntemi kullanılarak elde edilen toplu üretim planları elitizm yöntemi uygulansın uygulanmasın beklenen değer seçim yöntemi kullanılarak elde edilen toplu üretim planlarından daha az maliyetli olmaktadır. Bu sonuç yalnızca popülasyon büyüklüğü 100 olduğunda elitizm yöntemiyle birlikte beklenen değer yönteminin uygulanması durumunda değişmiştir.

II - GATÜP programı içerisinde genetik algoritma çözümünde iki noktalı çaprazlama yöntemi kullanılmaktadır. Ancak program kullanıcılara 0,6, 0,65, 0,7 ve 0,75 olmak üzere dört farklı çaprazlama olasılığı kullanabilme olanağı sunmaktadır. Bu olasılık değerlerine göre popülasyon içerisinde kaç tane kromozomun çaprazlamaya uğrayacağı belirlenmektedir. Bu çaprazlama olasılıklarının çözüm üzerindeki

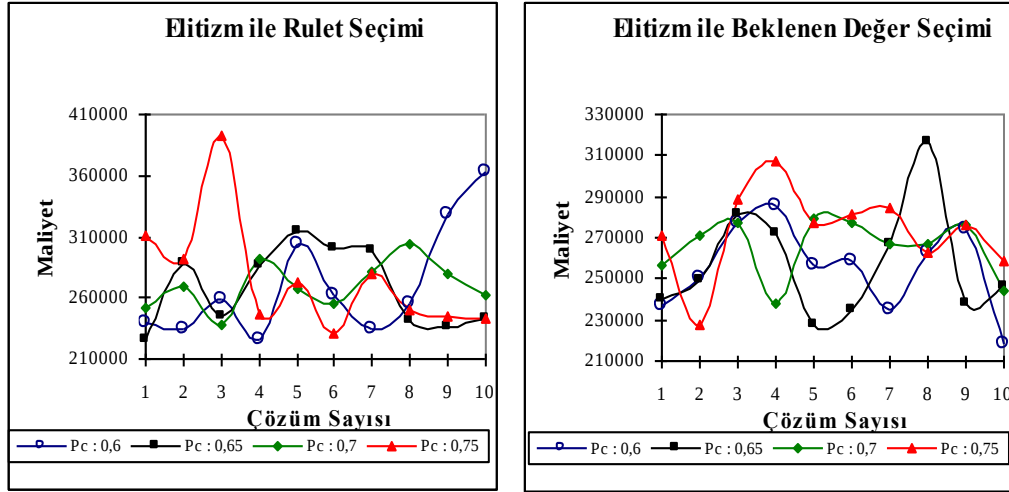
etkilerinin belirlenebilmesi için **Stockton ve diğ. (2004b)**'nin yaptıkları çalışmaya benzer olarak genetik algoritma popülasyon büyüklüğü 100 ve mutasyon olasılığı 0,001 olarak sabit kabul edilmiştir. Seçim yöntemi olarak elitizm yöntemiyle birlikte rulet seçim yöntemi ve beklenen değer yöntemi kullanılmıştır. Elitizm uygulanmadan kullanılan seçim yöntemlerinin etkin olmadığı görüldüğü için seçim yöntemlerinin basit hallerinde çaprazlama olasılığının etkisi ilgili bir araştırma yapılmamıştır.

Toplu üretim planlama problemi, her seçim yönteminde tüm çaprazlama olasılıkları için 10 kez çözülmüştür ve her durum için 10 çözümden minimum değere sahip toplu üretim planı maliyet değerleri ve ortalama maliyet değerleri Tablo 6.4.'de sunulmuştur. Bu değerler problemin toplam 80 kez GATÜP programı ile çözülmesiyle elde edilmiştir.

Tablo 6.4: Çaprazlama Olasılığına Göre Elde Edilen Minimum ve Ortalama Toplam Maliyet Değerleri.

Seçim Yöntemi	Toplam Maliyet (£)							
	$P_c = 0,6$		$P_c = 0,65$		$P_c = 0,7$		$P_c = 0,75$	
	minimum	ortalama	minimum	ortalama	minimum	ortalama	minimum	ortalama
Rulet ve Elitizm	226020	270761	225340	267931	237910	269879	230180	276275
Beklenen Değer ve Elitizm	218290	255482	227470	257230	238290	265508	227870	273589

Tüm denemelerin sonuçları Şekil 6.4.'deki grafiklerde gösterilmektedir.



Şekil 6.4: Çaprazlama Olasılığının Etkisi.

Stockton ve diğ. (2004b)'nin yaptıkları çalışmada rulet ve elitizm yönteminde 0,65 çaprazlama olasılığı en düşük maliyetli toplu üretim planını vermiştir. Yaptığımız çalışmada da elitizm ile birlikte rulet seçim yöntemini uyguladığımızda 0,65 çaprazlama olasılığı en düşük maliyetli toplu üretim planını vermiştir.

Elitizm ile birlikte beklenen değer yöntemi uyguladığımızda ise 0,6 çaprazlama olasılığı en düşük maliyetli toplu üretim planını vermiştir.

Çaprazlama operatörü genetik algoritmanın optimum veya optimuma yakın çözümlere ulaşabilmesi için araştırma uzayının farklı alanlarına ulaşmasını sağlar. Bu nedenle çaprazlama olasılığının büyük olması algoritmanın araştırma yapabileceği alanı genişletmektedir. Ancak yapılan araştırma sonucunda çaprazlama olasılığının düşük değerlerinde minimum maliyetli toplu üretim planları elde edilmiştir. Bu durum problemin yapısından kaynaklanabilmektedir. Daha karmaşık problemlerde daha yüksek çaprazlama olasılıkları gerekli olabilir.

III - GATÜP programı içerisinde genetik algoritma çözümünde mutasyon yöntemi olarak karşılık değiştirme yöntemi kullanılmaktadır. Programda kullanıcı 0,0001, 0,001 ve 0,005 olmak üzere üç farklı mutasyon olasılıklarından birini seçebilmektedir. Bu olasılık değerlerine göre popülasyon içerisinde kaç tane kromozomun mutasyona uğrayacağı belirlenmektedir. Bu mutasyon olasılıklarının çözüm üzerindeki etkilerinin belirlenebilmesi için **Stockton ve diğ. (2004b)**'nin yaptıkları çalışmaya benzer olarak genetik algoritma popülasyon büyüklüğü 100 ve çaprazlama olasılığı 0,60 olarak sabit kabul edilmiştir. Seçim yöntemi olarak elitizm yöntemiyle birlikte rulet seçim yöntemi ve beklenen değer yöntemi kullanılmıştır.

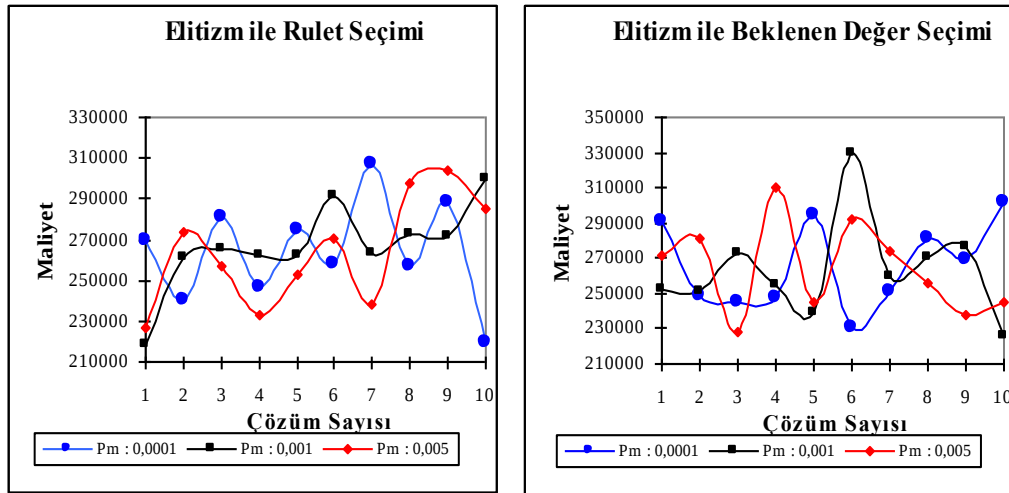
Elitizm uygulanmadan kullanılan seçim yöntemlerinin etkin olmadığı görüldüğü için seçim yöntemlerinin basit hallerinde mutasyon olasılığının etkisi ilgili bir araştırma yapılmamıştır.

Toplu üretim planlama problemi, her seçim yönteminde tüm mutasyon olasılıkları için 10 kez çözülmüştür ve her durum için 10 çözümden minimum değere sahip toplu üretim planı maliyet değerleri ve ortalama maliyet değerleri Tablo 6.5.'de sunulmuştur. Bu değerler problemin toplam 60 kez GATÜP programı ile çözülmesiyle elde edilmiştir.

Tablo 6.5: Mutasyon Olasılığına Göre Elde Edilen Minimum ve Ortalama Toplam Maliyet Değerleri.

Seçim Yöntemi	Toplam Maliyet (£)					
	$P_m = 0,0001$		$P_m = 0,001$		$P_m = 0,005$	
	minimum	ortalama	minimum	ortalama	minimum	ortalama
Rulet ve Elitizm	219010	264211	218810	266735	226960	263777
Beklenen Değer ve Elitizm	230010	265786	225800	263227	228510	264308

Tüm denemelerin sonuçları Şekil 6.5.'deki grafiklerde gösterilmektedir.



Şekil 6.5: Mutasyon Olasılığının Etkisi.

Stockton ve diğ. (2004b)'nin yaptıkları çalışmada rulet ve elitizm yönteminde 0,005 mutasyon olasılığı en düşük maliyetli toplu üretim planını vermiştir. Yaptığımız çalışmada her iki seçim yöntemini uyguladığımızda da 0,001 mutasyon olasılığı en

düşük maliyetli toplu üretim planını vermiştir. Bu sonuç, **Stockton ve diğ. (2004b)**'nin çalışmasından farklıdır. Elde edilen ortalama maliyet değerleri incelendiğinde rulet ve elitizm yönteminde minimum maliyet **Stockton ve diğ. (2004b)**'nin yaptığı çalışma ile aynı sonucu vererek 0,005 mutasyon olasılığında en düşük maliyet değerini vermiştir. Minimum maliyet değeri ile ortalama maliyet değeri arasında farklılığın oluşması genetik algoritmanın rassallığından kaynaklanmaktadır.

Yüksek mutasyon olasılıkları genetik algoritmanın rassal araştırma tekniğinin prosedürlerini bozabilmektedir. Buna karşın düşük mutasyon olasılıkları ise popülasyon içerisinde birbirine benzer bireylerin oluşarak çözümün optimum olmayan bir noktaya yakınsaması problemini önleyemeyebilmektedir. Yaptığımız çalışmada da 0,0001 gibi düşük bir mutasyon olasılığı veya 0,005 gibi yüksek bir mutasyon olasılığı toplu üretim planı problemi için en düşük maliyetli sonucu verememiştir.

GATÜP programının parametre değerlerinin literatürdeki değerlerle uyumlu olması geliştirilen algoritmanın geçerli olduğunun da bir göstergesidir.

6.6. GATÜP Programının Bir Şirkette Uygulaması

Daha öncede belirtildiği gibi bu çalışmada amaç genetik algoritmanın toplu üretim planlama problemlerinin çözümünde kullanımının incelenmesidir. Bu amaçla geliştirilen ve literatürdeki uygulamalardan yararlanılarak çalışma performansı uygun bulunan GATÜP programının sektörde uygulanabilirliğinin araştırılmasının genetik algoritmaların pratikte uygulanabilirliğinin belirlenebilmesine bir ışık tutacağı düşünülmektedir. Bu bölümde, çalışma için geliştirilen GATÜP programı Türkiye'nin büyük ölçekli sanayi kuruluşlarından biri olan ve elektronik sektöründe faaliyet gösteren Beko Elektronik Anonim Şirketi'nde uygulanmıştır.

6.6.1. Şirket Tanıtımı

Bu bölümde GATÜP programının uygulanabilirliğinin araştırılması için verilerin elde edildiği şirket hakkında kısaca bilgi verilmektedir.

22 Eylül 1966'da kurulan Beko Elektronik A.Ş. bugün Türkiye'nin ve Avrupa'nın tüketici elektroniği konusunda faaliyet gösteren en büyük şirketlerinden biridir.

180 bin m² arazi üzerinde kurulu bulunan 151 bin m² kapalı alana sahip modern tesislerinde, yurtiçi ve yurtdışı pazarlar için çeşitli tip ve modellerde televizyon, LCD TV, Plazma TV, dijital TV (iDTV), yazarkasa, uydu alıcı, set top box (STB) cihazları ve PC üretmekte; müzik seti, DVD oynatıcı/kaydedici, ev sinema sistemi ve diğer tüketici elektroniği ürünlerini pazarlayarak satış sonrası servis sağlamaktadır.

Bir Koç Topluluğu şirketi olan Beko Elektronik, 3750 çalışanı, 2004 yılında gerçekleştirdiği 1 milyar 124 milyon EURO cirosu ile Türkiye'nin en büyük özel sanayi kuruluşlarından biridir.

Uluslararası alanda güçlü bir marka ile yer alma hedefi ile Grundig 29 Ocak 2004'te İngiliz Alba Plc. fiması ile %50 ortaklıkla satın alınmıştır. Endüstriyel ve Elektronik tasarım alanında katma değer yaratan Beko Elektronik, Grundig'in alınmasıyla da Avrupa'da satış ve dağıtım kanallarına ve global bir markaya sahip olmak gibi iki yeni katma değer yaratmıştır.

Beko Elektronik, dijital dünyaya geçişteki en önemli ürün olan Set Top Box (STB) ile ilgili teknolojiyi geliştirerek pazarlamak üzere İngiltere'de, Fusion Digital LTD. Şirketini kurmuştur. 2003 Martta kurulan Fusion Digital Technologies Ltd., güçlü ve alanında uzman Ar-Ge ve satış teşkilatını kısa bir sürede İngiltere'nin en önemli Set Top Box şirketlerinden biri haline gelmiştir.

Beko Elektronik hızla gelişen ve değişen teknolojileri daha yakın takip edebilmek amacı ile İzmir'de de teknolojisini hızla geliştirmek için dijital teknoloji üzerine çalışacak, bir ekip oluşturmuştur. Konusunda uzman mühendislerden oluşan bu ekip hızla büyümektedir.

6.6.2. Uygulama ve Sonuçları

Beko Elektronik Şirketi'nde üretim planlama departmanı koordinatörlerinden Mehmet ÇOKKEÇEÇİ ile yapılan görüşmeler sonucunda ürün gruplarından biri olan 55 ekran televizyon grubunun 2004 yılında toplu üretim planının hazırlanabilmesi için gerekli veriler şirketten elde edilmiştir. Bu veriler Tablo 6.6'da sunulmaktadır.

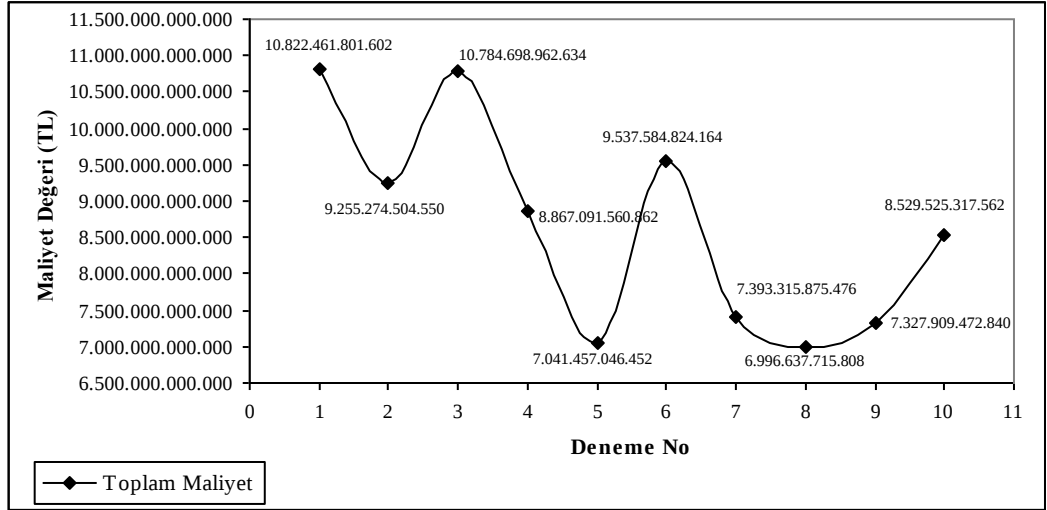
Belirlenen ürün grubu için elde edilen veriler GATÜP programına aktarılmıştır. Bir önceki bölümde programın genetik operatörleri üzerinde yapılan araştırmalarda en iyi sonuçları veren parametre seti bu problemin çözümü için seçilmiştir. Buna göre seçim yöntemi olarak elitizm ile birlikte rulet seçimi, çaprazlama olasılığı olarak

0,65, mutasyon olasılığı olarak 0,001 ve popülasyon büyüklüğü olarak 110 tercih edilmiştir. 55 ekran televizyon grubu toplu üretim planlama probleminin genetik algoritma yaklaşımı ile çözümünde rassallıktan kaynaklanabilecek sapmaları engellemek amacıyla problem GATÜP programında 10 kez çözülmüştür. Bu çözümler sonucunda elde edilen toplam maliyet değerleri Şekil 6.6.'da gösterilen grafikte sunulmuştur.

Tablo 6.6: 55 Ekran Televizyon Grubuna İlişkin Veriler.

Planlanan Dönem	Tahmini Talep Değerleri (adet)	Diğer Veriler
Ocak	70000	Ek mesai üretim maliyeti : 14.453.652 TL/adet
Şubat	70000	Fason üretim maliyeti : 37.425.312 TL/adet
Mart	90000	Geç teslim maliyeti : 12.146.258 TL/adet
Nisan	100000	Mamül elde tutuma maliyeti : 11.941.258 TL/adet
Mayıs	110000	Aylık ortalama 55 ekran üretim miktarı : 77.500 adet
Haziran	130000	Başlangıç stok miktarı : 0 adet
Temmuz	140000	
Ağustos	140000	
Eylül	100000	
Ekim	80000	
Kasım	80000	
Aralık	90000	

Şekil 6.6. incelendiğinde program sonuçlarının yapılan 10 deneme için 10.822.461.801.602 TL ile 6.996.637.715.808 TL arasında değiştiği görülmektedir. GATÜP programının denemeler arasında 55 ekran televizyon grubu için sunduğu minimum maliyetli (toplam maliyet 6.996.637.715.808 TL) toplu üretim planı Şekil 6.7.'de gösterilmektedir.



Şekil 6.6: 55 Ekran Televizyon için GATÜP Deneme Sonuçları.

Genetik Algoritma ile Toplu Üretim Planlama

YARDIM

Çaprazlama Olasılığı: 0,65 Başlangıç Stok Miktarı: 0

Mutasyon Olasılığı: 0,001 Fason Üretim Birim Maliyeti: 37425312

Popülasyon Büyüklüğü: 110 Birim Başına Mesai Maliyeti: 14453652

Seçim Yöntemi: Rulet Yöntemi Geç Teslim Birim Maliyeti: 12146258

Elitizm Yöntemini Uygula ☒ Birim Stok Maliyeti: 11941258

	Üretim Miktarı	Talep Tahmini	MESAI	FASON ÜRETİM	GEÇ TESLİM	BAŞLANGIÇ STOK	SON STOK
OCAK	77500	70000	0	0	0	0	7500
ŞUBAT	77500	70000	0	0	0	7500	15000
MART	77500	90000	0	0	0	15000	2500
NİSAN	77500	100000	20224	0	0	2500	8224
MAYIS	77500	110000	30165	4360	0	8224	10249
HAZİRAN	77500	130000	27173	11416	3662	10249	0
TEMMUZ	77500	140000	30165	26137	9860	-3662	0
AĞUSTOS	77500	140000	28510	1797	42053	-9860	0
EYLÜL	77500	100000	26137	8013	30403	-42053	0
EKİM	77500	80000	24930	3261	4712	-30403	0
KASIM	77500	80000	20110	0	0	-4712	12898
ARALIK	77500	90000	1645	50	0	12898	2093

TOPLAM MALİYET: 6996637715808

HESAPLA

Şekil 6.7: 55 Ekran Televizyon Grubu İçin Minimum Maliyetli Toplu Üretim Planı.

Şekil 6.7. incelendiğinde yılın ilk üç ayında mesai üretim veya fason üretim ihtiyacı olmadığı görülmektedir. Bu durum bu dönemlerde tahmini taleplerin normal üretim miktarından düşük olmasından kaynaklanmaktadır. Plana göre yıl içerisinde 55 ekran televizyonların toplam 217059 adedi mesai yapılarak, 55034 adedi ise fason üretimle

sağlanmalıdır. Fason üretim maliyetinin yüksek olması mesai üretilmesi gerekli miktarı artırmıştır. Bu üretim stratejileriyle yıllık toplam 3137295249468 TL’lik mesai maliyetine, 2059664620608 TL’lik fason üretim maliyetine katlanması gerekmektedir. Toplu üretim planında 90690 adet 55 ekran televizyonun Haziran ve Ekim ayları arasında geç teslim edilmesi sonucunda toplam 1101544138020 TL’lik geç teslim maliyetine katlanılacaktır. Haziran ve Ekim ayları arasında tahmini talep miktarlarının normal üretim kapasitesinden çok fazla olması geç teslimi gerekli kılmıştır. Toplu üretim planına göre yıllık katlanması gerekli elde bulundurma maliyeti stoklanan 58464 adet televizyon için 698133707712 TL’dir.

GATÜP programı ile elde edilen Şekil 6.7.’deki toplu üretim planı Beko Elektronik Şirketi üretim planlama koordinatörü Mehmet ÇOKKEÇECİ tarafından değerlendirilmiştir. **Çokkeçeci (2005)**’e göre GATÜP programının elde ettiği toplam maliyet değeri firma tarafından kabul edilebilir bir değerdir. Şirket içerisinde bu şekilde bir toplu üretim planlama oluşturulmadığı için elde edilen değer şirket değerleri ile birebir karşılaştırılamamaktadır. Birçok maliyet parametresi bulunduran ve bir yıllık bir planlama dönemini içeren bir toplu üretim planı için elde edilen toplam maliyet değeri **Çokkeçeci (2005)**’ye göre uygun görülmüştür. Ancak programda mesai üretim ve fason üretim miktarları için sabit sınırların getirilebilmesi şirketler için daha uygun olacaktır. Çünkü elde edilen toplu üretim planında mesai üretim için belirlenen miktarlar bazı aylar için yüksektir.

Çokkeçeci (2005)’ye göre GATÜP programıyla birçok parametre kullanarak hızlı bir şekilde toplu üretim planının elde edilmesi ve değişen verilerde çözümün kolayca tekrarlanabilmesi firmalar için önemlidir.

7. SONUÇ VE TARTIŞMA

Doğadaki canlıların evrim sürecini taklit ederek zor optimizasyon problemleri için optimuma yakın çözümleri arayan genetik algoritmalar bu çalışmanın temelini oluşturmaktadır. Bu çalışmada genetik algoritmaların uygulama alanlarından biri olan toplu üretim planlama problemlerinin çözümünde kullanılması açıklanmıştır.

Günümüzde mevcut çok farklı üretim planlama yöntemleri mevcuttur. Bu yöntemler basit grafik yöntemlerinden kompleks bilgisayar tabanlı araştırma prosedürlerine kadar değişkenlik göstermektedir. Yöntemlerin kullanılabildiği durumlar kısıtlıdır ve yöntemlerin pratikte kullanılabilirliklerinde de bazı kısıtlar mevcuttur. Bu nedenle planlamacılar kendi firmalarının içinde bulunduğu duruma göre en uygun yöntemi seçmelidirler. Genetik algoritmalar toplu üretim planlama problemleri için sahip oldukları birçok avantajlar nedeniyle uygun bir çözüm yöntemi olabilmektedirler. Çalışmada bu konu ayrıntılı olarak incelenmiştir.

Çalışmada öncelikli olarak literatürden belirlenen bir toplu üretim planlama problemi için genetik algoritma prosedürleri geliştirilerek bu prosedürlerin kolayca uygulanabileceği bir bilgisayar programı olan GATÜP programı .Net programlama dilinde hazırlanmıştır. Bu programın etkinliğini test etmek ve uygun parametre setlerini belirlemek için literatürden elde edilen problemler üzerinde denemeler yapılmıştır. GATÜP programının firmalarda uygulanabilirliği büyük ölçekli bir elektronik şirketi olan Beko Elektronik Şirketi'nde denenmiştir.

Literatürden elde edilen bir toplu üretim planlama problemi için GATÜP programının sunduğu toplu üretim planının toplam maliyet değeri literatürdeki değerlerden daha az bulunmuştur. Özellikle programda yönetim hedef değerlerini daha esnek olması maliyet değerini azaltılmasında etkin olmasına karşın oluşturulan genetik algoritma prosedürlerinin de etkin olarak çalıştığının bir göstergesidir.

Üretim sistemlerinde mevcut problemlerin çözümü için kullanılan genetik algoritmalarda farklı genetik parametre değerleri kullanılmaktadır. Toplu üretim planlama problemi türüne uygun spesifik genetik parametre setinin araştırılması önemlidir. Çünkü genetik algoritmaların uygun sonuçlar verebilmesi için problem türlerine uygun genetik parametrelerin kullanılması çok önemlidir. Bu nedenle GATÜP programı için uygun parametre değerlerinin belirlenebilmesi için birçok denemeler yapılmıştır. Bu denemeler sonucunda elde edilen sonuçlar aşağıda özetlenmiştir;

- Toplu üretim planlama problemi için genetik algoritmalarda seçim sürecinde elitizm stratejisinin uygulanması çözüm etkinliğini artırmaktadır. Çünkü elitizm stratejisinin uygulanmasıyla iyi sonuçlar gelecek nesillere geçebilmekte ve kaybolmamaktadır.
- Toplu üretim planlama problemi için GATÜP programında popülasyon büyüklüğü 90 ve 110 için elitizm yöntemiyle birlikte rulet seçimi, 100 için elitizm yöntemiyle beklenen değer seçimi uygulamaları minimum maliyetli planları oluşturmuştur.
- GATÜP programında toplu üretim planlama problemi için rulet ve elitizm yöntemi için 0,65, beklenen değer ve elitizm yöntemi için 0,6 çaprazlama olasılıkları en düşük maliyetli sonuçları vermiştir.
- Toplu üretim planlama problemi için GATÜP programı her iki seçim yönteminde de 0,001 mutasyon olasılığında en düşük maliyetli sonucu vermiştir.

Bu sonuçlar literatürdeki benzer çalışmalarla karşılaştırıldığında çoğunlukla benzer sonuçlar göstermektedir. Çaprazlama yeni araştırma alanlarının incelenebilirliğini artıran bir parametredir. Bu nedenle olasılığının yüksek olması çözümün daha fazla yeni alanlarda araştırılmasına olanak sağlayacaktır. Ancak bu çalışmada çaprazlama olasılıklarının düşük değerleri optimuma yakın çözümleri vermiştir. Bu durum problemin büyüklüğünden kaynaklanmış olabilir. Daha karmaşık toplu üretim planı problemleri daha yüksek çaprazlama olasılıkları gerektirebilir.

Mutasyon olasılığının en yüksek değerinin veya en düşük değerinin en düşük maliyetli çözümü vermemesi mantıklıdır. En yüksek değer genetik algoritmanın araştırma tekniğini bozabilir, en düşük değer ise çözüm çeşitliliğinin artmasını sağlayamayabilir.

Yapılan bu parametre seti araştırma sonuçları ile GATÜP programının nasıl daha etkin çalıştığının belirlenmesi ve geliştirilen algoritmanın geçerliliğinin test edilmesiyle birlikte gelecekte yapılacak çalışmalarda parametre seçiminin kolaylaştırılması umulmaktadır.

GATÜP programı, toplu üretim planlama probleminin matematiksel olarak modellenmesini gerektirmemektedir. Genetik algoritma konusunda uzman bilgisine gerek olmadan firma içerisinde toplu üretim planlama problemlerinin çözümünü kolayca sunabilecek bir programdır. Bu nedenle genetik algoritma yöntemi geliştirilen GATÜP programı ile birlikte pratikte firmalardaki toplu üretim planlama stratejilerinin belirlenmesi gibi kararlarda uygulanabilir hale gelmiştir. Programın uygulanabilirliği Beko Elektronik Şirketi'nden 55 ekran televizyon grubu için 2004 yılına ait elde edilen veriler doğrultusunda incelenmiştir. Bu verilere göre program yapılan denemeler içerisinde minimum 6.996.637.715.808 TL maliyetli Şekil 6.7'de sunulan toplu üretim planını elde etmiştir. Bu sonuç şirketin üretim planlama koordinatörü tarafından uygun bulunmuştur. Bu sonuca göre programın firmalarda uygulanabilir olduğu kabul edilmiştir.

Geliştirilen GATÜP programı ile toplu üretim planlama problemleri değişen veriler ışığında kısa sürede çözülebilecek duruma gelmiştir. Program genetik algoritma prosedürlerini kullanmaktadır. Genetik algoritmalar ana üretim çizelgeleme ve diğer toplu üretim planı ile bağlantılı problemlerin çözümünde kullanılabilmektedir. Bu nedenle GATÜP programı toplu üretim planlama probleminin yanı sıra bağlantılı diğer problemlerin çözümlerini de verecek şekilde geliştirilebilir.

GATÜP programı genetik algoritma yönteminin sahip olduğu birçok üstünlüğe sahiptir ancak firmalarda toplu üretim planlama problemlerinde genetik algoritmaların uygulanabilirliğini gösteren bir başlangıç noktasıdır. Bu nedenle geliştirilmesi gerekli birçok unsur içermektedir.

GATÜP programı literatürden elde edilen bir toplu üretim planlama problemi için geliştirilmiş bir programdır. Bu nedenle bu problem koşullarına uygun firmalarda uygulanabilirlik sağlayabilecektir. Program içerisine işgücü seviyesi gibi diğer tekniklerde ilave edilerek ve çözüm sürecinde hangi tekniklerin uygulanacağı kullanıcıya bırakılarak programın daha fazla firmalarda uygulanabilirliği sağlanabilir. Ayrıca program firmaların koşullarına özgü olarak geliştirilerek firma için daha anlamlı bilgiler sunabilir.

Geliştirilen GATÜP programının eksikliklerinden biri işgücü seviyesinin değiştirilmesiyle tahmini taleplerin karşılanması tekniğini içermemektedir. Günümüzde personel çıkarımının firmalara çok yönlü olumsuzluklar ve bir takım maliyetler getirmesi nedeniyle sürekli işçi çıkarımı ve alımı ile müşteri taleplerinin karşılanması birçok sektördeki şirketlerde tercih edilmeyen bir yöntemdir. Ancak bu tekniği uygulayan şirketlerde mevcuttur. İşgücü seviyesinin değiştirilmesine bağlı olarak oluşan maliyetlerin programa dahil edilmesi programın daha karmaşık toplu üretim planlama problemlerine çözüm bulabilmesine olanak sağlayacaktır.

GATÜP programında tahmini taleplerin karşılanabilmesi için uygulanacak tekniklerin sınırlandırılabilmesi için rassal bir formül kullanmıştır. Aslında bu sınırlar firmaların yöneticileri tarafından kesin çizgilerle belirlenebilmektedir. Örneğin Beko Elektronik Şirketi için yapılan uygulamada mesai üretim miktarları biraz yüksek bulunmuştur. Mesai üretim miktarı için program firma yöneticisine bağlı olmaksızın kendi çalışma prosedürlerinde mantıklı bir sınır koymaktadır. Ancak bu sınır değeri yönetici hedefini karşılamayabilmektedir. Bu durumun ortadan kalkması için program, planlama tekniklerinin sınır değerlerini kullanıcılarından alabilecek şekilde geliştirilebilir.

GATÜP programı için geliştirilen genetik algoritma prosedürlerinde toplam maliyet değeri, uygulanan stratejilere bağlı olarak maliyet değerlerin doğrusal olarak arttığı varsayılarak hesaplanmıştır. Aslında bazı stratejilerde (örneğin; geç teslim) maliyet değerleri farklı unsurların etkisiyle doğrusal olmamaktadır ki günümüz koşullarında genellikle firmalar bu durumla karşı karşıyadır. GATÜP programının mevcut uygunluk değeri fonksiyonu maliyetlerin doğrusal olmadığı şekilde geliştirilerek programın daha etkin ve pratik koşullara daha uygun sonuçlar vermesi sağlanabilir.

Bilgisayar tabanlı problem çözme tekniklerinin işletmelerde yaygın hale gelmesiyle birçok alanda olduğu gibi üretim sistemleriyle ilgili problemlerin çözümüyle ilişkili kararlarının da kısa sürede, daha güvenilir ve objektif olması sağlanacaktır. Bu çalışmada geliştirilen bilgisayar tabanlı problem çözme tekniklerinden biri olarak GATÜP programının firmalarda uygulanabilir olması toplu üretim planlama problemleriyle ilgili daha etkin kararların alınmasını sağlayacak bir temel oluşturulmuştur.

Çalışmada geliştirilen algoritma sonucunda bulunan optimum çözümler yardımıyla üretim planlamacılar daha sağlıklı ve gerçekçi kararlar verebilecektir. Böylece kaynakların daha etkin kullanımı sağlanacak ve müşteri ihtiyaçları zamanında karşılanarak işletme maliyetleri aşağı çekilebilecektir.

KAYNAKLAR

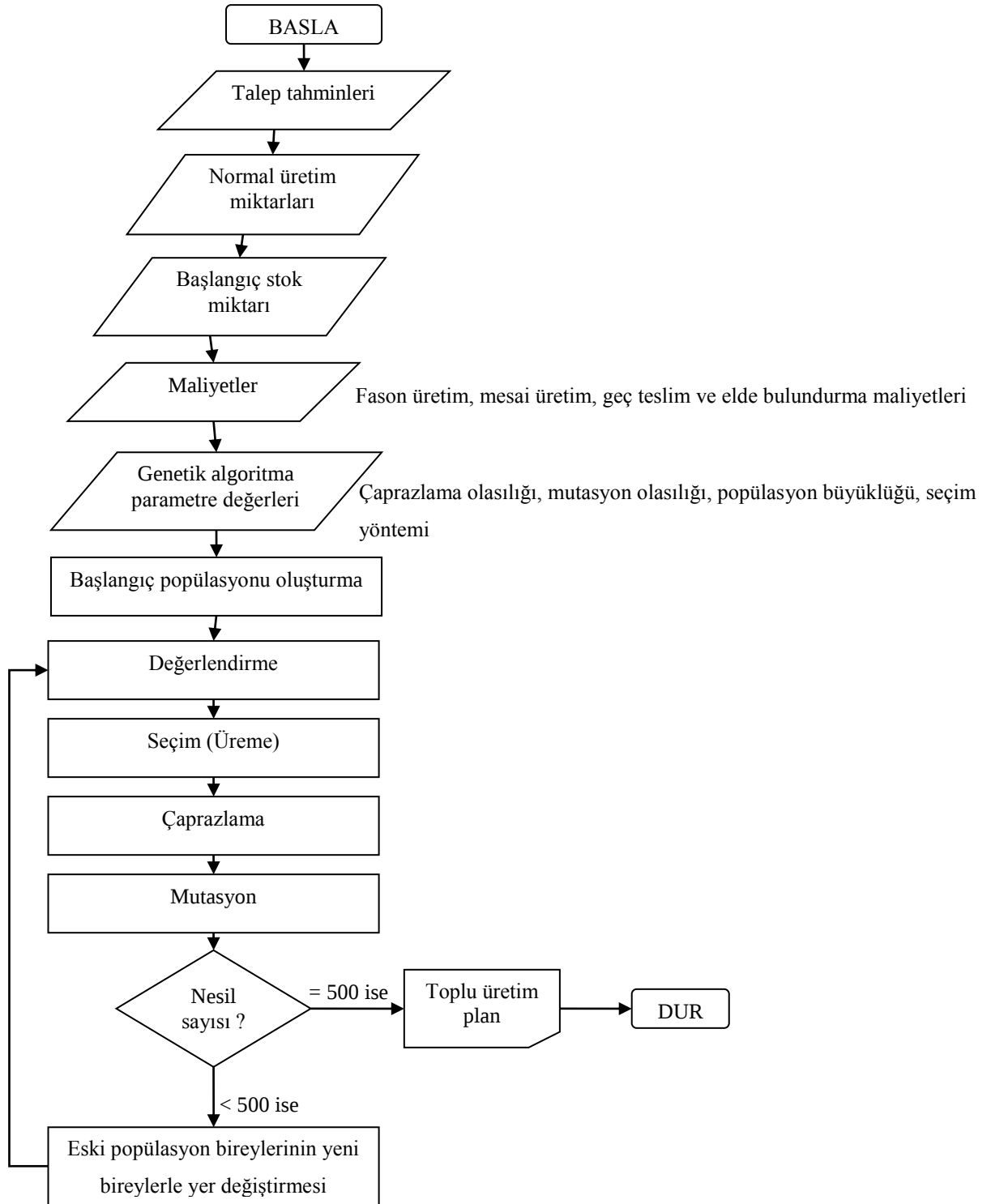
- Arroyo, J. E. C. and Armento, V. A.,** 2004. Genetic local search for multi-objective flow shop problems, *European Journal of Operational Research*,--, ----.
- Bayramoğlu, Y.,** 2005. Kişisel görüşme, Taylor Nelson Sofres Araştırma Danışmanlık ve Ticaret A.Ş., İstanbul
- Biethahn, J. and Nissen, V.,** 1995. An introduction to evolutionary algorithms, in *Evolutionary Algorithms in Management Applications*, pp.3-39, Eds. Biethahn, J. & Nissen, V., Springer-Verlag, Heidelberg.
- Bodenhofer, U.,** 2003. Genetic Algorithms: Theory and Applications, Fuzzy Logic Lab.Linz, Hagenbegr.
- Bolat, A. and Al-Harkan, I. and Al-Harbi, B.,** 2005. Flow-shop scheduling for three serial stations with the last two duplicate, *Industrial Engineering Department*, **32**, 647-667.
- Buckles, B.P. and Petry, F. E.,** 1992. An overview of genetic algorithms and their applications, in *Genetic Algorithms*, pp.5-11, Eds. Buckles, B.P. & Petry, F.E., IEEE Computer Society Pres, California.
- Chen, C.L. and Vempati, V., S. and Aljaber, N.,** 1995. An application of genetic algorithms for flow shop problems, *European Journal of Operational Research*, **80**, 389-396.
- Cheng, R. and Gen, M. and Tsujimura, Y.,** 1999. A tutorial survey of jop-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, *Computers & Industrial Engineering*, **36**, 343 -364.
- Chipperfield, A.,** 1997. Introduction to genetic algorithms, in *Genetic Algorithms in Engineering Systems*, pp. 1-19, Eds. Zalzala, A.M.S. & Fleming, P.J., The Institution of Electrical Engineers, London.
- Chrysosolouris, G. and Subramaniam, V.,** 2001. Dynamic scheduling of manufacturing jop shops using gegentic algorithms, *Journal of Intelligent Manufacturing* **12**, 281-293.
- Coley, D. A.,** 1999. An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific, London.
- Çokkeçeci, M.,** 2005. Kişisel görüşme, Beko Elektronik A.Ş., İstanbul.

- Deb, K.**, 1996. Genetic algorithms for function optimization, in *Genetic Algorithms and Soft Computing*, pp. 3-19, Eds. Herrera, F. & Verdegay, J L., Physica – Verlag, Heidelberg.
- Dianati, M. and Song, I. and Treiber, M.**, 2003. An introduction to genetic algorithms and evolution strategies, <http://www.swen.uwaterloo.ca/~mdianati/articles/gaes.pdf>, 28.11.2004.
- Engin, O.**, 2001. Akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm performansının artırılmasında parametre optimizasyonu, *Doktora Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Eren, H.**, 2002. Akış tipi çizelgeleme problemlerinin genetik algoritma (GA) ile çözüm performansının artırılmasında deney tasarımı uygulaması, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Erfan, S.**, 1994. Toplu üretim planlama ve bilgisayar destekli bir uygulama, *Yüksek Lisans Tezi*, İ.T.Ü. Sosyal Bilimler Enstitüsü, İstanbul.
- Forrest, S.**, 1996. Genetic algorithms, *ACM Computing Surveys*, **28**, 77-80.
- Grefenstette, J.J.**, 1992. Optimization of control parameters for genetic algorithms, in *Genetic Algorithms*, pp.5-11, Eds. Buckles, B.P. & Petry, F.E., IEEE Computer Society Pres, California.
- Gilbert, K. C. and Madan, M.S.**, 1991, A heuristics fir a class of production planning and scheduling problems, *IEE Transactions*, **23(5)**, 282-289.
- Gonçalves, J. F. and Mendes, J. J. M. and Resende, M. G. C.**, 2004. A hybrid genetic algorithm for the jop shop scheduling problem, *European Journal of Operational Research*,--, ----.
- Gonzales, B. and Torres, M. and Moreno, M. J.**, 1995. A hybrid genetic algorithm approach for the “no-wait” flow-shop scheduling problem, *Genetic Algorithms in Engineering Systems: Innovations and Applications Conference*, 12-14 September, 59-64.
- Ho, J. C. and Chang, Y.**, 1991. A new heuristic for the n-jop, M- machine flow-shop problem, *European Journal of Operational Research*, **52**, 194-202.
- Holt, C. C. and Modigliani, F. and Muth, J. F. and Simon, H. A.**, 1960. Plannig, Production, Inventories and Work Force, Prentice-Hall, Canada.
- Kamien, M.I. and Li, L.**, 1990. Subcontracting, coordination, flexibility, and production smoothing in aggregate planning, *Management Science*,**36**, 1352-1363.
- Kulluk, S. ve Türkbey, O.**, 2004. Tesis yerleşimi problemi için bir genetik algoritma, *Yöneylem Araştırması/Endüstri Mühendisliği – XXIV Ulusal Kongresi*, Gaziantep-Adana, 15-18 Haziran.
- Kurt, M. ve Semetay, C.**, 2001. Genetik algoritma ve uygulama alanları, http://www.mmo.org.tr/muhendismakina/arsiv/2001/ekim/Genetik_Algoritma.htm, 28.11.2004.
- Kuşsan, N.**, 1991. Sistem yaklaşımıyla üretim planlama ve kontrol, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

- Liu, Y. C. and Smith, S. S. F.,** 2001. The application of multi-level genetic algorithms in assembly planning, *Journal of Industrial Technology*, **17(4)**, 1-9.
- Marczyk, A.,** 2004. Genetic algorithms and evolutionary computation, <http://www.talkorigins.org/faqs/genalg/genalg.html>, 28.11.2004
- Michalewicz, Z.,** 1992. Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, Heidelberg.
- Morad, N. and Zalzal, A.M.S.,** 1997. Integrated production planning and scheduling in cellular manufacturing using genetic algorithms, *Proceedings of Genetic Algorithms in Engineering Systems: Innovations and Application Conference*, September 1997, 332-338.
- Nam, S. and Logendran, R.,** 1992. Aggregate production planning – a survey of models and methodologies, *European Journal of Operational Research*, **61**, 255-272.
- Odetayo, M. O.,** 1993. Optimum population size for genetic algorithms: an investigation, *IEE Transactions*, 1-3.
- Onwubolu, G.C.,** 2002. Emerging Optimization Techniques in Production Planning and Control, Imperial College Pres, America.
- Öztürk, E. ve Yılmaz, E.,** 2003. Makine tasarım optimizasyon problemlerinin genetik algoritma ile çözümü, *International XII. Turkish Symposium on Artificial Intelligence and Neural Networks*, <http://www.ijci.org/product/tainn/T08002.pdf>, 28.11.2004.
- Park, B. J. and Choi, H. R. and Kim, H. S.,** 2003. A hybrid genetic algorithm for the job-shop scheduling problems, *Computers & Industrial Engineering*, **45**, 597-613.
- Reeves, C.R.,** 1995. A genetic algorithm for flowshop sequencing, *Computer Ops. Res.*, **22**, 5-13.
- Sakawa, M.,** 2002. Genetic Algorithms and Fuzzy Multiobjective Optimization, Kulwer Academic Publishers, London.
- Setyan, H.,** 1994. Toplu üretim planlamadan ana üretim programlamaya geçiş ve bilgisayar destekli bir uygulama, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Spina, R. and Galantucci, L. M. and Dassisti, M.,** 2003. A hybrid approach to the single line scheduling problem with multiple products and sequence-dependent time, *Computers & Industrial Engineering*, **47**, 573-583.
- Stockton, D.J. and Quinn, L.,** 1995. Aggregate production planning using genetic algorithms, *Proceedings of the Institution of Mechanical Engineers*, **209(B)**, 201-209.
- Stockton, D. J. and Quinn, L. and Khalil, R. A.,** 2004a. Use of genetic algorithms in operations management Part 1: applications, *Proceedings of the Institution of Mechanical Engineers*, **219 (B)**, 315-327.
- Stockton, D. J. and Quinn, L. and Khalil, R. A.,** 2004b. Use of genetic algorithms in operations management Part 2: results, *Proceedings of the Institution of Mechanical Engineers*, **218 (B)**, 329-343.

- Tanyaş, M.**, 1982. Çok amaçlı toplu üretim planlama için yeni bir model ve çözüm yöntemi, *Doktora Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- Wang, H. F. and Wu, K. Y.**, 2002. Modeling and analysis for multi-period, multi-product and multi-resource production scheduling, *Journal of Intelligent Manufacturing*, **14**, 297-309.
- Whitley, D.**, 2001. An overview of evolutionary algorithms: practical issues and common pitfalls, *Information & Software Technology*, **43(14)**, 817-831.
- Yu, H. And Liang, W.**, 2001. Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling, *Computers & Industrial Engineering*, **39**, 337-356.
- Zhou, H. and Feng, Y. And Han, L.**, 2001. A hybrid heuristic genetic algorithm for the job shop scheduling, *Computers & Industrial Engineering*, **40**, 191-200.

EK A: GATÜP programının çalışma akış şeması.



Şekil A.1: GATÜP Programının Çalışma Akış Şeması.

EK B: GATÜP programının çalışma şekli.

The screenshot shows the 'Genetik Algoritma ile Toplu Üretim Planlama' (Genetic Algorithm for Batch Production Planning) window. It includes a 'YARDIM' (Help) button and several input fields for parameters: Çaprazlama Olasılığı (Crossover Probability), Mutasyon Olasılığı (Mutation Probability), Popülasyon Büyüklüğü (Population Size), Seçim Yöntemi (Selection Method), Elitizm Yöntemini Uygula (Apply Elitism Method) checkbox, Başlangıç Stok Miktarı (Initial Stock Quantity), Fason Üretim Birim Maliyeti (Setup Production Unit Cost), Birim Başına Mesai Maliyeti (Unit Labor Cost), Geç Teslim Birim Maliyeti (Late Delivery Unit Cost), and Birim Stok Maliyeti (Unit Stock Cost).

	Üretim Miktarı	Talep Tahmini	MESAI	FASON ÜRETİM	GEÇ TESLİM	BAŞLANGIÇ STOK	SON STOK
OCAK	0	0					
ŞUBAT	0	0					
MART	0	0					
NİSAN	0	0					
MAYIS	0	0					
HAZİRAN	0	0					
TEMMUZ	0	0					
AĞUSTOS	0	0					
EYLÜL	0	0					
EKİM	0	0					
KASIM	0	0					
ARALIK	0	0					

Below the table, there is a 'TOPLAM MALİYET' (Total Cost) field and a 'HESAPLA' (Calculate) button.

Şekil B.1: GATÜP Programı Kullanıcı Arayüzü.

Program kullanıcıları GATÜP programının çalıştırılması ve seçmeleri gerekli genetik algoritma parametreleri hakkında YARDIM butonuna basarak bilgi alabilmektedirler. Çaprazlama olasılığı, mutasyon olasılığı, popülasyon büyüklüğü değerleri, seçim yöntemi ve elitizm uygulanıp uygulanmaması gerektiği seçimi kullanıcı tarafından yapılmalıdır. Toplu üretim planlama problemi ile ilgili fason üretim birim maliyeti, mesai üretim birim maliyeti, geç teslim birim maliyeti ve stok birim maliyeti gibi maliyet değerleri, aylık tahmini talep miktarları, normal üretim miktarları ve başlangıç stok miktarı kullanıcı tarafından programa girilerek problem tanımlanmaktadır. HESAPLA butonuna basıldığında program kullanıcıdan aldığı verilere uygun olarak genetik algoritma prosedürlerini kullanarak hesapladığı minimum maliyetli toplu üretim planını ve planın maliyet değerini kullanıcıya sunmaktadır.

EK C: GATÜP programının yazılım kodları.

Genetik algoritma yönteminde öncelikli olarak toplu üretim planlama problemine ait çözümlerin rassal olarak belirlenerek başlangıç popülasyonun oluşturulması gerekmektedir. GATÜP programında bu aşama aşağıdaki kod dizisiyle sağlanmaktadır;

```
public void createBits()
{
    bits = new bit[12];
    lateDelivery = new int[12];
    finishingStock = new int[12];
    openingStock = new int[12];
    openingStock[0]= parameters.beginningInventory;
    int y1 = rnd.Next(parameters.overtimeUpperLimit)+1;
    int y2 = rnd.Next(parameters.subcontractUpperLimit)+1;
    int x1 = parameters.overTimeCost + parameters.subcontractCost;
    int[] fact= new int[12];
    for (int months = 0;months<12;months++)
    {
        fact[months] = parameters.forecastDemand[months ] - parameters.normalProduction[months ] -
        openingStock[months ];
        Double pxo = 1 + (double) ( (double) ((rnd.Next(parameters.overtimeUpperLimit)+1) *
        parameters.subcontractCost) / (double) (fact[months] * parameters.overtimeUpperLimit * x1 ));
        Double pxs = 1 +(double) ( (double) ((rnd.Next(parameters.subcontractUpperLimit)+1) *
        parameters.overTimeCost) / (double) (fact[months] * parameters.subcontractUpperLimit * x1 ));

        if (pxo>1)      {      bits[months ].overtime = rnd.Next(y1+1);      }
        else           {      bits[months ].overtime = 0;      }
        if (pxs>1)      {      bits[months ].subContract= rnd.Next(y2+1);      }
        else           {      bits[months ].subContract= 0;      }
        int z = bits[months ].overtime + parameters.normalProduction[months ] + bits[months
        ].subContract + openingStock[months ] - parameters.forecastDemand[months ];

        if (months!=11) openingStock[months + 1] = z;
        if (z<0)        {      lateDelivery[months ]= -1 * z;  finishingStock[months ] = 0;      }
        else            {      lateDelivery[months ]= 0;      finishingStock[months ] = z;      }
    }
    public void calculateDeducedParameters()
    {
        lateDelivery = new int[12];
        finishingStock = new int[12];
        openingStock = new int[12];
        openingStock[0]= parameters.beginningInventory;
        for (int months = 0;months<12;months++)
        {
            int z = bits[months ].overtime + parameters.normalProduction[months ] + bits[months
            ].subContract + openingStock[months ] - parameters.forecastDemand[months ];
            if(months!=11) openingStock[months + 1] = z;
```

```

if (z<0)      {      lateDelivery[months ]= -1 * z;  finishingStock[months ] = 0;  }
else         {      lateDelivery[months ]= 0;      finishingStock[months ] = z;   }
              }      }

public void calculateTotalCost()
{
    int overtimeTotal=0;
    for (int months=0;months<12;months++)
    {
        overtimeTotal += bits[months].overtime;
    }
    int totalOvertimeCost = overtimeTotal * parameters.overTimeCost;
    int subcontractTotal=0;
    for (int months=0;months<12;months++)
    {
        subcontractTotal+= bits[months].subContract;
    }
    int totalSubcontractCost= subcontractTotal * parameters.subcontractCost;
    int lateDeliveryTotal=0;
    for (int months=0;months<12;months++)
    {
        lateDeliveryTotal += lateDelivery[months];
    }
    int totalLateDeliveryCost= lateDeliveryTotal * parameters.lateDeliveryCost;
    int totalFinishingStock = 0;
    for (int months=0;months<12;months++)
    {
        totalFinishingStock += finishingStock[months];
    }
    int totalFinishingStockCost = totalFinishingStock * parameters.stockCost;

    totalCost = totalFinishingStockCost + totalLateDeliveryCost + totalOvertimeCost +
    totalSubcontractCost;
}

```

Başlangıç popülasyonunun oluşturulmasından sonra bu popülasyon üzerinde genetik algoritma prosedürlerinin uygulanması gerekmektedir. GATÜP programında bu genetik prosedürler aşağıdaki kod dizilimiyle uygulanmaktadır;

```

public class population
{
    private int totalScore;
    private static Random rnd= new Random();
    public individual[] individuals;
    public population(int populationNumber)
    {
        individuals = new individual[populationNumber];
        for (int counter=0;counter<individuals.Length;counter++)
        {
            individuals[counter]= new individual();
        }
        calculateScoresAndExpectedValues();
    }
    public void crossover(individual ind1,individual ind2)
    { if ( rnd.NextDouble()<parameters.crossoverProbability )
        {
            int startPoint,stopPoint;
            startPoint = rnd.Next(0,10);
            stopPoint = rnd.Next(startPoint+1,11);
            for (int i = startPoint;i<=stopPoint;i++)
            {
                int tempSwap = ind1.bits[i];
                ind1.bits[i] = ind2.bits[i];
                ind2.bits[i] = tempSwap;
            }
            ind1.calculateDeducedParameters();
            ind1.calculateTotalCost();
            ind2.calculateDeducedParameters();
            ind2.calculateTotalCost();
        }
    }
    public void mutation(individual ind1,individual ind2)
    { if ( rnd.NextDouble()<parameters.mutationProbability )
        {
            int mutationPoint1 = rnd.Next(0,23);
            Boolean isOvertime1 = mutationPoint1%2==0;

```

```

        mutationPoint1 /= 2;
        int mutationPoint2 = rnd.Next(0,23);
        Boolean isOvertime2 = mutationPoint2%2==0;
        mutationPoint2 /= 2;
        int tempSwap;
        if (isOvertime1)    {    tempSwap = ind1.bits[mutationPoint1].overtime;
        if (isOvertime2)
{ind1.bits[mutationPoint1].overtime  = ind2.bits[mutationPoint2].overtime;
ind2.bits[mutationPoint2].overtime  = tempSwap;          }
        else
{ind1.bits[mutationPoint1].overtime  = ind2.bits[mutationPoint2].subContract;
ind2.bits[mutationPoint2].subContract = tempSwap;          }          }
        else
{tempSwap = ind1.bits[mutationPoint1].subContract;
        if (isOvertime2)
{ind1.bits[mutationPoint1].subContract  = ind2.bits[mutationPoint2].overtime;
ind2.bits[mutationPoint2].overtime  = tempSwap;          }
        else
{ind1.bits[mutationPoint1].subContract  = ind2.bits[mutationPoint2].subContract;
ind2.bits[mutationPoint2].subContract = tempSwap;          }          }
        ind1.calculateDeducedParameters();
        ind1.calculateTotalCost();
        ind2.calculateDeducedParameters();
        ind2.calculateTotalCost();          }          }
public void calculateScoresAndExpectedValues()
{
    int maxTotalCost = -1;
    for (int counter=0;counter<individuals.Length;counter++)
    {
        if (individuals[counter].totalCost > maxTotalCost)
        {
            maxTotalCost = individuals[counter].totalCost ;    }
    }
    totalScore=0;
    for (int counter=0;counter<individuals.Length;counter++)
    {
        individuals[counter].score = maxTotalCost - individuals[counter].totalCost;
        totalScore += individuals[counter].score;    }
    for (int counter=0;counter<individuals.Length;counter++)
    {
        individuals[counter].expectedValue =(double)((double) (individuals[counter].score * 30)
/(double) totalScore);
    }
}
public int getMate()
{
    Double rouletteWheel =(double) (rnd.NextDouble() * (double) totalScore);
    int sum=0;
    int individualCounter=0;
    do
    {
        sum += individuals[individualCounter++].score;    }
    while (sum < rouletteWheel && individualCounter<individuals.Length );
    return individualCounter;
}

public void nextGeneration(selectionMethodType selectedSelectionMethodType,Boolean
isElitizimSelected)
{
    if (selectedSelectionMethodType == selectionMethodType.expectedValue)
    {
        individual[] tempIndividuals= new individual[individuals.Length];
        int counter=0;
        for (int i=0;i<individuals.Length;i++)
        {
            for (int j=0;j<(int) individuals[i].expectedValue;j++)
            {
                tempIndividuals[counter++]=individuals[i];    }
        }
        int individualsCounter=0;
        do

```

```

        {tempIndividuals[counter++]=individuals[individualsCounter++];}
while (counter<individuals.Length);
individual[] tempNextGeneration= new individual[individuals.Length];
for(int i =0 ; i<individuals.Length/2; i++)
{
    int tempIndividualIndex1 = rnd.Next(0,individuals.Length-1);
    int tempIndividualIndex2 = rnd.Next(0,individuals.Length-1);
individual tempIndividual1 = new individual( tempIndividuals[tempIndividualIndex1]);
individual tempIndividual2 = new individual( tempIndividuals[tempIndividualIndex2]);
crossover(tempIndividual1,tempIndividual2);
tempNextGeneration[2*i] = tempIndividual1;
tempNextGeneration[2*i+1] = tempIndividual2;
}
for(int i =0 ; i<individuals.Length/2; i++)
{
    int tempIndividualIndex1 = rnd.Next(0,individuals.Length-1);
    int tempIndividualIndex2 = rnd.Next(0,individuals.Length-1);
mutation(tempNextGeneration[tempIndividualIndex1],tempNextGeneration[tempIndividualIndex
2]);
}
individual tempElitIndividual = individuals[0];
individuals = tempNextGeneration;
calculateScoresAndExpectedValues();
sortIndividuals();
if (isElitizimSelected) {
    individuals[individuals.Length-1] = tempElitIndividual;
    calculateScoresAndExpectedValues();
    sortIndividuals();
}
else { individual[] tempNextGeneration= new individual[individuals.Length];
for(int i =0 ; i<individuals.Length/2; i++)
{
    int tempIndividualIndex1 = getMate();
    int tempIndividualIndex2 = getMate();
individual tempIndividual1 = new individual( individuals[tempIndividualIndex1]);
individual tempIndividual2 = new individual( individuals[tempIndividualIndex2]);
crossover(tempIndividual1,tempIndividual2);
tempNextGeneration[2*i] = tempIndividual1;
tempNextGeneration[2*i+1] = tempIndividual2;
}
for(int i =0 ; i<individuals.Length/2; i++)
{
    int tempIndividualIndex1 = rnd.Next(0,individuals.Length-1);
    int tempIndividualIndex2 = rnd.Next(0,individuals.Length-1);
mutation(tempNextGeneration[tempIndividualIndex1],tempNextGeneration[tempIndividualIndex
2]);
}
individual tempElitIndividual = individuals[0];
individuals = tempNextGeneration;
calculateScoresAndExpectedValues();
sortIndividuals();

if (isElitizimSelected)
{
    individuals[individuals.Length-1] = tempElitIndividual;
    calculateScoresAndExpectedValues();
    sortIndividuals();
}
}

```

ÖZGEÇMİŞ

1979 İstanbul doğumlu olan Mehtap Vural, İstanbul Ticaret Odası Anadolu Teknik Lisesi Bilgisayar Bölümü'nden mezun olduktan sonra 1997 yılında İstanbul Teknik Üniversitesi Kimya-Metalurji Fakültesi Gıda Mühendisliği Bölümü'ne girmiştir. Bu fakülteden 2002 yılında fakülte birincisi olarak mezun olan Mehtap Vural, aynı yıl İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü İşletme Anabilim Dalı İşletme Mühendisliği Programı'nda yüksek lisans eğitimine başlamıştır. Mehtap Vural'ın Selçuk Üniversitesi tarafından 2004 yılında düzenlenen 4. Ulusal Üretim Araştırmaları Sempozyumu'nda sunulmuş ve bildiriler kitabında yayımlanmış "Tedarikçi Değerlendirme ve Seçimi İçin Bir Uzman Sistem Yaklaşımı" isimli bir çalışması bulunmaktadır.

2002 – 2005 yılları arasında PQS Danışmanlık ve 3A Danışmanlık firmalarında, şirketlere ISO 9001:2000 Kalite Yönetim Sistemi ve HACCP Gıda Güvenlik Sistemi kurulum aşamasında sistem danışmanlığı yapan Mehtap Vural, aynı zamanda şirketlere ISO 9001:2000 Kalite Yönetim Sistemi, HACCP Gıda Güvenlik Sistemi ve personel hijyeni eğitimleri vermiştir.