YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BULANIK ANALİTİK HİYERARŞİ PROSESİ

Matematik Mühendisi Tamer ÜZGÜN

**FBE Matematik Anabilim Dalında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof.Dr. Fatma TİRYAKİ

İSTANBUL, 2006

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

$w_i, w_j, w_1, w_{2,...}$	Faaliyetlere göre ağırlıklar
d	Sentetik derece değeri,
d	Tolerans parametresi
A	İkili karşılaştırmalar matrisi
a_{ij}	A matrisinin i. satır, j. sütun elemanı
$\forall$	Her
W	Ağırlıklar vektörü
Σ	Toplam
$\prod$	Çarpım
α	α-kesim seviyesi
μ	Bulanık kümenin üyelik işlevi
λ	İyimserlik katsayısı
u	Üst (Upper)
m	Orta (Middle)
l	Alt (Lower)
S_i	i.nesneye ait sentetik derece değeri
$\otimes$	Bulanık sayılarda çarpma işlemi
$\oplus$	Bulanık sayılarda toplama işlemi

KISALTMA LİSTESİ

ÇKKV	Çok Kriterli Karar Verme
ÇAKV	Çok Amaçlı Karar Verme
ÇNKV	Çok Nitelikli Karar Verme
ÜBS	Üçgensel Bulanık Sayılar
YBS	Yamuksal Bulanık Sayılar
BTP	Bulanık Tercih Programlaması
LP	Lineer Programlama
AHP	Analitik Hiyerarşi Prosesi
CR	Tutarlılık Oranı
CI	Tutarlılık İndeksi
RI	Rassal İndeks

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖNSÖZ

Çalışmam sırasında çalışmamın sonuca ulaşması için sonsuz yardımlarından dolayı eşim Sayın Aylan VARDAR ÜZGÜN'e, desteğini benden esirgemeyen ve göstermiş olduğu büyük sabır ve özveri için danışmanım Sayın Prof.Dr.Fatma TİRYAKİ'ye teşekkürlerimi sunarım.

ÖZET

Günümüzde hayat şartlarının zorluğu, seçeneklerin çokluğu gibi birçok sebepten dolayı özellikle iş hayatında doğru kararlar verebilmek ve bu doğru kararlar ışığında başarılı olmak çok önemlidir. İnsanlar karar verme aşamasında genelde içgüdüsel hareket etmektedirler. Fakat bazen karşımıza o kadar kompleks durumlar çıkar ki karar verirken neyi nasıl karşılaştıracağımızı bilemeyiz. Bu gibi durumlarda işimizi kolaylaştırmak için durumu başlıklar halinde maddelendirip bu maddeleri de alt başlıklara indirgeyebilmek gerekir. İşte karar verme sürecinde bu işlemi yaptıktan sonra alt maddelere indirgediğimiz durumun bu alt maddelerini kendi aralarında daha sonra üst başlıklarını ve asıl hedefimiz kapsamında bu üst başlıkları kendi aralarında mukayese etmek daha kolaydır. Bu mukayeselerde 1970'lerde T.L. Saaty tarafından geliştirilen Analitik Hiyerarşi Prosesini kullanmak uygun olacaktır. Karar vericilerin değerlendirmelerinde kesin sayılardan ziyade belirsizliği bulanık sayılar ile ifade etme istekleri sonucu bulanık kavramlar katılarak AHP bulanık çerçeveye genişletilmiştir. Böylece karar verme aşamasında kişi ya da kurumların verecekleri önemli bir karar daha ayrıntılı analiz edilebilir.

Bulanık Analitik Hiyeraşi Prosesi başlıklı bu çalışmada altı farklı yöntem ayrıntılı bir şekilde ele alınmıştır. Bu metodlar için öncelikle gerekli olan temel bulanıklık kavramı, bu kavram altında yapılan işlemler, kümeler ve özellikleri verilmiştir. Ayrıca incelenen bulanık analitik hiyerarşi metodlarının avantaj ve dezavantajları ile genel değerlendirilmesi yapılmıştır. Son olarak bu tezde, Enea ve Piazza'nın önerdiği iki yaklaşımdan birincisinin Java programlama dili ile yazdığımız programı bulunmaktadır.

Anahtar Kelimeler: Karar verme, analitik hiyerarşi prosesi, bulanık analitik hiyerarşi prosesi, çok kriterli karar verme, çok amaçlı karar verme.

ABSTRACT

Especially for business life, it's very hard and important to make right and good decisions and be succesfull, because there are a lot of difficulties of life and choices. Human moves with his sense at decision period but sometimes there can be very complex situations that we can't know how to compare which of these criteria and candidates with others. At that time we have to reduce the problem into little parts to see it easly. Then now, after these reduction we can compare all these little parts, (criteria, candidates) one by one with others under our main goal. It is convenient to use Saaty's Analytical Hierarchy Process approach in these comparement. Since decision makers need to express uncertainty with fuzzy numbers instead of real numbers, this approach extended to fuzzy environment. Thus we can analyse this important decision more detailed.

In this project under Fuzzy Analytical Hierarchy Process subject, there are six methods studied. Fuzziness, fuzzy operators, fuzzy sets and properties, detailed desciriptions of studied methods are subjects of this project. Also the advantages and disadvantages of these methods are judged. Finally there is a program written in Java for Enea and Piazza's I. method.

Keywords: Decision making, analytical hierarchy process, fuzzy analytical hierarchy process, multi-criteria decision making, multi-objective decision making.

1. GİRİŞ

İnsanlar günlük hayatlarında her an kesin olarak bilemedikleri ya da bazen kesin sandıkları durumlarla karşılaşırlar. Bu durumlarda karar vermeleri gerektiğinde, öncelikle konunun zorluğunu aşabilmek için eldeki verileri nasıl değerlendirmeleri gerektiğini düşünürler. Bu aşamada eldeki verilerin özellikleri çok önemlidir. Eğer veriler tek tipte değilse problemin zorluk derecesi artacaktır. Bu veriler incelendiğinde; nitel ve ya nicel (kalitatif ve ya kantitatif) şekillerde olabileceklerini görürüz. Örneğin üniversitede işe almak üzere iki öğretim üyesini kıyaslamak zorunda olsak mesela bu kişilerin yayınlarını kıyaslarken "3 indeksli dergide yayını, 7 ulusal toplantı sunumu var" şeklinde elimizde gerçel sayılarla durumu ifade edebilecek veriler bulunur. İkinci aday içinde elimizde sayısal veriler bulunacağından bu iki kişiyi verileri oranlayarak kıyaslayabiliriz. Fakat bu kişileri yaratıcılıkları açısından ya da araştırma yeteneği açısından değerlendirmeye kalkarsak elimizde "iyi, çok iyi, kötü, çok kötü" gibi yoruma dayalı nitel veriler bulunur. Bu şekilde karar vermemiz gereken problemde bulanık bir ortam oluşur ve çözümlenmesi için de yine bulanık yapılarda çözüm yapmamızı sağlayan yöntemleri kullanmak doğru olur.

Araştırmacılar bu gibi durumlar için birçok yöntem geliştirmişlerdir. Bu tezin amacı karar verme sürecinde bulanık mantığı kullanarak çoklu kriterlere göre klasik ve bulanık analitik hiyerarşi sürecini ve metodlarını incelemektir. Bu tezde amaçlanan, belirli değer önceliklerini belirli kriterler ve kısıtlar altında kullanarak en uygun şekilde karar vermeye yardımcı olan yöntemleri tanıtmaktır. Kullanılacak yöntemlerle kişi kriterler arasında ikili karşılaştırmalar yaparak kendisine en uygun olanı seçebilecektir. Alternatiflerin hep birlikte değil de ikili olarak kriterlere bağlı değerlendirilmesi ve kişinin önceliklerine göre sentezlenmesi kararın uygunluğunun belirlenmesinde etkili olacaktır.

Bu çalışmada ilk olarak Zadeh'in temellerini attığı bulanık matematik, bulanık kümeler, üyelik fonksiyonları, bulanık sayılar ve bulanık matematiksel işlemler hakkında detaylı bilgiler verilmektedir.

Daha sonraki bölümlerde bulanık çok kriterli karar verme başlığı altında çok amaçlı karar verme, çok nitelikli karar verme, klasik analitik hiyerarşi prosesi ele alınmıştır. Tezin asıl konusu olan Bulanık Analitik Hiyerarşi Prosesi ve Yöntemleri başlığı altında ise en küçük logaritmik farkların karelerini kullanan Laarhoven ve Pedrycz yaklaşımı, yamuksal bulanık sayıları kullanan Buckley yaklaşımı, sentetik derece değerlerini kullanarak çözüm öneren Chang yaklaşımı, lineer ve lineer olmayan denklem sistemi çözümlerini kullanan Mikhailov yaklaşımı, son olarak da Chang'ın yaklaşımını geliştirerek yeni iki farklı yöntem öneren Enea ve Piazza'nın metodlarının detaylı anlatımları bulunmaktadır. Bu yöntemler örneklerle detaylı şekilde açıklanmıştır..

Takip eden bölümde ise açıklanan metodlardan Enea ve Piazza yaklaşımının aritmetik ortalama kullanılarak uygulanan birinci metodunun java programlama dili ile yazdığımız bilgisayar programı yer almaktadır.

Sonuç bölümünde ayrıntılı anlatılan yöntemlerin avantaj ve dezavantajları, birbirlerine göre etkin ve eksik yönleri değerlendirilmiştir.

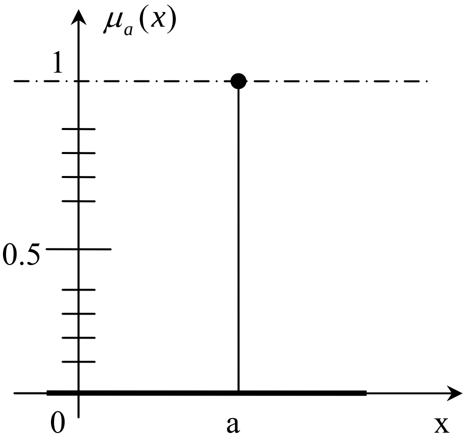
2. BULANIK SAYILAR, KÜMELER VE İŞLEMLER [Kaufmann ve Gupta, 1988, 19-35]

Bulanık sayılar ilk olarak 1965 yılında Zadeh tarafından yapılan çalışmalar ile duyulmaya başlanmıştır. Daha sonraları belirsiz, bulanık durumların sayısal olarak ifade edilir hale gelmesi ve dolayısıyla bu durumlarda işlemler yapılarak matematiksel çözümler oluşturulabilmesi ile bir çok karar metodu ortaya çıkmıştır. Bu bulanık yapıların temel özellikleri ve bu yapılardaki işlemler aşağıda anlatılmıştır.

2.1. Güven Aralığı

Şekil 2.1 $a \in R$ olacak şekilde bir adi a sayısını göstermektedir. Burada

$$\mu_a(x) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases} \qquad (2.1)$$



Şekil 2.1. Adi bir a sayısı

Şekil 2.2. de gösterildiği gibi [a_1,a_3] güven aralığında bir A $\in R$

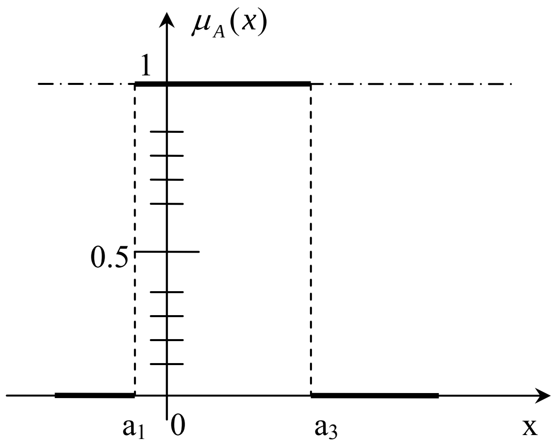
$$\mu_a(x) = \begin{cases} 0, & x < a_1 \\ 1, & a_1 \leq x \leq a_3 \\ 0, & x > a_3 \end{cases} \qquad (2.2)$$

sayısı tanımlayalım.

R'de bir güven aralığı bir çeşit belirsizlik ifade eden R'nin adi bir alt kümesidir. Biliyoruz ki A a_1'den daha küçük ve a_3'ten daha büyük olamaz. Güven aralığının sembolik gösterimi genellikle

$$A = [\, a_1, a_3 \,] \tag{2.3}$$

şeklinde de yazılır.



Şekil 2.2. Güven aralığı ile bir A sayısı

Şimdi Şekil 2.3.'teki gibi bir bulanık sayıyı dikkate alalım. Genel olarak bulanık bir sayı R'de "normal" ve "konveks" olan bulanık bir alt kümedir. Burada normallik

$$\exists x \in R : \underset{x}{\forall}\, \mu_A(x) = 1\,, \tag{2.4}$$

yani bulanık kümenin R'deki en büyük değeri "1"dir anlamındadır.

"konveks"lik, x eksenine paralel olan bir $\alpha -$ kesimi,

$$A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right]\text{'nın,} \tag{2.5}$$

$$\alpha' < \alpha \Rightarrow \left(a_1^{(\alpha')} \le a_1^{(\alpha)}, a_3^{(\alpha')} \ge a_3^{(\alpha)}\right) \tag{2.6}$$

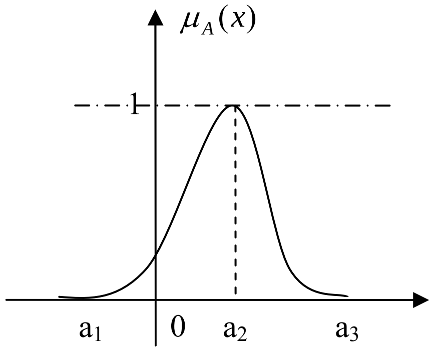
özelliğini sağladığı anlamındadır. Diğer bir deyişle A_α *ile* $\alpha -$ kesimini

$$A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right],\ A_{\alpha'} = \left[a_1^{(\alpha')}, a_3^{(\alpha')}\right]$$

olarak ifade edersek, konvekslik şartı

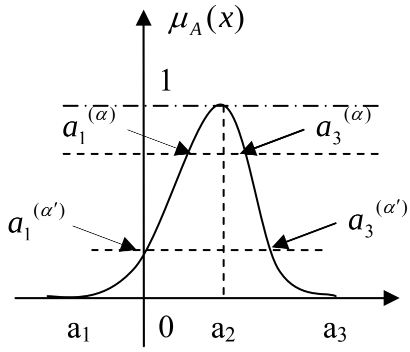
$$\alpha' < \alpha \Rightarrow \left(A_\alpha \subset A_{\alpha'}\right) \tag{2.7}$$

anlamına gelir.



Şekil 2.3. A bulanık sayısı

Şekil 2.4. bir A bulanık sayısı için konvekslik ve normallik özelliklerini göstermektedir.



Şekil 2.4. $\alpha-$kesimi ile normal ve konveks bir bulanık sayı

2.2. Güven Aralıklarında ve Bulanık Sayılarda Aritmetik İşlemler

Güven aralıkları ile tanımlanmış A ve B sayıları için bulanık sayıların aritmetiği, bazı işlemlerle aşağıda tanımlanmıştır.

$\forall a_1, a_3, b_1, b_3 \in R:$

$A = [a_1, a_3],\ B = [b_1, b_3]$

olmak üzere

(i) Toplama:

$$[a_1, a_3](+)[b_1, b_3] = [a_1 + b_1, a_3 + b_3], \tag{2.8}$$

(ii) Çıkarma:

$$[a_1, a_3](-)[b_1, b_3] = [a_1 - b_3, a_3 - b_1], \tag{2.9}$$

(iii) Çarpma:

$$[a_1, a_3](.)\ [b_1, b_3] = [a_1.b_1 \wedge a_1.b_3 \wedge a_3.b_1 \wedge a_3.b_3, a_1.b_1 \vee a_1.b_3 \vee a_3.b_1 \vee a_3.b_3], \tag{2.10}$$

(iv) Ters alma:

$a_1 \leq 0 \leq a_3$ dışındaki tüm değerler için

$$[a_1, a_3]^{-1} = \left[\frac{1}{a_1} \wedge \frac{1}{a_3}, \frac{1}{a_1} \vee \frac{1}{a_3}\right], \tag{2.11}$$

(v) Bölme:

$a_1 \leq 0 \leq a_3$ dışındaki tüm değerler için

$$[a_1, a_3](:)\,[b_1, b_3] = \left[\frac{a_1}{b_1} \wedge \frac{a_1}{b_3} \wedge \frac{a_3}{b_1} \wedge \frac{a_3}{b_3}, \frac{a_1}{b_1} \vee \frac{a_1}{b_3} \vee \frac{a_3}{b_1} \vee \frac{a_3}{b_3}\right], \tag{2.12}$$

Eğer bulanık küme R^+'da tanımlı ise son üç formül

(iii') Çarpma:

$$[a_1, a_3](.)\,[b_1, b_3] = [a_1.b_1, a_3.b_3], \tag{2.10'}$$

(iv') Ters alma:

$$[a_1, a_3]^{-1} = \left[\frac{1}{a_3}, \frac{1}{a_1}\right], \tag{2.11'}$$

(v') Bölme:

$$[a_1, a_3](:)\,[b_1, b_3] = \left[\frac{a_1}{b_3}, \frac{a_3}{b_1}\right], \tag{2.12'}$$

şeklinde sadeleştirilebilir.

Diğer iki işlem

(vi) Minimum:

$$[a_1, a_3](\wedge)\,[b_1, b_3] = [a_1 \wedge b_1, a_3 \wedge b_3], \tag{2.13}$$

(vii) Maksimum:

$$[a_1, a_3](\vee)\,[b_1, b_3] = [a_1 \vee b_1, a_3 \vee b_3], \tag{2.14}$$

Örnek 2.1: A ve B sayıları güven aralığında aşağıdaki şekilde tanımlansın:

$A = [3.25, 5.46]$ ve $B = [-2.12, 7.51]$.

Buradan

$$A\,(+)B = [3.25, 5.46](+)\,[-2.12, 7.51] = [1.13, 12.97].$$

$$A\,(-)\,B = [3.25, 5.46](-)\,[-2.12, 7.51] = [4.26, 7.58].$$

$$B^{-1} = [2.12, 7.51]^{-1} = \left[\frac{1}{2.12} \wedge \frac{1}{7.51}, \frac{1}{2.12} \vee \frac{1}{7.51}\right] = [0.13, 0.47].$$

$$\begin{aligned}
A\,(.)\,B &= [3.25, 5.46](.)\,[-2.12, 7.51] \\
&= [(3.25).(-2.12) \wedge (3.25).(7.51) \wedge (5.46).(-2.12) \wedge (5.46).(7.51), \\
&\quad (3.25).(-2.12) \vee (3.25).(7.51) \vee (5.46).(-2.12) \vee (5.46).(7.51)] \\
&= [-11.57, 41.00].
\end{aligned}$$

$$\begin{aligned}
A\,(:)\,B &= [3.25, 5.46](:)\,[-2.12, 7.51] \\
&= \left[\frac{(3.25)}{(-2.12)} \wedge \frac{(3.25)}{(7.51)} \wedge \frac{(5.46)}{(-2.12)} \wedge \frac{(5.46)}{(7.51)}, \frac{(3.25)}{(-2.12)} \vee \frac{(3.25)}{(7.51)} \vee \frac{(5.46)}{(-2.12)} \vee \frac{(5.46)}{(7.51)}\right], \\
&= [0.43, 2.57].
\end{aligned}$$

$$A\,(\wedge)\,B = [3.25, 5.46](\wedge)[-2.12, 7.51] = [3.25 \wedge -2.12, 5.46 \wedge 7.51] = [-2.12, 5.46].$$

$$A\,(\vee)\,B = [3.25, 5.46](\vee)[-2.12, 7.51] = [3.25 \vee -2.12, 5.46 \vee 7.51] = [3.25, 7.51].$$

Yukarıda belirtilen ifadeler her α seviyesindeki A ve B bulanık sayıları için genişletilebilir.

Buna toplama üzerinden bir örnek verecek olursak

$\forall \alpha \in [0,1], a_1^{(\alpha)}, a_3^{(\alpha)}, b_1^{(\alpha)}, b_3^{(\alpha)} \in R$ olmak üzere,

$$\left[a_1^{(\alpha)}, a_3^{(\alpha)}\right] + \left[b_1^{(\alpha)}, b_3^{(\alpha)}\right] = \left[a_1^{(\alpha)} + b_1^{(\alpha)}, a_3^{(\alpha)} + b_3^{(\alpha)}\right] \qquad (2.15)$$

olur.

Şimdi bu elde ettiğimiz sonuçları üyelik fonksiyonlarını kullanarak genişletirsek,

$\forall x, y, z \in R$ olmak üzere

Toplama:

$$\mu_{A(+)B}(z) = \bigvee_{z=x+y} (\mu_A(x) \wedge \mu_B(y)), \qquad (2.16)$$

Çıkarma:

$$\mu_{A(-)B}(z) = \bigvee_{z=x-y} (\mu_A(x) \wedge \mu_B(y)), \qquad (2.17)$$

Çarpma:

$$\mu_{A(.)B}(z) = \bigvee_{z=x.y} (\mu_A(x) \wedge \mu_B(y)), \qquad (2.18)$$

Bölme:

$$\mu_{A(:)B}(z) = \bigvee_{z=x/y} (\mu_A(x) \wedge \mu_B(y)), \qquad (2.19)$$

Minimum:

$$\mu_{A(\wedge)B}(z) = \bigvee_{z=x\wedge y} (\mu_A(x) \wedge \mu_B(y)), \qquad (2.20)$$

Maksimum:

$$\mu_{A(\vee)B}(z) = \bigvee_{z=x\vee y} (\mu_A(x) \wedge \mu_B(y)). \tag{2.21}$$

Tüm güven aralıkları ve bulanık sayılar adi bir sayı ile çarpılabilir. Unutmayalım ki

$$[a,a] = a,\ a \in R \tag{2.22}$$

de adi bir sayıdır.

Çarpma için yukarıda verilen tüm formüller

$$a.[b_1,b_3] = [a.b_1 \wedge a.b_3,\ a.b_1 \vee a.b_3] \tag{2.23}$$

şeklindeki basit formda ifade edilebilir.

Örnek 2.2: (2.23) denklemini uygulamalı olarak gösterelim.

(i) $\quad 3.18\,[-0.52\ ,\ 2.17] = [(3.18)(-0.52) \wedge (3.18)(2.17),(3.18)(-0.52) \vee (3.18)(2.17)]$
$\quad = [-1.65 \wedge 6.90\ ,\ -1.65 \vee 6.90]$
$\quad = [-1.65\ ,\ 6.90]$

(ii) $\quad -4.15\,[-3.55\ ,\ -0.21] = [(-4.15)(-3.55) \wedge (-4.15)(-0.21),(-4.15)(-3.55) \vee (-4.15)(-0.21)]$
$\quad = [14.73 \wedge 0.87\ ,\ 14.73 \vee 0.87]$
$\quad = [0.87\ ,\ 14.73].$

Her α seviyesindeki bulanık sayıları ifade ederek (2.23) ifadesinden çıkan sonuçlar aynı zamanda bulanık sayılar için genişletilebilir. Örneğin $\forall \alpha \in [0,1]$, *ve* $b_1^{(\alpha)}, b_3^{(\alpha)} \in R$ için

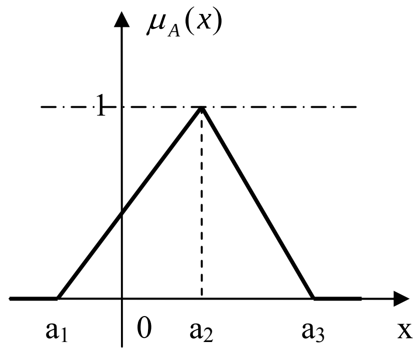
$$a.\left[b_1^{(\alpha)}, b_3^{(\alpha)}\right] = \left[a.b_1^{(\alpha)} \wedge a.b_3^{(\alpha)}, a.b_1^{(\alpha)} \vee a.b_3^{(\alpha)}\right] \quad , \tag{2.24}$$

dir.

Bu bağlantılar bulanık doğal ve tam sayılar için geçerlidir.

2.3. Üçgensel bulanık sayılar (Ü.B.S.)

Bulanık sayıların sonsuz bir kümesi vardır, ancak burada biz bu kümenin üçgensel bulanık sayılar denilen özel bir sınıfını tanımlayacağız.



Şekil 2.5. Üçgensel bulanık sayı $A=(a_1,a_2,a_3)$

A üçgensel bulanık sayısı (a_1,a_2,a_3) üçlüsü ile tanımlanabilir. Şekil 2.5.'te gösterilen üçgensel bulanık sayılar için üyelik fonksiyonu

$$\mu_A(x) = \begin{cases} 0 \quad , & x < a_1 \\ \dfrac{x-a_1}{a_2-a_1}, & a_1 \le x \le a_2 \\ \dfrac{a_3-x}{a_3-a_2}, & a_2 \le x \le a_3 \\ 0 \quad , & x > a_3 \end{cases} \tag{2.25}$$

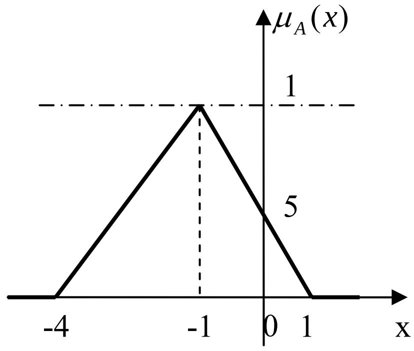
şeklinde tanımlanır.

Alternatif olarak α seviyesinde güven aralığını tanımlayarak üçgensel bulanık sayıyı $\forall \alpha \in [0,1]$ için

$$A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right] = [(a_2-a_1)\alpha + a_1, -(a_3-a_2)\alpha + a_3] \tag{2.26}$$

şeklinde karakterize ederiz.

Örnek 2.3: Şekil 2.6.'da gösterilen Ü.B.S. $(a_1,a_2,a_3) = (-4,-1,1)$ üçlüsü ile tanımlanmıştır. Daha sonra bu Ü.B.S. için üyelik fonksiyonu aşağıdaki şekilde verilmiştir.



Şekil 2.6. Üçgensel bulanık sayı A=(-4,-1,1)

$$\mu_A(x) = \begin{cases} 0 & , \quad x < -4 \\ \dfrac{x+4}{3} & , \quad -4 \le x \le -1 \\ \dfrac{1-x}{2} & , \quad -1 \le x \le 1 \\ 0 & , \quad x > 1 \end{cases}$$

Alternatif olarak $\alpha \in [0,1]$ seviyesinde güven aralığını kullanarak Ü.B.S.

$$\left[a_1^{(\alpha)}, a_3^{(\alpha)}\right] = [3\alpha - 4, -2\alpha + 1]$$

şeklinde karakterize edilebilir.

Ü.B.S. Üzerinde Tanımlanan Cebirsel İşlemlerin Bazıları

A=(a₁,a₂,a₃) ve *B=(b₁,b₂,b₃)* Ü.B.S.'lar olmak üzere

(i) Toplama:

$$\text{A } (+)\text{B} = (a_1, a_2, a_3)(+)(b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3), \qquad (2.27)$$

(ii) Çıkarma:

A (-) B = $(a_1, a_2, a_3)(-)(b_1, b_2, b_3) = (a_1 - b_1, a_2 - b_2, a_3 - b_3)$, (2.28)

(iii) Simetriğini alma:

-(A) = $(-a_3, -a_2, -a_1)$, (2.29)

(iv) Çarpma, bölme ve tersini alma:

Çarpma, ters alma ve bölme işlemleri için üçlüler kullanılmaz. Bununla beraber hesaplama her α seviyesi için güven aralıklarını kullanarak yapılabilir. R'deki Ü.B.S.'lar için hesaplamada α, "0" dan "1"e yükselirken pozitif ve negatif değerler ile minimum ve maksimumun etkilerini incelemek yoluyla seviyeleri çözümlemeliyiz. R^+'da hesaplamalar çok kolaydır.

Not :

Toplama, çıkarma ve simetriğini alma işlemlerinin sonuçları birer Ü.B.S. dır.

Ü.B.S.'lar üzerindeki çarpma, bölme ve tersini alma işlemleri muhakkak Ü.B.S. vermez.

Ü.B.S.'lar üzerindeki maximum ve minimum işlemleri muhakkak Ü.B.S. vermez, ancak Ü.B.S.'larla bu işlemlerin sonuçlarına yaklaşabiliriz.

R^+'da çarpma için bir örnek verelim.

Örnek 2.4: A=(-3,2,4) ve B=(-1,0,5) Ü.B.S.'ları için

A (+)B = $(-3,2,4)(+)(-1,0,5) = (-4,2,9)$, (2.30)

Her α seviyesi için güven aralıklarını kullanarak aşağıdaki hesaplamayı kullanarak iki Ü.B.S.'yı toplayacağız.

$$A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right] = \left[(a_2 - a_1)\alpha + a_1, -(a_3 - a_2)\alpha + a_3\right] = \left[5\alpha - 3, -2\alpha + 4\right],$$

$$B_\alpha = \left[b_1^{(\alpha)}, b_3^{(\alpha)}\right] = \left[(b_2 - b_1)\alpha + b_1, -(b_3 - b_2)\alpha + b_3\right] = \left[\alpha - 1, -5\alpha + 5\right]$$

Bu şekilde toplam

$$A_\alpha(+)B_\alpha = \left[5\alpha - 3 + \alpha - 1, -2\alpha + 4 - 5\alpha + 5\right] = \left[6\alpha - 4, -7\alpha + 9\right]. \tag{2.31}$$

(2.30) ve (2.31)' den elde edilen sonuçları $\alpha = 0$ ve $\alpha = 1$ güven seviyelerinde doğruluğu kanıtlanabilir. Böylece

$\alpha = 0$ için $A_0(+)B_0 = [-4,9]$

ve

$\alpha = 1$ için $A_1(+)B_1 = [2,2] = 2$

olarak bulunur. Üyelik fonksiyonları kullanmak problemi daha karmaşık hesaplamalara sürüklemektedir ,dolayısıyla burada onları kullanmayacağız.

Önce güven seviyesini daha sonra da üçlüleri kullanarak A, B için çıkartma işlemini inceleyeceğiz.

$$A_\alpha(-)B_\alpha = \left[5\alpha - 3 - (5\alpha + 5), -2\alpha + 4 - (\alpha - 1)\right] = \left[10\alpha - 8, -3\alpha + 5\right], \tag{2.32}$$

$$A\,(-)\,B = (-3,2,4)(-)(-1,0,5) = (-3 - 5, 2 - 0, 4 - (-1)) = (-8,2,5). \tag{2.33}$$

$\alpha = 0$ için $A_0(-)B_0 = [-8,5]$ ve $\alpha = 1$ için $A_1(-)B_1 = [2,2] = 2$ olduğunu belirtelim.

Örnek 2.5: A=(2,3,5) ve B=(1,4,8) Ü.B.S.'ları için güven aralıkları

$$A_\alpha = [(3-2)\alpha + 2, -(5-3)\alpha + 5] = [\alpha + 2, -2\alpha + 5],$$

$$B_\alpha = [(4-1)\alpha + 1, -(8-4)\alpha + 8] = [3\alpha + 1, -4\alpha + 8],$$

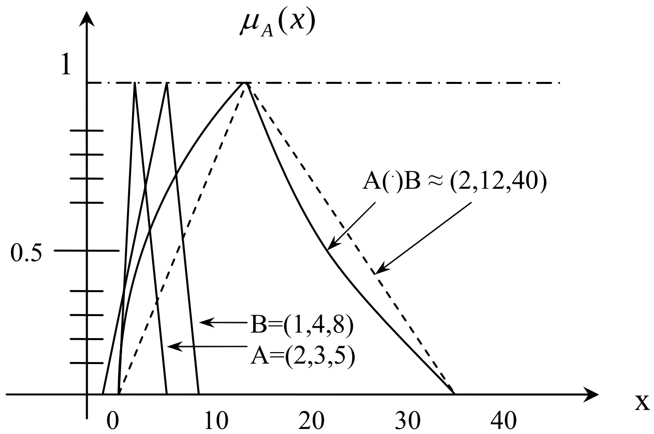
dır. Her bir $\alpha \in [0,1]$ seviyesinde çarpım

$$\begin{aligned} A_\alpha (.) B_\alpha &= [\alpha + 2, -2\alpha + 5](.)[3\alpha + 1, -4\alpha + 8], \\ &= [(\alpha + 2)(3\alpha + 1), (-2\alpha + 5)(-4\alpha + 8)] \\ &= [3\alpha^2 + 7\alpha + 2, 8\alpha^2 - 36\alpha + 40] \end{aligned} \tag{2.34}$$

ile verilmiştir.

$\alpha = 0$ için $A_0 (.) B_0 = [2,40]$ ve $\alpha = 1$ için $A_1 (.) B_1 = [12,12] = 12$ oduğunu belirtelim.

Şekil 2.7. de A ve B Ü.B.S.'ları ile bunların çarpımlarının sonuçları verilmektedir. Dikkat edilmelidir ki çarpımın segmentleri düzgün doğrular değil parabollerdir. Bunula beraber Ü.B.S.'lar ile bu sonuç $A(.)B \approx [2,12,40]$ sonucuna yaklaştırılabilir.

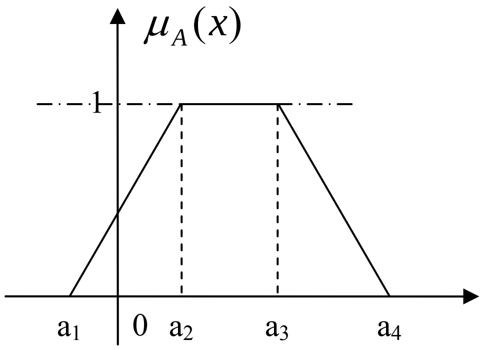


Şekil 2.7. A ve B Ü.B.S.'larının çarpı

Çarpma işlemine benzer şekilde, Ü.B.S.'yı her $\alpha \in [0,1]$ seviyesinde güven aralığı cinsinden ifade ederek ve (2.8) eşitliğinden (2.12) eşitliğine kadarki sonuçları kullanarak Ü.B.S.'larda bölme, ters alma, maksimum ve minimum işlemleri de genişletilebilirler.

2.4. Yamuksal Bulanık Sayılar (Y.B.S.)

Şimdi diğer bir önemli bulanık sayı çeşidi olan yamuksal bulanık sayıları tanımlayalım. Bu durumda $\alpha = 1$ için bir noktamız yok, bunun yerine Şekil 2.8.'den de görülebilen (a_2,a_3) aralığında düzgün uzanan bir doğrumuz var.



Şekil 2.8. A=(a_1,a_2,a_3,a_4) Yamuksal Bulanık Sayısı (Y.B.S.)

Y.B.S.'ların özellikleri Ü.B.S.'lar ile benzerdir. Bir Y.B.S.

$$A=(a_1, a_2, a_3, a_4) \tag{2.35}$$

şeklindeki bir dörtlü ile tam olarak ifade edilir.

Bir Y.B.S.'yı α seviyesindeki güven aralığı ile karakterize edebiliriz.

Böylece $\forall \alpha \in [0,1]$ için

$$A_\alpha = [(a_2 - a_1)\alpha + a_1, -(a_4 - a_3)\alpha + a_3] \tag{2.36}$$

olur.

Y.B.S.'nın üyelik fonksiyonu

$$\mu_A(x) = \begin{cases} 0 & , \quad x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2 \\ 1 & , \quad a_2 \le x \le a_3 \\ \dfrac{a_4 - x}{a_4 - a_3}, & a_3 \le x \le a_4 \\ 0 & , \quad x > a_4 \end{cases} \tag{2.37}$$

şeklinde karakterize edilir.

Bir Ü.B.S.'nın $a_2 = a_3$ durumu ile bir Y.B.S.'nın özel bir durumu olduğuna dikkat edilmelidir. Ü.B.S.'ların tüm cebirsel işlem sonuçlarını Y.B.S.'lara genişletebiliriz. Bunlardan bazıları aşağıda özetlenmiştir.

Y.B.S. Üzerinde Tanımlanan Cebirsel İşlemlerin Bazıları

$A=(a_1,a_2,a_3,a_4)$ ve $B=(b_1,b_2,b_3,b_4)$ Y.B.S.'lar olmak üzere

(i) Toplama:

$$\text{A }(+)\text{B} = (a_1, a_2, a_3, a_4)(+)(b_1, b_2, b_3, b_4) = (a_1 + b_1, a_2 + b_2, a_3 + b_3, a_4 + b_4), \tag{2.38}$$

(ii) Çıkarma:

$$\text{A }(-)\text{ B} = (a_1, a_2, a_3, a_4)(-)(b_1, b_2, b_3, b_4) = (a_1 - b_1, a_2 - b_2, a_3 - b_3, a_4 - b_4), \tag{2.39}$$

(iii) Simetriğini alma:

$$-(\text{A}) = (-a_4, -a_3, -a_2, -a_1), \tag{2.40}$$

Not :

Toplama, çıkarma ve simetriğini alma işlemlerinin sonuçları birer Y.B.S. dır.

Y.B.S.'lar üzerindeki çarpma, bölme ve tersini alma işlemleri muhakkak Y.B.S. vermez.

Y.B.S.'lar üzerindeki maximum ve minimum işlemleri muhakkak Y.B.S. vermez, ancak Y.B.S.'larla bu işlemlerin sonuçlarına yaklaşabiliriz.

Bununla beraber Ü.B.S. olduğu gibi, bu işlemlerin sonuçlarına Y.B.S.'larla yaklaşabiliriz.

Şimdi birkaç örneği inceleyelim.

Örnek 2.6. Şekil 2.9.da gösterilen A ve B, Y.B.S.'ları

A=(-3,-1,2,7) ve B=(-1,5,6,8)

şeklinde verilmiş olsun. A ve B toplamı

$$A\,(+)B = (-3,-1,2,7)(+)(-1,5,6,8) = (-4,4,8,15) \tag{2.41}$$

olarak bulunur. Bu hesaplama α seviyesinde güven aralıkları cinsinden de ifade edilebilir. Böylece

$$A_\alpha = \left[a_1^{(\alpha)}, a_4^{(\alpha)}\right] = [2\alpha - 3, -5\alpha + 7]$$

$$B_\alpha = \left[b_1^{(\alpha)}, b_4^{(\alpha)}\right] = [6\alpha - 1, -2\alpha + 8]$$

$$A_\alpha + B_\alpha = [8\alpha - 4, -7\alpha + 15] \tag{2.42}$$

elde edilir.

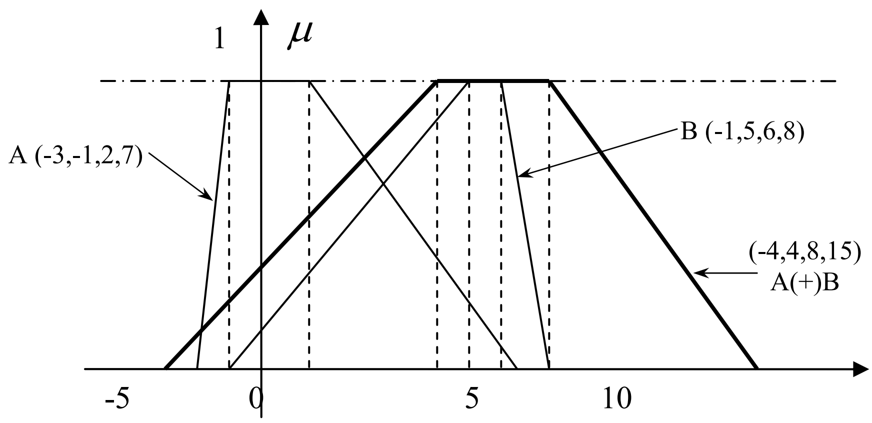
Bu sonuçları $\alpha = 0$ *ve* $\alpha = 1$ için sağlarsak

$\alpha = 0$ için A_0+B_0 = [-4,15]

$\alpha = 1$ için A_1+B_1 = [4,8]

bulunur.

Bu sonuçlar (2.41) ile aynıdır ve Şekil 2.9.' da görülmektedir.



Şekil 2.9 İki Y.B.S.'nın toplamı

Çarpma, ters alma ve bölme işlemleri için dörtlüler kullanılmaz. Bununla beraber hesaplama her α seviyesi için güven aralıklarını kullanarak yapılabilir. R'deki Y.B.S.'lar için hesaplamada α "0" dan "1"e yükselirken pozitif ve negatif değerler ile minimum ve maksimumun etkilerini incelemek yoluyla seviyeleri çözümlemeliyiz. R^+'da hesaplamalar çok kolaydır.

Örnek 2.7. Şekil 2.10. R^+'da iki Y.B.S.'yı göstermektedir. Bu iki sayı

A=(1,5,6,9) ve B=(2,3,5,8)

dörtlüleri ile verilmiştir. Bu Y.B.S.'lar güven aralıkları kullanılarak karakterize edilebilir.

Buradan

$$A_\alpha = [4\alpha + 1, -3\alpha + 9]$$

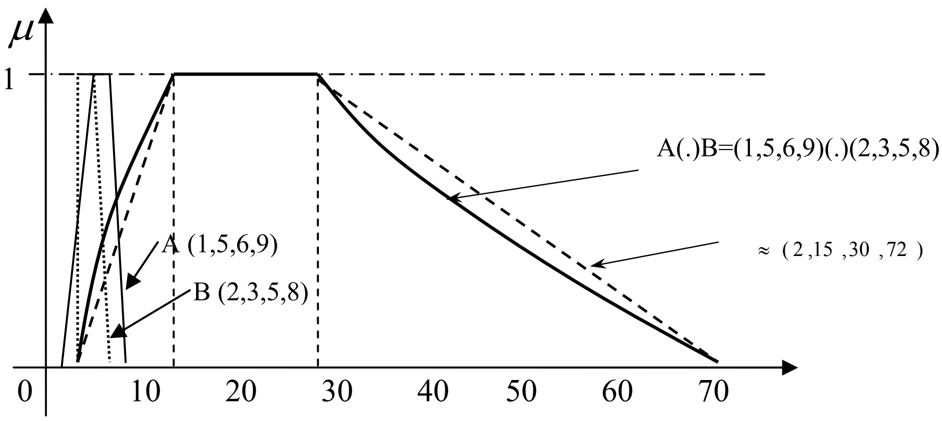
$$B_\alpha = [\alpha + 2, -3\alpha + 8]$$

olur. Her bir α seviyesi için çarpım

$$A_\alpha (.) B_\alpha = [(\alpha + 2).(4\alpha + 1), (-3\alpha + 8).(-3\alpha + 9)] = [(4\alpha^2 + 9\alpha + 2), (9\alpha^2 - 51\alpha + 72)]$$

şeklinde verilir.

Şekil 2.10.' da A ve B, Y.B.S.'ları ile bunların çarpımlarının sonuçları verilmektedir. Dikkat edilmelidir ki çarpımın segmentleri düzgün doğrular değillerdir. Bununla beraber bu sonuç Y.B.S.'lar ile $A(.)B \approx [2,15,30,72]$ sonucuna yaklaştırılabilir.



Şekil 2.10 İki Y.B.S.'nın çarpımı

3. BULANIK ÇOK KRİTERLİ KARAR VERME [Zimmerman, 1991, 265]

Yakın geçmişte faaliyetlerin istenirliklerine göre karşılaştırılmaları, ürünlerin uygunluğuna karar verilmesi veya karar problemlerinde optimal çözümlerin belirlenmesi bir çok durumda tek bir kriter veya tek bir amaç fonksiyonu kullanılarak yapılamadığı çok daha fazla aşikar hale gelmiştir. Bu durum çok kriterli karar vermeyi günümüzde çok daha gerekli hale getirmiştir.

Çok kriterili karar vermede karar vermeye konsantre olmuş iki ana dal geliştirilmiştir. Bunların ilki Çok Amaçlı Karar Verme, diğeri ise Çok Nitelikli Karar Verme'dir. Bu iki ana yaklaşım arasındaki temel fark karar uzaylarından kaynaklanır. Çok amaçlı karar verme sürekli karar uzaylarına yoğunlaşırken, çok nitelikli karar verme ayrık karar uzaylarına yoğunlaşır. Fakat bu kuralla birlikte çok amaçlı tamsayılı programlama gibi istisnalar da vardır.

Tek kriter ya da tek amaç fonksiyonunun çoğu zaman karar almada yeterli olmamasından dolayı çok kriterili karar verme son yıllarda literatür olarak oldukça gelişmiştir. Bulanık küme teorisinin bu alana oldukça önemli katkıları olmuştur.

3.1. Çok Amaçlı Karar Verme (ÇAKV) [Zimmerman, 1991, 265-267]

Matematiksel programlamada ÇAKV-problemi, sıkça vektör-maksimum problemi olarak anılır, ilk olarak Kuhn-Tucker tarafından 1951'de dile getirilmiştir.

Tanım 3.1. $Z(x) = (z_1(x),...,z_k(x))$, $x \in R^n$'den R^k'ya vektör değerli bir fonksiyon ve X çözüm uzayı olmak üzere vektör-maksimum problemi

$$\text{“maksimum”} \left\{ Z(x) \mid x \in X \right\}$$

şeklinde tanımlanır.

Kategorisel olarak vektör-maksimum optimizasyonu en az iki aşamaya ayrılabilir.

1) Etkin çözümlerin belirlenmesi,
2) Optimal uzlaşık bir çözümün belirlenmesi.

Tanım 3.2. $Max\{Z(x) \mid x \in X\}$ Tanım 3.1.'de verildiği gibi bir vektör-maksimum problemi olsun. Eğer

$$z_i(\hat{x}) \geq z_i(\overline{x}) \quad i = 1,2,...,k$$

ve en az bir

$$z_i(\hat{x}) > z_i(\overline{x}) \quad i = 1,2,...,k$$

koşullarını sağlayan bir $\hat{x} \in X$ yoksa $\overline{x}$ bir "etkin çözüm" olur.

Tüm etkin çözümlerin kümesine "tam çözüm" denir.

Tanım 3.3. Bir vektör-maksimum probleminin uzlaşık optimal çözümü karar verici tarafından diğer vektör değerli amaç fonksiyonun içerdiği bütün kriterler gözönüne alınarak tam çözüm uzayından diğer bütün kriterlere göre tercih edilen çözümlere bir $x \in X$ çözümü denir.

Burada vektör, vektör değerli amaç fonksiyonları olan lineer programlama problemlerinin çözümleri belirlenmeye çalışılmıştır.

1) Fayda yaklaşımı ,

2) Hedef programlama ,

3) Etkileşimli yaklaşımlar .

Bu yaklaşımlardan ilk ikisi, karar vericinin tercih fonksiyonunu alarak, bireysel amaç fonksiyonlarını, ya ağırlıklar ya da uzaklık fonksiyonları (burada uzaklık ile ideal çözümden olan uzaklık ifade edilmektedir) ile yapılan kombinasyonlarına göre belirleyebileceğini varsaymaktadır. Bu yaklaşımlar genellikle, bireysel amaç fonksiyonlarının kombinasyonu, lineer kombinasyonlarla elde edilen en yüksek faydaya sahip uzlaşık çözüme götürür kabulünü yapmaktadır. Üçüncü yaklaşım sadece yerel bilgileri kullanarak kabul edilebilir uzlaşık çözüme ulaşır.

3.2. Çok Nitelikli Karar Verme (ÇNKV) [Chen ve Hwang, 1992, 289-290]

Çok nitelikli karar verme problemi A_i $i = 1,2,...,m$ mümkün alternatifleri (adayları), X_j ($j=1,2,...,n$) alternatiflerin performanslarının ölçüldüğü kriterleri, x_{ij} ($i = 1,2,...,m$; $j=1,2,...,n$), A_i alternatifinin X_j kriterine göre skorunu, w_j ($j=1,2,...,n$) ise X_j kriterinin ağırlığını ifade etmek üzere

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & & X_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \end{array}$$

$W=[w_1 , w_2 ,... , w_n]$

matris formunda verilir.

Klasik ÇNKV yöntemlerinde x_{ij} ve w_j'ler kesin sayılar olarak bilinmektedir. Fayda fonksiyonu, dolaylı ya da dolaysız yoldan karar verici tarafından tanımlanır. A_i fayda fonksiyonu karar verici tarafında kapalı ya da açık olarak $U(x_1,x_2,...x_m)$ şeklinde tanımlanır. Fayda fonksiyonu, A_i alternatifi için performans ağırlıkları olan x_{ij}'leri birleştirerek sonuç fayda ağırlığı U_i'yi oluşturur. Burada sonuç fayda ağırlığı alternatifin karar vericinin faydasını ne kadar sağladığını ifade eder. Karar verici tarafından yüksek sonuç faydasını veren alternatifler tercih edilir. Sonuç faydaları reel sayı olduğundan tercih edilen alternatifler yüksek sonuç faydasına sahiptirler.

Gerçek hayat problemlerinde alternatiflerin performanslarına değer verirken elde ettiğimiz x_{ij} değerleri kesin, bulanık ve /ya da dilsel olabilir. Örneğin; bir üniversitede bir profesör kadrosuna 3 adayın başvuruda bulunduğunu düşünelim. Kriter olarak, Yaratıcılık (X_1), Olgunluk(X_2), İnsani İlişkiler(X_3) ve Yayın sayısı (X_4) ele alınsın. İlk üç kriter için performans ağırlıkları ölçülebilir olmayacaktır onun yerine "iyi", "orta", " yetersiz" vb. gibi dilsel değerlendirmeler kullanılır. X_4 kriteri için ise tam sayı değerleri kullanılabilir. Bu ÇKKV probleminde kesin ve bulanık veriler bir arada kullanılmıştır. Gerçek hayatta da birçok problem bu tiptedir.

Bulanık çok kriterli karar verme metotları, bulanık veri içeren problemleri çözmek için önerilmiştir. İlk olarak Bellman ve Zadeh 1970 yılında yaptıkları çalışmada, karar verme problemlerini bulanık küme teorisi ile ilişkilendirmişlerdir. 1977'de Baas ve Kwakernaak bulanık ÇNKV için çok yaygın olarak kullanılan bir sıralama metodu önermişlerdir. Geçtiğimiz yirmi yılda, çok sayıda bulanık ÇNKV metodları önerilmiştir.

3.3. Analitik Hiyerarşi Prosesi (AHP) [Kara Harp Okulu, 2003, 9. bölüm 1-29]

1970'lerin başında, Thomas Lorie Saaty, ABD Savunma Bakanlığında silahsızlanma, Orta Doğu sorunu, Sudan için ulaştırma sisteminin geliştirilmesi gibi karmaşık problemler üzerinde çalışmıştır. Yöneylem araştırması ve matematik alanına birçok teorik katkıda bulunan Profesör Saaty, giderek karmaşıklaşan modelleme yaklaşımlarının karar problemlerinin çözümünde beklenen etkiyi yapmadığını görmüş ve karmaşık karar problemlerinin çözümünde kullanılmak üzere matematiksel sadeliği sebebiyle kolay anlaşılan ve uygulanan bir teknik geliştirme uğraşına girmiştir. Çalışmalarının sonucunda bugün Analitik Hiyerarşi Prosisi (Analytical Hierarchy Process - AHP) adı ile anılan tekniği geliştirmiştir. AHP tekniği, karar vericilerin çok farklı alanlardaki karar problemlerini yapılandırma ve analiz etme sürecine büyük başarı ile hizmet etmiş ve yoğun olarak uygulaması yapılmıştır.

AHP çoklu kriter içeren kompleks problemleri çözmek için tasarlanmıştır. Süreç, karar vericinin belirlediği her bir kriterin göreceli önemlerini belirlemesine ve daha sonra her bir kritere göre karar alternatifleri arasında seçim yapmasına gerek duyar. AHP, kişileri nasıl karar vermeleri gerektiği konusunda bir yöntem kullanmaya zorunlu kılmak yerine, onlara kendi karar verme sistemlerini tanıma imkanı sağlayarak, daha iyi karar verilmesini sağlayan bir karar verme modelidir.

AHP, karar vericiye, problem için belirlediği her factor veya kriter için karşılaştırma imkanı verirken, belirlemiş olduğu factor veya kriterleri art arda gelen seviyelerde bir hiyerarşik yapı içerisinde sıralamasına da olanak sağlamaktadır.

AHP'nin gerçek yaşamda oldukça geniş bir kullanım alanı mevcuttur. Özellikle ekonomik ve politik kararlar alınırken üst düzey yöneticiler bu yönteme sık sık başvururlar.

Günümüzde çok çeşitli alanlarda uygulanan bu tekniğe ilişkin çok sayıda çalışmaya ulaşılmıştır. 1980 yılında, yayınlanan ilk eser olan "Analitik Hiyerarşi Süreci"nde AHP'nin temelini oluşturan hiyerarşinin tanımı ve hiyerarşik yapılandırma ile ilgili hususlar verilmiştir. 1982 yılında Saaty ve Vargas tarafından yayınlanan eserde AHP, sosyo-psikolojik, teknolojik ve ekonomik problemlerin çözümünde kullanılabilecek bir teori olarak tanımlanmıştır. Saaty tarafından 1987 yılında yayınlanan çalışmada AHP'de kullanılmakta olan sıralama ve bu sıralamanın ortaya çıkışı açıklanmıştır.

Yöntemin Algoritması;

Adım 1: Problemi tanımla ve uygulanacak çözüm yöntemini belirle.

Adım 2: Hiyerarşik yapıyı, genel hedeften alternatiflere veya alternatiflerden genel hedefe doğru oluştur.

Adım 3: İkili karşılaştımalar matrisini her düzey elemanları için oluştur.

Adım 4: Tüm karşılaştırmaları ve matrisleri istenen yapıya göre düzenle.

Adım 5: Problemi çöz ve tutarlılık indeksini control et.

Adım 6: Tüm seviye ve kriterler için 3, 4 ve 5. adımları tekrar et ve problemi çöz.

Tüm ikili karşılaştırmalar yapıldıktan sonra, öz değeri (λ_{max}) kullanılarak tutarlılık hesaplanır. Tutarlılık indeksi (CI), n matris boyutu olmak üzere şu şekilde hesaplanır;

$$CI = (\lambda_{max} - n) / (n - 1)$$

Değerlendirmenin tutarlılık oranı (CR), Tablodaki uygun değer ile CI'nın oranı alınarak hesaplanır.

$$CR = CI / RI$$

Burada RI tesadüfi bir indekstir. Eğer tutarlılık oranı %10'un altında ise kabul edilebilirdir. Eğer fazla ise değerlendirme matrisi tutarsızdır.Tutarsızlığı gidermek için tekrar değerlendirme yapılır.

RI rassal indeks değerleri karşılaştırma matrisinin boyutuna göre aşağıda verilmiştir [Saaty, 1990, 21].

Matrisin Boyutu (n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rassal İndeks(Rİ)	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Hiyerarşik Yapının Oluşturulması;

AHP'de karar vericinin amacı doğrultusunda kriterlerin ve ona ait olan alt kriterlerin belirlenip, hiyerarşik yapının oluşturulması ilk adımdır. AHP'de öncelikle amaç belirlenir ve bu amaç doğrultusunda seçimi etkileyen kriterler ortaya konur. Daha sonra kriterler göz önüne alınarak potansiyel alternatifler belirlenir. Sonuçta karar için hiyerarşik bir yapı oluşturulmuş olur.

Oluşturulan karşılaştırma matrislerinde göreceli öncelikleri ifade etmek üzere Saaty tarafından ortaya atılan aşağıdaki önceliklendirme ölçeği kullanılır.

Tablo 3.1. AHP'de kullanılan ikili karşılaştırma skalası [Saaty, 1990, 54]

Önem Derecesi	**Değerlendirme**
1	Eşit önem (equal importance)
2	Eşit-orta arası önem (equal to moderate importance)
3	Orta derecede önemli (moderate importance of one attribute over another)
4	Orta -güçlü arası önem (moderate to strong importance)
5	Güçlü (Kuvvetli) derecede önemli (strong or essential importance)
6	Güçlü-çok güçlü arası önem (strong to very strong importance)
7	Çok güçlü önem (very strong importance)
8	Çok güçlü-mutlak arası önem (very strong to extreme importance)
9	Mutlak (Kesin) önem (extreme or absolute importance)

Saaty tarafından önerilen Analitik Hiyerarşi prosesi sayısal bir örnekle açıklanmıştır:

Örnek 3.1. [Chen ve Hwang (1992), 336-339]

Doktora derecesini henüz almış bir kişi 3 iş arasından kendisine en uygununu seçmek istemektedir. Bunun için 6 adet kriter belirlemiştir, bu kriterler sırasıyla C_1: araştırma, C_2: gelişim, C_3: fayda, C_4: iş arkadaşlığı, C_5: yerleşim yeri ve C_6: tanınırlık dır.

Kriterler arası ikili karşılaştırma matrisi

	Araştırma	Gelişim	Fayda	İş Ark.	Yerleşim Yeri	Tanınırlık
Araştırma	1	1	1	4	1	1/2
Gelişim	1	1	2	4	1	1/2
Fayda	1	1/2	1	5	3	1/2
İş Ark.	1/4	1/4	1/5	1	1/3	1/3
Yerleşim Yeri	1	1	1/3	3	1	1
Tanınırlık	2	2	2	3	1	1

şeklinde oluşturulmuştur.

Herbir kriter için işlerin ikili karşılaştırma matrisleri ise şöyle oluşmuştur.

Araştırma	A	B	C
A	1	1/4	1/2
B	4	1	3
C	2	1/3	1

Gelişim	A	B	C
A	1	1/4	1/5
B	4	1	1/2
C	5	2	1

Fayda	**A**	**B**	**C**
A	1	3	1/3
B	1/3	1	1
C	3	1	1

İş Ark.	**A**	**B**	C
A	1	1/3	5
B	3	1	7
C	1/5	1/7	1

Yerleşim	**A**	**B**	**C**
A	1	1	7
B	1	1	7
C	1/7	1/7	1

Tanınırlık	**A**	**B**	C
A	1	7	9
B	1/7	1	5
C	1/9	1/5	1

Kriterlerin ikili karşılaştırılmalarının yapıldığı ilk matrisin maksimum özdeğeri;

λ_{max} = 6.35 ve buna karşılık gelen özvektörü:

C_1	C_2	C_3	C_4	C_5	C_6
B_1=[0.16	0.19	0.19	0.05	0.12	0.30 $]^T$

dır.

Ve işlerin kriterlere göre ikili karşılaştırma matrislerine karşılık gelen özdeğerleri ve özvektörleri;

λ_{max}=[3.02	3.02	3.56	3.05	3.0	3.21]

$$B_2 = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \end{array} \\ \begin{bmatrix} 0.14 & 0.10 & 0.32 & 0.28 & 0.47 & 0.77 \\ 0.63 & 0.33 & 0.22 & 0.65 & 0.47 & 0.17 \\ 0.24 & 0.57 & 0.46 & 0.07 & 0.07 & 0.05 \end{bmatrix} \end{array}$$

dır.

Hiyerarşi sonucuna göre alternatiflerin ağırlıklar vektörü;

$$W = B_2 x B_1 = \begin{array}{c} A \\ B \\ C \end{array} \begin{bmatrix} 0.40 \\ 0.34 \\ 0.26 \end{bmatrix}$$

olur.

A, B ve C işlerine karşılık gelen ağırlıklar karşılaştırıldığında A işini tercih etmenin akıllıca bir tercih olacağı açıkça görülmektedir.

Tutarlılık [Kara Harp Okulu, 2003, 9. bölüm, 28-29]:

AHP için önemli başka bir kavram da karar vericinin yargılarında tutarlı olmasıdır. Eğer kriter C_1, kriter C_2 ile karşılaştırıldığında 3 değerini; kriter C_2, kriter C_3 ile karşılaştırıldığında 2 değerini alıyorsa, kriter C_1 kriter C_3 ile karşılaştırıldığında 3 * 2 = 6 değerini almalıdır. Eğer bu karşılaştırmada alınan değerler 4 ve ya 5 ise karşılaştırmada tutarsızlık söz konusudur.

AHP'de ikili karşılaştırmalar için tutarlılık oranı geliştirilmiştir. Bu oran 0.10'dan büyük ise karşılaştırmalar tutarsız, küçük ise karşılaştırmalar tutarlıdır.

Örnek problemimiz için tutarlılık oranı aşağıdaki şekilde elde edilir.

İkili karşılaştırmalar matrisinin ilk sütunu ile ilk kriterin üstünlük değeri, matrisin ikinci stünu ile ikinci kriterin üstünlük değeri çarpılır. Bu işlem matrisin tüm sütunları için yapılır. Bu şekilde ağırlıklı ortalama bulunur. Kriterler için oluşturulan karşılaştırma matrisine göre

	Araştırma	Gelişim	Fayda	İş Ark.	Yerleşim Yeri	Tanınırlık
Araştırma	1	1	1	4	1	1/2
Gelişim	1	1	2	4	1	1/2
Fayda	1	1/2	1	5	3	1/2
İş Ark.	1/4	1/4	1/5	1	1/3	1/3
Yerleşim Yeri	1	1	1/3	3	1	1
Tanınırlık	2	2	2	3	1	1
TOPLAM	**6.25**	**5.75**	**6.533**	**20**	**7.333**	**3.833**

matrisinin sütunları normalize edilirse

	Araştırma	Gelişim	Fayda	İş Ark.	Yerleşim Yeri	Tanınırlık
Araştırma	0.16	0.174	0.153	0.2	0.136	0.13
Gelişim	0.16	0.174	0.306	0.2	0.136	0.13
Fayda	0.16	0.087	0.153	0.25	0.41	0.13
İş Ark.	0.04	0.043	0.031	0.05	0.046	0.088
Yerleşim Yeri	0.16	0.174	0.051	0.15	0.136	0.261
Tanınırlık	0.32	0.348	0.306	0.15	0.136	0.261
TOPLAM	**1**	**1**	**1**	**1**	**1**	**1**

elde edilir. Son olarak kriterlerin üstünlük derecelerini belirlemek için her bir satırın ortalama değeri bulunur.

	Araştırma	Gelişim	Fayda	İş Ark.	Yerleşim Yeri	Tanınırlık	**Üstünlük**
Araştırma	0.16	0.174	0.153	0.2	0.136	0.13	**0.159**
Gelişim	0.16	0.174	0.306	0.2	0.136	0.13	**0.184**
Fayda	0.16	0.087	0.153	0.25	0.41	0.13	**0.198**
İş Ark.	0.04	0.043	0.031	0.05	0.046	0.088	**0.050**
Yerleşim Yeri	0.16	0.174	0.051	0.15	0.136	0.261	**0.155**
Tanınırlık	0.32	0.348	0.306	0.15	0.136	0.261	**0.254**
TOPLAM	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Ağırlıklı ortalama matrisi

$$0.159\begin{bmatrix}1\\1\\1\\1/4\\1\\2\end{bmatrix}+0.184\begin{bmatrix}1\\1\\1/2\\1/4\\1\\2\end{bmatrix}+0.198\begin{bmatrix}1\\2\\1\\1/5\\1/3\\2\end{bmatrix}+0.050\begin{bmatrix}4\\4\\5\\1\\3\\3\end{bmatrix}+0.155\begin{bmatrix}1\\1\\3\\1/3\\1\\1\end{bmatrix}+0.254\begin{bmatrix}1/2\\1/2\\1/2\\1/3\\1\\1\end{bmatrix}=\begin{bmatrix}1.023\\1.221\\1.291\\0.312\\0.968\\1.641\end{bmatrix}$$

bulunur. Ağırlıklı ortalama matrisinin elemanları herbir kriterin ilgili üstünlük değerine bölünür.

Araştırma: $\frac{1.023}{0.159}=6.43$ Fayda: $\frac{1.291}{0.198}=6.52$ Yerleşim Yeri: $\frac{0.968}{0.155}=6.25$

Gelişim: $\frac{1.221}{0.184}=6.64$ İş Ark.: $\frac{0.312}{0.050}=6.24$ Tanınırlık: $\frac{1.641}{0.254}=6.46$

Hesaplanan bu değerlerin aritmetik ortalaması

$$\lambda_{\max}=\frac{(6.43+6.64+6.52+6.24+6.25+6.46)}{6}=6.42$$

bulunur. Buradan tutarlılık indeksi CI = (λ_{max} – n) / (n - 1) = (6.42 - 6) / (6 – 1) = 0.084 olarak elde edilir.

Son olarak tutarlılık oranı CR = CI / RI = 0.084 / 1.24 = 0.0677 olarak bulunur. Bulunan tutarlılık oranı 0.10’dan daha küçük olduğu için karşılaştırma değerlerinin tutarlı olduğu söylenebilir.

4. BULANIK ANALİTİK HİYERARŞİ PROSESİ VE YÖNTEMLERİ

Analitik Hiyerarşi Prosesi 1977-1978 yıllarında ilk olarak Thomas Lorie Saaty tarafından önerilmiştir. Saaty'nin yaklaşımında a_{ij} ikili karşılaştırma oranları $\forall i, j$ için reel sayılardır. Her bir ikili karşılaştırmalar matrisi özvektör metodu kullanılarak çözülmüştür. Aynı zamanda sonuç ağırlıkları ve performans skorları da kesin reel sayılardır. [Chen ve Hwang, 1992, 330]

Saaty'nin AHP metodu 1983 yılında Van Laarhoven ve Pedrycz tarafından bulanık sayılar kullanılarak genişletilmiştir. Van Laarhoven ve Pedrycz alternatifler arasında yaptıkları mukayeselerde dilsel değişkenleri a_{ij} şeklinde üçgensel bulanık sayılar kullanarak ifade etmişlerdir. Ayrıca karar vericilerin aynı alternatifler (nitelikler) çifti üzerinde kendi oranlarını ayrı ayrı bildirme hakkını vermişlerdir. Bu durumda p_{ij} terimi karşılaştırma oranlarını bildiren kişilerin sayılarını belirtmek üzere ikili karşılaştırma oranları a_{ijk}'lar $\left(k = 0,1,..., p_{ij}\right)$ ile ifade edilmiştir. [Chen ve Hwang, 1992, 339]

1984-1985'te yaptığı çalışmalarda Buckley de Saaty'nin AHP metodunu karar vericilerin kendi tercihlerini kesin oranlar yerine bulanık oranlar ile ifade edebildikleri bir duruma genişletmiştir. a_{ij} bulanık oranlar yamuksal bulanık sayılarla verilmiştir. [Chen ve Hwang, 1992, 351]

Chang (1996), bulanık AHP'nin ikili karşılaştırma skalası için üçgensel bulanık sayıların kullanılması ve ikili karşılaştırmaların sentetik derece değerleri için derece analiz yöntemininin kullanılmasını içeren yeni bir yaklaşım ortaya koymaktadır. [Chang, 1996]

Mikhailov (2003), AHP yönteminde ikili karşılaştırma matrislerinden önceliklerin elde edilmesinin AHP'nin en önemli bileşenlerinden biri olduğunu belirterek, öncelik vektörünün bu matrislerden elde edilmesi için en fazla kullanılan özvektör metodu ve logaritmik en küçük kareler yönteminin yerine önceliklendirme sürecinin geometric temsiline dayanan alternatif bir yeni bulanık programlama metodu önermektedir. Bu yöntemde, önceliklendirme problemi standart lineer program şeklinde çözülebilen bulanık programlama problemine dönüşmektedir. Bu yöntemin özellikle karar vericinin tercihleri oldukça tutarsız olduğu durumda diğer yöntemlerden daha iyi sonuç vereceği belirtilmektedir. [Mikhailov, 2003]

Enea ve Piazza (2004), bir çok olası proje alternatifleri arasından bir tanesinin seçimi problemi için AHP yönteminin bulanık uzantısını kullanmaktadırlar. Çalışma tüm elverişli bilgilerin hesaba katılması için bulanık AHP'de düşünülmesi gereken kısıtlar üzerinde yoğunlaşmaktadır. Sunulan yöntemde iki tane yeni algoritma önerilmektedir. [Enea ve Piazza, 2004]

Bulanık AHP yöntemlerinden bazılarının avantaj ve dezavantajları Tablo 4.1.'deki gibidir.

Tablo 4.1. Bulanık AHP metodlarının karşılaştırılması [Büyüközkan, Kahraman ve Ruan, 2004]

Kaynak	**Metodun önemli karakteristikleri**	**Avantaj (+) ve dezavantajları (-)**
Van Laarhoven ve Pedrycz (1983)	• Saaty'nin AHP metodunun üçgen bulanık sayılar kullanılarak uygulanmasıdır	+ Birden fazla karar vericinin düşünceleri karşılıklı (reciprocal) matrislerde modellenebilir. - Küçük bir problem için bile çok fazla matematiksel işlem gerektirir. - Sadece üçgen bulanık sayıların kullanılmasına izin verir.
Buckley (1985)	• Saaty'nin AHP metodunun yamuk bulanık sayılar kullanılarak uygulanmasıdır. • Geometrik ortalama kullanarak bulanık ağırlıkları ve performans skorlarını elde eder.	+ Bulanık duruma genişletmek kolaydır + Tek bir sonucu garanti eder - Hesap gereksinimi çok fazladır
Boender et al. (1989)	• Van Laarhoven ve Pedrycz'in metodunun biraz geliştirilmişidir. • Yerel önceliklerin normalizasyonu için daha sağlam bir yaklaşım sunar	+ Birden fazla karar vericinin düşünceleri modellenebilir -Hesap gereksinimi çok fazladır
Chang (1996)	• Sentetik derece değerleri • Seviye basit sıralaması • Karma toplam sıralama	+ Hesap gereksinimi daha azdır +Klasik AHP'nin adımlarını izler. İlave işlem gerektirmez. - Sadece üçgen bulanık sayılar kullanılabilir
Cheng (1996)	• Bulanık standartlar oluşturur • Performans skorlarını üyelik fonksiyonları ile ifade eder • Toplam agırlıkları hesaplamak için entropi kavramlarını kullanır	+ Çok fazla hesap gerektirmez - Olasılık dağılımı bilindiğinde entropi kullanılır. Metod hem olasılık hem de olabilirlik ölçülerine dayanır.

Şimdi Bulanık AHP yöntemlerinden Van Laarhoven ve Pedrycz (1983), Buckley (1985), Chang (1996), Mikhailov (2003), Enea ve Piazza (2004)'nın yaklaşımlarını inceleyerek örneklerle açıklayalım.

4.1. Laarhoven ve Pedrycz Yaklaşımı [Chen ve Hwang, 1992, 339-351]

Laarhoven ve Pedrycz, Saaty'nin AHP'sinin kesin genişlemesi olan bir algoritma önermişlerdir. AHP metodunda bulanıklılığın bulanık notasyon ile ifade edilmediğini, ancak bir karşılaştırma matrisi oluşturularak karar probleminin dolaylı bir şekilde modellendirildiğini hatırlayalım. AHP'nin genişletilmiş olan bu versiyonunda karşılaştırma matrisinde yer alan elemanlar üçgensel bulanık sayılar ile ifade edilmiştir. Hesaplama adımları AHP'dekilerle aynıdır. Bulanık ağırlıklar ve bulanık performans ağırlıklarını elde etmek için Lootsma'nın logaritmik en küçük kareler metodunu kullanmışlardır. Bulanık faydaların hesaplanmasında bulanık üçgensel sayılar için aritmetik işlemler kullanılmıştır. Çoklu karar vericilerin görüşleri aynı zamanda karşılaştırma matrislerinde modellebilir. Bu metod aşağıda ayrıntıları ile işlenmiştir.

Lootsma'nın Logaritmik En Küçük Kareler Metodu:

Bu metodun tahmin edilen ağırlıkları belirlemek için seçilmesinin sebebi, birden çok karar vericinin fikirlerini ele almak için uygun olması ve bulanık durumlara kolayca genelleştirilebilmesidir.

A pozitif karşılaştırma (reciproal) matrisi aşağıdaki şekilde gösterilsin,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

burada a_{ij}' ler reel sayılardır ve i. alternatifin *j.* kritere göre aldığı değerlendirme skorunu gösterir.

$$\sum_{i<j} \left(\ln a_{ij} - \ln(w_i / w_j) \right)^2$$

toplamını minimize edilerek, tahmin edilen $\underline{w} = (w_1, w_2, \ldots, w_n)$ ağırlıklar vektörü elde edilir.

Eğer birden çok karar verici var ise

$$\sum_{i<j}\sum_{k=1}^{P_{ij}}\left(\ln a_{ijk} - \ln(w_i / w_j)\right)^2$$

toplamını minimize edilerek, tahmin edilen $\underline{w}$ ağırlık vektörü elde edilir.

Burada a_{ijk}' lar , $k = 1,2,..., p_{ij}$ w_i / w_j için p_{ij} tahminleridir. P_{ij} değerleri hiçbir karar verici tahminde bulunmadığı durumda 0, tek bir karar vericinin tahminde bulunduğu durumlarda 1 yada daha çok karar vericinin olması durumunda 1'den büyük olacağını belirtmeliyiz.

Eğer $y_{ijk} = ln\ a_{ijk}$, $z_i = ln\ w_i$ *ve* $z_j = ln\ w_j$ olarak alınırsa

$$z_i \sum_{\substack{j=1 \\ j\neq i}}^{n} p_{ij} - \sum_{\substack{j=1 \\ j\neq i}}^{n} p_{ij} z_j = \sum_{\substack{j=1 \\ j\neq i}}^{n}\sum_{k=1}^{P_{ij}} y_{ijk} \quad ; \qquad i = 1,...,n$$

z_i için ortak normal eşitliklerini çözerek

$$\sum_{i<j}\sum_{k=1}^{P_{ij}}\left(y_{ijk} - z_i + z_j)\right)^2$$

ifadesini minimize edebiliriz.

Buradan z_i'lerin üstelini alarak ve normalize ederek $\underline{w} = (w_1,w_2,...,w_n)$ ağırlıklar vektörünü elde edebiliriz.

Algoritma:

<u>**Adım 1:**</u> Karar vericilere danışarak, aşağıdaki biçimde verilen *n* + 1 boyutlu bulanık karşılaştırma matrisleri elde edilir;

$$D=\begin{bmatrix} (1,1,1) & \begin{matrix} a_{121} \\ a_{122} \\ \vdots \\ a_{12p_{12}} \end{matrix} & \cdots & \begin{matrix} a_{1n1} \\ a_{1n2} \\ \vdots \\ a_{1np_{1n}} \end{matrix} \\ \begin{matrix} a_{211} \\ a_{212} \\ \vdots \\ a_{21p_{21}} \end{matrix} & (1,1,1) & \cdots & \begin{matrix} a_{2n1} \\ a_{2n2} \\ \vdots \\ a_{2np_{2n}} \end{matrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{matrix} a_{n11} \\ a_{n12} \\ \vdots \\ a_{n1p_{n1}} \end{matrix} & \begin{matrix} a_{n21} \\ a_{n22} \\ \vdots \\ a_{n2p_{n2}} \end{matrix} & \cdots & (1,1,1) \end{bmatrix}$$

Burada a_{ijPij}'ler çok sayıda karar verici tarafından tahmin edilen bulanık oranlardır. P_{ij} tahminde bulunan karar verici sayısına bağlı olarak 0, 1 ya da 1 den daha büyük değerler alabilir.

Adım 2: $z_i = (l_i, m_i, u_i)$ olsun. Aşağıdaki lineer denklemler çözülür,

$$l_i\left(\sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}\right) - \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}u_j = \sum_{\substack{j=1 \\ j\neq i}}^{n}\sum_{k=1}^{P_{ij}}\left(\ln l_{ijk}\right) \ , \ \forall i \tag{4.1}$$

$$m_i\left(\sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}\right) - \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}m_j = \sum_{\substack{j=1 \\ j\neq i}}^{n}\sum_{k=1}^{P_{ij}}\left(\ln m_{ijk}\right) \ , \ \forall i \tag{4.2}$$

$$u_i\left(\sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}\right) - \sum_{\substack{j=1 \\ j\neq i}}^{n} P_{ij}l_j = \sum_{\substack{j=1 \\ j\neq i}}^{n}\sum_{k=1}^{P_{ij}}\left(\ln u_{ijk}\right) \ , \ \forall i \tag{4.3}$$

$1n(l_{ijk})$ ve $1n(u_{ijk})$ ifadeleri $1n(a_{ijk}) = -1n(a_{jik})$'nın alt ve üst değerleri olmak üzere aşağıdaki eşitlik sağlanır.

$$1n\ (l_{ijk}) + 1n\ (l_{jik}) = 1n\ (u_{ijk}) + 1n\ (u_{jik}) = 0\ \forall i, j, k\ .$$

(4.1), (4.2) ve (4.3) deklemleri lineer bağımlıdır. Genel olarak bu denklemler için t_1 ve t_2'nin keyfi olarak seçilebileceği

$$z_i = (l_i+t_1, m_i+t_2, u_i+t_1)\ \forall\ i \tag{4.4}$$

çözümü verilebilir.

__Adım 3:__ Yukarıda lineer sistemde, tüm eşitliklerin sağ yanlarında logaritma alındığına dikkat edilmelidir. Dolayısıyla şimdi bulanık w_i ağırlıklarını hesaplayabilmek için l_i, m_i ve u_i 'lerin üstelleri alınmalıdır.

$$w_i = (\lambda_1 \exp(l_i), \lambda_2 \exp(m_i), \lambda_3 \exp(u_i)) \tag{4.5}$$

Burada

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

dir.

(4.5) eşitliği aynı zamanda r_{ij} performans skorlarını belirlemede kullanılabilir.

__Adım 4:__ Tüm karşılaştırma (reciprocal) matrisleri çözünceye kadar Adım l' den Adım 3' e kadar yapılacak işlemler tekrarlanır. Bulanık ağırlıklar ve performans skorları ile A_i alternatifi için bulanık fayda

$$U_i = \sum_{j=1}^{n} w_i r_{ij} \quad , \forall i, j$$

formülü ile hesaplanır.

Örnek 4.1. [Laarboven ve Pedrycz, 1983]

Bir üniversitede açılan bir doçentlik kadrosuna başvuran 3 aday vardır. Bu adayları değerlendirmek için C_1: Matematiksel Yaratıcılık, C_2: Bilimsellik, C_3: Araştırma Yeteneği ve C_4: İnsani İlişkiler kriterleri ele alınmıştır. Adayı seçecek komitede 3 karar verici bulunmaktadır. Problemin çözümü için aşağıdaki adımları sırasıyla uygulayalım.

Adım 1: Karar vericilerden, alternatifler arasında karşılaştırmalı önem ağırlıkları alınarak elde edilen matrisler Tablo 4.2.- 4.6.'da verilmiştir. Bu matrislerde bazı elemanların olmadığı görülür bunun nedeni karar vericilerin bu karşılaştırmaları yapmamış olmasıdır (missing value). Bu karşılaştırma matrislerindeki elemanlar bulanık sayılardır. Örneğin $a_{ij} = \left(\frac{2}{3}, 1, \frac{3}{2}\right)$ olması i. alternatifin j. alternatife göre hemen hemen eşit önemli olduğu ya da $a_{ij} = \left(\frac{3}{2}, 2, \frac{5}{2}\right)$ gibi bir karşılaştırma i. alternatifın j. alternatife göre hemen hemen iki kat önemli olduğu anlamını taşır.

Tablo 4.2. Matematiksel Yaratıcılık Kriterine Göre İkili Karşılaştırmalar Matrisi

C_1	A_1	A_2	A_3
A_1	(1,1,1)	(2/3,1,3/2) (2/3,1,3/2)	(2/3,1,3/2) (2/5,1/2,2/3)
A_2	(2/3,1,3/2) (2/3,1,3/2)	(1,1,1)	(2/5,1/2,2/3)
A_3	(2/3,1,3/2) (3/2,2,5/2)	(3/2,2,5/2)	(1,1,1)

Tablo 4.3. Bilimsellik Kriterine Göre İkili Karşılaştırmalar Matrisi

C_2	A_1	A_2	A_3
A_1	(1,1,1)	(5/2,3,7/2)	(3/2,2,5/2)
A_2	(2/7,1/3,2/5)	(1,1,1)	-
A_3	(2/5,1/2,2/3)	-	(1,1,1)

Tablo 4.4. Araştırma Yeteneği Kriterine Göre İkili Karşılaştırmalar Matrisi

C_3	A_1	A_2	A_3
A_1	(1,1,1)	(5/2,3,7/2) (5/2,3,7/2) (3/2,2,5/2)	(5/2,3,7/2)
A_2	(2/7,1/3,2/5) (2/7,1/3,2/5) (2/5,1/2,2/3)	(1,1,1)	(2/3,1,3/2)
A_3	(2/7,1/3,2/5)	(2/3,1,3/2)	(1,1,1)

Tablo 4.5. İnsani İlişkiler Kriterine Göre İkili Karşılaştırmalar Matrisi

C_4	A_1	A_2	A_3
A_1	(1,1,1)	-	(3/2,2,5/2) (2/5,1/2,2/3)
A_2	-	(1,1,1)	(3/2,2,5/2)
A_3	(2/5,1/2,2/3) (3/2,2,5/2)	(2/5,1/2,2/3)	(1,1,1)

Tablo 4.6. Kriterler Arasında İkili Karşılaştırmalar Matrisi

	C_1	C_2	C_3	C_4
C_1	(1,1,1)	(2/3,1,3/2) (2/5,1/2,2/3) (3/2,2,5/2)	(2/3,1,3/2)	(2/7,1/3,2/5) (2/7,1/3,2/5) (2/5,1/2,2/3)
C_2	(2/3,1,3/2) (3/2,2,5/2) (2/5,1/2,2/3)	(1,1,1)	(5/2,3,7/2) (5/2,3,7/2)	(2/3,1,3/2) (2/3,1,3/2) (3/2,2,5/2)
C_3	(2/3,1,3/2)	(2/7,1/3,2/5) (2/7,1/3,2/5)	(1,1,1)	(2/5,1/2,2/3)
C_4	(5/2,3,7/2) (5/2,3,7/2) (3/2,2,5/2)	(2/3,1,3/2) (2/3,1,3/2) (2/5,1/2,2/3)	(3/2,2,5/2)	(1,1,1)

Adım 2: Her bir matris için aşağıdaki lineer denklem sistemi kurulur.

$$l_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} u_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln\left(l_{1jk}\right)$$

$$l_2 \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} u_j = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln\left(l_{2jk}\right)$$

$$l_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} u_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln\left(l_{3jk}\right)$$

$$m_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} m_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln\left(m_{1jk}\right)$$

$$m_2 \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} m_j = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln\left(m_{2jk}\right)$$

$$m_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} m_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln\left(m_{3jk}\right)$$

$$u_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} l_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln\left(u_{1jk}\right)$$

$$u_2 \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} l_j = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln\left(u_{2jk}\right)$$

$$u_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} l_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln\left(u_{3jk}\right)$$

Bu lineer denklemler “Matematiksel Yaratıcılık Kriteri”ne göre kurulan ikili karşılaştırmalar matrisine uygulanırsa.

$$l_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} u_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln\left(l_{1jk}\right) \quad \Rightarrow \quad l_1 (P_{12} + P_{13}) - P_{12} u_2 - P_{13} u_3 = \sum_{j=2}^{3} \sum_{k=1}^{2} \ln(l_{1jk})$$

$$\Rightarrow \quad l_1 (2+2) - 2u_2 - 2u_3 = \sum_{j=2}^{3} \ln(l_{1j1}.l_{1j2})$$

$$\Rightarrow \quad 4l_1 - 2u_2 - 2u_3 = \ln(l_{121}.l_{122}.l_{131}.l_{132})$$

$$\Rightarrow \quad 4l_1 - 2u_2 - 2u_3 = \ln(\frac{2}{3}.\frac{2}{3}.\frac{2}{3}.\frac{2}{5})$$

$$\Rightarrow \quad 4l_1 - 2u_2 - 2u_3 = \ln(\frac{16}{135})$$

$$\Rightarrow \quad 4l_1 - 2u_2 - 2u_3 = -2.133$$

$$l_2 \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j \neq 2}}^{3} P_{2j} u_j = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln\left(l_{2jk}\right) \quad \Rightarrow \quad l_2 (P_{21} + P_{23}) - P_{21} u_1 - P_{23} u_3 = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \sum_{k=1}^{2} \ln(l_{2jk})$$

$$\Rightarrow \quad l_2 (2+1) - 2u_1 - 1u_3 = \sum_{\substack{j=1 \\ j \neq 2}}^{3} \ln(l_{2j1}.l_{2j2})$$

$$\Rightarrow \quad 3l_2 - 2u_1 - u_3 = \ln(l_{211}.l_{212}.l_{231})$$

$$\Rightarrow \quad 3l_2 - 2u_1 - u_3 = \ln(\frac{2}{3}.\frac{2}{3}.\frac{2}{5})$$

$$\Rightarrow \quad 3l_2 - 2u_1 - u_3 = \ln(\frac{8}{45})$$

$$\Rightarrow \quad 3l_2 - 2u_1 - u_3 = -1.727$$

$$l_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} u_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln\left(l_{3jk}\right) \quad \Rightarrow \quad l_3 (P_{31} + P_{32}) - P_{31} u_1 - P_{32} u_2 = \sum_{j=1}^{2} \sum_{k=1}^{2} \ln(l_{3jk})$$

$$\Rightarrow \quad l_3 (2+1) - 2u_1 - 1u_2 = \sum_{j=1}^{2} \ln(l_{3j1}.l_{3j2})$$

$$\Rightarrow \quad 3l_3 - 2u_1 - u_2 = \ln(l_{311}.l_{312}.l_{321})$$

$$\Rightarrow \quad 3l_3 - 2u_1 - u_2 = \ln(\frac{2}{3}.\frac{3}{2}.\frac{3}{2})$$

$$\Rightarrow \quad 3l_3 - 2u_1 - u_2 = \ln(\frac{3}{2})$$

$$\Rightarrow \quad 3l_3 - 2u_1 - u_2 = 0.405$$

$$m_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} m_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln(m_{1jk}) \qquad \Rightarrow \qquad 4m_1 - 2m_2 - 2m_3 = -0.693$$

$$m_2 \sum_{\substack{j=1 \\ j\neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j\neq 2}}^{3} P_{2j} m_j = \sum_{\substack{j=1 \\ j\neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln(m_{2jk}) \qquad \Rightarrow \qquad 3m_2 - 2m_1 - m_3 = -0.693$$

$$m_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} m_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln(m_{3jk}) \qquad \Rightarrow \qquad 3m_3 - 2m_1 - m_2 = 1.386$$

$$u_1 \sum_{j=2}^{3} P_{1j} - \sum_{j=2}^{3} P_{1j} l_j = \sum_{j=2}^{3} \sum_{k=1}^{P_{ij}} \ln(u_{1jk}) \qquad \Rightarrow \qquad 4u_1 - 2l_2 - 2l_3 = 0.811$$

$$u_2 \sum_{\substack{j=1 \\ j\neq 2}}^{3} P_{2j} - \sum_{\substack{j=1 \\ j\neq 2}}^{3} P_{2j} l_j = \sum_{\substack{j=1 \\ j\neq 2}}^{3} \sum_{k=1}^{P_{ij}} \ln(u_{2jk}) \qquad \Rightarrow \qquad 3u_2 - 2l_1 - l_3 = 0.405$$

$$u_3 \sum_{j=1}^{2} P_{3j} - \sum_{j=1}^{2} P_{3j} l_j = \sum_{j=1}^{2} \sum_{k=1}^{P_{ij}} \ln(u_{3jk}) \qquad \Rightarrow \qquad 3u_3 - 2l_1 - l_2 = 2.238$$

Bulduğumuz bu denklemleri sırasıyla alt alta yazarsak

$$\begin{aligned}
&4l_1 - 2u_2 - 2u_3 = -2.133\\
&3l_2 - 2u_1 - u_3 = -1.727\\
&3l_3 - 2u_1 - u_2 = 0.405\\
&4m_1 - 2m_2 - 2m_3 = -0.693\\
&3m_2 - 2m_1 - m_3 = -0.693\\
&3m_3 - 2m_1 - m_2 = 1.386\\
&4u_1 - 2l_2 - 2l_3 = 0.811\\
&3u_2 - 2l_1 - l_3 = 0.405\\
&3u_3 - 2l_1 - l_2 = 2.238
\end{aligned}$$

lineer denklem sistemi elde edilir.

Şimdi bu denklem sistemini çözebilmek için katsayılar matrisini oluşturalım.

$$\begin{array}{c}
\begin{array}{ccccccccc} l_1 & l_2 & l_3 & m_1 & m_2 & m_3 & u_1 & u_2 & u_3 \end{array} \\
\left[\begin{array}{ccccccccc|c}
4 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -2 & -2.133 \\
0 & 3 & 0 & 0 & 0 & 0 & -2 & 0 & -1 & -1.727 \\
0 & 0 & 3 & 0 & 0 & 0 & -2 & -1 & 0 & 0.405 \\
0 & 0 & 0 & 4 & -2 & -2 & 0 & 0 & 0 & -0.693 \\
0 & 0 & 0 & -2 & 3 & -1 & 0 & 0 & 0 & -0.693 \\
0 & 0 & 0 & -2 & -1 & 3 & 0 & 0 & 0 & 1.386 \\
0 & -2 & -2 & 0 & 0 & 0 & 4 & 0 & 0 & 0.811 \\
-2 & 0 & -1 & 0 & 0 & 0 & 0 & 3 & 0 & 0.405 \\
-2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 2.238
\end{array}\right]
\end{array}$$

oluşturduğumuz bu katsayılar matrisinde m_i (i=1,2,3) değerlerinin katsayılarından oluşan sistemin l_i ve u_i'leri içermemesinden dolayı m_i değerlerini bağımsız olarak da çözebiliriz.

$$\begin{array}{c}
\begin{array}{ccc} m_1 & m_2 & m_3 \end{array} \\
\left[\begin{array}{ccc|c}
4 & -2 & -2 & -0.693 \\
-2 & 3 & -1 & -0.693 \\
-2 & -1 & 3 & 1.386
\end{array}\right]
\end{array}$$

m_i'ler için oluşturduğumuz lineer denklem sisteminin determinantı "0" olduğundan sistemi çözebilmek için m_2'yi keyfi olarak "0" alalım. Bu durumda

$2m_1 + m_3 = 0.693$

$-2m_1 + 3m_3 = 1.386$

dır. Buradan

$m_1 = 0.086625$

$m_2 = 0$

$m_3 = 0.51975$

olarak bulunur.

Şimdi de l_i ve u_i'ler için çözüm matrisini oluşturalım.

$$\begin{array}{cccccccc} l_1 & l_2 & l_3 & u_1 & u_2 & u_3 & & \\ \end{array}$$

$$\left[\begin{array}{cccccccc} 4 & 0 & 0 & 0 & -2 & -2 & \vdots & -2.133 \\ 0 & 3 & 0 & -2 & 0 & -1 & \vdots & -1.727 \\ 0 & 0 & 3 & -2 & -1 & 0 & \vdots & 0.405 \\ 0 & -2 & -2 & 4 & 0 & 0 & \vdots & 0.811 \\ -2 & 0 & -1 & 0 & 3 & 0 & \vdots & 0.405 \\ -2 & -1 & 0 & 0 & 0 & 3 & \vdots & 2.238 \end{array}\right]$$

l_i ve u_i'ler için oluşturduğumuz lineer denklem sisteminin de determinantı “0” olduğundan sistemi çözebilmek için l_1'i keyfi olarak “0” alalım. Bu durumda

$3l_2-2u_1-u_3 = -1.727$

$3l_3-2u_1-u_2 = 0.405$

$-2l_2-2l_3+4u_1 = 0.811$

$-l_3+3u_2 = 0.405$

$-l_2+u_3 = 2.238$

dır. Buradan

$l_1 = 0$

$l_2 = -0.0077$

$l_3 = 0.5627$

$u_1 = 0.48025$

$u_2 = 0.32256$

$u_3 = 0.7434$

olarak bulunur.

Bulduğumuz tüm değerleri aşağıdaki tabloda özetleyerek verelim.

Matematiksel Yaratıcılık Kriterine Göre l_i , m_i ve u_i değerleri

i	l_i	m_i	u_i
1	0	0.087	0.481
2	-0.007	0	0.323
3	0.563	0.520	0.744

"Matematiksel Yaratıcılık Kriteri"ne göre uygulanan bu işlemler diğer kriterlere ve Kriterler Arasında İkili Karşılaştırmalar Matrisine de uygulanırsa aşağıdaki sonuçlar elde edilir.

Bilimsellik Kriterine Göre l_i , m_i ve u_i değerleri

i	l_i	m_i	u_i
1	0	0.8283	-0.0005
2	-0.9706	0	-0.63416
3	-1.1993	-0.1353	-0.68783

Araştırma Yeteneği Kriterine Göre l_i , m_i ve u_i değerleri

i	l_i	m_i	u_i
1	0	0.9827	-0.03916
2	-1.2	0	-0.76616
3	-1.2316	-0.05814	-0.8555

İnsani İlişkiler Kriterine Göre l_i , m_i ve u_i değerleri

i	l_i	m_i	u_i
1	0	-0.693	0.5109
2	0.6609	0	1.1717
3	0.2555	-0.693	0.2554

Kriterler Arasında İkili Karşılaştırmalar Göre l_i , m_i ve u_i değerleri

i	l_i	m_i	u_i
1	0	-0.4955	0.2085
2	0.3664	0	0.7649
3	-0.30625	-0.8258	-0.1430
4	0.5277	0.0826	0.8639

Adım 3: l_i, *mi ve* u_i (i = 1, 2, 3)'lerin üstelleri alınırsa.

Matematiksel Yaratıcılık Kriterine Göre

i	exp(l_i)	exp (m_i)	exp (u_i)
1	1.000	1.091	1.618
2	0.993	1.000	1.381
3	1.757	1.682	2.104

Bilimsellik Kriterine Göre

i	exp(l_i)	exp (m_i)	exp (u_i)
1	1.000	2.2894	0.9995
2	0.3788	1.000	0.5304
3	0.3014	0.8734	0.5027

Araştırma Yeteneği Kriterine Göre

i	exp(l_i)	exp (m_i)	exp (u_i)
1	1.000	2.6716	0.9616
2	0.3012	1.000	0.4648
3	0.2918	0.9435	0.4251

İnsani İlişkiler Kriterine Göre

i	exp(l_i)	exp (m_i)	exp (u_i)
1	1.000	0.5001	1.6668
2	1.9365	1.000	3.2275
3	1.2911	0.5001	1.2910

Kriterler Arasında İkili Karşılaştırmalar Göre

i	exp(l_i)	exp (m_i)	exp (u_i)
1	1.000	0.6093	1.2318
2	1.4425	1.000	2.1488
3	0.7362	0.4379	0.8667
4	1.6950	1.0861	2.3724

değerleri elde edilir. Bu hesaplanan değerler yardımıyla performans skorlarını hesaplarız.

Örneğin Matematiksel Yaratıcılık Kriteri'ne göre elde edilen üstel sayıları kullanarak (4.5) denkleminden bulanık performans skorlarını hesaplayalım.

$$r_{11} = (\lambda_1 \exp(l_1), \lambda_2 \exp(m_1), \lambda_3 \exp(u_1) = (0.1959, 0.2890, 0.4310)$$

$$r_{12} = (\lambda_1 \exp(l_2), \lambda_2 \exp(m_2), \lambda_3 \exp(u_2) = (0.1947, 0.2650, 0.3680)$$

$$r_{13} = (\lambda_1 \exp(l_3), \lambda_2 \exp(m_3), \lambda_3 \exp(u_3) = (0.3441, 0.4457, 0.5607)$$

Burada $\lambda_1 = \left(\sum_{i=1}^{N} \exp(u_i) \right)^{-1} = (1.6181 + 1.3810 + 2.1040)^{-1} = 0.1959$

ve benzer şekilde

$$\lambda_2 = \left(\sum_{i=1}^{N} \exp(m_i) \right)^{-1} = 0.2650$$

$$\lambda_3 = \left(\sum_{i=1}^{N} \exp(l_i) \right)^{-1} = 0.2665$$

olarak hesaplanır.

Aynı şekilde diğer kriterler için de devam edilirse r_{ij} performans değerleri *i = 1,2,3,4* ve *j = 1,2,3* için aşağıdaki şekilde bulunur.

$$\begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & C_3 & C_4 \end{array} \\ \begin{bmatrix} (0.196,0.286,0.431) & (0.405,0.546, 0.714) & (0.540,0.579,0.603) & (0.162,0.250,0.394) \\ (0.195,0.265,0.368) & (0.162,0.182,0.204) & (0.163,0.217,0.292) & (0.313,0.500,0.763) \\ (0.344,0.446,0.561) & (0.277,0.273,0.340) & (0.158,0.205,0.267) & (0.209,0.250,0.305) \end{bmatrix} \end{array}$$

Dört kriter için uyguladığımız tüm adımları kriterlerin karşılaştırma matrisine de uygulayarak

$\underline{w}$=[(0.149,0.194,0.256), (0.235,0.319,0.431), (0.112,0.140,0.180), (0.263,0.347,0.451)

bulanık ağırlıklar vektörünü elde edebiliriz.

<u>Adım 4</u>: Bu verilerden yararlanarak, alternatiflerden elde edilen bulanık faydalar;

$$U_i = \sum_{j=1}^{n} w_i r_{ij}$$

formülü yardımıyla

$$U_1 = \sum_{j=1}^{4} w_j r_{1j} = w_1.r_{11} + w_2.r_{12} + w_3.r_{13} + w_4.r_{14}$$

$$= (0.149, 0.194, 0.256).(0.196, 0.286, 0.431) + (0.235, 0.319, 0.431).(0.405, 0.546, 0.714) + (0.112, 0.140, 0.180).(0.540, 0.579, 0.603) + (0.263, 0.347, 0.451).(0.162, 0.250, 0.394)$$

$$= (0.227, 0.398, 0.705)$$

olarak hesaplanır. Benzer şekilde U_2 ve U_3 değerlerini de hesaplayarak aşağıdaki tabloyu elde ederiz.

U_1	U_2	U_3
(0.227,0.398, 0.705)	(0.168,0.313,0.579)	(0.188,0.289,0.504)

4.2. Buckley Yaklaşımı [Chen ve Hwang, 1992, 351-363]

Buckley 1985 yılında Saaty'nin Analitik Hiyerarşi Prosesinin genişleterek a_{ij} bulanık karşılaştırma oranları üzerinde çalışmıştır. Buckley, Laarhoven ve Pedrycz' in metodunu iki yönden eleştirmiştir. Bunlardan ilki Laarhoven ve Pedrycz'in metodunda yer alan lineer denklemlerin her zaman tek çözümünün olmaması, ikincisi de ağırlıkların bulunmasında üçgensel bulanık sayıların kullanılmasında ısrar etmeleridir.

Bunun üzerine Buckley, bulanık ağırlıkları ve performans skorlarını elde edebilmek için geometrik ortalama metodunu kullanmıştır. Bu metodun kullanılmasının nedeni bulanık durumlara kolayca genelleştirilebilmesi ve karşılaştırma matrislerinden tek çözüm elde edilmesini garantilemesidir. Ayrıca Buckley karar vericilerin karşılaştırma oranlarını gösterebilmek için üçgensel bulanık sayılar yerine yamuksal *(a,b,c,d)* bulanık sayıları kullanmıştır.

Bu metod aşağıda ayrıntıları ile açıklanmıştır.

Geometrik Ortalama Metodu

Bu metod ağırlıkları elde etmede kolaylık sağlar ve bulanık durumlarda kolayca uygulanabilir.

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

bulanık pozitif karşılaştırma matrisi olsun.

Her bir satırın geometrik ortalaması

$$z_i = \left[\prod_{j=1}^{n} a_{ij} \right]^{1/n}, i = 1,2,...,n$$

formülüyle hesaplanır.

w_i ağırlıkları da $w_i = z_i / \{z_1 + ... + z_n)$, $\forall\ i$, formülü ile hesaplanır.

Algoritma:

Bu algoritma hem tek karar verici için hem de birden çok karar verici için kullanılabilir. Tek karar verici için geçerli durum aşağıda açıklanmıştır.

Adım 1: Elemanları $\overline{a}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij}), \quad \forall i, j$ şeklinde yamuksal bulanık sayılar olan A ikili karşılaştırmalar matrisi karar vericilerden elde edilir.

Adım 2: w_i ağırlıkları aşağıdaki şekilde hesaplanabilir.

Her bir satır için geometrik ortalama $z_i = (\overline{a}_{i1} \otimes ... \otimes \overline{a}_{in})^{1/n} \ \forall i$ formülü ile bulunur, burada "$\otimes$" işareti bulanık çarpma işlemini ifade etmektedir.

Bulanık ağırlık, $w_i = z_i \otimes (z_1 \otimes ... \otimes z_n)^{-1}$ şeklinde hesaplanır, burada "$\otimes$" işareti bulanık toplama işlemini ifade etmektedir.

Bulanık ağırlıkları elde etmek için aşağıdaki ayrıntıları verelim. $\bar{a}_{ij}$ yamuksal bulanık sayılarının güven aralıkları cinsinden sağ ve sol yan geometrik ortalamaları aşağıdaki şekilde tanımlansın;

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

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

dır. Ayrıca;

$$a_{ij} = \left[\prod_{j=1}^{n} \left(a_{ij} \right) \right]^{1/n}$$

ve

$$a = \sum_{i=1}^{m} a_i$$

olsun.

Benzer şekilde b_i ve b, c_i ve c, d_i ve d'leri de tanımlayabiliriz.

Bulanık ağırlık $w_i = \left(\frac{a_i}{d}, \frac{b_i}{c}, \frac{c_i}{b}, \frac{d_i}{a} \right)$ şeklinde hesaplanır.

x yatay eksen üzerinde reel bir sayı olsun. $\mu_{w_i}(x)$ üyelik fonksiyonu , aşağıdaki gibi tanımlanır.

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

$x \in [a_i / d, b_i / c]$ olduğunda $x = f_i(\alpha) / g(\alpha)$ eşitliğinden elde edilir.

Benzer şekilde $x \in [c_i / b, d_i / a]$ olduğunda da $x = g_i(\alpha) / f(\alpha)$ denkleminden bulunur.

Burada

$$f(\alpha) = \sum_{i=1}^{m} f_i(\alpha) \ ve \ g(\alpha) = \sum_{i=1}^{m} g_i(\alpha)$$

dir. Tüm r_{ij} , $\forall\, i$, j performans skorları elde edilinceye kadar Adım 2 tekrarlanır.

Adım 3: Bulanık ağırlıklar ve bulanık performans skorları, bulanık çok kriterli karar verme problemi olarak birleştirilir.

Bulanık faydalar $U_i = \sum_{j=1}^{n} w_j r_{ij}$, $\forall i.$ ile hesaplanır.

Örnek 4.2. [Buckley, 1985]

Bir şirket A_1, A_2 ve A_3 kimyasal maddelerini çevreye zararlarına göre en zararsızdan en zararlıya doğru sıralamak istemektedir. Bunu yaparken 3 kriteri göz önünde bulunduracaktır. Bunlar C_1: Suda yaşayan canlılara etki, C_2: Tarımsal ürünlere etki, C_3: Orman ürünlerine etki.

<u>Adım 1</u>: Uzman tüm nitelikler için her bir alternatifin göreceli ağırlık oranlarını ve bunun gibi niteliklerin kendi aralarındaki göreceli ağırlık oranlarını tayin eder. Bu oranlar aşağıdaki karşılaştırma matrislerinde verilmiştir.

Suda Yaşayan Canlılara Etki Kriterine Göre İkili Karşılaştırmalar Matrisi

C_1	A_1	A_2	A_3
A_1	(1, 1, 1, 1)	(1/4, 1/3, 1/3, 1/2)	(1/2, 1/2, 1/2, 1/2)
A_2	(2,3,3,4)	(1, 1, 1, 1)	(1,1,2,2)
A_3	(2,2,2,2)	(1/2, 1/2, 1, 1)	(1, 1, 1, 1)

Tarımsal Ürünlere Etkisi Kriterine Göre İkili Karşılaştırmalar Matrisi

C_2	A_1	A_2	A_3
A_1	(1, 1, 1, 1)	(6,6,6,7)	(2,2,4,4)
A_2	(1/7, 1/6, 1/6, 1/6)	(1, 1, 1, 1)	(1/2, 1/2, 1, 1)
A_3	(1/4, 1/4, 1/2, 1/2)	(1,1,2,2)	(1, 1, 1, 1)

Orman Ürünlerine Etkisi Kriterine Göre İkili Karşılaştırmalar Matrisi

C_3	A_1	A_2	A_3
A_1	(1, 1, 1, 1)	(1,2,2,3)	(7,8,8,8)
A_2	(1/3, 1/2, 1/2, 1)	(1, 1, 1, 1)	(3,3,4,4)
A_3	(1/8, 1/8, 1/8, 1/7)	(1/4, 1/4, 1/3, 1/3)	(1, 1, 1, 1)

Kriterler Arası İkili Karşılaştırmalar Matrisi

	C_1	C_2	C_3
C_1	(1, 1, 1, 1)	(1/7, 1/6, 1/6, 1/5)	(1/3, 1/2, 1/2, 1)
C_2	(5,6,6,7)	(1, 1, 1, 1)	(3,3,3,3)
C_3	(1,2,2,3)	(1/3, 1/3, 1/3, 1/3)	(1, 1, 1, 1)

<u>Adım 2</u>: Birinci karşılaştırma matrisi için geometrik ortalama aşağıdaki şekildedir.

$$a_1 = \left(\prod_{j=1}^{3} a_{1j} \right)^{1/3} = \left(a_{11} x a_{12} x a_{13} \right)^{1/3} = \left(1x\frac{1}{4}x\frac{1}{2} \right)^{1/3} = 0.5$$

$$a_2 = \left(\prod_{j=1}^{3} a_{2j} \right)^{1/3} = 1.2599$$

$$a_3 = \left(\prod_{j=1}^{3} a_{3j} \right)^{1/3} = 1.000$$

Bu nedenle $a = \sum_{i=1}^{3} a_i = 2.7599$ olur.

Benzer şekilde b_i ve b, c_i ve c, d_i ve d'leri de hesaplayabiliriz. Bu değerler,

	1	**2**	**3**
$\mathbf{a_i}$	0.500	1.2599	1.000
$\mathbf{b_i}$	0.5503	1.4422	1.000
$\mathbf{c_i}$	0.5503	0.8171	1.2599
$\mathbf{d_i}$	0.6300	2.000	1.2599

matrisinde özetlenmiştir.

Sonuç olarak *(a, b, c, d)* = (2.7599, 2.9925, 3.6273, 3.8899) olarak bulunur.

$\forall$ *i* için r_{i1} performans skorları aşağıdaki gibi elde edilebilir.

$$r_{11} = \left(\frac{a_1}{d}, \frac{b_1}{c}, \frac{c_1}{b}, \frac{d_1}{a} \right) = \left(0.1285, 0.1517, 0.1839, 0.2283 \right)$$

$$r_{21} = \left(\frac{a_2}{d}, \frac{b_2}{c}, \frac{c_2}{b}, \frac{d_2}{a} \right) = \left(0.3239, 0.3976, 0.6072, 0.7247 \right)$$

$$r_{31} = \left(\frac{a_3}{d}, \frac{b_3}{c}, \frac{c_3}{b}, \frac{d_3}{a} \right) = \left(0.2751, 0.2757, 0.4210, 0.4565 \right)$$

Adım 2'yi diğer karşılaştırma matrisleri için de tekrarlarız ve birer bulanık sayı olan r_{i2}, r_{i3} ve w_j, $\forall\, i, j$ sonuçlarını elde ederiz.

Adım 3: Tüm bulanık sayılar bulanık çok nitelikli karar verme problemi olarak birleştirilir. Bu değerler aşağıdaki gibi bulunur.

$$\begin{array}{cccc} & C_1 & C_2 & C_3 \\ \begin{array}{c} A_1 \\ A_2 \\ A_3 \end{array} & \multicolumn{3}{c}{\begin{bmatrix} (0.1285, 0.1517, 0.1839, 0.2283) & (0.4991, 0.5162, 0.8594, 0.9107) & (0.3957, 0.6106, 0.6495, 0.8936) \\ (0.3239, 0.3976, 0.6072, 0.7247) & (0.0904, 0.0985, 0.1640, 0.1650) & (0.2069, 0.2774, 0.3248, 0.4918) \\ (0.2751, 0.2757, 0.4210, 0.4565) & (0.1373, 0.1420, 0.2980, 0.2999) & (0.0652, 0.0763, 0.0894, 0.1123) \end{bmatrix}} \end{array}$$

$$\begin{array}{ccc} w_1 & w_2 & w_3 \\ \end{array}$$
$$\big[(0.0834, 0.1111, 0.1111, 0.1663) \quad (0.5678, 0.6667, 0.6667, 0.7833) \quad (0.1596, 0.2222, 0.2222, 0.2839)\big]$$

Adım 2'de kullanılan hesaplama prosedürlerini göstermek için U_1'in ayrıntılı olarak hesaplanmasını aşağıda verelim. Herşeyden önce $w_1 r_{11}$ çarpımı yamuksal bulanık sayıların çarpım kuralına göre

$$w_1 \times r_{11} = \{(a_1 a_2)[L_1, L_2], b_1 b_2, c_1 c_2, (d_1 d_2)[R_1, R_2]\}$$

$$= \{(0.0107)[0.00064, 0.00549], 0.0168, 0.0204, (0.0379)[0.00244, -0.0199]\}$$

şeklinde elde edilir. Burada

$$r_{11} = (0.1285, 0.1517, 0.1839, 0.2283) = (a_1, b_1, c_1, d_1)$$

ve

$$w_1 = (0.0834, 0.1111, 0.1111, 0.1663) = (a_2, b_2, c_2, d_2)$$

$$\begin{aligned} L_1 &= (b_1 - a_1)(b_2 - a_2) \\ L_2 &= a_2(b_1 - a_1) + a_1(b_2 - a_2) \\ R_1 &= (d_1 - c_1)(d_2 - c_2) \\ R_2 &= -[d_2(d_1 - c_1) + d_1(d_2 - c_2)]dir. \end{aligned}$$

Aynı şekilde $w_2 r_{12}$ ve $w_3 r_{13}$'ler de hesaplanır. Tüm hesaplanan değerler aşağıdaki tabloda yer almaktadır.

j	$w_j r_{1j}$
1	$\{(0.0107)[0.00064,0.00549],0.0168,0.0204,(0.0379)[0.00244,-0.0199]\}$
2	$\{(0.2834)[0.00169,0.05907],0.3441,0.5730,(0.7134)[0.00598,-0.14637]\}$
3	$\{(0.0632)[0.01345,0.05907],0.1357,0.1443,(0.2537)[0.01506,-0.12444]\}$

Bu üç bulanık sayıyı yamuksal bulanık sayıların toplama kurallarına göre toplayarak U_1'i elde edebiliriz.

$$U_1 = \{(0.3573)[0.0158,0.1236],0.4966,0.7377,(1.0050)[0.0235,-0.2907]\}$$

$\mu_{U_1}(x)$ 'in üyelik fonksiyon değeri

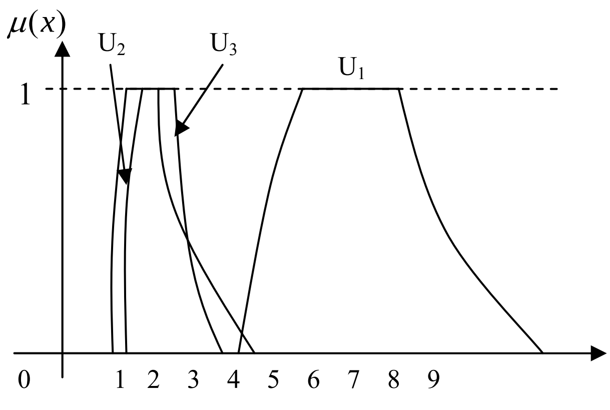
X	$\mu_{U_1}(x)$
≤ 0.3573	0
≥ 1.005	0
$0.4966 \leq x \leq 0.7377$	1
$0.3573 \leq x \leq 0.4966$	$\alpha \in [0,1]$
$0.4966 \leq x \leq 0.7377$	$\alpha \in [0,1]$

şekilde özetlenebilir ve

$$x \in [0.3573,0.4966] \quad iken \quad x = (0.01578)\alpha^2 + (0.12363)\alpha + 0.3573 \quad ve$$
$$x \in [0.7377,1.0050] \quad iken \quad x = (0.02348)\alpha^2 + (-0.29071)\alpha + 1.0050$$

şeklinde tanımlıdır.

U_2 ve U_3 bulanık faydaları da benzer biçimde elde edilebilir. Bunlar Şekil 4.1. 'de görülmektedir.



Şekil 4.1. Bulanık fayda

4.3. Chang'ın Bulanık AHP Yöntemi [Chang, 1996]

$X = \{x_1. x_2, . . ., x_n\}$ nesneler kümesi olsun, ve $G = \{g_1. g_2. . . ., g_m\}$de bir amaç kümesi olsun. Chang'ın derece analizi modeline (extent anlysis model) göre, her bir nesne alınır ve her bir amaç için derece analizi sırasıyla uygulanır. Bu yüzden her bir nesne için aşağıda gösterildiği gibi m tane derece analiz değeri elde edilir.

$M^1{}_{gi}, M^2{}_{gi},...,M^m{}_{gi}, \quad i = 1, 2, ..., n$

Burada tüm $M^j{}_{gi}$'ler, $(j = 1, 2, ..., m)$ üçgensel bulanık sayılardır.

Tanım 4.1: m tane amaç için i'inci nesnenin derece analizi değerleri $M^1{}_{gi}, M^2{}_{gi},...,M^m{}_{gi}$ olsun. *i.* nesneye göre bulanık sentetik derece değeri

$$S_i = \sum_{j=1}^{m} M_{g_i}^{j} \otimes \left[\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} \right]^{-1} \tag{4.6}$$

şeklinde tanımlanır.

AHP' de ilk olarak aynı hiyerarşideki her faktör çifti için göreceli önemlere karar verilir. Üçgensel bulanık sayılar kullanılarak ikili karşılaştırmalar yoluyla elde edilen bulanık değerlendirme matrisi $A = (a_{ij})_{nxm}$ yapılandırılır. Örneğin, belli bir kritere göre i. elemanın j. elemana üstünlüğü güçlü bir şekilde önemli ise, $a_{ij} = (l, 5, u)$ olacaktır. Burada, l ve u kararın bulanıklık derecesini göstermektedir. $u - l$ büyüdükçe bulanıklık derecesi de büyüyecektir. $u - l = 0$ olduğunda karar bulanık olmayan bir sayı olacaktır. Eğer j elemanının önemi i elemanına göre çok güçlü ise, ikili karşılaştırma skalası, aşağıdaki şekilde bulanık bir sayı ile gösterilebilir.

$$a_{ij}^{-1} = \left(\frac{1}{u}, \frac{1}{m}, \frac{1}{l} \right)$$

$A= (a_{ij})_{nxm}$ bir bulanık ikili karşılaştırına matrisi olsun. Burada $a_{ij}= (l_{ij}, m_{ij}, u_{ij})$ aşağıdaki şartı sağlar,

$$l_{ij} = \left(\frac{1}{l_{ji}}\right),\ m_{ij} = \left(\frac{1}{m_{ji}}\right),\ u_{ij} = \left(\frac{1}{u_{ji}}\right)$$

Her bir kriter altındaki ağırlık vektörleri için tahminlerin elde edilebilmesi için, bulanık sayıların karşılaştırılmasını sağlayan bir prensip gereklidir. Bunun için, olabilirlik derecesinin elde edilmesi gerekir.

Tanım 4.2. $M_1 \geq M_2$ için olabilirlik derecesi

$$V(M_1 \geq M_2) = sup[min(\mu_{M_1}(x), \mu_{M2}(y))] \tag{4.7}$$

şeklinde tanımlanır.

$x \geq y$ *ve* $\mu_{M_1} = \mu_{M2} = 1$ olacak şekilde bir (x, y) çifti olduğunda $V(M_1 \geq M_2) = 1$ 'dir. M_1 ve M_2 konveks bulanık sayılar olduğundan

$$V(M_1 \geq M_2) = 1 \Leftrightarrow m_1 \geq m_2,$$

$$V(M_2 \geq M_1) = hgt(M_1 \cap M_2) = \mu_{M_2}(d) \tag{4.8}$$

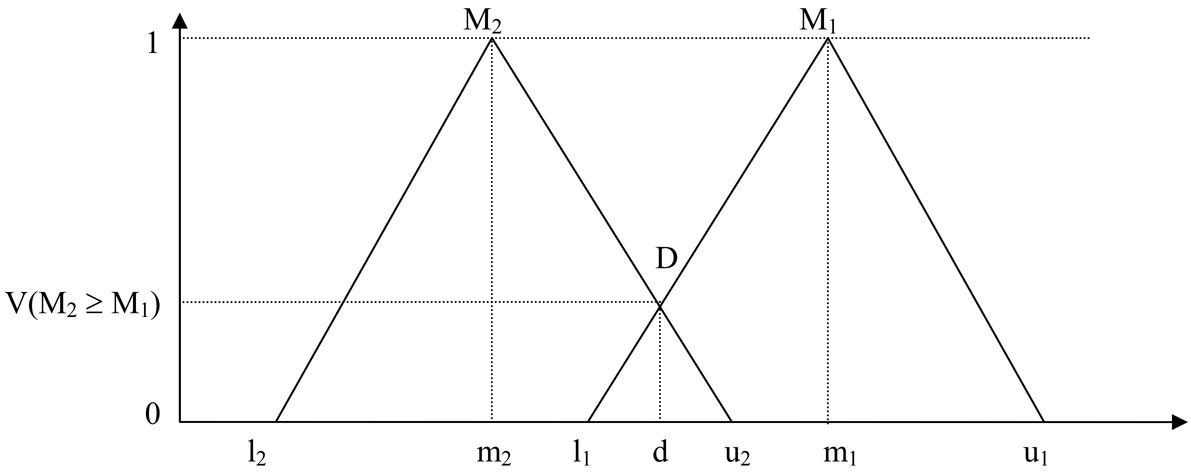
dir. Burada d, μ_{M_1} ve μ_{M2} arasındaki en yüksek kesişim noktasının (D noktasının) ordinatıdır. (Şekil 4.2.)

$M_1 = (l_1, m_1, u_1)$ ve $M_2 = (l_2, m_2, u_2)$ olduğunda D'nin ordinatı

$$V(M_2 \geq M_1) = hgt(M_1 \cap M_2) = \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)} \tag{4.9}$$

denklemiyle verilir.

M_1 ve M_2'yi karşılaştırmak için $V(M_1 \geq M_2)$ ve $V(M_2 \geq M_1)$ değerlerinin her ikisine de ihtiyacımız vardır.



Şekil 4.2. Bulanık sayıların en yüksek D kesişim noktası

Tanım 4.3: Bir konveks bulanık sayının k tane konveks bulanık sayıdan, M_i (i = 1, 2, . . ., k), daha büyük olması için olabilirliğinin derecesi aşağıdaki şekilde tanımlanır.

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

$$= \min V(M \geq M_i), \quad i = 1, 2, 3, \ldots, k. \tag{4.10}$$

$k = 1, 2, \ldots, n;\ k \neq i$ için

$$d'(A_i) = \min V(S_i \geq S_k) \tag{4.11}$$

olduğunu varsayalım. Bu durumda ağırlık vektörü

$$W' = (d'(A_1), d'(A_2), ..., d'(A_n))^T \tag{4.12}$$

şeklinde verilir. Burada A_i ‘ler (i = 1,2,...,n) n tane elemandır.

Bu değerleri normalize ederek, normalize edilmiş ağırlık vektörünü elde ederiz.

$$W = (d(A_1), d(A_2), ..., d(A_n))^T \tag{4.13}$$

Burada W bulanık bir sayı değildir.

Örnek 4.3. [Chang, 1996]

Aşağıdaki örnek orjinali van Laarhoven tarafından sunulan problemin bir modifikasyonudur. Bir üniversitede açılan bir doçentlik kadrosuna başvuran 3 aday vardır. Bu adayları A_1, A_2, A_3 ile gösterelim. İş için en uygun adayı belirlemek için bir komite toplanmıştır. Komite üç üyeden oluşmaktadır. Komite aşağıdaki karar kriterlerini belirlemiştir:

1) Matematiksel yaratıcılık (C_1);

2) Uygulama yaratıcılığı (C_2);

3) Yöneticilik yeteneği (C_3);

4) İnsani olgunluk (C_4).

<u>**Adım 1**</u>**.**İkili karşılaştırmalar yoluyla amaca uygun olan $\Re$ bulanık değer matrisi Tablo 4.7.' de verilmiştir.

Tablo 4.7. Performans Kriterlerinin İkili Karşılaştırmalar Matrisi ($\Re$)

	C_1	C_2	C_3	C_4
C_1	(1,1,1)	(2/3,1,3/2) (2/5,1/2,2/3) (3/2,2,5/2)	(2/3,1,3/2)	(2/7,1/3,2/5) (2/7,1/3,2/5) (2/5,1/2,2/3)
C_2	(2/3,1,3/2) (3/2,2,5/2) (2/5,1/2,2/3)	(1,1,1)	(5/2,3,7/2) (5/2,3,7/2)	(2/3,1,3/2) (2/3,1,3/2) (3/2,2,5/2)
C_3	(2/3,1,3/2)	(2/7,1/3,2/5) (2/7,1/3,2/5)	(1,1,1)	(2/5,1/2,2/3)
C_4	(5/2,3,7/2) (5/2,3,7/2) (3/2,2,5/2)	(2/3,1,3/2) (2/3,1,3/2) (2/5,1/2,2/3)	(3/2,2,5/2)	(1,1,1)

Bulanık sayılarda tanımlanan toplama işlemini kullanılarak ve ortalama değerlerini alınarak aşağıdaki işlemler yapılırsa Tablo 4.8. elde edilir.

C_1 kriterinin C_1 kriterine üstünlüğü ve benzer şekilde C_2 kriterinin C_2 kriterine üstünlüğü C_3 kriterinin C_3 kriterine üstünlüğü C_4 kriterinin C_4 kriterine üstünlüğü (1,1,1) olacaktır.

C_1 kriterinin C_2 kriterine üstünlüğü hesaplanırken aritmetik ortalama yoluyla

$[(2/3,1,3/2) \oplus (2/5,1/2,2/3) \oplus (3/2,2,5/2)] \;/\; 3 = (2/3+2/5+3/2, 1+1/2+2, 3/2+2/3+5/2) \;/\; 3$

$= (2.57,3.5,4.67) \;/\; 3$

$= (0.86,1.17,1.56)$

bulunur.

C_1 kriterinin C_3 kriterine üstünlüğü (0.67,1,1.5)

C_1 kriterinin C_4 kriterine üstünlüğü

$[(2/7,1/3,2/5) \oplus (2/7,1/3,2/5) \oplus (2/5,1/2,2/3)] \;/\; 3 = (2/7+2/7+2/5, 1/3+1/3+1/2, 2/5+2/5+2/3) \;/\; 3$

$= (0.97,1.17,1.47) \;/\; 3$

$= (0.33,0.39,0.49)$

C_2 kriterinin C_3 kriterine üstünlüğü

$[(5/2,3,7/2) \oplus (5/2,3,7/2)] \;/\; 2 = (5/2+5/2, 3+3, 7/2+7/2) \;/\; 2$

$= (2.5,3,3.5)$

C_2 kriterinin C_4 kriterine üstünlüğü

$[(2/3,1,3/2) \oplus (2/3,1,3/2) \oplus (3/2,2,5/2)] \;/\; 3 = (2/3+2/3+3/2, 1+1+2, 3/2+3/2+5/2) \;/\; 3$

$= (2.83,4,5.5) \;/\; 3$

$= (0.95,1.33,1.83)$

C_3 kriterinin C_4 kriterine üstünlüğü (0.4,0.5,0.67) olarak elde edilir.

Bu değerler kullanılarak $C_2 - C_1$, $C_3 - C_1$, $C_3 - C_2$, $C_4 - C_1$, $C_4 - C_2$, $C_4 - C_3$ karşılaştırmaları için karşıt (reciprocal) değerler bulunur. Örneğin $C_1 - C_2$: (0.86,1.17,1.56) iken $C_2 - C_1$: (1/1.56,1/1.17,1/0.86) = (0.64,0.85,1.16) olarak elde edilir.

Tablo 4.8. ($\Re'$)

	C_1	C_2	C_3	C_4	W_c
C_1	(1,1,1)	(0.86,1.17,1.56)	(0.67,1,1.5)	(0.33,0.39,0.49)	0.13
C_2	(0.64,0.85,1.16)	(1,1,1)	(2.5,3,3.5)	(0.95,1.33,1.83)	0.41
C_3	(0.67,1,1.5)	(0.29,0.33,0.4)	(1,1,1)	(0.4,0.5,0.67)	0.03
C_4	(2.04,2.56,3)	(0.55,0.75,1.05)	(1.5,2,2.5)	(1,1,1)	0.43

(4.6) formülü kullanılarak S_1, S_2, S_3, S_4 sentetik derece değerlerini aşağıdaki şekilde elde ederiz.

$$S_1 = \sum_{j=1}^{4} M_{g_i}^{j} \otimes \left[\sum_{i=1}^{4} \sum_{j=1}^{4} M_{g_i}^{j} \right]^{-1}$$

$$= \left[(1,1,1) \oplus (0.86,1.17,1.56) \oplus (0.67,1,1.5) \oplus (0.33,0.39,0.49) \right] \otimes \left(\frac{1}{23.18}, \frac{1}{18.88}, \frac{1}{15.59} \right)$$

$$= (2.86,3.56,4.55) \otimes \left(\frac{1}{23.18}, \frac{1}{18.88}, \frac{1}{15.59} \right)$$

$$= (0.12,0.19,0.29)$$

$$S_2 = (5.09,6.18,7.49) \otimes \left(\frac{1}{23.18}, \frac{1}{18.88}, \frac{1}{15.59} \right)$$

$$= (0.22,0.32,0.48)$$

$$S_3 = (2.56,2.83,3.56) \otimes \left(\frac{1}{23.18}, \frac{1}{18.88}, \frac{1}{15.59} \right)$$

$$= (0.11,0.15,0.23)$$

$$S_4 = (5.08,6.31,7.58) \otimes \left(\frac{1}{23.18}, \frac{1}{18.88}, \frac{1}{15.59} \right)$$

$$= (0.21,0.33,0.49)$$

(4.8) ve (4.9) formüllerini kullanarak olabilirlilik derecelerini aşağıdaki gibi hesaplayalım.

$$V(S_1 \geq S_2) = \frac{0.22-0.29}{(0.19-0.29)-(0.32-0.22)} = 0.35$$

$$V(S_1 \geq S_3) = 1$$

$$V(S_1 \geq S_4) = \frac{0.21-0.29}{(0.19-0.29)-(0.33-0.21)} = 0.32$$

$$V(S_2 \geq S_1) = 1$$

$$V(S_2 \geq S_3) = 1$$

$$V(S_2 \geq S_4) = \frac{0.21-0.48}{(0.32-0.48)-(0.33-0.21)} = 0.96$$

$$V(S_3 \geq S_1) = \frac{0.12-0.23}{(0.15-0.23)-(0.19-0.12)} = 0.73$$

$$V(S_3 \geq S_2) = \frac{0.22-0.23}{(0.15-0.23)-(0.32-0.22)} = 0.06$$

$$V(S_3 \geq S_4) = \frac{0.21-0.23}{(0.15-0.23)-(0.33-0.21)} = 0.10$$

$$V(S_4 \geq S_1) = 1$$

$$V(S_4 \geq S_2) = 1$$

$$V(S_4 \geq S_3) = 1$$

Son olarak (4.10) formülünü kullanarak

$$d'(C_1) = \min V(S_1 \geq S_2, S_3, S_4) = \min (0.35,1,0.32) = 0.32$$

$d'(C_2) = \min V(S_2 \geq S_1, S_3, S_4) = \min(1,1,0.96) = 0.96$

$d'(C_3) = \min V(S_3 \geq S_1, S_2, S_4) = \min(0.73,0.06,0.10) = 0.06$

$d'(C_4) = \min V(S_4 \geq S_1, S_2, S_3) = \min(1,1,1) = 1$

değerleri elde edilir. Buradan da (4.11) formülü gereğince W' ağırlık vektörü

$W' = (0.32,0.96,0.06,1)^T$

şeklinde bulunur.

Bu değerler normalize edilerek C_1, C_2, C_3, C_4 karar kriterlerine göre W ağırlık vektörü

$W = (0.13,0.41,0.03,0.43)^T$

şeklinde bulunur.

<u>Adım 2</u>: Karar prosedürünün ikinci aşamasında komite ayrı ayrı her bir kriter altında A_1, A_2 ve A_3 adaylarını karşılaştırır. Bu karşılaştırma matrisleri ($\Re_1, \Re_2, \Re_3, \Re_4$) aşağıda verilmiştir.

$\Re_1$ matrisi

C_1	A_1	A_2	A_3
A_1	(1,1,1)	(2/3,1,3/2) (2/3,1,3/2)	(2/3,1,3/2) (2/5,1/2,2/3)
A_2	(2/3,1,3/2) (2/3,1,3/2)	(1,1,1)	(2/5,1/2,2/3)
A_3	(2/3,1,3/2) (3/2,2,5/2)	(3/2,2,5/2)	(1,1,1)

$\Re_2$ matrisi

C_2	A_1	A_2	A_3
A_1	(1,1,1)	(5/2,3,7/2)	(3/2,2,5/2)
A_2	(2/7,1/3,2/5)	(1,1,1)	-
A_3	(2/5,1/2,2/3)	-	(1,1,1)

$\Re_3$ matrisi

C_3	A_1	A_2	A_3
A_1	(1,1,1)	(5/2,3,7/2) (5/2,3,7/2) (3/2,2,5/2)	(5/2,3,7/2)
A_2	(2/7,1/3,2/5) (2/7,1/3,2/5) (2/5,1/2,2/3)	(1,1,1)	(2/3,1,3/2)
A_3	(2/7,1/3,2/5)	(2/3,1,3/2)	(1,1,1)

$\Re_4$ matrisi

C_4	A_1	A_2	A_3
A_1	(1,1,1)	-	(3/2,2,5/2) (2/5,1/2,2/3)
A_2	-	(1,1,1)	(3/2,2,5/2)
A_3	(2/5,1/2,2/3) (3/2,2,5/2)	(2/5,1/2,2/3)	(1,1,1)

Bu dört matris için birinci adımdaki tüm işlemler uygulanırsa aşağıdaki tablolar sırasıyla elde edilir (Tablo 4.9-4.12).

Tablo 4.9. ($\Re_1'$)

C_1	A_1	A_2	A_3	Wc_1
A_1	(1,1,1)	(0.67,1,1.5)	(0.54,0.75,1.1)	0.28
A_2	(0.67,1,1.5)	(1,1,1)	(0.4,0.5,0.6)	0.21
A_3	(0.91,1.33,1.85)	(1.5,2,2.5)	(1,1,1)	0.51

Tablo 4.10. ($\Re_2'$)

C_2	A_1	A_2	A_3	Wc_2
A_1	(0.33, 0.33, 0.34)	(0.28,0.33,0.39)	(0.25,0.33,0.42)	0.66
A_2	(0.29,0.33,0.4)	(0.33, 0.33, 0.34)	-	0.16
A_3	(0.24,0.32,0.43)	-	(0.33, 0.33, 0.34)	0.19

Tablo 4.11. ($\Re_3'$)

C_3	A_1	A_2	A_3	Wc_3
A_1	(0.33, 0.33, 0.34)	(0.27,0.33,0.40)	(0.28,0.33,0.39)	0.35
A_2	(0.29,0.32,0.4)	(0.33, 0.33, 0.34)	(0.21,0.32,0.47)	0.33
A_3	(0.28,0.32,0.39)	(0.21,0.32,0.47)	(0.33, 0.33, 0.34)	0.32

Tablo 4.12. ($\Re_4'$)

C_4	A_1	A_2	A_3	Wc_4
A_1	(1,1,1)	-	(0.95,1.25,1.59)	0.22
A_2	-	(1,1,1)	(1.5,2,2.5)	0.42
A_3	(0.95,1.25,1.59)	(0.4,0.5,0.67)	(1,1,1)	0.36

Bu matrisler ağırlık tahminlerini elde etmek için kullanılmıştır. Buradan da her bir adayın ayrı ayrı her bir kriter altındaki ağırlıkları Tablo 4.13'da özetlenmiştir.

Tablo 4.13. Kriter Ağırlıkları

Kriter	A_1	A_2	A_3
C_1	0.28	0.21	0.51
C_2	0.66	0.16	0.19
C_3	0.35	0.33	0.32
C_4	0.22	0.42	0.36

Son olarak her bir kriterin ağırlığının (W) adaylar için elde edilen kriterler altındaki ağırlıkları ile çarpılarak toplanmasıyla sonuç skorları elde edilir. Buradan elde edilen değerler Tablo 4.14'te verilmiştir.

Örneğin: A_1 için sonuç skoru aşağıdaki şekilde hesaplanır.

$$0.28\times0.13+0.66\times0.41+0.35\times0.03+0.22\times0.43 = 0.0364+0.2706+0.0105+0.0946=0.4121$$

Tablo 4.14. Sonuç Skorları

	A_1	A_2	A_3
Sonuç Skorları	0.41	0.28	0.25

4.4. Mikhailov Yöntemi [Mikhailov, 2003]

4.4.1. Bulanık Tercih Programlama Metodu (BTP)

$\alpha = \alpha_l$ seviyesinde $F_l = \{l_{ij}(\alpha_l), u_{ij}(\alpha_l)\}$, $m \leq n(n-1)/2$ aralık ikili karşılaştırma kararları kümesini gözönüne alalım. Aralık kararları tutarlı olduğunda elemanları

$$l_{ij}(\alpha) \leq \frac{w_i}{w_j} \leq u_{ij}(\alpha) \tag{4.14}$$

eşitsizliğini sağlayan birçok öncelik vektörü vardır.

Eğer kararlar tutarsız ise tüm aralık kararlarını aynı anda sağlayan herhangi bir öncelik vektörü yoktur. Bunula beraber, tüm kararları mümkün olduğunca sağlayan bir vektör deneyip bulmak mantıklıdır. Bu yeterince iyi bir çözüm vektörünün tüm aralık kararlarını yaklaşık olarak sağlamak zorunda olduğu anlamına gelir ya da

$$l_{ij}(\alpha) \tilde{\leq} \frac{w_i}{w_j} \tilde{\leq} u_{ij}(\alpha) \tag{4.15}$$

dır. Burada $\tilde{\leq}$ sembolu “bulanık küçük eşit” anlamındadır.

Yukarıdaki eşitsizlikleri kullanarak tek taraflı bulanık kısıtlar kümesini aşağıdaki şekilde kolayca ifade edebiliriz.

$$\begin{aligned} & w_i - w_j u_{ij}(\alpha) \tilde{\leq} 0 \\ & -w_i + w_j l_{ij}(\alpha) \tilde{\leq} 0 \end{aligned} \tag{4.16}$$

Yukarıdaki *2m* tane bulanık kısıtın kümesi

$$Rw \tilde{\leq} 0 \tag{4.17}$$

matris formunda verilebilir. Burada $R \in \Re^{2m \times n}$ dir. $R_k w \tilde{\leq} 0,\ k = 1,2,...,2m$ tarafından belirtilen (4.17) bağıntısının k'ıncı satırı bulanık bir lineer kısıtı ifade eder ve

$$\mu_k(R_k w) = \begin{cases} 1 - \dfrac{R_k w}{d_k}, & R_k w \leq d_k, \\ 0 \quad , & R_k w \geq d_k, \end{cases} \tag{4.18}$$

tipinde bir lineer üyelik fonksiyonu tarafından karakterize edilebilir. Burada d_k , $R_k w \leq 0$ kesin eşitsizliğinin yaklaşık tatmininin uygun aralığını ifade eden bir tolerans parametresidir. (4.18) üyelik fonksiyonu k. tek taraf (4.16) kısıtına göre özel bir üstünlük vektörü ile karar vericilerin tatminini ifade eder. $\mu_k(R_k w)$ üyelik fonksiyonun değeri, ona karşılık gelen kesin $R_k w \leq 0$ kısıtı bozulduğunda, sıfıra eşittir. $\mu_k(R_k w)$ üyelik fonksiyonu değeri, kısıt yaklaşık olarak sağlandığında lineer artan ve 1'den küçük pozitif değerler, kısıt tam olarak sağlandığında 1'den büyük değerler alır.

$\mu_k(R_k w)$, $k = 1,2,...,2m$ için (n-1) boyutlu

$$Q^{n-1} = \{(w_1,..., w_n) \mid w_i > 0,\ w_1 + ... + w_n = 1\} \tag{4.19}$$

simpleksi üzerinde $R_k w \tilde{\leq} 0$ bulanık kısıtlarının üyelik fonksiyonları olsun.

Tanım 4.4: Q^{n-1} simpleksi üzerinde

$$\mu_{\tilde{P}}(w) = [\min\{\mu_1(R_1 w),..., \mu_m(R_m w)\} \mid w_1 + ... + w_n = 1] \tag{4.20}$$

üyelik fonksiyonu ile tanımlanan $\tilde{P}$ bulanık uygun (feasible) alanı bir bulanık kümedir.

$\tilde{P}$ bulanık uygun (feasible) alanı simpleks üzerindeki tüm bulanık kısıtların bir kesişimi olarak tanımlanır. Eğer başlangıç aralık kararları tutarsız ise "yeterince geniş" tolerans parametrelerini seçerek boş olmayan bir bulanık uygun alan elde edebiliriz. Q^{n-1} simpleksi üzerinde boş olmayan $\tilde{P}$ uygun alanının bir konveks bulanık küme olduğu kolayca gösterilebilir.

$\widetilde{P}$ konveks bulanık uygun alanı, özel bir w kesin üstünlük vektörü ile karar vericilerin tam tatminini ifade eder. Karar vericinin mümkün olan en iyi sonuç ile ilgilendiğini varsayarak, karar vericinin tüm tatmin derecelerini maksimize eden bir üstünlük vektörü belirlemek mantıklıdır.

Tanım 4.5: Maksimizasyon çözümü

$$\mu_{\widetilde{P}}(w^*) = \max[\min\{\mu_1(R_1 w),...,\mu_m(R_m w)\} \mid w_1 + ... + w_n = 1] \quad (4.21)$$

bulanık uygun alanının maksimumuna karşılık gelen bir w^* kesin vektörüdür.

$\widetilde{P}$ bulanık uygun alanı konveks bir küme olduğundan ve tüm bulanık kısıtlar konveks küme olarak tanımlandıklarından, $\widetilde{P}$'de bir maksimum üyelik derecesine sahip olan simpleks üzerinde her zaman bir w^* noktası vardır.

Maksimizasyon problemi türetmek için kullanılan max-min işlemi bulanık kısıtlar ve bulanık amaçlar ile genel karar verme problemleri için Belman ve Zadeh tarafından önerilmiştir. Bu yaklaşım, amaç ve kısıtları birleştirir, bu şekilde aralarındaki farklılık yok olur. Zimmermann, Bellman ve Zadeh'in lineer bulanık kısıtlar ve lineer bulanık amaçlar ile ilgili fikrini problemler için çalıştırır ve max-min bulanık lineer problemi bir geleneksel lineer programa dönüştürülebilir olduğunu gösterir.

Aynı şekilde yeni bir λ değişkeni tanımlayarak bulanık uygun $\widetilde{P}$ alanında verilen üstünlük vektörünün üyelik derecesinin ölçülmesi ve (4.18) ve (4.21)'i kullanarak lineer maksimizasyon problemini

$$\begin{aligned} &maksimize \quad \lambda \\ &\qquad d_k\lambda + R_k w \le d_k \\ &\qquad \sum_{i=1}^{n} w_i = 1,\ w_i > 0,\ i = 1,2,...,n,\ k = 1,2,...,2m \end{aligned} \quad (4.22)$$

şeklinde tanımlayabiliriz.

Yukarıda lineer problemin optimal çözümü olan *(w^*, λ^*)*'da birinci bileşen bulanık uygun alanda maksimum üyelik derecesine sahip olan üstünlük vektörünü ve ikinci bileşen $\lambda^* = \mu_{\tilde{P}}(w^*)$ olan maksimum derece değerini verir. λ^* değeri tatmin derecesini ölçer ve karar vericilerin tutarsızlıkları için doğal bir göstergedir. Bu sebepten λ'ya tutarlılık indeksi denir. Aralık kararları tutarlı olduğunda $\lambda^* \geq 1$ olur. Tutarsız kararlar için ise d_k tolerans parametresi ve tutarsızlık derecesine bağlı olarak "0" ile "1" arasında değerler alır.

Tolerans parametresi, λ^*'ın pozitif değerini ve uygun alanın boş olmamasını kesinleştirmek için yeterince büyük seçilmelidir. Bir sonraki kısımda parametre değeri "1"e eşit ve ya "1"den büyük olduğunda bu gibi gereksinimleri sağladığı görülmektedir.

Eğer bulanık karar simetrik ise tolerans parametresini "1"e eşitlemek mantıklı olur, çünkü karar vericinin kişisel ikili karşılaştırmalarında bir tercihi yoktur. Dikkat edilmelidir ki eşit tolerans prametreleri w^* maksimizasyon çözümüne etki etmez.

4.4.2. Sayısal Sonuçlar

Önerilen yaklaşımı uygulayarak iki boyutlu örneği inceleyelim.

Örnek 4.4.

$\tilde{a}_{12} = (1,2,3)$ olan iki boyutlu bir karşılaştırmalı matrisini gözönüne alalım.

	A_1	A_2
A_1	(1,1,1)	(1,2,3)
A_2	(1/3,1/2,1)	(1,1,1)

Karar verici tarafından verilen bulanık karar Ü.B.S.'dır. α -kesim aralıklarının sınırları

$$l_{12}(\alpha) = \alpha(m_{12} - l_{12}) + l_{12},$$
$$u_{12}(\alpha) = \alpha(m_{12} - u_{12}) + u_{12}$$

olduğunda, her α -kesim seviyesinde öncelik oranları (4,14)'ü sağlamalıdır.

Her bir α - kesim seviyesinde problemin çözümü

$$\begin{aligned} &\textit{maksimize} \quad \lambda \\ &d_1\lambda + w_1 - u_{12}(\alpha)w_2 \leq d_1, \\ &d_2\lambda - w_1 + l_{12}(\alpha)w_2 \leq d_2, \\ &w_1 + w_2 = 1,\ w_i > 0,\ i = 1,2 \end{aligned} \tag{4.23}$$

tipindeki bir Lineer Programlama (LP) ile çözülür. Eğer $d_1 = d_2 = d$ olursa çözüm d'ye bağlı olmaz. $\alpha = 0$ için çözüm:

$$\begin{aligned} &d\lambda + w_1 - 3w_2 \leq d, \\ &d\lambda - w_1 + w_2 \leq d, \\ &w_1 + w_2 = 1,\ w_1, w_2 > 0, \end{aligned}$$

denklemleri taraf tarafa çıkarılarak

$2w_1 - 4w_2 \leq 0$

bulunur. Buradan

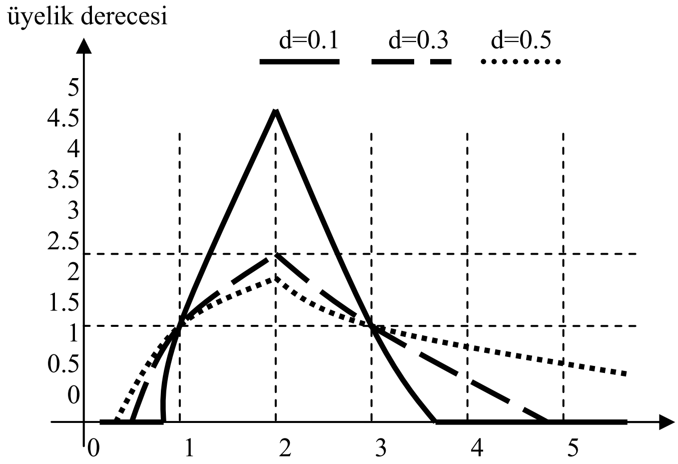
$2w_1 \leq 4w_2$ ve $\frac{w_1}{w_2} \leq 2$ elde edilir. $w_1 + w_2 = 1,\ w_1, w_2 > 0,$ koşulu gereğince

$w_1=0.667$

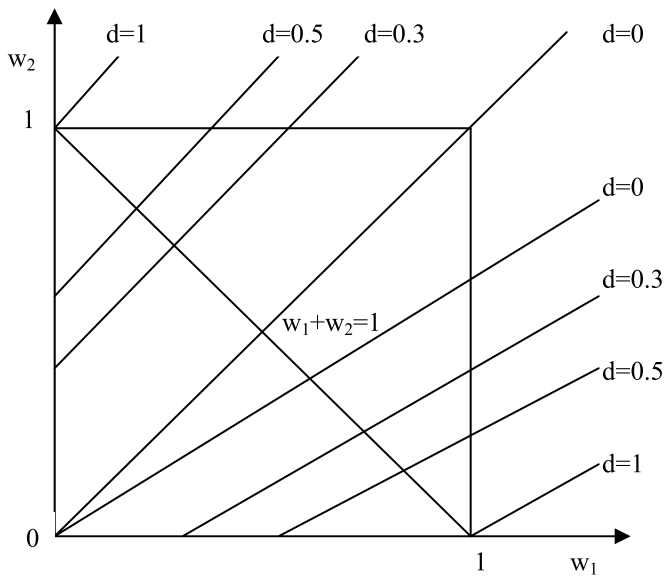
ve

$w_2=0.333$

bulunur.



Şekil 4.3. Tolerans parametresine bağlı bulanık uygun alan



Şekil 4.4. İki boyutlu önceliklendirme probleminin bulanık lineer kısıtları

w_1-w_2 düzlemindeki bulanık lineer kısıtların iz düşümleri farklı d tolerans parametresi için Şekil 4.4.'te gösterilmiştir. Açıkça görülebilir ki tolerans parametresi $d=1$ iken uygun alan çok geniştir ve tüm $w_1+w_2=1$ simpleks doğrusunu kapsar. Hatta birçok tutarsız karar olsa bile tolerans parametresi herhangi bir değeri için bulanık uygun alanın doluluğunu garanti eder. İki boyutlu öncelik problemleri her zaman tutarlıdır.

Örnek 4.4.’ün çözümleri $w_1=0.667$ ve $w_2=0.333$ ‘tür. Çünkü başlangıç bulanık kararı simetriktir ve α-kesim aralıklarının ortası bulanık kararın ortasına eşittir. Optimal çözüm oranı bulanık kararın ortasına karşılık gelir ve her aralık sınırlı kısıtları eşit olarak sağlar. Her α-kesim seviyesi için λ’nın “1”den büyük olması tutarlı örneklendirme problemlerine sahip olduğumuzu gösterir.

Yine iki boyutlu bir önceliklendirme problemini ele alalım. Bulanık karar simetrik değilse α-seviyelerindeki lineer programlama sonuçları eşit olmaz.

Örnek 4.5.

$\tilde{a}_{12} = (0.5,1,3)$ olsun. Bu problemin sonuçları Örnek 4.4.’teki şekilde çözülerek Tablo 4.15.’te verilmiştir.

Tablo 4.15. Örnek 4.5. için sonuçlar.

α	$l_{12}(\alpha)$	$l_{12}(\alpha)$	w_1	w_2	w_1/w_2	λ
0	0.5	3	0.636	0.364	1.750	1.455
0.1	0.55	2.8	0.626	0.374	1.675	1.421
0.2	0.60	2.6	0.615	0.385	1.600	1.385
0.3	0.65	2.4	0.604	0.396	1.525	1.347
0.4	0.70	2.2	0.592	0.408	1.450	1.306
0.5	0.75	2	0.579	0.421	1.375	1.263
0.6	0.80	1.8	0.565	0.435	1.300	1.217
0.7	0.85	1.6	0.551	0.449	1.225	1.169
0.8	0.90	1.4	0.535	0.465	1.150	1.116
0.9	0.95	1.2	0.518	0.482	1.075	1.060
1.0	1.0	1.0	0.500	0.500	1.000	1.000

Tablo 4.15 çözüm oranlarının daha önceki örnekte olduğu gibi kesinlikle α-seviye aralıklarının ortasında olduğunu gösterir.

Yaklaşımın performansını daha iyi görebilmek için bazı aralık kararlarının tutarsız olduğu Boender’in örneğini inceleyelim.

Örnek 4.6. [Mikhailov Yöntemi (2003)]

$\tilde{a}_{21} = (2.5,3,3.5)$, $\tilde{a}_{31} = (4,5,6)$, $\tilde{a}_{32} = (1.5,2,2.5)$ olsun. Tüm tolerans parametrelerini "1"e eşitleyerek önerilen yaklaşımı uygularsak Tablo 4.16.'daki sonuçları elde ederiz.

Tablo 4.16. Örnek 4.6. için sonuçlar.

α	w_1	w_2	w_3	λ
0	0.1111	0.3333	0.5556	1.056
0.1	0.1099	0.3297	0.5604	1.049
0.2	0.1087	0.3261	0.5652	1.043
0.3	0.1075	0.3226	0.5699	1.038
0.4	0.1075	0.3197	0.5728	1.029
0.5	0.1081	0.3171	0.5748	1.020
0.6	0.1087	0.3146	0.5767	1.010
0.7	0.1093	0.3123	0.5785	1.001
0.8	0.1099	0.3099	0.5802	0.9913
0.9	0.1105	0.3077	0.5818	0.9817
1.0	0.1111	0.3056	0.5833	0.9722

Tutarlılık indeksinin tüm $\alpha \leq 0.7$ için $\lambda > 1$ olması karşılık gelen α -seviyeleri için aralık kararlarının tutarlı olduğu anlamına gelir. Bu şekilde çözüm oranları bu aralıklarda bulunur.

4.4.3. Tolerans Parametrelerinin Ayarlanması

Önceki örneklerden gözlendiği üzere kusursuz tutarlı durumlarda çözüm oranlarımız tolerans parametreleri eşit olduğunda aralıkların ortasında yeralmaktadır. Hatta bulanık kararlar simetrik ise çözüm oranları bulanık kararların ortasına eşittir. Genel olarak $m_{ij}=m_{ik}.m_{kj}$ kusursuz tutarlılık şartı varsa ve bulanık kararlar simetrik ise BTP ile elde edilen sonuçlar her zaman kararların m_{ij} değerine eşit olacaktır. Orta noktalar başlangıç bulanık kararları içinde en yüksek üyelik derecesine sahip olan değerler oldukları için üyelik fonksiyonundaki tolerans parametrelerini ayarlamaya çalışmak mantıklı olacaktır. Bu şekilde kararlar simetrik olmadığında ve ya tutarsız olduğunda çözüm oranlarını maksimum üyelik derecesine getirmeye çalışırız. $a_{ij}=[l_{ij}(\alpha),u_{ij}(\alpha)]$ olacak şekilde tek başına bir aralığı gözönüne alalım.

$$\mu_{ij}{}^{L} = 1 - \frac{\left(-w_i + w_j l_{ij}(\alpha)\right)}{d_{ij}{}^{L}} = 1 + \frac{\left(w_i / w_j\right) - l_{ij}(\alpha)}{d_{ij}{}^{L} / w_j}$$

$$\mu_{ij}{}^{U} = 1 - \frac{\left(w_i - w_j u_{ij}(\alpha)\right)}{d_{ij}{}^{U}} = 1 + \frac{u_{ij}(\alpha) - \left(w_i / w_j\right)}{d_{ij}{}^{U} / w_j} \tag{4.24}$$

olduğunda en düşük ve en yüksek sınırlara karşılık gelen lineer üyelik fonksiyonlarının pozitif kısımlarının kesişimi Şekil 4.3.'te gösterilen tipin $\mu_{ij} = \min\left(\mu_{ij}{}^{L}, \mu_{ij}{}^{U}\right)$ aralığı için bir konveks üyelik fonksiyonu tanımlar. Burada $d_{ij}{}^{L} ve\, d_{ij}{}^{U}$ bu aralık için en düşük ve en yüksek tolerans parametrelerini ifade eder.

$\mu_{ij} = \min\left(\mu_{ij}{}^{L}, \mu_{ij}{}^{U}\right)$ üyelik fonksiyonu aralığı (4.24)'ten kolaylıkla elde edilebilen

$$\frac{w_i}{w_j} = \frac{d_{ij}{}^{U} l_{ij}(\alpha) + d_{ij}{}^{L} u_{ij}(\alpha)}{d_{ij}{}^{L} + d_{ij}{}^{U}}, \tag{4.25}$$

$\mu_{ij}{}^{L} = \mu_{ij}{}^{U}$ kesişim noktasında maksimuma sahiptir.

Karar simetrik ise bu eşit tolerans parametrelerini uygulayarak maksimize edilen oran

$$\frac{w_i}{w_j} = \frac{l_{ij}(\alpha) + u_{ij}(\alpha)}{2} \tag{4.26}$$

başlangıç bulanık kararının üyelik fonksiyonu içerisinde maksimum dereceye sahip olur. Fakat karar simetrik değilse maksimize oranı m_{ij}'ye karşılık gelmesi için bir tolerans parametresi seçilebilir. Bu yolla oran $\tilde{a}_{ij}=[l_{ij},m_{ij},u_{ij}]$ bulanık kümesinde maksimum üyelik derecesine sahip olur.

(4.25)'te $w_i / w_j = m_{ij}$ değişken değişimi yaparak ve $\Delta^{L}(\alpha) = m_{ij} - l_{ij}(\alpha),\ \Delta^{U}(\alpha) = u_{ij}(\alpha) - m_{ij}$ düzenlemesiyle

$$\frac{\Delta^L(\alpha)}{d_{ij}{}^L} = \frac{\Delta^U(\alpha)}{d_{ij}{}^U}\,. \tag{4.27}$$

(4.27) eşitliğini kullanılarak bulanık kararların üyelik derecelerini maksimize eder, tolerans parametrelerinin otomatik olarak ayarlanmasını sağlar ve çözüm oranını elde edebiliriz. Çünkü her aralık için iki tolerans parametresi vardır. $\Delta^L(\alpha)$ *ve ya* $\Delta^U(\alpha)$'ya karşılık gelen örneğin "1"e eşitlenerek sabitlenmelidir. Sonra (4.26)'dan elde edilen ikinci parametre "1"den büyük bir değer alacaktır ki bu bize boş olmayan bulanık uygun alanı garanti eder. Eğer bulanık karar simetrikse $\Delta^L(\alpha) = \Delta^U(\alpha)$ ve bunu takiben $d_{ij}{}^L = d_{ij}{}^U = 1$ olur. Önerilen prosedürü her α-seviyesinde ikinci örnekteki iki boyutlu simetrik olmayan tolerans parametrelerini ayarlamak için uygulayarak $w_1= w_2=0.5$ değerini elde ederiz. Böylece tüm karar oranları bulanık kararın ortasına eşit olur. Örnek 4.6.'daki bulanık kararların hepsi simetrik olduğundan tolerans parametresi ayarlamaya ihtiyaç yoktur.

4.4.4. Üstünlüklerin Birleştirilmesi

BTP metodunu uygulayarak ve α-kesimleriyle bulanık karşılaştırma kararlarını aralık kararları serisine ayrıştırarak, her α-kesim seviyesine karşılık gelen kesin üstünlüklerin dizisini elde ederiz.

$$w(\alpha_l) = \left(w_1(\alpha_l), w_2(\alpha_l), \ldots, w_n(\alpha_1)\right)^T,\ l = 1,\ldots,L,\ \ 0 = \alpha_1 < \alpha_1 < \ldots < \alpha_L = 1$$

Bununla beraber üstünlüklerin göreceli önemleri α-seviyesine bağlı ve aynı değildir. α'nın küçük değeri aralık kararlarının yapısını verir. α geniş açılımlara sahiptir ve yüksek seviyeli belirsizliklerle düşük güvenli üstünlüklere işaret eder. Temel olarak bulanık kararın desteği en güvenlidir, fakat bilinmeyen üstünlük oraları için en kötümser destektir. Büyük α değeri daha küçük fakat daha iyimser aralık kararları verir. Bu karar aralıklarının alt ve üst sınırları başlangıç bulanık kümesi üyelik derecelerinden daha büyüktür. Bulanık kararlar Ü.B.S. ile gösterildiğinde α=1 kesimi kesin kararlar üretir. Bu a_{ij} bulanık karşılaştırma oranlarının en mümkün m_{ij} değerine eşittir. Bu nedenle karar verici üstünlüklerine yüksek α değerlerine denk gelecek daha fazla güven koyar. Bu sebeplerle α değeri çözümleri ağırlıklandıran faktör gibi kullanılabilir. Buradan birleştirilmiş üstünlük değerlerini ağırlıklı toplama

$$W_j = \sum_{l=1}^{L} \alpha_l w_j(\alpha_l) / \sum_{l=1}^{L} \alpha_l \tag{4.28}$$

şeklinde elde edebiliriz.

Önceden verilen tüm örnekler için birleştirilmiş üstünlük değerleri Tablo 4.17. ve 4.18'de görülür.

Tablo 4.17. İki Boyutlu Örneklerin Sonuçları

	Lineer Metod (birleştirilmiş üstünlükler)				Lineer Olmayan (non-linear) Metod		
	Eşit Tolerans Parametreleri		Ayarlanmış Tolerans Parametreleri				
	W_1	W_2	W_1	W_2	w_1	w_2	λ
Örnek 1	0.667	0.333	0.667	0.333	0.667	0.333	1.0
Örnek 2	0.548	0.452	0.5	0.5	0.5	0.5	1.0

Tablo 4.18. Üç Boyutlu Örneğin Sonuçları

Lineer Metod (birleştirilmiş üstünlükler)			Lineer Olmayan (non-linear) Metod			
W_1	W_2	W_3	w_1	w_2	w_3	λ
0.1095	0.3126	0.5779	0.1093	0.3121	0.5786	0.708

Üstünlükleri belirlemek için önerilen yaklaşım AHP içindeki global üstünlükleri bulmada kullanılabilir, fakat her α-seviye kesimi için tüm alternatiflerin tüm skorlarının bulunmasından sonra birleştirme uygulanmalıdır. n kriterli ve m alternatifli üç boyutlu özel bir karar hiyerarşisi olduğunu varsayalım. İkinci aşamada tüm kriterlerin ikili karşılaştırılmasıyla $F = \{\tilde{a}_{ij}\},\ i = 1,2,...,n-1,\ j = 2,3,...n,\ j > i,$ bulanık karşılaştırma karar kümesini elde ederiz. Herbir C_i kriteri için bu kritere dikkat ederek alternatiflerin göreceli önemlerini hesaplayarak diğer bir $S_i = \{\tilde{b}_{ij}\},\ i = 1,2,...,m-1,\ j = 2,3,...,m,\ j > i,$ bulanık karşılaştırmalar kümesini oluşturabiliriz. α-kesimleriyle tüm karşılaştırma kararlarını dağıtarak ve BTP metodunu uygulayarak daha sonra birleştirme prosedüründe kullanılan

$$s_j(\alpha_l) = \left(s_{j1}(\alpha_l), s_{j2}(\alpha_l), ..., s_{jn}(\alpha_1)\right)^T, \; j = 1,2,...,m, \;\; l = 1,...,L,$$

skorlarını ve

$$w(\alpha_l) = \left(w_1(\alpha_l), w_2(\alpha_l), ..., w_n(\alpha_1)\right)^T$$

kesin ağırlıklarının dizisini elde edebiliriz. Klasik AHP'de olduğu gibi her α-kesim seviyesindeki j'inci alternatifin tüm skorları

$$r_j(\alpha_l) = \sum_{i=1}^{n} s_{ji}(\alpha_l) w_i(\alpha_l) \tag{4.29}$$

ağırlıklar toplamıyla hesaplanır. Aynı yolla türeyen tüm alternatiflerin skorları kesin değerlerdir.

Tanım 4.6. A_i alternatifi A_j alternatifine tüm l'ler için ancak ve ancak $r_i(\alpha) > r_j(\alpha)$ olduğunda güçlü bir şekilde üstündür. Eğer varsa diğer alternatiflerden güçlü üstünlüğü olan alternatif final değerlendirilmesinde seçilmelidir. Tüm α değerleri için üstün bir alternatif yoksa (4.27)'de olduğu gibi ilave bir toplama sıralaması yapılabilir. j. alternatifin tüm ağırlıklı skorları aşağıdaki gibi hesaplanır.

$$R_j = \sum_{l=1}^{L} \alpha_l r_j(\alpha_l) / \sum_{l=1}^{L} \alpha_l \; . \tag{4.30}$$

4.4.5. Lineer Olmayan Önceliklendirme

Önceliklendirme için önerilen yaklaşımın (4.22) lineer programlarını ve neticede farklı α-seviyelerinde türeyen önceliklerin birleştirilmesini çözebilmek için α-kesim sayısına ihtiyacı vardır. Bu adımların bazılarından kaçınmak sebebiyle önceliklendirme için lineer olmayan bir metod önerilmektedir. Bu metod ÜBS ile gösterilen karşılaştırma kararları kümesinden önceliklerin kesin değerlerini doğrudan bulmamızı sağlar.

Kesin değerli öncelik vektörünü bulmak istiyoruz. Böylece oranlar (4.15) başlangıç bulanık kararlarını yaklaşık olarak sağlarlar. İki taraflı eşitsizlikleri tek taraflı lineer bulanık kısıtlara çevirmek yerine her bir bulanık karar için w_i/w_j ile bağlı lineer bir üyelik fonksiyonu yapılandırabiliriz.

$$\mu_{ij}\left(\frac{w_i}{w_j}\right)=\begin{cases}\dfrac{\left(\dfrac{w_i}{w_j}\right)-l_{ij}}{m_{ij}-l_{ij}}, & \dfrac{w_i}{w_j}\le m_{ij},\\[2ex] \dfrac{u_{ij}-\left(\dfrac{w_i}{w_j}\right)}{u_{ij}-m_{ij}}, & \dfrac{w_i}{w_j}\ge m_{ij},\end{cases} \tag{4.31}$$

Bu fonksiyon $(-\infty, m_{ij})$ aralığında lineer artan ve (m_{ij}, ∞) aralığında lineer azalandır ve $w_i/w_j<l_{ij}$ ve ya $w_i/w_j>u_{ij}$ iken negatif değerler alır ve $w_i/w_j=1=m_{ij}$ de maksimum değere sahiptir. (l_{ij},u_{ij}) değerlerinde (4.31) üyelik fonksiyonu ÜBS kararı ile uyumludur.

w_i/w_j oranlarına dikkat ederek (4.22) üyelik fonksiyonu ile (4.31) fonksiyonunu kıyaslarsak (4.22) w_i önceliklerine göre lineerdir fakat uygun alan üyelik fonksiyonunun lineer olmayan olmasına sebep olur, (4.31) ise karar değerlerine göre lineer olmayandır fakat bulanık uygun alanı sağlar ve bu oranlarda lineerdir.

4.4.1. kısımdaki gibi (4.19) simpleksi üzerinde tüm (4.31) üyelik fonksiyonlarının kesişimi olan ve maksimize edilen çözümü bulmak için max-min uygulayan bir bulanık uygun alan tanımlayabiliriz. Bu

$$\begin{aligned}&\textit{maksimize} \quad \lambda\\ &(m_{ij}-l_{ij})\lambda w_j - w_i + l_{ij}w_j \le 0,\\ &(u_{ij}-m_{ij})\lambda w_j + w_i - u_{ij}w_j \le 0,\\ &\sum_{k=1}^{n} w_k = 1,\ w_k>0,\ k=1,2,\dots,n,\\ &i=1,2,\dots,n-1,\ \ j=2,3,\dots,n,\ \ j>i\end{aligned} \tag{4.32}$$

lineer olmayan optimizasyon problemine sebep olur.

Lineer ve simpleks yöntemiyle kolay çözülebilen (4.22) optimizasyon probleminin aksine (4.32) lineer olmayan problemin çözümü, çalıştırılabilmesi için uygun bir sayısal metoda ihtiyaç duyar λ tutarlılık indeksinin optimal değeri (eğer pozitif ise) tüm çözüm oranlarının başlangıç bulanık kararlarını sağladığını gösterir. Aksi halde bulanık kararlar ciddi derecede tutarsız ve çözüm oranları bunları yaklaşık olarak sağlar demektir. (4.18)'deki gibi tolerans parametreleriyle üyelik fonksiyonunu şekillendirerek tutarsız oranlar için pozitif olan üyelik fonksiyonuna sahip bir genişletilmiş bulanık uygun alan elde edebiliriz. Böylece tutarsız durumlarda tutarlılık indeksi de pozitif değerlere sahip olur.

Tutarlı iki boyutlu simetrik ÜBS kullanılan örneklerde lineer olmayan metoddan türeyen öncelikler lineer olanlara eşittir, çünkü lineer ve lineer olmayan uygun alanları aynı orana denk gelir, fakat bulanık kararlar kaymış ya da tutarsızsa sonuçlar çok az farklı olur.

4.5. Enea ve Piazza Yaklaşımı [Enea ve Piazza, 2004]

Olası alternatifler kümesinden bir proje seçmek karar vericiler için her zaman zor olmuştur. Dikkate alınması gereken çok sayıda nitelik ve farklı hedefler bu seçimi zorlaştıran başlıca sebeplerdir. Burada Bulanık Analitik Hiyerarşi Prosesi'nin bir genişlemesi olan Enea Piazza yaklaşımını açıklayacağız. Bu yaklaşım mümkün olan tüm bilgiyi dikkate alarak kısıtlar üzerinde yoğunlaşmaktadır. Kesinlik ve güvenilirlik açısından bu yaklaşım kısıtlardan yola çıkılarak daha iyi sonuçlar almayı hedeflemektedir.

Kısıtlı Bulanık AHP

I. Metod: Karşılaştırılacak n tane farklı faktörümüzün olduğu durumu gözönünde bulunduralım. AHP metodunun ilk adımı her bir faktör çiftinin göreceli önemini hesaplamak ve ikili karşılaştırma matrisini oluşturmaktır. Eğer bulanık AHP'yi dikkate alırsak matrisimiz de bulanık sayılardan oluşur. Üçgensel bulanık sayıları kullandığımız takdirde ikili karşılaştırma matrisinin genel elemanları $a_{ij} = (l_{ij}, m_{ij}, u_{ij})$ ile gösterilebilir. Ayrıca

$$a_{ji} = (1/u_{ij}, 1/m_{ij}, 1/l_{ij}) \quad \forall\, i \neq j \tag{4.33}$$

ve

$$a_{ji} = (1,1,1) \quad \forall\, i = j \tag{4.34}$$

dır. Burada sadece pozitif destekli üçgensel bulanık sayılar ele alınmıştır.

Matris oluşturulduktan sonra öncelikle proje ağırlıklarının bulanık tahminleri olan bulanık sentetik derece değerlerini (bulanık AHP etki skorlarını) hesaplamalıyız. i'nci bulanık sentetik derece değerini hesaplamak için Chang'ın standart bulanık aritmetiğini kullanılarak, i'inci satırın elemanlarını toplayıp elde edilen sonuçları matrisin tüm elemanlarının toplamına bölmeliyiz.

$S_i=(S_{li},S_{mi},S_{ui})$ bulanık sentetik derece değerleri olsun. Burada *l*, *m* ve *u* indisleri sırasıyla en düşük (lower), orta (medium) ve en yüksek (upper) anlamına gelmektedir. S_{mi}'yi hesaplamak için Chang tarafından önerilen

$$S_{mi} = \sum_{j=1}^{n} m_{ij} \times \left[\sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij} \right]^{-1} \tag{4.35}$$

denkelemini yazarız.

Bununla birlikte S_{li}'yi hesaplamak için uygun kesin bir $B_i=[b_{kj}]$ matrisini, bulanık ikili karşılaştırma matrisine bağlı olarak

$$b_{jj}=1 \tag{4.36}$$

$$b_{jk}=1/b_{kj} \tag{4.37}$$

$$b_{ij}=l_{ij} \quad \forall\ j \neq i \tag{4.38}$$

$$\mathrm{b}_{\mathrm{kj}} = \{\mathrm{x} \mid \mathrm{y} = \max\ (\mathrm{x} + \frac{1}{\mathrm{x}}),\ \forall\, \mathrm{x} \in \left[\ell_{\mathrm{kj}}, \mathrm{u}_{\mathrm{kj}}\right]\} \quad \forall\ k \neq i,\ \ j \neq i;\ \ j > k \tag{4.39}$$

kısıtları altında oluşturmalıyız.

Kurulan kesin B_i matrisine aşağıdaki kesin formülü uygulamalıyız.

$$S_{li} = \sum_{j=1}^{n} b_{ij} \times \left[\sum_{j=1}^{n} \sum_{k=1}^{n} b_{kj} \right]^{-1} \tag{4.40}$$

(4.36)-(4.38) ifadelerini kullanarak çözüm uzayı indirgenirken, imkansız sonuçlar ihmal edilir. (4.39) ifadesi S_{li} değerinin sentetik derece değerinin minimumu olduğunu garanti eder. Geçekten (4.38)- (4.39) kısıtların uygulaması S_i^k 'yı minimum olmaya zorlar.

Aynı sebeplerden dolayı S_{ui} değerini hesaplamak ve aşağıdaki eşitliği uygulamak için $C_i=[c_{kj}]$ kesin matrisini oluşturmalıyız.

$$S_{ui} = \sum_{j=1}^{n} c_{ij} \times \left[\sum_{j=1}^{n} \sum_{k=1}^{n} c_{kj} \right]^{-1} \tag{4.41}$$

Aşağıdaki bağıntılara göre C_i matrisi yapılandırılır.

$$c_{jj} = 1 \tag{4.42}$$

$$c_{ij} = 1/c_{ji} \tag{4.43}$$

$$c_{ij} = u_{ij} \,, \forall j \neq i \tag{4.44}$$

$$\mathrm{c}_{\mathrm{kj}} = \{\mathrm{x} \mid \mathrm{y} = \min\,(\mathrm{x} + \frac{1}{\mathrm{x}}),\ \forall\, \mathrm{x} \in [\ell_{\mathrm{kj}}, \mathrm{u}_{\mathrm{kj}}]\},\ \ \forall\ k \neq i,\ \ j \neq i;\ \ j > k \tag{4.45}$$

Yeniden (4.42)-(4.44) ifadelerini kullanarak çözüm uzayı indirgenirken, imkansız sonuçlar ihmal edilir. (4.45) ifadesi S_{ui} değerinin sentetik derece değerinin maksimumu olduğunu garanti eder. Geçekten (4.44)-(4.45) kısıtların uygulaması S_i^k 'yı maksimum olmaya zorlar.

Tarif edilen prosedür bulanık sentetik derece değerinin (bulanık AHP etki skoru) hesaplanmasına olanak sağlar.

$S_i = (S_{li}, S_{mi}, S_{ui})$

II.Metod: Daha önce saptanan bulanık değerler, Lootsma'nın (1997) geometrik ortalama metoduyla hesaplanabilir. Bu durumda B_i ve C_i ikili matrislerini oluşturmak için standart bir prosedüre gerek yoktur. Bununla birlikte, basit bir matematiksel programlama modeli kullanılarak, AHP tarafından ortaya konan kısıtları ve (4.33)-(4.34) eşitliklerini gözönünde bulundurarak S_i değeri hesaplanabilir. Kesin S_{mi} değeri aşağıdaki formülünü kullanarak hesaplanabilir.

$$S_{mi} = \left[\left(\prod_{j=1}^{n} m_{ij}\right)^{1/n}\right] \Big/ \sum_{k=1}^{n}\left[\left(\prod_{j=1}^{n} m_{kj}\right)^{1/n}\right] \tag{4.46}$$

$$a_{kj} \in \lfloor l_{kj}, u_{kj} \rfloor \forall j > k$$

$$a_{jk} = 1/a_{kj} \quad \forall j < k$$

$$a_{jj} = 1$$

kısıtlarına bağlı olarak, kesin matematiksel programlama modeli yoluyla S_{li} *ve* S_{ui}

$$S_{li} = \min \left[\left(\prod_{j=1}^{n} a_{ij}\right)^{1/n}\right] \Big/ \sum_{k=1}^{n}\left[\left(\prod_{j=1}^{n} a_{kj}\right)^{1/n}\right] \tag{4.47}$$

$$S_{ui} = \max \left[\left(\prod_{j=1}^{n} a_{ij}\right)^{1/n}\right] \Big/ \sum_{k=1}^{n}\left[\left(\prod_{j=1}^{n} a_{kj}\right)^{1/n}\right] \tag{4.48}$$

dir.

Örnek 4.7. [Enea ve Piazza, 2004]

Aşağıdaki uygulama van Laarhoven (van Laarhoven ve Pedrcz, 1983) tarafından önerilen bir örnekten türetilmiştir.

Üç üyeden oluşan bir komitenin üç farklı proje (P_1, P_2, P_3) için karar vermek zorunda olduğunu varsayalım. Bu üyeler aşağıdaki şekilde dört farklı kriter belirlerler.

1. Projenin riski (A_1)
2. Projenin maliyeti (A_2)
3. Projenin çevresel etkisi (A_3)
4. Projenin süresi (A_4)

Üyeler bulanık AHP'yi uygulayabilmek için her bir projenin göreceli bir ağırlığını ve bir sıralamasını belirlemek istemektedirler.

Komitenin ilk işi her bir kriterin göreceli önemine karar vermektir. İkili karşılaştırmaları kullanarak A matrisi Tablo 4.19.'daki gibi oluşturulur.

Geometrik ortalamayı kullanarak A matrisinin hücrelerindeki tüm girdiler her bir kriter çifti için komitenin görüşünü ifade eden tek bir bulanık sayı ile yer değiştirilir. Bu işlem sonucunda Tablo 4.20.'de verilen A' matrisi elde edilir.

Chang'ın standart bulanık aritmetiğini uygulayarak ve verilen bağıntıları kullanarak S_{li} değerini hesaplamak için kullanılacak olan kesin değerli $B_i=[b_{jk}]$ matrislerini oluşturabiliriz.

Tablo 4.19. Kriterlerin önemini hesaplamak için kullanılan ikili karşılaştırma matrisi (**A**)

	A_1	A_2	A_3	A_4
A_1	(1,1,1)	(2/3,1,3/2)	(2/3,1,3/2)	(2/7,1/3,2/5)
		(2/5,1/2,2/3)		(2/7,1/3,2/5)
		(3/2,2,5/2)		(2/5,1/2,2/3)
A_2	(2/3,1,3/2)	(1,1,1)	(5/2,3,7/2)	(2/3,1,3/2)
	(3/2,2,5/2)		(5/2,3,7/2)	(2/3,1,3/2)
	(2/5,1/2,2/3)			(3/2,2,5/2)
A_3	(2/3,1,3/2)	(2/7,1/3,2/5)	(1,1,1)	(2/5,1/2,2/3)
		(2/7,1/3,2/5)		
A_4	(5/2,3,7/2)	(2/3,1,3/2)	(3/2,2,5/2)	(1,1,1)
	(5/2,3,7/2)	(2/3,1,3/2)		
	(3/2,2,5/2)	(2/5,1/2,2/3)		

Tablo 4.20. Kriterlerin önemini hesaplamak için kullanılan ikili karşılaştırma matrisi (**A'**) – grup kararları

	A_1	A_2	A_3	A_4
A_1	(1,1,1)	(0.737,1,1.357)	(0.667,1,1.5)	(0.32,0.381,0.474)
A_2	(0.737,1,1.357)	(1,1,1)	(2.5,3,3.5)	(0.873,1.256,1.778)
A_3	(0.667,1,1.5)	(0.286,0.333,0.4)	(1,1,1)	(0.4,0.5,0.667)
A_4	(2.108,2.62,3.128)	(0.562,0.793,1.447)	(1.5,2,2.5)	(1,1,1)

Grup kararları için oluşturulan ikinci karşılaştırma matrisi ilk verilen ikili karşılaştırma matrisindeki bir kriterin diğerine üstünlüğünü belirten karar vericilerin belirttiği değerlerin geometrik ortalaması alınarak elde edildiğini biliyoruz. Buna göre örneğin: A_1'in A_2'ye üsütünlüğünün belirtildiği (2/3,1,3/2) (2/5,1/2,2/3) (3/2,2,5/2) değerlerinden

$$0.737 = \sqrt[3]{\frac{2}{3}.\frac{2}{5}.\frac{3}{2}}, \qquad 1 = \sqrt[3]{1.\frac{1}{2}.2}, \qquad ve \;\; 1.357 = \sqrt[3]{\frac{3}{2}.\frac{2}{3}.\frac{5}{2}}$$

elde edilir.

i=1 için B_1 matrisi

B_1

b_{11}	b_{12}	b_{13}	b_{14}
b_{21}	b_{22}	b_{23}	b_{24}
b_{31}	b_{32}	b_{33}	b_{34}
b_{41}	b_{42}	b_{43}	b_{44}

şekilde olduğuna göre (4.36)-(4.39) kriterlerini kullanarak matrisin elemanlarını bulalım.

(4.36)'ya göre $b_{11} = b_{22} = b_{33} = b_{44} = 1$

(4.37)'ye göre $b_{12} = l_{12} = 0.74$ $(l_{12}, m_{12}, u_{12}) = (0.737, 1, 1.357)$, $b_{13} = 0.67$, $b_{14} = 0.32$

(4.39)'a göre $b_{23} = 3.50 \left(\left\{ x \mid y = \max(2.5 + \frac{1}{2.5}, 3 + \frac{1}{3}, 3.5 + \frac{1}{3.5}) \right\} \right)$, $b_{24} = 1.78$, $b_{34} = 0.40$

(4.38)'e göre $b_{32} = 1/b_{23} = 1 / 3.50 = 0.29$

Son olarak $b_{42} = 0.56$, $b_{43} = 2.50$, $b_{21} = 1.36$, $b_{31} = 1.50$, $b_{41} = 3.13$

Bu değerleri yerine yazarak B_1 matrisi aşağıdaki şekilde oluşur.

Tablo 4.21. B_1 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	0.74	0.67	0.32
A_2	1.36	1.00	3.50	1.78
A_3	1.50	0.29	1.00	0.40
A_4	3.13	0.56	2.50	1.00

Benzer şekilde B_2, B_3, B_4 matrisleri elde edilir.

Tablo 4.22. B_2 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	1.36	0.67	0.32
A_2	0.74	1.00	2.50	0.87
A_3	1.50	0.40	1.00	0.40
A_4	3.13	1.14	2.50	1.00

Tablo 4.23. B_3 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	0.74	1.50	0.32
A_2	1.36	1.00	3.50	1.78
A_3	0.67	0.29	1.00	0.40
A_4	3.13	0.56	2.50	1.00

Tablo 4.24. B_4 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	0.74	0.67	0.47
A_2	1.36	1.00	3.50	1.78
A_3	1.50	0.29	1.00	0.67
A_4	2.11	0.56	1.50	1.00

i = 1 için C_1 matrisi

C_1

c_{11}	c_{12}	c_{13}	c_{14}
c_{21}	c_{22}	c_{23}	c_{24}
c_{31}	c_{32}	c_{33}	c_{34}
c_{41}	c_{42}	c_{43}	c_{44}

şekilde olduğuna göre (4.42)- (4.45) kriterlerini kullanarak matrisin elemanlarını bulalım.

(4.42)’ye göre $c_{11} = c_{22} = c_{33} = c_{44} = 1$

(4.43)’e göre $c_{12} = u_{12} = 1.36$ $(l_{12},m_{12},u_{12}) = (0.737, 1, 1.357)$, $c_{13} = 1.50$, $c_{14} = 0.47$

(4.45)'e göre $c_{23} = 2.50 \left(\left\{ x \mid y = \min(2.5 + \frac{1}{2.5}, 3 + \frac{1}{3}, 3.5 + \frac{1}{3.5}) \right\} \right)$, $b_{24} = 0.87$, $b_{34} = 0.67$

(4.44)'e göre $c_{32} = 1 / c_{23} = 1 / 2.50 = 0.40$

Son olarak $b_{42} = 1.14$, $b_{43} = 1.50$, $b_{21} = 0.74$, $b_{31} = 0.67$, $b_{41} = 2.11$

Bu değerleri yerine yazarak C_1 matrisi aşağıdaki şekilde oluşur.

Tablo 4.25. C_1 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	1.36	1.50	0.47
A_2	0.74	1.00	2.50	0.87
A_3	0.67	0.40	1.00	0.67
A_4	2.11	1.14	1.50	1.00

Benzer şekilde C_2, C_3, C_4 matrisleri elde edilir.

Tablo 4.26. C_2 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	0.74	1.50	0.47
A_2	1.36	1.00	3.50	1.78
A_3	0.67	0.29	1.00	0.67
A_4	2.11	0.56	1.50	1.00

Tablo 4.27. C_3 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	1.36	0.67	0.47
A_2	0.74	1.00	2.50	0.87
A_3	1.50	0.40	1.00	0.67
A_4	2.11	1.14	1.50	1.00

Tablo 4.28. C_4 matrisi

	A_1	A_2	A_3	A_4
A_1	1.00	1.36	1.50	0.32
A_2	0.74	1.00	2.50	0.87
A_3	0.67	0.40	1.00	0.40
A_4	3.13	1.14	2.50	1.00

(4.35), (4.40) ve (4.41) formüllerinden yararlanarak

S_{m1} $= (m_{11}+m_{12}+m_{13}+m_{14})\ /$

$(m_{11}+m_{12}+m_{13}+m_{14}+m_{21}+m_{22}+m_{23}+m_{24}+m_{31}+m_{32}+m_{33}+m_{34}+m_{41}+m_{42}+m_{43}+m_{44})$

= (1+1+1+0.381) / (1+1+1+0.381+1+1+3+1.256+1+0.333+1+0.5+2.62+0.793+2+1)

= 3.381 * 0.053 = 0.179

S_{m2} = 6.256 * 0.053 = 0.331

S_{m3} = 2.833 * 0.053 = 0.150

S_{m4} = 6.413 * 0.053 = 0.340

S_{l1} değerlerini B_1 matrisinden şu şekilde elde ederiz.

S_{l1} $= (b_{11}+b_{12}+b_{13}+b_{14})\ /$

$(b_{11}+b_{12}+b_{13}+b_{14}+b_{21}+b_{22}+b_{23}+b_{24}+b_{31}+b_{32}+b_{33}+b_{34}+b_{41}+b_{42}+b_{43}+b_{44})$

= (1+0.74+0.67+0.32) /

(1+0.74+0.67+0.32+1.36+1+3.5+1.78+1.5+0.29+1+0.40+3.13+0.56+2.5+1)

= 2.73 * 0.048 = 0.131

S_{l2} değerlerini B_2 matrisinden şu şekilde elde ederiz.

S_{l2} $= (b_{21}+b_{22}+b_{23}+b_{24})\ /$

$(b_{11}+b_{12}+b_{13}+b_{14}+b_{21}+b_{22}+b_{23}+b_{24}+b_{31}+b_{32}+b_{33}+b_{34}+b_{41}+b_{42}+b_{43}+b_{44})$

= 5.11 * 0.051 = 0.262

S_{l3} değerlerini B_3 matrisinden şu şekilde elde ederiz.

$$S_{l3} = (b_{31}+b_{32}+b_{33}+b_{34}) / (b_{11}+b_{12}+b_{13}+b_{14}+b_{21}+b_{22}+b_{23}+b_{24}+b_{31}+b_{32}+b_{33}+b_{34}+b_{41}+b_{42}+b_{43}+b_{44})$$
$$= 2.36 * 0.048 = 0.113$$

S_{l4} değerlerini B_4 matrisinden şu şekilde elde ederiz.

$$S_{l4} = (b_{41}+b_{42}+b_{43}+b_{44}) / (b_{11}+b_{12}+b_{13}+b_{14}+b_{21}+b_{22}+b_{23}+b_{24}+b_{31}+b_{32}+b_{33}+b_{34}+b_{41}+b_{42}+b_{43}+b_{44})$$
$$= 5.17 * 0.052 = 0.270$$

S_{u1} değerlerini C_1 matrisinden şu şekilde elde ederiz.

$$S_{u1} = (c_{11}+c_{12}+c_{13}+c_{14}) / (c_{11}+c_{12}+c_{13}+c_{14}+c_{21}+c_{22}+c_{23}+c_{24}+c_{31}+c_{32}+c_{33}+c_{34}+c_{41}+c_{42}+c_{43}+c_{44})$$
$$= (1+1.36+1.5+0.47) / (1+1.36+1.5+0.47+0.74+1+2.50+0.87+0.67+0.40+1+0.67+2.11+1.14+1.50+1)$$
$$= 4.33 * 0.056 = 0.242$$

S_{u2} değerlerini C_2 matrisinden şu şekilde elde ederiz.

$$S_{u2} = (c_{21}+c_{22}+c_{23}+c_{24}) / (c_{11}+c_{12}+c_{13}+c_{14}+c_{21}+c_{22}+c_{23}+c_{24}+c_{31}+c_{32}+c_{33}+c_{34}+c_{41}+c_{42}+c_{43}+c_{44})$$
$$= 7.64 * 0.052 = 0.397$$

S_{u3} değerlerini C_3 matrisinden şu şekilde elde ederiz.

$$S_{u3} = (c_{31}+c_{32}+c_{33}+c_{34}) / (c_{11}+c_{12}+c_{13}+c_{14}+c_{21}+c_{22}+c_{23}+c_{24}+c_{31}+c_{32}+c_{33}+c_{34}+c_{41}+c_{42}+c_{43}+c_{44})$$
$$= 3.57 * 0.056 = 0.199$$

S_{u4} değerlerini C_4 matrisinden şu şekilde elde ederiz.

$$S_{u4} = (c_{41}+c_{42}+c_{43}+c_{44}) / (c_{11}+c_{12}+c_{13}+c_{14}+c_{21}+c_{22}+c_{23}+c_{24}+c_{31}+c_{32}+c_{33}+c_{34}+c_{41}+c_{42}+c_{43}+c_{44})$$

$$= 7.77 * 0.051 = 0.396$$

$S_i = (S_{li}, S_{mi}, S_{ui})$ formülünden yararlanarak

$S_1 = (0.131, 0.179, 0.242)$

$S_2 = (0.262, 0.331, 0.399)$

$S_3 = (0.113, 0.150, 0.199)$

$S_4 = (0.270, 0.340, 0.398)$

olarak düzenlenir.

Kriter ağırlıklarının bulanık tahminleri hesaplandıktan sonra komite her bir kriter için bağımsız olarak üç projeyi karşılaştırmak zorundadır (Tablo 4.29).

Geometrik ortalamayı kullanarak matrislerin hücrelerindeki tüm girdiler her bir proje çiftinin her bir kriter için komite görüşünü ifade eden tek bir bulanık sayı ile yer değiştirir.

Tablolarda bulunan eksik değerleri (missing values) bulmak için aslında bildiğimiz oranları kullanabiliriz. Eğer a_{ij} i. alternatifin *j.* alternatife göre ağırlık oranına w_i/w_j, *i.* alternatifin *k.* alternatife olan ağırlık oranına w_i/w_k ve *j.* alternatifin k. alternatife ağırlık oranına w_j/w_k dersek $a_{ij}=w_i/w_j=(w_i/w_k)*(w_k/w_j)$ olarak elde edilir. Bu metodu uygulayarak uzmanların doğrudan vermediği karşılaştırma değerlerini hesaplayabiliriz. Bu işlem sonucunda Tablo 4.30.'da verilen matris elde edilir.

Bu matrisler bulanık ağırlıkları hesaplamak için, özellikle her bir projenin bulanık ağırlığının her bir kriter için karşılaştırılmasında, kullanılır. Bu tahminler Tablo 4.31.'de gösterilmiştir.

Tablo 4.29. Kriter 1-4 e bağlı olarak Projelerin ikili karşılaştırma matrisleri

	P_1	P_2	P_3
Kriter 1			
P_1	(1,1,1)	(2/3,1,3/2) (2/3,1,3/2)	(2/3,1,3/2) (2/5,1/2,2/3)
P_2	(2/3,1,3/2) (2/3,1,3/2)	(1,1,1)	(2/5,1/2,2/3)
P_3	(2/3,1,3/2) (3/2,2,5/2)	(3/2,2,5/2)	(1,1,1)
Kriter 2			
P_1	(1,1,1)	(5/2,3,7/2)	(3/2,2,5/2)
P_2	(2/7,1/3,2/5)	(1,1,1)	-
P_3	(2/5,1/2,2/3)	-	(1,1,1)
Kriter 3			
P_1	(1,1,1)	(5/2,3,7/2) (3/2,2,5/2)	(5/2,3,7/2)
P_2	(2/7,1/3,2/5) (2/7,1/3,2/5) (2/5,1/2,2/3)	(1,1,1)	(2/3,1,3/2)
P_3	(2/7,1/3,2/5)	(2/3,1,3/2)	(1,1,1)
Kriter 4			
P_1	(1,1,1)	-	(3/2,2,5/2)
P_2	-	(1,1,1)	(3/2,2,5/2)
P_3	(2/5,1/2,2/3) (3/2,2,5/2)	(2/5,1/2,2/3)	(1,1,1)

Tablo 4.30. Kriter 1-4 e bağlı olarak Projelerin ikili karşılaştırma matrisleri için orta değerler.

	P_1	P_2	P_3
Kriter 1			
P_1	(1,1,1)	(0.67,1,1.5)	(0.52,0.71,1)
P_2	(0.67,1,1.5)	(1,1,1)	(0.40,0.50,0.67)
P_3	(1,1.41,1.94)	(1.50,2,2.50)	(1,1,1)
Kriter 2			
P_1	(1,1,1)	(2.50,3,3.50)	(1.50,2,2.50)
P_2	(0.29,0.33,0.40)	(1,1,1)	(1,1.50,2.3)
P_3	(0.40,0.50,0.67)	(0.43,0.67,1)	(1,1,1)
Kriter 3			
P_1	(1,1,1)	(2.11,2.62,3.13)	(2.50,3,3.50)
P_2	(0.32,0.38,0.47)	(1,1,1)	(0.67,1,1.5)
P_3	(0.29,0.33,0.40)	(0.67,1,1.5)	(1,1,1)
Kriter 4			
P_1	(1,1,1)	(0.31,0.5,0.86)	(0.77,1,1.29)
P_2	(1.16,2,3.23)	(1,1,1)	(1.50,2,2.50)
P_3	(0.77,1,1.29)	(0.40,0.50,0.67)	(1,1,1)

Tablo 4.30.'dan yararlanarak her bir kriter için sentetik derece değerlerini hesaplayalım. Bunun için öncelikle B_i ve C_i ($i=1,2,3$) matrislerini oluşturalım.

1. Kriter için ikili karşılaştırmalar matrisi

	P_1	P_2	P_3
P_1	(1,1,1)	(0.67,1,1.5)	(0.52,0.71,1)
P_2	(0.67,1,1.5)	(1,1,1)	(0.40,0.50,0.67)
P_3	(1,1.41,1.94)	(1.50,2,2.50)	(1,1,1)

B_1 matrisi

	P_1	P_2	P_3
P_1	1.00	0.67	0.52
P_2	1.50	1.00	0.40
P_3	1.94	2.50	1.00

B_2 matrisi

	P_1	P_2	P_3
P_1	1.00	1.50	0.52
P_2	0.67	1.00	0.40
P_3	1.94	2.50	1.00

B_3 matrisi

	P_1	P_2	P_3
P_1	1.00	0.67	1.00
P_2	1.50	1.00	0.67
P_3	1.00	1.50	1.00

C_1 matrisi

	P_1	P_2	P_3
P_1	1.00	1.50	1.00
P_2	0.67	1.00	0.67
P_3	1.00	1.50	1.00

C_2 matrisi

	P_1	P_2	P_3
P_1	1.00	0.67	1.00
P_2	1.50	1.00	0.67
P_3	1.00	1.50	1.00

C_3 matrisi

	P_1	P_2	P_3
P_1	1.00	1.50	0.52
P_2	0.67	1.00	0.40
P_3	1.94	2.50	1.00

1. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 2.71 * 0.104 = 0.282

S_{m2} = 2.50 * 0.104 = 0.260

S_{m3} = 4.41 * 0.104 = 0.459

S_{l1} = 2.19 * 0.095 = 0.208

S_{l2} = 2.07 * 0.095 = 0.197

S_{l3} = 3.50 * 0.107 = 0.375

S_{u1} = 3.50 * 0.107 = 0.375

S_{u2} = 3.17 * 0.107 = 0.339

S_{u3} = 5.44 * 0.095 = 0.517

2. Kriter için ikili karşılaştırmalar matrisi

	P_1	P_2	P_3
P_1	(1,1,1)	(2.50,3,3.50)	(1.50,2,2.50)
P_2	(0.29,0.33,0.40)	(1,1,1)	(1,1.50,2.3)
P_3	(0.40,0.50,0.67)	(0.43,0.67,1)	(1,1,1)

B_1 matrisi

	P_1	P_2	P_3
P_1	1.00	2.50	1.50
P_2	0.40	1.00	2.30
P_3	0.67	0.43	1.00

B_2 matrisi

	P_1	P_2	P_3
P_1	1.00	3.50	2.50
P_2	0.29	1.00	1.00
P_3	0.40	1.00	1.00

B_3 matrisi

	P_1	P_2	P_3
P_1	1.00	3.50	2.50
P_2	2.29	1.00	2.30
P_3	0.40	0.43	1.00

C_1 matrisi

	P_1	P_2	P_3
P_1	1.00	3.50	2.50
P_2	0.29	1.00	1.00
P_3	0.40	1.00	1.00

C_2 matrisi

	P_1	P_2	P_3
P_1	1.00	2.50	1.50
P_2	0.40	1.00	2.30
P_3	0.67	0.43	1.00

C_3 matrisi

	P_1	P_2	P_3
P_1	1.00	2.50	1.50
P_2	0.40	1.00	1.00
P_3	0.67	1.00	1.00

2. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 6.00 * 0.091 = 0.545

S_{m2} = 2.83 * 0.091 = 0.257

S_{m3} = 2.17 * 0.091 = 0.197

S_{l1} = 5.00 * 0.093 = 0.463

S_{l2} = 2.27 * 0.086 = 0.196

S_{l3} = 1.83 * 0.069 = 0.127

S_{u1} = 7.00 * 0.086 = 0.599

S_{u2} = 3.70 * 0.093 = 0.343

S_{u3} = 2.67 * 0.099 = 0.265

3. Kriter için ikili karşılaştırmalar matrisi

	P_1	P_2	P_3
P_1	(1,1,1)	(2.11,2.62,3.13)	(2.50,3,3.50)
P_2	(0.32,0.38,0.47)	(1,1,1)	(0.67,1,1.5)
P_3	(0.29,0.33,0.40)	(0.67,1,1.5)	(1,1,1)

B_1 matrisi

	P_1	P_2	P_3
P_1	1.00	2.11	2.50
P_2	0.47	1.00	0.67
P_3	0.40	1.50	1.00

B_2 matrisi

	P_1	P_2	P_3
P_1	1.00	3.13	3.50
P_2	0.32	1.00	0.67
P_3	0.29	1.50	1.00

B_3 matrisi

	P_1	P_2	P_3
P_1	1.00	3.13	3.50
P_2	0.32	1.00	1.50
P_3	0.29	0.67	1.00

C_1 matrisi

	P_1	P_2	P_3
P_1	1.00	3.13	3.50
P_2	0.32	1.00	0.67
P_3	0.29	1.50	1.00

C_2 matrisi

	P_1	P_2	P_3
P_1	1.00	2.11	2.50
P_2	0.47	1.00	1.50
P_3	0.40	0.67	1.00

C_3 matrisi

	P_1	P_2	P_3
P_1	1.00	2.11	2.50
P_2	0.47	1.00	0.67
P_3	0.40	1.50	1.00

3. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 6.62 * 0.088 = 0.584

S_{m2} = 2.38 * 0.088 = 0.210

S_{m3} = 2.33 * 0.088 = 0.206

S_{l1} = 5.61 * 0.094 = 0.527

S_{l2} = 1.99 * 0.081 = 0.160

S_{l3} = 1.96 * 0.081 = 0.158

S_{u1} = 7.63 * 0.081 = 0.615

S_{u2} = 2.97 * 0.094 = 0.279

S_{u3} = 2.90 * 0.094 = 0.272

4. Kriter için ikili karşılaştırmalar matrisi

	P_1	P_2	P_3
P_1	(1,1,1)	(0.31,0.5,0.86)	(0.77,1,1.29)
P_2	(1.16,2,3.23)	(1,1,1)	(1.50,2,2.50)
P_3	(0.77,1,1.29)	(0.40,0.50,0.67)	(1,1,1)

B_1 matrisi

	P_1	P_2	P_3
P_1	1.00	0.31	0.77
P_2	3.23	1.00	2.50
P_3	1.29	0.40	1.00

B_2 matrisi

	P_1	P_2	P_3
P_1	1.00	0.86	0.77
P_2	1.16	1.00	1.50
P_3	1.29	0.67	1.00

B_3 matrisi

	P_1	P_2	P_3
P_1	1.00	0.31	1.29
P_2	3.23	1.00	2.50
P_3	0.77	0.40	1.00

C_1 matrisi

	P_1	P_2	P_3
P_1	1.00	0.86	1.29
P_2	1.16	1.00	1.50
P_3	0.77	0.67	1.00

C_2 matrisi

	P_1	P_2	P_3
P_1	1.00	0.31	0.77
P_2	3.23	1.00	2.50
P_3	1.29	0.40	1.00

C_3 matrisi

	P_1	P_2	P_3
P_1	1.00	0.86	0.77
P_2	1.16	1.00	1.50
P_3	1.29	0.67	1.00

4. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 2.50 * 0.100 = 0.250

S_{m2} = 5.00 * 0.100 = 0.500

S_{m3} = 2.50 * 0.100 = 0.250

S_{l1} = 2.08 * 0.087 = 0.181

S_{l2} = 3.66 * 0.108 = 0.396

S_{l3} = 2.17 * 0.087 = 0.189

S_{u1} = 3.15 * 0.108 = 0.341

S_{u2} = 6.73 * 0.087 = 0.585

S_{u3} = 2.96 * 0.108 = 0.320

Tablo 4.31. Ayrı ayrı her bir kriter altında, bulanık ağırlık tahminleri

	P_1	P_2	P_3
C_1	(0.208,0.281,0.375)	(0.196,0.260,0.339)	(0.375,0.457,0.517)
C_2	(0.462,0.545,0.599)	(0.196,0.258,0.345)	(0.147,0.197,0.265)
C_3	(0.527,0.584,0.615)	(0.160,0.210,0.275)	(0.157,0.206,0.272)
C_4	(0.181,0.250,0.341)	(0.396,0.500,0.588)	(0.189,0.250,0.320)

Her bir projenin bulanık ağırlıklarını, o projeye karşılık gelen kriterin bulanık ağırlığıyla çarparak, topladığımızda sonuç skorlarını hesaplayabiliriz. Bu sonuçlar Tablo 4.32'de gösterilmiştir.

Tablo 4.32'i oluşturmak için 1. proje için sonuç skorlarını hesaplayalım. Bunun için daha önce bulduğumuz

S_1 = (0.131, 0.179, 0.242)

S_2 = (0.262, 0.331, 0.399)

S_3 = (0.113, 0.150, 0.199)

S_4 = (0.270, 0.340, 0.398)

sentetik derece değerlerini Tablo 4.31'deki değerlerle çarpalım.

0.208*0.131+0.462*0.262+0.527*0.113+0.181*0.270 = 0.257

0.281*0.179+0.545*0.331+0.584*0.150+0.250*0.340 = 0.404

0.375*0.242+0.599*0.399+0.615*0.199+0.341*0.398 = 0.585

Benzer şekilde diğer projeler için de sonuç skorlarını hesaplayabiliriz.

Tablo 4.32. 1.Metoda göre projelerin bulanık sonuç değerleri.

	P_1	P_2	P_3
Sonuç skoru	(0.257,0.404,0.585)	(0.202,0.333,0.507)	(0.157,0.263,0.409)

Metod II

Burada özel bir matris yapısına gerek yoktur. A' matrisinden başlayıp, Lootsma'nın (1997) geometrik ortalama metodunu uygulayarak ve kısıtları göz önünde bulundurarak, $S_i=(S_{li},S_{mi},S_{ui})$ bulanık sentetik derece değerlerini hesaplayabiliriz.

Dolayısıyla (4.46), (4.47) ve (4.48) formüllerini kullanarak

S_1 = (0.142, 0.185, 0.239)
S_2 = (0.262, 0.328, 0.396)
S_3 = (0.118, 0.150, 0.194)
S_4 = (0.269, 0.336, 0.400)

değerleri bulunur.

Şimdi bu değerleri nasıl elde ettiğimizi ayrıntılı olarak açıklayalım.

(4.46)'dan i = 1 için

$$S_{m1} = \frac{(m_{11}.m_{12}.m_{13}.m_{14})^{1/4}}{(m_{11}.m_{12}.m_{13}.m_{14})^{1/4} + (m_{21}.m_{22}.m_{23}.m_{24})^{1/4} + (m_{31}.m_{32}.m_{33}.m_{34})^{1/4} + (m_{41}.m_{42}.m_{43}.m_{44})^{1/4}}$$

$$S_{m1} = \frac{\sqrt[4]{(1.1.1.0,381)}}{\sqrt[4]{(1.1.1.0,381} + \sqrt[4]{1.1.3.1,256} + \sqrt[4]{1.0,333.1.0,5} + \sqrt[4]{2,62.0,793.2.1)}} = 0,185$$

benzer şekilde sırasıyla i = 2, 3 ve 4 için

$S_{m2} = 0{,}328$

$S_{m3} =$ 0,150

$S_{m4} =$ 0.336

değerleri bulunur.

(4.47)'den S_{li} değerini B matrislerinden şu şekilde elde edebiliriz.

$$S_{li} = \frac{\min\left[\left(B_i b_{i1} * B_i b_{i2} * B_i b_{i3} * B_i b_{i4}\right)^{1/4}\right]}{\sum_{j=1}^{4}\left(B_i b_{ij} * B_i b_{ij} * B_i b_{ij} * B_i b_{ij}\right)^{1/4}}$$

S_{l1} değerini bulmak için tüm B matrislerinin birinci satırının geometrik ortalamasının minimum olanını seçip bu minimum değeri aynı matristeki tüm satırların geometrik ortalamalarının toplamına bölmeliyiz.

$$S_{l1} = \frac{\min\begin{bmatrix}(1*0.74*0.67*0.32)^{1/4}, \\ (1*1.36*0.67*0.32)^{1/4}, \\ (1*0.74*1.5*0.32)^{1/4}, \\ (1*0.74*0.67*0.47)^{1/4}\end{bmatrix}}{\begin{bmatrix}\left(1*0.74*0.67*0.32\right)^{1/4} + \left(1.36*1*3.5*1.78\right)^{1/4} + \left(1.5*0.29*1*0.40\right)^{1/4} \\ + \left(3.13*0.56*2.5*1\right)^{1/4}\end{bmatrix}}$$

Bu şekilde

S_{l1} = 0.142

S_{l2} = 0.262

S_{l3} = 0.118

$S_{l4} = 0.269$

elde edilir.

(4.48)'den S_{ui} değerini C matrislerinden şu şekilde elde edebiliriz.

$$S_{ui} = \frac{\max\left[\left(C_i c_{i1} * C_i c_{i2} * C_i c_{i3} * C_i c_{i4}\right)^{1/4}\right]}{\sum_{j=1}^{4}\left(C_i c_{ij} * C_i c_{ij} * C_i c_{ij} * C_i c_{ij}\right)^{1/4}}$$

S_{u1} değerini bulmak için tüm C matrislerinin birinci satırlarının geometrik ortalamalarının maksimum olanını seçip bu maksimum değeri aynı matristeki tüm satırların geometrik ortalamalarının toplamına bölmeliyiz.

$$S_{u1} = \frac{\max\begin{bmatrix}(1*1.36*1.50*0.47)^{1/4}, \\ (1*0.74*1.50*0.47)^{1/4}, \\ (1*1.36*0.67*0.47)^{1/4}, \\ (1*1.36*1.50*0.32)^{1/4}\end{bmatrix}}{\begin{bmatrix}(1*1.36*1.50*0.47)^{1/4} + (0.74*1*2.5*0.87)^{1/4} + (0.675*0.40*1*0.67)^{1/4} \\ + (2.11*1.14*1.5*1)^{1/4}\end{bmatrix}}$$

Bu şekilde

$S_{u1} = 0.239$

$S_{u2} = 0.396$

$S_{u3} = 0.194$

$S_{u4} = 0.400$

elde edilir.

Aynı yolla her bir projenin bulanık ağırlıklarını, her bir kriter için karşılaştırarak hesaplayabiliriz.

2. metod kullanılarak hesaplanan her bir kriterin, projenin bulanık ağırlık tahminlerine oranı Tablo 4.33.'te verilmiştir.

Son olarak her bir projenin bulanık ağırlıklarını, o projeye karşılık gelen kriterin bulanık ağırlığıyla çarparak, topladığımızda sonuç skorlarını hesaplayabiliriz. Elde edilen sonuçlar Tablo 4.34.'te verilmiştir.

Tablo 4.30.'dan ve B_i ile C_i ($i=1,2,3$) matrislerinden yararlanarak her bir kriter için sentetik derece değerlerini ayrıntılı olarak aşağıda hesaplayalım.

1. kriter için elde edilen sentetik derece değerleri:

$$S_{m1} = \frac{\sqrt[3]{(1.1.0{,}71)}}{\sqrt[3]{(1.1.0{,}71} + \sqrt[3]{1.1.0{,}50} + \sqrt[3]{1{,}41.2.1)}} = 0{,}288$$

S_{m2} = 0.794 / 3.099 = 0.256

S_{m3} = 1.413 / 3.099 = 0.456

$$S_{l1} = \frac{\min\left[(1*0.67*0.52)^{1/3}, (1*1.50*0.52)^{1/3}, (1*0.67*1)^{1/3}\right]}{\left[(1*0.67*0.52)^{1/3} + (1.50*1*0.40)^{1/3} + (1.94*2.50*1)^{1/3}\right]} = \frac{0.704}{3.240} = 0.217$$

S_{l2} = 0.645 / 3.259 = 0.198

S_{l3} = 1.145 / 3.018 = 0.379

$$S_{u1} = \frac{\max\left[(1*1.50*1)^{1/3}, (1*0.67*1)^{1/3}, (1*1.50*0.52)^{1/3}\right]}{\left[(1*1.50*1)^{1/3} + (0.67*1*0.67)^{1/3} + (1*1.50*1)^{1/3}\right]} = \frac{1.145}{3.053} = 0.375$$

S_{u2} = 1 / 3.019 = 0.331

S_{u3} = 1.693 / 3.259 = 0.519

2. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 1.817 / 3.303 = 0.550

S_{m2} = 0.240

S_{m3} = 0.210

S_{l1} = 1.554 / 2.762 = 0.562

S_{l2} = 0.662 / 3.460 = 0.191

S_{l3} = 0.556 / 3.491 = 0.159

S_{u1} = 2.061 / 3.46 = 0.595

S_{u2} = 0.973 / 3.187 = 0.305

S_{u3} = 0.875 / 3.166 = 0.277

3. kriter için elde edilen sentetik derece değerleri:

S_{m1} = 1.988 / 3.403 = 0.584

S_{m2} = 0.724 / 3.403 = 0.213

S_{m3} = 0.691 / 3.403 = 0.203

$S_{l1} = 1.741 / 3.264 = 0.533$

$S_{l2} = 0.599 / 3.578 = 0.167$

$S_{l3} = 0.579 / 3.583 = 0.162$

$S_{u1} = 2.221 / 3.578 = 0.621$

$S_{u2} = 0.890 / 3.276 = 0.272$

$S_{u3} = 0.843 / 3.264 = 0.258$

4. kriter için elde edilen sentetik derece değerleri:

$S_{m1} = 0.794 / 3.166 = 0.250$

$S_{m2} = 1.587 / 3.166 = 0.500$

$S_{m3} = 0.794 / 3.166 = 0.250$

$S_{l1} = 0.620 / 3.428 = 0.181$

$S_{l2} = 1.203 / 3.028 = 0.397$

$S_{l3} = 0.675 / 3.418 = 0.197$

$S_{u1} = 1.035 / 2.754 = 0.376$

$S_{u2} = 2.006 / 3.428 = 0.585$

$S_{u3} = 0.953 / 3.028 = 0.315$

Tablo 4.33. Ayrı ayrı her bir kriter altında, projelerin bulanık ağırlık tahminleri

	P_1	P_2	P_3
C_1	(0.245,0.300,0.365)	(0.229,0.275,0.332)	(0.365,0.425,0.475)
C_2	(0.450,0.494,0.530)	(0.225,0.266,0.316)	(0.197,0.240,0.292)
C_3	(0.481,0.520,0.551)	(0.205,0.244,0.291)	(0.199,0.236,0.280)
C_4	(0.216,0.272,0.340)	(0.381,0.457,0.522)	(0.231,0.272,0.319)

Tablo 4.34.'ü oluşturmak için 1. proje için sonuç skorlarını hesaplayalım. Bunun için daha önce bulduğumuz

$S_1 = (0.142, 0.185, 0.239)$

$S_2 = (0.262, 0.328, 0.396)$

$S_3 = (0.118, 0.150, 0.194)$

$S_4 = (0.269, 0.336, 0.400)$

sentetik derece değerlerini Tablo 4.33.'teki değerlerle çarpalım.

$0.217*0.142+0.462*0.262+0.533*0.118+0.181*0.269 = 0.263$

$0.287*0.185+0.550*0.328+0.584*0.150+0.250*0.336 = 0.405$

$0.375*0.239+0.595*0.396+0.621*0.194+0.376*0.400 = 0.596$

Benzer şekilde diğer projeler için de sonuç skorlarını hesaplayabiliriz.

Tablo 4.34. II.Metoda göre adayların üç proje için bulanık sonuç skorları.

	P_1	P_2	P_3
Sonuç skoru	(0.267,0.387,0.542)	(0.218,0.328,0.473)	(0.189,0.284,0.413)

5. ENEA VE PİAZZA YAKLAŞIMININ BİLGİSAYAR PROGRAMLAMASI

Burada java programlama dili kullanılarak Enea ve Piazza'nın I. metodunun çözüm yöntemi kodlanmıştır. Program öncelikle problemde kaç kriter, kaç aday ve kaç karar vericinin olduğunu sorgulayıp girilen değerlere göre ekrana bir tablo getirmektedir. Tablodaki hücrelere tek tek tıklayıp girilerek hazır üçgensel sayı formatındaki boş yerlere ondalıklı değerler yazılır. Programda kayıp değerlerin yerine de manuel olarak değerlerin hesaplanıp girilmesi gereklidir. Birinci tabloda karşılaştırma değerleri girildikten sonra aşağıdaki >> butonuyla sonraki tablolara ve en sonunda da sonuç skorlarına ulaşılır.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

import com.borland.jbcl.layout.*;

/**
 * <p>Title: </p>
 *
 * <p>Description: </p>
 *
 * <p>Copyright: Copyright (c) 2006</p>
 *
 * <p>Company: </p>
 *
 * @author not attributable
 * @version 1.0
 */
public class FDialog
    extends JDialog {

  FuzzyNumber number=null;
  JPanel panel1 = new JPanel();
  JLabel jLabel1 = new JLabel();
  JTextField jTextField1 = new JTextField();
  JLabel jLabel2 = new JLabel();
  JLabel jLabel3 = new JLabel();
  JTextField jTextField2 = new JTextField();
  JTextField jTextField3 = new JTextField();
  JLabel jLabel4 = new JLabel();
  JPanel jPanel1 = new JPanel();
  GridBagLayout gridBagLayout1 = new GridBagLayout();
  JPanel jPanel2 = new JPanel();
  JButton jButton1 = new JButton();
  JButton jButton2 = new JButton();
```

```
GridBagLayout gridBagLayout2 = new GridBagLayout();
GridBagLayout gridBagLayout3 = new GridBagLayout();
public FDialog(Frame owner, String title, boolean modal) {
  super(owner, title, modal);
  try {
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    jbInit();
    pack();
    setModal(true);
  }
  catch (Exception exception) {
    exception.printStackTrace();
  }
}

public FDialog() {
  this(new Frame(), "FuzzyNumbeDialog", false);
}

private void jbInit() throws Exception {
  panel1.setLayout(gridBagLayout2);
  jLabel1.setText("(");
  jLabel2.setText(",");
  jLabel3.setText(",");
  jLabel4.setText(")");
  jPanel1.setBorder(BorderFactory.createEtchedBorder());
  jPanel1.setLayout(gridBagLayout1);
  jPanel2.setLayout(gridBagLayout3);
  jButton1.setText("OK");
  jButton1.addActionListener(new FDialog_jButton1_actionAdapter(this));
  jButton2.setText("Cancel");
  jButton2.addActionListener(new FDialog_jButton2_actionAdapter(this));
  this.getContentPane().add(panel1, java.awt.BorderLayout.CENTER);
  jPanel1.add(jTextField1, new GridBagConstraints(1, 0, 1, 1, 1.0, 0.0
      , GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL,
      new Insets(11, 0, 15, 0), 38, 5));
  jPanel1.add(jTextField3, new GridBagConstraints(5, 0, 1, 1, 1.0, 0.0
      , GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL,
      new Insets(11, 0, 15, 0), 38, 5));
  jPanel1.add(jLabel4, new GridBagConstraints(6, 0, 1, 1, 0.0, 0.0
                                   , GridBagConstraints.WEST,
                                   GridBagConstraints.NONE,
                                   new Insets(11, 10, 15, 20), 11,
                                   10));
  jPanel1.add(jLabel3, new GridBagConstraints(4, 0, 1, 1, 1.0, 1.0
                                   , GridBagConstraints.WEST,
                                   GridBagConstraints.NONE,
                                   new Insets(11, 0, 15, 0), 8, 10));
  jPanel1.add(jLabel1, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0
                                   , GridBagConstraints.WEST,
                                   GridBagConstraints.NONE,
```

```
                              new Insets(11, 39, 15, 0), 8,
                              10));
  jPanel1.add(jLabel2, new GridBagConstraints(2, 0, 1, 1, 1.0, 1.0
                              , GridBagConstraints.WEST,
                              GridBagConstraints.NONE,
                              new Insets(11, 6, 15, 6), 7, 10));
  jPanel1.add(jTextField2, new GridBagConstraints(3, 0, 1, 1, 1.0, 0.0
     , GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL,
     new Insets(11, -6, 15, 6), 38, 5));
  panel1.add(jPanel1, new GridBagConstraints(0, 0, 1, 1, 1.0, 1.0
                              , GridBagConstraints.CENTER,
                              GridBagConstraints.BOTH,
                              new Insets(0, 0, 0, 67), 74, 0));
  panel1.add(jPanel2, new GridBagConstraints(0, 1, 1, 1, 1.0, 1.0
                              , GridBagConstraints.CENTER,
                              GridBagConstraints.BOTH,
                              new Insets(0, 0, 179, 67), 50,
                              29));
  jPanel2.add(jButton1, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0
                              , GridBagConstraints.CENTER,
                              GridBagConstraints.NONE,
                              new Insets(12, 55, 29, 0), 38,
                              0));
  jPanel2.add(jButton2, new GridBagConstraints(1, 0, 1, 1, 0.0, 0.0
                              , GridBagConstraints.CENTER,
                              GridBagConstraints.NONE,
                              new Insets(12, 54, 29, 50), 20,
                              0));
}
FuzzyNumber getNumber(){
  return number;
}

public void jButton1_actionPerformed(ActionEvent e) {
  this.setVisible(false);
  double l=Double.parseDouble(jTextField1.getText());
  double m=Double.parseDouble(jTextField2.getText());
  double u=Double.parseDouble(jTextField3.getText());
  number = new FuzzyNumber(l,m,u);
}

public void jButton2_actionPerformed(ActionEvent e) {
  number = null;
  this.setVisible(false);
}


public void setNumber(FuzzyNumber number) {
  if(number==null){
    jTextField1.setText("");
    jTextField2.setText("");
```

```
      jTextField3.setText("");


    }else{
    this.number = number;
    jTextField1.setText(""+number.l);
    jTextField2.setText(""+number.m);
    jTextField3.setText(""+number.u);
  }
  }

}

class FDialog_jButton1_actionAdapter
    implements ActionListener {
  private FDialog adaptee;
  FDialog_jButton1_actionAdapter(FDialog adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton1_actionPerformed(e);
  }
}

class FDialog_jButton2_actionAdapter
    implements ActionListener {
  private FDialog adaptee;
  FDialog_jButton2_actionAdapter(FDialog adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton2_actionPerformed(e);
  }
}


import java.awt.*;
import javax.swing.*;
import javax.swing.BorderFactory;
import java.awt.BorderLayout;
import com.borland.jbcl.layout.XYLayout;
import com.borland.jbcl.layout.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 * <p>Title: </p>
 *
 * <p>Description: </p>
```

```
 *
 * <p>Copyright: Copyright (c) 2006</p>
 *
 * <p>Company: </p>
 *
 * @author not attributable
 * @version 1.0
 */
public class Frame1
    extends JFrame {
  int index = 0;
  int aday = 3;
  int criter = 2;
  int karar = 3;

  BorderLayout borderLayout1 = new BorderLayout();
  JPanel jPanel1 = new JPanel();
  JButton jButton1 = new JButton();
  JButton jButton2 = new JButton();
  FuzzyInputPane[] panel; // = new FuzzyInputPane(5, 3);
  CardLayout cl = new CardLayout();
  JPanel cntr = new JPanel(cl);
  Object[] syms;

  public Frame1(int a,int c,int k) {
    aday=a;
    criter=c;
    karar=k;
    try {
      jbInit();
    }
    catch (Exception exception) {
      exception.printStackTrace();
    }
  }

  private void jbInit() throws Exception {

    panel = new FuzzyInputPane[criter + 1];
    panel[0] = new FuzzyInputPane("criter", criter + 1, karar);
    syms = new Object[criter + 1];
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    cntr.add(panel[0], "0");
    for (int i = 1; i < panel.length; i++) {
      panel[i] = new FuzzyInputPane("criter " + i, aday + 1, karar);
      cntr.add(panel[i], "" + i);
    }

    this.getContentPane().setLayout(borderLayout1);
    jButton1.setText(">>");
    jButton1.addActionListener(new Frame1_jButton1_actionAdapter(this));
```

```
    jButton2.setText("<<");
    jButton2.addActionListener(new Frame1_jButton2_actionAdapter(this));
    getContentPane().add(cntr, BorderLayout.CENTER);
    this.getContentPane().add(jPanel1, java.awt.BorderLayout.SOUTH);
    jPanel1.add(jButton2, null);
    jPanel1.add(jButton1, null);
  }

  public static void main(String[] args) {
    Frame1 frame1 = new Frame1(3,2,3);
    frame1.setTitle("FUZZY");
    frame1.setBounds(200, 200, 400, 400);
    frame1.setVisible(true);
  }

  public static FuzzyNumber[] getSymNumber(FuzzyNumber[][] mtrx) {
    Object[] b = new Object[mtrx.length];
    Object[] c = new Object[mtrx.length];
    for (int i = 0; i < mtrx.length; i++) {
      b[i] = getB_Mtrx(i, mtrx);
      c[i] = getC_Mtrx(i, mtrx);
    }
    FuzzyNumber[] s = new FuzzyNumber[mtrx.length];

    double sum_b = 0;
    double sum_c = 0;
    double summ = 0;
    for (int i = 0; i < s.length; i++) {
      for (int j = 0; j < s.length; j++) {
        summ += mtrx[i][j].m;
      }
    }
    double sumt = 0;

    for (int i = 0; i < s.length; i++) {
      sumt = 0;
      for (int j = 0; j < s.length; j++) {
        sumt += mtrx[i][j].m;
      }
      s[i] = new FuzzyNumber();
      s[i].m = sumt / summ;
    }
    for (int i = 0; i < s.length; i++) {
      sum_b = sum( (double[][]) b[i]);
      sum_c = sum( (double[][]) c[i]);

      s[i].l = sum( ( (double[][]) b[i])[i]) / sum_b;
      s[i].u = sum( ( (double[][]) c[i])[i]) / sum_c;

    }
```

```
  return s;

}

public FuzzyNumber[] finalResult(Object[] synh) {
  FuzzyNumber[] result=new FuzzyNumber[aday];
  for (int i = 0; i < aday; i++) {
    result[i]=new FuzzyNumber(0,0,0);
    for (int j = 1; j < synh.length; j++) {
      result[i].l+=((FuzzyNumber[])synh[0])[j-1].l*((FuzzyNumber[])synh[j])[i].l;
      result[i].m+=((FuzzyNumber[])synh[0])[j-1].m*((FuzzyNumber[])synh[j])[i].m;
      result[i].u+=((FuzzyNumber[])synh[0])[j-1].u*((FuzzyNumber[])synh[j])[i].u;
    }
  }
  return result;
}
public void jButton1_actionPerformed(ActionEvent e) {

  FuzzyNumber[][] mtrx = panel[index].getFuzzyMatrix();
  syms[index] = getSymNumber(mtrx);
  index = index + 1;
  if (index <= criter)
    cl.show(cntr, "" + index);
  else{
    FuzzyNumber [] res=finalResult(syms);
    ResultPanel p=new ResultPanel(res,2,aday+1);
    cntr.add(p,"result");
    cl.show(cntr,"result");
  }


}

public void jButton2_actionPerformed(ActionEvent e) {
  cl.previous(cntr);

}

public static double sum(double[][] m) {
  double sum = 0;
  for (int i = 0; i < m.length; i++) {
    for (int j = 0; j < m[i].length; j++) {
      sum += m[i][j];
    }
  }
  return sum;
}

public static double sum(double[] m) {
  double sum = 0;
  for (int i = 0; i < m.length; i++) {
```

```
    sum += m[i];
  }
  return sum;
}

static double[][] getB_Mtrx(int r, FuzzyNumber[][] mtrx) {
  double[][] result = new double[mtrx.length][mtrx.length];
  for (int i = 0; i < mtrx.length; i++) {
    for (int j = i + 1; j < mtrx[i].length; j++) {
      /*double m = max(mtrx[i][j].l + 1 / mtrx[i][j].l,
                    mtrx[i][j].m + 1 / mtrx[i][j].m,
                    mtrx[i][j].u + 1 / mtrx[i][j].u);*/
      double m = max2(mtrx[i][j].l, mtrx[i][j].m, mtrx[i][j].u);
      result[i][j] = m;
      result[j][i] = 1 / m;

    }
  }
  for (int i = 0; i < mtrx.length; i++) {
    result[r][i] = min(mtrx[r][i].l, mtrx[r][i].m, mtrx[r][i].u);
    result[i][r] = 1 / result[r][i];
    result[i][i] = 1;
  }
  return result;
}

static double[][] getC_Mtrx(int r, FuzzyNumber[][] mtrx) {
  double[][] result = new double[mtrx.length][mtrx.length];
  for (int i = 0; i < mtrx.length; i++) {
    for (int j = i + 1; j < mtrx[i].length; j++) {
      double m = min2(mtrx[i][j].l, mtrx[i][j].m, mtrx[i][j].u);

      result[i][j] = m;
      result[j][i] = 1 / m;

    }
  }
  for (int i = 0; i < mtrx.length; i++) {
    result[r][i] = max(mtrx[r][i].l, mtrx[r][i].m, mtrx[r][i].u);
    result[i][r] = 1 / result[r][i];
    result[i][i] = 1;
  }
  return result;
}

static double max(double a, double b, double c) {
  double max = a;
  if (b > max)
    max = b;
  if (c > max)
    max = c;
```

```
    return max;
  }

  static double min(double a, double b, double c) {
    double min = a;
    if (b < min)
      min = b;
    if (c < min)
      min = c;
    return min;
  }

  static double max2(double a, double b, double c) {
    double a2 = a + 1 / a;
    double b2 = b + 1 / b;
    double c2 = c + 1 / c;
    double max2 = a2;
    double max = a;
    if (b2 > max2) {
      max = b;
      max2 = b2;
    }
    if (c2 > max2) {
      max = c;
    }
    return max;
  }

  static double min2(double a, double b, double c) {
    double a2 = a + 1 / a;
    double b2 = b + 1 / b;
    double c2 = c + 1 / c;
    double min2 = a2;
    double min = a;
    if (b2 < min2) {
      min = b;
      min2 = b2;
    }
    if (c2 < min2) {
      min = c;
    }
    return min;
  }

}

class Frame1_jButton2_actionAdapter
    implements ActionListener {
  private Frame1 adaptee;
  Frame1_jButton2_actionAdapter(Frame1 adaptee) {
    this.adaptee = adaptee;
```

```
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton2_actionPerformed(e);
  }
}

class Frame1_jButton1_actionAdapter
    implements ActionListener {
  private Frame1 adaptee;
  Frame1_jButton1_actionAdapter(Frame1 adaptee) {
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton1_actionPerformed(e);
  }
}


import java.awt.*;
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;


public class FuzzyInputPane
    extends JPanel implements MouseListener {
  public FuzzyInputPane() {

  }

  static FDialog dialog = new FDialog();

  int dim;
  int level;
  FuzzyNumber[][][] numbers;
  FzyLabel[][][] lbls;
  public FuzzyInputPane(String name, int dim, int level) {
    setName(name);
    this.dim = dim;
    this.level = level;
    numbers = new FuzzyNumber[dim - 1][dim - 1][level];
    lbls = new FzyLabel[dim-1][dim-1][level];
    setLayout(new GridLayout(dim, dim));
    try {
      jbInit();
    }
```

```
  catch (Exception ex) {
    ex.printStackTrace();
  }
}

private void jbInit() throws Exception {

  for (int i = 0; i < dim; i++) {
    for (int j = 0; j < dim; j++) {
      JPanel label = new JPanel();
      Border b = BorderFactory.createLineBorder(Color.black);
      label.setBorder(b);
      add(label);
      if (i == 0 && j == 0) {
        JLabel l = new JLabel(getName());
        l.setHorizontalAlignment(JLabel.CENTER);
        label.setLayout(new BorderLayout());
        label.add(l);
      }
      if (i == 0 && j != 0) {
        JLabel l = new JLabel(getName().equals("criter") ? "A" + j : "P" + j);
        l.setHorizontalAlignment(JLabel.CENTER);
        label.setLayout(new BorderLayout());
        label.add(l);

      }
      if (j == 0 && i != 0) {
        JLabel l = new JLabel(getName().equals("criter") ? "A" + i : "P" + i);
        l.setHorizontalAlignment(JLabel.CENTER);
        label.setLayout(new BorderLayout());
        label.add(l);

      }
      if (i > 0 && j > 0) {
        if (i != j)
          for (int k = 0; k < level; k++) {

            FzyLabel label2 = new FzyLabel(i, j, k);
            lbls[i-1][j-1][k]=label2;
            Border b2 = BorderFactory.createLineBorder(Color.GRAY, 1);
            label2.setBorder(b2);
            label.setLayout(new GridLayout(level, 1));
            label2.addMouseListener(this);
            label2.setHorizontalAlignment(JLabel.CENTER);
            label2.setVerticalAlignment(JLabel.CENTER);
            label.add(label2);
          }
        else {
          numbers[i - 1][j - 1][0] = new FuzzyNumber();
          label.setLayout(new BorderLayout());
          JLabel l = new JLabel(numbers[i - 1][j - 1][0].toString());
```

```
        l.setHorizontalAlignment(JLabel.CENTER);
        label.add(l);

      }
    }
  }
}

}

public FuzzyNumber[][] getFuzzyMatrix() {
 FuzzyNumber[][] matrix = new FuzzyNumber[dim - 1][dim - 1];

 for (int i = 0; i < dim - 1; i++) {
  for (int j = 0; j < dim - 1; j++) {
    int count = 0;
    if (i != j) {
     double l = 1, m = 1, u = 1;
     for (int k = 0; k < level; k++) {
      if (numbers[i][j][k] != null) {
        l *= numbers[i][j][k].l;
        m *= numbers[i][j][k].m;
        u *= numbers[i][j][k].u;
        count++;
      }

     }
     if (count != 0) {
      l = Math.pow(l, 1.0 / count);
      m = Math.pow(m, 1.0 / count);
      u = Math.pow(u, 1.0 / count);
      matrix[i][j] = new FuzzyNumber(l, m, u);
     }
    }
    else
     matrix[i][j] = numbers[i][j][0];
  }

 }
 return matrix;
}

public void mouseClicked(MouseEvent e) {
 FzyLabel lbl = (FzyLabel) e.getSource();
 FzyLabel inv = lbls[lbl.y - 1][lbl.x - 1][lbl.z];
 dialog.setNumber(numbers[lbl.x - 1][lbl.y - 1][lbl.z]);

 dialog.setLocation(e.getX(), e.getY());
 dialog.setVisible(true);
 FuzzyNumber nmr = dialog.getNumber();
 if (nmr == null)
```

```
    ;
   //lbl.setText("");
   else {
     lbl.setText(nmr.toString());
     inv.setText(nmr.inverse().toString());
     numbers[lbl.x - 1][lbl.y - 1][lbl.z] = nmr;
     numbers[lbl.y - 1][lbl.x - 1][lbl.z] = nmr.inverse();
   }

  }

  public void mousePressed(MouseEvent e) {
  }

  public void mouseReleased(MouseEvent e) {
  }

  public void mouseEntered(MouseEvent e) {
  }

  public void mouseExited(MouseEvent e) {
  }

}


/**
 * <p>Title: </p>
 *
 * <p>Description: </p>
 *
 * <p>Copyright: Copyright (c) 2006</p>
 *
 * <p>Company: </p>
 *
 * @author not attributable
 * @version 1.0
 */
public class FuzzyNumber {
   double l;
   double m;
   double u;
  public FuzzyNumber() {
    l=1;
    u=1;
    m=1;
  }
  public FuzzyNumber(double l,double m, double u) {
    this.l=l;this.m=m;this.u=u;
  }
  public void setL(double l) {
```

```
    this.l = l;
  }

  public void setM(double m) {
    this.m = m;
  }

  public void setU(double u) {
    this.u = u;
  }

  public double getL() {
    return l;
  }

  public double getM() {
    return m;
  }

  public double getU() {
    return u;
  }
  public FuzzyNumber inverse(){
    return new FuzzyNumber(1/u,1/m,1/l);
  }
  public String toString(){
    //return "( " + l + ", " + m + ", " + u +")";
    return String.format("( %4.3f , %4.3f , %4.3f )",l,m,u);
  }
}


import javax.swing.*;

/**
 * <p>Title: </p>
 *
 * <p>Description: </p>
 *
 * <p>Copyright: Copyright (c) 2006</p>
 *
 * <p>Company: </p>
 *
 * @author not attributable
 * @version 1.0
 */
public class FzyLabel
    extends JLabel {
  public int x,y,z;
  public FzyLabel(int x,int y, int z) {
    this.x=x;
```

```
    this.y=y;
    this.z=z;
    //setText("" + x + "," + y + "," + z);
  }

}


import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.XYLayout;
import com.borland.jbcl.layout.*;
import javax.swing.BorderFactory;
import java.awt.Color;

/**
 * <p>Title: </p>
 *
 * <p>Description: </p>
 *
 * <p>Copyright: Copyright (c) 2006</p>
 *
 * <p>Company: </p>
 *
 * @author not attributable
 * @version 1.0
 */
public class ResultPanel
    extends JPanel {
  GridLayout gridLayout1;
  int c;
  FuzzyNumber[] number;
  public ResultPanel(FuzzyNumber[] number, int r, int c) {
    this.c = c;
    this.number = number;
    gridLayout1 = new GridLayout(r, c);
    try {
      jbInit();
    }
    catch (Exception exception) {
      exception.printStackTrace();
    }
  }

  private void jbInit() throws Exception {
    this.setLayout(gridLayout1);
    add(new JLabel());
    for (int i = 1; i < c; i++) {
      JLabel l = new JLabel("P" + i);
      l.setHorizontalAlignment(JLabel.CENTER);
      add(l);
```

```
    }
    add(new JLabel("Final Result"));
    for (int i = 1; i < c; i++) {
      JLabel l = new JLabel(number[i-1].toString());
      l.setHorizontalAlignment(JLabel.CENTER);
      add(l);
    }

  }
}


import java.awt.*;
import javax.swing.*;
import com.borland.jbcl.layout.XYLayout;
import com.borland.jbcl.layout.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SettingFrame
    extends JFrame {
  JLabel jLabel1 = new JLabel();
  JLabel jLabel2 = new JLabel();
  JLabel jLabel3 = new JLabel();
  JTextField jTextField1 = new JTextField();
  JTextField jTextField2 = new JTextField();
  JTextField jTextField3 = new JTextField();
  JButton jButton1 = new JButton();
  GridBagLayout gridBagLayout1 = new GridBagLayout();
  public SettingFrame() {
    try {
      jbInit();
    }
    catch (Exception exception) {
      exception.printStackTrace();
    }
  }

  private void jbInit() throws Exception {
    getContentPane().setLayout(gridBagLayout1);
    jLabel1.setText("Aday");
    jLabel3.setText("Karar verici");
    jLabel2.setText("Kriter");
    jButton1.setText("OK");
    jButton1.addActionListener(new SettingFrame_jButton1_actionAdapter(this));
    this.getContentPane().add(jButton1,
                              new GridBagConstraints(0, 3, 2, 1, 0.0, 0.0
        , GridBagConstraints.CENTER, GridBagConstraints.NONE,
        new Insets(22, 132, 32, 134), 44, -3));
    this.getContentPane().add(jLabel1,
                              new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0
```

```
      , GridBagConstraints.SOUTHWEST, GridBagConstraints.NONE,
      new Insets(0, 81, 0, 0), 62, 4));
    this.getContentPane().add(jTextField1,
                    new GridBagConstraints(1, 0, 1, 1, 1.0, 0.0
      , GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL,
      new Insets(50, 0, 0, 87), 0, 0));
    this.getContentPane().add(jLabel2,
                    new GridBagConstraints(0, 1, 1, 1, 0.0, 0.0
      , GridBagConstraints.WEST, GridBagConstraints.NONE,
      new Insets(22, 81, 0, 0), 0, 0));
    this.getContentPane().add(jTextField2,
                    new GridBagConstraints(1, 1, 1, 1, 1.0, 0.0
      , GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL,
      new Insets(22, 0, 0, 87), 0, 0));
    this.getContentPane().add(jLabel3,
                    new GridBagConstraints(0, 2, 1, 1, 0.0, 0.0
      , GridBagConstraints.NORTHWEST, GridBagConstraints.NONE,
      new Insets(21, 81, 1, 0), 0, 0));
    this.getContentPane().add(jTextField3,
                    new GridBagConstraints(1, 2, 1, 1, 1.0, 0.0
      , GridBagConstraints.NORTHWEST, GridBagConstraints.HORIZONTAL,
      new Insets(17, 0, 12, 87), 0, 0));

  }

  public static void main(String[] args) {
    SettingFrame s = new SettingFrame();
    s.setBounds(250,100,400,400);
    s.setVisible(true);
  }

  public void jButton1_actionPerformed(ActionEvent e) {
    this.setVisible(false);
    int aday=Integer.parseInt(jTextField1.getText());
    int kriter=Integer.parseInt(jTextField2.getText());
    int karar=Integer.parseInt(jTextField3.getText());
    Frame1 frame1 = new Frame1(aday,kriter,karar);
    frame1.setTitle("FUZZY");

    frame1.setBounds(200, 200, 200*aday, 200*aday+50);

    frame1.setVisible(true);


  }
}

class SettingFrame_jButton1_actionAdapter
    implements ActionListener {
  private SettingFrame adaptee;
  SettingFrame_jButton1_actionAdapter(SettingFrame adaptee) {
```

```
    this.adaptee = adaptee;
  }

  public void actionPerformed(ActionEvent e) {
    adaptee.jButton1_actionPerformed(e);
  }
}
```

6. SONUÇLAR

Bulanık Analitik Hiyeraşi Prosesi başlıklı bu çalışmada altı farklı yöntem ayrıntılı bir şekilde ele alınmıştır. İncelenen bulanık analitik hiyerarşi metodlarının gerek işlemsel olarak gerek önemle üzerinde durdukları çözüm şekillerine göre bazı avantaj ve dezavantajları vardır. İncelenen yaklaşımların karakteristik özelliklerine göre bu avantaj ve dezavantajlarını kısaca özetlemek gerekirse:

Laarhoven ve Pedrycz Yaklaşımı bulanık ağırlıkları ve bulanık performans ağırlıklarını elde etmek için Lootsma'nın logaritmik en küçük kareler metodunu, bulanık faydaların hesaplanmasında bulanık üçgensel sayılar için aritmetik işlemler kullanılmıştır ve çoklu karar vericilerin görüşleri aynı zamanda karşılaştırma matrislerinde modellebilir. Saaty'nin AHP metodunun üçgen bulanık sayılar kullanılarak uygulanmasıdır. Avantajı Birden fazla karar vericinin düşünceleri karşılıklı (reciprocal) matrislerde modellenebilir olmasıdır. Dezavantajları: Küçük bir problem için bile çok fazla matematiksel işlem gerektirmesi, metotta yer alan lineer denklemlerin her zaman tek çözümünün olmaması ve sadece üçgen bulanık sayıların kullanılmasına izin vermesidir.

Buckley Yaklaşımı bulanık ağırlıkları ve performans skorlarını elde edebilmek için geometrik ortalama metodunu kullanmıştır. Yöntem bulanık durumlara kolayca genelleştirilebilir ve karşılaştırma matrislerinden tek çözüm elde edilmesini garantiler. Saaty'nin AHP metodunun yamuksal bulanık sayılar ile uygulanmış halidir ve yöntem hem tek karar verici için hem de birden çok karar verici için kullanılabilir. Avantajları: Bulanık duruma genişletmek kolaydır, tek bir sonucu garanti eder. Dezavantajı: hesap gereksinimi çok fazladır.

Chang'ın Bulanık AHP Yönteminde her bir nesne alınır ve her bir amaç için derece analizi sırasıyla uygulanır. Sırasıyla sentetik derece değerlerini, olabilirlilik derecelerini ve normalize edilmiş ağırlık vektörünü bulur. Adayların kriterler altındaki ağırlıkları ile kriterlerin ağırlıkları çarpılarak toplanmasıyla kesin sonuç skorları elde edilir. Avantajları: hesap gereksinimi daha azdır, klasik AHP'nin adımlarını izler ve ilave işlem gerektirmez. Dezavantajı: sadece üçgensel bulanık sayılar kullanılabilir.

Enea ve Piazza Yaklaşımı mümkün olan tüm bilgiyi dikkate alarak kısıtlar üzerinde yoğunlaşmaktadır. Kesinlik ve güvenilirlik açısından bu yaklaşım kısıtlardan yola çıkılarak daha iyi sonuçlar almayı hedeflemektedir. Avantajları: kısıtlar üzerinde yoğunlaşır, geometrik ortalama kullanarak daha hassas sonuçlar elde eder. Dezavantajı: klasik AHP'nin adımlarını izler fakat ilave işlemler de gerektirir, sadece üçgen bulanık sayılar kullanılabilir.

Mikhailov Yaklaşımının genel karakteristikleri: öncelikleri bulanık ikili karşılaştırma kararlarından elde eder ve karşıt elemanların bulunmasına ihtiyaç yoktur. Dolayısı ile karşıtlar bulunurken oluşan kaymalarla uğraşmaz. Tamamlanmamış kararlardan önceliklendirmeleri bulmaya izin verir. Max-min optimizasyon problemini kullanır. Kesin öncelikler türetir ve başka sıralama prosedürüne ihityaç duymaz. Grup kararlarına uygulanabilir. Avantajları: Lineer Programlama metodu için: Tüm bulanık kararları aynı şekilde değerlendirir ve özel bulanık küme şekillerine ve ya formlara göre değişmez. Karışık tipte karşılaştırma kararları ile çözüm yapabilmeyi sağlar. Kolay çözülebilirdir. Lineer olmayan Programlama için: İlave olarak birleştirme ya da sıralamaya ihtiyaç duymaz. Üçgensel bulanık kümelele ifade edilmiş önceliklendirme problemlerinde kullanımı daha uygundur. Dezavantajı: Lineer olmayan Programlama çözümü gerektirir. Diğer bulanık karar yapıları için de uyarlanabilir.

Son olarak incelenen metodlardan dördünde çözümlenen örnekler için kullanılan verilerin ortak olmasından dolayı elde edilen sonuçlar aşağıda beraberce görülmektedir.

Metod	Aday I	Aday II	Aday III
Laarhoven ve Pedrycz	(0.227,0.398, 0.705)	(0.168,0.313,0.579)	(0.188,0.289,0.504)
Chang	0.41	0.28	0.25
Enea ve Piazza (I. Metod)	(0.257,0.404,0.585)	(0.202,0.333,0.507)	(0.157,0.263,0.409)
Enea Piazza (II. Metod)	(0.267,0.387,0.542)	(0.218,0.328,0.473)	(0.189,0.284,0.413)

KAYNAKLAR

Buckley, J.J., (1985), Fuzzy Hierarchical Analysis, Fuzzy Sets and Systems, 17: 233- 247.

Büyüközkan, G., Kahraman, C. and Ruan, D., (2004), A Fuzzy Multi-Criteria Decision Approach for Software Development Strategy Selection, International Journal of General Systems, 33 (2-3), 259-280, Taylor & Francis.

Chang, D.- Y., (1996), Applications of the Extent Analysis Method on Fuzzy AHP, European Journal of Operational Research, 95: 649 -655.

Chen, S. J. and Hwang, C. L., (1992), Fuzzy Multiple Attribute Decision Making, Springer-Verlag, Germany.

Enea, M. and Piazza, T., (2004), Project Selection by Constrained Fuzzy AHP, Fuzzy Optimization and Decision Making, 3: 39-62.

Kaufmann, A. and Gupta, M. M., (1988), Fuzzy Mathematical Models in Engineering and Management Science, Elsevier Science Publishers B.V. Amsterdam, The Netherlands.

Klır, G.J.and Folger T.A., (1988), Fuzzy Sets, Uncertainty and Information, Prentice Hall, New York.

Mikhailov L., (2003), Deriving Priorities from Fuzzy Pairwise Comperison Judgements, Fuzzy Sets andSystems, 134:365-385

Saaty, T. L., (1990), The Analytic Hierarchy Process, McGraw-Hill, 2nd Edition, RWS, New York.

Van Laarhoven, P. J. M., Pedrycz, W., (1983), A Fuzzy Extension of Saaty's Priority Theory, Fuzzy Sets and Systems, 11:229 -241.

Yuan B. O., Klır J.G.,(1994) ,Fuzzy Sets and Fuzzy Logic Theory and Aplications, New York

Zimmermann, H. J., (1985), Fuzzy Set Theory and Its Applications, Kluwer, Nijhoff Publishing, Boston.

8. Kara Harp Okulu Savunma Bilimleri Enstitüsü Harekat Araştırması Anabilim Dalı Harekat Analiz Bölümü, Çok Amaçlı KararVerme, Ankara, 2003.

ÖZGEÇMİŞ

Doğum tarihi	03.07.1979	
Doğum yeri	İstanbul	
Lise	1994-1997	Vefa Anadolu Lisesi
Lisans	1997-2001	Yıldız Teknik Üniversitesi Kimya-Metalurji Fakültesi Matematik Mühendisliği Bölümü
Yüksek Lisans	2003-2005	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı

Çalıştığı kurumlar

	1999-2001	DorukNet
	2001-Devam ediyor	Bahçeşehir Üniversitesi Fen-Edebiyat Fakültesi Araştırma Görevlisi