

PERFORMANS ANALİZİ İÇİN YENİ KARAR DESTEK SİSTEMİ

Volkan Soner ÖZSOY

**YÜKSEK LİSANS TEZİ
İSTATİSTİK**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

EYLÜL 2012

ANKARA

Volkan Soner ÖZSOY tarafından hazırlanan “PERFORMANS ANALİZİ İÇİN YENİ KARAR DESTEK SİSTEMİ” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. İhsan ALP

.....

Tez Danışmanı, İstatistik Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile İstatistik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Ertan GÜNER

.....

Endüstri Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Prof. Dr. İhsan ALP

.....

İstatistik Anabilim Dalı, Gazi Üniversitesi

Prof. Dr. Hasan BAL

.....

İstatistik Anabilim Dalı, Gazi Üniversitesi

Tarih: 06/09/2012

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Şeref SAĞIROĞLU

.....

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Volkan Soner ÖZSOY

PERFORMANS ANALİZİ İÇİN YENİ KARAR DESTEK SİSTEMİ

(Yüksek Lisans Tezi)

Volkan Soner ÖZSOY

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Eylül 2012

ÖZET

Performans analizi yöntemlerinden biri olan Veri Zarflama Analizi, karar verme birimlerine ait değişkenlerin gözlem değerleri kullanılarak, göreceli etkinliğin ölçüldüğü parametrik olmayan bir yöntemdir. Veri Zarflama Analizi, hangi karar verme biriminin ne derece etkin olduğu ve etkin olmayan karar verme birimlerinin etkin olabilmelerini sağlayacak önerilerde bulunduğu için tercih edilen bir yöntemdir. Veri Zarflama Analizi için farklı türde bilgisayar paket programları kullanılabilmektedir. Ancak yabancı dillerde hazırlanmış bu paket programlarının birçok eksik yönleri bulunmaktadır. Bu çalışmada eksik yönlerin giderilebilmesi için, girdi ve çıktı yönlü CCR (Charnes Cooper Rhodes) - BCC (Banker Charnes Cooper) modelleri, Window analizi, kontrol edilemeyen değişken durumları gibi çeşitli özelliklerin her birini kapsayan, Microsoft Visual C# programlama dili ile ve tamamen Türkçe bir performans analizi paket programı geliştirilmiştir. Bu paket program, hem muadillerinden daha kullanışlı hem de yerli üretim olması sebebiyle ülkemize katma değer getirmesi düşünülmektedir.

Bilim Kodu : 205.1.148

Anahtar Kelimeler : Veri Zarflama Analizi, Performans, Etkinlik

Sayfa Adedi : 112

Tez Yöneticisi : Prof. Dr. İhsan ALP

NEW DECISION SUPPORT SYSTEM FOR PERFORMANCE ANALYSIS
(M.Sc. Thesis)

Volkan Soner ÖZSOY

GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
September 2012

ABSTRACT

Data Envelopment Analysis, one of the method of performance analysis, is the non-parametric method in which relative efficiency is measured by using obsevrance values of variables belonging to decision making units. Data Envelopment Analysis is a preferable method since it not only shows which decision making unit is effective but also gives advices about turning inefficient decision making units into efficient decision making units. Different types of computer packet programmes may be used for Data Envelopment Analysis. However, there are some deficiencies in these packet programmes prepared in foreign languages. In this study, completely Turkish performance analysis package program has been developed with Microsoft Visual C# programming language covering each of the various features such as the input-oriented and output-oriented CCR (Charnes Cooper Rhodes) - BCC (Banker Charnes Cooper) models, Window analysis, uncontrolled variable status in order to complete the deficiencies. This packet programme is thought to bring added value to our country as it is more useful than its equivalents and it is a domestic article.

Science Code : 205.1.148
Key Words : Data Envelopment Analysis, Performance, Efficiency
Page Number : 112
Adviser : Prof. Dr. İhsan ALP

TEŞEKKÜR

Çalışmalarım boyunca beni yönlendiren değerli Hocam Prof. Dr. İhsan ALP'e ve jüri üyelerim Prof. Dr. Hasan BAL ile Prof. Dr. Ertan GÜNER'e, yüksek lisans eğitimim boyunca üzerimde emeği geçen tüm Gazi Üniversitesi İstatistik Anabilim Dalında görev yapan hocalarıma, ayrıca lisans eğitimim boyunca üzerimde emeği geçen tüm Karadeniz Teknik Üniversitesi İstatistik ve Bilgisayar Bilimleri Bölümünde görev yapan hocalarıma teşekkür ederim.

Son olarak ama sonsuz olarak her şeyin daha güzel olmasına vesile olan değerli aileme teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER.....	vii
ÇİZELGELERİN LİSTESİ	x
ŞEKİLLERİN LİSTESİ	xi
RESİMLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ	1
2. PERFORMANS ANALİZİ	4
2.1. Performansın Tanımı.....	4
2.2. Performans Analizi ile İlgili Temel Kavramlar	4
2.2.1. Verimlilik.....	4
2.2.2. Etkinlik.....	5
2.2.3. Etkililik.....	7
2.3. Performans Ölçüm Yöntemleri.....	7
2.3.1. Oran analizi.....	8
2.3.2. Parametrik yöntemler	8
2.3.3. Parametrik olmayan yöntemler.....	11
3. VERİ ZARFLAMA ANALİZİ.....	14
3.1. Veri Zarflama Analizinin Tarihçesi	14
3.2. Veri Zarflama Analizi Kavramı	15

	Sayfa
3.3. Veri Zarflama Analizi Kullanım Amaçları.....	20
3.4. Veri Zarflama Analizi Modelleri.....	21
3.4.1. CCR(Charnes, Cooper, Rhodes) modelleri	22
3.4.2. BCC(Banker, Charnes, Cooper) modelleri	24
3.5. Ölçek ve Çapraz Etkinlik	28
3.6. Window Analizi.....	29
3.7. Veri Zarflama Analizinin Avantajları ve Dezavantajları.....	32
3.8. Veri Zarflama Analizinde Kullanılan Paket Programlar	35
3.8.1. Ticari veri zarflama analizi paket programları	37
3.8.2. Ticari olmayan veri zarflama analizi paket programları.....	38
4. DOĞRUSAL PROGRAMLAMA VE SİMPLEKS YÖNTEM	40
4.1. Doğrusal Programlama	40
4.2. Simpleks Yöntem	44
4.3. Simpleks Yöntemin Uygulanması.....	44
4.4. Problemin Standart Forma Getirilmesi	45
4.4.1. Kısıtlamalar.....	45
4.4.2. Değişkenler	46
4.4.3. Amaç fonksiyonu	46
4.5. Simpleks Yöntem ile Çözüm	46
4.6. Simpleks Yöntemin Akış Şeması.....	49
4.7. Büyük M Metodu	50
5. KARAR DESTEK SİSTEMİ	53
5.1. Veri Giriş Arayüzü	53
5.1.1. Manuel veri girişi	54

	Sayfa
5.1.2. Excel ile veri giriři	54
5.1.3. Not Defteri ile veri giriři	55
5.2. VZA Modellerinin Simpleks Bařlangıç Tablosu	56
5.3. Menüler	64
5.4. Analiz ve Rapor Arayüzü	65
5.4.1. Panel olmayan veriler için analiz	65
5.4.2. Panel veriler için analiz.....	67
5.5. Karar Destek Sisteminin Diğerk Programlar ile Karřılařtırılması	69
6. SONUÇ VE ÖNERİLER.....	78
KAYNAKLAR.....	80
EKLER.....	85
EK-1 Charnes ve arkadaşlarının çalışmasına (1985) ait window analizi	86
EK-2 Alp ve arkadaşlarının çalışmasına (2011) ait window analizi	87
EK-3 Örneğea ait veriler ve sonuçlar	88
EK-4 Window analizi örneğine ait veriler ve sonuçları.....	89
EK-5 İkinci örneğea ait sonuçlar	90
EK-6 Kullanılan Kodlar	92
ÖZGEÇMİř	112

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Temel başlangıç simpleks tablosu	49
Çizelge 4.2. Örnek için başlangıç simpleks tablo	52
Çizelge 4.3. Örnek için birinci simpleks tablo	52
Çizelge 4.4. Örnek için ikinci simpleks tablo (Optimal)	52
Çizelge 5.1. Girdi yönlü CCR primal model için başlangıç simpleks tablo	57
Çizelge 5.2. Girdi yönlü CCR dual model için başlangıç simpleks tablo	58
Çizelge 5.3. Çıktı yönlü CCR primal model için başlangıç simpleks tablo	59
Çizelge 5.4. Çıktı yönlü CCR dual model için başlangıç simpleks tablo	60
Çizelge 5.5. Girdi yönlü BCC primal model için başlangıç simpleks tablo	61
Çizelge 5.6. Girdi yönlü BCC dual model için başlangıç simpleks tablo	62
Çizelge 5.7. Çıktı yönlü BCC primal model için başlangıç simpleks tablo	63
Çizelge 5.8. Çıktı yönlü BCC dual model için başlangıç simpleks tablo	64
Çizelge 5.9. Örneğe ait CCR çıktı yönlü modelin EMS analiz sonuçları	74
Çizelge 5.10. Örneğe ait CCR çıktı yönlü modelin KDS analiz sonuçları	75

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 4.1. Simpleks yöntemine ait akış şeması	50

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 5.1. Veri giriş arayüzüne ait ekran görüntüsü	53
Resim 5.2. Manuel veri giriş arayüzü (Ek 3, Çizelge 3.1).....	54
Resim 5.3. Excel'deki verilere ait ekran görüntüleri (Ek 3, Çizelge 3.1).....	55
Resim 5.4. Not defteri'ndeki verilere ait ekran görüntüleri (Ek 3, Çizelge 3.1)	56
Resim 5.5. Menülere ait ekran görüntüsü	64
Resim 5.6. Dosya menüsü ve aktarma seçenekleri	64
Resim 5.7. Düzen menüsü	65
Resim 5.8. Görünüm menüsü	65
Resim 5.9. Analiz menüsü	65
Resim 5.10. Görünüm menüsü	65
Resim 5.11. Yardım menüsü	65
Resim 5.12. Analiz arayüzü	66
Resim 5.13. Rapor arayüzü (Ek 3, Çizelge 3.2)	66
Resim 5.14. Analiz sonucu etkinlik grafiği	67
Resim 5.15. Panel veriler için veri giriş arayüzü (Ek 4, Çizelge 4.1)	68
Resim 5.16. Window analizi arayüzü	68
Resim 5.17. Window analiz için rapor arayüzü (Ek 4, Çizelge 4.2)	69
Resim 5.18. EMS model seçim ekranı	69
Resim 5.19. EMS analiz sonucu	70
Resim 5.20. KDS model seçim ekranı	70
Resim 5.21. KDS analiz sonucu (Ek 3, Çizelge 3.2)	71
Resim 5.22. Örneğe ait verilerin ekran görüntüsü.....	72
Resim 5.23. Örneğe ait EMS analiz sonucu	73

Resim**Sayfa**

Resim 5.24. Örneğe ait KDS analiz sonucu (Ek 4)	73
---	----

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
BCC	Banker - Charnes - Cooper
BCC_D-I	Girdi Yönlü BCC Dual Modeli
BCC_D-O	Çıktı Yönlü BCC Dual Modeli
BCC_P-I	Girdi Yönlü BCC Primal Modeli
BCC_P-O	Çıktı Yönlü BCC Primal Modeli
CCR	Charnes - Cooper - Rhodes
CCR_D-I	Girdi Yönlü CCR Dual Modeli
CCR_D-O	Çıktı Yönlü CCR Dual Modeli
CCR_P-I	Girdi Yönlü CCR Primal Modeli
CCR_P-O	Çıktı Yönlü CCR Primal Modeli
CRS	Constant Return to Scale - Ölçeğe Göre Sabit Getiri
DEA	Data Envelopment Analysis
DFA	Distribution Free Approach – Serbest Dağılım Yaklaşımı
DMU	Decision Making Unit
DP	Doğrusal Programlama
EA	Etkinlik Analizi
FDH	Free Disposal Hull - Serbest Atılabilir Zarflama Analizi
KVB	Karar Verme Birimi
SFA	Stochastic Frontier Approach – Stokastik Sınır Yaklaşımı
TFA	Thick Frontier Approach - Kalın Sınır Yaklaşımı
VRS	Variable Return to Scale - Ölçeğe Göre Değişen Getiri
VZA	Veri Zarflama Analizi

1. GİRİŞ

İnsanlığın geçmişten beri en temel sorunu kaynakların yetersiz oluşudur. Bugün petrol, doğalgaz kaynaklarının kıtlığından bahsediliyorken, yarın su kaynaklarının kıtlığından da bahsedilmesi muhtemeldir. Kaynakların hızla artan nüfus karşısında veya bilinçsiz kullanılması sonucunda miktarı azalmaktadır. Kaynakların etkin kullanılması o kaynaklara sahip olan ülkeler açısından çok önemlidir. Türkiye İstatistik Kurumu verilerine göre 2011 yılı sonu itibari ile yıllık bazda yüzde 8,5'lik büyümeyle dünyanın en hızlı büyüyen ikinci ekonomisine sahip olan Türkiye, bu büyümesini koruyabilmek için stratejiler geliştirmelidir. Yapılan birçok çalışmada kaynakların etkin kullanımı ile ilgili yeni ölçüm yöntemleri geliştirildiği görülmektedir. Kaynakların etkin kullanımı ve en uygun şekilde dağıtılması performans analizi yöntemleri ile mümkündür.

İçinde bulunduğumuz teknoloji çağının bu hızlı değişim sürecinde; büyük kurum ve kuruluşların mükemmellik arayışları kaçınılmaz bir hale gelmiştir. İşletmelerinin başarılı olabilmeleri ve pazar rekabetinde uzun süre yaşayabilmeleri, ancak mevcut rekabet güçlerini doğru olarak ölçmeleri ve gerekli önlemleri zamanında almaları ile mümkündür. Söz konusu hedeflere erişebilmek için işletmeler, ellerindeki kaynakları en etkin bir şekilde kullanmak zorundadırlar. Bu amaçla, işletme iki temel alt sisteme gereksinim duyar. Birincisi, geçmişe yönelik faaliyetlerdeki başarının değerlendirilmesi için gerekli olan kontrol alt sistemidir. Bu alt sistem, arzulanan sonuçlara erişilip erişilmediğini ya da hangi oranda erişildiğini araştırır. İkincisi ise kontrol alt sisteminin ürettiği bilgilerin ışığında ileriye yönelik kararların alınması için gereken planlama alt sistemidir. Bu iki alt sistemin eşzamanlı olarak yürütülmesi, işletmenin rekabet gücü ya da mevcut etkenlik düzeyini artırmasına imkân sağlar. Bu nedenle, kontrol alt sistemi bünyesindeki etkenlik ölçütleri işletmelerin gelecek dönemlerdeki başarıları açısından çok yararlı göstergelerdir [Yolalan, 1990].

Günümüzde organizasyonların gelecekleri açısından, performans analizi daha önemli bir hale gelmiştir. Performans analizi, hem çalışanların görevli oldukları işlerde hem

de işletmenin diğer işletmeler arasında ne kadar etkili ve verimli olduklarını gösteren bir ölçümdür. Bu süreçte ilk olarak, performans amaçları ve standartları belirlenir. Daha sonra, çalışanın ve işletmenin performansı ölçülür ve hedeflenen standartlarla kıyaslanır. İşletmelerde performans analizi yapılarak, çalışanların başarı veya başarısızlık durumlarının belli bir süreç içinde tespit edilmesi ve bu doğrultuda çalışanların potansiyel güçlerinin belirlenip, doğacak olumsuz durumlara karşı önlemler alınması sağlanır. Performans analizi, işletmede çalışanlara adil ödül dağıtımı sağlandığı boyutta faydalı olacaktır. Nitekim bu sayede; çalışanların iş yapma istekleri artırılabilecek, ücret ve terfilerin kime verileceği belirlenecek, insan kaynaklarının çalışmaları planlanacak ve personel seçim yöntemlerinin ne derece etkili olduğuna karar verilecektir [Zerey, 2010].

Performans analizi için birçok yöntem bulunmaktadır. Ancak literatürde en çok tercih edilen yöntem olan Veri Zarflama Analizi (VZA), Karar Verme Birim(KVB)'lerine ait çeşitli girdi ve çıktı değişkenlerinin gözlem değerleri kullanılarak, göreceli etkinliğin ölçüldüğü parametrik olmayan bir yöntemdir. VZA kullanılarak yapılan bir etkinlik ölçümleme çalışması sonuçları ışığında hangi KVB'nin ne derece etkin olduğu ve etkin olmayan KVB'lerinin hangi girdi ya da çıktıyı ne derece ve ne yönde değiştirmesi gerektiği konusunda yorum yapılabilmektedir. Burada bahsedilen KVB, birbirine benzeyen girdiler yardımıyla birbirine benzeyen çıktılar üreten işletme, kurum, firma, şirket, futbol takımları gibi etkinliği incelenen birimlerdir. Performans analizi için farklı türde bilgisayar paket programları kullanılabilmektedir. Ancak yabancı dillerde hazırlanmış bu paket programlarının hedef değerlerinin oluşturulması, referans kümelerinin oluşturulması, window analizi gibi eksik yönleri bulunmaktadır. Söz konusu durumları giderebilmek ve tüm eksiklikleri kapsayan tek bir paket program olmamasından dolayı bu çalışma ortaya çıkmıştır. Geliştirilen bu paket program, diğer yazılımların kullandığı gibi tek duyarlı sisteme göre değil daha hassas olan çift duyarlı sisteme göre Microsoft Visual C# programlama dili ile hazırlanmıştır. Geliştirilen KDS bütün eksik yönleri içermemektedir ancak girdi ve çıktı yönlü hem CCR hem de BCC modellere göre analiz yapabilmektedir. Aynı zamanda yapılan analiz sonucunda ise her bir KVB'nin etkinlik değeri ve etkin olmayan KVB'nin referans olacağı KVB ve

referans olacağı yüzde oranını verebilmektedir. Aynı zamanada referans olan etkin KVB'lerinin referans olma sıklığı da analiz sonucunda gösterilmektedir. Etkinlik değerlerinin çubuk grafiği analiz sonucunda kullanıcının görsel yorum yapabilmesi için sunulmaktadır. Bu paket program, hem muadillerinden daha kullanışlı hem de yerli üretim olması sebebiyle ülkemize katma değer katması düşünülmektedir. Bu sayede özellikle sağlık, eğitim, bankacılık, sanayi, ulaşım ve tarım sektörlerinde performans konusunda farkındalık artırılacak ve Türkiye ekonomisine ülke çapında katkıda bulunabilecektir.

Bu çalışma altı bölümden oluşmakta olup bölümleri şu şekilde sınıflandırılmaktadır; Bölüm 2 de performans analizi ile ilgili verimlilik, etkinlik, etkililik gibi temel kavramların yanında performans ölçüm yöntemleri olan oran analizi parametrik ve parametrik olmayan yöntemlerden bahsedilmiştir. Bölüm 3 de ise VZA'ne ait temel bilgiler, kullanım amaçları ve VZA'ne ait modeller tanıtılmıştır. VZA modellerinin Doğrusal Programlama (DP) ile çözüldüğünden, DP ile ilgili bilgiler Bölüm 4 de verilmiştir. Paket program geliştirilirken kullanılan en temel çözüm yöntemi DP'nin bir metodu olan Simpleks Yöntemden de Bölüm 4 de bahsedilmiştir. Bölüm 5 de ise geliştirilen yazılıma ait menü ve ekran görüntüleri paylaşılmış, örnek uygulamalar gösterilmiştir. Son bölüm olan 6. Bölümde ise çalışma ile ilgili sonuçlara ve bazı önerilere yer verilmiştir.

2. PERFORMANS ANALİZİ

2.1. Performansın Tanımı

Sözlükteki karşılığı kapasite kullanım derecesi olan performans, belirli bir amaca ulaşılabilmek için yapılan bir işin sonucunda, amaçlanan nicel ve nitel hedeflere ne ölçüde ulaşılabilindiğinin bir göstergesidir [Charnes ve ark., 1978].

Performans, Tetik'in 2003 yılında yaptığı çalışmada bir işletmenin belli bir zaman diliminde elde ettiği başarı derecesi olarak veya işletmenin amaçlanan hedefe yönelik nereye varabileceğinin nicel ve nitel anlatımı olarak tanımlanmaktadır. Birer performans boyutu olan etkinlik ve verimlilik kavramları kâr amacı gütsün ya da gütmesin bütün birimlerin varlıklarını sürdürebilmeleri için yaşamsal öneme sahiptir. Birimler, benzer birimler içerisinde nerede olduklarını, üstün yönlerini ve zayıf yönlerini görebilmeleri için ölçülebilir verilerle periyodik olarak performans ölçümü yapmalıdırlar [Ertuğrul, 2010].

2.2. Performans Analizi ile İlgili Temel Kavramlar

2.2.1. Verimlilik

Performans ölçütlerinden biri olan ve yaygın olarak kullanılan verimlilik (productivity) kavramının çoğu kez etkinlik kavramından farkı anlaşılmadan, yanlış kullanıldığı görülmektedir. Verimlilik veya diğer adıyla üretkenlik veya prodüktivite kavramı, kullanılan kaynaklar ile elde edilen çıktı miktarı arasındaki ilişki olarak tanımlanır ve basit olarak bir kurumun amacına uygun olarak oluşturdukları ürünün, bu ürünü ortaya koyabilmek için harcadığı kaynağa oranlanmasıyla hesaplanır [Yolalan, 1993]. Ancak, girdi ve çıktılardaki niteliksel farklılıklar bu hesaplamayı zorlaştırmaktadır. Özellikle günümüzde kullanılan birbirinden farklı kaynaklar ve bunların sonucunda elde edilen birçok farklı ürün etkinliğinin değerlendirilmesini güçleştirmektedir. Ayrıca, bunların yanında girdi ve çıktı birimlerinin farklı olması

da karşılaşılan zorlukları artırmaktadır. Bir başka deyişle verimlilik, kaynakların ne ölçüde etkin ve etkili kullanıldığının bir ölçüsüdür.

Verimliliğin yükselmesi demek, elde mevcut olan emek, sermaye ve toprak gibi kaynaklardan eskisine göre daha çok ürün elde etmek demektir. Verimlilik artışları bireylerin yaşam koşullarını derinden etkileyen sayılı değişkenlerden biridir. Genel olarak toplumu oluşturan bireylerden hiçbirininkini azaltmadan en azından bir kişinin gelirini yükseltebilmek, yalnızca verimlilik artışlarına bağlıdır [Yapıcı, 2002].

2.2.2. Etkinlik

Verimlilik ve etkinlik çoğu zaman aynı anlamda kullanılmakla birlikte; etkinlik, sektördeki mevcut teknoloji ile firmaların ne kadar iyi bir performansa sahip olduğunu belirtirken, verimlilik ise kullanılan teknolojinin zaman içindeki evrimini gösterir. Etkinlik, eldeki üretim kaynakları veya girdilerin ne derecede iyi kullanılarak çıktı üretilebileceğini gösterir. Yararlı çıktıların üretilmesi için kullanılan işçilik, hammadde ve malzeme, dışarıdan sağlanan fayda ve hizmetler gibi kaynakların ne kadar etkin kullanıldığını anlatan bir kavramdır. Başka bir deyişle üretim biriminin elde edebileceği maksimum verimliliktir [Boussofiane, 1991]. Etkinlik verimliliğin tamamlayıcı bir unsurudur. Etkinlik çıktılarla ilgilenirken, verimlilik girdilerle ilgilenmektedir. Diğer bir deyişle etkinlik amaç ve hedeflerle ilgilenirken, verimlilik en düşük kaynak tüketimi ile ilgilenmektedir. Bir işletmede var olan tüm imkânlar içindeki erişilebilir en yüksek kuramsal etkinlik düzeyi 1'e eşittir. Bu düzeyde birim maliyetler en düşük, üretim miktarı (çıktı) ise en yüksek düzeydedir. Bu durumda işletme hem etkindir hem de verimlidir. Sonuç olarak, etkinlik sağlanamadan yüksek verimlilik düzeyine ulaşmak mümkün değildir. Etkinlik, en temel olarak çıktının girdiye oranıdır [Kulaksızoğlu, 2011].

Etkinlik analizinde çeşitli etkinlik türleri söz konusudur. Bunlar teknik etkinlik, ölçek etkinliği ve tahsis(fiyat) etkinliğidir [Kılıçkaplan ve Karpaz, 2004].

Teknik etkinlik

Bir üretim sürecinde yer alan KVB'nin girdilerini sabit tutarak mümkün olan maksimum çıktılar elde edilmesindeki başarıyı göstermektedir. KVB'nin girdileri sabit tutularak çıktılar daha fazla artırılmıyorsa, KVB teknik olarak etkindir [Güneş, 2006].

Ölçek etkinlik

Ölçek büyümesi sonucu organizasyonda birim başına ortalama maliyetteki artışı (ya da azalışı) saptamak için kullanılan performans değerlendirme ölçütüdür. Uygun ölçekte üretim yapmadaki başarı da ölçek etkinliği olarak tanımlanmaktadır. Teknik etkinlik ile ölçek etkinliğin çarpımı ise toplam etkinliği vermektedir.

Tahsis (Fiyat) etkinlik

Tahsis etkinliği veya literatürdeki diğer adıyla fiyat etkinliği, şirketin davranışsal olarak amacına ulaşması yani kârını maksimize edebilmesi için teknik etkinlik ve ölçek etkinliğini sağlaması gerekliliğinin yanında bir de girdi fiyatlarını minimize etme zorunluluğu vardır. İşte bu noktada tahsis etkinliği üçüncü bir etkinlik unsuru olarak ortaya çıkmaktadır. Diğer bir ifade ile tahsis etkinliği girdi ve çıktı fiyatları dikkate alınarak, üretim maliyetini minimum yapacak optimal girdi bileşimini seçme öngörüsünü kazanabilmektir.

Tahsis etkinliği, girdiye ve çıktıya yönelik olmak üzere iki grupta incelenebilir.

Girdiye yönelik tahsis etkinliği; işletmedeki girdi fiyatları dikkate alınarak en uygun girdi kombinasyonunu seçebilme öngörüsü olarak ifade edilebilir. İstenilen sürece ait girdi fiyatları belirli iken, KVB'leri ekonomik olmayan bir girdi kombinasyonu seçtiğinde, bu tercihin getireceği maliyet yükü tahsis etkinliği ile değerlendirilir.

Çıktıya yönelik tahsis etkinliği ise; çıktı fiyatları göz önüne alınarak, KVB'nin kazanacağı geliri maksimum yapacak çıktı kombinasyonunu seçebilme öngörüsüdür. O dönemdeki çıktı fiyatlarına bakıldığında, KVB'nin uygunsuz çıktı bileşimini tercih etmesinin sebep olacağı gelir kaybı da çıktıya yönelik tahsis etkinliği ile değerlendirilir [Zerey, 2010].

2.2.3. Etkililik

Etkililik (effectiveness) bir KVB'nin önemli bir hedef, amaç ya da misyonu başarabilme yeteneğidir. Diğer bir ifadeyle, yapılan faaliyetler neticesinde temin edilen çıktı ile istenilen çıktı düzeyi arasında yapılan karşılaştırmadır. Buradan da anlaşılabileceği gibi etkililik, firma ya da işletmelerin girdileriyle değil, yalnız çıktılarıyla ilgilidir. Bir başka ifadeyle etkililik, toplam performans göstergesidir. Toplam performans, yöneticilerin ve çalışanların bilgi ve becerileri, teknoloji kapasitesi, kullanılan yöntemler ve hatta çevre ilişkileri gibi çok çeşitli etmenlerin birbiri ile ilişkileri sonunda oluşur. Eğer KVB doğru amaçlarla çalışmıyorsa, doğru olan ya da yapması gereken işleri yapmıyorsa etkili değildir ve ne kadar etkin olursa olsun sonuçta başarısız sayılabilir Bir işin verimli sonuçlandırılması, o işin etkili olduğunu göstermez. Etkililik, gerçekleşen sonucun, beklenen sonuca oranı olarak formüle edilir [Akal, 2005].

2.3. Performans Ölçüm Yöntemleri

Performans ölçümü, ülkemizde özellikle özel sektörde kullanılan önemli yöntemlerden bir tanesidir. Kamu da ise 2006 yılında çıkan Stratejik Planlama kanuna göre zorunlu hal alan performans ölçümü kamu kuruluşlarında da yapılmaya başlanmıştır. Performans ölçümü, kurumun misyonuna ve vizyonuna uygun davranış sergileyip sergileyemediği bilgisini sunarken, aynı zamanda personelin ücret artışı, terfi edilmesi ya da eğitimi gibi şirket misyonuna uygun kararlar alırken faydalı bilgiler sunmaktadır. Performans ölçümü ile işletme etkin olmayan kaynaklarını görme şansına sahip olacak ve bu şekilde performansını artırma yoluna gidecektir. Ayrıca bu sayede, bulunduğu sektörde rakip işletmeler arasındaki yerini belirleyecek

ve eldeki imkânlarla ne yapması gerektiğine daha kolay karar verebilecektir. Bu amaçlara yönelik gerçekleştirilen performans ölçümü; oran analizi, parametrik yöntemler ve parametrik olmayan yöntemler olmak üzere 3 farklı yöntemle gerçekleştirilir.

2.3.1. Oran analizi

Oran analizi; tek girdi ve tek çıktıya dayalı olduğundan dar kapsamlı olup, çok fazla bilgi gerektirmediğinden sıkça tercih edilen basit bir yöntemdir. Bu yöntemde, her oran diğer performans göstergeleri saf dışı bırakılarak, bir girdi ve bir çıktıyla dikkate alınır. Dolayısıyla, bu yaklaşımla birden fazla çıktı ve girdinin bulunduğu durumlarda netice alınamayacağı gibi, girdi ve çıktıların tümünün aynı birimle incelenemediği hallerde, girdi ve çıktılar ayrı ayrı ele alınır [Kıllı, 2004]. Kolay hesaplanmasına rağmen bu yöntemin birçok zayıf yönü de vardır. Oran analizinin zayıf yönlerinden bir tanesi de mutlaka bir şeylerle karşılaştırılmaya gereksinim duymalarıdır. Örneğin, oranla performans ölçümü yapılan bir işletmedeki sayısal sonuçlar, ya kendi içeriğindekiyle ya da diğer işletmelerin benzer değerleri ile ilişkilendirilirler. Oran analizindeki oranlama, göreceli de olsa en iyiye göre değil, var olan değerlerin birbirlerine bölümüyle elde edilir. Bu ise, bir performans iyileştirilmesine yönelik bir teknik değil, yalnızca bir durum belirlemesidir. Oran analizi ile yapılan ölçümlerde bazı oranlar, işletme son derece başarılı gösterirken bazı oranlar da işletmeyi oldukça başarısız gösterebilmektedir. Bu olumsuzluğun giderilebilmesi için, tekil oranların tek boyutluluğunu dengeleyen “genişletilmiş oran kümeleri” geliştirilmiş ise de bunlar da tek boyutlu yapıdan kurtulamamıştır. Ayrıca durgun bir yapıya da sahip olduğundan, işletmelerin sadece belli bir zaman dilimindeki performanslarına dayalıdır.

2.3.2. Parametrik yöntemler

Oran analizinin sakıncalarını azaltabilme düşüncesi ile parametrik yöntemlere başvurulmaktadır. Parametrik yöntemler, birçok girdi ile tek bir çıktı arasında nedensellik ilişkisi kuran ve parametrelili bir yöntem olan regresyon analizinde olduğu

gibi; uygulama yapılacak endüstri dalına göre üretim fonksiyonunun analitik bir yapıya sahip olduğu varsayımı yapılır ve bu fonksiyonun parametrelerinin belirlenmesine çalışılır. Yani oran analizleri ile tek bir girdi ile tek bir çıktı arasındaki ilişki izlenebilirken regresyon analizinde birden çok girdinin tek bir çıktı üzerindeki etkisini ölçmek mümkün olmaktadır. Regresyon analizi yapılabilmesi için bir gözlem kümesi gereklidir. Bu gözlem kümesinden hareket ile bir regresyon çizgisi oluşturulmaktadır. Oluşturulan bu çizgiye, etkinlik sınırı (efficiency-frontier) denir. Etkinlik sınırının üstünde kalan birimler etkin, dışında kalan birimlerse etkin olmayan olarak değerlendirilir. Ancak, hiçbir gözlemin tam olarak uyuşmadığı bir etkinlik sınırının oluşması da mümkündür. Bu durumda, parametrik yöntemlerde etkin olmayan veya/ve rassal hataya sahip gözlemlerin hata dağılımlarının da araştırılması gerekecektir. Bir gözlemin etkin olduğuna ise, ancak ölçüm hatalarının giderilmesinden sonra karar verilmektedir. Ayrıca artık değerler pozitif çıktığında birimlerin etkin, negatif çıktığında ise birimlerin etkin olmadığı kanısına varılır. İki den fazla değişkenle değerlendirme yapabilme bakımından oran analizine göre daha kapsamlı ve daha gerçekçi olan regresyon tekniğiyle ölçüm yapmanın da temelde üç sakıncası vardır.

Bunlardan birincisi; çıktıların ölçü birimleri farklı olduğunda birtakım zorluklarla karşılaşılmasıdır. İkincisi ise, bu metotta kullanılan en küçük kareler yönteminde çok sağlıklı olmayan referans çevresinin seçimi sonucu ortaya çıkan görecelilikten kaynaklanır. Parametrik yöntemlerde, ölçüm hatalarının kolayca saptanmasına imkan veren rassal hataların var olması, bu yöntemi diğerlerinden daha avantajlı hale getirir. Bunun yanında, rassal hata ve etkin olmamanın ayırt edilmesi de zordur. Parametrik yöntemler bu ayrımı yapmak için kullandıkları dağılım varsayımıyla birbirinden ayrılırlar. Böylece parametrik yöntemlerde etkinlik sınırından sapmaların, rassal hata ve etkin olmama gibi iki unsurdan oluştuğu ve bu iki bileşenin birbirinden ayırt edilebilmesinin büyük önem taşıdığı ortaya çıkar. İşte bu ayrımı yapmak için kullandıkları varsayım dağılımlarına göre üçe ayrılırlar. Bunları şöyle sıralayabiliriz: Stokastik sınır yaklaşımı, serbest dağılım yaklaşımı ve kalın sınır yaklaşımı.

Stokastik sınır yaklaşımı (Stochastic frontier approach - SFA)

Ekonometrik bir yaklaşım olan SFA modelinde üretim, kar ve maliyet gibi açıklanan değişkenler, girdi, çıktı ve çevresel faktör gibi açıklayıcı değişkenlerle birlikte belli bir hata payı çerçevesinde ilişki kurar. Bu yaklaşımda rassal hatanın normal, gözlemlenen değerlerle varsayılan değerler arasındaki farkı oluşturan etkinsiz gözlemlerin ise asimetrik dağılım gösterdiği varsayılmaktadır. Fakat çeşitli çalışmalarda bu varsayımlara uymayan durumların söz konusu olduğu görülmektedir. Yanlış varsayımlarla çalışmak ise çalışmanın güvenilirliği üzerindeki en büyük problemidir.

Serbest dağılım yaklaşımı (Distribution free approach - DFA)

SFA yönteminin varsayımları ile ilgili getirilen eleştirilere karşılık olarak, yeni bir sınır yaklaşım yöntemi geliştirilmiştir. DFA yaklaşımı, stokastik modeldeki değişken ve rassal hata ile ilgili yapılan dağılım varsayımları yerine bu dağılımların herhangi bir dağılım olabileceğini benimsemektedir. Ancak panel verinin varlığı altında kullanılabilen DFA yönteminde her birimin uzun vadede etkinliği sabittir en azından istikrarlıdır ve ölçüm hataları da sıfıra yakınsar. Bu varsayımlar etkinsizliğin pozitif olması şartıyla gerçekleşebilir [Berger ve Humphrey, 1997].

Kalın sınır yaklaşımı (Thick frontier approach - TFA)

Berger ve Humphrey (1992), kalın sınır yaklaşımı olarak isimlendirdikleri bu yöntemde, gözlenen ve beklenen değerler arasındaki en küçük ve en büyük farkın rassal hatayı, geri kalanının ise, etkin olmayan gözlemlerin oluşturduğunu varsaymaktadır [Miller ve ark., 2005]. Elde edilen en küçük ve en büyük değerler kullanılarak ortalama bir fonksiyon oluşturulmaktadır. Rassal hata olarak kabul edilen değerler dışındaki gözlem değerleri, etkin olmayan kabul edilmektedir.

Yukarıdaki açıklamalardan da anlaşıldığı üzere; TFA, tek bir üretim biriminin etkinliğinin tahmininde kullanılmaya elverişli olup, genel etkinlik düzeyinin

hesaplanmasında kullanılamamaktadır. Yukarıda açıklanan her üç yöntemden SFA ve DFA'nın, TFA'ya göre daha sık kullanılmasına karşılık, her üç yöntemde de birden fazla açıklayıcı değişken kullanabilmesine rağmen, ancak bir tane açıklanan değişken kullanmaktadır. Bu da birden fazla çıktının olduğu sistemler için bu yöntemleri kullanışsız hale getirmektedir. Aynı zamanda yöntemler, girdiler ve çıktılar arasındaki işlevsel bir ilişkinin kurulması için bazı varsayımları kabul etmektedir. Bu varsayımlardan herhangi birinin geçersiz ya da yanlış olduğu durumlarda, modellerin de geçersiz olacağı açıktır [Lorcu, 2008].

2.3.3. Parametrik olmayan yöntemler

Oran analizi ve parametrik yöntemlere göre daha esnek olan parametrik olmayan yöntemler, üretim fonksiyonunun ardında analitik bir formun varlığına gerek duymamaktadır. Bu yöntem doğrusal programlama kökenli teknikler kullanarak hesaplama sonucunda elde edilen etkinlik değerinin etkinlik sınırına olan uzaklığını ölçer. Parametrik olmayan yöntemlerin veri temininin diğer yöntemlere göre daha detaylı olması, uygulamasının da detaylı olmasına neden olmaktadır. İçeriğinde sahip olduğu çok girdi ve çok çıktı oranlaması da parametrik olmayan yöntemlerin performans ölçümüne tam anlamıyla uygun olduklarını ifade etmektedir. Farklı çözüm teknikleri kullanılan bu yöntemlerde detaylı bilgi verdikleri ve uygulaması da detaylı bilgilerin varlığına rağmen basit olduğu için genel olarak performans ölçümlerinde parametrik olmayan yöntemler tercih edilmektedir. Ölçütlerin girdi ve çıktıdan bağımsız olduğu parametrik olmayan yöntemlerde, işletme farklı açılardan eş zamanlı ölçülebilmesine imkân sağlamaktadırlar. Bu ölçütler her bir KVB için göreceli etkinliği hesaplarken amaç fonksiyonlarını ayrı ayrı en iyiler ve her bir KVB için en uygun amaç kümesini belirlerler. Oysa parametrik yöntemler endüstri grubunun tümünü göz önüne alıp ortalama etkinliğe göre ölçüm yapmaktadırlar. Bu yöntemlerde, gözlem kümesi etkin olan ve olmayan seklinde iki ana gruba ayrılır ve etkin olmayan her bir birimin etkin hale dönüştürülebilmesi için ne gibi önlemler alınması gerektiğine ilişkin önemli bilgiler türeterek yöneticilere yol gösterirler.

Ancak her yöntemin olduğu gibi bu yöntemin de bazı kısıtları olduğu unutulmamalıdır. Bunların başında, parametrik olmayan etkinlik ölçütlerinin veri tabanına karşı son derece hassas olmalarıdır. Bu nedenle etkinlik ölçümlerinde kullanılan diğer istatistiki yöntemlerde olduğu gibi girdi ve çıktı verilerinin olabilecek hatalardan arındırılması için özen gösterilmelidir. Ayrıca, seçilen girdi ve çıktı bileşenlerinin üretim dönüşümünü iyi bir şekilde temsil edemediği durumlarda etkinlik ölçümü başarısız olmaktadır [Besen, 1994].

Parametrik olmayan yöntemlerden en çok kullanılanı 1978 yılında Charnes, Cooper ve Rhodes üçlüsü tarafından oluşturulan VZA yöntemidir. Ayrıca VZA'nın özel bir hali olarak geliştirilen en çok bankacılık ve finans alanında kullanılan Serbest Atılabilir Zarflama Analizi (Free Disposal Hull-FDH) de, parametrik olmayan bir başka yöntemdir. VZA ileride ayrıca açıklanacağından burada yalnızca Serbest Atılabilir Bölge yöntemi hakkında kısaca bilgi verilecektir.

Serbest atılabilir bölge (Free disposal hull - FDH)

Serbest Atılabilir Bölge (FDH) kavramı ilk defa Deprins, Simar ve Tulken tarafından geliştirilmiştir. Serbest Atılabilir Bölge, VZA yönteminin özel bir uygulama şekli olarak tanımlanmaktadır. FDH analizi, diğer yöntemler gibi performans karşılaştırmaları yolu ve en iyi pratikleri yansıtan bir üretim sınırı ile üreticilerin etkinlik sıralamaları için bir çerçeve sağlamaktadır. FDH analizinin birinci aşaması, her bir düzeydeki girdi değişkeni kullanımı ve örnekteki üreticiler arasında gözlemlenen en yüksek çıktı düzeyini içeren bir üretim sınırı belirlemektir. Bu üretim sınırı bir kez oluşturulduğunda, etkin olmayan üreticilerin sıralamaları da belirlenebilir. Etkin olmayan üreticiler, belirli bir girdi seviyesi ile üretilebilecek mümkün çıktıdan daha az üreten kurumlardır. FDH ve VZA orijinal olarak girdileri çıktılarına dönüştüren KVB'ler için oluşturulmuş ve kullanılmışlardır. FDH analizinin diğer üretim sınırı yaklaşımlarına göre avantajı, diğerlerine oranla üretim sınırını oluşturmada daha cimri bir yaklaşım sergilemesidir. Temel yaklaşım, girdi ve çıktılar serbestçe elden çıkartıldığında, aynı üretim teknolojisi bulunması varsayımı ile devam eden bir üretim sınırı herhangi bir girdi sayesinde mümkün olan en yüksek

çıktıyı verecektir. Ayrıca FDH analizi yalnızca etkin olan ve olmayan birimlerin ayırt edilmesinde kullanılabilir. Amaç etkinlik karşılaştırması yapmak ise, bu yaklaşımın oldukça yetersiz kaldığı açıktır. Ayrıca başka bir zayıf noktası da, ele alınan gözlemlere göre bir üretim fonksiyonu oluşturarak gözlemleri sadece bu fonksiyonda sıralamasıdır. Böyle bir sınırlandırma, aynı girdi seviyesinde daha yüksek bir çıktı sağlayabilen ya da aynı çıktı seviyesini daha düşük bir girdi seviyesi ile sağlayabilen bir firmayı fonksiyonun dışında bıraktığı için analiz sonuçları iyileştirme amaçlı olarak da kullanılamayacaktır [İnan, 2000]. Yani bu yaklaşım aynı girdiyi kullanarak aynı çıktıyı üreten homojen birimlerin etkinliklerini değerlendirmekte ve her birimi en verimli birim veya birimlerle karşılaştırmaktadır. Bu bakımdan diğer yaklaşımlara göre etkinlik ölçümünde homojen küme anlayışına sahip VZA'nın kullanılması daha uygun görülmektedir.

3. VERİ ZARFLAMA ANALİZİ

Literatürde Data Envelopment Analysis (DEA) olarak geçen görelî etkinlik ölçümü, Türkiye’de Veri Zarflama Analizi (VZA) veya Etkinlik Analizi (EA) olarak bilinmektedir. Önceki bölümde etkinlik çıktının girdiye oranı olarak ifade edilmiştir. Fakat işletmeler bir girdi yerine personel sayısı, ücretler, çalışma saatleri, reklâm bütçesi gibi birçok girdi, bir çıktı yerine karlılık, pazar payı, büyüme hızı gibi de birçok çıktı kullanır. Yöneticiler için, kullanılan birçok girdinin sonucu, bu girdilerin dönüştürülmesi suretiyle elde edilen çıktılarından; dönüştürme işleminde “hangi birimlerin etkinliğinin düşük olduğunu tespit etmek” oldukça güçtür. Bu noktada VZA, yöneticilere görelî etkinlik belirlemede yardımcı olmaktadır.

3.1. Veri Zarflama Analizinin Tarihçesi

VZA’nın tarihi Edwardo Rhodes’in Carneige Mellon Üniversitesi Kamu İşleri dalında yaptığı doktora tezi ile başlamaktadır. William. W. Cooper danışmanlığında Edwardo Rhodes, federal hükümetin desteğiyle Amerika’da devlet okullarına devam eden (çoğunlukla siyah ve İspanyol) dezavantajlı öğrenciler için olan Follow Through eğitim programını değerlendiriyordu. Çalışma, bu programa katılan ve katılmayan okul gruplarının performanslarını karşılaştırmayı içermektedir. Bu program, “kendine güveni yükselen dezavantajlı çocuk” olarak belirlenen çıktı değişkeni ve “annenin çocuğuna kitap okurken harcadığı zaman” olarak belirlenen girdi değişkeni ile okulların dönemsel performanslarını ölçüyordu. Bu esnada Farrell’in 1957 yılında “Journal of the Royal Statistical Society” isimli dergide yayınlanan “The Measurement of Productive Efficiency” adlı makalesi Cooper’ın dikkatini çekti [Cooper ve ark., 2000, Sarıca, 2007]. Birden çok girdisi ve tek bir çıktısı olan birimlerin etkinliklerinin incelendiği bu çalışmada Farrell ilk kez etkinlik ölçümünde doğrusal programlamayı kullanmıştır. Ancak Cooper ve arkadaşları gerçekleştirdikleri çalışmalarında, girdi ve çıktıların birden çok sayıda olması sebebiyle Farrell yaklaşımını yetersiz bulmuşlardır [Güneş, 2006]. Farrell (1958)’in çalışmasından 20 yıl sonra Charnes ve arkadaşları (1978) onun fikirlerinin üzerine çok girdi ve çok çıktı durumunu tatmin edici şekilde değerlendiren VZA adlı güçlü

bir yöntem geliştirmişlerdir. VZA'yı ilk duyuran çalışma "European Journal of Operations Research" dergisinde 1978 yılında yayınlanmıştır [Sarica, 2007].

Bu çalışmada 70 tane okulun göreceli teknik etkinliğini, fiyatları göz ardı ederek çoklu girdi ve çıktıların tahmin etme isteği, CCR (Charnes, Cooper, Rhodes) formülasyonu olarak bilinen VZA orantısal formülasyonunu doğurmuştur. Etkin sınırın bulunması ve etkin sınır içinde kalan etkin olmayan noktaların merkeze olan radyal uzaklıklarının hesaplanması problemi Charnes, Cooper ve Rhodes tarafından parametrik olmayan programlama tabanlı olarak çözülmüştür. İncelemeye alınan KVB'leri arasında verimli olanların oluşturduğu doğrusal etkinlik eğrisi, tüm gözlemlerin etrafını sarıp bir "zarf" oluşturmaktadır [Hollingsworth, 2003]. Bu zarf eğrisine göre tüm gözlemlerin etkinlik skorları bulunmaktadır. İşte bu sebeple bir matematiksel programlama tekniği olan bu analize, Veri Zarflama Analizi adı verilmiştir. İlk önceleri ölçeğe göre sabit getiri varsayımı altında sadece teknik etkinliğin ölçümünde kullanılan VZA yaklaşımı daha sonra yapılan bazı değişikliklerle; Banker, Charnes ve Cooper (1984) tarafından ölçek etkinliğinin ölçülmesinde de kullanılmaya başlanmıştır. Banker, Charnes ve Cooper ayrıca ölçek getirisi ve teknik etkinliğin tahmini için CCR'ın lineer programlama formülasyonunda bir değişikliği öngörmüşlerdir. Banker (1984) daha çok verimli ölçek kavramını geliştirmiş ve CCR doğrusal programlama formülünün, ölçek getirisinin tahmini için nasıl kullanılabileceğini göstermiştir.

3.2. Veri Zarflama Analizi Kavramı

VZA, doğrusal programlama ilkelerine dayanan, aynı amaç ve hedeflere sahip, aynı tür girdileri kullanarak aynı tür çıktıları üreten KVB'lerinin, birbirlerine göre göreceli etkinliklerini ölçen parametrik olmayan bir yöntemdir. VZA'da bir KVB'nin göreceli etkinliği, toplam ağırlıklı çıktıların toplam ağırlıklı girdilere oranı olarak tanımlanabilir. Bu bağlamda ilk soru, söz konusu ağırlıkların farklı birimler taşıyan girdi ve çıktıların nasıl verilmesi gerektiğidir. VZA'nın özü burada yatmaktadır. VZA, her KVB için girdi ve çıktıların ağırlıklandırılma şansını vermektedir. Bu ağırlıklandırma bazı kısıtlar altında yapılmaktadır. Bu kısıtlar iki grupta toplanabilir.

İlk olarak, hiç bir ağırlık negatif değer taşıyamaz. İkinci olarak, KVB'nin ağırlıkları o şekilde seçilmelidir ki, bu ağırlıklar, analize konu olan diğer KVB'lerine de uygulandığı zaman, hiç bir KVB'nin etkinliği % 100' den fazla bulunmasın. Böylece, KVB'leri için seçilebilecek birçok ağırlık kümesi bulunmaktadır. Aslında, tüm birimlerin seçim yaptıkları ağırlık kümesi ortak amaçları doğrultusunda olacaktır. Bu bakımdan, VZA, her KVB'nin kendi etkinlik skorunu maksimize edecek şekilde girdi ve çıktı ağırlıklarını seçeceğini varsaymaktadır. KVB'leri farklı çıktıları oluşturmak için farklı girdi bileşenlerini oluşturacaklarından, ağırlıkların bu çeşitliliği yansıtacak şekilde seçilmesi beklenebilir. Genellikle KVB'leri en az kullandıkları girdilere ve en çok ürettikleri çıktıları en yüksek ağırlıkları vermek eğiliminde olacaklardır. Burada dikkat edilmesi gereken nokta, ağırlıkların fiyat ile ilişkili olmasının gerekmediği, fakat onların KVB'nin etkinliğini maksimize edecek karar değişkenlerini oluşturduğudur. Her bir KVB için kurulacak olan ve karar değişkenleri olarak girdi ve çıktı ağırlıklarını kabul eden kesirli matematiksel program, kolaylıkla denk bir doğrusal programlama modeline dönüştürülebilir. Bu model ise “Simpleks Algoritması” yardımıyla çözülebilir. Çözüm, modelin ilgili olduğu KVB için ağırlıkların değerlerini ve bu KVB'nin göreceli etkinliğini verecektir. Analizde seçilen girdi sayısı m ve çıktı sayısı s ise en az $m + s + 1$ tane KVB'nin alınması, araştırmanın güvenilirliği açısından önemli bir kısıttır. Diğer bir kısıt ise araştırma kapsamına alınan KVB sayısının, toplam değişken sayısının en az üç katı olmasıdır [Küçükşimşek, 2004].

VZA tekniğinde KVB'leri, gözlem grubu içerisinde optimal girdi bileşimiyle elde edilebilecek en yüksek üretim miktarlarının oluşturduğu teorik sınır yani etkinlik sınırında yer alan KVB'lere göre değerlendirilmektedir. KVB'nin bir çıktı setini oluşturmak için kaynakları ne kadar etkin kullandığını ölçmeyi amaçlayan VZA' da en iyi performans gösteren KVB'nin etkinlik skoru % 100 iken diğer KVB'lerin etkinlik skoru 0–100 arasında değişmektedir.

VZA yöntemi, birden fazla girdili ve birden fazla çıktı KVB'leri fonksiyonel bir yapıya gerek duymadan aynı anda değerlendirebilir ve etkin KVB'leri etkin olmayan KVB'lerinden ayırabilir. Ayrıca etkin KVB'leri içerisinden etkin olmayan KVB'ler

için referans kümeleri belirleyebilmektedir. Dolayısıyla VZA yönteminde etkinlik sınırı varsayılan bir durum değil, gerçekleşen bir durumdur. Eğer bir KVB etkin değilse, VZA yöntemi bu birimin etkinliğini artırabilmek için gerekli olan stratejileri etkin KVB'lerine referans vererek önerir. Bu bilgiler ışığında işletme yönetimi, etkin olmayan KVB'nin hangi girdileri gereğinden ne kadar fazla kullandığı, hangi çıktılar açısından ne ölçüde yetersiz üretim yaptığı ve etkin olması için ne yapması gerektiği hakkında değerlendirme yapabilir. VZA başlangıçta kâr amaçlı olmayan, kamu sektörü gibi kuruluşların karşılaştırılmalı verimliliklerinin ölçülmesini amaçlamışken, sonraları kâr amaçlı üretim ve hizmet sektöründe de yaygın bir şekilde kullanılmıştır [Kutlar ve ark., 2004]. VZA'nın uygulama bulunduğu alanlar; hastaneler, sigorta şirketleri, hava ve deniz yolu taşımacılığı, eczaneler, postaneler, bankacılık, eğitim, spor, reklâm, fast food restoranlar, şehirler, bölge ve ülkelerin kıyaslanmasında ve daha birçok homojen gruplarda kullanılmıştır. Türkiye'de VZA, bankacılık, sigortacılık, sağlık ve eğitim alanlarında bulmuş olduğu uygulama fırsatları ile yayınlanan makale ve tez sayısındaki artışla bir geçiş dönemi yaşamış ve ilgi görmeye başlamıştır. Bu ilgi bilgisayar teknolojilerinin hızla gelişimi, her alanda üretilen yeni yazılımlar VZA tekniğine de çözüm kolaylıkları sağlamıştır.

VZA'nın uygulama aşamalarını aşağıdaki gibi özetlenebilir:

Öncelikle etkinlik ölçümü yapılacak KVB seçilecektir. Bu KVB'nin birbirleriyle kıyaslanabilir ve homojen olması gereklidir. Belirlenen KVB sayısı, girdi ve çıktı sayılarının toplamından bir fazla olması gerekmektedir. İkinci bir kısıt da KVB sayısının, değişken sayısının en az 3 katı olmasıdır. KVB'nin nitelik ve nicelik olarak tespit edilmesinden sonra, roller belirlenerek, buna uygun amaçlar anlaşılır bir şekilde ifade edilmelidir. Daha sonra KVB'nin üretim teknolojisi doğrultusunda girdi ve çıktılar tanımlanır. Girdi ve çıktı değişkenleri belirlendikten sonra veriler toplanmalı, kodlandırılmalı ve hatalar saf dışı bırakılmalıdır. Ölçüm yapacak kişi duruma uygun olan VZA modelini girdi ya da çıktıya yönelik seçmelidir, bu modele göre de analiz yapılmalıdır. Son olarak, KVB'nde etkin olan ve olmayanlarla ilgili ortak noktalar bulunarak, içinde bulunulan endüstri dalına göre değerlendirmeler yapılmalıdır. Doğrusal programlama vasıtasıyla elde edilen değerler yorumlanarak,

etkin olmayan KVB'lerin nasıl etkin hale getirilebileceği konusunda kısa veya uzun vadeli planlar yapılmalıdır.

VZA yönteminin daha kolay anlaşılabilmesi için bazı kavramların açıklanmasının yararlı olacağı düşünülmüştür. Bu yüzden aşağıda VZA literatüründe en çok karşılaşılan terimlerin açıklanmıştır.

Girdi ve Çıktı: Bir organizasyonda, hizmetlerin yerine getirilebilmesi için; personel, tüketilebilen kaynak, sermaye, nakit gibi üretim sürecine sokulan faktörlerin her birine girdi; bu üretim sürecinin sonunda elde edilen ürün veya hizmete ise çıktı denir [Aslankaraoğlu, 2006].

Karar Verme Birimi: VZA'nde birbirine benzeyen girdiler yardımı ile birbirine benzeyen çıktılar üreten işletme, kurum, firma, şirket gibi göreceli etkinliği incelenen organizasyonel birimlere denir [Aydemir, 2002].

Ölçeğe Göre Sabit Getiri: Eğer bir KVB'nin girdisindeki artış çıktısında eşit bir artışa denk geliyorsa bu ölçeğe göre sabit getiridir. Bu demektir ki KVB'nin ölçeği ne olursa olsun verimliliği değişmez.

Ölçeğe Göre Azalan Getiri: Girdiler artarken çıktılarının aynı oranda artmaması hatta azalması durumudur.

Ölçeğe Göre Artan Getiri: Girdiler artarken çıktılarının daha fazla artması durumudur.

Kontrol Edilebilen Girdiler (Controlled-discretionary-Inputs): Bir kontrol edilmiş girdinin, KVB'nin idaresi üzerinde kontrolü vardır. Bunun sonucunda kullanılmış olan miktarı KVB yönetimi belirleyebilir.

Maliyet Etkinliği (Cost Efficiency): Maliyet verimliliği (ekonomik verimlilik) minimum maliyetin gerçek (gözlemlenmiş) maliyete oranıdır.

Etkin/Etkinlik Sınırı (Efficient/Efficiency Frontier): Etkinlik sınırı, en iyi performansı temsil eden ve girdi ve çıktıları en verimli şekilde birbirine dönüştüren veri kümesindeki KVB'lerinden oluşan sınırdır. Sınırı belirleyen KVB'leri %100 verimliliğe sahiptirler. Sınırdan olmayan herhangi bir KVB %100'ün altında bir verimliliğe sahiptir.

Etkinlik Değeri (Efficiency Score): VZA, her KVB için bir etkinlik değeri üretir. Bu skor 0 ve 1 arasındadır. %100 skora sahip KVB etkindir. %100'den daha düşük değer alan birimler ise etkin değildir.

Homojen (Homogeneous): VZA çalışmaları homojen KVB'leri kümesini gerektirmektedir. Homojenlik, KVB'ler arasında benzerlik derecesini ifade eder. KVB'lerinin operasyonel amaçları karakterlerinde olduğu gibi benzer olmalıdır.

Etkin Birim (Efficient Unit): Etkin birim, analizlerdeki diğer KVB'leri tarafından başarılan gerçek performansla karşılaştırıldığında, aynı çıktıları daha az girdilerle üretebilen ya da daha yüksek seviyedeki çıktıları aynı miktardaki girdilerle üretebilen KVB'leri olarak tanımlanmaktadır.

Girdi Enküçükleme (Input Minimization): Belirli çıktıların üretiminde kullanılan girdi miktarını küçültmeye çalışan analizlerde VZA adaptasyonuna girdi en küçükleme denir.

Çıktı Enbüyükleme (Output Maximization): Çıktı enbüyükleme, belirli girdi miktarıyla en fazla çıktı üretmeye çalışan analizlerde adapte edilmiş VZA'dır.

Referans Katkısı (Reference Contribution): Referans katkısı, bir KVB için etkinlik değerinin hesaplanmasına, referans KVB katkısının derecesini belirtmesinde kullanılan bir terimdir.

Bolluk (Slack): Bolluk, az üretim çıktısını ya da fazla girdi kullanımını gösterir. Etkin olmayan KVB'ni etkin hale getirmek için gerekli iyileştirmeyi gösterir. Bu iyileştirmeler girdi ve çıktıdaki artış ya da azalma şeklinde olabilir.

Hedefler (Targets): Etkin olmayan KVB'nin etkin olabilmesi için girdi ve çıktı değerlerinin gerekli düzenlemeleri ifade etmektedir.

Değişken (Variable): KVB'lerinin işlemindeki önemini belirten girdi ve çıktı faktörleridir. Örneğin çalışan sayısı, satışlar, kira, kullanılan alan gibi girdi ve çıktı sınıflandırması ölçülen sürece ve KVB'lerinin ölçülmesine karşı olan hedeflere bağlıdır.

Sanal Girdi ve Çıktılar (Virtual Input/Output): Sanal girdiler, girdi değerini, asıl (primal) model çözümü olarak verilen KVB için karşılık gelen optimal ağırlıkla çarparak bulunur. Sanal girdi ve çıktılar her faktör için öngörülen değer derecesini belirtir. Her KVB'nin sanal girdilerinin toplamı, girdiye yönelik modellerde daima 1'e eşittir. Sanal çıktıların toplamı ise birimin verimlilik skorunu gösterir.

3.3. Veri Zarflama Analizi Kullanım Amaçları

VZA'nın uygulanmasındaki amaçlar aşağıda yer alan maddeler halinde ifade edilebilmektedir:

- a. Etkinlikleri karşılaştırılacak KVB'lerin her biri için girdi-çıktı bileşenlerinin herhangi birinde veya fazlasında, göreceli etkinsizliğin kaynaklarının ve miktarının belirlenmesi
- b. KVB'lerin etkinliklerine göre sınıflandırılması
- c. Etkinlikleri karşılaştırılacak KVB'lerin yönetim performanslarının değerlendirilmesi
- d. Etkinlikleri karşılaştırılacak KVB'lerin kendi kontrolleri dışındaki program ve politikaların verimliliklerini değerlendirerek, program etkinsizliği ile yönetsel etkinsizliğin ayırt edilmesi

- e. Değerlendirmeye konu KVB'ler için kaynakların yeniden atanması amacı ile niceliksel bir temelin oluşturulması (Bu yeniden atama politikalarının genel amacı: sınırlı kaynakları, istenilen çıktıları üretmekte daha etkin kullanabilecek birimler arasında değiştirmektir.)
- f. KVB'lerin birbirlerine göre etkinlikleri karşılaştırılarak; KVB'lerin doğrudan doğruya ilişkili olmayan amaçları için etkin KVB'lerinin ya da etkin girdi-çıkıtı ilişkilerinin belirlenmesi
- g. Her bir KVB'nin mevcut girdileri ile gerçekleştirdiği performansın yürürlükteki standart performansa göre incelenmesi ve gözden geçirilmesi
- h. Önceki çalışmalardaki sonuçların karşılaştırılması

şeklinde maddelere ayrılabilir [Özcan, 2007].

3.4. Veri Zarflama Analizi Modelleri

VZA yönteminde kullanılan en yaygın modeller CCR(Charnes-Cooper-Rhodes), ve BCC(Banker-Charnes-Cooper) modellerdir. Veri Zarflama Analizi hem ölçeğe göre sabit getiri hem de değişen getiri varsayımı altında kullanılabilir. KVB'lerinin toplam teknik etkinliklerini ölçmekte VZA yönteminin temel varsayımı ölçeğe göre sabit getiri varsayımıdır. Bu varsayıma göre geliştirilen VZA modeli CCR Modeli olarak da bilinmektedir. Bu varsayım daha sonra Banker, Charnes ve Cooper (1984) tarafından değiştirilerek ölçeğe göre değişken getiri varsayımı geliştirilmiştir. Bu varsayıma göre geliştirilen VZA modeli ise BCC modeli olarak bilinmektedir. VZA yöntemi girdiye ve çıktıya yönelik olmak üzere iki şekilde kullanılabilir. Girdi yönlü (input-oriented) modellerde çıktıları sabit tutarak girdiyi minimize etmeye çalışır, çıktı yönlü (output-oriented) modellerde ise girdiler sabit kalmak üzere çıktı maksimum yapılmaya çalışılır.

VZA yönteminde etkinlik ölçümünde kullanılan çeşitli modeller, etkinliğin çıktıların ağırlıklı toplamının, girdilerin ağırlıklı toplamına oranı olarak ölçüldüğü oran modelinden türetilir [Cooper ve ark., 2000]. X_{ij} ($i = 1, 2, \dots, m$) ve Y_{rj} ($r =$

1, 2, ..., s) olarak j. örneğin (birimin) i. girdisini ve r. çıktısını tanımlamak üzere oran formu aşağıdaki gibi tanımlanabilir:

$$\max_{\mu_r, v_i} \sum_{r=1}^s \mu_r Y_{r0} / \sum_{i=1}^m v_i X_{i0} \quad (3.1)$$

$$\sum_{r=1}^s \mu_r Y_{rj} / \sum_{i=1}^m v_i X_{ij} \leq 1 \quad j = 1, 2, \dots, n \quad (3.2)$$

$$\mu_r, v_i \geq 0$$

Etkinliği ölçülen KVB'nin ağırlıklı girdi toplamı $\sum_{i=1}^m v_i X_{i0}$ 1'e eşitlenir ise CCR modeli olarak bilinen temel etkinlik modeli elde edilmiş olur.

3.4.1. CCR(Charnes, Cooper, Rhodes) modelleri

Ölçeğe göre sabit getiri (CRS - Constant Return to Scale) varsayımı girdileri bir kat arttırdığımızda, çıktılarda bir kat artacaktır. Aynı yolla eğer girdiler yarıya düşürülürse, çıktılarda yarıya düşecektir. Literatürde ağırlıklı model olarak adlandırılan primal model ve zarflama model olarak adlandırılan dual model bulunmaktadır. Dual modellerde etkinlik skorunun yanında çıktıdaki azlıklar ile girdideki fazlalıklar da ayrıca hesaplanabilmektedir.

Girdi yönlü CCR primal model (CCR_P-I)

$$E_k = \max_{\mu_r, v_i} \sum_{r=1}^s \mu_r Y_{r0} \quad (3.3)$$

$$\sum_{i=1}^m v_i X_{i0} = 1 \quad (3.4)$$

$$\sum_{r=1}^s \mu_r Y_{rj} - \sum_{i=1}^m v_i X_{ij} \leq 0 \quad (3.5)$$

$$\mu, v \geq 0$$

Girdi yönlü CCR dual model (CCR_D-I)

$$E_k = \min_{\theta, \lambda, S_r^+, S_i^-} \theta - \varepsilon \sum_{r=1}^s S_r^+ - \varepsilon \sum_{i=1}^m S_i^- \quad (3.6)$$

$$\sum_{j=1}^n Y_{rj} \lambda_j - S_r^+ = Y_{r0} \quad (3.7)$$

$$\theta X_{i0} - \sum_{j=1}^n X_{ij} \lambda_j - S_i^- = 0 \quad (3.8)$$

$$\lambda, S_r^+, S_i^- \geq 0$$

$$\varepsilon > 0$$

Çıktı yönlü CCR primal model (CCR_P-O)

$$E_k = \min_{\mu_r, v_i} \sum_{i=1}^m v_i X_{i0} \quad (3.9)$$

$$\sum_{r=1}^s \mu_r Y_{r0} = 1 \quad (3.10)$$

$$\sum_{r=1}^s \mu_r Y_{rj} - \sum_{i=1}^m v_i X_{ij} \leq 0 \quad (3.11)$$

$$\mu, v \geq \varepsilon$$

Çıktı yönlü CCR dual model (CCR_D-O)

$$E_k = \max_{\phi, \lambda, S_r^+, S_i^-} \phi + \varepsilon \sum_{r=1}^s S_r^+ + \varepsilon \sum_{i=1}^m S_i^- \quad (3.12)$$

$$\phi Y_{r0} - \sum_{j=1}^n Y_{rj} \lambda_j + S_r^+ = 0 \quad (3.13)$$

$$\sum_{j=1}^n X_{ij} \lambda_j + S_i^- = X_{i0} \quad (3.14)$$

$$\lambda, S_r^+, S_i^- \geq 0$$

$$\varepsilon > 0$$

3.4.2. BCC(Banker, Charnes, Cooper) modelleri

BCC modellerinin, CCR modellerinden tek farkı; sabit ölçek altında değil, değişken dönüşümlü ölçek varsayımı (VRS - Variable Return to Scale) altında çalışmasıdır. 1984 yılında R.D. Banker, A. Charnes ve W.W. Cooper tarafından ilk olarak ortaya atılan ve bu kişilerin soy isimlerinin baş harfleriyle (BCC) ile gösterilen bu modelin matematiksel ifadelerini aşağıdaki formüllerle özetlemek mümkündür. Denklemlerde CCR modellerinden farklı olarak; U_o : ölçeğe göre getirinin yönüyle ilgili değişken olarak tanımlanmaktadır. Modellerde yer alan U_o değişkeni ölçeğe göre değişken getiri kavramıyla ilgilidir. Modelin çözümünde U_o değişkeninin pozitif değer alması KVB'nin ölçeğe göre azalan getiri, negatif değer alması ölçeğe göre artan getiri ve sıfır değerini alması ölçeğe göre sabit getirili olduğunu göstermektedir. Kısıtlarda CCR modelinden farklı olarak $\sum_{j=1}^n \lambda_j = 1$ dışbükeylik kısıtı olduğu görülmektedir. Bu kısıt, etkinlik sınırının en iyi gözlemin çoklu doğrusal kombinasyonlarından oluşması ve göreceli etkinliğin daha esnek bir tanımlamaya kavuşması olarak açıklanmaktadır. BCC modelinde de CCR modelinde olduğu gibi hem girdi yönlü hem de çıktı yönlü model bulunmaktadır.

Girdi yönlü BCC primal model (BCC_P-I)

$$E_k = \max_{\mu_r, v_i, U_0} \sum_{r=1}^s \mu_r Y_{r0} + U_0 \quad (3.15)$$

$$\sum_{i=1}^m v_i X_{i0} = 1 \quad (3.16)$$

$$\sum_{r=1}^s \mu_r Y_{rj} - \sum_{i=1}^m v_i X_{i0} + U_0 \leq 0 \quad (3.17)$$

$$\mu_r, v_i \geq \varepsilon$$

U_0 serbest

Girdi yönlü BCC dual model (BCC_D-I)

$$E_k = \min_{\theta, \lambda, S^+, S^-} \theta - \varepsilon \sum_{r=1}^s S_r^+ - \varepsilon \sum_{i=1}^m S_i^- \quad (3.18)$$

$$\sum_{j=1}^n Y_{rj} \lambda_j - S_r^+ = Y_{r0} \quad (3.19)$$

$$\theta X_{i0} - \sum_{j=1}^n X_{ij} \lambda_j - S_i^- = 0 \quad (3.20)$$

$$\sum_{j=1}^n \lambda_j = 1 \quad (3.21)$$

$$\lambda_j, S_r^+, S_i^- \geq 0$$

Çıktı yönlü BCC primal model (BCC_P-O)

$$E_k = \min_{\mu_r, v_i, U_0} \sum_{i=1}^m v_i X_{i0} + U_0 \quad (3.22)$$

$$\sum_{r=1}^s \mu_r Y_{r0} = 1 \quad (3.23)$$

$$\sum_{r=1}^s \mu_r Y_{rj} - \sum_{i=1}^m v_i X_{ij} - U_0 \leq 0 \quad (3.24)$$

$$\mu, v \geq \varepsilon$$

U_0 serbest

Çıktı yönlü BCC dual model (BCC_D-O)

$$E_k = \max_{\phi, \lambda, S^+, S^-} \phi + \varepsilon \sum_{r=1}^s S_r^+ + \varepsilon \sum_{i=1}^m S_i^- \quad (3.25)$$

$$\phi Y_{r0} - \sum_{j=1}^n Y_{rj} \lambda_j + S_r^+ = 0 \quad (3.26)$$

$$\sum_{j=1}^n X_{ij} \lambda_j + S_i^- = X_{i0} \quad (3.27)$$

$$\sum_{j=1}^n \lambda_j = 1 \quad (3.28)$$

$$\lambda, S^+, S^- \geq 0$$

Bütün CCR ve BCC modellerinde;

E_k : k'inci KVB'nin etkinlik değeri,

μ_r : k'inci KVB tarafından r'inci çıktıya verilen ağırlık,

v_i : k'inci KVB tarafından i'inci girdiye verilen ağırlık,

Y_{rk} : k'inci KVB tarafından üretilen r'inci çıktı,

X_{ik} : k'inci KVB tarafından kullanılan i'inci girdi,

Y_{rj} : j'inci KVB tarafından üretilen r'inci çıktı,

X_{ij} : j'inci KVB tarafından kullanılan i'inci girdi,

ε : yeterince küçük pozitif bir sayı (örneğin 0,001),

θ : Büzülme katsayısı (Çıktı miktarında bir değişiklik yapmadan girdi miktarının ne kadar azaltılabileceğini gösterir),

ϕ : Genişleme katsayısı (Girdi miktarında bir değişiklik yapmadan çıktı miktarının ne kadar arttırılabileceğini gösterir),

λ_j : girdi ve çıktılara atanan ağırlık vektörü,

S_i^- : k'inci KVB'nin i'inci girdisine ait artık değişken (Girdilerdeki fazlalık),

S_r^+ : k'inci KVB'nin r'inci çıktısına ait artık değişken (Çıktılardaki azlık),

$i = 1, 2, \dots, m$ (girdi sayısı), $r = 1, 2, \dots, s$ (çıktı sayısı) ve $j = 1, 2, \dots, n$ (KVB sayısı) olarak tanımlanmaktadır.

CCR ve BCC primal modellerin çözümünde; eğer $E_k = 1$ (veya 100) olduğunda etkinliği ölçülen KVB'nin etkin olduğu söylenir. Primal modellerde etkinlik skorunun 1 (veya 100) olmadığı KVB'leri etkin değildir. Etkin olmayan bir KVB etkin olmak için örnek alabileceği referans kümesini (örneğin girdi yönlü model) için $\sum_{r=1}^s \mu_r Y_{rj} - \sum_{i=1}^m v_i X_{ij} \leq 0$ veya $\mu, v \geq 0$ gibi kısıtların 0'a eşit olması durumunda bulunabilir.

CCR ve BCC dual modellerin çözümünde; eğer $E_k = 1$ (veya 100) ve $\max(S_r^+, S_i^-) = 0$ yani S_r^+, S_i^- artık değişkenlerin tümü sıfır ise k. KVB etkindir. Ancak $E_k < 1$ (çıktı yönlü model olması durumunda $E_k > 1$) ve $\max(S_r^+, S_i^-) > 0$ ise k. KVB etkin değildir. Etkin olmayan KVB (örneğin çıktı yönlü modelde) etkin olması için E_k değeri eğer 1'den büyükse çıktısını o oranda artırmalı ve girdilerindeki fazlalıkları düşürmeli, çıktılarındaki azlıkları telafi etmelidir.

Kullanılan modellerdeki bütün değişkenler kontrol edilebilen değişkenlerdir. Örneğin bir mağazanın yaşı girdi olsun eğer fazlalık varsa mağazanın yaşından çıkarmamız gerekecektir. Ancak mağazanın yaşından çıkarmak mümkün değildir. Aynı şekilde

çalışan sayısının girdi olduğu durumlarda fazlalık çıkmışsa eğer onu azaltmamız gerekmektedir. Ama hiçbir çalışanı çıkarmak istemediğimizde bu da mümkün olmayacaktır. Bir başka problemde çevreye atılan gazlar çıktı değişken olsun ve çıktıların arttırılması sonucu çıktığında çevreye atılan gazların artırılması istenilen bir durum değildir. Bu gibi kontrol edilemeyen değişkenlerin var olması durumunda o değişkene ait girdi fazlalıkları (S_i^-) veya çıktı azlıkları (S_i^+) modelden çıkarılmalıdır. Bu sayede kontrol edilemeyen değişkenlerin durumları da modellerde göz önüne alınmış olmaktadır.

3.5. Ölçek ve Çapraz Etkinlik

Scale Efficiency olarak bilinen ölçek etkinlik, kısaca ölçeğe göre sabit getirinin ölçeğe göre değişken getiriye oranıdır. Bu oran eğer 1 çıkarsa bu KVB girdi ve çıktıları en iyi kullanan yani minimum girdiyle maksimum çıktı elde eden KVB olarak nitelendirilir. CCR etkinliği global maksimum, BCC etkinliğini de yerel maksimum olarak değerlendirilmektedir. Ölçek etkinliği 1 olan KVB'ne en verimli ölçek büyüklüğüne (Most Productive Scale Size – MPSS) sahip karar verme birimi denir. VZA'de KVB'leri, yapılan çözümleme sonunda bulunan etkinlik skorlarına göre sıralanmaktadır. En yüksek etkinlik değerine göre sahip KVB birinci sırada yer alırken, en düşük etkinlik değerine sahip KVB son sırada yer almaktadır. Fakat VZA'de etkin bulunan KVB'lerine "1" etkinlik değeri atanması, etkin olan birimlerin kendi aralarında bir sıralama yapılmasına imkân vermemektedir. Bu güçlüğü aşmak için çeşitli yöntemler geliştirilmiştir. Bunların bir özetini Adler (2002) vermiştir.

Etkin birimlerin sınıflaması için önerilen ilk metot Sexton (1986) tarafından ortaya konulan çapraz etkinlik matrisidir [Doyle ve Green, 1993]. Basitçe her bir KVB'nin etkinliğini optimal ağırlıklara göre n defa tekrar hesaplama esasına dayanır. Bu yaklaşımda, önce her bir KVB için VZA ile etkinlik skorları ve ağırlıklar bulunur. Bulunan bu ağırlıklar ile diğer KVB'lerinin etkinlik skoru hesaplanır. Bu şekilde elde edilen çapraz etkinlik skorlarının oluşturduğu çapraz etkinlik matrisi bulunur.

$$h_{kj} = \sum_{r=1}^s \mu_{rk} Y_{rj} / \sum_{i=1}^m v_{ik} X_{ij} \quad j, k = 1, 2, \dots, n \quad (3.29)$$

Eş 3.29’da gösterilen h_{kj} ; k. birimin optimal ağılıklarına göre etkinliği hesap edilmiş j. birimi göstermektedir. Bu matrisin tüm elemanları [0,1] aralığında değer alıp h_{kk} birimin kendi etkinlik skorunu yansıtmaktadır. Eş 3.29’da gösterilen formül kullanılarak 3.30’da gösterilen çapraz etkinlik matrisi oluşturulur.

$$\begin{array}{ccccc} KVB & 1 & 2 & \dots & n \\ 1 & h_{11} & h_{12} & \dots & h_{1n} \\ 2 & h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & h_{n1} & h_{n2} & \dots & h_{nn} \end{array} \quad (3.30)$$

Çapraz etkinlik matrisinde h_{11} ; KVB_1 ’in ağırlıkları kullanılarak elde edilen KVB_2 ’nin etkinlik değeridir. Her bir KVB’nin kendisinin ve diğer KVB’lerinin optimal ağırlıklarına göre aldıkları etkinlik sonuçlarının istatistiksel eğilimi o KVB’nin etkinlik sıralaması için kullanılabilir.

3.6. Window Analizi

VZA uygulamalarında genellikle kesit verisi kullanılır ve her bir KVB için analizde bir kez çözüm yapılır. Eğer panel veri varsa her bir KVB’nin bireysel performansı yanında, performansın zaman içindeki değişimine odaklanmak da önemlidir. Diğer yandan değişkenler zaman içinde genellikle bir önceki ve bir sonraki dönemden bağımsız değildirler. Bu iki bakış açısını dikkate alarak zaman dizilerindeki hareketli ortalamalar mantığıyla performans hesabı yapılabilir. Bağımlılık süresine bağlı dönemler birlikte değerlendirilebilir. Bu tür performans hesaplamaları window analizi olarak isimlendirilir. VZA tabloları “durağan”, window analizi tabloları ise “dinamik” performans tabloları olarak nitelendirilir. Window analizinde KVB, her bir zaman periyodunda ayrı bir KVB olarak değerlendirilir. Window analizi ile bir

KVB kendi dönemi yanında diğer periyotların verilerine ve diğer KVB'lerine göre de test edilmiş olur.

Önceki bölümlerde anlatılan modellerde her bir KVB için yalnızca bir dönem verileri ile yapılan analizleri içermektedir. Ancak zamanla değişen bilgi için de performans değerlendirmesi yapabilmek son derece önemli olup bu değerlendirmeyi yapabilmek için de Window analizi kullanılmaktadır. Window analizi, bir çeşit hareketli ortalama yaklaşımı yansıtmak için her bir analiz için değişen, zamana bağlı KVB içeren analiz serisinden meydana gelmektedir. Performansın zaman içinde izlenmesi, KVB'lerin farklı zaman dilimlerinde değerlendirilmeleri ile yapılmaktadır [Colbert ve ark., 2000].

Örneğin, ilk analiz; birinci, ikinci ve üçüncü üç aylık işlemlere ilişkin verilerden meydana gelebilir. Bir sonraki analiz; ikinci, üçüncü ve dördüncü üç aylık işlemlere ilişkin verilerden meydana gelebilir. Üçüncü analiz ise; bir yıldan üçüncü ve dördüncü üç aylık verileri ve bir sonraki yıl için birinci üç aylık veriyi içerebilir. Böylece her bir analiz, ayrı etkinlik dereceleriyle sonuçlanan, yeni ve farklı bir KVB setine sahip olmaktadır. Üç aylık verilerin bu tarzda kullanılması sadece KVB'lerin sayısını artırmakla kalmaz, aynı zamanda, KVB'lerin kümesi değiştikçe, her bir KVB için etkinlik derecelerinin güvenilirliğinin çalışılması için bir pencere yaratır. Eğilimler ve / veya zamana bağımlı davranışların diğer tipleri için yapılacak çalışmalarda kullanılacak başka bilgilere de ulaşılabilir [Bowlin, 1998].

Ancak, window analizi, her bir zaman dilimi için tanımlanan bir KVB'nin analizde kullanılmasını gerektirdiğinden, hesaplamaların sayısını oldukça artırmaktadır. Thanassoulis ve Dyson (1992), alternatif girdi-çıktı hedef seviyelerinin tahmininde kullanılmak ve göreceli olarak etkin olmayan organizasyonel birimlerin etkin hale getirilmesine yardımcı olmak üzere çeşitli VZA temelli modeller geliştirmişlerdir [Colbert ve ark., 2000].

Örnek olarak Ek.1'de gösterilen Charnes ve arkadaşlarının uçak bakım işlemleri ile ilgili yaptığı çalışmasında (1985) 14 KVB için 7 aylık dönemler boyunca ABD Hava

Kuvvetleri'nde taktik bombardıman uçağı kanatları verilerini kullanarak window analizi yapılmıştır. Ek.1.'de gösterildiğı gibi genişliğı 3'er aylık olarak alındığında ilk olarak ilk üç ay (M1,M2,M3) her bir KVB için hesaplanmaktadır. Son pencere aralığına gelene kadar (M5, M6, M7) pencereler kaydırılır ve her bir KVB için hesaplanmaktadır. KVB sayısı 14 olduğı için her pencere aralığında 42 (14x3) işlem yapılmaktadır. Bu açıdan işlem yükü ağır olan bir analiz türüdür.

Bir başka örnek olarak ise Alp ve arkadaşlarının (2011) Avrupa Ülkeleri ve Türkiye'nin enerji ve çevre performans değerlendirmesi çalışmasında KVB sayısı (n) 29, periyot sayısı(w) (1998–2005) 8, pencere uzunluğu(p) ise 3 olarak alınmıştır. Böylece her KVB için pencere sayısı $w=k-p+1=8-2+1=6$ 'dır. Genel olarak $k-p+1$ farklı pencerenin her birinde $n \times p$ KVB için hesaplama yapılır. Window analizi sonuçlarının Ek-2'deki gibi düzenlenmesi performans değişimini, performans trendini, referans kümelerinin kalıcılığını ve diğer mümkün yorumları kolaylaştırır. Her bir KVB, ilk pencerede (1998, 1999, 2000) üç ardışık yılın her biri için farklı bir KVB gibi gösterilir ve böylece $29 \times 3=87$ KVB için hesaplama yapılır. İkinci pencere için, ikinci üç yıldaki(1999, 2000, 2001) 87 KVB için performans değerleri hesaplanır. İşlem bu biçimde tekrarlanarak devam eder. Son olarak (2003, 2004, 2005) yılları için 87 KVB'nin performansı bulunur. Performans hesaplaması yapılan toplam farklı KVB sayısı = $n. w. p = 29 \times 6 \times 3=522$ 'dir. Ek-2 window analizinin gerisindeki yapıyı resmetmektedir. Örneğın A ülkesi için birinci pencere performans değerleri şöyle bulunmaktadır. Kullanılan VZA modelinde ilk yıla ait girdi-çıkıtı verilerinin arkasına ikinci ve üçüncü yılın verileri de eklenir. Amaç fonksiyonu A ülkesinin birinci yıl verilerine göre oluşturulduğunda birinci, ikinci yıl verilerine göre oluşturulduğunda ikinci ve üçüncü yıl verilerine göre oluşturulduğunda üçüncü yıla ilişkin performans değerleri bulunur. M1 için Ek-2'de İsviçre için bu değerler (1998, 1999, 2000) yılları için sırasıyla (65.2, 63.6, 70.4) bulunmuştur. Her bir ülkeye ait "satır görüntüsü" o ülkenin performans trendlerini netleştirir. Örneğın Ek-2'de Çek Cumhuriyeti 1998–2002 yılları arasında 100 etkinlik skoruna sahip iken 2003'ten sonra performansını düşürmüştür (98.92, 97.53, 99.76). İsviçre ise etkinlik skorunu 65.17'den (1998) hemen hemen her yıl arttırarak 78.56'ya çıkarmıştır. Trendi giderek artan ilerlemeyi göstermektedir. Bu kısıtlı tabloda Bulgaristan ve Türkiye

M1'e göre etkindirler. M2 için Ek-2'ye göre, Fransa'nın etkinlik skoru 71.38'den başlayarak her geçen dönem biraz daha azalarak 57.64'e düşmüştür. Norveç 87.97'den başlayarak bazı dönemlerde performansını arttırmış olsa bile altıncı pencerenin son yılı olan 2005 yılında performansını 67.65'e düşürmüştür. Türkiye bazı dönemler etkinliğini kaybetse de analizimizin son iki penceresinde de etkin çıkmıştır. Bulgaristan ise her dönemde etkin olduğu gözlemlenmiştir.

3.7. Veri Zarflama Analizinin Avantajları ve Dezavantajları

Her teknikte olduğu üzere VZA'nın de kullanımında avantajlarının yanında dezavantajları bulunmaktadır. Bu anlamda çeşitli çalışmalarda değerlendirilen VZA'nın güçlü ve zayıf yönleri incelenecektir.

VZA'nın güçlü yönleri şöyle sıralanmaktadır.

- VZA'nın en önemli güçlü yanı nesnelliğidir. Çünkü VZA sayısal verilere dayanan etkinlik skorları sağlar ve insanların öznel düşüncelerini kullanmaz. Ulaşılan verileri mümkün olan en yüksek derecede nesnel bir şekilde değerlendiren değerli bir tekniktir [Ramanathan, 2003].
- VZA, etkin olmayan bir KVB'nin performansını, kümesindeki etkinlik sınırında yer alan KVB'lerin seviyesine çıkarmak için, alternatif yollar belirler. Karar verici ise KVB'ne uygun iyileştirme yolunu yargısı ve tecrübesi ile seçer [Özcan, 2007].
- VZA'nın uygulamasında ilgili tüm girdi ve çıktıları tanımlandığı için karar vericiler üretim sürecini daha iyi tanıyabilirler [Canbek, 2007].
- VZA çalışmasında kullanılan ve analiz sonuçlarını içeren verilerden hareketle ayrıntılı bir veri tabanı oluşturulabilir [Oruç, 2008].
- VZA parametrik olmayan bir yöntem olduğu için belirli bir fonksiyonel dağılım kuralına uyması gibi bir varsayımı taşımaz. Girdi ve çıktı verileri deterministik olarak varsayıldığı için veriler rassal bir mekanizma ile üretilmez [Canbek, 2007].

- VZA merkezi eğilim ölçülerinden ziyade sınıra yönlenmektedir [Çağlar, 2003].
- Regresyon tekniklerinde ortalama değere göre karşılaştırma yapılırken, VZA ile KVB'leri doğrudan etkin olan bir referans birim ya da referans kümesi ile karşılaştırılmaktadır [Bektaş, 2007].
- Üretim fonksiyonunun çoğunlukla bir tek çıktı ve çok girdiyi ilişkilendirerek tahmin yapılan regresyon tekniklerinden ziyade, VZA çok girdi ve çıktı içeren modeller için kullanılabilir [Oruç, 2008].
- Girdilerin ve çıktıların çok farklı birimlere sahip olduğu durumlarda aralarında zorunlu olarak bir dönüşüm yapmaya gerek yoktur [Ramanathan, 2003]. Çağlar (2003) birden çok girdi ve çıktısı olan KVB'lerinin etkinliklerinin tek bir değere indirgenerek ölçülebilmesi durumunu VZA'nın en önemli avantajı olarak değerlendirmektedir.
- VZA, göreceli etkinliği hesaplarken her KVB için kullandığı formülasyonu ayrı ayrı en iyiler. Ayrıca, her bir etkin olmayan KVB için neler yapılması gerektiğine dair öneriler sunar [Bektaş, 2007].
- VZA ile her bir KVB için teknik ve ölçek etkinlikleri ayrı ayrı hesaplanabilmektedir. Böylece üretimden ya da ölçekten kaynaklanan etkin olmayanları da ayrı ayrı belirlenebilmektedir [Çağlar, 2003].

VZA'nın zayıf yönleri ise şöyle sıralanmaktadır.

- VZA'nın deterministik yapısı ve bir uç nokta tekniği olması ölçüm hatalarına ve değişken seçimine karşı oldukça duyarlı olmasına neden olmaktadır [Oruç, 2008]. Bir KVB'nin aykırı değerlere sahip olması etkinlik sınırının biçiminin bozulmasına yol açacağından dolayı elde edilecek etkinlik skorları gerçeği yansıtmaktan uzak olacaktır [Çağlar, 2003]. VZA etkinliklerinin küçük hatalara bile oldukça duyarlı olması duyarlılık analizlerini VZA sonrası prosedürün önemli bir parçası haline getirmektedir [Ramanathan, 2003].
- VZA'nın diğer bir zayıf yönü, elde edilen etkinlik skorlarının yalnızca incelenen gözlem kümesi için geçerli olmasıdır. Farklı KVB'lerinin yer alacağı başka bir

çalışmadan elde edilen etkinlik skorları farklı olacağı için karşılaştırma yapmak mümkün olmayacaktır [Çağlar, 2003].

- VZA'nın kullanımında diğer bir sıkıntı ise serbestlik derecesi problemidir. İncelenen KVB'lerin toplam sayısı, KVB'lerine ilişkin girdi ve çıktı değerlerinin toplam sayısının en az üç katı olursa VZA sonuçları güvenilir olacaktır. Aksi durumda VZA çok sayıda KVB'ni etkin olarak değerlendirecektir [Çağlar, 2003].
- VZA genel olarak fiziksel girdi ve çıktı ölçütleri ile test edildiğinden teknik girdi-çıkıtı etkinliği ile sınırlıdır. Yöntemin yetenekleri girdi ve çıktılara göreli fiyatlar ya da öncelikli ağırlıklar atanarak güçlendirilebilir [Özcan, 2007].
- KVB'lerin değerlendirileceği girdi ve çıktıların belirlenmesi üretim sürecinin doğru olarak analiz edilmesi ve yöntemin sonuçlarının sağlıklı bir şekilde elde edilmesi açısından önemlidir. Kritik bir girdi ya da çıktı araştırma dışında bırakıldığında sonuçlar yanıltıcı ve yanıltıcı olabilir [Canbek, 2007].
- Her KVB için ayrı bir doğrusal programlama modelinin çözümünün gerekliliği, büyük boyutlu problemlerin çözümünde zaman alıcı olabilir. Ancak günümüzde bilgisayarların hesaplama gücü ve VZA uygulamalarındaki KVB sayıları dikkate alındığında bu ciddi bir problem değildir [Ramanathan, 2003].
- VZA, parametrik olmayan bir teknik olduğundan sonuçlara istatistiksel hipotez testlerinin uygulanması oldukça zordur [Oruç, 2008].
- VZA modelleri, tek zaman kesitinde değerlendirildiği için statik modellerdir. Gerçek hayat dinamik bir özellik göstermektedir, yani KVB'lerinin bazı girdilerini çıktılara dönüştürebilmesi bir periyottan daha uzun bir süre almaktadır. Bu sebeple farklı periyotlardaki veriler için uygun indirgeme oranlarının kullanılması gerekecektir [Özcan 2007].
- Gözlemlenen performansın en iyi performansla olan farkının nedeni sadece etkin olmayan olarak değerlendirilerek uç gözlem noktalarında meydana gelebilecek ölçüm hataları göz ardı edilmektedir. Bu durum yanıltıcı sonuçlara neden olabilir [Canbek 2007].
- VZA'da etkinlik skorları belli sayıda doğrusal programlama problemlerinin çözümü ile elde edildiğinden, iki girdi ve çıktıdan fazla olan durumda teknik

bilgisi olmayan biri için uygulama ve sonuçların değerlendirilmesi zor olabilir [Ramanathan, 2003].

- Referans kümesine dâhil olan KVB'lerinin kendi başlarında değerlendirildiğinde de gerçekten etkin olup olmadıkları (mutlak etkinlik) hakkında bir yorum yapılabilmesi güçtür. Çünkü VZA etkinlikleri göreceli olarak değerlendirmektedir [Oruç, 2008].
- VZA sadece bir ya da daha fazla girdi ve çıktı olması durumunda etkinlik skoru hesaplamak için tasarlanır. Yöntem, performansı değerlendirirken hiç girdi ve çıktı olmaması durumunu içerecek esneklikte olması durumunda daha iyi bir duruma gelecektir. Literatürde sıkça geçen okullarda performans değerlemesi örneğinde Hollanda'da yer alan tüm okulların girdilerinin aynı olduğu gözlenmiştir. Bu nedenle herhangi bir girdi kullanılması gerekmediği varsayılmıştır. Ancak herhangi bir girdi olmadan VZA analizini doğrudan uygulamak mümkün değildir. Bu problemin üstesinden gelebilmek için tüm okullar için aynı değere sahip bir kukla değişken tanımlamak yeterli olsa da, bu durum tekniğin bir kısıtıdır [Ramanathan, 2003].

3.8. Veri Zarflama Analizinde Kullanılan Paket Programlar

VZA doğrusal programlamaya dayanan bir teknik olduğundan dolayı doğrusal programlama çözümü için uygun olan herhangi bir yazılım paketi, temelde VZA uygulamaları için de kullanılabilir. DP çözüm programlarına örnek olarak; LINDO, GAMS, QSB ve STORM bilgisayar programları verilebilir [Özcan, 2007]. Ancak VZA'nın standart doğrusal programlama paketleri ile çözülemeyen önemli karakteristikleri vardır. Bu nedenle özel yazılımlara ihtiyaç duyulmaktadır. Bu karakteristikler Ramanathan (2003) tarafından aşağıdaki şekilde sıralanmaktadır:

- VZA uygulamalarında doğrusal programlama problemleri çalışmaya dâhil olan KVB'lerin her biri için ayrı ayrı çözülmelidir. n tane KVB için doğrusal programlama yazılımı her seferinde amaç fonksiyonu ve diğer parametreleri değiştirilerek n kez çalıştırılmalıdır. Bu durum yorucu ve zaman alıcıdır.

- Çarpan VZA modelleri normalleştirme kısıtları içerir. Örneğin, çıktıyı maksimize eden çarpan problemi girdilerin ağırlıklı toplamını normalleştiren $U_m^T X_m = 1$ kısıtını gerektirir. Buradan açıkça görüleceği üzere, u_i ağırlıkları girdilerin büyüklüğü ile ters orantılıdır: Eğer X_i büyükse, ona ilişkin u_i nin de küçük olması gerekir. Eğer girdilerin değeri yeteri kadar büyükse oluşan küçük ağırlık değerleri doğrusal programlama algoritmasında optimallik testini karıştırabilir.
- Bir VZA modelinin optimal çözümü genellikle normal doğrusal programlama modellerinde karşılaşılandan daha fazla sıfır değeri içerir. Örneğin bir zarflama VZA modeli etkin bir KVB için çözüldüğünde optimal çözümde sadece θ değişkeni ve ona ilişkin λ pozitifdir ve diğer tüm değerler sıfıra eşittir. Bu durum ele alındığında, VZA modellerinin bozulmaya neden olabileceği görülmektedir. Doğrusal programlamada temel değişkenlerden en az biri sıfır değerine sahip olduğunda uygun çözüme ilişkin bir temelin bozulabileceği görülür. Bir VZA modeli normal bir doğrusal programlama paketi ile çözüldüğünde, birçok iterasyonun temel çözümleri birçok sıfır temel değişkenlerini içerebilir ve bu durum temel çözümün bozulmasına yol açabilir. Bozulmanın teorik olarak ana çıkış noktası simpleks işleminin hedef değerini geliştirmeden ve hesaplamaları sonlandırmadan aynı dizi iterasyonları devam ettirdiği döngü olayıdır. Bu nedenle hesaplama hızını düşüren döngüleri elimine etmek gerekir.
- Arşimedgil olmayan çok küçük sayıların kullanımı doğrusal programlama yazılımı kullanımı ile çözümde hesaplama zorlukları yaratabilir. Arşimedgil olmayan çok küçük değerler sayı olmadıkları için, standart doğrusal programlama paketleri bunların küçük sayılar şeklinde temsil edilmesini gerektirir. Mevcut hesaplamalarda bu çok küçük değerler optimizasyonun etkilenmemesi için diğer girdi ve çıktılarından daha küçük olacak şekilde seçilmelidir.

VZA’nde kullanılan ticari ve ticari olmayan paket programlar şunlardır.

3.8.1. Ticari veri zarflama analizi paket programları

DEA-Solver-Pro Version 6.0

Geliştirici: SAITECH, Inc

Fiyat: \$ 800

Sistem Gereksinimleri: MS Windows, MS Excel 97

Web Sitesi: www.saitech-inc.com

Frontier Analyst Version 4

Geliştirici: Banxia Yazılım Ltd.

Fiyat : £ 395

Sistem Gereksinimleri: Microsoft işletim sistemi Windows 2000, XP veya Vista. Sabit diskinizde boş alan en az 150 MB olmalıdır. Bilgisayar 128 MB RAM veya daha fazla kullanılabilir RAM olması gerekir. İşlemci en az Pentium, optik sürücü ve Internet erişimi olmalıdır. Ekran en az 800 x 600 piksel veya 1280 x 1024 piksel çözünürlükte en iyi görüntü sağlanabilir.

Web Sitesi: www.banxia.com

KonSi DEA Analysis

Geliştirici: Konsinye Ltd.

Fiyat: £ 432

Web Sitesi : www.dea-analysis.com

xlDEA 2.0

Geliştirici: Kimon Spiliopoulos

Fiyat: \$ 180

Sistem Gereksinimleri: Windows 2000, NT veya XP. MS Excel 2000 veya 2003

Web Sitesi: www.prodtools.com

PIM DEAsoft-V2

Geliştirici: Emmanual Thanassoulis ve Ali Emrouznejad

Fiyat: £ 390

Sistem Gereksinimleri: En az Pentium işlemci, 128 MB RAM ve 100 MB boş alan gereklidir. İşletim sistemi Windows XP'dir.

Web Sitesi: www.deasoftware.co.uk

GAMS / DEA

Geliştirici: Anthony Brooke, David Kendrick ve Alex Meeraus

Sistem Gereksinimleri: Birçok işletim sistemi platformları ile uyumludur.

Web Sitesi: www.gams.com

DEAFrontier

Geliştirici: Joe Zhu

Fiyat: \$ 2559

Sistem Gereksinimleri: Yalnızca MS-Excel 97, 2000, 2003, 2007 gereklidir.

Web Sitesi: www.deafontier.com

3.8.2. Ticari olmayan veri zarflama analizi paket programları

DEAP Version 2.1

Geliştirici: Tim Coelli

Sistem gereksinimleri: Bu program MS komut istemine göre çalışır.

Web Sitesi: www.uq.edu.au/economics/cepa/coelli.htm

Win4DEAP Version 1.1.2

Geliştirici: Michel Deslieries

Sistem Gereksinimleri: 32-bit platform için işletim sistemi Windows 95, 98, ME, 2000 ve XP olan Windows arayüzünde çalışır.

Web Sitesi: www.umoncton.ca/desliem/dea/install.html

EMS - Efficiency Measurement System Version 1.3

Geliştirici: Holger Scheel - Dortmund Üniversitesi - Almanya

Sistem Gereksinimleri: Windows 9x/NT

Web Sitesi: www.holger-scheel.de/ems/

DEA-Solver-LV

Geliştirici: SAITECH Inc

Sistem Gereksinimleri: Windows 2000, NT veya XP. MS Excel 2000 – 2003.

Web Sitesi: www.prodtools.com/Products.html

myDEA

Geliştirici: Robert Wilken ve Gert Harren

Web Sitesi: www.mydea.org

Qhull

Geliştirici: Minnesota Üniversitesi Geometri Merkezi

Sistem Gereksinimleri: Bu uygulama Windows DOS komut istemi gerektirir.

Web Sitesi: www.qhull.org

iDEAs

Geliştirici: Keck Sanal Fabrika Lab

Sistem Gereksinimleri: Web tabanlı bir uygulamadır.

Web Sitesi: www.isye.gatech.edu/ideas

Dea Solver Online Version 1.2

Geliştirici: Dennis Sebastian

Sistem Gereksinimleri: JavaScript ile bir tarayıcı ile internetten Hohenheim Üniversitesi web servisine aktarır ve servisin kullanımı ücretsizdir.

Web Sitesi: www.dea.uni-hohenheim.de

D.E.A.O.S. - Data Envelopment Analysis Online Software

Geliştirici: İran Uluslararası Enstitüsü

Sistem Gereksinimleri: Online

Web Sitesi: www.deaos.com

4. DOĞRUSAL PROGRAMLAMA VE SİMPLEKS YÖNTEM

4.1. Doğrusal Programlama

Doğrusal Programlama (DP) amaç fonksiyonu ve kısıtlayıcıları karar değişkenlerinin doğrusal fonksiyonu biçiminde olan problemlerin çözümüyle uğraşan bir optimizasyon tekniğidir. İşletme problemlerinin, sayısal verilerle en basit şekilde anlatımı DP ile mümkündür. DP, belli doğrusal eşitliklerin veya eşitsizliklerin kısıtlayıcı koşulları altında doğrusal bir amaç fonksiyonunu optimum hale getirmek biçiminde tanımlanabilir. Optimum hale getirmek, belli bir amaca en az masrafla ulaşmak ya da belli kaynaklarla en çok ürünü sağlamak anlamına gelir [Esin, 1998]. DP sürecinde, önce gerekli bilgiler toplanır, probleme ait bir model kurulur ve daha sonra bu modelin çözümleri bilgisayar destekli yazılım paketleri ile bulunur. DP'nin üç önemli bileşeni vardır: Amaç fonksiyonu, Kısıtlayıcı fonksiyonlar ve Pozitif kısıtlama [Alan ve Yeşilyurt, 2004].

Amaç Fonksiyonu: DP modelinde doğrusal biçimde ifade edilen bir amaç fonksiyonu vardır. Amaç fonksiyonu, kâr maksimizasyonu ya da maliyet minimizasyonu şeklinde olur. Amaç fonksiyonu Z , kontrol edilebilir değişkenler X_j ($j = 1, 2, \dots, n$) ve sabit katsayılar (birim başına kâr ya da birim başına maliyet katsayıları) c_j ($j = 1, 2, \dots, n$) olmak üzere

$$Z = \sum_{j=1}^n c_j X_j \quad (4.1)$$

Biçiminde ifade edilebilir. Bu amaç fonksiyonun açık yazılımı ise şöyledir.

$$Z = c_1 X_1 + c_2 X_2 + \dots + c_n X_n \quad (4.2)$$

Kısıtlayıcı Fonksiyonlar: İşletmeler, faaliyetlerini bir takım kısıtlayıcı koşullar altında sürdürürler. Makinelerin kapasite kullanımları, iş gücü, finansman, zaman

sınırlılığı vb. gibi koşullar bu kısıtlayıcılara örnek olarak verilebilir. Kısıtlayıcılar, teknoloji matrisi a_{ij} , ihtiyaç vektörü b_i olmak üzere standart maksimizasyon probleminde Eş. 4.3.'de gösterildiği gibi standart minimizasyon probleminde ise Eş. 4.4.'de gösterildiği gibi ifade edilirler.

$$\sum_{j=1}^n a_{ij}X_j \leq b_i, \quad i = 1, 2, \dots, m \quad (4.3)$$

$$\sum_{j=1}^n a_{ij}X_j \geq b_i, \quad i = 1, 2, \dots, m \quad (4.4)$$

Standart DP problemlerinde “ $\geq$ ” ya da “ $\leq$ ” yanı sıra “ $=$ ” işareti hem maksimizasyonda hem de minimizasyon problemlerinde kullanılabilir. Örneğin makinelerin tam kapasite ile çalışmaları durumunda “ $=$ ”lik kullanılır. Standart olmayan DP problemlerinde kısıtlayıcıların sağındaki işaretler “ $\geq$ ”, “ $\leq$ ” ya da “ $=$ ” işaretleri karışık olarak ta kullanılabilmektedirler [Hacısalıhoğlu, 1992].

Pozitif Kısıtlama: İşletme faaliyetleri koordinat düzleminin birinci bölgesinde meydana gelir. Yani, negatif üretim ya da negatif maliyet olmayacağından karar değişkenleri X_j 'lerin negatif olması düşünülemez. Bu durum matematiksel olarak Eş 4.5.'de gösterildiği gibi ifade edilir.

$$X_j \geq 0, \quad j = 1, 2, \dots, n \quad (4.5)$$

Yukarıdaki açıklamalar doğrultusunda bir doğrusal programlama probleminin genel yapısı Eş. 4.6. Eş. 4.7. ve Eş. 4.8.'de gösterildiği gibi olacaktır.

Amaç fonksiyonu;

$$\{\max, \min\} Z = \sum_{j=1}^n c_j X_j, \quad j = 1, 2, \dots, n \quad (4.6)$$

Kısıtlayıcılar;

$$\sum_{j=1}^n a_{ij}X_j \{ \leq, \geq, = \} b_i, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \quad (4.7)$$

Pozitif kısıtlama;

$$X_j \geq 0, \quad j = 1, 2, \dots, n \quad (4.8)$$

Yukarıda genel matematiksel modeli verilen DP modeli daha açık biçimde aşağıdaki gibi yazılabilir.

Amaç fonksiyonu:

$$\{ \max, \min \} Z = c_1X_1 + c_2X_2 + \dots + c_nX_n \quad (4.9)$$

Kısıtlayıcılar:

$$\begin{aligned} a_{11}X_{11} + a_{12}X_{12} + \dots + a_{1n}X_{1n} &\{ \leq, \geq, = \} b_1 \\ a_{21}X_{21} + a_{22}X_{22} + \dots + a_{2n}X_{2n} &\{ \leq, \geq, = \} b_2 \\ \vdots & \\ a_{m1}X_{m1} + a_{m2}X_{m2} + \dots + a_{mn}X_{mn} &\{ \leq, \geq, = \} b_m \end{aligned} \quad (4.10)$$

Pozitif kısıtlama:

$$X_1 \geq 0, X_2 \geq 0, \dots, X_n \geq 0 \quad (4.11)$$

Bu model, matris gösterimi ile de aşağıdaki gibi yazılabilir. a_{ij} katsayılarından oluşan teknolojik matris;

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (4.12)$$

İhtiyaç vektörü;

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (4.13)$$

Fiyat (ya da maliyet) katsayılarından oluşan vektörü de

$$C = [c_1 \quad c_2 \quad \cdots \quad c_n] \quad (4.14)$$

Karar değişkenleri vektörü ise

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \quad (4.15)$$

şeklinde verilirse,

Amaç fonksiyonu;

$$\{max, min\} Z = [c_1 \quad c_2 \quad \cdots \quad c_n] \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \quad (4.16)$$

Kısıtlayıcılar,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \begin{matrix} \geq \\ \leq \\ = \end{matrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \quad (4.17)$$

Pozitif kısıtlama;

$$X_j \geq 0, \quad j = 1, 2, \dots, n \quad (4.18)$$

şeklinde olur.

4.2. Simpleks Yöntem

Doğrusal programlama problemlerini çözmede yaygınca kullanılan simpleks yöntemi ilk kez 1947 yılında G.B. Dantzig tarafından geliştirilerek ABD Hava Kuvvetlerinin planlanmasında kullanılmıştır [Öztürk, 2001]. Daha sonra Charnes, Cooper ve arkadaşları ekonomik analizler için uygulamalı çalışmalar yapmışlardır. Simpleks Yöntemi cebirsel tekrarlar (iterasyon) işlemine dayanır. Yöntemde önce başlangıç simpleks tablosu düzenlenir sonra tekrarlayıcı işlemler ile belirli bir hesap yöntemi içinde gelişen çözümlere doğru ilerleyerek optimal çözüme ulaşıncaya kadar işlemler sürdürülür. Gelişen çözüm tablolarında amaç fonksiyonunun ve karar değişkenlerinin değişen değerleri gözlenebilir.

4.3. Simpleks Yöntemin Uygulanması

Simpleks yöntemde en iyi çözüme ulaşmaya çalışırken çözüm alanı içinde farklı noktalar denenir. Bu deneme sırasında aynı standart hesaplamalar yenilenerek en iyi çözüme erişinceye kadar ardarda çözümler geliştirilir. Simpleks yönteminin bir başka özelliği de, her yeni çözümün en az bir önceki çözüm kadar iyi olmasıdır. Bu önemli özellik, kesinlikle en iyi çözüme yaklaşılmamasını sağlar. Sonuçta yöntem, en iyi çözüme erişildiğini kendisi belirler. Birçok sorun için belirli sayıda yineleme sonunda çözüme ulaşır. Yineleme özelliği ve standart hesaplama yolları nedeniyle bilgisayara uyarlanması oldukça kolaydır [Tütek ve Gümüsoğlu, 2000].

Doğrusal Programlama sorununun Simpleks Yöntemi ile çözümü şu biçimde gerçekleştirilebilir:

- Problemin standart forma getirilmesi,
- Sistematiik yordam ve kurallar izleyerek çözüme erişme.

4.4. Problemin Standart Forma Getirilmesi

Simpleks y nteme bařlamadan  nce problemlerin dođru bi imde ifade edilmesi gerekir ve DP problemlerinin standart řekilde olması i in ařađıdaki  zellikleri tařıması gerekmektedir:

- B t n kısıtlamalar eřitlik olmalıdır.
- B t n deđiřkenler pozitif olmalıdır.
- Ama  fonksiyonu maksimizasyon ya da minimizasyon olmalıdır.

4.4.1. Kısıtlamalar

Standart formda kısıtlamaların eřitlik olması gerekmektedir. Bunu yapabilmek i in eřitsizliđin t r ne g re yeni deđiřkenler kullanılması gerekir. Her bir “ $\leq$ ” eřitsizliđi i in negatif olmayan aylak deđiřkeni (s) eřitsizliđin sađ tarafı ile sol tarafı arasındaki farkı ifade eder. Yani b yle bir eřitsizliđin sol tarafına bir aylak deđiřken eklenerek eřitsizlik, eřitlik haline getirilebilir.  rneđin kısıtlardan biri $x_1 + 2x_2 \leq 6$ řeklinde ise $x_1 + 2x_2 + s = 6, s \geq 0$ řeklinde olacaktır. Her bir “ $\geq$ ” Eřitsizliđi i in negatif olmayan artık deđiřken (s) vardır ki, bunun deđeri eřitsizliđin sađ ve sol tarafı arasındaki farkı g sterir. Yani eřitsizliđin sađ tarafına bir deđiřken eklenerek eřitsizlik, eřitlik haline d n řt r lebilir. Kısıtlamaların sađ taraflarında sadece deđerlerin olması gerekmektedir. Bu nedenle eklenen yeni deđiřken eřitsizliđin sol tarafından  ıkartılır.  rneđin kısıtlardan biri $3x_1 + 2x_2 - 3x_3 \geq 15$ řeklinde ise $3x_1 + 2x_2 - 3x_3 - s_2 = 15, s_2 \geq 0$ řeklinde olacaktır. Negatif olmayan aylak deđiřken kısıtlamada kullanılmayan ve bořa harcanan ya da yitirilen kaynakları, artık deđiřken ise genellikle fazla kaynak veya kapasiteyi g sterir. Standart formda eřitliđin sađ tarafının pozitif olması gerekmektedir. Eđer eřitliđin sađ tarafında negatif bir deđer var ise eřitlik “-1” ile  arpılır.  rneđin kısıtlardan biri $3x_1 + 2x_2 - 3x_3 - s_2 = -15$ řeklinde ise $-3x_1 - 2x_2 + 3x_3 + s_2 = 15$ řeklinde olacaktır.

4.4.2. Değişkenler

Standart formda bütün değişkenlerin sıfır veya sıfırdan büyük, pozitif değerler taşıyacak değişkenler olması gerekmektedir. Eğer problemde sınırlandırılmamış, negatif ya da pozitif değerler alabilecek bir değişken var ise, bu değişken negatif olmayan iki ayrı değişken arasındaki fark olarak tanımlanabilir. Bu durumda optimal simpleks çözümünde değişkenlerden sadece bir tanesi pozitif değer alacaktır. Örneğin x_1 değişkeni sınırsız ise, $x_1 = x_{11} - x_{12}$ şeklinde bir tanımlama yapılarak iki yeni değişken sisteme eklenir.

4.4.3. Amaç fonksiyonu

Her ne kadar standart form her tür amaç fonksiyonu kullanmaya izin veriyorsa da bazı durumlarda maksimizasyon ve minimizasyon arasında dönüşüm yapmak gerekebilir. Herhangi bir maksimizasyon problemi, minimizasyon problemi olarak veya bunun karşıtı minimizasyon problemi, maksimizasyon problemi olarak işlem görebilir. Böyle bir işlem amaç fonksiyonunda yer alan her katsayının işaretini değiştirmekle sağlanabilir. Yani amaç fonksiyonu “-1” ile çarpılır. Örneğin $\max Z = 3x_1 + 2x_2 - 3x_3$ matematiksel olarak $\min -Z = -3x_1 - 2x_2 + 3x_3$ ‘e eşittir. Böyle işlem gördürülse dahi problemdeki karar değişkenlerinin optimal değerleri aynı olacaktır. Yani amaç fonksiyonunda yer alan karar değişkenlerinin optimal değerleri yalnız işaretleri değişmek üzere aynı kalır.

4.5. Simpleks Yöntem ile Çözüm

Bir eşitlik sisteminde m eşitlik ve n değişken varsa ve değişken sayısı eşitlik sayısından büyükse tam çözüm bulmak matematiksel olarak mümkün değildir. Bu durumda (n-m) tane değişkene herhangi bir değer atanır ve geri kalan n değişkenin değeri bulunur. Simpleks metodunda da bu durumla karşılaşılabilir. Böyle bir durumda n tane değişken temel değişkenler olarak belirlenir ve geri kalan (n-m) tane değişken temel dışı değişken olur, sıfır değerini alırlar. Bir temel çözümün uygun

olabilmesi için bütün temel değişkenlerin değerinin sıfırdan büyük olması gereklidir.

Simpleks metodu uygun bir çözüm ile başlar ve her aşamada daha iyi bir çözüm bularak ilerler. Sistemli bir şekilde ilerlerken daha iyi bir çözüm bulunamayacak duruma gelindiğinde optimal çözüm bulunmuş olur. Simpleks yönteminin çalışabilmesi için ilk olarak uygun temel çözümün bulunması gerekmektedir. İlk uygun temel çözüme ulaşabilmek için kısıtlamalarda ve amaç fonksiyonunda bazı değişiklikler yapmak durumunda kalınabilir.

$$\begin{aligned} \min Z &= CX \\ AX &= b \\ x &\geq 0 \end{aligned} \tag{4.18}$$

İlk çözüme giren temel değişkenler X_B karşı gelen A 'nın sütunları B , temel dışı değişkenler X_R ve bunlara karşı gelen katsayılar matrisi R ile gösterilirse, $AX=b$

$$\begin{bmatrix} R & B \end{bmatrix} \begin{bmatrix} X_R \\ X_B \end{bmatrix} = b \tag{4.19}$$

şeklinde yazılır. $X_R = 0$ iken $X_B = B^{-1}b$ elde edilir. $X_B \geq 0$ ise temel uygun çözüm olur. $n > m$ olan DP problemlerinde temel uygun çözüm sayısı en fazla $n!/(n-m)!n!$ kadardır.

Simpleks algoritmasının adımlarını şu şekilde sıralayabiliriz.

1. *Adım:* Başlangıç temel uygun çözüm bulunur.
2. *Adım:* Temele girecek değişken belirlenir.
3. *Adım:* Temelden çıkacak değişken belirlenir.
4. *Adım:* Çözümün optimum olup olmadığı kontrol edilir.
5. *Adım:* Optimum ise durulur, değil ise 2. Adıma geri dönülür.

Simpleks algoritmasının ilk adımı, başlangıç temel uygun çözümün kolay bulunması için temel değişkenlere karşı gelen B matrislerinin birim matris olarak alınmasından oluşur. A da birim matris bulunmaz ise yapay değişkenlerin eklenmesiyle $I=BCA$ elde edilir. DP problemi eşitlik haline getirildikten sonra A içerisinde birim matris bulunmadığında birim matris oluşturulmak için kullanılan değişkenlere yapay değişken adı verilir. Böyle yapay değişkenleri çözümden çıkarmak için amaç fonksiyondaki karşı gelen fiyatları bu fonksiyonu ters yönde etkileyen yeterince büyük sayılan ($M>0$) alınır. Maksimum problemlerinde $-M$, minimum problemlerinde M sayısı alınır. Yapay değişken ekleyerek çözüm yapan bu algoritmaya M-yöntemi adı da verilir.

Simpleks Algoritması:

1. *Adım:* Verilen DP problemi uygun değişkenlerin ilavesiyle $b \geq 0$ olmak üzere temel değişkenlere karşı gelen sütunlar birim matris oluşturacak şekilde $AX = b ; x \geq 0, \min Z$ yazılıp, bir başlangıç uygun çözüm bulunur.

2. *Adım:* $B^{-1}R = y_j$ göz önüne alınarak $z_j = c_B y_j$ eşitliği ile temel dışı değişkenler için $c_j - z_j$ bulunur. Burada C_B ; temel değişkenlerin fiyatı, y_j ise j'inci temel dışı değişkenine ait katsayılarından oluşmaktadır (sütun vektörü). Her j için $c_j - z_j \geq 0$ (maksimum problemi ise $c_j - z_j \leq 0$) ise en iyi çözüm bulunmuştur. Negatif $c_j - z_j$ değerleri arasından mutlak değerce en büyüğüne karşı gelen x_k değişkeni temele girecek değişken olarak belirlenir. Maksimum probleminde ise en büyük $c_j - z_j$ değerine karşı gelen değişken temele alınır.

3. *Adım:* $\bar{x}_B$, temel değişkenlerin sayısal değerleri ve y_k, x_k ya karşı gelen sütun olmak üzere $\min \left\{ \frac{\bar{x}_B}{y_{Bk}} y_{Bk} > 0 \right\} = \frac{\bar{x}_i}{y_{ik}}$ ilişkisi ile temelden çıkacak x_i değişkeni bulunur.

4. *Adım:* x_i değişkeni ile x_k değişkeni yer değiştirilir. Yani B temel matrisi içindeki x_i ye karşı gelen sütun ile R matrisi içindeki x_k ya karşı gelen sütun yer değiştirilerek,

(aynı zamanda C_B ve x_B de de uygun deęişiklik yapılır) elemanter satır ve sütun işlemleriyle yeni B matrisi birim matris haline getirilir. Böylece $\bar{x}_B$ çözümü de (B_b^{-1}) bulunur daha sonrada $z_j = C_B y_j$ eşitliğinden, z_j deęerleri hesaplanır ve $c_j - z_j$ deęerleri bulunur ve ikinci adıma gidilerek işlemler sürdürölür. Bir DP probleminin verilen bu algoritmadan hareketle tablo kullanılarak çözümü daha pratiktir. Bu çözümde kullanılan tablolara simpleks tablosu denilmektedir. Çizelge 4.1.'de verilen bir problemin ilk simpleks tablosu aşığıdaki gibi kurulabilir.

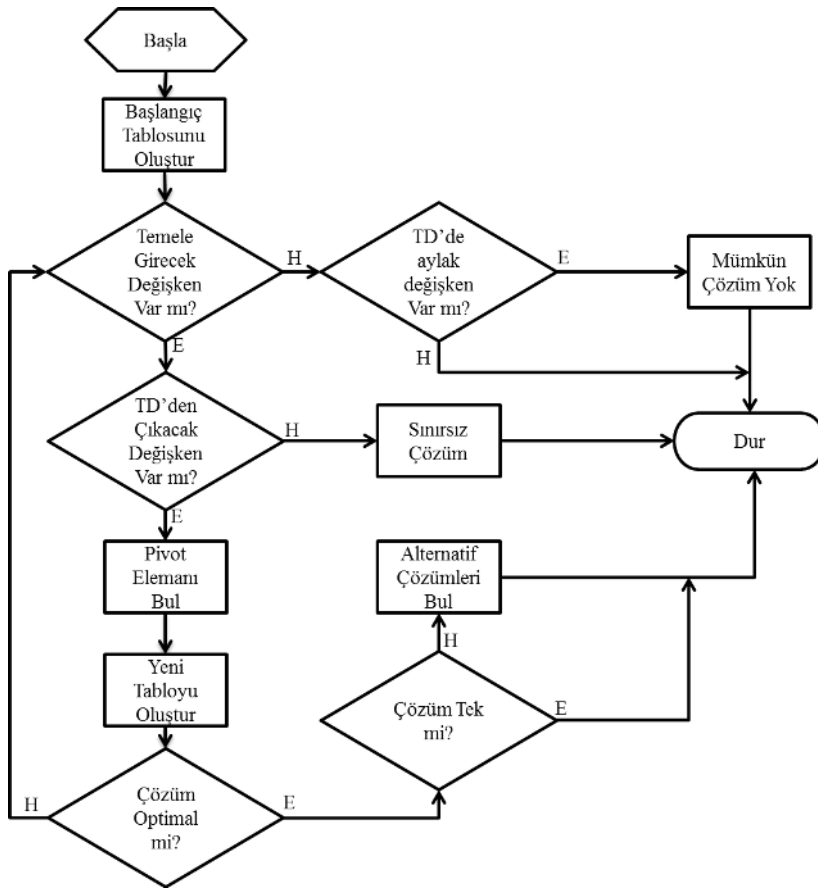
Çizelge 4.1. Temel başlangıç simpleks tablosu

			C_R	C_B
C_B	X_B	$\bar{x}_B$	x_R	x_B
C_1	x_1			
$\vdots$	$\vdots$	B_b^{-1}	$B^{-1}R = y_j$	$B^{-1}B = I$
C_m	x_m			
			$C_j - Z_j$	

Yukarda verilen simpleks tablosunun ilk sütunu temele giren deęişkenlerin amaç fonksiyonundaki katsayıları, fiyatları oluşturur. İkinci sütun temel deęişkenleri, üçüncü sütun bu deęişkenlerin aldıkları deęerleri gösterir. Tablonun dięer sütunları temel dışı deęişkenlerin matrisi $B^{-1}R$ ile temel deęişkenler I matrisinden oluşur. $C_j - Z_j$ ise her j inci, temel dışı deęişkenlerin marjinal katkısını gösterir. C_R temel dışı deęişkenlerin fiyatını gösterir.

4.6. Simpleks Yöntemin Akış Şeması

Simpleks yöntem için uygulanan akış şeması Şekil 5.5.'te gösterildięi gibidir. KDS şekilde gösterilen akış şemasına göre geliştirilmiştir.



Şekil 4.1. Simpleks yöntemine ait akış şeması

4.7. Büyük M Metodu

Eğer kısıtlamalar " $\leq$ " yerine " $\geq$ " olsaydı, standart formata geçirildiğinde, eklenecek değişkenlerin katsayıları "-1" olacak ve eşitliklerin sağ tarafının pozitif olması şartı sebebiyle temel değişkenlerin seçiminde eklenen değişkenler kullanılamayacaktı. Bu durumda sisteme yeni yapay değişkenler eklenerek problem çözülebilir. Yapay değişkenlerin çözüme etki etmemesi için, çözümde bu değişkenlerin 0 olmasının sağlanması gerekmektedir. Bunu yapmak için amaç fonksiyonuna "M" olarak göstereceğimiz büyük bir katsayı ile eklenir ve optimum çözümde sistemin bu değişkenleri 0 yapması sağlanır.

Örneğin

$$\min Z = 4x_1 + x_2$$

$$3x_1 + x_2 = 3$$

$$4x_1 + 3x_2 \geq 6$$

$$x_1 + x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

Standart formda yazılınca:

$$\min Z = 4x_1 + x_2$$

$$3x_1 + x_2 = 3$$

$$4x_1 + 3x_2 - x_3 = 6$$

$$x_1 + x_2 + x_4 = 4$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Birinci ve ikinci satırların istenilen değişkenleri bulunmamaktadır. Burada iki yapay değişken eklenerek sorun giderilir.

$$\min Z = 4x_1 + x_2 + MR_1 + MR_2$$

Kısıtlamalar:

$$3x_1 + x_2 + R_1 = 3$$

$$4x_1 + 3x_2 - x_3 + R_2 = 6$$

$$x_1 + x_2 + x_4 = 4$$

$$x_1, x_2, x_3, x_4, R_1, R_2 \geq 0$$

Tabloyu oluştururken amaç satırında temel değişkenlerin katsayısının 0 olması gerekmektedir. Bunu başarmak için R_1 ve R_2 eşitliklerden çekilerek amaç fonksiyonunda yerine konulur ve yeni amaç fonksiyonu bulunur.

$$R_1 = 3 - 3x_1 - x_2$$

$$R_2 = 6 - 4x_1 - 3x_2 + x_3$$

Amaç fonksiyonu

$$Z = 4x_1 + x_2 + M(3 - 3x_1 - x_2) + M(6 - 4x_1 - 3x_2 + x_3)$$

$$Z = (4 - 7M)x_1 - (1 - 4M)x_2 - Mx_3 + 9M$$

Tabloda görünecek şekli:

$$Z - (4 - 7M)x_1 - (1 - 4M)x_2 - Mx_3 = 9M$$

Dikkat edilmesi gereken bir diğer olay da problemin en küçükleme problemi olduğudur. Bu durumda giriş değişkeni seçilirken en küçük olan değil, en büyük katsayısı olan seçilecektir. Problem hiç pozitif katsayı kalmayınca duracaktır.

$$\begin{aligned}
 \max Z &= 3x_1 + x_2 & \max Z &= 3x_1 + x_2 + 0s_1 - MY_1 - MY_2 \\
 x_1 + 2x_2 &\geq 5 & x_1 + 2x_2 - s_1 + Y_1 &= 5 \\
 2x_1 + x_2 &\leq 6 & 2x_1 + x_2 + s_2 &= 6 \\
 x_1 + x_2 &= 3 & x_1 + x_2 + Y_2 &= 3 \\
 x_1, x_2 &\geq 0 & x_1, x_2, s_1, Y_1, Y_2 &\geq 0
 \end{aligned}$$

Çizelge 4.2. Örnek için başlangıç simpleks tablo

C_j			3	1	0	0	-M	-M
	Değişken	Nicelik	x_1	x_2	s_1	s_2	Y_1	Y_2
-M	Y_1	5	1	2	-1	0	1	0
0	s_2	6	2	1	0	1	0	0
-M	Y_2	3	1	1	0	0	0	1
Z_j		-8M	-2M	-3M	M	0	-M	-M
$C_j - Z_j$			3+2M	1+3M	-M	0	0	0

Çizelge 4.3. Örnek için birinci simpleks tablo

C_j			3	1	0	0	-M	-M
	Değişken	Nicelik	x_1	x_2	s_1	s_2	Y_1	Y_2
1	x_2	5/2	1/2	1	-1/2	0	1/2	0
0	s_2	7/2	3/2	0	1/2	1	-1/2	0
-M	Y_2	1/2	1/2	0	1/2	0	-1/2	1
Z_j		5/2-M/2	1/2-M/2	1	-1/2-M/2	0	1/2+M/2	-M
$C_j - Z_j$			5/2+M/2	0	1/2+M/2	0	-1/2-3/2M	0

Çizelge 4.4. Örnek için ikinci simpleks tablo (Optimal)

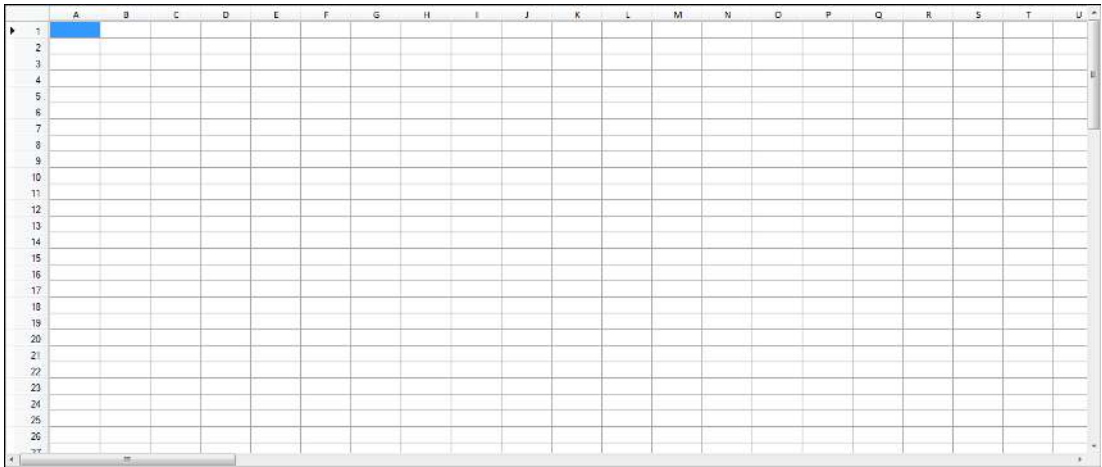
C_j			3	1	0	0	-M	-M
	Değişken	Nicelik	x_1	x_2	s_1	s_2	Y_1	Y_2
1	x_2	2	0	1	-1	0	1	-1
0	s_2	2	0	0	-1	1	1	-3
3	x_1	1	1	0	1	0	-1	2
Z_j		5	3	1	2	0	-2	5
$C_j - Z_j$			0	0	-2	0	-M+2	-M-5

5. KARAR DESTEK SİSTEMİ

5.1. Veri Giriş Arayüzü

Geliştirilen bu KDS’de, kullanıcıların aşina oldukları SPSS, Excel gibi birçok paket programın kullandığı veri giriş arayüzü tercih edilmiştir. Bu sayede diğer ortamlardan veri alışverişi kolayca sağlanacak ve kullanıcı tarafından daha hızlı benimsenecektir. Veri giriş arayüzü, DataGridView (Microsoft Visual C#) kullanılarak geliştirilmiştir.

Resim 5.1.’de veri giriş arayüzüne ait ekran görüntüsü verilmiş olup, analizin sağlıklı bir şekilde yapılabilmesi için satırlarda KVB’lerinin isimlerini sütunlarda ise girdi ve çıktı değişkenlerinin değerlerinin bulunması gerekmektedir.



Resim 5.1. Veri giriş arayüzüne ait ekran görüntüsü

KDS, 4000 KVB için 655 girdi-çıktı verilerini kullanabilecek düzeyde ve tercih edilen VZA modelini çözebilecek şekilde geliştirilmiştir. KVB sayısı ile girdi çıktı sayılarını artırmak mümkündür, ancak KDS’nin daha hızlı ve daha verimli çalışması için belirli bir sayı ile sınırlandırılmıştır.

Veri giriş arayüzüne üç farklı şekilde veri aktarılabilir. Bunlar manuel olarak veri girişi, Excel ile veri girişi ve Not defteri ile veri giriştir.

5.1.1. Manuel veri girişı

KDS'nin kendi veri giriř arayüzüne klavye aracılığı ile manuel olarak veri giriř imkânı sağlanmaktadır. Resim 5.2.'de gösterildiđi gibi KVB'lerini ilk sütuna her bir satıra bir KVB olacak şekilde KVB isimlerini ardından ise ilk satıra ikinci sütundan başlayarak girdi ve çıktı isimlerini yazılmalıdır. Burada dikkat edilecek husus girdi ve çıktı deđişkenlerinin tespit edilmesi ilk harfin deđerine göre yapıldıđı için mutlaka ilk harf ya girdi için "g/G" ya da çıktı için "ç/Ç" olmalıdır. Örneđin üç girdi iki çıktı olan probleme ait girdi çıktı isimleri řu şekilde olabilir; "Çıktıfıyat", "çık", "girdi33", "GİRDİ", "girdison". Veri giriři esnasında girdi(ler)in veya çıktı(lar)ın hangisinin ilk yazıldıđının hiçbir önemi yoktur. İlk harfe göre ayırım yapıldıđı için isteđe bađlı olarak karışık dahi yazmak mümkündür.

	A	B	C	D	E	F	G	H	I	J	K
1	KVB	G-Cost	G-Clients	Ç-Revenue							
2	F1	152.65	128.63	122.85							
3	F2	138.92	157.66	153.48							
4	F3	137.19	106.78	131.6							
5	F4	74.9	63.25	81.31							
6	F5	98.47	117.57	98.56							
7	F6	91.93	155.2	98.22							
8	F7	74.51	61.14	62.56							
9	F8	91.06	235.15	109.85							
10	F9	64.7	51.73	50.86							
11	F10	72.92	81.61	81.25							
12	F11	190.01	179.52	203.5							
13	F12	83.32	107.19	86.35							
14	F13	68.63	117.57	76.85							
15	F14	95.36	79.95	101.17							

Resim 5.2. Manuel veri giriř arayüzü (Ek 3, Çizelge 3.1)

5.1.2. Excel ile veri giriři

Resim 5.3.'te gösterildiđi gibi Excel ortamında tutulan veriler menüler yardımıyla KDS veri giriř arayüzüne aktarılmaktadır. Excel'den veri aktarılırken dikkat edilecek

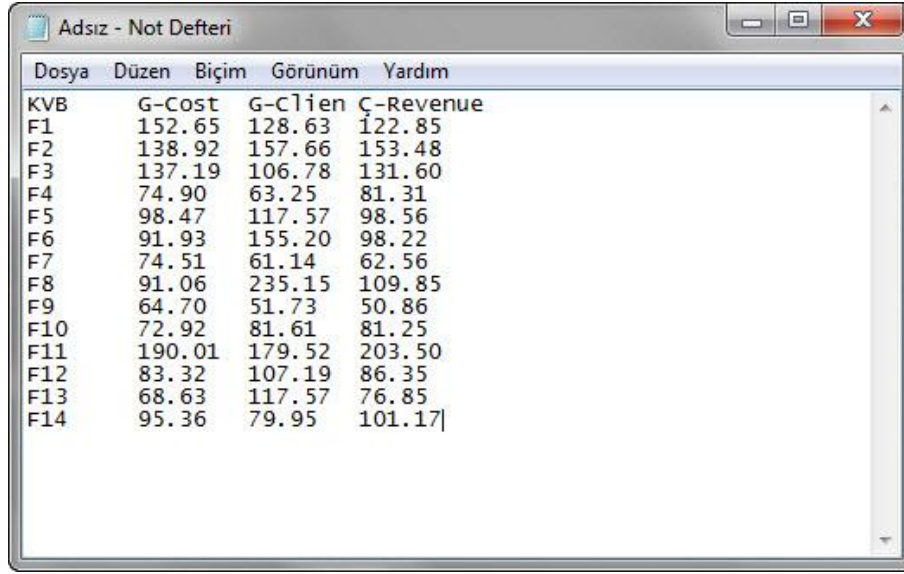
husus ise çalışma sayfasının adı Sayfa1 olmalıdır. Ayrıca MS Office 97-2003 sürümlerini (*.xls) ve 2007 sonrası sürümlerini (*.xlsx) desteklemektedir.

	A	B	C	D	E	F	G	H	I
1	KVB	G-Cost	G-Clients	Ç-Revenue					
2	F1	152.65	128.63	122.85					
3	F2	138.92	157.66	153.48					
4	F3	137.19	106.78	131.60					
5	F4	74.90	63.25	81.31					
6	F5	98.47	117.57	98.56					
7	F6	91.93	155.20	98.22					
8	F7	74.51	61.14	62.56					
9	F8	91.06	235.15	109.85					
10	F9	64.70	51.73	50.86					
11	F10	72.92	81.61	81.25					
12	F11	190.01	179.52	203.50					
13	F12	83.32	107.19	86.35					
14	F13	68.63	117.57	76.85					
15	F14	95.36	79.95	101.17					

Resim 5.3. Excel'deki verilere ait ekran görüntüleri (Ek 3, Çizelge 3.1)

5.1.3. Not Defteri ile veri girişi

Klasik Notepad olarak bilinen Not Defteri programından da veri almak mümkündür. Not defterinde tutulan verilerin “tab” tuşu ile sekmelere ayrılmasına dikkat edilerek KDS'ne aktarma işlemi yapılabilir.



Dosya	Düzen	Biçim	Görünüm	Yardım
KVB	G-Cost	G-Clıen	Ç-Revenue	
F1	152.65	128.63	122.85	
F2	138.92	157.66	153.48	
F3	137.19	106.78	131.60	
F4	74.90	63.25	81.31	
F5	98.47	117.57	98.56	
F6	91.93	155.20	98.22	
F7	74.51	61.14	62.56	
F8	91.06	235.15	109.85	
F9	64.70	51.73	50.86	
F10	72.92	81.61	81.25	
F11	190.01	179.52	203.50	
F12	83.32	107.19	86.35	
F13	68.63	117.57	76.85	
F14	95.36	79.95	101.17	

Resim 5.4. Not defteri'ndeki verilere ait ekran görüntüleri (Ek 3, Çizelge 3.1)

Excel ve Not defteri yardımıyla aktarılan veriler manuel olarak girilmiş gibi Resim 5.2.'de gösterilen veri giriş arayüzünde ki gibi gösterilecek bu sayede yanlışlık var ise manuel olarak düzeltilebilecektir.

5.2. VZA Modellerinin Simpleks Başlangıç Tablosu

VZA'nın bütün modellerinin simpleks tablo ile çözülebilmesi için başlangıç tablosu oluşturmak gerekmektedir. Yedi KVB için girdi değişkenleri sırasıyla 2, 3, 6, 9, 5, 4, 10 ve çıktı değişkenleri 2, 5, 7, 8, 3, 1, 7 olan VZA modelleri aşağıdaki gibi oluşturulmaktadır.

Girdi yönlü CCR primal modeli (CCRP-I)

Üçüncü bölümde bahsedilen girdi yönlü CCR Primal modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır. Sağ tarafta ise standart form haline getirilmesi gösterilmektedir.

$$\begin{array}{l|l}
E_k = \max 2\mu_1 & E_k = \max 2\mu_1 + 0Y_1 + 0Y_2 + 0Y_3 + 0Y_4 + 0Y_5 + 0Y_6 \\
& + 0Y_7 - Mq_1 \\
2\mu_1 - 2v_1 \leq 0 & 2\mu_1 - 2v_1 + Y_1 = 0 \\
5\mu_1 - 3v_1 \leq 0 & 5\mu_1 - 3v_1 + Y_2 = 0 \\
7\mu_1 - 6v_1 \leq 0 & 7\mu_1 - 6v_1 + Y_3 = 0 \\
8\mu_1 - 9v_1 \leq 0 & 8\mu_1 - 9v_1 + Y_4 = 0 \\
3\mu_1 - 5v_1 \leq 0 & 3\mu_1 - 5v_1 + Y_5 = 0 \\
1\mu_1 - 4v_1 \leq 0 & 1\mu_1 - 4v_1 + Y_6 = 0 \\
7\mu_1 - 10v_1 \leq 0 & 7\mu_1 - 10v_1 + Y_7 = 0 \\
2v_1 = 1 & 2v_1 + q_1 = 1
\end{array}$$

Çizelge 5.1. Girdi yönlü CCR primal model için başlangıç simpleks tablo

			2	0	0	0	0	0	0	0	0	0	-M
C_j	Değişken	Nicelik	μ_1	v_1	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	q_1	
0	Y_1	0	2	-2	1	0	0	0	0	0	0	0	0
0	Y_2	0	5	-3	0	1	0	0	0	0	0	0	0
0	Y_3	0	7	-6	0	0	1	0	0	0	0	0	0
0	Y_4	0	8	-9	0	0	0	1	0	0	0	0	0
0	Y_5	0	3	-5	0	0	0	0	1	0	0	0	0
0	Y_6	0	1	-4	0	0	0	0	0	1	0	0	0
0	Y_7	0	7	-10	0	0	0	0	0	0	1	0	0
-M	q_1	1	0	2	0	0	0	0	0	0	0	0	1
Z_j													
$C_j - Z_j$													

Girdi yönlü CCR dual modeli (CCR_D-I)

Üçüncü bölümde bahsedilen girdi yönlü CCR Dual modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır.

$$\begin{aligned}
E_k &= \min \theta - \varepsilon S_1^+ - \varepsilon S_1^- \\
2\lambda_1 + 5\lambda_2 + 7\lambda_3 + 8\lambda_4 + 3\lambda_5 + 1\lambda_6 + 7\lambda_7 - S_1^+ &= 2 \\
2\theta - 2\lambda_1 - 3\lambda_2 - 6\lambda_3 - 9\lambda_4 - 5\lambda_5 - 4\lambda_6 - 10\lambda_7 - S_1^- &= 0 \\
\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^- &\geq 0
\end{aligned}$$

Girdi yönlü CCR dual modeli standart form haline getirildiğinde ise aşağıdaki gibi olacaktır.

$$E_k = \min \theta - \varepsilon S_1^+ - \varepsilon S_1^- + Mq_1 + Mq_2$$

$$2\lambda_1 + 5\lambda_2 + 7\lambda_3 + 8\lambda_4 + 3\lambda_5 + 1\lambda_6 + 7\lambda_7 - S_1^+ + q_1 = 2$$

$$2\theta - 2\lambda_1 - 3\lambda_2 - 6\lambda_3 - 9\lambda_4 - 5\lambda_5 - 4\lambda_6 - 10\lambda_7 - S_1^- + q_2 = 0$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^-, q_1, q_2 \geq 0$$

Çizelge 5.2. Girdi yönlü CCR dual model için başlangıç simpleks tablo

			1	0	0	0	0	0	0	0	$-\varepsilon$	$-\varepsilon$	M	M
C_j	Değişken	Nicelik	θ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	S_1^+	S_1^-	q_1	q_2
M	q_1	2	0	2	5	7	8	3	1	7	-1	0	1	0
M	q_2	0	2	-2	-3	-6	-9	-5	-4	-10	0	-1	0	1
Z_j														
$C_j - Z_j$														

Çıktı yönlü CCR primal modeli (CCR_{P-O})

Üçüncü bölümde bahsedilen çıktı yönlü CCR Primal modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır. Sağ tarafta ise standart form haline getirilmesi gösterilmektedir.

$E_k = \min 2v_1$	$E_k = \min 2v_1 + 0Y_1 + 0Y_2 + 0Y_3 + 0Y_4 + 0Y_5 + 0Y_6 + 0Y_7 + Mq_1$
$2\mu_1 - 2v_1 \leq 0$	$2\mu_1 - 2v_1 + Y_1 = 0$
$5\mu_1 - 3v_1 \leq 0$	$5\mu_1 - 3v_1 + Y_2 = 0$
$7\mu_1 - 6v_1 \leq 0$	$7\mu_1 - 6v_1 + Y_3 = 0$
$8\mu_1 - 9v_1 \leq 0$	$8\mu_1 - 9v_1 + Y_4 = 0$
$3\mu_1 - 5v_1 \leq 0$	$3\mu_1 - 5v_1 + Y_5 = 0$
$1\mu_1 - 4v_1 \leq 0$	$1\mu_1 - 4v_1 + Y_6 = 0$
$7\mu_1 - 10v_1 \leq 0$	$7\mu_1 - 10v_1 + Y_7 = 0$
$2\mu_1 = 1$	$2\mu_1 + q_1 = 1$

Çizelge 5.3. Çıktı yönlü CCR primal model için başlangıç simpleks tablo

			0	2	0	0	0	0	0	0	0	0	M
C_j	Değişken	Nicelik	μ_1	v_1	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	q_1	
0	Y_1	0	2	-2	1	0	0	0	0	0	0	0	
0	Y_2	0	5	-3	0	1	0	0	0	0	0	0	
0	Y_3	0	7	-6	0	0	1	0	0	0	0	0	
0	Y_4	0	8	-9	0	0	0	1	0	0	0	0	
0	Y_5	0	3	-5	0	0	0	0	1	0	0	0	
0	Y_6	0	1	-4	0	0	0	0	0	1	0	0	
0	Y_7	0	7	-10	0	0	0	0	0	0	1	0	
M	q_1	1	2	0	0	0	0	0	0	0	0	0	1
Z_j													
$C_j - Z_j$													

Çıktı yönlü CCR dual modeli (CCR_D-O)

Üçüncü bölümde bahsedilen çıktı yönlü CCR Dual modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır.

$$E_k = \max \phi + \varepsilon S_1^+ + \varepsilon S_1^-$$

$$2\phi - 2\lambda_1 - 5\lambda_2 - 7\lambda_3 - 8\lambda_4 - 3\lambda_5 - 1\lambda_6 - 7\lambda_7 + S_1^+ = 0$$

$$2\lambda_1 + 3\lambda_2 + 6\lambda_3 + 9\lambda_4 + 5\lambda_5 + 4\lambda_6 + 10\lambda_7 + S_1^- = 2$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^- \geq 0$$

Çıktı yönlü CCR dual modeli standart form haline getirildiğinde ise aşağıdaki gibi olacaktır.

$$E_k = \max \phi + \varepsilon S_1^+ + \varepsilon S_1^- - Mq_1 - Mq_2$$

$$2\phi - 2\lambda_1 - 5\lambda_2 - 7\lambda_3 - 8\lambda_4 - 3\lambda_5 - 1\lambda_6 - 7\lambda_7 + S_1^+ + q_1 = 0$$

$$2\lambda_1 + 3\lambda_2 + 6\lambda_3 + 9\lambda_4 + 5\lambda_5 + 4\lambda_6 + 10\lambda_7 + S_1^- + q_2 = 2$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^-, q_1, q_2 \geq 0$$

Çizelge 5.4. Çıktı yönlü CCR dual model için başlangıç simpleks tablo

			1	0	0	0	0	0	0	0	ε	ε	-M	-M
C_j	Değişken	Nicelik	ϕ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	S_1^+	S_1^-	q_1	q_2
-M	q_1	0	2	-2	-5	-7	-8	-3	-1	-7	1	0	1	0
-M	q_2	2	0	2	3	6	9	5	4	10	0	1	0	1
Z_j														
$C_j - Z_j$														

Girdi yönlü BCC primal model (BCC_{P-I})

Üçüncü bölümde bahsedilen girdi yönlü BCC Primal modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır. Sağ tarafta ise standart form haline getirilmesi gösterilmektedir.

$$E_k = \max 2\mu_1 + U_0$$

$$2\mu_1 - 2v_1 + U_0 \leq 0$$

$$5\mu_1 - 3v_1 + U_0 \leq 0$$

$$7\mu_1 - 6v_1 + U_0 \leq 0$$

$$8\mu_1 - 9v_1 + U_0 \leq 0$$

$$3\mu_1 - 5v_1 + U_0 \leq 0$$

$$1\mu_1 - 4v_1 + U_0 \leq 0$$

$$7\mu_1 - 10v_1 + U_0 \leq 0$$

$$2v_1 = 1$$

$$E_k = \max 2\mu_1 + U_0 + 0Y_1 + 0Y_2 + 0Y_3 + 0Y_4$$

$$+ 0Y_5 + 0Y_6 + 0Y_7 - Mq_1$$

$$2\mu_1 - 2v_1 + U_0 + Y_1 = 0$$

$$5\mu_1 - 3v_1 + U_0 + Y_2 = 0$$

$$7\mu_1 - 6v_1 + U_0 + Y_3 = 0$$

$$8\mu_1 - 9v_1 + U_0 + Y_4 = 0$$

$$3\mu_1 - 5v_1 + U_0 + Y_5 = 0$$

$$1\mu_1 - 4v_1 + U_0 + Y_6 = 0$$

$$7\mu_1 - 10v_1 + U_0 + Y_7 = 0$$

$$2v_1 + q_1 = 1$$

Çizelge 5.5. Girdi yönlü BCC primal model için başlangıç simpleks tablo

			2	0	1	0	0	0	0	0	0	0	-M
C_j	Değişken	Nicelik	μ_1	v_1	U_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	q_1
0	Y_1	0	2	-2	1	1	0	0	0	0	0	0	0
0	Y_2	0	5	-3	1	0	1	0	0	0	0	0	0
0	Y_3	0	7	-6	1	0	0	1	0	0	0	0	0
0	Y_4	0	8	-9	1	0	0	0	1	0	0	0	0
0	Y_5	0	3	-5	1	0	0	0	0	1	0	0	0
0	Y_6	0	1	-4	1	0	0	0	0	0	1	0	0
0	Y_7	0	7	-10	1	0	0	0	0	0	0	1	0
-M	q_1	1	0	2	0	0	0	0	0	0	0	0	1
Z_j													
$C_j - Z_j$													

Girdi yönlü BCC dual model (BCC_{D-I})

Üçüncü bölümde bahsedilen girdi yönlü BCC Dual modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır.

$$E_k = \min \theta - \varepsilon S_1^+ - \varepsilon S_1^-$$

$$2\lambda_1 + 5\lambda_2 + 7\lambda_3 + 8\lambda_4 + 3\lambda_5 + 1\lambda_6 + 7\lambda_7 - S_1^+ = 2$$

$$2\theta - 2\lambda_1 - 3\lambda_2 - 6\lambda_3 - 9\lambda_4 - 5\lambda_5 - 4\lambda_6 - 10\lambda_7 - S_1^- = 0$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 = 1$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^- \geq 0$$

Kurulan model standart form haline getirildiğinde ise aşağıdaki gibi olacaktır.

$$E_k = \min \theta - \varepsilon S_1^+ - \varepsilon S_1^- + Mq_1 + Mq_2 + Mq_3$$

$$2\lambda_1 + 5\lambda_2 + 7\lambda_3 + 8\lambda_4 + 3\lambda_5 + 1\lambda_6 + 7\lambda_7 - S_1^+ + q_1 = 2$$

$$2\theta - 2\lambda_1 - 3\lambda_2 - 6\lambda_3 - 9\lambda_4 - 5\lambda_5 - 4\lambda_6 - 10\lambda_7 - S_1^- + q_2 = 0$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + q_3 = 1$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^-, q_1, q_2, q_3 \geq 0$$

Çizelge 5.6. Girdi yönlü BCC dual model için başlangıç simpleks tablo

			1	0	0	0	0	0	0	0	$-\varepsilon$	$-\varepsilon$	M	M	M
C_j	Değişken	Nicelik	θ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	S_1^+	S_1^-	q_1	q_2	q_3
M	q_1	2	0	2	5	7	8	3	1	7	-1	0	1	0	0
M	q_2	0	2	-2	-3	-6	-9	-5	-4	-10	0	-1	0	1	0
M	q_3	1	0	1	1	1	1	1	1	1	0	0	0	0	1
Z_j															
$C_j - Z_j$															

Çıktı yönlü BCC primal modeli (BCC_{P-O})

Üçüncü bölümde bahsedilen çıktı yönlü BCC Primal modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır. Sağ tarafta ise standart form haline getirilmesi gösterilmektedir.

$$E_k = \min 2v_1 + U_0$$

$$2\mu_1 - 2v_1 - U_0 \leq 0$$

$$5\mu_1 - 3v_1 - U_0 \leq 0$$

$$7\mu_1 - 6v_1 - U_0 \leq 0$$

$$8\mu_1 - 9v_1 - U_0 \leq 0$$

$$3\mu_1 - 5v_1 - U_0 \leq 0$$

$$1\mu_1 - 4v_1 - U_0 \leq 0$$

$$7\mu_1 - 10v_1 - U_0 \leq 0$$

$$2\mu_1 = 1$$

$$\min q_0 = 2v_1 + U_0 + 0Y_1 + 0Y_2 + 0Y_3 + 0Y_4 + 0Y_5 + 0Y_6 + 0Y_7 + Mq_1$$

$$2\mu_1 - 2v_1 - U_0 + Y_1 = 0$$

$$5\mu_1 - 3v_1 - U_0 + Y_2 = 0$$

$$7\mu_1 - 6v_1 - U_0 + Y_3 = 0$$

$$8\mu_1 - 9v_1 - U_0 + Y_4 = 0$$

$$3\mu_1 - 5v_1 - U_0 + Y_5 = 0$$

$$1\mu_1 - 4v_1 - U_0 + Y_6 = 0$$

$$7\mu_1 - 10v_1 - U_0 + Y_7 = 0$$

$$2\mu_1 + q_1 = 1$$

Çizelge 5.7. Çıktı yönlü BCC primal model için başlangıç simpleks tablo

			0	2	1	0	0	0	0	0	0	0	M
C_j	Değişken	Nicelik	μ_1	v_1	U_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	q_1
0	Y_1	0	2	-2	-1	1	0	0	0	0	0	0	0
0	Y_2	0	5	-3	-1	0	1	0	0	0	0	0	0
0	Y_3	0	7	-6	-1	0	0	1	0	0	0	0	0
0	Y_4	0	8	-9	-1	0	0	0	1	0	0	0	0
0	Y_5	0	3	-5	-1	0	0	0	0	1	0	0	0
0	Y_6	0	1	-4	-1	0	0	0	0	0	1	0	0
0	Y_7	0	7	-10	-1	0	0	0	0	0	0	1	0
M	q_1	1	2	0	0	0	0	0	0	0	0	0	1
Z_j													
$C_j - Z_j$													

Çıktı yönlü BCC dual model (BCC_D-O)

Üçüncü bölümde bahsedilen çıktı yönlü BCC Dual modelin matematiksel formuna göre örnek model aşağıdaki gibi kurulmaktadır.

$$E_k = \max \phi + \varepsilon S_1^+ + \varepsilon S_1^-$$

$$2\phi - 2\lambda_1 - 5\lambda_2 - 7\lambda_3 - 8\lambda_4 - 3\lambda_5 - 1\lambda_6 - 7\lambda_7 + S_1^+ = 0$$

$$2\lambda_1 + 3\lambda_2 + 6\lambda_3 + 9\lambda_4 + 5\lambda_5 + 4\lambda_6 + 10\lambda_7 + S_1^- = 2$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 = 1$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^- \geq 0$$

Çıktı yönlü BCC dual modeli standart form haline getirildiğinde ise aşağıdaki gibi olacaktır.

$$E_k = \max \phi + \varepsilon S_1^+ + \varepsilon S_1^- - Mq_1 - Mq_2 - Mq_3$$

$$2\phi - 2\lambda_1 - 5\lambda_2 - 7\lambda_3 - 8\lambda_4 - 3\lambda_5 - 1\lambda_6 - 7\lambda_7 + S_1^+ + q_1 = 0$$

$$2\lambda_1 + 3\lambda_2 + 6\lambda_3 + 9\lambda_4 + 5\lambda_5 + 4\lambda_6 + 10\lambda_7 + S_1^- + q_2 = 2$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + q_3 = 1$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, S_1^+, S_1^-, q_1, q_2, q_3 \geq 0$$

Çizelge 5.8. Çıktı yönlü BCC dual model için başlangıç simpleks tablo

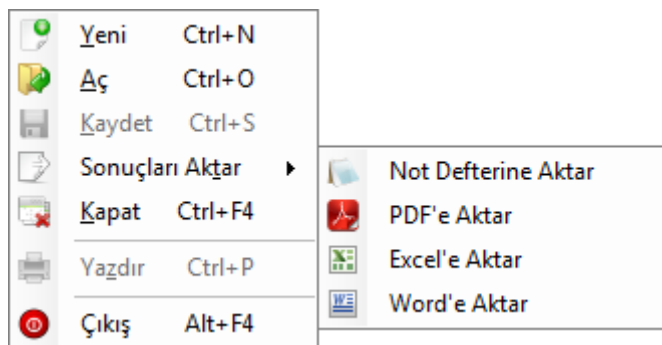
			1	0	0	0	0	0	0	0	ε	ε	-M	-M	-M
C_j	Değişken	Nicelik	ϕ	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	S_1^+	S_1^-	q_1	q_2	q_3
-M	q_1	0	2	-2	-5	-7	-8	-3	-1	-7	1	0	1	0	0
-M	q_2	2	0	2	3	6	9	5	4	10	0	1	0	1	0
-M	q_3	1	0	1	1	1	1	1	1	1	0	0	0	0	1
	Z_j														
	$C_j - Z_j$														

5.3. Menüler

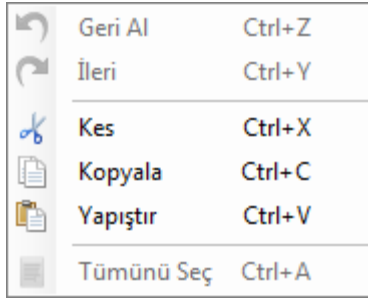
Geliştirilen KDS'ne ait altı ana menü bulunmaktadır. Veri aktarma işleminin yapılacağı Resim 5.6.'da gösterilen dosya menüsü, veri düzenleme ile ilgili işlemlerin yapılacağı Resim 5.7.'de gösterilen düzen menüsü, menü ve durum çubuğunu göster/gizle seçeneği sunan Resim 5.8.'de gösterilen görünüm menüsü bulunmaktadır. Bunların yanında VZA modellerinin uygulanmasına ait Resim 5.9.'de gösterilen analiz menüsü ve KDS'ye ait pencereler ile yardım menüleri de Resim 5.10. ve Resim 5.11.'de gösterildiği gibi bulunmaktadır. KDS'ne sonraki versiyonlarında yer alması düşünülen bazı aktif olmayan menüler önceden eklenmiştir.



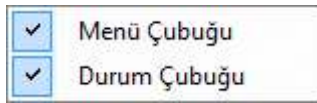
Resim 5.5. Menülere ait ekran görüntüsü



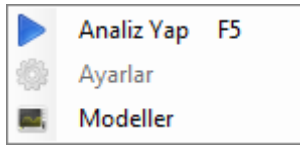
Resim 5.6. Dosya menüsü ve aktarma seçenekleri



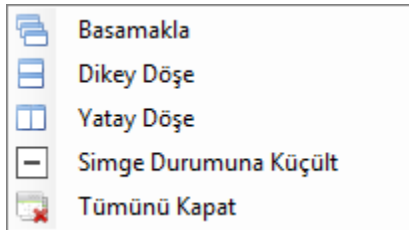
Resim 5.7. Düzen menüsü



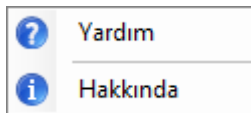
Resim 5.8. Görünüm menüsü



Resim 5.9. Analiz menüsü



Resim 5.10. Görünüm menüsü



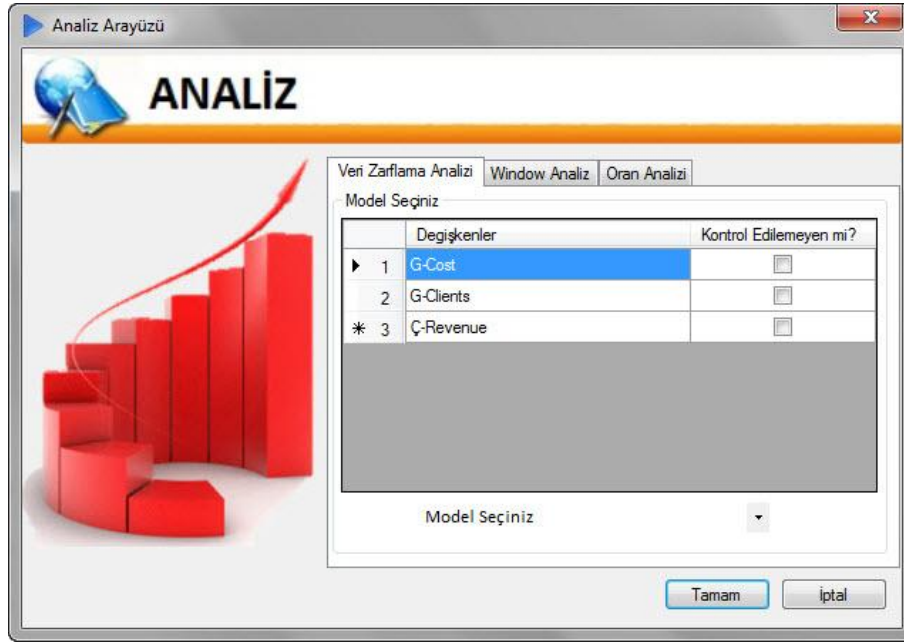
Resim 5.11. Yardım menüsü

5.4. Analiz ve Rapor Arayüzü

5.4.1. Panel olmayan veriler için analiz

Dış ortamlardan veya manuel olarak veri giriş arayüzü aracılığı ile girilen veriler Resim 5.12.'de gösterilen analiz arayüzü ile performans analizi yapılmaktadır.

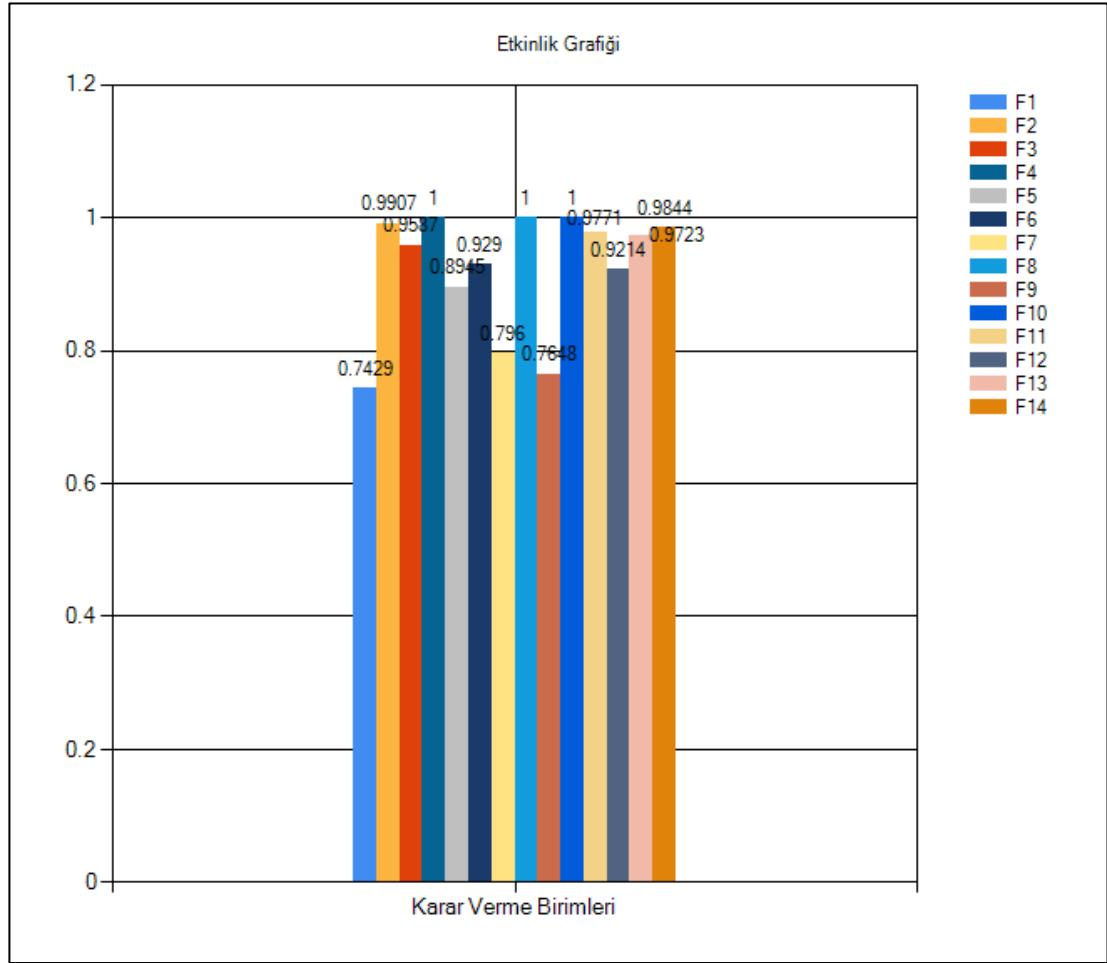
Kullanıcı tarafından seçilen modele ait arka planda algoritmalar çalışacak ve model kurulup çözülecektir. Problemin çözme işleminin ardından etkinlik skorlarının gösterileceği Resim 5.13.'te gösterilen rapor ekranı kullanıcının bilgisine sunulacaktır.



Resim 5.12. Analiz arayüzü

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Karar Verme Birimleri :	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
2	Etkinlik Değerleri :	0.7429	0.9907	0.9587	1	0.8945	0.929	0.796	1	0.7648	1	0.9771	0.9214	0.9723	0.9844
3	Referanslar Olma Sıklığı :				6				5		6				
4	Referanslar :	F4 (1.51)	F8 (0.02)	F4 (1.62)		F8 (0.05)	F8 (0.36)	F4 (0.77)		F4 (0.63)		F4 (1.57)	F8 (0.1)	F8 (0.3)	F4 (1.24)
5			F10 (1.87)			F10 (1.15)	F10 (0.72)					F10 (0.93)	F10 (0.93)	F10 (0.54)	
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
* 19															

Resim 5.13. Rapor arayüzü (Ek 3, Çizelge 3.2)



Resim 5.14. Analiz sonucu etkinlik grafiği

Resim 5.13.'te gösterilen rapor arayüzünde her bir sütun KVB'lerine ait etkinlik değerlerini ve referans kümeleri göstermektedir. 14 KVB içinde etkinlik değeri 1 çıkan F4, F8 ve F10 KVB'leri etkin olarak tespit edilmiştir. İlgili değerler kullanıcı tarafından yoruma açık bir şekilde sunulmaktadır. Aynı zamanda Resim 5.14.'de etkinlik skorlarının grafiği kullanıcıya sunulmaktadır.

5.4.2. Panel veriler için analiz

Eğer veriler belli bir periyotta ise yani Resim 5.15.'de gösterildiği gibi ise panel veriler için kullanılan window analizi arayüzü Resim 5.16.'de gösterildiği gibi açılmaktadır. Kullanıcı tarafından seçilen modele ait arka planda algoritmalar çalışacak ve model kurulup çözülecektir. Problemin çözme işleminin ardından

etkinlik skorlarının gösterileceği Resim 5.17.'te gösterilen rapor ekranı kullanıcının bilgisine sunulacaktır.

Performans Analizi Karar Destek Sistemi

Dosya Düzen Görünüm Analiz Pencere Yardım

Veri Giriş Arayüzü Sonuç Arayüzü

Veri Giriş Arayüzü

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	K-2000	gird1	gird2	çık1	K-2001	gird11	gird21	çık11	K-2002	gird12	gird22	çık12	K-2003	gird13	gird23	çık13	K-2004	gird14	gird24	çık14	
2	KVB1	417	206	315	KVB1	324	94	496	KVB1	239	455	182	KVB1	97	126	467	KVB1	316	349	221	
3	KVB2	241	355	134	KVB2	404	349	164	KVB2	90	226	391	KVB2	40	28	443	KVB2	441	325	412	
4	KVB3	458	237	438	KVB3	227	484	194	KVB3	217	311	240	KVB3	143	366	93	KVB3	358	139	348	
5	KVB4	313	25	81	KVB4	207	203	150	KVB4	217	477	430	KVB4	346	332	260	KVB4	113	331	37	
6	KVB5	427	190	286	KVB5	356	416	152	KVB5	199	378	474	KVB5	466	452	428	KVB5	32	138	424	
7	KVB6	175	80	105	KVB6	440	212	61	KVB6	474	324	197	KVB6	393	34	250	KVB6	256	279	417	
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					
25																					
26																					
27																					

D:\Akademik Dosyalar\TEZ\Örnek Veriler\Window\Örnek2.xlsx

Resim 5.15. Panel veriler için veri giriş arayüzü (Ek 4, Çizelge 4.1)

Analiz Arayüzü

ANALİZ

Veri Zarflama Analizi Window Analiz Oran Analizi

Periyot: 5

Genişlik: 4

Karar Veme Birimi Sayısı: 6

Model Seçiniz: ▼

Tamam İptal

Resim 5.16. Window analizi arayüzü

Performans Analizi Karar Destek Sistemi

Dosya Düzen Görünüm Analiz Pencere Yardım

Veri Giriş Arayüzü Sonuç Arayüzü

Window Analiz

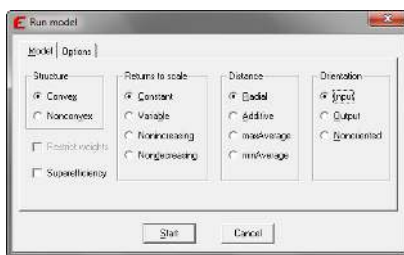
	A	B	C	D	E	F	G
1	K-2000	K-2001	K-2002	K-2003	K-2004	Ortalama	
2	KVB1	0.0966	0.3335	0.0688	0.4347		0.2334
3			0.3335	0.0645	0.421	0.0618	0.2202
4	Ortalama	0.0966	0.3335	0.06665	0.42785	0.0618	0.19728
5							
6	KVB2	0.0502	0.0367	0.3923	1		0.3698
7			0.0363	0.3571	1	0.0842	0.3694
8	Ortalama	0.0502	0.0365	0.3747	1	0.0842	0.30912
9							
10	KVB3	0.1168	0.0772	0.0999	0.0587		0.08815
11			0.0716	0.096	0.0533	0.1582	0.094775
12	Ortalama	0.1168	0.0744	0.09795	0.056	0.1582	0.10067
13							
14	KVB4	0.2048	0.0654	0.1789	0.0679		0.12925
15			0.0644	0.1654	0.0669	0.0264	0.080775
16	Ortalama	0.2048	0.0649	0.17215	0.0674	0.0264	0.10713
17							
18	KVB5	0.0951	0.0386	0.2151	0.0829		0.107925
19			0.0376	0.2019	0.0817	1	0.3303
20	Ortalama	0.0951	0.0381	0.2085	0.0823	1	0.2848
21							
22	KVB6	0.083	0.0182	0.0384	0.4647		0.151075
23			0.0182	0.0384	0.4647	0.144	0.166325
24	Ortalama	0.083	0.0182	0.0384	0.4647	0.144	0.14966
* 25							

D:\Akademik Dosyalar\TEZ\Örnek Veriler\WindowÖrnek2.xlsx

Resim 5.17. Window analiz için rapor arayüzü (Ek 4, Çizelge 4.2)

5.5. Karar Destek Sisteminin Diğer Programlar ile Karşılaştırılması

Geliştirilen KDS'nin doğruluğunu test etmek için sonuçlar ticari olmayan Efficiency Measurement System (EMS 1.3.0) paket programı ile kıyaslanması tercih edilmiştir. EMS'de ilgili model seçmek için Resim 5.18.'de gösterildiği gibi radiobutonlar kullanılmaktadır. Model seçiminin ardından Resim 5.19.'da gösterilen şekilde analiz sonucu kullanıcıya sunulmaktadır.



Resim 5.18. EMS model seçim ekranı

	DMU	Score	Cost {}/{}V	Client {}/{}V	Reve {}/{}V	Benchmarks	{S} Cost {}	{S} Client {}	{S} Reve {}
1	F1	74.29%	0.00	1.00	1.00	4 (1.51)	0.24	0.00	0.00
2	F2	99.07%	0.94	0.06	1.00	8 (0.02) 10 (1.87)	0.00	0.00	0.00
3	F3	95.87%	0.00	1.00	1.00	4 (1.62)	10.30	0.00	0.00
4	F4	100.00%	0.85	0.15	1.00	6			
5	F5	89.45%	0.93	0.07	1.00	8 (0.05) 10 (1.15)	0.00	0.00	0.00
6	F6	92.90%	0.91	0.09	1.00	8 (0.36) 10 (0.72)	0.00	0.00	0.00
7	F7	79.60%	0.00	1.00	1.00	4 (0.77)	1.68	0.00	0.00
8	F8	100.00%	1.00	0.00	1.00	5			
9	F9	76.48%	0.00	1.00	1.00	4 (0.63)	2.63	0.00	0.00
10	F10	100.00%	0.93	0.07	1.00	6			
11	F11	97.71%	0.91	0.09	1.00	4 (1.57) 10 (0.93)	0.00	0.00	0.00
12	F12	92.14%	0.93	0.07	1.00	8 (0.10) 10 (0.93)	0.00	0.00	0.00
13	F13	97.23%	0.91	0.09	1.00	8 (0.30) 10 (0.54)	0.00	0.00	0.00
14	F14	98.44%	0.00	1.00	1.00	4 (1.24)	0.67	0.00	0.00

Resim 5.19. EMS analiz sonucu

Geliştirilen KDS'nde ise Analiz arayüzünden Resim 5.20.'de gösterildiği gibi model seçilmektedir. KDS'ne ait analiz sonucu ise Resim 5.21.'de gösterilmektedir. Bunun yanında etkinlik değerlerine bağlı olarak grafikleri de gösterilmektedir.

Analiz Arayüzü

ANALİZ

Veri Zarflama Analizi | Window Analiz | Oran Analizi

Model Seçiniz

	Değişkenler	Kontrol Edilemeyen mi?
1	G-Cost	☐
2	G-Clients	☐
* 3	Ç-Revenue	☐

CCR Girdi Yönlü Model

Tamam İptal

Resim 5.20. KDS model seçim ekranı

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Karar Verme Birimleri :	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
2	Etkinlik Değerleri :	0.7429	0.9907	0.9587	1	0.8945	0.929	0.796	1	0.7648	1	0.9771	0.9214	0.9723	0.9844
3	Referanslar Olma Sıklığı :				6				5		6				
4	Referanslar :	F4 (1.51)	F8 (0.02)	F4 (1.62)		F8 (0.05)	F8 (0.36)	F4 (0.77)		F4 (0.63)		F4 (1.57)	F8 (0.1)	F8 (0.3)	F4 (1.24)
5			F10 (1.87)			F10 (1.15)	F10 (0.72)					F10 (0.93)	F10 (0.93)	F10 (0.54)	
6															
7															
8															
9															
10															
11															
12															
13															
14															
► 15															
16															
17															
18															
* 19															

Resim 5.21. KDS analiz sonucu (Ek 3, Çizelge 3.2)

Yukarıda gösterilen ufak örnekte de görüleceği gibi hem KDS hem de EMS paket programı aynı sonuçları vermektedir. Büyük bir örnek için iki paket programın sonuçlarını inceleyecek olursak:

Ornek2KDS [Uyumluluk Modu] - Microso...

Dosya Giri: Ekle Sayf Forr Veri Göz Görn Loai ABB Tear ?

Yapıştır Pano Yazı Tipi Hizalama Sayı Stiller Hücreler Düzenleme

F49 102.19

	A	B	C	D	E	F	G
1		GCost	Gstuff	Gclients	Çrevenue	Çsatisfaction	
2	F1	152.65	129.95	128.63	122.85	154.12	
3	F2	138.92	134.55	157.66	153.48	171.37	
4	F3	137.19	115.32	106.78	131.6	144.7	
5	F4	74.9	95.25	63.25	81.31	73.78	
6	F5	98.47	88.5	117.57	98.56	105.04	
7	F6	91.93	82.08	155.2	98.22	73.71	
8	F7	74.51	59.96	61.14	62.56	42.89	
9	F8	91.06	106	235.15	109.85	121.86	
10	F9	64.7	64.7	51.73	50.86	32.98	
11	F10	72.92	81.9	81.61	81.25	91.68	
12	F11	190.01	140	179.52	203.5	187.36	
13	F12	83.32	77.6	107.19	86.35	73.78	
14	F13	68.63	62.25	117.57	76.85	30.82	
15	F14	77.46	84.2	64.53	103.05	76.8	
16	F15	95.36	111.46	79.95	101.17	129.84	
17	F16	97.92	111.17	160.88	113.08	118.1	
18	F17	142.35	156.2	23.52	144.49	170.9	
19	F18	71.75	72.58	79.95	66.55	49.01	
20	F19	132.78	125.15	198.1	172.16	122.81	
21	F20	58.42	58.52	14.11	47.93	24.34	
22	F21	71.11	70.14	67.14	74.23	57.52	
23	F22	133.67	90.99	155.2	112.56	142.48	
24	F23	64.9	69.13	67.6	67.46	41.91	
25	F24	150.95	141.2	120.27	86.46	129.07	
26	F25	86.69	86.78	131.23	74.93	55.29	
27	F26	62.86	50.49	107.19	43.7	26.04	
28	F27	61.63	57.38	65.83	27.92	11.97	
29	F28	67.72	114.9	14.11	55.96	84.55	
30	F29	64.66	71.72	51.73	66.53	43.58	
31	F30	60.16	52.07	65.83	42.49	12.06	
32	F31	101.9	119.49	18.81	99.04	131.57	
33	F32	103.55	107.58	56.35	112.08	130.35	
34	F33	123.08	116.92	14.11	142.86	162.81	
35	F34	69.43	73.19	65.83	70.9	56.13	
36	F35	136.85	158.07	235.15	212.97	176.6	
37	F36	71.82	90.13	61.14	93.35	85.44	
38	F37	111.01	121.35	117.57	148.43	174.3	
39	F38	70.15	71.72	64.96	76.31	58.3	
40	F39	94.51	104.71	117.57	102.38	121.53	
41	F40	68.16	70.57	37.62	74.52	59.11	
42	F41	73.47	80.04	81.88	74.76	83.7	
43	F42	181.56	158.11	180.04	139.02	177.74	
44	F43	144.14	136.99	155.2	109.71	151.16	
45	F44	102.67	103.85	94.06	125.16	146.5	
46	F45	80.96	92.95	79.95	82.45	71.58	
47	F46	105.3	130.1	118.69	101.35	124.27	
48	F47	217.04	233.66	155.2	159.99	186.37	
49	F48	104.76	68.45	115.71	116.82	102.19	
50							

Sayfa1 Hazır %100

Resim 5.22. Örneğe ait verilerin ekran görüntüsü

	DMU	Score	Cost (I)\(V)	stuff (I)\(V)	client- (I)\(V)	reven- (O)\(V)	satisf-	Benchmarks	{S} Cost (I)	{S} stuff (I)	{S} client- (I)	{S} reven	{S} satisf-
1	F1	80.51%	0.11	0.84	0.05	0.00	1.00	22 (0.45) 33 (0.28) 37 (0.25)	0.00	0.00	0.00	5.64	0.00
2	F2	87.82%	0.15	0.85	0.00	0.14	0.86	22 (0.05) 37 (0.78) 48 (0.27)	0.00	0.00	7.27	0.00	0.00
3	F3	86.28%	0.10	0.86	0.04	0.13	0.87	22 (0.19) 33 (0.32) 37 (0.20) 48 (0.29)	0.00	0.00	0.00	0.00	0.00
4	F4	80.97%	0.84	0.00	0.16	1.00	0.00	33 (0.27) 35 (0.20)	0.00	13.85	0.00	0.00	5.55
5	F5	81.48%	0.15	0.85	0.00	0.15	0.85	22 (0.02) 37 (0.40) 48 (0.31)	0.00	0.00	9.04	0.00	0.00
6	F6	80.59%	0.41	0.59	0.00	1.00	0.00	35 (0.26) 48 (0.37)	0.00	0.00	21.54	0.00	9.70
7	F7	71.54%	0.42	0.47	0.11	1.00	0.00	33 (0.15) 35 (0.04) 48 (0.28)	0.00	0.00	0.00	0.00	17.15
8	F8	87.83%	1.00	0.00	0.00	0.53	0.47	35 (0.10) 37 (0.60)	0.00	4.85	13.07	0.00	0.00
9	F9	60.29%	0.38	0.53	0.10	1.00	0.00	33 (0.17) 35 (0.11) 48 (0.03)	0.00	0.00	0.00	0.00	16.94
10	F10	81.97%	0.84	0.00	0.16	0.81	0.19	33 (0.01) 35 (0.05) 37 (0.47)	0.00	1.51	0.00	0.00	0.00
11	F11	93.39%	0.43	0.44	0.13	1.00	0.00	33 (0.28) 35 (0.07) 48 (1.28)	0.00	0.00	0.00	0.00	0.56
12	F12	77.67%	0.37	0.48	0.15	1.00	0.00	33 (0.06) 35 (0.23) 48 (0.25)	0.00	0.00	0.00	0.00	1.86
13	F13	83.68%	0.40	0.60	0.00	1.00	0.00	35 (0.21) 48 (0.27)	0.00	0.00	17.13	0.00	34.35
14	F14	99.44%	0.85	0.00	0.15	1.00	0.00	33 (0.35) 35 (0.25)	0.00	3.48	0.00	0.00	23.97
15	F15	90.03%	0.85	0.00	0.15	0.00	1.00	33 (0.16) 37 (0.59)	0.00	9.37	0.00	10.09	0.00
16	F16	81.79%	1.00	0.00	0.00	0.55	0.45	35 (0.20) 37 (0.48)	0.00	1.68	28.73	0.00	0.00
17	F17	89.86%	0.97	0.00	0.03	0.00	1.00	33 (0.98) 37 (0.06)	0.00	17.87	0.00	5.19	0.00
18	F18	68.11%	0.36	0.51	0.13	1.00	0.00	33 (0.13) 35 (0.19) 48 (0.07)	0.00	0.00	0.00	0.00	12.14
19	F19	94.69%	0.39	0.61	0.00	1.00	0.00	35 (0.53) 48 (0.51)	0.00	0.00	4.23	0.00	22.65
20	F20	68.83%	0.95	0.00	0.05	1.00	0.00	33 (0.30) 35 (0.02)	0.00	1.43	0.00	0.00	28.75
21	F21	79.23%	0.37	0.51	0.11	1.00	0.00	33 (0.19) 35 (0.17) 48 (0.09)	0.00	0.00	0.00	0.00	13.02
22	F22	100.00%	0.09	0.91	0.00	0.00	1.00		8				
23	F23	74.94%	0.35	0.53	0.12	1.00	0.00	33 (0.17) 35 (0.20) 48 (0.00)	0.00	0.00	0.00	0.00	21.39
24	F24	63.10%	0.10	0.86	0.04	0.00	1.00	22 (0.20) 33 (0.24) 37 (0.36)	0.00	0.00	0.00	22.73	0.00
25	F25	60.94%	0.35	0.48	0.17	1.00	0.00	33 (0.01) 35 (0.26) 48 (0.15)	0.00	0.00	0.00	0.00	7.88
26	F26	55.76%	0.43	0.57	0.00	1.00	0.00	35 (0.08) 48 (0.23)	0.00	0.00	14.63	0.00	11.45
27	F27	34.86%	0.38	0.49	0.13	1.00	0.00	33 (0.05) 35 (0.06) 48 (0.07)	0.00	0.00	0.00	0.00	13.70
28	F28	92.67%	0.96	0.00	0.04	0.00	1.00	33 (0.46) 37 (0.06)	0.00	45.97	0.00	17.98	0.00
29	F29	77.38%	0.85	0.00	0.15	1.00	0.00	33 (0.23) 35 (0.16)	0.00	3.58	0.00	0.00	21.91
30	F30	56.11%	0.39	0.47	0.14	1.00	0.00	33 (0.05) 35 (0.08) 48 (0.16)	0.00	0.00	0.00	0.00	26.45
31	F31	96.28%	0.96	0.00	0.04	0.00	1.00	33 (0.74) 37 (0.07)	0.00	20.80	0.00	16.11	0.00
32	F32	87.70%	0.90	0.00	0.10	0.00	1.00	33 (0.40) 37 (0.37)	0.00	2.17	0.00	0.62	0.00
33	F33	100.00%	0.00	0.00	1.00	1.00	0.00		33				
34	F34	74.83%	0.36	0.53	0.11	1.00	0.00	33 (0.20) 35 (0.20) 48 (0.00)	0.00	0.00	0.00	0.00	11.63
35	F35	100.00%	1.00	0.00	0.00	1.00	0.00		27				
36	F36	96.83%	0.84	0.00	0.16	1.00	0.00	33 (0.31) 35 (0.23)	0.00	14.66	0.00	0.00	5.51
37	F37	100.00%	1.00	0.00	0.00	0.00	1.00		20				
38	F38	81.22%	0.37	0.52	0.11	1.00	0.00	33 (0.22) 35 (0.19) 48 (0.04)	0.00	0.00	0.00	0.00	14.75
39	F39	81.90%	1.00	0.00	0.00	0.00	1.00	37 (0.70)	0.00	1.14	14.31	1.11	0.00
40	F40	86.19%	0.89	0.00	0.11	1.00	0.00	33 (0.35) 35 (0.12)	0.00	1.73	0.00	0.00	18.08
41	F41	75.06%	0.37	0.55	0.08	0.76	0.24	33 (0.01) 35 (0.05) 37 (0.41) 48 (0.01)	0.00	0.00	0.00	0.00	0.00
42	F42	75.93%	0.11	0.84	0.06	0.00	1.00	22 (0.51) 33 (0.14) 37 (0.47)	0.00	0.00	0.00	8.45	0.00
43	F43	75.21%	0.10	0.85	0.06	0.00	1.00	22 (0.28) 33 (0.02) 37 (0.63)	0.00	0.00	0.00	16.66	0.00
44	F44	97.92%	0.08	0.88	0.04	0.13	0.87	22 (0.09) 33 (0.13) 37 (0.63) 48 (0.02)	0.00	0.00	0.00	0.00	0.00
45	F45	74.01%	0.82	0.00	0.18	1.00	0.00	33 (0.22) 35 (0.24)	0.00	5.18	0.00	0.00	6.63
46	F46	75.16%	1.00	0.00	0.00	0.00	1.00	37 (0.71)	0.00	11.27	5.39	4.48	0.00
47	F47	58.02%	0.87	0.00	0.13	0.00	1.00	33 (0.37) 37 (0.72)	0.00	4.48	0.00	0.29	0.00
48	F48	100.00%	0.00	1.00	0.00	1.00	0.00		21				

Resim 5.23. Örneğe ait EMS analiz sonucu

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
Karar Verme Birimleri :	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	
Etkinlik Değerleri :	0.8051	0.8782	0.8628	0.8097	0.8148	0.8059	0.7154	0.8783	0.6029	0.8197	0.9339	0.7767	0.8368	0.9944	0.9003	0.8179	0.8986	0.6811	
Referanslar Olma Şekli :																			
Referanslar :	F22 (0.45)	F22 (0.05)	F22 (0.19)	F33 (0.27)	F22 (0.02)	F35 (0.26)	F33 (0.15)	F35 (0.1)	F33 (0.17)	F33 (0.01)	F33 (0.28)	F33 (0.06)	F35 (0.21)	F33 (0.35)	F33 (0.16)	F35 (0.2)	F33 (0.98)	F33 (0.13)	
	F33 (0.28)	F37 (0.78)	F33 (0.32)	F35 (0.2)	F37 (0.4)	F48 (0.37)	F35 (0.04)	F37 (0.6)	F35 (0.11)	F35 (0.05)	F35 (0.07)	F35 (0.23)	F48 (0.27)	F35 (0.25)	F37 (0.59)	F37 (0.48)	F37 (0.06)	F35 (0.15)	
	F37 (0.25)	F48 (0.27)	F37 (0.2)		F48 (0.31)		F48 (0.28)		F48 (0.03)	F37 (0.47)	F48 (1.28)	F48 (0.25)						F48 (0.07)	
			F48 (0.29)																
T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34	F35	F36	F37	F38
0.9469	0.6883	0.7923	1	0.7494	0.631	0.6094	0.5576	0.3486	0.9267	0.7738	0.5611	0.9628	0.877	1	0.7483	1	0.9683	1	0.8122
		8												33		27		20	
F35 (0.53)	F33 (0.3)	F33 (0.19)		F33 (0.17)	F22 (0.2)	F33 (0.01)	F35 (0.08)	F33 (0.05)	F33 (0.46)	F33 (0.23)	F33 (0.05)	F33 (0.74)	F33 (0.4)		F33 (0.2)		F33 (0.31)		F33 (0.22)
F48 (0.51)	F35 (0.02)	F35 (0.17)		F35 (0.2)	F33 (0.24)	F35 (0.26)	F48 (0.23)	F35 (0.06)	F37 (0.06)	F35 (0.16)	F35 (0.08)	F37 (0.07)	F37 (0.37)		F35 (0.2)		F35 (0.23)		F35 (0.15)
		F48 (0.09)		F48 (0)	F37 (0.36)	F48 (0.15)		F48 (0.07)			F48 (0.16)				F48 (0)				F48 (0.04)
AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW										
F39	F40	F41	F42	F43	F44	F45	F46	F47	F48										
0.819	0.8619	0.7506	0.7593	0.7521	0.9792	0.7401	0.7516	0.5802	1										
									21										
F37 (0.7)	F33 (0.35)	F33 (0.01)	F22 (0.51)	F22 (0.28)	F22 (0.09)	F33 (0.22)	F37 (0.71)	F33 (0.37)											
	F35 (0.12)	F35 (0.05)	F33 (0.14)	F33 (0.02)	F33 (0.13)	F35 (0.24)		F37 (0.72)											
		F37 (0.41)	F37 (0.47)	F37 (0.63)	F37 (0.63)														
		F48 (0.01)		F48 (0.02)															

Resim 5.24. Örneğe ait KDS analiz sonucu (Ek 4)

Büyük örneğe ait CCR girdi yönlü modellerin her iki paket programda da sonuçları yukarda gösterilmektedir. Şimdi ise CCR çıktı yönlü modellerin sonuç çizelgelerini verelim.

Çizelge 5.9. Örneğe ait CCR çıktı yönlü modelin EMS analiz sonuçları

DMU	Score	Benchmarks
F1	124,21%	22 (0,56) 33 (0,35) 37 (0,32)
F2	113,87%	22 (0,06) 37 (0,89) 48 (0,31)
F3	115,90%	22 (0,23) 33 (0,37) 37 (0,23) 48 (0,34)
F4	123,50%	33 (0,33) 35 (0,25)
F5	122,73%	22 (0,03) 37 (0,49) 48 (0,38)
F6	124,08%	35 (0,32) 48 (0,46)
F7	139,77%	33 (0,21) 35 (0,05) 48 (0,40)
F8	113,86%	35 (0,11) 37 (0,68)
F9	165,87%	33 (0,28) 35 (0,18) 48 (0,05)
F10	122,00%	33 (0,01) 35 (0,06) 37 (0,57)
F11	107,08%	33 (0,30) 35 (0,07) 48 (1,37)
F12	128,75%	33 (0,08) 35 (0,30) 48 (0,32)
F13	119,50%	35 (0,25) 48 (0,32)
F14	100,57%	33 (0,35) 35 (0,25)
F15	111,07%	33 (0,18) 37 (0,66)
F16	122,26%	35 (0,24) 37 (0,58)
F17	111,28%	33 (1,09) 37 (0,07)
F18	146,83%	33 (0,18) 35 (0,28) 48 (0,11)
F19	105,61%	35 (0,56) 48 (0,54)
F20	145,29%	33 (0,44) 35 (0,03)
F21	126,22%	33 (0,24) 35 (0,22) 48 (0,11)
F22	100,00%	8
F23	133,44%	33 (0,22) 35 (0,27) 48 (0,00)
F24	158,48%	22 (0,31) 33 (0,38) 37 (0,57)
F25	164,08%	33 (0,01) 35 (0,43) 48 (0,25)
F26	179,35%	35 (0,14) 48 (0,42)
F27	286,83%	33 (0,14) 35 (0,18) 48 (0,20)
F28	107,91%	33 (0,50) 37 (0,06)
F29	129,24%	33 (0,30) 35 (0,20)
F30	178,22%	33 (0,10) 35 (0,13) 48 (0,28)
F31	103,87%	33 (0,77) 37 (0,07)
F32	114,02%	33 (0,46) 37 (0,42)
F33	100,00%	33

Çizelge 5.9. (Devam) Örneğe ait CCR çıktı yönlü modelin EMS analiz sonuçları

DMU	Score	Benchmarks
F34	133,63%	33 (0,27) 35 (0,26) 48 (0,00)
F35	100,00%	27
F36	103,28%	33 (0,32) 35 (0,24)
F37	100,00%	20
F38	123,12%	33 (0,27) 35 (0,24) 48 (0,05)
F39	122,10%	37 (0,85)
F40	116,03%	33 (0,40) 35 (0,14)
F41	133,23%	33 (0,02) 35 (0,07) 37 (0,55) 48 (0,01)
F42	131,70%	22 (0,67) 33 (0,18) 37 (0,62)
F43	132,96%	22 (0,37) 33 (0,02) 37 (0,83)
F44	102,12%	22 (0,09) 33 (0,14) 37 (0,64) 48 (0,02)
F45	135,11%	33 (0,30) 35 (0,32)
F46	133,04%	37 (0,95)
F47	172,36%	33 (0,64) 37 (1,24)
F48	100,00%	21

Çizelge 5.10. Örneğe ait CCR çıktı yönlü modelin KDS analiz sonuçları

Karar Verme Birimleri	Etkinlik Değerleri	Referanslar Olma Sıklığı	Referanslar			
F1	1,2421		F22 (0,56)	F33 (0,35)	F37 (0,32)	
F2	1,1387		F22 (0,06)	F37 (0,89)	F48 (0,31)	
F3	1,159		F22 (0,23)	F33 (0,37)	F37 (0,23)	F48 (0,34)
F4	1,235		F33 (0,33)	F35 (0,25)		
F5	1,2273		F22 (0,03)	F37 (0,49)	F48 (0,38)	
F6	1,2408		F35 (0,32)	F48 (0,46)		
F7	1,3977		F33 (0,21)	F35 (0,05)	F48 (0,4)	
F8	1,1386		F35 (0,11)	F37 (0,68)		
F9	1,6587		F33 (0,28)	F35 (0,18)	F48 (0,05)	
F10	1,22		F33 (0,01)	F35 (0,06)	F37 (0,57)	
F11	1,0708		F33 (0,3)	F35 (0,07)	F48 (1,37)	
F12	1,2875		F33 (0,08)	F35 (0,3)	F48 (0,32)	
F13	1,195		F35 (0,25)	F48 (0,32)		
F14	1,0057		F33 (0,35)	F35 (0,25)		
F15	1,1107		F33 (0,18)	F37 (0,66)		
F16	1,2226		F35 (0,24)	F37 (0,58)		

Çizelge 5.10. (Devam) Örneğe ait CCR çıktı yönlü modelin KDS analiz sonuçları

Karar Verme Birimleri	Etkinlik Değerleri	Referanslar Olma Sıklığı	Referanslar			
F17	1,1128		F33 (1,09)	F37 (0,07)		
F18	1,4683		F33 (0,18)	F35 (0,28)	F48 (0,11)	
F19	1,0561		F35 (0,56)	F48 (0,54)		
F20	1,4529		F33 (0,44)	F35 (0,03)		
F21	1,2622		F33 (0,24)	F35 (0,22)	F48 (0,11)	
F22	1	8				
F23	1,3344		F33 (0,22)	F35 (0,27)	F48 (0)	
F24	1,5848		F22 (0,31)	F33 (0,38)	F37 (0,57)	
F25	1,6408		F33 (0,01)	F35 (0,43)	F48 (0,25)	
F26	1,7935		F35 (0,14)	F48 (0,42)		
F27	2,8683		F33 (0,14)	F35 (0,18)	F48 (0,20)	
F28	1,0791		F33 (0,50)	F37 (0,06)		
F29	1,2924		F33 (0,30)	F35 (0,20)		
F30	1,7822		F33 (0,10)	F35 (0,13)	F48 (0,28)	
F31	1,0387		F33 (0,77)	F37 (0,07)		
F32	1,1402		F33 (0,46)	F37 (0,42)		
F33	1	33				
F34	1,3363		F33 (0,27)	F35 (0,26)	F48 (0)	
F35	1	27				
F36	1,0328		F33 (0,32)	F35 (0,24)		
F37	1	20				
F38	1,2312		F33 (0,27)	F35 (0,24)	F48 (0,05)	
F39	1,221		F37 (0,85)			
F40	1,1603		F33 (0,40)	F35 (0,14)		
F41	1,3323		F33 (0,02)	F35 (0,07)	F37 (0,55)	F48 (0,01)
F42	1,317		F22 (0,67)	F33 (0,18)	F37 (0,62)	
F43	1,3296		F22 (0,37)	F33 (0,02)	F37 (0,83)	
F44	1,0212		F22 (0,09)	F33 (0,14)	F37 (0,64)	F48 (0,02)
F45	1,3511		F33 (0,30)	F35 (0,32)		
F46	1,3304		F37 (0,95)			
F47	1,7236		F33 (0,64)	F37 (1,24)		
F48	1	21				

Çizelge 5.9. ve Çizelge 5.10.'dan görüldüğü üzere KVB'lerinin etkinlik skorları, referans kümeler ve referans olma sıklığı hem KDS hem de EMS paket programında aynı olarak ölçülmektedir.

Ayrıca EMS'de manuel veri giriş arayüzü olmamasına rağmen KDS'de mevcuttur. EMS'de yalnızca MS Office 97-2003 Excel (*.xls) sürümünü desteklerken KDS'de ise MS Office 97-2003 Excel sürümünün yanında 2007 sürümünü de desteklemektedir. EMS'de not defterinden veri almak mümkün değilken KDS'de mümkündür. Bu iki programın yapılan analizler sonrasında raporları sunduğu platform açısından da farklılıklar bulunmaktadır. Sonuçları, EMS yalnızca Excel'e aktarma yaparken KDS ise Word, PDF, Excel ve Not Defteri gibi platformlara aktarma yapabilmektedir.

6. SONUÇ VE ÖNERİLER

Etkinlik ve verimlilik gibi kavramlar, kaynakların sınırlı olduğu dünyada her zaman önemli olmuş ve olmaya devam edecektir. Kaynakların verimli kullanılması zorunluluğu mal ve hizmet üretiminde performans ölçümünü gerekli kılmaktadır. Organizasyonlar için yüksek düzeyde bir performans yakalamak başarının anahtar faktörüdür. Bu nedenle sahip olunan performansı iyileştirmek ve organizasyonun neden etkin işlemediğini anlamak günümüz için önemli bir amaçtır. Performans analizi yöntemlerinden biri olan VZA, KVB'lerine ait çeşitli girdi ve çıktı değişkenlerinin gözlem değerleri kullanılarak, göreceli etkinliğin ölçüldüğü parametrik olmayan bir yöntemdir.

Bu çalışmada tamamen Türkçe olan, bazı performans analizi bilgisayar paket programlarının eksiklerini de tamamlayan ve Microsoft Visual C# programlama dili ile bir performans analizi paket programı geliştirilmiştir. KDS'nin çalışabilmesi için .NET Framework 4 gereklidir. Örneğin performans analizi yapan DEAP Windows arayüzü olmayan bir paket programıdır. Geliştirilen KDS'nde DEAP paket programının bu eksikliği dikkate alınarak Windows işletim sisteminde çalışan bir paket programı geliştirilmiştir. Veri girişlerde yaşanan bazı sıkıntıları giderebilmek için Excel, Not Defteri, Manuel giriş gibi birçok seçenek kullanımına sunulmuştur. Bu sayede kullanıcı birçok ortamdan veri alabilecek ve rahatlıkla analizlerini yapabilecektir. Aynı şekilde yapılan analizlerin sonuçlarını da PDF, Excel, Word ve Not Defteri gibi ortamlara kolaylıkla aktarabilmesi diğer paket programlarda olmayan bir başka özelliktir. Yapılan bazı çalışmalarda örneğin girdi değişken olan mağazanın yaşının analiz sonucunda fazlalık verdiği varsayılabilir. Mağaza yaşını azaltmak mümkün değildir. Bu örnekte olduğu gibi kontrol edilemeyen değişkenlerin var olması durumunda o değişkene ait girdi fazlalıkları veya çıktı azlıkları modelden çıkarılmalıdır. Geliştirilen karar destek sistemi bu durumu dikkate alıp kullanıcının bilgisine başvurmuştur. Eğer değişken kontrol edilemeyen değişken ise karar destek sistemi o değişkeni modelden çıkarıp çözüme o şekilde gitmektedir. Karar destek sisteminin dikkate değer bir diğer özelliği ise kontrol edilemeyen değişkenleri dikkate almasıdır. Belli süreçler için yapılan window analizi de karar destek

sisteminde mevcuttur. Window analizi yapılan birçok çalışmada dönemler için ortalama değer diğer yollarla hesaplanırken karar destek sisteminde analiz esnasında hesaplanıp kullanıcının bilgisine sunulmaktadır.

Karar destek sisteminde şu an mevcut olmayan Malmquist index, toplamsal model gibi daha başka performans analiz yöntemleri eklenmesi düşünülmektedir. Geliştirilen bu karar destek sisteminin yerli üretim olması sebebiyle ülkemizde özellikle sağlık, eğitim, bankacılık, sanayi, ulaşım ve tarım sektörlerinde performans konusunda farkındalığın artması ve Türkiye ekonomisine ülke çapında katkıda bulunması düşünülmektedir.

KAYNAKLAR

Adler, N., Friedman, L., Sinuany-Stern Z., “Review of ranking methods in the data envelopment analysis context”, *European Journal of Operational Research*, 140: 249-265 (2002).

Alp, İ., Sözen, A., “Window Analizi ile Avrupa Ülkeleri ve Türkiye'nin Enerji ve Çevre Performans Değerlendirmesi”, *9th International Conference On Knowledge, Economy & Management Proceedings*, Saraybosna-Bosna Hersek, 1185-1887 (2011).

Akal, Z., “İşletmelerde Performans Ölçüm ve Denetimi: Çok Yönlü Performans Göstergeleri”, *MPM Yayınları*, Ankara, 473:463 (2005).

Akyol, M., “Veri Zarflama Analizi Ve Yükseköğretimde Bir Uygulama”, Yüksek Lisans Tezi, *Gazi Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara, 5-80 (2009).

Alan, M. A., Yeşilyurt, C., “Doğrusal programlama problemlerinin Excel ile Çözümü”, *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 5(1):151-162 (2004).

Aslankaraoğlu, N., “Veri zarflama analizi ve temel bileşenler analizi ile Avrupa birliği ülkelerinin sıralanması”, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 140-144 (2006).

Aydemir, Z.C., “Bölgesel rekabet edebilirlik kapsamında illerin kaynak kullanım görece verimlilikleri: Veri zarflama analizi uygulaması”, Uzmanlık Tezi, *Devlet Planlama Teşkilatı*, Ankara, 1-36 (2002).

Bal, H., “Optimizasyon Teknikleri”, *Gazi Üniversitesi*, Ankara, 36-46 (1995).

Banker, R. D., Charnes, A., Cooper W. W., “Some Models For Estimating Technical And Scale Inefficiencies In Data Envelopment Analysis”, *Management Science*, 30(9): 1078-1092 (1984).

Banker, R. D., “Estimation of returns to scale using data envelopment analysis”, *European Journal of Operational Research*, 174(3): 62 - 84 (1984).

Barr, R. S., “DEA software tools and technology”, *Kluwer Academic Publishers*, Boston, 539-566 (2004).

Bektaş, B., “Türkiye’de Faaliyet Gösteren Bankaların Farklı Yöntemlerle Sınıflandırılması ve Etkinliklerinin Belirlenmesi Üzerine Bir Araştırma”, Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 35-36 (2007).

Berger, A. H., Humprey, D. B., “Efficiency of financial institutions: International survey and directions for future research”, *European Journal of Operational Research*, 98: 175-212 (1997).

Besen, F. B., “Performans Yönetim Sistemi ve Veri Zarflama Analizinin Sağlık Sektöründe Uygulaması”, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 27-29 (1994).

Boussofiane, A., Dyson, R.G., Thanassoulis, E., “Applied Data Envelopment Analysis”, *European Journal of Operational Research*, 52(1): 1-15 (1991).

Bowlin, W.F., “Measuring Performance: An Introduction to Data Envelopment Analysis (DEA)”, *The Journal of Cost Analysis*, 3-27 (1998).

Bozdağ, A. E., “İMKB’ye Kote Olan Bankaların Etkinlik Analizleri”, Yüksek Lisans Tezi, *Gazi Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara 6-23 (2010).

Budak, H., “Veri Zarflama Analizi ve Türk Bankacılık Sektöründe Uygulaması” *Marmara Üniversitesi Fen Bilimleri Dergisi*, 23(3): 95-110 (2011).

Canbek, Z. F., “Veri zarflama analizi ile İstanbul’da bulunan özel hastanelerin etkinliklerinin incelenmesi”, Yüksek Lisans Tezi, *Anadolu Üniversitesi Sosyal Bilimler Enstitüsü*, Eskişehir (2007).

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

Charnes, A., Clark, C.T., Cooper, W.W., Golany, B., “A developmental study of data envelopment analysis in measuring the efficiency of maintenance units in the US air forces”, *Annals Of Operation Research*, 2:95-112 (1985).

Cingi, S., Tarım, A., “Türk Banka Sisteminde Performans Ölçümü Dea-Malmquist TFFV Endeksi Uygulaması”, *Türkiye Bankalar Birliği Araştırma Tebliğleri Serisi*, İstanbul, 1: 4-15 (2000).

Colbert, A., Levary, R.R., Shaner, M.C., “Determining the Relative Efficiency of MBA Programs Using DEA”, *European Journal of Operational Research*, 125: 656-669 (2000).

Cooper, W.W., Seiford, L.M., Tone, K., “Data Envelopment Analysis: A Comprehensive Text with Models, Application References and DEA-Solver Software”, *Kluwer Academic Publisher*, Boston USA, 100-175 (2000).

Cooper, W.W., Seiford, L.M., Zhu, J., “Handbook on data envelopment analysis”, *International Series in Operations Research and Management Science*, 1-36 (2004).

Çağlar, A., “Veri Zarflama Analizi ile Belediyelerin Etkinlik Ölçümü”, Doktora Tezi, *Hacettepe Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 15-170 (2003).

Deniz, N., “Türkiye’deki İllerin Kaynak Kullanımlarına Göre Göreli Etkinliklerinin Klasik ve Bulanık Veri Zarflama Analizi Yöntemleri ile Belirlenmesi”, Yüksek Lisans Tezi, *Anadolu Üniversitesi Fen Bilimleri Enstitüsü*, Eskişehir, 16-69 (2009).

Doyle, J., Green, R., “Data Envelopment Analysis and multiple criteria decision making”, *OMEGA*, 21(6): 713–715 (1993).

Ertuğrul, E., “Bireysel Emeklilik Sektörü İçin Etkinlik Analizi: Türkiye Örneği (2004–2008)”, Yüksek Lisans Tezi, *Gazi Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara 33-58 (2010).

Esin, A., “Yöneylem Araştırmasında Yararlanılan Karar Yöntemleri” , *Gazi Üniversitesi Fen Fakültesi*, Ankara,5:10-95 (1984)

Gökgöz, F., “Veri Zarflama Analizi ve Finans Alanına Uygulanması”, *Ankara Üniversitesi Siyasal Bilgiler Fakültesi Dergisi*, 597-598 (2009).

Güneş, T., “Bulanık veri zarflama analizi”, Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü*, Ankara 15-140 (2006).

Hacısalıhoğlu, H., “Genel Matematik”, *Anadolu Üniversitesi Yayınları*, 242, Eskişehir, (1992).

Hollingsworth, B., Smith, P., “Use of Ratios in Data Envelopment Analysis”, *Applied Economics Letters*, 10(11): 733 (2003).

İnan, E. A., “Banka Etkinliğinin Ölçülmesi ve Düşük Enflasyon Sürecinde Bankacılıkta Etkinlik”, *Bankacılık Dergisi*, 34:85 (2000).

İslamoğlu, A., “ Türkiye Bankacılık Sektöründe 2005 - 2007 Yılları Arasında Bankaların Veri Zarflama Analizi ile Performans Değerlendirmesi”, Yüksek Lisans Tezi, *Gebze Yüksek Teknoloji Enstitüsü Sosyal Bilimler Enstitüsü*, Gebze, 7-10 (2010).

Karaca, C., “Veri Zarflama Analizi ile Antalya Bölgesindeki Ziraat Bankası Şubelerinin Performans Değerlendirmesi” Yüksek Lisans Tezi, *Gazi Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 3-5 (2010).

Kılıçkaplan, S., Karpat, G., “Türkiye Hayat Sigortası Sektöründe Etkinliğin İncelenmesi”, *Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 19(1): 2 (2004).

Kıllı, M., “Toplam Etkinlik ve Veri Zarflama Analizi Üzerine Karşılaştırmalı Yaklaşımlar ve Bir Uygulama”, Yüksek Lisans Tezi, *Gazi Üniversitesi Sosyal Bilimler Enstitüsü*, Ankara, 4-5 (2004).

Kulaksızoğlu, E. Ş., “Veri Zarflama Analizi ve Çapraz Etkinlik” Yüksek Lisans Tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara 1-39 (2010).

Kutlar, A., Gülcü, A., Karagöz, Y., “Cumhuriyet Üniversitesi Fakültelerinin Performans Değerlendirmesi”, **Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi**, 5(2): (2004).

Küçükşimşek, M., “1999 – 2003 Döneminde Türkiye’deki Ortaöğretim Kurumlarının Öss’deki Etkinliklerinin DEA-Malmquist TFP Endeksi ile İncelenmesi”, Yüksek lisans tezi, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 1-33 (2004).

Lorcu, F., “Veri Zarflama Analizi ile Türkiye ve Avrupa Birliği Ülkelerinin Sağlık Alanındaki Etkinliklerinin Değerlendirilmesi”, Doktora Tezi, **İstanbul Üniversitesi Sosyal Bilimler Enstitüsü**, İstanbul, 9 (2008).

Miller, S. M., Clauretie T. M., Springer T. M., “Economies of Scale and Cost Efficiencies: A Panel Data Stochastic Frontier Analysis of Real Estate Investment Trusts”, **University of Connecticut Department of Economics**, 1-20 (2005).

Oruç, K. O., “Veri Zarflama Analizi ile Bulanık Ortamda Etkinlik Ölçümleri ve Üniversitelerde Bir Uygulama”, Doktora Tezi, **Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü**, Isparta, 24-28 (2008).

Öner, A., “Veri Zarflama Analizi ve Finans Sektöründe Bir Uygulama”, Yüksek Lisans Tezi, **Marmara Üniversitesi Sosyal Bilimler Enstitüsü**, İstanbul, 14-29 (2008).

Öner, N., “Sağlık Bakanlığına Bağlı Ağız Ve Diş Sağlığı Kurumlarının Veri Zarflama Analizi Yöntemi ile Performansının Değerlendirilmesi”, Yüksek Lisans Tezi, **Gazi Üniversitesi Sosyal Bilimler Enstitüsü**, Ankara, 49-63 (2010).

Özcan, G., “Veri Zarflama Analizi ve Bankacılık Sektöründe Bir Uygulama”, Yüksek Lisans Tezi, **Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü**, Kütahya, 2-10 (2007).

Öztürk, A., “Yöneylem Araştırması”, **Ekin Kitabevi Yayınları**, Bursa, 7:77 (2001).

Ramanathan, R., “An introduction to data envelopment analysis- a tool for performance measurement”, **Sage Publications**, New Delhi India, 8-65 (2003).

Sarı, T., “Üretimde Maliyet İyileştirme Takımlarının Verimlilik Performansının Veri Zarflama Analizi ile Ölçülmesi ve Bir Uygulama”, Yüksek Lisans Tezi, **Gebze Yüksek Teknoloji Enstitüsü Sosyal Bilimler Enstitüsü**, Gebze, 15-73 (2011).

Sarıca, S., “Üniversitelerin performansa göre yönetimi için veri zarflama Analizi tabanlı bir karar destek sisteminin tasarımı ve geliştirilmesi”, Yüksek Lisans Tezi, **Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü**, Eskişehir, 1-152 (2007).

Sarıkaya, M., “İllerin Sağlık Alanındaki Etkinliklerinin Değerlendirilmesi (Veri Zarflama Analizine Dayalı Bir Uygulama)”, Yüksek Lisans Tezi, **Gazi Üniversitesi Sosyal Bilimler Enstitüsü**, Ankara, 5-80 (2010).

Sexton, T.R., Silkman, R.H., Hogan, A.J., “Data envelopment analysis: Critique and extension Measuring Efficiency: An Assessment of Data Envelopment Analysis”, **Jossey-Bass**, San Francisco, 73-105 (1986).

Tarkoçin, C., Gençer, M., “Farklı Girdi ve Çıktı Yaklaşımlarının Veri Zarflama Analizi Etkinlik Sonuçlarına Etkisi ve Türk Ticari Bankaları Uygulaması”, **Bankacılar Dergisi**, 72: 19-32 (2010).

Tetik, S., “İşletme Performansını Belirlemede Veri Zarflama Analizi”, **Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Yönetim ve Ekonomi Dergisi**, 10(2): 221-229 (2003).

İnternet: Türkiye İstatistik Kurumu “Gayri Safi Yurtiçi Hasıla IV. Dönem: Ekim, Kasım, Aralık / 2011” <http://www.tuik.gov.tr/PreHaberBultenleri.do?id=10785> (2011).

Yapıcı, M., “Eğitimde Verimlilik”, **Verimlilik Dergisi Milli Prodüktivite Merkezi Yayını**, 1-20 (2002).

Yolalan, R., “İşletmeler arası Görelî Etkinlik Ölçümü”, **Milli Prodüktivite Merkezi Yayınları**, Ankara, 483: 5-7, 27-29 (1993).

Zerey, G., “Veri Zarflama Analizi Yardımıyla Etkinlik Ölçümü ve Bir Uygulama”, Yüksek Lisans Tezi, **Marmara Üniversitesi Sosyal Bilimler Enstitüsü**, İstanbul 12-54 (2010).

EKLER

EK-1 Charnes ve arkadaşlarının çalışmasına (1985) ait window analizi

Wing	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7
Wing-A	97,89	97,31 97,36	98,14 97,53 96,21	97,04 95,92 95,79	94,54 94,63 94,33	97,64 97,24	97,24
Wing-B	93,90	95,67 96,72	96,14 96,42 95,75	94,63 94,14 94,54	93,26 93,46 93,02	96,02 96,02	94,49
Wing-C	93,77	91,53 91,77	95,26 95,55 93,21	94,29 95,04 93,20	94,83 93,09 93,59	92,21 92,32	92,83
Wing-D	99,72	96,15 97,91	95,06 95,70 94,79	100,0 100,0 99,71	94,51 94,39 94,95	94,76 94,67	89,37
Wing-E	100,0	100,0 100,0	100,0 100,0 98,97	100,0 99,05 99,37	100,0 100,0 100,0	100,0 100,0	100,0
Wing-F	97,42	93,48 93,60	96,07 96,24 94,46	93,56 91,75 91,73	92,49 92,32 92,68	92,35 91,98	99,64
Wing-G	90,98	92,80 93,67	95,96 96,80 93,34	99,52 94,48 91,94	91,73 89,79 89,35	95,58 95,14	96,38
Wing-H	100,0	100,0 100,0	100,0 100,0 100,0	100,0 100,0 100,0	100,0 100,0 100,0	100,0 100,0	100,0
Wing-I	99,11	95,94 96,04	99,76 100,0 98,16	100,0 98,99 98,97	94,59 94,62 94,68	99,16 98,92	97,28
Wing-J	92,85	90,90 91,50	91,62 92,12 90,26	94,75 93,39 92,92	93,83 93,84 94,52	95,33 96,07	94,43
Wing-K	86,25	84,42 84,98	84,03 84,47 83,37	93,74 82,54 82,39	80,26 80,14 80,96	79,58 78,66	79,75
...							
Wing-N	100,0	100,0 100,0	98,63 100,0 99,45	100,0 100,0 100,0	100,0 100,0 100,0	100,0 100,0	100,0

EK-2 Alp ve arkadaşlarının çalışmasına (2011) ait window analizi

	1998	1999	2000	2001	2002	2003	2004	2005	Ort,	Std,S,	Aralık
Bulgaristan	100,0	100,0	100,0						100,0	0,0	0,0
		100,0	100,0	100,0							
			100,0	100,0	100,0						
				100,0	100,0	100,0					
					100,0	100,0	100,0				
						100,0	100,0	100,0			
Çek Cum,	100,0	100,0	100,0						99,1	1,6	4,8
		100,0		100,0							
			100,0	99,9	100,0						
				100,0	100,0	96,0					
					100,0	96,6	95,2				
						98,9	97,5	99,8			
İsviçre	65,2	63,6	70,4						70,5	7,2	17,6
		63,7	51,5	67,1							
			72,8	68,6	68,6						
				70,4	70,3	68,9					
					81,2	78,2	74,7				
						79,7	75,7	78,6			
Türkiye	100,0	100,0	100,0						100,0	0,0	0,0
		100,0	100,0	100,0							
			100,0	100,0	100,0						
				100,0	100,0	100,0					
					100,0	100,0	100,0				
						100,0	100,0	100,0			
Yıllara Göre Genel Ortalama	92,4	91,9	90,6	90,6	90,3	90,5	90,0	87,2	G.Ort. 90,4	G.Std.Sp. 14,6	G.Aralık 63,6
Bulgaristan	100,0	100,0	100,0						100,0	0,0	0,0
		100,0	100,0	100,0							
			100,0	100,0	100,0						
				100,0	100,0	100,0					
					100,0	100,0	100,0				
						100,0	100,0	100,0			
Fransa	71,4	68,2	65,8						63,4	4,0	12,7
		70,0	66,9	61,9							
			66,0	61,3	59,7						
				65,1	62,8	59,7					
					63,5	60,5	58,3				
						61,9	59,9	57,6			
Norveç	87,9	87,5	91,6						88,1	10,7	32,7
		95,3	100,0	99,9							
			95,2	94,6	92,7						
				95,6	93,9	73,1					
					93,9	73,1	67,3				
						94,9	82,0	67,6			
Türkiye	100,0	100,0	100,0						99,7	0,6	1,5
		100,0	98,5	100,0							
			100,0	100,0	98,5						
				100,0	98,6	99,2					
					100,0	100,0	100,0				
						100,0	100,0	100,0			
Yıllara Göre Genel Ortalama	84,5	83,6	83,3	82,9	82,8	82,3	81,6	80,9	G. Ort. 82,8	G. Std. Sp. 18,7	G.Aralık 63,6

EK-3 Örneğe ait veriler ve sonuçlar

Çizelge 3.1. Örneğe ait veriler

KVB	G-Cost	G-Clients	Ç-Revenue
F1	152,65	128,63	122,85
F2	138,92	157,66	153,48
F3	137,19	106,78	131,60
F4	74,90	63,25	81,31
F5	98,47	117,57	98,56
F6	91,93	155,20	98,22
F7	74,51	61,14	62,56
F8	91,06	235,15	109,85
F9	64,70	51,73	50,86
F10	72,92	81,61	81,25
F11	190,01	179,52	203,50
F12	83,32	107,19	86,35
F13	68,63	117,57	76,85
F14	95,36	79,95	101,17

Çizelge 3.2. Örneğe ait sonuçlar

Karar Verme Birimleri :	F1	F2	F3	F4	F5	F6	F7
Etkinlik Değerleri :	0,7429	0,9907	0,9587	1	0,8945	0,929	0,796
Referanslar Olma Sıklığı :				6			
Referanslar :	F4 (1,51)	F8 (0,02)	F4 (1,62)		F8 (0,05)	F8 (0,36)	F4 (0,77)
		F10 (1,87)			F10 (1,15)	F10 (0,72)	
Karar Verme Birimleri :	F8	F9	F10	F11	F12	F13	F14
Etkinlik Değerleri :	1	0,7648	1	1	0,9214	0,9723	0,9844
Referanslar Olma Sıklığı :	5		6				
Referanslar :		F4 (0,63)		F4 (1,57)	F8 (0,1)	F8 (0,3)	F4 (1,24)
				F10 (0,93)	F10 (0,93)	F10 (0,54)	

EK-4 Window analizi örneğine ait veriler ve sonuçları

Çizelge 4.1. Window analizi örneğine ait veriler

K-2000	girdi1	girdi2	çıktı1	K-2001	girdi1	girdi2	çıktı1	K-2002	girdi1	girdi2	çıktı1
KVB1	417	206	315	KVB1	324	94	496	KVB1	239	455	182
KVB2	241	355	134	KVB2	404	349	164	KVB2	90	226	391
KVB3	458	237	438	KVB3	227	484	194	KVB3	217	311	240
KVB4	313	25	81	KVB4	207	203	150	KVB4	217	477	430
KVB5	427	190	286	KVB5	356	416	152	KVB5	199	378	474
KVB6	175	80	105	KVB6	440	212	61	KVB6	474	324	197
K-2003	girdi1	girdi2	çıktı1	K-2004	girdi1	girdi2	çıktı1				
KVB1	97	126	467	KVB1	316	349	221				
KVB2	40	28	443	KVB2	441	325	412				
KVB3	143	366	93	KVB3	358	139	348				
KVB4	346	332	260	KVB4	113	331	37				
KVB5	466	452	428	KVB5	32	138	424				
KVB6	393	34	250	KVB6	256	279	417				

Çizelge 4.1. Window analizi örneğine ait sonuçlar

	K-2000	K-2001	K-2002	K-2003	K-2004	Ortalama
KVB1	0,0966	0,3335	0,0688	0,4347		0,2334
		0,3335	0,0645	0,421	0,0618	0,2202
Ortalama	0,0966	0,3335	0,06665	0,42785	0,0618	0,19728
KVB2	0,0502	0,0367	0,3923	1		0,3698
		0,0363	0,3571	1	0,0842	0,3694
Ortalama	0,0502	0,0365	0,3747	1	0,0842	0,30912
KVB3	0,1168	0,0772	0,0999	0,0587		0,08815
		0,0716	0,096	0,0533	0,1582	0,094775
Ortalama	0,1168	0,0744	0,09795	0,056	0,1582	0,10067
KVB4	0,2048	0,0654	0,1789	0,0679		0,12925
		0,0644	0,1654	0,0669	0,0264	0,080775
Ortalama	0,2048	0,0649	0,17215	0,0674	0,0264	0,10713
KVB5	0,0951	0,0386	0,2151	0,0829		0,107925
		0,0376	0,2019	0,0817	1	0,3303
Ortalama	0,0951	0,0381	0,2085	0,0823	1	0,2848
KVB6	0,083	0,0182	0,0384	0,4647		0,151075
		0,0182	0,0384	0,4647	0,144	0,166325
Ortalama	0,083	0,0182	0,0384	0,4647	0,144	0,14966

EK-5 İkinci örneğe ait sonuçlar

Karar Verme Birimleri :	F1	F2	F3	F4	F5	F6	F7
Etkinlik Değerleri :	0,8051	0,8782	0,8628	0,8097	0,8148	0,8059	0,7154
Referanslar Olma Sıklığı :							
Referanslar :	F22 (0,45)	F22 (0,05)	F22 (0,19)	F33 (0,27)	F22 (0,02)	F35 (0,26)	F33 (0,15)
	F33 (0,28)	F37 (0,78)	F33 (0,32)	F35 (0,2)	F37 (0,4)	F48 (0,37)	F35 (0,04)
	F37 (0,25)	F48 (0,27)	F37 (0,2)		F48 (0,31)		F48 (0,28)
			F48 (0,29)				
Karar Verme Birimleri :	F8	F9	F10	F11	F12	F13	F14
Etkinlik Değerleri :	0,8783	0,6029	0,8197	0,9339	0,7767	0,8368	0,9944
Referanslar Olma Sıklığı :							
Referanslar :	F35 (0,1)	F33 (0,17)	F33 (0,01)	F33 (0,28)	F33 (0,06)	F35 (0,21)	F33 (0,35)
	F37 (0,6)	F35 (0,11)	F35 (0,05)	F35 (0,07)	F35 (0,23)	F48 (0,27)	F35 (0,25)
		F48 (0,03)	F37 (0,47)	F48 (1,28)	F48 (0,25)		
Karar Verme Birimleri :	F15	F16	F17	F18	F19	F20	F21
Etkinlik Değerleri :	0,9003	0,8179	0,8986	0,6811	0,9469	0,6883	0,7923
Referanslar Olma Sıklığı :							
Referanslar :	F33 (0,16)	F35 (0,2)	F33 (0,98)	F33 (0,13)	F35 (0,53)	F33 (0,3)	F33 (0,19)
	F37 (0,59)	F37 (0,48)	F37 (0,06)	F35 (0,19)	F48 (0,51)	F35 (0,02)	F35 (0,17)
				F48 (0,07)			F48 (0,09)
Karar Verme Birimleri :	F22	F23	F24	F25	F26	F27	F28
Etkinlik Değerleri :	1	0,7494	0,631	0,6094	0,5576	0,3486	0,9267
Referanslar Olma Sıklığı :	8						
Referanslar :		F33 (0,17)	F22 (0,2)	F33 (0,01)	F35 (0,08)	F33 (0,05)	F33 (0,46)
		F35 (0,2)	F33 (0,24)	F35 (0,26)	F48 (0,23)	F35 (0,06)	F37 (0,06)
		F48 (0)	F37 (0,36)	F48 (0,15)		F48 (0,07)	
Karar Verme Birimleri :	F29	F30	F31	F32	F33	F34	F35
Etkinlik Değerleri :	0,7738	0,5611	0,9628	0,877	1	0,7483	1
Referanslar Olma Sıklığı :					33		27
Referanslar :	F33 (0,23)	F33 (0,05)	F33 (0,74)	F33 (0,4)		F33 (0,2)	
	F35 (0,16)	F35 (0,08)	F37 (0,07)	F37 (0,37)		F35 (0,2)	
		F48 (0,16)				F48 (0)	

EK-5 (Devam) İkinci örneğe ait sonuçlar

Karar Verme Birimleri :	F36	F37	F38	F39	F40	F41	F42
Etkinlik Değerleri :	0,9683	1	0,8122	0,819	0,8619	0,7506	0,7593
Referanslar Olma Sıklığı :		20					
Referanslar :	F33 (0,31)		F33 (0,22)	F37 (0,7)	F33 (0,35)	F33 (0,01)	F22 (0,51)
	F35 (0,23)		F35 (0,19)		F35 (0,12)	F35 (0,05)	F33 (0,14)
			F48 (0,04)			F37 (0,41)	F37 (0,47)
						F48 (0,01)	
Karar Verme Birimleri :	F43	F44	F45	F46	F47	F48	
Etkinlik Değerleri :	0,7521	0,9792	0,7401	0,7516	0,5802	1	
Referanslar Olma Sıklığı :						21	
Referanslar :	F22 (0,28)	F22 (0,09)	F33 (0,22)	F37 (0,71)	F33 (0,37)		
	F33 (0,02)	F33 (0,13)	F35 (0,24)		F37 (0,72)		
	F37 (0,63)	F37 (0,63)					
		F48 (0,02)					

EK-6 Kullanılan Kodlar

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using Microsoft.Office.Interop.Excel;
using iTextSharp.text.pdf;
using iTextSharp.text;
using System.Diagnostics;
using System.Windows.Forms.DataVisualization.Charting;

namespace VolkanSonerOzsoy
{
    public partial class AnaPencere : Form
    {
        public int KararVericiSayısı;
        public int DegiskenSayısı;
        private const double EPSILON = 1.0E-10;
        private double[,] tablo;
        private int KisitSayisi;
        private int DegiskenSayisi;
        private int[] basis;
        public static int Boyut = 200;
        public double M = 100000.0;
        public string[] TutulanSatırlar;
        double[,] c = new double[Boyut];
        public static double[,] girdi = new double[Boyut, Boyut];
        public static double[,] çıktı = new double[Boyut, Boyut];

        public AnaPencere()
        { InitializeComponent(); }

        private void AnaPencere_Load(object sender, EventArgs e)
        {
            this.Cursor = Cursors.WaitCursor;
            helpProvider1.SetShowHelp(this, true);
            char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
            int ToplamSütunSayısı = Boyut;
            int ToplamSatırSayısı = Boyut;
            for (int i = 0; i < 26; i++)
            {
                dGVVeriGirisArayüzü.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
                dGVVeriGirisArayüzü.Columns[i].Width = 62;
            }
            for (int i = 26; i < ToplamSütunSayısı; i++)
            {
                dGVVeriGirisArayüzü.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString());
                dGVVeriGirisArayüzü.Columns[i].Width = 62;
            }
            dGVVeriGirisArayüzü.RowCount = ToplamSatırSayısı;
            for (int i = 0; i < ToplamSatırSayısı; i++)
            {
                dGVVeriGirisArayüzü.Rows[i].Height = 20;
                dGVVeriGirisArayüzü.RowHeadersWidth = 35 + 7 * (dGVVeriGirisArayüzü.RowCount.ToString().Length);
                SatırNumarasıVermeFonksiyonu(dGVVeriGirisArayüzü);
                groupBoxWindow.Visible = false;
                int HeightFark = 151;
                int WidthFark = 23;
                int FormHeight = AnaPencere.ActiveForm.Size.Height;
                int FormWidth = AnaPencere.ActiveForm.Size.Width;
                tabControl1.Width = FormWidth;
                tabControl1.Height = FormHeight;
                groupBoxEtkinlikSkoru.Width = (FormWidth - WidthFark) / 2;
                groupBoxEtkinlikSkoru.Height = AnaPencere.ActiveForm.Size.Height - HeightFark;
            }
        }
    }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

groupBoxWindow.Width = FormWidth;
groupBoxWindow.Height = FormHeight;
groupBoxGrafik.Location = new System.Drawing.Point((FormWidth - WidthFark) / 2, 0);
groupBoxGrafik.Width = (FormWidth - 30) / 2;
groupBoxGrafik.Height = FormHeight - HeightFark;
chart1.Location = new System.Drawing.Point(10, 15);
chart1.Width = groupBoxGrafik.Width - 20;
chart1.Height = groupBoxGrafik.Height - 20;
groupBoxVeriGirisArayuzu.Width = FormWidth - WidthFark;
groupBoxVeriGirisArayuzu.Height = FormHeight - HeightFark;
this.Cursor = Cursors.Default; }

private void YeniTSMI_Click(object sender, EventArgs e)
{ DataGridleriTemizle(dGVVeriGirisArayüzü); }

private void AçTSMI_Click(object sender, EventArgs e)
{
    OpenFileDialog dialog = new OpenFileDialog();
    dialog.Filter = "Excel Dosyaları(*.xls;*.xlsx)|*.xls;*.xlsx|TXT Dosyaları(*.txt)|*.txt|Tüm Dosyalar (*.*)|*.*";
    dialog.Title = "Lütfen Dosya Seçiniz";
    if (dialog.ShowDialog() == DialogResult.OK)
    {
        string uzanti = Path.GetExtension(dialog.FileName.ToString());
        if (uzanti == ".txt")
            NotDefterindenOkuma(dialog);
        else if (uzanti == ".xls" || uzanti == ".xlsx")
            ExceldenOkuma(dialog);
        else
            MessageBox.Show("Dosya Okunamadı Başka Bir Türe Sahip Dosya Seçiniz...", "Uyarı", MessageBoxButtons.OK,
            MessageBoxIcon.Exclamation); } }

public void NotDefterindenOkuma(OpenFileDialog dialog)
{
    FileStream aç = new FileStream(dialog.FileName, FileMode.Open, FileAccess.Read);
    StreamReader aç2 = new StreamReader(aç, Encoding.GetEncoding(1254));
    KararVericiSayısı = 0;
    string isim;
    while ((isim = aç2.ReadLine()) != null)
    {
        TutulanSatırlar = isim.Split('\r');
        DegiskenSayısı = TutulanSatırlar.Length;
        for (int i = 0; i < TutulanSatırlar.Length; i++)
            dGVVeriGirisArayüzü[i, KararVericiSayısı].Value = TutulanSatırlar[i];
        KararVericiSayısı++;
    }
    aç.Close();
    aç2.Close();
}

public void ExceldenOkuma(OpenFileDialog dialog)
{ try { OleDbConnection baglanti = new OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
+ dialog.FileName + "; Extended Properties=Excel 12.0 Xml");
    baglanti.Open();
    string sorgu = "select * from [Sayfa1$] ";
    OleDbDataAdapter data_adaptor = new OleDbDataAdapter(sorgu, baglanti);
    baglanti.Close();
    System.Data.DataTable dt = new System.Data.DataTable();
    data_adaptor.Fill(dt);
    dGVGeçici.DataSource = dt;
    for (int j = 0; j < dGVGeçici.Rows.Count; j++)
        for (int i = 0; i < dGVGeçici.Columns.Count; i++)
        { dGVVeriGirisArayüzü[i, 0].Value = dGVGeçici.Columns[i].HeaderText.ToString();
        dGVVeriGirisArayüzü[i, j + 1].Value = dGVGeçici[i, j].Value; }
    data_adaptor.Dispose(); dt.Dispose(); baglanti.Close(); } catch (Exception ex)
{ MessageBox.Show("Hata: " + ex.ToString()); } }

```

EK-6 (Devam) Kullanılan Kodlar

```

private void pDfFeAktarTSMI_Click(object sender, EventArgs e)
{
    if (dGVETkinlikSkoru.Visible || dGVWindow.Visible)
    {
        if (dGVWindow.Visible)
        {
            PDfFeAktar(dGVWindow);
        }
        else PDfFeAktar(dGVETkinlikSkoru);
    }
    else MessageBox.Show("Analiz yapmadan aktarma yapamazsınız", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Stop);}

public void PDfFeAktar(DataGridView DGV)
{
    iTextSharp.text.Document document = new iTextSharp.text.Document();
    PdfWriter.GetInstance(document, new FileStream(@"D:\Analiz Sonuçları.pdf", FileMode.Create));
    BaseFont arial = BaseFont.CreateFont("C:\\windows\\fonts\\arial.ttf", BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
    if (document.IsOpen() == false)
    {
        document.Open();
        iTextSharp.text.Image img = iTextSharp.text.Image.GetInstance(@"PDFUstKisim.jpg");
        document.Add(img);
        for (int j = 0; j < DGV.Rows.Count; j++)
        {
            string text = "";
            for (int i = 0; i < DGV.Columns.Count; i++)
            {
                text = text + PDfFeAktarmaiçinTurkceKarakter(Convert.ToString(DGV[i, j].Value)) + " \t ";
            }
            document.Add(new Paragraph(text));
        }
        document.Close(); } }

public string PDfFeAktarmaiçinTurkceKarakter(string text)
{
    text = text.Replace("İ", "\u0130");
    text = text.Replace("ı", "\u0131");
    text = text.Replace("Ş", "\u015e");
    text = text.Replace("ş", "\u015f");
    text = text.Replace("Ğ", "\u011e");
    text = text.Replace("ğ", "\u011f");
    text = text.Replace("Ö", "\u00d6");
    text = text.Replace("ö", "\u00f6");
    text = text.Replace("ç", "\u00e7");
    text = text.Replace("Ç", "\u00c7");
    text = text.Replace("ü", "\u00fc");
    text = text.Replace("Ü", "\u00dc");
    return text; }

private void NotDeferineAktarTSMI_Click(object sender, EventArgs e)
{
    if (dGVETkinlikSkoru.Visible || dGVWindow.Visible)
    {
        SaveFileDialog dosyaKaydet = new SaveFileDialog();
        dosyaKaydet.Filter = "Text Dosyası(txt)*.txt";
        dosyaKaydet.InitialDirectory = "D:";
        dosyaKaydet.Title = "Kaydet";
        dosyaKaydet.DefaultExt = ".txt";
        if (dosyaKaydet.ShowDialog() == DialogResult.OK)
        {
            string uzanti = Path.GetExtension(dosyaKaydet.FileName.ToString());
            if (uzanti == ".txt")
            {
                if (dGVWindow.Visible)
                {
                    NotDeferineAktar(dosyaKaydet, dGVWindow);
                }
                else NotDeferineAktar(dosyaKaydet, dGVETkinlikSkoru);
            }
            else
            {
                MessageBox.Show("Kaydedeceğiniz Verinin Uzantısı 'txt' olmalıdır...", "Yanlış Uzanti", MessageBoxButtons.OK, MessageBoxIcon.Exclamation); } }
        else MessageBox.Show("Analiz yapmadan aktarma yapamazsınız", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Stop);}

```

EK-6 (Devam) Kullanılan Kodlar

```

public void NotDefterineAktar(SaveFileDialog dosyaKaydet, DataGridView AktarılacakDGV)
{
    StreamWriter Kaydet = new StreamWriter(dosyaKaydet.FileName);
    if (AktarılacakDGV.RowCount > 0)
    for (int i = 0; i < AktarılacakDGV.RowCount; i++)
    {
        string temp = "";
        for (int j = 0; j < AktarılacakDGV.ColumnCount; j++)
        {
            temp = temp + Convert.ToString(AktarılacakDGV[j, i].Value) + "\t" + "\t";
            if (j == AktarılacakDGV.ColumnCount - 1)
                break;
        }
        Kaydet.WriteLine(temp);
        if (i == AktarılacakDGV.RowCount - 1)
            break;
    }
    else
        MessageBox.Show("Etkinlik Skoru Hesaplandıktan Sonra Kaydetmeyi Deneyiniz...", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    MessageBox.Show("Dosyanız Başarıyla Kaydedildi...", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    Kaydet.Close(); }

private void wordeAktarTSMI_Click(object sender, EventArgs e)
{
    if (dGVEtkinlikSkoru.Visible || dGVWindow.Visible)
    {
        SaveFileDialog sfd = new SaveFileDialog();
        sfd.Filter = "Word Documents (*.doc)*.doc";
        sfd.FileName = "Analiz Sonuçları.doc";
        if (sfd.ShowDialog() == DialogResult.OK)
        {
            if (dGVWindow.Visible)
                WordeAktar(dGVWindow, sfd.FileName);
            else WordeAktar(dGVEtkinlikSkoru, sfd.FileName); } }
    else MessageBox.Show("Analiz yapmadan aktarma yapamazsınız", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Stop);}

public void WordeAktar(DataGridView dGV, string filename)
{
    string stOutput = "";
    string sHeaders = "";
    for (int j = 0; j < dGV.Columns.Count; j++)
        sHeaders = sHeaders.ToString() + Convert.ToString(dGV.Columns[j].HeaderText) + "\t";
    stOutput += sHeaders + "\r\n";
    for (int i = 0; i < dGV.RowCount - 1; i++)
    {
        string stLine = "";
        for (int j = 0; j < dGV.Rows[i].Cells.Count; j++)
            stLine = stLine.ToString() + Convert.ToString(dGV.Rows[i].Cells[j].Value) + "\t";
        stOutput += stLine + "\r\n";
    }
    Encoding utf16 = Encoding.GetEncoding(1254);
    byte[] output = utf16.GetBytes(stOutput);
    FileStream fs = new FileStream(filename, FileMode.Create);
    BinaryWriter bw = new BinaryWriter(fs);
    bw.Write(output, 0, output.Length);
    bw.Flush(); bw.Close(); fs.Close(); }

private void ExceleAktarTSMI_Click(object sender, EventArgs e)
{ if (dGVEtkinlikSkoru.Visible || dGVWindow.Visible)
  { if (dGVWindow.Visible)
    ExceleAktar(dGVWindow);
    else ExceleAktar(dGVEtkinlikSkoru); }
  else MessageBox.Show("Analiz yapmadan aktarma yapamazsınız", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Stop);}

```

EK-6 (Devam) Kullanılan Kodlar

```

public void ExceleAktar(DataGridView ExceleAktarilacakDGV)
{
    ApplicationClass excel = new ApplicationClass();
    excel.Visible = true;
    object Missing = Type.Missing;
    Workbook workbook = excel.Workbooks.Add(Missing);
    Worksheet sheet1 = (Worksheet)workbook.Sheets[1];
    int StartCol = 1;
    int StartRow = 1;
    for (int j = 0; j < ExceleAktarilacakDGV.Columns.Count; j++)
    {
        Range myRange = (Range)sheet1.Cells[StartRow, StartCol + j];
        myRange.Value2 = ExceleAktarilacakDGV.Columns[j].HeaderText;
    }
    StartRow++;
    for (int i = 0; i < ExceleAktarilacakDGV.Rows.Count; i++)
    {
        for (int j = 0; j < ExceleAktarilacakDGV.Columns.Count; j++)
        {
            try {
                Range myRange = (Range)sheet1.Cells[StartRow + i, StartCol + j];
                myRange.Value2 = ExceleAktarilacakDGV[j, i].Value == null ? "" : ExceleAktarilacakDGV[j, i].Value;
                myRange.Select();
            } catch { ; } } } }

private DataGridViewCell GetStartCell(DataGridView dgvView)
{
    if (dgvView.SelectedCells.Count == 0)
        return null;
    int rowIndex = dgvView.Rows.Count - 1;
    int colIndex = dgvView.Columns.Count - 1;
    foreach (DataGridViewCell dgvCell in dgvView.SelectedCells)
    {
        if (dgvCell.RowIndex < rowIndex)
            rowIndex = dgvCell.RowIndex;
        if (dgvCell.ColumnIndex < colIndex)
            colIndex = dgvCell.ColumnIndex;
    }
    return dgvView[colIndex, rowIndex];
}

private Dictionary<int, Dictionary<int, string>> ClipBoardValues(string clipboardValue)
{
    Dictionary<int, Dictionary<int, string>>
    copyValues = new Dictionary<int, Dictionary<int, string>>();
    String[] lines = clipboardValue.Split("\n");
    for (int i = 0; i <= lines.Length - 1; i++)
    {
        copyValues[i] = new Dictionary<int, string>();
        String[] lineContent = lines[i].Split('\t');
        if (lineContent.Length == 0)
            copyValues[i][0] = string.Empty;
        else
        {
            for (int j = 0; j <= lineContent.Length - 1; j++)
                copyValues[i][j] = lineContent[j];
        }
    }
    return copyValues;
}

private void dGVVeriGirisArayüzü_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)
{
    if (dGVVeriGirisArayüzü.SelectedCells.Count > 0)
        dGVVeriGirisArayüzü.ContextMenuStrip = contextMenuStrip2;
}

private void kesToolStripMenuItem_Click(object sender, EventArgs e)
{
    Kopyala();
    foreach (DataGridViewCell dgvCell in dGVVeriGirisArayüzü.SelectedCells)
        dgvCell.Value = string.Empty;
}

private void kopyalaToolStripMenuItem_Click(object sender, EventArgs e)
{
    Kopyala();
}

```

EK-6 (Devam) Kullanılan Kodlar

```

private void Kopyala()
{
    DataObject dataObj = dGVVeriGirisArayüzü.GetClipboardContent();
    if (dataObj != null)
        Clipboard.SetDataObject(dataObj);
}

private void yapıştırToolStripMenuItem_Click(object sender, EventArgs e)
{
    Yapıştır();
}

private void Yapıştır()
{
    if (dGVVeriGirisArayüzü.SelectedCells.Count == 0)
    {
        MessageBox.Show("Please select a cell", "Paste",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    DataGridViewCell startCell = GetStartCell(dGVVeriGirisArayüzü);
    Dictionary<int, Dictionary<int, string>> cbValue = Clipboard.Values(Clipboard.GetText());
    int iRowIndex = startCell.RowIndex;
    foreach (int rowKey in cbValue.Keys)
    {
        int iColIndex = startCell.ColumnIndex;
        foreach (int cellKey in cbValue[rowKey].Keys)
        {
            if (iColIndex <= dGVVeriGirisArayüzü.Columns.Count - 1 && iRowIndex <= dGVVeriGirisArayüzü.Rows.Count - 1)
            {
                DataGridViewCell cell = dGVVeriGirisArayüzü[iColIndex, iRowIndex];
                cell.Value = cbValue[rowKey][cellKey];
                iColIndex++;
            }
            iRowIndex++;
        }
    }
}

private void AnalizTSMI_Click(object sender, EventArgs e)
{
    try {
        if (dGVVeriGirisArayüzü[0, 0].Value != null || dGVVeriGirisArayüzü[1, 1].Value != null)
        {
            AnalizFormu AnalizF = new AnalizFormu();
            AnalizF.anaform = this;
            AnalizF.Show();
        }
        else
        {
            MessageBox.Show("Veriler olmadan analiz yapamazsınız", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Stop);
        }
    } catch (Exception) {
        MessageBox.Show("Veriler olmadan analiz yapamazsınız", "Veri Yok", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }
}

public void BütünAnalizler(int n, int gs, int cs, int comboBox1SelectedIndex, double[,] girdi, double[,] çıktı, int[] KontrolEdilemeyenGirdiSıraNo, int[] KontrolEdilemeyenÇıktıSıraNo)
{
    switch (comboBox1SelectedIndex)
    {
        case 0: AnaSimplex("CCRDİ"); break;
        case 1: AnaSimplex("CCRPO"); break;
        case 2: AnaSimplex("BCCDİ"); break;
        case 3: AnaSimplex("BCCPO"); break;
        default: MessageBox.Show("yok"); break;
    }
}

public void AnaSimplex(string HangiModel)
{
    DegiskenSayısı = -1;
    for (int i = 0; i < dGVVeriGirisArayüzü.Columns.Count; i++)
    {
        if (dGVVeriGirisArayüzü[i, 0].Value == null)
            break;
        DegiskenSayısı++;
    }
    KararVericiSayısı = 0;
    for (int i = 1; i < dGVVeriGirisArayüzü.Rows.Count; i++)
    {
        if (dGVVeriGirisArayüzü[0, i].Value == null)
            break;
        KararVericiSayısı++;
    }
    int[] GirdiSıraNo = new int[DegiskenSayısı];
    int[] ÇıktıSıraNo = new int[DegiskenSayısı];
    int GirdiSayısı = 0;
    int ÇıktıSayısı = 0;
}

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int i = 0; i <= DegiskenSayısı; i++)
{
    if (dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'G' ||
        dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'g')
    {
        GirdiSıraNo[GirdiSayısı] = i;
        GirdiSayısı++;
    }
    else if (dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'Ç' ||
        dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'ç')
    {
        ÇıktıSıraNo[ÇıktıSayısı] = i;
        ÇıktıSayısı++; } }
double[,] girdi = new double[GirdiSayısı, KararVericiSayısı];
double[,] çıktı = new double[ÇıktıSayısı, KararVericiSayısı];

for (int i = 0; i < KararVericiSayısı; i++)
{
    for (int j = 0; j < GirdiSayısı; j++)
        girdi[j, i] = Convert.ToDouble(dGVVeriGirisArayüzü[GirdiSıraNo[j], i + 1].Value);
    for (int j = 0; j < ÇıktıSayısı; j++)
        çıktı[j, i] = Convert.ToDouble(dGVVeriGirisArayüzü[ÇıktıSıraNo[j], i + 1].Value);
}

switch (HangiModel)
{
    case "CCRDİ": CCR_GIRDİ(DegiskenSayısı, KararVericiSayısı, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
    case "CCRPO": CCR_ÇIKTI(DegiskenSayısı, KararVericiSayısı, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
    case "BCCDİ": BCC_GIRDİ(DegiskenSayısı, KararVericiSayısı, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
    case "BCCPO": BCC_ÇIKTI(DegiskenSayısı, KararVericiSayısı, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
    default: MessageBox.Show("Girilen model bulunamadı!"); break; } }

public void CCR_GIRDİ(int DegiskenSayısı, int KararVericiSayısı, int GirdiSayısı, int ÇıktıSayısı, double[,] girdi, double[,]
    çıktı)
{
    int EtkinlikToplamSütunSayısı = Math.Max(KararVericiSayısı, DegiskenSayısı) + 1;
    int EtkinlikToplamSatırSayısı = Math.Max(KararVericiSayısı, DegiskenSayısı) + 3;
    char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
    if (EtkinlikToplamSütunSayısı < 26)
        for (int i = 0; i < EtkinlikToplamSütunSayısı; i++)
        {
            dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
    else if (EtkinlikToplamSütunSayısı > 26)
    {
        for (int i = 0; i < 26; i++)
        {
            dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
        for (int i = 26; i < EtkinlikToplamSütunSayısı; i++)
        {
            dGVEtkinlikSkoru.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() +
                Harfler[i % 26]).ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
    }
    dGVEtkinlikSkoru.RowCount = EtkinlikToplamSatırSayısı + 2;
    for (int i = 0; i < EtkinlikToplamSatırSayısı; i++)
        dGVEtkinlikSkoru.Rows[i].Height = 20;
    dGVEtkinlikSkoru.RowHeadersWidth = 35 + 7 * (dGVEtkinlikSkoru.RowCount.ToString().Length);
    SatırNumarasıVermeFonksiyonu(dGVEtkinlikSkoru);
    dGVEtkinlikSkoru[0, 0].Value = "Karar Verme Birimleri :";
    dGVEtkinlikSkoru[0, 1].Value = "Etkinlik Değerleri :";
    dGVEtkinlikSkoru[0, 2].Value = "Referanslar Olma Sıklığı :";
    dGVEtkinlikSkoru[0, 3].Value = "Referanslar :";
    int[] ReferansAdedi = new int[KararVericiSayısı];

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int k = 0; k < KararVericiSayısı; k++)
{
double[] c = new double[DegiskenSayısı];
for (int i = 0; i < ÇıktıSayısı; i++)
c[i] = (double)çıktı[i, k];

double[] b = new double[KararVericiSayısı + 1];
for (int i = 0; i < KararVericiSayısı; i++)
b[i] = 0;
b[KararVericiSayısı] = 1;

double[,] A = new double[KararVericiSayısı + 1, DegiskenSayısı];
for (int i = 0; i < KararVericiSayısı; i++)
{
for (int j = 0; j < ÇıktıSayısı; j++)
A[i, j] = çıktı[j, i];
for (int j = ÇıktıSayısı; j < ÇıktıSayısı + GirdiSayısı; j++)
A[i, j] = -girdi[j - ÇıktıSayısı, i];
}
for (int j = 0; j < GirdiSayısı; j++)
A[KararVericiSayısı, ÇıktıSayısı + j] = girdi[j, k];
Simplex(A, b, c);

dGVEtkinlikSkoru[k + 1, 0].Value = dGVVeriGirisArayüzü[0, k + 1].Value;
dGVEtkinlikSkoru[k + 1, 1].Value = Math.Round(valuee(), 4);
double[] y = dual();
int gec = 0;
for (int j = 0; j < y.Length - 1; j++)
{
if (Math.Round(valuee(), 4) == 1)
break;
if (y[j] > 0)
{
dGVEtkinlikSkoru[k + 1, 3 + gec].Value = (Convert.ToString(dGVVeriGirisArayüzü[0, j + 1].Value) + " (" + Math.Round(y[j],
2) + ")");
ReferansAdedi[j]++;
gec++; } } }
for (int i = 0; i < KararVericiSayısı; i++)
{
if (ReferansAdedi[i] == 0)
dGVEtkinlikSkoru[i + 1, 2].Value = null;
else
dGVEtkinlikSkoru[i + 1, 2].Value = ReferansAdedi[i];
} GrafikCiz(KararVericiSayısı); }

public void BCC_GIRDI(int DegiskenSayısı, int KararVericiSayısı, int GirdiSayısı, int ÇıktıSayısı, double[,] girdi, double[,]
çıktı)
{
int EtkinlikToplamSütunSayısı = KararVericiSayısı + 1;
int EtkinlikToplamSatırSayısı = KararVericiSayısı + 3;
char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
if (EtkinlikToplamSütunSayısı < 26)
for (int i = 0; i < EtkinlikToplamSütunSayısı; i++)
{
dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
dGVEtkinlikSkoru.Columns[i].Width = 62;
}
else if (EtkinlikToplamSütunSayısı > 26)
{
for (int i = 0; i < 26; i++)
{
dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
dGVEtkinlikSkoru.Columns[i].Width = 62;
}
}
}

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int i = 26; i < EtkinlikToplamSütunSayısı; i++)
{
    dGVEtkinlikSkoru.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() +
    Harfler[i % 26]).ToString());
    dGVEtkinlikSkoru.Columns[i].Width = 62;
}
}
dGVEtkinlikSkoru.RowCount = EtkinlikToplamSatırSayısı + 2;
for (int i = 0; i < EtkinlikToplamSatırSayısı; i++)
dGVEtkinlikSkoru.Rows[i].Height = 20;
dGVEtkinlikSkoru.RowHeadersWidth = 35 + 7 * (dGVEtkinlikSkoru.RowCount.ToString().Length);
SatırNumarasıVermeFonksiyonu(dGVEtkinlikSkoru);
dGVEtkinlikSkoru[0, 0].Value = "Karar Verme Birimleri :";
dGVEtkinlikSkoru[0, 1].Value = "Etkinlik Değerleri :";
dGVEtkinlikSkoru[0, 2].Value = "Referanslar Olma Sıklığı :";
dGVEtkinlikSkoru[0, 3].Value = "Referanslar :";
int[] ReferansAdedi = new int[KararVericiSayısı];

for (int k = 0; k < KararVericiSayısı; k++)
{
    double[] c = new double[DegiskenSayısı + 1];
    for (int i = 0; i < ÇıktıSayısı; i++)
    c[i] = (double)çıktı[i, k];
    c[DegiskenSayısı] = 1;

    double[] b = new double[KararVericiSayısı + 1];
    for (int i = 0; i < KararVericiSayısı; i++)
    b[i] = 0;
    b[KararVericiSayısı] = 1;

    double[,] A = new double[KararVericiSayısı + 1, DegiskenSayısı + 1];
    for (int i = 0; i < KararVericiSayısı; i++)
    {
        for (int j = 0; j < ÇıktıSayısı; j++)
        A[i, j] = çıktı[j, i];
        for (int j = ÇıktıSayısı; j < ÇıktıSayısı + GirdiSayısı; j++)
        A[i, j] = -girdi[j - ÇıktıSayısı, i];
        A[i, DegiskenSayısı] = 1;
    }
    for (int j = 0; j < GirdiSayısı; j++)
    A[KararVericiSayısı, ÇıktıSayısı + j] = girdi[j, k];

    Simplex(A, b, c);
    dGVEtkinlikSkoru[k + 1, 0].Value = dGVVeriGirisArrayüzü[0, k + 1].Value;
    dGVEtkinlikSkoru[k + 1, 1].Value = Math.Round(valuee(), 4);
    double[] y = dual();
    int gec = 0;
    for (int j = 0; j < y.Length - 1; j++)
    {
        if (Math.Round(valuee(), 4) == 1)
        break;
        if (y[j] > 0)
        {
            dGVEtkinlikSkoru[k + 1, 3 + gec].Value = (Convert.ToString(dGVVeriGirisArrayüzü[0, j + 1].Value) + " (" + Math.Round(y[j],
            2) + ")");
            ReferansAdedi[j]++;
            gec++; } }
    for (int i = 0; i < KararVericiSayısı; i++)
    {
        if (ReferansAdedi[i] == 0)
        dGVEtkinlikSkoru[i + 1, 2].Value = null;
        else
        dGVEtkinlikSkoru[i + 1, 2].Value = ReferansAdedi[i];
    }
    GrafikCiz(KararVericiSayısı);
}

```

EK-6 (Devam) Kullanılan Kodlar

```

public void CCR_CIKTI(int DegiskenSayısı, int KararVericiSayısı, int GirdiSayısı, int ÇıktıSayısı, double[,] girdi, double[,]
çıktı)
{
    int EtkinlikToplamSütunSayısı = KararVericiSayısı + 1;
    int EtkinlikToplamSatırSayısı = KararVericiSayısı + 3;
    char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
    if (EtkinlikToplamSütunSayısı < 26)
        for (int i = 0; i < EtkinlikToplamSütunSayısı; i++)
        {
            dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
    else if (EtkinlikToplamSütunSayısı > 26)
    {
        for (int i = 0; i < 26; i++)
        {
            dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
        for (int i = 26; i < EtkinlikToplamSütunSayısı; i++)
        {
            dGVEtkinlikSkoru.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() +
            Harfler[i % 26]).ToString());
            dGVEtkinlikSkoru.Columns[i].Width = 62;
        }
    }
    dGVEtkinlikSkoru.RowCount = EtkinlikToplamSatırSayısı + 2;
    for (int i = 0; i < EtkinlikToplamSatırSayısı; i++)
        dGVEtkinlikSkoru.Rows[i].Height = 20;
    dGVEtkinlikSkoru.RowHeadersWidth = 35 + 7 * (dGVEtkinlikSkoru.RowCount.ToString().Length);
    SatırNumarasıVermeFonksiyonu(dGVEtkinlikSkoru);
    dGVEtkinlikSkoru[0, 0].Value = "Karar Verme Birimleri :";
    dGVEtkinlikSkoru[0, 1].Value = "Etkinlik Değerleri :";
    dGVEtkinlikSkoru[0, 2].Value = "Referanslar Olma Sıklığı :";
    dGVEtkinlikSkoru[0, 3].Value = "Referanslar :";
    int[] ReferansAdedi = new int[KararVericiSayısı];
    for (int k = 0; k < KararVericiSayısı; k++)
    {
        double[] c = new double[KararVericiSayısı + DegiskenSayısı + 1];
        c[0] = 1;
        for (int i = 1; i <= KararVericiSayısı; i++)
            c[i] = 0;
        for (int i = KararVericiSayısı + 1; i < KararVericiSayısı + DegiskenSayısı + 1; i++)
            c[i] = -EPSILON;
        double[] b = new double[DegiskenSayısı];
        for (int i = 0; i < GirdiSayısı; i++)
            b[i + ÇıktıSayısı] = girdi[i, k];
        double[,] A = new double[DegiskenSayısı, KararVericiSayısı + DegiskenSayısı + 1];
        for (int i = 1; i < KararVericiSayısı + 1; i++)
        {
            for (int j = 0; j < ÇıktıSayısı; j++)
            {
                A[j, i] = -çıktı[j, i - 1];
                A[j, 0] = çıktı[j, k];
            }
            for (int j = 0; j < GirdiSayısı; j++)
            {
                A[j + ÇıktıSayısı, i] = girdi[j, i - 1];
            }
        }
        for (int i = KararVericiSayısı + 1; i < KararVericiSayısı + DegiskenSayısı + 1; i++)
            A[i - KararVericiSayısı - 1, i] = 1;
        Simplex(A, b, c);
        dGVEtkinlikSkoru[k + 1, 0].Value = dGVVeriGirisArrayüzü[0, k + 1].Value;
        dGVEtkinlikSkoru[k + 1, 1].Value = Math.Round(valuee(), 4);
    }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

double[] x = primal();
int gec = 0;
for (int i = 1; i < x.Length; i++)
{ if (Math.Round(valuee(), 4) == 1)
break;
if (x[i] > 0)
{
dGVEtkinlikSkoru[k + 1, 3 + gec].Value = (Convert.ToString(dGVVeriGirisArrayu[0, i].Value) + " (" + Math.Round(x[i], 2)
+ ")");
ReferansAdedi[i - 1]++;
gec++; } }
for (int i = 0; i < KararVericiSayısı; i++)
{
if (ReferansAdedi[i] == 0)
dGVEtkinlikSkoru[i + 1, 2].Value = null;
else
dGVEtkinlikSkoru[i + 1, 2].Value = ReferansAdedi[i];
} GrafikCiz(KararVericiSayısı); }

public void BCC_CIKTI(int DegiskenSayısı, int KararVericiSayısı, int GirdiSayısı, int ÇıktıSayısı, double[,] girdi, double[,]
çıktı)
{
int EtkinlikToplamSütunSayısı = KararVericiSayısı + 1;
int EtkinlikToplamSatırSayısı = KararVericiSayısı + 3;
char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
if (EtkinlikToplamSütunSayısı < 26)
for (int i = 0; i < EtkinlikToplamSütunSayısı; i++)
{
dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
dGVEtkinlikSkoru.Columns[i].Width = 62;
}
else if (EtkinlikToplamSütunSayısı > 26)
{
for (int i = 0; i < 26; i++)
{
dGVEtkinlikSkoru.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
dGVEtkinlikSkoru.Columns[i].Width = 62;
}
for (int i = 26; i < EtkinlikToplamSütunSayısı; i++)
{
dGVEtkinlikSkoru.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() +
Harfler[i % 26]).ToString());
dGVEtkinlikSkoru.Columns[i].Width = 62; } }
dGVEtkinlikSkoru.RowCount = EtkinlikToplamSatırSayısı + 2;
for (int i = 0; i < EtkinlikToplamSatırSayısı; i++)
dGVEtkinlikSkoru.Rows[i].Height = 20;
dGVEtkinlikSkoru.RowHeadersWidth = 35 + 7 * (dGVEtkinlikSkoru.RowCount.ToString().Length);
SatırNumarasıVermeFonksiyonu(dGVEtkinlikSkoru);
dGVEtkinlikSkoru[0, 0].Value = "Karar Verme Birimleri :";
dGVEtkinlikSkoru[0, 1].Value = "Etkinlik Değerleri :";
dGVEtkinlikSkoru[0, 2].Value = "Referanslar Olma Sıklığı :";
dGVEtkinlikSkoru[0, 3].Value = "Referanslar :";
int[] ReferansAdedi = new int[KararVericiSayısı];
for (int k = 0; k < KararVericiSayısı; k++)
{
double[] c = new double[KararVericiSayısı + DegiskenSayısı + 1];
c[0] = 1;
for (int i = 1; i <= KararVericiSayısı; i++)
c[i] = 0;
for (int i = KararVericiSayısı + 1; i < KararVericiSayısı + DegiskenSayısı + 1; i++)
c[i] = -EPSILON;
double[] b = new double[DegiskenSayısı + 1];
for (int i = 0; i < GirdiSayısı; i++)
b[i + ÇıktıSayısı] = girdi[i, k];
b[DegiskenSayısı] = 1;
double[,] A = new double[DegiskenSayısı + 1, KararVericiSayısı + DegiskenSayısı + 1];

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int i = 1; i < KararVericiSayısı + 1; i++)
{
    for (int j = 0; j < ÇıktıSayısı; j++)
    {
        A[j, i] = -çıktı[j, i - 1];
        A[j, 0] = çıktı[j, k];
    }
    for (int j = 0; j < GirdiSayısı; j++)
    {
        A[j + ÇıktıSayısı, i] = girdi[j, i - 1];
    }
    A[DegiskenSayısı, i] = 1;
}
for (int i = KararVericiSayısı + 1; i < KararVericiSayısı + DegiskenSayısı + 1; i++)
    A[i - KararVericiSayısı - 1, i] = 1;
Simplex(A, b, c);
dGVEtkinlikSkoru[k + 1, 0].Value = dGVVeriGirisArayüzü[0, k + 1].Value;
dGVEtkinlikSkoru[k + 1, 1].Value = Math.Round(valuee(), 4);
double[] x = primal();
int gec = 0;
for (int i = 1; i < x.Length; i++)
{
    if (Math.Round(valuee(), 4) == 1)
        break;
    if (x[i] > 0)
    {
        dGVEtkinlikSkoru[k + 1, 3 + gec].Value = (Convert.ToString(dGVVeriGirisArayüzü[0, i].Value) + " (" + Math.Round(x[i], 2) + ")");
        ReferansAdedi[i - 1]++;
        gec++;
    }
}
for (int i = 0; i < KararVericiSayısı; i++)
{
    if (ReferansAdedi[i] == 0)
        dGVEtkinlikSkoru[i + 1, 2].Value = null;
    else
        dGVEtkinlikSkoru[i + 1, 2].Value = ReferansAdedi[i];
}
GrafikCiz(KararVericiSayısı);

public void Simplex(double[,] A, double[] b, double[] c)
{
    KisitSayisi = b.Length;
    DegiskenSayisi = c.Length;
    tablo = new double[KisitSayisi + 1, DegiskenSayisi + KisitSayisi + 1];
    for (int i = 0; i < KisitSayisi; i++)
    for (int j = 0; j < DegiskenSayisi; j++)
        tablo[i, j] = A[i, j];
    for (int i = 0; i < KisitSayisi; i++)
        tablo[i, DegiskenSayisi + i] = 1.0;
    for (int j = 0; j < DegiskenSayisi; j++)
        tablo[KisitSayisi, j] = c[j];
    for (int i = 0; i < KisitSayisi; i++)
        tablo[i, KisitSayisi + DegiskenSayisi] = b[i];
    basis = new int[KisitSayisi];
    for (int i = 0; i < KisitSayisi; i++)
        basis[i] = DegiskenSayisi + i;
    solve();
    Debug.Assert(check(A, b, c));
}

private void solve()
{
    while (true)
    {
        int q = bland();
        if (q == -1) // optimal
            break;
        int p = minRatioRule(q);
        if (p == -1)
            throw new Exception("Sınırsız Çözüm");
        pivot(p, q);
        basis[p] = q;
    }
    private int bland()
    {
        for (int j = 0; j < KisitSayisi + DegiskenSayisi; j++)
            if (tablo[KisitSayisi, j] > 0)
                return j;
        return -1; // optimal
    }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

private int dantzig()
{ int q = 0;
  for (int j = 1; j < KisitSayisi + DegiskenSayisi; j++)
    if (tablo[KisitSayisi, j] > tablo[KisitSayisi, q])
      q = j;
  if (tablo[KisitSayisi, q] <= 0) // optimal
    return -1;
  else
    return q; }

private int minRatioRule(int q)
{ int p = -1;
  for (int i = 0; i < KisitSayisi; i++)
  {
    if (tablo[i, q] <= 0)
      continue;
    else if (p == -1)
      p = i;
    else if ((tablo[i, KisitSayisi + DegiskenSayisi] / tablo[i, q]) < (tablo[p, KisitSayisi + DegiskenSayisi] / tablo[p, q]))
      p = i; } return p; }

private void pivot(int p, int q)
{ for (int i = 0; i <= KisitSayisi; i++)
  for (int j = 0; j <= KisitSayisi + DegiskenSayisi; j++)
    if (i != p && j != q)
      tablo[i, j] -= tablo[p, j] * tablo[i, q] / tablo[p, q];
  for (int i = 0; i <= KisitSayisi; i++)
    if (i != p)
      tablo[i, q] = 0.0;
  for (int j = 0; j <= KisitSayisi + DegiskenSayisi; j++)
    if (j != q)
      tablo[p, j] /= tablo[p, q];
  tablo[p, q] = 1.0; }

public virtual double valuee()
{ return -tablo[KisitSayisi, KisitSayisi + DegiskenSayisi]; }

public virtual double[] primal()
{ double[] x = new double[DegiskenSayisi];
  for (int i = 0; i < KisitSayisi; i++)
    if (basis[i] < DegiskenSayisi)
      x[basis[i]] = tablo[i, KisitSayisi + DegiskenSayisi];
  return x; }

public virtual double[] dual()
{ double[] y = new double[KisitSayisi];
  for (int i = 0; i < KisitSayisi; i++)
    y[i] = -tablo[KisitSayisi, DegiskenSayisi + i];
  return y; }

private bool isPrimalFeasible(double[,] A, double[] b)
{ double[] x = primal();
  for (int j = 0; j < x.Length; j++)
    if (x[j] < 0.0)
      { MessageBox.Show("x[" + j + "] = " + x[j] + " negatif");
        return false; }
  for (int i = 0; i < KisitSayisi; i++)
  { double sum = 0.0;
    for (int j = 0; j < DegiskenSayisi; j++)
      sum += A[i, j] * x[j];
    if (sum > b[i] + EPSILON)
      { MessageBox.Show("Primal uygun değil");
        MessageBox.Show("b[" + i + "] = " + b[i] + ", Toplam = " + sum);
        return false; } }
  return true; }

```

EK-6 (Devam) Kullanılan Kodlar

```

private bool isDualFeasible(double[,] A, double[] c)
{
    double[] y = dual();
    for (int i = 0; i < y.Length; i++)
        if (y[i] < 0.0)
        {
            MessageBox.Show("y[" + i + "] = " + y[i] + " negatif");
            return false; }
    for (int j = 0; j < DegiskenSayisi; j++)
    {
        double sum = 0.0;
        for (int i = 0; i < KisitSayisi; i++)
            sum += A[i, j] * y[i];
        if (sum < c[j] - EPSILON)
        {
            MessageBox.Show("Dual uygun değil");
            MessageBox.Show("c[" + j + "] = " + c[j] + ", Toplam = " + sum);
            return false; } }
    return true; }

private bool isOptimal(double[] b, double[] c)
{
    double[] x = primal();
    double[] y = dual();
    double value = valuee();
    double value1 = 0.0;
    for (int j = 0; j < x.Length; j++)
        value1 += c[j] * x[j];
    double value2 = 0.0;
    for (int i = 0; i < y.Length; i++)
        value2 += y[i] * b[i];
    if (Math.Abs(value - value1) > EPSILON || Math.Abs(value - value2) > EPSILON)
    {
        MessageBox.Show("Değerler = " + value + ", cx = " + value1 + ", yb = " + value2);
        return false; }
    return true; }

private bool check(double[,] A, double[] b, double[] c)
{
    return isPrimalFeasible(A, b) && isDualFeasible(A, c) && isOptimal(b, c); }

public virtual void show()
{
    MessageBox.Show("M = " + KisitSayisi);
    MessageBox.Show("N = " + DegiskenSayisi);
    for (int i = 0; i <= KisitSayisi; i++)
        for (int j = 0; j <= KisitSayisi + DegiskenSayisi; j++)
            MessageBox.Show("%7.2f ", tablo[i, j].ToString());
    MessageBox.Show("value = " + valuee());
    for (int i = 0; i < KisitSayisi; i++)
        if (basis[i] < DegiskenSayisi)
            MessageBox.Show("x_" + basis[i] + " = " + tablo[i, KisitSayisi + DegiskenSayisi]); }

public void WindowDGVAlaniAyırma(int Periyot, int Genişlik, int KararVericiSayısı, string[] KVBİsimleri, string[] Periyotİsimleri)
{
    int WindowDGVToplamSütunSayısı = Periyot + 2;
    int HerKVBneAitSatır = Periyot - Genişlik + 1;
    int WindowDGVToplamSatırSayısı = KararVericiSayısı * (HerKVBneAitSatır + 2) - 1;
    char[] Harfler = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
    if (WindowDGVToplamSütunSayısı < 26)
        for (int i = 0; i < WindowDGVToplamSütunSayısı; i++)
        {
            dGVWindow.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
            dGVWindow.Columns[i].Width = 62;
        }
    else if (WindowDGVToplamSütunSayısı > 26)
    {
        for (int i = 0; i < 26; i++)
            dGVWindow.Columns.Add(Harfler[i].ToString(), Harfler[i].ToString());
        dGVWindow.Columns[i].Width = 62; }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int i = 26; i < WindowDGVToplamSütunSayısı; i++)
{
    dGVWindow.Columns.Add((Harfler[i / 26 - 1].ToString() + Harfler[i % 26]).ToString(), (Harfler[i / 26 - 1].ToString() +
    Harfler[i % 26]).ToString());
    dGVWindow.Columns[i].Width = 62; } }
dGVWindow.RowCount = WindowDGVToplamSatırSayısı + 2;
for (int i = 0; i < WindowDGVToplamSatırSayısı; i++)
dGVWindow.Rows[i].Height = 20;
dGVWindow.RowHeadersWidth = 35 + 7 * (dGVWindow.RowCount.ToString().Length);
SatırNumarasıVermeFonksiyonu(dGVWindow);
for (int i = 0; i < Periyot; i++)
{ dGVWindow[i + 1, 0].Value = Periyotİsimleri[i]; }
dGVWindow[WindowDGVToplamSütunSayısı - 1, 0].Value = "Ortalama";
for (int i = 1; i < WindowDGVToplamSatırSayısı; i = i + HerKVBneAitSatır + 2)
{
    dGVWindow[0, i].Value = KVBİsimleri[i / (HerKVBneAitSatır + 2)];
    dGVWindow[0, i + HerKVBneAitSatır].Value = "Ortalama"; } }

public void SatırNumarasıVermeFonksiyonu(DataGridView GelenDataGridView)
{
    if (GelenDataGridView != null)
    for (int count = 0; count <= (GelenDataGridView.Rows.Count - 1); count++)
    GelenDataGridView.Rows[count].HeaderCell.Value = string.Format((count + 1).ToString(), "0");
    GelenDataGridView.RowsAdded += delegate(object oo, DataGridViewRowsAddedEventArgs ee)
    {
        for (int i = ee.RowIndex; i < ee.RowIndex + ee.RowCount; i++)
        GelenDataGridView.Rows[i].HeaderCell.Value = (i + 1).ToString(); } }

public void DataGridViewleriTemizle(DataGridView DataGridView)
{
    for (int i = 0; i < DataGridView.Columns.Count; i++)
    for (int j = 0; j < DataGridView.Rows.Count; j++)
    DataGridView.Rows[j].Cells[i].Value = ""; }

public void GrafikCiz(int n)
{
    if (AnalizFormu.Periyot < 2)
    {
        string[] seriesArray = new string[n];
        double[] pointsArray = new double[n];
        for (int i = 0; i < n; i++)
        {
            seriesArray[i] = dGVEtkinlikSkoru[i + 1, 0].Value.ToString();
            pointsArray[i] = (double)dGVEtkinlikSkoru[i + 1, 1].Value;
            System.Windows.Forms.DataVisualization.Charting.Series series = chart1.Series.Add(seriesArray[i]);
            series.Label = pointsArray[i].ToString();
            series.AxisLabel = "Karar Verme Birimleri";
            series.Points.Add(pointsArray[i]);
        }
        this.chart1.Titles.Add("Etkinlik Grafiği"); } }

private void AnaPencere_SizeChanged(object sender, EventArgs e)
{
    int HeightFark = 151;
    int WidthFark = 23;
    int FormHeight = AnaPencere.ActiveForm.Size.Height;
    int FormWidth = AnaPencere.ActiveForm.Size.Width;
    tabControl1.Width = FormWidth;
    tabControl1.Height = FormHeight;
    groupBoxEtkinlikSkoru.Width = (FormWidth - WidthFark) / 2;
    groupBoxEtkinlikSkoru.Height = AnaPencere.ActiveForm.Size.Height - HeightFark;
    groupBoxWindow.Width = FormWidth;
    groupBoxWindow.Height = FormHeight;
    groupBoxGrafik.Location = new System.Drawing.Point((FormWidth - WidthFark) / 2, 0);
    groupBoxGrafik.Width = (FormWidth - 30) / 2;
    groupBoxGrafik.Height = FormHeight - HeightFark;
    chart1.Location = new System.Drawing.Point(10, 15);
    chart1.Width = groupBoxGrafik.Width - 20;
    chart1.Height = groupBoxGrafik.Height - 20;
    groupBoxVeriGirisArayuzu.Width = FormWidth - WidthFark;
    groupBoxVeriGirisArayuzu.Height = FormHeight - HeightFark; }

```

EK-6 (Devam) Kullanılan Kodlar

```

private void KapatTSMI_Click(object sender, EventArgs e)
{
    foreach (Form childForm in MdiChildren)
        childForm.Close();
}

private void ÇıkışTSMI_Click(object sender, EventArgs e)
{
    System.Windows.Forms.Application.Exit();
}

private void HakkındaTSMI_Click(object sender, EventArgs e)
{
    RamGec hakkında = new RamGec();
    hakkında.Show();
}

private void TakvimTSB_Click(object sender, EventArgs e)
{
    Takvim takvimformu = new Takvim();
    takvimformu.MdiParent = this.MdiParent;
    takvimformu.ShowDialog();
}

private void AlgoritmaTSB_Click(object sender, EventArgs e)
{
    Algoritma algoritmaformu = new Algoritma();
    algoritmaformu.MdiParent = this.MdiParent;
    algoritmaformu.ShowDialog();
}

private void menüÇubuğuTSMI_Click(object sender, EventArgs e)
{
    toolStrip1.Visible = menüÇubuğuTSMI.Checked;
}

private void durumÇubuğuTSMI_Click(object sender, EventArgs e)
{
    statusStrip1.Visible = durumÇubuğuTSMI.Checked;
}

private void BasamaklaTSMI_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.Cascade);
}

private void DikeyDoseTSMI_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileVertical);
}

private void YatayDoseTSMI_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.TileHorizontal);
}

private void TümünüKapatTSMI_Click(object sender, EventArgs e)
{
    foreach (Form childForm in MdiChildren)
        childForm.Close();
}

private void SimgeDurumunaTSMI_Click(object sender, EventArgs e)
{
    LayoutMdi(MdiLayout.ArrangeIcons);
}

private void YardımTSMI_Click(object sender, EventArgs e)
{
    System.Windows.Forms.SendKeys.Send("{F1}");
}

//Analiz
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Linq;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace VolkanSonerOzsoy
{
    public partial class AnalizFormu : Form
    {
        public AnaPencere anaform;
        public AnalizFormu()
        {
            InitializeComponent();
        }

        public static int Boyut = AnaPencere.Boyut;
        public static int KüçükBoyut = 10;
        public static double[,] girdi = new double[Boyut, Boyut];
    }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

public static double[,] çıktı = new double[Boyut, Boyut];
public int DegiskenSayısı;
public int KararVericiSayısı;
public static int Periyot;
public int Genişlik;
public int[] KontrolEdilemeyenlerinSıraNumarası = new int[KüçükBoyut];
public int[] KontrolEdilemeyenGirdiSıraNo = new int[KüçükBoyut];
public int[] KontrolEdilemeyenÇıktıSıraNo = new int[KüçükBoyut];
public string[] KVBİsimleri = new string[KüçükBoyut];
public string[] Periyotİsimleri = new string[KüçükBoyut];
private void btnAnalizİptal_Click(object sender, EventArgs e)
{ this.Close(); }

private void AnalizFormu_Load(object sender, EventArgs e)
{
    DegiskenSayısı = -1;
    for (int i = 0; i < anaform.dGVVeriGirisArayüzü.Columns.Count; i++)
    {
        if (anaform.dGVVeriGirisArayüzü[i, 0].Value == null)
            break;
        DegiskenSayısı++;
    }
    KararVericiSayısı = 0;
    for (int i = 1; i < anaform.dGVVeriGirisArayüzü.Rows.Count; i++)
    {
        if (anaform.dGVVeriGirisArayüzü[0, i].Value == null)
            break;
        KararVericiSayısı++;
    }
    Periyot = 0;
    for (int i = 0; i <= DegiskenSayısı; i++)
    if (anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'K' ||
        anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'k')
        Periyot++;
    KVBİsimleri = new string[KararVericiSayısı];
    for (int i = 1; i <= KararVericiSayısı; i++)
        KVBİsimleri[i - 1] = anaform.dGVVeriGirisArayüzü[0, i].Value.ToString();
    Periyotİsimleri = new string[Periyot];
    int temp = 0;
    for (int i = 0; i <= DegiskenSayısı; i++)
    {
        if (anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'K' || anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0]
            == 'k')
        {
            Periyotİsimleri[temp] = anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString();
            temp++;
        }
    }
    numKVB.Value = KararVericiSayısı;
    numKVB.Enabled = false;
    numPeriyot.Value = Periyot;
    if (Periyot > 1)
    {
        DegiskenSayısı = (DegiskenSayısı + 1) / Periyot - 1;
        numGenişlik.Maximum = Periyot - 1;
        numGenişlik.Minimum = 2;
        numGenişlik.Value = Periyot - 1;
        numPeriyot.Enabled = false;
        tabControl1.SelectedIndex = 1;
    } else
    { numGenişlik.Maximum = 0;
        numPeriyot.Maximum = Periyot;
        numGenişlik.Enabled = false;
        numPeriyot.Enabled = false; }
    Genişlik = (int)numGenişlik.Value;
    KontrolEdilemeyenlerinSıraNumarası = new int[DegiskenSayısı];

```

EK-6 (Devam) Kullanılan Kodlar

```

dGVDeğiskenÖzellikleri.RowCount = DeğiskenSayısı;
dGVEtkinliğinOran.RowCount = DeğiskenSayısı;
dGVDeğiskenÖzellikleri.RowHeadersWidth = (int)(40 + 5 * (Math.Floor(Math.Log10(DeğiskenSayısı)) + 1));
dGVEtkinliğinOran.RowHeadersWidth = (int)(40 + 5 * (Math.Floor(Math.Log10(DeğiskenSayısı)) + 1));
anaform.SatırNumarasıVermeFonksiyonu(dGVDeğiskenÖzellikleri);
anaform.SatırNumarasıVermeFonksiyonu(dGVEtkinliğinOran);
for (int i = 0; i < DeğiskenSayısı; i++)
{
    dGVDeğiskenÖzellikleri[0, i].Value = anaform.dGVVeriGirisArayüzü[i + 1, 0].Value.ToString();
    dGVEtkinliğinOran[0, i].Value = anaform.dGVVeriGirisArayüzü[i + 1, 0].Value.ToString();
}
}
private void btnAnalizTamam_Click(object sender, EventArgs e)
{
    this.Close();
    double[] Matrisim = new double[DeğiskenSayısı * KararVericiSayısı + 1];
    for (int i = 1; i <= KararVericiSayısı; i++)
    for (int j = 1; j <= DeğiskenSayısı; j++)
        Matrisim[j + (i - 1) * DeğiskenSayısı] = Convert.ToDouble(anaform.dGVVeriGirisArayüzü[j, i].Value);
    anaform.M = Matrisim.Max() * 40;
    anaform.DataGridleriTemizle(anaform.dGVEtkinlikSkoru);
    anaform.DataGridleriTemizle(anaform.dGVWindow);
    anaform.tabControl1.SelectedIndex = 1;
    switch (tabControl1.SelectedIndex)
    {
        case 0:
        {
            int[] GirdiSıraNo = new int[DeğiskenSayısı];
            int[] ÇıktıSıraNo = new int[DeğiskenSayısı];
            int GirdiSayısı = 0;
            int ÇıktıSayısı = 0;
            for (int i = 0; i <= DeğiskenSayısı; i++)
            {
                if (anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'G' ||
                    anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'g')
                {
                    GirdiSıraNo[GirdiSayısı] = i;
                    GirdiSayısı++;
                }
                else if (anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'Ç' ||
                    anaform.dGVVeriGirisArayüzü[i, 0].Value.ToString()[0] == 'ç')
                {
                    ÇıktıSıraNo[ÇıktıSayısı] = i;
                    ÇıktıSayısı++;
                }
            }
            KontrolEdilemeyenGirdiSıraNo = new int[GirdiSayısı];
            KontrolEdilemeyenÇıktıSıraNo = new int[ÇıktıSayısı];
            for (int i = 0; i < KararVericiSayısı; i++)
            {
                for (int j = 0; j < GirdiSayısı; j++)
                    girdi[i, j] = Convert.ToDouble(anaform.dGVVeriGirisArayüzü[GirdiSıraNo[j], i + 1].Value);
                for (int j = 0; j < ÇıktıSayısı; j++)
                    çıktı[i, j] = Convert.ToDouble(anaform.dGVVeriGirisArayüzü[ÇıktıSıraNo[j], i + 1].Value);
            }
            for (int j = 0; j < GirdiSayısı; j++)
            if (KontrolEdilemeyenlerinSıraNumarası[GirdiSıraNo[j] - 1] == 1)
                KontrolEdilemeyenGirdiSıraNo[j] = 1;
            for (int j = 0; j < ÇıktıSayısı; j++)
            if (KontrolEdilemeyenlerinSıraNumarası[ÇıktıSıraNo[j] - 1] == 1)
                KontrolEdilemeyenÇıktıSıraNo[j] = 1;
            if (KararVericiSayısı > 0 && GirdiSayısı > 0 && ÇıktıSayısı > 0 && comboBoxVZAiçin.SelectedIndex + 1 > 0)
                anaform.BütünAnalizler(KararVericiSayısı, GirdiSayısı, ÇıktıSayısı, comboBoxVZAiçin.SelectedIndex,
                    girdi, çıktı, KontrolEdilemeyenGirdiSıraNo, KontrolEdilemeyenÇıktıSıraNo);
            else
                MessageBox.Show("Yanlış Değer Girdiniz!", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
        break;
    }
}

```

EK-6 (Devam) Kullanılan Kodlar

```

case 1:
{
if (Periyot <= 1)
MessageBox.Show("Verileriniz window analizine uygun olmayabilir." + "\n" + "Karar verme birimlerini belirtmemiş olabilirsiniz." + "\n" + "\n" + "Karar verme birimlerini doğru belirtmek için, Veri Zarflama Analizi için karar verme birimlerinin olduğu sütunun ilk satırında 'KVB', Window Analizi için ise 'KVB-Tarih' yazmalıdır...", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
else
{
Periyot = (int)numPeriyot.Value;
Genişlik = (int)numGenişlik.Value;
KararVericiSayısı = (int)numKVB.Value;
int HerKVBneAitSatır = Periyot - Genişlik + 1;
anaform.groupBoxWindow.Visible = true;
anaform.groupBoxEtkinlikSkoru.Visible = false;
anaform.WindowDGValanıAyrırma(Periyot, Genişlik, KararVericiSayısı, KVBİsimleri, Periyotİsimleri);
for (int K = 0; K <= Periyot - Genişlik; K++)
{
object[,] BirleştirilmişMatris = new object[DegiskenSayısı + 1, KararVericiSayısı * Genişlik + 1];
for (int i = 0; i <= DegiskenSayısı; i++)
BirleştirilmişMatris[i, 0] = (anaform.dGVVeriGirisArayüzü[i, 0].Value);
for (int g = 0; g < Genişlik; g++)
for (int i = 0; i <= DegiskenSayısı; i++)
for (int j = 1; j <= KararVericiSayısı; j++)
BirleştirilmişMatris[i, j + g * KararVericiSayısı] = (anaform.dGVVeriGirisArayüzü[K * (DegiskenSayısı + 1) + i + g * (DegiskenSayısı + 1), j].Value);
int[] GirdiSıraNo = new int[DegiskenSayısı];
int[] ÇıktıSıraNo = new int[DegiskenSayısı];
int GirdiSayısı = 0;
int ÇıktıSayısı = 0;
for (int i = 0; i <= DegiskenSayısı; i++)
{
if (BirleştirilmişMatris[i, 0].ToString()[0] == 'G' || BirleştirilmişMatris[i, 0].ToString()[0] == 'g')
{
GirdiSıraNo[GirdiSayısı] = i;
GirdiSayısı++;
}
else if (BirleştirilmişMatris[i, 0].ToString()[0] == 'Ç' || BirleştirilmişMatris[i, 0].ToString()[0] == 'ç')
{
ÇıktıSıraNo[ÇıktıSayısı] = i;
ÇıktıSayısı++;
}
}
girdi = new double[GirdiSayısı, KararVericiSayısı * Genişlik + 1];
çıktı = new double[ÇıktıSayısı, KararVericiSayısı * Genişlik + 1];
for (int i = 0; i < KararVericiSayısı * Genişlik; i++)
{
for (int j = 0; j < GirdiSayısı; j++)
girdi[j,i] = Convert.ToDouble(BirleştirilmişMatris[GirdiSıraNo[j], i + 1]);
for (int j = 0; j < ÇıktıSayısı; j++)
çıktı[j,i] = Convert.ToDouble(BirleştirilmişMatris[ÇıktıSıraNo[j], i + 1]);
}
if (KararVericiSayısı > 0 && GirdiSayısı > 0 && ÇıktıSayısı > 0 && comboBoxWindow için.SelectedIndex + 1 > 0)
switch (comboBoxWindow için.SelectedIndex)
{
case 0: anaform.CCR_GIRDI(DegiskenSayısı, KararVericiSayısı * Genişlik, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
case 1: anaform.CCR_CIKTI(DegiskenSayısı, KararVericiSayısı * Genişlik, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
case 2: anaform.BCC_GIRDI(DegiskenSayısı, KararVericiSayısı * Genişlik, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
case 3: anaform.BCC_CIKTI(DegiskenSayısı, KararVericiSayısı * Genişlik, GirdiSayısı, ÇıktıSayısı, girdi, çıktı); break;
default: MessageBox.Show("Girilen model bulunamadı!"); break;
}
}
else
MessageBox.Show("Yanlış Değer Girdiniz!", "Uyarı", MessageBoxButtons.OK, MessageBoxIcon.Warning);
for (int i = 1; i <= Genişlik; i++)
{
int temp = 0;

```

EK-6 (Devam) Kullanılan Kodlar

```

for (int j = 1; j <= KararVericiSayısı; j++)
{
    anaform.dGVWindow[K + i, K + j + temp].Value = Convert.ToString(anaform.dGVEtkinlikSkoru[(i - 1) * KararVericiSayısı
    + j, 1].Value);
    temp += Periyot - Genişlik + 2; } }
for (int i = 1; i <= KararVericiSayısı; i++)
{
    for (int p = 0; p < Periyot; p++)
    {
        double ortalama = 0.0;
        double toplam = 0.0;
        int boşhücre sayısı = 0;
        for (int j = 1; j <= HerKVBneAitSadır; j++)
        {
            if ((anaform.dGVWindow[p + 1, j + (i - 1) * (HerKVBneAitSadır + 2)].Value) == null)
                boşhücre sayısı++;
            toplam += Convert.ToDouble(anaform.dGVWindow[p + 1, j + (i - 1) * (HerKVBneAitSadır + 2)].Value);
            ortalama = toplam / (HerKVBneAitSadır - boşhücre sayısı);
        }
        anaform.dGVWindow[p + 1, i * (HerKVBneAitSadır + 2) - 1].Value = ortalama.ToString();
    }
}
for (int i = 0; i < KararVericiSayısı * (HerKVBneAitSadır + 2); i++)
{
    int boşhücre sayısı = 0;
    double toplam = 0;
    double ortalama = 0;
    for (int p = 0; p < Periyot; p++)
    {
        if ((anaform.dGVWindow[p + 1, i + 1].Value) == null)
            boşhücre sayısı++;
        toplam += Convert.ToDouble(anaform.dGVWindow[p + 1, i + 1].Value);
        ortalama = toplam / (Periyot - boşhücre sayısı);
    }
    if ((Periyot - boşhücre sayısı) != 0)
        anaform.dGVWindow[Periyot + 1, i + 1].Value = ortalama.ToString();
    else anaform.dGVWindow[Periyot + 1, i + 1].Value = null;
}
}
break;
}
}
private void dGVDegiskenÖzellikleri_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex == -1)
        return;
    dGVDegiskenÖzellikleri.CommitEdit(DataGridViewDataErrorContexts.Commit);
    bool seçilimi;
    seçilimi = (bool)dGVDegiskenÖzellikleri.Rows[e.RowIndex].Cells[e.ColumnIndex].Value;
    if (seçilimi == true)
        KontrolEdilemeyenlerinSıraNumarası[e.RowIndex] = 1;
    else
        KontrolEdilemeyenlerinSıraNumarası[e.RowIndex] = 0;
}
}
}

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖZSOY, Volkan Soner
 Uyruğu : T.C.
 Doğum yılı ve yeri : 1988 Aksaray
 Medeni hali : Bekâr
 Telefon : 0 (544) 669 38 50
 e-mail : volkansoner1@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Gazi Üniversitesi - İstatistik A.B.D.	2012
Lisans	Karadeniz Teknik Üniversitesi - İstatistik ve Bilgisayar Bilimleri Bölümü	2010
Lise	Aksaray Lisesi - Fen Bilimleri Bölümü	2005

Yabancı Dil

İngilizce

Yayınlar

- Özsoy, V.S., Alp. İ., Kınacı, H., “Performans Analizi İçin Yeni Karar Destek Sistemi”, *5. Mühendislik ve Teknoloji Sempozyumu*, Ankara, 114-117 (2012).
- Özsoy, V.S., Kurt, Y., “Görüntü İşlemede Sıklık Uygulamaları”, *7. İstatistik Öğrenci Kolokyum*, Afyonkarahisar, (2010).

İlgi Alanları

Bilgisayar teknolojileri, Kriptoloji, Yazılım