

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

TOPLAM GEÇ BİTİRME ZAMANININ EN KÜÇÜKLENMESİ
PERFORMANS ÖLÇÜTLÜ PERMÜTASYON AKIŞ TİPİ ÇİZELGELEME
PROBLEMLERİNİN ÇÖZÜMÜNDE GENETİK ALGORİTMA YAKLAŞIMI

Talip KELLEGÖZ

TEMMUZ 2006

Fen Bilimleri Enstitüsünce Yüksek Lisans Tezi olarak uygun bulunmuştur.

...../...../2006

Prof. Dr. M. Yakup ARICA

Müdür

Bu tezin Yüksek Lisans tezi olarak Endüstri Mühendisliği Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Bilal TOKLU

Anabilim Dalı Başkanı

Bu tezi okuduğumuzu ve Yüksek Lisans tezi olarak bütün gerekliliklerini yerine getirdiğini onaylarız.

Prof. Dr. Bilal TOKLU

Danışman

Jüri Üyeleri

Prof. Dr. Bilal TOKLU

Doç. Dr. Burak BİRGÖREN

Yrd. Doç. Dr. Mustafa YÜZÜKIRMIZI

ÖZET

TOPLAM GEÇ BITİRME ZAMANININ EN KÜÇÜKLENMESİ PERFORMANS ÖLÇÜTLÜ PERMÜTASYON AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜMÜNDE GENETİK ALGORİTMA YAKLAŞIMI

KELLEĞÖZ, Talip

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi

Danışman : Prof. Dr. Bilal TOKLU

Temmuz 2006, 102 sayfa

Bu tez çalışması kapsamında toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri detaylı olarak incelenmiş ve bu problemlerin çözümü için bir genetik algoritma geliştirilmiştir. Ayrıca, probleme özgü bilgiler kullanılarak etkin bir çözüm iyileştirme algoritması ve uygun bir çaprazlama operatörü geliştirilip bu genetik algoritmaya entegre edilmiştir.

Karşılaştırma problemleri kullanılarak, geliştirilen genetik algoritma literatürde en iyi performansa sahip olduğu iddia edilen sezgiselle karşılaştırılmış, yapılan simülasyon çalışması sonucunda geliştirilen algoritmanın daha iyi performans sergilediği gösterilmiştir.

Anahtar Kelimeler : Çizelgeleme, Permütasyon Akış Tipi Çizelgeleme,
Genetik Algoritmalar, Toplam Geç Bitirme Zamanı

ABSTRACT

A GENETIC ALGORITHM APPROACH FOR MINIMIZING TOTAL TARDINESS IN PERMUTATION FLOWSHOP SCHEDULING PROBLEMS

KELLEĞÖZ, Talip

Kırıkkale University

Graduate School Of Natural and Applied Sciences

Department of Industrial Engineering, M. Sc. Thesis

Supervisor : Prof. Dr. Bilal TOKLU

July 2006, 102 pages

In this research, a detailed study of the permutation flowshop scheduling problem with the objective of minimizing total tardiness was presented and a genetic algorithm solution procedure was developed for such problems. Also, using problem specific knowledge, an efficient solution improvement scheme and a appropriate crossover operator were developed and integrated into the genetic algorithm.

Using benchmarking problems, the algorithm was compared with a heuristic algorithm which was claimed to have the best performance in the literature. The performance of the developed algorithm has shown to be superior using a simulation study.

Key Words : Scheduling, Permutation Flowshop Scheduling, Genetic Algorithms, Total Tardiness

TEŐEKKÜR

Tezimin hazırlanması esnasında her türlü yardımını esirgemeyen ve biz genç arařtırmacılara büyük destek olan, danıřman hocam, sayın Prof. Dr. Bilal TOKLU 'ya, her konuda olduđu gibi tez alıřmalarım da desteđini esirgemeyen arkadařım Yük. Müh. İsmail KARAOĐLAN 'a ve eđitimim için her türlü fedakarlıđı fazlasıyla gösteren eřim Neslihan KELLEGÖZ ve ođlum Sefa KELLEGÖZ 'e sonsuz teőekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	I
ABSTRACT	III
TEŞEKKÜR.....	V
İÇİNDEKİLER	VI
ÇİZELGELER DİZİNİ	VIII
ŞEKİLLER DİZİNİ	IX
SİMGELER DİZİNİ	X
1. GİRİŞ.....	1
1.1. Çizelgeleme Problemleri	1
1.2. Çizelgeleme Modelleri	4
1.3. Literatür Araştırması.....	8
1.4. Çalışmanın Amacı.....	27
2. MATERYAL VE YÖNTEM	29
2.1. Permütasyon Akış Tipi Çizelgeleme Problemleri.....	29
2.1.1. $F / \text{permu} / \Sigma T_j$ Problemlerinin Formülasyonu	30
2.1.2. $F / \text{permu} / \Sigma T_j$ Problemlerinin Sınıfı	32
2.1.3. $F / \text{permu} / \Sigma T_j$ Problemlerinin Çözüm Yöntemleri	34
2.2. Genetik Algoritmalar.....	39
3. ARAŞTIRMA BULGULARI	48
3.1. Geliştirilen Genetik Algoritma	48
3.1.1. Parametre Kodlama.....	51

3.1.2. Başlangıç Popülasyonu	52
3.1.3. Seçim İşlemi	52
3.1.4. Çaprazlama Operatörü	56
3.1.5. Mutasyon Operatörü	59
3.1.6. Gecikme Tabanlı İyileştirme Algoritması	60
3.1.7. Verilen İki Çizelgenin Benzerlik Oranının Hesaplanması.....	63
3.2. Geliştirilen Genetik Algoritmada Kullanılan Parametre Değerleri	64
3.3. Geliştirilen Genetik Algoritmanın Literatürde Aynı Problem Üzerinde En İyi Performansa Sahip Algoritma İle Karşılaştırılması	65
4. TARTIŞMA VE SONUÇ.....	72
KAYNAKLAR	74
Ek 1: Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri	85
Ek 2: Karşılaştırma Problemleri İçin Üretilen Teslim Tarihlerinde Kullanılan Rastsal Sayıların Başlangıç Çekirdek Değerleri.....	96
Ek 3: Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler	97

ÇİZELGELER DİZİNİ

ÇİZELGE

2.1.Rastsal Olarak Üretilmiş Beş Adet Kromozomdan Oluşan Başlangıç Popülasyonu	46
2.2. Tipik Bir İterasyon Sonucunda Elde Edilen Yeni Popülasyon	47
3.1. Önerilen Genetik Algoritma İçin Tespit Edilen Uygun Parametre Değerleri	65
3.2. 20 İşli Problemler Üzerinde Yapılan Deneme Sonuçları	68
3.3. 50 İşli Problemler Üzerinde Yapılan Deneme Sonuçları	69
3.4. 100 İşli Problemler Üzerinde Yapılan Deneme Sonuçları	70

ŞEKİLLER DİZİNİ

ŞEKİL

1.1. Üretim Sistemlerinde Bilgi Akış Diyagramı.....	3
2.1. Akış Tipi Çizelgeleme Problemleri	29
2.2. Akış Tipi Çizelgeleme Problemlerinde İşlerin Öncelik Yapısı	30
2.3. Standard Genetik Algoritma	43
2.4. Genetik Algoritma Operatörleri	44
3.1. Popülasyondaki Bireylere Ait Seçim Olasılıklarının Derece Tabanlı Hesaplanması.....	55
3.2. Keyfi Üç İş Değiştirme Mutasyon Operatörü Çalışma Şekli	59
3.3. Araya İş Ekleme Komşu Üretim Mekanizması.....	61
3.4. Yapılan Eşleştirilmiş-t Testinin SPSS 12.0 Paket Programı Çıktısı.....	71

SİMGELER DİZİNİ

n	Çizelgeleme problemindeki iş sayısı
m	Çizelgeleme problemindeki makine sayısı
d_j	Çizelgeleme problemindeki j işinin teslim tarihi

KISALTMALAR

PATP	Permütasyon Akış Tipi Çizelgeleme Problemi
TGPATP	Toplam Geç Bitirme Zamanının En Küçüklenmesi Performans Ölçütlü Permütasyon Akış Tipi Çizelgeleme Problemi
GA	Genetik Algoritmalar
TA	Tabu Arama
TB	Tavlama Benzetimi
NEH	Nawaz, Ensore, Ham Yöntemi

1. GİRİŞ

1.1. Çizelgeleme Problemleri

Çizelgeleme, bir çok üretim ve servis endüstrisinde önemli rol oynayan bir karar verme prosesi olup tedarik ve üretimde, taşıma ve dağıtımda, ve bilgi işleme ve iletişimde kullanılır. Çizelgeleme işlevi, matematiksel teknikler veya sezgisel yöntemler kullanarak, bir firmadaki kıt kaynakları yapılması gereken görevlere atar. Kaynakların iyi atanması, firmanın performans ölçütleri ve amaçlarını en iyilemesini sağlar. Buradaki kıt kaynaklar bir atölyedeki tezgahlar, bir hava alanındaki pistler, bir inşaattaki çalışanlar veya bilgisayardaki işlem üniteleri olabilir. Görevler ise üretimdeki işlemler, havaalanındaki iniş ve kalkışlar, inşaat projesindeki safhalar veya çalıştırılması gereken bilgisayar programları olabilir. Her bir görev bir öncelik seviyesine, mümkün en erken başlama zamanına veya en geç tamamlanma zamanına sahip olabilir. Bunun yanı sıra performans kriterleri de çeşitli formlarda karşımıza çıkabilir. Örneğin tüm görevlerin yapılması için gereken zamanın veya geciken görev sayısının en aza indirilmesi vb. olabilir⁽¹⁾.

İdeal amaç fonksiyonu, çizelgeleme kararlarına bağlı sistemdeki tüm maliyetleri kapsayacak yapıda olmalıdır. Buna karşın, uygulamada bu maliyetlerin tam anlamıyla tanımlanması ve hesaplanması çoğunlukla mümkün değildir. Gerçekte, planlama işlevi tarafından sadece kolayca belirlenebilecek temel işlem maliyetleri göz önünde bulundurulurken çizelgelemeye bağlı izole edilmesi güç olan diğer maliyetlerin sabit olarak alınması eğilimi vardır. Çizelgelemede, tamamlanma süresi, zamanında

tamamlanma ve çıktı miktarı olmak üzere yaygın olarak kullanılan üç farklı grup karar verme kriteri bulunur. Çıktı miktarı grubunda bulunan kriterlerin bütün işleri göz önünde bulundurmasına karşın, ilk iki grupta bulunan kriterler verilen tek bir iş için söz konusu olabileceğinden amaç fonksiyonunun bütün işleri göz önünde bulunduracak şekilde geliştirilmesi gerekir⁽²⁾.

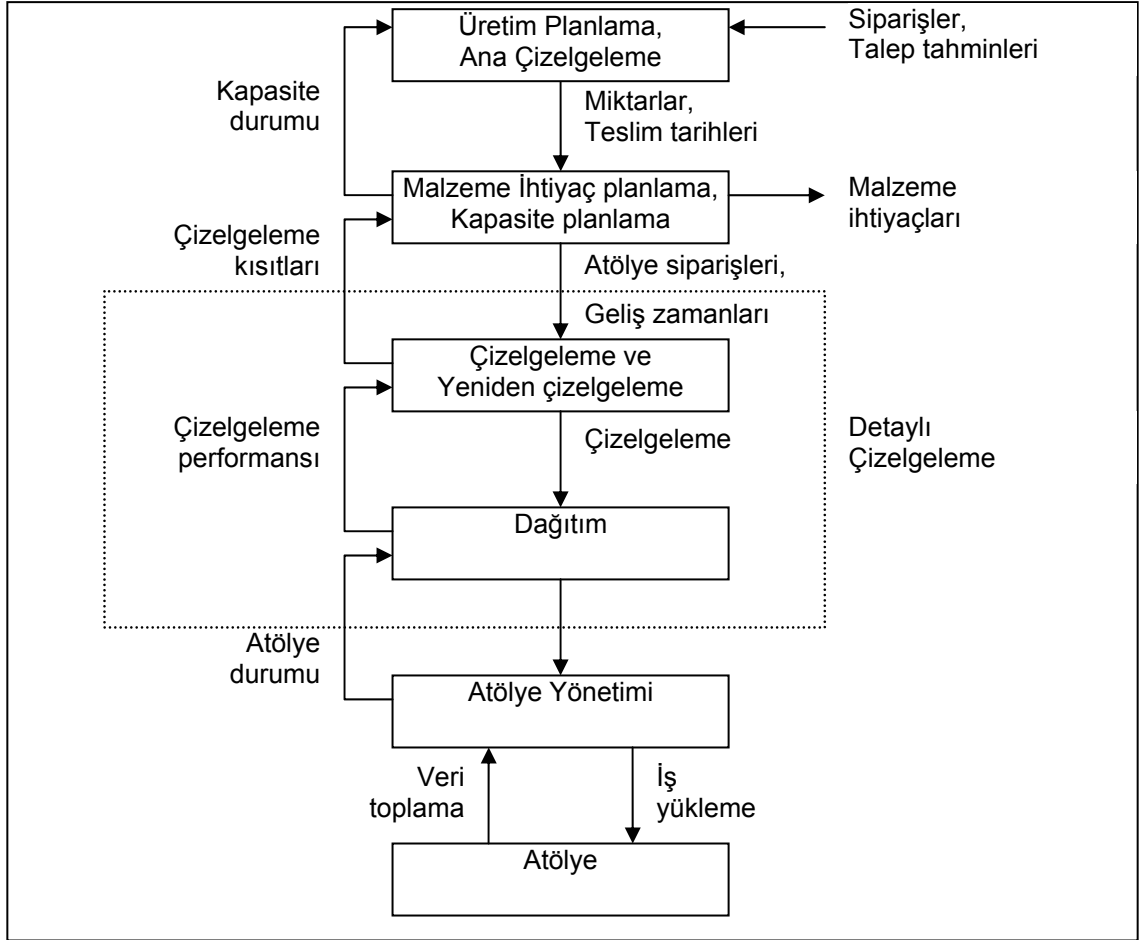
Çizelgeleme problemlerinde iki farklı türden uygunluk kısıtlarıyla karşılaşılmaktadır. Bunlardan ilki makina kapasitelerindeki sınırlamalar, ikincisi ise bazı işlerin işlem sırası üzerindeki teknolojik sınırlamalardır. Çizelgeleme probleminin çözümü, iki gruba ayrılabilen bu kısıtların her hangi bir uygun çözümünü ifade eder. Dolayısıyla çizelgeleme problemini çözme işlemi aşağıdaki sorulara cevap arama işlemini kapsar. Diğer bir deyişle, çizelgeleme problemi atama ve sıralama işlemlerinden oluşur⁽²⁾;

- i. Hangi kaynak hangi göreve atanacak?
- ii. Her bir görev ne zaman yapılacaktır?

Üretim çizelgeleme, bir veya birkaç performans ölçütünü en iyileyecek şekilde üretim kaynaklarının zaman bazında çeşitli görevlere atanmasını kapsar⁽³⁾.

Üretim sistemine gelen siparişler öncelikle, teslim tarihlerini de içeren işlere çevrilmelidir. Genellikle bu işlere ait operasyonların belirli bir sırada atölyedeki bazı makineler üzerinde yapılması gerekir. Buna karşın bu operasyonların, örneğin makinelerin meşgul olması gibi nedenlerle bazı durumlarda bekletilmesi gerekebilir. Bunun yanı sıra makina arızaları, işlem zamanlarının beklenenden daha uzun sürmesi gibi beklenmeyen atölye olaylarının da göz önünde bulundurulması gerekir. Yapılması gereken

görevlere ilişkin detaylı bir çizelgenin hazırlanması işlemlerin kontrolü ve etkinliğinin sürdürülmesinde faydalı olacaktır⁽¹⁾.



Şekil 1.1: Üretim sistemlerinde bilgi akış diyagramı⁽¹⁾

Bir organizasyondaki çizelgeleme işlevi, Şekil 1.1'de görüldüğü gibi, sadece atölyeden değil, aynı zamanda orta ve uzun dönemli planlamadan sorumlu üretim planlama işlevinden de etkilenir. Üretim planlama işlevi, kaynak ihtiyaçları, talep tahminleri ve stok seviyelerini göz önünde bulundurarak uzun dönemli kaynak tahsisinin yanı sıra firmanın ürün karışımını da en iyilemeyi amaçlar. Bu yüksek planlama seviyesindeki

kararlar çizelgeleme işlevini doğrudan etkileyecektir⁽¹⁾.

1.2. Çizelgeleme Modelleri

Çizelgeleme problemleri $\alpha/\beta/\gamma$ şeklinde üç parametrelili bir gösterimle ifade edilir. Bu gösterimdeki α parametresi makina ortamını göstermekte olup, tek bir girdiye sahiptir. β parametresi ise işleme özellikleri ve kısıtlarıyla ilgili detaylı bilgiler sağlar ve problemin özelliğine bağlı olarak hiçbir girdisi olmayacağı gibi bir veya birden fazla girdiye sahip olabilir. γ parametresi ise genellikle tek girdiye sahip olup problemdeki en küçüklenecek performans ölçütünü ifade eder⁽⁴⁾.

α parametresinin alabileceği bazı değerler aşağıdaki gibidir.

- Tek makina(1),
- Benzer özellikli paralel makinalar (P_m): Benzer özelliğe sahip m adet paralel makina söz konusudur. j işine ait tek bir operasyon bulunur ve bu operasyon m adet makinadan her hangi birisinde yapılabilir.
- Farklı hızlara sahip paralel makinalar (Q_m): Farklı hızlara sahip m adet paralel makina söz konusu olup i makinasının hızı v_i notasyonuyla ifade edilir. P_j işlem zamanına sahip j işi i makinasında $P_{ij} = P_j / v_i$ süresini harcar.
- İlişkisiz paralel makinalar (R_m): Her bir iş için farklı hızlara sahip m adet paralel makina söz konusudur. i makinası j işini v_{ij} hızıyla yapabilmekte olup j işinin i makinasında harcadığı P_{ij} zamanı

$P_{ij} = P_j / v_{ij}$ 'ye eşittir.

- Akış tipi (Fm): Seri sıralanmış m adet söz konusudur. Her bir iş aynı rotayı izleyecek şekilde m adet makinanın her birinde işlem görür.
- Esnek akış tipi (FFs): Toplam s adet seri aşama bulunmakta olup her bir aşamada benzer özellikli m adet paralel makina vardır. Her bir iş aynı rotayı izleyecek şekilde s adet aşamanın her birinde bulunan m adet makinanın sadece birinde işlem görür.
- Açık tip (Om): Modelde m adet makina söz konusu olup her bir iş her bir makinada işlem görür. Bazı işlerin bazı makinalardaki işlem süreleri sıfır olabileceği gibi farklı işler farklı rotalara da sahip olabilir.
- Atölye tipi (Jm): Modelde m adet makina söz konusu olup her bir işin her bir makinada işlem görme zorunluluğu yoktur. Yani her bir iş kendine ait bir rotayı takip eder ve her hangi bir makinada yapılması gereken birden fazla operasyona sahip olabilir.

β parametresinin alabileceği bazı değerler aşağıdaki gibidir.

- Geliş zamanı (r_j): İş j 'nin işlenmesine r_j geliş zamanından önce başlanamaz.
- Sıra bağımlı hazırlık zamanları (S_{jk}): Eğer çizelgede j işi k işinden önce geliyorsa S_{jk} , k işine başlanabilmesi için gereken hazırlık zamanını ifade eder. Eğer j ve k işleri arasındaki hazırlık zamanı makinaya da bağımlı ise hazırlık zamanı notasyonuna i indisi eklenerek S_{ijk} şeklinde gösterilir.
- Bölünebilme ($prmp$): İşin tamamlanana kadar makinada kalması

zorunlu değildir. Her hangi bir zamanda her hangi bir işin işlenmesi durdurularak makinaya farklı bir iş yerleştirilebilir. İşlemi yarıda kesilen iş ilgili makinaya tekrar konduğunda sadece kalan süre kadar işlem görür.

- Öncelik kısıtları (*prec*): Bazı işlerin işlenmesine başlanmadan önce diğer bazı işlerin tamamlanması gerektiğiyle ilgili kısıtlamaları ifade eder.
- Arızalanma (*brkdown*): Makinaların tamamı veya bir kısmı arızalanmalar nedeniyle sürekli olarak işlem yapmaya uygun değildir.
- Permütasyon (*prmu*): Akış tipi makina ortamında karşılaşılan bu parametre, makinalar arasındaki kuyruk disiplininin FIFO (ilk gelen ilk işlem görür) olduğunu ifade eder.
- Bloklanma (*block*): Akış tipi makina ortamında karşılaşılan bu parametre, bir birini takip eden iki makina arasındaki kuyruğun sınırlı bir kapasiteye sahip ve dolu olduğu zaman önceki makinanın işlemini bitirdiği işi sonraki makinaya gönderemeyeceğini ifade eder. Bu durum bloklanma olarak bilinir.
- Beklemesiz (*nwt*): Akış tipi makina ortamında karşılaşılan bu parametre, işlerin birbirini takip eden iki makina arasında beklemeyeceğini ifade eder. Bu kısıtlamanın olduğu modelde de kuyruk disiplini FIFO 'dur.
- Yeniden dolaşım (*recrc*): Atölye tipi makina ortamında karşılaşılan bu parametre, her hangi bir işin her hangi bir makinayı birden fazla ziyaret edebileceğini ifade eder.

Çizelgeleme probleminde en küçüklenecek performans ölçütü her zaman işlerin tamamlanma zamanlarına bağlı bir fonksiyonu ifade eder. j işinin i makinasındaki tamamlanma zamanı C_{ij} ve sistemde geçirdiği zaman C_j notasyonu ile gösterilmekte olup, performans ölçütü aynı zamanda teslim zamanı d_j 'nin de bir fonksiyonu olabilir. j işinin gecikmesi $L_j = C_j - d_j$, j işinin geç bitmesi $T_j = \max\{L_j, 0\}$ ve j işinin gecikme durumu $U_j = \begin{cases} 1 & C_j > d_j \\ 0 & \text{diğer} \end{cases}$ fonksiyonlarıyla hesaplanır. Performans ölçütünü ifade eden γ parametresinin bu fonksiyonlara bağlı alabileceği bazı parametreler aşağıdaki gibidir.

- Maksimum tamamlanma zamanı (C_{\max}): Formülasyonu $C_{\max} = \max(C_1, \dots, C_n)$ olup son işin sistemi terk etme zamanını ifade eder. En küçüklenmesi genellikle yüksek makina verimliliğini sağlar.
- Maksimum gecikme (L_{\max}): Formülasyonu $L_{\max} = \max(L_1, \dots, L_n)$ olup teslim zamanından sapmaların en büyüğünü ifade eder.
- Akış zamanı ($\sum C_j$).
- Ağırlıklı akış zamanı ($\sum w_j C_j$): Stok taşıma maliyeti gibi çizelgenin neden olduğu maliyetlerin bir göstergesidir.
- Toplam geç bitirme ($\sum T_j$).
- Ağırlıklı geç bitirme ($\sum w_j T_j$).
- Toplam geciken iş ($\sum U_j$).

- Ağırlıklı geciken iş ($\sum w_j U_j$).

Bu çalışma kapsamında $F / prmu / \sum T_j$ gösterimiyle ifade edilen toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri (TGPATP) göz önünde bulundurulacaktır.

1.3. Literatür Araştırması

Bu bölüm kapsamında toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü tek makinalı çizelgeleme problemleri ($1 // \sum T_j$), toplam ağırlıklı geç bitirme zamanının en küçüklenmesi performans ölçütlü tek makinalı çizelgeleme problemleri ($1 // \sum w_j T_j$) ve toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü akış tipi çizelgeleme problemleri ($F // \sum T_j$) için mevcut literatür çalışmaları özetlenmiştir.

Elmaghraby⁽⁵⁾, $1 // \sum T_j$ problemi için amaç fonksiyonunun en kısa yola çevrildiği bir network modeli önermiş, ihtiyaç duyulan işlem zamanının azaltılması için bu modele dal-sınır algoritmasının nasıl uygulanabileceğini göstermiştir. Ayrıca, teslim tarihi tüm işlerin işlem zamanları toplamından büyük olan işin çizelgenin sonuna yerleştirilebileceğini belirtmiştir.

Emmons⁽⁶⁾, $1 // \sum T_j$ problemi için optimal çizelgedeki işlerin önceliklerini belirleyen bazı teoremler geliştirmiştir. Bu teoremlerden biri, $P_i \leq P_j$ ve $d_i \leq d_j$ ise i -işinin j -işinden önce geleceği en az bir optimal çizelge olduğunu, diğer bir teorem ise $P_i \leq P_j$ ve $d_i \geq d_j$ ise i -işinin j -

işinden sonra geleceği en az bir optimal çizelge olduğunu ifade etmektedir. Bu teoremler “Emmons'un Üstünlük Durumları” olarak bilinmektedir.

Srinivasan⁽⁷⁾, $1//\sum T_j$ problemleri için Emmons⁽⁶⁾'un üstünlük durumlarını kullanan dinamik programlama tabanlı, üç safhalı, hibrit bir algoritma önermiştir. İlk iki safhada Emmons⁽⁶⁾'un üstünlük durumları kullanılarak mümkün olduğunca çok sayıda iş çizelgenin sonuna veya başlangıcına taşınmakta, üçüncü safhada ise modifiye edilmiş bir dinamik programlama algoritması kullanılarak problem çözülmektedir.

Wilkerson v.d.⁽⁸⁾, $1//\sum T_j$ problemi için, C mevcut çizelgedeki en büyük tamamlanma zamanı olmak üzere $C + \max\{P_i, P_j\} > \max\{d_i, d_j\}$ eşitsizliğinin sağlanmaması durumunda i ve j komşu işlerinden teslim zamanı küçük olan işe öncelik verilmesi temeline kullanan bir sezgisel önermiştir. Her adımda, EDD sırasındaki çizelgenmemiş işlerden ilk sırada bulunan alınıp bu kural göre kısmi çizelgeye yerleştirilmektedir. Bütün işler çizelgeye yerleştirildikten sonra işlerin yer değiştirmesi yoluyla elde edilen çizelge iyileştirilmeye çalışılmaktadır.

Shwimer⁽⁹⁾, tek makinalı problemlerde geç bitirme zamanlarının bir fonksiyonunu en küçükleme amacıyla bir dal-sınır algoritması geliştirmiştir. Bu algorithmada başlangıç dallandırma miktarının azaltılması için sezgisel yöntemlerin kullanılmasını önermiştir.

Rinnooy Kan v.d.⁽¹⁰⁾, $1//\sum T_j$ problemi için Emmons⁽⁶⁾'un üstünlük durumlarına bazı eklentiler getirmiş, elde edilen durumları geliştirdiği dal-sınır algoritmasında kullanmıştır. Önerilen algorithmada başlangıç aşamalarında

güçlü alt sınır değerlerine ihtiyaç duyulduğunu, sonraki aşamalarında ise daha basit alt sınır değerlerinin daha etkili olduğunu belirtmiştir.

Fisher⁽¹¹⁾'in $1//\sum T_j$ problemi için önerdiği dal-sınır algoritması dual problemin kullanılmasıyla üretilen bir uygun çözümden işleme başlamakta olup alt sınır değerlerinin tespit edilmesinde üretilen lagrangian problemi kullanılmaktadır.

Lawler⁽¹²⁾, $1//\sum T_j$ problemi bir ayrıştırma prensibi geliştirmiş ve bu prensibi kullanan bir dal-sınır algoritması önermiştir. Geliştirdiği ayrıştırma prensibi, teslim zamanları eşit olan işler kendi içerisinde SPT sırasında olmak üzere tüm işler EDD'ye göre sıralandıktan sonra işlem zamanı en büyük olan j -işi $j \leq k$ olmak üzere k -inci pozisyonda ise, bu pozisyon baz alınarak problemin ayrıştırılabileceğini ifade etmektedir. Elde edilen alt problemler SPT veya EDD ile çözülene kadar veya ayrıştırma mümkün olmayana kadar, belirtilen prensip kullanılarak ayrıştırma işlemine devam edilmektedir.

Baker v.d.⁽¹³⁾, $1//\sum T_j$ problemi için bir dinamik programlama yaklaşımı geliştirmiştir. Zincir Algoritması adını verdiği bu algoritmada Emmons⁽⁶⁾'un üstünlük durumlarını kullanmış olup CPU zamanı açısından Rinnooy Kan v.d.⁽¹⁰⁾'in metodundan daha iyi performans sergilediğini belirtmiştir.

Picard v.d.⁽¹⁴⁾, $1//\sum T_j$ problemi için alt sınır değerlerinin hesaplanmasında gezgin satıcı problemlerinin kullanıldığı bir dal-sınır algoritması önermiştir.

Schrage v.d.⁽¹⁵⁾, $1//\sum T_j$ problemi için öncelik kısıtlarının

oluşturulmasında Emmons⁽⁶⁾'un üstünlük durumlarının kullanıldığı, etiketleme tabanına dayanan bir dinamik programlama algoritması geliştirmiştir.

Baker v.d.⁽¹⁶⁾, $1//\sum T_j$ problemi için EDD tabanlı, modifiye edilmiş teslim tarihi adı verilen bir kural geliştirmiştir. Bu kurala göre işler C , en son kısmi çizelgeye yerleştirilmiş işin tamamlanma zamanı olmak üzere $MDD = \max\{C + P_j, d_j\}$ formülasyonu ile hesaplanan modifiye edilmiş teslim tarihleri içerisinde en küçük değere sahip olanı seçilerek kısmi çizelgeye dinamik olarak yerleştirilmektedir.

Potts v.d.⁽¹⁷⁾, $1//\sum T_j$ problemi için Lawler⁽¹²⁾'in ayrıştırma prensibini kullanan bir dal-sınır algoritması geliştirmiştir. Çizelgedeki her bir aday pozisyon en büyük işlem zamanına sahip işin ayrıştırma prensibine bağlı olarak çizelgelenmesi amacıyla dal-sınır algoritması kullanılarak araştırılmakta, elde edilen alt problemler Schrage v.d.⁽¹⁵⁾'nin dinamik programlama algoritmasıyla çözülmektedir. Önerilen algoritmanın 100'e kadar işe sahip problemleri çözebildiği belirtilmiştir.

Sen v.d.⁽¹⁸⁾, $1//\sum T_j$ problemi için MINIT adı verilen ve iş değişkenleri arasındaki mevcut ilişkileri kullanan bir dal-sınır algoritması geliştirmiştir. Sınırlı hafıza söz konusu olduğunda bu algoritmanın avantajlı olacağını belirtmiştir.

Potts v.d.⁽¹⁹⁾, $1//\sum w_j T_j$ problemi için, toplam ağırlıklı tamamlanma zamanı performans ölçütlü alt problemlerin kullanıldığı ve Lagrangia gevşetmesi yöntemiyle alt sınır değerlerinin hesaplandığı bir dal-sınır algoritması önermiştir. Ayrıca bu algorithmada, dal-sınır ağacındaki işler

arasında bulunan üstünlük ilişkilerinin kontrol edilmesi amacıyla dinamik programlama yaklaşımı da kullanılmıştır.

Ow⁽²⁰⁾, $F // \sum w_j T_j$ problemi için komşu iki iş arasındaki yerel optimumluk durumlarını geliştirmiş, bu durumları orantılı $F // \sum w_j T_j$ problemleri için “boş zaman kuralı” adını verdiği sezgiseli geliştirmek için kullanmıştır. Orantılı $F // \sum w_j T_j$ problemlerinde her bir makinaya ait bir orantı sabiti olup her hangi bir işin bu makinadaki operasyon zamanı, bu sabitle referans makinadaki ilgili işin operasyon zamanının çarpımına eşittir. Boş zaman kuralı sezgiseli her bir adımda, çizelgelenmemiş işler arasında hesaplanan en büyük öncelik değerine sahip işi seçerek kısmi çizelgenin sonuna ekler ve bu işleme tüm işler çizelgeye dahil edilinceye değin devam eder.

Potts v.d.⁽²¹⁾, $1 // \sum T_j$ problemi için Lawler⁽¹²⁾’ın ayrıştırma prensibine bazı modifikasyonlar getirmiştir. SPT ve EDD ile çözülemeyen alt problemlerin Scrage v.d.⁽¹⁵⁾’nin dinamik programlama algoritmasıyla çözülmesini önermiştir.

Rachamadugu⁽²²⁾, $1 // \sum w_j T_j$ problemi için optimal çizelgede komşu iş çiftlerini karakterize eden bir durum geliştirmiş, bu durumun sayımlama metotlarında etkin bir budama aracı olarak kullanılabileceğini belirtmiştir. Ayrıca problemin tamamen çözülmeyen optimal çizelgedeki ilk işin tespit edilmesiyle ilgili bazı durumları da sunmuş ve probleme ait en az bir optimal çizelgenin Baker v.d.⁽¹⁶⁾’nin MDD kuralını sağlayacağını belirtmiştir.

Fry v.d.⁽²³⁾, $1 // \sum T_j$ problemi için SPT, EDD ve en küçük boş zaman

kurallarıyla oluşturulan başlangıç çizelgelerine üç farklı iyileştirme prosedürü uygulanarak elde edilen dokuz farklı çizelge içerisinde en iyisinin seçildiği bir sezgisel önermiştir. İlk iyileştirme prosedürü, başlangıç çizelgesinin sol kısmından başlayarak komşu iş çiftlerinin yer değiştirmesi yoluyla çizelgeyi iyileştirmeye çalışmaktadır. İkinci prosedür ise aynı iyileştirme işlemini çizelgenin sağ tarafından başlayarak uygulamaktadır. Üçüncü ve son prosedürde ise tüm iş çiftleri yer değiştirilerek çizelge iyileştirilmeye çalışılmaktadır.

Sen v.d.⁽²⁴⁾, $F2 // \sum T_j$ problemi için komşu işlerden hangisinin çizelgede daha önce yer alması gerektiğiyle ilgili yerel optimumluk durumu geliştirmiş, bu durumu kullanan bir dal-sınır algoritması önermiştir. Ayrıca bu durumdan yola çıkarak üç farklı sezgisel de önermiştir. Bu sezgisellerle üretilen çözümlerin komşu iş çiftlerinin yer değiştirmesiyle daha iyi çözümlere ulaşılabileceği de belirtilmiştir.

Du v.d.⁽²⁵⁾ $1 // \sum T_j$ problemlerinin NP-zor sınıfında yer aldığını göstermiştir.

Potts v.d.⁽²⁶⁾, $1 // \sum T_j$ problemi için iş çiftlerinin yer değiştirmesi mekanizmasını kullanan, tavlama benzetimi tabanlı bir sezgisel geliştirmiştir. Ayrıca, bu çalışmada iniş algoritmasının entegre edildiği bir tavlama benzetimi sezgiseli de önermiştir. Bu sezgiselde, amaç fonksiyonunda iyileşme olması durumunda iş çiftlerinin yer değiştirmesi mekanizmasını kullanan iniş algoritması çağırılarak mevcut çözümün yerel optimuma taşınması sağlanmakta, bu işlemten sonra tavlama benzetimi sürecine devam edilmektedir.

Sen v.d.⁽²⁷⁾, $1 // \sum T_j$ problemi için iş değişkenleri arasında bulunan bazı ilişkileri geliştirerek, bu ilişkileri kullanan bir dal-sınır algoritması önermiştir. Geliştirilen algoritma Schrage v.d.⁽¹⁵⁾'nin dinamik programlama algoritmasıyla karşılaştırılmış, dinamik programlamanın geliştirilen algoritmadan daha iyi performans sergilediği görülmüştür.

Holsenback v.d.⁽²⁸⁾, $1 // \sum T_j$ problemi için “Yer Değiştirmenin Net Kazancı” adı verilen bir sezgisel önermiştir. Bu sezgisel EDD çizelgesiyle işleme başlamakta, her bir iş için pozisyonunu değiştirmesi durumunda toplam geç bitirme zamanındaki net kazanç hesaplanmakta, bu kazanç negatifse işin pozisyonu değiştirilmektedir. Kazançlı bir pozisyon değiştirme işlemi bulunamayınca sezgisel sona ermektedir.

Adenso-Diaz⁽²⁹⁾, orantılı $F / \text{permu} / \sum w_j T_j$ problemi için tabu arama tabanlı bir sezgisel geliştirmiştir. Geliştirilen sezgisel, Widmer v.d.⁽³⁰⁾ 'nin $F / \text{permu} / C_{\max}$ problemi için önerdiği tabu arama sezgiselinin modifiye edilmesi yoluyla geliştirilmiştir. Sezgiselde başlangıç çözümü $Ow^{(20)}$ 'ın önerdiği sezgiselle elde edilmekte, komşu üretme mekanizması olarak araya iş ekleme mekanizması ve sabit tabu listesi uzunluğu kullanılmakta olup orta ve uzun dönemli hafıza yapısına yer verilmemiştir.

Kim⁽³¹⁾, $F2 // \sum T_j$ problemi için bir dal-sınır algoritması geliştirmiştir. Algoritma, başlangıç üst sınır değerini EDD sırasının toplam geç bitirme zamanı değeri olarak tespit etmekte olup, teslim zamanı tüm işlerin tamamlanabileceği en büyük zamandan sonra olan işleri çizelgenin sonuna atarak göz ardı etmektedir. Dallandırma çizelgenin sonundan başlanmakta

olup çizelgeye eklenmemiş işler için Johnson⁽³²⁾ algoritmasıyla maksimum tamamlanma zamanları tespit edilmekte, bu zamanlar ise makinaların çizelgelenmiş işleri yapmaya başlayacağı zamanın alt sınır değeri olarak alınmaktadır. Çizelgedeki işlerin alt sınır değeri ise önerilen bir formülasyonla hesaplanmaktadır.

Kim⁽³³⁾, $F / \text{permu} / \sum T_j$ problemi için bazı sezgiseller önermiştir. Bu sezgiseller, daha önce $F / \text{permu} / C_{\max}$ ve $F / \text{permu} / \sum w_j T_j$ için önerilmiş, her seferinde tek bir işin kısmi çizelgeye eklenmesi yoluyla başlangıç çizelgesini oluşturan ve daha sonra bu çizelgeyi komşu arama metotlarıyla iyileştiren sezgisellerin $F / \text{permu} / \sum T_j$ problemine uyarlanması yoluyla geliştirilmiştir. Yapılan deneysel çalışmalarda Widmer v.d.⁽³⁰⁾'nin tabu arama algoritmasının incelenen problem için modifiye edilmiş halinin en iyi performansı sergilediği görülmüştür.

Panwalkar v.d.⁽³⁴⁾ 'nin $1 // \sum T_j$ problemi için önerdiği sezgisel SPT başlangıç sırasıyla işleme başlamakta, iş sayısına eşit sayıdaki geçişin her birinde bu sıradan bir işi aktif iş olarak tespit edip çizelgeye eklemektedir. Çizelgelenmemiş SPT sırasındaki geciken ilk iş aktif iş olarak seçilmekte, böyle bir iş bulunamazsa teslim tarihi sağ tarafındaki tüm işlerden daha küçük olan iş aktif iş olmaktadır. Ayrıca çalışma kapsamında, önerilen sezgiselin Holsenback v.d.⁽²⁸⁾'nin sezgiselinden daha iyi performans sergilediği belirtilmiştir.

Scwarc v.d.⁽³⁵⁾, ağırlıkların işlem zamanlarıyla orantılı olduğu $1 // \sum w_j T_j$ problemi için iki aşamalı bir ayrıştırma mekanizması önermiştir. Bu

mekanizma problemi ya tamamen çözmekte veya daha küçük alt problemlere ayırmaktadır.

Fadlalla v.d.⁽³⁶⁾'nin $1//\sum T_j$ problemi için önerdiği sezgiselde çizelgelenmemiş işler içerisinde çizelgenin sonunda yer alma eğilimi en yüksek olanı seçilerek çizelgeye yerleştirilmektedir. Bu süre in problemdeki iş sayısına eşit sayıda tekrarlanması yoluyla çizelge oluşturulmaktadır. Önerilen sezgiselin Wilkerson v.d.⁽⁸⁾'nin sezgiselinden daha iyi performansa sahip olduğu ve problem boyutunun artmasıyla sezgiselin performansının yükseldiği belirtilmiştir.

Kondakci v.d.⁽³⁷⁾, $1//\sum T_j$ problemi için bir alt sınır ve birkaç üst sınır hesaplama yöntemi ve bu yöntemleri kullanan dal-sınır algoritması, baskınlık özellikleri ve öncelik ilişkileri geliştirmiştir. Önerilen dal-sınır algoritmasında Schrage v.d.⁽¹⁵⁾'nin etiketleme yöntemi kullanılmaktadır.

Chang v.d.⁽³⁸⁾, $1//\sum T_j$ problemi için Lawler⁽¹²⁾'in ayrıştırma prensiplerine sol ayrıştırma pozisyonunu içeren bazı eklentiler yapmıştır.

Kim⁽⁴⁰⁾, $F / permu / \sum T_j$ problemi için her bir düğümü tam çizelgenin bitiş yönündeki kısmi parçasını ifade edecek şekilde bir dal-sınır algoritması önermiştir. Bir düğümüne ait alt sınır değeri, ilgili düğümdeki kısmi çizelgenin ve çizelgelenmemiş işlerin alt sınır değerlerinin toplanmasıyla elde edilmekte, başlangıç üst sınır değeri olarak ise Kim⁽³³⁾'in önerdiği EDD sırasını iyileştiren tabu arama sezgiselinin bulunduğu toplam geç bitirme zamanı dikkate alınmaktadır. Diğer çizelgelere baskın olan bazı çizelgelerin belirlenmesiyle ilgili çeşitli özellikler geliştirilmiş ve önerilen dal-sınır algoritmasında

kullanılmıştır. Ayrıca algoritmanın başlangıç adımında teslim tarihi en büyük tamamlanma zamanından daha büyük olan işler çizelgenin sonuna atılarak göz ardı edilmektedir.

Raman⁽⁴¹⁾, $F2 // \sum T_j$ problemi için işler arasındaki yerel optimallik durumlarını belirlemiş ve bu durumları $F / \text{permu} / \sum T_j$ problemi için önerdiği sezgisel algoritmayı geliştirmek amacıyla kullanmıştır.

Ben-Daya v.d.⁽⁴²⁾, $1 // \sum T_j$ problemi için tavlama benzetimi tabanlı bir sezgisel geliştirmiş, Fry v.d.⁽²³⁾ ve Holsenback v.d.⁽²⁸⁾'nin sezgiselleriyle karşılaştırmış ve önerilen sezgiselin daha iyi performansa sahip olduğunu belirtmiştir.

Szwarc v.d.⁽⁴³⁾, $1 // \sum T_j$ problemi için yeni bir ayrıştırma kuralı geliştirmiş ve sadece ayrıştırma tabanına sahip bir dal-sınır algoritması önermiştir.

Yu⁽⁴⁴⁾, $1 // \sum T_j$ problemi için kısmi bir alt çizelgeden başlanıp Emmons⁽⁶⁾'un üstünlük kriterleri kullanılarak elde edilecek kısmi çizelgenin genel olarak tutarlı olmayacağını örnekler vererek göstermiş, buna karşın boş bir çizelgeden başlanması durumunda elde edilecek kısmi çizelgenin tutarlı olacağını belirtmiştir.

Russell v.d.⁽⁴⁵⁾, $1 // \sum T_j$ problemi için daha önce önerilen Holsenback v.d.⁽²⁸⁾'nin sezgiselinin Panwalkar v.d.⁽³⁴⁾'nin sezgiselinden, Panwalkar v.d.⁽³⁴⁾'nin iddia ettiklerinin aksine, daha iyi performansa sahip olduğunu göstermek amacıyla yapılan deneysel çalışma sonuçlarını sunmuştur.

Russell v.d.⁽⁴⁶⁾, $1//\sum T_j$ problemi için daha önce önerdikleri Holsenback v.d.⁽²⁸⁾'nin sezgiseli üzerinde çeşitli modifikasyonlar yapmış, sezgiselin yeni halinin hesaplama zamanında artış olmaksızın öncekinden daha iyi performans sergilediğini göstermiştir.

Alidaee v.d.⁽⁴⁷⁾, $1//\sum T_j$ problemi için Panwalkar v.d.⁽³⁴⁾'nin önerdiği sezgiselin Baker v.d.⁽¹⁶⁾'nin önerdiği MDD sezgiselinin bir uygulaması olduğunu göstermiş, Wilkerson v.d.⁽⁸⁾'nin sezgiseliyle MDD sezgiseli arasındaki ilişkileri araştırmıştır.

Tansel v.d.⁽⁴⁸⁾, Emmons⁽⁶⁾'un $1//\sum T_j$ problemi için geliştirdiği iş çiftleri arasındaki ilişkilerle ilgili teoremleri geometrik açıdan incelemiş, üstünlük özelliklerine yeni bakış açıları geliştirerek kolay ve zor problem sınıflarının birbirinden daha net olarak ayırıldığını sağlamıştır.

Koulamas⁽⁵⁰⁾, $1//\sum T_j$ problemi için EDD ve SPT sırasının aynı olması durumunda bu sıranın optimal çizelgeyi vereceğini göstermiş, bu sonucun $P//\sum T_j$ problemleri için de geçerli olduğunu belirtmiştir.

Akturk v.d.⁽⁵¹⁾'nin $1//\sum w_j T_j$ problemi için önerdiği üstünlük kuralı, problem için yerel optimalite durumunu tanımlamaktadır. Akturk v.d.⁽⁵¹⁾, önerilen kuralı bozan çizelgeye kural uygulandığı zaman çizelgenin toplam ağırlıklı geç bitirme zamanının ya azalacağını veya da aynı kalacağını göstermiştir. Bu nedenle bu kuralın optimal bulan algoritalarda alternatif sayısının azaltılması için kullanılabileceği belirtilmiştir.

Koulamas⁽⁵²⁾, $1//\sum T_j$ problemi için daha önce sunulmuş olan

Lawler⁽¹²⁾'in ayrıştırma prensiplerini ve Potts v.d.⁽¹⁷⁾'nin dal-sınır algoritmasını kullanarak $F2//\sum T_j$ problemi için bir sezgisel geliştirmiştir. Sezgisel, öncelikle $F2//\sum T_j$ problemini $1//\sum T_j$ problemine çevirmekte, belirtilen yöntemleri bu probleme uygulayarak $F2//\sum T_j$ problemi için yaklaşık çözüm bulmaktadır.

Parthasarathy v.d.⁽⁵³⁾, $F / permu / \sum T_j$ ve $F / permu / \sum w_j T_j$ problemleri için tavlama benzetimi tabanlı bir algoritma geliştirmiştir. Geliştirilen bir kurucu sezgiselle başlangıç çözümünün elde edildiği tavlama benzetimi sezgiselinde iki farklı komşu üretme mekanizması kullanılmıştır. Bu mekanizmalardan birincisinde, mevcut çizelgede bulunan her bir iş bulunduğu pozisyonun sağ veya solu yönünde rastsal bir pozisyona taşınmakta, ikinci mekanizmada ise her bir iş tek bir yönde pozisyonunu değiştirmekte, hangi yönde pozisyonunu değiştireceği 0.5 olasılıkla belirlenmektedir.

Hirakawa⁽⁵⁴⁾, $1//\sum T_j$ problemi için Lawler⁽¹²⁾'in ayrıştırma teoremine dal-sınır algoritmasını direkt uyguladığı bir optimal çözüm algoritması önermiştir. Algoritma kapsamında etkileşimli amaç fonksiyonu hesaplama yaklaşımı kullanılmış olup, algoritmanın hızlı bir şekilde çözüme ulaştığı belirtilmiştir.

Armentano v.d.⁽⁵⁵⁾, $F / permu / \sum T_j$ problemi için tabu arama tabanlı bir sezgisel geliştirmiştir. Dinamik tabu listesi uzunluğunun kullanıldığı sezgiselde komşular, tabu listesinde bulunmayan her bir işin orijinal pozisyonu haricinde diğer bütün pozisyonlara yerleştirilmesi yoluyla elde

edilmekte, hareket edilecek komşu çözümü oluşturan iş tek başına tabu listesine eklenmektedir. Ayrıca çeşitlendirme stratejisi olarak frekans tabanlı hafıza yapısı kullanılmıştır.

Holsenback v.d.⁽⁵⁶⁾, $1//\sum T_j$ problemi için işlerin $\max\{P_j, d_j\}$ değerlerine göre büyükten küçüğe sıralanmasıyla elde edilen başlangıç çizelgesini kullanan bir kurucu sezgisel önermiştir. Bu sezgiselde, çizelgeye yerleştirilmemiş işler içerisinde, önerilen bir formülasyonla hesaplanan en büyük performans ölçütündeki net azaltma miktarına sahip iş seçilerek kısmi çizelgenin önüne eklenmektedir.

Biskup v.d.⁽⁵⁷⁾, Kondakci v.d.⁽³⁷⁾'nin $1//\sum T_j$ problemi için önerdiği dal-sınır algoritmasını incelemiş, bu algortmada özellikle Schrage v.d.⁽¹⁵⁾'nin etiketleme yönteminin kullanılmasının açık olmadığını ve toplam sayımlama sayısının azaltılmasında hata yapıldığını ileri sürmüştür.

Dimopoulos v.d.⁽⁵⁸⁾, $1//\sum T_j$ problemi için ayrıştırma kuralı temeline dayanan çizelgeleme politikaları oluşturmak amacıyla Genetik Programlama yaklaşımı kullanmıştır.

Szwarc v.d.⁽⁵⁹⁾, $1//\sum T_j$ problemi için Szwarc v.d.⁽⁴³⁾'nin önerdiği dal-sınır algoritmasında daha iyi bir ayrıştırma kuralının kullanılması gibi çeşitli değişiklikler yaparak performansını geliştirmeye çalışmıştır. Ayrıca Chang v.d.⁽³⁸⁾'nin Lawler⁽¹²⁾'in ayrıştırma prensibine yaptığı eklentilerin daha da geliştirilebileceğini göstermiştir.

Tansel v.d.⁽⁶⁰⁾, $1//\sum T_j$ problemi için daha önce diğer çalışmalarda sunulmuş çeşitli teoremlerin entegrasyonu tabanına dayanan bir dal-sınır

algoritması ve bu algoritmadan türetilen bir sezgisel önermiştir. Önerilen dal-sınır algoritmasının temel bileşenlerini basit bir optimalite kontrolü, yeni bir ayrıştırma teorisi, yeni bir alt sınır hesaplama yaklaşımı, alt problemlerin daha önce çözümlenip çözülmediğine ilişkin bir kontrol mekanizması oluşturmakta olup, bu bileşenler sayesinde hem dal-sınır ağıcının boyutu hem de çözüm zamanında azalma olduğu belirtilmiştir.

Naidu v.d.⁽⁶¹⁾, $1//\sum T_j$ problemi için Baker v.d.⁽¹⁶⁾'nin MDD sezgiselinin ayrıştırma yapısına sahip olduğunu göstermiştir. Bu tespit ışığında ayrıştırma tabanına dayanan yeni bir sezgisel önermiş, bu sezgiselin MDD sezgiseliyle benzerliklerini vurgulamıştır. Ayrıca MDD sezgiselinin optimumu garantilediği durumları sunmuştur.

Pan v.d.⁽⁶²⁾, $F2//\sum T_j$ problemi için optimal çizelgedeki işlerin bir birlerine göre öncelik ilişkilerini belirleyen bazı üstünlük kuralları geliştirmiş, Kim⁽³¹⁾ ve Pan v.d.⁽⁴⁹⁾'nin önerdiği üstünlük kurallarını da kullanan bir dal-sınır algoritması önermiştir.

Avcı v.d.⁽⁶³⁾, $1//\sum w_j T_j$ problemi için genetik algoritma tabanlı bir sezgisel geliştirmiştir. Önerilen sezgiselde arama uzayındaki komşu çözüm yapısının geliştirilmesi amacıyla yerel ve global zaman bağımlı üstünlük kriterleri kullanılmış olup deney tasarımıyla uygun parametre değerleri tespit edilmiştir.

Maheswaran v.d.⁽⁶⁴⁾, $1//\sum w_j T_j$ problemi için öncelikle geriye doğru başlangıç çözümünü kuran ve daha sonra ileriye doğru bu çözümü iyileştiren algoritmanın performansını çeşitli senaryolarla üretilen test problemleri

üzerinde incelemiş ve bu algoritmanın kabul edilebilir zamanda iyi çözümler üretebildiğini belirtmiştir.

Naidu⁽⁶⁵⁾, $1//\sum T_j$ problemi için Rachamadugu⁽²²⁾'nin MDD kuralına ilişkin tespitlerinin her zaman geçerli olmadığını iddia etmiş, MDD sezgiselinin bazı durumlarda optimal çizelgeyi sağlamayacağını göstermiştir.

Croce v.d.⁽⁶⁶⁾, $1//\sum T_j$ problemi için literatürdeki kurucu sezgisellerin ve ayrıştırma tabanlı sezgisellerin optimum çözüme yakınsama oranlarını incelemiş, EDD sezgiseli haricinde incelenen sezgisellerin sabit bir yakınsama oranına sahip olmadıklarını ve yakınsama oranlarının da kötü olduğunu belirtmiştir.

Babu v.d.⁽⁶⁷⁾ $1//\sum w_j T_j$ problemi için Lagrangian ayrıştırması yoluyla hesaplanan alt sınır değerlerinin kullanıldığı bir dal-sınır algoritması geliştirmiştir. Lagrangian ayrıştırması için duali alınan 0-1 tam sayılı model, zaman indeksli 0-1 karar değişkenlerini içerecek şekilde kurulmakta olup Emmons⁽⁶⁾'un üstünlük durumları ve EDD sezgiselinden yola çıkılarak geliştirilen bir üstünlük durumu dal-sınır ağacının boyutunun azaltılması amacıyla kullanılmaktadır.

Grosso v.d.⁽⁶⁸⁾, $1//\sum w_j T_j$ problemi için dinamik komşu arama tabanlı bir sezgisel geliştirmiştir. Yer değiştirme mekanizmasının yanı sıra araya sokma mekanizmasına da izin veren bir komşu üretme mekanizmasının kullanıldığı algoritmada, geliştirilen iki farklı eleme kuralı bu mekanizmanın ürettiği komşu çözüm sayısının azaltılması amacıyla kullanılmaktadır.

Kanet v.d.⁽⁶⁹⁾, $1// \sum w_j T_j$ problemi için ağırlıklı modifiye edilmiş teslim tarihi (WMDD) adı verilen bir sezgisel geliştirmiştir. Sezgiselin geliştirilmesinde Baker v.d.⁽¹⁶⁾'nin MDD sezgiseli hareket noktası olarak alınmış olup geliştirilen sezgisel hem statik hem de dinamik ortamda daha önce önerilen diğer sıralama sezgiselleriyle karşılaştırılmış, hem uygulaması hem de anlaşılması basit olması ve de iyi sonuçlar üretmesi nedeniyle $1// \sum w_j T_j$ problemleri için kullanılabileceği sonucuna varılmıştır.

Hasija v.d.⁽⁷⁰⁾, $F / \text{permu} / \sum T_j$ problemi için tavlama benzetimi tabanlı bir sezgisel geliştirmiştir. Önerilen algoritmanın temel özelliği, aynı çalışmada geliştirilen iki farklı komşu üretme mekanizmasını aynı algoritma içerisinde kullanması ve yine bu çalışma kapsamında geliştirilen bir çözüm iyileştirme algoritmasını başlangıç ve bitiş çözüm setlerinin iyileştirilmesi amacıyla kullanmasıdır. Algoritma başlangıçta 10 adet çözümden oluşan bir set oluşturmaktadır. Bu çözümlerin 9 adeti rastsal üretilirken bir adeti Parthasarathy v.d.⁽⁵³⁾ tarafından geliştirilen kurucu sezgisel ile elde edilmekte ve önerilen iyileştirme algoritmasıyla iyileştirilmekte, bu çözüm aynı zamanda ulaşılan en iyi çözüm olarak saklanmaktadır. Önerilen ve JSB olarak adlandırılan ilk komşu üretim mekanizması şu şekilde işlemektedir. Öncelikle mevcut çözümden bulunan her bir iş için mevcut çözüm ve ulaşılan en iyi çözümde bulunan işlere ait pozisyon farkları kullanılarak bir seçilme olasılığı hesaplanmakta, bu olasılıklar yoluyla da pozisyonu değiştirilecek iş tespit edilmektedir. Seçilen işin hangi yönde pozisyonunun değişeceği olasılıklı olarak tespit edilmekte (0.5 olasılıkla sağ yönde; 0.5 olasılıkla sol yönde), tespit edilen yöndeki yeni pozisyonu da rastsal olarak belirlenmektedir. PSS

adıyları önerilen ikinci komşu üretme mekanizması ise birbirine yakın işlerin olasılıklı olarak yer deęiştirilmesi esasına dayanan bir yapıya sahiptir. Tavlama benzetimi kabul sürecinin uygulanacağı komşu çözüm, öncelikle JSB mekanizmasının iş sayısı kadar, daha sonrada PSS mekanizmasının 2 kez çalıştırılmasıyla (mevcut çözüm deęişmeden) elde edilen komşu çözümlerin en iyisinin seçilmesi yoluyla belirlenmektedir. Sistem sıcaklığının bitiş sıcaklığına veya arama durumuna göre deęer alan bir parametrenin amaç deęerine ulaşmasıyla arama sürecine son verilmektedir. Yani sürecin bitmesi için gerçekleştirilmesi gereken amaç fonksiyonu hesaplanma sayısı sabit deęil deęişkendir. Arama süreci sonunda, saklanan 10 adet çözüm önerilen iyileştirme algoritmasıyla iyileştirilmekte ve en küçük performans ölçütü deęerine sahip olanı sezgisel sonucu olarak sunulmaktadır. Önerilen sezgisel Armentano v.d.⁽⁵⁵⁾'nin tabu arama sezgiseli ve Parthasarty v.d.⁽⁵³⁾'nin tavlama benzetimi sezgiseliyle karşılaştırılmıştır. Karşılaştırmada özel bir formülasyonla teslim zamanları belirlenerek Taillard⁽⁷¹⁾'in $F / \text{permu} / C_{\max}$ için önerdiği karşılaştırma problemlerinden 20, 50 ve 100 işe sahip olanları kullanılmıştır. Karşılaştırmalar sonucunda önerilen sezgiselin performansının dięer sezgisellerden açık ara iyi olduęu görülmüştür.

Leksakul v.d.⁽⁷²⁾, $1 // \sum T_j$ probleminin de arasında bulunduęu tek makinalı bazı tek ve iki kriterli çizelgeleme problemlerinde yapay sinir aęları yaklaşımının nasıl kullanılabileceğini göstermiştir. Her bir problem için öncelikle doğrusal olmayan tamsayılı programlama modeli oluşturulmuş, bu modeller kullanılarak kuadratik enerji fonksiyonu, yapay sinir aęları modeli ve diferansiyel eşitlikler sistemi oluşturulmuştur. Yapılan deneysel çalışmalarda

yapay sinir ağlarının çizelgeleme problemlerinde kullanılabileceği sonucuna varılmıştır.

Schaller⁽⁷³⁾, $F2 // \sum T_j$ problemi için Pan v.d.⁽⁶²⁾'nin önerdiği üstünlük kriterlerini iyileştirmiş, bu kriterlerle birlikte yeni geliştirdiği üstünlük kriterini kullanan bir dal-sınır algoritması önermiştir. Yeni kriterlerin dal-sınır algoritmasının etkinliğini arttırdığını göstermiştir.

Tian v.d.⁽⁷⁴⁾, belirli sayıda farklı teslim tarihine sahip işlerden oluşan $1 // \sum T_j$ problemi için bazı optimalite özelliklerini sunmuş ve bu problemlerin polinomial olarak çözülebileceğini göstermiştir. Ayrıca sonuçları $1/r_j / \sum T_j$ problemi için de genişletmiştir.

Cheng v.d.⁽⁷⁵⁾, $1 // \sum w_j T_j$ problemi için bir sezgisel önermiştir. Çalışmada öncelikle teslim tarihleri işlem zamanlarının doğrusal bir fonksiyonu olduğu ve ağırlıkların ise işlem zamanlarıyla orantılı olduğu tek makinalı modelin çözüm zorluğu ve çözüm yöntemi incelenmiş, daha sonra teslim tarihleri işlem zamanlarından aynı miktarda sapmaya sahip olan model için benzer işlem tekrarlanmıştır.

Holthaus v.d.⁽⁷⁶⁾, $1 // \sum w_j T_j$ problemi için karınca kolonileri tabanlı bir sezgisel geliştirmiştir. Algoritmanın başlamasından sonra her bir iterasyonda üretilen çözüm, fenomen miktarı ve koku yoğunluğu güncellenmeden önce, geliştirilen bir yerel arama yöntemiyle iyileştirilmektedir. Yapılan deneysel çalışmalarda önerilen algoritmanın çözüm zamanı açısından ilgi çekici olduğu belirtilmiştir.

Maheswaran v.d.⁽⁷⁷⁾, $1//\sum w_j T_j$ problemi için her bir kromozomda yerel arama algoritmasının kullanıldığı genetik algoritma (memetik algoritma) tabanlı bir sezgisel geliştirmiştir. Sırala çaprazlama ve yer değiştirme mutasyon operatörü kullanan algoritma, mutasyon işleminden sonra her bir kromozoma yerel arama uygulamaktadır.

Maheswaran v.d.⁽⁷⁸⁾, $1//\sum w_j T_j$ problemi için evrimsel algoritma tabanlı bir sezgisel geliştirmiştir. Önerilen sezgisel tek bir çözüm çiftinden arama işlemine başlamakta, sıralı çaprazlama ve yerel olarak iyileştirilen rastsal yer değiştirme mutasyon operatörü kullanmaktadır.

Merkle v.d.⁽⁷⁹⁾, $1//\sum w_j T_j$ problemi için karınca kolonileri tabanlı bir sezgisel geliştirmiştir. Geliştirilen sezgiselde karıncalar işleri çizelgedeki pozisyonlara, alışılmışın tersine, sıralı olarak değil rastsal olarak atamaktadır. Bu yaklaşımın fenomen bilgisini daha iyi ayarladığı ve hem sıralı hem de rastsal atamanın birlikte kullanılmasının permütasyon tipi çizelgeleme problemlerinde daha iyi sonuç vereceği belirtilmiştir.

Chung v.d.⁽⁸⁰⁾, $F / \text{permu} / \sum T_j$ problemi için bir dal-sınır algoritması önermiştir. Algoritmanın başlangıç adımında Kim⁽⁴⁰⁾ algoritmasında olduğu gibi teslim tarihi en büyük tamamlanma zamanından daha büyük olan işler çizelgenin sonuna atılarak göz ardı edilmektedir. Algoritmada alt sınır değerleri makina bazında hesaplanmakta ve düğümlerin budanmasında önerilen bir baskınlık testi kullanılmaktadır.

Szwarc⁽⁸¹⁾, $1//\sum T_j$ problemi için Lawler⁽¹²⁾'in ayrıştırma prensiplerini genişletmiş ve ayrıştırılamayan problemler için bu prensiplerin nasıl

kullanılabileceğini göstermiştir.

Ergun v.d.⁽⁸²⁾, $1//\sum w_j T_j$ problemi için komşu üretme mekanizmalarından olan yer değiştirme mekanizmasının arama karmaşıklığını azaltan bir yöntem önermiş, bu yöntemin arama algoritmalarının arama karmaşıklığını geliştireceğini göstermiştir.

Bozejko v.d.⁽⁸³⁾, $1//\sum w_j T_j$ problemi için tabu arama tabanlı bir sezgisel geliştirmiştir. Sezgiselde komşu üretme mekanizması olarak araya sokma ve yer değiştirme mekanizmaları kullanılmakta olup, komşu sayısının azaltılması amacıyla işlere ait blok özellikleri kullanılmaktadır. Dinamik tabu listesi uzunluğunun kullanıldığı algoritmanın kısa hesaplama zamanlarında oldukça kaliteli çözümler ürettiği belirtilmiştir.

Bilge v.d.⁽⁸⁴⁾, $1//\sum w_j T_j$ problemi için tabu arama tabanlı bir sezgisel geliştirmiştir. Önerilen sezgiselde komşu üretme mekanizması olarak yer değiştirme ve araya sokma mekanizmaları birlikte kullanılmakta, tabu listesi uzunluğu dinamik olarak değiştirilmekte ve probleme özgü bilgilerin kullanıldığı aday liste stratejileri kullanılmaktadır.

1.4. Çalışmanın Amacı

Tüm işlerin, seri sıralanmış makinalar üzerinde aynı sırayı takip ederek işlenmesini ve makinaların kuyruk disiplininin FIFO (ilk gelen ilk işlem görür) olmasını ifade eden permütasyon akış tipi çizelgeleme problemleri üretim planlama alanında karşılaşılan en önemli problemlerden birisini teşkil eder⁽⁸⁶⁾. Akış tipi çizelgeleme problemlerinde öncü çalışma Johnson⁽³²⁾

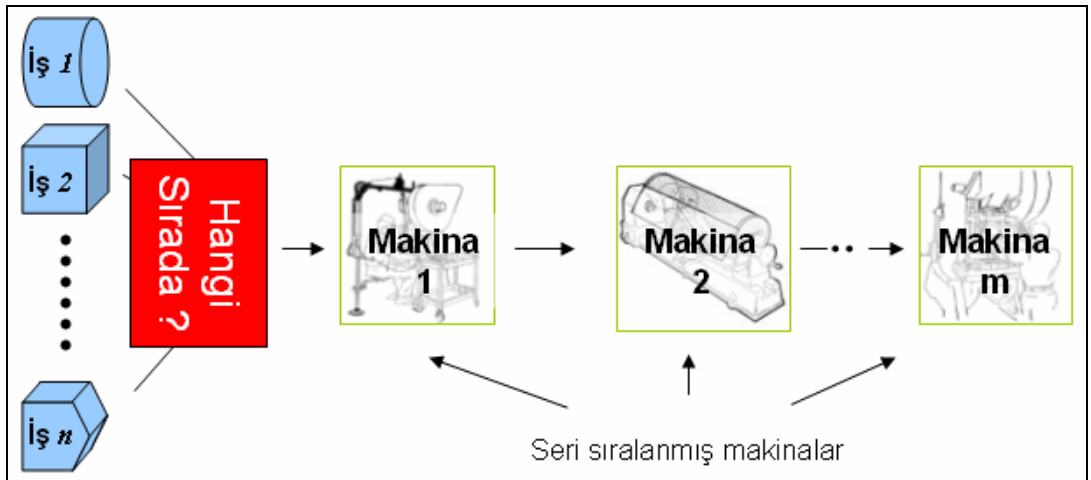
tarafından yapılmıştır. Bu çalışmada Johnson, maksimum tamamlanma zamanı performans ölçütlü iki makinalı akış tipi çizelgeleme problemlerinin optimum çözümünü bulan basit bir algoritma geliştirmiştir. Buna karşın çizelgeleme problemlerinin büyük bir kısmı NP sınıfında yer almakta olup bu problemlerin optimal çözümünün bulunabilmesi için birerleme yöntemlerinin kullanılması gerekir. Problem boyutunun büyümesiyle işlem zamanlarındaki üstel artım nedeniyle birerleme yöntemlerinin kullanılmasının pratik olmadığı da bir gerçektir. Bu nedenlerle zor problemler için yapılan araştırmalar sezgisellerin geliştirilmesi üzerine yoğunlaşmıştır⁽⁸⁶⁾. Yapay zeka tekniklerinden olan Genetik Algoritmalar, Tabu Arama ve Tavlama Benzetimi sezgiselleri bu tip problemlerin optimuma yakın çözümlerinin bulunmasında büyük başarılar göstermiştir⁽⁸⁷⁾.

Bu çalışmanın temel amacı, toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü akış tipi çizelgeleme problemlerinin (TGPATP) çözülmesi için Genetik Algoritma yaklaşımının kullanılmasıdır. Bu yolla literatürde yer alan bu boşluğun doldurulması hedeflenmektedir. Geliştirilen genetik algoritma, parametreleri tespit edildikten sonra, literatürde aynı problem üzerinde en iyi performansı sergilediği iddia edilen ve tavlama benzetimi algoritması temeline dayanan sezgisel yöntemle karşılaştırılacaktır.

2. MATERYAL VE YÖNTEM

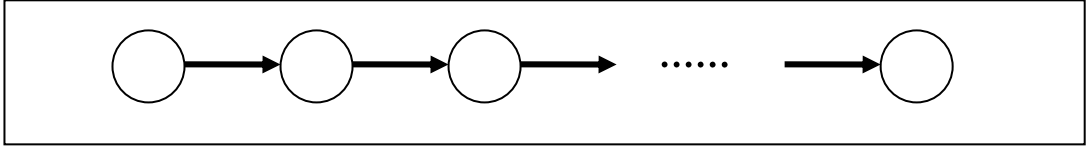
2.1. Permütasyon Akış Tipi Çizelgeleme Problemleri

Akış tipi çizelgeleme problemleri, n adet iş veya görevin, aynı rotayı takip etmek koşuluyla, m adet ardışık makinada verilen bir performans kriterini en küçükleyecek şekilde işlem sırasının bulunması olarak bilinir⁽⁸⁸⁾. Akış tipi çizelgeleme problemlerinde makinalar seri olarak sıralanmıştır ve her hangi bir makinada operasyonu tamamlanan iş bir sonraki makinanın kuyruğuna katılır⁽¹⁾. Permütasyon akış tipi çizelgeleme problemleri ise akış tipi çizelgeleme problemlerinin özel bir sınıfı olup, bu sınıftaki problemlerde bütün makinalardaki iş sırası aynıdır (yani tüm makinalardaki kuyruk disiplini ilk gelen ilk işlem görür prensibini ifade eden FIFO 'dur). Bu problemlerde uygun bir çözümü gösteren çizelge tek bir permütasyonla tanımlanır⁽²⁾. Bu tür çizelgeleme problemleri, işlerin makinadan makinarya malzeme taşıma sistemleriyle taşındığı üretim sistemlerinde ve montaj hatlarında görülür⁽⁸⁹⁾.



Şekil 2.1: Akış Tipi Çizelgeleme Problemleri

Akış tipi çizelgeleme problemlerinde iş üzerinde yapılması gereken işlemler, her biri ayrı makinalar gerektiren ve operasyon adı verilen bağımsız görevlere ayrılmıştır. Bu bağlamda iş, özel bir öncelik yapısına sahip operasyonlar kümesini ifade eder. Şekil 2.2’de gösterildiği gibi, ilk operasyondan sonraki her bir operasyon bir öndüle ve son operasyondan önceki her bir operasyon da bir ardıla sahiptir. Bu nedenle her bir iş, tamamlanabilmesi için operasyonlarının belirli bir sırada yapılmasına ihtiyaç duyar. Bu yapı, Doğrusal Öncelik Yapısı olarak adlandırılır⁽²⁾.



Şekil 2.2:Akış Tipi Çizelgeleme Problemlerinde İşlerin Öncelik Yapısı

2.1.1. $F / \text{permu} / \sum T_j$ Problemlerinin Formülasyonu

Permütasyon akış tipi çizelgeleme problemleri şu şekilde formüle edilebilir. $J = \{1,2,\dots,n\}$ iş setindeki her bir iş $1,2,\dots,m$ olmak üzere m adet makina üzerinde, makinaların indekslenme sırasına göre işlenecektir. $j \in J$ olmak üzere her bir j işi makina sayısına eşit ve $O_{j1}, O_{j2}, \dots, O_{jm}$ notasyonu ile gösterilen m adet operasyondan oluşur. Bu gösterimdeki O_{jk} operasyonu j işinin k makinasında kesintiye uğramaksızın $P_{jk} \geq 0$ süresince işlenmesini ifade eder. Ayrıca $j \in J$ olmak üzere j işinin teslim

zamanı d_j notasyonuyla gösterilir. Bu problemdeki temel amaç her bir makinadaki iş sırasının aynı olduğu ve toplam geç bitirme zamanı değerini en küçükleyen çizelgenin bulunmasıdır. Başka bir ifadeyle, $\pi(i)$, π çizelgesinde i -nci pozisyonda yer alan işi ifade etmek üzere, işlerin oluşturduğu her bir çizelge $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ permütasyonu ile tanımlanabilir. J iş seti üzerinde bu şekilde oluşturulabilen permütasyonların tümünün kümesi Π ve tüm işlerin makinalarda π çizelgesinde belirtilen sırada yapılması durumunda toplam geç bitirme zamanı ise $T(\pi)$ olsun. Problemdeki temel amaç $\pi^* \in \Pi$ olmak üzere aşağıdaki eşitliği sağlayan π^* çizelgesinin bulunmasıdır.

$$T(\pi^*) = \min_{\pi \in \Pi} T(\pi) \quad (2.1)$$

C_{jk} , j işinin k makinasında tamamlanma zamanı olsun. Bilindiği gibi toplam geç bitirme süresi çizelgedeki tüm işlerin gecikmeleri toplamını ifade eder ve aşağıdaki notasyonla hesaplanır:

$$T(\pi) = \sum_{i=1}^n [\max(C_{\pi(i)m} - d_{\pi(i)}, 0)]. \quad (2.2)$$

Çizelgedeki $\pi(i)$ işinin k makinasındaki tamamlanma zamanı aşağıdaki eşitlikle tespit edilebilir.

$$C_{\pi(i)k} = \max\{C_{\pi(i-1)k}, C_{\pi(i)k-1}\} + P_{\pi(i)k} \quad i = 1, 2, \dots, n; k = 1, 2, \dots, m \quad (2.3)$$

Bu ifadede $\pi(0) = 0$, $k = 1, 2, \dots, m$ olmak üzere $C_{0k} = 0$ ve $j = 1, 2, \dots, n$ olmak üzere $C_{j0} = 0$ olarak alınır.

Akış tipi çizelgeleme problemlerinin temel varsayımları aşağıdaki gibidir⁽⁸⁸⁾.

- j işi birim zamanda tek bir k makinasında işlem görebilir.
- k makinası birim zamanda tek bir j işini yapabilir.
- İşin bölünmesine izin verilmez, öyle ki k makinası j işini yapmaya başladığı zaman kesintiye uğratmaksızın tamamlaması gerekir.
- Bütün işler birbirinden bağımsız ve 0 zamanında hazırdır.
- İşlerin makinalardaki hazırlık zamanları önemsizdir ve çizelgeleme sırasında göz ardı edilebilir.
- Makinalar çizelgeleme boyunca çalışır durumdadır.
- Süreç içi stoğa izin verilir. Herhangi bir iş, çizelgenin bir sonraki adımında ihtiyaç duyduğu makina meşgulse, ilgili makinanın kuyruğuna girerek bekleyebilir.

$F // \sum T_j$ notasyonu ile gösterilen toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü akış tipi çizelgeleme problemleri için genel olarak $(n!)^m$ adet alternatif çizelge söz konusu olur. İş geçişine izin verilmeyen ve $F / pmu / \sum T_j$ notasyonu ile gösterilen toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemlerinde takip eden makinalarda iş sırası değişmediğinden çözüm uzayında $n!$ alternatif çizelge bulunmaktadır⁽⁸⁸⁾.

2.1.2. $F / pmu / \sum T_j$ Problemlerinin Sınıfı

Bir çok problem, bazı karar değişkenlerinin fonksiyonu olarak ifade edilebilen amaç fonksiyonu ve kısıtlar şeklinde formüle edilebilir.

Kombinatoriyel terimi, karar deęişkenlerinin kesikli deęerler aldığı problemleri ifade eder. Bu problemlerde çözüm, tamsayıların veya bazı dięer kesikli unsurların sırası veya setidir. Bu tip problemlerin optimum çözümünü bulma problemi kombinatoriyel optimizasyon olarak adlandırılır⁽⁹⁰⁾. Akış tipi çizelgeleme problemleri de kombinatoriyel optimizasyon problemleri sınıfında yer almaktadır⁽⁹¹⁾.

Kombinatoriyel optimizasyon problemlerinin hesaplama karmaşıklığı teorisine göre, çizelgeleme problemleri P ve NP olmak üzere iki farklı sınıfa ayrılır⁽⁹²⁾. P sınıfı problemlerin temel özellięi, polinomial bir çözüm algoritmasına sahip olmalarıdır. NP sınıfı problemlerde ise, optimum çözüme ulaşma zamanında problem boyutuna baęlı polinomial olmayan artış hızları söz konusudur⁽⁹³⁾. Eęer NP sınıfındaki her problem polinomial zamanda bir P problemine dönüştürülebiliyorsa, ilgili P problemine NP -zor problem denir. Bununla birlikte P probleminin kendisi de NP sınıfında bir problemse bu durumda P problemine NP -tam problem denir⁽⁹⁴⁾. NP -tam problemlerin iki önemli özellięi aşıęıdaki şekilde verilebilir⁽⁹²⁾:

1. Hiçbir NP -tam problemi işlem zamanı problem boyutunun polinomial fonksiyonu olan bir çözüm algoritmasına sahip deęildir.
2. NP -tam sınıfındaki her hangi bir problem için polinomial zamanlı bir algoritma bulunabilirse, bu sınıftaki tüm problemler için de polinomial zamanlı bir algoritma bulunabilir. Bu durumda $P = NP$ olduęu gösterilmiş olacaktır.

Toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri tek bir makinaya sahip olsalar

bile NP -zor sınıfında yer almaktadır⁽⁴⁰⁾.

2.1.3. $F / \text{permu} / \sum T_j$ Problemlerinin Çözüm Yöntemleri

TGPATP çözüm yaklaşımları üç temel grupta toplanabilir. Bunlar optimumu bulan yöntemler, sezgisel yöntemler ve meta sezgisel yöntemlerdir.

Optimumu bulan yöntemler sınıfında literatürde geliştirilen çeşitli dal sınır algoritmaları yer alır. Bu algoritmaların temel eksikliği, dal sınır çözüm ağacının boyutunu azaltabilecek baskın bir alt sınır değeri veren özelliği kullanmamalarıdır. Bunun sonucu olarak bu yöntemler sadece küçük boyutlu problemlere uygulanabilmektedir⁽⁹⁵⁾.

Literatürde önerilen sezgisel yöntemler arasında bulunan Liste Sıralama Algoritmaları atölye tipi çizelgeleme problemlerinin çözüm yöntemlerinden esinlenilerek Kim⁽³³⁾ tarafından önerilmiştir⁽⁹⁵⁾. Şimdiye kadar çizelgeye yerleştirilmiş işlerin sırası σ , $t = \sum_{i \in \sigma} P_{i1}$ olmak üzere çizelgeye yerleştirilmemiş işlerin seçim için göz önünde bulundurulacağı zaman ise t olsun; $i \notin \sigma$ olmak üzere i işi σ sırasından hemen sonra çizelgeye eklenmesi durumunda i işinin tamamlanma zamanı da $C_i(\sigma)$ olsun. Bu durumda, aşağıdaki değerlerin en küçüğüne sahip i işi t zamanında çizelgeye eklenmek üzere seçilecektir⁽³³⁾:

- En Küçük Teslim Tarihi (EDD): d_i ;
- Boş Zaman (SLACK): $d_i - C_i(\sigma)$;

- Kalan İş Başına Boş Zaman (S/RMWK): $(d_i - C_i(\sigma)) / \sum_{j=1}^m P_{ij}$;
- Modifiye Edilmiş Teslim Tarihi (MDD): $\max\{d_i, C_i(\sigma)\}$.

Kim'in⁽³³⁾ yaptığı deneysel çalışmalarda liste sıralama yöntemleri içerisinde MDD algoritmasının en iyi performansa sahip olduğu görülmüştür.

Literatürde geliştirilmiş diğer sezgisel algoritmalar ise aşağıdaki şekilde özetlenebilir.

- Değiştirilmiş NEH Sezgiselleri: Nawaz v.d.⁽⁹⁶⁾ tarafında maksimum tamamlanma zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri için önerilen sezgiselin toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü problemlere uyarlanmış halidir. Uyarlama işlemi (1) başlangıç sırasının, işlerin teslim tarihine göre küçükten büyüğe sıralanması (EDD) yoluyla elde edilmesi⁽³³⁾ veya işlerin teslim tarihine göre büyükten küçüğe sıralanması (LDD) yoluyla elde edilmesi⁽³³⁾ veya her adımda $\max\{d_i, C_i(\sigma)\}$ değeri en küçük olan işin σ kısmi çizelgesine eklenmesi yoluyla elde edilmesi⁽⁵⁵⁾ ve (2) performans ölçütü olarak maksimum tamamlanma zamanı yerine toplam geç bitirme zamanının göz önünde bulundurulması şeklinde yapılmaktadır. Orijinal NEH Algoritması bir kısmi birerleme yöntemi olup, en büyük toplam işlem zamanına sahip işin diğer işlerden daha öncelikli olması gerektiği varsayımına dayanmaktadır. Algoritmadaki toplam birerleme sayısı $(n(n+1)/2)-1$ adet olup bunların n adetinde çizelgenin tamamı göz önüne alınmaktadır. Orijinal NEH algoritmasının işlem adımları

aşağıda verilmiştir⁽⁹⁶⁾:

1. Sıralama İşlemi: Aşağıdaki eşitliği kullanarak her bir iş için toplam işlem zamanı T_j 'yi hesapla ve işleri T_j değerlerine göre büyükten küçüğe sırala.

$$T_j = \sum_{i=1}^m P_{ij} \quad (2.4)$$

2. Adım 1'de oluşturulan listedeki işlerden ilk ikisini al ve mümkün iki çizelgeden en küçük maksimum tamamlanma zamanı değerine sahip olanını tespit et. Algoritmanın kalan adımlarında bu iki işin göreceli pozisyonlarını sabitle. $j = 3$ al.
3. Adım 1'de oluşturulan listeden j pozisyonunda bulunan işi al ve bir önceki adımda oluşturulan çizelgedeki işlerin göreceli pozisyonlarını değiştirmeden bu işi j adet mümkün pozisyona yerleştirerek en küçük maksimum tamamlanma zamanına sahip olan çizelgeyi bul. Bu adımdaki birerleme sayısı j değerine eşit olacaktır.
4. Eğer $j = n$ ise dur, değilse $j = j + 1$ al ve Adım 3'e git.

- En Erken Bölüştürülmüş Teslim Tarihi Metodu: Parthasarathy v.d.⁽⁵³⁾ tarafından önerilen bu sezgisel işe ait teslim tarihinin işin operasyonlarına bölüştürülmesi temeline dayanır. Bu algoritmanın işlem adımları aşağıda verilmiştir⁽⁵³⁾:

1. Aşağıdaki formülasyonu kullanarak, işe ait teslim tarihini işin operasyonlarına paylaştır.

$$d'_{i1} = (d_i \times P_{i1}) / \sum_{j=1}^m P_{ij} \text{ ve}$$

$$d'_{ij} = d'_{ij-1} + (d_i \times P_{ij}) / \sum_{j=1}^m P_{ij} \text{ for } j = 2, 3, \dots, m. \quad (2.5)$$

2. İşleri, işe ait her bir operasyon (veya makina) bazında d'_{ij} değerlerine göre küçükten büyüğe sırayarak m adet (m : makina sayısı) farklı çizelge oluştur. Bu çizelgeler içerisinde en küçük toplam geç bitirme zamanı performans ölçütü değerine sahip olan çizelgeyi seç.

Kombinatoryal optimizasyon problemlerine en çok uygulanan meta sezgisel algoritmalar; Tabu Arama (TA), Tavlama Benzetimi (TB) ve Genetik Algoritmalar (GA)dır. Bu yöntemler içerisinde TGPATP için geliştirilen algoritmalar TA (Kim⁽⁴⁰⁾, Armentano v.d.⁽⁵⁵⁾) ve TB (Parthasarathy v.d.⁽⁵³⁾, Hasija v.d.⁽⁷⁰⁾) algoritmaları olup hiç bir GA çalışması yapılmamıştır. TA ve TB ile ilgili özet bilgiler aşağıda sunulmuş olup GA ile ilgili detaylı bilgi Bölüm 2.2 kapsamında verilmiştir.

İnsan hafızasını taklit eden Tabu Arama algoritması 1986 yılında Glover⁽⁹⁷⁾ tarafından önerilmiştir. Komşu arama yaklaşımı ve tabu listesi, bu metodun temel prensiplerini oluşturur. Metod, arama işlemi yönlendirmek ve aramanın yerel optimuma düşmesini engellemek için arama sürecine çeşitli kısıtlamalar getirir. TA algoritması, bir başlangıç çözümünden başlayıp mevcut çözümün belirlenen kısıtı sağlayan komşusuna ilerlemek yoluyla arama işlemi gerçekleştirir. Her bir iterasyonda mevcut çözümün komşuları içerisinde en iyi performans ölçütü değerine sahip olan çözüme, mevcut çözümden iyi olmasına bakılmaksızın hareket edilir. Metod, döngüyü

engellemek ve arama işlemini çözüm uzayının aranmamış bölgelerine yönlendirmek için belirli özelliklere sahip bazı çözümleri yasaklar (tabu). Bu işlemi bazı özel kısa ve uzun dönemli hafıza fonksiyonlarıyla gerçekleştirir. Kısa dönemli ve yenilik tabanlı hafıza “Tabu Listesi” olarak adlandırılır. TA algoritmasının en basit yapısında bulunması gereken temel bileşenler: başlangıç çözümü, komşu çözüm üretme mekanizması, tabu listesi ve durdurma kriteridir⁽⁹⁸⁾.

Fiziksel tavlama sürecini taklit eden Tavlama Benzetimi algoritması, kombinatoriyal optimizasyon problemlerinde iyi sonuçlar veren bir komşu arama tekniğidir. Verilen bir başlangıç çözümünden arama işlemine başlayan algoritma, her bir safhada mevcut çözümün komşuları arasından birine (1) komşu çözümün amaç fonksiyonu değeri mevcut çözümün amaç fonksiyonu değerine eşit veya küçükse, (2) komşu çözümün amaç fonksiyonu değeri daha büyük olmasına rağmen amaç fonksiyonları arasındaki fark ve mevcut durum sıcaklığına bağlı bir fonksiyonla hesaplanan olasılık değeri sağlanıyorsa, hareket eder. Pozitif bir sayı olan mevcut durum sıcaklığı arama süreci ilerledikçe belirlenen bir soğutma planına uygun şekilde azaltılır. Bu yolla kötü çözümlerin seçilme olasılığı yüksek bir değerden sıfıra doğru hareket eder. Bunun sonucu olarak, arama sürecinin başlangıcında oldukça kötü çözümlerin bile seçilme olasılığı yüksekken, sürecin sonuna doğru genellikle sadece mevcut çözümü iyileştiren komşu çözümler seçilecek, yani metod mevcut çözümü yerel optimuma yakınsayacaktır. Buna karşın yüksek sıcaklıklarda arama sürecini çözüm bölgesinin aranmamış bölümlerine yönlendiren bir çok hareket yapılacağından, daha iyi yerel optimuma ulaşılacağı ümit edilir⁽⁹⁹⁾.

2.2. Genetik Algoritmalar

Genetik Algoritmalar (GA), doğal seçim ve doğal genetik mekanizması temeline dayanan arama algoritmalarıdır⁽¹⁰⁰⁾. Holland ve meslektaşları tarafından 1960'lı yıllarda Michigan Üniversitesinde geliştirilmiş olan GA'nın teorik ve sistematik yapısı Holland'ın 1975 yılında yayınlanan "Doğal ve Yapay Sistemlerde Adaptasyon" isimli kitabında sunulmuştur. Yöneylem araştırması açısından bakıldığı zaman GA, rastsal arama işleminin zeki olarak kullanımı şeklinde ifade edilebilir. Literatürde çözüm uzayının rastsal olarak örneklenmesi yoluyla elde edilen bilgileri kullanarak problemlerin çözülmesi yönünde bir çok çalışma bulunmasına rağmen, bir çok problem tipi için GA yaklaşımı bu yöntemlerden daha esnek ve genel bir yapı sunar⁽⁹⁴⁾.

Genetik Algoritma terimi, kompleks yapıların bileşenler vektörü olarak ifade edilmiş hali ile biyoloji bilimindeki kromozomların genetik yapıları arasındaki benzerlikten esinlenilerek ortaya atılmıştır. Doğal sistemlerde bitki ve hayvanların seçici çoğalması, yavruların ebeveynlerde bulunan genetik yapılardaki iyi özelliklerini ihtiva edecek şekilde kromozom birleşmeleriyle ortaya çıkması şeklinde gerçekleşmektedir. Benzer bir yaklaşımla yapay sistemlerde (yani GA 'da) kompleks problemlerin iyi çözümlerine yönelme işlemi, mevcut çözümleri oluşturan bilgilerin seçici kullanımı yoluyla yeni çözümlerin elde edilmesi şeklinde gerçekleşmektedir⁽⁹⁴⁾.

GA'yı diğer arama ve optimizasyon yöntemlerinden farklı kılan dört özellik aşağıda sunulmuştur⁽¹⁰⁰⁾:

1. GA parametrelerin kendileri yerine parametre setlerinin kodlarıyla ilgilenirler.

2. GA tek bir nokta yerine noktaların oluşturduğu popülasyondan arama yaparlar.
3. GA amaç fonksiyonu bilgisini kullanırlar. Türev veya başka bir yardımcı bilgiyi kullanmazlar.
4. GA deterministik kurallar yerine stokastik geçiş kurallarını kullanırlar.

GA'nın temelinde potansiyel çözümlerin genetik olarak kodlanması yoluyla kromozomların elde edilmesi ve bu kromozomlara genetik operatörlerin uygulanması yer almaktadır⁽¹⁰¹⁾. Yani GA, her biri probleme ait potansiyel bir çözümü ifade eden bireylerin oluşturduğu popülasyonu değişime uğratarak arama işlemini gerçekleştirir. Her bir birey uygunluk değerine bağlı bir derece verilmesi amacıyla değerlendirilir. Yeni bireylerin oluşturulması amacıyla genetik operatörlerin uygulanması bazı bireyleri stokastik dönüşüme uğratar. GA kapsamında iki tip dönüştürme işleminden bahsedilebilir. Bunlardan ilki, tek bir birey üzerinde değişim yapmak yoluyla yeni bireylerin üretilmesini sağlayan mutasyon işlemi, diğeri ise iki farklı bireye ait bilgileri birleştirerek yeni bireyler elde edilmesini sağlayan çaprazlama işlemidir. Bu yolla elde edilen ve yavru olarak isimlendirilen yeni bireyler daha sonra değerlendirme işlemine tabi tutulur. Ebeveyn popülasyon ve yeni oluşan yavru popülasyondaki uygunluğu yüksek bireylerin seçilmesi yoluyla yeni popülasyon oluşturulur. Belirtilen üretim işleminin çok sayıda tekrarlanması sonucu algoritmanın, problem için optimal veya yerel optimal ifade eden en iyi bireylere yakınsaması beklenir⁽¹⁰²⁾.

Potansiyel çözümlerin genetik olarak kodlanması, orijinal problemin bir bölgeden başka bir bölgeye dönüştürülmesine eşdeğerdır. Genetik

algoritmaların başarısında potansiyel çözümlerin genetik olarak kodlanması kritik bir rol oynar. İyi bir kodlama problemin çözümünü kolaylaştıracak, kötü bir kodlama ise arama sürecini ters yönde etkileyecektir. Yani her hangi bir probleme genetik algoritmaların uygulanmasında kritik nokta etkin aramayı sağlayacak bir kodlama yapısının tespit edilmesidir⁽¹⁰¹⁾.

Bir genetik algoritma, aşağıda özetlendiği gibi, genel olarak beş temel bileşene sahiptir⁽¹⁰²⁾:

1. Probleme ait çözümlerin genetik algoritmaya uygun yapıda gösterilmesi,
2. Başlangıç popülasyonunda yer alacak çözümlerin elde edilme yöntemi,
3. Çözümlerin uygunluk değerlerine göre derecelendirilmesini sağlayacak değerlendirme fonksiyonu,
4. Yeniden üretim sırasında bireylerin genetik bileşimini değiştirecek genetik operatörler,
5. Genetik algoritma parametrelerine ait değerler.

Basit genetik algoritma olarak da adlandırılan standart genetik algoritmada⁽¹⁰⁰⁾ ikili kodlama yapısı, tek noktalı çaprazlama ve bit değiştirme mutasyon operatörü kullanılmaktadır. İkili kodlama yapısında probleme ait her bir çözüm belli sayıda 0 veya 1 değeri içeren kromozomla ifade edilir. Tek noktalı çaprazlama işleminde ise seçilen bir çaprazlama noktasına kadar olan genler birinci ebeveynden geriye kalanlar ise diğer ebeveynden alınarak birinci yavru kromozom oluşturulur. İkinci yavru kromozom elde edilirken ebeveynlerin rolü değiştirilir⁽¹⁰¹⁾. Aşağıda verilen ve her biri altı değişkene

sahip a ve b kromozomlarının her hangi bir probleme ait iki farklı çözümlü ifade ettiği varsayalım⁽⁹⁰⁾:

$$(a_1, a_2, a_3, a_4, a_5, a_6) \text{ ve } (b_1, b_2, b_3, b_4, b_5, b_6)$$

Çaprazlama noktasının rastsal olarak 1,...,5 sayıları arasından seçildiği ve 2 olduğu varsayılırsa, tek noktali çaprazlama işlemi sonucunda elde edilecek yavru kromozomlar aşağıdaki şekilde ortaya çıkacaktır.

$$(a_1, a_2, b_3, b_4, b_5, b_6) \text{ ve } (b_1, b_2, a_3, a_4, a_5, a_6)$$

Standard GA'da mutasyon işlemi ise bit bazında gerçekleştirilir. Öyle ki, kromozomda bulunan her bir bit, belirli bir mutasyon olasılığıyla, 0 ise 1 değerine veya 1 ise 0 değerine dönüştürülür⁽¹⁰¹⁾. Örneğin aşağıdaki kromozomun her hangi bir probleme ait bir çözümü ifade ettiği varsayalım⁽⁹⁰⁾:

(011001)

Bu kromozomda üçüncü ve beşinci genlerin mutasyona uğratılması sonucu oluşacak mutant kromozom aşağıdaki şekilde ortaya çıkacaktır.

(010011)

Standard genetik algoritmanın adımları Şekil 2.3'de verilmiştir⁽¹⁰¹⁾.

1. $P(0)$ başlangıç popülasyonunu üret ve $i = 0$ al;
2. Bitiş Koşulu sağlanıncaya değin aşağıdaki işlemleri tekrar et.
 - a. $P(i)$ 'yi değerlendir.
 - b. Uygunluk değerlerine bağlı olarak ebeveynleri $P(i)$ 'den aşağıdaki yöntemle seç:

f_1, f_2, \dots, f_n popülasyonda bulunan n adet bireyin uygunluk değerleri ise, i -nci bireyi aşağıdaki olasılıkla seç:

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

Bu yöntem Rulet Çemberi Seçim Mekanizması olarak adlandırılır.
 - c. Seçilen ebeveynlere çaprazlama operatörünü uygula;
 - d. Çaprazlama sonucunda oluşan bireylere mutasyon operatörünü uygula;
 - e. $P(i)$ 'deki ebeveynleri yavrularla değiştirerek $P(i+1)$ popülasyonunu oluştur ve $i = i+1$ al.

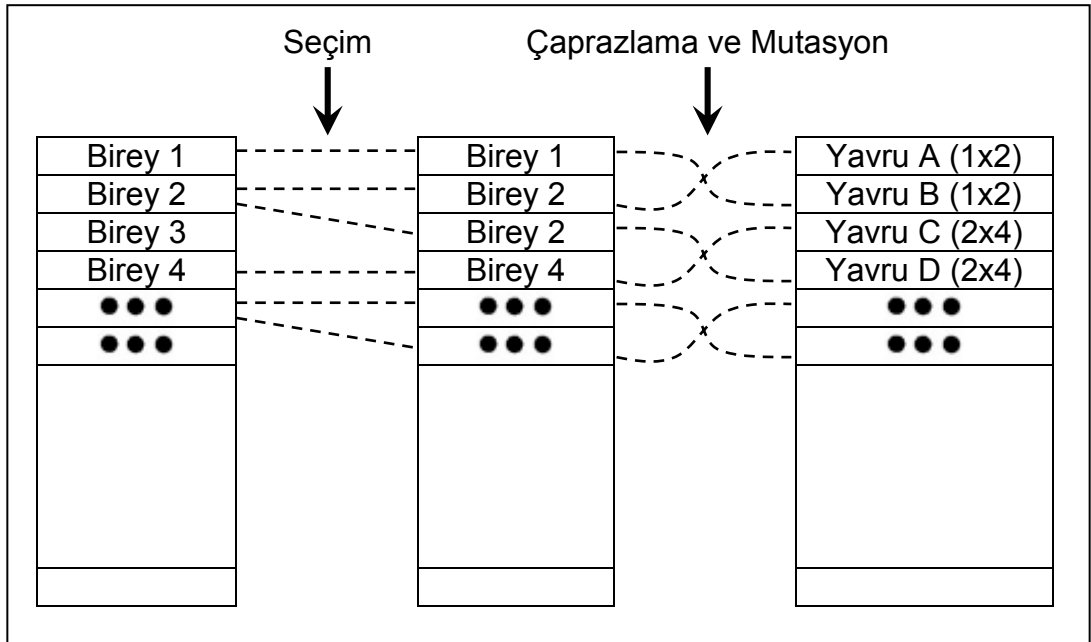
Şekil 2.3: Standard Genetik Algoritma

Çaprazlama ve mutasyon işlemi yeni bireylerin üretilmesi amacıyla kullanılan temel araçlardır. Bununla birlikte, bu operatörlerin hangi kromozomlara uygulanacağına belirlenmesi, popülasyona bütün olarak hangi işlemlerin uygulandığının anlaşılması açısından oldukça önemlidir⁽⁹⁰⁾. Seçim mekanizmasının temelinde seçim işleminin kromozomların uygunluk değerleriyle bağlantılı olması gerektiği yatmaktadır⁽¹⁰³⁾. GA üzerinde ilk çalışmaları yapan Holland, ebeveynlerden en az birinin uygunluk değerleri baz alınarak seçilmesini önermektedir⁽⁹⁰⁾. Arama sürecinde daha iyi bireylere ulaşılması için, uygunluk değeri açısından iyi bireylerin kötü bireylere kıyasla daha yüksek seçilme olasılığına sahip olmaları gerekir⁽¹⁰¹⁾. Seçim operatörü en iyi olanın yaşamasını öngören doğal seçim mekanizmasının yapay bir

versiyonudur⁽¹⁰⁰⁾.

GA arama sürecinde uygunluk değerleri hesaplanmış M adet kromozomdan (yani potansiyel çözümden) oluşan bir popülasyon kullanır. Her bir kromozom probleme ait bir çözümlü kodlarken uygunluk değeri ise kodladığı çözümün amaç fonksiyonu değeriyle ilişkili olacak şekilde hesaplanır⁽⁹⁴⁾.

Genetik algoritmalar mevcut bireylerin özelliklerini taşıyan yeni bireylerin üretilmesi yoluyla evrim geçiren çözümlerin oluşturduğu popülasyon üzerinde işlem yapar. Her bir iterasyonda, seçim operatörü yeni yavruların üretilmesinde kullanılacak ebeveynlerin belirlenmesini sağlarken çaprazlama operatörü yeni yavruların üretilmesini, mutasyon operatörü ise yeni yavruların özelliklerini değiştirerek popülasyonun çeşitlenmesini sağlar. Şekil 2.4 genetik operatörlerin fonksiyonunu göstermektedir⁽¹⁰³⁾.



Şekil 2.4 :Genetik Algoritma Operatörleri

GA çözümlerin oluşturduğu popülasyon üzerinde seçici üreme stratejilerini kullandığından, arama süreci ilerledikçe elde edilen çözümlerin kalitesi de artmaktadır. Çaprazlama ve mutasyon gibi basit genetik operatörler eski çözümlerin sahip olduğu özellikleri kullanarak yeni çözümlerin elde edilmesinde kullanılmakta, bu yolla popülasyonu gittikçe iyileştirmektedir⁽⁹⁰⁾.

Genetik benzerlikten yola çıkılırsa, problemdeki değişkenler gen, bu değişkenlerin alabileceği mümkün değerler alel, değişkenin kromozomdaki pozisyonu lokus, algoritmanın arama sürecinde kullandığı kodlanmış çözüm genotip, genotiplerin kodlarının çözülmesiyle elde edilen parametrelerin seti ise fenotip olarak adlandırılır⁽⁹⁰⁾.

Sadece basit çaprazlama ve mutasyon operatörü kullanan Standart GA'nın bir çok problem için oldukça etkin çözümlere ulaşabildiği kanıtlanmıştır⁽⁹⁴⁾. Faydalı olacağı düşüncesiyle Standart GA'nın işleyişi basit bir örnekle daha detaylı olarak açıklanacaktır. Bu amaçla aşağıda verilen problemi göz önünde bulunduralım⁽⁹⁰⁾:

$$\text{Maksimum } f(x) = x^3 - 60x^2 + 900x + 100 \quad (2.6)$$

Bu problemde x , 5 bit uzunluğunda ikili kodlama yapısıyla, [0,31] aralığında değer alacak şekilde kodlanmıştır. Bu durumda, örneğin (00101) kromozomu için, $x = 5$ ve $f = 3225$ değerini alacaktır. GA'nın başlatılması amacıyla Çizelge 2.1'de verildiği gibi rastsal olarak beş adet çözüm üretildiği ve başlangıç popülasyonunun (büyüklüğü 5 olacak şekilde) oluşturulduğu varsayalım.

Çizelge 2.1:Rastsal Olarak Üretilmiş Beş Adet Kromozomdan Oluşan Başlangıç Popülasyonu

No	Kromozom	x	$f(x)$	$P[Seçim]$
1	10011	19	2399	0.206
2	00101	5	3225	0.277
3	11010	26	516	0.044
4	10101	21	1801	0.155
5	01110	14	3684	0.317
Ortalama uygunluk:2325				

$f(x)$ değerleri uygunluk değerleri olarak göz önünde bulundurulursa, her bir kromozomun birinci ebeveyn olarak seçilme olasılığı doğrudan ilgili kromozomun uygunluk değeriyle orantılıdır ve bu olasılıklar $P[Seçim]$ sütununda sunulmuştur. 0-1 aralığında üretilecek uniform rastsal sayı ve $P[Seçim]$ olasılıkları kullanılarak ebeveynler seçilmiştir. Rastsal seçilmiş çaprazlama noktası kullanılarak $P_c = 1$ çaprazlama olasılığıyla çaprazlama işlemi gerçekleştirilmiş, çaprazlama işlemi her bir gen için $P_m = 0.02$ mutasyon olasılığıyla mutasyon işlemi izlemiştir. Tipik bir iterasyonun sonucu Çizelge 2.2’de sunulmuştur.

Belirtilen arama işlemine seçim olasılıklarının tekrardan hesaplanması ve seçim, çaprazlama ve mutasyon döngüsünün uygulanması yoluyla bitirme koşulu sağlanıncaya değin devam edilir.

Belirtilen basit örnekte anlatılan yöntem üzerinde çeşitli varyasyonlar söz konusu olabilir. Bunlardan bazıları⁽⁹⁴⁾:

- Örnekte verilen yöntem her bir iterasyonda popülasyonun tamamını blok olarak yeniden üretmektedir. Her seferinde üretilen yavru kromozom popülasyondan rastsal olarak seçilecek bir kromozomun yerini alabilir.

Çizelge 2.2: Tipik Bir İterasyon Sonucunda Elde Edilen Yeni Popülasyon

Adım	Ebeveyn 1	Ebeveyn 2	Çapraz. Noktası	Mutasyon?	Yavru	
					Kromozom	$f(x)$
1	1	2	4	HHHEH	10001	2973
2	5	3	2	HHHHH	01010	4100
3	5	2	3	HHHHH	01101	3857
4	4	2	1	HEHHH	11101	129
5	2	5	4	HHHHH	00100	2804
Ortalama uygunluk: 2773						

- Her bir çaprazlama işleminde iki adet yavru üretmesine rağmen örnekte bunlardan sadece birisi yeni popülasyona kopyalanmıştır. Her iki yavru da yeni popülasyona kopyalanabilir.
- Örnekte yüksek değişkenlik içeren yerine koyarak stokastik seçim prosedürü uygulanmıştır. Yerine koymadan stokastik seçim prosedürü kullanılabilir.
- Örnekte uygunluk değerlerinin tanımlanmasında doğrudan $f(x)$ değerleri kullanılmıştır. Bunun yerine kullanılacak ölçeklendirme yoluyla sıralama prosedürleri iyi ve kötü kromozomları daha etkin olarak ayırt edebilir.

3. ARAŞTIRMA BULGULARI

Bu bölüm kapsamında toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemlerinin (TGPATP) çözümü için geliştirilen Genetik Algoritma (GA) yaklaşımının adımları ve bileşenleri sunulmuş, algoritmanın parametre değerleri verilmiş ve geliştirilen GA, literatürde aynı problem üzerinde en iyi performansı sergileyen sezgiselle karşılaştırılarak sonuçları belirtilmiştir.

3.1. Geliştirilen Genetik Algoritma

TGPATP 'in çözülmesi amacıyla bu çalışma kapsamında geliştirilen ve probleme özgü bilgileri kullanan genetik algoritmanın adımları aşağıda verilmiştir.

Adım 1 : Başlangıç işlemlerini yap.

- ◆ *Çaprazlama _ Operatörü , Mutasyon _ Operatörü ,*
Popülasyon _ Genişliği , Çaprazlama _ Oranı , Mutasyon _ Oranı ,
Benzerlik _ Eşik _ Değeri , İyileştirme _ İterasyon _ Sayısı
parametrelerini belirle.

- ◆ *Mevcut _ İterasyon = 0 , İyileşen _ Son _ İterasyon = 0* al.

Adım 2 : Başlangıç popülasyonunu üret (Bakınız Bölüm 3.1.2).

Adım 3 : Başlangıç popülasyonunu değerlendir.

- ◆ Bireylerin amaç fonksiyonu değerlerini hesapla (Bakınız Bölüm 2.1.1).

- ◆ Bireylerin uygunluk deęerlerini hesapla (Bakınız Bölüm 3.1.3).
- ◆ Bireyleri uygunluk deęerlerine göre büyükten küçüęe sırala.
- ◆ Bireylerin seęilme olasılıklarını hesapla (Bakınız Bölüm 3.1.3).
Hesaplanan seęilme olasılıklarını kullanarak birikimli seęilme olasılıklarını hesapla.

Adım 4 : Parametreleri güncelle.

- ◆ $Mevcut_İterasyon = Mevcut_İterasyon + 1$ yap.
- ◆ $Son_İyi_Deęer = \{Popülasyondaki\ en\ iyi\ bireyin\ amaç\ fonksiyonu\ deęeri\}$ olarak ata.

Adım 5 : Mevcut popülasyonu kullanarak yeni popülasyonu üret.

- ◆ Mevcut popülasyondaki en iyi bireyi yeni popülasyona aynen kopyala.
- ◆ Çaprazlama : Yeni popülasyonun genişlięi $Popülasyon_Genişlięi$ parametre deęerine ulařana kadar ařaęıdaki işlemleri tekrar et.
 - Belirlenen seęim mekanizmasını kullanarak (Bakınız Bölüm 3.1.3) mevcut popülasyondan iki ebeveyn seę.
 - Çaprazlama kořulu saęlanıyorsa (yani üretilecek rastsal sayı $Çaprazlama_Oranı$ 'dan küçükse) seęilen ebeveynleri belirlenen $Çaprazlama_Operatörü$ 'nü kullanarak çaprazla ve oluřan yavruları yeni popülasyona ekle. Kořul saęlanmıyorsa seęilen ebeveynleri deęişiklik yapmaksızın yeni popülasyona ekle.
- ◆ Mutasyon : Yeni popülasyondaki ilk birey dıřındaki her bir birey için, mutasyon kořulu saęlanıyorsa (yani üretilecek rastsal sayı

$Mutasyon_Oranı$ 'dan küçükse) belirlenen

$Mutasyon_Operatörü$ 'nü kullanarak bireyi mutasyona uğrat.

- ◆ Yeni popülasyonu mevcut popülasyon olarak belirle.

Adım 6 : Mevcut popülasyonu değerlendir (Adım 3'de verilen işlemlerin aynısını uygula).

Adım 7 : Mevcut popülasyondaki en iyi bireyin amaç fonksiyonu değeri

$Son_İyi_Değer$ 'den daha iyi ise aşağıdaki işlemleri yap.

- ◆ Popülasyondaki en iyi bireyi Gecikme Tabanlı İyileştirme Algoritmasını kullanarak (Bakınız Bölüm 3.1.6) iyileştir.

- ◆ $İyileşen_Son_İterasyon = Mevcut_İterasyon$ al.

Adım 8 : $Mevcut_İterasyon > İyileşen_Son_İterasyon > İyileştirme_İterasyon_Sayısı$

koşulu sağlanıyorsa aşağıdaki işlemleri yap.

- ◆ $İyileşen_Son_İterasyon = Mevcut_İterasyon$ al.

- ◆ Popülasyondaki en son bireyi Gecikme Tabanlı İyileştirme Algoritmasını kullanarak (Bakınız Bölüm 3.1.6) iyileştir ve bu bireyin iyileştirmeden önceki halini $İyileştirilen_Son_Birey$ olarak sakla.

- ◆ Popülasyondaki son bireyin bir önünde yer alan bireyden başlayarak popülasyonun ilk bireyine kadar aşağıdaki adımları tekrarla.

- Mevcut bireyle $İyileştirilen_Son_Birey$ benzerlik oranını hesapla (Bakınız Bölüm 3.1.7).

- Hesaplanan benzerlik oranı $Benzerlik_Eşik_Değeri$ 'den küçükse mevcut bireyi Gecikme Tabanlı İyileştirme

Algoritmasını kullanarak (Bakınız Bölüm 3.1.6) iyileştir ve mevcut bireyin iyileştirmeden önceki halini *İyileştirilen_Son_Birey* olarak sakla.

- ◆ Mevcut popülasyonu değerlendir (Adım 3'de verilen işlemlerin aynısını uygula).
- ◆ Mevcut popülasyondaki en iyi bireyin amaç fonksiyonu değeri *Son_İyi_Değer*'den daha iyi ise *İyileşen_Son_İterasyon = Mevcut_İterasyon* olarak al.

Adım 9 :Bitirme koşulu sağlanmamışsa Adım 4'e git.

Adım 10:Popülasyondaki en iyi bireyi ulaşılan en iyi çözüm olarak yaz.

3.1.1. Parametre Kodlama

GA'ların başarısında potansiyel çözümlerin genetik olarak kodlanması kritik bir rol oynar⁽¹⁰¹⁾. Goldberg⁽¹⁰⁰⁾'e göre GA'da probleme ilişkin kodlama yapısının seçilmesinde göz önünde bulundurulması gereken iki temel prensip aşağıdaki şekildedir:

- Kodlama yapısı problemin anlaşılması açısından kısa ve basit kurallara sahip olmalıdır.
- Problemin doğal yapısını ifade edebilecek kabiliyete sahip bir kodlama yapısı oluşturabilen en küçük boyutlu alfabe seçilmelidir.

Permütasyon tipi çizelgeleme problemlerinde temel kod yapısının, diğer bir adıyla permütasyon tipi temsilin iş dizilişini basitleştirici bir etkisi bulunmaktadır⁽¹⁰⁴⁾. Bu nedenle bu çalışma kapsamında incelenen

permütasyon akış tipi çizelgeleme problemlerinin çözümlerinin kodlanmasında da temel kod yapısı kullanılmıştır.

Temel kod yapısında kromozom dizisi üzerinde bulunan her bir rakam bir işi temsil etmektedir. Örneğin 8 işe sahip bir akış tipi çizelgeleme probleminde {1-2-3-4-5-6-7-8} şeklinde verilen temel kod yapısındaki rakamlar sıra ile problemdeki işleri temsil etmektedir⁽¹⁰⁴⁾.

3.1.2. Başlangıç Popülasyonu

Yapılan bazı çalışmalar başlangıç popülasyonunda rastsal üretilmiş çözümler yerine probleme özgü sezgisel yöntemlerle elde edilecek yüksek kalitedeki çözümlerin kullanılmasının GA'ın daha kısa sürede daha kaliteli çözümlere ulaşmasını sağlayabileceğini göstermiştir. Buna karşın bu yöntem GA'ın erken yakınsamasına sebep olabilir⁽⁹⁴⁾. Bu nedenle bu çalışma kapsamında önerilen genetik algoritmanın kullanacağı başlangıç popülasyonunun oluşturulmasında rastsal üretilmiş kromozomlar kullanılmıştır.

3.1.3. Seçim İşlemi

GA'da her bir kromozom probleme ait bir çözümü kodlarken, bu kromozomun uygunluk değeri ise ilgili çözümün amaç fonksiyonu değeriyle ilişkili olacak şekilde hesaplanır⁽⁹⁴⁾. Seçim mekanizmasının temelinde seçim işleminin kromozomların uygunluk değerleriyle bağlantılı olması gerektiği yatmaktadır⁽¹⁰³⁾. Arama sürecinde daha iyi bireylere ulaşılması için, uygunluk

değeri açısından iyi bireylerin kötü bireylere kıyasla daha yüksek seçilme olasılığına sahip olmaları gerekir⁽¹⁰¹⁾.

Bu açıklamalar ışığında seçim işlemi kapsamında incelenmesi gereken üç temel adım aşağıdaki şekilde verilebilir:

1. Popülasyondaki her bireyin amaç fonksiyonu değerinin yani toplam geç bitirme zamanının hesaplanması,
2. Amaç fonksiyonu değerleriyle bağlantılı olacak şekilde uygunluk değerlerinin ve seçim olasılıklarının hesaplanması,
3. Seçim mekanizması.

Akış tipi çizelgeleme problemlerinde, verilen bir çözümün toplam geç bitirme zamanının hesaplanmasıyla ilgili formülasyon Bölüm 2.1.1 kapsamında sunulmuştu. İncelenen problemin en küçükleme problemi olması nedeniyle popülasyondaki bireylere ait bu formülasyonla bulunan amaç fonksiyonu değerleri kullanılarak uygunluk değerlerinin hesaplanması gerekir. Bu süreçte göz önünde bulundurulması gereken temel kriter daha iyi bireylerin daha büyük uygunluk değerine sahip olması gerekliliğidir. Bu çalışma kapsamında popülasyondaki bireylere ait uygunluk değerlerinin hesaplanmasında Gen v.d.⁽¹⁰²⁾'nin maksimum ağırlıklı mutlak gecikme zamanının en küçüklenmesi performans ölçütlü paralel makineli çizelgeleme problemleri için önerdiği aşağıdaki formülasyon kullanılmıştır.

$$f_t = \frac{1}{O_t}, \quad t = 1, 2, \dots, n \quad (3.1)$$

Bu formülasyonda f_t popülasyondaki t . bireyin uygunluk değerini, O_t bireyin amaç fonksiyonu değerini, n ise popülasyon genişliğini ifade

etmektedir.

Bu çalışma kapsamında kullanılan seçim mekanizması Rulet Çemberi Seçim Mekanizması'dır. Genişliği n olan bir popülasyonda bireylerin uygunluk değerleri sırasıyla f_1, f_2, \dots, f_n olmak üzere, bu mekanizma bireylerin seçim olasılıklarını aşağıdaki formülasyona göre doğrudan uygunluk değerlerini kullanarak hesaplar⁽¹⁰¹⁾:

$$p_t = \frac{f_t}{\sum_{i=1}^n f_i}, \quad t = 1, 2, \dots, n \quad (3.2)$$

Bu formülasyondaki p_t popülasyondaki t . bireyin seçilme olasılığını ifade etmektedir. Seçilme olasılıklarının bu şekilde hesaplanması bazı durumlarda çeşitli problemlere sebep olmaktadır. Örneğin popülasyonda bir veya birkaç birey çok uygun olmasına karşın optimum çözümü ifade etmiyorlarsa ve kalan bireylerde iyi değilse, iyi bireyler hızlı bir şekilde tüm popülasyona egemen olacak yani popülasyon büyük ölçüde iyi bireylere benzer hale gelecektir. Bu durum ise popülasyonun potansiyel olarak daha iyi bireyleri aramasına engel olacaktır. Diğer taraftan popülasyondaki tüm bireyler uygunluk değerleri açısından birbirlerine çok benziyorlarsa, seçim olasılıkları da birbirlerine çok yakın olacaktır. Bu durum da benzer şekilde popülasyonun potansiyel olarak daha iyi bireyleri aramasına engel olacaktır. Belirtilen problemlerin ortadan kaldırılması amacıyla literatürde çeşitli ölçeklendirme metotları önerilmiştir. Bu metotlar seçim olasılıkları hesaplanmadan önce uygunluk değerlerinin ölçeklendirilmesinde kullanılır. Bu metotlardan biri, seçim olasılıklarının derece (yani rank) tabanlı hesaplanmasıdır. Bu metot bireyleri uygunluk değerlerine göre küçükten

büyükçe sıralayarak her bir bireye derece atamakta ve bu dereceleri seçim olasılıklarının hesaplanmasında kullanmaktadır ⁽¹⁰¹⁾. Şekil 3.1’de bu metoda ilişkin bir örnek verilmiştir. Bu çalışma kapsamında da seçim olasılıklarının hesaplanmasında bu yaklaşım kullanılmıştır.

Birey	O_t	f_t	Sıralama işlemi ↓	Birey	O_t	f_t	Derece	p_t
2-3-5-1-4	53	0.019			5-2-3-4-1	64	0.016	1
5-2-3-4-1	64	0.016		2-3-5-1-4	53	0.019	2	2/14
3-2-1-4-5	22	0.046		4-1-2-3-5	40	0.025	3	3/14
4-1-2-3-5	40	0.025		3-2-1-4-5	22	0.046	4	4/14
1-5-2-4-3	22	0.046		1-5-2-4-3	22	0.046	4	4/14
Toplam:							14	14/14

Şekil 3.1:Popülasyondaki Bireylere Ait Seçim Olasılıklarının Derece Tabanlı Hesaplanması

Seçim mekanizmasına ilaveten çoğunlukla Elit Strateji adı verilen bir strateji kullanılmaktadır. Bu strateji popülasyondaki en iyi bir veya birkaç bireyi her hangi bir değişiklik yapmaksızın bir sonraki popülasyona kopyalar⁽¹⁰¹⁾. Yöntemin avantajı, en iyi uygunluk değerine sahip bireyin örnekleme hatası yada genetik operatörlerin kullanılması ile kaybolmasını önlemesidir⁽¹⁰⁵⁾. Bu çalışma kapsamında geliştirilen genetik algorithmada da popülasyondaki en iyi birey her hangi bir değişiklik yapılmaksızın bir sonraki popülasyona kopyalanmaktadır.

3.1.4. Çaprazlama Operatörü

Çaprazlama operatörü en basit anlatımla, ebeveyn kromozomlar arasında bilgi (gen) değişimini gerçekleştirir⁽¹⁰³⁾.

Önerilen genetik algorithmada bu çalışma kapsamında geliştirilen ve Gecikme Tabanlı Çaprazlama Operatörü (GTX) adı verilen çaprazlama operatörü kullanılmıştır.

Arama sürecinde daha iyi bireylere ulaşılması için, uygunluk değeri açısından iyi bireylerin kötü bireylere kıyasla daha yüksek seçilme olasılığına sahip olmaları gerekir⁽¹⁰¹⁾. Seçim mekanizmasının temelini oluşturan bu yaklaşımın çaprazlama işleminde de kullanılması amacıyla, göz önünde bulundurulmuş performans ölçütünün yapısından esinlenilerek yeni bir çaprazlama operatörü geliştirilmiş ve bu operatöre Gecikme Tabanlı Çaprazlama Operatörü adı verilmiştir. Formülasyonu Bölüm 2.1.1'de sunulduğu gibi verilen bir çizelgenin toplam geç bitirme zamanı performans ölçütünün hesaplanabilmesi için öncelikle çizelgede bulunan tüm işlerin tamamlanma zamanları ve teslim zamanlarının bir fonksiyonu olan gecikmelerinin (L_j) hesaplanması gerekmektedir. Bir işe ait gecikmenin pozitif olması o işin teslim zamanından sonra tamamlandığını, negatif olması ise teslim zamanından önce tamamlandığını göstermektedir. Bu noktadan hareketle, gecikmesi pozitif olan işlerin çizelgede daha erken pozisyonlara alınması, negatif olan işlerin ise sonraki pozisyonlara alınması yoluyla diğer işlere erken başlama fırsatının verilmesi fikri bu operatörün temelini oluşturmaktadır. Yani, tasarlanan yeni operatörde her hangi bir işin baz alınan ebeveyndeki gecikmesinin mutlak değeri ne kadar küçükse, o işin

pozisyonunu koruma olasılığı o derece yüksek olmaktadır. Pozisyonunu koruyamayan işler ise diğer ebeveyndeki göreceli pozisyonlarını koruyacak şekilde yeni yavruya kopyalanmaktadır. Bu operatörün temel adımları aşağıda sunulmuştur.

Adım 1 :Pozisyonunu koruyacak iş sayısını rastsal olarak belirle.

Adım 2 :Birinci ebeveyni göz önünde bulundurarak her bir işin mutlak gecikmesini hesapla.

Adım 3 :İşleri mutlak gecikmelerine göre büyükten küçüğe sırala. Mutlak gecikmesi en büyük iş veya işlere 1, ikinci sıradaki işlere 2 derecesini verecek şekilde tüm işleri derecelendir.

Adım 4 :Her bir iş için kendi derecesini toplam dereceye bölerek pozisyonunu koruma olasılığını hesapla ve bu olasılıkları birikimli olasılığa çevir.

Adım 5 :Hesaplanan olasılıkları kullanarak Rulet Çemberi yöntemiyle pozisyonunu koruyacak iş sayısı kadar işi birinci yavruya, birinci ebeveyndeki pozisyonunu koruyacak şekilde, taşı.

Adım 6 :Birinci yavruya boş kalan diğer pozisyonları ikinci ebeveynin işlerin göreceli pozisyonlarını koruyacak şekilde doldur.

Adım 7 :Ebeveynlerin görevlerini değiştirip Adım 2'den Adım 6'ya kadar uygulayarak ikinci yavruyu oluştur.

Önerilen operatörün uygulanmasına ilişkin bir örnek aşağıda sunulmuştur.

Örnek Uygulama

7 işli bir probleme ait iki farklı ebeveynin aşağıdaki şekilde olduğu

varsayılınsın.

1. ebeveyn kromozom:	3	1	5	7	2	4	6
2. ebeveyn kromozom:	5	3	6	2	1	4	7

Adım 1 :Pozisyonunu koruyacak iş sayısının rastsal olarak 3 üretildiği varsayılınsın.

Adım 2 :Birinci ebeveyn göz önünde bulundurularak işlere ait gecikme ve mutlak gecikmelerin aşağıdaki şekilde hesaplandığı varsayılınsın.

İş (j):	1	2	3	4	5	6	7
Gecikme (L_j):	-10	5	1	12	10	0	-6
Mutlak Gecikme ($ L_j $):	10	5	1	12	10	0	6

Adım 3-4:

İş (j):	4	1	5	7	2	3	6
Mutlak Gecikme ($ L_j $):	12	10	10	6	5	1	0
Derece:	1	2	2	3	4	5	6
Olasılık:	0.044	0.088	0.088	0.130	0.173	0.216	0.261
Birikimli Olasılık:	0.044	0.132	0.220	0.350	0.523	0.739	1.000

Adım 5 :Hesaplanan birikimli olasılıklar kullanılarak Rulet Çemberi yöntemiyle 7., 3. ve 6. işlerin birinci yavruda aynı pozisyona kopyalandığı varsayılınsın. Bu durumda birinci yavru aşağıdaki şekilde olacaktır.

1. yavru:	3			7			6
-----------	---	--	--	---	--	--	---

Adım 6 :Birinci yavrudaki boş pozisyonlar ikinci ebeveynden işlerin göreceli

pozisyonları korunarak doldurulur. Bu durumda oluşan birinci yavrunun son hali aşağıdaki şekilde olacaktır.

1. yavru:

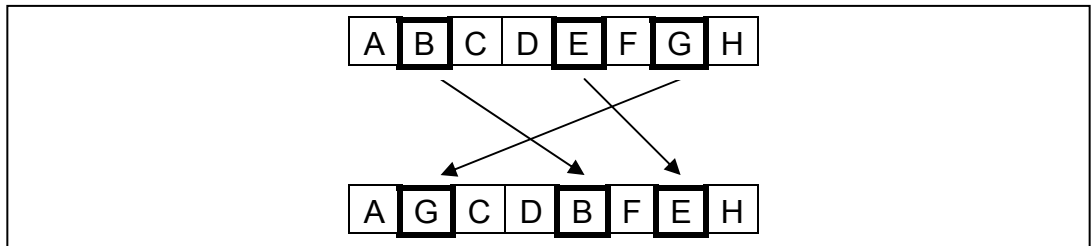
3	5	2	7	1	4	6
---	---	---	---	---	---	---

Adım 7 :Ebeveynlerin rolleri değiştirilip Adım 1'den Adım 6'ya kadar uygulanarak ikinci yavru elde edilir.

3.1.5. Mutasyon Operatörü

Engin⁽¹⁰⁴⁾, maksimum tamamlanma zamanının en küçüklenmesi performans ölçütlü akış tipi çizelgeleme problemleri üzerinde bir çok mutasyon operatörünün performansını karşılaştırmış, bu sınıftaki problemlerde iyi sonuç veren üç operatörün Keyfi Üç İş Değiştirme, Ters Mutasyon ve Araya İş Ekleyerek Değiştirme operatörleri olduğunu tespit etmiştir. Bu çalışma kapsamında önerilen genetik algorithmada Keyfi Üç İş Değiştirme mutasyon operatörü kullanılmıştır.

Keyfi üç iş değiştirme işlemi, rastsal olarak seçilen üç farklı pozisyonda bulunan işlerin rastsal olarak yer değiştirilmesiyle yapılır. Şekil 3.2 bu mutasyon operatörüne ait bir örneği göstermektedir⁽¹⁰⁴⁾.

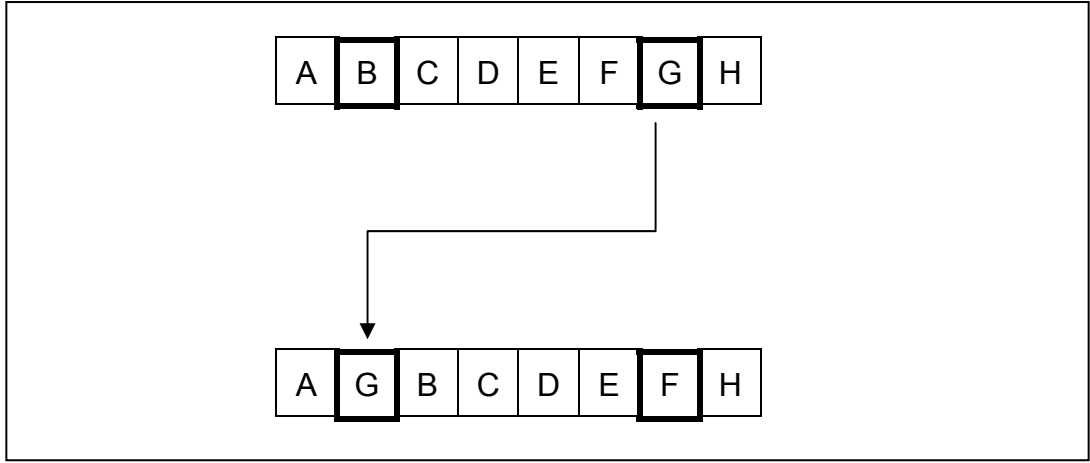


Şekil 3.2 :Keyfi Üç İş Değiştirme Mutasyon Operatörü Çalışma Şekli

3.1.6. Gecikme Tabanlı İyileştirme Algoritması

Genetik algoritmanın başarılı bir şekilde uygulanması, çözülecek probleme iyi bir şekilde adapte olmasına bağlıdır. Aramada kullanılacak çözümlerin tamamının veya bir kısmının kurucu bir sezgiselle elde edilmesi veya yerel arama yöntemlerinin genetik algoritmaya entegre edilmesi adaptasyon yöntemleri içerisinde çekici yaklaşımlardır⁽⁹⁴⁾. Tabu arama, tavlama benzetimi gibi birçok yerel arama metodunun temelini oluşturan iniş algoritması, sadece mevcut çözümün amaç fonksiyonu değerini iyileştiren komşulara hareket edilmesine ve hareket edilen komşunun mevcut çözüm olarak seçilmesine izin verir. İniş algoritmasında böyle bir komşunun bulunamaması algoritmayı sona erdirir⁽⁹⁴⁾. Buna karşın, permütasyon akış tipi çizelgeleme problemlerinde verilen bir çözüme ait komşu çözümlerin tümünün amaç fonksiyonu değerlerinin yani toplam geç bitirme zamanlarının hesaplanması bilgisayar zamanı açısından oldukça külfetlidir. Bu nedenle komşu arama mekanizmasının ürettiği toplam komşu sayısının, sadece ümit vaat eden komşularının incelenmesini sağlayacak şekilde azaltılması gerekmektedir. Daha önce de belirtildiği gibi verilen bir çizelgenin toplam geç bitirme zamanının hesaplanabilmesi için öncelikle çizelgede bulunan tüm işlerin gecikmelerinin (L_j) hesaplanması gerekmektedir. Bir işe ait gecikmenin pozitif olması o işin teslim zamanından sonra tamamlandığını, negatif olması ise teslim tarihinden önce tamamlandığını göstermektedir. Önerilen iyileştirme algoritmasının temel özelliği, iyileştirilecek çizelgedeki işlere ait mutlak gecikme bilgilerini kullanılarak komşu çözüm sayısının azaltılmasını sağlamasıdır. Mevcut çizelgede bulunan işler içerisinde mutlak

gecikmesi en büyük olan iş pozisyonu değiştirilecek iş olarak seçilerek Şekil 3.3'de verilen araya iş ekleme mekanizmasıyla komşular üretilmekte, bu komşular içerisinde en iyi olan komşu mevcut çözümle karşılaştırılmaktadır. Mevcut çözüm iyileşiyorsa seçilen komşu mevcut çözüm olarak alınmakta, iyileşmiyorsa mutlak gecikmesi en büyük olan bir sonraki iş baz alınarak aynı işlem tekrarlanmaktadır. Bütün işlerin incelenmesine rağmen mevcut çözüm iyileştirilemiyorsa, arama işlemi sonlandırılmaktadır.



Şekil 3.3 : Araya İş Ekleme Komşu Üretme Mekanizması

Önerilen Gecikme Tabanlı İyileştirme algoritmasının adımları aşağıda verilmiştir.

Adım 1: Mevcut çözümdeki işlerin mutlak gecikmelerini ($|L_j|$) hesapla ve

işleri mutlak gecikmelerine göre büyükten küçüğe sıralayarak mutlak gecikme sırasını tespit et. $i = 1$ al.

Adım 2: Mutlak gecikme sırasındaki i . işi seç ve Şekil 3.3'de verilen araya iş ekleme mekanizmasına uygun şekilde seçilen işi orijinal pozisyonu

haricindeki tüm pozisyonlara yerleştirerek mevcut çözümün komşularını üret ve bu komşular içerisinde toplam geç bitirme zamanı en küçük olan komşuyu seç.

Adım 3: Seçilen komşu mevcut çözümü iyileştiriyorsa Adım 4'e iyileştirmiyorsa Adım 5'e git.

Adım 4: Seçilen komşuyu mevcut çözüm olarak belirle ve Adım 1'e git.

Adım 5: n problemdeki toplam iş sayısı olmak üzere, eğer $i = n$ ise dur, değilse $i = i + 1$ al ve Adım 2'ye git.

Önerilen algoritmanın uygulanmasına ilişkin bir örnek aşağıda sunulmuştur.

Örnek

5 iş 3 makinalı bir problemin işlem zamanları ve teslim tarihleri aşağıdaki tabloda verildiği gibi olsun. Verilenlere göre, 1-2-3-4-5 çizelgesinin önerilen iyileştirme algoritması kullanılarak iyileştirileceği varsayalım.

İş	Makine			Teslim Tarihi
	1	2	3	
1	10	19	19	165
2	11	2	19	49
3	9	8	17	67
4	16	5	2	83
5	19	13	12	46

Adım 1 :Toplam geç bitirme zamanı 90 olarak hesaplanan 1-2-3-4-5 çizelgesinde bulunan işlerin mutlak gecikme zamanları ve bu zamanlara göre işlerin büyükten küçüğe sırası aşağıdaki tabloda verilmiş olup $i = 1$ alınmıştır.

İş	L_j	$ L_j $	Seçim Sırası
1	-117	117	1
2	18	18	3
3	17	17	4
4	3	3	5
5	52	52	2

Adım 2 : $i = 1$. sırada bulunan 1 işine göre araya iş ekleme mekanizmasıyla elde edilen komşu çözümler ve toplam geç bitirme zamanları aşağıdaki tabloda verilmiştir. Tabloya göre toplam geç bitirme zamanı en küçük olan 2-3-4-5-1 çizelgesi en iyi komşu olarak seçilmiştir.

Komşu Çözüm	T_j
2-1-3-4-5	53
2-3-1-4-5	44
2-3-4-1-5	50
2-3-4-5-1	34

Adım 3 :Seçilen komşunun toplam geç bitirme zamanı mevcut çözümden daha küçük olduğu için ($34 < 90$) Adım 4'e gidilir.

Adım 4 :seçilen 2-3-4-5-1 komşusu mevcut çözüm olarak alınır ve Adım 1'e gidilir.

Algoritmaya uygun şekilde bu işlemlere devam edilmiş ve iyileştirme süreci sonunda toplam geç bitirme zamanı 14 olan 2-5-3-4-1 çizelgesine ulaşılmıştır.

3.1.7. Verilen İki Çizelgenin Benzerlik Oranının Hesaplanması

Önerilen genetik algorithmada verilen iki çizelge arasındaki benzerlik

oranının hesaplanmasında Armentano v.d.⁽⁵⁵⁾ tarafından önerilen aşağıdaki algoritma kullanılmıştır.

Adım 1 : $Benzerlik_katsayısı = 0$ al ve çizelgelerde bulunan her bir pozisyon için aşağıdaki işlemleri uygula.

- ◆ Eğer incelenen pozisyonda bulunan işler her iki çizelgede de aynı ise $Benzerlik_katsayısı = Benzerlik_katsayısı + 2$ al. İşler aynı değil fakat birinci çizelgenin incelenen pozisyonundaki iş diğer çizelgenin incelenen pozisyonunun bir öncesi veya bir sonrasındaki pozisyonda ise

$Benzerlik_katsayısı = Benzerlik_katsayısı + 1$ al.

Adım 2 : n problemde bulunan toplam iş sayısı olmak üzere aşağıdaki formülasyonu kullanarak benzerlik oranını hesapla.

$$Benzerlik_oranı = 100 \times (Benzerlik_katsayısı / (2 \times n)) \quad (3.3)$$

3.2. Geliştirilen Genetik Algoritmada Kullanılan Parametre Değerleri

Daha önce sunulan bilgiler ve yapılan ön çalışmalar sonucunda önerilen GA 'da Çizelge 3.1'de verilen parametre değerlerinin kullanılmasına karar verilmiştir.

Çizelge 3.1 : Önerilen GA İçin Tespit Edilen Uygun Parametre Değerleri

Parametre	Değer
Popülasyon Genişliği	30
Çaprazlama Olasılığı	0.80
Mutasyon Olasılığı	0.40
Çaprazlama Operatörü	Gecikme Tabanlı Çaprazlama Operatörü
Mutasyon Operatörü	Keyfi Üç İş Değiştirme
İyileştirme İterasyon Sayısı (n , problemdeki iş sayısı)	$3n/4$
Benzerlik Eşik Değeri	40

3.3. Geliştirilen Algoritmanın Literatürde Aynı Problem Üzerinde En İyi Performansa Sahip Algoritma İle Karşılaştırılması

Geliştirilen genetik algoritma sezgiseli literatürde en iyi performansa sahip olduğu öne sürülen Hasija v.d.⁽⁷⁰⁾'nin sezgiseliyle karşılaştırılacaktır. Bu sezgisel HTB notasyonu ile gösterilecektir. HTB sezgiseli ve önerilen sezgisel C++ programlama dili kullanılarak kodlanmıştır. HTB sezgiseli tavlama benzetimi tabanlı bir sezgisel olup, Hasija v.d.⁽⁷⁰⁾ tarafından Armentano v.d.⁽⁵⁵⁾'nin tabu arama sezgiseli ve Parthasarty v.d.⁽⁵³⁾'nin tavlama benzetimi sezgiseliyle karşılaştırılmış, HTB sezgiselinin diğer sezgisellerden daha iyi performansa sahip olduğu görülmüştür.

Karşılaştırma işleminde Hasija v.d.⁽⁷⁰⁾'nin çalışmasında olduğu gibi Taillard'ın⁽⁷¹⁾ maksimum tamamlanma zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri için önerdiği karşılaştırma problemlerinden, her bir boyutta 10 farklı problem içermek üzere 5 makina 20 iş, 10 makina 20 iş, 20 makina 20 iş, 5 makina 50 iş, 10 makina 50 iş, 20 makina 50 iş, 5 makina 100 iş, 10 makina 100 iş, 20

makina 100 iş boyutlarında olan 90 farklı problem kullanılmış, bu problemlere ait teslim tarihleri ise yine Hasija v.d. ⁽⁷⁰⁾'nin karşılaştırmalarda kullandığı (3.5) notasyonu ile rastsal olarak üretilmiştir. Karşılaştırma problemleri için (3.5) notasyonu ile rastsal üretilen teslim tarihleri Ek 1'de, (3.5) notasyonunda bulunan u rastsal sayılarının Taillard⁽⁷¹⁾ yöntemiyle üretilmesinde kullanılan başlangıç seed değerleri ise problem bazında Ek 2'de sunulmuştur.

Önerilen genetik algoritma sezgiseli ve HTB sezgiseli olasılıklı yapıda olmaları nedeniyle her bir problem için her bir sezgiselle 3'er deneme yapılmış ve bu denemelerin ortalamaları alınmıştır. HTB sezgiseli, durdurma kriterinin yapısı nedeniyle her hangi bir problem üzerinde yapılan farklı denemelerde birbirine yakın olmasına rağmen çözüm uzayında farklı sayıda çözümü kontrol etmekte yani farklı sayıda amaç fonksiyonu değeri hesaplamaktadır. Bu nedenle, her bir problem üzerinde öncelikle HTB sezgiseli çalıştırılmış ve ortalama amaç fonksiyonu değeri hesaplama sayıları belirlenmiştir. Bu bilgiler önerilen genetik algoritma için durdurma koşulu olarak kullanılmış, yani hem HTB hem de önerilen sezgiselin performansları çözüm uzayında aynı sayıda çözümü kontrol edecek şekilde karşılaştırılmıştır. 20 işe sahip problemlerde elde edilen sonuçlar Çizelge 3.2, 50 işe sahip problemlerde elde edilen sonuçlar Çizelge 3.3 ve 100 işe sahip problemlerde elde edilen sonuçlar Çizelge 3.4'de sunulmuştur. Notasyonların indislerinde kullanılan x için, $x = p$ önerilen sezgisel ve $x = HTB$ Hasija v.d. ⁽⁷⁰⁾'nin tavlama benzetimi sezgiselini ifade etmek üzere, bu tablolarda kullanılan notasyonlar ve açıklamaları aşağıdaki şekildedir;

N_{fe} : İlgili problem boyutu için HTB sezgiselinin çözüme ulaşmak için

gerçekleştirdiği ortalama performans ölçütü değeri hesaplama sayısı,

μ_x : x sezgiselinin ulaştığı performans ölçütü değerlerinin ortalaması,

σ_x : x sezgiselinin ulaştığı performans ölçütü değerlerinin standart sapması,

φ_x : x sezgiselinin ulaştığı performans ölçütü değerlerinin değişim katsayısı ($\varphi_x = (\sigma_x / \mu_x) \times 100$).

V_{\min} : İlgili problem boyutunda ulaşılan en küçük performans ölçütü değ.,

I_x : x sezgiselinin ulaştığı performans ölçütü değerleri ortalamasının ulaşılan en küçük performans ölçütü değerinden görelî yüzde sapması ($I_x = (\mu_x - V_{\min}) \times 100 / V_{\min}$).

Ayrıca, önerilen GA sezgiseli kullanılarak her bir problem için yapılan 3 deneme içerisinde en küçük toplam geç bitirme zamanlarına sahip çizelgeler Ek 3'te verilmiştir.

Yapılan denemeler sonucunda ulaşılan performans ölçütü değerlerinin istatistiksel olarak karşılaştırılması amacıyla eşleştirilmiş-t testi kullanılmıştır.

μ_{I_p} , önerilen genetik algoritma sezgiselinin her bir problem için ulaştığı performans ölçütü değerleri ortalamasının ilgili problem için ulaşılan en küçük performans ölçütü değerinden görelî yüzde sapmaları ortalaması ve $\mu_{I_{HTB}}$, HTB sezgiselinin her bir problem için ulaştığı performans ölçütü değerleri ortalamasının ilgili problem için ulaşılan en küçük performans ölçütü değerinden görelî yüzde sapmaları ortalaması olmak üzere aşağıda verilen H_0 hipotezi H_1 hipotezine karşı eşleştirilmiş-t testi ile test edilmiştir.

Makine Sayısı	N_{fe}	Problem No	V_{min}	Önerilen Genetik Algoritma Sezgiseli				HTB Sezgiseli			
				μ_p	σ_p	φ_p	I_p	μ_{HTB}	σ_{HTB}	φ_{HTB}	I_{HTB}
5	72.299	1	1,286.00	1,286.00	0.00	0.00	0.0000	1,323.67	55.99	4.23	2.9292
		2	3,268.00	3,278.67	11.02	0.34	0.3265	3,525.00	0.00	0.00	7.8641
		3	3,848.00	3,900.67	53.00	1.36	1.3688	4,011.67	47.27	1.18	4.2534
		4	2,516.00	2,516.00	0.00	0.00	0.0000	2,571.33	47.92	1.86	2.1991
		5	1,775.00	1,775.00	0.00	0.00	0.0000	1,784.00	15.59	0.87	0.5070
		6	2,325.00	2,325.00	0.00	0.00	0.0000	2,351.33	44.75	1.90	1.1325
		7	1,504.00	1,509.33	4.62	0.31	0.3544	1,512.00	0.00	0.00	0.5319
		8	2,245.00	2,245.00	0.00	0.00	0.0000	2,264.00	17.06	0.75	0.8463
		9	1,086.00	1,091.67	9.81	0.90	0.5221	1,131.33	24.95	2.21	4.1740
		10	2,368.00	2,384.00	14.73	0.62	0.6757	2,412.67	77.36	3.21	1.8864
10	72.295	1	1,470.00	1,470.00	0.00	0.00	0.0000	1,470.00	0.00	0.00	0.0000
		2	900.00	903.67	6.35	0.70	0.4078	921.67	28.54	3.10	2.4078
		3	1,634.00	1,639.67	6.03	0.37	0.3470	1,692.67	65.86	3.89	3.5906
		4	1,124.00	1,124.33	0.58	0.05	0.0294	1,170.00	64.09	5.48	4.0925
		5	1,570.00	1,579.00	15.59	0.99	0.5732	1,633.00	0.00	0.00	4.0127
		6	2,088.00	2,097.33	15.31	0.73	0.4468	2,152.00	54.84	2.55	3.0651
		7	1,790.00	1,792.33	4.04	0.23	0.1302	1,896.33	19.76	1.04	5.9402
		8	2,045.00	2,053.33	14.43	0.70	0.4073	2,045.00	0.00	0.00	0.0000
		9	1,507.00	1,507.00	0.00	0.00	0.0000	1,588.33	70.44	4.43	5.3968
		10	2,794.00	2,794.00	0.00	0.00	0.0000	2,868.33	71.59	2.50	2.6603
20	72.274	1	962.00	962.00	0.00	0.00	0.0000	962.00	0.00	0.00	0.0000
		2	749.00	749.00	0.00	0.00	0.0000	749.00	0.00	0.00	0.0000
		3	1,057.00	1,057.00	0.00	0.00	0.0000	1,057.00	0.00	0.00	0.0000
		4	1,269.00	1,269.00	0.00	0.00	0.0000	1,269.00	0.00	0.00	0.0000
		5	1,074.00	1,074.00	0.00	0.00	0.0000	1,221.67	125.29	10.26	13.7495
		6	1,248.00	1,248.00	0.00	0.00	0.0000	1,248.00	0.00	0.00	0.0000
		7	416.00	416.00	0.00	0.00	0.0000	416.00	0.00	0.00	0.0000
		8	762.00	762.00	0.00	0.00	0.0000	762.00	0.00	0.00	0.0000
		9	1,243.00	1,243.00	0.00	0.00	0.0000	1,243.00	0.00	0.00	0.0000
		10	1,488.00	1,488.00	0.00	0.00	0.0000	1,580.67	30.02	1.90	6.2278

Çizelge 3.2 : 20 İşli Problemler Üzerinde Yapılan Deneme Sonuçları

Makine Sayısı	N_{fe}	Problem No	V_{min}	Önerilen Genetik Algoritma Sezgiseli				HTB Sezgiseli			
				μ_p	σ_p	φ_p	I_p	μ_{HTB}	σ_{HTB}	φ_{HTB}	I_{HTB}
5	452.604	1	34,347.00	34,871.00	527.53	1.51	1.5256	34,829.67	178.50	0.51	1.4053
		2	37,538.00	37,808.33	405.84	1.07	0.7202	38,467.00	376.44	0.98	2.4748
		3	37,925.00	38,274.00	303.88	0.79	0.9202	38,316.00	538.78	1.41	1.0310
		4	38,438.00	38,841.67	349.63	0.90	1.0502	39,194.00	451.30	1.15	1.9668
		5	40,541.00	40,840.33	284.82	0.70	0.7383	41,313.00	198.03	0.48	1.9042
		6	35,613.00	35,996.67	478.62	1.33	1.0773	36,498.67	341.10	0.93	2.4869
		7	34,480.00	34,736.33	230.35	0.66	0.7434	34,751.67	53.15	0.15	0.7879
		8	33,866.00	34,695.33	307.48	0.89	2.4489	34,267.00	352.68	1.03	1.1841
		9	37,676.00	37,925.33	274.20	0.72	0.6618	38,506.67	758.01	1.97	2.2048
		10	36,988.00	37,687.67	632.53	1.68	1.8916	37,705.67	364.02	0.97	1.9403
10	452.619	1	31,625.00	32,129.00	185.28	0.58	1.5937	32,344.33	1,012.37	3.13	2.2746
		2	23,768.00	24,384.00	538.31	2.21	2.5917	25,334.67	90.91	0.36	6.5915
		3	20,303.00	20,598.00	399.01	1.94	1.4530	20,803.67	201.52	0.97	2.4660
		4	31,812.00	32,211.67	359.86	1.12	1.2563	34,299.33	1,205.83	3.52	7.8188
		5	28,139.00	28,288.00	241.81	0.85	0.5295	29,421.67	400.43	1.36	4.5583
		6	27,816.00	28,389.67	512.42	1.80	2.0624	28,496.67	346.93	1.22	2.4470
		7	31,539.00	31,673.67	211.97	0.67	0.4270	32,141.67	153.58	0.48	1.9109
		8	30,480.00	31,033.00	362.34	1.17	1.8143	30,901.00	365.07	1.18	1.3812
		9	30,977.00	31,412.00	439.05	1.40	1.4043	31,754.33	667.26	2.10	2.5094
		10	27,508.00	28,201.00	73.57	0.26	2.5193	28,630.33	974.98	3.41	4.0800
20	452.557	1	6,904.00	7,079.33	151.93	2.15	2.5395	7,384.00	393.20	5.32	6.9525
		2	5,548.00	5,662.67	158.86	2.81	2.0669	6,482.67	410.72	6.34	16.8470
		3	6,250.00	6,565.33	292.22	4.45	5.0453	7,683.00	360.06	4.69	22.9280
		4	5,529.00	5,578.33	62.78	1.13	0.8922	6,240.00	229.87	3.68	12.8595
		5	7,530.00	7,568.33	39.02	0.52	0.5090	8,338.33	243.53	2.92	10.7348
		6	10,511.00	10,615.67	108.70	1.02	0.9958	11,812.33	339.91	2.88	12.3806
		7	12,487.00	13,277.33	595.54	4.49	6.3292	13,449.67	950.25	7.07	7.7094
		8	8,335.00	8,507.67	206.21	2.42	2.0716	9,383.67	399.23	4.25	12.5815
		9	4,972.00	5,043.33	105.89	2.10	1.4346	5,722.00	131.64	2.30	15.0845
		10	11,852.00	12,225.67	580.14	4.75	3.1528	12,406.67	13.28	0.11	4.6800

Çizelge 3.3 : 50 İşli Problemler Üzerinde Yapılan Deneme Sonuçları

Makine Sayısı	N_{fe}	Problem No	V_{min}	Önerilen Genetik Algoritma Sezgiseli				HTB Sezgiseli			
				μ_p	σ_p	φ_p	I_p	μ_{HTB}	σ_{HTB}	φ_{HTB}	I_{HTB}
5	1.811.272	1	194,037.00	195,759.00	652.60	0.33	0.8875	194,848.00	1,027.88	0.53	0.4180
		2	187,564.00	190,276.33	1,503.35	0.79	1.4461	187,632.33	111.50	0.06	0.0364
		3	184,409.00	189,679.33	1,476.40	0.78	2.8580	185,423.00	1,205.58	0.65	0.5499
		4	176,601.00	177,142.33	230.14	0.13	0.3065	177,592.33	892.25	0.50	0.5613
		5	186,331.00	187,923.00	1,717.99	0.91	0.8544	187,211.67	663.25	0.35	0.4726
		6	180,908.00	181,992.00	1,008.69	0.55	0.5992	183,231.67	1,109.41	0.61	1.2844
		7	185,642.00	185,987.00	299.00	0.16	0.1858	187,815.33	691.75	0.37	1.1707
		8	182,812.00	183,716.00	282.93	0.15	0.4945	183,546.67	812.30	0.44	0.4019
		9	187,758.00	191,883.00	448.50	0.23	2.1970	189,620.33	1,616.34	0.85	0.9919
		10	184,679.00	189,783.33	1,885.49	0.99	2.7639	184,867.33	294.74	0.16	0.1020
10	1.811.203	1	185,492.00	187,252.33	1,687.89	0.90	0.9490	187,834.67	1,206.86	0.64	1.2629
		2	163,203.00	164,918.00	1,629.98	0.99	1.0508	165,647.33	602.79	0.36	1.4977
		3	171,157.00	172,654.67	2,525.93	1.46	0.8750	172,655.67	1,013.58	0.59	0.8756
		4	187,878.00	192,091.00	1,201.74	0.63	2.2424	190,410.67	2,668.56	1.40	1.3480
		5	175,467.00	177,976.33	511.54	0.29	1.4301	176,632.33	1,055.16	0.60	0.6641
		6	159,919.00	163,377.00	926.29	0.57	2.1623	160,507.67	665.00	0.41	0.3681
		7	167,920.00	171,065.33	2,866.89	1.68	1.8731	169,409.33	502.93	0.30	0.8869
		8	179,491.00	180,640.67	1,907.89	1.06	0.6405	182,759.67	1,105.10	0.60	1.8211
		9	177,821.00	182,021.33	856.95	0.47	2.3621	178,303.33	425.21	0.24	0.2712
		10	189,190.00	191,675.67	234.51	0.12	1.3138	190,251.67	1,060.00	0.56	0.5612
20	1.811.234	1	143,908.00	144,612.33	1,130.77	0.78	0.4894	144,034.33	204.27	0.14	0.0878
		2	140,714.00	142,503.67	1,765.04	1.24	1.2718	146,313.33	2,212.17	1.51	3.9792
		3	139,155.00	144,294.33	1,082.26	0.75	3.6932	141,366.67	1,997.10	1.41	1.5894
		4	163,263.00	166,788.00	3,092.05	1.85	2.1591	167,860.67	2,512.77	1.50	2.8161
		5	146,419.00	147,901.67	1,402.84	0.95	1.0126	148,337.33	2,002.36	1.35	1.3102
		6	145,649.00	147,442.33	1,171.41	0.79	1.2313	148,167.67	2,346.48	1.58	1.7293
		7	145,060.00	146,354.33	1,151.94	0.79	0.8923	149,049.67	1,316.73	0.88	2.7504
		8	153,023.00	153,526.33	442.82	0.29	0.3289	153,259.33	359.79	0.23	0.1544
		9	147,856.00	148,775.33	1,119.49	0.75	0.6218	151,005.67	1,210.17	0.80	2.1302
		10	154,255.00	156,170.33	1,902.14	1.22	1.2417	162,707.00	1,658.13	1.02	5.4792

Çizelge 3.4 : 100 İşli Problemler Üzerinde Yapılan Deneme Sonuçları

$$H_0 : \mu_{I_p} = \mu_{HTB}$$

$$H_1 : \mu_{I_p} \neq \mu_{HTB}$$

Test, $\alpha = 0.05$ anlam düzeyinde (Tip-I hata) SPSS 12.0 paket programı kullanılarak yapılmış olup programın çıktısı Şekil 3.4'de verilmiştir. Bu şekilden de görülebileceği gibi $p = 0.000$ değerini almış olup $p = 0.000 < \alpha = 0.05$ olduğu için H_0 hipotezi reddedilir. Sonuç olarak Şekil 3.4'ün "Paired Samples Statistics" bölümünden, önerilen genetik algoritma sezgiselinin HTB sezgiseline kıyasla ulaşılan en küçük performans ölçütü değerlerinden daha küçük sapmalara sahip performans ölçütü değerleri ürettiği istatistiksel olarak gösterilmiş olmaktadır.

T-Test										
Paired Samples Statistics										
		Mean	N	Std. Deviation	Std. Error Mean					
Pair 1	GA	1,094324	90	1,1278791	,1188889					
	HTB	3,235788	90	4,2242463	,4452747					
Paired Samples Correlations										
		N	Correlation	Sig.						
Pair 1	GA & HTB	90	,323	,002						
Paired Samples Test										
		Paired Differences								
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference		t	df	Sig. (2-tailed)	
					Lower	Upper				
Pair 1	GA - HTB	-2,14146	4,0046534	,4221275	-2,9802	-1,3027	-5,073	89	,000	

Şekil 3.4 : Yapılan Eşleştirilmiş-*t* Testinin SPSS 12.0 Paket Programı Çıktısı

4. TARTIŞMA VE SONUÇ

Bu tez çalışması kapsamında, $F / \text{permu} / \sum T_j$ notasyonu ile gösterilen toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri göz önünde bulundurulmuştur. Literatürde bu modele ait küçük boyutlu problemlerin optimum çözümlerinin bulunmasında başarı sağlayan bazı dal-sınır algoritmaları bulunmasına rağmen orta ve büyük boyutlu problemlerin makina sayısı tek olduğunda bile optimum çözümlerini bulabilen etkin bir algoritma geliştirilememiştir. Bu nedenle, literatürde bazı kurucu sezgisellerin yanı sıra iyileştirici sezgisellerden olan tavlama benzetimi ve tabu arama tabanlı sezgiselleri içeren çeşitli çalışmalar yer almaktadır. Buna karşın, göz önünde bulundurulan problemlerin çözümü için diğer bazı çizelgeleme modellerinin yaklaşık çözümlerinin bulunmasında oldukça iyi performans sergileyen Genetik Algoritma yaklaşımını inceleyen her hangi bir çalışma bulunmamaktadır. Bu tez çalışması kapsamında literatürdeki bu boşluğun doldurulması amacıyla göz önünde bulundurulan çizelgeleme modelinin çözümünde Genetik Algoritma yaklaşımı incelenmiştir.

Çalışmanın özünde, incelenen çizelgeleme modelinin çözümü için geliştirilen Genetik Algoritma ve bu algortmada kullanılan Gecikme Tabanlı İyileştirme Algoritması ve Gecikme Tabanlı Çaprazlama Operatörü yer almaktadır. Bu çalışma kapsamında geliştirilen Gecikme Tabanlı İyileştirme Algoritması ve Gecikme Tabanlı Çaprazlama Operatörü göz önünde bulundurulan probleme özgü bilgileri kullanmaktadır.

Önerilen Genetik Algoritma literatürde bulunan ve aynı çizelgeleme modelinde en iyi performansı sergileyen iyileştirme sezgiseliyle karşılaştırılmıştır. Karşılaştırmada literatürde en büyük tamamlanma zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemleri için önerilen karşılaştırma problemleri, yine literatürde önerilen bir yöntemle teslim tarihleri üretilerek kullanılmıştır. Karşılaştırılan sezgisellerin çeşitli boyutlardaki problemler üzerinde ulaştığı performans ölçütü değerleri eşleştirilmiş t testi kullanılarak istatistiksel olarak karşılaştırılmış ve geliştirilen Genetik Algoritma sezgiselinin karşılaştırmada kullanılan diğer sezgiselden daha iyi sonuçlar ürettiği gösterilmiştir.

Önerilen Genetik Algoritma rastsal üretilmiş başlangıç popülasyonundan arama işlemine başlamaktadır. Literatürde önerilen kurucu sezgisellerle elde edilecek çözümleri kısmen veya tamamen içeren başlangıç popülasyonunu kullanacak Genetik Algoritmanın performansı incelenmemiştir. Bunun yanı sıra alternatif seçim mekanizmalarıyla da ilgili bir çalışma yapılabilir. Ayrıca probleme özgü bilgilerin kullanılması yoluyla çözüm uzayının daraltılıp daraltılamayacağı konusunda da her hangi bir çalışma yapılmamıştır. Böyle bir çözüm uzayı daraltma işleminin gerçekleştirilebilmesi önerilen sezgiselin performansını artırmasının yanı sıra Tabu Arama ve Tavlama Benzetimi gibi diğer iyileştirici sezgisellerin de performansını olumlu yönde etkileyebileceği aşikardır. Belirtilen eksikliklerin ileriki çalışmalara ışık tutacağı düşünülmektedir.

KAYNAKLAR

1. M. Pinedo ve X. Chao, Operation scheduling with applications in manufacturing and services, Irwin/McGraw-Hill, New York, 1999.
2. K. Baker, Elements of Sequencing and Scheduling, Tuck School of Business Administration College, Hanover, NH, 1997.
3. S. Sarin ve M. Lefoka, Scheduling Heuristic for the n-Job m-Machine Flow Shop, OMEGA, **21**(2), 229-234, (1993).
4. M. Pinedo, Scheduling: Theory, Algorithms and Systems, Prentice Hall, New Jersey, 1995.
5. S.E. Elmaghraby, The one-machine sequencing problem with delay costs, Journal of Industrial Engineering, **19**, 105-108, (1968).
6. H. Emmons, One-machine sequencing to minimize certain functions of job tardiness, Operations Research, **17**, 701-715, (1969).
7. V. Srinivasan, A Hybrid Algorithm for the one machine sequencing problem to minimize total tardiness, Naval Res. Logist. Quart., **18**, 317-327, (1971).
8. L.J. Wilkerson ve J.D. Irwin, An improved method for scheduling independent tasks, AIIE Transactions, **3**, 239-245, (1971).
9. J. Shwimer, On the n-jobs, one machine, sequence-independent scheduling problem with penalties: A branch-and-bound solution, Management Science, **18**, 301-313, (1972).
10. A.H.G. Rinnooy Kan, B.J. Lageweg ve J.K. Lenstra, Minimizing total costs in one-machine scheduling, Operations Research, **23**, 908-927, (1975).
11. M.L. Fisher, A dual algorithm for the one-machine scheduling problem, Mathematical Programming, **11**, 229-251, (1976).

12. E.L. Lawler, A 'pseudopolynomial' algorithm for sequencing jobs to minimize total tardiness, *Annals of Operations Research*, **1**, 331-342, (1977).
13. K.R. Baker ve L.E. Schrage, Finding an optimal sequence by dynamic programming: An extension to precedence-related tasks, *Operations Research*, **26**, 111-120, (1978).
14. J. Picard ve M. Queyranne, The time-dependent travelling salesman problem and its application to the tardiness problem in one machine scheduling, *Operations Research*, **26**, 86-100, (1978).
15. L. Schrage ve K.R. Baker, Dynamic programming solution of sequencing problems with precedence constraints, *Operations Research*, **26**, 444-449, (1978).
16. K.R. Baker ve J.W. Bertrand, A dynamic priority rule for scheduling against due dates, *Journal of Operations Management*, **3**, 37-42, (1982).
17. C.N. Potts ve L.N. Van Wassenhove, A decomposition algorithm for the single machine total tardiness problem, *Operations Research Letters*, **1**, 177-181, (1982).
18. T. Sen, L.M. Austin ve P. Ghandforoush, An algorithm for the single-machine sequencing problem to minimize total tardiness, *AIIE Transactions*, **15**, 363-366, (1983).
19. C.N. Potts ve L.N. Van Wassenhove, A branch and bound algorithm for the total weighted tardiness problem, *Operations Research*, **33**, 363-377, (1985).
20. P.S. Ow, Focused scheduling in proportional flowshops, *Management Science*, **31**, 852-869, (1985).
21. C.N. Potts ve L.N. Van Wassenhove, Dynamic programming and decomposition approaches for the single machine total tardiness problem, *European Journal of Operational Research*, **32**, 404-414, (1987).

22. R.M.V. Rachamadugu, A note on the weighted tardiness problem, *Operations Research*, **35**, 450-451, (1987).
23. T.D. Fry, L. Vicens, K. MacLeod ve S. Fernandez, A heuristic solution procedure to minimize \bar{T} on a single-machine, *Journal of the Operational Research Society*, **40**, 293-297, (1989).
24. T. Sen, P. Dileepan ve JND Gupta, The two machine flowshop scheduling problem with total tardiness, *Computers & Operations Research*, **16**, 333-340, (1989).
25. J. Du ve J.Y.T. Leung, Minimizing total tardiness on one processor is NP-hard, *Mathematics of Operations Research*, **15**, 483-495, (1990).
26. C.N. Potts ve L.N. Van Wassenhove, Single machine tardiness sequencing heuristics, *IEE Transactions*, **23**, 346-354, (1991).
27. T. Sen ve B.N. Borah, On the single-machine scheduling problem with tardiness penalties, *Journal of Operations Research Society*, **42**, 695-702, (1991).
28. J.E. Holsenback ve R.M. Russell, A heuristic algorithm for sequencing on one machine to minimize total tardiness, *Journal of the Operational Research Society*, **43**, 53-62, (1992).
29. B. Adenso-Diaz, Restricted neighborhood in the tabu search for the flow shop problem, *European Journal of Operational Research*, **62**, 27-37, (1992).
30. M. Widmer ve A. Hertz, Theory and methodology-a new heuristic method for the flow shop sequencing problem, *European Journal of Operational Research*, **41**, 186-193, (1989).
31. Y. D. Kim, A new branch and bound algorithm for minimizing mean tardiness in two machine flowshops, *Computers and Operations Research*, **20**, 391-401, (1993a).
32. S. M. Johnson, Optimal two and three-stage production schedules with set-up times included, *Naval Research Logistics Quarterly*, **1**, 61-68, (1954).

33. Y.D. Kim, Heuristics for flowshop scheduling problems minimizing mean tardiness, *Journal of Operations Research Society*, **44**(1), 19-28, (1993b).
34. S.S. Panwalkar, M.L. Smith ve C.P. Koulamas, A heuristic for the single machine tardiness problem, *European Journal of Operational Research*, **70**, 304-310, (1993).
35. W. Szwarc ve J.J. Liu, Weighted tardiness single machine scheduling with proportional weights, *Management Science*, **39**, 626-632, (1993).
36. A. Fadlalla, J.R. Evans ve M.S. Levy, A greedy heuristic for the mean tardiness sequencing problem, *Computers and Operations Research*, **21**, 329-336, (1994).
37. S. Kondakci, Ö. Kirca ve M. Azizoğlu, An efficient algorithm for the single machine tardiness problem, *International Journal of Production Economics*, **36**(2), 213-219, (1994).
38. S. Chang, Q. Lu, G. Tang ve W. Yu, On decomposition of the total tardiness problem, *Operations Research Letters*, **17**, 221-229, (1995).
39. O. Etiler ve B. Toklu, Çizelgeleme problemlerinde kullanılan genetik çaprazlama operatörlerinin birbirleri ile karşılaştırılması, *Gazi Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, **14**(1), (2001).
40. Y.D. Kim, Minimizing total tardiness in permutation flowshops, *European Journal of Operational Research*, **85**, 541-555, (1995).
41. N. Raman, Minimum tardiness scheduling in flowshops: construction and evaluation of alternative solution approaches, *Journal of Operations Management*, **12**, 131-151, (1995).
42. M. Ben-Daya ve M. Al-Fawzan, A simulated annealing approach for the one machine mean tardiness problem, *European Journal of Operational Research*, **89**, 100-107, (1996).
43. W. Szwarc ve S.K. Mukhopadhyay, Decomposition of the single-machine total tardiness problem, *Operations Research Letters*, **19**, 243-250, (1996).

44. W. Yu, Augmentations of consistent partial orders for the one-machine total tardiness problem, *Discrete Applied Mathematics*, **68**, 189-202, (1996).
45. R.M. Russell ve J.E. Holsenback, Evaluation of leading heuristics for the single machine tardiness problem, *European Journal of Operational Research*, **96**(3), 538-545, (1997).
46. R.M. Russell ve J.E. Holsenback, Evaluation of greedy, myopic and less-greedy heuristics for the single machine, total tardiness problem, *Journal of the Operational Research Society*, **48**(6), 640-646, (1997).
47. B. Alidaee ve S. Gopalan, A note on the equivalence of two heuristics to minimize total tardiness, *European Journal of Operational Research*, **96**, 514-517, (1997).
48. B.C. Tansel ve I. Sabuncuoglu, New insights on the single machine total tardiness problem, *Journal of the Operational Research Society*, **48**, 82-89, (1997).
49. J. Pan ve E. Fan, Two-machine flow-shop scheduling to minimize total tardiness. *International Journal of Systems Science*, **28**, 405-414, (1997).
50. C. Koulamas, Polynomially solvable total tardiness problems:review and extensions, *Omega*, **25**, 235-239, (1997).
51. M.S. Akturk ve M.B. Yildirim, A new lower bounding scheme for the total weighted tardiness problem, *Computers and Operations Research*, **25**, 265-278, (1998).
52. C. Koulamas, A Guaranteed accuracy shifting bottleneck algorithm for the two-machine flowshop total tardiness problem, *Computers and Operations Research*, **25**, 83-89, (1998).
53. S. Parthasarathy ve C. Rajendran, Scheduling to minimize mean tardiness and weighted tardiness in flowshop and flowline-based manufacturing cell, *Computers & Industrial Engineering*, **34**(2), 531-546, (1998).

54. Y. Hirakawa, A quick optimal algorithm for sequencing on one machine to minimize total tardiness, *International Journal Production Economics*, **60**, 549–555, (1999).
55. V. Armentano ve D. P. Ronconi, Tabu search for total tardiness minimization in flowshop scheduling problems, *Computers & Operations Research*, **26**, 219-235, (1999).
56. J.E. Holsenback, R.M. Russell, R.E. Markland ve P.R. Philipoom, An improved heuristic for the single-machine, weighted-tardiness problem, *Omega*, **27**(4), 485-495, (1999).
57. D. Biskup ve W. Piewitt, A note on “An efficient algorithm for the single-machine tardiness problem”, *International Journal of Production Economics*, **66**(3), 287-292, (2000).
58. C. Dimopoulos ve A.M.S. Zalzalá, Investigating the use of genetic programming for a classic one-machine scheduling problem, *Advances in Engineering Software*, **32**(6), 489-498, (2001).
59. W. Szwarc, A. Grosso ve F.D. Croce, Algorithmic paradoxes of the single-machine total tardiness problem, *Journal of Scheduling*, **4**(2), 93-104, (2001).
60. B. Tansel, B.Y. Kara ve I. Sabuncuođlu, An efficient algorithm for the single machine total tardiness problem, *IIE Transactions*, **33**, 661-674, (2001).
61. J.T. Naidu, J.N.D. Gupta ve B. Alidaee, Insights into two solution procedures for the single machine tardiness problem, *Journal of the Operational Research Society*, **53**, 800–806, (2002).
62. J.C. Pan, J.S. Chen ve C.M. Chao, Minimizing tardiness in a two-machine flow-shop, *Computers & Operations Research* **29**, 869-885, (2002).
63. S. Avci, M.S. Akturk ve R.H. Storer, A problem space algorithm for single machine weighted tardiness problems, *IIE Transactions*, **35**, 479–486, (2003).

64. R. Maheswaran ve S.G. Ponnambalam, An investigation on single machine total weighted tardiness scheduling problems, *The International Journal of Advanced Manufacturing Technology*, **22**, 243–248, (2003).
65. J.T. Naidu, A note on a well-known dispatching rule to minimize total tardiness, *Omega*, **31**(2), 137-140, (2003).
66. F.D. Croce, A. Grosso ve V.T. Paschos, Lower bounds on the approximation ratios of leading heuristics for the single-machine total tardiness problem, *Journal of Scheduling*, **7**, 85-91, (2004).
67. P. Babu, L. Peridy ve E. Pinson, A Branch and bound algorithm to minimize total weighted tardiness on a single processor, *Annals of Operations Research*, **129**, 33–46, (2004).
68. A. Grosso, F.D. Croce ve R. Tadei, An enhanced dynasearch neighborhood for the single-machine total weighted tardiness scheduling problem, *Operations Research Letters*, **32**, 68–72, (2004).
69. J.J. Kanet ve X. Li, A Weighted modified due date rule for sequencing to minimize weighted tardiness, *Journal of Scheduling*, **7**, 261–276, (2004).
70. S. Hasija ve C. Rajendran, Scheduling in flowshops to minimize total tardiness of jobs, *International Journal of Production Research*, **42**(11), 2289-2301, (2004).
71. E. Taillard, Benchmarks for basic scheduling problems, *European Journal of Operational Research*, **64**, 278–285, (1993).
72. K. Leksakul ve A. Techanitisawad, An application of the neural network energy function to machine sequencing, *Computational Management Science*, **2**, 309–338, (2005).
73. J. Schaller, Note on minimizing total tardiness in a two-machine flowshop, *Computers & Operations Research*, **32**, 3273–3281, (2005).

74. Z.J. Tian, C.T. Ng ve T.C.E. Cheng, On the single machine total tardiness problem, *European Journal of Operational Research*, **165**, 843–846, (2005).
75. T.C.E. Cheng, C.T. Ng, J.J. Yuan ve Z.H. Liu, Single machine scheduling to minimize total weighted tardiness, *European Journal of Operational Research*, **165**, 423–443, (2005).
76. O. Holthaus ve C. Rajendran, A fast ant-colony algorithm for single-machine scheduling to minimize the sum of weighted tardiness of jobs, *Journal of the Operational Research Society*, **56**, 947–953, (2005).
77. R. Maheswaran, S.G. Ponnambalam ve C. Aravindan, A meta-heuristic approach to single machine scheduling problems, *The International Journal of Advanced Manufacturing Technology*, **25**, 772–776, (2005).
78. R. Maheswaran ve S.G. Ponnambalam, An intensive search evolutionary algorithm for single-machine total-weighted-tardiness scheduling problems, *The International Journal of Advanced Manufacturing Technology*, **26**, 1150–1156, (2005).
79. D. Merkle ve M. Middendorf, On solving permutation scheduling problems with ant colony optimization, *International Journal of Systems Science*, **36**(5), 255–266, (2005).
80. C.S. Chung, J. Flynn ve Ö. Kirca, A branch and bound algorithm to minimize the total tardiness for m-machine permutation flowshop problems, *European Journal of Operational Research*, (2005-**BASIMDA**).
81. W. Szwarc, Some remarks on the decomposition properties of the single machine total tardiness problem, *European Journal of Operational Research*, (2006-**BASIMDA**).
82. Ö. Ergun ve J.B. Orlin, Fast neighborhood search for the single machine total weighted tardiness problem, *Operations Research Letters*, **34**, 41–45, (2006).

83. W. Bozejko, J. Grabowski ve M. Wodecki, Block approach—tabu search algorithm for single machine total weighted tardiness problem, *Computers & Industrial Engineering*, (2006-**BASIMDA**).
84. Ü. Bilge, M. Kurtulan ve F. Kiraç, A tabu search algorithm for the single machine total weighted tardiness problem, *European Journal of Operational Research*, (2006-**BASIMDA**).
85. S.H. Zegordi, K. Itoh ve T. Enkawa, Minimizing makespan for flowshop scheduling by combining simulated annealing with sequencing knowledge. *European Journal of Operational Research*, **85**, 515–531, (1995).
86. C.L. Chen, V.S. Vempati ve N. Aljaber, An application of genetic algorithms for flow shop problems, *European Journal of Operational Research*, **80**, 389-396, (1995).
87. C.R. Reeves, Improving the efficiency of tabu search for machine scheduling problems. *Journal of the Operational Research Society*, **44** (4), 375-382, (1993).
88. R. Ruiz ve C. Maroto, A comprehensive review and evaluation of permutation flowshop heuristics, *European Journal of Operational Research*, **165**, 479-494, (2005).
89. J.A.V. Moccellini, A new heuristic method for the permutation flow shop scheduling problem. *Journal of the Operational Research Society* **46**, 883–886, (1995).
90. V.J. Rayward-Smith, I. H. Osman, C.R. Reeves ve G.D. Smith, *Modern Heuristic Search Methods*, John Wiley And Sons, New York, 1996.
91. A.C. Nearchou, A novel metaheuristic approach for the flow shop scheduling problem, *Engineering Applications of Artificial Intelligence*, **17**, 289–300, (2004).
92. T. Sen, J.M. Sulek ve P. Dileepan, Static scheduling research to minimize weighted and unweighted tardiness: A state-of-the-art survey, *International Journal of Production Economics*, **83**, 1-12, (2003).

93. A. Döyen, Akış Tipi Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemleri İle Çözümü ve Parametre Optimizasyonu, Yüksek Lisans Tezi, Selçuk Üniversitesi-Fen Bilimleri Enstitüsü, Konya, 2004.
94. C.R. Reeves, Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill Book Company, UK, 1995.
95. C. Koulamas, The total tardiness problem:Review and extensions, Operations Research, **42**(6), 1025-1041, (1994).
96. M. Nawaz, E.E. Enscore ve I. Ham, A heuristic algorithm for machine job flow-shop scheduling problem, Omega, **11**, 91-95, (1983).
97. F. Glover, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research **13**, 533-549, (1986).
98. M. Ben-Daya ve M. Al-Fawzan, A tabu search approach for the flow shop scheduling problem, European Journal of Operational Research, **109**, 88-95, (1998).
99. C. Low, J. Yeh ve K. Huang, A robust simulated annealing heuristic for flow shop scheduling problems, Int. The International Journal of Advanced Manufacturing Technology, **23**, 762-767, (2004).
100. D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Massachusetts-USA, 1989.
101. R. Sarker, M. Mohammadian ve X. Yao, Evolutionary Optimization, Kluwer Academic Publishers, NewYork, 2002.
102. M. Gen ve R. Cheng, Genetic Algorithms and Engineering Optimization, John Wiley Sons Inc., USA, 2000.
103. F. Glover ve G. A. Kochenberger, Handbook of Metaheuristics, Kluwer Academic Publishers, NewYork, 2003.
104. O. Engin, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma İle Çözüm Performansının Arttırılmasında Parametre Optimizasyonu, Doktora Tezi, İstanbul Teknik Üniversitesi-Fen Bilimleri Enstitüsü, İstanbul, 2001.

105. N. etin, Genetik Algoritma, Yksek Lisans Tezi, Yıldız Teknik niversitesi-Fen Bilimleri Enstits, İstanbul, 2002.

Ek 1: Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	989	772	523	743	407	357	460	173	935	532
2	1,084	308	271	370	1,380	85	871	439	574	439
3	287	890	634	918	378	694	635	493	602	427
4	764	891	237	333	520	780	680	637	195	882
5	643	937	1,021	1,291	350	431	656	973	425	202
6	1,071	631	413	260	1,279	841	855	264	1,310	1,046
7	721	571	983	676	363	738	918	495	624	475
8	292	842	727	898	590	722	1,109	1,287	880	419
9	478	479	290	434	601	1,044	243	413	1,158	392
10	788	411	449	898	532	1,142	182	593	701	998
11	826	489	778	1,068	837	253	539	312	1,151	251
12	543	381	546	577	513	538	1,006	460	842	642
13	532	1,314	496	554	434	333	924	1,031	291	712
14	654	107	482	319	736	324	555	1,065	862	859
15	761	450	233	1,025	1,014	581	853	378	391	567
16	719	409	548	467	1,193	842	613	1,183	531	661
17	613	583	693	873	282	458	617	746	1,090	386
18	1,368	666	363	231	280	853	779	935	1,014	429
19	635	564	357	1,134	844	363	515	469	911	337
20	827	949	533	902	476	202	518	677	463	854

Tablo 1: 5 Makina 20 İş Problemleri İçin Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	915	2,238	1,650	1,329	1,941	554	498	1,910	1,343	904
2	419	1,243	1,554	694	560	557	2,055	1,003	816	924
3	709	2,017	1,453	1,067	994	1,257	1,653	531	799	804
4	574	462	1,136	503	1,688	1,176	803	748	1,049	854
5	1,301	2,586	543	474	1,311	650	1032	460	751	1,397
6	1,177	1,762	1,168	1,210	1,931	1,036	795	1,753	1,405	725
7	1,090	1,076	1,222	1,427	2,016	1,087	1,333	617	1,775	707
8	2,471	796	1,288	860	512	1,511	1,387	1,433	735	716
9	1,499	1,486	745	1,111	1,097	1,531	1,189	2,131	801	762
10	771	1,020	1,850	895	569	1,303	540	542	1,221	1,723
11	742	1,162	722	1,100	1,102	746	766	952	537	638
12	1,821	879	916	1,612	636	457	800	1,645	1,244	831
13	1,156	1,325	837	730	1,028	762	1,996	917	581	1,061
14	1,786	2,052	1,043	543	564	1,747	1,137	928	1,322	433
15	630	1,053	1,044	1,158	690	476	792	661	800	2,288
16	417	2,396	632	1,101	1,036	1,285	586	613	988	588
17	1,553	771	1,040	1,330	1,021	944	743	929	963	1,982
18	765	1,119	1,179	997	941	1,219	606	1,469	961	1,986
19	1,862	641	793	1,519	680	1,171	465	754	1,968	746
20	691	926	962	849	1,036	1,336	743	600	1,628	1,117

Tablo 2: 10 Makina 20 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	1,065	3,508	2,609	1,304	3,532	2,344	3,413	2,423	3,493	1,962
2	1,362	899	838	1,025	1,383	3,556	4,575	2,290	1,267	1,450
3	1,068	2,748	3,558	921	3,387	3,044	4,077	1,565	1,107	1,793
4	2,289	3,700	962	970	1,664	2,120	1,502	3,163	3,139	996
5	903	1,709	1,466	1,970	1,192	2,811	1,428	1,223	1,485	1,564
6	1,087	3,555	938	1,328	3,524	2,215	2,192	1,512	1,611	2,459
7	2,246	3,716	1,081	3,783	1,329	4,181	3,730	1,080	1,368	1,861
8	1,473	1,753	1,528	1,587	1,384	1,151	3,312	3,300	1,067	2,468
9	3,039	920	1,628	3,618	2,033	2,876	2,316	3,088	2,641	1,021
10	2,137	1,218	2,965	2,020	1,371	1,290	1,528	1,310	2,854	1,217
11	4,389	3,355	3,496	2,839	2,279	3,020	1,145	3,225	4,212	1,830
12	4,220	2,566	1,250	2,809	3,231	4,977	3,466	1,201	3,452	3,581
13	4,201	3,220	2,329	1,502	1,153	1,036	1,055	3,580	985	1,713
14	3,993	1,323	2,968	2,027	1,245	979	927	4,138	1,156	2,023
15	2,968	2,773	2,092	1,375	3,814	2,305	2,252	2,694	2,928	1,267
16	1,099	1,670	3,857	1,650	2,257	966	1,953	1,358	1,625	1,124
17	2,099	2,350	3,022	3,277	3,932	1,198	918	3,111	1,914	2,771
18	1,784	987	3,762	3,508	790	1,056	4,534	1,099	1,819	2,963
19	2,426	1,227	3,425	1,340	3,142	2,623	1,613	1,678	1,317	1,636
20	1,633	1,016	1,040	1,598	3,265	1,052	3,373	3,970	4,179	1,013

Tablo 3: 20 Makina 20 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	284	655	680	1,135	1,180	270	479	860	350	596
2	533	635	737	537	1,252	517	1,005	582	772	1,081
3	505	313	754	971	603	807	812	1,221	492	520
4	952	491	628	861	741	288	610	358	366	1,143
5	593	300	837	1,085	435	501	622	735	934	1,182
6	354	417	255	262	985	577	911	489	336	519
7	526	276	950	352	201	1,437	903	183	252	1,023
8	994	573	250	569	585	745	359	543	388	519
9	892	1,026	652	266	411	743	881	681	584	1,250
10	198	632	915	524	535	373	971	788	225	718
11	897	601	361	748	991	336	1,308	1,220	343	1,153
12	539	1,125	206	349	695	274	818	526	317	478
13	849	771	628	885	311	439	720	533	194	288
14	495	595	313	990	522	532	597	795	374	451
15	1,067	426	429	303	356	1,045	513	708	864	697
16	330	865	227	344	1,028	965	624	692	584	759
17	723	932	298	1,093	821	1,379	892	643	313	1,233
18	409	718	297	899	369	628	531	856	743	331
19	576	1,110	759	1,054	681	807	499	322	731	750
20	692	1,081	570	1,056	358	406	789	365	1,138	325
21	859	352	392	295	626	200	969	577	630	544
22	754	1,162	324	302	830	735	444	723	1,010	549
23	236	511	187	653	744	987	839	525	723	612
24	372	675	773	1,083	578	472	870	544	216	829
25	968	584	442	335	602	700	201	369	725	584
26	435	1,195	627	440	626	929	721	422	958	353
27	1,234	375	465	268	428	552	205	551	414	482
28	1,133	781	756	527	474	642	204	907	254	827
29	1,273	741	716	948	709	583	753	1,190	419	492
30	580	972	597	121	607	515	316	754	972	263
31	451	697	276	673	316	395	446	421	504	512
32	938	832	1,042	419	914	819	958	1059	847	387
33	1,023	531	530	827	354	593	946	919	478	804
34	878	518	551	547	975	763	513	234	344	500
35	567	725	313	480	396	1,041	622	301	885	428
36	435	203	366	955	711	838	265	475	675	540
37	631	831	368	328	447	949	376	597	317	363
38	448	497	976	752	869	825	597	325	459	1,133
39	243	585	559	663	857	686	759	1,098	795	941
40	433	326	387	382	931	257	243	310	349	1,284
41	440	1,082	553	604	880	216	691	489	415	387
42	738	363	345	438	430	949	740	389	650	627
43	762	552	1,069	774	391	527	801	515	406	744
44	467	986	1,028	329	663	857	1,226	413	445	561
45	876	557	378	966	248	1,529	688	1,429	530	860
46	455	858	841	855	673	913	1,029	1,006	184	1,081
47	788	520	403	890	1,034	720	513	584	451	704
48	455	853	565	952	406	679	590	1,200	545	330
49	500	180	475	471	935	784	829	327	388	1,113
50	589	184	1,288	485	137	340	599	576	682	300

Tablo 4 : 5 Makina 50 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	654	1,407	1,205	1,657	705	2,122	1,740	1,717	1,096	1,914
2	955	2,062	1,725	2,189	1,843	2,391	1,055	1,460	1,846	1,562
3	1,213	391	1,668	1,293	1,825	1,332	1,188	1,304	2,013	793
4	1,534	1,746	1,049	1,961	1,485	883	2,194	1,940	933	1,244
5	1,198	1,304	595	490	2,184	746	777	956	609	1,009
6	1,862	1,664	1,299	655	748	1,463	533	1,338	1,760	995
7	589	1,465	956	645	1,295	1,363	1,364	1,017	550	1,207
8	949	1,130	696	958	1,325	1,526	1,162	1,220	1,059	813
9	2,350	1,111	1,965	1,280	970	1,688	1,211	556	1,237	973
10	1,507	1,144	1,662	1,305	510	1,012	2,062	1,931	1,780	1,216
11	2,059	1,346	1,643	1,232	1,091	1,491	1,475	826	2,316	1,084
12	887	1,401	414	2,282	1,325	563	1,615	1,315	765	656
13	1,392	1,205	2,030	1,225	1,246	825	852	749	1,435	926
14	1,787	1,085	532	750	1,848	2,174	763	481	2,089	1,129
15	1,047	704	1,051	919	900	1,227	771	1,701	849	1,263
16	1,328	2,391	1,216	963	1,017	1,358	1,354	1,698	871	1,470
17	968	1,283	710	1,108	2,419	1,539	432	776	846	1,355
18	795	1,934	592	1,742	1,566	2,003	749	1,883	1,547	1,140
19	1,232	1,296	664	473	1,010	1,181	1,379	1,023	1,120	650
20	1,731	1,702	1,786	576	1,687	686	1,241	1,305	1,457	1,422
21	703	2,429	1,678	1,462	616	1,210	1,141	716	1,394	909
22	827	1,153	1,107	504	2,003	1,532	1,059	651	1,286	2,166
23	2,535	936	1,327	597	658	1,269	1,523	1,153	1,526	1,753
24	571	575	984	1,116	2,159	1,039	936	1,447	1,855	1,983
25	489	933	544	706	661	627	689	764	838	2,069
26	1,598	1,256	1,737	2,209	1,306	1,565	658	757	642	1,180
27	984	2,017	467	1,365	612	1,489	779	1,851	2,000	1,087
28	583	1,414	474	1,119	979	1,213	1,244	696	693	881
29	791	2,093	2,378	1,042	684	1,495	699	1,536	468	813
30	1,396	1,911	839	1,502	1,371	2,356	1,948	1,763	439	700
31	663	1,313	883	1,651	1,149	851	1,203	1,557	1,796	1,243
32	865	2,025	1,255	690	887	532	1,574	919	660	1,997
33	631	1,044	1,081	966	867	1,693	1,298	2,123	255	1,112
34	1,493	1,295	2,284	1,696	701	1,148	482	1,594	1,793	1,515
35	1,512	986	1,370	1,489	2,002	1,481	1,850	1,250	1,131	2,739
36	1,727	1,252	1,391	648	865	1,830	687	478	1,459	2,155
37	767	1,697	440	1,481	1,056	658	1,215	1,798	579	1,481
38	2,104	1,629	1,678	687	696	1,190	1,837	880	1,354	1,213
39	920	655	488	1,522	1,622	785	1,425	1,399	1,970	1,568
40	1,999	777	1,435	866	876	595	803	474	1,576	1,788
41	1,092	741	1,839	1,219	1,924	548	700	688	1,929	584
42	1,072	1,200	1,729	900	974	1,471	1,999	868	487	1,088
43	554	1,094	2,285	1,704	1,227	2,197	1,312	1,730	656	1,559
44	1,064	1,071	1,944	584	1,652	1,607	1,458	1,847	924	1,414
45	1,600	527	1,091	1,048	1,796	910	1,121	670	508	1,515
46	672	1,269	586	1,267	1,829	1,131	1,251	2,158	930	2,202
47	1,801	1,099	2,287	2,204	2,186	661	2,535	1,163	647	1,218
48	496	440	1,219	806	830	1,417	1,260	997	1,748	960
49	1,839	1,517	1,191	683	682	1,023	598	665	877	933
50	2,033	629	2,236	605	868	1,989	2,268	686	1,543	1,850

Tablo 5 : 10 Makina 50 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	3,303	2,817	1,623	2,044	3,664	2,342	3,759	2,801	3,634	1,012
2	1,921	3,427	3,354	3,745	3,856	3,013	3,873	2,048	4,418	1,002
3	3,250	1,915	1,699	2,395	2,964	1,468	2,790	4,188	2,830	2,653
4	3,109	3,834	2,595	3,318	1,806	1,149	1,732	3,890	3,003	2,980
5	3,367	1,720	1,337	1,456	2,038	1,261	1,252	2,198	3,507	4,019
6	4,540	2,250	2,570	1,769	1,116	1,652	1,235	1,268	1,437	3,615
7	2,925	2,121	3,305	2,829	1,347	3,160	2,776	2,473	4,125	1,520
8	2,519	904	1,250	1,132	2,036	3,733	3,310	3,516	1,594	1,326
9	2,971	3,731	4,034	3,290	2,142	2,212	3,357	3,654	2,520	3,715
10	1,326	3,042	2,531	2,314	1,402	4,237	1,772	2,141	3,672	1,851
11	1,493	2,868	1,389	3,843	1,369	3,169	1,244	2,953	3,297	1,053
12	1,242	1,500	3,399	3,825	1,668	1,599	3,520	3,030	1,646	1,870
13	3,221	2,452	3,707	2,179	1,599	2,332	2,418	1,882	3,884	1,433
14	3,015	2,605	2,044	1,225	4,624	2,464	1,751	3,186	2,409	3,202
15	3,497	1,722	3,109	3,319	3,366	1,618	3,384	2,243	990	1,141
16	1,685	3,378	2,409	2,992	1,289	2,054	2,418	1,741	2,310	1,601
17	2,850	2,228	2,103	1,134	3,472	2,818	2,315	2,079	2,262	4,072
18	1,845	3,394	2,287	1,257	4,520	2,926	1,314	2,155	3,667	2,547
19	1,294	3,047	3,112	2,378	2,174	2,840	1,471	1,795	1,622	3,592
20	2,271	2,414	4,362	1,252	3,507	2,499	2,710	1,215	1,807	1,702
21	3,466	2,623	1,765	2,330	3,372	2,267	1,236	1,519	1,751	3,248
22	3,617	3,340	1,201	1,801	3,361	1,065	2,369	1,826	4,040	3,396
23	5,058	3,125	1,870	2,962	1,855	1,114	2,476	2,768	3,410	3,061
24	1,095	1,670	2,382	3,820	1,686	1,618	2,870	3,430	3,908	2,063
25	3,105	3,625	2,899	2,290	999	4,005	1,430	1,335	3,719	1,970
26	3,373	3,298	2,427	1,562	1,937	2,349	2,479	2,741	3,636	3,266
27	3,126	2,855	2,093	2,497	1,326	3,189	3,484	2,531	1,873	3,199
28	3,214	2,636	2,440	2,139	1,772	1,938	3,971	1,739	1,881	3,104
29	1,429	3,994	4,142	2,653	3,593	1,742	2,627	1,317	1,300	3,487
30	3,570	1,582	3,438	2,620	1,981	4,217	2,440	2,029	2,233	2,661
31	2,188	3,006	2,826	4,233	1,151	3,237	2,100	3,333	2,257	1,681
32	1,644	2,556	3,817	3,925	1,273	3,487	2,191	2,042	3,530	1,591
33	1,378	1,244	1,632	4,714	1,550	995	1,755	2,861	3,364	1,856
34	3,723	1,573	2,874	3,102	2,986	2,685	2,769	1,647	1,455	3,096
35	4,534	3,927	4,094	3,946	2,968	2,607	1,886	3,865	2,159	3,130
36	3,871	3,193	1,564	2,733	2,070	2,369	3,559	2,195	3,131	1,325
37	2,759	2,479	3,688	1,180	1,324	1,599	1,125	3,304	2,471	1,129
38	1,143	3,320	1,772	1,612	2,973	1,153	2,766	1,635	3,153	1,355
39	2,767	1,595	2,451	2,289	3,087	2,167	2,936	1,353	2,153	2,023
40	3,621	2,092	1,960	2,741	3,488	4,580	2,088	2,748	2,665	3,634
41	1,672	2,058	1,361	3,855	3,798	2,930	2,067	2,729	3,919	1,323
42	3,885	2,594	1,361	2,755	3,328	1,160	3,214	3,326	1,107	1,304
43	1,554	2,369	2,051	3,107	3,017	2,655	1,207	3,433	2,580	3,149
44	3,557	2,324	2,029	929	3,612	1,727	2,298	1,952	1,034	2,638
45	1,526	3,990	1,575	2,468	3,833	3,678	3,286	2,968	1,580	2,269
46	1,285	1,681	2,630	4,079	1,691	3,016	2,815	3,531	1,377	3,904
47	3,705	1,893	3,876	1,846	3,370	2,439	1,453	1,968	2,315	3,683
48	1,615	3,666	3,483	981	3,350	2,301	2,222	3,611	1,367	1,167
49	1,326	960	1,856	4,432	3,055	3,114	1,927	4,473	5,320	2,530
50	2,673	1,982	3,206	1,177	3,711	2,238	3,786	992	1,848	3,333

Tablo 6 : 20 Makina 50 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	400	562	626	738	175	482	403	509	728	962
2	887	1,038	779	705	468	663	581	546	674	106
3	1,017	813	794	478	472	664	547	758	1,021	1,216
4	862	756	974	835	675	112	852	347	271	517
5	497	501	605	560	734	732	577	697	972	659
6	527	491	285	1,045	458	695	223	364	464	534
7	762	689	515	841	607	770	579	410	1,003	492
8	800	677	542	673	463	535	763	432	877	383
9	981	985	516	593	674	725	672	647	594	650
10	241	704	695	982	554	1,058	1,057	1,097	625	390
11	439	801	441	871	366	860	320	791	699	666
12	530	395	246	337	178	871	527	856	959	684
13	967	1,174	208	703	1,103	773	337	460	1,234	292
14	763	569	674	582	461	394	203	466	1,351	1,439
15	764	336	624	1,023	1,092	806	289	576	1,075	754
16	776	344	267	437	753	434	741	896	633	453
17	469	817	628	465	786	744	281	823	736	835
18	1,002	511	619	255	252	953	439	470	928	312
19	259	546	870	752	566	233	743	552	291	659
20	491	201	978	246	301	586	577	708	417	191
21	531	719	844	461	521	602	879	514	227	791
22	1,131	715	254	350	313	1,025	691	553	870	685
23	771	990	211	515	456	845	459	655	726	272
24	361	386	835	425	451	817	353	419	211	984
25	573	446	652	999	797	929	1,302	464	461	824
26	590	840	573	789	655	408	360	611	1,270	439
27	551	1,162	762	533	865	501	616	444	662	361
28	512	950	454	606	230	374	111	1,308	727	954
29	785	783	468	393	808	101	599	470	655	486
30	480	707	638	891	678	602	920	186	1,201	1,511
31	410	425	584	835	772	288	1,233	733	822	473
32	922	1,075	404	490	450	560	600	780	523	515
33	859	327	1,134	683	334	641	827	702	1,168	483
34	504	745	393	1,257	1,117	715	676	570	741	999
35	701	1,130	944	892	996	1,174	234	324	814	425
36	845	1,046	636	1,148	967	261	362	323	1,006	547
37	1,125	879	799	242	747	761	444	1,152	1,159	328
38	1,057	545	377	151	1,131	1,062	896	751	914	710
39	423	326	258	318	308	648	1,254	888	806	1,017
40	983	613	530	902	477	658	369	413	565	939
41	526	1,181	191	1,117	719	998	997	532	1,161	992
42	871	337	284	401	1,368	240	530	118	919	507
43	624	608	297	668	456	777	678	1,040	233	795
44	970	1,335	445	554	492	443	869	832	459	371
45	631	648	388	445	522	984	557	303	768	850
46	615	310	941	621	1,119	425	526	712	1,223	271
47	978	986	429	1,073	409	522	957	622	249	613
48	802	303	738	862	1,063	521	1,106	751	184	496
49	934	826	682	511	435	484	634	274	275	713
50	352	898	529	516	288	689	643	1,153	430	601

Tablo 7 : 5 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
51	465	449	457	525	338	352	510	708	505	301
52	454	498	1,027	219	788	1,089	1086	744	967	1,510
53	959	612	364	1,327	420	341	475	430	419	720
54	355	839	724	575	579	322	360	971	742	570
55	339	1,225	150	290	415	437	181	436	791	504
56	547	871	505	357	890	669	827	154	661	867
57	1,143	524	295	374	352	394	1,024	711	261	990
58	782	752	328	573	410	319	934	938	660	708
59	738	807	701	281	499	550	1,176	297	445	939
60	232	465	174	438	427	804	399	1,233	527	651
61	606	822	837	367	314	237	653	385	757	425
62	1,048	1,173	966	1,022	1,020	1,208	508	969	1,064	393
63	1,052	990	839	927	769	698	1,136	836	249	351
64	528	481	604	544	846	512	573	687	878	799
65	667	480	488	170	523	739	1,047	611	910	235
66	240	453	415	328	994	1,152	753	1,009	329	746
67	546	478	378	460	948	490	648	611	785	696
68	482	763	189	667	521	341	776	701	532	220
69	559	242	603	584	780	326	419	457	516	782
70	586	750	531	363	396	500	679	689	168	543
71	224	884	244	603	893	279	182	377	645	813
72	631	708	1,033	959	500	747	1,015	580	214	878
73	990	582	1,207	494	675	770	359	594	310	920
74	600	387	805	690	144	467	1,252	365	374	139
75	1,076	405	515	472	787	885	719	407	847	356
76	861	660	1,054	514	847	246	789	393	526	908
77	449	463	566	672	372	708	299	535	1,058	1,505
78	1,075	362	1,047	469	615	898	798	948	587	1,411
79	376	257	853	438	237	892	247	1,023	282	362
80	440	507	906	435	743	313	441	241	860	786
81	736	1,122	861	793	613	666	744	1,176	464	890
82	820	553	555	217	663	873	287	412	674	811
83	812	151	463	461	760	196	1,100	577	381	199
84	255	702	1,385	486	887	1,136	317	754	318	337
85	681	794	592	537	323	414	479	864	771	869
86	1,130	719	307	679	397	270	1,017	624	488	1,122
87	446	837	526	776	656	722	731	207	897	599
88	644	413	585	309	421	686	785	569	358	478
89	1,059	1,019	672	664	450	770	396	516	925	642
90	805	455	481	501	430	284	479	365	800	666
91	1,293	862	754	653	779	718	657	719	1,200	685
92	870	387	1,071	812	596	443	905	401	830	386
93	476	520	689	644	837	1,041	684	813	804	659
94	421	536	373	849	308	311	939	508	371	322
95	733	647	689	571	954	177	443	509	282	989
96	1,051	914	792	151	1,055	606	376	471	380	951
97	865	725	643	329	734	900	677	168	818	930
98	441	523	855	595	513	869	433	430	673	596
99	999	185	823	1,148	891	770	730	323	559	865
100	396	276	681	577	475	418	283	823	950	387

Tablo 7 (Devam): 5 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	1,281	1,505	2,053	618	1,336	1,774	1,324	1,363	1,308	1,314
2	979	987	2,120	478	1,034	847	1,796	541	841	505
3	1,291	598	1,201	637	1,310	418	685	1,156	1,061	790
4	763	1,196	949	2,974	1,425	399	1,011	930	703	1,516
5	630	1,981	589	627	1,514	743	490	695	1,319	2,067
6	1,726	808	1,471	554	956	1,089	752	1,979	1,131	1,151
7	2,077	482	2,393	966	1,858	670	1,268	1,867	2,094	1,181
8	808	636	819	1,003	2,284	1,523	1,911	1,678	1,869	1,096
9	2,231	1,466	945	2,053	1,487	974	1,201	1,073	862	1,041
10	2,045	1,051	1,136	1,582	1,402	1,926	1,519	1,035	1,989	1,328
11	1,650	1,164	1,450	689	1,839	1,057	1,872	1,440	1,064	1,864
12	689	1,049	713	796	1,278	1,730	1,231	1,383	640	1,708
13	1,252	1,941	2,032	1,086	606	2,084	935	1,715	1,310	874
14	1,930	768	1,154	509	860	1,188	1,052	518	1,111	414
15	518	482	1,392	1,312	795	1,434	509	1,415	863	1,632
16	624	1,652	1,676	1,236	1,872	2,276	2,011	1,017	1,446	1,032
17	1,201	895	1,563	777	1,500	2,180	700	1,142	592	1,354
18	728	1,315	666	1,249	1,314	652	1,161	1,514	1,527	2,068
19	687	1,690	2,100	614	1,235	1,235	978	1,539	1,164	1,379
20	2,009	1,668	835	628	1,494	375	2,151	1,055	1,792	1,219
21	1,011	408	1,168	1,310	1,746	1,035	707	1,235	1,265	1,717
22	1,334	1,984	1,517	1,468	622	1,035	1,382	945	2,411	1,510
23	1,093	1,277	983	980	2,095	920	1,166	1,189	1,818	2,046
24	903	376	649	716	1,776	1,098	1,309	1,923	733	650
25	1,649	730	1,193	1,191	648	1,388	638	1,956	928	1,059
26	1,527	775	977	674	1,526	1,263	1,407	1,631	1,503	1,583
27	1,051	1,533	781	1,574	2,107	875	1,191	1,316	2,100	836
28	879	966	1,417	658	1,109	522	1,398	1,004	1,628	1,074
29	820	785	851	2,050	1,070	1,144	976	2,251	1,649	551
30	1,691	1,753	594	719	1,468	1,813	2,364	886	1,169	2,004
31	553	1,419	1,226	943	663	1,376	1,917	582	1,159	1,355
32	2,269	615	1,990	1,663	679	804	905	2,347	1,572	609
33	2,012	1,968	2,278	1,291	308	1,297	692	615	1,986	1,391
34	1,691	792	1,236	1,530	1,318	1,014	1,055	932	985	383
35	917	693	1,361	1,317	559	1,686	1,370	2,047	2,027	877
36	653	998	837	2,536	580	1,585	1,507	1,024	947	598
37	1,000	1,969	565	1,865	2,175	1,633	1,869	728	1,458	1,810
38	868	1,181	1,803	886	1,595	920	939	483	1,517	1,234
39	1,637	1,518	464	733	707	1,613	1,275	1,512	1,149	1,190
40	1,117	790	1,918	761	1,689	951	943	640	2,090	677
41	1,131	2,397	476	936	1,603	1,241	957	1,052	718	1,618
42	669	2,219	741	886	1,698	1,664	536	1,987	1,839	771
43	520	1,179	1,711	1,990	1,764	1,514	1,868	1,753	1,194	1,882
44	1,170	752	1,970	2,236	1,697	1,025	1,839	1,582	1,444	1,168
45	1,305	1,943	521	1,060	571	956	582	882	2,129	708
46	966	353	1,529	920	833	665	2,039	1,055	732	1,735
47	1,058	751	1,526	1,310	1,236	564	491	667	1,554	1,065
48	1,351	2,851	1,664	1,171	1,787	1,742	1,060	1,033	1,638	750
49	1,453	621	2,370	2,123	1,228	877	1,190	789	1,002	603
50	1,766	1,877	1,928	1,634	502	858	1,881	877	2,224	2,157

Tablo 8 : 10 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
51	680	910	1,266	1,956	456	1,781	1,150	2,480	1,892	996
52	1,215	1,718	1,580	962	1,297	1,082	377	2,281	1,616	1,335
53	1,669	1,965	1,441	1,558	1,503	1,191	822	1,363	1,745	1,005
54	2,037	1,565	1,526	2,013	2,490	791	598	1,506	1,262	1,027
55	1,696	1,096	964	452	1,121	2,029	585	513	2,159	783
56	1,196	1,368	1,083	735	1,452	635	844	834	2,185	822
57	1,657	1,884	652	1,328	792	1,256	1,212	745	2,024	536
58	1,026	1,783	803	2,034	1,200	509	1,167	2,481	2,208	1,815
59	859	1,694	1,915	770	1,692	1,261	951	842	1,110	1,191
60	1,369	906	832	1,016	782	1,395	2,545	1,925	1,758	1,183
61	1,353	1,079	1,432	829	1,015	2,470	1,346	1,413	1,532	1,049
62	1,316	1,299	2,186	602	1,803	699	582	2,221	999	691
63	976	1,717	1,128	1,905	1,057	861	712	1,222	2,072	1,315
64	616	577	1,254	1,311	2,018	625	1,328	2,281	567	835
65	1,717	460	553	1,059	803	1,155	1,388	964	1,390	964
66	1,123	1,056	1,574	2,545	1,243	1,670	2,075	482	1,716	861
67	1,665	1,463	1,088	1,783	1,083	1,722	306	1,417	1,532	1,187
68	2,521	545	2,187	1,487	2,300	1,484	1,610	578	1,359	795
69	1,609	942	1,722	1,068	1,733	1,124	1,333	664	1,313	1,018
70	522	2,027	2,068	1,856	764	589	1,249	1,013	1,052	1,753
71	1,176	1,357	1,794	1,282	590	2,039	1,052	1,379	1,572	740
72	926	1,354	1,129	723	2,024	864	1,701	999	1,979	1,292
73	1,074	472	1,229	2,424	1,383	2,373	1,047	1,371	1,002	876
74	1,514	1,713	919	565	693	1,455	1,421	1,769	1,543	1,977
75	1,975	1,135	2,009	963	1,316	1,067	1,412	930	1,546	1,078
76	940	714	1,373	978	1,397	539	843	910	617	580
77	384	788	1,898	413	1,386	1,295	528	1,823	2,047	660
78	576	970	1,031	2,149	1,337	992	2,366	1,648	1,866	957
79	1,051	1,402	1,077	1,835	709	1,242	1,292	1,911	1,340	1,394
80	876	462	1,178	1,370	681	1,941	1,595	667	753	912
81	704	1,524	1,472	1,386	815	1,697	853	1,431	563	1,120
82	961	1,279	1,195	1,039	659	538	1,311	1,618	2,255	1,178
83	1,276	1,180	650	1,419	1,090	1,490	985	703	625	1,354
84	573	1,582	2,062	888	1,095	604	646	1,964	1,735	356
85	2,359	688	1,936	509	1,219	1,930	1,105	516	1,130	736
86	1,033	860	1,471	1,941	800	384	1,026	955	1,603	2,118
87	1,360	1,467	1,427	1,126	1,516	1,495	855	1,754	1,388	1,215
88	2,171	879	1,133	873	659	927	655	2,146	864	599
89	1,982	521	1,738	1,389	1,925	2,431	1,579	1,408	1,518	758
90	1,042	1,252	1,306	724	965	1,813	2,118	1,003	1,418	2,167
91	1,716	449	1,517	917	1,188	876	865	1,401	403	1,653
92	1,136	1,212	800	2,102	1,249	621	1,467	672	605	1,210
93	1,529	1,490	2,044	1,649	903	1,374	2,297	874	1,812	1,716
94	1,650	1,739	734	1,075	1,727	1,721	1,886	723	553	1,371
95	711	720	710	883	1,069	1,646	2,106	1,515	1,272	1,472
96	1,892	1,099	1,289	1,216	1,364	1,354	2,203	1,022	840	1,736
97	2,261	2,239	1,043	1,462	2,018	1,548	979	782	697	400
98	586	1,811	1,482	1,547	585	1,122	1,547	1,000	556	1,156
99	1,437	616	978	1,417	1,443	810	1,216	1,357	1,671	1,424
100	978	1,787	1,030	1,508	653	1,919	665	997	1,457	594

Tablo 8 (Devam): 10 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
1	1,966	2,511	2,044	2,507	2,590	3,028	2,724	2,621	2,415	2,303
2	3,847	3,645	1,331	2,626	2,729	2,670	3,094	2,286	1,675	1,362
3	3,286	3,208	2,310	967	1,816	3,356	1,010	1,029	2,507	1,442
4	3,349	2,969	1,119	3,405	2,199	3,476	2,289	1,425	3,383	3,012
5	3,250	3,277	3,906	3,053	2,960	2,603	2,470	1,162	1,722	1,579
6	3,225	2,526	4,005	2,373	3,731	2,973	3,355	1,431	3,486	3,974
7	3,204	2,130	1,674	1,394	2,338	3,440	3,161	3,596	1,518	3,792
8	729	3,087	1,184	1,268	1,783	1,573	3,425	1,032	1,405	2,839
9	1,834	1,852	2,038	2,554	3,214	1,867	3,411	2,811	3,016	1,990
10	2,349	3,830	1,879	1,865	1,807	4,106	2,411	2,731	1,201	3,249
11	1,150	4,144	2,550	3,275	3,232	1,469	1,720	1,698	2,202	1,715
12	3,501	2,430	1,815	1,662	857	2,917	2,512	1,621	2,632	1,907
13	3,066	1,304	3,301	4,109	2,050	1,051	1,369	1,576	3,987	1,230
14	1,164	3,327	2,015	2,019	2,488	2,876	2,441	3,372	2,759	3,004
15	1,587	3,322	3,451	1,865	1,748	1,200	1,942	1,394	3,352	1,588
16	3,240	1,158	3,343	3,883	1,842	1,171	4,397	1,363	3,784	2,162
17	3,895	3,569	2,328	3,135	3,247	3,273	2,759	1,701	2,636	2,928
18	2,018	3,526	2,559	2,177	3,445	2,034	2,180	3,177	3,102	3,410
19	2,467	2,080	1,043	3,485	1,969	1,100	1,877	1,762	1,004	3,308
20	2,518	3,725	2,994	1,476	1,353	3,416	2,701	2,659	4,198	2,792
21	1,117	4,016	3,331	1,873	1,238	3,462	1,427	1,037	1,815	2,019
22	1,478	1,687	3,796	2,510	3,522	2,352	2,347	1,168	2,692	3,109
23	4,324	1,167	2,157	1,780	3,576	1,443	3,721	3,128	3,631	2,054
24	3,495	2,128	2,748	3,939	1,230	3,008	3,498	3,949	2,871	2,440
25	1,603	1,346	3,783	1,005	3,070	3,974	1,626	3,160	3,067	4,593
26	1,504	2,479	1,789	2,928	2,394	1,933	1,204	913	3,820	3,573
27	2,533	1,139	1,351	2,324	1,459	1,620	2,005	3,213	2,131	1,506
28	3,583	3,347	1,206	2,831	3,339	1,882	3,216	1,692	3,248	1,619
29	4,296	3,022	1,355	2,239	4,284	3,661	3,938	2,900	1,590	2,367
30	2,025	1,481	3,289	2,048	3,115	3,651	3,411	3,309	2,706	1,089
31	1,693	1,261	3,240	3,736	2,242	791	2,960	3,407	3,150	1,736
32	1,741	3,705	2,729	2,960	2,042	1,830	1,917	2,926	3,668	3,217
33	3,139	4,695	1,438	1,393	1,425	1,265	2,268	1,718	4,289	3,750
34	1,746	3,656	1,948	2,760	3,059	1,788	1,427	2,394	2,294	2,484
35	3,347	2,452	3,828	1,232	1,491	1,285	2,565	3,637	933	1,200
36	997	3,945	2,968	1,041	1,610	2,122	3,013	2,744	2,381	3,517
37	2,091	2,241	2,637	2,560	3,912	1,415	2,755	1,237	1,076	2,116
38	981	1,045	1,938	1,373	2,051	1,542	3,703	3,402	1,885	3,395
39	3,304	1,014	3,676	2,061	1,020	2,605	2,692	1,960	3,306	2,544
40	2,459	1,650	3,066	1,925	3,095	2,874	3,148	1,829	2,605	4,669
41	1,469	3,492	1,320	1,873	4,605	1,802	1,125	3,256	1,933	2,988
42	2,274	1,271	2,383	4,043	2,997	1,349	1,456	1,006	1,763	2,025
43	2,076	3,429	2,921	3,519	1,426	1,544	1,671	3,191	2,030	1,754
44	2,195	2,700	1,341	3,001	3,185	1,458	1,311	3,159	1,688	914
45	2,644	2,061	2,513	2,985	2,121	1,150	3,309	1,933	1,412	2,623
46	1,658	1,502	2,264	2,299	3,468	4,392	4,042	1,306	1,705	1,023
47	1,024	3,519	1,829	1,845	851	1,380	3,551	2,595	3,570	1,137
48	3,850	2,313	2,049	909	1,111	2,715	2,666	2,653	4,348	2,511
49	3,436	1,117	1,237	1,896	1,215	3,644	2,068	3,135	1,600	2,247
50	1,210	1,912	2,971	4,280	2,095	3,907	1,701	1,358	3,717	1,263

Tablo 9 : 20 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 1 (Devam): Karşılaştırma Problemleri İçin Üretilen Teslim Tarihleri

İş	Problem No									
	1	2	3	4	5	6	7	8	9	10
51	1,033	4,102	1,441	2,529	920	3,638	1,247	2,255	1,731	1,367
52	1,576	3,673	2,346	1,716	2,121	3,403	1,453	3,288	2,144	1,301
53	3,970	3,128	3,962	1,027	4,588	2,003	2,766	2,028	2,192	2,091
54	1,141	2,791	1,815	1,377	2,207	2350	3,642	3,005	2,260	1,617
55	1,642	2,483	2,492	2,017	2,003	3,906	2,463	2,448	2,039	2,332
56	2,680	2,066	3,198	2,274	1,820	3,225	2,998	2,222	1,226	2,730
57	2,457	1,495	3,127	1,839	4,222	2,212	2,487	4,415	2,743	2,979
58	2,969	2,043	3,542	3,450	2,974	1,167	3,872	2,243	1,041	3,543
59	1,044	2,266	1,812	3,136	1,805	3,627	2,113	1,275	3,575	2,537
60	2,235	2,144	3,428	2,512	3,298	1,861	2,213	3311	2,232	1,861
61	1,527	1,491	2,052	1,245	2,344	3,068	2,608	1,619	2,025	1,283
62	1,520	4,035	1,183	3853	2,813	1,429	1,858	1,384	3,251	2,656
63	2,567	2,605	3,273	2,036	1,829	1,399	1,412	2,386	3,594	2,524
64	3,996	2,733	1,315	3,518	2,328	2,256	1,999	1,829	1,612	945
65	2,002	2,666	2,404	1,906	3,452	1,455	3,524	4166	3,790	3,272
66	3,195	842	3,519	1,927	1,339	1,591	1,964	2,036	1,228	2,667
67	2,433	1,744	2,440	1,064	2,958	3,900	3,353	4,491	2,971	2,580
68	1,256	2,535	4,518	3,168	3,855	1,563	2,813	2,269	3,019	3,205
69	1,374	1,070	1,115	1,535	1,438	2,618	1,715	2,050	2,536	2,673
70	2,516	3,960	1,856	1,274	3,002	3,005	3,802	2,426	1,368	3,840
71	3,645	4,509	3,085	1,852	1,244	2,515	2,766	2,913	1,800	1,658
72	1,105	2,327	4,006	1,798	1,795	2,308	2,261	2,110	2,768	1,702
73	969	1,346	2,465	1,578	2,829	968	4,094	2,163	2,655	2,405
74	1,798	4,002	2,413	2,577	1,678	4,125	2783	1,090	1,576	4,166
75	3,292	2,128	1,437	1,148	3,202	3,826	2,402	3,652	1,182	3,545
76	3,107	3,030	3,249	1,226	1,673	3,467	3,121	3,798	3,108	2,677
77	2,104	1,262	1,477	3,082	1,524	1,864	3,475	4,272	1,065	1,698
78	1,808	1,385	2,232	3,921	2,644	3,357	1,690	1,976	3,518	2,316
79	3,369	2,537	2,066	1,215	3,106	1,675	2,228	1,031	3,901	2,351
80	1,653	2,831	3,563	1,719	3,282	2,947	1,054	1,825	1,593	3,042
81	2,497	2,308	3,755	1,665	2,654	4,255	2,989	4,277	1,440	3,002
82	2,465	1,877	3,347	1,901	1,686	1,127	2,038	1,578	2,578	3,028
83	1,143	2,157	1,429	3,165	1,545	1,836	856	4,477	3,504	1,668
84	2,827	1,276	3,245	2,796	2,908	990	1,893	4,365	1,267	2,036
85	1,561	4,119	3,303	2,339	2,730	2,802	3,247	2,900	3,353	937
86	3,320	4,482	1,934	3,041	1,261	2,566	878	3,686	4,452	3,571
87	2,064	1,509	3,205	1,800	3,681	3,518	3,779	1,050	2,699	3,169
88	2,142	3,668	1,475	1,911	3,260	1,317	1,344	3,155	1,838	1,967
89	2,804	3,464	3,749	1,350	1,756	3,254	3,781	2,803	3,069	1,055
90	2,836	2,989	2,390	2,659	2,210	2,664	3,366	3,132	1,300	3,015
91	2,634	2,057	3,435	2,778	1,316	4,121	3,663	3,456	1,922	3,187
92	1,591	1,351	1,690	2,037	3,954	2,778	2,269	2,267	1,570	2,659
93	3,162	1,844	1,974	1,854	3,082	2,412	1,055	4,202	3,832	2,517
94	3,347	2,495	2,376	2,442	2,605	2,078	2,705	3,824	2,898	3,622
95	4,210	3,183	1,649	2,070	2,004	4,672	1,990	3,752	1,450	2,554
96	4,288	1,186	2,809	3,464	3,089	1,208	1,058	1,124	1,286	1,731
97	970	1,437	1,000	2,220	3,944	3,305	3,082	1,447	2,528	2,508
98	1,175	3,509	4,122	3,069	2,132	3,604	2,299	3,722	1,461	1,309
99	3,308	2,318	3,271	2,030	2,280	2,295	1,244	2,507	2,944	2,698
100	4,806	3,579	2,666	2,776	2,090	1,633	2,127	1,583	3,049	2,349

Tablo 9 (Devam): 20 Makina 100 İş Problemleri İçin Teslim Tarihleri

Ek 2: Karşılaştırma Problemleri İçin Üretilen Teslim Tarihlerinde Kullanılan Rastsal Sayıların Başlangıç Çekirdek Değerleri

Problem Boyutu		Problem No									
İş Sayısı	Makina Sayısı	1	2	3	4	5	6	7	8	9	10
20	5	346504868	1477884696	1879216959	692319206	1731115964	940299360	342891009	572427014	820767066	371481563
	10	913749963	1561626094	756798727	1957434793	1167688998	671712641	53792935	2008082301	1280998795	1306630875
	20	1736698224	1953500554	1979271924	179653728	2010971621	925783402	1262860189	1673399771	429689938	2101261661
50	5	474299425	1640149059	677537931	1824591207	827832688	30065947	413393145	1065098264	1178487838	1984907315
	10	503064771	1816027906	341592165	1904923653	136880498	1156207916	1817797325	1700897230	26512813	2124064270
	20	327168648	1391001006	1641278494	1356017220	144491033	1863370961	226526760	832662028	54297051	750800963
100	5	1469312312	394865896	156232975	1267712008	479887078	612906556	457963363	259840067	774789116	931461386
	10	2132055482	710368959	1375451416	412198440	1126516596	718797408	2088956668	123493448	1846011315	1266541451
	20	318774091	2118147224	1815821959	1314593488	1257717040	716006518	631538548	789199248	505402366	1149896317

Ek 3: Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
5	1	1,286	3-15- 8- 9-13-12-17-19-14- 5- 7-11-10-16- 1- 2- 6-20- 4-18
	2	3,268	15-12- 2-16-14-19-10- 7- 9-11-17- 4- 8-18-20- 3- 5- 6- 1-13
	3	3,848	3- 4-15-19-14-16- 9-13-20-18-12- 2-11-17- 6- 8-10- 1- 7- 5
	4	2,516	13-18-16-14- 6- 9- 2-12- 7- 4-17- 8- 3-20-15-11-10- 1- 5-19
	5	1,775	3-12-17- 5- 7- 1- 4-20- 8-10- 9-18-13-19-11-15-14-16- 6- 2
	6	2,325	14- 2-20-11-13-17- 5-12-19- 1- 7- 3- 8-16-18- 6- 9- 4-15-10
	7	1,504	10- 9-20- 1-19-11-14-16-17-18- 5- 4-15- 7- 2-13-12- 8- 3- 6
	8	2,245	1-12- 6- 2- 9- 7- 3-11-17-20- 4-10-18-15-14-13- 5-16-19- 8
	9	1,086	4- 2- 3- 5-15-20-16- 7-10-13-12- 1-14- 8-19-17-11- 9- 6-18
	10	2,368	5-11-19-16-18- 9- 7-15-17-12- 1- 2- 8- 3-20-13- 6-10-14- 4
10	1	1,470	18- 4- 2-15- 3-16-10-11- 1-20- 7- 6- 5-13- 9-17-14-12-19- 8
	2	900	19- 4-17-12- 8-20-15-18-11-10-13- 7- 9- 2- 6- 1- 3-16- 5-14
	3	1,634	4- 7- 9- 5-19-13-16-12-20-17-14-15-18-11- 6- 3- 8- 2- 1-10
	4	1,124	3- 4- 2- 5- 8-13-20-10-18-11-14-16-15- 9- 6-17- 1- 7-19-12
	5	1,570	16- 8-14- 2-20-10-18-15- 3-17-13-12-11- 9-19- 5- 1- 7- 4- 6
	6	2,088	6-15- 2-12-11- 1- 5-13- 4-18- 3- 7-16-17-10-20- 9-19- 8-14
	7	1,790	19- 4-17-10-16-11- 1- 6-15-20- 5-18- 9- 7-12-14- 8-13- 3- 2
	8	2,045	7-10-20-19-15- 4-16-17-14- 3- 5- 2-13- 8-11-18-12- 6- 1- 9
	9	1,507	11-17- 8- 2- 5- 3- 9-18-16-15-13-12-10-14- 4- 1- 6-20- 7-19
	10	2,794	12-16-14-19- 4- 6- 2- 9- 3-13- 7-11- 1-20- 5- 8-10-17-18-15
20	1	962	16- 3- 1- 6- 5- 2-20-18- 8-17-10-19- 7- 4-15- 9-14-11-13-12
	2	749	18-20- 2- 9-19-14-10-16- 5- 8-13-11- 1-15-12- 3-17- 4- 6- 7
	3	1,057	2- 6- 4-20-12- 7- 8- 9- 5-15-14-17-11-13- 1-18-16- 3-10-19
	4	1,269	3- 2- 4- 1- 6- 8-13-20-16-15-19-14- 5-10-12-11-17- 7- 9-18
	5	1,074	18-10- 5-13- 7-14- 8- 4- 2- 9-19-16-11- 3-17-15-12-20- 6- 1
	6	1,248	16-14-13- 8-20-18-10-17- 4- 1-15-19-12- 9- 6- 5-11- 7- 2- 3
	7	416	14-17-13-11- 5- 4-19-10-16-15- 9- 6- 1-20- 8- 7- 3- 2-18-12
	8	762	5- 7-18-12-16- 6- 3-19-10- 1- 9-20- 8-17- 2-13-14-15-11- 4
	9	1,243	13- 2-14-19- 7- 3- 8- 6-16- 5-17-18-10- 4- 1-11-20- 9-15-12
	10	1,488	7- 9-16- 4-20- 5-15- 2-19-10-11- 3-13- 1-14- 8- 6-17-12-18

Tablo 1: Önerilen Genetik Algoritmanın 20 İşe Sahip Problemlere İçin

Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler

Ek 3 (Devam): Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
5	1	34,347	10-31-39- 1- 6-18-38-24-41-44-49-12-20-46-30-36-13-37-50-17-42-19-40-32-11-23-22-35- 3- 7- 2-26- 5- 4-47-48-27-28-43-21-25- 8- 9-29-34-14-33-16-15-45
	2	37,538	49-50- 5-36-42- 3-15- 6-38-33-47-34-23-10-29-18-37-21-28-43-20- 2-46-11-45-13- 4-41-19-14-32- 1-31- 8-44-40-39-22-17-25- 9-16- 7-35-12-30-27-26-48-24
	3	37,925	27- 8-16-23- 2-22-36-15-18-48-11- 6-37-35-20- 3-30-12-49- 4-17-21-26- 1- 5-45- 9- 7-31-34-24-19-46-39-32-25-41-29-10-14-40-38-43-47-50-28-42-44-33-13
	4	38,438	42-26-30-12- 9-22-37-50-25-44-49- 7-31-41-15-16- 2- 4-11-36-23- 6-13-27-47-34-28- 3-43-46-39-24-20-21-35-18-19-14- 1-45-40-38- 5-10-17-33-29-32-48- 8
	5	40,541	46-48- 7- 5-35-27-45-10-28- 3-29-33-21-50-42-19- 9-44-34-30- 8-24-20-4-18-40-39-32- 1-37-17-13-12-43- 6-31-25-36-49-14- 2-15-16-23-22-26-11-38-41-47
	6	35,613	4-21- 1- 5-41-24-31-50-30-25-20-12-11-28-22-48-47-27-46-32-34- 8-33-40-10-43-29-44-16-14-36-26- 9-35- 6-39-37-19-23-18-49- 3-15- 7-17-38- 2-42-13-45
	7	34,480	22-28-27-37-25-40- 8-34- 1-38-19- 5-47-13-42-43-18- 4-15-30-16-39-14-49-21-32-26-20- 2-24-45-12-17-36- 3-41-46- 6-29- 9-10-31-48-35-11-44-23-50- 7-33
	8	34,347	34-40- 7-35-25- 8-31- 4-47-50- 2-24-36-23-37-38-14- 9-33-15-30-21-27-10-17- 1-13-26-12-49-43- 5-19-46-41-18-42-22-20- 6-28-39-16-32- 3-11-48-29-45-44
	9	37,676	46-13-24-17-10-12- 7- 1- 3-29-44-47-16-45-14-40- 8-50-38-49-31-11-15-39-21-34-32-42-28- 9- 4-43-22-41- 2-37-26- 5-36- 6-27-25-23-35-18-20-48-30-33-19
	10	36,988	6-30-50-48-20-27-35-32-36-37- 1-41-21-34-18-42-19-43-15-23-31-33-24-22- 8-26-45-29-11- 7-44-14-12-49-25- 4-39-17- 9-28-10-13-47-46- 3-40- 2-16-38- 5
10	1	31,976	25-43-18- 8-33-37-12-32-17-27-22-31-24-29-28-13-44-41-39- 1-21- 7-46-42-19-10-20-14- 2-48-34-47-45- 4-30- 6-26-16-50-49-36-40-11-35-9- 3-15-38-23- 5
	2	23,768	35- 3-24-50-31-42-46-10-12-40-33-44-47-23-15-22-14- 9-11- 5- 1-26-17-49-43-28-48- 7-34-13-38-36-45- 2- 4-25-41-27-30-32-18-29-16-19-6-37-20-21- 8-39
	3	20,303	24- 4-28-25-12-19-39- 8-46-14-18-27-37-15-49- 5-17-31-16-35- 6-33- 1-32-45-21- 3-11-10-26-22-36- 9- 7-23- 2-30-43-48-44-34-40-41-29-50-13-42-20-47-38
	4	31,812	20-22- 5- 6- 7-23-42-48-19-44-40-33- 8-45-36-17-10-13-27-46-39-11- 9-25-37-29-41-31-21-24-34-30-35-28-15-12-18- 3-50-16- 4-32- 1- 2-14-43-49-38-26-47
	5	28,139	6-10-42-25-31- 1-33-11-36-48-16-21- 9-40-49-23-27- 8-34-15-12-32-30-39- 7-20-18-50-26-19-43- 2-46- 3-22-44-35-41-29-45-47- 4- 5-24-28-38-13-17-37-14

Tablo 2: Önerilen Genetik Algoritmanın 50 İşe Sahip Problemlere İçin

Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler

Ek 3 (Devam): Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
10	6	27,816	3-24-40-25-45-39-37-41-38-47-49-12- 5-23-28-46-20-34- 6-21-42-11-31-44-22-10-19-35-29-48-27-16-15-30- 9- 7-32-33- 2-50-26- 4-17-43-14-13- 1-36- 8-18
	7	31,539	17-26-48-41- 6-49-40- 2-21-13-27- 9-37-20-28-46-44-11-15-18-34-32-16-22-12-23-33-24-31- 1-10-50-42-45-35-38- 4- 8-39-29-36- 3-30-43-25- 7-19- 5-14-47
	8	30,690	26- 9-28-13-11-21-40-36- 5-42- 7-32-17-22-14-45-39-35-38- 3-20-48-31-29-15-41- 1-19-46-43-12- 6-44-37-49- 4-25-23- 8-27-30-34-16-33- 2-47-24-10-18-50
	9	30,977	33-44-42-29-47-45-46-16-35- 5-17-25-30-21-22-38-18-13- 7-20- 1-15-36-32-19- 6-23-50-39-34-27-10-40-24- 2-31-12-37-11-41-14- 8-26- 9-43-28-49- 4-48- 3
	10	28,117	49-11-19-13- 9-29-28- 6-21-41-44-38-27-14- 2-15-20- 8-30-16-34- 7-33-10-42-40- 4-39-45-17-24-43-37-22- 1-48-12-50-23-47-46- 5-32-18-36-26- 3-31-25-35
20	1	6,904	20-12-24-49-38-33-43-45-11-29-41-32-10-18-46-31-48-19-16- 2-37- 8-7-50-39-28-14-27-17- 9-25-13- 4- 3- 5-26-21- 1-15-44-40-22-30-34-36-47-42-35- 6-23
	2	5,548	33-20- 8-49-43-40-34-15-39-12- 5-24- 3-46-47-50-32- 7-41-30- 6-42-44-13-14-17-28- 1-21-37-31-11-19-10-36-27-22-26-38- 2-23-25-18- 9- 4-16-35-45-29-48
	3	6,250	27-24-22-33-11-41- 8-36-16- 3-45-21- 1-23-40-49- 5-14-44-17-43-39-42-18-46-10-26- 6-28-25-34- 4-31-38-15-19-12- 2- 7-50-30-37-32-13-47-48- 9-35-29-20
	4	5,529	20- 5- 8-37-14-44-48-38-50-18-47-30- 6-17-28- 1-22-39-13-10-45-25-21-19-26-29-36-40-27- 3- 7-23-34-43-16-42- 9-15- 4-41-12- 2-24-11-35-32-46-31-49-33
	5	7,530	48- 4-31-32-25-16- 6-28-24-12-37-11-13-33-27-30- 7-46- 5-36-19- 8-26-23-10- 9-43-40-49-34-35-15-39-38- 3-42-21-22-47-17-20-29-44-50-45-1-41- 2-14-18
	6	10,511	37-33-14-50- 3- 5-22-42-24-23- 6-15-13-29-38- 4-16-47- 9-21-48-39-36-44-34-26-35-12-19-17-28-46-18- 2- 1-49-31- 7-41-27-32-11-45- 8-20-43-25-30-10-40
	7	12,639	4-23-11- 5-10- 6-41-25-37-47-33-43-18-32-35-14-44-19-40-49-21-48-16-17-13-31-38-46-30-26-39-34-24-20-29-45-42- 7- 9-15- 8-22-12- 3- 1-36- 2-27-28-50
	8	8,335	39-32-38-20-29- 6-21-28-50-16-44-19-13-22-18- 2-30-47-15-36-25-10-7-26-17-27-34-23- 1- 5-41-11-40-45-33-14-12-43-31-42- 8-37-48-35- 9-24-46- 4- 3-49
	9	4,972	15-28-46-44-29- 8-34-42-21-45- 6-12-50-27-31-30-20-35-39-48-47-16-17-37-14- 9-40-43- 3-19-38- 4-36-32-11-23-33- 5- 1-26-10-18-25-13-41-24-22- 7- 2-49
	10	11,852	33-12- 1- 8-32-48-15- 2-38- 7-16-36-41-20-25-31-24-42-39-10-11-37-49-45-18- 3-30-13-44-23-28-34- 4-35-14-50-21-43-27-26-22- 6-29-40-19-46- 5-17- 9-47

Tablo 2 (Devam): Önerilen Genetik Algoritmanın 50 İşe Sahip Problemlere

İçin Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler

Ek 3 (Devam): Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
5	1	195,235	60-10-39-72-93-56- 5-12-20-46-15-55-65-84-76-59-24-27-16-61-82-19-62-23-50-68-32-53-90-71-92-64-36-40-74- 8-77-13-21-54-99-28-26-30-49-43-86-66-11-79- 1-83-85-80-34-97- 7-18-95-96-45-38-88-44-51-58-3-29-33-35-87-31-52-47-14-25-17-63- 4- 9-98-78-42-89- 6-73-67-70-57-69-41-81-91-37- 2-22-94-48-75-100
	2	188,792	20-69-100-67-66-83-79-80-75-99-33-29-14-95-56-10-46- 1-15-61-37- 9-92-88-16-43-78-55-53-65-98-82-45-17- 8-27-23-35-50- 6-52-26-12-24-25-60-34-19-57-76-58-54-47-51-39-93- 5-86-85-40-31-73-96- 3-41-13-59-62-63-48-72-77-42-68-90- 7-70-84-18-81-94-30-97-91-71-87-89-22-38-36-32-74-44-28-21-49- 2-64-11- 4
	3	188,687	14-41-59-42-94-57-60-68-51-89-75-55-39-23-43-38- 5-25-30-13-11-96-82-10-95-32-17-58-35-44-47-65-70-76-87-64-71- 2-92-26-56-69- 1-22-62-73-28-12-66-34-15-77-85-48-40- 6-90-54-19- 7-36-16-53-97-18-31-9-88-100-61-91-20-98-86-99-33-52-67-46-80-83-93-50-78-45-29-21-79-4-81- 8- 3-37-27-24-49-63-72-74-84
	4	176,898	67-82-20-55-38-56-96-18-79-32-61-28-88-49-94-57-81-42-45-51- 3-93-23-85-52-70-91- 7-83-72-48-95-89-99-21-16-37-26-17-59-22-63-66-40-2-69-54-92- 6-31-50-14- 4-97-80-77-75- 9-71-78-64-43-84-35-73-44-98-65-39-74-27-33-76- 5-86-15-12-13-36-90-47-46- 1-58-10-60-25-62-11-87-34-19-24-30-53-68- 8-100-41-29
	5	186,331	10-12- 1-74-68-11-86-77-57-94-79-24-82-78-50-28-22-98- 2-29-84-58-33-72-67-93-40-44-87-41-51-88-75-23-18-37- 5- 4-48-100-83-89-60-70-80-39-61-91-99-47-32-27-64-15-16-97-92-76-53-17-95-52-26-62- 8-73-55-38-35-63-21-34-25- 7-30-43-54-19-81-20-71-13-56-46-66-59-69-90-6-85-96- 9-65- 3-14-45-31-42-49-36
	6	180,908	61- 8-29-70-20- 4-31-37-57-19-54-27-92-49-76-80-47-83-53-91-51-21-73-75-71-60-39-95-42- 6-14-90-65-85-15-26- 1-88-72-23-68-36-40-69-34- 7-33- 2-97-22-58-99-46- 3-77-98-67-100-78-50-59-81-56-48- 5-82-18-38-28-32-79-17-87-86-93-24-12-55-96-30-74-64-16-89-94-25-11-43-52-44-10- 9-35-62-45-63-13-66-41-84
	7	185,642	79-35-50- 6-19-13-85-28-15-62-14-27- 2-66-91-98-71-84-11-20- 7-76-40- 9-81-26-38-96-92-83-75- 1-64-55- 3- 8-16-89-51-24-46-80-18-90-72- 5-69-12-36- 4-93-17-57-43-70-29-52-82-78-49-60-30-10-99-42-58-41-22-48-54-95-77-73-61-86-33-74-87-68-100-94-97-25-44-67-45-65-63-56-39-31-59-21-53-23-37-34-47-32-88
	8	183,499	26-56-14-42-87-67-86-85-69-53-30- 1-49- 7-36-29-17-54-15- 6-27-25-74-90-75-21-12-31-37-97-50-98-52-95-96-76- 8-35- 2-18-93-82- 9-59-51-66-46-19-65-48-40-68-91-44-13-60-41-33-100-80-32- 3-45-62-99-55-89-64-20-47-34-78-72-70-22-11-61-24-83-43-94- 5-73-79-88-84-39-58- 4-28-77-71-23-16-63-10-57-38-81-92
	9	191,391	57-70- 9- 4-48-72-96-47- 1-24- 6-32-69-63-34-60-61-51-31-83-94-90-28-39-58-16-89-43-44-19-82- 5-84-79-46-80-21-100-75-81-78-49-35-56-95-22-29-10-65-93-54-73-38-86-40-17-23-53-33- 8-11-85-59-50-42-15-55-99-92-68-52-98- 7-67-27-12-20-14-74-45-66-88-18-36- 2- 3-87-71-30-26-97-76-62-91-64-13-41-25-37-77
	10	187,626	2-29-20-50-74-51-62-83-31-100- 7-92-94-93-91-10-42-68-72-23-32-64-9-84-47-28-46-69- 1-54-18-55-75-27- 4-60-15-12-98-76-26-87-90-44-11-45-95-70-22-56-49-80-17-65-59-67-58-37-24-13- 6-33-16-66-57-77-61-63-40-34-43-19-35-53-85-79-97-73-71-25-88-48-78-99-36-81-39-82-41- 5-14-21-30-86- 8-38-96- 3-52-89

Tablo 3: Önerilen Genetik Algoritmanın 100 İşe Sahip Problemlere İçin

Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler

Ek 3 (Devam): Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
10	1	185,492	70-58-61-31-15-77-39-36-60-17-12-43-64-72- 5-79-42-80-83-48-81-29-37-62-100-16-49-44-56-28-21-18- 2-24-14-38-96-20-40-59-87-35-19-26-76-33-94-93-45-30-74-91-99-47-78-73-88- 6-85-57-55-75-82-10-46-51-63- 9- 3-13-84-53-69-50-97-98-71-54-90-66-89-86-65-27-67-11-32-8- 7-52-34-92- 4-41-25-22-95- 1-23-68
	2	163,203	24-12- 3-80- 8-46-66-11-99-65-76- 4-86-73-69-15-49-28-60-81-89-95-23-51-87-72-98- 2-74-21-75-10-17-68-44-40-78-91-42-64-22-77-37-57-7-84-43-55-54-56-71-62- 9-67-85-16-33-61- 1-39-88- 6-58-83-36-27-63-97-14-31-20-26-18-53-38-79-34-52-30-41-32-50-13-70-92-25-82-35-29-90-47-19-100-94-93-48- 5-45-59-96
	3	171,157	45-58-22-83-42-30-94-27-65-96-56-10-39-38- 4-21-23-41-34- 8-63-29-15-12- 3-74-17-54- 9-72- 6-61-57-16-87-25-78-73-99-49-84-52-76-55-51-64-86-66-28-37-32-53-90-91-44-81-14- 5-88-92-18-13-89-24-80- 2-79-35-19-95-97-75-40-62-93-48- 7-11-46-26-43-50-59-20-67-100-69-82-77-70- 1-36-85-31-98-68-33-60-47-71
	4	191,010	77- 5-24-35-19-56-61-93-52-95-22-85-55-12-47-72- 3-46-90-60-42-33-62- 2-82-21-99-26-41-76-39-14-74-30-13-17-23-48-63-88-98-94-40-65-79-71-15- 6-16-64-68-89-31-83-53-97-69-67-28-50-44-29-81- 9-38-59-34-87-18- 7-27- 8-11-86-49-43-32- 1-80-36-37-70-78-96-20-57-92-54-91-10-58-45-75-66-84-51-73-100-25- 4
	5	177,427	79-33-83-65-75-56-57-55-80-74-88-10-50-48-91-36-90-95-78- 6-25-70-73-19-35-71-13-45-28-15-32-82-81-94-66- 7-31-100-87- 8-22-86-20-38-63-77-47-72-84-42-39-52- 9-18-14-40-97-44-46-76-49-53-26-29- 1-92-12- 2-99-34-30-43-21-51-24- 3-93- 5-37-23-64-98-67-17-59-11-69-62-41-68-16- 4-58-61-54-60-89-96-85-27
	6	162,324	78-20-46-64- 9-24-68- 3-79-84-38-21- 7-76-47-86-30-19-92- 5-28-32-22- 4-45- 8-41-58-60-18-36-40-26-14-54-39-43-48-99-52-33-53-59-63-34-35-75-70-44-77-88-74-15-56-71-57-25-82-65-11-49-12- 6-72-37-95-29- 2-80-93-85-91-87-100-42-98-81-23-97-83-66-67-31-55-96-13-10-16-51-50-90-61-94-69-73-27-17- 1-89-62
	7	167,920	56-53-14-67-97-21-52-64-27-13- 4-87-55- 5-48-47-83-99-81-54-77-92-15-86-76-100-38-62-12-59- 8-89-40-74-50-39- 3- 1-71-25-88-35-11-91-7-79-69-32-24-33-42-68-95- 6-70-23-44-34-65-22-80-18-73-96-93-16-9-58-30-37-43-29-26-19-31-10-85-49-28-45-20-61-57-41-46-98-75-63-66- 2-94-90-84-17-78-51-60-72-36-82
	8	179,491	73-78-31-67-70-38-93-98-48-86-41-92-12-77-33-28-87-76-53-14-16- 3-90-72-96-45-15-17-40-56-80-83-44-10-89-55-81-66- 1-91-97- 5-21-11-6-23-84-13-79-64-36-47-94-46-85-19-71-63- 9- 8-69-39- 2-100-22-34-35-88-32-61-50-20- 7-75-51-30-26-24-65-59-74- 4-60-43-99-25-29-68-95-27-54-57-62-42-18-49-37-58-82-52
	9	181,362	87-23-15-95-21-46-61-76-54- 5-64-73- 4-92-25- 2-86-19-32-41-80-43-38-53-79-36-42-11-12-26- 3-60-65-81-75-94-55-70-91-66-68-90-98-67-34-37-63-30-44- 7-97-58-20-17-83-78-18- 9-85-45-50-96-47-56-49-29-62-88- 1-24-52-40-16-84-100-71-59-77-27-28-99-51- 8-89-10-72-74-82-57-69-93-33-48-35-22-13-39-31- 6-14
	10	191,405	63-84-66-97-14-71-76-100- 2-88-99-57-29-17-92-82-54-89-80- 4-21- 7-35-85-68-48-32-34-87-95-73-75-25-52-38-19-78- 8- 3-81-37-41-15-49-42-94- 5-53-10-28-55-67-24-33- 1-74-79-86- 9-20-69-11-83-27-47-56-98-18-70-96-93-44-26-22-62-61-64-31-13-36-51-16-39-72-23-65-77-30-46-91-40-90-59-12- 6-60-43-50-58-45

Tablo 3 (Devam): Önerilen Genetik Algoritmanın 100 İşe Sahip Problemlere

İçin Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler

Ek 3 (Devam): Önerilen Genetik Algoritmanın Ulaştığı En Küçük Toplam Geç Bitirme Zamanına ($\sum T_j$) Sahip Çizelgeler

Makina Sayısı	Problem No	$\sum T_j$	Çizelge
20	1	143,952	22-54-59-31-21-38-11-83-55-61- 1- 8-32-25-81-73-78-36-80-51-89-27-47-92-41-74-98-46-97-82-79-42-69-15-77-37-19-30- 3-60-50-87-88-34-18- 4-39-40-52-65-43-13-44-72-76-28-58-62-85-94- 5-66-45-12-99-90-75-68-67-14-91-17-53-71- 2- 9-84-100-95- 7-86-35-70-24-57-16-23-20-56-49-26-63-96-64-48-93- 6-29-10-33
	2	140,714	50-68-19-97-31-49-96-66-29-46-87-61-60-76-75-82-42-59-58-13-77-93-69- 9-16-72-22-37-55-83-81-99-79- 1-92-38- 5-65-57- 6-54-94-95-34-84-91-25-35-41-14-39-23-89-36-27-28-32-12-64- 4-24-10-73-100-52-62-40-26-44-15-74-11-70-56-45-63-98-67-53-48- 8-90- 3-51-43-80-88-17-20- 2-86-21-85-33-18- 7-78-47-71-30
	3	143,415	64-87-10-51-94-11-75-29-83-28-23- 8-41-70-44-54- 7-88-77-67- 9-56-4-48-33-43-90-37-74-59-86-61-18-95-30-92-32-78-12-20-26-52-65-45-21-66-15-24-96-84-97-80-36-81-69-62-19- 3-55-42-91-31-27-58-100- 2-60-34-63-38-79-17-98-14-39-40-50-93-71-82-46-22-16-72-85-49-76-99-25-89-73-53-68- 1-47-57-13-35- 6- 5
	4	163,263	1-60- 8-79-69-66-88-99-20-73-48-35-87-53-12-33-89-80-61-49-65-84-5-93-47-52-36-57-40-100-67-51-15-26-63-29-14-98-27-39-32-23-25-11-77-34-91-21-95-82-46-56-64-85-37-62-17- 2-30- 9- 3-72-94-45-86-44-59-71-96-43-68- 7-10-81- 4-54- 6-92-55-83-90-75-18-58-70-97-42-78-19-28-22-24-31-41-50-74-13-38-76-16
	5	146,419	64-10-98-51- 5-33-36-69-91-71-77-86-39-12-49-13-52-15-72-27-62-32-16-45-61-90-54-83-48-74-59-99-95-50-19-28-38-20-47-56-58-21-75-60-85-30-42-70-14- 7- 1-78-84-79- 3-55-18-26-24-87-22- 4-93-23-40-37-81-94-67-17-73-97-65-68- 2-44- 9-43-34-41-57-96- 8-25-76-31-89-80-100-11- 6-46-92-63-53-66-29-35-88-82
	6	146,230	56-82-83-19-31-15-63-32-73-47-38-65-96-12-37-69-66-26-77- 9-100-39-45-33-43-60-35-34-11-90-42-64-44-72-79-86-84-18-76-23- 1-93-53-27-13-30-16- 2-58-89- 8-88-48- 7-36-78-24-50-41-22-49-17-67-92-59-99-28-85- 6-61-51-21-75- 4-71- 3-87-29-94-80-62-54- 5-57-20-40-68-25-91-97-70-46-10-55-98-14-95-74-52-81
	7	145,060	57-78-13-93-85-43-80-52-26-96-27-86-64-88-44-60-21-32-66-79-50-100-42-84-99- 3-94-75-49-14-11-22-19-20- 9-95-41-10- 6-67-77-69-24-62-63-37-83- 1-59-15- 4-71-53- 5-92-30-36-28-81-98-51-74-38-35-17-65-68-25-33-12-97-89-45-61-56-72-76- 2-46-47-73-31-34-18- 7-54-55-40-91- 8-82-87-70-58-90-39-23-16-48-29
	8	153,023	39-22-28- 8-59-62- 3- 4-74-87- 6- 5-46-33-17-72-66-40-16-69-97-15-79-73-45-37-12-56-64-68-21-78- 1-52-47-55- 2-92-26-13- 9-90-25-29-99-36-54-51-19-50-71-32-91-86-10-70-100-41-61-11-58-30-63-18- 7-20-42-34-60-14-82-31-27-48-35-24-38-88-89-43-53-75-57-23-95-94-98-67-77-65-83-84-93-76-49-85-96-81-80-44
	9	147,856	43-89-47-90-66-42-95-80-84-19-75-10-88-74-64-35-44-81-29-41-97-38-7-45-76-96-24-21- 1- 9- 2-36-58-34-55-15- 5-59-73-31-37- 4-71-91-22-83-62-57-51-68- 6-60-77-50-17-46-13-52-78-48- 8-53-56-65-26-30-70-49-20-98-72-54-67-99-27-40- 3-87-61-14-23-11-93-32-63-79-12-85-16-92-100-86-28-18-33-25-69-94-82-39
	10	154,255	77-46-83-64-35-96-54-51-11-27-81-85-12-47-100-15-52- 1-34-56-49-67-44-72-24-43-31-71-55-91-98-21-73-88-23-70-89-93-78-53-90-20-39-17-50-28- 8-57-82-45-76-18- 2-75- 7-59-41-26-48-42-33-66-30- 6-25-60-61-65-80-14-95-38-86-63-94-32-97-68-22-92-40-16-58-19-69- 9-84-10-62-74- 4-13- 3-36-79-29- 5-99-37-87

Tablo 3 (Devam): Önerilen Genetik Algoritmanın 100 İşe Sahip Problemlere

İçin Ulaştığı En Küçük Toplam Geç Bitirme Zamanına Sahip Çizelgeler