TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SIRA BAĞIMLI AYAR ZAMANI VE FAZLA MESAİ İLE MAKİNE ÇİZELGELEME

YÜKSEK LİSANS TEZİ

Zeynep BÜLBÜL

Endüstri Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Hakan GÜLTEKİN

AĞUSTOS 2017

Fen Bilimleri Enstitüsü Onayı

………………………..
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksininlerini sağladığını onaylarım.

……………………….
Prof. Dr. Tahir HANALİOĞLU
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 151311020 numaralı Yüksek Lisans Öğrencisi **Zeynep BÜLBÜL** 'ün ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"SIRA BAĞIMLI AYAR ZAMANI VE FAZLA MESAİ İLE MAKİNE ÇİZELGELEME"** başlıklı tezi **15.08.2017** tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı : **Doç. Dr. Hakan GÜLTEKİN** …........................
TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Gültekin KUYZU(Başkan)** ....
TOBB Ekonomi ve Teknoloji Üniversitesi

Yrd. Doç. Dr. Selçuk GÖREN ....
Abdullah Gül Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

.

Zeynep BÜLBÜL

ÖZET

Yüksek Lisans

SIRA BAĞIMLI AYAR ZAMANI VE FAZLA MESAİ İLE MAKİNE ÇİZELGELEME

Zeynep BÜLBÜL

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hakan GÜLTEKİN

Tarih: Ağustos 2017

Bu çalışma kapsamında sıra bağımlı ayar zamanı ile tek makine çizelgeleme problemi ele alınmaktadır. Bir gerçek hayat uygulamasını temel alan problemin amacı, en küçük toplam fazla mesai değeri ile müşteri taleplerinin teslim tarihinden önce hazır olmasını sağlayan haftalık üretim çizelgesini belirlemektir. Fazla mesai ücretleri fazla mesai yapılan süreye bağlı olarak hesaplanmaktadır. Bu problem için günlük detayda çizelgeler oluşturulmaktadır. Problem için öncelikle karma tam sayılı matematiksel programlama modeli geliştirilmiştir. Bu optimizasyon problemi için yapılan testlerde gerçek hayat örnekleri için makul sürelerde çözüm bulunamadığı görüldüğünden çözüm zamanını azaltmak amacıyla bir ayrıştırma algoritması geliştirilmiştir. Mantıksal Benders ayrıştırma algoritması olarak adlandırılan bu yaklaşımla karmaşık esas problem, karma tam sayılı ana problem ve çeşitli sayıda kısıt programlama alt problemleri olarak, görece kolay çözülebilir problemlere ayrıştırılmaktadır. Ana problemde işlerin günlere atanmasına karar verilirken, alt problemlerde bu atamalara göre günlük çizelgeler belirlenmektedir. Algoritma alt problemlerin çözümüne göre tanımlanan olurluluk veya optimallik kesilerinin ana probleme eklenmesi ile devam etmektedir. Çalışma kapsamında algoritma, kesilerin eklenme şekline göre iteratif ve dallandırma ve kontrol olmak üzere iki farklı yaklaşımla uygulanmıştır. Ayrıştırma algoritması ile, makul süre içerisinde optimal çözümü belirlemenin zorlaştığı örnekler için, ayrıca bir tavlama benzetimi algoritması geliştirilmiştir. Tavlama benzetimi için başlangıç çözümü, geliştirilen çözüm kuran algoritmayla, soğuma şeması ise gerçekleştirilen parametre analizleriyle belirlenmiştir. Geliştirilen bütün çözüm yöntemlerinin performansı yapılan deneysel çalışmalarla analiz edilmiştir.

Anahtar Kelimeler: Makine çizelgeme, Sıra bağımlı ayar zamanı, Fazla mesai, Benders ayrıştırma algoritması, Kısıt programlama, Tavlama benzetimi

ABSTRACT

Master of Science

MACHINE SCHEDULING WITH SEQUENCE DEPENDENT SETUP TIME AND OVER TIME

Zeynep BÜLBÜL

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Industrial Engineering Science Programme

Supervisor: Assoc. Prof. Hakan GÜLTEKİN

Date: Augusut 2017

In this study we considered a single machine scheduling problem with sequence dependent setup times from a real-world application. Our goal is to determine the weekly production schedule that minimizes the overtime with respect to deadlines. Here the available overtime periods are defined at the end of each day and related costs are calculated based on overtime use. Therefore the amount of overtime used at each day is determined through detailed schedules. We developed a mixed integer programming model for the problem. However our experimental studies showed that this optimization problem is intractable for real-world instances. A decomposition algorithm is proposed in order to decrease solution time. We developed a logic-based Benders decomposition algorithm which decomposes the original problem to a daily assignment problem and scheduling problems based on these assignments for each day. In this approach the master assignment problem is solved with mixed integer programming while sub scheduling problems are solved with constraint programming. The algorithm proceeds as the feasibility and optimality cuts, defined by the solutions obtained from subproblems, are added to the master problem. In the scope of this study related cuts are added either iteratively or with branch-and-check approach. For the instances that the decomposition algorithm could not find the optimal solution in a reasonable time, simulated annealing algorithm is used to obtain feasible solutions. In this context initial solution is determined through the constructive heuristic developed while cooling schedule is defined through parameter analyses. The performances of these solution approaches are tested through experimental studies.

Keywords: Machine scheduling, Sequence dependent setup time, Overtime, Benders decomposition algorithm, Constraint programming, Simulated annealing algorithm

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Hakan GÜLTEKİN'e, kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Endüstri Mühendisliği Bölümü öğretim üyelerine ve destekleriyle her zaman yanımda olan aileme ve arkadaşlarıma çok teşekkür ederim. Son olarak yüksek lisans eğitimimde sağladığı burstan dolayı TOBB Ekonomi ve Teknoloji Üniversitesine teşekkürlerimi sunarım.

İÇİNDEKİLER

ŞEKİL LİSTESİ

<u>Sayfa</u>

ÇİZELGE LİSTESİ

Sayfa

1. GİRİŞ

Rekabet arttıkça, firmaların karlılığında müşteri memnuniyetinin önemi artmaktadır. Bu anlamda firmalar üretim veya servislerini müşteri taleplerini zamanında karşılayabilecek şekilde çizelgelemeyi amaçlamaktadırlar. Bu doğrultuda literatürde, en küçük geç kalma değerine sahip çizelgeyi belirleme gibi, tamamlanma zamanına yönelik amaçlarla tanımlanan problemler değerlendirilmektedir. Fakat günümüz rekabet koşulları, artık en küçük geç kalmanın belirlenmesinin ötesinde müşteri taleplerini doğrudan zamanında karşılayacak çizelgelerin belirlenmesini gerektirmektedir. Bu yüzden firmalar, müşteri memnuniyetsizliğinin neden olabileceği kayıplara katlanmamak adına maliyetlerini arttırmak durumunda kalabilmektedirler. Burada sıklıkla karşılaşılan bir uygulama, fazla mesaidir. Firmalar, müşteri taleplerini mesai süresi içerisinde karşılayacak bir çizelge belirlenememesi durumunda, sabit mesai maliyetinin dışında bir fazla mesai maliyetine katlanarak talepleri hazırlamaya çalışmaktadırlar. Fakat müşteri memnuniyetinin sağlanmasında en az taleplerinin zamanında karşılanması kadar, düşük fiyatlarla sunulması da önemlidir. Bu anlamda fiyatları arttırmadan karlılığı koruyabilmek adına, çizelgenin müşteri taleplerinin karşılanmasını sağlayacak en küçük fazla mesai değeriyle belirlenmesi gerekmektedir. Bu çalışma kapsamında değerlendirilen problem bu doğrultuda tanımlanan bir gerçek hayat problemini temel almaktadır. Problem tanımına göre firma, haftalık belirlenen üretim planı için, müşteri siparişlerini en küçük toplam fazla mesai maliyetiyle teslim tarihlerine hazırlayacak üretim çizelgesinin belirlenmesini amaçlamaktadır. Burada fazla mesai, günlük olarak normal mesainin ardına tanımlanmaktadır. Bu doğrultuda üretim çizelgesi günlük detayda oluşturulmaktadır. Bu çalışma kapsamında temel alınan uygulamada öncelik talebin zamanında karşılanması, ardından bu doğrultuda katlanılan maliyetlerin azaltılmasıdır. Olurlu bir çizelge için bütün siparişlerin teslim tarihlerine kadar yetişmesi gerekmektedir. Bu anlamda bir ceza maliyeti ödünleşimi ile siparişlerin üretiminin teslim tarihinden geç tamamlanmasına izin verilmemektedir. Bu uygulamada firma, çalışanlarına normal mesai ücretini sabit olarak öderken, fazla mesai yapılan süre için normal mesai ücretinin 1,5 katı ücret ödemektedir. Bu doğrultuda çalışma kapsamında normal mesai maliyeti değerlendirilmeksizin, toplam fazla mesai maliyeti en küçüklenmektedir. Burada fazla mesai maliyeti, fazla mesai yapılan işten bağımsız olarak doğrudan fazla mesai yapılan süreye bağlı hesaplanmaktadır.

Çalışma kapsamında üretim tek makineyle çizelgelenmektedir. Burada siparişlerin toplam üretim zamanı sabit ve bilinmektedir. Bu noktada problem tanımı adına önemli bir varsayım, siparişlerin üretimleri arasındaki geçişte sıra bağımlı ayar zamanlarına katlanılmasıdır. Çizelgeleme aşamasında sıra bağımlı ayar zamanlarının değerlendirilmemesi, planlananın üzerinde fazla mesai yapılmasına hatta firmanın olursuz bir çizelgeyi uygulanmasına neden olabilmektedir. Firma haftalık üretim planlarını, ana üretim planlama aşamasında belirlenen aylık planlar doğrultusunda oluşturmaktadır. Bu doğrultuda siparişlerin üretimi için gerekli malzemeler, çizelgeleme periyodunun başına hazır olacak şekilde planlama yapılmaktadır. Bu anlamda siparişlerin üretime hazır olma tarihleri hafta başı olarak tanımlanmaktadır. Siparişlerin teslim tarihi ise gün sonu olarak belirtilmektedir.

Toplam geç kalma zamanını en küçüklemek amacı için ilgili problemin NP-Zor olduğu bilinmektedir. Benzer doğrultuda tanımlanan mevcut çalışmada da problemin karmaşıklığının fazla mesai kararlarını içermesi nedeniyle, artacağı bilinmektedir. Bu sebeple problemin çözümü için literatürde çizelgeleme gibi birçok karmaşık optimizasyon probleminde kullanılan ayrıştırma yöntemi önerilmiştir. Bu yöntem büyük boyutlu bir problemi çözmek yerine, daha küçük boyutlu birden çok problemi çözmenin görece kolay olacağı temeline dayanmaktadır. Bu çalışmada literatürde mantıksal Benders ayrıştırma algoritması olarak adlandırılan ayrıştırma yaklaşımı ile haftalık çizelgeleme problemi, gün içerisinde üretilecek ürünlerin belirlenmesi ve buna göre günlük çizelgelerin oluşturulması şeklinde ayrıştırılacaktır. Bu yaklaşımda öncelikle haftalık işlerin üretimi matematiksel programlama ile günlere atanacak, ardından bu atamalar için kısıt programlama ile günlük çizelgeler oluşturacaktır.

Ayrıştırma yönteminin makul süreler içerisinde optimal çözümü veremediği örnekler için, kısa sürede kaliteli çözümlere üretecek sezgisel yöntemler geliştirilecektir. Bu amaçla tasarlanan tavlama benzetimi algoritması optimal çözümü garanti etmemektedir, fakat ayrıştırma algoritması ile olurlu çözüm belirlemenin zorlaştığı örneklerde kaliteli çözümler türetebilmek adına etkin bir yaklaşım olduğu gözlemlenmiştir.

Altı bölümden oluşan bu tez çalışmasının bir sonraki bölümünde literatür taraması yapılmıştır. Bölüm 3'de problem detaylıca tanımlanarak karma tam sayılı matematiksel programlama formülasyonu verilecektir. Problem tanımının ardından Bölüm 4'te problem için geliştirilen çözüm yöntemleri sunulacaktır. Bölüm 5'te geliştirilen çözüm yöntemlerinin performanslarını test etmek için yapılan sayısal çalışmaya yer verilmektedir. Son olarak Bölüm 6 çalışmanın sonuçlarını ve gelecek çalışma önerilerini içermektedir.

2. LİTERATÜR

Literatürde 1950 yıllarından itibaren çizelgeleme problemlerine yönelik pek çok çalışma yapılmıştır. Bu alanda en çok çalışılan problemlerden biri olan tek makine çizelgeleme problemi için günümüze kadar çok çeşitli varsayımlar incelenmiştir. Bu anlamda literatürde çizelgeleme problemlerinin önemi arttıkça, gerçek hayat problemlerini daha iyi ifade edebilen problemler tanımlanmaya başlanmıştır.

Bu bölüm içerisinde, bu çalışma kapsamında tanımlanan gerçek hayat probleminin varsayımları için literatürde yapılan çalışmalar incelenecektir. Burada temel olarak değerlendirilen varsayımlar, sıra bağımlı ayar zamanı ve fazla mesaidir. Bunun dışında problemin çözümü için önerilen yönteme yönelik literatürde yapılan çalışmalar da ele alınacaktır.

2.1 Ayar Zamanı

Ayar, bir üretim veya servis kaynağında bir işlemi gerçekleştirebilmek adına yapılması gereken hazırlık, ayar zamanı ise bu hazırlık için harcanan zaman olarak tanımlanmaktadır [1]. Bu anlamda bir üretim sisteminde ayar yapılması gereken süreçlere örnek olarak üretimde kullanılacak aletlerin hazırlanması, bir servis sisteminde ise servis sunulacak ortamın hazırlanması süreçleri verilebilir.

Ayar, özellikle bir kaynak üzerine birden fazla işlemin atandığı çizelgelenme problemlerinde karşılaşılan bir durumdur [2]. Buna karşılık çizelgeleme literatürünün başlangıcında yapılan çalışmalarda, çözüm kolaylığı sağlaması adına ayar zamanları ihmal edilmiş veya işlem zamanlarına dâhil edilmiştir.

Fakat çizelgeleme sürecinde ayar zamanlarının dikkate alınmasıyla elde edilebilecek kazanımların fark edilmesiyle, literatürde ayar zamanlarını değerlendiren çalışmaların sayısı önemli bir şekilde artmıştır [3].

Wortman [4] çalışmasında kapasitenin etkili kullanımı adına ayar zamanlarının göz önünde bulundurulmasının gerekliliğinden bahsetmiştir. Wilbrecht ve Presscot'un [5] yaptıkları çalışmanın sonucu da tam kapasite çalışan bir atölye tipi üretim ortamında, ayar zamanlarının değerlendirilmesinin önemini vurgulamıştır. Krajewski vd. [6] ise çalışmasında parti büyüklüğü ile birlikte ayar zamanlarını azaltmanın, maliyetleri azaltmak ve müşteri memnuniyetini arttırmak için en etkili yöntem olduğunu ifade etmiştir. Trovinger ve Bohn [7] çalışmasında elektronik devre kartı üretiminde etkin kapasitenin yaklaşık yarısının ayar zamanları için harcandığını ve ayar zamanlarının azaltılmasıyla yıllık yaklaşık 1,8 milyon dolar kazanç sağlanabileceğini belirtmiştir. Benzer şekilde Loveland vd. [8] Dell için yaptıkları çalışmada, ayar zamanlarını azaltarak üretim kapasitesini %35 artırıp 1 milyon doların üzerinde bir yıllık kazanç sağlamışlardır. Allahverdi ve Soroush [2] ayar içeren yaklaşık 50 çizelgeleme uygulamasını tarayarak, ayar zamanlarının değerlendirilmesinin katkılarından bahsetmiştirler.

Literatürde ayar zamanları temel olarak, sıra bağımlı ve bağımsız olarak iki sınıf altında değerlendirilmektedir. Bir iş için harcanan ayar zamanı, öncesinde tamamlanan işe bağlı olarak değişiyor ise sıra bağımlı olarak tanımlanmaktadır. Sıra bağımlı ayar zamanı içeren süreçlere örnek, tekstil endüstrisinde kumaş boyama sürecinden verilebilir. Bu aşamada siyah boyanın ardından beyaz kullanılacaksa, boyama yapılan alanın tamamen temizlenmesi gerekmektedir. Buna kıyasla, beyaz boyanın ardından siyah kullanılması durumunda daha yüzeysel yani görece daha kısa bir temizlik yeterli olmaktadır [9]. Ayar zamanı sadece işin kendisi ile ilişkiliyse sıra bağımsız olarak adlandırılmaktadır.

Çizelgeleme literatürüne dair yapılan tarama çalışmalarında ayar zamanı ilk defa Gupta ve Kypraisis'in 1987 yılında yaptıkları çalışmada incelenmiştir [1]. Tek makine çizelgeleme literatüründen 171 çalışmanın incelendiği bu taramada, ayar zamanı içeren 13 çalışmaya yer verilmiştir. Doğrudan ayar zamanlarını içeren çalışmalar için hazırlanan ilk kapsamlı tarama ise Allahverdi vd. [1]'in hazırladığı taramadır. İlgili çalışma ve devamında Allahverdi vd. [3] ve Allahverdi [9]'nin yaptıkları taramalar, 1960 yıllarının ortasından 2014 yılının sonuna kadar ayar zamanı ile makine çizelgeleme problemleriyle ilgili yaklaşık 1000 çalışmayı kapsamaktadır. Bu taramalarda ele alınan çalışmalar üretim ortamlarına ve ayar zamanlarına göre sınıflandırılmıştır. Yapılan sınıflandırmalarda ayar zamanlarının sıra bağımlı ve bağımsız olmasının yanı sıra, ürün aile yapısı da incelenmiştir. Ürün aile yapısının bulunduğu problemlerde, benzer ürün özelliklerine sahip ürünler bir ürün ailesini ifade etmektedir ve aynı ailenin elemanı ürünler arasındaki üretim geçişinde ayar yapılmadığı varsayılmaktadır. Ürün aileleri arasındaki geçişte katlanılan ayar zamanları ise ayrıca sıra bağımlı veya bağımsız olarak ayrılmaktadır.

Literatürde ayar zamanı ile makine çizelgeleme problemlerinde amaç genellikle üretim tamamlanma zamanları üzerinden tanımlanmaktadır. Çizelgelenen en son işin tamamlanma zamanı, literatürde *yayılma zamanı* olarak tanımlanmaktadır. Pinedo [10] çalışmasında tek makinede sıra bağımlı ayar zamanı ile çizelgeleme probleminin, yayılma zamanını en küçüklemek amacı için NP-Zor olduğunu göstermiştir. Bu sebeple, ilgili problem için polinom zamanda çözüm üretilemeyeceği bilinmektedir. Toplam geç kalma zamanını en küçüklemek amacı için tanımlanan problem, polinom zamanda ilgili probleme indirgenebileceği için bu problemin de NP-Zor olduğu sonucuna varılmaktadır [11]. Bu nedenle endüstride karşılaşılan büyük boyutlu problemlere çözüm sunabilmek adına, literatürde sezgisel çözüm yöntemleri tercih edilmektedir. Bu çalışma kapsamında ele alınan problemde amaç, literatürde değerlendirilen tamamlanma zamanına yönelik amaçlardan farklı olarak, üretim kapasitesi üzerinden tanımlanmaktadır. Üretimin teslim tarihine yetişmesini sağlamak adına gereken en küçük fazla mesai değerinin belirlenmesi istenmektedir. Fazla mesai üretimin geç kalmasını engellemek adına kullanıldığı için, bu bölüm içerisinde bu amaca benzer olan, toplam geç kalma zamanını en küçüklemek amacıyla yapılan çalışmalar incelenmiştir.

Ragatz [12] ilgili amaç için bir dal ve sınır algoritması önermiştir. Bu algoritma ile küçük boyutlu örneklerde optimal sonuç elde edilebilirken, gerçek hayatta karşılaşılan büyük boyutlu örnekler için makul süre içerisinde çözüm elde edilememektedir. Rubin ve Ragatz [13] ilgili problem için sezgisel çözüm yöntemleri önermişlerdir. Tasarladıkları genetik ve rasgele arama algoritmalarından elde ettikleri sonuçları, Ragatz [12] tarafından tasarlanan dal ve sınır algoritması ile karşılaştırmıştırlar ve dal ve sınır algoritması ile optimal çözümün elde edilemediği örneklerde, önerilen çözüm yöntemlerinin kısa süre içerisinde oldukça iyi çözümler üretebildiğini göstermiştirler. Burada genetik ve rasgele arama algoritmaları arasında yapılan karşılaştırma sonucunda ise, rasgele arama algoritmasının çözüm performansının daha iyi olduğu sonucuna varılmıştır. Tan ve Narishman [14], bu rasgele arama algoritmasından elde edilen sonuçları geliştirdikleri tavlama benzetimi algoritmasının sonuçlarıyla karşılaştırmıştır. Bu çalışmanın sonucuna göre rasgele arama algoritması daha kısa sürede çözüm vermesine rağmen, tavlama benzetimi ile daha uzun sürede daha iyi çözümler elde edilebilmektedir. Ragatz [12], Rubin ve Ragatz [13] ve Tan ve Narishman [14] çalışmalarının ardından Tan vd. [15] bu çalışmaları bir araya getirerek önerilen yöntemleri karşılaştırmıştır. Buna göre tavlama benzetimi ve rasgele arama algoritmalarının büyük boyutlu örneklerde iyi sonuçlar üretmek, dal ve sınır algoritmasının ise küçük boyutlu örneklerde optimal çözümü elde etmek adına en uygun yöntemler olduğu sonucuna varılmıştır.

2.2 Fazla Mesai

Makine çizelgeleme problemleri, özellikle tek makine çizelgeleme problemi ile ilgili literatürde geniş bir içerik bulunmaktadır. Bu çalışmalarda genellikle üretim kapasitesinin sabit olduğu varsayılmaktadır. Fakat bu varsayımın aksine günümüz rekabet koşullarında, endüstride karşılaşılan çizelgeleme problemlerinde, müşteri memnuniyetini ve neticesinde karlılığı sağlayabilmek adına firmalar kapasitelerini talebe göre ayarlamaktadırlar.

Talebin üretim kapasitesini aşması durumunda, üretimin bir kısmının taşeron firmalara verilmesi ve/veya sabit mesai süresinin üzerine fazla mesai yapılması gibi uygulamalarla üretimin gecikmesi önlenebilmektedir. Taşeron firmalarla anlaşma firmanın karlılığını azaltacağı için, genellikle bu uygulama öncelikli olarak tercih edilmemektedir [16]. Bu çalışma kapsamında da problem tanımda temel alınan uygulamada firma, kapasitesini fazla mesai ile ayarlamaktadır.

Literatürde fazla mesai kararını üretim planlama aşamasında değerlendiren çalışmalar yer almaktadır. Zobolas vd. [16], ana üretim planını (MPS), üretimin teslim tarihinden erken veya geç tamamlanması durumunda katlanılan ceza maliyetleri ile fazla mesai maliyetinin toplamını azaltacak şekilde güncelleyen bir karar destek sistemi geliştirmiştir. Çözüm yöntemi olarak bir metasezgisel algoritma önermişlerdir.

Akkan [17] ise yapılacak fazla mesainin üretim planlama aşamasında yaklaşık bir şekilde belirlenmesinin, olursuz üretim çizelgelerine neden olabileceğini belirtmektedir. Bu sebeple, fazla mesainin detaylı üretim çizelgeleri üzerinden belirlenmesini önermiştir. Fazla mesai çizelgeleme problemi olarak tanımladığı çalışma kapsamında Akkan [17], bir gerçek zamanlı üretim çizelgeleme problemini ele almıştır. Bu çalışmada, yeni üretim emirlerinin mevcut üretim çizelgesine en küçük fazla mesai maliyetiyle eklenmesi için tasarlanan farklı sezgisel algoritmalar kullanılmıştır.

Yang vd. [18] ve Jaramillo vd. [19] sırasıyla proje ve uçak bakım çizelgeleme problemlerinde fazla mesai uygulamasını ele almışlardır. İşin geç kalması durumunda katlanılan ceza maliyeti ve fazla mesai maliyeti toplamının en küçüklenmesinin amaçlandığı problemler için, karma tam sayılı matematiksel modeller geliştirilmiştir. Bu çalışmalarda geç kalma ceza maliyetleri iş önceliklerine göre ağırlıklandırılmıştır. Yang vd. [18] bir iş tamamlanmadan başka bir işe başlanmasına izin verilmeyeceğini varsaymıştır. Bu durum literatürde iş kesmesiz çizelgeleme olarak tanımlanmaktadır. Jaramillo vd. [19] ise bu varsayımdan farklı olarak, iş kesmeye izin vermiştir. İki çalışmada da çözüm yöntemi olarak sezgisel algoritmalar önerilmiştir.

Düşük miktarlarda, yüksek çeşitlilikle üretim yapılan üretim ortamlarına "*atölye tipi üretim*" denir. Yoda vd. [20] atölye tipi üretim ortamları için fazla mesaiyle çizelgeleme problemini ele almıştır. Problemin amacı en küçük fazla mesai ile ürünlerin teslim tarihine yetişmesini sağlayacak üretim çizelgesini belirlemektir. Problem kapsamında üretimi gün içerisinde tamamlanamayacak bir ürünün ilgili güne çizelgelenmesine izin verilmemektedir. Bu doğrultuda ürünlerin işlem zamanlarının normal mesai süresine eşit veya ilgili süreden az olduğu varsayılmıştır. Çözüm yöntemi olarak genetik algoritma geliştirilmiştir. Algoritmayla ilk olarak fazla mesai amacından bağımsız olarak bütün ürünlerin teslim tarihine yetişeceği bir çizelge üretilmiş, ardından bu çizelgeye ait sıralama için düşük fazla mesai değerine sahip bir çizelge belirlenmeye çalışılmıştır.

Literatürde fazla mesai ile ilgili yapılan bu çalışmalarda problem ya üretim planlama problemi olarak değerlendirilmiştir ya da detaylı çizelgeleme problemleri olarak tanımlanan problemlerde sıra bağımlı ayar zamanları ele alınmamıştır. Bu çalışma kapsamında ise, belirli bir haftalık üretim planı için sıra bağımlı ayar zamanlarıyla makine çizelgeleme problemi incelenecektir. Bu anlamda literatürde yer alan çalışmalardan farklıdır.

Freeman vd. [21]'nin ele aldığı çizelgeleme problemi, sıra bağımlı ayar zamanları ve fazla mesainin birlikte değerlendirildiği ilk problemdir. Alakasız paralel makinelerin çizelgelendiği problemde amaç, fazla mesai maliyeti ile birlikte kusurlu üretimler neticesinde katlanılan atık maliyetini en küçüklemektir. Problemde ürünlerin üretime başlayabileceği en erken zaman veya üretiminin tamamlanabileceği en geç zaman belirtilmemektedir. Fazla mesai, üretimlerin çizelgeleme periyodu içerisinde tamamlanmasını sağlamak için tanımlanmıştır. Freeman vd. [21], mevcut çalışmadan farklı olarak, doğrudan çizelgeleme periyodu boyunca yapılması gereken toplam fazla mesai değerini belirlemiştir. Bu çalışma kapsamında ise fazla mesai, ürünlerin gün detayında belirlenen teslim tarihlerine yetişmesi için günlük olarak belirlenmek durumundadır. Bu anlamda Freeman vd. [21] çalışmasında ele alınan problem paralel makineleri içermesi nedeniyle mevcut çalışmadan daha karmaşık olarak değerlendirilse de, ilgili çalışmada detaylı bir çizelge belirlenmemesi, matematiksel anlamda daha kolay ifade edilmesini sağlamaktadır.

2.3 Benders Ayrıştırma Yöntemi

Optimizasyon problemlerinin çözümünde genel olarak tek aşamalı yaklaşımlar kullanılmaktadır. Fakat bu yaklaşım karar değişkenlerinin ve kısıtların sayısı fazla olduğu karmaşık modellerin çözümü için yetersiz kalmaktadır. Bu sebeple, geleneksel tek aşamalı çözüm yöntemlerine alternatif çok aşamalı yöntemler önerilmektedir. Literatürde ayrıştırma yöntemi olarak tanımlanan bu yöntemler, karmaşık değişkenlerle tanımlanan büyük bir problemi çözmek yerine, birden çok küçük problemi çözmenin daha kolay olacağı ilkesine dayanmaktadır.

Benders [22] tarafından karma tam sayılı matematiksel programlama problemleri için önerilen ayrıştırma algoritmasında, tam sayılı değişkenlerin değerlerinin sabitlenmesiyle problem, ilgili değerlere göre tanımlanan doğrusal programlama problemlerine dönüşmektedir. Bu noktada doğrusal programlama modellerinin kolay çözülebilir olmasından yararlanılarak, karmaşık problemler için çözüm elde edilebilmektedir.

Benders ayrıştırma algoritmasına göre esas problem, "*ana problem*" ve "*alt problemler*" olarak ayrıştırılmaktadır. Ana problem, esas problemin gevşetmesi olarak tanımlanmaktadır. Bu anlamda ana problemin çözümü, esas problem için bir alt sınır (en küçükleme) değeri belirtmektedir. Ana problemde genellikle tam sayılı değişkenler gibi, karmaşık değişken olarak ifade edilen değişkenler ele alınmaktadır. İlgili değişkenlerin ana problem çözümünden elde edilen değerlere göre, alt problemler oluşturulmaktadır. Burada alt problemlerin etkili çözüm yöntemleri bilinen bağımsız optimizasyon problemlerine dönüşmesi, çok aşamalı çözüm yaklaşımlarının temelini oluşturmaktadır [23]. Bu alt problemlerin çözümü neticesinde kalan değişkenlerin değerleri elde edilmektedir. Burada elde edilen değerler, esas problemin çözümüyle ilgili bilgiler vermektedir ve bu bilgiler bir kesi ile ifade edilmektedir. Alt problemin çözümü esas problem ile ilgili yeni bir bilgi türetmeyene, bir başka ifadeyle gevşetilmiş ana probleme eklenebilecek kesiler tükenene kadar, ilgili kesiler ana probleme eklenerek problem tekrar çözdürülmektedir. Benders [22] tarafından geliştirilen yöntemde, kesiler doğrusal programlama eşleniklik teoremine göre belirlenmektedir. Bu anlamda aşağıda (2.1) ile belirtilen problemi değerlendirelim.

$$\text{Min } c^T x + f^T y \tag{2.1}$$

Öyle ki

$$Ax + By = b \tag{2.1a}$$

$$x \geq 0, y \in R^n \tag{2.1b}$$

Esas problem (2.1), problem (2.2) şeklinde ifade edilebilmektedir.

$$\text{Min } f^T y + g(y) \tag{2.2}$$

Öyle ki

$$y \in R^n \tag{2.2a}$$

Burada $g(y)$ ile ifade edilen, y ile tanımlanan karmaşık değişkenin belirli bir değeri için tanımlanan alt problemdir.

$$\text{Min } c^T x \tag{2.3}$$

Öyle ki

$$Ax = b - By \tag{2.3a}$$

$$x \geq 0 \tag{2.3b}$$

Alt problem (2.3), y değişkeninin değerine bağlıdır. Alt problem (2.3) güçlü eşleniklik teoremine göre eşleniği ile de ifade edilebilmektedir. Bu sayede eşlenik alt problem (2.4) ile y değişkeninden bağımsız bir model tanımlanabilmektedir.

$$\text{Min } a^T (b - By) \tag{2.4}$$

Öyle ki

$$A^T a \leq c \quad (2.4a)$$

$$a \text{ sınırsız} \quad (2.4b)$$

Burada problem (2.4) olursuz ise, ya y değişkeninin tüm değerleri için birincil problem (2.3) de olursuzdur ya da y değişkeninin bazı değerleri için birincil problem (2.3) sınırsızdır. Eğer birincil problem (2.3) olursuz ise, esas problem (2.1) de olursuz ve benzer şekilde birincil problem (2.3) sınırsız ise, esas problem (2.1) de sınırsız olacaktır. Bu sebeple burada tanımlar, eşlenik problem (2.4) için olurlu çözüm olduğu varsayımına göre yapılmaktadır.

Buna göre eşlenik problem (2.4) uç noktalar $\left(a_p^1, \dots, a_p^I\right)$ ve ışınlar $\left(a_r^1, \dots, a_r^J\right)$ üzerinden tanımlanabilmektedir. Problem (2.5) ile ifade edilen bu gösterime göre sırasıyla Kısıt (2.5a) ve (2.5b) ile eşlenik problemin (2.4) çözüm alanı sınırlandırılmakta ve eşlenik problem (2.4) için en büyük amaç fonksiyonunun değeri tanımlanmaktadır. Bir başka ifadeyle Kısıt (2.5a) ile birincil problemin olurlu olması sağlanırken, Kısıt (2.5b) ile y değişkeninin belirli bir değeri için ilgili problemin optimal değeri belirlenmektedir.

$$\text{Min } g \quad (2.5)$$

Öyle ki

$$\left(a_r^j\right)^T (b - By) \leq 0 \qquad \forall j = 1, \dots, J \quad (2.5a)$$

$$\left(a_P^j\right)^T (b - By) \leq g \qquad \forall i = 1, \dots, I \quad (2.5b)$$

$$g \text{ sınırsız} \quad (2.5c)$$

Bu doğrultuda $g(y)$ ile belirtilen alt problem (2.3), y değişkeninin belirli bir değeri için tanımlanan problem (2.5) ile ifade edilebilmektedir. Bu tanıma göre problem (2.2), problem (2.6) şeklinde düzenlenebilmektedir.

$$\text{Min } f^T y + g \quad (2.6)$$

Öyle ki

$$\left(a_r^j\right)^T (b - By) \leq 0 \qquad \forall j = 1, \dots, J \quad (2.6a)$$

$$\left(a_P^j\right)^T (b - By) \leq g \qquad \forall i = 1, \dots, I \quad (2.6b)$$

$$y \in R^n, g \text{ sınırsız} \quad (2.6c)$$

Benders ayrıştırma algoritmasına göre, problem (2.6)'yı tanımlamak için gereken uç noktalar ve ışınların tamamını ilk aşamada belirlemek yerine, problem öncelikle ilgili nokta ve ışınların bir alt kümesi için değerlendirilmektedir. Esas problem (2.1)'in gevşetilmesi ile oluşturulan problem (2.6), ana problem olarak tanımlanmaktadır. Ana problemin çözümünden (y^*, g^*) değerleri elde edilmektedir. Alt problemin eşleniğinin (2.4) çözümünden ise $g(y^*)$ değeri belirlenmektedir. Bu noktada eğer eşlenik problem (2.4), y değişkeninin belirli bir değeri için olursuz ise, Kısıt (2.6a) şeklinde ifade edilen ilgili olurluluk kesisi ana probleme eklenerek olursuz çözüm kısıtlandırılmaktadır. Eğer eşlenik problemin çözümünden elde edilen amaç fonksiyonu değeri $g(y^*)$, ana problemde ilgili değeri belirten değişken g^* değerinden büyük ise, Kısıt (2.6b) şeklinde ifade edilen optimallik kesisi ana probleme eklenerek ilgili değişkenin değerinin atandığı kısıt sıkılaştırılmaktadır. Algoritma $g(y^*) = g$ sağlandığında (y^*, g^*) optimal çözümü ile sonlanmaktadır. [24]

Benders ayrıştırma algoritması sayesinde büyük bir problem aynı anda çözülebilecek birden çok küçük probleme ayrıştırıldığı için, problemin çözüm süresi azalmaktadır. Bu nedenle literatürde planlama ve çizelgeleme, taşıma ve araç rotalama, tesis yer seçimi problemi gibi pek çok karmaşık optimizasyon probleminin çözülmesinde bu algoritma kullanılmaktadır [25]. Benders ayrıştırma algoritması kullanılan çalışmaların artmasıyla, algoritmanın türevleri geliştirilmiştir. Geoffrion [23] çalışmasında Benders ayrıştırma algoritmasını doğrusal olmayan programlama problemleri için tanımlamaktadır. Literatürde genelleştirilmiş Benders ayrıştırma algoritması olarak bilinen bu yöntem sayesinde, doğrusal olmayan konveks eşleniklik teoreminden yararlanarak konveks alt problemlerden kesi türetilmektedir. Hooker ve Ottoson [26] tarafından tasarlanan mantıksal Benders ayrıştırma algoritmasında ise problemin doğrusal veya doğrusal olmayan programlama problemi olması gerekmemektedir. Bu ayrıştırma yöntemine göre alt problemler karma tam sayılı doğrusal programlama, doğrusal olmayan programlama ve kısıt programlama gibi herhangi bir programlama problemi olarak tanımlanabilir. Literatürde mantıksal ifadeler içeren optimizasyon problemlerinin yaygınlaşması doğrultusunda önerilen bu ayrıştırma algoritmasının en önemli katkılarından biri, literatüre matematiksel ve kısıt programlama yaklaşımlarını bir arada içerebilen bir çözüm yöntemi kazandırmasıdır [26].

Mantıksal Benders ayrıştırma algoritmasında kesiler, geleneksel Benders algoritmasıyla benzer şekilde, alt problemin eşleniğinin çözümünden elde edilmektedir. Fakat burada geleneksel yaklaşımdan farklı olarak, eşlenik problem tüme varım eşleniği olarak tanımlanmaktadır ve bu eşlenik problemin çözümünden her problem için genel geçerli kesiler türetilememektedir. Farklı problemlerin tüme varım eşleniğinin çözümünden elde edilen optimallik kanıtına göre, probleme özgü kesilerin tanımlanması gerekmektedir [26].

Zarandi ve Beck [27] yer seçim – atama problemini, tasarladıkları mantıksal Benders ayrıştırma algoritması ile çözmüştür. Esas problem, açılacak tesislerin ve ilgili tesislere atanacak araç sayısının belirlendiği ana problem ve araçların müşterilere atandığı alt problemler olarak ayrıştırılmıştır. Burada ana problem tam sayılı programlama, alt problemler ise kısıt programlama modeli olarak tanımlanmıştır. Bu çalışmanın sonucunda geliştirilen mantıksal Benders ayrıştırma algoritması ile tam sayılı programlamaya göre daha kısa sürede çözüm elde edildiği gösterilmiştir. Ayrıca tabu arama algoritmasıyla yapılan karşılaştırmaya göre, tasarlanan ayrıştırma algoritmasının tabu arama algoritmasından daha kısa sürede daha iyi olurlu çözümler ürettiği belirtilmiştir.

Wheatley vd. [28] uçak parçası tedarik eden bir firma için servis seviyesi kısıtlarıyla tanımlanan bir yer seçim – atama problemini ele almıştır. Problem stokastik ve doğrusal olmayan bir karma tam sayılı matematiksel model olarak ifade edilmiştir. Tasarlanan ayrıştırma algoritmasına göre ana problem, esas problemde doğrusal olmayan servis seviyesi kısıdının kaldırılması ile tanımlanmıştır. Olurluluk problemi olarak tanımlanan alt problemlerde ise, ana problemden elde edilen çözümün ilgili servis seviyesi kısıdına uygunluğu kontrol edilmektedir. Algoritma, tanımlanan olurluluk kısıtları sayesinde ana problemden elde edilen alt sınır değeri ve alt problemlerden elde edilen üst sınır değeri birbirine eşit olduğunda sonlanmaktadır.

Hooker [29] paralel kaynak kullanan ürünlerin kaynaklara atanması ve çizelgelenmesi problemi için karma tam sayılı matematiksel programlama ve kısıt programlamayla bir ayrıştırma algoritması tasarlamıştır. Problem maliyet, tamamlanma zamanı ve toplam gecikme olmak üzere üç ayrı amaç için tanımlanmıştır. Ana problem ürünlerin kaynaklara atanması, alt problemler ise bu atamalara göre çizelgelerin belirlenmesi şeklinde ayrıştırılmıştır. Bu ayrıştırma algoritması sayesinde optimal çözüm karma tam sayılı ve kısıt programlama yaklaşımlarına göre daha kısa sürede belirlenebilmiştir. Bunun dışında, çalışma sonucunda ayrıştırma algoritmasının büyük örnekler için sınır değeri belirlemek adına da güçlü bir yöntem olduğu gösterilmiştir.

Tran vd. [30] başka bir atama ve çizelgeleme problemini ele almıştır. Sıra bağımlı ayar zamanları ile paralel makine çizelgeleme problemi, karma tam sayılı matematiksel model olarak ifade edilmiştir. Ana problem, esas problem değişkenlerinin gevşetilmesiyle elde edilen karma tam sayılı atama problemidir. Hooker [29]'ın çalışmasından farklı olarak burada ana problemdeki sıralama değişkenleri tamamen kaldırılmak yerine, daha güçlü bir alt sınır değeri elde edebilmek adına gevşetilmiştir. Ana problemde yapılan makine atamaları doğrulusunda, alt problemler asimetrik gezgin satıcı problemi olarak ayrıştırılmıştır. Problemin amacı yayılma zamanını en küçüklemektir. Problem tanımı kapsamında ana problemde yapılan bütün atamalar esas problem için olurlu bir çizelgeyi, bir başka ifadeyle bir üst sınır değerini belirtmektedir. Burada tanımlanan optimallik kesileri ile gevşetilmiş ana problem güçlendirilerek, ana problemin belirttiği alt sınır değeri sıkılaştırılmıştır. Mevcut çalışmada Tran vd. [30] farklı olarak, tek makine çizelgeme problemi ele alınmaktadır. Bu anlamda paralel makine çizelgeleme problemi daha karmaşık bir problem seviyesini belirtmesine rağmen, tek makine problemi ayrıştırma algoritmasının uygulanması adına daha karmaşık bir yapı oluşturmaktadır. Bunun dışında Tran vd. [30]'de teslim tarihine ait kısıtlar bulunmadığı için, herhangi bir atama olurlu bir çizelge oluşturmaktadır. Bu çalışmada ise teslim tarihleri anlamında olurlu bir çizelge tanımlayabilmek adına olurluluk kesilerinin tanımlanması gerekmektedir. Tran vd. [30] çalışması kapsamında algoritma, kesilerin iki farklı yaklaşımla eklenmesiyle uygulanmaktadır. İteratif yaklaşım olarak adlandırılan ilk yaklaşımda kesiler ana problemin tamamen çözülmesinin ardından tanımlanmaktadır ve bu kesilerin eklenmesiyle ana problem tekrar çözülmektedir. Literatürde dallandırma ve kontrol olarak adlandırılan diğer yaklaşımda ise kesiler, ana problemin çözümü esnasında dal ve sınır algoritmasında yapılan dallandırmalar üzerinden tanımlanmaktadır. Dallandırma ve kontrol yaklaşımında, iteratif yaklaşımdan farklı olarak ana problem sadece bir kere tamamen çözülmektedir. Bu çalışma sonucunda yapılan hesaplamalı çalışmalarla dal ve kontrol yaklaşımının, iteratif yaklaşımdan üstün bir performans sağladığı gösterilmiştir.

Çoban ve Hooker [31], literatürde ayrıştırma algoritmasının uygulandığı diğer problemlerden farklı yapıda bir problem ele almışlardır. Literatürde yer alan diğer çalışmalar genellikle, atama ve çizelgeleme gibi iki karar içermekte ve problem bu kararlardan bir tanesinin sabitlenmesi ile kolayca ayrıştırılabilmektedir. Çoban ve Hooker [31] ise sadece çizelgeleme kararını içeren tek kaynaklı, zaman pencereli bir çizelgeleme problemini geliştirdikleri ayrıştırma algoritmasıyla çözmüştürler. Ana problem işlerin katman olarak adlandırılan zaman dilimlerine atanmasını, alt problemler ise bu atanmalara göre oluşturulan çizelgeleme problemlerini içermektedir. Burada ana problem karma tam sayılı matematiksel programlama ile alt programlar ise kısıt programlama ile çözülmüştür. Problem iki farklı varsayıma göre tanımlanmıştır. İlk durumda, işlerin başladıkları katman içerisinde tamamlanması gerektiği varsayılmıştır. Katmansız yaklaşım olarak adlandırılan diğer yaklaşımda ise, işlerin bir katmandan diğerine aktarılmasına izin verilmektedir. Çalışma kapsamında olurluluk, tamamlanma zamanı ve toplam geç kalma problemleri ele alınmıştır. Fakat Çoban ve Hooker [31]'ın ele aldığı amaçlardan farklı olarak, mevcut çalışmada toplam fazla mesai en küçüklenmektedir. Bunun dışında, Çoban ve Hooker [31]'ın çalışmasında ele alınan problemler ayar zamanlarını içermemektedir. Bu anlamda, mevcut çalışma kapsamında ele alınan problemin daha karmaşık olduğu bilinmektedir.

2.4 Özet

Çalışma kapsamında bir gerçek hayat problemi temel alınmaktadır. Literatür taramasında bu problem tanımında yapılan varsayımlar için yapılan çalışmalar incelenmiştir.

Literatürde çizelgeleme sürecinde sıra bağımlı ayar zamanlarının değerlendirilmesine yönelik pek çok çalışma yapılmıştır, fakat fazla mesai uygulaması henüz yeteri kadar değerlendirilmemiştir. Sıra bağımlı ayar zamanları ve fazla mesai birlikte ilk defa Freeman vd. [21] çalışmasında değerlendirilmiştir. Bu çalışma kapsamında tanımlanan problemden farklı olarak, Freeman vd. [21] çalışmasında teslim tarihine ilişkin kısıtlar bulunmamaktadır. Bu doğrultuda Freeman vd. [21] çalışmasında doğrudan çizelgeleme periyodu boyunca yapılması gereken toplam fazla mesai belirlenmektedir. Mevcut çalışmada ise, işlerin tamamlanma zamanlarının gün sonlarında tanımlanan teslim tarihlerine uygunluğunu değerlendirebilmek adına günlük fazla mesailerin belirlenmesi gerekmektedir. Bu anlamda bu çalışma kapsamında günlük detayda bir haftalık üretim çizelgesi belirlenirken, Freeman vd. [21] çalışmasında değerlendirilen çizelgeleme periyodu bir güne denk gelmektedir.

Problem tanımına yönelik tarama ile birlikte problem için önerilen çözüm yöntemine yönelik çalışmalar değerlendirilmiştir. Bu anlamda Tran vd. [30] çalışması çizelgeleme sürecinde sıra bağımlı ayar zamanlarının değerlendirilmesi anlamında, mevcut çalışma kapsamında yapılan uygulamaya benzemektedir. Fakat optimallik kesileri temel alınan ilgili çalışmada, teslim tarihine ilişkin kısıt bulunmadığı için olurluluk kesisi tanımlanmamıştır. Bununla birlikte mevcut çalışmadan farklı olarak, Tran vd. [30] çalışmasında ele alınan paralel makine çizelgeleme problemi, literatürde incelenen diğer atama ve çizelgeleme problemleri gibi, karar yapısı nedeniyle kolayca ayrıştırılabilmektedir.

Bu anlamda bu çalışma kapsamında tanımlanan çizelgeleme probleminin yapısı, Çoban ve Hooker [31]'ın tanımladığı katmansız çizelgeleme problemine benzemektedir. Çoban ve Hooker [31] çalışmalarında bu problem için toplam geç kalma zamanı ele almamıştır. Ayrıca bu çalışmadan farklı olarak, mevcut çalışma sıra bağımlı ayar zamanlarını içermektedir. Mevcut çalışma burada özetlenen bu farklılıklarıyla literatüre katkı sağlamaktadır. .

3. PROBLEM TANIMI VE MATEMATİKSEL MODEL

Çalışma kapsamında bir haftalık üretim çizelgeleme problemi ele alınmaktadır. Üretim tek makinede gerçekleşmektedir. Çizelgeleme ufku, Şekil 3.1'de gösterildiği gibi gün bazında periyotlara bölünmüştür. Her gün, belirli bir normal mesai ve fazla mesai süresinden oluşmaktadır. Fazla mesai, normal mesainin ardından günlük olarak tanımlanmaktadır. Yapılan birim fazla mesai için, sabit normal mesai maliyeti dışında, bir maliyete katlanılmaktadır. Bu maliyet ürün veya üretim miktarından bağımsız, doğrudan fazla mesai yapılan süre ile orantılıdır.

İşler üretime başlamadan önce, makine ilgili işin üretimine hazırlanmak üzere ayarlanmaktadır. Ayar zamanları sıra bağımlıdır. Bu anlamda, bir k işi için yapılan ayar ve ayar zamanı, kendisinden önce üretimi tamamlanan j işine göre değişkenlik göstermektedir ve bu ayar zamanı A_{jk} ile gösterilmektedir. Şekil 3.1'de verilen çizelgede 3 numaralı iş ile 1 numaralı iş arasında A_{31} kadar bir zaman ayar yapılmıştır. Ayar zamanları $A_{jk} \leq A_{jm} + A_{mk}$ ile tanımlanan üçgen eşitsizliğine uymaktadır. Yani bir j işinin üretiminden doğrudan bir k işinin üretimine geçerken katlanılan ayar zamanı, j işinden herhangi bir m işine ve ardından k işinin üretimine geçerken katlanılan toplam ayar zamanından fazla değildir. Bu, gerçekçi bir varsayımdır. Çünkü, aksi durumda, bir işten diğerine geçerken, önce toplam ayar süresini azaltacak alakasız bir iş için ayar yapılıp, bu iş hiç üretilmeden, ardından asıl üretilecek iş için ayar yapılabilir ve ayar süresi düşürülebilir. Gün başına çizelgelenen işin üretimi için yapılacak ayar, bir önceki gün sonunda tamamlanan işe bağlıdır. Bu anlamda Şekil 3.1 ile gösterilen örnek çizelgede, ikinci günün başında üretimine başlanan iş için yapılması gereken ayar zamanı A_{26}'dır. Haftalık çizelgenin en başına çizelgelenecek iş için yapılacak ayar ise, makinenin hazır konumundan ilgili işin üretimine geçişte yapılacak ayardır.

Şekil 3.1: Örnek çizelge

İşler çizelgeleme periyodunun başında üretime hazır bulunmaktadırlar. Bir j işinin üretiminin en geç tamamlanabileceği tarih ise D_j ile gösterilmektedir. Burada teslim tarihleri gün sonlarına denk gelmektedir. Problem kapsamında teslim tarihlerinin aşılmasına izin verilmemektedir ve ilgili kısıtlar katı kısıt olarak ele alınmaktadır.

Problem tanımına göre iş, müşteri siparişini ifade etmektedir. Bu anlamda bir iş, bir siparişe ait birden çok ürünün üretimini kapsamaktadır. Burada ürünlerin üretim süreleri belirlidir ve bir işin toplam işlem zamanı P_j ile gösterilmektedir. Üretimde iş kesmeye izin verilmemektedir. Yani bir işin üretiminin başlangıcından ilgili P_j işlem zamanı tamamlanana kadar olan zamana başka bir iş çizelgelenemez. Bunun dışında üretimin ardışık günler arasında parçalanmasına izin verilmektedir. Bu durumda i günü içerisinde üretimi tamamlanmayan bir ürünün üretimine $(i+1)$ gününde devam edilebilmektedir. Şekil 3.1'de üretimi ikinci gün içerisinde tamamlanmayan ve üçüncü günün başında devam eden 4 numaralı iş parçalı üretime bir örnektedir. Burada üretim ayarları parçalanamamaktadır. Bu doğrultuda ayar zamanı gün içerisinde tamamlanamayacak bir ürün ilgili güne çizelgelenememektedir. Bunun yanı sıra, ayarı yapılarak üretime hazırlanan bir işin üretimine ertesi gün başlanmasına izin verilmemektedir.

Problem kapsamında olurlu bir çizelge, işlerin üretiminin teslim tarihlerinden önce tamamlandığı bir çizelgeyi ifade etmektedir. Bu anlamda normal mesai süresi (ns) içerisinde olurlu bir çizelge elde edilememesi durumunda fazla mesai üretimi ile işler yetiştirilmektedir.

Problemin amacı ilgili fazla mesai maliyetini en küçükleyen haftalık üretim çizelgesini belirlemektir. Geliştirilen karma tam sayılı doğrusal programlama modeli ve metin boyunca kullanılan notasyon aşağıda verilmiştir.

Kümeler

N	=	$\{1, \ldots, n\}$: İş kümesi
T	=	$\{1, \ldots, t\}$: Gün kümesi

Parametreler

P_j	=	j işinin sabit işlem zamanı,	$j \in N$
D_j	=	j işinin teslim tarihi,	$j \in N$
A_{jk}	=	j işinden k işine geçişte katlanılan ayar zamanı,	$k, j \in N$
ns	=	Normal mesai içerisinde üretime ayrılan süre	
ds	=	Gün içerisinde üretime ayrılan toplam süre	

Karar Değişkenleri

y_{jk}	=	$\begin{cases} 1 & j \text{ işi } k \text{ işinden önce çizelgelendiyse} \\ 0 & \text{diğer durumlarda} \end{cases}$	$k, j \in N$
st_{ji}	=	$\begin{cases} 1 & j \text{ işi } i \text{ gününde üretime başladıysa} \\ 0 & \text{diğer durumlarda} \end{cases}$	$j \in N, i \in T$
z_{ji}	=	$\begin{cases} 1 & j \text{ işi için } i \text{ gününde fazla mesai yapıldıysa} \\ 0 & \text{diğer durumlarda} \end{cases}$	$j \in N, i \in T$
d_{ji}	=	$\begin{cases} 1 & j \text{ işinin üretimi } i \text{ günü normal mesaide başladıysa} \\ 0 & j \text{ işinin üretimi } i \text{ günü fazla mesaide başladıysa} \end{cases}$	$j \in N, i \in T$
o_{ji}	=	$\begin{cases} 1 & j \text{ işi üretimi } i \text{ gününde veya öncesinde tamamlandıysa} \\ 0 & \text{diğer durumlarda} \end{cases}$	$j \in N, i \in T$
S_j	=	j işinin üretime başlama zamanı	$j \in N$
C_j	=	j işinin üretiminin tamamlanma zamanı	$j \in N$
F_{ji}	=	j işi için i gününde yapılan fazla mesai	$j \in N, i \in T$
B_{ji}	=	j işi için i gününde yapılabilecek fazla mesai	$j \in N, i \in T$
θ_{ji}	=	$S_j \cdot z_{ji}$ doğrusallaştırmak için tanımlanan değişken	$j \in N, i \in T$

Min $\sum_{i=1}^{t}\sum_{j=1}^{n} F_{ji}$

Öyle ki

$$\sum_{j=0}^{n} y_{jk} = 1 \qquad \forall k \in N \qquad (3.1)$$

$$\sum_{k=1}^{n} y_{jk} \leq 1 \qquad \forall j \in N \cup \{0\} \qquad (3.2)$$

$$\sum_{i=1}^{t} (i-1) \cdot ds \cdot st_{ji} \leq S_j \qquad \forall j \in N \qquad (3.3)$$

$$\sum_{i=1}^{t} st_{ji} = 1 \qquad \forall j \in N \qquad (3.4)$$

$$\sum_{i=1}^{t} i \cdot ds \cdot st_{ji} - S_j \geq \sum_{k=0}^{n} A_{jk} \cdot y_{jk} + 1 \qquad \forall j \in N \qquad (3.5)$$

$$S_k \geq C_j - M(1 - y_{jk}) \qquad \forall j,k \in N \cup \{0\} \qquad (3.6)$$

$$C_j \leq D_j \qquad \forall j \in N \qquad (3.7)$$

$$C_k = S_k + P_k + \sum_{j=0}^{n} A_{jk} \cdot y_{jk} + \sum_{i=1}^{t} (B_{ki} - F_{ki}) \qquad \forall k \in N \qquad (3.8)$$

$$(i \cdot ds) - (ds - ns) - S_j \leq M \cdot d_{ji} \qquad \forall j \in N, i \in T \qquad (3.9)$$

$$S_j - (i \cdot ds) + (ds - ns) \leq M(1 - d_{ji}) \qquad \forall j \in N, i \in T \qquad (3.10)$$

$$(i \cdot ds) - C_j \leq M \cdot o_{ji} \qquad \forall j \in N, i \in T \qquad (3.11)$$

$$C_j - (i \cdot ds) \leq M(1 - o_{ji}) \qquad \forall j \in N, i \in T \qquad (3.12)$$

$$B_{ji} \geq C_j - (i \cdot ds) + (ds - ns) - M(2 - d_{ji} - o_{ji}) \qquad \forall j \in N, i \in T \qquad (3.13)$$

$$B_{ji} \geq P_j + \sum_{j=0}^{n} A_{jk} \cdot y_{jk} - M\left(1 + d_{ji} - o_{ji}\right) \qquad \forall j \in N, i \in T \qquad (3.14)$$

$$F_{ji} \geq P_j + \sum_{j=0}^{n} A_{jk} \cdot y_{jk} - M(1 + d_{ji} - o_{ji}) \qquad \forall j \in N, i \in T \qquad (3.15)$$

$$B_{ji} \geq (i \cdot ds) - S_j - M(d_{ji} + o_{ji}) \qquad \forall j \in N, i \in T \qquad (3.16)$$

$$B_{ji} \geq (ds - ns) - M(1 - d_{ji} + o_{ji}) \qquad \forall j \in N, i \in T \qquad (3.17)$$

$$F_{ji} \leq B_{ji} \qquad \forall j \in N, i \in T \qquad (3.18)$$

$$\theta_{ji} \leq i \cdot ds \cdot z_{ji} \qquad \forall j \in N, i \in T \qquad (3.19)$$

$$\theta_{ji} \leq S_j \qquad \forall j \in N, i \in T \qquad (3.20)$$

$$\theta_{ji} \geq S_j - M(1 - z_{ji}) \qquad \forall j \in N, i \in T \qquad (3.21)$$

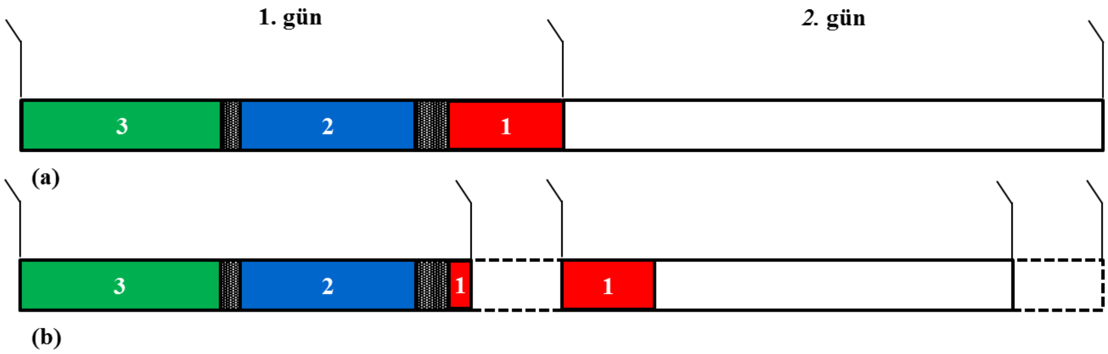
$$(i \cdot ds \cdot z_{ji}) - (ds - \ ns) \leq C_j \qquad \forall j \in N, i \in T \qquad (3.22)$$

$$y_{jk}, st_{ji}, z_{ji}, d_{ji}, o_{ji} \in \{0,1\} \qquad \forall j, k \in N, i \in T \qquad (3.23)$$

$$S_j, C_j, B_{ji}, F_{ji} \ \geq 0 \qquad \forall j \in N, i \in T \qquad (3.24)$$

Bu modelde, ilk denklem amaç fonksiyonunu belirtmektedir. Amaç fonksiyonu çizelgeleme periyodu boyunca yapılan toplam fazla mesaiyi göstermektedir. Kısıtlar (3.1) ve (3.2) sırasıyla öncül ve ardıl ilişkilerini belirten çizelgeleme kısıtlarıdır. Buna göre Kısıt (3.1) ile her işin öncesinde bir başka işin üretiminin çizelgelenmesi gerekliliği gösterilmektedir. Burada üzerinde kısıt tanımlanan işler doğrudan işler kümesinin (N) elemanlarını içerirken, öncül işlerin tanımlandığı küme bu işlerin yanında bir yapay iş içermektedir. Bu yapay iş, makinenin hazır durumunu ifade etmektedir. Kısıt (3.2) ise çizelgelenen son ürün hariç, her işin ardından başka bir işin üretileceğini göstermektedir. Üretime başlama zamanı ikili değişkeni st_{ji} ve sürekli değişkeni S_j Kısıt (3.3) ile ilişkilendirilmektedir. Burada başlama zamanını gösteren S_j, st_{ji} değişkeni ile gün biriminde ifade edilmektedir. Kısıt (3.4) her j işinin tek bir i gününde başlayabileceğini belirtmektedir. Başlama zamanı üzerine tanımlanan bir başka kısıt ise, Kısıt (3.5) ile gösterilen ayar zamanının parçalanamayacağını ifade eden kısıttır. Bu kısıt ile ayar zamanı i günü içerisinde tamamlanamayacak bir j işi ilgili güne atanmamaktadır. Burada aynı zamanda ayar zamanı tamamlanarak üretime hazırlanan işin üretimine ertesi gün başlanamayacağı da belirtilmektedir. Kısıt (3.6) üretimlerin çakışmasını önlemek adına, bir k işinin üretiminin, öncülü j işinin üretimi tamamlanmadan başlayamayacağını belirtmektedir. Teslim tarihine dair katı kısıtlar (3.7) ile ifade edilmektedir.

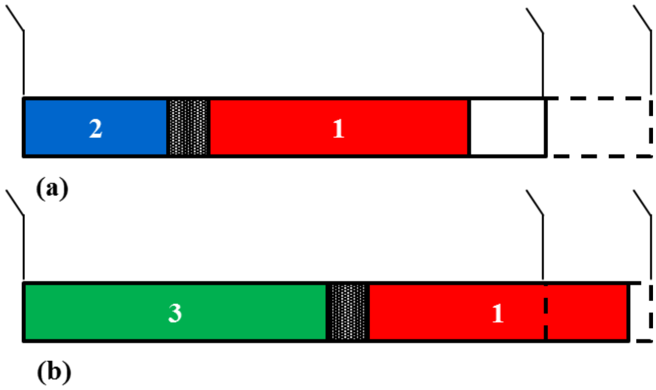
Kısıt (3.8) ile üretimin tamamlanma zamanı hesaplanmaktadır. Üretim kapasitesinin göz önünde bulundurulmadığı bir çizelgeleme probleminde, tamamlanma zamanları ardışık bir şekilde başlangıç zamanının üzerine ayar (A_{jk}) veya işlem (P_k) zamanının eklenmesiyle elde edilebilmektedir. Fakat bu problem kapsamında kapasite, amaç fonksiyonunda fazla mesaiyi en küçükleyecek şekilde değerlendirilmektedir. Bu anlamda fazla mesainin tamamen kullanılmaması durumunda tamamlanma zamanları ardışık değerlendirilemeyecektir. Sabit üretim kapasitesinin sınırsız kabul edilmesi ve fazla mesainin değerlendirilmesi durumlarının çizelgeleme sürecine etkisi sırayla Şekil 3.2 (a) ve Şekil 3.2 (b) ile gösterilmektedir. Şekil 3.2 (b) ile gösterilen şekilde, 1 numaralı iş, 1. gün üretime başlamış, üretimi fazla mesaide durmuş, 2. gün kaldığı yerden devam ederek tamamlanmıştır.



Şekil 3.2: (a) Üretim kapasitesinin sınırsız (b) Fazla mesai ile değerlendirmesinin çizelgelemeye etkisi

Fazla mesai yaklaşımında tamamlanma zamanı $B_{ji} - F_{ji}$ şeklinde hesaplanan miktar kadar ötelenmektedir. Burada B_{ji} ile ifade edilen j işi için i gününde yapılabilecek fazla mesai miktarı, d_{ji} ve o_{ji} ikili karar değişkenleri kullanılarak hesaplanmaktadır. Üretimin başlama zamanına ilişkin durumları belirten d_{ji} karar değişkeni Kısıt (3.9) ve (3.10) ile ifade edilmektedir. Burada j işinin üretimine i gününün normal mesaisi içerisinde başlanması durumu (3.9), fazla mesaisi içerisinde başlaması durumu ise (3.10) ile gösterilmektedir. Bunun yanı sıra Kısıt (3.11) ve (3.12) ise üretimin tamamlanmasına dair durumları ifade eden o_{ji} karar değişkenini hesaplamaktadır. Burada Kısıt (3.12) j işinin üretiminin i günü normal mesaisi veya fazla mesaisi içerisinde veya öncesinde tamamlandığı durumu ifade etmektedir.

Kısıt (3.13) ile ifade edilen j işinin üretiminin i günü normal mesaisinde başlayıp ($d_{ji} = 1$) sırasıyla ilgili günün normal ve fazla mesaisi içerisinde tamamlanması durumları ($o_{ji} = 1$) sırayla Şekil 3.3 (a) ve Şekil 3.3 (b) ile gösterilmektedir. İlk şekilde 1 numaralı iş normal mesaide başlayıp normal mesaide tamamlanmıştır. İkincisinde ise, normal mesaide başlayıp aynı günün fazla mesaisinde tamamlanmıştır. Üretimin normal mesai içerisinde tamamlandığı ilk durumda $B_{ji} = F_{ji} = 0$, fazla mesaide tamamlandığı ikinci durumda ise $B_{ji} = F_{ji} > 0$ sağlanacak ve ilgili işin üretimin tamamlanma zamanı, doğrudan başlama zamanının üzerine ayar ve işlem zamanlarının eklenmesiyle belirlenecektir.

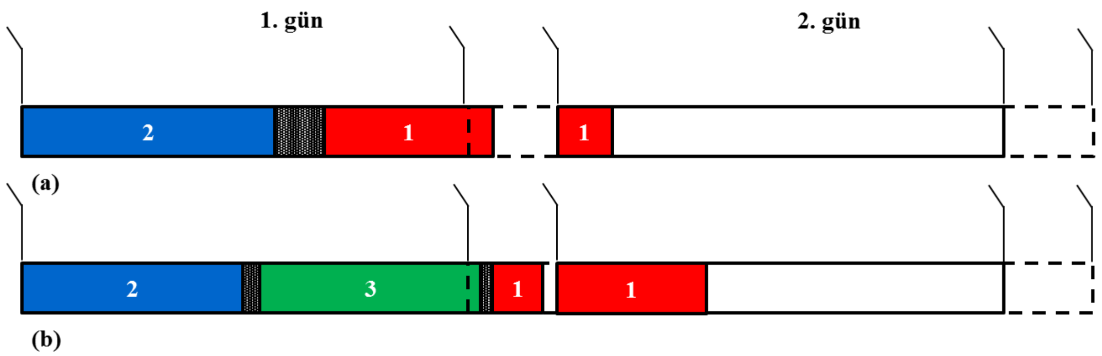


Şekil 3.3: Normal mesai içerisinde başlayıp (a) normal mesai (b) fazla mesai içerisinde tamamlanma durumu

Benzer şekilde, j işinin üretimine i günü fazla mesaisinde başlanıp $(d_{ji} = 0)$ ilgili gün içerisinde tamamlanması $(o_{ji} = 1)$ durumunda da $B_{ji} = F_{ji} = P_j + \sum_{k=0}^{n} A_{kj} \cdot y_{kj}$ sağlanacak, fakat bu değişkenlerin değerleri arasındaki fark, bir başka ifadeyle öteleme miktarı, sıfıra eşit olacaktır. Modelde Kısıt (3.14) ve (3.15) ile ifade edilen bu durum Şekil 3.4 ile gösterilmektedir. Bu şekilde, 1 numaralı iş fazla mesaide başlayıp yine fazla mesaide tamamlanmıştır.

Şekil 3.4: Fazla mesai içerisinde başlayıp gün içerisinde tamamlanma durumu

Son olarak Kısıt (3.16) ve (3.17) sırasıyla j işinin üretiminin i günü normal mesai $(d_{ji} = 1)$ ve fazla mesaisi $(d_{ji} = 0)$ içerisinde başladığı ve gün içerisinde tamamlanmadığı durumları belirtmektedir. Şekil 3.5 (a)'da gösterilen ilk durumda 1 numaralı işin üretimine 1. günün fazla mesaisi süresince devam edilebilmektedir. İkinci durumda ise 1 numaralı iş için üretime başlanan zamandan ilgili günün sonuna kadar fazla mesai yapılabilmektedir. Bu durum Şekil 3.5 (b)'de gösterilmektedir.



Şekil 3.5: (a) Normal mesai (b) fazla mesai içerisinde başlayıp gün içerisinde tamamlanmama durumu

Fazla mesai yapılabilecek süre B_{ji} ve yapılan süre F_{ji} arasındaki ilişki Kısıt (3.18) ile belirtilmektedir. Kısıt (3.19) – (3.22) ile fazla mesai ikili değişkeni z_{ji}, başlama zamanı S_j ve tamamlanma zamanı C_j ilişkilendirilmektedir. Burada bir j işi için i gününde fazla mesai yapılıyorsa, (3.19) - (3.21) numaralı kısıtlar bu işin üretimine ilgili fazla mesaide veya daha önce başlanmış olması gerektiğini belirtmektedir. Bunun yanı sıra Kısıt (3.22) ise bu j işinin tamamlanma zamanının fazla mesai yapılan i gününden önce olmayacağını göstermektedir. (3.23) ve (3.24) numaralı kısıtlar ise karar değişkenlerine ait işaret kısıtlarıdır.

Geliştirilen bu model karma tam sayılı doğrusal bir yapıdadır. IBM ILOG CPLEX OPL dilinde kodlanan bu model ile yapılan testlerde, iş sayısının az olduğu örnekler için optimal çözüme makul sürelerde ulaşılmıştır. Fakat, iş sayısının fazla olduğu büyük boyutlu gerçek hayat örneklerinde makul sürelerde optimal çözüme ulaşılamadığından, çözüm için Benders ayrıştırması tabanlı kesin çözüm yöntemi ve tavlama benzetimi sezgiseli geliştirilmesine karar verilmiştir. Sonraki bölümde geliştirilen çözüm yöntemleri detaylıca açıklanacaktır.

4. ÇÖZÜM YÖNTEMİ

4.1 Benders Ayrıştırma Yöntemi

Bu kısımda problemin çözümü için mantıksal Benders ayrıştırma algoritması geliştireceğiz. Literatürde ayrıştırma algoritmalarının kullanıldığı çalışmalarda, atama ve çizelgeleme problemleri gibi, genellikle aşamalı kararlar içeren ve bir değişkenin değerinin sabitlenmesi ile doğrudan bu değere bağlı kolay çözülebilir problemlere ayrıştırılabilen problemler ele alınmıştır. Bu çalışmada incelenen problem ise sadece çizelgeleme kararını içermektedir. Bu anlamda problem, literatürde karşılaşılan diğer çalışmalar gibi açıkça birbirinden ayrıştırılabilecek kararlardan oluşmamaktadır. Bu nedenle ayrıştırma doğrudan esas problemde karar değişkenlerinin değerinin sabitlenmesi veya kısıtların kaldırılması şeklinde tanımlanamamaktadır. Burada kullanılan ayrıştırma yaklaşımında, Çoban ve Hooker [31] çalışmasında uygulanan ayrıştırma yaklaşımı temel alınmıştır.

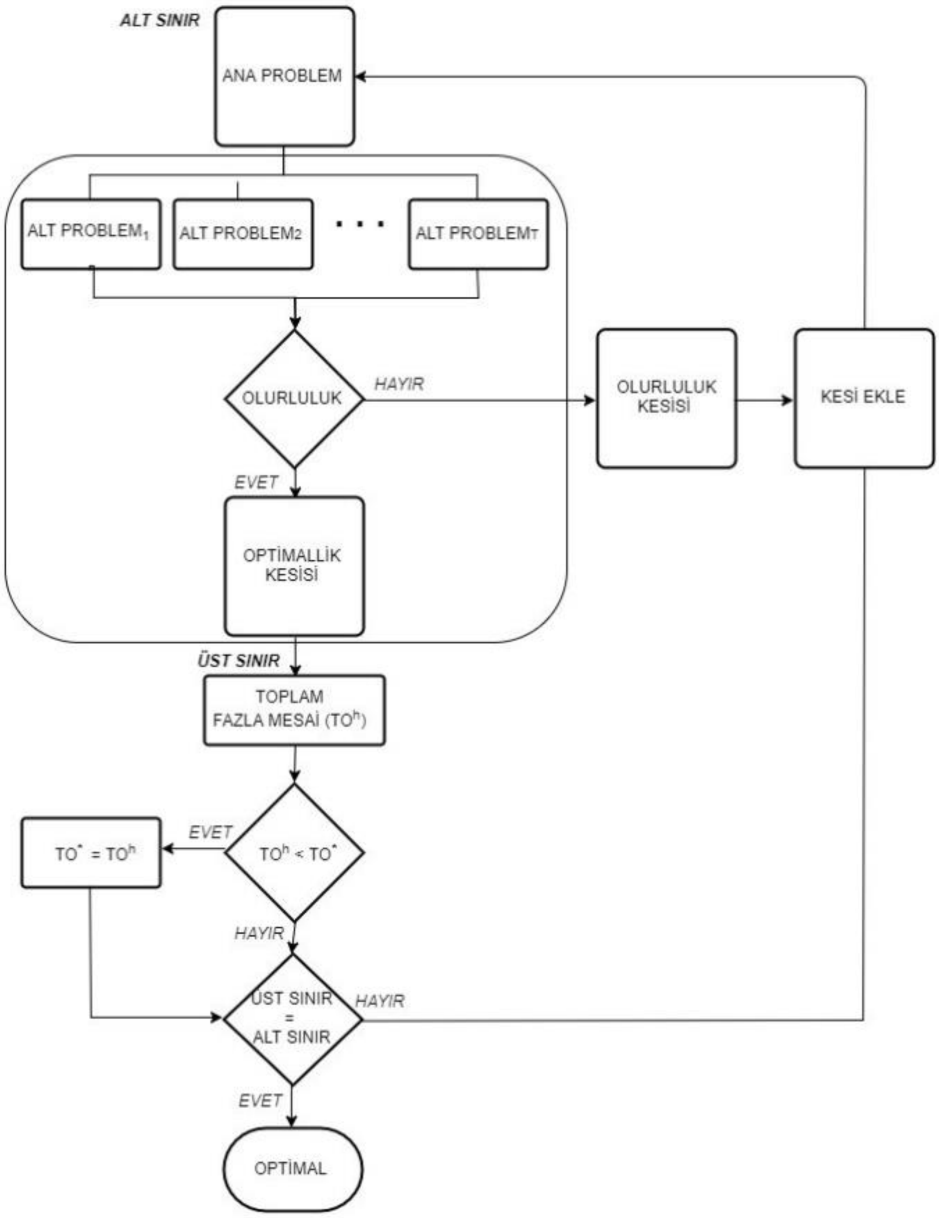
Bu doğrultuda ana problem ve alt problem, problem tanımı esas alınarak ayrıca tanımlanmıştır. Ana problem üretimin haftanın günlerine atanma problemi, alt problemler ise bu atamalar doğrultusunda günlük çizelgelerin belirlenmesi problemi olarak ayrıştırılmıştır.

4.1.1 Mantıksal Benders ayrıştırma algoritması

Geliştirilen mantıksal Benders ayrıştırma algoritması ile esas problem karma tam sayılı atama ana problemine ve kısıt programlama çizelgeleme alt problemlerine ayrıştırılmaktadır. Ana problemde yapılan gün atamaları doğrultusunda günlük çizelgeme alt problemleri oluşturulmaktadır. Günlük alt problemler sırayla çözdürülmektedir. Tasarlanan algoritma kapsamında bir iterasyon, ana problem doğrultusunda oluşturulan alt problemlerin çözülmesiyle, bu çözümlerden ilgili kesilerin tanımlanmasına kadar olan adımların tamamı olarak tanımlanmaktadır.

Herhangi bir alt problemden, esas problem için olurlu bir çizelge elde edilememesi durumunda olursuzluğun nedenine göre ilgili olurluluk kesisi tanımlanarak iterasyon sonlandırılmaktadır. Olurlu alt problemler için ise optimallik kesisi oluşturulmaktadır. Alt problemlerden herhangi biri olursuz olduğunda ilgili iterasyon sonlandırılacağı için, olursuz alt probleme kadar tanımlanan bütün optimallik kesileri ilgili olurluluk kesisiyle birlikte ana probleme eklenmektedir. Bunun dışında, eğer ana problemde yapılan bütün atamalar için alt problemlerde olurlu çizelgeler belirlenebiliyor ise, sınır değerlerini sıkılaştırarak optimal çözümü belirlemek adına tanımlanan bütün optimallik kesileri ana probleme eklenerek, iterasyonlar tekrarlanmaktadır.

Tasarlanan ayrıştırma algoritmasının adımları Şekil 4.1 ile özetlenmektedir.



Şekil 4.1: Mantıksal Benders ayrıştırma algoritmasının adımları

4.1.2 Ana problem

Ana problem çizelgelenecek işlerin üretimlerinin günlere atanması olarak tanımlanmıştır. Problem karma tam sayılı matematiksel programlama modeli ile ifade edilmiştir. Bu atama problemi kapsamında karmaşık sıralama kararları incelenmemekte, problemin ifadesinde sadece atama karar değişkenleri kullanılmaktadır. Bu doğrultuda y_{ji} ikili değişkeni j işinin i gününe atanıp atanmadığını göstermektedir. Problem tanımına göre üretimin parçalanmasına, yani i günü içerisinde üretimi tamamlanamayan bir j işinin üretimine $(i+1)$ gününde devam edilmesine izin verilmektedir. Bu şekilde parçalı üretimle atanan işler y_{jik} ikili karar değişkeni ile tanımlanmıştır. Eğer bir j işi parçalı üretimle i gününün başına atandıysa $k=1$ durumu için, benzer şekilde sonuna atandıysa da $k=2$ durumu için 1 değerini almaktadır. Üretimine başlanan gün içerisinde tamamlanan işler, bir başka ifadeyle parçalı üretime atanmayan işler ise $k=0$ durumuyla belirtilmektedir. Parçalı üretime atanan işlerin atanma miktarı ise x_{ji} sürekli değişkeni ile gösterilmektedir. Burada atanma miktarı x_{ji} ile j işinin i gününe atanan işlem zamanı ifade edilmektedir. Atama ikili değişkeni ile benzer şekilde x_{jik} tanımlanmıştır. Burada işlerin gün başına atanan miktarı $k=1$ durumu, gün sonuna atanan miktarı ise $k=2$ durumu ile ifade edilmiştir. Ana problemin tanımında kullanılan notasyon ve model aşağıda sunulmuştur.

Karar Değişkenleri

$y_{ji} \quad = \quad \begin{cases} 1 & j \text{ işinin üretimi } i \text{ gününe atandıysa} \\ 0 & \text{diğer durumlarda} \end{cases} \qquad j \in N, i \in T$

$y_{ji0} \quad = \quad \begin{cases} 1 & j \text{ işi } i \text{ günü içerisinde başlayıp tamamlandıysa} \\ 0 & \text{diğer durumlarda} \end{cases} \qquad j \in N, i \in T$

$y_{ji1} \quad = \quad \begin{cases} 1 & j \text{ işi } i \text{ gününün başına parçalı üretime atandıysa} \\ 0 & \text{diğer durumlarda} \end{cases} \qquad j \in N, i \in T$

$y_{ji2} \quad = \quad \begin{cases} 1 & j \text{ işi } i \text{ gününün sonuna parçalı üretime atandıysa} \\ 0 & \text{diğer durumlarda} \end{cases} \qquad j \in N, i \in T$

x_{ji} = j işinin i gününde üretilen miktarı $\qquad j \in N, i \in T$

x_{ji1} = j işinin i gününün başında üretilen miktarı $\qquad j \in N, i \in T$

x_{ji2} = j işinin i gününün sonunda üretilen miktarı $\qquad j \in N, i \in T$

C_{max}^{i} = i gününün yayılma zamanı $\qquad i \in T$

O_i = i gününün fazla mesai miktarı $\qquad i \in T$

Min $\sum_{i=1}^{t} O_i$

Öyle ki

$$\sum_{i=1}^{t} x_{ji} = P_j \qquad \forall j \in N \qquad (4.1)$$

$$\sum_{i=1}^{t} y_{ji} \geq 1 \qquad \forall j \in N \qquad (4.2)$$

$$y_{ji} = y_{ji0} + y_{ji1} + y_{ji2} \qquad \forall j \in N, i \in T \qquad (4.3)$$

$$\sum_{i=1}^{t} y_{jik} \leq 1 \qquad \forall j \in N, k \in \{0, \dots, 2\} \qquad (4.4)$$

$$\sum_{i=1}^{t} (y_{ji0} + y_{ji2}) = 1 \qquad \forall j \in N \qquad (4.5)$$

$$y_{ji2} \leq y_{j(i+1)1} \qquad \forall j \in N, i \in T - \{t\} \qquad (4.6)$$

$$y_{j11} = y_{jt2} = 0 \qquad \forall j \in N \qquad (4.7)$$

$$x_{ji1} \leq P_j \cdot y_{ji1} \qquad \forall j \in N, i \in T \qquad (4.8)$$

$$x_{ji2} \leq P_j \cdot y_{ji2} \qquad \forall j \in N, i \in T \qquad (4.9)$$

$$x_{ji} = P_j \cdot y_{ji0} + x_{ji1} + x_{ji2} \qquad \forall j \in N, i \in T \qquad (4.10)$$

$$\sum_{j=1}^{n} x_{ji} \leq ds \qquad \forall i \in N \qquad (4.11)$$

$$\sum_{j=1}^{n} y_{ji2} = 1 \qquad \forall i \in T - \{t\} \qquad (4.12)$$

$$x_{ji2} \geq y_{ji2} \qquad \forall j \in N, i \in T \qquad (4.13)$$

$$(i \cdot ds) \cdot y_{jik} \leq D_j \qquad \forall j \in N, i \in T, k \in \{0,2\} \qquad (4.14)$$

$$C_{max}^{i} = \sum_{j=1}^{n} x_{ji} \qquad \forall i \in T \qquad (4.15)$$

$$O_i \geq C_{max}^{i} - ns \qquad \forall i \in T \qquad (4.16)$$

$$y_{ji}, y_{ji0}, y_{ji1}, y_{ji2} \in \{0,1\} \qquad \forall j \in N, i \in T \qquad (4.17)$$

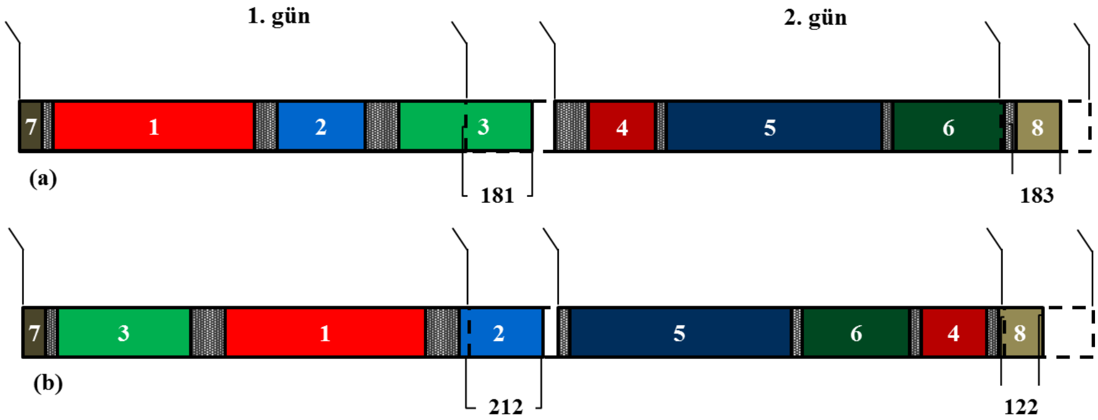
$$x_{ji}, x_{ji1}, x_{ji2}, C_{max}^{i}, O_i \geq 0 \qquad \forall j \in N, i \in T \qquad (4.18)$$

Ana problemin amacı, esas problem ile aynı şekilde haftalık toplam fazla mesai miktarını en küçüklemek olarak tanımlanmaktadır. Fakat ana problem kapsamında sıralama kararları ve bunun neticesinde sıra bağımlı ayar zamanları değerlendirilmediği için, bu formülasyonda fazla mesai değeri sadece işlem zamanları dikkate alınarak hesaplanmıştır. Bu anlamda ana problemin amaç fonksiyonunun değeri, esas problem için bir alt sınır ifade etmektedir.

Çizelgelenecek işlerin toplamda işlem zamanı kadar işlenmesi gerektiği Kısıt (4.1) ile belirtilmektedir. Kısıt (4.2) bütün işlerin en az bir güne atanması gerektiğini göstermektedir. Atama durumu değişkeni y_{jik} ile atama değişkeni y_{ji} arasındaki ilişki Kısıt (4.3) ile tanımlanmıştır. Buna göre, bir j işi i gününe atandıysa, bu iş ya günün başına $(k = 1)$ veya sonuna $(k = 2)$ parçalı üretime atanmak üzere, ya da üretimine başlanan gün tamamlanacak şekilde $(k = 0)$ atanma durumlarından herhangi birine atanması gerekmektedir. İşler bu atanma durumlarından herhangi birine en fazla bir kere atanabilir. Bununla birlikte üretimine başlanan gün içerisinde tamamlanmak üzere, herhangi bir günde $k = 0$ durumuna atanan bir iş, parçalı üretim durumlarından herhangi birine tekrar atanamaz. Bu durumlar sırasıyla Kısıt (4.4) ve (4.5) ile sağlanmaktadır. Problem tanımı kapsamında iş kesmeye izin verilmediği için, Kısıt (4.6) ile i günü içerisinde üretimi tamamlanamayan bir işin parçalı üretimle gün sonuna $(k = 2)$ atanması durumunda, $(i + 1)$ gününde tamamlanmak üzere, parçalı üretimle gün başına $(k = 1)$ atanması gerektiği gösterilmektedir.

Haftalık üretim çizelgeleri birbirinden bağımsız belirlendiği için, Kısıt (4.7) ile haftanın başına ve sonuna parçalı üretim ataması yapılamayacağı ifade edilmektedir. Kısıt (4.8) ve (4.9) ile parçalı üretim ikili değişkeni (y_{jik}) ve sürekli değişkeni (x_{jik}) ilişkilendirilmektedir. Atama miktarı değişkeni (x_{ji}) Kısıt (4.10) ile belirlenmektedir. Bu kısıda göre bir j işinin i gününe atanan miktarı, bu iş üretimine başlanan gün içerisinde tamamlanıyor ise doğrudan toplam işlem zamanına, tamamlanmıyor ise de parçalı üretime atanan miktara eşittir. Kısıt (4.11) ile bir güne atanan miktarın gün sınırlarını geçemeyeceği belirtilmektedir.

Ana problemden elde edilen atama kararları doğrultusunda alt problemler tanımlanmaktadır. Burada tanımlanan alt problemlerin kolay çözülebilir olması önerilen çözüm yönteminin etkinliği adına önemlidir. Bu doğrultuda büyük ve karmaşık bir haftalık çizelgeleme problemini tek adımda değerlendirmek yerine, ilgili problem atama kararlarına göre daha küçük günlük çizelgeleme alt problemlerine ayrıştırılmaktadır. Sıralamanın yapıldığı bu alt problemlerde, sıra bağımlı ayar zamanları değerlendirilmektedir. Parçalı üretim ile üretimine ertesi gün devam edilen bir iş için ilgili günün başında ayar yapılmamaktadır, fakat gün başında üretimine yeni başlanan bir iş için önceki günün sonunda tamamlanan işe göre ayar yapılmaktadır. Bu doğrultuda ilgili ardışık günlerin çizelgeleme alt problemlerinin birlikte değerlendirilmesi gerekecektir. Şekil 4.2'de bu şekilde bir atama durumunda, ardışık günlerin alt problemlerinin birlikte değerlendirilmemesinin çözüme etkisi bir örnek ile gösterilmektedir. Burada ardışık günler Şekil 4.2 (a)'da verilen çözümde ayrı değerlendirilmiş, öncelikle ilk gün için fazla mesai değerini en küçükleyen sıra belirlenip ardından bu sıranın en sonuna çizelgelenen işe göre ikinci gün için fazla mesai değeri en küçüklenmiştir. Şekil 4.2 (b)'de verilen şekilde çizelge ise toplam fazla mesai en küçüklenecek şekilde oluşturulmuştur. Bu anlamda ayrı değerlendirildiğinde ilk gün yapılan fazla mesai 181 değerini almıştır. Bu değer, günler birlikte değerlendirildiğinde yapılan fazla mesaiye göre daha az olmasına rağmen, iki günün toplam fazla mesai değerlerine bakıldığında ilk durumda 364, ikincisinde ise 334 birim fazla mesai yapılmıştır.



Şekil 4.2: Ardışık günlerin (a) ayrı (b) birlikte değerlendirilmesi durumları

Bir başka ifadeyle, başına parçalı üretim atanan günden sonuna parçalı üretim atanan ilk güne kadar aradaki bütün günlerin çizelgeleri birbirine bağımlı olacaktır. Alt problemlerin birbirinden bağımsız değerlendirilebilmesini sağlamak için, gün sonlarına çizelgelenen işlerin bilinmesi gerekmektedir. Bu sebeple, Kısıt (4.12) ile her günün sonuna bir iş atanması zorunluluğu tanımlanmıştır. Bu kısıt parçalı üretim atama değişkeni ile tanımlanmıştır. Fakat Kısıt (4.8) ve (4.13) birlikte değerlendirildiğinde, gün sonuna atanan parçalı üretim miktarının toplam işlem zamanına eşit ($x_{ji2} = P_j$) ve devamında gün başına atanan miktarın da sıfır ($x_{j(i+1)1} = 0$) olabileceği belirtilmektedir. Bu durumda gün sonuna atanan parçalı iş aslında atanan gün içerisinde tamamlanmakta ve ertesi gün başına atanan parça bir yapay değişken görevi görmektedir. Bir başka ifade ile Kısıt (4.12) bir sıralama kararı gibi, gün içerisinde tamamlanan işin pozisyonunun gün sonuna sabitlenmesini sağlamaktadır. Bu sayede alt problemler birbirinden bağımsız değerlendirilebilmektedir. Gün sonuna yapay değişken olarak parçalı üretim atanmasına izin verilmemektedir, çünkü problem tanımına göre ayarı yapılarak üretime hazırlanan bir işin akabinde üretime başlaması gerekmektedir.

Teslim tarihine ilişkin kısıtlar için Kısıt (4.14) tanımlanmıştır. Ana problem sıralama kararı içermediği için işlerin tamamlanma zamanları tam olarak belirlenememektedir. Fakat işlerin teslim tarihleri gün sonlarında tanımlandığı için, teslim tarihine ilişkin kısıtlar atama değişkenleri (y_{jik}) üzerinden tanımlanabilmektedir. Bu anlamda (4.14) ile başlanan gün içerisinde tamamlanmak ve parçalı şekilde gün sonunda üretilmek üzere atanan işler için teslim tarihine uygunluk sağlanmaktadır. Parçalı üretimle gün başına atanan yapay işler nedeniyle bu atamalar için teslim tarihine ilişkin bir kısıt tanımlanamamaktadır. Son olarak da Kısıt (4.15) ve (4.16) ile günlük fazla mesai miktarı, gün içerisine atanan işlerin toplam işlem zamanlarının günlük mesaiyi aşan miktarı olarak hesaplanmaktadır.

4.1.3 Alt problem

Ana problemde yapılan atamalar doğrultusunda esas haftalık çizelgeleme problemi, günlük çizelgeleme problemlerine ayrıştırılmaktadır. Bu günlük çizelgeleme problemleri, alt problemler olarak tanımlanmaktadır. Ana problemden elde edilen çözümler, alt problemlerde $\bar{y}_{ji}$, $\bar{y}_{jik}$, $\bar{x}_{ji}$ ve $\bar{x}_{jik}$ ile belirtilmektedir. Alt problemler kısıt programlama modeli olarak ifade edilmiştir.

Mantıksal ifadeler içerebilen bir modelleme yaklaşımı olan kısıt programlamayla pek çok kombinatoryal optimizasyon problemi, özellikle de çizelgeleme problemleri başarıyla çözülebilmektedir [32]. Smith vd. [33] ve Darby-Dowman vd. [34] çalışmalarında değerlendirdikleri çizelgeleme problemleri için, kısıt programlama yaklaşımıyla, tam sayılı programlama yaklaşımına göre çok daha kısa sürede çözüm elde edilebildiğini göstermişlerdir. Jain ve Grossman [35] ise, bir makine çizelgeleme problemi için tam sayılı ve kısıt programlama ile tanımladıkları melez yaklaşımın gerek tam sayılı programlama gerekse kısıt programlama yaklaşımından daha iyi sonuç verdiğini belirtmişlerdir. Bu doğrultuda bu çalışma kapsamında da çizelgeleme alt problemlerinin kısıt programlama modeli olarak ifade edilmesinin ayrıştırma algoritmasının etkinliğini arttıracağı öngörülmüştür.

Bu amaçla, her gün $(i \in T)$ için bir çizelgeleme alt problemi oluşturulmaktadır. Bir i günü için tanımlanan alt problemde çizelgelenecek işler, ana problemden elde edilen çözüme göre $N_i = \{j | \bar{y}_{ji} = 1\}$ şeklinde esas problem işlerinin bir alt kümesi olarak ayrıştırılmaktadır. Burada gün başına ve sonuna parçalı üretime atanan işler sırasıyla j_1 ve j_2 ile gösterilirken, başlanan gün içerisinde tamamlanan işler ise $N_{i0} = \{j | \bar{y}_{ji0} = 1\}$ kümesiyle gösterilmektedir. Kısıt programlama modelinde bu işler aralık değişkenleri olarak ifade edilmektedir. Alt problemlerin ifadesinde kullanılan notasyon ve problem aşağıda sunulmuştur.

Karar Değişkenleri

S_j	=	j işinin üretime başlama zamanı	$j \in N$
C_j	=	j işinin üretiminin tamamlanma zamanı	$j \in N$
C_{max}^{i}	=	i günün ilgili alt probleme göre yayılma zamanı	$i \in T$

(SP1) Min C_{max}^{i}

Öyle ki

$$(i-1) \cdot ds \leq S_j \quad \forall j \in N_i \tag{4.19}$$

$$noOverlap(S_j, P_j, A_{jk}) \tag{4.20}$$

$$C_{max}^{i} = max_{j \in N_i}(C_j) \tag{4.21}$$

SP1 olarak tanımlanan alt problemler için amaç, yayılma zamanını en küçüklemektir. Burada bir günlük problem değerlendirildiği için, yayılma zamanını en küçüklemek fazla mesaiyi en küçüklemeye eşdeğerdir. Ana problemde yapılan atamaların alt problemlerin tamamında olurlu bir çizelge belirtmesi durumunda, esas problem için bir üst sınır elde edilmektedir. Bu üst sınır değeri ilgili atamalar için tanımlanan alt problemlerden elde edilen fazla mesai değerlerinin toplamı olarak hesaplanmaktadır.

Kısıt (4.19) ile çizelgeleme yapılan günün başlangıcı gösterilmektedir. Bu kısıda göre işlerin mümkün olan en erken başlama zamanı tanımlanmaktadır. Çizelgeleme kısıtları ise *noOverlap* kısıdı ile sağlanmaktadır. Kısıt (4.20) ile gösterilen bu kısıt, ardışık çizelgelenen iki aralık değişkeninin başlangıç zamanları arasında en az öncül işin işlem zamanı (P_j) ile üretim geçişleri arasında yapılması gereken ayar zamanı (A_{jk}) kadar zaman olması gerektiğini belirtmektedir. Bir başka ifade ile *noOverlap* ifadesi çizelgede yer alan iki işin birbiriyle çakışmamasını sağlanmaktadır. Günlük yayılma zamanı ise Kısıt (4.21) ile en son çizelgelenen işin tamamlanma zamanı olarak hesaplanmaktadır.

Alt problemlerin çözümünden elde edilen değerler $\bar{S}_j$, $\bar{C}_j$ ve $\bar{C}_{max}^{i}$ ile gösterilecektir. Bununla birlikte alt problemin amaç fonksiyonu değeri $\bar{C}_{max}^{i}$ doğrultusunda tanımlanan yayılma zamanının günü aşan miktarı (θ_i), aşağıda belirtilen şekilde hesaplanmaktadır.

$$\theta_i = \bar{C}^i_{max} - ds$$

Bu değerlere göre olurluluk ve optimallik kesileri aşağıda verilen şekilde tanımlanmıştır.

4.1.4 Olurluluk kesileri

Durum 1 : $\boldsymbol{\bar{C}_{j_1} > D_{j_1}}$

Ana problemde atamalar sıralama kararı ile birlikte yapılmadığı için, bu aşamada üretim tamamlanma zamanları belirlenememektedir. Bu nedenle atama değişkenleri üzerinden tanımlanan teslim tarihi kısıtlarında, parçalı üretim ile gün başına atanan işlerin teslim tarihine uygunlukları değerlendirilememektedir. Bu doğrultuda yapılan bu atamanın teslim tarihi anlamında olurluluğu kontrol edilmelidir.

Gün başına yapılan parçalı üretim ataması neticesinde olurlu bir çizelge oluşturulamaması durumunda, Kısıt (4.22) ile gösterilen olurluluk kesisi ana probleme eklenerek gevşetilmiş problem sıkılaştırılmaktadır.

$$x_{j_1\hat{\imath}} \leq 0 \qquad \forall\, \hat{\imath} \in T : \hat{\imath} \geq i \tag{4.22}$$

Bu kesi ile ana problemde parçalı üretimle $\hat{\imath}$ gününün başına atanan işin, ilgili $\hat{\imath}$ gününe ve bu $\hat{\imath}$ gününden sonraki günlere atandığı çözümler olurlu çözüm kümesinden çıkarılmaktadır. Burada olurluluk atama ikili değişkeni (y_{ji}) üzerinden değil, sürekli değişkeni (x_{ji}) üzerinden tanımlanmaktadır, çünkü burada atanmadan ziyade tamamlanma zamanını kısıtlamak amaçlanmaktadır. Bu sayede, parçalı üretim ile gün başına atanmasına rağmen ($y_{ji} = 1$) üretimi önceki günün sonunda tamamlanan ($x_{ji} = 0$) yapay atamaların yapılmasına izin verilebilmektedir.

Durum 2 : $\boldsymbol{\theta_i > 0}$

Ana problemin sıralama kararlarını içermemesi nedeniyle atamalar yapılırken sıra bağımlı ayar zamanları değerlendirilememektedir. Bu doğrultuda ana problemde toplam işlem zamanları bir günü aşmayacak şekilde yapılan atamalar, alt problemde ayar zamanlarıyla çizelgelendiğinde yayılma zamanı bir günü aşabilir. Bu durum ana probleme göre teslim tarihleri anlamında olurlu bir atamanın, esas problem için olursuz olacağı anlamına gelmektedir. Bu nedenle alt problemlerde üretim tamamlanma zamanları kontrol edilmelidir.

Yayılma zamanının günü aşması durumunda, ilgili atamaları içeren çözüm bir olurluk kesisi ile gevşetilmiş problemin olurlu çözüm kümesinden çıkarılmaktadır. Burada tanımlanan olurluluk kesileri, Çoban ve Hooker [31] çalışmalarında tanımladıkları olurluk kesileri temel alınarak oluşturulmuştur. Tanımlana kesiler atamaların durumuna ve bu durumlar için θ_i değerine göre şekillenmektedir. Aşağıdaki alt durumlar ortaya çıkmaktadır:

Durum 2.1 : $\bar{y}_{ji1}$ = 1 ve $\bar{y}_{ji2}$ = 0

Sadece gün başına parçalı üretim atanması durumu çizelgeleme periyodunun sonunda, yani haftanın son gününde görülmektedir. Bu durum için tanımlanacak olurluluk kesileri gün başına atanan parçalı üretim miktarının ($\bar{x}_{ij1}$) günü aşan miktarla (θ_i) ilişkisine göre değişmektedir. Bu durum için aşağıdaki alt durumlar ortaya çıkmaktadır.

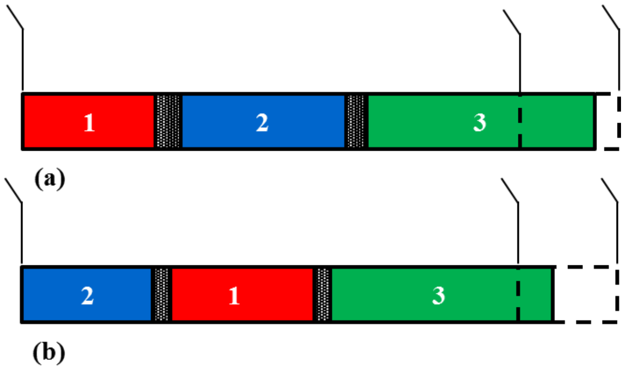
Durum 2.1.1 : $\theta_i < \bar{x}_{ij_1 1}$

Günü aşan miktarın güne atanan parçalı üretim miktarından az olması durumunda olurluluk, ancak mevcut atama durumlarında (y_{jik}) veya parçalı atama miktarında (x_{ji}) yapılacak değişikliklerle sağlanabilir. Bu durumda olurlu bir çizelge için ya Kısıt (4.23)'de yer alan atama durumlarından en az birinin değiştirilmesi ya da Kısıt (4.24)'de yer alan parçalı üretime atanan miktarın azaltılması gerekmektedir. Burada φ_h ilgili iki kısıttan bir tanesinin sağlanması için tanımlanan ikili değişkeni ifade etmektedir. Bu anlamda bu değişkenin kullanıldığı kısıtları birbirinden ayırmak adına iterasyon sayısını belirten h indisi kullanılmaktadır.

$$\varphi_h + \left(1 - y_{j_1 i1}\right) + \sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \ \geq 1 \tag{4.23}$$

$$x_{j_1 i1} \ \leq (\bar{x}_{j_1 i1} - \theta_i) \ + \ P_{j_1}(1 - \varphi_h) \tag{4.24}$$

Kısıt (4.23)'te atama durumu değişiklikleri için tanımlanan kısıt, doğrudan atama ikili değişkeni (y_{ji}) üzerinden değil, atama durumu ikili değişkeni (y_{jik}) üzerinden tanımlanmıştır. Bunun sebebi sıra bağımlı ayar zamanları nedeniyle aynı atamaların farklı durumlarda yapılmasıyla yayılma zamanının değişebilmesidir. Bu duruma bir örnek Şekil 4.3'te verilmiştir. Burada Şekil 4.3 (a)'da atanan işler, Şekil 4.3 (b)'de farklı atama durumları için atandığında yayılma zamanı değeri daha düşük bir çizelge ortaya çıkmıştır.



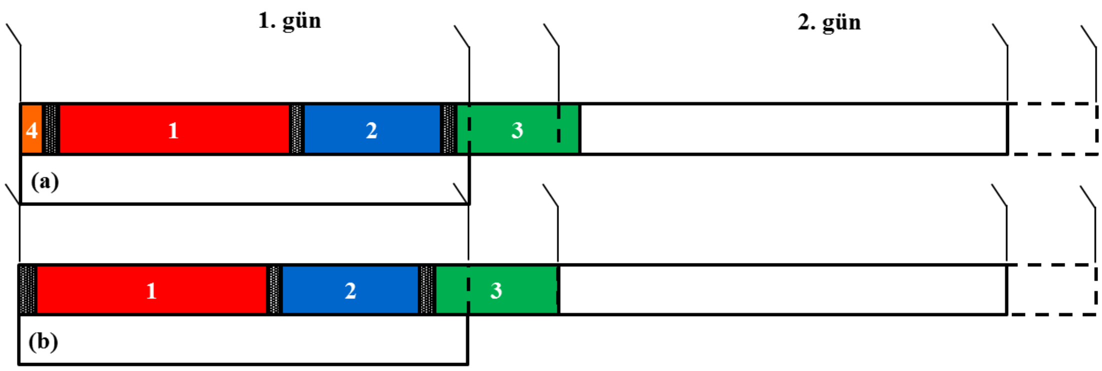
Şekil 4.3: Atama durumlarının yayılma zamanına etkisi

Burada atanan parçalı üretim miktarı, üretimi aşan miktardan büyük olduğu için üretim miktarı (x_{ji}) üzerinde yapılan bir değişiklikle de olurlu bir çizelge tanımlanabilmektedir. Bu doğrultuda tanımlanan Kısıt (4.24) ile basit bir ifadeyle, mevcut olursuzluğu engellemek için parçalı üretim miktarının en az üretimi aşan miktar kadar azaltılması gerektiği ifade edilmektedir.

Durum 2.1.2 : $\boldsymbol{\theta_i = \bar{x}_{ij_1 1}}$

Günü aşan miktarın güne atanan parçalı üretim miktarına eşit olması durumunda, Durum 2.1.1 ile benzer şekilde olurluluk, mevcut atama durumlarında veya atama miktarlarında yapılacak değişikliklerle sağlanabilir. Bu anlamda Durum 2.1.1 ile aynı olurluluk kesileri tanımlanmıştır.

Burada farklı olarak atama miktarında yapılacak değişiklik için tanımlanan Kısıt (4.24) ile gün başına atanan parçalı üretim miktarının günü aşan miktar kadar azaltılması, ilgili üretim miktarının sıfırlanması demektir. Burada gün başına sıfır üretim miktarı ile yapay üretim atanabileceği için, üretim miktarının sıfırlanması ve atamanın tamamen kaldırılması farklı durumları ifade etmektedir. Şekil 4.4 (a) ile ilgili duruma örnek bir olursuz çizelge ve Şekil 4.4 (b)'de Kısıt (4.24) ile sağlanan değişiklik ile tanımlanan olurlu çizelge verilmiştir.



Şekil 4.4: Durum 2.1.2 için (a) olursuz çizelge (b) Kısıt (4.24) doğrultusunda oluşturulan olurlu çizelge

Durum 2.1.3 : $\boldsymbol{\theta_i > \bar{x}_{ij_1 1}}$

Günü aşan miktarın güne atanan parçalı üretim miktarından fazla olması durumunda, parçalı üretim miktarının azaltılması hatta sıfırlanması ile olurlu bir çizelge tanımlanması mümkün değildir. Fakat parçalı atamanın kaldırılması durumunda yayılma zamanı, bu atamanın üretim miktarı ve bu atamaya bağlı yapılan ayar zamanı kadar azalacaktır. Bu durumda parçalı atamanın tamamen kaldırılması durumunda olurlu bir çizelge tanımlanmasını sağlayacak bir azalma olabilir.

Burada yayılma zamanında meydana gelebilecek azalmanın, doğrudan parçalı üretim ve ilgili ayar zamanının toplamıyla karşılaştırılması yeterli olmayacaktır, çünkü gün başına ve sonuna sabitlenen işler değiştirildiğinde sıra bağımlı ayar zamanları nedeniyle SP_1 ile belirlenen çizelge değişebilmektedir. Bu doğrultuda SP_2 olarak tanımlanan olurluluk alt problemi ile parçalı üretim atamasının kaldırılması ile olurluluğun sağlanıp sağlanmayacağı ayrıca kontrol edilmektedir.

(SP_2)

$$(i-1)\cdot ds \leq S_j \qquad \forall j \in N_i - \{j_1\} \qquad (4.25)$$

$$C_j \leq i \cdot ds \qquad \forall j \in N_i - \{j_1\} \qquad (4.26)$$

$$C_j \leq D_j \qquad \forall j \in N_i - \{j_1\} \qquad (4.27)$$

$$noOverlap(S_j, P_j, A_{jk}) \qquad (4.28)$$

Kısıt (4.25), SP_1 için tanımlanan Kısıt (4.19) ile aynıdır. Kısıt (4.26) ise benzer şekilde, gün sonunu ve buna göre işlerin mümkün olan en geç tamamlanma zamanını belirtmektedir. Teslim tarihi Kısıt (4.27) ile kontrol edilmektedir. Ana problemde, sadece gün başına atanan parçalı üretimlerin teslim tarihine uygunluğu kontrol edilmemektedir. Dolayısıyla üretimine başlanan gün içerisinde tamamlanmak üzere atanan ($k = 0$) işlerin tamamlanma zamanları günü aşmadığı sürece, ilgili atamalar teslim tarihleri anlamında olurlu olacaktır. Bu anlamda Kısıt (4.26) ve (4.27) ile parçalı atamanın kaldırılması durumunda olurluluğun sağlanıp sağlanmayacağı belirlenmektedir.

SP_2 olursuz ise, parçalı üretimin kaldırılması durumunda mevcut atamalarla esas problemin olurluluğu sağlanamayacak demektir. Bu durumda olurluk kesisi Kısıt (4.29) ile, diğer atama durumlarının en az birinde değişiklik yapılması şeklinde tanımlanmaktadır. Parçalı üretimin kaldırılması ile SP_2 olurlu olursa ana problemin olurluluğu Kısıt (4.30) ile sağlanmaktadır.

$$\sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \geq 1 \tag{4.29}$$

$$\left(1 - y_{j_1 i1}\right) + \sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \geq 1 \tag{4.30}$$

Durum 2.2 : $\bar{y}_{ij1}$ = 0 ve $\bar{y}_{ij2}$ = 1

Gün başına parçalı atama yapılmaksızın sadece gün sonuna parçalı üretim atanması durumu çizelgeleme periyodunun başında, yani haftanın ilk gününde görülmektedir. Durum 2.1 ile benzer şekilde, bu durumda tanımlanan kesiler de gün sonuna atanan parçalı üretim miktarı ($\bar{x}_{ij2}$) ve günü aşan miktara (θ_i) göre şekillenmektedir. Buna bağlı olarak aşağıdaki alt durumlar ortaya çıkmaktadır.

Durum 2.2.1 : $\theta_i < \bar{x}_{ij_2 2}$

Durum 2.1.1 ile benzer şekilde bu durumda da olurlu bir çizelge tanımlamak için mevcut atama durumlarında veya parçalı atama miktarında değişiklik yapılması gerekmektedir. Bu doğrultuda atama durumlarında yapılacak değişiklikler için (4.31), atama miktarında yapılacak değişiklik için ise (4.32) ile belirtilen kesiler oluşturulmaktadır.

$$\varphi_h + \left(1 - y_{j_2 i2}\right) + \sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \geq 1 \tag{4.31}$$

$$x_{j_2 i2} \leq (\bar{x}_{j_2 i2} - \theta_i) + P_{j_2}(1 - \varphi_h) \tag{4.32}$$

Durum 2.2.2 : $\boldsymbol{\theta_i = \bar{x}_{ij_2 2}}$

Bu durumda Durum 2.1.2 ile farklı olarak, olurluluğun sağlanması sadece atama durumları üzerinde yapılacak değişiklikler ile mümkündür. Burada parçalı üretim miktarının günü aşan miktar kadar azaltılması için ilgili atamanın tamamen kaldırılması gerekmektedir, çünkü gün sonuna üretim miktarı sıfır olan yapay atamalar yapılamamaktadır. Bu doğrultuda olurluluk için sadece Kısıt (4.33) tanımlanmaktadır.

$$\left(1 - y_{j_2 i2}\right) + \sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \geq 1 \tag{4.33}$$

Durum 2.2.3 : $\boldsymbol{\theta_i > \bar{x}_{ij2}}$

Bu durumda Durum 2.1.2 ile benzer şekilde kesiler, gün sonuna atanan parçalı üretimin kaldırılmasının olurluluğa etkisine göre belirlenmektedir. Bu doğrultuda parçalı atamanın kaldırılmasıyla olurluluk problemi SP_3 tanımlanmaktadır.

(SP_3)

$$(i-1) \cdot ds \leq S_j \qquad \forall j \in N_i - \{j_2\} \tag{4.34}$$

$$S_j \leq i \cdot ds \qquad \forall j \in N_i - \{j_2\} \tag{4.35}$$

$$C_j \leq T_j \qquad \forall j \in N_i - \{j_2\} \tag{4.36}$$

$$noOverlap(S_j, P_j, A_{jk}) \tag{4.37}$$

Parçalı üretimin kaldırılması ile olurlu bir çizelge oluşturulamaması durumunda Kısıt (4.29) ile diğer atamalardan en az birinin değiştirilmesi sağlanmaktadır. SP_3 olurlu ise de, ana problemin olurluluğu için Kısıt (4.38) ile mevcut atamaların herhangi birinde değişiklik yapılmaktadır.

$$\left(1 - y_{j_2 i2}\right) + \sum_{j \in N_{i0}} \left(1 - y_{ji0}\right) \geq 1 \tag{4.38}$$

Durum 2.3 : $\boldsymbol{\bar{y}_{ij1}}$ = 1 ve $\boldsymbol{\bar{y}_{ij2}}$ = 1

Çizelgeleme periyodunun başı ve sonu dışında, bütün günlerin başına ve sonuna parçalı üretim atanmaktadır. Bu durumda olurluluk kesileri gün başına ve sonuna atanan parçalı üretim miktarlarının toplamının ($\bar{x}_{ij1} + \bar{x}_{ij2}$) günü aşan miktar ($\theta_i$) ile ilişkisi değerlendirilerek tanımlanmaktadır. Aşağıdaki alt durumlar ortaya çıkmaktadır.

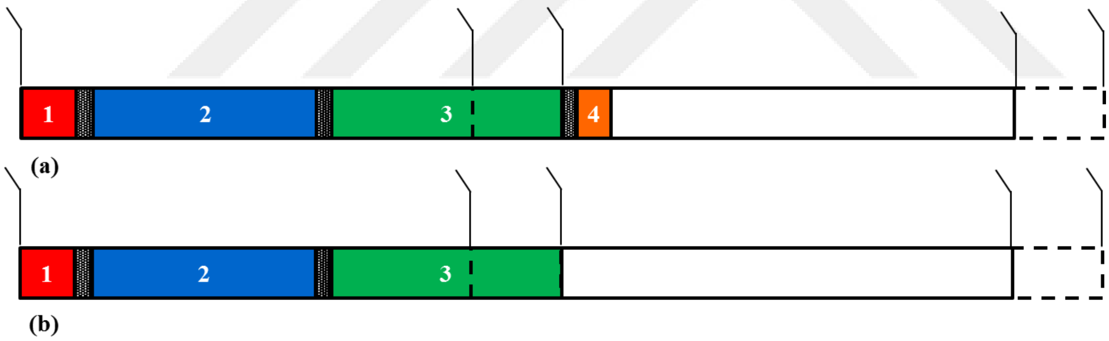
Durum 2.3.1 : $\boldsymbol{\theta_i < \bar{x}_{ij_1 1} + \bar{x}_{ij_2 2}}$

Bu durumda olurluluk kesileri Durum 2.1.1 ve 2.2.1 ile aynı doğrultuda tanımlanmaktadır. Kısıt (4.39) mevcut atama durumları, Kısıt (4.40) ise atama miktarları için tanımlanan kesileri ifade etmektedir.

$$\varphi_h + (1 - y_{j_1 i1}) + (1 - y_{j_2 i2}) + \sum_{j \in N_{i0}} (1 - y_{ji0}) \geq 1 \tag{4.39}$$

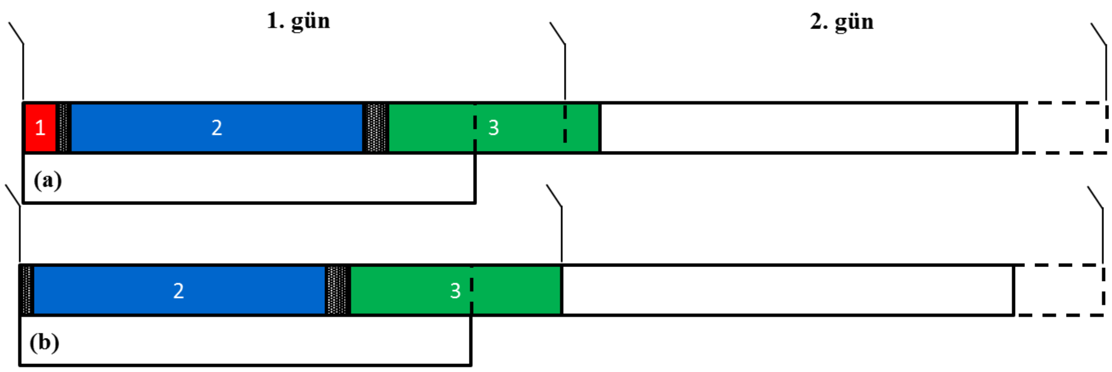
$$x_{j_1 i1} + x_{j_2 i2} \leq (\bar{x}_{j_1 i1} + \bar{x}_{j_2 i2} - \theta_i) + (P_{j_1} + P_{j_2})(1 - \varphi_h) \tag{4.40}$$

Bu anlamda olurlu bir çizelge tanımlamak için (4.39) ile mevcut parçalı atama durumlarından sadece birinin değiştirilmesi yeterli olabilmektedir. Bu durum için bir örnek Şekil 4.5'de gösterilmektedir. Burada Şekil 4.5 (a) ile verilen olursuz çizelgede gün sonuna atanan parçalı üretimin Kısıt (4.39) doğrultusunda kaldırılması ile Şekil 4.5 (b) ile verilen çizelge elde edilmektedir.



Şekil 4.5: Durum 2.3.1 için (a) olursuz çizelge (b) Kısıt (4.39) doğrultusunda oluşturulan çizelge

Benzer şekilde, Şekil 4.6 ile gösterilen örnekte Kısıt (4.40) ile sadece bir parçalı atamanın üretim miktarının azaltması ile olurlu bir çizelge tanımlanabilmektedir. Bu örnekte Şekil 4.6 (a) ile verilen olursuz çizelge, gün başına atanan parçalı üretim miktarının Kısıt (4.40) ile sıfırlanmasıyla Şekil 4.6 (a) ile verilen çizelgeye dönüşmektedir.



Şekil 4.6: Durum 2.3.1 için (a) olursuz çizelge (b) Kısıt (4.40) doğrultusunda oluşturulan olurlu çizelge

Durum 2.3.2 : $\boldsymbol{\theta_i = \bar{x}_{ij_1 1} + \bar{x}_{ij_2 2}}$

Üretimi aşan miktarın toplam parçalı üretime eşit olması durumunda sıra bağımlı ayar zamanları nedeniyle olurluluk sadece tek bir parçalı atamanın kaldırılmasıyla da sağlanabilmektedir. Burada sadece gün başına parçalı atama yapılan olursuz bir çizelgede ilgili üretim miktarının günü aşan miktara eşit olması durumu gibi, tek başına parçalı atama miktarının sıfırlanmasıyla olurlu bir çizelge tanımlanmayacaktır. Benzer şekilde sadece gün sonuna atanan üretim miktarı üzerinde yapılan bir değişiklik de olursuzluğu ortadan kaldırmayacaktır. Bu doğrultuda üretim miktarları anlamında, iki parçalı üretimin de birlikte sıfırlanması gerekmektedir. Bu durum gün sonuna atanan işin kaldırılması anlamına gelirken, gün başına atanan işin ise sıfırlanması veya kaldırılması anlamına gelmektedir. Bu anlamda parçalı atamalarda yapılacak bir değişiklik, kesinlikle ilgili atamalardan en az birinin kaldırılmasını içermektedir. Bununla birlikte başlanan gün içerisinde tamamlanan işlerden en az birinin kaldırılması da olurlu bir çizelge oluşturulmasını sağlayabilir. Bu doğrultuda bu durum için olurluluk Kısıt (4.41) ile sağlanmaktadır.

$$(1 - y_{j_1 i1}) + (1 - y_{j_2 i2}) + \sum_{j \in N_{i0}} (1 - y_{ji0}) \geq 1 \qquad (4.41)$$

Durum 2.3.3 : $\boldsymbol{\theta_i > \bar{x}_{ij_1 1} + \bar{x}_{ij_2 2}}$

Bu durumda Durum 2.1.3 ve Durum 2.2.3 ile benzer şekilde, parçalı atamaların kaldırılması durumunda olurlu bir çizelge tanımlanıp tanımlanamayacağı kontrol edilmektedir.

(SP_4)

$$(i-1)\cdot ds \leq S_j \qquad \forall j \in N_{i0} \qquad (4.42)$$

$$S_j \leq i \cdot ds \qquad \forall j \in N_{i0} \qquad (4.43)$$

$$C_j \leq T_j \qquad \forall j \in N_{i0} \qquad (4.44)$$

$$noOverlap(S_j, P_j, A_{jk}) \qquad (4.45)$$

Bu doğrultuda tanımlanan problem SP_4 olursuz ise, olurlu bir çizelge için Kısıt (4.29) ile diğer atamalardan en az birinden değişiklik yapılması sağlanmaktadır. Parçalı üretimlerin kaldırılması ile SP_4 çözümünden olurlu bir çizelge edilmesi durumunda ise, Durum 2.3.2 ile aynı şekilde (4.41) tanımlanmaktadır.

4.1.5 Optimallik kesileri

Ana problem, esas problemin bir gevşetmesi olduğu için, bu problemin amaç fonksiyonu değeri esas problem için bir alt sınır değeri belirtmektedir. Bununla birlikte ana problemde yapılan atamaların esas problem için olurlu bir çizelge belirtmesi durumunda, ilgili atamalara göre tanımlanan alt problemlerden $\sum_{i \in T} \max(0, \bar{C}_{max}^{i} - ns)$ şeklinde, bir üst sınır değeri tanımlamaktadır. Fakat üst ve alt sınır değerleri birbirine eşit değilken, sınır tanımlayan olurlu çözümün optimallik durumu hakkında yorum yapılamamaktadır. Bu doğrultuda tanımlanan kesiler ile optimal çözümü belirlemek hedeflenmektedir. Burada tanımlanan kesiler Tran vd. [30] çalışmasında tanımlanan kesiler temel alınarak oluşturulmuştur.

Optimallik kesileri atamaların durumuna göre şekillenmektedir. Fakat genel anlamda, ana problemle i gününe yapılan atamalara göre, ilgili günün yayılma zamanı için bir sınır tanımlamaktadırlar. Bu sınır değeri ilgili atamalar için tanımlanabilecek en küçük yayılma zamanı değeri olarak, alt problemin çözümünden elde edilen yayılma zamanı değeri $\bar{C}_{max}^{i}$ ile belirlenmektedir. Bu anlamda, gevşetilmiş ana problemde değerlendirilmeyen sıra bağımlı ayar zamanlarının yayılma zamanına eklenmesiyle, ana problemden elde edilen alt sınır değeri sıkılaştırılmaktadır. Bununla birlikte optimallik kesisi ilgili i gününe yapılan atamalarda yapılabilecek değişikliklerin $\bar{C}_{max}^{i}$ değerine etkisini belirtmektedir. Bu sayede ana problemle bu etkilere göre farklı atamaların denenmesi ve üst sınır değerinin iyileştirilmesi amaçlanmaktadır.

Her bir değişikliğin $\bar{C}^{i}_{max}$ değerine kesin etkisinin tam olarak hesaplanması ve kesi ile ifade edilmesi güç olacağı için, değişikliklerin etkisi yaratabilecekleri en büyük etki üzerinden yaklaşık bir şekilde tanımlanmaktadır. Burada tanımın en büyük etki üzerinden yapılmasıyla kesinin optimal çözümü kesmesini engellemek amaçlanmaktadır. Aksi takdirde kesi yeterince gevşek tanımlanmadıysa, belirlenen alt sınır değeri optimal değerden yüksek çıkabilir.

En büyük etkinin değeri hesaplanırken, ayar zamanları anlamında yaratılabilecek en büyük etkinin hesaplanması gerekmektedir. Bu amaçla tanımlanan $maksOn_j$, i gününe atanan bir j işinin, kendisiyle aynı güne atanan diğer işlerden, ardılında çizelgelenmesi durumunda en büyük sıra bağımlı ayar zamanına katlanmasına neden olacak k işiyle arasında yapılan ilgili ayar zamanını belirtmektedir. Benzer şekilde $maksArd_j$ ise, i gününe atanan bir j işinin diğer işlerden öncülüne atanması durumunda katlanılacak en büyük sıra bağımlı ayar zamanını ifade etmektedir. Bu iki ayar zamanı sırasıyla aşağıda belirtilen şekilde hesaplanmaktadır.

$$maksOn_j = \max_{k \in N_i}(A_{jk})$$

$$maksArd_j = \max_{k \in N_i}(A_{kj})$$

Bu doğrultuda başlanan gün içerisinde tamamlanmak üzere atanan bir işin ilgili günün çizelgesinden kaldırılması durumunda değişikliğin yaratabileceği en büyük etki, ilgili işin işlem ve ayar zamanı etkilerinin toplanması ile elde edilmektedir. Bu atama durumunda çizelgelenen bir işin hem öncülüne hem de ardılına başka işler çizelgelendiği için, ilgili işin kaldırılması durumunda ayar zamanı anlamında en fazla ilgili iki ayar zamanının toplamı kadar değişiklik gözlemlenebilir. Bu doğrultuda σ_j ile ifade edilen değişikliğin en büyük etkisinin değeri (4.46) ile verilen şekilde hesaplanmaktadır.

$$\sigma_j = P_j + maksOn_j + maksArd_j \qquad (4.46)$$

Burada sadece mevcut atama durumun kaldırılmasında meydana gelebilecek değişiklik değerlendirilmektedir, çünkü farklı bir duruma atanması durumunda oluşabilecek değişikliklerin etkilerinin değerlendirilebilmesi için ilgili durumlar için yayılma zamanı değerlerinin hesaplanması yani alt problemlerin tekrar çözdürülmesi gerekmektedir. Bu noktada tanımlanan kesinin kuvveti ile hesaplama yükü arasında bir ödünleşim bulunmaktadır.

Gün başına parçalı üretime atanan bir işte yapılan değişikliğin yaratabileceği etki ise, üretim miktarı üzerinde yapılan değişiklikle ayar zamanına bağlı gözlemlenecek etkinin toplamı olarak belirlenmektedir. Bu duruma atanan bir işin öncülü bir iş bulunmadığı için ayar zamanı anlamında gözlemlenebilecek en büyük etki $maksOn_j$ ile belirlenmektedir. Bu doğrultuda gün başına atanan bir işte yapılan değişikliğin etkisi (4.47) ile belirtilen şekilde hesaplanmaktadır.

$$\delta_{j_1} = maksOn_{j_1}(1 - y_{j_1 i1}) + (\bar{x}_{j_1 i1} - x_{j_1 i1}) \quad (4.47)$$

Gün sonuna atanan bir işte yapılan değişikliğin etkisi de benzer şekilde hesaplanmaktadır, fakat bu durumda tam tersine ayar zamanı anlamında gözlemlenebilecek etki $maksArd_j$ üzerinden değerlendirilmektedir. Toplam değişiklik (4.48) ile verilen şekilde belirlenmektedir.

$$\delta_{j_2} = maksArd_{j_2}(1 - y_{j_2 i2}) + (\bar{x}_{j_2 i2} - x_{j_2 i2}) \quad (4.48)$$

Bu doğrultuda sadece başında ve sadece sonunda parçalı üretim olan bir gün için tanımlanan optimallik kesileri sırasıyla Kısıt (4.49) ve (4.50) ile belirtilirken, başında ve sonunda parçalı üretim olan bir gün için Kısıt (4.51) ile belirtilen kesi tanımlanmıştır.

$$C_{max}^i \geq \bar{C}_{max}^i - \delta_{j_1} - \sum_{j \in N_i} \sigma_j \cdot (1 - y_{ji0}) \quad (4.49)$$

$$C_{max}^i \geq \bar{C}_{max}^i - \sum_{j \in N_i} \sigma_j \cdot (1 - y_{ji0}) - \delta_{j_2} \quad (4.50)$$

$$C_{max}^i \geq \bar{C}_{max}^i - \delta_{j_1} - \sum_{j \in N_i} \sigma_j \cdot (1 - y_{ji0}) - \delta_{j_2} \quad (4.51)$$

(4.49) - (4.51) ile verilen optimallik kesileri sayesinde esas problem için optimal çözüme yaklaşılabileceği, her kesi için tek tek gösterilmek yerine, iki durumu birlikte ifade ederek (4.49) ve (4.50)'yi içeren Kısıt (4.51) üzerinden gösterilecektir. Tanımlanan kesinin mevcut çözümü keserken, optimal çözümü kesmeyeceği ispatlanmalıdır. Mevcut çözümü kestiğinin ispatlanması daha kolaydır: Tanımlanan kesi sayesinde ilgili ana problemin atamalarına göre bir yayılma zamanı sınırı belirlenmektedir. Bu sınır değerinin ya ilgili atamalar için ana problemde belirlenen yayılma zamanını arttırıp sıkılaştırarak, ya da bu durumdan kaçınmak adına atamalarda değişiklik yaparak ilgili çözümü kestiği açıktır. Bu esnada optimal çözümü kesilmeyeceği ise Teorem 1 ile ispatlanmıştır.

Teorem 1 Denklem (4.51) ile belirlenen optimallik kesisi herhangi bir optimal çözümü kesmemektedir.

İspat: Ana problem ile i gününe atanan N_i işler kümesi için tanımlanan optimallik kesisini ele alalım. İlgili alt problemin çözümüne göre N_i için optimal yayılma zamanı değeri $\bar{C}^i_{max}$ olarak gösterilsin. Bununla birlikte N_i işler kümesinde gün başına (j_1) ve sonuna (j_2) atanan işlerin atama durumları ve miktarları sırasıyla $\bar{y}_{j_k ik}$ ve $\bar{x}_{j_k ik}$ $(k = 1, 2)$ ile gösterilsin. Bu çözüm için tanımlanan kesinin, N_i^* ile gösterilen işler kümesinin atandığı optimal çözümü kestiğini, yani bu çözümü dışladığını varsayalım. Bu optimal çözümün yayılma zamanı değeri $\left(C^i_{max}\right)^*$ ile gösterilsin. Bu doğrultuda üretimine başlanan gün içerisinde tamamlanmak üzere N_i kümesine atanan, fakat N_i^* kümesine atanmayan işlerin kümesini $\breve{N}_{i0}$ ile gösterelim. N_i işler kümesinde j_1 ve j_2 ile gösterilen işlerin N_i^* için atama durumlarını ve miktarlarını sırasıyla $y^*_{j_k ik}$ ve $x^*_{j_k ik}$ ile gösterelim.

Tanımlanan kesinin optimal çözümü kesmesi, optimal çözümün kesiyi sağlamaması anlamına gelmektedir. Bu da optimal çözüme ait yayılma zamanı değerinin, aşağıda (4.52) ile ifade edilen ilişkiyi sağlaması anlamına gelmektedir. Bu denklemde δ_{j_1} ve δ_{j_2} ifadeleri sırayla (4.47) ve (4.48) numaralı denklemlerde $y_{j_k ik} = y^*_{j_k ik}$ ve $x_{j_k ik} = x^*_{j_k ik}$ $(k = 1, 2)$ şeklinde tanımlanmasıyla elde edilmektedir.

$$\left(C^i_{max}\right)^* < \bar{C}^i_{max} - \delta_{j_1} - \sum_{j \in \breve{N}_{i0}} \sigma_j \cdot \left(1 - y_{ji0}\right) - \delta_{j_2} \qquad (4.52)$$

İlgili optimal çözüme atanan işlerinin kümesini, mevcut çözümle kesişiminin elemanlarına indirgeyelim. Bu küme $\widehat{N}_i = N_i \cap N_i^*$ olarak tanımlansın. N_i ve N_i^* kümelerinde gün başına veya sonuna atanan işler de ortak ise, ilgili atamaların miktarını optimal çözüme atanan miktar olarak kullanalım. İndirgenen kümeyle yapılan atamaların, optimal çözümde belirlenen sırada çizelgelenmesi ile elde edilen çizelgenin yayılma zamanı değerini $\hat{C}^i_{max}$ ile gösterelim. Sıra bağımlı ayar zamanlarının üçgen eşitsizliğini sağlaması nedeniyle (4.53) ile belirtilen ilişki sağlanacaktır.

$$\hat{C}^{i}_{max} \leq \left(C^{i}_{max}\right)^{*} < \bar{C}^{i}_{max} - \delta_{j_1} - \sum_{j \in \breve{N}_{i0}} \sigma_j \cdot \left(1 - y_{ji0}\right) - \delta_{j_2} \quad (4.53)$$

(4.53) ile belirtilen ilişkinin sağlanamayacağının gösterilmesi durumunda, (4.52) ile belirtilen ilişkinin sağlanamayacağı, yani tanımlanan kesinin engellediği bir optimal çözümün olmadığı ispatlanacaktır. Bu doğrultuda, $\widehat{N}_i$ işler kümesine indirgenen çizelgeye mevcut çizelgenin $\breve{N}_{i0}$ kümesine atanan ve mevcut çizelgede indirgenen çizelgeden farklı yapılan parçalı atamaların dahil edilmesiyle, indirgenen çizelge mevcut çizelge atamalarını içerecek şekilde genişletilecektir. $\breve{N}_{i0}$ kümesinden eklenen işler, indirgenen çizelgede belirlenen sırada başlanan gün içerisinde tamamlanacak şekilde son çizelgelenen işin ardına eklenecektir. Parçalı atamada yapılan değişiklikler ise uygun şekilde gün başında veya sonunda yapılacaktır. Yayılma zamanı $\ddot{C}^{i}_{max}$ ile ifade edilen ilgili çizelge için (4.54) ile belirtilen ilişki sağlanacaktır.

$$\begin{aligned} \ddot{C}^{i}_{max} \leq \hat{C}^{i}_{max} + \sum_{j \in \breve{N}_{i0}} \left(P_j + maksOn_j + maksArd_j\right) \quad (4.54) \\ - \left(maksOn_{j_1}\left(1 - y^{*}_{j_1 i1}\right) + \left(\bar{x}_{j_1 i1} - x^{*}_{j_1 i1}\right)\right) \\ - \left(maksArd_{j_2}\left(1 - y^{*}_{j_2 i2}\right) + \left(\bar{x}_{j_2 i2} - x^{*}_{j_2 i2}\right)\right) \end{aligned}$$

$\breve{N}_{i0}$ kümesinden eklenen bir iş $\hat{C}^{i}_{max}$ değerini ilgili işin toplam işlem zamanı (P_j) kadar arttıracaktır. Gün başına veya sonuna atanan işlerin üretim miktarı $y^{*}_{j_1 i1}$ ve $y^{*}_{j_2 i2}$ üzerinde yapılan değişikliklerin $\hat{C}^{i}_{max}$ değerine etkisi ise sırayla $\left(\bar{x}_{j_1 i1} - x^{*}_{j_1 i1}\right)$ ve $\left(\bar{x}_{j_2 i2} - x^{*}_{j_2 i2}\right)$ şeklinde hesaplanmaktadır. Bu noktada eğer optimal çizelgede atanan $x^{*}_{j_1 i1}$ ($x^{*}_{j_2 i2}$) değeri mevcut çizelgede atanan $\bar{x}_{j_1 i1}$ ($\bar{x}_{j_2 i2}$) değerinden az ise, $\hat{C}^{i}_{max}$ değeri aradaki fark kadar azalacaktır. Aksi takdirde, $\hat{C}^{i}_{max}$ değeri ilgili fark kadar artacaktır. Bununla birlikte $\breve{N}_{i0}$ kümesinden eklenen bir iş için $\hat{C}^{i}_{max}$ değerine ilgili işin hem öncülüyle hem de ardılı yapacağı ayar zamanı eklenecektir. Benzer şekilde $\hat{C}^{i}_{max}$ değeri gün başına eklenen bir iş için ardılıyla, sonuna eklenen iş için ise öncülüyle yapılacak ayar zamanı kadar artacaktır. Fakat, ilgili işlerin öncül ve ardılları bilinmemektedir. Burada $maksOn_j$ ve $maksArd_j$ ifadeleri, N_i kümesine atanan bütün işler arasında en büyük ayar zamanı değerlerini belirttiği için (4.54) ile verilen ilişki tanımlanabilmektedir.

Bu ilişki, Denklem (4.55) ile belirtilen şekilde de ifade edilebilir.

$$\begin{aligned} \hat{C}^i_{max} \geq \ddot{C}^i_{max} - \sum_{j \in \breve{N}_{i0}} \left(P_j + maksOn_j + maksArd_j\right) \\ + \Big(maksOn_{j_1}\left(1 - y^*_{j_1 i1}\right) + \left(\bar{x}_{j_1 i1} - \ x^*_{j_1 i1}\right)\Big) \\ + \ \Big(maksArd_{j_2}\left(1 - y^*_{j_2 i2}\right) + \left(\bar{x}_{j_2 i2} - \ x^*_{j_2 i2}\right)\Big) \end{aligned} \tag{4.55}$$

$\bar{C}^i_{max}$ ile belirtilen yayılma zamanının doğrudan N_i kümesine atanan işler için tanımlanması nedeniyle, Denklem (4.56) ile belirtilen ilişki de sağlanacaktır.

$$\begin{aligned} \hat{C}^i_{max} \geq \bar{C}^i_{max} - \sum_{j \in \breve{N}_{i0}} \left(P_j + maksOn_j + maksArd_j\right) \\ + \Big(maksOn_{j_1}\left(1 - y^*_{j_1 i1}\right) + \left(\bar{x}_{j_1 i1} - \ x^*_{j_1 i1}\right)\Big) \\ + \ \Big(maksArd_{j_2}\left(1 - y^*_{j_2 i2}\right) + \left(\bar{x}_{j_2 i2} - \ x^*_{j_2 i2}\right)\Big) \end{aligned} \tag{4.56}$$

(4.53) ile verilen ilişkide δ_{j_1} ve δ_{j_2} ifadeleri için yapılan tanım değerlendirildiğinde, (4.53) ve (4.56) eşitsizliklerinin sağ taraflarının birbirine eşit olduğu gözlemlenmektedir. Bu anlamda bu ilişki (4.53) ile belirtilen ifadeyle çelişmektedir. Bu nedenle bu ifadenin ve devamında (4.52) ile belirtilen ifadenin sağlanmayacağı gösterilmektedir. Bu anlamda tanımlanan kesinin optimal çözümü kesmeyeceği ispatlanmıştır. ■

Bu bölümde kesin çözüm yöntemi olarak geliştirilen mantıksal Benders ayrıştırma algoritmasının detayları sunulmuştur. Bu algoritma Java programalama dilinde IBM ILOG Concert Technology kullanılarak kodlanmıştır. Yapılan testlerde, bu yöntemin önceki bölümde geliştirilen matematiksel programlama formülasyonuna göre çözüm süresini oldukça düşürdüğü gözlemlenmiştir. Fakat, problemin karmaşıklığından dolayı, hala belirli problemler için makul sürelerde çözümler elde edilememiştir. Bu sebeple, kısa sürede kaliteli çözümler elde edebilmek için bir sonraki bölümde bir tavlama benzetimi algoritması geliştirilmiştir.

4.2 Tavlama Benzetimi

Tavlama benzetimi algoritması, istatiksel mekanik esasları temel alınarak türetilmiş bir sezgisel algoritmadır. Tavlama işleminde metaller belirli bir dereceye kadar ısıtılıp ardından yavaşça soğutularak yapıları güçlendirilmektedir. Bu işlem esnasında başlangıç sıcaklık değerinin yeterince yüksek belirlenmesi ve soğutma işleminin yeterince yavaş yapılması istenilen güçlü, kristal yapının elde edilmesinde çok önemlidir [36].

Metropolis vd. [37] tavlama işlemini modelleme adına bir algoritma tasarlamıştır. Kirkpatrick vd. [38] ve Cerny [39] bu tavlama benzetimi algoritmasının optimizasyon problemleri için uygulanabileceğini belirtmiştir. Bu yaklaşım çizelgeleme problemleri gibi pek çok kombinatoriyal optimizasyon problemine kolayca uygulanarak, etkili çözümler elde edilmesini sağlamaktadır [36]. Tan vd. [15] tek makinede sıra bağımlı ayar zamanlarıyla toplam geç kalmayı en küçükleme problemi için, tavlama benzetimi algoritmasının büyük boyutlu gerçek hayat örneklerini çözebilmek adına dal ve sınır algoritması ve genetik algoritmadan etkili bir yöntem olduğunu belirtmiştir. Park ve Kim [40] ve Radhakrishnan ve Ventura [41] ise eş paralel makinelerde sıra bağımlı ayar zamanlarıyla çizelgeme probleminin çözümü için tavlama benzetimi algoritmasından yararlanmıştır. Hazır olma ve geç kalma zamanlarına yönelik çizelgeleme amaçlarının ele alındığı çalışmalarının sonucunda, Park ve Kim [40], tavlama benzetimi algoritmasının tabu arama algoritmasından daha üstün sonuç verdiğini göstermiştir. Kim vd. [42] ise ilgili problemi alakasız paralel makineler kullanarak ele almıştır. Tavlama benzetimi ve komşu arama algoritmalarını karşılaştırdıkları çalışmada, tavlama benzetimi sezgisel yaklaşımının daha etkili bir yöntem olduğu sonucuna varılmıştır. Bu çalışma kapsamında da kesin çözüm yöntemleriyle makul süreler içerisinde çözüm almanın zorlaştığı örneklerde yararlanmak adına bir tavlama benzetimi algoritması geliştirilmiştir.

Tavlama benzetimi algoritması soğuma şemasına göre belirlenen bir başlangıç sıcaklığında, rasgele veya bir çözüm kuran algoritmayla belirlenen bir başlangıç çözümü ile başlamaktadır. Bu başlangıç çözümü üzerinden soğuma şeması ile belirlenen denge noktasına ulaşana kadar rastgele komşuluklar ile yeni çözümler türetilmektedir. Burada başlangıç çözümü ebeveyn çözüm (s), komşu çözümler ise çocuk çözüm ($\hat{s}$) olarak adlandırılmaktadır. Türetilen en iyi çocuk çözümün amaç fonksiyonu değeri, ilgili ebeveyn çözümle karşılaştırılmaktadır. Bu çocuk çözüm ebeveyn çözümden daha iyiyse, ilgili çocuk çözüm doğrudan algoritmanın diğer iterasyonunda ebeveyn çözüm olarak değerlendirilmek üzere seçilmektedir. Aksi takdirde ise çocuk çözüm belirli bir olasılıkla seçilmektedir. Bu olasılıksal seçim süreci nedeniyle tavlama benzetimi algoritması stokastik bir algoritmadır. Burada kötüleşen bir çözüm seçilerek, algoritmanın yerel optimal çözümlere takılmasını engellemek amaçlanmaktadır. Kötü bir çözümün seçilme olasılığı, kötüleşme miktarına ve tavlama sürecinin soğuma şemasına göre tanımlanan sıcaklık parametresine bağlıdır. Bu olasılık aşağıda gösterilen şekilde hesaplanmaktadır.

$$P(\Delta E, T) = e^{-\frac{f(\hat{s})-f(s)}{T}}$$

Burada $f(\hat{s}) - f(s)$ ifadesi ile belirtilen çocuk çözüm ve ebeveyn çözümün amaç fonksiyon değerlerinin farkı, yani kötüleşme miktarıdır. T sıcaklık değeri ise, ilgili iterasyonun sıcaklık değerini göstermektedir. Algoritma kapsamında bir iterasyon, ebeveyn çözümden türetilen en iyi komşu çözümün seçilme kararının alınmasına kadar olan adımlar olarak tanımlanmaktadır. Algoritma, iterasyonların sonunda soğuma şemasından belirtilen yönteme göre sıcaklık değerinin düşürülmesi ile işletilmektedir ve belirli bir sıcaklık değerine ulaştığında sonlandırılmaktadır. Algoritmanın genel adımları aşağıda verilen Algoritma 1 ile özetlenmektedir

Algoritma 1 Tavlama Benzetimi Algoritması

Girdi: Soğuma şeması, Başlangıç çözümü (s_0)
$T \rightarrow T_{maks}$, $s \rightarrow s_0$
$T > T_{min}$ olduğu sürece tekrarla
Denge noktasına kadar tekrarla
Rasgele komşu çözüm ($\hat{s}$) türet
    $\Delta E = f(\hat{s}) - f(s)$
        Eğer $\Delta E \leq 0$ ise Komşu çözümü ebeveyn olarak belirle ($s = \hat{s}$)
        Değilse, komşu çözümü $e^{-\frac{\Delta E}{T}}$ olasılıkla ebeveyn olarak kabul et
Soğuma yöntemine göre $T = g(T)$ güncelle
Çıktı: Türetilen en iyi çözüm

4.2.1 Çözüm kuran algoritma

Tavlama benzetimi algoritması bir başlangıç çözümünün belirlenmesi ile başlamaktadır. Bu amaçla, problem tanımına göre açgözlü bir sezgisel algoritma geliştirilmiştir.

Çalışma kapsamında değerlendirilen problem, teslim tarihine mutlak uyma zorunluluğundan dolayı, literatürde geç kalma amacına yönelik tanımlanan çizelgeleme problemlerine benzemektedir. Bu yüzden başlangıç çözümü belirlenirken, geç kalma amacına yönelik geliştirilen çizelgeleme yaklaşımları temel alınmıştır. Tek makinede en büyük geç kalmanın en küçüklenmesi problemleri için, en erken teslim tarihine sahip işin ilk önce çizelgelenmesi yaklaşımı optimal çözümü vermektedir [10]. Literatürde EDD olarak adlandırılan bu yaklaşımda, işler teslim tarihlerine göre en erken teslim tarihinden en geç teslim tarihine göre sıralanmaktadır. Çözüm kuran algoritma kapsamında, işler öncelikle EDD yaklaşıma göre sıralanmaktadır. Geç kalmanın gözetilmesi sayesinde hem olurlu bir çizelge belirlemek hem de yapılması gereken fazla mesaiyi azaltmak kolaylaşmaktadır.

Teslim tarihleri aynı olan işler ise, ayar zamanlarına göre açgözlü bir şekilde seçilmektedir. Bu yöntemde işler, aynı teslim tarihine sahip henüz çizelgelenmemiş işlerden en son çizelgelenen işle arasında en düşük ayar zamanı olan iş önce olacak şekilde sıralanmaktadır.

Bu sıralamaya göre işler öncelikle üretim tamamen normal mesai içerisinde tamamlanacak şekilde, fazla mesai uygulamaksızın çizelgelenmektedir. Bir işin tamamlanma zamanı teslim tarihini geçiyor ise, ilgili geç kalma miktarı hesaplanmaktadır. Üretimi geç kalan işin üretimi, fazla mesai kullanarak teslim tarihinden önce tamamlanabiliyor ise, gerekli fazla mesai ilgili güne atanmaktadır. Bunun için işin teslim tarihinden geç kalan miktar ile teslim tarihine kadar yapılabilecek toplam fazla mesai miktarı karşılaştırılmaktadır. Fazla mesainin geç kalmayı karşılayabilmesi durumunda, çizelgenin toplam fazla mesai miktarı geç kalma miktarı kadar arttırılmaktadır. Burada yapılacak fazla mesai ilgili teslim tarihinden başlayarak, geriye doğru atanmaktadır. Bu atamanın ardından üretim, belirlenen fazla mesaiyle tamamlanmak üzere tekrar çizelgelenmektedir. Bu adımlar çözüm kuran algoritma ile belirlenen sıralama için olurlu bir çizelge oluşturana kadar tekrarlanmaktadır. Bu aşamada tüm işler normal mesai içerisinde, fazla mesai atanmasına gerek kalmadan tamamlanabiliyorsa elde edilen çizelge optimal çizelgedir. Bu durumda tavlama benzetimi algoritması işletilmemektedir.

Çözüm kuran algoritma ve çizelgeleme algoritması sırayla Algoritma 2 ve 3 ile özetlenmektedir.

Algoritma 2 Çözüm Kuran Algoritma

Girdi: İş sayısı (n), Gün sayısı (t), Gün süresi (ds), Teslim tarihi (D_j), Sıra bağımlı ayar zamanı (A_{jk})

$i = 1; i \leq t; i++$
$G_i \rightarrow \emptyset$
    $j = 1; j \leq n; j++$
    Eğer $D_j = i \cdot ds$ ise
    G_i kümesine j işini ekle
    döngü $\{j\}$
döngü $\{i\}$
$h \rightarrow 1$ ve $sıra_h \rightarrow 0$
$i = 1; i \leq t; i++$
$k \rightarrow 1$
    $k \leq |G_i|$ olduğu sürece tekrarla
    $sıra_h = \min\limits_{j \in G_i}\left(A_{sıra_{(h-1)}j}\right)$
    G_i kümesinden j işini çıkar
    $h = h + 1$ ve $k = k + 1$ olarak güncelle
döngü $\{i\}$

Çıktı: Başlangıç sıralama

Algoritma 3 Çizelgeleme Algoritması

Girdi: Sıralama, İş sayısı (n), Gün sayısı (t), Normal mesai süresi (ns), Gün süresi (ds), Teslim tarihi (D_j) , Sıra bağımlı ayar zamanları (A_{jk})

$mesai_i \to ns$
$h = 1; h \leq n; h++$
$sıra_h$ çizelgele ve tamamlanma zamanı ($C_{sıra_h}$) belirle
  Eğer $C_{sıra_h} > D_{sıra_h}$ ise
  $L_{sıra_h} \to C_{sıra_h} - D_{sıra_h}$, $d \to \frac{D_{sıra_h}}{ds}$
  $A \to 0$
    $i = 1; i \leq d; i++$
    $A = A + (ds - mesai_i)$
    Döngü $\{i\}$
      Eğer $A \geq L_{sıra_h}$ ise
      $L_{sıra_h} > 0$ olduğu sürece tekrarla
        $mesai_d = mesai_d + min\left((ds - mesai_d), L_{sıra_h}\right)$
        $L_{sıra_h} = maks\left(0, L_{sıra_h} - (ds - mesai_d)\right)$
        $d = d - 1$ olarak güncelle
      $h = 1$ olarak güncelle
      Eğer $A < L_{sıra_h}$ ise
      Olursuz çizelge
döngü $\{h\}$
$i = 1; i \leq t; i++$ [1] [1]
$F_i = mesai_i - ns$
döngü $\{i\}$

Çıktı: Belirtilen sıralama için toplam fazla mesai miktarı

Bu algoritmaların uygulanmasına yönelik bir örnek problem ve çözümü aşağıda verilmiştir.

Örnek 1 Çizelge 4.1 ile işlem zamanları ($\boldsymbol{P_j}$) ve teslim tarihleri ($\boldsymbol{D_j}$) ve Çizelge 4.2 ile sıra bağımlı ayar zamanları ($\boldsymbol{A_{jk}}$) belirtilen problem için başlangıç sıralaması çözüm kuran algoritma ile oluşturulmuş ve ardından çizelgeleme algoritması ile bu sıralama için toplam fazla mesai değeri belirlenmiştir.

Çizelge 4.1: Örnek 1: İşlem zamanları ve teslim tarihleri

j	P_j	D_j
1	590	1440
2	630	1440
3	740	2880

Çizelge 4.2: Örnek 1: Sıra bağımlı ayar zamanları

A	0	1	2	3
0	0	100	60	75
1	80	0	70	20
2	90	80	0	50
3	85	50	45	0

Çizelge 4.3, işlerin çözüm kuran algoritmaya göre sıralanması ile oluşturulan başlangıç çözümü gösterilmektedir. Burada aynı teslim tarihine ait işlerden en son çizelgelenen işle, yani bu örnek için makinenin hazır durumuyla (0 numaralı iş), en az sıra bağımlı ayar zamanına sahip olan iş önce sıralanmaktadır.

Çizelge 4.3: Örnek 1: Çözüm kuran algoritma ile oluşturulan sıralama

Sıra	1	2	3
j	2	1	3

Bu sıralama çizelgeleme algoritmasına göre normal mesai içerisinde çizelgelenmeye başlandığında Çizelge 4.4 ile belirtilen şekilde, ikinci işin üretim tamamlanma zamanı teslim tarihini geçmektedir.

Çizelge 4.4: Örnek 1: Çizelgeleme algoritması (1)

	S_j	C_j
2	0	690
1	690	1600
3	-	-

Çizelgeleme algoritmasına göre yapılabilecek fazla mesai miktarı ($\mathbf{A = 240}$), geç kalma miktarını ($\mathbf{L_j = 160}$) karşılayabileceği için ($\mathbf{A \geq L_j}$) bu sıralama için olurlu bir çizelge tanımlanabilmektedir. Bu doğrultuda gerekli fazla mesai miktarının atanmasının ardından üretim tekrar çizelgelenmektedir. Algoritma sonlandırıldığında oluşan çizelge Çizelge 4.5, bu çizelge için günlük ve toplam fazla mesai değerleri ise Çizelge 4.6'da verilmiştir.

Çizelge 4.5: Örnek 1: Çizelgeleme algoritması (2)

	S_j	C_j
2	0	690
1	690	1360
3	1440	2200

Çizelge 4.6: Örnek 1: Başlangıç çözümü için günlük ve toplam fazla mesai değerleri

Gün	Fazla Mesai
1	160
2	0
Toplam	160

4.2.2 Soğuma şeması

Tavlama benzetimi algoritmasının en önemli adımlarından biri soğuma şemasının tanımlanmasıdır. Soğuma şeması ile başlangıç sıcaklık değeri, denge noktası, soğuma yöntemi ve son sıcaklık değeri belirlenmektedir.

Başlangıç sıcaklık değeri

Başlangıç sıcaklık değeri tasarlanan algoritma bir rasgele arama algoritmasına dönüşmeyecek kadar düşük ve de yerel optimal çözüme komşu çözümlerin denenmesini sağlayacak kadar yüksek bir değer olacak şekilde belirlenmelidir. Bu noktada tasarlanan algoritmanın başlangıç sıcaklık değeri yüksek, orta ve düşük sıcaklık değerleri ile yapılan parametre analizleri doğrultusunda belirlenmiştir.

Denge noktası

Algoritmanın her iterasyonunun sıcaklık değeri denge noktasına ulaşana kadar, rasgele komşu çözümler üretilmektedir. Bir başka ifade ile denge noktası bir iterasyonda ebeveyn çözümden üretilecek çocuk çözümün sayısını belirtmektedir. Tasarlanan algoritma kapsamında çocuk çözümler ikili rastgele değişiklikler ile üretilmektedir. Denge noktasının değeri çizelgelenecek iş sayısı ile orantılı olarak yapılan parametre analizlerine göre belirlenmiştir.

Soğuma yöntemi

Tavlama benzetimi algoritmasından elde edilen çözümlerin kalitesi ile soğuma hızı arasında önemli bir ödünleşim vardır. Bu noktada sıcaklığın yavaş düşürülmesiyle çözüm kalitesi artacak, fakat çözüm süresi uzayacaktır. Soğuma yöntemi için genellikle geometrik yaklaşım tercih edilmektedir. Bu doğrultuda tasarlanan algoritma kapsamında da sıcaklık aşağıda belirtilen şekilde güncellenmektedir.

$$T = \alpha T$$

Yapılan çalışmalar, soğuma oranı α değerinin 0,5 ve 0,99 arasında seçilmesini önermektedir [36]. Farklı soğuma oranı değerleri için yapılan analizler doğrultusunda tasarlanan algoritma için soğuma yöntemi belirlenmiştir.

Son sıcaklık değeri

Algoritma komşu çözümlerin kabul edilme olasılığı ihmal edilebilecek kadar azaldığında sonlandırılmaktadır. Bu anlamda belirli bir son sıcaklık değerine ulaşmak veya belirli bir sayıda ard arda iyileşmeyen komşu çözümün ardından durmak gibi yaklaşımlar uygulanabilir. Bu çalışma kapsamında geliştirilen algoritma, literatürde genellikle tercih edilen şekilde belirli bir son sıcak değerine ulaşınca sonlandırılmaktadır. Bu noktada son sıcaklık değeri yeterince düşük değerleri ile yapılan parametre analizleri doğrultusunda belirlenmiştir.

Parametre analizi

Parametre analizleri ile algoritmanın performansı, üç farklı başlangıç sıcaklığı, soğuma oranı, son sıcaklık ve iki farklı denge noktası değerinin kombinasyonları için çözüm değeri ve süresi anlamında değerlendirilmiştir. Bu analizler için toplam 54 farklı soğuma şeması tanımlanmıştır. Algoritmanın olasılıksal seçme süreci dolayısıyla her bir şema için algoritma üç kere koşturulmuştur. Bu soğuma şemaları için algoritmanın performansı optimal çözümün belirlenebildiği örnekler için ilgili çözümle, diğer örneklerde ise belirlenen en iyi çözümle yani üst sınır değeriyle karşılaştırılarak ölçülmektedir. Bu anlamda optimal çözümle yapılan karşılaştırmada ilgili çözümün türetilebildiği, diğer örneklerde ise, ilgili örnek için matematiksel model veya ayrıştırma algoritmasından belirlenen üst sınır değerinden iyi çözümlerin türetilebildiği soğuma şemaları belirlenmiştir. Bununla birlikte bu çözüm değerlerinden sapmalar da değerlendirilmiştir. Burada sapma değerleri sırasıyla aşağıda belirtilen şekilde hesaplanmaktadır.

$$Optimal\ Sapma = \frac{|Sezgisel - Optimal|}{Optimal} \cdot 100$$

$$En\ \dot{I}yi\ Olurlu\ \text{Çözüm}\ Sapma = \frac{|\text{Üst}\ S\imath n\imath r\ - Sezgisel|}{\text{Üst}\ S\imath n\imath r} \cdot 100$$

EK 1 ve EK 2 ile 10 işlik optimal çözümü bilinen örnekler için yapılan analizler verilmiştir. Bu analize göre tavlama benzetimi algoritmasının çözüm performansını etkileyen ayırt edici parametrenin soğuma hızı olduğu belirlenmiştir. Bununla birlikte EK 3ve EK 4 ile verilen 20 işlik örnek değerlendirildiğinde de optimal çözümün düşük soğuma hızı yani büyük soğuma çarpanıyla (0,99) tanımlanan şema ile belirlenebildiği gözlemlenmektedir. İki örnek için de düşük başlangıç, son sıcaklık ve denge noktası değerleriyle tanımlanan soğuma şeması ile optimal çözüm değerine ulaşılabilmiştir. EK 5 ve EK 6 ile verilen 30 işlik örnek incelendiğinde ise bir kez daha soğuma hızının tasarlanan algoritmaya etkisi gözlemlenmiştir. Bu analize göre düşük soğuma hızında tavlama benzetimi algoritmasının ortalama ve en fazla sapma değerlerinin görece az olduğu belirlenmiştir. Fakat denge noktası değerlendirildiğinde, 10 ve 20 işlik örneklerin aksine 30 işlik örnek için yüksek denge noktası ($15 \cdot N/_{10}$) değerinde daha az sapma gözlemlenmiştir. Bununla birlikte 10 ve 20 işlik örneklerde yüksek denge noktası değeri için sapmalar ihmal edilebilecek kadar düşüktür. Sıcaklık parametreleri değerlendirildiğinde ise, yüksek başlangıç (99999) ve orta son sıcaklık (0,1) değerlerinin tüm örneklerde performansının iyi olduğu belirlenmiştir. Bu doğrultuda tasarlanan tavlama benzetimi algoritmasının soğuma şemasının başlangıç sıcaklığı, denge noktası ve soğuma çarpanı değerlerinin yüksek, son sıcaklık değerinin ise orta olarak tanımlanmasına karar verilmiştir.

4.2.3 Tavlama benzetimi algoritması

Algoritma soğuma şemasında belirtilen başlangıç sıcaklık değerinde (99999), çözüm kuran algoritmadan elde edilen başlangıç çözümü ile başlamaktadır. Ebeveyn çözüm olarak belirlenen başlangıç çözümünden, soğuma şemasının denge noktası ($15 \cdot N/_{10}$) kadar komşu çocuk çözümler türetilmektedir. Komşu çözümlerin toplam fazla mesai değeri, çizelgeleme algoritmasına göre hesaplanmaktadır. Türetilen çocuk çözümlerden en iyi olan, yani en küçük fazla mesai değeriyle tamamlanan çizelge, ebeveyn çözümle karşılaştırılmaktadır. Bu çizelge ile çözüm iyileştiriliyorsa, ilgili çocuk çözüm doğrudan ebeveyn çözüm olarak belirlenmektedir. Aksi takdirde, ilgili iterasyonun sıcaklığına göre belirlenen olasılıkla iyileştirmeyen çözüm kabul edilmektedir. Algoritma sıcaklığın, soğuma şemasında belirlenen oranla (0,99) geometrik bir şekilde azaltılmasıyla işlemektedir. Bu esnada her iterasyonun ardından algoritma boyunca türetilen en iyi çözüm güncellenerek saklanmaktadır. Soğuma şemasında tanımlanan son sıcaklık değerinde (0,1), algoritma boyunca belirlenen en iyi çözüm sezgiselin çözümü olmak üzere algoritma sonlandırmaktadır.

Önceki bölümde geliştirilen matematiksel programlama formülasyonu ve bu bölümde geliştirilen Benders ayrıştırma algoritması ve Tavlama Benzetimi algoritmalarının performansları bir sonraki bölümde sunulan deneysel çalışmalarla test edilmiştir.

5. DENEYSEL ÇALIŞMA

Önceki bölümlerde önerilen çözüm yöntemlerinin performansları, bu performansları etkileyebilecek farklı problem parametreleri için test edilmiştir. Matematiksel model ve geliştirilen algoritmaların çözümünde Intel Xeon E5645 2.4 GHz 12 çekirdekli, 12 GB bellek ve 4 paralel işlemcili bir makine kullanılmıştır. Matematiksel model, IBM ILOG CPLEX 12.6.2 çözdürülmüştür. Geliştirilen ayrıştırma ve tavlama benzetimi algoritmaları Java ile kodlanmıştır. Ayrıştırma algoritmasında karma tam sayılı programlama modelleri ILOG CPLEX ile kısıt programlama modelleri ise ILOG CP Optimizer ile çözdürülmüştür. Matematiksel model ve ayrıştırma algoritması için 3600 saniye zaman limiti (ZL) tanımlanmıştır. Matematiksel model ve ayrıştırma algoritmasının bu limit içerisinde optimal çözüme ulaşamaması durumunda, elde edilen en iyi çözüm ve bu çözümün optimale uzaklık değerleri belirlenmiştir.

Problem kapsamında günlük detayda haftalık üretim çizelgesi belirlenmektedir. Bu doğrultuda çizelgeleme periyodu hafta, ayrıştırma algoritması kapsamında değerlendirilen katman boyutu ise gün şeklinde sabit parametreler olarak ele alınmıştır. Gün içerisinde yapılabilecek normal ve fazla mesai süresinin ise çözüm yöntemlerinin performansına bir etkisi olmayacağı öngörüldüğü için ayrıca incelenmemiştir. Bununla birlikte problem boyutunun, yani çizelgelenecek iş sayısının çözüm yöntemlerinin performansı adına önemli bir etkisi olduğu bilinmektedir. Bu sebeple önerilen yöntemlerin performansı 10, 20 ve 30 iş olmak üzere üç farklı boyutta problem için incelenmiştir. İşlem zamanları [100, 1200] değerleri arasında tek düze dağılım gösteren değerler arasından rastgele türetilmiştir. Ayar zamanları üçgen eşitsizliğini sağlayacak şekilde Tran vd. [30]'nin çalışmasında önerilen şekilde üretilmiştir. Bu yaklaşımda koordinat sisteminde $[l, u]$ sınır değerleri arasında tek düze dağılıma göre türetilen rastgele noktalar arasındaki düz çizgili mesafeler, üçgen eşitsizliğine göre ayar zamanlarını belirlemektedir. Ardından, belirlenen alt ve üst sınır değerlerine göre ayar zamanları hesaplanmaktadır. Burada j işinden k işine geçişteki ayar zamanı, k işinden j işine geçişteki ayar zamanından farklı olabileceği için ayar zamanlarını içeren matris asimetrik yapıda olmalıdır.

Bunu sağlayabilmek için, her iş iki farklı koordinat sisteminde birer nokta olarak tanımlanmıştır. Bu anlamda, j işi ilk koordinat sisteminde (x_{j1}, γ_{j1}), ikinci koordinat sisteminde ise (x_{j2}, γ_{j2}) noktalarıyla gösterilmektedir. Buna göre bir j (k) işinden k (j) işine geçerken yapılacak ayarın zamanı aşağıda belirtilen şekilde hesaplanmaktadır.

$$A_{jk} = l + \frac{(u-l)}{100}\left[\left|x_{j1} - x_{k1}\right| + \left|\gamma_{j1} - \gamma_{k1}\right|\right]$$

$$\left(A_{kj} = l + \frac{(u-l)}{100}\left[\left|x_{j2} - x_{k2}\right| + \left|\gamma_{j2} - \gamma_{k2}\right|\right]\right)$$

Burada $[l, u]$ alt ve üst sınır değerlerini belirtmektedir. Bu çalışma kapsamında önerilen çözüm yöntemleri ayar zamanı varyansının yüksek ve düşük olduğu durumlar için karşılaştırılmıştır. Yüksek varyans için ayar zamanları [10, 40], düşük varyans için ise [20, 30] arasında tanımlanmıştır. Çalışma kapsamında son olarak teslim tarihlerinin önerilen yöntemin performansına etkisi incelenmiştir. Bu doğrultuda işlerin teslim tarihlerinin birbirinden farklı olduğu ve hepsinin aynı (son gün) olduğu örnekler oluşturulmuştur. Bu ikinci durum Freeman vd. [21] çalışmasında tanımlanana benzer şekilde, işlerin çizelgeleme periyodu içerisinde tamamlanmasına eşdeğerdir.

Deneysel çalışma için ayrıştırma algoritması iteratif ve dallandırma ve kontrol [43] olmak üzere iki farklı yaklaşımla uygulanmıştır. Birinci yaklaşım için iterasyon, ana problemin çözülmesi ile kesilerin oluşturulmasına kadar olan adımlar olarak tanımlanmaktadır. Bu anlamda iteratif yaklaşım ile mantıksal kesilerin oluşturulması için ana problemin tamamen çözülmesi beklenmektedir. Bu yaklaşımda algoritma, ana problemden belirlenen üst sınır değeri ile alt problemlerden elde edilen alt sınır değeri eşit olana veya belirli bir zaman limitine kadar iteratif bir şekilde işlemektedir. Literatürde dallandırma ve kontrol yaklaşımı (DK) olarak tanımlanan yaklaşımda ise, kesiler dallandırma ve sınırlandırma algoritmasının dallandırma aşamasında oluşturulmaktadır. Bu anlamda kesiler, ana problemin optimal çözümünden değil, ana problem için elde edilen olurlu çözümlerden türetilmektedir. Bu yaklaşımda ana problem iteratif bir şekilde tekrar tekrar çözdürülmek yerine tek bir kere çözdürülmektedir. Beck [44] iki yaklaşımı karşılaştırdığı çalışmasında, doğrudan bir yaklaşımın bütün problemler için diğerinden üstün olduğunu belirtmenin doğru olmayacağı, yaklaşımların performansının ana problem ve alt problemlerinin yapısına bağlı olduğu sonucuna varmıştır. Burada ana problemin çözümünün alt problemlere kıyasla zor olduğu problemler için, DK yaklaşımı ile ana problemin tek bir kere çözülmesinin faydalı olacağı belirtilmiştir. Bu durumun aksine alt problemlerin görece zor olması durumunda ise, DK yaklaşımıyla ana problemin her olurlu çözümü için alt problemleri çözmek yerine iteratif yaklaşımla sadece optimal ana problemin çözümü doğrultusunda alt problemlerin çözülmesinin faydalı olacağı ifade edilmiştir. Bu çalışma kapsamında çizelgeleme alt problemlerinin kısıt programlama yaklaşımıyla etkin bir şekilde çözülebilmesi nedeniyle DK yaklaşımının faydalı olabileceği öngörülmüştür. DK yaklaşımı, ana problemin kodlandığı IBM ILOG CPLEX OPL programlama dilinde tanımlanan "*Callback*" fonksiyonu kullanılarak uygulanmaktadır. Bu fonksiyon ile alt problemlerin çözümünün ardından tanımlanan mantıksal kesiler, ana problemin kısıt havuzuna eklenmektedir. Dallandırma ve sınırlandırma algoritmasında ana problem için tanımlanan olurlu çözümler, bu havuzda yer alan kısıtlar doğrultusunda tekrar kontrol edilmektedir. Eğer tanımlanan bir olurlu çözüm, bu kısıtlardan herhangi birini sağlamıyorsa ilgili çözüm, yani dal kesilmektedir [45]. Bu yaklaşımda, dallandırma ve sınırlandırma algoritması uygulanmaya başladığında ana problem değişkenlerinin bilinmesi gerekmektedir. Fakat bu çalışmada ana problemde kullanılacak değişkenlerin sayısı, mantıksal kesilerle eklenen yeni değişkenler doğrultusunda değişebilmektedir. Bu nedenle algoritma uygulanmaya başladığında kullanılacak değişken sayısı tam olarak bilinmemektedir. Bu yüzden ana problem değişkenlerinin sayısı için bir üst sınır değeri tanımlanmaktadır.

Teslim tarihlerinin farklı, ayar zamanı varyansının yüksek olduğu durumda 10 iş ile oluşturulan örnekler için matematiksel model ve geliştirilen çözüm yöntemlerinden elde edilen sonuçlar Çizelge 5.1 ve Çizelge 5.2 ile özetlenmektedir. Bu örnekler için zaman limiti içerisinde matematiksel modelle optimal çözüm belirlenebilmiştir. Geliştirilen ayrıştırma ve tavlama benzetimi (TB) algoritmalarıyla da optimal çözümlere ulaşılabilmiştir. Ayrıştırma algoritması, hem iteratif yaklaşımla hem de DK yaklaşımıyla uygulandığında, dakikadan kısa bir süre içerisinde sonuç vermiştir. Bu noktada matematiksel model ve geliştirilen çözüm yöntemleri, Çizelge 5.2 ile çözüm süreleri anlamında karşılaştırıldığında, ayrıştırma yönteminin performansının üstün olduğu gözlemlenmektedir. Burada elde edilen en iyi değerler koyu renk ile işaretlenmiştir.

Çizelge 5.1: 10 iş ile farklı teslim tarihleri ve yüksek ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	655	655	655	655
	567	567	567	567
	319	319	319	319
	243	243	243	243
	830	830	830	830

Çizelge 5.2: Farklı teslim tarihleri, yüksek ayar zamanı varyansı için 10 iş ile oluşturulan örneklerin çözüm süreleri (sn)

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	7	9	**4**	42
	167	27	**3**	42
	1239	48	**20**	39
	168	12	**5**	41
	6	10	**4**	42
Ortalama	317,40	21,20	**7,20**	41,40

10 iş ile oluşturulan örnekler dışında, 20 ve 30 iş için zaman limiti içerisinde matematiksel modelle olurlu bir çözüm üretilememiştir. Matematiksel model ile çözüm üretilemeyen örneklerin hepsi için, ayrıştırma ve tavlama benzetimi algoritmalarıyla Çizelge 5.3 ile gösterilen üst sınır değerleri elde edilmiştir. Bununla birlikte Çizelge 5.4 ile zaman limitinin sonunda ayrıştırma algoritmasının alt ve üst sınırlarının yüzde sapma değerleri gösterilmektedir. Bu yüzde sapma değeri aşağıda belirtilen şekilde hesaplanmaktadır.

$$Alt\ Sınır\ Yüzde\ Sapma\ (ASYS) = \frac{Üst\ Sınır - Alt\ Sınır}{Alt\ Sınır} \cdot 100$$

Çizelge 5.3: 20 ve 30 iş ile farklı teslim tarihleri ve yüksek ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	**1070**	**1070**	1079
	804	**802**	805
	1198	**1188**	1207
	406	**406**	407
	775	**771**	775
30 İş	890	871	**853**
	972	**969**	974
	401	**398**	402
	892	889	**877**
	484	**483**	497

Çizelge 5.4: Farklı teslim tarihleri, yüksek ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örnekler için ASYS değerleri

	Ayrıştırma Algoritması	
	İteratif	DK
20 İş	19,55	**14,85**
	12,76	**0,0**
	16,31	**0,67**
	14,68	**0,0**
	16,54	36,72
Ortalama	15,97	**10,45**
30 İş	51,60	**51,23**
	33,01	**20,14**
	53,36	**52,11**
	60,18	**51,56**
	3,64	**2,43**
Ortalama	40,36	**35,49**

Çizelge 5.4'de gösterilen ASYS değerleri incelendiğinde, bir örnek hariç tüm örneklerde DK yaklaşımının sapma değerlerinin iteratif yaklaşımdan daha düşük olduğu gözlemlenmektedir. Ortalama ASYS değerleri değerlendirildiğinde de, iteratif yaklaşım ile sapmanın daha iyi olduğu örneğe rağmen, DK yaklaşımının daha iyi sonuç verdiği görülmektedir. Bununla birlikte Çizelge 5.3'de verilen olurlu çözüm değerleri incelendiğinde, DK yaklaşımı ile elde edilen üst sınır değerlerinin tüm örneklerde iteratif yaklaşımdan daha düşük olduğu görülmektedir. Bu anlamda bir örnekte iteratif yaklaşımın sapma değerinin DK yaklaşımına göre daha iyi olmasının nedeni, DK yaklaşımından elde edilen alt sınır değerinin daha kötü olmasıdır.

Geliştirilen çözüm yöntemlerinin çözüm süreleri ve bu süre içerisinde ayrıştırma ve tavlama benzetimi algoritmasıyla elde edilen en iyi olurlu çözümden yüzde sapma değerleri sırasıyla Çizelge 5.5 ve Çizelge 5.6 ile sunulmuştur. Çizelge 5.5'de görüldüğü üzere, iş sayısı arttıkça, ayrıştırma algoritması zaman limiti içerisinde optimal çözümü belirleyememektedir. Buna karşılık, TB algoritmasının çözüm zamanında iş sayısına bağlı olarak küçük bir artış gözlemlenmektedir, fakat çözüm süreleri hala makul seviyelerdedir. Burada Çizelge 5.6'da verilen sapma değerleri zaman limiti içerisinde belirlenen en iyi olurlu çözümden, yani en küçük üst sınır değerinden sapma olarak aşağıda belirtilen şekilde hesaplanmaktadır.

$$En\ \text{İ}yi\ Olurlu\ \text{Çözüm}\ Yüzde\ Sapma\ (EOÇYS) = \frac{En\ \text{İ}yi\ Olurlu\ \text{Çözüm} - Yöntemin\ \text{Çözümü}}{En\ \text{İ}yi\ Olurlu\ \text{Çözüm}} \cdot 100$$

Çizelge 5.5: Farklı teslim tarihleri, yüksek ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örneklerin çözüm süreleri (sn)

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	ZL	ZL	**49**
	ZL	2330	**49**
	ZL	ZL	**44**
	ZL	2727	**50**
	ZL	ZL	**46**
30 İş	ZL	ZL	**70**
	ZL	ZL	**90**
	ZL	ZL	**76**
	ZL	ZL	**63**
	ZL	ZL	**58**

Çizelge 5.6: Farklı teslim tarihleri, yüksek ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örnekleri için EOÇYS değerleri

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	**0,00**	**0,00**	0,84
	0,25	**0,00**	0,37
	0,84	**0,00**	1,60
	0,00	**0,00**	0,25
	0,52	**0,00**	0,52
Ortalama	0,32	**0,00**	0,72
30 İş	4,34	2,11	**0,00**
	0,31	**0,00**	0,52
	0,75	**0,00**	1,01
	1,71	1,37	**0,00**
	0,21	**0,00**	2,90
Ortalama	1,46	**0,70**	0,88

20 iş ile oluşturulan örneklerde Çizelge 5.4'de gösterilen optimal çözümün belirlenebildiği örnek sayısı ve ASYS değerleri ile birlikte, EOÇYS değerleri değerlendirildiğinde de, bu örnekler için DK yaklaşımının iteratif yaklaşımdan daha etkili olduğu sonucuna varılmaktadır. Bununla birlikte bu örnekler için genel olarak ayrıştırma algoritmasıyla TB algoritmasından daha güçlü üst sınır, yani daha düşük EOÇYS değerleri tanımlanabildiği gözlemlenmektedir. Fakat TB algoritmasıyla da kaliteli çözümler elde edilebilmektedir. Bu noktada çözüm kalitesi ve süresi arasında bir ödünleşim bulunmaktadır. Benzer şekilde 30 işlik örneklerde de DK yaklaşımının çözüm performansı, gerek Çizelge 5.4 ile gösterilen ASYS gerekse Çizelge 5.6 ile gösterilen EOÇYS değerlerine göre, iteratif uygulamadan iyidir. Fakat 20 işlik örneklerden farklı olarak, 30 işlik örneklerde 2 örnek için TB algoritmasıyla DK yaklaşımına göre daha iyi olurlu çözümler üretilebilmiştir. Bununla birlikte ortalama EOÇYS değerleri incelendiğinde, TB algoritmasının çözüm performansının iteratif uygulamadan daha iyi olduğu gözlemlenmektedir. Bu sonuç ve ASYS değerleri birlikte değerlendirildiğinde, iş sayısıyla problem boyutu arttıkça TB algoritmasının görece etkili bir yaklaşım haline geldiği sonucuna varılmaktadır.

Çizelge 5.7'de ayar zamanı varyansının düşük olduğu durumda 10 iş ile oluşturulan örneklerden elde edilen sonuçlar verilmiştir. Bu durumda da gerek matematiksel model gerekse geliştirilen çözüm yöntemleri ile optimal çözüm elde edilebilmiştir. Bu doğrultuda Çizelge 5.8 de verilen çözüm süreleri incelendiğinde, yüksek varyans durumunda gözlemlendiği gibi, ayrıştırma algoritmasının performansının matematiksel modelden daha iyi olduğu gözlemlenmektedir. Bununla birlikte yüksek ve düşük varyans durumları arasında, DK yaklaşımının çözüm süresinde ortalama %5,56 artış gözlemlenirken, matematiksel modelde ortalama %125,52 artış gözlemlenmiştir. Düşük varyans durumunda, iteratif yaklaşım ile DK yaklaşımı arasında çözüm süresi anlamında daha keskin bir ayrım vardır. Burada iteratif yaklaşım ile sadece bir örnek için optimal çözüm dakikadan kısa bir süre içerisinde belirlenebilirken, DK yaklaşımıyla hala hiçbir örnek için çözüm süresi dakikayı geçmemektedir. 10 işlik örnekler için, ortalama çözüm süresi anlamında TB algoritması iteratif yaklaşıma göre daha iyi bir performans gösterirken, DK yaklaşımı hala önerilen yöntemler içerisinde en iyisidir.

Çizelge 5.7: 10 iş ile farklı teslim tarihleri ve düşük ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	694	694	694	694
	608	608	608	608
	350	350	350	350
	258	258	258	258
	870	870	870	870

Çizelge 5.8: Farklı teslim tarihleri, düşük ayar zamanı varyansı için 10 iş ile oluşturulan örneklerin çözüm süreleri (sn)

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	9	79	**3**	43
	167	119	**5**	42
	1153	286	**23**	50
	2245	73	**4**	39
	5	40	**3**	53
Ortalama	715,80	119,40	**7,60**	45,30

Düşük varyans durumunda da 20 ve 30 iş ile oluşturulan örnekler için matematiksel modelle zaman limiti içerisinde olurlu çözüm belirlenememiştir. Bu örnekler için ayrıştırma ve tavlama benzetimi algoritmaları ile üretilen üst sınır değerleri Çizelge 5.9'da, ayrıştırma algoritmasından elde edilen alt ve üst sınır değerlerinin yüzde sapması ise Çizelge 5.10'da sunulmuştur. Bu örneklerin tamamı için DK yaklaşımının hem ASYS hem de üst sınır değerleri iteratif yaklaşımdan iyidir. Bu anlamda DK yaklaşımının iteratif yaklaşıma göre üstünlüğünü koruduğu sonucuna varılmaktadır.

Çizelge 5.9: 20 ve 30 iş ile farklı teslim tarihleri ve düşük ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	1172	**1168**	1172
	930	**928**	931
	1305	**1295**	1299
	546	**544**	548
	899	**896**	896
30 İş	1065	1055	**1050**
	1139	**1133**	1137
	526	**525**	528
	1098	1097	**1093**
	685	**680**	680

Çizelge 5.10: Farklı teslim tarihleri, düşük ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örneklerin ASYS değerleri

	Ayrıştırma Algoritması	
	İteratif	DK
20 İş	22,98	**13,49**
	24,33	**19,50**
	24,40	**22,59**
	25,51	**24,52**
	26,08	**23,33**
Ortalama	24,66	**20,67**
30 İş	54,08	**50,44**
	39,01	**29,43**
	62,66	**59,65**
	84,53	**60,61**
	53,14	**30,72**
Ortalama	56,68	**46,17**

Bu örnekler için ayrıştırma ve TB algoritmalarının çözüm süreleri Çizelge 5.11 ve bu süre içerisinde elde edilen EOÇYS değerleri ise Çizelge 5.12 ile verilmiştir. Bu durumda da yüksek varyans durumu ile benzer şekilde, ayrıştırma algoritmasıyla zaman limiti içerisinde optimal çözümün elde edilmesi mümkün olmazken, TB algoritmasının ortalama çözüm süresinde küçük bir artış gözlemlenmektedir. Bununla birlikte EOÇYS değerlerine göre ortalamada, DK yaklaşımı ile TB algoritmasına göre daha iyi olurlu çözümler türetilebilmiştir. 20 iş ile oluşturulan tüm örneklerde ise, yüksek varyans durumunda gözlemlendiği gibi, en iyi olurlu çözümler DK yaklaşımından elde edilmiştir. Bu örnekler için TB algoritmasının ortalama EOÇYS değeri ise%0,34 olarak hesaplanmıştır. Bu değer yüksek varyans durumu için %0,72 olarak belirlenmiştir. Bununla birlikte 30 iş ile oluşturulan örneklerde yüksek ve düşük varyans durumu için EOÇYS değerleri karşılaştırıldığında, yüksek varyans durumunda DK yaklaşımının ortalama EOÇYS değerinin TB algoritmasından ortalama %25,71 daha iyi, düşük varyans durumunda ise %5,88 daha iyi olduğu gözlemlenmiştir. Bu doğrultuda EOÇYS değerleri ile birlikte, düşük ve yüksek varyans durumlarında belirlenen ASYS değerleri değerlendirdiğinde ayrıştırma algoritmasının yüksek varyans durumunda daha etkili olduğu sonucuna varılmaktadır. Bu sonuç ayar zamanlarının varyansının azalmasıyla bir çözümü diğerinden açıkça ayırmanın zorlaşması ile açıklanmaktadır. Bu durumda kesilerle tanımlanan bilgilerin ayırt ediciliği yani işlevselliği azalmaktadır.

Çizelge 5.11: Farklı teslim tarihleri, düşük ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örneklerin çözüm süreleri (sn)

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	ZL	ZL	**51**
	ZL	ZL	**46**
	ZL	ZL	**48**
	ZL	ZL	**48**
	ZL	ZL	**44**
30 İş	ZL	ZL	**75**
	ZL	ZL	**105**
	ZL	ZL	**63**
	ZL	ZL	**75**
	ZL	ZL	**100**

Çizelge 5.12: Farklı teslim tarihleri, düşük ayar zamanı varyansı için 20 ve 30 iş ile oluşturulan örnekler için EOÇYS değerleri

	Ayrıştırma Algoritması		TB Algoritması
	İteratif	DK	
20 İş	0,34	**0,00**	0,84
	0,22	**0,00**	0,37
	0,77	**0,00**	1,60
	0,33	**0,00**	0,25
	0,41	**0,00**	0,52
Ortalama	0,41	**0,00**	0,34
30 İş	1,43	0,48	**0,00**
	0,53	**0,00**	0,35
	0,19	**0,00**	0,57
	0,73	0,37	**0,00**
	0,74	**0,00**	**0,00**
Ortalama	0,72	**0,17**	0,18

Haftalık üretim planında çizelgelenecek işlerinin tamamının teslim tarihlerinin son gün olarak tanımlanması durumu, Freeman vd. [21] çalışmasında değerlendirilen duruma benzemektedir. Bu anlamda üretimin teslimi üzerinde herhangi bir kısıt tanımlanmaksızın, üretim çizelgesinin planlama periyodu içerisinde tamamlanmasını amaçlamak yeterli olacaktır. Fakat burada tanımlanan problem, ayar varsayımları nedeniyle Freeman vd. [21] çalışmasında tanımlanan problemden farklıdır. Üretim ayarlarının parçalanamıyor olması, problemin doğrudan yayılma zamanını en küçükleme amacıyla tanımlanmasına engel olmaktadır. Bu anlamda burada tanımlanan problemin modellenmesi, Freeman vd. [21] çalışmasından değerlendirilen problemden daha karmaşıktır.

Teslim tarihlerinin haftanın sonunda ve aynı olduğu durumda, ayar zamanı varyansının yüksek olduğu örnekler için matematiksel model ve geliştirilen çözüm yöntemlerinin sonuçlar Çizelge 5.13'de verilmiştir. Burada optimal çözümün belirlenebildiği örnekler, gevşetme sayesinde haftalık üretim çizelgesinin normal mesai içerisinde tamamlanabildiği, görece kolay örneklerdir. Bu doğrultuda gerek matematiksel model gerekse geliştirilen çözüm yöntemleri ile optimal çözüme ulaşabilmiştir. Çizelge 5.14 ile gösterilen çözüm süreleri incelendiğinde, bu örnekler için matematiksel model ve ayrıştırma algoritmasının çözüm sürelerinin oldukça kısa olduğu gözlemlenmektedir. Matematiksel model ve ayrıştırma algoritması ile zaman limiti içerisinde optimal çözümün belirlenemediği örneklerde ise, TB algoritması kısa süre içerisinde çözüm vermektedir.

Çizelge 5.13: Haftanın sonunda ve aynı teslim tarihleri ve yüksek ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	**117**	134	128	**117**
	0	**0**	**0**	**0**
	144	157	144	**131**
	0	**0**	**0**	**0**
	466	476	470	**463**
20 İş	1020	1065	1052	**985**
	802	880	792	**750**
	1167	1220	1188	**1128**
	389	425	414	**359**
	770	767	757	**719**
30 İş	858	850	853	**773**
	903	949	958	**873**
	0	67	0	**0**
	896	903	893	**821**
	469	471	470	**389**

Çizelge 5.14: Teslim tarihlerinin sonda ve aynı olduğu durumda, yüksek ayar zamanı varyansı için oluşturulan örneklerin çözüm süreleri (sn)

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	ZL	ZL	ZL	**46**
	2	5	**2**	63
	ZL	ZL	ZL	**63**
	2	14	11	83
	ZL	ZL	ZL	**55**
20 İş	ZL	ZL	ZL	**76**
	ZL	ZL	ZL	**64**
	ZL	ZL	ZL	**69**
	ZL	ZL	ZL	**75**
	ZL	ZL	ZL	**54**
30 İş	ZL	ZL	ZL	**106**
	ZL	ZL	ZL	**92**
	153	ZL	**18**	53
	ZL	ZL	ZL	**99**
	ZL	ZL	ZL	**92**

Bu durumda, farklı teslim tarihleriyle oluşturulan örneklerden farklı olarak, 20 ve 30 iş örnekleri için zaman limiti içerisinde matematiksel model ile olurlu çözüm türetilebilmiştir. Bu durum teslim tarihlerinin gevşetilmesiyle olurlu çözüm alanının genişlemesinin bir sonucudur. Fakat olurlu alanının genişlemesi, denenmesi gereken çözüm sayısını arttırarak optimal çözümün belirlenmesini zorlaştırmaktadır. Çizelge 5.15'de verilen yüksek ASYS değerleri ve 10 iş ile oluşturulan örneklerde, farklı teslim tarihleri tanımlanması durumuna göre, optimal çözümün belirlendiği örnek sayısında gözlemlenen düşüş, bu doğrultuda açıklanabilmektedir. Burada özellikle matematiksel model için ASYS değerinin oldukça yüksek, ayrıştırma algoritmasının ise görece düşük olduğu gözlemlenmektedir Çizelge 5.13'de verilen değerler incelendiğinde ayrıştırma algoritması ile belirlenebilen üst sınır değerlerinin matematiksel modele göre daha zayıf, yani daha yüksek olduğu görülmektedir. Bu anlamda ayrıştırma algoritmasının gerek DK gerek iteratif yaklaşım için ASYS değerlerinin daha düşük olması, önerilen çözüm yöntemi ile matematiksel modele göre daha sıkı, yani daha yüksek alt sınır değerleri belirlenebildiğini göstermektedir. Ayrıştırma algoritmasının uygulamalarının ASYS değerleri karşılaştırıldığında ise, bir kez daha DK yaklaşımın iteratif uygulamaya göre daha başarılı olduğu gözlemlenmektedir.

Çizelge 5.15: Teslim tarihlerinin sonda ve aynı olduğu durumda, yüksek ayar zamanı varyansı için oluşturulan örnekler için ASYS değerleri

	MIP	Ayrıştırma Algoritması	
		İteratif	DK
10 İş	**100,00**	**100,00**	**100,00**
	0,00	**0,00**	**0,00**
	100,00	**100,00**	**100,00**
	0,00	**0,00**	**0,00**
	100,00	34,63	**31,27**
Ortalama	60,00	46,93	**46,25**
20 İş	100,00	45,29	**30,32**
	100,00	62,96	**39,85**
	100,00	41,20	**27,27**
	100,00	75,76	**75,12**
	100,00	64,56	**40,02**
Ortalama	100,00	57,95	**42,52**
30 İş	100,00	53,64	**52,70**
	100,00	99,36	**50,31**
	0,00	100,00	**0,00**
	100,00	52,15	**51,62**
	100,00	95,96	**95,95**
Ortalama	80,00	80,22	**50,12**

Teslim tarihlerinin aynı, ayar zamanı varyansının yüksek olduğu örnekler için elde edilen EOÇYS değerleri Çizelge 5.16'da gösterilmektedir. Bu sapma değerlerine göre, tüm örneklerde en iyi olurlu çözüm değeri TB algoritması ile elde edilmiştir. Bununla birlikte TB algoritmasıyla bu çözümler, matematiksel model ve ayrıştırma algoritmasına göre çok daha kısa süre içerisinde türetilmiştir. EOÇYS değerlerine göre de, bu örnekler için DK yaklaşımının iteratif yaklaşımdan üstün olduğu gözlemlenmektedir. Matematiksel model ve genel olarak ayrıştırma algoritması karşılaştırıldığında ise, iş sayısının artmasıyla matematiksel modelden elde edilen çözümlerden sapmanın azaldığı belirlenmiştir. Bu doğrultuda problem boyutu arttıkça ayrıştırma yönteminin matematiksel model üzerinde üstünlük kazanmaya başladığı sonucuna varılmaktadır.

Çizelge 5.16: Teslim tarihlerinin sonda ve aynı olduğu durumda, yüksek ayar zamanı varyansı için oluşturulan örnekler için EOÇYS değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	**0,00**	14,53	9,40	**0,00**
	0,00	**0,00**	**0,00**	**0,00**
	9,92	19,85	9,92	**0,00**
	0,00	**0,00**	**0,00**	**0,00**
	0,65	2,81	1,51	**0,00**
Ortalama	2,11	7,44	4,17	**0,00**
20 İş	3,55	8,12	6,80	**0,00**
	6,93	7,73	5,60	**0,00**
	3,46	8,16	5,32	**0,00**
	8,36	18,38	15,32	**0,00**
	7,09	6,68	5,29	**0,00**
Ortalama	5,88	9,81	7,67	**0,00**
30 İş	11,00	9,96	7,76	**0,00**
	3,44	8,71	9,74	**0,00**
	0,00	**0,00**	**0,00**	**0,00**
	9,14	9,99	8,77	**0,00**
	20,57	21,08	20,82	**0,00**
Ortalama	8,83	9,95	9,42	**0,00**

Son olarak bu durum için ayar zamanı varyansının düşük olduğu örnekler değerlendirilmiştir. Çizelge 5.17 ve Çizelge 5.18 ile elde edilen sonuçlar özetlenmiştir. Bu durumda da, yüksek varyans durumunda belirtilen görece kolay örneklerde, gerek matematiksel model gerekse geliştirilen çözüm yöntemleri ile optimal çözüm belirlenebilmiştir. Bu örnekler için ayrıştırma algoritmasının çözüm süresinde yüksek varyans durumuna göre bir artış gözlemlenmesine rağmen, DK yaklaşımı ile hala dakikadan kısa bir süre içerisinde optimal çözüm üretilebilmektedir. Optimal çözümün belirlenemediği örneklerde ise, yine TB algoritması ile kısa süre içerisinde olurlu çözümler türetilmektedir.

Çizelge 5.17: Haftanın sonunda ve aynı teslim tarihleri ve düşük ayar zamanı varyansı durumunda oluşturulan örnekler için çözüm değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	185	187	186	**182**
	0	**0**	**0**	**0**
	216	221	220	**215**
	0	**0**	**0**	**0**
	544	548	547	**540**
20 İş	1171	1182	1172	**1154**
	918	929	930	**906**
	1293	-	1297	**1281**
	537	549	543	**523**
	892	905	897	**880**
30 İş	1050	1055	1053	**1020**
	1142	1139	1138	**1101**
	143	167	163	**120**
	1086	1090	1088	**1057**
	673	675	674	**647**

Çizelge 5.18: Teslim tarihlerinin sonda ve aynı olduğu durumda, düşük ayar zamanı varyansı için oluşturulan örneklerin çözüm süreleri (sn)

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	ZL	ZL	ZL	**46**
	2	79	26	38
	ZL	ZL	ZL	**43**
	1	189	16	41
	ZL	ZL	ZL	**41**
20 İş	ZL	ZL	ZL	**77**
	ZL	ZL	ZL	**80**
	ZL	ZL	ZL	**82**
	ZL	ZL	ZL	**59**
	ZL	ZL	ZL	**71**
30 İş	ZL	ZL	ZL	**114**
	ZL	ZL	ZL	**106**
	ZL	ZL	ZL	**67**
	ZL	ZL	ZL	**108**
	ZL	ZL	ZL	**100**

Bu durumda da Çizelge 5.19'da gösterilen şekilde zaman limiti içerisinde ayrıştırma algoritması ile türetilen alt ve üst sınırların yüzde sapma değerleri, matematiksel modele göre daha düşüktür. Bununla birlikte Çizelge 5.17 ile verilen üst sınır değerleri incelendiğinde, matematiksel model ile belirlenen değerlerin daha düşük olduğu görülmektedir. Bu anlamda ayrıştırma algoritmasında tanımlanan gevşetilmiş problem ile, matematiksel modele göre daha sıkı alt sınır değerleri belirlenebildiği sonucuna varılmaktadır. Bununla birlikte bu durumda 10 iş ile oluşturulan örneklerde, iteratif yaklaşım için ASYS değerlerinin DK yaklaşımına göre daha düşük olduğu gözlemlenmektedir. Fakat Çizelge 5.17 ile verilen üst sınır değerleri incelendiğinde, DK yaklaşımının değerlerinin daha düşük olduğu görülmektedir. Bu sebeple bu örnekler için en sıkı alt sınır değerlerinin iteratif yaklaşım ile belirlendiği sonucuna varılmaktadır. 20 ve 30 iş ile oluşturulan örnekler incelendiğinde ise hem ASYS hem de üst sınır değerleri anlamında DK yaklaşımının iteratif yaklaşımdan üstün olduğu gözlemlenmektedir. Bununla birlikte iteratif yaklaşımla Çizelge 5.17'de çözüm değeri "-" ile gösterilen örnek için olurlu çözüm belirlenememektedir. DK yaklaşımı ile ise tüm örnekler için olurlu çözüm ve matematiksel model ile belirlenebilen alt sınır değerinden güçlü alt sınır değerleri belirlenebilmektedir. Bu doğrultuda örneklerin çoğunluğu değerlendirildiğinde, DK yaklaşımının iteratif yaklaşımdan üstün olduğu yorumu yapılabilmektedir.

Çizelge 5.19: Teslim tarihlerinin sonda ve aynı olduğu durumda, düşük ayar zamanı varyansı için oluşturulan örnekler için ASYS değerleri

	MIP	Ayrıştırma Algoritması	
		İteratif	DK
10 İş	**100,00**	**100,00**	**100,00**
	0,00	**0,00**	**0,00**
	100,00	**100,00**	**100,00**
	0,00	**0,00**	**0,00**
	100,00	40,51	40,97
Ortalama	60,00	**48,10**	48,19
20 İş	100,00	61,25	**37,45**
	100,00	91,15	**47,74**
	100,00	-	**33,38**
	100,00	81,23	**81,03**
	100,00	96,73	**48,71**
Ortalama	100,00	82,59	**49,66**
30 İş	100,00	62,65	**62,58**
	100,00	58,20	**58,17**
	100,00	**100,00**	**100,00**
	100,00	60,36	**60,29**
	100,00	**97,18**	**97,18**
Ortalama	100,00	75,68	**75,64**

Çizelge 5.20'de gösterilen en iyi olurlu çözümden yüzde sapma değerleri incelendiğinde ise, yüksek varyans durumunda gözlemlendiği gibi, en iyi olurlu çözümlerin TB algoritması ile elde edildiği gözlemlenmektedir. Fakat bu durumunda ortalama EOÇYS değerleri değerlendirildiğinde, matematiksel model ve ayrıştırma algoritmasının sapma değerlerinin genel olarak yüksek varyans durumuna göre azaldığı görülmektedir. Burada iteratif yaklaşım için ortalama EOÇYS değeri hesaplanırken, olurlu çözümün belirlenemediği örnek hesaplamaya dahil edilmemiştir. DK yaklaşımı için hesaplanan ortalama EOÇYS değerleri, yüksek varyans durumuna göre 10 ve 20 iş ile oluşturulan örneklerde sırasıyla %72,18 ve %70,79 kadar azalırken, 30 iş ile oluşturulan örneklerin EOÇYS değerlerinin ortalamasında %5,20 artış gözlemlenmiştir. Bu doğrultuda teslim tarihlerinin aynı olduğu durumda ayar zamanların varyansının azalmasıyla genel anlamda TB algoritmasının DK yaklaşımı üzerindeki üstünlüğünün görece azaldığı sonucuna varılmaktadır.

Çizelge 5.20: Teslim tarihlerinin sonda ve aynı olduğu durumda, düşük ayar zamanı varyansı için oluşturulan örnekler için EOÇYS değerleri

	MIP	Ayrıştırma Algoritması		TB Algoritması
		İteratif	DK	
10 İş	1,65	2,75	2,20	**0,00**
	0,00	**0,00**	**0,00**	**0,00**
	0,47	2,79	2,33	**0,00**
	0,00	**0,00**	**0,00**	**0,00**
	0,74	1,48	1,30	**0,00**
Ortalama	0,57	1,40	1,16	**0,00**
20 İş	1,47	2,43	1,56	**0,00**
	1,32	2,54	2,65	**0,00**
	0,94	-	1,25	**0,00**
	2,68	4,97	3,82	**0,00**
	1,36	2,84	1,93	**0,00**
Ortalama	1,55	3,19	2,24	**0,00**
30 İş	2,94	3,43	3,24	**0,00**
	3,72	3,45	3,36	**0,00**
	19,17	39,17	35,83	**0,00**
	2,74	3,12	2,93	**0,00**
	4,02	4,33	4,17	**0,00**
Ortalama	6,52	10,70	9,91	**0,00**

Teslim tarihlerinin sonda ve aynı olduğu durumdan elde edilen sonuçlar genel olarak değerlendirildiğinde ise ayrıştırma algoritmasının farklı teslim tarihleriyle tanımlanan örnekler için daha etkili bir çözüm yöntemi olduğu sonucuna varılmaktadır. Bu durum teslim tarihlerinin gevşetilmesiyle ana problemde yapılan atama kısıtlamalarının azalmasıyla açıklanmaktadır. Burada olurlu çözüm sayısının artmasıyla, optimal çözümün belirlenmesi için çözülmesi gereken ana problem ve devamında alt problemlerin sayısı artmaktadır. Bu nedenle zaman limiti içerisinde belirlenebilen ASYS değeri yükselmektedir. Fakat bu sapma değeri hala matematiksel modele göre daha düşüktür. Bu doğrultuda ayrıştırma algoritmasının bir üst sınır değeri belirlemek için kullanılabileceği gibi, alt sınır belirlemek adına da kullanılabileceği vurgulanmaktadır. Burada üst sınır değeri ve EOÇYS değerleri değerlendirildiğinde ise, tüm örnekler için en iyi olurlu çözümlerin TB algoritması ile belirlendiği gözlemlenmektedir. Bu anlamda gerek çözüm süresi gerekse bu süre içerisinde elde edilen çözümlerin kalitesine göre, teslim tarihlerinin aynı olduğu durum için sezgisel TB algoritmasıyla çözüm üretmenin, DK yaklaşımı ile kesin çözüm aramaktan daha etkili bir yöntem olduğu sonucuna varılmaktadır.

6. SONUÇ VE DEĞERLENDİRME

Bu çalışmada bir gerçek hayat üretim çizelgeleme problemi ele alınmıştır. Üretim sıra bağımlı ayar zamanları ile tek makinede gerçekleşmektedir. Problem kapsamında en küçük toplam fazla mesai değeri ile siparişlerin teslim tarihlerine yetiştirilmesini sağlayacak üretim çizelgesini belirlemek amaçlanmaktadır. Burada çizelge günlük detayda oluşturulmaktadır. Olurlu bir çizelge için tüm siparişlerin gün sonlarında tanımlanan teslim tarihlerine yetişmesi gerekmektedir. Bu anlamda eğer üretim normal mesai süresi içerisinde tamamlanamıyorsa, normal mesainin ardına her gün belirli bir süreyi aşmayacak şekilde fazla mesai yapılmaktadır. Günlük olarak tanımlanan fazla mesai, yapılan süre ile orantılı olarak maliyetlendirilmektedir.

Bu doğrultuda öncelikle problem tanımına yönelik bir literatür taraması yapılmıştır. Literatür taramasında problem varsayımlarına göre yapılan çalışmalarla birlikte, problemin çözümü için önerilen ayrıştırma yöntemine yönelik çalışmalar da incelenmiştir. Problemin çözümü için öncelikle bir karma tam sayılı matematiksel programlama modeli geliştirilmiştir. Problemin günlük bir planlama ufku içeren özel durumu ele alındığında, toplam fazla mesai en küçükleme problemi, tek makine, sıra bağımlı ayar zamanıyla yayılma zamanının en küçüklenmesi problemine dönüşmektedir. Bu problemin NP-Zor olduğu bilindiğinden, mevcut çalışmada ele alınan genel problem de NP-Zor statüsündedir. Bu sebeple problemin çözümü için, literatürde kombinatoryal optimizasyon problemlerine başarıyla uygulanan Benders ayrıştırma yöntemi geliştirilmiştir. Literatürde mantıksal Benders ayrıştırma algoritması olarak adlandırılan bu yaklaşıma göre haftalık üretim çizelgeleme problemi, işlerin haftanın günlerine atanması ve bu atamalar doğrultusunda günlük çizelgelerin belirlenmesi şeklinde ana ve alt problemlere ayrıştırılmaktadır. Geliştirilen algoritmaya göre, ana problem karma tam sayılı programlama ile çizelgeleme alt problemleri ise kısıt programlama ile çözülmektedir. Ana problemden elde edilen toplam fazla mesai değeri esas problem için bir alt sınır oluştururken, alt problemlerin tamamının olurlu olması durumunda bu çözümden elde edile edilen toplam fazla mesai değeri ise bir üst sınır oluşturmaktadır. Algoritma alt ve üst sınır değerleri birbirine eşit olduğunda, optimal çözümle sonlanmaktadır. Mevcut çalışmada, Benders ayrıştırma algoritmasının iteratif ve DK şeklinde iki farklı uyarlaması yapılmıştır. Çalışma kapsamında ayrıştırma algoritması dışında, makul süreler içerisinde optimal çözümü belirlemenin zorlaştığı örnekler için bir sezgisel yaklaşım önerilmiştir. Bu doğrultuda tasarlanan TB algoritması ile kısa süre içerisinde kaliteli çözümler üretilebilmektedir. Geliştirilen çözüm yöntemleri, performanslarını etkileyebilecek çeşitli problem parametreleri için üretilen örnekler kullanarak test edilmiştir.

İşlerin teslim tarihlerinin farklı günlerde tanımlandığı 10 iş ile oluşturulan örneklerde ayrıştırma algoritmasının DK yaklaşımıyla uygulanmasıyla hem matematiksel modelden hem de TB algoritmasından daha kısa sürede optimal çözümler elde edilebilmiştir. Ayar zamanı varyansına bağlı olarak DK yaklaşımının çözüm süresinde ortalama %5,56 artış gözlemlenirken, matematiksel modelle yüksek ve düşük varyans durumları arasında ortalama %125,52 artış gözlemlenmiştir. Bununla birlikte, teslim tarihlerinin farklı olduğu durumda 20 ve 30 iş ile oluşturulan örneklerde matematiksel modelle olurlu bir çözüm türetilemezken, DK yaklaşımı ile iki örnek için optimal çözüm diğer tüm örnekler için de olurlu bir çözüm türetilebilmiştir. DK yaklaşımı ile optimal çözümün belirlenemediği örneklerde alt ve üst sınır arasında gözlemlenen en küçük yüzde sapma değeri %0,67 ve en büyük yüzde fark değeri ise %60,61 olarak ortaya çıkmıştır. Burada en küçük ASYS değeri ayar zamanı varyansının yüksek olduğu 20 iş ile oluşturulan örneklerde gözlemlenirken, en büyük fark değeri ise, varyansın düşük olduğu 30 iş ile oluşturulan örneklerde gözlemlenmiştir. Bununla birlikte 20 iş ile oluşturulan örneklerde ayar zamanının yüksek ve düşük durumlarının ikisi için de en iyi olurlu çözümler, DK yaklaşımından elde edilmektedir. 30 iş ile oluşturulan örneklerde ise DK yaklaşımının ortalama ve en büyük EOÇYS değerlerinin düşük varyans durumunda daha düşük olduğu ortaya çıkmıştır. DK yaklaşımının ortalama EOÇYS değeri yüksek varyans durumunda %0,70, düşük varyans durumunda ise %0,17 olarak belirlenmiştir. Bununla birlikte bu örneklerde TB yaklaşımının ortalama EOÇYS değerleri ise sırayla %0,88 ve %0,18 olarak belirlenmiştir. Bu anlamda yüksek varyans durumunda DK yaklaşımının ortalama EOÇYS değeri TB yaklaşımından ortalama %25,71 daha iyiyken, düşük varyans durumunda bu değer

%5,88'e düşmektedir. Bu doğrultuda ASYS ve EOÇYS değerlerine göre, teslim tarihlerin farklı olduğu durumda DK yaklaşımının, ayar zamanı varyansının yüksek olduğu görece az iş ile oluşturulan örneklerde daha etkili bir yaklaşım olduğu sonucuna varılmıştır.

Teslim tarihleri tüm işler için aynı ve çizelgeleme periyodunun sonunda tanımlandığında matematiksel model ile tüm örnekler için olurlu bir çözüm üretilebilmiştir. Olurlu çözüm üretmenin kolaylaştığı bu örneklerde, genişleyen çözüm alanına bağlı olarak optimal çözümü belirlemek ise zorlaşmıştır. Bu örneklerde matematiksel model ile optimal çözümün belirlenebildiği toplam örnek sayısı önceki duruma göre yarıya düşmüştür. Bununla birlikte olurlu çözümlerle belirlenen üst sınır değerleri ile alt sınır değerleri arasında oldukça yüksek yüzde fark değerleri gözlemlenmiştir. Optimal çözümün belirlenemediği örneklerde matematiksel model için ASYS değerleri %100'ün altına düşmezken, DK yaklaşımı için ise en küçük ASYS değerleri sırayla %27,27 olarak ortaya çıkmıştır. Elde edilen üst sınır değerleri incelendiğinde, matematiksel model ile ayrıştırma algoritmasına göre daha güçlü olurlu çözümler türetilebilmesine rağmen alt sınır değerlerinin oldukça zayıf olması nedeniyle yüzde sapma değerinin yüksek olduğu gözlemlenmiştir. Bu anlamda ayrıştırma algoritmasının bu örnekler için alt sınır değeri belirlemek için kullanılabileceği sonucuna varılmıştır. Bu noktada en iyi olurlu çözümler için EOÇYS değerleri incelendiğinde ise, tüm örnekler için TB algoritmasının sapma değerlerinin sıfır olduğu, yani en iyi çözümün bu algoritma ile belirlenebildiği gözlemlenmektedir. Burada çözüm süresi de değerlendirildiğinde tavlama benzetimi algoritmasının bu örnekler için oldukça etkili bir yaklaşım olduğu sonucuna varılmıştır. Bununla birlikte ortalama EOÇYS değerlerine göre ayar süresinin varyansının azalmasıyla, TB algoritmasının DK yaklaşımı üzerindeki üstünlüğünün görece azaldığı yorumu yapılmıştır.

Bu çalışmanın devamında problem tamamen kısıt programlama yaklaşımıyla çözülerek, elde edilen sonuçlar matematiksel programlama ve matematiksel programlama ile kısıt programlama yaklaşımının birlikte değerlendirildiğinde ayrıştırma yaklaşımından elde edilen sonuçlarla karşılaştırılabilir. Burada geliştirilen ayrıştırma algoritmasının etkinliği olurlu çözüm alanının genişlemesi ile azalmaktadır. Gelecekte optimal çözümü kesmeyecek, fakat görece güçlü optimallik kesilerinin tanımlanmasına yönelik çalışmalar yapılabilir. Benzer şekilde tavlama benzetimi algoritmasında komşu türetmek adına yapılan rasgele ikili değişikliklerin etkinliğini arttırmak adına probleme yönelik üstünlük kuralları tanımlanabilir. Son olarak ele alınan problemin, paralel makineler için veya akış ve iş atölyeleri için genişletilmesi de gelecekte yapılabilecek çalışmalar arasındadır.

KAYNAKLAR

[1] **A. Allahverdi, J. Gupta ve T. Aldowaisan,** A review of scheduling research involving setup considerations, *Omega*, 27, 219-239, 1999.

[2] **A. Allahverdi ve H. Soroush,** The significance of reducing setup times/setup costs, *European Journal of Operational Research*, 187, 978-984, 2008.

[3] **A. Allahverdi, C. Ng, T. Cheng ve M. Kovalyov,** A survey of scheduling setup problems with setup times or costs, *European Journal of Operational Research*, 187, 985-1032, 2008.

[4] **D. Wortman,** Managing capacity: getting most from your firm's asset, *Industrial Engineering*, 24, 47-49, 1992.

[5] **J. Wilbrecht ve W. Prescott,** The influence of setup time on job shop performance, *Management Science*, 16, B274-B280, 1969.

[6] **L. Krajewski, B. King, L. Ritzman ve D. Wong,** Kanban, MRP and shaping the manufacturing enviroment, *Management Science*, 33, 39-57, 1987.

[7] **S. Trovinger ve R. Bohn,** Setup time reduction for electronics assembly: Combining simple (SMED) and IT-based methods, *Production and Operations Management*, 14, 205-217, 2005.

[8] **J. Loveland, S. Monkman ve D. Morrice,** Dell uses a new production scheduling algorithm to accomodate increased production variety, I*nterfaces*, 37:3, … 209-219, 2007.

[9] **A. Allahverdi,** The third comprehensive survey on scheduling with setup times/costs, *European Journal of Operational Research*, 246, 345-378, 2015.

[10] **M. Pinedo,** Scheduling: Theory, algorithms and systems, Prentice Hall: Springer Science Business Media , 2008.

[11] **K. Ying, S. Lin ve C. Huang,** Sequencing single-machine tardiness problems with sequence dependent setup times using an iterated greedy heuristic, *Expert Systems with Application*, 36, 7087-7092, 2009.

[12] **G. Ragatz,** A branch-and-bound method for minimum tardiness sequencing on a single processor with sequence-dependent setup time, *In:Proceeding: twenty-fourth annual meeting of the Decisions Sciences Institute*, 1375-1377, 1993.

[13] **P. Rubin ve G. Ragatz,** Scheduling in a sequence dependent setup enviroment with genetic search, *Computers and Operations Research*, 22:1, 85-99, 1995.

[14] **K. Tan ve R. Narasimhan,** Minimizing tardiness on a single processor with sequence-dependent setup times: A simulated annealing approach, *Omega*, 25;6, 619-634, 1997.

[15] **K. Tan, R. Narasimhan, P. Rubin ve G. Ragatz,** A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times, *Omega,* 28, 313-326, 2000.

[16] **G. Zobolas, C. Tarantilis ve G. Ioannou,** Extending capacity planning by positive lead times and optional overtime, earliness and tardiness for effective master production scheduling, *International Journal of Production Research*, 46:12, 3359-3386, 2008.

[17] **C. Akkan,** Overtime scheduling: An application in finite-capacity real-time scheduling, *The Journal of the Operational Research Society,* 47:9, 1137-1149, 1996.

[18] **B. Yang, J. Geunes ve W. O'Brien,** Overtime scheduling: A heuristice approach for minimizing weighted tardiness and overtime costs in single resource scheduling, *Computers and Operations Research,* 31, 1273-1301, 2004.

[19] **F. Jaramillo ve M. Erkoc,** Minimizing total weighted tardiness and overtime costs for single machine preemptive scheduling, *Computers and Industrial Engineering,* 107, 109-119, 2017.

[20] **M. Yoda, T. Eguchi ve T. Murayama,** Job shop scheudling for meeting due dates and minimizing overtime using genetic algorithm incorporating new priority rules, *Journal of Advanced Mechanical Design, Systems and Manufacturing,* 8:5, 2014.

[21] **N. Freeman, J. Mittenthal ve S. Melouk,** Parallel-machine scheduling to minimize overtime and waste costs, *IIE Transactions,* 46:6, 601-618, 2014.

[22] **J. Benders,** Partitioning procedures for solving mixed-variables programming problems, *Numerische Mathematik,* 4:1, 238-252, 1962.

[23] **A. Geoffrion,** Generalized Benders Decomposition, *Journal of Optimization Theory and Application,* 10:4, 237-260, 1972.

[24] **Z. Taşkın,** Benders Decomposition, Wiley, 2010.

[25] **R. Rahmaniani, T. Crainic, M. Gendreau ve W. Rei,** The Benders Decomposition Algorithm: A literature review, *European Journal of Operations Research,* 259, 801-817, 2016.

[26] **J. Hooker ve G. Ottoson,** Logic-based Benders decomposition, *Mathematical Programming,* 96, 33-60, 2003.

[27] **M. Fazel-Zarandhi ve J. Beck,** Using Logic-based Benders decomposition to solve the capacity and distance-constrained plant location problem, *INFORMS Journal on Computing,* 24:3, 387-398, 2012.

[28] **D. Wheatley, F. Gzara ve E. Jewkes,** Logic-based Benders decomposition for an inventory-location problem with service constraints, *Omega,* 55, 10-23, 2015.

[29] **J. Hooker,** Planning and scheduling by logic-based Benders decomposition *Operations Research,* 55:3, 588-602, 2007.

[30] **T. Tran, A. Araujo ve J. Beck,** Decomposition methods for the parallel machine scheduling problem with setups, *INFORMS Journal on Computing,* 28:1, 83-95, 2016.

[31] **E. Çoban ve J. Hooker,** Single-facility scheduling by logic-based Benders decomposition, *Annals of Operations Research,* 210, 245-272, 2013.

[32] **S. Heist,** *A comparison of constraint programming and integer programming for an industrial planning problem*, 2003.

[33] **B. Smith, S. Brailsford, P. Hubbard ve H. Williams,** The progressive part problem: Integer linear programming and constraint programming compared, *Constraints,* 1, 119-138, 1996.

[34] **K. Darby-Dowman, J. Little, G. Mitra ve M. Zaffalon,** Constraint logic programming and integer programming approaches and their collaboration in solving assignment scheduling problem, *Constraints,* 1, 245-264, 1997.

[35] **V. Jain ve I. Grossmann,** Algorithms for hybrid MILP/CP models for a class of optimization problems, *INFORMS Journal on Computing,* 13, 1, pp. 258-276, 2001.

[36] **E. Talbi,** Single solution based metaheuristics, *Metaheuristics from design to implementation*, John Wiley & Sons, 2009, 126-139.

[37] **N. Metropolis, A. Rosenbulth, M. Rosenbulth, A. Teller ve E. Teller,** Equation of state calculations by fast computing machines, *Journal of Chemical Physics*, 21, 1087-1092, 1953.

[38] **S. Kirkpatrick, C. Gelatt, M. Rosenbulth ve M. Vecchi,** Optimization by simulated annealing, *Science*, 220:4598, 671-680, 1983.

[39] **V. Cerny,** A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Apllications,* 45, 41-51, 1985.

[40] **M. Park ve Y. Kim,** Search Heuristics for a Parallel Machine Scheduling Problem with Ready Times and Due Dates, *Computers & Industrial Engineering*, 33, 793-796, 1997.

[41] **S. Radhakrishnan ve J. Ventura,** Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times, *Internatiınal Journal of Production Research*, 38:10, 2233-2252, 2000.

[42] **D. Kim, K. Kim, W. Jang ve F. Chen,** Unrelated parallel machine scheduling with setup times using simulated annealing, *Robotics and Computer Integrated Manufacturing*, 18, 223-231, 2002.

[43] **E. Thorsteinsson,** Branch-and-Check: A hybrid framework integrating mixed integer programming and constraint logic programming, *Proceedings of the Seventh International Conference on Principles and Practice of Constraint Programming (CP2001),* 16-30, 2001.

[44] **J. Beck,** Checking up on Branch-and-Check, *Proceedings of the Sixteenth International Conference on Principles and Practice of Constraint Programming (CP2010),* 84-98, 2010.

[45] **Differences between uset cuts and lazy constraints,** [Çevrimiçi]. Available: https://www.ibm.com/support/knowledgecenter. [Erişildi: Eylül 2016].

EKLER

EK 1: 10 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{10} \cdot {}^{\boldsymbol{N}}/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	Optimal Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	27,97	0,00	0,00	3,00
		Orta (0,89)	24,90	0,29	0,86	2,00
		Düşük (0,79)	24,67	0,57	0,86	1,00
	Orta (0,1)	Yüksek (0,99)	28,17	0,00	0,00	3,00
		Orta (0,89)	25,93	0,00	0,00	3,00
		Düşük (0,79)	24,67	0,95	1,14	0,00
	Düşük (0,01)	Yüksek (0,99)	28,30	0,00	0,00	3,00
		Orta (0,89)	25,53	0,29	0,86	2,00
		Düşük (0,79)	24,73	0,00	0,00	3,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	28,10	0,00	0,00	3,00
		Orta (0,89)	25,17	0,29	0,86	2,00
		Düşük (0,79)	24,40	0,57	0,86	1,00
	Orta (0,1)	Yüksek (0,99)	28,37	0,00	0,00	3,00
		Orta (0,89)	27,17	0,57	0,86	1,00
		Düşük (0,79)	25,77	0,57	0,86	1,00
	Düşük (0,01)	Yüksek (0,99)	28,80	0,00	0,00	3,00
		Orta (0,89)	26,00	0,29	0,86	2,00
		Düşük (0,79)	26,13	0,57	0,86	1,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	28,30	0,00	0,00	3,00
		Orta (0,89)	26,70	0,57	0,86	1,00
		Düşük (0,79)	24,57	0,76	1,43	1,00
	Orta (0,1)	Yüksek (0,99)	28,00	0,00	0,00	3,00
		Orta (0,89)	24,63	0,86	0,86	0,00
		Düşük (0,79)	24,87	0,86	0,86	0,00
	Düşük (0,01)	Yüksek (0,99)	27,80	0,00	0,00	3,00
		Orta (0,89)	24,90	0,57	0,86	1,00
		Düşük (0,79)	24,47	0,57	0,86	1,00

EK 2: 10 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{15} \cdot {}^{N}\!/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	Optimal Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	28,40	0,00	0,00	3,00
		Orta (0,89)	25,43	0,29	0,86	2,00
		Düşük (0,79)	24,60	0,57	0,86	1,00
	Orta (0,1)	Yüksek (0,99)	29,90	0,00	0,00	3,00
		Orta (0,89)	26,97	0,57	0,86	1,00
		Düşük (0,79)	24,83	0,57	0,86	1,00
	Düşük (0,01)	Yüksek (0,99)	29,50	0,00	0,00	3,00
		Orta (0,89)	25,67	0,57	0,86	1,00
		Düşük (0,79)	24,77	0,29	0,86	2,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	27,97	0,00	0,00	3,00
		Orta (0,89)	24,90	0,86	0,86	0,00
		Düşük (0,79)	24,77	0,86	0,86	0,00
	Orta (0,1)	Yüksek (0,99)	28,40	0,00	0,00	3,00
		Orta (0,89)	25,00	0,57	0,86	1,00
		Düşük (0,79)	24,40	0,29	0,86	2,00
	Düşük (0,01)	Yüksek (0,99)	29,33	0,00	0,00	3,00
		Orta (0,89)	25,97	0,57	0,86	1,00
		Düşük (0,79)	24,40	0,86	0,86	0,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	27,67	0,00	0,00	3,00
		Orta (0,89)	25,10	0,29	0,86	2,00
		Düşük (0,79)	24,90	0,86	0,86	0,00
	Orta (0,1)	Yüksek (0,99)	28,17	0,00	0,00	3,00
		Orta (0,89)	25,10	0,86	0,86	0,00
		Düşük (0,79)	24,27	0,57	0,86	1,00
	Düşük (0,01)	Yüksek (0,99)	28,13	0,00	0,00	3,00
		Orta (0,89)	25,13	0,29	0,86	2,00
		Düşük (0,79)	24,77	0,86	0,86	0,00

EK 3: 20 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{10} \cdot {}^{\boldsymbol{N}}/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	Optimal Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	34,13	0,17	0,25	1,00
		Orta (0,89)	27,53	0,54	0,87	0,00
		Düşük (0,79)	27,80	0,96	1,37	0,00
	Orta (0,1)	Yüksek (0,99)	32,97	0,08	0,25	2,00
		Orta (0,89)	29,60	0,62	0,75	0,00
		Düşük (0,79)	27,13	0,83	1,25	0,00
	Düşük (0,01)	Yüksek (0,99)	37,27	0,12	0,37	2,00
		Orta (0,89)	27,67	0,75	1,75	0,00
		Düşük (0,79)	27,23	1,62	2,00	0,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	31,53	0,25	0,25	0,00
		Orta (0,89)	27,43	0,29	0,37	0,00
		Düşük (0,79)	26,33	1,29	2,00	0,00
	Orta (0,1)	Yüksek (0,99)	34,73	0,29	0,37	0,00
		Orta (0,89)	27,50	0,50	1,00	0,00
		Düşük (0,79)	26,57	2,04	3,62	0,00
	Düşük (0,01)	Yüksek (0,99)	34,00	0,00	0,00	3,00
		Orta (0,89)	27,73	0,96	1,37	0,00
		Düşük (0,79)	27,40	0,75	1,25	0,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	30,80	0,21	0,37	1,00
		Orta (0,89)	27,47	0,79	1,12	0,00
		Düşük (0,79)	25,93	2,91	3,99	0,00
	Orta (0,1)	Yüksek (0,99)	32,00	0,29	0,37	0,00
		Orta (0,89)	27,70	0,62	1,25	0,00
		Düşük (0,79)	26,07	2,20	2,99	0,00
	Düşük (0,01)	Yüksek (0,99)	34,23	0,00	0,00	3,00
		Orta (0,89)	27,80	0,46	0,87	0,00
		Düşük (0,79)	26,97	0,50	1,00	0,00

EK 4: 20 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{15} \cdot {}^{N}\!/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	Optimal Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	31,73	0,12	0,25	1,00
		Orta (0,89)	27,03	1,04	1,50	0,00
		Düşük (0,79)	26,00	2,08	3,62	0,00
	Orta (0,1)	Yüksek (0,99)	33,33	0,00	0,00	3,00
		Orta (0,89)	27,00	0,37	0,50	0,00
		Düşük (0,79)	27,03	1,25	1,62	0,00
	Düşük (0,01)	Yüksek (0,99)	32,73	0,25	0,50	1,00
		Orta (0,89)	28,10	0,58	1,75	2,00
		Düşük (0,79)	26,50	34,37	100,00	0,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	31,00	0,17	0,50	2,00
		Orta (0,89)	26,77	1,75	2,37	0,00
		Düşük (0,79)	25,47	2,29	3,37	0,00
	Orta (0,1)	Yüksek (0,99)	32,53	0,21	0,37	1,00
		Orta (0,89)	27,63	1,54	2,00	0,00
		Düşük (0,79)	26,27	1,75	2,00	0,00
	Düşük (0,01)	Yüksek (0,99)	34,67	0,12	0,37	2,00
		Orta (0,89)	29,00	1,66	2,74	0,00
		Düşük (0,79)	27,40	0,91	1,50	1,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	30,17	0,08	0,25	2,00
		Orta (0,89)	29,90	0,83	1,25	0,00
		Düşük (0,79)	25,50	1,66	2,74	0,00
	Orta (0,1)	Yüksek (0,99)	30,23	0,17	0,50	2,00
		Orta (0,89)	28,83	1,08	1,75	0,00
		Düşük (0,79)	26,37	2,33	3,12	0,00
	Düşük (0,01)	Yüksek (0,99)	31,57	0,00	0,00	3,00
		Orta (0,89)	26,77	0,42	0,75	0,00
		Düşük (0,79)	25,80	1,66	2,12	0,00

EK 5: 30 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{10} \cdot \mathbf{N}/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	İyi Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	35,60	0,97	1,45	0,00
		Orta (0,89)	28,60	4,55	6,63	0,00
		Düşük (0,79)	28,20	6,56	9,52	0,00
	Orta (0,1)	Yüksek (0,99)	38,20	0,76	1,24	0,00
		Orta (0,89)	29,00	3,59	4,14	0,00
		Düşük (0,79)	28,70	6,00	7,66	0,00
	Düşük (0,01)	Yüksek (0,99)	39,30	1,24	2,07	1,00
		Orta (0,89)	29,70	4,00	4,55	0,00
		Düşük (0,79)	29,10	4,83	4,97	0,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	34,30	1,79	3,93	1,00
		Orta (0,89)	28,20	5,45	6,42	0,00
		Düşük (0,79)	27,40	6,00	9,73	0,00
	Orta (0,1)	Yüksek (0,99)	36,60	2,07	2,90	0,00
		Orta (0,89)	28,60	3,24	4,55	0,00
		Düşük (0,79)	28,00	4,28	5,18	0,00
	Düşük (0,01)	Yüksek (0,99)	37,20	1,38	1,86	0,00
		Orta (0,89)	28,80	2,90	3,31	0,00
		Düşük (0,79)	28,50	4,21	6,00	0,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	32,70	2,14	3,93	0,00
		Orta (0,89)	28,10	5,04	8,70	0,00
		Düşük (0,79)	26,60	6,14	9,32	0,00
	Orta (0,1)	Yüksek (0,99)	34,30	0,76	2,28	2,00
		Orta (0,89)	28,00	3,59	4,35	0,00
		Düşük (0,79)	27,20	5,04	7,04	0,00
	Düşük (0,01)	Yüksek (0,99)	35,80	1,59	1,66	0,00
		Orta (0,89)	28,50	3,66	3,93	0,00
		Düşük (0,79)	27,30	6,28	9,32	0,00

EK 6: 30 iş ile oluşturulan örnek için düşük denge noktası değeri ($\mathbf{15} \cdot {}^{N}/_{\mathbf{10}}$) ile yapılan parametre analizi

Başlangıç Sıcaklık	Son Sıcaklık	Soğuma Oranı	Ortalama Çözüm Süresi (saniye)	Ortalama Sapma (%)	En Fazla Sapma (%)	İyi Çözüm Adeti
Yüksek (99999)	Yüksek (1)	Yüksek (0,99)	39,00	0,97	1,45	0,00
		Orta (0,89)	31,70	4,55	6,63	0,00
		Düşük (0,79)	28,70	6,56	9,52	0,00
	Orta (0,1)	Yüksek (0,99)	43,30	0,76	1,24	0,00
		Orta (0,89)	31,70	3,59	4,14	0,00
		Düşük (0,79)	28,90	6,00	7,66	0,00
	Düşük (0,01)	Yüksek (0,99)	44,20	1,24	2,07	1,00
		Orta (0,89)	30,70	4,00	4,55	0,00
		Düşük (0,79)	30,50	4,83	4,97	0,00
Orta (9999)	Yüksek (1)	Yüksek (0,99)	35,80	1,79	3,93	1,00
		Orta (0,89)	29,00	5,45	6,42	0,00
		Düşük (0,79)	28,20	6,00	9,73	0,00
	Orta (0,1)	Yüksek (0,99)	38,40	2,07	2,90	0,00
		Orta (0,89)	29,30	3,24	4,55	0,00
		Düşük (0,79)	28,70	4,28	5,18	0,00
	Düşük (0,01)	Yüksek (0,99)	42,80	1,38	1,86	0,00
		Orta (0,89)	30,00	2,90	3,31	0,00
		Düşük (0,79)	29,50	4,21	6,00	0,00
Düşük (999)	Yüksek (1)	Yüksek (0,99)	34,30	2,14	3,93	0,00
		Orta (0,89)	28,50	5,04	8,70	0,00
		Düşük (0,79)	27,10	6,14	9,32	0,00
	Orta (0,1)	Yüksek (0,99)	37,20	0,76	2,28	2,00
		Orta (0,89)	29,00	3,59	4,35	0,00
		Düşük (0,79)	28,10	5,04	7,04	0,00
	Düşük (0,01)	Yüksek (0,99)	29,50	3,66	3,93	0,00
		Orta (0,89)	28,50	6,28	9,32	0,00
		Düşük (0,79)	39,00	0,97	1,45	0,00

ÖZGEÇMİŞ

Ad-Soyad **:** Zeynep BÜLBÜL

Uyruğu **:** T.C.

Doğum Tarihi ve Yeri **:** 31.08.1993 Ankara

E-posta **:** zeynepxbulbul@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** **:** 2015, TOBB Ekonomi ve Teknoloji Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü
- **Yüksek Lisans** **:** 2017, TOBB Ekonomi ve Teknoloji Üniversitesi, Endüstri Mühendisliği Anabilim Dalı, Endüstri Mühendisliği Programı

MESLEKİ DENEYİM VE ÖDÜLLER:

Yıl	**Yer**	**Görev**
2015-2017	TOBB Ekonomi ve Teknoloji Ünv.	Burslu Yüksek Lisans Öğr.

YABANCI DİL: İngilizce (Çok iyi), Fransızca (İyi)

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Bülbül, Z.,** Kakillioğlu, E.A., Gültekin, H., 2016. Fazla mesai ve sıra bağımlı ayar zamanı ile makine çizelgeleme problemi, Bildiri: Yöneylem Araştırması ve Endüstri Mühendisliği (YAEM) 36. Ulusal Kongresi, 13-15 Temmuz, İzmir, Türkiye.
- Gültekin, H., **Bülbül, Z.,** 2017. Sıra bağımlı ayar zamanı ve fazla mesai ile makine çizelgeleme probleminin Benders ayrıştırma algoritması ile çözümü, Bildiri: Yöneylem Araştırması ve Endüstri Mühendisliği (YAEM) 37. Ulusal Kongresi, 5-7 Temmuz, İstanbul, Türkiye.