T.C
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN HİBRİT GENETİK ALGORİTMA İLE ÇÖZÜMÜ VE KARAR AĞACI İLE İNCELENMESİ

Ömer KAYA

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA

2020

Doç. Dr. İbrahim Berkan AYDİLEK'in danışmanlığında, Ömer KAYA'nın hazırladığı **"Akış Tipi Çizelgeleme Problemlerinin Hibrit Genetik Algoritma ile Çözümü ve Karar Ağacı ile İncelenmesi"** konulu bu çalışma 12/03/2020 tarihinde aşağıdaki jüri tarafından oy birliği ile Harran Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir

İmza

Danışman : Doç. Dr. İ. Berkan AYDİLEK ...……

Üye :Dr. Öğr. Üyesi Serkan KAYA .....…….

Üye :Dr. Öğr. Üyesi Abdurrahman ÖZBEYAZ .....…….....

Bu Tezin Elektrik-Elektronik Mühendisliği Anabilim Dalında Yapıldığını ve Enstitümüz Kurallarına Göre Düzenlendiğini Onaylarım.

Doç.Dr. İsmail HİLALİ
Enstitü Müdürü

Bu çalışma TÜBİTAK Tarafından Desteklenmiştir.
Proje No: 118E355

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümler tabidir.

İÇİNDEKİLER

Sayfa No

ÖZET i

ABSTRACT ii

TEŞEKKÜR iii

ŞEKİLLER DİZİNİ iv

ÇİZELGELER DİZİNİ v

SİMGELER DİZİNİ vi

1.GİRİŞ 1

2. ÖNCEKİ ÇALIŞMALAR 5

3. MATERYAL ve YÖNTEM 11

3.1. Veri Seti 11

3.2. Materyal 11

3.2.1. Çizelgeleme 11

3.2.2. Genetik Algoritmalar 26

3.2.3. Veri Madenciliği 35

3.3. Önerilen Yöntem Hibrit-GA 41

3.3.1. Kullanılan Algoritmalar 44

4. ARAŞTIRMA BULGULARI ve TARTIŞMA 47

5. SONUÇLAR ve ÖNERİLER 62

5.1. Sonuçlar 62

5.2. Öneriler 64

KAYNAKLAR 66

ÖZGEÇMİŞ 73

EKLER 74

EK-1 CMAX DEĞERLERİ 74

EK-2 GA-PRO VE HİBRİT-GA ÇALIŞTIRMA SONUÇLARI 82

EK-3 TAILLARD PROBLEMLERİ ÜST DEĞERLERİ 83

EK 4 HİBRİT-GA YÖNTEMİNE AİT EN KÜÇÜK TAMAMLANMA SÜRELERİ ve İŞ SIRALAMALARI 85

ÖZET

Yüksek Lisans Tezi

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN HİBRİT GENETİK ALGORİTMA İLE ÇÖZÜMÜ VE KARAR AĞACI İLE İNCELENMESİ

Ömer KAYA

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. İ. Berkan AYDİLEK
Yıl: 2020 Sayfa:86

Üretim faaliyetlerinde verimlilik, düşük maliyet ve gerçekleşeceği en kısa zaman gibi kriterler göz önünde bulundurularak hazırlanan iş bölümü hesaplamaları çizelgeleme olarak adlandırılmaktadır. Çizelgeleme hesapları tek makineli sistemlerde matematiksel bir model geliştirilerek yapılmaktadır. Makine ve iş sayısı arttıkça teslim süresi, stok maliyeti, üretim verimliliği gibi sorunlar ortaya çıkmaktadır. Günümüzde büyük boyutlu çizelgeleme problemlerinin çözümü için meta sezgisel yöntemlere sıkça başvurulmaktadır. Meta sezgisel yöntemler problemin çözümünü gerek zaman gerekse işlem kolaylığı açısından daha kolay hale getirmektedir. Akış tipi çizelgeleme, makinaların seri şekilde yerleşmesi ve tüm işlerin makinalarda aynı sıra ile işlem görmesi prensibine dayanmaktadır. Bu çalışmada NP-Zor sınıfında yer alan akış tipi çizelgeleme problemlerinin çözümü için hibrit genetik algoritma sunulmuştur. Başlangıç popülasyonuna Palmer, CDS ve Neh yöntemleri adapte edilerek hibrit hale getirilen algoritmanın çözüm kalitesi daha da geliştirilmiştir. Algoritmanın verimliliği literatürde yer alan Taillard problem setleri ile ölçülmüştür. Elde edilen sonuçlar güncel literatür sonuçlarıyla karşılaştırılmıştır. Seçilen on adet problemin üst değerlerini oluşturan sıralamalardan iş süresi, standart sapma, iş sırası ve çarpıklık öz nitelikleri çıkarılıp, bu öz niteliklerle karar ağacı oluşturularak çeşitli kurallar ortaya konulmuştur. Bu kurallar ile şimdiye kadar ulaşılan en iyi sonuçlar arasında benzerlikler aranmıştır. 7 problemde karşılaştırılan çalışmalardan daha iyi sonuçlar elde edilmiştir.

ANAHTAR KELİMELER: Akış Tipi Çizelgeleme, Genetik Algoritma, Karar Ağaçları

ABSTRACT

MSc Thesis

SOLUTION OF FLOW SHOP SCHEDULING PROBLEMS WITH HYBRID GENETIC ALGORITHM AND ANALYSIS WITH DECISION TREE

Ömer KAYA

Harran University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. İ. Berkan AYDİLEK
Year: 2020, Page: 86

The division of labor calculations, which are prepared by considering the criteria such as efficiency, low cost and shortest time in production activities, are called scheduling. Scheduling calculations are made by developing a mathematical model in single machine systems. As the number of machines and work increases, problems such as delivery time, inventory cost and production efficiency arise. Today, meta heuristic methods are frequently used to solve large-scale scheduling problems. Meta heuristic methods make the solution of the problem easier in terms of both time and ease of operation. Flow type scheduling is based on the principle that machines are settled in series and all jobs are processed on machines in the same order. In this study, hybrid genetic algorithm is presented for the solution of flow type scheduling problems in NP-Hard class. The solution quality of the algorithm, which has been hybridized by adapting the Palmer, CDS and Neh methods to the initial population, has been further improved. The efficiency of the algorithm was measured by Taillard problem sets in the literature. The results obtained were compared with the current literature results. Business rules, standard deviation, work order and skewness attributes were extracted from the rankings that constitute the upper values of the selected ten problems, and various rules were introduced by creating a decision tree with these attributes. Similarities were sought between these rules and the best results achieved so far. Better results were obtained from the studies compared in 7 problems.

KEYWORDS: Flow Shop Scheduling, Genetic Algorithm, Decision Trees

TEŞEKKÜR

Tezin konusunun seçiminde, uygulamasında ve çalışmamda yardımlarını esirgemeyen danışmanım Sayın Doç. Dr. İ. Berkan AYDİLEK'e ve tez çalışmam sırasında desteklerini her zaman hissettiğim Aileme teşekkürü bir borç bilirim. Ayrıca bu tezin çalışma konusu, 118E355 proje numarası ile TÜBİTAK tarafından desteklenmiştir. Teşekkür ederiz.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1.1. Veri madenciliği işleyiş modelinin temeli (Ceran, 2006)……3
Şekil 1.2. Veri madenciliği disiplini (Gaber, 1999)……4
Şekil 3.1. Tek makineli sistem……...15
Şekil 3.2. Akış tipi çizelgelemede iki adet seri makine……16
Şekil 3.3. Atölye tipi çizelgeleme probleminin Gantt diyagramı (Şevkli, 2005)……...17
Şekil 3.4. Paralel Makine Modeli (Ceran, 2006)……21
Şekil 3.5. Akış atölyesi yerleşimi (Dağ, 2012)……..23
Şekil 3.6. Bir iş atölyesinde iş akışı (Pinedo, 2009: 24)……25
Şekil 3.7. Açık atölye modeli (Sule, 2008: 367)……26
Şekil 3.8. Genetik algoritma başlıca kodlama türleri (Eroğlu, 2018)……29
Şekil 3.9. Alışveriş sepeti analizi(Han ve Kamber, 2006)……38
Şekil 3.10. Hibrit-GA yöntem algoritması……42
Şekil 4.1. Tai010 hesaplamaları……51
Şekil 4.2. Tai010 karar ağacı……..52
Şekil 4.3. Tai020 hesaplamaları……52
Şekil 4.4. Tai020 karar ağacı……..53
Şekil 4.5. Tai030 hesaplamaları……54
Şekil 4.6. Tai030 karar ağacı……..54
Şekil 4.7. Tai040 hesaplamaları……55
Şekil 4.8. Tai040 karar ağacı……..55
Şekil 4.9. Tai050 hesaplamaları……56
Şekil 4.10. Tai050 karar ağacı……56
Şekil 4.11. Tai060 hesaplamaları……...57
Şekil 4.12. Tai060 karar ağacı……57
Şekil 4.13. Tai070 hesaplamaları……...58
Şekil 4.14. Tai070 karar ağacı……58
Şekil 4.15. Tai080 hesaplamaları……...59
Şekil 4.16. Tai080 karar ağacı……59
Şekil 4.17. Tai090 hesaplamaları……...60
Şekil 4.18. Tai090 karar ağacı……60
Şekil 4.19. Tai0100 hesaplamaları……61
Şekil 4.20. Tai0100 karar ağacı……..61

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 4.1. Palmer yöntemi için sonuçlar…….48
Çizelge 4.2. CDS yöntemi için sonuçlar……49
Çizelge 4.3. NEH yöntemi için sonuçlar……49
Çizelge 4.4. Genetik Algoritma yöntemi için sonuçlar……..50
Çizelge 4.5. Hibrit-GA yöntemi için sonuçlar……...50
Çizelge 5.1. Yöntem sonuçları karşılaştırması……...62
Çizelge 5.2. OGS ortalama karşılaştırması……64

SİMGELER DİZİNİ

Cj	j İşinin Tamamlanma Süresi
Cmax	Maksimum Tamamlanma Süresi
Dj	Teslim Zamanı
Emax	Maksimum Erken Tamamlanma
GA	Genetik Algoritmalar
Jm	Atölye Tipi
Lj	Geç kalma süresi
Lmax	Maksimum Pozitif Geç kalma Süresi
NP	Polinomsal Olmayan
Om	Açık Atölye Tipi
Pij	İşlem Zamanı
Rij	Serbest Zaman
Sjk	Sıraya Bağlı Hazırlık Süresi
Sjo	j İşinden Sonra Temizlik Zamanı
Tj	Pozitif Geç Kalma Süresi
Uj	Geç Teslim Edilen İş Sayısı
VM	Veri Madenciliği
Vij	J İşinin i Makinesindeki Hızı
Wj	İşin Ağırlığı
ΣT	Toplam Gecikme

1.GİRİŞ

Günümüzde endüstride müşterilerin taleplerinin en seri biçimde cevap verebilme, tesise ait kaynakların en verimli, en az miktarda kullanılması zorunluluk haline gelmiştir. Dolayısıyla firmalar gelen taleplere göre kendini ayarlama, ihtiyaca cevap verebilmek için çözüm aramışlardır (Engin, 2001).

Çizelgeleme basit olarak makine, zaman, kapasite kısıtlarının bulunduğu bir problemde işin hangi sırayla yapılacağının planlanması olarak tarif edilebilir. Bu sıralamada çeşitli kriterler göz önünde bulundurulabilir. Bu kriterlerden birkaçı toplam akış zamanı, toplam makinelerin boş bekleme zamanı, toplam geciken iş sayısıdır (Döyen, 2004).

Optimizasyon için seçilen yöntemlerden biri de genetik algoritmalardır çünkü çözümü zor ya da çok zor problemlerde matematiksel modeller yetersiz kalmaktadır. Genetik Algoritmanın avantajı ise belirlenen uygunluk fonksiyonu şartına uyan ihtimallerin hayatta kalması, diğerlerinin de elenmesidir. Böylece giderek istenen amaca yakın çözümler ortaya çıkacaktır (Chen ve ark., 1996).

Bu çalışmada daha önce ortaya atılan çeşitli matematiksel, sezgisel ve metasezgisel yöntemler incelenip, genetik algoritmanın başlangıç popülasyonuna diğer klasik algoritmaların çözümleri dahil edilerek genetik algoritmanın tek başına bulduğu sonuçlardan daha iyi sonuçlar elde edilmesi hedeflenmektedir.

Genetik Algoritma, 1975 yılında John Holland tarafından ortaya atılan bu metotta ancak baskın olanın yaşam hakkı bulunduğu fikriyle yola çıkılan Darwin'in doğal seleksiyon teorisinden esinlenilmiştir (Türkeli, 2010).

1985 yılında David E. Goldberg yazdığı doktora tezi ile doğalgaz hatlarının kontrolü maksadıyla genetik algoritmalardan yararlanılabileceğini doğrulamıştır. (Goldberg, 1989). 1992 yılında ise John Koza genetik algoritma tekniklerinden yararlanarak genetik programlama kavramını ortaya atmıştır (Koza, 1992).

Genetik algoritmaların daha çok; matematiksel olarak herhangi bir yapı kurulamayan, çözüm uzayı çok hacimli, etki eden öznenin sayıca çok sayıda olduğu problemlerin çözüm yolu olarak etkili bir biçimde kullanılmaktadır (Erdem, 2008).

Konuya hâkim isimlerden Piatetsky-Shapiro veri madenciliğini şöyle tarif etmektedir. Verilerin daha önceden belirlenmemiş, gizli, muhtemelen işe yarar bilginin belli bir süreçte çıkartılması işlemi olarak adlandırılmaktadır (Akpınar, 2000).

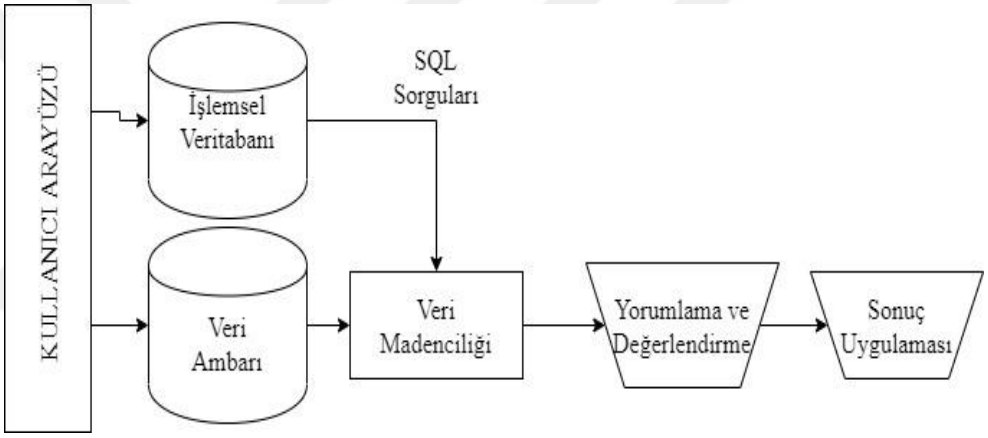
Veri madenciliği, elde bulunan yapısı karmaşık veri içinden, anlam ifade eden ve kullanılabilecek bilgiyi analiz edip arasından çıkarmaya yarayacak işlemleri tahlil edip ve yürütmeye dair yapılan çalışmaların bütünüdür (Vahaplar, 2004).

Veri madenciliği, çok miktarda veri miktarı arasından geleceğe dair öngörü yapılmasını sağlayacak eşitlik ve kuralların algoritmalar vasıtasıyla aranmasıdır (Alpaydın, 2000).

Veri madenciliği, devasa boyutlardaki verinin ayıklanması ve işe yarayacak bilgiye ulaşarak kurumlardaki karar verme süreçlerini iyileştirmektedir (Karahoca, 2004).

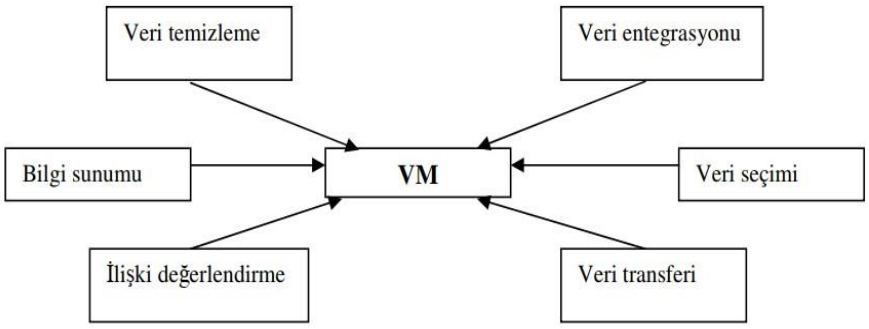
Veri tabanlarındaki anlam ifade eden veriler tespit edilirken öncelikle veri kütleleri içinden işe yarayan veriler alınır, sınıflandırma işlemi yapılır ve nihayetinde işlenir. Sınıflandırma, ilişki tanıma gibi bölümlerde çalışmalar kaydeden veri madenciliği ve makine öğrenmesinde mevcut hastanın o andaki durumunu kaydetme, hastanede işlem yapan tüm hastaların laboratuvar sonuçları, hastalıkla ilgili veriler ve göz, beyin, kalp sinyalleri gibi verilerinin kalitesine göre önceden kestirimde bulunabilme önemli bir sorundur (Bojarczuk, 2004).

Şekil 1.1'de veri madenciliğinin işleyişinin basit hali gösterilmektedir.



Şekil 1.1. Veri madenciliği işleyiş modelinin temeli (Ceran, 2006)

Şekil 1.2'de veri madenciliği düzeni gösterilmiştir.



Şekil 1.2. Veri madenciliği disiplini (Gaber, 1999)

2. ÖNCEKİ ÇALIŞMALAR

Akış tipi problemlerle ilgili yapılan çalışmaların bazıları şunlardır:

Akış tipi çizelgeleme için gerek akış zamanı gerekse tamamlanma süresini minimize etmenin NP yani polinomal olmayan zor sınıfa giren bir problem olduğu Garey ve ark. (1976) tarafından ortaya atılmıştır. Bu sebeplerden dolayı optimum sonucu aramak yerine sezgisel ve metasezgisel yöntemlerle belli bir çözüm uzayında ve belli sürelerde, daha iyi çözümlere odaklanılmıştır. Tamamlanma süresini minimize amacıyla Palmer (1965), Campbell ve ark. (1970) tarafından CDS; Nawaz ve ark. (1983) tarafından NEH sezgisel metotları önerilmiştir. Tez çalışmamızda Palmer, CDS ve NEH metotları ile elde edilen çözümler genetik algoritmanın başlangıç popülasyonunda kullanılarak hibrit bir model önerilmiştir.

Santos ve ark. (1995), Cmax yani maksimum tamamlama süresini minimize etmek amaçlı akış tipi problem için global alt sınırlar ortaya koymuştur. En uygun çözüm olmadığı durumlarda sezgisel yöntemlerin kalitesini değerlendirmek amaçlı önerilen global alt sınırlardan yararlanılabileceğini göstermişlerdir.

Liu ve Reeves (2001) yerel arama metotlarını da dahil eden çok sayıda yapıcı sezgiseli hibrit eden bir sezgisel önermiştir. Elde ettikleri bu yapıcı sezgiselin önceki metotlara göre daha iyi sonuç verdiğini kanıtlamışlardır.

Allahverdi ve Aldowaisan (2002) ise ikili yer değiştirmeler yapılarak performansın artırılabileceğini ortaya koymuştur.

Framinan ve ark. (2005) genel akış tipi çizelgeleme problemleri için var olan metasezgisel yöntemlere genel bir bakış sunmuş ve ek olarak diğer sezgisellere göre daha iyi performans sonuçları veren metasezgisel önermişlerdir. Çalıştığı toplam akış tipi çizelgeleme problemi için metasezgisel yöntemlerin önceki sezgisellere nazaran daha iyi sonuçlar verdiğini gözlemlemiştir. Genel olarak metasezgiseller sezgisellerden sonuçlarda daha iyi olduğunu fakat daha fazla zaman aldığını belirtmiştir.

Rajendran ve Ziegler (2004) toplam akış zamanı ve tamamlanma süresini minimize için M-MMAS and PACO ismini verdikleri iki karınca koloni optimizasyon metodu ortaya koymuşlardır. Metotların performansı Liu ve Reeves (2001) algoritması ile karşılaştırmış ve Taillard'ın kıyaslama problemlerine göre çok daha iyi sonuçlar elde edilmiştir. Rajendran ve Ziegler (2005) iki karınca kolonisi yöntemi oluşturarak daha verimli çözümlere ulaşmışlardır.

Döyen (2004), genetik algoritmanın performans geliştirilmesinde, öğrenen Genetik Algoritma modellerini önermiş ve bu modeli diğer sezgisel yöntemler (NEH, CDS) ile karşılaştırıp daha etkin olduğunu belirlemiştir. Tez çalışmamızda ise Palmer, CDS ve NEH algoritmaları genetik algoritmanın başlangıç popülasyonunda kullanılarak hibrit bir model önerilmiştir.

Oğuz C. ve ark. (2004), ortaya koydukları çok işlemli hibrit akış problemleri için bir sezgisel kullanmışlar ve tabu arama yöntemiyle geliştirmişlerdir.

Tang ve ark. (2006), n iş s aşamaya sahip esnek akış tipi çizelgeleme üzerine çalışmış, Cmax'ı minimize etmek amacıyla tam sayılı programlama algoritması, Lagrangian gevşetmesi ile dinamik programlamayı çalışmasında kullanıp bir çizelgeleme oluşturmuşlardır.

Tasgetiren ve ark. (2007) parçacık sürü optimizasyon algoritması önermiş ve Liu ve Reeves (2001), ve Rajendran ve Ziegler (2005) tarafından geliştirilen algoritmalara göre çok daha iyi sonuçlar elde etmişlerdir.

Dağ (2012) çok amaçlı akış tipi problemler için sezgisel yöntemlerle optimizasyonu önermiş, kablo üreten bir endüstri tesisinin verilerinden yararlanmıştır. Geliştirdiği HAMC algoritması ve karınca koloni algoritmasıyla problemleri çözüp diğer metasezgisel yöntemlerle karşılaştırmıştır.

Şahin (2014) akış tipi problemlere paralel doyumsuz algoritmayla çözüm önermiş, Ancâu (2012)'nin önerdiği Yapıcı Doyumsuz Algoritma ve Sündüz (2012) tarafından önerilen Çok Amaçlı Karınca Koloni Sistemiyle karşılaştırıldığında daha iyi sonuçlara ulaşmıştır.

Veri madenciliği tekniklerinin çizelgeleme alanında optimizasyon amaçlı kullanıldığı çalışmaların bazıları şunlardır:

Koonce ve ark. (1997) sıralama problemleri içinden atölye tipi çizelgeleme problemi üzerinde uygulamışlardır. Yapılan çalışmada, genetik algoritma kullanılarak optimum iş sıraları bulunmuş, süreç ve iş yükü dikkate alınarak birtakım kurallar belirlenmiştir. Sonucun tartışıldığı kısımda veri madenciliği yönteminin kompleks sistemlerin anlaşılmasına yardımcı olarak kullanılabileceğine vurgu yapılmıştır.

Koonce ve Tsai (2000) çalışmalarında oldukça fazla sayıda bulunan atölye tipi çizelgeleme probleminden anlamlı bilgi çıkarımı için veri madenciliği tekniklerini kullanmıştır. Yapılan çalışmada, atölye tipi iş sıralama problem verisi olarak Muth ve Thompson (1963) çalışmasında kullandıkları problemlerden yararlanılmıştır.

Santos ve ark. (2006) çalışmalarında araçların hedef rotalama problemi gibi daha önceden bilinen bir problem üzerine yoğunlaşmış, işlem performansı artırmak için birtakım alternatifler sunmuşlardır. Evrimsel algoritma, yerel arama ile evrimsel algoritma, veri madenciliği kısmını içeren evrimsel algoritma ve yerel arama ile veri madenciliğini barındıran evrimsel algoritma olmak üzere dört adet algoritma oluşturmuşlardır. Veri madenciliği ve yerel arama çözüm verimini artırmak maksatlı kullanılmıştır. Çalışma sonucunda elde edilen veriler, veri madenciliği ve yerel arama modüllü evrimsel algoritmanın daha iyi sonuçlar içerdiğini ortaya koymuştur.

Yang ve ark. (2009) hastanelerin acil kısmındaki nöbet işgücünün planlamasına dair bir çalışma yapmışlardır. Veri madenciliğinde sıklıkla kullanılan karar ağacı yöntemiyle gün içindeki iş yoğunluğu tahmininde kullanmışlardır. Yapı kurulup veriler analiz edilmesinin hemen ardından işgücü sayısı tespit edilmiş ve belirlenmiş zaman aralıklarına göre işgücü planlanmıştır.

Ceran (2006) yaptığı çalışmada, maksimum bitme süresinin minimize etmeye yönelik akış tipi çizelgeleme problemlerinin daha verimli çözümü için genetik algoritma ile beraber veri madenciliğini kullanmıştır. Veri madenciliğinde kullanılacak gerekli bilgiyi elde etmek için genetik algoritma yöntemi uygulanarak ulaşılan iş sıraları ve maksimum tamamlama değerleri kullanılmıştır. İşler üç kısma ayrılmıştır. İşlerin zorluk düzeyine göre öncelik verilmiş, her kısımdaki işlerin ortalama alacakları zaman hesaplanmıştır. Bulunan ortalamalar karşılaştırılıp, en küçükten en büyüğüne doğru 1 ile 3 arasında sıralanmıştır. Ayrıca elde edilen nesil dizilişleri de dikkate alınmış olup, maksimum tamamlama değerleri en küçükten en büyüğe doğru 1 ile 3 arası değer almıştır. Çalışma sonucunda her türlü problem türüne uygulanabilecek, genetik algoritma sonucunda elde edilen çözümlere karşılık olarak çeşitli kurallar bulunmuştur. Bu kurallar içinden Cmax'ı minimize etme ve en çok sayıda kullanılan kurala göre, elde edilen sıralamanın da en küçük ortalamalı iş topluluğundan uzun olana mı yoksa uzun olan gruptan kısa olana doğru olup olmadığı konusunda sonuçlar bulunmuştur. Daha önceden bilinmeyen fakat kullanılabilecek bu kurallardan çıkarılacak sonuçların diğer tekniklerle beraber çözümüyle çok daha iyi çözümlere ulaşılabilmesinin mümkün olduğu anlaşılmıştır.

Kumar ve Rao (2009) ise karınca kolonisi yöntemi ve yeni sonuçlar oluşturmak amaçlı bazı genetik algoritma aşamalarını kullanarak bloklu akış tipi çizelgeleme problemlerine yeni bir bakış açısı getirmişlerdir. Çalışmalarında Sekiz adet iş dört adet makine akış tipi problemini çalışmışlardır. Her bir sıralama için feromon değerini hesaplayıp en iyi sıralamayı elde edilmiştir.

Gönülol (2009) çalışmasında veri madenciliği tabanlı yeni bir sezgisel yaklaşım sergileyip, birçok farklı alanda sıklıkla kullanılan Gezgin Satıcı Probleminin (GSP) çözüm verimliliğini araştırmıştır. Çok sayıda rastgele çözüm ile bir eşik değer, kural içeren matris oluşturmuş ve oluşan kurallara göre turu belirlemiştir. Bu eşik değer, üretilen rastgele çözümlerin ortalama değeridir. Kural ilişki matrisinde bulunan satır ve sütün problem büyüklüğünü temsil etmektedir. Bu matriste her yapılan turda, tüm sırada olan şehirler için bir yüzdelik hesaplanmıştır. Bu oran bulunması, tur uzunluğu ile eşik değerinin farkının tur uzunluğuna oranlanmasıyla bulunur. Her tura için bu oran tekrar hesaplanıp tüm bağlı olan şehir için bulunan bu oran atanır. Bunun dışındaki tüm turların oranları ele alınan matriste bağlı olan şehirlere eklenip toplama işlemi yapılır. Tur oluştururken, kural ilişki matrisinde bulunan iller arasındaki en büyük oranlı iki adet şehir birbirine bağlanır. Daha sonra ikinci büyük oran atanan şehirler birbirine bağlanarak, bu şekilde tüm şehirler için uygulanır ve tur tamamlanır. Bu yöntemin sonuçlarının belirlenmesi maksatlı 15 adet test problemi kullanılmıştır. Çalışma sonucunda elde edilen sonuçlar, aynı problemin önceki sonuçları ile karşılaştırılmıştır. Geliştirilen yeni veri madenciliği yaklaşımının, 300'den daha az şehirli olmak şartıyla gezgin satıcı problemlerinde daha iyi sonuçlar verdiği tespit edilmiştir.

Olafsoon ve Li (2010) on iş tek makine bulunan problemi; işlerin geliş zamanı, işlerin teslim zamanı, yapılan işlem süresi, mevcut işlerin ağırlığı gibi çeşitli kriterlerden yararlanarak, hem veri madenciliği tekniklerinde kullanılan karar ağacı ile hem daha da geliştirilmiş karar ağacı algoritmasıyla çözmüşlerdir.

Thabtah ve Cowling (2010) personel çalışma çizelgeleme problemini için bir çözüm uzayı ortaya koymuşlardır. Eğer problem birçok sezgisel çözüme sahipse, hangi yöntemin daha verimli olacağı kararını vermek amacıyla bir sezgisel yöntemi daha dahil etmişler ve her bir aşama içerisinde C4.5 algoritmasının da dahil olduğu çeşitli sınıflandırma yöntemleri kullanılarak, bu algoritmalar arasında karşılaştırma yapılmıştır.

Zare ve Fakhizoal (2011) esnek akış tipi çizelgeleme probleminde gecikme süresinin minimize edilmesini hedeflemektedir. Uygun sonuçlar bulmak amaçlı genetik algoritmayla veri madenciliğini birlikte kullanılmıştır. Çalışmada parametre olarak, Koonce ve ark., (1997) ile Koonce ve Tsai, (2000) çalışmalarını kaynak göstererek birkaç parametreyi kullanmışlardır. Yapılan çalışmalardan farklı yanı çözümü daha da geliştirmek amacıyla bulanık mantık yöntemini kullanılmıştır. Genetik algoritma, veri madenciliği ve bulanık mantığın bir araya getirilmesiyle oluşan modelin, NP zor problemler arasında kabul edilen akış tipi çizelgeleme problemi için oldukça iyi sonuçlar verdiği gözlemlenmiştir.

Balasundaram ve ark. (2012) iki makine beş işten oluşan akış tipi çizelgeleme problemleri için veri madenciliği yöntemleri arasında bulunan karar ağaçlarını kullanarak çeşitli kurallar üretmişlerdir.

Özcan (2013) çalışmasında akış tipi çizelgeleme problemlerinin çözümü adına veri madenciliği ve rota birleştirme algoritmalarını kullanmış diğer metasezgisel yöntemlerle rekabet edecek bir çözüm önermiştir.

Özet olarak, veri madenciliği genelde küçük problemler üzerine uygulanmış ya da hibrit amaçlı olarak kullanılmıştır.

3. MATERYAL ve YÖNTEM

3.1. Veri Seti

Tez uygulamalarında Taillard'ın akış tipi için önerdiği veri seti kullanılmaktadır. Taillard veri seti Eric Taillard tarafından 1993 yılında akış tipi, atölye tipi ve açık tip problemler için ortaya koyduğu çalışmadır (Taillard, 1993).

Akış tipi için kullandığımız Taillard veri seti 12 ana problem ve her problemin içinde 10 alt problem olmak üzere 120 adet problemden oluşmaktadır. Bu çalışmada 100 adet problem çalışılmıştır. Bu veri seti 20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20 ve 200x10 problemlerini içermektedir. İlk sayı işi, ikinci sayı ise makine sayısını temsil etmektedir.

3.2. Materyal

3.2.1. Çizelgeleme

Çizelgeleme temel olarak bir veya birden çok performans ölçütünün optimizasyonunu sağlamak hedefiyle süreçler için sınırlı kaynakların ayrılması şeklinde tanımlanabilir (Leung, 2004).

Bir başka tanımda ise üretim çizelgeleme, mevcut tesis olanaklarının hedef ürünlerin üretimini sağlamak amacıyla sınırlı bir zaman aralığı içindeki kullanımının belirlenen bazı başarı kriterlerini hedefe uygun şekilde planlanması olarak ifade edilmiştir (Rodammer ve White, 1988).

Çizelgelemenin bir tanımı da şöyle yapılabilir. Matematiksel teknikler ya da sezgisel algoritmalar kullanılarak kapasitesi oldukça sınırlı kaynakların istenen görevler için planlanmasıdır. Kaynakların hedeflere uyumlu olarak düzenlenmesi ile firmanın kendisinden talep edilen isteklere en doğru şekilde çözüm üretmesini sağlar. Çizelgeleme konusu; parametre belirliliği, makine sayısı, geliş aşamalarının farklılaştığı birçok problem çeşitlerini kapsar. Son dönemlerde birden fazla amacın iyileştirildiği çizelgeleme çalışmalarına hız verilmiştir. Bu problemlerin çizelgelenmesi tek ölçütlü problemlerde olduğu gibi kolay değildir çünkü ilgisi bulunmayan amaçların uygun bir noktada buluştukları tek bir çizelgeyi oluşturmak ciddi çalışma gerektirir. Çizelgeleme problemleri tüm faktörlerin en iyi durumunu bulma yani birden fazla amaç için en doğru kararı vermeyi hedeflediği için en iyi çözümlere rastlamak oldukça zordur (Ceran, 2004).

Büyük üretim planlarını çizelgelemedeki ana amaç, ürünün işlem başlangıcından bitimine kadar geçen zaman aralığı içinde üretim sürecini minimum, tesisin sahip olduğu cihaz ve işgücünü kullanma oranını ise maksimum hale getirmektedir (Todd ve Sen, 1997).

Genel olarak kesikli-parça üretiminin gerçekleştirildiği endüstri tesislerinde görülebilen üretim çizelgeleme, belli bir süre hedefi içinde yapılması zorunlu olan işler için bir ya da birden fazla hedefi optimize edecek şekilde sınırlı kaynakların işlere atanması işlemidir. Bu sebeple imalat sektöründe oldukça önem verilen bir role sahip bir karar verme sürecidir. Var olan tesis imkanlarının amaç doğrultusunda planlanması ile işletmenin hedeflerine en çabuk ve en iyi biçimde varması sağlanır. Çizelgeleme probleminde bulunan sınırlar makine ve teknolojiyle ilgilidir. Çizelgeleme problemine getirilecek çözüm ise bu iki tür sınırın birbiriyle uyumlu çözümüyle mümkündür. Eğer çözüm sonucunda meydana gelen sonuç her bir işlem için hangi kaynağın atandığı ve hangi görevin hangi zaman aralığında işleme gireceğini belirliyorsa anlamlı olacaktır. Özet olarak, birçok çizelgeleme problemi sınırlara bağlı en uygun şekle sokma sorunu olarak görülmektedir (Cowling ve ark., 2002).

Hazırlık işlemleri kaynaklarda iki kısımda ele alınmıştır. İlki, öncesinde yapılacak hazırlıklar yalnızca işlem görülecek işle ilgili olup sıra-bağımsız hazırlık zamanı şeklinde tarif edilmektedir. Diğerinde ise hazırlık iki işe bağlıdır; o esnada işlem görecek işe hem de hemen öncesindeki işe bağlıdır. Bu durum sıra-bağımlı hazırlık zamanı olarak tarif edilmektedir (Allahverdi ve diğ., 1999). Sıra-bağımlı hazırlık zamanı uygulamaları matbaa sektöründe rastlanabilir. Cihazın temizlik işlemlerinin yapılması ve işleme hazırlanması, en son işlemde kullanılan mürekkebin rengine, kullanılacak kâğıt boyutuna ve özelliğiyle ilgilidir. Ayrıca metal, ilaç ve kimya alanlarında da sıra-bağımlı hazırlık zamanı uygulamalarına bolca rastlanmaktadır (Yang ve Liao, 1999).

Çizelgelemenin başarılı olabilmesi için önemli noktalar: kapasite, yeterlilik, işin gereksinimleri ve ölçüm standartlarıdır (Boray, 2007: 4).

a) Kapasite: Çizelgeleme çalışmalarının yapılacağı tesiste öncelikle kapasite bilinmelidir. Firmayı maliyet açısından zorlayacak üretim programlarının herhangi bir anlamı yoktur. Diğer taraftan kapasite değişen bir kavramdır ve üretilen malın niceliğine göre değişir.

b) Yeterlilik: Makinelerin ve işçilerin talep edilen özelliklere uygun olması önemlidir.

c) İşin Gereksinimleri: İşte istenilen kalite ve maliyet standartları bilinmelidir.

d) Ölçüm Standartları: Çizelgelemeyi etkileyecek faktörlerin (zaman, maliyet, kalite ve kapasite) tahsis edilmesiyle ilgili standartlar oluşturulmalıdır.

Rodammer ve White (1988)'e göre tipik bir üretim çizelgesinin özellikleri şöyledir: Bir çizelge 1) Belirli bir zaman aralığında çalıştırılabilmeli, 2) İş büyüklüğünü belirlemeli, sıraya-bağlı hazırlık zamanları ile oluşabilecek maliyetler hesaba katmalı, 3) Ürün rotalamada oldukça esnek olmalı, 4) Olasılıklı çizelgeleme ortamını temsil edebilme; yani ani olayları ve rahatsızlıkları içerme yeteneğine sahip

olmalı ve 5) Birden fazla performans kriterlerini maksimum iyilemeli ve bünyesinde bulundurmalıdır (Rodammer ve White, 1988: 843).

Çizelgeleme problemlerine ait bazı notasyon bilgileri şöyledir.

Bir üretim ortamında işlerin düzenlenmesiyle alakalı problem olduğu gelen taleplerin karşılanamaması gerek makine gerek insan gücünün verimsiz kullanılması, gereğinden fazla üretimin yapılmasıyla oluşan ürün birikmeleri gibi problemlerle anlaşılabilir.

Çizelgeleme sırasında iş ve makineler sınırlı olarak kabul edilir. m harfi makineyi, n harfi işi gösterir. Alt indislerden i harfi makinayı, j harfi ise işi gösterir.

Bazı temel zaman ifadeleri şöyledir:

İşleme zamanı: Pij ile gösterilir. i makinesinde j işinin işleme süresini gösterir.

Geliş zamanı: Rj ile gösterilir. j işinin mümkün olan en yakın zaman diliminde başlayacağını temsil eder.

Teslim zamanı: Dj ile gösterilir. Verilen j işinin teslim süresini temsil eder.

Ağırlık: Wj, j işinin ağırlığını temsil eder ve j işinin diğer işlere göre önceliğini gösterir. Stok yapılan üretim tesislerde ürünün gereğinden fazla üretilmesinin getirileceği maliyet gösterilebilir.

Çizelgeleme problemleri α/β/γ şeklinde gösterilen üç adet parametre ile temsil edilir. α parametresi, atölyedeki cihaz ortamını gösterir ve tek girişe sahiptir. β parametresi, işlemeyle ilgili detaylar ve bunlara ait kısıtlarıyla ilgilidir ve birden çok giriş bulunabilir. γ parametresi ise tek giriş olup sorunda en küçüklenmesi hedeflenen

değeri yani amaç fonksiyonunu gösterir (Pinedo, 2008: 14 ve Graham ve ark., 1979: 288).

α parametresinin alması muhtemel değerleri aşağıdaki gibidir (Pinedo, 2008: 15 ve Rola, 2011: 6):

a) Tek makina (1): Tüm işler sadece bir makinede muamele görür. Makine tipleri içinde en basit yapıda olanıdır. Şekil 3.1'de örnek bir tek makinalı sistem bulunmaktadır.



Şekil 3.1. Tek makineli sistem

b) Özdeş paralel makineler (Pm): Yakın özellik taşıyan m adet paralel makineyi barındırır. i işine bağlı sadece bir aşama bulunur ve bu aşamayı m adet makineden rastgele birisi yapabilir.

c) Ayrı hızlı sahip paralel makineler (Qm): Çalışma hızları başka olan m adet paralel makine bulunmakta olup, j makinesinin hızı vj ile ifade edilir. Pi işlem zamanı olan i işi j makinesinde Pij = Pi / vj süresinde meydana gelir.

d) İlişkisiz paralel makineler (Rm): Birbirinden bağımsız olan farklı hızlarla çalışan m tane paralel makine vardır. j makinesi i işini vij süresinde yapmakta olup, i işinin j makinesinde bulunduğu Pij zamanı Pij= Pi / vij formülüyle hesaplanır

e) Akış tipi (Fm): Bir akış temsil edecek şekilde seri halde sıralanmış m adet makineden oluşmaktadır**.** Tanımlanan tüm işlerin makinelerden geçiş sırası aynıdır. Şekil 3.2'de seri şekilde sıralanmış akış tipli iki adet makine sistemi gösterilmektedir.

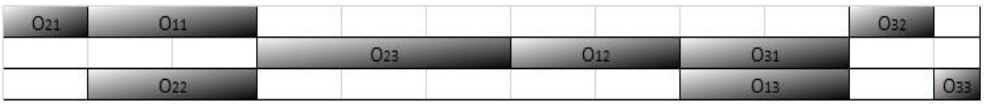


Şekil 3.2. Akış tipi çizelgelemede iki adet seri makine

f) Esnek akış tip (FFs): Toplam s tane seri kısma sahip olup, her bir seviyede aynı özelliklere sahip m tane paralel makina bulunmaktadır. Her iş benzer güzergahı izleyecek biçimde s adet adımdan hepsinde bulunan m tane makinanın ancak birinde işlem görür.

g) Açık tip (Om): Bu modelde m tane makina bulunmakta olup, her iş herhangi bir işlem görebilir. Farklı işler ait farklı yörüngeler olabileceği gibi bazı işlerin işlem zamanı sıfır olabilir.

h) Atölye tipi (Jm): Bu yapıda m adet makina bulunmakta olup her işin her bir makinada muamele görmesi şartı yoktur. Her iş kendine özel yörüngeyi takip eder ve bir makinede birden fazla işlem yapılabilir. Örnek bir atölye tipi sistem şekil 3.3'te gösterilmektedir.



Şekil 3.3. Atölye tipi çizelgeleme probleminin Gantt diyagramı (Şevkli, 2005)

β parametresine ait bazı değerler şunlardır (Ruiz ve Rodriguez, 2010: 5):

a) Geliş zamanı (r_i): i işinin başlangıcı r_i çıkış (serbest kalış) süresinden önce başlanamaz.

b) Permütasyon (p_{rmu}): Akış tipi makine problemlerinde bulunan bu kriter, makineler arasındaki sıra kuralının FCFS (önce gelen önce işleme konulur) olduğunu gösterir. İşlerin makinelerde işlem görme sırası aynıdır.

c) Öncelik sınırları (p_{rec}): Bazı işlemlerin başlanmadan önce diğer işlerin bitmesi zorunluluğuyla ilgili sınırları temsil eder.

d) Sıra bağımlı hazırlık zamanları (S_{ik}): Yapılan planda i işi k işinden önce işlem görüyorsa S_{ik}, k işleminin başlayabilmesi için gerekli olan hazırlık süresini gösterir.

e) İş aileleri (f_{mls}): n tane işin F ayrı iş ailesine bağlı olduğunu gösterir.

f) Yığın işleme (batch): Bir makinenin aynı sırada birden fazla işlem yapabileceğini gösterir.

g) Bölünebilme (prmp): İş bitine kadar makinede bulunması zorunlu değildir. Bir çalışma o anda iptal edilip aynı makineye ayrı bir iş verilebilir. İşlemi yarıda kalan iş aynı makinede tekrar işleme konulduğunda, artık o makinedeki işlem süresi önceki iş süresinden geriye kalan kadardır.

h) Arızalanma (brkdwn): Makinelerin, bozulma sebebiyle sürekli işlem yapamayacağını anlatır.

i) Bloklanma (block): Akış tipi problemlerde görülen bu parametre, artarda iki makine arasındaki sıranın sınırlı bir yeteneğe sahip olduğu ve dolu olduğu taktirde önceki makinenin işlemini tamamladığı işi bir sonraki makineye iletemeyeceğini gösterir. Bu duruma bloklanma denir.

j) Beklemesiz (nwt): Akış tipi makine problemlerinde bulunan bu kriter, işlerin arka arkaya gelen iki makine arasında duramayacağını anlatmaktadır. Bu kriterin geçerli olduğu formda da ilk gelen iş ilk işleme konulur.

k) Yeniden dolaşım (rcrc): Atölye tipi üretimlerde karşılaşılan bu kriter, aynı işin aynı makineye birden çok gelişini ifade eder.

Çizelgelerin değerlendirilmesi için amaç kriteri adına verilen ve γ ile sembolize edilen bu kriter, çizelgeleme sonucunda istenen amacın matematiksel ifadesidir

Performans kıstası iki bölüme ayrılır (Shahzad, 22.11.2011: 14):

a) Minimax: Maksimum değerin küçüklenmesini gösterir.

b) Minisum: Bir toplam değerin küçüklenmesini gösterir.

Literatürde çok fazla denk gelinen ‘Minimax’ kriteri son olarak çizelgelenen işin bitme zamanı ya da maksimum bitme zamanıdır, Cmax sembolü ile gösterilir ve kaynaklarda ‘makespan’ olarak belirtilir. Cmax=max(Ci) ile gösterilir ve Ci işlerin tamamlanma zamanını gösterir. Bitiş zamanına bağlı diğer minimax kriterleri şöyledir (T’kindt ve Billaut, 2006: 13):

a) Maksimum akış zamanı (F_{max}): Sistemde içinde bulunan en fazla zamanı belirtir, F_{max}= max(F_i) ve F_i= C_i-r_i ’dir. R_i işlerin geliş sürelerini gösterir.

b) En fazla boş makine süresi (I_{max}): En fazla boşta bekleyen makinenin boşta bekleme süresini gösterir. I_{max}= max(I_k) ile ifade edilir, I_k bir makinenin toplamda ne kadar boşta kaldığını gösterir.

c) Maksimum gecikme (L_{max}): Teslimat için müşteriye beyan edilen zamanının aşılmasının en kötüsünü ifade eder. L_{max} =max(L_i) ve $L_i=C_i - d_i$ şeklinde ifade edilir. d_i işlerin teslim zamanlarını gösterir.

d) Maksimum pozitif gecikme (T_{max}): Pozitif gecikmeler içinde en büyüğüdür. T_{max}=max(T_i) ve T_i=max(0, $C_i - d_i$) şeklinde gösterilir.

e) Maksimum erken bitirme (E_{max}): Söz verilen teslim tarihinin öncesinde tamamlanan işlerin en büyüğüdür. E_{max}=max(E_i) ve E_i=max(0, $d_i - C_i$) şeklinde ifade edilir.

Kaynaklarda çokça rastlanan 'Minisum' kriterleri ise şöyledir (T'kindt ve Billaut, 2006: 13 ve Andresen v.d., 20.09.2010: 11):

a) Toplam akış süresi (ΣF): Toplam akış süresini ifade eder.

b) Toplam geçlik (ΣT): n adet işin toplamda ne kadar geciktiğini ifade eder.

c) Toplam Makinelerin Boş bekleme zamanı (ΣI): Tesiste bulunan m tane makinedeki toplam boş bekleme süresidir.

d) Toplam geciken iş sayısı (ΣU): Teslim tarihi aşıldıktan sonra tamamlanan işlerin toplam sayısını verir.

Çizelgeleme problemlerini temelde üç farklı bölümde incelemek mümkündür (Artiba ve Elmaghraby, 1997: 274).

a) Tek makine problemleri,
b) Paralel makine problemleri,

c) Akış tipi çizelgeleme

Çizelgeleme problemlerinin anlaşılmasında en temel model, tek makine modelleridir. Tek makine modellerinde amaca uygun iş sıralarının bulunmaktadır. Daha sonra gelecek daha karmaşık çizelgeleme modellerinin anlaşılabilmesi adına temel oluşturduğu için tek makine problemlerinin çok iyi anlaşılması gerekmektedir. Tek makine problemleri en başta çok basit zannedildiği halde n! kadar ihtimalle sıralamak mümkündür. Çizelgeleme problemlerinin çözümü adına geliştirilmiş farklı yöntemler bulunmaktadır. Bu yaklaşımdan bazıları dal-sınır yöntemleri, dinamik programlama ve ödünleşim eğrileridir. Metasezgisel algoritmalar içinden tavlama benzetimi, tabu arama, karınca koloni optimizasyonu, genetik algoritma ve parçacık sürü optimizasyonu gibi yaklaşımlar örnek verilebilir. Bu problemlerde işler nasıl sıralandığı fark etmeksizin tüm işlerin bitirilmesi için gereken bitiş süresi değişmez (Dağ, 2012).

Makul zaman dilimlerinde analiz edilebilir olan tek makine problemlerinde performans ölçütünün belirlenmesi hayati önem taşımaktadır. Çeşitli performans kıstasları için geliştirilmiş çeşitli çözüm algoritmaları mevcuttur. En çok kullanılan performans kriterlerine bağlı sıralama şekilleri aşağıdaki gibi açıklanmıştır (Nahmias, 2009: 425 ve Heizer ve Render, 2008: 612):

İlk Gelen İlk Hizmet Görür (FCFS): İlk işin sıralamada ilk olarak işlem görmesi kuralına dayanır.

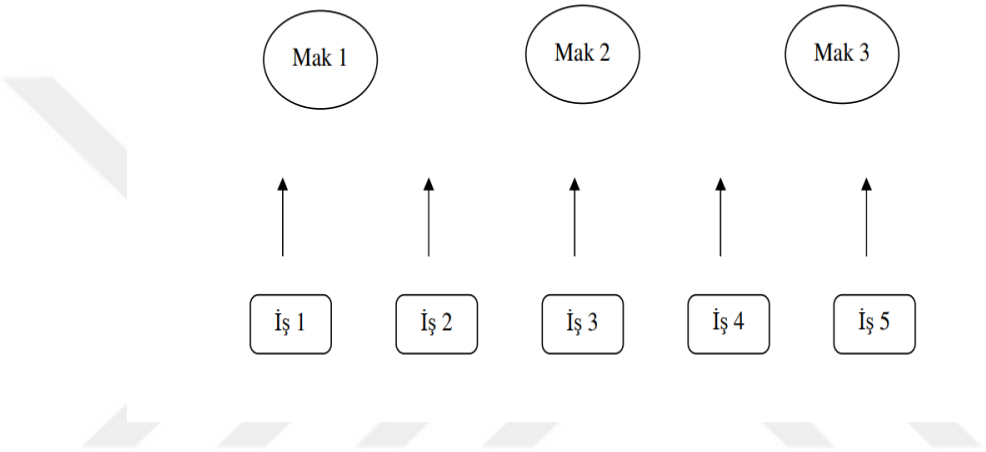
İlk Giren Son Hizmet Görür (FCLS): İlk işin en sonda işlem görmesini kuralına dayanır.

En Kısa İşlem Süresi (SPT): İşlem sürelerinin tespit edilip küçükten büyüğe sıralanır. SPT kuralında toplam akış süresi en küçüklemektedir.

En Uzun İşlem Zamanlı Sıralama (LPT): İşler işlem süresine göre azalan şekilde sıralanır

En Erken Teslim Zamanlı Sıralama (EDD): İşler teslim sürelerine göre artan şekilde sıralanır.

Paralel makine problemleri, m tane özdeş makinenin birbiriyle aynı hizada olacak şekilde konumlandırıldığı sistemlerdir. Her işe ait sadece bir işleme vardır ve bu m adet makinenin hangisi olduğu fark etmeksizin rastgele birinde gerçekleşebilir. Ekseriyetle makine adedi, iş adedinden az olmaktadır. Örnek bir paralel makine problemi Şekil 3.4'te gösterilmiştir (Döyen, 2004).



Şekil 3.4. Paralel makine modeli (Ceran, 2006)

Makinelerin seri biçimde sıralanarak yerleştirildiği düzene akış tipi yerleşim ismi verilmektedir (Pinedo, 2002).

Akış tipi çizelgelemede işler farklı makinelerde işlem görür ve aynı sırayla işleme girerler. Her bir makinada ürünün farklı kısımları üretilmektedir. İşlem şekli doğrusal olduğundan atölye tipi problemlere göre daha kolaydır.

Akış tipi çizelgeleme temel olarak birbirinden farklı m tane makine n tane işin bulunduğu; bulunan her işe ait belli sayıda işlem bulunduğu, her işlemin farklı makinelerde işlem gördüğü ve tüm işlerin operasyonlarının aynı sırayla yapıldığı

problemler şeklinde ifade edilebilir. Akış tipi çizelgeleme problemlerinde bulunan tüm işler seri olarak yerleştirilmiş makinelerde tamamen aynı sırayla işleme girmektedir (Akçay, 2009:6).

Makinelerin seri olarak artarda sıralandığı çalışma düzenine akış tipi (flow shop) yerleşim olarak ifade edilmektedir (İşler, Toklu ve Çelik, 2009: 230). Akış tipi atölye düzeninde işler; her bir işlemin farklı bir makinede uygulandığı, öncelik ilişkisi bulunan, operasyon ismi verilen iş emirlerine ayrılırlar. Akış atölyelerinde tüm işler üretimin aşamalarını aynı sıralama ile geçerler (Weng, 2000: 1359).

Akış tipi yerleşimin ana özellikleri arasında şunlar sayılabilir (Baker, 1994):

- İlk başlangıç anında n tane bağımsız ve çok aşamalı iş hazır bulunmaktadır.
- Aşamaların hazırlık süreleri işlem sırasından tamamen bağımsız olup işlem zamanına eklenmiştir.
- İşle ilgili tanımları detaylı şekilde bilinmektedir.
- Bütün makineler devamlı olarak kullanıma hazır haldedir.
- Bir işlem başlangıcı ile işlem bitişi arasında kesinti mümkün değildir.

Çoğunlukla kullanılan başarı ölçütü en büyük tamamlanma süresinin en küçüklenmesidir. Tezgâh kullanım oranının en büyüklenmesi ile tamamlama süresinin en küçüklenmesi birbiriyle ilişkilidir. Zamana göre en erken, maksimum tamamlanma süresi, geciken işlerin adedi, gecikme, ağırlıklandırılmış akış zamanı gibi kriterler de önem kazanmıştır (Cihanlı, 2010).

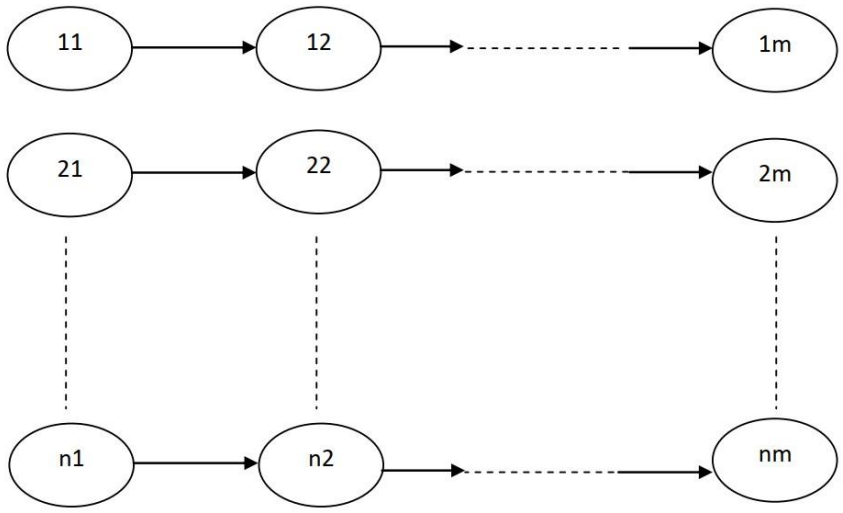
Akışı seri biçimde olan n adet iş ve m adet makineden meydana bir sistemde i makinesindeki jk işinin tamamlanma süresi aşağıdaki denklem (3.1), (3.2) ve (3.3) ile hesaplanabilir (Seçme, 2006).

$$C_{i,\,j1} = \sum_{l=1}^{i} p_{l,\,j1} \qquad \text{i=1, …, m} \qquad (3.1)$$

$$C_{1,\,jk} = \sum_{l=1}^{k} p_{l,\,j1} \qquad \text{k=1, …, n} \qquad (3.2)$$

$$C_{i,\,jk} = \max(C_{i,1\,jk}, C_{i,\,jk\,1}) + p_{i,\,jk} \qquad i=2,\ \ldots,\ m;\ k=2,\ \ldots,\ n \qquad (3.3)$$

Şekil 3.5'te örnek bir akış atölyesi yerleşimi gösterilmiştir.



Şekil 3.5. Akış atölyesi yerleşimi (Dağ, 2012)

Permütasyon tipi akış problemlerinde uygun çözümü bulabilmek için sırf, iş sıralarının her makinede eşit olduğu permütasyon hesaplarını gözlemlemek yeterli olmadığı gibi, $(n!)^m$ tane çizelgenin de gözlemlenmesi çoğu kez pratik değildir (Güldalı, 1990).

İş atölyelerinde her iş tipine göre bütün makinelerde veya makinaların bazılarında işlenecek adımlardan oluşur. Her işin belli bir akışı vardır. Akış tipinden ayıran yönü ise iş istikametinin her iş için aynı olması gibi bir zorunluluğu yoktur. Buradan şu sonuç anlaşılabilir, bir iş atölyesinde çizelgeleme yapılabilmesi için, her makinede operasyona girecek işlerin sıraya konulmasının önceden belirlenmesi şarttır. Bir kısım üretimlerde bir iş bir makinede bir kez işlem görürken, bazı üretimlerde ise birçok işlem söz konusudur. (Cheng, Gen ve Tsujimura, 1999: 344, Pinedo 2008: 179, Pinedo 2009: 83).

Atölye tipi çizelgeleme problemi şöyle tanımlanabilir. M adet makinede işlem görmek üzere n adet işi, daha önceden kesinleşmek bir akış sırası ve kapasite engellerini göz önünde bulundurarak, amaç fonksiyonunu eniyileyecek hedefiyle her bir işin başlama süresinin tespiti olarak anlatılmaktadır. Bu gibi problemlerde her işin istikameti yani makinelerden geçiş istikameti ayrı olacağından dolayı iki farklı kısıtlama önümüze çıkmaktadır. İlk kısıt, bir işe ait işlem bitirilmeden aynı işe ait diğer operasyonun başlayamamasıdır. Bununla beraber bir iş aynı zamanda sadece bir makinede işlem görebilir. İkinci kısıt ise, bir makinenin yapabileceği bir tanedir (Şevkli ve Yenisey, 2006: 59).

Bu kısıtlar haricinde klasik atölye tipi çizelgeleme problemlerinde bir takım temel kabuller vardır (Carlier ve Pinson, 1989: 164, Sule, 2008: 327):

a) Bir operasyon iş sırasında yarıda bırakılamaz.

b) Bir makine bir zaman diliminde sadece bir işi yapma kapasitesine sahiptir.

c) Bütün siparişlerin, işlem zamanları ve işlem durakları kesin olarak bilinmeli ve çalışmaya hazır halde bulunmalıdır.

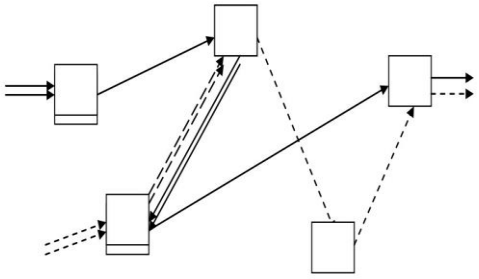
d) Hazırlık ve taşıma zamanları işlem zamanına dahildir.

e) İş durakları çizelgeleme işlemi başlangıcında uygundur, asla makina ve personel konusunda aksilik bulunmaz.

f) Her iş durağı tektir.

g) Tüm işlerin ehemmiyeti aynıdır, birbirine önceliği yoktur.

Bir iş atölyesindeki iş akışı Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Bir iş atölyesinde iş akışı (Pinedo, 2009: 24)

n adet iş m adet makine için hazırlanan atölye tipi çizelgeleme probleminde tüm ihtimallerin sayısı $(n!)^m$ ile hesaplanabilir. Makine ve iş sayısı fazlalaştıkça çizelge sayısı da tabii olarak artacaktır. Bu sebeple atölye çizelgeleme problemleri tanımı zor problemler olarak adlandırılır ve en uygun çözümleri bulmak zor olduğundan çözüm amaçlı bazı sezgisel algoritmalar üretilmiştir. (Şevkli ve Yenisey, 2006: 59).

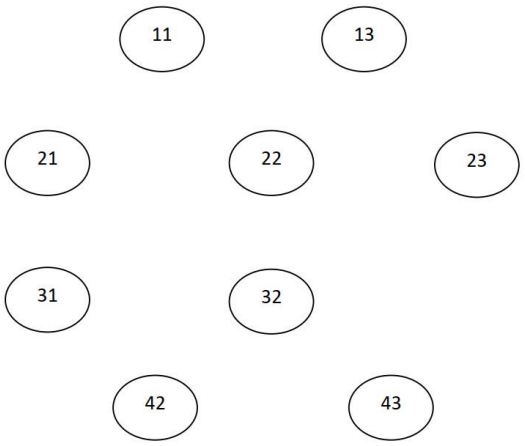
Farklı bir atölye modeli de açık atölyedir. Açık atölyede, bir işi oluşturan bölümlerin ayrı makinelerde işlenme sırasının hükmü yoktur. İşlemler gelişigüzel bir sırada olabilir, sıralama için herhangi bir engel bulunmamaktadır. Açık atölyelere en güzel örnek okul ders çizelgeleri ve sınav tarihi programları verilebilir (Liaw, 2003: 2082).

Açık atölyeler atölye tipi çizelgelemenin farklı bir sınıfını oluşturur ve açık atölyede şu özellikler bulunmaktadır:

a) Her bir i işi m tane işlem içerir Oij (j=1, …, m) ve Oij makine Mj üzerinde operasyon görmek zorundadır.

b) Bir işin adımları arasında tercih sırası yoktur (Brucker, 2007: 158).

Açık atölye sistemi Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Açık atölye modeli (Sule, 2008: 367)

3.2.2. Genetik Algoritmalar

Darwin'in evrim teorisinden yola çıkılarak canlılarda meydana gelen evrim sürecinin bilgisayar ortamında benzetimi fikrini ortaya atan John Holland, tek bir mekanik yapıya herhangi bir kavramı öğretip geliştirmek yerine bu tür yapılarda oluşan bir topluluğun üreme, çaprazlama, mutasyon, vb. adımlardan geçerek sağlıklı yeni kromozomların oluşturabildiğini fark etmiş ve bu yönteme Genetik Algoritmalar adını vermiştir.

Çözümü zor veya imkânsız olan bol sayıda problem bulunmaktadır. Bu problemlerin çözümü için ortaya konulmuş bir matematiksel yapı olmamakla birlikte yapılan çözümlerin hesaplamaları ve optimum çözümün seçimi ciddi süreler

alacaktır. Mümkün olan tüm çözümler yerine bazı çözümlerin seçilip denenmesiyle, beklenen uygun sonucu verecek yöntemler araştırılmış ve bu konuda pek çok algoritma yaratılmıştır. Bu yöntemler arasında Genetik Algoritma da yer almaktadır (Baskak ve ark., 2004).

Genel anlamda, genetik algoritmalar şu aşamalardan oluşmaktadır. Kromozomlardan oluşan bir topluluğa seçim, çaprazlama ve mutasyon işlemlerinin uygulanmasını ifade etmektedir. Bahsi geçen bu operatörler kullanılarak popülasyon içindeki bireylerden oluşan yeni bir popülasyon meydana getirilmektedir. Çocuk bireylerden meydana gelen yeni popülasyon ile ebeveyn kromozomların meydana getirdiği eski popülasyon yer değiştirilmektedir. Her bir birey daha önceden tanımlanmış fonksiyona göre bir uygunluk değerini taşır ve kromozomları seçme işlemi uygunluk değerine bakılarak yapılmaktadır. Uygunluk değeri ortalama uygunluk değerinden daha büyük olan bireylerin gelecek nesillere bulunma ihtimali daha büyüktür. Popülasyonun ortalama uygunluk değerinin devamlı artmasını sağlayan bu döngü, gelecek nesillerin daha da büyük uygunluk değerlerine sahip olacağını göstermektedir (Taşkın, Emel 2009: 13-14). Elde edilen son nesildeki birey problemi optimum çözüme ulaştırmaktadır. Bu çözüm daima optimum olmayabilir fakat optimuma yakın olduğu bir gerçektir (Karaboğa 2017: 75).

Genetik operatörler, başlangıç popülasyonundaki bireylere uygulanan yöntemlerdir. Hedefi en iyi özelliğe sahip yeni nesilleri arttırmak ve arama işlemini en geniş popülasyonda yapabilmektir. Uygulamalara göre operatör seçimi farklılık arz etmekle birlikte genelde 4 temel operatör kullanılır.

1. Yeniden Üretim
2.Seçim
3. Çaprazlama
4. Mutasyon (Öznur ve Korukoğlu, 2003)

Kromozom, bünyesinde bir bilgi taşıyan en ufak kısma gen denir (Balin, 2010). Gen, bir canlının doğuştan gelen özelliklerinden birini bulunduran kısma verilen isimdir. Başka bir şekilde anlatılacak olursa bir sorunda kaç tane karar değeri bulunursa o kadar gen vardır fikri oluşabilir (Soyaslan, 2010).

Genetik algoritmalar, evrendeki evrim oluşumundan ilham alındığı için benzer terimler kullanılmaktadır. Popülasyona ait bireylerin öznel özelliklerine karar veren irsi kısımlara gen ismi verilmektedir (Oraler, 1990; Özkan, 2003)**.**

Bir ya da birçok genin birlikteliğinden probleme ait bilgi içeren yeni yapıya kromozom denir (Balin, 2010). Şöyle de denilebilir genlerin beraberliğinden oluşan bireyler kromozomlar olarak tanımlanır. Kromozomda bulunan her bir gen bir ya da birkaç özelliği göstermektedir. Genler, gösterdikleri değişkenin değerini gösterir ve aldığı değere göre kromozomların özelliği değişmektedir. Örneğin ten rengiyle ilgili bir gende renkler birer değerdir (Michalewicz, 1996).

Kromozomların bir araya gelmesiyle oluşturduğu topluluğa popülasyon ismi verilmektedir. Popülasyon, tüm çözüm uzayından seçilmiş kısmi çözüm kümesidir. Popülasyondaki birey adedi (kromozom) genellikle sabittir. Genetik Algoritma ’da topluluktaki kromozom sayısı ile ilgili kesin bir kural yoktur. Topluluktaki birey adedi arttıkça çözüme ulaşmak daha da hızlanır (Şeker, 2007).

Kodlama ise şöyle tarif edilebilir:

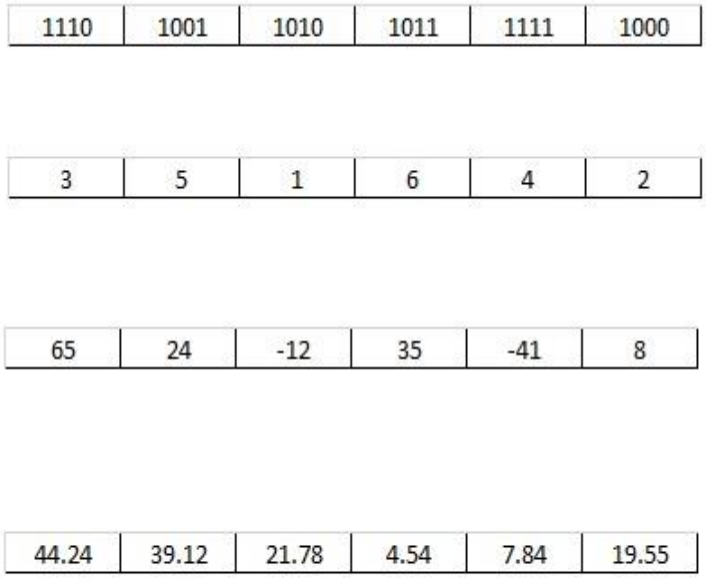
Problemde Genetik Algoritma işlemleri yürütülmeden önce, verinin doğru bir şekilde kodlanmasına ihtiyaç vardır. Doğru kodlamayla doğru bir genetik yapı oluşur ve kodlamanın tarzı probleme göre değişmektedir. Bireylerin kodlanması hususunda dikkate alınması gereken üç nokta bulunmaktadır (Çeyrekoğlu, 2017). Bunlar;

- Kromozomların uygunluk değeri kontrol edilir.
- Kromozomların belirtilen şartları sağlaması kontrol edilir.
- Kodlama evreninde, bireyin tek olduğuna dikkat edilir.

Bazı kodlama teknikleri şunlardır:

- İkili Kodlama
- Permütasyonlu Kodlama
- Değer Kodlama
- Ağaç Kodlama

Şekil 3.8'de başlıca kodlama türleri gösterilmiştir.



Şekil 3.8. Genetik algoritma başlıca kodlama türleri (Eroğlu, 2018)

Uygunluk fonksiyonu amacın matematiksel modeli olarak da isimlendirebilecek, optimize edilmesi istenen fonksiyondur. Uygunluk fonksiyonu ile bireyin çözüm için durumu belirlenir.

Yeniden üretim aşamasında, mevcut nesil yerine daha sağlıklı nesil oluşturma işlemine; nesil üretimi, nesilde bulunan bazı kromozomlardan yeni kromozomların değiştirme işlemine de kararlı durum üretimi olarak ifade edilmektedir (Öznur ve Korukoglu, 2003). Üretim, uyumluluk değerine göre yapılmaktadır. Bu değeri yüksek bir kromozomun kalıtsal özelliklerinin, yeni nesile geçmesi işlemini yapar. Seçilen uyumluluğu yüksek bir kromozom çiftinin, yeni kuşağın kromozom oluşumuna yardımcı olur. Bu kromozom çiftine ebeveyn ismi verilmektedir. Ailenin özellikleri, yeni kuşağa aktarılması aşamasında, oluşturulan yeni kromozomlarda aynı işlemin tekrarına düşülmemesi ve bu bireylerin ailelerinin aynısı olmasını engellemek için çaprazlama ve mutasyon operatörleri kullanılması gerekmektedir (Paksoy, 2007).

Popülasyonun içinden hayatta kalabilen kromozomlara karar verilmesi için çeşitli yöntemler uygulanabilir. Yöntemin etkin olup olmadığı Genetik Algoritmanın çalışma hızını ve daha iyi sonuç alınmasını etkilemektedir. En basit ve yoğunlukla kullanılan yöntemler arasında rulet çemberi, orantılı, sabit durum, sıralama ve turnuva seçim yöntemleri de bulunmaktadır (Nabiyev, 2010; Elmas, 2011).

Holland tarafından Rulet tekerleği seçim şekli sunulmuştur. Bu seçim şekline göre uyum değeri hesaplanan her bir kromozomun başarı oranına göre seçilme ihtimali ortaya çıkar (Gen ve Cheng, 2000). Popülasyondaki her birey uyumluluk fonksiyonu değeriyle ölçülü olarak rulet tekerleğinde bir kısma yerleştirilir. Tahmin edilebileceği gibi iyi çözümler geniş bir alanda gösterilmektedir. Bu sayede seçilme ihtimali artar. Rulet tekerleği yeniden üretimde kullanılmak üzere aday birey seçimi için tekrar döndürülür (Sastry ve ark., 2005).

Sıralama seçimi şeklinde ise popülasyondaki her bir kromozom uyum değerine göre sıralanır. En kötü uyum değerine sahip kromozom (1'den) en iyiye doğru (N) artan şekilde sıralanır. Ancak uyum fonksiyon değeri kötü olanların seçilme ihtimallerini daha az tutar. Bu durum birey çeşitliliğini koruduğu gibi başarılı bir sonuca ortam oluşturur. Yeni oluşacak nesilleri oluşturacak kromozomlar bir turnuva ile seçilir (Sivanandam ve Deepa, 2008):

Turnuva yönteminde ise N_u tane kromozom yarış için seçilir. Gruptaki bulunan birey sayısına turnuva genişliği denir. Turnuvada olan bireyler (kromozomlar), rassal seçilir. Kazanan kromozom yeni popülasyona eklenir (Bolat ve ark., 2004).

Denge durumunda meydana geldiğinde sıralı seçim metoduna öncelik verilerek seçilen bir ya da iki bireye genetik işlemler uygulanır. Yeni nesiller ile uyum değeri kötü nesiller yer değiştirilir. Bu sayede yeni popülasyon meydana gelir (Altıparmak, 1996).

Çaprazlama operatörü, genetik algoritmanın özellik aktarımı ve performansını önemli ölçüde değiştiren parametrelerinden biridir. Çaprazlama, ana kromozomda seçili genler üzerinden yeni bireyler oluşturulur. Bunun en basit tatbikatı, rasgele bir nokta oluşturup, bu noktadan önceki tüm genleri ilk ebeveyn, sonraki tüm genleri ikinci ebeveynden alıp yeni bireyi oluşturmaktır (Çalışkan ve ark., 2016).

Çaprazlama yöntemi, genlerin mevcut uyumluluk değerlerini kontrol için kullanılır. Kodlanmış tüm bilgiyi seçilen topluluk içermez ise, çaprazlama yöntemi istenilen çözümü üretemez. Bu nedenle, mevcut bireyden yeni birey oluşturan bir operatör gerekmektedir. Mutasyon bunu sağlamaktadır. Mutasyon operatörü, çözümdeki başarılı genleri kayıplara karşı koruma sağlamaktadır (Goldberg, 1989).

Nesildeki bireylerin çeşitliliğinin artması için bireylerin bazılarına mutasyon işlemi uygulanır. Bu işlemin amacı popülasyonun içindeki değişimi oldukça sınırlamaktır. Mutasyon işlemi sırasında genelde kromozomdaki gen adedi sabit kalır. Ekseriyetle önerilen mutasyon oranı 0.005/bit-nesildir. Bu işlem, çaprazlama sonrasında yapılmaktadır. Mutasyon ihtiyacı olasılık testiyle anlaşılır (Engin, 2013).

Mutasyon çok basit olarak kromozomun herhangi bir hanesindeki 0 değerini 1 ya da 1 değerini 0 yapma işlemidir. Bu işlem bir veya birden fazla hanede de yapılabilir. Böylece bir değerin bile değişimi ile bile çok farklı bir değer çıkar (Balin, 2010).

Genetik Algoritma evrende var olan en iyinin yaşayıp zayıfların elenmesi döngüsüne dayanarak devamlı iyileşen yapılar meydana gelir. Bu algoritmada iyinin tanımı olan uygunluk fonksiyonu ve yeni çözümler oluşturmak için yeniden kopyalama, değiştirme gibi işlevler kullanır. Genetik Algoritmanın önemli bir farkı da tüm çözümlerle değil bir kısım çözümlerle uğraşmasıdır. Böylece tüm çözümler üzerinde gereksiz vakit kaybı yaşamayıp çok sayıda çözümün içinden belli kriterler doğrultusunda iyileri seçilip zayıf olanlar elenebilir.

Genetik algoritmaların çalışma prensibi şöyledir.

Genetik algoritmanın temel kodu aşağıda verilmiştir (Gen ve Cheng, 2000).

A(t): t. nesildeki popülasyon

B (t): t. nesildeki yeni bireyler (kromozomlar)

begin

t=0;

A(t); Başlangıç popülasyonunu yarat

A(t); Kromozomların uyum değerlerini bul

while (bitme koşulu sağlamıyorsa) do

begin
B (t) = A(t) yi operatörler ile değiştir.
B (t) nin kromozomlarındaki uyum değerlerini hesapla
A (t+1) (yeni popülasyon) = B (t) ve A(t) den uygun olanları al
t=t+1;
end;
end;

Genetik algoritmada bir takım genetik operatörler kullanılmaktadır. Genetik operatörlerin kullanılma gayesi yeni kromozomlar yeniden üretimle oluşturulur. Yeniden üretilen kromozomlardan hangilerinin hayatta kalacağına karar verecek seçim için belirlenen yöntem ve onun oranına bağlıdır. Bu sınırın altında kalan bireyler elenir. Bu yeni nesiller yapılırken sürekli benzer işlemlerin yapılmasını önlemek adına başarılı kromozomların bir kısım genleri çaprazlanır ve böylece yeni neslin başarısı artmış olur. Mutasyon işlemiyle de başarılı kromozomların çeşitliliği korunur, genelde 0,05 civarında seçilir.

Genetik Algoritmada çözüm uzayı belirlenirken belli bir iterasyondan sonra nesil çeşitliliği olmayacağı için çözüme ulaşamama, kısır döngü sorunu oluşmaktadır. Nesil çeşitliliğini oluşturmak ve uyumluluk değeri en iyi olan bireylerin genlerinin yeni nesillere aktarılması için bireylere genetik operatörler uygulanır. Böylece çeşitlik sorunu oluşmaz. (Elmas, 2011, s. 393). Bu operatörler ve seçilen oranlar algoritmanın başarısını oldukça etkiler (Nabiyev, 2010).

Genetik algoritma yapısı gereği sürekli daha iyiyi aramaya meyillidir ancak bu uzun süreler alacağından dolayı belli bir sınır değerinde çalışmayı bitirmesi gerekmektedir. Bitirilme sınırı doğru tahmin edildiği müddetçe çalışmanın sonunda elde edilecek çözüm en iyi çözüm olacaktır. Durması için sınır dışında artık daha uzun süreler çalıştırmaya değecek bir başarı yoksa belli bir değerden sonra durdurulur.

Genetik algoritmalara, sonlandırma hedefi verilmezse sürekli devam eder. Dolayısıyla başlangıçta, amaç doğrultusunda bir durdurma kriteri tespit edilmelidir. Genetik algoritma başlangıçta girilen adım sayısı kadar çalıştırılır ve durdurulur. En sağlıklı çözüm son nesildeki en iyi nesildir (Şen, 2004).

Uyum değeri belirli bir noktaya eriştiğinde genetik algoritma durdurulur (Safe ve ark., 2004). Nesiller arası değişimde kayda değer farklılık olmuyorsa yani başarı artmıyorsa algoritma durdurulur. Hedef fonksiyona ait standart sapma ya da amaç fonksiyonun ortalaması istenen değere ulaşırsa durdurulur (Safe ve ark., 2004; Şen, 2004).

Proyag ve ark. (2019)'nın oluşturduğu GA-Pro yönteminin kaba kodu ise şöyledir

1. n iş sırasını artan sırayla başlat. (1 2 ... n)
2. 4 iş ile başla (1 2 3 4). 4! = 24 dizi. i = 4.
3. 24 dizinin her biri için tamamlama süresi hesapla. Artan sırada düzenle
4. En iyi k (k = 20) dizileri seç.
5. 3 * 20 = 60 diziye sahip olmak için Rulet Tekerleği Seçimini kullan.
6. En iyi k dizilerini seç.
7. k dizilerinin her biri için, diğer k-1 dizileri ile çaprazlama uygulayın.
8. En iyi k dizilerini seçin.
9. k dizilerinin her birine Ters Mutasyon uygulayın
10. En iyi k dizilerini seçin.
11. Her k dizisinin üzerine İkili Takas Mutasyonu uygulayın
12. En iyi k dizilerini seçin.
13. k dizilerinin her biri için çaprazlama uygulayın
14. En iyi k dizilerini seçin.

15. i> n ise, Adım 18'e gidin.

16. Her olası pozisyonda k dizilerinin her birine i işi ekleyin.

17. Adım 4'e geçin.

18. En iyi diziyi seçin. Dizinin çıktısını ve karşılık gelen tamamlama süresini bul.

3.2.3. Veri Madenciliği

Günümüzde artık veriye ulaşma problemi yerine çok fazla verinin içinden sağlıklı analiz yapılabilmesi sorunuyla karşı karşıyayız. Örneğin bir markette markete ait müşteri kartı vasıtasıyla ürün, harcama, yoğunluk saatleri gibi verilere ulaşılabilir ancak bu veriler doğru bir şekilde yorumlanmazsa belleklerde yer kaplamaktan öte bir anlam ifade etmeyecektir. Aynı marketin müşteri memnuniyeti adına yoğunluğun bulunduğu saatlerde kasiyer sayısını arttırması, müşterilerin genelde ikili olarak satın aldıkları reyonların yan yana getirilmesi, müşteri yoğunluğunun olmadığı saatlerde bakım çalışmalarının yapılması veriyi yorumladığı anlamına gelmektedir. İşte veri madenciliği de büyük veri yığınlarından anlamlı bilginin süzülüp karar verme süreçlerinde kullanılması anlamına gelir.

Verinin devasa boyutlarda olması ve çeşitli kaynaklardan toplanması, verinin gürültü, hata ve içerikle alakasız olmasında önemli rol oynamaktadır (Akpınar, 2014).

Veri işlenmesi, veri analizinin doğru bir biçimde yapılabilmesi için verinin uyumlulaştırılması, verinin gereksiz kısımlardan temizlenmesi, verinin uygun formata adapte edilmesi ve istenilen yapıya indirgenmesi gibi aşamalardan oluşmaktadır.

Veri işlenmesinin yanlış yapılması halinde çıkan sonuçlar da bir anlam ifade etmeyecek ve iş-zaman kaybına neden olacaktır. Verinin ortak anlaşılacak bir formatta olması hem çalışmanın verimli olması hem de gelecek çalışmalara kaynak teşkil etmesi bakımından oldukça önemlidir.

Veri Entegrasyonu çeşitli veri depolarından toplanan verilerin bir yerde birleştirilmesi veya bütün olarak kullanıcıya takdim edilmek amacıyla için zorunlu işlemlerin uygulanmasıdır (Akpınar, 2014).

Verilerin sadece aynı kategoride olması değil aynı birimde olması da oldukça önemlidir. Örneğin uzunluk için bir veri km birimi, diğer bir veri mil birimini kullanmış olabilir. Uygun dönüşüm yapılmadan bu iki verinin beraber kullanılması sonuçların hatalı çıkmasına neden olacaktır.

Verilerin kullanılan yöntem ya da algoritmayla uyumlu olması oldukça önemlidir. Örneğin virgülden sonraki iki ya da üç basamağın önemli olduğu bir işlemde yuvarlanan veriler kullanılırsa sonuç hatalı olacaktır. Başka bir örnekle ifade edecek olursak bir matematik fonksiyonun değer verisi yerine metin verisi girilmesi durumunda algoritma hata verecektir.

Konuyla ilgili verinin seçilmesi de oldukça önemlidir. Örneğin çayın kalitesiyle alakalı bir veri setinde kişinin cep telefonu verisinin bir önemi yoktur. Bu verinin bulunması en başta işlemin uzamasına neden olacaktır. Bunun dışında çay kalitesiyle cep telefonu numarasıyla bir ilişki bulmaya çalışacağından anlamsız sonuçlar çıkmasına da neden olacaktır.

Veri madenciliğinde kullanılan yöntemler iki bölümden oluşmaktadır. Bunlar tahmin edici ve tanımlayıcı olarak iki ayrı kısımda incelenmektedir (Çerkez, 2003).

- Tahmin Edici Modeller:

Sonuçları önceden bilinen verilerden yola çıkılarak bir yapı oluşturulması ve oluşturulan bu yapıdan yola çıkarak, sonuçları bilinmeyen verilerin sonuçlarının öngörüde bulunulması hedeflenmektedir.

- Tanımlayıcı Modeller:

Karar almaya destek olacak eldeki verilerin iç bağlantısını anlaşılabilir hale getirecek denklemlerin tanımlanması sağlanmaktadır. Veri madenciliği aşamalarında hedef, verinin içyapısını modellemek, nasıl kümelendikleri ve bu kümelerin yapısını çıkarmak, var olan veriler ışığında çeşitli tahminlerde yapmak, özetle verinin sistemini anlamaktır (Tantuğ, 2002).

Veri madenciliği modelleri, ortaya koydukları fonksiyona göre şu biçimde sıralanabilir (Çerkez, 2003). Sınıflandırma, tahmin, öngörü, birliktelik kuralları ve ardışık zamanlı örüntüler, kümeleme ve tanımlamadır.

Öngörüyü sınıflandırma ve tahmin modelinden ayıran en önemli fark, kayıtların, mevcut, gelecek için tahmin edilen eylem ve değerler haricinde kategorize edilmesidir. (Çerkez, 2003). Öngörü modelinde oluşturulan sınıflandırmanın sağlamasını yapmanın tek yolu, bekle ve gör kuralıdır (Tantuğ, 2002).

Birliktelik kuralları ve ardışık zamanlı örüntüler, ticari maksatlı olarak Pazar sepeti analizi olarak adlandırılıp veri madenciliği alanında oldukça yer almaktadır. Aynı zamanda bu teknikler; tıp, finans ve farklı vakaların kendi içinde ilgisinin olduğunun anlaşılması birlikte anlamlı bilgi ediniminin mümkün olduğu çalışmalarda da dikkate değerdir (Çerkez, 2003).

Şekil 3.9'da alışveriş sepet analizi gösterilmektedir.



Şekil 3.9. Alışveriş sepeti analizi (Han ve Kamber, 2006)

Tahmin yöntemi şöyle tanımlanabilir. Sınıflandırma modelinde bağımlı değişkenler belli bir değerde iken, tahmin modelinde ise bundan ayrı olarak bahsi geçen değerler süreklilik arz etmektedir. Giriş amaçlı kullanılan veriden öngörü işleminin sonunda niteliği bilinmeyen fakat devamlılık gösteren değişkenler için veri üretilmiştir (Çerkez, 2003).

Sınıflandırma, veri madenciliği modelleri içinde en geniş kullanıma sahip olanıdır. Sınıflandırma, yeni bir verinin vasıfları analiz edip, hiç tanımlanmamış sınıflar içinde müsait olanına eklenmesi olayıdır. Veri madenciliğinde işlem yapılacak veriler, veri tabanında kayıt olarak gösterilmekte ve sınıflandırma işlemi nihayetinde her bir kayıt için önceden belirlenmiş bir sınıf kodu bulunmaktadır (Çerkez, 2003).

Sınıflandırma, hangi sınıfa sahip olduğu bilinmeyen verilerin hangi sınıfa ait olduğunu belirlemek için, veri setleri arasından veri sınıflarını veya kavramlarını

ifade eden bir belirleme sürecidir. Sınıflandırma modeli, sınıf etiketi önceden tanımlanmış bir takım eğitim verisinin model yapısından geçirilmesine dayanmaktadır (Han ve Kamber, 2006).

Veri madenciliği algoritmaları mühim istatistiksel yöntemlerinden olan bayes ve regresyon sınıflandırma algoritmaları ile karar ağaçları, destek vektör makinesi, yapay sinir ağları algoritmaları gibi metotları da kullanmaktadır (Lippmann, 1987).

Verinin hangi özellikleri barındığı, bir bakıma robot resminin çıkarılması sonrasında yapılacak analizlerin de sağlıklı olup olmayacağını etkileyecektir.

Veri tabanında bulunanları vasıflandırıp, verileri ortaya çıkaran ister müşteri olun ister ürün olsun daha iyi tanınmasına yardımcı olur. Bir aksiyon doğru tanımlandığı ölçüde, o net sonuçlar verilebilir. Doğru yapılan bir tanım başlangıç konusunda fikir verir (Çerkez, 2003).

Kümeleme basitçe tarif edilecek olursa veri setinde içinde bulunan ve benzerlik gösteren elemanların aynı kümede olacak şekilde seçilmesidir. Kümeleme yönteminde, aynı kümede bulunan elamanların olabildiğince kendi arasında bağdaşık, farklı kümelerde yer alan elemanlar ile mümkün olduğu kadar ayrı olması amaçlanır (Akpınar, 2014).

Birçok fiziksel veya soyut objeyi denk objelerin sınıflarına yerleştirmesine kümeleme denir. Kümenin anlamı, aynı küme içerisinde benzerlik taşıyan ve diğer kümelerdeki objelerle benzerlik taşımayan veri bütünüdür (Han ve Kamber, 2006).

Kümeleme yöntemi ile tahmin çalışması yürütülecek bir ayırım olmadığı durumlarda, bunun dışındaki örneklerin kendiliğinden gruplara yerleştiği zamanlarda

uygulanabilir. Örnekler arasında bir kısmının diğerlerine göre kendi aralarında daha iyi bir benzerlik ilişkisine sahip oldukları bir yapıdır (Witten ve Frank, 2005).

3.2.3.1. Karar Ağaçları

1970'lerin sonuna doğru J.Ross Quinlan tarafından ID3 yöntemi olarak isimlendirilen yöntem, karar ağacı algoritması olarak vasıflandırılmaktadır. Quinlan gelecek yıllarda yeni danışmanlı öğrenme yöntemlerine kaynak teşkil edecek olan C4.5 algoritmasını ortaya atmıştır (Han, 2006). Karar kümelerini dizaynı, ağaçlardan esinlenilmiştir. Bu kararlar veri kümeleri arası bağıntılardan doğan sınıflandırılma kategorisindeki yöntemleri oluşturur. Özel karar ağaçları örnekleri olarak CART ve CHAID metotları sayılabilir (Yurtsever, 2002).

Verileri sınıflandırmak ve bu verilerle öngörü yapmak, birbirleriyle oldukça yakın ilişkili işlemlerdir. Bu işlemler karar ağaçları algoritmasıyla net şekilde anlaşılabilir. Mesela; önceden tespit edilen bir kişinin, seçilen bir hareketi önceden tahmin edilmek istenebilir. Bu durumda o kişinin, evvelden bilinen bir kişi sınıfında bulunduğu ve dolayısıyla belirli davranış kalıplarını göstereceği kesindir.

Karar ağacı aslında ağaç şeklinde olan bir akış şeması olarak tasvir edilebilir. Ağaçta bulunan her bir düğüm, bir vasıfla ilgili bir çalışmayı gösterir. Her dallanma test işleminin çözümünde çıkan değeri gösterir ve sonuç olarak ağaç sınıflar ile nihayete erer (Han, 2001).

Karar ağaçlarının gerçekleştirildiği analizler genellikle, farklı olayların yüksek, orta, düşük olarak nitelendirilmesi gibi çeşitli kısımlara ayrılması, sonuçta meydana gelebilecek verilerin kestirimi için çeşitli prensiplerin oluşturulmasında, belirlenen alt gruplara ait olan bağların ortaya konulması, kategorilerin toplanması, tıbbi

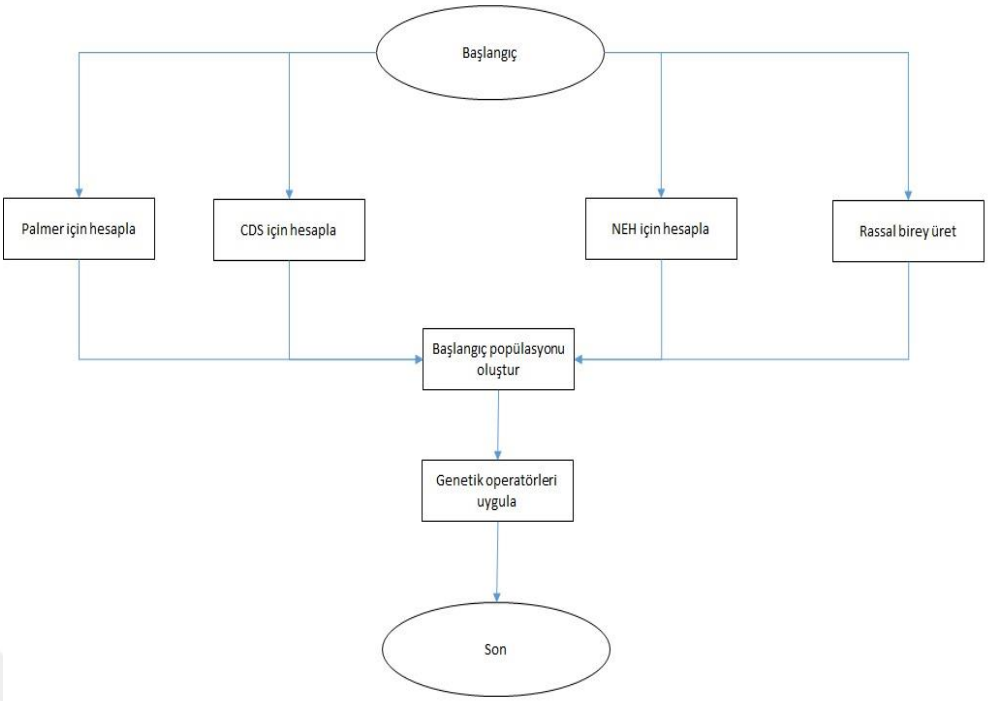
gözlem verilerinden yola çıkıp doğru kararların verilmesi hedefiyle çeşitli sorunlarda kullanılmaktadır **(**Bishop, 1996).

Tümevarım algoritmaları, büyük veri uzaylarını analiz etmede çok iyidir ve tümevarım yöntemleri ile meydana gelen karar ağacı ile karar süreci oldukça hızlanır. Yapılan tahminler de daha açık neticeler verir **(**Komorowski, 1997).

Karar ağacı temelli analizler birçok alanda kullanılmaktadır. Bunlara örnek olarak kişilerin kullanım alışkanlıklarına bağlı olarak bankaların verilecek krediye karar vermesi, iş yerinde verimli olan personele ait özelliklerin tespiti, üretim yapılan bir tesiste hataya neden olan hususların belirlenmesi, analiz edilmiş verilerden gelecekteki muhtemel olayların tahmin edilmesi, hangi birey gruplarının hangi reklam çalışmasına daha uygun olduğu gibi örnekler verilebilir.

3.3. Önerilen Yöntem Hibrit-GA

Bu çalışmada sezgisel yöntemlerin rastgele seçilecek bir kromozoma göre daha iyi çözümler içeren sonuçlarını başlangıç popülasyonunda kullanarak iyiler arasından daha iyi ulaşacak hibrit bir yöntem hedeflenmiştir. Palmer, CDS ve NEH algoritmalarıyla elde edilen sonuçlar genetik algoritmanın başlangıç popülasyonunda kullanılarak genetik algoritmanın tek başına elde ettiği sonuçlardan daha iyi sonuçlar elde edildiği görülmüştür. Hibrit-GA yöntemin algoritması Şekil 3.10'de gösterilmiştir.



Şekil 3.10. Hibrit-GA yöntem algoritması

Veri seti olarak Taillard (1993)'ın akış tipi problemler için oluşturduğu 100 adet problem kullanılmıştır. Çalışmada kullanılan Taillard problemleri 20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10, 100x20 ve 200x20 problemlerinden oluşmaktadır. 20x5 probleminde 20 iş sayısını 5 ise makine sayısını temsil etmektedir. Amaç fonksiyonu denklem (3.3)'te verilmiştir.

Veri seti Palmer, CDS, NEH, Genetik ve önerdiğimiz Hibrit-GA algoritmalarıyla çalıştırılmıştır. Hibrit-GA yöntemimiz; Taillard veri setinden rastgele oluşturulacak kromozomlar ile elde edilen Palmer, CDS, NEH sıralama sonuçlarını tekrar genetik algoritmanın başlangıç popülasyonuna dahil edilerek iyi sonuçlar içinden en iyisinin elde edilmesidir. Ardından 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Problemlerin üst değerlerini oluşturulan değerlerden yararlanıp dört adet öz nitelik seçilerek (iş süresi, standart sapma, iş sırası ve çarpıklık) bunlara veri madenciliği yöntemlerinden olan karar ağaçları uygulanmıştır. Karar ağaçları sonucu

ortaya çıkan kurallar ile o probleme ait üst değeri oluşturan kromozomun ne kadar örtüştüğü gözlemlenmiştir.

Elde edilen sonuçlar Wang (2017) ve Öztekin (2009) tarafından önerilen yöntemle karşılaştırılmıştır.

Veri setinin testi sırasında genetik ve Hibrit-GA yöntemde popülasyon büyüklüğü, iterasyon sayısı, çaprazlama oranı ve mutasyon oranı olarak Proyag (2019) tarafından kullanılan parametreler muhafaza edilmiştir.

Popülasyon büyüklüğü, iterasyon sayısı gibi değerlerin daha büyük seçilmesi de mümkündür ancak işlem süreleri ciddi miktarda artacağı için mevcut algoritmalardan iyi sonuç ve makul sürelerde sonlanmasına karar verilmiştir.

Çizelgeleme uygulaması Proyag (2019)'ın bu konudaki açık kaynak kodlu yazılımı temel alınarak Python tabanlı bir alt modül geliştirilmiştir.

Verilerin işlenmesi süreci Microsoft Excel 2019 programıyla yapılmıştır.

Veri madenciliği uygulaması Weka 3.8.4 programıyla yapılmış olup, karar ağaçları yöntemiyle (programda geçen ismiyle J48) çeşitli kurallar oluşturulmuştur.

Uygulama Windows 10 Enterprise 2015 işletim sistemi, i7-7500u, 2.70 GHz işlemci ve 8 GB RAM'e sahip bir notebook üzerinde çalıştırılmıştır.

Dil olarak Python seçilmesinin nedeni tüm platformlarda çalışabiliyor olması, geniş kapsamlı kütüphanelere sahip olması, düzenleme, hata, ayıklama döngüsünün hızlı olması gibi gerekçeler sayılabilir. Python geliştirme ortamı olarak Spyder tercih edilmiştir.

3.3.1. Kullanılan Algoritmalar

Akış tipi çizelgeleme problemleri için çalışma 1954 yılında Johnson yapılmıştır. İki makine bulunan sistemlerde optimum sonuca ulaştırmak için basit bir algoritma bulmuştur. Makine sayısının artmasıyla çözüm için çeşitli sezgiseller geliştirilmiştir. Bu yöntemlerin birkaçı şunlardır (Engin, 2001:2):

- Palmer Yöntemi (1965)
- CDS (Campbell, Dudek ve Smith) Algoritması (1970)
- Gupta Yöntemi (1971)
- Dannenbring Yöntemi (1972)
- NEH (Nawaz, Enscore, Ham) Yöntemi (1983)
- Hundal ve Rajgopal Yöntemi (1988)
- Widmer ve Hertz Yöntemi (1989)
- HC (Ho ve Chang) Yöntemi (1991)

Zikredilen klasik yaklaşımlar, problemleri çözümünde uzun zaman alan işlem süreleri ve problemlere ait olarak yapılarında hareket esnekliği bulundurmamaları sebebiyle zaman içinde tercih edilmemeye başlamıştır. Bu sebeplerle, daha az parametre kullanılarak daha iyi sonuçlara, daha az zamanda ulaştıran ve probleme ait tüm özellikleri yansıtılan yeni yaklaşım olarak metasezgisel yöntemlere başvurulmuştur. Metasezgisel algoritmanın ana fikri evrenden yararlanma, sürü benzeri yaklaşımlar ve yapay zekâ teknolojileri etkilidir. Günümüzde birçok metasezgisel, birlikte kullanım ile hibrit edilerek çok daha etkili çözümlere ulaşılmıştır.

Problem çözümünde kullandığımız yöntemlerin detayları ise şöyledir.

3.3.1.1. Palmer yöntemi

Bu yöntem iki aşamada uygulanır.

i. n iş m makineden oluşan bir akış tipi problemde, her iş için denklem 3.4'de Aj değerini hesaplanır.

$$\text{Aj}=\sum_{i=1}^{m}\{m-(2i-1)\}\,pij \tag{3.4}$$

ii. Aj değerlerine göre işler azalan şekilde sıralanır

3.3.1.2. CDS yöntemi

- Bu algoritma, n iş m makineden (makine sayısı ikiden büyük olmak üzere) oluşan bir sistemi, c adet alt-problemine indirir (c=m-1).
- Her problem Johnson algoritması ile çözülür ve Cmax değerleri bulunur.
- Alt problemler arasında minimum Cmax değerini veren çizelge, çözüm çizelgesi olarak kabul edilir.
- Alt çözümler denklem 3.5 ve 3.6'da verilen denklemle bulunabilir

$$M_1' = \sum_{i}^{k} P_{ij} \tag{3.5}$$

$$M_2' : \sum_{i=m-k+1}^{m} Pij \tag{3.6}$$

3.3.1.3. NEH yöntemi

NEH algoritması adım adım ilerler. Algoritmanın adımları aşağıdaki gibidir:

Her bir işin, tüm makinelerde işlem gördükten sonra oluşan sürelerin toplamı bulunur. Tj değeri denklem 3.7 ile hesaplanabilir.

$$T_j = \sum_{i=1}^{m} p_{ij} \tag{3.7}$$

İşler Tj değerlerine göre sıralanır. Bu sıralama azalan şeklinde yapılır.

Listedeki ilk iki iş alınır ve birbirinin yerine konularak iki alternatif alt sıralama oluşturulur. Bu durumda oluşturulan alt sıralamaların çizelgelenmesiyle oluşan Cmax değerleri bulunur. Cmax değeri yüksek olan sıralama elimine edilir. Cmax değeri düşük olan sıralama dikkate alınır.

Daha sonra bir sonraki iş alınır ve mevcut sıralamadaki tüm yerlere sırayla koyularak denenir. Elde edilecek tüm alt sıralamaların Cmax değerleri bulunur.

Cmax değeri yüksek olan sıralama elimine edilir. Cmax değeri düşük olan sıralama dikkate alınır.

Eğer sıralanmamış iş varsa döngü devam eder, yoksa durulur.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Palmer, CDS, NEH, Genetik ve Hibrit-GA yöntemimizin performansını değerlendirmek amacıyla Taillard (1993)'ın veri seti olarak kullandığımız probleminin en üst değerlerine ne kadar yaklaştığına dair bir hesaplama yapılmıştır. Bu hesaplama Ortalama Göreli Sapma olarak adlandırılmakta olup denklem 4.1'de gösterilmektedir. (Öztekin, 2009)

$$OGS = \frac{(\mathrm{Hi} - \mathrm{Ui})\mathrm{x}100}{\mathrm{Ui}} \tag{4.1}$$

Burada Ui değeri Taillard veri setinde o problem için en üst değeri temsil etmektedir. Hi değeri ise o problem için bulduğumuz Cmax değerini temsil etmektedir. R değeri problemin kaç tekrarla çözüldüğünü göstermektedir. Yani bulacağımız fark değeri Taillard'ın hedeflediği değerden saptığını göstermektedir. Örneğin Taillard'ın 001 nolu probleminde Ui değeri 1278'dir. Bu değere ne kadar yaklaştığımızı fark değerine bakarak anlaşılmıştır.

Karar ağaçları için seçilen birinci kriter standart sapmadır. Standart sapma bir veri topluluğunun nasıl yayılım gösterdiğini özetleyen bir ölçüttür. Eğer verinin çoğunluğunun ortalamaya yakın olması halinde standart sapma değeri küçük olacak, eğer verinin büyük çoğunluğu ortalamadan uzakta bulunuyorlarsa standart sapma değeri büyük olur. Veri değerlerinin aynı olması halinde ise standart sapma değeri sıfırdır.

$$SS = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}} \tag{4.2}$$

x veri kümesindeki herhangi bir değeri, ¯x veri kümesinin ortalamasını, n ise veri adedini göstermektedir.

Karar ağaçları için seçilen ikinci kriter çarpıklıktır. Çarpıklık bir reel-değerli veri kümesinin olasılık dağılımının simetrik olup olmadığının ölçülmesidir. Başka bir deyişle bir dağılımın ortalaması çevresindeki asimetri oranını gösterir.

$$\frac{n}{(n-1)(n-2)}\sum\left(\frac{x_j-\bar{x}}{s}\right) \tag{4.3}$$

Palmer yöntemi için 100 adet problem çalıştırılmış; tamamlama süresi, çalışma süresi(sn.) ve Ortalama Göreli Sapma değerlerinin kendi içinde ortalamaları hesaplanıp Çizelge 4.1'de gösterilmiştir.

Çizelge 4.1. Palmer yöntemi için sonuçlar

Problem	Palmer C_{max} (ort.)	OGS(ort.)	Süre(ort.)
20x5	1354,7	10,87	0,00
20x10	1745,9	15,28	0,00
20x20	2601	16,34	0,00
50x5	2882,4	5,34	0,00
50x10	3400,4	14,03	0,00
50x20	4306,6	16,11	0,01
100x5	5370,2	2,38	0,00
100x10	6138	9,13	0,01
100x20	7218,3	14,61	0,02
200x10	11207,1	5,04	0,02
Ort.	***4622,46***	***10,91***	***0,01***

CDS yöntemi için 100 adet problem çalıştırılmış; tamamlama süresi, çalışma süresi(sn.) ve Ortalama Göreli Sapma değerlerinin kendi içinde ortalamaları hesaplanıp, Çizelge 4.2'de gösterilmiştir.

Çizelge 4.2. CDS yöntemi için sonuçlar

Problem	CDS C_{max} (ort.)	OGS(ort.)	Süre(ort.)
20x5	1338,40	9,63	0,00
20x10	1695,80	12,03	0,00
20x20	2445,10	9,39	0,02
50x5	2910,80	6,38	0,00
50x10	3369,50	12,97	0,01
50x20	4227,00	13,94	0,05
100x5	5506,30	5,00	0,01
100x10	6142,40	9,16	0,02
100x20	7134,00	13,28	0,09
200x10	11463,50	7,45	0,05
Ort.	***4623,28***	***9,92***	***0,03***

NEH yöntemi için 100 adet problem çalıştırılmış; tamamlama süresi, çalışma süresi(sn.) ve Ortalama Göreli Sapma değerlerinin kendi içinde ortalamaları hesaplanıp, Çizelge 4.3'te gösterilmiştir.

Çizelge 4.3. NEH yöntemi için sonuçlar

Problem	NEH C_{max} (ort.)	OGS(ort.)	Süre(ort.)
20x5	1261,1	3,30	0,02
20x10	1583,1	4,60	0,05
20x20	2318,7	3,73	0,10
50x5	2756,1	0,73	0,40
50x10	3134,8	5,07	0,74
50x20	3956,9	6,65	1,31
100x5	5272,3	0,53	2,87
100x10	5752	2,21	4,57
100x20	6634,4	5,34	9,42
200x10	10804,2	1,26	36,94
Ort.	***4347,36***	***3,34***	***5,64***

Genetik Algoritma yöntemi için 100 adet problem çalıştırılmış ve tamamlama süresi, çalışma süresi(sn.) ve Ortalama Göreli Sapma değerlerinin kendi içinde ortalamaları hesaplanıp Çizelge 4.4'te gösterilmiştir.

Çizelge 4.4. Genetik Algoritma yöntemi için sonuçlar

Problem	GA-Pro C_{max} (ort.)	OGS(ort.)	Süre(ort.)
20x5	1243,5	1,75	36,20
20x10	1577,3	4,25	64,76
20x20	2312,4	3,46	127,64
50x5	2758	0,80	249,85
50x10	3116,1	4,47	440,25
50x20	3934,9	6,05	873,93
100x5	5274,6	0,57	1206,35
100x10	5761,1	2,37	2001,07
100x20	6630,6	5,28	3752,86
200x10	10818,2	1,39	8334,73
Ort.	***4342,67***	***3,04***	***1708,77***

Hibrit-GA yöntemi için 100 adet problem çalıştırılmış ve tamamlama süresi, çalışma süresi(sn.) ve Ortalama Göreli Sapma değerlerinin kendi içinde ortalamaları hesaplanıp Cmax, çalışma süresi ve Ortalama Göreli Sapma değerleri hesaplanıp Çizelge 4.5'te gösterilmiştir.

Çizelge 4.5. Hibrit-GA yöntemi için sonuçlar

Problem	Hibrit-GA C_{max} (ort.)	OGS(ort.)	Süre(ort.)
20x5	1244,5	1,91	35,86
20x10	1568,3	3,62	64,15
20x20	2294,8	2,67	127,34
50x5	2749,6	0,49	255,22
50x10	3104,7	4,07	445,24
50x20	3927,5	5,86	885,22
100x5	5264,8	0,39	1146,89
100x10	5727,8	1,78	1841,71
100x20	6604	4,86	3660,22
200x10	10797,2	1,19	8840,12
Ort.	***4328,32***	***2,68***	***1730,20***

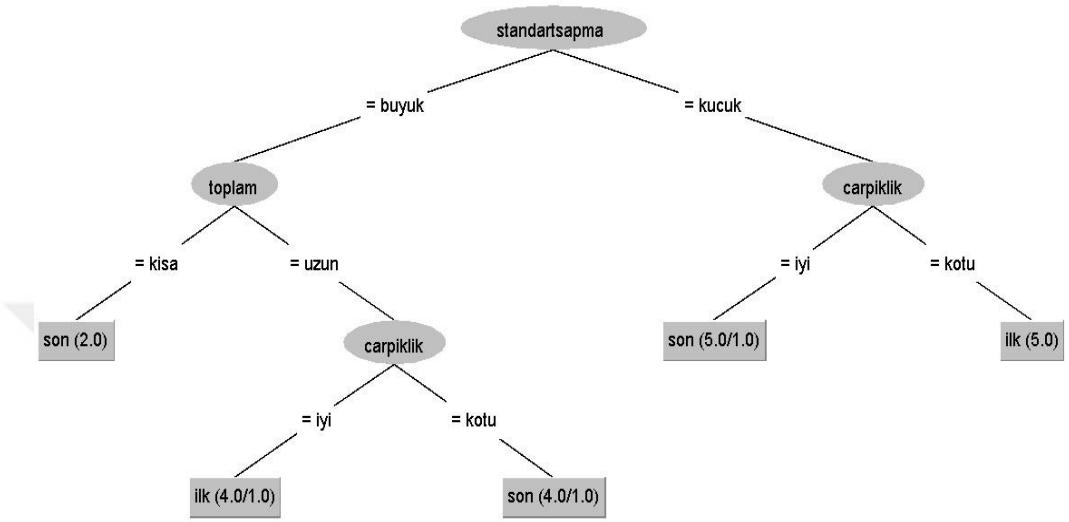
Veri madenciliği aşamasında ise tai010, tai020, …, tai100. problemlerin üst değerlerinden yararlanarak seçilen iş süresi, standart sapma, iş sırası ve çarpıklık öznitelikleriyle karar ağaçları yöntemiyle kurallar oluşturulmuştur. Bir probleme ait şu ana kadar yapılan en iyi çözüm Taillard'ın sitesinden alınmış, bu probleme ait 20x5'lik matristen yararlanarak 4 ayrı hesaplama yapılmıştır. İlk hesaplamada bir işe ait toplam süre hesaplanıp buna iş süresi adı verilmiştir. İş süresi hesaplaması sonucu çıkan 20 değer küçükten büyüğe sıralanmış ilk 10'u kısa, son 10 tanesi uzun olarak adlandırılmıştır. İkinci hesaplamada bir iş için standart sapma değeri hesaplanıp küçükten büyüğe sıralanmış ilk 10'u küçük, son 10 tanesi büyük olarak adlandırılmıştır. Üçüncü hesaplamada aynı işler için çarpıklık değeri hesaplanıp küçükten büyüğe sıralanmış ilk 10'u iyi, son 10 tanesi kötü olarak adlandırılmıştır. Son olarak işlere ait sıralama hesaplanıp iş sırası olarak adlandırılmış küçükten büyüğe sıralanarak ilk 10'u ilk son 10 tanesi son olarak isimlendirilmiştir. Tüm bu işlemler diğer 20, 30, 40, …, 100. problem için de uygulanmıştır.

Tai010 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	27	92	75	94	18	41	37	58	56	20	2	39	91	81	33	14	88	22	36	65
	79	23	66	5	15	51	2	81	12	40	59	32	16	87	78	41	43	94	1	93
	22	93	62	53	30	34	27	30	54	77	24	47	39	66	41	46	24	23	68	50
	93	22	64	81	94	97	54	82	11	91	23	32	26	22	12	23	34	87	59	2
	38	84	62	10	11	93	57	81	10	40	62	49	90	34	11	81	51	21	39	27
Toplam	259	314	329	243	168	316	177	332	143	268	170	199	262	290	175	205	240	247	203	237
İŞ SIRASI	12	13	16	19	14	3	2	6	20	4	1	7	10	11	17	5	18	9	8	15
Sıralama	11	7	6	10	16	8	12	19	18	13	14	1	2	5	20	3	15	17	4	9
sapma	28,743	33,0539	4,83322	36,1143	30,8584	26,5511	20,006	20,3234	21,5741	26,2496	23,0391	7,19444	31,9537	25,7138	24,4704	23,143	21,9363	33,6369	23,1569	31,2192
çarpıklık	0,57395	-0,56959	1,75108	-0,08052	2,01431	0,43683	-0,79442	-1,37808	0,60858	0,36532	0,06041	0,16198	0,38047	-0,41055	1,09312	0,92867	1,33693	0,62669	-0,84995	-0,01742
iş süresi	uzun	uzun	uzun	uzun	kisa	uzun	kisa	uzun	kisa	uzun	kisa	kisa	uzun	uzun	kisa	kisa	kisa	uzun	kisa	kisa
standart sapma	buyuk	buyuk	kucuk	buyuk	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk	kucuk	buyuk
iş sırası	son	son	son	son	son	ilk	ilk	ilk	son	ilk	ilk	ilk	ilk	son	son	ilk	son	ilk	ilk	son
çarpıklık	iyi	kotu	iyi	kotu	iyi	iyi	kotu	kotu	iyi	kotu	kotu	kotu	iyi	kotu	iyi	iyi	iyi	iyi	kotu	kotu

Şekil 4.1. Tai010 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %85 olarak elde edilmiştir. 20 işten 17'si doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



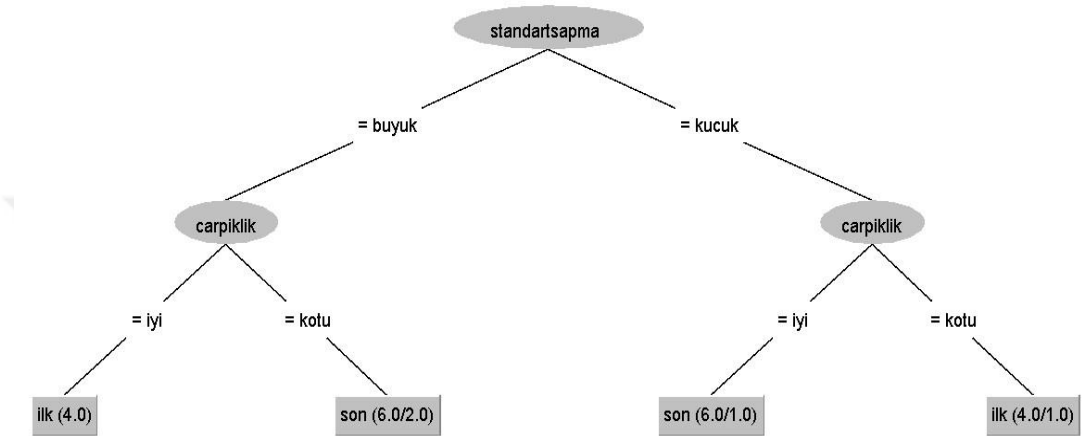
Şekil 4.2. Tai010 karar ağacı

Tai020 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	74	26	84	41	3	46	39	90	62	63	74	4	37	75	72	30	43	95	41	97
	70	92	36	70	37	36	42	34	29	80	91	56	71	14	31	75	80	73	32	56
	84	20	64	44	69	63	94	86	66	97	25	90	12	62	83	32	12	38	22	92
	63	61	46	6	83	49	78	84	97	56	76	9	38	21	80	47	24	95	80	65
	72	91	21	70	12	67	65	90	47	82	62	3	84	30	91	5	11	41	14	28
	78	58	53	17	36	3	83	91	73	81	74	2	31	37	78	74	51	41	78	26
	33	70	10	45	70	2	92	50	54	64	44	4	99	43	43	11	60	53	45	63
	87	20	71	95	71	99	79	19	41	74	8	55	87	87	37	52	36	6	42	30
	3	86	3	89	57	27	28	88	48	26	58	6	33	3	25	61	93	69	53	47
	28	36	39	63	66	44	18	67	17	84	9	33	80	27	42	60	96	81	31	88
Toplam	592	560	427	540	504	436	618	699	534	707	521	262	572	399	582	447	506	592	438	592
İŞ SIRASI	19	17	12	11	1	6	8	5	18	9	20	2	4	16	14	13	7	10	3	15
Sıralama	**5**	**12**	**19**	**13**	**8**	**6**	**17**	**7**	**10**	**18**	**4**	**3**	**16**	**15**	**20**	**14**	**2**	**9**	**1**	**11**
sapma	26,5962	27,4554	25,0042	27,4263	25,7146	27,8216	26,3431	25,1454	21,5926	18,7566	27,8907	29,4068	28,558	25,7117	23,5236	23,2897	29,7463	27,0289	20,6	25,4354
çarpıklık	-1,10828	-0,05102	-0,03189	-0,25553	-0,74676	0,25545	-0,40495	-1,10017	0,32942	-1,24589	-0,50047	1,15876	-0,03358	0,56541	-0,02476	-0,42039	0,24357	-0,40055	0,64609	0,14214
iş süresi	uzun	uzun	kisa	uzun	kisa	kisa	uzun	uzun	kisa	uzun	kisa	kisa	uzun	kisa	uzun	kisa	kisa	uzun	kisa	uzun
standart s	buyuk	buyuk	kucuk	buyuk	kucuk	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk	buyuk	kucuk	kucuk
iş sırası	son	son	son	son	ilk	ilk	ilk	ilk	son	ilk	son	ilk	ilk	son	son	son	ilk	ilk	ilk	son
carpıklık	kotu	kotu	ivi	kotu	kotu	ivi	kotu	kotu	ivi	kotu	kotu	ivi	ivi	ivi	ivi	kotu	ivi	kotu	ivi	ivi

Şekil 4.3. Tai020 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %80 olarak elde edilmiştir. 20 işten 16'si doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



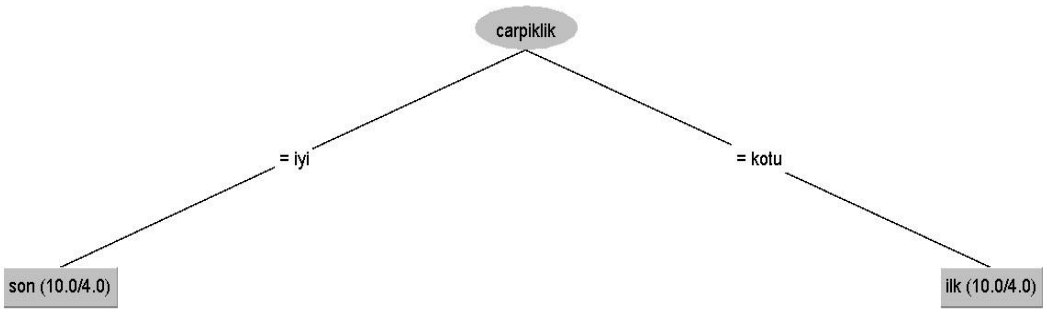
Şekil 4.4. Tai020 karar ağacı

Tai030 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	55	56	21	77	83	43	11	64	59	37	96	27	9	63	82	31	27	67	82	65
	74	41	10	36	69	1	21	72	37	54	88	65	97	71	13	90	88	39	46	89
	99	82	14	14	63	67	16	46	28	91	69	34	9	7	43	62	29	70	28	62
	34	67	16	83	64	32	43	67	37	79	31	41	3	11	71	21	26	98	39	73
	59	33	71	49	22	6	68	14	36	17	30	83	94	53	30	74	24	47	76	28
	20	35	25	81	19	4	42	58	18	19	73	14	44	13	11	55	91	51	42	47
	94	43	68	77	42	75	17	85	54	83	30	52	89	54	74	86	27	22	24	11
	12	8	15	21	75	84	29	15	26	83	38	32	13	10	23	28	42	68	78	39
	76	4	45	25	23	77	3	20	49	35	29	59	14	94	11	17	80	70	20	25
	12	78	58	13	22	53	92	7	62	94	8	81	5	24	98	85	75	57	37	96
	97	44	93	11	74	69	60	95	83	97	37	86	61	68	87	19	12	39	85	6
	50	71	27	86	24	58	20	12	61	57	34	64	43	94	37	13	28	2	82	10
	5	87	66	33	2	32	43	16	96	22	17	75	31	62	76	30	8	34	35	73
	76	5	59	52	16	14	3	1	24	47	19	76	13	67	7	7	14	99	5	11
	84	21	4	37	72	86	17	96	38	43	13	81	12	57	62	87	90	94	94	81
	4	24	88	53	7	35	45	45	43	31	58	88	52	99	87	47	32	24	99	22
	75	39	38	72	26	50	83	15	71	13	94	16	28	17	95	65	84	93	98	65
	79	35	97	68	93	29	94	14	84	29	35	13	51	49	40	11	98	61	19	53
	18	85	7	2	93	20	50	95	3	30	44	97	13	7	72	49	83	75	63	47
	99	52	21	41	69	88	80	83	4	99	59	47	92	11	73	31	70	15	57	19
Toplam	1122	910	843	931	958	923	837	920	913	1060	902	1131	773	931	1092	908	1028	1125	1109	922
İŞ SIRASI	7	8	1	14	13	4	2	16	18	10	12	3	5	20	9	19	11	6	15	17
Sıralama	3	7	12	6	13	18	1	2	15	10	17	11	5	4	19	8	20	9	16	14
sapma	33,2173	25,7051	29,7343	26,532	29,4566	28,0076	28,6588	33,1858	25,0984	29,1445	26,1398	26,3903	31,9003	30,7823	30,1868	27,6431	31,0538	27,7613	28,8435	27,862
çarpıklık	-0,32059	0,07675	0,4874	-0,00874	0,01807	-0,0682	0,46327	0,21516	0,22797	0,32902	0,68727	-0,28831	0,7367	0,11794	-0,26078	0,31812	0,15476	-0,14348	0,01128	0,13744
iş süresi	uzun	kisa	kisa	uzun	uzun	kisa	kisa	kisa	kisa	uzun	kisa	uzun	kisa	uzun	uzun	kisa	uzun	uzun	uzun	kisa
standart s	buyuk	kucuk	buyuk	kucuk	buyuk	kucuk	kucuk	buyuk	kucuk	buyuk	kucuk	kucuk	buyuk	buyuk	buyuk	kucuk	buyuk	kucuk	buyuk	kucuk
iş sırası	ilk	ilk	ilk	son	son	ilk	ilk	son	son	ilk	son	ilk	ilk	son	ilk	son	son	ilk	son	son
çarpıklık	kotu	kotu	iyi	kotu	kotu	kotu	iyi	iyi	iyi	iyi	iyi	kotu	iyi	kotu	kotu	iyi	iyi	kotu	kotu	iyi

Şekil 4.5. Tai030 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %60 olarak elde edilmiştir. 20 işten 12'si doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



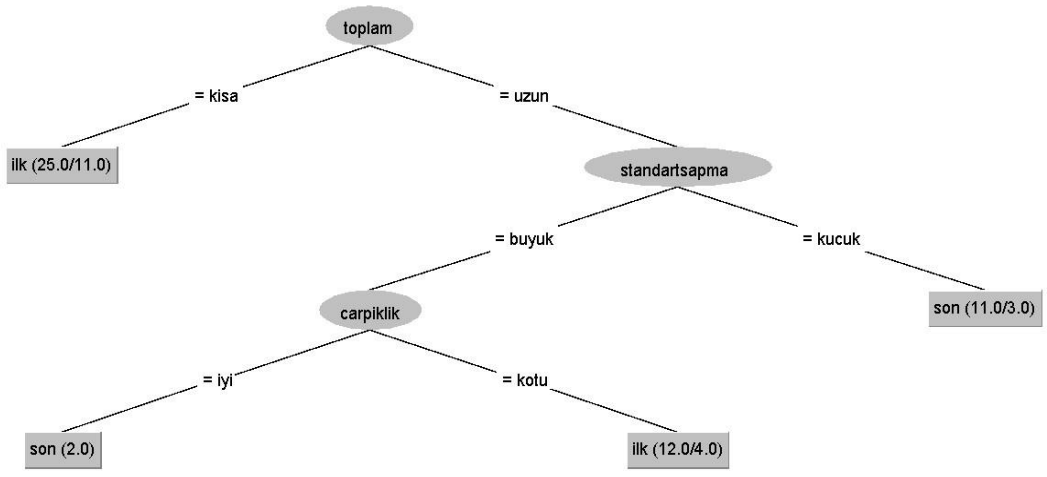
Şekil 4.6. Tai030 karar ağacı

Tai040 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	24	1	57	86	35	9	55	8	93	11	39	10	84	80	59	37	80	69	13	19
	51	57	82	47	82	14	79	57	88	79	22	72	58	78	18	48	72	38	71	82
	8	54	15	47	73	33	73	35	45	75	83	12	59	95	96	75	62	44	43	30
	59	85	73	87	87	89	48	57	59	73	66	95	61	13	79	88	19	18	21	33
	67	99	57	47	93	35	32	77	28	75	95	15	18	97	37	53	97	73	60	39
Toplam	209	296	284	314	370	180	287	234	313	313	305	204	280	363	289	301	330	242	208	203
İŞ SIRASI	10	2	44	43	35	1	31	34	29	25	3	6	45	8	13	16	33	9	24	23
Sıralama	**6**	**2**	**11**	**34**	**35**	**12**	**27**	**14**	**18**	**1**	**24**	**29**	**15**	**30**	**48**	**16**	**41**	**47**	**36**	**43**
sapma	22,2477	33,6714	22,9991	19,3536	20,5718	28,3972	17,0247	23,5151	24,8564	25,8735	27,0924	35,6505	21,2885	30,7675	28,0243	18,6054	26,1457	20,402	22,1233	21,6942
çarpıklık	-0,60605	-0,9143	-1,29348	0,6098	-1,70379	1,54306	-0,20285	-0,69677	-0,04734	-2,20393	-0,2982	0,78507	-1,01235	-1,89762	-0,09172	0,46526	-1,18662	-0,19927	-0,01352	1,73002
iş süresi	kisa	uzun	uzun	uzun	uzun	kisa	uzun	kisa	uzun	uzun	uzun	kisa	uzun	uzun	uzun	uzun	uzun	kisa	kisa	kisa
standart s	kucuk	buyuk	kucuk	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	buyuk	buyuk	buyuk	kucuk	buyuk	buyuk	kucuk	buyuk	kucuk	kucuk	kucuk
iş sırası	ilk	ilk	son	son	son	ilk	son	son	son	ilk	ilk	ilk	son	ilk	ilk	ilk	son	ilk	ilk	ilk
çarpıklık	kotu	kotu	kotu	iyi	kotu	iyi	kotu	kotu	iyi	kotu	kotu	iyi	kotu	kotu	kotu	iyi	kotu	kotu	iyi	iyi

Şekil 4.7. Tai040 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %64 olarak elde edilmiştir. 50 işten 32'si doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



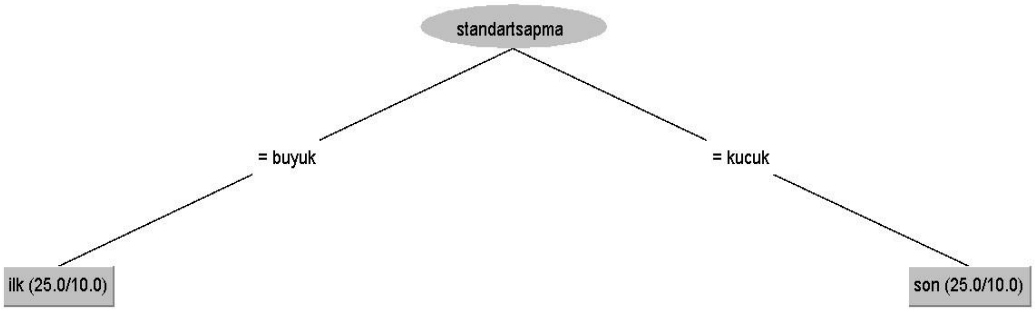
Şekil 4.8. Tai040 karar ağacı

Tai050 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	18	77	73	91	96	10	58	70	42	21	11	88	91	78	70	53	51	96	27	47
	97	49	33	48	68	36	66	6	54	16	7	35	64	18	8	61	41	65	71	34
	14	17	31	37	85	84	7	54	40	95	97	44	4	46	10	63	76	23	95	73
	74	29	77	14	30	14	48	1	4	71	54	30	42	43	33	32	12	68	8	82
	75	43	70	11	64	88	94	9	29	58	90	78	63	34	71	49	69	97	10	43
	27	98	62	17	35	71	94	58	77	29	61	41	47	2	16	41	59	27	18	33
	69	28	95	38	89	19	6	52	70	4	81	1	36	30	13	90	30	66	28	36
	65	21	60	72	4	85	47	80	71	26	73	36	38	13	66	82	2	76	29	90
	89	41	16	7	84	53	56	71	40	17	33	3	51	76	2	63	84	55	3	82
	83	3	99	87	60	92	75	53	30	59	95	85	63	21	78	72	87	52	39	93
Toplam	611	406	616	422	615	552	551	454	457	396	602	441	499	361	367	606	511	625	328	613
İŞ SIRASI	11	48	33	39	36	10	24	12	29	3	20	49	44	50	8	13	5	22	43	15
Sıralama	**49**	**38**	**10**	**27**	**17**	**28**	**42**	**15**	**39**	**6**	**1**	**8**	**16**	**47**	**20**	**44**	**26**	**24**	**29**	**11**
sapma	28,6756	27,1006	26,0546	29,996	28,384	31,3394	28,9671	27,6413	21,4991	27,8359	31,7484	29,4158	21,7828	24,056	29,3157	16,9068	28,0373	23,517	27,7121	23,5544
çarpıklık	-0,70573	0,95662	-0,34576	0,52796	-0,7958	-0,29814	-0,50553	-0,67646	-0,23486	0,73991	-0,59471	0,17878	-0,28527	0,65416	0,31155	0,09321	-0,45748	-0,24257	1,33426	0,05969
iş süresi	uzun	kisa	uzun	kisa	uzun	uzun	uzun	kisa	kisa	kisa	uzun	kisa	kisa	kisa	kisa	uzun	uzun	uzun	kisa	uzun
standart s	buyuk	buyuk	kucuk	buyuk	buyuk	buyuk	buyuk	buyuk	kucuk	buyuk	buyuk	buyuk	kucuk	kucuk	buyuk	kucuk	buyuk	kucuk	buyuk	kucuk
iş sırası	ilk	son	son	son	son	ilk	ilk	ilk	son	ilk	ilk	son	son	son	ilk	ilk	ilk	ilk	son	ilk
çarpıklık	kotu	iyi	kotu	iyi	kotu	kotu	kotu	kotu	kotu	iyi	kotu	iyi	kotu	iyi	iyi	iyi	kotu	kotu	iyi	iyi

Şekil 4.9. Tai050 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %60 olarak elde edilmiştir. 50 işten 30'u doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



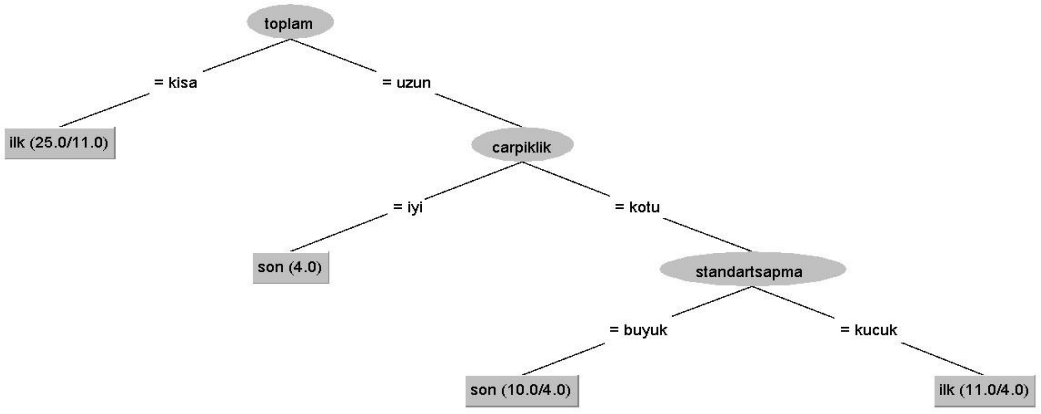
Şekil 4.10. Tai050 karar ağacı

Tai060 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	4	20	62	71	69	87	96	25	98	41	99	1	94	91	36	71	28	50	11	14
	44	54	7	54	61	18	45	64	81	57	13	45	60	12	89	20	47	81	11	56
	15	80	61	37	72	73	55	19	80	48	10	17	67	21	9	79	89	20	94	47
	24	75	31	34	87	88	62	59	39	4	74	90	29	99	74	4	86	22	10	96
	58	55	34	66	67	70	16	25	34	1	18	2	86	34	82	55	59	21	31	53
	53	26	89	34	29	32	28	46	49	55	8	34	40	48	68	59	87	1	87	91
	80	35	39	85	45	13	81	78	25	81	21	87	40	30	11	76	79	75	77	17
	56	8	1	51	73	4	58	24	69	39	33	16	75	19	1	30	44	8	45	19
	11	64	32	10	37	86	35	41	7	56	10	15	25	15	54	3	43	10	27	10
	64	94	23	88	52	23	26	9	63	24	8	32	97	17	76	45	28	89	75	71
	50	83	82	78	44	23	66	24	56	92	30	31	88	85	50	34	91	55	7	49
	22	1	88	69	50	79	59	18	98	21	35	94	57	71	76	97	7	87	89	53
	50	62	28	4	11	13	46	23	41	74	15	67	39	4	13	43	84	41	33	41
	24	2	51	49	39	62	85	70	86	18	90	36	51	20	23	17	23	47	66	3
	56	43	63	60	3	89	52	42	24	53	77	12	71	14	62	75	97	3	54	41
	59	41	64	30	51	16	64	72	68	31	93	86	91	34	6	27	62	35	57	67
	72	81	8	73	93	38	85	94	88	69	90	26	23	44	97	67	53	75	69	44
	41	80	74	15	89	24	36	86	98	51	32	92	51	98	88	1	13	50	80	56
	18	39	96	18	16	48	22	53	71	24	17	47	44	62	4	85	74	59	41	13
	90	34	66	10	95	51	15	4	66	24	53	87	26	64	13	16	23	61	78	58
Toplam	891	977	999	936	1083	937	1032	876	1241	863	826	917	1154	882	932	904	1117	890	1042	899
İŞ SIRASI	11	7	13	50	32	43	40	4	9	27	12	2	35	6	39	19	33	25	3	21
Sıralama	**33**	**12**	**19**	**8**	**22**	**14**	**2**	**50**	**9**	**40**	**1**	**11**	**3**	**36**	**34**	**32**	**25**	**47**	**16**	**29**
sapma	23,39973	27,78718	28,15755	26,07604	26,35009	29,14322	23,38033	26,1641	26,27256	24,31106	32,22127	32,1283	24,09793	30,43008	32,93691	29,2243	28,22813	27,87203	28,49017	25,39975
çarpıklık	-0,008323	-0,188078	-0,114238	-0,106933	-0,217127	0,183125	0,149765	0,360239	-0,401167	0,153825	0,694233	0,326335	0,198276	0,596499	-0,03435	0,047115	-0,158226	-0,010368	-0,23092	0,155053
iş süresi	kisa	kisa	kisa	kisa	uzun	kisa	uzun	kisa	uzun	kisa	kisa	kisa	uzun	kisa	kisa	kisa	uzun	kisa	uzun	kisa
standart sa	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	kucuk	kucuk	kucuk	kucuk	buyuk	buyuk	kucuk	buyuk	buyuk	buyuk	buyuk	buyuk	buyuk	kucuk
iş sırası	ilk	ilk	ilk	son	son	son	son	ilk	ilk	son	ilk	ilk	son	ilk	son	ilk	son	ilk	ilk	ilk
çarpıklık	iyi	kotu	kotu	kotu	kotu	iyi	iyi	iyi	kotu	iyi	iyi	iyi	iyi	iyi	iyi	iyi	kotu	iyi	kotu	iyi

Şekil 4.11. Tai060 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %62 olarak elde edilmiştir. 50 işten 31'i doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



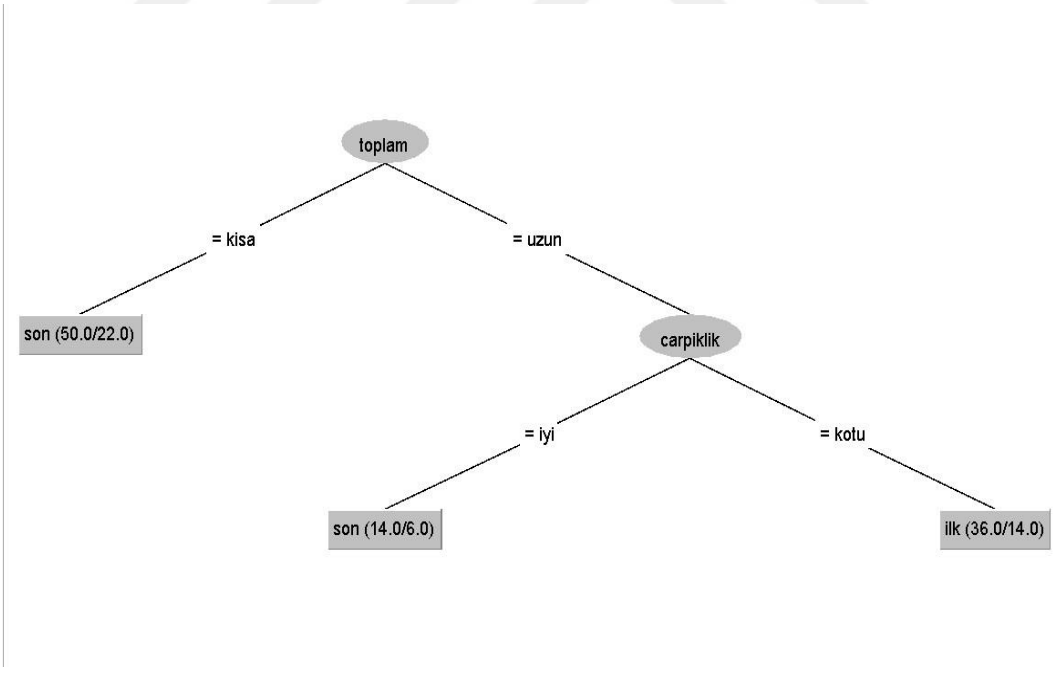
Şekil 4.12. Tai060 karar ağacı

Tai070 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	88	11	99	20	63	67	79	91	13	77	13	52	84	73	61	27	67	5	3	13
	43	9	81	83	90	44	60	70	85	11	88	44	97	96	28	93	28	95	94	6
	2	10	40	22	98	85	11	94	49	57	47	65	22	21	18	72	72	25	41	31
	95	10	84	34	26	76	32	50	44	36	18	25	28	81	57	48	79	14	88	8
	20	5	96	32	33	79	35	4	22	13	64	9	55	92	85	20	86	57	49	61
Toplam	248	45	400	191	310	351	217	309	213	194	230	195	286	363	249	260	332	196	275	119
İŞ SIRASI	27	100	79	57	66	51	88	69	55	53	75	46	62	70	22	10	12	13	5	4
Sıralama	**24**	**55**	**31**	**20**	**19**	**66**	**56**	**77**	**22**	**16**	**99**	**17**	**18**	**86**	**88**	**48**	**43**	**83**	**93**	**64**
sapma	36,6639	2,09762	21,1376	23,0512	29,0448	14,3304	23,6356	32,9751	25,0647	25,4433	28,1496	19,8293	29,6607	27,0451	24,0948	27,3715	20,2445	32,9751	33,3047	20,5854
çarpıklık	0,09756	-1,74437	-1,6725	1,93081	-0,02964	-1,40811	0,30722	-1,13009	0,79169	0,41797	0,29448	-0,3765	0,15735	-1,75856	0,06127	0,41383	-1,68592	1,00207	-0,38991	1,40157
iş süresi	kisa	kisa	uzun	kisa	uzun	uzun	kisa	uzun	kisa	kisa	kisa	kisa	uzun	uzun	kisa	uzun	uzun	kisa	uzun	kisa
standart s	buyuk	kucuk	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	kucuk	buyuk	buyuk	kucuk	buyuk	buyuk	kucuk	buyuk	kucuk	buyuk	buyuk	kucuk
iş sırası	ilk	son	son	son	son	son	son	son	son	son	son	ilk	son	son	ilk	ilk	ilk	ilk	ilk	ilk
çarpıklık	iyi	kotu	kotu	iyi	iyi	kotu	iyi	kotu	iyi	iyi	iyi	kotu	iyi	kotu	iyi	iyi	kotu	iyi	kotu	iyi

Şekil 4.13. Tai070 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %58 olarak elde edilmiştir. 100 işten 58'i doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



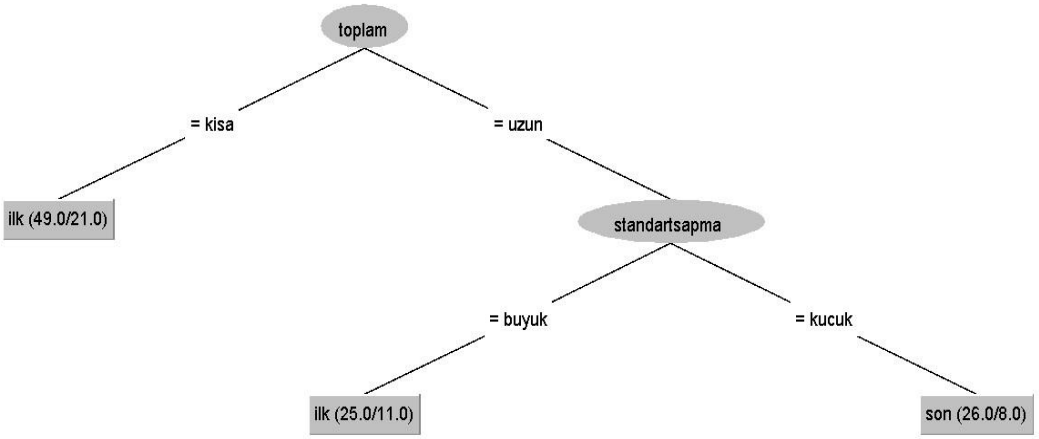
Şekil 4.14. Tai070 karar ağacı

Tai080 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	80	65	79	47	17	93	56	24	22	40	71	96	59	6	49	63	45	61	33	29
	13	20	78	41	80	12	48	56	95	66	27	67	93	43	99	4	27	18	72	12
	14	17	56	83	77	39	24	63	18	85	30	51	44	95	23	67	75	75	62	71
	90	52	6	48	74	63	46	57	17	18	47	71	92	4	46	60	32	38	46	46
	46	50	52	73	58	19	51	11	70	24	62	68	2	20	38	18	18	53	66	75
	73	7	55	74	30	15	25	18	74	45	99	49	2	69	18	33	38	70	22	19
	84	81	92	19	83	3	71	18	10	34	71	89	52	47	8	91	68	58	63	44
	69	91	39	9	36	15	71	55	21	86	82	67	47	32	97	61	29	49	54	92
	35	23	57	85	59	80	81	45	23	43	43	16	70	22	26	8	13	67	8	85
	77	27	24	70	37	90	10	74	63	3	78	69	90	41	67	28	73	55	45	18
Toplam	581	433	538	549	551	429	483	421	413	444	610	643	551	379	471	433	418	544	471	491
İŞ SIRASI	32	63	75	33	55	3	86	7	16	67	19	78	41	25	27	87	6	77	23	17
Sıralama	**57**	**97**	**6**	**71**	**66**	**17**	**8**	**92**	**63**	**37**	**80**	**76**	**33**	**87**	**52**	**9**	**20**	**28**	**11**	**89**
sapma	27,4316	27,4192	24,6812	25,1295	22,4341	33,4648	21,8451	21,2153	29,1206	26,0814	22,3875	21,1047	31,7662	26,685	30,1146	27,5465	21,5907	15,8379	19,6543	28,2292
çarpıklık	-0,68884	0,47127	-0,41774	-0,60289	-0,31007	0,44063	-0,27706	-0,17696	0,6931	0,33375	-0,07001	-0,89812	-0,55792	0,84739	0,70938	0,08021	0,48636	-1,13209	-0,75844	0,19188
iş süresi	uzun	kisa	uzun	uzun	uzun	kisa	kisa	kisa	kisa	kisa	uzun	uzun	uzun	kisa	kisa	kisa	kisa	uzun	kisa	kisa
standart s	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	kucuk	kucuk	kucuk	buyuk	buyuk	buyuk	buyuk	kucuk	kucuk	kucuk	buyuk
iş sırası	ilk	son	son	ilk	son	ilk	son	ilk	ilk	son	ilk	son	ilk	ilk	ilk	son	ilk	son	ilk	ilk
çarpıklık	kotu	iyi	kotu	kotu	kotu	iyi	kotu	kotu	iyi	iyi	kotu	kotu	kotu	iyi	iyi	iyi	iyi	kotu	kotu	iyi

Şekil 4.15. Tai080 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %60 olarak elde edilmiştir. 100 işten 60'ı doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



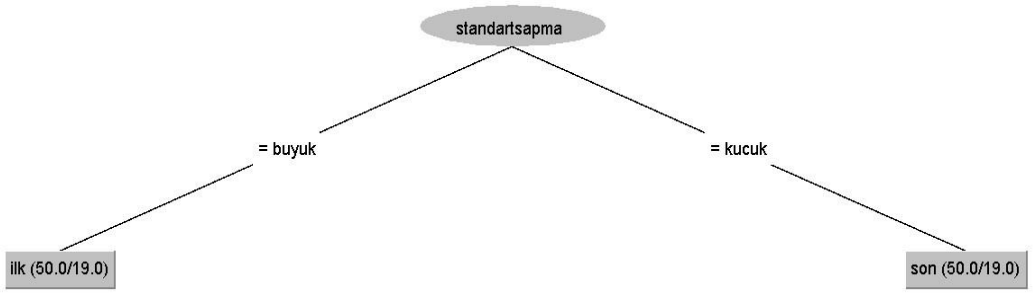
Şekil 4.16. Tai080 karar ağacı

Tai090 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	50	69	14	61	88	5	52	87	92	78	25	40	79	79	24	91	3	19	91	27
	43	66	85	81	68	48	63	73	92	63	42	72	71	46	56	79	59	55	91	25
	2	38	79	78	8	87	75	53	96	27	25	41	99	25	16	27	92	37	74	78
	37	31	36	73	88	9	51	24	65	99	22	34	44	43	53	44	10	79	73	30
	66	46	31	21	54	71	42	81	8	32	12	41	12	71	31	60	25	18	81	7
	3	3	42	46	84	38	92	86	93	22	26	93	64	32	75	63	5	55	14	63
	14	68	51	62	67	21	94	64	43	12	1	86	84	87	80	38	15	4	13	46
	7	95	72	16	15	69	4	31	16	26	15	28	24	55	19	15	20	63	43	76
	53	63	79	98	11	89	22	50	86	84	56	32	4	87	58	3	99	81	51	35
	96	28	67	99	97	35	26	61	11	95	3	24	65	76	58	19	42	28	7	48
	30	34	90	42	56	24	36	4	79	60	74	94	53	74	54	48	80	71	91	37
	57	11	72	35	66	56	36	16	96	97	59	65	87	12	13	5	6	55	95	82
	64	13	43	55	58	8	82	37	76	90	42	35	10	15	21	47	30	12	83	2
	70	86	95	81	48	38	1	97	24	69	61	84	46	92	12	97	65	14	90	80
	65	75	38	51	65	87	86	35	58	65	72	24	70	41	73	41	80	14	69	23
	92	1	39	97	92	59	60	1	78	3	65	80	98	76	40	58	36	23	22	21
	28	48	54	78	29	29	63	15	12	48	22	39	83	15	73	89	27	23	11	8
	39	60	80	99	48	97	28	84	11	98	37	93	4	73	61	84	16	80	14	70
	75	81	18	12	29	83	78	16	73	38	18	43	38	83	83	6	45	91	39	77
	16	60	31	34	94	94	18	30	84	95	37	80	37	31	26	30	32	49	33	30
Toplam	907	976	1116	1219	1165	1047	1009	945	1193	1201	714	1128	1072	1113	926	944	787	871	1085	865
İŞ SIRASI	1	64	14	79	88	87	38	98	97	24	7	62	65	61	42	86	68	40	66	89
Sıralama	**1**	**47**	**30**	**45**	**98**	**97**	**11**	**99**	**90**	**52**	**54**	**51**	**73**	**3**	**83**	**50**	**95**	**72**	**80**	**57**
sapma	27,60847	27,34337	24,13421	27,59434	27,56968	30,20476	28,02584	29,64772	32,74794	31,10703	21,90457	25,3464	30,4046	26,52032	23,65608	29,27729	29,33986	26,91185	31,93568	26,10148
çarpıklık	0,046073	-0,25639	-0,02445	-0,218	-0,39055	-0,03628	-0,09768	0,119414	-0,54803	-0,29682	0,277705	0,276026	-0,25816	-0,30732	-0,03053	0,14723	0,698398	0,242516	-0,18698	0,163072
iş süresi	kisa	kisa	uzun	uzun	uzun	uzun	uzun	kisa	uzun	uzun	kisa	uzun	uzun	uzun	kisa	kisa	kisa	kisa	uzun	kisa
standart	kucuk	kucuk	kucuk	kucuk	kucuk	buyuk	buyuk	buyuk	buyuk	buyuk	kucuk	kucuk	buyuk	kucuk	kucuk	buyuk	buyuk	kucuk	buyuk	kucuk
iş sırası	ilk	son	ilk	son	son	son	ilk	son	son	ilk	ilk	son	son	son	ilk	son	son	ilk	son	son
çarpıklık	iyi	kotu	kotu	kotu	kotu	kotu	kotu	iyi	kotu	kotu	iyi	iyi	kotu	kotu	kotu	iyi	iyi	iyi	kotu	iyi

Şekil 4.17. Tai090 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %62 olarak elde edilmiştir. 100 işten 62'si doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



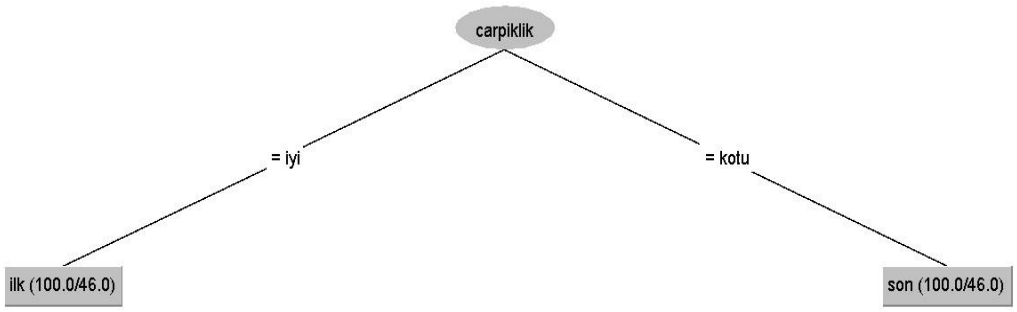
Şekil 4.18. Tai090 karar ağacı

Tai100 problemi için seçilen dört nitelikle hesaplamalar yapılmıştır.

Problem	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
	7	3	45	44	54	36	79	40	10	51	39	45	96	30	89	70	96	32	55	98
	88	75	17	86	77	83	27	65	58	15	28	23	50	73	36	2	46	49	51	44
	69	74	88	93	19	26	92	76	16	16	55	10	71	20	46	43	9	57	85	5
	22	83	19	8	62	40	71	48	49	40	83	57	68	50	99	89	90	40	1	26
	26	5	53	1	56	85	21	8	47	66	12	4	50	44	16	37	73	46	8	88
	60	37	28	70	4	71	65	83	67	93	43	9	75	22	6	49	71	82	72	4
	68	42	14	51	1	60	44	49	89	52	53	35	80	18	46	60	41	36	78	81
	2	62	62	58	14	32	27	52	90	34	90	81	70	37	89	91	75	82	40	32
	34	49	84	38	88	55	2	10	44	75	37	63	47	1	88	97	96	60	14	98
	48	12	62	95	91	52	70	89	18	18	83	4	72	62	77	89	90	63	70	77
Toplam	424	442	472	544	466	540	498	520	488	460	523	331	679	357	592	627	687	547	474	553
İŞ SIRASI	74	52	179	145	143	65	185	23	150	181	47	95	166	4	161	154	167	148	12	147
Sıralama	**148**	**83**	**177**	**14**	**143**	**85**	**149**	**123**	**24**	**127**	**138**	**19**	**165**	**168**	**190**	**152**	**145**	**121**	**60**	**195**
sapma	27,2478	28,3012	25,8333	31,2832	32,7237	19,799	28,124	26,3515	26,9919	25,1714	24,5481	26,1054	14,4738	20,8377	31,7641	28,8827	27,0335	16,6796	29,0041	35,4092
çarpıklık	0,07181	-0,23069	0,21172	-0,37926	-0,12294	0,27695	-0,19016	-0,41882	0,1126	0,41539	0,19987	0,50218	0,14961	0,25729	-0,38029	-0,74352	-1,11264	0,48986	-0,43741	-0,22343
iş süresi	kisa	kisa	kisa	uzun	kisa	uzun	kisa	uzun	kisa	kisa	uzun	kisa	uzun	kisa	uzun	uzun	uzun	uzun	kisa	uzun
standart s	buyuk	buyuk	kucuk	buyuk	buyuk	kucuk	buyuk	kucuk	buyuk	kucuk	kucuk	kucuk	kucuk	kucuk	buyuk	buyuk	buyuk	kucuk	buyuk	buyuk
iş sırası	ilk	ilk	son	son	son	ilk	son	ilk	son	son	ilk	ilk	son	ilk	son	son	son	son	ilk	son
çarpıklık	iyi	kotu	iyi	kotu	kotu	iyi	kotu	kotu	iyi	iyi	iyi	iyi	iyi	iyi	kotu	kotu	kotu	iyi	kotu	kotu

Şekil 4.19. Tai100 hesaplamaları

Ardından bu öznitelikler kullanılarak karar ağacı oluşturulmuştur. Doğruluk oranı %54 olarak elde edilmiştir. 200 işten 108'i doğru sınıflandırılmıştır. Bu değerler sonucunda elde edilen ağaç şekildeki gibidir.



Şekil 4.20. Tai100 karar ağacı

5. SONUÇLAR ve ÖNERİLER

5.1. Sonuçlar

Elde edilen sonuçların güncel literatürle karşılaştırılması için Wang (2017) makalesi ve Öztekin (2009) tezi kullanılmıştır. Ga-Neu yöntemine ailt sonuçlar, Nearchou (2004) sonuçları Zhang ve ark (2010) ve Wang (2017) makalelerinden, PSOPSV, DPSO yöntemlerine ait sonuçlar Öztekin (2009) tezinden alınmıştır. GA-Pro sonuçları ise Proyag (2019) tarafından Github'da paylaşılan genetik algoritma kodları ile elde edilmiştir. 10 adet problemin GA-Neu, PSOPSV, DPSO, GA-Pro ve Tezde önerdiğimiz Hibrit-GA yöntemlerinin tamamlama ve Ortalama Göreli Sapma değerleri 10 kez çalıştırılarak elde edilmiş ve bu sonuçların aritmetik ortalamaları karşılaştırılmış olup Çizelge 5.1'de gösterilmiştir.

Çizelge 5.1. Yöntem sonuçları karşılaştırması

Prob.	Üst değer	GA-Neu	GA-Neu OGS	PSOPSV	PSO PSV OGS	DPSO	DPSO OGS	GA-Pro	GA-Pro OGS	Hibrit-GA	Hibrit-GA OGS
Tai010	1108,0	1135,6	2,49	1135,8	2,51	1131,2	2,09	1132,9	2,25	***1130,5***	2,03
Tai020	1591,0	***1632,8***	2,63	1656,1	4,09	1639,6	3,05	1652,6	3,87	1645,1	3,40
Tai030	2178,0	***2237,5***	2,73	2251,9	3,39	2251,0	3,35	2275,6	4,48	2258,9	3,71
Tai040	2782,0	2815,7	1,21	2795,3	0,48	2801,9	0,72	2790,2	0,29	**2785,0**	0,11
Tai050	3065,0	3225,5	5,24	3229,0	5,35	3257,4	6,28	***3211,1***	4,77	3219,9	5,05
Tai060	3756,0	4008,6	6,73	3976,0	5,86	4046,9	7,74	3976,5	5,87	***3970,9***	5,72
Tai070	5322,0	5372,3	0,95	5344,8	0,43	5377,8	1,05	5350,1	0,53	***5340,0***	0,34
Tai080	5845,0	6056,3	3,62	5962,3	2,01	5988,0	2,45	5940,8	1,64	***5907,1***	1,06
Tai090	6434,0	6910,3	7,40	6862,8	6,66	6929,1	7,70	6719,1	4,43	***6674,4***	3,74
Tai100	10675,0	11025,6	3,28	10907,9	2,18	10926,2	2,35	10826,7	1,42	***10796,5***	1,14
Ort.	4275,6	4442,02	3,628	4412,19	3,296	4434,91	3,678	4387,56	2,955	***4372,83***	***2,63***

20x5 probleminin 10. problemi incelendiğinde en küçük Cmax değerini veren yöntemin Hibrit-GA olduğu görülecektir. 20x10 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren yöntemin GA-Neu olduğu görülecektir. 20x20 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren yöntemin GA-Neu olduğu görülecektir. 50x5 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA yöntem olduğu görülecektir. 50x10 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren yöntemin GA-Pro olduğu görülecektir. 50x20 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA yöntem olduğu görülecektir. 100x5 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA yöntem olduğu görülecektir. 100x10 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA yöntem olduğu görülecektir. 100x20 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA yöntem olduğu görülecektir. 200x10 probleminin 10.problemi incelendiğinde en küçük Cmax değerini veren Hibrit-GA olduğu görülecektir.

Genel olarak incelenirse;

2 problemde GA-Neu, 1 problemde Ga-Pro, 7 problemde Hibrit-GA yöntemi öne çıkmıştır.

En önemli kriter olarak ortalama göreli sapma değerini gösterebiliriz çünkü ortalama göreli sapma değeri Taillard veri setinde belirlenen üst değerden ne kadar uzaklaştığımızı gösteren dolayısıyla çözümün ne kadar sağlıklı olduğunu gösteren bir veridir. Ortalama göreli sapma değerleri incelendiğinde 2 problemde GA-Neu, 7 problemde Hibrit-GA, 1 problemde Ga-Pro yöntemi öne çıkmıştır.

Çizelge 5.2'de, Çizelge 4.1 ile Çizelge 4.5 arasında elde edilen genel sonuçlar verilmiştir.

Çizelge 5.2. OGS ortalama karşılaştırması

Yöntem	OGS (ort)
Palmer	10,91
CDS	9,92
NEH	3,34
GA-Pro	3,04
Hibrit-GA	***2,68***

Buna göre; Palmer, CDS, NEH, GA-Pro ve Hibrit-GA yöntemlerinin 10 problem grubundaki 100 problem için verdiği OGS değerlerinin ortalaması incelendiğinde Hibrit-GA'nın diğer yöntemlere göre daha başarılı olduğu yani Taillard üst değerine en çok yaklaşan yöntem olduğu görülmektedir.

5.2. Öneriler

Algoritmanın daha da geliştirilmesi adına seçim yöntemi olarak farklı yöntemler kullanılabilir ya da farklı popülasyon ve iterasyon sayıları seçilerek problemin tekrar çözülerek daha iyi sonuçlar elde edilebilir.

Karar ağaçları için yeni bazı öznitelikler eklenebilir, ayrıca karar ağaçlarından elde edilen kurallarla yeni kromozomlar üretilip, bu kromozomlar başlangıç popülasyonuna dahil edilerek daha iyi çözümler üretilebilir.

Farklı veri madenciliği teknikleri uygulanıp büyük problem için güncel literatürden daha iyi sonuçlara ulaşılabilir. Farklı metasezgisel yöntemler kullanılarak farklı hibrit yöntemler geliştirilebilir. Çok amaçlı problemler çalışılarak birden fazla kriter aynı problemde iyileştirilebilir.

KAYNAKLAR

AKÇAY, E., 2009. Akış Tipi İş Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemi ile Çok Amaçlı Optimizasyonuna Yönelik Bir Model Önerisi. Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Kocaeli.

AKPINAR, H., 2000. Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği. İÜ İşletme Fakülte Dergisi, 29(17): 1-22.

AKPINAR, H., 2014. Data Veri Madenciliği Veri Analizi. Papatya Yayıncılık Eğitim, İstanbul, 552s.

ALPAYDIN, E., 2000. Zeki Veri Madenciliği. Bilişim 2000 Eğitim Semimeri Notları. www.cmpe.boun.edu.tr/~ethem (Son erişim tarihi: Ocak 2020)

ALLAHVERDİ A., ALDOWAISAN T., 2002. New heuristics to minimize total completion time in m-machine flowshops. International Journal of Production Economics, 77(1): 71-83.

ALLAHVERDİ A., GUPTA J. N. D., ALDOWAISAN T., 1999. A review of scheduling research involving set-up consideration. International Journal of Management Science, 27(2): 219-239.

ALTIPARMAK, F., 1996. Genetik Algoritma ile Haberleşme Şebekelerinin Topolojik Optimizasyonu. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara.

ARORA, J. S., 2004. Introduction to Optimum Design. Elsevier Acedemic Press, Iowa, 880s.

ARTIBA, A., ELMAGHRABY, S.E., 1997. The Production and Scheduling of Production System. Chapman and Hall, New York, 368s.

BAKER, K., 1994. Elements of Sequencing And Scheduling. Amos Tuck School of Business, D. Hanover, NH.

BALİN, A., 2010. Genetik Algoritma ile Hat Dengeleme. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

BASKAK, M., EROL, V., 2004. Sipariş Tipi Atölyelerde İş Sıralama Problemi İçin Bir Genetik Algoritma Uygulaması. Endüstri Mühendisliği Kongresi, Bildiriler Kitabı, s.187-189.

BISHOP, C. M., 1996. Neural Networks for Pattern Recognition. Clarendon Press, Oxford,482s.

BOJARCZUK C. C., LOPES, H. S., 2004. A Constrained-Syntax Genetic Programming System for Discovering Classification Rules: Application to Medical Data Sets. Artificial Intelligence in Medicine, 30(1): 27–48.

BOLAT, B., EROL, K. O., VE İMRAK, C. E., 2004. Genetic algorithms in engineering applications and the Function of operators. Sigma, 4., 265-271.

BORAY, B., 2007. Paralel Tezgahlarda Arıza Halinde Çok Amaçlı Çizelgeleme. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 134s.

BRUCKER, P., 2007. Scheduling Algorithms. Springer-Verlag, Berlin,383s.

CARLIER J., PINSON E., 1989. An Algorithm for Solving Job Shop Scheuling Problem. Management Science, 35(2): 164-176.

CERAN, G., 2006. Esnek akış tipi çizelgeleme problemlerinin veri madenciligi ve genetik algoritma kullanılarak çözülmesi. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya, 134s.

CHEN, C. L., NEPPALLI. R. V., ALJABER, N., 1996. Genetic Algorithms Applied To The Continuous Flow Shop Problem. Computers And Industrial Engineering**.** 30(4): 919-929.

CHENG, R., GEN M. VE TSUJIMURA Y., 1999. A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms, Part II: Hybrid Genetic Search Strategies. Computers and Industrial Engineering, C:36, s.343-364.

CİHANLI, Ö., 2010, Hibrit Akış Tipi Atölye Çizelgeleme. Uludağ Üniversitesi, Fen Bilimleri Enstitüsü , Yüksek Lisans Tezi, Bursa, 69s.

COWLING, P., JOHANSSON, M., 2002. Using real time information for effective dynamic scheduling. European Journal of Operational Research, 139(2): 230-244.

ÇALIŞKAN, F., YÜKSEL, H., DAYIK, M., 2016. Genetik Algoritmaların Tasarım Sürecinde Kullanılması. SDU Teknik Bilimler Dergisi, 6(2):21-27.

ÇEYREKOĞLU, S., 2017. Araç Rotalama Problemine Genetik Algoritma Yaklaşımı ve Örnek Bir Uygulama. İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

ÇERKEZ, H.S., 2003, Müşteri İlişkileri Yönetiminde İş Zekası Ve Veri Madenciliği Yöntemleri. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 112s.

DAĞ, S., 2012. Akış Tipi Çizelgeleme Problemlerinin Sezgisel Yöntemlerle Optimizasyonu. İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, İstanbul, 162s.

DÖYEN, A., 2004. Akış Tipi Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemleri İle Çözümü ve Parametre Optimizasyonu. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya, 88s.

ELMAS, Ç., 2011. Yapay Zeka Uygulamaları. Seçkin Yayıncılık, İstanbul, 448s.

ENGİN, T., 2013. Genetik Algoritma ile Toplu Ulaşım Sistemi Hareket Çizelgesi Optimizasyonu: Çanakkale Örneği. Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, İstanbul, 175s.

ENGİN, O., 2001, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözümün Performansının Artırılmasında Parametre Optimizasyonu. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul, 215s.

ERDEM, A., 2008. Bakım Kısıtı Altında Genetik Algoritmalarla Üretim Çizelgeleme. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, Sakarya.

FRAMINAN, J. M., LEISTEN, R., USANO, R., 2005. Comparison of heuristics for minimisation in permutation flowshops. Computers and Operations Research, 32(5): 1237-1254.

GABER, M.M., 2002. A framework for a scalable distributed data mining model. University of Louisville, Department of Computer Engineering and Computer Science, PhD thesis, Louisville.

GEN, M., CHENG, R., 2000. Genetic algorithm and engineering optimization. John Wily and Sons, New York, 512s.

GOLDBERG, D. E., 1989. Genetic Algorithms in Search Optimization And Machine Learning. Addison Wesley Publishing Company, U.S.A, 412s.

GÖNÜLOL, S., 2009. Gezgin satıcı problemi için veri madenciliği tabanlı sezgisel bir yaklaşım. Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kocaeli, 78s.

GÜLDALI, A., 1990. Seri İş-Akışlı Atölye Çizelgelemesinde Sezgisel Teknikler. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.

HAN, J., KAMBER, M., 2001. Data Mining: Concept and Techniques. Morgan Kaufmann Publishers, San Francisco, 703s.

HAN, J., KAMBER M., 2006. Data Mining Concepts and Techniques(second edition), Morgan Kaufmann, San Francisco, 703s.

İŞLER, M., TOKLU, B., ÇELİK, V., 2009. Öğrenme Etkili Erken/Geç Tamamlanma Çizelgeleme Problemleri İçin bir Literatür Araştırması. Pamukkale Üniversitesi, Mühendislik Bilimleri Dergisi, 15(2): 227-252.

KARABOĞA, D., 2017. Yapay Zekâ Optimizasyon Algoritmaları. Nobel Akademik Yayıncılık, Ankara, 246s.

KARAHOCA, A., 2004. Yapay Sinir Ağları Yardımı ile Telekom Sektöründe Stratejik Veri Madenciliği. www.bilisimhaftasi.org.tr/program_akademik.html (Son erişim tarihi: Ocak 2020)

KOONCE, D.A., TSAI S.-C., 2000, Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. Computers&Industrial Engineering, 38(3):361-374.

KOZA, J.R., 1992. Genetic Programming: on the Programming of Computers by Means of Natural Selection. MIT Press, London, 805s.

KOMOROWSKI, J., ZYTKOW, J., 1997. Principles Of Data Mining And Knowledge Discovery. Springer, Germany, 510s.

KUMAR, S., RAO C. S. P., 2009. Application of antcolony, genetic algorithm and data miningbased techniques for scheduling. Robotics and Computer-Integrated Manufacturing, 25(6):901-908.

LEUNG, J. Y-T., 2004. Handbook of Scheduling Algorithms, Models and Performance Analysis. CRC Press Company, United States of America, 1216s.

LIAW, C.F., 2003. An Efficient Tabu Search Approach for the Two-Machine Preemptive Openshop Scheduling Problem. Computers and Operations Research, 30(14): 2081-2095.

LIPPMANN, R.**,** 1987**.** An introduction to computing with neural nets. IEEE ASSP Magazine, 4(2): 4–22.

LIU J., REEVES C. R., 2001. Constructive and composite heuristic solutions to the P //∑ Ci scheduling problem**.** European Journal of Operational Research, 132(29): 439-452.

MELANIE, M., 1999. An Introduction to Genetic Algorithms (Fifth printing ed. Vol3). A Bradford Book The MIT Press, Cambridge, 221s.

MICHALEWICZ, Z., 1996. Genetic algorithms + data structures = evolution programs (3rd ed.). Berlin: Springer-Verlag, New York, 387s.

NABIYEV, V.V., 2010. Yapay Zeka(3.baskı). Seçkin Yayıncılık, Ankara, 776s.

NAHMIAS, S., 2009. Mathematical Formulation and Scheduling Heurıstics for Cyclic Permutation Flow Shops. Ohio University, The Russ College of Engineering Technolog, PhD Thesis, Ohio, 169s.

NAWAZ, M., ENSCORE, E., HAM, I., 1983. A heuristic algorithm for the m-Machine, n-Job flowshop sequencing problem. International Journal of Management Science, 11(1): 91-95.

NEARCHOU, A. C., 2004. The effect of various operators on the genetic search for large scheduling problems. International Journal of Production Economy, 88(2): 191–203.

OĞUZ, C., ZINDER, Y., DO, V.H., JANIAK, A., LICHTENSTEIN M., 2004. Hybrid flow-shop scheduling problems with multiprocessor task systems. European Journal Of Operational Research, 152(1): 115-131.

OLAFSSON S., LI X., 2010. Learning effective new single machine dispatching rules from optimal scheduling data. Int. J. Production Economics, 128(1): 118-126.

ORALER, G., 1990. Genetik Ders Notları. Gazi Üniversitesi, Fen Fakültesi, Ankara.

ÖZKAN, R., 2003. Tek Modelli Deterministik Montaj Hattı Dengeleme Problemlerine Genetik Algoritma ile Çözüm Yaklaşımı. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 113s.

ÖZNUR, G., KORUKOĞLU, S., 2003. Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama. Yönetim ve Ekonomi: Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 10 (2), 191-208.

ÖZTEKİN, S. M., 2009. Discrete Particle Swarm Optimization And Differential Evolution Algorithms For Flow Shop Scheduling Problems. Fatih University, The Graduate Institute of Sciences and Engineering, Master Thesis, Istanbul, 81s.

PAKSOY, S., 2007. Genetik Algoritma ile Proje Çizelgeleme. Çukurova Üniversitesi, Sosyal Bilimler Enstitüsü, Doktora Tezi, Adana.

PALMER, D., 1965. Sequencing jobs through a multi-stage process in the minimum total time a quick method of obtaining a near optimum. Operations Research Quarterly,16(1): 101-107.

PINEDO, M., 2002. Scheduling Theory, Algorithms and Systems. Prentice Hall(Second Edition), Upper Saddle River, New Jersey, 671s.

PINEDO, M.L., 2008. Scheduling: Theory, Algorithm, and Systems. Springer, New York, 671s.

PROYAG, P., KAUSTAV B., TRIPARNA M., 2019. Master's degree project at St. Xavier's College, Kolkata. Flowshop Project https://github.com/Proyag/FlowShopProject (Son erişim tarihi: Mart 2020)

RAJENDRAN, C., ZIEGLER, H., 2004. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan total flowtime of jobs. European Journal of Operational Research, 155(2): 426-438.

RODAMMER, F. A., K. P. WHITE., 1988. A Recent Survey of Production Scheduling. IEEE Transactions on Systems, Man. And Cybernetics, 18 (6): 841-851.

RUIZ, R., RODRIQUEZ, A.V., 2010. The Hybrid Flow Shop Scheduling Problem. European Journal of Operational Research, 205(1):1-18.

SAFE, M., CARBALLIDO, J., PONZONI, I., BRİGNOLE, N., 2004. On stopping criteria for genetic algorithms. Advances in Artificial Intelligence, 3171(1):405-413

SANTOS, D. L., HUNSUCKER, J. L., DEAL, D. E., 1995. Global lower bounds for flow shops with multiple processors. European Journal of Operational Research, 80(1):112-120.

SANTOS H. G., OCHİ L. S., MARİNHO E. H., DRUMMOND L.M.A., 2006. Combining an evolutionary algorithm with data mining to solve a single-vehicle routing problem. Neurocomputing, 70(1): 70-77.

SASTRY, K., GOLDBERG, D., KENDALL, G., 2005. Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Springer, USA.

SEÇME, G., 2006. Akış Tipi Çizelgeleme Problemlerinin Yapay Sinir Ağları İle Modellenmesi. Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, Kayseri.

SIVANANDAM, S., DEEPA, S., 2008. Introduction to genetic algorithms, Springer, Berlin, 442s.

SHAHZAD, A., 2011. A Single Machine Scheduling Problem with Individual Job Tardiness Based Objectives. http://oro.univ-nantes.fr/sujets-09-10/shahzad.pdf (Son erişim: Mart 2020)

SOYASLAN, G., 2010. Genetik Algoritmalarda Atama Problemi ve Buna Yönelik Bir Uygulaması. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

ŞEKER, Ş., 2007. Araç Rotalama Problemleri ve Zaman Pencereli Stokastik Araç Rotalama Problemine Genetik Algoritma Yaklaşımı. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

ŞEN, Z., 2004. Genetik Algoritmalar ve En İyileme Yöntemleri. Su Vakfı, İstanbul, 142s.

ŞEVKLİ, M., 2005. Atölye Tipi Çizelgeleme Problemlerine Parçacık Sürü Optimizasyonu Yaklaşımı ve Genetik Algoritma Modeli ile Karşılaştırılması. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul, 175s.

TAILLARD, E., 1993. Benchmarks For Basic Scheduling Problems. European Journal of Operational Research, 64(2):278-285. http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html (Son erişim tarihi: Mart 2020)

TANG, L., XUAN, H., LIU, J., 2006. A new Lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time. Computers & Operations Research, 33(11): 3344-3359.

TANTUĞ, A.C., 2002. Veri madenciliği ve demetleme. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul, 73s.

TAŞGETİREN, M.F., LİANG, Y.C., ŞEVKLİ M., GENÇYILMAZ, G., 2007. A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem. European Journal of Operational Research, 177(3): 1930–1947.

TAŞKIN, Ç., GÖKAY, G.E., 2009. Sayısal Yöntemlerde Genetik Algoritmalar. Alfa Aktüel Yayınları, Bursa, 153s.

THABTAH, F., COWLING, P., 2008. Mining the data from a hyperheuristic approach using associative classification. Expert Systems with Applications, 34(2): 1093-1101.

T'KINDT, V., BILLAUT, J.C., 2006. Multicriteria Scheduling :Theory, Models and Algorithms. Springer-Verlag, Berlin, 359s.

TODD, D., P. SEN., 1997. Multiple Criteria Scheduling Using Genetic Algorithms in a Shipyard Environment. Proc. Ninth Int. Conf. Computer Applications in Shipbuilding**,** 234-265

TÜRKELİ, B., 2010. Solution of Bicriteria Open Shop Scheduling Problem with Genetic Algorithm and Tabu Search and Detailed Analysis of Results. Marmara Unıversity, Institute for Graduate Studies in Pure and Applied Sciences, Thesis for The Degree of Master, Istanbul.

VAHAPLAR., 2004. Veri Madenciliği ve Elektronik Ticaret. www.bayar.edu.tr/bid/dokumanlar/inceoglu.doc(Son erişim: Ocak 2020)

WANG, H., WANG, W., SUN, H., CUI, Z., RAHNAMAYAN, S., ZENG, S., 2017. A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems. Soft Comput, 21(15): 4297–4307.

WENG, M. X., 2000. Scheduling Flowshops with Limited Buffer Spaces. Proceedings of the Winter Simulation Conference, Tampa, Florida, s. 1359–1363.

WITTEN, I. H., & FRANK, E.**,** 2005. Data Mining Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, San Francisco, 525s.

YANG, W. H., LIAO, C. J., 1999. Survey of scheduling research involving setup times. International Journal of Systems Science, 30(1): 143-155.

YURTSEVER, U., 2002. Veri Madenciliği ve Uygulaması (Yayınlanmamış Yüksek Lisans Tezi). Sakarya Üniversitesi, Sosyal Bilimler Enstitüsü, Sakarya.

ZARE, H. K., FAKHRZAD, M. B., 2011. Solving flexible flow-shop problem with a hybrid genetic algorithm and data mining: A fuzzy approach. Expert Systems with Applications, 38(6): 7609-7615.

ZHANG, C., NING, J., OUYANG, D., 2010. A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem. Comput Ind Eng, 58(1):1–11

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Ömer KAYA
Uyruğu : Türkiye Cumhuriyeti
Doğum Tarihi ve Yeri : 1989-Şanlıurfa
Telefon : 05448626439
E-mail : okaya1989@gmail.com

EĞİTİM

Derece	Adı,İlçe,İl	Bitirme Yılı
Lise	Orhangazi Lisesi, Şanlıurfa	2006
Üniversite	Mustafa Kemal Üniversitesi,Hatay	2014
Yüksek Lisans	Harran Üniversitesi	2020

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2014	Edessa Enerji	Enerji Kayıpları Saha Yöneticisi
2017	Reaktif Elektrik	Kırsal Arıza-Onarım-Bakım Şefi
2020	Aksa Doğalgaz	Tarife ve Raporlama Uzmanı

UZMANLIK ALANI

Kumanda Sistemleri, Otomasyon

YABANCI DİLLER

İngilizce

EKLER

EK-1 CMAX DEĞERLERİ

Tablo 1 Palmer sonuçları

Problem	Palmer Çalışma Zamanı	Palmer Cmax
tai001	0,001	1384
tai002	0,002	1439
tai003	0,002	1162
tai004	0,001	1490
tai005	0,001	1360
tai006	0,001	1344
tai007	0,001	1400
tai008	0,001	1313
tai009	0,001	1426
tai010	0,001	1229
tai011	0,002	1790
tai012	0,002	1948
tai013	0,002	1729
tai014	0,002	1585
tai015	0,002	1648
tai016	0,002	1527
tai017	0,002	1735
tai018	0,003	1763
tai019	0,003	1836
tai020	0,002	1898

Problem	Palmer Çalışma Zamanı	Palmer Cmax
tai021	0,004	2818
tai022	0,003	2331
tai023	0,003	2678
tai024	0,003	2629
tai025	0,004	2704
tai026	0,004	2572
tai027	0,003	2456
tai028	0,003	2435
tai029	0,004	2754
tai030	0,003	2633
tai031	0,002	2774
tai032	0,004	3041
tai033	0,004	2777
tai034	0,004	2860
tai035	0,003	2963
tai036	0,003	3090
tai037	0,002	2845
tai038	0,002	2826
tai039	0,004	2733
tai040	0,002	2915

Problem	Palmer Çalışma Zamanı	Palmer Cmax
tai041	0,004	3478
tai042	0,004	3313
tai043	0,006	3321
tai044	0,004	3511
tai045	0,005	3427
tai046	0,006	3323
tai047	0,005	3457
tai048	0,004	3356
tai049	0,004	3414
tai050	0,004	3404
tai051	0,008	4272
tai052	0,008	4303
tai053	0,007	4210
tai054	0,008	4233
tai055	0,007	4376
tai056	0,01	4312
tai057	0,007	4306
tai058	0,008	4310
tai059	0,006	4547
tai060	0,008	4197

Problem	Palmer Çalışma Zamanı	Palmer Cmax
tai061	2,148	5749
tai062	2,36	5316
tai063	2,522	5325
tai064	2,373	5049
tai065	3,344	5317
tai066	2,105	5274
tai067	2,248	5376
tai068	2,292	5263
tai069	6,272	5606
tai070	3,053	5427
tai071	5,543	6161
tai072	4,482	5889
tai073	5,755	6127
tai074	4,323	6313
tai075	4,264	6070
tai076	4,244	5870
tai077	4,25	6442
tai078	4,304	6168
tai079	4,271	6081
tai080	4,247	6259

Problem	Palmer Çalışma Zamanı	Palmer Cmax
tai081	0,014	7075
tai082	0,023	7058
tai083	0,019	7221
tai084	0,017	7039
tai085	0,019	7259
tai086	0,019	7109
tai087	0,016	7279
tai088	0,019	7567
tai089	0,013	7271
tai090	0,02	7305
tai091	0,013	11443
tai092	0,016	10986
tai093	0,022	11336
tai094	0,025	11221
tai095	0,017	11125
tai096	0,02	10865
tai097	0,014	11303
tai098	0,014	11275
tai099	0,018	11184
tai100	0,017	11333

TABLO 2 CDS Sonuçları

Problem	CDS Çalışma Zamanı	CDS Cmax
tai001	0,001	1390
tai002	0,002	1424
tai003	0,002	1255
tai004	0,001	1418
tai005	0,001	1323
tai006	0,001	1312
tai007	0,001	1393
tai008	0,001	1345
tai009	0,001	1360
tai010	0,001	1164
tai011	0,004	1757
tai012	0,004	1854
tai013	0,004	1651
tai014	0,004	1547
tai015	0,004	1558
tai016	0,004	1591
tai017	0,007	1630
tai018	0,008	1766
tai019	0,005	1720
tai020	0,005	1884

Problem	CDS Çalışma Zamanı	CDS Cmax
tai021	0,017	2559
tai022	0,018	2285
tai023	0,018	2565
tai024	0,018	2415
tai025	0,020	2506
tai026	0,020	2422
tai027	0,017	2489
tai028	0,017	2362
tai029	0,027	2409
tai030	0,022	2439
tai031	0,003	2883
tai032	0,005	3032
tai033	0,005	2703
tai034	0,003	2884
tai035	0,003	3038
tai036	0,005	3031
tai037	0,003	2944
tai038	0,003	2867
tai039	0,003	2784
tai040	0,003	2942

Problem	CDS Çalışma Zamanı	CDS Cmax
tai041	0,011	3382
tai042	0,014	3263
tai043	0,013	3287
tai044	0,013	3393
tai045	0,013	3375
tai046	0,011	3400
tai047	0,011	3530
tai048	0,011	3371
tai049	0,014	3265
tai050	0,010	3429
tai051	0,039	4324
tai052	0,046	4216
tai053	0,048	4203
tai054	0,044	4267
tai055	0,035	4122
tai056	0,056	4238
tai057	0,040	4134
tai058	0,050	4283
tai059	0,040	4219
tai060	0,054	4264

Problem	CDS Çalışma Zamanı	CDS Cmax
tai061	0,005	5592
tai062	0,005	5548
tai063	0,006	5493
tai064	0,005	5273
tai065	0,006	5484
tai066	0,006	5259
tai067	0,006	5561
tai068	0,007	5387
tai069	0,007	5758
tai070	0,008	5708
tai071	0,044	6239
tai072	0,019	5851
tai073	0,029	6023
tai074	0,027	6408
tai075	0,019	6018
tai076	0,021	5751
tai077	0,023	6202
tai078	0,020	6196
tai079	0,021	6349
tai080	0,018	6387

Problem	CDS Çalışma Zamanı	CDS Cmax
tai081	0,096	6962
tai082	0,098	6970
tai083	0,086	7233
tai084	0,098	7148
tai085	0,092	7118
tai086	0,106	7279
tai087	0,095	7124
tai088	0,089	7181
tai089	0,090	7181
tai090	0,079	7144
tai091	0,047	11610
tai092	0,044	11358
tai093	0,054	11732
tai094	0,051	11381
tai095	0,038	11324
tai096	0,063	11337
tai097	0,040	11649
tai098	0,036	11470
tai099	0,049	11259
tai100	0,039	11515

TABLO 3 NEH SONUÇLARI

Problem	NEH Çalışma Zamanı	NEH Cmax
tai001	0,021	1286
tai002	0,041	1365
tai003	0,027	1159
tai004	0,022	1325
tai005	0,022	1305
tai006	0,019	1228
tai007	0,019	1278
tai008	0,020	1223
tai009	0,030	1291
tai010	0,027	1151
tai011	0,050	1680
tai012	0,045	1729
tai013	0,052	1557
tai014	0,044	1439
tai015	0,045	1502
tai016	0,038	1453
tai017	0,056	1562
tai018	0,043	1609
tai019	0,051	1647
tai020	0,046	1653

Problem	NEH Çalışma Zamanı	NEH Cmax
tai021	0,101	2410
tai022	0,103	2150
tai023	0,102	2411
tai024	0,097	2262
tai025	0,100	2397
tai026	0,096	2349
tai027	0,102	2362
tai028	0,093	2249
tai029	0,109	2320
tai030	0,095	2277
tai031	0,340	2733
tai032	0,538	2843
tai033	0,479	2640
tai034	0,568	2782
tai035	0,372	2868
tai036	0,418	2850
tai037	0,340	2758
tai038	0,332	2721
tai039	0,334	2576
tai040	0,326	2790

Problem	NEH Çalışma Zamanı	NEH Cmax
tai041	0,650	3135
tai042	0,654	3032
tai043	0,644	2986
tai044	0,695	3198
tai045	0,693	3160
tai046	0,682	3178
tai047	0,651	3277
tai048	0,947	3123
tai049	1,079	3002
tai050	0,726	3257
tai051	1,307	4082
tai052	1,411	3921
tai053	1,266	3927
tai054	1,306	3969
tai055	1,219	3835
tai056	1,545	3914
tai057	1,199	3952
tai058	1,245	3938
tai059	1,189	3952
tai060	1,426	4079

Problem	NEH Çalışma Zamanı	NEH Cmax
tai061	2,148	5519
tai062	2,360	5348
tai063	2,522	5219
tai064	2,373	5023
tai065	3,344	5266
tai066	2,105	5139
tai067	2,248	5259
tai068	2,292	5120
tai069	6,272	5489
tai070	3,053	5341
tai071	5,543	5846
tai072	4,482	5453
tai073	5,755	5824
tai074	4,323	5929
tai075	4,264	5679
tai076	4,244	5375
tai077	4,250	5704
tai078	4,304	5760
tai079	4,271	6032
tai080	4,247	5918

Problem	NEH Çalışma Zamanı	NEH Cmax
tai081	8,605	6541
tai082	11,810	6523
tai083	9,434	6639
tai084	9,371	6557
tai085	9,978	6695
tai086	10,141	6664
tai087	9,297	6632
tai088	8,572	6739
tai089	8,513	6677
tai090	8,451	6677
tai091	32,371	10942
tai092	34,000	10716
tai093	44,227	11025
tai094	42,207	11057
tai095	32,649	10645
tai096	43,299	10458
tai097	32,752	10989
tai098	33,121	10829
tai099	41,835	10574
tai100	32,928	10807

TABLO 4. GA-PRO ALGORİTMA SONUÇLARI

Problem	GA-Pro Çalışma Zamanı	Genetik Cmax
tai001	33,017	1297
tai002	42,175	1383
tai003	41,066	1099
tai004	41,103	1330
tai005	34,200	1258
tai006	34,254	1214
tai007	34,280	1251
tai008	34,012	1223
tai009	34,011	1269
tai010	33,891	1132
tai011	62,738	1642
tai012	62,426	1730
tai013	62,717	1554
tai014	62,937	1443
tai015	62,685	1489
tai016	62,685	1477
tai017	62,501	1564
tai018	74,799	1614
tai019	67,346	1624
tai020	66,801	1652

Problem	GA-Pro Çalışma Zamanı	Genetik Cmax
tai021	127,610	2365
tai022	127,951	2156
tai023	127,538	2416
tai024	127,598	2271
tai025	126,832	2365
tai026	126,889	2307
tai027	127,310	2367
tai028	127,262	2293
tai029	127,839	2340
tai030	129,576	2275
tai031	226,977	2744
tai032	273,335	2882
tai033	290,233	2667
tai034	323,409	2774
tai035	243,647	2864
tai036	225,819	2846
tai037	232,751	2736
tai038	232,214	2706
tai039	225,055	2575
tai040	225,068	2790

Problem	GA-Pro Çalışma Zamanı	Genetik Cmax
tai041	429,666	3114
tai042	425,079	3082
tai043	419,602	2980
tai044	436,845	3156
tai045	427,526	3131
tai046	430,784	3130
tai047	427,194	3210
tai048	446,926	3117
tai049	503,809	3018
tai050	455,095	3211
tai051	927,841	4084
tai052	923,828	3871
tai053	845,802	3872
tai054	864,926	3984
tai055	854,084	3834
tai056	867,832	3933
tai057	854,995	3931
tai058	836,627	3887
tai059	843,313	3957
tai060	920,040	3976

Problem	GA-Pro Çalışma Zamanı	Genetik Cmax
tai061	1030,704	5527
tai062	1150,777	5290
tai063	1088,957	5221
tai064	938,211	5038
tai065	1184,850	5255
tai066	1084,175	5161
tai067	987,120	5285
tai068	1041,432	5137
tai069	1705,205	5504
tai070	1852,073	5350
tai071	2230,047	5871
tai072	2763,107	5458
tai073	2828,254	5782
tai074	1966,070	5979
tai075	1703,068	5610
tai076	1705,124	5453
tai077	1696,775	5692
tai078	1699,586	5744
tai079	1702,851	6024
tai080	1715,835	5940

Problem	GA-Pro Çalışma Zamanı	Genetik Cmax
tai081	3291,830	6557
tai082	5549,592	6513
tai083	3696,254	6569
tai084	3697,118	6586
tai085	3980,633	6682
tai086	3596,082	6697
tai087	3491,785	6542
tai088	3698,222	6780
tai089	3267,016	6643
tai090	3260,106	6719
tai091	7866,004	10994
tai092	8409,974	10701
tai093	8591,578	11079
tai094	8008,593	11057
tai095	8485,212	10690
tai096	8035,600	10467
tai097	7822,323	11006
tai098	7919,827	10869
tai099	9672,516	10519
tai100	8535,690	10826

TABLO 5. HİBRİT ALGORİTMA SONUÇLARI

Problem	Hibrit-GA Çalışma Zamanı	Hibrit-GA Cmax
tai001	33,639	1286
tai002	38,570	1365
tai003	39,929	1132
tai004	40,809	1304
tai005	34,540	1269
tai006	34,179	1224
tai007	34,140	1253
tai008	34,411	1222
tai009	34,056	1261
tai010	34,286	1130
tai011	62,709	1620
tai012	62,718	1729
tai013	62,648	1538
tai014	62,521	1433
tai015	62,559	1479
tai016	62,969	1453
tai017	63,075	1526
tai018	68,604	1605
tai019	66,695	1647
tai020	67,024	1645

Problem	Hibrit-GA Çalışma Zamanı	Hibrit-GA Cmax
tai021	127,394	2370
tai022	127,716	2148
tai023	126,974	2400
tai024	127,160	2250
tai025	126,781	2360
tai026	127,855	2290
tai027	126,690	2330
tai028	126,908	2239
tai029	129,273	2313
tai030	126,607	2258
tai031	226,269	2729
tai032	288,943	2843
tai033	289,606	2634
tai034	306,202	2782
tai035	225,776	2868
tai036	256,195	2841
tai037	250,608	2734
tai038	235,636	2703
tai039	225,294	2576
tai040	247,701	2785

Problem	Hibrit-GA Çalışma Zamanı	Hibrit-GA Cmax
tai041	424,666	3119
tai042	423,534	3025
tai043	426,297	2970
tai044	428,384	3165
tai045	430,486	3094
tai046	430,044	3114
tai047	422,835	3235
tai048	537,443	3123
tai049	483,972	2995
tai050	444,722	3219
tai051	913,300	4042
tai052	876,410	3919
tai053	875,287	3878
tai054	858,298	3954
tai055	845,123	3835
tai056	873,198	3889
tai057	840,118	3948
tai058	842,101	3902
tai059	915,277	3937
tai060	1013,135	3970

Problem	Hibrit-GA Çalışma Zamanı	Hibrit-GA Cmax
tai061	1152,316	5495
tai062	1110,119	5302
tai063	1035,829	5219
tai064	955,509	5023
tai065	1256,313	5261
tai066	910,344	5139
tai067	1046,150	5259
tai068	1022,070	5120
tai069	1769,107	5489
tai070	1211,125	5340
tai071	2125,647	5846
tai072	1763,537	5453
tai073	2295,537	5773
tai074	1696,567	5919
tai075	1696,267	5589
tai076	1696,827	5375
tai077	1696,691	5690
tai078	1703,846	5716
tai079	2046,523	6014
tai080	1695,697	5907

Problem	Hibrit-GA Çalışma Zamanı	Hibrit-GA Cmax
tai081	3746,312	6541
tai082	4523,523	6523
tai083	3535,715	6605
tai084	3674,757	6527
tai085	3873,789	6610
tai086	3675,131	6650
tai087	3760,353	6562
tai088	3276,637	6739
tai089	3268,020	6625
tai090	3267,915	6674
tai091	8114,030	10942
tai092	8087,418	10705
tai093	8866,820	11025
tai094	11375,445	11057
tai095	7846,364	10645
tai096	8211,425	10458
tai097	7943,484	10986
tai098	8285,375	10829
tai099	11803,076	10518
tai100	7867,770	10796

EK-2 GA-PRO VE HİBRİT-GA ÇALIŞTIRMA SONUÇLARI

TABLO 1 GA-PRO SONUÇLARI

	Çalıştırma No.									
Prob.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	1137	1137	1124	1127	1127	1127	1127	1151	1145	1127
20	1646	1664	1647	1653	1642	1659	1661	1653	1646	1655
30	2282	2270	2287	2294	2285	2282	2256	2284	2248	2268
40	2809	2783	2789	2783	2786	2786	2789	2794	2797	2786
50	3222	3196	3250	3206	3216	3204	3195	3195	3185	3242
60	4017	3952	3999	4019	3922	4005	3993	3968	3912	3978
70	5342	5344	5360	5357	5349	5352	5362	5342	5346	5347
80	5945	5918	5906	5903	6006	5903	5976	6041	5907	5903
90	6742	6711	6729	6733	6696	6700	6732	6747	6671	6730
100	10812	10827	10776	10819	10827	10805	10843	10819	10928	10811

TABLO 2 HİBRİT-GA SONUÇLARI

	Çalıştırma No.									
Prob.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	1127	1127	1109	1127	1128	1151	1127	1128	1130	1151
20	1643	1644	1646	1641	1653	1643	1653	1642	1639	1647
30	2262	2237	2270	2238	2239	2277	2277	2266	2246	2277
40	2789	2786	2786	2784	2783	2783	2783	2784	2783	2789
50	3217	3227	3211	3242	3207	3203	3238	3215	3244	3195
60	4014	3963	3982	3917	3974	3986	3972	3976	3945	3980
70	5341	5341	5341	5341	5341	5341	5331	5341	5341	5341
80	5903	5903	5918	5903	5914	5903	5903	5903	5918	5903
90	6676	6670	6674	6668	6674	6677	6676	6677	6676	6676
100	10807	10806	10807	10788	10807	10807	10777	10807	10789	10770

EK-3 TAILLARD PROBLEMLERİ ÜST DEĞERLERİ

Problem	Problem Sayısı	İş	Makine	Üst değer	Problem	Problem Sayısı	İş	Makine	Üst değer
tai 20X5	tai001	20	5	1278	**tai 50X5**	tai031	50	5	2724
	tai002	20	5	1359		tai032	50	5	2834
	tai003	20	5	1081		tai033	50	5	2621
	tai004	20	5	1293		tai034	50	5	2751
	tai005	20	5	1235		tai035	50	5	2863
	tai006	20	5	1195		tai036	50	5	2829
	tai007	20	5	1234		tai037	50	5	2725
	tai008	20	5	1206		tai038	50	5	2683
	tai009	20	5	1230		tai039	50	5	2552
	tai010	20	5	1108		tai040	50	5	2782
tai 20X10	tai011	20	10	1582	**tai 50X10**	tai041	50	10	2991
	tai012	20	10	1659		tai042	50	10	2867
	tai013	20	10	1496		tai043	50	10	2839
	tai014	20	10	1377		tai044	50	10	3063
	tai015	20	10	1419		tai045	50	10	2976
	tai016	20	10	1397		tai046	50	10	3006
	tai017	20	10	1484		tai047	50	10	3093
	tai018	20	10	1538		tai048	50	10	3037
	tai019	20	10	1593		tai049	50	10	2897
	tai020	20	10	1591		tai050	50	10	3065
tai 20X20	tai021	20	20	2297	**tai 50X20**	tai051	50	20	3850
	tai022	20	20	2099		tai052	50	20	3704
	tai023	20	20	2326		tai053	50	20	3640
	tai024	20	20	2223		tai054	50	20	3723
	tai025	20	20	2291		tai055	50	20	3611
	tai026	20	20	2226		tai056	50	20	3681
	tai027	20	20	2273		tai057	50	20	3704
	tai028	20	20	2200		tai058	50	20	3691
	tai029	20	20	2237		tai059	50	20	3743
	tai030	20	20	2178		tai060	50	20	3756

Problem	Problem Sayısı	İş	Makine	Üst değer	Problem	Problem Sayısı	İş	Makine	Üst değer
tai 100X5	tai061	100	5	5493	**tai 100X20**	tai081	100	20	6202
	tai062	100	5	5268		tai082	100	20	6183
	tai063	100	5	5175		tai083	100	20	6271
	tai064	100	5	5014		tai084	100	20	6269
	tai065	100	5	5250		tai085	100	20	6314
	tai066	100	5	5135		tai086	100	20	6364
	tai067	100	5	5246		tai087	100	20	6268
	tai068	100	5	5094		tai088	100	20	6401
	tai069	100	5	5448		tai089	100	20	6275
	tai070	100	5	5322		tai090	100	20	6434
tai 100X10	tai071	100	10	5770	**tai 200X10**	tai091	200	10	10862
	tai072	100	10	5349		tai092	200	10	10480
	tai073	100	10	5676		tai093	200	10	10922
	tai074	100	10	5781		tai094	200	10	10889
	tai075	100	10	5467		tai095	200	10	10524
	tai076	100	10	5303		tai096	200	10	10329
	tai077	100	10	5595		tai097	200	10	10854
	tai078	100	10	5617		tai098	200	10	10730
	tai079	100	10	5871		tai099	200	10	10438
	tai080	100	10	5845		tai100	200	10	10675

EK 4 HİBRİT-GA YÖNTEMİNE AİT EN KÜÇÜK TAMAMLANMA SÜRELERİ ve İŞ SIRALAMALARI

Tai010: 11,7,6,16,12,1,13,10,18,2,8,3,5,14,19,15,20,17,4,9
Makespan:1109

Tai020: 5,4,19,7,17,14,16,20,13,8,6,9,10,2,3,15,18,1,11,12
Makespan:1639

Tai030: 3,7,12,6,13,2,5,15,17,1,18,9,10,8,4,20,11,19,16,14
Makespan: 2237

Tai040:6,30,44,19,27,2,18,8,11,35,5,21,1,48,10,40,33,49,31,15,17,28,20,3,13,41,45,36,29,42,4,34,24,12,39,37,14,46,50,43,38,16,47,7,25,23,9,26,32,22
Makespan: 2783

Tai050:49,6,48,9,15,27,3,17,1,38,39,22,4,34,42,10,13,20,16,29,18,8,11,30,44,26,24,31,43,35,14,12,25,19,41,5,37,28,7,40,36,50,32,23,47,33,46,21,2,45
Makespan: 3203

Tai060:12,10,1,19,8,22,23,32,40,3,36,47,50,9,31,49,24,5,18,21,29,44,25,16,14,2,33,46,42,11,28,35,26,27,15,38,13,45,41,20,17,30,37,6,43,7,48,39,4,34
Makespan: 3917

Tai070:24,31,19,29,41,84,55,34,18,47,82,93,43,14,40,74,21,90,63,81,20,52,88,57,86,76,48,17,83,45,61,60,4,53,73,64,80,16,85,15,30,58,87,71,72,56,79,25,5,59,67,94,37,44,8,9,77,68,97,28,6,22,69,3,32,99,78,13,11,42,50,51,89,38,95,49,66,96,70,33,23,36,46,39,65,54,27,12,26,1,10,91,100,92,98,75,7,35,62,2
Makespan: 5331

Tai080:63,71,66,89,88,94,75,92,87,22,80,81,58,78,54,44,14,21,40,30,1,65,95,97,83,11,4,8,42,7,13,79,52,2,17,82,9,20,28,6,35,5,18,15,23,43,12,59,33,76,37,51,45,25,10,47,19,24,46,49,50,29,64,68,38,26,99,60,48,41,56,57,61,70,72,39,36,85,96,53,31,55,69,67,74,3,16,98,32,77,91,86,84,27,93,90,73,100,62,34
Makespan: 5903

Tai090:77,11,34,15,27,20,56,28,81,100,38,35,21,88,67,24,7,1,54,42,51,62,89,6,60,87,79,71,74,47,12,33,23,41,13,25,83,92,98,37,86,40,52,78,48,2,46,72,5,32,65,84,10,97,9,91,95,68,26,18,22,43,8,30,31,4,50,69,96,61,16,66,90,82,14,99,3,76,63,36,94,29,17,70,49,80,45,64,57,44,75,73,39,58,19,55,59,85,53,93
Makespan: 6668

Tai100:149,74,184,148,161,43,127,138,24,143,158,185,130,69,109,85,46,29,102,122,190,107,68,117,118,78,81,11,19,47,38,86,14,28,15,57,82,33,181,61,3,58,9,71,20,53,147,121,123,160,116,96,32,141,8,159,101,103,151,173,13,18,152,84,113,36,42,21,135,64,153,100,16,92,80,34,144,200,44,26,98,174,188,77,133,59,50,4,49,128,17,93,179,35,23,6,105,39,182,88,22,177,10,165,56,194,124,1,91,94,129,193,75,191,40,106,55,156,110,131,95,163,51,139,157,45,37,111,70,90,167,178,83,99,195,60,120,73,12,171,132,31,87,114,112,199,146,154,137,180,168,27,7,5,25,108,183,66,72,162,134,2,115,175,62,65,172,54,125,142,176,89,166,186,192,63,136,67,48,140,145,170,41,196,197,164,189,119,150,187,104,169,30,79,97,198,155,52,126,76
Makespan: 10777