

T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN
ÇOK AMAÇLI GENETİK ALGORİTMA İLE ÇÖZÜMÜ

End. Müh. Kenan KESKİN

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Konya, 2010

T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇOK
AMAÇLI GENETİK ALGORİTMA İLE ÇÖZÜMÜ**

End. Müh. KENAN KESKİN

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 10.03.2010 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Doç. Dr. Orhan Engin
(Danışman)

Prof. Dr. Ahmet Peker
(Üye)

Yrd. Doç. Dr. Ahmet Sarucan
(Üye)

ÖZET

Yüksek Lisans Tezi

BEKLEMESİZ AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇOK AMAÇLI MELEZ GENETİK ALGORİTMA İLE ÇÖZÜMÜ

End. Müh. Kenan KESKİN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Orhan ENGİN

2010, 94 Sayfa

Jüri: Prof. Dr. Ahmet PEKER

Doç. Dr. Orhan ENGİN

Yrd. Doç. Dr. Ahmet SARUCAN

Bu çalışmada, beklemesiz akış tipi çizelgeleme problemleri için bir çok amaçlı melez genetik algoritma önerilmiştir. Geliştirilen çok amaçlı genetik algoritma üretilen üç seviyeli problemlerle test edilmiş ve yapay sinir ağları (YSA) algoritmasıyla kıyaslanmıştır. Kıyaslama sonucunda çok amaçlı melez genetik algoritma, baskın olmayan çözüm uzayında daha iyi bir çözüm bulmakta üstün performans göstermiş ve beklemesiz akış tipi problemlerde toplam akış zamanı (F_i) ve en fazla tamamlanma zamanının (C_{max}) en küçüklemeinde çok etkili bir araç olduğunu kanıtlamıştır.

Anahtar Kelimeler: Çizelgeleme, Beklemesiz Akış Tipi Problemler, Çok Amaçlı Melez Genetik Algoritma.

ABSTRACT

Master Thesis

SOLVING THE NO WAIT FLOW SHOP SCHEDULING PROBLEMS BY MULTI OBJECTIVE HYBRID GENETIC ALGORITHM

Kenan KESKIN

Selcuk University

Graduate School of Natural and Applied Sciences

Department of Industrial Engineering

Supervisor: Associate Prof. Dr. Orhan ENGIN

2010, 94 Pages

Jury: Prof. Dr. Ahmet PEKER

Assoc. Prof. Dr. Orhan ENGIN

Ass. Prof. Dr. Ahmet SARUCAN

In this study, a hybrid multi objective genetic algorithm has been proposed for no wait flow shop scheduling problems. Enhanced multi objective hybrid genetic algorithm tested by three level problems and solutions compared with artificial neural networks algorithm. As a result of comparison, the multi objective hybrid genetic algorithm showed that superior performance to find a better solution in the non-dominated solution space and proved that very effective tool for minimizing total flow time (F_i) and makespan (C_{max}) criterions in no wait flow shop problems.

Key Words: Scheduling, No Wait Flow Shop Problems, Multi Objective Hybrid Genetic Algorithm.

ÖNSÖZ

Bu çalışmanın hazırlanmasında gösterdiği çaba ve katkılarından dolayı tez danışmanım, Doç. Dr. Orhan Engin'e ve geliştirilen algoritmanın kodlanmasındaki yardımlarından dolayı değerli meslektaşım, End. Müh. Mustafa Kerim Yılmaz'a şükranlarımı sunarım.

Ayrıca tez çalışmam boyunca göstermiş olduğu anlayış ve katkılarından dolayı eşim, Özgül Karaman Keskin'e ve aileme minnettar olduğumu belirtmek isterim.

End. Müh. Kenan KESKİN

Mart – 2010

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
KISALTMALAR ve SİMGELER	viii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	3
2.1. Beklemesiz Akış Tipi Problemler İçin Kaynak Araştırması	3
2.2. Çok Amaçlı Genetik Algoritma İçin Kaynak Araştırması	12
3. MATERYAL ve METOD	17
3.1. Beklemesiz Akış Tipi Problemler (BATP)	17
3.1.1. Akış tipi problemlerin (ATP) temel yapısı	17
3.1.2. BATP'nin temel yapısı	23
3.1.2.1. BATP'nin uygulama alanları	25
3.1.3. Çok amaçlı beklemesiz akış tipi problemler (ÇABATP)	30
3.1.3.1. Amaç fonksiyonları	30
3.1.3.2. ÇABATP'nin yapısı	31
3.2. Genetik Algoritmalar (GA)	32
3.2.1. Yazılımlar ve adaptasyon	32
3.2.2. Biyolojik evrim	33
3.2.3. GA'nın temel yapısı	36
3.2.4. Genetik operatörler	39
3.2.4.1. Seçme	39
3.2.4.2. Çaprazlama	39
3.2.4.3. Mutasyon	40
3.2.5. Elitizm	41

3.2.6. Çaprazlama ve mutasyon olasılığı	42
3.2.7. GA'ların akışı	42
3.2.8. GA'ların kullanılma nedenler	44
3.2.9. Fonksiyon maksimizasyon örneği	46
3.2.10. GA'ların genel uygulama alanları	50
3.2.10.1. Optimizasyon	50
3.2.10.2. Otomatik programlama ve bilgi sistemler	51
3.2.10.3. Mekanik öğrenme	51
3.2.10.4. Ekonomik ve sosyal sistem modelleri	52
3.2.10.5. Finans	52
3.2.10.6. Pazarlama	52
3.2.10.7. Üretim-lojistik	53
3.2.11. Çok amaçlı GA'lar (ÇAGA)	54
3.2.12. GA ve çizelgeleme	56
3.2.13. Parametre optimizasyonu	57
4. ARAŞTIRMA SONUÇLARI ve TARTIŞMA	59
4.1. Veri Toplama	59
4.1.1. Geliştirilen çok amaçlı melez GA'nın yapısı	62
4.2. Parametre Optimizasyonu	68
4.3. YSA ile Önerilen Çok Amaçlı Melez GA'nın Karşılaştırılması	70
4.4. Çok Amaçlı Uygulamalar ve Memnuniyet Fonksiyonu	79
5. SONUÇ ve ÖNERİLER	90
6. KAYNAKLAR	92

ŞEKİL LİSTESİ

Şekil 3.1	İş sıralaması için gant şeması ve farklı işler için bekleme süreleri	18
Şekil 3.2	Yarı mamul stoksuz akış tipi problem için (1,2,3) optimal çizelgesinin gant şemasıyla gösterimi	21
Şekil 3.3	Yarı mamul stoksuz akış tipi problem için (1,3,2) optimal çizelgesinin gant şemasıyla gösterimi	22
Şekil 3.4	Yarı mamul stoksuz akış tipi problemler için (5,3,2,1,4) sıralaması	22
Şekil 3.5	BATP örneği $n=4$, $m=4$	24
Şekil 3.6	Ham petrolün işlenme süreci	25
Şekil 3.7	Metal enjeksiyon kalıplama süreci	27
Şekil 3.8	Etil Alkol Üretim Süreci	28
Şekil 3.9	İki tabanından mutasyona uğramış kod ve on tabanından karşılığı	41
Şekil 3.10	GA'nın akış şeması	44
Şekil 3.11	Rulet çemberi yöntemi	47
Şekil 3.12	GA akış şeması	55
Şekil 4.1	Memnuniyet miktarı fonksiyonu	61
Şekil 4.2	Memnuniyet fonksiyonu oranı	61
Şekil 4.3	Çok amaçlı melez GA yapısı	63
Şekil 4.4	Dikey dizi çaprazlama yöntemi	64
Şekil 4.5	Ekleme arama algoritması	65
Şekil 4.6	Kes tamir et fonksiyonlu ekleme arama algoritması	66
Şekil 4.7	Mutasyon akış şeması	67

TABLO LİSTESİ

Tablo 3.1	Beş iş üç makine için örnek problem	18
Tablo 3.2	Üç iş üç makine için örnek problem	21
Tablo 4.1	Seçilen çok amaçlı melez GA parametreleri	69
Tablo 4.2	İki makineli iş kombinasyonu için çözümlerin karşılaştırılması	71
Tablo 4.3	Üç makineli iş kombinasyonu için çözümlerin karşılaştırılması	72
Tablo 4.4	Beş makineli iş kombinasyonu için çözümlerin karşılaştırılması	73
Tablo 4.5	Sekiz makineli iş kombinasyonu için çözümlerin karşılaştırılması	74
Tablo 4.6	On makineli iş kombinasyonu için çözümlerin karşılaştırılması	75
Tablo 4.7	On beş makineli iş kombinasyonu için çözümlerin karşılaştırılması	76
Tablo 4.8	Yirmi makineli iş kombinasyonu için çözümlerin karşılaştırılması	77
Tablo 4.9	Yirmi beş makineli iş kombinasyonu için çözümlerin karşılaştırılması	78
Tablo 4.10	İki makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	80
Tablo 4.11	Üç makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	81
Tablo 4.12	Beş makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	82
Tablo 4.13	Sekiz makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	83
Tablo 4.14	On makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	84
Tablo 4.15	On beş makineli kombinasyonlarının tek amaçlı çözümden sapmaları	85
Tablo 4.16	Yirmi makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	86
Tablo 4.17	Yirmi beş makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları	87
Tablo 4.18	Çok amaçlı yapıdaki F_i ve C_{max} değerlerinin tek amaçlı çözümden sapmaları	88

KISALTMALAR ve SİMGELER

(I/O)	Giriş-çıkış kontrolleri
ATP	Akış tipi problemler
BATP	Beklemesiz akış tipi problemler
BOSGA 2	Baskın olmayan sıralama yapılı genetik algoritma 2
C_{max}	Son işin tamamlanma zamanı
CPU	Merkezi işlem birimi
ÇABATP	Çok amaçlı beklemesiz akış tipi problemler
ÇAGA	Çok amaçlı genetik algoritma
F_i	Toplam akış süresi
GA	Genetik algoritma
I_{SUM}	Makinelerin toplam boş kalma süresi
NEH	Nawaz, Enscore, Ham algoritması
NP	Polinom olarak ifade edilemeyen
NP-Tam	Karmaşıklığı arttığında hesaplama süresi de sınırlandırılmamış biçimde artan ve polinom olarak ifade edilemeyen problemler.
NP-Zor	Polinom olarak ifade edilemeyen zor problemler
N_T	Geciken işlerin toplam sayısı
TB	Tavlama Benzetimi
TAA	Tabu arama algoritması
T_{max}	En fazla gecikme süresi
YSA	Yapay sinir ağları algoritması

1. GİRİŞ

BATP, ATP'nin özel bir halidir. ATP işlerin, seri halde dizilmiş ya da böyle modellenebilecek makineler üzerinden işlenmesi biçiminde düşünülebilir. Bazı ATP'de belirli işler aynı makine üzerinde birden fazla işlem görebilirler. Yine bazılarında belirli işler belirli makineleri atlayabilirler.

BATP ile ilgili literatürde -beklemesiz kısıdına ek olarak başka kısıtların da eklendiği- birçok uygulama bulunmaktadır. Ayrıca endüstride de pek çok uygulama alanı vardır. Çünkü beklemesiz kısıdının başka bir anlamı da -en azından beklemesiz olan süreçler arasında- yarı mamul stokunun olmamasıdır.

GA'lar ilk ortaya atıldığı 1975 yılından bu yana oldukça farklı uygulama alanları olan, parametreleri üzerinde birçok çalışma yapılmış güçlü sezgisellerdir. GA'lar belirleyici algoritmalarla çözülemeyen çok amaçlı problemlere optimal ya da optimale yakın çözüm bulmada da oldukça etkili tekniklerdir. Gücünü doğal seçim olarak adlandırılan, canlıların temel hayatta kalma ve soyunu sürdürme prensibinden alan GA çözüm uzayında kısa sürede iyi çözümler elde edebilmektedir. Mutasyon, üreme, çaprazlama gibi çözümleri çeşitlendiren araçlara sahip olan GA'lar literatürde ve pratikte en çok kullanılan ve geliştirilen sezgisel algoritmalarından birisidir. GA ve ondan esinlenerek geliştirilen diğer evrimsel algoritmalar, yaygın olarak kabul görmektedir. GA'lar gerek ATP'lerin gerekse BATP'nin çizelgelenmesinde uygun ve etkili araçlardır.

Bu çalışmada B ATP için bir çok amaçlı melez genetik algoritma önerilmiştir. Sonuçlar, benzer problemler üzerinde “Yapay Sinir Ağları Algoritması (YSA)” ile kıyaslanmıştır.

Çalışmanın ikinci bölümde B ATP ve GA ile ilgili literatür araştırması yapılmış olup kronolojik sırayla listelenmiştir.

Üçüncü bölümde B ATP problemler ile ilgili bilgi verilmiş olup, kısaca çok amaçlı yapıdan bahsedilmiş ve GA’ların ortaya çıkışı ile doğal yaşamla olan benzerlikleri, genel yapısı, işleyişi, özellikleri, parametreleri gibi kavramlardan bahsedilmiştir.

Dördüncü bölümde B ATP’ler de F_i ve C_{max} amaçlarını en küçükleme için bir çok amaçlı melez GA uygulaması bulunmaktadır. Algoritma, Delphi programla dilinde kodlanmış, 7.0 versiyonu derleyicide derlenerek uygulama dosyası haline getirilmiştir. Kodlanan çok amaçlı melez GA uygulaması problem yapısına ve büyüklüklerine göre farklı parametreler için çalıştırılmış ve YSA ile karşılaştırılmıştır.

Beşinci bölümde uygulama sonuçları genel olarak değerlendirilmiş ve sonraki çalışmalarla ilgili önerilere yer verilmiştir.

2. KAYNAK ARAŞTIRMASI

2.1 BATP İçin Kaynak Araştırması

Qian ve ark. (2009) bir melez türevsel evrim algoritması önermişlerdir. Bunun için öncelikle iş permütasyonlarını; gerçek vektörlerden türevsel evrimin içerisindeki bireylere dönüştüren “en büyük sipariş miktarı” kuralı dikkate alınmıştır. Ardından türevsel evrim yapılı paralel evrim mekanizması uygulanarak basit fakat verimli bir yerel arama algoritması geliştirilmiştir. Son olarak evrim metodunu hızlandıran, araya yerleştirme tabanlı komşuluk araştırma yöntemi eklenmiştir. Önerdikleri diferansiyel evrim algoritması Carlier, Reeves ve Heller problemleri ile test etmişlerdir.

Czogalla ve ark. (2009) BATP için evrimsel algoritmaların analizi ve tasarımı adlı çalışmalarında, yerel arama algoritması eklenerek melez hale getirilmiş parçacık sürü optimizasyonu algoritması ile yalın haldeki parçacık sürü optimizasyonu ve GA'yı karşılaştırılmıştır. Özellikle çözüm kalitesi, popülasyon farklılığı, yakınsama davranışı ve popülasyon farklılığının korunmasını analiz etmişlerdir. T_{max} 'ı (en fazla gecikme süresi) en küçüklemeyi hedefleyen algoritmalar, yerel arama algoritmalarının komşuluk yapısına, karşılaştırma problemlerinin büyüklüğüne ve çaprazlama operatörüne göre farklı performans göstermişlerdir.

Pan ve ark. (2009) BATP'de T_{max} ve C_{max} 'ı en küçükleyen, bir çok amaçlı diferansiyel evrimsel algoritma geliştirmişlerdir. Bunun için kesikli diferansiyel evrimsel algoritma içine bir yerel arama algoritması yerleştirilmiştir. Geliştirdikleri

algoritmayı, Carrier, Heller, Reeves problemleri için, Qian ve ark. önerdiği melez diferansiyel evrimsel algoritma ve Ishibuchi ve ark. önerdiği çok amaçlı genetik algoritma ile kıyaslamışlar, daha düşük C_{max} ve T_{max} değerleri elde etmişlerdir.

Jolai ve ark. (2009) beklemesiz çok aşamalı esnek -bazı işlerin iptal edilebildiği- akış tipi problemler için bir GA önermişler ve bir karışık tam sayılı doğrusal programlama modeli kurarak, modeli GA, TAA (tabu arama algoritması) ve LINGO 8 yazılımı ile çözmüşlerdir. Hesaplama süreleri bakımından GA ve TAA ile LINGO 8' e göre kıyaslanamayacak kadar düşük CPU (merkezi işlem birimi) zamanları elde etmiştir. Ayrıca GA, TAA'ya göre az da olsa daha iyi işlemci zamanları elde edilmiştir. Problemin büyüklüğü arttıkça işlemci zamanları arasındaki farklılık artmaktadır. GA; her bir aşamada; optimal çözümün LINGO 8 ile bulunabildiği durumlar için en fazla %1 oranında optimal çözümden sapma göstermiştir. Bu sapmanın doğrudan iş ve makine sayısı ile ilgili olduğu gözlenmiş olup GA ile %65'den fazla oranda optimal çözüm elde edilmiştir.

Kim ve ark. (2009) beklemesiz esnek parti akışı problemlerinin çizelgelenmesi için, bir genetik algoritma uyarlamışlardır. Esnek parti akışı problemleri, farklı ürünlerin sipariş miktarlarını alt öbeklere ayırmayı ve farklı süreç zamanları olan alternatif makineleri dikkate almayı gerektirmektedir. Belirli bir alt öbeğin birbiriyle karışmasına izin verilmemektedir ve birbirlerine göre öncelikleri yoktur. Algoritma, C_{max} 'ı en küçükleme amacıyla, üç aşamalı olarak oluşturulmuştur. İlk olarak pozisyon tabanlı çaprazlama ve yerel arama tabanlı mutasyon daha iyi döller elde edilmesi için atanmıştır. İkinci olarak mevcut nesli geliştirmek için yenilemeli bir tepe tırmanma algoritması geliştirilmiştir. Son olarak mutasyon ve çaprazlama oranları düzenlenmiştir. Klasik GA, yenilemeli tepe

tırmanma algoritması eklenmiş GA ve hem yenilemeli hem de çaprazlama ve mutasyon oranlarının düzenlendiği adapte edilmiş GA farklı ürün ve alt öbek büyüklükleri için kıyaslanmıştır. Adapte edilmiş GA'nın daha düşük CPU zamanlarıyla benzer C_{max} değerlerine ulaştığı gözlemlenmiştir.

Laha ve ark. (2008) beklemez akış tipi problemlerinde araya iş ekleme durumu için yapısal bir sezgisel yöntem geliştirmişlerdir. Geliştirdikleri yöntemi Rajendaran, Gangadharan ve Rajendran, Aldowaisan ve Allahverdi, Osman ve Potts'un önerdiği algoritmalarla kıyaslamışlar; istatistiksel anlamda daha iyi C_{max} değerleri ve -problemin tipine göre değişmekle birlikte- daha iyi CPU zamanları elde etmiştir.

Pan ve ark. (2008) BATP için, C_{max} ' ı en küçükleyen, bir melez kesikli parçacık sürü optimizasyonu geliştirmişlerdir. Kesikli parçacık-sürü optimizasyonu algoritmasına toplam CPU zamanlarını azaltan bir yerel arama algoritması eklemiştir. Geliştirdikleri algoritmanın sonuçlarını kesikli parçacık-sürü algoritmasıyla Carlier, Reeves ve Heller problemleri için kıyaslamışlardır.

Pan ve ark. (2008) en fazla gecikme zamanı ve en erken tamamlanma zamanlı BATP'in çözümü için bir çok amaçlı kesikli parçacık optimizasyon tekniği önermiştir. Öncelikle çalışmada iş permütasyonları gibi parçacıklar tekrar ele alınmış ve pareto optimizasyonu kavramı, çok amaçlı optimizasyonun farklı çözümlerinin karşılaştırılmasında uygulanmıştır. Belirtilen baskın olmayan çözümlerin tutulması ve güncellenmesi sağlanmıştır. İkinci olarak PWQ (Pan, Wang, Qian) olarak adlandırılan, yeni bir çok amaçlı sezgisel, iş permütasyonlarının üretilmesinde kullanılmış ve nispeten iyi bir performans göstermiştir. Üçüncü olarak basit fakat verimli bir çok amaçlı yerel arama algoritması, global tarama ve yerel faydalanma

arasındaki dengeyi kurmak için geliştirilmiştir. Ek olarak iki hızlandırıcı yöntem, iş sıralamalarını değerlendirmek ve onun nispi komşuluklarını eklemek için tasarlanmıştır. Geliştirilen algoritma Carlier, Heller ve Reeves problemleri üzerinde denemişler, daha önce geliştirilmiş olan çok amaçlı türevsel evrim algoritmasına göre arama kalitesi, çeşitlilik seviyesi gibi kriterler bakımından daha üstün olduğunu tespit etmişlerdir.

Moghaddam ve ark. (2008) BATP için ağırlıklandırılmış ortalama C_{\max} ve T_{\max} değerlerini en azlayan bir yapay bağışıklık sistemi algoritması önermişlerdir. Önerdikleri algoritmayı, güçlü pareto evrimsel algoritmasıyla karşılaştırmışlardır. Her algoritmanın bulduğu baskın olmayan çözüm sayısı, algoritma tarafından bulunan baskın olmayan çözümlerin bir arada tutulması ve çözümlerin birbirlerinden farklılığı gibi kriterler bakımından karşılaştırılmış ve çok amaçlı yapay bağışıklık sistemi algoritmasının iyi bir performans gösterdiği saptanmıştır.

Lee ve ark. (2008) BATP için bir türevsel, evrim tabanlı, memetik algoritma önermişlerdir. Öncelikle, akış tipi problemlere türevsel evrim algoritmasının uygulanabilmesi için iş sıralamalarının gerçek vektörleri, türevsel evrim algoritmasının bireyleri haline dönüştürülmüştür. İkinci olarak türevsel evrim tabanlı paralel evrim mekanizması ve etkin bir arama sağlamak için geliştirilen çeşitli yerel arama algoritmaları, çok amaçlı ATP'lere uygulanmıştır. Üçüncü olarak hesaplamayı hızlandıran bir yöntem BATP'ler için geliştirilmiştir. Carlier, Reeves, Heller, Taillard's problemleri üzerinde I_{SUM} (Makinelerin toplam boş kalma süresi) ve N_T (Geciken işlerin toplam sayısı) ile C_{\max} ve T_{\max} amaçları kombine edilerek; çok amaçlı memetik algoritmayla, aynı işlem süreleri için toplam baskın olmayan vektör

üretimi, toplam baskın olmayan çözüm sayısı, referans çözüme olan uzaklık, baskın olmayan çözüm oranı gibi kriterler kıyaslanmıştır.

Javadi ve ark. (2008) BATP'in çizelgelenmesi için bulanık çok amaçlı doğrusal programlama yöntemi geliştirmişlerdir. Önerdikleri model, ortalama tamamlanma zamanlarını, ağırlıklandırılmış ortalama tamamlanma zamanını ve ağırlıklandırılmış ortalama en erken başlama zamanını en küçüklemektedir. Amaçlardan her birinin tek başına dikkate alındığı 2 senaryo ile amaçlardan birinin sayısal örneklerdeki orijinal değerlerini aldığı, diğerinin ise bulanık olarak ayarlandığı 2 senaryodan oluşan, toplam 4 senaryo üzerinde çalışmışlardır.

Vahed ve ark. (2008) BATP'de ağırlıklandırılmış ortalama tamamlanma zamanı ve ağırlıklandırılmış ortalama gecikme sürelerini en küçükleyen, bir çok amaçlı yayılarak arama algoritması geliştirmişlerdir. Geliştirdikleri algoritmayı, Ziltler ve ark.'nın geliştirdikleri güçlü pareto evrimsel algoritması ile karşılaştırmışlardır. Optimal pareto çözümlerinin sayısı, hata oranı, en iyi çözüme olan uzaklık gibi kıstaslar için algoritmaları karşılaştırmışlar ve büyük çaptaki problemlerin çözümünde yayılarak arama algoritmasının iyi sonuç verdiğini gözlemişlerdir.

Filho ve ark. (2007) BATP' de F_i 'yi en küçüklemek için bir melez evrimsel algoritma geliştirmişlerdir. Geliştirdikleri algoritmayı Taillard's problemleri için, Olivera ve Lorana tarafından geliştirilmiş olan evrimsel kümelenendirme algoritması ile kıyaslamışlar; -50 iş ve 5 makineli bir problem hariç- 59 problem için daha düşük F_i değerleri elde etmişlerdir.

Kalczynski ve ark. (2007) beklemesiz ve boş durmasız akış tipi problemlerde, C_{max} 'ı küçüklemeyi hedeflemişlerdir. Beklemesiz ve boş durmasız olma şartlarını,

Elmaghraby ve Kamburowski tarafından geliştirilen genelleştirilmiş öncelik ilişkilerinin olduğu ağ aktivitesi ile modellemiştir. Her iki ağdaki kritik yol uzunlukları beklemez şartla, C_{max} 'ı ve boş durmasız şartla, C_{max} 'ı vermektedir. Beklemez ve boş kalmasız şartlar altında C_{max} 'ın en küçüklenmesi, bazı işlerin işlem sürelerini düşürebilmiştir. Ayrıca problem bu haliyle, beklemez akış tipinden gezgin satıcı problemine dönüştürülebilmektedir.

Liu ve ark. (2007) BATP'nin çözümü için bir melez parçacık-sürü optimizasyon algoritması önermişlerdir. Buna göre Nawaz-Enscore-Ham (NEH) sezgisel yönteminde olduğu gibi meta-Lamarckian öğrenme stratejili tavlama benzetimini yerel arama yöntemi olarak, parçacık-sürü algoritmasına eklemiştir. C_{max} 'ı en küçükmeyi hedefleyen algoritma, Carlier ve Heller problemleri için, Tasgetiren ve ark.'nın önerdiği değişken komşuluk arama tabanlı parçacık sürü algoritması ve Wang ve arkadaşlarının geliştirdiği sezgisel bir algoritma ile kıyaslanmış; nispi hata oranı, ortalama nispi hata oranı ve CPU süreleri bakımından iyi bir performans göstermiştir.

Ruiz ve ark. (2007) BATP'de işlem zamanlarından ayrı olarak düşünülmüş hazırlık zamanlarının bulunduğu durumlar için T_{max} 'ı en küçükleyen GA'lar, algoritmalarındaki baskınlık ilişkisi üzerine çalışmışlardır. Çalışmalarında öncelikle 5 farklı GA'yı, 3 makineli problemler üzerinde test etmişlerdir. (Bunlar parametrelerdeki değişiklikler ve yerel arama algoritmasının eklenip eklenmemesine durumlarına göre farklılık göstermektedir.) H5 (test sonuçlarına göre üretilen sezgisel) ve yenilemeli yerel arama algoritması ile üzerinde çalışılan rastgele yenilemeli yerel arama algoritmaları meta sezgiselleri, farklı hazırlık zamanları için diğer genetik algoritmalar ile kıyaslamışlardır. H5, NEH vb. literatürde iyi bilinen 7

sezgisel algoritmayı, 20 ile 100 arasında deęişen iş süreleri ve 5 ile 20 arasında deęişen makine sayılarının olduęu test problemleriyle, ortalama nispi oran daęılımları bakımından kıyaslamışlardır. H5 kendisine en fazla yaklaşan NEH algoritmasına göre çok kötü CPU zamanları vermesine rağmen çok düşük T_{max} deęerleri elde etmiştir.

Wang ve ark. (2006) TB'yi (tavlama benzetimi) ve GA'yı, 10 ile 300 arasında deęişen iş sayıları, 2 ve 3 makineli B ATP'in çizelgelenmesinde kullanarak bunları çizelgeleme verimlilięi bakımından karşılaştırmıştır. Karşılaştırma, benzer CPU zamanlarında, C_{max} 'ı en küçüklemek için yapılmıştır. TB aynı C_{max} deęeri için GA'ya göre daha iyi CPU zamanları vermiştir. Özellikle büyük çaptaki problemlerde GA'nın CPU sürelerinde hızlı bir artış olduğunu gözlemlemişlerdir.

Kubzin ve ark. (2005) 2 makineli B ATP'in çizelgelenmesinde makinelerden birisinin zorunlu olarak bakımının yapıldığı durumlar üzerinde çalışmışlardır. C_{max} 'ı en küçüklemeyi hedefleyen algoritmada, bakım periyotunun uzunluęu, bakımın başlangıç zamanına göre deęişen ve artmayan bir fonksiyon olarak tanımlanmıştır.

Grabowski ve ark. (2005) B ATP'in çözümü için C_{max} 'ı en azlayan farklı yerel arama yöntemleri (azalan arama algoritması, çok amaçlı azalan arama algoritması ve çok işlemcili tabu arama algoritması) geliştirmişler ve bunları karşılaştırmışlardır. Geliştirilen yerel arama algoritmalarında, çoklu işlemler tek bir iterasyonda yapılarak, iyi bir çözüme yakınsama durumu hızlandırılmıştır. Ayrıca tabu arama algoritmalarında yerel bir optimuma takılmayı engellemek için dinamik tabu listeleri oluşturmuşlardır. Kodlanan yerel arama algoritmaları, bazı Reeves ve Heller problemleri üzerinde test edilmiş; azalan arama algoritması ve tabu arama

algoritmasına göre, çok işlemcili yapıdaki azalan arama algoritması ve tabu arama algoritması daha düşük CPU süreleri elde etmişlerdir.

Spieksma ve ark. (2005) B ATP’de “beklemesiz akış tipi paradoksu” isimli çalışmasında, işlem sürelerinin kısalmasının –eğer bu kısalma diğer işlem süreleriyle birlikte düzgün bir dağılım göstermiyorsa– C_{max} ’ın daha büyük bir değere almasına neden olabileceğini göstermiştir.

Allahverdi ve ark. (2004) ilk kez T_{max} ile T_{max} ve C_{max} ’ın birlikte en azlandığı B ATP için sezgisel geliştirmişlerdir. Melez TB ve melez GA, T_{max} veya C_{max} ’ı en küçükmek ve her ikisini birlikte en küçükmek için kullanılmıştır. Ayrıca baskınlık ilişkisi ve dal sınır algoritması iki makineli problemler için geliştirilmiş olup, sonuçları önerilen sezgisellerle karşılaştırılmış, farkı büyüklükteki problemler algoritma yardımıyla çözülmüştür. C_{max} ’ın en küçüklendiği problemler için klasik GA ve TB’ye göre önerilen GA ve TB’nin daha iyi uygunluk değerlerine sahip popülasyon ürettiği gözlemlenmiştir.

Espinouse ve ark. (2001) sınırlı sayıda makinenin kullanılabilir olduğu durumlarda, iki makineli beklemesiz akış tipi problemlerin çözümü için yaklaşımsal algoritmalar ve bunların karmaşıklık seviyeleri üzerinde çalışmışlardır. Makinelerin yalnız birinin (önleyici bakım vb. sebeplerden dolayı) çalışmaya elverişli olmayacağını kabul etmişler ve C_{max} ’ı en küçükmeyi hedeflemişlerdir. Buna göre iki makineli B ATP’nin çizelgelenmesinde herhangi bir dönemin uygun olmadığı durumlarda problem NP-Zor (Polinom olarak ifade edilemeyen zor problemler) olmaktadır.

Sidney ve ark. (2000) makinelerin hazırlık zamanlarını ikiye ayırmıştır. Hazırlık zamanının ilk bölümünde, makine iş gelmeden önce yapılmak zorunda

olduğu (ölçümleme veya bir aletin ayarlanması gibi), ikinci hazırlık kısmında ise iş makineye gelmiş ya da gelmemiş olmasının önemli olmadığı 2 makineli beklemesiz akış tipi süreçler için, bir sezgisel geliştirmişlerdir. Geliştirdikleri sezgisel yöntemin C_{max} değerleri için en kötü durum analizi yapmışlardır. Bunun için olası bir C_{max} süresinin, algoritma ile belirlenmiş C_{max} süresinin $4/3$ ' ünden fazla olmayacağını kabul etmişlerdir.

Glass ve ark. (1999) bazı işlerin yalnızca ilk makinelerde işlendiği durumlarda, iki makineli BATP için Gilmore ve Gomory algoritmasını analiz etmişlerdir. Buna göre iki makineli eksik operasyonlu BATP, NP-Zor problemler olsa da, eksik operasyonla ilgili kısım dönüştürülerek polinom haline getirilebilir. Ayrıca problemlerde eksik operasyonların sürelerine "0" atandığında sezgisel yöntemlerle çözülebilecek hale gelir.

Espinouse ve ark. (1999) iki makineli BATP için C_{max} 'ın en küçüklendiği ve sınırlı sayıda makinenin uygun olduğu durumlar (bakım, bazı işlem öncesi hazırlıklar vb.) üzerinde çalışmışlardır. Ayrıca problemlerdeki hata sınırı analizleri için bir sezgisel geliştirmişlerdir.

Bianco ve ark. (1999) değişken hazırlık zamanının olduğu BATP'in; ziyaret zamanlarının dâhil edildiği asimetrik gezgin satıcı problemleri olarak ele alınabileceğini göstermiştir. Rastgele üretilmiş, 2 ile 10 makine ve 50 ile 700 iş arasında değişen problem büyüklükleri için iki farklı sezgisel önermişlerdir. C_{max} 'ı en küçüklemeyi hedefleyen sezgiselleri CPU zamanı ve en iyi C_{max} değerleri bakımından kıyaslamışlardır.

Agnetis (1997) büyük öbek büyüklüğüne sahip beklemesiz akış tipi problemler üzerinde çalışmışlardır. Makine sayısı 2'den büyük olan problemlerin, NP-Tam

(Karmaşıklığı arttığında hesaplama süresi de sınırlanmamış biçimde artan ve polinom olarak ifade edilemeyen problemler) olduğunu belirtmiş ve bunun çözümü için bir sezgisel önermiştir. Algoritmayı taşıma problemleri tabanlı olarak geliştirmişler ve problem büyüdükçe algoritmanın en iyi çözüme yakınsadığı gözlemişlerdir. Geliştirilen algoritmayı, rastgele üretilen örnekler üzerinde, Rock ve Schmidts'in algoritmasıyla kıyaslamışlar ve özellikle büyük çaptaki problemlerde daha iyi sonuç elde etmişlerdir.

Emmons ve ark. (1995) B ATP'de öbek büyüklüğü konusunda çalışmışlardır. Büyük ve değişken parti büyüklüklerinin olduğu, sık tekrarlanan basit parti tipi işlerin gezgin satıcı problemi gibi modellenebileceğini göstermiştir.

Strusevich (1995) iki makineli B ATP üzerinde çalışmıştır. Makine hızlarının değişebildiği fakat makineleri hızlı çalıştırmanın maliyetli olduğu varsayımıyla, modele bir ceza fonksiyonu ekleyerek, C_{max} 'ı en küçüklemeyi hedefleyen bir algoritma geliştirmiştir.

2.2 Çok Amaçlı Genetik Algoritma (ÇAGA) İçin Kaynak Araştırması

Zandieh ve ark. (2010) dizi bağımlı hazırlık zamanları olan melez esnek akış tipi problemlerinde, çok amaçlı grup çizelgelerinin oluşturulması için uyarlanmış, çoklu popülasyona sahip bir GA geliştirmişlerdir. C_{max} ve toplam ağırlıklandırılmış gecikme zamanını en küçüklemeyi hedefleyen algoritma, iki aşamadan oluşmaktadır. İlk aşama amaçların kombine edilmesi, ikinci aşama ise ilk aşamadaki çözümlerin kullanılarak alt popülasyonların üretilmesi ile ilgilidir. Geliştirilen algoritma ÇAGA

ve BOSGA 2 (baskın olmayan sıralama yapılı genetik algoritma 2) ile kıyaslamışlar ve daha iyi çözümler elde etmişlerdir.

Kachitvichyanukul ve ark. (2009) iş istasyonları çizelgeleme problemleri için iki aşamalı bir ÇAGA geliştirmişlerdir. Algoritma C_{max} , toplam ağırlıklandırılmış gecikme zamanı kriterlerini en küçüklemeyi hedeflemektedir. İlk aşama, her bir amaç fonksiyonu için GA uygulamakta ve sonuçları popülasyona kaydetmektedir. İkinci aşama, amaç fonksiyonları ağırlıklarının kararlı durumdaki GA'dan toplanmasıyla oluşturulan evrim sürecinden oluşmaktadır. Geliştirilen algoritma her bir amaç için TAA, karınca kolonileri algoritması, parçacık sürü optimizasyonu algoritması, klasik GA gibi sezgisellerle kıyaslanmıştır. C_{max} için kıyaslandığında parçacık sürü optimizasyonu ve karınca kolonileri algoritmasından daha iyi toplam çözüm kalitesi elde edilmiş; çok amaçlı olarak da, çok aşamalı GA'dan orta büyüklükte ve büyük çaptaki problemler için, daha verimli sonuçlar elde edildiği gözlenmiştir. Ancak geliştirilen iki aşamalı GA'nın hesaplama süresi daha uzundur.

Rodriguez ve ark. (2009) çok amaçlı iş sıralama problemleri için yeni bir yönlendirme kuralı yapılı ÇAGA geliştirmişlerdir. Geliştirdikleri algorithmada BOSGA 2'nin seçim mekanizmasını kullanmışlardır.

Asiabar ve ark. (2009) YSA'dan etkilenererek oluşturulan "kendisini düzenleyen harita" olarak adlandırdıkları öğrenme yöntemini, değişken komşuluk arama algoritmasıyla birleştirerek, yerel arama algoritmalı ÇAGA geliştirmişlerdir. Geliştirdikleri algoritmayı, barajlardan elde edilecek su miktarı ve elektrik üretimi amaçlarını en iyilemek için kullanmışlardır. Sonuçları BOSGA 2 ile kıyaslamışlardır. Geliştirilen algoritma ek olarak, yerel arama ve öğrenme yöntemleri kullandığından,

CPU zamanları BOSGA 2'den fazladır. Ancak çözümün performansı bakımından oldukça üstün olduğu belirlenmiştir.

Blecic ve ark. (2007) baskın olmayan çözüm dizilerinin kullanıldığı, bayesian çok amaçlı karar modelleri analizi için, ÇAGA'dan yararlanmışlardır. Buna göre karar stratejileri değişiminin analizinde her bir strateji ikili sistemde kodlanmış bir kromozom ile ifade edilmiştir.

Zheng ve ark. (2007) bir çok amaçlı, sıklık yapılı GA önermişlerdir. Etki ve yoğunluk fonksiyonlarının analizi sonucunda; çözüm noktalarının yoğunluğu ile bu çözümlerin uygunluk değerlerinin, GA'nın çeşitliliğini artıracak biçimde kullanılabileceğini göstermişlerdir. Çözüm aralıkları, CPU zamanları, bireyler arasındaki uzaklıklar gibi kriterler için BOSGA 2 ve "Güçlü Pareto Evrimsel Algoritması 2" ile çok amaçlı sıklık yapılı GA'yı kıyaslamışlar ve daha iyi sonuç elde etmişlerdir.

Kabayashi ve ark. (2007) GA'lardaki çözüm çeşitliliği ile ilgi çalışmışlardır. Başlangıç popülasyonunun büyük olduğu problemlerde, hesaplama süresi uzamakta, küçük olduğunda ise yeterli çeşitlilik sağlanamamakta ve çözümler yerel bir optimuma yakınsamaktadır. Bu sorunu gidermek için YSA algoritmasından yararlanmışlardır. Önerilen ÇAGA, BOSGA 2'nin içerisine YSA algoritmasının yerel bir optimuma yakınsamayı engellemek için kullanılmasından oluşmaktadır. Kabayashi ve ark. daha az sayıda evrimsel fonksiyon çağırarak BOSGA 2'den daha iyi popülasyon çeşitliliği elde etmişlerdir.

Lei ve ark. (2006) hem özel sektör hem de kamu sektörleri tarafından paylaşılan, merkezi transferler ve vergilerdeki kayıplar gibi kısıtlara sahip olan, verimlilik ve eşitlik amaçlarının olduğu bir çok amaçlı, çok kısıtlı halk yatırım

modeli önermişlerdir. Bu model için ÇAGA'dan yararlanarak bölgesel ve sektörel yatırım analizleri yapmışlardır.

Chen ve ark. (2006) yap, işlet ve devret tipi otoyol tasarım problemleri için bir benzetim yapılı, ÇAGA geliştirmişlerdir. Bunun için GA'ya olasılıklı benzetim tekniği, trafik atama algoritması ve mesafe tabanlı arama yöntemi eklenmiştir. Otoyol tasarım probleminin iki seviyeli olasılıklı matematiksel model olarak ele alındığı çalışmada, talebin belirsiz olduğu durumlar için en uygun yol ücreti ve en uygun yol kapasitesi amaçlar olarak belirlenmiştir. Algoritma baskın olmayan çözümlerin kalitesi bakımından ağırlıklı ortalama yöntemine göre daha iyi sonuçlar vermiştir.

Luque ve ark. (2006) salt metinden oluşan bir çevrede sözel sabit sorguların modellenmesi için ÇAGA kullanmışlardır. Mantıksal operatörlerin kullanıldığı modelde sözel terimler bir kromozomun genleri biçiminde ele alınmıştır.

Pasupathy ve ark. (2006) ATP'ler için bir çok amaçlı (C_{max} ve F_i) pareto derecelendirme yapılı ve yerel arama algoritmalı, GA geliştirmişlerdir. Önerilen algoritmayı NEH gibi, literatürde sıklıkla kullanılan farklı sezgisellerle kıyaslamışlar ve algoritmanın baskın olmayan çözümlerin bulunmasında daha iyi sonuç verdiğini tespit etmişlerdir.

Yun ve ark. (2005) her bir bireye uygunluk değerinin nasıl atanacağı, bireyler arası çeşitliliğin nasıl sağlanacağı konuları üzerine çalışmışlar ve yeni bir ÇAGA geliştirmişlerdir. Yeni ÇAGA genelleştirilmiş veri geliştirme analizi kullanmakta ve az sayıda evrimsel fonksiyon çağırarak ortalama pareto cephesi kadar iyi dağılmış sonuçlar üretmektedir. Yun ve ark. (2005) geliştirdikleri algoritmayı BOSGA gibi sezgisellerle kıyaslamış ve daha iyi bir çözüm dağılımı elde etmişlerdir.

Khoo ve ark. (2000) çok amaçlı üretim çizelgeleme problemleri için çok amaçlı bir GA prototipi geliştirmişlerdir. Buna göre; C_{max} ve T_{max} 'ın en küçüklemeyi amaçlayan algoritma, akış tipi, iş istasyonu biçiminde ve hücre tipi üretim sistemleri ile bunların kombine edildiği üretim sistemlerinin çizelgelenmesinde uygun bir araçtır. Uygun kısıtlamalar ve C_{max} , T_{max} amaçları ile farklı sistemlere adapte edilebilen algoritma optimale yakın sonuç vermektedir. Khoo ve ark. çözümü sembolize eden kromozomların göreceli uygunluk değerlerini sonraki kuşakların uygunluk değerlerinin iyileştirilmesi için kullanmışlardır. Geliştirilen prototip ÇAGA; Nokano, Fang ve Yamada'nın geliştirmiş olduğu sezgisellerle kıyaslanmış, daha küçük popülasyon büyüklüğüyle benzer amaç fonksiyonu değerleri vermiştir.

3. MATERYAL VE METOD

Bu çalışmada, BATP materyal, çok amaçlı melez GA ise metot olarak belirlenmiştir.

3.1. BATP

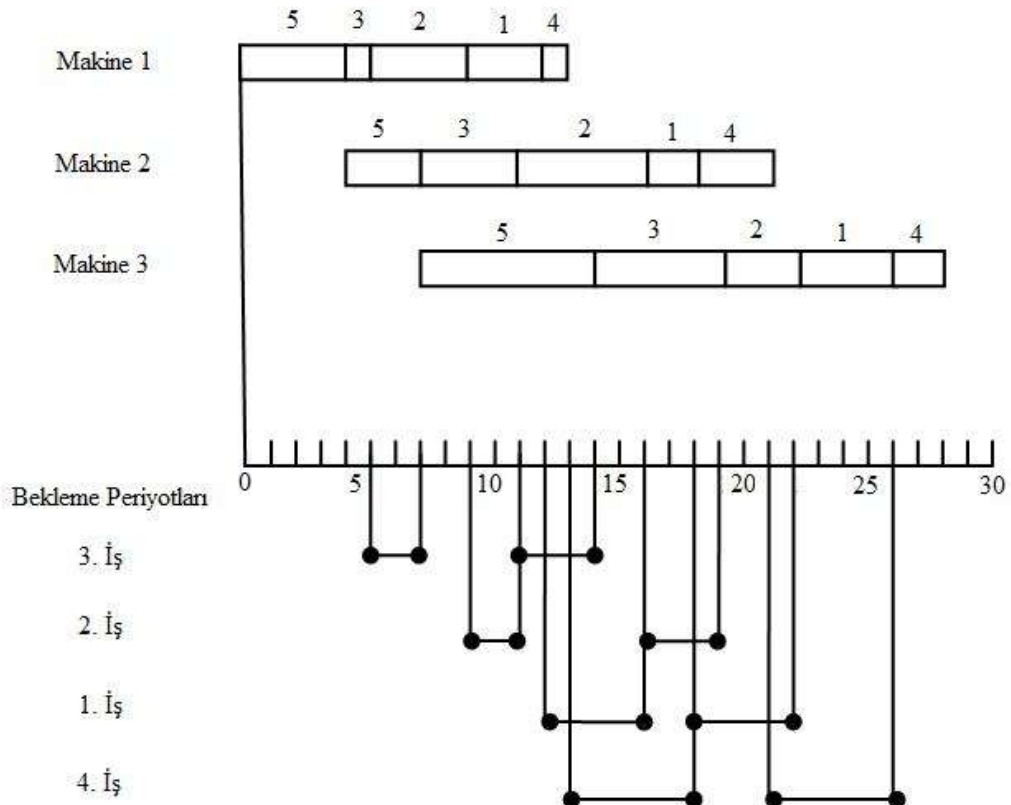
3.1.1. Akış tipi problemlerin (ATP) temel yapısı

ATP literatürde çok kez ele alınmış bir konudur. Bu problemleri çözebilmek için çok sayıda algoritma geliştirilmiştir ve hemen hepsinde benzer kabuller, akış tipi problem dizileri ile ilişkilendirilmiştir. Bu kabullerden birisi de bir sonraki makine meşgul olduğundan işlem aralarındaki stok miktarlarının var olduğudur. Bundan başka, problemlerle ilişkilendirilmiş iki varsayımda, n işin m makine de işlem gördüğü durumlarda, tüm işler tüm makinelerde işlenmektedir ve işlem süreleri bilinmektedir. Buradaki problem n işin tamamlanma zamanının tüm makineler üzerinde en küçüklenmesidir. Şimdi beş işin ve üç makinenin olduğu ve her işin her makine üzerindeki işlem sürelerin farklı olduğu bir örnek düşünelim. Eğer her bir makinedeki iş dizileri (5,3,2,1,4) biçiminde Şekil 3.1'deki gibi gant şemasında gösterilirse, 3 numaralı iş, 5. ile 7. periyotlar ve 11. ile 14. Periyotlarda, 2. ve 3. makineler meşgul olduğundan bekler. 2., 1. ve 4. işler benzer şekilde makineler meşgul olduğundan beklerler. Gant şemasında görüldüğü gibi bir çizelge yaptığımızda en az 3 iş için yarı mamul stoklama alanına ihtiyaç vardır. (13. ve 14.

periyotta 3 işin aynı anda sürece dâhil olmak için beklediğini görmekteyiz.)
(Ramamoorthy ve ark. 1971)

Tablo 3.1 5 iş 3 makine için işlem süreleri tablosu

M a k i n e	İş				
	1.	2.	3.	4.	5.
1.	3	4	1	1	4
2.	2	5	4	3	3
3.	4	3	5	2	7



Şekil 3.1 (5,3,2,1,4) iş sıralaması için gant şeması ve farklı işler için bekleme süreleri

Bu tür iş sıralama problemlerine, herhangi bir yarı mamul stoklamayla ilişkili olmaksızın, literatürde ve endüstride, üretim çizelgelemede sıkça rastlanmaktadır. Sonsuz yarı mamul stokları varsayımı kabul edilemez bir kavram olmayabilir. Ancak akış tipi sıralama, bilgisayar sistemlerine uygulandığında bu varsayım, yarı mamul stokunun sonlu olabileceği yönündedir.

Akış tipi sıralama konseptinin bilgisayar sistemlerine uygulanmasını aşağıdaki gibi açıklayabiliriz: Hesaplama hızını artırmak ve kaynaklardan yararlanmak için, günümüz bilgisayarlarının çoğu, bir boru hattı ya da montaj hattına benzer bir yapıdadır. Bilgisayarın farklı işlevsel üniteleri eş zamanlı ve bağımsız işlemler yapar. Bu sayede bilgisayarlar birçok işi aynı zaman diliminde yürütebilirler. Bilgisayarın fonksiyonel üniteleri olan CPU, (I/O) (giriş-çıkış kontrolleri) ve belleğin çalışma biçimi buna bir örnektir. Bu bilgisayar çoklu programlanmıştır ve birbirlerinden bağımsız programların (işlerin) bir akışını icra ettiğini düşünebiliriz. Her program belleğin, CPU'nun ve (I/O)'nun belirli bir miktar kullanımına ihtiyaç duyar. (Bu problemde CPU'nun ve (I/O)'nun yeterli miktarda kendisine ait belleği olduğunu ve kendi içinde bellekten bağımsız olarak çalıştığını varsayıyoruz). Burada; yürütülen tüm programların tamamlanma zamanını en aza indirmeye çalıştığımızda, asıl olarak bir akış tipi problem dizisini görmekteyiz. Bu şekilde bilgisayarın sisteminin sonsuz yarı mamul stoklamasını canlandırabiliriz. Farklı fonksiyonel üniteler üzerindeki iş süreleri Tablo 3.1'deki gibi olan 5 tane iş olduğunu ve Tablo 3.1'deki makinelerinde bellek, CPU ve (I/O) olduğunu farz edelim. Her makinedeki (5,3,2,1,4) iş sıralamaları için, sonsuz yarı mamul stok kapasitesi içerisinde en az 3 üniteye sahip olunmalıdır. (Burada fazlaca basitleştirdiğimiz depolama gereksinimi işlerin doğasına göre ve fonksiyonel ünitelerin hangi iş için beklediğine göre değişir.)

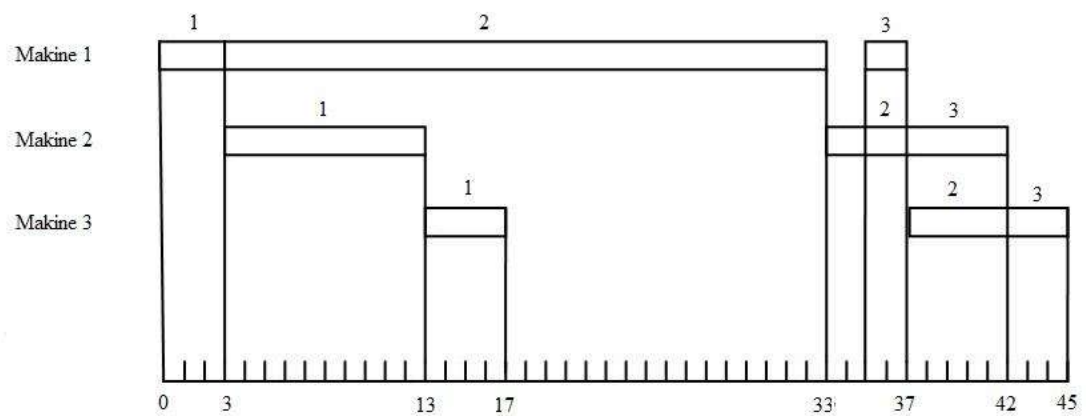
Örneğin 2. işin depolama gereksinimi CPU için beklerken, (I/O) için beklediğinden farklı olabilir ve 3 numaralı işin gereksinimleri tamamıyla farklı olabilir (ancak örnek aslında yarı mamul stoklama da etkisini gösterir ve bundan dolayı her işin -beklenen fonksiyonel üniteden bağımsız olarak- bir birim depolama alanına ihtiyaç duyduğunu kabul ederiz). Genellikle, stoklama alanını bir tampon bölge gibi kabul etmek pratikte maliyetlidir. Fakat problem yeterli stoklama şartıyla durmaz. Burada ayrıca bekleyen işlerin uygun biçimde rotalanması problemi vardır. Örneğin, 14. zamanda yalnızca 3. işin, -1. ve 4. işlerin değil- yarı mamul stokundan alınması ve uygun bir biçimde (I/O)'ya gönderilmesi gerekmektedir. Bu problem bir denetleme problemi gibidir ve problem hacmi büyüdüğünde, problem çok karmaşık bir hal alır. (Ramamoorthy ve ark., 1971). Bilgisayarlardaki akış tipi dizilerle ilgili çeşitli örnekler vermişlerdir.

Yarı mamul stoku olmaksızın akış tipi problemler bilgisayar sistemleri için önemlidir. Çünkü sıfır olmayan sonlu yarı mamul stokuna doğru bir adımdır. Geniş hacimli bilgisayar ağlarında büyük tampon bölgeler pratikte var olmadığından ve karışık denetleme sistemlerinin elenerek bekleyen işler uygun biçimde rotalandığı için, oldukça faydalıdır. Sonlu yarı mamul stoklu akış tipi problem dizilerini çözmek için farklı teknikler mevcuttur. Ancak sonlu yarı mamul stoklu akış tipi problemler, yarı mamul stoksuz problemler gibi olmak zorunda değildir ve bu durum 3 iş ve 3 makinenin işlem sürelerinin olduğu Tablo 3.2 ile açıklanmaktadır. Eğer problem sonlu yarı mamul stoklu akış tipi problem biçiminde ise optimal çizelge toplam tamamlanma zamanını en küçüklediğinde (1,3,2) ya da (3,1,2) biçiminde olur ve tamamlanma zamanı 44 birimdir.

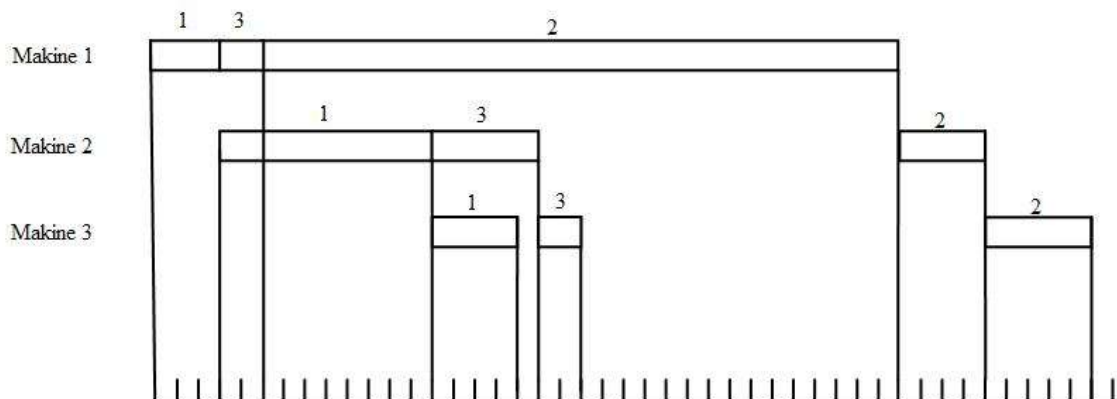
Tablo 3.2 3 iş 3 makine için örnek problem

M a k i n e	İş		
		1	2
1	3	30	2
2	10	4	5
3	4	5	3

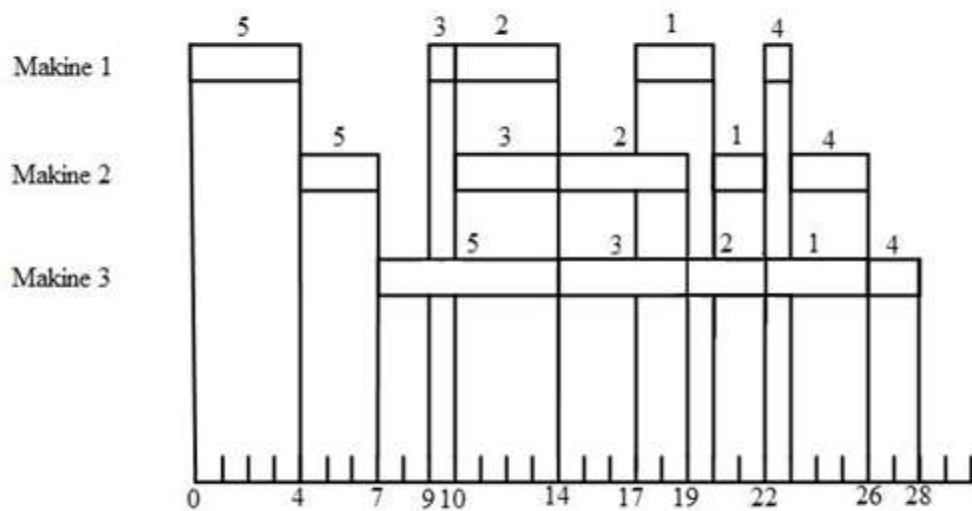
Eğer problem yarı mamul stoksuz akış tipi ise, Şekil 3.2’de görüldüğü gibi, optimal çizelge (1,2,3) ve tamamlanma zamanı 45 birimdir. Bu yapı için, (1,3,2) ve (3,1,2) çizelgelerinin tamamlanma zamanı sırasıyla 52 ve 46’dır. (Pan ve ark., 2008)



Şekil 3.2 Tablo 2’ deki yarı mamul stoksuz akış tipi problem için (1,2,3) çizelgesinin gant şemasıyla gösterimi



Şekil 3.3 Tablo 2' deki yarı mamul stoksuz akış tipi problem için (1,3,2) optimal çizelgesinin gant şemasıyla gösterimi



Şekil 3.4 Yarı mamul stoksuz akış tipi problemler için (5,3,2,1,4) sıralaması

(Çizelge: 5/0, 3/9, 2/10, 1/17, 4/22'dir.).

Yarı mamul stoksuz akış problemi, Tablo 3.1 'deki örnek üzerinde (5,3,2,1,4) iş sıralamasıyla, Şekil 3.4'te gösterildi. En az tamamlanma zamanı, işlerin beklememe koşuluyla, 28 olarak bulundu. Yarı mamul stoksuz akış tipi problemlerin çözümü, işlerin bir sıralamasından oluşmaktadır ve işler süreç için beklemediğinden, hangi makinelerin önce başlayacağına dair en uygun karar verildiğinde, tüm işler için tamamlanma zamanı en az olur. Bir başka deyişle çözüm; işlerin yeniden sıralandığı bir permütasyon ve bir n adet zaman veri seti ile onların içeriği olur; ilk makineden başlayacak olan işlerin bir permütasyonunu verir. Örneğin 3 makine ve 3 işin olduğu yarı mamul stoksuz akış tipi problemleri çözümü (1,2,3) sıralamasıyla Tablo 3.2'de ve Şekil 3.3'de gösterilmişti. Zaman veri seti ile birlikte çizelge (1/0, 2/3, 3/35) olarak sunulabilir.

Çözümde, her makine için sıralamanın aynı olması gerekmektedir. Aksi halde süreç beklemeli bir hale dönüşecektir. Böylelikle $n!$ adet iş sıralamasına ihtiyaç olacağı düşünülebilir (Reddi ve ark. 1972).

3.1.2. BATP'lerin temel yapısı

n işli ve m makineli BATP aşağıdaki gibi tanımlanabilir: i işinin j makinesi üzerindeki işlem süresi $p(i,j)$ sırasıyla makine 1,2,..., m üzerinde işlem görür. Herhangi bir zamanda, her bir makinede, en fazla bir iş işlem görebilir. Her bir makine için işlerin işlem görme sırası aynıdır. Beklemesiz kısıdından yola çıkarsak, bir işin son operasyonunun tamamlanma zamanı ile ilk operasyonunun başlangıç zamanı arasındaki fark, o işin operasyonlarının tamamlanma zamanına eşittir. Diğer bir deyişle her işin her operasyonu, ardışık makineler arasında herhangi bir engel

olmaksızın işlem görmelidir. Problem, tüm işlerin tüm makineler üzerindeki süreçlerinin bir sıralama ya da dizisini bularak, verilen bir veya fazla sayıda amacı optimize etmektir. En yaygın kullanılan amaç, en fazla tamamlanma zamanını (C_{\max}) en küçüklemeektir. $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, işlem gören işlerin sıralaması; $P_{sum}(\pi_j)$, π_j işinin tüm makinelerdeki toplam işlem süresi; $MD(\pi_{j-1}, \pi_j)$ bekleme süresi altında ve π_j ile π_{j-1} 'in başlangıçları arasındaki ilk makine üzerinde en az ertelenme olsun. MD aşağıdaki gibi hesaplanır (Qian ve ark., 2009):

$$MD(\pi_{j-1}, \pi_j) = p(\pi_{j-1}, l) + \max\{0, \max_{2 \leq k \leq m} \{\sum_{h=2}^k p(\pi_{j-1}, h) - \sum_{h=1}^{k-1} p(\pi_j, h)\}\} \quad (3.1)$$

Böylelikle C_{\max} aşağıdaki gibi tanımlanabilir.

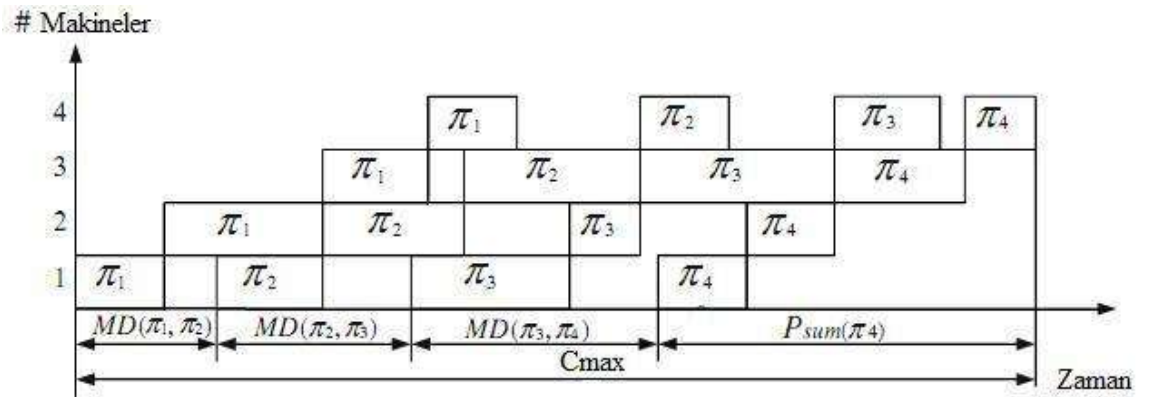
$$C_{\max}(\pi) = \sum_{j=2}^n MD(\pi_{j-1}, \pi_j) + P_{sum}(\pi_n) \quad (3.2)$$

Burada $P_{sum}(\pi_n) = \sum_{k=1}^m p(\pi_n, k)$ 'dir.

BATP'de C_{\max} hedeflediğinde tüm Π dizisinin içerisinde π^* 'in bir permütasyonu aşağıdaki gibi bulunur:

$$C_{\max}(\pi^*) = \min_{\pi \in \Pi} C_{\max}(\pi). \quad (3.3)$$

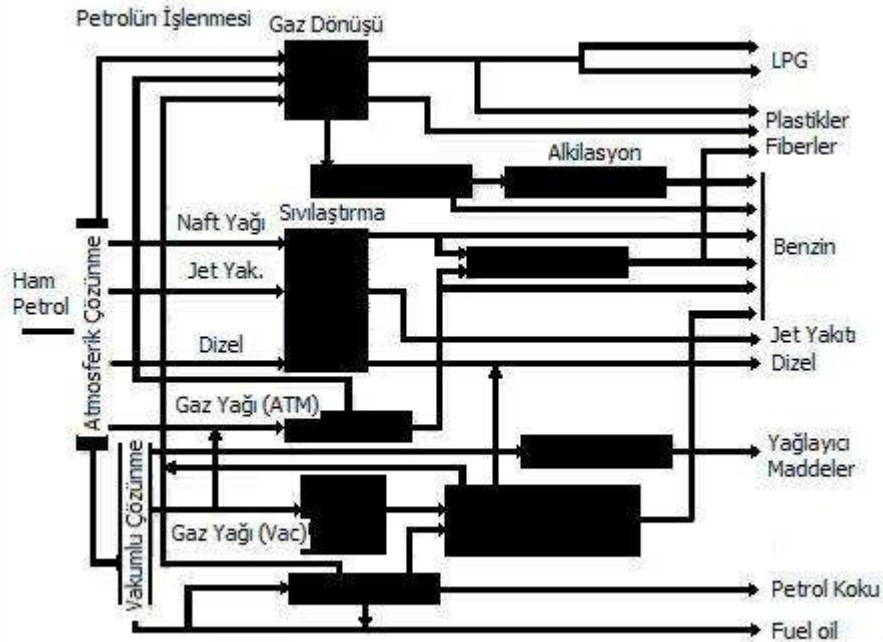
Şekil 3.5, BATP için iş sayısının ve makine sayısının 4 olduğu bir örnek göstermektedir.



Şekil 3.5 BATP örneği n=4, m=4 (Qian ve ark., 2009)

3.1.2.1. BATP'in uygulama alanları

Petrol ve Petrol Ürünleri: Şekil 3.6'da gösterildiği gibi petrol ürünlerinin hemen hepsi petrol rafinerilerinde damıtılmış petrolden oluşmaktadır. Rafineriler, ham petrolün kalitesine ve ihtiyaca göre petrol ürünleri üretirler. Petrol genellikle enerji kaynağı olarak değerlendirilir. Çeşitli derecelerde fuel oil ve benzine dönüştürülür. Rafineriler diğer kimyasal maddeleri de üretebilir; bunlardan bazıları, plastik veya diğer petrol ürünlerini elde etmek için kullanılır. Petrol sülfür içerdiği için, bu sülfürün büyük bir kısmı da yine petrol ürünü olarak ayrıştırılır. Hidrojen ve karbon da, petrol kolası formunda ürün olarak kullanılabilir. Hidrojen genellikle rafinerideki diğer işlemlerde katalizör olarak veya hidrodesülfürizasyonda kullanılır.

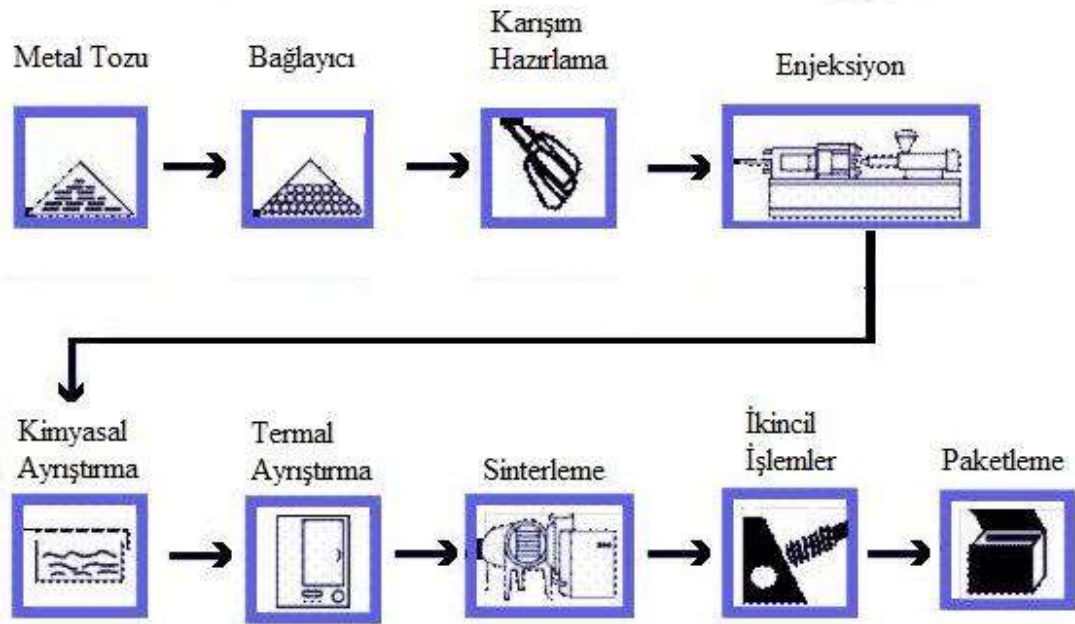


Şekil 3.6 Ham petrolün işleme süreci (www.exxonmobile.com)

- 150 °C'a kadar ham benzin,
- 150-250 °C'a kadar kerosin, jet yakıtı,
- 250-350 °C'a kadar dizel yakıtı,
- 350 °C'dan sonra da ağır yağlar elde edilir
(http://tr.wikipedia.org/wiki/Ana_Sayfa).

Ham petrolden farklı petrol ürünleri elde ederken, yarı ürünler, beklemeksizin, sürekli bir akış veya işlem halinde bulunurlar.

Sıcak Haddelme ve Sıcak Metal İşleme: Seri halde dizilmiş haddelerin içerisinden geçen metal şeritler her bir hadde silindiriyle biraz daha incelirler. Bu tip süreçlerde en fazla gider kalemi genellikle enerjidir. Metali erime sıcaklığına yakın bir sıcaklığa ya da erime sıcaklığına getirene kadar ısıtmak çok maliyetlidir. Bundan dolayı sıcak olarak uygulanan işlemler sona ermeden metalin soğuması istenmez. Bu durum da, bu tür süreçler çizelgelenirken, beklemesiz kısıdıyla modellenir. Şekil 3.7'de gösterilen, metal enjeksiyon kalıplama da bu süreçlerden birisidir (<http://estorototamiroda.com/makaleler/1.htm>).



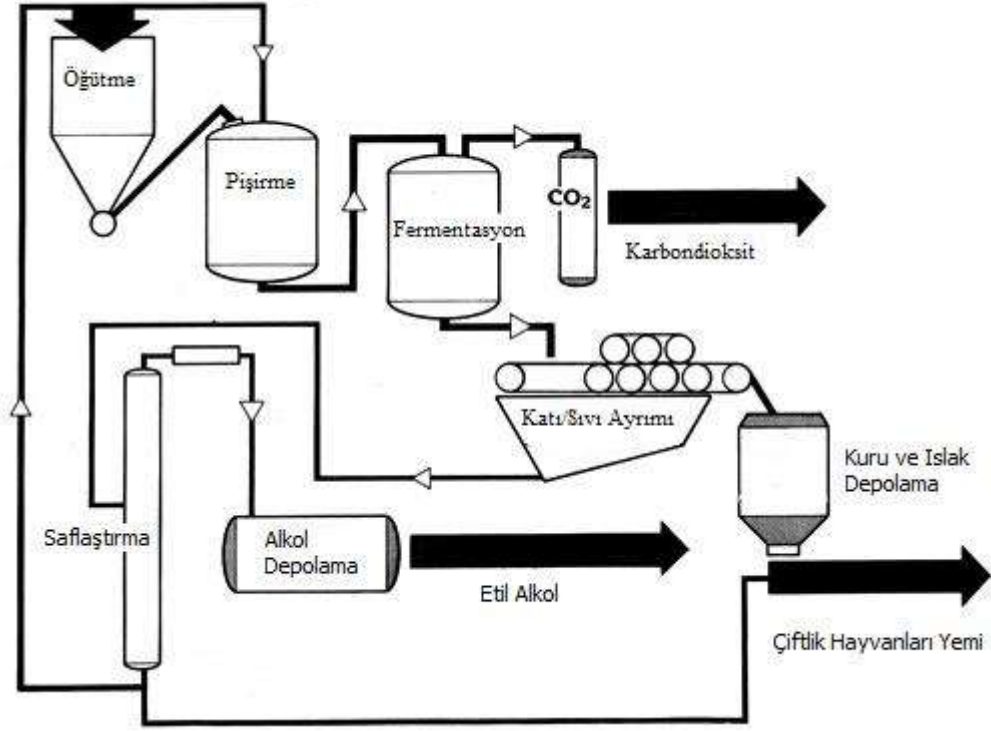
Şekil 3.7 Metal enjeksiyon kalıplama süreci

(<http://estorototamiroda.com/makaleler/1.htm>)

Metal enjeksiyon kalıplama ya da literatürde bilinen adıyla MIM teknolojisi özellikle son 15-20 yıl içerisinde alternatif bir imalat metodu olarak kendini göstermeye başlamıştır. Göreceli olarak küçük, hassas ve kompleks şekilli parçaların ekonomik biçimde üretilmesine imkân tanıyan MIM teknolojisi bugün itibarıyla tüm dünyada sıklıkla kullanılan bir üretim tekniğidir. MIM’de, metal parçaların soğumadan sonraki sürece geçmesi gerekmektedir. Bu nedenle yarı mamul stoklamak ve yarı mamulleri bekletmek, oldukça maliyetlidir ve istenmeyen bir durumdur (<http://estorototamiroda.com/makaleler/1.htm>).

İlaç Üretimi ve Kimyasal Süreçler İçeren Ürünler: İlaç sanayinde de bazı süreçlerin beklemesiz kısıdıyla modellenmesi gerekmektedir. Peşi sıra uygulanan kimyasal işlemlerde araya zaman girmesi, kimyasalın özelliğini kaybetmesine veya

işlenmesinin daha zor ya da maliyetli olmasına yol açar. İlaç sanayinde, kozmetik ürünlerin, bazı içeceklerin üretiminde kullanılan ve Şekil 3.8’de gösterilen etil alkol üretimi, yine beklemesiz akış tipi üretim örneklerindedir.



Şekil 3.8 Etil Alkol Üretim Süreci

(http://journeytoforever.org/biofuel_library/ethanol_motherearth/z/fig1-1.jpg)

Tahıl bazlı ürünlerden etanol üretimi aşağıdaki şekilde yapılır. Kuru öğütme yapılacaksa önce mısır taneleri veya diğer hububatlar öğütülerek un haline getirilir. Daha sonra un ve su ilave edilerek mayşe hazırlanır. Mayşeye enzim ilave edilerek nişastanın dekstroza dönüştürülmesi sağlanır. Ortama hem pH'ın kontrolü için hem de mayalara besin olarak amonyak ilave edilir. Fermantasyondan önce mayşenin

bakteri yükünü azaltmak için yüksek sıcaklıkta pişirme işlemi uygulanır. Daha sonra mayşe soğutularak fermantasyon tanklarına alınır ve ortama maya ilave edilerek şekerin alkol ve karbondioksite dönüşümü sağlanır. Fermantasyon işlemi genellikle 40 ile 50 saat arasında sürer. Bu işlem sırasında mayanın rahat çalışması için fermantör sıcaklığı kontrol edilir ve mayşe karıştırılır. Fermantasyondan sonra elde edilen karışım distilasyon kolonlarına aktararak etanol ayrıştırılır. Konvansiyonel distilasyonla ve dehidrasyonla karışımın alkol derecesi 200'e yükseltilir. Fermantasyondan sonra distilasyon ile kazanılan alkol elde edildikten sonra geriye kalan artık santrifüje gönderilir. İri taneler ayrılır. Geri kalan kısım %30 kuru madde içerecek şekilde evaporasyonla konsantre edilir. İri taneler ile karıştırılarak yüksek kaliteli hayvan yemi olarak kullanılır. Fermantasyonla elde edilen karbondioksit ise ortamda tutularak gazlı içecek sanayinde veya kuru buz elde etmede kullanılabilir. Bu tip üretim süreçlerinin de -yarı mamullerin mayalanmasını gibi kimyasal işlemleri içerdiğinden- beklemesiz akış tipi modellenmesi uygundur (www.solvekimya.com).

Bazı süreçlerde ürünü yarı mamul halinde bekletmek uygun bekleme alanı olmadığından ya da yarı mamul stoklamanın çok zahmetli oluşundan dolayı istenmez. Özellikle kısıtlı bir alanda büyük ebatta ürünlerin üretildiği süreçler böyledir. Beton köşk olarak da adlandırılan beton trafo üretimi bunlara bir örnektir. Beton köşkler, içlerine jeneratör de yerleştirilebilen elektrik dağıtımının yapıldığı modüler hücrelerdir. Hacimleri 60 m³'den fazla olabilen ve ağırlığı 24 tonu bulan bu yapıları taşımak oldukça zahmetlidir. Üretim tesisi içerisinde özel vinç hatlarıyla taşınabilen beton köşkler, üretim alanı dışına da ancak koruyucu bir ambalaj içerisinde ve kamyonlara yüklenerek çıkartılabilmektedir (<http://www.ulusoyelektrik.com.tr>). Hacimce büyük ürünlerin birleştirildiği montaj

hatları da benzer bir yapı sergiler. Montajı yapılan ürünlerden biri, hat üzerinden seri halde akan ürünlerin birinde gecikme olduğunda, hattı tıkayarak tüm üretimin durmasına neden olabileceğinden bu tür süreçler de beklemesiz olarak modellenebilir.

3.1.3. Çok amaçlı beklemesiz akış tipi problemler (ÇABATP)

3.1.3.1. Amaç fonksiyonları

Bir ATP için $C(\pi_j, k)$, π_j işinin k makinesi üzerindeki tamamlanma zamanını belirtsin. d_j , j işinin tamamlanması için gereken zaman ise $L(\pi_j) = C(\pi_j, m) - d_j$ ifadesini j . işin gecikmesi olarak tanımlayabiliriz. π_j işinin en erken başlama ve gecikmesini, sırasıyla, $T(\pi_j) = \max\{L(\pi_j), 0\}$ ve $E(\pi_j) = \max\{-L(\pi_j), 0\}$ şeklinde tanımlayabiliriz. Bazı gerçek uygulamalarda makinelerin boş kalması düşünülebilir. $I_k(\pi_j) = C(\pi_j, k) - \sum_{j=1}^n p(\pi_j, k)$ ifadesi k . makinenin boş kalma süresine işaret eder. Ayrıca π_j 'nin gecikmiş ya da gecikmemiş olduğunu gösteren $U(\pi_j)$ işaret fonksiyonunu kullanabiliriz. ($U(\pi_j) = 0$ ya da $U(\pi_j) = 1$) λ_j , j işinin mümkün olan bir ağırlığı olarak kabul edilirse, sıklıkla BATP için kullanılan amaç fonksiyonlarını aşağıdaki gibi tanımlayabiliriz (Pinedo, 2002).

- (1) En fazla tamamlanma zamanı $C_{\max}(\pi) = C(\pi_n, m)$

- (2) Toplam ağırlıklandırılmış tamamlanma zamanı $C_w(\pi) = \sum_{j=1}^n \lambda_j C(\pi_j, m)$

- (3) En fazla gecikme $T_{\max}(\pi) = \max T(\pi_j)$

- (4) Toplam ağırlıklandırılmış gecikme $T_w(\pi) = \sum_{j=1}^n \lambda_j T(\pi_j)$

- (5) Makinelerin toplam boş kalma süresi $I_{\text{sum}}(\pi) = \sum_{k=1}^m Ik(\pi)$
- (6) En fazla en erken başlama zamanı $E_{\text{max}}(\pi) = \max E(\pi_j)$
- (7) Toplam ağırlıklandırılmış en erken başlama zamanı $E_w = \sum_{j=1}^n \lambda_j E(\pi_j)$
- (8) Geç kalan işlerin toplam sayısı $N_T(\pi) = \sum_{j=1}^n U(\pi_j)$

3.1.3.2 ÇABATP'in yapısı

ÇABATP yapısı için yukarıda sunulan bazı amaçların eş zamanlı olarak kullanılması düşünülebilir. Ancak bazı amaçlar birbirleri arasında çelişebilir. Amaçlardan birindeki iyileşme diğerindeki kötüleşme anlamına gelebilir. Genellikle tüm amaçları da eş zamanlı en iyileyen bir çözüm yoktur. Çok amaçlı optimizasyon algoritması optimal çözümlerin bir kümesini bulmaya gereksinim duyar. Buna baskın olmayan çözümlerde denir (Ishibuchi ve ark. 2003, Arroyo ve ark. 2005). w adet amacı olan bir çok amaçlı optimizasyon problemi genel olarak aşağıdaki gibi tanımlanabilir:

$$\text{Min } f_1(x), f_2(x), \dots, f_w(x) \quad x \in X \text{ olmak üzere;} \quad (3.4)$$

Burada $f_1(x), f_2(x), \dots, f_w(x)$ ifadesi w adet amacı en küçüklemeindedir ve x, karar değişkenlerinin bir vektörüdür. Çok amaçlı optimizasyonda aşağıdaki tanımlamalar önemlidir.

- (1) Pareto baskınlığı: Eğer x_1 çözümü, x_2 çözümünden baskın ise:

$$(\text{her } i \in \{1, 2, \dots, w\}: f_i(x_1) < f_i(x_2)) \text{ ve } (\text{her } j \in \{1, 2, \dots, w\}: f_j(x_1) < f_j(x_2)) \quad (3.5)$$

- (2) Optimal pareto çözümlü: Bir x_1 çözümlü eğer pareto optimal çözümlü ise yalnızca x_2 çözümlü $x_2 > x_1$ şartını sağlayabilir.
- (3) Optimal pareto kümesi: Eğer küme tüm optimal pareto çözümlerini kapsıyorsa buna optimal pareto kümesi denir.
- (4) Optimal pareto tarafı: Eğer amaç fonksiyonu değerlerinin tamamının bir kümesi optimal pareto kümesindeki çözümlerle uyuşuyorsa bu kümeyle optimal pareto tarafı denir (Qian ve ark. 2009).

3.2. Genetik Algoritmalar

3.2.1. Yazılımlar ve adaptasyon

Bilgisayar programları durağan bir yapıdadır. Özel bir izi takip ettiğini düşünmeden A noktasından başlayıp B noktasına giderler. Birçok yazılımda yazılımın etkinliğini artıran belirleyici algoritmalar kullanılmıştır. Ancak yazılımlarda önemli bir kaide olan adapte edilebilirlik göz ardı edilmiştir. Uygulamaların çoğunda yaratılan nesnelere umulmadık durumlarda onları kodlayan programcılar olmaksızın kendilerini yenileyemezler. Bununla birlikte bazı problemler belirleyici algoritmalar ile kolaylıkla çözülür.

Adapte edilebilirlik modelini incelemek istersek biyolojiden daha iyi bir kaynak bulamayız. Yaşayan varlıkların bazı kimyasal prensipler kümesini esas aldıklarını görürüz. Dünya milyonlarca yıldır değişirken olağan üstü bir adaptasyon ve esneklik göstermişlerdir. Biyolojik modellerin uygulanmasıyla, çözümler üreten yazılımlar

yaratılmıştır. Bazı durumlarda, biyolojik algoritma, onu programlayan programcının hiç hayal etmediği çözümler bulabilmektedir ve model, yazılımı, insanın yaratıcı bakış açısının ötesine götürmektedir.

3.2.2. **Biyolojik evrim**

19. yüzyılın ortalarında Charles Darwin, değişmeyen türlerin, değişen çevre koşulları karşısında uyumsuz bir hale geldiğini gözlemledi. Yavruların ebeveynlere olan benzerliği Darwin'i özelliklerin sonraki nesillere geçtiği fikrine itti. Ayrıca kardeşler arasında önemli farklılıklar olduğunu, bu tür havuzundaki eşsiz bireylerin yiyecek ve çiftleşmek için rekabet halinde olduklarını keşfetti (Darwin, 1906).

Modern bilim biyolojik varlıklar yaratılmasını bir evrim mekanizması olarak tanımlamaktadır. Evrim teorisi Darwin'den bu yana farklılaşmıştır ve yalnızca temeli değişmeden kalmıştır. Günümüzde evrim teorisini geçmiş türlerin fosilleriyle açıklayabiliyoruz ve bu sayede, çevre şartları değişirken organizmaların nasıl değişerek hayatta hakkında bilgi sahibi oluyoruz.

Doğal seçim mekanizmasının işleyişi hakkında klasik bir örnekten bahsedelim. İngiltere'de endüstriyel devrimden önce açık renkli biber güveleri ağaçlardaki beyaz likenlerle bir aradaydı ve kendilerini kuşlardan gizlemektedirler. Koyu renkli güveler beyaz renkli likenlerle kontrast oluşturduklarından genellikle kuşlara yem olmaktadır. Ancak İngiltere'de yeni kömür yakıtlı fabrikaların dumanı likenleri öldürüp ağaçları katranla kapladığında açık renkli güveler kuşlar için görülebilir hedef haline geldiler. Birkaç yıl içerisinde biber güvesi popülasyonu neredeyse

tamamen koyu renkliydi ve doğal seçim mekanizmasıyla yeni çevreye adapte oldular.

Hayatta kalan bireyler yeni nesillerin karakteristiklerini belirlediğinden, bir popülasyonun üretkenlik başarısını tamamıyla o türdeki evrim belirlemektedir. Doğal seçim popülasyonun karakteristikleriyle sınırlıdır. Sıklıkla en iyi adapte olanın hayatta kalması olarak tanımlanan doğal seçim, en iyi mevcut organizmanın hayatta kalmasını sağlamaktadır.

Darwin, hangi karakteristiklerin ebeveynlerden döllere aktarıldığını bilmiyordu. Yalnızca gözlemliyordu ve kayıp parça onun yaşadığı dönemden uzaktaydı. Bu gizem neredeyse yüz yıl sonra çözülecekti. 1951 yılında Francis Crick ve James Watson adlı biyologlar ilk kez deoksiribonucleic acid (DNA)'yı basit kimyasallar kullanarak sarmal bir yapıda tanımladı. DNA protein ve enzimler için kimyasal tarifleri şifrelemiştir ve inanılmaz derecede küçük bir hacme çok fazla miktarda bilgi sığdırmaktadır. Eğer bir insan hücresinin DNA'sı düz bir biçime getirilebilseydi neredeyse 70 cm. uzunluğunda olurdu.

Biyologlar halen yaşamın en temel parçasındaki gizemi araştırmaktadır. DNA'nın sıkıca helezonlaşmış teli, bir organizmanın planının bağımsız parçalarını taşıyan genleri içerir. İnsan DNA'sı, 200 binden fazla gen içerir ve bunlar göz renginden potansiyel gelişim gösterecek bir hastalığa kadar her şeyin kontrolünden sorumludur.

Bir ebeveyn den elde edilen genler sayesinde, döllere özellikler miras kalır. Fungi ve bakteri gibi basit organizmalar, kendilerinin bir kopyasını oluşturarak ürerler. Örneğin tek hücreli bir amip aynı DNA'ya sahip iki organizma biçiminde

bölünerek ürer. Böylelikle tek ebeveynli üreme ile birbirlerinden çok az farklı bulunan iki birey oluşur.

Döllerde bulunan ebeveynlerin gen kombinasyonu ile en karmaşık organizmalar iki ebeveynli üreme ile ürerler. Cinsiyete dayalı üreme sayesinde türlerdeki gen kombinasyonu oldukça fazla olur. Olasılıklar neredeyse sonsuzdur. Örneğin bir insan çifti için, 7 trilyon farklı bireyin dünyaya gelme olasılığı vardır.

Bir popülasyondaki genetik bilgilerin toplanması bir gen havuzunu oluşturur. Büyük gen havuzları küçük olanlardan daha sağlıklıdır. Çünkü daha büyük çapta genetik kombinasyonlara imkân tanırılar. Bir gen havuzundaki daha fazla çeşitlilik daha az çekinik genetik hastalıklara eğilimli olmak anlamına gelir. Küçük bir gen havuzunda akraba ilişkisi sonucunda ortaya çıkan problemler daha çok gözlemlenir.

Doğal seçim, sıklıkla popülasyondaki gen sıklığını değiştirir. Ancak yeni genler üretmez.

İlk yaşam formları kendilerini kopyalayan kimyasallardan oluşmakta ve karşılıklı olarak birbirlerini sınırlamaktaydılar. İlk tam organizma amipin bir benzeriydi ve günümüzde var olan organizmalara ait gen yapısını taşımamaktaydı. Mutasyon ile organizmalar çeşitlilik kazanabilmektedir. Mutasyon organizmaların genlerindeki rastgele değişikliktir. Basit bir organizmanın genlerinin rastgele değişerek çevreye uyum sağlar hale gelmesi oldukça düşük bir olasılıktır. Bununla birlikte çok fazla genin eş zamanlı olarak değiştiği mutasyonların bir etkisi yoktur. İnsan DNA'sı üzerindeki değişiklikler uzun "çöp gen" dizilerinin herhangi bir değişikliğe hizmet etmediğini ve anlamsız olarak yorumlandığını göstermektedir.

Bazen hücreler hasar görmüş DNA'ları tamir ederek mutasyonların döllere ya da hücrelere geçmesini önlemektedir.

Eğer türler çeşitli baskılara maruz kalmazaydı, adaptif yayılım olarak bilinen süreç kolaylıkla artış gösterirdi. Dinozorlar 65 milyon yıl önce ortadan kaybolduklarında yerlerini memeliler aldı ve adaptif yayılım sayesinde ayılardan balinalara kadar binlerce tür memeli yaşam alanlarını genişletti.

1980'lerde biyolog Niles Eldredge ve Stephen Jay Gould evrim teorisinde önemli bir değişiklik yaptı. Evrimin durağan olmadığını ve türden türe sürekli bir değişim içerisinde olduğunu önerdi. Fosil kalıntıları, çevre durağan olduğunda türde bir değişim olmadığını ancak çevre faktörleri değiştikçe türlerde de değişimler olduğunu gösterdi (Eldredge, 1985).

3.2.3. GA'nın temel yapısı

Birkaç basit mekanizma sayesinde; yaşam, türleri, sürekli değişen çevredeki sayılamayacak kadar çok kalıba sokar. Türlerin yaşamındaki bu kalıplar, problemlerin çözümleri olarak da düşünülebilir ve doğal yöntemler, problemlerin çözümlerinin aranması için kullanılabilir. Yazılım dünyası, biyolojik dünyadan daha karmaşık değildir ve hiçbir yazılım parçası tüm yaşamsal teknikleri içermez. Hâlbuki yaşamla ilgili şeyler değişen çevre şartları içerisinde esnek rollere ihtiyaç duyar. Bilgisayar algoritmaları ise sıklıkla, yalnızca, önceden hazırlanmış bir sorunun spesifik cevabını bulmaya çalışır.

Evrimi bir yazılımda aramak yeni değildir. 1975 yılında Michigan Üniversitesi'nde John Holland "Adaptation in Natural and Artificial Systems" adlı kitabında genetik algoritma isimli bir konsept tanımladı. Doğal seleksiyondan ileri gelen hayatın sağlam gövdesi, Holland'a biyolojinin yapay sistemler için bir mecazi anlamı olabileceğini düşündürmüştür. Holland, biyolojik evrim mekanizmalarını kodlara indirgemeye başarmıştır. Böylelikle bu prensipleri yazılım geliştirerek uygulamıştır (Holland, 1975).

GA'nın kesin tanımı ve sınırları halen tartışılan bir konudur. En geniş anlamıyla GA, verilen bir çevre içerisinde üretilen uygunluk değeri yapılı bir çözüm seti yaratır. Süreç aşağıdaki gibi işler (Polat, 2006);

- 1) Rastgele çözümlerden oluşan bir başlangıç popülasyonu oluşturulur.
- 2) Her popülasyon üyesi için bir uygunluk değeri atanır.
- 3) En yüksek uygunluk değerli çözüm; üreme boyunca, yüksek olasılıkla, yeni çözümler üreten ebeveynler olacaktır.
- 4) Yeni çözüm seti eskilerle yer değiştirir ve bir nesil tamamlanmış olur. Süreç ikinci adımdan devam eder.

Bu uygulama dizisi oldukça basit bir yöntemeye dayanır: En iyi uygunluk değerine sahip olan hayatta kalır. Çözümün üretkenlik başarısı, evrim boyunca belirlenmiş olan uygunluk değeriyle doğrudan bağlantılıdır. Bu sezgisel süreçte en az uygunluk değerine sahip olan çözüm, üremek için çok az bir şans bulabilecektir. Bununla birlikte, en fazla uygunluk değerine sahip çözüm üreyen tüm bireylerin ebeveyni olmayacaktır. GA'nın yapısı olasılıklar üzerine kümelenmiştir ve yalnızca biyolojik bir başarı gibi değişim üzerine kurulmuştur.

Burada herhangi bir betimleyici soruya rastgele bir süreç nasıl olurda cevap verebilir sorusu aklımıza gelebilir. Aslında cevap yine canlıların farklı yaşam şartlarına uyum sağlayarak soyunu devam ettirmesiyle hemen hemen aynıdır. Bilgisayar yazılımlarının yönettiği bir çevrede farklı problemler biyolojik dünyadan daha basittir ve doğadaki güçlü araçlar nispeten önemsiz problemlere çözüm bulmada herhangi bir sorunla karşılaşmayacaktır.

GA çözümünün standart modeli, kromozom olarak adlandırılan bir bit yapısıdır. Evrim sürecince kromozomların kod yapısı çözümlenir. DNA'lar 4'lü tabanda kodlanmıştır. İki tabanlı diziler ise 0 ve 1'lerden oluşmaktadır. Üreme işlemiyle, ebeveyn çözümlerin bir kombinasyonundan kromozomlar oluşur ve yeni nesiller yaratırken mutasyona uğrarlar (Ladd, 1996).

Bir problemin GA ile çözümünde takip edilecek işlem adımları aşağıda verilmektedir (Polat, 2006):

1. Arama uzayındaki bütün muhtemel çözümler, dizi olarak kodlanır. Bu diziyi (kromozom) oluşturan her bir elemana gen denir. Her bir dizi, arama uzayında belirli bir bölgeye tekabül eder.
2. Genellikle rastsal bir çözüm seti seçilir ve başlangıç popülasyonu olarak kabul edilir.
3. Her bir dizi için bir uygunluk değeri hesaplanır; bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir.
4. Bir grup dizi (kromozom) belirli bir olasılık değerine göre rastsal olarak seçilip üreme işlemi gerçekleştirilir.
5. Üreme işleminden sonra çeşitli genetik operatörler kullanılabilir.

Uygunluk fonksiyonu, GA'da probleme özel olarak kurulan tek kısımdır. Bu fonksiyon, bireyleri problemin parametreleri haline getirir ve bireylerin uygunluk değerlerine bakarak herhangi bir bireyin problem için ne kadar uygun bir çözüm olacağına karar verir. GA'ın problemin çözümündeki başarısı, çoğunlukla bu fonksiyonun etkin kurulmasına bağlıdır.

3.2.4. Genetik operatörler

3.2.4.1. Seçme

Üreme işlemi; uygunluk değeri yüksek çözümlerin ve bunların çok sayıdaki kopyalarının, yeni nesilde bulunmasını sağlayan işlemidir. Uygunluk değeri, ortalama değer ile normalize edilir. Ortalamanın üstünde uygunluk değerine sahip olan diziler, ortalamanın altında uygunluk değerine sahip olan dizilerden daha fazla üreme şansına sahip olurlar. Uygunluk değerlerine göre dizileri kopyalamak, bir sonraki nesilde, daha kaliteli dizilerin oluşma olasılığını arttırmak demektir. Doğal seçim, oluşturulan diziler arasından en uygun olanının kalması olarak tanımlandığında, bu işlem, doğal seçimin bir yapay sürümü olacaktır.

3.2.4.2. Çaprazlama

GA çaprazlama işlemi ile arasındaki bilgi değişimini gerçekleştirerek daha iyi bireylerin üretilmesini sağlar. Çaprazlama yapılırken iki bireyin belirli genleri

karşılıklı olarak yer değiştirir. Böylelikle iki yeni birey, başka bir deyişle iki yeni olası çözüm üretilmiş olur. Çaprazlama işlemini gerçekleştirmek için, öncelikle, üreme işlemi ile oluşturulmuş eşleştirme havuzundaki yeni kopyalanmış dizinin elemanları rastgele eşlenir. İkinci olarak, seçilen dizilerin bitleri, rastgele seçilmiş çaprazlama noktasından itibaren, karşılıklı olarak değiştirilirler. Çaprazlamalar, problemlerin türüne göre değişiklikler göstermektedir. Düzenli çaprazlama en çok kullanılan çaprazlama türlerindedir.

Düzenli (uniform) çaprazlama: Bu çaprazlama türünde sadece rastgele seçilen genler yer değiştirir.

Ebeveyn 1: 1 0 1 1 0 1 0 1

1. Birey 1 0 0 1 0 1 1 1

↓ ↓

Ebeveyn 2: 1 1 0 1 1 1 1 0

1. Birey 1 1 1 1 1 1 0 0

Düzenli çaprazlamada her genin değiştirilme olasılığı %50'dir.

Düzenli çaprazlamanın dışında kısmi planlı çaprazlama, dairesel çaprazlama, çok noktalı çaprazlama gibi çaprazlama çeşitleri de bulunmaktadır.

3.2.4.3. Mutasyon

Popülasyon içerisinde tek tip bireyler varsa çaprazlamalarından elde edilecek bireylerde yine aynı tip olur. Dolayısıyla popülasyonu oluşturan bireyler arasında çeşitlilik olmaz. Bunu engellemek için mutasyon işlemcisi kullanılır. Günümüzde farklı mutasyon işlemcileri kullanılmaktadır. Örneğin, rastgele ikili yer değiştirme mutasyon işlemcisinde; rastgele seçilen iki genin yerleri değiştirilip, oluşan dizideki

sıralamanın uygunluğu kontrol edilerek, çözüm kümesine dâhil edilir. Tek genin değiştiği mutasyonlarda ise dizilerin kodlanmasına göre, Şekil 3.9'daki gibi bir yöntem izlenebilir. Benzer şekilde, elde edilen genin uygun bir çözüm olup olmadığı kontrol edilmelidir.

Tek genin değiştiği mutasyonlarda, ikili kodlamada, “1” olan gen “0” olur, “0” olan gen ise “1” olur. Onluk tabanda ise durum farklıdır. Bir rakamın alabileceği 10 değer vardır. Rakamın değeri, mutasyon sırasında geriye kalan 9 değerden biri olur.

<u>İkili kod</u>	<u>Onlu kod</u>
10110101	181
↕	
11110101	245

Şekil 3.9 İki tabanından mutasyona uğramış kod ve on tabanından karşılığı

3.2.5. Elitizm

Üreme, çaprazlama ve mutasyon işlemleri sonrasında kuşakta bulunan en iyi uygunluk değerine sahip birey, sonraki kuşağa aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra oluşan yeni kuşağa bir önceki kuşağın en iyi (elit) bireyi, yeni kuşaktaki herhangi bir birey ile değiştirilir. Buna elitizm adı verilir (Mansfield, 1990).

3.2.6. Çaprazlama ve mutasyon olasılığı

GA tekniğinin çaprazlama olasılığı ve mutasyon olasılığı olmak üzere iki basit parametresi vardır. Çaprazlama olasılığı, çaprazlamanın hangi sıklıkta yapılacağını 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. Çaprazlama eski bireylerden iyi taraflar alınarak elde edilen yeni bireylerin daha iyi olması umuduyla yapılır.

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

3.2.7. GA'ların akışı

GA'nın işleyişi aslında çok basit prensiplere dayanmaktadır. Yapılacak işlemler çok az sayıdadır ve bunların sıralaması da bellidir. GA, bütün problemleri aynı mantıkla çözer. Yani problem değişince genellikle program değiştirmeye gerek kalmaz. Çünkü GA'ya problemi nasıl çözeceğini anlatmaya gerek yoktur. GA problemi nasıl çözeceğini bilir. GA, verilen hata fonksiyonuna göre kendine bir

çözüm yolu bulur. Çözüm başarılı olmazsa başka çözümü dener. Algoritmanın işleyişi ise oldukça basittir. İşleyişini kısaca aşağıdaki şekilde açıklayabiliriz:

1. İlk önce rastgele bireyler (sayı toplulukları) oluşturulur. Bu bizim başlangıç topluluğumuzdur. Oluşturulacak birey sayısı, programın işleyişi bakımından çok önemlidir. Genellikle her programda 50'den fazla birey oluşturulur. Ancak yapılan işlem karmaşıkta 150 birey bile yetmeyebilir.

2. GA'da, probleme özel tek kısım uygunluk fonksiyonudur. Uygunluk fonksiyonunun amacı bireylerin yalnızca o anki toplum içindeki uygunluklarını bulmaktır. Her birey teker teker uygunluk fonksiyonuna sokulur ve bireydeki kromozomlara göre o bireyin hayatta kalma şansı hesaplanır. Bulunan bu değerler ise bir tabloya aktarılır.

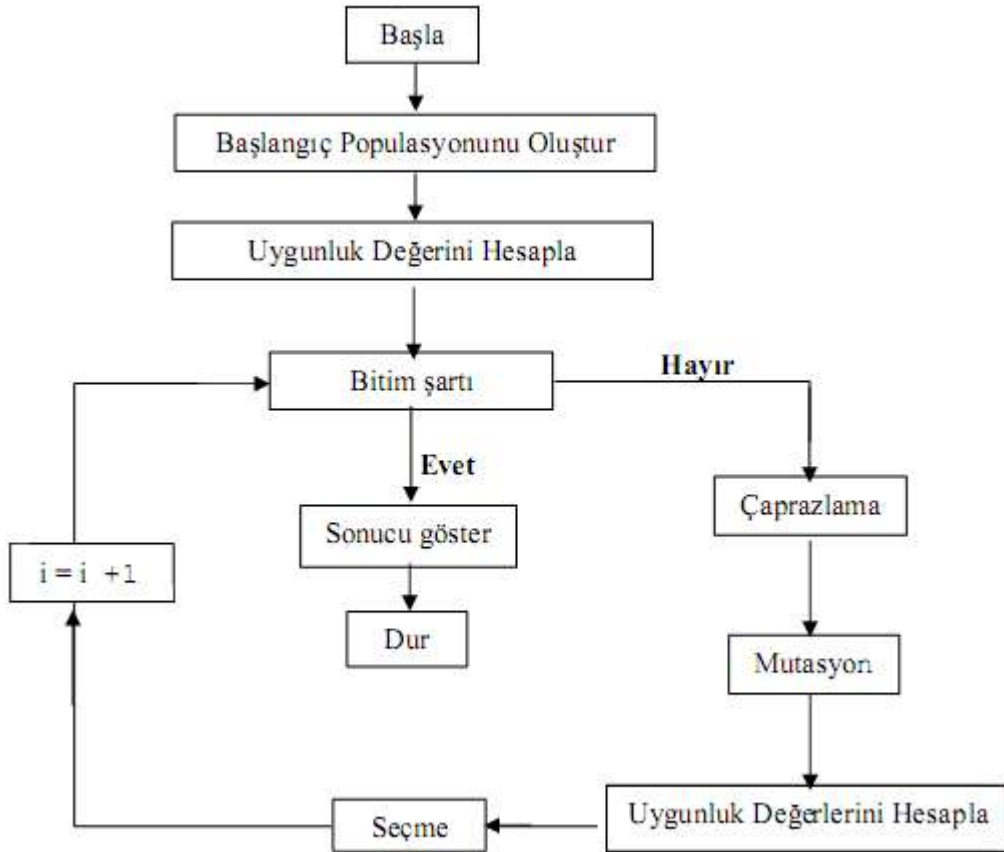
3. Eşleştirme, hangi bireylerin hangi bireylerle çaprazlanacağını belirlemektir. Eşleştirmeyi rulet çemberi adını verdiğimiz bir fonksiyonla yapabiliriz.

4. Eşleştirilen bireyler sırasıyla ikişerli olarak çaprazlanırlar. Çaprazlama yaparken hangi çaprazlama yönteminin kullanılacağı probleme göre değişir. Genellikle tek noktalı ya da iki noktalı çaprazlama kullanılır. Çaprazlama, iyi özelliklerin bir araya gelip sonuca daha çabuk ulaşmayı sağlar. Ancak çeşitliliği sağlamak için sadece çaprazlama yeterli değildir. Bazı bireyleri mutasyona sokmak gerekebilir.

5. Çaprazlamadan sonra bireylere, mutasyon oranına uygun olarak, mutasyon işlemi uygulanır. Mutasyon oranı genelde düşük tutulur. Örneğin 0,01 uygun bir değer olabilir. Mutasyon oranının 0,01 olması gen havuzundaki her 100 genden 1 tanesinin değiştirileceğini gösterir.

6. Elde ettiğimiz yeni bireylerden ve önceden elimizde bulunan bireyler arasından iyilerini seçmek için hayatta kalma şansları tekrar hesaplanır. Bütün bireyler doğal seleksiyona sokulur. Bu eleme işlemi sayesinde topluluk büyüklüğü sabit tutulur ve güçsüz bireyler elenmiş olur.

7. İstenilen sonuç bulunmuşsa program sonlandırılır. Ancak halen istenilen sonuç bulunamamışsa ya da yeterli iterasyon sayısına ulaşılammışsa ikinci adıma geri dönülür ve aynı işlemler tekrarlanır.



Şekil 3.10 GA'nın akış şeması

3.2.8. GA'ların kullanılma nedenleri

Öncelikle diğer yöntemlerin tercih edilmeme sebepleri incelenmelidir (Polat, 2006). Denklem optimizasyonunda;

1. Türev-İntegral hesabına (calculus) dayananlar,
2. Sayılamaya (enumeration) dayananlar,
3. Rastgele aramalar (random searches) olmak üzere üç tip yöntemden bahsedilir.

Türev-integral hesaplamalarına dayanan hesaplama yöntemleri çok derinlemesine çalışılmıştır. Bu yöntemler fonksiyonun türevinin köklerinin, fonksiyonun en küçük ve en büyük değer veren noktaları olmasından yararlanır. Gerçek problemler için sıfır veren noktaları bulmak da ayrı bir problemdir.

Diğer bir yöntem ise alınan bir noktadan sadece yukarı ilerleyerek en iyi sonucu bulmayı hedefler. Tepe tırmanma (hill climbing) denen bu yöntem, fonksiyon grafiğinin tepelerini tırmanır. Ancak çok sayıda dönme noktası içeren bir fonksiyonda çok sayıda tepe oluşur. Hangi tepenin en iyi çözüm olduğunu bilinemez. Sayımlama yöntemleri ise oldukça alışlagelmiştir. Sürekli, gerçel sayı aralıkları belli sayıda parçaya ayrılarak parçalar denenir. Ancak problemler, böyle çözmek için büyük olabilir. Bu yöntemin biraz daha geliştirilmiş şekli dinamik programlamayla (dynamic programming) oluşturulur. Parçalar arasından iyi görünenler seçilir. Bu parçalar da parçalara ayrılarak işlem tekrarlanır. Bu yöntem de tepe tırmanma yöntemi gibi yanlış tepeleri araştırabilir. Dinamik programlama tepelerin olmadığı aralıklarda, başarılı ve hızlıdır.

Optimizasyonun;

- Bir işin daha iyi yapılması,
- En doğru şekilde yapılması olmak üzere iki amacı vardır.

Günümüzde rastgele aramaların kullanımı artmaktadır. Bu tip aramalar, optimizasyonu daha iyi yapma amacını sağlamakta daha başarılılardır. İnsanların bilgisayarlardan genel beklentisi mükemmellik olduğu için bu tip aramalar başarısız görünebilir. Ancak GA gibi sezgisel yöntemler, klasik yöntemlerin çok uzun zamanda yapacakları işlemleri kısa bir zamanda çok net olmasa da yeterli bir doğrulukla yapabilmektedir.

3.2.9. Fonksiyon maksimizasyon örneği

Buradaki amaç GA'nın bilgisayarda nasıl çalışacağını kâğıt üzerinde açıklayıcı şekilde anlatmaktır. Amaç fonksiyonu: $f(x)=x^2$, $x \in [0,69]$ şeklinde verilen bir fonksiyonun, verilen aralıkta maksimizasyonunun yapılması istenmektedir. Bir fonksiyon maksimizasyon örneğinin çözümünü adım adım açıklayacak olursak;

1. Adım: İlk olarak x sayısının kodlanması işlemi yapılmalıdır. x 'in 0 ve 1'lerden oluşan ikili tabandaki gösterimi kullanılacaktır. Dolayısıyla x , 6 bit uzunluğunda bir kodla temsil edilecektir.

2. Adım: Popülasyon birey sayısı $n=4$ olarak seçilmiştir. Popülasyonu oluşturan dört birey, her biri 6 bit uzunluğunda birer kromozomla temsil edildiği için toplam 24 kere yazı tura atmak suretiyle belirlenmiştir. Elde edilen birey kromozomları aşağıdadır.

$$\text{Birey 1: } 011010 \quad x = 26 \quad \rightarrow \quad x^2 = 676$$

$$\text{Birey 2: } 110001 \quad x = 49 \quad \rightarrow \quad x^2 = 2401$$

$$\text{Birey 3: } 010001 \quad x = 17 \quad \rightarrow \quad x^2 = 289$$

$$\text{Birey 4: } 100110 \quad x = 38 \quad \rightarrow \quad x^2 = 1444$$

3. Adım: Yukarıda belirlenen bireyler için $f(x)=x^2$, bireylerin uygunluk değerlerini verir. Dört bireyin toplam uygunluk değerleri “ $676+2401+289+1444=4810$ ” dur. Dolayısıyla her bir bireyin rulet çemberinde kaplayacağı alan aşağıdaki şekilde hesaplanır:

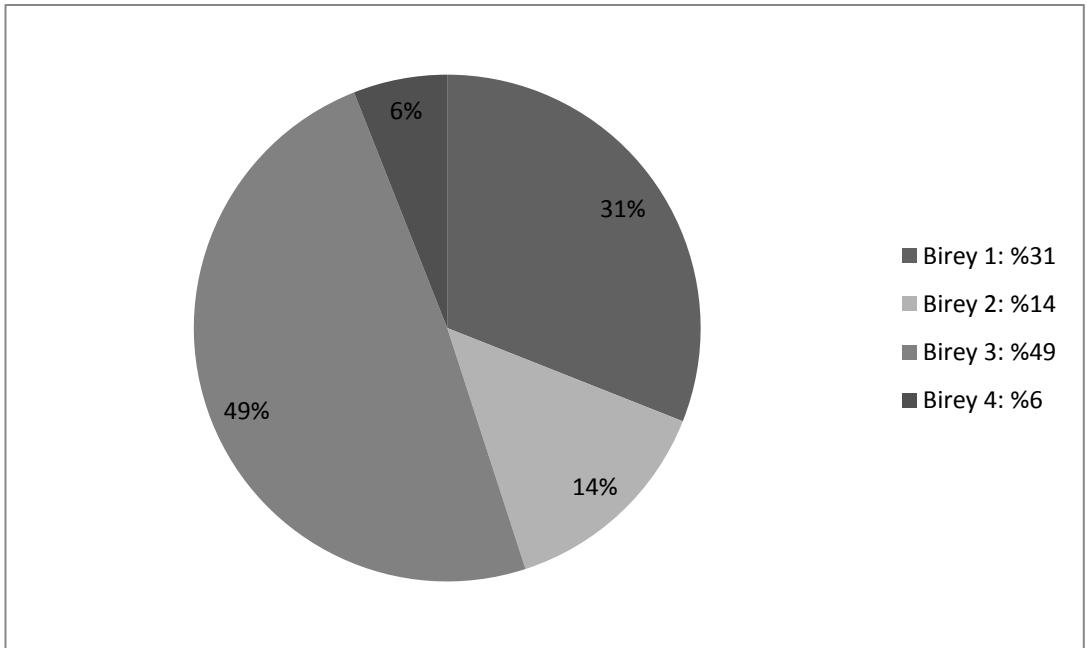
$$\text{Birey 1: } 676/4810 = 0,14 : \%14$$

$$\text{Birey 2: } 2401/4810 = 0,49 : \%49$$

$$\text{Birey 3: } 289/4810 = 0,06 : \%6$$

$$\text{Birey 4: } 1444/4810 = 0,31 : \%31$$

Bu değerler, rulet çemberinin her çevrilişinde hangi olasılıkla hangi bireyin seçileceğini belirtir, yani 0.14 olasılıkla 1 numaralı birey seçilecektir. Rulet çemberi ve bireylerin tekerlek üzerindeki dağılımları Şekil 3.11’de gösterilmiştir.



Şekil 3.11 Rulet çemberi dağılımı

4. Adım: Popülasyondaki birey sayısının sabit kaldığı varsayıldığından dolayı, rulet çemberi 4 kere çevrilerek çiftleşme havuzu oluşturulacaktır. Rulet çemberi döndürülmüş ve aşağıdaki sonuçlar elde edilmiştir.

Birey 1 : 1 kez

Birey 2 : 2 kez

Birey 3 : 0 kez

Birey 4 : 1 kez

Bunun sonucunda elde edilen çaprazlama havuzu aşağıdaki gibidir;

Aday 1 : 011010 (Birey 1)

Aday 2 : 110001 (Birey 2)

Aday 3 : 110001 (Birey 2)

Aday 4 : 100110 (Birey 4)

5. Adım: Çaprazlama havuzu belirlendikten sonra iki aşamalı çaprazlama uygulanır. İlk aşamada kromozomlar çaprazlamak üzere rastgele olarak eşlenirler. Her ikili grup için bir kere zar atılarak çaprazlamanın oluşacağı nokta belirlenir. Rastgele eşleştirme yapılmış ve bunun sonucunda (Aday 1, Aday 2) ve (Aday 3, Aday 4) ikili grupları oluşmuştur. Çaprazlama noktaları da zar atılarak 1. grup için $k=4$ ve 2. grup içinde $k=2$ olarak belirlenmiştir. Bu aşamadan sonra çaprazlama gerçekleştirilmiş ve şu sonuçlar oluşmuştur (çaprazlama noktaları “/” ile belirtilmiştir).

Çaprazlama grubu 1: ($k=4$)

Aday 1 : 0110/10 oluşan Birey 1 : 011001

Aday 2 : 1100/01 oluşan Birey 2 : 110010

Çaprazlama grubu 2 : ($k=2$)

Aday 3 : 11/0001 oluşan Birey 3 : 110110

Aday 4 : 10/0110 oluşan Birey 4 : 100001

6. Adım: Son aşama olan mutasyon, bitler düzeyinde uygulanır. Bu örnekte her bir bit için (toplam 24 bit var) mutasyon olma olasılığı 0,01 olarak seçilmiştir. Dolayısıyla her bir bit için ağırlıklı yazı/tura (mutasyon olasılığına göre) atılarak hangi bitlerin mutasyona uğrayacağı belirlenir. Bu işlem yapılmış ve sonuçta oluşan Birey 3'ün 2 numaralı bitinde mutasyon olacağı ortaya çıkmıştır.

Oluşan Birey 3: 110110

Mutasyon sonucu oluşan Birey 3: 100110

Bu adımın tamamlanmasıyla bir sonraki kuşağı oluşturacak toplumun bireyleri belirlenmiş olur. Yeni popülasyon şu şekildedir;

Birey 1 : 011001 $x=25$ $x^2=625$

Birey 2 : 110010 $x=50$ $x^2=2500$

Birey 3 : 100110 $x=38$ $x^2=1444$

Birey 4 : 100001 $x=33$ $x^2=1089$

Üç temel operatörden oluşan GA her aşamada yeni oluşan kuşağa uygulanarak bir sonraki kuşak elde edilecektir (Polat, 2006).

Yukarıdaki örnekte tek bir iterasyon yapılmış ve başlangıç toplumundan bir sonraki kuşak oluşturulmuştur. Ancak genetik algoritmanın çalışmasının tam olarak gözlenebilmesi için, tek bir iterasyon yeterli değildir. Yukarıdaki işlemler fazlaca rastgele gibi görünse de, uygunluk değeri yüksek olan bireylerin seçilme ve çiftleşme olasılıkları yüksek olduğu için, kuşaklar ilerledikçe, toplumu oluşturan bireylerin uygunluk değerlerinin ortalamasının da arttığı gözlenecektir. Bunun için ise tek bir iterasyon yeterli değildir.

3.2.10. GA'ların genel uygulama alanları

Karmaşık problemleri hızlı ve optimale yakın olarak çözebilen genetik algoritmalar, çeşitli problem tiplerine uygulanabilmektedir. Büyük çözüm uzaylarının geleneksel yöntemlerle taranması hesaplama zamanını arttırmaktadır. Ancak bu tip problemlere, genetik algoritmalar ile kısa sürede, kabul edilebilir çözümler bulunabilmektedir. GA'lar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir.

GA'ların uygulama alanları genel uygulama alanları ve işletmelerdeki yaygın uygulama alanları olmak üzere iki sınıfa ayrılarak incelenmiştir.

3.2.10.1. Optimizasyon

Bir arama yöntemi olan genetik algoritmalar, farklı bilim dallarındaki optimizasyon problemlerini çözmeye kullanılmaktadır. GA'ların uygulandığı optimizasyon problemleri, fonksiyon optimizasyonu ve çok amaçlı (combinatorial) optimizasyonu altında toplanabilir. GA araştırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. GA'lar; geleneksel optimizasyon tekniklerine göre, zor, süreksiz ve gürültü (noisy) içeren fonksiyonları çözmeye daha etkinlerdir. Optimize edilecek amaç fonksiyonunun süreksiz olması halinde, süreksizlik noktalarında fonksiyonun türevi alınamayacağından, türev almaya dayalı optimizasyon yöntemleri kullanılamamaktadır. Oysa GA'lar, problemlerin çözümü için türev veya diğer

yardımcı bilgilere gereksinim duymadığından, özellikle bu tip problemlerin çözümünde, geleneksel yöntemlere göre önemli bir üstünlük sağlamaktadır.

3.2.10.2. Otomatik programlama ve bilgi sistemleri

GA'ların yaygın olarak kullanıldığı alanlardan biri, belirli ve özel görevler için gerekli olan bilgisayar programlarını geliştirmektir. Ayrıca, diğer hesaplama gerektiren yapıların tasarımı için de kullanılmaktadır. Bunlara örnek olarak, bilgisayar yongaları tasarımı, ders programı hazırlanması ve ağların çizelgelenmesi verilebilir.

GA'ların uygulandığı, bir diğer optimizasyon problem sınıfı olan, çok amaçlı optimizasyon problemleri ise, istenen amaçlara ulaşmak için sınırlı kaynakların etkin tahsis edilmesiyle ilgilidir. Bu sınırlar; genel olarak, işgücü, tedarik veya bütçe ile ilgilidir. Sözü edilen “çok amaçlılık”, yalnızca sonlu sayıda alternatif uygun çözümün mevcut olmasıyla ilişkilidir. Çok amaçlı optimizasyon, iyi tanımlanmış bir problem uzayında, bir veya daha fazla optimal çözüm bulma sürecidir.

3.2.10.3. Mekanik öğrenme

GA'lar, sınıflama sistemlerinde, kural bulma mekanizması olarak kullanılmaktadırlar. GA ayrıca, sinir ağlarında ve proteinin yapısal analizinde de kullanılmaktadır.

3.2.10.4. Ekonomik ve sosyal sistem modelleri

Genetik algoritmalar, yenilik sürecinin modellenmesi amacıyla da kullanılmaktadır. Ayrıca GA'ların, fiyat verme stratejilerinin gelişim süreçlerini ve kazanç getiren pazarların ortaya çıkış süreçlerini modelleme alanlarında kullanımları oldukça yaygındır. GA'lar, sosyal sistemlerin evrimsel yönlerini anlamak amacıyla kullanılmaktadır. Bunlara örnek olarak işbirliğinin evrimi, iletişimin evrimi ve karıncalardaki iz takibi davranışının evrimi verilmektedir.

3.2.10.5. Finans

GA'lar, finansal modelleme uygulamaları için son derece uygunlardır. GA'lar amaç fonksiyonu odaklıdır. Finans problemlerinde genel olarak, amaç fonksiyonlarını tahmin etme gücüne veya bir kıyaslama sonucuna bağlı getirilerdeki gelişmeleri içerir. Kullanılan araç ve problemler arasında mükemmel bir eşleşme mevcuttur. Özellikle hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede, kaynak tahsisi ve uluslararası sermaye tahsisi stratejilerini belirlemede genetik algoritmalar kullanılabilir.

3.2.10.6. Pazarlama

Tüketicilere ait verileri analiz etmek, çeşitli tüketici profilleri çıkarmak ve bu profiller dayanarak pazarlama stratejilerini uygulamak, pazarlamanın en önemli

fonksiyonlarından biridir. Tüketicilerin profilleri çıkarılarak, belirli satın alma eğilimleri yakalanabilmektedir. GA tabanlı yaklaşım kullanılarak, veri yığınlarından satın alma eğilim modelleri elde edilmektedir.

3.2.10.7. Üretim - lojistik

GA'ların en çok uygulandığı alanların başında üretim-lojistik gelmektedir. Burada üretim-lojistik alanıyla ilgili çeşitli problemlerin, GA'lar kullanılarak çözümü yapılabilir. Bu problemler;

- Montaj Hattı Dengeleme Problemi,
- Gezgin Satıcı Problemi,
- Tesis Yerleşim Problemi,
- Atama Problemi,
- Hücresel Üretim Problemi,
- Sistem Güvenilirliği Problemi,
- Taşıma Problemi,
- Araç Rotalama Problemi,
- Minimum Yayılan Ağaç Problemi,
- Çizelgeleme Problemi'dir.

3.2.11. Çok amaçlı GA'lar

Bütün optimizasyon ve araştırma problemleri, mevcut durumu bir tek kritere göre değerlendirir ve bu kriter, amaç fonksiyonu olarak veya GA'lar da uygunluk fonksiyonu olarak adlandırılır. Bu yaklaşım birçok problemde başarıyla uygulanır. Ancak aynı anda birçok kriterin bulunduğu problemleri, tek kriter haline getirmek oldukça zordur. Bu durumda çok amaçlı ve çok kriterli problemler ortaya çıkar.

Çok amaçlı modeller, tasarım planlama modellerinde, günlük hayattaki karmaşık sistemlerin modellenmesinde, taşıma, bütçeleme vb. problemlere çözüm bulmada kullanılmaktadır.

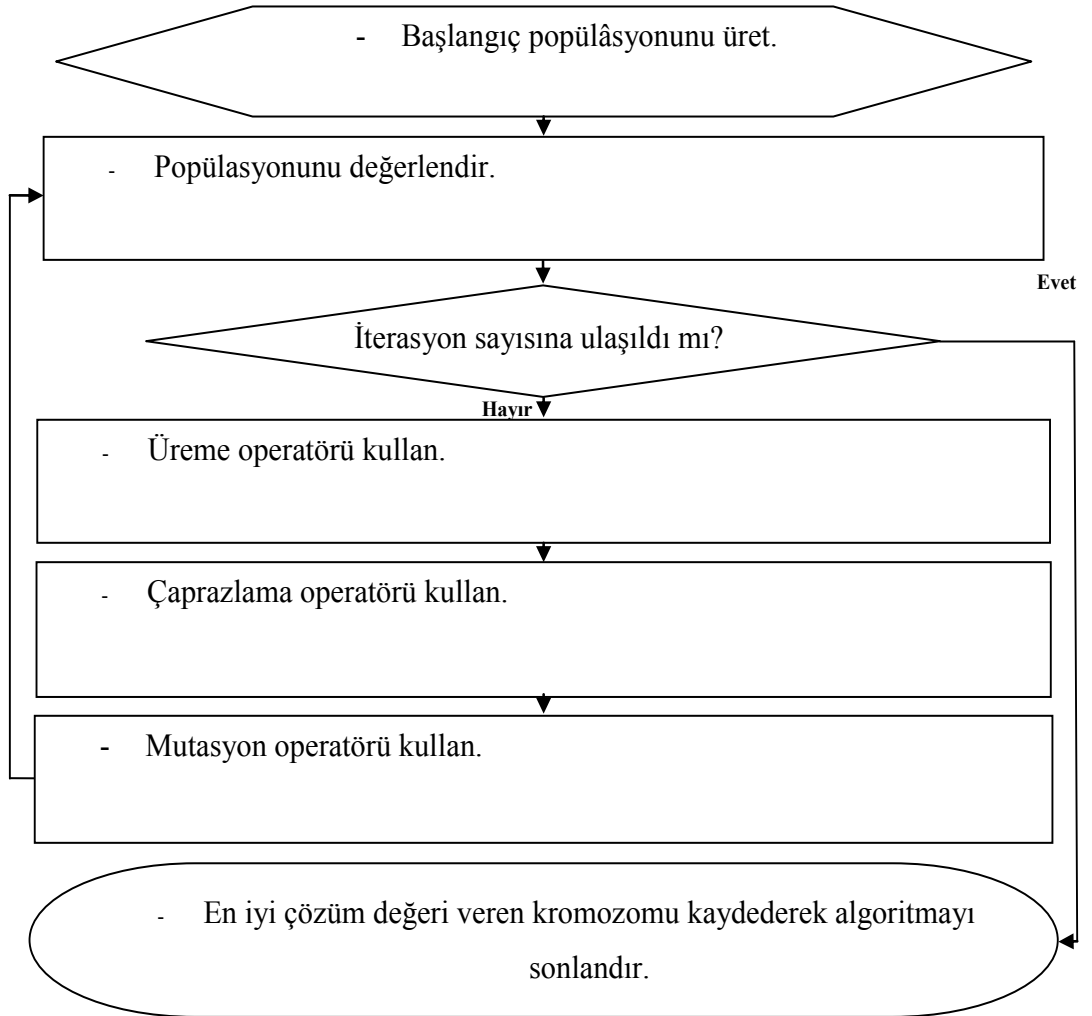
Şekil 3.12' de genetik algoritma akış şeması görülmektedir. Çok amaçlı ve tek amaçlı genetik algoritma yapıları arasındaki temel fark uygunluk fonksiyonunun yapısıdır.

GA, birbirlerinden farklı, çok sayıda amaca sahip optimizasyon problemlerinde rahatlıkla kullanılabilir. GA, popülasyondan popülasyona global arama yaptığından, ileri düzeyde matematiksel hesaplamalara ihtiyaç duymamaktadır. Bu nedenle oldukça kompleks problemler GA ile çözülebilmektedir. Çok amaçlı optimizasyon problemlerinin GA ile çözümünde ilk ortaya çıkan sorun, GA'da kullanılan uygunluk değerinin nasıl belirleneceğidir.

Uygunluk fonksiyonunun belirlenmesi ile ilgili son on yılda çeşitli yaklaşımlar geliştirilmiştir Bunlar;

1. Vektör değeri yaklaşımı, (VEGA)

2. Ağırlıklar toplamı yaklaşımı,
3. Pareto tabanlı yaklaşım,
4. Uzlaşma yaklaşımı,
5. Amaç programlama yaklaşımı



Şekil 3.12 GA akış şeması

Çok amaçlı GA optimizasyonlarında ilk geliştirilen yaklaşım, VEGA'dır. Bu yaklaşımda belirli bir ölçekteki uygunluk değeri yerine, her bir kromozomla geliştirilen uygunluk değeri kullanılır.

Ağırlıklar toplamı yaklaşımında, her bir amaç fonksiyonuna bir ağırlık değeri verilir ve bu ağırlıkla amaç fonksiyonu tek bir fonksiyon haline getirilir. Sabit ağırlık yönteminde, nesil sayısı boyunca amaç denkleminin değeri değişmez. Murata, Ishibuchi ve Tanaka, rastsal ağırlıklı bir yaklaşım önermişlerdir. Bu yaklaşımla her nesilde, rastsal olarak amaç fonksiyonları için yeni ağırlık değerleri oluşturmaktadır. Pareto yaklaşımı, Horn, Nafpliotis ve Goldberg tarafından geliştirilmiştir (Engin, 2001).

3.2.12 GA ve çizelgeleme

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

Çizelgeleme problemlerinde temel amaç, her bir operasyonun her bir makinede yapılış sırasını belirleyerek, minimum tamamlanma süresini elde etmektir. GA'nın çizelgeleme problemlerinde kullanımı iki farklı şekilde gerçekleşmiştir. Bunlar;

1. GA yardımı ile permütasyon tipi iş sırası bulmak.

2. GA ile bulunan permütasyon sırasının, bilinen sezgisel yöntemler ile karşılaştırılması.
3. GA'da, optimuma yakın çözüm veren parametreler belirlemek ve kısa sürede uygun çözüme ulaşılması için üç kategoride çalışma yapılmaktadır.

Bunlar;

1. Genetik operatörlerin belirlenmesi ile ilgili çalışma,
2. GA ile ilgili sezgisel özellikte operatör geliştirme,
3. Melez (hibrit) genetik algoritma geliştirme.

Giffler ve Thompson'ın önerdiği algoritma tabanlı çaprazlama operatörü, sezgisel özellikte geliştirilmiş bir operatördür. Bu çaprazlama yöntemi, Yamada ve Nakano tarafından geliştirilmiştir. Komşuluk arama tabanlı mutasyon yöntemi de bu kategoride geliştirilmiş bir yöntemdir. Melez genetik algoritma kapsamında yapılan çalışmalara, GA ile yerel aramanın birleştirilmesi, GA ile Giffler Thompson metodunun birleştirilmesi ve GA ile darboğaz giderme sezgisel yönteminin birleşmesi, örnek verilebilir.

3.2.13. Parametre optimizasyonu

GA'lar, çözüm uzayında arama yaparken bir takım parametreleri kullanırlar. Bunlar; üreme, çaprazlama ve mutasyon yöntemi ile çaprazlama mutasyon oranları ve başlangıç popülasyonundan oluşur. Kısa sürede çözüme ulaşmak için, bu parametrelerin seçimi ile ilgili herhangi bir kural mevcut değildir. Herhangi bir

problemin çözümünde kullanılan GA için, optimum veya optimuma yakın çözümü veren kontrol parametresi seti, başka bir GA uygulaması için genelleştirilemez. GA'nın uygulandığı problemlerde bu parametreler, deneme-yanılma yöntemi ile seçilir (Engin, 2001).

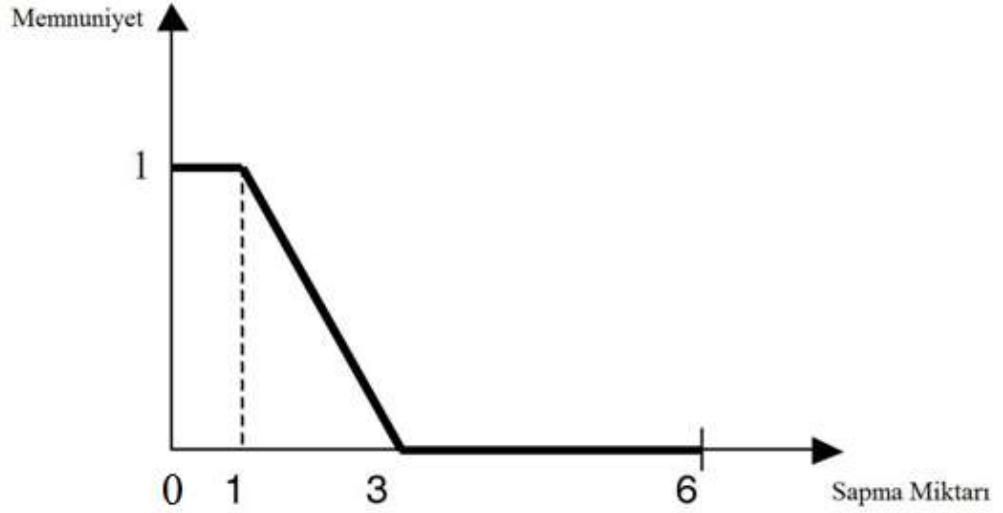
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Veri Toplama

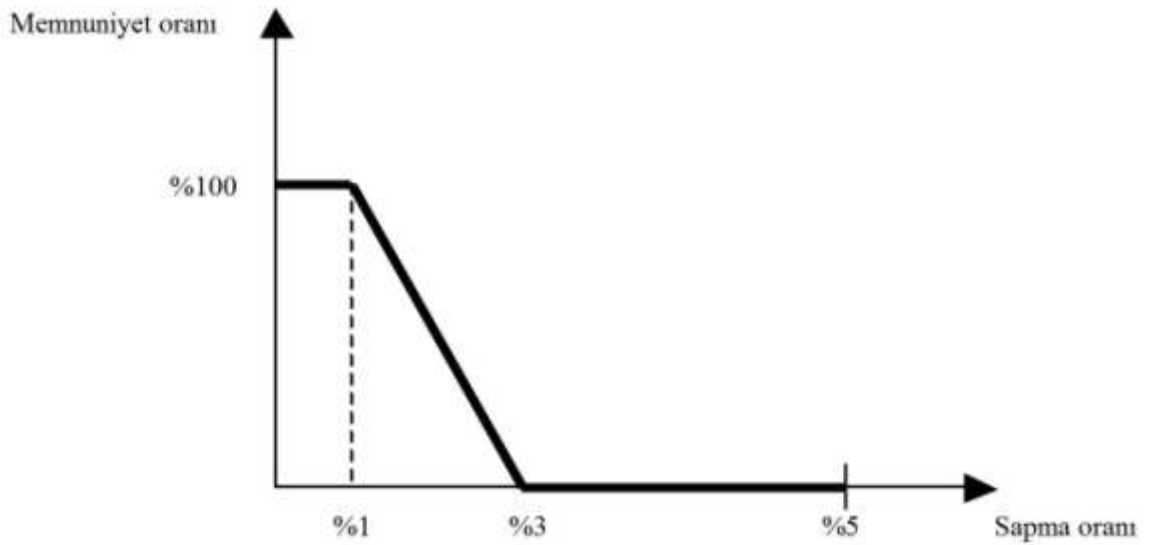
BATP'lerin çözümü için bir melez ÇAGA geliştirilmiştir. Bu algorithmada, Tseng ve ark.'nın (2009) ATP için geliştirmiş olduğu melez GA'ya benzer olarak, yerel ve global arama yöntemleri kullanılmıştır. Tseng ve ark. C_{max} ve F_i amaçlarını tek amaçlı bir yapı ile en iyilemeyi hedefleyip, her bir amaç için farklı parametre seti ile çalışırken, geliştirilen çok amaçlı GA bu iki amacı birlikte en iyilemeyi hedeflemiştir. Bununla birlikte Tseng ve ark.'nın algoritma akışında kullandıkları bazı verimsiz işlemler (benzer iş komşuluğu kombinasyonlarını tekrarlayan arama işlemleri, birbirini kapsayan ya da biri çağırıldığında diğerini gereksiz kılan arama ve çaprazlama fonksiyonları) çıkartılarak buna bağlı alt fonksiyonlar çağırılmamış böylelikle algoritmanın hızı artırılmıştır. Geliştirilen algoritma Delphi dilinde kodlanmış ve 7.0 sürümü derleyicide derlenerek uygulama dosyası haline getirilmiştir.

Amaç birleştirilirken, Allouche ve ark.'nın (2009) ATP'ler için geliştirdikleri, memnuniyet fonksiyonu tabanlı yapıdan yararlanılmıştır. Buna göre, her amacın tek başına en iyilenmesiyle bulunan değer kullanılarak, her amacın memnuniyet ve ret sınırları belirlenmektedir. Eğer çok amaçlı olarak bulunan sonuç, tek amaçlı olarak bulunan değerden tam memnuniyet sınırı kadar fazla olursa (bizim de kullanmış olduğumuz en küçüklenen amaçlar için) bu yönetici için tam memnun edicidir, dolayısıyla kabul edilir. Diğer taraftan bulunan değer tam memnuniyet sınırından

fazla fakat memnuniyet aralığı içerisindeyse, daha az da olsa, memnun edicidir. Eğer kabul aralığında ise bu yöneticiyi memnun etmeyecektir. Ancak yine de iş çizelgesi kabul edilecektir. Elbette memnuniyet aralığı içerisindeki bir çözüm, kabul aralığı üst sınırına yaklaştıkça, yöneticinin memnuniyeti de doğrusal olarak azalacaktır.



Şekil 4.1 Memnuniyet miktarı fonksiyonu



Şekil 4.2 Memnuniyet fonksiyonu oranı

Şekil 4.1’de 1-3 aralığında memnuniyet oranı 3’e yaklaştıkça düşmektedir. 3’ten sonra memnuniyet oranı 0’dır. Ancak çözüm halen kabul edilebilir. 6’dan sonra ise artık çözüm kabul edilemez bir haldedir.

Bu çalışmadan farklı olarak memnuniyet fonksiyonu oransal olarak ele alınmıştır. Böylelikle aynı sınırlar oransal olarak tüm problemlere uygulanabilir haldedir. Farklı problemler için tek amacın en iyilenmesiyle, çok amaçlı yapı arasındaki farklılık, seçmiş olduğumuz F_i ve C_{max} amaçları için daha iyi gözlenebilir hale gelmiştir. Bu durum Şekil 4.2’ de gösterilmiştir. Buna göre tek amaçlı olarak bulunan en iyi çözümden %1’e kadar tam memnun edici kabul edilirken %1 ile %3 arasındaki memnuniyet azalmakta %3-%5 arasında sonuç memnun edici bulunmamakta ama kabul edilmekte ve %5’ten sonraki sapma oranları için çözüm kabul edilmemektedir.

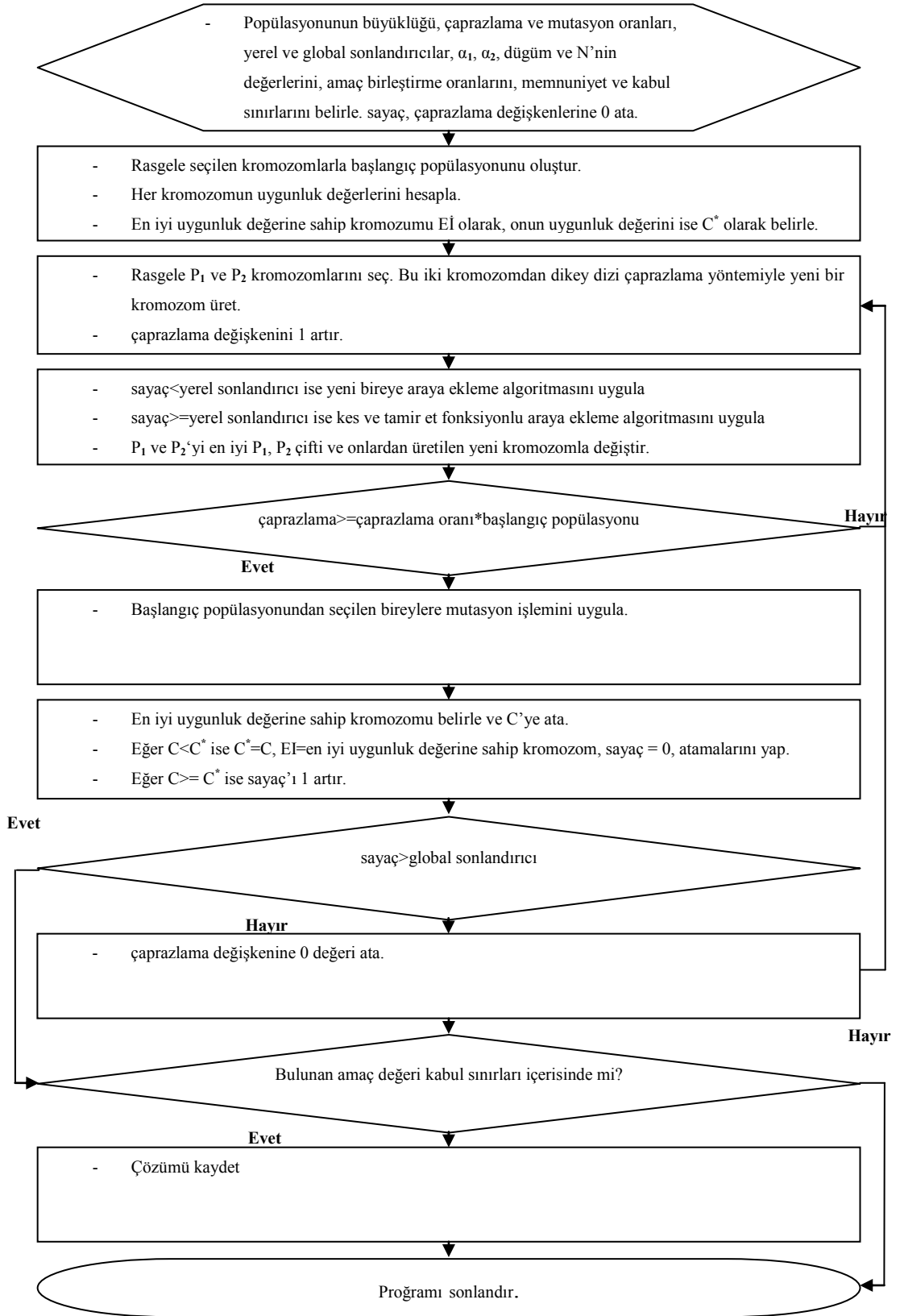
Geliştirilen algoritma, 2.2 GHz 64 bit AMD işlemcili, Windows Vista 32 bit işletim sistemli bilgisayarda; değerleri uniform dağılım gösteren, 192 farklı makine ve iş kombinasyonuna sahip problemler üzerinde test edilmiştir. Algoritmanın etkinliğini kanıtlamak için, Günaydın’ın (2008) C_{max} ’ı en küçüklemeyi hedefleyen BATP için geliştirmiş olduğu YSA algoritmasından sapma oranları, 4.1 ifadesinde gösterildiği gibi hesaplanmıştır (Günaydın, 2008).

$$C_{max}' \text{ in } \% \text{ Sapma Değeri} = \frac{\text{Karşılaştırılan sonuç} - \text{önerilen sonuç}}{\text{önerilen sonuç}} \times 100 \quad (4.1)$$

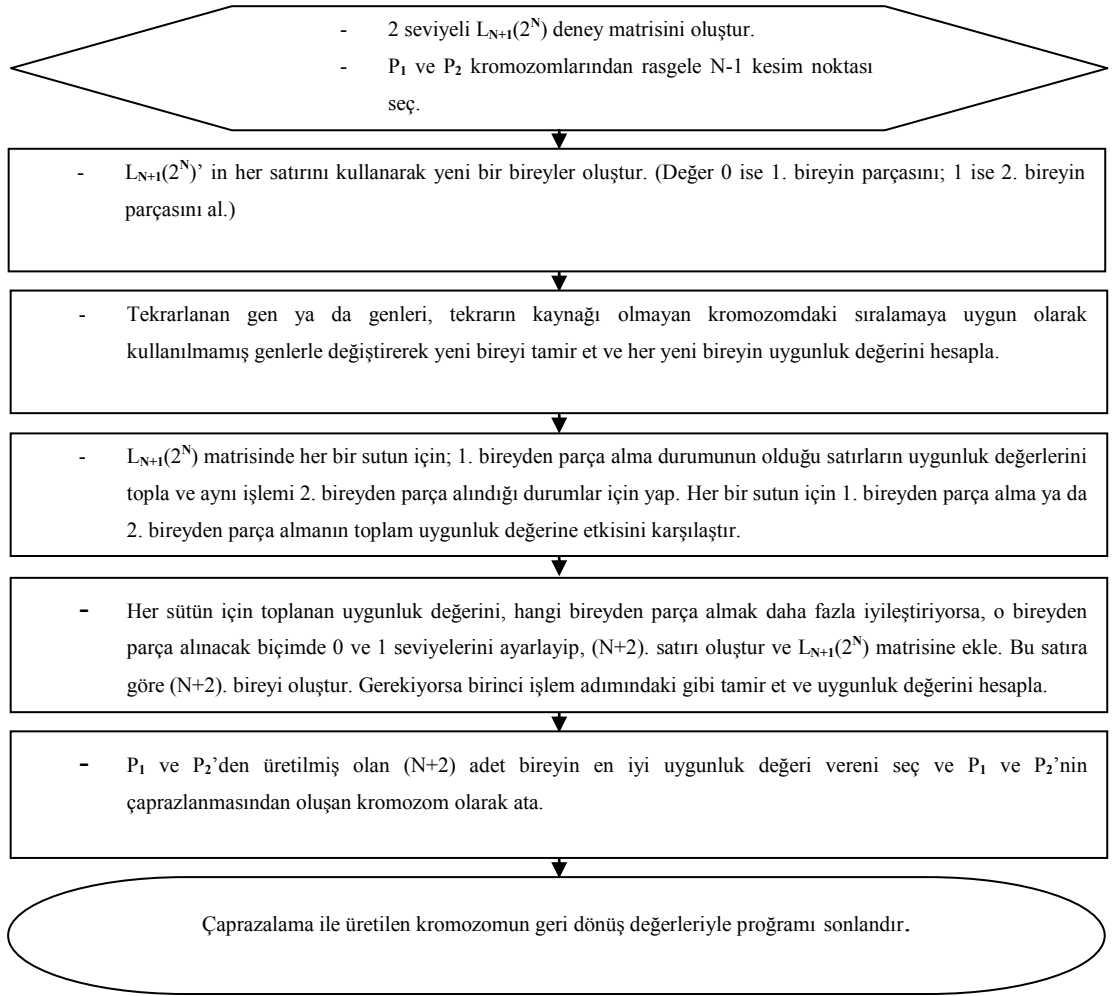
Algoritmanın etkinliđi kanıtlandıktan sonra, benzer problemler için çok amaçlı olarak çalıştırılmış ve Şekil 4.2' de verilen memnuniyet oranı fonksiyonuna göre analiz edilmiştir.

4.1.1 Geliştirilen çok amaçlı melez GA'nın yapısı

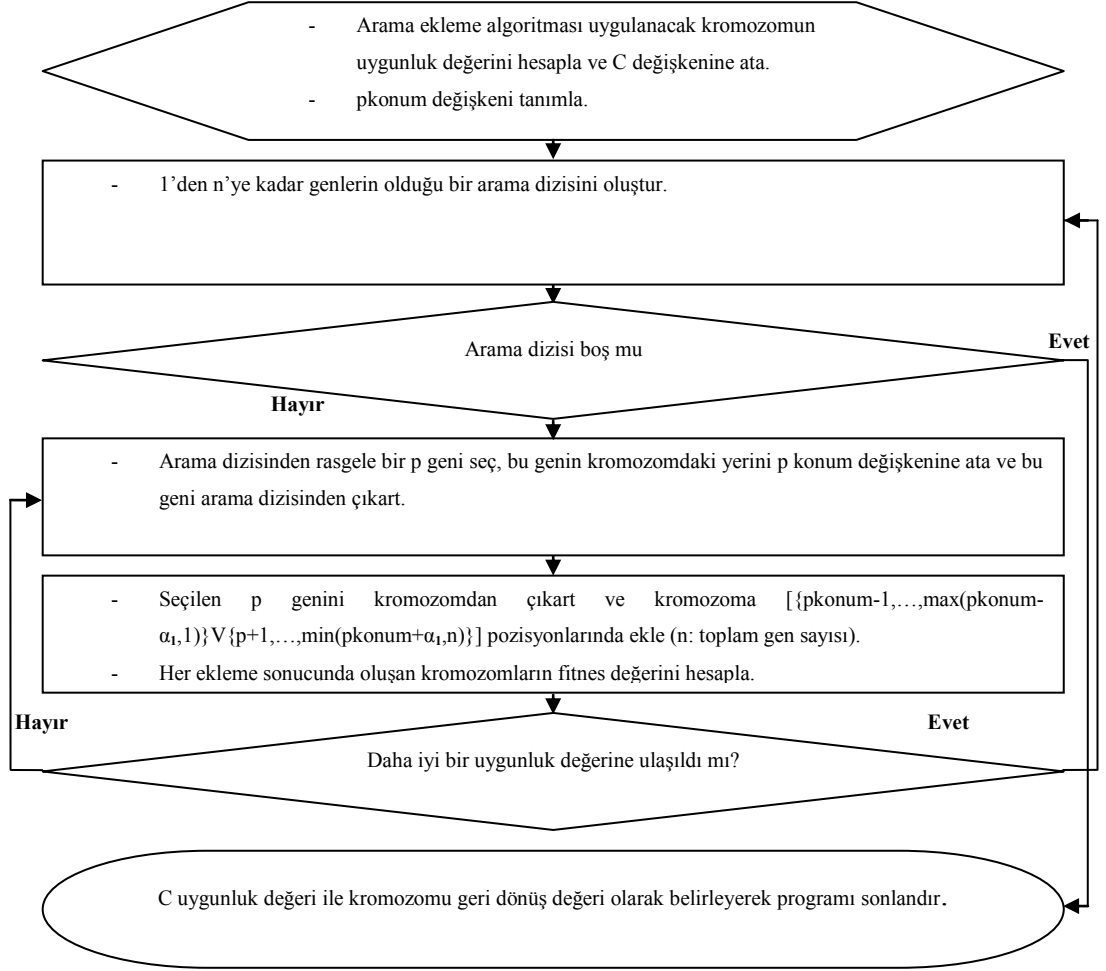
Bu kısımda, Şekil 4.3'de görüldüğü gibi, geliştirilen algoritmanın genel akış şeması ile alt basamaklarının akış şemaları verilmiştir. Şekil 4.5'deki ekleme arama algoritması (Insertion Search Algorithm) yerel arama algoritması olarak kullanılmış, Şekil 4.6'daki kes ve tamir et fonksiyonlu ekleme arama algoritması ise global arama algoritması olarak GA'ya eklenmiştir. Ayrıca, bu algoritmanın alt fonksiyonu olan, kes ve tamir et fonksiyonunun akış şeması Şekil 4.7'de görülmektedir. Bilinen çaprazlama yöntemlerinden farklı olarak, Şekil 4.4'de açıklanan dikey dizi çaprazlama yöntemi kullanılmıştır. Bu yöntem, kalite geliştirmek için kullanılan deney tasarımı matrisinden esinlenerek oluşturulmuştur. İki seviyeli tasarım matrisindeki her bir seviye, ebeveyn durumundaki iki kromozomdan hangisinden gen parçalarının alınacağını; her bir faktör de, kromozomun hangi konumdan kesileceğini göstermektedir (Tseng ve ark. 2009). Son olarak mutasyon akış şeması Şekil 4.8'de verilmiştir.



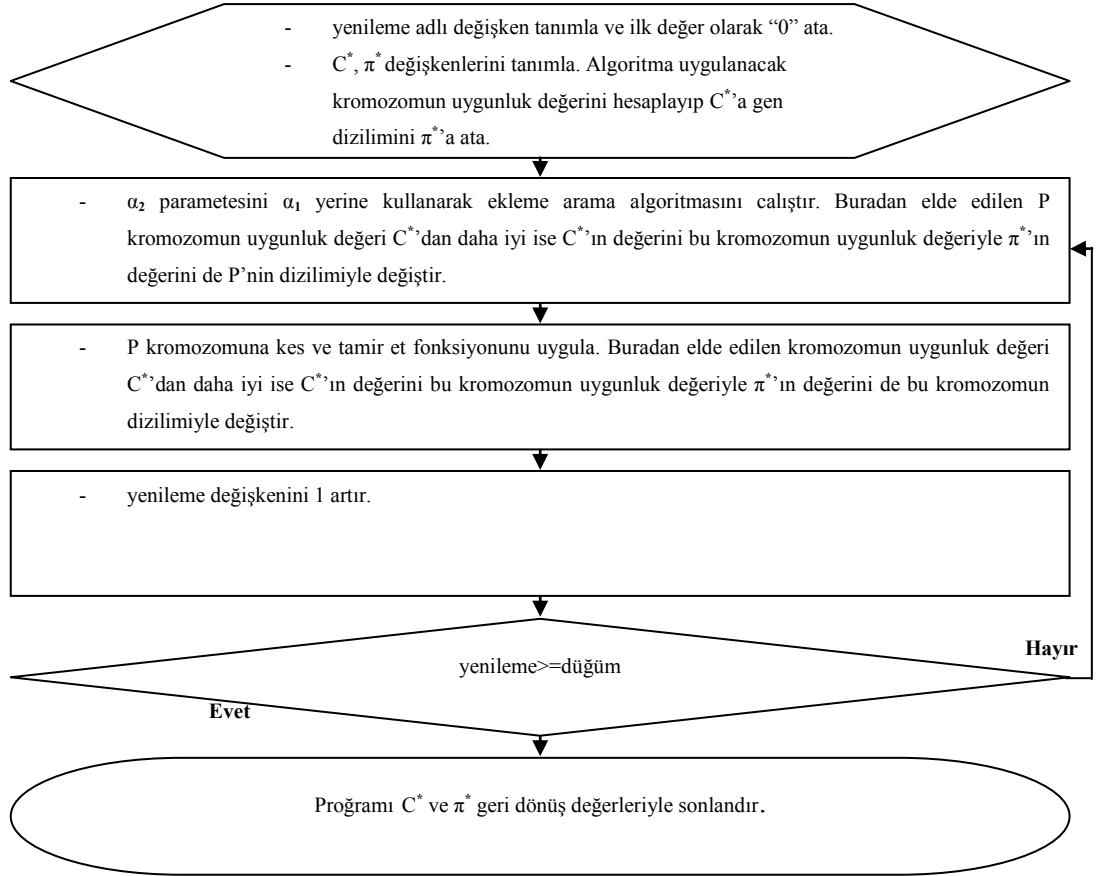
Şekil 4.3 Melez ÇAGA yapısı



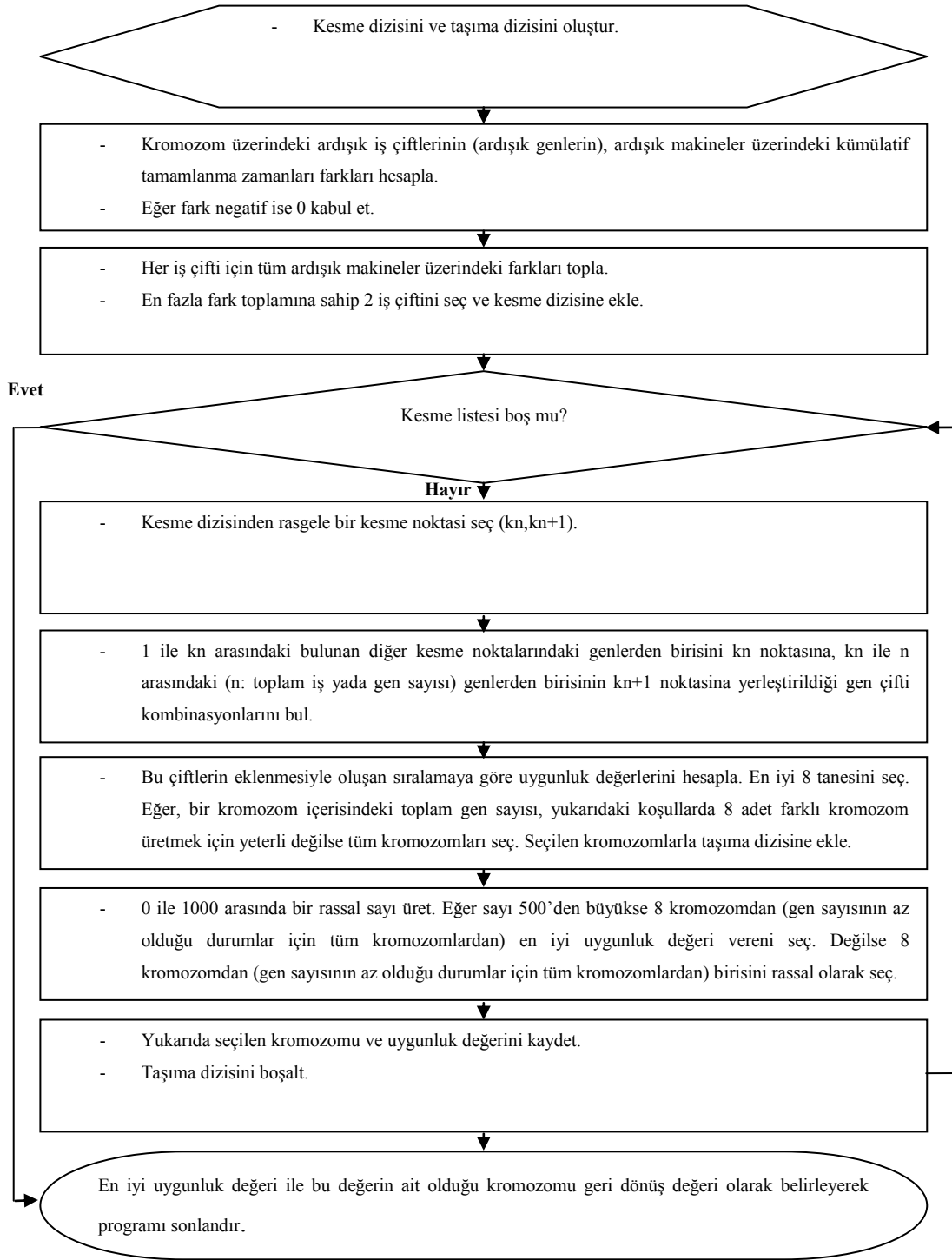
Şekil 4.4 Dikey dizi çaprazlama yöntemi



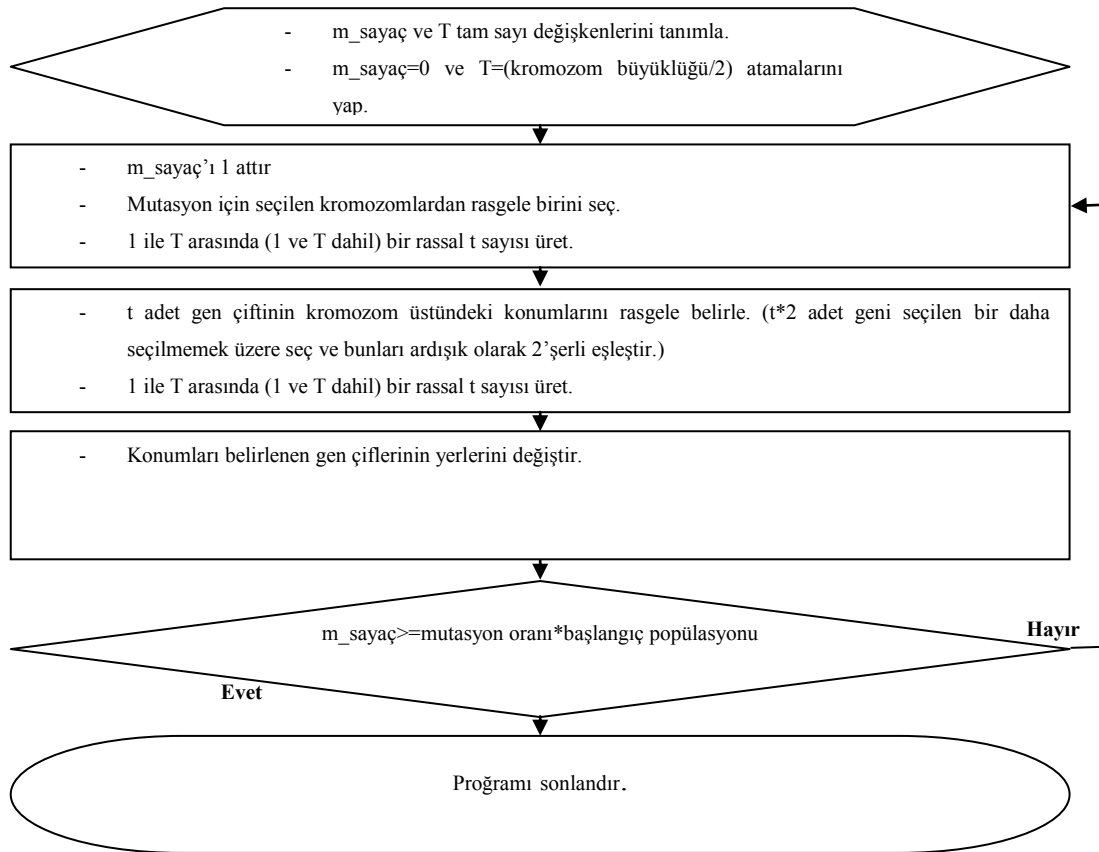
Şekil 4.5 Ekleme arama algoritması



Şekil 4.6 Kes tamir et fonksiyonlu ekleme arama algoritması



Şekil 4.7 Kes ve tamir et fonksiyonu



Şekil 4.8 Mutasyon akış şeması

4.2 Parametre Optimizasyonu

Geliştirilen melez GA'nın parametreleri seçilirken, test problemleri üzerinde, tüm anlamlı parametre olasılıkları denenmiştir. Buna göre, genellikle, parametre değerleri yükseldikçe, daha iyi çözüm bulma olasılığı da artmaktadır. Ancak bu durum CPU zamanlarını artırmaktadır. Bu nedenle parametrelerin artırılmasıyla iyileşen amaç fonksiyonu değerlerinin nispi faydası göz önünde bulundurulmuştur. Farklı büyüklükteki problemler için uygulanan koşullarda çaprazlama oranının 0,3'den fazla artırılmasının, sonucu, 0,3'ten öncesine göre daha az etkilediği saptanmıştır. Benzer biçimde yerel arama yöntemi ile gen komşulukları yoğun bir

şekilde incelendiğinden başlangıç popülasyonu çok düşük tutulmuştur. Geliştirilen melez çok amaçlı GA'da Tablo 4.1'deki parametrelerden faydalanılmıştır. Tablodaki parametreler hazırlanan problemin, her bir parametresine farklı değerler verilerek – diğer parametreler sabit tutulduğunda– sonucu hangi oranda etkilediği ve işlem süresini ne kadar değiştirdiği göz önünde bulundurularak seçilmiştir. Buna göre; örneğin, 0,3 olan çaprazlama oranı diğer parametreler sabitken 0,2 ve 0,1 değerlerinde kullanıldığında, sonuçlar nispeten kötüleşirken CPU sürelerinde aynı oranda azalmaya neden olmamıştır. Aynı şekilde 0,4 ve 0,5 seçildiğinde algoritma performansı, büyük problemlerde bile, kabul edilebilir derecede artmamıştır ve işlemci sürelerinin YSA ile kıyaslanamayacak düzeyde olduğu gözlenmiştir. Başlangıç popülasyonunun da 10 seçildiği düşünülürse bir ondalık basamağa sahip olan 0,3 çaprazlama oranının uygun parametre olacağına karar verilmiştir. Diğer parametrelerin belirlenmesinde de; parametrelerin artması ya da azalmasıyla, algoritma performansının ve CPU zamanlarının ne kadar arttığı ya da azaldığı dikkate alınmıştır. Burada dikkate alınan başka bir hususta, Tseng ve ark.'nın (2009) çalışmalarında kullanmış olduğu parametreler ve bunların birbirleriyle olan ilişkileridir.

Tablo 4.1 Seçilen melez çok amaçlı GA parametreleri

Fonksiyon	Parametre Değeri
Başlangıç Popülasyonu	10
Çaprazlama Oranı	0,3
Mutasyon Oranı	0,1
Yerel Sonlandırıcı	3
Global Sonlandırıcı	6
Düğüm Sayısı	1
N (kesme noktası)	3
α_1	2
α_2	4

4.3. YSA ile Önerilen Melez ÇAGA'nın Karşılaştırılması

Çözülen problemlerin sonuçları ile YSA yöntemiyle, Günaydın (2008) tarafından çözülen problemlerinin C_{max} değerleri karşılaştırılmıştır. Test problemleri, Aldowaisan ve Allahverdi ile Shyu ve ark. tarafından oluşturulan problemlerin verileri referans alınarak oluşturulmuştur. İş sayıları, $n = 8, 10, 12, 50, 100, 150, 200, 250$ olmak üzere 8 farklı iş sayısı olarak belirlenmiştir. Makine sayıları ise $m = 2, 3, 5, 8, 10, 15, 20, 25$ olarak seçilmiştir. Makinelerin hazırlık süreleri $[1,10]$ arasında uniform dağılıma göre rastsal olarak üretilmiştir. İşlem zamanları için; üç çeşit işlem zamanı seçilmiştir. Birinci tip işlem zamanı a ; $[1,10]$ arasında, ikinci tip işlem zamanı b ; $[1,50]$ arasında ve üçüncü tip işlem zamanı c ; $[1,100]$ arasında uniform dağılıma göre rastsal olarak üretilmiştir. Bu şekilde $(m \times n \times 3)$ olarak toplam $8 \times 8 \times 3 = 192$ problem rastsal olarak geliştirilmiştir. Her bir tablo, makine sayısı bakımından farklılık göstermektedir. Her makine ve iş kombinasyonu için 1-10 dakika arası işlem süreleri olan, 1-50 dakika arasında işlem süreleri olan ve 1-100 dakika arasında işlem süreleri olan 3 tip problem rastsal olarak üretilmiştir (Günaydın, 2008).

Tablo 4.2 2 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
2	8	0-10	0
		0-50	0
		0-100	1
	10	0-10	0
		0-50	2
		0-100	1
	12	0-10	1
		0-50	0
		0-100	1
	50	0-10	5
		0-50	9
		0-100	11
	100	0-10	8
		0-50	13
		0-100	17
	150	0-10	8
		0-50	15
		0-100	17
	200	0-10	8
		0-50	15
		0-100	16
250	0-10	9	
	0-50	16	
	0-100	17	

Tablo 4.2'ye göre; 2 makineli iş kombinasyonlarında, 4 iş için aynı C_{max} değerleri bulunmuş, işlerin 20'sinde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.3 3 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
3	8	0-10	0
		0-50	5
		0-100	1
	10	0-10	3
		0-50	1
		0-100	3
	12	0-10	3
		0-50	1
		0-100	5
	50	0-10	9
		0-50	15
		0-100	14
	100	0-10	10
		0-50	18
		0-100	18
	150	0-10	12
		0-50	17
		0-100	22
	200	0-10	12
		0-50	20
		0-100	22
	250	0-10	13
		0-50	20
		0-100	22

Tablo 4.3'e göre; 3 makineli iş kombinasyonlarında, 1 iş için aynı C_{max} değerleri bulunmuş, işlerin 23'ünde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.4 5 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
5	8	0-10	0
		0-50	1
		0-100	0
	10	0-10	1
		0-50	11
		0-100	4
	12	0-10	7
		0-50	7
		0-100	7
	50	0-10	11
		0-50	19
		0-100	19
	100	0-10	13
		0-50	22
		0-100	22
	150	0-10	14
		0-50	22
		0-100	23
	200	0-10	15
		0-50	21
		0-100	26
	250	0-10	15
		0-50	22
		0-100	27

Tablo 4.4'e göre; 5 makineli iş kombinasyonlarında, 2 iş için aynı C_{max} değerleri bulunmuş, işlerin 22'sinde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.5 8 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
8	8	0-10	0
		0-50	0
		0-100	0
	10	0-10	1
		0-50	9
		0-100	10
	12	0-10	5
		0-50	6
		0-100	5
	50	0-10	14
		0-50	20
		0-100	21
	100	0-10	15
		0-50	24
		0-100	23
	150	0-10	14
		0-50	23
		0-100	27
	200	0-10	16
		0-50	23
		0-100	24
	250	0-10	16
		0-50	26
		0-100	26

Tablo 4.5'e göre; 8 makineli iş kombinasyonlarında, 3 iş için aynı C_{max} değerleri bulunmuş, işlerin 21'inde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.6 10 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
10	8	0-10	1
		0-50	0
		0-100	1
	10	0-10	5
		0-50	11
		0-100	6
	12	0-10	9
		0-50	9
		0-100	7
	50	0-10	11
		0-50	17
		0-100	21
	100	0-10	16
		0-50	22
		0-100	23
	150	0-10	16
		0-50	23
		0-100	26
	200	0-10	18
		0-50	24
		0-100	26
	250	0-10	18
		0-50	23
		0-100	26

Tablo 4.6'ya göre; 10 makineli iş kombinasyonlarında, 1 iş için aynı C_{max} değerleri bulunmuş, işlerin 23'ünde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.7 15 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
15	8	0-10	0
		0-50	1
		0-100	0
	10	0-10	2
		0-50	4
		0-100	5
	12	0-10	9
		0-50	11
		0-100	10
	50	0-10	13
		0-50	15
		0-100	21
	100	0-10	17
		0-50	22
		0-100	26
	150	0-10	19
		0-50	27
		0-100	26
	200	0-10	18
		0-50	24
		0-100	29
	250	0-10	19
		0-50	27
		0-100	27

Tablo 4.7'ye göre; 15 makineli iş kombinasyonlarında, 2 iş için aynı C_{max} değerleri bulunmuş, işlerin 22'sinde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.8 20 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
20	8	0-10	0
		0-50	1
		0-100	0
	10	0-10	3
		0-50	4
		0-100	4
	12	0-10	6
		0-50	9
		0-100	6
	50	0-10	11
		0-50	21
		0-100	23
	100	0-10	16
		0-50	28
		0-100	21
	150	0-10	18
		0-50	26
		0-100	27
	200	0-10	16
		0-50	27
		0-100	26
	250	0-10	20
		0-50	26
		0-100	28

Tablo 4.8'e göre; 20 makineli iş kombinasyonlarında, 2 iş için aynı C_{max} değerleri bulunmuş, işlerin 22'sinde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Tablo 4.9 25 makineli iş kombinasyonu için çözümlerin karşılaştırılması

Makine Sayısı	İş Sayısı	Rastsal Süreler	Melez ÇAGA Sapma%
25	8	0-10	0
		0-50	2
		0-100	0
	10	0-10	3
		0-50	4
		0-100	2
	12	0-10	7
		0-50	9
		0-100	10
	50	0-10	11
		0-50	15
		0-100	17
	100	0-10	20
		0-50	27
		0-100	24
	150	0-10	20
		0-50	24
		0-100	28
	200	0-10	17
		0-50	26
		0-100	28
	250	0-10	21
		0-50	25
		0-100	29

Tablo 4.9'a göre; 25 makineli iş kombinasyonlarında, 2 iş için aynı C_{max} değerleri bulunmuş, işlerin 22'sinde çok amaçlı melez GA daha iyi sonuç vermiştir. İş sayısı arttıkça sapma oranı da artmaktadır.

Buraya kadarki karşılaştırma tablolarını özetleyecek olursak, 192 problemde yalnızca 17'sinde YSA ile çok amaçlı melez GA aynı sonucu vermiştir. Bunlarda biri 10 iş diğeri 12 iş problemidir. Her iki problemde 2 makinelidir. Diğer 15 problem ise 8 işli problemlerdir. Bunlar haricindeki 175 problemde çok amaçlı melez GA daha

üstündür. Problem büyüklüğü arttıkça bu üstünlükte artmaktadır. Bu durum çok amaçlı melez GA'nın etkinliğini kanıtlamaktadır.

4.4 Çok Amaçlı Uygulamalar ve Memnuniyet Fonksiyonu

Bu bölümde C_{max} ve F_i değerleri için, çok amaçlı melez GA uygulanmıştır. Memnuniyeti tamamen sağlayan %0 - %1 sapma aralığı, tam memnuniyet aralığı; memnuniyetin %100'den %0'a doğrusal olarak azaldığı bölge olan %1 - %3 aralığı, memnuniyet aralığı; %3 - %5 aralığı, kabul aralığı ve %5'ten sonrası ise ret sınırının geçilmesi olarak tanımlanmıştır. Oluşturulan tablolarda C_{max} ve F_i değerleri için tek amaçlı olarak bulunan çözümlerden sapma oranları verilmiştir. Bu tablolarda tüm problemler için C_{max} ve F_i amaçları aynı öneme sahiptir. Amaç birleştirme ağırlıkları birbirine eşittir.

Tablo 4.10 2 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
2	8	0-10	0	0
		0-50	0	0
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	0
	10	0-10	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-50	0	2
		0-100	0	1
	12	0-10	0	1
		0-50	0	0
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	50	0-10	0	5
		0-50	0	10
		0-100	0	11
	100	0-10	0	7
		0-50	0	14
		0-100	4	17
	150	0-10	0	8
		0-50	0	16
		0-100	0	15
	200	0-10	1	10
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-100	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	250	0-10	0	9
		0-50	0	16
0-100		1	18	

Tablo 4.10'a göre; 2 makineli iş kombinasyonları için, F_i sonuçlarının 23'ü tam memnuniyet aralığında, 1 tanesi kabul aralığındadır. C_{max} sonuçlarının 10 tanesi tam memnuniyet aralığında, 1 tanesi memnuniyet aralığında, 1 tanesi kabul aralığında, 12 tanesi ise ret sınırını geçmiştir.

Tablo 4.11 3 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
3	8	0-10	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-50	1	3
		0-100	0	1
	10	0-10	1	1
		0-50	1	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	12	0-10	0	2
		0-50	0	0
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	50	0-10	0	9
		0-50	0	14
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	16
	100	0-10	0	11
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	14
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	18
	150	0-10	1	11
		0-50	0	19
		0-100	3	23
	200	0-10	0	12
		0-50	0	19
		0-100	2	23
250	0-10	0	13	
	0-50	1	19	
	0-100	0	21	

Tablo 4.11'e göre; 2 makineli iş kombinasyonları için, F_i sonuçlarının 22 tanesi tam memnuniyet aralığında, 2 tanesi memnuniyet aralığındadır. C_{max} sonuçlarının 7 tanesi tam memnuniyet aralığında, 2 tanesi memnuniyet aralığında, 15 tanesi ret sınırını geçmiştir.

Tablo 4.12 5 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
5	8	0-10	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	1
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	10	0-10	3	0
		0-50	0	9
		0-100	3	2
	12	0-10	1	6
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	5
		0-100	2	7
	50	0-10	2	9
		0-50	0	19
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	17
	100	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	13
		0-50	4	24
		0-100	1	25
	150	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	13
		0-50	0	19
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	20
	200	0-10	1	15
		0-50	0	23
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
250	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	14	
	0-50	1	23	
	0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	21	

Tablo 4.12'ye göre; 5 makineli iş kombinasyonları için, F_i sonuçlarının 19 tanesi tam memnuniyet aralığında, 4 tanesi memnuniyet aralığında, 1 tanesi kabul aralığındadır. C_{max} sonuçlarının 5 tanesi tam memnuniyet aralığında, 1 tanesi memnuniyet aralığında, 1 tanesi kabul aralığında, 17 tanesi ise ret sınırını geçmiştir.

Tablo 4.13 8 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
8	8	0-10	0	0
		0-50	1	Çok amaçlı yapı daha iyi sonuç verdi.
		0-100	0	Çok amaçlı yapı daha iyi sonuç verdi.
	10	0-10	0	0
		0-50	0	1
		0-100	1	10
	12	0-10	1	1
		0-50	4	Çok amaçlı yapı daha iyi sonuç verdi.
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	5
	50	0-10	Çok amaçlı yapı daha iyi sonuç verdi.	9
		0-50	1	17
		0-100	0	17
	100	0-10	1	0
		0-50	Çok amaçlı yapı daha iyi sonuç verdi.	0
		0-100	1	0
	150	0-10	3	15
		0-50	0	23
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	25
	200	0-10	Çok amaçlı yapı daha iyi sonuç verdi.	16
		0-50	0	24
		0-100	1	26
250	0-10	Çok amaçlı yapı daha iyi sonuç verdi.	15	
	0-50	2	25	
	0-100	Çok amaçlı yapı daha iyi sonuç verdi.	22	

Tablo 4.13'e göre; 8 makineli iş kombinasyonları için, F_i sonuçlarının 21 tanesi tam memnuniyet aralığında, 2 tanesi memnuniyet aralığında, 1 tanesi kabul aralığındadır. C_{max} sonuçlarının 10 tanesi tam memnuniyet aralığında, 1 tanesi kabul aralığında, 13 tanesi ise ret sınırını geçmiştir.

Tablo 4.14 10 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
10	8	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	1
		0-50	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-100	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	10	0-10	0	3
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	4
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	12	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	5
		0-50	3	0
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	3
	50	0-10	1	11
		0-50	5	19
		0-100	3	20
	100	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	13
		0-50	0	21
		0-100	4	25
	150	0-10	0	16
		0-50	1	24
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	26
	200	0-10	3	17
		0-50	0	23
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	29
	250	0-10	1	17
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	25
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	27

Tablo 4.14'e göre; 10 makineli iş kombinasyonları için, F_i sonuçlarının 19 tanesi tam memnuniyet aralığında, 3 tanesi memnuniyet aralığında, 2 tanesi memnuniyet aralığındadır. C_{max} sonuçlarının 5 tanesi tam memnuniyet aralığında, 2 tanesi memnuniyet aralığında, 2 tanesi kabul aralığında, 15 tanesi ret sınırını geçmiştir.

Tablo 4.15 15 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}
15	8	0-10	0	Çok amaçlı yapı daha iyi sonuç verdi.
		0-50	0	1
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	Çok amaçlı yapı daha iyi sonuç verdi.
	10	0-10	0	Çok amaçlı yapı daha iyi sonuç verdi.
		0-50	0	Çok amaçlı yapı daha iyi sonuç verdi.
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	2
	12	0-10	4	8
		0-50	Çok amaçlı yapı daha iyi sonuç verdi.	Çok amaçlı yapı daha iyi sonuç verdi.
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	1
	50	0-10	0	11
		0-50	2	13
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	15
	100	0-10	Çok amaçlı yapı daha iyi sonuç verdi.	13
		0-50	2	Çok amaçlı yapı daha iyi sonuç verdi.
		0-100	6	24
	150	0-10	2	17
		0-50	0	24
		0-100	3	23
	200	0-10	0	18
		0-50	0	26
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	24
	250	0-10	1	20
		0-50	1	26
		0-100	Çok amaçlı yapı daha iyi sonuç verdi.	27

Tablo 4.15'e göre; 15 makineli iş kombinasyonları için, F_i sonuçlarının 18 tanesi tam memnuniyet aralığında, 4 tanesi memnuniyet aralığında, 1 tanesi kabul aralığında, 1 tanesi de ret sınırını geçmiştir. C_{max} sonuçlarının 6 tanesi tam memnuniyet aralığında, 1 tanesi memnuniyet aralığında, 17 tanesi ise ret sınırını geçmiştir.

Tablo 4.16 20 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma % F_i	Tek Amaçtan Sapma % C_{max}
20	8	0-10	0	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-50	0	1
		0-100	0	0
	10	0-10	1	0
		0-50	0	4
		0-100	2	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	12	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>
	50	0-10	2	12
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	12
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	14
	100	0-10	3	18
		0-50	1	24
		0-100	2	27
	150	0-10	1	19
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	22
		0-100	2	31
	200	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	17
		0-50	0	27
		0-100	1	30
250	0-10	0	21	
	0-50	2	28	
	0-100	2	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	

Tablo 4.16'ya göre; 20 makineli iş kombinasyonları için, F_i sonuçlarının 17 tanesi tam memnuniyet sağlamış, 7 tanesi memnuniyet aralığındadır. C_{max} 9 tanesi tam memnuniyet aralığında, 1 tanesi kabul aralığında, 14 tanesi ise ret sınırını geçmiştir.

Tablo 4.17 25 makineli iş kombinasyonlarının tek amaçlı çözümden sapmaları

Makine Sayısı	İş Sayısı	Rastsal Süreler	Tek Amaçtan Sapma %F _i	Tek Amaçtan Sapma %C _{max}	
25	8	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
		0-50	2	2	
		0-100	0	10	
	10	0-10	0		0
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
		0-100	1	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
	12	0-10	1		7
		0-50	1	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
		0-100	2	8	
	50	0-10	2		16
		0-50	2	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	18	
	100	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>		13
		0-50	3		30
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>	
	150	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>		16
		0-50	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>		26
		0-100	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>		26
	200	0-10	2		21
		0-50	2		24
		0-100	1		29
250	0-10	<i>Çok amaçlı yapı daha iyi sonuç verdi.</i>		20	
	0-50	0		29	
	0-100	1		28	

Tablo 4.17'ye göre; 25 makineli iş kombinasyonları için, F_i sonuçlarının tamamı kabul edilmiş, 17 tanesi tam memnuniyet aralığında, 7 tanesi memnuniyet aralığındadır. C_{max} sonuçlarının 7 tanesi tam memnuniyet aralığında, 1 tanesi memnuniyet aralığında, 16 tanesi ise ret aralığındadır.

Bu sonuçları Tablo 4.18'deki gibi özetleyebiliriz. Buna göre;

Tablo 4.18 Çok amaçlı yapıdaki F_i ve C_{max} değerlerinin tek amaçlı çözümden sapmaları

Sapma F_i	Makine Sayısı								Toplam	Ortalama
	2	3	5	8	10	15	20	25		
Tam memnuniyet ($\leq\%1$)	23	22	19	21	19	18	17	17	156	19,50
Memnun edici ($\%1-\%3$]	0	2	4	2	3	4	7	7	29	3,63
Kabul ($\%3-\%5$]	1	0	1	1	2	1	0	0	6	0,75
Ret ($\geq\%5$)	0	0	0	0	0	1	0	0	1	0,13
Sapma C_{max}	Makine Sayısı								Toplam	Ortalama
	2	3	5	8	10	15	20	25		
Tam memnuniyet ($\leq\%1$)	10	7	5	10	5	6	9	7	59	7,38
Memnun edici ($\%1-\%3$]	1	2	1	1	2	1	0	1	9	1,13
Kabul ($\%3-\%5$]	1	0	0	0	2	0	1	0	4	0,50
Ret ($\geq\%5$)	12	15	17	13	15	17	14	16	119	14,88

- C_{max} değerleri problem belirli bir büyüklüğe eriştiğinde benzer ret oranlarına sahip olmaktadır. Kalanlar yüksek oranda tam memnuniyet sağlayan çözümler olmaktadır.
- C_{max} değerleri genellikle problem çapı büyüdükçe tek amaçlılık için elde ettiği çözümlerden yüksek oranda saptmaktadır.
- C_{max} değerleri için nadiren de olsa çok amaçlı yapı daha iyi sonuçlar elde edilmesini sağlayabilir.
- F_i değerleri çok amaçlı yapıdan fazla etkilenmemektedir. Bulunan sapmalar C_{max} 'a göre daha karardır ve daha dar bir aralıkta dağılmaktadır.
- F_i için tüm sonuçlar gözlemlendiğinde çok amaçlılık tek amaçlılığa göre daha iyi sonuç elde edilmesini sağlamıştır.
- F_i değerleri çok yüksek oranda kabul sınırı içerisinde çıkmış, sonuçların çoğu tam memnuniyet sağlamıştır.

5. SONUÇ VE ÖNERİLER

GA'lar çok farklı problemlere uyarlanabilen, sayısız uygulaması olan ve gücünü doğal hayatın devamlılığından alan çok etkili ve verimli yöntemlerdir. Bu yöntemlerin etkinliği, doğal yaşamın veya canlıların bazı tipik özelliklerinin taklit edilmesi ve modellenmesiyle oluşan çok sayıda sezgiselin üretilmesine neden olmuştur. Zamanla, bu sezgiseller de geliştirilmiş, etkin olan kısımları birbirleriyle kombine edilerek, bazen de, olasılıklı ya da belirleyici eklemeler veya yenilemeler yaparak çeşitlendirilmiştir. Özellikle, belirleyici yöntemlerle çözülmesi çok zor olan problemlere, zaman zaman optimal; büyük çaptaki problemler için de, optimale yakın çözüm bulunmasında çok verimli araçlardır.

BATP, farklı süreç guruplarının çizelgelenmesinde sıklıkla kullanılan bir model haline gelmiştir ve son yıllarda önemi daha da artmıştır. Ancak bu tip problemlerin doğal yapısı, çok hızlı artan sayıda olasılıkların test edilmesini gerektirmektedir.

Literatürde BATP için çok sayıda amaç üzerinde çalışılmıştır. Çoğu zaman bu amaçların birlikte kullanıldığı modellere rastlamaktayız.

Bu çalışmada, çok amaçlı bir yapıdaki BATP'ye optimale yakın bir çözüm elde edilmeye çalışıldı. Bunun için ÇAGA yapısını temel alındı. Ancak araştırmalar sonucunda, GA'ya yapılan bazı eklemelerin ve algoritma yapısındaki değişikliklerin sonuçların daha kısa sürede ve daha iyi uygunluk değerleriyle elde edilmesini sağladığı tespit edilmiştir. Bu nedenle GA'daki üreme operatörü algoritmadan çıkartılmış, yerel ve global arama yöntemleri eklenmiştir. Hazırlanan çok amaçlı melez GA'nın, benzer BATP üzerinde, YSA ile C_{max} amacı bakımından kıyaslanarak etkinliği kanıtlanmıştır. Etkinliği kanıtlanan çok amaçlı melez GA; aynı miktarda

ağırlık verilmiş, F_i ve C_{\max} kriterleri için benzer problemler seti için çözülmüş, bu amaçların tek amaçlı yapıdaki çözümlerden sapmaları analiz edilmiştir. Ayrıca, bu amaçlara göre çizelgeleme yapacak olan olası bir yönetici için memnuniyet fonksiyonu tanımlanmış, makine sayılarına göre çok amaçlı yapı ile yöneticinin belirlediği kriterler arasındaki ilişki gözlemlenmiştir.

Uygulama tabloları analiz edildiğinde, bazı problemde, çok amaçlı yapıda tek amaçlı ya göre daha iyi sonuç elde edilebilmesi gerçekten manidardır. Geliştirilen algoritmanın yoğun bir yerel arama yapması ve iş komşuluklarını yüksek oranda kontrol ediyor olması, rastsallığın yoğun olduğu algoritmalarda, yüksek seçim baskısının yerel optimum noktalara takılmada ciddi bir tehlike olabileceğini düşündürmüştür. Öyle ki; tek bir amaç üzerindeki seçim baskısının azaltılması, hem bu amacı hem de diğer amacı iyileştirmiştir.

Bu çalışma, BATP literatüründe kullanılan diğer amaçlar için ve bunların, olası bir yönetici memnuniyet fonksiyonu arasındaki ilişkinin analizi için çeşitlendirilebilir. Yine çok amaçlı melez GA yapısı ve GA operatörlerinin etkinliği; BATP için, deney tasarımı yöntemiyle elde edilen çaprazlama operatöründe olduğu gibi, optimize edilebilir ve yeni operatörler tanımlanabilir. Bu durum BATP’de yeni ve daha etkin çok amaçlı melez GA yapılarının elde edilmesini sağlayacaktır.

6. KAYNAKLAR

- Agnētis, A., 1997. No-wait flow shop scheduling with large lot sizes, *Annals of Operations Research*, 70:415–438.
- Allahverdi, A., Aldowaisan, Tariq., 2004. No-wait flowshops with bicriteria of makespan and maximum lateness, *European Journal of Operational Research* 152:132–147.
- Allouche, M.A., Aouni, B., Martel, J.M., Loukil T., Rebaï, A., 2009. Solving multi-criteria scheduling flow shop problem through compromise programming and satisfaction functions, *European Journal of Operational Research*, 192:460-467.
- Arroyo, J. E. C., Armentano VA, 2005. Genetic local search for multi-objective flowshop scheduling problems., *Eur J Oper Res*, 167,717–738.
- Asiabar, M.H., Ghodsypour, S.H., Kerachian R., 2009. Multi objective genetic local search algorithm using Kohonen's neural map, *Computers & Industrial Engineering* 56:1566-1576.
- Bianco, L., Dell'olmo, P., Giordani, S., 1999. Flow shop no wait scheduling with sequence dependent setup times and dates, *INFOR* vol. 37:3-19.
- Blecic, I., Cecchini, A., Trunfio, G.A., 2007. A decision support tool coupling a casual model and a multi objective genetic algorithm, *Appl Intell*, 26:125-137.
- Chen A., Subprasom, K., Ji Z., 2006. A simulation based multi objective genetic algorithm (SMAGO) procedure for BOT network design problem, *Optim. Eng.* 7:225-227.
- Czogalla, J., Fink A., 2009. Design and analysis of evolutionary algorithms for the no-wait flow shop scheduling problem, *Metaheuristics in the Service Industry, Lecture Notes in Economics and Mathematical Systems* 624:99-126.
- Darwin, C. A., 1906. *On the origin of species by means of natural selection*, London.
- Eldredge, N. A., 1985. *Time frames: The rethinking of darwinian evolution and the theory of punctuated equilibrium.*, NY.
- Emmons, H., Mathur, K., 1995. Lot sizing in a no-wait flow shop, *Operations Research Letters* 17:159-164.
- Engin, O., 2001. Akış tipi problemlerin genetik algoritma ile çözüm performansının artırılmasında parametre optimizasyonu, *Doktora Tezi, Fen Bilimleri Enstitüsü, İstanbul Teknik Üniversitesi.*
- Espinouse, M.L., Formanowicz, P., Penz, B., 1999. Minimizing the makespan in two machine no wait flow shop with limited machine availability, *Computers & Industrial Engineering* 37:497-500.
- Espinouse, M.L., Formanowicz, P., Penz, B., 2001. Complexity results and approximation algorithms for two machine no wait flow shop limited machine availability, *Journal of Operation Research Society*, 52:161-121.
- Filho, G.B., Negano, M.,S., Lorena, L.A.N., 2007. Hybrid evolutionary algorithm for flowtime minimization no wait flowsop scheduling, *Mexican International Conference on Artificial Intelligence*, 1099-1109.
- Glass, C.A., Gupta, J.N.D., Potts, C.N., 1999. Two machine no wait flow shop scheduling with missing operations, *Mathematics of Operations Research* 24:911-924.
- Grabowski, J., Pempera, J., 2005. Some local search algorithms for no-wait flow

- shop problem with makespan criterion, *Computers & Operations Research* 32:2197–2212.
- Günaydın, C., 2008. Beklemesiz akış tipi problemlerin çözümünde yapay sinir ağları yaklaşımı, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Selçuk Üniversitesi.
- Holland, J.H., 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*, University of Michigan Press, Ann Harbor, Michigan.
- Ishibuchi, H., Yoshida, T., Murata., T., 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling., *IEEE T Evol Comput* 7,204–223.
- Javadi, B., Mehrabad, M.S, Haji, A., Mahdavi, I., Jolai, F., Amiri, N.M., 2008. No-wait flow shop scheduling using fuzzy multi-objective linear programming, *Journal of the Franklin Institute* 345:452–467.
- Jolai, F., Sheikh, S., Rabbani, M., Karimi, B., 2009. A genetic algorithm for solving no wait flexible flow lines with due window and job rejection, *Int. J. Adv. Manuf. Technol.*, 42:523-532.
- Kachitvichyanukul, V., Sitthitham., S., 2009. A two stage genetic algorithm for multi objective job shop scheduling problems, *J. Intell Manuf*, 20:611–623.
- Kalczynski, P.J., Kamburowski., J., 2007. On no-wait and no-idle flow shops with makespan criterion, *European Journal of Operational Research* 178: 677–685.
- Kaya., M., 2006. Multi objective genetic algorithm based approaches for mining optimized fuzzy association rules, *Soft. Comput.*, 10:578-586.
- Khoo, L.P., Lee, S.G.,Yin, X.F., 2000. A prototype genetic algorithm enhanced multi objective scheduler for manufacturing systems, *Int. J. Adv. Manuf. Technol.* 16:131-138.
- Kim, K., Jeong, I.J., 2009. Flow shop scheduling with no-wait flexible lot streaming using an adaptive genetic algorithm, *Int. J. Adv. Manuf. Technol.* 44:1181–1190.
- Kayabashi. K., Hiroyasu T., Miki, M., 2007. Mechanism of multi objective genetic algorithm for maintaining the solution diversity using neural network, *EMO*, 216-226.
- Kubzin, M.A., Strusevich, V.A., 2005. Two-machine flow shop no-wait scheduling with machine maintenance, *A Quarterly Journal of Operations Research* 3:303–313.
- Kurt, M., Semetay, C., 2001. Genetik algoritma ve uygulama alanları, mühendis ve makine, 42(501), 19-24.
- Ladd, R. S., 1996. *Genetic Algorithms in C++*, NY
- Laha, D., Chakraborty, U. K., 2008. A constructive heuristic for minimizing makespan in no-wait flow shop scheduling problem, *Int J Adv Manuf Technol.* 41,97-109.
- Lee, S. C., Hsu, T.S., Chauang, S. P., Yang, C. L., 2008. Enhanced performance of overlap flow-shop scheduling involving reworking and a time buffer, *Int J Adv Manuf Technol.* 39,141-150.
- Lei T., Lieli, L., Liyan., H., Huang H., 2006. A genetic algorithm based double objective multi constraint optimal cross region cross sector public investment model, *International Conference on Neural Computation*, 470-479.
- Liu, B., Wang, L., Jin, Y.H., 2007. An effective hybrid particle swarm optimization for no wait flow shop scheduling, *Int. J. Adv. Manuf. Technol.* 31:1001–1011.

- Luque M., Cordon, O., Viedma, E.H., 2006. A multi objective genetic algorithm for learning linguistic persistent queries in text retrieval environments, *Studies in Computational Intelligence*, 16:601-627.
- Mansfield, R.A., 1990, Genetic algorithms, University of Wales College of Cardiff.
- Moghaddam, R.T., Vahed, A.R.R., Mirzaei, A.H., 2008. Solving a multi-objective no-wait flow shop scheduling problem with an immune algorithm, *Int. J. Adv. Manuf. Technol.* 36:969–981.
- Pan, Q.K., Wang, L., Qian, B., 2008. A novel multi objective particle swarm optimization algorithm for no wait flow shop scheduling problems, *IMEchE Vol. 222 Part B: J. Engineering Manufacture*, 519-538.
- Pan, Q.K., Wang, L., Qian, B., 2009. A novel differential evolution algorithm for bi criteria no wait flow shop scheduling problems, *Computers & Industrial Engineering* 36:2498-2511.
- Pan, Q.K., Wang, L., Taşgetiren, F., Zhao, B.H., 2008. A hybrid discrete particle swarm optimization algorithm for no wait flow shop scheduling problem with makespan criterion, *Int. J. Adv. Manuf. Technol.*, 38:337-347.
- Pasupaty, T., Rajendran C., Suresh, R.K., 2006. A multi objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs., *Int. J. Adv. Manuf. Technol.* 27:804-815.
- Pinedo, M., 2002. *Scheduling: theory, algorithms and systems*, NY
- Polat, A., 2006. Kafes sistemlerinin genetik algoritma ile çok amaçlı optimizasyonu, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Fırat Üniversitesi.
- Qian, B., Wang, L., Hu, R., Huang, D. X., Wang, X., 2009. A DE-based approach to no-wait flow-shop scheduling., *Computers&Industrial Engineering* (2009), 57, 787-805.
- Qian, B., Wang, L., Huang, D.X., 2009. Multi-objective no-wait flow-shop scheduling with a memetic algorithm based on differential evolution., *Soft Comput.*,13:847-869.
- Reddi, S. S., Ramamoorthy C. V., 1972. On the flow-shop sequencing problem with no wait in process, *Operational Research Quarterly* (1970-1977), Vol. 23, No. 3 (Sep., 1972), 323-331.
- Rodriguez, J.A.V., Petroviç, S., 2009. A new dispatching rule based genetic algorithm for multi objective job shop problem, *J. Heuristics*.
- Ruiz, R., Allahverdi, A., 2007. No-wait flowshop with separate setup times to minimize maximum lateness, *Int. J. Adv. Manuf. Technol.* 35:551–565.
- Sidney J. B., Potts., C.,N., Sriskandarajah, C., 2000. A heuristic for scheduling two machine no wait flow shop with anticipatory setups, *Operation Research Letters*, 26:165-173.
- Spieksma, F.C.R., Woeginger, G.J., 2005. The no-wait flow-shop paradox, *Operations Research Letters* 33:603–608.
- Strusevich, V.A., 1995. Two machine flow shop scheduling problem with no wait in process: Controllable machine speeds, *Discrete Applied Mathematics* 59:75-86.
- Tseng, L. Y., Lin, Y.T., 2009. A hybrid genetic local search algorithm for permutation flowshop scheduling problem, *European Journal Of Operation Research*, 198:84-92.
- Vahed, A.R.R., Javadi, B., Rabbani, M., Moghaddam, R.T., 2008. A multi objective

- scatter search for a bi-criteria no wait flow shop scheduling problem, *Engineering Optimization*, April:331-346.
- Wang, T.Y., Yang, Y.H., Lin, H.J., 2006. Comparison of scheduling efficiency in two/tree machine no wait flow shop problem using simulated annealing and genetic algorithm, *Asia-Pasific Journal Of Operation Research*, 23:41-59.
- Yun, Y., Yoon, M., Nakayama, H., 2005. Genetic algorithm for multi objective optimization using GDEA, *International Conferance on Neural Computation*, 409-416.
- Zandieh, M., Karimi., N., 2010. An adaptive multi population genetic algorithm to solve the multi objective group scheduling problem in hybrid flexible flowshop with sequence dependent setup times, *J. Intell Manuf.*, 45:263-285.
- Zheng, J., Xiao, G., Song., W., Li., X., Ling., C.X., 2007. A multi objective genetic algorithm based on density ,*International Symposium on Intelligence Computation and Aplications*, 4683 12-25.
- <http://estorototamiroda.com/makaleler/1.htm>
- http://journeytoforever.org/biofuel_library/ethanol_motherearth/z/fig1-1.jpg
- http://tr.wikipedia.org/wiki/Ana_Sayfa
- <http://www.ulusoyelektrik.com.tr/>
- www.exxonmobile.com
- www.solvekimya.com