

**ENDÜSTRİ MÜHENDİSLİĞİ**  
**TMMOB MAKİNA MÜHENDİSLERİ ODASI**  
**YAYINIDIR**

**3 Ayda Bir Yayınlanır**  
**Yerel Süreli Yayın**  
**Hakemli Bir Dergidir**

**OCAK-HAZİRAN 2018**  
January/June

Cilt / Vol: 29 Sayı / No: 1-2

**Makina Mühendisleri Odası Adına Sahibi**  
**Publisher**  
Yunus YENER

**Sorumlu Yazı İşleri Müdürü / Executive Editor**  
Yunus YENER

**Yayın Sekreteri / Editorial Secretary**  
Aylin Sıla AYTEMİZ

**Baş Editör / Editor in Chef**  
Ferda Can ÇETİNKAYA - Çankaya Üniversitesi

**Yardımcı Editörler / Associate Editors**  
Özgür YALÇINKAYA - Dokuz Eylül Üniversitesi  
Güzin ÖZDAĞOĞLU - Dokuz Eylül Üniversitesi

**Yayın Kurulu / Editorial Board**  
Bülent DURMUŞOĞLU - İstanbul Teknik Üniversitesi  
Aydın KAYNARCA - Ortadoğu Teknik Üniversitesi  
Olca POLAT - Pamukkale Üniversitesi

**Yayın Danışma Kurulu / Editorial Advisory Board**  
Cafer ÇELİK - Atatürk Üniversitesi  
Emin KAHYA - Eskişehir Osmangazi Üniversitesi  
Fikri EGE - Toros Üniversitesi  
Hakan ÖZCAN - Ondokuz Mayıs Üniversitesi  
Kemal YILDIZLI - Ondokuz Mayıs Üniversitesi  
Zerrin ALADAĞ - Kocaeli Üniversitesi

**Redaksiyon / Redaction**  
Burak KAYAOĞLU

**Teknik Sorumlu / Technical Manager**  
Mehmet AYDIN

**Sayfa Tasarımı / Page Design**  
Münevver POLAT

**Baskı / Printed By**  
Ankamat Matbaacılık Sanayi Ltd. Şti.  
30. Cadde 538. Sokak No:60  
İvedik Organize Sanayi - ANKARA  
Tel: (0 312) 394 54 94

**Baskı Sayısı / Circulation**  
3000

**Baskı Tarihi / Publishing Date**  
9 Kasım 2018

**Yönetim Yeri / Head Office**  
TMMOB Makina Mühendisleri Odası  
Meşrutiyet Cad. 19/6.Kat Kızılay-ANKARA  
Tel: 0 850 495 0 666 (06)  
Fax: 0 312 417 86 21  
e-posta: yayin@mmo.org.tr  
http://www.mmo.org.tr/endustri



Endüstri Mühendisliği dergisi EBSCO Veri Tabanında ve ayrıca International Abstracts in Operations Research tarafından taranmaktadır.

## S U N U Ş

Değerli Okuyucularımız,

2018 yılının ilk sayısında iki makaleyle birlikte Makina Mühendisleri Odası tarafından düzenlenen Endüstri İşletme Mühendisliği Kurultayı Sonuç Bildirisine, Online makale yönetim sistemimiz ve geçtiğimiz yılın hakem listesi ve yazı dizini yer almaktadır.

Bu sayımızda yer alan ilk makale, Tuğba Saraç, Kumsal Erten ve Elif Yılmaz tarafından yazılan “Kaynak Makinelerinin Vardiyada Çizelgelenmesi Problemi İçin İki Aşamalı Bir Çözüm Yaklaşımı” başlıklı çalışmadır. Makalede, beyaz eşya üreticisi bir işletmenin bünyesinde bulunan kaynak makinelerinde ürünlerin hangi makinelerde, hangi sırayla üretilmesi gerektiğinin belirlendiği probleminin özel bir hali ele alınmıştır. Literatürde yer alan çalışmalarda genellikle işlerin sırası belirlenmekte ancak işlerin hangi vardiyada üretileceği dikkate alınmamaktadır. Bu çalışmada önerilen model hem işlerin hangi makinada hangi sıraya atanacağını hem de üretileceği vardiyayı belirlemektedir.

Ezgi Eroğlu ve Meral Azizoğlu tarafından yazılan “Yönlü İki Objektifli Çinli Postacı Problemi İçin Kesin Çözüm Yaklaşımları” başlıklı ikinci makalede, iki toplamsal kriterli yönlü çinlipostacı problemi ele alınmış ve tüm bastırılmayan objektif vektörlerini yaratmak için iki çözüm yaklaşımı geliştirilmiştir. Birinci yaklaşım, klasik yaklaşım, karmaşık kesikli doğrusal programların optimal çözümlerini kullanmakta ve bastırılmayan objektif vektör setini seri olarak yaratmaktadır. İkinci yaklaşım, dal ve sınır algoritması, doğrusal programlama gevşetmelerinin optimal çözümlerini kullanmakta ve bastırılmayan objektif vektör setindeki çözümleri aynı anda yaratmaktadır.

9. Endüstri İşletme Mühendisliği Kurultayı sonuç bildirisini, önümüzdeki dönem gerçekleşecek olan Kurultay haberini de dergimiz sayfalarından okuyabilirsiniz.

Dergimizde yayımlanan makalelere <http://www.mmo.org.tr/endustri> adresinden erişebilirsiniz; makale ve yazılarınızı <http://omys.mmo.org.tr/> endustri adresinden bize iletebilirsiniz.

Dergimizin bu sayısında çalışmalarına yer verdiğimiz yazarlara ve çalışmaların değerlendirme sürecine katkı sağlayan hakemlerimize teşekkür ederiz.

İyi okumalar dileğiyle,

**TMMOB Makina Mühendisleri Odası**  
**Yönetim Kurulu**

# KAYNAK MAKİNELERİNİN VARDİYA BAZINDA ÇİZELGELENMESİ PROBLEMİ İÇİN İKİ AŞAMALI BİR ÇÖZÜM YAKLAŞIMI

Tuğba Saraç\*, Kumsal Erten, Elif Yılmaz

Eskişehir Osmangazi Üniversitesi, Endüstri Mühendisliği Bölümü, Eskişehir  
tsarac@ogu.edu.tr, kumsal.erten@gmail.com, elifyilmaz@gmail.com

Geliş Tarihi: 14.11.2015; Kabul Ediliş Tarihi: 02.05.2018

## ÖZ

Bu çalışmada, beyaz eşya üreticisi bir işletmenin bünyesinde bulunan kaynak makinelerinde ürünlerin hangi makinelerde, hangi sırayla üretilmesi gerektiğinin belirlendiği  $n$  ürün,  $m$  ilişkisiz paralel makina çizelgeleme probleminin özel bir hali ele alınmıştır. Literatürde yer alan çalışmalarda genellikle işlerin sırası belirlenmekte ancak işlerin hangi vardiyada üretileceği dikkate alınmamaktadır. Oysa gerçek hayatta vardiya bazında dikkate alınması gereken farklı kısıtlar söz konusudur. Bu çalışmada önerilen model hem işlerin hangi makina da hangi sıraya atanacağını hem de üretileceği vardiyayı belirlemektedir. Böylece vardiyaya özel kapasite kısıtı gibi durumları dikkate almak mümkün olabilmektedir. Ele alınan problemde, hazırlık süreleri ihmal edilebilecek kadar küçük olmasına rağmen, birbirine benzemeyen işlerin art arda üretilmeleri üretim sürecini zorlaştırması nedeniyle istenmemektedir. Ürünlerin farklılıkları, en, boy, marka gibi özelliklerine bağlı olarak değişmektedir. Bu çalışmada öncelikle tüm ürünlerin ardışık üretilmeleri durumunda benzemezlik düzeylerini temsil edebilecek bir farklılık katsayısı tanımlanmış ve bu katsayının hesaplanabilmesi için bir yaklaşım geliştirilmiştir. Problemin amaçları son işin tamamlanma zamanının ve farklılık katsayılarının toplamının en küçüklenmesidir. Çok amaçlı yapıda olan ve sürece özel kısıtlar içeren problem için bir hedef programlama modeli önerilmiştir. Önerilen bütünlükmatematiksel model küçük boyutlu bir problem ve gerçek hayat problemi kullanılarak test edilmiştir. Gerçek hayat problemini bütünlük model ile çözebilme mümkün olmamıştır. Büyük problemlerin çözümü için bütünlük model yerine birbirini ardı sıra çalışacak iki aşamalı bir çözüm yaklaşımı önerilmiştir. Yöntemin ilk aşamasında işlerin atanacağı makineler belirlenmekte, ikinci aşamada ise, birinci aşamada belirlenen, makinalara atanmış işlerin, sıralarına ve vardiyalarına karar verilmektedir. Önerilen yaklaşım gerçek hayat probleminin çözülebilmesini mümkün kılmıştır.

**Anahtar Kelimeler:** İlişkisiz paralel makine çizelgeleme problemi, vardiya bazlı çizelgeleme, hedef programlama, kaynak makineleri

## A TWO STAGE SOLUTION APPROACH FOR THE SHIFT-BASED SCHEDULING PROBLEM ON WELDING MACHINES

### ABSTRACT

In this study, one of the special version of  $n$  product,  $m$  unrelated parallel machine scheduling problem is considered. This problem occurs in a firm which produces white goods and it is determining that which job produces at which welding machine and in what order. Generally, alignment of workload is determined on scheduling problems but the shifts of the workload are not considered. However, in real-life, there are some restrictions that needs attention, in shift basis. Therefore, this study proposes a model both scheduling the jobs on which machine needs to be used and align it correctly, also designates the shift of production. With this model, special circumstances such as specific capacity for a shift could be considered. In this problem, there are preparations periods depending on alignment. These periods are changing depending on the weight, height and brand of product. In this study first, a coefficient of variation has been identified that could represent their preparation time and an approach has been developed to calculate these coefficients. In this way, measurement of setup time will consist of in case of sequential production of all products requirements can be eliminated. Purpose of this problem is minimizing the sum of finishing period of the last assignment and discrepancy factor. A goal programming model is developed for this multi-objective problem which includes special process constraints. This suggested mathematical model is tested with small sized and real-life problems. A solution is obtained for the small sized problem but real-life problem could not be solved with GAMS/CPLEX. Therefore, the two-stage approach which will work successively has been proposed instead of integrated models for the solution of major problems. In the first stage of the procedure, Works are assigned to shifts. In the second stage, a mathematical model is solved for each shifts. The real life problem can be solved by proposed solution approach.

**Keywords:** Unrelated parallel machine scheduling problem, shift-based scheduling, goal programming, welding machines

\* İletişim yazarı

TÜBİTAK 2241/A (2209/B) - Sanayi Odaklı Lisans Bitirme Tezi Destekleme Programı 2015 Yılı 5. Döneminde TÜBİTAK tarafından destek almıştır.

## 1. GİRİŞ

Bir üretim sisteminde çalışanların ve kullanılan makinelerin etkinlik ve verimliliği üretim sürecini önemli derecede etkiler. Etkin ve verimli bir sürecin yaratılmasında birçok faktör rol oynamaktadır. Bunlardan önemli bir tanesi de çizelgelemedir.

Çizelgeleme fonksiyonu bir şirkette matematiksel veya sezgisel teknikler yardımı ile işlerin gerçekleştirilmesinde sınırlı kaynakların tahsis edilmesine olanak sağlar. Kaynakların uygun tahsisi ise şirketin amaçlarını eniyilemesine ve hedeflerine erişmesine olanak sağlar (Baker, 1974; Eren ve Güner, 2002).

Çizelgeleme problemleri; parametrelerin deterministik (belirli) ya da stokastik (belirsiz) olması üretim ortamının tek makinalı ya da çok makinalı olması, geliş sürecinin durağan (statik) ya da dinamik olmasına göre değiştiği çeşitli problem yapılarını kapsar (Eren ve Güner, 2002). Çizelgeleme problemlerinde genellikle sonlu sayıda makine olduğu kabul edilmektedir ve iş sayısı  $n$  ve makine sayısı ise  $m$  ile ifade edilmektedir (Pinedo, 2016).

Bir çizelgeleme probleminin en çok kabul gören sınıflandırma biçimi  $\alpha | \beta | \gamma$  üçlü gösterimidir. Bu üçlü gösterimde;  $\alpha$  makine ortamını,  $\beta$  problemde ele alınan kısıtları,  $\gamma$  ise amaç fonksiyonunu ya da fonksiyonlarını göstermektedir. Bu çalışmada ele alınan problem, ilişkisiz paralel makine probleminin özel bir halidir. Literatürde çizelgeleme problemleri için kullanılan sınıflandırma biçimine göre,  $R_m | M_j, \text{ sürece özel kısıtlar} | C_{enb}, \Sigma S_{ij}$  olarak gösterilebilir. Burada  $R_m$  problemin ilişkisiz paralel makina çizelgeleme problemi olduğunu,  $M_j$  makina uygunluk kısıtlarının dikkate alındığını, sürece özel kısıtlar, taşıma sistemi kısıtı gibi işletmenin üretim sürecinden kaynaklanan kısıtların varlığını,  $C_{enb}$  ve  $\Sigma S_{ij}$  problemin son işin tamamlanma zamanını ve farklılık katsayıları toplamının enküçüklenmesi olmak üzere iki amacının olduğunu ifade etmektedir.

Bir çizelgeleme problemi tek amaçlı ya da çok amaçlı olabilmektedir. Literatürde çoğunlukla tek amaçlı problemler üzerinde çalışılmış olmasına karşın; gerçek hayatta çok amaçlı olanlarla daha çok karşılaşmaktadır. Bu sebeple çok amaçlı çizelgelemenin tek amaçlı çizelgelemeye kıyasla daha gerçekçi ve uygulanabilir

olduğu söylenebilir. Ayrıca literatürde paralel makine çizelgeleme problemlerini ele alan çalışmaların önemli bir kısmı özdeş paralel makinaları ele almıştır. Bu çalışmada, ilişkisiz paralel makine problemi üzerine çalışılmıştır.

İlişkisiz paralel makine çizelgeleme problemini ele alan çalışmalar incelendiğinde, Liaw ve ark. (2003) toplam ağırlıklı gecikmeyi en küçük yapacak şekilde bir çizelgeleme oluşturmayı amaçlamışlardır. Bu NPzor problemin çözümü için dal sınır algoritması uygulamışlar ve çözüm aralığı için alt ve üst sınırlar üretmişlerdir. Mokotoff ve Chretienne (2002)  $R_m // C_{enb}$  problemi için kesme algoritmasını uygulamışlar ve elde ettikleri çözümleri yaklaşım algoritmaları ile desteklemişlerdir. Li ve Yang (2008), Chen (2005) ve Rabadi ve ark. (2006) yaptıkları çalışmalarda sezgisel yöntemler kullanmışlardır. Pinedo (2016) ve Peyro ve Ruiz (2011) ise ilişkisiz makinelerin çizelgenmesi üzerine çeşitli araştırmalar yapmışlardır. Rocha ve ark. (2008) ilişkisiz paralel makinelerin hazırlık sürelerini de dikkate alarak önerdikleri tam sayılı programlama modelini Cplex 9.0 ve dal-sınır algoritması ile çözerek elde ettikleri sonuçları karşılaştırmışlardır. Martello ve ark. (1997) ilişkisiz paralel makine çizelgeleme probleminde son işin tamamlanma zamanını enküçüklemek için yaklaşım algoritmaları önermişlerdir.

Vardiya çizelgeleme problemlerine literatürde çok yer verilmemiştir. Vardiya çizelgeleme konusundaki tam sayılı matematiksel modellerden ilki 1954 yılında George Dantzig tarafından, ikincisi ise 1979 yılında Elbridge Keith tarafından geliştirilmiştir (Thompson, 1999). Bechtold ve Jacobs (1990), Aykin (1996), Thompson (1999), vb. vardiya çizelgelemesi yapmak üzere tam sayılı matematiksel modeller geliştirmişlerdir.

Literatürde çizelgeleme problemleri için geliştirilen pek çok farklı çözüm yöntemi ve ele alınan pek çok farklı süreç özelliği mevcuttur. Bu çalışmada, şimdiye kadar yapılmış çoğu çalışmanın aksine vardiya bazında kısıtların olduğu ilişkisiz paralel makine çizelgeleme problemi ele alınmıştır. Erişilen literatür incelendiğinde ilişkisiz paralel makinelerde vardiya bazında çizelgeleme probleminin ele alındığı bir çalışmaya rastlanmamıştır.

Sadece iş sırasını belirleyen klasik çizelgeleme yaklaşımına göre bir problem çözüldüğünde bu sonu-

cun gerçek hayatta uygulanması genellikle mümkün değildir. Çünkü işletmelerin her vardiyaya özel kapasite ve uygunluk kısıtları mevcuttur. Çizelgenin bu kısıtlara uygun hazırlanabilmesi ise işlerin hangi vardiya ya da vardiyalarda üretileceğinin kesin olarak bilinmesini gerektirmektedir. Bu nedenle bu çalışmada vardiya bazında çizelgeleme yapılmıştır. Bu durum literatürde yer alan klasik uygunluk kısıtlarından farklıdır. Literatürdeki kısıtlar genellikle bir işin bir makinaya atanıp atanamayacağı yönündedir. Ancak bu çalışmada ele alındığı haliyle, işlerin ayrıca vardiya bazında da atanmaları gerekmektedir. Bu da problemi daha zor bir yapıya dönüştürmektedir. Bu çalışmada hem vardiya bazında uygunluk kısıtları (her vardiyada her makinanın çalışmaması durumu) hem de vardiya bazında kapasite kısıtları (her vardiyada otomatik taşıma sisteminin ve sonraki süreçlerin kapasiteleri vardır. Dolayısıyla üretim çizelgesi buna göre hazırlanmalıdır.) mevcuttur. Bu haliyle çalışma erişilen literatürdeki benzerlerinden farklılık göstermektedir.

## 2. PROBLEMİN TANIMI

Problemin yaşandığı üretim işletmesinde çizelgeleme, mümkün olduğunca işleri geciktirmemeye özen göstererek planlama sorumlusunun tecrübeleri doğrultusunda sistematik bir yaklaşım kullanılmadan yapılmaktadır. İşletmede sipariş tipi üretim yapılmakta ve 3200 çeşit ürün üretilmektedir. Sık sık yeni siparişler geldiği için planlama periyodu 24 saat olarak belirlenmiştir. Böylece yeni siparişler gelmesi durumunda üretim planındaki değişiklikleri en kısa sürede göz önünde bulunduran, etkili ve verimli bir planlama mümkün olacaktır. Bir planlama periyodunda 30 çeşide kadar ürün üretilebilmektedir.

İşletmede 4 adet paralel kaynak makinesi bulunmaktadır. Bu makinelerin farklı teknik özelliklere sahip olmaları sebebiyle her ürün her makinede üretilememektedir. Ayrıca, her kaynak makinesi aynı ürünü farklı hızlarda üretmektedir.

Kaynak makinelerinde bir işten diğerine geçilirken yapılacak hazırlığın süresi ihmal edilebilir düzeydedir. Ancak birbirine benzeyen işlerin ardışık çizelgelenmesi işlem kalitesini ve yaşanacak üretim problemlerini azaltmaktadır. Bu nedenle firma mümkün olduğunca benzer işlerin arka arkaya çizelgelenmesini istemektedir.

Ele alınan problemde, bir ürünün hatve, yükseklik, derinlik, uzunluk ve marka olmak üzere beş karakteristik özelliği vardır. İki ürünün bu beş özelliği de aynı ise, ard arda üretildiklerinde herhangi bir problem yaşanmamaktadır. Çizelgeleme yapılırken birbiri ardına çizelgelenen ürünlerin mümkün olduğunca çok özelliğinin aynı olması istenmektedir. Ancak mevcut uygulamada ürün çeşitliliğinin çok olması nedeniyle bu her zaman mümkün olamamaktadır. Birbirinden çok farklı ürünlerin arka arkaya çizelgelenmesi, makinelerin verimliliğinin düşmesine sebep olmaktadır. Bir başka deyişle, birbirine benzeyen ürünlerin aynı makinelere atanması, bu mümkün olamıyorsa art arda üretilen ürünlerin olabildiğince az farklılığa sahip olması gerekmektedir. İşletmede, ürünlerin işleme girme sırası sistematik bir yaklaşımla belirlenmediği için dikkate alınması gereken tüm seçenekler üretilemeyip gözden kaçırılmaktadır. Bu durum, fazla fire oluşmasına ve makinelerin üretiminin yavaşlamasına sebep olmaktadır.

Bir diğer önemli konu da, uzman personelin her vardiyada olamaması nedeniyle, bazı makinelerin bazı vardiyalarda çalışmamasıdır. Bu durum, vardiyaya özel çizelgeleme yapılmasını gerektirmektedir.

Kaynak işlemleri tamamlanan paneller, panel taşıma askısına takılarak, otomatik taşıma sistemi aracılığı ile boyahaneye taşınmaktadır. Otomatik taşıma sistemi sabit bir hızla çalışmaktadır ve belli sayıda taşıma askısına sahiptir. Her taşıma askısına bir ya da birden fazla panel takılması mümkündür. Taşıma askılarının uzunluğu standarttır. Bir taşıma askısına, asılacak panellerin uzunlukları toplamı taşıma askısının uzunluğunu geçmemelidir. Bir vardiyada ortalama 900 adet panel taşıma askısının kullanılması mümkündür. İşletme, kaynağı tamamlanmış panellerin ara stok yapılmaksızın boyahane bölümüne taşınmasını istediğinden, çizelgeleme yapılırken taşıma sisteminin kapasitesi göz önünde bulundurulmalıdır. Bir askıya asılan ortalama panel sayısı 3'ü geçtiğinde işletmenin paketleme bölümünün kapasitesi aşılmaktadır. Bu nedenle otomatik taşıma sistemi yüklenirken bir taşıma askısına düşen ortalama panel sayısının 3'ü geçmemesi gerekmektedir.

### 3. BÜTÜNLEŞİK MATEMATİKSEL MODEL

Bu çalışmada, çok amaçlı yapıda olan ve sürece özel kısıtlar içeren ilişkisiz paralel makine çizelgeleme problemi için bir hedef programlama modeli geliştirilmiştir. Amaç, son işin tamamlanma zamanının ve farklılık katsayıları toplamının, hedef değerlerinden pozitif sapmalarının toplamını enküçükmektir. Önerilen bütünleşik matematiksel model ile büyük boyutlu problemlerin çözülebilmesi mümkün görünmemektedir. Bu nedenle büyük boyutlu problemlerin çözümü için iki aşamalı bir çözüm yaklaşımı önerilmiştir. Yöntemin ilk aşamasında işler vardiyalara bir matematiksel model yardımı ile atanmaktadır. İkinci aşamada ise işlerin hangi makinelerin hangi sırasına atanacaklarını belirlemek üzere, her vardiya için bir matematiksel model çözülmüştür. İki aşamalı yaklaşım, bütünleşik modelin çözemediği büyüklükteki problemlerin çözülebilmesine imkân sağlayabilecek ve böylece firmanın beklentilerini karşılayan iş çizelgeleri kısa sürede oluşturulabilecektir.

#### 3.1 Bütünleşik Matematiksel Model

Bu bölümde, geliştirilen hedef programlama modeli ve çözümünü elde edilen sonuçlar ayrıntılı bir şekilde ele alınmıştır. Hedef programlama yaklaşımında, hedeflerden istenmeyen yöndeki sapmalar enküçüklenmeye çalışılır. Önerilen bütünleşik modelin varsayımları, parametreleri, karar değişkenleri, kısıtları, hedef kısıtları ve amaç fonksiyonu aşağıda verilmiştir.

*Varsayımlar:*

- İşçi ve ara mamul kısıtları nedeniyle herhangi bir vardiyada çalışmayan makineler bulunabilir
- Makineler hız bakımından birbirinden farklıdır.
- Tüm makineler ve işler sıfır zamanında hazırdır.
- Bir makine aynı anda tek bir işi işleyebilir.
- Bir işe başlandıktan sonra o iş bölünmeksizin veya başka bir makineye aktarılmaksızın işlenmelidir.

*Kümeler:*

$N = \{1, 2, \dots, n\}$  iş kümesi

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

$V = \{1, 2, \dots, o\}$  vardiya kümesi

*İndisler:*

$i$  ve  $j \in N$  belirli bir işi göstermek için kullanılan indislerdir.

$k \in N$  iş sırasını göstermek için kullanılan indistir.  
 $l \in M$  bir makineyi göstermek için kullanılan indistir.  
 $v \in V$  bir vardiyayı göstermek için kullanılan indistir.

*Parametreler:*

$n$ : iş sayısı

$m$ : makine sayısı

$o$ : vardiya sayısı

$r_{jl}$ :  $j$  işin  $l$  makinadaki birim işlem süresi

$a_j$ :  $j$  işin sipariş miktarı

$p_{jl}$ :  $j$  işin  $l$  makinadaki işlem süresi ( $p_{jl} = r_{jl} * a_j$ )

$s_{ij}$ :  $i$  iş ile  $j$  işin benzemezlik düzeyini gösteren farklılık katsayısı

$e$ : bir vardiyanın çalışma süresi

$y_{lv}$ :  $l$  makine  $v$  vardiyada üretime elverişli ise 1, diğer durumlarda 0

$b_j$ : çizelgelenen işlerin, hangi makinelerde üretilebildiğini göstermektedir. ( $j$  iş  $l$  makinede yapılabilirse 1, diğer durumlarda 0).

$g$ : çok büyük pozitif bir sayı

$h$ : planlama periyodu ( $C_{enb}$  için hedef değer)

$u_j$ :  $j$  panelin uzunluğu

$t_j$ : bir taşıma askısına  $j$  panelden kaç adet asılacağı

*Karar Değişkenleri:*

$w_{ij}$ :  $i$  işten sonra  $j$  işin üretilmesi halini gösteren karar değişkeni ( $i$  işten sonra  $j$  iş üretiliyorsa 1, diğer durumlarda 0)

$x_{jlk}$ :  $j$  işi  $l$  makinanın  $k$  sırasına atadıysa 1, diğer durumlarda 0

$C_j$ :  $j$  işin tamamlanma zamanı

$C_{enb}$ :  $m$  makine içinde işlem gören son işin tamamlanma zamanı

$q_{jv}$ :  $j$  işin hangi vardiyaya atandığını gösteren karar değişkeni ( $j$  iş  $v$  vardiyaya atandıysa 1, diğer durumlarda 0)

$f_j$ :  $j$  işin tamamlandığı vardiyayı gösteren karar değişkeni

$S_1^+$ : hedeflenen farklılık düzeyinden pozitif yönlü sapma

$S_1^-$ : hedeflenen farklılık düzeyinden negatif yönlü sapma

$S_2^+$ : Son işin tamamlanma zamanı için hedeflenen değerden pozitif yönlü sapma

$S_2^-$ : Son işin tamamlanma zamanı için hedeflenen değerden negatif yönlü sapma

*Kısıtlar:*

$$C_j + g^*(1 - x_{jlk}) \geq p_{jl}^* a_j \quad \forall j, k=1, l \quad (1)$$

$$C_j + g^*(2 - x_{ilk-l} - x_{jlk}) \geq C_i + p_{jl}^* a_j \quad \forall i \neq j, k > 1, l \quad (2)$$

$$\sum_j x_{jlk} \leq 1 \quad \forall k, l \quad (3)$$

$$\sum_k \sum_l x_{jlk} = 1 \quad \forall j \quad (4)$$

$$x_{jlk} \leq b_{jl} \quad \forall j, k, l \quad (5)$$

$$f_j \geq (C_j/e) \quad \forall j \quad (6)$$

$$f_j \leq (C_j/e) + 1 \quad \forall j \quad (7)$$

$$\sum_j x_{jlk} - \sum_i x_{ilk-l} \geq 0 \quad \forall k > 1, l \quad (8)$$

$$C_{enk} \geq C_j \quad \forall j \quad (9)$$

$$\sum_v q_{jv} \geq f_j - g^*(1 - x_{jlk}) \quad \forall j, k=1, l \quad (10)$$

$$\sum_v q_{jv} \leq f_j + g^*(1 - x_{jlk}) \quad \forall j, k=1, l \quad (11)$$

$$f_j + g^*(1 - x_{jlk}) \geq v^* q_{jv} \quad \forall j, k=1, l, v \quad (12)$$

$$\sum_v q_{jv} \geq f_j - f_i + 1 - g^*(2 - x_{ilk-l} - x_{jlk}) \quad \forall i \neq j, k > 1, l \quad (13)$$

$$\sum_v q_{jv} \leq f_j - f_i + 1 + g^*(2 - x_{ilk-l} - x_{jlk}) \quad \forall i \neq j, k > 1, l \quad (14)$$

$$f_j + g^*(2 - x_{ilk-l} - x_{jlk}) \geq v^* q_{jv} \quad \forall i \neq j, k > 1, l, v \quad (15)$$

$$f_j - g^*(2 - x_{ilk-l} - x_{jlk}) \leq v^* q_{jv} + g^*(1 - q_{jv}) \quad \forall i \neq j, k > 1, l, v \quad (16)$$

$$e \geq q_{jv} \quad \forall j, v \quad (17)$$

$$y_{lv} + 1 \geq q_{jv} + x_{jlk} \quad \forall j, k, l, v \quad (18)$$

$$w_{ij} + g^*(2 - x_{ilk-l} - x_{jlk}) \geq 1 \quad \forall i, j, l, k > 1 \quad (19)$$

$$\sum_i \sum_j w_{ij}^* s_{ij} - S_1^+ + S_1^- = 0 \quad (20)$$

$$C_{enb} - S_2^+ + S_2^- = h \quad (21)$$

$$\sum_j (a_j * u_j * q_{jv} / 100) / (\sum_j (a_j * u_j) * \sum_j (t_j * a_j) / \sum_j a_j) \leq 3 \quad \forall v \quad (22)$$

$$\sum_j (q_{jv} * a_j) / t \leq 900 \quad \forall v \quad (23)$$

$$x_{jlk}, w_{ij}, q_j \in \{0, 1\} \quad (24)$$



$$f_j \geq 0 \text{ ve tamsayı} \quad \forall j \quad (25)$$

$$C_j, S_1^+, S_1^-, S_2^+, S_2^-, C_{enb} \geq 0 \quad (26)$$

$$enkz = S_1^+ + S_2^+ \quad (27)$$

(1) numaralı kısıt, her makinada ilk sıraya atanan işlerin tamamlanma zamanlarını hesaplar. (2) numaralı kısıt, ikinci ya da daha büyük bir sıraya atanan işlerin tamamlanma zamanlarını hesaplar. (3) numaralı kısıt herhangi bir makinede herhangi bir sıraya en fazla bir iş atanmasını sağlar. (4) numaralı kısıt her işin herhangi bir makinenin herhangi bir sırasına atanmasını garanti eder. (5) numaralı kısıt, teknik özellikler sebebiyle ürünlerin üretilmedikleri makinalara atanmasını engeller. (6) ve (7) numaralı kısıtlar bir işin tamamlandığı vardiyayı belirler. (8) numaralı kısıt işlerin makinelerin sıra atlamadan atanmasını sağlar. (9) numaralı kısıt son işin tamamlanma zamanının ( $C_{enb}$ ) hesaplanmasını sağlar. (10), (11) ve (12) numaralı kısıtlar herhangi bir makinede ilk sıraya atanmış işlerin başlangıç ve bitiş vardiyaları arasında işlem görmesini sağlar. (13), (14), (15) ve (16) numaralı kısıtlar ikinci ya da daha büyük sıraya atanmış olan işlerin başlangıç ve bitiş vardiyaları arasında hangi vardiyalarda işlem görmeye devam ettiğinin belirlenmesini sağlar. (17) numaralı kısıt resmi bayram ve tatil gibi durumlarda çalışma yapılmayacak vardiyalara iş atmasını engeller. (18) numaralı kısıt, işlerin çalışmaya elverişli olmayan makinelere atanmasını engeller. (19) numaralı kısıt  $i$  işinden sonra  $j$  işinin üretileceği bilgisini taşır. (20) numaralı kısıt mümkün olduğunca benzer işlerin ard arda üretilmesini hedeflemektedir. Kısıttaki  $S_1^+$  değişkeni hedeflenen değerden pozitif yönlü sapmayı temsil eder. Pozitif yöndeki bu sapma istenmeyen bir durum olduğu için bu değer amaç fonksiyonunda enküçüklenmek istenir. (21) numaralı kısıt ile son işin tamamlanma zamanının planlama döneminin toplam süresine eşit olması hedeflenir. Kısıtta yer alan,  $S_2^+$  değişkeni hedeflenen değerden pozitif yönlü sapmayı temsil etmektedir. Pozitif yöndeki bu sapma istenmeyen bir durum olduğu için bu değer amaç fonksiyonunda enküçüklenmesi istenir. (22) ve (23) nolu kısıtlar işletmenin otomatik taşıma sisteminin kapasitesi ile ilgili kısıtlardır. (22) nolu kısıt, her vardiyada askılara takılan ortalama panel sayısının 3'ü geçmemesini sağlar.

(23) numaralı kısıt ise her vardiyada en fazla 900 adet askının kullanılmasını garanti eder. (24), (25) ve (26) numaralı kısıtlar matematiksel modelde yer alan karar değişkenlerine ait işaret kısıtlarıdır. (27) matematiksel modelin amaç fonksiyonunu belirtir. Amaç fonksiyonu hedef değerlerinden pozitif yönlü sapmaların toplamının enküçüklenmesidir.

Bütünleşik modelin kısıt sayısı;  $4n^3m+n^2m(o+1)+nm(1-o)+5n+no+2o$ 'dur.

### 3.2 İki Aşamalı Çözüm Yaklaşımı

Büyük boyutlu problemlerin çözümü için iki aşamalı bir çözüm yaklaşımı önerilmiştir. Birinci aşamada işlerin atanacağı makinaların belirlenmesi hedeflenmiştir (M1). İkinci aşamada da, birinci aşamada belirlenen, makinalara atanmış işlerin, sıralarına ve vardiyalarına karar vermek hedeflenmiştir (M2).

#### M1:

*Karar Değişkenleri:*

$xx_{jl}$ :  $j$  iş  $l$  makineye atandıysa 1 diğer durumlarda 0

$TP_l$ :  $l$  makinadaki toplam üretim süresi

$TP_{enb}$ : en büyük toplam üretim süresi

$Senb_l$ :  $l$ . makinenin en büyük farklılık katsayısı

$$TP = \sum_j p_{jl} * xx_{jl} \quad \forall l \quad (1.1)$$

$$\sum_l xx_{jl} = 1 \quad \forall j \quad (1.2)$$

$$xx_{jl} \leq b_{jl} \quad \forall j, l \quad (1.3)$$

$$TP_{enb} \geq TP_l \quad \forall l \quad (1.4)$$

$$Senb_l \geq s_{ij} - g * (2 - xx_{il} - xx_{jl}) \quad \forall i \neq j, l \quad (1.5)$$

$$enkz_l = TP_{enb} + \sum_l Senb_l \quad (1.6)$$

M1'de (1.1) numaralı kısıt,  $l$ . makineye atanan işlerin toplam üretim süresini verir. (1.2) numaralı kısıt, her işin

bir makinaya atanmasını sağlar. (1.3) numaralı kısıt işlemin üretilmedikleri makinalara atanmasını engeller. (1.4) numaralı kısıt son işin tamamlanma zamanını hesaplar. (1.5) numaralı kısıt her makinanın en büyük farklılık katsayısını hesaplamaktadır. Pozitif yöndeki bu sapma istenmeyen bir durum olduğu için bu değer amaç fonksiyonunda enküçüklenmek istenir. (1.6) toplam üretim süresiyle, makinalara atanan işlerin öncelik değerinin toplamının enküçüklenmesi istenir. M1'in kısıt sayısı :  $n^2m+nm+n+m$ 'dir.

**M2:**

*Parametreler:*

$xxx_{jl}$ :  $j$  iş  $l$  makinaya atanmışsa 1 diğer durumlarda 0.

$d_l$ :  $l$  makinaya atanan iş sayısı  $\sum_j xxx_{jl} = d_l$

*Karar Değişkenleri:*

$x_{jk}$ :  $j$  işi  $k$ . sırasına atadıysa 1, diğer durumlarda 0.

$$C_j + g^*(1 - x_{jk}) \geq p_{jl} \quad \forall j, k=1, k \leq d_l, xxx_{jl}=1, l \quad (2.1)$$

$$C_j + g^*(2 - x_{ik-l} - x_{jk}) \geq C_i + p_{jl} \quad \forall i \neq j, k > 1, k \leq d_l, xxx_{jl}=1, xxx_{il}=1, l, \quad (2.2)$$

$$\sum_{k|k \leq d_l} x_{jk} = 1 \quad \forall j, l, xxx_{jl} \quad (2.3)$$

$$\sum_j |xxx_{jl}=1} x_{jk} = 1 \quad \forall k \leq d_l, l \quad (2.4)$$

$$f_j \geq (C_j / \text{sure}) \quad \forall j \quad (2.5)$$

$$f_j \leq (C_j / \text{sure}) + 1 \quad \forall j \quad (2.6)$$

$$C_{\text{emb}} \geq C_j \quad \forall j \quad (2.7)$$

$$\sum_v q_{jv} \geq f_j - g^*(1 - x_{jk}) \quad \forall j, k=1, xxx_{jl}=1, l \quad (2.8)$$

$$\sum_v q_{jv} \leq f_j + g^*(1 - x_{jk}) \quad \forall j, k=1, xxx_{jl}=1, l \quad (2.9)$$

$$f_j + g^*(1 - x_{jk}) \geq v * q_{jv} \quad \forall j, k=1, xxx_{jl}=1, l, v \quad (2.10)$$

$$\sum_v q_{jv} \geq f_j - f_i + 1 - g^*(2 - x_{ik-l} - x_{jk}) \quad \forall i \neq j, k > 1, xxx_{il}=1, xxx_{jl}=1, k \leq d_l, l \quad (2.11)$$

$$\sum_v q_{jv} \leq f_j - f_i + 1 + g^*(2 - x_{ik-l} - x_{jk}) \quad \forall i \neq j, k > 1, xxx_{il}=1, xxx_{jl}=1, k \leq d_l, l \quad (2.12)$$

$$f_j + g^*(2 - x_{ik-l} - x_{jk}) \geq v * q_{jv} \quad \forall i \neq j, k > 1, xxx_{il}=1, xxx_{jl}=1, k \leq d_l, l \quad (2.13)$$

$$f_j - g^*(2 - x_{ik-l} - x_{jk}) \leq v * q_{jv} + \text{sayı} * (1 - q_{jv}) \quad \forall i \neq j, k > 1, xxx_{il}=1, xxx_{jl}=1, k \leq d_l, l \quad (2.14)$$

$$e \geq q_{jv} \quad \forall j, v \quad (2.15)$$

$$y_{lv} + 1 \geq q_{jv} + x_{jk} \quad \forall j, k, l, v, xxx_{jl}=1, k \leq d_l \quad (2.16)$$

$$w_{ij} + g^*(2 - x_{ik-l} - x_{jk}) \geq 1 \quad \forall i, j, k > 1, l, xxx_{il}=1, xxx_{jl}=1, k \leq d_l \quad (2.17)$$

$$\sum_i \sum_j w_{ij} * s_{ij} - S_1^+ + S_1^- = 0 \quad (2.18)$$

$$C_{\text{emb}} - S_2^+ + S_2^- = h \quad (2.19)$$



$$\sum_j (a_j * u_j * q_{jv} / 100) / (\sum_j (a_j * u_j) * \sum_j (t_j * a_j) / \sum_j a_j) \leq 3 \quad \forall v \quad (2.20)$$

$$\sum_j (q_{jv} * a_j) / t_j \leq 900 \quad \forall v \quad (2.21)$$

$$x_{jk} \in \{0,1\} \quad \forall j, k \quad (2.22)$$

$$w_{ij} \in \{0,1\} \quad \forall i, j \quad (2.23)$$

$$q_{jv} \in \{0,1\} \quad \forall j, v \quad (2.24)$$

$$f_j \geq 0 \text{ ve tamsayı} \quad \forall j \quad (2.25)$$

$$C_j \geq 0 \quad \forall j \quad (2.26)$$

$$S_1^+, S_1^-, S_2^+, S_2^-, C_{emb} \geq 0 \quad (2.27)$$

$$enkz_2 = S_1^+ + S_2^+ \quad (2.28)$$

M2'de (2.1) numaralı kısıt,  $j$ . iş  $l$ . makinaya atanmış ise  $j$  işine ait tamamlanma zamanının, işlem süresi toplamından büyük ya da eşit olmasını sağlar. (2.2) numaralı kısıt,  $j$  işi bir makinada ikinci ya da daha büyük bir sıraya atanırsa, tamamlanma zamanının bir önceki işin tamamlanma zamanı ve ilgili işin işlem süresi toplamından büyük ya da eşit olmasını sağlar. (2.3) numaralı kısıt her işin atandığı makinada herhangi bir sıraya atanmış olmasını garanti eder. (2.4) numaralı kısıt  $l$  makinaya herhangi bir sıraya en fazla bir iş atanmasını sağlar. (2.5) ve (2.6) numaralı kısıtlar bir işin tamamlandığı vardiyayı temsil eden karar değişkeni değerinin, o işin tamamlanma zamanının vardiya çalışma süresine oranlanmasıyla ortaya çıkan değerden büyük en küçük tamsayı olmasını sağlar. (2.7) numaralı kısıt  $C_{emb}$  değişkeni değerinin tüm işler arasında tamamlanma zamanı en büyük olan değerden büyük ya da o değere eşit olmasını sağlar. (2.8) ve (2.9) numaralı kısıtlar çizelgenin ilk sırasında yer alan bir işin işlem gördüğü toplam vardiya sayısının o işin tamamlandığı vardiya değeri kadar olmasını sağlar. (2.10) numaralı kısıt ise ilk sıraya atanmış olan işin tamamlandığı vardiyadan sonraki vardiyalara atanmasını engelleyerek o işin başlangıç ve bitiş vardiyaları arasında hangi vardiyalarda işlem görmeye devam ettiğinin belirlenmesini sağlar. (2.8), (2.9) ve (2.10) numaralı kısıtlar herhangi bir makinada ilk sıraya atanacak olan

herhangi bir işin başlangıç ve bitiş vardiyaları arasında işlem görmeye devam ettiğinin belirlenmesini sağlar. (2.11), (2.12), (2.13) ve (2.14) numaralı kısıtlar ikinci ya da daha büyük sıraya atanmış olan işlerin başlangıç ve bitiş vardiyaları arasında hangi vardiyalarda işlem görmeye devam ettiğinin belirlenmesini sağlar. (2.15) numaralı kısıt resmi bayram ve tatil gibi durumlarda çalışma yapılmayacak vardiyalara iş atamasının yapılmasını engeller. (2.16) numaralı makina ve sürece özel birtakım sebeplerle bazı makinaların bazı vardiyalarda çalışmaya elverişli olmadığı durumlarda, işlerin bu makinalara atamasını engeller. (2.17) numaralı kısıt ayar  $i$  den sonra  $j$  nin üretilmesiyle ayar süresi arasında ilişki kurmayı sağlar. (2.18) numaralı kısıt her bir işin hedeflenen farklılık katsayısından sapmalarını hesaplamaktadır. Kısıttaki  $S_1^+$  değişkeni hedeflenen değerinin pozitif yönlü sapmayı temsil eder. Pozitif yöndeki bu sapma istenmeyen bir durum olduğu için bu değer amaç fonksiyonunda enküçüklenmek istenir. (2.19) numaralı kısıt ile son işin tamamlanma zamanının planlama döneminin toplam süresine eşit olması hedeflenir. Kısıtta yer alan,  $S_2^+$  değişkeni hedeflenen değerden pozitif yönlü sapmayı temsil etmektedir. Pozitif yöndeki bu sapma istenmeyen bir durum olduğu için bu değer amaç fonksiyonunda enküçüklenmek istenir. (2.20) ve (2.21) nolu kısıtlar işletmenin otomatik taşıma sisteminin kapasitesi ile

ilgili kısıtlardır. (2.20) nolu kısıt, her vardiyada askılara takılan ortalama panel sayısının 3'ü geçmemesini sağlar. (2.21) numaralı kısıt ise her vardiyada en fazla 900 adet askının kullanılmasını garanti eder. (2.22)-(2.27) numaralı kısıtlar matematiksel modelde yer alan karar değişkenlerine ait işaret kısıtlarıdır. Amaç fonksiyonu (2.28) her iş için hedeflenen sıfır öncelik değerinden olan pozitif yönlü sapmalar toplamının ve son işin tamamlanma zamanı için hedeflenen değerden pozitif yönlü sapmanın enküçüklenmesidir. M2'in kısıt sayısı:  $n(6n^2m+5m+4) + v(n^2m+nm+n+2)$ 'dir.

#### 4. DENEYSEL SONUÇLAR

Geliştirilen bütünleşik matematiksel modelin ve iki aşamalı çözüm yaklaşımının başarısını gösterebilmek için öncelikle küçük boyutlu bir problem türetilmiştir. Daha sonra bir gerçek hayat probleminin çözümü araştırılmıştır. Tüm testler Intel (R) Core (TM) i7-4790 CPU@3.66 GHz işlemcisi, 16 GB belleği olan bir bilgisayarda yapılmıştır. Matematiksel modeller GAMS 24.1.3 ile kodlanmış ve çözücü olarak Cplex kullanılmıştır.

##### 4.1 Küçük Örnek Problem

3 adet kaynak makinesi vardır. İşletme tek çeşit hatve, tek çeşit marka, 2 çeşit derinlik, 3 çeşit yükseklik ve 4 çeşit uzunlukta ürün üretebilmektedir. Ürünlerin özellikleri Tablo 1'de verilmiştir.

**Tablo 1.** Ana Ürün Tablosu

Ürün Numarası	Hatve	Marka	Derinlik	Yükseklik	Uzunluk
1	1	1	1	3	1
2	1	1	1	3	3
3	1	1	1	3	4
4	1	1	2	2	1
5	1	1	2	2	3
6	1	1	1	3	1
7	1	1	1	3	3
8	1	1	2	1	4
9	1	1	2	1	3
10	1	1	1	1	2

Tablo 1'de her farklı özellik bir rakamla ifade edilmiştir. Sözelimi birinci satırda 8. ürünün, 1. hatvede, 1. markada, 2. derinlikte, 1. yükseklik ve 4. uzunlukta üretilmesi gerektiği görülmektedir. Derinlik, yükseklik ve uzunluk değişmesi durumunda farklılık katsayısını hesaplamakta kullanılan değerler sırasıyla Tablo 2, Tablo 3 ve Tablo 4'de verilmiştir. Yükseklik, derinlik ve uzunluk değişimleri zorluk derecelerini temsil edecek şekilde farklı ölçeklendirilmiştir. Örnek problemde, yükseklik dönüşümü en zor dönüşüm olduğu için [500,600] aralığında, derinlik dönüşümü ikinci zor değişim olduğundan [200-300] aralığında ve uzunluk dönüşümü en kolay dönüşüm olduğundan [1-10] aralığında değer alacak şekilde tasarlanmıştır. Ürünlerin farklılık katsayıları ( $s_{ij}$ ), ilgili dönüşüm değerlerinin toplanmasıyla elde edilmiş ve Tablo 5'de verilmiştir.

**Tablo 2.** Derinlik Dönüşümü

Derinlik	1	2
1	-	250
2	250	-

**Tablo 3.** Yükseklik Dönüşümü

Yükseklik	1	2	3
1	-	550	600
2	550	-	500
3	600	500	-

**Tablo 4.** Uzunluk Dönüşümü

Uzunluk	1	2	3	4
1	-	3,5	4	8,5
2	3,5	-	1	5,5
3	4	1	-	5
4	8,5	5,5	5	-

Tablo 5'ten de görüldüğü gibi 3. üründen sonra 4. ürünün üretilmesi halinde derinlik 1. çeşitten 2. çeşide geçecek ve 250 değeri atanacaktır. Aynı zamanda yükseklik 3. çeşitten 2. çeşide geçecek ve buna karşılık 500 değeri, uzunluk özelliğinde de 1. çeşitten 3. çeşide geçilecek ve 4 değeri atanacaktır. Toplamda 3. üründen sonra 4. ürünün üretilmesi 759 gibi bir değere karşılık gelecektir. Örnek probleme ilişkin diğer parametreler Tablo 6-9'da verilmiştir.

**Tablo 5.** Farklılık Katsayısı

$s_{ij}$	1	2	3	4	5	6	7	8	9	10
1	0	4	8,5	750	754	600	604	258,5	254	3,5
2	4	0	5	754	750	604	600	255	250	1
3	8,5	5	0	759	755	608,5	605	250	255	5,5
4	750	754	759	0	4	800	804	509	504	754
5	754	750	755	4	0	804	800	505	500	751
6	600	604	608,5	800	804	0	4	858,5	854	603,5
7	604	600	605	804	800	4	0	855	850	601
8	258,5	255	250	509	505	858,5	855	0	5	255,5
9	254	250	255	504	500	854	850	5	0	251
10	3,5	1	5,5	754	751	603,5	601	255,5	251	0

**Tablo 6.** Mevcut İşlere Ait İşlem Süresi

$r_{jt}$	1	2	3
1	16,20	9,25	9,26
2	21,73	18,3	18,4
3	32,81	23,08	23,09
4	0	10,1	10,2
5	0	18,3	18,4
6	0	8	9
7	0	18,3	18,4
8	0	23,08	23,09
9	0	18,3	18,3
10	20,81	18,2	18,2

**Tablo 7.** Parametre Değerleri

$u_j$	$t_j$	$a_j$
100	3	480
150	2	400
200	2	250
100	4	180
150	2	350
100	4	100
150	2	140
200	1	250
150	2	170
140	2	200

**Tablo 8.** İş-Makine İlişkisi

$b_{jt}$	1	2	3
1	1	1	1
2	1	1	1
3	1	1	1
4	0	1	1
5	0	1	1
6	0	1	1
7	0	1	1
8	0	1	1
9	0	1	1
10	1	1	1

**Tablo 9.** Makine-Vardiya İlişkisi

$y_{vt}$	1	2	3
1	1	1	1
2	1	1	1
3	1	1	1

Tablo 6'da işlerin makinelerine göre işlem süreleri verilmiştir. Buna göre 10. işin özelliklerinde bir ürünün 1. makinede üretilmesi durumundaki işlem süresi 20,81 birim saniye, 2. ve 3. makinede üretilmesi halinde ise 18,2 birim saniyedir. Tablo 7.'de de çizelgelenecek işlere ait üretim adedi ( $a_j$ ), uzunluğu ( $u_j$ ) ve bir taşıma askısına kaç adet panel asılabileceği ( $t_j$ ) verileri yer almaktadır.

Tablo 8'de işlerin hangi makinalarda üretilebilir olduğu bilgisi bulunmaktadır. Tüm işler tüm makinalarda üretilememektedir. Tablo 9'da ise hangi makinaların hangi vardiyalarda çalışabilir durumda olduğu verilmektedir. Örnek problem için tüm makinaların tüm vardiyalarda çalışabildiği varsayılmıştır.

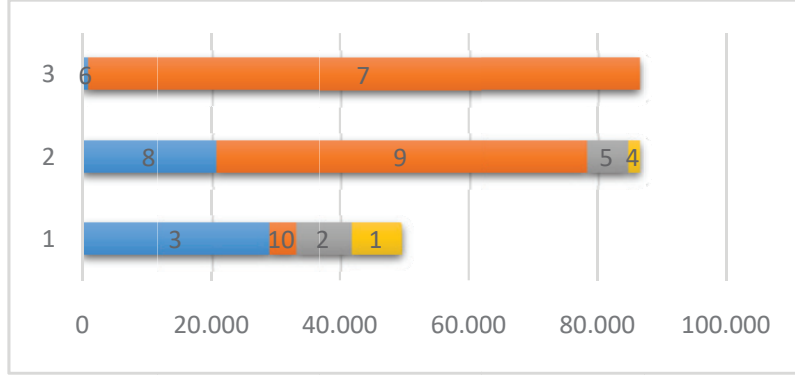
Örnek problem, bütünlük modelile 21 dakikada çözülmüş ve eniyi çözüm elde edilmiştir. Elde edilen çözümün Gantt

Şeması Şekil 1'de, karar değişkeni değerleri ise Tablo 10 ve 11'de verilmiştir.

Tablo 10'da işlerin hangi makineye hangi sırada atandığı yer almaktadır. Örneğin 8 numaralı iş, 2 numaralı makineye 1. sırada atanmıştır. Tablo 11'de işlerin hangi vardiyalara atandığı bilgisi verilmektedir.

Örneğin 3 numaralı iş 1. ve 2. vardiya işlem görmüştür. Amaç fonksiyonu ( $z$ ) 3,852 değerini almıştır. Amaç fonksiyonunda yer alan hedefler incelendiğinde hedeflenen sıfır farklılık düzeyinden pozitif sapma ( $S_1^+$ ) 523,500 değerini almaktadır.

Daha sonra örnek problem geliştirilen iki aşamalı çözüm yaklaşımıyla çözülmüştür. Elde edilen çözümün Gantt Şeması Şekil 2'de, karar değişkenlerinin değerleri ise Tablo 12 ve 13'de verilmiştir.



Şekil 1. Bütünleşik Model ile Elde Edilen Çözümün Gantt Şeması

Tablo 10. Sıra / Makine İlişkisi

Sıra/Makine	Makine 1	Makine 2	Makine 3
1	3	8	6
2	10	9	7
3	2	5	
4	1	4	

Tablo 11. İş / Vardiya İlişkisi

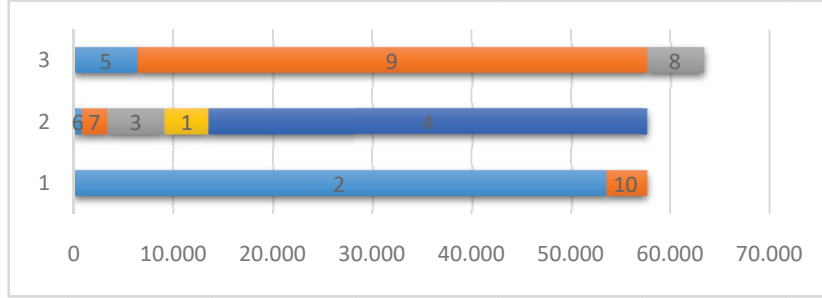
$q_{jv}$	1	2	3
1		1	
2		1	
3	1	1	
4			1
5			1
6	1		
7	1	1	1
8	1		
9		1	1
10		1	

Tablo 12. Sıra / Makine İlişkisi

Sıra/Makine	Makine 1	Makine 2	Makine 3
1	2	6	5
2	10	7	9
3		3	8
4		1	
5		4	

Tablo 13 . İş / Vardiya İlişkisi

$q_{jv}$	1	2	3
1	1		
2	1	1	
3	1		
4	1	1	
5	1		
6	1		
7	1		
8			1
9	1	1	
10		1	



Şekil 2. İki Aşamalı Çözüm Yaklaşımıyla Elde Edilen Çözümün Gantt Şeması

Tablo 12’de işlerin hangi makineye hangi sırada atandığı bilgileri yer almaktadır. Örneğin 6 numaralı iş, 2 numaralı makineye 1. sırada atanmıştır. İşlerin atandığı vardiyalar Tablo 13’de verilmiştir. Buna göre 1. iş 1. vardiyaya atanmıştır, 4. iş ise hem 1. vardiya hem de 2. vardiyalarda işlem görmüştür.

Örnek problemin amaç fonksiyonu ( $z$ ) 1873,500 değerini almıştır. Amaç fonksiyonunda yer alan hedefler incelendiğinde hedeflenen sıfır farklılık düzeyinden pozitif sapmalar toplamı da ( $S_1^+$ ) 1873,5 değerini almaktadır. Bütünleşik model ile  $z = 3,852$  bulunmuş ve problemin çözümü 21 dakika sürmüştür. İki aşamalı model ile ise  $z = 1873,5$  bulunmuş ve çözümü 0,109 dakikada elde edilmiştir. Bütünleşik model amaç fonksiyonunun aldığı değer açısından daha başarılı olurken, iki aşamalı çözüm yaklaşımı ise çözüm süresi açısından avantaj sağlamıştır.

#### 4.2 Gerçek Hayat Problemi

Gerçek hayat probleminde, 4 adet ilişkisiz paralel kaynak makinesi ve bu makinelerde üretililecek 34 iş ele alınmıştır. Bu işlere ait farklılık katsayıları ( $s_{ij}$ ), işlerin marka, yükseklik,

derinlik, hatve ve uzunluk özellikleri dikkate alınarak Excel’de hesaplanmıştır.

Gerçek hayat problemi bütünleşik model ile çözülememiştir.

Önerilen iki aşamalı yaklaşım ile son işin tamamlanma zamanı ( $C_{enb}$ ) 66.856,52 saniye ve hedeflenen sıfır farklılık düzeyinden pozitif sapma ( $S_1^+$ ) 4459,500 değerini almaktadır.

Mevcut durumda son işin tamamlanma zamanı ( $C_{enb}$ ) 68.528,68 saniye ve hedeflenen sıfır farklılık düzeyinden pozitif sapma da ( $S_1^+$ ) 1921 değerini almaktadır.

Son işin tamamlanma zamanını temsil eden  $C_{enb}$  değeri işletme tarafından hazırlanan çizelgede 68.528,68 saniye değerini alırken önerilen modelle elde edilen çizelgede 66.856,52 saniye değerini almaktadır. Sonuçlar karşılaştırıldığında önerilen modelle elde edilen çizelge ile son işin tamamlanma zamanı için %2,44’lük bir iyileştirme sağlanmıştır. Ayrıca her iki hedef değerine de ulaşılmıştır.

## 5. SONUÇ VE ÖNERİLER

Bu çalışmada  $n$  iş  $m$  makinenin olduğu ilişkisiz paralel makine çizelgeleme problemi ele alınmıştır. Çalışmada süreç özelliklerine ve vardiya çizelgelemesine yer verilmiştir. Gerçek hayatta çizelgeleme yapılırken vardiyalara ait bilgilerin dâhil edilmesi kaçınılmazdır. Önerilen model bunu mümkün kılmaktadır. Ele alınan problemin çözümü için bir hedef programlama modeli geliştirilmiştir. Geliştirilen hedef programlama modeli ile işletmeden alınan verilere dayalı bir örnek problemin çözümüne ulaşılabilirken, büyük boyutlu gerçek problem için ise çözüm elde edilememiştir. Büyük boyutlu problemlerin çözümü için iki aşamalı bir çözüm yaklaşımı geliştirilmiştir. Gerçek problem iki aşamalı çözüm yaklaşımı ile çözülmüş ve işletmenin çizelgesine kıyasla %2,44'lük bir iyileşme sağlanmıştır.

## KAYNAKÇA

1. **Ak, B.** 2012. "Seçilmiş Bir Endüstriyel Tesiste Üretim Çizelgeleme Problemi ve Genetik Algoritma Yöntemi İle Optimizasyon," Yüksek lisans tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, s. 140.
2. **Arnaou, J. P., Rabadi, G., Musa, R.** 2009. "A Twostage Ant Colonyoptimizationalgorithmto Minimize Themakespan on Unrelatedparallelmachineswithsequence-Dependentsetup times," Springer US, pp. 693-701.
3. **Aykin T.** 1996. "Optimal Shift Scheduling with Multiple Break Windows," Management Science, vol. 42, no. 4, pp. 475-627.
4. **Bechtold, S. E. ve Jacobs, L. W.** 1990. "Implicit Modeling of Flexible Break Assignments in Optimal Shift Scheduling," Management Science, vol. 36, no. 11, pp. 1293-1416.
5. **Chen, J. F.** 2005. "Unrelatedparallelmachinescheduling-withsecondaryresourceconstraints," The International Journal of Advanced ManufacturingTechnology, vol. 26, Issue 3, pp. 285-292.
6. **Eren, T. ve Güner, E.** 2002. "Tek ve Paralel Makinalı Problemlerde Çok Ölçütlü Çizelgeleme Problemleri için Bir Literatür Taraması," Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, cilt 17, sayı 4, s. 37-69.
7. **Küçük, B.** 2010. "Özdeş Paralel Makineli Bir Üretim Sisteminin Karınca Koloni Algoritması ile Çizelgelemesi," Doktora tezi, T. C. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, s. 97.
8. **Liaw, C. F., Lin, Y. K., Chen, C. Y., Chen, M.** 2003. "Schedulingunrelatedparallelmachines to Minimize Total Weighted tardiness," Computers & Operations Research, vol. 30, pp. 1777-1789.
9. **Martello, S., Soumis, F., Toth, P.** 1997. "Exact and approximation algorithms for makespan minimization on Unrelated parallel machines," Discrete Applied Mathematics, vol. 75, pp. 169-188.
10. **Mokotoff, E., Chretienne, P.** 2002. "A cutting plane algorithm for the unrelated parallel machines scheduling Problem," European Journal of Operational Research, vol. 141, Issue 3, pp. 515-525.
11. **Özkan, S. E.** 2009. "Farklı Kapasiteli Paralel Makinelerin Dinamik Çizelgenmesi İçin Sezgisel Bir Algoritma ve Uygulaması," Yüksek lisans tezi, Çukurova Üniversitesi Fen Bilimleri Enstitüsü, s. 75.
12. **Peyro L. F., Ruiz R.** 2011. "Size-reduction heuristics for the unrelated parallel machines scheduling Problem," Computers & Operations Research, vol. 38, Issue 1, January 2011, p. 301-309.
13. **Pinedo M.L.** 2016. "Scheduling: Theory, Algorithms, And systems," Springer, Fifth Edition, New York, U.S.A.
14. **Rabadi G., Moraga R. J., Al-Salem, A.** 2006. "Heuristics for the Unrelated Parallel Machine Scheduling Problem with Setup Times," vol. 17, Issue 1, pp. 85-97.
15. **Rocha P. L., Ravetti, M. G., Mateus, G. R.** 2008. "Exact algorithms for a Scheduling Problem With unrelated parallel machines and sequence and machine-Dependent-setup times," Computers & Operations, vol. 35, Issue 4, pp. 1250-1264.
16. **Saraç, T. ve Kaya, S.** 2013 "Plastik Enjeksiyon Makinelerinin Vardiya Bazında Çizelgenmesi Problemi İçin Bir Hedef Programlama Modeli," Endüstri Mühendisliği Dergisi, cilt 24, Sayı 1-2, s. 12-26.
17. **Saraç, T., Sipahioğlu, A.** 2009. "Plastik Enjeksiyon Makinelerinin Çizelgenmesi Problemi," Endüstri Mühendisliği Dergisi, cilt: 20, sayı 2, s. 2-14.
18. **Thompson, G. M.** 1999. "Labor scheduling, Part 4: Controlling work force schedules in Real Time," Cornell Hotel and Restaurant Administration Quarterly, 40(3), 85-86. doi: 10.1016/S0010-8804(99)80041-X.



# EXACT SOLUTION APPROACHES FOR THE DIRECTED BI-OBJECTIVE CHINESE POSTMAN PROBLEM

Ezgi Erođlu<sup>1</sup>, Meral Azizođlu<sup>2\*</sup>

<sup>1</sup>Middle East Technical University, Ankara 06800, Turkey  
e-mail: ezgieroglu@metu.edu.tr

<sup>2</sup>Department of Industrial Engineering  
Middle East Technical University, Ankara 06800, Turkey  
e-mail: ma@metu.edu.tr

Geliş Tarihi:18.05.2018; Kabul Ediliş Tarihi: 17.09.2018

## ABSTRACT

In this study, we consider a directed bi-objective Chinese Postman Problem with two additive objectives (like total cost and total distance) and propose two solution approaches to generate all non-dominated objective vectors. The first approach, namely classical approach, uses the optimal solutions of the mixed integer linear programs and generates the non-dominated objective vectors' set sequentially. The second approach, namely branch and bound algorithm takes its spirit from the optimal solutions of the linear programming relaxations and generates the non-dominated objective vectors' set simultaneously. The results of our extensive computational study show that our approaches are capable of solving large-sized problem instances in reasonable times.

**Keywords:** Bi-objective Programming, Chinese Postman Problem, Mixed Integer Linear Programming, Classical Approach, Branch and Bound Algorithm

## YÖNLÜ İKİ OBJEKTİFLİ ÇİNLI POSTACI PROBLEMİ İÇİN KESİN ÇÖZÜM YAKLAŞIMLARI

### ÖZ

Bu çalışmada iki toplamsal kriterli (toplam maliyet ve toplam mesafe gibi) yönlü Çinli postacı problemi ele alınmış ve tüm bastırılmayan objektif vektörlerini yaratmak için iki çözüm yaklaşımı geliştirilmiştir. Birinci yaklaşım, klasik yaklaşım, karmaşık kesikli doğrusal programların optimal çözümlerini kullanmak ve bastırılmayan objektif vektör setini seri olarak yaratmaktadır. İkinci yaklaşım, dal ve sınır algoritması, doğrusal programlama gevşetmelerinin optimal çözümlerini kullanmak ve bastırılmayan objektif vektör setindeki çözümlerini aynı anda yaratmaktadır. Deneysel çalışmalarımızın sonuçları yaklaşımlarımızın büyük boyutlu problemler için makul sürelerde çözüldüğünü göstermektedir.

**Anahtar Kelimeler:** İki-objektifli programlama, Çinli postacı problemi, karmaşık tam sayılı doğrusal programlama, klasik yaklaşım, dal ve sınır algoritması

\* Corresponding Author

## INTRODUCTION

The arc routing problems entail determining a minimum-cost traversal of a specified subset of arcs on a graph. The earliest study on the arc routing problems is the Königsberg Bridge Problem that is concerned with the existence and determination of a closed walk traversing each of the seven bridges once and exactly once. The bridges of Königsberg are represented as arcs and each arc joins a pair of the four junction points (nodes) that can be considered as islands and shores. The Königsberg Bridge Problem was solved by the Swiss mathematician Leonhard Euler in 1736. The Chinese Postman Problem (CPP) deals with the situations where a closed walk does not exist and determines a minimum length walk covering each arc at least once. The CPP was initially posed by Guan (1962) and stated as follows: Leaving from his post office, a postman needs to visit the households on each block in his route, delivering and collecting letters, and then returns to the post office. He would like to cover this route by traveling the minimum possible distance.

A wide range of applications relevant to transportation, urban planning and manufacturing have been mentioned in the literature.

Some noteworthy transportation and urban planning applications of the CPP addressed in the literature and are of interest for many countries include postal delivery, parcel services, garbage collection, milk delivery, snow clearance, street sweeping services, parking meter coins collection, meter reading, routing of salt trucks and inspection of streets for maintenance and airline scheduling.

Amine and Djellab (2013) show the meaningfulness of the CPP concepts in a wide range of manufacturing applications. Dewill et al. (2011) and Han & Na (1999) discuss the laser or water cutting technology that use water or laser to cut materials and follow Chinese walks for efficiency concerns.

Amine and Djellab (2013) discuss the use of CPP walks in programming automatic arm in the surface painting, like in cars manufacturing, especially in automotive car body painting. The task consists of using an atomizer over a painting surface using a back and forth movement so called CPP walk, above materials.

Manber & Israni (1984) introduce the Cutting Path Determination Problem to solve a flame torch paths problem of material cutting. The problem was modeled as a dynamic rural postman problem, where the graph can be changed each time a piece is cut out. Imahori et al. (2008) and Rodrigues & Ferreira (2012) study the cutting hard materials addressing both packing and cutting path problems. They note that their problem motivates an arc routing modeling as the cutter head requires more time to cut each piece.

The classical CPP assumes that each arc is represented by a single weight and the objective is to determine the tour with the minimum total weight. The bi-objective CPPs assume that each arc is defined by two weights, like cost and distance, distance and priority, cost and time. Many practical implications might require consideration of two weights, thereby two objectives. For example, the total distance of the route should be minimized subject to the constraint that its total cost is below a threshold value. Some other concern might be minimizing the total cost subject to the constraint the total travel distance is below a specified level. Moreover, the decision maker may like to see the solutions over which he/she could make trade-offs among two objectives. For example one may like to see the amount of compromise that should be made on the total distance to reduce the total cost to some specified level.

Despite its practical importance, the literature on the bi-objective CPP is quite scarce. To the best of our knowledge, there is a unique study by Prakash et al. (2009) that is limited to the complete enumeration of all feasible solutions among which the non-dominated ones are selected. Recognizing this important gap in the literature, we study a bi-objective CPP with two additive objectives (total cost and total distance) and propose two solution approaches: a classical approach and a branch and bound algorithm. The classical approach, uses the optimal solutions of the mixed integer linear programs. The branch and bound algorithm generates all non-dominated (efficient) points and benefits from the optimal solutions of the Linear Programming Relaxation (LPR) to define our branching scheme and lower and upper bounds.

The rest of the paper is organized as follows: In Section 2, we review the related literature. Section 3 defines our problem and presents related mathematical models. In Section 4, we first define the efficient solutions together with their properties and then state the classical approach that sequentially generates all non-dominated objective vectors. In Section 5 we first settle the complexity of the problem and discuss the branch and bound algorithm that simultaneously generates all nondominated objective vectors. Section 6 discusses the results of our extensive computational experiment and Section 7 concludes the study.

## 2. BASICS AND THE RELATED RESEARCH

The CPP is an arc routing problem in which a single postman serves a number of streets from a post office. The classical CPP finds the minimum cost tour passing through every arc of graph at least once. This problem is firstly studied by Guan (1962). To find minimum total cost tour, the concept of unicursality is used by Ford and Fulkerson (1962). A directed graph is said to be unicursal (Eulerian) if for each node, the number of entering arcs is equal to the number of leaving arcs. An undirected graph is said to be universal if even number of arcs are incident to each node. To make a graph Eulerian a minimum weight perfect matching is defined and optimal solution to the CPP is found. The problem is solved by a polynomial time algorithm developed by Edmonds and Johnson (1973).

Eiselt et al. (1995) give two mathematical models for the undirected CPP. The first model is based on a perfect matching idea that converts a nonunicursal graph into a unicursal one by using the shortest paths. The second model, on the other hand, is based on Edmonds' blossom inequalities (see Edmonds, 1963) and it depends on the density of arcs.

In the directed case, the problem has a feasible solution provided that the graph is strongly connected, i.e., there exists a directed path between every pair of nodes. The optimal solution is found by a polynomial time algorithm as proposed in Eiselt et al (1995). It is shown that a minimum cost unicursal graph can be

constructed using the transportation model where the decision variables represent the number of times each arc has to be traversed.

Windy graphs are undirected graphs in which the cost of traversing an arc depends on the direction of travel. Windy postman problem (WPP) was firstly introduced by Miniéka (1979). Brucker (1981) shows that the WPP is strongly NP-hard and Win (1989) shows that the problem is polynomially solvable when the graph is unicursal. Grötschel and Win (1992) describe the cutting plane algorithm to solve the WPP.

Mixed graphs contain both undirected and directed arcs and consist of determining a least-cost traversal through every arc of the graph at least once. Papadimitriou (1976) shows that finding the minimum cost solution to the mixed CPP is strongly NP-hard and Edmonds and Johnson (1973) show that the problem is polynomially solvable when the graph is unicursal. To find the optimal solution, some authors, including Christofides et al. (1984) and Nobert and Picard (1991), have used integer linear programming formulations. The branch and cut algorithm is proposed by Corberan et al. (2012) to solve large size Mixed CPP instances.

Prakash et al. (2009) consider two additive objectives for the CPP and presents a heuristic procedure that finds a set of feasible solutions giving higher priority to the first objective. The performance of Prakash et al. (2009)'s procedure depends on the choice of the priorities and the procedure may return a single nondominated point or a subset of all nondominated points or a set of all nondominated points.

The biobjective arc routing problems with multi postmen have also been studied in the last decade. Lacomme et al. (2006), Mei et al. (2011) and Grandinetti et al. (2012) propose heuristic approaches that return the approximate set of nondominated points with respect to the total cost and maximum cost criteria.

In this study, we propose two approaches each returns the set of all nondominated points for the CPP. To the best of our knowledge, our study is the first attempt for the exact solution of the bi-objective arc routing problems.

### 3. PROBLEM DEFINITION AND THE RELATED MODELS

Consider a directed graph  $G = (N, A)$  that consists of a set of  $A$  arcs and a set of  $N$  nodes. Arc  $(i, j)$  establishes a connection between nodes  $i$  and  $j$  and is characterized by two weights,  $c_{ij}$  and  $d_{ij}$ .  $c_{ij}$  may stand for the cost of traversing arc  $(i, j)$  whereas  $d_{ij}$  is its travel time or distance. The main decision of the problem is explained via the variable  $X_{ij}$  as follows:

$X_{ij}$  = number of times arc  $(i, j)$  is traversed

The objective of the classical CPP is to minimize the total cost that is expressed as

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} X_{ij} \quad (1)$$

The constraint sets are stated below:

i. Each arc should be traversed at least once:

$$X_{ij} \geq 1 \text{ and integer } \forall (i, j) \in A \quad (2)$$

ii. The flow to a node should be conserved that is for each entering arc, there should be a departing counterpart.

$$\sum_{j | (i,j) \in A} X_{ij} = \sum_{j | (j,i) \in A} X_{ji} \quad \forall i \in N \quad (3)$$

We simply refer to the above CPP model that minimizes (1) while satisfying (2) and (3) as  $P_C$ .  $P_D$  is the another CPP model that minimizes (4) below while satisfying (2) and (3).

$$\text{Minimize } \sum_{(i,j) \in A} d_{ij} X_{ij} \quad (4)$$

#### Solution Procedures for the classical CPP

Eiselt et al (1995) show that the classical CPP can be represented by a transportation model, hence can be solved in polynomial time. The decision variable set of the transportation model,  $X_{ij}$  is as defined below:

$X_{ij}$  = number of times the shortest path between  $(i, j)$  is traversed

The transportation model uses the following definitions:

- i.  $d_i^+ = \begin{cases} \text{number of incoming arcs to node } i \\ \text{indegree of node } i \end{cases}$   
 $d_i^- = \begin{cases} \text{number of outgoing arcs from node } i \\ \text{outdegree of node } i \end{cases}$

Node  $i$  is called balanced if  $d_i^+ = d_i^-$ .

ii. The sets below are defined for the unbalanced nodes:

$$I = \text{Set of supply nodes} = \{i | d_i^+ > d_i^-\}$$

$$J = \text{Set of demand nodes} = \{i | d_i^+ < d_i^-\}$$

$$\text{iii. } S_i = d_i^+ - d_i^- \quad \forall i \in I$$

$$D_j = d_j^- - d_j^+ \quad \forall j \in J$$

iv.  $SP_{ij}$  = length of the shortest path between nodes  $i$  and  $j$

$$\forall i \in I, \quad \forall j \in J$$

The model is as stated below:

$$\text{Minimize } \sum_{i \in I} \sum_{j \in J} SP_{ij} X_{ij} + \sum_{(i,j) \in A} c_{ij} X_{ij}$$

subject to

$$\sum_{j \in J} X_{ij} = S_i \quad \forall i \in I$$

$$\sum_{i \in I} X_{ij} = D_j \quad \forall j \in J$$

$$X_{ij} \geq 0 \text{ and integer } \quad \forall i \in I, j \in J$$

An optimal objective function value of the above transportation model gives the optimal total cost of the classical CPP.

#### The Constrained Chinese Postman Problem

Consider the following constraint that imposes an upper bound  $k$  on the second criterion, i.e., total distance travelled:

$$\sum_{(i,j) \in A} d_{ij} X_{ij} \leq k \quad (5)$$

Minimizing (1) subject to the constraint sets (2), (3) and (5) is a single constrained CPP model. We hereafter refer to the single constrained CPP as  $P_C, k$ .

Figure 1 illustrates the feasible and optimal solutions of the  $P_C$  and  $P_C, k$  problems via a 25 nodes and 43 arcs instance taken from Malandraki and Daskin (1993).

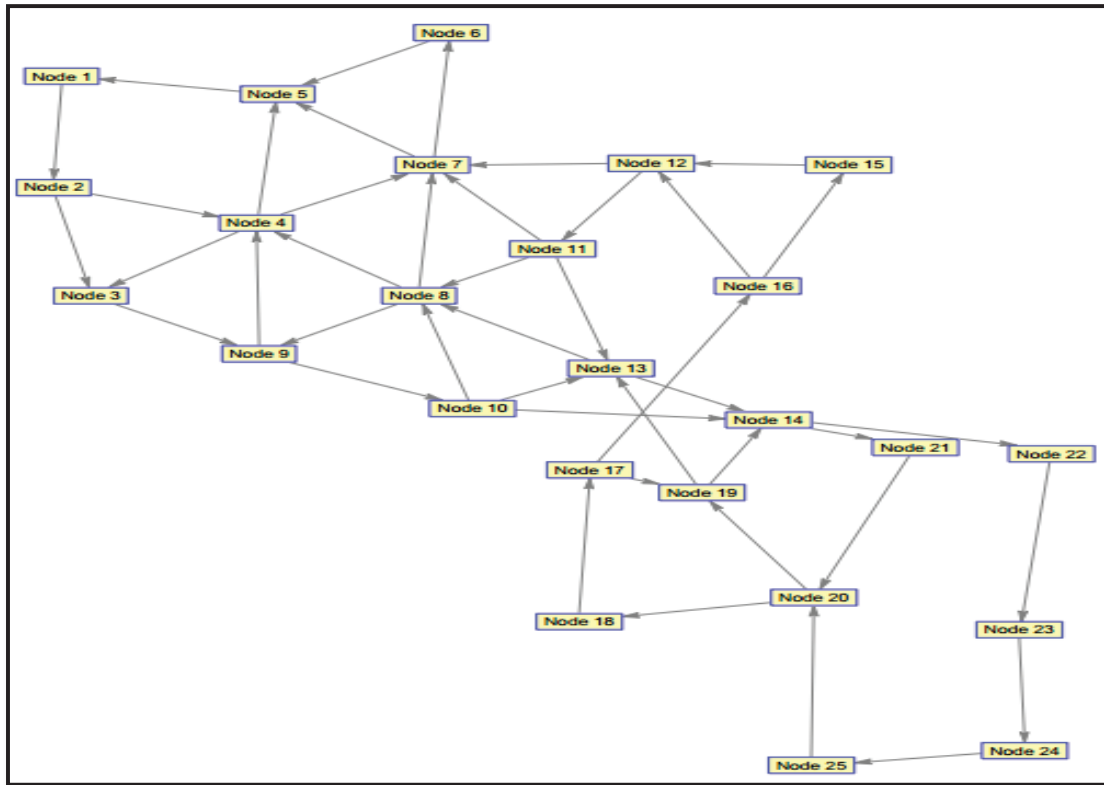


Figure 1. The Example Instance

The arc weights, i.e.  $(c_{ij}, d_{ij})$  values are given in Table 1 below.

Table 1. The Arc Weights of the Example Instance

$(i, j)$	$(c_{ij}, d_{ij})$	$(i, j)$	$(c_{ij}, d_{ij})$	$(i, j)$	$(c_{ij}, d_{ij})$	$(i, j)$	$(c_{ij}, d_{ij})$
(1,2)	(40,40)	(8,4)	(50,50)	(12,7)	(90,90)	(18,17)	(30,30)
(2,3)	(30,30)	(8,7)	(30,30)	(12,11)	(20,20)	(19,13)	(40,40)
(2,4)	(40,12)	(8,9)	(60,60)	(13,8)	(70,70)	(19,14)	(70,70)
(3,9)	(40,40)	(9,4)	(70,70)	(13,14)	(70,21)	(20,18)	(30,30)
(4,3)	(40,12)	(9,10)	(60,60)	(14,21)	(20,80)	(20,19)	(30,30)
(4,5)	(30,30)	(10,8)	(60,60)	(14,22)	(40,40)	(21,20)	(20,70)
(4,7)	(50,50)	(10,13)	(60,9)	(15,12)	(40,20)	(22,23)	(30,30)
(5,1)	(50,50)	(10,14)	(30,36)	(16,12)	(40,60)	(23,24)	(30,30)
(6,5)	(50,15)	(11,7)	(80,80)	(16,15)	(40,20)	(24,25)	(80,20)
(7,5)	(50,50)	(11,8)	(70,70)	(17,16)	(40,40)	(25,20)	(70,20)
(7,6)	(30,9)	(11,13)	(30,30)	(17,19)	(30,30)		

The optimal solution to the  $P_C$  problem is stated below:

$$X^*_{1,2} = 5, X^*_{2,3} = 4, X^*_{5,1} = 5, X^*_{7,5} = 3, X^*_{3,9} = 5, X^*_{9,10} = 5, X^*_{12,11} = 3, X^*_{16,12} = 3,$$

$$X^*_{10,14} = 3, X^*_{13,14} = 2, X^*_{17,16} = 4, X^*_{18,17} = 5, X^*_{20,18} = 5, X^*_{21,20} = 5, X^*_{14,21} = 5$$

and all other  $X^*_{i,j}$  values are equal to 1.

The optimal total cost  $Z_C^* = \sum_{(i,j) \in A} c_{ij} X_{ij}^*$

$$Z_C^* = 40 \times 5 + 30 \times 4 + 40 \times 1 + \dots + 30 \times 1 + 80 \times 1 + 70 \times 1 = 3700$$

The optimal solution to the  $P_D$  problem is stated below:

$$X^*_{1,2} = 5, X^*_{2,4} = 4, X^*_{4,3} = 4, X^*_{5,1} = 5, X^*_{6,5} = 3, X^*_{7,6} = 3, X^*_{3,9} = 5, X^*_{9,10} = 5, X^*_{12,11} = 3, X^*_{15,12} =$$

$$3, X^*_{16,15} = 3, X^*_{10,13} = 3, X^*_{13,14} = 4, X^*_{17,16} = 4, X^*_{18,17} = 5, X^*_{20,18} = 5, X^*_{22,23} = 5, X^*_{14,22} = 5,$$

$$X^*_{23,24} = 5, X^*_{24,25} = 5, X^*_{25,20} = 5$$

and all other  $X^*_{i,j} = 1$ .

The optimal total distance  $Z_D^* = \sum_{(i,j) \in A} d_{ij} X_{ij}^*$

$$Z_D^* = 40 \times 5 + 30 \times 1 + \dots + 30 \times 5 + 20 \times 5 + 20 \times 5 = 3755$$

Now assume  $k$  is 3890.

The constraint  $\sum_{(i,j) \in A} d_{ij} X_{ij}^* \leq 3890$  is not satisfied since  $\sum_{(i,j) \in A} d_{ij} X_{ij}^* = 3917$ .

The  $(P_C, k)$  problem with  $k$  value of 3890 returns the following optimal solution:

$$X^*_{1,2} = 5, X^*_{2,3} = 4, X^*_{5,1} = 5, X^*_{6,5} = 3, X^*_{7,6} = 3, X^*_{3,9} = 5, X^*_{9,10} = 5, X^*_{12,11} = 3, X^*_{16,12} = 3,$$

$$X^*_{10,14} = 3, X^*_{13,14} = 2, X^*_{17,16} = 4, X^*_{18,17} = 5, X^*_{20,18} = 5, X^*_{21,20} = 5, X^*_{14,21} = 5$$

where all other  $X^*_{i,j}$  values are equal to 1.

The optimal total cost, which is denoted by  $(Z_C, k)^*$ , is equal to:

$$Z_{C,k}^* = \sum_{(i,j) \in A} c_{ij} X_{ij}^* = 40 \times 5 + 30 \times 4 + 40 \times 1 + \dots + 30 \times 1 + 80 \times 1 + 70 = 3760$$

The travel distance constraint is also satisfied:

$$\sum_{(i,j) \in A} d_{ij} X_{ij}^* = 3865 < 3890$$

Note that  $(Z_C, k)^*$  is bigger than  $Z_C^*$ , i.e.,  $Z_C^* = 3700 < 3760 = Z_{C,k}^*$



The steps our study can be summarized as follows: We first define the properties of the efficient solutions and use them in our mixed integer linear model and its linear programming relaxation. We use mixed integer linear model in classical approach for sequential generation of all nondominated objective vectors. We use the linear programming relaxation of the model in branch and bound algorithm for simultaneous generation of all nondominated objective vectors. The final step of the study evaluates the performances of the classical approach and branch and bound algorithm.

#### 4. NONDOMINATED OBJECTIVE VECTORS AND THE CLASSICAL APPROACH

A solution  $u$  is called efficient if there is no other solution  $v$ , having  $Z_C^v \leq Z_C^u$  and  $Z_D^v \leq Z_D^u$  with strict inequality holding at least once. The resulting objective function vector  $(Z_C^u, Z_D^u)$  is said to be nondominated and we say objective vector  $(Z_C^v, Z_D^v)$  dominates the objective vector  $(Z_C^u, Z_D^u)$ .

In subsection 4.1 we define the properties of the efficient solutions. In subsection 4.2 we define a way to find a single nondominated objective vector and subsection 4.3 we discuss the procedure that finds all nondominated objective vectors.

##### 4.1 Properties of the Efficient Solutions

We present three properties of the efficient solutions.

*Property 1.* In all efficient solutions, for any node  $i$  whose indegree is 1, i.e.,  $d_i^+ = 1$ ,  $x_{ji} \geq d_i^-$  where arc  $(j, i)$  is incident to node  $i$ .

*Proof.* There are at least  $d_i^-$  inflows to node  $i$ ; hence, at least  $d_i^-$  outflows from node  $i$ . The only way of realizing outflows is traversing arc  $(j, i)$ , at least  $d_i^-$  times. It follows that in all feasible, hence efficient, solutions,  $x_{ji} \geq d_i^-$ .

*Property 2.* In all efficient solutions, for any node  $i$  whose outdegree is 1, i.e.  $d_i^- = 1$ ,  $x_{ij} \geq d_i^+$  for the arc  $(i, j)$  incident to node  $i$ .

*Proof.* Similar to that of Property 1; hence, it is omitted.

*Property 3.* In all efficient solutions, an arc incident to node  $r$  is traversed exactly once if node  $r$  is not on any shortest path that connects  $i \in I$  to  $j \in J$ .

*Proof.* An optimal solution to the transportation problem (discussed in Section 3) that is equivalent to the  $P_C$  problem ( $P_D$  problem) with arc weights  $c_{ij}$  (arc weights  $d_{ij}$ ) traverses any incident arc to a balanced node  $r$  exactly once. This follows that if any balanced node is not on any shortest path between  $i \in I$  to  $j \in J$ , relative to both arc weights, i.e.,  $c_{ij}$  and  $d_{ij}$  then it is traversed exactly once in all efficient solutions.

Using the result of Property 3, if node  $r$  is not on any shortest path between  $i \in I$  to  $j \in J$ , relative to both arc weights we set the flows of all arcs incident to node  $r$  to 1, and call node  $r$  as eliminated node. We define

$$A_{ir} = \{i \mid \text{arc}(r, i) \text{ exists}\}$$

$$B_{jr} = \{j \mid \text{arc}(j, r) \text{ exists}\}$$

We thereafter refer set  $E$  as the set of eliminated nodes and redefine  $N$  as  $N/E$ .

As  $X_{j,r} = X_{r,i}$  for all  $(j, r)$  and  $(r, i)$  the flow conservation relations are rewritten as:

$$|A_{ir}| + \sum_j X_{i,j} = \sum_j X_{j,i} + |B_{jr}|$$

$$\forall r \in E, \forall i, j \text{ incident to } r$$

Using the results of Property 1 and Property 2, the following constraints are added.

$$X_{ij} \geq d_i^+ \quad \forall i \mid d_i^- = 1$$

$$X_{ij} \geq d_i^- \quad \forall i \mid d_i^+ = 1$$

##### 4.2 Finding A Nondominated Objective Vector

Consider the following modified  $P_C$  model.

$$\text{Minimize } \sum_{(i,j) \in A} c_{ij} X_{ij} \quad (5)$$

subject to

$$X_{ij} \geq 1 \quad (6)$$

$$X_{ij} \geq d_i^+ \quad \forall i \mid d_i^- = 1 \quad (7)$$

$$X_{ij} \geq d_i^- \quad \forall i \mid d_i^+ = 1 \quad (8)$$

$$\sum_j X_{ij} = \sum_j X_{ji} \quad \forall i \in N \quad (9)$$

$$|A_{ir}| + \sum_j X_{i,j} = \sum_j X_{j,i} + |B_{jr}| \quad (10)$$

$\forall r \in E, \forall i, j \text{ incident to } r$

We hereafter refer to constraint sets (6) through (10) as  $x \in X$ .

Let  $Z_C^*$  be its optimal  $Z_C$  value.  $Z_C^*$  is a valid lower bound on the  $Z_C$  values of all efficient solutions. Any optimal solution to the  $P_C$  problem may not be efficient since there may exist alternative optimal solutions having smaller  $Z_D$  values.

Among the alternative optimal solutions to the  $P_C$  problem, the one having the smallest  $Z_D$  value requires the solution of the following problem.

$$\text{Minimize } \sum_{(i,j) \in A} d_{ij} X_{ij}$$

subject to  $x \in X$

$$\sum_{(i,j) \in A} c_{ij} X_{ij} = Z_C^*$$

We hereafter refer to the above model as  $(P_D, Z_C^*)$ . Note that  $(P_D, Z_C^*)$  requires solving the  $P_C$  problem. In place of solving the  $P_C$  problem and then the  $(P_D, Z_C^*)$  problem, one may solve the following problem:

$$\text{Minimize } \left( \sum_{(i,j) \in A} c_{ij} X_{ij} \right) + \epsilon_D \left( \sum_{(i,j) \in A} d_{ij} X_{ij} \right)$$

subject to  $x \in X$

The value of  $\epsilon_D$  should be small enough that the smallest  $\sum_{(i,j) \in A} c_{ij} X_{ij}$  value should not increase even for the largest possible value of  $\sum_{(i,j) \in A} d_{ij} X_{ij}$ . Accordingly,

$$Z_C^* + \epsilon_D D_{max} \leq Z_C^* + 1 + \epsilon_D D_{min} \quad (11)$$

where  $D_{min}$  and  $D_{max}$  are the smallest and largest possible value of the total distance, respectively. The inequality (11) follows that

$$\epsilon_D (D_{max} - D_{min}) \leq 1$$

$$\epsilon_D \leq \frac{1}{(D_{max} - D_{min})}$$

$D_{min}$  is the optimal  $Z_D$  value of the  $P_D$  problem.  $D_{max}$  is an upper bound on the  $Z_D$  value of all efficient solutions; hence, it is the  $(\sum_{(i,j) \in A} d_{ij} X_{ij})$  value of the  $(P_C)$  problem. In other words, first we solve the  $P_C$  problem and obtain optimal  $X_{ij}$  values, then by using those  $X_{ij}$  values in equation of  $(\sum_{(i,j) \in A} d_{ij} X_{ij})$ , we obtain the  $D_{max}$  value.

In our experiments, we set  $\epsilon_D$  to  $\frac{1}{(D_{max} - D_{min}) + 1}$

### 4.3 Finding the Set of All Nondominated Objective Vectors

An optimal solution to the following constrained problem,  $P_C, k$ , gives a nondominated objective vector and a corresponding efficient solution (see Haimes et al., 1971).

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

subject to  $x \in X$

$$Z_D \leq k$$

This follows, all nondominated objective vectors and corresponding efficient solutions can be generated by solving  $(P_C, k)$  for all  $k$  in  $[D_{min}, D_{max}]$ . The following procedure systematically varies the value of  $k$  and returns the set of all nondominated objective vectors.

#### Procedure 1. The Classical Approach (CA)

Step 0. Solve  $(P_C)$  Minimize  $Z_C$  subject to  $x \in X$   
Let  $D_{max}$  be the  $Z_D$  value of the  $P_C$

Solve  $(P_D)$  Minimize  $Z_D$  subject to  $x \in X$   
Let  $D_{min}$  be the  $Z_D$  value of the  $P_D$

$$\text{Let } \epsilon_D = \frac{1}{(D_{max} - D_{min}) + 1}$$

Step 1. Solve  $(P_C, k)$  Minimize  $Z_C + \epsilon_D Z_D$   
subject to  $x \in X$   
 $Z_D \leq k$

Let  $(Z_C^*, Z_D^*)$  be the solution.

Set  $r = r + 1$

Step 2. If  $Z_D^* = D_{min}$  then stop, all  $r$  nondominated objective vectors are generated.

Else set  $k = Z_D^* - 1$  and go to Step 1.

Note that each iteration Procedure 1 generates a nondominated objective vector. When the procedure terminates, the set of all nondominated objective vectors are generated.

The number of nondominated objective vectors,  $r$ , is at most  $D_{max} - D_{min} + 1$ . Hence, the Procedure 1 iterates pseudo polynomial number of times.

## 5. COMPLEXITY AND THE BRANCH AND BOUND ALGORITHM

In this section, we first settle the complexity of the constrained CPP, i.e.,  $P_C$ ,  $k$  and then present a Branch and Bound algorithm for its exact solution.

*Theorem 1.* The constrained CPP is strongly NP-hard.

*Proof.* Recall that the CPP reduces to the following ‘Transportation Problem’ (see Section 3).

$$\begin{aligned} & \text{Minimize } \sum_{i \in I} \sum_{j \in J} SP_{ij} X_{ij} + \sum_{(i,j) \in A} c_{ij} \\ & \sum_{j \in J} X_{ij} = S_i \quad \forall i \in I \\ & \sum_{i \in I} X_{ij} = D_j \quad \forall j \in J \\ & X_{ij} \geq 0 \text{ and integer} \quad \forall i \in I, j \in J \end{aligned}$$

Hence, our constrained CPP problem is equivalent to a ‘Capacitated Transportation Problem’ with the following capacity constraint.

$$\sum_{i \in I} \sum_{j \in J} d_{ij} X_{ij} \leq \bar{D} \text{ where } \bar{D} = k - \sum_{(i,j) \in A} d_{ij}$$

When all supply and demand amounts are ‘1’, the ‘Capacitated Transportation Problem’ reduces to the Generalized Assignment Problem (GAP). It follows that the  $P_C$ ,  $k$  problem reduces to the GAP. The GAP is NP-hard in the strong sense (see, Martello and Toth, 1990), so is the  $P_C$ ,  $k$  problem.

The problem of generating a single nondominated objective vector is strongly NP-hard as the constrained CPP is strongly NP-hard. It follows that the problem of generating all nondominated objective vectors with respect to two objectives is strongly NP-hard.

Attributing to the complexity of the problem, Procedure 1 that generates all nondominated objective vectors is likely to fall into computational burden. To dispel the burden to some extent, we present an implicit enumeration technique – a Branch and Bound (BAB) algorithm. Our aim is to attain optimal solutions to the large sized instances in reasonable solution times.

Our BAB algorithm generates one nondominated objective vector at a time. It starts with an initial upper bound (incumbent solution) and updates it whenever a complete solution with a better objective function value is reached.

We solve the Linear Programming Relaxation (LPR), i.e., first relax the integrality constraints on the  $X_{ij}$  values and then solve the resulting problem to optimality. We benefit from the optimal solution of the LPR to find an initial upper bound on the  $Z_C$  value. Below is the stepwise description of our upper bounding procedure.

### Procedure 2 - Finding an Initial Upper Bound

*Step 1.* Solve the LPR.

*Step 2.* Let  $\tau$  be a cycle that resides all fractional variables. Let  $X_{rs}^L$  be the smallest fractional variable in  $\tau$ .

*Step 3.* Let  $X_{rs} = \lceil X_{rs}^L \rceil$  and update the fractional variables around  $\tau$  so as to preserve the flow conservation relations.

*Step 4.* If  $X_{rs} = \lceil X_{rs}^L \rceil$  gives a feasible solution, then go to Step 5.

Else let  $X_{rs} = \lfloor X_{rs}^L \rfloor$  and update the fractional variables around.

*Step 5.* If all variables are integers, stop.

Else go to Step 2.

At each node of the BAB tree, we solve the LPR and explore the tree using the fractional variables of the optimal LPR solution. We let  $X_{rs}$  be the maximum fractional value and generate the following two child nodes:

*Child Node I.* Add constraint  $X_{rs} > \lceil X_{rs}^L \rceil$

*Child Node II.* Add constraint  $X_{rs} \leq \lfloor X_{rs}^L \rfloor$

We use the optimal solution value of the LPR model

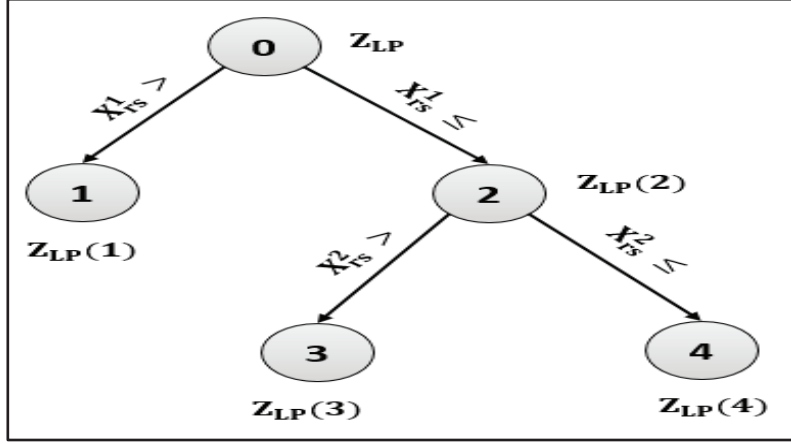


Figure 2. The BAB Tree

as a lower bound. We select the node having the smaller lower bound. For the selected node, we find an upper bound using the upper bounding procedure. Figure 2 illustrates our BAB tree.

Nodes 1 and 2 are generated based on the optimal solution of the LPR solved at Node 0. Node 2 is selected for further branching as  $Z_{LP}^2 \leq Z_{LP}^1$ . The child nodes of Node 2, i.e., Node 3 and Node 4, use the additional constraints  $X_{rs}^2 \geq \lceil X_{rs}^2 \rceil$  and  $X_{rs}^2 \leq \lfloor X_{rs}^2 \rfloor$ , respectively. The tree explores either from Node 3 if  $Z_{LP}^3 \leq Z_{LP}^4$  or from Node 4 if  $Z_{LP}^3 > Z_{LP}^4$ .

A node is fathomed if one of the following conditions holds:

- i. The resulting LPR leads to an infeasible solution.
- ii. The resulting LPR gives all integer decision variables. In such a case, the solution  $(Z_C^l, Z_D^l)$  is updated if  $Z_C < Z_C^l$  or  $Z_C = Z_C^l$  and  $Z_D < Z_D^l$ .
- iii. The objective function value of the resulting LPR is no better than the incumbent solution. That is,  $Z_C^{LP} > Z_C^l$  or  $Z_C^{LP} = Z_C^l$  and  $Z_D^{LP} > Z_D^l$ .

We employ a depth first strategy due to its relatively low memory requirements. According to the strategy, we start from the root node and explore branching from the node having smaller  $Z_C$  value or smaller  $Z_D$  value when the  $Z_C$  values are equal.

If both nodes are fathomed, we backtrack to the previous level. We stop when we backtrack to Level 1. The best solution at termination, i.e., incumbent solution is the optimal solution.

We illustrate our branching scheme on example instance I. The optimal solution of the LPR at the initial level is 4211 when  $k = 3794$ . The initial upper bound is found as 4400.

The BAB tree of the instance is presented in Figure 3.

The figures on the nodes indicate the order at which they are created. At each node, the lower and upper bound values are reported. At Node 1, the following LPR is solved:

$$\begin{aligned} & \text{Minimize } Z_C + \epsilon_D Z_D \\ & \text{Subject to } x \in X \\ & \quad X_{14,21} \leq 4 \end{aligned}$$

The optimal objective function value of the associated LPR is found as  $Z_{C,k} = 4250$  for  $k = 3794$ . We update the upper bound at that node. According to the optimal solution of the LPR, the cycle that resides the fractional variables is given in Figure 4.

We choose the smallest fractional variable in the cycle, i.e.,  $X_{10,13} = 1.5$ . We let  $X_{10,13} = 2$  and update the fractional variables around the cycle as shown in Figure 5.

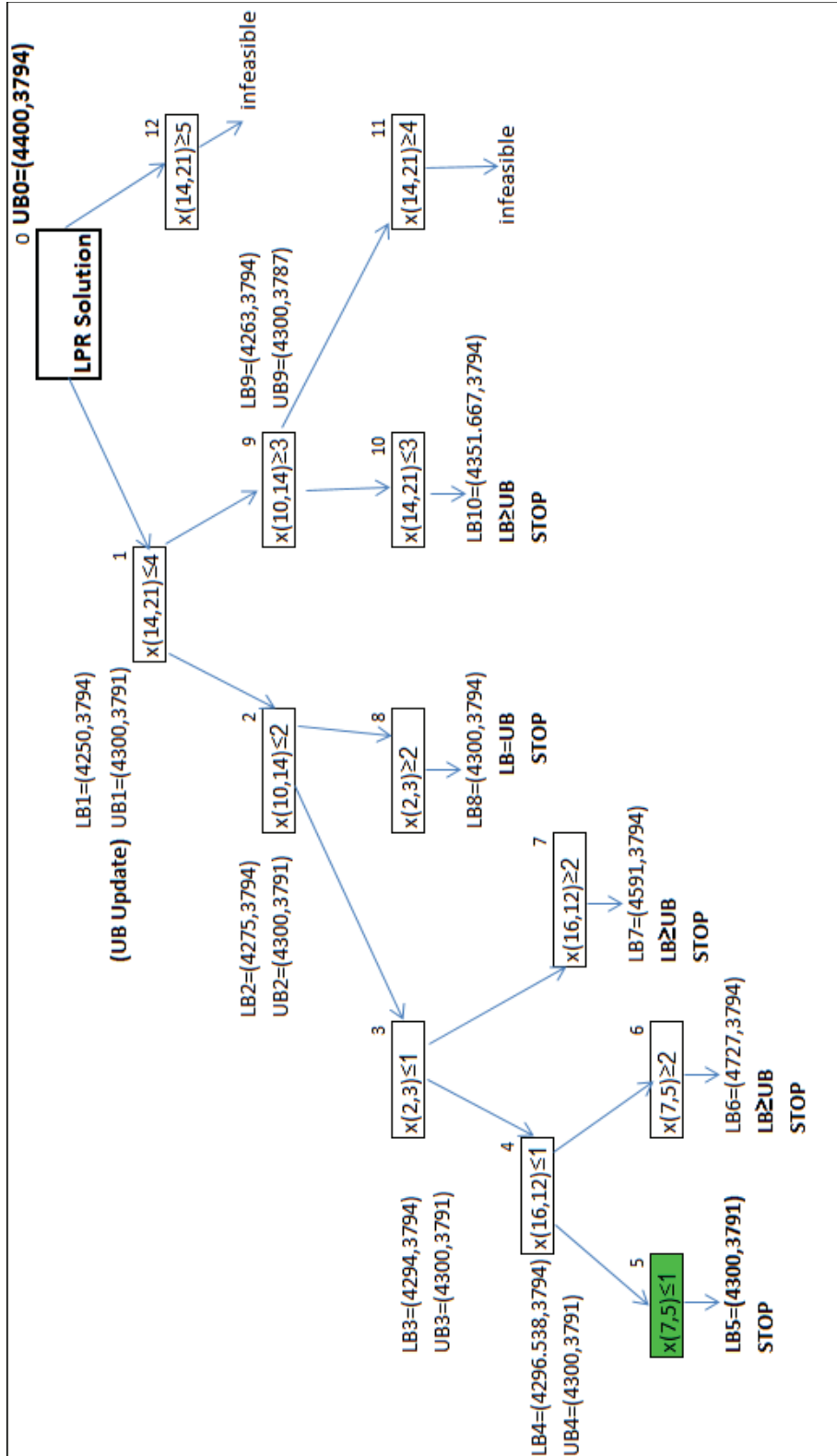
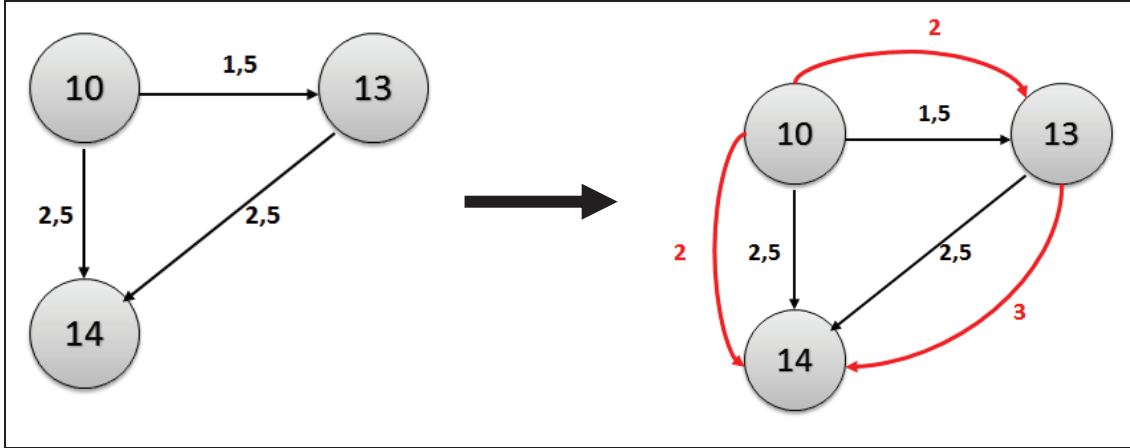


Figure 3. The BAB Tree of the Instance



**Figure 4.** The Illustration of the Upper Bounding Procedure

The upper bounds for the total cost and total distance are found as:

$$UB_C^1 = 4250 + 0.5 (60 - 30 + 70) = 4300$$

$$UB_D^1 = 3794 + 0.5 (9 - 36 + 21) = 3791$$

Then, we update the best upper bound from  $(UB_C, UB_D) = (4400, 3794)$  to  $(4300, 3791)$ . At Node 2, we solve the following LPR model:

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

$$\text{Subject to } x \in X$$

$$X_{14,21} \leq 4, X_{10,14} \leq 2$$

The upper bound in terms of total cost and total distance is then equal to:

$$UB_C^1 = 4250 + 0.5 (60 - 30 + 70) = 4300$$

$$UB_D^1 = 3794 + 0.5 (9 - 36 + 21) = 3791$$

Then, we update the upper bound from  $(UB_C, UB_D) = (4400, 3794)$  to  $(4300, 3791)$ .

At Node 2, we solve the following LPR model:

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

$$\text{Subject to } x \in X$$

$$X_{14,21} \leq 4, X_{10,14} \leq 2$$

The optimal solution of the above LPR is found  $Z_{C,k} = 4275$  for  $k = 3794$ . The upper bound is not updated as it is no better than the one found in Node 1. According to the solution of the LPR at Node 2, the cycle that resides fractional variables are  $X_{2,3} = 1.5, X_{2,4} = 3.5, X_{4,3} = 3.5$ . We choose the smallest fractional variable in the cycle, i.e.,  $X_{2,3} = 1.5$ . Then, at Node 3, we solve the following LPR model:

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

$$\text{Subject to } x \in X$$

$$X_{14,21} \leq 4, X_{10,14} \leq 2, X_{2,3} \leq 1$$

The optimal solution of the above LPR is found as  $Z_{C,k} = 4294$  for  $k = 3794$ .

According to the solution of LPR at Node 3,  $X_{15,12} = 2.85, X_{16,12} = 1.15, X_{16,15} = 2.85$  resides fractional cycle. We choose the smallest fractional variable in the cycle, i.e.,  $X_{16,12} = 1.15$ . Then, at Node 4, we solve the following LPR model:

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

$$\text{Subject to } x \in X$$

$$X_{14,21} \leq 4, X_{10,14} \leq 2, X_{2,3} \leq 1, X_{16,12} \leq 1$$

The optimal solution of the LPR is found as  $Z_{C,k} = 4296,538$  for  $k = 3794$ . According to the solution of LPR



at Node 4, the cycle that resides fractional variables are  $X_{6,5} = 2.885$ ,  $X_{7,5} = 1.115$ ,  $X_{7,6} = 2.885$ .

At Node 5, when we solve the following LPR by choosing the smallest fractional variable in the cycle, the optimal solution is equal to the updated upper bound value at Node 1, i.e.,  $(LB_C, LB_D) = (4300, 3791)$ . Since the resulting LPR gives all integer decision variables, we stop branching and continue to explore from the other node at the same level.

$$\text{Minimize } Z_C + \epsilon_D Z_D$$

$$\text{Subject to } x \in X$$

$$X_{14,21} \leq 4, X_{10,14} \leq 2, X_{2,3} \leq 1, X_{16,12} \leq 1, X_{7,5} \leq 1$$

Nodes 6, 7, 8 and 10 are fathomed since the objective functions of the resulting LPRs are greater than or equal to those of the incumbent solution. Nodes 11 and 12 are also fathomed as their since the resulting LPRs leads to an infeasible solution. The best solution at termination is optimal and it is found at Node 5.

## 6. COMPUTATIONAL EXPERIMENT

We design an experiment to evaluate the performance of our algorithms. The number of nodes and number of arcs used in our experiment are tabulated in Table 2.

**Table 2.** The (N, M) Values

<b>N</b>	100	100	300	300	300	500	500
<b>M</b>	200	400	600	675	750	1200	1300

The networks with 100 nodes are from <http://www.ing.unibs.it/~orgroup/instances.html>. The networks with 300 and 500 nodes are from <http://www.uv.es/corberan/instancias.html>. We generate the arc weights ( $c_{ij}$  and  $d_{ij}$  values) from discrete uniform distributions in  $[1, 100]$ .

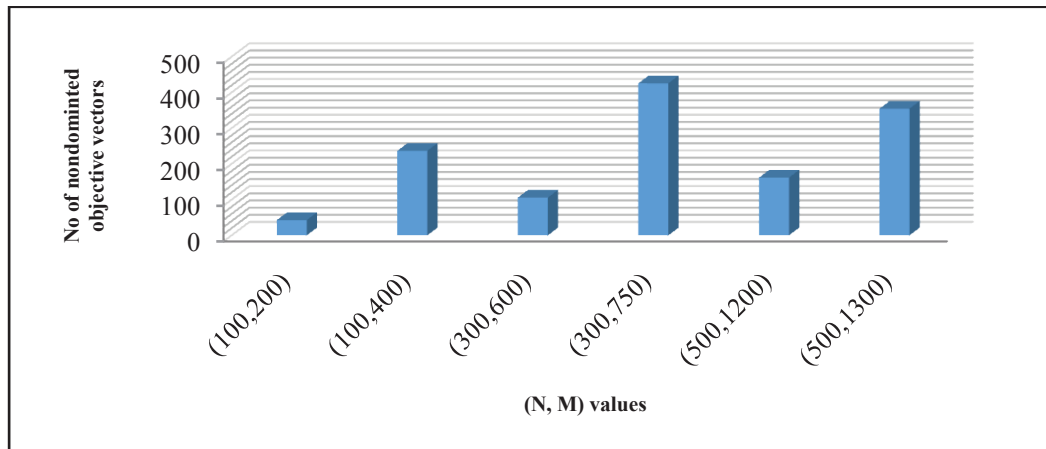
All experiments are carried out on an Intel(R) Core(TM) i5-3317U and clocked at 1.70 GHz with 4 GB RAM. The BAB algorithm is coded in Java, Eclipse Luna version 4.4.0.

For each (N, M) combination, we generate 10 problem instances. Hence, our experiment resides 60 problem instances. For each instance, we generate the set of all nondominated objective vectors; hence solve many combinatorial optimization problems.

We set a termination limit of 3600 seconds for the BAB algorithm and CA.

**Table 3.** The Number of the Nondominated Objective Vectors

N	M	Number of nondominated objective vectors	
		Average	Maximum
100	200	43	150
100	400	237	425
300	600	106	152
300	750	426	904
500	1200	162	294
500	1300	355	503



**Figure 5.** The Average Number of the Nondominated Objective Vectors

The number of nondominated objective vectors is reported in Table 3. The table includes the average and maximum number of nondominated objective vectors for each problem combination.

The average number of the nondominated objective vectors for each combination is also represented in Figure 5. The figure shows the effect of the problem size parameters on the number of the nondominated objective vectors.

Note from Figure 5 that the number of the nondominated objective vectors is highly dependent on the problem size. Note from the figure that for  $N = 100$ , the average number of the nondominated objective vectors increases from 43 to 237 as  $M$  increases from 200 to 400. For  $N = 300$ , these increases are from 106 to 426 when  $M$  increases from 600 to 750. For  $N = 500$ , the average CPU times increases more than two times when  $M$  increases by 100.

Table 4 reports the average and maximum CPU times in seconds and the number of nodes generated BAB algorithm in generating a nondominated objective vector.

We observe from Table 4 that as  $N$  or  $M$  increases, the CPU time to find a nondominated objective vector increases significantly. Finding a nondominated objective vector takes 0.382 seconds on average for the networks with 100 nodes and 200 arcs, while it reaches to 1.854 seconds for the network with  $(N, M)$  values of  $(500, 1200)$ . When  $N = 100$  and  $M = 200$ , the average number of nodes generated is 16 and the average CPU time to find a nondominated objective vector is 0.382 seconds. For  $N = 100$ , when  $M$  increases to 400, the average number of nodes and CPU time rise to 141 and 1.243 seconds, respectively. The increases in the complexity of the solutions with increases in  $N$  or  $M$  can be attributed to the increases in the complexity of the LPR models.

**Table 4.** The Performance of the BAB and CA-per Nondominated Objective Vector

N	M	CA CPU Time		BAB CPU Time		BAB Nodes	
		Average	Maximum	Average	Maximum	Average	Maximum
100	200	0.298	0.490	0.382	1.069	16	27
100	400	0.287	0.311	1.243	1.613	141	197
300	600	2.593	3.115	1.01	1.319	70	96
300	750	2.292	2.993	2.779	3.765	177	235
500	1200	5.183	5.727	1.854	2.431	187	384
500	1300	5.286	5.696	4.224	5.563	152	202

**Table 5.** The Performance of the BAB and CA-per Nondominated Objective Vector

N	M	CA CPU Time		BAB CPU Time		BAB Nodes	
		Average	Maximum	Average	Maximum	Average	Maximum
100	200	11.448	33.43	8.543	19.097	848	3138
100	400	67.179	118.23	306.863	634.37	35030	70660
300	600	267.643	352.9	107.245	182.76	7505	13478
300	750	1017.37	2210.95	1254.83	2812.6	79855	175564
500	1200	833.533	1506.47	306.508	561.688	24377	32992
500	1300	1879.3	2604.95	1528.88	2716.86	54747	93794

Table 5 reports on the performance of generating the set of nondominated objective vectors.

Note from Table 5 that when  $N$  or  $M$  increases, the number of nondominated objective vectors and time to generate a single nondominated objective vector increase significantly. These in turn, increase the time to generate the set of all nondominated objective vectors. For the  $(N, M)$  values of  $(100, 400)$ ,  $(300, 750)$  and  $(500, 1300)$ , the average CPU times are about 307, 1255 and 1539 seconds, respectively. The CPU times are almost tripled when the problem sizes are increased from  $(100, 400)$  to  $(500, 1300)$ .

Tables 4 and 5 also report on the performance of the CA. We observe that as  $N$  or  $M$  increases, the CA finds the exact nondominated objective vector set in considerably higher CPU times. The significant increases in the CPU times can be attributed to the increases in the complexity of the integer models that return a single nondominated objective vector and increases in the number of nondominated objective vectors.

The differences between the performances of the BAB algorithm and CA increase as  $N$  increases. Note from Table 4 that when  $N = 300$  and  $M = 600$ , the average CPU times per nondominated objective vector are 2.593 and 1.01 seconds for the CA and BAB algorithm, respectively. As another notable example, for  $(N, M)$  values of  $(500, 1200)$ , the average CPU times are 5.183 and 1.854 seconds for the CA and BAB algorithm, respectively.

We also observe that the BAB algorithm behaves more consistent than the CA. Note from Table 5 that, for  $(N, M)$  values of  $(500, 1200)$ , the average and maximum CPU times by the BAB algorithm are 306.508 and 561.688 seconds, respectively. For the CA, the respective average and maximum CPU times are 833.533 and 1506.47 seconds.

## 7. CONCLUSIONS

In this study, we consider a bi-objective CPP with two additive objectives, like total cost and total distance, total distance and total priority, total cost and total time.

We propose two algorithms to generate the exact set of all nondominated objective vectors. The Classical Approach uses the optimal solutions of the mixed integer programs and BAB Algorithm uses the optimal solutions of the linear programming relaxations. The BAB Algorithm returns the set of all nondominated objective vectors for problem instances with up to 500 nodes and 1300 arcs in less than one hour and is superior to the Classical Approach.

To the best of our knowledge, our study is the first attempt for the exact solutions of the bi-objective arc routing problems. Our results derived for the CPP can be extended to more general arc routing problems with more than one postman.

## REFERENCES

1. **Amine, K. & Djellab, R.** 2013. Industrial and Urban Applications of Eulerian and Chinese Walks, In Graph Theory for Operations Research and Management: Applications in Industrial Engineering, IGI-Global, Hershey, 271-279.
2. **Brucker, P.** 1981. "The Chinese Postman Problem For Mixed Graphs. Graph Theoretic Concepts in Computer Science," Springer Berlin Heidelberg.
3. **Christofides, N., Benavent, E., Campos, V., Corberán, A., & Mota, E.** 1984. "An Optimal Method for the Mixed Postman Problem." System Modelling and Optimization. Springer Berlin Heidelberg.
4. **Corberán, A., Oswald, M., Plana, I., Reinelt, G., & Sanchis, J. M.** 2012. "New Results on the Windy Postman Problem." Mathematical Programming, vol. 132, p. 309-332.
5. **Dewil, R., Vansteenwegen, P., & Cattrysse, D.** 2011. Cutting path optimization using Tabusearch. Key Engineering Materials, vol. 473, p. 739-748.
6. **Edmonds, J.** 1963. Chinese postmen problem. Operations Research, 13, B73-B77.
7. **Edmonds, J., & Johnson, E. L.** 1973. "Matching, Euler Tour and the Chinese Postman Problem." Mathematical Programming, vol. 5, p. 88-124.
8. **Eiselt, H. A., Gendreau, M., & Laporte, G.** 1995. "Arc

- Routing Problems, Part I: The Chinese Postman Problem." *Operations Research*, vol. 43, p. 231-242.
9. **Ford, L.R., & Fulkerson, D. R.** 1962. "Flows in Networks," Princeton University Press: Princeton.
  10. **Grandinetti, L., Guerriero, F., Laganà, D., & Pisacane, O.** 2012. "An Optimization-Based Heuristic for the Multi-objective Undirected Capacitated Arc Routing Problem." *Computers & Operations Research*, 39, 2300-2309.
  11. **Grötschel, M., & Win, Z.** 1992. "A Cutting Plane Algorithm for the Windy Postman Problem." *Mathematical Programming*, vol. 55, p. 339-358.
  12. **Guan, M.** 1962. Graphic programming using odd or even points. *Chinese Math*, vol. 110, p. 273-277.
  13. **Haimes, Y. Y., Lasdon, L. S., & Wismer, D. A.** 1971. "On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization." *IEEE Transactions on Systems Man and Cybernetics*, vol. 1, p. 296-297.
  14. **Han, G., & Na, S.** 1999. "A Study on Torch Path Planning in Laser Cutting Processes Part 2: Cutting Path Optimization Using Simulated Annealing." *Journal of Manufacturing Processes*, 1, 62-70.
  15. **Imahori, S., Kushiya, M., Nakashima, T., & Sugihara, K.** 2008. "Generation of Cutter Paths" *Processing Technology*, vol. 206 (1-3), p. 453-461.
  16. **Lacomme, P., Prins, C., & Sevaux, M.** 2006. "A Genetic Algorithm for a Bi-objective Capacitated Arc Routing Problem." *Computers & Operations Research*, vol. 33, p. 3473-3493.
  17. **Malandraki, C., & Daskin, M. S.** 1993. "The Maximum Benefit Chinese Postman Problem and the Maximum Benefit Traveling Salesman Problem." *European Journal of Operational Research*, vol. 65, p. 218-234.
  18. **Manber, U., & Israni, S.** 1984. "Pierce Point Minimization and Optimal Torch Path Determination in Flame Cutting." *Journal of Manufacturing Systems*, vol. 3, p.81-89.
  19. **Martello, S., & Toth, P.** 1990. "An Exact Algorithm for Large Unbounded Knapsack Problems." *Operations Research Letters*, vol. 9, p. 15-20.
  20. **Mei, Y., Tang, K., & Yao, X.** 2011. "Decomposition-Based Memetic Algorithm for Multiobjective Capacitated Arc Routing Problem." *IEEE Transactions on Evolutionary Computation*, vol. 15, p. 151-165.
  21. **Minieka, E.** 1979. "The Chinese Postman Problem For Mixed Networks." *Management Science*, vol. 25, p. 643-648.
  22. **Nobert, Y. & Picard, J.** 1991. "An Optimal Algorithm for the Mixed Chinese Postman Problem." *Centre de Recherche Sur Les Transports Publication*.
  23. **Papadimitriou, C. H.** 1976. "On the Complexity of Edge Traversing." *Journal of the ACM*, vol. 23, p. 544-554.
  24. **Prakash, S., Sharma, M. K., & Singh, A.** 2009. "A Heuristic For Multi-Objective Chinese Postman Problem." *Computers & Industrial Engineering Conference Proceedings*, p. 596-599.
  25. **Rodrigues, A. M., & Ferreira, J. S.** 2012. "Cutting path as a Rural Postman Problem: Solutions by memetic algorithms." *International Journal of Combinatorial Optimization Problems and Informatics*, vol. 3, p. 31-46.
  26. **Win, Z.** 1989. "On the Windy Postman Problem on Eulerian Graphs." *Mathematical Programming*, vol. 44, p.97-112.

## XI. ENDÜSTRİ VE İŞLETME MÜHENDİSLİĞİ KURULTAYI SONUÇ BİLDİRİSİ



Türkiye`de Endüstri Mühendisliği eğitimi 1969 yılında ODTÜ`de ve hemen akabinde de İTÜ`de başlamıştır. 2017 yılında endüstri ve işletme mühendisliği alanı için yurt genelinde 93 üniversitede 93`ü örgün, 12`si ikinci öğretim olmak

üzere toplam 105 Endüstri ve İşletme Mühendisliği Bölümü bulunmaktadır. Bunların 68`i devlet, 37`si vakıf üniversitesidir. Bu rakamlar, endüstri ve işletme mühendisliğinin birçok üniversitede kolayca açılabilen mühendislik bölümlerinden biri haline geldiğini göstermektedir. Kasım 2017 itibariyle odamıza kayıtlı endüstri ve işletme mühendisi sayısı 8.050`yi bulmuştur.

Endüstri ve işletme mühendislerinin imza yetkisi olması için iki uzmanlık alanı tanımlanmış ve Stratejik Planlama ve Yatırım Hizmetleri Yönetimi Mühendis Yetkilendirme yönetmelikleri Resmi Gazete`de yayımlanmıştır. Bunlardan ilki olan stratejik planlama uzmanlığı; ülkeyi yönetenlerin heveslerinin ve hayallerinin gerçekleştirilmesi için değil ülke olanakları ve kabiliyetleri ile dünyadaki fırsatlar ve tehditler göz önüne alınarak stratejik planlar hazırlanması ve bu plana ulaşmak için tüm ülke ekonomisinin ve sanayisinin ortak bir istikamete yönlendirilmesi amacıyla yöneliktir.

Yatırım Hizmetleri Yönetimi ise mevcut kaynakları en uygun ve en etkili şekilde kullanmamız için gereken mesleki bilgi birikiminin ülke ve toplum yararına kullanılmasına yöneliktir. Ancak bu sayede ilerlemek ve halkımız için fark yaratacak değerleri ortaya çıkartmak mümkün olabilecektir.

Kurultayımızda gelişen ve değişen teknolojiye uyumlu, insanı tüm üretim ve hizmet sektörlerinin odağında tutabilecek bir vizyonun endüstri ve işletme mühendisliği açısından önemi üzerine değerlendirmeler yapılmış ve üniversiteler ile sanayi arasında işbirliği çerçevesinde kamusal değerlerle gerçekleştirilmesinin önemine vurgu yapılmıştır.

Kurultayımız;

- Endüstriyel dönüşümde endüstri işletme mühendisleri olarak sistemin insan odağını ve sistem çevre ile uyumunu optimum düzeyde değerlendirmek,
- Teknolojik ilerlemenin zorunluluğu ile oluşacak yararları, meslektaşımızın büyük fotoğrafının neresinde olduğunu ve ilerleyen süreçte nerede olacağını öngörmek,
- Stratejik Planlama ve Yatırım Hizmetleri uzmanlık alanlarının, belgelendirmeye ilişkin yönünün endüstriyel dönüşüme adaptasyonu ve uygulanabilirliğini incelemek,
- Teknolojik ilerlemede toplumsal yararların önemsenmesini sağlamak, şirketlere ve kamu tarafından içselleşmesine katkı sağlamak,
- Üniversitelerin eğitim programlarına endüstriyel dönüşüm ve toplumsal kazanımları içeren derslerin dahil edilmesini sağlamak,
- Öğrencilere TMMOB geleneği, Makina Mühendisleri Odası bakış açısı, öğrenci üyelik ve örgütlü yapının gerekliliğinin önemini aktarabilmek, vb. değerli amaçların dile getirilmesi yanı sıra günümüzde yaygın olarak konuşulan dijital dönüşüm/Endüstri 4.0`ın yalnızca üretim ve hizmet sunma açısından değil, kamusal-toplumsal yarar ve insan odaklı bir yaklaşım ile birlikte ele alınması gerektiği vurgulanmıştır. ◀◀

**TMMOB Makina Mühendisleri Odası**



## ENDÜSTRİ MÜHENDİSLİĞİ DERGİSİ YAYIN POLİTİKASI

Endüstri Mühendisliği Dergisi, TMMOB-MMO tarafından üç ayda bir yayımlanan süreli ve hakemli bir yayındır.

### Hedef Okuyucu Kitlesi

Endüstri Mühendisliği (EM) ve Yöneylem Araştırması (YA) konularında araştırma yapan, eğitim veren, eğitim gören ve bu alanlarda çalışanlardan oluşur.

### Yayın Amaçları

EM ve YA alanlarındaki gelişmeler, çalışmalar ve araçlarla ilgili akademik nitelikli yayın yapar.

EM ve YA alanlarındaki başarılı uygulamaların yaygınlaştırılması ve deneyimlerin paylaşılması için yayın yapar.

Meslek ile ilgili görüşlerin aktarılmasını ve tartışılmasını sağlar. EM ve YA alanlarında ortak bir dilin oluşmasına katkıda bulunur.

### Yayın İlkeleri

EM Dergisi, Yayın Kurulu (YK) tarafından yayına hazırlanır. YK yazıların seçimini hakem görüşlerini alarak yapar.

Yazarlara, okurlara ve kurumlara tarafsız yaklaşır.

Konu zenginliğinin korunup geliştirilmesini teşvik eder.

İçerik, dil ve biçim açısından nitelikli yayın yapar. Yayın dili Türkçe ve İngilizce'dir.

Yazının EM Dergisine gönderilmesi,

- yazının herhangi bir yayın organında yayımlanmamış olduğunu,
  - EM Dergisindeki değerlendirme süreci boyunca başka bir yayın organının değerlendirme sürecinde yer almayacağını,
  - yazı kabul edildiğinde yazının basım haklarının EM Dergisine geçtiğini ve başka bir dilde ve/veya ortamda, yayıncının onayı olmaksızın yayımlanamayacağını
- gösterir.

### Yazı Türleri ve Değerlendirme

EM Dergisi, yayın amaçları ve ilkeleri doğrultusunda hedef okuyucu kitlesini ilgilendiren *Makale, Uygulama, Teknik Not, İletişim, Doktora Tez Özeti ve Ödül Almış Çalışma* gibi farklı türde yazılara yer verir.

Makale, literatüre katkı sağlayan özgün yazıdır.

Uygulama, mesleki pratiğe katkı sağlayan ve mesleki bir konuda tutarlı, rasyonel ve başarılı uygulamaları anlatan yazıdır.

Teknik Not, Makale'ye göre dar kapsamlı, literatüre katkı sağlayan özgün yazıdır.

İletişim, eğitime, mesleğin icrası ve uygulamalarına genel anlamda katkı sağlayan; mesleğe yönelik felsefi tartışmalar başlatma ve mesleğe yeni açılımlar kazandırma potansiyeli taşıyan yazıdır. Meslek ve alanla ilgili eser, kitap ve yazılımları tanıtan değerlendiren yazılar da bu kapsamdadır.

Doktora Tez Özeti, doktorasını son iki yıl içerisinde tamamlamış araştırmacıların doktora tez özeti'dir.

Ödül Almış Çalışma, (bilinen) bir ödül için jüri tarafından belirli ölçütlere göre değerlendirilmiş ve ödüle layık bulunmuş yazıdır.

Makale, Uygulama, Teknik Not ve İletişim yazıları EM Dergisi yayın amaçları ve ilkeleri ışığında YK tarafından ön değerlendirmeye alınır, hakemlik sürecinin başlatılmasına ya da yazının ret edilmesine karar verilir. Hakemlik sürecine alınan yazı en az iki hakem tarafından değerlendirilir. Bu süreçte adlar iki taraftan da gizlenir. YK, hakemlerin görüşleri doğrultusunda yazıyı kabul veya ret eder veya yazının revize edilmesini ister. Değerlendirme sırasında tüm haberleşme iletişim yazarı ile yapılır.

Doktora Tez Özeti ve Ödül Almış Çalışma türü yazılar YK tarafından değerlendirilir. Gerekirse hakem görüşü alınır.

Ayrıca, EM Dergisinde tanıtım yazısı, haber, söyleşi, anı ve çeviri gibi farklı yazı türleri YK değerlendirmesi ile yayımlanabilir.

### Yazı Gönderme

EM Dergisi Yazı Kuralları'na uygun bir şekilde yazılmış yazılar, elektronik ortamda <http://omys.mmo.org.tr/endustri/> adresinden gönderilir. İletişim yazısının e-posta ve posta adresleri, faks ve telefon numaraları açıkça belirtilmelidir.

## JOURNAL OF INDUSTRIAL ENGINEERING EDITORIAL POLICY

Journal of Industrial Engineering (EMD) is a refereed periodical which is published quarterly by TMMOB-MMO (Turkish Chamber of Mechanical Engineers).

### Target Audience

The targeted audience of the journal comprises researchers, educators and practitioners in the fields of Industrial Engineering (IE) and Operations Research (OR).

### Objectives of Publication

It publishes academic manuscripts on the developments, processes, and tools in the fields of IE and OR.

It publishes for the purpose of extending the successful practices in IE and OR and enabling the sharing of experiences.

It provides a ground to transfer different views on the profession and discuss these viewpoints.

It promotes the formation of a common professional language in the fields of IE and OR.

### Principles of Publication

EMD is prepared for publication by the Editorial Board. The Editorial Board selects the material to be published by consulting the referees.

It holds an objective attitude towards authors, readers, and institutions.

It ensures and encourages variety in topics.

It publishes manuscripts which are qualified in terms of content, language and form.

Publication language is Turkish and English.

The fact that a manuscript is sent to EMD indicates that:

- The relevant manuscript has not been published previously in another journal.
- It will not be under the editorial evaluation of another journal as long the evaluation process in EMD continues.
- Once it has been approved for publication, EMD acquires the right to publish the manuscript and the manuscript cannot be published in a different language or domain without the approval of the publisher.

### Types of Publication and Evaluation

In accordance with its publication objectives and principles, EMD gives place to a diversity of studies that are of interest to its readers such as manuscripts, applications, technical notes, communication articles, dissertation abstracts, and works which have received an award.

A manuscript is an original work which contributes to the relevant literature.

An application is an article that describes the consistent, rational and successful applications related with a professional topic, and thus, contributes to the practice of the profession.

A technical note is an original article which contributes to the relevant literature but which is limited in scope compared to a manuscript.

A communication article is an article which contributes to the practice and applications of the profession and which has a potential to initiate philosophical discussions and bring in new developments regarding the profession. Reviews of an article, a book or software related with the field are treated in this category.

A dissertation abstracts is the summary of the dissertations of the researchers who completed their PhD within last 2 years.

A prize-awarded work is an article which has been evaluated according to certain criteria by a jury and deemed worthy for a prize (that is acknowledged).

Manuscripts, applications, technical notes, and communication articles are first taken under pre-evaluation by the Editorial Board in accordance with the EMD objectives and principles of publication and a decision is made whether to initiate the process of referee evaluation or to reject the work. In the process of referee evaluation, the work is evaluated by at least two referees. The names of the both parties are kept anonymous in this process. The Editorial Board approves or rejects the articles in accordance with the comments of the referees or it asks for further revision of the articles. Throughout the evaluation process, all the communication is carried out with the contact author.

Dissertation summary and prize-awarded articles are evaluated by the Editorial Board. If needed, referee opinion can be asked.

In addition, works as diverse as reviews, news, interviews, and memoirs can be published in EMD as long as they are evaluated by the Editorial Board.

### Manuscript Submission

The manuscripts complying with the norms of publication in EMD are sent electronically to <http://omys.mmo.org.tr/endustri/>. E-mail and postal addresses and fax and telephone numbers of the contact author should be clearly stated.