

**AKILLI ÇOK ÖLÇÜTLÜ YASAL TAKİP AVUKATLIK OFİSİ
PERFORMANS YÖNETİMİ SİSTEMİ**

YÜKSEK LİSANS TEZİ

Erdinç URUÇ

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Doç.Dr. Sezi ÇEVİK ONAR

HAZİRAN, 2020

**AKILLI ÇOK ÖLÇÜTLÜ YASAL TAKİP AVUKATLIK OFİSİ
PERFORMANS YÖNETİMİ SİSTEMİ**

YÜKSEK LİSANS TEZİ

Erdoğan URUÇ

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Doç.Dr. Sezi ÇEVİK ONAR

HAZİRAN, 2020

ÖNSÖZ

Ödenmeyen borçların hızla arttığı günümüzde borç takibini yapmak önemli bir konum almaktadır. Ödenmeyen borçlar yasal süreçlerle tahsil edilmeye çalışıldığında bu süreci takip etmesi için dış kaynak olarak avukatlık ofisleri ile çalışılmaktadır. Çalışılan ofis sayısı fazla olduğundan bu ofislerin performans ölçümleri ve karşılaştırmaları alacak takibi yapılacak dosyaların dağıtımında ve asıl amaç olan borç toplama miktarının en büyüklenmesinde önemli rol oynamaktadır.

Firmaların performans ölçümünde ve bu ölçüm yapılırken kullandıkları kriterlerin karşılaştırılıp ağırlıklarının belirlenmesinde yaşadığı problemlerin çözümü için sözel olarak belirttikleri kriterleri sayısallaştırarak performans hesaplaması ve sıralamasını yapmak amacıyla bu çalışma gerçekleştirilmiştir.

Yüksek lisans eğitimi ve tez geliştirilmesi sürecinde bana yol gösteren ve beni destekleyen tez danışmanım sayın Doç.Dr.Sezi ÇEVİK ONAR'a, bu süreçte bilgi ve tecrübeleri ile her zaman destek olan sayın Doç.Dr.Başar ÖZTAYŞI ve sayın Prof.Dr.Cengiz KAHRAMAN'a, yüksek lisans eğitimi konusunda beni teşvik eden ve süreç boyunca destekleyen sayın Doç.Dr.Nurdan MEMİŞOĞLU APAYDIN'a, eğitim hayatımın her aşamasında maddi ve manevi desteklerini hiçbir zaman esirgemeyen sevgili aileme ve her zaman yanımda olan eşime teşekkür ederim.

Haziran 2020

Erdoğan URUÇ

Bilgisayar Mühendisi

İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ.....	iv
İÇİNDEKİLER.....	iv
KISALTMALAR.....	viii
ŞEKİL LİSTESİ.....	x
TABLO LİSTESİ	xii
ÖZET	1
SUMMARY.....	3
1.GİRİŞ.....	5
2.AVUKATLIK OFİSLERİNİN PERFORMANSI	9
3.KARAR VERME VE TEMEL KAVRAMLAR	21
3.1 Karar Verme	21
3.2 Karar Verme Sürecinin Aşamaları	22
3.3 Karar Modelinin Öğeleri	23
3.3.1 Karara Konu Problem	23
3.3.2 Karar Verme Sürecinin Katılımcıları.....	24
3.3.3 Seçenekler	25
3.3.4 Amaçlar,kriterler ve ölçütler	25
3.3.5 Yöntemler	27
3.4 Karar Modelleri.....	27
3.4.1 Belirlik Halinde Karar Verme.....	28
3.4.2 Belirsizlik Halinde Karar Verme	28
3.4.3 Risk Halinde Karar Verme.....	28
3.4.4 Kısmi Bilgi Halinde Karar Verme	29
3.4.5 Rekabet Halinde Karar Verme.....	29
4. ÇOK ÖLÇÜTLÜ KARAR VERME	31
4.1 Çok Ölçütlü Karar Verme Yöntemlerinin Sınıflandırılması	32
4.2 Çok Ölçütlü Karar Verme Yöntemleri	34
4.2.1 Baskınlık Yöntemi	34
4.2.2 Birleştiren Yöntem.....	34
4.2.3 Ayıran Yöntem.....	35
4.2.4 Ardışık Sırasal Yöntem.....	35
4.2.5 Özelliklerine Göre Eleme Yöntemi	36

4.2.6 Maximin Yöntemi	37
4.2.7 Maximax Yöntemi	38
4.2.8 Basit Toplamlı Ağırlıklandırma Yöntemi	38
4.2.9 Ağırlıklı Çarpım Yöntemi	39
4.2.10 TOPSIS	40
4.2.11 ELECTRE	42
4.2.12 PROMETHEE	42
4.2.13 AHP (Analitik Hiyerarşi Prosesi)	44
5.BULANIK MANTIK	45
5.1 Bulanık Küme Teorisi	48
5.1.1 Bulanık Kümelerin Özellikleri	48
5.1.2 Bulanık Üyelik Fonksiyonları	51
5.2 Bulanık Küme İşlemleri	54
5.3 Üyelik Fonksiyonlarını Değiştiren Bulanık Küme İşlemleri	55
5.3.1 Kuvvet	55
5.3.2 Derişme	56
5.3.3 Genişleme	56
5.3.4 Yoğunlaşma	57
5.4 Bulanık Sayılarda Aritmetik İşlemler	57
5.4.1 Bulanık Sayılarda Yaklaşık İşlemler	57
5.4.2 Bulanık Sayılarda Güven Aralıkları İle Aritmetik İşlemler	58
5.4.3 Bulanık Sayılarda Üyelik Fonksiyonları İle Yapılan Aritmetik İşlemler ..	60
5.5 Bulanık Sayılarda Sıralama İşlemleri	61
5.5.1 Liou ve Wang Yöntemi	62
5.5.2 Abdel-Kader ve Dugdale'ın Yöntemi	62
6. ANALİTİK HİYERARŞİ PROSESİ (AHP)	63
6.1 Hiyerarşik Yapı	64
6.2 Ölçme ve Yapı Oluşturma	66
6.3 İkili Karşılaştırmalar Matrisi ve Ağırlık Kümesi	69
6.4 Analitik Hiyerarşi Prosesinde Kullanılan Ölçek	71
6.5 Özvektörlerin Bulunması	72
6.5.1 En Basit ve Sapmalı Yöntem	72
6.5.2 Daha İyi Bir Yöntem	72
6.5.3 İyi Bir Yöntem	72
6.5.4 En İyi Yöntem	72

6.6 Tutarlılık	72
7.BULANIK AHP	75
7.1 Van Laarhoven ve Pedrycz Yaklaşımı	76
7.2 Buckley Yaklaşımı	79
7.3 Chang'in Mertebe Analizi Yaklaşımı	81
7.4 Kararsız(Hesitant) Bulanık AHP	83
7.5 Küresel(Spherical) Bulanık AHP	87
7.6 Sezgisel Bulanık AHP	90
7.7 Diğer Bulanık AHP Yöntemleri	92
8. YAZILIM UYGULAMASI	95
8.1 Uygulama Tasarımı	96
8.2 Uygulama.....	97
8.2.1 Kriterlerin Kaydedilmesi	99
8.2.2 AHP İle Kriter Ağırlıklarının Hesaplanması	100
8.2.3 BAHP İle Kriter Ağırlıklarının Hesaplanması.....	102
8.2.4 Performans Notu Hesaplama	103
9.AHP ve BAHP KULLANILARAK BORÇ TAKİBİ YAPAN AVUKATLIK OFİSLERİNİN PERFORMANS KARŞILAŞTIRMASI	109
9.1 Hiyerarşik Yapı.....	109
9.2 Uygulamanın Genel İşleyişi	110
9.3 AHP Uygulaması.....	111
9.4 BAHP Uygulaması	120
9.5 AHP ve BAHP Değerlerinin Karşılaştırılması	127
SONUÇ VE ÖNERİLER	129
KAYNAKLAR	133
EKLER	139
EK A. Analitik Hiyerarşi Prosesi Kullanılarak Yapılan Avukatlık Ofisi Performans Hesaplamaları	139
EK B. Bulanık Analitik Hiyerarşi Prosesi Kullanılarak Yapılan Avukatlık Ofisi Performans Hesaplamaları.....	154
EK C. Yazılım Uygulaması Kodları.....	169
Ek D. AHP İle Performans Hesaplama Kodları	201
Ek D. BAHP İle Performans Hesaplama Kodları.....	211
ÖZGEÇMİŞ	223



KISALTMALAR

KV	: Karar Verici
ÇÖKV	: Çok Ölçütlü Karar Verme
ÇKKV	: Çok Kriterli Karar Verme
ÇAKV	: Çok Amaçlı Karar Verme
MODM	: Multi Objective Decision Making
MADM	: Multi Attribute Decision Making
AHP	: Analitik Hiyerarşi Prosesi
ÜBS	: Üçgensel Bulanık Sayı
YBS	: Yamuk Bulanık Sayı
SBS	: S Bulanık Sayı
ÇEBS	: Çan Eğrisi Bulanık Sayı
ABD	: Amerika Birleşik Devletleri
BAHP	: Bulanık Analitik Hiyerarşi Prosesi
KBAHP	: Kararsız Bulanık Analitik Hiyerarşi Prosesi
TOPSIS	: Technique For Order Preference By Similarity To An Ideal Solution
ELECTRE	: ELimination Et Choice Translating REality
PROMETHEE	: Preference Ranking Organization METHod for Enrichment of Evaluations



ŞEKİL LİSTESİ

Sayfa

Şekil 4. 1 Analiz Yöntemlerinin Sınıflandırılması (Lavelle ve Lingett)	33
Şekil 4. 2 Analiz Yöntemlerinin Sınıflandırılması (Kolli ve Diğerleri)	33
Şekil 4. 3 PROMETHEE Fonksiyonları	43
Şekil 5. 1 Bulanık Küme - Destek Kümesi	49
Şekil 5. 2 Bulanık Küme - Geçiş Noktası	49
Şekil 5. 3 Bulanık Küme – Bulanık Teklik	49
Şekil 5. 4 Bulanık Küme - α -Kesim Kümesi	50
Şekil 5. 5 Bulanık Küme – Normalizasyon.....	51
Şekil 5. 6 Bulanık Sayılar – Üçgen Üyelik Fonksiyonu	52
Şekil 5. 7 Bulanık Sayılar – Yamuk Üyelik Fonksiyonu	53
Şekil 5. 8 Bulanık Sayılar – S Üyelik Fonksiyonu	53
Şekil 5. 9 Bulanık Sayılar – Çan Eğrisi Üyelik Fonksiyonu.....	54
Şekil 5. 10 Bulanık Kümelerde Birleşim İşlemi	54
Şekil 5. 11 Bulanık Kümelerde Birleşim İşlemi	55
Şekil 5. 12 Bulanık Kümelerde Birleşim İşlemi	55
Şekil 5. 13 Bulanık Kümelerde Derişme İşlemi	56
Şekil 5. 14 Bulanık Kümelerde Genişleme İşlemi	56
Şekil 5. 15 Bulanık Kümelerde Yoğunlaşma İşlemi.....	57
Şekil 5. 16 Üçgensel Bulanık Sayılarda Güven Aralığı.....	59
Şekil 5. 17 Yamuk Bulanık Sayılarda Güven Aralığı.....	59
Şekil 6. 1 Basit bir AHP Yapısı	66
Şekil 7. 1 Çoklu karar vericilerin bulanık karşılık matrisi	76
Şekil 7. 2 M1 ve M2 arasındaki kesişim noktası	83
Şekil 8. 1 Yazılım Uygulamasının Modeli.....	96
Şekil 8. 2 Yazılım Uygulaması Akış Diyagramı.....	97
Şekil 8. 3 Yazılım Uygulaması Kullanım Durumu Diyagramı.....	98
Şekil 9. 1 Hiyerarşik Yapı.....	110
Şekil 9. 2 AHP Uygulamasında Kullanılan PROMETHEE Fonksiyonları	117



TABLO LİSTESİ

Sayfa

Tablo 4. 1 TOPSIS Karar Matrisi [68]	40
Tablo 6. 1 AHP Ölçeği [31].....	71
Tablo 6. 2 Tesadüfilik Göstergesi[66]	73
Tablo 7. 1 Bulanık Önem Dereceleri[54]	76
Tablo 7. 2 BAHP Yöntemlerinin Kıyaslanması [56].....	77
Tablo 7. 3 Varlıkların İkili Karşılaştırma Matrisinde Yorumlanması[59]	80
Tablo 7. 4 Kararsız Bulanık AHP Dönüşüm Tablosu [60].....	85
Tablo 7. 5 Küresel Bulanık AHP Dönüşüm Tablosu [62].....	88
Tablo 7. 6 Aralık Değerli Sezgisel Bulanık AHP Dönüşüm Tablosu [64].....	90
Tablo 9. 1 AHP Uygulaması İçin Parametre – Fonksiyon Eşleşme Tablosu	111
Tablo 9. 2 AHP Uygulaması İçin KV Sözel Değerlendirmeleri Tablosu	112
Tablo 9. 3 AHP Uygulaması İçin KV Sözel Değerlendirmeleri Tablosu – (Eksik Alanların Doldurulmuş Hali)	112
Tablo 9. 4 AHP Sözel-Sayısal Değerler Referans Dönüşüm Tablosu	113
Tablo 9. 5 AHP Uygulaması İçin KV Değerlendirmeleri – Sayısal.....	113
Tablo 9. 6 AHP Uygulaması – Satır Toplamları	114
Tablo 9. 7 AHP Uygulaması – Normalize Edilmiş Tablo	114
Tablo 9. 8 AHP Uygulaması – Satır Ortalamaları	115
Tablo 9. 9 AHP Uygulaması – Kriter Ağırlıkları	115
Tablo 9. 10 Rassal Tutarlılık İndeksi (RI)	116
Tablo 9. 11 AHP Uygulaması Hesaplanan Değer	118
Tablo 9. 12 AHP - Ağırlıklı Performans Puanları Tablosu	119
Tablo 9. 13 AHP - Sıralı Ağırlıklı Performans Puanları Tablosu	120
Tablo 9. 14 AHP Uygulaması İçin Parametre – Fonksiyon Eşleşmesi	120
Tablo 9. 15 BAHP Uygulaması İçin KV Değerlendirmeleri – Sözel.....	121
Tablo 9. 16 BAHP Sözel-Sayısal Değerler Referans Dönüşüm Tablosu.....	121
Tablo 9. 17 AHP Uygulaması İçin KV Değerlendirmeleri – Sayısal.....	122
Tablo 9. 18 BAHP Geometrik Ortalama Matrisi.....	123
Tablo 9. 19 BAHP Sütun Normalizasyon Katsayıları	123
Tablo 9. 20 BAHP Sütun Normalize Tablo.....	124
Tablo 9. 21 BAHP Satır Normalizasyon Değerleri	124
Tablo 9. 22 BAHP – Kriter ağırlıkları	125
Tablo 9. 23 BAHP Uygulaması Hesaplanan Değer	125
Tablo 9. 24 BAHP - Ağırlıklı Performans Puanları Tablosu	126
Tablo 9. 25 BAHP - Sıralı Ağırlıklı Performans Puanları Tablosu	127
Tablo 9. 26 AHP ve BAHP Ağırlık Karşılaştırmaları	127
Tablo 9. 27 AHP ve BAHP Kullanılarak Performans Puanı Karşılaştırmaları	128

AKILLI ÇOK ÖLÇÜTLÜ YASAL TAKİP AVUKATLIK OFİSİ PERFORMANS YÖNETİMİ SİSTEMİ

ÖZET

Yapılan tez kapsamında, zamanında ödenmeyen borçlar için firmaların birlikte çalıştığı yasal takip süreçlerini yürüten avukatlık ofislerinin performanslarının ölçülmesi için bir model geliştirilmiştir. Model geliştirilirken analitik hiyerarşi süreci ve bulanık analitik hiyerarşi temel alınarak iki farklı yöntem ile hesaplama yapılmıştır. Hesaplamaları yapmak üzere Java dili kullanılarak bir yazılım uygulaması geliştirilmiştir. Yazılım uygulaması, hem AHP(Analitik Hiyerarşi Prosesi) hem de BAHP(Bulanık Analitik Hiyerarşi Prosesi) için firmaların performanslarını hesaplamakta ve hesaplama sonucunda avukatlık ofislerini performans puanına göre sıralamaktadır.

Günümüzde ödenmeyen borçların miktarı gün geçtikçe artmaktadır. Firmalar müşterilerine ürün ve hizmetlerini sunmakta ancak her zaman bunların karşılığında ödemelerini zamanında alamamaktadır. Bu durum firmaların nakit akışlarını, cirolarını, kredi puanlarını ve hatta marka değerlerini ciddi anlamda etkilemektedir. Bu nedenle firmalar ödenmeyen borçların tahsilatı konusuna büyük önem vermektedirler.

Firmalar alacaklarını son ödeme tarihinden sonraki 90 gün içerisinde tahsil edemediklerinde yasal takip sürecini başlatırlar. Bu süreçte borçlar mahkeme aracılığıyla tahsil edilmeye çalışılır. Borçlu sayısının fazlalığı, farklı bölge ve şehirlerde yaşamaları gibi nedenlerle firmalar yasal süreci takip etmeleri için avukatlar veya avukatlık ofisleri ile çalışırlar ve hizmetleri karşılığı bu kişi veya ofislere ödeme yaparlar.

Firmaların amacı ödenmeyen alacakların en yüksek miktarda tahsil edilmesidir. Bu nedenle yasal süreci takip etmesi için anlaşılan avukatlık ofislerinin performansı alacaklı firmalar için büyük önem arz etmektedir. Avukatlık ofislerinin performansına göre birlikte çalışmaya devam edip etmeme veya ne kadar büyük bir borçlu kitlesi için o ofisle çalışılacağı kararları verilmektedir. Bu nedenle birlikte çalıştıkları avukatlık ofislerinin performanslarının doğru şekilde ölçülmesi firmalar için önem arz etmektedir.

Avukatlık ofislerinin performansının belirlenmesi için; kullanımı kolay, esnek, değerlendirme kriterlerinin önemlerinin sözel olarak ifade edilebildiği ve bu sözel ifadelerin sayısallaştırılarak hesaplamalara dahil edilebildiği ve otomatize edilmiş bir model yoktur. Bu nedenlerle; avukatlık ofislerinin performansının değerlendirilmesinde kullanılmak üzere, değerlendirme kriterlerinin önem derecelerinin birbirleri ile kıyaslanabildiği ve bu kıyaslamanın sözel değerlerle yapılabildiği, analitik hiyerarşi prosesi ve bulanık analitik hiyerarşi prosesi kullanarak kriter ağırlıklarının hesaplanabildiği ve tüm süreci otomatize eden bir bilgisayar uygulaması içeren bir model geliştirilmiştir.

Yapılan literatür araştırmaları ve ortak çalışılan bir firmanın ihtiyaçları da göz önüne alınarak Analitik hiyerarşi süreci ve Bulanık analitik hiyerarşi süreci temelli bir sistem geliştirilmesi uygun görülmüştür.

Geliştirilen modelde, kriterler ve kriterlerin birbirleri ile kıyaslanmaları değiştirilebilir olarak tasarlanmıştır. Bu sayede, geliştirilen sistem değişen ihtiyaçlar kapsamında avukatlık ofislerinin performans değerlendirmesi için kullanılan kriterler değiştirilebileceği gibi değerler arasındaki önemler de değiştirilerek değişen ihtiyaçlara cevap verebilecek şekilde tasarlanmıştır. Geliştirilen yazılım ile kriterler, kriterlerin birbirleri ile sözlü kıyaslamaları ve avukatlık ofislerinin değerlendirme kriterleri için gerçekleşen puanları sisteme girdi olarak verilerek analitik hiyerarşi süreci veya bulanık analitik hiyerarşi süreci kullanılarak performans puanları hesaplanabilir.

Tez çalışması kapsamında yapılan uygulama bünyesinde, telekomünikasyon sektöründeki bir firma ile ortak çalışma yapılarak avukatlık ofislerinin değerlendirilmesinde aşağıdaki beş kriterlerin kullanılmasına karar verilmiştir:

1. Asıl alacak miktarı
2. Denetim performansı
3. Müşteri memnuniyeti
4. Dosya zorluğu
5. Genel değerlendirme puanı

Bu beş kriter ile sistem simüle edilmiş ve firmanın uzmanları tarafından kriterlerin sözel olarak karşılaştırılması sonucu oluşan önem ağırlıkları kullanılarak 15 adet avukatlık ofisi için performans değerlendirme yapılmıştır.

SMART, MULTI CRITERIA PERFORMANCE MANAGEMENT SYSTEM FOR LEGAL PROCEEDING LAW OFFICES

SUMMARY

Within the scope of the thesis, a model has been developed to measure the performances of the lawyers' offices that carry out the legal proceeding processes that the companies work together for the debts that are not paid on time. While developing the model, two different methods were calculated based on the analytical hierarchy process and the fuzzy analytical hierarchy. A software application has been developed using the Java language to make calculations. The software application calculates firms' performances for both AHP (Analytical Hierarchy Process) and FAHP (Fuzzy Analytical Hierarchy Process) and sorts the law firms according to their performance score.

Today, the amount of unpaid debts is increasing day by day. Companies offer their products and services to their customers, but they are not always able to receive their payments on time. This seriously affects firms' cash flows, turnovers, credit ratings and even brand values. Therefore, companies attach great importance to the collection of unpaid debts.

When companies cannot collect their receivables within 90 days after the last payment date, they start the legal processing process. In this process, debts are tried to be collected through the court. Firms work with lawyers or law offices to follow the legal process due to the high number of borrowers, living in different regions and cities, and pay for these services.

The purpose of the firms is to collect the unpaid receivables at the highest amount. Therefore, the performance of the lawyers' offices, which are agreed to follow the legal process, has a great importance for creditor companies. Depending on the performance of lawyers' offices, decisions are made whether to continue working together or not, working with how large group of borrowers. For this reason, it is important for companies to measure the performance of the lawyers' offices they work with.

To determine the performance of the law firms; There is no automated model that is easy to use, flexible, the importance of evaluation criteria can be expressed verbally and these verbal expressions can be digitized and included in calculations. For these reasons; In order to evaluate the performance of law firms, a model has been developed that includes a computer application that automates the whole process, where the severity of the evaluation criteria can be compared with each other and this comparison can be made with verbal values, the criteria weights can be calculated using the analytical hierarchy process and the fuzzy analytical hierarchy process.

Considering the literature researches and the needs of a collaborated company, it was deemed appropriate to develop a system based on Analytical hierarchy process and Fuzzy analytical hierarchy process.

In the developed model, the criteria and the comparisons of the criteria are designed interchangeably. In this way, the developed system was designed to meet the changing needs by changing the importance between the values as well as changing the criteria used for performance evaluation of the lawyers' offices within the scope of changing needs. Performance scores can be calculated by using analytical hierarchy process or fuzzy analytical hierarchy process with the developed software and criteria, verbal comparisons of the criteria and the scores realized for the evaluation criteria of the law offices as input.

Within the scope of the application made within the scope of the thesis, it was decided to use the following five criteria in the evaluation of the lawyers' offices by collaborating with a company in the telecommunication sector.

1. The amount of the original debt
2. Audit performance
3. Customer satisfaction
4. File difficulty
5. Overall assessment score

The system was simulated with these five criteria and performance evaluations were carried out for 15 lawyers' offices using the importance weights of the company by verbal comparison of the criteria.

1.GİRİŞ

Ödenmeyen borçların miktarı her geçen gün artmaktadır. Firmalar müşterilerine ürün ve hizmetlerini sunmakta ancak her zaman bunların karşılığında ödemelerini ödenmesi gereken zamanda alamamaktadır. Bu durum firmalar üzerinde çeşitli olumsuz etkiler yaratmakta olduğundan firmalar ödenmeyen borçların tahsilatı konusuna büyük önem vermektedirler.

Ödenmeyen borçların takip edilmesi ve tahsilat çalışmaları ciddi miktarda kaynak gerektirmektedir. Firmalar bu işlemleri kendi bünyelerinde oluşturdukları birimlere yaptırabileceği gibi dış kaynak kullanarak da süreci yönetebilmektedir. Konu hakkındaki uzmanlıkları, iş büyüklüğüne göre dinamik olarak ayrılan kaynağın değiştirilebilmesi ve maliyet avantajları nedeniyle dış kaynaklar tercih edilebilir olmaktadır.

Dış kaynaklar ile çalışırken, çalışılan kişi veya firmaların performansını takip edebilmek, dış kaynak kullanımından sağlanacak faydanın artırılması için elzemdir. Gösterdikleri performansa göre çalışmaya devam edip etmeme, işbirliği boyunu artırma veya azaltma gibi kararlar alınabilmektedir. Özellikle, aynı veya çok benzer işler için çalışılan dış kaynak firmalar arasında performans karşılaştırılmasının doğru ve etkili bir biçimde yapılması, ilgili konudaki çalışma boyutu ve kaynak dağılımını belirlemede kritik bir rol oynamaktadır.

Tubitak tarafından TEYDEB 1505 programı ile desteklenen “Akıllı Çok Ölçütlü Yasal Takip Avukatlık Ofisi Notlandırma ve Performans Yönetimi Sistemi” projesi kapsamında yapmış olduğumuz bu tez çalışmasında, Türkiye’deki GSM operatörlerinin biriyle ortak çalışılmıştır. Tahsil edemedikleri borçları için yasal takip sürecinde birçok avukatlık ofisi ile dış kaynak olarak çalışan bu firma, takip edilecek tahsilat dosyalarının dağıtımında ofislerin gösterdiği performans sıralamasını bilmek ve dağıtımı ofis performanslarını gözeterek yapmak suretiyle tahsilat miktarını en büyükleme istemiştir.

Tez çalışması kapsamında, bahsi geçen firmanın borç takibi için dış kaynak olarak anlaşılan avukat ofislerinin performans ölçüm ve sıralanmasını sağlamak amacıyla sözel ifadelerle değerlendirmeyi mümkün kılması nedeniyle analitik hiyerarşi prosesi ve sözel ifadeler ile çalışabilmesinin yanı sıra belirsizliği minimize

etmesi nedeniyle bulanık analitik hiyerarşi prosesi kullanarak bir model geliştirilmiştir. Bu modelin uygulanması, Java dili ve güncel teknolojiler kullanılarak bir bilgisayar yazılımı geliştirilmiştir. Yazılımda kriterler ve kriterlerin birbirleri ile kıyaslanmaları değiştirilebilir olarak tasarlanmıştır. Bu sayede geliştirilen sistem değişen ihtiyaçlar kapsamında avukatlık ofislerinin performans değerlendirmesi için kullanılan kriterler değiştirilebileceği gibi değerler arasındaki önem sıralaması da değiştirilerek değişen ihtiyaçlara cevap verebilecek şekilde tasarlanmıştır.

Geliştirilen yazılım ile kriterler, kriterlerin birbirleri ile sözlü kıyaslamaları ve avukatlık ofislerinin değerlendirme kriterleri için gerçekleşen puanları sisteme girdi olarak verilerek analitik hiyerarşi süreci veya bulanık analitik hiyerarşi süreci kullanılarak performans puanları hesaplanabilir. Aynı zamanda bir kez hesaplanan ağırlıklar sistemde tutulduğundan yalnızca ofislerin gerçekleşen puanları sisteme girdi olarak verilerek tekrar tekrar hesaplama yapılabilir. Tüm süreci otomatize eden bu yazılım sayesinde hesaplamalar hızlı bir şekilde yapılırken hesaplama hataları ortadan kaldırılmıştır.

Tez çalışması kapsamında, ilgili firmadan borç tahsilatı ekibi bünyesinde görev alan ve avukatlık ofislerinin performansları konusunda uzman 6 kişi ile birlikte çalışılmıştır. Performans kriterlerinin ve bu kriterlerin önem derecelerinin belirlenmesinde bu uzmanların görüşlerinden yararlanılmıştır. Ayrıca geliştirilen model ve yazılımın kullanılarak yapılan simülasyon için firmanın birlikte çalışmakta olduğu 15 adet avukatlık ofisinin verileri kullanılmıştır.

Çalışmanın 2. bölümünde avukatlık ofisleri, türleri ve uzmanlık alanları; dış kaynak kullanmanın avantajları,dezavantajları ve borçların yasal takibinde dış kaynak olarak avukatlık ofislerini kullanmanın avantaj ve dezavantajları; çeşitli alanlarda yapılan performan ölçüm ve değerlendirmeleri; avukatlık ofislerinin performansları ve dış kaynak olarak avukatlık ofislerinin performans ölçüm ve karşılaştırmaları ile ilgili literatür taramasına yer verilmiştir.

3. bölümde Karar verme, Karar verme süreci ve aşamaları, Karar verme modelinin öğeleri ve karar modelleri ile bu başlık altında; belirlilik halinde karar verme, belirsizlik halinde karar verme,risk halinde karar verme, kısmi bilgi halinde karar verme, rekabet halinde karar verme tipleri incelenmiştir.

4. bölümde çok ölçütlü karar verme, çok ölçütlü karar verme yöntemlerinin sınıflandırılması, çok ölçütlü karar verme yöntemleri ile bu başlık altında baskınlık, birleştiren, ayıran, ardışık sırasal, maximax, maximin, basit toplam ağırlıklandırma, ağırlıklı çarpım, TOPSIS, ELECTRE, PROMETHEE yöntemleri incelenmiş, AHP ile ilgili kısa bilgi verilerek detaylar için 6. Bölüm referans verilmiştir.

5. bölümde, bulanık mantık, bulanık sayılar ve türleri, bulanık kümeler, bulanık kümelerin özellikleri; 6. bölümde AHP; 7. Bölümde bulanık ahp, Van Laarhoven ve Pedrycz Yaklaşımı, Buckley Yaklaşımı, Chang'ın Mertebe Analizi Yaklaşımı, hesitant bulanık AHP, spherical bulanık AHP, intuitionistic bulanık AHP incelenmiş, pythagoreon bulanık AHP, neutrosophic bulanık AHP ve Z-Bulanık AHP ile ilgili örnek çalışmalar gösterilmiştir.

8. bölümde yazılım, uygulama tasarımı, kullanım ve akış diyagramları, uygulama fonksiyonları ve hesaplamalar açıklanmıştır. Uygulamanın kodları ekte verilmiştir.

9. bölümde geliştirilen model kapsamında AHP ve bulanık AHP uygulamalarının simülasyonu yapılmıştır. Uygulama simülasyonunda yukarıda belirtildiği gibi 5 kriter için 15 avukatlık ofisi değerlendirmeye tabi tutularak performans puanları hesaplanmış ve ofisler bu puanlara göre sıralanmıştır.



2.AVUKATLIK OFİSLERİNİN PERFORMANSI

Bu bölümde avukatlık ofisleri, türleri ve uzmanlık alanları; dış kaynak kullanmanın avantajları, dezavantajları ve borçların yasal takibinde dış kaynak olarak avukatlık ofislerini kullanmanın avantaj ve dezavantajları; çeşitli alanlarda yapılan performan ölçüm ve değerlendirmeleri; avukatlık ofislerinin performansları ve dış kaynak olarak avukatlık ofislerinin performans ölçüm ve karşılaştırmaları ile ilgili literatür taramasına yer verilmiştir.

Avukatlık ofisi bir veya daha fazla avukat tarafından avukatlık işlerini gerçekleştirmek amacıyla kurulmuş olan iş yerleridir. Hukuk ofisi, hukuk bürosu veya avukatlık bürosu olarak da isimlendirilir. Avukatlık ofisleri müşterileri adına hukuksal süreçleri takip ettikleri gibi danışmanlık hizmeti de sunarlar. Bir avukatın çalışma alanında sınır olmamakla birlikte kapsamın genişliği ve kanunların ayrışmasına bağlı olarak avukatlar veya avukatlık ofisleri de ceza hukuku, medeni hukuk, miras hukuku, ticaret hukuku, borçlar hukuku gibi belirli hukuki alanlarda uzmanlaşırlar. Ayrıca avukatlık ofisleri büyüklüklerine göre de gruplandırılabilir. Aşağıda çeşitli avukatlık ofis tipleri özetlenmiştir :

Tek Kişilik Avukatlık Ofisleri : Tek kişilik avukatlık ofisleri isminden de anlaşılacağı üzere bir avukat tarafından işletilen işletmelerdir. Bu firmalar bir alanda uzmanlaşabileceği gibi birden fazla alanda da hizmet verebilirler.

Küçük Avukatlık Ofisleri : Küçük avukatlık ofisleri genelde 2 ile 10 arasında avukatın çalıştığı işletmelerdir. Genelde alacak tahsilatı gibi karmaşık işler için çalışanlarına diğer avukatlarla işbirliği fırsatı sunarlar.

Büyük Avukatlık Ofisleri : Büyük avukatlık ofisleri onlarca avukat ve çalışana sahip olabileceği gibi binlerce avukat ve çalışana da sahip olabilirler. Bu tarz ofisler birden fazla şehir, bölge veya ülkede faaliyet gösterebilirler. Büyük avukatlık ofisleri birçok alanda hizmet verir ve firma içerisinde borçlar bölümü, ceza bölümü gibi bölümlere ayrılmışlardır.

Dava – İşlemsel Avukatlık Ofisleri : Avukatlık ofisleri sundukları hizmete göre de bölünebilirler. Müvekkillerini mahkemelerde savunmaya odaklı çalışan dava ofisleri olduğu gibi evrak işlerini takip etmek gibi işlemsel alanlara odaklı olarak da çalışabilirler.

Ceza Avukatlık Ofisleri : Avukatlık ofisleri gasp, zimmet, rüşvet, nitelikli dolandırıcılık gibi suçları işlemiş kişileri savunma konusunda uzmanlaşabilirler[1]

Firmalar alacaklarını son ödeme tarihinden sonraki 90 gün içerisinde tahsil edemediklerinde yasal takip sürecini başlatırlar. Bu süreçte borçlar mahkeme aracılığıyla tahsil edilmeye çalışılır. Borçluların sayısının fazlalığı, farklı bölge ve şehirlerde yaşamaları gibi nedenlerle firmalar yasal süreci takip etmeleri için dış kaynak olarak avukatlar veya avukatlık ofisleri ile çalışırlar ve hizmetleri karşılığı bu kişi veya ofislere ödeme yaparlar.

Tanyeri ve Fırat(2005), dış kaynak kullanımını ortaya çıkaran nedenlerini aşağıdaki şekilde sıralamışlardır[2] :

- **Organizasyonel Nedenler :** Firmalar, yapılanmalarını kaynak kullanımını minimumda tutarak daha güçlü olacak şekilde düzenlemeye çalışırlar.
- **Mali Nedenler :** Firmalar, yeterince başarılı olamadıkları alanlarda, gerekli verimi sağlayamadıkları işlerde dış kaynak kullanımına yönelerek bu alanlardaki gereksiz maliyetlerini kısarak sermaye getirisi elde ederler.
- **Finansal Nedenler :** Dış kaynak kullanımı, firmaların kendi kaynaklarını ve enerjilerini iyi oldukları alanlara yöneltip, bu alanlarda daha yoğun değer üretmesine olanak verir.
- **İnsan Kaynakları :** Firmalar, artan maliyetler, çalışanlara sağlanması gereken yan haklar ve artan işgücü çeşitliliği gibi sebeplerle insan kaynakları alanında dış kaynak kullanımına yönelirler.
- **Kaliteyi Artırma :** Firmalar, alanlarındaki uzmanlardan faydalanarak üretilen mal veya hizmetin kalitesini artırmak amacıyla dış kaynak kullanımını tercih ederler.
- **Teknolojik Yenilikler :** Teknolojinin hızla değiştiği günümüzde firmalar hayatta kalabilmek için değişen teknolojilere ayak uydurmak zorundadırlar. Teknolojiyi yakalamak için de bu alanlarda uzman dış kaynak desteği kullanırlar.

Dış kaynak kullanmanın bazı avantajları vardır. Bunlar çeşitli kaynaklarda farklı şekillerde sıralanmakla birlikte temel olarak;

- Maliyeti azaltmak: Firmalar yapılacak iş personel istihdam etmek yerine dış kaynaktan faydalanarak personel yetiştirme gibi maliyetlerde tasarrufa giderler.
- Kaliteyi artırmak : Alanında uzmanlaşmış dış kaynak firma desteği ile yapılacak olan iş için gerekli bilgi ve deneyim seviyesinde çalışanlardan hizmet alarak yapılan işin kalitesi artırılır. Alanında uzman kişilerden alınacak destek aynı zamanda iş süreçlerinin kalitesini de artırır.
- Esneklik : Firmalar değişen iş yoğunluklarına göre dış kaynak desteğini de değiştirebilirler. İş yoğunluğunun arttığı dönemlerde personel istihdam etmek veya iş yoğunluğunun azaldığı dönemlerde istihdamı azaltmak yerine bu dinamik süreçleri dış kaynak miktarını değiştirerek yönetebilirler.

Dış kaynak kullanımının bu faydaların yanında uzmanlığın dış kaynaklarda kalması, dış kaynaktan beklenen verimin alınamaması, müşteri ilişkilerinin dış kaynak tarafından zedelenmesi gibi dezavantajları da vardır.

Yasal takip sürecine giren alacak dosyalarının takibinin dış kaynak olarak anlaşılan avukatlık ofislerine devredilmesi, yukarıda bahsedildiği şekilde avantaj sağladığı gibi bazı dezavantajları da vardır. Avantajlar açısından bakıldığında, borçlar hukuku konusunda uzmanlaşmış ofisler ile çalışılarak bu alanda uzman avukatların istihdam edilmesinin getireceği maliyetten ve aynı zamanda uzman avukat bulunması ve işe alınması süreçlerinden de kurtulunmaktadır. Anlaşılan ofisler alacak takibi konusunda uzman olduklarından, toplanan alacaklarda artış olması öngörülmektedir. Ayrıca, değişen ekonomik şartlarda ödenmeyen alacakların miktarında artış ve azalışlar olacağından dinamik planlama yapılmasına da olanak vermektedir. Bunların yanı sıra, sorunlu bir süreç alacak sahibi firma dışında kişiler tarafından yönetileceğinden borçlu ile olan ticari ilişkiler daha az zedeleneyecektir. Diğer yandan, dezavantajlar açısından bakıldığında ise müşteri ile ilişkilerinin bozulma riski vardır. Aynı zamanda, benzer borçlu profillerinin davranışlarının benzer olacağını düşünürsek bu kitleye yaklaşımlar konusundaki bilgi ve tecrübe avukatlık ofislerinde kalacaktır.

Firmaların amacı ödenmeyen alacakların en yüksek miktarda tahsil edilmesidir. Bu nedenle yasal süreci takip etmesi için anlaşılan avukatlık ofislerinin performansı alacaklı firmalar için büyük önem arz etmektedir. Avukatlık ofislerinin performansına göre çalışmaya devam edip etmeme veya ne kadar büyük bir borçlu kitlesi için o ofisle çalışılacağı kararları verilmektedir. Bu nedenle birlikte çalıştıkları avukatlık ofislerinin performanslarının doğru şekilde ölçülmesi firmalar için önem arz etmektedir.

Avukatlık ofislerinin performansları yapılan iş alanına göre değişir. Ceza avukatları müvekkillerine ne kadar az ceza aldırır veya ceza aldırılmazsa o kadar iyi bir performans sergilemiş olurlar. Çalışmamızda avukatlık ofisleri ile ödenmeyen alacakların tahsilatı için anlaşıldığından ofislerin performansındaki temel gösterge ödenmesi sağlanan alacak miktarıdır. Ancak, her borç birbiri ile eşdeğer değildir. Bir borcun tahsil edilebilirliği değişkenlik göstermektedir.

Bir borcun ödenmesine etki eden çeşitli faktörler vardır. Onar,Öztayşi ve Kahraman tarafından yapılan çalışmada [3] borcun ödenmesi aşağıdaki kriterlere göre değerlendirilmiştir:

- Borç miktarı
- Borçlunun mal varlığı
- Borçlunun diğer borçlarının miktarı
- Borçlunun geçmişi
- Borçlunun müşterilik süresi

Łada, Monika., Wejer-Kudelko 2018 yılında Polonya’da yaptıkları araştırmada[4], anket yöntemi kullanarak borç toplama performansına artıran ve sınırlayan faktörler üzerine çalışmışlardır. Çalışmada Polonya Kurumsal Hazineciler Birliği (Polish Corporate Treasurers Association’s) üyesi 148 kişiye anket gönderilmiş, 83 kişiden yanıt alınmıştır. Anket sonucunda ortaya çıkan sonuçlar ve yüzdeleri aşağıdaki gibidir :

Borç toplama performansını artıran faktörler :

- Başarılı kapanan dosyalar için işlemi gerçekleştiren çalışana ödenen ücret - %76,1
- Dosyaların online yönetilebilmesi - %55

- Borçlu ile olan ticari ilişkiler - %48
- Borç zorluk puanı - %42
- Yargı sürecinin netliği - %38
- Çalışan başarı kriterlerinin açık ve net olması - %37
- Son ödeme tarihi öncesi borçlu ile iletişim kurulması - %31
- Anlaşmazlıkların elektronik ortamda çözülebilirliği - %30
- Yargı süreci ile ilgili borçlunun bilgilendirilmesi - %28
- İcra için elektronik açık arttırma olması - %28
- Anlaşmazlık çözümlerinde alternatif yolların kullanılması - %24
- Güvenilir kar – zarar analizinin olması - %21
- Borç toplama işlemleri için dış kaynak kullanımı - %11 (Borç toplama işlemlerinde dış kaynak kullanımının müşteri ile olan ticari ilişkilere daha az zarar verdiği görüşü belirtilmiştir)
- Yargı sürecinde arabulucuların kullanılması - %11

olarak sıranlamıştır.

Borç toplama performansını sınırlayan faktörler ise aşağıdaki şekilde sıralanmıştır[4]:

- Yargı süreçlerinin ve icra işlem sürelerinin uzun olması - %86
- Yargı ve icra süreçlerinin çok çeşitli olması - %63
- Borç toplama maliyetinin yüksek olması - %54
- Kesin olmayan yasal hükümler - %54
- Borç toplama sürecinin her adımında avukat danışmanlığı alma - %44
- Müşteri kaybetme korkusuyla borç toplama adımlarının ilettilmesinin geciktirilmesi - %38
- Borç toplamakla görevli çalışanın negatif algıları - %35
- Elektronik ortamdaki yazışmaların mahkemelerce yeterli görülmemesi - %34
- Borç toplama işlemini yapan çalışan ile borç veren arasındaki iletişim sorunları – %32
- Anlaşmazlık çözümlerinde yasal görevliler tarafından mahkumiyet kararlarının eksikliği - %20
- Arabulucuların yetersiz olması - %18

Beck, Grunert ve Jens tarafından 2017 yılında çalışmada[5], borç toplama ajanslarının başarı oranlarına etki eden faktörler incelenmiştir. Çalışma sonucunda, borçlunun yaşı ile ödeme oranı arasında negatif bir ilişki tespit edilmiştir. Ödenmeyen borcun son ödeme tarihinden itibaren geçen süre arttıkça ödeme oranı azalmaktadır. Ayrıca eğer borçlunun daha önce başka borçları borç toplama ajansları ile iletişimi olmuşsa ödeme oranı azalmaktadır. Borçlunun geçmişinde ödenmeyen borçlar az ise ödeme oranı artmaktadır. Bunun yanı sıra eğer borç taksitlendirilirse ödeme oranı artmaktadır. Son olarak, eğer borç miktarı düşükse ödeme oranı artmaktadır[5].

Phillips ve Moggridge tarafından 2019 yılında yapılan çalışmada[6], borç toplamada yapay zeka üzerine çalışılmıştır. Hali hazırda borç toplama endüstrisinde yapay zeka kullanılmaktadır. Yapay zeka, büyük miktardaki geçmiş dönem verisinden faydalanarak mesaj kampanyaları, müşteri (borçlu) hizmetlerinin kişiselleştirilmesi ve boç yönetimi alanlarında kullanılmaktadır. Yapılan araştırmalar müşteri hizmetlerinin kişiselleştirilmesinin geri ödeme oranlarında artış sağladığı görülmüştür. Ancak bu sistemler büyük miktarda geçmiş dönem verisi ve günce veriye ihtiyaç duyduğundan büyük kuruluşlar için etkiliyken küçük ve orta ölçekli kuruluşlar tarafında yeterince kullanılamamaktadır.

Yapılan çalışmada, yukarıda anlatıldığı gibi interaktif ve gerçek zamanlı olmayan uygulamaların yerine borçluyla anlık iletişim kurabilecek ve pazarlık yapabilecek interaktif ve gerçek zamanlı çalışacak sanal temsilciler önerilmiştir. Borç toplama için kullanılacak olan bu sanal temsilci borçlunun yalan söylediğini tespit edebilecek, yararlanabileceği alanları belirleyebilecek ve bir pazarlık stratejisi oluşturabilecektir. Sanal temsilciler borç toplama işlemlerinde daha iyi çıktılar, müşteri bazlı hizmet, sorunlu müşteri tespiti, karmaşık borçlarla başa çıkabilme yetisi, paralel olarak yüksek miktarda borçlu takibi yapabilme ve kuralları belirli olduğundan yasal süreçlere kesin uyum gibi olumlu katkılar sağlayabilecektir. Bu da firmalara büyüme, karlılık, iş zekası ve müşteri memnuniyeti açısından fayda sağlayacaktır[6].

Literatürde performans ölçümü ve kıyaslaması çok farklı alanlarda ve çok farklı yöntemlerle yapılmıştır. Örneğin; Kaptanoğlu[7], bulanık analitik hiyerarşik süreci kullanarak akademik performans değerlendirmesi yapar. Tyagi ve Singh, çok kriterli karar verme yöntemi ile hastanelerde performans yönetimi üzerine bir model sunmuşlardır.

Keleş,Mehmet Özkan 2018 yılında yaptığı tez çalışmasında[8], Türkiye’deki koster işletmelerinde Gemicilik Anahtar Performans Göstergelerini (Shipping KPIs) kullanarak dış kaynak kullanımının firmaların performansları üzerindeki etkisini incelemiştir.

Bucovetchi, Stanciu, ve Badea 2019 yılında yaptıkları çalışmada[9], ÇKKV yöntemlerini kullanarak, firmalar ile yaptıkları anket çalışmalarıyla, firmaların lojistik birimlerini dış kaynak olarak kullanmanın firmanın genel performansına etkisini incelemiştir. Yaptıkları çalışma sonucunda düşük maliyet, alanlarında uzman olmaları, işin boyutuna göre anlaşmanın değiştirilebilir olması, düşük risk ve coğrafi olarak daha geniş alanlara hizmet edebilmeleri gibi nedenlerle lojistik biriminin üçüncü firmalara verilmesinin performansı arttırdığı sonucuna varmışlardır.

Bernard, Moxnes ve Saito, 2019 yılındaki çalışmalarında[10] üretim ağı ve coğrafi konumlanmanın firma performansına etkilerini incelemiştir. Çalışmalarında alıcı-satıcı arasındaki ilişkiyi kendi geliştirdikleri modelle ölçmüş, arama ve dış kaynak kullanım maliyetlerinin azalmasına bağlı olarak aramaların arttığını ve dış kaynak kullanımının artarak genel maliyetin azaltıldığını tespit etmişlerdir. Japonya’daki hızlı tren projesi üzerinde modellerini uygulayarak testlerini gerçekleştirmişlerdir.

Świtła, Cichosz ve Joanna, 2019 yılında yaptıkları çalışmada[11] lojistik tercihlerinin müşteri memnuniyetine etkisini incelemiştir. Bu çalışmada, müşterilere uygulanan anket çalışması sonucunda, müşterilerinin %30’unun firmanın dağıtım performansından memnun olmadığı, %70’nin ise memnun olduğu tespit edilmiştir. Müşteri grupları ve çalıştıkları firmalar analiz edildiğinde, dağıtım performansından memnun olmayan kesimin çalıştığı firmaların daha küçük ölçekli ve lojistik için dış kaynak kullanımlarının düşük olduğu tespit edilmiştir. Dağıtım performansından memnun olan kesimin çalıştığı firmaların ise daha büyük ölçekli ve düzenli olarak iş ortaklığı yaptıkları geniş erişim ağı olan lojistik dış kaynaklarının olduğu tespit edilmiştir.

Espino-Rodríguez ve Ramírez-Fierro, 2018 yılında yaptıkları çalışmada[12] İspanya’da turistik bölgede yer alan bir otel üzerinde dış kaynak performansını ölçmeye çalışmışlardır. Değerlendirme girdisi olarak finansal dış kaynak performansı, stratejik dış kaynak performansı, genel memnuniyet, dış kaynakların görülen faydaları ve dış kaynaklarla çalışma oranı belirlenmiştir. Hesaplama yöntemi olarak kısmi en

küçük kareler yöntemi kullanılmıştır. Çalışmanın sonucunda aşağıdaki çıkarımlar elde edilmiştir;

- Yöneticilerin dış kaynakla çalışmanın faydalarını bilmeli ve yakın ilişkide olmanın farkında olmalıdır,
- Yöneticiler dış kaynak organizasyonlarının yapısına hakim olmalı ve yol gösterecek yeterlilikte olmalıdır,
- Dış kaynakla çalışma oranının artışı işbirliğini artırmaktadır.
- Yöneticiler, dış kaynakların performansını artırmak için daha yüksek işbirliği içerisinde olmalıdır,
- Dış kaynaklarla olan işbirliği seviyesinin artması finansal ve stratejik açıdan performans artışını mümkün kılmaktadır,

Prajapati, Kant ve Tripathi, 2020 yılında yaptıkları çalışmada [13] dış kaynakların kritik başarı faktörlerini ve bu faktörlerin uyum nedeniyle performans çıktılarının önceliklendirilmesini incelemişlerdir. Stratejik faktörler, organizasyonel faktörler, finansal ve mali faktörler, operasyonel faktörler ve insani faktörler olmak üzere 5 grup altında 26 adet başarı faktörünün kullanıldığı çalışmada 13 tane de performans çıktısı kullanılmıştır. Çalışmanın sonucunda kritik başarı faktörleri stratejik faktörler başta olmak üzere organizasyonel faktörler ve finansal ve mali faktörler şeklinde sıralanmıştır. Performans çıktısı olarak da; temel yetkinliklerdeki artış, kapasitenin dinamik olması, iyileşen finansal performans, kaynak kullanımının optimizasyonu elde edilmiştir.

Öztürk, Murat ve Elevli'nin 2019 yılında Samsun'daki bir hastanenin çağrı merkezinde yaptıkları çalışmada[14], çağrı merkezi çalışanlarının performansını ölçmede kullanılan metriklerin değişkenliklerini ölçmeyi hedeflemişlerdir. Performans ölçümünde kullanılan metrikler; ortalama çağrı yanıtlama süresi, tamamlanmayan çağrı oranı ve hizmet seviyesidir. Yapılan ölçümler sonucunda metrikler için elde edilen değerler fazla değişkenlik göstermektedir. Performansın bu kadar değişken olmasının önüne geçilmesi için çalışanlara daha iyi eğitim verilmesi, çalışanların sözleşmelerinin iyileştirilmesi, daha iyi çağrı merkezi teknolojilerinin kullanılması ve güç,telefon hattı gibi altyapıların iyileştirilmesi önerilmiştir.

Bulut ve Ataay tarafından 2017 yılında yapılan çalışmada[15] çağrı merkezi çalışanlarının performanslarını etkileyen faktörler incelenmiştir. Parametre olarak

alışanların deneyimi, iş yükü ve alışma süreleri alınmıştır. İnceleme sonucunda alışanların deneyim süreleri ile performansları arasında pozitif bir ilişki görülürken iş yükü ve alışma süreleri ile performansları arasında negatif bir ilişki tespit edilmiştir. Yani daha deneyimli alışanlar daha iyi performans sergilerken iş yükü ve / veya alışma süresi uzun olan alışanlar daha kötü performans sergilemiştir.

Karakuş ve Aydın, 2016 yılında yaptıkları alışmada[16], büyük veri analizi yöntemleri kullanarak aęrı merkezi alışanlarının performanslarını ve hizmet kalitesini ölçmeyi hedeflemişlerdir. alışanın aęrı kalite performansı yönetici tarafından alışanın yaptığı görüşmelerden rastgele seçilen birkaçı dinlenerek değerlendirilmektedir. Dinlenen örneklerin toplam aęrı sayısına oranı çok düşük olduğundan bu değer gerçek performansı yansıtmamaktadır. Bu nedenle bu alışma tüm aęrı kayıtları üzerinden büyük veri analizi ve metin benzerlik algoritmalarından yararlanarak ölçümün doğruluğunu arttırmanın yanı sıra maliyetlerden tasarruf sağlamayı, gelirleri arttırmayı ve müşteri memnuniyetini arttırmayı hedeflemiştir.

Hsu, Chen, Chan ve Chang tarafından 2016 yılında alışmada[17], sinir aęları kullanarak aęrı merkezi alışanlarının performans değerlendirmesi üzerine alışmışlardır. Performans değerlendirme kriterleri;

- Servis kalite göstergesi
- Test puanı
- Personel devamlılığı
- Saatlik toplam aęrı sayısı
- İlk aęrı çözünürlüğü
- Anket başarı oranı
- Müşteri memnuniyeti

olarak belirlenmiştir.

Avukatlık ofislerinde performans ölçümü daha çok yöneticilerin kişisel değerlendirmeleri ile yapılmaktadır[18].

Lopes,Duerta ve Lopes yaptıkları alışmada[18], yapay sinir aęlarını kullanarak avukatların performansını tahmin etmeye alışmışlardır. Sistemin eğitilmesi için geriye dönük performansları kullanılmıştır. Bu performanslar firma sahipleri tarafından avukatların puanlanması ile elde edilmiştir. Puanlama teknik ve kişisel

yetkinlikler başlıkları altında yapılmıştır. Teknik yetkinlikler teknik bilgi ve üretilen çözümler, iletişim ve müşteri odaklılık gibi kriterleri barındırırken, kişisel yetkinlikler iş geliştirme, firma odaklılık, liderlik, kaynak yönetimi ve başarı odaklılık gibi kriterler barındırmaktadır. Daha sonra bu puanlar avukatların tecrübelerine (tecrübesiz, orta seviye tecrübeli ve tecrübeli) göre ağırlıklandırılarak nihai performans puanları elde edilmiştir. Performans tahmini yapılan iki yılda avukatların performans sıralaması sırasıyla %67 ve %69'luk doğrulukla bilinmiştir[18].

Compensation & Benefits for Law Offices'te yayınlanan avukatlık ofisinizde performans kültürü oluşturma'nın anahtarları isimli makalede[19], bir avukatlık ofisinde performans kültürü oluşturmak için aşağıdaki adımlar önerilmiştir :

- Yüksek hedefler konulması,
- Ofis hedefleri ile performans arasında sarsılmaz bir ilişki kurulması,
- Performansı ödüllendiren bir ücretlendirme politikası,
- Problemlerin ertelenmesi ya da saklanması yerine çözülmesinin teşvik edilmesi,
- Bireysel olarak en iyi performansa ulaşılabilmesi için gerekli eğitimlerin verilmesi,
- Pozitif baskı oluşturulması[19]

Dış kaynak olarak avukatlık ofislerinin performansı üzerine ise az sayıda çalışma vardır. Kahraman, Öztayşi ve Cevik Onar [1] tarafından yapılan çalışmada, Z-Tipi bulanık analitik hiyerarşi süreci kullanılarak avukatlık ofislerinin performansları karşılaştırılmıştır. Burada performans değerlendirme ölçütü olarak atanan dosyalardaki alacak toplama başarısı (\$), erken alacak toplama başarısı(%), atanan dosyaların zorluk katsayısı, hedeflerine uygunluk ve müşteri memnuniyeti kullanılmıştır.

Yine Kahraman, Öztayşi ve Cevik Onar tarafından yapılmış diğer bir çalışmada[20] dış kaynak olarak çalışılan avukatlık ofislerinin performans analizi tek ve aralık değerli neutrosophic AHP kullanılarak değerlendirilmiştir. Burada performans değerlendirme ölçütü olarak atanan dosyalardaki alacak toplama başarısı (\$), erken alacak toplama başarısı(%), atanan dosyaların zorluk katsayısı, hedeflerine uygunluk ve firma bilinirliği parametreleri kullanılmıştır.

- Atanan dosyalardaki alacak toplama başarısı, avukatlık ofislerinin kendilerine verilen dosyaların ne kadar fazlasını kapatırlarsa o kadar başarılı olduklarını ifade eder.
- Erken alacak toplama başarısı, firmaların alacakları ne kadar erken tahsil ederlerse o kadar başarılı olduklarını ifade eder.
- Firmalara verilen dosyalar aynı zorlukta olmadığını belirtmiştik. Atanan dosyaların zorluk katsayısı , bir avukatlık ofisi için kendilerine verilen zor dosyaları tahsil etmeleri durumunda daha başarılı olacaklarını ifade eder.
- Hedeflerine uygunluk, avukatlık ofisi çalışanlarının borç toplamının etik kurallarına riayetleri arttıkça performansın artacağını ifade eder.
- Firma bilinirliği, avukatlık ofislerinin geçmiş dönemlerdeki çalışmalarına istinaden insanların sempati beslemesi,saygı duyması yada takdir etmesi durumunda daha fazla tercih edileceğini ifade eder. [20]

Tubitak tarafından TEYDEB 1505 programı ile desteklenen “Akıllı Çok Ölçütlü Yasal Takip Avukatlık Ofisi Notlandırma ve Performans Yönetimi Sistemi” projesi kapsamında yapmış olduğumuz bu tez çalışmasında literatürde üzerinde az çalışılmış olan avukatlık ofislerinin yasal takip dosyalarındaki performanslarının ölçülmesi ve karşılaştırılması üzerine hem analitik hiyerarşi prosesi hem de bulanık analitik analitik prosesi kullanılarak alternatifli bir sistem geliştirilmeye çalışılmıştır.



3.KARAR VERME VE TEMEL KAVRAMLAR

3.1 Karar Verme

Tarihin başlangıcından günümüze kadar insanlar yaşamlarının neredeyse her anında karşılarına çıkan problemleri çözmek için entellektüel birikimlerini kullanarak çeşitli kararlar vermek mecburiyetinde kalmıştır. Karar verme insanların hayatında, kişisel veya organizasyonel boyutta bir çok farklılıklar göstermekle birlikte vazgeçilmez bir olgu olarak mevcuttur ve mevcut olmaya devam edecektir. İçinde bulunduğumuz günün şartlarını göz önüne aldığımızda; gelişen teknoloji sayesinde globalleşen dünyada hızlı bir şekilde değişen sosyal ve ekonomik şartlar karşısında alınan kararların önemi ve etki alanı dikkate alındığında verilen her kararın ne kadar önemli olduğu görülebilir.

Literatürde karar verme ile ilgili olarak pek çok tanım yapılmıştır. Bunlardan bazıları;

Karar verme; bireyin, kurumun , kurumdaki bir yöneticinin veya herhangi bir karar vericinin birbirlerinin yerine geçebilme özelliğine sahip birden fazla seçenek arasından bir tanesini tercih etmesi olarak tanımlanabilir[21].

Bir amacı gerçekleştirmek için alternatifler arasından en iyi hareket biçimini seçmektir.[22].

Karar verme, bir amacın gerçekleştirilmesi için en iyi hareket biçimini seçme, seçilen hareketi uygulama ve uygulamayı kontrol etme adımlarından oluşan bilinçli bir hareketler bütünüdür[23].

Karar verme, mümkün olan tüm seçenekler arasından amaç veya amaçlara en fazla hizmet etmeye uygun bir veya daha fazlasını seçme sürecidir[24].

İnsanın yaşamına başlamasıyla birlikte ortaya çıkan karar verme olgusu tüm yaşamı süresince farklı koşullarda ve formlarda varlığını sürdürür. Benzer şekilde kurum ve kuruluşlar da farklı şartlar altında birbirini takip eden karar verme durumuyla her zaman karşılaşır. Karar verme zincirinin bir halkasını oluşturan etmenler aşağıdaki şekilde özetlenebilir:

- Karar vermekle yükümlü ve yetkili kişi (veya kişiler)
- Karar verme ortamı (sınırlar)

- Amaçlar (kriterler, hedefler)
- Seçenekler
- Kaynaklar

Yukarıda verilen tanımlamalarda da görüldüğü gibi bir karar verici, içerisinde bulunduğu şartlar ve elindeki imkanlar dahilinde tatminkar bir çözüme ulaşabilmek için birden fazla kriteri hesaba katmak durumundadır.

3.2 Karar Verme Sürecinin Aşamaları

Bir kararın alınmasında kullanılan teknik ve yöntemlerin düzeni karar verme süreci olarak ifade edilir. Bu şekilde, karmaşık problemler sistemli bir şekilde incelenmekte ve alternatifler arasından en iyi karara ulaşabilmek için şartlara en fazla uyum gösteren yöntemler belirlenerek verilecek olan kararın kalitesi artırılmaktadır. Karar teorisi, karar vericilere destek olmak amacıyla karar sürecindeki belirsizlik ve karmaşayı azaltarak alternatiflerin düzenli bir şekilde değerlendirilmesini hedeflemektedir.

Karar verme süreci literatürde farklı araştırmacılar tarafından farklı şekillerde tanımlanmıştır.

Çelikyay ,

1. Karar verme amacının belirlenmesi,
2. Kontrolü sağlanabilen parametrelerin tayin edilmesi,
3. Kontrolü sağlanamayan parametrelerin tayin edilmesi,
4. Kısmen kontrolü sağlanabilen parametreleri ve bu parametreler ile kontrolü sağlanabilen parametreler arasındaki ilişkilerin tayin edilmesi,
5. Olası her bir kararın etkisinin amaca bağlı olarak tayin edilmesi,
6. Kararın verilmesi,
7. Verilen kararın sonuçlarının değerlendirilmesi,
8. Karar verme sürecinin daha sonraki çalışmalar için tekrar edilmesi [25] olarak tanımlamıştır.

Karar verme sürecinin adımları Hicks tarafından 5 adımda sıralanmıştır;

1. Karar verme amaçlarının tayini,
2. Amaçlara ulaşmayı sağlayacak seçeneklerin belirlenmesi,
3. Değerlendirme için kullanılacak olan kriterlerin/yöntemlerin tayini,

4. Amaca ulaşmak için en iyi hareket yönteminin seçilmesi,
5. Seçilen hareket yönteminin uygulanması [26].

Koçel karar verme süreç adımlarını aşağıdaki şekilde sıralamıştır;

1. Amacın belirlenmesi veya sorunun tanımlanması,
2. Amaç ve sorunların irdelenerek önceliklerin belirlenmesi,
3. Alternatiflerin belirlenmesi,
4. Alternatiflerin irdelenerek değerlendirilmesi,
5. Seçim kriterlerinin belirlenerek alternatifler arasından seçim yapılması [27].

Aytaç ve Gürsakal ise karar verme süreç adımlarını aşağıdaki 8 adımla sıralamıştır;

1. Karar verilecek konunun veya problemin tanımlanması,
2. Problemi doğuran nedenlerin ortaya konulması,
3. Karar verilecek alternatiflerin belirlenmesi,
4. Alternatif seçeneklerin ortaya konulması,
5. Bazı seçeneklerin elenmesi,
6. Gerçekçi olmayan seçeneklerin elenmesi ve geçici kararın verilmesi,
7. En uygun seçeneğin seçilmesi,
8. Verilen kararın uygulanması[21].

3.3 Karar Modelinin Öğeleri

3.3.1 Karara Konu Problem

Problem, kişinin içerisinde bulunduğu durumdan veya ulaşmaya çalıştığı bir amaca ulaşmasından kaynaklanan hoşnutsuzluk, negatif bir uyarım hissi veya uzak durmak, değiştirmek istediği bir engel olarak ifade edilir. Farklı bir ifade ile, kişinin içinde yer almak istediği durum ile mevcut durumu arasındaki fark problemdir [28].

Bir problemin ortaya çıkışı karar verme sürecinin ilk aşamasıdır. Problem karar vericinin karşısına iki farklı şekilde çıkabilir;

1. Karar verme yeti ve yetkisine sahip olan kişi çevrelerinde karar vermelerine neden olabilecek değişiklikleri ya da karar vermelerini gerektirecek yeni koşulları tespit edebilmek amacıyla teknik, siyasi, iktisadi ve toplumsal konuları irdeler. Örgüt için çözülmesi gereken bir problem olup olmadığını araştırır.

2. Karar vericiler araştırmadan da problemlerle karşı karşıya kalabilirler. Bu durum örgüt dışı grupların çeşitli isteklerde bulunması ya da problemin örgütün içerisinde çıkmasıyla oluşur. Problem, örgüt içerisinde karar verici pozisyonun üst veya alt kademelerinden gelebilir. Emir veya genelgelerdeki belirsizlik, emirlerdeki veya yetki alanlarındaki çatışma vb. nedenlerle astlar üstlerinden bir problemle ilgili olarak karar vermesini üstlerinden isteyebilir. Bunun yanı sıra astlar, kendileri ile ilgili bir problem için üstlerinden çözüm bulmalarını isteyebilirler[29]

Problem ile her ne şekilde karşılaşılırsa karşılaşılсын, öncelikle detaylı bir şekilde incelenmelidir. Kararın iyiliği, karşılaşılan problemin her yönüyle net, açık ve doğru ifade edilmesine bağlıdır.

3.3.2 Karar Verme Sürecinin Katılımcıları

Her ne kadar karar verme süreci içerisinde bir karar verici varmış gibi görünse de bu sürecin işleyişi esnasında bu karar verici başka insanların etkisi altında kalır. Dış etmenlerden etkilenmeden tek başına karar veren bir karar verici mevcut değildir. Karar verici, dolaylı olarak bir grup adına karar veren bir kişi olabileceği gibi direkt bir karar vericinin olmadığı durumlar da mevcuttur[30,31].

Bir sorunun çözümü için karar vermeyi gerektiren herhangi bir süreç aşağıdaki kişilerin katılımıyla ilerler;

- Karar Verici / Sorun Sahibi

Alınmaya çalışılan karar ve sorun durumunun belirli yönleri üzerinde kontrol sahibi olan kişi veya grup

- Çözüm Kullanıcı

Karar verici tarafından oluşturulan çözümü kullanan ve/veya kararı uygulayan fakat çözüm/karar üzerinde herhangi bir değişiklik yetkisi olmayan kişi/kişiler

- Karardan Etkilenenler

Çözümün veya kararın sonuçlarından faydalanan ya da zarar gören kişiler

- Karar Analisti / Sorun Çözücü

Sorunun analizini yapan ve karar verici için çözüm geliştiren ya da çözüm sürecinde karar vericiye yardım eden analist[28].

3.3.3 Seçenekler

Bir karar verme sorunu ile ilgili tüm alternatifler karar verme sürecinin seçeneklerini oluşturur. Sorunun çözümünde kullanılması olası tüm alternatifler bulunmaya çalışılır. Alternatif kümesi ne kadar genişse en iyi çözüme ulaşmak o kadar kolaylaşır. Çözüm için bulunan alternatifleri içeren gruba seçenek kümesi denir.

Seçenek kümeleri iki farklı yöntemle gösterilebilir :

- Seçeneklerin sonlu ve sayılabilir sayıda olduğu durumlarda eleman listesi halinde,
- Seçeneklerin sonsuz ya da sayılamayacak kadar fazla olduğu durumlarda ise matematiksel ifadeler ile[32].

Karar verme sırasında her zaman göz önünde bulundurulması gereken iki nokta mevcuttur ;

1. Düşünülmeyen bir seçenek seçilemez(Seçenek kümesinde yer almayan bir seçenek seçilemez)
2. Seçenekler ne kadar fazla olursa olsun seçilen alternatif, eldeki kümenin en iyisidir.

Buradan da anlaşıldığı üzere seçenek kümesinin olabildiğince büyük(alternatifli) olması iyi bir çözüm bulma olasılığını artırır. Eldeki seçenekler çok fazla ise karar vericinin isteğine göre eleme yapılabilir.

3.3.4 Amaçlar,kriterler ve ölçütler

ÇKKV için eldeki seçenekleri değerlendirmek için faydalanılan

- Amaç,
- Kriter,
- Ölçüt

gibi kavramlar farklı kişilere farklı anlamlar ifade etmektedir. Duruma bağlı olarak birbirlerinin yerini alabilen ve üzerinde anlaşılmış tek bir tanıma sahip olmayan bu kavramlar aşağıdaki başlıklar altında açıklanmıştır.

Karar ölçütleri veya kriterler, kararda esas alınacak temel kuralları, karar değişkenleri arasındaki ilişki şeklinde belirler. Bu ilişkinin, arzu edilen daha iyi bir yöne doğru yönlendirilmesi, bunları amaçlara, amaç fonksiyonlarına dönüştürür. Amaçlar için belirli sabit değer belirlemek hedefleri tanımlayacaktır. Literatürde bu tanımlar arasında kesin bir ayrım gözetilmeyip, değişik yerlerde, değişik kişilerce farklı şekillerde kullanılmaktadır[33].

3.3.4.1 Amaçlar

Kriterlerin KV'nin isteklerine göre yönlendirilmesi amaç olarak tanımlanabilir. Üç özellikle tanımlanır:

- Karar kapsamı,
- Karar konusu ,
- KV tercihlerinin yönü.

Bu durumda amaç, KV'nin arzuları doğrultusunda seçeneklerin temel özelliklerinin ve değerlendirme parametrelerinin yönlendirilmiş şekli olarak tanımlanabilir [33].

Amaçlar, seçeneklerin değerlendirilmesinin temelini inşa eder. Karar verme durumu ile ilintili tüm amaçların tanımlanarak karar verme sürecinde bunlardan faydalanılması dengeli bir karar alınmasına olanak sağlar.

Amaçlar,

- aranan bilginin belirlmesine
- KV'nin verdiği kararı diğer insanlara açıklamasına
- verilen kararın ne kadar önemli olduğunu ve nihai olarak zaman ve emek ihtiyacının belirlenmesine katkı sağlar [34].

3.3.4.2 Kriterler

Kriterler, karar problemlerinde alternatiflerin değerlendirilerek en uygun olanının seçilmesi, alternatiflerden birinin yeterliliğinin kanıtlanabilmesi veya amacın gerçekleştirilmesinde ne derece katkı sağlayabileceğinin belirlenmesinde kullanılan elemanlardır.

Karar verme sürecinde yönlendirici rol oynayan ölçü, kural ve standart bütünü kriter olarak isimlendirilir. Kriterler, alternatifler arasında karşılaştırma yapılmasını sağlar. Kriter ve ana amaç ile alternatifleri birbirleri ile ilişkilendirebilmek için alt amaç ve ölçüler içeren ara kademelere sahip bir hiyerarşik yapı kurulabilir. Kriter KV'nin tercihine daha fazla etki eden alternatif özelliklerinin yanında ulaşmak istediği hedefi de gösterir[28].

3.3.4.3 Ölçütler

Ölçütler, ulaşmak istenen amaç için elde olan seçeneklerin temel özelliklerini, kalitelerini ve verimliliklerini ölçmeye yarayan parametrelerdir. Ölçütler, amaca giden yoldaki alternatiflerin kriterlere göre değerlendirilmesinde kullanılır. Özet olarak ölçüt, amaca ne kadar ulaşıldığını gösteren ölçüdür denilebilir.

Bir ölçüt için aranan özellikler aşağıdaki gibidir :

- **Ölçülebilirlik** : Değer yargılarını değerlendirerek uygun olmayanları devre dışı bırakmalı ve böylece amaçlar için ölçüm yaparak anlamlandırılmalıdır.
- **Kullanılabilirlik** : Muhtemel sonuçların tanımını amaca uygun olarak yapmalı ve amacın başarı oranını değer yargılarına göre göstermelidir.
- **Anlaşılabilirlik** : Sonuçların ölçütler cinsinden yorumlanmasının net olması gerekmektedir [28].

3.3.5 Yöntemler

Yöntem, belirli bir sonuca erişmek için bir plana göre izlenen dizgisel yoldur. Karar verme süreci için sorun yapılandırılması, model kurulması, sentez ve duyarlılık safhalarıyla ilgili çeşitli yaklaşımlar baz alınarak ortaya konulmuş birçok yöntem vardır. Bu yöntemlerle ilgili detaylar ilerleyen bölümlerde verilecektir.

3.4 Karar Modelleri

Karar verme sürecinde çeşitli modeller kullanılır. Bu modeller arasındaki farklılık ortaya çıkması beklenen olay ve KV'nin olaylar hakkındaki bilgi seviyesinden kaynaklanır. Karar modelleri, olaylar ve gerçekleşme olasılıklarına göre aşağıdaki başlıklar altında toplanabilir :

1. Belirlilik halinde karar verme
2. Belirsizlik halinde karar verme

3. Risk halinde karar verme
4. Kısmi bilgi halinde karar verme
5. Rekabet halinde karar verme [27]

3.4.1 Belirlik Halinde Karar Verme

Tüm alternatiflerin sonuçlarının karar verici tarafından bilinmesi haline belirlilik denir. Bu durumda alternatifler arası karşılaştırma yapmak kolay hale gelir. Bir KV, belirlik halinde karar veriyorsa problemle ilgili olarak tüm alternatifleri, alternatifler tarafından oluşturulan koşulları bilmekte ve sonuçları belirleyebilmektedir. Her alternatifin kazanımları bilinir ve stabildir[35].

3.4.2 Belirsizlik Halinde Karar Verme

Belirsizlik altında karar verme durumlarında gelecekte gerçekleşmesi beklenen olaylar için ihtimal hesabı yapmak söz konusu değildir. Bazı karar verme durumlarında gelecekteki olaylar için ihtimal belirlemek uygun değildir ya da imkansızdır.

Karar teorisinde belirsizlik halleri için en iyi kriter henüz belirlenmiş değildir. En iyi kriter yerine, doğrulayacak gerçekçi yönleri bulunan birçok kriter mevcuttur[35].

Yukarıdaki kriterlerden bazıları :

- Maximin (Kötümserlik) Kriteri
- Maximax (İyimserlik) Kriteri
- Minimax (Pişmanlık) Kriteri
- Hurwicz (Uzlaşma) Kriteri
- Laplace Kriteri

3.4.3 Risk Halinde Karar Verme

Tamamıyla belirsiz ortamlar gerçek hayatta ender olarak karşımıza çıkar. Belirli olmayan ancak tamamıyla da belirsiz olmayan durumlarda karar problemi risk halinde karar verme problemi olarak ifade edilir. Risk durumu, alternatiflerin sağlayacağı olası sonuçlar için net bilginin sağlanamadığı durumlardır. Risk halinde karar verme durumunda, bağımlı parametrelerin hepsi tespit edilebilmekle birlikte, bağımlı parametrelerin ve bun arasındaki ilişkinin değeri olasılıklı olarak ifade edilir.

Belirli sayıda olayın var olduğu bu tip problemlerde olaylar için meydana gelme ihtimallerinin de tespit edilebildiği varsayılır. Olayların olasılık dağılımları bilinerek uygulanacak olan karar kriteri, "Optimum Beklenen Değeri" en iyi olan

alternatifin bulunması problemidir. Problemden yer alan olayların gerekleşmesinin dağılıma baėlı olduėu durumlarda dağılımın diėer karakteristikleri (varyans vs.) de kullanılabilir. [35]

3.4.4 Kısmi Bilgi Halinde Karar Verme

Hangi olasılık dağılımına (Normal, Poisson, Binominal, vb.) uygun hareket edildiėi ve bu dağılımın deėişkenleri ile karakteristik(Ortalama, Mod, Medyan vb.) deėerleri biliniyorsa, kısmi bilgi dahilinde karar verme durumu oluşur. KV'nin en iyi kararı verebilmesi için olaylara dair ek bilgi talebi olabilir. Yeni bilgiler ışığında ön olasılıklar güncelleştirilerek KV'nin olaylar hakkında son kararı verebilmesi için eskisinden daha iyi sonuçlar verecek olasılık tahminleri yapılır[21].

3.4.5 Rekabet Halinde Karar Verme

Oyun teorisi,her biri kazanma arzusuna sahip iki ya da daha fazla sayıdaki tarafın en uygun stratejisinin nasıl formüle edileceėinin analizini yapar. İşletme ve ekonomi alanlarından bakıldığında oyun, ilerleyen zamanlarda ortaya çıkması beklenen ödemeler için önceden tahminde bulunmak zorunda kalan taraflar arasındaki rekabeti ve çıkar çatışmasını ifade eder. Böyle bir ortamda KV'nin en iyi stratejisinin belirlenmesi,yalnızca kendi içerisinde deėerlendirilememekte,diėer oyuncu ya da oyuncuların uygulayabileceėi stratejilere de baėlı olmaktadır[21].



4. ÇOK ÖLÇÜTLÜ KARAR VERME

Günümüzde insanlar ister bireysel olsun ister daha büyük ölçekli olsun bir karar vermeleri gerektiğinde verecekleri karar için birden fazla kriteri göz önünde bulundurmak durumunda kalırlar. Örneğin bir aile, araba satın alma probleminde seçenekler arasında karar vermeye çalışırken fiyat, konfor, yakıt tüketimi, güvenlik, amortisman, estetik vb. gibi kriterleri göz önünde bulundurur. Keza, bir mühendis yeni bir işe girmeyi düşündüğünde yeni iş seçimi için elindeki alternatifleri maaş, konum, titr, yan haklar, çalışma koşulları, kariyer olanakları vb. gibi kriterleri hesaba katarak karar verir.

Sayılabılır sayıda seçeneğin seçilme, sıralarına, sınıflandırına, önceliklendirine veya eleme amacıyla, çoğunlukla ağırlık atanmış, birbirleri ile çelişen ve farklı ölçü birimini kullanan, bazıları nitel değerlerle ifade edilen birçok ölçütten faydalanılarak değerlendirilmesi işlemi Çok Ölçütlü Karar Verme olarak isimlendirilir [36]. Genel olarak seçeneklerin ölçütlere göre performans değerlerini yansıtan bir karar matrisi Çok Ölçütlü Karar Verme Yöntemleri için kullanılır. Çok Ölçütlü Karar Verme, Çok Kriterli Karar Verme (ÇKKV)'nin bir alt başlığıdır.

Çok Kriterli Karar Verme (ÇKKV) , bir KV'nin sayılabılır sonlu veya sayılamaz sayıda alternatif içeren bir küme içerisinde en az iki kriter kullanarak seçim yapması işlemi ya da farklı bir ifadeyle, bir KV'nin iki veya daha fazla kritere dayanarak değerlendirme yapıp alternatifler içerisinde seçim yapması olarak tanımlanabilir[37].

ÇKKV yöntemleri seçenek sayısına göre ikiye ayrılır :

- 3 Çok Amaçlı Karar Verme(ÇAKV)(Multi Objective Decision Making-MODM)
Sonsuz sayıda seçeneğin olduğu sürekli durumlarda alternatiflerin matematiksel programlamayla dolaylı olarak tanımlanması temellidir.
- 4 Çok Ölçütlü Karar Verme(ÇÖKV)(Multi Attribute Decision Making-MADM)
alternatiflerin sonlu sayıda bir liste ile gösterilebildiği kesikli durumlarda karar verme temellidir.[37]

Belirtilen tüm durumlar aşağıda açıklana ortak özellikleri gösterir.

Alternatifler : Tüm alternatifler içerisinde sınırlı sayıda alternatif elenir, önceliklendirilir, seçilir ve/veya sıralanır.

Çok Ölçütlülük : Her problem için birden fazla ölçüt içerisinde ilgili ölçütler ayıklanır. Karar vermek için göz önüne alınması gereken çok sayıda etken olmasına rağmen KV kendine göre önem sırasında öncelikli olanları ölçüt olarak kabul edebilir.

Aynı Birimle Ölçülememe : Her ölçüt aynı ölçüm birimine sahip olmak zorunda değildir.

Ölçüt Ağırlıkları : Neredeyse tüm ÇÖKV yöntemleri her ölçütün sonuç üzerindeki etkisini belirleyebilmek için bilgiye ihtiyaç duyar.

Karar Matrisi : ÇÖKV problemleri, sütunlar ölçütleri; satırlar ise alternatifleri belirtecek şekilde basit bir matris formatında gösterilir.

4.1 Çok Ölçütlü Karar Verme Yöntemlerinin Sınıflandırılması

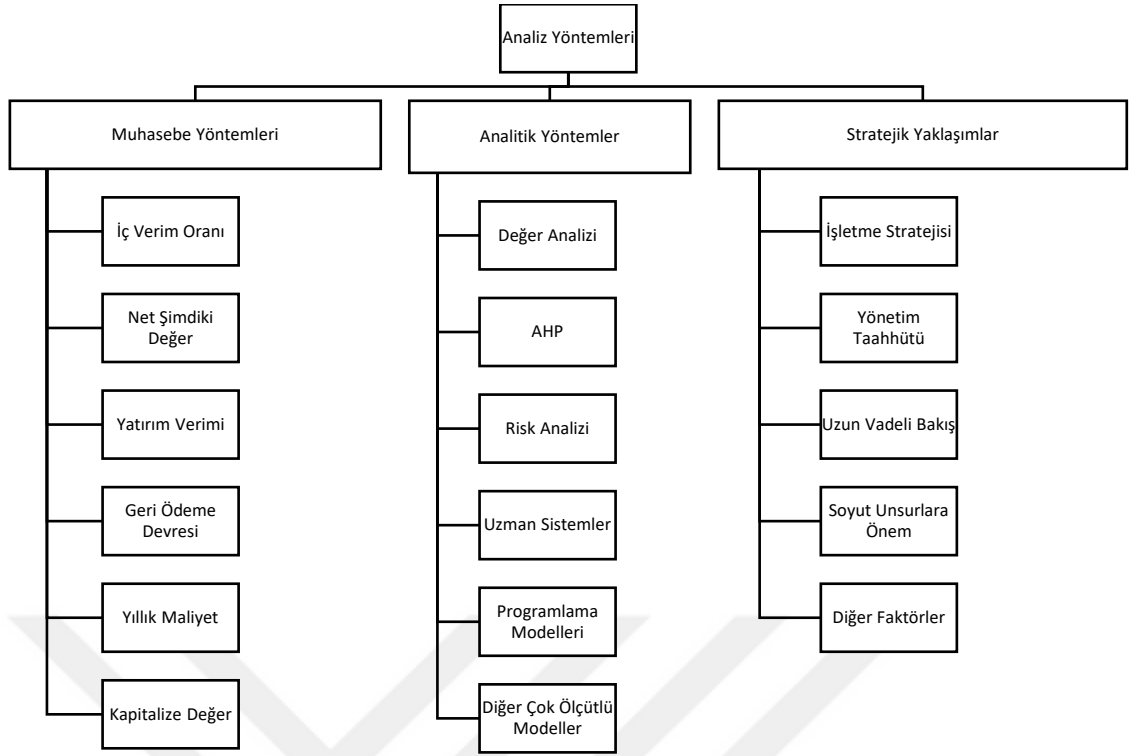
ÇÖKV yöntemleri ile ilgili literatür incelendiğinde birden fazla sınıflandırma ile karşılaşilmektedir. Aşağıda bunların bir kısmı gösterilmiştir.

Meridith ve Suresh (1986);

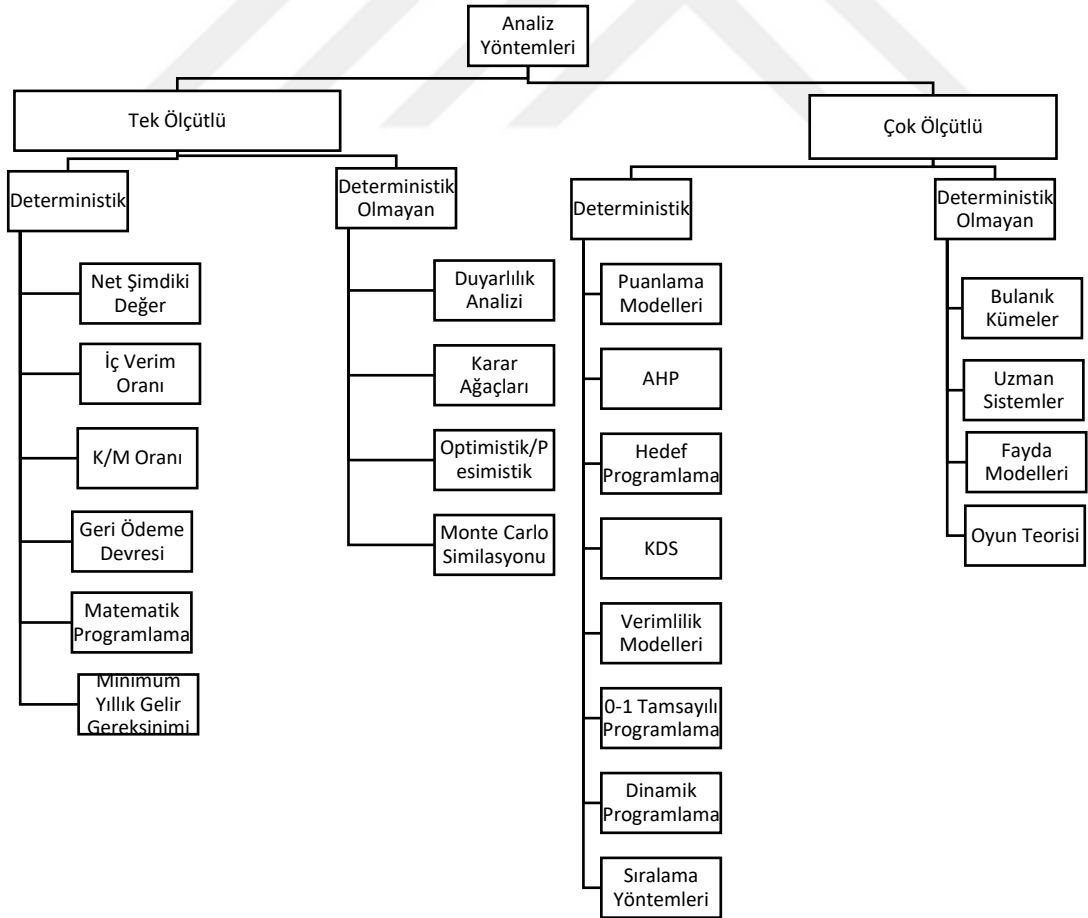
1. İmalat sistemleri için tek başına kullanılabilecek yöntemler,
2. Bağlantılı sistem için uygun yöntemler,
3. Daha stratejik içeriğe sahip ve bütünleşik sistemlere uygulanabilen yöntemler[38].

Lavelle ve Lingett (1992);

1. Muhasebe Yöntemleri,
2. Analitik Yöntemler,
3. Stratejik Yaklaşımlar[39].



Şekil 4. 1 Analiz Yöntemlerinin Sınıflandırılması (Lavelle ve Lingett) [18]



Şekil 4. 2 Analiz Yöntemlerinin Sınıflandırılması (Kolli ve Diğerleri)[40]

4.2 Çok Ölçütlü Karar Verme Yöntemleri

4.2.1 Baskınlık Yöntemi

Bir alternatif, eğer başka herhangi bir alternatif tarafından en az bir ölçütte geçilir ve kalan ölçütler için de eşit olursa, baskın olmayan alternatiftir. Ters açıdan bakıldığında, bir alternatif eğer tüm ölçütler içinde başka bir alternatif tarafından geçilemiyor veya eşit ise, baskındır. Baskın alternatifler ikili karşılaştırmalar ile belirlenir. İlk iki alternatif karşılaştırılır; baskın olmayan gruptan çıkarılır. Sonra çıkmayan alternatifler üçüncü alternatif ile karşılaştırılır ve baskın olmayan alternatif çıkarılır. Sonra dördüncü alternatif vb. şekilde ikili karşılaştırmalar tamamlanır. Eğer karşılaştırmalara m adet alternatifle başlarsak; baskın alternatif (m-1) basarmakta elde edilir.

Hedeflenen çözüm bastırılmış alternatifleri belirlemek olduğu zaman kullanılabilir. Bir bütün olarak bakıldığında bastırılmış alternatifler bastırılmamış olanlardan daha iyi olabilir ve bu yöntemde bu alternatifler kaçırılmış olur.[7]

4.2.2 Birleştiren Yöntem

KV tarafından belirlenen, her ölçüt için karşılanması istenilen minimum değerleri karşılayan alternatifleri "Kabul edilebilir", karşılamayan alternatifleri ise "Kabul edilemez" şeklinde sınıflandırarak alternatif kümesini ikiye bölen yöntem Birleştiren Yöntem olarak adlandırılır. Bu yöntemi uygulamak için, KV, her bir ölçüt için kabul edilebilir minimum ölçüt değerlerini belirler. Eğer bu değerler çok yüksek ise, bütün alternatifler elenir; eğer nispeten düşükse, bir kaç alternatif "Kabul edilebilir" olur. Böylelikle belirlediğimiz minimum standartları azar azar artırarak, alternatiflerin sayısını azaltır ve sonuçta tek bir alternatife ulaşırız. Minimum değerleri geçen alternatifler "Kabul edilebilir" alternatiflerdir.

Telafi esaslı olmayan bu yöntemde seçilebilmek için, her alternatif bütün ölçütlerde belirlenen minimum değerleri geçmek zorundadır. Alternatiflerin sadece bir ölçüte dahi takılmaları elenmelerini gerektirir.

A_i alternatifinin seçilebilir olması için ;

$X_{ij} \geq X_j^0, j=1 \text{ ve } 2 \text{ ve } \dots \text{ve } n \text{ olması gerekir.}$

J^0_j . Ölçüt için minimum kabul edilebilir seviyedir.

Bu yöntemde;diğer ölçüt değeri çok iyi olsa bile sadece bir tane “Kabul edilemez” ölçüt değeri olan alternatif reddedilecektir. Ehliyet almak için tüm testlerden alınması gereken minimum puan olması durumu bu yöntem için bir örnektir.[7]

4.2.3 Ayıran Yöntem

Ayıran yöntemde KV tarafından belirlenen her ölçütün karşılanması istenilen baz düzeylerin en az birini sağlayan alternatif “Kabul edilebilir”, diğer alternatifler “Kabul edilemez” olarak tanımlanarak alternatifler iki kümeye bölünr. Bir alternatif bir veya daha fazla ölçütte baz değeri aşarsa “Kabul edilebilir” sayılır. Bu yöntemde alternatifin herhangi bir ölçütte yeter seviyede olması seçilebilmesi için kafidir.

A_i alternatifinin kabul edilebilir olması için ;

$X_{ij} \geq X_j^0$, $j=1$ veya 2 veya ... veya n olması gerekmektedir.

$J^0 = j$. Ölçüt için minimum kabul edilebilir düzeydir.

Bu yöntem amaçlanan çözüm kabul edilemez seçenekleri belirlemek olduğunda kullanılırç Basit,anlaması ve kullanması kolayd bir yöntemdir. Standartlar çok yüksek belirlenmediği sürece neredeyse tüm alternatiflerin kabul edilebilir olacağı düşünülebilir. Eğer kabul edilebilir olan alternatif olmaz ise KV bir veya daha fazla baz değeri düşürmelidir.[41]

4.2.4 Ardışık Sırasal Yöntem

Alternatifler önem derecesi en yüksek olan ölçüte göre birbirleriyle karşılaştırılarak değerlendirilir. Bir alternatif diğer alternatiflerin herhangi birinden daha yüksek ölçüt değeri alırsa, yüksek ölçüt değeri olan alternatif seçilir ve karar süreci sonlanır.Bunun yanı sıra alternatifler arasında ölçüt değeri birbirlerine eşit olanlar var ise bu alternatifler ikinci önemli ölçüte göre değerlendirilir. Süreç tek bir alternatife ulaşılması veya bütün ölçütlere göre değerlendirme yapılması durumunda sonlanır. Bu yöntem aynı zamanda Sözlük Sırasal Yöntem olarak da bilinir.

X_1 , KV için en önemli ölçüt; X_2 de ikinci en önemli ölçüt olsun.

$$A^1 = \{A_i | \max x_{i1}\} , i=1,2,...,m \quad (4.1)[41a]$$

Eğer, A^1 tek eleman ise,bu eleman en fazla tercih edilen alternatif olur. Eğer farklı alternatifler de mevcutsa;

$$A^2 = \{A^1 \mid \max x_{i2}\}, i \in \{A^1\} \quad (4.2)[41]$$

Eğer A^2 tek eleman ise değerlendirme sonlanır ve bu eleman seçilir. Aksi durumlarda bu süreç tek eleman değerine değeri ulaşmaya kadar veya tüm ölçütler değerlendirilene kadar sürer. Sonuçta birden fazla alternatifin kalması durumunda birbirlerine eşit oldukları kabul edilir.

Kolay bir yöntem olan ardışık sıralama yöntemi, önem derecesi en yüksek olan ölçüte göre iyi durumdaki bir alternatifi diğer tüm ölçütlere göre durumu en kötü olsa bile birinci sıraya koyması nedeniyle eleştirilebilir.[41]

4.2.5 Özelliklerine Göre Eleme Yöntemi

Ardışık sıralama yöntemi ile çok fazla benzerlik gösteren bu yöntem Tversky(1972) tarafından ortaya konulmuştur. Ardışık sıralama yönteminde olduğu gibi alternatifler arasında tek ölçüte göre sıralama yapılır. Standartları sağlayamayan alternatifler elenir ve biri dışında kalan alternatifler eleninceye kadar bu işlem sürer. Bu yöntemi ardışık sıralama yönteminden ayıran kısım ise ölçütlerin önem sırası yerine probabilistik ayırım gücüne göre sıralanmasıdır.

X_1 : En fazla alternatifi eleyecek en etkili birinci özellik

X_2 : En fazla alternatifi eleyecek en etkili ikinci özellik

$$A^1 = \{A_i \mid X_{i1} X_1' \text{'i karşılar}\}, i=1,2,...,m \quad (4.3)[41]$$

Eğer A^1 tek eleman ise en fazla tercih edilen alternatif olur. Eğer başka eşit seviyede alternatif var ise X_2 özelliği de göz önüne alınarak değerlendirme yapılır.

$$A^2 = \{A^1 X_{i2} X_2' \text{'i karşılar}\}, i \in \{A^1\} \quad (4.4)[41]$$

Eğer A^2 tek eleman ise değerlendirme sonlanır ve bu alternatif seçilir. Aksi halde X_3 özelliğine bakılarak devam edilir ve bu süreç tek eleman $\{A^k\}$ değerine ulaşana veya bütün ölçütler değerlendirilene kadar devam eder.

Bu yöntemde KV tüm ölçütler için kabul edilebilir en düşük seviyeleri belirler. Eleme işlemi olasılıksal olarak en yüksek ayırım gücünde olan ölçüt ilk sırada olacak şekilde eleme işlemi yapılır, ölçütün değerini sağlamayan alternatifler reddedilir.

Basit, anlaması ve kullanılması kolay bir yöntem olmakla birlikte diğer ölçüt değerleri çok iyi olmasına rağmen bir tane kabul edilemez ölçüt değeri olan seçeneği reddetmesi dezavantaj oluşturur.[7]

4.2.6 Maximin Yöntemi

Bir zincirin kuvvetini en zayıf halkası belirler. Bu bakış açısıyla bir alternatifi en zayıf veya en güçsüz ölçütü onun toplam performansını tanımlar. Bu durumlarda KV ölçütlerden hangisinin toplam performans üzerinde belirleyici rol oynadığı hakkında yeterli bilgiye sahip değil ise pesimist bir yaklaşımla kötünün iyisini seçmeye çalışır. Ölçütler arasındaki en düşük değerlerin maksimumunu seçmeye “Maximin” veya “Kötünün iyisi” denir.

Bir astronotun uzaya çıktıktan , sonra hayatta kalması onun en güçsüz hayati organıyla bağlantılıdır. Birden fazla astronot arasından seçim yapmaya çalışırken birinin en güçsüz organının karaciğer, diğerlerinin en güçsüz organının ise böbrekleri olduğunu varsayalım. Bu astronotların hayatta kalma ihtimallerini hesaplayabilmek için ortak bir ölçek gerekecektir. Aksi halde böbrek ile karaciğer karşılaştırılmaz. Bu sebepten ötürü ölçütler arası kıyaslama yapabilmek için ortak bir ölçeğe ihtiyaç duyulur.[21]

Maximin yöntemin seçim süreci iki basamaklı bir prosedürden oluşur. İlk basamak her bir alternatif için en düşük ölçüt değerinin belirlenmesi; ikinci basamak ise belirlenmiş olan en düşük ölçüt değerleri arasından en iyi değer seçilmesidir.

Matematiksel olarak göstermek isterseniz;

$$A^+ = \{A, | \max (i) \min(j) r_{ij} \} \quad (4.5)[21]$$

4.5’te yer alan denklemi sağlayan A^+ alternatifi seçilir. Denklemde yer alan r_{ij} , x_{ij} ’nin karşılaştırma ölçeğidir. KV, minimum getirili alternatiflerin en yüksekini tercih ettiği zaman ne olursa olsun getirisinin en az bu miktarda olacağını düşünür ve bu düşünce onu tatmin edebilir.

Tutuculuğu ve karamsarlığı maximin yönteminin zayıf yönleridir. Her alternatif sadece en kötü getirisine göre değerlendirilmekte, diğer getiriler ise göz ardı edilmektedir. En kötü durum olan A^+ dışındaki durumlarda diğer alternatiflerin daha fazla getirisi olsa dahi yok sayılacaktır. Bu yöntemde KV kötümser tabiatlıdır[35].

4.2.7 Maximax Yöntemi

Maximax yöntemi, maximin yönteminin tersine güçsüz ölçütler yerine iyi alternatifler arasından bir alternatifin seçilmesi şeklinde çalışır. Profesyonel futbolcuları düşünürsek; bir futbolcu fark yaratan bir çalım yeteğine, istisnai bir koşu veya topa vuruş yeteneğine göre seçilir. Bu konularda ortalama sayılabilecek veya daha az bir yeteğe sahip olan futbolcu seçilmez.

Maximax yönteminde de yalnızca bir ölçüt alternatifi temsil eder ve diğer ölçütler göz ardı edilir. Bu nedenle maximax yönteminde de maximin yöntemi gibi ölçütler arasında karşılaştırma yapılması ihtiyacı vardır. Ölçütler arası karşılaştırma yapabilmek için ilk olarak her bir alternatifin en iyi ölçütü belirlenir ve sonrasında belirlenen en iyi ölçütler içerisinde maximum değer seçilir.[21]

Matematiksel olarak göstermek isterseniz;

$$A^+ = \{A, | \max (i) \max(j) r_{ij} \} \quad (4.6)[21]$$

Maximax yönteminde de bir seçeneği temsilen sadece tek bir ölçüt kullanılarak diğer ölçütler göz ardı edilir.

4.2.8 Basit Toplamlı Ağırlıklandırma Yöntemi

Basit Toplamlı Ağırlıklandırma Yöntemi yada Ağırlıklı Ortalama Yöntemi, alternatiflerin her birinin, farklı ölçütler baz alınarak ulaşılan performans değerlerini tek boyutlu değer fonksiyonlarının yerine kullanarak ölçütlerin göreceli önemlerine göre ağırlıklandırılıp bir toplam puana ulaşılmasına dayanan çok kriterli bir problem çözme yöntemidir.

Basit Toplamlı Ağırlıklandırma Yöntemi literatürde özellikle tek boyutlu karar problemlerinde en bilinen ve yaygın kullanılan ÇÖKV yöntemidir[21]. Bu yöntemde göre, alternatifin her bir kritere göre puanı sayısal ve tüm kriterler ilgili kriter ağırlıkları ile çarpılır. Elde edilen değerler toplanarak ulaşılan toplam sonuç değerini belirtir. Sonuç değerleri arasında en yüksek değere sahip olan alternatif, en iyi alternatif olarak seçilir.

$$A^*_{WSM} = \max_i \sum_{j=1}^n A_{ij} W_j \quad (4.7)[41]$$

Burada n karar kriteri sayısını, A_{ij} i. Alternatifin j. Kriter bazındaki değerini, W_j de j. Kriterin önem ağırlığını temsil eder. A^*_{WSM} ise en iyi alternatif için ağırlıklı toplam skordur.

Basit Toplamlı Ağırlıklandırma Yönteminde, her ölçütün toplama katkısı diğer ölçütlere bağımlı değildir. Bu sebeple KV'nin yaptığı tercih, herhangi bir ölçüt değerinin hiçbir şart altında diğer ölçüt değerlerinden etkilenmesine müsaade edilmemesini gerektirir[42].

Edwards[43] ve Farmer[44] çalışmaları ölçütler arasındaki bağımsızlık ortadan kalkmadığı sürece Basit Toplamlı Ağırlıklandırma Yöntemi ile gerçek değer fonksiyonuna çok yakın değerlerin elde edilebileceğini göstermiştir.

Basit Toplamlı Ağırlıklandırma Yönteminde normalizasyon yapılırken doğrusal normalizasyon kullanılırsa ideal alternatifin puanları toplamı 1 olacaktır. Diğer alternatiflerden herhangi birinin toplam puanının 1'le arasındaki uzaklık, ilgili alternatifin ideal alternatifle arasındaki mesafe hakkında bilgi verir.

Basit Toplamlı Ağırlıklandırma Yönteminde, eğer ölçütler birbirlerinin tamamlayıcı ise (bir ölçütün değerinin yüksek olması diğer bir ölçütün değerine bağlı ise), hesaplanan puan her ölçütün ayrılabilir faydası varsayımını ihlal etmektedir[41].

4.2.9 Ağırlıklı Çarpım Yöntemi

Basit Toplamlı Ağırlıklandırma Yönteminde ölçüt değerlerinin birbirleri ile toplanabilmeleri için gerekli farklı ölçüm üniteleri normalizasyon yapıldığında boyutsuz hale gelirler. Ancak ölçütler birbirlerine çarpımla bağlıysalar bu tarz bir dönüşüme gerek kalmaz. Ölçütlerin değerlerinin birbirleriyle çarpılması durumunda ölçüt ağırlıkları ilgili ölçütün değerinin üssü şeklini alırlar. Kar ölçütleri pozitif, maliyet ölçütleri ise negatif işaretli üs olarak kullanılırlar.

$$R(A_k / A_l) = \prod_{j=1}^n (a_{kj} / a_{lj})^{W_j} \quad (4.8)[41]$$

Burada n karar kriteri sayısını, a_{ij} i. Alternatifin j. Kriter bazındaki değerini, W_j de j. Kriterin önem ağırlığını temsil etmektedir. Eğer hesaplanan $R(A_k / A_l)$ değeri 1'den büyük çıkıyor ise A_k alternatifinin A_l alternatifine nazaran daha arzu edilir bir alternatif olduğunu söyleriz(maksimizasyon).

Ağırlıklı Çarpım Yöntemi, mantık ve basit hesaplama süreçlerini içermesine rağmen henüz geniş bir kullanım alanına sahip değildir. Ağırlıklı Çarpım Yönteminin

en önemli özelliği Basit Toplamlı Ağırlıklandırma Yöntemi gibi karar matrisi üzerinde normalizasyon yapılmasına gerektirmemesidir. Ağırlıklı Çarpım Yöntemi, Basit Toplamlı Ağırlıklandırma Yöntemi gibi ideal alternatifte olan mesafeyle ilgili KV'yi bilgilendirmez. Ağırlıklı Çarpım Yöntemi, basit olmasının yanı sıra anlaşılması kolay bir yöntemdir. KV'den göreceli önemler dışında bilgi istemez fakat bu yöntem de nicel performans değerlerini kullanamaz[28].

4.2.10 TOPSIS

TOPSIS yöntemi, Hwang ve Yoon'un (1981) çalışmalarının referans alınmasıyla Chen ve Hwang [45] tarafından ortaya konulmuştur. Bu yöntemde, alternatiflerin belirli kriterlere göre ve kriterlerin alabileceği en yüksek ve en düşük değerler arasındaki ideal durumlarına göre karşılaştırılması gerekir[46]. n 'in alternatif sayısını, m 'in kriter sayısını temsil ettiği ÇKKV problemi, m boyutlu uzayda n noktaları ile gösterilebilir. Hwang ve Yoon (1981)'un TOPSIS yöntemini oluşturdukları varsayıma göre; çözüm alternatifi, pozitif ideal çözüm noktasına en kısa mesafede ve negatif ideal çözüm noktasına en uzak mesafededir.

Bu yöntem aşağıdaki adımlardan oluşur.

1.Adım : m alternatif ve n kriterden oluşan karar matrisi oluşturulur. $(X_{ij})_{m \times n}$

Alternatifler	Kriterler			
	k_1	k_2	k_3	k_4
X_1	X_{11}	X_{12}	X_{13}	X_{14}
X_2	X_{21}	X_{22}	X_{23}	X_{24}
X_3	X_{31}	X_{32}	X_{33}	X_{34}

Tablo 4. 1 TOPSIS Karar Matrisi [41]

2.Adım : $(X_{ij})_{m \times n}$ matrisinin normalizasyonu yapılır

$$R = (r_{ij})_{m \times n}$$

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^m x_{kj}^2}}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \quad (4.9)[41]$$

3.Adım : Normalizasyon adımından sonra, karar matrisinde yer alan elemanlar kriterlerin önemleriyle bağlantılı olarak ağırlıklandırılır.

$$T_{ij} = r_{ij} \cdot w_j, \quad i=1,2,\dots,m, \quad j=1,2,\dots,n$$

$$w_j = W_j / \sum_{k=1}^n W_k, \quad j = 1, 2, \dots, n, \quad \sum_{i=1}^n w_i = 1 \quad (4.10)[41]$$

W_j her bir j . kriterin ağırlığıdır.

4.Adım : Pozitif ve negatif ideal çözüm değerleri hesaplanır.

$$A_w = \{ \langle \max(t_{ij} \mid i = 1, 2, \dots, m) \mid j \in J_- \rangle, \langle \min(t_{ij} \mid i = 1, 2, \dots, m) \mid j \in J_+ \rangle \} \equiv \{t_{wj} \mid j = 1, 2, \dots, n\},$$

$$A_b = \{ \langle \min(t_{ij} \mid i = 1, 2, \dots, m) \mid j \in J_- \rangle, \langle \max(t_{ij} \mid i = 1, 2, \dots, m) \mid j \in J_+ \rangle \} \equiv \{t_{bj} \mid j = 1, 2, \dots, n\},$$

$J_+ = \{j = 1, 2, \dots, n \mid j\}$ pozitif çözüme etki eden kriterleri,

$J_- = \{j = 1, 2, \dots, n \mid j\}$ negatif çözüme etki eden kriterleri,

A_w negatif, A_b pozitif ideal çözümü ifade eder. (4.11)[41]

5.Adım : Herbir alternatifin negatif ve pozitif ideal çözüme olan uzaklıkları hesaplanır.

$$d_{iw} = \sqrt{\sum_{j=1}^n (t_{ij} - t_{wj})^2}, \quad i = 1, 2, \dots, m,$$

$$d_{ib} = \sqrt{\sum_{j=1}^n (t_{ij} - t_{bj})^2}, \quad i = 1, 2, \dots, m$$

d_{iw} ve d_{ib} sırasıyla negatif ve pozitif ideal çözüme olan uzaklığı ifade eder(4.12)[41].

6.Adım : Negatif ideal çözüme olan benzerlikler hesaplanır.

$$s_{iw} = d_{iw} / (d_{iw} + d_{ib}), \quad 0 \leq s_{iw} \leq 1, \quad i = 1, 2, \dots, m.$$

$S_{iw} = 1$ sadece ve sadece alternatif çözüm en iyi çözüm ise,

$S_{iw} = 0$ sadece ve sadece alternatif çözüm en iyi çözüm ise. (4.13)[41]

7.Adım : En büyük S_{iw} değerli alternatiften başlayarak alternatifler sıralanır .

A_w , ölçütlerin tümüne göre minimum performansı sağlayan ve gerçekte ulaşılması çok zor olan ideal bir alternatifi temsil ederken, A_b ölçütlerin tümüne göre maksimum performansı sağlayan alternatifi temsil eder.

TOPSIS yöntemi, basit varsayımlara dayanarak KV'nin ideal çözüme yakın bir çözüm bulma isteğini karşılarken, nitel değerlere uygulanamaz. [41]

4.2.11 ELECTRE

1960'ların sonlarına doğru Roy[47] tarafından ortaya konulan ELECTRE Yöntemini, Nijkamp ve Van Delft[48] ve Voogd[49] geliştirerek bugünkü haline getirilmiştir. ELECTRE seçim sorunlarının çözümü için kurgulanmış bir yöntemdir.

Bu yöntem aşağıdaki adımlardan oluşur.[21]

1.Adım : n kritere göre değerlendirilen m adet alternatifin yer aldığı karar matrisinin oluşturulması.

2.Adım : Karar matrisinde normalizasyon işleminin yapılması.

3.Adım : Normalize edilmiş karar matrisinin ağırlıklandırılması.

4.Adım : Uyum kümesinin ve uyumsuzluk kümesinin oluşturulması.

5.Adım : Uyum matrisinin ve uyumsuzluk matrisinin oluşturulduktan sonra uyum ve uyumsuzluk indekslerinin hesaplanması.

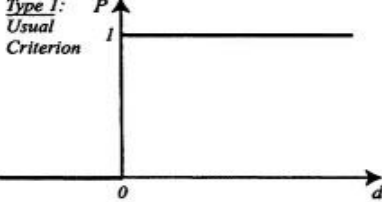
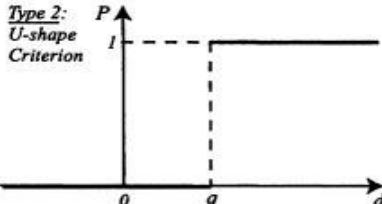
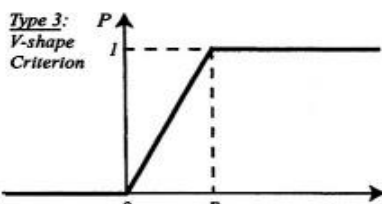
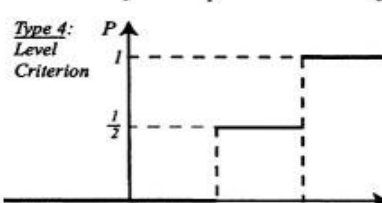
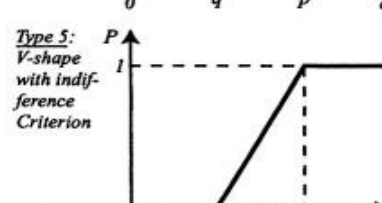
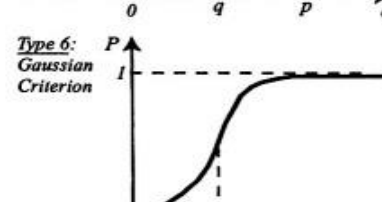
6.Adım : Üstünlük karşılaştırmasının yapılması.,

ELECTRE yönteminin güçlü yönü karar matrisindeki mevcut tüm bilgiyi kullanmasıdır ancak seçenek sayısı arttıkça hesaplama yönteminin ayrıntılı ve uzun olması nedeniyle hesaplama zamanı da çok artmaktadır.

4.2.12 PROMETHEE

PROMETHEE (Prefence Ranking Organization Method for Enrichment Evaluations) alternatifleri seçilen kriterleri kullanarak tercih fonksiyonlarına dayanan ikili karşılaştırmalar yaparak değerlendiren bir ÇÖKV yöntemidir. PROMETHEE fonksiyonları Şekil 4.3'te gösterildiği gibidir.

Bu yöntemde karar verme süreci içerisinde başlangıçta verilen ağırlık puanları değiştirilebilir. Normalizasyon işlemi yapılmasının yanında, ikili karşılaştırmalar yapılırken her bir değerlendirme ölçütü için farklı fonksiyon tiplerinin kullanılabilmesi yöntemin ayırt edici özelliğidir.

Generalised criterion	Definition	Parameters to fix
<i>Type 1:</i> <i>Usual</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ 1 & d > 0 \end{cases}$	—
<i>Type 2:</i> <i>U-shape</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq q \\ 1 & d > q \end{cases}$	q
<i>Type 3:</i> <i>V-shape</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ \frac{d}{p} & 0 \leq d \leq p \\ 1 & d > p \end{cases}$	p
<i>Type 4:</i> <i>Level</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq q \\ \frac{1}{2} & q < d \leq p \\ 1 & d > p \end{cases}$	p, q
<i>Type 5:</i> <i>V-shape</i> <i>with indif-</i> <i>ference</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq q \\ \frac{d-q}{p-q} & q < d \leq p \\ 1 & d > p \end{cases}$	p, q
<i>Type 6:</i> <i>Gaussian</i> <i>Criterion</i> 	$P(d) = \begin{cases} 0 & d \leq 0 \\ 1 - e^{-\frac{d^2}{2s^2}} & d > 0 \end{cases}$	s

Şekil 4. 3 PROMETHEE Fonksiyonları

4.2.13 AHP (Analitik Hiyerarşi Prosesi)

AHP, 1970'li yıllarda Thomas L. Saaty'nin geliştirdiği, karmaşık ÇKKV problemlerinin çözümünde kullanılan bir yöntemdir. Kriterlerin tamamı için göreceli önemlerinin derecelerini belirlerken KV'nin fikirlerine muhtaç olan bir tekniktir. Kriterlerin tümü değerlendirmeye tabi tutularak karar alternatifleri için öncelik sırası elde edilir[50].

AHP'nin diğer ÇÖKV yöntemlerine karşı üstün olduğu en önemli noktalar kolay kullanımı ve objektif yargılar ile birlikte subjektif yargılar da içeren karmaşık karar problemlerinin çözümünde de başarılı bir şekilde uygulanabilir olmasıdır[51].

AHP, kriter ve alt kriterlerin önem seviyelerini belirleyerek çok boyutlu olan problemi tek boyuta indirgemektedir. Öncelik vektörleri kullanılarak saptanan kararlar sıralamayı oluşturmaktadır ve bu da olası alternatifler sonuçlar içerisinde en iyi sonucun elde edilmesini sağlamaktadır. İşlemlerde karar alınırken katkı sağlayacak birçok karar verici bulunabilmektedir. AHP ile tek bir sonuca ulaşmak için farklı tecrübe, bilgi ve eğitim sahibi kişilerin kararları birleştirilir [52].

AHP ile ilgili detaylı bilgi bir sonraki bölümde açıklanmaktadır.

5.BULANIK MANTIK

Günlük hayatımızda kullandığımız kelimelerin bir çoğu bulanık anlamlar içerir. Bir tanımlama ve ifade esnasında kullandığımız kısa, uzun, sıcak, soğuk, uzun, kısa, genç, yaşlı, bulutlu, yağmurlu, hızlı, yavaş vb. gibi kelimeler aslında bulanık anlamlar içerir. Günlük hayat içerisinde birşeyleri tartışırken yada bir aksiyon geliştirmek için karar alırken kesin olmayan,güncel ve bulanık kelimeler kullanırız.

Günlük konuşmalarımızda hava sıcak dediğimiz zaman herkes hava kelimesi ile normal kullanımdaki anlamını kastedtiğimizi anlayacaktır fakat sıcak kelimesi ile anlatılmak istenen karşımızdaki kişiye göre değişebilir. Kutuplarda yaşayan bir insan sıcak kelimesinden 15 °C algımlarken ekvator civarında yaşayan bir insan 35 °C algılayabilir. Buradan da anlaşılacağı üzere sıcak kelimesi belirsizlik içerir. Sıcak kelimesindeki olduğu gibi bazı kelimelerin anlamları belirsizdir ve benzer şekilde kelimelerin içerdikleri belirsizliklere belirsizlik (fuzziness) denir[53]. Bulanık kelimesi, daha genel anlamıyla, iyi tanımlanmış sınırları net olmayan işler kümesi veya gözlemlerle tariflenemediği durumlar için kullanılır[45].

Belirsizlik ve belirsizliğin tanımlanması ile ilgili olarak Einstein “Matematiğin kavramları kesin oldukları sürece gerçeği yansıtmazlar,gerçeği yansıttıkları sürece kesin değillerdir.” demiştir.

Bulanıklık kavramından önce Aristo mantığı, ikili mantık, daha yaygın olarak kullanılırdı. Aristo mantığında olaylar evet-hayır,0-1 vb. gibi iki seçenекle ifade edilmekte olup, belirlenen iki değеr arasında kalanlar kesin olmadıkları için bu değеrlere yer vermez. Basit ve belirli sınırlara sahip koşulların değеrlendirilmesinde klasik mantığa dayanan matematiksel analiz yöntemleri uygulanabilmesine rağmen karmaşık ve kişisel değеrlendirmelerin esas alındığı sistemlerin çözümünde bu yöntemler yetersiz kalmaktadır[54].

Geçmiş zamanlarda belirsizliğin işlenebilmesi için ihtimaller teorisi kullanılmıştır. Belirsizliğin rastgele olduğu kavramı yaygın olarak kabul edilmiştir. Rastgelelik sonuçların ortaya çıkmasında tamamen şansın rol alması ve öngörü ve tahminlerin kesinlik içerecek şekilde önce yapılamaması gibi özellikleri barındırır. Fakat bildiğimiz belirsizliklerin tamamı rastgele karakterde olmak zorunda değildir ve günlük hayatta karşımıza çıkan belirsiz durumlar rastgele olmayabilir[55].

Belirsizliğin etkisinin anlaşılması, ikili mantıktan belirsizliğin baz alındığı anlayışa geçiş için gerekli olan dönüşümün başlamasını sağlamıştır. Belirsizlik temelli yeni anlayışın kullanılmaya başlanmasıyla deterministik anlayışın alışılmış görüşünün aksi yönünde olması sebebiyle soğuk karşılanmıştır.

İlk olarak Dr.Zadeh'in 1965 yılında "Bulanık Kümeler" adlı çalışmasının "Information and Control" dergisinde yayınlanmasıyla ortaya çıkan bulanık mantık, günümüze kadar bir çok araştırmacı, mühendis, bilim insanı ve matematikçinin ilgisini çekmiştir. Bulanık mantık günümüzde otomobillerin fren sistemlerinin farklı hava ve yer şartlarına uygun olarak geliştirilmesinden; deterjan, su ve elektriği çamaşır miktarına ve kirine göre ayarlayan çamaşır makinelerine kadar çok çeşitli alanlarda kullanılmaktadır.

Gerçek dünyada belirsizliğin artması ile karmaşıklık da artmaktadır. Bu belirsizlik ortamında bizler, insanlar, karşılaşılan sorunlara cevap aramakta ve durumlar karşısında aksiyon almak için karar vermek durumundadır. Karmaşık sistemleri anlamak için karmaşık sistemin ortaya koyduğu davranışı anlayabilmek gerekir. Bu şekilde problem hakkında genel bir bilgi sahibi olunabilir. Dr.Zadeh" göre "Gerçek dünya problemlerine ne kadar yakın bakılırsa bulanıklık o kadar çözüm haline gelir."[55].

Bulanık mantık, özellikle anlaşılması kolay olmayan ve yorum temelli karmaşık sistemler ile insan muhakemesi, algılaması veya karar verme olgusunu temel alan süreçlerde çok faydalı olmaktadır.

Bulanık kümeler teorisi asıl gücünü ve önemini, dilsel değişken ve sözel ifadeleri sayısal değişkenlerin yerine kullanmak suretiyle kesin olmayan (imprecise) kavramları temsil etmesinden almaktadır[55]. Günümüzde doğruluk durumları tartışılan önermelerin gelecek olaylarla sınırlı olmadığı anlaşılmıştır. Önermelerin doğruluk dereceleri; doğru için 1, yanlış için 0 ve belirsiz için "sıfır ile bir arasında" değerlerinden birini alabilmektedir. Üçgensel mantığın faydalı ve anlamlı bulunmasıyla birlikte 1930'larda n değerli mantık geliştirilmiştir. n değerli mantığı, üçgensel mantığın genelleştirmesini yaparak ilk defa öneren Lukasiewicz'tir.

N değerli mantığın doğruluk değerlerini içeren T_n kümesi,

$$T_n \{ 0 = \frac{0}{n-1}, \frac{1}{n-1}, \dots, \frac{n-2}{n-1}, \frac{n-1}{n-1} = 1 \} \quad [55]$$

ile gösterilmektedir. Çok değerli mantıkların bulanık küme teorisinde karşılıkları vardır ve bulanık mantığın özünü bunlar oluşturur. Bulanık küme teorisini temel alır. Burada amaç, kesin olarak ifade edilemeyen önermelerin yaklaşık karşılaştırmaları için kurallar belirlemektir. Bulanık mantık, net olmayan ifadelerin (sıcak, soğuk, uzun, kısa, bol, çok, genellikle v.b.) ve göreceli doğruluk ifadelerinin (çok doğru, az doğru v.b.) bulanık değerlerle sayısal olarak ölçülmesi için fırsat yaratır[56].

Bulanık mantığın temel özellikleri Askerzade tarafından aşağıdaki maddeler ile belirtilmiştir :

- Yaklaşık değerlerle düşünme, kesin değerlere göre düşünmenin yerini almıştır
- Bulanık mantıkta tüm değerler $[0,1]$ aralığındadır.
- Bulanık mantıkta bilgi az, kısa, sıcak, soğuk gibi sözel olarak ifade edilir.
- Bulanık çıkarım işlemleri sözel ifadeler arasındaki kurallara dayanılarak yapılır.
- Mantıksal tüm sistemler bulanık şekilde gösterilebilir.
- Bulanık mantık, matematiksel olarak modellemenin çok zor olduğu sistemleri modellemek için çok uygundur.
- Bulanık mantık, tamamı bilinmeyen veya bilgi eksikliği ile işlem yapma yeteneğine sahiptir[57].

Bulanık mantık ile çalışmanın tercih edilme sebepleri :

- Bulanık mantık, basit teorik matematiksel dayanağa sahiptir ve kolay anlaşılır.
- Bulanık mantık yaklaşımının doğal ve karmaşıklıktan ve komplekslikten uzak oluşu mantığı çekici kılmaktadır.
- Bulanık mantık esnek bir yapıdadır.
- Bulanık mantık eksik yada veri ile de işlem yapabilmektedir.
- Karmaşık ve doğrusal olmayan fonksiyonlar da bulanık mantık ile modellenebilmektedir.
- Bulanık mantık kullanılarak uzmanların bilgisinden, tecrübe ve görüşlerinden yararlanılabilir.
- Bulanık mantığın günlük dil ile çalışıyor olması en büyük avantajıdır[58].

5.1 Bulanık Küme Teorisi

İlk olarak 1965 yılında L.Zadeh tarafından ortaya atılan bulanık küme teorisi, kesintisiz üyelik derecelikleri olan bir sınıf olarak tanımlanır [59].

X ,elemanları x ile gösterilen, evren olarak isinlendireceğimiz, klasik bir küme olsun. X 'in bir alt kümesi olan A kümesinin üyelik fonksiyonu, μ_A , A 'dan $\{0,1\}$ 'e , $\{0,1\}$ değer kümesi ,1 üye ve 0 üye değil olmak üzere;

$$\mu_A(x) = \begin{cases} 1 & \text{iff } x \in A \\ 0 & \text{iff } x \notin A \end{cases} \quad (5.1)[60]$$

olarak ifade edilir.

Eğer $[0,1]$ değer kümesi bir aralık olarak ifade edilirse, A kümesi $\tilde{A}$ $[2,3,6,8,57,58,114,133]$ olarak gösterilir. $\mu_{\tilde{A}}(x)$ x elemanının $\tilde{A}$ kümesine üyelik derecesini ifade etmek üzere;

$$\mu_{\tilde{A}} : X \rightarrow [0,1] \quad (5.2)[60]$$

$\mu_{\tilde{A}}(x)$ değerinin 1'e olan yakınlığı, x değerinin $\tilde{A}$ kümesine ne kadar dahil olduğunu gösterir.

$$\tilde{A} = \{ (x, \mu_{\tilde{A}}(x)), x \in X \} \quad (5.3)[60]$$

5.1.1 Bulanık Kümelerin Özellikleri

5.1.1.1 Bulanık Kümelerin Eşitliği

İki bulanık küme $\tilde{A}$ ve $\tilde{N}$, sadece ve sadece

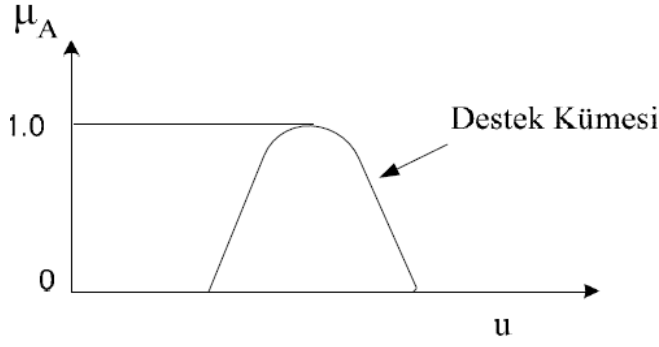
$$\forall x \in X \quad \mu_{\tilde{A}}(x) = \mu_{\tilde{N}}(x) \quad \tilde{A} = \tilde{N} \quad (5.4)[60]$$

ise eşittir.

5.1.1.2 Destek Kümesi

Bir bulanık kümede, kümeye üyelik derecesi 0'dan büyük olan elemanlardan oluşan kümeye destek kümesi denir.

$$\text{supp}\tilde{A} = \tilde{A}^{+0} = \{x \in X, \mu_A(x) > 0\} \quad (5.5)[60]$$



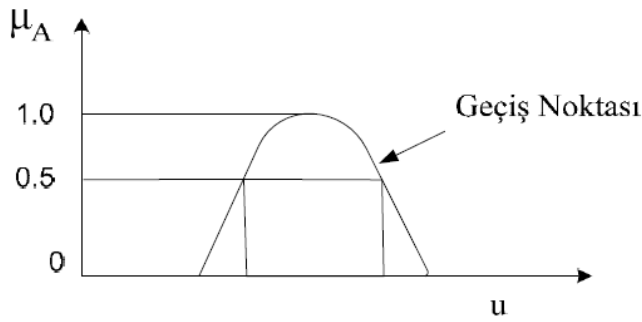
Şekil 5. 1 Bulanık Küme - Destek Kümesi [61]

5.1.1.3 Geçiş Noktası

Bir bulanık kümede, kümeye üyelik derecesi 0.5 olan nokta geçiş noktası olarak isimlendirilir.

$$\mu_{\tilde{A}}(x) = 1/2$$

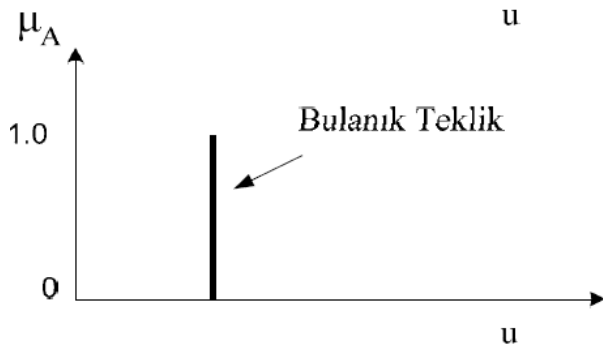
(5.6)[60]



Şekil 5. 2 Bulanık Küme - Geçiş Noktası[61]

5.1.1.4 Bulanık Teklik

EvrenselU kümesi içerisinde yer alan bulanık bir kümenin destek bölgesi tek noktadan ibaret ise buna bulanık teklik (singleton) denir.[60]



Şekil 5. 3 Bulanık Küme – Bulanık Teklik [61]

5.1.1.5 Bulanık Kümenin Yüksekliği

Bulanık bir kümede ulaşılan en yüksek üyelik değeri bulanık kümenin yüksekliği olarak isimlendirilir.

$$\text{hgt}(\tilde{A}) = \sup_{x \in X} \mu_{\tilde{A}}(x) \quad (5.7)[60]$$

5.1.1.6 Normal ve Normal-altı Kümeler

Yükseliği 1'e eşit olan bulanık kümeler normal bulanık küme olarak isimlendirilir.

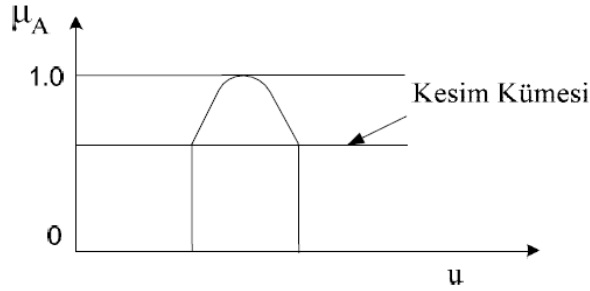
$$\exists x \in X, \mu_{\tilde{A}}(x) = 1 \quad (5.8)[60]$$

Yükseliği 1'e eşit olmayan bulanık kümeler normal-altı bulanık küme olarak isimlendirilir.

5.1.1.7 Bulanık Kümenin α -Kesim Kümesi

Bulanık bir A kümesi içerisinde üyelik derecesi α ' dan büyük olan elemanların oluşturduğu küme α -kesim kümesidir.

$$A^\alpha = \{ x \in X, \mu_{\tilde{A}}(x) \geq \alpha \} \quad (5.9)[60]$$



Şekil 5. 4 Bulanık Küme - α -Kesim Kümesi[61]

5.1.1.8 Bulanık Kümenin Konveksliği

Bir $\tilde{A}$ bulanık kümesi tüm $x_1, x_2 \in R, \lambda \in [0,1]$ değerleri için

$$\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)) \quad (5.10)[60]$$

ise konvektir.

5.1.1.9 Bulanık Kümenin Kardinalitesi

X sonlu bir küme ve $\tilde{A}, X$ üzerinde tanımlı bir bulanık küme ise, $\tilde{A}$ 'nın kardinalitesi

$$|\tilde{A}| = \sum_{x \in \tilde{A}} \mu_{\tilde{A}}(x) \quad (5.11)[60]$$

şeklinde gösterilir.

Eğer X sonsuz bir küme ise

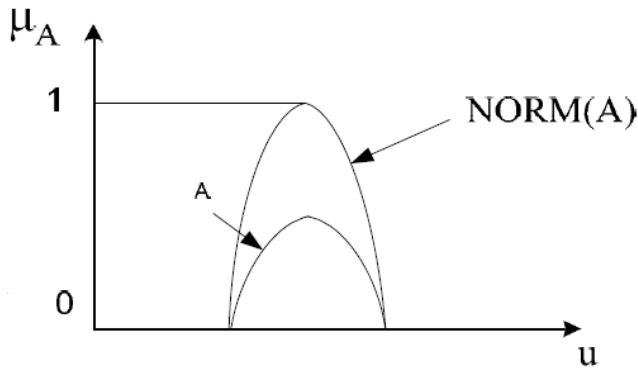
$$|\tilde{A}| = \int \mu_{\tilde{A}}(x) dx \quad (5.12)[60]$$

şeklinde gösterilir.

5.1.1.10 Bulanık Kümenin Normalizasyonu

Bulanık kümede yer alan her bir elemanın üyelik derecesinin, kümede yer alan en büyük üyelik dereceli elemanın derecesine bölünmesiyle normalize edilen küme yeniden ölçeklendirilmiş olur.

$$\mu_{\text{NORM}} = \mu_A(x) / \max(\mu_A(x)) \quad \forall x \in U \quad (5.13)[60]$$



Şekil 5. 5 Bulanık Küme – Normalizasyon[61]

5.1.1.11 Bulanık Kümenin Çekirdeği

Bulanık kümeye üyelik derecesi 1 olan elemanlardan oluşa kümeye denir.

$$\mu_{\tilde{A}}(x) = 1 \quad (5.14)[60]$$

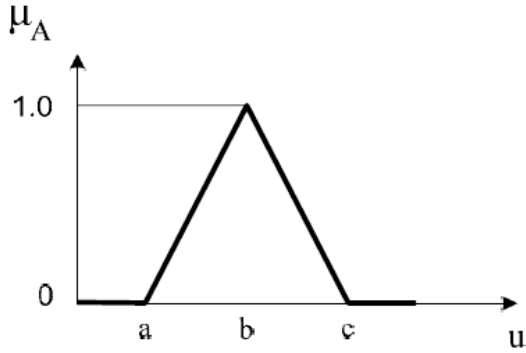
5.1.2 Bulanık Üyelik Fonksiyonları

Bulanık sayıların üyelik fonksiyonları “Üçgen Üyelik Fonksiyonu”, ”Yamuk Üyelik Fonksiyonu”, ”S Üyelik Fonksiyonu” ve “Çan Eğrisi Üyelik Fonksiyonu” olmak üzere 4 farklı şekilde tanımlanır.

5.1.2.1 Üçgen Üyelik Fonksiyonu

Üyelik fonksiyonu 5.15 ile verilen $\tilde{A}$ bulanık sayısı, üçgensel bir bulanık sayıdır(ÜBS). Burada a minimum olası değeri, b en umut verici değeri, c ise maximum olası değeri ifade edecek şekilde $a \leq b \leq c$ 'dir. Üçgensel olarak ifade edilen sayılar (a, b, c) şeklinde gösterilir.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & , \quad x < a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ \frac{c-x}{c-b} & , \quad b \leq x \leq c \\ 0 & , \quad x > c \end{cases} \quad (5.15)[60]$$

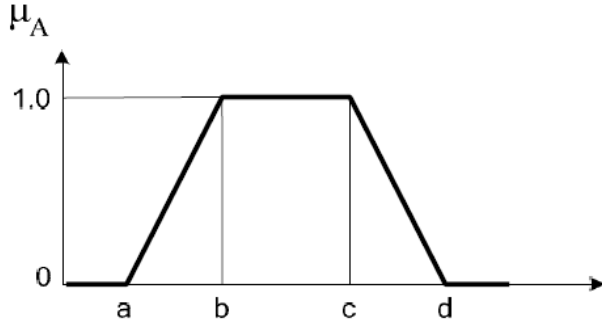


Şekil 5. 6 Bulanık Sayılar – Üçgen Üyelik Fonksiyonu[61]

5.1.2.2 Yamuk Üyelik Fonksiyonu

Üyelik fonksiyonu 5.16 ile verilen $\tilde{A}$ bulanık sayısı, yamuk bir bulanık sayıdır(YBS). Burada $a \leq b \leq c \leq d$ 'dir. Yamuk bulanık sayı olarak ifade edilen sayılar (a, b, c, d) şeklinde gösterilir.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & , \quad x < a \\ \frac{x-a}{b-a} & , \quad a \leq x < b \\ 1 & , \quad b \leq x \leq c \\ \frac{d-x}{d-c} & , \quad c < x \leq d \\ 0 & , \quad x > d \end{cases} \quad (5.16)[60]$$

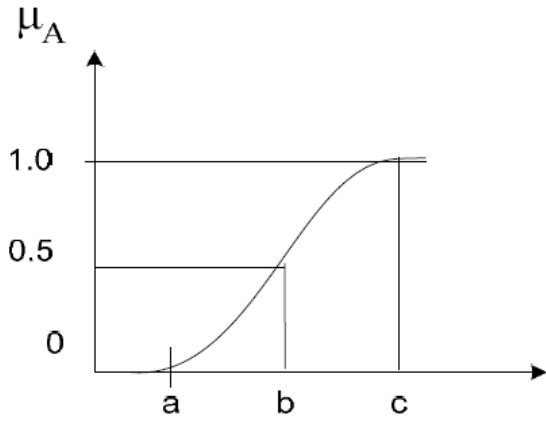


Şekil 5. 7 Bulanık Sayılar – Yamuk Üyelik Fonksiyonu[61]

5.1.2.3 S Üyelik Fonksiyonu

Üyelik fonksiyonu 4.17 ile verilen $\tilde{A}$ bulanık sayısı, s şeklinde bir bulanık sayıdır(SBS). Burada $a \leq b \leq c$ 'dir.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & , \quad x < a \\ 2[(x-a)/(c-a)]^2 & , \quad a \leq x \leq b \\ 1 - 2[(x-c)/(c-a)]^2 & , \quad b \leq x \leq c \\ 1 & , \quad x > c \end{cases} \quad (5.17)[60]$$

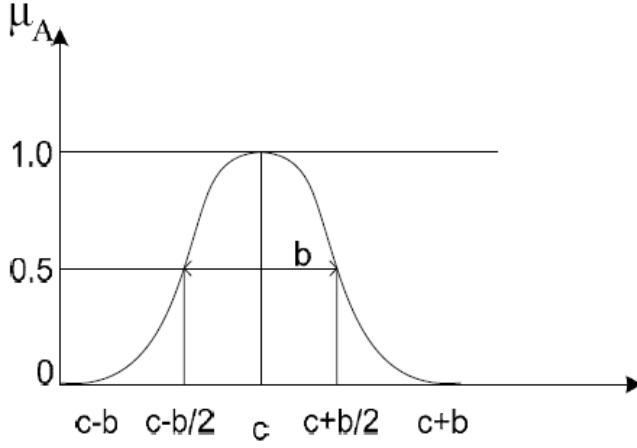


Şekil 5. 8 Bulanık Sayılar – S Üyelik Fonksiyonu[61]

5.1.2.4 Çan Eğrisi Üyelik Fonksiyonu

Üyelik fonksiyonu 5.18 ile verilen $\tilde{A}$ bulanık sayısı, çan eğrisi şeklinde bir bulanık sayıdır(ÇEBS).

$$\mu_{\tilde{A}}(x) = \begin{cases} S\left(x; c - b, c - \frac{b}{2}, c\right) & , x \leq c \\ 1 - S\left(x; c, c + \frac{b}{2}, c + b\right) & , x \geq c \end{cases} \quad (5.18)[60]$$



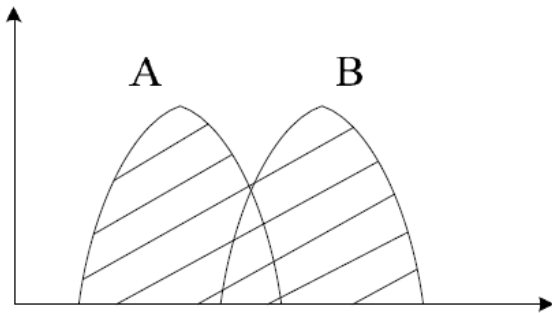
Şekil 5. 9 Bulanık Sayılar – Çan Eğrisi Üyelik Fonksiyonu[61]

5.2 Bulanık Küme İşlemleri

5.2.1 Bulanık Kümelerde Birleşim İşlemi

İki bulanık kümenin birleşimi 5.19 ile verilen üyelik fonksiyonu ile ifade edilir.

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad \forall x \in U \quad (5.19)[60]$$

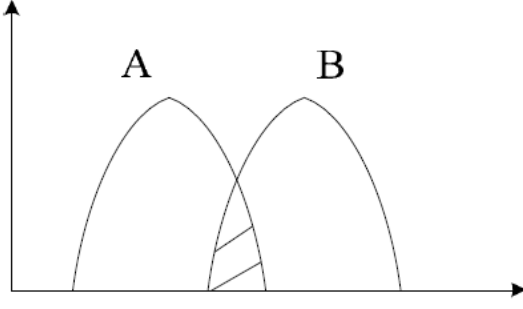


Şekil 5. 10 Bulanık Kümelerde Birleşim İşlemi[61]

5.2.2 Bulanık Kümelerde Kesişim İşlemi

İki bulanık kümenin kesişimi 5.20 ile verilen üyelik fonksiyonu ile ifade edilir.

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad \forall x \in U \quad (5.20)[60]$$

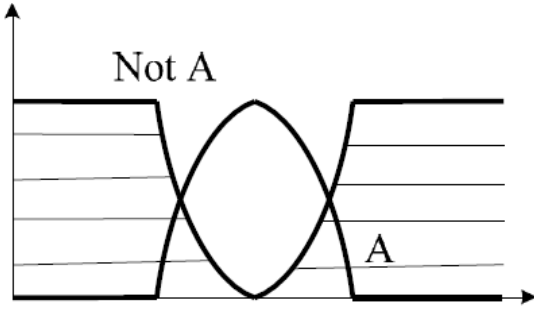


Şekil 5. 11 Bulanık Kümelerde Birleşim İşlemi[61]

5.2.3 Bulanık Kümelerde Tümleyen İşlemi

Bulanık bir A kümesinin normalize edilmiş halinin tümleyen bulanık kümesi aynı evrensel kümede 5.21 ile verilen üyelik fonksiyonu ile tanımlanır.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U \quad (5.21)[60]$$



Şekil 5. 12 Bulanık Kümelerde Birleşim İşlemi[61]

5.3 Üyelik Fonksiyonlarını Değiştiren Bulanık Küme İşlemleri

Bulanık kümelerin üyelik fonksiyonunun şekli aşağıdaki başlıklarda açıklanan işlemler ile değiştirilebilir.

5.3.1 Kuvvet

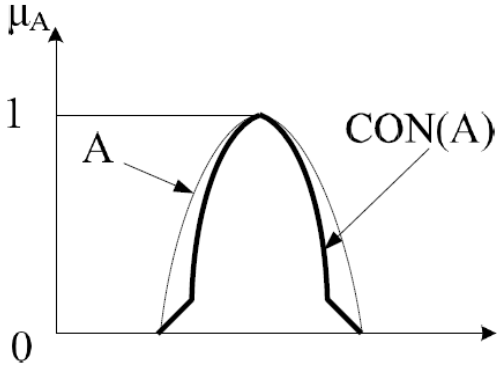
Eğer p pozitif sayılar kümesinden bir sayı ve A, $\mu_A(x)$ üyelik fonksiyonu ile tanımlanmış bir bulanık küme ise: A'nın p. kuvveti 5.22'deki şekilde tanımlanır:

$$A^p = \{(x, \mu_A(x))^p = \{(x, (\mu_A(x))^p\} \quad (5.22)[60]$$

5.3.2 Derişme

Eğer bir bulanık kümenin üyelik fonksiyonu, daha yüksek üyelik dereceli elemanlarının üyeliğine göre değiştirilirse yoğunlaşır.

$$\mu_{\text{CON}(A)}(x) = (\mu_A(x))^2 \quad \forall x \in U \quad (5.23)[60]$$

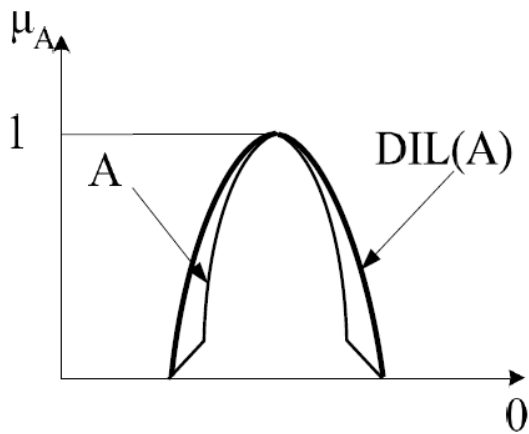


Şekil 5. 13 Bulanık Kümelerde Derişme İşlemi[61]

5.3.3 Genişleme

Eğer bir bulanık kümenin üyelik fonksiyonu, daha düşük üyelik dereceli elemanlarının üyeliğine göre değiştirilirse yoğunlaşır.

$$\mu_{\text{DIL}(A)}(x) = (\mu_A(x))^{0.5} \quad \forall x \in U \quad (5.24)[60]$$

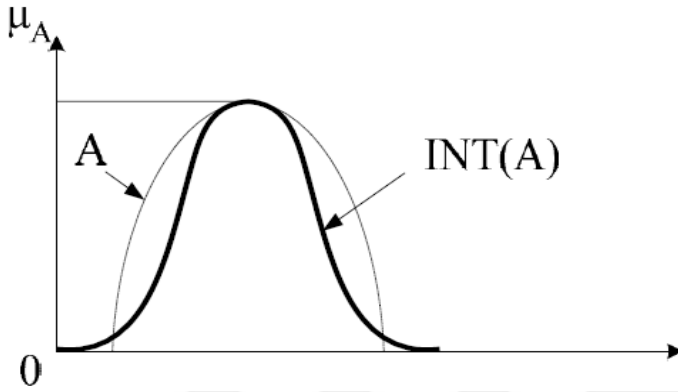


Şekil 5. 14 Bulanık Kümelerde Genişleme İşlemi[61]

5.3.4 Yoğunlaşma

Normalize edilmiş bulanık bir kümenin, üyelik derecesi 0.5'den büyük olan değerlerinin artırılması ve 0.5'den küçük olan değerlerinin azaltılması suretiyle keskin küme olmaya yakınlaştırılmasıdır.

$$\mu_{\text{INT}(A)}(x) = \begin{cases} 2(\mu_A(x))^2, & 0 \leq \mu_A(x) \leq 0.5 \\ 1 - 2(1 - \mu_A(x))^2, & 0.5 \leq \mu_A(x) \leq 1 \end{cases} \quad (5.25)[60]$$



Şekil 5. 15 Bulanık Kümelerde Yoğunlaşma İşlemi[61]

5.4 Bulanık Sayılarda Aritmetik İşlemler

Bu kısımda bulanık sayılar ile yapılan aritmetik işlemler incelenmiştir. Bulanık sayılarla aritmetik işlemler, yaklaşık işlemler, güven aralıkları ile aritmetik işlemler ve üyelik fonksiyonları ile aritmetik işlemler şeklinde üç başlık altında incelenmektedir.

5.4.1 Bulanık Sayılarda Yaklaşık İşlemler

Yaklaşık aritmetik işlemler, üçgensel bulanık ve yamuk bulanık sayılar üzerinde gösterilmiştir.

5.4.1.1 Üçgensel Bulanık Sayılarda Yaklaşık Aritmetik İşlemler

$A = (a_1, a_2, a_3)$ ve $B = (b_1, b_2, b_3)$ olarak tanımlanmış üçgen bulanıklar sayı olsun.

5.4.1.1.1 Toplama İşlemi

$A (+) B = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$ olarak gösterilir ve işlemin sonucu da üçgensel bir bulanık sayıdır.[7]

5.4.1.1.2 Çıkarma İşlemi

$A (-) B = (a_1 - b_3, a_2 - b_2, a_3 - b_1)$ olarak gösterilir ve işlemin sonucu yine üçgensel bir bulanık sayıdır. [7]

5.4.1.1.3 Çarpma İşlemi

$A (.) B = (a_1 . b_1, a_2 . b_2, a_3 . b_3)$ olarak gösterilir ve işlemin sonucu yine üçgensel bir bulanık sayıdır. [7]

5.4.1.1.4 Bölme İşlemi

$A (:) B = (a_1 : b_3, a_2 : b_2, a_3 : b_1)$ olarak gösterilir ve işlemin sonucu yine üçgensel bir bulanık sayıdır. [7]

5.4.1.1 Yamuk Bulanık Sayılarda Yaklaşık Aritmetik İşlemler

$A = (a_1, a_2, a_3, a_4)$ ve $B = (b_1, b_2, b_3, b_4)$ olarak tanımlanmış yamuk bulanık sayılar olsun.

5.4.1.1.1 Toplama İşlemi

$A (+) B = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4)$ olarak gösterilir ve işlemin sonucu yamuk bir bulanık sayıdır [7].

5.4.1.1.2 Çıkarma İşlemi

$A (-) B = (a_1 - b_4, a_2 - b_3, a_3 - b_2, a_4 - b_1)$ olarak gösterilir ve işlemin sonucu yamuk bir bulanık sayıdır [7].

5.4.1.1.3 Çarpma İşlemi

$A (.) B = (a_1 . b_1, a_2 . b_2, a_3 . b_3, a_4 . b_4)$ olarak gösterilir ve işlemin sonucu yamuk bir bulanık sayıdır [7].

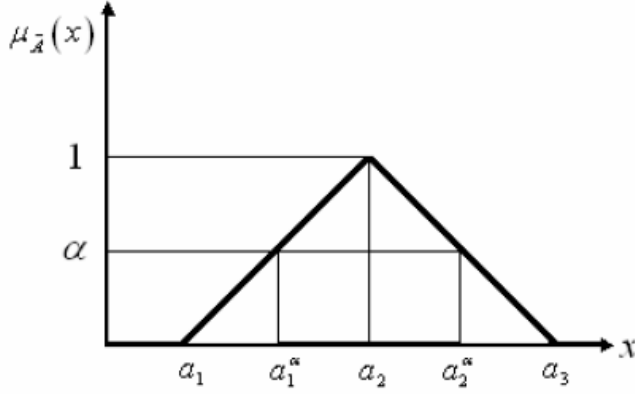
5.4.1.1.4 Bölme İşlemi

$A (:) B = (a_1 : b_4, a_2 : b_3, a_3 : b_2, a_4 : b_1)$ olarak gösterilir ve işlemin sonucu yamuk bir bulanık sayıdır [7].

5.4.2 Bulanık Sayılarda Güven Aralıkları İle Aritmetik İşlemler

Bulanık sayılarda güven aralıkları ile yapılan işlemler kesin bilgiyi taşıdıkları için elde edilen sonuçlar da kesindir.

A bir bulanık küme olmak üzere; $\alpha \in [0,1]$ ve $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$ kümesi A bulanık kümesinin α seviyesinde güven aralığıdır. A_α güven aralığı Şekil 5.16'da gösterilmiştir.



Şekil 5. 16 Üçgensel Bulanık Sayılarda Güven Aralığı[62]

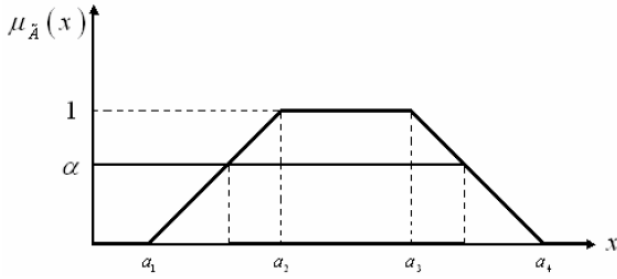
A_α güven aralığı:

$$A_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1) \alpha + a_1, - (a_3 - a_2) \alpha + a_3] \quad (5.24)[62]$$

$A = (a_1, a_2, a_3, a_4)$ yamuk bulanık sayı olsun. Bu yamuk bulanık sayının A_α güven aralığı Şekil 5.17’de gösterilmektedir.

A_α güven aralığı:

$$A_\alpha = [a_1^\alpha, a_2^\alpha] = [(a_2 - a_1) \alpha + a_1, - (a_4 - a_3) \alpha + a_4] \quad (5.25)[62]$$



Şekil 5. 17 Yamuk Bulanık Sayılarda Güven Aralığı[62]

5.4.2.1 Toplama İşlemi

$A, B \subset R$ olmak üzere ve $\alpha \in [0,1]$ seviyesindeki güven aralığı kullanılarak

$$A_\alpha = \{x | \mu_A(x) \geq \alpha\}$$

$$B_\alpha = \{x | \mu_B(x) \geq \alpha\}$$

olarak tanımlanmış A_α ve B_α bulanık sayılarının toplama işlemi

$$A_\alpha + B_\alpha = [a_1^\alpha + b_1^\alpha, a_2^\alpha + b_2^\alpha] \quad (5.26)[62]$$

şekinde tanımlanır.

5.4.2.2 Çıkarma İşlemi

$A, B \subset R$ olmak üzere ve $\alpha \in [0,1]$ seviyesindeki güven aralığı kullanılarak

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$$

$$B_\alpha = \{x \mid \mu_B(x) \geq \alpha\}$$

olarak tanımlanmış A_α ve B_α bulanık sayılarının çıkarma işlemi

$$A_\alpha - B_\alpha = [a_1^\alpha - b_2^\alpha, a_2^\alpha - b_1^\alpha] \quad (5.27)[62]$$

şekinde tanımlanır.

5.4.2.3 Çarpma İşlemi

Bulanık sayılarda çarpma işlemi yalnızca pozitif reel sayılar için tanımlanabilir. A ve B pozitif reel sayılar kümesinde tanımlı bulanık sayılar ve bunların α güven aralıkları;

$$A_\alpha = [a_1^\alpha, a_2^\alpha] \text{ ve } B_\alpha = [b_1^\alpha, b_2^\alpha] \text{ olarak tanımlanmış olsun.}$$

A ve B bulanık sayılarının güven aralıkları kullanılarak çarpımı;

$$A_\alpha (.) B_\alpha = [a_1^\alpha \cdot b_1^\alpha, a_2^\alpha \cdot b_2^\alpha] \quad (5.28)[62]$$

şeklinde tanımlanır.

5.4.2.4 Bölme İşlemi

Bulanık sayılarda bölme işlemi yalnızca pozitif reel sayılarda tanımlı olup $b_1^\alpha > 0$ ve $b_2^\alpha > 0$ olmak üzere $\alpha \in [0,1]$ için güven aralığı cinsinden

$$A_\alpha (:) B_\alpha = \left[\frac{a_1^\alpha}{b_2^\alpha}, \frac{a_2^\alpha}{b_1^\alpha} \right] \quad (5.29)[62]$$

şeklinde tanımlanır.

5.4.3 Bulanık Sayılarda Üyelik Fonksiyonları İle Yapılan Aritmetik İşlemler

$\forall x, y, z \in R$ için üyelik fonksiyonlarını kullanarak yapılan aritmetik işlemler aşağıdaki şekilde tanımlanmıştır.

5.4.3.1 Toplama İşlemi

$$\mu_{A(+)B}(Z) = \vee_{z=x+y} (\mu_A(x) \wedge \mu_B(y)) \quad (5.30)[62]$$

5.4.3.2 Çıkarma İşlemi

$$\mu_{A(-)B}(Z) = \vee_{z=x-y} (\mu_A(x) \wedge \mu_B(y)) \quad (5.31)[62]$$

5.4.3.3 Çarpma İşlemi

$$\mu_{A(.).B}(Z) = \bigvee_{z=x.y} (\mu_A(x) \wedge \mu_B(x)) \quad (5.32)[62]$$

5.4.3.4 Bölme İşlemi

$$\mu_{A(.).B}(Z) = \bigvee_{z=x/y} (\mu_A(x) \wedge \mu_B(x)) \quad (5.333)[62]$$

5.5 Bulanık Sayılarda Sıralama İşlemleri

Bulanık sayıların belirsiz, kesin tariflenemeyen ortamlarda, net olmayan değerleri sayısallaştırabilmek için kullanılması nedeniyle bulanık sayılar içerisinde kıyaslama ya da sıralama yapılabilmesi önem arz etmektedir. Bulanık sayılarda sıralama veya derecelendirilme bulanık optimizasyon ve karar verme yöntemlerindeki problemlerin başında gelir.

Literatürde bulanık sayıların sıralanması ile ilgili pek çok çalışma mevcuttur. Sıralama yöntemlerini Chen ve Hwang (1992) aşağıdaki şekilde gruplamıştır:

- Eniyileme derecesi kullanan sıralama yöntemleri
- Hamming mesafesi kullanarak sıralama yöntemleri
- Alfa kesitleri kullanan sıralama yöntemleri
- Kıyaslama fonksiyonu kullanan sıralama yöntemleri
- Bulanık ortalama ve yayılma kullanan sıralama yöntemleri
- Sol ve sağ değerleri kullanan sıralama yöntemleri
- Merkez endeksi kullanan sıralama yöntemleri
- Alan ölçüsü kullanan sıralama yöntemleri
- Sözel sıralama yöntemleri[45]

Prodanovic ve Simonovic (2002) ise sıralama yöntemlerini aşağıdaki özelliklere göre incelemiştir

- Sıralama yöntemi sadece üçgensel veya yamuk bulanık sayıları değil, değişik şekillerde bulanık kümeleri de sıralayabilmelidir
- Sıralama yöntemi normal olmayan ve konveks olmayan bulanık kümeleri de sıralayabilmelidir
- Sıralama yöntemi sadece iki bulanık seçeneği kıyaslamak veya bir listeden en iyi olanı seçmek yerine birçok bulanık kümeyi sıralayabilmelidir.
- Hangi seçeneklerin en çok tercih edilebilir olduğunu gösteren bir sayısal tercih ilişkisi olmalıdır.

- Sıralama seçenekleri için bir sözel yorum olabilmelidir
- Tercih ilişkisi rasyonel olmalıdır[63]

Bulanık sayıların sıralanmasında kullanılan Liou ve Wang Yöntemi ile Abdel-Kader ve Dugdale'ın Yöntemi ile ilgili detaylar aşağıda verilmiştir.

5.5.1 Liou ve Wang Yöntemi

Liou ve Wang yönteminde;

$\alpha \in [0,1]$ iyimselik endeksi olmak üzere;

$\tilde{A} = (a,b,c)$ olarak verilen ÜBS için, toplam entegral değer:

$$I_T^a(\tilde{A}) = \frac{1}{2}a(b+c) + \frac{1}{2}(1-a)(a+b) = \frac{[ac+b+(1-a)a]}{2} \quad (5.33)[64]$$

$\tilde{A}_i$ ve $\tilde{A}_j$ ÜBS için değer;

$$I_T^a(\tilde{A}_i) < I_T^a(\tilde{A}_j) \Rightarrow \tilde{A}_i < \tilde{A}_j \quad (4.34)[64]$$

$$I_T^a(\tilde{A}_i) = I_T^a(\tilde{A}_j) \Rightarrow \tilde{A}_i = \tilde{A}_j \quad (5.35)[64]$$

$$I_T^a(\tilde{A}_i) > I_T^a(\tilde{A}_j) \Rightarrow \tilde{A}_i > \tilde{A}_j \quad (5.36)[64]$$

$\tilde{A} = (a,b,c,d)$ olarak verilen bir YBS için, toplam entegral değer:

$$I_T^a(\tilde{A}) = \frac{1}{2}[a(c+d) + (1-a)(a+b)] \quad (5.37)[64]$$

5.5.2 Abdel-Kader ve Dugdale'ın Yöntemi

Abdel-Kader ve Dugdale yönteminde;

$\alpha \in [0,1]$ iyimselik endeksi olmak üzere;

$\tilde{A}_1 = (a_1,b_1,c_1)$, $\tilde{A}_2 = (a_2,b_2,c_2)$, $\tilde{A}_3 = (a_3,b_3,c_3)$ olarak verilen ÜBS için;

$S = (a_1,b_1,c_1, a_2,b_2,c_2, a_3,b_3,c_3)$ ve

$V(\tilde{A}_k)$ ise $\tilde{A}_k$ 'nın değeri olmak üzere

$$V(\tilde{A}_k) = (b_k) \left\{ (\alpha) \left[\frac{ck - X_{min}}{X_{max} - X_{min} + ck - bk} \right] + (1 - \alpha) \left[1 - \frac{X_{max} - ak}{X_{max} - X_{min} + bk - ak} \right] \right\} \quad (5.38)[65]$$

$$X_{min} = \inf S \quad (5.39)[65]$$

$$X_{max} = \sup S \quad (5.40)[65]$$

6. ANALİTİK HİYERARŞİ PROSESİ (AHP)

Saaty'nin 1977 yılında geliştirerek literatüre kazandırdığı bir yöntem olan AHP, insan yargılarını karar süreçlerine entegre edebilen bir yöntemdir[66].

AHP yöntemi, karar verici grup veya bireylere, süreç içerisindeki kalitatif ve kantitatif etkenleri birleştirebilme olanağı sağlayan güçlü ve anlaşılması kolay bir yöntemdir[67,68].

Saaty'nin ortaya attığı AHP yönteminin ortaya çıkışı 1970'li yıllara uzanmaktadır. İlk kullanımı 1972 yılında, elektriğin firmalara ülkedeki refaha verdikleri katkı oranında dağıtılması amacıyla ABD'de kullanılmıştır. 1973 yılında "Sudan Ulaşım Projesi"nde kullanılarak olgulaşan AHP, 1974-1978 yılları arasında gelişmiştir.

AHP, insanoğlunun herhangi bir şekilde kendisine öğretilmemesine rağmen içgüdüsel olarak uyguladığı karar verme sistemine dayanır. Yalnızca bir kısmını kontrol edebileceğiniz fazla sayıda ve ilişkili öğeleri bazı özelliklerine göre gruplamaya çalışın. AHP de insanların doğuştan sahip olduğu bu gruplama işlemi sistemleştirir. Bu gruplar farklı özelliklerine göre tekrar gruplanarak bizim asıl amacımız olan karar verme seviyesine gelene kadar tekrarlanır[33].

Diğer bir deyişle AHP, aynı anda farklı etkenleri değerlendirip sayısal değişim işlemleri ile sonuç elde etmeye çalışan, hem parçadan bütüne hem de bütünden parçaya doğru tasarlamaya imkan veren ve doğrusal olmayan bir ÇKKV yöntemidir. AHP, karar verme esnasında bireyin dışındaki tarafsız, elle tutulur fiziksel etmenler ve bireyin kendine veya toplumun tamamına ait subjektif fikir, duygu ve inanışlarından oluşan soyut etmenleri de sürece dahil edebilmesi sayesinde diğer çoğu karar verme yönteminden daha gerçekçidir[69].

AHP'nin büyük bir ilgi görerek geniş alanlarda kullanılması, kolay anlaşılır olması ve hesaplamalarının basit olması sayesinde olmuştur. AHP yöntemi, politik adaylardan askeri araç seçimlerine kadar çok çeşitli seçim problemlerinin çözümünde başarıyla uygulanmıştır.

AHP'nin hiyerarşik ağaç yapısı minimum üç seviyeden oluşur: İlk seviyede(en üstte) hedef, ikinci seviyede kriterler, üçüncü seviyede de alternatifler yer alır. Eğer kriterler yeterince somut veya sınırları belirli değil ise ikinci seviye kendi içerisinde

birden fazla kademe içerebilir. Öznel karşılaştırma, birden fazla KVyi hesaba katabilme, tercihlerin tutarlı oluşu gibi özellikler AHP'yi kullanışlı hale getiren karakteristikleridir [70].

AHP'nin pratik uygulaması Canada ve Sullivan(1996) tarafından beş temel adımla ifade edilmiştir:

1. Karar hiyerarşisinin yapılandırılması
2. Verilerin birebir karşılaştırmalar ile toplanması
3. Nesnel yargıların tutarlılığının kontrol edilmesi
4. Ağırlıkların hesabı için özvektör yönteminin kullanılması
5. Karar alternatiflerinin sıralamasının belirlenmesi için ağırlıkların bir araya getirilmesi[71]

Saaty ise AHP yöntemini 6 adımda ifade etmiştir;

1. Karar verilecek problemin tanımlanması
2. Kriterlerin birbirleriyle ikili karşılaştırma matrisinin oluşturulması
3. Kriterlerin önem derecelerinin belirlenmesi
4. Kriterler arası kıyaslamalardaki tutarlılığın ölçümü
5. Kriterlerin herbiri için m karar noktasındaki yüzde önem dağılımlarının belirlenmesi
6. Karar noktalarındaki sonuç dağılımının bulunması[72]

6.1 Hiyerarşik Yapı

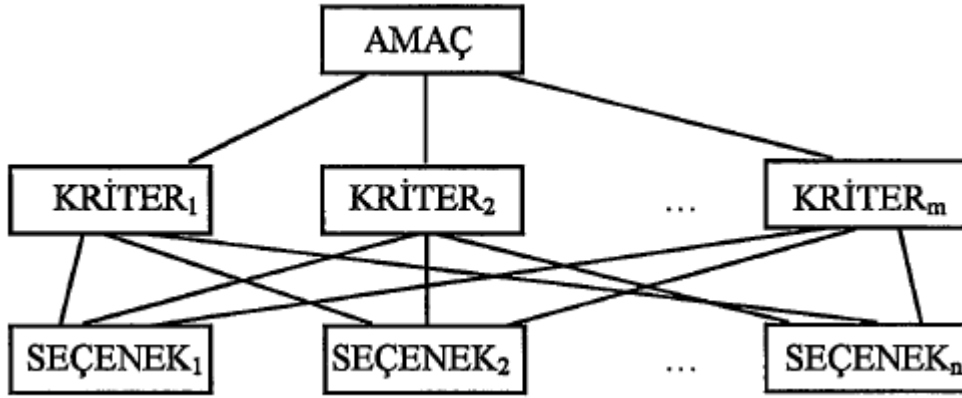
Gün geçtikçe bir sistem içerisindeki öğelerin sistemin tamamı üzerindeki etkisini değerlendirerek, öğelerin göreceli önemlerini bulmaya yönelik çalışan sistem yaklaşımının sosyal ve davranış bilimlerindeki önemi anlaşılmakta olup, AHP bu yaklaşımı temel alan süreçler içermektedir. Sistemler, bileşenlerinin yapısı, bileşenlerin her birinin işlevi, bileşenlerinin hizmet ettikleri amaçlar ve bu amaçların parçası oldukları daha üst seviye amaçlar ile sistemin nihai amacı saptanmaya çalışılarak incelenir. AHP sürecinde, sistemin birbirlerinden ayrılmayan yapısı ve işlevi bir bütün olarak ele alınarak aynı zaman diliminde irdelenmeye çalışılır.

Süreç tarafından kullanılan hiyerarşik yapılar, öğeler arasındaki işlevsel ilişkiyi ve ilişkilerin sistemin tümü üzerindeki etkilerini tespit etmek amacıyla hiyerarşik yapıyı

ortaya koymak için oluşturulurlar. Hiyerarşik yapılar çeşitli şekillerde olabilmekle birlikte ana bir amaç ile başlayıp alt amaçlara, alt amaçlara etki eden kuvvetlere ve bu kuvvetler üzerinde etkisi olan kişiler ile bu kişilerin amaç, politika ve stratejilerine ve bunların çıktılarına doğru bir yayılım gösterir. AHP yönteminin kullanılabilmesinin için ilk aşama hiyerarşik yapının oluşturulmasıdır.

Hiyerarşi, içerisinde yer alan ögelerin ayrık bir şekilde gruplanabildiği ve her bir grubun içerisindeki ögelerin diğer gruplarda yer alan ögeleri etkileyebileceği varsayımına dayanan özel bir sistem tipidir [67]. Grup içerisinde yer alan ögeler bağımsız kabul edilmekle birlikte bu ögeler arasında da ilişkiler mevcutsa önce bağımsızlık sonrasında da bağımlılık durumları incelenerek birleştirilir. Hiyerarşik yapı içerisindeki düzeylerden birinde yer alan bir ögenin, kendi düzeyinin bir altında yer alan düzeydeki ögelerin tamamıyla ilişki içerisinde olması gerekli değildir. Hiyerarşide yer alan her bir düzey problemle ilgili farklı bir kesiti yansıtır olabilir. Bunun yanında, KV sisteme düzey veya ögeler ekleyebilir ya da var olanları çıkartarak göreceli önemleri belirginleştirebilir veya sistemin belirli bir kısmına yoğunlaşabilir. KV, ögelerin tamamı için değişken önemler saptandıktan sonra, ana amaç üzerindeki etkisi daha az olan ögeler sistem dışına çıkartılarak göreceli önemleri yeniden hesaplayabilir.

İnsanoğlunun gerçekleri gruplandırmak amacıyla kullandığı ana yol hiyerarşilerdir. Hiyerarşi kurmanın en basit yöntemi, bitişik düzeylerde yer alan elemanların arasında bağların kurulduğu doğrusal hiyerarşidir. Matematiksel hiyerarşik teoriye göre bir düzey, bir altındaki düzey tarafından etkilenir. Ayrıca, bitişik düzeylerin birbirlerine olan etkisi bir düzeyde yer alan elemanların göreceli öncelikleri ile bitişik olan düzeydeki elemanların her biriyle aralarındaki ilişkilerin karması ile değerlendirilir[33].



Şekil 6. 1 Basit bir AHP Yapısı

Hiyerarşik yapının avantajları:

- i. Hiyerarşik olarak gösterilen bir sistem, bir ögenin önceliğin değişmesi halinde daha alt seviyede yer alan öğelerin önceliklerinde meydana gelen değişimlerin görülebilmesine olanak verir.
- ii. Hiyerarşik gösterimde seviyeler üzerinde aşağıdan yukarıya doğru gidildiğinde detay azalır. Hiyerarşide aşağıda yer alan düzeylerde sistem hakkında yapısal ve işlevsel olarak verilen bilgi detaylıyken, yukarıdaki seviyelere doğru gidildikçe detay azalır fakat bu seviyelerde de problem bir bütün olarak görülebilir.
- iii. Hiyerarşik yapıda doğal sistemler temsil edilebilir ve etkin bir şekilde geliştirilebilir.
- iv. Yapı içerisinde yapılacak ufak değişiklikler hiyerarşide ufak etkiler yapabileceğinden hiyerarşik yapı katıdır. Diğer yandan eğer yapı iyi oluşturulabilirse, yapıya daha sonra ekleme yapılması hiyerarşi tarafından gösterilen performansı bozmayacağından da esnektir[73].

6.2 Ölçme ve Yapı Oluşturma

Karar vermek için AHP metodu kullanıldığında KV'nin karşısına ölçmeyle ilgili genellikle üç problem çıkar. Bunlardan ilki ölçme tekniğiyle ilgilidir. Ağırlık ölçmek için gerekli bir alet olmadığını varsayarsak, yeterince hafif ve elle kaldırılacak bir grup nesneyi tartmak için kullanılabilecek yöntemlerin ilki nesnelerin göreceli

ağırlıklarının tahmin edilmesidir. Tahmin için yollardan biri, grup içerisindeki nesneleri teker teker kaldırarak grubun tümü ile karşılaştırılmasıyla direk ağırlıkları hakkında tahminde bulunmaktır. Nesnelerin herbiri için bu yöntemle bulunan tahmini ağırlık, ağırlıkların toplamına bölünerek göreceli ağırlık bulunur. İkinci bir yöntem ise nesneleri ikili ikili karşılaştırmaktır. Nesneler ikili olarak kaldırılarak birbirleriyle kıyaslanır ve sonuçta göreceli ağırlıklar belirlenene kadar bu ikili karşılaştırmalara devam edilir. İkinci yöntem her adımda yalnızca iki nesneyi kıyaslayarak birbirleri ile ilişkilerini belirlemeye yöneliktir ve ilk yöntemle göre daha uzun olmasına rağmen daha basittir. Bu nedenle, sonuca götürecek ölçeğin bulunmadığı durumlarda ikili karşılaştırma yöntemine başvurulur.

Hassas ölçüm aletleri ile yapılan ölçümler de dahil olmak üzere, ölçümlerin tümü deneysel hata veya ölçme aletinden kaynaklanan hatalar nedeniyle tutarsız sonuçlara yol açabileceği için hiçbir ölçüm türünde kesin tutarlılık garantisi verilemez. Yukarıdaki ağırlık problemini tekrar ele aldığımızda, eğer ölçümde böyle bir hata varsa nesne ağırlıkları baz alınarak yapılan sıralamada değişiklik olur. Böyle bir hata, özellikle ölçme aletinin yeterince hassas olmaması durumunda ağırlıkları birbirine yakın nesneler için oluşur. Bazı problemler için tutarsızlık önemli sonuçlara yol açabilir[33].

Tutarlılık karar teorisinin temel taşlarından biridir. Fakat mükemmel tutarlılığa erişmek, ölçüm sırasında en duyarlı aletler kullanılsa dahi, neredeyse imkansızdır. AHP, seçeneklerin karşılaştırılmasındaki tutarsızlıktan öte üzerinde çalışılan problemdeki tutarlılık varsayımından sayısal olarak sapma derecesini önemser. Bu sapmayla ilişkili olarak da bir tutarlılık indeksi kullanır.

AHP problemlerinde, KV'nin problemle ilgili bilgisinin artmasına bağlı olarak ilgili problemin modelinin tutarlılık seviyesinin artması beklenir. İkili karşılaştırmalar ise KV'nin olabildiğince fazla bilgi kullanarak tutarlılığın artmasına katkı sağlar. Unutulmamalıdır ki tutarlılık, gerçekçi bir sonuç için tercih edilen ve lüzumlu bir hedef olmasına rağmen yeterli değildir. Hakkında karar verilmeye çalışılan problemi karşılayacak çözümler geliştirmek için, problemle ilgili düşünce, inanç ve yargıları ortaya koyarak aralarındaki farkları tespit etmek amacıyla sayısal bir ölçek bulunmalıdır. Bu şekilde KV'nin yargıları ile sayısal değerler güvenilir bir ilişki içerisinde değerlendirilebilir ve varılan sonuçların tutarlılığı ölçülebilir.

Ölçme ile ilgili karşılaşılan problemlerin ikincisi, tahmine dayanarak yapılan ölçümlerin büyük oran sabit kalacak ve değişmeyecek bir ölçüm yapısına uyarlanmasıdır. ÇDKV problemlerinde değerini uzun süre koruyabilecek özellikler sabit değişken kabul edilir. Bu yaklaşım ile Kvler, hiyerarşinin ölçülmesine ve analiz edilmesine yönelir. Değişik yollardan tekrarlayan şekilde değişik hiyerarşiler oluşturularak problem çözülür ve elde edilen sonuçlardan faydalanılarak yapılan ölçümlerin değişmezliği konusunda bilgi edinilebilir.

Ölçme ile ilgili karşılaşılan problemlerin üçüncüsü ise, problemlerin formüle edilmesi ve problem özelliklerinin tanımlanması için gerekli yargıların elde edilmesini sağlayacak uygun ortamların oluşturulmasıdır. İkili karşılaştırmaların kullanılması, ilgili kişilerle direk iletişime geçilip anket yapılması suretiyle problemin içeriğindeki alternatiflerin birbirlerine olan göreceli üstünlüklerinin elde edilmesini sağlar. Anketin yapılacağı kitle, konu hakkında uzman olmasalar da konu hakkında fikir sahibi olan, aşına olan, kişiler olmalıdır. Fakat insanoğlunun tutarsızlığı düşünüldüğünde, karşılaştırmalara verecekleri tutarsız yanıtlar, problemle ilgili önceliklerin ikili karşılaştırmalar yoluyla belirlenmesini ve problemin çözülmesini zorlaştırır. Cevap vericinin konuya hakim olmamasını da buna eklediğimizde tutarsızlık daha da artacaktır.

Bir grubun katılımı sonucu karar alınabilen durumlarda , grupta yer kişiler hem bireysel ilgi alanları ile ilgili olan konuya dair kararlarını ifade ederek birbirlerini tamamlayabilir hem de diğer kişilerin karar vermeleri esnasında konuya müdahil olup kararlarını netleştirmelerine katkı sunabilirler. Karar vericilerden oluşan grup, karar aşamasında bir uzlaşmaya varırsa sorun oluşmayacaktır. Ancak uzlaşmanın sağlanamadığı durumlarda pazarlık süreci başlar. Pazarlık sürecinde üçüncü şahıslardan yararlanılarak farklı sonuçlardan bir sentez yapılabilir veya çıkar çatışması içerisindeki grupta yer alan üyelerin herbirinden empati yaparak durumla ilgili kendi bakış açılarının yanı sıra diğer çıkar sahiplerinin gözünden bakmaları istenerek elde edilen çıktılar diğerlerinden elde edilen sonuçlar ile karşılaştırılabilir. Bu yöntemle işbirlikçi bir şekilde sonuca ulaşılabilir.

6.3 İkili Karşılaştırmalar Matrisi ve Ağırlık Kümesi

Bu kısımda, karar verici grup tarafından varılan sayısal yargıların elde edilmesinde kullanılan metoloji ve faaliyetler ile ilgili ağırlık kümelerinin oluşturulma yöntemi açıklanmıştır. AHP, bir düzeydeki öğelerin tümüyle bir üst düzeyde yer alan öğelerden bi tekinin örnek alınmasıyla, alt düzey öğelerinin tümünün üst düzey ögesine göreceli etkilerinin ikili olarak karşılaştırılıp bir matris oluşturulması ve oluşturulan matrisin kullanılmasıyla en büyük öz değerli özvektörün bulunması temeline dayanmaktadır.

N adet faaliyet gerçekleştiren bir grubun amaçlarının:

- İlgili faaliyetler için göreceli önemleri için yargıların geliştirilmesi,
- İlgili faaliyetler için verilen yargıları ölçmede kullanılmak üzere bir ölçü sistemi kullanarak yargıların ölçümünün garantilenmesi,

olduğunu varsayalım.

$F_1, F_2, \dots, F_n$ faaliyet seti, F_i, F_j faaliyet çiftinin sayısal yargıları, $n \times n$ boyutlu A matrisi ile aşağıdakiş şekilde ifade edilir.

$$A = (a_{ij}), \quad i, j = 1, 2, \dots, n \quad (6.1)[25]$$

$$a_{ij} = \alpha \Rightarrow a_{ji} = 1/\alpha, \quad \alpha \neq 0 \text{ olur. } a_{12}$$

$$A = \begin{bmatrix} 1 & a_{12} & \dots & a_{1n} \\ 1/a_{12} & 1 & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ 1/a_{1n} & 1/a_{2n} & \dots & 1 \end{bmatrix} \quad (6.2)[25]$$

Problem il ilgili matematiksel formülasyon ve ağırlıklar kümesi oluşturulması üç adımda gösterilmiştir.

$$1. \quad W_i / W_j = a_{ij} \quad i, j = 1, 2, \dots, n \quad (6.3)[25]$$

$$A = \begin{bmatrix} \frac{W1}{W1} & \frac{W1}{W2} & \dots & \frac{W1}{Wn} \\ \frac{W1}{W2} & \frac{W2}{W2} & \dots & \frac{Wn}{W2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{Wn}{W1} & \frac{Wn}{W2} & \dots & \frac{Wn}{Wn} \end{bmatrix}$$

2. A matrisinin i'inci satırının

$$a_{i1}, a_{i2}, \dots, a_{ij}, \dots, a_{in} \quad (6.4)[25]$$

ideal bir örnek için sonuçlarının tam çıktığı varsayılırsa, oran ve değerler eşit olurlar. Bu durumda, ideal durumda bu satırın ilk girdisini W_1 , ikinci girdisini W_2 vs. ile çarptığımızda sonuçların tümünün W_i çıkması gerekir.

$$\frac{W_i}{W_1} \cdot W_1 = \frac{W_i}{W_2} \cdot W_2 = \frac{W_i}{W_j} \cdot W_j = \frac{W_i}{W_1} \cdot W_1 = \dots = \frac{W_i}{W_n} \cdot W_n = W_i \quad (6.5)[25]$$

Gerçek hayatta bahse konu değerler W_i 'ye eşit olmaktan çok civarındadır. Bu sebeple W_i 'nin bu değerlerin ortalamasına eşitlenmesi daha makul olacaktır. Bu nedenle, $W_i/W_j = a_{ij}$ ideal durumu yerine $W_i = \text{Ort}(a_{i1}W_1, a_{i2}W_2, \dots, a_{in}W_n)$ olacaktır. Ortalama ise,

$$W_i = \left(\frac{1}{n}\right) \sum_{j=1}^n a_{ij} \cdot W_j \quad i=1,2,\dots,n \quad (6.6)[25]$$

ifadesi ile gösterilebilir.

3. a_{ij} değerinin iyi bir şekilde tahmin edildiği durumda, bahsedilen değer w_i/w_j 'ye yakın olur ancak, a_{ij} değerinin idealden saptığı durumlarda w_i ve w_j değerlerinin değişebilmesi, n değerinin değişmesini gerektireceğinden, n yerine en büyük özdeğer ($\lambda_{\max}$) kullanılır.

İkili karşılaştırma matrisinin tutarsız olma ihtimali hesaba katıldığında kullanılan formül,

$$W_i = 1/\lambda_{\max} \sum_{j=1}^n a_{ij} \cdot W_j \quad i=1,2,\dots,n \quad (6.7)[25]$$

şeklinde ifade edilir. 4.7'de yer alan ifadenin genelleştirilmesiyle $A \cdot w = \lambda_{\max} w$ ifadesi elde edilir.

Hesaplamanın tutarlı olduğu durumda $\lambda_{\max} = n$ olduğunda göre, $\lambda_{\max}$ 'ın n değerinden sapma derecesini göstermek için kullanılan tutarlılık endeksi $(\lambda_{\max} - n) / (n - 1)$ formülü ile hesaplanır. Genelde hesaplanan oranın %10 ve altında bir değerde olması yeterli olarak görülür.

6.4 Analitik Hiyerarşi Prosesinde Kullanılan Ölçek

Hiyerarşide bir düzeyi oluşturan faktörler arasındaki göreceli önemleri, ikili karşılaştırmalara dayanarak belirlenen bir ölçeğe göre puanlanır. AHP'de önceliklendirme için kullanılan ölçek ve puanlaması şekil 6.2'de gösterilmiştir.

ÖNEM DERECEŚİ	TANIM	AÇIKLAMA
1	Aynı Derecede Önemli	Her İki Faaliyetin Amaca Eşit Derecede Katkı Sağlaması Durumu
3	Orta Derecede Daha Önemli	Bir Kriterin Amaca Katkısının Diğerinden Orta Derecede Daha Fazla Etki Etmesi Durumu
5	Kuvvetli Derecede Daha Önemli	Bir Kriterin Amaca Katkısının Diğerinden Kuvvetli Derecede Daha Fazla Etki Etmesi Durumu
7	Çok Kuvvetli Derecede Daha Önemli	Bir Kriterin Amaca Katkısının Diğerinden Çok Kuvvetli Derecede Daha Fazla Etki Etmesi Durumu
9	Aşırı Derecede Daha Önemli	Bir Kriterin Amaca Katkısının Diğerinden Aşırı Derecede Daha Fazla Etki Etmesi Durumu
2,4,6,8	Ortalama Değerler	İki Ana Değerin Arasına Tekabül Eden Değerler

Tablo 6. 1 AHP Ölçeği [52]

6.5 Özvektörlerin Bulunması

6.5.1 En Basit ve Sapmalı Yöntem

Bu yönteme göre ilk olarak ikili karşılaştırmalardan oluşan matrisin her satırındaki elemanlar toplanır, sonra her satırın toplamı ikili karşılaştırmalar matrisinin tüm elemanlarının toplam değerine bölünmek suretiyle normalizasyon işlemi gerçekleştirilir. Bu yöntem ile ulaşılan öncelikler vektöründeki birinci eleman birinci alternatifin önceliğini göstermek suretiyle bu şekilde sırayla devam eder [21].

6.5.2 Daha İyi Bir Yöntem

Bu yönteme göre ikili karşılaştırmalardan oluşan matrisinin sütunlarında yer alan elemanlar sütun bazlı gruplanarak toplanır ve elde edilen toplamın tersi alınır. Ardından işlem sonucunda ulaşılan tersi alınmış değer, tersi alınmış değerlerin toplamına bölünerek normalizasyon işlemi gerçekleştirilir [21].

6.5.3 İyi Bir Yöntem

Bu yöntemde ikili karşılaştırmalardan oluşan matrisin sütunlarında yer alan elemanlar sütun bazlı gruplanarak toplanır, sütunda yer alan elemanlar içerisinde bulundukları sütunun elemanlarının toplam değerine bölünür. Oluşan matrisin satırlarında yer alan elemanlar satır bazlı gruplanarak toplanır. Ardından ulaşılan bu toplam değerler satırdaki elemanların sayısına bölünür [21].

6.5.4 En İyi Yöntem

İkili karşılaştırmalardan oluşan matrisin her bir satırında yer alan (n tane) elemanlar birbirleri ile çarpıldıktan sonra eleman sayısının (n) derecesinden kökü alınır. Sonuç normalize edilir[21].

6.6 Tutarlılık

AHP yönteminde tutarlılık, ikili karşılaştır ile ulaşılan değerlerin birbirleri ile olan mantıksal ve matematiksel ilişkisini ifade eder.

$$a_{jk} = a_{ik} / a_{ij} \quad i,j,k = 1,2,...,n \quad (6.8)[35]$$

6.8’de gösterilen eşitlik sağlandığında tutarlılık da sağlanmış olur. Bir A matrisinde tutarlılığın sağlanması A’nın en büyük özdeğerinin (λ_{max}) A’nın boyutuna (n)’ye eşit olması gerekli ve yeterli şarttır.

Matematiksel olarak ifade etmek gerekirse;

$$\sum_{i=1}^n \lambda_i = \text{İz}(A) = n \rightarrow \lambda_{\max} = n \quad (6.9)[35]$$

Tutarlılık göstergesi (CI) aşağıdaki formülle gösterilir :

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (6.10)[35]$$

Tutarlılık derecesi (CR), tutarlılık göstergesinin(CI) tesadüflük göstergesine (RI) bölünmesiyle elde edilir.

$$CR = \frac{CI}{RI} \quad (6.11)[35]$$

CR değerinin 0.10 değerinden küçük olması tutarlılık açısından uygun görülmektedir.

Tesadüflük Matris Boyutu														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tesadüflük Göstergesi														
0.0	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Tablo 6. 2 Tesadüflük Göstergesi[35]



7.BULANIK AHP

ÇÖKV tekniklerinden biri olan BAHP, AHP'yi baz alarak insanların düşünme tarzını simüle etmek için geliştirilmiş olmasına bağlı olarak yaygın olarak kullanılan bir yöntemdir. İlk olarak Saaty tarafından önerilen AHP' de, a_{ij} , ikili karşılaştırma oranları kesin sayılar olarak ele alınmıştır. Her bir ikili karşılaştırma matrisinin özvektör yöntemi kullanılarak çözülmesiyle elde edilen ağırlıklar ve performans puanları da kesin, reel sayılardır [45].

Çeşitli yazarlar tarafından önerilmiş olan birçok bulanık AHP yöntemi, alternatifler arasından seçim yapılması ve gerekçe problemlerinin çözümünde kullanılan ve bulanık küme teorisini temel alan sistematik yaklaşımlardır. KV' ler karşılaştırma yönteminin doğasında var olan bulanıktan dolayı tercihleri hakkında kesin değillerdir ve bundan ötürü genellikle aralık değerlendirmeleri sabit değerlendirmelerden daha güvenli bulurlar.

Bulanık AHP alanındaki en erken çalışma, Van Laarhoven ve Pedrycz'in üçgensel üyelik fonksiyonları ile tanımlanmış bulanık oranları karşılaştıran 1983 yılındaki çalışmalarında görülmüştür. Buckley (1985), karşılaştırma oranlarının bulanık önceliklerini yamuksal üyelik fonksiyonları ile belirtmiştir. Chang (1996), bulanık AHP'yi ele alırken bulanık AHP'nin ikili karşılaştırma ölçeği için üçgensel bulanık sayıların kullanımı ve ikili karşılaştırmaların sentetik derece değeri için derece analiz yönteminin kullanımı ile yeni bir yaklaşım sunmuştur.

Kuo ve Kao. 1999 yılında BAHP kullanıp, nüfusun özellikleri, pazarın özellikleri, yerin çekiciliği, rekabet, uygunluk ve ulaşılabilirlik kriterlerini ele alarak en iyi market yeri seçimi problemini değerlendirmiştir. Kriterlerin önceliklerinin belirlenmesi için üst düzey yöneticiler, market çalışanları, profesyonel danışmanlar, akademisyenler ve konuyla ilgili uzmanlara anket yapılmıştır.

Bulanık AHP için kullanılan Van Laarhoven ve Pedrycz Yaklaşımı, Buckley Yaklaşımı ve Chang'ın Mertebe Analizi Yaklaşımı sırasıyla 7.1,7.2 ve 7.3 başlıklarında, bulanık AHP uygulama türlerinden Kararsız(Hesitant) Bulanık AHP, Küresel(Spherical) Bulanık AHP ve Sezgisel Bulanık AHP yöntemleri sırasıyla 7.4, 7.5, 7.6 başlıklarında açıklanmıştır. 7.7 başlığında ise yine bulanık AHP uygulama türlerinden olan Pythagorean Bulanık AHP, Neutrosophic Bulanık AHP ve Z-Bulanık AHP ile ilgili literatürde yer alan kullanım örnekleri verilmiştir

Sözel Önem	Bulanık Ölçek	Karşılık Ölçek
Aynı Derecede Önemli	(1,1,1)	(1/1, 1/1, 1/1)
Biraz Daha Önemli	(1,3,5)	(1/5, 1/3, 1/1)
Kuvvetli Derecede Daha Önemli	(3,5,7)	(1/7, 1/5, 1/3)
Çok Kuvvetli Derecede Daha Önemli	(5,7,9)	(1/9, 1/7, 1/5)
Tamamıyla Daha Önemli	(7,9,9)	(1/9, 1/9, 1/7)

Tablo 7. 1 Bulanık Önem Dereceleri[74]

7.1 Van Laarhoven ve Pedrycz Yaklaşımı

Laarhoven ve Pedrycz'nin, doğrudan Saaty'nin AHP'sinin bir uzantısı olarak önerdikleri yöntemde AHP karşılık matristeki elemanları üçgensel bulanık sayılardır. Hesaplama adımları AHP ile aynı olan bu yöntemde bulanık ağırlıkları ve bulanık performans değerlerini elde etmek için Lootsman'ın logaritmik en küçük kareler yöntemi kullanılır.

Yöntemin adımları aşağıdaki gösterildiği şekildedir :

Adım 1 : Karar vericilere danışılarak şekil 7.1 'de görülen matrise benzer n+1 karşılık bulanık matrisler oluşturulur.

$$\tilde{A} = \begin{bmatrix} (1,1,1) & \begin{matrix} \tilde{a}_{121} \\ \tilde{a}_{122} \\ \vdots \\ \tilde{a}_{12P_{12}} \end{matrix} & \dots & \begin{matrix} \tilde{a}_{1n1} \\ \tilde{a}_{1n2} \\ \vdots \\ \tilde{a}_{1nP_{1n}} \end{matrix} \\ \begin{matrix} \tilde{a}_{211} \\ \tilde{a}_{212} \\ \vdots \\ \tilde{a}_{21P_{21}} \end{matrix} & (1,1,1) & \dots & \begin{matrix} \tilde{a}_{2n1} \\ \tilde{a}_{2n2} \\ \vdots \\ \tilde{a}_{2nP_{2n}} \end{matrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{matrix} \tilde{a}_{n11} \\ \tilde{a}_{n12} \\ \vdots \\ \tilde{a}_{n1P_{n1}} \end{matrix} & \begin{matrix} \tilde{a}_{n21} \\ \tilde{a}_{n22} \\ \vdots \\ \tilde{a}_{n2P_{n2}} \end{matrix} & \dots & (1,1,1) \end{bmatrix}$$

Şekil 7. 1 Çoklu karar vericilerin bulanık karşılık matrisi[75]

Kaynaklar	Yöntemin Temel Özelleikleri	Avantajları	Dezavantajları
Van Laarhoven ve Pedrycz (1983)	- Saaty'nin geliştirdiği AHP yönteminin ÜBS kullanılarak uygulanmasıdır - Bulanık ağırlıklar ve performans puanları Lootsman'ın logaritmik en küçük kareler yöntemi kullanılarak elde edilir	-Birden fazla sayıda karar vericinin fikirleri karşılık matriste modellenenbilmektedir	-Lineer denklemler hepsi çözülemeyebilir -Çok fazla hesaplama gerektirmektedir -Yalnızca ÜBS kullanılabilir.
Buckley (1985)	-Saaty'nin AHP yönteminin YBS kullanılarak uygulanmasıdır - Bulanık ağırlıklar ve performans puanları geometrik ortalama yöntemi kullanılarak elde edilir	-Bulanık uyarlaması kolaydır -Karşılık kıyaslama matrisi için tek bir çözüm garanti eder	-Çok fazla hesaplama gerektirmektedir
Boender ve dipğerleri (1989)	- Laarhoven ve Pedrycz yönteminin geliştirilmiştir -Yerel önceliklerin normalize edilmesi için daha sağlam bir yaklaşım sunar	-Birden çok karar vericinin fikri modellenenbilmektedir	-Çok fazla hesaplama gerektirmektedir
Chang(1996)	-Sentetik derece değerleri -Basit seviye sıralaması -Birleşik toplam sıralama	-Hesaplama gereksinimi diğer yöntemlere göre azdır -Klasik AHP yönteminin adımları takip edilir,ek bir işlem gerektirmez	-Sadece ÜBS kullanılabilir
Cheng (1996)	-Bulanık standartlar geliştirir -Performans puanları üyelik fonksiyonları ile gösterilir -Birleşik ağırlıkları hesaplamak için entropi kullanılır	-Hesaplama gereksinimi çok düşüktür	-Entropi olasılık dağılımı bilindiği zaman kullanılır -Yöntem hem olasılık hem de olabilirlik ölçülerine dayanmaktadır

Tablo 7. 2 BAHF Yöntemlerinin Kıyaslanması [76]

Burada $\tilde{a}_{ij}$ değerleri, çoklu karar vericiler tarafından belirlenen bulanık oranlardır. p_{ij} hiçbir KVnin fikir beyan etmediği durumlarda 0 değerini alabileceği gibi birden fazla KVnin görüş bildirmesi durumunda 1'den büyük bir değer alabilir.

Adım 2 : $z_i = (l_i, u_i, m_i)$ olsun.

$$l_i \left(\sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} \right) - \sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} u_j = \sum_{j=1}^n \sum_{k=1}^{P_{ij}} (\ln l_{ijk}), \quad \forall i$$

(7.1)[7]

$$m_i \left(\sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} \right) - \sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} m_j = \sum_{j=1}^n \sum_{k=1}^{P_{ij}} (\ln m_{ijk}), \quad \forall i$$

(7.2)[7]

$$u_i \left(\sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} \right) - \sum_{\substack{j=1 \\ j \neq i}}^n P_{ij} l_j = \sum_{j=1}^n \sum_{k=1}^{P_{ij}} (\ln u_{ijk}), \quad \forall i$$

(7.3)[7]

$\ln(a_{ijk}) = -\ln(a_{ijk})$ 'nın alt ve üst değerleri olan $\ln(l_{ijk})$ ve $\ln(u_{ijk})$ aşağıdaki eşitliği sağlamalıdır.

$$\ln(l_{ijk}) + \ln(l_{jik}) = \ln(u_{ijk}) + \ln(u_{jik}) = 0, \quad \forall i, j, k.$$

(7.4)[7]

Böylece 7.1 ve 7.3'te verilen denklemler lineer bağımlıdır. Bu 7.2'deki denklem için de geçerlidir. Genel olarak 7.1, 7.2 ve 7.3'teki denklemlerin çözümü şu şekilde verilir:

$$z_i = (l_i + t_1, m_i + t_2, u_i + t_1), \quad \forall i$$

(7.5)[7]

Burada t_1 ve t_2 rastgele seçilmektedir.

Adım 3 : Yukarıda verilen lineer sistemde, denklemlerin tümünün sağ taraflarında logaritmik işlemler yer almıştır. Buradan 7.6'daki şekilde bulanık ağırlık hesaplanır.

$$w_i = (\lambda_1 \exp(l_i), \lambda_2 \exp(m_i), \lambda_3 \exp(u_i))$$

(7.6)[7]

$$\lambda_1 = \left[\sum_{i=1}^n \exp(u_i) \right]^{-1} \quad \lambda_2 = \left[\sum_{i=1}^n \exp(m_i) \right]^{-1} \quad \lambda_3 = \left[\sum_{i=1}^n \exp(l_i) \right]^{-1}$$

7.6'daki eşitlik r_{ij} performans puanını hesaplamak için de kullanılabilir.

Adım 4 : Tüm karşılık matrisleri çözülene kadar Adım1 - 2 - 3 tekrar edilir. Bulanık ağırlıklar ve performans puanları kullanılarak $\tilde{A}_i$ seçeneğinin bulanık fayda değeri 7.7'de gösterildiği şekilde hesaplanır.

$$u_i = \sum_{j=1}^n w_j r_{ij} \quad (7.7)[7]$$

7.2 Buckley Yaklaşımı

Buckley, a_{ij} bulanık kıyaslama oranlarını kullanarak Saaty'nin AHP yönteminin başka bir formunu geliştirmiştir. Buckley, Laarhoven ve Pedrycz'in yöntemindeki lineer denklemlerin tek bir çözümünün olmaması ve mutlak suretle ÜBS kullanılması gereksinimlerine dikkat çekmiştir.

Buckley yaklaşımının adımları aşağıdaki gibidir :

Adım 1 : Karar vericiye danışılarak elemanları $\tilde{a}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij}) \forall i, j$ YBS olan karşılaştırma matrisi oluşturulur.[77]

Adım 2 : Geometrik ortalama

$$\tilde{z}_i = \left[\prod_{j=1}^n \tilde{t}_{ij} \right]^{1/n} \quad \forall i \quad (7.8) [77]$$

Bulanık ağırlık ise

$$w_i = \tilde{z}_i \oplus \left[\sum_{j=1}^n \tilde{z}_j \right]^{-1} \quad (7.9)[77]$$

şeklinde hesaplanır.

$\tilde{a}_{ij}$ 'nin sol ve sağ kısımları sırasıyla

$$f_i(\alpha) = \left[\prod_{j=1}^n ((b_{ij} - a_{ij})\alpha + a_{ij}) \right]^{1/n}, \alpha \in [0,1] \quad (7.10) [77]$$

$$g_i(\alpha) = \left[\prod_{j=1}^n ((c_{ij} - d_{ij})\alpha + d_{ij}) \right]^{1/n}, \alpha \in [0,1] \quad (7.11) [77]$$

olarak tanımlanır. Ayrıca,

$$a_i = \left[\prod_{j=1}^n \tilde{t}_{ij} \right]^{1/n} \quad (7.12) [77]$$

ve

$$a = \sum_{i=1}^m a_i. \quad (7.13) [77]$$

Benzer şekilde b_i ve b , c_i ve c , d_i ve d hesaplanır. Bulanık ağırlık w_i de

$$w_i = (a_i/d, b_i/c, c_i/b, d_i/a), \forall i \quad (7.14)[77]$$

şeklinde hesaplanır.

x , yatay ekseninde yer alan bir reel sayı olmak üzere, $\mu_{w_i}(x)$ şu şekilde özetlenir :

x	$\mu_{w_i}(x)$
$\leq (a_i/d)$	0
$\geq (a_i/d)$	0
$\left[\frac{b_i}{c}, \frac{c_i}{b} \right]$	1
$\left[\frac{a_i}{d}, \frac{b_i}{c} \right]$	$\alpha \in [0,1]$
$\left[\frac{c_i}{b}, \frac{d_i}{a} \right]$	$\alpha \in [0,1]$

Tablo 7. 3 Varlıkların İkili Karşılaştırma Matrisinde Yorumlanması[78]

$$f(\alpha) = \sum_{i=1}^m f_i(\alpha) \quad (7.15)[77]$$

$$g(\alpha) = \sum_{i=1}^m g_i(\alpha) \quad (7.16) [77]$$

olmak üzere,

$$x \in [a_i / d, b_i / c] \rightarrow x = f_i(a) / g_i(a) \quad (7.17) [77]$$

$$x \in [c_i / b, d_i / a] \rightarrow x = g_i(a) / f_i(a) \quad (7.18) [77]$$

olur.

Bulanık performans puanı r_{ij} bulunana kadar Adım – 2 tekrar edilir.

Adım 3 : Bulanık ağırlıklar ile bulanık performans puanları birleştirilir. Bulanık fayda değerleri U_i 'ler 7.19 'daki gibi hesaplanır.

$$U_i = \sum_{j=1}^n w_j r_{ij}, \quad \forall i. \quad (7.19)[77]$$

7.3 Chang'in Mertebe Analizi Yaklaşımı

$X = \{x_1, x_2, \dots, x_n\}$ ölçüt kümesi, $U = \{u_1, u_2, \dots, u_n\}$ ise amaç kümesi olsun.

Bu yöntemde ölçütlerin her biri alınarak mertebe analizi yöntemine tabi tutulur. Böylece ölçütlerin her biri için m adet mertebe analizi değerine ulaşılır.. Bu değerler

$$M_{gi}^1, M_{gi}^2, \dots, M_{gi}^m, \quad i = 1, 2, \dots, n \quad (7.20)[74]$$

şeklinde ifade edilir. Buradaki her $M_{gi}^j (j=1, 2, \dots, m)$ 'ler ÜBSdir.

Chang'in mertebe analizi yönteminin adımları aşağıdaki gibidir.

Adım 1 : Ölçüt i ye göre bulanık sentetik mertebenin değeri

$$S_i = \sum_{j=1}^m M_{gi}^j \otimes \left[\sum_{i=1}^n \sum_{j=1}^m M_{gi}^j \right]^{-1} \quad (7.21)[74]$$

şeklinde hesaplanır.

7.21 'de yer alan $\sum_{j=1}^m M_{gi}^j$ değerini bulmak için 7.22'de yer alan bulanık toplama işlemi uygulanır.

$$\sum_{j=1}^m M_{gi}^j = \left(\sum_{j=1}^m l_i, \sum_{j=1}^m m_i, \sum_{j=1}^m u_i \right) \quad (7.22)[74]$$

$$\sum_{i=1}^n \sum_{j=1}^m = \left(\sum_{i=1}^n l_i, \sum_{i=1}^n m_i, \sum_{i=1}^n u_i \right) \quad (7.23) [74]$$

$$\left[\sum_{i=1}^n \sum_{j=1}^m M_{gi}^j \right]^{-1} = \left(\frac{1}{\sum_{i=1}^n u_i}, \frac{1}{\sum_{i=1}^n m_i}, \frac{1}{\sum_{i=1}^n l_i} \right) \quad (7.24) [74]$$

Daha sonra 7.23'teki vektörün tersi 7.24 kullanılarak elde edilir.

Adım 2 : $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ olabilirlik derecesi

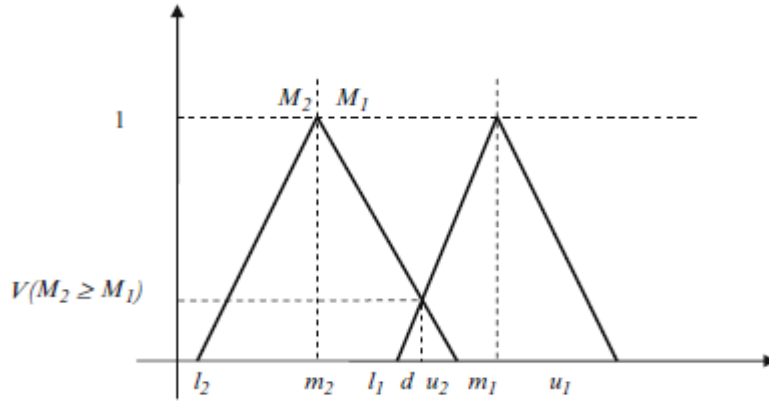
$$V(M_2 \geq M_1) = \sup_{y \geq x} [\min(\mu_{m1}(x), \mu_{m2}(y))] \quad (7.25) [74]$$

şeklinde tanımlanır.

$$V(M_2 \geq M_1) = \text{hgt}(M_1 \cap M_2) = \mu_{M2}(d) = \begin{cases} 1, & \text{eğer } m_2 \geq m_1 \\ 0, & \text{eğer } m_1 \geq m_2 \\ \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & \text{diğer} \end{cases}$$

Yine denk şekilde şekil 7.2' deki gibi de ifade edilebilir.

M_1 ve M_2 'yi kıyasa tabi tutabilmek için için $V(M_2 \geq M_1)$ ve $V(M_1 \geq M_2)$ değerlerinin bilinmesi gerekmektedir.



Şekil 7. 2 M_1 ve M_2 arasındaki kesişim noktası[74]

Adım 3 : Bir bulanık konveks sayının k tane bulanık konveks sayıdan M_i ($i = 1, 2, \dots, k$) büyük olabilme derecesi aşağıdaki gibi gösterilir..

$$V(M \geq M_1, M_2, \dots, M_k) = V = [(M \geq M_1) \text{ ve } (M \geq M_1) \text{ ve } \dots \text{ ve } (M \geq M_k)]$$

$$\min = V(M \geq M_i), \quad i = 1, 2, \dots, k \quad (7.26) [74]$$

$$d'(A_i) = \min V (S_i \geq S_k) \quad (6.27) [74]$$

olduğunu varsayarsak $k = 1, 2, \dots, n$; $k \neq i$ için ağırlık vektörü

$$W' = (d'(A_1), d'(A_2), \dots, d'(A_n))^T \quad (7.28) [74]$$

şeklinde gösterilir. A_i ($i=1, 2, \dots, n$) n sayısı kadardır.

Adım 4: Normalize edilmiş ağırlık vektörleri, W bulanık olmayan bir sayı olmak üzere,

$$W = (d(A_1), d(A_2), \dots, d(A_n))^T$$

şeklinde gösterilir [54].

7.4 Kararsız(Hesitant) Bulanık AHP

Literatürde KBAHP (Kararsız Bulanık Analitik Hiyerarşi Prosesi) ile bulunan ilk çalışma 2014 yılında Zhu ve Xu [77] tarafından yapılmıştır. Zhu ve Xu, kararsız çarpımsal tercih ilişkilerinden oran ölçeği türetmek için yeni bir önceliklendirme yöntemi olarak kararsız çarpımsal programlama yöntemini (hesitant multiplicative programming method – HMPM) geliştirdi.

KBAHP yönteminin Buckley'in AHP metodu üzerinden geliştirilen ilerleme adımları aşağıdaki gibidir. [79]

Adım 1 : KV'lerden tüm kriter ve alt kriterler için karşılaştırma matrisi verileri,dilsel değişkenler olarak, toplanır.

Adım 2 : Tablo 6.3 kullanılarak dilsel değişkenler üçgensel bulanık sayılara dönüştürülür.

$$\tilde{A}^k = \begin{bmatrix} 1 & \tilde{a}_{12}^k & \dots & \tilde{a}_{1n}^k \\ \tilde{a}_{21}^k & 1 & \dots & \tilde{a}_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1}^k & \tilde{a}_{n2}^k & \dots & 1 \end{bmatrix} \quad (7.29)[79]$$

$\tilde{A}^k$ ikili karşılaştırma matrisi olup,matrisin her bir elemanı $\tilde{a}_{ij}^k$ k. Karar vericinin i. Kriterin j. Kriteria göre önem derecesi için beyan ettiği görüşüdür.

Her bir ikili karşılaştırma matrisinin tutarlılığı kontrol edilir. Bu kontrol aşağıdaki şekilde yapılır.

$\tilde{A} = [\tilde{a}_{ij}]$ bulanık pozitif ters matris ve $A = [a_{ij}]$ bulanık olmayan pozitif ters matristir. Eğer $A = [a_{ij}]$ tutarlı ise $\tilde{A} = [\tilde{a}_{ij}]$ de tutarlıdır[77]. Kademeli ortalama entegrasyon yöntemine göre üçgen bulanık bir $\tilde{A}=(l,m,u)$ sayısı net A sayısına

$$A = \frac{l+4m+u}{6} \quad (7.30)[79]$$

şeklinde dönüştürülür. Eğer bu kontrol sonucunda ikili karşılaştırma matrisi tutarlı değilse Kvler karşılaştırmalarını değiştirmek durmundaadır.

Adım 3 : Bu adımda çatışmalar,uyumsuzluklar,belirlenir ve değerlendirmeler yenilenir. Kvlerin değerlendirmelerinin birbirlerine olan yakınlıkları kontrol edilir. Eğer değerlendirmeler yeterince yakın değilse durum KV'lere aktarılır ve durmu tatışarak değerlendirmelerini yenilemeleri istenir.

Adım 4 : Tablo 7.4'te verilen değerlerin sıralanması sonucu (s_0) en küçük ve (s_g) en büyük değerler olur. KV değerlendirmelerinin s_i ve s_j değerleri arasında olduğunu ve

$$s_0 \leq s_i < s_j \leq s_g \quad (7.31)[79]$$

varsayalım.

$\tilde{A} = (a,b,c,d)$ yamuk bulanık sayısının a ve d değerleri aşağıdaki şekilde hesaplanır.

Dilsel Değişkenler	Kısaltmalar	Üçgensel Bulanık Sayılar
Kesinlikle Yüksek Önemli	KYÖ	(7,9,9)
Çok Yüksek Önemli	ÇYÖ	(5,7,9)
Gerçekte Yüksek Önemli	GYÖ	(3,5,7)
Zayıf Yüksek Önemli	ZYÖ	(1,3,5)
Eşit Yüksek Önemli	EYÖ	(1,1,3)
Kesin Eşit	KE	(1,1,1)
Eşit Düşük Önemli	EDÖ	(0.33,1,1)
Zayıf Düşük Önemli	ZDÖ	(0.2,0.33,1)
Gerçekte Düşük Önemli	GDÖ	(0.14,0.2,0.33)
Çok Düşük Önemli	ÇDÖ	(0.11,0.14,0.2)
Kesinlikle Düşük Önemli	KDÖ	(0.11,0.11,0.14)

Tablo 7. 4 Kararsız Bulanık AHP Dönüşüm Tablosu [79]

$$a = \min\{a_L^i, a_M^i, a_M^{i+1}, \dots, a_M^j, a_R^j\} = a_L^i$$

$$d = \max\{a_L^i, a_M^i, a_M^{i+1}, \dots, a_M^j, a_R^j\} = a_R^j$$

(7.32)[79]

$$b = \begin{cases} a_m^i, & \text{if } i+1=j \\ OWA_{w^2} \left(a_m^i, \dots, a_m^{\frac{i+j}{2}} \right), & \text{if } i+j \text{ is even} \\ OWA_{w^2} \left(a_m^i, \dots, a_m^{\frac{i+j-1}{2}} \right), & \text{if } i+j \text{ is odd} \end{cases}$$

$$c = \begin{cases} a_m^{i+1}, & \text{if } i+1=j \\ OWA_{w^1} \left(a_m^j, a_m^{j-1}, \dots, a_m^{\frac{i+j}{2}} \right), & \text{if } i+j \text{ is even} \\ c = OWA_{w^1} \left(a_m^j, a_m^{j-1}, \dots, a_m^{\frac{i+j+1}{2}} \right), & \text{if } i+j \text{ is odd} \end{cases}$$

(7.33)[79]

OWA operasyonlarında kullanılan ağırlıklar α $[0,1]$ aralık değeri kullanılarak aşağıdaki gibi iki farklı şekilde tanımlanır.

Bu yöntemlerden ilki $W^1 = (w_1^1, w_2^1, \dots, w_n^1)$

$$w_1^1 = \alpha_2, w_2^1 = \alpha_2(1-\alpha_2), \dots, w_n^1 = \alpha_2(1-\alpha_2)^{n-2} \quad (7.34)[79]$$

şeklinde tanımlanır.

İkincisi ise $W^2 = (w_1^2, w_2^2, \dots, w_n^2)$

$$w_1^2 = \alpha_1^{n-1}, w_2^2 = (1-\alpha_1)\alpha_1^{n-2}, \dots, w_n^2 = 1-\alpha_1, \quad (7.35)[79]$$

Şeklinde tanımlanır. Burada g değerlendirme kriterindeki eleman sayısı, j en yüksek değerlendirme derecesi ve i en düşük değerlendirme değeri olmak üzere $\alpha_1 = \frac{g-(j-i)}{g-1}$

ve $\alpha_2 = \frac{(j-i)-1}{g-1}$ şeklinde hesaplanır.

Adım 5 : Ortak ikili karşılaştırma matrisi ($\sim C$)

$\tilde{c}_{ij} = (c_{ij_1}, c_{ij_{m1}}, c_{ij_{m2}}, c_{ij_u})$ olacak şekilde;

$$\tilde{C} = \begin{bmatrix} 1 & \tilde{c}_{12} & \dots & \tilde{c}_{1n} \\ \tilde{c}_{21} & 1 & \dots & \tilde{c}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{c}_{n1} & \tilde{c}_{n2} & \dots & 1 \end{bmatrix} \quad (7.36)[79]$$

şeklinde ifade edilir.

Önceki adımda gösterilen yamuk bulanık sayının tersi

$$\tilde{c}_{ji} = \left(\frac{1}{c_{ij_u}}, \frac{1}{c_{ij_{m2}}}, \frac{1}{c_{ij_{m1}}}, \frac{1}{c_{ij_1}} \right) \quad (7.37)[79]$$

şeklinde elde edilir.

Adım 6 : 6.38’de yer alan eşitlik ile karşılaştırma matrisinin her satırının bulanık geometrik ortalaması hesaplanır.

$$\tilde{r}_i = (\tilde{c}_{i1} \otimes \tilde{c}_{i2} \dots \otimes \tilde{c}_{in})^{1/n} \quad (7.38)[79]$$

Adım 7 : Her bir kriterin bulanık ağırlığı ($\tilde{w}$) bir önceki adımda hesaplanan bulanık geometrik ortalamaları kullanılarak hesaplanır.

$$\tilde{w}_i = \tilde{r}_i \otimes (\tilde{r}_1 \oplus \tilde{r}_2 \dots \oplus \tilde{r}_n)^{-1} \quad (7.39)[79]$$

Adım 8 : Bu adımda her bir alternatif için bulanık performans puanı hesaplanır. İlk 7 adım her bir ikili karşılaştırma matrisi için tekrarlanarak hesaplanır. Bulanık puanlar 6.40’daki gibi hesaplanır.

$$\tilde{S}_i = \sum_{j=1}^n \tilde{w}_j \tilde{S}_j, \forall i. \quad (7.40)[79]$$

Adım 9 : Alternatiflerin önem derecelerine karar vermek için yamuk bulanık sayılar durulaştırılır(defuzification). Yamuk bulanık sayıların durulaştırılması 6.41’deki eşitlik kullanılarak gerçekleştirilir.

$$D = \frac{c_l + 2c_m + c_u}{6} \quad (7.41)[79]$$

Adım 10 : Durulaştırılmış olan değerlere göre alternatifler sıralanır ve en iyi alternatif seçilir.

7.5 Küresel(Spherical) Bulanık AHP

Küresel bulanık AHP, Kutlu Gündoğdu ve Kahraman [80] tarafından aşağıdaki adımlarla gösterilmiştir.

Adım 1 : Hiyerarşik yapı oluşturulur. Bu yapıda en az üç kademe bulunmalıdır. Bunlar; amaç,kriterler ve alternatiflerdir.

Adım 2 : Tablo 7.5’te gösterilen dilsel değişkenler kullanılarak ikili karşılaştırma matrisi oluşturulur.

	(μ, ν, π)	Score Index (SI) Puan Endeksi (PE)
Kesinlikle Yüksek Önemli (KYÖ)	(0.9, 0.1, 0.0)	9
Çok Yüksek Önemli (ÇYÖ)	(0.8, 0.2, 0.1)	7
Yüksek Önemli (YÖ)	(0.7, 0.3, 0.2)	5
Biraz Çok Önemli (BÇÖ)	(0.6, 0.4, 0.3)	3
Eşit Önemli (EÖ)	(0.5, 0.4, 0.4)	1
Biraz Az Önemli (BAÖ)	(0.4, 0.6, 0.3)	1/3
Düşük Önemli (DÖ)	(0.3, 0.7, 0.2)	1/5
Çok Düşük Önemli (ÇDÖ)	(0.2, 0.8, 0.1)	1/7
Kesinlikle Düşük Önemli(KDÖ)	(0.1, 0.9, 0.0)	1/9

Tablo 7. 5 Küresel Bulanık AHP Dönüşüm Tablosu [80]

$$SI = \sqrt{100 * \left[\left(\mu_{\bar{A}_s} - \pi_{\bar{A}_s} \right)^2 - \left(\nu_{\bar{A}_s} - \pi_{\bar{A}_s} \right)^2 \right]} \quad (7.42)[80]$$

$$\frac{1}{SI} = \frac{1}{\sqrt{100 * \left[\left(\mu_{\bar{A}_s} - \pi_{\bar{A}_s} \right)^2 - \left(\nu_{\bar{A}_s} - \pi_{\bar{A}_s} \right)^2 \right]}} \quad (7.43)[80]$$

Puan endeksi hesaplamalarında, KYÖ(AMI), ÇYÖ(VHI), YÖ(HI), BÇÖ(SMI), ve EÖ(EI) için 7.42; EÖ(EI), BAÖ(SMI), DÖ(LI), ÇDÖ(VLI) ve KDÖ(ALI) için 7.43 eşitliği kullanılır.

Adım 3 : Tüm ikili karşılaştırma matrisleri için tutarlılık kontrolü yapılır. Tutarlılık kontrolü yapılabilmesi için dilsel değişkenler ile ifade edilen karşılaştırma matrisleri Tablo 6.4 kullanılarak sayısal değerlere dönüştürülür (SI).

Adım 4 : Kriter ve alternatifler için küresel bulanık yerel ağırlıklar hesaplanır. Her bir alternatifin ağırlığı 6.43'te gösterilen SWAM eşitliği ile hesaplanır. Küresel bulanık ağırlıkların hesaplanmasında ağırlıklı aritmetik ortalama kullanılır.

$$\begin{aligned}
SWAM_w(A_{S1}, \dots, A_{Sn}) &= w_1 A_{S1} + w_2 A_{S2} + \dots + w_n A_{Sn} \\
&= \left\langle \left[1 - \prod_{i=1}^n (1 - \mu_{A_{Si}}^2)^{w_i} \right]^{1/2}, \prod_{i=1}^n v_{A_{Si}}^{w_i}, \right. \\
&\quad \left. \left[\prod_{i=1}^n (1 - \mu_{A_{Si}}^2)^{w_i} - \prod_{i=1}^n (1 - \mu_{A_{Si}}^2 - \pi_{A_{Si}}^2)^{w_i} \right]^{1/2} \right\rangle
\end{aligned} \tag{7.43}[80]$$

Adım 5 : Global ağırlıkları elde etmek için hiyerarşik seviye yapısı kurulur. Alternatifler için final sıralamasını tahmin etmek amacıyla küresel bulanık sayılar toplanır. Bu hesaplamalar aşağıdan (alternatifler) yukarı(hedef) doğru yapılır.

Hesaplamaların yapılması için iki farklı yöntem vardır. Bunlardan ilki; 7.44'te gösterilen skor fonksiyonu kullanılarak bulanık ağırlıkların berraklaştırılması, ardından 7.45'teki eşitlik kullanılarak normalize edilmesi ve 7.46'da gösterilen küresel bulanık çarpma uygulanması.

$$S(\tilde{w}_j^s) = \sqrt{\left| 100 * \left[\left(3\mu_{\tilde{A}_s} - \frac{\pi_{\tilde{A}_s}}{2} \right)^2 - \left(\frac{v_{\tilde{A}_s}}{2} - \pi_{\tilde{A}_s} \right)^2 \right] \right|} \tag{7.44}[80]$$

$$\bar{w}_j^s = \frac{S(\tilde{w}_j^s)}{\sum_{j=1}^n S(\tilde{w}_j^s)} \tag{7.45}[80]$$

$$\begin{aligned}
\tilde{A}_{S_{ij}} = \bar{w}_j^s \cdot \tilde{A}_{Si} &= \left\langle \left(1 - \left(1 - \mu_{\tilde{A}_s}^2 \right)^{\bar{w}_j^s} \right)^{1/2}, v_{\tilde{A}_s}^{\bar{w}_j^s}, \right. \\
&\quad \left. \left(\left(1 - \mu_{\tilde{A}_s}^2 \right)^{\bar{w}_j^s} - \left(1 - \mu_{\tilde{A}_s}^2 - \pi_{\tilde{A}_s}^2 \right)^{\bar{w}_j^s} \right)^{1/2} \right\rangle \quad \forall i
\end{aligned} \tag{7.46}[80]$$

İkinic yöntem ise aşağıdaki gibi berraklaştırma yapılmadan devam edilmesidir. Bu yöntemde, global küresel bulanık ağırlıklar 7.47 kullanılarak hesaplanır. Nihai sonuç ise 7.46 kullanılarak hesaplanır.

$$\prod_{j=1}^n \tilde{A}_{S_{ij}} = \tilde{A}_{S_{i1}} \otimes \tilde{A}_{S_{i2}} \cdots \otimes \tilde{A}_{S_{in}} \quad \forall i$$

$$\text{i.e. } \tilde{A}_{S_{i1}} \otimes \tilde{A}_{S_{i2}} = \left\langle \mu_{\tilde{A}_{S_{i1}}} \mu_{\tilde{A}_{S_{i2}}}, \left(v_{\tilde{A}_{S_{i1}}}^2 + v_{\tilde{A}_{S_{i2}}}^2 - v_{\tilde{A}_{S_{i1}}}^2 v_{\tilde{A}_{S_{i2}}}^2 \right)^{1/2}, \right. \\ \left. \left(\left(1 - v_{\tilde{A}_{S_{i2}}}^2 \right) \pi_{\tilde{A}_{S_{i1}}}^2 + \left(1 - v_{\tilde{A}_{S_{i1}}}^2 \right) \pi_{\tilde{A}_{S_{i2}}}^2 - \pi_{\tilde{A}_{S_{i1}}}^2 \pi_{\tilde{A}_{S_{i2}}}^2 \right)^{1/2} \right\rangle \quad (7.47)[80]$$

Adım 6 : 6.44'teki eşitlik kullanılarak her bir alternatif için nihai sonuç berraklaştırılır.

Adım 7 : Berraklaştırılmış nihai sonuçlara göre alternatifler sıralanır ve en yüksek değere sahip olan alternatif seçilir.

7.6 Sezgisel Bulanık AHP

Sezgisel bulanık sayılar ilk olarak 1986 yılında Atanassov tarafından geliştirilmiş olup, dahil olma ve olmama durumlarının toplamını 1 veya daha küçük bir değerde olacak şekilde gösterilmiştir. Aralık değerli sezgisel bulanık AHP işleyişi, Öztayşi, Çevik Onar ve Kahraman tarafından aşağıdaki 9 adımda gösterilmiştir[81].

Adım 1 : Kvler tarafından Tablo 6.5'te gösterilen dilsel değişkenler kullanılarak ikili karşılaştırma matrisleri doldurulur.

Dilsel Değişkenler	Üyelik & Üyelik Dışı Değerleri
Kesinlikle Düşük (KD)	([0.10, 0.25], [0.65, 0.75])
Çok Düşük (ÇD)	([0.15, 0.30], [0.60, 0.70])
Düşük (D)	([0.20, 0.35], [0.55, 0.65])
Orta Düşük (OD)	([0.25, 0.4], [0.50, 0.60])
Yaklaşık Eşit (YE)	([0.45, 0.55], [0.30, 0.45])
Orta Yüksek (OY)	([0.50, 0.60], [0.25, 0.40])
Yüksek (Y)	([0.55, 0.65], [0.20, 0.35])
Çok Yüksek (ÇY)	([0.60, 0.70], [0.15, 0.30])

Tablo 7. 6 Aralık Değerli Sezgisel Bulanık AHP Dönüşüm Tablosu [81]

Adım 2 : Sezgisel ikili karşılaştırma karşılaştırma matrislerini ve kümelenmiş ikili karşılaştırma matrislerini($\sim R_a$) elde etmek için dilsel değişkenler Tablo 7.6 kullanılarak aralık değerli sezgisel bulanık sayılara dönüştürülür.

$$\tilde{R}_g = \begin{bmatrix} ([\mu_{g_{11}}^-, \mu_{g_{11}}^+], [v_{g_{11}}^-, v_{g_{11}}^+]) & \dots & ([\mu_{g_{1n}}^-, \mu_{g_{1n}}^+], [v_{g_{1n}}^-, v_{g_{1n}}^+]) \\ & \ddots & \\ ([\mu_{g_{n1}}^-, \mu_{g_{n1}}^+], [v_{g_{n1}}^-, v_{g_{n1}}^+]) & \dots & ([\mu_{g_{nn}}^-, \mu_{g_{nn}}^+], [v_{g_{nn}}^-, v_{g_{nn}}^+]) \end{bmatrix}$$

(7.48)[81]

Adım 3 : 7.49’da gösterilen puanlama fonksiyonu kullanılarak skor karar matrisleri($\sim S$) elde edilir.

$$\tilde{S} = \begin{bmatrix} [\mu_{g_{11}}^- - v_{g_{11}}^+, \mu_{g_{11}}^+ - v_{g_{11}}^-] & \dots & [\mu_{g_{1n}}^- - v_{g_{1n}}^+, \mu_{g_{1n}}^+ - v_{g_{1n}}^-] \\ \vdots & \ddots & \vdots \\ [\mu_{g_{n1}}^- - v_{g_{n1}}^+, \mu_{g_{n1}}^+ - v_{g_{n1}}^-] & \dots & [\mu_{g_{nn}}^- - v_{g_{nn}}^+, \mu_{g_{nn}}^+ - v_{g_{nn}}^-] \end{bmatrix}$$

(7.49)[81]

Adım 4 : 7.50’de gösterildiği gibi aralıklı üstel matrisler hesaplanır.

$$\tilde{A} = \begin{bmatrix} [e^{(\mu_{g_{11}}^- - v_{g_{11}}^+)}, e^{(\mu_{g_{1n}}^+ - v_{g_{1n}}^-)}] & \dots & [e^{(\mu_{g_{1j}}^- - v_{g_{1j}}^+)}, e^{(\mu_{g_{1j}}^+ - v_{g_{1j}}^-)}] \\ \vdots & \ddots & \vdots \\ [e^{(\mu_{g_{n1}}^- - v_{g_{n1}}^+)}, e^{(\mu_{g_{n1}}^+ - v_{g_{n1}}^-)}] & \dots & [e^{(\mu_{g_{nn}}^- - v_{g_{nn}}^+)}, e^{(\mu_{g_{nn}}^+ - v_{g_{nn}}^-)}] \end{bmatrix}$$

$$\begin{bmatrix} [\tilde{a}_{11}^-, \tilde{a}_{11}^+] & \dots & [\tilde{a}_{1n}^-, \tilde{a}_{1n}^+] \\ \vdots & \ddots & \vdots \\ [\tilde{a}_{n1}^-, \tilde{a}_{n1}^+] & \dots & [\tilde{a}_{nn}^-, \tilde{a}_{nn}^+] \end{bmatrix}$$

(7.50)[81]

Adım 5 : 7.51 kullanılarak aralıklı üstel matrislerin öncelik vektörleri hesaplanır.

$$\tilde{w}_i = \left[\frac{\sum_{j=1}^n \tilde{a}_{ij}^-}{\sum_{i=1}^n \sum_{j=1}^n \tilde{a}_{ij}^+}, \frac{\sum_{j=1}^n \tilde{a}_{ij}^+}{\sum_{i=1}^n \sum_{j=1}^n \tilde{a}_{ij}^-} \right] = [w_i^-, w_i^+], i = 1, \dots, n$$

(7.51)[81]

Adım 6 : 7.52 ve 7.53 kullanılarak olasılık derece matrisleri elde edilir.

$$P(\tilde{w}_i > \tilde{w}_j) = p_{ij} = \frac{\max(0, w_i^+ - w_j^-) - \max(0, w_i^- - w_j^+)}{(w_i^+ - w_i^-) + (w_j^- + w_j^+)}$$

(7.52)[81]

$$P(\tilde{w}_j > \tilde{w}_i) = p_{ji} = \frac{\max(0, w_j^+ - w_i^-) - \max(0, w_j^- - w_i^+)}{(w_i^+ - w_i^-) + (w_j^+ - w_j^-)}$$

(7.53)[81]

Adım 7: Olasılık dereceleri 7.54 kullanılarak önceliklendirilir.

$$w_i = \frac{\sum_{j=1}^n p_{ij} - 1}{n} + 0.5$$

(7.54)[81]

Adım 8 : Ağırlıklar 7.55 kullanılarak normalize edilir.

$$w_i^T = \frac{w_i}{\sum_{i=1}^n w_i}$$

(7.55)[81]

Adım 9 : Adımlar her her bir kriter için tekrar edilir. En sonunda her bir kriterin global ağırlığı hesaplanmış olur.

7.7 Diğer Bulanık AHP Yöntemleri

Pythagorean Bulanık AHP : Pythagorean BAHP, Yager [82] tarafından 2013 yılında geliştirilmiştir. Karasan, Ilbahar, ve Kahraman tarafından 2018 yılında düzenli depolama sahalarının seçim problemlerinde[83]; Kahraman, Öztayşi, Cevik Onar, ve Ilbahar tarafından 2019 yılında GSM operatörleri arasında seçim probleminde[84] ve

Guleria, Bajaj ve Kahraman tarafından 2020 yılında örüntü tanıma ve tıbbi tanı gibi birçok problemde[85] açıklanmış ve uygulanmıştır.

Neutrosophic Bulanık AHP : Neutrosophic BAHP, Smarandache[86] tarafından 2006 yılında geliştirilmiştir. Karasan ve Cengiz tarafında 2018 yılında Birleşmiş Milletler ulusal sürdürülebilir kalkınma hedeflerinin önceliklendirilmesi çalışmasında[87]; Kahraman, Öztayşi ve Cevik Onar tarafından 2020 yılında dış kaynak olarak avukatlık firmalarının performans ölçümü çalışmasında[20]; Yörükoğlu, Aydın ve Kahraman tarafından 2020 yılında akıllı konteyner değerlendirme çalışmasında[88] ve Karasan, Boltürk ve Kahraman tarafından 2020 yılında sürdürülebilir kentlerin zorlukları çalışmalarında[89] açıklanmış ve uygulanmıştır.

Z-Bulanık AHP : Kahraman, Öztayşi ve Cevik Onar tarafından 2019 yılında avukatlık ofislerinin performans karşılaştırması ve borç dosyalarının ideal tahsisi çalışmasında[1a], Otay ve Kahraman tarafından 2018 yılında güneş pili santral lokasyonu seçimi çalışmasında[90] ve Tüysüz ve Kahraman tarafından 2020 yılında yapılan, negatif ideal sonuca olan uzaklığa göre karar alınmasını sağlayan CODAS yönteminin Z-Bulanık Sayılar kullanılarak geliştirilmesinde ve geliştirilen bu yöntemin tedarikçi seçimi problemine uygulanması çalışmasında[91] açıklanmış ve kullanılmıştır.



8. YAZILIM UYGULAMASI

Çalışma kapsamında hesaplamaların gerçekleştirilmesi için bir uygulama(yazılım) geliştirilmiştir. “Performan Ölçüm Motoru” olarak çalışan bu uygulama Java (versiyon 7) programlama dilinde geliştirilmiş olup katmanlı mimari uygulanmıştır. Uygulamada esneklik amacıyla veri tabanı kullanılmamış olup, kısıtlı miktarda saklanması gereken veri(kriter verileri) dosya sisteminde saklanmıştır. Uygulamanın geliştirilmesi esnasında SOLID prensipleri uygulanmıştır. Paketleme yapısı (build automation) olarak maven kullanılmıştır. Geliştirme yöntemi olarak test odaklı geliştirme prensipleri izlenmiştir.

Uygulama, hem AHP hem de BAHP için ağırlıkların hesaplamasını gerçekleştirecek iki farklı operasyon içermektedir. Böylece kullanıcılar istedikleri yöntemi kullanarak ağırlık hesaplaması yapabilecektir. Aynı zamanda kriterlerin ve ağırlıkların değiştirilmesine olanak verecek tasarlanmıştır. Bu sayede değişen şartlara uyum sağlayacak esnekliktedir.

JAVA : Java 1995 yılında Sun mikrosistemleri tarafından geliştirilerek kullanıma açılan açık kaynak kodlu, nesneye dayalı bir programlama dilidir. JVM(Java Virtual Machine) platformu üzerinde çalışan java, JVM’in çalıştığı tüm ortamlarda çalışabilir. Java, hali hazırda dünyada en fazla kullanılan ikinci yazılım dilidir. Gömülü sistemler, mühendislik uygulamaları, finansal uygulamalar, web uygulamaları, mobil uygulamalar vb. çok geniş bir kullanım alanına sahiptir.

SOLID : SOLID kelimesindeki her bir harf bir prensibi temsil eder. SOLID prensibine uygun olarak geliştirilen uygulamalar esneklik, yeniden kullanılabilirlik, anlaşılabilirlik gibi açılardan istenen kriterleri sağlar.

S (Single Responsibility Principle) : Her birimin (fonksiyon, sınıf vb.) yerine getirilmesi gereken tek bir sorumluluğu olmalıdır.

O (Open Closed Principle) : Her birim davranışını değiştirmeye kapalı, geliştirilmeye açık olmalıdır.

L (Liskov ‘s Substitution Principle) : Yerine geçme prensibi olarak bilinen bu özellik, kalıtım yapılırken, kalıtım alan sınıfın üst sınıfının tüm özelliklerini kullanması ve bu özelliklerin üzerine yeni özellikler katması anlamına gelir.

I (Interface Segregation Principle) : Arayüzlerde kullanılmayacak, ihtiyaç olmayan, eklentilerin olmaması gerektiğini ifade eder.

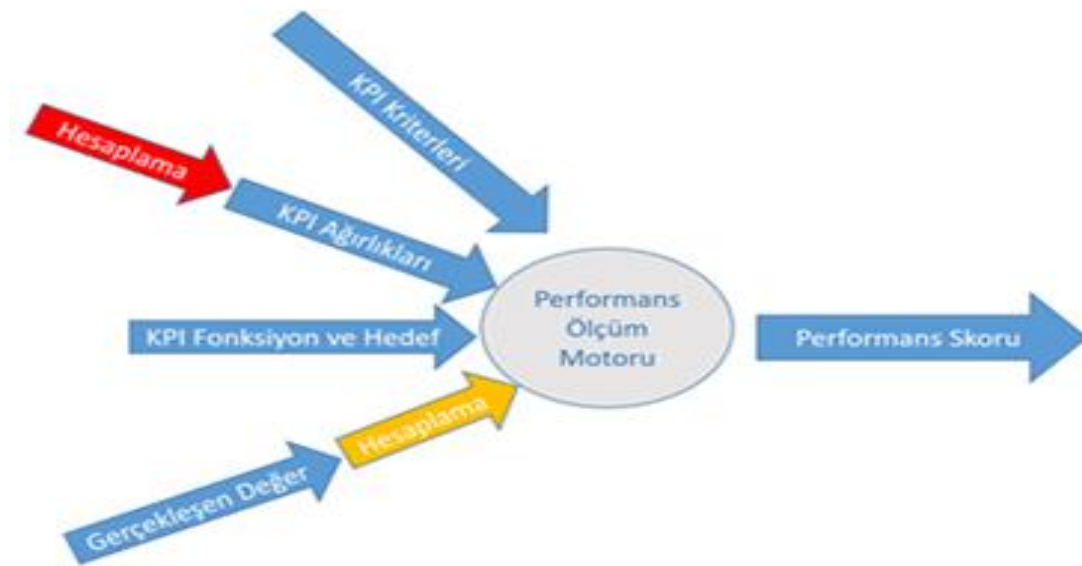
D (Dependency Inversion Principle) : Kalıtım yapılarında, kalıtım alan alt sınıfta yapılan bir değişiklik kalıtım alınan üst sınıfı etkilememelidir.

MAVEN : Apache tarafından geliştirilmiş olan araç özellikle java projelerinde bağımlılıkların yönetilmesi ve paketleme işlemleri için kullanılmaktadır. Maven projeleri POM (Project Object Model) dosyası ile konfigüre edilir. Maven, yapılandırma işlemini kolaylaştırmasının yanında kaliteli bir yapılandırma sistemi sunar ve bağımlılıkların düzenli olarak görülmesini sağlayarak kaliteli proje bilgisi elde edilmesine olanak verir.

8.1 Uygulama Tasarımı

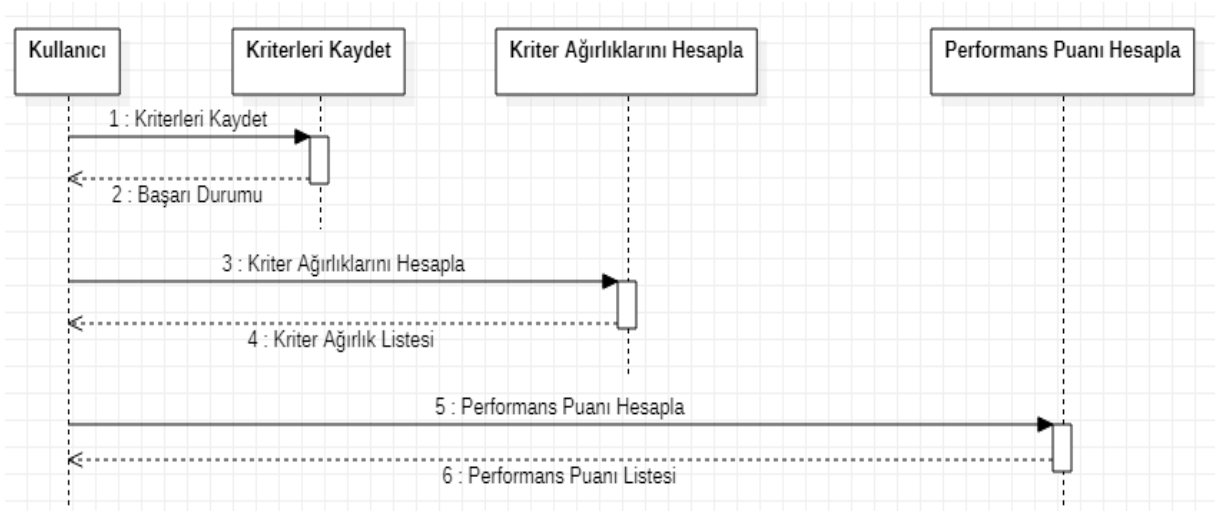
“Performans Ölçüm Motoru” olarak adlandırılan uygulama girdi olarak “KPI Kriterleri”, “KPI Ağırlıkları” (İkili karşılaştırma matrisi verileri ile hesaplanır), “KPI Fonksiyon ve Hedefleri” ve “Gerçekleşen Değerleri” alır ve çıktı olarak “Performans Skoru” verir. Uygulama modeli aşağıda gösterilmiştir.

Uygulama arayüzü üç adımdan oluşmaktadır. Bu adımlar; kriterlerin kaydedilmesi, kriter ağırlıklarının hesaplanması ve performans puanının hesaplanması. Kriter ağırlıklarının hesaplanması ve performans puanının hesaplanması ilerleyen kısımlarda anlatılacak olan alt adımlar içermektedir.



Şekil 8.1 Yazılım Uygulamasının Modeli

- Sarı ok, avukatlık ofislerinden gelen gösterge ham notlarının gösterge performans fonksiyonları ile işlenerek gösterge performans notunun hesaplanması işlemi ifade etmektedir.
- Mavi girdi okları firmadan ham halde alınacak olan verileri ifade etmektedir.
- Kırmızı ile işaretli olan ok performans kriterlerinin ağırlıklarının hesaplanması işlemi ifade etmektedir.



Şekil 8. 2 Yazılım Uygulaması Akış Diyagramı

Şekil 8.2’de yer alan akış diyagramından da görüleceği üzere kullanıcı ilk olarak kriterleri kaydetme operasyonu ile hesaplamalarda kullanacağı kriterleri sisteme tanıtır. Kriter kaydetme operasyonu başarılı yanıt döner ise bir sonraki adımda ikili karşılaştırma matrisini parametre olarak vererek kriter ağırlıklarının hesaplanmasını sağlar. Kriter ağırlıklarının hesaplanmasında AHP veya BAHF kullanılmasını tercih edebilir. AHP veya BAHF kullanarak hesaplaması yapılması için iki ayrı operasyon mevcuttur. Son adımda da avukatlık ofislerinin gerçekleşen değerlerini ve kriter ağırlıklarını kullanarak performans puanı hesaplaması yapar.

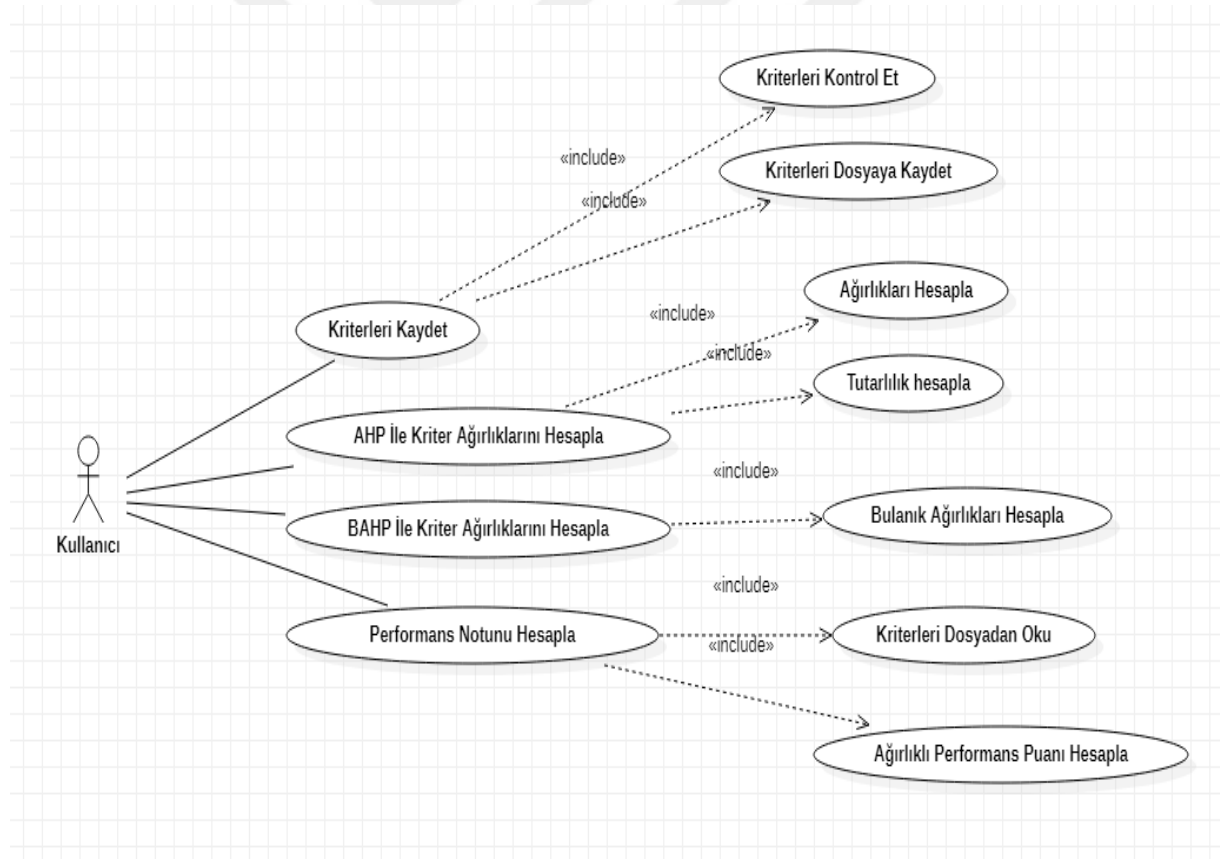
8.2 Uygulama

Uygulama kullanıcıya arayüz olarak dört operasyondan oluşan bir arayüz sunmaktadır. Bu operasyonlar; kriterleri kaydetme (saveCriterias), AHP ile ağırlık hesaplama (calculateWeights), BAHF ile ağırlık hesaplama (calculateFuzzyWeights)

ve performans notu hesaplamadır(`calculatePerformanceNote`). Bu operasyonların Java dilindeki tanımlamaları aşağıdaki gibidir :

```
public boolean saveCriteria (List<Criteria> criteriaList);  
  
public List<BigDecimal> calculateWeights (BigDecimal[][] matrix,int size)  
  
public List<BigDecimal> calculateFuzzyWeights (FuzzyComparisonValues[][]  
matrix, int size);  
  
public List<PerformanceNote> calculatePerformanceNote  
(List<LawOfficeCriteriaValue> officeCriteriaList);
```

Operasyonların işleyiş şekilleri alt başlıklarda detaylıca anlatılmıştır.



Şekil 8.3 Yazılım Uygulaması Kullanım Durumu Diyagramı

8.2.1 Kriterlerin Kaydedilmesi

“Kriterleri Kaydet” operasyonu parametre olarak kriter objesi listesi alır ve işlem sonucuna göre doğru veya yanlış bir değer cevabı verir. Parametre olarak alınan kriter objesi listesinin elemanlarının oluşturan kriter objesi içerisinde kriter sırası ve fonksiyon tipi objesi vardır. Fonksiyon tipi objesi fonksiyon tipi (hangi PROMETHEE fonksiyonu ile hesaplanacağı), x ve y eksen değeri listesini barındırır. x ve y değerleri liste olarak gösterilmekle birlikte fonksiyonun hesaplama ihtiyacına göre tek değer barındırabilir veya hiç değer barındırmayabilir.

Operasyon ilk olarak kriterlerin geçerliliğini kontrol eder. Her fonksiyon tipi için farklı bir geçerlilik kontrolü yapılır. Kontrol detayları aşağıdaki gibidir.

Fonksiyon 1: y eksen için yalnızca bir değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 ? true : false)
```

Fonksiyon 2: x ve y eksenleri için birer değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 1 ? true : false)
```

Fonksiyon 3: x ve y eksenleri için birer değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 1 ? true : false)
```

Fonksiyon 4: x ve y eksenleri için en az birer değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 1 ? true : false)
```

Fonksiyon 5: x eksen için iki, y eksen için bir değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 2 ? true : false)
```

Fonksiyon 6: x eksen için iki, y eksen için bir değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 2 ? true : false)
```

Fonksiyon 7: x ve y eksenleri için birer değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 1 && xAxisValues.size == 1 ? true : false)
```

Fonksiyon 8: x eksenini için iki değer verilmiş olup olmadığını kontrol eder.

```
if(xAxisValues.size == 2 ? true : false)
```

Fonksiyon 9: x eksenini için üç, y eksenini için iki değer verilmiş olup olmadığını kontrol eder.

```
if(yAxisValues.size == 2 && xAxisValues.size == 3 ? true : false)
```

Geçerlilik kontrolü sonrasında tüm kriterlerin gerekli şartları sağlaması durumunda kriterlere ait bilgiler dosyaya kaydedilir. Dosya çalışılan sistemin yetki kısıtlarına takılmaması için projenin altındaki bir konuma kaydedilir. Kriterler “criteriaDb” ismi ile kaydediliyor olup insanların anlayabileceği formatta değildir.

Kriterler bir kez kaydedildikten sonra tekrar tekrar kullanılabilir. Aynı kriterler, aynı tip fonksiyonlar ve kısıt değerleri kullanılacaksa tekrar tekrar kullanılabilir, her kullanımda tekrar kayıt yapılmasına gerek yoktur. Bu şartlar değişmediği sürece kaydetme fonksiyonunun tekrar çağırılması gerekmemektedir. Aynı kriterlerle farklı dönemlerde performans hesaplaması için kullanılabilir. Örneğin; 6 aylık değerlendirme periyotlarında performans hesaplaması yapan bir firma olduğunu varsayarsak kriterleri bir defaya mahsus kaydetmeleri halinde, hesaplama yöntemi veya kısıt değerleri değişmediği sürece, her 6 ayda farklı opsiyon ve gerçekleşen ölçüm değerleri ile performans hesaplaması yapılabilir.

8.2.2 AHP İle Kriter Ağırlıklarının Hesaplanması

“Ağırlık Hesaplama” operasyonu parametre olarak ikili karşılaştırma matrisi ile kriter sayısını alır ve hesaplanan ağırlıkları kriter sırası ile aynı sırada bir liste olarak şekilde cevap verir.

Karşılaştırma matrisi tam dolu olmak zorunda değildir. Karşılaştırma matrisinin satırları ve sütunları kriterleri ifade ettiği için, karşılaştırma yapılan iki kriter için tek yönlü karşılaştırma olması yeterlidir. Karşılık gelen değer verilen değer

tersidir. Yani, kriter bir kriter ikiden oldukça önemli (fairly strong) ise kriter iki kriter birden oldukça önemsizdir (reverse fairly strong). Ayrıca karşılaştırma matrisinin diyagonalı üzerinde yer alan değerler kriterlerin kendileri ile karşılaştırmaları anlamına gelen değerler olması nedeniyle birdir (Equal).

Operasyon ilk olarak karşılaştırma matrisinde, eğer varsa, boşlukları giderir. Ardından ağırlıkları hesaplar ve son olarak da tutarlılık indeksini hesaplar. Eğer tutarlılık indexi istenilen seviyedeysen ağırlıkların listesini, değilse “null” değer döner.

Ağırlıkların hesaplanması dışarıya kapalı (private) ayrı bir operasyondur. Bu operasyon parametre olarak ikili karşılaştırma matrisi ile kriter sayısını alır ve hesaplanan ağırlıkları kriter sırası ile aynı sırada bir liste olarak şekilde cevap verir. Bu operasyon kendi içerisinde yine dışarıya kapalı üç operasyon çağırır. Bunlar; “sütun ortalamalarını hesapla” (calculateColumnSum) , “sütunları normalize et” (normalizeColumn) ve “satır ortalamalarını hesapla” (calculateRowAverage) operasyonlarıdır.

“Sütun ortalamalarını hesapla” operasyonu, parametre olarak aldığı karşılaştırma matrisinin sütunlarını sırayla işleyen bir döngü içerisinde sütun elemanlarını toplayan ikinci bir döngü içerir. Her bir sütunda yer alan değerler toplandıktan sonra bu toplamaları bir liste olarak geri döndürür.

“Sütunları normalize et” operasyonu parametre olarak ikili karşılaştırma matrisi, kriter sayısı ve ağırlık listesini alır ve herhangi birşey dönmez. Bunun yerine parametre olarak aldığı matris üzerinde değişiklik yapar. Bu operasyon içerisinde, matrisin her bir sütunu üzerinde çalışan bir döngü içerisinde, üzerinde çalışılan sütunun her bir elemanı ilgili sütunun toplamına bölünerek normalizasyon işlemi gerçekleştirilir.

“Satır ortalamalarını hesapla” operasyonu parametre olarak normalize edilmiş ikili karşılaştırma matrisi ile kriter sayısını alır ve satır ortalamalarını bir liste olarak geri döndürür. Bu operasyon içerisinde, satırlar üzerinde çalışan bir döngü içerisinde, üzerinde çalışılan satırın elemanlarını toplayan ikinci bir döngü yer alır. Toplanan satır elemanları satır sayısına bölünerek satır ortalamaları alınır ve bu ortalamalar listeye aynı sırayla eklenir ve bu liste geri döndürülür. Döndürülen liste kriter ağırlıkları listesidir.

“Ağırlık Hesaplama” operasyonu son olarak tutarlılık indeksini hesaplayan “tutarlılığı kontrol et” (checkConsistency) operasyonunu çağırır. Bu operasyon parametre olarak ikili karşılaştırma matrisi, kriter sayısı ve ağırlık listesini alır ve doğru veya yanlış bir değer cevabı verir. Gizli bir operasyon olan “calculate” operasyonu kullanılarak karşılaştırma matrisinin satırlarında yer alan her bir elemanı ilgili sütuna karşılık gelen ağırlık değeri ile çarpılarak ağırlıklı satır toplamaları hesaplanır. Sonrasında bu satır toplamaları listesi ağırlık listesinde aynı sıraya karşılık gelen ağırlığa bölünür. Bu bölümlerin ortalaması alınarak λ_m değeri bulunur.

Daha sonra gizli bir operasyon olan “consistencyIndex” operasyonu ile 8.1’deki eşitlik kullanılarak tutarlılık indeksi hesaplanır. Burada n kriter sayısıdır.

$$CI = \frac{\lambda_m - n}{n - 1} \quad (8.1)$$

Sonrasında, hesaplanan CI değeri literatürde belirtilen ve sistemde yer alan rassal tutarlılık indeksine(RI) bölünerek tutarlılık oranı hesaplanır. Bu oran %10’dan küçükse hesaplanan değerler tutarlı olarak kabul edilir ve hesaplanan ağırlık listesi “Ağırlık Hesaplama” operasyonu tarafından geri döndürülür.

8.2.3 BAHF İle Kriter Ağırlıklarının Hesaplanması

“Bulanık Ağırlık Hesaplama” operasyonu parametre olarak bulanık ikili karşılaştırma matrisi ile kriter sayısını alır ve hesaplanan ağırlıkları kriter sırası ile aynı sırada bir liste olacak şekilde cevap verir.

“Ağırlık Hesaplama” operasyonunda olduğu gibi bu operasyonda da ilk olarak matriste, eğer varsa, boşlukları giderir. Aynı zamanda matriste yer alan değerlerin geçerliliğini de kontrol eder.

Daha sonra, gizli bir operasyon olan calculate operasyonu çağırır. Calculate operasyonu da parametre olarak bulanık ikili karşılaştırma matrisi ile kriter sayısını alır ve ağırlık listesi geri döner. Bu operasyon geometrik ortalama hesaplama, sütun toplamalarının tersini hesaplama, matris normalizasyonu ve ağırlık hesaplama olmak üzere dört tane gizli operasyon çağırır.

Geometrik ortalama hesaplama operasyonu parametre olarak bulanık ikili karşılaştırma matrisi ile kriter sayısını alır ve normal bir nxn boyunda matris döner. Geometrik ortalama bir satırda yer alan sağ, orta ve sol parametrelerin ayrı ayrı toplanarak kriter sayısı derecesinde kökü alınarak elde edilir.

Sütun toplamlarının tersini hesaplama operasyonu parametre olarak geometrik ortalama matrisini alır ve liste olarak $1 / (\text{sütun toplamı})$ değerlerini döner. Bu işlem için matrisin sütunları üzerinde çalışan bir döngü ile sütunlarda yer alan elemanlar toplanarak $1 / (\text{sütun toplamı})$ işlemi yapılarak liste halinde geri döndürülür.

Matris normalizasyonu operasyonu geometrik ortalama matrisi, kriter sayısı ve sütun toplamlarının tersini alır ve matrisi geri döner. Burada satırlar üzerinde çalışan bir döngü her bir elemanı karşılık gelen sütun toplam tersiyle çarpar ve normalize edilmiş matris elde eder.

Ağırlık hesaplama operasyonu parametre olarak normalize dilmiş matris ve kriter sayısını alır ve ağırlık listesini geri döner. Burada satırlar üzerinde çalışan bir döngü ile satırların toplam değerleri elde edilir. Daha sonra listeki değerlerin her biri bu toplam değere bölünür ve elde edilen sonuçlar ağırlıkları verir.

8.2.4 Performans Notu Hesaplama

“Performans Notu Hesaplama” operasyonu parametre olarak avukatlık ofisi değerleri objesinin listesini alır ve cevap olarak performans notları listesi geri döner.

Avukatlık ofisi değerleri objesi ofis numarası ve kriter değerleri objesinin listesini barındırır. Kriter değerleri objesi ise içerisinde kriter sırası, gerçekleşen değer ve kriter ağırlığı bilgilerini barındırır. Performans notu hesaplama operasyonu ilk olarak kriter kaydetme operasyonu ile dosyaya kaydedilmiş olan kriterleri okur. Okuduğu kriterleri sıra numarası – hesaplama yöntemi şeklinde eşleştirerek bir map içerisine koyar.

Daha sonra, avukatlık ofisi değerleri objesi listesi üzerinde çalışan bir döngü içerisinde aynı obje içerisinde yer alan kriter değerleri listesi üzerinde çalışan ikinci bir döngü ile her bir avukatlık ofisinin her bir kriteri için puan hesaplaması yapar. Kriter bazında gerçekleştirilen bu hesaplamaların sonuçları karşılık gelen kriter ağırlığı ile çarpılır ve elde edilen çarpımlar toplanarak ağırlık performans puanı hesaplanır. Hesaplanan ağırlıklı performans puanı listeye eklenir. Tüm avukatlık ofisleri için ağırlıklı performans hesaplaması tamamlanarak listeye eklendikten sonra liste geri döndürülerek süreç tamamlanır.

Her bir kriterin değerinin hesaplanması için farklı bir PROMETHEE fonksiyonu kullanılabilir. Kriterin hesaplanmasında hangi fonksiyonun kullanılacağı

kriterleri kaydetme esnasında belirtilir. Her bir fonksiyonun hesaplanması aşağıda açıklanmıştır.

Fonksiyon 1 :

Eğer gerçekleşen değer 0 ise 0, değil ise y eksenini için belirtilen değerdir.

```
if(value == 0) {  
    return 0;  
}else{  
    return getyAxisValues().get(0);  
}
```

Fonksiyon 2 :

Eğer gerçekleşen değer x eksenini için belirtilen değerden küçük ise 0, x eksenini için belirtilen değere eşit veya bu değerden büyük ise y eksenini için belirtilen değerdir.

```
if (value < getxAxisValues().get(0)) {  
    return 0;  
} else {  
    return getyAxisValues().get(0);  
}
```

Fonksiyon 3 :

Eğer gerçekleşen değer x eksenini için belirtilen değerden büyük veya eşit ise y eksenini için belirtilen değer, x eksenini için belirtilen değerden küçük ise y eksenini için belirtilen değerin x eksenini için bölünen değere bölümünün girilen değer ile çarpımıdır.

```
if(value >= getyAxisValues().get(0)) {  
    return parameters.getyAxisValues().get(0);  
}else {  
    Value * (getyAxisValues().get(0) / getxAxisValues().get(0))  
}
```

Fonksiyon 4 :

Eğer girilen değer x eksenini için belirtilen değerden küçük veya eşit ise karşılık gelen y eksenini için karşılık gelen değerdir. Aksi halde bu işlem özyinelemeli olarak çağırılır. Özyinelemeli çağırımlar çağırım sayısı x eksenini için belirtilen toplam eleman sayısına eşit oluncaya kadar tekrarlanır. Eşit olduğunda ise y eksenini için girilen en büyük değerdir.

```
private      BigDecimal      recursiveCalc(List<BigDecimal>
xAxisValues,List<BigDecimal> yAxisValues,BigDecimal value,int size,int counter)
{
    if(counter >= size ) {
        return yAxisValues.get(size-1);
    }
    if(value <= xAxisValues.get(counter)) {
        return yAxisValues.get(counter);
    }else {
        counter++;
        return recursiveCalc(xAxisValues, yAxisValues, value, size, counter);
    }
}
```

Burada size değeri x eksenini için belirtilen değer sayısı, counter ise çalışma sayısıdır ve ilk değeri 0'dır.

Fonksiyon 5 :

Eğer gerçekleşen değer x eksni için belirtilen ilk değerden küçük veya eşit ise 0'dır.

Eğer gerçekleşen değer x eksenini için belirtilen ikinci değerden küçük ise gerçekleşen değerden x eksenini için belirtilen ilk değerinin çıkarılmasının x eksenini için belirtilen ikincinci değerden ilk değerinin çıkarılmasına bölünmesidir.

Eğer gerçekleşen değer x eksenini için belirtilen ikinci değerden büyük veya eşit ise y eksenini için belirtilen ilk değerdir.

```
if(parameters.getValue().compareTo(xAxisValues.get(0)) <= 0) {
    return new BigDecimal(0);
}
```

```

if(value < xAxisValues.get(1)) {
    BigDecimal y = value - xAxisValues.get(0);
    BigDecimal x = xAxisValues.get(1)- xAxisValues.get(0);
    return y / x;
}else {
    return yAxisValues().get(0);
}

```

Fonksiyon 6 :

Eğer gerçekleşen değer x eksenini için belirtilen ikinci değerden büyük veya eşit ise y eksenindeki birinci değerdir.

Eğer gerçekleşen değer x eksenini için belirtilen ilk iki değerini ortalamasından büyük ise x eksenini için belirtilen ikinci değeri x eksenini için belirtilen ikinci değeri birinci değeri çıkarılmasına bölümünün gerçekleşen değeri çıkarılmasıyla elde edilen sonuca a dersek; a değerinin karesinin iki katının 1 den çıkarılmasıdır.

Eğer gerçekleşen değer x eksenini için belirtilen ilk iki değeri ortalamasından küçük veya eşit ise a değerinin karesinin iki katıdır.

```

if(value >= xAxisValues.get(1)) {
    return yAxisValues.get(0);
}else {
    BigDecimal
    average
    =
    (xAxisValues.get(0).add(xAxisValues.get(1)))/2;

```

```

if(value > average) {
    a = (value - (xAxisValues.get(1)))/((xAxisValues.get(1) -
    (xAxisValues.get(0))));
    return 1 - (2 * (a*a));
}else {
    a = (value - (xAxisValues.get(1)))/((xAxisValues.get(1) -
    (xAxisValues.get(0))));
    return (2 * (a*a));
}

```

```
}
```

Fonksiyon 7 :

Y eksenini için belirtilen değerin x eksenini için belirtilen değere bölümünün gerçekleşen değeri ile çarpımıdır.

```
return value * (yAxisValues.get(0) / xAxisValues.get(0));
```

Fonksiyon 8 :

Eğer gerçekleşen değeri x eksenini için belirtilen ikinci değerden büyük ise x eksenini için belirtilen ikinci değerin ilk değere bölümüdür. Aksi halde gerçekleşen değerin x eksenini için belirtilen ilk değere bölümüdür.

```
if(value > xAxisValues.get(1)) {  
    return xAxisValues.get(1) / xAxisValues.get(0);  
}else{  
    return value / xAxisValues.get(0);  
}
```

Fonksiyon 9:

Eğer gerçekleşen değeri x eksenini için belirtilen üçüncü değerden büyük ise x eksenini için belirtilen üçüncü değerden ilkinin çıkarılmasının ikinci değerden ilkinin çıkarılmasına bölümüdür.

Eğer gerçekleşen değeri x eksenini için belirtilen ilk değerden küçük veya eşit ise 0'dır.

Diğer hallerde değeri x eksenini için belirtilen üçüncü değerden büyük ise gerçekleşen değerden x eksenini için belirtilen ilk değerin çıkarılmasının x eksenini için belirtilen üçüncü değerden ilkinin çıkarılmasına bölümüdür.

```
if(value > xAxisValues.get(2)) {  
    y = xAxisValues.get(2)- xAxisValues.get(0);  
    x = xAxisValues.get(1)- xAxisValues.get(0);  
    return y / x;  
}
```

```
if(value <= xAxisValues.get(0)) {  
    return 0;  
}  
  
x = xAxisValues.get(1) - xAxisValues.get(0);  
y = value - xAxisValues.get(0);  
  
return y / x;
```

Bu yazılım uygulaması sayesinde hesaplamalar tekrar tekrar aynı veya farklı parametrelerle gerçekleştirilebilir. Yazılım uygulaması, hesaplama işlemlerinin süresini kısalttığı gibi hesaplama hatalarını da ortadan kaldırır. KV'ler süreç içerisinde kriterler arasındaki önem derecelerinin değiştiğini düşünebilir yada değişen şartlar kriterlerin arasındaki ilişkilerin değişmesine sebep olabilir. Değişen şartlar yazılımda yalnızca ilgili karşılaştırmaları içeren parametrelerin değişmesi ile devreye alınabilir.

Yani yazılım uygulaması kullanıcılara esneklik sağlar. Ayrıca yazılım uygulaması hesaplamaların standart olmasını sağlar. Aynı değerlerle her çalışmada aynı sonucu verir ve hesaplamalar esnasında insan faktörü devrede olmayacağı için hesap hatalarını ortadan kaldırır.

Yazılım uygulaması sayesinde çok sayıda hesaplama kısa sürede yapılabilir. Bizim uygulamamızda 15 tane avukatlık ofisi için 5 farklı kritere göre hesaplama yapmaktadır. Bunun için öncelikle kriterlerin oluşturulup, ağırlıkların hesaplanması ve her bir ofisin her bir kritere göre gerçekleşen değerinin uygun fonksiyon kullanarak hesaplanması ve en son da hesaplanan değerlerin ağırlıklandırılarak performans notuna dönüştürülmesi sürecinde insan eliyle işlem yapılması hem uzun hem de hataya açık olan bir süreç iken yazılım kullanılarak kolayca gerçekleştirilir.

9. AHP ve BAHP KULLANILARAK BORÇ TAKİBİ YAPAN AVUKATLIK OFİSLERİNİN PERFORMANS KARŞILAŞTIRMASI

Firmalar, zamanında ödenmeyen borçları için süreci takip etmek adına avukatlarla veya avukatlık ofisleri ile anlaşılmaktadırlar. Ödenmeyen borçlara ait dosyalar bu avukatlık ofisleri ile paylaşılmakta ve borç takibi ofislerce gerçekleştirilmektedir. Takip işlemi karşılığında avukatlık ofislerine ödeme yapılmaktadır. Firmalar, avukatlık ofislerinin kendilerine verilen dosyalardaki tahsilat performansını ölçerek maksimum borç ödemesinin sağlanması için çalışacakları ofisleri veya bu ofislere verilecek dosyaları kategorize etmek istemektedirler.

Bu çalışmada, borç toplama avukatlık ofisleri ile çalışan bir firmanın birlikte çalıştığı ofislerin performansını ölçmek için AHP ve BAHP yöntemleri kullanılarak detayları ilerleyen bölümlerde anlatılan bir performans yönetim sistemi geliştirilmiştir.

Firma ile yapılan görüşmelerde, performansı etkileyen faktörler arasındaki bağlantının rakamsal olarak ifade edilmesinin zor olduğu ancak faktörlerin birbirleri ile sözel olarak kıyaslanmasının mümkün olduğu görülmüştür. Bu nedenle AHP tabanlı sistem tercih edilmiştir. Hem AHP hem de BAHP sistemleri için bir yazılım geliştirilmiştir.

9.1 Hiyerarşik Yapı

Sistemin kurulmasında esnek bir yapı olması gözetilmiştir. Sistem avukatlık ofislerinin ve değerlendirme kriterlerinin hem sayısının hem de çeşitliliğinin değişebileceği şekilde tasarlanmıştır. Onar,Öztayşi ve Kahraman tarafından yapılan çalışma[3] baz alınmış, geçmiş dönem verileri analiz edilmiş ve firma yetkileleri ile de görüşülüp onların öncelikleri de göz önüne alınarak uygulamada kullanılacak değişkenler belirlenmiştir.

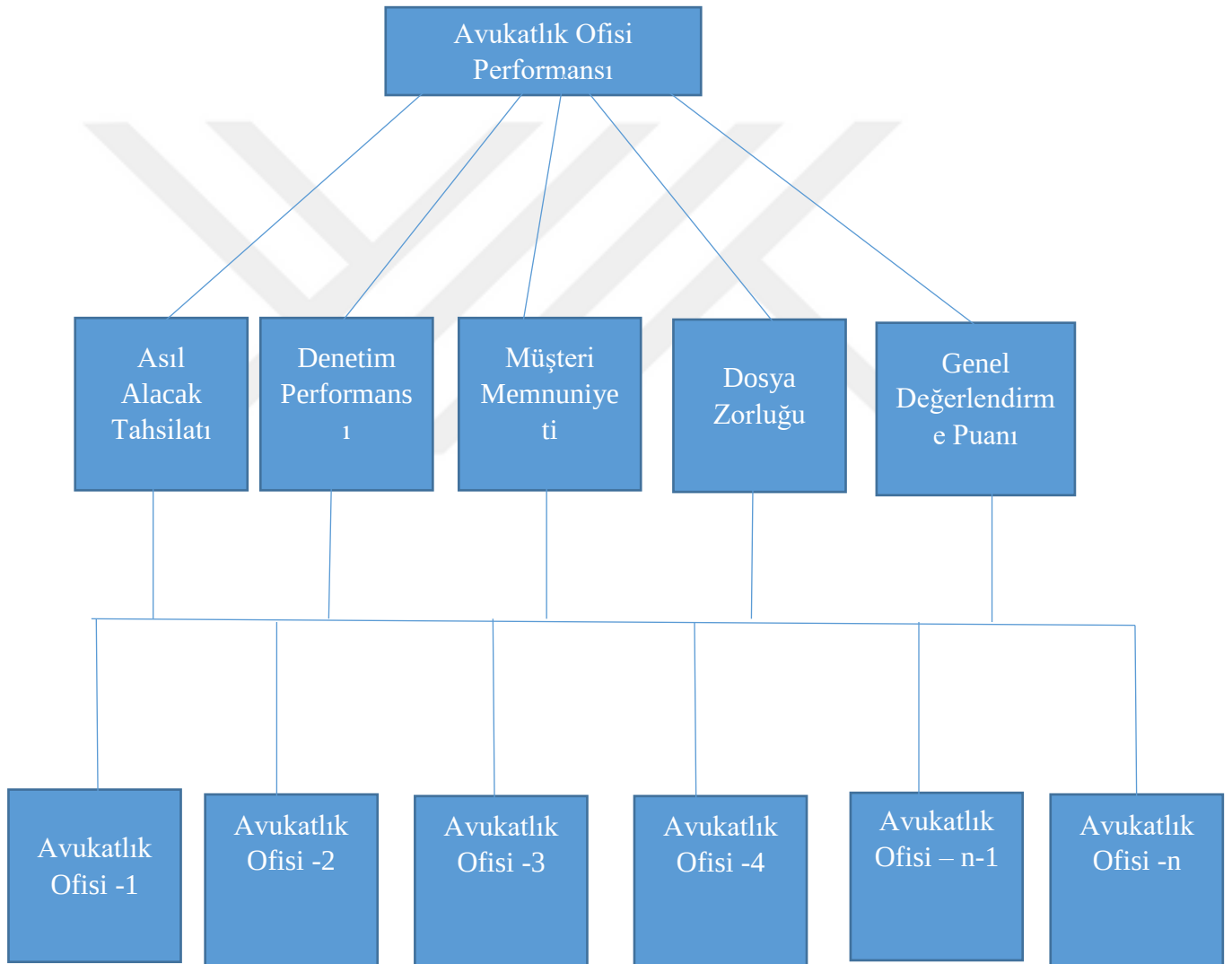
Bu değişkenler ;

- Asıl alacak tahsilatı,
- Denetim performansı,
- Müşteri memnuniyeti,
- Dosya zorluğu

- Genel değerlendirme puanıdır.

Bu değişkenlerin sayısı ihtiyaçlara göre artırılabilir veya azaltılabilir. Bu artış veya azalış modeli etkilemezken karşılaştırma matrisini değiştirir.

Bu bilgiler ışığında oluşacak hiyerarşik yapı Şekil 9.1'deki gibidir. Burada her bir avukatlık ofisi belirtilen kriterlere göre puanlandırılır. Yani avukatlık ofisi -1 için asıl alacak tahsilatı, denetim performansı, müşteri memnuniyeti, dosya zorluğu ve genel değerlendirme puanı için puanlama yapılır.



Şekil 9. 1 Hiyerarşik Yapı

Uygulama temelde 4 adımdan oluşmaktadır. Önceden tanımlanmış fonksiyonlar ile kriterlerin eşleştirilerek hesaplama yöntemlerinin belirlenmesi, tercih edilen yöntemle göre AHP veya BAHP ile ağırlıkların hesaplanması, performansların

karşılaştırılması ve son olarak da yapılan hesaplamaların güvenilirliklerinin test edilmesi.

Firmaların kriterler için aldıkları puan PROMETHEE fonksiyonları kullanılarak değerlendirilir. Bu kapsamda ilk olarak hangi kriterin hangi PROMETHEE fonksiyonu ile değerlendirileceği sisteme tanımlanır. Örnek olarak; birinci sırada gelecek kriter tip-7 PROMETHEE fonksiyonu ile değerlendirilecektir şeklinde tanımlamalar yapılır.

Daha sonra KV'ler (firma çalışanları) tarafından yapılan karşılaştırmalar sözel ifadelerle gerçekleştirilir. İlk adımda bu karşılaştırmalar AHP ve BAHP için sayısal değerlere dönüştürülür. Daha sonra ilerleyen kısımlarda anlatılacak olan AHP ve BAHP adımları ile ağırlıkları hesaplanır.

AHP veya BAHP ile hesaplanan ağırlıklar ve ilk adımda bulunan değerlendirme puanları kullanılarak alternatiflerin performans puanları elde edilir. Hesaplanan puanların tutarlılık ve rassal tutarlılık indeksleri hesaplanarak güvenilirlikleri test edilir.

9.3 AHP Uygulaması

AHP ile yapılan uygulamada kullanılacak kriterler sırasıyla asıl alacak tahsilatı,denetim performansı, müşteri memnuniyeti, dosya zorluğu ve genel değerlendirme puanıdır. Bu kriterlerin değerlendirilmesinde kullanılacak olan PROMETHEE fonksiyonları Tablo 9.1'de gösterilmiştir.

PARAMETRE	PROMETHEE FONKSİYONU
Asıl Alacak Tahsilatı	Fonksiyon - 7
Denetim Performansı	Fonksiyon - 4
Müşteri Memnuniyeti	Fonksiyon - 4
Dosya Zorluğu	Fonksiyon - 7
Genel Değerlendirme Puanı	Fonksiyon - 4

Tablo 9. 1 AHP Uygulaması İçin Parametre – Fonksiyon Eşleşme Tablosu

KV'ler tarafından değerlendirme kriterleri arasındaki karşılaştırmalar sözel olarak yapılır. Burada diagonale göre karşılıklarının çarpımının 1 olması gerektiği için tablonun tamamının doldurulmasına gerek yoktur. Örneğin,Tablo 9.2'de denetim performansı müşteri memnuniyetine göre zayıftır. Bu durumda müşteri memnuniyeti

denetim performansına göre ters zayıftır. Geliştirilmiş olan yazılım tabloda yer alan boşlukları dolduracak şekilde tasarlanmıştır.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	E	FS	VS	W	A
Denetim Performansı		E	W		FS
Müşteri Memnuniyeti			E		E
Dosya Zorluğu		W	FS	E	FS
Genel Değerlendirme Puanı			E		E

Tablo 9. 2 AHP Uygulaması İçin KV Sözel Değerlendirmeleri Tablosu

Yukarıda bahsedilen boşlukların doldurulması sonucu oluşan değerlendirme matrisi aşağıdaki gibidir. Yazılım tarafından doldurulan alanlar kırmızı ile gösterilmiştir.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	E	FS	VS	W	A
Denetim Performansı	TFS	E	W	TW	FS
Müşteri Memnuniyeti	TVS	TW	E	TFS	E
Dosya Zorluğu	TW	W	FS	E	FS
Genel Değerlendirme Puanı	TA	TFS	E	TFS	E

Tablo 9. 3 AHP Uygulaması İçin KV Sözel Değerlendirmeleri Tablosu – (Eksik Alanların Doldurulmuş Hali)

Sözel değerlendirmeler kısaltmalarla ifade edilmiş olup, ikili karşılaştırma tablolarında bu kısaltmalar kullanılmıştır.

Sözel Değerlendirme	Kısaltma	Sayısal Değer
Kesin	A	9
Çok Güçlü	VS	7
İyice Güçlü	FS	5
Zayıf	W	3
Eşit	E	1
Ters Eşit	TE	1
Ters Zayıf	TW	0,333
Ters İyice Güçlü	TFS	0,200
Ters Çok Güçlü	TVS	0,143
Ters Kesin	TA	0,111

Tablo 9. 4 AHP Sözel-Sayısal Değerler Referans Dönüşüm Tablosu

KV'ler tarafından sözel olarak belirtilen karşılaştırma matrisinin hesaplamalar için sayısal değerlerle dönüştürülmüş hali Tablo 9.5'te gösterilmiştir.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	1	5	7	3	9
Denetim Performansı	0,2	1	3	0,333	5
Müşteri Memnuniyeti	0,143	0,333	1	0,2	1
Dosya Zorluğu	0,333	3	5	1	5
Genel Değerlendirme Puanı	0,111	0,2	1	0,2	1

Tablo 9. 5 AHP Uygulaması İçin KV Değerlendirmeleri – Sayısal

Normalizasyon işlemini gerçekleştirmek için tablonun sütunları toplanır. Sütunlar ve toplam değerleri Tablo 9.6’da gösterilmiştir.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	1	5	7	3	9
Denetim Performansı	0,2	1	3	0,333	5
Müşteri Memnuniyeti	0,143	0,333	1	0,2	1
Dosya Zorluğu	0,333	3	5	1	5
Genel Değerlendirme Puanı	0,111	0,2	1	0,2	1
Toplamlar	1,787	9,533	17	4,733	21

Tablo 9. 6 AHP Uygulaması – Satır Toplamları

Her bir değer içerisinde yer aldığı sütunun toplam değerine bölünerek elimizdeki değerlendirme tablosu normalize edilir. Normalizasyon sonrası elde edilen değerler Tablo 9.7’de gösterilmiştir.

	0,56	0,524	0,412	0,634	0,429
	0,112	0,105	0,176	0,07	0,238
	0,08	0,035	0,059	0,042	0,048
	0,186	0,315	0,294	0,211	0,238
	0,062	0,021	0,059	0,042	0,048
Toplam	1	1	1	1	1

Tablo 9. 7 AHP Uygulaması – Normalize Edilmiş Tablo

Normalize olmuş tablonun satır ortalamaları alınarak ağırlıklar bulunur.

						W
	0,56	0,524	0,412	0,634	0,429	0,512
	0,112	0,105	0,176	0,07	0,238	0,14
	0,08	0,035	0,059	0,042	0,048	0,053
	0,186	0,315	0,294	0,211	0,238	0,249
	0,062	0,021	0,059	0,042	0,048	0,046
Toplam	1	1	1	1	1	1

Tablo 9. 8 AHP Uygulaması – Satır Ortalamaları

Yaptığımız bu hesaplamalar sonucunda Tablo 9.8’da görülen W kolonu kriterlerin ağırlıklarını göstermektedir. Her bir satırın W kolon değeri o sıradaki kriterin ağırlığını ifade eder. Uygulamamızda, “Asıl Alacak Tahsilatı” kriterinin ağırlığı 0,512; “Denetim Performansı” kriterinin ağırlığı 0,14; “Müşteri Memnuniyeti” kriterinin ağırlığı 0,053; ”Dosya Zorluğu” kriterinin ağırlığı 0,249; “Genel Değerlendirme Puanı” kriterinin ağırlığı 0,046 olur.

Kriter	Ağırlık
Asıl Alacak Tahsilatı	0,512
Denetim Performansı	0,14
Müşteri Memnuniyeti	0,053
Dosya Zorluğu	0,249
Genel Değerlendirme Puanı	0,046

Tablo 9. 9 AHP Uygulaması – Kriter Ağırlıkları

Bulunan ağırlıkların ne kadar tutarlı olduğunun kontrol edilmesi için tutarlılık indeksi, rassal tutarlılık indeksi (RI) ve tutarlılık oranı hesaplamaları yapılır. Tutarlılık oranının n%10’un altında olması beklenmektedir.

Öncelikle değerlendirme matrisi ile ağırlıklar çarpılır.

$$Aw_1 = (1 \times 0.512 + 5 \times 0.140 + 7 \times 0.053 + 3 \times 0.249 + 9 \times 0.046) = 2.744$$

$$Aw_2 = (0.200 \times 0.512 + 1 \times 0.140 + 3 \times 0.053 + 0.333 \times 0.249 + 5 \times 0.046) = 0.714317$$

$$Aw_3 = (0.143 \times 0.512 + 0.333 \times 0.140 + 1 \times 0.053 + 0.200 \times 0.249 + 1 \times 0.046) = 0.268636$$

$$Aw_4 = (0.333 \times 0.512 + 3 \times 0.140 + 5 \times 0.053 + 1 \times 0.249 + 5 \times 0.046) = 1.334496$$

$$Aw_5 = (0.111 \times 0.512 + 0.200 \times 0.140 + 1 \times 0.053 + 0.200 \times 0.249 + 1 \times 0.046) = 0.233632$$

Bu çarpımların sonuçları ağırlıklara bölünerek normalize edilir.

$$Aw_1 / w_1 = 2.744 / 0.512 = 5.359$$

$$Aw_2 / w_2 = 0.714317 / 0.140 = 5.102$$

$$Aw_3 / w_3 = 0.268636 / 0.053 = 5.069$$

$$Aw_4 / w_4 = 1.334496 / 0.429 = 5.359$$

$$Aw_5 / w_5 = 0.233632 / 0.046 = 5.079$$

Normalize edilen değerlerin ortalaması alınarak λ_m değeri hesaplanır.

$$\lambda_m = (5.359 + 5.102 + 5.069 + 5.359 + 5.079) / 5 = 5.194$$

$$CI = \frac{\lambda_m - n}{n - 1} = \frac{5.194 - 5}{4} = 0.049$$

n	RI
3	0,52
4	0,89
5	1,11
6	1,25
7	1,35
8	1,4
9	1,45

Tablo 9. 10 *Rassal Tutarlılık İndeksi (RI)*

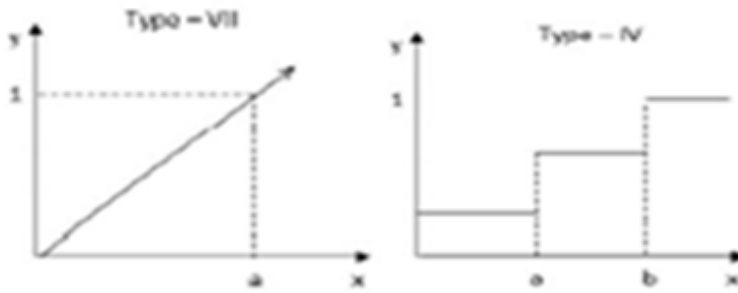
Tablo 9.10'da n eleman sayısını ifade etmektedir. Bizim uygulamamızda n 5 değerini alır. Yani RI değerimiz 1,11'dir.

Tutarlılık oranı, CR, aşağıdaki formül ile hesaplanır.

$$CR = \frac{CI}{RI} = \frac{0,049}{1,11} = 0,044 \text{ olur.}$$

Yukarıda belirtildiği gibi hesaplanan CR değerinin %10'dan (0,1) küçük olması durumunda hesaplanmış olduğumuz ağırlıklar tutarlıdır. Bulmuş olduğumuz CR değeri, 0.044 , 0.1'den küçük olduğundan yapmış olduğumuz hesaplama tutarlıdır.

Uygulamanın sonraki adımında değerlendirme kriterleri için alternatiflerin (avukatlık ofisleri) puanları belirlenen fonksiyonlara göre hesaplanıp ağırlandırılarak toplam performans puanları elde edilir ve elde edilen toplam performans puanlarına göre alternatifler sıralanır.



Şekil 9. 2 AHP Uygulamasında Kullanılan PROMETHEE Fonksiyonları

AHP uygulamasında Şekil 9.2'te görülen PROMETHEE fonksiyonlarından faydalanılmıştır. Burada;

“Asıl Alacak Tahsilatı” kriteri için Type-VII fonksiyonu kullanılacak olup, x ve y eksenlerindeki sınır değerleri 1’dir.

“Denetim Performansı” ” kriteri için Type-IV fonksiyonu kullanılacak olup x eksen değeriyle 25,50,75,90 ve 100; karşılık gelen y eksen değeriyle 0,0.4,0.6,0.8, ve 1dir.

“Müşteri Memnuniyeti” kriteri için Type-IV fonksiyonu kullanılacak olup x eksen değeriyle 0,1,2,3; karşılık gelen y eksen değeriyle 1,0.67,0.33 ve 0’dır. Burada grafik tersten kurgulanmıştır.

“Dosya Zorluğu” kriteri için Type-VII fonksiyonu kullanılacak olup, x ve y eksenlerindeki sınır değerleri 1’dir.

“Genel Değerlendirme Puanı” kriteri için Type-IV fonksiyonu kullanılacak olup x eksenli değerleri sırasıyla 25,50,75,90 ve 100; karşılık gelen y eksenli değerleri ise sırasıyla 0,0.4,0.6,0.8, ve 1dir.

Her bir alternatif ofisin her bir değerlendirme kriteri için gerçekleştirdiği bir değer mevcuttur. Bu değerler, yukarıdaki fonksiyonlar ve sınır değerler kullanılarak hesaplanır. Bu hesaplamalar ve sonuçlar alternatiflerden biri için Tablo 9.11’de gösterilmiştir. Diğer alternatifler için bu değerler ve hesaplama detayları bu tezin son kısmında yer alan EK.A’da verilmiştir.

Avukatlık Ofisi - 1						
	Kriter	Skor Ölçüm Fonksiyonu	X Eksenli Değerleri	Y Eksenli Değerleri	Gerçekleşe n Değer	Hesaplanan Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1
	Denetim Performansı	Type 4	25,50,75,90, 100	0,0.4,0.6,0 .8,1	77	0.8
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.3 3,0	2	0.33
	Dosya Zorluğu	Type 7	1	1	0.4	0.4
	Genel Değerlendirme Puanı	Type 4	25,50,75,90, 100	0,0.4,0.6,0 .8,1	59	0.6

Tablo 9. 11 AHP Uygulaması Hesaplanan Değer

Fonksiyonlarla hesaplanan sonuç değerleri daha önceki adımlarda elde edilmiş ve Tablo 7.7’de gösterilmiş olan ağırlıklarla çarpılır ve her bir kriter için bulunan ağırlıklı değerler toplanarak toplam ağırlıklı performans değeri elde edilir.

$$\begin{aligned} \text{Ağırlık Performans Puanı} &= 0.1 * 0.512 + 0.8 * 0.14 + 0.33 * 0.053 + 0.4 * 0.249 + \\ &0.6 * 0.046 \\ &= 0.051 + 0.112 + 0.017 + 0.100 + 0.028 \\ &= 0.308 \end{aligned}$$

Tüm ofisler için ağırlıklı performans değerleri hesaplanır. Her bir ofis için hesaplamaları içeren tablolar EK.A’da verilmiştir. Hesaplamalar sonucunda alternatiflerin değerler aşağıda gösterilmiştir.

Avukatlık Ofisi	Ağırlıklı Performans Puanı
1	0.308
2	0.378
3	0.251
4	0.293
5	0.276
6	0.287
7	0.313
8	0.269
9	0.338
10	0.240
11	0.303
12	0.301
13	0.284
14	0.267
15	0.261

Tablo 9. 12 AHP - Ağırlıklı Performans Puanları Tablosu

Avukatlık Ofisi	Ağırlıklı Performans Puanı
10	0.240
3	0.251
15	0.261
14	0.267
8	0.269
5	0.276
13	0.284
6	0.287
4	0.293
12	0.301
11	0.303
1	0.308
7	0.313
9	0.338
2	0.378

Tablo 9. 13 AHP - Sıralı Ağırlıklı Performans Puanları Tablosu

9.4 BAHP Uygulaması

BAHP ile yapılan uygulamada kullanılacak kriterler sırasıyla asıl alacak tahsilatı,denetim performansı, müşteri memnuniyeti, dosya zorluğu performansı,genel değerlendirme,borçlunun diğer borçları,gerçekleşen arama sayısı,büro çalışan sayısı ve bürodaki yönetici sayıdır. Bu kriterlerin değerlendirilmesinde kullanılacak olan PROMETHEE fonksiyonları Tablo 9.14’de gösterilmiştir.

PARAMETRE	PROMETHEE FONKSİYONU
Asıl Alacak Tahsilatı	Fonksiyon - 7
Denetim Performansı	Fonksiyon - 4
Müşteri Memnuniyeti	Fonksiyon - 4
Dosya Zorluğu	Fonksiyon - 7
Genel Değerlendirme Puanı	Fonksiyon - 4

Tablo 9. 14 AHP Uygulaması İçin Parametre – Fonksiyon Eşleşmesi

KV’ler tarafından yapılan değerlendirmeler kriterleri arasındaki sözel karşılaştırmalar Tablo 9.15’te gösterilmiştir.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	E	FS	VS	W	A
Denetim Performansı	TFS	E	W	TW	FS
Müşteri Memnuniyeti	TVS	TW	E	TFS	E
Dosya Zorluğu	TW	W	FS	E	FS
Genel Değerlendirme Puanı	TA	TFS	E	TFS	E

Tablo 9. 15 BAHP Uygulaması İçin KV Değerlendirmeleri – Sözel

Sözel Değerlendirme	Kısaltma	Sayısal Değer
Kesin	A	(3,5;4;4,5)
Çok Güçlü	VS	(2,5;3;3,5)
İyice Güçlü	FS	(1,5;2;2,5)
Zayıf	W	(0,67;1;1,5)
Eşit	E	(1;1;1)
Ters Eşit	TE	(1;1;1)
Ters Zayıf	TW	(0,67;1;1,5)
Ters İyice Güçlü	TFS	(0,4;0,5;0,67)
Ters Çok Güçlü	TVS	(0,29;0,33;0,4)
Ters Kesin	TA	(0,22;0,25;0,29)

Tablo 9. 16 BAHP Sözel-Sayısal Değerler Referans Dönüşüm Tablosu

KV'ler tarafından sözel olarak belirtilen karşılaştırma matrisinin hesaplamalar için sayısal değerlerle dönüştürülmüş hali Tablo 9.17'de gösterilmiştir.

	Asıl Alacak Tahsilatı	Denetim Performansı	Müşteri Memnuniyeti	Dosya Zorluğu	Genel Değerlendirme Puanı
Asıl Alacak Tahsilatı	(1;1;1)	(1,5;2;2,5)	(2,5;3;3,5)	(0,67;1;1,5)	(3,5;4;4,5)
Denetim Performansı	(0,4;0,5;0,67)	(1;1;1)	(0,67;1;1,5)	(0,67;1;1,5)	(1,5;2;2,5)
Müşteri Memnuniyeti	(0,29;0,33;0,4)	(0,67;1;1,5)	(1;1;1)	(0,4;0,5;0,67)	(1;1;1)
Dosya Zorluğu	(0,67;1;1,5)	(0,67;1;1,5)	(1,5;2;2,5)	(1;1;1)	(1,5;2;2,5)
Genel Değerlendirme Puanı	(0,22;0,25;0,29)	(0,4;0,5;0,67)	(1;1;1)	(0,4;0,5;0,67)	(1;1;1)

Tablo 9. 17 AHP Uygulaması İçin KV Değerlendirmeleri – Sayısal

Tüm satırlar için aynı satırda yer alan değerlerin sağ parametresi, orta parametresi ve sol parametrelerinin geometrik ortalaması alınır. Tablo 7.15 referans alınarak yapılacak hesaplama aşağıdaki gibidir.

$$[0][0] = (1*1.5*2.5*0.67*3.5)^{1/5} = 1.545$$

$$[0][1] = (1*2*3*1*4)^{1/5} = 1.888$$

$$[0][2] = (1*2.5*3.5*1.5*4.5)^{1/5} = 2.26$$

$$[1][0] = (0.4*1*0.67*0.67*1.5) = 0.769$$

$$[1][1] = (0.5*1*1*1*2) = 1$$

$$[1][2] = (0.67*1*1.5*1.5*2.5) = 1.304$$

$$[2][0] = (0.29*0.67*1*0.4*1) = 0.600$$

$$[2][1] = (0.33*1*1*0.5*1) = 0.697$$

$$[2][2] = (0.4*1.5*1*0.67*1) = 0.833$$

$$[3][0] = (0.67*0.67*1.5*1*1.5) = 1$$

$$[3][1] = (1*0.67*1.5*1*1.5) = 1.319$$

$$[3][2] = (1.5*1.5*2.5*1*2.5) = 1.697$$

$$[4][0] = (0.22*0.4*1*0.4*1) = 0.512$$

$$[4][1] = (0.25*0.5*1*0.5*1) = 0.574$$

$$[4][2] = (0.29*0.67*1*0.67*1) = 0.665$$

Bu hesaplamalar sonucunda elde edilen matris tablo 9.18’de gösterilmiştir.

1.545	1.888	2.26
0.769	1	1.304
0.6	0.697	0.833
1	1.319	1.697
0.512	0.574	0.665

Tablo 9. 18 BAHP Geometrik Ortalama Matrisi

Sütun toplamaları alınarak -1. kuvvetleri alınır. Sonrasında sütunlar 1. sütunun toplamının tersi son sütuna denk gelecek şekilde yeni bir satır oluşturulur. Bu değerler normalizasyon katsayılarıdır. Yapılan işlemler sonucunda oluşan matris Tablo 9.19’da gösterilmiştir.

$$\sum \text{Sütun1} = 1.545 + 0.769 + 0.6 + 1 + 0.512 = 4.428$$

$$\sum \text{Sütun2} = 1.888 + 1 + 0.697 + 1.319 + 0.574 = 5.479$$

$$\sum \text{Sütun3} = 2.26 + 1.304 + 0.833 + 1.697 + 0.665 = 6.76$$

$$\text{Sütun 1} = 1 / 4.428 = 0.225841$$

$$\text{Sütun 2} = 1 / 5.479 = 0.182500$$

$$\text{Sütun 3} = 1 / 6.76 = 0.147930$$

0.147930	0.182500	0.225841
----------	----------	----------

Tablo 9. 19 BAHP Sütun Normalizasyon Katsayıları

Geometrik ortalama matrisi ile normalizasyon katsayıları çarpılarak matris normalize edilir.

$1.545 * 0.147930$	$1.888 * 0.1825$	$2.26 * 0.225841$
$0.769 * 0.147930$	$1 * 0.1825$	$1.304 * 0.225841$
$0.6 * 0.147930$	$0.697 * 0.1825$	$0.833 * 0.225841$
$1 * 0.147930$	$1.319 * 0.1825$	$1.697 * 0.225841$
$0.512 * 0.147930$	$0.574 * 0.1825$	$0.665 * 0.225841$

0.2285	0.3445	0.5105
0.1138	0.1825	0.2944
0.0887	0.1272	0.1882
0.1482	0.2408	0.3832
0.0757	0.1048	0.1502

Tablo 9. 20 BAHP Sütun Normalize Tablo

Tablo 9.20’de gösterilen matriste her satırın ortalaması alınır.

$$\text{Satır1} = (0.2285 + 0.3445 + 0.5105) / 3 = 0.361225$$

$$\text{Satır2} = (0.1138 + 0.1825 + 0.2944) / 3 = 0.196921$$

$$\text{Satır 3} = (0.0887 + 0.1272 + 0.1882) / 3 = 0.134747$$

$$\text{Satır 4} = (0.1482 + 0.2408 + 0.3832) / 3 = 0.257409$$

$$\text{Satır 5} = (0.0757 + 0.1048 + 0.1502) / 3 = 0.110260$$

0.361225
0.196921
0.134747
0.257409
0.110260

Tablo 9. 21 BAHP Satır Normalizasyon Değerleri

Her bir değer bu değerlerin toplamına bölünerek normalize edilir ve değerlendirme kriterlerinin ağırlıkları bulunur. Bulunan ağırlıklar Tablo 9.22’de gösterilmiştir.

$$0.361225 + 0.196921 + 0.134747 + 0.257409 + 0.110260 = 1.060562$$

$$0.361225 / 1.060562 = 0.340598$$

$$0.196921 / 1.060562 = 0.185676$$

$$0.134747 / 1.060562 = 0.127052$$

$$0.257409 / 1.060562 = 0.242710$$

$$0.110260 / 1.060562 = 0.103964$$

Kriter	Ağırlık
Asıl Alacak Tahsilatı	0.341
Denetim Performansı	0.186
Müşteri Memnuniyeti	0.127
Dosya Zorluğu	0.243
Genel Değerlendirme Puanı	0.104

Tablo 9. 22 BAHP – Kriter ağırlıkları

Avukatlık Ofisi - 1						
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	77	0.8
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33
	Dosya Zorluğu	Type 7	1	1	0.4	0.4
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	59	0.6

Tablo 9. 23 BAHP Uygulaması Hesaplanan Değer

Fonksiyonlarla hesaplanan sonuç değerleri daha önceki adımlarda elde edilmiş ve Tablo 9.22’de gösterilmiş olan ağırlıklarla çarpılır ve her bir kriter için bulunan ağırlıklı değerler toplanarak toplam ağırlıklı performans değeri elde edilir.

$$\begin{aligned}
\text{Ağırlık Performans Puanı} &= 0.1 * 0.341 + 0.8 * 0.186 + 0.33 * 0.127 + 0.4 * 0.243 + \\
&0.6 * 0.104 \\
&= 0.0341 + 0.149 + 0.042 + 0.097 + 0.062 \\
&= 0.384
\end{aligned}$$

Tüm ofisler için ağırlıklı performans değerleri hesaplanır. Her bir avukatlık için yapılan hesaplamaları içeren tablolar EK.B’de verilmiştir. Hesaplamalar sonucunda alternatiflerin değerler aşağıda gösterilmiştir.

Avukatlık Ofisi	Ağırlıklı Performans Puanı
1	0.384
2	0.474
3	0.274
4	0.405
5	0.312
6	0.355
7	0.419
8	0.331
9	0.377
10	0.265
11	0.324
12	0.315
13	0.408
14	0.384
15	0.333

Tablo 9. 24 BAHP - Ağırlıklı Performans Puanları Tablosu

Avukatlık Ofisi	Ağırlıklı Performans Puanı
10	0.265
3	0.274
5	0.312
12	0.315
11	0.324
8	0.331
15	0.333
6	0.355
9	0.377
14	0.384
1	0.384
4	0.405
13	0.408
7	0.419
2	0.474

Tablo 9. 25 BAHP - Sıralı Ağırlıklı Performans Puanları Tablosu

9.5 AHP ve BAHP Değerlerinin Karşılaştırılması

AHP ve BAHP uygulanarak elde edilen sonuçlar burada birlikte gösterilmiştir. Tablo 9.26’da her iki yöntem ile aynı girdiler kullanılarak elde edilen ağırlık değerleri gösterilmiştir. Tablo 9.27’de ise her iki yöntem ile elde edilen ağırlıklar kullanılarak elde edilen avukatlık ofisleri performans puanları gösterilmiştir.

Kriter	AHP Ağırlıkları	BAHP Ağırlıkları
Asıl Alacak Tahsilatı	0,512	0.341
Denetim Performansı	0,14	0.186
Müşteri Memnuniyeti	0,053	0.127
Dosya Zorluğu	0,249	0.243
Genel Değerlendirme Puanı	0,046	0.104

Tablo 9. 26 AHP ve BAHP Ağırlık Karşılaştırmaları

Avukatlık Ofisi	AHP Ağırlıklı Performans Puanı	BAHP Ağırlıklı Performans Puanı
1	0.308	0.384
2	0.378	0.474
3	0.251	0.274
4	0.293	0.405
5	0.276	0.312
6	0.287	0.355
7	0.313	0.419
8	0.269	0.331
9	0.338	0.377
10	0.240	0.265
11	0.303	0.324
12	0.301	0.315
13	0.284	0.408
14	0.267	0.384
15	0.261	0.333

Tablo 9. 27 AHP ve BAHP Kullanılarak Performans Puanı Karşılaştırmaları

SONUÇ VE ÖNERİLER

Ödenmeyen borçların miktarının artış gösterdiği günümüz dünyasında bu borçların tahsil edilmesi alacaklı firmalar için büyük önem arz etmektedir. Borçların yasal yollarla tahsil süreci uzmanlık gerektirmektedir. Alacaklı firmalar bu süreçleri takip etmek için uzman kadrolardan oluşan birimler tahsis edebileceği gibi bu konuda uzmanlaşmış dış kaynaklardan da faydalanabilirler. Yasal takip sürecindeki borçların tahsil sürecinde de konu hakkındaki uzmanlıkları, alacak miktarına göre ayrılan kaynağın değiştirilebilmesi ve maliyet avantajları nedeniyle dış kaynaklar tercih edilebilir olmaktadır.

Yasal takip sürecinde avukatlık ofisleri ile dış kaynak olarak çalışılmaktadır. Alacaklı firmalar, yasal takip süreçlerinden en yüksek seviyede tahsilat yapmak amacıyla hareket etmektedir. Bu nedenle, bu süreç için birlikte çalıştıkları avukatlık ofislerinin performansları alacaklı firmalar için çok önemlidir. Avukatlık ofislerinin göstermiş oldukları performansa göre yasal takip sürecine girmiş borç dosyalarının bu ofisler arasındaki dağılımını belirlemek istemektedirler.

Firmalar, birlikte çalıştıkları avukatlık ofislerinden beklentilerini temel olarak bilmekle birlikte avukatlık ofisleri arasında sayısal performans ölçümünü yapmakta zorlanmaktadırlar. Hangi başlıklarda ne tür beklentileri olduklarını bilmekte fakat bu beklentilerinin avukatlık ofislerinin performans puanına ne derece etki ettiğini belirleyememektedirler. Kriterler arasında sözel olarak bir karşılaştırma yapabilmelerine rağmen bunları net sayısal ağırlıklarla ifade edememektedirler.

Bu nedenlerle, Tubitak tarafından TEYDEB 1505 programı ile desteklenen “Akıllı Çok Ölçütlü Yasal Takip Avukatlık Ofisi Notlandırma ve Performans Yönetimi Sistemi” projesi kapsamında yapmış olduğumuz bu tez çalışmasında, Türkiye’deki GSM operatörlerinden biriyle ortak çalışılarak tahsil edemedikleri borçları için yasal takip sürecinde dış kaynak olarak çalıştıkları avukatlık ofislerinin performansını ölçmek ve ofisler arası karşılaştırma yapmak için çok ölçütlü bir model geliştirilmiştir.

Geliştirilen çözüm modelinde, sözel ifadelerle değerlendirmeyi mümkün kılması nedeniyle analitik hiyerarşi prosesi ve sözel ifadeler ile çalışabilmesinin yanı sıra belirsizliği minimize etmesi nedeniyle bulanık analitik hiyerarşi prosesi

kullanılmıştır. Böylece karar vericilerin performans değerlendirmede belirlenen kriterler arasındaki ilişkileri sayısal olarak ifade edebilmeleri sağlanmıştır.

Geliştirilen çözüm modeli, yine tez kapsamında geliştirilen yazılım uygulaması ile otomatize edilmiştir. Yazılım uygulaması, hesaplama işlemlerinin süresini kısalttığı gibi hesaplama hatalarını da ortadan kaldırmaktadır. Karar vericiler süreç içerisinde kriterler arasındaki önem derecelerinin değiştirebilir ya da değişen şartlar; kriterlerin arasındaki ilişkilerin değişmesine sebep olabilir. Geliştirilen yazılım uygulaması istenildiği sayıda karşılaştırma kriteri oluşturularak kullanılabilmesi sayesinde karar vericiye esnek değerlendirme fırsatı tanımaktadır.

Tez çalışması kapsamında yapılan uygulamada 15 tane avukatlık ofisi için 5 farklı kritere göre hesaplama yapılmıştır. Bunun için öncelikle kriterlerin oluşturulup, ağırlıkların hesaplanması ve her bir ofisin her bir kritere göre gerçekleşen değerinin uygun fonksiyon kullanarak hesaplanması ve en son da hesaplanan değerlerin ağırlıklandırılarak performans notuna dönüştürülmesi gerçekleştirilmiştir. Böylece insan eliyle gerçekleştirildiğinde hem uzun hem de hataya açık olan bir süreç, yazılım aracılığıyla sistematik bir hale getirilmiştir.

Geliştirilen çözüm modeli değiştirebilir sayıda parametre alması ve değerlendirme kriterleri arasındaki ilişkilerin değiştirilebilir olması sayesinde farklı alanlarda da uygulanabilir. Çözüm modeli karar vericiye elindeki alternatifleri hızlı bir biçimde değerlendirebilmesi için matematiksel bir veri sağlar. Ayrıca yazılım uygulaması sayesinde karar verici hesaplama detaylarından da soyutlanır.

Daha önce avukatlık ofislerinin dış kaynak olarak performanslarının ölçülmesi ve karşılaştırılması ile ilgili yapılan çalışmalarda bulanık analitik hiyerarşi prosesi (z bulanık ahp ve sezgisel bulanık ahp) kullanılmıştı. Bu tez çalışmasında hem analitik hiyerarşi prosesi hem de bulanık analitik hiyerarşi prosesi kullanılmıştır. Ayrıca bu iki yöntemden elde edilen sonuçlar arasında karşılaştırma yapılmıştır.

Yapılan uygulamada avukatlık ofislerinin yasal takip sürecindeki performansları “Asıl Alacak Tahsilatı”, “Denetim Performansı”, “Müşteri Memnuniyeti”, “Dosya Zorluğu” ve “Genel Değerlendirme Puanı” olmak üzere 5 kriter üzerinden değerlendirilmiştir. Bu kriterlerin gerçekleşen puanları PROMETHEE fonksiyonları kullanılarak hesaplanmıştır.

AHP ve BAHP uygulanarak hesaplanan kriter ağırlıkları Tablo 9.26’da gösterilmiştir. Bu ağırlıklar kullanılarak hesaplanan avukatlık ofisleri performans puanları ise Tablo9.27’de gösterilmiştir.

Ağırlıkların iki yöntemde farklı çıkmasına bağlı olarak avukatlık ofislerinin sıralamaları değişiklik göstermiştir. Her ne kadar kriter ağırlıkları farklı da olsa kriterlerin ağırlık sıralamaları her iki yöntemde de aynı çıkmaktadır. Önem sıralaması yaptığımızda kriterler “Asıl Alacak Tahsilatı”, “Dosya Zorluğu”, ”Denetim Performansı”, ”Müşteri Memnuniyeti” , “Genel Değerlendirme Puanı” şeklinde sıralanmaktadır.

Yapılan çalışmada yasal takip sürecindeki dosyaların takibi için çalışılan dış kaynak avukatlık ofislerinin performansının hem analitik hiyerarşi süreci hem de bulanık analitik hiyerarşi süreci ile çözülebileceği gösterilmiştir. Ayrıca, süreç ve buna bağlı olarak geliştirilen yazılım parametrik olduğundan farklı alanlarda farklı parametrelerle aynı süreç kullanılarak performans hesaplaması ve sıralaması yapılabilir.

Tez çalışması esnasında dış kaynak avukatlık ofisleri ile ilgili az sayıda çalışma olması nedeniyle kriterlerin belirlenmesi noktasında zorluk yaşandığından konunun uzmanları ile defalarca yüz yüze görüşmeler yapılarak veri toplanmıştır. Daha sonrasında çok sayıdaki farklı veri değerlendirilerek beş adet ortak kriterin belirlenmesi sağlanmıştır. Çözüm modeli oluşturulduktan sonra bu modelin matematiksel ifadelerinin, hesaplamalarının yazılım içerisinde gerçekleştirilebilmesi için algoritmaya dönüştürülmesi süreci ve yazılım geliştirildikten sonra birlikte çalışılan GSM firmasının sistemine entegrasyonunda çeşitli zorluklar yaşanmıştır. Bu zorlukların aşılması için GSM firması ile bir süre birlikte çalışılmıştır.

Bu çalışmadan faydalanarak kriterler alt kriterlere bölünerek daha detaylı irdelemeler yapılabilir, AHPve BAHP dışında farklı karar verme yöntemleri yazılım uygulamasına eklenerek karar vericiye farklı alternatifleri görme fırsatı sunulabilir.

Bu tezin sonucunda, bu alanda yapılacak çalışmalar için araştırmacıların AHP yerine BAHP kullanması kanaatine varılmıştır.



KAYNAKLAR

- [1] **Kahraman, C., Öztayşi, B. ve Cevik Onar, S.** (2019). Performance Comparisons of Law Offices and Optimum Allocation of Debt Files Using Z-Fuzzy AHP, 11th Conference of the European Society for Fuzzy Logic and Technology.
- [2] **Tanyeri, Mustafa. ve Fırat, Aytekin.** (2005), Rekabet Değişkeni Olarak Dış Kaynak Kullanımı, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, Cilt:7, Sayı:3
- [3] **Cevik Onar, S., Öztayşi ve B., Kahraman, C.** (2018), A Fuzzy Rule Based System for Early Debt Collection, Technological and Economic Development of Economy, Vol.24 : 1845 – 1865
- [4] **Łada, Monika. ve Wejer-Kudelko, Małgorzata.** (2018), SUCCESS FACTORS AND BARRIERS TO THE EFFECTIVE DEBT COLLECTION PROCESS, RESEARCH PAPERS OF WROCLAW UNIVERSITY OF ECONOMICS, Vol. 515
- [5] **Beck, Timo., Grunert, Jens. ve Neus, Werner.** (2017), What Determines Collection Rates of Debt Collection Agencies?, The Financial Review 52 , 259–279
- [6] **Phillips, Lisa. ve Moggridge, Paul.** (2019), Artificial Intelligence in Debt Collection, Credit Control. 2019, Vol. 40 Issue 3/4, p15-27. 13p.
- [7] **Kaptanoğlu, D.** (2004), Akademik Performans Değerlendirmesi İçin Bir Çok Ölçütlü Bulanık Karar Verme Modeli, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul
- [8] **Keleş, Mehmet Özkan.** (2018), Türk koster işletmeciliğinde üçüncü taraf ve tam kurum içi yönetimler arası gemicilik performans endekslerine dayalı değerlendirme, İ.T.Ü. Denizcilik Fakültesi, İstanbul
- [9] **BUCOVETCHI ,Olga., STANCIU, Radu D. ve BADEA, Dorel.** (2019), Is outsourcing the best solution in increasing organisational performance? , Quality-Access to Success, 20 (S1), January 2019
- [10] **Bernard, Andrew B., Moxnes, Andreas., Saito ve Yukiko U.** (2019), Production Networks, Geography, and Firm Performance, Journal of Political Economy, Apr2019, Vol. 127 Issue 2, p639-688
- [11] **Świtala, Marcin., Cichosz, Marzenna. ve Trześciok Joanna.** (2019), HOW TO ACHIEVE CUSTOMER SATISFACTION? PERSPECTIVE OF LOGISTICS OUTSOURCING PERFORMANCE, Scientific Journal of Logistics, 2019, 15 (1), 39-51
- [12] **Espino-Rodríguez, Tomás F., Ramírez-Fierro. ve Juan Carlos.** (2018), Outsourcing Performance in Hotels: Evaluating Partnership Quality, Sustainability (2071-1050), Aug2018, Vol. 10 Issue 8, p2766
- [13] **Prajapati ,Himanshu., Kant, Ravi. ve Tripathi, Sandeep Mani.** (2020), An integrated framework for prioritizing the outsourcing performance outcomes, Journal of Global Operations and Strategic Sourcing, 2020
- [14] **ÖZTÜRK, Hakan., MURAT, Naci. ve ELEVLİ, Sermin.,** (2019), QUALITY CONTROL CHARTS FOR MONITORING PERFORMANCE OF HOSPITAL CALL CENTER, Sigma Mühendislik ve Fen Bilimleri Dergisi, 37 (4), 2019, 1393-1406

- [15] **BULUT,Umut. ve ATAAY, Aylin.** (2017), Çağrı Merkezlerinde Çalışan Performansını Etkileyen Faktörler: Bir İnceleme , İşletme Araştırmaları Dergisi, 2017, Vol 9, Iss 4, Pp 1-16
- [16] **Karakus, Betul. ve Aydın, Galip.** (2016), Call center performance evaluation using big data analytics, International Symposium on Networks, Computers and Communications, 2016
- [17] **Hsu, Hui-Huang., Chen, Te-Chan., Chan, Wei-Tsung. Ve Chang, Jung-Kuei.** (2016), Performance Evaluation of Call Center Agents by Neural Networks, 30th International Conference on Advanced Information Networking and Applications Workshops, 2016
- [18] **Lopes, Susana Almeida., Duarte , Maria Eduarda. ve Lopes, João Almeida.** (2018), Can artificial neural networks predict lawyers' performance rankings?, International Journal of Productivity and Performance Management, 2018, Vol. 67, Issue 9, pp. 1940-1958.
- [19] Key Steps to Grow a Performance Culture in Your Law Firm, Compensation & Benefits for Law Offices
- [20] **Kahraman, C., Öztayşi, B. ve Cevik Onar, S.** (2020), Single & interval-valued neutrosophic AHP methods: Performance analysis of outsourcing law firms, Journal of Intelligent & Fuzzy Systems 38 (2020) 749–759
- [21] **Aytaç, Mustafa. ve Gürsakal, Necmi.** (2015), Karar Verme, ISBN:978-605-9929-27-1
- [22] **Evren,R. ve Ülegin,F.** (1992a), Yönetimde Çok Amaçlı Karar Verme, İTÜ Matbaası, No.1490, İstanbul
- [23] **Demir,M.H.,Bircan, B. Ve Tütek,H.** (1985) Yönetimsel Karar Verme, Bilgehan Basımevi, İzmir
- [24] **Özden, K.** (1989), Yöneylem Araştırması, Hava Harp Okulu Yayınları, İstanbul.
- [25] **Çalikyay,S.** (2002), Çok Amaçlı Savaş Uçağı Seçiminde Çok Ölçütlü Karar Verme Yöntemlerinin Uygulanması, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul
- [26] **Yayan,İ.** (2001), Türk Hava Kuvvetlerinde Taarruz Taktik Keşif Helikopteri Etkinlik Değerlendirmesi ve Seçiminde Analitik Hiyerarşi Yaklaşımı, Yüksek Lisans Tezi, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul
- [27] **Koçel,Tamer.** (2001), İşletme Yöneticiliği, 8. Baskı Beta Basım Yayın Dağıtım A.Ş., 2001.
- [28] **Topçu,Y.İ.** (2000), Çok Ölçütlü Sorun Çözümüne Yönelik Bir Bütünleşik Karar Destek Modeli, İ.T.Ü. Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul
- [29] **Onaran, O.** (1971), Örgütlerde Karar Verme, SBF Yayınları, No:321, Ankara
- [30] **Zionts , S.** (1994), The challange that lies ahead, Proceeding of the 10th Multiple Criteria Decision Making International Conference, Taipei, Taiwan, 1992, 17-27.
- [31] **Zionts , S.** (1997), Decision making:Some experiences, myths and observations, Proceeding of the 12th Multiple Criteria Decision Making International Conference, Hagen, Germany, 1995, 233-241.

- [32] **Vincke, Ph.** (1992), Multi Criteria Decision Aid., John Wiley&Sons, Inc., West Sussex.
- [33] **Evren, R. ve Ülegin, F.** (1992b), Yönetimde Çok Amaçlı Karar Verme, İTÜ Matbaası, No.1490, İstanbul
- [34] **Rammond, J.S., Keeney, R.L. and Raiffa, H.** (1999), Smart Choices, Harvard Business School Press, Boston
- [35] **Yıldırım, B. ve Önder, E.** (2015), Çok Kriterli Karar Verme Yöntemleri, Dora, 2015
- [36] **Yoon, K.P. and Hwang C-L.** (1995), Multi Attribute Decision Making: An Introduction. Sage University Papers Series, Quantitative Applications in the Social Sciences, No 07-104, Sage Pubn., London.
- [37] **Korhonen, P., Moskowitz, H. and Wallenius, J.** (1992), Multiple Criteria Decision Making - A review, Eur. J. Oper. Res., 63, 361-375.
- [38] **Meridith, J.R and Suresh, N.C.** (1986), Justification Techniques For Advanced Manufacturing Technologies, International Journal Of Production Research, Vol. 24, 1043-1057.
- [39] **Lavelle, J.R. and Lingett, H.R.** (1992), Economic Methods For Evaluating Investment in Advanced Manufacturing Technologies, Economic And Financial Justification Of Advanced Manufacturing Technologies.
- [40] **Kolli, S. and Parsei, H.R.** (1992), Multi- Criteria Analysis in the Evaluation of Advanced Manufacturing Technology Using PROMETHEE, 14th Conference of Computers and Industrial Engineering, Florida.
- [41] **Tekeş, M.** (2002), Çok Ölçütlü Karar Verme Yöntemleri Ve Türk Silahlı Kuvvetleri'nde Kullanılan Tabancaların Bulanık Uygunluk İndeksli Analitik Hiyerarşi Prosesi İle Karşılaştırılması, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul
- [42] **Fishburn, P.C.** (1976), Utility Independence on Subsets of Product Sets, Operations Research 24, pp. 245-255.
- [43] **Edwards, W.** (1977), Use of Multiattribute Utility Measurement for Social Decision Making, In D.E. Bell, R.N. Keeney, H. Raiffa (Eds.), Conflicting Objectives in Decisions (pp. 247-276), New York: Wiley.
- [44] **Farmer, T.A.** (1987), Testing the Robustness of Multiattribute Utility Theory in An Applied Setting, Decision Sciences 18, pp. 178-193.
- [45] **CHEN Shu-.Jen. and HWANG, Ching, Lai.** (1992), Fuzzy Multiple Attribute Decision Making: Methods and Applications, Springer –Verlag- Berlin
- [46] **YURDAKUL, Mustafa. ve İÇ TANSEL, Yusuf.** (2003), Türk Otomotiv Firmalarının Performans Ölçümü ve Analizine Yönelik TOPSIS Yöntemini Kullanan Bir Örnek Çalışma, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, Cilt. 18, No:1, ss 1-18
- [47] **Roy, B.** (1971), Problems and Methods with Multiple Objective Functions, Mathematical Programming 1, pp. 239-266.

- [48] **Nijkamp, P. and Delft, A.** (1977), Multi-Criteria Analysis and Regional Decision Making, Leiden, The Netherlands: Martinus Nijhoff.
- [49] **Voogd, H.** (1983), Multicriteria Evaluation for Urban and Regional Planning, London: Pion.
- [50] **Anderson, D.R., Sweeney, D.J., Williams, T.A, et al.** (2008), Quantitative Methods for Business, 11th edition, South-West Cengage Learning
- [51] **Timor, M.** (2011), Analitik Hiyerarşi Prosedürü, Türkmen Kitabevi, İstanbul
- [52] **Saaty, Thomas L.** (2008), The Analytic Hierarchy and Analytic Network Measurement Processes; Applications to Decisions Under Risk. European Journal of Pure and Applied Mathematics, Vol.1, No.1, 122-196
- [53] **Şen, Z.** (2011), Bulanık Mantık Ve Modelleme İlkeleri, Bilgi Kültür Sanat Yapım Yayınevi, İstanbul
- [54] **Kaufmann, A. ve Gupta, M.** (1988), Fuzzy mathematical models in engineering and management science, North-Holland Book Co, Amsterdam, s:338
- [55] **Ross, T.J.** (1995), Fuzzy logic with engineering applications, Mc-Graw Hill, New York.
- [56] **Kahraman, C.** (1995) İleri İmalat Teknolojilerinin Ekonomik Analizi ye Esneklik Faktörünün Sayısallaştırılmasına Bulanık Külmeler Yaklaşımı, İ.T.Ü. Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- [57] **Elmas, Ç.** (2003) Bulanık Mantık Denetleyiciler, Seçkin Yayıncılık San.Tic.A.Ş.
- [58] **Kurtcan, E.** (2009). Yalın Lojistik Tabanlı Sistemin Bağımsızlık ye Bilgi Aksiyonları Kullanılarak Tasarlanması ye Bir Firma Uygulaması, (Doktora Tezi), Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul
- [59] **Zadeh, L.A.** (1965), Fuzzy Sets. Information and Control 8, 338–353
- [60] **Aliev, Refik Aziz.** (2013), Fundamentals of the Fuzzy Logic-Based Generalized Theory of Decisions, Springer
- [61] **Url-1** <http://Websitem.karatekin.edu.tr>, date retrieved 10.06.2020
- [62] **Çitli, Neslihan.** (2006), Bulanık Çok Kriterli Karar Verme, Yıldız Teknik Üniversitesi Fen Bilimleri enstitüsü, İstanbul
- [63] **Prodanovic, P. ve Simonovic, S.P.** (2002), Comparison of fuzzy set ranking methods for implementation in water resources decision-making, Canadian Journal of Civil Engineering, 29, 692-701
- [64] **Liou, T.S. ve Wang, M.J.** (1992), Ranking Fuzzy Numbers With İntegral Value, Fuzzy Sets and Systems, 11, 229-241.
- [65] **Abdel-Kader, M., Dugdale, D. ve Taylor, P.** (1998), Investment Decisions in Advanced Manufacturing Technology: A Fuzzy Set Theory Approach, Aldershot, Ashgate.
- [66] **Kücü, H.** (2007). PROMETHEE sıralama yöntemi ile personel seçimi ve bir işletmede uygulanması, (Yayınlanmış Yüksek Lisans Tezi), Gazi Üniversitesi Fen Bilimleri Enstitüsü

- [67] **Saaty, T.L.** (1989), Multicriteria Decision Making: The Analytic Hierarchy Process. RWS Publ., Pittsburg.
- [68] **Saaty, T.L.** (1996), Decision Making for Leaders. RWS Publ., Pittsburg.
- [69] **Pamukçu, B.** (2004), Analitik ağ süreci ve bir uygulama, (Yüksek Lisans Tezi), İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, İstanbul.
- [70] **Triantaphyllou, E.** (2000), Multi-criteria decision making methods: a comparative study, Kluwer Academic Publishers, Dordrecht.
- [71] **Canada, J.R., Sullivan, W.G. and White, J.A.** (1996), Capital Investment Analysis for Engineering and Management. Pretice Hall, London.
- [72] **Saaty, T.L.** (1980), The Analytic Hierarchy Process , McGraw Hill International.
- [73] **Saaty, T.L.** (1990), An exposition of the AHP, Reply to Remarks on the Analytic Hierarchy Process by Dyer, J.S., Management Science, 36/3, 259-268.
- [74] **Kahraman, C., Cebeci, U., and Ruan, D.** (2004), Multi-attribute comparison of catering service companies using fuzzy AHP: The case of TURKEY, International Journal of Production Economics, 87: 171–184.
- [75] **Boenders, C.G.E., Graan, J.G., ve Lootsma, F.A.** (1989), Multicriteria decision analysis with fuzzy pairwise comparisons, Fuzzy Sets and Systems, 29, 133-143.
- [76] **Büyüközkan, G., Kahraman, C., ve Ruan, D.** (2004), A fuzzy multi-criteria decision approach for software development strategy selection, International Journal of General Systems, Vol. 33 (2-3), pp. 259-280
- [77] **B. Zhu, and Z. Xu,** Analytic hierarchy process-hesitant group decision making, European Journal of Operational Research, vol. 239, pp. 794–801, 2014.
- [78] **J.J. Buckley.** (1985), Fuzzy hierarchical analysis, Fuzzy Sets and Systems, vol. 17, pp. 233–247, 1985.
- [79] **Oztaysi, Basar; Onar, Sezi Cevik; Bolturk, Eda ve Kahraman, Cengiz.** (2015), HESITANT FUZZY ANALYTIC HIERARCHY PROCESS, IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) , 2015, p1-7, 7p
- [80] **Kutlu Gündoğdu, Fatma ve Kahraman, Cengiz.** (2019), A novel spherical fuzzy analytic hierarchy process and its renewable energy application, In: Soft Computing: A Fusion of Foundations, Methodologies and Applications, 2020, 24(6):4607-4621, Springer Berlin Heidelberg
- [81] **Öztaysi, Başar., Cevik Onar, Sezi. Ve Kahraman, Cengiz.** (2019), Performance Measurement Model for Software Development Teams Using Interval-valued Intuitionistic Fuzzy Analytic Hierarchy Process”, J. of Mult.-Valued Logic & Soft Computing, Vol. 33, pp. 321–339
- [82] **Yager, R.R.** (2013) Pythagorean fuzzy subsets. In: 2013 Joint IFSA world congress and NAFIPS annual meeting (IFSA/NAFIPS). IEEE, pp 57–61
- [83] **Karasan, Ali., Ilbahar, Esra. ve Kahraman, Cengiz.** (2018), A novel pythagorean fuzzy AHP and its application to landfill site selection problem, Soft Computing (2019) 23:10953–10968

- [84] **Kahraman, C., Öztayşi, B., Cevik Onar, S. ve Ilbahar, E.** (2019), Selection Among GSM Operators Using Pythagorean Fuzzy WASPAS Method, J. of Mult.-Valued Logic & Soft Computing, Vol. 33, pp. 459–469
- [85] **Guleria, Abhishek., Bajaj, Rakesh Kumar. ve Kahraman, Cengiz.** (2020), Journal of Intelligent & Fuzzy Systems; 2020, Vol. 38 Issue 1, p761-777, 17p , Pythagorean fuzzy (R, S)-norm discriminant measure in various decision making processes.
- [86] **Smarandache, F.** (2006), Neutrosophic set – a generalization of the intuitionistic fuzzy set, IEEE Int Conf Granular Comput 38–42
- [87] **Karasan, Ali. ve Kahraman, Cengiz.** (2018), A novel interval-valued neutrosophic EDAS method: prioritization of the United Nations national sustainable development goals, Soft Computing, Vol. 22, pages 4891–4906
- [88] **Yörükoğlu, Mehmet., Aydın, Serhat. ve Kahraman, Cengiz.** (2020), Smart container evaluation by neutrosophic MCDM method, Journal of Intelligent & Fuzzy Systems; 2020, Vol. 38 Issue 1, p723-733, 11p
- [89] **Karasan, Ali., Boltürk, Eda. ve Kahraman, Cengiz.** (2020), An Integrated Interval-Valued Neutrosophic AHP and TOPSIS Methodology for Sustainable Cities' Challenges, Intelligent and Fuzzy Techniques in Big Data Analytics and Decision Making : Proceedings of the INFUS 2019 Conference, Istanbul, Turkey, July 23-25, 2019. :653-661
- [90] **Otay, I. Ve Kahraman, C.** (2018), Solar PV power plant location selection using a Z-fuzzy number based AHP, International Journal of the Analytic Hierarchy Process, 2018, 10(3):409-430
- [91] **Tüysüz, Nurdan. ve Kahraman, Cengiz.** (2020), CODAS method using Z-fuzzy numbers, Journal of Intelligent & Fuzzy Systems. 2020, Vol. 38 Issue 2, p1649-1662. 14p.

EKLER**EK A. Analitik Hiyerarşi Prosesi Kullanılarak Yapılan Avukatlık Ofisi Performans Hesaplamaları**

Avukatlık Ofisi - 1								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.512	0.051
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	77	0.8	0.140	0.112
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.053	0.017
	Dosya Zorluğu	Type 7	1	1	0.4	0.4	0.249	0.1
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	59	0.6	0.046	0.028
	Ağırlıklı Performans Puanı							0.308

Avukathlık Ofisi - 2								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.14	0.14	0.512	0.072
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	52	0.6	0.140	0.084
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.053	0.036
	Dosya Zorluğu	Type 7	1	1	0.599	0.599	0.249	0.149
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	89	0.8	0.046	0.037
	Ağırlıklı Performans Puanı							0.378

Avukathlık Ofisi - 3								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.13	0.13	0.512	0.067
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	20	0	0.140	0
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.053	0.017
	Dosya Zorluğu	Type 7	1	1	0.599	0.599	0.249	0.149
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	46	0.4	0.046	0.018
	Ağırlıklı Performans Puanı							0.251

Avukathlık Ofisi - 4								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.059	0.059	0.512	0.031
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	29	0.4	0.140	0.056
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.053	0.053
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.249	0.125
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	58	0.6	0.046	0.028
	Ağırlıklı Performans Puanı							0.293

Avukathık Ofisi - 5								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.19	0.19	0.512	0.097
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	25	0	0.140	0
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.053	0.017
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.249	0.125
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	90	0.8	0.046	0.037
	Ağırlıklı Performans Puanı							0.276

Avukathık Ofisi - 6								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.179	0.179	0.512	0.092
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	50	0.4	0.140	0.056
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.053	0.036
	Dosya Zorluğu	Type 7	1	1	0.29	0.29	0.249	0.075
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	61	0.6	0.046	0.028
	Ağırlıklı Performans Puanı							0.287

Avukathık Ofisi - 7								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.512	0.051
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	29	0.4	0.140	0.056
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.053	0.053
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.249	0.125
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	65	0.6	0.046	0.028
	Ağırlıklı Performans Puanı							0.313

Avukathlık Ofisi - 8								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.512	0.051
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	95	1	0.140	0.14
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.053	0
	Dosya Zorluğu	Type 7	1	1	0.2	0.2	0.249	0.050
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	55	0.6	0.046	0.028
	Ağırlıklı Performans Puanı							0.269

Avukathlık Ofisi - 9								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.14	0.14	0.512	0.072
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	45	0.4	0.140	0.056
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.053	0.036
	Dosya Zorluğu	Type 7	1	1	0.69	0.69	0.249	0.174
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	21	0	0.046	0.0000
	Ağırlıklı Performans Puanı							0.338

Avukatlık Ofisi - 10								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.08	0.08	0.512	0.041
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	43	0.4	0.140	0.056
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.053	0
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.249	0.125
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	41	0.4	0.046	0.018
	Ağırlıklı Performans Puanı							0.240

Avukatlık Ofisi - 11									
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer	
	Asıl Alacak Tahsilatı	Type 7	1	1	0.11	0.11	0.512	0.056	
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	50	0.4	0.140	0.056	
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.053	0.017	
	Dosya Zorluğu	Type 7	1	1	0.69	0.69	0.249	0.174	
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	21	0	0.046	0	
	Ağırlıklı Performans Puanı								0.303

Avukatlık Ofisi - 12								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.179	0.179	0.512	0.092
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	20	0	0.140	0
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.053	0.017
	Dosya Zorluğu	Type 7	1	1	0.69	0.69	0.249	0.174
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	32	0.4	0.046	0.018
	Ağırlıklı Performans Puanı							0.301

Avukatlık Ofisi - 13								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.05	0.05	0.512	0.026
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	79	0.8	0.140	0.112
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.053	0.053
	Dosya Zorluğu	Type 7	1	1	0.299	0.299	0.249	0.075
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	38	0.4	0.046	0.018
	Ağırlıklı Performans Puanı							0.284

Avukatlık Ofisi - 14								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.512	0.051
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	66	0.6	0.140	0.084
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.053	0.036
	Dosya Zorluğu	Type 7	1	1	0.2	0.2	0.249	0.050
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	93	1	0.046	0.046
	Ağırlıklı Performans Puanı							0.267

Avukatlık Ofisi - 15								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.059	0.059	0.512	0.031
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	61	0.6	0.140	0.084
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.053	0
	Dosya Zorluğu	Type 7	1	1	0.4	0.4	0.249	0.1
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	93	1	0.046	0.04
	Ağırlıklı Performans Puanı							0.261

EK B. Bulanık Analitik Hiyerarşi Prosesi Kullanılarak Yapılan Avukatlık Ofisi Performans Hesaplamaları

Avukatlık Ofisi - 1								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.341	0.034
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	77	0.8	0.186	0.149
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.127	0.042
	Dosya Zorluğu	Type 7	1	1	0.4	0.4	0.243	0.097
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	59	0.6	0.104	0.062
	Ağırlıklı Performans Puanı							0.384

Avukatlık Ofisi - 2								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.14	0.14	0.341	0.048
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	52	0.6	0.186	0.112
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.127	0.085
	Dosya Zorluğu	Type 7	1	1	0.599	0.599	0.243	0.146
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	89	0.8	0.104	0.083
	Ağırlıklı Performans Puanı							0.474

Avukatlık Ofisi - 3								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.13	0.13	0.341	0.044
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	20	0	0.186	0.000
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.127	0.042
	Dosya Zorluğu	Type 7	1	1	0.599	0.599	0.243	0.146
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	46	0.4	0.104	0.042
	Ağırlıklı Performans Puanı							0.274

Avukatlık Ofisi - 4								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.0599	0.0599	0.341	0.020
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	29	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.127	0.127
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.243	0.122
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	58	0.6	0.104	0.062
	Ağırlıklı Performans Puanı							0.405

Avukatlık Ofisi - 5								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.19	0.19	0.341	0.065
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	25	0	0.186	0.000
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.127	0.042
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.243	0.122
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	90	0.8	0.104	0.083
	Ağırlıklı Performans Puanı							0.312

Avukatlık Ofisi - 6								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1799	0.1799	0.341	0.061
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	50	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.127	0.085
	Dosya Zorluğu	Type 7	1	1	0.299	0.299	0.243	0.073
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	61	0.6	0.104	0.062
	Ağırlıklı Performans Puanı							0.355

Avukatlık Ofisi - 7								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.341	0.034
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	29	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.127	0.127
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.243	0.122
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	65	0.6	0.104	0.062
	Ağırlıklı Performans Puanı							0.419

Avukatlık Ofisi - 8								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.341	0.034
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	95	1	0.186	0.186
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.127	0.000
	Dosya Zorluğu	Type 7	1	1	0.2	0.2	0.243	0.049
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	55	0.6	0.104	0.062
	Ağırlıklı Performans Puanı							0.331

Avukatlık Ofisi - 9								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.14	0.14	0.341	0.048
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	45	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.127	0.085
	Dosya Zorluğu	Type 7	1	1	0.699	0.699	0.243	0.170
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	21	0	0.104	0.000
	Ağırlıklı Performans Puanı							0.377

Avukatlık Ofisi - 10								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.08	0.08	0.341	0.027
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	43	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.127	0.000
	Dosya Zorluğu	Type 7	1	1	0.5	0.5	0.243	0.122
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	41	0.4	0.104	0.042
	Ağırlıklı Performans Puanı							0.265

Avukathık Ofisi - 11								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.11	0.11	0.341	0.038
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	50	0.4	0.186	0.074
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.127	0.042
	Dosya Zorluğu	Type 7	1	1	0.699	0.699	0.243	0.170
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	21	0	0.104	0.000
	Ağırlıklı Performans Puanı							0.324

Avukatlık Ofisi - 12								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.179	0.179	0.341	0.061
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	20	0	0.186	0.000
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	2	0.33	0.127	0.042
	Dosya Zorluğu	Type 7	1	1	0.699	0.699	0.243	0.170
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	32	0.4	0.104	0.042
	Ağırlıklı Performans Puanı							0.315

Avukatlık Ofisi - 13								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.05	0.05	0.341	0.017
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	79	0.8	0.186	0.149
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	0	1	0.127	0.127
	Dosya Zorluğu	Type 7	1	1	0.299	0.299	0.243	0.073
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	38	0.4	0.104	0.042
	Ağırlıklı Performans Puanı							0.408

Avukatlık Ofisi - 14								
	Kriter	Skor Ölçüm Fonksiyonu	X Ekseni Değerleri	Y Ekseni Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.1	0.1	0.341	0.034
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	66	0.6	0.186	0.112
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	1	0.67	0.127	0.085
	Dosya Zorluğu	Type 7	1	1	0.2	0.2	0.243	0.049
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	93	1	0.104	0.104
	Ağırlıklı Performans Puanı							0.384

Avukatlık Ofisi - 15								
	Kriter	Skor Ölçüm Fonksiyonu	X Eksen Değerleri	Y Eksen Değerleri	Gerçekleşen Değer	Hesaplanan Değer	Kriter Ağırlığı	Ağırlıklı Değer
	Asıl Alacak Tahsilatı	Type 7	1	1	0.059	0.059	0.341	0.020
	Denetim Performansı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	61	0.6	0.186	0.112
	Müşteri Memnuniyeti	Type 4	0,1,2,3	1,0.67,0.33,0	3	0	0.127	0.000
	Dosya Zorluğu	Type 7	1	1	0.4	0.4	0.243	0.097
	Genel Değerlendirme Puanı	Type 4	25,50,75,90,100	0,0.4,0.6,0.8,1	93	1	0.104	0.104
	Ağırlıklı Performans Puanı							0.333

EK C. Yazılım Uygulaması Kodları

```
public interface AHPService {  
    public boolean saveCriteria(List<Criteria> criteriaList);  
    public List<BigDecimal> calculateWeights(BigDecimal[][] matrix,int size)  
throws Exception;  
    public List<BigDecimal>  
calculateFuzzyWeights(FuzzyComparisonValues[][] matrix,int size)throws  
Exception;  
    public List<PerformanceNote>  
calculatePerformanceNote(List<LawOfficeCriteriaValue> officeCriteriaList) throws  
Exception;  
}  
  
public class AHPServiceImpl implements AHPService {  
    CriteriaConverter converter = new CriteriaConverter();  
    @Override  
    public boolean saveCriteria(List<Criteria> criteriaList) {  
        CriteriaValidation validation = new CriteriaValidation();  
  
        if (validation.validateCriteria(criteriaList)) {  
            FileWriter writer = new FileWriter();  
            writer.write(converter.convertToEntity(criteriaList));  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    @Override  
    public List<BigDecimal> calculateWeights(BigDecimal[][] matrix, int size)  
throws Exception {  
        MatrixFiller matrixFiller = new MatrixFiller();
```

```

        WeightCalculator calculator = new WeightCalculator();
        ConsistencyCalculator consistencyCalculator = new
ConsistencyCalculator();
        CopyMatrix matrixCopier = new CopyMatrix();

        matrixFiller.fillBlankCell(matrix, size);
        BigDecimal[][] copyMatrix = matrixCopier.copy(matrix, size);
        List<BigDecimal> weightList = calculator.calculate(matrix, size);

        if (consistencyCalculator.checkConsistency(copyMatrix, size,
weightList)) {
            return weightList;
        } else {
            return null;
        }
    }

    @Override
    public List<BigDecimal>
calculateFuzzyWeights(FuzzyComparisonValues[][] matrix, int size) throws
Exception {
        MatrixFiller matrixFiller = new MatrixFiller();
        FuzzyWeightCalculator calculator = new FuzzyWeightCalculator();

        FuzzyValues[][] fuzzyValues = matrixFiller.fillFuzzyMatrix(matrix,
size);

        return calculator.calculate(fuzzyValues, size);
    }

    @Override
    public List<PerformanceNote>
calculatePerformanceNote(List<LawOfficeCriteriaValue> officeCriteriaList)

```

```

        throws Exception {
            FileReader reader = new FileReader();

            Map<Integer, CriteriaEntity> criteriaEntityMap =
reader.read(officeCriteriaList.get(0).getCriteriaValueList().size());
            List<PerformanceNote> responseList = new ArrayList<>();

            if (criteriaEntityMap.isEmpty()) {
                throw new Exception("File reading error");
            }

            Map<Integer, Criteria> criteriaMap =
converter.convertToDtoMap(criteriaEntityMap);

            for (LawOfficeCriteriaValue lawOffice : officeCriteriaList) {
                BigDecimal calculatedNote = BigDecimal.ZERO;

                for (CriteriaValue criteriaValue :
lawOffice.getCriteriaValueList()) {
                    CalculationDto calcDto = new CalculationDto();

                    Criteria criteria =
criteriaMap.get(criteriaValue.getOrder());

                    CalculationType calculator =
calculationTypeFactory(criteria.getFunctionType().getTypeId());
                    calcDto.setValue(criteriaValue.getValue());

                    calcDto.setxAxisValues(criteria.getFunctionType().getxAxisValues());

                    calcDto.setyAxisValues(criteria.getFunctionType().getyAxisValues());
                    BigDecimal calculated = calculator.calculate(calcDto);

```

```

        BigDecimal calculatedValueForCriteria =
criteriaValue.getWeight().multiply(calculated).setScale(3,
BigDecimal.ROUND_HALF_UP);

```

```

        calculatedNote =
calculatedNote.add(calculatedValueForCriteria);
    }
    PerformanceNote performanceNote = new PerformanceNote();

```

```

    performanceNote.setOfficeNumber(lawOffice.getOfficeNumber());

```

```

    performanceNote.setPerformanceNote(calculatedNote.setScale(3,
BigDecimal.ROUND_HALF_UP));
        responseList.add(performanceNote);
    }
    return responseList;
}

```

```

private CalculationType calculationTypeFactory(int order) {
    if (order == 1) {
        return new CalculateType1();
    }

    if (order == 2) {
        return new CalculateType2();
    }

    if (order == 3) {
        return new CalculateType3();
    }

    if (order == 4) {
        return new CalculateType4();
    }
}

```

```

        if (order == 5) {
            return new CalculateType5();
        }

        if (order == 6) {
            return new CalculateType6();
        }

        if (order == 7) {
            return new CalculateType7();
        }

        if (order == 8) {
            return new CalculateType8();
        }

        if (order == 9) {
            return new CalculateType9();
        }

        return null;
    }
}

```

```

public class FileReader {

```

```

    public Map<Integer, CriteriaEntity> read(int size) {
        FileInputStream fi;
        ObjectInputStream oi;
        Map<Integer, CriteriaEntity> crMap = new HashMap<>();
        BigDecimal counter = new BigDecimal(0);

```

```

    try {
        CriteriaEntity cr;
        fi = new FileInputStream(new File("criteriaDb"));
        oi = new ObjectOutputStream(fi);

        while(size > 0) {
            cr = (CriteriaEntity)oi.readObject();
            counter = counter.add(cr.getWeight());
            crMap.put(cr.getOrder(), cr);
            size--;
        }

        oi.close();
        fi.close();

        return crMap;
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    return crMap;
}

}

public class FileWriter {

    public void write(List<CriteriaEntity> criteriaList) {

        try {
            FileOutputStream f = new FileOutputStream(new
File("criteriaDb"));

```

```

        ObjectOutputStream o = new ObjectOutputStream(f);
        BigDecimal weight = BigDecimal.ONE.divide(new
BigDecimal(criteriaList.size()),5,BigDecimal.ROUND_HALF_UP);

        for (CriteriaEntity criteria : criteriaList) {
            criteria.setWeight(weight);
            o.writeObject(criteria);
        }

        o.close();
        f.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

```

public class ConsistencyCalculator {

```

```

    private Map<Integer,BigDecimal> consistencyIndexMap;

```

```

    public ConsistencyCalculator() {
        consistencyIndexMap = new HashMap<>();
        consistencyIndexMap.put(3, new BigDecimal(0.52));
        consistencyIndexMap.put(4, new BigDecimal(0.89));
        consistencyIndexMap.put(5, new BigDecimal(1.11));
        consistencyIndexMap.put(6, new BigDecimal(1.25));
        consistencyIndexMap.put(7, new BigDecimal(1.35));
        consistencyIndexMap.put(8, new BigDecimal(1.4));
        consistencyIndexMap.put(9, new BigDecimal(1.45));
    }
}

```

```

    public boolean checkConsistency(BigDecimal[][] matrix, int size,
List<BigDecimal> weights) {
        BigDecimal lambda = calculate(matrix, size, weights);
        BigDecimal consistencyIndex = consistencyIndex(lambda, size);
        BigDecimal reference = new
BigDecimal(0.1).setScale(3, BigDecimal.ROUND_HALF_UP);

        if(reference.subtract(consistencyIndex).compareTo(BigDecimal.ZERO) > 0)
        {
            return true;
        } else {
            return false;
        }
    }

    private BigDecimal consistencyIndex(BigDecimal lambda, int size) {
        BigDecimal index = lambda.subtract(new
BigDecimal(size)).divide(new
BigDecimal(size-1), 3, BigDecimal.ROUND_HALF_UP);
        BigDecimal randomIndex = consistencyIndexMap.get(size);

        return
index.divide(randomIndex, 3, BigDecimal.ROUND_HALF_UP);
    }

    private BigDecimal calculate(BigDecimal[][] matrix, int size,
List<BigDecimal> weights) {
        List<BigDecimal> averageWeight = calculateAverageWeight(matrix,
size, weights);

        return calculateAverage(averageWeight, size, weights);
    }

```

```

    private List<BigDecimal> calculateAverageWeight(BigDecimal[][] matrix,
    int size, List<BigDecimal> weights) {
        List<BigDecimal> averageWeight = new ArrayList<>();
        for (int i = 0; i < size; i++) {
            BigDecimal sum = BigDecimal.ZERO;
            for (int j = 0; j < size; j++) {
                sum = sum.add(matrix[i][j].multiply(weights.get(j)));
            }
            averageWeight.add(sum);
        }

        return averageWeight;
    }

```

```

    private BigDecimal calculateAverage(List<BigDecimal> averageWeights, int
    size, List<BigDecimal> weights) {
        BigDecimal average = BigDecimal.ZERO;
        for(int i=0;i<size;i++) {
            BigDecimal divide =
            averageWeights.get(i).divide(weights.get(i),3,BigDecimal.ROUND_HALF_UP);
            average = average.add(divide);
        }

        return average.divide(new
        BigDecimal(size),3,BigDecimal.ROUND_HALF_UP);
    }
}

```

```

public class CopyMatrix {

```

```

    public BigDecimal[][] copy(BigDecimal[][] originalMatrix,int size){
        BigDecimal[][] copyMatrix = new BigDecimal[size][size];

        for(int i=0;i<size;i++) {
            for(int j=0;j<size;j++) {

```

```

        copyMatrix[i][j] = originalMatrix[i][j];
    }
}
return copyMatrix;
}
}

```

```

public class FuzzyWeightCalculator {

```

```

    public List<BigDecimal> calculate(FuzzyValues[][] matrix, int size) {
        FuzzyValues[] geoMatrix = generateGeoMatrix(matrix, size);
        FuzzyValues normalizer = calculateNormalizer(geoMatrix);
        FuzzyValues[] normalizedMatrix = normalizeMatrix(geoMatrix, size,
normalizer);

```

```

        return calculateWeights(normalizedMatrix, size);
    }

```

```

    private List<BigDecimal> calculateWeights(FuzzyValues[]
normalizedMatrix, int size) {

```

```

        List<BigDecimal> resp = new ArrayList<>();
        for (int i = 0; i < size; i++) {
            FuzzyValues values = normalizedMatrix[i];
            BigDecimal sum =
values.getFirst().add(values.getSecond()).add(values.getThird());
            resp.add(sum.divide(new BigDecimal(3), 6,
BigDecimal.ROUND_HALF_EVEN));

```

```

        }
        BigDecimal sum = BigDecimal.ZERO;

```

```

        for (BigDecimal val : resp) {
            sum = sum.add(val);
        }

```

```

        for (int i = 0; i < size; i++) {
            resp.set(i, resp.get(i).divide(sum,
BigDecimal.ROUND_HALF_EVEN));
        }
        return resp;
    }

```

```

    private FuzzyValues[] normalizeMatrix(FuzzyValues[] matrix, int
size, FuzzyValues normalizer){
        FuzzyValues[] normalizedMatrix = new FuzzyValues[size];
        for(int i=0; i < size; i++) {
            FuzzyValues value = matrix[i];
            normalizedMatrix[i] = new
FuzzyValues(value.getFirst().multiply(normalizer.getFirst()),value.getSecond().multi
ply(normalizer.getSecond()),value.getThird().multiply(normalizer.getThird()));
        }
        return normalizedMatrix;
    }

```

```

    private FuzzyValues calculateNormalizer(FuzzyValues[] fuzzyList) {

        BigDecimal firstColumnSum = BigDecimal.ZERO;
        BigDecimal secondColumnSum = BigDecimal.ZERO;
        BigDecimal thirdColumnSum = BigDecimal.ZERO;

        for (FuzzyValues val : fuzzyList) {
            firstColumnSum = firstColumnSum.add(val.getFirst());
            secondColumnSum =
secondColumnSum.add(val.getSecond());
            thirdColumnSum = thirdColumnSum.add(val.getThird());
        }

        BigDecimal first = BigDecimal.ONE.divide(thirdColumnSum, 6,
BigDecimal.ROUND_HALF_EVEN);

```

```

        BigDecimal second = BigDecimal.ONE.divide(secondColumnSum, 6,
        BigDecimal.ROUND_HALF_EVEN);

```

```

        BigDecimal third = BigDecimal.ONE.divide(firstColumnSum, 6,
        BigDecimal.ROUND_HALF_EVEN);

```

```

        return new FuzzyValues(first, second, third);
    }

```

```

    private FuzzyValues[] generateGeoMatrix(FuzzyValues[][] matrix, int size) {
        FuzzyValues[] geoAvgMatrix = new FuzzyValues[size];

```

```

        for (int i = 0; i < size; i++) {

```

```

            BigDecimal first = BigDecimal.ONE;

```

```

            BigDecimal second = BigDecimal.ONE;

```

```

            BigDecimal third = BigDecimal.ONE;

```

```

            for (int j = 0; j < size; j++) {

```

```

                first = first.multiply(matrix[i][j].getFirst());

```

```

                second = second.multiply(matrix[i][j].getSecond());

```

```

                third = third.multiply(matrix[i][j].getThird());

```

```

            }

```

```

            double firstVal = Math.pow(first.doubleValue(),

```

```

                BigDecimal.ONE.divide(new

```

```

                BigDecimal(size), 6, BigDecimal.ROUND_HALF_EVEN).doubleValue());

```

```

            double secondVal = Math.pow(second.doubleValue(),

```

```

                BigDecimal.ONE.divide(new

```

```

                BigDecimal(size), 6, BigDecimal.ROUND_HALF_EVEN).doubleValue());

```

```

            double thirdVal = Math.pow(third.doubleValue(),

```

```

                BigDecimal.ONE.divide(new

```

```

                BigDecimal(size), 6, BigDecimal.ROUND_HALF_EVEN).doubleValue());

```

```

            geoAvgMatrix[i] = new FuzzyValues(new

```

```

                BigDecimal(firstVal), new BigDecimal(secondVal),

```

```

        new BigDecimal(thirdVal));
    }
    return geoAvgMatrix;
}
}

public class MatrixFiller {

    public void fillBlankCell(BigDecimal[][] matrix,int size) throws Exception {
        for(int i=0;i<size;i++) {
            for(int j=0; j<size;j++) {
                if(i == j) {
                    matrix[i][j] = BigDecimal.ONE;
                } else {
                    BigDecimal value = matrix[i][j];
                    if(value != null &&
BigDecimal.ZERO.compareTo(value) != 0) {
                        BigDecimal symmetricValue =
matrix[j][i];

                        if(symmetricValue != null) {

                            if(symmetricValue.multiply(value).setScale(0,
BigDecimal.ROUND_HALF_UP).intValue() != 1) {
                                matrix[j][i] =
BigDecimal.ONE.divide(value,3,BigDecimal.ROUND_HALF_UP);
                            }
                        } else {
                            matrix[j][i] =
BigDecimal.ONE.divide(value,3,BigDecimal.ROUND_HALF_UP);
                        }
                    } else {
                        BigDecimal symmetricValue =
matrix[j][i];

```

```

public FuzzyValues<J>[] fillFuzzyMatrix(FuzzyComparisonValues<J>[] matrix, int size)
throws Exception{
    Map<FuzzyComparisonValues, FuzzyValues> converterMap =
mapFiller();
    Map<FuzzyComparisonValues, FuzzyComparisonValues>
mappingMap = mappingMapFiller();

```

182

```

        }
        matrix[i][j] =
mappingMap.get(matrix[j][i]);
    }
}
resp[i][j] = converterMap.get(matrix[i][j]);
}
}

```

```

    return resp;
}

private Map<FuzzyComparisonValues,FuzzyComparisonValues>
mappingMapFiller(){
    Map<FuzzyComparisonValues,FuzzyComparisonValues> map= new
    HashMap<>();
    map.put(FuzzyComparisonValues.ABSOLUTE,
    FuzzyComparisonValues.REVERSE_ABSOLUTE);
    map.put(FuzzyComparisonValues.VERY_STRONG,
    FuzzyComparisonValues.REVERSE_VERY_STRONG);
    map.put(FuzzyComparisonValues.FAIRLY_STRONG,
    FuzzyComparisonValues.REVERSE_FAIRLY_STRONG);
    map.put(FuzzyComparisonValues.WEAK,
    FuzzyComparisonValues.REVERSE_WEAK);
    map.put(FuzzyComparisonValues.EQUAL,
    FuzzyComparisonValues.REVERSE_EQUAL);
    map.put(FuzzyComparisonValues.REVERSE_EQUAL,
    FuzzyComparisonValues.EQUAL);
    map.put(FuzzyComparisonValues.REVERSE_WEAK,
    FuzzyComparisonValues.WEAK);
    map.put(FuzzyComparisonValues.REVERSE_FAIRLY_STRONG,
    FuzzyComparisonValues.FAIRLY_STRONG);

```

```

        map.put(FuzzyComparisonValues.REVERSE_VERY_STRONG,
FuzzyComparisonValues.VERY_STRONG);
        map.put(FuzzyComparisonValues.REVERSE_ABSOLUTE,
FuzzyComparisonValues.ABSOLUTE);

        return map;
    }

    private Map<FuzzyComparisonValues,FuzzyValues> mapFiller(){
        Map<FuzzyComparisonValues,FuzzyValues> map = new
HashMap<FuzzyComparisonValues, FuzzyValues>();
        map.put(FuzzyComparisonValues.ABSOLUTE, new
FuzzyValues(new BigDecimal(3.50), new BigDecimal(4), new BigDecimal(4.50)));
        map.put(FuzzyComparisonValues.VERY_STRONG, new
FuzzyValues(new BigDecimal(2.50), new BigDecimal(3), new BigDecimal(3.50)));
        map.put(FuzzyComparisonValues.FAIRLY_STRONG, new
FuzzyValues(new BigDecimal(1.50), new BigDecimal(2), new BigDecimal(2.50)));
        map.put(FuzzyComparisonValues.WEAK, new FuzzyValues(new
BigDecimal(0.67), BigDecimal.ONE, new BigDecimal(1.50)));
        map.put(FuzzyComparisonValues.EQUAL, new
FuzzyValues(BigDecimal.ONE, BigDecimal.ONE, BigDecimal.ONE));
        map.put(FuzzyComparisonValues.REVERSE_EQUAL, new
FuzzyValues(BigDecimal.ONE, BigDecimal.ONE, BigDecimal.ONE));
        map.put(FuzzyComparisonValues.REVERSE_WEAK, new
FuzzyValues(new BigDecimal(0.67), BigDecimal.ONE, new BigDecimal(1.50)));
        map.put(FuzzyComparisonValues.REVERSE_FAIRLY_STRONG,
new FuzzyValues(new BigDecimal(0.40), new BigDecimal(0.50), new
BigDecimal(0.67)));
        map.put(FuzzyComparisonValues.REVERSE_VERY_STRONG, new
FuzzyValues(new BigDecimal(0.29), new BigDecimal(0.33), new
BigDecimal(0.40)));
        map.put(FuzzyComparisonValues.REVERSE_ABSOLUTE, new
FuzzyValues(new BigDecimal(0.22), new BigDecimal(0.25), new
BigDecimal(0.29)));

```

```

        return map;
    }

}

public class WeightCalculator {

    public List<BigDecimal> calculate(BigDecimal[][] matrix, int size) {
        List<BigDecimal> columnSum = calculateColumnSum(matrix, size);
        normalizeColumn(matrix, size, columnSum);
        return calculateRowAverage(matrix, size);
    }

    public void normalizeColumn(BigDecimal[][] matrix, int size, List<BigDecimal>
weights) {
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                matrix[j][i] = matrix[j][i].divide(weights.get(i), 3,
BigDecimal.ROUND_HALF_UP);
            }
        }
    }

    private List<BigDecimal> calculateColumnSum(BigDecimal[][] matrix, int size) {
        List<BigDecimal> columnSum = new ArrayList<>();
        for (int i = 0; i < size; i++) {
            BigDecimal sum = BigDecimal.ZERO;
            for (int j = 0; j < size; j++) {
                sum = sum.add(matrix[j][i]);
            }
            columnSum.add(sum);
        }
    }
}

```

```

        return columnSum;
    }

    private List<BigDecimal> calculateRowAverage(BigDecimal[][] matrix, int
size) {
        List<BigDecimal> rowSum = new ArrayList<>();
        for (int i = 0; i < size; i++) {
            BigDecimal sum = BigDecimal.ZERO;
            for (int j = 0; j < size; j++) {
                sum = sum.add(matrix[i][j]);
            }
            rowSum.add(sum.divide(new BigDecimal(size), 3,
BigDecimal.ROUND_HALF_UP));
        }
        return rowSum;
    }
}

public enum AHPComparisonValues {
    ABSOLUTE(new BigDecimal(9)), VERY_STRONG(new
BigDecimal(7)), FAIRLY_STRONG(new BigDecimal(5)), WEAK(new
BigDecimal(3)), EQUAL(new BigDecimal(1)), REVERSE_EQUAL(new
BigDecimal(1)),
    REVERSE_WEAK(new
BigDecimal(0.333)), REVERSE_FAIRLY_STRONG(new BigDecimal(0.2)),
    REVERSE_VERY_STRONG(new
BigDecimal(0.143)), REVERSE_ABSOLUTE(new BigDecimal(0.111));

    private AHPComparisonValues(BigDecimal value) {
        this.value = value;
    }

    private BigDecimal value;

```

```

        public BigDecimal getValue() {
            return value;
        }
    }

    public enum FuzzyComparisonValues {

        ABSOLUTE("A"), VERY_STRONG("VS"), FAIRLY_STRONG("FS"), WEAK("W")
        ,EQUAL("E"),
        REVERSE_EQUAL("RE"), REVERSE_WEAK("RW"), REVERSE_FAIRLY_STR
        ONG("RFS"),
        REVERSE_VERY_STRONG("RVS"), REVERSE_ABSOLUTE("RA");

        private FuzzyComparisonValues(String value) {
            this.value = value;
        }

        private String value;

        public String getValue() {
            return value;
        }
    }
}

public class CriteriaEntity implements Serializable{

    private static final long serialVersionUID = -3353208830978421194L;
    private int order;
    private BigDecimal weight;
    private FunctionType functionType;

    public int getOrder() {
        return order;
    }

    public void setOrder(int order) {
        this.order = order;
    }
}

```

```

    }
    public BigDecimal getWeight() {
        return weight;
    }
    public void setWeight(BigDecimal weight) {
        this.weight = weight;
    }
    public FunctionType getFunctionType() {
        return functionType;
    }
    public void setFunctionType(FunctionType functionType) {
        this.functionType = functionType;
    }
}

public class CalculationDto {

    private BigDecimal value;
    private List<BigDecimal> xAxisValues;
    private List<BigDecimal> yAxisValues;

    public BigDecimal getValue() {
        return value;
    }
    public void setValue(BigDecimal value) {
        this.value = value;
    }
    public List<BigDecimal> getXAxisValues() {
        return xAxisValues;
    }
    public void setXAxisValues(List<BigDecimal> xAxisValues) {
        this.xAxisValues = xAxisValues;
    }
}

```

```

    public List<BigDecimal> getyAxisValues() {
        return yAxisValues;
    }

    public void setyAxisValues(List<BigDecimal> yAxisValues) {
        this.yAxisValues = yAxisValues;
    }
}

```

```

public class Criteria implements Serializable{

```

```

    private static final long serialVersionUID = 8535500318953846562L;
    private int order;
    private FunctionType functionType;

```

```

    public Criteria() {
        functionType = new FunctionType();
    }

```

```

    public int getOrder() {
        return order;
    }

```

```

    public void setOrder(int order) {
        this.order = order;
    }

```

```

    public FunctionType getFunctionType() {
        return functionType;
    }

```

```

    public void setFunctionType(FunctionType functionType) {
        this.functionType = functionType;
    }

```

```
}
```

```
public class CriteriaValue {
```

```
    private int order;
```

```
    private BigDecimal value;
```

```
    private BigDecimal weight;
```

```
    public int getOrder() {
```

```
        return order;
```

```
    }
```

```
    public void setOrder(int order) {
```

```
        this.order = order;
```

```
    }
```

```
    public BigDecimal getValue() {
```

```
        return value;
```

```
    }
```

```
    public void setValue(BigDecimal value) {
```

```
        this.value = value;
```

```
    }
```

```
    public BigDecimal getWeight() {
```

```
        return weight;
```

```
    }
```

```
    public void setWeight(BigDecimal weight) {
```

```
        this.weight = weight;
```

```
    }
```

```
}
```

```
public class FunctionType implements Serializable{
```

```
    private static final long serialVersionUID = 4416501801981110397L;
```

```
    private Integer typeId;
```

```

private List<BigDecimal> xAxisValues;
private List<BigDecimal> yAxisValues;

public FunctionType() {
    xAxisValues = new ArrayList<>();
    yAxisValues = new ArrayList<>();
}

public Integer getTypeId() {
    return typeId;
}

public void setTypeId(Integer typeId) {
    this.typeId = typeId;
}

public List<BigDecimal> getXAxisValues() {
    return xAxisValues;
}

public void setXAxisValues(List<BigDecimal> xAxisValues) {
    this.xAxisValues = xAxisValues;
}

public List<BigDecimal> getYAxisValues() {
    return yAxisValues;
}

public void setYAxisValues(List<BigDecimal> yAxisValues) {
    this.yAxisValues = yAxisValues;
}
}

public class FuzzyValues {

    private BigDecimal first;
    private BigDecimal second;
    private BigDecimal third;

```

```

public FuzzyValues(BigDecimal first, BigDecimal second, BigDecimal third) {
    this.first=first;
    this.second=second;
    this.third = third;
}

public BigDecimal getFirst() {
    return first;
}

public void setFirst(BigDecimal first) {
    this.first = first;
}

public BigDecimal getSecond() {
    return second;
}

public void setSecond(BigDecimal second) {
    this.second = second;
}

public BigDecimal getThird() {
    return third;
}

public void setThird(BigDecimal third) {
    this.third = third;
}
}

```

```

public class LawOfficeCriteriaValue {

    private Long officeNumber;

    private List<CriteriaValue> criteriaValueList;

```

```

    public Long getOfficeNumber() {
        return officeNumber;
    }

    public void setOfficeNumber(Long officeNumber) {
        this.officeNumber = officeNumber;
    }

    public List<CriteriaValue> getCriteriaValueList() {
        return criteriaValueList;
    }

    public void setCriteriaValueList(List<CriteriaValue> criteriaValueList) {
        this.criteriaValueList = criteriaValueList;
    }
}

public class PerformanceNote {

    private Long officeNumber;
    private BigDecimal performanceNote;

    public Long getOfficeNumber() {
        return officeNumber;
    }

    public void setOfficeNumber(Long officeNumber) {
        this.officeNumber = officeNumber;
    }

    public BigDecimal getPerformanceNote() {
        return performanceNote;
    }

    public void setPerformanceNote(BigDecimal performanceNote) {
        this.performanceNote = performanceNote;
    }
}

```

```

public class CriteriaConverter {

    public List<CriteriaEntity> convertToEntity(List<Criteria> criteriaList) {

        List<CriteriaEntity> entityList = new ArrayList<>();

        for(Criteria criteria : criteriaList) {

            CriteriaEntity entity = new CriteriaEntity();
            entity.setFunctionType(criteria.getFunctionType());
            entity.setOrder(criteria.getOrder());

            entityList.add(entity);
        }

        return entityList;
    }

    public Map<Integer, Criteria> convertToDtoMap(Map<Integer,
CriteriaEntity> entityMap) {

        Map<Integer, Criteria> criteriaMap = new HashMap<>();

        for (Map.Entry<Integer, CriteriaEntity> entry : entityMap.entrySet())
        {

            criteriaMap.put(entry.getKey(),
convertToDto(entry.getValue()));
        }

        return criteriaMap;
    }
}

```

```

    private Criteria convertToDto(CriteriaEntity entity) {
        Criteria criteria = new Criteria();
        criteria.setFunctionType(entity.getFunctionType());
        criteria.setOrder(entity.getOrder());

        return criteria;
    }
}

public class CommonOperations {

    public static BigDecimal calculateSlope(BigDecimal x, BigDecimal y) {
        return y.divide(x, 3, BigDecimal.ROUND_HALF_EVEN);
    }
}

public interface CalculationType {

    public BigDecimal calculate(CalculationDto parameters);
}

public class CalculateType1 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {
        if(BigDecimal.ZERO.compareTo(parameters.getValue()) == 0) {
            return BigDecimal.ZERO;
        }
        return parameters.getyAxisValues().get(0);
    }
}

```

```

public class CalculateType2 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {

        BigDecimal xAxisLimit = parameters.getxAxisValues().get(0);

        if (parameters.getValue().compareTo(xAxisLimit) < 0) {
            return BigDecimal.ZERO;
        } else {
            return parameters.getyAxisValues().get(0);
        }
    }
}

```

```

public class CalculateType3 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {

        BigDecimal xAxisValue = parameters.getxAxisValues().get(0);

        if(parameters.getValue().compareTo(xAxisValue) >= 0) {
            return parameters.getyAxisValues().get(0);
        } else {
            BigDecimal slope =
CommonOperations.calculateSlope(xAxisValue,
parameters.getyAxisValues().get(0));
            return slope.multiply(parameters.getValue());
        }
    }
}

```

```

public class CalculateType4 implements CalculationType {

```

```

public BigDecimal calculate(CalculationDto parameters) {

    int size = parameters.getxAxisValues().size();
    List<BigDecimal> xAxisValues = parameters.getxAxisValues();
    List<BigDecimal> yAxisValues = parameters.getyAxisValues();

    return recursiveCalc(xAxisValues,          yAxisValues,
parameters.getValue(), size, 0);
}

private          BigDecimal          recursiveCalc(List<BigDecimal>
xAxisValues,List<BigDecimal> yAxisValues,BigDecimal value,int size,int counter)
{

    if(counter >= size ) {
        return yAxisValues.get(size-1);
    }

    if(value.compareTo(xAxisValues.get(counter)) <= 0) {
        return yAxisValues.get(counter);
    } else {
        counter++;
        return recursiveCalc(xAxisValues, yAxisValues, value, size,
counter);
    }
}
}

```

```

public class CalculateType5 implements CalculationType {

```

```

    public BigDecimal calculate(CalculationDto parameters) {

```

```

List<BigDecimal> xAxisValues = parameters.getxAxisValues();

if(parameters.getValue().compareTo(xAxisValues.get(0)) <= 0) {
    return new BigDecimal(0);
}

if(parameters.getValue().compareTo(xAxisValues.get(1)) < 0) {
    BigDecimal y =
parameters.getValue().subtract(xAxisValues.get(0));
    BigDecimal x =
xAxisValues.get(1).subtract(xAxisValues.get(0));
    return CommonOperations.calculateSlope(x,y);
} else {
    return parameters.getyAxisValues().get(0);
}
}
}

```

```

public BigDecimal calculate(CalculationDto parameters) {

```

```

    List<BigDecimal> xAxisValues = parameters.getxAxisValues();
    List<BigDecimal> yAxisValues = parameters.getyAxisValues();

    if(parameters.getValue().compareTo(xAxisValues.get(1)) >= 0) {
        return yAxisValues.get(0);
    } else {
        BigDecimal average =
(xAxisValues.get(0).add(xAxisValues.get(1))).divide(new BigDecimal(2));

        if(parameters.getValue().compareTo(average) > 0) {
            BigDecimal temporary =
(parameters.getValue().subtract(xAxisValues.get(1))).divide((xAxisValues.get(1).su
btract(xAxisValues.get(0))),3,BigDecimal.ROUND_HALF_UP);

```

```

        return BigDecimal.ONE.subtract((new
BigDecimal(2).multiply(temporary.multiply(temporary))));
    } else {
        BigDecimal temporary =
(parameters.getValue().subtract(xAxisValues.get(0))).divide((xAxisValues.get(1).su
btract(xAxisValues.get(0))),3,BigDecimal.ROUND_HALF_UP);
        return new
BigDecimal(2).multiply(temporary.multiply(temporary));
    }
}
}
}

```

```

public class CalculateType7 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {

        List<BigDecimal> xAxisValues = parameters.getxAxisValues();

        return CommonOperations.calculateSlope(xAxisValues.get(0),
parameters.getyAxisValues().get(0)).multiply(parameters.getValue());
    }
}

```

```

public class CalculateType8 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {

        List<BigDecimal> xAxisValues = parameters.getxAxisValues();

        if(parameters.getValue().compareTo(xAxisValues.get(1)) > 0) {

            return
CommonOperations.calculateSlope(xAxisValues.get(0), xAxisValues.get(1));
        }
    }
}

```

```

    }

    if(parameters.getValue().compareTo(BigDecimal.ZERO) < 0) {
        return BigDecimal.ZERO;
    }

    return CommonOperations.calculateSlope(
xAxisValues.get(0),parameters.getValue());
    }
}

```

```

public class CalculateType9 implements CalculationType {

    public BigDecimal calculate(CalculationDto parameters) {

        List<BigDecimal> xAxisValues = parameters.getXAxisValues();

        if(parameters.getValue().compareTo(xAxisValues.get(2)) > 0) {
            BigDecimal y =
xAxisValues.get(2).subtract(xAxisValues.get(0));
            BigDecimal x =
xAxisValues.get(1).subtract(xAxisValues.get(0));

            return CommonOperations.calculateSlope(x, y);
        }

        if(parameters.getValue().compareTo(xAxisValues.get(0)) <= 0) {
            return BigDecimal.ZERO;
        }

        BigDecimal x = xAxisValues.get(1).subtract(xAxisValues.get(0));
        BigDecimal y = parameters.getValue().subtract(xAxisValues.get(0));
    }
}

```

```

        return CommonOperations.calculateSlope(x, y);
    }
}

```

Ek D. AHP İle Performans Hesaplama Kodları

```

public class TestAHP {

    public static void main(String[] args) {
        AHPService service = new AHPServiceImpl();
        List<Criteria> criteriaList = new ArrayList<>();

        criteriaList.add(generateCriteria1());
        criteriaList.add(generateCriteria2());
        criteriaList.add(generateCriteria3());
        criteriaList.add(generateCriteria4());
        criteriaList.add(generateCriteria5());

        service.saveCriterias(criteriaList);
        List<BigDecimal> weights = new ArrayList<>();
        try {
            weights = service.calculateWeights(generateMatrix(), 5);
        } catch (Exception e1) {
            e1.printStackTrace();
        }
    }
}

```

```

        List<LawOfficeCriteriaValue> officeCriteriaList =
generateOfficeList(weights);

```

```

        try {
            List<PerformanceNote> calculatePerformanceNote =
service.calculatePerformanceNote(officeCriteriaList);
            for(int i=0;i<calculatePerformanceNote.size();i++) {

            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    private static List<LawOfficeCriteriaValue>
generateOfficeList(List<BigDecimal> weights) {
        List<LawOfficeCriteriaValue> officeCriteriaList = new
ArrayList<>();

```

```

        LawOfficeCriteriaValue officeCriteria1 = new
LawOfficeCriteriaValue();
        officeCriteria1.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
                new BigDecimal(77), new BigDecimal(2), new
BigDecimal(0.4), new BigDecimal(59))));
        officeCriteria1.setOfficeNumber(1L);
        officeCriteriaList.add(officeCriteria1);

```

```

        LawOfficeCriteriaValue officeCriteria2 = new
LawOfficeCriteriaValue();
        officeCriteria2.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.14),

```

```

        new BigDecimal(52), new BigDecimal(1), new
        BigDecimal(0.6), new BigDecimal(89)))));

```

```

        officeCriteria2.setOfficeNumber(2L);
        officeCriteriaList.add(officeCriteria2);

```

```

        LawOfficeCriteriaValue officeCriteria3 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria3.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.13),
        new BigDecimal(20), new BigDecimal(2), new
        BigDecimal(0.6), new BigDecimal(46))));

```

```

        officeCriteria3.setOfficeNumber(3L);
        officeCriteriaList.add(officeCriteria3);

```

```

        LawOfficeCriteriaValue officeCriteria4 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria4.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.06),
        new BigDecimal(29), new BigDecimal(0), new
        BigDecimal(0.5), new BigDecimal(58))));

```

```

        officeCriteria4.setOfficeNumber(4L);
        officeCriteriaList.add(officeCriteria4);

```

```

        LawOfficeCriteriaValue officeCriteria5 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria5.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.19),
        new BigDecimal(25), new BigDecimal(2), new
        BigDecimal(0.5), new BigDecimal(90))));

```

```

        officeCriteria5.setOfficeNumber(5L);
        officeCriteriaList.add(officeCriteria5);

```

```

        LawOfficeCriteriaValue officeCriteria6 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria6.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.18),
                new BigDecimal(50), new BigDecimal(1), new
BigDecimal(0.3), new BigDecimal(61))));
        officeCriteria6.setOfficeNumber(6L);
        officeCriteriaList.add(officeCriteria6);

```

```

        LawOfficeCriteriaValue officeCriteria7 = new
LawOfficeCriteriaValue();
        officeCriteria7.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
                new BigDecimal(29), new BigDecimal(0), new
BigDecimal(0.5), new BigDecimal(65))));
        officeCriteria7.setOfficeNumber(7L);
        officeCriteriaList.add(officeCriteria7);

```

```

        LawOfficeCriteriaValue officeCriteria8 = new
LawOfficeCriteriaValue();
        officeCriteria8.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
                new BigDecimal(95), new BigDecimal(3), new
BigDecimal(0.2), new BigDecimal(55))));
        officeCriteria8.setOfficeNumber(8L);
        officeCriteriaList.add(officeCriteria8);

```

```

        LawOfficeCriteriaValue officeCriteria9 = new
LawOfficeCriteriaValue();
        officeCriteria9.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.14),
                new BigDecimal(45), new BigDecimal(1), new
BigDecimal(0.7), new BigDecimal(21))));
        officeCriteria9.setOfficeNumber(9L);
        officeCriteriaList.add(officeCriteria9);

```

```

        LawOfficeCriteriaValue      officeCriteria10      =      new
LawOfficeCriteriaValue();
        officeCriteria10.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.08),
                new BigDecimal(43), new BigDecimal(3), new
BigDecimal(0.5), new BigDecimal(41))));
        officeCriteria10.setOfficeNumber(10L);
        officeCriteriaList.add(officeCriteria10);

```

```

        LawOfficeCriteriaValue      officeCriteria11      =      new
LawOfficeCriteriaValue();
        officeCriteria11.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.11),
                new BigDecimal(50), new BigDecimal(2), new
BigDecimal(0.7), new BigDecimal(21))));
        officeCriteria11.setOfficeNumber(11L);
        officeCriteriaList.add(officeCriteria11);

```

```

        LawOfficeCriteriaValue      officeCriteria12      =      new
LawOfficeCriteriaValue();
        officeCriteria12.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.18),
                new BigDecimal(20), new BigDecimal(2), new
BigDecimal(0.7), new BigDecimal(32))));
        officeCriteria12.setOfficeNumber(12L);
        officeCriteriaList.add(officeCriteria12);

```

```

        LawOfficeCriteriaValue      officeCriteria13      =      new
LawOfficeCriteriaValue();
        officeCriteria13.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.05),
                new BigDecimal(79), new BigDecimal(0), new
BigDecimal(0.3), new BigDecimal(38))));
        officeCriteria13.setOfficeNumber(13L);

```

```

        officeCriteriaList.add(officeCriteria13);

        LawOfficeCriteriaValue officeCriteria14 = new
LawOfficeCriteriaValue();
        officeCriteria14.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
                new BigDecimal(66), new BigDecimal(1), new
BigDecimal(0.2), new BigDecimal(93))));
        officeCriteria14.setOfficeNumber(14L);
        officeCriteriaList.add(officeCriteria14);

        LawOfficeCriteriaValue officeCriteria15 = new
LawOfficeCriteriaValue();
        officeCriteria15.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.06),
                new BigDecimal(61), new BigDecimal(3), new
BigDecimal(0.4), new BigDecimal(93))));
        officeCriteria15.setOfficeNumber(15L);
        officeCriteriaList.add(officeCriteria15);

        return officeCriteriaList;
    }

    private static List<CriteriaValue> generateCriteriaList(List<BigDecimal>
weights, List<BigDecimal> values) {
        List<CriteriaValue> kpiValList = new ArrayList<>();

        CriteriaValue criteriaVal1 = new CriteriaValue();
        criteriaVal1.setOrder(1);
        criteriaVal1.setValue(values.get(0));
        criteriaVal1.setWeight(weights.get(0).setScale(3,
BigDecimal.ROUND_HALF_UP));
        kpiValList.add(criteriaVal1);

```

```
CriteriaValue criteriaVal2 = new CriteriaValue();
criteriaVal2.setOrder(2);
criteriaVal2.setValue(values.get(1));
criteriaVal2.setWeight(weights.get(1).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal2);
```

```
CriteriaValue criteriaVal3 = new CriteriaValue();
criteriaVal3.setOrder(3);
criteriaVal3.setValue(values.get(2));
criteriaVal3.setWeight(weights.get(2).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal3);
```

```
CriteriaValue criteriaVal4 = new CriteriaValue();
criteriaVal4.setOrder(4);
criteriaVal4.setValue(values.get(3));
criteriaVal4.setWeight(weights.get(3).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal4);
```

```
CriteriaValue criteriaVal5 = new CriteriaValue();
criteriaVal5.setOrder(5);
criteriaVal5.setValue(values.get(4));
criteriaVal5.setWeight(weights.get(4).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal5);
```

```
return kpiValList;
```

```
}
```

```
private static Criteria generateCriteria1() {
    Criteria criteria = new Criteria();
    criteria.setOrder(1);
```

```

        FunctionType funcType = new FunctionType();
        funcType.setTypeId(7);
        funcType.setxAxisValues(Arrays.asList(new BigDecimal(1)));
        funcType.setyAxisValues(Arrays.asList(new BigDecimal(1)));
        criteria.setFunctionType(funcType);
        return criteria;
    }

```

```

    private static Criteria generateCriteria2() {
        Criteria criteria = new Criteria();
        criteria.setOrder(2);

        FunctionType funcType = new FunctionType();
        funcType.setTypeId(4);
        funcType.setxAxisValues(Arrays.asList(new BigDecimal(25), new
        BigDecimal(50), new BigDecimal(75),
            new BigDecimal(90), new BigDecimal(100)));
        funcType.setyAxisValues(
            Arrays.asList(BigDecimal.ZERO, new
            BigDecimal(0.4).setScale(2, BigDecimal.ROUND_HALF_UP),
                new BigDecimal(0.6).setScale(2,
        BigDecimal.ROUND_HALF_UP),
                new BigDecimal(0.8).setScale(2,
        BigDecimal.ROUND_HALF_UP), BigDecimal.ONE));
        criteria.setFunctionType(funcType);
        return criteria;
    }

```

```

    private static Criteria generateCriteria3() {
        Criteria criteria = new Criteria();
        criteria.setOrder(3);

        FunctionType funcType = new FunctionType();

```

```

funcType.setTypeId(4);
funcType.setxAxisValues(Arrays.asList(BigDecimal.ZERO,
BigDecimal.ONE, new BigDecimal(2), new BigDecimal(3)));
funcType.setyAxisValues(
    Arrays.asList(BigDecimal.ONE, new
    BigDecimal(0.67).setScale(2, BigDecimal.ROUND_HALF_UP),
    new BigDecimal(0.33).setScale(2,
    BigDecimal.ROUND_HALF_UP), BigDecimal.ZERO));
criteria.setFunctionType(funcType);
return criteria;
}

```

```

private static Criteria generateCriteria4() {
    Criteria criteria = new Criteria();
    criteria.setOrder(4);

    FunctionType funcType = new FunctionType();
    funcType.setTypeId(7);
    funcType.setxAxisValues(Arrays.asList(new BigDecimal(1)));
    funcType.setyAxisValues(Arrays.asList(new BigDecimal(1)));
    criteria.setFunctionType(funcType);
    return criteria;
}

```

```

private static Criteria generateCriteria5() {
    Criteria criteria = new Criteria();
    criteria.setOrder(5);

    FunctionType funcType = new FunctionType();
    funcType.setTypeId(4);
    funcType.setxAxisValues(Arrays.asList(new BigDecimal(25), new
    BigDecimal(50), new BigDecimal(75),
    new BigDecimal(90), new BigDecimal(100)));
    funcType.setyAxisValues(

```

```

        Arrays.asList(BigDecimal.ZERO, new
        BigDecimal(0.4).setScale(2, BigDecimal.ROUND_HALF_UP),
        new BigDecimal(0.6).setScale(2,
        BigDecimal.ROUND_HALF_UP),
        new BigDecimal(0.8).setScale(2,
        BigDecimal.ROUND_HALF_UP), BigDecimal.ONE));
        criteria.setFunctionType(funcType);
        return criteria;
    }

```

```

    private static BigDecimal[][] generateMatrix() {
        BigDecimal[][] matrix = new BigDecimal[5][5];

        matrix[0][0] = AHPComparisonValues.EQUAL.getValue();
        matrix[0][1] = AHPComparisonValues.FAIRLY_STRONG.getValue();
        matrix[0][2] = AHPComparisonValues.VERY_STRONG.getValue();
        matrix[0][3] = AHPComparisonValues.WEAK.getValue();
        matrix[0][4] = AHPComparisonValues.ABSOLUTE.getValue();

        matrix[1][1] = AHPComparisonValues.EQUAL.getValue();
        matrix[1][2] = AHPComparisonValues.WEAK.getValue();
        matrix[1][4] = AHPComparisonValues.FAIRLY_STRONG.getValue();

        matrix[2][2] = AHPComparisonValues.EQUAL.getValue();
        matrix[2][4] = AHPComparisonValues.EQUAL.getValue();

        matrix[3][1] = AHPComparisonValues.WEAK.getValue();
        matrix[3][2] = AHPComparisonValues.FAIRLY_STRONG.getValue();
        matrix[3][3] = AHPComparisonValues.EQUAL.getValue();
        matrix[3][4] = AHPComparisonValues.FAIRLY_STRONG.getValue();
    }

```

```

matrix[4][2] = AHPComparisonValues.EQUAL.getValue();
matrix[4][4] = AHPComparisonValues.EQUAL.getValue();

return matrix;
}
}

```



Ek D. BAHP İle Performans Hesaplama Kodları

```

public class TestFuzzyAHP {

    public static void main(String[] args) {
        AHPService service = new AHPServiceImpl();
        List<Criteria> criteriaList = new ArrayList<>();

        criteriaList.add(generateCriteria1());
        criteriaList.add(generateCriteria2());
        criteriaList.add(generateCriteria3());
    }
}

```

```
criteriaList.add(generateCriteria4());
criteriaList.add(generateCriteria5());
```

```
service.saveCriterias(criteriaList);
List<BigDecimal> weights = new ArrayList<>();
try {
    weights = service.calculateFuzzyWeights(generateMatrix(), 5);
} catch (Exception e1) {
    e1.printStackTrace();
}
```

```
List<LawOfficeCriteriaValue> officeCriteriaList =
generateOfficeList(weights);

try {
    List<PerformanceNote> calculatePerformanceNote =
service.calculatePerformanceNote(officeCriteriaList);
} catch (Exception e) {
    e.printStackTrace();
}
}
```

```
private static List<LawOfficeCriteriaValue>
generateOfficeList(List<BigDecimal> weights) {
    List<LawOfficeCriteriaValue> officeCriteriaList = new
ArrayList<>();
```

```
        LawOfficeCriteriaValue officeCriteria1 = new
LawOfficeCriteriaValue();
        officeCriteria1.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
new BigDecimal(77), new BigDecimal(2), new
BigDecimal(0.4), new BigDecimal(59))));
```

```
officeCriteria1.setOfficeNumber(1L);
officeCriteriaList.add(officeCriteria1);
```

```
LawOfficeCriteriaValue officeCriteria2 = new
LawOfficeCriteriaValue();
officeCriteria2.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.14),
new BigDecimal(52), new BigDecimal(1), new
BigDecimal(0.6), new BigDecimal(89))));
officeCriteria2.setOfficeNumber(2L);
officeCriteriaList.add(officeCriteria2);
```

```
LawOfficeCriteriaValue officeCriteria3 = new
LawOfficeCriteriaValue();
officeCriteria3.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.13),
new BigDecimal(20), new BigDecimal(2), new
BigDecimal(0.6), new BigDecimal(46))));
officeCriteria3.setOfficeNumber(3L);
officeCriteriaList.add(officeCriteria3);
```

```
LawOfficeCriteriaValue officeCriteria4 = new
LawOfficeCriteriaValue();
officeCriteria4.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.06),
new BigDecimal(29), new BigDecimal(0), new
BigDecimal(0.5), new BigDecimal(58))));
officeCriteria4.setOfficeNumber(4L);
officeCriteriaList.add(officeCriteria4);
```

```
LawOfficeCriteriaValue officeCriteria5 = new
LawOfficeCriteriaValue();
officeCriteria5.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.19),
```

```

        new BigDecimal(25), new BigDecimal(2), new
        BigDecimal(0.5), new BigDecimal(90)))));

```

```

        officeCriteria5.setOfficeNumber(5L);
        officeCriteriaList.add(officeCriteria5);

```

```

        LawOfficeCriteriaValue officeCriteria6 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria6.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.18),
        new BigDecimal(50), new BigDecimal(1), new
        BigDecimal(0.3), new BigDecimal(61))));

```

```

        officeCriteria6.setOfficeNumber(6L);
        officeCriteriaList.add(officeCriteria6);

```

```

        LawOfficeCriteriaValue officeCriteria7 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria7.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.1),
        new BigDecimal(29), new BigDecimal(0), new
        BigDecimal(0.5), new BigDecimal(65))));

```

```

        officeCriteria7.setOfficeNumber(7L);
        officeCriteriaList.add(officeCriteria7);

```

```

        LawOfficeCriteriaValue officeCriteria8 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria8.setCriteriaValueList(generateCriteriaList(weights,
        Arrays.asList(new BigDecimal(0.1),
        new BigDecimal(95), new BigDecimal(3), new
        BigDecimal(0.2), new BigDecimal(55))));

```

```

        officeCriteria8.setOfficeNumber(8L);
        officeCriteriaList.add(officeCriteria8);

```

```

        LawOfficeCriteriaValue officeCriteria9 = new
        LawOfficeCriteriaValue();

```

```

        officeCriteria9.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.14),
                new BigDecimal(45), new BigDecimal(1), new
BigDecimal(0.7), new BigDecimal(21))));
        officeCriteria9.setOfficeNumber(9L);
        officeCriteriaList.add(officeCriteria9);

```

```

        LawOfficeCriteriaValue officeCriteria10 = new
LawOfficeCriteriaValue();
        officeCriteria10.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.08),
                new BigDecimal(43), new BigDecimal(3), new
BigDecimal(0.5), new BigDecimal(41))));
        officeCriteria10.setOfficeNumber(10L);
        officeCriteriaList.add(officeCriteria10);

```

```

        LawOfficeCriteriaValue officeCriteria11 = new
LawOfficeCriteriaValue();
        officeCriteria11.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.11),
                new BigDecimal(50), new BigDecimal(2), new
BigDecimal(0.7), new BigDecimal(21))));
        officeCriteria11.setOfficeNumber(11L);
        officeCriteriaList.add(officeCriteria11);

```

```

        LawOfficeCriteriaValue officeCriteria12 = new
LawOfficeCriteriaValue();
        officeCriteria12.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.18),
                new BigDecimal(20), new BigDecimal(2), new
BigDecimal(0.7), new BigDecimal(32))));
        officeCriteria12.setOfficeNumber(12L);
        officeCriteriaList.add(officeCriteria12);

```

```

        LawOfficeCriteriaValue officeCriteria13 = new
LawOfficeCriteriaValue();
        officeCriteria13.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.05),
                new BigDecimal(79), new BigDecimal(0), new
BigDecimal(0.3), new BigDecimal(38))));
        officeCriteria13.setOfficeNumber(13L);
        officeCriteriaList.add(officeCriteria13);

```

```

        LawOfficeCriteriaValue officeCriteria14 = new
LawOfficeCriteriaValue();
        officeCriteria14.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.1),
                new BigDecimal(66), new BigDecimal(1), new
BigDecimal(0.2), new BigDecimal(93))));
        officeCriteria14.setOfficeNumber(14L);
        officeCriteriaList.add(officeCriteria14);

```

```

        LawOfficeCriteriaValue officeCriteria15 = new
LawOfficeCriteriaValue();
        officeCriteria15.setCriteriaValueList(generateCriteriaList(weights,
Arrays.asList(new BigDecimal(0.06),
                new BigDecimal(61), new BigDecimal(3), new
BigDecimal(0.4), new BigDecimal(93))));
        officeCriteria15.setOfficeNumber(15L);
        officeCriteriaList.add(officeCriteria15);

```

```

        return officeCriteriaList;
    }

```

```

    private static List<CriteriaValue> generateCriteriaList(List<BigDecimal>
weights, List<BigDecimal> values) {
        List<CriteriaValue> kpiValList = new ArrayList<>();

```

```
CriteriaValue criteriaVal1 = new CriteriaValue();
criteriaVal1.setOrder(1);
criteriaVal1.setValue(values.get(0));
criteriaVal1.setWeight(weights.get(0).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal1);
```

```
CriteriaValue criteriaVal2 = new CriteriaValue();
criteriaVal2.setOrder(2);
criteriaVal2.setValue(values.get(1));
criteriaVal2.setWeight(weights.get(1).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal2);
```

```
CriteriaValue criteriaVal3 = new CriteriaValue();
criteriaVal3.setOrder(3);
criteriaVal3.setValue(values.get(2));
criteriaVal3.setWeight(weights.get(2).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal3);
```

```
CriteriaValue criteriaVal4 = new CriteriaValue();
criteriaVal4.setOrder(4);
criteriaVal4.setValue(values.get(3));
criteriaVal4.setWeight(weights.get(3).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal4);
```

```
CriteriaValue criteriaVal5 = new CriteriaValue();
criteriaVal5.setOrder(5);
criteriaVal5.setValue(values.get(4));
criteriaVal5.setWeight(weights.get(4).setScale(3,
BigDecimal.ROUND_HALF_UP));
kpiValList.add(criteriaVal5);
```

```

        return kpiValList;
    }

```

```

private static Criteria generateCriteria1() {
    Criteria criteria = new Criteria();
    criteria.setOrder(1);

    FunctionType funcType = new FunctionType();
    funcType.setTypeId(7);
    funcType.setxAxisValues(Arrays.asList(new BigDecimal(1)));
    funcType.setyAxisValues(Arrays.asList(new BigDecimal(1)));
    criteria.setFunctionType(funcType);
    return criteria;
}

```

```

private static Criteria generateCriteria2() {
    Criteria criteria = new Criteria();
    criteria.setOrder(2);

    FunctionType funcType = new FunctionType();
    funcType.setTypeId(4);
    funcType.setxAxisValues(Arrays.asList(new BigDecimal(25), new
BigDecimal(50), new BigDecimal(75),
        new BigDecimal(90), new BigDecimal(100)));
    funcType.setyAxisValues(
        Arrays.asList(BigDecimal.ZERO, new
BigDecimal(0.4).setScale(2, BigDecimal.ROUND_HALF_UP),
        new BigDecimal(0.6).setScale(2,
BigDecimal.ROUND_HALF_UP),
        new BigDecimal(0.8).setScale(2,
BigDecimal.ROUND_HALF_UP), BigDecimal.ONE));
    criteria.setFunctionType(funcType);
    return criteria;
}

```

```
}
```

```
private static Criteria generateCriteria3() {  
    Criteria criteria = new Criteria();  
    criteria.setOrder(3);  
  
    FunctionType funcType = new FunctionType();  
    funcType.setTypeId(4);  
    funcType.setxAxisValues(Arrays.asList(BigDecimal.ZERO,  
BigDecimal.ONE, new BigDecimal(2), new BigDecimal(3)));  
    funcType.setyAxisValues(  
        Arrays.asList(BigDecimal.ONE, new  
BigDecimal(0.67).setScale(2, BigDecimal.ROUND_HALF_UP),  
        new BigDecimal(0.33).setScale(2,  
BigDecimal.ROUND_HALF_UP), BigDecimal.ZERO));  
    criteria.setFunctionType(funcType);  
    return criteria;  
}
```

```
private static Criteria generateCriteria4() {  
    Criteria criteria = new Criteria();  
    criteria.setOrder(4);  
  
    FunctionType funcType = new FunctionType();  
    funcType.setTypeId(7);  
    funcType.setxAxisValues(Arrays.asList(new BigDecimal(1)));  
    funcType.setyAxisValues(Arrays.asList(new BigDecimal(1)));  
    criteria.setFunctionType(funcType);  
    return criteria;  
}
```

```
private static Criteria generateCriteria5() {  
    Criteria criteria = new Criteria();  
    criteria.setOrder(5);
```

```

        FunctionType funcType = new FunctionType();
        funcType.setTypeId(4);
        funcType.setxAxisValues(Arrays.asList(new BigDecimal(25), new
BigDecimal(50), new BigDecimal(75),
                new BigDecimal(90), new BigDecimal(100)));
        funcType.setyAxisValues(
                Arrays.asList(BigDecimal.ZERO, new
BigDecimal(0.4).setScale(2, BigDecimal.ROUND_HALF_UP),
new BigDecimal(0.6).setScale(2, BigDecimal.ROUND_HALF_UP),
new BigDecimal(0.8).setScale(2, BigDecimal.ROUND_HALF_UP),
BigDecimal.ONE));
        criteria.setFunctionType(funcType);
        return criteria;
    }

```

```

    private static FuzzyComparisonValues[][] generateMatrix() {
        FuzzyComparisonValues[][] matrix = new
FuzzyComparisonValues[5][5];

```

```

        matrix[0][0] = FuzzyComparisonValues.EQUAL;
        matrix[0][1] = FuzzyComparisonValues.FAIRLY_STRONG;
        matrix[0][2] = FuzzyComparisonValues.VERY_STRONG;
        matrix[0][3] = FuzzyComparisonValues.WEAK;
        matrix[0][4] = FuzzyComparisonValues.ABSOLUTE;

```

```

        matrix[1][1] = FuzzyComparisonValues.EQUAL;
        matrix[1][2] = FuzzyComparisonValues.WEAK;
        matrix[1][4] = FuzzyComparisonValues.FAIRLY_STRONG;

```

```

        matrix[2][2] = FuzzyComparisonValues.EQUAL;
        matrix[2][4] = FuzzyComparisonValues.EQUAL;

```

```

        matrix[3][1] = FuzzyComparisonValues.WEAK;

```

```
matrix[3][2] = FuzzyComparisonValues.FAIRLY_STRONG;  
matrix[3][3] = FuzzyComparisonValues.EQUAL;  
matrix[3][4] = FuzzyComparisonValues.FAIRLY_STRONG;  
  
matrix[4][2] = FuzzyComparisonValues.EQUAL;  
matrix[4][4] = FuzzyComparisonValues.EQUAL;  
  
return matrix;  
}  
}
```





ÖZGEÇMİŞ

Ad-Soyad : Erdinç URUÇ
Doğum Tarihi ve Yeri : 10.03.1990 İstanbul
E-posta : erdinc.uruc@gmail.com

ÖĞRENİM DURUMU:

- **Lise** : 2009, Kadıköy Anadolu Lisesi
- **Lisans** : 2014, Gebze Yüksek Teknoloji Enstitüsü, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- **2012 – 2015** : 32 Bit bilgisayar Hizmetleri, Yazılım Mühendisi
- **2015 – 2016** : Mastercard, Yazılım Mühendisi
- **2016 – 2018** : Softtech, Yazılım Mühendisi
- **2018 – 2020** : Softtech, Yazılım Analisti
- **2020 -** : Softtech, Yazılım Mimarı