



Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü

İşletme Anabilim Dalı

**KOMPOZİT ÜRETİMİNDE OTOKLAV YÜKLEME VE
ÇİZELGELEME İÇİN YENİ BİR MODEL ÖNERİSİ: SAVUNMA
SANAYİİ UYGULAMASI**

Volkan HASKILIÇ

Doktora Tezi

Ankara, 2019

KOMPOZİT ÜRETİMİNDE OTOKLAV YÜKLEME VE ÇİZELGELEME İÇİN YENİ BİR
MODEL ÖNERİSİ: SAVUNMA SANAYİİ UYGULAMASI

Volkan HASKILIÇ

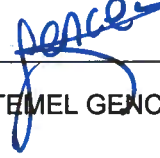
Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü
İşletme Ana Bilim Dalı

Doktora Tezi

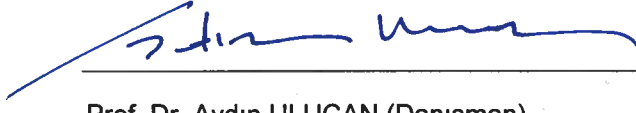
Ankara, 2019

KABUL VE ONAY

Volkan HASKILIÇ tarafından hazırlanan "KOMPOZİT ÜRETİMİNDE OTOKLAV YÜKLEME VE ÇİZELGELEME İÇİN YENİ BİR MODEL ÖNERİSİ: SAVUNMA SANAYİİ UYGULAMASI" başlıklı bu çalışma, 29.05.2019 tarihinde yapılan savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından DOKTORA TEZİ olarak kabul edilmiştir.



Prof. Dr. Cevriye TEMEL GENCER (Başkan)



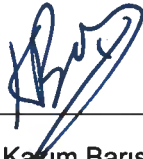
Prof. Dr. Aydın ULUCAN (Danışman)



Prof. Dr. Fazıl GÖKGÖZ



Prof. Dr. Umur TOSUN



Doç. Dr. Kazım Barış ATICI

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylıyorum.

Prof. Dr. Musa Yaşar SAĞLAM

Enstitü Müdürü

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinleri yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan "**Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**" kapsamında tezim aşağıda belirtilen koşullar haricince YÖK Ulusal Tez Merkezi / H.Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir. ⁽¹⁾
- Enstitü / Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ay ertelenmiştir. ⁽²⁾
- Tezimle ilgili gizlilik kararı verilmiştir. ⁽³⁾

29/05/2019


Volkan HASKILIÇ

¹"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"

- (1) Madde 6. 1. Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.
- (2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internette paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.
- (3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir *. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.
Madde 7.2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

ETİK BEYAN

Bu çalışmadaki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi, görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, yararlandığım kaynaklara bilimsel normlara uygun olarak atıfta bulunduğumu, tezimin kaynak gösterilen durumlar dışında özgün olduğunu, Tez Danışmanının **Prof. Dr. Aydın ULUCAN** danışmanlığında tarafımdan üretildiğini ve Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü Tez Yazım Yönergesine göre yazıldığını beyan ederim.



Volkan HASKILIÇ

Melis ve Yusuf'a ...

TEŞEKKÜR

Öncelikle bu zorlu sürecin tamamlanmasında aldığım en büyük manevi güç olan sevgili babam merhum Yusuf Haskılıç'ın aziz hatırası karşısında saygıyla eğiliyorum.

Her zaman yanımda olan ve benden desteğini hiçbir zaman esirgemeyen sevgili eşime ve çocuklarıma, çok kıymetli ve değerli anneme, kardeşime ve tüm aileme,

Akademik hayatımın her anında varlığı ve desteğiyle bana yardımcı olan ve yönlendiren, bu çalışmanın hazırlanmasında ve bilimsel çalışmanın nasıl yapılması gerektiği hakkında bana yol gösteren değerli hocam, danışmanım Prof. Dr. Aydın Ulucan'a,

Görüşleri ve katkılarıyla bu çalışmanın ortaya çıkmasında emeği geçen, bana her konuda destek veren ve yardımcı olan değerli hocam Doç. Dr. Kazım Barış Atıcı'ya,

Bana her konuda destek veren ve yardımcı olan değerli arkadaşım Dr. Bülent Çekiç'e,

Değerli katkılarından dolayı Prof. Dr. Fazıl Gökgöz, Prof. Dr. Cevriye Gencer ve Prof. Dr. Umur Tosun'a,

İsmini burada saymadığım ancak üzerimde her zaman emekleri olan Hacettepe Üniversitesi İşletme Bölümündeki tüm saygıdeğer hocalarıma teşekkürü bir borç bilirim.

ÖZET

HASKILIÇ, Volkan. *Kompozit Üretiminde Otoklav Yükleme Ve Çizelgeleme İçin Yeni Bir Model Önerisi: Savunma Sanayii Uygulaması*, Doktora Tezi, Ankara, 2019.

Kompozit malzemelerin yüksek mukavemet, hafiflik, tasarım esnekliği, kolay şekillendirme, rijitlik, korozyon ve kimyasal etkilere karşı dayanıklılık, ısı ve ateşe karşı dayanıklılık, titreşim sönümlenme özelliklerinden dolayı tüm üretim endüstrilerinde, özellikle de havacılık endüstrisinde kullanımı gittikçe artmaktadır. Havacılık endüstrisinde kullanılan kompozit malzemeler otoklav denilen ısı ve basınç fırınlarında kürlenme işlemine girerler. Oldukça yüksek ilk yatırım ve işletim maliyeti olan otoklav fırınlarının verimli kullanılması maliyetlerin kontrol altına alınması için önem arz etmektedir.

Bu tez çalışmasında otoklav yükleme ve çizelgeleme için bir karma tam sayı kuadratik kısıtlı programlama modeli geliştirilmiştir. Otoklav yükleme ve çizelgeleme problemlerinin ayrı ayrı ele alınması ile birlikte modellenmesi ile elde edilen sonuçlar karşılaştırılmış, kullanılan takım sayısının artırılmasının sonuca etkisi incelenmiş, yükleme ve çizelgelemenin birlikte ele alındığı model firmanın gerçek verileri üzerinden çalıştırılarak modelin pratikte kullanılabilirliği analiz edilmiştir.

Anahtar Sözcükler

Kutulama, Çizelgeleme, Otoklav Yükleme, Yığın Çizelgeleme, Karma Tamsayı Kuadratik Kısıtlı Programlama

ABSTRACT

HASKILIÇ, Volkan. *A New Model Proposal For Autoclave Loading and Scheduling: A Defense Industry Application*, Ph.D. Dissertation, Ankara, 2019.

Due to their high strength, lightness, design flexibility, easy shaping, rigidity, resistance to corrosion and chemical effects, heat and fire resistance, and vibration damping, composite materials are increasingly used in all manufacturing industries, especially in the aerospace industry. The composite materials used in the aviation industry are cured in heat and pressure furnaces called autoclaves. Because of their high initial investment and operating costs, efficient use of autoclave furnaces is important to control costs.

In this dissertation, a mixed integer linear programming model for autoclave loading and scheduling has been developed. The results of the modeling of autoclave loading and scheduling problems were compared and the effects of increasing the number of tools used on the result were examined and the practical usability of the model where loading and scheduling handled together was analyzed by running with the actual data of the firm.

Keywords

Bin Packing , Scheduling , Autoclave Loading , Batch Scheduling, Mixed Integer Quadratically Constrained Programming

İÇİNDEKİLER

KABUL VE ONAY	i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI	ii
ETİK BEYAN	iii
ADAMA	iv
TEŞEKKÜR	v
ÖZET	vi
ABSTRACT	vii
İÇİNDEKİLER	viii
TABLOLAR DİZİNİ	xi
ŞEKİLLER DİZİNİ	xiii
GİRİŞ	1
1. BÖLÜM: KOMPOZİT ÜRETİM	3
1.1. KOMPOZİT ÜRETİM TARİHÇESİ	6
1.2. KOMPOZİT ÜRETİM ENDÜSTRİSİ	8
1.3. KOMPOZİT ÜRETİM SÜRECİ	14
2. BÖLÜM: LİTERATÜR TARAMASI VE PROBLEM TANIMI	16
2.1. ÜRETİM PLANLAMA	16
2.2. ÇİZELGELEME PROBLEMİ	18
2.2.1. Çizelgeleme Ortamları.....	20
2.2.2. Çizelgeleme Problemi Kategorileri.....	20
2.2.3. Çizelgeleme Notasyonu.....	21
2.3. KUTULAMA PROBLEMİ	23
2.3.1. Bir Boyutlu Kutulama Problemi.....	24

2.3.2. İki Boyutlu Kutulama Problemi.....	25
2.3.3. Üç Boyutlu Kutulama Problemi.....	26
2.4. OTOKLAV YÜKLEME VE ÇİZELGELEME ÜZERİNE ÇALIŞMALAR.....	27
2.4.1. Lockheed Martin Otoklav Yükleme ve Çizelgeleme.....	27
2.4.2. Simülasyon Tabanlı Otoklav Yükleme ve Çizelgeleme Optimizasyonu.....	28
2.4.3. Tekrar Otoklava Girmesi Gereken Parçalar.....	28
2.4.4. Bir Kompozit Üretim Tesisinde Otoklav Yükleme ve Çizelgeleme Optimizasyonu.....	29
2.4.5. Otoklav Yüklemede Kompozit Parça Yerleşiminin Optimizasyonu.....	29
2.5. PROBLEM TANIMI.....	31
3. BÖLÜM: MATEMATİKSEL MODELLER VE UYGULAMA.....	33
3.1. OTOKLAV YÜKLEME MATEMATİKSEL MODELİ.....	35
3.2. OTOKLAV ÇİZELGELEME MATEMATİKSEL MODELİ.....	39
3.3. OTOKLAV YÜKLEME VE ÇİZELGELEME MATEMATİKSEL MODELİ.....	43
3.4. UYGULAMA.....	48
3.5. OTOKLAV SAYISININ ETKİSİ.....	52
3.5.1. Otoklav Yükleme ve Çizelgeleme Bağımsız Modeller.....	52
3.5.2. Otoklav Yükleme ve Çizelgeleme Tek Model.....	61
3.6. TAKIM SAYISININ ETKİSİ.....	67
3.6.1. Otoklav Yükleme ve Çizelgeleme Bağımsız Modeller.....	68
3.6.2. Otoklav Yükleme ve Çizelgeleme Tek Model.....	73
3.7. OTOKLAV YÜKLEME VE ÇİZELGELEMENİN BAĞIMSIZ VE TEK MODEL İLE ÇÖZÜMÜ.....	77
3.8. MODELİN FARKLI GÜNLER İÇİN ÇALIŞTIRILMASI.....	82

3.9. MODELİN PLANLAMA UFKUNUN GENİŞLETİLMESİ.....	87
SONUÇ	91
KAYNAKÇA.....	94
EK 1. Orijinallik Raporu.....	101
EK 2. Etik Kurul Muafiyet Formu.....	102
EK 3. Otoklav Yükleme Matematiksel Modeli Python Kaynak Kodu.....	103
EK 4. Otoklav Çizelgeleme Matematiksel Modeli Python Kaynak Kodu.....	113
EK 5. Otoklav Yükleme ve Çizelgeleme Matematiksel Modeli Python Kaynak Kodu.....	124
EK 6. Bağımsız Model Gantt Şemaları.....	139
EK 7. Tek Model Gantt Şemaları.....	142
EK 8. Farklı Günler için Gantt Şemaları.....	145
EK 9. Geniş Planlama Ufku için Gantt Şeması.....	152

TABLolar DİZİNİ

Tablo 1 Dünya’da Kompozit Üretimini Değer Olarak Sektörlere Dağılımı....	11
Tablo 2 Çizelgeleme Ortamı Notasyonu.....	21
Tablo 3 İş Özellikleri Notasyonu.....	22
Tablo 4 Amaç Fonksiyonu Notasyonu.....	22
Tablo 5 Otoklav Yükleme ve Çizelgeleme Üzerine Yapılan Çalışmalar.....	30
Tablo 6 Otoklav Kapasiteleri.....	49
Tablo 7 Bağımsız Modeller Alan Kullanımı ve Talep Karşılama.....	52
Tablo 8 Bağımsız Modeller Yükleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı).....	58
Tablo 9 Bağımsız Modeller Çizelgeleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı).....	60
Tablo 10 Tek Model Alan Kullanımı ve Talep Karşılama.....	62
Tablo 11 Tek Model - Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı)..	66
Tablo 12 Bağımsız Modeller Alan Kullanımı ve Talep Karşılama.....	68
Tablo 13 Bağımsız Modeller – Yükleme - Model ve Çözüm Sayısal Değerleri.....	72
Tablo 14 Bağımsız Modeller – Çizelgeleme - Model ve Çözüm Sayısal Değerleri.....	73
Tablo 15 Tek Model Alan Kullanımı ve Talep Karşılama.....	73
Tablo 16 Tek Model - Model ve Çözüm Sayısal Değerleri.....	76
Tablo 17 Bağımsız ve Tek Model - Talep ve Alan Kullanım Karşılaştırması...	79

Tablo 18 Bağımsız ve Tek Model Amaç Fonksiyonu Değeri ve Çözüm Süresi.....	80
Tablo 19 Farklı Günler Çalışma Alan Kullanımı ve Talep Karşılama.....	83
Tablo 20 Farklı Günler Çalışma - Model ve Çözüm Sayısal Değerleri.....	87
Tablo 21 Geniş Planlama Ufku Alan Kullanımı ve Talep Karşılama.....	88
Tablo 22 Geniş Planlama Ufku - Model ve Çözüm Sayısal Değerleri.....	90

ŞEKİLLER DİZİNİ

Şekil 1 Otoklav.....	3
Şekil 2 Takım.....	4
Şekil 3 Serim İşlemi.....	6
Şekil 4 Dünya’da Karbon Elyaf Talebinin Yıllara Göre Değişimi.....	9
Şekil 5 Boeing Firmasının Bazı Uçak Modellerinde Malzeme Dağılımı.....	10
Şekil 6 Airbus Firmasının Bazı Uçak Modellerinde Malzeme Dağılımı.....	11
Şekil 7 Anka İnsansız Hava Aracı Malzeme Kullanım Oranları.....	13
Şekil 8 Thermocouple ve Vacuumport.....	15
Şekil 9 Üretim Planlama.....	17
Şekil 10 Örnek Gantt Şeması.....	19
Şekil 11 Bir Boyutlu Kutulama Örnek Gösterim.....	25
Şekil 12 İki Boyutlu Kutulama Örnek Gösterim.....	26
Şekil 13 Üç Boyutlu Kutulama Örnek Gösterim.....	26
Şekil 14 Çalışma Planı.....	50
Şekil 15 Bağımsız Modeller - Alan Kullanımı (Otoklav Sayısı).....	54
Şekil 16 Bağımsız Modeller – Talep Karşılama (Otoklav Sayısı).....	55
Şekil 17 Bağımsız Modeller – Yükleme Amaç Fonksiyonu (Otoklav Sayısı).....	56
Şekil 18 Bağımsız Modeller – Çizelgeleme Amaç Fonksiyonu (Otoklav Sayısı).....	57
Şekil 19 Bağımsız Modeller – Otoklav Kullanım Durumu.....	57

Şekil 20	Tek Model - Alan Kullanımı (Otoklav Sayısı).....	63
Şekil 21	Tek Model – Talep Karşılama (Otoklav Sayısı).....	64
Şekil 22	Tek Model – Amaç Fonksiyonu (Otoklav Sayısı).....	65
Şekil 23	Bağımsız Modeller – Otoklav Kullanım Durumu.....	65
Şekil 24	Bağımsız Modeller - Alan Kullanımı.....	69
Şekil 25	Bağımsız Modeller – Talep Karşılama.....	70
Şekil 26	Bağımsız Modeller – Yükleme Amaç Fonksiyonu.....	71
Şekil 27	Bağımsız Modeller – Çizelgeleme Amaç Fonksiyonu.....	72
Şekil 28	Tek Model - Alan Kullanımı.....	74
Şekil 29	Tek Model – Talep Karşılama.....	75
Şekil 30	Tek Model – Amaç Fonksiyonu.....	76
Şekil 31	Bağımsız ve Tek Model – Kullanılan Alan Karşılaştırması.....	78
Şekil 32	Bağımsız ve Tek Model – Talep Karşılama Karşılaştırması.....	79
Şekil 33	Bağımsız ve Tek Model – Amaç Fonksiyonu Karşılaştırması.....	81
Şekil 34	Bağımsız ve Tek Model – Süre Karşılaştırması.....	82
Şekil 35	Farklı Günler Çalışma – Alan Kullanımı.....	84
Şekil 36	Farklı Günler Çalışma – Talep Karşılama.....	85
Şekil 37	Farklı Günler Çalışma – Amaç Fonksiyon Değeri.....	86
Şekil 38	Geniş Planlama Ufku – Alan Kullanımı.....	89
Şekil 39	Geniş Planlama Ufku – Talep Karşılama.....	89

GİRİŞ

Kompozit malzemeler iki veya daha fazla malzemenin bir araya gelmesiyle oluşurlar ve kendilerini oluşturmak için bir araya getirilen malzemelerin en iyi özelliklerini taşırlar veya kendilerini oluşturan malzemelerde bulunmayan yeni bir özellik kazanabilirler. Yüksek mukavemet ve hafiflik istenen durumlarda genellikle metaller yerine kompozit malzemeler kullanılmaktadır. Kompozit malzemeler sadece yüksek mukavemet ve hafiflik istenen durumlarda değil, tasarım esnekliği, kolay şekillendirme, rijitlik, korozyon ve kimyasal etkilere karşı dayanıklılık, ısı ve ateşe karşı dayanıklılık, titreşim sönümlenme istenen durumlarda da kullanılır. Bir kompozit malzeme oluştururken tüm bu özellikleri bir arada sağlamak mümkün değildir, kompozit malzeme hangi alanda kullanılacaksa o alanda daha iyi olması gereken özellikleri artırılır.

Bu tez çalışmasında otoklav yöntemi ile kompozit malzeme üretimi üzerinde durulacaktır. Otoklav yöntemi ile kompozit malzeme üretiminde, bir reçine yardımı ile bir araya getirilen katmanlar otoklav denilen yüksek ısı ve basınç üreten fırınlarda birleştirilir. Katmanların reçine ile birleşebilmesi için belirli bir süre hem ısı hem de basınç uygulanması gerekmektedir, bu işleme kürlenme denilmektedir. Büyük boyutları, 800.000 – 1.200.000 ABD doları aralığındaki ilk yatırım maliyetleri ve 500.000 ABD doları civarında yıllık enerji sarfiyatı otoklavları bir fabrikadaki en pahalı ekipmanlardan biri yapmaktadır.

Uygulamanın gerçekleştirildiği firma savunma sanayiinde üretim yapan büyük ölçekli bir şirkettir. Birçok üretim alanının yanı sıra bu tez çalışmasının konusu olan kompozit üretimi 3 temiz oda, 7 otoklav ile yürütülmektedir. Kompozit ürünlerin tasarımı ve imalatı alanında, dünya havacılık endüstrisinde lider şirketlerden biri olmayı hedefleyen firma, kompozit ürün imalat alanında geniş bir kabiliyet ağına sahiptir. Aynı zamanda ileri teknolojilere yatırım yaparak bu alanda imalat kabiliyetlerini sürekli iyileştirmektedir.

Bu çalışma ile havacılık endüstrisinde kompozit parçaların otoklava yüklenmesi ve çizelgelenmesi için bir matematiksel model geliştirilecektir. Problem yükleme (kutulama) ve çizelgeleme olarak 2 kısımdan oluşmaktadır. Her ikisi de NP-Zor

olan bu problemlerin çözümü için tek bir karma tam sayı kuadratik kısıtlı programlama modeli kurulacaktır. Literatür taraması bu problemin daha önce bu kadar kapsamlı olarak ele alınmadığını göstermiştir.

Bu tez çalışması üç bölümden oluşmaktadır. Birinci bölümde kompozit malzemeler tanımlanacak, kompozit üretim süreci ile ilgili bilgi verilecektir.

İkinci bölümde bu tez konusu ile ilgili literatür incelenecek ve problem tanımlanacaktır. Tezin temelini oluşturan üretim planlama, çizelgeleme problemi ve kutulama problemi ile ilgili tanımlar yapıldıktan sonra otoklav yükleme ve çizelgeleme ile ilgili bugüne kadar yapılmış çalışmalar hakkında bilgi verilecektir. Daha sonra otoklav yükleme ve çizelgelemenin neden önemli olduğu ve firmanın bu süreçte karşılaştığı problemler ortaya konulacaktır.

Üçüncü bölümde otoklav yükleme ve çizelgeleme problemini çözmek üzere bu tez çalışması kapsamında oluşturulan karma tamsayı kuadratik kısıtlı programlama matematiksel modelleri açıklanacaktır. Bu modellerin çalıştırılması ile elde edilen sonuçlar incelenecek, firmanın problemine uygulanarak, çözümü sağladığı örneklerle gösterilecektir.

Son bölümde ise sonuç ve ileride konu ile ilgili yapılacak çalışmalar için öneriler yer almaktadır.

1. BÖLÜM

KOMPOZİT ÜRETİM

Otoklav fırınları iç basıncı ve sıcaklığı ayarlanabilen büyük basınç kaplarıdır. Şekil-1’de örnek olarak firmadaki otoklavlardan biri görülebilir. Her kompozit parça kürlenmek için özel bir sıcaklık ve basınç profiline ihtiyaç duyar, buna reçete adı verilir. Benzer uygulamalar için kullanılan kompozit parçalar genellikle aynı reçeteye sahiptir ve otoklav kısıtlarını karşıladıkları sürece beraber kürlenirler.



Şekil - 1 Otoklav

Kompozit parça üretimi süreci kit kesme alanında başlar, burada farklı katmanları oluşturacak malzemeler rulolardan kesilerek hazırlanır. Tek bir parçayı oluşturan malzeme katmanlarının hepsi kit olarak paketlenir. Kitler daha sonra serim alanına gönderilir. İlgili parçanın üretimi için özel eğitilmiş deneyimli personel takım üzerinde katmanların serim işlemini gerçekleştirir. Bu takımlar tüm kürlenme

işlemi süresince parça ile beraber hareket eder. Şekil – 2 ‘de örnek bir takım görülebilir.



Şekil – 2 Takım

Otoklavı doldurmaya yetecek kadar parçanın serim işlemi tamamlandığında parçalar otoklava yüklenir, aynı reçete ile otoklavda kürlenirler ve otoklavdan çıktıktan sonra takım çıkarma alanına gönderilir ve takımlardan ayrılırlar. Takımlar temizlenir ve bir sonraki parçanın serimi için hazır duruma getirilirler. Bu şekilde üretim süreci parçaların işlemleri tamamlandıkça sonraki istasyona gönderildiği itme sistemine uymaktadır. Bu üretim sistemi süreçlerdeki verimsizliklerin temel kaynağını oluşturmaktadır. Örneğin; eğer parçalar otoklava

düzensiz olarak gelirse birkaç farklı grupta kürlenebilirler. Her kürlenme otoklavı ısıtma ve otoklavı kapasitesinin altında yükleme gibi maliyetler içermektedir. Otoklavı ısıtma maliyeti çoğunlukla otoklava yüklenen parça sayısından bağımsızdır. Bu nedenle kür sayısını azaltmak ve her kürde kullanılan kapasiteyi arttırmak toplam maliyeti azaltacak ve toplam kompozit üretim kapasitesini arttıracaktır.

Kompozit üretimi için birçok yöntem mevcuttur. Bu yöntemler arasında 3 temel fark vardır.

Birinci fark parçayı oluşturan fiber malzemenin serilme yöntemidir. Doğrusal veya geçişmeli olarak serim yapılabilir. Doğrusal yapı daha önce reçine emdirilmiş fiber malzemelerin üst üste dizilmesi şeklindedir. Geçişmeli yapı dokuma şeklinde yapılır ve 2 veya 3 boyutlu parçalar üretilebilir. Firma sadece doğrusal yapıda üretim yaptığı için bu tez çalışmasında doğrusal yöntem incelenecektir.

İkinci fark fiber malzemeye reçinenin emdirilmesi yöntemidir. Reçine hemen serim öncesi uygulanabileceği gibi hazır reçine uygulanmış fiber malzemeler de kullanılabilir. Daha önce reçine uygulanmış fiber malzemelere "prepreg" adı verilmektedir, prepreg malzemeler soğutucuda saklanmak zorundadır. Soğutucu dışındaki malzeme ömrü kısadır. Firma sadece prepreg malzeme ile kompozit üretimi yapmaktadır.

Üçüncü fark parçanın termoset veya termoplastik olmasıdır. Termoset parçaların sertleşmesi için gerekli olan kimyasal tepkime ısıya ihtiyaç duyar. Diğer taraftan termoplastik parçaları sertleştirmek için soğutmak gerekmektedir. Firmanın ürettiği tüm kompozit parçalar termoset parçalardır ve sertleşmesi için gerekli kimyasal tepkime için lazım olan ısı ve basınç otoklavlar ile sağlanır.

Fiber malzemeler, takımlar üzerine elle serilir. Tabaka malzemenin özellikleri izotropik olmadığı için katmanlar malzemenin özellikleri göz önüne alınarak farklı yönlerde serilir. Parçanın en önemli yüzü takım üzerine gelecek şekilde yerleştirilir, diğeri vakum torbası ile kaplanır. Serim işlemi tamamlanan parça ve takım kürlenmek üzere otoklava yerleştirilir.



Şekil – 3 Serim İşlemi

1.1. KOMPOZİT ÜRETİM TARİHÇESİ

Kompozitlerin ilk kullanımı milattan önce 3400 yılına kadar uzanır. Mısırlılar ve Mezopotamyalılar binalarının dayanıklı ve sağlam olması için çamur ve samanı karıştırarak kullanmışlardır. Saman daha sonraları da toprak kapları ve kayıkları güçlendirmek için kullanılmaya devam etmiştir.

Milattan sonra 1200 yıllarında Moğollar ilk kompozit yayı yapmışlardır. Tahta ve kemik hayvansal bir tutkal ile birleştirilmiş, sıkıştırılan malzemeler ağaç kabuğu ile kaplanarak yaylar yapılmıştır. Moğolların kullandıkları bu yaylar zamanının en güçlü ve isabetli olanlarıydı. Bu kompozit yaylar Cengiz Han'ın askeri üstünlüğünü korumasını sağlamıştır.

Kompozitlerin modern çağı, plastiğin icadıyla başlar. 1900'lerin başında, vinil, polistiren, fenolik ve polyester gibi plastikler geliştirilmiştir. Bu yeni sentetik malzemeler daha önce kompozit malzeme yapımı için kullanılan ve doğadan elde

edilen reçinelere göre çok üstün özelliklere sahipti fakat bazı yapısal uygulamalar için plastikler de tek başına gerekli sağlamlığı sağlayamıyordu.

1935 yılında ilk cam elyaf Owens Corning tarafından tanıtıldı. Plastik bir polimer ile birleştirilen cam elyaf çok güçlü ve aynı zamanda da çok hafif bir yapı oluştuyordu. Bu çalışmalar Elyaf Takviyeli Polimer endüstrisinin başlangıcını oluşturdu.

Kompozit malzemelerdeki en önemli ilerlemeler 2. Dünya Savaşı sırasında doğan ihtiyaçlar sayesinde oldu. Moğolların savaş için kompozit yayları kullanması gibi 2. Dünya Savaşı da Elyaf Takviyeli Polimer endüstrisini laboratuvar ortamından çıkarıp üretim ortamına geçirmiştir.

Daha hafif askeri uçaklar üretmek için kompozit malzemeler vazgeçilmezdi fakat kısa sürede cam elyaf malzemelerin hafifliği ve sağlamlığı dışındaki özellikleri de keşfedildi. Örneğin cam elyaf kompozitler radyo dalgalarını yansıtmıyordu, bu sayede radara yakalanmayan uçaklar geliştirildi.

2. Dünya Savaşının sona ermesiyle kompozit malzemelere olan talep de azalmaya başladı. Kompozit malzeme üreticileri kendilerine yeni pazarlar bulmak için arayış içerisine girdiler. Tekneler kompozit malzemelerin kullanıldığı ilk ticari ürünlerdendi. 1946 yılında ilk kompozit ticari tekne gövdesi tanıtıldı.

Bu sıralarda kompozitin babası olarak tanımlanan Brandt Goldsworthy birçok yeni üretim süreci ve ürün geliştirdi. Bunlardan biri olan ilk fiberglas sörf tahtası sörf sporunda bir devrim yarattı.

Güvenilir sağlam fiberglas güçlendirilmiş ürünlerin üretilebilmesini sağlayan pultrizyon yöntemi de Goldsworthy tarafından geliştirilmiştir. Günümüzde üretilen katlanır merdivenler, borular, miller, zırhlar, tren zeminleri ve tıbbi araçlar bu yöntemle üretilen ürünlere örnek verilebilir.

1970'lerde kompozit endüstrisi olgunlaşmaya başladı. Daha iyi plastik reçineler ve güçlendirici elyaf geliştirildi. DuPont, yüksek gerilme mukavemeti, yüksek yoğunluğu ve hafifliği nedeniyle vücut zırhında tercih edilen ürün haline gelen

Kevlar olarak bilinen bir aramid lifi geliştirmiştir. Bu süre zarfında karbon elyaf da geliştirilmiştir. Karbon elyaf giderek çeliğin yerine geçmeye başlamıştır.

Kompozit endüstrisi sürekli bir gelişim halindedir, son zamanlarda yenilenebilir enerji ve hava araçları üretiminde kompozit malzemelerin kullanımı gittikçe daha çok olmaktadır ve ileri kompozit malzemelere ihtiyaç sürekli artmaktadır.

Türkiye’de kompozit malzemeler polyester su depoları ve Anadol marka otomobil kaportası ile başlamıştır. Günümüzde ileri düzey kompozit üretiminde dünyada öncü duruma geçecek çalışmalara önem verilmektedir.

1.2. KOMPOZİT ÜRETİM ENDÜSTRİSİ

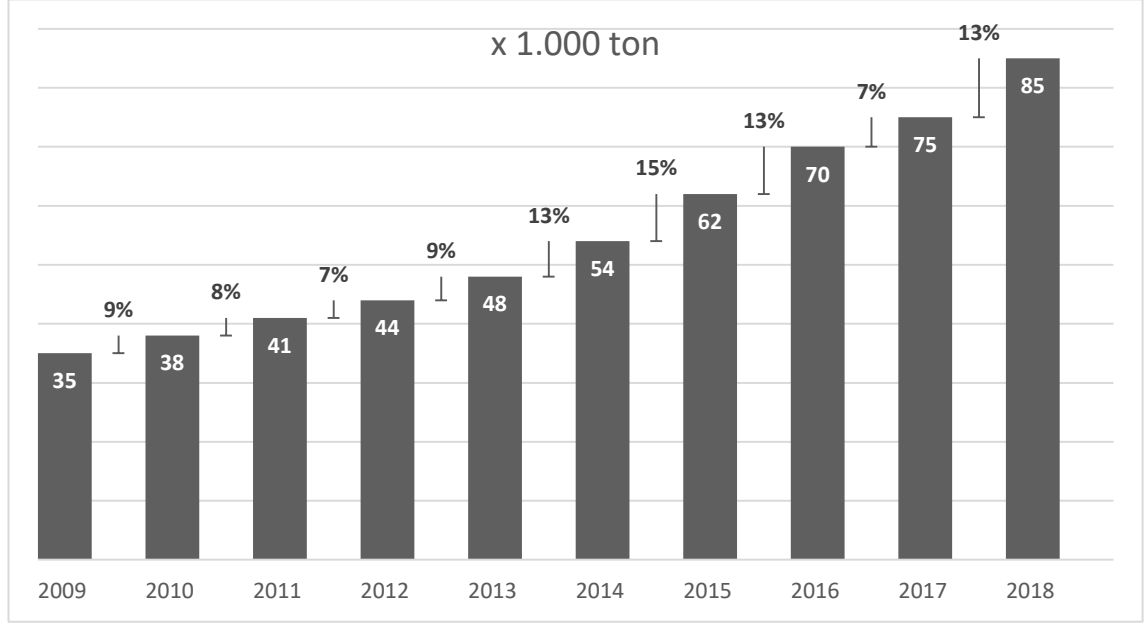
Kompozit üretim endüstrisinde kullanılan elyafların büyük çoğunluğu cam elyaftır. Cam elyaf, ağır metallerin yerine en çok kullanılan ve en eski güçlendirilmiş malzemedir. Cam elyafların kullanılmadığı en belirgin sektör havacılık ve uzay sektörüdür.

Havacılık ve uzay endüstrisi yakıt verimliliğini artırma ve emisyonu azaltma hedefleri doğrultusunda uçak ağırlıklarını azaltmaya çalışmaktadır, bu doğrultuda kompozit malzeme kullanımı bu endüstri için önemini arttırarak devam ettirmektedir. Buna rağmen, kompozit malzeme kullanan diğer endüstrilerle karşılaştırıldığında, havacılık ve uzay endüstrisinde yeni malzeme kabul onay süreci önemli derecede uzun ve zordur ve fırsat maliyeti yüksektir.

Kompozit malzeme pazarında, havacılık ve uzay endüstrisinin uçak ağırlıklarını hafifletmeye yönelik çalışmalarından en çok faydalanan karbon elyafı ile güçlendirilmiş polimerler olmuştur. Birincil yük taşıyan yapısalarda alüminyum ve çeliğin yerine karbon elyaf kullanmanın yüksek potansiyeli nedeniyle karbon elyaf kompozitlerin pazar payının önümüzdeki yıllarda daha da artması muhtemeldir. (Karthikeyan, 2017)

Bu tez çalışmasının yapıldığı firmada da sadece karbon elyaf kompozit üretimi yapılmaktadır.

Dünyada karbon elyaf pazarı son 10 yılda, yılda ortalama %10 büyüme göstererek toplamda %143 büyümüş ve yıllık 85 bin ton kullanıma ulaşmıştır. (Şekil - 4)



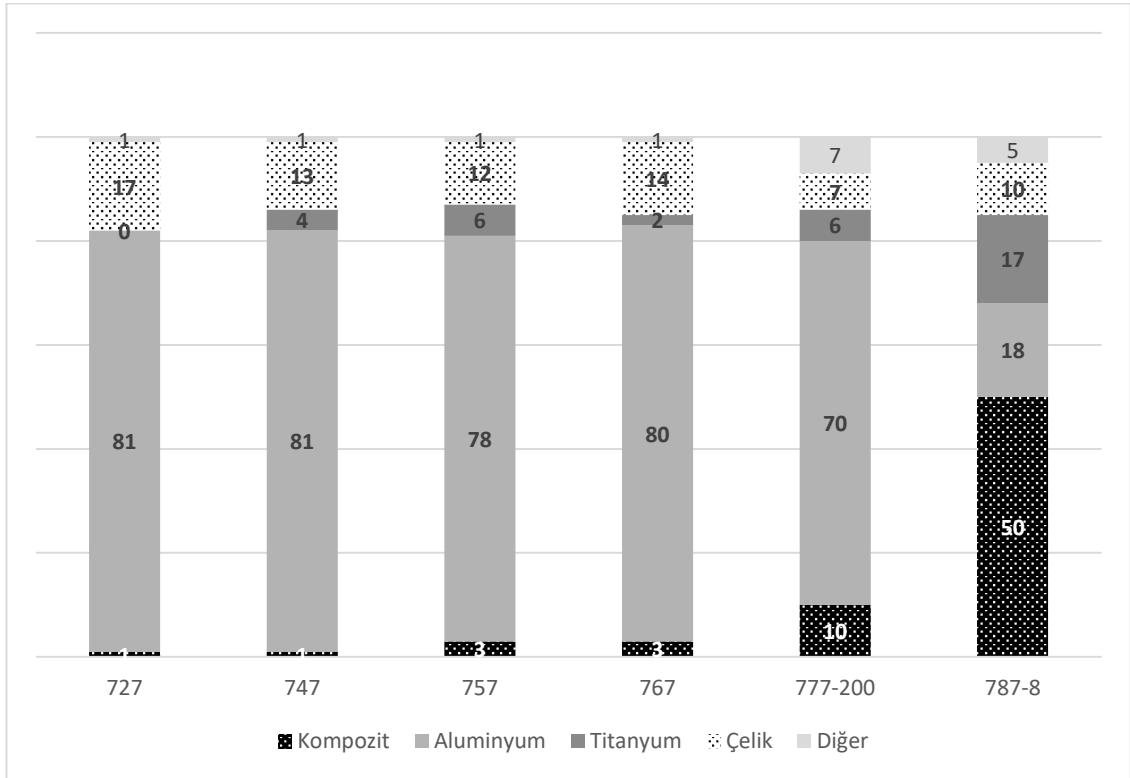
Şekil – 4 Dünyada Karbon Elyaf Talebinin Yıllara Göre Değişimi

Karbon elyaf kompozit malzeme talebinin önümüzdeki yıllarda da aynı oranda artarak devam etmesi öngörülmektedir. Pazarda her yıl en az %8 büyüme beklenmektedir, beklentilerin gerçekleşmesi durumunda 2025 yılında pazarın büyüklüğünün en az 145 bin ton olması beklenmektedir.

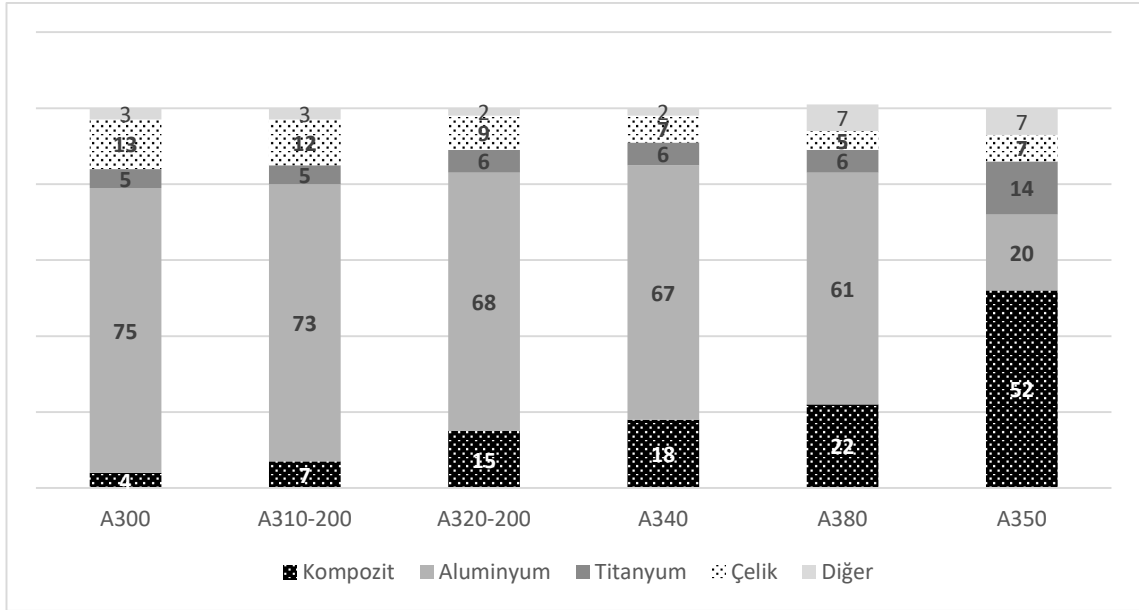
Pazardaki segmentler, büyüme için önemli bir potansiyel göstermektedir. Havacılık, rüzgar türbini kanatları, spor malzemeleri ve kalıplar en iyi bilinen segmentlerdir ve gittikçe daha fazla karbon elyaf kullanan ürünler geliştirildiği için bu sektörün büyümeye devam edeceği öngörülmektedir. Basınç kapları (sıkıştırılmış doğal gaz, sıvılaştırılmış petrol gazı, hidrojen vb. için), inşaat ve altyapı uygulamaları ise daha yeni kullanım alanlarıdır ve üretim yöntemlerinin gelişmesi ile önemli bir büyüme potansiyeline sahiptir. Gelişim ve uyarlama bugünkü hızıyla devam ederse gelecekteki pazar yapısı günümüzdekinden oldukça farklı bir görünüme sahip olacaktır. (Pichler, 2019)

Tüm dünyada üniversitelerin ve araştırma enstitülerinin güçlü desteği ile havacılık endüstrisi metal güdümlü bir üretimden kompozit ağırlıklı yeni bir standarda geçmiştir.

Geçtiğimiz 15 yılda, Boeing 787 Dreamliner ve Airbus A350 XWB gibi iki önemli uçak programında kompozit malzeme kullanımı önemli derecede artmıştır. (Şekil - 5, Şekil - 6)



Şekil – 5 Boeing Firmasının Bazı Uçak Modellerinde Malzeme Dağılımı



Şekil – 6 Airbus Firmasının Bazı Uçak Modellerinde Malzeme Dağılımı

Dünyada kompozit üretimin değer olarak sektörler dağılımına bakıldığında Havacılık ve Uzay sektörünün %23 ile en yüksek paya sahip olduğu görülmektedir. Sektörlerin kullanım değerine göre sıralanması Tablo – 1’de gösterilmektedir. (Bulut, 2004)

Sektör	Kullanım Oranı
Havacılık ve Uzay	23%
Otomotiv ve Taşımacılık	22%
Yapı ve İnşaat	14%
Rüzgar Enerjisi	12%
Elektrik Elektronik	12%
Tüketici Malları	9%
Denizcilik	5%
Boru ve Tank	2%

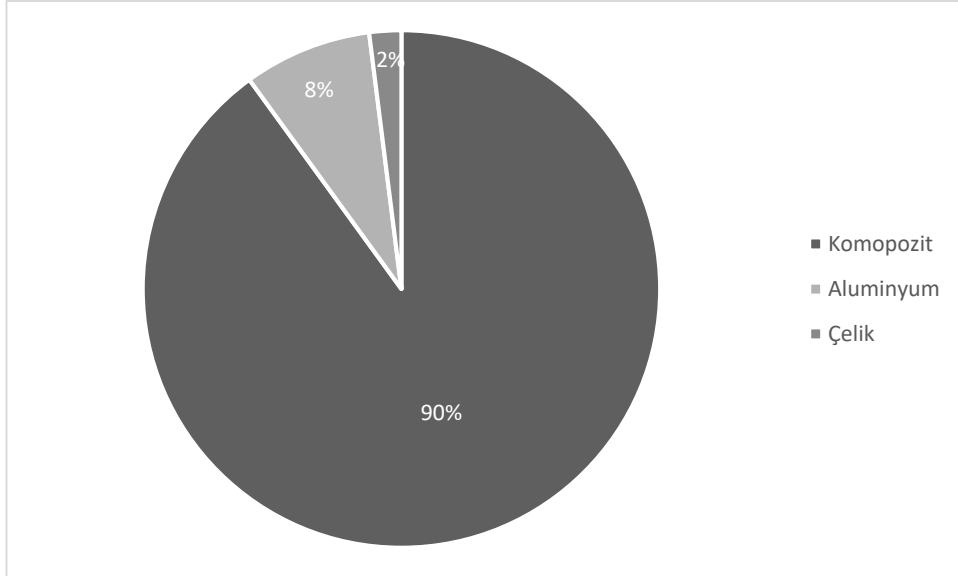
Tablo – 1 Dünya’da Kompozit Üretiminin Değer Olarak Sektörlere Dağılımı

Türkiye’de kompozit sektörü ekonomideki küresel gelişmeler doğrultusunda diğer sektörlerde de olduğu gibi hızlı bir gelişme göstermiştir. Türkiye kompozit malzeme pazarı (cam elyaf, karbon elyaf vb. tüm kompozit malzemeler) 1 milyar Avro ve 200.000 ton hacme ulaşmıştır.

Türkiye’deki kompozit sektörü krizden önceki senelerde %8 - %12 arası büyümüş, krizden sonraki senelerde ise %9 - %10 arası büyüme göstererek Avrupa ve hatta Dünya’daki kompozit sektörünün büyüme oranından daha yüksek bir büyüme göstermiştir. (Bulut, 2014)

Türkiye’de kompozit sektörü Türk Havacılık Uzay Sanayii liderliğinde büyümektedir. Türk Havacılık Uzay Sanayii, Airbus, Boeing, Lockheed Martin gibi uluslararası firmalarla ortak üretim çalışmaları yürütürken aynı zamanda ATAK Helikopteri, Hürkuş Eğitim Uçağı, Anka İnsansız Hava Aracı, Göktürk Uydusu, Hürjet Jet Eğitim Uçağı ve Milli Muharip Uçak gibi özgün ürünlerini geliştirmekte ve üretmektedir. 2030’lu yıllardan itibaren kademeli olarak devreden çıkartılması düşünülen F-16 uçaklarının yerini alacak olan Milli Muharip Uçak projesinde ihtiyaç duyulacak kompozit malzemelerin üretimi için Türkiye’nin en büyük, Dünya’nın tek çatı altındaki dördüncü büyük kompozit üretim tesisinin inşa çalışmaları da Türk Havacılık ve Uzay Sanayii bünyesinde başlamıştır.

Türk Havacılık ve Uzay Sanayii özgün ürünlerinden Anka İnsansız Hava Aracı’nın %90’ı karbon elyaf kompozit malzemedir (Şekil - 7)



Şekil – 7 Anka İnsansız Hava Aracı Malzeme Kullanım Oranları

Savunma Sanayiinde kompozit üretim konusunda Türkiye’de faaliyet gösteren diğer firmalar aşağıda listelenmiştir.

- i. Yüksel Kompozit Teknolojileri A.Ş.
- ii. Epsilon Havacılık Uzay ve Savunma Sanayi
- iii. C.E.S İleri Kompozit ve Savunma Teknolojileri San. ve Tic. A.Ş.
- iv. Global Teknik Elektronik Yazılım Müh. Havacılık San. ve Tic. A.Ş.
- v. Barış Elektrik Endüstrisi A.Ş.
- vi. NuroI Teknoloji A.Ş.
- vii. Odak Kompozit Teknolojileri A.Ş.
- viii. Atard Savunma ve Havacılık Sanayi İleri Teknoloji Uygulamaları Araştırma ve Geliştirme A.Ş.
- ix. Karma Grup Kompozit ve Teknoloji Makine Sanayi ve Ticaret A.Ş.

1.3. KOMPOZİT ÜRETİM SÜRECİ

Otoklav çizelgeleme firmanın kompozit işlemleri üretim planlama ve kontrol birimi tarafından gerçekleştirilmektedir.

Otoklavlarda kürlenme işleminden önce her biri farklı boyutta olan parçalar kendine özel takımlarla birleştirilip kürlenmek üzere hazır hale getirilmektedir. Temizlenmiş takımların üzerine, ihtiyaca göre önceden hazırlanmış, en iyi özellikleri sağlayacak kompozit malzeme reçine ile yapıştırılmaktadır. Bu parçalar plastik bir malzeme ile torbalanıp vakumlanarak kürlenme işlemi için otoklav yerleşimine hazır hale getirilmekte ve kompozit malzemenin kullanım ömrü dolmadan kürlenmektedir. Bu işlemler temiz odada yapılmaktadır.

Her bir parça serilen kompozit malzemenin ısı ve basınç ihtiyacına göre gruplara yerleştirilmiştir. Parça üzerinde serilen, aynı basınç ve ısıya ihtiyaç duyan malzemeler aynı kür grubuna aittir. Bir otoklava bir seferde yalnızca aynı kür grubuna ait parçalar atanabilmektedir. Ayrıca her kür grubu için belirli bir otoklavda kalma süresi mevcuttur.

Süreçte darboğaz olan kürlenme işlemi, temiz odada hazırlanmış malzemelere, ısı ve basınç uygulanan otoklav tipi fırınlarda gerçekleşmektedir. Tesis içerisinde farklı boyutlarda 7 adet otoklav bulunmaktadır. Mevcut sistemde kürlenme işlemi için otoklavlar 8'er saatten oluşan 3 vardiya şeklinde çalışmaktadırlar. Her bir otoklavın kapasitesi farklıdır ve bu kapasiteler taban alanları göz önüne alınarak hesaplanmaktadır.

Kürleme işlemi sırasında parçanın ısısının kontrol edilebilmesi için otoklavlarda "thermocouple" olarak adlandırılan ısı ölçerler ve basınç uygulaması için de "vacuum port" olarak adlandırılan aparatlar bağlanmaktadır (Şekil – 8). Bu aparatlar her bir otoklavda belirli sayıda bulunmaktadır ve her parçaya, parça özelliğine göre belirli sayıda aparat bağlanması gerekmektedir. Yerleşim tamamlandıktan sonra her bir parçaya uygulanan basıncın ve ısının kontrol edilmesini sağlayan bu aparatlar bir ucu otoklavda diğer ucu parçada olacak şekilde bağlanmaktadır.



Şekil – 8 Thermocouple ve Vacuumport

2. BÖLÜM

LİTERATÜR TARAMASI VE PROBLEM TANIMI

Bu bölümde üretim planlama, çizelgeleme problemi ve kutulama problemi hakkında genel bilgiler verilecek, kompozit üretim çizelgeleme ile ilgili daha önce yapılan çalışmalar özetlenecek ve firmanın karşılaştığı problem tanımlanacaktır.

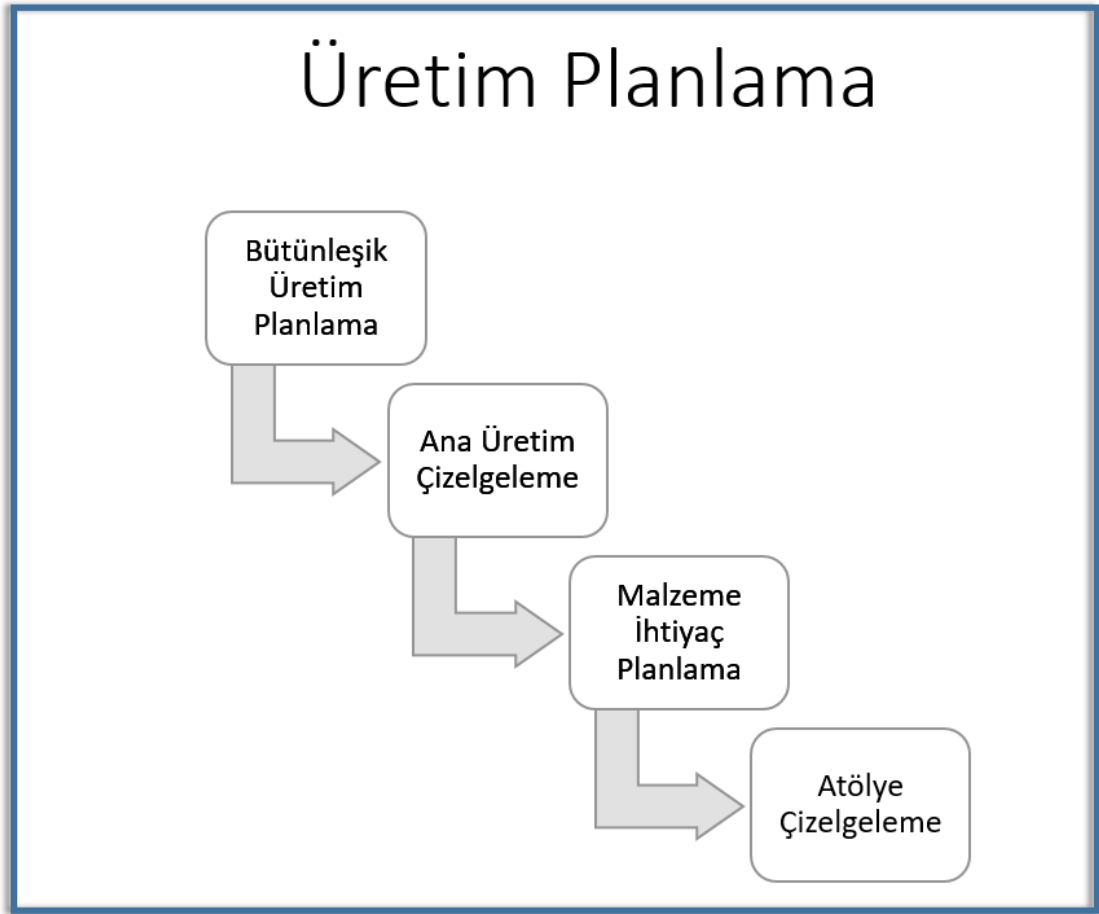
2.1. ÜRETİM PLANLAMA

Üretim firmalarının var olma amaçları müşterinin ihtiyacı olan ürünleri, müşterinin ödemek istediği fiyat ile üretmesi ve tedarik etmesidir. Bir firma bu amacı yerine getirirken doğal olarak kar etme amacını güder. Tekel konumunda çok az firma vardır, firmaların büyük çoğunluğu açık pazarda rekabet içerisinde. Ürünün fiyatı da doğal olarak rekabet ortamında belirlenir. Firmaların karını arttırmasının tek yolu üretim ve dağıtım maliyetlerini azaltmasıdır. Maliyet azaltmak için firmayı verimli bir şekilde işletmek gerekir.

Üretim planlama, verimi ve etkinliği arttırmak için en önemli konuların başında gelmektedir.

Üretim planlama, müşterinin ihtiyacını karşılamak üzere, fazla stok ve geri sipariş tutmadan, neyin, ne zaman ve nasıl üretileceğini belirler (Dileep, 2008).

Üretim planlama, son ürünü oluşturmak için beraber kullanılan farklı proseslerin, envanterin ve kaynakların koordinasyonu için gereklidir. Bu kaynakların uygun koordinasyonu maliyetlerin azaltılmasını ve müşteri ihtiyaçlarını daha iyi karşılayan üretim süreçlerinin oluşmasını sağlar. Yönetimsel aktiviteler 3 alana ayrılır: stratejik planlama, taktik planlama ve operasyonel kontrol (Anthony, 1965). Bu alanlar üretim yönetimine uygulandığında stratejik planlama çok uzun dönemi, genellikle 2 yıl ve üstünü kapsar ve genellikle tezgah ve iş gücü yatırım planlarının oluşturulmasında kullanılır. Taktik seviye kaynakların verimli kullanımını kapsar ve 6 – 24 ay arasında bir planlama ufkuna sahiptir. Operasyonel kontrol ise çok kısa zamanlı planlama ufkuna yöneliktir ve çok detaylıdır, çizelgeleme buna bir örnek olarak görülebilir.



Şekil – 9 Üretim Planlama

Üretim planlama; bütünleşik planlama, ana üretim çizelgesi, malzeme ihtiyaç planlama ve atölye çizelgelemesini kapsar. Bütünleşik üretim planlama orta vadede toplam ne üretileceğini ve bu üretimi gerçekleştirebilmek için gerekli toplam kaynak malzeme miktarını ortaya koyar. Bütünleşik üretim planının çıktısı ana üretim çizelgesinin girdisi olur. Ana üretim çizelgesi ne zaman ne üretileceğini belirler, son ürün bazlıdır, üretim için gerekli malzeme kaynağını göstermez. Ana üretim çizelgesi, envanter seviyeleri, sipariş politikaları, temin süreleri, ürün ağaçları, kullanım miktarları vb. bilgiler kullanılarak malzeme ihtiyaç planları çıkarılır. Malzeme ihtiyaç planı, ana üretim çizelgesinde belirlenen son ürünleri planlanan tarihte üretebilmek için ne zaman, hangi malzemeye, ne kadar ihtiyaç olacağını (satın alma veya üretim) gösterir. Malzeme ihtiyaç planı ne zaman ne üretileceğini gösterir fakat atölyedeki tezgahlarda hangi işlemin ne zaman, hangi

sıra ile yapılacağını vermez. Atölye çizelgeleme malzeme ihtiyaç planı çıktılarını kullanarak işlerin çizelgesini belirler.

Bu tez çalışmasında üretim planlama konularından biri olan çizelgeleme kullanılacaktır.

Üretim planlama yöntemleri itme ve çekme sistemler olarak 2 gruba ayrılabilir. İtme sisteminde, üretim planlama en baştan tüm seviyeler için yapılır, çekme sisteminde ise ürünler sadece bir sonraki seviye talep ettiğinde üretilir. Her iki sistemin de hem avantajları hem de dezavantajları mevcuttur.

İtme sistemler talebi tahmin edebilir ve bunu üretim kararlarında, katile miktarı belirlemede kullanır, bu sayede kurulum maliyetlerini azaltır. Buna karşın itme sistemler çekme sistemlerden daha fazla çalışılan iş stoku tutmaya çalışır. Çekme sistemler çalışılan iş stokunu azaltır, bu da elde tutma ve hurda gibi maliyetlerin azaltılmasını sağlar. İtme sistemler ayrıca katileler şeklinde değil de birer birer çalışmayı ön planda tuttuğu için kalite sorunlarının erken fark edilmesini sağlar. (Nahmias ve Olsen, 2015)

2.2. ÇİZELGELEME PROBLEMİ

Optimal çizelgeleme hem teorik hem de uygulama yönlerinden dolayı yöneylem araştırmasının önemli alanlarından biridir.

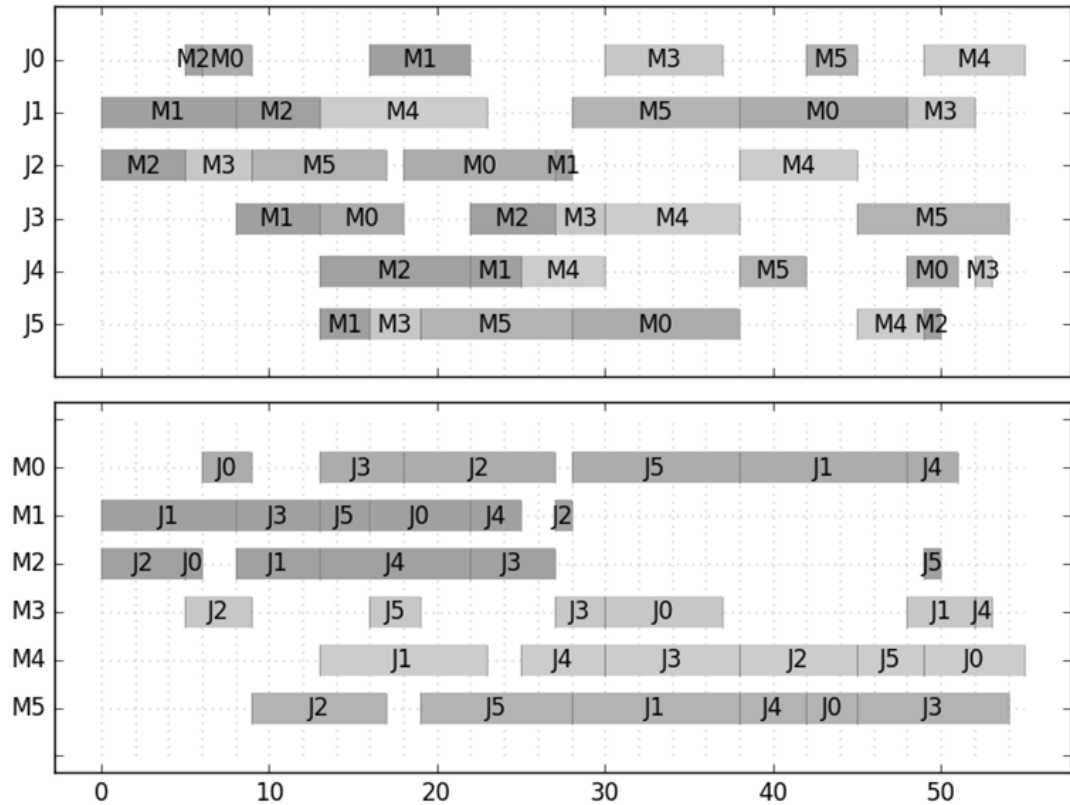
Çizelgeleme, kısıtlı kaynakların bir veya birden fazla performans ölçütünü optimize etmek amacıyla belirli bir zaman içindeki aktivitelere atanmasıdır (Lawler, Lenstra, Rinnooy Kan ve Shmoys, 1993). Kaynaklar ve etkinlikler birçok farklı biçimde olabilir. Kaynaklar bir fabrikadaki tezgahlar, bir hastanede hemşireler, otobüs sürücüleri, bir montaj tesisinde makineler, bir bulut hizmeti sağlayıcısında CPU'lar, bir otomobil tamirhanesinde teknisyenler vb. olabilir. Faaliyetler ise üretim sürecindeki işlemler, bir hastanede hemşirelerin görevleri, bilgisayar programlarının yürütülmesi, bir otomobil tamirhanesinde otomobillerin tamiri vb. olabilir. Ayrıca optimizasyon için birçok farklı performans ölçütü de bulunmaktadır. Bir amaç, ortalama akış süresinin en aza indirilmesi olabilirken,

bir başka amaç da miadından sonra tamamlanan iş sayısının en aza indirilmesi olabilir.

Çizelgeleme, işletme, endüstri mühendisliği ve bilgisayar mühendisliği araştırmacıları tarafından 60 yıldan fazla bir süredir yoğun olarak çalışılmaktadır.

Çizelgelemenin gerçek hayatta uygulamalarının genellikle büyük boyutlu olması ve hızlı karar verilmesini gerektirdiği için çoğunlukla sezgisel yöntemler tercih edilmektedir. Hızlı karar alabilmek için basit algoritmaların kullanılması genellikle optimalden oldukça uzak sonuçlar elde edilmesine yol açmaktadır.

Bir çizelge, her iş için bir veya daha fazla makineye bir veya daha fazla zaman aralığı tahsis edilmesidir. Belirli kısıtları sağlayacak şekilde bir çizelge bulunması problemine çizelgeleme problemi denir (Brucker, 2007). Çizelgeler Gantt şemaları ile gösterilebilirler. Gantt şemaları makine odaklı veya iş odaklı olabilir.



Şekil - 10 Örnek Gantt Şeması

2.2.1 Çizelgeleme Ortamları

Conway, Maxwell ve Miller (1967), çizelgeleme ortamının dört tip bilgi ile sınıflandırılabilceğini söylemişlerdir. Bunlar:

1. İşlenecek işler ve işlemler
2. Atölyede yer alan makinelerin sayısı ve tipi
3. İş atamalarının yapıma şeklini kısıtlayan kurallar
4. Çizelgelemenin değerdendirileceği kriterler

Bu dört tip bilgiye göre sınıflandırılan üretim ortamları aşağıda belirtilmiştir (Alharkan, 2005):

- i. **Tek makine üretim:** Tek makine ve işlenmesi gereken n sayıda iş vardır.
- ii. **Seri üretim:** Seri halinde m sayıda makine ve işlenmesi gereken n sayıda iş vardır.
- iii. **Atölye Tipi Üretim:** m sayıda makine ve n sayıda iş vardır. Her işin akışı belirlidir, işler bu akış sırasında makinelere bir veya birden fazla kez uğrayabilir.
- iv. **Açık Atölye Tipi Üretim:** m sayıda makine ve n sayıda iş vardır. Atölye tipi üretimden farkı işlerin akışı belirli değildir, makineler üzerinde işleme rotası ile ilgili bir kısıt bulunmamaktadır.
- v. **Kapalı Atölye Tipi Üretim:** Tüm üretim iş emirlerinin stok yenileme amaçlı olduğu atölye tipi üretimdir. Üretim müşteri siparişinden etkilenmez.

2.2.2. Çizelgeleme Problemi Kategorileri

Çizelgeleme problemleri aşağıdaki kategorilere ayrılabilir.

- i. **Deterministik çizelgeleme problemleri:** İşlerin atölyeye gelme durumu, tamamlanma tarihi, sıralaması, işlem süreleri ve makinelerin uygunluğu gibi çizelgeleme probleminin tüm unsurlarının stokastik faktörler içermediği, önceden bilindiği durumdur.

- a. **Statik çizelgeleme problemleri:** Statik çizelgelemede iş kümesi zamanla değişmez ve önceden temin edilebilir.
 - b. **Dinamik çizelgeleme problemleri:** İş kümesi zamanla değişir ve işler farklı zamanda gelirler.
- ii. **Stokastik çizelgeleme problemleri:** Problemin en az bir unsurunun stokastik faktör içerdiği durumdur.

2.2.3. Çizelgeleme Notasyonu

Graham, Lawler, Lenstra ve Kan (1979) çalışmalarında çizelgeleme problemlerinin sınıflandırılması için bir notasyon ortaya koymuşlardır. Makina ortamı, iş ve çizelgelemenin çeşitli özellikleri üç alanlı bir notasyon şeklinde $\alpha | \beta | \gamma$ olarak ifade edilir.

α : çizelgeleme ortamını belirtir

β : iş özelliklerini ve kısıtları belirtir

γ : amaç fonksiyonunu belirtir

Tablo – 2 α notasyonunun yani çizelgeleme ortamının alabileceği değerleri göstermektedir.

Tek Makine	1
Özdeş Paralel Makineler	P_m
Eşdeğer Paralel Makineler	Q_m
Bağımsız Paralel Makineler	R_m
Seri Üretim	F_m
Atölye Tipi Üretim	J_m
Açık Atölye	O_m

Tablo - 2 Çizelgeleme Ortamı Notasyonu (α)

Tablo – 3 β notasyonunun yani iş özelliklerinin alabileceği değerleri göstermektedir.

Başlangıç Zamanı	r_j
Kesinti	$Prmp$
Öncelik Kısıtları	$Prec$
Arıza	$Brkdwn$
Devridaim	$Recrc$
Permutasyon	$Prmu$

Tablo - 3 İş Özellikleri Notasyonu (β)

Tablo – 4 γ notasyonunun yani amaç fonksiyonunun alabileceği değerleri göstermektedir.

Tamamlanma zamanı	C_{max}
Maksimum gecikme	L_{max}
Toplam ağırlıklı tamamlanma süresi	$\sum w_j C_j$
Toplam ağırlıklı gecikme	$\sum w_j T_j$
Toplam tamamlanma süresi	$\sum C_j$

Tablo – 4 Amaç Fonksiyonu Notasyonu (γ)

2.3. KUTULAMA PROBLEMİ

Kutulama problemi geleneksel sırt çantası problemlerinin bir genellemesi olarak düşünülebilir ve sırt çantası problemi terminolojisi ile tanımlanabilir. Geleneksel sırt çantası problemlerinde hacimleri ve değerleri verilen nesnelere bir sırt çantasına (veya kutuya), nesnelere değerinin maksimum olacağı şekilde yerleştirilmeye çalışılırken, kutulama probleminde verilen nesnelere kullanılan alanı maksimize edecek (kullanılan kutu sayısını minimize edecek) şekilde birden fazla kutuya yerleştirilmeye çalışılırlar. Kutulama problemi aşağıdaki şekilde modellenir.

w_j = j nesnesinin ağırlığı

c = her kutunun kapasitesi

$N = 1, \dots, n$

$$y_i = \begin{cases} 1 & i \text{ kutusu kullanıldıysa} \\ 0 & \text{diğer} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & j \text{ nesnesi } i \text{ kutusuna atandıysa} \\ 0 & \text{diğer} \end{cases}$$

Amaç:

$$\min z = \sum_{i=1}^n y_i$$

Kısıtlar:

$$\sum_{j=1}^n w_j * x_{ij} \leq c * y_i \quad i \in N$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j \in N$$

NP-zor bir problem olan kutulama problemi (Garey ve Johnson 1979), bilgisayar bilimleri, ulaşım, lojistik ve iletişim gibi birçok alanda kullanılmıştır (Hashim,

Zulkipli, Januri ve Shariff, 2014). Konteynır ykleme, birden ok sabit diske veri yerleřtirme, iř planlaması, reklamları sabit uzunlukta radyo veya televizyon istasyonlarında paketleme, CD'lere byk bir mzik koleksiyonu saklama gibi rneklendirilebilir.

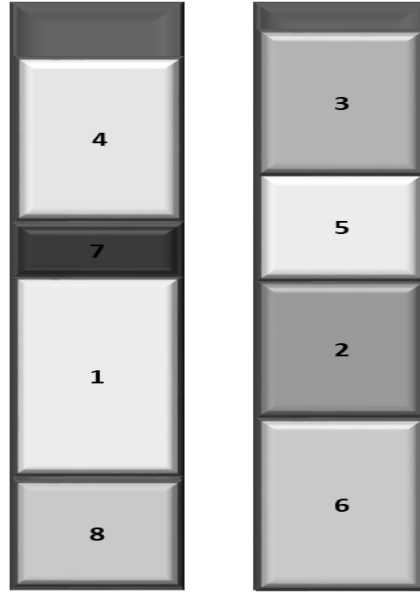
Kutulama problemi boyutlarına gre incelendiđinde bir boyutlu, iki boyutlu ve  boyutlu olmak zere sınıflandırılır.

2.3.1 Bir Boyutlu Kutulama Problemi

Geleneksel kutulama problemi bir boyutlu olarak tanımlanmıřtır. Bir boyutlu kutulama problemlerinde ykleme yapılacak olan nesnelerin sadece tek boyutları ele alınmakta ve buna bađlı olarak optimum ykleme yapılmaya alıřılmaktadır. Bu problemde ama bir izgi veya hat gibi grlebilecek yapının ierisine farklı boyutlardaki izgileri yerleřtirmek olarak dřnlebilir. Bir boyutlu olmayan nesnelere de alan ve/veya hacim deđerleri kullanılarak bir boyutlu kutulama problemi olarak ele alınabilir. Bir boyutlu kutulama probleminde, farklı boyutlardaki nesnelere kullanılan kutuların sayısını en aza indirecek řekilde sınırlı kapasiteli kutulara yerleřtirilmelidir. Bu problemde yapılan yklemelerin maksimize edilmesinin yanı sıra kullanılan kutu sayısını azaltmak amalanmıřtır (Hashim ve diđer., 2014)

rneđin ders programına derslerin verimli bir řekilde yerleřtirilmesi bir boyutlu problem olarak deđerlendirilebilir. Bu problemde her dersin farklı zaman gerektirdiđini ve her gne ait sınırlı sayıda alıřma saati olduđunu dřnrsek en fazla dersin en az zamanda yapılması gerekmektedir.

Bilgisayar bilimlerinde hafızanın daha verimli kullanılması iin yani bellek ynetimi iin programları hafızanın ierisine en verimli řekilde yerleřtirmek gerekmektedir. Bu tr problemler de bir boyutlu kutulama problemine rnek gsterilebilir.



Şekil - 11 Bir Boyutlu Kutulama Örnek Gösterim

2.3.2 İki Boyutlu Kutulama Problemi

İki boyutlu kutulama probleminde cismin iki boyutu (en ve boy) ele alınmaktadır. Bu problemlerde amaç yine en az sayıda kutu kullanmak ve boş alanı minimum seviyeye indirmektir.

Kesme ve yükleme problemleri en yaygın olarak bilinen örneklerdir. Depolama alanından tasarruf etmek için dörtgen şekilli ürünler depo raflarına verimli bir şekilde yerleştirilmelidir. Yayıncılık endüstrisinde makale ve reklamların en az sayıda sayfaya bölünmesi gerekir. Gazetelerde haberler en ve boyları göz önüne alınarak sayfalara yerleştirilir, bu problemler de iki boyutlu kutulama problemine örnek gösterilebilir.

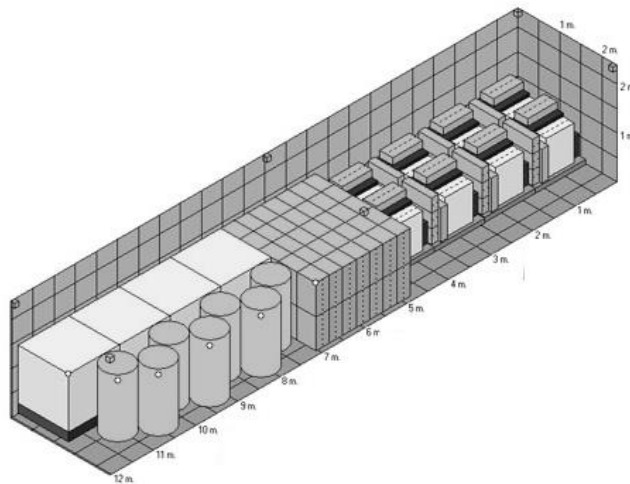


Şekil – 12 İki Boyutlu Kutulama Örnek Gösterim

2.3.3 Üç Boyutlu Kutulama Problemi

Üç boyutlu kutulama en zor kutulama problemi olarak tanımlanır. Bu problemde şeklin en, boy ve yükseklik boyutları ele alınır. Genellikle konteynır yükleme problemi olarak karşımıza çıkmaktadır. 3 boyutlu kutulama problemi tüm kutuları ortogonal olarak en az sayıda kutuya yerleştirmekten oluşur (Martello, Pisinger ve Vigo, 2003)

Lojistik uygulamalarında kullanılan konteynıra palet yerleştirme problemi üç boyutlu kutulama problemine örnek verilebilir. Burada cismin genişlik, yükseklik ve derinliği baz alınmaktadır. Bu tip problemlerde cismin döndürölüp döndürölmemeyeceđi de kriter olarak belirlenmektedir. Ayrıca cisimlerin girintili olup olmamasına göre de problemin zorluk derecesi artmaktadır.



Şekil - 13 Üç Boyutlu Kutulama Örnek Gösterim

2.4. OTOKLAV YÜKLEME VE ÇİZELGELEME ÜZERİNE ÇALIŞMALAR

Kompozit üretim tesislerinde otoklav yükleme ve çizelgeleme ile ilgili literatürde çok fazla çalışma bulunmamaktadır. Konu ile ilgili daha önce yapılan çalışmalar bu bölümde özetlenmiştir.

2.4.1. Lockheed Martin Otoklav Yükleme ve Çizelgeleme

Lockheed Yapay Zeka Merkezi araştırmacıları tarafından otoklav yükleme ve çizelgeleme ile ilgili iki ayrı sistem ile ilgili çalışma yayınlanmıştır. Hennesy ve Hinkle (1992) çalışmalarında otoklav yükleme için geliştirilen Clavier adı verilen sistemden bahsetmektedir. Clavier, önem sırasına göre, daha önce kürlenmiş parça grupları ile kürlenmeye girecek parça gruplarını durum tabanlı çıkarsama yöntemi ile karşılaştırır. Eğer kesin bir eşleşme bulunursa, bu yük önerilir. Eğer kesin bir eşleşme bulunamazsa, sistem benzer grupları kullanarak otoklav operatörüne yeni yük önerir. Operatör bu öneriyi kabul edebilir veya reddedebilir, eğer kabul ederse daha önce kürlenmiş parça grupları veritabanına eklenir. Çalışmada sadece otoklav yükleme problemi üzerinde durulmaktadır.

Lockheed Yapay Zeka Merkezi araştırmacıları tarafından yayınlanan ikinci çalışma, ilk sistem olan Clavier sisteminin çıktılarını kullanarak durum tabanlı çizelgeleme yapar. (Cook, Hinkle ve Bickmore, 1990). Bu sistem Clavier tarafından oluşturulan yükleri alır ve yükleri çizelgelemek için sezgisel bir yöntem uygular. Makale, kullanılan sezgisel yöntem, hesaplama sonuçları veya bu yöntemlerle çözülebilecek problemlerin boyutu hakkında bilgi içermemektedir. İlk çalışma sadece otoklav yükleme üzerine odaklanırken bu çalışma önceki çalışmanın sonuçlarını kullanarak otoklav çizelgeleme problemini çözmeye çalışmaktadır. Otoklav yükleme ve çizelgeleme birbirinden bağımsız problemler şeklinde ayrı ele alınmıştır.

2.4.2. Simülasyon Tabanlı Otoklav Yükleme ve Çizelgeleme Optimizasyonu

Hindle ve Duffin (2006) simülasyon ile otoklav yükleme ve çizelgeleme optimizasyonu üzerine bir çalışma yapmışlardır. Simul8, Visual8 firması tarafından geliştirilmiş, simülasyon tabanlı çizelgeleme yapan bir yazılımdır. Bu yazılım bir kompozit üretim tesisinin tamamını simüle edebilmektedir, bu sayede bir çizelgenin sürecin farklı basamaklarını nasıl etkileyeceği belirlenebilmektedir. Simul8, çizelge oluşturmak için sezgisel kural kümelerini kullanır, seçilen kural kümesi kullanıcı tarafından değiştirilebilir. Otoklavlarda belirli bir döngü başlatılmaya hazır olduğunda, sistem hangi parçaların otoklava toplu olarak girilmesi için uygun olduğunu arar. Birkaç yük mevcutsa, en az gevşek zamana sahip olan seçilir. Makalede, çizelgelemede kullanılan sezgisel yöntemler veya hesaplama sonuçları ile ilgili bir bilgi vermemektedir. Çalışma sadece otoklav çizelgeleme problemi üzerine odaklanmıştır.

2.4.3. Tekrar Otoklava Girmesi Gereken Parçalar

Ye, Li, Chen, Ma ve Leng (2014) otoklavda kürlenecek parçaların çizelgelenmesi için bir sezgisel yöntem geliştirmiştir. Geliştirdikleri sezgisel yöntem öncelik kuralları kullanarak hangi parçanın ne zaman çizelgeleneceğini belirlemekte ve otoklav yükleme ve çizelgeleme problemlerini aynı anda çözmektedir. Bazı parçalar otoklavdan çıktıktan sonra ikinci bir serim işleminden geçerek tekrar otoklava girebilmektedir. Aynı parçanın tekrar girebilmesi için ilk otoklavdan çıktıktan sonra belirli bekleme sürelerinin sağlanması, serim işleminin tekrarlanması için süre ayrılması gerekmektedir. Yöntem aynı zamanda tekrar otoklava giren operasyonları da dikkate almaktadır. Çizelgeleme amaç fonksiyonu; tüm otoklavların ortalama toplam kullanım oranlarını maksimize etmek ve aynı zamanda tamamlanma zamanını mümkün olduğunca kısaltmaktır. Çalışma otoklav yükleme ve çizelgelemeyi aynı anda ele almakla beraber matematiksel bir model geliştirmemiş, sezgisel yöntemle çözüm bulmaya çalışmıştır.

2.4.4. Bir Kompozit Üretim Tesisinde Otoklav Yükleme ve Çizelgeleme Optimizasyonu

Collart (2015) çalışmasında bir kompozit üretim tesisindeki otoklavların yüklenmesi ve çizelgelenmesini optimize etmeye çalışmıştır. Çizelgeleme ve yükleme (kutulama) problemleri ayrı ele alınmış, yükleme problemi meta sezgisel yöntemlerden biri olan tepe tırmanma sezgisel metodu değiştirilerek çözülmüş, yükleme problemi çözümünün çıktıları kullanılarak çizelgeleme problemi karma tamsayı doğrusal modelleme ile çözülmüştür.

Otoklav yükleme probleminin amacı toplam yük sayısını minimize etmek, aynı zamanda da otoklav önceliğini ve otoklav kullanım oranını maksimize etmektir.

Çizelgeleme probleminin amacı ise toplam maliyeti minimize etmektir.

Problemin çözümünde tamamlanma süreleri ve mialar doğrudan dahil edilmemiştir. Mialara uyum otoklav yükleme probleminin çözümü sırasında haftalık olarak yüklerin oluşturulması ve her yükün ait olduğu hafta için çizelgelenmesiyle çözülmeye çalışılmıştır.

2.4.5. Otoklav Yüklemede Kompozit Parça Yerleşiminin Optimizasyonu

Dios, Gonzalez, Dios ve Maffezzoli (2016) çalışmalarında otoklava yüklenecek parçaların yerleşimini optimize edecek bir karma tam sayı doğrusal programlama modeli önermişlerdir. Ana karar değişkenleri parçaların otoklava yerleşimi ve yönünü belirlerken amaç fonksiyonu ise kür süresinin maksimum etki zamanını minimize etmeye çalışmaktadır.

Çalışmada aşağıdaki varsayımlar yapılmıştır:

- Her parçanın konulduğu tepsinin ölçüleriyle sınırlandırılmış dikdörtgen bir yapısı vardır.
- Parçanın yüksekliği katlar arasındaki mesafeden daha küçüktür.
- Parçalar paralel veya dik bir şekilde tepsilere yerleştirilebilir ve parçanın paralel veya dik yerleştirilmesi parçanın termal özelliklerini etkilemez.

- Aynı kattaki parçaların yerleşimlerinden kaynaklı etkileşimler göz ardı edilmiştir.

Çalışma oluşturulan yük sayısının veya kullanılan alanın minimize edilmesini değil kütleme işleminin kalitesini arttırmak için maksimum etki süresinin minimize edilmesini sağlamaktadır.

Tablo – 5 otoklav yükleme ve çizelgeleme ile ilgili daha önce yapılan çalışmaları özetlemektedir.

Çalışma	Firma	Özet	Kapsam	Yöntem
Hennesy ve Hinkle, 1992	Lockheed Martin	Önem sırasına göre, daha önce kürlenmiş parça grupları ile kürelemeye girecek parça gruplarını durum tabanlı çıkarsama yöntemi ile karşılaştırır. Eğer kesin bir eşleşme bulunursa, bu yük önerilir. Eğer kesin bir eşleşme bulunamazsa, sistem benzer grupları kullanarak otoklav operatörüne yeni yük önerir. Operatör bu öneriyi kabul edebilir veya reddedebilir, eğer kabul ederse daha önce kürlenmiş parça grupları veritabanına eklenir.	Otoklav yükleme	Durum tabanlı çıkarsama (Case based reasoning)
Cook, Hinkle ve Bickmore, 1990	Lockheed Martin	Bu sistem Clavier tarafından oluşturulan yükleri alır ve yükleri çizelgelemek için sezgisel bir yöntem uygular.	Otoklav çizelgeleme	Sezgisel
Hindle ve Duffin, 2006	Northrop Grumman	Otoklavlarda belirli bir döngü başlatılmaya hazır olduğunda, sistem hangi parçaların otoklava toplu olarak girilmesi için uygun olduğunu arar. Birkaç yük mevcutsa, en az gevşek zamana sahip olan seçilir.	Otoklav çizelgeleme	Simulasyon
Ye, Li, Chen, Ma ve Leng, 2014		Öncelik kuralları kullanarak hangi parçanın ne zaman çizelgeneceğini belirlemekte ve otoklav yükleme ve çizelgeleme problemlerini aynı anda çözmektedir. Yöntem aynı zamanda tekrar otoklava giren operasyonları da dikkate almaktadır.	Otoklav yükleme ve çizelgeleme	Sezgisel
Collart, 2015	AeroSpace Technologies of Australia (ASTA)	Yükleme problemi meta sezgisel yöntemlerden biri olan tepe tırmanma sezgisel metodu değiştirilerek çözülmüş, yükleme problemi çözümünün çıktıları kullanılarak çizelgeleme problemi karma tamsayı doğrusal modelleme ile çözülmüştür.	Otoklav yükleme ve çizelgeleme	Otoklav yükleme sezgisel, otoklav çizelgeleme karma tamsayı doğrusal programlama
Dios, Gonzalez, Dios ve Maffezzoli, 2016		Otoklava yüklenecek parçaların yerleşimini optimize edecek bir karma tam sayı doğrusal programlama modeli oluşturulmuştur.	Otoklav yükleme	Karma tamsayı doğrusal programlama

Tablo – 5 Otoklav Yükleme ve Çizelgeleme Üzerine Yapılan Çalışmalar

2.5. PROBLEM TANIMI

Kompozit İşlemleri Üretim Planlama ve Kontrol bölümünde ana problem, darboğaz olan otoklavların çizelgelenmesi sırasında oluşan sorunlar ve manuel yapılan çizelgelemenin yetersiz kalmasıdır. Çizelgeleme işlemi, yıllardır yapılan yükleme işlemlerinden deneme yanılma yöntemiyle elde edilmiş olan tecrübeler doğrultusunda, yalnızca belirli kür gruplarını otoklavlara atayarak yapıldığı için kapasitenin normalin altında kullanıldığı, ürünlerin scrap olma oranının yüksek olduğu tespit edilmiştir. Mevcut sistem firma gerekliliklerini sağlayamamaktadır. Aşağıdaki başlıklarda kürleme işleminde yaşanan genel sorunlar sıralanmıştır;

Temiz Oda: Temizlenmiş kalıpların üzerine, ihtiyaca göre önceden hazırlanmış olan kompozit malzemeler yapıştırılmakta ve takımlar üzerindeki bu parçalar torbalanıp otoklav yerleşimine hazır hale getirilmektedir. Temiz oda belirli bir kapasite ile çalışmaktadır, kalıpların bulunamaması ve zamanında temizlenememesi gibi nedenler ile temiz oda programının gerisinde kalmakta ve manuel yapılan otoklav çizelgelemesinin mevcut parçalar ve yeni gelenler ile tekrarlanması gerekmektedir. Tekrarlanan çizelgede hazırlanan parçaların son kullanım ömrü ve ihtiyaç zamanı sıkıntı yaratmakta ve her bir parçanın scrap olma ihtimali artmaktadır. Scrap olan parçaların kalıbı temizlendikten sonra ihtiyaç tarihinden önce karşılanması için önceliklendirilerek temiz odadan gelip otoklava yerleştirilmesi gerekmektedir.

Kür Grupları: Her bir parça serilen kompozit malzemenin ısı ve basınç ihtiyacına göre gruplara yerleştirilmiştir. Parça üzerinde serilen, aynı basınç ve ısıya ihtiyaç duyan malzemeler aynı kür grubuna aittir. Bir otoklava bir seferde yalnızca aynı kür grubuna ait parçalar atanabilmektedir. Manuel çizelgelemede, birlikte otoklava girebilen gruplar oluşturulmuş ve sadece bu gruplar otoklavlara atanmıştır, sürekli atanan bu grupların ihtiyaca uymaması kapasitenin verimsiz kullanılmasına, ihtiyacın gecikmesine ve otoklavların yetersiz görülmesine neden olmaktadır.

Otoklav Kapasiteleri: Parçalar atandıkları otoklavın önünde arabalara yerleştirilmektedir. Çizelgeleme yapılırken parça ve otoklav boyutları kontrol edilmediği için yerleşimde sorunlar yaşanmaktadır.

Thermocouple ve Vacuumport Kapasiteleri: Yerleşim tamamlandıktan sonra her bir parçaya uygulanan basıncın ve ısının kontrol edilmesini sağlayan bu aparatlar bir ucu otoklavda diğer ucu parçada olacak şekilde bağlanmaktadır. Otoklavlarda belirli sayılarda bulunan bu aparatlar parça aparat ihtiyaçları göz önüne alınmadığında sorun yaratmakta, parçanın atanan zamanda atanan otoklava girememesine neden olmaktadır.

Mevcut sorunlar göz önüne alındığında otoklav çizelgelemesi yapılırken tüm çevresel faktörlerin dahil edilmesi ve maksimum kapasite kullanımı amaçlanmıştır. Böylelikle enerji kullanımı da minimize edilmiş olacaktır. Problem kısıtları aşağıda sıralanmıştır;

- Parçaların ihtiyaç tarihinden önce otoklavlara atılması
- Parçaların ait oldukları kür grubu ile birlikte otoklava yerleştirilmesi
- Parçaların kendi boyutuna uygun olan otoklavlara atanabilmesi
- Atama yapılır iken otoklav ve parça için thermocouple ve vacuumport miktarının kontrolünün yapılması
- Takımların temizlenip diğer parçaya hazırlanması için gerekli minimum çevrim süresi kısıtına uyulması

3. BÖLÜM

MATEMATİKSEL MODELLER VE UYGULAMA

Sıklıkla değişen öncelikler, üretim sürecindeki belirsizlikler ve yönetim kararları nedeni ile günlük çizelgelemenin ötesinde bir ufukta yapılan çizelgelemenin değiştirilmeden uygulanması mümkün olmamasına rağmen firma en az 2 haftalık planlama ufukunda bir çizelgeye sahip olmak istemektedir. Her ne kadar değişeceği bilinse de 2 haftalık çizelge, malzeme ve iş gücü / vardiya planlarının yapılabilmesi için gerekli görülmektedir. Çizelgenin değiştirilmesi gerektiği durumlarda da matematiksel modelin hızlı bir şekilde optimum sonucu vermesi beklenmektedir, bazen günde birkaç kez çizelge değişimine gidilebildiği için 2 haftalık bir çizelge için maksimum 30 dakikalık bir çalışma süresi makul olarak kabul edilmektedir.

Bugüne kadar otoklav yükleme ve çizelgeleme üzerine yapılan çalışmalarda otoklav yükleme ve çizelgelemenin aynı anda ele alındığı bir matematiksel modele rastlanmamıştır.

Hennesy ve Hinkle (1992) sadece otoklav yükleme üzerine çalışmış ve bu amaçla durum tabanlı çıkarsama yöntemini kullanmışlardır, Cook ve diğer. (1990) ise aynı problemin çizelgeleme kısmını ele almış, Hennesy ve Hinkle'ın (1992) çalışmasının sonucunda çıkan yükleri kullanarak durum tabanlı çizelgeleme yöntemi ile çizelge oluşturmuşlardır. Hindle ve Duffin (2006) her ne kadar hem yükleme hem de çizelgeleme problemini bir arada ele alsada matematiksel bir optimizasyon modeli ile değil simülasyon modelleme ile bir çözüm elde etmişlerdir. Ye ve diğer. (2014) de yükleme ve çizelgeleme problemini bir arada ele almış fakat matematiksel bir model üzerine çalışmamışlar, sezgisel model geliştirmişlerdir. Collart (2015) yükleme ve çizelgeleme problemlerini ayrı ele almış, otoklav yükleme için sezgisel bir yöntem kullanmış, otoklav çizelgelemesi için ise bu sezgisel yöntemin sonuçlarını kullanacak şekilde bir matematiksel optimizasyon modeli geliştirmiştir. Dios ve diğer. (2016) ise sadece yükleme üzerine odaklanmış, yükleme yaparken parça yerleşimlerinin optimizasyonu için bir matematiksel model geliştirmişlerdir.

Bu tez kapsamında;

- i. Otoklav sayısının yükleme ve çizelgeleme problemlerinin ayrı ayrı ele alınması ve birlikte modellenmesi durumunda sonuca etkileri incelenecektir. Otoklav maliyetleri 800.000 – 1.200.000 ABD doları aralığındadır, bir otoklavın işletim maliyeti ise yıllık 500.000 ABD dolarına ulaşmaktadır. Optimum otoklav sayısının tespiti firmanın otoklav yatırım kararlarını almasında destek olacaktır, mevcut otoklavların optimum düzeyde çalıştırılması ise işletim maliyetlerini en aza indirgeyerek maliyet avantajı sağlayacaktır.
- ii. Daha önce hiçbir çalışmada rastlanmayan, takım sayısının arttırılmasının sonuca etkisi incelenecektir. Takım maliyetleri 10.000 – 500.000 ABD doları arasında değişse de genellikle 200.000 ABD doları civarında maliyetleri bulunmaktadır. Her parça için optimum sayıda takımın bulundurulması takımların yüksek maliyetleri nedeni ile önemli bir yatırım kararı oluşturmaktadır.
- iii. Otoklav yükleme ve çizelgeleme problemlerinin ayrı ayrı ele alınması ve birlikte modellenmesi ile elde edilen sonuçlar karşılaştırılacaktır.
- iv. Yükleme ve çizelgelemenin birlikte ele alındığı model firmanın gerçek verileri üzerinden 14 farklı gün seçilerek çalıştırılacak ve modelin pratikte kullanılabilirliği analiz edilecektir.
- v. Oluşturulan modelin firmanın en az 2 haftalık planlama ufkuunda çizelgeleme talebini karşılayıp karşılayamayacağını görmek üzere firmanın gerçek verileri üzerinde 1 ay seçilerek model çalıştırılacak ve pratikte kullanılabilirliği analiz edilecektir.

Bu tez çalışmasında 3 matematiksel model oluşturulmuştur;

- i. Otoklav yükleme ve çizelgelemeyi ayrı ele almak üzere otoklav yükleme matematiksel optimizasyon modeli
- ii. Otoklav yükleme ve çizelgelemeyi ayrı ele almak üzere otoklav çizelgeleme matematiksel optimizasyon modeli
- iii. Otoklav yükleme ve çizelgelemeyi bir arada ele almak üzere otoklav yükleme ve çizelgeleme matematiksel optimizasyon modeli.

3.1. OTOKLAV YÜKLEME MATEMATİKSEL MODELİ

Bu tez çalışmasında ele alınacak olan otoklav yükleme problemi 1 boyutlu kutulama problemleri sınıfında yer alan NP-Zor bir kombinatoriyal optimizasyon problemidir. Klasik kutulama problemi amaç fonksiyonlarından farklı olarak amaç kutu sayısını minimize etmek değil otoklavın kullanılmayan alanını minimize ederken karşılanmayan talebi de en aza indirmek olarak belirlenmiştir.

Otoklav yükleme problemi 3 boyutlu bir kutulama problemi olarak da ele alınabilir bu tez çalışmasında aşağıdaki belirtilen nedenler ile 1 boyutlu olarak ele alınmasının sonucu etkilemeyeceği görülmüştür:

- i. Parça yükseklikleri otoklav yüksekliğinden çok küçüktür, ısı ve basınç etkileri katlı yüklemelerde farklılık gösterebildiği ve kompozit parçanın kalitesini etkilediği için firma tarafından katlı yükleme tercih edilmemektedir. Bu nedenlerle yükseklik boyutu yükleme probleminde göz ardı edilmiştir.
- ii. Parçanın eni ve/veya boyu otoklavın en ve/veya boyundan büyük olduğu durumlarda parçanın otoklava yüklenmesinin engellenmesi gerekmektedir, bu durumda yükleme probleminin boy ve en boyutlarını kullanması gerekmektedir. Bu kısıt otoklava girebilecek parçaların kümesi oluşturularak ve uygun olmayan parçaların otoklava girmesi engellenerek farklı bir şekilde ele alınmıştır.
- iii. Thermocouple sayısı kısıtı, vacuumport sayısı kısıtı ve kür grubu kısıtları nedeniyle genellikle otoklavın tüm alanını doldurmak mümkün olmamaktadır, yükleme sonrası boşluklar olduğu için yüklemeyi yapan operatör parçanın yönünü değiştirerek uygun yerleşimi oluşturabilir, parçanın boy ve en boyutları tek boyutta (alan boyutu) birleştirilerek modele dahil edilmiştir.

Vacuumport sayısının tek başına kısıtlayıcı olmadığı, thermocouple sayısı kısıtının vacuumport sayısını da kısıtladığı tespit edilmiştir. Bu nedenle modele sadece thermocouple sayısı kısıtı eklenmiştir.

Kompozit parçalar serim aşamasında temiz odada özel takımlar üzerine yerleştirilmekte ve bu takımlar ile otoklava girmektedirler. Her parça için kullanılabilir takım sayısı kısıtlıdır.

Yükleme yapılırken, ısı ve basınç karakterleri aynı olan sadece aynı kür grubundaki parçalar aynı anda otoklava girebilmektedir. Bu kısıt otoklav yükleme problemini sınıf kısıtlı kutulama problemi sınıfına sokmaktadır.

Otoklav yükleme problemi için matematiksel optimizasyon modeli aşağıdaki şekilde oluşturulmuştur.

Kümeler:

P : Kompozit parçalar

O : Otoklavlar

Z : Zaman dilimleri

C : Kür grupları

I : Yükler

Parametreler:

$CS_{c,o}$: "c" kür grubunun "o" otoklavındaki çevrim süresi (Tamsayı)

T_p : "p" parçası için uygun takım sayısı (Tamsayı)

$D_{p,z}$: "p" parçasının "z" zamandaki toplam talebi (Tamsayı)

TCO_o : "o" otoklavındaki thermocouple sayısı (Tamsayı)

TCP_p : "p" parçasının işlem görebilmesi için gerekli thermocouple sayısı (Tamsayı)

AO_o : "o" otoklavının alanı (Tamsayı)

AP_p : "p" parçasının alanı (Tamsayı)

M : Çok büyük pozitif tamsayı

$B_{p,o}$: $\begin{cases} 1, & p \text{ parçasının boyutu } o \text{ otoklavına uygunsa} \\ 0, & \text{diğer} \end{cases}$

CGP_p : "p" parçasının kür grubu

Değişkenler:

$y_{i,o}$: $\begin{cases} 1, & i \text{ yükü } o \text{ otoklavında kullanıldıysa} \\ 0, & \text{diğer} \end{cases}$

$q_{p,i,o}$: o otoklavında i yüküne atanan p parçası sayısı

$x_{p,i,o}$: $\begin{cases} 1, & i \text{ yükü } o \text{ otoklavına en az } 1 \text{ } p \text{ parçası atandıysa} \\ 0, & \text{diğer} \end{cases}$

$w_{c,i,o}$: $\begin{cases} 1, & o \text{ otoklavındaki } i \text{ yükü } c \text{ kür grubundaysa} \\ 0, & \text{diğer} \end{cases}$

$UA_{o,i}$: o otoklavının i yükünde kullanılmayan alanı

TUD_p : p parçasının toplam karşılanamayan talebi

Amaç Fonksiyonu:

$$\text{Minimize } a1 * \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + a2 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) \quad (0)$$

Kısıtlar:

$$q_{p,i,o} \leq M * y_{i,o} \quad \forall p, i, o \quad (1)$$

$$q_{p,i,o} \leq M * x_{p,i,o} \quad \forall p, i, o \quad (2)$$

$$x_{p,i,o} \leq y_{i,o} \quad \forall p, i, o \quad (3)$$

$$\sum_{p \in UP_o} AP_p * q_{p,i,o} = AO_o * y_{i,o} - UA_{o,i} \quad \forall i, o \quad (4)$$

$$\sum_{p \in UP_o} TCP_p * q_{p,i,o} \leq TCO_o * y_{i,o} \quad \forall o, i \quad (5)$$

$$q_{p,i,o} \leq T_p \quad \forall p, i, o \quad (6)$$

$$x_{p,i,o} \leq B_{p,o} \quad \forall p, i, o \quad (7)$$

$$y_{i,o} \leq y_{i-1,o} \quad \forall i, o \quad i > 1 \quad (8)$$

$$\sum_{i \in I} \sum_{o \in O} q_{p,i,o} = \sum_{z \in Z} D_{p,z} - TUD_p \quad \forall p \quad (9)$$

Sınıf Kısıtlı Kutulama:

$$\sum_{c \in C} w_{c,i,o} = y_{i,o} \quad \forall i, o \quad (10)$$

$$w_{c,i,o} + x_{p,i,o} \leq 1 \quad \forall p, i, o, c \quad CGP_p \neq c \quad (11)$$

(0) Amaç fonksiyonunu tanımlar. Otoklavların kullanılmayan alanının yüzdesini ve toplam karşılanmayan talep yüzdesini minimize eder.

(1) Parça yüke atanmadıysa yükün kullanımını engeller.

(2) Parçanın otoklava yüklenmesi kısıtı.

(3) Yük otoklava atanmadıysa parçanın atanmasını engeller.

(4) Otoklavda çizelgelenecek toplam ürün alanının otoklav alanından küçük olmasını sağlar.

(5) Otoklava yüklenecek parçalar için gereken toplam thermocouple sayısının, otoklavın thermocouple sayısından küçük olmasını sağlar.

(6) Yüke atanan parça sayısının, parçanın takım sayısından fazla olmasını engeller.

(7) Parçanın boyutunu uygun olmayan otoklava atanmasını engeller.

(8) Yüklerin sırayla atanmasını sağlar

- (9) Tüm talebi karşılayacak şekilde yüklerin atanmasını sağlar.
- (10) Otoklava atanan bir yükte birden fazla kür grubunun olmasını engeller.
- (11) Otoklavdaki yüke farklı kür grubundan parçalar atanmasını engeller.

3.2. OTOKLAV ÇİZELGELEME MATEMATİKSEL MODELİ

Otoklav çizelgeleme problemi, yükleme probleminin optimizasyonu sonucunda oluşan yükleri çizelgelemek üzere formüle edilmiştir.

Amaç fonksiyonu zamanında karşılanmayan talebi ve planlama ufku sonunda toplam karşılanmayan talebi minimize etmektedir.

Kompozit parçalar serim aşamasında temiz odada özel takımlar üzerine yerleştirilmekte ve bu takımlar ile otoklava girmektedirler. Takımlar otoklavdan çıktıktan sonra kompozit parçalar üzerlerinden sökülümekte ve tekrar kullanıma hazır hale getirmek için temizlenmektedir. Her takım için otoklavdan çıktıktan tekrar kullanıma alınıncaya kadar gerekli minimum çevrim süresi tanımlıdır. Çizelgeleme problemi takımların otoklavda olduğu sürece ve çıktıktan sonra çevrim süresi boyunca kullanımını engelleyecek şekilde formüle edilmiştir.

Yükleme problemi kullanılan otoklav alanını maksimize edecek şekilde çalıştığından, sonucunda oluşan yüklerden sadece biri aynı anda otoklavda olacak şekilde kısıtlanmıştır.

Otoklav çizelgeleme problemi için matematiksel optimizasyon modeli aşağıdaki şekilde oluşturulmuştur.

Kümeler:

P : Kompozit parçalar

O : Otoklavlar

Z : Zaman dilimleri

C : Kür grupları

I : Yükler

Parametreler:

CST_p : "p" parçası için minimum takım çevrim süresi (Tamsayı)

T_p : "p" parçası için uygun takım sayısı (Tamsayı)

$CS_{c,o}$: "c" kür grubunun "o" otoklavındaki çevrim süresi (Tamsayı)

$D_{p,z}$: "p" parçasının "z" zamandaki toplam talebi (Tamsayı)

TCO_o : "o" otoklavındaki thermocouple sayısı (Tamsayı)

TCP_p : "p" parçasının işlem görebilmesi için gerekli thermocouple sayısı (Tamsayı)

AO_o : "o" otoklavının alanı (Tamsayı)

AP_p : "p" parçasının alanı (Tamsayı)

CZ_o : "o" otoklavının çalışmadığı zamanlar

M : Çok büyük pozitif tamsayı

CGP_p : "p" parçasının kür grubu

$y_{i,o}$: $\begin{cases} 1, & i \text{ yükü } o \text{ otoklavında kullanıldıysa} \\ 0, & \text{diğer} \end{cases}$

$q_{p,i,o}$: o otoklavında i yüküne atanan p parçası sayısı

$$x_{p,i,o} : \begin{cases} 1, & \text{i yükü o otoklavına en az 1 p parçası atandıysa} \\ 0, & \text{diğer} \end{cases}$$

$$w_{c,i,o} : \begin{cases} 1, & \text{o otoklavındaki i yükü c kür grubundaydısa} \\ 0, & \text{diğer} \end{cases}$$

Değişkenler:

$qz_{p,i,o,z}$: z zamanında o otoklavında i yüküne atanan p parçası sayısı

$$Yuk_{i,o,z} : \begin{cases} 1, & \text{o otoklavındaki i yükü z zamanında çizelgelendiye} \\ 0, & \text{diğer} \end{cases}$$

TUD_p : p parçasının toplam karşılanamayan talebi

$UD^+_{p,z}$: p parçasının z zamanında fazla karşılanan talebi

$UD^-_{p,z}$: p parçasının z zamanında karşılanamayan talebi

Amaç Fonksiyonu:

$$\text{Minimize } a1 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + a2 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD^+_{p,z}}{D_{p,z}} \right) + a3 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD^-_{p,z}}{D_{p,z}} \right) \quad (0)$$

Kısıtlar:

$$\sum_{z \in Z} Yuk_{i,o,z} \leq y_{i,o} \quad \forall i, o \quad (1)$$

$$\sum_{i \in I} \sum_{o \in O} \sum_{z \in Z} qz_{p,i,o,z} = \sum_{z \in Z} D_{p,z} - TUD_p \quad \forall p \quad (2)$$

$$\sum_{i \in I} Yuk_{i,o,z} \leq 1 \quad \forall o, z \quad (3)$$

$$\sum_{i \in I} \sum_{z \in Z_o} Yuk_{i,o,z} = 0 \quad \forall o \quad (4)$$

$$\sum_{c \in C} \sum_{i1 \in I} \sum_{z1 \in Z} Yuk_{i1,o,z1} * w_{c,i1,o} \leq M * (1 - Yuk_{i,o,z}) \quad \forall i, o, z, z1 \geq z, z1 \leq z + CS_{c,o}, i1 \neq i, z1 \neq z \quad (5)$$

$$\sum_{i \in I} \sum_{o \in O} \sum_{z1 \in Z} q_{z_{p,i,o,z1}} = D_{p,z} - UD_{p,z}^- + UD_{p,z}^+ \quad \forall p, z, z1 \leq z - CS_{CGP_{p,o}} \quad (6)$$

$$\min(CS_{CGP_{p,o}} + CST_{p,Z-z}) \sum_{t=0} \sum_{i \in I} \sum_{o \in O} q_{z_{p,i,o,\min(z+t,Z)}} \leq T_p \quad \forall z \in Z, p \in P \quad (7)$$

$$(q_{z_{p,i,o,z}} - q_{p,i,o}) \leq M * (1 - Yuk_{i,o,z}) \quad \forall p, i, o, z \quad (8)$$

$$\sum_{p \in P} q_{z_{p,i,o,z}} \leq M * Yuk_{i,o,z} \quad \forall i, o, z \quad (9)$$

$$Yuk_{i,o,z} \leq \sum_{p \in P} q_{z_{p,i,o,z}} \quad \forall i, o, z \quad (10)$$

(0) Amaç fonksiyonunu tanımlar. Toplam karşılanmayan talep yüzdesini, zamanında karşılanamayan talep yüzdesini ve talep zamanından önce karşılanan talep yüzdesini minimize eder.

- (1) Otoklava atanmayan yükün çizelgelenmesini engeller.
- (2) Parçanın tüm talebinin son Z zamanına kadar karşılanmasını sağlar.
- (3) Otoklavda aynı zaman diliminde birden fazla yükün çizelgelenmesini engeller.
- (4) Otoklavın kapalı olduğu zamanlarda yük çizelgelenmesini engeller.
- (5) Bir otoklavda z zamanında bir yük başlatıldıysa, bu yükün çevrim süresi boyunca atanan otoklavda başka yükün başlatılmasını engeller.
- (6) Parçanın otoklavda talep tarihinden çevrim süresi kadar önce başlatılmasını sağlar.
- (7) Takımın çevrim süresi boyunca kullanılmasını engeller.

(8) z zamanında i yükü o otoklavına yüklendiyse, p parçasının z zamanında yüklenen o otoklavındaki i yükünün miktarının p parçasının o otoklavındaki i yükündeki miktarına eşit olmasını sağlar

(9) Yük, otoklava çizelgelenmediyse miktarın 0 olmasını sağlar.

(10) Çizelgelenen miktar 0 ise yükün otoklava çizelgelenmesini engeller.

3.3. OTOKLAV YÜKLEME VE ÇİZELGELEME MATEMATİKSEL MODELİ

Otoklav yükleme ve çizelgeleme problemini bir arada optimize etmek amacıyla matematiksel model aşağıdaki şekilde oluşturulmuştur.

Kümeler:

P : Kompozit parçalar

O : Otoklavlar

Z : Zaman dilimleri

C : Kür grupları

I : Yükler

Parametreler:

CST_p : "p" parçası için minimum takım çevrim süresi (Tamsayı)

T_p : "p" parçası için uygun takım sayısı (Tamsayı)

$CS_{c,o}$: "c" kür grubunun "o" otoklavındaki çevrim süresi (Tamsayı)

$D_{p,z}$: "p" parçasının "z" zamandaki toplam talebi (Tamsayı)

TCO_o : "o" otoklavındaki thermocouple sayısı (Tamsayı)

TCP_p : "p" parçasının işlem görebilmesi için gerekli thermocouple sayısı (Tamsayı)

AO_o : "o" otoklavının alanı (Tamsayı)

AP_p : "p" parçasının alanı (Tamsayı)

CZ_o : "o" otoklavının çalışmadığı zamanlar

M : Çok büyük pozitif tamsayı

$B_{p,o}$: $\begin{cases} 1, & p \text{ parçasının boyutu } o \text{ otoklavına uygunsa} \\ 0, & \text{diğer} \end{cases}$

CGP_p : "p" parçasının kür grubu

Değişkenler:

$y_{i,o}$: $\begin{cases} 1, & i \text{ yükü } o \text{ otoklavında kullanıldıysa} \\ 0, & \text{diğer} \end{cases}$

$q_{p,i,o}$: o otoklavında i yüküne atanan p parçası sayısı

$qz_{p,i,o,z}$: z zamanında o otoklavında i yüküne atanan p parçası sayısı

$x_{p,i,o}$: $\begin{cases} 1, & i \text{ yükü } o \text{ otoklavına en az } 1 \text{ } p \text{ parçası atandıysa} \\ 0, & \text{diğer} \end{cases}$

$Yuk_{i,o,z}$: $\begin{cases} 1, & o \text{ otoklavındaki } i \text{ yükü } z \text{ zamanında çizelgelendiye} \\ 0, & \text{diğer} \end{cases}$

$w_{c,i,o}$: $\begin{cases} 1, & o \text{ otoklavındaki } i \text{ yükü } c \text{ kür grubundaysa} \\ 0, & \text{diğer} \end{cases}$

TUD_p : p parçasının toplam karşılanamayan talebi

$UD^+_{p,z}$: p parçasının z zamanında fazla karşılanan talebi

$UD^-_{p,z}$: p parçasının z zamanında karşılanamayan talebi

$UA_{o,i}$: o otoklavının i yükünde kullanılmayan alanı

Amaç Fonksiyonu:

$$\begin{aligned} \text{Minimize } a1 * \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + a2 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + a3 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}^+}{D_{p,z}} \right) \\ + a4 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}^-}{D_{p,z}} \right) \end{aligned} \quad (0)$$

Kısıtlar:**Kutulama:**

$$q_{p,i,o} \leq M * y_{i,o} \quad \forall p, i, o \quad (1)$$

$$q_{p,i,o} \leq M * x_{p,i,o} \quad \forall p, i, o \quad (2)$$

$$x_{p,i,o} \leq y_{i,o} \quad \forall p, i, o \quad (3)$$

$$\sum_{p \in UP} AP_p * q_{p,i,o} = AO_o * y_{i,o} - UA_{o,i} \quad \forall i, o \quad (4)$$

$$\sum_{p \in UP} TCP_p * q_{p,i,o} \leq TCO_o * y_{i,o} \quad \forall o, i \quad (5)$$

$$x_{p,i,o} \leq B_{p,o} \quad \forall p, i, o \quad (6)$$

$$y_{i,o} \leq y_{i-1,o} \quad \forall i, o \quad i > 1 \quad (7)$$

Sınıf Kısıtlı Kutulama:

$$\sum_{c \in C} w_{c,i,o} = y_{i,o} \quad \forall i, o \quad (8)$$

$$w_{c,i,o} + x_{p,i,o} \leq 1 \quad \forall p, i, o, c \quad CGP_p \neq c \quad (9)$$

Çizelgeleme:

$$\sum_{z \in Z} Yuk_{i,o,z} \leq y_{i,o} \quad \forall i, o \quad (10)$$

$$\sum_{i \in I} \sum_{o \in O} \sum_{z \in Z} q_{z,p,i,o,z} = \sum_{z \in Z} D_{p,z} - TUD_p \quad \forall p \quad (11)$$

$$\sum_{i \in I} Yuk_{i,o,z} \leq 1 \quad \forall o, z \quad (12)$$

$$\sum_{i \in I} \sum_{z \in Z_o} Yuk_{i,o,z} = 0 \quad \forall o \quad (13)$$

$$\sum_{c \in C} \sum_{i1 \in I} \sum_{z1 \in Z} Yuk_{i1,o,z1} * w_{c,i1,o} \leq M * (1 - Yuk_{i,o,z}) \quad \forall i, o, z, z1 \geq z, z1 \leq z + CS_{c,o}, i1 \neq i, z1 \neq z \quad (14)$$

$$\min(CS_{CGP_{p,o}} + CST_{p,Z-z}) \sum_{t=0} \sum_{i \in I} \sum_{o \in O} q_{z,p,i,o,\min(z+t,Z)} \leq T_p \quad \forall z \in Z, p \in P \quad (15)$$

$$(q_{z,p,i,o,z} - q_{p,i,o}) \leq M * (1 - Yuk_{i,o,z}) \quad \forall p, i, o, z \quad (16)$$

$$\sum_{p \in P} q_{z,p,i,o,z} \leq M * Yuk_{i,o,z} \quad \forall i, o, z \quad (17)$$

$$Yuk_{i,o,z} \leq \sum_{p \in P} q_{z,p,i,o,z} \quad \forall i, o, z \quad (18)$$

$$\sum_{i \in I} \sum_{o \in O} \sum_{z1 \in Z} q_{z,p,i,o,z1} = D_{p,z} - UD_{p,z}^- + UD_{p,z}^+ \quad \forall p, z, z1 \leq z - CS_{CGP_{p,o}} \quad (19)$$

(0) Amaç fonksiyonunu tanımlar. Otoklavların kullanılmayan alan yüzdesini, toplam karşılanmayan talep yüzdesini ve talep zamanından önce karşılanan talep yüzdesini minimize eder.

(1) Parça yüke atanmadıysa yükün kullanımını engeller.

(2) Parçanın otoklava yüklenmesi kısıtı.

(3) Yük otoklava atanmadıysa parçanın atanmasını engeller.

- (4) Otoklavda çizelgelenecek toplam ürün alanının otoklav alanından küçük olmasını sağlar.
- (5) Otoklava yüklenecek parçalar için gereken toplam thermocouple sayısının, otoklavın thermocouple sayısından küçük olmasını sağlar.
- (6) Parçanın boyutunu uygun olmayan otoklava atanmasını engeller.
- (7) Yüklerin sırayla atanmasını sağlar
- (8) Otoklava atanan bir yükte birden fazla kür grubunun olmasını engeller.
- (9) Otoklavdaki yüke farklı kür grubundan parçalar atanmasını engeller.
- (10) Otoklava atanmayan yükün çizelgelenmesini engeller.
- (11) Parçanın tüm talebinin son Z zamanına kadar karşılanmasını sağlar.
- (12) Otoklavda aynı zaman diliminde birden fazla yükün çizelgelenmesini engeller.
- (13) Otoklavın kapalı olduğu zamanlarda yük çizelgelenmesini engeller.
- (14) Bir otoklavda z zamanında bir yük başlatıldıysa, bu yükün çevrim süresi boyunca atanan otoklavda başka yükün başlatılmasını engeller.
- (15) Takımın çevrim süresi boyunca kullanılmasını engeller.
- (16) z zamanında i yükü o otoklavına yüklendiye, p parçasının z zamanında yüklenen o otoklavındaki i yükünün miktarının p parçasının o otoklavındaki i yükündeki miktarına eşit olmasını sağlar
- (17) Yük, otoklava çizelgelenmediyse miktarın 0 olmasını sağlar.
- (18) Çizelgelenen miktar 0 ise yükün otoklava çizelgelenmesini engeller.
- (19) Parçanın otoklavda talep tarihinden çevrim süresi kadar önce başlatılmasını sağlar.

3.4. UYGULAMA

Bölüm 3.1, 3.2 ve 3.3'de oluşturulan matematiksel modeller Python programlama dili ve IBM Decision Optimization Modeling for Python (DOcplex) kütüphanesi kullanarak modellenmiş ve IBM ILOG Cplex yazılımı kullanılarak çözülmüştür. Oluşturulan modellerin Python kaynak kodları Ek.1, Ek.2 ve Ek.3'de görülebilir.

Python, Guido van Rossum tarafından geliştirilmiş, ilk sürümü 1991 yılında ortaya çıkmış bir yazılım geliştirme dilidir. Yorumlanan ve dinamik bir dil olarak öne çıkan Python, prosedürel ve nesne tabanlı programlama yaklaşımlarını desteklemektedir.

Günümüzde Python, kolay öğrenilmesi, kolay kullanılması ve çeşitli bilim dallarında kullanılmak üzere özel hazırlanmış kütüphanelere sahip olması sayesinde özellikle bilimsel araştırmalarda oldukça popüler hale gelmiş ve yaygın bir kullanıcı kitlesine ulaşmıştır.

Bu tez çalışmasında da kolay öğrenilmesi, kolay kullanılması ve çok sayıda hazır kütüphaneye sahip olması nedeni ile Python dili tercih edilmiştir.

Doğrusal tamsayı programlarını çözmek için birçok optimizasyon yazılımı mevcuttur. Bunlar arasından üçü; IBM ILOG Cplex, Gurobi ve GLPK öne çıkmaktadır. IBM Cplex ve Gurobi ticari paket yazılım, GLPK ise açık kaynak kodlu ücretsiz yazılımlardır. Bu tez çalışmasında IBM ILOG Cplex optimizasyon paketi kullanılmıştır.

Python yazılım geliştirme dili ve IBM ILOG Cplex optimizasyon paketi seçimi sonucu olarak matematiksel modelleri kodlamak için DOcplex Python kütüphanesi kullanılmıştır.

Firmada farklı boyutlarda 7 otoklav fırını bulunmaktadır, bu fırınların boyutları ve thermocouple sayıları aşağıda Tablo – 5'de gösterilmiştir.

Otoklav Numarası	Uzunluk (m)	Çap (m)	Alan (m ²)	Termocouple Sayısı
1	14	3,6	50,4	74
2	4	2	8,0	15
3	12	4	48,0	71
4	1,5	0,6	0,9	12
5	14	4	56,0	120
6	7	3,5	24,5	96
7	2	0,9	1,8	20

Tablo - 6 Otoklav Kapasiteleri

Tablo – 6’da gösterilen otoklavları yüklemek ve çizelgelemek üzere 16 ayrı veri seti oluşturulmuş ve farklı senaryolar altında çözülmüştür.

Parçaların kür sürelerinin 4 ve katlarına yakın şekilde olduğu görüldüğü için zaman birimi 4 saat olarak alınmıştır.

Tez çalışmasında matematiksel modeller Intel Core i7 8650 CPU @ 1.90Ghz-2.11 Ghz, 8 GB RAM’e sahip Windows 10 64 bit işletim sistemi kurulu Dell Latitude 7490 dizüstü bilgisayar üzerinde çözülmüştür.

Çalışma kapsamında 5 senaryo oluşturulmuş, toplam 120 model çalıştırılmış ve sonuçları incelenmiştir. Çalışma planı detayları Şekil-’de görülmektedir.

Otoklav Yükleme ve Çizelgeleme Problemi

1

Otoklav Sayısının Etkisi

Model Sayısı	105		
Çözüm Süresi	12.4 saat		
Bağımsız Modeller- Yükleme		Bağımsız Modeller-Çizelgeleme	
Otoklav Sayısı	1 – 7	Otoklav Sayısı	1 – 7
Takım Sayısı	1 – 5	Takım Sayısı	1 – 5
Model Sayısı	35	Model Sayısı	35
Değişken Sayısı	118-778	Değişken Sayısı	224 - 1520
Kısıt Sayısı	347-2381	Kısıt Sayısı	474-2982
Çözüm Süresi	0.02 –1.81	Çözüm Süresi	0.02 –0.06

Tek Model

Otoklav Sayısı	1 – 7
Takım Sayısı	1 – 5
Model Sayısı	35
Değişken Sayısı	388-2668
Kısıt Sayısı	696-4536
Çözüm Süresi	1.88-7229

2

Takım Sayısının Etkisi

Model Sayısı	15		
Çözüm Süresi	2 saat		
Bağımsız Modeller-Yükleme		Bağımsız Modeller-Çizelgeleme	
Otoklav Sayısı	7	Otoklav Sayısı	7
Takım Sayısı	1 – 5	Takım Sayısı	1 – 5
Model Sayısı	5	Model Sayısı	5
Değişken Sayısı	778	Değişken Sayısı	1520
Kısıt Sayısı	2381	Kısıt Sayısı	2982
Çözüm Süresi	0.08-1.25	Çözüm Süresi	0.01-0.03

Tek Model

Otoklav Sayısı	7
Takım Sayısı	1 – 5
Model Sayısı	5
Değişken Sayısı	2668
Kısıt Sayısı	4536
Çözüm Süresi	4 - 7220

Şekil – 14 Çalışma Planı

Otoklav Yükleme ve Çizelgeleme Problemi

3

Bağımsız ve Tek Model Karşılaştırması

Model Sayısı		15	
Çözüm Süresi		2 saat	
Bağımsız Modeller-Yükleme		Bağımsız Modeller-Çizelgeleme	
Otoklav Sayısı	7	Otoklav Sayısı	7
Takım Sayısı	1 – 5	Takım Sayısı	1 – 5
Model Sayısı	5	Model Sayısı	5
Değişken Sayısı	778	Değişken Sayısı	1520
Kısıt Sayısı	2381	Kısıt Sayısı	2982
Çözüm Süresi	0.08-1.25	Çözüm Süresi	0.01-0.03

Tek Model

Otoklav Sayısı	7
Takım Sayısı	1 – 5
Model Sayısı	5
Değişken Sayısı	2668
Kısıt Sayısı	4536
Çözüm Süresi	4 - 7220

4

Farklı Günler

Otoklav Sayısı	7
Takım Sayısı	1
Model Sayısı	14
Değişken Sayısı	2.352-10.582
Kısıt Sayısı	3.829-19.229
Çözüm Süresi	1475

5

Geniş Planlama Ufku

Otoklav Sayısı	7
Takım Sayısı	1
Model Sayısı	1
Değişken Sayısı	99.553
Kısıt Sayısı	121.554
Çözüm Süresi	36740

Şekil –14 Çalışma Planı (Devam)

3.5. OTOKLAV SAYISININ ETKİSİ

Firmada çeşitli boyutlarda 7 adet otoklav bulunmaktadır. Otoklavlar bu tez çalışmasında numaralandırıldığı sıra ile satın alınmış ve kullanılmaya başlanmıştır, yani 1 numaralı otoklav firmanın ilk otoklavı iken 7 numaralı otoklav sonuncusudur. Gerçek veriler arasından bir örneklem seçilmiş ve otoklav yükleme ve çizelgelemenin bağımsız olduğu ve beraber ele alındığı matematiksel modeller bu örnek veri seti ile otoklavların firmada kullanıma alınış sırasına göre arttırılarak ve her otoklav eklendiğinde takım sayısı 1'den 5'e kadar artacak şekilde 35 kez çalıştırılmıştır. Örnek veri seti içerisinde 2 ayrı kür grubuna ait 8 parça için toplam 1 günlük 52 adet talep bulunmaktadır. Talep miktarları takım sayısının etkisini gösterebilmek için gerçek değerlerine göre arttırılmıştır.

3.5.1 Otoklav Yükleme ve Çizelgeleme Bağımsız Modeller

Otoklav yükleme ve çizelgeleme matematiksel modellerinin birbirinden bağımsız olarak çalıştırıldığı durumda alan kullanımı ve karşılanan talep Tablo-7'de görülmektedir.

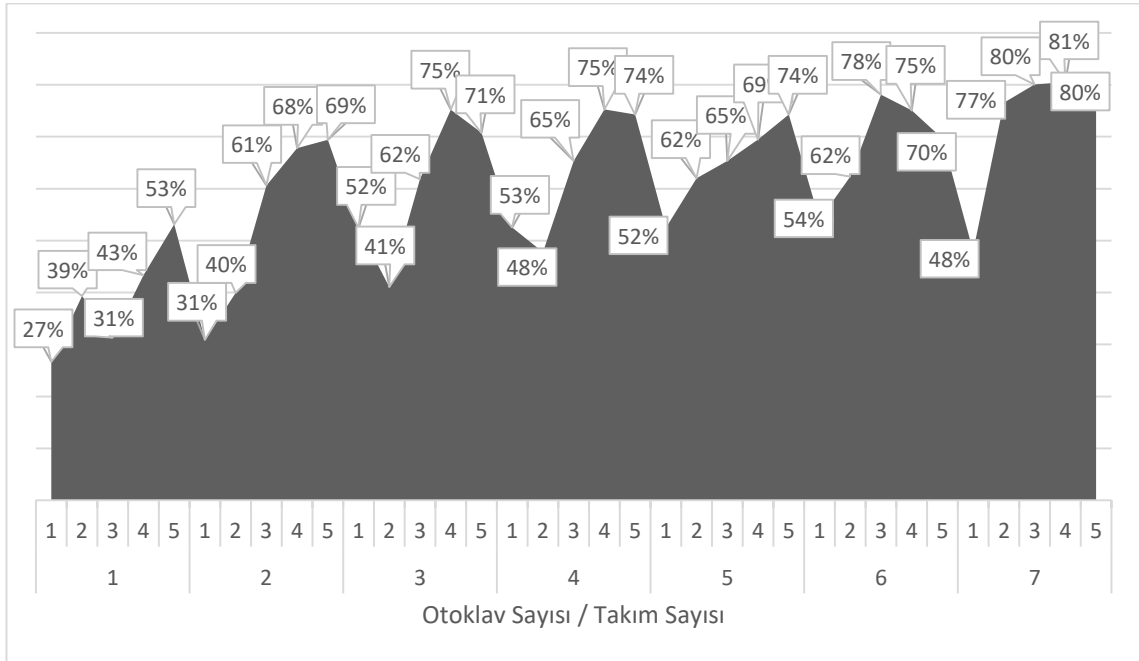
Otoklav Sayısı	Takım Sayısı	Alan Kullanımı	Karşılanan Talep
1	1	27%	10
1	2	39%	26
1	3	31%	28
1	4	43%	32
1	5	53%	43
2	1	31%	16
2	2	40%	26
2	3	61%	33
2	4	68%	36
2	5	69%	36
3	1	52%	14

Tablo – 7 Bağımsız Modeller Alan Kullanımı ve Talep Karşılama

Otoklav Sayısı	Takım Sayısı	Alan Kullanımı	Karşılanan Talep
3	2	41%	26
3	3	62%	33
3	4	75%	40
3	5	71%	36
4	1	53%	14
4	2	48%	28
4	3	65%	35
4	4	75%	40
4	5	74%	48
5	1	52%	14
5	2	62%	30
5	3	65%	35
5	4	69%	40
5	5	74%	48
6	1	54%	14
6	2	62%	26
6	3	78%	32
6	4	75%	40
6	5	70%	52
7	1	48%	13
7	2	77%	28
7	3	80%	36
7	4	81%	40
7	5	80%	42

Tablo – 7 Bağımsız Modeller Alan Kullanımı ve Talep Karşılama (Devam)

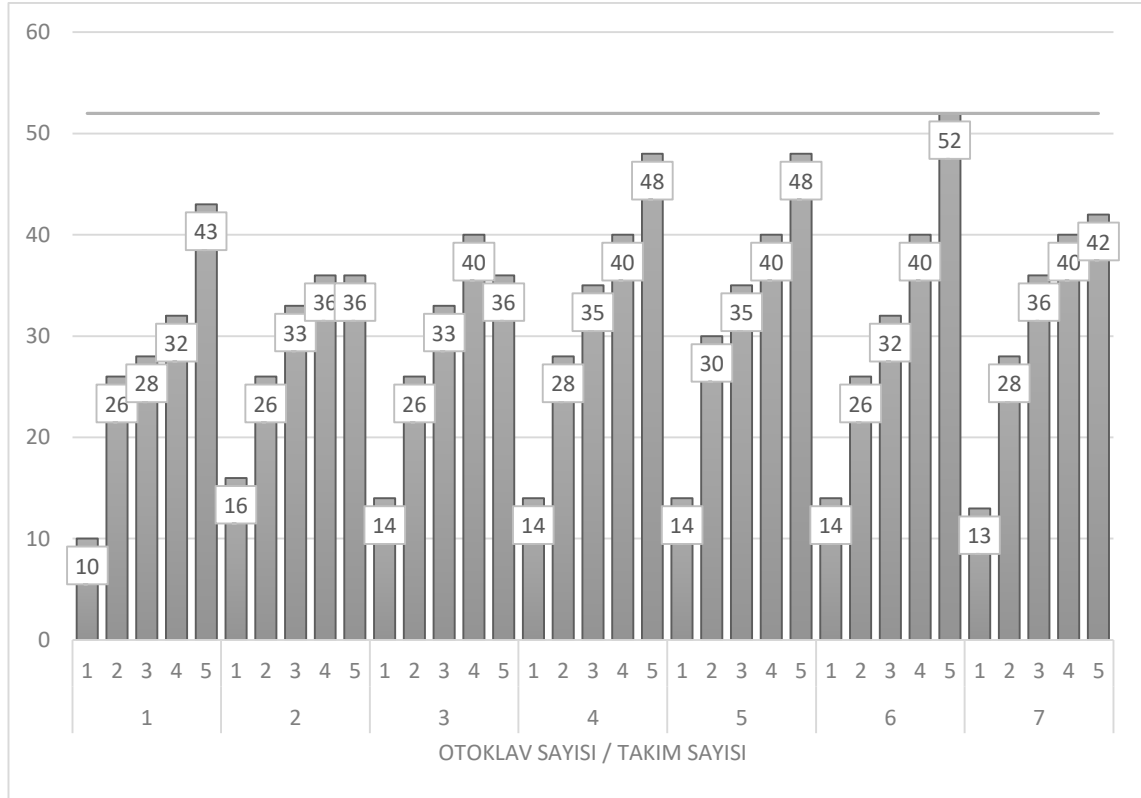
Şekil – 15’de görüldüğü üzere alan kullanımı hem yeni otoklavlar devreye girdikçe hem de aynı sayıda otoklav için yeni takımların kullanımına imkan verilince artmıştır. Bir otoklav bir takım mevcut olduğu durumda alan kullanımı %27 iken, 7 otoklavın kullanımı ve 5 takım olması durumunda %80 kullanım oranına ulaşılmıştır.



Şekil - 15 Bağımsız Modeller - Alan Kullanımı (Otoklav Sayısı)

Şekil – 16 kullanıma alınan otoklav sayısını ve takım sayısının arttırdığımız zaman karşılanan talep miktarının değişimini göstermektedir. Takım sayısını arttırmak gittikçe azalan bir oranda karşılanan talep miktarını arttırmaktadır.

Şekil – 16’da da görüleceği üzere takım sayısını arttırmanın karşılanan talep miktarını arttırmakta otoklav sayısından daha etkili olduğu görünmektedir.

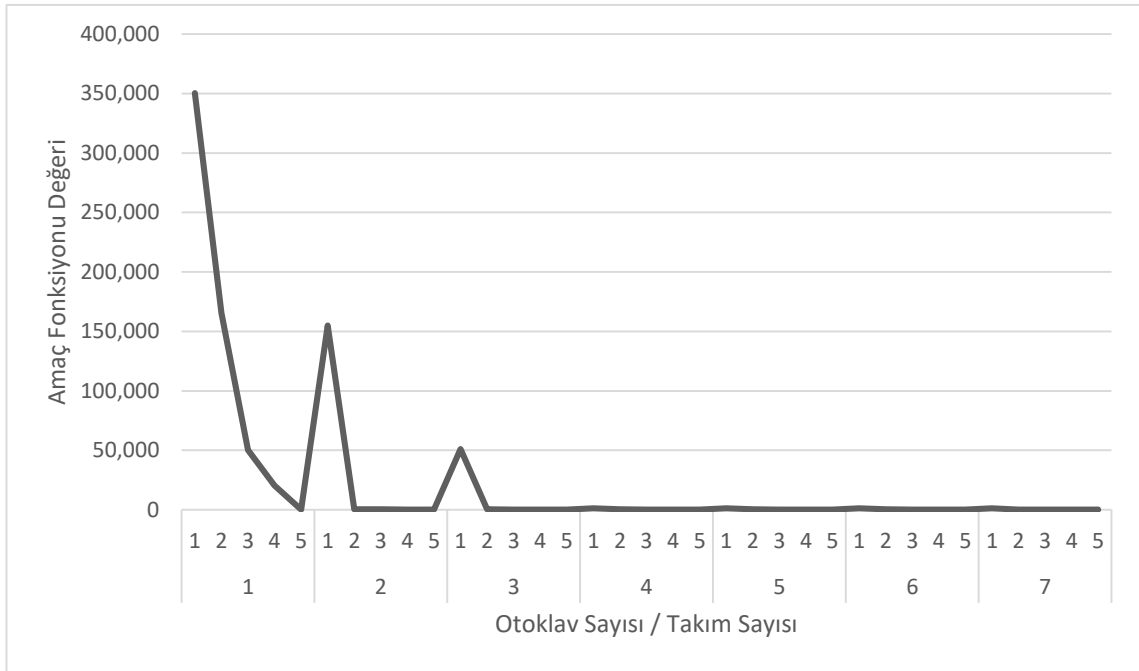


Şekil – 16 Bağımsız Modeller - Talep Karşılama (Otoklav Sayısı)

Otoklav yükleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + 1000 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right)$$

Karşılanmayan talebin minimize edilmesine çok büyük (1.000) ağırlık verilerek yükler oluşturulurken tüm talebin karşılanması sağlanmıştır. Şekil-17'de otoklav sayısı ve takım sayısı ekseninde yükleme amaç fonksiyonu değerleri görülmektedir, otoklav sayısı ve takım sayısı arttıkça amaç fonksiyonu değeri azalmaktadır. Takım sayısı arttırıldıkça aynı anda aynı parçadan birden fazla adet otoklava yüklenebilmesi imkanı doğmakta bu da amaç fonksiyonu değerinin azalmasını sağlamaktadır. Yeni otoklavların devreye alınması diğer otoklavlara sığmayan parçaların kürlenmesi imkanını sağlamakta bu durum da amaç fonksiyonu değerinin azalması yönünde etkilemektedir.



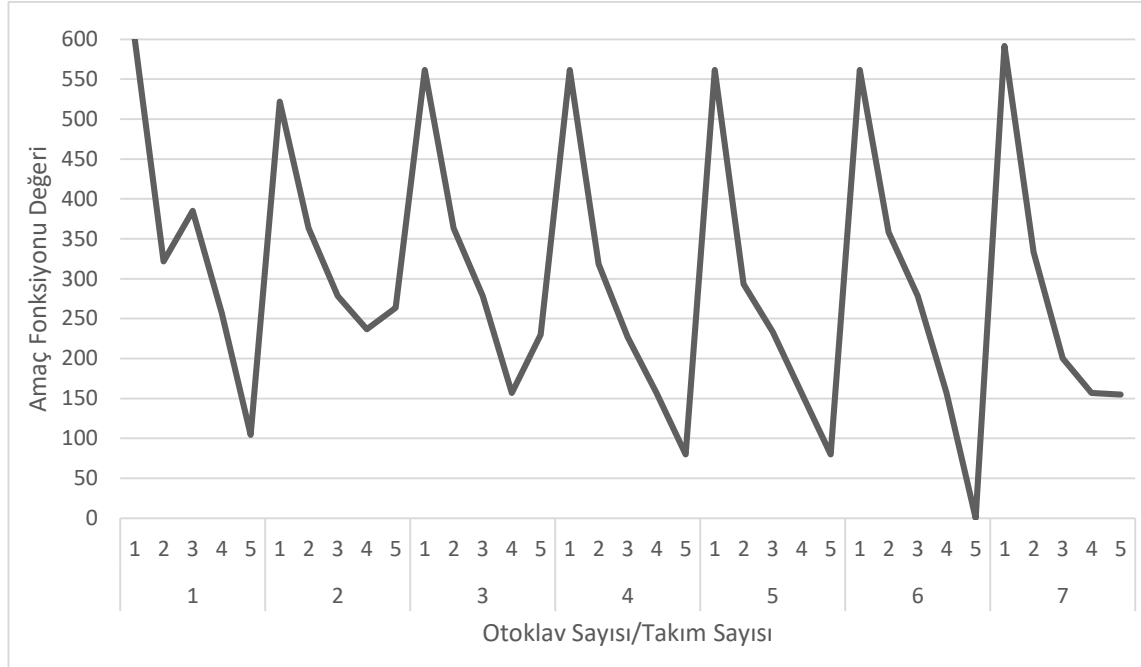
Şekil - 17 Bağımsız Modeller – Yükleme Amaç Fonksiyonu (Otoklav Sayısı)

Otoklav çizelgeleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}}{D_{p,z}} \right)$$

Şekil-18'de otoklav sayısı ve takım sayısı ekseninde çizelgeleme amaç fonksiyonu değerleri görülmektedir, otoklav sayısı ve takım sayısı arttıkça amaç fonksiyonu değeri azalmaktadır. Normalde beklenen, otoklav sayısı arttıkça zamanında karşılanmayan talebin ve toplam karşılanmayan talebin azalması, dolayısıyla da amaç fonksiyonu değerinin azalmasıdır. Yalnız yükleme matematiksel modeli çizelgeleme kısıtlarını içermediği için toplam talebi karşılayacak şekilde kullanılmayan alanı minimize etmeye çalışmış, bu modelden çıkan yüklerin çizelgelenmesi beklenen sonuç ile uyumlu olmamıştır. Şekil-18'de görüldüğü gibi otoklav sayısının artması amaç fonksiyonu değerinde her

durumda bir azalmaya yol açmamıştır. Takım sayısının artırılması ise çizelgeleme amaç fonksiyonu değerini azaltmıştır.



Şekil - 18 Bağımsız Modeller – Çizelgeleme Amaç Fonksiyonu (Otoklav Sayısı)

Şekil – 19 mevcut otoklav sayısı ve takım sayısına göre optimum çözümlerde otoklavların kullanım durumunu göstermektedir. Dört otoklava kadar tüm otoklavlar kullanılırken, beşinci otoklavdan sonra hem yükleme hem de çizelgeleme amaç fonksiyonlarında azalma olmasına rağmen (Şekil-17, Şekil-18) kullanılan otoklav sayısı beşi aşmamıştır.

Otoklav	Otoklav Sayısı / Takım Sayısı																																					
	1					2					3					4					5					6					7							
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5			
1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
3	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
4	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
6	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
7	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Şekil - 19 Bağımsız Modeller – Otoklav Kullanım Durumu

Tablo-8 yükleme matematiksel modeli ile ilgili bilgileri içermektedir. Otoklav sayısı arttıkça kısıt sayısı 347 adetten 2,381 adete çıkmakta değişken sayısı ise 118 adetten 778 adete çıkmaktadır. Çözüm süreleri kısıt ve değişken sayısının yanında takım sayısı kısıtının etkisi ile de değişkenlik göstermektedir. En düşük çözüm süresi 0.02 saniye iken en yüksek çözüm süresi ise 22 dakikadır.

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	1	350,370.93	0.02	347	118
1	2	165,359.80	0.03	347	118
1	3	50,325.89	0.02	347	118
1	4	20,323.97	0.03	347	118
1	5	223.85	0.03	347	118
2	1	154,861.97	0.03	686	228
2	2	481.66	0.13	686	228
2	3	284.71	0.33	686	228
2	4	166.84	0.22	686	228
2	5	160.53	0.22	686	228
3	1	50,957.60	0.05	1,025	338
3	2	476.08	0.16	1,025	338
3	3	279.10	0.34	1,025	338
3	4	161.40	1.00	1,025	338
3	5	155.15	0.42	1,025	338
4	1	1,361.05	0.05	1,364	448
4	2	424.53	0.25	1,364	448
4	3	216.29	0.14	1,364	448
4	4	161.17	1.14	1,364	448
4	5	154.86	0.84	1,364	448

Tablo – 8 Bağımsız Modeller Yükleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı)

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
5	1	1,358.51	0.17	1,703	558
5	2	424.53	0.31	1,703	558
5	3	216.29	0.23	1,703	558
5	4	159.57	1.81	1,703	558
5	5	154.86	1.36	1,703	558
6	1	1,280.99	0.08	2,042	668
6	2	323.27	0.33	2,042	668
6	3	168.12	0.55	2,042	668
6	4	137.22	0.53	2,042	668
6	5	121.46	0.63	2,042	668
7	1	1,164.23	0.08	2,381	778
7	2	271.84	0.45	2,381	778
7	3	162.10	0.64	2,381	778
7	4	134.96	1.25	2,381	778
7	5	121.42	1.02	2,381	778

Tablo – 8 Bağımsız Modeller Yükleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı) (Devam)

Tablo-9 çizelgeleme matematiksel modeli ile ilgili bilgileri içermektedir. Otoklav sayısı arttıkça kısıt sayısı 474 adetten 2982 adete çıkmakta değişken sayısı ise 224 adetten 1520 adete çıkmaktadır. Çözüm süreleri kısıt ve değişken sayısına göre önemli bir değişkenlik göstermektedir. Çözüm süresi ortalama 0.03 saniyedir.

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	1	600.00	0.02	474	224
1	2	321.67	0.02	474	224
1	3	385.00	0.02	474	224
1	4	256.67	0.02	474	224
1	5	104.17	0.02	474	224
2	1	521.67	0.02	892	440
2	2	363.33	0.03	892	440
2	3	278.33	0.03	892	440
2	4	236.67	0.02	892	440
2	5	263.33	0.03	892	440
3	1	561.67	0.02	1,310	656
3	2	363.33	0.02	1,310	656
3	3	278.33	0.03	1,310	656
3	4	156.67	0.03	1,310	656
3	5	230.00	0.06	1,310	656
4	1	561.67	0.02	1,728	872
4	2	318.33	0.03	1,728	872
4	3	226.67	0.02	1,728	872
4	4	156.67	0.06	1,728	872
4	5	80.00	0.03	1,728	872
5	1	561.67	0.03	2,146	1,088
5	2	293.33	0.03	2,146	1,088
5	3	233.33	0.02	2,146	1,088
5	4	156.67	0.02	2,146	1,088

Tablo – 9 Bağımsız Modeller Çizelgeleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı)

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
5	5	80.00	0.03	2,146	1,088
6	1	561.67	0.03	2,564	1,304
6	2	358.33	0.03	2,564	1,304
6	3	278.33	0.03	2,564	1,304
6	4	156.67	0.02	2,564	1,304
6	5	0	0.03	2,564	1,304
7	1	591.67	0.03	2,982	1,520
7	2	333.33	0.03	2,982	1,520
7	3	200.00	0.03	2,982	1,520
7	4	156.67	0.02	2,982	1,520
7	5	155.00	0.03	2,982	1,520

Tablo – 9 Bağımsız Modeller Çizelgeleme Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı) (Devam)

Bağımsız modeller kullanılması ile firma elindeki talebi 6 otoklav ile karşılayabilmektedir (Şekil-16). Otoklav yatırımına karar verirken bağımsız modellerin çalıştırılması ile firma 1 milyon ABD doları civarında bir yatırımı yapmama kararı verebilecektir.

3.5.2 Otoklav Yükleme ve Çizelgeleme Tek Model

Otoklav yükleme ve çizelgelemenin tek matematiksel model olarak çalıştırıldığı durumda alan kullanımı ve karşılanan talep Tablo-10'da görülmektedir.

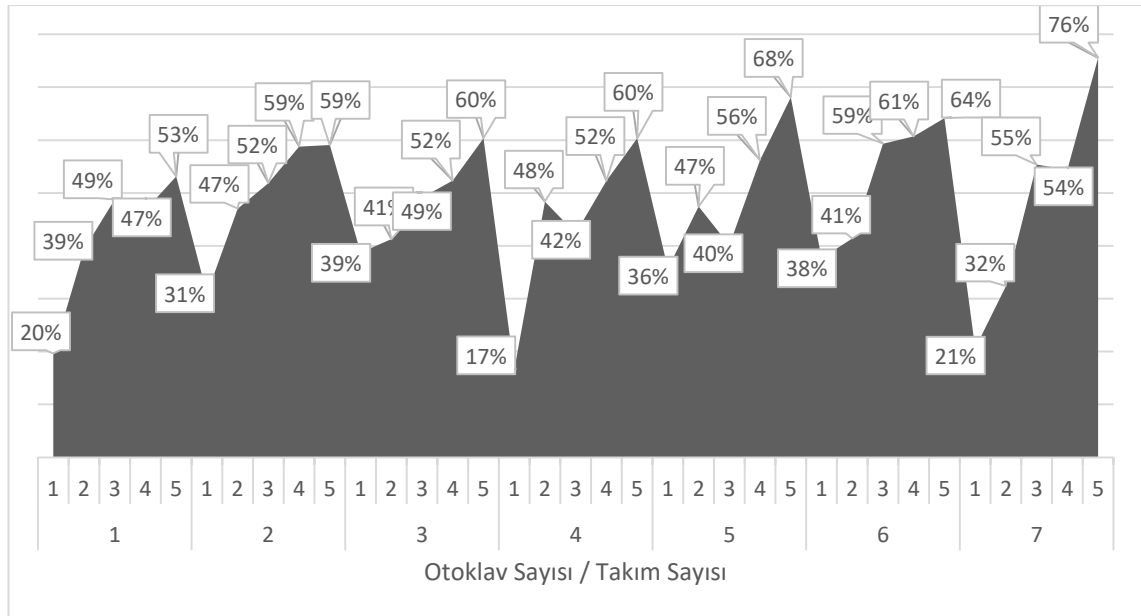
Otoklav Sayısı	Takım Sayısı	Alan Kullanımı	Karşılanan Talep
1	1	20%	13
1	2	39%	26
1	3	49%	33
1	4	47%	41
1	5	53%	43
2	1	31%	16
2	2	47%	32
2	3	52%	42
2	4	59%	48
2	5	59%	52
3	1	39%	16
3	2	41%	32
3	3	49%	42
3	4	52%	48
3	5	60%	52
4	1	17%	16
4	2	48%	32
4	3	42%	42
4	4	52%	48
4	5	60%	52
5	1	36%	16
5	2	47%	32
5	3	40%	42
5	4	56%	48
5	5	68%	52

Tablo – 10 Tek Model Alan Kullanımı ve Talep Karşılama

Otoklav Sayısı	Takım Sayısı	Alan Kullanımı	Karşılanan Talep
6	1	38%	16
6	2	41%	32
6	3	59%	42
6	4	61%	48
6	5	64%	52
7	1	21%	16
7	2	32%	32
7	3	55%	42
7	4	54%	48
7	5	76%	52

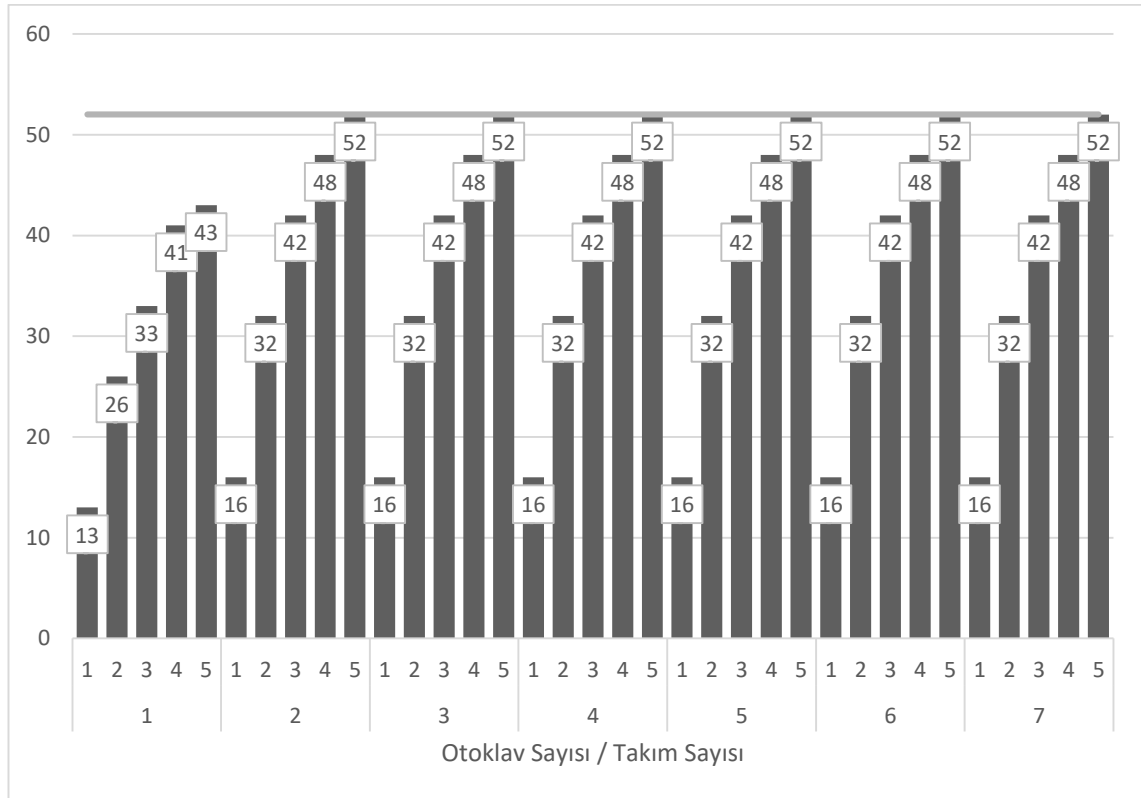
Tablo – 10 Tek Model Alan Kullanımı ve Talep Karşılama (Devam)

Şekil-20’de görüldüğü üzere otoklav sayısından bağımsız olarak kullanılabilir takım sayısı arttıkça alan kullanımı da artmaktadır. Otoklav sayısının artışı ile alan kullanımı arasında anlamlı bir ilişki görülmemektedir.



Şekil - 20 Tek Model - Alan Kullanımı (Otoklav Sayısı)

Şekil-21 kullanıma alınan otoklav sayısını ve takım sayısının arttırdığımız zaman karşılanan talep miktarının değişimini göstermektedir. Takım sayısını arttırmak gittikçe azalan bir oranda karşılanan talep miktarını arttırmaktadır. Şekil-21’de de görüleceği üzere takım sayısını arttırmanın karşılanan talep miktarını arttırmakta otoklav sayısından daha etkili olduğu görünmektedir.



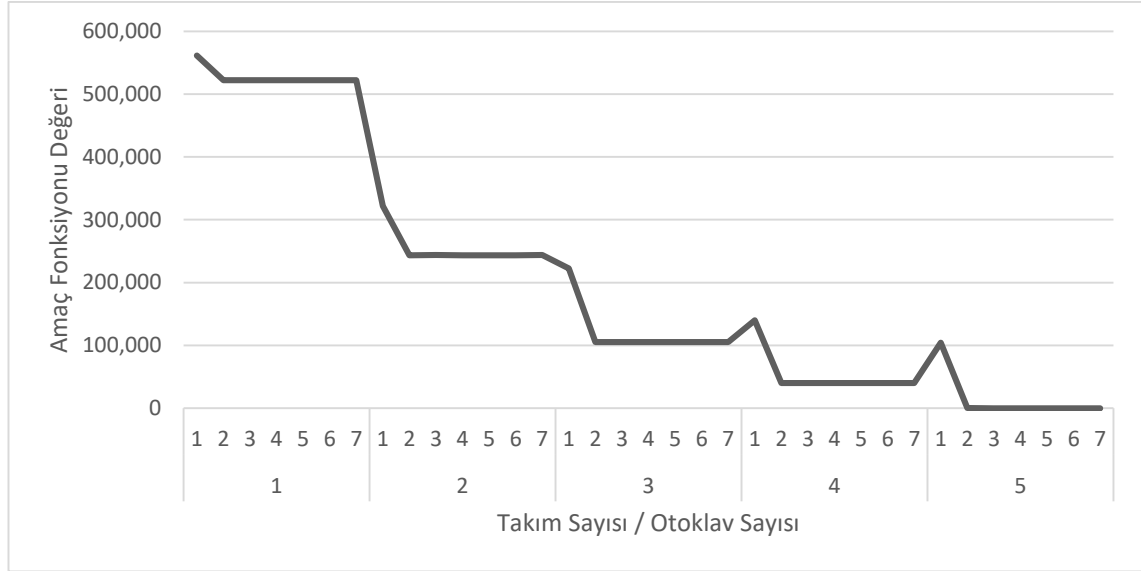
Şekil - 21 Tek Model - Talep Karşılama (Otoklav Sayısı)

Otoklav yükleme çizelgeleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + 1000 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + 1000 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}}{D_{p,z}} \right)$$

Şekil-22’de takım ve otoklav sayısı ekseninde amaç fonksiyonu değerleri görülmektedir, takım sayısı arttıkça amaç fonksiyonu değeri de azalan bir oranda

azalmaktadır. Otoklav sayısını arttırmanın etkisi takım sayısını arttırmaya oranla çok daha az görünmektedir.



Şekil - 22 Tek Model –Amaç Fonksiyonu (Otoklav Sayısı)

Şekil-23 mevcut otoklav sayısı ve takım sayısına göre optimum çözümlerde otoklavların kullanım durumunu göstermektedir. Üç otoklava kadar tüm otoklavlar kullanılırken, dördüncü otoklavdan sonra amaç fonksiyonunda azalma olmasına rağmen (Şekil-22) kullanılan otoklav sayısı beşi aşmamıştır.

Otoklav	Otoklav Sayısı / Takım Sayısı																																						
	1					2					3					4					5					6					7								
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5				
1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■		
2																																							
3																																							
4																																							
5																																							
6																																							
7																																							

Şekil - 23 Tek Model – Otoklav Kullanım Durumu

Tablo-11 matematiksel model ile ilgili bilgileri içermektedir. Otoklav sayısı arttıkça kısıt sayısı 696 adetten 4,536 adete çıkmakta değişken sayısı ise 388 adetten 2668 adete çıkmaktadır. Çözüm süreleri kısıt ve değişken sayısının yanında

takım sayısı kısıtının etkisi ile de değişkenlik göstermektedir. En düşük çözüm süresi 2 saniye iken en yüksek çözüm süresi ise 2 saattir.

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	1	561,074.38	21.84	696	388
1	2	321,848.77	44.42	696	388
1	3	222,653.71	39.50	696	388
1	4	140,159.68	187.13	696	388
1	5	104,307.24	216.77	696	388
2	1	521,954.39	2.34	1,336	768
2	2	243,546.41	1.88	1,336	768
2	3	105,290.83	7.34	1,336	768
2	4	40,264.79	514.75	1,336	768
2	5	246.23	7,200.8	1,336	768
3	1	522,152.40	11.89 ⁴	1,976	1,148
3	2	243,721.43	4.77	1,976	1,148
3	3	105,285.04	3.67	1,976	1,148
3	4	40,256.75	8.50	1,976	1,148
3	5	199.69	7,209.4	1,976	1,148
4	1	522,020.23	31.34 ⁴	2,616	1,528
4	2	243,550.27	2.88	2,616	1,528
4	3	105,334.03	9.06	2,616	1,528
4	4	40,248.96	4.16	2,616	1,528
4	5	198.55	7,220.2	2,616	1,528
5	1	521,994.88	52.48 ³	3,256	1,908
5	2	243,558.77	3.91	3,256	1,908

Tablo – 11 Tek Model - Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı)

Otoklav Sayısı	Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
5	3	105,252.03	13.22	3,256	1,908
5	4	40,300.00	12.34	3,256	1,908
5	5	191.53	7,215.3	3,256	1,908
6	1	522,054.18	5.59	3,896	2,288
6	2	243,709.20	6.13	3,896	2,288
6	3	105,214.72	10.64	3,896	2,288
6	4	40,301.08	20.50	3,896	2,288
6	5	143.93	7,229.3	3,896	2,288
7	1	522,096.37	4.00	4,536	2,668
7	2	243,726.43	16.67	4,536	2,668
7	3	105,223.21	11.19	4,536	2,668
7	4	40,290.45	43.36	4,536	2,668
7	5	147.23	7,220.1	4,536	2,668

Tablo – 11 Tek Model - Model ve Çözüm Sayısal Değerleri (Otoklav Sayısı)
(Devam)

Tek model kullanılması ile firma elindeki talebi 2 otoklav ile karşılayabilmektedir (Şekil-21). Otoklav yatırımına karar verirken tek modelin çalıştırılması ile firma 5 milyon ABD doları civarında bir yatırımı yapmama kararı verebilecektir.

3.6. TAKIM SAYISININ ETKİSİ

Kompozit parçalar serim aşamasında temiz odada özel takımlar üzerine yerleştirilmekte ve bu takımlar ile otoklava girmektedirler. Takımlar otoklavdan çıktıktan sonra kompozit parçalar üzerlerinden sökülümekte ve tekrar kullanıma hazır hale getirmek için temizlenmektedir. Her parça için özel olan bu takımların sayısı otoklavlara aynı anda yüklenebilecek parça sayısını da etkilemektedir. Örneğin, 1 adet takıma sahip olan bir parçadan 7 adet ihtiyaç varsa, 7 otoklav da boş olsa sadece aynı anda 1 adet parça kürlenmek üzere otoklava

girebilmektedir. Firmadaki her bir parça için genellikle 1 adet takım bulunmaktadır. Takımlar pahalı ekipmanlar olduğu için sayılarını arttırmak maliyeti arttırmaktadır, bu nedenle en uygun takım sayısını bulmak önemli hale gelmektedir.

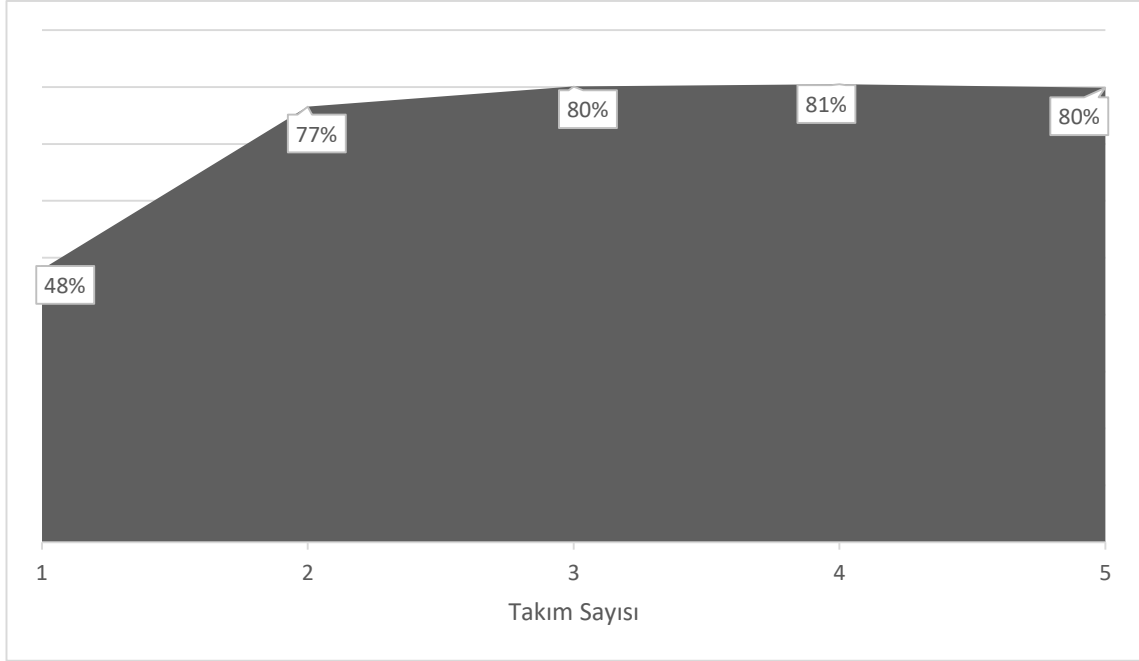
Gerçek veriler arasından bir örneklem seçilmiş ve otoklav yükleme ve çizelgelemenin bağımsız olduğu ve beraber ele alındığı matematiksel modeller bu örnek veri seti ile takım sayısı 1'den 5'e kadar artacak şekilde 5 kez çalıştırılmıştır. Örnek veri seti içerisinde 2 ayrı kür grubuna ait 8 parça için toplam 1 günlük 52 adet talep bulunmaktadır. Talep miktarları takım sayısının etkisini gösterebilmek için gerçek değerlerine göre arttırılmıştır.

3.6.1 Otoklav Yükleme ve Çizelgeleme Bağımsız Modeller

Otoklav yükleme ve çizelgeleme matematiksel modellerinin birbirinden bağımsız olarak çalıştırıldığı durumda Şekil-24 'de görüldüğü üzere alan kullanımı parçalar için 1 takım olduğu durumda %48 iken, 2 takım mevcut olduğunda %77'ye çıkmıştır. Her parça için mevcut takım sayısını 3'e çıkarmak alan kullanımını %3 daha arttırarak %80'e çıkarmış daha sonraki takım sayısındaki artışlar alan kullanımında anlamlı bir artış sağlamamıştır.

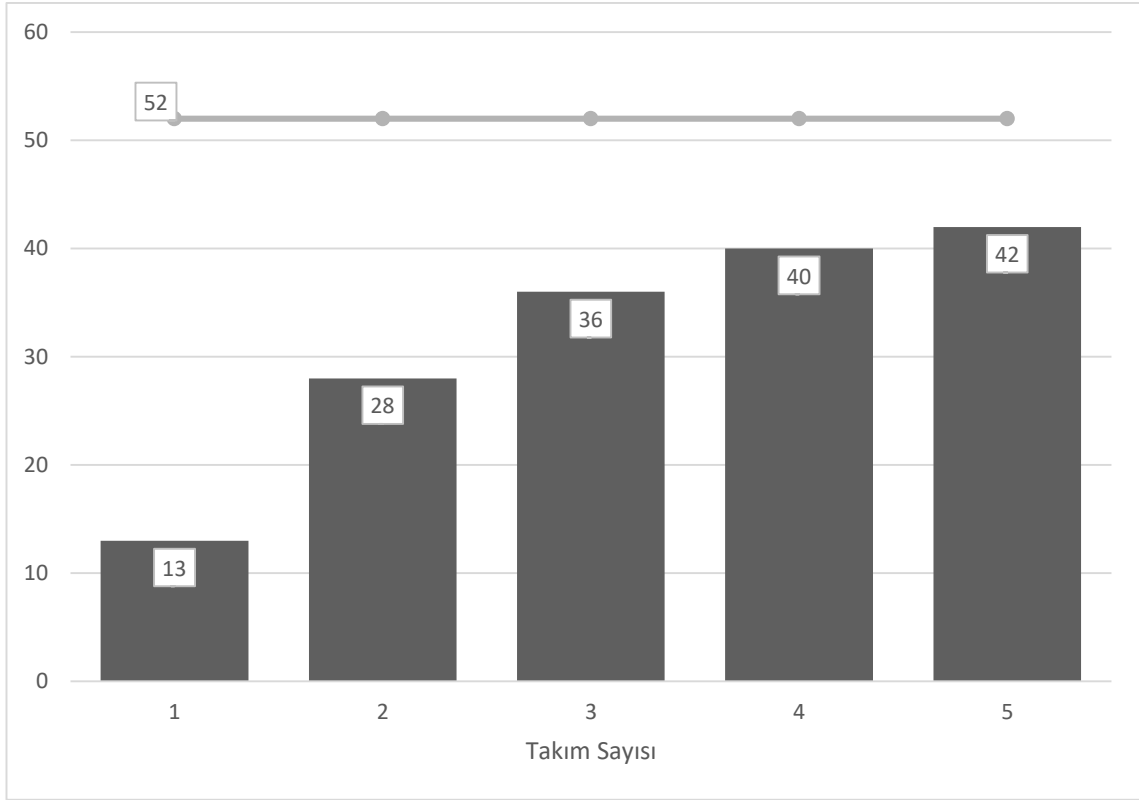
Takım Sayısı	Alan Kullanımı	Karşılanan Talep
1	48%	13
2	77%	28
3	80%	36
4	81%	40
5	80%	42

Tablo – 12 Bağımsız Modeller Alan Kullanımı ve Talep Karşılama



Şekil – 24 Bağımsız Modeller - Alan Kullanımı

Şekil-25 takım sayısının arttırdığımız zaman karşılanan talep miktarının nasıl değiştiğini göstermektedir. Takım sayısını arttırmak gittikçe azalan bir oranda karşılanan talep miktarını arttırmaktadır.

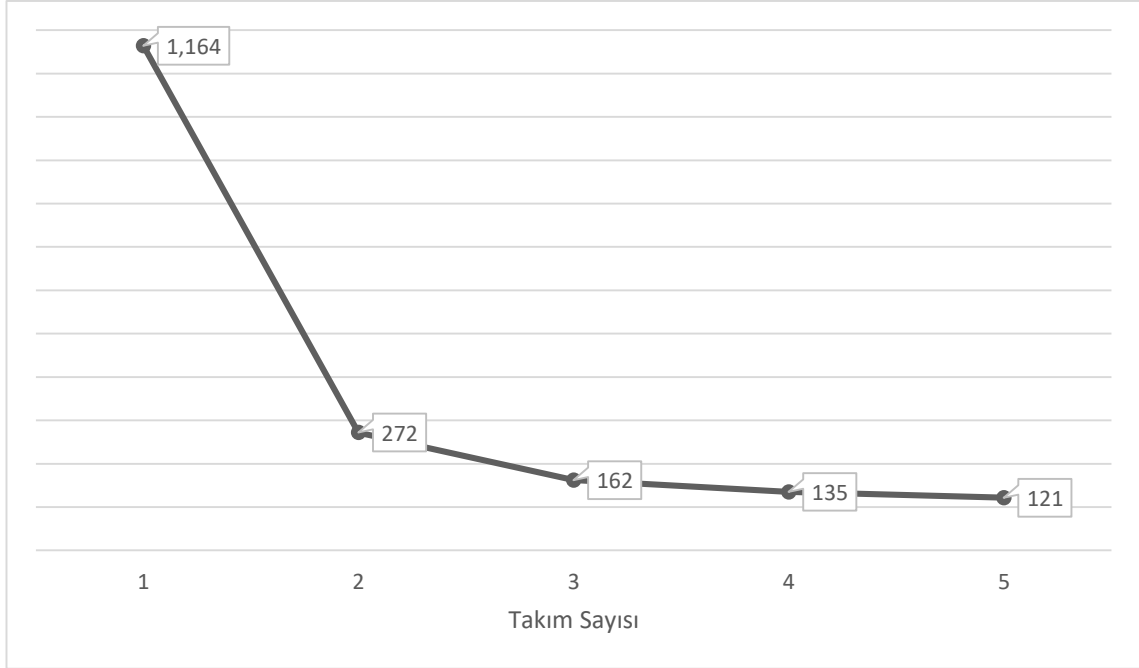


Şekil – 25 Bağımsız Modeller - Talep Karşılama

Otoklav yükleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + 1000 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right)$$

Karşılanmayan talebin minimize edilmesine çok büyük (1.000) ağırlık verilerek yükler oluşturulurken tüm talebin karşılanması sağlanmıştır. Şekil-26'da takım sayısı ekseninde amaç fonksiyonu değerleri görülmektedir, takım sayısı arttıkça amaç fonksiyonu değeri de azalan bir oranda azalmaktadır.

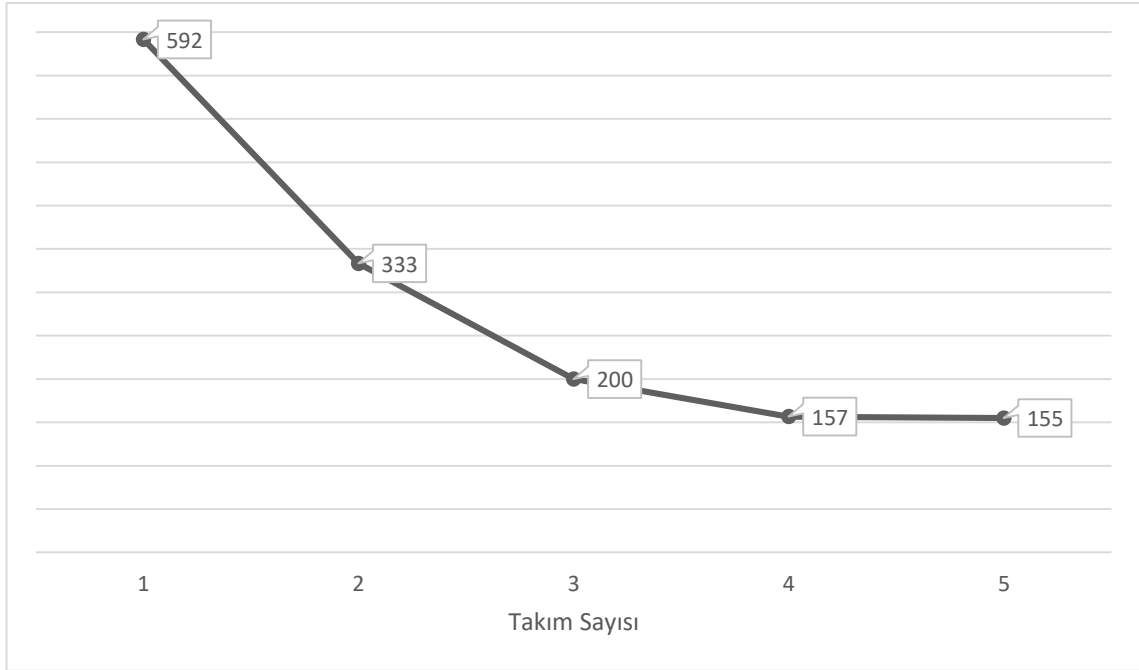


Şekil – 26 Bağımsız Modeller - Yükleme Amaç Fonksiyonu

Otoklav çizelgeleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}}{D_{p,z}} \right)$$

Şekil-27'de takım sayısı ekseninde amaç fonksiyonu değerleri görülmektedir, takım sayısı arttıkça amaç fonksiyonu değeri de azalan bir oranda azalmaktadır.



Şekil – 27 Bağımsız Modeller - Çizelgeleme Amaç Fonksiyonu

Otoklav yükleme ve çizelgeleme matematiksel modellerinin bağımsız çözüldüğü durumda takım sayısını arttırmanın hem alan kullanımı hem de karşılanan talepte azalan bir şekilde artış sağladığı gözlenmiştir.

Tablo-13 yükleme matematiksel modeli ile ilgili bilgileri içermektedir. Model 778 değişken ve 2381 kısıttan oluşmaktadır. Çözüm süresi modelde kullanılabilir olarak tanımlanan takım sayısına göre 0.078 saniye ve 1.25 saniye arasında değişiklik göstermektedir.

Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	1,164.23	0.078	2381	778
2	271.84	0.453	2381	778
3	162.10	0.641	2381	778
4	134.96	1.25	2381	778
5	121.42	1.016	2381	778

Tablo – 13 Bağımsız Modeller – Yükleme - Model ve Çözüm Sayısal Değerleri

Tablo-14 çizelgeleme matematiksel modeli ile ilgili bilgileri içermektedir. Model 1520 değişken ve 2982 kısıttan oluşmaktadır. Çözüm süresi modelde kullanılabilir olarak tanımlanan takım sayısına göre anlamlı değişiklik göstermektedir.

Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	591.67	0.032	2982	1520
2	333.33	0.032	2982	1520
3	200.00	0.031	2982	1520
4	156.67	0.015	2982	1520
5	155.00	0.031	2982	1520

Tablo – 14 Bağımsız Modeller – Çizelgeleme - Model ve Çözüm Sayısal Değerleri

Otoklav yükleme ve çizelgelemenin bağımsız modeller olarak çalıştırıldığı durumda oluşan çizelgelerin gantt şemaları Ek.6'da verilmiştir.

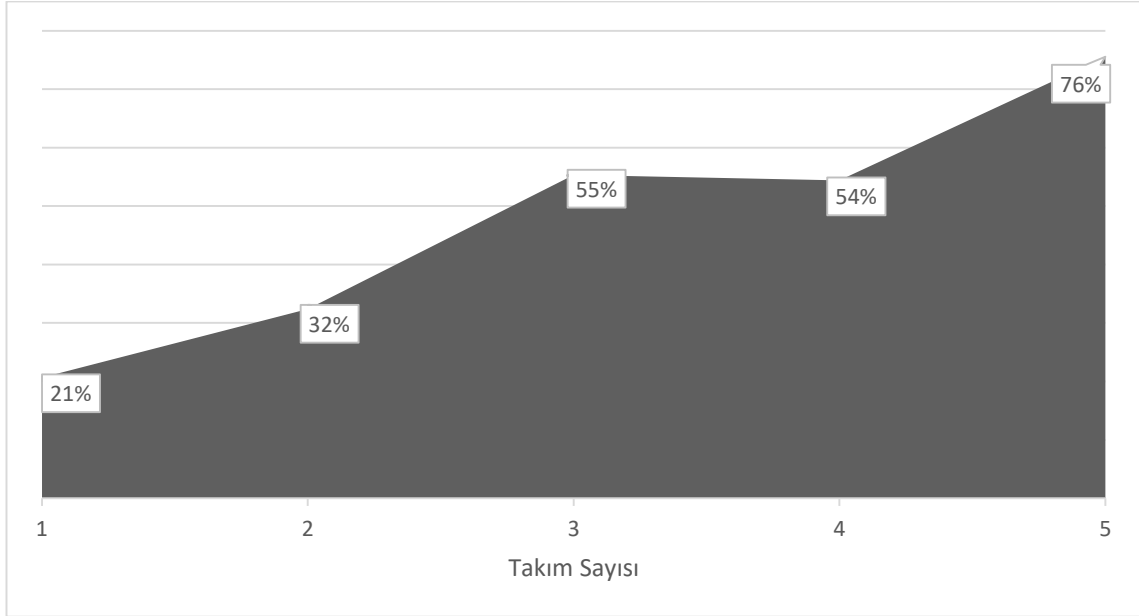
3.6.2 Otoklav Yükleme ve Çizelgeleme Tek Model

Otoklav yükleme ve çizelgelemenin tek matematiksel model olarak çalıştırıldığı durumda alan kullanımı ve karşılanan talep Tablo-15'de görülmektedir.

Takım Sayısı	Alan Kullanımı	Karşılanan Talep
1	21%	16
2	32%	32
3	55%	42
4	54%	48
5	76%	52

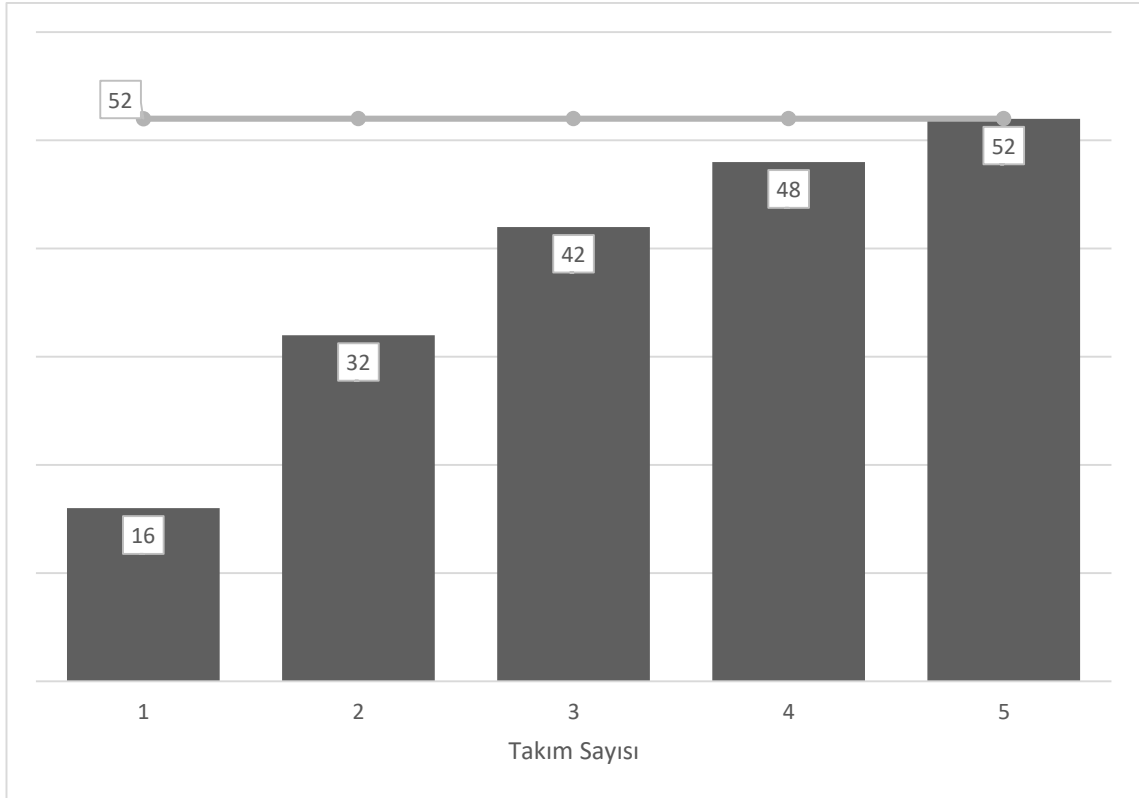
Tablo – 15 Tek Model Alan Kullanımı ve Talep Karşılama

Şekil-28'de görüldüğü üzere alan kullanımı parçalar için 1 takım olduğu durumda %21 iken, 2 takım mevcut olduğunda %32'ye çıkmıştır. Her parça için mevcut takım sayısını 3'e çıkarmak alan kullanımını %23 daha arttırarak %55'e çıkarmış, takım sayısını 5'e çıkarmak ise alan kullanımını %76'ya çıkarmıştır.



Şekil – 28 Tek Model - Alan Kullanımı

Şekil-29 takım sayısının arttırdığımız zaman karşılanan talep miktarının nasıl değiştiğini göstermektedir. Takım sayısını arttırmak gittikçe azalan bir oranda karşılanan talep miktarını arttırmaktadır.

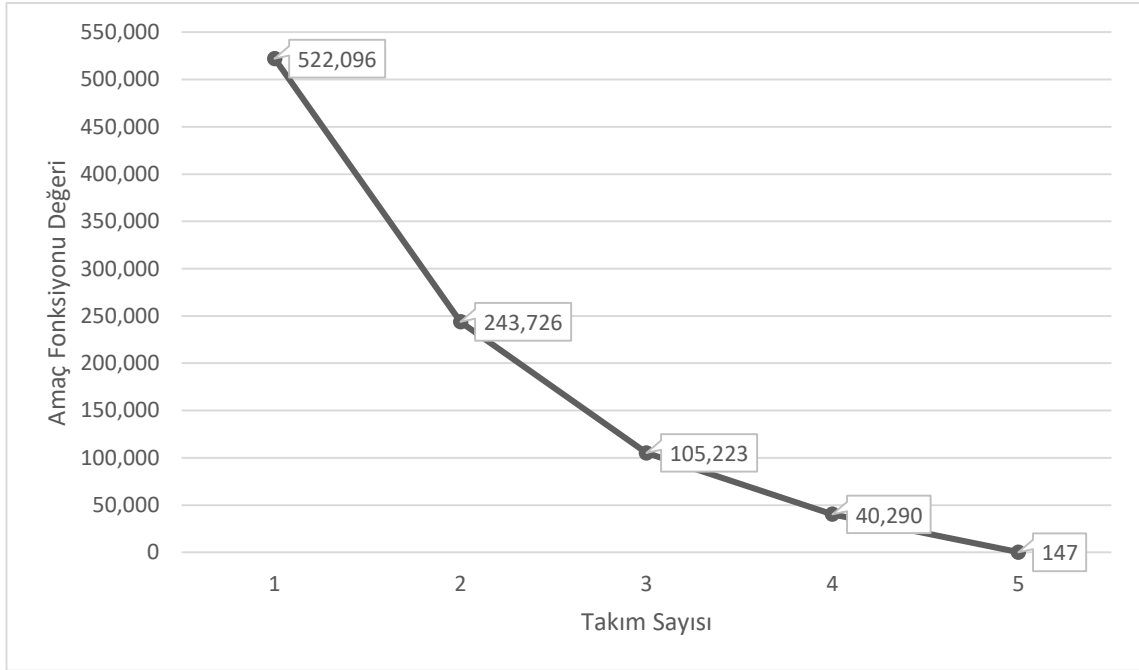


Şekil – 29 Tek Model - Talep Karşılama

Otoklav yükleme çizelgeleme matematiksel modeli amaç fonksiyonu aşağıdaki şekilde belirlenmiştir.

$$\text{Minimize } \sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + 1000 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right) + 1000 * \sum_{p \in P} \sum_{z \in Z} \left(\frac{UD_{p,z}}{D_{p,z}} \right)$$

Şekil-30'da takım sayısı ekseninde amaç fonksiyonu değerleri görülmektedir, takım sayısı arttıkça amaç fonksiyonu değeri de azalan bir oranda azalmaktadır.



Şekil – 30 Tek Model - Amaç Fonksiyonu

Otoklav yükleme ve çizelgelemenin tek matematiksel model ile çözüldüğü durumda takım sayısını arttırmanın hem alan kullanımı hem de karşılanan talepte azalan bir şekilde artış sağladığı gözlenmiştir.

Tablo-16 matematiksel model ile ilgili bilgileri içermektedir. Model 2688 değişken ve 4536 kısıttan oluşmaktadır. Çözüm süresi modelde kullanılabilir olarak tanımlanan takım sayısına göre 4 saniye ve 2 saat arasında değişiklik göstermektedir.

Takım Sayısı	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
1	522,096.37	4.00	4536	2668
2	243,726.43	16.67	4536	2668
3	105,223.21	11.19	4536	2668
4	40,290.45	43.36	4536	2668
5	147.23	7,220.19	4536	2668

Tablo – 16 Tek Model - Model ve Çözüm Sayısal Değerleri

Otoklav yükleme ve çizelgelemenin tek model olarak çalıştırıldığı durumda oluşan çizelgelerin gantt şemaları Ek.7'de verilmiştir.

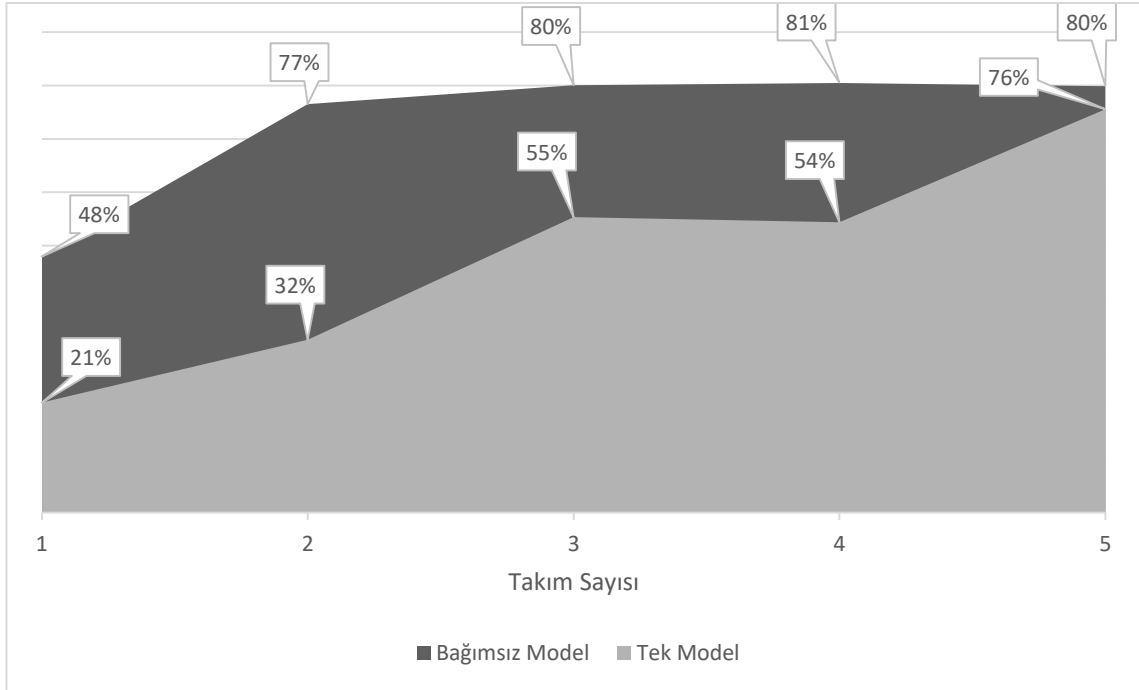
3.7. OTOKLAV YÜKLEME VE ÇİZELGELEMENİN BAĞIMSIZ VE TEK MODEL İLE ÇÖZÜMÜ

Yapılan literatür taramasında bugüne kadar otoklav yükleme ve çizelgelemenin tek bir matematiksel model ile çözüldüğü bir çalışmaya rastlanamamıştır. Problemin yükleme ve çizelgeleme olarak ikiye bölünerek çözülmesi ile tek bir model olarak çözülmesi arasında bir fark olup olmadığı bu bölümde örnek veri seti üzerinden incelenecektir.

Gerçek veriler arasından seçilen örnek veri seti ile otoklav yükleme ve çizelgelemenin bağımsız ve tek olarak ele alındığı matematiksel modeller bu örnek veri seti ile takım sayısı 1'den 5'e kadar artacak şekilde 5'er kez çalıştırılmıştır. Örnek veri seti içerisinde 2 ayrı kür grubuna ait 8 parça için toplam 1 günlük 52 adet ihtiyaç bulunmaktadır. İhtiyaç miktarları takım sayısının etkisini gösterebilmek için gerçek değerlerine göre artırılmıştır.

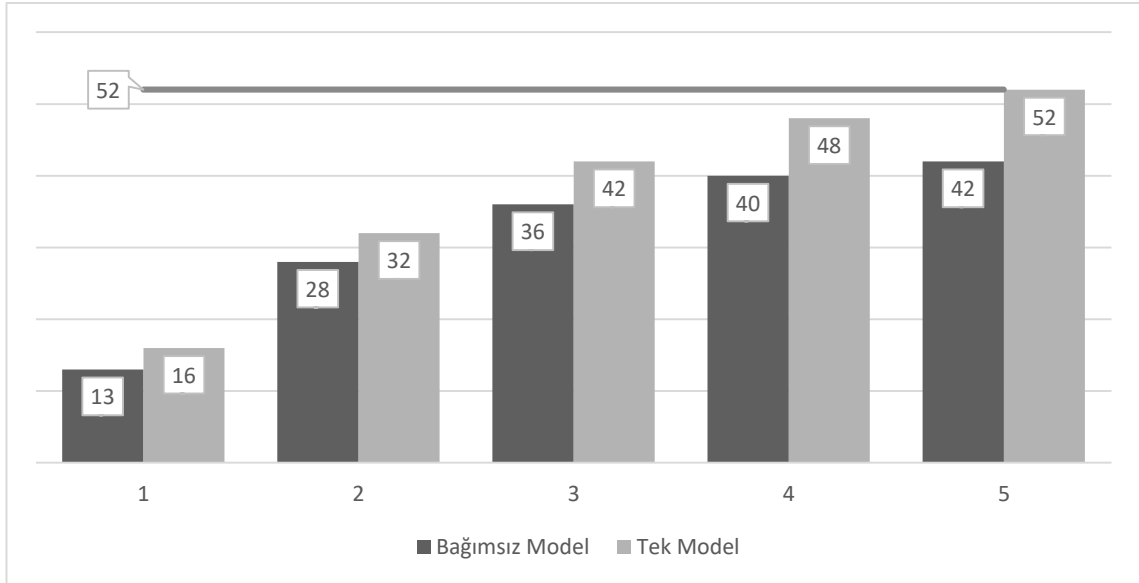
Şekil-31'de otoklav yükleme ve çizelgelemenin bağımsız olarak ele alındığı modellerin çözümü ile tek model olarak çözümün otoklav alan kullanımı yönünden karşılaştırması görülebilir.

Bağımsız modelde yükler oluşturulurken çizelgeleme kısıtları olmadığı için alan kullanımının yüksek olması beklenen bir sonuçtur. Takım sayısının artması ile orantılı olarak otoklav yükleme ve çizelgelemenin aynı anda ele alındığı matematiksel modelin çözümündeki otoklav alan kullanımı bağımsız olarak ele alındığı modellerin çözümündeki alan kullanımına yaklaşmaktadır. 5 takım mevcut olması durumunda aradaki fark %4'e kadar inmektedir.



Şekil – 31 Bağımsız ve Tek Model - Kullanılan Alan Karşılaştırma

Bağımsız modelde otoklav alan kullanımının yüksek olması beklenirken yükler oluşturulurken çizelge kısıtları olmadığı için talep karşılama miktarlarının da tek modele göre daha az olması beklenmektedir. Şekil – 32’de görüldüğü üzere bağımsız modeldeki talep karşılama miktarı 5 takım kullanılması durumunda bile tek modeldeki 3 takım kullanıldığı duruma gelebilmektedir. Tek modelin çözümündeki talep karşılama miktarları bağımsız modelin çözümüne göre yüksektir ve 5 takım mevcut olması durumunda tek modelin çözümünde tüm ihtiyaç karşılanabilmektedir.



Şekil – 32 Bağımsız ve Tek Model - Talep Karşılama Karşılaştırması

Tablo-17, 1 – 5 arası takım kullanımında bağımsız ve tek modelin talep karşılama ve alan kullanım performansını karşılaştırmaktadır.

Takım Sayısı	Talep Karşılama		Alan Kullanım		Toplam Alan Kullanım		Parça Başı Alan Kullanım	
	Bağımsız	Tek	Bağımsız	Tek	Bağımsız	Tek	Bağımsız	Tek
1	13	16	48.0%	20.6%	24.82	32.52	1.91	2.03
2	28	32	76.6%	32.4%	51.84	65.04	1.85	2.03
3	36	42	80.1%	55.4%	74.4	82.16	2.07	1.96
4	40	48	80.5%	54.4%	75.78	88.22	1.89	1.84
5	42	52	79.9%	75.6%	64.43	88.78	1.53	1.71

Tablo – 17 Bağımsız ve Tek Model - Talep ve Alan Kullanım Karşılaştırması

Firma için her yüklemdeki otoklav alan kullanımının yüksek olması maliyetleri azaltma açısından önemli olsa da talebin karşılanması daha büyük bir öneme sahiptir. Bu nedenle tek modelde amaç fonksiyonunda talep karşılama oranının ağırlığı alan kullanımına göre çok yüksek bir değer olarak verilmiştir (1/1.000).

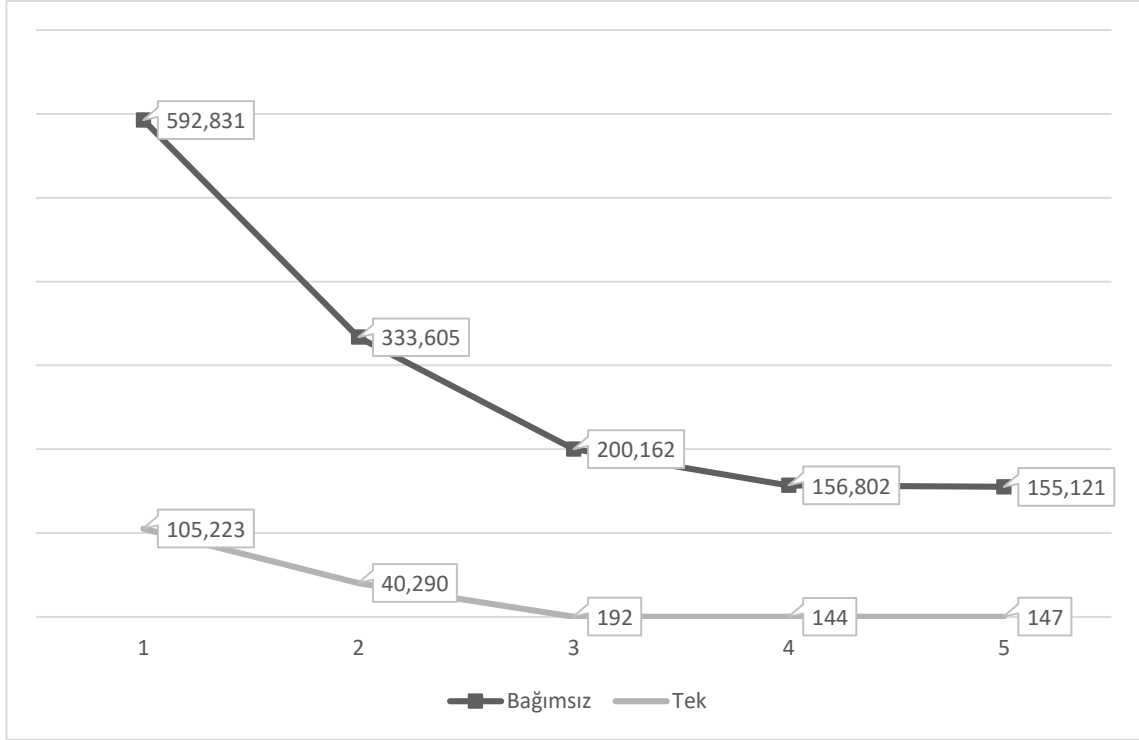
Bağımsız model sonuçları ile tek modelin sonuçlarının karşılaştırılabilmesi için her iki modelin çözümünden elde edilen sonuçlar aşağıdaki amaç fonksiyonuna uygulanarak karşılaştırılmıştır.

$$\sum_{o \in O} \sum_{i \in I} \frac{UA_{o,i}}{AO_o} + 1000 * \sum_{p \in P} \left(\frac{TUD_p}{\sum_{z \in Z} D_{p,z}} \right)$$

Şekil-33 bağımsız ve tek modelin çözümünün ortaklaştırılmış amaç fonksiyonu ile karşılaştırmasını göstermektedir. Tek model tüm takım sayılarında daha iyi sonuçlar elde edilmesini sağlamıştır.

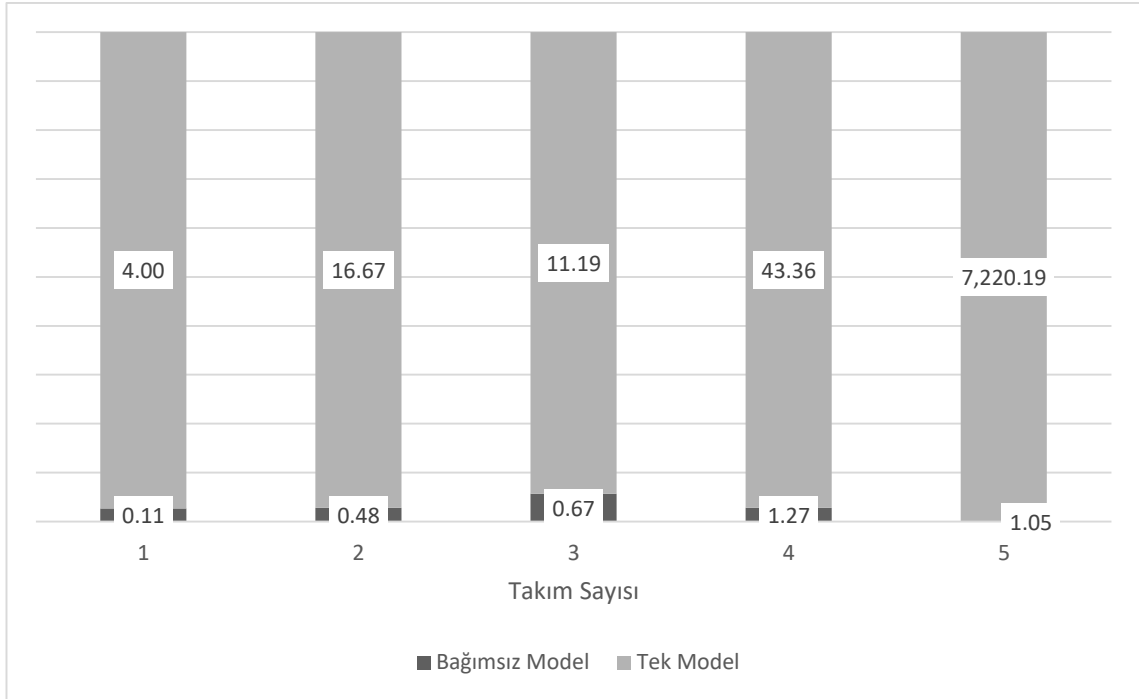
Takım Sayısı	Amaç Fonksiyonu Değeri		Çözüm Süresi (sn.)	
	Bağımsız	Tek	Bağımsız	Tek
1	592,831	105,223	0.11	4
2	333,605	40,290	0.485	16.672
3	200,162	192	0.672	11.188
4	156,802	144	1.265	43.36
5	155,121	147	1.047	7220.187

Tablo – 18 Bağımsız ve Tek Model Amaç Fonksiyonu Değeri ve Çözüm Süresi



Şekil – 33 Bağımsız ve Tek Model - Amaç Fonksiyonu Karşılaştırması

Sıklıkla değişen öncelikler, üretim sürecindeki belirsizlikler ve yönetim kararları oluşturulan matematiksel modelin gün içerisinde tekrar çözülmesini gerektirebilmektedir. Bu nedenle modelin çözüm süresi firma için önemli bir parametre olarak görülmektedir. Şekil-34 bağımsız ve tek modellerin farklı takım sayıları ile çözüldüğündeki süreleri karşılaştırmaktadır. Tek modelin çözüm süresi bağımsız modele göre oldukça yüksek olmasına rağmen en yüksek çözüm süresine sahip olan 5 takım ile oluşturulan modelde bile 20 dk. gibi kabul edilebilir bir çözüm süresi elde edilmiştir.



Şekil – 34 Bağımsız ve Tek Model - Süre Karşılaştırması

3.8. MODELİN FARKLI GÜNLER İÇİN ÇALIŞTIRILMASI

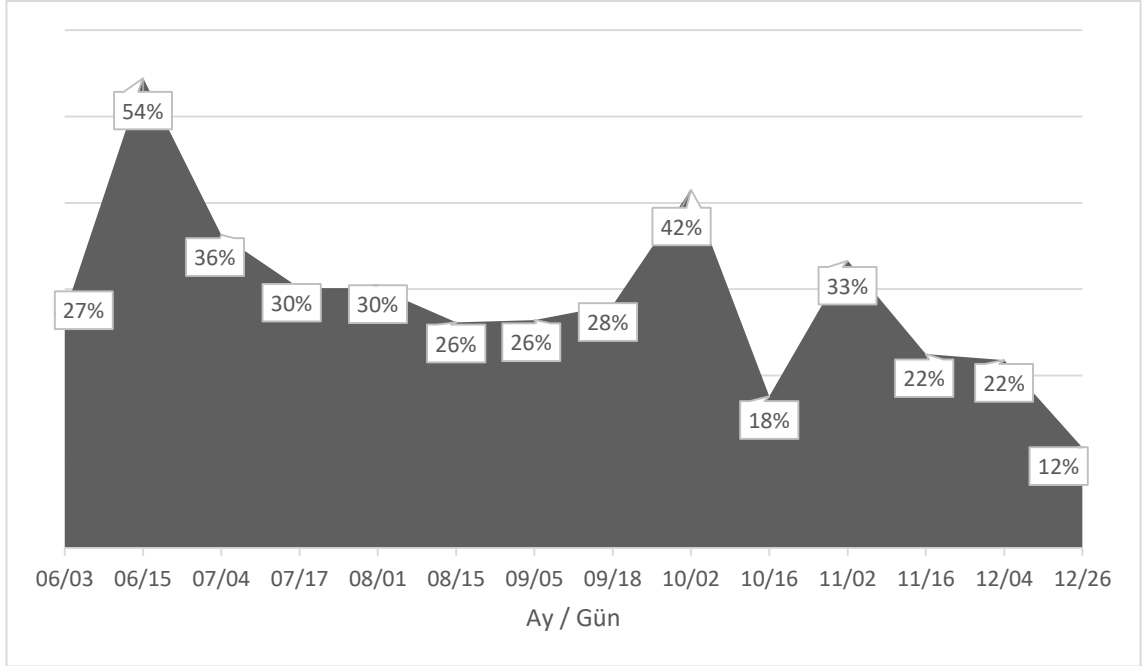
Sıklıkla değişen öncelikler, üretim sürecindeki belirsizlikler ve yönetim kararları nedeni ile günlük çizelgelemenin ötesinde bir ufukta yapılan çizelgelemenin değiştirilmeden uygulanması mümkün olmamasına rağmen firma en az 2 haftalık planlama ufkunda bir çizelgelemeye sahip olmak istemektedir. 2 haftalık bir planlama ufkuna sahip olabilmek için her günün ayrı optimize edilmesi bir alternatif olarak değerlendirilebilir.

Modelin birden fazla gün için çalıştığını görebilmek amacıyla firmanın 2017 yılı verileri arasından Haziran – Aralık dönemi için her ay başında ve ortasında olmak üzere 14 ayrı örneklem seçilmiştir. Otoklav yükleme ve çizelgelemenin beraber ele alındığı matematiksel modelin açık bir şekilde daha iyi sonuçlar verdiği görüldüğü için seçilen veri setleri tek model ile çözülmüştür. Firmada parçalar için genellikle 1 takım bulundurulduğu için model 1 takım kullanacak şekilde oluşturulmuştur.

Şekil-35 modelin çözüldüğü her gün için alan kullanımını göstermektedir. İlgili günde talep edilen parçaların özelliklerine ve talep miktarlarına göre otoklav kullanım oranları %12 ile %54 arasında değişmektedir.

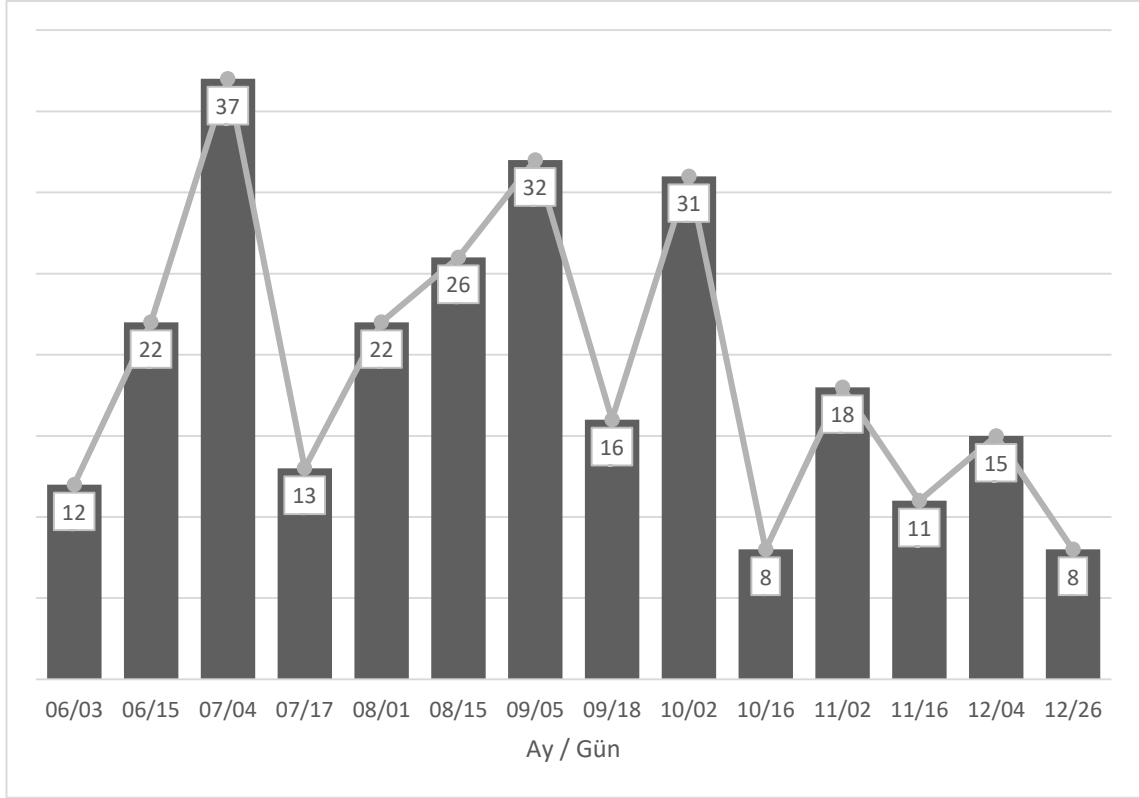
Ay / Gün	Alan Kullanım	Karşılanan Talep	Toplam Talep
06/03	27%	12	12
06/15	54%	22	22
07/04	36%	37	37
07/17	30%	13	13
08/01	30%	22	22
08/15	26%	26	26
09/05	26%	32	32
09/18	28%	16	16
10/02	42%	31	31
10/16	18%	8	8
11/02	33%	18	18
11/16	22%	11	11
12/04	22%	15	15
12/26	12%	8	8

Tablo – 19 Farklı Günler Çalışma Alan Kullanımı ve Talep Karşılama



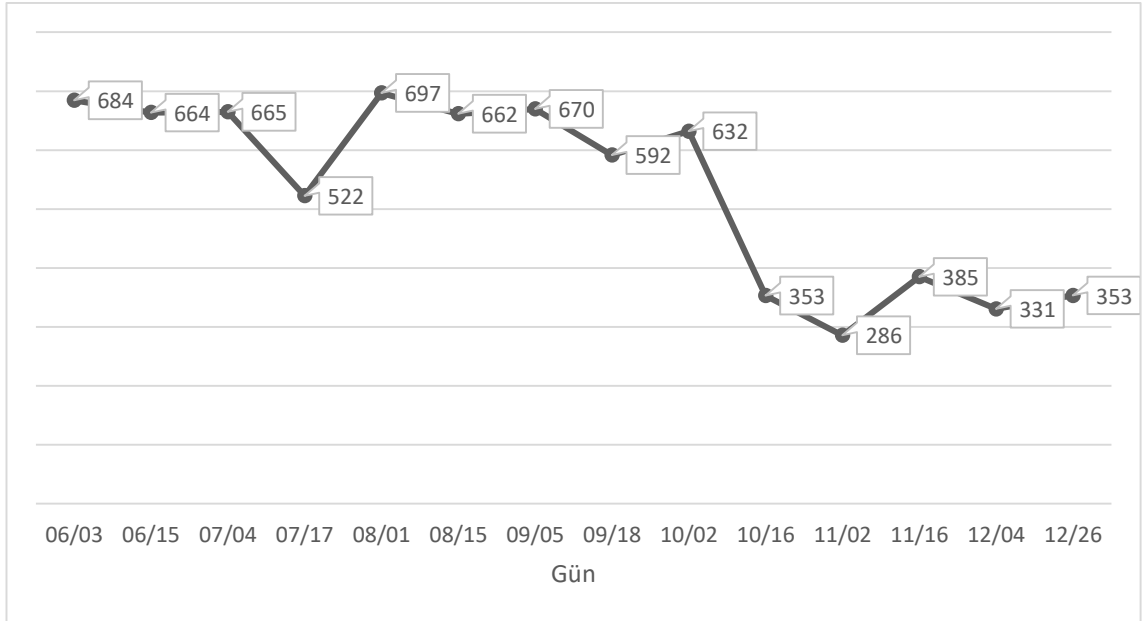
Şekil – 35 Farklı Günler Çalışma - Alan Kullanımı

Şekil-36 modelin çözüldüğü her gün için talep karşılama miktarlarını göstermektedir. Şekilde görülebileceği üzere talebin karşılanamadığı bir gün olmamıştır.



Şekil – 36 Farklı Günler Çalışma - Talep Karşılama

Şekil-37 amaç fonksiyonunun modelin çözüldüğü günler için aldığı değerleri göstermektedir. İlgili günde talep edilen parçaların özelliklerine ve talep miktarlarına göre amaç fonksiyonu değeri değişim göstermektedir.



Şekil – 37 Farklı Günler Çalışma - Amaç Fonksiyonu Değeri

Otoklav yükleme ve çizelgeleme modelinin farklı günler için çalıştırıldığı durumda oluşan çizelgelerin gantt şemaları Ek.8'de verilmiştir.

Ek.8'de görülebileceği üzere tüm otoklavlar 14 günde 612 otoklav/saat çalışarak talebi karşılayabilmiştir. Firma aynı talebi 7 otoklavın 3 vardiya (24 saat) 14 gün çalışması ile 2352 otoklav/saat çalışarak karşılayabilmiştir. Optimizasyon modelinin kullanılması ile otoklavlar %74 daha az çalıştırılabilmiştir. Yıllık 1 otoklav çalışma maliyetinin 500.000 ABD doları olduğu göz önüne alındığında 7 otoklav için yıllık 2.590.000 ABD doları kazanç sağlanmaktadır.

Tablo-20 modelin oluşturulduğu ve çalıştırıldığı her gün için model ve çözüm ile ilgili bilgileri göstermektedir.

Tarih	Amaç Fonksiyonu Değeri	Çözüm Süresi	Kısıt Sayısı	Değişken Sayısı
06/03	684.46	77.64	8,064	3,932
06/15	663.63	54.44	7,308	4,354
07/04	664.89	252.59	19,229	10,852
07/17	522.45	15.69	7,224	3,862
08/01	696.90	295.19	12,747	6,426
08/15	661.55	160.30	14,672	7,796
09/05	669.72	172.59	16,744	9,447
09/18	591.60	122.33	9,352	4,986
10/02	632.06	262.08	14,700	8,850
10/16	353.12	11.73	4,256	2,633
11/02	285.75	6.61	8,526	5,443
11/16	385.04	4.14	3,829	2,352
12/04	330.67	36.27	5,537	3,476
12/26	353.46	3.16	4,256	2,633

Tablo – 20 Farklı Günler Çalışma - Model ve Çözüm Sayısal Değerleri

3.9. MODELİN PLANLAMA UFKUNUN GENİŞLETİLMESİ

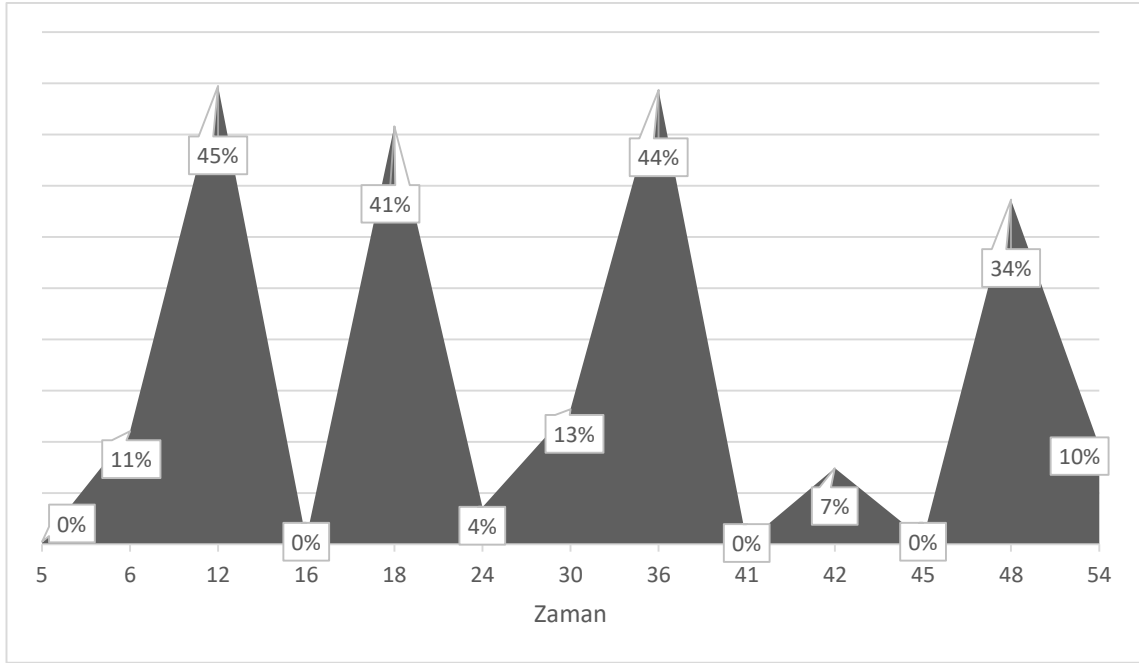
Her gün için modelin çözülmesi bir alternatif olarak görülebilir fakat gün içerisinde karşılanamayan talep olması durumunda planlama ufkunun genişletilerek modelin bir seferde çalışmasının daha doğru bir sonuç vereceği açıktır. Modelin planlama ufkunu genişletildiğinde çalıştığını görebilmek amacıyla firmanın 2017 yılı verileri arasından Aralık ayındaki ilk 15 gün içerisindeki 9 sıralı gün alınmıştır.

Otoklav yükleme ve çizelgelemenin beraber ele alındığı tek model kullanılmıştır. Firmada parçalar için genellikle 1 takım bulundurulduğu için model 1 takım kullanacak şekilde oluşturulmuştur.

Şekil-38 alan kullanımını göstermektedir, toplam talebin karşılanabilmesi için talep olmayan zamanlarda da yük çizelgelenmiştir, bu zamanlarda alan kullanımı çok düşük değerlerde olmaktadır.

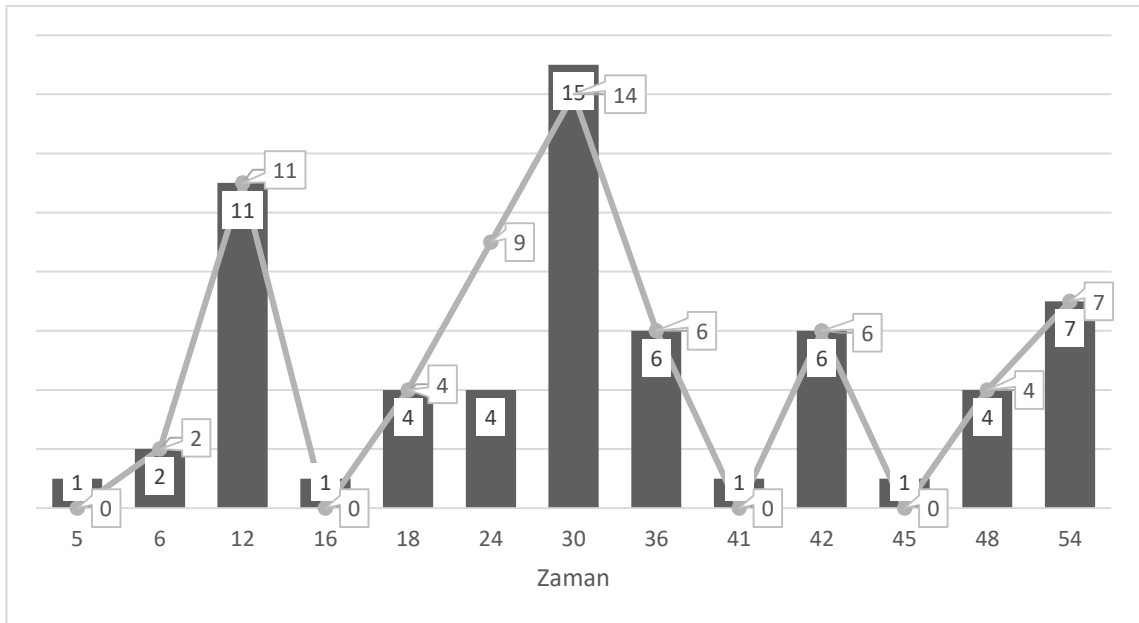
Zaman	Alan Kullanım	Karşılanan Talep	Toplam Talep
5	0%	1	0
6	11%	2	2
12	45%	11	11
16	0%	1	0
18	41%	4	4
24	4%	4	9
30	13%	15	14
36	44%	6	6
41	0%	1	0
42	7%	6	6
45	0%	1	0
48	34%	4	4
54	10%	7	7

Tablo – 21 Geniş Planlama Ufku Alan Kullanımı ve Talep Karşılama



Şekil – 38 Geniş Planlama Ufku - Alan Kullanımı

Şekil-39 Talep karşılama durumunu göstermektedir. Yirmi dördüncü zamandaki altı adet talebin planlama ufku sonunda karşılanabilmesi için hem talep zamanından önceye hem de talep zamanından sonra çizelgeleme yapılmıştır.



Şekil – 39 Geniş Planlama Ufku - Talep Karşılama

Modelin çözümlü yaklaşık 10 saat sürmüştür. Tablo-22 model ve çözümlü ile ilgili sayısal değerleri içermektedir.

Amaç Değeri	85.825,5
Çözüm Durumu	Tamsayı optimal
Çözüm Süresi	36.740 sn.
Kısıt Sayısı	121.554
Değişken Sayısı	99.553

Tablo – 22 Geniş Planlama Ufku - Model ve Çözüm Sayısal Değerleri

Otoklav yükleme ve çizelgeleme modelinin genişletilmiş planlama ufku çalıştırıldığı durumda oluşan çizelgenin gantt şeması Ek.9'da verilmiştir.

SONUÇ

Kompozit üretiminde otoklavı verimli kullanacak ve bu sayede maliyet avantajı sağlayacak yükleme ve çizelgeleme bu sektörde üretim yapan firmalarda çözümü zor bir problem olarak karşımıza çıkmaktadır. Yükleme ve çizelgeleme işini çalışan personelin tecrübesine bırakmak en iyi çözümlerden uzaklaştırmakta, iş gücü ve zaman kaybına yol açmaktadır. Bu tez çalışmasında oluşturulan karma tam sayı kuadratik kısıtlı programlama modelinin firmanın otoklav yükleme ve çizelgeleme problemine çözüm olabileceği, model gerçek verilerle çalıştırılarak gösterilmiştir. Önerilen model otoklav yükleme ve çizelgelemeyi beraber ele aldığı ve sezgisel yöntemler yerine doğrudan matematiksel model olarak oluşturulduğu için bugüne kadar yapılan çalışmalardan farklılık göstermektedir.

Otoklav yükleme ve çizelgeleme problemlerini bağımsız olarak ele alınıp çözülmesi çizelgeleme ve yükleme arasında bir geri bildirim olmadığı için en iyi çözümden uzaklaşabilmektedir. Bu tez çalışmasında otoklav yükleme ve çizelgeleme problemleri bağımsız ve tek model ile çözümlenerek yükleme ve çizelgeleme arasında bir geri bildirim olmasının elde edilen çözümün daha iyi olmasını sağladığı gösterilmiştir. Literatürde otoklav yükleme ve çizelgelemenin geri bildirimli tek bir modelde çözüldüğü bir çalışmaya rastlanamamıştır.

Sezgisel yöntemler hızlı olmasına rağmen en iyi sonuca her durumda ulaşmanın mümkün olmadığı aşikardır, bu tez çalışmasında sezgisel yöntemler yerine matematiksel model geliştirildiği için literatürde benzer problemlerin çözümü için sezgisel yöntemler kullanan çalışmalara göre daha iyi bir çözüme ulaşabilmeyi sağlamıştır.

Bölüm 3.5'de otoklavlar firmanın kullanıma aldığı sıra ile modele dahil edilmiş, her durum için takım sayısı artırılarak model çalıştırılmıştır. Toplam 35 model sonucu incelenmiş ve takım sayısını arttırmanın çözüme etkisinin otoklav sayısını arttırmaktan daha fazla olduğu görülmüştür. Otoklav sayısının çözümüne etkisi ile ilgili bir çalışmaya literatür taramasında rastlanamamıştır.

Bölüm 3.6'da kompozit parçaların üzerine serildiği ve otoklava girdiği takım sayılarının sonuca etkisi analiz edilmiştir. Takım sayısını arttırmanın azalan oranda bir fayda sağladığı görülmüştür. Takım sayısının otoklav yükleme ve çizelgeleme problemi çözümüne etkisi ile ilgili bir çalışmaya literatür taramasında rastlanamamıştır.

Bölüm 3.7'de otoklav yükleme ve çizelgeleme problemlerinin bağımsız çalıştırılması ile birbirine geri bildirim sağlayacak şekilde birlikte modellenmesi karşılaştırılmıştır. Örnek veri seti ile farklı takım sayılarında elde edilen sonuçlar yükleme ve çizelgelemenin tek modelde çalıştırılmasının daha iyi sonuç verdiğini göstermiştir.

Bölüm 3.8. ve 3.9'da oluşturulan modelin firmanın gerçek verileri ile çalıştırılması sağlanmış ve bu tez çalışması sonucu çıkan modelin firmanın problemini çözdüğü gösterilmiştir.

Bu tez çalışmasında oluşturulan modeller ve çözüm yaklaşımları otoklav ile kompozit üretimi gerçekleştiren tüm firmalarda kullanılabilir. Türkiye'de bu çalışmanın uygulanabileceği firmalar Bölüm 1.2'de listelenmiştir. Türkiye'de toplam 10 firmada 35 adet otoklav olduğu hesaplanmaktadır. Tez çalışmasında 7 otoklav için işletme giderlerinde yıllık 2.590.000 ABD doları kazanç sağlandığı görülmüştür. Bu çalışmada geliştirilen modellerin kullanımı ile Türkiye çapında kompozit üretimi yapan firmaların işletim maliyetlerinde toplam 13 milyon ABD doları kazanç sağlanması öngörülmektedir. Türk Havacılık ve Uzay Sanayii'nin yeni kompozit tesisinin devreye alınması ile bu kazanç katlanarak artacaktır.

Bu tez çalışmasında oluşturulan modeller ve çözüm yaklaşımı bazı değişiklikler ile farklı sektörlere de uygulanabilir. Yarı iletken üretim, ayakkabı üretimi, fırın kullanımı gerektiren metal işleri, seramik üretimi bu sektörlere örnek verilebilir. Literatür taramasında bu sektörlerde, özellikle yarı iletken üretimi sektöründe yapılan bir çok çalışma bulunduğu görülmüştür, bu çalışmalarda otoklav yükleme ve çizelgeleme problemine göre daha az sayıda kısıt bulunduğu gözlemlenmiştir. Bazı kısıtlar çıkarılarak bu tez çalışmasındaki modeller diğer sektörlere de uygulanabilir.

Otoklavlar oldukça fazla enerji tüketimine sahip olan ekipmanlardır, bu tez çalışmasında otoklavın kullanılmayan alanı minimize edilerek daha az çalışması ve enerji tüketiminin minimize edilmesi amaçlanmıştır. Yalnız firmaların enerji tüketim profillerine göre farklı fiyat tarifeleri olabilmektedir, ilerideki çalışmalarda enerji tüketim tarifeleri modele eklenerek daha maliyet etkin çözümlerin elde edilmesi sağlanabilir.

Bazı kompozit parçaların birden fazla kez kütleme işleminden geçmesi gerekebilmektedir. Aynı parçanın tekrar otoklavda kütleme işlemine girebilmesi için ilk otoklavdan çıktıktan sonra belirli bekleme sürelerinin sağlanması ve serim işleminin tekrarlanması için süre ayrılması gerekmektedir. Firmanın ürettiği parçalarda bu tip parçalar az olduğu için oluşturulan modellerde bu kısıt ele alınmamıştır. Tekrar otoklava girmesi gereken parçaların çok olduğu durumlar için modele yeni kısıtlar eklenerek geliştirilebilir.

Kompozit parçalar otoklava yüklenmeden önce temiz odada serim işleminden geçmektedir. Bu tez çalışmasında serim işlemi tamamlanmış parçaların otoklava yüklenmesi ve çizelgelenmesi ele alınmıştır. Serim işleminin bağımsız olması genel üretim sürecinde verimli bir sonuca ulaşmayı engelleyebilecektir. Otoklav yükleme ve çizelgeleme ile temiz oda serim işlemi çizelgelemesi arasında bir geri bildirim sağlanması genel üretim süreci açısından daha iyi sonuç alınmasını sağlayabilecektir. Temiz oda serim işlemlerinin de çizelgelenmesi de modele dahil edilerek model geliştirilebilir.

KAYNAKÇA

- Dünyada ve Türkiye’de Kompozit Sektörünün Görünümü. (2015, Mart). *Composites Turkey*, 30–33. <http://www.kompozit.org.tr/wp-content/uploads/2017/03/Composites-Turkey-09.pdf> adresinden erişildi.
- Alharkan, I. M. (2005). *Algorithms for Sequencing and Scheduling*. King Saud University.
- Anthony, R. N. (1965). *Planning and Control Systems: A Framework for Analysis. Division of Research*. Cambridge MA: Harvard University Press.
- Arıcasoy, O. (2006). *Kompozit Sektör Raporu*. İstanbul Ticaret Odası.
- Azizoglu, M. ve Webster, S. (2001). Scheduling a batch processing machine with incompatible job families. *Computers & Industrial Engineering*, 39(3–4), 325–335.
- Brucker, P. (2007). *Scheduling algorithms*. Springer.
- Bulut, M. (2014). *Türkiyede kompozit malzeme üretimi ve kompozit malzeme sektörünün genel değerlendirilmesi*. Gazi Üniversitesi.
- Castro, P. M. ve Grossmann, I. E. (2006). An efficient MILP model for the short-term scheduling of single stage batch plants. *Computers & Chemical Engineering*, 30(6–7), 1003–1018.
- Castro, P. M. ve Grossmann, I. E. (2005). New Continuous-Time MILP Model for the Short-Term Scheduling of Multistage Batch Plants. *Industrial & Engineering Chemistry Research*, 44(24), 9175–9190.
- Coffman Jr., E. G., Csirik, J., Galambos, G., Martello, S. ve Vigo, D. (2013). Bin Packing Approximation Algorithms: Survey and Classification. P. M. Pardalos, D.-Z. Du ve R. L. Graham (Ed.), *Handbook of Combinatorial Optimization* içinde (ss. 455–531). New York, NY: Springer New York.

- Collart, A. (2015). *An Application of Mathematical Optimization to Autoclave Packing and Scheduling in a Composites Manufacturing Facility*. Dalhousie University.
- Conway, R. W., Maxwell, W. L. ve Miller, L. W. (1967). *Theory of scheduling*. Reading, Mass.; Mineola, NY: Addison-Wesley ; Dover Pub.
- Cook, L. K., Hinkle, D. A. ve Bickmore, T. W. (1990). Planning for the manufacturing domain: long-term and reactive scheduling. *First International Conference on Expert Planning Systems* içinde (ss. 6–10). IET.
- de Carvalho, J. M. (1999). Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, 86(0), 629–659.
- Dileep, R. S. (2008). *Production planning and industrial scheduling : examples, case studies, and applications*. CRC Press.
- Dios, M., Gonzalez-R, P. L., Dios, D. ve Maffezzoli, A. (2016). A mathematical modeling approach to optimize composite parts placement in autoclave. *International Transactions in Operational Research*, 24, 115–141.
- Epstein, L., Imreh, C. ve Levin, A. (2010). Class constrained bin packing revisited. *Theoretical Computer Science*, 411(34–36), 3073–3089.
- Eren, T. ve Güner, E. (2002). Tek ve Paralel Makinalı Problemlerde Çok Ölçütlü Çizelgeleme Problemleri için Bir Literatür Taraması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 17(4).
- Fanti, M. P., MAIONE, B., PISCITELLI, G. ve Turchiano, B. (1996). Heuristic scheduling of jobs on a multi-product batch processing machine. *International Journal of Production Research - INT J PROD RES*, 34, 2163–2186.

- French, S. (1982). *Sequencing and scheduling: an introduction to the mathematics of the job-shop*. E. Horwood.
- Garey, M. R. ve Johnson, D. S. (1978). `` Strong ’ ’ NP-Completeness Results: Motivation, Examples, and Implications. *Journal of the ACM*, 25(3), 499–508.
- Garey, M. R. ve Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman.
- Gokhale, R. ve Mathirajan, M. (2014). Minimizing total weighted tardiness on heterogeneous batch processors with incompatible job families. *The International Journal of Advanced Manufacturing Technology*, 70(9–12), 1563–1578.
- Graham, R. L., Lawler, E. L., Lenstra, J. K. ve Kan, A. H. G. R. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. *Annals of Discrete Mathematics*, 5, 287–326.
- Gupta, S. ve Karimi, I. A. (2003). An Improved MILP Formulation for Scheduling Multiproduct, Multistage Batch Plants. *Industrial & Engineering Chemistry Research*, 42(11), 2365–2380.
- Hashim, N., Zulkipli, F., Januri, S. ve Shariff, S. (2014). An Alternative Heuristics for Bin Packing Problem. *iieom.org* içinde (ss. 1560–1568).
- Hennessy, D. ve Hinkle, D. (1992). Applying case-based reasoning to autoclave loading. *IEEE Expert*, 7(5), 21–26.
- Hillier, F. S. ve Lieberman, G. J. (1986). Introduction to operations research, 888.
- Hindle, K. ve Duffin, M. (2006). SIMUL8-planner for Composites Manufacturing. *Proceedings of the 38th Conference on Winter Simulation* içinde , WSC '06 (ss. 1779–1784). Winter Simulation Conference.
- Hinrichsen, J. (2018, Ocak). The Aerospace Market. *Composites Manufacturing*, 21–22.

<http://compositesmanufacturingmagazine.com/digital/2018/Composites-Manufacturing-January-February-2018-Issue.pdf> adresinden erişildi.

K. Friesen, D. ve Langston, M. (1986). Variable Sized Bin Packing. *SIAM Journal on Computing*, 15, 222–230.

Karthikeyan, D. (2017, Ocak). The Aerospace Market. *Composites Manufacturing*, 24–25.

<http://compositesmanufacturingmagazine.com/digital/2017/Composites-Manufacturing-January-February-2017-Issue.pdf> adresinden erişildi.

Koblasa, F., Vavroušek, M. ve Manlig, F. (2015). Two-dimensional Bin Packing Problem in batch scheduling. *Proceedings of 33rd International Conference Mathematical Methods in Economics* içinde (C. 1, ss. 354–359).

Koh *, S.-G., Koo, P.-H., Ha, J.-W. ve Lee, W.-S. (2004). Scheduling parallel batch processing machines with arbitrary job sizes and incompatible job families. *International Journal of Production Research*, 42(19), 4091–4107.

Koh, S.-G., Koo, P.-H., Kim, D.-C. ve Hur, W.-S. (2005). Scheduling a single batch processing machine with arbitrary job sizes and incompatible job families. *International Journal of Production Economics*, 98(1), 81–96.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. ve Shmoys, D. B. (1993). Chapter 9 Sequencing and scheduling: Algorithms and complexity. *Handbooks in Operations Research and Management Science*, 4, 445–522.

Luh, P. B., Wang, J. H., Wang, J. L., Tomastik, R. N. ve Howes, T. D. (1997). Near-Optimal Scheduling of Manufacturing Systems with Presence of Batch Machines and Setup Requirements. *CIRP Annals*, 46(1), 397–402.

Martello, S., Pisinger, D. ve Vigo, D. (2000). The Three-Dimensional Bin Packing Problem. *Operations Research*, 48(2), 256–267.

- Martello, S. ve Toth, P. (1990). Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28(1), 59–70.
- Mathirajan, M., Chandru, V. ve Sivakumar, A. I. (2007). Heuristic algorithms for scheduling heat-treatment furnaces of steel casting industries. *Sadhana - Academy Proceedings in Engineering Sciences*, 32(5), 479–500.
- Mathirajan, M. ve Sivakumar, A. I. (2006). Minimizing total weighted tardiness on heterogeneous batch processing machines with incompatible job families. *The International Journal of Advanced Manufacturing Technology*, 28(9–10), 1038–1047.
- Mathirajan, M., Sivakumar, A. I. ve Chandru, V. (2004). Scheduling algorithms for heterogeneous batch processors with incompatible job-families. *Journal of Intelligent Manufacturing*, 15(6), 787–803.
- Mathirajan, M. ve Sivakumar, A. I. I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *The International Journal of Advanced Manufacturing Technology*, 29(9), 990–1001.
- Mathirajan, M., Sivakumar, A. I. ve Chandru, V. (2004). Scheduling algorithms and sensitivity analysis for heterogeneous batch processors with incompatible job families. *Journal of the Chinese Institute of Industrial Engineers*, 21(1), 18–26.
- Meindl, B. ve Templ, M. (2013). Analysis of commercial and free and open source solvers for linear optimization problems.
- Méndez, C. A., Cerdá, J., Grossmann, I. E., Harjunkoski, I. ve Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30(6–7), 913–946.
- Murgolo, F. D. (1987). An Efficient Approximation Scheme for Variable-Sized Bin Packing. *SIAM Journal on Computing*, 16(1), 149–161.

- Nahmias, S. ve Olsen, T. (2015). *Production and operations analysis : strategy, quality, analytics, application*.
- Palacio, J. D. ve Larrea, O. L. (2017). A lexicographic approach to the robust resource-constrained project scheduling problem. *International Transactions in Operational Research*, 24(1–2), 143–157.
- Parhizkary, M. (2006). Visual8 Compass SmartFlow for Composites Manufacturing Center. *Technology Review Journal, Fall/Winte*.
- Pichler, D. (2019, Ocak). The Carbon Fiber Market. *Composites Manufacturing*, 17–18.
- Ram, B. ve Patel, G. (1998). Modelling furnace operations using simulation and heuristics. *Simulation Conference Proceedings, 1998. Winter içinde* (C. 2, ss. 957–963). IEEE.
- Spearman, M. L., Woodruff, D. L. ve Hopp, W. J. (1990). CONWIP: a pull alternative to kanban. *International Journal of Production Research*, 28(5), 879–894.
- Taha, H. A. (2011). *Operations research : an introduction*. Prentice Hall.
- Tajan, J. B. C., Sivakumar, A. I. ve Gershwin, S. B. (2011). Control of a Single Batch Processor With Incompatible Job Families and Future Job Arrivals. *IEEE Transactions on Semiconductor Manufacturing*, 24(2), 208–222.
- Tamiz, M., Jones, D. ve Romero, C. (1998). Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research*, 111(3), 569–581.
- Valério de Carvalho, J. M. (2002). LP models for bin packing and cutting stock problems. *European Journal of Operational Research*, 141(2), 253–273.
- Winston, W. L. ve Goldberg, J. B. (2004). *Operations research : applications and algorithms*. Thomson/Brooks/Cole.

- Witten, E. (2019, Ocak). The European Market. *Composites Manufacturing*, 20–21. <http://compositesmanufacturingmagazine.com/digital/2019/CM-issue-january-february-2019.pdf> adresinden erişildi.
- Xavier, E. C. ve Miyazawa, F. K. (2008). The class constrained bin packing problem with applications to video-on-demand. *Theoretical Computer Science*, 393(1–3), 240–259.
- Ye, W. H., Li, J., Chen, W. F., Ma, W. T. ve Leng, S. (2014). Study on Scheduling Method for Reentrant Autoclave Moulding Operation of Composite Materials. *Applied Mechanics and Materials*, 490–491, 14–18.
- Yılmaz, U. ve Evcı, C. (2015). Havacılık ve Savunma Sektöründe Kompozit Malzemelerin Geleceği. *Savunma Bilimleri Dergisi*, 14(2), 77–109.
- Zee, D.-J. Vander, Harten, A. V. an ve Schuur, P. (2001). On-line scheduling of multi-server batch operations. *IIE Transactions*, 33(7), 569–586.

EK 1. Orijinallik Raporu

 <p>HACETTEPE ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU</p>
<p>HACETTEPE ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ İŞLETME ANABİLİM DALI BAŞKANLIĞI'NA</p> <p style="text-align: right;">Tarih: 19/06/2019</p> <p>Tez Başlığı : Kompozit Üretiminde Otoklav Yükleme Ve Çizelgeleme İçin Yeni Bir Model Önerisi: Savunma Sanayii Uygulaması</p> <p>Yukarıda başlığı gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 110 sayfalık kısmına ilişkin, 19/06/2019 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda işaretlenmiş filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 4 'tür.</p> <p>Uygulanan filtrelemeler:</p> <ol style="list-style-type: none"> 1- <input checked="" type="checkbox"/> Kabul/Onay ve Bildirim sayfaları hariç 2- <input checked="" type="checkbox"/> Kaynakça hariç 3- <input checked="" type="checkbox"/> Alıntılar hariç 4- <input type="checkbox"/> Alıntılar dâhil 5- <input checked="" type="checkbox"/> 5 kelimedenden daha az örtüşme içeren metin kısımları hariç <p>Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.</p> <p>Gereğini saygılarımla arz ederim.</p> <div style="text-align: right;">  19/06/2019 Tarih ve İmza </div> <p>Adı Soyadı: Volkan Haskalç _____</p> <p>Öğrenci No: 18508289742A _____</p> <p>Anabilim Dalı: İşletme _____</p> <p>Programı: İşletme _____</p> <p>Statüsü: <input checked="" type="checkbox"/> Doktora <input type="checkbox"/> Bütünleşik Dr. _____</p>
<p><u>DANIŞMAN ONAYI</u></p> <p>UYGUNDUR.</p>  <p>Prof. Dr. Aydın ULUCAN</p>

EK 2. Etik Kurul Muafiyet İzni

 <p>HACETTEPE ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ TEZ ÇALIŞMASI ETİK KOMİSYON MUAFİYETİ FORMU</p>
<p>HACETTEPE ÜNİVERSİTESİ SOSYAL BİLİMLER ENSTİTÜSÜ İŞLETME ANABİLİM DALI BAŞKANLIĞI'NA</p> <p style="text-align: right;">Tarih: 19/06/2019</p> <p>Tez Başlığı: Kompozit Üretiminde Otoklav Yükleme Ve Çizelgeleme İçin Yeni Bir Model Önerisi: Savunma Sanayii Uygulaması</p> <p>Yukarıda başlığı gösterilen tez çalışmam:</p> <ol style="list-style-type: none"> 1. İnsan ve hayvan üzerinde deney niteliği taşımamaktadır, 2. Biyolojik materyal (kan, idrar vb. biyolojik sıvılar ve numuneler) kullanılmasını gerektirmemektedir. 3. Beden bütünlüğüne müdahale içermemektedir. 4. Gözlemsel ve betimsel araştırma (anket, mülakat, ölçek/skala çalışmaları, dosya taramaları, veri kaynakları taraması, sistem-model geliştirme çalışmaları) niteliğinde değildir. <p>Hacettepe Üniversitesi Etik Kurullar ve Komisyonlarının Yönergelerini inceledim ve bunlara göre tez çalışmamın yürütülebilmesi için herhangi bir Etik Kurul/Komisyon'dan izin alınmasına gerek olmadığını; aksi durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.</p> <p>Gereğini saygılarımla arz ederim.</p> <p style="text-align: right;">  19/06/2019 Tarih ve İmza </p> <p> Adı Soyadı: Volkan Haskılıç Öğrenci No: 18508289742A Anabilim Dalı: İşletme Programı: İşletme Statüsü: <input type="checkbox"/> Yüksek Lisans <input checked="" type="checkbox"/> Doktora <input type="checkbox"/> Bütünleşik Doktora </p>
<p><u>DANIŞMAN GÖRÜŞÜ VE ONAYI</u></p> <p style="text-align: center;">  Prof. Dr. Aydın ULUCAN </p> <p> Detaylı Bilgi: http://www.sosyalbilimler.hacettepe.edu.tr Telefon: 0-312-2976860 Faks: 0-3122992147 E-posta: sosyalbilimler@hacettepe.edu.tr </p>

EK 3. Otoklav Yükleme Matematiksel Modeli Python Kaynak Kodu

```

from docplex.mp.model import Model
from docplex.util.environment import get_environment
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile
from openpyxl import load_workbook

def solveModel(noAutoclaves,noTools):
    NUMBER_OF_AUTOCLAVES=noAutoclaves
    NUMBER_OF_TOOLS=noTools

    mdl = Model(name="O%s_T%s_BP" % (NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS))

    path = "D:/OneDrive - hacettepe.edu.tr/doktora/python/tez/data/"
    dataFile = path + "data.xlsx"
    solutionFile="O%s_T%s_BP" % (NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS)
    dfParts = pd.read_excel(dataFile, sheet_name="Parts")
    dfTools = pd.read_excel(dataFile, sheet_name="Tools")
    dfDemand = pd.read_excel(dataFile, sheet_name="Demand")
    dfRecipes = pd.read_excel(dataFile, sheet_name="Recipes")
    dfAutoclaves = pd.read_excel(dataFile,
sheet_name="Autoclaves").head(NUMBER_OF_AUTOCLAVES)
    dfRecipesAutoclaves = pd.read_excel(dataFile, sheet_name="Recipes-
Autoclaves")

    # Kümeler
    P = dfParts['Part Id']
    O = dfAutoclaves['Autoclave Number']
    Z = range(1, 7)
    C = dfRecipes['Recipe Number']
    I = range(1, 6)

    # Parametreler
    T = dict.fromkeys(P, 0)

    for i in dfParts.index:

```



```

    for j in dfTools.index:
        if dfTools['Tool Id'][j]==dfParts['Tool Id'][i]:
            T[dfParts['Part Id'][i]]=NUMBER_OF_TOOLS #dfTools['Tools
Available'][j]

    CS = dict.fromkeys([(c, o) for c in C for o in O], 0)
    for i in dfRecipesAutoclaves.index:
        CS[dfRecipesAutoclaves['Recipe
Number'][i],dfRecipesAutoclaves['Autoclave
Number'][i]]=dfRecipesAutoclaves['Processing Time'][i]

    D = dict.fromkeys([(p, z) for p in P for z in Z], 0)

    for i in dfDemand.index:
        D[dfDemand['Part Id'][i], dfDemand['Time'][i]] =
dfDemand['Demand'][i]

    TCO = dict.fromkeys(O, 0)
    AO = dict.fromkeys(O, 0)
    for i in dfAutoclaves.index:
        TCO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Thermos'][i]
        AO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Area'][i]

    TCP = dict.fromkeys(P, 0)
    AP = dict.fromkeys(P, 0)
    CGP = dict.fromkeys(P, 1)
    for i in dfParts.index:
        TCP[dfParts['Part Id'][i]]=dfParts['Thermos'][i]
        AP[dfParts['Part Id'][i]]=dfParts['Area'][i]
        CGP[dfParts['Part Id'][i]]=dfParts['Recipe Number'][i]

    M = 9999
    B = dict.fromkeys([(p, o) for p in P for o in O], 1)

```

```

# Değişkenler
y = mdl.binary_var_matrix(I, 0, name="y")
q = mdl.integer_var_cube(P, I, 0, name="q")
x = mdl.binary_var_cube(P, I, 0, name="x")
w = mdl.binary_var_cube(C, I, 0, name="w")
UA = mdl.continuous_var_matrix(0, I, name="UA")
TUD = mdl.continuous_var_dict(P, name="TUD")

# Ağırlıklar
a1 = 1
a2 = 1000

DZ=dict.fromkeys(P, 0)
for p in P:
    DZ[p]=sum(D[p,z] for z in Z)

# Amaç fonksiyonu

mdl.obj = a1 * mdl.sum(UA[o, i]/A0[o]*100 for o in O for i in I) +
a2*mdl.sum(TUD[p]/DZ[p]*100 for p in P)

mdl.minimize(mdl.obj)

# Kısıtlar
# 1
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(q[p, i, o] <= M*y[i, o], "K1_%s.%s.%s" %
(p, i, o))

# 2
for p in P:
    for i in I:
        for o in O:

```

```

        mdl.add_constraint(q[p, i, o] <= M * x[p, i, o], "K2_%s.%s.%s"
% (p, i, o))

# 3
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(x[p, i, o] <= y[i, o], "K3_%s.%s.%s" %
(p, i, o))

# 4
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(AP[p] * q[p, i, o] for p in P) ==
AO[o] * y[i, o] - UA[o, i], "K4_%s.%s" % (i, o))

# 5
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(TCP[p] * q[p, i, o] for p in P) <=
TCO[o] * y[i, o], "K5_%s.%s" % (i, o))

# 6
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(q[p, i, o] <= T[p], "K6_%s.%s.%s" %
(p,i,o))

# 7
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(x[p, i, o] <= B[p, o], "K7_%s.%s.%s" %
(p,i,o))

# 8
for o in O:

```

```

        for i in I:
            if i > 1:
                mdl.add_constraint(y[i, o] <= y[i-1, o], "K8_%s.%s" % (o, i))
# 9
        for p in P:
            mdl.add_constraint(mdl.sum(q[p,i,o] for i in I for o in
O)==mdl.sum(D[p,z] for z in Z) - TUD[p],"K9_%s" % p)

# 10
        for i in I:
            for o in O:
                mdl.add_constraint(mdl.sum(w[c, i, o] for c in C) == y[i, o],
"K10_%s.%s" % (i, o))

# 11
        for i in I:
            for o in O:
                for c in C:
                    for p in P:
                        if CGP[p] != c:
                            mdl.add_constraint(w[c, i, o] + x[p, i, o] <= 1,
"K11_%s.%s.%s.%s" % (i, o, p, c))

        mdl.export_as_lp(path + "lp/", "tezBP_0%s_T%s" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS))

        if mdl.solve(log_output=True):

            excelWriter = pd.ExcelWriter(path + "xlsx/" + solutionFile + ".xlsx",
engine="xlsxwriter")

            edata = []
            edata.append(["Objective value",mdl.objective_value])

```

```

edata.append(["Solution staus",mdl.solve_details.status])
edata.append(["Solution time",mdl.solve_details.time])
edata.append(["MIP gap",mdl.solve_details.mip_relative_gap])
edata.append(["Binary variables",mdl.number_of_binary_variables])
edata.append(["Constraints",mdl.number_of_constraints])
edata.append(["Continuous
variables",mdl.number_of_continuous_variables])
edata.append(["Equivalence
constraints",mdl.number_of_equivalence_constraints])
edata.append(["Indicator
constraints",mdl.number_of_indicator_constraints])
edata.append(["Integer variables",mdl.number_of_integer_variables])
edata.append(["Linear constraints",mdl.number_of_linear_constraints])
edata.append(["Piecewise constraints",mdl.number_of_pwl_constraints])
edata.append(["Quadratic
constraints",mdl.number_of_quadratic_constraints])
edata.append(["Range constraints",mdl.number_of_range_constraints])
edata.append(["Semicontinuous
variables",mdl.number_of_semicontinuous_variables])
edata.append(["Semiinteger
variables",mdl.number_of_semiinteger_variables])
edata.append(["SOS",mdl.number_of_sos])
edata.append(["SOS1",mdl.number_of_sos1])
edata.append(["SOS2",mdl.number_of_sos2])
edata.append(["Variables",mdl.number_of_variables])

df=pd.DataFrame(columns=["Baslik","Deger"], data=edata)
df.to_excel(excelWriter, sheet_name="stat", index=False)
edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(i)
            row.append(o)
            edata.append(row)

```

```

df=pd.DataFrame(columns=["Yuk","Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="y", index=False)

edata=[]
for p in P:
    for i in I:
        for o in O:
            if q[p,i,o].solution_value>0:
                row=[]
                row.append(p)
                row.append(i)
                row.append(o)
                row.append(q[p,i,o].solution_value)
                edata.append(row)

df=pd.DataFrame(columns=["Parca","Yuk","Otoklav","Miktar"],
data=edata)
df.to_excel(excelWriter, sheet_name="q", index=False)

edata=[]
for p in P:
    for i in I:
        for o in O:
            if x[p,i,o].solution_value>0:
                row=[]
                row.append(p)
                row.append(i)
                row.append(o)
                edata.append(row)

df=pd.DataFrame(columns=["Parca","Yuk","Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="x", index=False)

edata = []
for c in C:
    for i in I:
        for o in O:

```

```

        if w[c,i,o].solution_value>0:
            row=[]
            row.append(c)
            row.append(i)
            row.append(o)
            edata.append(row)

df=pd.DataFrame(columns=["Kur","Yuk","Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="w", index=False)

edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(o)
            row.append(i)
            row.append(UA[o,i].solution_value)
            row.append(AO[o])
            row.append(1-UA[o,i].solution_value/AO[o])
            edata.append(row)

df=pd.DataFrame(columns=["Otoklav","Yuk","Alan","Toplam Alan","Alan
Kullanim"], data=edata)
df.to_excel(excelWriter, sheet_name="UA", index=False)
excelWriter.save()

excelWriterSummary = pd.ExcelWriter(path + "xlsx/Summary.xlsx",
engine="openpyxl")
excelWriterSummary.book = load_workbook(path + "xlsx/Summary.xlsx")
startrowSummary = excelWriterSummary.book["BP"].max_row
excelWriterSummary.sheets = {ws.title:ws for ws in
excelWriterSummary.book.worksheets}

```

```

excelWriterAll = pd.ExcelWriter(path + "xlsx/BP.xlsx",
engine="openpyxl")
excelWriterAll.book = load_workbook(path + "xlsx/BP.xlsx")
startrowAll = excelWriterAll.book["data"].max_row
excelWriterAll.sheets = {ws.title:ws for ws in
excelWriterAll.book.worksheets}

edata = []
row=[]
row.append(noAutoclaves)
row.append(noTools)
row.append mdl.objective_value)
row.append(mdl.solve_details.status)
row.append(mdl.solve_details.time)
row.append(mdl.solve_details.mip_relative_gap)
row.append(mdl.number_of_binary_variables)
row.append(mdl.number_of_constraints)
row.append(mdl.number_of_continuous_variables)
row.append(mdl.number_of_equivalence_constraints)
row.append(mdl.number_of_indicator_constraints)
row.append(mdl.number_of_integer_variables)
row.append(mdl.number_of_linear_constraints)
row.append(mdl.number_of_pwl_constraints)
row.append(mdl.number_of_quadratic_constraints)
row.append(mdl.number_of_range_constraints)
row.append(mdl.number_of_semicontinuous_variables)
row.append(mdl.number_of_semiinteger_variables)
row.append(mdl.number_of_sos)
row.append(mdl.number_of_sos1)
row.append(mdl.number_of_sos2)
row.append(mdl.number_of_variables)
edata.append(row)

df=pd.DataFrame(columns=["Otoklav", "Takım Sayısı", "Objective
value", "Solution status", "Solution time", "MIP gap", "Binary

```



```

variables", "Constraints", "Continuous variables", "Equivalence
constraints", "Indicator constraints", "Integer variables", "Linear
constraints", "Piecewise constraints", "Quadratic constraints", "Range
constraints", "Semicontinuous variables", "Semiinteger
variables", "SOS", "SOS1", "SOS2", "Variables"], data=edata)

df.to_excel(excelWriterSummary, sheet_name="BP",
startrow=startrowSummary, header=None, index=False)
excelWriterSummary.save()

edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(noAutoclaves)
            row.append(noTools)
            row.append(o)
            row.append(i)
            row.append(UA[o,i].solution_value)
            row.append(AO[o])
            row.append(1-UA[o,i].solution_value/AO[o])
            edata.append(row)

df=pd.DataFrame(columns=["Otoklav Sayisi", "Tool
Sayisi", "Otoklav", "Yuk", "Toplam Alan", "Alan", "Alan Kullanim"], data=edata)

df.to_excel(excelWriterAll, sheet_name="data", startrow=startrowAll,
header=None, index=False)
excelWriterAll.save()

# Save the CPLEX solution as "solution.json" program output
with get_environment().get_output_stream(path + "json/" +
solutionFile + ".json") as fp:
    mdl.solution.export(fp, "json")
else:
    print("Problem has no solution")

```

EK 4. Otoklav Çizelgeleme Matematiksel Modeli Python Kaynak Kodu

```

from docplex.mp.model import Model
from docplex.util.environment import get_environment
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile
from openpyxl import load_workbook

def solveModel(noAutoclaves,noTools):
    NUMBER_OF_AUTOCLAVES=noAutoclaves
    NUMBER_OF_TOOLS=noTools

    mdl = Model(name="tezSched_0%s_T%s" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS))
    path = "D:/OneDrive - hacettepe.edu.tr/doktora/python/tez/data/"
    dataFile = path + "data.xlsx"
    solutionFile="0%s_T%s_Sched" % (NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS)

    dfParts = pd.read_excel(dataFile, sheet_name="Parts")
    dfTools = pd.read_excel(dataFile, sheet_name="Tools")
    dfBlackout = pd.read_excel(dataFile, sheet_name="Blackout")
    dfDemand = pd.read_excel(dataFile, sheet_name="Demand")
    dfRecipes = pd.read_excel(dataFile, sheet_name="Recipes")
    dfAutoclaves = pd.read_excel(dataFile,
sheet_name="Autoclaves").head(NUMBER_OF_AUTOCLAVES)
    dfRecipesAutoclaves = pd.read_excel(dataFile, sheet_name="Recipes-
Autoclaves")

    dataFile = path + "xlsx/" + "0%s_T%s_BP" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS) + ".xlsx"
    dfY = pd.read_excel(dataFile, sheet_name="y")
    dfQ = pd.read_excel(dataFile, sheet_name="q")
    dfX = pd.read_excel(dataFile, sheet_name="x")
    dfW = pd.read_excel(dataFile, sheet_name="w")

```

```

# Kümeler
P = dfParts['Part Id']
O = dfAutoclaves['Autoclave Number']
Z = range(1, 7)
C = dfRecipes['Recipe Number']
I = range(1, 5)

# Parametreler
CST = dict.fromkeys(P, 0)
T = dict.fromkeys(P, 0)

for i in dfParts.index:
    for j in dfTools.index:
        if dfTools['Tool Id'][j]==dfParts['Tool Id'][i]:
            CST[dfParts['Part Id'][i]]=dfTools['Absolute Minimum Tool
Recycle Time'][j]
            T[dfParts['Part Id'][i]]=NUMBER_OF_TOOLS

CS = dict.fromkeys([(c, o) for c in C for o in O], 0)
for i in dfRecipesAutoclaves.index:
    CS[dfRecipesAutoclaves['Recipe
Number'][i],dfRecipesAutoclaves['Autoclave
Number'][i]]=dfRecipesAutoclaves['Processing Time'][i]

D = dict.fromkeys([(p, z) for p in P for z in Z], 0)

for i in dfDemand.index:
    D[dfDemand['Part Id'][i], dfDemand['Time'][i]] =
dfDemand['Demand'][i]

TCO = dict.fromkeys(O, 0)
AO = dict.fromkeys(O, 0)
for i in dfAutoclaves.index:
    TCO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Thermos'][i]
    AO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Area'][i]

TCP = dict.fromkeys(P, 0)

```

```

AP = dict.fromkeys(P, 0)
CGP = dict.fromkeys(P, 1)
for i in dfParts.index:
    TCP[dfParts['Part Id'][i]]=dfParts['Thermos'][i]
    AP[dfParts['Part Id'][i]]=dfParts['Area'][i]
    CGP[dfParts['Part Id'][i]]=dfParts['Recipe Number'][i]

CZ = dict.fromkeys(O, {})
M = 9999

y = dict.fromkeys([(i,o) for i in I for o in O],0)
for i in dfY.index:
    y[dfY["Yuk"][i],dfY["Otoklav"][i]]=1

q = dict.fromkeys([(p,i,o) for p in P for i in I for o in O],0)
for i in dfQ.index:
    q[dfQ["Parca"][i],dfQ["Yuk"][i],dfQ["Otoklav"][i]]=dfQ["Miktar"][i]

x = dict.fromkeys([(p,i,o) for p in P for i in I for o in O],0)
for i in dfX.index:
    x[dfX["Parca"][i],dfX["Yuk"][i],dfX["Otoklav"][i]]=1

w = dict.fromkeys([(c,i,o) for c in C for i in I for o in O],0)
for i in dfW.index:
    w[dfW["Kur"][i],dfW["Yuk"][i],dfW["Otoklav"][i]]=1

# Değişkenler
qz = mdl.integer_var_dict([(p, i, o, z) for p in P for i in I for o in O
for z in Z], name="qz")
Yuk = mdl.binary_var_cube(I, O, Z, name="Yuk")
TUD = mdl.continuous_var_dict(P, name="TUD")

DZ=dict.fromkeys(P, 0)
for p in P:
    DZ[p]=sum(D[p,z] for z in Z)

# Ağırlıklar

```

```

a1 = 1

# Amaç fonksiyonu
mdl.obj = a1 * mdl.sum(TUD[p]/DZ[p]*100 for p in P)

mdl.minimize(mdl.obj)

# Kısıtlar

# 1
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for z in Z) <= y[i, o],
"K1_%s.%s" % (i, o))

# 2
for p in P:
    mdl.add_constraint(mdl.sum(qz[p, i, o, z] for i in I for o in O for z
in Z) == mdl.sum(D[p, z] for z in Z) - TUD[p], "K2_%s" % p)

# 3
for o in O:
    for z in Z:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for i in I) <= 1,
"K3_%s.%s" % (o, z))

# 4
for o in O:
    if len(CZ[o]) > 0:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for i in I for z in
CZ[o]) == 0, "K4_%s" % o)

# 5
for i in I:
    for o in O:

```

```

        for z in Z:
            mdl.add_constraint(mdl.sum(Yuk[i1, o, z1]*w[c, i1, o] for c
in C for i1 in I if i1 != i for z1 in Z if z <= z1 <= z + CS[c, o]) <= M * (1
- Yuk[i, o, z]), "K5_%s.%s.%s" % (i, o, z))

# 7
for z in Z:
    for p in P:
        # Tool cure cycle + tool cycle süresince kullanılır
        mdl.add_constraint(mdl.sum(qz[p, i, o, min(z + t, Z[-1])]) for i
in I for o in O for t in range(0, min(CS[CGP[p], o]+CST[p], Z[-1]-z)+1)) <=
T[p], "K7_%s.%s" % (z, p))

# 8
for p in P:
    for i in I:
        for o in O:
            for z in Z:
                mdl.add_indicator(Yuk[i, o, z], qz[p, i, o, z] - q[p, i,
o]==0,1, "K8_%s.%s.%s.%s" % (p,i,o,z))

# 9
for i in I:
    for o in O:
        for z in Z:
            mdl.add_indicator(Yuk[i,o,z], mdl.sum(qz[p,i,o,z] for p in
P)==0,0, "K18_%s.%s.%s" % (i,o,z))

# 10
for i in I:
    for o in O:
        for z in Z:
            mdl.add_constraint(Yuk[i, o, z]<=mdl.sum(qz[p, i, o, z] for p
in P), "K19_%s.%s.%s" % (i,o,z))

mdl.export_as_lp(path + "lp/", "0%s_T%s_Sched" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS))

```

```

if mdl.solve(log_output=True):

    excelWriter = pd.ExcelWriter(path + "xlsx/" + solutionFile + ".xlsx",
engine="xlsxwriter")

    edata = []
    edata.append(["Objective value",mdl.objective_value])
    edata.append(["Solution status",mdl.solve_details.status])
    edata.append(["Solution time",mdl.solve_details.time])
    edata.append(["MIP gap",mdl.solve_details.mip_relative_gap])
    edata.append(["Binary variables",mdl.number_of_binary_variables])
    edata.append(["Constraints",mdl.number_of_constraints])
    edata.append(["Continuous
variables",mdl.number_of_continuous_variables])
    edata.append(["Equivalence
constraints",mdl.number_of_equivalence_constraints])
    edata.append(["Indicator
constraints",mdl.number_of_indicator_constraints])
    edata.append(["Integer variables",mdl.number_of_integer_variables])
    edata.append(["Linear constraints",mdl.number_of_linear_constraints])
    edata.append(["Piecewise constraints",mdl.number_of_pwl_constraints])
    edata.append(["Quadratic
constraints",mdl.number_of_quadratic_constraints])
    edata.append(["Range constraints",mdl.number_of_range_constraints])
    edata.append(["Semicontinuous
variables",mdl.number_of_semicontinuous_variables])
    edata.append(["Semiinteger
variables",mdl.number_of_semiinteger_variables])
    edata.append(["SOS",mdl.number_of_sos])
    edata.append(["SOS1",mdl.number_of_sos1])
    edata.append(["SOS2",mdl.number_of_sos2])
    edata.append(["Variables",mdl.number_of_variables])

df=pd.DataFrame(columns=["Baslik", "Deger"], data=edata)

```

```

df.to_excel(excelWriter, sheet_name="stat", index=False)

edata = []

for p in P:
    for i in I:
        for o in O:
            for z in Z:
                if qz[p,i,o,z].solution_value>0:
                    row=[]
                    row.append(p)
                    row.append(i)
                    row.append(o)
                    row.append(z)
                    row.append(qz[p,i,o,z].solution_value)
                    edata.append(row)

df=pd.DataFrame(columns=["Parca", "Yuk", "Otoklav", "Zaman", "Miktar"],
data=edata)
df.to_excel(excelWriter, sheet_name="qz", index=False)

edata = []
for i in I:
    for o in O:
        for z in Z:
            if Yuk[i,o,z].solution_value>0:
                row=[0,0,0,0,0]
                row[0]=o
                row[1]=i
                row[2]=z
                for c in C:
                    if w[c,i,o]==1:
                        row[3]=c
                        row[4]=CS[c,o]

```



```

        edata.append(row)

df=pd.DataFrame(columns=["Otoklav", "Yuk", "Zaman", "Kur Grubu", "Kur
Suresi"], data=edata)
df.to_excel(excelWriter, sheet_name="Yuk", index=False)

edata = []
for p in P:
    tmpsum=sum(D[p,z] for z in Z)
    if tmpsum>0:
        row=[]
        row.append(p)
        row.append(TUD[p].solution_value)
        row.append(tmpsum)
        row.append(1-TUD[p].solution_value/(tmpsum if tmpsum>0 else
1))

        edata.append(row)

df=pd.DataFrame(columns=["Parca", "Miktar", "Toplam
Miktar", "Karsilama"], data=edata)
df.to_excel(excelWriter, sheet_name="TUD", index=False)
excelWriter.save()

excelWriterSummary = pd.ExcelWriter(path + "xlsx/Summary.xlsx",
engine="openpyxl")
excelWriterSummary.book = load_workbook(path + "xlsx/Summary.xlsx")
startrowSummary = excelWriterSummary.book["Sched"].max_row
excelWriterSummary.sheets = {ws.title:ws for ws in
excelWriterSummary.book.worksheets}

excelWriterAll = pd.ExcelWriter(path + "xlsx/Sched.xlsx",
engine="openpyxl")
excelWriterAll.book = load_workbook(path + "xlsx/Sched.xlsx")
startrowSched = excelWriterAll.book["Sched"].max_row
startrowDemand = excelWriterAll.book["Demand"].max_row

```

```

    excelWriterAll.sheets = {ws.title:ws for ws in
excelWriterAll.book.worksheets}

    edata = []
    row=[]
    row.append(noAutoclaves)
    row.append(noTools)
    row.append mdl.objective_value)
    row.append(mdl.solve_details.status)
    row.append(mdl.solve_details.time)
    row.append(mdl.solve_details.mip_relative_gap)
    row.append(mdl.number_of_binary_variables)
    row.append(mdl.number_of_constraints)
    row.append(mdl.number_of_continuous_variables)
    row.append(mdl.number_of_equivalence_constraints)
    row.append(mdl.number_of_indicator_constraints)
    row.append(mdl.number_of_integer_variables)
    row.append(mdl.number_of_linear_constraints)
    row.append(mdl.number_of_pwl_constraints)
    row.append(mdl.number_of_quadratic_constraints)
    row.append(mdl.number_of_range_constraints)
    row.append(mdl.number_of_semicontinuous_variables)
    row.append(mdl.number_of_semiinteger_variables)
    row.append(mdl.number_of_sos)
    row.append(mdl.number_of_sos1)
    row.append(mdl.number_of_sos2)
    row.append(mdl.number_of_variables)
    edata.append(row)

    df=pd.DataFrame(columns=["Otoklav", "Takım Sayısı", "Objective
value", "Solution status", "Solution time", "MIP gap", "Binary
variables", "Constraints", "Continuous variables", "Equivalence
constraints", "Indicator constraints", "Integer variables", "Linear
constraints", "Piecewise constraints", "Quadratic constraints", "Range
constraints", "Semicontinuous variables", "Semiinteger
variables", "SOS", "SOS1", "SOS2", "Variables"], data=edata)

```

```

df.to_excel(excelWriterSummary, sheet_name="Sched",
startrow=startrowSummary, header=None, index=False)
excelWriterSummary.save()

edata = []
for i in I:
    for o in O:
        for z in Z:
            if Yuk[i,o,z].solution_value>0:
                row=[0,0,0,0,0,0,0]
                row[0]=noAutoclaves
                row[1]=noTools
                row[2]=o
                row[3]=i
                row[4]=z
                for c in C:
                    if w[c,i,o]==1:
                        row[5]=c
                        row[6]=CS[c,o]

                edata.append(row)

df=pd.DataFrame(columns=["Otoklav Sayisi", "Tool
Sayisi", "Otoklav", "Yuk", "Zaman", "Kur Grubu", "Kur Suresi"], data=edata)
df.to_excel(excelWriterAll, sheet_name="Sched",
startrow=startrowSched, header=None, index=False)

edata = []
for p in P:
    tmpsum=sum(D[p,z] for z in Z)
    if tmpsum>0:
        row=[]
        row.append(noAutoclaves)
        row.append(noTools)
        row.append(p)
        row.append(TUD[p].solution_value)

```

```

        row.append(tmpsum)
        row.append(1-TUD[p].solution_value/(tmpsum if tmpsum>0 else
1))

        edata.append(row)

df=pd.DataFrame(columns=["Otoklav Sayisi", "Tool
Sayisi", "Parca", "Miktar", "Toplam Miktar", "Karsilama"], data=edata)
df.to_excel(excelWriterAll, sheet_name="Demand",
startrow=startrowDemand, header=None, index=False)
excelWriter.save()

excelWriterAll.save()

with get_environment().get_output_stream(path + "json/" +
solutionFile + ".json") as fp:
    mdl.solution.export(fp, "json")
else:
    print("Problem has no solution")

```

EK 5. Otoklav Yükleme ve Çizelgeleme Matematiksel Modeli Python Kaynak Kodu

```

from docplex.mp.model import Model
from docplex.util.environment import get_environment
import pandas as pd
from pandas import ExcelWriter
from pandas import ExcelFile
from openpyxl import load_workbook

def solveModel(noAutoclaves,noTools,fileTemplate):
    NUMBER_OF_AUTOCLAVES=noAutoclaves
    NUMBER_OF_TOOLS=noTools

    path = "D:/OneDrive - hacettepe.edu.tr/doktora/python/tez/data/"
    dataFile = path + "data_"+fileTemplate+".xlsx"
    solutionFile="O%s_T%s_%s" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS,fileTemplate)
    dfParts = pd.read_excel(dataFile, sheet_name="Parts")
    dfTools = pd.read_excel(dataFile, sheet_name="Tools")
    dfBlackout = pd.read_excel(dataFile, sheet_name="Blackout")
    dfDemand = pd.read_excel(dataFile, sheet_name="Demand")
    dfRecipes = pd.read_excel(dataFile, sheet_name="Recipes")
    dfAutoclaves = pd.read_excel(dataFile,
sheet_name="Autoclaves").head(NUMBER_OF_AUTOCLAVES)
    dfRecipesAutoclaves = pd.read_excel(dataFile, sheet_name="Recipes-
Autoclaves")

    mdl = Model(name="O%s_T%s_%s" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS,fileTemplate))
    # Kümeler
    P = dfParts['Part Id']
    O = dfAutoclaves['Autoclave Number']
    Z = range(1, 55)
    C = dfRecipes['Recipe Number']
    I = range(1, 6)

```

```

# Parametreler
CST = dict.fromkeys(P, 0)
T = dict.fromkeys(P, 0)

for i in dfParts.index:
    for j in dfTools.index:
        if dfTools['Tool Id'][j]==dfParts['Tool Id'][i]:
            CST[dfParts['Part Id'][i]]=dfTools['Absolute Minimum Tool
Recycle Time'][j]
            T[dfParts['Part Id'][i]]= NUMBER_OF_TOOLS

CS = dict.fromkeys([(c, o) for c in C for o in O], 0)
for i in dfRecipesAutoclaves.index:
    CS[dfRecipesAutoclaves['Recipe
Number'][i],dfRecipesAutoclaves['Autoclave
Number'][i]]=dfRecipesAutoclaves['Processing Time'][i]

D = dict.fromkeys([(p, z) for p in P for z in Z], 0)

for i in dfDemand.index:
    D[dfDemand['Part Id'][i], dfDemand['Time'][i]] =
dfDemand['Demand'][i]

TCO = dict.fromkeys(O, 0)
AO = dict.fromkeys(O, 0)
for i in dfAutoclaves.index:
    TCO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Thermos'][i]
    AO[dfAutoclaves['Autoclave Number'][i]]=dfAutoclaves['Area'][i]

TCP = dict.fromkeys(P, 0)
AP = dict.fromkeys(P, 0)
CGP = dict.fromkeys(P, 1)
for i in dfParts.index:
    TCP[dfParts['Part Id'][i]]=dfParts['Thermos'][i]
    AP[dfParts['Part Id'][i]]=dfParts['Area'][i]
    CGP[dfParts['Part Id'][i]]=dfParts['Recipe Number'][i]

```

```

CZ = dict.fromkeys(O, {})
M = 99
B = dict.fromkeys([(p, o) for p in P for o in O], 1)

# Değişkenler
y = mdl.binary_var_matrix(I, O, name="y")
q = mdl.integer_var_cube(P, I, O, name="q")
qz = mdl.integer_var_dict([(p, i, o, z) for p in P for i in I for o in O
for z in Z], name="qz")
x = mdl.binary_var_cube(P, I, O, name="x")
Yuk = mdl.binary_var_cube(I, O, Z, name="Yuk")
w = mdl.binary_var_cube(C, I, O, name="w")
TUD = mdl.continuous_var_dict(P, name="TUD")
UA = mdl.continuous_var_matrix(O, I, name="UA")
UDP = mdl.continuous_var_matrix(P, Z, name="UDP")
UDM = mdl.continuous_var_matrix(P, Z, name="UDM")

# Ağırlıklar
a1 = 1
a2 = 1000

DZ=dict.fromkeys(P, 0)
for p in P:
    DZ[p]=sum(D[p,z] for z in Z)

# Amaç fonksiyonu
mdl.obj = a1 * mdl.sum(UA[o, i]/AO[o]*100 for o in O for i in I) + a2 *
mdl.sum(TUD[p]/DZ[p]*100 for p in P) + a2 * mdl.sum(UDP[p,z]/D[p,z]*100 for p
in P for z in Z if D[p,z]>0) + a3 * mdl.sum(UDM[p,z]/D[p,z]*100 for p in P
for z in Z if D[p,z]>0)

mdl.minimize(mdl.obj)

# Kısıtlar
# 1

```

```

for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(q[p, i, o] <= M*y[i, o], "K1_%s.%s.%s" %
(p, i, o))

# 2
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(q[p, i, o] <= M * x[p, i, o], "K2_%s.%s.%s"
% (p, i, o))

# 3
for p in P:
    for i in I:
        for o in O:
            mdl.add_constraint(x[p, i, o] <= y[i, o], "K3_%s.%s.%s" %
(p, i, o))

# 4
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(AP[p] * q[p, i, o] for p in P) ==
AO[o] * y[i, o] - UA[o, i], "K4_%s.%s" % (i, o))

# 5
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(TCP[p] * q[p, i, o] for p in P) <=
TCO[o] * y[i, o], "K5_%s.%s" % (i, o))

# 6
for p in P:
    for i in I:
        for o in O:

```



```

        mdl.add_constraint(x[p, i, o] <= B[p, o], "K6_%s.%s.%s" %
(p,i,o))

# 7
for o in O:
    for i in I:
        if i > 1:
            mdl.add_constraint(y[i, o] <= y[i-1, o], "K7_%s.%s" % (o, i))

# 8
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(w[c, i, o] for c in C) == y[i, o],
"K8_%s.%s" % (i, o))

# 9
for i in I:
    for o in O:
        for c in C:
            for p in P:
                if CGP[p] != c:
                    mdl.add_constraint(w[c, i, o] + x[p, i, o] <= 1,
"K9_%s.%s.%s.%s" % (i, o, p, c))

# 10
for i in I:
    for o in O:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for z in Z) <= y[i, o],
"K10_%s.%s" % (i, o))

# 11
for p in P:
    mdl.add_constraint(mdl.sum(qz[p, i, o, z] for i in I for o in O for z
in Z) == mdl.sum(D[p, z] for z in Z) - TUD[p], "K11_%s" % p)

# 11
for p in P:

```

```

        for z in Z:
            mdl.add_constraint(mdl.sum(qz[p, i, o, z] for i in I for o in O)
== D[p, z] - UDM[p,z] + UDP[p,z], "K11.1_%s.%s" % (p,z))

# 12
for o in O:
    for z in Z:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for i in I) <= 1,
"K12_%s.%s" % (o, z))

# 13
if len(CZ[o]) > 0:
    for o in O:
        mdl.add_constraint(mdl.sum(Yuk[i, o, z] for i in I for z in
CZ[o]) == 0, "K13_%s" % o)

# 14
for i in I:
    for o in O:
        for z in Z:
            mdl.add_constraint(mdl.sum(Yuk[i1, o, z1]*w[c, i1, o] for c
in C for i1 in I if i1 != i for z1 in Z if z <= z1 <= z + CS[c, o]) <= M * (1
- Yuk[i, o, z]), "K14_%s.%s.%s" % (i, o, z))

# 15
for z in Z:
    for p in P:
        # Tool cure cycle + tool cycle süresince kullanılır
        mdl.add_constraint(mdl.sum(qz[p, i, o, min(z + t, Z[-1])]) for i
in I for o in O for t in range(0, min(CS[CGP[p], o]+CST[p], Z[-1]-z)+1)) <=
T[p], "K16_%s.%s" % (z, p))

# 16
for p in P:
    for i in I:

```

```

        for o in O:
            for z in Z:
                mdl.add_indicator(Yuk[i, o, z], qz[p, i, o, z] - q[p, i,
o]==0,1, "K17_%.%.%.%" % (p,i,o,z))

# 17
for i in I:
    for o in O:
        for z in Z:
            mdl.add_indicator(Yuk[i,o,z],mdl.sum(qz[p,i,o,z] for p in
P)==0,0, "K18_%.%.%" % (i,o,z))

# 18
for i in I:
    for o in O:
        for z in Z:
            mdl.add_constraint(Yuk[i, o, z]<=mdl.sum(qz[p, i, o, z] for p
in P), "K19_%.%.%" % (i,o,z))

mdl.export_as_lp(path + "lp/", "0%s_T%s" %
(NUMBER_OF_AUTOCLAVES,NUMBER_OF_TOOLS))

if mdl.solve(log_output=True):

    excelWriter = pd.ExcelWriter(path + "xlsx/" + solutionFile + ".xlsx",
engine="xlsxwriter")

    edata = []
    edata.append(["Objective value",mdl.objective_value])
    edata.append(["Solution staus",mdl.solve_details.status])
    edata.append(["Solution time",mdl.solve_details.time])
    edata.append(["MIP gap",mdl.solve_details.mip_relative_gap])
    edata.append(["Binary variables",mdl.number_of_binary_variables])
    edata.append(["Constraints",mdl.number_of_constraints])

```

```

        edata.append(["Continuous
variables",mdl.number_of_continuous_variables])
        edata.append(["Equivalence
constraints",mdl.number_of_equivalence_constraints])
        edata.append(["Indicator
constraints",mdl.number_of_indicator_constraints])
        edata.append(["Integer variables",mdl.number_of_integer_variables])
        edata.append(["Linear constraints",mdl.number_of_linear_constraints])
        edata.append(["Piecewise constraints",mdl.number_of_pwl_constraints])
        edata.append(["Quadratic
constraints",mdl.number_of_quadratic_constraints])
        edata.append(["Range constraints",mdl.number_of_range_constraints])
        edata.append(["Semicontinuous
variables",mdl.number_of_semicontinuous_variables])
        edata.append(["Semiinteger
variables",mdl.number_of_semiinteger_variables])
        edata.append(["SOS",mdl.number_of_sos])
        edata.append(["SOS1",mdl.number_of_sos1])
        edata.append(["SOS2",mdl.number_of_sos2])
        edata.append(["Variables",mdl.number_of_variables])

df=pd.DataFrame(columns=["Baslik","Deger"], data=edata)
df.to_excel(excelWriter, sheet_name="stat", index=False)

edata = []

for p in P:
    for i in I:
        for o in O:
            for z in Z:
                if qz[p,i,o,z].solution_value>0:
                    row=[]
                    row.append(p)
                    row.append(i)
                    row.append(o)
                    row.append(z)
                    row.append(qz[p,i,o,z].solution_value)

```

```

        edata.append(row)

df=pd.DataFrame(columns=["Parca", "Yuk", "Otoklav", "Zaman", "Miktar"],
data=edata)
df.to_excel(excelWriter, sheet_name="qz", index=False)

edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(i)
            row.append(o)
            edata.append(row)

df=pd.DataFrame(columns=["Yuk", "Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="y", index=False)

edata=[]
for p in P:
    for i in I:
        for o in O:
            if q[p,i,o].solution_value>0:
                row=[]
                row.append(p)
                row.append(i)
                row.append(o)
                row.append(q[p,i,o].solution_value)
                edata.append(row)

df=pd.DataFrame(columns=["Parca", "Yuk", "Otoklav", "Miktar"],
data=edata)
df.to_excel(excelWriter, sheet_name="q", index=False)

edata=[]
for p in P:
    for i in I:
        for o in O:

```

```

if x[p,i,o].solution_value>0:
    row=[]
    row.append(p)
    row.append(i)
    row.append(o)
    edata.append(row)

```

```

df=pd.DataFrame(columns=["Parca","Yuk","Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="x", index=False)

```

```

edata = []
for i in I:
    for o in O:
        for z in Z:
            if Yuk[i,o,z].solution_value>0:
                row=[]
                row.append(o)
                row.append(i)
                row.append(z)
                for c in C:
                    if w[c,i,o].solution_value==1:
                        row.append(c)
                        row.append(CS[c,o])
                        break

                edata.append(row)

```

```

df=pd.DataFrame(columns=["Otoklav","Yuk","Zaman","Kur Grubu","Kur
Suresi"], data=edata)
df.to_excel(excelWriter, sheet_name="Yuk", index=False)

```

```

edata = []
for c in C:
    for i in I:
        for o in O:
            if w[c,i,o].solution_value>0:

```

```

        row=[]
        row.append(c)
        row.append(i)
        row.append(o)
        edata.append(row)

df=pd.DataFrame(columns=["Kur","Yuk","Otoklav"], data=edata)
df.to_excel(excelWriter, sheet_name="w", index=False)

edata = []
for p in P:
    if sum(D[p,z] for z in Z)>0:
        row=[]
        row.append(p)
        row.append(TUD[p].solution_value)
        row.append(sum(D[p,z] for z in Z))
        row.append(1-TUD[p].solution_value/sum(D[p,z] for z in Z))
        edata.append(row)

df=pd.DataFrame(columns=["Parca","Miktar","Toplam
Miktar","Karsilama"], data=edata)
df.to_excel(excelWriter, sheet_name="TUD", index=False)

edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(o)
            row.append(i)
            row.append(UA[o,i].solution_value)
            row.append(AO[o])
            row.append(1-UA[o,i].solution_value/AO[o])
            edata.append(row)

df=pd.DataFrame(columns=["Otoklav","Yuk","Alan","Toplam Alan","Alan
Kullanim"], data=edata)

```

```

df.to_excel(excelWriter, sheet_name="UA", index=False)
excelWriter.save()

excelWriterSummary = pd.ExcelWriter(path + "xlsx/Summary.xlsx",
engine="openpyxl")
excelWriterSummary.book = load_workbook(path + "xlsx/Summary.xlsx")
startrowSummary = excelWriterSummary.book["Both"].max_row
excelWriterSummary.sheets = {ws.title:ws for ws in
excelWriterSummary.book.worksheets}

excelWriterAll = pd.ExcelWriter(path + "xlsx/Both.xlsx",
engine="openpyxl")
excelWriterAll.book = load_workbook(path + "xlsx/Both.xlsx")

startrowAllBP = excelWriterAll.book["BP"].max_row
startrowAllSched = excelWriterAll.book["Sched"].max_row
startrowAllDemand = excelWriterAll.book["Demand"].max_row
excelWriterAll.sheets = {ws.title:ws for ws in
excelWriterAll.book.worksheets}

edata = []
row=[]
row.append(noAutoclaves)
row.append(noTools)
row.append(fileTemplate)
row.append mdl.objective_value)
row.append(mdl.solve_details.status)
row.append(mdl.solve_details.time)
row.append(mdl.solve_details.mip_relative_gap)
row.append(mdl.number_of_binary_variables)
row.append(mdl.number_of_constraints)
row.append(mdl.number_of_continuous_variables)
row.append(mdl.number_of_equivalence_constraints)
row.append(mdl.number_of_indicator_constraints)
row.append(mdl.number_of_integer_variables)
row.append(mdl.number_of_linear_constraints)

```



```

row.append mdl.number_of_pwl_constraints)
row.append mdl.number_of_quadratic_constraints)
row.append mdl.number_of_range_constraints)
row.append mdl.number_of_semicontinuous_variables)
row.append mdl.number_of_semiinteger_variables)
row.append mdl.number_of_sos)
row.append mdl.number_of_sos1)
row.append mdl.number_of_sos2)
row.append mdl.number_of_variables)
edata.append(row)

```

```

df=pd.DataFrame(columns=["Otoklav", "Takım Sayısı", "Tarih", "Objective
value", "Solution status", "Solution time", "MIP gap", "Binary
variables", "Constraints", "Continuous variables", "Equivalence
constraints", "Indicator constraints", "Integer variables", "Linear
constraints", "Piecewise constraints", "Quadratic constraints", "Range
constraints", "Semicontinuous variables", "Semiinteger
variables", "SOS", "SOS1", "SOS2", "Variables"], data=edata)

```

```

df.to_excel(excelWriterSummary, sheet_name="Both",
startrow=startrowSummary, header=None, index=False)
excelWriterSummary.save()

```

```

edata=[]
for i in I:
    for o in O:
        if y[i,o].solution_value>0:
            row=[]
            row.append(noAutoclaves)
            row.append(noTools)
            row.append(fileTemplate)
            row.append(o)
            row.append(i)
            row.append(UA[o,i].solution_value)
            row.append(AO[o])
            row.append(1-UA[o,i].solution_value/AO[o])
            edata.append(row)

```

```

df=pd.DataFrame(columns=["Otoklav Sayisi","Tool
Sayisi","Tarih","Otoklav","Yuk","Toplam Alan","Alan","Alan Kullanim"],
data=edata)

df.to_excel(excelWriterAll, sheet_name="BP", startrow=startrowAllBP,
header=None, index=False)

```

```

edata = []
for i in I:
    for o in O:
        for z in Z:
            if Yuk[i,o,z].solution_value>0:
                row=[0,0,0,0,0,0,0,0]
                row[0]=noAutoclaves
                row[1]=noTools
                row[2]=fileTemplate
                row[3]=o
                row[4]=i
                row[5]=z
                for c in C:
                    if w[c,i,o].solution_value == 1:
                        row[6]=c
                        row[7]=CS[c,o]

                edata.append(row)

```

```

df=pd.DataFrame(columns=["Otoklav Sayisi","Tool
Sayisi","Tarih","Otoklav","Yuk","Zaman","Kur Grubu","Kur Suresi"],
data=edata)

df.to_excel(excelWriterAll, sheet_name="Sched",
startrow=startrowAllSched, header=None, index=False)

```

```

edata = []
for p in P:
    if sum(D[p,z] for z in Z)>0:
        row=[]

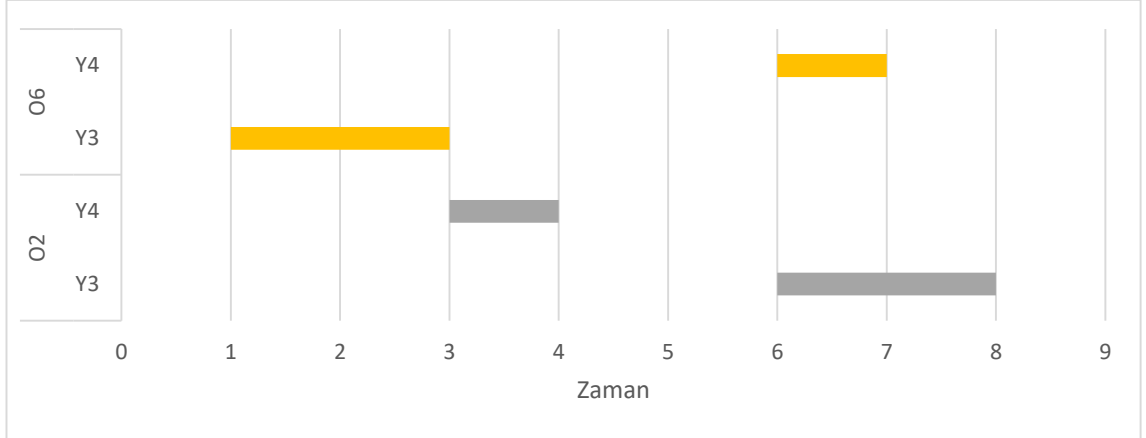
```

```
        row.append(noAutoclaves)
        row.append(noTools)
        row.append(fileTemplate)
        row.append(p)
        row.append(TUD[p].solution_value)
        row.append(sum(D[p,z] for z in Z))
        row.append(1-TUD[p].solution_value/sum(D[p,z] for z in Z))
    edata.append(row)

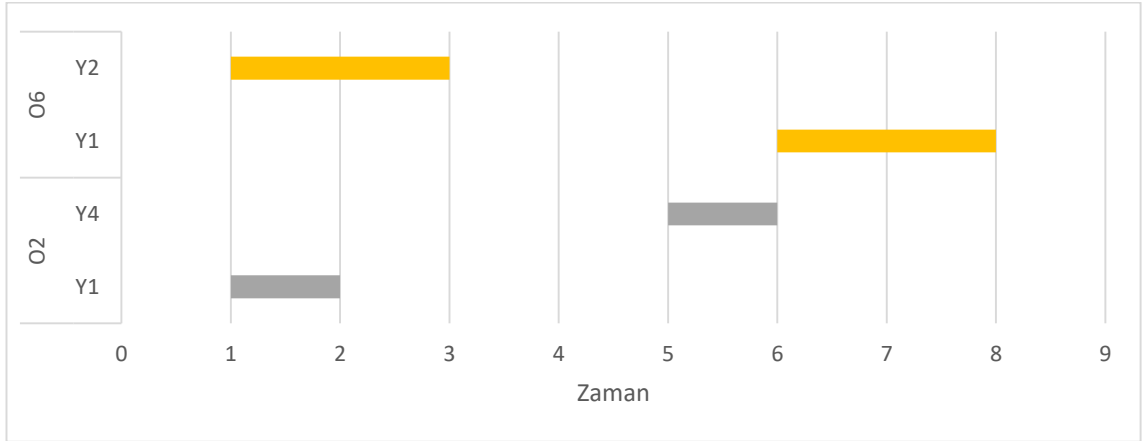
    df=pd.DataFrame(columns=["Otoklav Sayisi", "Tool
Sayisi", "Tarih", "Parca", "Miktar", "Toplam Miktar", "Karsilama"], data=edata)
    df.to_excel(excelWriterAll, sheet_name="Demand",
startrow=startrowAllDemand, header=None, index=False)

    excelWriterAll.save()
    with get_environment().get_output_stream(path + "json/" +
solutionFile + ".json") as fp:
        mdl.solution.export(fp, "json")
    else:
        print("Problem has no solution")
```

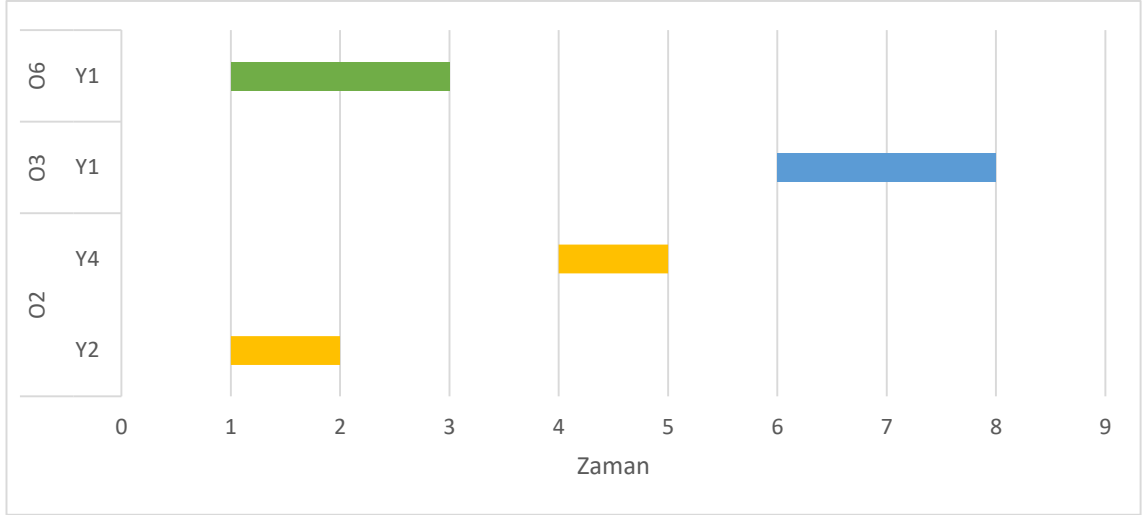
EK 6. Bağımsız Model Gantt Şemaları



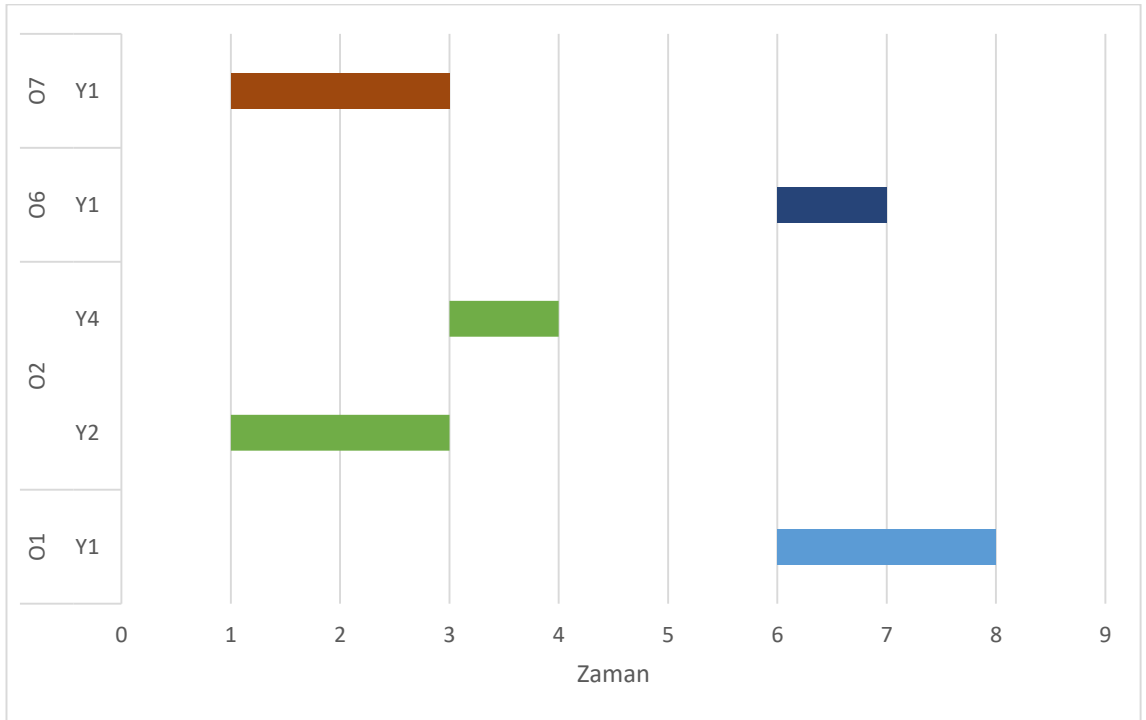
Şekil E6.1 Bir Takım Olması Durumunda Çizelge



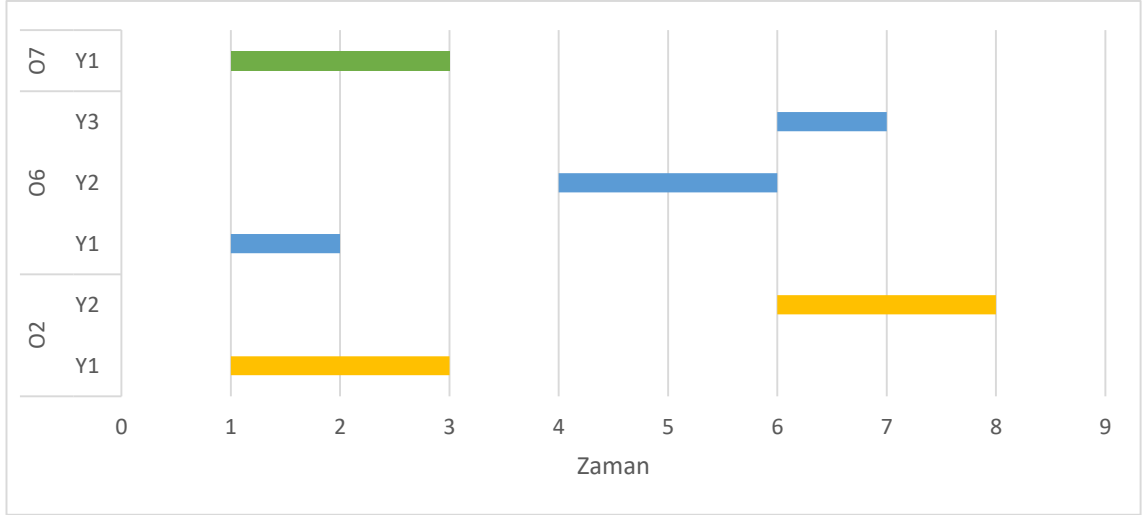
Şekil E6.2 İki Takım Olması Durumunda Çizelge



Şekil E6.3 Üç Takım Olması Durumunda Çizelge

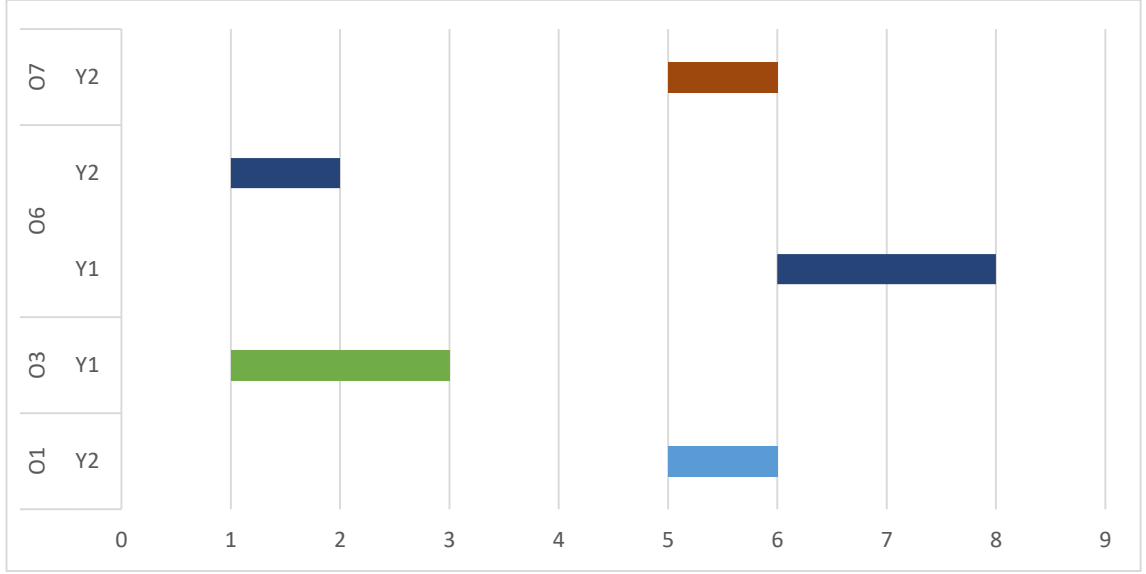


Şekil E6.4 Dört Takım Olması Durumunda Çizelge

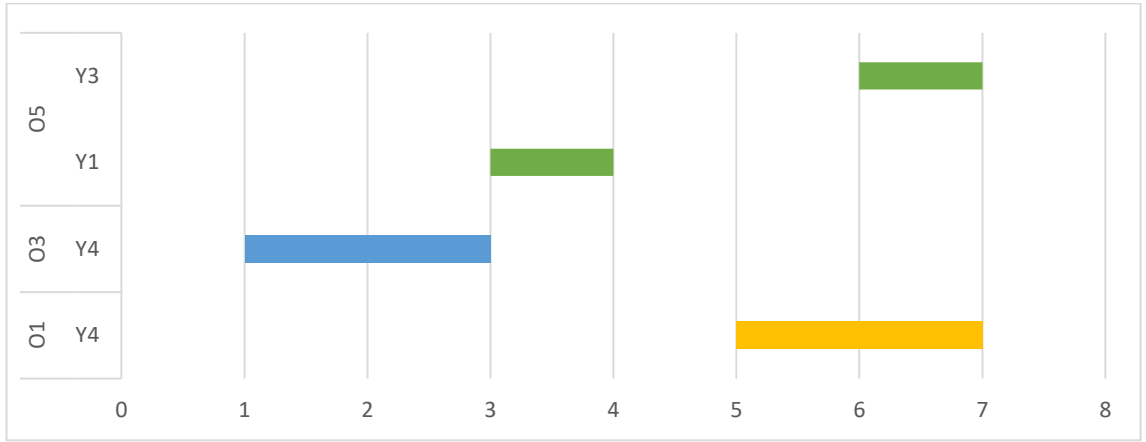


Şekil E6.5 Beş Takım Olması Durumunda Çizelge

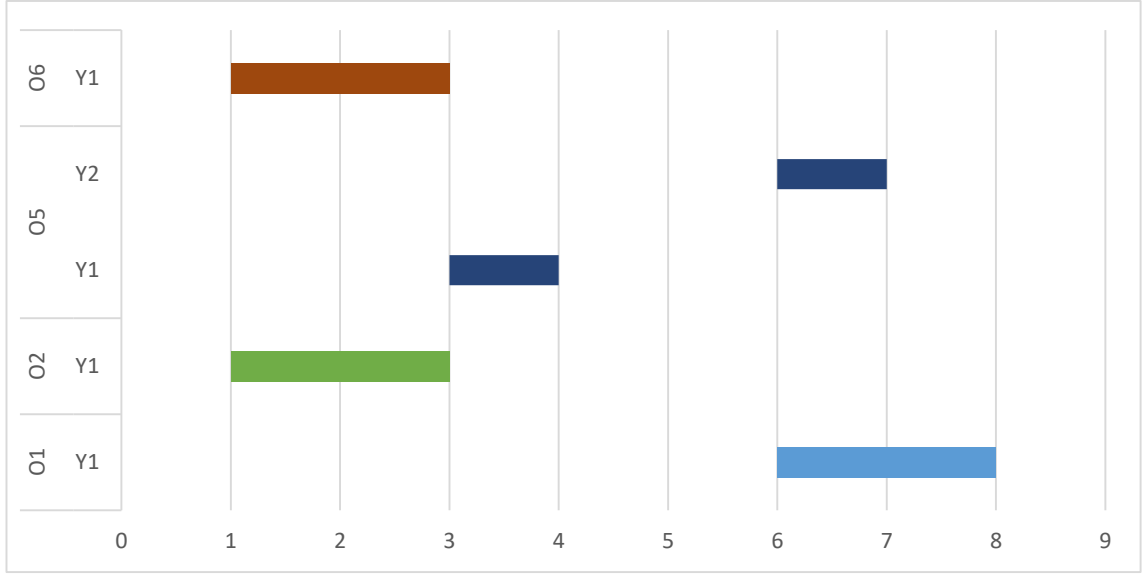
EK 7. Tek Model Gantt Şemaları



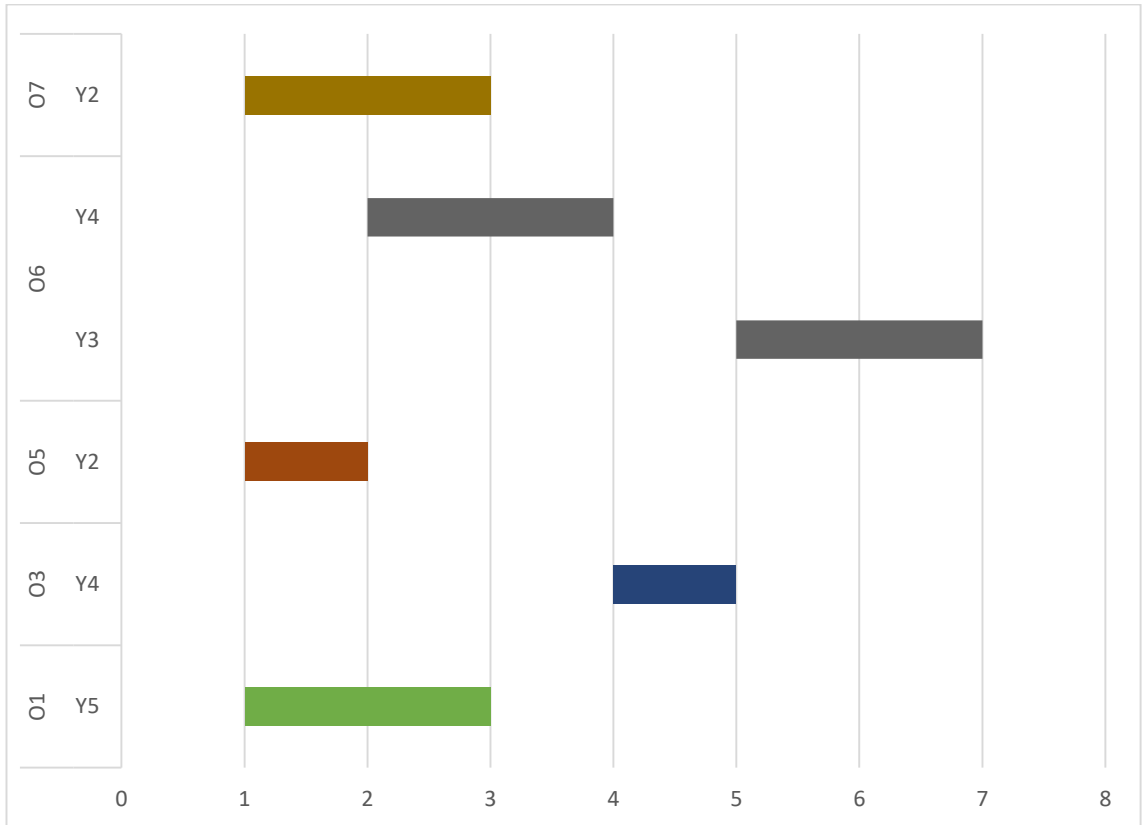
Şekil E7.1 Bir Takım Olması Durumunda Çizelge



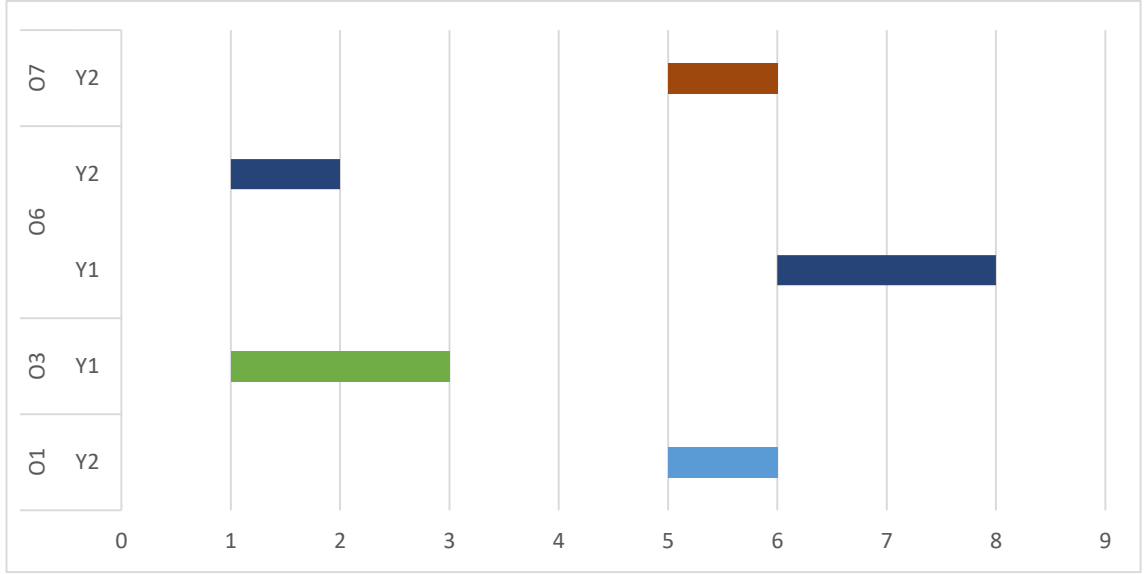
Şekil E7.2 İki Takım Olması Durumunda Çizelge



Şekil E7.3 Üç Takım Olması Durumunda Çizelge

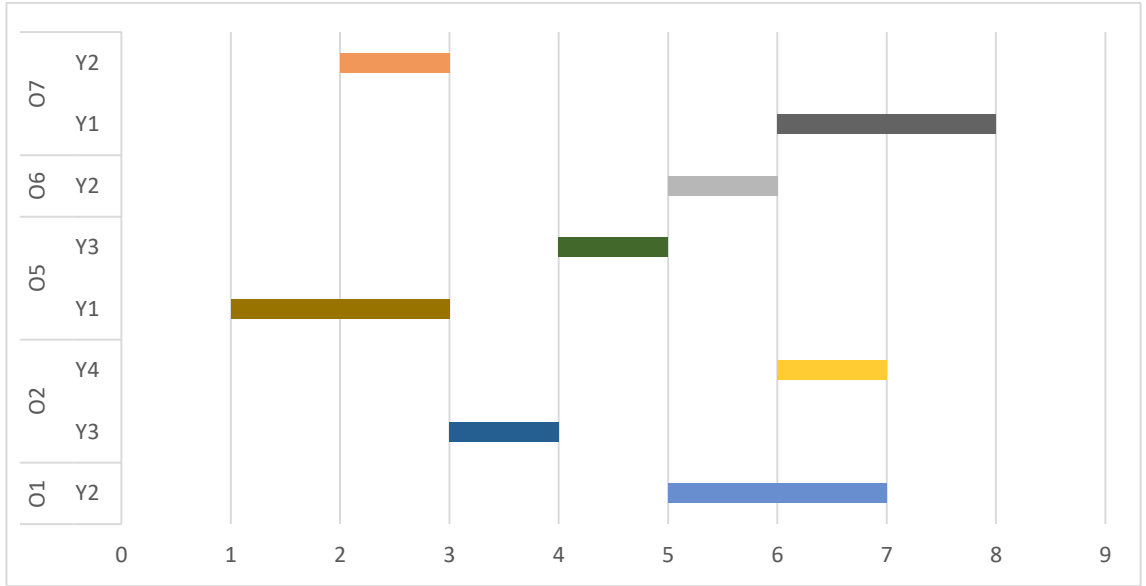


Şekil E7.4 Dört Takım Olması Durumunda Çizelge

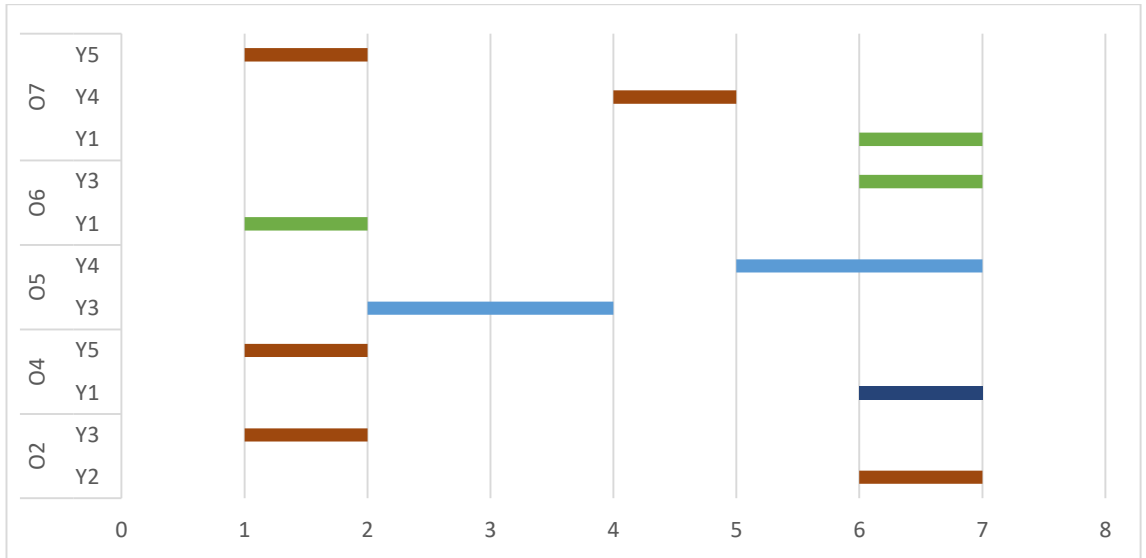


Şekil E7.5 Beş Takım Olması Durumunda Çizelge

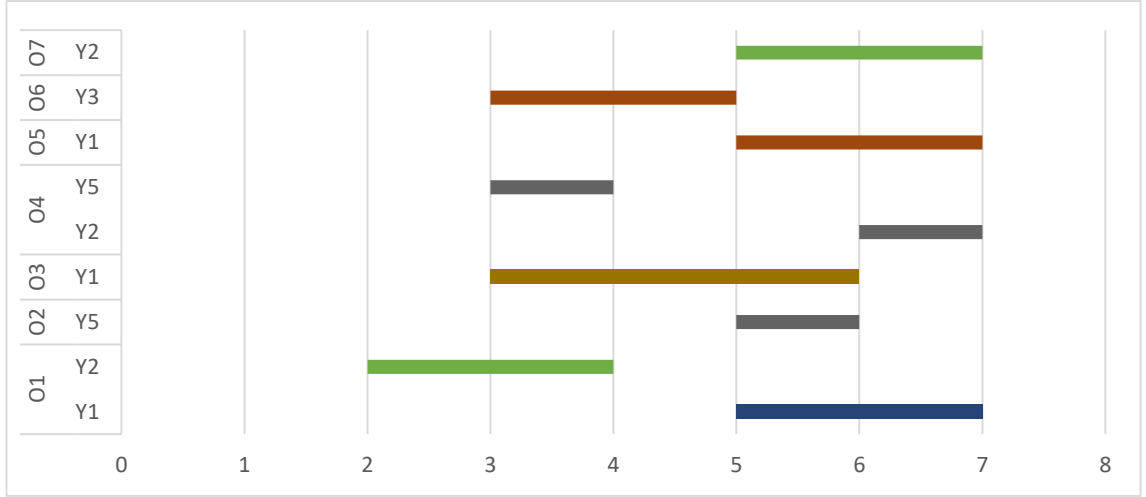
EK 8. Farklı Günler için Gantt Şemaları



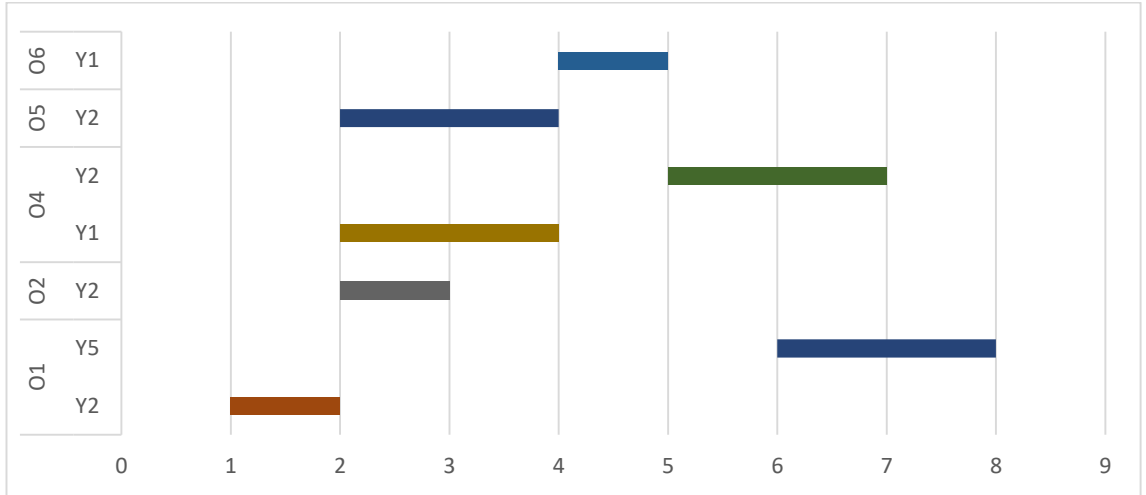
Şekil E8.1 Birinci Gün Çizelge



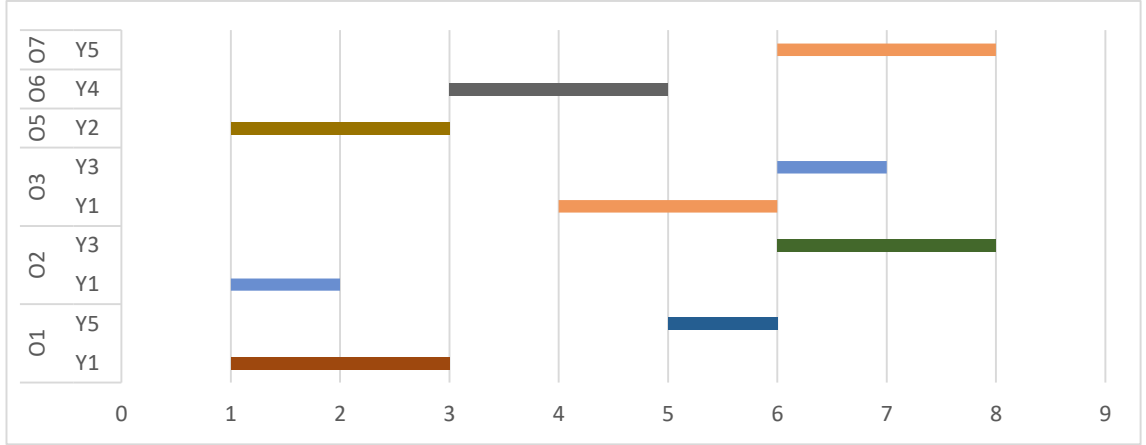
Şekil E8.2 İkinci Gün Çizelge



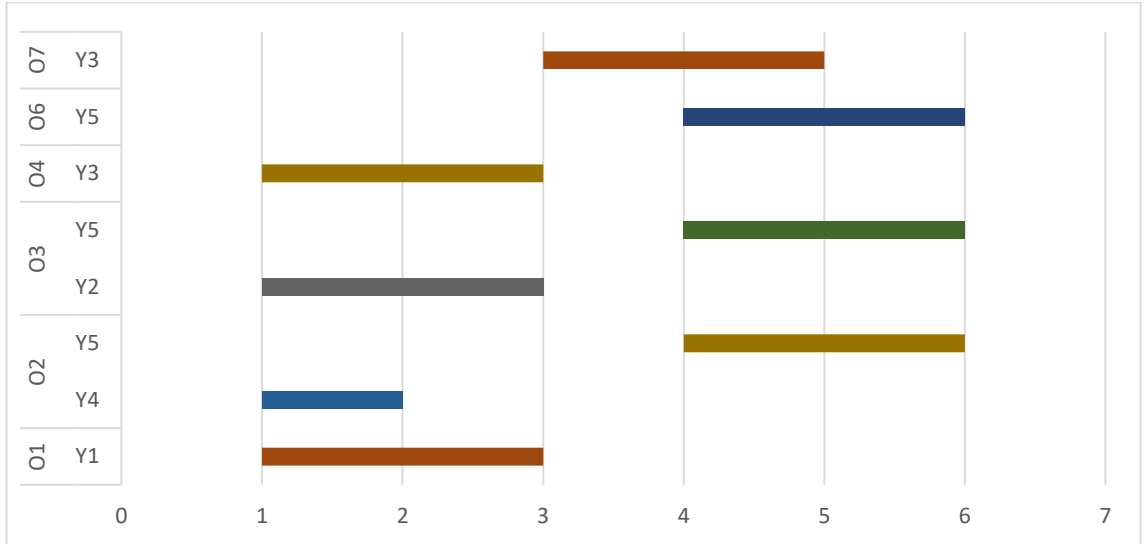
Şekil E8.3 Üçüncü Gün Çizelge



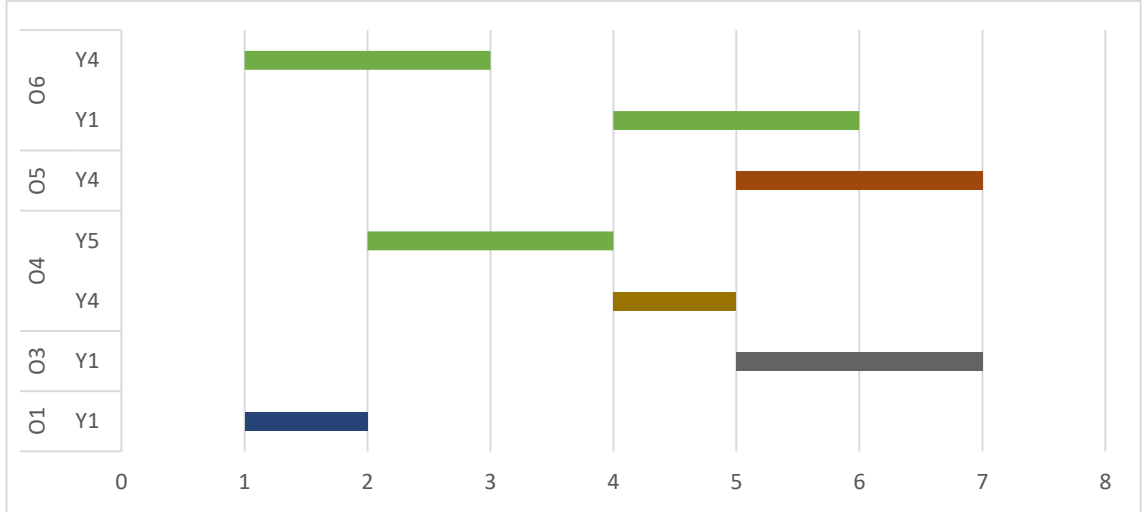
Şekil E8.4 Dördüncü Gün Çizelge



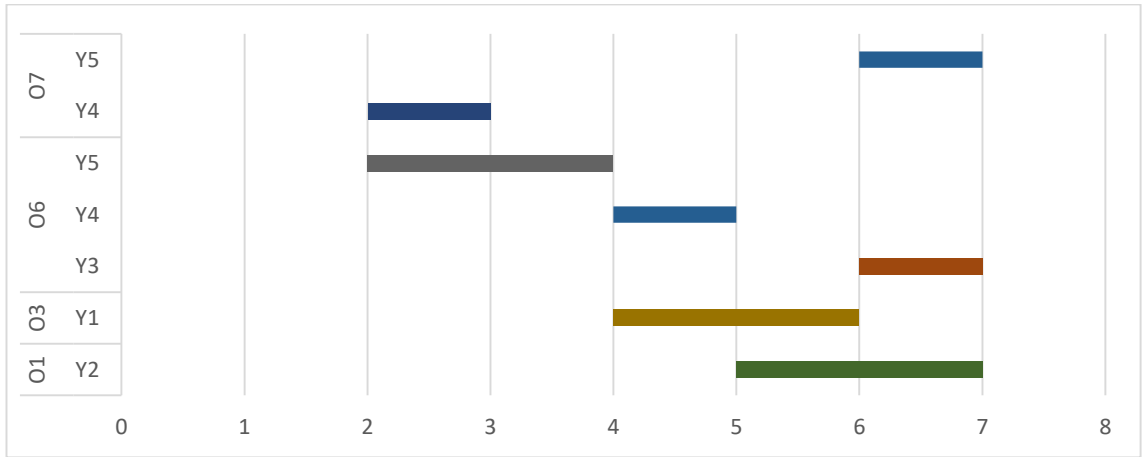
Şekil E8.5 Beşinci Gün Çizelge



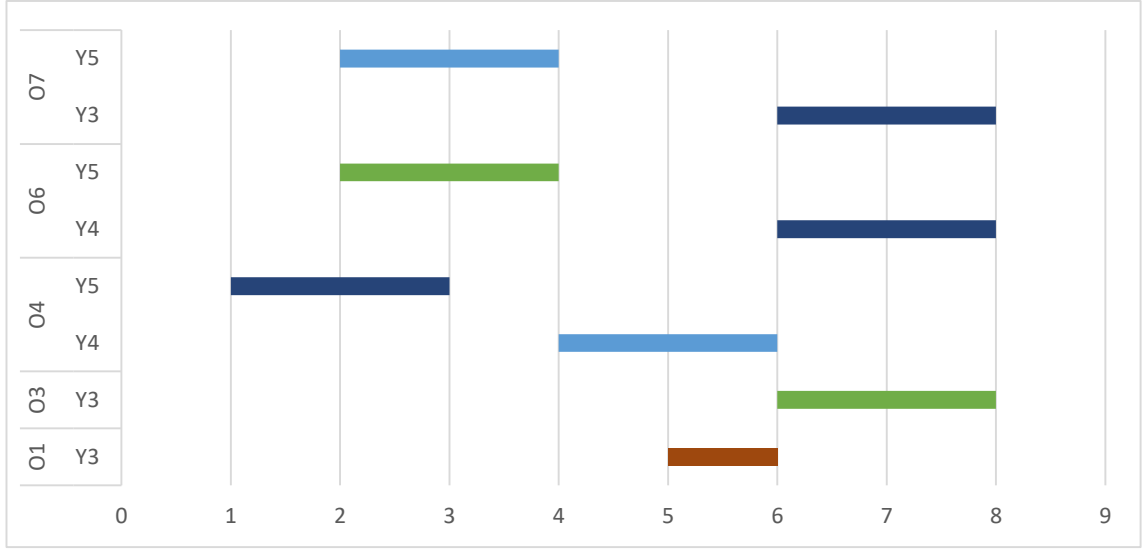
Şekil E8.6 Altıncı Gün Çizelge



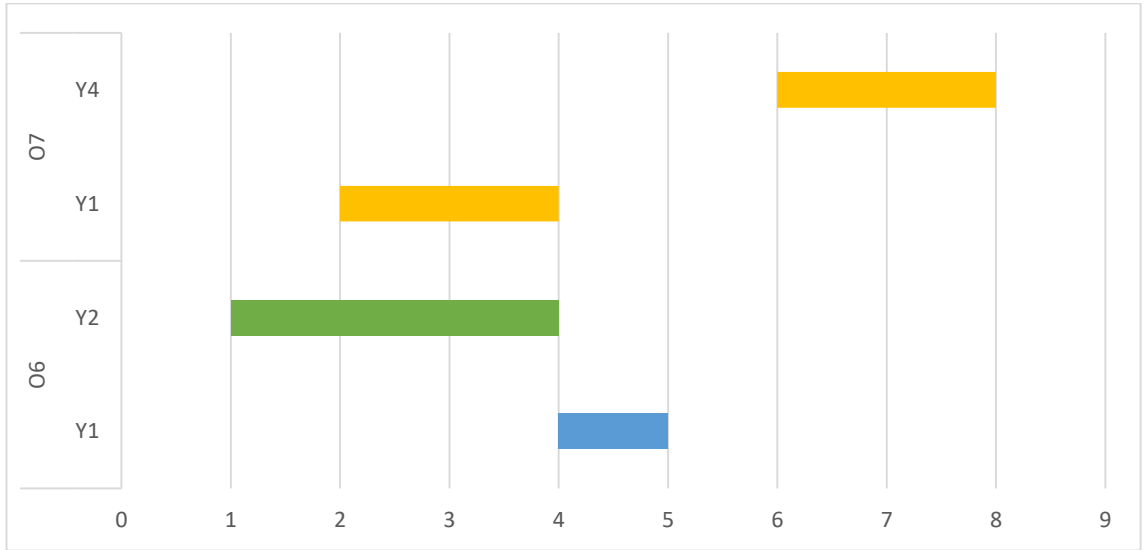
Şekil E8.7 Yedinci Gün Çizelge



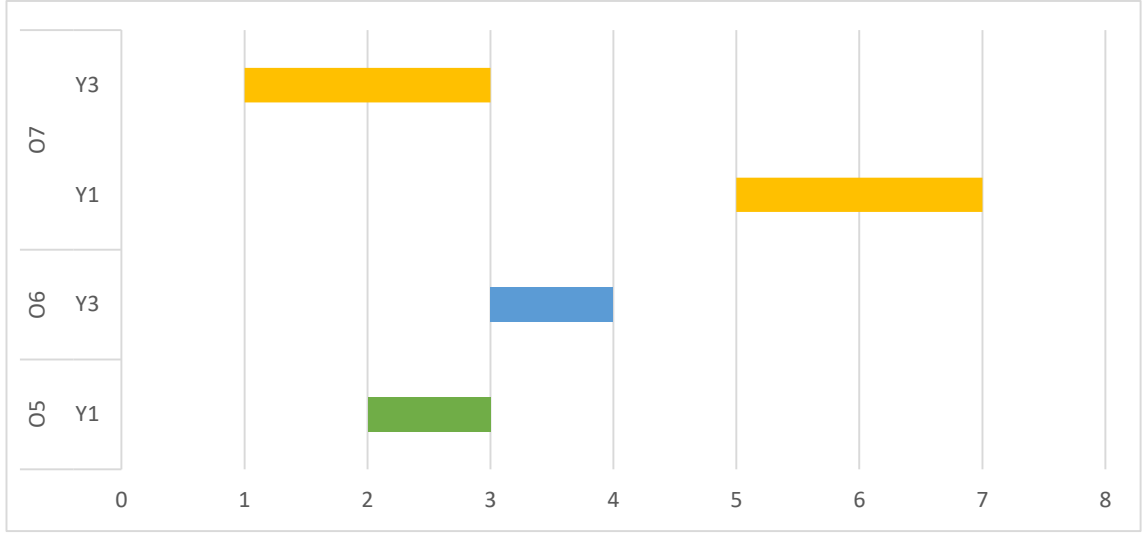
Şekil E8.8 Sekizinci Gün Çizelge



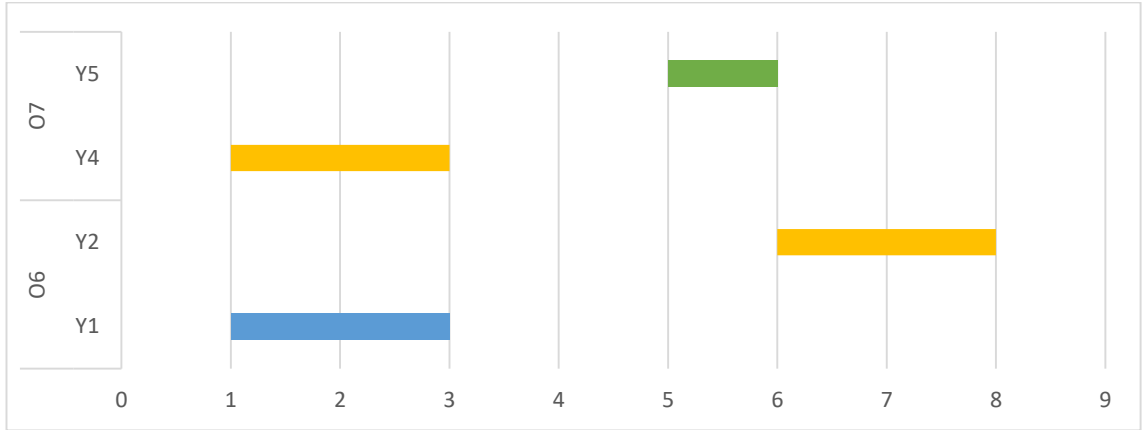
Şekil E8.9 Dokuzuncu Gün Çizelge



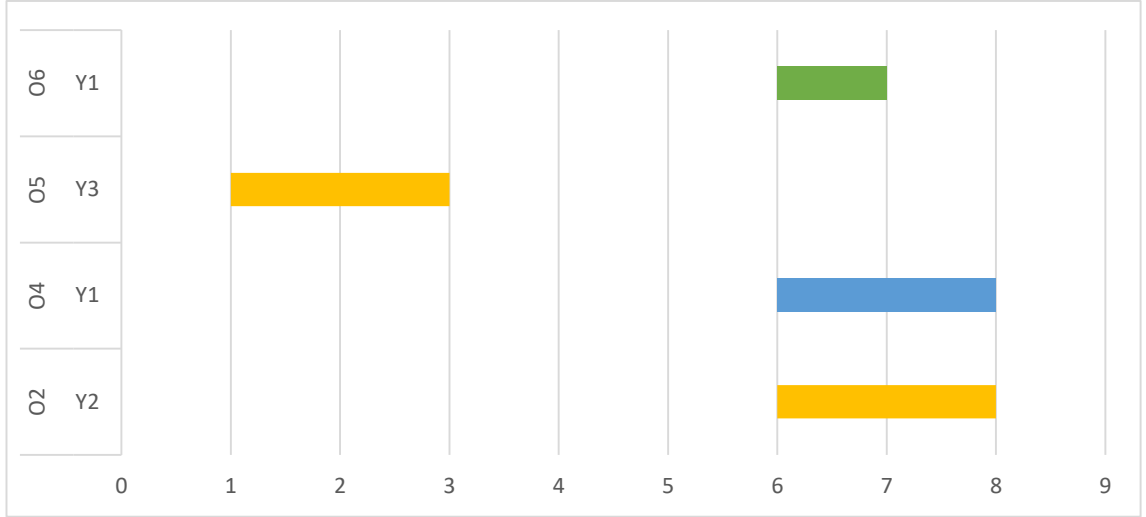
Şekil E8.10 Onuncu Gün Çizelge



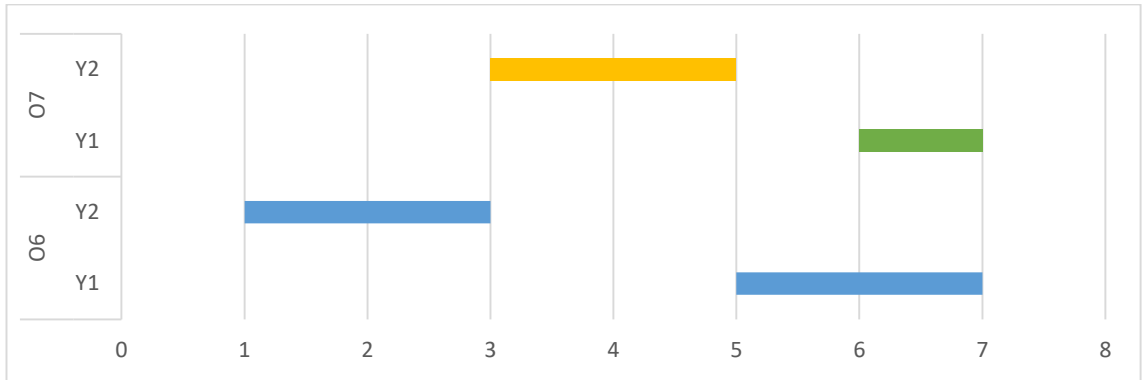
Şekil E8.11 Onbirinci Gün Çizelge



Şekil E8.12 Onikinci Gün Çizelge

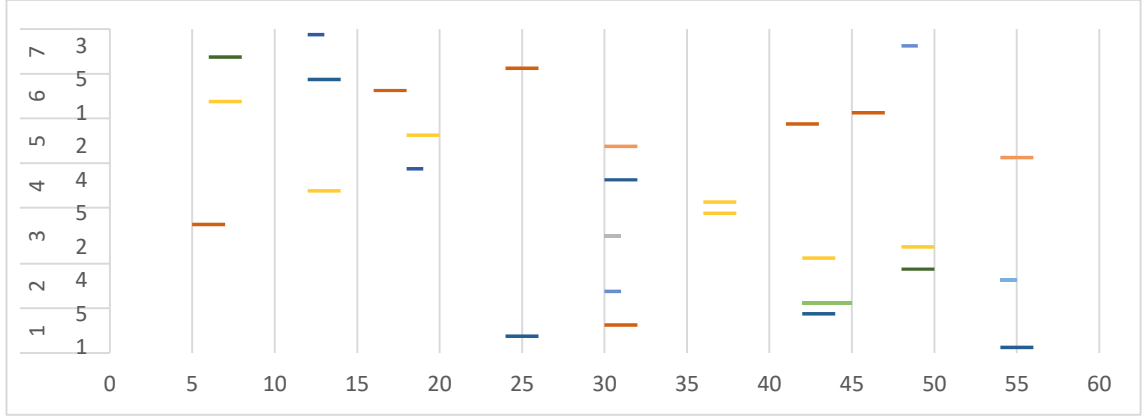


Şekil E8.13 Onüçüncü Gün Çizelge



Şekil E8.14 Ondördüncü Gün Çizelge

EK 9. Geniş Planlama Ufku için Gantt Şeması



Şekil E9.1 Geniş Planlama Ufku için Çizelge