

Özdeş Paralel Makinelerde  
Toplam Gecikmeyi Enküçükleme Amaçlı  
Çizelgeleme Problemi için Sezgisel Bir Çözüm Yaklaşımı

Gülcan Gocuklu

**YÜKSEK LİSANS TEZİ**

Endüstri Mühendisliği Anabilim Dalı

Haziran 2011

A Heuristic Solution Approach for  
Identical Parallel Machine Scheduling Problem  
with Minimum Total Tardiness

Gülcan Gocuklu

**MASTER OF SCIENCE THESIS**

Department of Industrial Engineering

June 2011

Özdeş Paralel Makinelerde  
Toplam Gecikmeyi Enküçükleme Amaçlı  
Çizelgeleme Problemi için Sezgisel Bir Çözüm Yaklaşımı

Gülcan Gocuklu

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği uyarınca  
Endüstri Mühendisliği Anabilim Dalı  
Yöneylem Araştırması Bilim Dalı' nda  
YÜKSEK LİSANS TEZİ  
olarak hazırlanmıştır.

Danışman: Prof. Dr. Müjgan Sağır

Haziran 2011

## ONAY

Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Gülcan Gocuklu' nun YÜKSEK LİSANS tezi olarak hazırladığı “Özdeş Paralel Makinelerde Toplam Gecikmeyi Enküçükleme Amaçlı Çizelgeleme Problemi için Sezgisel Bir Çözüm Yaklaşımı” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Prof. Dr. Müjgan SAĞIR

### **Yüksek Lisans Tez Savunma Jürisi:**

**Üye** : Prof. Dr. Müjgan SAĞIR (Danışman)

**Üye** : Prof. Dr. Nihat YÜZÜGÜLLÜ

**Üye** : Doç. Dr. Şerafettin ALPAY

**Üye** : Yrd. Doç. Dr. Ediz ATMACA

**Üye** : Yrd. Doç. Dr. Servet HASGÜL

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

## ÖZET

Bu çalışmada, özdeş paralel makine çizelgeleme probleminin, işlerin toplam gecikme süresini enküçükleme amacına yönelik olarak çözümü ele alınmıştır. Problem için, literatürde yer alan modellerden yararlanılarak 0-1 karma tamsayılı matematiksel model geliştirilmiştir. Problemin NP-zor problem sınıfında olması sebebi ile özellikle büyük boyutlu veri setlerinde eniyi çözüme matematiksel model ile ulaşmanın nereden ise imkansız olması, sezgisel bir yaklaşım ihtiyacını kaçınılmaz kılmıştır. Bu amaçla geliştirilen sezgisel algoritma, C# programlama dili kullanılarak yazılmıştır. Kullanıcı etkileşimli arayüzler, sistemin kullanımını kolaylaştırmakta ve dinamik ortamlarda planlama esnekliği sağlamaktadır. Gerçek veri setleri kullanılarak matematiksel model ve sezgisel algoritmanın çözüm sonuçları karşılaştırılmıştır.

Anahtar Kelimeler: Özdeş paralel makineler, çizelgeleme, sezgisel algoritma.

## SUMMARY

A heuristic solution approach is developed to address the identical parallel machine scheduling problem with minimal total tardiness. Motivated by the current mathematical models in the literature, a revised and combined new version (a new mathematical model) is developed. On the other hand due to the NP-hard characteristic of the defined problem, the need for a heuristic solution approach is inevitable. The algorithm is coded by C# programming language. User interfaces provide a flexible environment in dynamic production system under consideration.

In order to evaluate the performance of the proposed heuristic, the current scheduling approach is also coded as an algorithm. By generating the previous schedules as many as this way, they are compared with the proposed ones. The total tardiness value is obtained as smaller in all the cases considered.

Keywords: Identical parallel machines, scheduling, heuristic algorithm.

## TEŐEKKÖR

Çalıőmalarım süresince yardımlarını esirgemeyen, bilgi ve deneyimleri ile çalıőmalarımı yönlendirmemde deęerli katkıları bulunan, sonsuz anlayıő ve sabır gösteren danıőman hocam Sayın Prof. Dr. Müjgan Saęır baőta olmak üzere; lisans ve yüksek lisans öęrenimim süresince üzerimde emeęi olan, yardım ve deęerli bilgilerini esirgemeyen tüm hocalarıma; her zaman bana güvenerek hep yanımda olan canım aileme ve sevgili arkadaşlarıma teőekkürlerimi sunarım.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET.....</b>	<b>v</b>
<b>SUMMARY.....</b>	<b>vi</b>
<b>TEŞEKKÜR.....</b>	<b>vii</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>x</b>
<b>ÇİZELGELER DİZİNİ.....</b>	<b>xi</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMLERİ VE LİTERATÜR ARAŞTIRMASI.....</b>	<b>4</b>
2.1. Özdeş Paralel Makine Çizelgeleme Problemlerinin Yapısı.....	8
2.2. Çalışmalar ve Sınıflandırmalar.....	9
2.2.1. Yıllara göre çalışmalar.....	13
2.2.2. Çözüm yöntemlerine göre çalışmalar.....	13
2.2.3. Performans ölçütlerine göre çalışmalar.....	15
<b>3. ÖZDEŞ PARALEL MAKİNELERDE ÇİZELGELEME PROBLEMİ İÇİN SEZGİSEL BİR YAKLAŞIM.....</b>	<b>17</b>
3.1. Matematiksel Model, Boyut Analizi ve Sezgisel Algoritma Geliştirme Gerekçeleri.....	17
3.2. Geliştirilen Sezgisel Algoritma Temelli Çözüm Yaklaşımı.....	24
3.2.1. Mevcut ve önerilen algoritmalar.....	24
3.2.2. Geliştirilen algoritmaya dayalı çizelgeleme sistemi –görsel.....	42



**İÇİNDEKİLER (devam)**

	<b><u>Sayfa</u></b>
<b>4. UYGULAMA.....</b>	<b>48</b>
4.1. İşletmenin Tanıtımı.....	48
4.2. İşletmedeki Üretim Süreci ve Ele Alınan Problem.....	50
4.3. Problemin Geliştirilen Yaklaşım Kullanılarak Çözümü.....	56
4.3.1. Mevcut ve önerilen yaklaşımların çözüm sonuçlarını karşılaştırma...	56
4.3.2. Karşılaştırma grafikleri.....	59
<b>5. SONUÇ VE ÖNERİLER.....</b>	<b>61</b>
<b>KAYNAKLAR DİZİNİ.....</b>	<b>63</b>
<b>EKLER</b>	

## ŞEKİLLER DİZİNİ

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
2.1. Yıllara göre Ö.P.M.Ç.P. yayın sayıları.....	13
2.2. Ö.P.M.Ç.P.' nin çözümü için kullanılan yaklaşımların dağılımı.....	14
2.3. Ö.P.M.Ç.P.' nin çözümünde kullanılan yaklaşımların yıllara göre dağılımı.....	15
2.4. Ö.P.M.Ç.P.' de kullanılan performans ölçütleri dağılımı.....	16
3.1. Geliştirilen sezgisel yaklaşıma ait akış diyagramı.....	27
3.2. Program arayüzü.....	42
3.3. Veri tablosu ekran görüntüsü.....	43
3.4. Mevcut yaklaşım ile çizelgeleme sonuçlarının ekran görüntüsü.....	44
3.5. Önerilen yaklaşım ile çizelgeleme sonuçlarının ekran görüntüsü.....	44
3.6. Şekil 3.4.' ün bir kısmı (24.01.2008 tarihi 9. saate kadar yapılan iş bilgileri).....	45
3.7. Şekil 3.5.' in bir kısmı (24.01.2008 tarihi 9. saate kadar yapılan iş bilgileri).....	45
3.8. Test problemlerinin matematiksel model ve sezgisel algoritma kullanılarak elde edilen gecikme süreleri (sa).....	47
4.1. Fabrika içi süreçler arası ilişkinin gösterimi.....	51
4.2. Fabrika içi süreçlerin akış şeması.....	53
4.3. Mevcut ve önerilen yaklaşıma ait gecikme süreleri (sa).....	59
4.4. Kullanılan çözüm yaklaşımı ve makine sayısına göre gecikme süreleri dağılımı.....	60

**ÇİZELGELER DİZİNİ**

<b><u>Cizelge</u></b>	<b><u>Sayfa</u></b>
3.1.(a) Matematiksel modele ilişkin boyut analizi (10 iş ve 2 makine için örnek).....	21
3.1.(b) Matematiksel modele ilişkin boyut analizi (1000 iş ve 10 makine için örnek).....	22
3.2. Örnek veri seti (10 iş ve 2 makine).....	33
3.3. Örnek problem verilerinin termin bilgisine göre sıralanmış şekli.....	33
3.4. Örnek çizelgeleme probleminin çözümü (10 iş ve 2 makine).....	41
3.5. Test problemleri, parametreleri, sezgisel algoritma ve matematiksel model sonuç karşılaştıma .....	46
4.1. Üretim hatları ve üretilen ürün özellikleri .....	49
4.2. Mevcut ve önerilen çözüm yaklaşımlarının karşılaştırılması.....	57

## 1. GİRİŞ

Üretim yapan işletmelerde, mevcut kaynakların verimli bir şekilde kullanılarak talebin karşılanabilmesi, doğru bir planlama ile gerçekleşebilir. Planlama sürecindeki her bir faaliyetin, süreçte yer alan diğer faaliyetlerle eşgüdüm içerisinde olması gerekmektedir.

Planlama süreci, işletmenin uzun vadedeki hedef ve politikalarının belirlendiği stratejik planlama ile başlar. Kaynak planlaması ve talep bilgileri dikkate alınarak üretim planlarının hazırlanmasıyla, altı ay veya daha uzun sürede kullanılacak işgücü veya kaynak miktarı belirlenmektedir. Sonrasında ise; üretim planlarından sağlanan bilgilerden, müşteri siparişlerinden ve talep tahminlerinden hareketle, her bir devrede ne üretileceği ve ne zaman üretileceği gibi sorulara cevap veren üretim programlarıyla, kullanılacak işgücü ve kaynak miktarları aylık veya üç aylık devrelere dönüştürülmektedir. Ardından hazırlanan malzeme gereksinim planlaması, üretimin gerçekleşmesi için gerekli malzemenin istenen zaman ve miktarda bulunmasını sağlamaktadır. Son olarak sipariş ve iş emirleri yaratılarak, atölyelere işler yollanıp, yükleme yapıldıktan sonra, çizelgeleme problemi ile karşılaşılır.

Üretim çizelgeleme, üretim planlama ve kontrol sürecinin bir parçası olup, içerisinde iki alt problem olarak sıralama ve zamanlama ele alınmaktadır. Öncelikle atölye içindeki işlerin sıraları, daha sonra, belirlenen sıraya bağlı olarak işlerin başlama ve bitiş zamanları belirlenmektedir. Sıralama ile birtakım ölçütler dikkate alınarak işlerin makine(ler)deki öncelikleri belirlenirken; zamanlama (timetabling), iş sıralarından ve rotalardan faydalanılarak işlerin makinelerde başlama ve bitiş zamanlarının belirlenmesidir.

Bir üretim sisteminde süreç içinde sürekli olarak stok bulunması, bazı makinelerin boş kalması, geciken işlerin çokluğu gibi sorunlar bir çizelgeleme probleminin varlığının göstergeleri olabilir. Bu sorunların genellikle kapasite yetersizliğinden kaynaklandığı düşünülmektedir ve kapasite artırımı gibi çarelere başvurulmaktadır. Oysa iyi bir çizelgeleme ile bu sorunların önemli bir kısmı ortadan kaldırılabilir veya azaltılabilir.

Gerçek hayatta karşılaşılan çizelgeleme problemleri oldukça karmaşık ve çözümleri zordur.  $n$ , iş;  $m$ , makine sayısı iken  $(n!)^m$  adet farklı sıralama yapmak söz konusudur. İş ve makine sayısındaki artış, alternatif sıralama sayısını üstel olarak arttırmaktadır.

İki özdeş paralel makinenin olduğu ve iş gecikmelerinin en küçüklendiği bir çizelgeleme problemi NP-zordur. Makine sayısı ikiden fazla olduğunda ise bu problemin zorluğu daha da artmaktadır (Chang vd., 2005).

Aynı işi yapabilen birden fazla sayıda makineler, paralel makineler olarak adlandırılmaktadır. Bu tip makinelerin çizelgenmesi, tek makine çizelgelemesine göre daha karmaşıktır. Paralel makine çizelgeleme problemi, gerçek hayatta çok sık var olması ve çok aşamalı daha karmaşık problemlerin de alt problemi olması sebebiyle oldukça önemlidir. Bu problemleri, parçaların makinelerde işlenme süreleri açısından aşağıdaki gibi üç gruba ayırmak mümkündür (Pinedo, 2002):

- Bir parça tüm makinelerde aynı sürede üretilebiliyorsa, bu makinelere özdeş (*identical*) makine,
- Bir parça tüm makinelerde aynı sürede üretilmiyor ancak süre farklılıkları parametrik bir ilişki ile açıklanabiliyorsa, bu tür makinelere düzgün (*uniform*) makine,
- Bir parçanın makinelerdeki üretim süreleri düzensiz bir şekilde farklılık gösteriyorsa, bu tür makinelere ilişkisiz (*unrelated*) makine denir.

Üretim çizelgeleme problemleri, üretim tipine göre farklı biçimlerde olabilir. Çizelgeleme problemleri işlem karmaşıklığı açısından ele alınırsa, göz önünde bulundurulması gereken kademe sayısına göre aşağıda belirtilen dört farklı başlıkta incelenebilirler (French, 1982):

- Tek kademe-tek makine problemi, en basit problem biçimidir. Burada bütün işler, tek makinede işlenmek üzere tek bir işlem kademesini gerektirmektedir.
- Tek kademe-paralel makine problemi, tek makine problemine benzemektedir. Her iş paralel makinelerin birisinde işlenmek üzere yine tek bir işlem kademesini gerektirmektedir. Ancak bu problemde, aynı işi yapan birden fazla makine mevcuttur.
- Çok kademe problemleri, her bir işin işlem sırasında çok kesin bir öncelik ilişkisinin bulunduğu durumdur. Her iş, makineler grubunda öncelik ilişkisine göre işlenmeyi gerektirir. Çok kademeli problemler, akış tipi ve atölye tipi olmak üzere iki şekilde

incelenebilir. Akış tipi problemde, bütün işler aynı işlem sırasıyla aynı makine grubunda işlenir. Diğer bir deyişle, işlerin makinelerdeki işlem sırası (teknolojik kısıt) ve öncelik ilişkisi aynıdır.

- Atölye tipi problem ise, sınıflandırmadaki en genel ve en karmaşık olanıdır. Belirli bir işe ait işlem kademeleri sayısı üzerinde hiçbir kısıt yoktur. Başka bir deyişle, atölye tipi problemde her bir iş, farklı makinelerde işlenmek üzere kendine özgü bir işlem sırasına sahiptir.

Çalışma kapsamında, bir işletmedeki bir işlem sürecinde kullanılan özdeş paralel makinelerin çizelgeleme problemi ele alınmıştır. Bu problem, French (1982)' in kademe sayısına göre gruplandığı üretim çizelgeleme problemlerinden, tek kademe-paralel makine problemleri grubuna girmektedir. Makine özellikleri bakımından ise Pinedo (2002)' nun sınıflandırmasına göre özdeş paralel makine tanımına uymaktadır. Çizelgenecek işlerin üretim süreleri, tüm makinelerde aynı olduğundan, makineler özdeş olup, iş gecikmelerinin en küçüklenmek istendiği bu problem NP-zor sınıfında yer almaktadır. Çalışmada; makine sayısının ikiden fazla olduğu durum için işletmeye ait süreçteki kısıtlar dikkate alınarak bir çözüm yaklaşımı önerilmiştir.

İkinci bölümde, paralel makine çizelgeleme problemlerinin yapısından bahsedilmiş, aynı zamanda yayın araştırması sonucunda erişilebilen ilgili çalışmaların; yıllara göre seyri, çözüm yöntemleri ve performans ölçütleri değerlendirilmiştir. Üçüncü bölümde, ele alınan probleme ilişkin önerilen çözüm yaklaşımı tanıtılmıştır. Çalışma kapsamında problemin en iyi çözümünün bulunabilirliğini sınamak amacıyla önce matematiksel model geliştirilmiş ve boyut analizi yapılmıştır. Problem boyutu büyüdükçe çözüm süresinin üstel olarak artması; hızlı çözüme ulaşan, değişikliklere duyarlı, kullanıcı etkileşimli bir başka yaklaşım ihtiyacı doğurmuştur. Tasarlanan algoritma C# programlama dili kullanılarak kodlanmış sezgisel bir algoritmadır. Dördüncü bölümde, çalışılan problemin bir işletmedeki uygulamasına yer verilmiştir. İşletmedeki üretim süreci ve mevcut sistemde yaşanan sıkıntılardan bahsedilmiş, problemin geliştirilen yaklaşım kullanılarak çözümü üzerinde durulmuştur. Geliştirilen sezgisel yaklaşım ve kullanılan matematiksel model temelinde veri setleri ile karşılaştırmalar yapılmıştır. Sonuç ve öneriler bölümünde ise; önerilen yaklaşımın genel bir değerlendirilmesi yapılarak, ileride ele alınabilir çalışmalarla ilgili öneriler belirtilmektedir.

## 2. ÖZDEŞ PARALEL MAKİNE ÇİZELGELEME PROBLEMLERİ VE LİTERATÜR ARAŞTIRMASI

İşletmelerin günümüz ekonomisinde rekabet edebilmeleri için müşteri taleplerine hızla cevap verebilmeleri bir zorunluluktur. Sipariş gecikmelerinden kaynaklanan müşteri kayıpları işletmeler açısından istenmez. Çizelgeleme yapılır iken işletmeler tarafından belirlenen amaçlar da bu doğrultuda pek çok farklı şekillerde olabilir. Bir amaç; son işin tamamlanma süresinin en küçüklenmesi olabilirken, bir başkası ise teslim zamanından sonra tamamlanan yani geciken işlerin sayısının en küçüklenmesi olabilir. Aynı şekilde iş yükü dengesinin sağlanması da bir başka amaçtır. İşlerin makinelere mümkün olduğunca dengeli bir şekilde dağıtılması ile darboğazlar elimine edilebilir, çıktı miktarı en büyüklenebilir ve bitmiş ürün stoklarının ve üretim maliyetlerinin azaltılması sağlanabilir (Rajakumar, et al., 2004).

Üretim çizelgeleme, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıda makinelerde hangi sırada ve ne zaman işleneceğinin saptanmasıdır. Üretim çizelgeleme problemleri, üretim tipine göre çok farklı biçimlerde olabilir. Literatürde, üretim çizelgeleme problemleri için pek çok sınıflandırmalar yapıldığı görülür. Herhangi bir sınıflandırmanın amacı, problem sınıflarının anlaşılmasını sağlamak ve her bir sınıflın farklı özelliklerini saptamaktır (Graves, 1981).

Üretim çizelgeleme, üretim planlamaya göre daha ayrıntılı ve kısa dönemlidir. Üretim çizelgeleri ürünlerin üretilmesi veya gerekli işlemlerin yerine getirilmesi için zaman ve sıralama açısından yapılan planlardır. Üretim kontrolü ise planlama yapılan zaman dilimi içinde çok önemlidir ve bu kontrol ile planların geçerliliği araştırılır. Kontrol gerçekleşmediği zaman çizelgelemenin performansını belirlemek güçtür. Etkin bir çizelgeleme ve kontrolün üstünlükleri çoktur. Kaynakların etkin kullanımı ile verimlilik yükselir. Müşteriler açısından etkin bir çizelgeleme ve kontrol, müşteri siparişlerinin zamanında karşılanması demektir.

Çizelgeleme problemlerinde, çizelgelenecek işlerin ve makinelerin önceden bilindiği varsayımı unutulmamalıdır. Çizelgelenecek işlerin sayısı  $n$ , makinelerin sayısı ise  $m$  ile ifade edildiğinde Pinedo (2002), temel kavramları aşağıdaki gibi tanımlamıştır:

İşlem zamanı (processing time –  $p_{ij}$ ):  $j$  işinin  $i$  makinesindeki işlem zamanını göstermektedir. Eğer  $j$  işinin işlem zamanı makineden bağımsız veya sadece tek makinede işlem söz konusu ise  $i$  indisi kullanılmayabilir.

Serbest bırakma zamanı (release date –  $r_j$ ): Serbest bırakma zamanı veya hazır olma zamanı,  $j$  işinin sistemde işlem görmeye başlayabileceği zamanı ifade eder.

Teslim tarihi (due date –  $d_j$ ):  $j$  işinin tamamlanmış olarak müşteriye teslim edileceği tarihi gösterir.

Ağırlık (weight –  $w_j$ ): Basit olarak bir öncelik faktörünü ifade eder.  $j$  işinin sistemdeki diğer işlere göre önemini gösterir. Örneğin, bu ağırlık faktörü işlerin sistemde kalmasının bir maliyeti olabilir. Stoklama veya elde tutma maliyeti olarak düşünülebileceği gibi verilen bir katma değer olarak da kabul edilebilir.

Çizelgeleme problemlerinde; makine ortamı, kısıt(lar) ve amaç bilgilerine ihtiyaç vardır. Bir çizelgeleme problemi  $\alpha / \beta / \gamma$  üçlüsü ile karakterize edilir.  $\alpha$ , makine ortam bilgisini tanımlar.  $\beta$  alanı işlem özellikleri ve kısıtları hakkında bilgi verir. Bu alana çoklu giriş olabildiği gibi bilgi girişi yapılmayabilir.  $\gamma$  alanı ise eniyilemek istenilen amacı belirtmek üzere genellikle bir giriş gerektirir (Pinedo, 2002).

$\alpha$  alanında belirtilen olası makine ortam bilgileri aşağıdaki gibidir:

- **$I$** : Olası makine çevrelerinin en basiti olan tek makineyi ifade eder. Tek makine diğer tüm karmaşık sistemlerin özel bir durumudur.
- **$P_m$** : Birbiriyle aynı özelliklere sahip  $m$  adet paralel makineyi ifade eder. İşler paralel bağlı makinelerin her hangi birinde işlem görebilir.
- **$F_m$** : Akış tipi yerleşim düzeninde (flowshop layout) birbirine seri olarak bağlı  $m$  adet makineyi ifade eder. Tüm işler aynı rotayı izleyerek işlem görürler. Bir makinede işlemi tamamlanan iş, sonraki makinede işlem görmek üzere kuyruğa girer ve işler genellikle ilk giren ilk çıkar kuralına göre işlem görürler. Bu gibi akış tipi yerleşimler permutasyon akışı olarak nitelendirilir ve  $\beta$  alanı  $prmu^1$  girişini içerir.
- **$FFC$** : Esnek akış tipi yerleşimi (flexible flowshop layout) ifade eder. Akış tipi yerleşim ve paralel makine çevrelerinin genelleştirilmiş halidir. Birbirine seri bağlı  $c$  adet aşamanın yanında her bir aşamada paralel makine durumunu işaret eder.

<sup>1</sup> İleride açıklanacağı üzere; akış tipi üretim ortamlarında görülen bu kısıtlama, makinelerin önünde işlem görmek için kuyrukta bekleyen işlerin ilk giren ilk çıkar kuralına göre işleme alınmasını, bir sıralama ve permutasyonu ifade eder.



- **$J_m$** : Atölye tipi yerleşim düzeninde (job shop layout), her bir işin kendisine ait, önceden belirlenmiş farklı bir rotası vardır. Bu makine çevresinde işler makineleri bir veya daha fazla sayıda ziyaret ederek rotalarını tamamlayabilirler.
- **$FJ_c$** : Esnek atölye ortamı (flexible job shop layout), atölye tipi yerleşimin ve paralel makine çevresinin genelleştirilmiş halidir. Her bir iş merkezinde belirli sayıda paralel makinenin bulunduğu  $c$  adet iş merkezini ifade eder.
- **$O_m$** : Açık atölye yerleşiminde her bir iş  $m$  adet makinenin her birinde yeniden işlem görebilir. Buna karşın, bu işlem sürelerinden bazıları sıfır olabilir. İşlerin makinelerdeki işlem rotalarına ait bir kısıtlama yoktur. Her bir iş için bir rota hazırlanır ve farklı işler farklı rotalara sahip olabilir.

$\beta$  alanında tanımlanan kısıtlara ait olası girişler ise aşağıdaki gibi olabilir:

- **$r_j$** : Serbest bırakma zamanı (release date),  $j$  işinin serbest bırakma zamanından önce işlem görememesi durumunu belirtmek için kullanılır. Eğer  $\beta$  alanında  $r_j$  değeri bulunmuyorsa, bu durumda  $j$  işinin işlemi herhangi bir zamanda başlayabilir anlamına gelir.
- **$s_{ij}$** : Hazırlık süresi (setup time),  $i$  işi ile  $j$  işi arasındaki hazırlık süresini gösterir.
- **$prmp$** : İşlerin bölünebilirliği (preemption), bir makinede işlem görmeye başlayan bir işin tamamlanmadan önce durdurulup yeni bir işe başlanabilmesi olarak tanımlanır. Eğer çizelgeleme problemlerinde işlerin bölünebilirliğine izin veriliyorsa  $\beta$  alanında  $prmp$  ifadesi bulunur. Aksi takdirde işlerin bir makinedeki işlemleri boyunca bölünmesine izin verilmez.
- **$prec$** : Öncelik kısıtları (precedence constraints) tek makine veya paralel makine ortamlarında, bir veya daha fazla işin bir makinede diğer işler işlem görmeden tamamlanması gerektiğini ifade eder.
- **$prmu$** : Akış tipi üretim ortamlarında görülen bu kısıtlama, makinelerin önünde işlem görmek için kuyrukta bekleyen işlerin, ilk giren ilk çıkar kuralına göre işleme alınmasını, bir sıralama ve permutasyonu ifade eder.
- **$nwt$** : Akış tipi üretim ortamında görülen özelliklerden birisi de beklemelere izin verilmemesi durumudur. Birbirini takip eden iki makine arasında işlerin beklemesine izin verilmemesini ifade eder. Böyle bir durumda gerekirse, bir işin sonraki makinelerde beklememesini sağlamak için ilk makinede bekletilmesi söz konusu olabilir.

• **recrc:** Yeniden dolaşım, bir işin bir makineyi veya iş merkezini birden fazla sayıda ziyaret etmesini gerektirecek atölye tipi ve esnek atölye tipi üretim ortamlarında görülür.

$\gamma$  alanlarında problemin amacına ait bilgiler yer alır. Eniyilenmeye çalışılan amaç, tamamlanma zamanının bir fonksiyonudur ve bu sebeple çizelgeye bağlıdır.

$j$  işinin  $i$  makinesindeki operasyonunun tamamlanma zamanı  $C_{ij}$  ile gösterilir.  $j$  işinin sistemden çıkış zamanı ise  $C_j$  ile ifade edilir. Ayrıca amaç, teslim tarihinin bir fonksiyonu da olabilir.  $j$  işinin tamamlanma zamanı ve termini arasındaki sapma (gecikme-lateness) - varsa-  $L_j=C_j-d_j$  ile gösterilir.

$L_j$ ;  $j$  işinin tamamlanma zamanı terminini aştığında pozitif,  $i$  işi erken tamamlandığında ise negatif değer alır.

Tamamlanma zamanının bir başka fonksiyonu olan,  $j$  işinin gecikme süresi (geçlik-tardiness) ise  $T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$  ile ifade edilir.

Sapma ( $L_j$ ) ve gecikme süresi ( $T_j$ ) arasındaki temel fark, gecikme süresi parametresinin alacağı değerin asla negatif olmamasıdır.

$\gamma$  alanlarında gösterilebilecek diğer amaçlar ise şunlar olabilir:

- *Akış süresini (flow time,  $F_j$ ) enküçüklemek:*  $j$  işinin sistemde geçirdiği zamanı gösterir ve  $F_j=C_j-r_j$  ifadesiyle hesaplanabilir.
- *En büyük gecikmeyi (lateness,  $L_{max}$ ) enküçüklemek:* Hedeflenen teslim tarihinden en büyük sapmanın enküçüklenmesini sağlar.  $Enb(L_1, \dots, L_n)$  olarak tanımlanır.
- *Ağırlıklandırılmış toplam tamamlanma zamanını  $\Sigma(w_j C_j)$  enküçüklemek:*  $n$  adet işin ağırlıklandırılmış toplam tamamlanma zamanı, çizelgeye bağlı olarak, toplam stok maliyeti veya elde tutma maliyetinin bir ölçüsünü verir.
- *Ağırlıklandırılmış toplam geçliği  $\Sigma(w_j T_j)$  enküçüklemek:* Toplam ağırlıklandırılmış tamamlanma zamanından daha genel bir maliyet fonksiyonudur.
- *Ağırlıklandırılmış geciken iş sayısını  $\Sigma(w_j U_j)$  enküçüklemek:* Toplam ağırlıklı geciken iş sayısının küçüklenmesini sağlar.
- *Erkenliği (earliness,  $E_j$ ) enküçüklemek:* Bir işin teslim tarihinden önce tamamlanmasının ölçüsüdür.  $E_j=\max(d_j-C_j, 0)$  ifadesi ile hesaplanmaktadır.

Çizelgeleme problemlerinin tanımlanmasında kullanılan gösterime ait örnekler vermek gerekirse;  $P_m \mid prmp \mid L_{max}$  gösterimi paralel makinelerin olduğu bir sistemde,

işlerin bölünebilirliği kısıtı altında en büyük gecikmenin enküçüklenmesinin amaçlandığı bir problemi tariflemektedir.

Çalışma kapsamında ele alınan problem  $P_m / s_{ij} / \sum T_j$  şeklinde tanımlanmaktadır. Bu gösterim, paralel makinelerin olduğu bir sistemde, işler arası geçişte hazırlık süresi bulunan, toplam gecikme süresini enküçükmek amaçlı çizelgeleme problemine karşılık gelmektedir.

## 2.1. Özdeş Paralel Makine Çizelgeleme Problemlerinin Yapısı

Paralel makine problemleri, aynı işi yapabilen  $m$  tane makinenin paralel olarak yerleştiği sistemlerdir. Her iş yalnız bir operasyona sahiptir ve bu  $m$  makinenin herhangi birinde işlenebilir. Paralel makineler; aynı tipte, hızları ise aynı veya farklı ya da farklı tipte, hızları ise aynı veya farklı olabilirler.

Paralel makineli bir üretim sisteminde özdeş, benzer ve birbirinden bağımsız makinelerden söz edilebilir. Makineler aynı işi, aynı sürede yapıyorsa önceki bölümlerde bahsedildiği gibi, özdeştirler. İşlemlerden her biri, sistemde yer alan herhangi bir makinede işlem görebilir ve işlem süresi değişmez. Birbirinden farklı hızlara sahip makineler ise benzer makineler olarak adlandırılırken, makine hızlarının işlere bağlı olarak değiştiği makineler birbirinden bağımsız paralel makineler adını almaktadır.

İşletmelerde aynı işi yapan birden fazla makinenin bulunduğu durumlarda çizelgeleme problemine makine seçimi boyutu eklenmektedir. Aynı işi yapan birden fazla makinenin bulunması işletmelere kapasite artışı ve esneklik sağlarken, eldeki kapasiteyi verimli yönetmek ciddi bir problem haline gelmektedir. Özellikle hazırlık sürelerinin yüksek olduğu üretim sistemlerinde, her bir paralel makine fazladan hazırlık süresi anlamına geldiği için makinelerin doğru çizelgelenmesi daha da önem kazanmaktadır.

Tek makineli çizelgeleme probleminde çizelgelenecek işler belirli iken, paralel makinelerin olduğu durumlarda birden fazla makineye işlerin dağıtılması ek bir problemdir. Paralel makinelerin çizelgelenmesi probleminde karşılaşılan bir başka durum ise iş ayırma veya birleştirme problemidir. Üretim sistemine bağlı olarak, bazı

durumlarda bir iş birden fazla makinede tamamlanabilmekte veya aynı özellikteki birkaç sipariş birleştirilerek bir makinede aynı anda üretilebilmektedir. İşlerin bu şekilde birleştirilmesi veya ayrılması durumunda, bu işlerin teslim tarihlerine uygun şekilde en iyi çizelgenin hazırlanması ek bir problem haline gelmektedir.

## 2.2. Çalışmalar ve Sınıflandırmalar

Çizelgeleme problemleri çeşitli benzetim çalışmaları, şebeke analizleri ve sezgisel çözüm yaklaşımlarının geliştirilmesi ve uygulanmasına odak noktası olmaktadır. Uygun tekniklerin seçimi; problemin karmaşıklığına, modelin yapısına ve amaca göre değişmektedir. Çizelgeleme problemlerinin sınıflandırılması; atölye tipine, işin geliş şekline, makine sayılarına ve performans ölçütlerine göre değişik şekillerde yapılmaktadır (Baker, 1974).

Çözüm yaklaşımları ise, problem yapısı ve atölye şekline bağlı olarak değişir. Eğer işlerin süreleri ve diğer parametreler kesin belirlenebilir ise deterministik, aksi halde stokastik çizelgeleme problemi sözkonusudur. Stokastik parametreler rasgele değişkenler şeklinde ele alınır. Bu değişkenler bilinen bir olasılık dağılımına göre dağılır. Bu modelin en belirgin özelliği parametrelerin davranışını yansıtan dağılıma uygun olmasıdır. Stokastik modeller gerçek uygulamalarda deterministik modellere göre daha iyi sonuç verir (Baker, 1974). Sisteme gelen işler tek bir işleme ihtiyaç duyuyor ise buradaki problem tek makineli çizelgeleme problemidir ve işlerin hangi sırada yapılacağına belirlenmesine çalışılır. Paralel makineli problemlerde sisteme gelen işler mevcut makinelerin herhangi birinde yapılabilir. Seri akışlı ve karmaşık akışlı problemlerde ise atölyeye gelen işler birden fazla işleme ihtiyaç duyar. Seri akışta tüm işlerin rotası aynı iken karmaşık akışlı problemlerde her bir işin rotası farklıdır (Baker, 1974).

Araştırmacılar dal-sınır (Sen and Gupta, 1983), dinamik programlama (Van and Gelders, 1978) ve iş çiftlerinin yer değiştirmesi tekniklerini çizelgeleme problemlerinin çözümünde yaygın olarak kullanmışlardır. Bu teknikler, çizelgeleme problemlerinin dahil olduğu çok geniş sayıdaki kombinatoriyal problemlerin çözümünde dikkate değer bir başarıya sahip bulunmaktadır.

Kombinatoryal eniyileme problemleri, “dikkate alınan amaç fonksiyonunu eniyileyen kesikli karar değişkenlerinin değerlerinin bulunması” ile ilgilenir.

NP-zor sınıftaki problemlerin çözümü için polinom zamanlı algoritma yoktur. 0 ya da 1 değerini alan  $n$  değişkene sahip bir problem için tüm çözümlerin birerleme (enumeration - sayımlama) zamanı  $O(2^n)$ 'dir. Küçük boyutlu problemler birerleme ile çözülebilmelerine rağmen, büyük boyutlu problemler için bu yöntem ile çözüme ulaşmak mümkün değildir. NP-zor sınıftaki problemler için dal-sınır ya da kesme-düzlemi gibi etkin yöntemlerin başarısız olmasının nedeni, bu yöntemlerinde üstel sınırlara sahip olmasıdır. NP-zor sınıfta yer alan bu gibi kombinatoryal eniyileme problemlerine örnek olarak, gezgin satıcı, araç rotalama, çizelgeleme ve yer seçimi problemleri verilebilir.

Paralel makine çizelgeleme, belirli amaç fonksiyonlarını sağlamak amacıyla, aynı işlevli bir dizi makineye işleri çizelgeleme problemidir. Bu problemler de kombinatoryal eniyileme problemleri sınıfının üyesidir ve NP-zor olarak bilinirler (Jeng and Lin, 2007).

Dal-sınır ve dinamik programlama tekniklerinin her ikisi de birerleme teknikleridir. Bu teknikler belli kısıtlayıcı kuralları akılcı bir şekilde uygulayarak çok sayıda aday çözümü elimine ederler. Ancak bu iki teknik de büyük boyutlu problemler için etkin değildir. Dinamik programlamada durum değişkenlerinin sayısı artarken problemleri çözmek için gereken işlemler de artar ve bu özellik büyük boyutlu problemlerin çözümünde dinamik programlama yaklaşımının kullanımını kısıtlar. Dal-sınır yaklaşımında çözüm zamanları, farklı veri setlerine göre önemli derecede değişkenlik gösterir. Dallanan değişken ile sınırlama yaklaşımının seçimi algoritmanın performansını önemli derecede etkiler.

Birçok araştırmacı, çizelgeleme problemlerinin değişik versiyonları için tamsayılı programlama modelleri geliştirmişlerdir. Bir çizelgeleme problemi tamsayılı programlama modeli olarak formüle edilebileceği için mevcut tamsayılı programlama algoritmalarıyla çözümü mümkündür. Ancak böyle bir yaklaşım sadece küçük ölçekli problemlere uygulanabilir. Çizelgeleme problemlerinin matematiksel programlama formülasyonu genellikle çok sayıda değişken ve kısıta ihtiyaç duyar. Mevcut tamsayılı programlama algoritmaları bu tür problemleri etkin bir şekilde çözmede başarılı değildir.

Kombinatoryal eniyileme problemleri için geliştirilen yöntemlerden bir diğeri de sezgisel yöntemlerdir. Eniyi çözümlerin elde edilmesindeki zorluk dolayısı ile çok sayıda problem için özel sezgisel teknikler geliştirilmiştir. Bu sezgiseller, çözüm kalitesi ve hesaplama karmaşıklığı arasındaki ödünleşmeleri dikkate alacak şekilde tasarlanmıştır. Sezgisel algoritmalarda ise eniyi sonuç bulunamamasına rağmen hesaplama süresi daha az olmaktadır (Blum ve Roli, 2003).

Eniyi çözümü elde etmek için gerekli olan hesaplama işlemleri açısından değerlendirildiğinde, kombinatoryal problemler oldukça zor problemlerdir. Bundan dolayı, büyük boyutlu problemlerin çözümünde uygun hesaplama süresi içerisinde eniyeye yakın çözümleri bulabilen yaklaşık (sezgisel) algoritmaları geliştirmek ve kullanmak önemli hale gelmiştir. Bu yüzden zeki ve etkili sezgisel yaklaşımların zor problemlerin çözümünde kullanılması üzerinde oldukça fazla araştırma yapılmıştır (Karaboğa, 2004).

Literatürde bölgesel eniyileme yöntemleri olarak da geçen klasik sezgisel yöntemlerde, çözüm uzayında arama, belirlenen komşuluk yapısı ile daha iyi bir komşu çözüm bulunamadığı durumda sonlandırılmaktadır. Bu sebeple bu yöntemler yerel enküçük noktalara takılmakta ve arama stratejisi kör bir şekilde uygulanmaktadır. Metasezgisel yöntemler ise yerel enküçük noktalardan kurtulmak için daha kötü çözümlerin de kabul edildiği bütünsel eniyileme yöntemleridir. Bu yöntemlerde arama, çözüm uzayının en umut verici noktalarında yapılmakta ve süreç, algoritma içine gömülü bir arama stratejisiyle yönetilmektedir. Bu yöntemlerin en büyük sakıncası ise algoritmanın ne zaman duracağını bilmemesidir (Breedam, 2001). Metasezgisel yöntemlerde klasik sezgisellere göre daha kesin çözüm bulunmasına ve yeni kısıtların algoritmaya kolay bir şekilde entegre edilebilmesine rağmen algoritmaların yazılım diline kodlanması karmaşık ve çözüm süreleri oldukça uzun olmaktadır (Tarantilis vd., 2004).

Çalışmada yukarıda bahsedilen nedenlerle; matematiksel model ve boyut analizinin ardından, özdeş paralel makinelerde çizelgeleme problemi (Ö.P.M.Ç.P.) için sezgisel bir çözüm yaklaşımı geliştirilmiştir. Ayrıca problemde hazırlık süreleri de göz önünde bulundurulmuştur.

Bir ürün çevriminin üretimini gerçekleştirmek için makine, proses veya tezgahları üzerinde yapılan işlemler hazırlık faaliyetleri olarak ifade edilir. Bunlar

gerekli ekipmanların tespiti, temini, ayarlanması, takılması gibi faaliyetlerden oluşur. Literatür incelendiğinde, erişilebilen çalışmaların bir kısmında hazırlık süresinin ihmal edildiği ya da bu sürelerin sıraya bağlı olmadığı durumlarda işlem sürelerine eklendiği görülmektedir. Ancak bazı problemlerde hazırlık süreleri ihmal edilemeyecek kadar önemlidir ve işlem sürelerinden ayrı olarak değerlendirilmeleri gerekir.

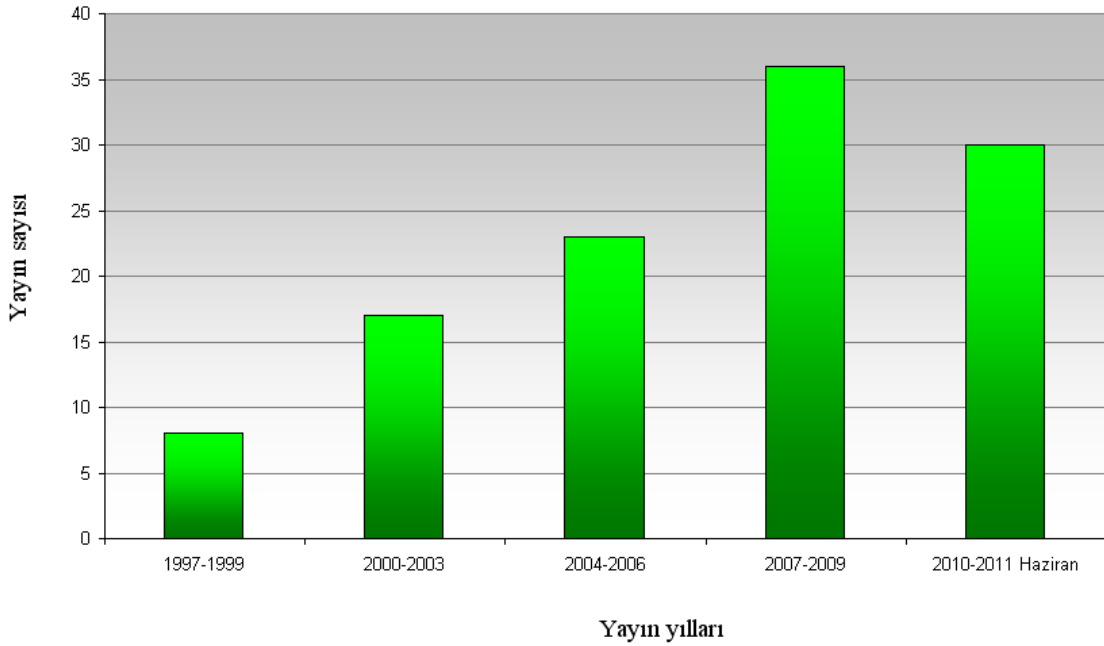
Hazırlık işlemleri ile ilgili problemler literatürde iki sınıfta ele alınmıştır. Birincisinde, hazırlıklar sadece işlem görecektir işe bağlı olup sıra-bağımsız hazırlık süresi olarak ifade edilir. Diğerinde ise hazırlık, hem o anda işlem görecektir işe hem de bir önceki işe bağlıdır. Bu durum ise sıra-bağımlı hazırlık süresi olarak ifade edilir (Allahverdi et al., 1999).

Çalışmada hazırlık süresi bilgisi, işler arasındaki geçişlerde yaşanan kalıp değişikliğinden kaynaklanmaktadır. Aynı türden kalıbı kullanan ürünler ilgili makinede ardi ardına işlenecekse, herhangi bir hazırlık süresi oluşmamaktadır. Ancak farklı kalıp kullanımını gerektiren ürünler arasındaki geçişlerde hazırlık süresi oluşmaktadır. Bu durum sıraya bağlı hazırlık süresinin varlığını göstermektedir ve problemin çözümünü zorlaştırmaktadır. Her farklı sıralamada, farklı hazırlık süreleri gerekebileceğinden, her bir alternatif sıralamanın ayrıca değerlendirilmesi gerekmektedir. Bu çalışmada da tek performans ölçütlü (toplam gecikmeyi enküçükmek), özdeş paralel makineli, sıra-bağımlı hazırlık süreli çizelgeleme problemi ele alınmıştır. Ele alınan performans ölçütü ise toplam gecikmeyi enküçükmektir.

Literatürde bu problemle ilgili farklı çözüm yaklaşımlarının geliştirildiği gözlenmiştir. Özdeş paralel makineleri konu alan ve 1997 – 2011 yılları arasında yayınlanmış 115 çalışma incelenmiş; yıllarına, çözüm yöntemlerine ve performans ölçütlerine göre dağılımları izleyen bölümlerde verilmiştir.

### 2.2.1. Yıllara göre çalışmalar

Paralel makinelerde çizelgeleme çalışmaları içinde, özdeş paralel makinelerle ilgili olanlar önemli bir yer tutmaktadır. Çalışma kapsamında da özdeş paralel makineleri konu alan yayınlar incelenmiştir. Son yıllarda konu ile ilgili yayın sayısında artış olduğu Şekil 2.1.' de görülmektedir. 2011 yılının 2. yarısında yayınlanması muhtemel çalışmalar da düşünüldüğünde artışın süreceği beklenmektedir. Aynı zamanda literatürde NP-zor problemler içinde yer alan bu konu, farklı çözüm yaklaşımları ile daha etkili çözümler üretebilmek için de araştırmacıların dikkatini çekmektedir.

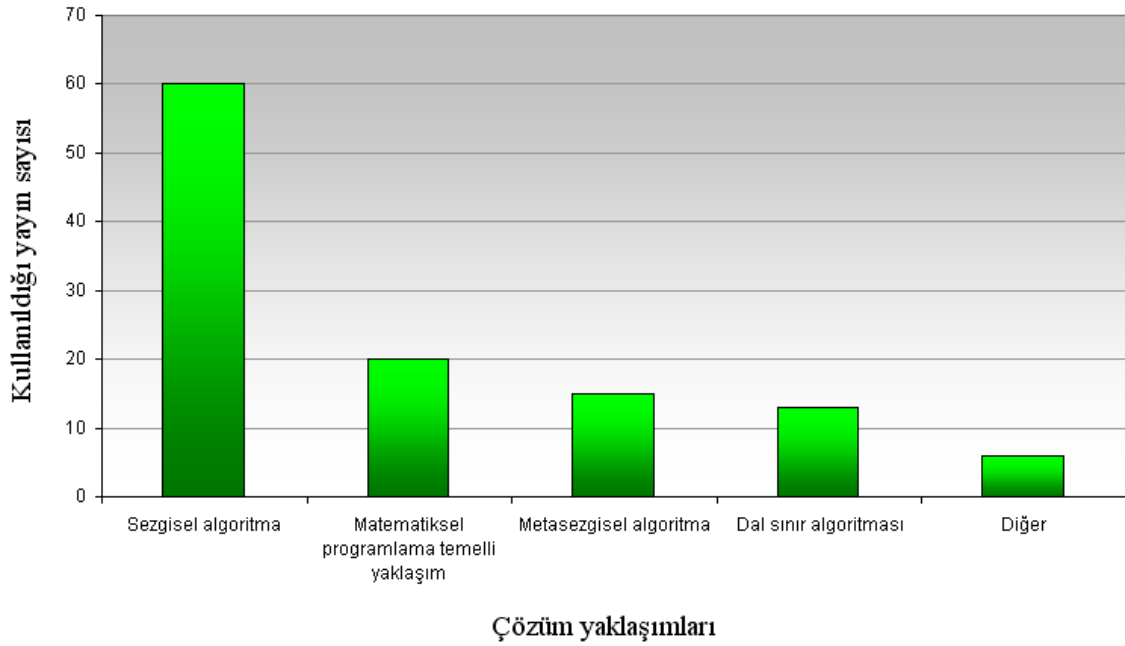


Şekil 2.1. Yıllara göre Ö.P.M.Ç.P. yayın sayıları

### 2.2.2 Çözüm yöntemlerine göre çalışmalar

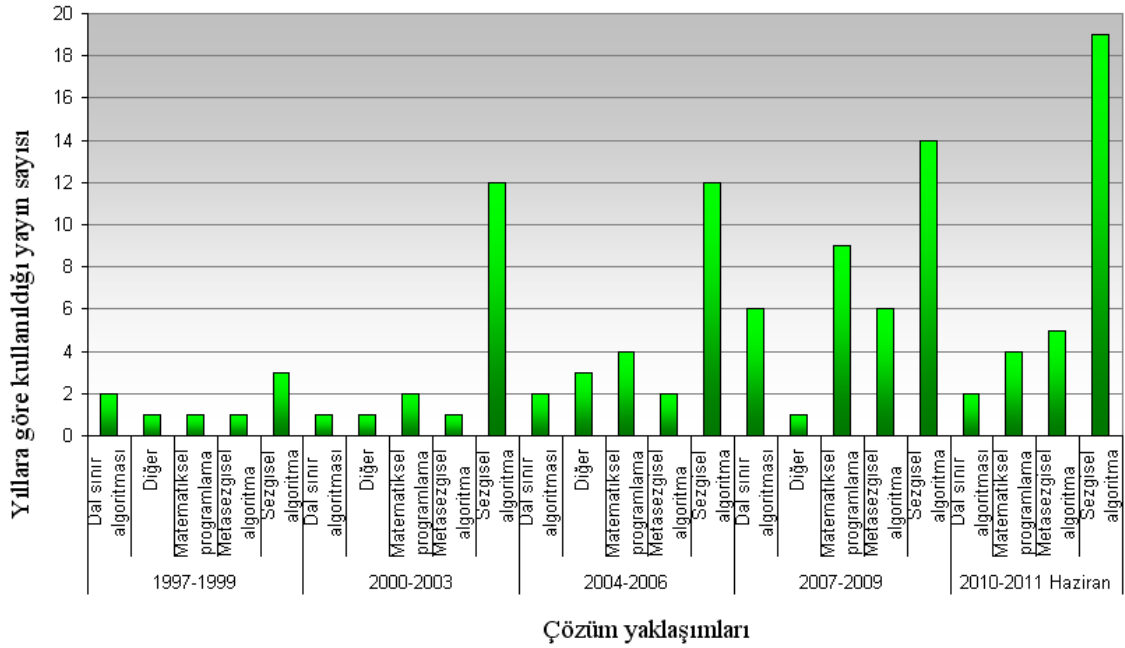
Özdeş paralel makinelerde çizelgeleme probleminin çözümünde dal-sınır algoritması, dinamik programlama gibi analitik yöntemler ile birlikte, sezgisel yaklaşımlar da kullanılmaktadır. Aynı zamanda karma yöntemlerin kullanıldığı çalışmalar da bulunmaktadır. Çözüm yöntemlerine ilişkin dağılım Şekil 2.2' de görülmektedir.





Şekil 2.2. Ö.P.M.Ç.P.'nin çözümü için kullanılan yaklaşımların dağılımı

Çözüm yöntemleri açısından irdelendiğinde, Şekil 2.2.'ye göre sezgisel yaklaşımların ağırlıklı olduğu ortaya çıkmaktadır. Bunda en büyük etken daha önce de belirtildiği gibi Ö.P.M.Ç.P.'nin NP-zor problemler arasında yer almasıdır. Şekil 2.3.'de ise, kullanılan çözüm yöntemlerinin yıllar bazında dağılımı görülmektedir. Sezgisel algoritmaların kullanımı her yıl aralığında, diğer çözüm yöntemlerine göre daha fazla olmakla birlikte, son yıllarda metasezgisel yöntemlerin kullanımının da arttığı görülmektedir.



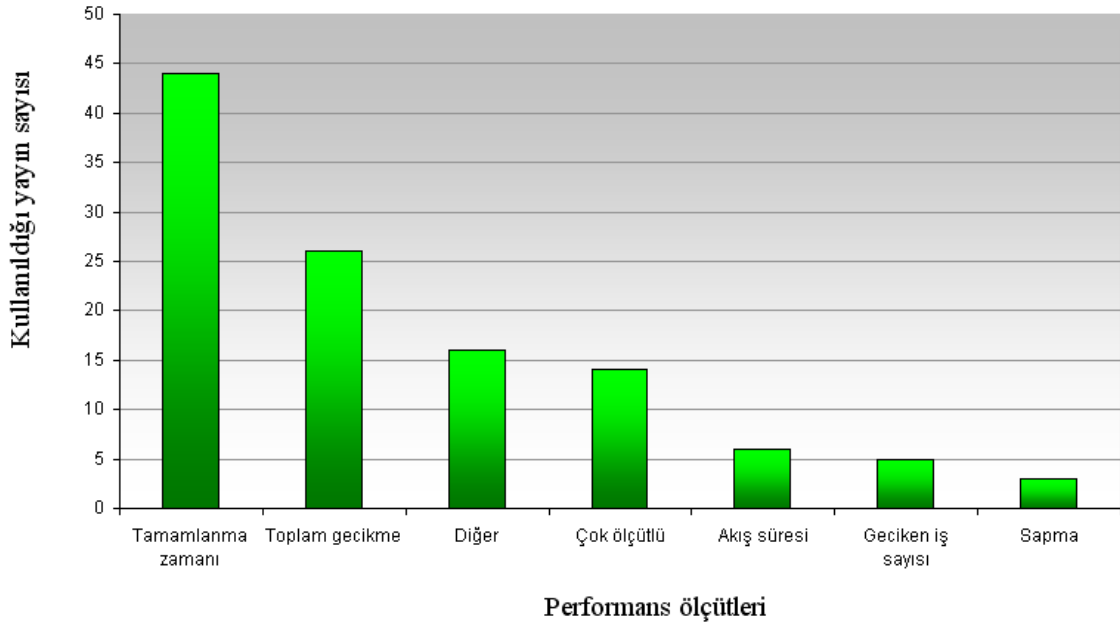
Şekil 2.3. Ö.P.M.Ç.P.' nin çözümünde kullanılan yaklaşımların yıllara göre dağılımı

### 2.2.3. Performans ölçütlerine göre çalışmalar

Çizelgelemenin ne ölçüde başarılı olduğunu değerlendirmek için kullanılan göstergeler, performans ölçütü olarak tanımlanmaktadır (French, 1982). Bu doğrultuda tek bir performans ölçütü kullanılabileceği gibi birden fazla ve birbirleriyle çelişen ölçütler de kullanmak mümkündür (Nelson, et al., 1986).

Yapısına göre problemi çözmek için kullanılan teknikler belirli performans ölçütlerini eniyilemektedirler. Bu nedenle çözüm tekniği belirlenirken performans ölçütünün de dikkate alınması gerekmektedir.

Literatürde yer alan çalışmalar incelendiğinde, çizelgeleme problemleri içerisinde çok sayıda performans ölçütüne rastlanmıştır. Şekil 2.4.' de, en fazla kullanılan performans ölçütlerinin, toplam gecikme süresi ve tamamlanma zamanı olduğu görülmektedir. Çok ölçütlü problemlerin de kullanım oranı yüksektir. Diğer olarak ifade edilen performans ölçütleri grubunda ise farklı olarak toplam hazırlık süresini enküçükleme, tam zamanında yapılan iş sayısını enbüyüklemek gibi kullanımına daha az rastlanmış ölçütler yer almaktadır.



Şekil 2.4. Ö.P.M.Ç.P.' de kullanılan performans ölçütleri dağılımı

Literatürde, çizelgeleme konusunda birçok çalışma yer almaktadır. Ancak ele alınan problemin özelliklerini (özdeş paralel makineler, birden fazla kalıp gözü, sıra bağımlı hazırlık süresi) birebir taşıyan bir çalışmaya rastlanmamıştır.

Problem için geliştirilen sezgisel yaklaşım, küçük boyutlu problemler sözkonusu olduğunda en iyi çözümü veren matematiksel modelle rekabet edebilir görülmektedir. Literatürde bu problemle tam olarak örtüşen bir çalışmaya rastlanamamış olmakla birlikte, gerçekte bu karakteristikte üretim sistemleri bulunmaktadır. Bu sebeple, ilgili problemin çözümü için geliştirilen yaklaşımların literatüre ve gerçek üretim sistemlerine katkısı olacağı umulmaktadır.

### 3. ÖZDEŞ PARALEL MAKİNELERDE ÇİZELGELEME PROBLEMİ İÇİN BİR YAKLAŞIM

Paralel makine çizelgeleme problemlerinde kullanılan matematiksel modeller incelendiğinde, işlerin yapılmasında kullanılan kalıplara ait göz sayısı bilgisine rastlanamamıştır. Ancak bu bilgi doğrudan üretim süresine etki etmekte, tamamlanma zamanını değiştirmektedir.

İşler arasında, sıraya bağımlı hazırlık süresinin olup olmaması durumu söz konusu olabileceği gibi, kimi işlerin kalıp gözlerinin sayısı da değişkendir. Bu durum işlem süresine yansımaktadır. Aynı kalıbın kullanılarak birim zamanda kaç adet ürün verebildiği bilgisi kalıp gözü sayısı bilgisi ile ilişkilidir. Kalıp iki gözlü ise, üretim süresi sonunda iki adet aynı ürün elde edilebiliyor demektir. Üretim süresi, kalıp gözü sayısı arttıkça artmakta fakat birim üretim süresi tek gözlü kalıp ile üretim süresine göre azalmaktadır.

İzleyen bölümde bu bilgiler ışığında geliştirilen matematiksel modele yer verilmektedir. Literatürde yer alan benzer konulu çalışmalarda kullanılan matematiksel modellerden esinlenilerek, yukarıdaki özel durumların entegre edildiği yeni bir model geliştirilmiştir.

#### 3.1. Matematiksel Model, Boyut Analizi ve Sezgisel Algoritma Geliştirme Gereçekleri

Probleme karşılık gelecek matematiksel model; sıra bağımlı hazırlık süreli  $N$  adet işin,  $M$  adet özdeş paralel makinede işlem göreceği, işlerin kullandığı kalıplara ilişkin kalıp gözü sayısının verildiği ve sürece ilişkin kısıtların yer aldığı, toplam gecikmenin enküçülenmek istendiği durum için oluşturulmuştur. Çalışma kapsamında farklı olarak parti büyüklüğü ve kalıp gözü parametreleri de dikkate alınmıştır. Bu kapsamda Biskup, Herrmann ve Gupta (2008) ile Ling-Huey Su (2009)' nun çalışmalarından yararlanılarak ve bahsedilen eklentilerle geliştirilen 0-1 karma tamsayılı matematiksel model aşağıdaki şekildedir:

### Varsayımlar

1. Tüm işler, çizelgeleme işlemi başlangıcında hazır bulunmaktadır.
2. Paralel makineli sistemde gelen işler ( $j=1,2,\dots,n$ ), mevcut paralel makinelerin ( $l=1,2,\dots,m$ ) herhangi birinde işlem görebilir. Makineler kapasite ve kullanım açısından özdeştir.
3. İş geçişlerinde yaşanan kalıp değişim süreleri önceden bilinmektedir. Farklı kalıbı kullanan her bir ürün geçişinde yaşanan hazırlık süresi sabittir.
4. Makineler çizelgeleme periyodu içerisinde çalışmaktadır ve bir makinede aynı anda tek bir iş yapılabilir.

### Kümeler

$N = \{1, 2, \dots, n\}$  iş kümesi

$M = \{1, 2, \dots, m\}$  makine kümesi

### İndisler

$i, j \in N$  iş

$k \in N$  iş sırası

$l \in M$  makine

### Parametreler

$n$  : iş sayısı

$m$  : makine sayısı

$p_j$  :  $j$ . işin işlem süresi (sn)

$h_j$  :  $j$ . iş, birinci sırada üretilirse gerekecek hazırlık süresi (başlangıç hazırlık süresi) (sn)

$s_{ij}$  :  $i$ . işten sonra  $j$ . işin üretimine geçilebilmesi için gerekecek hazırlık süresi (sn)

$G$  : büyük bir reel sayı

$d_j$  :  $j$ . işin terminine göre elde olan süre (sn)

$$v_j = \begin{cases} 1, & j \text{ işinin gerektirdiği kalıp iki gözlü ise} \\ 0, & j \text{ işinin gerektirdiği kalıp tek gözlü ise} \end{cases}$$

$q_j$  :  $j$  işinin sipariş miktarı

### Karar Değişkenleri

$C_j$  :  $j$  işinin tamamlanma zamanı (sn)

$T_j$  :  $j$  işinin gecikme süresi (sn)

$$y_{jkl} = \begin{cases} 1, j. \text{ iş; } l. \text{ makinede, } k. \text{ sıraya çizelgelendiyse} \\ 0, \text{ diğer durumlarda} \end{cases}$$

### Amaç Fonksiyonu

$$EnkZ = \sum_j T_j$$

### Kısıtlar

$$C_j \geq p_j * q_j / (v_j + 1) \quad \forall j \quad j \in N \dots\dots\dots(1)$$

$$C_j + G * (1 - y_{jkl}) \geq h_j + p_j * \frac{q_j}{(v_j + 1)} \\ \forall(i, j, k, l) \quad i \neq j, k = 1, l \in M, (i, j) \in N \dots\dots\dots(2)$$

$$C_j - C_i + G * (2 - y_{ik-1l} - y_{jkl}) \geq s_{ij} + p_j * \frac{q_j}{(v_j + 1)} \\ \forall(i, j, k, l) \quad i \neq j, k > 1, l \in M, (i, j, k) \in N \dots\dots\dots(3)$$

$$\sum_j y_{jkl} \leq 1 \quad \forall(k, l) \quad k \in N, l \in M \dots\dots\dots(4)$$

$$\sum_k \sum_l y_{jkl} = 1 \quad \forall j \quad j \in N \dots\dots\dots(5)$$

$$T_j \geq C_j - d_j \quad \forall j \quad j \in N \dots\dots\dots(6)$$

$$\sum_j y_{jkl} - \sum_i y_{jk-1l} \leq 0 \quad \forall(k, l) \quad k > 1, k \in N, l \in M \dots\dots\dots(7)$$

$$T_j \geq 0 \quad \forall j \quad j \in N \dots\dots\dots(8)$$

$$C_j \geq 0 \quad \forall j \quad j \in N \dots\dots\dots(9)$$

- (1) nolu kısıta göre, bir işin tamamlanma zamanı işlem süresinden büyük ya da eşit olmalıdır.
- (2) nolu kısıta göre, bir iş herhangi bir makinede ilk sıraya yerleştirilmişse, tamamlanma zamanı, işlem süresi ve başlangıç hazırlık süresinin toplamından büyük ya da eşit olmalıdır.
- (3) nolu kısıt, bir iş herhangi bir makinede ikinci ya da daha büyük bir sıraya yerleştirilmişse, tamamlanma zamanının, işlem süresi ve sıraya bağımlı hazırlık süresinin toplamından büyük ya da eşit olması gerektiğini gösterir.
- (4) nolu kısıt, bir makinenin her bir işlem sırasına en çok bir işin atanması gerektiği ile ilgilidir.
- (5) nolu kısıt, her bir işin bir makinenin bir sırasına atanmasını sağlar.
- (6) nolu kısıt, bir işin tamamlanma zamanının termininden büyük olması durumundaki gecikme miktarı (amaç fonksiyonu ile de en küçük değeri hedeflenen)  $T_j$ 'yi ölçen kısıttır.
- (7) nolu kısıt, işlerin ardı ardına atanmasını sağlar.
- (8) ve (9) nolu kısıtlar, sırasıyla karar değişkenlerinin pozitif tamsayı olma koşullarını göstermektedir.

### Matematiksel model için boyut analizi

Bu bölümde, geliştirilen matematiksel modelin, farklı parametre setleri için ulaşacağı boyut (karar değişkeni ve kısıt sayısı açısından) analiz edilmektedir.  $i, j$  çizelgelenecek iş,  $l$  makine sayısı ve  $k$  makinedeki iş sırası iken Çizelge 3.1.(a) ve Çizelge 3.1.(b) sırasıyla  $i, j=10$  ve  $l=2$  ayrıca  $i, j=1000$  ve  $l=10$  parametre değerleri için problemin toplam kısıt ve değişken sayılarını göstermektedir.

Çizelge 3.1.(a) Matematiksel modele ilişkin boyut analizi

(10 iş ve 2 makine için örnek) ( $i, j, k=10, l=2$ )

Kısıt no	İndisler	Kısıt sayısı	Problem için kısıt ve değişken sayıları
1	$j$	$j$	10
2	$i, j, k, l$	$i*(j-1)*l*l$	180
3	$i, j, k, l$	$i*(j-1)*(k-1)*l$	1.620
4	$k, l$	$k*l$	20
5	$j$	$j$	10
6	$j$	$j$	10
7	$k, l$	$(k-1)*l$	18
8	$j$	$j$	10
9	$j$	$j$	10
Değişken	İndisler	Değişken sayısı	
$C_j$	$j$	$j$	10
$T_j$	$j$	$j$	10
$y_{jkl}$	$j, k, l$	$j*k*l$	200
Toplam kısıt sayısı		$5j+k*l*(2+i*(j-1))-l$	1.888
Toplam değişken sayısı		$j*(2+k*l)$	220



Çizelge 3.1.(b) Matematiksel modele ilişkin boyut analizi  
(1000 iş ve 10 makine için örnek) ( $i, j, k=1000$   $l=10$ )

Kısıt no	İndisler	Kısıt sayısı	Problem için kısıt ve değişken sayıları
1	$j$	$j$	1000
2	$i, j, k, l$	$i*(j-1)*l*l$	9.990.000
3	$i, j, k, l$	$i*(j-1)*(k-1)*l$	9.980.010.000
4	$k, l$	$k*l$	10.000
5	$j$	$j$	1.000
6	$j$	$j$	1.000
7	$k, l$	$(k-1)*l$	9.990
8	$j$	$j$	1.000
9	$j$	$j$	1.000
Değişken	İndisler	Değişken sayısı	
$C_j$	$j$	$j$	1.000
$T_j$	$j$	$j$	1.000
$y_{jkl}$	$j, k, l$	$j*k*l$	10.000.000
Toplam kısıt sayısı		$5j+k*l*(2+i*(j-1))-l$	9.990.024.990 (Dokuz milyar dokuz yüz doksan milyon yirmi dört bin dokuz yüz doksan)
Toplam değişken sayısı		$j*(2+k*l)$	10.002.000

Tamsayılı programlama modellerinin formülasyonu, sürekli değişkenli<sup>2</sup> matematiksel modellerin formülasyonuna önemli derecede benzerlik göstermesine rağmen, bu benzerlik bazı durumlarda aldatıcı olabilir. Amaç fonksiyonu ve kısıtların cebirsel ifadesi iki tür modelde aynı gibi durabilmesine rağmen, bazı ya da tüm değişkenlerin tamsayı olmasını sağlayan bazı kısıtların eklenmesi, hesaplama bakımından tamsayılı problemleri daha zor hale getirir. Çoğu tamsayılı programlama modelleri önceki bölümlerde de bahsedildiği gibi NP-zor sınıfına aittir. Dolayısıyla genel doğrusal programlama modelleri polinom zamanda çözülebilirken, aynı modelin tamsayılı çözümünü bulmak, üstel hesaplama zamanı gerektirebilir (Bakır ve Altunkaynak, 2003).

<sup>2</sup>  $[0-\infty)$  aralığında tanımlanan karar değişkenidir.

Geliştirilen matematiksel model, 10 iş ve 2 makineli bir problem için LINGO 11.0 yazılımı ile Intel® Core™ i5 CPU/2.4GHz/4Gb Ram kapasiteli kişisel bilgisayarda çözdürülmüştür. Küçük boyutlu bir problem olmasına karşın, yaklaşık 7 gün çalışmış, EK-1.' de verilen en iyi çözüme ulaşılmıştır. Amaç fonksiyonu değeri 282 birim süre olarak bulunmuştur. 10 iş, 3 makineli problem ise yaklaşık 10 günde çözüme ulaşmıştır. Problemin en iyi çözümü EK-2.' de verilmiştir. 15 iş, 3 makineli problem ise iki haftada çözülememiş ve süreç durdurulmuştur.

Gerçek hayatta karşılaşılabılır problemlerde de, çizelgelenecek binlerce iş, kullanılabilir onlarca makine olabilir. Bu durumda matematiksel model kullanarak en iyi çözüme kısa sürede ulaşmak neredeyse imkansız olacaktır. Problem boyutu arttıkça, çözüm süresinin üstel olarak artıyor olması -Çizelge 3.1.(b)' deki verilerden de problem boyutu görüleceği gibi- en iyi çözümü veren matematiksel model yerine problemin en iyi çözümünü vermeyen ancak en iyiye yakın çözümü kısa sürede veren sezgisel algoritma kullanımını gerektirmektedir.

Karaboğa (2004) da, yukarıdaki gerekçeleri destekleyerek, sezgisel algoritmalara gerek duyulmasının sebeplerini aşağıdaki gibi belirtmiştir:

- Eniyileme problemi kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar verici açısından çok daha basit olabilir.
- Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt eniyi çözümden daha büyük hatalara sebep olabilir.

### 3.2. Geliştirilen Sezgisel Algoritma Temelli Çözüm Yaklaşımı

Önceki bölümde de görüldüğü gibi bazı problem türleri için en iyi çözümü araştırmak çok fazla işlem yükü ve bilgisayar zamanı gerektirebilmektedir. Ayrıca, bir çözüm yaklaşımı olarak matematiksel model, değişiklikler olduğu durumda anlık çözümler üretebilme konusunda çekiciliğini de yitirmektedir. Bu nedenle, en iyiye yakın çözümler veren sezgisel yöntemler kısa zamanda çözüm elde edilebilmesi sebebiyle tercih edilebilirler. Kullanılacak sezgisel yaklaşım; genetik algoritmalar, tavlama benzetimi ve sinir ağları gibi yapay zekâ teknikleri arasından seçilebileceği gibi probleme özel bir algoritma geliştirme yoluna da gidilebilir.

#### 3.2.1. Mevcut ve önerilen algoritmalar

Aşağıda, geliştirilen sezgisel algoritmanın çalışma mantığı açıklanmaktadır. İzleyen kısımlarda algoritmanın adımları sistematik olarak da yer almaktadır.

Algoritma, işleri, termin bilgileri küçükten büyüğe olacak şekilde sıralama ile başlar. Her bir iş için öncelikle hangi makineye atanabileceği kararı verilir ve bu durumda toplam gecikme (varsa) hesaplanır. Hangi makinedeki toplam gecikme daha az ise işin o makineye ataması gerçekleşir. Makineye yapılabilecek olası atamalar araştırılırken, aynı zamanda aynı kalıbı kullanan işlerin olup olmadığı da göz önünde bulundurulur. Eğer atama yapılabilirliği araştırılan makinede, atanacak ürün ile aynı kalıbı kullanan ya da aynı partiden ürün varsa bu ürünün de kendisine en yakın terminli olan ürünle birlikte üretilmesi durumu dikkate alınmaktadır. Tüm bu sınamalar sonunda işin hangi makineye atanması durumunda gerçekleşecek toplam gecikme en küçük ise o makineye atama gerçekleşir.

Çalışmada, termini aynı olan ürünler içinde işlem süresi en küçük olan iş öne alınarak toplam gecikmeyi en küçükleme amacı ile birlikte ortalama akış süresi de en küçüklene çalışılmaktadır. İş merkezine gelen işler arasından en kısa işlem süresine sahip olan işlerin önceliklendirildiği “en kısa işlem süreli işi önce yap” (Shortest Processing Time - SPT) kuralı ile üretim sisteminin çıktısı enbüyüklenmekte ve geciken işlerin oranı azalmaktadır (Saad, et al., 1997). Termini aynı olan ürünlerin önceliklendirilmesi bu kural dâhilinde yapılarak, ortalama akış süresi de iyilenmeye

çalışılmaktadır. Bahsedilen işlem süresi bilgisi, ilgili ürünün, üretimi istenen adet miktarı ile üretim süresinin çarpımından oluşur.

Nihai ürünün teslim tarihi, ürün üzerindeki diğer işlemlerin yapılacağı prosesler dikkate alınarak belirlendiğinden, prosesler arası geçişlerin zamanında yapılması da önem taşımaktadır. Diğer prosesleri göz ardı edip, tek bir proste gecikmeleri enküçükleme yanlışı olacağından, geliştirilen sezgisel algoritmada “asıl termin” ifadesi ile, ilgili işlem için, yukarıda açıklandığı gibi bir termin kastedilmektedir. Bir başka deyişle bir iş, bir sonraki aşamaya tüm süreç sonunda nihai ürünün termininde gecikmeye yol açmayacak erkenlikte geçmelidir. Bu iş için asıl termin, bir sonraki prosese yetişmesi gereken tarih olacaktır. Bu tarih, işletmede çalışılabilir sürede beklenen makine arızaları vb. bazı tahminler ile belirlenebilir. Bu sebeple, asıl terminine çizelgeleme işlemini yapacak kullanıcı karar verecektir. İlerleyen bölümlerde bu konuya ayrıntılı olarak değinilecektir.

Geliştirilen algoritma, literatürde yer alan çalışmalardan farklı olarak, işlerin üretiminde kullanılan kalıpların kaç gözlü olduğu bilgisini de dikkate almaktadır.

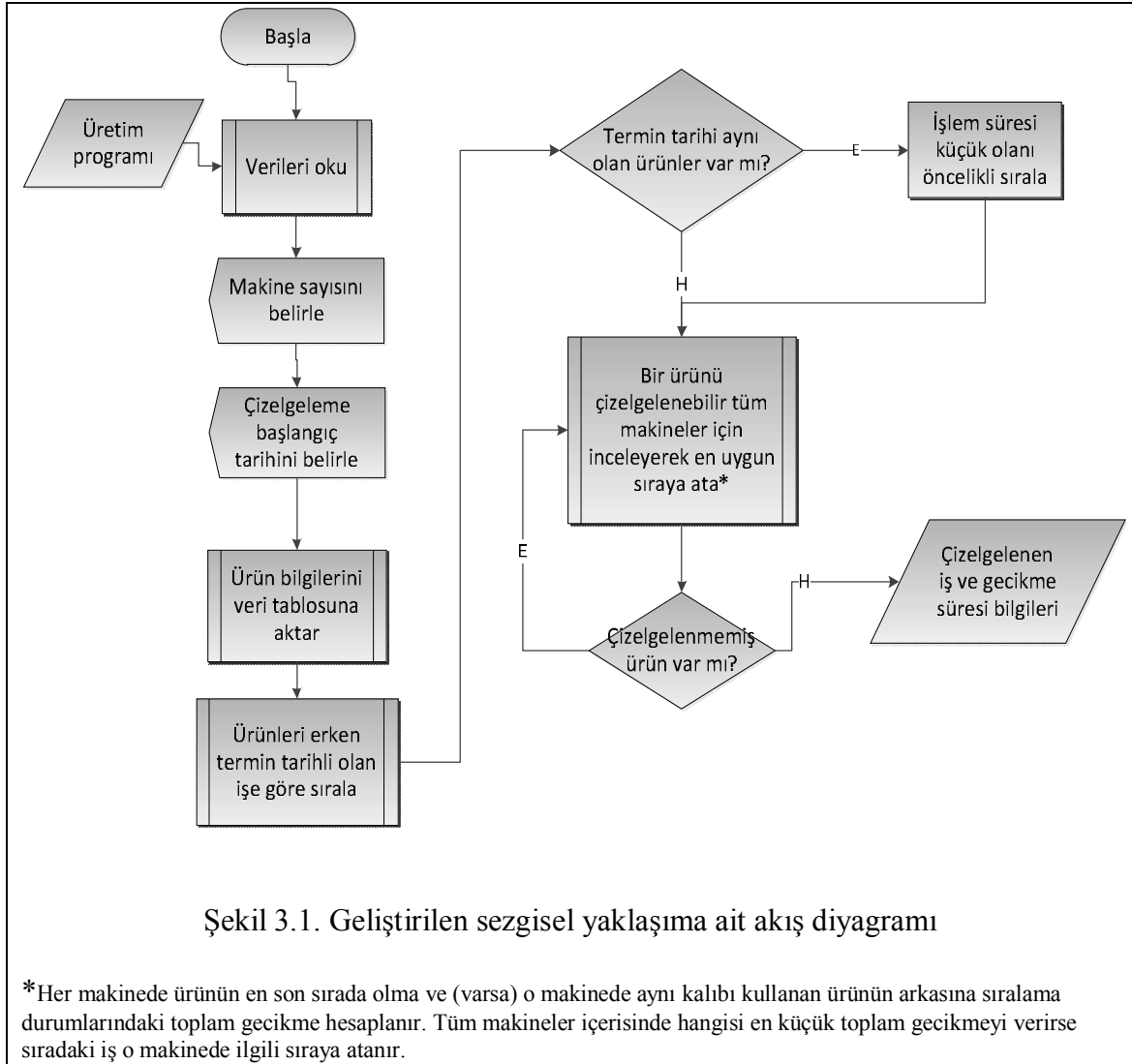
İşletmelerde çizelgeleme işlemi için ilgili sürecin gerektirdiği bazı özel durumları dikkate alan, farklı sezgisel yaklaşımlar kullanılabilir. Talepteki değişikliklere bağlı olarak üretim adetlerinde değişiklik olabilir, kimi ürünlerin termini öne çekilebilir ya da ötelenebilir. Tüm bu gerçekleşebilecek ani değişikliklere, anlık çözümler verebilecek bir sistemin oluşturulması gerekmektedir. Aksi takdirde problemin çözüm süresi uzayabilir, ya da sonuç alınamayabilir. Çalışmada geliştirilen sezgisel algoritma; üretim için kalıplar gerektiren, bazı işlerin aynı kalıbı kullanabildiği, en az gecikme süresini amaçlayan çizelgeleme problemine çözüm sunmaktadır. Bu tür yaklaşımlar bir algoritma kapsamında yer alıp sistematik olarak ifade edilebildiğinde planlama ve çözüm süreci uzmana bağımlı olmaktan çıkabilecektir.

İşlerin terminine yetiştirilmesi düşüncesine dayanan ve uygulama yapılan işletmede hali hazırda kullanılmakta olan mevcut yöntem, termini en önce olan işin önce yapılması (Early Due Date - EDD) kuralını temel almaktadır. İşlem görmek için sırada olan işlerin gecikmelerini en küçüklemeyi amaçlayan bu kurala göre sisteme gelen işler arasından en erken teslim tarihine sahip olan iş ilk olarak yapılmaktadır (Pinedo and Chao, 1999). Bu kurala göre, işlerin, söz verilen tarihlerde teslim edilme performansları artmaktadır (Saad, et al., 1997). Bu yöntem ile mevcut süreçte elde

edilen gecikme süreleri toplamı ile geliştirilen sezgisel algoritmanın sonuçları karşılaştırılmış ve geliştirilen algoritmanın EDD kuralı ile sıralamaya göre daha iyi sonuç verdiği görülmüştür.

Ayrıca, önceki bölümlerde bahsedildiği gibi, çizelgeleme problemleri NP-zor problemler sınıfında yer almaktadır. Bu durum, problem boyutunun büyümesi ile çözüm süresinin üstel olarak artmasına ve en iyi çözüme ulaşılmasının zorlaşmasına neden olmaktadır. Bu sebeple, en iyi çözüm yerine en iyiye yakın çözümler elde edebilme yoluna gidilmektedir. Bunun için sezgisel çözüm yaklaşımları geliştirilmektedir.

Algoritma özet olarak, üretimi, termin bilgilerine göre gecikmeye yol açmayacak ve toplam iş gecikmelerini en küçükleyecek şekilde sıralamaktadır. Gecikme oluşması durumunda, bu gecikmeyi önleyebilecek farklı seçeneklerin olup olmadığını araştırmaktadır. Gecikmenin kaçınılmaz olması durumunda ise ilgili işin diğer tüm makinelere atanması ile elde edilen varsa alternatif gecikmeleri hesaplamakta, buna göre hangi makinede gecikme daha az ise o makineye atamayı gerçekleştirerek, toplam gecikmenin enazlanmasını sağlamaktadır. Önerilen sezgisel algoritmanın adımlarına ilişkin akış şeması Şekil 3.1.' de verilmiştir.



### Algoritmanın adımları

İzleyen kesimde önce, işletmede mevcut durumda kullanılan yaklaşıma karşı gelen (bu çalışma kapsamında adımları tariflenmiş), daha sonra ise bu çalışma kapsamında önerilen algoritmaya yer verilecektir. Her ikisinde de kullanılacak notasyon şu şekildedir:

### Parametre ve karar değişkenleri

$P = \{i \mid i = 1, 2, \dots, n\}$  İş indis kümesi

$SP = \{k \mid k = 1, 2, \dots, s\}, s \leq n$  Çizelgelenmiş iş indis kümesi

$USP = \{z \mid z = 1, 2, \dots, t\}, t \leq n$  Çizelgelenmemiş iş indis kümesi

$SP \cup USP = P, SP \cap USP = \emptyset$

$M = \{l \mid l = 1, 2, \dots, m\}$  Makine indis kümesi

*UII*: Uygun iş indisi

*UM*: Uygun makine indisi

*TG*: Uygun makinede gerçekleşen toplam gecikme süresi (sn)

*Toplamgecikme*: Gecikme süreleri toplamı (sn)

*hazırlık<sub>ki</sub>*: *k*. işten *i*. işe geçildiğinde gereken hazırlık süresi (sn)

*q<sub>i</sub>*: *i*. işin sipariş miktarı (adet)

*p<sub>i</sub>*: *i*. işin birim işlem süresi (sn)

*v<sub>i</sub>*: *i*. işinin gerektirdiği kalıp gözü sayısı

*C<sub>i</sub>*: *i*. işinin tamamlanma zamanı (sn)

$$C_i = C_k + \text{hazırlık}_{ki} + \frac{q_i * p_i}{v_i}$$

$\{i \in P, k \in SP\}$

*termin<sub>i</sub>*: *i*. işin tamamlanması gereken tarih (teslim zamanı)

*t<sub>i</sub>*: *i*. işin gecikme süresi (sn)

$$t_i = \max \{0, C_i - \text{termin}_i\}$$

$e$  : Çizelgelemek üzere seçilmiş iş indisi

$em$  : Atama yapmak üzere seçilen makine indisi

$se$  : Atama yapılmak üzere seçilmiş makinede, çizelgelemek üzere seçilmiş iş ile aynı kalıbı kullanan iş indisi

$j$  : Çizelgelemek üzere seçilmiş işin, atama yapmak üzere seçilen makinedeki yer indisi

$isyuku_l$  :  $l$ . makineye atanmış son sıradaki işin tamamlanma zamanı

$$isyuku_l = C_i \quad \begin{matrix} l \in M \\ i \in P \end{matrix}$$

++: değeri 1 arttır

### ***Erken teslim zamanlı iş öncelikli mevcut sıralama yaklaşımı:***

Geliştirilen algoritmaya geçmeden önce hali hazırda işletmede kullanılan çizelgeleme yaklaşımı tanıtılmaktadır. Bu yaklaşım mevcut sistemde algoritmik bir yapıda tarifli durumda değildir. Fakat bu çalışma kapsamında, hali hazırda kullanılmakta olan bu yöntemin adımlarına karşı gelen aşağıdaki algoritma tanımlanmıştır. Bu sayede geçmiş aylara ilişkin kullanılmış olan mevcut çizelgeler daha sistematik ve daha çok sayıda türetilmiştir. Bu durum istendiği kadar gerçek çizelge üretmeyi de sağlamıştır. Çünkü sistemde geçmiş verileri istenen formatta ve yeterlilikte elde etmek mümkün olmamıştır.

Adım 1: Veri okuma ve gerekli bilgilerin türetilmesi

1.1. Ürünlere ait model, kalıp, kalıp gözü sayısı, birim üretim süreleri, talep ve termin bilgilerini oku,

1.2. Okunan verileri, *USP* kümesine ata,

Adım 2: Okunan verilerin sıralanması

2.1. *USP* kümesindeki işleri, termini en yakından en geçe doğru olacak şekilde sırala,

2.2. Termini aynı olan işler arasında, üretim süresi daha kısa olan işi daha erkene sırala,



### Adım 3. Çizelgeleme

3.1. Çizelgelenecek işi (termini enküçük olan sıradaki iş) seç,

$$e = \arg \min_{\{i \in USP\}} \{termin_i\}$$

3.2. İş yükü en az olan makineyi (tamamlanma zamanı en küçük olan makine) seç,

$$em = \arg \min_{l \in M} \{isyuku_l\}$$

3.3. Seçilen işi, seçilen makinede çizelgelenmiş en son işin yanına ekle,

$$USP = USP / \{e\}, \quad SP = SP \cup \{e\}$$

$$isyuku_{em} = C_e$$

3.4.  $s(USP) > 0$  ise Adım 3' e git.

### Adım 4. Geciken iş kontrolü ve toplam gecikmenin hesaplanması

4.1.  $t_i \geq 0$  ise gecikme vardır, çizelgelenmiş işler için gecikmeleri hesapla,

4.2. Çizelgelenmiş işlerin gecikme sürelerini topla,

$$Toplamgecikme = \sum_{i \in SP} t_i$$

Bu algoritma, işletmelerde teslim tarihi öncelikli olan işin önce yapıldığı mevcut durumu örnelemektedir ve işletmeye özgü kimi kısıtlar altında farklı yaklaşımlar uygulanabilmektedir ve genelde kullanılan yaklaşım da erken teslim tarihli işi öncelikli olarak sıralamaktır. Ancak bu yaklaşımdan daha etkili sonuç veren bir başka yaklaşım tarafımızca önerilmektedir.

### Önerilen yaklaşım:

Geliştirilen sezgisel algoritmanın izleyen adımları, toplam gecikmenin enazlanmasını hedefleyen bir yaklaşımı temel almaktadır.

Adım 1: Veri okuma ve gerekli bilgilerin türetilmesi

1.1. Ürünlere ait model, kalıp, kalıp göz sayısı, birim üretim süreleri, talep, termin bilgilerini oku,

1.2. Okunan verileri,  $USP$  kümesine ata,

Adım 2: Okunan verilerin sıralanması

2.1.  $USP$  kümesini, termin bilgileri küçükten büyüğe doğru olacak şekilde sırala,

2.2. Termini aynı olan işler arasında, üretim süresi küçük olan iş öncelikli olacak şekilde sırala,

Adım 3. Çizelgeleme

3.1. Çizelgelenecek işi (termini en küçük olan sıradaki iş) seç,

$$e = \arg \min_{\{i \in USP\}} \{termin_i\}$$

3.2. Uygun makine bul,

3.2.1. Başlangıç:

$$UM=0, UIH=0, l_{l \in M} = 0, j=0,$$

$TG$ = büyük bir reel sayı

3.2.2.  $j++$ ,  $l++$ ,  $e$  ürününü al.  $l$ . makineye atanmış son işin yanına ata.

$$USP = USP / \{e\}$$

$$SP = SP \cup \{e\}$$

$$isyuku_l = C_e$$

$l \in M \quad e \in P$

3.2.3.  $j$  yer indisine, o makinedeki toplam iş sayısını ata.

$l$ . makineye atanmış işlerin gecikme sürelerini hesapla.

$$j = s(SP)_l$$

$$Toplamgecikme_l = \sum_{i \in SP} t_i$$

Eğer  $Toplamgecikme_l < TG$  ise

$UII=j$ ,  $TG = Toplamgecikme_l$  ve  $UM=M_l$  olur.

3.2.4.  $e$  işini  $l$ . makinede yerleştirildiği sıradan çıkar ve aynı makinede  $e$  tipinde iş (aynı kalıbı kullanan iş ya da aynı iş) ara, -varsa- kendi terminine en yakın olan işin  $(se)_{sira}$  yanına koy.

Bu durumda  $l$ . makineye atanmış işlerin gecikme sürelerini hesapla.

$$j=sira+1$$

$$Toplamgecikme_l = \sum_{i \in SP} t_i$$

Eğer  $Toplamgecikme_l < TG$  ise

$UII=j$ ,  $TG = Toplamgecikme_l$  ve  $UM=M_l$  olur.

3.2.5.  $e$  işini  $M_l$ ' den çıkart.  $l \leq m$  ise Adım 3. 2. 2' ye git.

(Bu sayede, son olarak en iyi sonucu veren makinenin bilgilerine ulaşılmış olur.)

3.3.  $e$  işini, uygun makinenin uygun iş indisine ata.

$$isyuku_{UM} = C_e, e \text{ işi uygun makineye atanmış son iş olabilir}$$

$$isyuku_{UM} = C_i, e \text{ işi uygun makinede ara yerlere atanabilir}$$

Adım 4. Geciken iş kontrolü

$$t_i = C_i - termin_i \geq 0 \text{ ise gecikme vardır.}$$

Adım 5. Toplam gecikme hesabı

5. 1. Çizelgelenmiş işler için gecikmeleri hesapla,

$$Toplamgecikme = \sum_{i \in SP} t_i$$

Yukarıda adımları açıklanan ve bu çalışma kapsamında geliştirilen sezgisel yaklaşım, 10 iş ve 2 makineli ( $M_1$  ve  $M_2$ ) bir problem üzerinde örneklenecek olursa 10 işe ait kalıp, üretim süresi ve termin bilgileri Çizelge 3.2.' de verilmiştir. Örnek problemde, işlem kolaylığı ve anlaşılabilirliği arttırmak amacıyla, süre ve termin bilgileri aynı birimden olup saat olarak alınmıştır. Örnek olarak termin bilgisinin "10" olması, 10. saatte işin teslim edilmesi gerektiğini ifade etmektedir. Farklı kalıbı kullanan ürünler arasındaki geçişlerde ise 2 saat hazırlık süresi gerekmektedir.

Çizelge 3.2. Örnek veri seti (10 iş ve 2 makine)

No	İş	Kalıp	Süre (sa)	Termin (sa)
1	a	x	2	10
2	b	y	4	12
3	c	x	2	9
4	d	x	5	13
5	e	y	6	14
6	f	y	3	11
7	g	x	3	10
8	h	y	4	10
9	ı	z	1	15
10	j	x	2	16

Önerilen algoritma ile çizelgeleme şu şekilde gerçekleşir:

- İlk olarak, işler termini en erkenden en geçe olacak şekilde Çizelge 3.3.' deki gibi sıralanır.

Çizelge 3.3. Örnek problem verilerinin termin bilgisine göre sıralanmış şekli

No	İş	Kalıp	Süre (sa)	Termin (sa)
3	c	x	2	9
1	a	x	2	10
7	g	x	3	10
8	h	y	4	10
6	f	y	3	11
2	b	y	4	12
4	d	x	5	13
5	e	y	6	14
9	ı	z	1	15
10	j	x	2	16

- En erken termin tarihli c işi,  $M_1$ ' de son iş olarak atanır. Toplam gecikme hesaplanır.

$[işlem süresi(c) = 2 < termin(c) = 9]$ , Toplam gecikme=0' dır.

Aynı makine üzerinde önceden ataması gerçekleşmiş,  $c$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $c$  işinin atandığı yer değişerek,  $c$  işi bu yeni işin yanına atanır. Bu durumdaki toplam gecikme hesaplanır. Öncesinde ataması gerçekleşmiş başka bir iş yoktur.  $TG_{M1}=0$

- $c$  işinin  $M_2$  makinesinde son iş olarak atanması halindeki toplam gecikme hesaplanır.

$[işlem süresi(c) = 2 < termin(c) = 9]$ , Toplam gecikme=0' dır.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $c$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $c$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Öncesinde ataması gerçekleşmiş başka bir iş yoktur.  $TG_{M2}=0$

$M_1$  ve  $M_2$  makinelerinde toplam gecikme değeri 0' dır. Ancak  $M_1$  makinesinin seçilmemesi için o makinedeki gecikmeden daha az gecikme veren başka bir makine olması gerektiğinden,  $c$  işi  $M_1$ ' e atanır.

- Sıradaki termini küçük olan a işinin,  $M_1$ ' de son iş olması durumundaki toplam gecikme hesaplanır.

$[işlem süresi(c) + işlem süresi(a) = 4] < [termin(a) = 10]$

Toplam gecikme=0' dır. Aynı zamanda, yine aynı makine üzerinde önceden ataması gerçekleşmiş ve  $a$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $a$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır. Bu örnekte böyle bir durum bu aşamada yoktur.  $TG_{M1}=0$

- İzleyen aşama olarak  $a$  işinin  $M_2$  makinesinde son iş olarak atanması durumundaki toplam gecikme hesaplanır.

$[işlem süresi(a) = 2 < termin(a) = 10]$ , Toplam gecikme=0' dır.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $a$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $a$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Öncesinde ataması gerçekleşmiş başka bir iş yoktur.  $TG_{M_2}=0$

$M_1$  ve  $M_2$  makinelerinde toplam gecikme değeri 0' dır. Ancak  $M_1$  makinesinin seçilmemesi için o makinedeki gecikmeden daha az gecikme veren başka bir makine olması gerektiğinden,  $a$  işi ilk makineye ve  $c$  işinden sonraya atanır.

- Sıradaki termini küçük olan  $g$  işi,  $M_1$ ' de son iş olarak ( $a$  ve  $c$ ' den sonraya) atanır. Bu durumdaki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) = 7] < [\text{termin}(g) = 10]$$

Toplam gecikme=0' dır.

Aynı makine üzerinde önceden ataması gerçekleşmiş,  $g$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $g$  işinin atandığı yer değiştirilerek  $g$  işi bu yeni işin yanına atanır. Bu durumdaki toplam gecikme hesaplanır.

Bu örnekte böyle bir durum bu aşamada yoktur.  $TG_{M_1}=0$

- İzleyen aşama olarak  $g$  işinin,  $M_2$  makinesinde atanması durumundaki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(g) = 3 < \text{termin}(g) = 10], \text{ Toplam gecikme}=0' \text{ dir.}$$

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $g$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $g$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Öncesinde ataması gerçekleşmiş başka bir iş yoktur.  $TG_{M_2}=0$

$M_1$  ve  $M_2$  makinelerinde toplam gecikme değeri 0' dır. Ancak  $M_1$

makinesinin seçilmemesi için o makinedeki gecikmeden daha az gecikme veren başka bir makine olması gerektiğinden,  $g$  işi ilk makineye ve  $c$  ile  $a$  işlerinden sonraya atanır.

- Sıradaki termini küçük olan  $h$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır. Aynı kalıbı kullanan  $c$ ,  $a$  ve  $g$  işleri sonrasında kalıp

değişikliği gerektiği için hazırlık süresi oluşacaktır. Bu süre önceden bahsedildiği gibi 2 saattir. Buna göre,

$$[\text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \text{hazırlık süresi}(2) + \text{işlem süresi}(h) = 13] > [\text{termin}(h) = 10]$$

Toplam gecikme=3 olur.

Aynı makine üzerinde önceden ataması gerçekleşmiş,  $h$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $h$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.  $h$  işi,  $y$  kodlu kalıbı, öncesinde ataması yapılmış ürünler ( $c$ ,  $a$ ,  $g$ ) ise  $x$  kodlu kalıbı kullanmaktadırlar. Bu durumda  $TG_{M1}=3$  olur.

- İzleyen aşama olarak  $h$  işinin  $M_2$  makinesinde son iş olarak atanması durumundaki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(h) = 4 < \text{termin}(h) = 10], \text{ Toplam gecikme} = 0 \text{ 'dır.}$$

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $h$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $h$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Öncesinde ataması gerçekleşmiş başka bir iş yoktur.  $TG_{M2}=0$

$M_2$  makinesindeki toplam gecikme,  $M_1$  makinesindeki toplam gecikme değerinden az olduğu için  $h$  işinin  $M_2$  makinesine ataması gerçekleştirilir.  $M_2$ ' de çizelgelenen ilk iş  $h$  işi olur.

- Sıradaki termini küçük olan  $f$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.  $M_1$ ' de önceki sıra  $c-a-g$  şeklindedir.

$$\left. \begin{array}{l} \text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \\ \text{hazırlık süresi}(2) + \text{işlem süresi}(f) = 12 \end{array} \right\} > [\text{termin}(f) = 11]$$

Toplam gecikme=1 olur.

Yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $f$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $f$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.  $f$  işi,  $y$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ( $c$ ,  $a$ ,  $g$ ) ise  $x$  kodlu kalıbı kullanmaktadırlar. Bu durumda  $TG_{M1}=1$  olur.

- $f$  işinin  $M_2$  makinesinde son iş olarak ( $h$ ' den sonraya) atanması durumundaki halindeki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(h) + \text{işlem süresi}(f) = 7] < [\text{termin}(f) = 11]$$

Toplam gecikme =0' dır.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $f$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $f$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Son iş olarak atanması durumu ile aynı sıralama olduğu için yine gecikme yoktur.  $TG_{M_2}=0$

$M_2$  makinesindeki toplam gecikme,  $M_1$  makinesindeki toplam gecikme değerinden az olduğu için  $f$  işinin  $M_2$  makinesine ataması gerçekleştirilir.

- Sıradaki termini küçük olan  $b$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \text{hazırlık süresi}(2) + \text{işlem süresi}(b) = 13] > [\text{termin}(b) = 12]$$

Toplam gecikme=1' dır.

Yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $b$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $b$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.  $b$  işi,  $y$  kodlu kalıbı, öncesinde ataması yapılmış ürünler ( $c$ ,  $a$ ,  $g$ ) ise  $x$  kodlu kalıbı kullanmaktadırlar. Bu durumda  $TG_{M_1}=1$  olur.

- $b$  işinin  $M_2$  makinesinde son iş olarak atanması halindeki ( $h-f-b$ ) toplam gecikme hesaplanır.

$$[\text{işlem süresi}(h) + \text{işlem süresi}(f) + \text{işlem süresi}(b) = 11] < [\text{termin}(b) = 12]$$

Toplam gecikme=0' dır.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $b$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $b$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır. Son iş olarak atanması durumu ile aynı sıralama olduğu için yine gecikme yoktur.  $TG_{M_2}=0$

$M_2$  makinesindeki toplam gecikme,  $M_1$  makinesindeki toplam gecikme değerinden az olduğu için  $M_2$  makinesine atama gerçekleştirilir.



- Sıradaki termini küçük olan  $d$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$[işlem süresi(c) + işlem süresi(a) + işlem süresi(g) + işlem süresi(d) = 12] < [termin(d) = 13]$$

Toplam gecikme=0' dır.

Aynı zamanda, yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $d$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $d$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Son iş olarak atanması durumu ile aynı sıralama oluştuğu için yine gecikme yoktur.  $TG_{M1}=0$

- $d$  işinin  $M_2$  makinesinde son iş olarak atanması halindeki toplam gecikme hesaplanır.

$$[işlem.süresi(h) + işlem.süresi(f) + işlem.süresi(b) + hazırlık.süresi(2) + işlem.süresi(d) = 18] > [termin(d) = 13]$$

Toplam gecikme=5' dir.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $b$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $b$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$d$  işi,  $x$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ise  $y$  kodlu kalıbı kullanmaktadırlar.  $TG_{M2}=5$

$M_1$  makinesindeki toplam gecikme,  $M_2$  makinesindeki toplam gecikme değerinden az olduğu için  $d$  işinin  $M_1$  makinesine ataması gerçekleştirilir.

- Sıradaki termini küçük olan  $e$  işi alınır,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$\left. \begin{array}{l} \{işlem.süresi(c) + işlem.süresi(a) + işlem.süresi(g) + işlem.süresi(d) + \\ \{hazırlık.süresi(2) + işlem.süresi(e) = 20 \end{array} \right\} > [termin(d) = 14]$$

Toplam gecikme=6' dır.

$e$  işi atandığı yerden alınır ve yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $e$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $e$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$e$  işi,  $y$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ise  $x$  kodlu kalıbı kullanmaktadırlar.  $TG_{M1}=6$

- $e$  işinin  $M_2$  makinesinde son iş olarak atanması halindeki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(h) + \text{işlem süresi}(f) + \text{işlem süresi}(b) + \text{işlem süresi}(e) = 17] > [\text{termin}(e) = 14]$$

Toplam gecikme=3' dür.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $e$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $e$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Son iş olarak atanması durumu ile aynı sıralama olduğu için yine gecikme aynı sürede gecikme vardır.  $TG_{M_2}=3$

$M_2$  makinesindeki toplam gecikme,  $M_1$  makinesindeki toplam gecikme değerinden az olduğu için  $M_2$  makinesine atama gerçekleştirilir.

- Sıradaki termini küçük olan  $e$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$\left. \begin{array}{l} \text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \text{işlem süresi}(d) + \\ \text{hazırlık süresi}(2) + \text{işlem süresi}(e) = 20 \end{array} \right\} > [\text{termin}(d) = 14]$$

Toplam gecikme=6' dır.

Aynı makine üzerinde önceden ataması gerçekleşmiş,  $e$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $e$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$e$  işi,  $y$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış

ürünler ise  $x$  kodlu kalıbı kullanmaktadırlar.  $\Rightarrow TG_{M_1}=6$

- $e$  işinin,  $M_2$  makinesinde son iş olarak atanması halindeki toplam gecikme hesaplanır.

$$[\text{işlem süresi}(h) + \text{işlem süresi}(f) + \text{işlem süresi}(b) + \text{işlem süresi}(e) = 17] > [\text{termin}(e) = 14]$$

Toplam gecikme=3' dür.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $e$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $e$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

Son iş olarak atanması durumu ile aynı sıralama olduğu için yine gecikme aynı sürede gecikme vardır.  $TG_{M_2}=3$

$M_2$  makinesindeki toplam gecikme,  $M_1$  makinesindeki toplam gecikme değerinden az olduğu için  $M_2$  makinesine atama gerçekleştirilir.

- Sıradaki termini küçük olan  $l$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$\left\{ \begin{array}{l} \text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \text{işlem süresi}(d) + \\ \text{hazırlık süresi}(2) + \text{işlem süresi}(l) = 15 \end{array} \right\} = [\text{termin}(l) = 15]$$

Toplam gecikme=0' dır.

Yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $l$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $l$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$l$  işi,  $z$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ise  $x$  kodlu kalıbı kullanmaktadırlar.  $TG_{M1}=0$

- $l$  işinin,  $M_2$  makinesinde son iş olarak atanması durumundaki toplam gecikme hesaplanır.

$$\left\{ \begin{array}{l} \text{işlem süresi}(h) + \text{işlem süresi}(f) + \text{işlem süresi}(b) + \text{işlem süresi}(e) + \\ \text{hazırlık süresi}(2) + \text{işlem süresi}(l) = 20 \end{array} \right\} > [\text{termin}(l) = 15]$$

$M_2$  makinesinde  $e$  işi 3 gün gecikmiştir,  $l$  işi ile birlikte toplam gecikme=5 olur.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $l$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $l$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$l$  işi,  $z$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ise  $y$  kodlu kalıbı kullanmaktadırlar.  $TG_{M2}=5$

$M_1$  makinesindeki toplam gecikme,  $M_2$  makinesindeki toplam gecikme değerinden az olduğu için  $M_1$  makinesine atama gerçekleştirilir.

- Sıradaki termini küçük olan  $j$  işi,  $M_1$ ' de son iş olarak atanır. Bu durumdaki toplam gecikme hesaplanır.

$$\left\{ \begin{array}{l} \text{işlem süresi}(c) + \text{işlem süresi}(a) + \text{işlem süresi}(g) + \text{işlem süresi}(d) + \\ \text{hazırlık süresi}(2) + \text{işlem süresi}(l) + \text{hazırlık süresi}(2) + \text{işlem süresi}(j) = 19 \end{array} \right\} > [\text{termin}(j) = 16]$$

Toplam gecikme=3' dür.

Yine aynı makine üzerinde önceden ataması gerçekleşmiş,  $j$  işi ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $j$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.

$j$  işi,  $x$  kodlu kalıbı kullanmaktadır.  $x$  kalıbını kullanan öncesinde ataması yapılmış işlerden kendisine en yakın olanın yanına atandığı durumda,

$$\left. \begin{array}{l} \text{işlem süresi}(c)+\text{işlem süresi}(a)+\text{işlem süresi}(g)+\text{işlem süresi}(d)+ \\ \text{işlem süresi}(j)+\text{hazırlık süresi}(2)+\text{işlem süresi}(t)=17 \end{array} \right\} > [\text{termin}(i) = 15]$$

Toplam gecikme=2' dir.  $TG_{M1}=2$

- $j$  işi,  $M_2$  makinesinde son iş olarak atanır. Toplam gecikme hesaplanır.

$$\left. \begin{array}{l} \text{işlem süresi}(h) + \text{işlem süresi}(f) + \text{işlem süresi}(b) \\ + \text{işlem süresi}(e) + \text{hazırlık süresi}(2) + \text{işlem süresi}(j) = 21 \end{array} \right\} > [\text{termin}(j) = 16]$$

Toplam gecikme=5' dir.

$M_2$  makinesi üzerinde önceden ataması gerçekleşmiş,  $j$  ile aynı kalıbı kullanan iş var mı kontrol edilir. -Varsa-  $j$  işi, bu işin yanına atanacaktır. Bu durumdaki toplam gecikme hesaplanır.  $TG_{M2}=5$

$j$  işi,  $x$  kodlu kalıbı kullanmaktadır. Öncesinde ataması yapılmış ürünler ise  $y$  kodlu kalıbı kullanmaktadırlar.

$M_1$  makinesinde,  $j$  işinin  $d$  işinden sonra sıralanması durumundaki toplam gecikme, diğer durumlardaki toplam gecikme değerlerinden küçük olduğu için,  $M_1$  makinesine atama gerçekleştirilir.

Uygun makinelerin uygun iş indislerine atamalarının yapılması ardından, -varsa- en az gecikmeyi veren çizelgeleme işlemi tamamlanmış olur. Çizelge 3.4.' de işlerin sıralanışı görülmektedir. Bu durumdaki toplam gecikme değerleri:

$ToplamGecikme_{M1} = 2$  saat ve  $ToplamGecikme_{M2} = 3$  saat şeklinde olur.

Çizelge 3.4. Örnek çizelgeleme probleminin çözümü (10 iş ve 2 makine)

Tamanlanma zamanı		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Makineler	$M_1$	$c$		$a$		$g$			$d$					$j$			$t$	
	$M_2$		$h$			$f$			$b$						$e$			

Yukarıda 10 iş ve 2 makineli durum örneklenmiştir. Fakat gerçek hayatta bu problemle, binlerce iş ve onlarca makine olduğu durumda karşılaşılabılır. Bu durumda da kullanıcı için kolaylık sağlayan, arka planda gerekli ardıştırmaları gerçekleştiren, arayüzlerle sadece problem için gerekli bilgilerin girilmesini sağlayan bir yaklaşım ihtiyacı belirmektedir. Bu sebeple, geliştirilen sezgisel algoritma, etkili bir programlama dili olan C# kullanılarak yazılmıştır. İlgili kaynak kodlarının bir kısmı örnek olarak EK-4.’ de verilmiştir.

### 3.2.2. Geliştirilen algoritmaya dayalı çizelgeleme sistemi - görsel

Geliştirilen algoritma, C# programlama dili kullanılarak kodlanmıştır. Kullanıcı etkileşimli arayüzler, sistemin kullanımını kolaylaştırmakta, anlaşılabilirliği arttırmaktadır. Programın açılması ile aktif hale gelen arayüz Şekil 3.2.’ de verilmiştir.



Şekil 3.2. Program arayüzü

Tasarlanan arayüzde, menü şeklinde oluşturulan “Bilgi Okuma” sekmesi yer almaktadır. Bu sekmeye tıklandığında, çizelgeleme işlemi yapılacak ürün bilgilerinin yer aldığı üretim programı dosyasına erişim sağlanır. İlgili dosyanın seçilmesinin ardından, herhangi bir yanlış veri yoksa (okunacak her bir veri doğru sütunlarda yer alıyorsa), “Ürün bilgileri okundu!” mesajı ekrana yansır. Sonrasında “Kullanılacak Makine Sayısı” ve takvimden de çizelgelemeye başlanılacak tarih seçilir. “Veri Tablosunu Çağır” düğmesi ile Şekil 3.3.’ de görüldüğü gibi üretim programından

okunan ürün adı, kalıp gözü sayısı, birim üretim süresi (sn), üretim adedi (adet) ve termin bilgileri ekrana yansımaktadır. Ekrana yansıyan toplam üretim süresi bilgisi ise, önceki bölümlerde bahsedilen işlem süresine karşılık gelmektedir. İlgili işin birim üretim süresi ile üretim adedinin çarpımından oluşur.

İzleyen tablolar, örnek olarak, 2008 yılı Ocak ayı üretim verilerini esas alarak 676 iş ve 4 makineli sistemde, mevcut ve önerilen çözüm yaklaşımları ile elde edilen çizelgeleri ve elde edilen gecikme sürelerini göstermektedir. Burada her bir işin, ilgilenilen makinelerdeki birim üretim süreleri 37 sn ve kimi ürünlerin kalıplarının 2 gözlü olup, örnek olarak “Makine 2” ye ilişkin çözüm sonuçlarına yer verilmiştir.

Urunler	Kalıp Gözü	Birim Üretim Süresi	Üretim Adedi	Termin	Toplam Üretim Süresi
FF_S54121	2	37	1000	03.01.2008	18500
FF_S54120	2	37	699	03.01.2008	12931
FF_S54120	2	37	106	03.01.2008	1961
FF_S54120	2	37	110	03.01.2008	2035
FF_S54120	2	37	212	04.01.2008	3922
FF_S54120	2	37	712	04.01.2008	13172
FF_S54120	2	37	997	05.01.2008	18444
FF_S54120	2	37	140	05.01.2008	2590
FF_S54120	2	37	230	07.01.2008	4255
FF_S54120	2	37	108	05.01.2008	1998
FF_S54120	2	37	998	07.01.2008	18463
FF_S54120	2	37	532	08.01.2008	9842

Şekil 3.3. Veri tablosu ekran görüntüsü

“ÇİZELGELE (Mevcut Yaklaşım)” ile mevcut yaklaşım ile çizelgeleme sonuçları ekrana yansımaktadır. Şekil 3.4.’ de görüldüğü gibi çıkan sonuç sayfasında, ilgili ürüne ait termin, üretim adedi, ilgili ürünün üretiminin başlama ve bitiş tarihleri ile varsa gecikme süresi bilgisi yer almaktadır. Başlangıçta belirtilen makine sayısı kadar her bir makine için bu sonuç sayfası görüntülenebilmektedir. Tahmini Gecikme Süresi sütununda yer alan “-----“ ile başlayan süre değerleri, ilgili işin erken tamamlandığını göstermektedir. Oluşan toplam gecikme süresi, ekranın üst kısmında yer alan vrd:sa:dk:sn cinsinden okunmaktadır. Sırasıyla işin kaç vardiya, kaç saat, kaç dakika ve kaç sn geciktiğini ifade eder.

Makineler: Makine 2 | Toplam Gecikme Süresi: 86vr:05sa:51dk:13sn

Ürün Adı	Termin Tarihi	Üretim Adedi	Tahmini Baslangic Tarihi	Tahmini Bitiş Tarihi	Tahmini Gecikme Süresi
FRZ_D54231	24.01.2008	100	23.01.2008 06:37	23.01.2008 07:12	---00vr:00sa:47dk:38sn
FF_D70470NE	24.01.2008	100	23.01.2008 07:50	23.01.2008 08:47	00vr:00sa:47dk:02sn
FF_D60330	24.01.2008	100	23.01.2008 09:25	23.01.2008 10:21	00vr:02sa:21dk:42sn
FRZ_K70475NE	24.01.2008	102	23.01.2008 10:59	23.01.2008 11:57	00vr:03sa:57dk:30sn
FRZ_D54231	24.01.2008	230	23.01.2008 12:35	23.01.2008 13:56	00vr:05sa:56dk:00sn
FRZ_K70475NE	24.01.2008	198	23.01.2008 14:34	23.01.2008 16:26	01vr:00sa:26dk:12sn
FF_D70450	24.01.2008	200	23.01.2008 17:04	23.01.2008 18:57	01vr:02sa:57dk:32sn
FF_D74575N	24.01.2008	300	23.01.2008 19:35	23.01.2008 22:25	01vr:06sa:25dk:32sn
FF_L54135	24.01.2008	354	23.01.2008 23:03	24.01.2008 02:24	02vr:02sa:24dk:08sn
FF_F60240	24.01.2008	578	24.01.2008 03:02	24.01.2008 08:29	03vr:00sa:29dk:40sn
FRZ_D70450	25.01.2008	12	24.01.2008 09:07	24.01.2008 09:14	00vr:01sa:14dk:28sn
FF_F54102	25.01.2008	43	24.01.2008 09:52	24.01.2008 10:04	00vr:02sa:04dk:39sn
FF_D70450	25.01.2008	24	24.01.2008 10:42	24.01.2008 10:56	00vr:02sa:56dk:15sn
FF_L60325	25.01.2008	50	24.01.2008 11:34	24.01.2008 12:02	00vr:04sa:02dk:35sn

Şekil 3.4. Mevcut yaklaşım ile çizelgeleme sonuçları ekran görüntüsü

“ÇİZELGELE (Önerilen Yaklaşım)” ile ise önerilen yaklaşım ile çizelgeleme sonuçları ekrana yansımaktadır. Şekil 3.4.’ de görüldüğü gibi mevcut yaklaşım ile “Makine 2” deki toplam gecikme süresi yaklaşık 86 vardiyadır. Şekil 3.5.’ de ise önerilen yaklaşım kullanılarak ulaşılan toplam gecikme süresinin, yaklaşık 40 vardiya olduğu görülebilmektedir. Bu süre değerlerinin kıyaslaması ilerleyen bölümlerde yapılmıştır.

Makineler: Makine 2 | Toplam Gecikme Süresi: 40vr:03sa:43dk:17sn

Ürün Adı	Termin Tarihi	Üretim Adedi	Tahmini Baslangic Tarihi	Tahmini Bitiş Tarihi	Tahmini Gecikme Süresi
FF_D74575N	24.01.2008	178	23.01.2008 02:58	23.01.2008 04:39	---00vr:03sa:20dk:55sn
FF_D70470NE	24.01.2008	198	23.01.2008 05:17	23.01.2008 07:09	---00vr:00sa:50dk:43sn
FRZ_K70475NE	24.01.2008	198	23.01.2008 07:47	23.01.2008 09:39	00vr:01sa:39dk:29sn
FRZ_D70450	24.01.2008	200	23.01.2008 10:17	23.01.2008 12:10	00vr:04sa:10dk:49sn
FF_D74575N	24.01.2008	300	23.01.2008 12:48	23.01.2008 15:38	00vr:07sa:38dk:49sn
FF_L54135	24.01.2008	354	23.01.2008 16:16	23.01.2008 19:37	01vr:03sa:37dk:25sn
FF_F60240	24.01.2008	578	23.01.2008 20:15	24.01.2008 01:42	02vr:01sa:42dk:57sn
FRZ_D70470NE	25.01.2008	126	24.01.2008 02:20	24.01.2008 03:05	---00vr:04sa:54dk:57sn
FRZ_D70550	25.01.2008	88	24.01.2008 03:43	24.01.2008 04:32	---00vr:03sa:27dk:05sn
FF_D70550	25.01.2008	88	24.01.2008 05:10	24.01.2008 06:00	---00vr:01sa:59dk:13sn
FF_D70550	25.01.2008	109	24.01.2008 06:00	24.01.2008 07:02	---00vr:00sa:57dk:27sn
FF_D70470NE	25.01.2008	100	24.01.2008 07:40	24.01.2008 08:37	00vr:00sa:37dk:13sn
FF_D70470NE	25.01.2008	126	24.01.2008 08:37	24.01.2008 09:48	00vr:01sa:48dk:37sn
FF_D54231	25.01.2008	100	24.01.2008 10:26	24.01.2008 11:23	00vr:03sa:23dk:17sn

Şekil 3.5. Önerilen yaklaşım ile çizelgeleme sonuçları ekran görüntüsü



Önerilen yaklaşım ile toplam gecikme süresi gibi ortalama akış süresi de kısalmaktadır. Şekil 3.6.’ da, mevcut çözüm yaklaşımı ile “24.01.2008” tarihinde başlayan üç işin, 9. saate kadar bittiği görülmektedir.

FRZ_K70475NE	24.01.2008	102	23.01.2008 10:59	23.01.2008 11:57	00vrd:03sa:57dk:30sn
FRZ_D54231	24.01.2008	230	23.01.2008 12:35	23.01.2008 13:56	00vrd:05sa:56dk:00sn
FRZ_K70475NE	24.01.2008	198	23.01.2008 14:34	23.01.2008 16:26	01vrd:00sa:26dk:12sn
FF_D70450	24.01.2008	200	23.01.2008 17:04	23.01.2008 18:57	01vrd:02sa:57dk:32sn
FF_D74575N	24.01.2008	300	23.01.2008 19:35	23.01.2008 22:25	01vrd:06sa:25dk:32sn
FF_L54135	24.01.2008	354	23.01.2008 23:03	24.01.2008 02:24	02vrd:02sa:24dk:08sn
FF_F60240	24.01.2008	578	24.01.2008 03:02	24.01.2008 08:29	03vrd:00sa:29dk:40sn
FRZ_D70450	25.01.2008	12	24.01.2008 09:07	24.01.2008 09:14	00vrd:01sa:14dk:28sn
FF_F54102	25.01.2008	43	24.01.2008 09:52	24.01.2008 10:04	00vrd:02sa:04dk:39sn
FF_D70450	25.01.2008	24	24.01.2008 10:43	24.01.2008 10:55	00vrd:01sa:11dk:12sn

Şekil 3.6. Şekil 3.4.’ ün bir kısmı (24.01.2008 tarihi 9. saate kadar yapılan iş bilgileri)

Şekil 3.7.’ de ise “24.01.2008” tarihinde, önerilen çözüm yaklaşımı ile 9. saate kadar altı işin bittiği görülmektedir.

FF_D74575N	24.01.2008	300	23.01.2008 12:48	23.01.2008 15:38	00vrd:07sa:38dk:49sn
FF_L54135	24.01.2008	354	23.01.2008 16:16	23.01.2008 19:37	01vrd:03sa:37dk:25sn
FF_F60240	24.01.2008	578	23.01.2008 20:15	24.01.2008 01:42	02vrd:01sa:42dk:57sn
FRZ_D70470NE	25.01.2008	126	24.01.2008 02:20	24.01.2008 03:05	---00vrd:04sa:54dk:57sn
FRZ_D70550	25.01.2008	88	24.01.2008 03:43	24.01.2008 04:32	---00vrd:03sa:27dk:05sn
FF_D70550	25.01.2008	88	24.01.2008 05:10	24.01.2008 06:00	---00vrd:01sa:59dk:13sn
FF_D70550	25.01.2008	109	24.01.2008 06:00	24.01.2008 07:02	---00vrd:00sa:57dk:27sn
FF_D70470NE	25.01.2008	100	24.01.2008 07:40	24.01.2008 08:37	00vrd:00sa:37dk:13sn
FF_D70470NE	25.01.2008	126	24.01.2008 08:37	24.01.2008 09:48	00vrd:01sa:48dk:37sn

Şekil 3.7. Şekil 3.5.’ in bir kısmı (24.01.2008 tarihi 9. saate kadar yapılan iş bilgileri)

Geliştirilen sezgisel algoritmanın performansı, matematiksel model çözümü ile karşılaştırılmıştır. Matematiksel modelin çözüm süresi, problem boyutunun büyümesi ile üstel olarak arttığından, karşılaştırma, küçük boyuttaki örneklerle gerçekleştirilebilmiştir. Burada, işlem, hazırlık ve toplam gecikme süreleri birimi saat cinsinden verilmiştir. Bu boyuttaki farklı iş ve makine sayıları ile oluşturulan bazı test örnekleri için, geliştirilen sezgisel algoritma ile elde edilen amaç fonksiyonu değerinin, matematiksel modelin verdiği en iyi çözüm değeri ile aynı olduğu görülmüştür. İlgili değerler Çizelge 3.5.’ de verilmiştir.

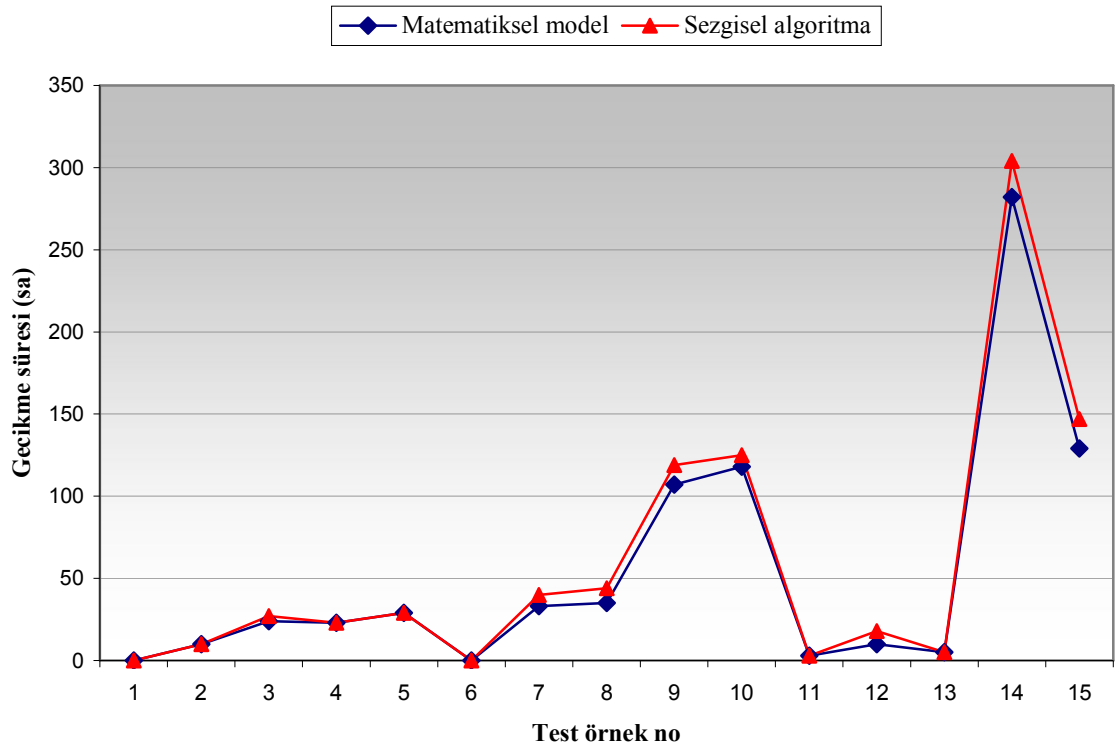


Çizelge 3.5. Test problemleri, parametreleri, sezgisel algoritma ve matematiksel model sonuç karşılaştırma

Test örnek no	Problem Parametreleri						Performans Göstergeleri			
	Makine sayısı	İş sayısı	İşlem süresi	Hazırlık süresi	Kalıp gözü sayısı	Termin	Toplam gecikme süresi (sa)		Çözüm süresi (sa:dk:sn)	
							Matematiksel model	Sezgisel algoritma	Matematiksel model	Sezgisel algoritma
1	2	5	2-6	0-2	1-2	9-14	0	0	00:00:01	00:00:01
			9-11	0-2	1-2	20-27	10	10	00:00:05	00:00:01
			14-23	0-2	1-2	29-37	24	27	00:00:06	00:00:01
			23-34	0-2	1-2	34-68	23	23	00:00:06	00:00:02
			34-39	0-5	1-2	65-82	29	29	00:00:05	00:00:02
2	2	7	2-6	0-2	1-2	9-14	0	0	00:00:02	00:00:00
			5-17	0-2	1-2	12-24	33	40	00:07:15	00:00:03
			13-32	0-2	1-2	23-56	35	44	00:04:35	00:00:04
			21-27	0-2	1-2	40-46	107	119	00:10:53	00:00:04
			30-36	0-2	1-2	60-72	118	125	00:12:43	00:00:04
3	2	8	2-6	0-2	1-2	9-14	3	3	00:20:33	00:00:04
			10-17	0-2	1-2	13-54	10	18	00:14:51	00:00:05
4	2	10	2-6	0-2	1-2	9-16	5	5	41:04.48	00:00:06
			20-75	0-5	1-2	30-100	<b>282</b>	<b>304</b>	<b>~ 7 gün</b>	<b>00:00:06</b>
5	3	10	20-75	0-5	1-2	30-100	<b>129</b>	<b>147</b>	<b>~10 gün</b>	<b>00:00:07</b>

Öte yandan çizelgede, toplam gecikme, çoğu durumda matematiksel model ile daha az bulunmuştur (eniye çözümlere ulaşılmıştır). Fakat sezgisel algoritma, gecikme anlamında eniyi yerine eniyeye yakın değerler verse de çözüm sürelerine bakıldığında matematiksel modelden çok daha kısa sürede sonuç vermekte olup bu açıdan üstündür. Ayrıca, örneğin, 4 ve 5 numaralı test örneklerindeki, sırası ile 282 ve 304 (toplam gecikme süresi) değerleri, iş başına ortalama gecikme olarak düşünüldüğünde 28,2 ve 30,4 saate karşılık gelecektir. Karşılık olarak 7 veya 10 gün yerine 6 veya 7 saniyede çözüme ulaşmanın iyi bir ödünleşme ve tercih sebebi olduğu düşünülmektedir.

Şekil 3.8.' de ise Çizelge 3.5.' de yer alan her bir test örneği için sezgisel algoritma ile elde edilen gecikme süresinin eniyi çözüme yakınlığı görülebilmektedir. Bu karşılaştırma, büyük boyutlu problemlerin çözüm süresinin uzun olması sebebiyle küçük boyutlu problemlerde yapılabilmiştir. Çizelge 3.5.' de görüldüğü gibi, 10 iş-2 makine ve 10 iş-3 makine örnekleri için çözüm süresi bir haftayı geçmiştir. Eniyi çözümü her problemde veremeyen sezgisel algoritma ise çözümü ortalama üç saniyede vermektedir. Bu durum büyük boyutlu problemler için önemli bir üstünlük sağlamakla birlikte, küçük boyutlu örneklerde eniyeye yakın çözümler elde ediliyor olması, çalışmada yer alan problem özelliklerinde, geliştirilen sezgisel algoritmanın performansının rekabet edilebilirliğini göstermektedir.



Şekil 3.8. Test problemlerinin matematiksel model ve sezgisel algoritma kullanılarak elde edilen gecikme süreleri (sa)

## 4. UYGULAMA

Çalışmanın konusu, beyaz eşya sektöründe faaliyet gösteren bir işletmedeki bir süreçte kullanılan makinelerde işlerin en etkin şekilde çizelgelenmesi ihtiyacından yola çıkılarak ortaya konmuştur.

İzleyen bölümlerde, ilgili işletme tanıtılmakta, mevcut süreç incelenmektedir. Ardından, ele alınan özdeş paralel makine problemi tanımına ve geliştirilen yaklaşım kullanılarak çözümüne yer verilmiştir.

### 4.1. İşletmenin Tanıtımı

Buzdolabı üretmek amacıyla 1973 yılında, Eskişehir Organize Sanayi Bölgesi'nde kurulmuş olan işletmede, 1975 yılında General Electric lisansı ile tek kapılı buzdolabı üretimine başlanmıştır. 1987 yılında 459.585 adet üretimle kurulu kapasite aşılmış ve bu kapasitesini ilk safhada 700.000 adet/yıl' a, ikinci safhada ise 900.000 adet/yıl' a çıkarma kararı alınmıştır.

Üretim için gereken plastik levha ve aksesuarlar, işletmenin bulunduğu mevcut yerleşke içerisinde yer alan plastik fabrikasında yapılmaktadır. Burada bulunan beş adet extruder<sup>3</sup> makinesinden büyük kapasiteli olan üçünde buzdolabının iç kısmını oluşturacak olan iç gövde levhası üretilir. Diğer ikisinde ise kapı içi levhası üretilmektedir. Bu levhalar ana fabrikada termofom makineleriyle iç gövde ve kapı iç plastiği haline getirilir.

Plastik fabrikasında aynı zamanda, buzdolabı için gereken plastik aksesuarlar da üretilmektedir. Bunlara örnek olarak; raflar, yumurtalıklar ve buzluklar verilebilir. Bu aksesuarları üretmek için fabrikada 42 adet enjeksiyon pres makinesi bulunmaktadır. Bu makineler 300 ile 1000 ton arası değişebilen tonajlardadır. Plastik fabrikasında aynı zamanda 8 tane serigrafi makinesi bulunmakta olup, 2 tanesi sıcak baskı yapmaktadır. Öte yandan plastik fabrikası, ana fabrikanın tüm plastik ihtiyacını karşılayamamaktadır (ihtiyacın ortalama %70' i karşılanmaktadır). Burada üretilen sözkonusu aksesuarlar,

---

<sup>3</sup> Extrüder makineleri, plastik toz granülleri belirli sıcaklıklar altında eritip, bu bileşimden istenilen formda plastikler elde etmeyi sağlar.

lojistik açıdan fabrikaya sorun olabilecek ve özel yatırım gerektirecek plastikler olup üretim süreci de zor olan parçalardır.

Buzdolabı fabrikasında; beşi ana binada, ikisi de ek binada olmak üzere toplam yedi üretim hattı bulunmaktadır. Her bir üretim hattında farklı özelliklerde buzdolabı üretimi gerçekleştirilmektedir. Çizelge 4.1.' de üretim hatlarında üretilen ürün bilgileri verilmiştir.

Çizelge 4.1. Üretim hatları ve üretilen ürün özellikleri

Üretim Hattı	Ürün Özelliği
1	• Tezgah seviyesi dolaplar (54-60 cm eninde ve en çok 140 cm yüksekliğinde)
2	• NoFrost, Çift kapılı konvansiyonel, Kombi tipi dolaplar (60-70 cm eninde)
3	• Çift kapılı konvansiyonel, Derin çift kapılı (110 cm), En küçük çift kapılı, Tek kapılı konvansiyonel, Frezer dolaplar (54-60-70 cm eninde)
4	• Çift kapılı konvansiyonel dolaplar (54-60-70 cm eninde)
5	• Kombi dolaplar (54-60-70 cm eninde)
6	• SidebySide (4 kapılı), Gardırop tipi dolaplar (84 cm'den daha geniş ende) Çevrim süresi uzun olan dolaplar
7	• Çok fazla işçilik gerektiren karmaşık özellikteki dolaplar

Bilgisayar bütünleşik imalat ve esnek üretim sistemi prensiplerinin uygulandığı işletmede, birçok noktada otomasyon adacıkları oluşturulmuştur. Üretim bantlarında süreçte bir hata oluşması durumunda istasyon durdurularak hata giderilmektedir. Kalite ile ilgili hedefler doğrultusunda bant üzerinde, tamamen otomatik çalışan test istasyonları yer almaktadır. Ayrıca gerek ürün gerekse üretim hakkındaki tüm bilgiler, sistem içinde yer alan elektronik, endüstriyel otomasyon cihazı (PLC - Programmable Logic Controller) ve bir bilgisayar aracılığı ile ortak veritabanında toplanabilmektedir. Bu süreçte, ürün etiket bilgileri referans olmaktadır. Bu sayede hat üzerindeki üretim

izlenebilmekte, talep deęişikleri üretime doęru zamanda yansıtılabilmekte, oluşan aksaklıklara anında müdahale edilebilmektedir.

İşletmede iki ayrı otomatik depolama ve boşaltma sistemi (AS/RS - Automated Storage and Retrieval System) bulunmaktadır. Biri plastik fabrikasının üretimleri ve yardımcı sanayilerden alınan malzemelerin stoklandığı yaklaşık 9000 göz kapasiteli parça AS/RS' dir. Bir vardiyada 1900 kutu yükleme ve boşaltma kapasitesi bulunmaktadır. Dięeri ise yaklaşık 900 kutu kapasiteli boyalı parça AS/RS' dir. Sistem vardiyada 700 kutu yükleme ve boşaltma kapasitesine sahiptir. Malzeme depolama ve dağıtım sistemleri sayesinde, kullanılacak malzeme bant kenarına otomatik beslenebilmektedir.

Buzdolapları, üretim bantlarından ambalaj bantlarına kadar paletler ile taşınmakta, burada paletlerden alınarak paketlenmekte ve boş paletler taşıma sistemi yardımıyla yeniden üretim bantlarına gönderilmektedir. Her bir üretim hattında palet genişlikleri farklıdır, bu nedenle hatlarda, farklı özelliklerde ürünler, aynı genişliğe sahip bazı benzer ürünler ise kimi bantlar arasında ortaklaşa üretilmektedir.

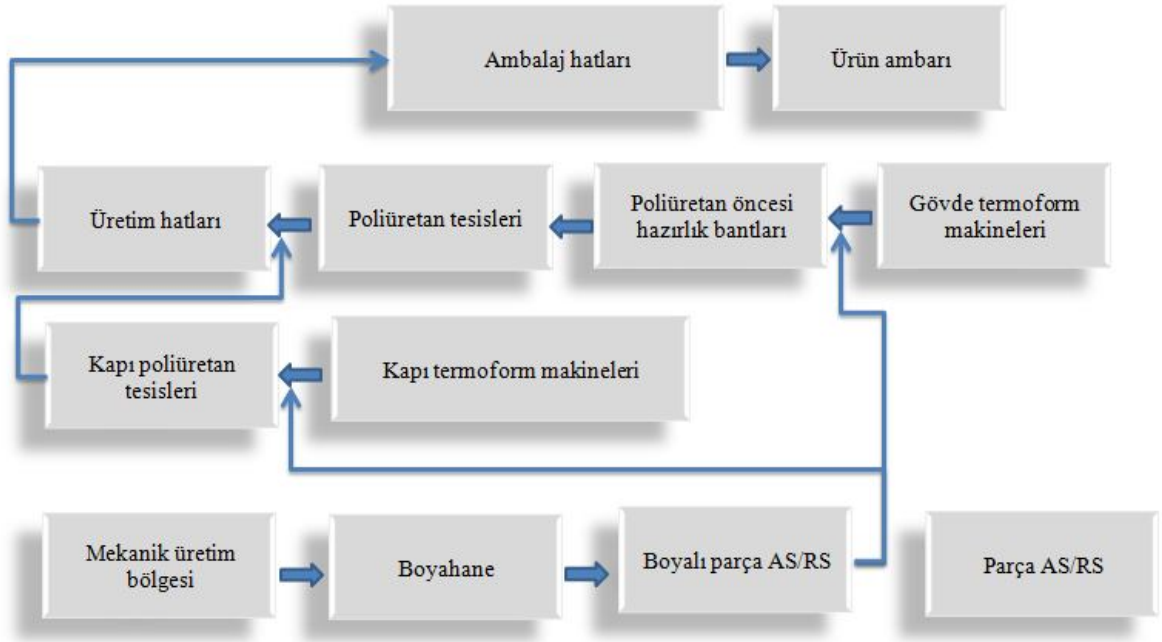
Fabrikada üretimi gerçekleştirilen toplam 116 ana model yer almaktadır. Ana modellerin kendi içerisinde gösterdiği deęişiklik ise farklı ürünler yaratmaktadır. Örneğin aynı modelin kapı plastiğinde farklılıklar olabilir. Bir modelde kapı plastięi 3 raflı iken (bir ürün), bir dięerinde 2 raflı (bir dięer ürün) olabilmektedir. Deęişiklik yaratan bir başka unsur ise kapı saplarının farklılığıdır. Bu sayede fabrikadaki ürün çeşitliği 1000 sayısını aşmaktadır. Bu durum da, ana modellerin farklı ürünlerinde, ürün dönüşlerinde hazırlık süreleri oluşabilmektedir.

#### **4.2. İşletmedeki Üretim Süreci ve Ele Alınan Problem**

İşletmede, Merkez Üretim Yöneticilięi' nin Satış ve Pazarlama Bölümü' nden alınan talepler doęrultusunda, istenen modellerde buzdolabı üretimi, istenen teslimat tarihlerinde sevkiyata hazır olacak şekilde yapılmaktadır.

Üretim Planlama Bölümü, Ürün Yönetimi ve Satış-Pazarlama Bölümü' nden aldığı talep bilgileri doęrultusunda, her bir üretim hattının kapasitesini dikkate alarak üretim programını hazırlamaktadır. Satış iptalleri ya da üretim adedindeki artış talepleri,

doğrudan programı etkilemekte, bu nedenle ilgili ayın öncesinde oluşturulan Taslak Üretim Programı, ay içerisinde 2-3 kez değişebilmektedir.



Şekil 4.1. Fabrika içi süreçler arası ilişkinin gösterimi

Şekil 4.1.' de fabrika içerisindeki süreçlerin birbiri ile olan ilişkisi ve akışı gösterilmektedir. Üretim hatları, 5 ayrı üretim hattını simgeler. Her hata ait gövde termoform makinesi, poliüretan öncesi hazırlık bandı ve poliüretan tesisi vardır. Her bir üretim hattında (ÜH) üretim süreci, gövde termoform (GTF) makinesi ile başlar ve her bir ürün için ayrı GTF kalıpları bulunmaktadır. Üretim programına göre, üretimi yaklaşan ürünün kalıbı GTF makinesine bağlanır. Plastik fabrikasından gelen plastik levhalar (buzdolabı gövdesi için); bu makineye yüklenerek, belirli sıcaklık altında kalıp şeklini alıp, buzdolabı gövdesi formuna ulaşır. Diğer taraftan, her bir üretim hattına hizmet eden kapı termoform (KTF) makineleri bulunmaktadır. GTF makinelerinde olduğu gibi, KTF makineleri için de her bir ürüne ait farklı kalıplar bulunmaktadır. Plastik fabrikasından gelen plastik levhalar (kapı iç plastiği için); KTF makinelerine yüklenerek, belirli bir sıcaklık altında takılan kalıp şeklini alıp, kapı iç plastiği formunu almaktadırlar.

Buzdolabının panel ve kapı sacları ise, işletmenin mekanik üretim bölgesinde yer alan mekanik hatlarda ilgili proseslerle hazırlanmaktadır.

GTF makineleri ile şekli verilen iç gövde plastikleri, poliüretan (PÜ) öncesi hazırlık bandı üzerine alınarak, gerekli soğutma elemanlarının montajı, mekanik takımından gelen sac panellerin ve buzdolabı arka duvarının da takılmasının ardından bir sonraki proses olan PÜ tesisine iletilir. Burada içerisine poliüretan enjekte edilen dolap, Şekil 4.1.' de görüldüğü gibi ilerlediği üretim hattından son montaj ve testleri yapılarak ayrılır ve ambalaj hatlarına iletilir.

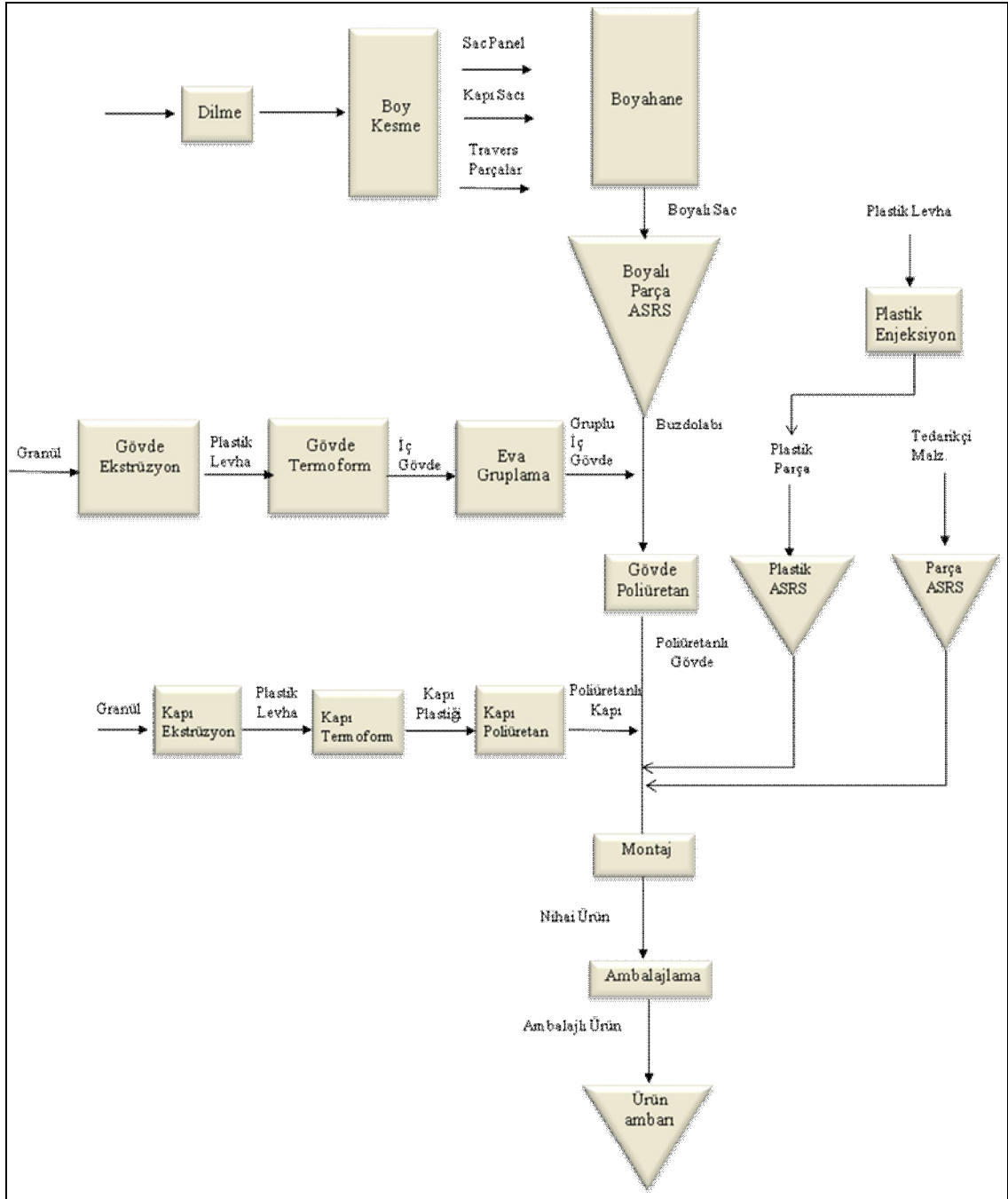
Kapı iç plastikleri ise, KTF makinelerinde ilgili modelin kapısının formunu aldıktan sonra kenar kesimleri yapılarak stok arabalarına yüklenip bir sonraki proses olan kapı poliüretan (KPÜ) tesisine gönderilmektedir. Burada, mekanik takımından gelen kapı sacı ile kapı iç plastikleri birleştirilip; içerisine poliüretan enjekte edilip, üretim bandına yollanmaktadır. Son olarak, montaj işlemleri biten buzdolabına kapısı da takılarak son ürün ambalaj bandına gönderilir.

Şekil 4.2.' de ise fabrika içi süreçlerin akış şeması görülmektedir. Tedarikçi firmalardan alınan sac bobinler, mekanik bölümündeki dilme ve boy kesme işlemleri ardından işlenebilir hali aldıktan sonra; panel hatları, kapı sacı hatları ve preslerdeki işlemler ardından sac panel, kapı sacı ve travers<sup>4</sup> parçalara dönüştürülmektedir. Bu levha, sac ve parçalar, boyanmak üzere boyahaneye gönderilmekte ve boyama süreci ardından boyalı saclar boyalı parça AS/RS' ye yüklenmektedir.

Diğer taraftan plastik fabrikasındaki ekstrüder makineleri ile granül malzemelerden plastik levhalar (gövde ve kapı levhaları) elde edilmektedir. Bu plastik levhalar, Şekil 4.1.' de hatlar ayrıntısında yerleşimi gösterilen GTF makinelerinde, buzdolabı iç gövdesi formunu almaktadır. Sonrasında ise yukarıda bahsedildiği gibi buzdolabı iç gövdelerine, soğutma ekipmanlarının montajı vb. işlemler yapılmaktadır. Bu işlemler ardından, hazırlanan gruplu iç gövde, boyalı parça AS/RS' den gelen sac paneller ile birleştirilmek üzere gövde poliüretan tesisine iletilmektedir.

---

<sup>4</sup> Travers parçalar, bükülmüş sac parçalardır. Buzdolabının kapı ve panel sacları arasındaki bağlantıyı sağlar.



Şekil 4.2. Fabrika içi süreçlerin akış şeması

Yukarıda bahsedildiği gibi; plastik fabrikasında, kapı iç plastiği için kullanılacak olan plastik levhaların üretiminin ardından, bu levhalar Şekil 4.1.' de fabrika içerisinde yerleşimi gösterilen KTF makinelerinde kapı iç plastiği formunu almaktadır. Hazırlanan kapı iç plastikleri bir sonraki işlem olan KPÜ tesisine gönderilmektedir. Bu işlemin ardından hazırlanan poliüretanlı kapı ise; poliüretanlı gövde ile birleştirilerek, Şekil



4.2.' de görüldüğü gibi plastik AS/RS ve parça AS/RS' den gelen malzemeler ile montaj işlemleri yapılmakta ve ardından son ürün (buzdolabı) halini almaktadır.

Ele alınan problem, KTF makinelerine işlerin atanması sürecinde, etkin bir makine çizelgelemeye duyulan ihtiyaçtan doğmuştur. Mevcut sistemde,

- Makinelere iş yüklemeyi kontrol eden bir proses operatörü bulunmakta, bu kişi hangi vardiyada ise, işletme içi ortak bir üretim izleme programından makinelerin ürettiği rakamları kendisi tarafından almaktadır.
- Kapı poliüretan tesislerinden üretim rakamları alınarak, bir Excel dosyasına kaydedilmektedir.
- Stok durumlarına bakarak üretim programında sırası yaklaşan modelin kalıbının hangi makineye bağlanması gerektiğine karar verilmektedir.
- Üretim planlama bölümü, fabrikadaki üretim hatlarının kapasiteleri ve üretebildikleri model türünü esas alarak hazırladığı, ilgili aya ilişkin üretim programını, bir önceki ayın 25' inde yayınlar. Yayınlanan taslak üretim programına bakılarak; gün ve adet bilgileri dikkate alınıp, olabilir makine arıza vs. gibi durumlara karşı, kapı poliüretan tesislerinden 1,5–2 vardiya önde gidecek şekilde üretim yapılmaktadır. İlgili proseste çalışan kişilerin bilgi ve tecrübelerine dayanılarak, önde gidilmesi istenen süreye bağlı kalınmaktadır. Ayrıca üretim hattının çevrim süresi, KTF makinelerindeki çevrim süresinden küçüktür. Örneğin, KTF makinelerindeki ortalama çevrim süresinin 34 sn, bandın ortalama çevrim süresinin ise 24 sn olduğu düşünülürse; üretim bandı kapasitesi 1063 adet/vardiya, bir KTF makinesinin kapasitesi ise 750 adet/vardiya olur (vardiyadaki çalışılabilir süre 425 dakika=25500 sn' dir). KTF süreci ve kapı plastiklerinin banda verilmesi arasında başka proseslerin de olduğu düşünülürse, KTF makineleri ile bandın eş zamanlı üretiminin söz konusu olamayacağı açıkça görülebilmektedir. Planlanan üretim adedinin gerçekleşebilmesi için KTF makinelerinin, üretimini, en az bir vardiya öncesinden tamamlaması gerekmektedir. Hatalı ürün oluşması, makine arızaları gibi durumlar da göz önüne alınacak olursa 1,5 -2 vardiya önden gidilmesi öngörülmektedir.

Kapı termoform bölgesindeki mevcut süreç, karar verici olan teknikerin bilgi birikimine dayanarak işlemektedir. Yaşanan değişiklikler, oluşan sorunlar sezgisel olarak çözülmektedir. Bu durum, bilgi ve tecrübelerinden dolayı ilgili çalışan kişi ya da kişilere bağımlı kalındığını ve bu uzmanların yokluğunda yaşanacak sorunlara müdahalenin gecikmeye sebep olacağını göstermektedir. Kapı termoform sonrasındaki işlem olan kapı poliüretan tesisine yetişebilmek, üretim hattını durdurmamak için gecikmeleri enazlamak hedeflenmektedir.

Bu çalışmada, ana fabrikada süregelen bu sorundan yola çıkılarak önerilen sistemde, süreci sistematik hale getirip, işlere ait –varsa- en az gecikmeyi sağlayan işlem sırasının, çizelgeleme yöntemleri ile belirlenmesi istenmektedir.

İşletmede 9 adet KTF makinesi kullanılmaktadır. Bu makineler özdeş paralel makineler olup her bir kapı iç plastiği, tüm makinelerde aynı sürede işlem görmektedir. Her iç kapı plastiğine ait kalıplar bulunmakta, bu kalıpların makineye bağlanma süresi, her bir kalıp için aynı olup, 38 dakikadır. Makine arızaları, bakım-onarım vb. olduğunda bu makinelerin hepsi aynı anda aktif olamadığından, kullanılabilir makine sayısı her zaman 9 değildir. Bu sebeple, toplam gecikme süresini enküçükleyecek sıralama ve bunu sağlayacak makine sayısı bilgisi önem taşımaktadır.

Buzdolaplarının kimisinin iki, kimisinin tek kapılı olması sebebiyle toplamda kapı iç plastiği sayısı, model sayısının üzerinde olmaktadır. Örneğin, FRZ\_ D54271 ve FF\_ D54271 aslında D54271 kodlu çift kapılı buzdolabının sırası ile üst ve alt kapıları olup, çalışmada, çizelgelenecek iki ayrı ürün olarak alınmıştır.

Aylık üretim programında yayınlanan ürünlerin, kullanılabilir makine sayısına bağlı olarak çizelgelenmesi istenmektedir. Üretimi gerçekleşecek kimi buzdolabı iç kapı plastiklerinin bazıları aynı kalıpla işlenebilmektedir. Bir işten diğer işe geçişteki kayıp zaman, varsa kalıp değişimi sebebiyle olduğundan aynı kalıbı kullanan ürünler arası geçişte hazırlık süresi olmamaktadır. İlgili modellerden sonra, kalıp değişimi gerektirmeyen bir sıralama olduğunda (sıralamada, aynı iç kapı özelliklerinde olan bir diğer model gelirse) kalıp hazırlık süresi “0” olmaktadır. Aksi takdirde kalıp hazırlık süresi 38 dakikadır.

Herhangi bir ayın üretim programında yer alan buzdolabı modelleri ve bu modellerin kapı iç plastiklerinin kodları (aynı kodlar, üretim için kullanılan kalıp

kodlarıdır), kalıp göz sayıları (kalıp iki gözlü ise; aynı anda iki kapı plastiği form alabiliyor demektir) bilgileri EK-3.' de verilmiştir.

### **4.3. Problemin Geliştirilen Yaklaşım Kullanılarak Çözümü**

İşletmedeki kapı termoform bölgesinde, bir kapı plastiğinin şekillendirilmesi birden fazla özdeş makinede yapılabilmektedir. Bu sebeple işlerin çizelgelenmesinde alternatifler söz konusudur. Hangi işin hangi makinede ve hangi sırada yapılacağı belirlenmesi için tamsayılı doğrusal karar modelleri ile en iyi çözüme ulaşılabilmesine karşın, oluşabilecek ani değişikliklerde anlık çözümler elde edilemediği için ve önceki bölümlerde bahsedilen tek bir uzmana bağlı olmadan çizelgeleme işlemine yönelik sezgisel bir algoritma geliştirilmiştir. Oluşturulan bilgisayar programındaki kullanıcı etkileşimli arayüzler ile sistemin kullanımı kolaylaştırılarak, anlaşılabilirliği artırılmıştır.

Bölüm 4.2.' de bahsedildiği gibi her ayın son haftasında, bir sonraki ayın üretim programı yayınlanmaktadır. Bu üretim programında; ürün bilgileri, üretileceği üretim hattı (montaj bandı), ürün kodu, stok numarası, teslim tarihi (ambalajlanmış son halini alması için) ve üretim adedi bilgileri yer almaktadır. Ele alınan problem, ilgili ürünlerin (buzdolabı) kapı plastiklerinin yapıldıkları makineleri çizelgeleme olduğu için, öncelikle ürünlerin kaç kapılı (tek ya da çift) olduğu bilgisi programdan okunmalıdır. Her bir farklı kapı plastiği, ayrı bir ürün olarak değerlendirilir ve üretim programı yanında, ürünlere ilişkin bu bilgilerin yer aldığı ayrı bir özellikler sayfası yer almalıdır. Çalışma kapsamında bu sayfa tasarımı yapılmış ve üretim programlarına entegre edilmiştir. Önceki bölümlerde bahsedildiği gibi bu sayfada yer alan bilgiler, EK-3.' de verilmiştir.

#### **4.3.1. Mevcut ve önerilen yaklaşımların çözüm sonuçlarını karşılaştırma**

Mevcut ve önerilen yaklaşım sonuçlarının karşılaştırıldığı tablo, Çizelge 4.2.' de verilmiştir. 12 ayın üretim programı verileri ve kullanılabilir makine sayıları ile birlikte iyilenmek istenen toplam gecikme süresi performans ölçütü, problem, mevcut ve önerilen yaklaşımlar ile çözdürülerek karşılaştırılmıştır. Tüm denemelerde önerilen yaklaşımın daha iyi sonuç verdiği Çizelge 4.2.'de görülebilmektedir.

Aylık üretim çizelgeleme işlemi için ilgili ayın başından bir gün öncesi çizelgeleme başlangıç tarihi olmaktadır. Program arka planda işlerin asıl terminini hesaplamakta, hesaplanan bu termin bilgisine göre sıralamayı yapmaktadır. Bu sayede üretim programında yer alan termin bilgisi yerine, bu tarihin kaç vardiya öncesinde işler tamamlanmalı ise o tarih asıl termin olmaktadır. 12 ayın üretim programları kullanılarak, bahsedildiği üzere başlangıç tarihleri belirlenmiştir. Çizelgeleme başlangıç tarihi değişikçe performans ölçütünün alacağı değer, tamamlanma zamanlarının değişmesi ile değişecektir.

Çizelge 4.2. Mevcut ve önerilen çözüm yaklaşımlarının karşılaştırılması

Örnek Grup No	Makine sayısı	İş sayısı	Toplam gecikme süresi (sa)		İyileşme oranı (%)
			Mevcut yaklaşım	Önerilen yaklaşım	
1	10	957	17,0	4,6	73
	9	957	28,4	19,7	31
	8	957	59,7	24,3	59
2	10	936	20,9	3,0	86
	9	936	34,5	23,5	32
	8	936	65,4	29,6	55
3	10	914	4	0	100
	9	914	9,75	0,2	98
	8	914	16	4,7	71
4	10	899	20,2	3,0	85
	9	899	34,5	20,8	40
	8	899	65,4	26,4	60
5	10	865	17,3	4,1	76
	9	865	27,7	23,0	17
	8	865	59,4	25,9	56
6	10	846	17,8	1,5	92
	9	846	26,9	20,2	25
	8	846	57	27,4	52
7	10	811	14,7	1,5	90
	9	811	27,0	9,2	66
	8	811	54,7	27,7	49

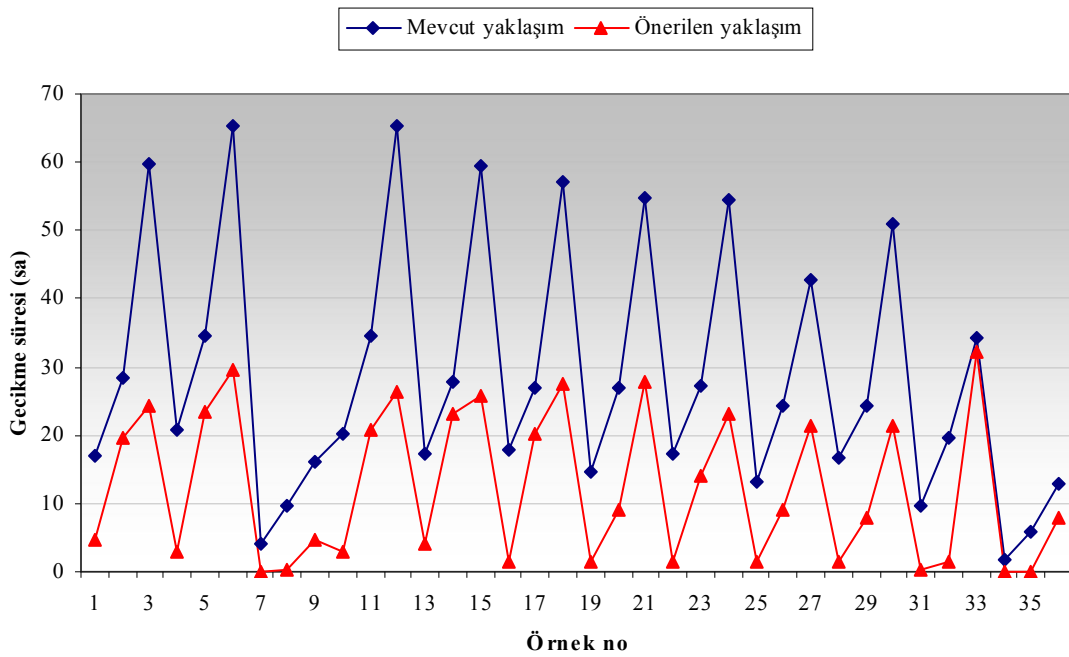
Çizelge 4.2. Mevcut ve önerilen çözüm yaklaşımlarının karşılaştırılması (devam)

Örnek Grup No	Makine sayısı	İş sayısı	Toplam gecikme süresi (sa)		İyileşme oranı (%)
			Mevcut yaklaşım	Önerilen yaklaşım	
8	10	798	17,3	1,5	91
	9	798	27,2	14	49
	8	798	54,5	23,2	57
9	10	783	13,2	1,5	89
	9	783	24,3	9	63
	8	783	42,7	21,4	50
10	10	779	16,6	1,5	91
	9	779	24,4	7,8	68
	8	779	50,9	21,4	58
11	10	760	9,7	0,2	98
	9	760	19,5	1,5	92
	8	760	34,2	32,2	6
12	10	740	1,9	0	100
	9	740	6	0	100
	8	740	13	7,9	39
Gecikme sürelerindeki ortalama iyileşme oranı (%)					66

Mevcut sistemde önceden bahsedildiği gibi 9 adet KTF makinesi bulunmaktadır. Ancak kullanılabilir makine sayısı, makine arıza, bakım onarım vb. nedenlerle değişebilmektedir. Çizelge 4.2.' deki örnek problemlerde ilgili makine ve iş sayıları ile elde edilen gecikme süreleri görülmektedir. Kapasite artışı düşüncesi ile bir makinenin daha alınması durumu da örneklenmiştir. İşletmede KTF makinelerinden sonra, üretim hattı üzerinde başka işlemler de olduğundan, burada oluşan darboğaz, bandı durdurabilmekte; bu durum da üretim programında değişikliğe, yok satmaya ve hatta sonradan karşılama maliyetine katlanmaya sebep olabilecektir. Sistemin gerçek ortalama iş sayısı ve kullanılan makinelerin sayısı dikkate alındığında, geliştirilen sezgisel algoritma ile ortalama olarak % 66 oranında iyileşme sağlandığı görülmektedir. Bu durum, mevcut sistem parametreleri ile şu anki uygulamadan daha etkili bir üretim çizelgelemesi yapılabileceğini göstermektedir.

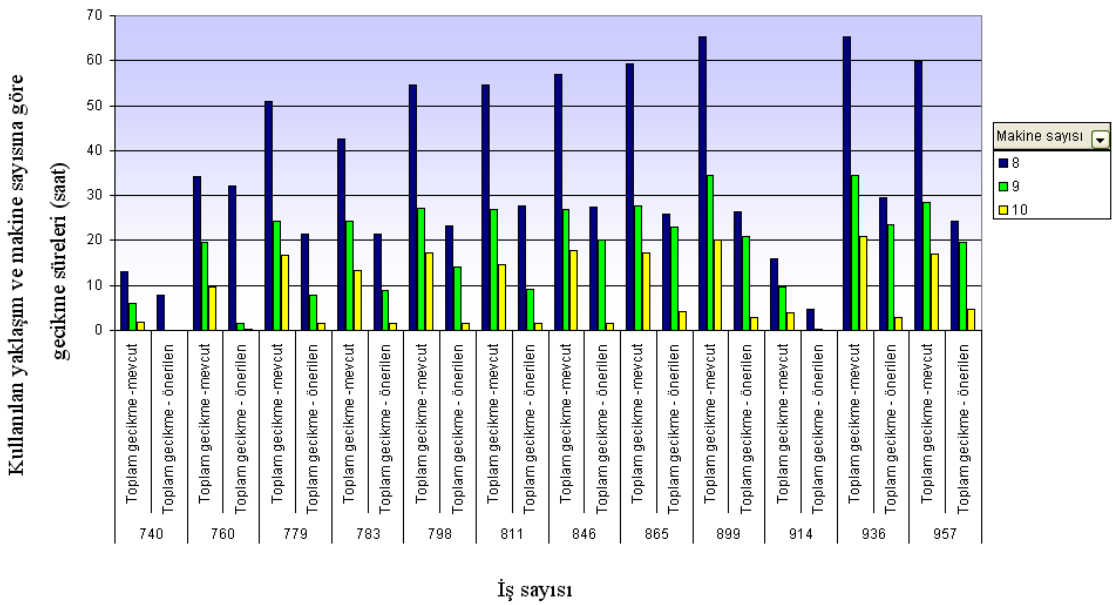
### 4.3.2. Karşılaştırma grafikleri

İşletmede kullanılan mevcut ve önerilen yaklaşımlara karşı gelen gecikme süreleri, geçmiş aylara ait veriler kullanılarak karşılaştırılmıştır. Şekil 4.3.'e göre, önerilen yaklaşım kullanılarak elde edilen gecikme sürelerinin mevcut yaklaşıma göre daha küçük olduğu görülmektedir.



Şekil 4.3. Mevcut ve önerilen yaklaşıma ait gecikme süreleri (sa)

Şekil 4.4.' de, 12 ayın üretim programından hareketle, toplam iş ve kullanılabilir makine sayılarına (8, 9, 10) ilişkin mevcut ve önerilen sistem ile elde edilen gecikme süreleri yer almaktadır. Buradan, önerilen yaklaşım ile elde edilen gecikme sürelerinin, kullanılabilir her makine sayısı için mevcut sistemde oluşan gecikme sürelerinden küçük olduğu görülmektedir. Bu durum, probleme ilişkin performans ölçütü olan toplam gecikme değerinin enküçüklenmesi amacının sağlanıyor olduğunu göstermektedir.



Şekil 4.4. Kullanılan çözüm yaklaşımı ve makine sayısına göre gecikme süreleri dağılımı

## 5. SONUÇ VE ÖNERİLER

Çalışmada, özdeş paralel makine çizelgeleme problemi ele alınmıştır. Literatürde yer alan bu tip problemlerde, erişilebildiği kadarıyla, ürün kalıp gözü sayısı bilgisi ele alınmamıştır. Çalışmada bu bilgiye yer verilerek, işlerin tamamlanma zamanına etkisine dikkat çekilmiş, toplam gecikme süresini enküçüklemeye yönelik sezgisel çözümler bulunmaya çalışılmıştır. Bunun için, önerilen yaklaşımın performansını görmek amacıyla, küçük problemler için, matematiksel model sonuçları (eniye çözüm) ile geliştirilen sezgisel algoritma sonuçları karşılaştırılmıştır. Eniye çözüm çoğu durumda sezgisel algoritma ile de elde edilmiştir. Büyük problemler için ise matematiksel modelin çözülememesi sebebiyle, sezgisel algoritma, gerçek veriler için kullanılmış, bu durumda da mevcut ve önerilen çizelgeler karşılaştırılmıştır.

Problem boyutunun artması ile LINGO 11.0'da matematiksel modelin çözüm süresi artmaktadır. Küçük problemler olması halinde bile makul süreler içerisinde en iyi çözüme ulaşılamamaktadır. Küçük örnekler için test örnekleri türetilmiştir. Ayrıca sezgisel yaklaşım, uygulamanın yapıldığı işletmede kullanılan yöntemle göre gecikmeleri azaltan sonuçlar vermiştir.

Çalışmada önerilen sezgisel çizelgeleme yaklaşımı, talep değişikliklerine duyarlı, imalat sektöründe kullanılan kalıp gözü sayısı bilgisini de dikkate alan dinamik bir yaklaşımdır. Bu yaklaşım, kalıp kullanımına bağlı olarak gereken hazırlık sürelerinin olduğu, kalıp göz sayısı ile üretim adedinin ilişkilendirildiği, gecikmelerin enazlanması hedefli tüm üretim sistemlerinde uygulama alanı bulabilir. Bu yöntem yardımıyla, kabul edilebilir bir sürede, amaca uygun sonuçlar elde edilebilmektedir. Önerilen yaklaşım, anlaşılır ve kolay uygulanabilir olmasının yanı sıra, kullanıcının uygulamada karşılaştığı farklı durumlar karşısında etkili karar alabilmesine yardımcı olacak bilginin üretilmesine de olanak tanımaktadır.

Geliştirilen program, kullanıcının üretim programından üretim adedi ve termin verilerini çekebilmesine, kullanılabilir makine sayısı, kalıp gözü sayısı ve çizelgeleme işlemi başlangıç tarihi gibi problem parametrelerinin değiştirilebilmesine izin veren esnek bir yapıya sahip olduğu için, benzer problemlerle karşılaşan farklı firmalar da bu sistemi kullanabilecektir.



Bundan sonraki alıřmalarda, geliřtirilen sezgisel yaklařımın performans lütü olan toplam gecikme faktörüne diđer performans lütleri de dahil edilebilir ya da farklı performans lütleri için özümler arařtırılabilir. Geliřtirilen sezgisel yaklařım, metasezgisel yöntemlerle karşılařtırılabilir.

## KAYNAKLAR DİZİNİ

Acar, N., 1997, Tam Zamanında Üretim, Ankara: MPM Yayınları, 20s.

Alagöz, O. and Azizoğlu, M., 2003, Rescheduling of identical parallel machines under machine eligibility constraints, *European Journal of Operational Research*, 149, 2.

Alcan, P., 2008, Benzer olmayan paralel makinelerin çizelgelenmesinde bulanık esaslı proses zamanlarının genetik algoritma uygulaması, Yüksek lisans tezi, 135s. (yayımlanmamış)

Alidaee, B. and Rosa, D., 1997, Scheduling parallel machines to minimize total weighted and unweighted tardiness, *Computers Ops Res.*, 24, 1.

Al-Khamis, T. and M'Hallah, R., 2011, A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity, *Computers & Operations Research* 38, 1747–1759.

Allahverdi A., Gupta J.N.D., Aldowaisan T., 1999, A review of scheduling research involving set-up consideration, *OMEGA - International Journal of Management Science*, 27, 219–239.

Armentano, V.A. and Filho, M.F., 2007, Minimizing total tardiness in parallel machine scheduling with setup times: An adaptive memory-based GRASP approach, *European Journal of Operational Research*, 183, 1.

Aydemir, E., 2009, Atölye tipi çizelgeleme problemlerinin öncelik kuralı tabanlı genetik algoritma yaklaşımıyla simülasyon destekli optimizasyonu, Yüksek lisans tezi, Süleyman Demirel Üniversitesi, 123s. (yayımlanmamış)

Azizoğlu, M., 2003, Preemptive scheduling on identical parallel machines subject to deadlines, *European Journal of Operational Research*, 148, 205–210.

Azizoglu, M. and Kirca, O., 1998, Tardiness minimization on parallel machines, *Int. J. Production Economics*, 55, 63–168.

Azizoglu, M. and Kirca, O., 1999, On the minimization of total weighted flow time with identical and uniform parallel machines, *European Journal of Operational Research*, 113, 91-100.

Bakır, M.A. ve Altunkaynak, B., 2003, Tamsayılı Programlama-Teori, Modeller ve Algoritmalar, Nobel Yayın Dağıtım Ltd. Şti., 148s.

Baker, K. R., 1974, *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York.

### KAYNAKLAR DİZİNİ (devam)

Balin, S., 2011, Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation, *Information Sciences* 181 3551–3569.

Baptiste, P. and Timkovsky, V.G., 2001, On preemption redundancy in scheduling unit processing time jobs on two parallel machines, *Operations Research Letters*, 28, 2.

Beraldi, P., Ghiani, G., Grieco, A. and Guerriero, E., 2007, Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequencedependent set-up costs, *Computers & Operations Research*, 234-112.

Bilge, Ü., Kırac, F., Kurtulan, M. ve Pekgün, P., A tabu search algorithm for parallel machine total tardiness problem, *Computers & Operations Research*, 31, 397–414.

Biskup, D., Herrmann, J. and Gupta, J. N. D., 2008, Scheduling identical parallel machines to minimize total tardiness, *Int. J. Production Economics*, 115, 134– 142.

Blum, C., Roli, A., 2003, Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison, *ACM Computing Surveys*, 35, 3, 268-308.

Boray, M. B., 2007, Paralel tezgahlarda arıza halinde çok amaçlı çizelgeleme, Yüksek lisans tezi, Yıldız Teknik Üniversitesi, 145s. (yayınlanmamış)

Breedam, A., V., 2001, “Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem”, *Computers & Operations Research*, 28, 289-315.

Brucker, P., Kravchenko, S. A., 2008, Scheduling jobs with equal processing times and time Windows on identical parallel machines, *Springer Science+Business Media*, 11, 239-257.

Cao, D., Chen, M. and Wan, G., 2005, Parallel machine selection and job scheduling to minimize machine cost and job tardiness, *Computers & Operations Research*, 32, 1995–2012.

Carpov, S., Carlier, J., Nace, D. and Sirdey, R., 2010, Two-stage hybrid flow shop with precedence constraints and parallel machines at second stage, *Computers & Operations Research*, 2010.

Ceppek, O. and Sung, S., 2005, A quadratic time algorithm to maximize the number of just-in-time jobs on identical parallel machines, *Computers & Operations Research*, 32, 3265–3271.

### KAYNAKLAR DİZİNİ (devam)

Chang, P.C., Chen, S.H., Lin, K.L., 2005, Two Phase Sub Population Genetic Algorithm for Paralel Machine- Scheduling Problem, Expert System with Applications, 29, 705-712.

Chung, T., Liao, C., and Su, L., 2010, Scheduling on identical machines with batch arrivals, Int. J. Production Economics, 123, 179–186.

Chauny, F., Haurie, A., Loulou, R., and Wagneur, E.,1987, Sequencing punch operations in an FMS: a three-dimensional space filling curve approach, INFOR, 25 (1), 25-45.

Cowling,, P. and Johansson, M., 2002, Using real time information for effective dynamic scheduling. European Journal of Operational Research, 139, 230-244.

Croce, F., Garaix, T. and Grosso, A., 2010, Iterated local search and very large neighborhoods for the parallel-machines total tardiness problem, Computers & Operations Research.

Dourado, M. C., Rodrigues, R. and Szwarcfiter, J. L. , 2010, Scheduling on parallel machines considering job-machine dependency constraints, Electronic Notes in Discrete Mathematics 36, 431–438.

Dell'Amico M. and Martello, S., 2005, A note on exact algorithms for the identical paralel machine scheduling problem, European Journal of Operational Research, 160, 576–578.

Dunstall, S. and Wirth, A., 2005, A comparison of branch-and-bound algorithms for a family scheduling problem with identical parallel machines, European Journal of Operational Research, 167, 283–296.

Dunstall, S. and Wirth, A., 2005, Heuristic methods for the identical parallel machine flowtime problem with set-up times, Computers & Operations Research, 32, 2479–2491.

Erbasta, E., 2010, Paralel tezgahlarda yükleme ve çizelgeleme problemi için karma tamsayıli modelleme ve genetik algoritma temelli yeni bir çözüm yaklaşımı, Eskişehir Osmangazi Üniversitesi, 84s. (yayınlanmamış)

Eren, H., 2002, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma(GA) ile Çözüm Performansının Artırılmasında Deney Tasarımı Uygulaması, İstanbul Teknik Üniversitesi, Yüksek lisans tezi. (yayınlanmamış)

### KAYNAKLAR DİZİNİ (devam)

Eren, T. ve Güler, E., 2005, İki ölçütlü paralel makineli çizelgeleme problemi: maksimum tamamlanma süresi ve maksimum erken bitirme, V. Ulusal Üretim Araştırmaları Sempozyumu.

Eren, T. ve Güner, E., 2006, Öğrenme etkili hazırlık ve taşıma zamanlı paralel makine çizelgeleme problemi, Havacılık ve Uzay Teknolojileri Dergisi, 2, 4, 67-72.

Eren, T. ve Güner, E., 2006, Paralel makineli çizelgelemede toplam tamamlanma zamanı ve maksimum gecikmenin enküçüklenmesi, Selçuk Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 21, 1-2.

Eren, T. and Güler, E., 2007, Minimizing total tardiness in a scheduling problem with a learning effect, Applied Mathematical Modelling, 31, 1351–1361.

Eren, T., 2008, Farklı geliş zamanlı öğrenme etkili paralel makine çizelgeleme problemi, Havacılık ve Uzay Teknolojileri Dergisi, 3, 4, 37-46.

Eren, T., 2009, A bicriteria parallel machine scheduling with a learning effect of setup and removal times, Applied Mathematical Modelling, 33, 1141–1150.

Eren, T., 2009, A note on minimizing maximum lateness in an m-machine scheduling problem with a learning effect, Applied Mathematics and Computation, 209, 186–190.

Eren, T. and Güner, E., 2009, A bicriteria parallel machine scheduling with a learning effect, Springer-Verlag London Limited.

Filippi, C., 2010, An approximate algorithm for a high-multiplicity parallel machine scheduling problem, Operations Research Letters 38, 312-317.

French, S., 1982, Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop, John Wiley & Sons, New York.

Gacias, B., Artigues, C. and Lopez, P., 2010, Parallel machine scheduling with precedence constraints and setup times, Computers & Operations Research.

Gharbi, A. and Haouari, M., 2007, An approximate decomposition algorithm for scheduling on parallel machines with heads and tails, Computers & Operations Research, 34, 15.

Gharehgozli, A.H., Tavakkoli-Moghaddam R. and Zaerpour, N., 2009, A fuzzy-mixed-integer goal programming model for a parallel-machine scheduling problem with sequence-dependent setup times and release dates, Robotics and Computer-Integrated Manufacturing, 25, 853–859.

**KAYNAKLAR DİZİNİ (devam)**

Graves, S. C., 1981, A Review of Production Scheduling, *Operations Research*, 29(4), 646-675.

Guo, S. and Kang, L., 2010, Online scheduling of malleable parallel jobs with setup times on two identical machines, *European Journal of Operational Research*, 206, 555–561.

Gupta, J. N. D. and Torres A., 2000, Minimizing makespan subject to minimum total flow-time on identical parallel machines, *European Journal of Operational Research*, 125, 370-380.

Gupta J.N.D. and Torres A., 2005, Generating efficient schedules for identical paralel machines involving flow-time and tardy jobs, *European Journal of Operational Research*, 167, 1-4.

Hall, N.G., Potts, C.N. and Sriskandarajah, C., 2000, Parallel machine scheduling with a common server, *Discrete Applied Mathematics*, 102, 1.

Haned, A., Soukhal, A., Boudhar, M. and Tuong, N., 2011, Scheduling on paralel machines with preemption and transportation delays, *Computers & Operations Research* 39, 374–381.

Hiraishi, K., Levner, E. and Vlach, M., 2002, Scheduling of parallel identical machines to maximize the weighted number of just-in-time jobs, *Computers & Operations Research* 29, 841-848.

Huang, X. and Wang, M., 2010, Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties, *Applied Mathematical Modelling*, 35, 349–1353.

Hu, P., 2004, Minimising total tardiness for the worker assignment scheduling problem in identical parallel-machine models, Springer-Verlag London Limited.

Jeng.A.A.K and Lin,B.M.T., 2007, A note on Parallel Machine Scheduling with deteriorating jobs, *Journal of Operational Research Society* 58:824-826.

Jiang, J., Chen, K. and Wee, H., 2007, A dynamic scheduling model of parallel-machine group for weapon production, Springer-Verlag London Limited.

Jouglet, A. and Savourey, D., 2011, Dominance rules for the parallel machine total weighted tardiness scheduling problem with release dates, *Computers & Operations Research* 38, 1259–1266.

Karaboğa, D., 2004, Yapay Zeka Optimizasyon Algoritmaları, Atlas Yayın Dağıtım, Cağaloğlu, İstanbul.

**KAYNAKLAR DİZİNİ (devam)**

Kelerler, H., 2008, An approximation algorithm for identical parallel machine scheduling with resource dependent processing times, *Operations Research Letters*, 36, 157 – 159.

Keskinturk, T., Yildirim M., and Barut, M., 2010, An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times, *Computers & Operations Research*.

Koulamas, C. and Kyparisis, G.J., 2000, Scheduling on uniform parallel machines to minimize maximum lateness, *Operations Research Letters*, 26, 1.

Koulamas, C. and Kyparisis, G.J., 2007, An improved delayed-start LPT algorithm for a partition problem on two identical parallel machines, *European Journal of Operational Research*, 1-2.

Küçük, B., 2010, Özdeş paralel makineli bir üretim sisteminin karınca koloni algoritması ile çizelgelenmesi, Doktora tezi, İstanbul Üniversitesi, 110s. (yayınlanmamış)

Lamothe, J. and Marmier, F., 2010, Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints, *Computers & Operations Research*.

Lann, A. and Mosheiov, G., 2003, A note on the maximum number of on-time jobs on parallel identical machines, *Computers & Operations Research*, 30, 1745–1749.

Leonardi, S. and Raz, D., 2006, Approximating total flow time on parallel machines, *Journal of Computer and System Sciences*, 73, 1-2.

Li, X., Chen, Y. and Sun, Y., 2010, Minimizing job completion time variance for service stability on identical parallel machines, *Computers & Industrial Engineering* 58, 729–738.

Li, K., Shi, Y., Yang, S. and Cheng, B., 2011, Parallel machine scheduling problem to minimize the makespan with resource dependent processing times, *Applied Soft Computing*.

Liao, L. and Sheen, G., 2006, Parallel machine scheduling with machine availability and eligibility constraints, *European Journal of Operational Research*, 3-2.

Lin, B. M. T. and Jeng, A. A. K., 2004, Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs, *Int. J. Production Economics*, 91, 121–134.

**KAYNAKLAR DİZİNİ (devam)**

Lin, C. and Liao, C., 2004, Makespan minimization subject to flowtime optimality on identical parallel machines, *Computers & Operations Research*, 31, 1655–1666.

Lin, S., Lee, Z., Ying, K. and Lu, C., 2010, Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates, *Computers & Operations Research*, 38, 809–815.

Liu, M. and Wu, C., 2003, Scheduling algorithm based on evolutionary computing in identical parallel machine production line, *Robotics and Computer Integrated Manufacturing* 19 (2003) 401–407.

Lushchakova, I. N., 2006, Two machine preemptive scheduling problem with release dates, equal processing times and precedence constraints, *European Journal of Operational Research*, 171, 107–122.

Mandel, M. and Mosheiov G., 2001, Minimizing maximum earliness on parallel identical machines, *Computers & Operations Research*, 28, 2-4.

Mazdeh, M. M., Zaerpour, F., Zareei, A. and Hajinezhad, A., 2010, Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs, *Applied Mathematical Modelling* , 34, 1498–1510.

Mellouli, R., Sadfi, C. and Chu, C. and Kacem, I., 2009, Identical parallel-machine scheduling under availability constraints to minimize the sum of completion times, *European Journal of Operational Research*, 197, 1150–1165.

Min, L. and Cheng, W., 1999, A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines, *Artificial Intelligence in Engineering* 13, 399–403.

Min, L. and Cheng, W., 2006, Genetic algorithms for the optimal common due date assignment and the optimal scheduling policy in paralel machine earliness/tardiness scheduling problems, *Robotics and Computer-Integrated Manufacturing*, 22, 279–287.

Mohri, S., Masuda, T. and Ishii, H., 1999, Bi-criteria scheduling problem on three identical parallel machines, *Int. J. Production Economics*, 60, 61, 529–536.

Mosheiov, G., 2001, A common due-date assignment problem on parallel identical machines, *Computers & Operations Research*, 28, 13.

Moukrim, A. and Quilliot, A., 2005, Optimal preemptive scheduling on a fixed number of identical parallel machines, *Operations Research Letters*, 33, 1.



### KAYNAKLAR DİZİNİ (devam)

Muralidhar, A. and Alwarsamy, T., 2009, Multi-objective Optimization of Parallel Machine Scheduling Using Fuzzy Logic and Simulated Annealing, *International Journal of Applied Engineering Research*, 4, 2141–2148.

Nelson, R.T., Sarin, R.K. and Daniels, R.L., 1986, Scheduling with multiple performance measures: the one machine case, *Management Science*, 32, 4, 464-479.

Nessah, R., Yalaoui, F. and Chu C., 2006, A branch-and-bound algorithm to minimize total weighted completion time on identical parallel machines with job release dates, *Computers & Operations Research*, 35, 1176 – 1190.

Okołowski, D. and Gawiejnowicz, S., 2010, Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect, *Computers & Industrial Engineering*, 59, 272–279.

Omar, M.K. and Teo, S.C., 2006, Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach, *Applied Mathematics and Computation*, 283-224.

Özkan, S. E., 2009, Farklı kapasiteli paralel makinelerin dinamik çizelgelenmesi için sezgisel bir algoritma ve uygulaması, Yüksek lisans tezi, Çukurova Üniversitesi, 84s. (yayınlanmamış)

Park, M. and Kim, Y., 1997, Search Heuristics for a Parallel Machine Scheduling Problem with Ready Times and Due Dates, *Computers ind. Engng*, 33, 2-3-4.

Park, Y., Kim, S. and Lee, Y., 2000, Scheduling jobs on parallel machines applying neural network and heuristic rules, *Computers & Industrial Engineering*, 38, 13.

Pinedo, M., 2002, *Scheduling Theory, Algorithms, and Systems*, Prentice Hall, New Jersey.

Pinedo, M. and Chao, X., 1999, *Operations Scheduling with Applications In Manufacturing And Services*, McGraw-Hill international Editions, Singapore.

Rajakumar, S., Arunachalam, V.P. and Salladurai, V., 2004, Workflow balancing strategies in parallel machine scheduling, *International Journal of advanced Manufacturing Technology*, 23, 366-374.

Saad, A., Kawamura, K., and Biswas, G., 1997, Performance Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems. *International Journal of Automation and Soft Computing*, 3, 229-248.

**KAYNAKLAR DİZİNİ (devam)**

Sağır, M. ve Saraç, T., 2007, Hazırlık zamanı kaynaklı üretim kayıplarını enazlayan çok amaçlı bir çizelgeleme algoritması, *Endüstri Mühendisliği Dergisi*, 18, 4, 14-25.

Saraç, T. ve Sipahioğlu, 2009, A., Plastik Enjeksiyon Makinelerinin Çizelgenmesi Problemi, *Endüstri Mühendisliği Dergisi*, 20, 2, 2-14.

Sarıççek, İ. and Çelik, C., 2011, Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness, *Applied Mathematical Modelling*, 35, 4117–4126.

Sarıkaya, B., 2005, Otomotiv yan sanayiinde bir üretim çizelgeleme probleminin incelenmesi, *Çukurova Üniversitesi*, 75s. (yayınlanmamış)

Schaller, J., 2009, Note on Shim and Kim's lower bounds for scheduling on identical paralel machines to minimize total tardiness, *European Journal of Operational Research*, 197, 422–426.

Sen, T., and Gupta, S. K., 1983 , A Branch and Bound to Solve a Bicriterion Scheduling Problem, *IEE Transactions*, 15, 84-88.

Shabtay, D. and Kaspi M., 2006, Parallel machine scheduling with a convex resource consumption function. *European Journal of Operational Research*, 173, 3–13.

Shim, S. and Kim, Y., 2007, Scheduling on parallel identical machines to minimize total tardiness, *European Journal of Operational Research*, 177, 135–146.

Shim, S. and Kim, Y., 2008, A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property, *Computers & Operations Research*, 35, 863 – 875.

Su, L., 2009, Scheduling on identical parallel machines to minimize total completion time with deadline and machine eligibility constraints, *Springer-Verlag London Limited*.

Sun, K. and Li, H., 2010, Scheduling problems with multiple maintenance activities and non-preemptive jobs on two identical parallel machines, *Int. J. Production Economics*, 124, 151–158.

Sun, H. and Wang, G., 2003, Parallel machine earliness and tardiness scheduling with proportional weights, *Computers & Operations Research*, 30, 1.

Süer, G. A., Pico, F. and Santiago, A., 1997, Identical machine scheduling to minimize the number of tardy jobs when lot splitting is allowed, *Computers & Industrial Engineering*, 33, 277-280.

**KAYNAKLAR DİZİNİ (devam)**

Tahar, D., Yalaoui, F., Chu, C., and Amodeo, L., 2006, A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times, *Int. J. Production Economic*, 99, 63–73.

Tan, Z., Chen, Y. and Zhang, A., Parallel machines scheduling with machine maintenance for minsum criteria, *European Journal of Operational Research* 212, 287–292.

Tang, L. and Liu, G., 2006, A mathematical programming model and solution for scheduling production orders in Shanghai Baoshan Iron and Steel Complex, *European Journal of Operational Research*, 182, 1453–1468.

Tarantilis, C. D., Ioannou, G. and Prastacos, G., 2004, Advanced Vehicle Routing Algorithms For Complex Operations Management Problems, *Journal of Food Engineering*, Elsevier Ltd.

Tavakkoli-Moghaddam, R. , 2007, A new ILP model for identical parallel-machine scheduling with family setup times minimizing the total weighted flow time by a genetic algorithm, *IJE Transactions*.

Timkovsky, V. G., 2003, Identical parallel machines vs. unit-time shops and preemptions vs. chains in scheduling complexity, *European Journal of Operational Research*, 149, 355–376.

Timkovsky, V. G., 1998, Is a unit-time job shop not easier than identical parallel machines?, *Discrete Applied Mathematics*, 85, 149-162.

Tuong, N., Soukhal, A. and Billaut, J., 2009, A new dynamic programming formulation for scheduling independent tasks with common due date on parallel machines, *European Journal of Operational Research*, 202, 646–653.

Van Wassenhove, L. N., and Gelders, F., 1978, Four Solution Techniques for a General One Machine Scheduling Problem: A Comparative Study, *European Journal of Operational Research*, 2, 4, 281-290.

Webster, S. and Azizoglu, M., 2001, Dynamic programming algorithms for scheduling parallel machines with family setup times, *Computers & Operations Research*, 28, 1.

Xu, D., Sun, K. and Li, H., 2008, Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan, *Computers & Operations Research*, 35, 1344 – 1349.

Xu, H., Chen, L., Ye, D. and Zhang, G., 2011, Scheduling on two identical machines with a speed-up resource, *Information Processing Letters* 111, 831–835.

**KAYNAKLAR DİZİNİ (devam)**

Unlu, Y. and Mason, S., 2010, Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems, *Computers & Industrial Engineering*, 58, 785–800.

Yalaoui, F. and Chu, C., 2002, Parallel machine scheduling to minimize total tardiness, *Int. J. Production Economics*, 76, 265–279.

Yang W.–H., Liao C.–J., 1999, Survey of scheduling research involving setup times, *International Journal of Systems Science*, 30, 2, 143–155.

Ying, K. and Cheng, H., 2010, Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic, *Expert Systems with Applications* 37, 2848–2852.

Yörükoğlu, V., 2007, Otomotiv sektöründe imalat çizelgeleme probleminin tamsayı programlama ile çözümü, *Kocaeli Üniversitesi*, 88s. (yayınlanmamış)

Yu, A. and Gu, X., 2008, A Coupled Transiently Chaotic Neural Network Approach for Identical Parallel Machine Scheduling, *Scimedirect*, 34, 6.

Zouba, M., Baptiste, P. and Rebaine, D., 2009, Scheduling identical parallel machines and operators within a period based changing mode, *Computers & Operations Research*, 36, 3231- 3239.

## **EKLER**

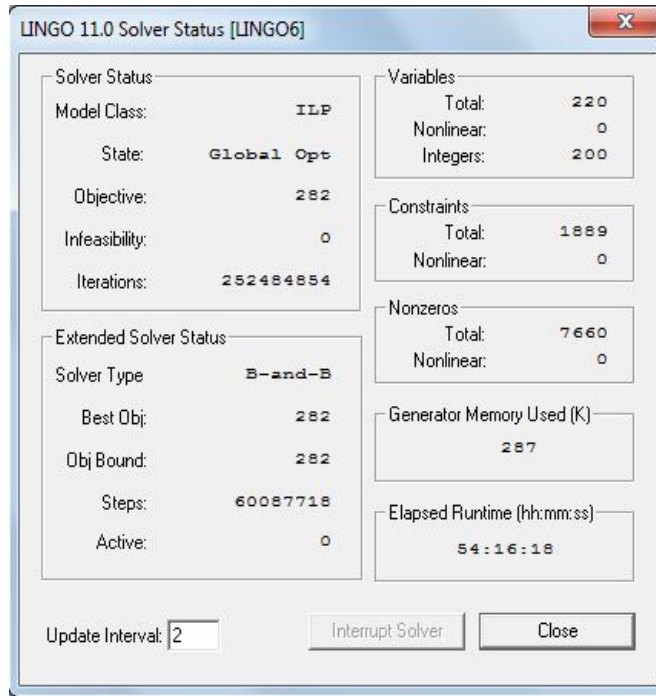
**EK-1.** 10 iş, 2 makineli örnek problemin en iyi çözümü (ekran görüntüsü)

**EK-2.** 10 iş, 3 makineli örnek problemin en iyi çözümü (ekran görüntüsü)

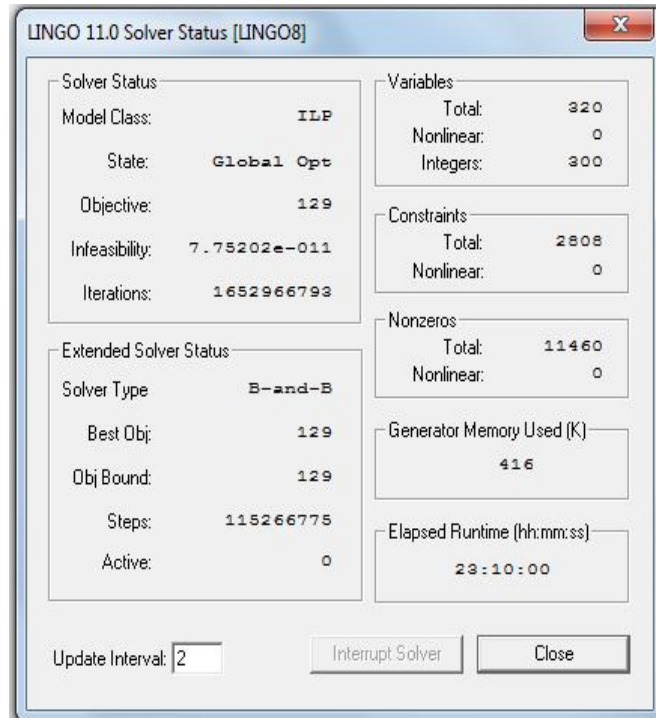
**EK-3.** Model, model için kullanılan kalıp kodu, kalıptaki göz sayısı ve karşı gelen birim üretim süresi (sn)

**EK-4.** C# programlama kodlarının bir kısmı (FrmMain.cs, FrmSonuc.cs, Frmveritablosu.cs, IParti.cs, IUrun.cs, IMakine.cs sınıflarına ait kodlar)

**EK-1.** 10 iş, 2 makineli örnek problemin en iyi çözümü (ekran görüntüsü)



**EK-2.** 10 iş, 3 makineli örnek problemin en iyi çözümü (ekran görüntüsü)



**EK-3.** Model, model için kullanılan kalıp kodu ve karşı gelen birim üretim süresi (sn)

<b>Model Kodu</b>	<b>Kalıp Kodu</b>	<b>Kalıp Gözü</b>	<b>Birim Üretim Süresi(sn)</b>
S54121	FF_S54121	2	37
S54120	FF_S54120	2	37
S54121B	FF_S54121B	2	34
L54135	FF_L54135	1	34
L54135B	FF_L54135B	1	34
F54102	FF_F54102	2	34
F54090N	FF_F54090N	2	34
F60096EB	FF_F60096EB	1	34
S60125B	FF_S60125B	1	34
L60130B	FF_L60130B	1	34
L54261	FF_L54261	1	34
S54201	FF_S54201	1	34
S54200B	FF_S54200B	1	34
K70475N	FF_K70475N	1	34
	FRZ_K70475N	2	42
D70465N	FF_D70465N	1	34
	FRZ_D70465N	2	42
D70500N	FF_D70500N	1	34
	FRZ_D70500N	2	42
D70500NE	FF_D70500NE	1	34
	FRZ_D70500NE	2	42
D70470N	FF_D70470N	1	34
	FRZ_D70470N	2	42
D70470NE	FF_D70470NE	1	34
	FRZ_D70470NE	2	42
D74500N	FF_D74500N	1	34
	FRZ_D74500N	1	34
D74500NE	FF_D74500NE	1	34
	FRZ_D74500NE	1	34
D70460N	FF_D70460N	1	34
	FRZ_D70460N	2	42
D60330N	FF_D60330N	1	34
	FRZ_D60330N	2	42
D60390N	FF_D60390N	1	34
	FRZ_D60390N	2	42
K70475NE	FF_K70475NE	1	34
	FRZ_K70475NE	1	34
D70545NE	FF_D70545NE	1	34
	FRZ_D70545NE	1	34
D70540NE	FF_D70540NE	1	34
	FRZ_D70540NE	1	34
D70540N	FF_D70540N	1	34
	FRZ_D70540N	1	34

**EK-3. (devam)**

D70540N	FF_D70540N	1	34
	FRZ_D70540N	1	34
D70430N	FF_D70430N	1	34
	FRZ_D70430N	1	34
D70450N	FF_D70450N	1	34
	FRZ_D70450N	1	34
D74575N	FF_D74575N	1	34
	FRZ_D74575N	1	34
D74575NE	FF_D74575NE	1	34
	FRZ_D74575NE	1	34
F60194N	FF_F60194N	1	34
F60260	FF_F60260	1	34
D70550	FF_D70550	1	34
	FRZ_D70550	1	34
D54177	FF_D54177	1	34
	FRZ_D54177	2	42
D54177B	FF_D54177B	1	34
	FRZ_D54177B	2	42
F54198N	FF_F54198N	1	34
F54210	FF_F54210	1	34
F54211	FF_F54211	1	34
L60325	FF_L60325	1	34
L60370	FF_L60370	1	34
S60330	FF_S60330	1	34
F54220N	FF_F54220N	1	34
F60240	FF_F60240	1	34
F60232NE	FF_F60232NE	1	34
D54231	FF_D54231	1	34
	FRZ_D54231	2	42
D54245N	FF_D54245N	1	34
	FRZ_D54245N	2	42
D70450	FF_D70450	1	34
	FRZ_D70450	1	34
D60300	FF_D60300	1	34
	FRZ_D60300	2	42
D60330	FF_D60330	1	34
	FRZ_D60330	2	42
D54220	FF_D54220	1	34
	FRZ_D54220	2	42
D54271	FF_D54271	1	34
	FRZ_D54271	2	42
D54221B	FF_D54221B	1	34
	FRZ_D54221B	2	42



**EK-3. (devam)**

D70451	FF_D70451	1	34
	FRZ_D70451	1	34
D60301	FF_D60301	1	34
	FRZ_D60301	2	42
K70401H	FF_K70401H	1	34
	FRZ_K70401H	1	34
K70415H	FF_K70415H	1	34
	FRZ_K70415H	1	34
K54242H	FF_K54242H	1	34
	FRZ_K54242H	2	42
K54242	FF_K54242	1	34
	FRZ_K54242	2	42
K60340	FF_K60340	1	34
	FRZ_K60340	2	42
K60320H	FF_K60320H	1	34
	FRZ_K60320H	2	42
K54299H	FF_K54299H	1	34
	FRZ_K54299H	2	42
K54325H	FF_K54325H	1	34
	FRZ_K54325H	2	42
K60315H	FF_K60315H	1	34
	FRZ_K60315H	2	42
K54242B	FF_K54242B	1	34
	FRZ_K54242B	2	42
K54270HB	FF_K54270HB	1	34
	FRZ_K54270HB	2	42
K54296	FF_K54296	1	34
	FRZ_K54296	2	42
K54308	FF_K54308	1	34
	FRZ_K54308	2	42
K54230HB	FF_K54230HB	1	34
	FRZ_K54230HB	2	42
K60320NE	FF_K60320NE	1	34
	FRZ_K60320NE	2	42
K60320N	FF_K60320N	1	34
	FRZ_K60320N	2	42
D70410	FF_D70410	1	34
	FRZ_D70410	1	34
D84610NE	FF_D84610NE	1	34
	FRZ_D84610NE	1	34
D84610N	FF_D84610N	1	34
	FRZ_D84610N	1	34
L54315EB	FF_L54315EB	1	34

**EK-3. (devam)**

L54210B	FF_L54210B	1	34
L54211	FF_L54211	1	34
K54243	FF_K54243	1	34
	FRZ_K54243	2	42
K60310H	FF_K60310H	1	34
	FRZ_K60310H	2	42
K60310	FF_K60310	1	34
	FRZ_K60310	2	42
F54102B	FF_F54102B	2	42
L54315	FF_L54315	1	34
L54315B	FF_L54315B	1	34

**EK-4. C# programlama kodları**FrmMain.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Collections;
using System.Configuration;

namespace UrunUretimPlanlayicisi
{
    public partial class FrmMain : Form
    {
        public List<IParti> UretimPartileri{get;set;}
        public List<IMakine> Makineler { get; set; }
        Hashtable uretilebilecekUrunler;
        public FrmMain()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            frmVeriTablosu veriTablosu = new
            frmVeriTablosu(this.uretilebilecekUrunler,this.UretimPartileri,
            monthCalendar1.SelectionStart);
            veriTablosu.Show();
        }
    }
}

```

**EK-4. (devam)**

```

private void FrmMain_Load(object sender, EventArgs e)
{
    uretilebilecekUrunler = new Hashtable();
    UretimPartileri = new List<IParti>();
    comboBoxMakineSayisi.SelectedIndex = 0;
}

private void ToolStripMenuItemUrunDosyasıAc_Click(object
sender, EventArgs e)
{
    //dosya acma penceresini yaratir.
    OpenFileDialog dialog = new OpenFileDialog();
    if (dialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)//Dosya acma penceresini gosterir
ve OK'e basilmasini kontrol eder
    {
        ExcelImporter impo=new ExcelImporter();//Yeni bir
excel importer nesnesi yaratir
        List<IUrun> urunler =
impo.UrunBilgileriniCek(dialog.FileName);//urun bilgileri excellden
cekiliyor.

        this.UretimPartileri =
impo.PartiBilgileriniCek(dialog.FileName,urunler);//Partiler excellden
cekiliyor

        foreach (IUrun urun in urunler)//Her bir urun bir
tabloya konuluyor.
        {
            this.uretilebilecekUrunler.Add(urun.ID,
urun);//Eger ozellik birden fazla girilmisse burasi hata verir.
        }
        MessageBox.Show("Ürün bilgileri okundu");//Urunlerin
okunma isleminin bittigi kullaniciya haber veriliyor
    }
}

/// <summary>
/// Uygun makineyi makine uzerindeki is yuku kontrol edilerek
bulan metoddur
/// </summary>
/// <returns>Uygun makine getirilir</returns>
public IMakine UygunMakineBul()
{
    int makineIndeks = 0;
    float makineIsyuku = float.MaxValue;
    for (int i = 0; i < this.Makineler.Count; i++)
    {
        if (this.Makineler[i].IslerinToplamBitmeSuresi() <
makineIsyuku)
        {

```

**EK-4. (devam)**

```

        makineIndeks = i;
        makineIsyuku =
this.Makineler[i].IslerinToplamBitmeSuresi();
    }
    }
    return this.Makineler[makineIndeks];
}

/// <summary>
/// Parti tipine gore uygun makine bulunmasi icin kullanılan
metoddur.
/// Toplam gecikme deęeri enküçük olan makineyi döner.

/// </summary>
/// <param name="yeniParti">Uygun makine aranılan
partidir.</param>
/// <returns>Uygun makineyi doner</returns>
public IMakine UygunMakineBul(IParti yeniParti,bool a)
{
    int makineIndeks = 0;
    int isIndeks = 0;
    IMakine[] makineKopyalari=new
OzdesMakineV1[this.Makineler.Count];
    this.Makineler.CopyTo(makineKopyalari);
    double makineToplamGecikmeSuresi = double.MaxValue;

    for (int i = 0; i < makineKopyalari.Length; i++)
    {

        IMakine makine = new OzdesMakineV1 //i. makinenin bir
kopyasini olusturyoruz
        {
            BaslangicTarihi =
makineKopyalari[i].BaslangicTarihi,
            Partiler = new
List<IParti>(makineKopyalari[i].Partiler)
        };
        makine.PartiAta(yeniParti);
        double toplamGecikme = makine.ToplamGecikmeSuresi;
        if (toplamGecikme < makineToplamGecikmeSuresi)
        {
            makineIndeks = i;
            isIndeks = makine.Partiler.Count;
            makineToplamGecikmeSuresi = toplamGecikme;
        }
    }
}

```

**EK-4. (devam)**

```

        int uygunisIndeksi = -1;
        for (int j = 0; j < this.Makineler[i].Partiler.Count;
j++)
        {
            if
(this.Makineler[i].Partiler[j].UretilecekUrun.UrunAdi.Equals(yeniParti
.UretilecekUrun.UrunAdi))
            {
                uygunisIndeksi = j;
            }
        }

        if (uygunisIndeksi != -1)//-1'e esit olmamasi makinede
ayni tipten urun oldugunu gosterir.
        {
            makine = new OzdesMakineV1//i. makinenin bir
kopyasini olusturyoruz
            {
                BaslangicTarihi =
makineKopyalari[i].BaslangicTarihi,
                Partiler = new
List<IParti>(makineKopyalari[i].Partiler)
            };
            makine.PartiAta(uygunisIndeksi+1,yeniParti);
            toplamGecikme = makine.ToplamGecikmeSuresi;
            if (toplamGecikme < makineToplamGecikmeSuresi)
            {
                makineIndeks = i;
                isIndeks = uygunisIndeksi+1;
                makineToplamGecikmeSuresi = toplamGecikme;
            }
        }

        //Bu noktada butun makineleri dolastik ve en iyi makinenin
en iyi yerini bulmus olduk
        //en iyi makine makineIndeks
        //en iyi yer ise isIndeks
        this.Makineler[makineIndeks].PartiAta(isIndeks,
yeniParti);//Makinede isIndeks olan yere atar
        return this.Makineler[makineIndeks];
    }

    private void buttonPlanlamayiBaslat_Click(object sender,
EventArgs e)
    {
        int makineSayisi=0;
        makineSayisi=
int.Parse(comboBoxMakineSayisi.SelectedItem.ToString());//Kac makine
ile planlama yapilacagini bulur

```

**EK-4. (devam)**

```

        List<IParti> partiler =
this.UretimPartileri.ToList();//partilerin orjinal sirasi bozulmasin
diye kopya yaratir.
        partiler.Sort();//Partileri siralar, partiv1 icerisindeki
compare metoduna gore
        this.Makineler = new List<IMakine>();
        for (int i = 0; i < makineSayisi; i++)//comboboxtan secili
makine sayisi kadar makine yaratir
        {
            IMakine makine = new OzdesMakineV1();
            makine.BaslangicTarihi =
monthCalendar1.SelectionStart;
            this.Makineler.Add(makine);
        }

        foreach (IParti parti in partiler)//Her bir parti icin
uygun makine bulur
        {

            UygunMakineBul().PartiAta(parti);
        }

        FrmSonuc sonuc = new FrmSonuc(this.Makineler);//Islemler
bittiginde kullanicya inceleyebilmesi icin
                                                    //Sonuc
ekranini gosterir
            sonuc.Show();
        }

        private void buttonSetupliAnalizV2_Click(object sender,
EventArgs e)
        {
            int makineSayisi = 0;
            makineSayisi =
int.Parse(comboBoxMakineSayisi.SelectedItem.ToString());//Kac makine
ile planlama yapilacagini bulur
            List<IParti> partiler =
this.UretimPartileri.ToList();//partilerin orjinal sirasi bozulmasin
diye kopya yaratir.
            partiler.Sort();//partileri siralar, partiv1deki compare
metodu ile
            this.Makineler = new List<IMakine>();
            for (int i = 0; i < makineSayisi; i++)//comboboxta secili
olan kadar makine yaratir.
            {
                IMakine makine = new OzdesMakineV1();
                makine.BaslangicTarihi =
monthCalendar1.SelectionStart;
                this.Makineler.Add(makine);
            }
        }

```

**EK-4. (devam)**

```

        foreach (IParti parti in partiler)//Her bir parti icin
        uygun makine bulur
        {
            IMakine makine = UygunMakineBul(parti,true);
        }
        FrmSonuc sonuc = new FrmSonuc(this.Makineler);//Islemler
        bittiginde kullanicya inceleyebilmesi icin
        //Sonuc ekranini gosterir
        sonuc.Show();
    }
}

```

**FrmSonuc.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace UrunUretimPlanlayicisi
{
    public partial class FrmSonuc : Form
    {
        public List<IMakine> makineler;
        public FrmSonuc()
        {
            InitializeComponent();
        }

        public FrmSonuc(List<IMakine> makineler)
        {
            InitializeComponent();
            this.makineler = makineler;
        }
        private void FrmSonuc_Load(object sender, EventArgs e)
        {
            this.vardiyaKacSattir =
            int.Parse(ConfigurationManager.AppSettings["BirVardiyaKacSaattir"]).ToString();
            listView1.Items.Clear();
        }
    }
}

```

**EK-4. (devam)**

```

        for (int i = 0; i < this.makineler.Count;i++ )//Comboboxa
Makine 1, Makine 2 ... seklinde yazar
        {
            comboBoxFrmSonucAtanmisMakineler.Items.Add("Makine " +
(i + 1));
        }
    }
    int vardiyaKacSattir = 8;

    /// <summary>
    /// Gecikme suresini Vardiya:Saate:Dakika:Saniye cinsine
cevirir
    /// </summary>
    /// <param name="gecikmeSuresi">Bir partinin gecikme
suresidir.</param>
    /// <returns>Ilgili stringi getirir.</returns>
    public string GecikmeSuresiTextiOlustur(double gecikmeSuresi)
    {
        string strGecikmeSuresi = "";
        TimeSpan span =
TimeSpan.FromSeconds(Math.Abs(gecikmeSuresi));

        int vardiyaSayisi = (int)((int)span.TotalHours/
vardiyaKacSattir);
        if (gecikmeSuresi >= 0)
        {
            strGecikmeSuresi =
string.Format("{0:D2}vrd:{1:D2}sa:{2:D2}dk:{3:D2}sn",
                vardiyaSayisi,
                (int)span.TotalHours - vardiyaSayisi *
vardiyaKacSattir,
                span.Minutes,
                span.Seconds);
        }
        else
        {
            strGecikmeSuresi = string.Format("----
{0:D2}vrd:{1:D2}sa:{2:D2}dk:{3:D2}sn",
                vardiyaSayisi,
                (int)span.TotalHours - vardiyaSayisi *
vardiyaKacSattir,
                span.Minutes,
                span.Seconds);
        }
        return strGecikmeSuresi;
    }
}

```



**EK-4. (devam)**

```

        /// <summary>
        /// List viewe secili makinenin partilerini doldurmak icin
        kullanilir.
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void
        comboBoxFrmSonucAtanmisMakineler_SelectedIndexChanged(object sender,
        EventArgs e)
        {
            IMakine aktif =
            this.makineler[comboBoxFrmSonucAtanmisMakineler.SelectedIndex];
            listView1.Items.Clear();
            int i = 0;
            double toplamGecikmeSuresi = 0;
            foreach (IParti parti in aktif.Partiler)//Her bir partiyi
            listview'e ekler.
            {
                ListViewItem item = new
                ListViewItem(parti.UretilecekUrun.UrunAdi);
                ListViewItem.ListViewSubItem termin = new
                ListViewItem.ListViewSubItem(item, parti.Termin.ToShortDateString());
                ListViewItem.ListViewSubItem adet = new
                ListViewItem.ListViewSubItem(item,
                parti.UretilecekUrunAdedi.ToString());
                ListViewItem.ListViewSubItem tahminiBaslangicTarihi =
                new ListViewItem.ListViewSubItem(item,
                aktif.BirIsinBaslamaTarihiniHesapla(i).ToShortDateString() + " " +
                aktif.BirIsinBaslamaTarihiniHesapla(i).ToShortTimeString());
                ListViewItem.ListViewSubItem tahminiBitisTarihi = new
                ListViewItem.ListViewSubItem(item,
                aktif.BirIsinBitmeTarihiniHesapla(i).ToShortDateString() + "
                "+aktif.BirIsinBitmeTarihiniHesapla(i).ToShortTimeString());
                double gecikmeSuresi =
                aktif.BirIsinGecikmeSuresiniHesapla(i);
                if(gecikmeSuresi>0)
                    toplamGecikmeSuresi += gecikmeSuresi;

                ListViewItem.ListViewSubItem tahminiGecikmeSuresi =
                new ListViewItem.ListViewSubItem(item,
                GecikmeSuresiTextiOlustur(gecikmeSuresi));

                item.SubItems.Add(termin);
                item.SubItems.Add(adet);
                item.SubItems.Add(tahminiBaslangicTarihi);
                item.SubItems.Add(tahminiBitisTarihi);
                item.SubItems.Add(tahminiGecikmeSuresi);
                listView1.Items.Add(item);
                i++;
            }
        }
    }

```

**EK-4. (devam)**

```

        labelFrmSonucToplamGecikmeSuresi.Text =
GecikmeSuresiTextiOlustur(toplamGecikmeSuresi);

    }

    private void listView1_SelectedIndexChanged(object sender,
EventArgs e)
    {

    }
}

```

**Frmveritablosu.cs**

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Collections;

namespace UrunUretimPlanlayicisi
{
    public partial class frmVeriTablosu : Form
    {
        Hashtable uretilebilecekUrunler;
        List<IParti> uretimPartileri;

        IUrun urun = null;
        IParti parti = null;

        DateTime baslangicZamani;

        public frmVeriTablosu()
        {
            InitializeComponent();
            uretilebilecekUrunler = new Hashtable();
            uretimPartileri = new List<IParti>();
        }
    }
}

```

**EK-4. (devam)**

```

        public frmVeriTablosu(Hashtable
uretilebilecekUrunler, List<IParti> uretimPartileri, DateTime
baslangicZamani)
        {
            InitializeComponent();
            this.uretilebilecekUrunler = uretilebilecekUrunler;
            this.uretimPartileri = uretimPartileri;
            this.baslangicZamani = baslangicZamani;
            foreach (IUrun urun in uretilebilecekUrunler.Values)
            {

                ((DataGridViewComboBoxColumn) dataGridViewVeriTablosu.Columns[0]).Items
                .Add(urun);

            }

            ((DataGridViewComboBoxColumn) dataGridViewVeriTablosu.Columns[0]).Items
            .Add("unassigned");

            ((DataGridViewComboBoxColumn) dataGridViewVeriTablosu.Columns[0]).Defau
            ltCellStyle.NullValue = "unassigned";

            ((DataGridViewComboBoxColumn) dataGridViewVeriTablosu.Columns[0]).Displ
            ayMember = "UrunAdi";

            ((DataGridViewComboBoxColumn) dataGridViewVeriTablosu.Columns[0]).Value
            Member = "Self";

            for (int i = 0; i < this.uretimPartileri.Count;
i++) //Uretim partileri listede gosterilmek icin hazirlaniyorlar
            {
                dataGridViewVeriTablosu.Rows.Add(1);
                IParti parti=(IParti) this.uretimPartileri[i];

                ((DataGridViewComboBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[0]).V
                alue = parti.UretilecekUrun;

                ((DataGridViewTextBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[1]).Va
                lue = parti.UretilecekUrun.KalipGozuSayisi.ToString();

                ((DataGridViewTextBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[2]).Va
                lue = parti.UretilecekUrun.BirimUretimSuresi.ToString();

                ((DataGridViewTextBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[3]).Va
                lue = parti.UretilecekUrunAdedi;

                ((DataGridViewTextBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[4]).Va
                lue = parti.Termin.ToShortDateString();

                ((DataGridViewTextBoxCell) dataGridViewVeriTablosu.Rows[i].Cells[5]).Va
                lue = parti.PartiUretimSuresi.ToString();
            }
        }

```

**EK-4. (devam)**

```

private void frmVeriTablosu_Load(object sender, EventArgs e)
{
    // Add this event handler to be able to
    // configure the embedded ComboBox control
    dataGridViewVeriTablosu.EditingControlShowing += new
DataGridViewEditingControlShowingEventHandler(dataGridViewVeriTablosu_
EditingControlShowing);
    // Change to edit mode when user enters a cell.
}
int rowIndex;
ComboBox comboBox=null;

private void
dataGridViewVeriTablosu_EditingControlShowing(object sender,
DataGridViewEditingControlShowingEventArgs e)
{
    if (dataGridViewVeriTablosu.CurrentCell.ColumnIndex == 0)
    {
        rowIndex = dataGridViewVeriTablosu.CurrentRow.Index;
        comboBox = e.Control as ComboBox;

        comboBox.SelectedIndexChanged += new
EventHandler(comboBox_SelectedIndexChanged);
    }
}
void comboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    int indeks = rowIndex;
    urun = comboBox.SelectedItem as IUrun;

    if (urun != null)
    {
        dataGridViewVeriTablosu.Rows[indeks].Cells[1].Value =
urun.KalipGozuSayisi.ToString();
        dataGridViewVeriTablosu.Rows[indeks].Cells[2].Value =
urun.BirimUretimSuresi.ToString();
    }
}

private void dataGridViewVeriTablosu_CellValueChanged(object
sender, DataGridViewCellEventArgs e)
{
    parti = null;
    if (e.ColumnIndex == 3&&e.RowIndex!=-1)
    {
        urun =
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[0].Value as IUrun;

```

**EK-4. (devam)**

```

foreach (IParti p in uretimPartileri)
{
    if (p.ID == e.RowIndex)
    {
        parti = p;
    }
}
if(parti==null)
{
    parti = new PartiV1();
    parti.ID = e.RowIndex;
    uretimPartileri.Add(parti);
}

parti.UretilecekUrun = urun;
string vas =
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[3].Value.ToString();
int uretilecekUrunAdedi=0;
if (int.TryParse(vas, out uretilecekUrunAdedi))
{
    parti.UretilecekUrunAdedi = uretilecekUrunAdedi;

dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[5].Value =
parti.PartiUretimSuresi.ToString();

}
else
{
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[3].Value = "";
}
}
else if (e.ColumnIndex == 4 && e.RowIndex != -1)
{
    urun =
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[0].Value as IUrun;
foreach (IParti p in uretimPartileri)
{
    if (p.ID == e.RowIndex)
    {
        parti = p;
    }
}
if (parti == null)
{
    parti = new PartiV1();
    parti.ID = e.RowIndex;
    uretimPartileri.Add(parti);
}
}

```

**EK-4. (devam)**

```

        parti.UretilecekUrun = urun;
        string date =
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[4].Value.ToString();
        DateTime vas;
        if (DateTime.TryParse(date, out vas))
        {
            if (vas < this.baslangicZamani)
            {

                MessageBox.Show("Lutfen geçerli bir tarih
giriniz!!!");

dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[4].Value = "";
            }
            else
            {
                parti.Termin = vas;
            }
        }
        else
        {
dataGridViewVeriTablosu.Rows[e.RowIndex].Cells[4].Value = "";
        }
    }
}

private void buttonVeriTablosuTAMAM_Click(object sender,
EventArgs e)
{
    this.Close();
}
}
}

```

**IMakine.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace UrunUretimPlanlayicisi
{
    public interface IMakine
    {
        /// <summary>
        /// Bir makedede islerin baslangic tarihi.
        /// </summary>
        DateTime BaslangicTarihi { get; set; }
    }
}

```

**EK-4. (devam)**

```

    /// <summary>
    /// Makine icerisindeki partilerin tutuldugu listedir.
    /// </summary>
    List<IParti> Partiler { get; set; }

    /// <summary>
    /// Parti listesine bir parti atamak icin kullanılan metoddur.
    /// </summary>
    /// <param name="parti">Listeye yeni eklenecek
partidir</param>
    void PartiAta(IParti parti);

    /// <summary>
    /// Parti listesine bir partiyi belirli bir konuma eklemek
için kullanılan metoddur.
    /// arkasından gelen parti elamanlari birer kaymaktadır.
    /// </summary>
    /// <param name="indeks">Partinin hangi konuma eklenecegini
gosteren parametredir.</param>
    /// <param name="parti">Listeye yeni eklenecek
partidir.</param>
    void PartiAta(int indeks, IParti parti);

    /// <summary>
    /// Bir makinenin izerindeki islerin toplam bitme suresini
saniye getirir.
    /// </summary>
    /// <returns>Islerin toplam bitme suresi sonucunu saniye
cinsinden getirir</returns>
    float IslerinToplamBitmeSuresi();

    /// <summary>
    /// Partiler içerisinde bulunan bir partinin tahmini bitme
suresini hesaplamak icin kullanılan
    /// metoddur. Verilen indeksin oncesindeki butun partiler ve
setup sureleri toplanir.
    /// </summary>
    /// <param name="isinIndeksi">Tahmini bitis suresi hesaplanmak
istenen partinin indeksidir.</param>
    /// <returns>Indeksi verilen partinin tahmini bitis suresi
sonucunu getirir </returns>
    float BirIsinBitmeSuresiniHesapla(int isinIndeksi);

    /// <summary>
    /// Indeksi verilen bir isin tahmini bitis tarihini hesaplamak
icin kullanılan metoddur.
    /// Oncelikle indeks verilen isin tahmini bitme suresi
hesaplanir daha sonra cizelgelemenin
    /// baslangic tarihine eklenir
    /// </summary>
    /// <param name="isinIndeksi">Tahmini bitis tarihi bulunacak
partinin indeksidir.</param>
    /// <returns>Tahmini bitis tarihi</returns>
    DateTime BirIsinBitmeTarihiniHesapla(int isinIndeksi);

```

**EK-4. (devam)**

```

        /// <summary>
        /// Indeksi verilen bir isin tahmini baslangic tarihini
hesaplamak icin kullanılan metoddur.
        /// Oncelikle bir isin tahmini bitis tarihi bulunur ve tahmini
tamamlanma suresi bu tarihten cikarilir.
        /// </summary>
        /// <param name="isinIndeksi">Tahmini baslangic tarihi
bulunacak partinin indeksidir</param>
        /// <returns>Tahmini baslangic tarihi</returns>
        DateTime BirIsinBaslamaTarihiniHesapla(int isinIndeksi);

        /// <summary>
        /// Indeksi verilen bir isin tahmini gecikmesini hesaplamak
icin kullanılan metoddur.
        /// Oncelikle isin tahmini bitis tarihi bulunur daha sonra o
isin gercek termini ile farki bulunur.
        /// </summary>
        /// <param name="isinIndeksi">Tahmini gecikme suresi bulunacak
partinin indeksidir.</param>
        /// <returns>Tahmini gecikme suresi</returns>
        double BirIsinGecikmeSuresiniHesapla(int isinIndeksi);

        /// <summary>
        /// Makine Uzerindeki toplam gecikme suresini getirir.
        /// </summary>
        double ToplamGecikmeSuresi { get; }
    }
}

```

**IParti.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace UrunUretimPlanlayicisi
{
    public interface IParti:IComparable
    {
        /// <summary>
        /// Belirleyici sayidir veri tabani olmasi halinde kullanisli
olacaktır
        /// </summary>
        int ID { get; set; }

        /// <summary>
        /// Bir partide uretilecek urun adedir
        /// </summary>
        int UretilecekUrunAdedi { get; set; }
    }
}

```



**EK-4. (devam)**

```

    /// <summary>
    /// Kullanıcının girdiği termin bilgisi
    /// </summary>
    DateTime Termin { get; set; }

    /// <summary>
    /// Üretilecek urundur
    /// </summary>
    IUrun UretilecekUrun { get; set; }

    /// <summary>
    /// Bir partinin gecikmesi bu tarih uzerinden hesaplanacak
    /// </summary>
    DateTime GercekTermin { get; }

    /// <summary>
    /// Parti üretim süresi saniye cinsindedir
    /// </summary>
    float PartiUretimSuresi { get; }

    /// <summary>
    /// Kullanici bir partinin termininden kac vardiya once
    bitecegine karar verir
    /// </summary>
    int PartiKacVardiyaOnceBitmeli { get; set; }
}
}

```

**IUrun.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;

namespace UrunUretimPlanlayicisi
{
    public interface IUrun
    {
        /// <summary>
        /// Veri tabani kullanimlarinda kullanılacak belirleyici
        elemadir.
        /// </summary>
        int ID { get; set; }

        /// <summary>
        /// Urun adidir.
        /// </summary>
        string UrunAdi { get; set; }
    }
}

```

**EK-4. (devam)**

```
    /// <summary>
    /// Bir urunun birim uretim suresidir.
    /// </summary>
    int BirimUretimSuresi { get; set; }

    /// <summary>
    /// Kalip gozu sayisidir.
    /// </summary>
    int KalipGozuSayisi { get; set; }

    /// <summary>
    /// Kendisini donduryor. Data grid view uygulaması icersinde
    kullanılması için eklenmiştir!!!
    /// </summary>
    IUrun Self { get; set; }
}
}
```