

**HAVAYOLLARI TAMİR BAKIM SERVİS SAĞLAYICILARI İÇİN
BİR ÜRETİM ÇİZELGELEME MODELİ VE ANALİZİ**

DOĞUKAN HAZAR ÜLKER

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ARALIK 2015

ANKARA

Fen Bilimleri Enstitü onayı

Prof. Dr. Osman EROĞUL

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Prof. Dr. Tahir HANALIOĞLU

Anabilim Dalı Başkanı

Doğukan Hazar ÜLKER tarafından hazırlanan HAVAYOLLARI TAMİR BAKIM SERVİS SAĞLAYICILARI İÇİN BİR ÜRETİM ÇİZELGELEME MODELİ VE ANALİZİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Kadir ERTOĞRAL

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Yrd. Doç. Dr. Murat ARIKAN

Üye: Yrd. Doç. Dr. Salih TEKİN

Üye: Doç. Dr. Kadir ERTOĞRAL

TEZ BİLDİRİMİ

Tez içindeki tüm bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yaptığımı bildiririm.

.....

Doğukan Hazar ÜLKER

Üniversitesi : TOBB Ekonomi ve Teknoloji Üniversitesi
Enstitüsü : Fen Bilimleri
Anabilim Dalı : Endüstri Mühendisliği
Tez Danışmanı : Doç. Dr. Kadir ERTOĞRAL
Tez Türü ve Tarihi : Yüksek Lisans- Aralık 2015

Doğukan Hazar ÜLKER

HAVAYOLLARI TAMİR BAKIM SERVİS SAĞLAYICILARI İÇİN BİR ÜRETİM ÇİZELGELEME MODELİ VE ANALİZİ

ÖZET

Havacılık sektöründe bakım onarım (BO) uçakların güvenli ve devamlı uçuşlarının sağlanması için regülasyona tabi tutulan zorunlu bir faaliyettir. Gerek ticari gerekse askeri uçaklar için belirli aralıklarla bakıma girme zorunluluğu vardır. BO faaliyetleri uçuş firmaları için yakıttan sonra en çok maliyet yaratan alandır. BO sistemlerini klasik üretim sistemlerinden ayıran önemli bir faktör bazı parça ya da modüllerin devinimli (rotatable) envanter şeklinde kullanılmasıdır. Devinimli envanter durumunda, BO işletmesine servis için gelen uçağın ilgili modülü BO'nun takas envanterinde bulunan uçuşa hazır modülle değiştirilir. Bu durumda müşteri olan havacılık şirketi nerdeyse anlık değişimle uçuşa ara vermeden faaliyetlerini sürdürür. BO şirketi takas yaptığı devinimli envanter türü modülü bakım onarım sürecinden geçirir ve sonra tekrar takasta kullanmak üzere gelecek talepler için envanterinde saklar. BO şirketi uçakların devinimli modül takas zamanlarını ve bakım onarım çizelgesini yaparken, uçakların en son bakım geçirdikleri zaman dikkate alındığında regülasyonlara göre bir sonraki bakım için gelebilecekleri en son tarihe veya bu tarihten önce ancak yakın bir zamanda gelip servis almalarını sağlamaya çalışmalıdır. Çünkü vaktinden erken gelişleri havacılık şirketleri ekonomik açıdan tercih etmeyecektir. BO şirketi için sabit bir tamir bakım kapasitesi ve limitli envanter üretim çizelgeleme üzerindeki önemli kısıtlardır. Bu çalışmada yukarıda açıklanan problem toplam erken servis alımların en azlanması amacıyla, belirli bir başlangıç stoğu, kısıtlı üretim kapasitesi, çok tipli devinimli envanter, ve sonlu planlama ufku varsayımları altında modellenmiş, sayısal çözümler yardımıyla problemin analizi yapılmıştır. Devinimli envanter başlangıç seviyesini veya proses kapasitesini bir birim artırmanın azalan marjinal faydası ortaya konulmuş ve bu faydanın proses süresiyle ve proses kapasitesiyle ilişkisi gösterilmiştir. Buna ek olarak problemin daha hızlı çözülebilmesi için sezgisel bir yaklaşım önerilmiş ve bu sezgisel

yaklaşımın iyileştirilmesi için çalışmalar yapılmıştır. Yapılan uygulamalar sonucunda sezgisel yaklaşımdan elde edilen çözümler, optimal değerler ile karşılaştırılıp önerilen yaklaşımın performansı ortaya konulmuştur.

Anahtar Kelimeler: Üretim çizelgeleme, tamir bakım, devinimli stok, data karıştırma meta sezgisel yaklaşımı

University : **TOBB University of Economy and Technology**
Institute : **Institute of Natural and Applied Sciences**
Science Program : **Industrial Engineering**
Supervisor : **Assoc. Prof. Dr. Kadir ERTOĞRAL**
Thesis Type and Date : **M.Sc. - December 2015**

Doğukan Hazar ÜLKER

**A PRODUCTION SCHEDULING MODEL AND ANALYSIS FOR THE
MAINTENANCE REPAIR AND OVERHAUL SERVICE PROVIDERS**

ABSTRACT

Maintenance, repair and overhaul (MRO) are regulated and necessary activities for the airline sector to ensure the safety and continuity of flights. Both military and commercial aircrafts must go through MRO at regulated intervals. MRO activities constitute the second largest cost component for the airline companies after the fuel. One of the important differences between MRO systems and traditional production systems is the use of rotatable inventory for some expensive parts or modules. In the rotatable inventory case, the arriving part for service is directly exchanged with a ready inventory. Thus, the customer airliner continues its flights with almost no break time. The MRO company overhauls the exchanged equipment and places it as ready inventory to be used for future demands. As MRO companies schedule the exchanges of rotatable inventory and overhauls, considering the last date of service, they should try to bring in the aircrafts as close as possible to their regulated final dates for overhauls, since the earlier arrivals will not be preferred economically by the airline companies. The limited production capacity and inventory are the main constraints for scheduling the production. In this study, the problem is modeled with the assumptions of limited initial inventory and production capacity, multiple type rotatable inventory and finite planning horizon, with the objective of minimizing the total earliness. We tested the problem thorough detailed numerical experiments. We have shown the decreasing marginal benefits for an additional unit of initial rotatable inventory or additional process line and the relationship between this benefit and process time and process capacity. In addition to this, a heuristic approach is proposed to solve the problem faster and some implementations are applied on the offered heuristic for improvement. As a result of the applied implementation on the proposed heuristic, the solutions of the proposed heuristic and optimal solutions are compared with each other to determine the performance of the offered heuristic.

Key Words: Production scheduling, maintenance repair and overhaul, rotatable inventory, perturbation heuristic

TEŞEKKÜR

TOBB Ekonomi ve Teknoloji Üniversitesi'nde yapmış olduğum bütün çalışmalarda yanımda olan ve desteklerini esirgemeyen sayın danışman hocam Doç. Dr. Kadir ERTOĞRAL'a, fikirleri ile tezimin ilerlemesine yardımcı olan Miami Üniversitesi öğretim üyesi sayın hocam Doç. Dr. Murat ERKOÇ'a, kıymetli tecrübelerinden yararlandığım ve her türlü soruma cevap sağlayan TOBB Ekonomi ve Teknoloji Üniversitesi Endüstri Mühendisliği öğretim üyelerine teşekkürlerimi sunarım.

Her zaman yanımda duran, desteklerini esirgemeyen, bana her türlü uygun ortamı sağlayan, yeri geldiğinde benimle birlikte makale okuyan, hayata umutla bakmamı sağlayan sevgili annem İclal ÜLKER'e, hayat görüşüyle, fikirleriyle, duruşuyla örnek aldığım, tavsiyelerini hayatım boyunca dikkate alacağım babam Muhammet Akif ÜLKER'e, her zaman beni destekleyen, bana verdiği büyük sevgiyle hayatıma ışık tutan, mutluluğumun temel kaynağı, sevgili eşim Hande CANAN ÜLKER'e ve çok değerli ailesine teşekkür ederim.

Hayatımın her döneminde olduğu gibi yüksek lisansım süresince beni destekleyen kardeşlerime, yine lisansüstü eğitimin sırasında desteklerini hiçbir konuda esirgemeyen arkadaşlarım Uğur YILDIZ, Yaşar KUMRU ve Mustafa BAŞBOĞA'ya çok teşekkür ederim.

Tez dönemim esnasında bana anlayışıyla destek veren, eğitime verdiği önemle yüreğimde kocaman bir yer edinen sevgili abim ve aynı zamanda şirketimizin genel müdürü Erdal BURAK'a ve onun değerli ortağı Fettah KÖYSÜREN'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR	viii
İÇİNDEKİLER.....	ix
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xi
KISALTMALAR.....	xii
SEMBOL LİSTESİ.....	xiii
1. GİRİŞ	1
2. PROBLEM TANIMI VE MODELLEME YAKLAŞIMI	5
3. LİTERATÜR.....	7
3.1. Bakım Onarım.....	7
3.2. Data Karıştırma (Perturbation) Meta Sezgisel Yaklaşımı	10
4. PROBLEMİN MATEMATİKSEL MODELİ	12
5. PROBLEMİN ÖN SAYISAL ANALİZİ	15
6. ÇÖZÜM YAKLAŞIMLARI	21
6.1. Geriye Doğru Çizelgeleme Sezgiseli	22
7. DATA KARIŞTIRMA META SEZGİSELİ	26
7.1. Sezgisel Akış Şeması ve Açıklama	26
7.2. İyileştirme Adımlı Veri karıştırma Meta Sezgiseli.....	29
7.3. Toplu sonuçlar ve karşılaştırma	31
8. SONUÇLAR.....	35
KAYNAKLAR	37
EKLER.....	39
ÖZGEÇMİŞ	115

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Devinimli envanter çalışmalarının karşılaştırılması	9
Çizelge 5.1. Koşturulan senaryolar ve sonuçlar	16
Çizelge 6.1. Geriye doğru çizelgeleme yaklaşımının sonuçlarının problem tipine göre yüzde sapması.....	24
Çizelge 7.1. Data karıştırma sezgiseli sonrası optimal sonuçlardan yüzde sapmalar	28
Çizelge 7.2. İyileştirme adımlı data karıştırma sezgiseli sonradan optimalden yüzde sapmalar	30
Çizelge 7.3. Tablosallaştırılmış değerler	32

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Devinimli stok akış sistemi.....	6
Şekil 5.1. $P=25,15,10$ olması durumunda optimal çözüm karşılaştırmaları	18
Şekil 5.2. $P=30,20,15$ olması durumunda optimal çözüm karşılaştırmaları	18
Şekil 5.3. $P=35,25,20$ olması durumunda optimal çözüm karşılaştırmaları	19
Şekil 6.1. Algoritma akış şeması	22
Şekil 7.1. Data karıştırma sezgiseli akış şeması	26

KISALTMALAR

Kisaltmalar	Açıklama
MRO	Maintenance Repair Overhaul
Vd.	Ve diğerleri
BO	Bakım onarım

SEMBOL LİSTESİ

Setler

T : Planlama ufkundaki periyot sayısı

J : Farklı tip devinimli envanter sayısı

$N(j)$: j tipi envanter için takas talep edilen periyotlar kümesi

İndisler

i : Talep indisi $i, i \in N(j)$.

j : Tip indisi, $j= 1,2,\dots,J$.

t : Periyot indisi, $t=1,2,\dots,T$.

Parametreler

s_j : j tipi envanterden başlangıç stok miktarı

k : Paralel eşdeğer üretim hattı/takımı sayısı

p_j : j tipi envanter için periyot sayısı cinsinden tamir bakım süresi

d_{ji} : j tip envanterin i . talebi için son değişim günü

w_{ji} : j tip envanterin, i . talebine karşılık gelen müşteri için ağırlık değeri.

R_t : içinde bulunduğumuz t periyodunda kullanılan bakım hattı sayısı

t^l : $t^l \leq t$; $R_{t^l} < K$ olduğu en büyük zaman

K : bakım için kullanılan paralel hat sayısı

Karar değişkenleri

X_{jit} : $\begin{cases} 1 & \text{Eğer } j \text{ tip envanterin } i. \text{ talebini } t \text{ anında takasla karşılanıyorsa} \\ 0 & \text{Aksi halde} \end{cases}$

Y_{jt} : j tip envanterden t periyodu başında bakıma başlayanlarının sayısı.

H_{jt} : j tip envanterden t periyodu sonunda değişime hazır stok miktarı.

B_{jt} : j tip envanterden t periyodu başı itibariyle bakım bekleyen stok miktarı.

1. GİRİŞ

Endüstriyel bakım faaliyetleri herhangi bir ekipmanın devamlılığının sağlanması amacıyla devam ettirilen tüm teknik ve buna bağlı olarak yürütülen yönetim uygulamalarının kombinasyonu olarak tanımlanabilir. Bu faaliyetler sonucunda ürünün fonksiyonel tutulması, güvenli bir şekilde çalışmasının sağlanması ve ömrün uzatılması temel hedefler arasındadır. Endüstriyel bakım faaliyetleri ile beraber ürünlerin kullanıldığı sektörlerdeki verimliliği artırılmış olup, bakım onarım faaliyetleri firmalara kısa vadede masraf gibi gözükmesine karşın uzun vadede kar sağlayan prosesler olarak karşımıza çıkmaktadır.

Özel olarak bakım onarım faaliyetlerini havacılık alanında inceleyecek olursak; havacılıkta bu faaliyetler regülasyona tabi tutulan zorunlu aktiviteler olarak karşımıza çıkmaktadır. Bu zorunlu aktiviteler ile beraber bakım onarım faaliyetlerinin temel hedeflerinde yer alan üç temel kriterde sağlanmaktadır. Hem sivil hem de askeri amaçlı yapılan uçuşlarda kullanılan uçakların belirli periyotlarla bakıma girmesi gerekmektedir. Buradaki esas amaç parçaların sürekliliğini korumasını sağlayarak istenmeyen hadiselerin önüne geçmektir. Bu bahsedilen bakım-onarım periyodu uçuş zamanına ya da uçuş sayısına bağlı olarak belirlenmektedir. Bu iki kısıttan hangisi önce gerçekleşirse, diğer kısıta bakılmaksızın uçak bakıma girmek zorundadır. Bakımdan sonra bu iki kısıt sıfırlanır ve bir sonraki bakım için yine bu iki temel kriterden birinin gerçekleşmesi beklenir. Bu döngü sürekli kontrol edilerek bakım onarım faaliyetlerinin tekrarlanması gerekmektedir.

Bakım onarım faaliyetlerinin maliyetlerini göz önüne alırsak havacılık sektöründe hizmet veren firmalar için yakıttan sonraki en büyük gider olarak karşımıza çıktığını görebiliriz. Bu bakımdan havacılık sektöründe artan rekabetin bir sonucu olarak firmalar kaliteden ödün vermeden bakım onarım faaliyetlerinin maliyetlerini azaltarak yürütmek istemektedirler.

Maliyetlerle ilgili olarak Standard and Poors'un raporunda sadece 2009 yılında bakım onarım faaliyetleri için harcanan meblağın 111 milyar dolar civarında olduğunu görmekteyiz. Bu tutarın 62 milyar doları askeri bakım onarım faaliyetleri için, 42.7 milyar doları ticari amaçlı hava ulaşımı için ve 6,2 milyar doları diğer işletme alanlarında gerçekleşmiştir.[1]

Bu tür harcamalarının uzun vadede katma değer yaratması ile beraber firmalar bakım onarım faaliyetlerine git gide önem göstermiş ve bunun sonucunda konuyla alakalı yeni sektörler doğmuştur.

Büyük ticari hava yolu şirketleri bakım onarım faaliyetlerini kendi bünyelerinde gerçekleştirirken, sektörde bu alanda çalışmalarını yürüten bağımsız şirketlerde mevcuttur. Kendi bakım onarım faaliyetlerini yürüten havayolları arasında Air France/KLM ve Luftansa Technic bu sektörde hizmet veren firmalar arasındadır. Bu şirketler aynı zamanda üçüncü partilere de hizmet vererek bakım onarım maliyetlerini minimum düzeye çekmek için çalışmalarını sürdürmektedirler.

Geçtiğimiz on yıl içerisinde Türk Hava Yolları'da bakım onarım faaliyetlerinin yürütülmesi amacıyla Turkish Technic'i kurarak bu alanda önemli bir adım atmıştır. 2008 yılında Turkish Technic gelirinin %25'ini Türk Hava Yolları'na ait olmayan uçaklardan elde etmiştir[2]. 2014 yılına bakıldığında ise Turkish Technic bünyesinde 2766 bakım gerçekleştirmiş olup, bu bakımların 121'ini Türk Hava Yolları dışındaki hava yolu şirketleri için uygulamıştır[3]. Turkish Teknik 2014 yılı itibarı ile İstanbul ve Ankara'da bulunan 4 hangarı ve komponent atölyeleri ile 100'den fazla uluslararası müşterisine bakım hizmeti vererek bu sektörde öncü firmalar arasında yer almıştır.

Bakım onarım şirketleri için üç temel kriter olarak kalite, bakım süresi(turn-around time) ve bakım masrafı karşımıza çıkmaktadır. Bu gibi bakım onarım hizmeti veren şirketlerden hava yolları gün ve hafta bazında değil; saat-dakika bazında hizmet beklemektedirler[4]. Bunun temel sebebi ise uçağın bakımdayken kullanılamamasından dolayı hava yolu şirketi için gelir kaybına neden olmasıdır. Bu değer on bin dolarlardan başlayıp yüz binlerce dolara kadar ulaşabilmektedir. Uçuş firmalarının bu konudaki titiz tavrı sonucunda sektördeki bakım onarım faaliyetleri hızlı bir şekilde devam etmektedir. Anlaşılacağı üzere bakım onarım faaliyetleri

yalnız ürünün devamlılığı ve güvenlik açısından değil, ekonomik süreklilik açısından da önem arz etmektedir.

Bakım onarım faaliyetlerinde bahsedilen bakım süresinin kısaltılması için yapılacak olan çalışmalar kompleks ve özel optimizasyon tekniklikleri gerektiren bir problemdir. Karmaşık ve büyük ölçekli söküm, onarım, montaj süreci planlaması, rassal talep gelişleri, devinimli(rotatable) envanter yönetimi bu problemin temel zorlukları arasında başlıcalarıdır.

Devinimli envanter kullanımı, bakım onarım sistemlerini klasik üretim yöntemlerinden ayıran en önemli özelliştir. Devinimli envanter kullanımı genellikle pahalı parça veya modüller için geçerlilik göstermektedir. Bu kısımda takas politikası uygulanır. Klasik üretim sistemlerinde ise takas modeli pek uygulanan bir yaklaşım değildir. Bu uygulamanın temelinde; tamir ve bakım için sisteme gelen parçayı veya modülü bakım onarım şirketi bünyesine alması ve devinimli envanter stokunda bulunan parçayı veya modülü müşteriye vermesi yatmaktadır. Bakıma ihtiyacı olan parça gerekli işlemlerden geçirildikten sonra devinimli envanter stokuna eklenir. Sonuçta bu tür pahalı parça veya modüllerde envanter tedarik suretiyle değil, dönüşüme tabi tutularak yenilenmiş olur.

Görüldüğü üzere; malzeme akışı klasik üretim sistemlerinde çoğunlukla tek yönlü iken, devinimli stok kullanılan sistemlerde akış eşit miktarda stoka ve stoktan olmak üzere iki yönlüdür[5]. Bu da envanter yönetiminin kapalı dönüşümlü tedarik zinciri yapısında olduğunu göstermektedir[6]. 2007 yılında dünya çapında sadece ticari hava ulaşımı için 44 milyar dolarlık yedek parça envanteri bulundurulduğu rapor edilmiştir [7]. Bu uygulama özellikle son yıllarda hava yolu şirketlerinin, bakım onarım servisi yapan bağımsız firmalara stok bulundurmayı şart koşmasıyla sonuçlanmıştır[8]. Bu uygulama bakıma ihtiyaç duyan parçaların değişimini neredeyse anlık hale getirmiştir. Böyle bir süreç bakım onarım servisi sağlayan şirketler için dikkatli bir iş planlama ve envanter kontrolü gerektirir.

Bakım onarım şirketlerinin stoklarında her zaman envanter bulunmayabilir. Havacılık alanında bu parçaların ve modüllerin bakım sürelerinin regülasyona tabi tutulduğu düşünüldüğünde, bakım onarım hizmeti veren şirketlerin müşteri firmalara bakım onarım randevusunu çok dikkatli bir şekilde vermeleri gerekmektedir.

Bu tür randevu düzenlemesi ve devinimli stok planlaması önemli kısıtlar altında yapılmaktadır. Bakım onarım servisi yapan şirketler, belirli bir tip devinimli modül için birden fazla müşteriye hizmet vermektedir. Bu durumun varlığı ile talep koordinasyonunun her bir devinimli parça veya modül için birden fazla müşteri ile yapılması gerekmektedir. Havacılıktaki kurallara göre bakım onarım zamanı aşılamayacak bir kısıt olarak karşımıza çıkmaktadır. Parça maliyetlerinin yüksek oluşu nedeni ile bakım onarım şirketleri belirli sayıda parçayı devinimli envanter stokuna eklediğinden bazı müşterilerine bakım onarım süresi gelmemesine rağmen erken randevu vermektedirler. Havayolu şirketlerinin parça kullanımını maksimize etmek istemesi ve bakım onarım servisi yapan şirketlerin randevu tarihini aşamayacak olması ile taraflar üstünde çift taraflı bir baskı ortaya çıkmaktadır.

Klasik üretim sistemleri için planlama, çizelgeleme ve envanter kontrolüne yönelik optimizasyon çalışmaları ve uygulamaları literatürde ve pratikte çokça çalışılmış olduğu halde BO alanı nispeten ihmal edilmiş bir alan olarak dikkat çekmektedir. Bu çalışmamızda önerilen üretim çizelgeleme modeli bu alandaki boşluğu doldurmaya yönelik, literatürdeki az sayıdaki ilgili modelden birisidir. Çalışmamızda BO şirketlerinde iş çizelgeleme ve envanter kontrol probleminin entegre şekilde ele alan literatür için yeni bir model geliştirilmiştir. Model verilen bir başlangıç envanter seviyesine göre, bakım onarım son tarihleri baz alınarak birden çok tipte devinimli envanter modülü ve müşterileri içeren talep varışlarının ve modül tamir ve bakım çizelgelemesinin optimizasyonunu amaçlar. Optimizasyonun amacı envanter ve tamir bakım kapasite kısıtları altında toplam erken takasları minimize etmektir. Model farklı parametrelerle çalıştırılarak problem analiz edilmiştir. 2. bölümde problemin tanımı ve modelleme yaklaşımı verilmiştir. Bölüm 3'te literatür incelenmiştir ve yapılan çalışmanın literatürdeki yeri tespit edilmiştir. Bölüm 4'te problemin matematiksel modeli verilmiştir. Bölüm 5'te ön nümerik çalışma yapılmış ve sonuçları değerlendirilmiştir. Bölüm 6'da optimizasyon zamanının uzun sürmesi sebebiyle probleme daha hızlı sonuç veren bir çözüm yöntemi önerilmiştir. Bölüm 7'de önerilen çözüm yönteminin iyileştirilmesi için data karıştırma sezgiseli ile ilgili detaylar verilmiştir. Takip eden sonuç bölümünde çalışma özetlenip gelecek olası çalışmalara ışık tutulmuştur.

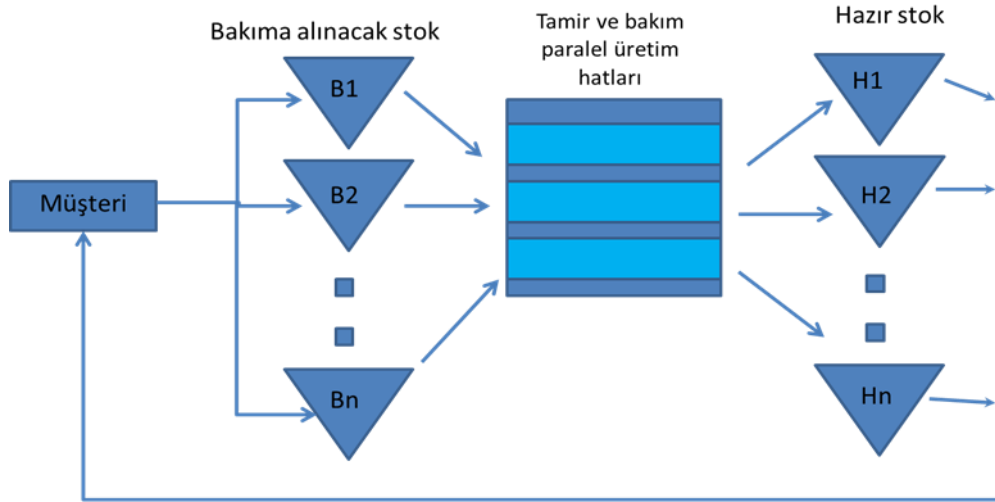
2. PROBLEM TANIMI VE MODELLEME YAKLAŞIMI

Bu problemde temel girdi olarak, belirli bir periyot içinde taleplerin karşılanabileceği son tarih ile ilgili bilgiler kullanılmıştır. Bu problemdeki temel amaç, devinimli envanter takaslarının mümkün olduğunca izin verilen son tarihlerde veya yakınında yapılmasını sağlamak ve buna bağlı olarak da takas için gelen taleplerin erken çağırılmasını minimize etmektir. Bu işlem yapılırken, ilgili parça veya modül için belirlenen son takas tarihinin aşılması devinimli stok dengesinin her periyotta korunması ve tamir-bakım proses kapasitesinin aşılması problemin esas kısıtlarıdır. Problemden periyotlar gün olarak belirlenmiş olup; sonlu periyotlu bir planlama ufku göz önüne alınmıştır. Periyodun birimi kullanıcı tarafından değiştirilebilir; örneğin hafta, ay veya yarım gün gibi alınması mümkündür. Buradaki önemli nokta talep tarihlerinin ve tamir bakım süresinin aynı birim periyot cinsinden olmasıdır.

Problemden her bir takas talebinin hangi gün gerçekleşeceğini takip etmek amacıyla ikili değer alan bir değişken kullanılmıştır. Temel karar ise her talebin ne zaman karşılanacağı (takas tarihi) ve her bir tip envanter için tamir bakımın ne zaman başlayıp biteceğinin belirlenmesidir. Bu kararlar her periyotta her bir tip devinimli envanterden kaç tanesinin tamir bakıma başladığını takip eden bir değişken yardımıyla modellenmiştir.

Probleme ilgili diğer bazı varsayımlarımız şunlardır; bu problemde tamir bakım kapasitesi eş değer paralel kaynaklar olarak göz önüne alınmıştır. Bu kaynaklar tamir bakım sistemi içerisinde işçi takımlarını veya üretim hatlarını temsil edebilir. Envanter tiplerine göre tamir bakım süreleri değişkenlik gösterir. Bununla beraber aynı tip envanter için tamir bakım süresi eşit ve sabittir. Bu kısımda üretim hattında meydana gelebilecek değişiklikler envanterin bakım süresini değiştirmeyecektir. Herhangi bir tamir bakım hattının bir ürünün bakım işlemine başlaması ile beraber ilgili bakım hattında o iş bitene kadar başka bir envanterin bakıma başlayamayacağı varsayılmıştır.

Devinimli stok kısmındaki envanter denge kısıtı problemi, klasik üretim sistemlerinden ayıran en önemli özelliktir. Planlama ufkunun başında her bir envanter tipinin belirli sayıda başlangıç stoku bulunduğu varsayılmıştır. Devinimli stok envanteri sistemde üç durumda bulunabilir; takas edilmeye hazır, tamir bakım sürecinde veya tamir bakım bekler durumda. Her takasta, takasa hazır bekleyen envanter bir azalır ve tamir bakım bekleyen envanter uçaktan çıkartılan modül nedeniyle bir artar. Ayrıca takasa hazır envanter, her periyot tamir bakımı biten envanter kadar artacaktır. Devinimli stokun hareketi ve sistem şekil 2.1’de gösterilmiştir. Burada B_j ve H_j sırasıyla bakım bekleyen ve takasa hazır, j . tip envanter seviyesini göstermektedir.



Şekil 2.1. Devinimli stok akış sistemi

3. LİTERATÜR

3.1. Bakım Onarım

BO servis optimizasyonu literatürde son yıllarda özellikle ağır sanayideki sermaye ve varlık artırımını ile araştırmacıların dikkatini çekmiştir. Önceki yıllarda dikkat daha çok klasik üretim sistemlerine verilmiştir. Bunun ana sebebi üretim sistemleri firmalara direk değer eklerken BO servisi üretim sistemlerinin devamlılığını sağlamak için yapılması gereken destek sınıfı kapsamında bir fonksiyon ve masrafa yol açtığı için kaçınılan bir işlem olagelmıştır. Rekabetin arttığı ve sürdürülebilirliğin önem kazandığı son yıllarda ise BO daha çok önemsenmeye başlamıştır. Bu alanda literatürdeki ilk çalışmalar daha çok bir firmanın kendi bakım faaliyetlerinin planlanmasına yöneliktir. Bu konuda yapılan çalışmalarının oldukça bilgilendirici bir özeti Dekker [9] tarafından rapor edilmiştir. Dekker, BO sektöründe karar vermeyi destekleyecek araç ve optimizasyon metotlarının seksenli ve doksanlı yıllarda sınırlı olduğunu belirtirken, gelişen teknik sistemler ve rekabet dolayısıyla gelecek yıllarda BO servis optimizasyona daha fazla önem verileceğini tahmin etmiştir. Bunu takip eden bir çalışmada Dekker ve Scarf [10] söz konusu alanda araştırmacıları bekleyen problem alanlarını tartışmışlardır. Bu konuda en önemli problemi şirketlerin operasyonlarını en az şekilde etkileyecek ve mevcut kapasiteyi en hesaplı şekilde kullandıracak bakım plan ve çizelgelemesi olarak tespit etmişlerdir ki bizim çalışmamızda önerilen model bu kapsamda düşünülebilir.

Son yıllarda BO sistemleri konusunda yapılan çalışmaların önemli bir kısmı ya yedek parça envanterine odaklanmakta ([11], [12]) veya stratejik düzlemde havacılık şirketleriyle BO firmaları arasındaki kontrat modellerini ele almaktadır ([1], [13], [14]). Şu ana kadarki tespitimize göre devinimli envanter kontrolünü ele alan ve bu problemi BO firması açısından operasyonel seviyede ilk inceleyen Rupp [15] olmuştur. Bu çalışmada talep gelişleri Poisson proses olarak modellenmiş ve stokastik sistemin analizi benzetim metodu kullanılarak gerçekleştirilmiştir. Problem

tamamen envanter odaklı olup talep yönetimi ve üretim çizelgeleme unsurları analize dahil edilmemiştir. Herhangi bir masraf analizi yapılmamış, servis seviyelerine odaklanılmıştır. Diğer bir çalışmada Romo ve Erkoc [16] rassal talep gelişleri altında devinimli envanter ve servis kapasite kararlarını gelir maksimizasyonunu amaçlayan bir modelle ele almış ve analitik çözümünü çalışmışlardır. Birçok küçük parça için BO şirketine talep gelişleri büyük ölçüde rassal olabilir. Fakat büyük devinimli modül bakım servislerinde talepler çoğunlukla randevu üzerinden olur ve genelde çok önceden bu devinimli modüllerin son BO servis tarihleri bilinir.

Bu tezde önerilen çalışmaya en yakın çalışmalar geçmişte Joo [17], Luh vd. [18] ve Arts vd. [19] tarafından yapılmıştır. Joo [17], problem ifadesi bölümünde bahsedilen tek tip devinimli modül türü problem için optimum talep çizelgelemesini çalışmıştır. Problemden amaç toplam erkenliliğin minimize edilmesidir. Envanter seviyesi veri olarak (exogenous) alınmış ve karar sürecine dahil edilmemiştir. Ayrıca çok tipli devinimli modül ve bunların paralel bakımının yol açtığı kapasite kısıtları da çalışmaya dahil edilmemiştir. Luh vd. [18] matematiksel modelleme kullanarak eş zamanlı çizelgeleme ve devinimli envanter problemini çalışmışlardır. Ancak önerilen çalışmadan farklı olarak talep gelişlerini veri olarak almışlar ve müşterilere randevu verme kararı modele dahil edilmemiştir. BO servisini konvansiyonel şekliyle ele almışlardır. Yani çalışmaları takaslama sistemi temelinde yapılmamıştır. Çizelgeleme daha çok servis faaliyetlerinin planlanmasına yöneliktir. Ayrıca ele alınan modelde servisin son tarihi geçmesi cezaya tabi olarak olurlu görülmüştür. Araştırmacılar Lagrangean gevşetme tekniğine bağlı çözüm algoritmaları önermektedirler. Daha öncede açıklandığı gibi son yıllarda BO firmaları takas politikaları izlemektedirler ve bu yüzden müşterilere randevu vererek talep gelişlerini çizelgelemektedirler [5]. Takasın ana amacı BO son tarihlerini geçmemeyi garantiye almaktır. Bu açıdan bu sistem altında uygulamada gecikmelere izin verilmez.

Yakın bir zamanda yapılan diğer bir çalışmada Arts vd. [19] eş zamanlı kapasite ve devinimli envanter kontrolü problemi için bütünleşik planlama modeli geliştirmişlerdir. Modelde uygulamaya paralel olarak, gecikmeye izin verilmemektedir. Ancak erkenlilik direkt olarak amaç fonksiyonuna dahil edilmemiştir. Kapasite iş gücü (adam-saat) bazında ele alınmıştır. Servis çizelgeleri bütünleşik planlama gereği ay bazında yuvarlanmıştır. Analiz daha çok üretim çizelgeleme için değil taktik seviyede kapasite planlamaya yöneliktir. Önerilen

çalışmada ise taleplerin tam tarihleri temel gün bazında alınmış, operasyonel bir model geliştirilerek toplam erkenlilik amaç fonksiyonu olarak kullanılarak üretim çizelgelenmesi hedeflenmiştir. Devinimli envanter içeren literatür aşağıdaki çizelgede özetlenmiştir.

Çizelge 3.1. Devinimli envanter çalışmalarının karşılaştırılması

	Detaylı BO ve takas çizelgele me	Birden çok tip envanter	Kapasite kısıtı	Erken gelişlerin en azlanması	Bütünleşik üretim ve kapasite planlama
Arts vd. [19]		x	x		x
Joo [17]	x		x	x	
Luh vd. [18]					
Romo ve Erkoc [16]	x			x	
Rupp [15]		x	x		
MacDonnell, M. and B. Clegg. 2011 [11]		x	x		
Bu çalışmamız	x	x	x	x	

3.2. Data Karıştırma (Perturbation) Meta Sezgisel Yaklaşımı

Bu yöntemdeki temel fikir, problemin temel sezgisel yönteminin avantajı ile yerel arama stratejilerinin avantajlarını kombineli olarak kullanmaktır. Problemin temel sezgisel yöntemi, verilen problem datasıyla problemin yapısını iyi olarak kullanmaktadır ve bu sezgisel yöntem tek bir çözüm üretmektedir. Diğer taraftan, yerel arama alternatif sonuçlar üretebilmek açısından iyi bir araç olarak karşımıza çıkmaktadır; ancak bunlar genellikle problemin yapısını görmezden gelmektedirler. Data karıştırma, bu ikisini birleştirerek problemin komşuluklarında alternatif çözümler elde etmektedir. Bu yöntem ilk olarak Storer, Wu ve Vaccari[20] tarafından önerilmiştir. Burada p_0 eldeki problemin data vektörü ve p_i 'de pertübasyon sonucunda elde edilen data vektörü konumundadır. P ise m pertübasyonun sonucunda elde edilen data vektör seti konumundadır.

$$P = \{p_i = p_0 + r_i, i = 1, 2, \dots, m\}$$

Yine burada r_i rastgele elde edilmiş pozitif ve negatif değerler içeren vektör konumundadır.

$$S = \{H(p_i), i = 1, 2, \dots, m\}$$

$H(p_i)$ ise data vektörü p_i kullanılarak sezgisel yöntemden elde edilen sonuç olarak karşımıza çıkmaktadır. Yerel arama problemin orjinal datasının komşuluklarında rassal olarak elde edilmiş problem örneklemeleri oluşturmada kullanılmaktadır. Daha sonra oluşan her bir problem, problemin özel sezgisel yöntemi ile çözülmektedir. Çözümün kalitesi orjinal verileri kullanarak değerlendirilir ve m çözümler arasından en iyi çözüm nihai çözüm olarak rapor edilir. Meta-sezgiselin problem uzayındaki komşuluk boyutu, r_i ve m için dağılımı iki temel parametre olarak karşımıza çıkmaktadır. Bu kısımda r_i problemin orjinal data yapısını tahrip etmeyecek şekilde olmalı ve aynı zamanda çeşitli sonuçlara etki edecek kadar anlamlı olmalıdır. m 'i arttırdıkça çözümün kalitesinin artacağını ve buna bağlı olarak çözüm süresinin uzayacağı beklenen bir durum olarak karşımıza çıkmaktadır. Buradaki önemli nokta, rassal pertübasyon yeterince büyük olmaz ise, m değerini arttırmanın çözümü iyileştirmeye etkisinin olmayabileceğidir. Wu ve arkadaşları yaptıkları çalışmada sezgiseli iş çizelgeleme problemine uygulamışlardır. Wu ve arkadaşlarının bu

önermesinden sonra sezgisel farklı problem alanlarında da kullanılmaya başlanmıştır. Renaud, Boctor, ve Laporte[21] bu sezgisel yöntemi gezgin satıcı problemi için uygulamışlardır. Bu çalışmalarında 7 adet pertürbasyon sezgiseli tanımlanmış ve karşılaştırılmıştır. 7 farklı pertürbasyon sezgiselinde adımlar ve parametreler değiştirilmiştir. Codenotti ve arkadaşları[22] pertürbasyon sezgiselini geniş kapsamlı öklid gezgin satıcı örneklemelerine ekleyerek yerel optimum noktalardan kaçmayı hedeflemişlerdir. Jeet ve Kutanoğlu[23] data karıştırma sezgiselini langranj gevşetmesi ile birlikte genelleştirilmiş atama problemi için uygulamışlardır. Burada pertürbasyon sezgiseli kaliteli üst sınırlar elde etmek için kullanılmıştır. Albercht, Panton, Lee [24] bu yaklaşımı parça bakım kesintileri altında uzun mesafeli tek parça ağlar için tren tarifelerinin oluşturulmasında kullanmıştır. Bu çalışmada data karıştırma sezgiselini hızlı alternatif tarifeler ve aksamalardan sonra revize edilmiş tarifeler oluşturulması için kullanmışlardır.

4. PROBLEMİN MATEMATİKSEL MODELİ

Uçak tamir bakım servisleri için önerdiğimiz üretim çizelgeleme modeli aşağıda verilmiştir;

- Amaç Fonksiyonu

$$\text{Min } \sum_{j=1}^J \sum_{i \in N(j)} \sum_{t=1}^{d_{ji}} w_{ji} (d_{ji} - t) * X_{jit}$$

- Kısıtlar

$$\sum_{t=1}^{d_{ji}} X_{jit} = 1 \quad \forall (j, i): j = 1, \dots, J; i \in N(j) \quad (1)$$

$$B_{jt} = B_{j,t-1} + \sum_{i \in N(j)} X_{jit} - Y_{jt} \quad \forall (j, t): j = 1, \dots, J, t = 1, \dots, T \quad (2)$$

$$H_{jt} = H_{j,t-1} - \sum_{i \in N(j)} X_{jit} + Y_{j,t-p_j} \quad \forall (j, t): t = p_j, \dots, T, j = 1, \dots, J \quad (3)$$

$$H_{jt} = H_{j,t-1} - \sum_{i \in N(j)} X_{jit} \quad \forall (j, t): t = 1, \dots, (p_j - 1), j = 1, \dots, J \quad (4)$$

$$\sum_{j=1}^J \sum_{l=t}^{\min(t+p_j-1, T)} Y_{jl} \leq K \quad \forall t: t = 1, \dots, T \quad (5)$$

$$\sum_{i \in N(j)} X_{jit} \leq H_{j,t-1} + Y_{j,t-p_j} \quad \forall (j, t): t = p_j, \dots, T, j = 1, \dots, J \quad (6)$$

$$\sum_{i \in N(j)} X_{jit} \leq H_{j,t-1} \quad \forall (j, t): t = 1, \dots, (p_j - 1), j = 1, \dots, J \quad (7)$$

$$H_{j,0} = s_j \quad \forall j: j = 1, \dots, J \quad (8)$$

$$B_{j,0} = 0 \quad \forall j: j = 1, \dots, J \quad (9)$$

$$X_{jit} \in \{0,1\} \quad \forall (j, i, t): j = 1, \dots, J, i \in N(j), t = 1, \dots, T \quad (10)$$

$$Y_{j,t} \geq 0 \text{ ve tamsayı} \quad \forall (j, t): j = 1, \dots, J, t = 1, \dots, T \quad (11)$$

Modelin amaç fonksiyonu, müşterilerin hangi tarihlerde çağrılacağına karar vererek, tüm müşteriler üzerinden toplam ağırlıklandırılmış erken çağırmanın minimize edilmesidir. Kısıt (1) her tip envanterin, her talebinin, o talep için daha önceden belirlenmiş olan son değişim gününe kadar karşılanmasını ve değişimini sağlamaktadır. Kısıt (2), periyot t de bakım bekleyen j tipi envanter sayısını belirlemek için kullanılmaktadır. Burada bir önceki periyotta bakım bekleyen envanter sayısına, içinde bulunulan periyotta gelen talep eklenir ve bu miktardan bakıma başlayan envanter sayısı çıkartılır. Bu işlem her tip için her dönemde uygulanmaktadır. Kısıt (3) ve (4) hazır stok dengesi kısıtlarını temsil etmektedir. Kısıt (4) j tipinden t anındaki hazır stok miktarını belirlemek için kullanılmaktadır. Burada bir önceki periyotta j tipi için eldeki hazır stoktan, o periyotta j tipinden takas yapılması halinde envanterden düşülür. Bu kısıttaki önemli nokta j tipi envanter için bakım onarım süresinin, içinde bulunulan periyottan küçük olmasından dolayı bakım hattından bakımı bitip çıkan ve hazır hale gelen herhangi bir j tipi envanter olmamasıdır. Kısıt (3) ise (4) teki gibi kullanıma hazır envanter dengesini, bulunulan periyot t 'nin bakım süresinden fazla olması durumu için yansıtır. Kısıt (3)'te içinde bulunulan periyot düşünüldüğünde, daha önceden bakıma giren j tipli bir ürünün bakım hattından çıkabilecektir ve hazır stoğa bakımdan çıkmış j ürünün girmesi sağlanır.

Kısıt (5) bakıma giren ürünlerin toplamının paralel bakım hatlarının sayısından küçük olması gerektiği kısıtını yansıtır. Bir j tip envanterden t periyodunda Y_{jt} adet tamir bakımı başlamış ise, Y_{jt} adet üretim hattı, t periyodundan $t + p_j - 1$ periyoduna kadar bu envanterlerin tamir bakımıyla ilgili olarak meşgul olacaktır. Burada $\min(t + p_j - 1, T)$ kullanılarak bakım süresinin bitişinin planlanan periyotun dışına çıkması engellenmektedir. Herhangi bir t periyodunda meşgul hatların toplamını bulmak için, her bir tip için meşgul hatlar tüm tipler bazında toplanır.

Kısıt (6) j tipinden t periyodunda gelen taleplerin sayısının eldeki hazır envanterden fazla olmamasını sağlamaktadır. Önemli nokta içinde bulunulan periyotun bakım süresinden büyük olmasından dolayı, bakımdan çıkan envanter olabileceğidir. Kısıt (7) bir önceki kısıtla aynı amacı taşımaktadır, tek fark içinde bulunulan periyot j tipi için bakım süresinden küçük olduğundan, bakımdan çıkacak envanter olmayacağından hazır stoğa eklenmeyecektir. Kısıt (8) ve (9) sırasıyla hazır envanter

ve tamir bakıma girecek envanter için planlama dönemi başlangıç değerlerini belirlerler.

Bu model başlangıç devinimli stok miktarlarını bir parametre olarak kabul etmektedir. Klasik üretim planlama modellerinde kullanılan üretim ve stok maliyetleri en azlanmasından çok farklı olarak, modelde toplam ağırlıklandırılmış erken çağırma en azlanmaya çalışılmaktadır. Genel olarak tabidir ki yüksek başlangıç stoklarının veya paralel üretim hattı kapasitesinin artması, toplam erken çağırmayı azaltacaktır. Öte yandan tamir bakım sürelerinin azalması da toplam erken takasları azaltacaktır. Bir sonraki kısımda bu parametrelerin etkileri ve ilişkileri sayısal örnek çözümleri üzerinden irdelenmiştir.

5. PROBLEMİN ÖN SAYISAL ANALİZİ

Ön sayısal analiz çalışmasında üç tip devinimli envanter içeren problemler kullanılmıştır. Bu problemler tamir bakım süreleri, planlama ufku, talepler arası süre itibariyle Amerika'daki bir BO şirketinin gerçek datasından esinlenilerek üretilmiştir. Otuz adet rassal problem farklı parametre kümelerinde çözdürülmüştür. Planlama periyodu olarak gün alınmıştır ve her problem 1100 günlük (yaklaşık üç yıl) planlama ufku içermektedir. Her bir tip devinimli envanter takas talep sayısı 1100 gün boyunca yaklaşık 50 dir ve toplamda dolayısıyla yaklaşık 150 takas talebi söz konusudur. 30 problem sadece içerdikleri talep serisi açısından farklıdır ve bu problemler üç tip envanter için olan takas taleplerinin (son) tarihlerinin, 1100 günlük planlama ufkunda rassal olarak 1 ile 1100 arası tek düze dağılıma uygun olarak dağıtılmasıyla oluşturulmuştur. Talep ağırlık değerleri her talep için aynı alınmıştır.

Üretim hattı sayılarındaki değişimin etkisinin görülebilmesi amacı ile üretim hattı sayıları olarak 3, 4, ve 5 kullanılmıştır. Tamir bakım süresi olarak 3 tip envanter için, 3 farklı küme belirlenmiştir; (35,25,30), (30,20,15), (25,15,10). Parantez içindeki değerler sırasıyla tip 1, 2, ve 3 envanter için, gün cinsinden tamir bakım süreleridir. Başlangıç stok seviyelerindeki değişikliklerin problemi nasıl etkileyeceğini görme amacı ile de eldeki başlangıç hazır stok miktarları 3 tip envanter için, 3 farklı şekilde belirlenmiştir. Bu değerler tip 1, 2, ve 3 envanter için olmak üzere (3,3,3), (4,4,4), ve (5,5,5) olarak kullanılmıştır. Yukarıdaki 3 faktörün 3 seviyesi göz önüne alındığında oluşan yirmi yedi farklı parametre senaryosu modele uygulanmış ve her senaryoda aynı 30 rassal problem çözülmüştür. Aşağıda, tüm senaryolar ve elde edilen sonuçlar hem çizelge 5.1 de verilmiş ve hem de karşılaştırma için 5.1, 5.2. ve 5.3. de grafiksel olarak gösterilmiştir. Çizelgede raporlanan iki sonuç her senaryoda 30 problem çözümünden elde edilen ortalama amaç değeri ve amaç değerlerinin standart sapmasının ortalamalarına oranından elde edilen belirlilik katsayılarıdır.

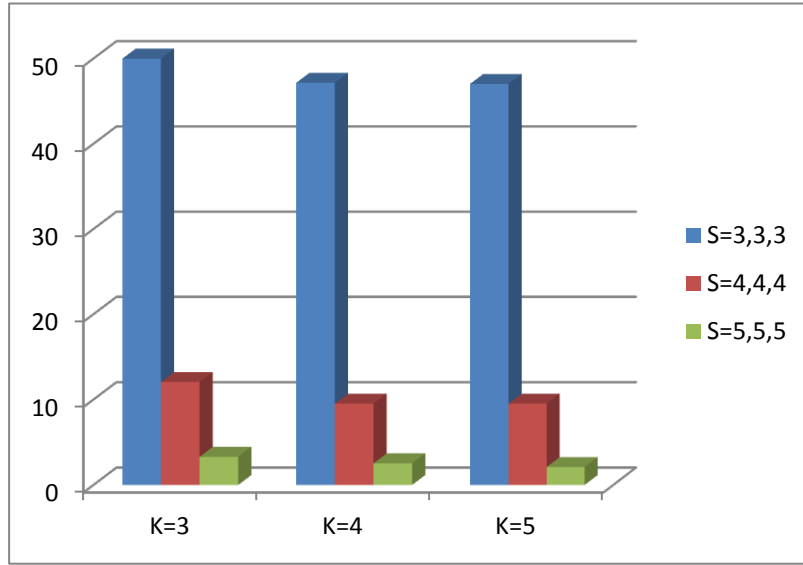
Çizelge 5.1. Koşturulan senaryolar ve sonuçlar

Tamir bakım süresi (gün)	Başlangıç stoğu	Üretim hattı sayısı	Senaryo	Amaç değeri ortalaması (gün)	Amaç değeri belirlilik katsayısı
35,25,30	3,3,3	3	1	Olursuz	Olursuz
		4	2	307,9	0,759
		5	3	183,4	0,485
	4,4,4	3	4	Olursuz	Olursuz
		4	5	61,93	1,343
		5	6	37,63	1,184
	5,5,5	3	7	Olursuz	Olursuz
		4	8	9,86	2,997
		5	9	6,56	3,48
30,20,15	3,3,3	3	10	350,46	0,848
		4	11	94,06	0,683
		5	12	93,9	0,661
	4,4,4	3	13	85,13	1,448
		4	14	20,93	1,548
		5	15	19,93	1,602
	5,5,5	3	16	17,56	2,703
		4	17	4,23	4,991
		5	18	3,46	4,885
25,15,10	3,3,3	3	19	55,03	0,878
		4	20	47,06	0,928
		5	21	47,03	0,940

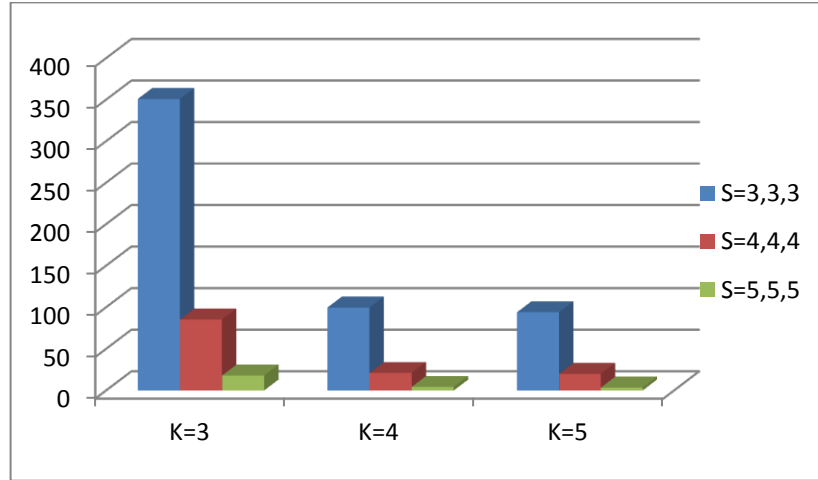
	4,4,4	3	22	12,06	2,489
		4	23	9,53	2,310
		5	24	9,5	2,310
	5,5,5	3	25	3,3	5,477
		4	26	2,53	5,472
		5	27	2,1	5,385

Yukarıdaki parametre aralığının dışındaki değerlerde problemlerin olursuz olma veya sıfır objektif değerine sahip olma durumlarıyla karşılaşılabilir. Örneğin üretim hattı sayısını 2'ye indirdiğimizde birçok problem olursuz olurken hat sayısını 6'ya getirdiğimizde çoğu senaryo ve problemin çözümünde objektif değeri sıfır olmaktadır. Dolayısıyla parametrelerin etkilerinin analizi açısından yukarıdaki parametre aralıkları uygun bulunmuştur.

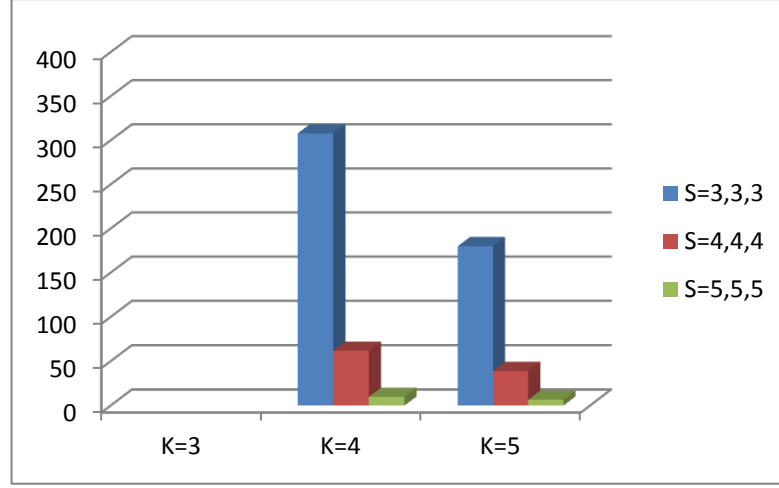
$P=35, 25, 20$ ve $K=3$ olarak belirlenen senaryo 1, 4 ve 7 deki problemlerde olurlu çözüm elde edilememiştir. Bunun nedeni düşük üretim hattı kapasitesi ve yüksek tamir bakım süresi içeren bu senaryolarda, belirlenen başlangıç stokları olurlu çözüm için yeterli gelmemiş olmasıdır. Bu tablosal değerler analiz için aşağıda grafikleştirilmiştir.



Şekil 5.1. P=25,15,10 olması durumunda optimal çözüm karşılaştırmaları



Şekil 5.2. P=30,20,15 olması durumunda optimal çözüm karşılaştırmaları



Şekil 5.3. P=35,25,20 olması durumunda optimal çözüm karşılaştırmaları

Şekil 5.1., 5.2., ve 5.3., deki grafiklere göre aşağıdaki değerlendirmeleri yapabiliriz; bir ek envanterin getirisi üstsel olarak azalmaktadır. Bunu tüm grafiklerde gözlemlemek mümkündür. Örneğin, şekil 5.2. ve K=3 senaryosunu göz önüne alalım. Bu senaryoda başlangıç stoklarını S=3 ten S=4 çıkarmanın getirisinin, S=4 ten S=5 e çıkartmanın getirisinden çok daha fazla olduğu görülebilir. Yani bir ek envanterin marjinal getirisi azalmıştır. Envanterdeki bu artışın getirisi orantısal olarak düşünülürse, getirinin üstel olarak azaldığını söyleyemeyiz. Örneğin yine şekil 5.2 ve K=3 senaryosunu göz önüne alalım. Bu senaryoda başlangıç stoklarını S=3 ten S=4'e çıkarmanın amaç değeri açısından yüzdesel getirisi $(100 \times (350,46 - 85,13) / 350,8) = \%75,7$ ve in S=4 ten S=5 e çıkartmanın getirisi $(100 \times (85,13 - 17,56) / 85,13) = \%79,0$ olmaktadır.

Ek bir envanterin net getirisinin düşük üretim hattı kapasitesinde daha fazla olduğu da grafiklere göre söylenebilir. Örneğin şekil 5.1'e bakıldığında, K değerleri arttıkça bir ek envanterin getirisinin azaldığı görülebilir. Başka bir gözlem ise yine bir ek envanterin amaç değerindeki getirisinin, yüksek tamir bakım sürelerinde daha fazla olduğudur. Bu etkiyi örneğin K=4 için amaç değerlerinin envantere göre değişimine, şekil 5.1., 5.2., ve 5.3. da baktığımızda görebiliriz. Şekil 5.3.'deki K=4 için ek bir envanterin net getirisi, şekil 5.1'dekinden çok daha fazladır. Bu gözlemden çıkarılacak sonuç başlangıç stoklarının artırmadan önce yöneticilerin dikkatli düşünmeleri gerektiğidir. Eğer tamir bakım süreleri az ya da mevcut üretim hattı

sayısı yeterince fazla ise, bu artışların toplam erken gelişleri azaltma anlamında hiçbir getirisi olmayabilir, ya da çok az getirisi olabilir.

Bu konuyla ilgili özel bir kısıt, üretim hattı sayısının toplam başlangıç stoğundan fazla olmasının hiçbir anlamı olamayacağıdır zira her üretim hattı aynı anda ancak bir envanterin tamir bakımıyla meşgul olabileceğinden, envanter sayısından fazla olan üretim hatları hep atıl kalacaktır.

Yukarıda bir ek envanterin getirisiyle ilgili gözlemlerin benzerleri bir ek üretim hattının getirisi içinde söylenebilir. Özet olarak bir ek üretim hattının üstel olarak azalan bir getirisi vardır ve bu getiri yüksek başlangıç stoklarında ve düşük tamir bakım sürelerinde daha az olmaktadır.

Önemli bir konu herhangi bir problem için bir ek envanter mi yoksa bir ek üretim kapasitesinin mi amaç değerini daha iyileştireceği sorusudur. Sonuçların geneline bakıldığında ek envanterin getirisinin ek proses kapasitesi getirisinden daha fazla olduğu söyleyebiliriz. Örneğin senaryo 2 ile senaryo 3 (ek envanter) ve senaryo 2 ile senaryo 5 (ek proses kapasitesi) karşılaştırmalarına baktığımızda ek envanterin getirisinin, ek proses kapasitesi getirisinden çok daha fazla olduğu görülebilir. Ek envanterin getirisi tüm tamir bakım sürelerinde görülürken, ek proses kapasitesinin getirisi tamir süresi yeterince düşükse sıfırlanmaktadır. Örneğin senaryo 20 ve 21'i karşılaştırdığımızda veya senaryo 23 ve 24'ü karşılaştırdığımızda bunu görebilmekteyiz. Buradan çıkarttığımız genel sonuç, özellikle bir birim ek envantere yapılacak yatırım ek proses kapasitesi için gereken yatırımdan az ise, ek envanter kapasitesi oluşturmanın, ek proses kapasitesi oluşturmaya nazaran daha mantıklı bir strateji olacaktır.

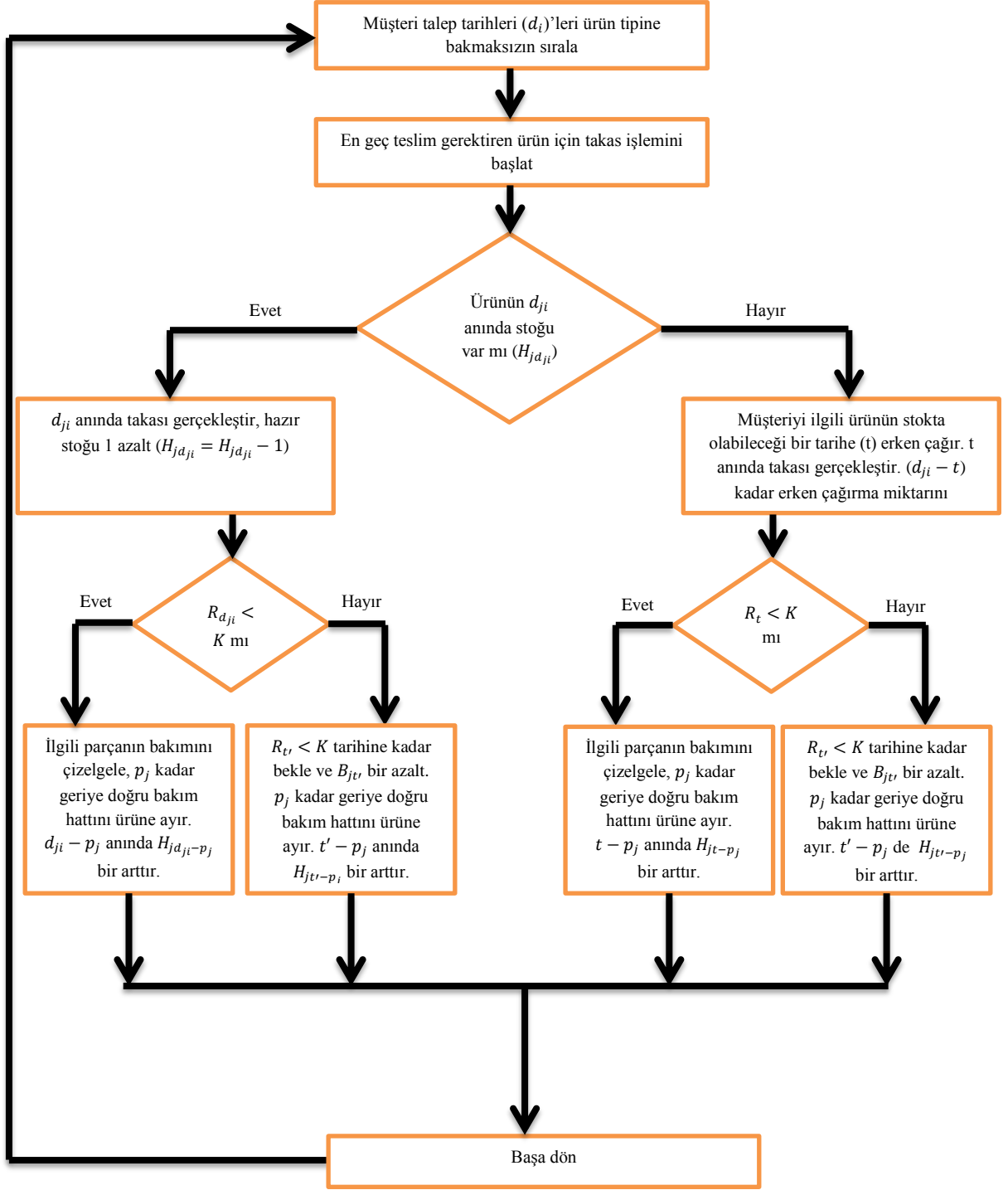
6. ÇÖZÜM YAKLAŞIMLARI

Problem için temel yaklaşım olarak “Geriye Doğru Çizelgeleme” sezgiseli belirlenmiştir. Bu sezgisel yöntemde esas olan teslim edilmesi gereken en son ürünü ilk çizelgeleyip, çizelgeyi sondan başa doğru talep tarihlerine göre büyükten küçüğe olacak bir sıralamayla oluşturulması üstüne kurulmuştur.

Bu temel sezgiselin performansı yeterli olmadığından “data karıştırma” meta-sezgisel yöntemi kullanılmıştır. Buradaki temel fikir, problemin özel sezgisel yönteminin sağladığı avantaj ile yerel arama stratejilerinin verdiği avantajı bir arada kullanmaktır. Data karıştırma sezgiseli, temel algoritmayı ve yerel aramanın avantajlarını kullanarak, problemin komşuluklarında alternatif çözümler elde etmektedir. Bu yöntem, orijinal problem datalarından, rassal olarak yeni problem örneklemi elde edildikten sonra uygulanır. Bu kısımda problemin orijinal datalarına $\pm\Delta$ eklenir ve sıralama yaptırılır. Elde edilen sıralama orijinal problemin veri kümesi kullanılarak “Geriye Doğru Çizelgeleme” sezgiseli uygulanır.

Geliştirme işleminden sonra performansı arttırmak için talepler arasından erken çağırılması gereken en kötü üç talep, sırasıyla beş kez sıralamada bir üste kaydırılarak yeni sıralamalara göre çözüm bulunur. Böylece daha iyi sonuçlar elde edilmeye çalışılmıştır.

6.1. Geriye Doğru Çizelgeleme Sezgiseli



Şekil 6.1. Algoritma akış şeması

Daha önce bahsettiğimiz varsayımlar ve senaryolar önerilen çözüm yaklaşımları için de geçerlilik göstermektedir. Amaç yine erken müşteri çağrılmasını minimize ederek ilgili ürünleri mümkün olduğunca zamanında müşteriye teslim etmektir. Önerilen çözüm yönteminin temel mantığı sondan başa doğru takasların ve tamir bakımın çizelgelenmesidir. Önerilen çözüm yöntemi, rassal olarak belirlenmiş olan talep tarihlerini büyükten küçüğe sıralamaktadır. Bu kısımda ürün tipleri göz ardı edilir. Teslim tarihi en geç olan ürün için stoktan malzeme vererek ve müşteriden bakıma ihtiyaç duyan parçası alınarak algoritma işleme başlar. Sondan başa doğru işlem yapar. Her tip ürün için başlangıç stokunun olduğu ve paralel bakım hatlarının müsait olduğu düşünüldüğünde; önerilen çözüm yöntemi, aldığı bakıma ihtiyacı olan ürün için bakım-onarım işlemini başlatır. Burada bakım için geçen süre önceki bölümde olduğu gibi her tip ürün için farklı bir sabit değer olarak belirlenmiştir.

Müşteri takası gerçekleştirildikten sonra algoritma, teslim tarihi en geç olan sonraki ürün ile ilgilenmeye başlamaktadır. Stoku kontrol ederek işlemler devam etmektedir. Müşteri teslim tarihinde ilgili ürün ile ilgili stok yok ise müşteriyi ilgili ürünün ulaşılabilir olduğu bir tarihe erken çağırarak iş planlaması yapılır. Bu döngü planlanan çalışma ufku boyunca devam etmektedir.

Performans Analizi

Bu kısımda bölüm beşte kullanılan datalar algoritma için de esas alınmıştır. Uygulanan senaryolarda yine bölüm beşte kullanılmış olan senaryolar ile aynıdır. Buradaki temel amaç algoritma performansını göstermektir. Grafiklerde farklı problem tiplerinde ve parametrelerde optimalden ortalama sapmalar gösterilmiştir.

Çizelge 6.1. Geriye doğru çizelgeleme yaklaşımının sonuçlarının problem tipine göre yüzde sapması

P=35,25,20	S	K=3	K=4	K=5
	3,3,3	olursuz	49,4%	8,3%
	4,4,4	olursuz	71,0%	13,6%
	5,5,5	olursuz	159,8%	1,5%

P=30,20,15	S	K=3	K=4	K=5
	3,3,3	79,8%	29,9%	2,6%
	4,4,4	164,1%	11,3%	0,0%
	5,5,5	245,0%	6,5%	0,0%

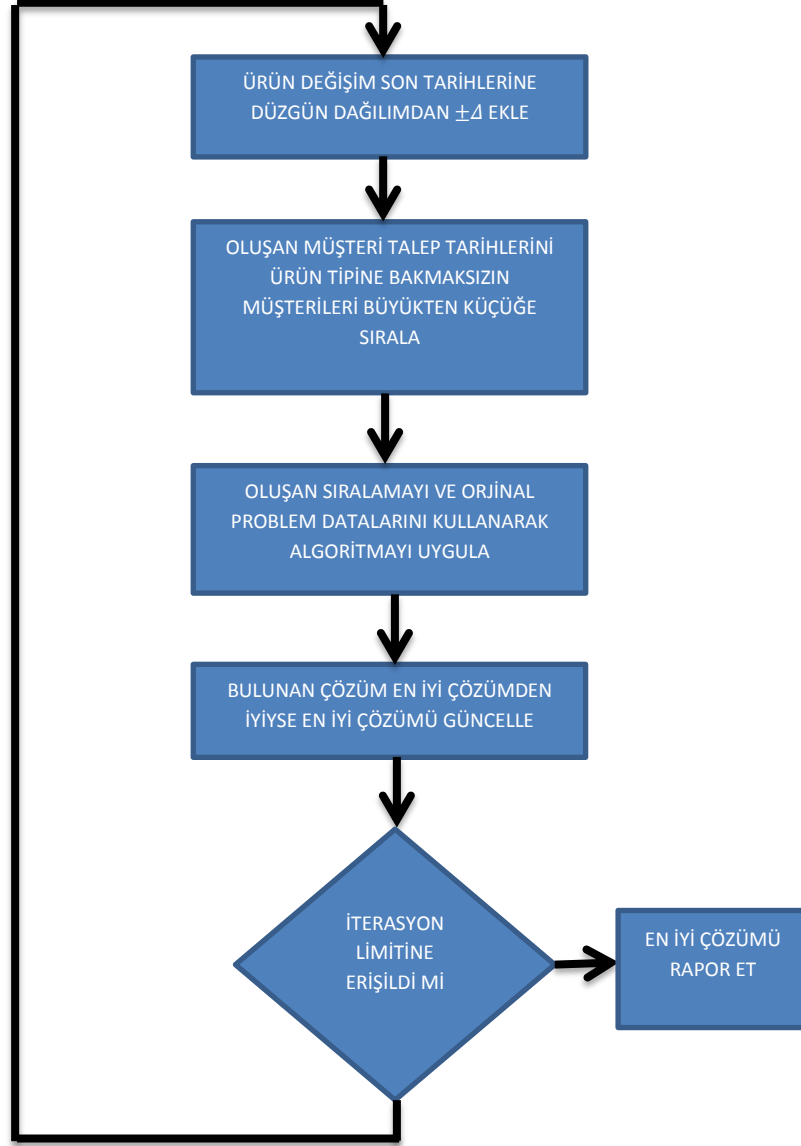
P=25,15,10	S	K=3	K=4	K=5
	3,3,3	28,1%	4,5%	1,5%
	4,4,4	13,3%	0,0%	0,0%
	5,5,5	0,0%	0,0%	0,0%

Algoritma performansı, optimal sonuçlar ile kıyaslandığında özellikle bakım hattı sayısının ve eldeki stok miktarının düşük olduğu durumlarda iyi gözükmemektedir. Bu sebeple bu algoritmayı temel alan aşağıdaki iki meta sezgisel yaklaşım uygulanmıştır.

7. DATA KARIŐTIRMA META SEZGİSELİ

7.1. Sezgisel Akıő Őeması ve Açıklama

Data kariőtirma meta sezgiselinde, ilk önce problemin datalarına $\pm\Delta$ büyüklüğünde bir deęer eklenir. $\pm\Delta$ müşteri taleplerinin ürün deęiőimi son tarihlerine eklenmektedir. Burada yeni bir data seti elde edilmektedir. Bu data setini büyükten küçüęe sıralayarak yeni bir müşteri sıralaması elde edilir. Ancak bu sıralama orijinal problem dataları kullanarak algoritma çalıştırılıp çizelge elde edilir ve sonuç kaydedilir. Bu işlemler 10000 kere tekrar ettirilerek en iyi sonuç kayıt edilir. Algoritmanın adımları aőađıda akıő diyagramı olarak gösterilmiőtir.



Şekil 7.1. Data karıştırma sezgiseli akış şeması

Performans Analizi

Sezgiselin çalışma performansını daha iyi anlayabilmek için senaryolar ve datalar bölüm beşte kullanılan datalar ile aynıdır. Bu kısımda Δ olarak sırasıyla 5 ve 10 değerleri kullanılmıştır. Bu değerlerin belirlenmesinde yapılan ön denemelerden yararlanılmış olup periyot içindeki sıralamanın en verimli olduğu bu iki değer üzerinden çalışmalar yürütülmüştür. Çizelge 7.1.'de sezgisel algoritma sonuçlarının,

$\pm\Delta$ 'nın sırasıyla "5 ve 10" olduğu durumlarda optimalden yüzde olarak sapma oranları parantez içindeki iki değer olarak verilmiştir.

Çizelge 7.1. Data karıştırma sezgiseli sonrası optimal sonuçlardan yüzde sapmalar

P=35,25,20	S	K=3	K=4	K=5
	3,3,3	olursuz	%(13,45;12,91)	%(4,59;3,56)
	4,4,4	olursuz	%(10,57;10,40)	%(7,86;5,31)
	5,5,5	olursuz	%(35,81;11,48)	%(0;0)

P=30,20,15	S	K=3	K=4	K=5
	3,3,3	%(20,55;35,29)	%(13,58;12,26)	%(2,20;1,77)
	4,4,4	%(47,27;46,41)	%(4,14;3,98)	%(0;0)
	5,5,5	%(19,82;20,21)	%(0;0)	%(0;0)

P=25,15,10	S	K=3	K=4	K=5
	3,3,3	%(10,72;12;05)	%(4,05;3,34)	%(1,41;1,20)
	4,4,4	%(6,62;6,90)	%(0;0)	%(0;0)
	5,5,5	%(0;0)	%(0,0)	%(0;0)

Data karıştırma sezgiseli uyguladıktan sonra algoritmanın performansının beklenildiği üzere iyileştirildiği görülmektedir. Ancak yine kapasitenin ve eldeki başlangıç stokunun kısıtlı olduğu durumlarda, sezgiselin yüzde sapması artmaktadır. Bunun için sezgiselle iyileştirme adımı eklenmiştir.

7.2. İyileştirme Adımlı Veri karıştırma Meta Sezgiseli

Data karıştırma meta sezgiselini geliştirmek için uygulanan yöntem sonucunda bir çözüm elde edilir. Elde edilen bu çözümde erken çağırma değeri en yüksek üç talebe odaklanılmıştır. Bu üç talebin her biri sıralamada bir, iki, üç, dört ve beş kere yukarı kaydırılarak öne alınmaktadır. Oluşan $5 \times 3 = 15$ sıralama kombinasyonunun en iyi çözümü dikkate alarak algoritma devam koşturulur. Data karıştırma sezgiselinden gelen çözüm ve ek olarak iyileştirme aşamasındaki 15 sıralama kombinasyonundan elde edilen çözümlerden en iyisi son çözüm olarak seçilir.

Performans Analizi

Bu kısımda yine ana problemin dataları ve koşturulan senaryolar beşinci bölümde kullanılanlar ile aynıdır. Data karıştırma sezgiselinin iyileştirilmesi sırasında $\pm\Delta$ sırasıyla “5 ve 10” olarak belirlenmiştir. Bu bölümden de elde edilen sonuçlar doğrultusunda en iyi değerler aşağıdaki çizelgede verilmiştir.

Çizelge 7.2. İyileştirme adımlı data karıştırma sezgiseli sonrası optimalden yüzde sapmalar

P=35,25,20	S	K=3	K=4	K=5
	3,3,3	olursuz	(12,7;11,5)%	(3,7;3,4)%
	4,4,4	olursuz	(9,7;10,4)%	(7,7;5,3)%
	5,5,5	olursuz	(35,8;16,6)%	(0,0;0,0)%

P=30,20,15	S	K=3	K=4	K=5
	3,3,3	(20,6;33,6)%	(12,2;12,1)%	(1,8;1,7)%
	4,4,4	(47,2;46,4)%	(4,1;4,0)%	(0,0;0,0)%
	5,5,5	(9,8;4,5)%	(0,0;0,0)%	(0,0;0,0)%

P=25,15,10	S	K=3	K=4	K=5
	3,3,3	(10,7;12,1)%	(3,5;3,3)%	(1,0;1,0)%
	4,4,4	(6,6;6,9)%	(0,0;0,0)%	(0,0;0,0)%
	5,5,5	(0,0;0,0)%	(0,0;0,0)%	(0,0;0,0)%

7.3. Toplu sonuçlar ve karşılaştırma

Sonuçların daha iyi karşılaştırılabilmesi için oluşturulan 27 farklı senaryonun (her birinde 30 problem); önerilen çözüm yöntemi, data karıştırma ± 5 , data karıştırma ± 5 iyileştirilmesi, data karıştırma ± 10 ve data karıştırma ± 10 iyileştirmeleri için yüzde standart sapma, çözemediği problem sayısı ve bunlara bağlı her bir problem için elde edilmiş olan en iyi değerler aşağıdaki çizelgede gösterilmiştir.

Çizelgeden de anlaşılacağı gibi önerilen çözüm yönteminin performansı genel olarak kabul edilebilir düzeydedir. Data karıştırma sezgiselleri ± 5 ve ± 10 durumlarında yaklaşık %7-7,5 optimalden sapma verirken en iyi sonuçların ortalama sapması %5,6'dır. Önerilen çözüm yönteminde, optimal değerden yüzde sapmanın yüksek olduğu bazı problem tipleri mevcuttur. Bu problem tiplerinde, stok sayısının kısıtlı olması, bakım hattı sayısının kısıtlı olması veya bu iki durumun aynı anda gerçekleşmesi temel sebep olarak karşımıza çıkmaktadır. Bu problemlerde kısıtlamanın fazla olması nedeniyle yanlış bir atama amaç değerini önemli ölçüde kötüleştirilmektedir.

Optimal değerler elde edilirken her bir problem için yaklaşık 3-5 dakika zaman harcanmıştır. Çözemediği problemleri düşündüğümüzde ve sistem hafızasının yetersiz kalmasından dolayı iptal edilen problemler ele alındığında, optimal değerleri elde etmek için yaklaşık olarak 25 saat harcanmıştır. Sezgisel yöntemde ise bütün

problemlerin çözüm süresi yaklaşık 2 saat olarak karşımıza çıkmaktadır. Optimal değerler elde edilirken harcanan süreyi de ele aldığımızda, önerilen çözüm yönteminin performansının kabul edilebilir düzeyde olduğu yine aşağıda yer alan çizelgede görülmektedir.

Çizelge 7.3. Tablosallaştırılmış değerler

Problemler	Algoritma		Pertürbasyon 5		Pertürbasyon 5+iyileştirme		Pertürbasyon 10		Pertürbasyon 10+iyileştirme		En İyi Değer
	Yüzde Standart Sapma	Çözeme diği Problem Sayısı	Yüzde Standart Sapma	Çözeme diği Problem Sayısı	Yüzde Standart Sapma	Çözeme diği Problem Sayısı	Yüzde Standart Sapma	Çözeme diği Problem Sayısı	Yüzde Standart Sapma	Çözeme diği Problem Sayısı	
k=3; s=3,3,3; p=25,15, 10	28,1	0,0	12,4	0,0	10,7	0,0	12,1	0,0	12,1	0,0	10,6
k=3; s=3,3,3; p=30,20, 15	79,8	10,0	20,6	10,0	20,6	10,0	35,3	10,0	33,6	10,0	18,5
k=3; s=3,3,3; p=35,25, 20	x	x	x	x	x	x	x	x	x	x	x
k=3; s=4,4,4; p=25,15, 10	13,3	0,0	6,6	0,0	6,6	0,0	6,9	0,0	6,9	0,0	6,6
k=3; s=4,4,4; p=30,20, 15	164,1	4,0	47,2	4,0	47,2	4,0	46,4	4,0	46,4	4,0	40,7
k=3; s=4,4,4; p=35,25, 20	x	x	x	x	x	x	x	x	x	x	x
k=3; s=5,5,5; p=25,15, 10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=3; s=5,5,5; p=30,20, 15	245,0	2,0	12,8	2,0	9,8	2,0	15,1	2,0	4,5	2,0	1,7
k=3;	x	x	x	x	x	x	x	x	x	x	x

s=5,5,5; p=35,25, 20											
k=4; s=3,3,3; p=25,15, 10	4,5	0,0	4,1	0,0	3,5	0,0	3,3	0,0	3,3	0,0	3,3
k=4; s=3,3,3; p=30,20, 15	29,9	0,0	13,6	0,0	12,2	0,0	12,3	0,0	12,1	0,0	11,7
k=4; s=3,3,3; p=35,25, 20	49,4	3,0	13,5	3,0	12,7	3,0	12,9	3,0	11,5	3,0	9,8
k=4; s=4,4,4; p=25,15, 10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=4; s=4,4,4; p=30,20, 15	11,3	0,0	4,1	0,0	4,1	0,0	4,0	0,0	4,0	0,0	4,0
k=4; s=4,4,4; p=35,25, 20	71,0	1,0	10,6	1,0	9,7	1,0	10,4	1,0	10,4	1,0	6,4
k=4; s=5,5,5; p=25,15, 10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=4; s=5,5,5; p=30,20, 15	6,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=4; s=5,5,5; p=35,25, 20	159,8	0,0	35,8	0,0	35,8	0,0	11,5	0,0	16,6	0,0	10,5
k=5; s=3,3,3; p=25,15, 10	1,5	0,0	1,4	0,0	1,0	0,0	1,2	0,0	1,0	0,0	1,0
k=5; s=3,3,3; p=30,20, 15	2,6	0,0	2,2	0,0	1,8	0,0	1,8	0,0	1,7	0,0	1,5
k=5; s=3,3,3; p=35,25, 20	8,3	0,0	4,6	0,0	3,7	0,0	3,6	0,0	3,4	0,0	3,1

20											
k=5; s=4,4,4; p=25,15, 10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=5; s=4,4,4; p=30,20, 15	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=5; s=4,4,4; p=35,25, 20	13,6	0,0	7,7	0,0	7,7	0,0	5,3	0,0	5,3	0,0	5,3
k=5; s=5,5,5; p=25,15, 10	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=5; s=5,5,5; p=30,20, 15	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
k=5; s=5,5,5; p=35,25, 20	1,5	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Ortala ma Yüzde Sapma	37,1		8,2		7,8		7,6		7,2		5,6

8. SONUÇLAR

Bu çalışmada kısıtlı kapasite ve kısıtlı stok altında havayolları bakım onarım şirketlerinin envanter planlamasına yardımcı olmak amacıyla rotatable parçaların sistematik bir şekilde talep tarihlerine uygun teslimini esas alan bir problem incelenmiştir.

Havayolu şirketlerinin yakıt sonrası en büyük maliyeti envanter kontrolü olarak kaşımıza çıkmaktadır. Havayolu şirketleri parça kullanım sürelerini maksimize ederek bu maliyetleri azaltma çabasına girmişlerdir. Problem bu sebepten ötürü büyük önem arz etmektedir.

Yapılan çalışmada, literatürde ilk defa yer alacak olan rotatable parçanın envanter kontrolü için bir model geliştirilmiş ve bu modele bağlı tamamen rassal taleplerden oluşan numerik bir çalışma ortaya konmuştur.

Problemin daha hızlı çözülebilmesi için sezgisel bir yöntem geliştirilmiştir.

Önerilen çözüm yönteminin performansının yeterli bulunmaması sebebiyle data karıştırma sezgiseli ve bu sezgiselin daha da iyileştirilmesi için yapılan çalışmalar yukarıda bulunan bölümlerde belirtilmiştir.

Data karıştırma sezgiselinin iyileştirilmesi sonucunda ana probleme uygulanan çözüm yöntemi performansı en iyiye çekilmiştir.

Yapılan işlemler sonucunda oluşturulan yirmi yedi farklı senaryo için optimalden ortalama olarak %5,61'lik bir sapma mevcuttur.

Optimal sonuçların elde ediliş sürelerinin uzun olduğunu ve önerilen çözüm yöntemini kullanılarak elde edilen sonuçların optimal sonuçlara yakınlığı düşünüldüğünde; problemleri sezgisel olarak çözümenin alternatif ve etkili bir yol olduğu saptanmıştır.

Problemlerin optimal çözümünü elde etmek için çözücü olarak "CPLEX OPL" kullanılmıştır. Yirmi yedi farklı senaryonun otuz farklı problem tipine

uygulanmasında programın hafıza hatası vermesinden dolayı işlemlerin otuz problem tipi ikiye böldürülmüştür ve on beşerli şekilde bir arada çözülmüştür. OPL de kullandığımız bu yaklaşım programın elli dört kere çalıştırılmasına sebep olmuştur. Bütün problem tiplerinin farklı senaryolarda çözüm işlemi yaklaşık on üç saat sürmüştür. Önerilen çözüm yönteminde ise, program dili olarak “JAVA” kullanılmıştır. Program, dataların hafızaya alınması kısmında ve otuz farklı problem tipine yirmi yedi farklı senaryonun uygulanmasında herhangi bir hata vermemiştir. Başlama talimatı verildikten sonra yaklaşık iki saat içerisinde problemler için çözüm bulmuştur.

Algoritma, data karıştırma sezgiseli ve data karıştırma sezgiseline uygulanan iyileştirme işleminden elde edilen sonuçların karşılaştırılması yapılmıştır.

Bu çalışmanın devamı niteliğinde yapılabilecek çeşitli çalışmalar vardır. Bunlardan birisi işçi kapasite planlamasının dahil olduğu modeldir. İşçi kapasitesine bağlı olarak bakım onarım süresi değişebilir. Başka bir versiyonu ise başlangıç envanter seviyesinin bir karar değişkeni olarak ele alındığı ve toplam envanter yatırımı üzerine bir bütçe kısıtının olduğu problemdir. Olası diğer bir çalışma mevcut probleme gevşetme tabanlı çözüm yaklaşımı getirmektir. Örneğin kapasite kısıtı gevşetildiğinde problem ürün bazlı kapasite kısıtsız problemlere ayrışacaktır.

KAYNAKLAR

- [1] Guajardo, J. A., M. Cohen, S.-H. Kim, and S. Netessine. 2012, "Impact of performance-based contracting on product reliability: an empirical analysis", *Management Science*, vol: 58(2), pp. 961-979.
- [2] TT. 2008. Turkish Technic Yıllık Faaliyet Raporu.
- [3] THY Teknik 2014 Yıllık Rapor (<http://www.thytekNIK.com/faaliyet-raporlari.html>).
- [4] Spafford, C., Hoyland, T. and Lehman, R. 2012. BO Industry Landscape, Marsh&McLennan Companies, www.oliverwyman.com
- [5] Kashyap, A. 2012. Supply chain optimization within aviation MRO, *International Journal of Computer Applications in Engineering Sciences*, vol:2 pp.95-101.
- [6] Kashyap, A. 2012. Supply chain optimization within aviation MRO, *International Journal of Computer Applications in Engineering Sciences*, vol:2 pp.95-101.
- [7] AeroStrategy/IATA 2009 Inventory & Logistics Survey - https://www.iata.org/whatwedo/workgroups/Documents/mro-logistics-survey-report-v3_2010.02.22.pdf
- [8] Ayeni, P., T. Baines, H. Lightfoot, and P. Ball. 2011, State-of-the-art of lean in the aviation maintenance, repair and overhaul industry, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, pp. 2108-2123.
- [9] Dekker, R. 1996, Applications of maintenance optimization models: a review and analysis, *Reliability Engineering and System Safety*, vol: 51, pp. 229-240.
- [10] Dekker, R. And P. A. Scarf. 1998. "On the impact of optimization models in maintenance decision making: the state of the art", *Reliability Engineering and System Safety*, vol: 60, pp. 111-119.
- [11] MacDonnell, M. and B. Clegg. 2011., A new inventory model for aircraft spares, in *Service Parts management*, edited by N. Altay and L. A. Litteral, Springer-Verlag London.
- [12] Van Jaarsveld, W., T. Dollevoet, and R. Dekker. 2015. "Improving spare parts inventory control at a repair shop", *OMEGA*, doi:10.1016/j.omega.2015.05.002
- [13] Kilpi, J., J. Töyli, and A. Vepsäläinen. 2009., Cooperative strategies for the availability services of repairable aircraft components, *International Journal of Production Economics*, vol: 117, pp. 360-370.
- [14] MacDonnell, M. and B. Clegg. 2007., Designing a support system for aerospace maintenance supply chains, *Journal of Manufacturing Technology Management*, vol: 18, pp.139-151.
- [15] Rupp, B., D. Pauli, S. Feller and M. Skytta. 2000., Workshop scheduling in the MRO context, *International Conference on Numerical Analysis and Applied Mathematics*, pp. 2000-2003.
- [16] Romo, S. and M. Erkoç. 2013. Exchange inventory policies and capacity management for maintenance overhaul and repair, *Teknik Rapor*, University of Miami, Coral Gables, FL.

- [17] Joo, S. H. 2009., Scheduling preventive maintenance for modular designed components: a dynamic approach, *European Journal of Operational Research*, vol: 192, pp. 512-520.
- [18] Luh, P. B., D. Yu, S. Soorapanth, A. I. Khibnik and R. Rajamani. 2005., A lagrangian relaxation based approach to schedule asset overhaul and repair services, *IEEE Transactions on Automation Science and Engineering*, vol: 2(2), pp.145-167.
- [19] Arts, J. and S. D. Flapper 2015, Aggregate overhaul and supply chain planning for rotables, *Annals of Operations Research*, January 2015, Volume 224, Issue 1, pp 77-100.
- [20] Storer, H.R., Wu, D.S., Vaccari, R., New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling, *Management Science*, Vol. 38, No. 10 (Oct., 1992), pp. 1495-1509.
- [21] Renaud, J., Boctor, F.F., Laporte, G., Perturbation heuristics for the pickup and delivery traveling salesman problem, *Computers & Operations Research* 29 (2002) 1129-1141.,2002.
- [22] Codenotti, B., Manzini, G., Margara, L., Resta, G., Perturbation: An Efficient Technique for the Solution of Very Large Instances of Euclidean TSP, *INFORMS journal on Computing*, May 1996, Volume 8, Issue 2, pp. 125-133.
- [23] Kutanoğlu, E., Candas, F., Jeet, V., Solution Methodologies for Integrated Network Design and Inventory Management in Service Parts Logistics, *IEE Annual Conference*, 1-6, 2006 .
- [24] Albrecht, A.R., Panton, D.M., Lee, D.H., Rescheduling rail networks with maintenance disruptions using Problem Space Search, *Computer and Operation Resarch*, March 2013, Volume 40, Issue 3, pp 703-712

EKLER

EK A: Optimal değerler, Önerilen Sezgisel Yaklaşım, İyileştirilmiş Sezgisel Yaklaşım ve En İyi Değer Çizelgeleri

Çizelge A.1. Problem tiplerine göre karşılaştırmalı sonuçlar 1

probl em no	k=3; s=3,3,3 ; p=25,1 5,10	k=3; s=3,3,3; p=25,15,10(alg otirma)	k=3; s=3,3,3; p=25,15,10(pertü rbasyon 5)	k=3; s=3,3,3; p=25,15,10(pertü rbasyon 5+iyileştirme)	k=3; s=3,3,3; p=25,15,10(pertü rbasyon 10)	k=3; s=3,3,3; p=25,15,10(pertü rbasyon 10+iyileştirme)	en iyi değ er
1	55	55	55	55	55	55	55
2	49	49	51	49	49	49	49
3	50	71	57	56	56	56	56
4	144	237	195	195	195	195	195
5	17	43	34	28	30	30	28
6	24	24	24	24	24	24	24
7	69	78	71	71	78	78	71
8	2	6	2	2	3	3	2
9	29	59	29	29	29	29	29
10	13	26	26	24	24	24	24
11	44	44	47	44	44	44	44
12	74	78	75	75	75	75	75
13	37	37	37	37	37	37	37
14	117	156	148	145	144	144	144
15	32	55	34	34	38	38	34
16	39	60	43	43	43	43	43
17	38	87	41	41	47	47	41
18	31	31	31	31	31	31	31
19	30	30	30	30	30	30	30
20	38	38	38	38	38	38	38
21	39	101	68	68	72	72	68
22	70	70	71	70	70	70	70
23	42	42	42	42	42	42	42
24	19	19	19	19	19	19	19
25	49	60	56	56	56	56	56
26	43	43	43	43	43	43	43
27	79	79	79	79	79	79	79
28	249	290	266	263	263	263	263
29	19	26	22	21	21	21	21
30	110	121	122	116	115	115	115

Çizelge A.2. Problem tiplerine göre karşılaştırmalı sonuçlar 2

problem no	k=3; s=3,3,3; p=30,20,15	k=3; s=3,3,3; p=30,20,15(algotirma)	k=3; s=3,3,3; p=30,20,15(pertürbasyon 5)	k=3; s=3,3,3; p=30,20,15(pertürbasyon 5 +iyileştirme)	k=3; s=3,3,3; p=30,20,15(pertürbasyon 10)	k=3; s=3,3,3; p=30,20,15(pertürbasyon 10+iyileştirme)	en iyi değer
1							
2	398						
3							
4							
5	259	792	575	575	585	585	575
6	117	589	257	257	272	272	257
7	268	630	487	487	537	537	487
8	126	196	126	126	174	174	126
9	234	711	375	375	480	480	375
10	143	906	409	409	415	412	409
11	236	869	581	581	574	574	574
12	352	407	360	360			360
13	172	370	256	256	252	252	252
14	506	1332	830	830	878	878	830
15	455	603	460	460	461	461	460
16	284						
17	150	367	323	323	305	305	305
18	82	149	112	112	103	103	103
19	908						
20	643						
21	733						
22	475						
23	121	222	172	172	147	147	147
24	204	869	541	541	537	505	505
25	509	879	593	593	590	590	590
26	112	246	167	167			167
27	158	356	277	277	257	257	257
28	1396	1901	1400	1400	1839	1770	1400
29	71	212	149	149	129	129	129
30							

Çizelge A.3. Problem tiplerine göre karşılaştırmalı sonuçlar 3

proble m no	k=3; s=4,4,4; p=25,15,1 0	k=3; s=4,4,4; p=25,15,10(algotırma)	k=3; s=4,4,4; p=25,15,10(pe rtürbasyon 5)	k=3; s=4,4,4; p=25,15,10(pertür basyon 5+iyileştirme)	k=3; s=4,4,4; p=25,15,10(pert ürbasyon 10)	k=3; s=4,4,4; p=25,15,10(pertürb asyon 5+iyileştirme)	en iyi değer
1	12	12	12	12	12	12	12
2	3	3	3	3	3	3	3
3	9	15	9	9	9	9	9
4	29	66	51	51	52	52	51
5	1	1	1	1	1	1	1
6	3	3	3	3	3	3	3
7	12	12	12	12	12	12	12
8	0	1	0	0	0	0	0
9	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	8	8	8	8	8	8	8
13	1	1	1	1	1	1	1
14	17	19	19	19	19	19	19
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	3	4	3	3	3	3	3
18	3	3	3	3	3	3	3
19	3	3	3	3	3	3	3
20	4	4	4	4	4	4	4
21	0	0	0	0	0	0	0
22	8	8	8	8	8	8	8
23	16	16	16	16	16	16	16
24	7	7	7	7	7	7	7
25	5	5	5	5	5	5	5
26	4	4	4	4	4	4	4
27	19	19	19	19	19	19	19
28	165	165	165	165	165	165	165
29	0	0	0	0	0	0	0
30	30	30	30	30	30	30	30

Çizelge A.4. Problem tiplerine göre karşılaştırmalı sonuçlar 4

proble m no	k=3; s=4,4,4; p=30,20,15	k=3; s=4,4,4; p=30,20,15(a lgotirma)	k=3; s=4,4,4; p=30,20,15(pe rtürbasyon 5)	k=3; s=4,4,4; p=30,20,15(pertür basyon 5+iyileştirme)	k=3; s=4,4,4; p=30,20,15(pe rtürbasyon 10)	k=3; s=4,4,4; p=30,20,15(pertür basyon 10+iyileştirme)	en iyi deęer
1	352						
2	31	470	213	213	227	227	213
3	38	401	177	177	171	171	171
4	196						
5	30	197	106	106	86	86	86
6	11	55	18	18	13	13	13
7	51	306	139	139	124	124	124
8	0	28	0	0	6	6	0
9	7	203	92	92	91	91	91
10	2	55	25	25	15	15	15
11	0	234	92	92	87	87	87
12	24	51	41	41	38	38	29
13	9	88	29	29	28	28	28
14	126	509	221	221	272	272	221
15	119	134	131	131	131	131	131
16	74	211	153	153	153	153	153
17	9	179	27	27	31	31	27
18	13	14	13	13	13	13	13
19	238	470	303	303	309	309	303
20	121	433	300	300	287	287	287
21	237	577	302	302	290	290	290
22	84						
23	29	40	29	29	29	29	29
24	19	122	101	101	58	58	58
25	64	88	64	64	69	69	64
26	9	63	12	12	16	16	12
27	34	51	41	41	39	39	39
28	538	859	626	626	654	654	626
29	4	8	4	4	4	4	4
30							

Çizelge A.5. Problem tiplerine göre karşılaştırmalı sonuçlar 5

pr obl em no	k=3; s=5,5,5; p=25,15,1 0	k=3; s=5,5,5; p=25,15,10(a lgotirma)	k=3; s=5,5,5; p=25,15,10(p ertürbasyon 5)	k=3; s=5,5,5; p=25,15,10(pertürb asyon 5+iyileştirme)	k=3; s=5,5,5; p=25,15,10(pe rtürbasyon 10)	k=3; s=5,5,5; p=25,15,10(pertürba syon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	2	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	99	99	99	99	99	99	99
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0

Çizelge A.6. Problem tiplerine göre karşılaştırmalı sonuçlar 6

pr obl em no	k=3; s=555; p=30,20, 15	k=3; s=555; p=30,20,15 (algotırma)	k=3; s=5,5,5; p=30,20,15(pe rtürbasyon 5)	k=3; s=5,5,5; p=30,20,15(pertürba syon 5+iyileştirme)	k=3; s=5,5,5; p=30,20,15(pe rtürbasyon 10)	k=3; s=5,5,5; p=30,20,15(pertürbas yon 10+iyileştirme)	en iyi değer
1	62						
2	0	69	23	23	17	17	17
3	0	46	8	8	13	10	8
4	27	230	107	107	115	95	95
5	0	25	0	0	0	0	0
6	0	0	0	0	0	0	0
7	3	291	10	10	13	9	9
8	0	0	0	0	0	0	0
9	0	4	0	0	0	0	0
10	0	2	0	0	2	0	0
11	0	14	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	9	0	0	0	0	0
14	16	182	66	66	80	68	66
15	0	9	0	0	0	0	0
16	0	58	22	22	35	26	22
17	0	8	0	0	0	0	0
18	0	0	0	0	0	0	0
19	6	43	9	9	9	7	7
20	0	88	19	19	13	13	13
21	37	85	44	44	37	37	37
22	0	175	23	23	17	17	17
23	3	3	3	3	3	3	3
24	0	18	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	3	3	3	3	3	3	3
28	152	335	218	203	209	209	203
29	0	0	0	0	0	0	0
30	218						

Çizelge A.7. Problem tiplerine göre karşılaştırmalı sonuçlar 7

pr obl em no	k=4; s=3,3,3; p=25,15,10	k=4; s=3,3,3; p=25,15,10(algotirma)	k=4; s=3,3,3; p=25,15,10(pe rtürbasyon 5)	k=4; s=3,3,3; p=25,15,10(pertürb asyon 5+iyileştirme)	k=4; s=3,3,3; p=25,15,10(pe rtürbasyon 10)	k=4; s=3,3,3; p=25,15,10(pertürb asyon 10+iyileştirme)	en iyi değer
1	55	55	55	55	55	55	55
2	49	49	49	49	49	49	49
3	35	40	35	35	35	35	35
4	90	90	90	90	90	90	90
5	13	13	13	13	13	13	13
6	24	24	24	24	24	24	24
7	67	71	71	69	67	67	67
8	2	2	2	2	2	2	2
9	22	27	23	22	22	22	22
10	13	13	13	13	13	13	13
11	44	44	46	44	44	44	44
12	55	55	55	55	55	55	55
13	26	27	27	27	27	27	27
14	46	47	48	46	46	46	46
15	24	24	24	24	24	24	24
16	30	39	39	39	39	39	39
17	32	32	32	32	32	32	32
18	31	31	31	31	31	31	31
19	24	24	24	24	24	24	24
20	38	38	38	38	38	38	38
21	26	26	26	26	26	26	26
22	70	70	70	70	70	70	70
23	41	41	41	41	41	41	41
24	19	19	19	19	19	19	19
25	48	48	48	48	48	48	48
26	43	43	43	43	43	43	43
27	79	79	79	79	79	79	79
28	249	282	282	282	282	282	282
29	19	19	19	19	19	19	19
30	92	97	97	96	96	96	96

Çizelge A.8. Problem tiplerine göre karşılaştırmalı sonuçlar 8

problemler no	k=4; s=3,3,3; p=30,20,15	k=4; s=3,3,3; p=30,20,15 (algotırma)	k=4; s=3,3,3; p=30,20,15 (per türbasyon 5)	k=4; s=3,3,3; p=30,20,15 (per türbasyon 5+iyileştirme)	k=4; s=3,3,3; p=30,20,15 (per türbasyon 10)	k=4; s=3,3,3; p=30,20,15 (per türbasyon 10+iyileştirme)	en iyi değer
1	105	110	105	105	105	105	105
2	94	117	95	94	94	94	94
3	98	125	102	100	105	105	100
4	199	311	235	235	233	233	233
5	61	78	74	69	68	68	68
6	57	57	57	57	57	57	57
7	123	125	123	123	123	123	123
8	23	32	23	23	23	23	23
9	81	109	99	89	89	89	89
10	51	51	52	51	51	51	51
11	94	100	98	94	94	94	94
12	143	145	145	145	145	145	145
13	65	73	73	73	73	73	73
14	168	203	181	180	180	179	179
15	51	51	51	51	51	51	51
16	65	82	71	71	69	69	69
17	65	79	72	72	72	72	72
18	71	71	73	71	71	71	71
19	60	66	64	63	63	63	63
20	92	98	98	98	98	98	98
21	69	151	91	85	85	85	85
22	135	135	135	135	135	135	135
23	87	87	87	87	87	87	87
24	44	46	49	44	44	44	44
25	85	89	85	85	89	89	85
26	73	73	73	73	73	73	73
27	130	130	130	130	130	130	130
28	355	360	360	360	365	361	360
29	58	60	63	60	60	60	60
30	2	429	221	222	216	216	216

Çizelge A.9. Problem tiplerine göre karşılaştırmalı sonuçlar 9

proble m no	k=4; s=3,3,3; p=35,25,20	k=4; s=3,3,3; p=35,25,20(a lgotirma)	k=4; s=3,3,3; p=35,25,20(p ertürbasyon 5)	k=4; s=3,3,3; p=35,25,20(pertürb asyon 5+iyileştirme)	k=4; s=3,3,3; p=35,25,20(pertürbasyon 10)	k=4; s=3,3,3; p=35,25,20(pertürb asyon 10+iyileştirme)	en iyi değer
1	346	703	500	500	494	494	494
2	256	640	392	392	339	339	339
3	323	705	501	500	505	505	500
4	490						
5	265	432	352	352	390	367	352
6	104	117	117	110	110	110	110
7	247	505	319	319	323	295	295
8	138	167	150	150	154	154	150
9	284	503	333	333	340	340	333
10	151	406	313	295	315	315	295
11	260	522	382	382	375	375	375
12	328	340	328	328	328	328	328
13	165	232	226	226	208	208	208
14	351	732	471	471	522	490	471
15	348	434	348	348	398	398	348
16	250						
17	133	282	212	212	190	190	190
18	140	145	140	140	140	140	140
19	385	658	467	467	479	479	467
20	246	526	394	394	374	374	374
21	448	812	674	674	633	633	633
22	257	787	498	498	505	505	498
23	173	213	194	194	194	194	194
24	237	451	319	290	284	284	284
25	247	385	281	281	265	265	265
26	129	150	146	146	134	134	134
27	216	259	251	251	244	244	244
28	943	1147	968	968	998	963	963
29	132	163	156	145	146	146	145
30	1245						

Çizelge A.10. Problem tiplerine göre karşılaştırmalı sonuçlar 10

problemler no	k=4; s=4,4,4; p=25,15,10	k=4; s=4,4,4; p=25,15,10(algoritma)	k=4; s=4,4,4; p=25,15,10(pertürbasyon 5)	k=4; s=4,4,4; p=25,15,10(pertürbasyon 5+iyileştirme)	k=4; s=4,4,4; p=25,15,10(pertürbasyon 10)	k=4; s=5,5,5; p=35,25,20(pertürbasyon 10+iyileştirme)	en iyi değer
1	12	12	12	12	12	12	12
2	3	3	3	3	3	3	3
3	9	9	9	9	9	9	9
4	15	15	15	15	15	15	15
5	1	1	1	1	1	1	1
6	3	3	3	3	3	3	3
7	12	12	12	12	12	12	12
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	8	8	8	8	8	8	8
13	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	3	3	3	3	3	3	3
18	3	3	3	3	3	3	3
19	3	3	3	3	3	3	3
20	4	4	4	4	4	4	4
21	0	0	0	0	0	0	0
22	8	8	8	8	8	8	8
23	16	16	16	16	16	16	16
24	7	7	7	7	7	7	7
25	5	5	5	5	5	5	5
26	4	4	4	4	4	4	4
27	19	19	19	19	19	19	19
28	120	120	120	120	120	120	120
29	0	0	0	0	0	0	0
30	30	30	30	30	30	30	30

Çizelge A.11. Problem tiplerine göre karşılaştırmalı sonuçlar 11

proble m no	k=4; s=4,4,4; p=30,20,15	k=4; s=4,4,4; p=30,20,15 (algotirma)	k=4; s=4,4,4; p=30,20,15 (pertürbasyon 5)	k=4; s=4,4,4; p=30,20,15 (pertürbasyon 5+iyileştirme)	k=4; s=4,4,4; p=30,20,15 (pertürbasyon 10)	k=4; s=4,4,4; p=30,20,15 (pertürbasyon 10+iyileştirme)	en iyi değer
1	22	22	22	22	22	22	22
2	16	16	16	16	16	16	16
3	14	24	14	14	14	14	14
4	55	71	71	71	71	71	71
5	6	6	6	6	6	6	6
6	11	11	11	11	11	11	11
7	31	31	31	31	31	31	31
8	0	0	0	0	0	0	0
9	1	9	1	1	1	1	1
10	2	2	2	2	2	2	2
11	0	0	0	0	0	0	0
12	18	18	18	18	18	18	18
13	9	11	11	11	11	11	11
14	30	31	30	30	30	30	30
15	5	5	5	5	5	5	5
16	0	0	0	0	0	0	0
17	8	8	8	8	8	8	8
18	13	13	13	13	13	13	13
19	10	10	10	10	10	10	10
20	14	14	14	14	14	14	14
21	0	0	0	0	0	0	0
22	37	37	37	37	37	37	37
23	29	29	29	29	29	29	29
24	12	12	12	12	12	12	12
25	10	10	10	10	10	10	10
26	9	9	9	9	9	9	9
27	34	34	34	34	34	34	34
28	173	173	173	173	173	173	173
29	4	4	4	4	4	4	4
30	55	89	63	63	62	62	62

Çizelge A.12. Problem tiplerine göre karşılaştırmalı sonuçlar 12

problemler no	k=4; s=4,4,4; p=35,25,20	k=4; s=4,4,4; p=35,25,20(a) (gotirma)	k=4; s=4,4,4; p=35,25,20(per) (türbasyon 5)	k=4; s=4,4,4; p=35,25,20(per) (türbasyon 5+iyileştirme)	k=4; s=4,4,4; p=35,25,20(per) (türbasyon 10)	k=4; s=4,4,4; p=35,25,20(per) (türbasyon 5+iyileştirme)	en iyi değer
1	38	132	87	87	81	81	81
2	40	128	58	58	49	49	49
3	43	130	72	72	81	81	72
4	170	301	235	235	227	227	227
5	30	94	50	45	46	46	45
6	24	24	24	24	24	24	24
7	61	256	73	73	87	87	73
8	6	18	9	9	9	9	9
9	27	136	47	47	56	56	47
10	12	63	14	12	12	12	12
11	18	71	37	37	33	33	33
12	50	50	50	50	50	50	50
13	25	49	32	32	33	33	32
14	107	183	138	136	137	137	136
15	59	71	59	59	63	63	59
16	31	114	60	60	84	84	60
17	14	18	16	16	16	16	16
18	23	23	23	23	23	23	23
19	28	83	63	63	62	62	62
20	29	107	67	67	56	56	56
21	56	190	139	139	148	148	139
22	77	144	99	99	86	86	86
23	53	53	53	53	53	53	53
24	32	34	35	32	32	32	32
25	28	49	38	38	32	32	32
26	14	14	14	14	14	14	14
27	62	63	62	62	62	62	62
28	289	462	320	316	315	315	315
29	12	12	12	12	12	12	12
30	400						

Çizelge A.13. Problem tiplerine göre karşılaştırmalı sonuçlar 13

proble m no	k=4; s=5,5,5; p=25,15,1 0	k=4; s=5,5,5; p=25,15,10(algotirma)	k=4; s=5,5,5; p=25,15,10(pe rtürbasyon 5)	k=4; s=5,5,5; p=25,15,10(pertürb asyon 5+iyileştirme)	k=4; s=5,5,5; p=25,15,10(pe rtürbasyon 10)	k=4; s=5,5,5; p=25,15,10(pertürb asyon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	76	76	76	76	76	76	76
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0

Çizelge A.14. Problem tiplerine göre karşılaştırmalı sonuçlar 14

proble m no	k=4; s=5,5,5; p=30,20,1 5	k=4; s=5,5,5; p=30,20,15 (algoritma)	k=4; s=5,5,5; p=30,20,15(p ertürbasyon 5)	k=4; s=5,5,5; p=30,20,15(pertür basyon 5+iyileştirme)	k=4; s=5,5,5; p=30,20,15(per türbasyon 10)	k=4; s=5,5,5; p=30,20,15(pertürba syon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	5	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	3	3	3	3	3	3	3
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	3	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	3	3	3	3	3	3	3
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	3	3	3	3	3	3	3
28	116	116	116	116	116	116	116
29	0	0	0	0	0	0	0
30	2	2	2	2	2	2	2

Çizelge A.15. Problem tiplerine göre karşılaştırmalı sonuçlar 15

proble m no	k=4; s=5,5,5; p=35,25,2 0	k=4; s=5,5,5; p=35,25,20 (algotirma)	k=4; s=5,5,5; p=35,25,20(pe rtürbasyon 5)	k=4; s=5,5,5; p=35,25,20(pertürb asyon 5+iyileştirme)	k=4; s=5,5,5; p=35,25,20(pert ürbasyon 10)	k=4; s=5,5,5; p=35,25,20(pertürb asyon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	30	0	0	1	0	0
4	33	157	47	47	47	47	47
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	12	12	12	12	12	12	12
8	0	0	0	0	0	0	0
9	0	8	0	0	0	0	0
10	0	0	0	0	0	0	0
11	3	3	3	3	3	3	3
12	0	0	0	0	0	0	0
13	3	3	3	3	3	3	3
14	14	39	39	39	21	19	19
15	0	0	0	0	0	0	0
16	0	11	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	1	0	0	0	0	0
20	4	4	4	4	4	4	4
21	0	29	0	0	0	0	0
22	5	8	5	5	5	5	5
23	8	8	8	8	8	8	8
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	10	10	10	10	10	10	10
28	156	156	156	156	156	156	156
29	0	0	0	0	0	0	0
30	48	290	115	115	60	60	60

Çizelge A.16. Problem tiplerine göre karşılaştırmalı sonuçlar 16

proble m no	k=5; s=3,3,3; p=25,15, 10	k=5; s=3,3,3; p=25,15,10(algotırma)	k=5; s=3,3,3; p=25,15,10(pe rtürbasyon 5)	k=5; s=3,3,3; p=25,15,10(pertürb asyon 5+iyileştirme)	k=5; s=3,3,3; p=25,15,10(pert ürbasyon10)	k=5, s=3,3,3 p=25,15,10(pertürb asyon 10+iyileştirme)	en iyi değer
1	55	55	55	55	55	55	55
2	49	49	49	49	49	49	49
3	35	35	35	35	35	35	35
4	90	90	90	90	90	90	90
5	13	13	13	13	13	13	13
6	24	24	24	24	24	24	24
7	67	71	71	71	71	71	71
8	2	2	2	2	2	2	2
9	22	22	22	22	22	22	22
10	13	13	13	13	13	13	13
11	44	44	45	44	44	44	44
12	53	53	53	53	53	53	53
13	24	24	24	24	24	24	24
14	46	51	51	48	51	48	48
15	24	24	24	24	24	24	24
16	39	39	39	39	39	39	39
17	32	32	32	32	32	32	32
18	31	31	31	31	31	31	31
19	24	24	24	24	24	24	24
20	38	38	38	38	38	38	38
21	26	26	26	26	26	26	26
22	70	70	70	70	70	70	70
23	41	41	41	41	41	41	41
24	19	19	19	19	19	19	19
25	48	48	48	48	48	48	48
26	43	43	43	43	43	43	43
27	79	79	80	79	79	79	79
28	249	261	257	257	257	257	257
29	19	19	19	19	19	19	19
30	92	92	93	92	92	92	92

Çizelge A.17. Problem tiplerine göre karşılaştırmalı sonuçlar 17

proble m no	k=5; s=3,3,3; p=30,20, 15	k=5; s=3,3,3; p=30,20,15(a lgotirma)	k=5; s=3,3,3; p=30,20,15(pe rtürbasyon 5)	k=5; s=3,3,3; p=30,20,15(pertürb asyon 5+iyileştirme)	k=5; s=3,3,3; p=30,20,15(pe rtürbasyon 10)	k=5; s=3,3,3; p=30,20,15(pertürb asyon 10+iyileştirme)	en iyi değer
1	105	105	105	105	105	105	105
2	94	94	94	94	94	94	94
3	66	71	66	66	66	66	66
4	186	191	191	191	191	191	191
5	51	51	51	51	51	51	51
6	57	57	57	57	57	57	57
7	122	138	136	134	136	134	134
8	23	23	23	23	23	23	23
9	68	78	69	69	68	68	68
10	51	51	51	51	51	51	51
11	94	94	98	98	98	98	94
12	119	124	121	121	120	120	120
13	58	63	63	63	63	63	63
14	147	157	157	157	157	157	157
15	49	49	49	49	49	49	49
16	65	65	65	65	65	65	65
17	65	65	65	65	65	65	65
18	71	71	71	71	71	71	71
19	60	60	60	60	60	60	60
20	92	98	98	98	98	98	98
21	51	51	51	51	51	51	51
22	134	134	135	134	134	134	134
23	87	87	87	87	87	87	87
24	44	44	44	44	44	44	44
25	79	79	80	79	79	79	79
26	73	73	73	73	73	73	73
27	128	128	129	128	128	128	128
28	351	360	356	356	353	353	353
29	58	58	61	58	58	58	58
30	169	171	173	171	172	171	171

Çizelge A.18. Problem tiplerine göre karşılaştırmalı sonuçlar 18

problemler no	k=5; s=3,3,3; p=35,25,20	k=5; s=3,3,3; p=35,25,20(algotirma)	k=5; s=3,3,3; p=35,25,20(pertürbasyon 5)	k=5; s=3,3,3; p=35,25,20(pertürbasyon 5+iyileştirme)	k=5; s=3,3,3; p=35,25,20(pertürbasyon 10)	k=5; s=3,3,3; p=35,25,20(pertürbasyon 10+iyileştirme)	en iyi değer
1	232	282	282	280	280	280	280
2	162	174	171	165	164	164	164
3	160	187	160	160	160	159	159
4	360	368	360	360	360	360	360
5	126	147	152	142	144	144	142
6	102	102	103	102	102	102	102
7	195	199	196	196	196	196	196
8	88	97	91	91	91	91	91
9	179	203	181	179	179	179	179
10	137	138	138	138	138	137	137
11	226	274	282	274	274	274	274
12	235	241	238	238	237	237	237
13	122	134	134	134	134	134	134
14	265	265	265	265	265	265	265
15	99	111	103	103	101	99	99
16	109	124	121	121	118	118	118
17	109	109	109	109	109	109	109
18	135	135	138	138	135	135	135
19	124	134	131	125	125	125	125
20	170	170	170	170	170	170	170
21	127	210	180	171	157	157	157
22	235	243	235	235	238	238	235
23	173	173	173	173	173	173	173
24	109	109	110	110	110	110	109
25	140	145	140	140	140	140	140
26	225	229	228	226	226	226	226
27	206	206	209	206	220	208	206
28	512	531	512	512	512	512	512
29	128	128	131	128	128	134	128
30	312	392	312	312	312	312	312

Çizelge A.19. Problem tiplerine göre karşılaştırmalı sonuçlar 19

pr obl em no	k=5; s=4,4,4; p=25,15,1 0	k=5; s=4,4,4; p=25,15,10(algotirma)	k=5; s=4,4,4; p=25,15,10(p ertürbasyon 5)	k=5; s=4,4,4; p=25,15,10(pertürb asyon 5+iyileştirme)	k=5; s=4,4,4; p=25,15,10(pert ürbasyon 10)	k=5; s=4,4,4; p=25,15,10(pertürb asyon 10+iyileştirme)	en iyi değer
1	12	12	12	12	12	12	12
2	3	3	3	3	3	3	3
3	9	9	9	9	9	9	9
4	15	15	15	15	15	15	15
5	1	1	1	1	1	1	1
6	3	3	3	3	3	3	3
7	12	12	12	12	12	12	12
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	8	8	8	8	8	8	8
13	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	3	3	3	3	3	3	3
18	3	3	3	3	3	3	3
19	3	3	3	3	3	3	3
20	4	4	4	4	4	4	4
21	0	0	0	0	0	0	0
22	8	8	8	8	8	8	8
23	16	16	16	16	16	16	16
24	7	7	7	7	7	7	7
25	5	5	5	5	5	5	5
26	4	4	4	4	4	4	4
27	19	19	19	19	19	19	19
28	120	120	120	120	120	120	120
29	0	0	0	0	0	0	0
30	30	30	30	30	30	30	30

Çizelge A.20. Problem tiplerine göre karşılaştırmalı sonuçlar 20

proble m no	k=5; s=4,4,4; p=30,20,15	k=5; s=4,4,4; p=30,20,15(algotirma)	k=5; s=4,4,4; p=30,20,15(per türbasyon 5)	k=5; s=4,4,4; p=30,20,15(pertürb asyon 5+iyileştirme)	k=5; s=4,4,4; p=30,20,15(pe rtürbasyon 10)	k=5; s=4,4,4; p=30,20,15(pertürb asyon 10+iyileştirme)	en iyi deęer
1	22	22	22	22	22	22	22
2	16	16	16	16	16	16	16
3	14	14	14	14	14	14	14
4	40	40	40	40	40	40	40
5	6	6	6	6	6	6	6
6	11	11	11	11	11	11	11
7	31	31	31	31	31	31	31
8	0	0	0	0	0	0	0
9	1	1	1	1	1	1	1
10	2	2	2	2	2	2	2
11	0	0	0	0	0	0	0
12	18	18	18	18	18	18	18
13	9	9	9	9	9	9	9
14	15	15	15	15	15	15	15
15	5	5	5	5	5	5	5
16	0	0	0	0	0	0	0
17	8	8	8	8	8	8	8
18	13	13	13	13	13	13	13
19	10	10	10	10	10	10	10
20	14	14	14	14	14	14	14
21	0	0	0	0	0	0	0
22	37	37	37	37	37	37	37
23	29	29	29	29	29	29	29
24	12	12	12	12	12	12	12
25	10	10	10	10	10	10	10
26	9	9	9	9	9	9	9
27	34	34	34	34	34	34	34
28	173	173	173	173	173	173	173
29	4	4	4	4	4	4	4
30	55	55	55	55	55	55	55

Çizelge A.21. Problem tiplerine göre karşılaştırmalı sonuçlar 21

proble m no	k=5; s=4,4,4; p=35,25, 20	k=5; s=4,4,4; p=35,25,20(algotırma)	k=5; s=4,4,4; p=35,25,20(per türbasyon 5)	k=5; s=4,4,4; p=35,25,20(pertürb asyon 5+iyileştirme)	k=5; s=4,4,4; p=35,25,20(pert ürbasyon 10)	k=5; s=4,4,4; p=35,25,20(pertürba syon 10+iyileştirme)	en iyi değer
1	38	38	38	38	38	38	38
2	40	40	40	40	40	40	40
3	21	35	23	23	23	23	23
4	98	115	115	115	115	115	115
5	11	16	11	11	11	11	11
6	24	24	24	24	24	24	24
7	61	61	61	61	61	61	61
8	6	6	6	6	6	6	6
9	18	25	18	18	18	18	18
10	12	12	12	12	12	12	12
11	18	18	18	18	18	18	18
12	5	41	37	37	37	37	37
13	25	25	25	25	25	25	25
14	56	61	58	58	58	58	58
15	10	10	10	10	10	10	10
16	5	5	5	5	5	5	5
17	13	13	13	13	13	13	13
18	23	23	23	23	23	23	23
19	20	20	20	20	20	20	20
20	29	29	29	29	29	29	29
21	9	9	9	9	9	9	9
22	77	77	77	77	77	77	77
23	53	53	53	53	53	53	53
24	17	17	17	17	17	17	17
25	21	21	21	21	21	21	21
26	14	14	14	14	14	14	14
27	62	62	62	62	62	62	62
28	234	295	265	265	235	235	235
29	12	12	12	12	12	12	12
30	97	106	107	107	103	103	103

Çizelge A.22. Problem tiplerine göre karşılaştırmalı sonuçlar 22

proble m no	k=5; s=5,5,5; p=25,15,1 0	k=5; s=5,5,5; p=25,15,10(algotirma)	k=5; s=5,5,5; p=25,15,10(pe rtürbasyon 5)	k=5; s=5,5,5; p=25,15,10(pertürb asyon 5+iyileştirme)	k=5; s=5,5,5; p=25,15,10(pe rtürbasyon 10)	k=5; s=5,5,5; p=25,15,10(pertürb asyon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	63	63	63	63	63	63	63
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0

Çizelge A.23. Problem tiplerine göre karşılaştırmalı sonuçlar 23

proble m no	k=5; s=5,5,5; p=30,20,15	k=5; s=5,5,5; p=30,20,15(algotirma)	k=5; s=5,5,5; p=30,20,15(pe rtürbasyon 5)	k=5; s=5,5,5; p=30,20,15(pertürb asyon 5+iyileştirme)	k=5; s=5,5,5; p=30,20,15(per türbasyon 10)	k=5; s=5,5,5; p=30,20,15(pertürb asyon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	3	3	3	3	3	3	3
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	3	3	3	3	3	3	3
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	3	3	3	3	3	3	3
28	93	93	93	93	93	93	93
29	0	0	0	0	0	0	0
30	2	2	2	2	2	2	2

Çizelge A.24. Problem tiplerine göre karşılaştırmalı sonuçlar 24

proble m no	k=5; s=5,5,5; p=35,25, 20	k=5; s=5,5,5; p=35,25,20(algotırma)	k=5; s=5,5,5; p=35,25,20(pe rtürbasyon 5)	k=5; s=5,5,5; p=35,25,20(pertürba syon 5+iyileştirme)	k=5; s=5,5,5; p=35,25,20(per türbasyon 10)	k=5; s=5,5,5; p=35,25,20(pertürb asyon 10+iyileştirme)	en iyi değer
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	8	8	8	8	8	8	8
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	12	12	12	12	12	12	12
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	3	3	3	3	3	3	3
12	0	0	0	0	0	0	0
13	3	3	3	3	3	3	3
14	5	8	5	5	5	5	5
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	4	4	4	4	4	4	4
21	0	0	0	0	0	0	0
22	5	5	5	5	5	5	5
23	8	8	8	8	8	8	8
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	10	10	10	10	10	10	10
28	128	128	128	128	128	128	128
29	0	0	0	0	0	0	0
30	11	11	11	11	11	11	11

EK B. CPLEX OPL İÇİN UYGULANAN KOD

```
int NbP=...;
range period=0..NbP;

int NbO=...;
range order=1..NbO;

int NbT=...;
range tip=1..NbT;

int duedate[tip,order]=...;
int S[tip]=...;
int K=...;
int P[tip]=...;

dvar float+ H[tip,period];
dvar float+ B[tip,period];
dvar boolean X[tip,order,period];
dvar float+ Y[tip,period];

minimize sum(j in tip,i in order, t in period: t<=duedate[j][i])((duedate[j][i]-t)*X[j,i,t]);
subject to{

forall(i in order,j in tip)
    ct1:
        sum(t in period:t>=1&& t<=duedate[j,i])
            X[j,i,t]==1;
forall(t in period,j in tip:t>=1)
    ct2:
```

```

B[j,t]==B[j,t-1]+(sum(i in order)X[j,i,t])-Y[j,t];

forall(t in period, j in tip:t>=1&& t<P[j])
  ct3:
    H[j,t]==H[j,t-1]-(sum(i in order)(X[j,i,t]));
forall(t in period,j in tip: t>=P[j])
  ct4:
    H[j,t]==H[j,t-1]-(sum(i in order)(X[j,i,t]))+Y[j,t-P[j]];
forall(t in period:t>=1)
  ct5:
    sum(z in period,j in tip : z>=t && z<=t+P[j]-1)Y[j,z]<=K;
forall(t in period, j in tip:t>=1&& t<=P[j])
  ct6:
    sum(i in order) X[j,i,t]<=H[j,t-1];
forall( t in period, j in tip:t>P[j])
  ct7:
    sum(i in order)X[j,i,t]<=H[j,t-1]+Y[j,t-P[j]];
forall(j in tip)
  {
    ct8:
    B[j,0]==0;
    H[j,0]==S[j];
    Y[j,0]==0;

  }

};

main { // main
for(var i=1; i<=20; i++){ // for

```

```

var starttime;

var start= new Date();
starttime=start.getTime();

var a= "data";

var source = new IloOplModelSource("Tez_Cok_Urun.mod");

var cplex = new IloCplex();

var def = new IloOplModelDefinition(source);

var opl = new IloOplModel(def,cplex);

var veri= a+i+".dat";

var data = new IloOplDataSource(veri);

opl.addDataSource(data);

opl.generate();

if (cplex.solve()) {
    write(veri  +" OBJ = " + cplex.getObjValue());
} else {
    writeln("No solution");
} //end of else

var after = new Date();

writeln (" Solving time is ", after.getTime()-starttime);

opl.end();

data.end();

def.end();

cplex.end();

source.end();

```

```
} // for  
} // main
```

EK C. ÖNERİLEN SEZGİSEL İÇİN UYGULANAN KOD

```
package solver;

import initialize.Problem;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Queue;

public class ProblemSolver {
    private List<Line> lines = new LinkedList<>();
    private List<Product> products = new LinkedList<>();
    private List<Pair<Integer, Product>> values;
    private Map<Product, Queue<Integer>> jokerNumbers;
    private Problem problem;

    public ProblemSolver(Problem problem) {
        this.problem = problem;
        List<List<Integer>> data = problem.getData();

        for (int i = 0; i < problem.getLineCount(); i++) {
            Line line = new Line();
            lines.add(line);
        }
        for (int i = 0; i < data.size(); i++) {
            Product product = new Product();
            product.setStockCount(problem.getStocks().get(i));

            product.setMaintenanceTime(problem.getMaintenanceTimes().get(i));
            product.setData(problem.getData().get(i));
            products.add(product);
        }

        values = new ArrayList<>();
        jokerNumbers = new HashMap<>();

        for (Product product : products) {
            for (Integer integer : product.getData()) {
                values.add(new Pair<>(integer, product));
            }
            jokerNumbers.put(product, new LinkedList<Integer>());
        }

        Collections.sort(values, new Comparator<Pair<Integer, Product>>() {
            @Override
```



```

        public int compare(Pair<Integer, Product> o1, Pair<Integer,
Product> o2) {
            return o2.getL() - o1.getL();
        }
    });
}

public int solve() {
    int result = 0;
    for (Pair<Integer, Product> pair : values) {
        Queue<Integer> queue = jokerNumbers.get(pair.getR());
        queue.add(pair.getL());
        if (pair.getR().getStockCount() > 0) {
            pair.getR().decreaseStock();
        } else {
            Integer remove = queue.remove();
            int delay = requestLine(remove, remove -
pair.getR().getMaintenanceTime());
            int startTime = remove - pair.getR().getMaintenanceTime() -
delay;

            if (startTime < pair.getL()) {
                result += pair.getL() - startTime;
            }
            problem.checkStartTime(startTime);
        }
    }

    return result;
}

private int requestLine(int endTime, int startTime) {
    int minRequestDelay = Integer.MAX_VALUE;
    Line bestLine = null;

    for (Line line : lines) {
        int requestDelay = line.request(endTime, startTime);
        if (requestDelay < minRequestDelay) {
            bestLine = line;
            minRequestDelay = requestDelay;
        }
    }

    bestLine.confirmRequest();

    return minRequestDelay;
}

public List<Double> getUtilizations () {
    List<Double> rates = new LinkedList<>();
    for (Line line : lines) {

```

```
        rates.add(line.getUtilization());
    }
    return rates;
}
}
```

EK D. DATA KARIŞTIRMA SEZGİSELİ İÇİN UYGULANAN KOD

```
package solver;

import initialize.Problem;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Queue;
import java.util.Random;

public class ProblemSolver {
    private List<Line> lines;
    private List<Product> products;
    private List<Pair<Integer[], Product>> values;
    private Map<Product, Queue<Integer>> jokerNumbers;
    private Problem problem;

    private static final int DELTA_VALUE = 5;
    private static final int TRYOUT_NUMBER = 10_000;

    public ProblemSolver(Problem problem) {
        this.problem = problem;
    }

    private static void sortValues(List<Pair<Integer[], Product>> values) {
        Collections.sort(values, new Comparator<Pair<Integer[], Product>>()
        {
            @Override
            public int compare(Pair<Integer[], Product> o1, Pair<Integer[],
Product> o2) {
                return o2.getL()[1] - o1.getL()[1];
            }
        });
    }

    public int[] solve() {
        int min_result = Integer.MAX_VALUE;
        int average = 0;

        Random random = new Random();
```

```

    for (int tryCount = 0; tryCount < TRYOUT_NUMBER; tryCount++)
    {

        int result = 0;

        int delta = 0;

        lines = new LinkedList<>();
        products = new LinkedList<>();
        for (int i = 0; i < problem.getLineCount(); i++) {
            Line line = new Line();
            lines.add(line);
        }
        for (int i = 0; i < problem.getData().size(); i++) {
            Product product = new Product();
            product.setStockCount(problem.getStocks().get(i));

product.setMaintenanceTime(problem.getMaintenanceTimes().get(i));
            product.setData(problem.getData().get(i));
            products.add(product);
        }

        values = new ArrayList<>(150);
        jokerNumbers = new HashMap<>();

        for (Product product : products) {
            for (Integer integer : product.getData()) {
                values.add(new Pair<>(new Integer[] { integer,
integer }, product));
            }
            jokerNumbers.put(product, new
LinkedList<Integer>());
        }

        for (Pair<Integer[], Product> pair : values) {
            delta = random.nextInt(2 * DELTA_VALUE + 1) -
DELTA_VALUE;
            pair.getL()[1] += delta;
        }

        sortValues(values);

        for (Pair<Integer[], Product> pair : values) {
            Integer value = pair.getL()[0];
            Queue<Integer> queue =
jokerNumbers.get(pair.getR());
            queue.add(value);
        }
    }
}

```

```

        if (pair.getR().getStockCount() > 0) {
            pair.getR().decreaseStock();
        } else {
            Integer remove = queue.remove();
            int delay = requestLine(remove, remove -
pair.getR().getMaintenanceTime());
            int startTime = remove -
pair.getR().getMaintenanceTime() - delay;
            if (startTime < value) {
                result += value - startTime;
            }
            problem.checkStartTime(startTime);
        }
    }

    average += result;

    if (result < min_result) {
        min_result = result;
        problem.setValues(values);
    }
}

average /= TRYOUT_NUMBER;

return new int[] { min_result, average };
}

private int requestLine(int endTime, int startTime) {
    int minRequestDelay = Integer.MAX_VALUE;
    Line bestLine = null;

    for (Line line : lines) {
        int requestDelay = line.request(endTime, startTime);
        if (requestDelay < minRequestDelay) {
            bestLine = line;
            minRequestDelay = requestDelay;
        }
    }
    bestLine.confirmRequest();
    return minRequestDelay;
}

public List<Double> getUtilizations() {
    List<Double> rates = new LinkedList<>();
    for (Line line : lines) {
        rates.add(line.getUtilization());
    }
}

```

```
    }  
    return rates;  
  }  
}
```

EK E. DATA KARIŞTIRMA SEZGİSELİNİ İYİLEŞTİRMEK İÇİN UYGULANAN KOD

```
package solver;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Queue;
import java.util.Random;

import initialize.Problem;

public class ProblemSolver {
    private List<Line> lines;
    private List<Product> products;
    private List<Pair<Integer[], Product>> values;
    private Map<Product, Queue<Integer>> jokerNumbers;
    private Problem problem;

    private static final int DELTA_VALUE = 10;
    private static final int TRYOUT_NUMBER = 10_000;
    private static final int SHIFT_VALUE = 3;

    public ProblemSolver(Problem problem) {
        this.problem = problem;
    }

    private static void sortValues(List<Pair<Integer[], Product>> values) {
        Collections.sort(values, new Comparator<Pair<Integer[], Product>>()
        {
            @Override
            public int compare(Pair<Integer[], Product> o1, Pair<Integer[],
Product> o2) {
                return o2.getL()[1] - o1.getL()[1];
            }
        });
    }

    public int[] solve() {
        int min_result = Integer.MAX_VALUE;
        int min_result2 = Integer.MAX_VALUE;
        int min_result3 = Integer.MAX_VALUE;
    }
}
```

```

int min_result4 = Integer.MAX_VALUE;
int min_result5 = Integer.MAX_VALUE;
List<Pair<Integer[], Product>> minValues = null;
List<Pair<Integer[], Product>> minValues2 = null;
List<Pair<Integer[], Product>> minValues3 = null;
List<Pair<Integer[], Product>> minValues4 = null;
List<Pair<Integer[], Product>> minValues5 = null;
int maxDelayIndex = -1;
int maxDelayIndex2 = -1;
int maxDelayIndex3 = -1;
int maxDelayIndex4 = -1;
int maxDelayIndex5 = -1;
int startTime = Integer.MAX_VALUE;
int startTime2 = Integer.MAX_VALUE;
int startTime3 = Integer.MAX_VALUE;
int startTime4 = Integer.MAX_VALUE;
int startTime5 = Integer.MAX_VALUE;

int average = 0;

Random random = new Random();
for (int tryCount = 0; tryCount < TRYOUT_NUMBER; tryCount++)
{
    int result = 0;
    int delta = 0;
    resetValues();
    for (Product product : products) {
        for (Integer integer : product.getData()) {
            values.add(new Pair<>(new Integer[] { integer,
integer }, product));
        }
    }
    for (Pair<Integer[], Product> pair : values) {
        delta = random.nextInt(2 * DELTA_VALUE + 1) -
DELTA_VALUE;
        pair.getL()[1] += delta;
    }
    sortValues(values);
    int[] results = solve(values, problem);
    result = results[0];
    if (result < min_result) {
        maxDelayIndex = results[1];
        min_result = result;
        problem.setValues(values);
        minValues = getCopyOfValues(values);
    }
}
}

```



```

    } else if (result < min_result2) {
        maxDelayIndex2 = results[1];
        min_result2 = result;
        problem.setValues(values);
        minValues2 = getCopyOfValues(values);
    } else if (result < min_result3) {
        maxDelayIndex3 = results[1];
        min_result3 = result;
        problem.setValues(values);
        minValues3 = getCopyOfValues(values);
    } else if (result < min_result4) {
        maxDelayIndex4 = results[1];
        min_result4 = result;
        problem.setValues(values);
        minValues4 = getCopyOfValues(values);
    } else if (result < min_result5) {
        maxDelayIndex5 = results[1];
        min_result5 = result;
        problem.setValues(values);
        minValues5 = getCopyOfValues(values);
    }
    average += result;
}

int aboveZeroMinimum = -1;

for (int i = 1; i <= SHIFT_VALUE; i++) {
    for (int j = 0; j < 5; j++) {
        int solve[];
        int result = 0;
        switch (j) {
            case 0:
                resetValues2(minValues);
                shiftPairs(maxDelayIndex, minValues);
                solve = solve(minValues, problem);
                result = solve[0];
                if (result < min_result) {
                    min_result = result;
                }
                startTime = problem.getStartTime();
                break;
            case 1:
                resetValues2(minValues2);
                shiftPairs(maxDelayIndex2, minValues2);
                solve = solve(minValues2, problem);
                result = solve[0];

```

```

        if (result < min_result2) {
            min_result2 = result;
        }
        startTime2 = problem.getStartTime();
        break;
    case 2:
        resetValues2(minValues3);
        shiftPairs(maxDelayIndex3, minValues3);
        solve = solve(minValues3, problem);
        result = solve[0];
        if (result < min_result3) {
            min_result3 = result;
        }
        startTime3 = problem.getStartTime();
        break;
    case 3:
        resetValues2(minValues4);
        shiftPairs(maxDelayIndex4, minValues4);
        solve = solve(minValues4, problem);
        result = solve[0];
        if (result < min_result4) {
            min_result4 = result;
        }
        startTime4 = problem.getStartTime();
        break;
    case 4:
        resetValues2(minValues5);
        shiftPairs(maxDelayIndex5, minValues5);
        solve = solve(minValues5, problem);
        result = solve[0];
        if (result < min_result5) {
            min_result5 = result;
        }
        startTime5 = problem.getStartTime();
        break;

    default:
        break;
    }

    if (problem.getStartTime() >= 0 && result <
aboveZeroMinimum) {
        result = aboveZeroMinimum;
    }
}
}

```

```

        average /= TRYOUT_NUMBER;

        problem.setStartTime(
            getEdgeValue(false, startTime, startTime2, startTime3,
            startTime4, startTime5));

        if (aboveZeroMinimum != -1) {
            return new int[] { aboveZeroMinimum, average };
        }

        return new int[] {
            getEdgeValue(false, min_result, min_result2,
            min_result3, min_result4, min_result5),
            average };
    }

    private static int getEdgeValue(boolean isBigger, int... numbers) {
        int edge;
        if (isBigger)
            edge = Integer.MIN_VALUE;
        else
            edge = Integer.MAX_VALUE;

        for (int i : numbers) {
            if (isBigger) {
                if (i > edge)
                    edge = i;
            } else if (i < edge)
                edge = i;
        }
        return edge;
    }

    private static void shiftPairs(int maxDelayIndex, List<Pair<Integer[],
    Product>> values2) {
        Pair<Integer[], Product> remove = values2.remove(maxDelayIndex);
        values2.add(maxDelayIndex - 1, remove);
    }

    private void resetValues() {
        lines = new LinkedList<>();
        products = new LinkedList<>();
        for (int i = 0; i < problem.getLineCount(); i++) {
            Line line = new Line();
            lines.add(line);
        }
    }

```

```

        for (int i = 0; i < problem.getData().size(); i++) {
            Product product = new Product();
            product.setStockCount(problem.getStocks().get(i));

product.setMaintenanceTime(problem.getMaintenanceTimes().get(i));
            product.setData(problem.getData().get(i));
            products.add(product);
        }

        values = new ArrayList<>(150);
        jokerNumbers = new HashMap<>();

        for (Product product : products) {
            jokerNumbers.put(product, new LinkedList<Integer>());
        }
        problem.resetStartTime();
    }

private void resetValues2(List<Pair<Integer[], Product>> minValues) {
    lines = new LinkedList<>();
    for (int i = 0; i < problem.getLineCount(); i++) {
        Line line = new Line();
        lines.add(line);
    }
    jokerNumbers = new HashMap<>();

    for (Pair<Integer[], Product> pair : minValues) {
        Queue<Integer> queue = jokerNumbers.get(pair.getR());
        if (queue == null) {
            jokerNumbers.put(pair.getR(), new
LinkedList<Integer>());
        }
    }
    problem.resetStartTime();
}

private int[] solve(List<Pair<Integer[], Product>> values, Problem problem)
{
    int result = 0;
    int maxDelayIndex = 0;
    int maxDelay = Integer.MIN_VALUE;
    for (int i = 0; i < values.size(); i++) {
        Pair<Integer[], Product> pair = values.get(i);
        Integer value = pair.getL()[0];
        Queue<Integer> queue = jokerNumbers.get(pair.getR());
        if (queue == null || value == null) {

```

```

        System.out.println(jokerNumbers);
        System.out.println("queue=" + queue);
        System.out.println("value=" + value);
        System.out.println("pair=" + pair.getR());
    }

    queue.add(value);
    if (pair.getR().getStockCount() > 0) {
        pair.getR().decreaseStock();
    } else {
        Integer remove = queue.remove();
        int delay = requestLine(remove, remove -
pair.getR().getMaintenanceTime());
        if (delay > maxDelay) {
            maxDelay = delay;
            maxDelayIndex = i;
        }

        int startTime = remove -
pair.getR().getMaintenanceTime() - delay;
        if (startTime < value) {
            result += value - startTime;
        }
        problem.checkStartTime(startTime);
    }
}
return new int[] { result, maxDelayIndex };
}

private static List<Pair<Integer[], Product>> getCopyOfValues(
    List<Pair<Integer[], Product>> values2) {
    List<Pair<Integer[], Product>> values = new ArrayList<>();
    for (Pair<Integer[], Product> pair : values2) {
        values.add(pair);
    }
    return values;
}

private int requestLine(int endTime, int startTime) {
    int minRequestDelay = Integer.MAX_VALUE;
    Line bestLine = null;

    for (Line line : lines) {
        int requestDelay = line.request(endTime, startTime);
        if (requestDelay < minRequestDelay) {
            bestLine = line;
            minRequestDelay = requestDelay;
        }
    }
}

```

```
        }
    }
    bestLine.confirmRequest();
    return minRequestDelay;
}

public List<Double> getUtilizations() {
    List<Double> rates = new LinkedList<>();
    for (Line line : lines) {
        rates.add(line.getUtilization());
    }

    return rates;
}
}
```

EK F. PROBLEM TİPİNE UYGULANAN TALEP TARİHLERİ

181	43	874
355	531	547
1024	70	1084
168	1055	762
233	147	850
315	712	951
1049	592	1076
346	712	636
971	181	1051
778	610	453
1008	902	322
392	655	88
554	1039	668
604	1089	291
186	349	82
161	740	969
564	124	485
935	691	863
305	503	955
298	815	397
69	1062	1009
559	40	312
233	1045	761
765	312	430
338	359	397
74	336	1034
241	578	504
76	837	424
411	658	668
1079	80	685
532	895	60
328	571	181
236	824	74
437	855	132
1052	632	533
628	757	60
128	950	262
433	261	401
526	264	723
363	428	79
496	405	93
246	65	756
176	556	481

841	500	456
500	392	963
750	701	1040
890	197	338
553	71	468

1041	907	287
592	324	168
570	254	1011
270	48	688
70	605	764
331	811	640
273	111	923
529	752	206
606	322	1018
482	170	158
786	364	401
396	696	263
1040	963	629
681	327	616
911	744	963
251	313	1074
53	236	624
653	848	693
827	885	335
906	39	256
829	762	755
138	521	545
268	712	250
742	1068	57
170	343	669
467	896	280
801	446	819
1010	823	519
178	729	291
492	276	217
338	904	506
833	554	153
865	756	77
177	247	696
360	927	294
464	676	437
732	911	658
81	198	409
229	330	424

471	118	1007
1088	988	537
998	634	731
455	830	437
998	824	45
537	575	79
267	707	137
409	117	538
522	493	826

407	78	1073
520	1060	261
876	234	890
302	1002	972
1007	609	746
205	361	470
867	869	994
642	784	291
691	250	916
911	963	428
804	232	227
143	447	354
297	292	445
1032	205	628
420	550	925
197	212	1093
955	402	1041
449	416	402
1075	1027	955
137	167	1044
631	752	284
905	900	345
571	915	980
514	624	848
65	923	827
83	458	113
291	564	312
867	946	676
190	731	750
910	876	572
194	712	289
683	486	1032
648	931	772
380	358	230
528	872	241

1086	387	148
362	400	810
945	394	478
646	902	802
592	230	1076
953	660	309
189	1024	810
780	136	548
72	917	115
811	599	330
326	168	931
457	977	1031
755	764	46

136	724	642
113	533	362
441	1056	1058
581	167	314
285	406	1072
925	883	569
1027	338	835
770	755	946
159	717	453
1045	1082	332
55	256	869
150	143	481
450	183	468
897	1078	852
697	917	1015
37	929	286
118	917	649
293	477	1012
1069	591	49
148	183	779
454	1063	463
263	100	342
523	307	925
410	220	154
143	568	161
222	817	45
851	154	907
456	350	86
995	167	315
551	551	1045
954	54	105

453	177	911
121	708	227
185	128	273
859	202	549
1045	158	1056
472	143	434
500	773	720
797	989	425
598	993	423
750	652	1008
721	489	273
104	1093	976
1062	591	477
859	943	366
287	1060	403
382	954	655
355	1094	329

738	1055	825
558	910	873
639	896	823
541	201	196
989	319	1022
822	722	41
289	749	820
132	89	240
157	939	630
201	568	559
352	403	552
945	881	214
586	600	294
870	944	541
443	83	836
202	41	699
542	614	277
606	573	1080
666	91	483
788	630	188
565	1090	39
731	154	658
422	268	243
1093	803	250
1071	931	641
583	1013	538
422	488	946

428	205	195
822	670	1016
361	1057	155
232	315	553
815	989	1040
114	459	810
887	781	637
856	564	306
1007	242	898
131	110	599
510	389	967
195	1050	443
603	129	605
771	517	623
766	549	222
775	75	311
983	1063	1087
166	489	736
272	987	600
288	161	160
446	817	723

364	328	786
214	649	945
285	965	183
639	695	740
1088	685	1003
549	569	121
682	286	1076
680	455	343
782	1014	181
364	1016	848
477	876	190
134	608	354
1089	844	647
529	525	1013
198	534	472
774	1065	722
56	338	226
475	230	921
989	807	637
992	842	92
140	959	270
853	1034	228
443	402	737

805	648	366
81	297	826
860	539	855
449	662	600
855	529	904
535	230	866
317	35	459
608	432	896
633	42	971
846	732	472
582	198	199
418	1008	860
68	157	740
223	927	276
604	839	120
997	542	496
570	805	1064
404	183	736
695	81	326
211	452	220
302	406	523
936	383	788
880	924	517
49	562	56
106	193	743

336	969	675
786	548	153
654	745	644
288	82	1096
915	60	1015
597	46	687
105	573	719
161	457	759
223	635	831
986	1061	738
696	1099	755
407	270	284
574	1040	1045
445	1065	696
869	369	84
821	975	516
753	267	1082
1073	892	899
776	534	350

36	1009	1059
613	1033	1075
583	574	255
229	658	854
45	233	504
218	922	732
69	625	252
376	165	488
617	450	827
173	499	100
600	524	388
245	976	863
362	224	797
636	945	176
784	479	726
989	663	47
94	557	175
239	613	164
207	1006	682
535	1031	674
237	1070	769
524	328	642
599	854	63
371	571	1004
893	290	50
880	517	922
397	865	907
138	155	186
822	39	528

856	722	820
83	439	952
1005	657	397
334	993	73
813	95	635
438	181	661
963	636	471
676	508	990
411	813	119
638	431	463
476	594	328
484	673	51
1074	186	45
128	1011	627
905	350	1002

31	169	318
177	143	308
93	474	735
964	729	845
1026	598	64
104	49	1057
379	872	711
902	332	405
363	295	972
1067	556	92
831	253	1051
1077	649	849
1052	133	941
572	1087	62
967	496	87
144	858	925
262	773	301
794	634	980
527	477	585
258	867	851
731	46	787
302	462	401
1004	397	455
944	300	560
31	215	706
30	450	287
195	898	241
757	457	345
490	753	198
419	351	156
168	472	686
538	508	122
1042	515	908
994	969	678
391	373	1096
575	252	970
165	95	286
75	261	806
291	58	964
132	639	546
402	606	362
205	841	880
1014	1015	361
253	870	174

709	431	610
57	106	437
698	870	74
908	607	249
249	811	524
523	392	272
187	674	468
742	891	761
735	96	414
530	747	387
303	382	705
1038	246	246
913	921	205
247	266	1024
718	114	255
942	64	521
661	808	149
69	1060	851
222	374	1015
479	566	907
378	445	355
562	787	69
420	796	386
902	267	389
731	481	514
996	466	595
535	969	277
515	185	866
43	278	319
1012	979	622
1048	737	225
859	666	274
853	391	571
239	855	960
835	827	1084
462	723	1012
1061	879	696
870	253	251
130	314	262
835	482	977
909	675	492
363	1004	141
1092	574	534
940	188	953
796	1012	557

806	326	92
170	507	362
666	921	786
332	1082	1017
885	607	844
741	627	569
438	327	556
1079	804	616
463	906	988
240	30	261
277	889	161
135	602	221
916	654	88
682	803	75
686	612	632
303	193	211
519	200	232
293	442	1010
91	1047	897
594	949	1013
757	741	368
305	662	375
442	1069	690
595	965	895
999	350	615
451	748	219
203	511	377
787	398	631
946	1080	1061
842	620	897
266	705	138
128	845	834
87	178	400
849	307	896
999	664	141
882	104	1031
1003	155	870
857	352	192
1097	487	126
586	617	471
1035	733	223
328	1067	259
234	897	568
435	67	350

184	1014	939
151	585	399
63	912	533
31	926	693
494	955	375
975	619	312
647	945	646
537	922	616
770	601	606
491	707	433
365	814	316
66	906	568
462	98	138
1044	1082	1100
126	156	443
1065	632	436
314	628	390
683	687	1056
494	992	856
75	301	249
552	298	897
94	959	864
1093	815	61
819	559	258
358	1024	310
1006	832	424
345	412	466
248	1097	268
548	408	254
333	305	970
277	985	533
858	71	796
939	461	693
493	561	1003
808	1097	867
411	134	666
554	956	977
1092	886	967
863	572	307
649	914	120
325	595	277
462	999	67
883	833	1019
1011	238	695

1044	884	21
453	64	245
316	858	58
241	1010	214
1029	1078	70
574	298	584
688	1004	555
812	656	492
303	860	547
360	419	251
585	949	995
796	510	70
376	86	499
909	541	281
939	209	971
98	1008	78
1096	847	631
463	31	666
80	75	887
637	54	168
655	626	1033
889	903	313
926	381	328
639	60	223
602	375	379
781	1011	574
79	729	286
707	345	797
106	733	853
305	600	864
229	630	720
551	188	774
771	815	246
992	636	785
483	1060	1079
929	1011	322
768	1032	779
601	86	798
910	48	566
738	891	626
305	251	651
30	85	132
1006	755	453
542	90	1095
211	858	547

1079	879	512
839	910	39
253	223	349
987	1087	869
876	975	240
1061	1051	208
130	272	871
580	1078	189
1044	795	545
641	253	593
949	887	876
849	118	569
303	508	88
344	1020	32
660	185	1074
378	713	361
91	253	522
904	827	169
34	271	810
695	969	186
96	233	277
35	426	77
1075	716	79
201	829	227
593	1042	175
492	437	633
896	345	702
176	616	494
1037	906	452
585	545	1093
271	187	138
980	674	463
853	584	141
376	398	532
114	286	884
631	502	592
366	891	125
275	688	268
989	454	785
987	662	397
715	329	442
472	74	822
496	339	867
781	821	184

156	336	238
160	112	696
763	43	281
76	497	426
872	604	532
867	753	876
345	747	305

54	169	639
873	1076	273
647	1068	836
654	283	557
1093	1007	346
964	1090	92
1046	967	767
448	555	593
99	838	220
774	488	285
873	170	807
585	532	765
164	390	616
473	256	959
835	327	569
321	711	1078
583	1031	63
785	76	635
167	50	442
451	400	758
471	501	963
161	628	838
444	110	792
617	1016	459
954	793	74
486	1014	39
761	92	877
341	467	350
198	267	768
511	254	397
501	668	1048
841	1039	535
836	626	540
976	73	643
493	447	610
105	1094	892
998	460	1083

342	464	669
198	1087	993
498	621	44
677	933	533
291	1083	366
1032	436	482
857	939	237
346	276	281
712	579	139
429	112	107
1078	860	504

763	890	284
651	728	313
859	241	780
849	197	240
563	622	198
34	1082	52
1027	483	486
1053	496	259
969	768	948
949	952	558
368	532	832
131	419	95
497	565	173
705	552	316
506	720	189
1067	609	379
972	591	35
287	332	248
337	1019	547
931	30	414
634	168	1081
740	1014	485
576	1055	753
557	1040	1026
760	505	169
643	873	242
944	1022	186
207	39	1032
1026	820	968
796	619	679
267	60	670
1007	214	683
582	599	457

359	674	105
835	553	515
156	1063	376
899	811	709
49	251	209
65	175	500
213	154	960
574	115	128
612	197	492
653	262	1016
536	626	392
286	308	278
470	416	801
492	689	970
930	926	274

1036	807	147
64	712	94
475	43	44
917	530	1090
755	829	134
129	78	110
699	837	423
1034	835	227
371	940	649
178	1084	368
64	1057	683
499	910	843
431	874	898
204	433	694
280	296	121
270	679	511
912	1061	221
163	86	715
690	116	319
130	365	183
157	713	733
255	553	267
503	443	185
741	289	371
239	76	461
469	542	782
582	385	1053
640	917	660
326	630	473

834	476	348
308	840	1008
861	744	254
211	804	369
1036	539	144
725	194	280
1000	168	628
568	888	530
940	31	370
94	861	605
640	1036	912
32	966	332
798	228	46
64	913	996
819	1088	494
783	558	74
539	517	114
67	81	1087
181	185	423

749	1063	503
485	819	204
221	681	629
410	738	625
568	147	179
33	778	1100
1031	513	758
800	184	541
742	816	1043
482	875	243
539	471	993
376	358	672
334	723	228
169	72	580
958	41	1001
684	172	474
292	123	777
377	385	1094
733	979	936
661	656	741
35	624	612
225	903	200
1008	406	858
485	306	510
220	513	226

671	90	437
591	177	935
843	552	1090
206	176	377
872	314	50
553	81	553
68	35	541
983	381	753
160	208	686
165	689	34
974	351	1061
1046	1070	357
372	915	261
982	548	510
340	974	1039
145	217	276
1081	553	426
1064	374	360
438	865	1004
383	183	1072
440	63	551
1056	605	696
711	910	775

904	846	205
449	252	909
645	531	1088
212	762	225
549	630	107
849	251	998
708	484	219
220	61	295
714	1056	421
902	63	283
1073	181	317
578	1074	72
750	110	968
530	272	455
104	407	966
317	567	975
97	1046	450
31	790	940
261	997	1025
283	402	865
830	960	297

1060	869	436
698	570	562
721	96	912
335	843	879
507	77	37
130	1041	669
829	946	50
702	144	568
530	671	584
467	456	600
610	467	511
223	313	919
809	474	420
287	43	347
548	838	184
740	950	984
1096	805	398
155	855	1085
144	1045	395
342	953	651
435	254	713
656	905	796
1055	1071	474
102	584	886
810	229	410
164	138	428
1083	960	66

69	162	593
1028	790	416
1071	503	421
396	393	767
1070	836	520
278	296	962
57	1099	124
163	959	779
142	654	531
516	808	199
506	279	789
651	920	141
75	506	470
170	582	119
180	288	827
1013	576	308
408	921	155

636	650	252
594	1080	687
221	800	321
757	472	79
263	808	153
928	804	668
231	473	1009
439	1043	995
130	540	1021
838	1046	474
307	601	510
630	619	387
1079	447	192
236	198	54
463	575	155
324	129	436
165	1100	733
633	1047	725
185	567	953
303	367	671
818	991	173
106	772	626
548	308	38
589	894	824
960	503	615
759	557	247
1043	1002	345
894	523	186
463	456	58
489	191	804
578	752	738
572	536	474
940	719	704
243	277	846
416	867	387
639	930	193
608	138	379
1078	299	205
539	934	50
233	876	923
39	575	614
732	53	1028
292	920	768
871	698	1052

552	466	922
608	230	417
136	847	496
478	476	913
758	642	872
1055	92	1066
181	44	1076
369	951	749
613	618	103
537	879	600
354	1039	423
460	493	51
994	321	78
348	802	309
379	259	287
297	853	682
42	170	41
102	312	962
859	144	594
980	564	1039
800	677	912
368	208	266
127	1050	329
42	513	794
415	540	697
436	1061	1015
405	348	261
977	546	775
213	932	1021
453	910	984
412	454	293
357	558	292
945	425	592
75	890	698
140	660	366
701	1089	119
524	771	198
395	71	670
320	971	470
721	744	1093
769	424	1062
917	207	1035
724	888	57
329	296	562

866	165	292
1097	146	390
87	874	986
997	950	520
187	951	682
746	101	131
287	713	44
924	867	938
452	276	693
197	267	103
1099	853	365
30	883	367
674	897	1005
188	110	148
757	821	763
281	703	703
691	397	1023
567	960	103
94	170	1035
929	953	653
235	1050	429
915	1064	105
644	720	649
613	841	327
400	665	163
765	688	62
88	457	846
970	427	354
852	134	386
542	763	765
962	329	350
632	227	287
949	721	880
689	745	176
410	862	103
506	118	895
861	169	985
1059	1085	1090
32	56	162
679	789	952
785	853	784
727	371	159
1035	737	952
1044	106	368

82	650	965
392	1068	388
594	756	87
957	325	54
55	372	649
137	919	690
488	106	541
206	602	1038
871	35	90
145	374	753
1086	272	864
937	423	264
507	100	672
325	48	1043
893	84	531
482	316	894
195	866	843
892	856	351
268	849	54
368	363	726
95	314	574
33	435	406
616	98	467
846	370	125
429	894	861
291	601	816
640	483	72
643	67	450
350	337	706
211	736	852
906	720	400
665	236	783
488	527	504
195	562	182
188	858	473
264	672	81
219	824	746
582	387	405
746	152	468
255	282	342
755	778	331
279	835	983
271	1005	1086
619	828	99

376	565	440
166	211	140
686	1026	438
903	678	972
73	111	97
645	342	122
52	47	817
328	646	835
979	660	111
646	460	430
449	599	96
64	203	841
582	1070	417
952	746	210
204	770	1073
508	1011	912
1026	829	112
957	508	497
912	670	239
338	282	275
136	699	880
488	1035	42
637	304	587
734	818	794
1054	562	43
146	307	613
838	230	589
1082	1080	974
438	858	51
530	1006	1058
637	451	592
246	210	188
414	34	1038
791	756	869
231	284	495
235	339	900
66	552	549
1027	166	1075
967	380	431
767	1024	663
643	813	580
201	131	994
33	449	899
1048	413	726
797	542	443

1034	40	627
95	375	925
846	970	110
712	380	806
878	464	818
238	228	784
1032	747	718
542	483	465
235	666	143
597	743	776
770	506	790
915	787	1050
731	433	357
452	824	807
990	421	97
753	846	676
906	640	998
191	1011	476
630	438	445
231	896	518
957	447	639
1049	402	292
667	368	860
586	193	179
467	430	454
885	156	725
214	672	962
382	271	441
177	388	573
802	987	634
290	590	147
117	1031	432
648	541	612
82	582	612
1046	496	339
367	209	355
1032	1007	828
941	1076	1084
393	468	419
62	927	812
137	33	264
186	685	359
804	340	541
508	151	154

772	164	995
154	385	133
121	1039	1083
635	473	250
1031	524	891
262	845	430

294	458	288
46	1066	574
1007	494	36
484	276	451
446	586	498
635	165	470
293	1040	913
346	565	64
1087	355	129
296	427	913
932	868	900
481	721	708
474	705	82
738	248	648
73	1052	240
690	435	722
516	79	636
899	143	1049
30	620	225
452	776	231
375	900	396
691	592	287
963	286	856
846	422	458
1001	42	97
1017	914	234
692	1075	534
870	776	209
962	418	862
293	762	578
542	219	926
633	284	959
610	470	891
508	108	91
217	478	189
878	197	664
383	717	940
537	245	843

983	201	987
1061	899	1008
328	413	789
681	872	1031
156	785	308
42	1029	1021
971	453	369
453	332	140
276	129	594
171	345	517

219	806	1015
120	732	72
95	407	156
672	911	364
405	628	561
797	622	50
260	705	1017
420	664	848
713	814	210
512	190	479
67	466	896
219	423	37
143	529	540
397	763	953
378	365	1095
981	719	1069
1088	275	653
801	52	544
917	624	561
156	509	1034
1038	218	992
347	87	43
554	907	310
61	111	950
364	98	321
51	547	672
250	727	183
1026	659	466
733	197	653
815	702	470
1085	1064	915
298	625	171
815	1005	50
295	816	1032

89	419	782
665	1060	773
1017	919	297
44	170	90
638	1070	938
174	467	345
987	456	397
837	778	91
200	501	946
277	321	321
858	476	590
1026	309	622
182	48	908
1030	566	691

492	827	880
131	531	785
57	678	514
1010	680	944
222	436	85
876	681	190
886	409	176
404	924	395
171	1066	100
219	166	930
507	608	629
261	235	274
365	230	400
414	111	224
564	446	407
403	1060	75
494	859	1058
279	847	318
669	499	207
592	319	358
1082	90	930
562	559	481
906	92	270
846	372	713
478	583	638
245	93	1090
228	649	642
888	1083	764
877	890	1006
937	166	965

47	261	83
668	704	726
809	710	570
1069	243	1099
755	52	93
46	142	560
707	113	283
814	396	182
401	733	143
831	169	693
1063	102	501
407	314	338
1090	36	766
744	442	564
104	876	980
1085	225	756
1084	405	302
517	481	773

460	761	86
79	589	371
1096	96	193
983	136	691
837	231	699
1093	637	942
443	1040	276
357	278	413
1088	684	201
397	449	793
299	967	456
55	826	608
378	795	329
777	119	454
1098	889	717
1053	665	850
1081	988	872
1089	787	93
273	1018	812
845	379	325
63	754	272
459	873	599
385	968	540
631	472	976
912	399	783
711	291	652

707	892	216
1086	123	823
633	507	682
865	907	942
795	579	833
729	827	636
114	118	234
734	819	417
598	1021	259
334	460	559
468	186	537
1059	794	571
182	890	204
841	756	194
741	615	734
604	518	921
422	150	775
299	585	433
1095	104	382
785	371	763
765	889	389
1085	974	992

439	160	21
245	1075	670
909	564	379
156	1020	771
115	207	841
166	1095	775
1069	1051	157
702	956	336
473	976	449
747	1005	870
259	1003	918
88	326	179
1011	64	923
476	127	799
710	69	994
293	861	337
413	1060	600
808	98	157
178	345	501
732	102	342
77	316	854
154	595	858

937	404	921
53	862	391
768	187	767
756	1045	986
903	864	150
883	477	753
346	415	423
779	468	525
340	867	703
610	303	847
651	461	476
679	1056	124
879	925	719
777	557	1040
414	45	64
515	443	957
419	851	236
691	574	762
83	579	503
247	627	828
1085	616	677
348	460	664
360	277	225
590	253	231
987	972	426
1086	275	480

1047	855	123
1097	833	967
190	1016	614
412	218	947
588	1052	75
376	877	881
358	99	549
247	37	305
903	353	973
1061	297	530
593	63	581
589	134	79
149	623	776
1020	280	553
46	125	251
624	271	307
121	302	565
107	528	286

230	670	201
788	609	494
786	819	340
362	853	791
228	851	653
292	540	158
819	296	461
141	185	749
270	366	939
996	217	873
709	87	108
860	296	568
368	597	1037
107	354	374
486	870	300
1021	790	420
409	31	146
759	79	1041
624	297	1007
372	1027	422
462	677	287
270	822	1020
585	865	195
73	758	720
163	1045	955
606	538	179
244	591	92
578	164	865
962	454	287
73	548	294

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÜLKER, Doğukan Hazar
Uyruğu : T.C.
Doğum tarihi ve yeri : 12.07.1989, Keşan
Medeni Hali : Evli
Telefon : +90 533 485 94 47
E-mail : dogukan.ulker@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	Dokuz Eylül Üniversitesi Metalurji ve Malzeme Mühendisliği Bölümü	2012

İş Deneyimi

Yıl	Yer	Görev
2012-2012	Dokuz Eylül Üniversitesi EMUM	Araştırma Mühendisi
2013-2013	MAN Finans	Proje Elemanı
2013-halen	Hidrosil Hidrolik Silindir	İş Geliştirme Müdürü

Yabancı Dil

İngilizce

Yayınlar

Ülker D.H., Erkoç M., Ertoğral K., “Havayolları Tamir Bakım Servis Sağlayıcıları İçin Bir Üretim Çizelgeleme Modeli ve Analizi”,2015, Gazi Univ. Müh. Mim. Fak. Der. Vol.30, No:3, 513-522,