T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BÜYÜK ÇAPLI VERİ ZARFLAMA ANALİZİ MODELLERİ İÇİN YENİ YAKLAŞIMLAR

Hazırlayan
Gazi Bilal YILDIZ

Danışman
Doç. Dr. Banu SOYLU

Yüksek Lisans Tezi

Temmuz 2013
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BÜYÜK ÇAPLI VERİ ZARFLAMA ANALİZİ MODELLERİ İÇİN YENİ YAKLAŞIMLAR

(Yüksek Lisans Tezi)

Hazırlayan
Gazi Bilal YILDIZ

Danışman
Doç. Dr. Banu SOYLU

Bu çalışma; Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından FBY-12-3926 kodlu proje ile desteklenmiştir.

Temmuz 2013
KAYSERİ

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



Gazi Bilal YILDIZ

"Büyük Çaplı Veri Zarflama Analizi Modelleri İçin Yeni Yaklaşımlar" adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Gazi Bilal YILDIZ

Danışman

Doç. Dr. Banu SOYLU

Doç. Dr. İbrahim DOĞAN

Doç. Dr. Banu SOYLU danışmanlığında **Gazi Bilal YILDIZ** tarafından hazırlanan **"Büyük Çaplı Veri Zarflama Analizi Modelleri İçin Yeni Yaklaşımlar"** adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

23 / 07 / 2013

JÜRİ:

Danışman : Doç. Dr. Banu SOYLU B. Soylu

Üye : Doç. Dr. İbrahim DOĞAN

Üye : Yrd. Doç. Dr. Erdal CANIYILMAZ

ONAY:
Bu tezin kabulü Enstitü Yönetim Kurulunun 23/08/2013 tarih ve 2013/37-05 sayılı kararı ile onaylanmıştır.

23/08/2013

Prof. Dr. Kazım KEŞLİOĞLU

Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, yakın ilgi ve yardımlarını esirgemeyen ve bu günlere gelmemde en büyük katkı sahibi sayın hocam Doç. Dr. Banu SOYLU'ya teşekkürü bir borç bilirim.

Bu tez çalışmasına maddi destek veren Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne (Proje No: FBY-12-3926) teşekkür ederim.

Gazi Bilal YILDIZ

Kayseri, Temmuz 2013

Büyük Çaplı Veri Zarflama Analizi Modelleri İçin Yeni Yaklaşımlar

Gazi Bilal YILDIZ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Temmuz 2013

Danışman: Doç. Dr. Banu SOYLU

KISA ÖZET

Bir veri zarflama modelinde Karar Birimi sayısı ve Girdi/Çıktı faktörü sayısı arttıkça etkin Karar Birimlerini tespit etmek zorlaşmaktadır. Bu çalışmada etkin olmayan Karar Birimlerinin etkinliklerini tahmin eden yeni bir algoritma geliştirilmiştir. İlk olarak orijinal problem alt problemlere bölünmüş ve bu alt problemler çözülerek ilk etkinler kümesi elde edilmiştir. Daha sonra bu küme ideal KB'ye göre potansiyel etkin olan KB'lerle genişletilmiştir. Bu küme diğer KB'lerin etkinlik skorlarını tahminlemede kullanılacaktır. Büyük veri setleriyle önerilen algoritma çalıştırılmış ve performansı Klasik VZA modeliyle ve literatürdeki iki algoritmayla karşılaştırılmış ve sonuçları sunulmuştur.

Anahtar Kelimeler: Çok Kriterli Karar Verme, VZA, Parçalama Yaklaşımı.

NEW APPROACHES FOR LARGE SCALE DEA DATA SETS

Gazi Bilal YILDIZ

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, July 2013

Supervisor: Assoc. Prof. Dr. Banu SOYLU

ABSTRACT

When there are huge number of Decision Making Units (DMUs) and large number of input/output, identifying efficient units via a DEA model could be computationally complex. In this study, we propose a new algorithm to determine efficient DMUs and to estimate efficiency scores of inefficient units. Initially, we partition the original problem into sub-problems and solve these sub-problems to construct a set of initial efficient units. Then this initial set is enlarged by adding some potentially efficient DMUs according to an ideal DMU. This set leads us to estimate the efficiency scores of other units. We apply the algorithm over large data sets and compare the performance with the classical DEA algorithm and two algorithms from the literature. We present our results.

Keywords: Multi criteria decision making, DEA, Decomposition.

KISALTMALAR VE SİMGELER

LP:	Doğrusal Programlama
VZA:	Veri Zarflama Analizi
KB:	Karar Birimi
CCR:	Charnes Cooper Rhodes VZA Yöntemi
BCC:	Banker Charnes Cooper VZA Yöntemi
θ:	Etkinlik Skoru
i:	çıktı faktörü indisi i=1,2,…,p
j:	girdi faktörü indisi j=1,2,…,m
k:	KB indisi k=1,2,…,n
o:	Değerlendirilen KB indisi
m:	girdi faktörü sayısı
p:	çıktı faktörü sayısı
u_j:	j. girdi faktörü ağırlığı
v_i:	i. çıktı faktörü ağırlığı
x_{jk}:	k. KB'nin j. girdi faktörü değeri
y_{ik}:	k. KB'nin i. çıktı faktörü değeri
SP_j:	j. alt problem
d:	Veri Setindeki Etkin Oranı
CPU:	Merkezi İşlemci Birimi
q:	Alt Problem Sayısı
ne:	KE Kümesi Genişletme Oranı

İÇİNDEKİLER

BİLİMSEL ETİĞE UYGUNLUK ..... **Hata! Yer işareti tanımlanmamış.**
YÖNERGEYE UYGUNLUK ..... **Hata! Yer işareti tanımlanmamış.**
TEŞEKKÜR iv
KISA ÖZET v
ABSTRACT vi
KISALTMALAR VE SİMGELER vii
İÇİNDEKİLER viii
TABLOLAR LİSTESİ ix
ŞEKİLLER LİSTESİ x
GİRİŞ 1
I. BÖLÜM 3
LİTERATÜR ARAŞTIRMASI 3
II. BÖLÜM 6
2.1 VZA Modelleri 6
2.2 Teorik Altyapı 9
2.3 Temel Alınan Algoritmalar 12
2.3.1 Parçalama Algoritması 12
2.3.2 BuildHull Algoritması 13
III. BÖLÜM 14
ÖNERİLEN MODEL 14
IV. BÖLÜM 24
DENEYSEL ÇALIŞMA VE SONUÇLARI 24
V. BÖLÜM 43
SONUÇ 43
KAYNAKÇA 44
ÖZGEÇMİŞ 46

TABLOLAR LİSTESİ

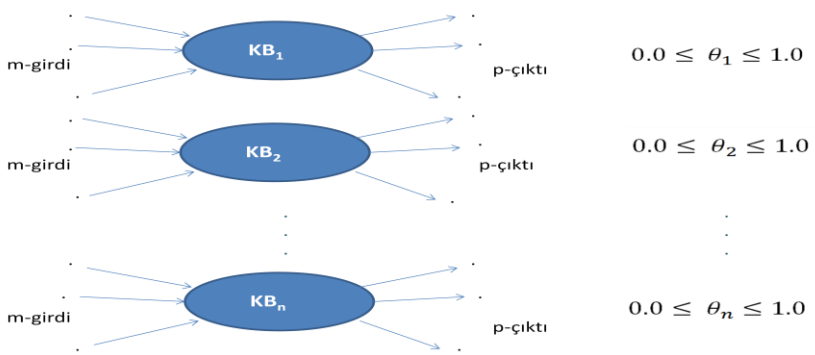
Tablo 1: Klasik VZA algoritması sonuçları: 26
Tablo 2: Classification algoritması sonuçları 28
Tablo 3: BuildHull algoritmasının sonuçları 31
Tablo 4: Önerilen algoritmanın sonuçları 34
Tablo 5: Önerilen algoritma için |KSuper|/|K| ve |E|/|KSuper| sonuçları 36
Tablo 6: Tüm algoritmaların sonuçları. 37
Tablo 7: Önerilen algoritmanın Classification algoritmasına göre iyileştirme yüzdesi .. 38
Tablo 8: Önerilen algoritmanın BuildHull algoritmasına göre iyileştirme yüzdesi. 40
Tablo 9: Önerilen algoritmanın Klasik VZA algoritmasına göre iyileştirme yüzdesi. ... 41

ŞEKİLLER LİSTESİ

Şekil 1:VZA elemanlarının temel gösterimi. 2
Şekil 2: KB'lerin temel alınan nokta ile oluşturdukları doğrunun eğimleri.....17
Şekil 3: Aşama 1'in süreci 18
Şekil 4: KSuper ve KE kümelerinin büyüklükleri arasındaki çelişki(tradeoff) 20
Şekil 5: Kümelerin şematik gösterimi. 24
Şekil 6: Klasik VZA algoritmasının sonuçları.. 28
Şekil 7: Classification algoritmasının sonuçları.....30
Şekil 8: BuildHull algoritmasının sonuçları.....33
Şekil 9: Önerilen algoritmanın sonuçları.. 36
Şekil 10: Algoritma sonuçlarının birbirleriyle karşılaştırılması.....43

GİRİŞ

Veri Zarflama Analizi (VZA), benzer girdi ve çıktılara sahip olan yani temelde benzer fonksiyonları olan birimlerin etkinliğini ölçmede kullanılan bir yöntemdir. Charnes, Cooper ve Rhodes [1] tarafından 1978 yılında geliştirilmiş ve bu tarihten itibaren VZA konusunda çok farklı bakış açılarıyla binlerce araştırma yapılmıştır. VZA'yı diğer yöntemlerden ayıran temel özellik, değerlendirme için kullanılan girdi ve çıktılara uygun ağırlıkların bir matematiksel model tarafından atanmasıdır ve bu ağırlıklar her Karar Birimi (KB) için özeldir. Bu ağırlıklar sayesinde KB, alabileceği en yüksek etkinlik skorunu elde eder. KB'ler girdi ve çıktı sayısı kadar boyutlu bir uzayda yerleştirilirse, bütün KB noktalarının tam üstünde ya da altında yerleştiği bir sınır oluşturulabilir. Bu sınıra bütün noktaları kapsadığı için zarf denir. Bu zarfı oluşturmanın çeşitli yöntemleri vardır. Örneğin, CCR, BCC, Additive… vs. Zarfın üst sınırına yerleşmiş KB'ler etkin (yani etkinlik skoru Q* = 1.0) olarak adlandırılır ve bütün KB'lerin etkinlikleri birbirlerinin pozisyonundan etkilendiği için hesaplanan skor aslında görecelidir. Model KB'lerini etkin olanlar ve etkin olmayanlar diye iki gruba ayırma eğilimindedir. Etkin olan KB'lerin etkinlik skoru 1.0 iken etkin olmayanların etkinlik skoru <=1.0 şeklindedir ve etkinlik skorunun 0'a yaklaşması düşük bir performansın belirtisidir. Şekil 1'de KB'ler, girdiler, çıktılar gösterilmiştir:



Şekil 1: VZA elemanlarının temel gösterimi.

VZA'nın birçok alanda başarılı uygulamaları mevcuttur. Bu alanlara örnek olarak hastaneler, eğitim kurumları, üretim tesisleri, bankalar, oteller, restoranlar, toptancı mağazaları, silahlı kuvvetler vb. kuruluşların KB lerinin etkinliklerinin ölçülmesi, uzay araştırmaları, spor, yönetim performanslarının değerlendirilmesi, tedarikçi değerlendirme problemleri, şehirlerin karşılaştırılması, kamu kurumlarının etkinliklerinin değerlendirilmesi verilebilir. Bütün bu örneklerde ana perspektif genellikle makro seviyededir. Fakat, birimler mikro seviyede de tanımlanabilir. Örneğin, bir banka/hastane şubesindeki personel, bir eğitim enstitüsündeki öğrenciler bir ülkenin iş sektörleri…vs. Şu açıktır ki; mikro seviyedeki KB sayısı büyük olabilir ve performans ölçümlerinin periyodik olarak yapılması gerekli olabilir (örneğin haftalık, aylık, yıllık…vs.).

Bu çalışmada Korhonen ve Siitari'nin [13] önerdiği parçalama algoritmasını geliştirerek büyük çaplı DEA modellerinin daha hızlı ve etkin bir çözümü için bir yaklaşım önerilmiştir. Önerilen model üç aşamadan oluşmaktadır. İlk adımda veri seti boyutsal olarak parçalanır ve tüm ikili kombinasyonlar için basit bir eğim hesabıyla etkin olduğuna emin olunan bazı KB'ler tespit edilerek bir kümede toplanır. Daha sonra veri setine göre bir *'ideal KB'* belirlenir. Diğer KB'lerin bu ideal KB'ye uzaklıklarına göre tahmini bir etkinlik skoru hesaplanır. İdeal KB'ye yakın olan KB'lerin etkin olma potansiyeli yüksektir. En yakınlardan bir grup KB seçilerek ve bir önceki adımdaki etkin KB'ler de eklenerek yeni bir '*Süper Küme'* oluşturulur. İkinci adımda, bu süper küme kullanılarak küme dışında kalan KB'ler değerlendirilir. Etkinlik skoru <1.0 olan KB'lerin etkin olmadıkları kesindir. Diğer taraftan etkinlik skoru = 1.0 olan KB'lerin ana modelde etkin olma ihtimalleri olması sebebiyle süper kümeye dahil edilirler. Son adımda Süper küme klasik yöntemle değerlendirilir. Bu yaklaşımın diğer avantajı etkin olmayan KB'lerin etkinlik skorlarını da çok küçük bir hata ile tahmin edebilmesidir.

Çalışmanın birinci bölümünde literatür araştırması verilmiştir. İkinci bölümde problemin tanımı yapılmış ve kullanılan notasyon hakkında bilgi verilmiştir. Üçüncü bölümde ise bu çalışmada önerilen model anlatılmıştır. Üçüncü bölümde veri setleriyle denemeler yapılmış ve modelin çıktıları literatürde bulunan iki algoritmayla ve klasik VZA modeli çıktılarıyla karşılaştırılmıştır. Dördüncü bölümde önerilen yaklaşımın sonuçları tartışılmıştır.

1. BÖLÜM
LİTERATÜR ARAŞTIRMASI

Literatürde pek çok VZA çalışması mevcuttur. Cook and Seiford [2] çalışmalarında yaklaşık 30 yıllık bir literatür incelemesi çalışması yapmışlar ve bu zaman zarfı içinde yapılan ve özellikle metodolojiye katkıda bulunan çalışmalara öncelik vermişlerdir ve bunu yaparken özellikle etkinlik ölçümü yapan, katsayılara kısıtlar dahil eden, değişkenlerin durumunu hesaba katan ve çeşitli veri setlerinin modellemesini yapan çeşitli modellere ehemmiyet vermişlerdir.

Liu vd. [3] Web of Science endeksli dergilerde yayımlanan VZA uygulamaların inceleyen bir araştırma yapmışlar ve yapılan çalışmaların 3'te 2'sinin verilerinin gerçek hayat problemine ait olduğu kalan kısmın metodolojik çalışmalar olduğu sonucuna ulaşmışlardır.

Liu vd. [4] VZA için merkezi önemde olan ve VZA'nın son zamanlarda aktif olduğu alt konuları ihtiva eden makaleleri bulmayı amaçlayan bir literetür incelemesi çalışması yapmışlardır ve bir takım istatistiksel sonuç çıkarmışlardır.

Emrouznejad vd. [5] başlangıçtan 2007'ye kadar yapılmış VZA gerçek-hayat uygulamalarını ve VZA'nın teorik gelişimini ele alan genişçe bir literatür araştırması çalışması yapmışlardır. Ayrıca bu çalışma; anahtar kelimelerle ilgili yaptıkları analizleri, kullanışlı bazı dergileri ve bir takım istatistikleri ihtiva etmektedir.

Klasik VZA modellerinde her bir kriterin girdi ya da çıktı olduğu kesin olarak belirlidir. Cook ve Zhu [6] geliştirdikleri bir modelle hem girdi hem de çıktı olabilecek kriterler için bir esneklik getirmişlerdir. Bu tip kriterlere esnek kriterler adını vermişler ve klasik VZA modelini esnek kriterleri ihtiva edecek şekilde modifiye ederek yeni bir model geliştirmişlerdir. Daha sonra Cook vd. [7] tarafından yapılan çalışmada, ara kriter

olarak adlandırılan kriterler, önce çıktı olarak daha sonra girdi olarak değerlendirilecek şekilde iki aşamalı bir model önerilmiştir.

Emrouznejad ve Shale [8] girdi katmanı, ara katman ve çıktı katmanı olmak üzere üç katmana sahip olan bir Yapay Sinir Ağı modeli geliştirmişler etkinlik değerlerini tahmin etmişlerdir. Veri setinden rastgele bir veri kümesini öğrenme, test ve doğrulama aşamalarından geçirerek modeli oluşturmuş daha sonra tüm KB'leri için etkinlik değerlerini hesaplamışlardır.

Sütün ekleme metodu, tam sayılı programlama problemlerinde büyük çaplı optimizasyon modellerinin çözümü için önemli bir araçtır. Tam sayılı programlama problemleri için temel çözüm teknikleri: dal-sınır tekniği ve kesme düzlemi tekniğidir. Ehrgott ve Tind [9] tam sayılı programlama problemlerinin çözümü için sütun ekleme metodunu ve dal-sınır tekniğini önermişlerdir. VZA modelinde optimal sonuca ulaşmak için bu metotla yapay KB'ler oluşturulmuş ve branch and bound algoritmasıyla dual problemin katsayıları hesaplanmıştır.

Günümüzde yapılan performans ölçümlerinde gittikçe daha fazla KB'nin ve girdi/çıktı verilerinin değerlendirilmesi gerekmektedir. Örneğin bir uygulama çalışmasında Chen vd. [10] Tayvan'da 19 belediye çöp yakma makinesi için farklı operasyonel durumlarının performanslarını göstermek için veri zarflama analizi metodunu kullanmışlardır. Bu çöp yakma makinelerinin operasyonel etkinliklerinin olasılık dağılımının hesabı için Monte Carlo simülasyonu kullanılmış ve 4 yıllık (2002-2005) veri seti kullanılarak oluşturulan büyük ölçekli modelin çözümü için veri zarflama analizi kullanılmıştır.

Büyük çaplı, girdi/çıktı ve KB sayısı çok fazla olan, veri setlerini çözmek model LP olsa bile oldukça zaman alıcıdır. Bunun temel sebebi her bir KB için bir LP çözülmesinin gerekmesidir. Örneğin 20 girdi/çıktı ve 100,000 KB'nin olduğu bir modeli çözmek yaklaşık olarak 6-7 saat sürmektedir [11]. Dula [12] büyük data setlerinin ölçüm süresi performansını inceleyen bir çalışma yapmıştır. Bu çalışmada büyük çaplı VZA modelleri için sentetik veri setleri oluşturulmuş ve standart veri zarflama analizi modeli çeşitli solver özellikleri ile bu veri seti üzerinde kapsamlı bir şekilde analiz edilmiştir. Bu sürelere bakıldığında hızlandırıcı algoritmalara ihtiyacın olduğu

görülmektedir. Literatürde büyük çaplı veri setleri üzerine yapılmış VZA uygulamaları oldukça yenidir.

Korhonen ve Siitari [13] etkin KB'lerinin belirlenmesinde işlem yükünü azaltmak için bir parçalama algoritması önermişlerdir. Farklı kombinasyonlarda girdi ve çıktı kriterlerinden oluşan daha küçük çaplı alt-problemler oluşturmuşlardır. Alt problemlerden elde edilen etkin KB'ler ana problem için de etkindir. Bu KB'leri bir kümeye eklemişlerdir. Ayrıca ana problemde etkin olma potansiyeline sahip KB'leri de dâhil ederek daha küçük bir model oluşturmuşlardır. Ana modele ait sonuca daha küçük modeller çözerek daha hızlı ve kolay bir şekilde ulaşmışlardır.

Chen ve Cho [14] büyük çaplı VZA modellerini daha hızlı çözebilmek için bir hızlandırma mekanizması geliştirmişlerdir. Buna göre referans olarak seçtikleri bir KB'nin α komşuluğundaki KB'leri bulmuşlar ve bu alt kümenin etkinlerini BCC modeli [15] çözerek belirlemişlerdir. Daha sonra bu etkinlerin bütün küme içinde etkin olup olmadığını Karush-Kuhn-Tucker (KKT) koşullarını kontrol ederek belirlemişlerdir.

Dula [11] VZA temelli iki aşamalı bir model geliştirmiş ve bu modelin ilk aşamasında uç noktalardaki KB'leri ihtiva eden bir zarf oluşturmuştur. Tüm KB'lerini etkinler ya da etkin olmayanlar şeklinde tasnif ettikten sonra etkin olmayanların etkinliğini bu zarfı kullanarak hesaplamıştır.

Fukuyama and Sekitani [16] çalışmalarında KB'lerini farklı kombinasyonlarda kullanarak konveks yüzler elde etmek suretiyle etkinlik çizgisini parçalamışlar ve geliştirdikleri modeli çeşitli gerçek hayat problemlerine ait veri setleriyle çalıştırarak literatüre katkısını göstermişlerdir.

Dikey boyutta parçalama adına da Pendharkar ve Troutt [17] iki gruplu bir sınıflandırma problemi için uç değerlere sahip örnekleri elimine ederek veri setini dikey olarak azaltacak bir yöntem önermişlerdir. İki aşamadan oluşan modelin ilk aşamasında uç değerlere sahip olan örnekler gruplandırılmış ve örtüşen örnekler belirlenmiştir. İkinci aşamada örtüşen örneklerin sınıflarını belirlemek üzere yanlış sınıflandırma maliyetinin minimizasyonunu amaçlayan bir model çözülmüştür. Dikey boyut azaltma problemi olarak adlandırılan bu problemin çözümünde veri zarflama analizi metodu kullanılmıştır.

2. BÖLÜM
PROBLEM TANIMI

2.1 VZA Modelleri

Bu bölümde VZA modeli gözden geçirilmiş, teorik alt yapısı incelenmiştir. 1 girdi ve 1 çıktısı olan problemler için genel olarak $\frac{çıktı}{girdi}$ oranı olarak bilinen verimlik hesabını yapmak kolaydır. Fakat daha fazla kriter varsa girdi ve çıktıların kriter ağırlıkları dikkate alınarak yeniden tanımlanması gereklidir. Literatürde CCR olarak bilinen model aşağıda verilmiştir.

Primal Model:

Notasyon:

i: çıktı faktörü indisi i=1,2,…,p

j: girdi faktörü indisi j=1,2,…,m

k: KB indisi k=1,2,…,n

o: Değerlendirilen KB indisi

m: girdi faktörü sayısı

p: çıktı faktörü sayısı

u_j: j. girdi faktörü ağırlığı

v_i: i. çıktı faktörü ağırlığı

x_{jk}: k. KB'nin j. girdi faktörü değeri

y_{ik}: k. KB'nin i. çıktı faktörü değeri

Amaç Fonksiyonu:

P_0^{CCR}: Max. $z_0 = \frac{\sum_{i=1}^{p} v_i * y_{i,0}}{\sum_{j=1}^{m} u_j * x_{j,0}}$

Kısıtlar:

$\frac{\sum_{i=1}^{p} v_i * y_{i,k}}{\sum_{j=1}^{m} u_j * x_{j,k}} <= 1.0$ ∀k (1)

$u_j, v_i \geq 0$ ∀i, ∀j (2)

Amaç fonksiyonu, basit verimlilik hesabı olan $\frac{Sanal\ Çıktı}{Sanal\ Girdi}$ oranını maksimize etmektedir. Burada ağırlıklandırılmış girdi ve çıktı sanal olarak ifade edilmiştir. (1) numaralı kısıt ise tüm KB'ler için etkinlik skoru 1.0'den büyük olmayacak şekilde $\frac{Sanal\ Çıktı}{Sanal\ Girdi}$ oranını sınırlamaktadır. (2) numaralı kısıt ise pozitif ağırlık kısıtıdır. Primal model bu haliyle doğrusal değildir ancak kolayca doğrusallaştırılabilir. Bunun için amaç fonksiyonunu paydası 1 olacak şekilde kısıtlara eklenir ve de (1) numaralı kısıt içler dışlar çarpımı yapılarak yeniden düzenlenir. Doğrusal model şu şekildedir:

Amaç Fonksiyonu:

P_0^{CCR}: Max. $z_0 = \sum_{i=1}^{p} v_i * y_{i,0}$

Kısıtlar:

$\sum_{j=1}^{m} u_j * x_{j,0} = 1.0$ (3)

$\sum_{i=1}^{p} v_i * y_{i,k} - \sum_{j=1}^{m} u_j * x_{j,k} \leq 0$ ∀k (4)

$v_i, u_j \geq 0$ ∀i, ∀j (5)

a. Dual Model:

P_0 modelinin dualini aldığımızda değerlendirilmekte olan KB_0 'ın referans kümesi bilgisine ulaşabiliriz. Dolayısıyla λ_k ; KB_0 'ın referans aldığı KB_k 'yı referans alma oranını verir. Eğer üretim mümkünüyet kümesindeki elemanların pozitif kombinasyonundan KB_0'ı domine edecek bir yansıma noktası oluşturulabilirse, KB_0'ın

etkin olmadığı anlaşılır. Bu yansıma noktası ancak ve ancak kendisine eşit olursa KB_0 etkindir. Bu yansıma noktası KB_0'ın minimum değişiklik ile etkin olabileceği noktadır. Başka bir deyişle pozitif λ değerine sahip olan her KB BCC-etkin olarak adlandırılır. Not edilmelidir ki bu noktadan sonra "etkin" ifadesi ile "BCC-etkinliği" işaret edilmektedir.

Amaç Fonksiyonu:

D_0^{CCR}: Min. θ_0

Kısıtlar:

$\sum_{k=1}^{n} \lambda_k * y_{i,k} \geq y_0$ ∀i (6)

$\theta_0 * x_{j,0} - \sum_{k=1}^{n} \lambda_k * x_{j,k} \geq 0$ ∀j (7)

$\lambda_k \geq 0$ ∀k (8)

θ_0 $free$ (9)

Dual modelin amaç fonksiyonu, θ_0'ı en küçüklemektir. Eğer KB_0'ın en az bir girdi ya da çıktısı yansıma noktasının o girdi ya da çıktısına eşitse $\theta_0 = 1$ olur. (6) numaralı kısıt yansıma noktasının çıktılarının KB_0 çıktılarından büyük eşit olmasını sağlar. (7) numaralı kısıt ise θ_0'ın bütün girdi oranlarından büyük olmasını sağlar. (8) numaralı kısıt pozitif olma şartıdır. Bu modelde θ_0 etkinlik skorunu (6) numaralı kısıt altında yansıma noktasının girdilerinin KB_0 girdilerine olan oranının en büyüğü belirler. Bu sebeple *girdi-temelli* model olarak adlandırılır.

Literatürde çeşitli VZA modelleri mevcuttur. BCC model [1] en popüler olan modellerden birisidir. Bu çalışma boyunca girdi temelli BCC modeli kullanılmıştır. CCR modelinin kısıtlarına $\sum_{k=1}^{n} \lambda_k = 1$ kısıtı eklenirse model BCC modeline dönüştürülmüş olur. (9) nolu kısıta u_0 değişkeni atanırsa BCC modelinin duali şu şekilde olur.

Amaç Fonksiyonu:

P_0^{BCC}: Max. $h_0 = \sum_{i=1}^{p} v_i * y_{i,0} + \beta$

Kısıtlar:

$\sum_{j=1}^{m} u_j * x_{j,0} = 1.0$ (10)

$\sum_{i=1}^{p} v_i * y_{i,k} - \sum_{j=1}^{m} u_j * x_{j,k} + \beta \leq 0$ ∀k (11)

$v_i, u_j \geq 0$ ∀i, ∀j (12)

$\beta\ free$ (13)

2.2 Teorik Altyapı

VZA'da en önemli husus girdi ve çıktılar boyutunda bütün KB'leri kapsayacak bir zarfın oluşturulmasıdır. Oluşturulan zarf konveks olmalıdır. $(\mathbf{x}_k, \mathbf{y}_k)$ k. KB'nin girdi ve çıktı vektörleri olsun. Konveks zarf şu P kümesinden oluşur [18]:

$P = \{(\boldsymbol{x}, \boldsymbol{y}) |\ \boldsymbol{x} \geq \sum_{k=1}^{n} \lambda_k \boldsymbol{x}_k\ ,\ \boldsymbol{y} \leq \sum_{k=1}^{n} \lambda_k \boldsymbol{y}_k, \sum_{k=1}^{n} \lambda_k = 1,\ \lambda_1, \lambda_1, \dots, \lambda_n \geq 0\ \}$

Bu küme üretim mümküniyet kümesi olarak da bilinir. Girdi ve çıktıların aynı yönde değerlendirilmesi için Q kümesi tanımlanmıştır:

$Q = \{(-\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+p} | -\boldsymbol{x} \leq -\sum_{k=1}^{n} \lambda_k \boldsymbol{x}_k\ ,\ \boldsymbol{y} \leq \sum_{k=1}^{n} \lambda_k \boldsymbol{y}_k,\ \sum_{k=1}^{n} \lambda_k = 1,\ \lambda_1, \lambda_1, \dots, \lambda_n \geq 0\ \}$

Çalışmada kullanacağımız algoritmalar problemi $m + p$ boyutundan daha küçük boyutlara parçalayarak analiz etmektedir. Boyutu daha küçük olan bu problemler alt-problem olarak adlandırılmaktadır. Bu alt-problemler için $\hat{Q}$ kümesi şu şekilde tanımlanmır:

$$\hat{Q} = \{(-\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}) \in R_+^{\overline{m}+\overline{p}} | \overline{m} + \overline{p} < m + p, -\overline{\boldsymbol{x}} \leq -\sum_{k=1}^{n} \lambda_k \overline{\boldsymbol{x}}_k\ , \overline{\boldsymbol{y}} \leq \sum_{k=1}^{n} \lambda_k \overline{\boldsymbol{y}}_k,\ \sum_{k=1}^{n} \lambda_k = 1, \lambda_1, \lambda_1, \dots, \lambda_n \geq 0\ \}$$

Önerme 1: Eğer $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktası $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktasına tam baskınsa; $(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q}$ noktası $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasına tam baskındır.

İspat:

$(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ ve $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q} \mid \ni i, j \quad -x_j' = -x_j'' \,,\; y_i' = y_i''$

$(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ ve $(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q} \mid \ni i, j \quad -x_j^* = -x_j^{**} \,,\; y_i^* = y_i^{**}$ olsun.

$(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktasının $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktasına tam baskın olduğu varsayıldığında $-\boldsymbol{x}^* > -\boldsymbol{x}'$ ve $\boldsymbol{y}^* > \boldsymbol{y}'$ olur, dolayısıyla $-\boldsymbol{x}^{**} > -\boldsymbol{x}''$ ve $\boldsymbol{y}^{**} > \boldsymbol{y}''$ olur.

□

Teorem 1:

Eğer $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası Q'da etkin olmayan bir nokta ise, tüm $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktaları da $\bar{Q}$ 'da etkin olmayan noktalardır.

İspat:

Bknz. Önerme 1.

Teorem 2:

Eğer $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası $\bar{Q}$'da etkinse $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası da Q'da etkindir.

İspat:

$(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası etkin olmayan bir nokta olsun. O halde önerme 1'de verildiği gibi bir $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktası olmalıdır. Ama bu $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasının etkin olduğu varsayımıyla çelişir.

□

Teorem 3:

$(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası etkinse $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası etkin yada etkin olmayan bir nokta olabilir.

İspat:

Durum a: Eğer $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası etkinse etkin olmayan bir $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası olabilir.

$(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ ve $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktaları birbirlerini domine etmeyen noktalar olsun. Ama $(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q}$ noktası $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasına tam baskın olabilir.

Durum b: Benzer bir şekilde, $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ ve $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktaları birbirlerini domine etmeyen noktalar olsun. Ama $(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q}$ noktası $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasına baskın olmayabilir.

□

Teorem 4:

Eğer $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası $\bar{Q}$'da etkin olmayan bir nokta ise $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası Q'da etkin bir nokta ya da etkin olmayan bir nokta olabilir.

İspat:

Durum a: Eğer $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası $\bar{Q}$'da etkin olmayan bir nokta ise $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası Q'da etkin olmayan bir nokta olabilir.

$(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q}$ noktası, $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasına $\bar{Q}$'da tam baskın olan bir nokta olsun. O halde $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktası $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktasına Q'da baskın bir nokta olabilir.

Durum b: Eğer $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktası $\bar{Q}$'da etkin olmayan bir nokta ise $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktası Q'da etkin bir nokta olabilir.

Benzer şekilde, $(-\boldsymbol{x}^{**}, \boldsymbol{y}^{**}) \in \bar{Q}$ noktası, $(-\boldsymbol{x}'', \boldsymbol{y}'') \in \bar{Q}$ noktasına $\bar{Q}$'da tam baskın olan bir nokta olsun. O halde $(-\boldsymbol{x}^*, \boldsymbol{y}^*) \in Q$ noktası $(-\boldsymbol{x}', \boldsymbol{y}') \in Q$ noktasına Q'da baskın olmayan bir nokta olabilir.

□

2.3 Temel Alınan Algoritmalar

2.3.1 Parçalama Algoritması

Korhonen ve Siitari [13] boyut olarak $m + p$'den daha küçük q tane alt problem oluşturarak ve bu alt problemleri çözerek etkin çözümler kümesini bulmuşlardır ve büyük veri setlerinde denemeler yapmışlardır. Bu şekilde bir parçalamanın klasik VZA modeli ile karşılaştırıldığında daha etkin çalıştığını göstermişlerdir. Bu amaçla girdi ve çıktılardan farklı seçimler yaparak q tane alt problem oluşturmuşlardır.

SP_t t. alt problemi tanımlamaktadır. K, tüm KB'leri içeren küme; KE, potansiyel etkin KB'lerden bazılarını içeren küme; KE_i, i. alt problemin etkin KB'lerini içeren küme; E, tüm etkin KB'leri içeren küme ve KSuper, tüm potansiyel etkin KB'leri içeren kümeler olsun. Yani $E \subseteq KSuper$ ve $KE_i \subseteq KE \subseteq KSuper$ 'dir.

Buna göre her bir alt problemi çözmüşler ve alt problemden elde ettikleri etkin KB'leri KE kümesine eklemişlerdir. Daha sonra potansiyel etkin KB'leri tespit etmek için $N - KE$ kümesinin tüm elemanları için kısıtları KE kümesinin elemanlarından oluşan lexicographic BCC modelini çözmüşler ve potansiyel etkin KB'leri $KSuper$ kümesine eklemişlerdir. Son adımda $KSuper$ kümesini çözerek etkin KB'leri E kümesine eklemişlerdir. Sonuç olarak tüm etkinleri ihtiva eden E kümesini elde etmişlerdir. Algoritma şu şekildedir:

Adım 0: Problemi Parçalama.

$KE = \emptyset,\ KSuper = \emptyset,\ E = \emptyset$

Parçalama Algoritması kullanılarak q adet boyutsal olarak parçalanmış alt problemin elde edilmesi

Adım 1: Alt problemlerin çözümü.

For i = 1 : q

$KE_i \leftarrow$ Solve alt problem SP_i

$KE \leftarrow KE \cup KE_i$

END

Adım 2: $N - KE$ kümesinden KE kümesinin elemanlarına ait kısıtlar kullanılarak potansiyel etkin KB'lerin bulunup $KSuper$ kümesine eklenmesi.

Ayrıca, $KSuper$ kümesi aşağıdaki gibi genişletilir:

$$KSuper \leftarrow KSuper \cup KE$$

Adım 3: $KSuper$ kümesindeki etkin KB'lerin bulunması ve E kümesine eklenmesi.

Return E.

2.3.2 BuildHull Algoritması

Dula [11], geliştirdiği algoritmayla VZA modelini iki aşamadan oluşan yeni bir yaklaşımla çözmüştür. Bunun için ilk olarak uç noktaları seçerek bir zarf oluşturmuş ardından zarf dışında kalan tüm KB'ler için LP çözerek zarfı her bir adımda güncellemiş ve sonuçta etkin KB'lerin oluşturduğu bir zarf elde etmiştir. Algoritma şu şekildedir:

Adım 0: K kümesinden bir adet uç KB'nin bulunması.

Adım 1:

While $K \neq \emptyset$

Select a DMU_k and $(h_k^*, \beta^*, v^*) \leftarrow$ solve P_k^{BCC}

If amaç değeri $h_k^* > 0$, zarf genişletilebilir. O halde optimum sonuç kullanılarak en büyük genişletme yapılır

$k^* = \text{argmax}_{\forall k \in K}\{\sum_{i=1}^{p} v_i^* y_{ik} + \beta^*\}$

$K \leftarrow K \backslash k^*$; $E \leftarrow E \cup k^*$

Else DMU_k ise zarf iyileştirilmez.

$K \leftarrow K \backslash k$

END

Adım 2: $K - E$ kümesinin elemanlarının etkinlik skoru E kümesinin elemanlarına ait kısıtlar kullanılarak hesaplanır.

3. BÖLÜM
ÖNERİLEN MODEL

Bu çalışmada önerilen model Korhonen ve Siitari'nin [13] çalışması baz alınarak iyileştirilmiştir. Dolayısıyla dikey boyutta parçalama yani girdi ve çıktılardan alt kümeler oluşturarak orijinal büyük modeli parçalama temelli bir yaklaşımdır. Problemin bu şekilde alt kümelere ayrılarak değerlendirilmesi problemi kolaylaştıracaktır. Ancak sadece alt problemleri çözerek bütün etkin KB'leri bulmak mümkün olmayabilir. Alt problemlerin çözümünden elde edilen etkin KB'ler daha sonraki aşamalarda VZA modeline girdi olarak verilir. Böylece VZA modelinin çözüm süresi düşürülebilir. Bu çalışmada önerilen model 3 aşamadan oluşmaktadır. İlk adımda 1 girdi ve 1 çıktısı olan problemler çözülerek KE kümesi oluşturulmuştur. İkinci adımda bazı potansiyel etkin KB'ler KE kümesine eklenerek KE kümesi genişletilmiştir. Bu amaçla KE kümesi dışında kalan KB'ler için LP çözülmüştür. Son adımda ise tüm potansiyel etkin KB'leri barındıran $KSuper$ kümesinden etkin olan KB'ler belirlenmiştir. İkinci ve üçüncü adımlarda çözülen LP'lerin sonuçları etkin olmayan KB'lerin etkinlik skorlarının tahmini için kullanılmıştır.

I. Aşama: *1girdi – 1 çıktı kombinasyonlarından ilk etkin kümenin oluşturulması*

m tane girdi ve p tane çıktısı olan bir kümeden 1 girdi (j') – 1 çıktı (i') ihtiva eden $m\ x\ p$ tane alt küme oluşturulur. Bu aşamada her KB'nin 1 girdi – 1 çıktı modelinde etkin olup olmadığı araştırılır ve bu işlem her KB için $m\ x\ p$ kere tekrar edilmelidir. Yani bu işlem için toplamda $m\ x\ p\ x\ n$ tane 3 kısıtlı ve n+1 değişkenli LP çözülmesi gerekir. Ancak problem 2 boyutlu olduğu için LP çözmeye gerek kalmadan

aşağıdaki prosedür uygulanarak alt problemlerin etkin kümesi tam olarak bulunabilir. Bunun için önce en küçük girdi değerine sahip olan nokta temel alınarak tüm noktalara olan eğim hesaplanır. Eğer en küçük girdi değerine sahip birden fazla nokta varsa bunların arasından çıktı değeri en büyük olan nokta seçilir. En büyük pozitif eğime sahip olan nokta etkin noktadır ve etkin kümesine dahil edilir (bkz. Teorem 5). Bir sonraki etkin noktayı bulmak için bu nokta temel alınır. Eğer maksimum eğime sahip birden fazla nokta varsa hepsi etkindir ve etkin kümesine dahil edilir. Yeni temel nokta da bunlardan girdisi en büyük olan olarak atanır. Çıktıdan dolayı zayıf etkin olan noktaların eğiminin 0 olduğuna dikkat edilmelidir.

Algoritma 1: 2 boyutlu $m \times p$ adet alt problemdeki etkin KB'lerin bulunması.

Prosedür *FindInitEff(K)*

1: $E = \emptyset,\ KE = \emptyset$

2: **For** j=1 : m

3: **For** i=1 : p

4: Set $(x_{j,base}, y_{i,base}) = \text{lexmin}_{\forall_{k\in K}}\{x_{j,k}, -y_{i,k}\}$

Loop

5: Find $k' = \text{argmax}_{\forall_{k\in K},\, x_{j,k} > x_{j,base}}\{m_{base,k}\}$ *! Tüm eğimleri hesapla* $m_{base,k}$

6: **If** $m_{base,k'} \leq 0$ or $x_{j,base} = \max_{\forall_k}\{x_{j,k}\}$ *!Her alt problem için algoritmanın durma şartı*

7: **Then** break;

8: **Else** $(x_{j,base}, y_{i,base}) = (x_{j,k'}, y_{i,k'})$ *!Sonraki etkin KB'yi temel KB olarak tanımla*

9: $KE = KE \cup DMU_{k'}$

10: $E = E \cup DMU_{k'}$

11: ***END***

12: ***END***

13: ***END***

14: **Return** E *and* KE;

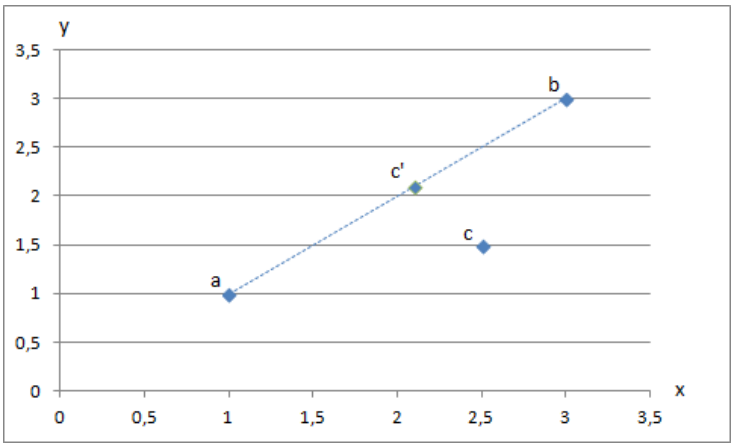
Algoritma 1'in zorluk derecesi (complexity) $O(m\ x\ p\ x\ n^2)$'dir.

Önerme 2:

1 girdi kriteri ve 1 çıktı kriterinden oluşan bir alt problemde bir KB etkin olmayan bir KB ise temel noktadan bu etkin olmayan KB'ye çizilen doğrunun eğiminden daha büyük eğime sahip olan ve temel noktadan başka bir noktaya çizilebilen bir başka doğru vardır.

İspat:

Bir (x_c, y_c) noktası etkin olmayan bir noktaysa etkinlik çizgisi üzerinde $x_c' < x_c \; ve \; y_c' > y_c$ olan bir yanımsa noktası (x_c', y_c') vardır.



Şekil 2: KB'lerin temel alınan nokta ile oluşturdukları doğrunun eğimleri.

a temel nokta, b etkin bir nokta ve c etkin olmayan bir nokta olacak şekilde a,b,c noktalarının olduğunu varsayalım. $\lambda \in [0,1]$ olmak üzere;

$$\begin{bmatrix} x_c' \\ y_c' \end{bmatrix} = \lambda \begin{bmatrix} x_a \\ y_a \end{bmatrix} + (1-\lambda) \begin{bmatrix} x_b \\ y_b \end{bmatrix}$$

Bu durumda temel olan a noktasından b noktasına çizilecek doğrunun eğimi:

$m_{ab} = \frac{y_b - y_a}{x_b - x_a}$ olacaktır ve temel olan a noktasından c' noktasına çizilecek doğrunun eğimi ise:

$$m_{ac'} = \frac{y_{c'} - x_a}{y_{c'} - x_a} = \frac{\lambda y_a + (1-\lambda) y_b - y_a}{\lambda x_a + (1-\lambda) x_b - x_a} = \frac{y_b - y_a}{x_b - x_a}$$

Yani $m_{ac'} = m_{ab}$

Daha önceden $x'_c < x_c \; ve \; y'_c > y_c$ olduğu belirtilmişti bu sebeple $m_{ac'} > m_{ac}$ dolayısıyla c noktası etkin olmayan bir noktadır.

□

Teorem 5: Procedure 1, 1 girdi – 1 çıktı modelinde tüm etkin KB'leri bulur.

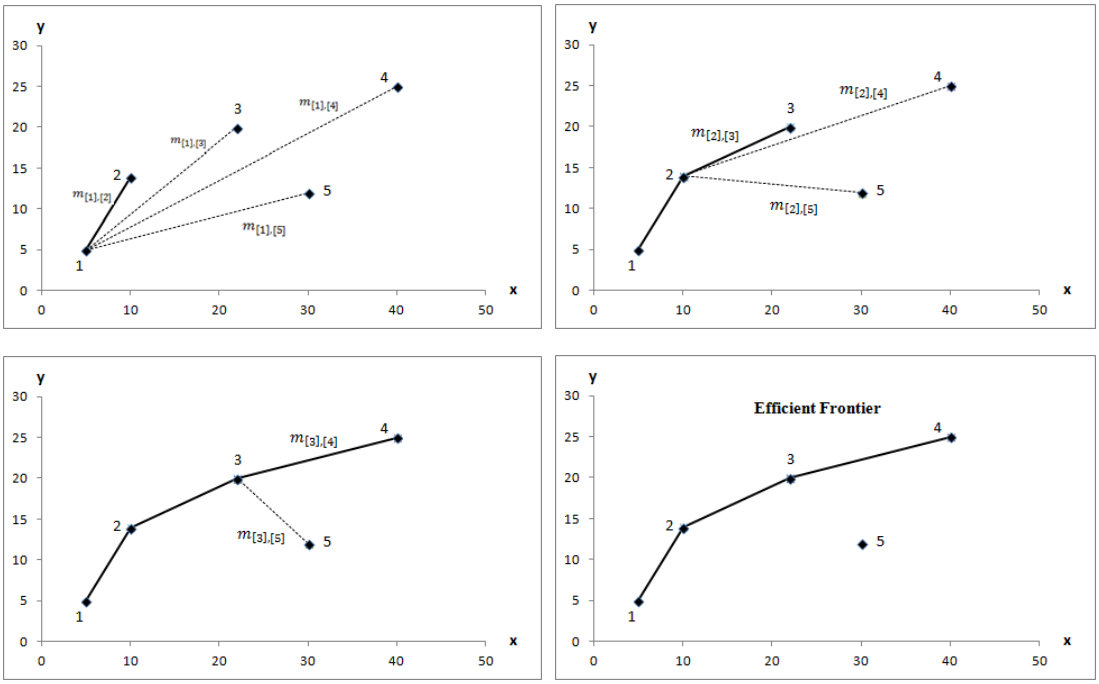
İspat: Etkin KB'lerin girdi kriterine göre küçükten büyüğe sıralandığını varsayalım.

$$x_{[1]} \leq x_{[2]} \leq \cdots \leq x_{[|E|]}$$

En küçük girdi değerine sahip olan nokta $x_{[1]}$ noktasıdır. Bu sebeple $x_{[1]}$ noktası etkindir. Şimdi Önerme 1'e göre $x_{[1]}$ noktası temel nokta seçildiğinde bu noktadan $x_{[2]}$ noktasına çizilen doğru en büyük eğime sahip olan doğru olacaktır. Aksi durumda zarfın konveks olma özelliği ihmal edilmiş olur. Böylece $x_{[2]}$ noktası sıradaki temel nokta olacaktır. Bu şekilde sonuncu $x_{[|E|]}$ noktasına ulaşılabilir. Bu noktadan sonra tüm eğimler ≤ 0 olacaktır. Dolayısıyla etkin KB'lerin eğimleri arasında aşağıdaki gibi bir ilişki vardır:

$$m_{[1],[2]} \geq \cdots \geq m_{[|E|-1],[|E|]}$$

□



Şekil 3: Aşama 1'in süreci.

II. Aşama: *Potansiyel etkin KB'lerin belirlenmesi.*

I. Aşamadan elde edilen KE kümesi bu aşamanın girdisidir. Bu aşamada bütün potansiyel etkin olan KB'ler bulunacaktır. Eğer bir KB KE kümesinden bir KB'yi referans alıyorsa o KB'nin etkin olmadığı kesindir. Ancak KE şu aşamada bütün etkin KB'leri ihtiva etmediği için KE'den referans almayan bir KB'nin etkin olduğu kesin değildir. Ancak potansiyel etkindir. Not edilmelidir ki bu aşamanın sonunda bütün etkin KB'lerin, $KSuper$ kümesine katıldığına emin oluruz. Bu aşamada ($\forall\, KB_0 \notin KE$) için çözülecek LP aşağıdaki gibidir:

Amaç Fonksiyonu:

D$(KE)_0$: Min. θ_0 $(\forall\, KB_0 \notin KE)$

Kısıtlar:

$$\theta_0 * x_{j,0} - \sum_{\forall k \in KE} \lambda_k * x_{j,k} \leq 0 \qquad \forall\, j \qquad (14)$$

$$\sum_{\forall k \in KE} \lambda_k * y_{i,k} \geq y_0 \qquad \forall\, i \qquad (15)$$

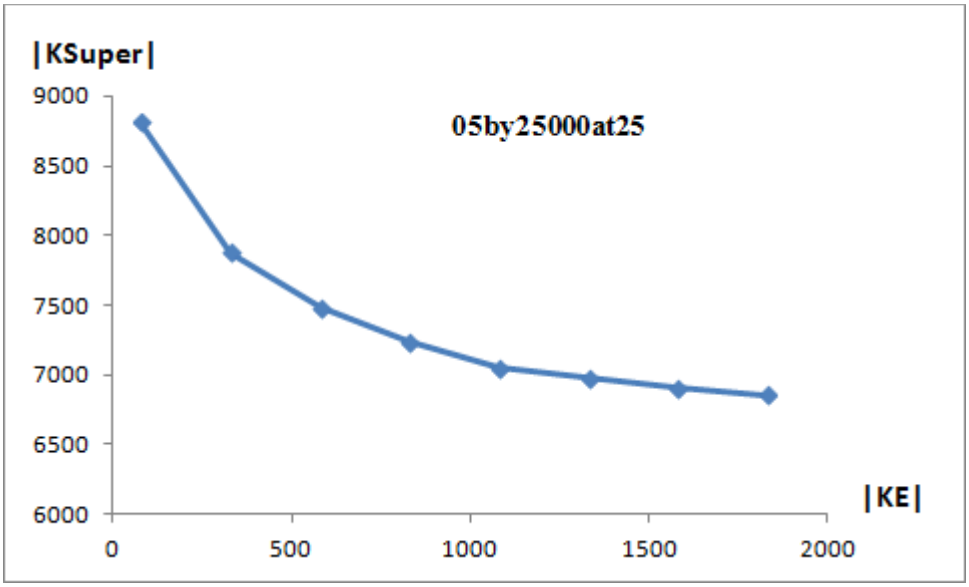
$$\sum_{\forall k \in KE} \lambda_k = 1 \qquad (16)$$

$$\lambda_k \geq 0 \qquad \forall\, k \in KE \qquad (17)$$

Bu aşamada D$(KE)_0$ modeli her $o \notin KE$ için çözülmelidir. Kısıtlar dual modelin kısıtlarıdır ancak $\forall\, k \in KE$ için yazılmıştır ve orijinal modelden değişken sayısı açısından daha küçüktür. Model çıktıları şu şekilde yorumlanır. Eğer bir $KB_0 \notin KE$ için $\theta_0^* = 1.0$ ve $\lambda_0^* = 1$ ise KB_0 , KE kümesinden başka bir KB'yi referans almamıştır ve potansiyel etkindir. Böylece KB_0 , $KSuper$ kümesine eklenir. Sonuç olarak tüm etkin KB'leri ihtiva eden ve K kümesine göre daha küçük bir küme elde edilecektir.

KE kümesi ne kadar küçükse $KSuper$ kümesi o kadar büyük olacaktır. II. Aşamanın çözüm süresi ile III. Aşamanın çözüm süresi arasında bir çelişki (tradeoff) vardır. Deneyimlerimize göre KE kümesine sadece I. Aşamadan gelen etkin KB'lerin eklenmesi yetersiz kalmıştır. Bu kümenin potansiyel etkinler ile desteklenmesi

performans iyileşmesine sebep olmuştur. Bu amaçla KE kümesi Algoritma 2 kullanılarak genişletilmiştir.



Şekil 4: KSuper ve KE kümelerinin büyüklükleri arasındaki çelişki(tradeoff).

KE kümesinin potansiyel etkin KB'lerle desteklenmesi için Algoritma 2'de kullanılmak üzere bir *İdeal KB* belirlenir. İdeal KB girdi kriterlerine göre her bir kriterde en düşük değere; çıktı kriterlerine göre ise her bir kriterde en büyük değere sahip olan bir KB'dir. Burada etkinlik çizgisi tek bir noktadan (İdeal KB'den) oluşmaktadır ve her KB İdeal KB'yi referans almaktadır. Tüm KB'lerin İdeal KB'ye olan uzaklıklarına göre bir göreli etkinlik skoru hesaplanmıştır. Yüksek skora sahip KB'ler İdeal KB'ye daha yakın olanlar ve potansiyel etkin KB'lerdir. Bu KB'lerle KE kümesini genişletmek uygun bir yaklaşım olacaktır. Bu aşamada oluşturulan İdela KB'ye göre etkinlik skorları $D(ideal)_0$ LP'si çözülerek hesaplanır:

Amaç Fonksiyonu:

$D(\text{İ}deal)_0$: $Min.\hat{\theta}_0$ $\forall o \notin KE$

Kısıtlar:

$\hat{\theta}_0 x_{j,0} - \lambda_{ideal} x_{j,ideal} - \sum_{k=1}^{n} \lambda_k x_{j,k} \geq 0$ $\forall$j (18)

$\lambda_{ideal} y_{i,ideal} + \sum_{k=1}^{n} \lambda_k y_{i,k} \geq y_{i,0}$ $\forall$i (19)

$$\lambda_{ideal} + \sum_{k=1}^{n} \lambda_k = 1 \qquad (20)$$

$$\lambda_1, \lambda_2, \dots, \lambda_n,\ \lambda_{ideal} \geq 0 \qquad (21)$$

Teorem 6: $D(Ideal)_o$ *modelinin optimum amaç değeri* $max_j\{\frac{x_{j,ideal}}{x_{j,o}}\}$*'a eşittir.*

İspat: (18) numaralı kısıt setini yeniden düzenlersek $\hat{\theta}_o \geq \lambda_o + \lambda_1 \frac{x_{j,1}}{x_{j,o}} + \lambda_2 \frac{x_{j,2}}{x_{j,o}} + \cdots + \lambda_n \frac{x_{j,n}}{x_{j,o}} + \lambda_{ideal} \frac{x_{j,ideal}}{x_{j,o}} \qquad \forall j = 1,2,\dots,m$ elde ederiz. Buradan en küçük katsayı $\frac{x_{j,ideal}}{x_{j,o}}$ olacağı için daima $\lambda_{ideal} = 1$'dir. Dolayısıyla $\hat{\theta}_0 \geq \frac{x_{j,ideal}}{x_{j,o}}$ $\forall j$olacaktır. (19) ve (20) kısıt setleri her zaman sağlanır. Amacımız $\hat{\theta}_0$'yı minimize etmek olduğu için $\hat{\theta}_0 = \max_j \left\{\frac{x_{j,ideal}}{x_{j,o}}\right\}$' dır.

□

Teorem 7: $\hat{\theta}_0$, θ_0^* *için bir alt sınır (lower bound) oluşturur.*

D_o^{BCC} modelinde optimum amaç değeri $\theta_o^* = \max_j\{\lambda_o + \lambda_1 \frac{x_{j,1}}{x_{j,o}} + \lambda_2 \frac{x_{j,2}}{x_{j,o}} + \cdots + \lambda_n \frac{x_{j,n}}{x_{j,o}}\}$. Herhangi uygun $\boldsymbol{\lambda}$ vektörü için her zaman θ_o^*, $\hat{\theta}_o$'dan büyük olacaktır.

□

Algoritma 2: Enlarge the KE set.

Prosedür *FindSomePotentialEff (K, KE, ne)*

Compute DMU_{ideal}
$x_{j,ideal} = \min_k\{x_{j,k}\} \qquad \forall j = 1,2,\dots,m$
$y_{j,ideal} = \max_k\{y_{i,k}\} \qquad \forall i = 1,2,\dots,p$
Compute $\hat{\theta}_k = \max_j\{\frac{x_{j,ideal}}{x_{j,k}}\} \qquad \forall k \notin KE$
    For g= 1: ne
        $k' = \text{argmax}_{k \notin KE}\{\hat{\theta}_k\}$
        $KE = KE \cup DMU_{k'}$
    END
Return *KE.*

Bu aşamanın sonunda K kümesine göre çok daha küçük olan ve tüm etkin KB'leri ihtiva eden bir $KSuper$ kümesi olulturulmuştur.

III. Aşama: *$KSuper$ kümesinin etkin KB'lerinin bulunması*

$KSuper$ kümesinde tüm etkin KB'ler mevcuttur ancak bunun yanı sıra etkin olmayan KB'lerde $KSuper$ kümesine eklenmiş olabilir. Etkin olmayan KB'leri ayıklamak için bu aşamada klasik VZA modeli çözülür:

Amaç Fonksiyonu:

$\mathrm{D(VZA)_0}$: Min. θ_0 $\qquad\qquad \forall KB_0 \in KSuper$

Kısıtlar:

$$\theta_0 * x_{j,0} - \sum_{\forall k \in KSuper} \lambda_k * x_{j,k} \geq 0 \qquad \forall \mathrm{j} \qquad (22)$$

$$\sum_{\forall k \in KSuper} \lambda_k * y_{i,k} \geq \mathrm{y_0} \qquad \forall \mathrm{i} \qquad (23)$$

$$\lambda_k >= 0 \qquad \forall k \in KSuper \qquad (24)$$

θ free

Algoritma 3: Etkin KB'lerin kümesinin bulunması, E

Prosedür *FindE* (KSuper)

1: **For** $k \in KSuper$
2: $\quad(\theta_k^*, \lambda_k^*) \leftarrow$ Solve $\mathrm{D(KSuper)_k}$
3: $\quad$**If** $\theta_k^* = 1.0$ ve $\lambda_k^* = 1$
4: $\quad\quad$**Then**
5: $\quad\quad E = E \cup DMU_k$
6: ***END.***
7: **Return** E

Bu noktada tüm etkin KB'ler bulunmuştur. Bu aşamada $k \in KSuper$ için hesaplanan etkinlik skorları bütün etkin KB'ler kısıt setinde olduğu için doğrudur. Ancak $k \notin$

$KSuper$ olanlar için zaten II. adımda bulunan skorlar tahmini olarak kullanılabilir. II. adımda KE kümesi bütün etkinleri içermeyebileceğinden dolayı bulunan skorlar doğru diyemeyiz ama yaklaşıktır. Algoritmanın tamamı şu şekildedir:

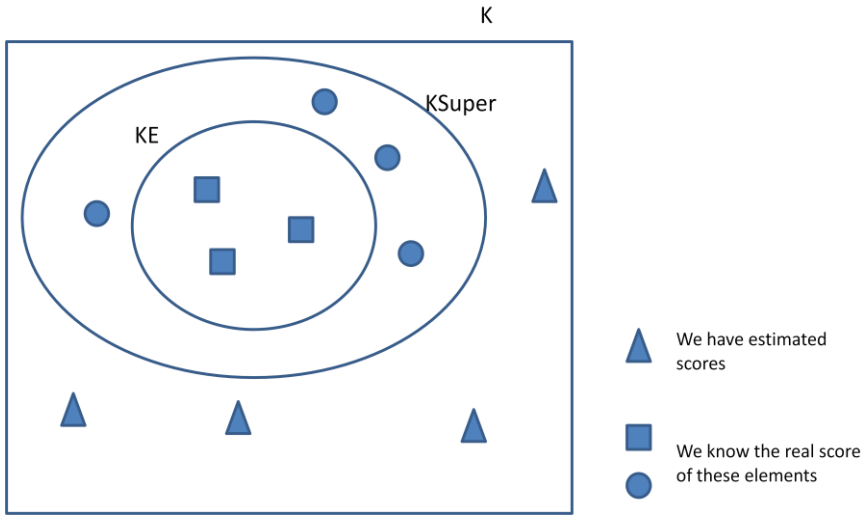
Algoritma 4: Ana Algoritma

Prosedür *ExtendedDecompAlg* (K, X, Y, ne)

1: $E = \emptyset,\ KE = \emptyset,\ KSuper = \emptyset$
2: $E, KE \leftarrow FindInitE\ (K)$!*Aşama I*
3: $KE \leftarrow FindSomePotentialEff\ (K, KE, ne)$!*Aşama II*
4: **For** k=1: n
5: **If** $k \notin KE$
6: **Then**
7: $(\theta_k^*, \lambda_k^*) \leftarrow$ solve $\mathrm{D}(KE)_k$
8: **If** $\theta_k^* = 1.0$ ve $\lambda_k^* = 1$
9: **Then**
10: $Ksuper = Ksuper \cup DMU_k$
11: ***END***.
12: $E \leftarrow FindE\ (KSuper)$!*Aşama III*
13: For $k \in K / KSuper$, **Return** $\hat{\theta}_k$.
14: **Return** E.

Bu çalışmada önerdiğimiz algoritma etkin olmayan KB'lerin etkinlik skorlarını ayrıca LP çözmeden tahmin etmektedir. Dula'nın [11] çalışmasının ikinci adımı bu amaçla yazılmış olduğundan bizim çalışmamız farklılık arz eder. Aslında $D(\dot{I}deal)_0$, $D(KE)_0$ ve $D(KSuper)_0$ modellerinin çözümleri bazı KB'lerin etkinlik skorları hakkında bilgi verir. Örneğin $D(\dot{I}deal)_0$ modelinin çözümü o aşamada her $KB_0 \notin KE$ için etkinlik skornunun tahminini verir. Ardından bu tahmin değerlerinin güncellenerek daha iyi tahminler yapılabilmesi için her $KB_0 \notin KE$ için $D(KE)_0$ modelinin sonuçlarının kullanılması gerekmektedir. Dikkat edilmesi gereken husu, her $KB_0 \in K - KSuper$ için burada bulunan tahmin değerleri nihai tahmin değerleri olacaktır. Daha sonra her

$KB_0 \in KSuper$ için çözülecek olan $D(KSuper)_0$ modeli gerçek etkinlik skorlarını verecektir. Bu husus Şekil 5'te gösterilmiştir:



Şekil 5: Kümelerin şematik gösterimi.

4. BÖLÜM

DENEYSEL ÇALIŞMA VE SONUÇLARI

Bu aşamada Prof. Dr. Jose DULA'dan [12] alınan veri setleri kullanılarak bir uygulama çalışması yapılmıştır. Bulunan sonuçlar literatürdeki algoritmaların sonuçlarıyla karşılaştırılmış ve önerilen algoritmanın etkinliği gösterilmiştir.

Data setinin özelliklerinin performansa etkisini analiz etmek için veri setinde üç karakteristik dikkate alınmıştır. Bunlar, KB sayısı, kriter sayısı (girdi sayısı + çıktı sayısı) ve etkin KB oranıdır. Buna göre veri seti şu şekilde değişim göstermektedir:

- Girdi-çıktı faktörü sayılarına göre;

1. $m + p = 5$
2. $m + p = 10$
3. $m + p = 15$
4. $m + p = 20$

- KB sayılarına göre;

1. $n = 25{,}000$
2. $n = 50{,}000$
3. $n = 75{,}000$
4. $n = 100{,}000$

- Etkin KB yoğunluğuna göre;

1. $d = 1\%$
2. $d = 10\%$
3. $d = 25\%$

Aşağıdaki algoritmalar ($m + p\ by\ n\ at\ d$) şeklinde kodlanmış 48 adet veri setiyle test edilmiştir. Bizim performans göstergelerimiz; CPU süreleri ve etkin olmayan KB'ler için etkinlik skorları tahminlerindeki hatalardır. Her bir örnek için ortalama ve maksimum tahmin hataları şu şekilde hesaplanmıştır:

$$Ortalama\ hata = \frac{\sum_{k\in K}|\theta_k^* - \hat{\theta}_k|}{n} \quad (25)$$

$$Maksimum\ hata = \max_{k\in K}\{\theta_k^* - \hat{\theta}_k\} \quad (26)$$

θ_k^*, KB_k'ya ait optimum etkinlik skorunu; $\hat{\theta}_k$ ise, 2. ve 3. adımlardan gelen tahmini değerleri göstermektedir.

1. Klasik VZA (BCC) Algoritması [1].
2. Classification Algoritması [13].
3. BuildHull Algoritması [11].
4. Önerilen Algoritma.

Kullanılan bilgisayarın işlemcisinin özellikleri: Intel(R) Core(TM) i3 CPU 550 @ 3.20GHz 3.20GHz'dir. Modeller C++ algoritmasında kodlanmış ve ILOG CPLEX 12.1 [19] çözücüsü callable library özelliği kullanılarak çağırılmıştır. VZA orijinal modeli çözülürken şu özellikler aktiftir:

1. No RBE: Etkin olmayan KB'ler bir sonraki aşamada modelden çekilmemiştir.
2. Hot starts: İleri optimizasyon özellikleri kullanılmıştır.
3. CPXlpopt: LP optimizasyon kullanılmıştır.

Tablo 1, klasik VZA algoritması sonuçlarını göstermektedir. Birinci sütunda çözülen problemin özelliği kodlanmıştır. Örneğin 1. problem $m + p = 5$ boyutunda, 25000 KB'nin olduğu ve de etkin KB yüzdesinin %1 olduğu bir problemi göstermektedir. 2. sütunda ise ölçülen süreler verilmiştir.

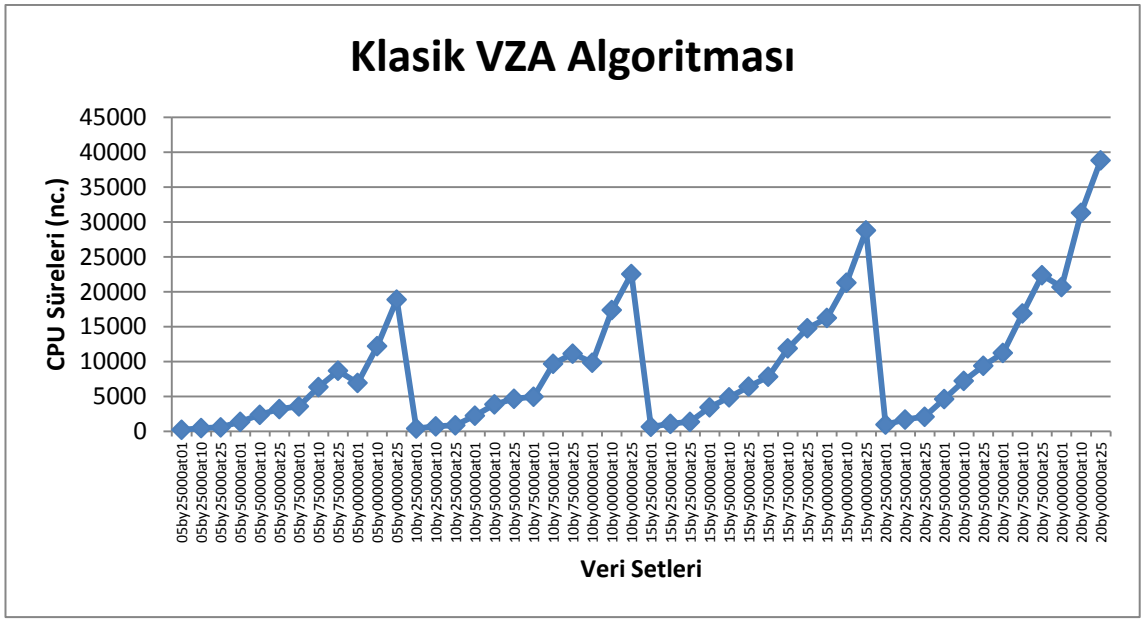
Tablo 1: Klasik VZA Algoritması sonuçları:

Classic DEA (m+p by n at d)	Süre (sn.)	Classic DEA (m+p by n at d)	Süre (sn.)
05by25000at01	229.9	15by25000at01	660.4
05by25000at10	471.8	15by25000at10	1078.8
05by25000at25	574.1	15by25000at25	1364.7
05by50000at01	1372.7	15by50000at01	3445.4
05by50000at10	2370.7	15by50000at10	4874.3
05by50000at25	3192.7	15by50000at25	6412
05by75000at01	3581.2	15by75000at01	7832.3
05by75000at10	6351.5	15by75000at10	11891.4
05by75000at25	8688.3	15by75000at25	14774.5

Tablo 1'in devamı

Classic DEA (m+p by n at d)	Süre (sn.)	Classic DEA (m+p by n at d)	Süre (sn.)
05by00000at25	18873.5	15by00000at25	28785.5
10by25000at01	410.3	20by25000at01	973.8
10by25000at10	718.9	20by25000at10	1693.3
10by25000at25	865.2	20by25000at25	2089.9
10by50000at01	2245.6	20by50000at01	4631
10by50000at10	3865.2	20by50000at10	7233.3
10by50000at25	4668	20by50000at25	9394.4
10by75000at01	4958.7	20by75000at01	11236.7
10by75000at10	9681.1	20by75000at10	16894
10by75000at25	11130.3	20by75000at25	22368.5
10by00000at01	9836.7	20by00000at01	20673.6
10by00000at10	17378.6	20by00000at10	31304.5
10by00000at25	22537.5	20by00000at25	38818.8

Görüldüğü gibi klasik VZA algoritması 11 saate varan sürelerle modelleri çözmüştür. Şekil 6'te çözüm sürelerindeki değişim grafiksel olarak gösterilmiştir. Burada dikkate değer bir husus da şudur; veri setlerinde etkin KB yoğunluğu arttıkça çözüm süresi artmaktadır. Kriter sayısının ve KB sayısının artması da çözüm süresini üstel olarak artırmaktadır. $m + p$ olarak gösterilen, girdi-çıktı faktörü sayıları arttıkça beklendiği gibi çözüm süreleri de artmaktadır.



Şekil 6: Klasik VZA algoritmasının sonuçları.

Tablo 2: Classification Algoritmasının [13] farklı boyuttaki alt problemleri için çözüm sürelerini göstermektedir. q ana problemin bölündüğü alt problem sayısını göstermektedir.

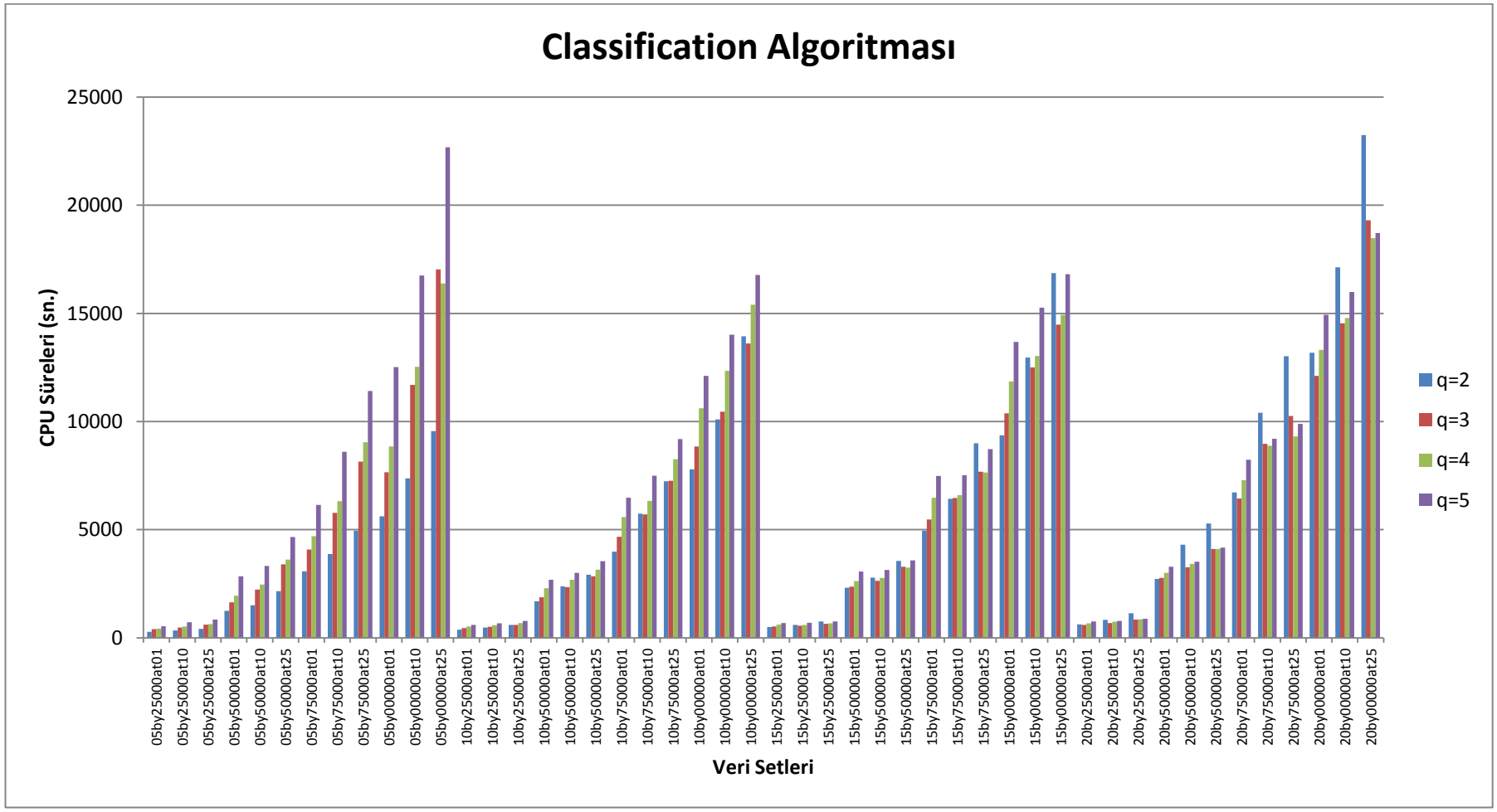
Tablo 2: Classification algoritmasının sonuçları.

Classification Algoritması	Süre (sn.) q=2	Süre (sn.) q=3	Süre (sn.) q=4	Süre (sn.) q=5
05by25000at01	286.2	403	424.8	536.6
05by25000at10	339.6	482.9	531.1	719.2
05by25000at25	412	608.1	635	845.1
05by50000at01	1252.4	1647.4	1954.1	2844
05by50000at10	1509.2	2234.6	2469.1	3327
05by50000at25	2155.2	3394.5	3615	4665.1
05by75000at01	3073.3	4086.7	4695.2	6145.1
05by75000at10	3882.2	5772.7	6318.1	8595.6
05by75000at25	4955.8	8144.6	9043	11407.9
05by00000at01	5616.7	7654.2	8840.3	12509.9
05by00000at10	7368.2	11696.9	12524.4	16750.1
05by00000at25	9560.7	17030.2	16380.5	22675.3
10by25000at01	373.4	453.4	530.2	604.9
10by25000at10	473.2	508.6	589.3	675.4
10by25000at25	601.1	593.7	686.4	786.1
10by50000at01	1689.7	1871.8	2288	2683.3
10by50000at10	2373.4	2337.9	2680.5	2998.9
10by50000at25	2921.5	2842.5	3154.2	3544.2
10by75000at01	3983.1	4674.9	5583.5	6474.6
10by75000at10	5743.8	5705.8	6334.4	7493.6
10by75000at25	7233.5	7260.7	8255.3	9185.1
10by00000at01	7795.7	8843.8	10614.4	12115.9
10by00000at10	10102.8	10457.4	12341.5	14007.5
10by00000at25	13940.4	13610.3	15404.5	16777.3
15by25000at01	498.1	529.2	606.5	684
15by25000at10	599.2	565.5	601.5	693.7
15by25000at25	755.6	646.3	678.1	760.6
15by50000at01	2319.7	2370.3	2628.9	3062.2
15by50000at10	2777.9	2630.7	2777.4	3141.7
15by50000at25	3562.5	3288.3	3245.9	3586.9
15by75000at01	4952.9	5471	6482	7483.6
15by75000at10	6431.1	6468.4	6605.6	7521.7
15by75000at25	8997.5	7681.2	7642.3	8727.4
15by00000at01	9359	10385.3	11858.3	13680.1
15by00000at10	12954.3	12501.8	13028.7	15268.5
15by00000at25	16864.2	14481.5	14929.7	16814.7
20by25000at01	621.3	604.3	669.6	754.9
20by25000at10	827.8	680.6	748.9	785.2

Tablo 2'nin devamı

Classification Algoritması	Süre (sn.) q=2	Süre (sn.) q=3	Süre (sn.) q=4	Süre (sn.) q=5
20by25000at25	1143.5	849.8	856.2	881.7
20by50000at01	2723.3	2773.6	3001.7	3285.3
20by50000at10	4309.8	3259.2	3423	3525
20by50000at25	5286.6	4108.3	4112.2	4165
20by75000at01	6722.7	6438.6	7284.2	8237.8
20by75000at10	10401.2	8963.9	8888	9205.4
20by75000at25	13013.6	10261.3	9307.1	9889.7
20by00000at01	13183.8	12116.5	13308.5	14926.8
20by00000at10	17133.6	14535.6	14783.6	15985.1
20by00000at25	23243.7	19298.6	18485.4	18709.3

q=1, 2, 3, 4 ve 5 için algoritma ayrı ayrı çözülmüş ve sonuçlar Tabloda ve Şekil 7'da gösterilmiştir. Şekil 7'den de görülebileceği gibi küçük q değerlerinde (özellikle q=1 ve q=2) daha iyi sonuçlar alınmıştır. Ayrıca diğer parametrelerin etkilerinin de klasik VZA algoritmasına benzer olduğu gözlemlenmiştir.



Şekil 7: Classification algoritmasının sonuçları.

Tablo 3, BuildHull algoritmasının sonuçlarını göstermektedir.

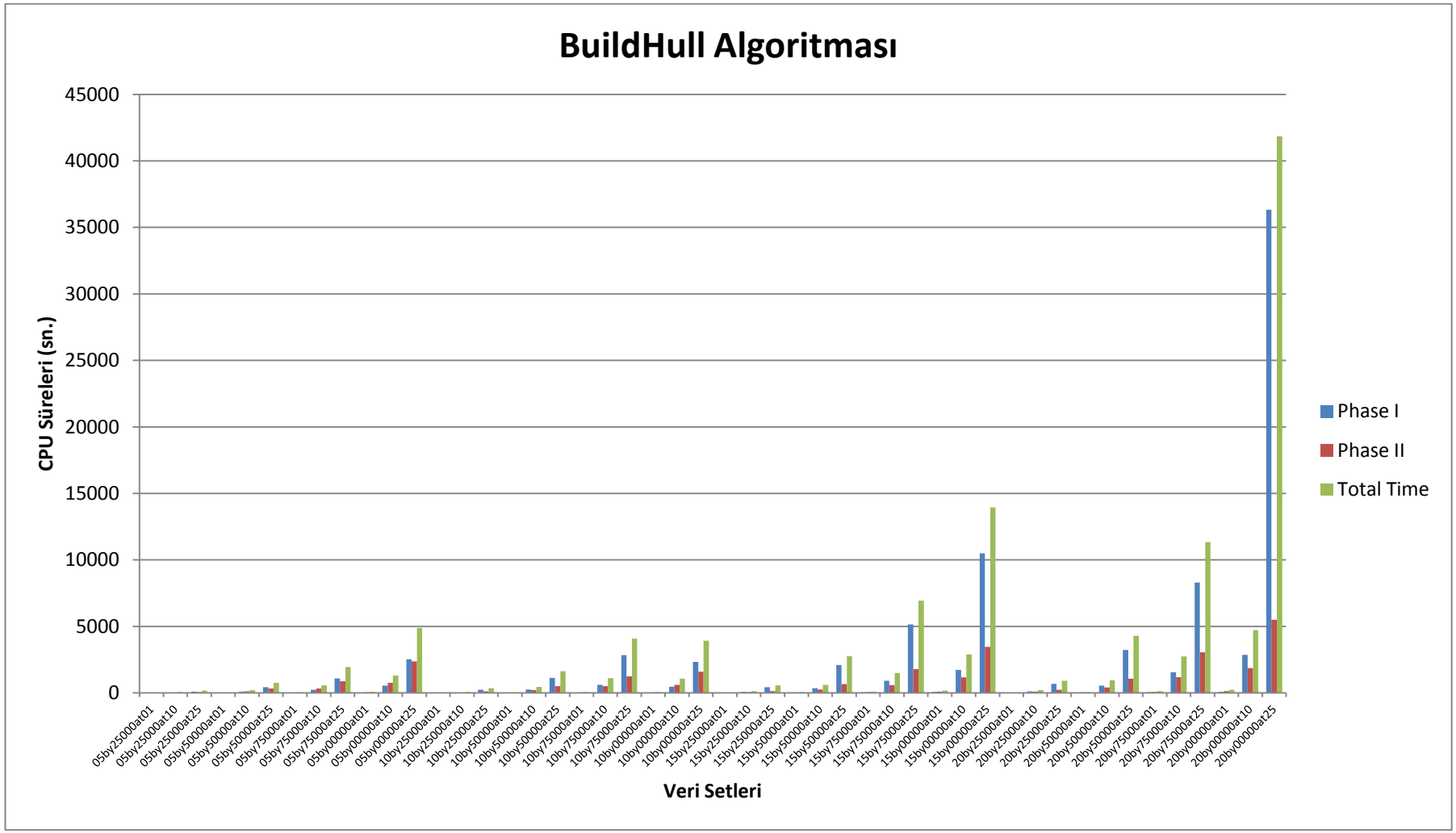
Tablo 3: BuildHull algoritmasının sonuçları.

BuildHull Algoritması	Süre (sn.) Aşama I	Süre (sn.) Aşama II (DEA)	Toplam Süre (sn.)
05by25000at01	2.6	3.1	5.7
05by25000at10	24.7	25.8	50.5
05by25000at25	107	65.1	172
05by50000at01	8.2	11.2	19.4
05by50000at10	88.8	121.3	210.1
05by50000at25	431.7	324.8	756.5
05by75000at01	16.2	23.9	40.1
05by75000at10	242	326.3	568.4
05by75000at25	1080	868.8	1948.8
05by00000at01	25.7	43	68.7
05by00000at10	535.1	765.6	1300.7
05by00000at25	2517.1	2364.8	4881.9
10by25000at01	2.6	3.1	5.7
10by25000at10	24.9	25.9	50.7
10by25000at25	236.2	111.5	347.7
10by50000at01	11.8	17.1	28.9
10by50000at10	246.7	207.2	453.9
10by50000at25	1126.1	502	1628.1
10by75000at01	24.1	32.9	57
10by75000at10	602.4	513.9	1116.3
10by75000at25	2841.7	1247.5	4089.1
10by00000at01	25.8	42.9	68.6
10by00000at10	470.1	595	1065.1
10by00000at25	2325.2	1591.5	3916.7
15by25000at01	7.3	7.8	15
15by25000at10	74	63.8	137.8
15by25000at25	421.3	139.2	560.5
15by50000at01	22.6	27.3	49.9
15by50000at10	353.2	250.5	603.7
15by50000at25	2098.2	654.7	2752.9
15by75000at01	40	54.7	94.6
15by75000at10	909.2	591.3	1500.5
15by75000at25	5140.6	1793	6933.6
15by00000at01	71.3	97.4	168.7
15by00000at10	1732.7	1163.8	2896.4
15by00000at25	10500.6	3449.2	13949.8
20by25000at01	7.3	10.9	18.1
20by25000at10	121.8	99.6	221.4
20by25000at25	686.6	234.3	920.9
20by50000at01	23.2	37.5	60.7

Tablo 3'ün devamı

BuildHull Algoritması	Süre (sn.) Aşama I	Süre (sn.) Aşama II (DEA)	Toplam Süre (sn.)
20by50000at10	536.3	410.2	946.5
20by50000at25	3229.8	1062.5	4292.3
20by75000at01	53.4	80.9	134.4
20by75000at10	1556.5	1192.9	2749.4
20by75000at25	8299.6	3055.2	11354.9
20by00000at01	86.5	140.8	227.3
20by00000at10	2854.9	1857.5	4712.5
20by00000at25	36335.1	5502.3	41837.3

İki aşamadan oluşan BuildHull Algorithm'da genel olarak aşamaların süreleri birbirine yakındır. Yine Tablodan görülebileceği gibi çözüm süreleri d parametresine duyarlıdır. Yoğunluk arttıkça çözüm süresi üstel olarak artmaktadır. Grafikte en çok göze çarpan kısım 20by00000at25 veri setine ait olan değerdir. Bu data setinde algoritmanın çözüm süresi dramatik olarak artmıştır. Ayrıca $m + p$ değeri arttıkça Phase I ve II çözüm süreleri arasındaki zaman farkı ciddi bir şekilde artmaktadır.



Şekil 8: BuildHull algoritmasının sonuçları.

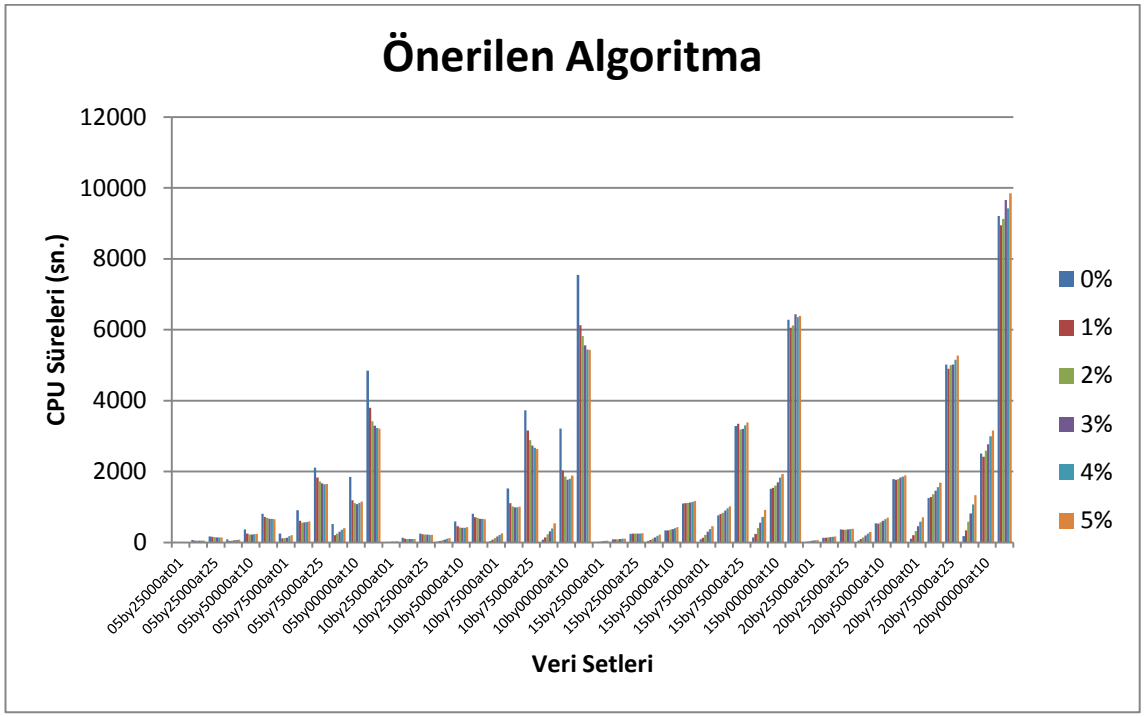
Tablo 4: Önerilen algoritmanın sonuçları

Önerilen Algoritma	ne=0%	ne=1%			ne=2%	ne=3%	ne=4%	ne=5%
	Süre (sn.)	Süre (sn.)	Tahmin Hataları		Süre (sn.)	Süre (sn.)	Süre (sn.)	Süre (sn.)
			Avg.	max.				
05by25000at01	19.3	10	2,47E-13	5E-13	10.9	13	15.6	18
05by25000at10	73.2	54	8,06E-12	2,76E-07	47.4	48	49.4	51.8
05by25000at25	172.1	156.2	4,84E-12	9,53E-08	146	143.5	140.5	141.1
05by50000at01	83.9	42.5	6,16E-14	5E-13	49	56.7	68.1	79.6
05by50000at10	368.3	249.4	5,62E-14	5E-13	223.9	222.2	227.4	239.4
05by50000at25	810.7	719.8	1,98E-12	1,93E-07	688.3	669.5	666.1	656.8
05by75000at01	249.1	115.3	1,24E-13	5E-13	125.3	136.7	175.9	200.8
05by75000at10	907.8	614.1	1,56E-13	4,31E-09	559.4	564	575.9	591.3
05by75000at25	2108.1	1833.3	5,64E-12	3,33E-07	1717.5	1663.4	1642.5	1645.4
05by00000at01	517.1	201.1	1,86E-13	5E-13	253	305.5	357.4	405.2
05by00000at10	1851.5	1184.4	6,22E-13	4,53E-08	1103.6	1081.7	1112.9	1152.4
05by00000at25	4845.9	3800.1	2,03E-12	1,14E-07	3419.9	3288.2	3232	3210.2
10by25000at01	7.7	11.3	2,48E-13	5E-13	16.3	21	25.6	30.2
10by25000at10	128.1	105.5	5,49E-13	2,79E-08	100.4	95.3	94.8	98.2
10by25000at25	250.8	229.7	9,15E-13	4,91E-08	222.1	219.9	216.8	216.9
10by50000at01	22.7	38.4	2,48E-13	5E-13	57.7	79	102.7	125.1
10by50000at10	594.2	455.9	3,13E-12	7,27E-08	418.5	416.6	413.5	431.8
10by50000at25	806.9	713.6	1,86E-13	5E-13	689.1	667.9	664.6	656.5
10by75000at01	42	77.8	2,49E-13	5E-13	119.6	164.5	210.9	260.6
10by75000at10	1520.4	1105.7	9,74E-13	3,74E-08	1013	991.1	992.9	1009.1
10by75000at25	3723.7	3158.9	5,99E-13	1,04E-08	2886.6	2731.7	2665.1	2632.6
10by00000at01	78.3	139.4	2,48E-13	5E-13	227.2	316.9	393.3	540.9
10by00000at10	3209	2029.1	2,26E-13	5E-13	1853.8	1764.3	1797.7	1880.6
10by00000at25	7543.5	6132	1,34E-11	9,24E-08	5820.7	5559.4	5440	5428.1
15by25000at01	15.4	19.3	2,47E-13	5E-13	26.2	33.7	41	48.6
15by25000at10	83.6	86.8	2,52E-13	2,52E-09	90.1	95.5	101.8	107.1
15by25000at25	243	245.8	1,72E-12	4,24E-08	245.7	245.3	248.6	257.3
15by50000at01	44	66.8	6,17E-14	5E-13	103	139	184.8	218.7
15by50000at10	338.4	340.1	1,06E-13	5,01E-09	362	379.2	401.8	433.2
15by50000at25	1098.3	1104.8	5,16E-14	4,72E-10	1117.7	1129.9	1145.6	1173.6
15by75000at01	84.1	132.2	1,24E-13	5E-13	212.7	301.1	377.9	459.7
15by75000at10	767.2	800.6	1,13E-13	5E-13	838.2	899.1	960	1018.7
15by75000at25	3286.6	3342.5	4,17E-13	1,52E-08	3183.7	3203.3	3300.9	3382.6
15by00000at01	143.4	241.6	1,86E-13	5E-13	400.3	553.4	723.3	919.7
15by00000at10	1511.3	1536.8	4,5E-13	2,81E-08	1604.1	1698.6	1829.7	1931.7
15by00000at25	6284.4	6058.5	1,21E-12	2,7E-08	6116.4	6433.4	6349.7	6389.3
20by25000at01	18.3	27.2	2,48E-13	5E-13	38.7	49.1	59.8	70.6
20by25000at10	134.2	136.4	1,36E-12	8,18E-08	143.8	152.9	162.1	173
20by25000at25	365.2	358.5	1,58E-12	4,52E-08	362.4	369.2	376.1	386.3
20by50000at01	53.9	93.3	2,49E-13	5E-13	140.1	192.6	248	298
20by50000at10	542.2	533.6	2,72E-13	1,14E-09	563.6	608.3	652.2	699.3

Tablo 4'ün devamı

Önerilen Algoritma	ne=0%	ne=1%			ne=2%	ne=3%	ne=4%	ne=5%
	Süre (sn.)	Süre (sn.)	Tahmin Hataları		Süre (sn.)	Süre (sn.)	Süre (sn.)	Süre (sn.)
20by50000at25	1785.8	1767.5	3,47E-13	4,04E-09	1788.9	1831.5	1855.1	1890.3
20by75000at01	106.5	202	2,47E-13	5E-13	324.2	456.8	581.6	709.9
20by75000at10	1249.6	1282.9	2,24E-13	5E-13	1364.1	1457.4	1558	1682.4
20by75000at25	5020.7	4903.4	9,13E-13	2,14E-08	4999	5020.8	5149.4	5270.2
20by00000at01	180.6	339.3	2,48E-13	5E-13	584.8	820.3	1071.9	1332.5
20by00000at10	2509.4	2412.4	2,26E-13	5E-13	2588.8	2769.3	2995.2	3157.8
20by00000at25	9205.9	8948.5	1,09E-12	1,81E-08	9125.6	9653.9	9422.4	9847.6

Burada küçük ne değerleri için daha iyi performans gözlemlenmektedir. İlginçtir ki 1%'lik genişletme diğer durumlara göre genelde daha iyi sonuçlar vermiştir. Bu durum için (25) ve (26) numaralı denklemlerde gösterildiği gibi ortalama hata ve maksimum hata değerlerini hesapladık. Bu sonuçlara göre tahmin hataları oldukça düşük çıkmıştır. Bu değerlerden tüm data setlerinde en büyüğü [0,1] aralığında 3.33E-07 olarak hesaplanmıştır. Şekil 9 incelendiğimizde diğer algoritmalarla benzer bir trend gözlemledik. Şekil 10'da tüm algoritmaların hesaplanma süreleri verilmiştir. Bu şekle göre CPU zamanı açısından önerilen algoritma tüm algoritmalardan daha iyi sonuçlara sahiptir. Eğer her bir algoritmayı CPU zamanı açısından tek tek karşılaştıracak olursak, önerilen algoritma Classification algoritmasını tüm data setlerinde en fazla 98.4%; en az 52.2% oranında iyileştirmiştir. Yine önerilen algoritmanın BuildHull algoritmasına olan iyileştirme değerleri en yüksek 78.6% ve en az -65.8% olarak hesaplanmıştır. Buradaki negatif değer BuildHull algoritmasının önerilen algoritmadan daha iyi bir performansa sahip olduğunu göstermektedir. Ayrıca K kümesini $KSuper$ kümesine hangi oranda indirgediğimizi ($\frac{|KSuper|}{|K|}$) ve $KSuper$ kümesinin E kümesine ne kadar yakın olduğunu hesapladık ($\frac{|E|}{|KSuper|}$). İdeal durumda $|KSuper|$ değerinin $|E|$ değerine yakın olması dolayısıyla $\frac{|KSuper|}{|K|}$ oranının d değerine yakın olması beklenir. Tablo 2'ye göre her data setinde $\frac{|KSuper|}{|K|}$ oranının d değerine çok yakın olduğu görülmektedir. Ayrıca $\frac{|E|}{|KSuper|}$ oranının 1.0 değerine yakın olması beklenir. Özellikle problemin büyüklüğü arttıkça önerilen algoritmanın $KSuper$ kümesini belirlemede daha iyi performans gösterdiğini gözlemlenmiştir.



Şekil 9: Önerilen algoritmanın sonuçları.

Tablo 5. Önerilen algoritma için |KSuper|/|K| ve |E|/|KSuper| sonuçları.

Veri Setleri	ne = 1%	
	|Ksuper|/|K|	|E|/|Ksuper|
05by25000at01	0,04	0,27
05by25000at10	0,15	0,68
05by25000at25	0,32	0,79
05by50000at01	0,04	0,26
05by50000at10	0,15	0,68
05by50000at25	0,31	0,80
05by75000at01	0,04	0,24
05by75000at10	0,15	0,68
05by75000at25	0,31	0,80
05by00000at01	0,04	0,25
05by00000at10	0,15	0,67
05by00000at25	0,32	0,79
10by25000at01	0,02	0,58
10by25000at10	0,18	0,57
10by25000at25	0,32	0,77
10by50000at01	0,02	0,62
10by50000at10	0,17	0,58
10by50000at25	0,31	0,80
10by75000at01	0,02	0,63
10by75000at10	0,17	0,60
10by75000at25	0,32	0,78

Tablo 5'in devamı

Veri Setleri	ne = 1%	
	|Ksuper|/|K|	|E|/|Ksuper|
10by00000at01	0,02	0,64
10by00000at10	0,17	0,60
10by00000at25	0,32	0,78
15by25000at01	0,02	0,56
15by25000at10	0,10	1,00
15by25000at25	0,25	1,00
15by50000at01	0,02	0,62
15by50000at10	0,10	1,00
15by50000at25	0,25	1,00
15by75000at01	0,02	0,63
15by75000at10	0,10	1,00
15by75000at25	0,25	1,00
15by00000at01	0,02	0,64
15by00000at10	0,10	1,00
15by00000at25	0,25	1,00
20by25000at01	0,02	0,59
20by25000at10	0,10	0,99
20by25000at25	0,25	1,00
20by50000at01	0,01	0,68
20by50000at10	0,10	1,00
20by50000at25	0,25	1,00
20by75000at01	0,01	0,69
20by75000at10	0,10	1,00
20by75000at25	0,25	1,00
20by00000at01	0,01	0,70
20by00000at10	0,10	1,00
20by00000at25	0,25	1,00

Tablo 6: Tüm Algoritmaların sonuçları

Veri Setleri	Klasik VZA (sn.)	BuildHull Algoritması (sn.)	Classification Algoritması (q=3) (sn.)	Önerilen Algoritma (%1) (sn.)
05by25000at01	229.9	5.7*	403	10
05by25000at10	471.8	50.5*	482.9	54
05by25000at25	574.1	172	608.1	156.2*
05by50000at01	1372.7	19.4*	1647.4	42.5*
05by50000at10	2370.7	210.1*	2234.6	249.4
05by50000at25	3192.7	756.5	3394.5	719.8*
05by75000at01	3581.2	40.1*	4086.7	115.3
05by75000at10	6351.5	568.4*	5772.7	614.1
05by75000at25	8688.3	1948.8	8144.6	1833.3*

Tablo 6'nın devamı

Veri Setleri	Klasik VZA (sn.)	BuildHull Algoritması (sn.)	Classification Algoritması (q=3) (sn.)	Önerilen Algoritma (%1) (sn.)
05by00000at25	18873.5	4881.9	17030.2	3800.1*
10by25000at01	410.3	5.7*	453.4	11.3
10by25000at10	718.9	50.7*	508.6	105.5
10by25000at25	865.2	347.7	593.7	229.7*
10by50000at01	2245.6	28.9*	1871.8	38.4
10by50000at10	3865.2	453.9*	2337.9	455.9
10by50000at25	4668	1628.1	2842.5	713.6*
10by75000at01	4958.7	57*	4674.9	77.8
10by75000at10	9681.1	1116.3	5705.8	1105.7*
10by75000at25	11130.3	4089.1	7260.7	3158.9*
10by00000at01	9836.7	68.6*	8843.8	139.4
10by00000at10	17378.6	1065.1*	10457.4	2029.1
10by00000at25	22537.5	3916.7*	13610.3	6132
15by25000at01	660.4	15*	529.2	19.3
15by25000at10	1078.8	137.8	565.5	86.8*
15by25000at25	1364.7	560.5	646.3	245.8*
15by50000at01	3445.4	49.9*	2370.3	66.8
15by50000at10	4874.3	603.7	2630.7	340.1*
15by50000at25	6412	2752.9	3288.3	1104.8*
15by75000at01	7832.3	94.6*	5471	132.2
15by75000at10	11891.4	1500.5	6468.4	800.6*
15by75000at25	14774.5	6933.6	7681.2	3342.5*
15by00000at01	16251	168.7*	10385.3	241.6
15by00000at10	21295.7	2896.4	12501.8	1536.8*
15by00000at25	28785.5	13949.8	14481.5	6058.5*
20by25000at01	973.8	18.1*	604.3	27.2
20by25000at10	1693.3	221.4	680.6	136.4*
20by25000at25	2089.9	920.9	849.8	358.5*
20by50000at01	4631	60.7*	2773.6	93.3
20by50000at10	7233.3	946.5	3259.2	533.6*
20by50000at25	9394.4	4292.3	4108.3	1767.5*
20by75000at01	11236.7	134.4*	6438.6	202
20by75000at10	16894	2749.4	8963.9	1282.9*
20by75000at25	22368.5	11354.9	10261.3	4903.4*
20by00000at01	20673.6	227.3*	12116.5	339.3
20by00000at10	31304.5	4712.5	14535.6	2412.4*
20by00000at25	38818.8	41837.3	19298.6	8948.5*

* Belirtilen veri seti için en iyi performansı gösteren algoritma.

Tablo 6'da tüm algoritmaların sonuçları aynı anda gösterilmiştir.

Tablo 7: Önerilen algoritmanın Classification algoritmasına göre iyileştirme yüzdesi

İyileştirmeler (%)		İyileştirmeler (%)	
05by25000at01	97.5	15by25000at01	96.4
05by25000at10	88.8	15by25000at10	84.7
05by25000at25	74.3	15by25000at25	62
05by50000at01	97.4	15by50000at01	97.2
05by50000at10	88.8	15by50000at10	87.1
05by50000at25	78.8	15by50000at25	66.4
05by75000at01	97.2	15by75000at01	97.6
05by75000at10	89.4	15by75000at10	87.6
05by75000at25	77.5	15by75000at25	56.5
05by00000at01	97.4	15by00000at01	97.7
05by00000at10	89.9	15by00000at10	87.7
05by00000at25	77.7	15by00000at25	58.2
10by25000at01	97.5	20by25000at01	95.5
10by25000at10	79.3	20by25000at10	80
10by25000at25	61.3	20by25000at25	57.8
10by50000at01	97.9	20by50000at01	96.6
10by50000at10	80.5	20by50000at10	83.6
10by50000at25	74.9	20by50000at25	57
10by75000at01	98.3	20by75000at01	96.9
10by75000at10	80.6	20by75000at10	85.7
10by75000at25	56.5	20by75000at25	52.2
10by00000at01	98.4	20by00000at01	97.2
10by00000at10	80.6	20by00000at10	83.4
10by00000at25	54.9	20by00000at25	53.6

Tablo 7'de gösterilen değerler önerilen algoritmanın Classification Algorithm'a göre sağladığı iyileştirme yüzdesini göstermektedir. Buna göre önerilen algoritma maksimum 98.4%; minimum 52.2% iyileştirme sağlamıştır.

Tablo 8: Önerilen Algoritmanın BuildHull algoritmasına göre iyileştirme yüzdesi

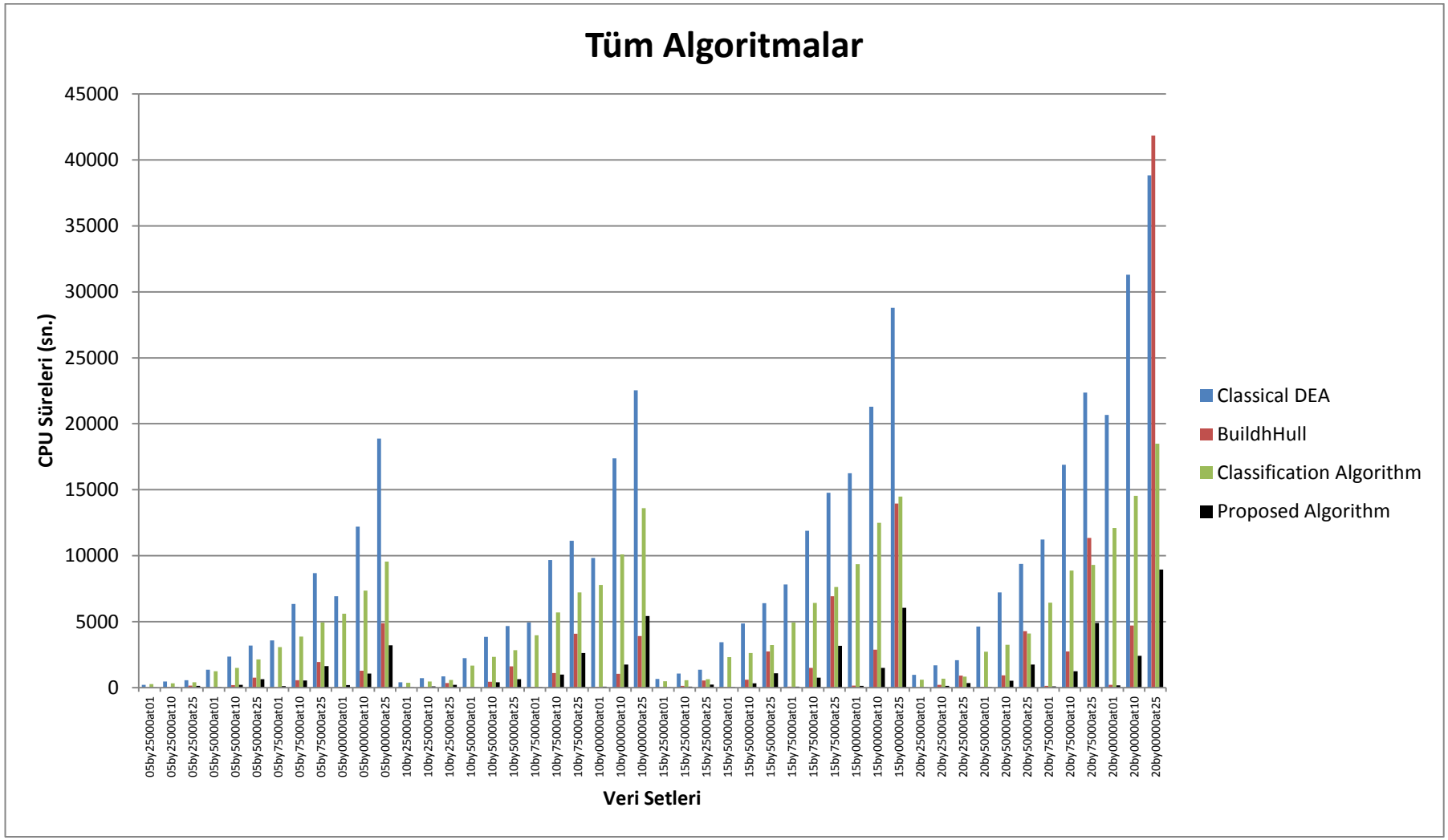
İyileştirmeler (%)		İyileştirmeler (%)	
05by25000at01	-43	15by25000at01	-22.3
05by25000at10	-6.5	15by25000at10	37
05by25000at25	9.2	15by25000at25	56.1
05by50000at01	-54.4	15by50000at01	-25.3
05by50000at10	-15.8	15by50000at10	43.7
05by50000at25	4.9	15by50000at25	59.9
05by75000at01	-65.2	15by75000at01	-28.4
05by75000at10	-7.4	15by75000at10	46.6
05by75000at25	5.9	15by75000at25	51.8
05by00000at01	-65.8	15by00000at01	-30.2
05by00000at10	8.9	15by00000at10	46.9
05by00000at25	22.2	15by00000at25	56.6
10by25000at01	-49.6	20by25000at01	-33.5
10by25000at10	-51.9	20by25000at10	38.4
10by25000at25	33.9	20by25000at25	61.1
10by50000at01	-24.7	20by50000at01	-34.9
10by50000at10	-0.4	20by50000at10	43.6
10by50000at25	56.2	20by50000at25	58.8
10by75000at01	-26.7	20by75000at01	-33.5
10by75000at10	0.9	20by75000at10	53.3
10by75000at25	22.7	20by75000at25	56.8
10by00000at01	-50.8	20by00000at01	-33
10by00000at10	-47.5	20by00000at10	48.8
10by00000at25	-36.1	20by00000at25	78.6

Tablo 8, iyileştirme yüzdelerini göstermektedir. Tabloya göre pozitif olan değerler önerilen algoritmanın BuildHull Algorithm'a göre sağladığı iyileştirme yüzdesini; negatif değerler ise BuildHull Algorithm'ın önerilen algoritmaya göre iyileştirme yüzdesini göstermektedir. Buna göre önerilen algoritma BuildHull Algorithm'a göre maksimum 78.6% iyileştirme sağlarken; BuildHull Algorithm ise önerilen algoritmaya göre maksimum 65.8% iyileştirme sağlamıştır.

Tablo 9: Önerilen algoritmanın Klasik VZA algoritmasına göre iyileştirme yüzdesi

İyileştirmeler (%)		İyileştirmeler (%)	
05by25000at01	95.7	15by25000at01	97.1
05by25000at10	88.6	15by25000at10	92
05by25000at25	72.8	15by25000at25	82
05by50000at01	96.9	15by50000at01	98.1
05by50000at10	89.5	15by50000at10	93
05by50000at25	77.5	15by50000at25	82.8
05by75000at01	96.8	15by75000at01	98.3
05by75000at10	90.3	15by75000at10	93.3
05by75000at25	78.9	15by75000at25	77.4
05by00000at01	97.1	15by00000at01	98.5
05by00000at10	90.3	15by00000at10	92.8
05by00000at25	79.9	15by00000at25	79
10by25000at01	97.2	20by25000at01	97.2
10by25000at10	85.3	20by25000at10	91.9
10by25000at25	73.5	20by25000at25	82.8
10by50000at01	98.3	20by50000at01	98
10by50000at10	88.2	20by50000at10	92.6
10by50000at25	84.7	20by50000at25	81.2
10by75000at01	98.4	20by75000at01	98.2
10by75000at10	88.6	20by75000at10	92.4
10by75000at25	71.6	20by75000at25	78.1
10by00000at01	98.6	20by00000at01	98.4
10by00000at10	88.3	20by00000at10	92.3
10by00000at25	72.8	20by00000at25	76.9

Tablo 9'da gösterilen değerler önerilen algoritmanın Classic DEA Algoritmasına göre sağladığı iyileştirme yüzdesini göstermektedir. Buna göre önerilen algoritma maksimum 98.6%; minimum 71.6% iyileştirme sağlamıştır.



Şekil 10: Algoritma sonuçlarının birbirleriyle karşılaştırılması

Şekil 10’da her bir algoritma her bir veri setinde en iyi performans gösterdiği parametreler ile çalıştırılmış ve bu grafik oluşturulmuştur. Şekilden de rahatlıkla görülebileceği gibi önerilen algoritma her veri seti için çok iyi bir performans göstermiştir.

5. BÖLÜM
SONUÇ

Bu çalışmada büyük data setleri için bir algoritma geliştiriliştir. Algoritma ilk olarak problemi girdi-çıktı faktörlerini parçalayarak iki boyutlu alt problemler elde etmektedir. Daha sonra etkin KB'ler sistematik olarak seçilmektedir. Bu çalışmada önerilen algoritma sadece etkin KB'leri bulmakla kalmayıp tüm KB'lerin etkinlik skorları için tahminler sunmaktadır. Eğer bir KB $KSuper$ kümesinin üyesiyse bulunan etkinlik skoru hatasızdır. Ayrıca yapılan tahmin değerleri incelendiğinde hataların oldukça düşük olduğu gözlemlenmiştir. Algoritma test edilmiş ve klasik VZA algoritmasıyla beraber literatürdeki iki algoritmayla daha karşılaştırılmıştır. Sonuç olarak önerilen algoritma büyük data setine sahip olan problemlerin çözüm sürelerini iyileştirmiştir.

Algoritmaların performansının daha da iyileştirilmesi mümkündür. Bu konu ileri araştırma konusu olarak düşünülebilir. Örneğin, 3. Aşamada etkin olmayan KB'ler için çözüm yaptığımızda bulunan referans kümesi etkin olarak işaretlenebilir ve daha sonraki LP çözümlerine dahil edilmeyebilir. Uygulama aşamasında ise Türkiye çapında veriler toplanarak uygulama yapılabilir. Örneğin, Türkiye'deki sağlık personellerinin performans analizi ve sonuçta elde edilen skorlara göre teşvik/prim uygulaması gibi.

KAYNAKLAR

1. Charnes, A., W.W. Cooper, E. Rhodes, 1978. Measuring the efficiency of decision making units. **European Journal of Operational Research, 2** (6): 429-444.

2. Cook, W.D., L.M. Seiford, 2009. Data envelopment analysis (DEA) – Thirty years on. **European Journal of Operational Research, 192** (1): 1-17.

3. Liu, J.S., Lu, L.Y.Y., Lu, W.M., Lin, B.J.Y., 2013. A survey of DEA applications. **Omega, 41** (5): 893-902.

4. Liu, J.S., Lu, L.Y.Y., Lu, W.M., Lin, B.J.Y., 2013. A citation-based literature survey, **Omega, 41** (1): 3-15.

5. Emrouznejad, A., Parker, B.R., Tavares, G., 2008. Evaluation of research in efficiency and productivity: A survey and analysis of the first 30 years of scholarly literature in DEA. **Socio-Economic Planning Sciences, 42** (3): 151-157.

6. Cook, W.D., Joe Zhu., 2007. Classifying inputs and outputs in data envelopment analyses. **European Journal of Operational Research, 180** (2): 692-699.

7. Chen, Y., W.D. Cook, J. Zhu., 2010. Deriving the DEA frontier for two-stage processes. **European Journal of Operational Research, 202** (1): 138-142.

8. Emrouznejad, A., E. Shale., 2009. A combined neural network and DEA for measuring efficincy of large scale datasets. **Computers and Industrial Engineering, 56** (1): 249-254.

9. Ehrgott M., Tind J., 2009. Column generation with free replicability in DEA. **Omega, 37** (5): 943-950.

10. Chen, H.W., N.B. Chang, J.C. Chen, S.J. Tsai., 2010. Environmental performance evaluation of large-scale municipal solid waste incinerators using data envelopment analysis. **Waste Management, 30** (7): 1371-1381.

11. Dula, J.H., 2011. An Algorithm for Data Envelopment Analysis, **Informs Journal on Computing. 23** (2): 284-296.

12. Dula, J.H., 2008. A computational study of DEA with massive data sets. **Computers & Operations Research, 35** (4): 1191-1203.

13. Korhonen, P.J., Siitari, P.-A., 2009. A dimensional decomposition approach to identifying efficient units in large-scale DEA models. **Computers & Operations Research, 36** (1): 234-244.

14. Chen, W.C., W.J. Cho., 2009. A procedure for large-scale DEA computations. **Computers & Operations Research, 36** (6): 1813-1824.

15. Banker, R.D., A. Charnes, W.W. Cooper., 1984. Some Models for Estimating Technical and Scale Inefficient in Data Envelopment Analysis. **Management Science 30** (9): 1078-1092.

16. Fukuyama, H., Sekitani, K., 2012. Decomposing the frontier of the DEA production possibility set into a smallest number of convex polyhedrons by mixed integer programming. **European Journal of Operational Research, 221** (1): 165-174.

17. Pendharkar, P.C., M.D. Troutt., 2011. DEA based dimensionality reduction for classification problems satisfying strict non-satiety assumption. **European Journal of Operational Research, 212** (1): 155-163.

18. Cooper, W.W., Seiford, L.M., 2006. Introduction To Data Envelopment Analysis And Its Uses. **Springer**.

19. ILOG. 2009. IBM ILOG CPLEX V12.1 User's Manual for CPLEX, **International Business Machines Corporation 1987.**

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Gazi Bilal YILDIZ
Uyruğu: Türkiye (TC)
Doğum Tarihi ve Yeri: 13 Ocak 1988, Şarkışla
Medeni Durumu: Bekâr
Tel: +90 532 797 55 98
email: bilalyildiz@erciyes.edu.tr
Yazışma Adresi: Erciyes Üniversitesi Endüstri Mühendisliği Bölümü 38039 Talas/KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	EÜ Fen Bilimleri Enstitüsü	2013
Lisans	EÜ Endüstri Mühendisliği	2010
Lise	Şarkışla Anadolu Lisesi, Sivas	2005

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2011- Halen	EU Endüstri Mühendisliği Bölümü	Araştırma Görevlisi

YABANCI DİL

İngilizce.