



EVLAT EDİNDİRMEDE AİLE SEÇİM SÜRECİNİN AHP VE TOPSIS TABANLI
BİLGİ SİSTEMİ İLE DEĞERLENDİRİLMESİ

Esra AKÇİN
Yüksek Lisans Tezi
Endüstri Mühendisliği Anabilim Dalı
Haziran - 2019

EVLAT EDİNDİRMEDE AİLE SEÇİM SÜRECİNİN AHP VE TOPSIS TABANLI
BİLGİ SİSTEMİ İLE DEĞERLENDİRİLMESİ

Esra AKÇİN

Kütahya Dumlupınar Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliği Uyarınca
Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman: Doç. Dr. Şafak KIRIŞ

Haziran – 2019

KABUL VE ONAY SAYFASI

Esra AKÇİN'in Yüksek Lisans tezi olarak hazırladığı “Evlat Edindirmede Aile Seçim Sürecinin AHP ve TOPSIS Tabanlı Bilgi Sistemi ile Değerlendirilmesi” başlıklı bu çalışma, jürimizce Kütahya Dumlupınar Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

26/06/2019

Prof. Dr. Önder UYSAL
Enstitü Müdürü, Fen Bilimleri Enstitüsü

Prof. Dr. Özden ÜSTÜN
Bölüm Başkanı, Endüstri Mühendisliği Bölümü

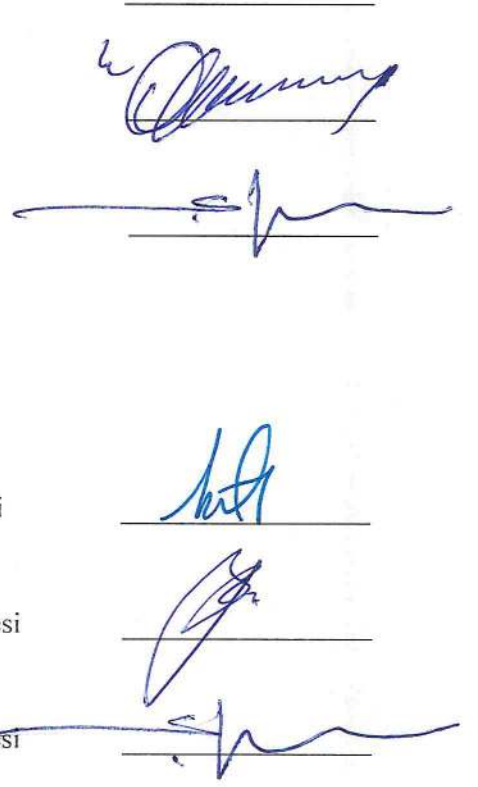
Doç. Dr. Şafak KIRIŞ
Danışman, Endüstri Mühendisliği Bölümü

Sınav Komitesi Üyeleri

Prof. Dr. Yaşar KASAP
Maden Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi

Prof. Dr. Berna HAKTANIRLAR ULUTAŞ
Endüstri Mühendisliği Bölümü, Eskişehir Osmangazi Üniversitesi

Doç. Dr. Şafak KIRIŞ
Endüstri Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi

The image shows three handwritten signatures in blue ink, each placed on a horizontal line. The first signature is at the top right, corresponding to Prof. Dr. Özden ÜSTÜN. The second signature is in the middle right, corresponding to Doç. Dr. Şafak KIRIŞ. The third signature is at the bottom right, corresponding to Doç. Dr. Şafak KIRIŞ. The signatures are stylized and cursive.

ETİK İLKE VE KURALLARA UYGUNLUK BEYANI

Bu tezin hazırlanmasında Akademik kurallara riayet ettiğimizi, özgün bir çalışma olduğunu ve yapılan tez çalışmasının bilimsel etik ilke ve kurallara uygun olduğunu, çalışma kapsamında teze ait olmayan veriler için kaynak gösterildiğini ve kaynaklar dizininde belirtildiğini, Yüksek Öğretim Kurulu tarafından kullanılmak üzere önerilen ve Dumlupınar Üniversitesi tarafından kullanılan İntihal Programı ile tarandığını ve benzerlik oranının %26 çıktığını beyan ederiz. Aykırı bir durum ortaya çıktığı takdirde tüm hukuki sonuçlara razı olduğumuzu taahhüt ederiz.



Doç. Dr. Şafak KIRIŞ



Esra AKÇİN

EVLAT EDİNDİRMEDE AİLE SEÇİM SÜRECİNİN AHP VE TOPSIS TABANLI BİLGİ SİSTEMİ İLE DEĞERLENDİRİLMESİ

Esra AKÇİN

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2019

Tez Danışmanı: Doç. Dr. Şafak KIRIŞ

ÖZET

Günümüz koşullarında yeni doğan her bebek dünyaya aynı şartlarda gelmemektedir. Ailelerinin ekonomik durumları, sağlık durumları, sosyal çevreden gelen baskılar, gayri meşru doğum gibi sebeplerden dolayı birçok küçük, devlet koruması altına verilmektedir. Kuruma verilen çocuklar devlet koruması altında; beslenme, barınma, sağlık, eğitim vb. konularda mümkün olduğunca en iyi şartlarda yetiştirilmektedirler. Fakat en sağlıklı olan küçüğün sıcak bir aile ortamında duygusal olarak da doyunluğa ulaşarak büyümesidir. Sağlıklı ilişkilerin olduğu bir ailede büyümek geleceğin temeli olan küçükler için olması gereken en doğru ortamdır. Ailelerinden ayrı büyümek zorunda kalan küçükler dışında evlat sahibi olamayan çok sayıda aile/birey bulunmaktadır. Evlat edinme işlemi aile sıcaklığı arayan küçükler ile evlat sahibi olmak isteyen ailelerin/bireylerin buluşması için önemlidir.

Ülkemizde kurum aracılığı ile evlat edinme işleminde sıraya alınma ölçütü olarak sadece evrak teslim tarihi dikkate alınmaktadır. Toplumunu oluşturan bireylerin gelişimini ve tüm hayatını etkileyen bir konuda daha fazla ölçüt göz önünde bulundurularak daha adil ve sağlıklı aile/birey sıralaması yapılmalıdır. Bu çalışmada, Bursa Aile, Çalışma ve Sosyal Hizmetler İl Müdürlüğü ile yapılan görüşmeler sonucunda evlat edinme sürecindeki etkili ölçütler belirlenerek, en uygun ailenin seçilmesi amacıyla AHP ve TOPSIS yöntemlerinin bütünleşik olarak kullanıldığı bir bilgi sistemi önerilmiştir. Bir örnek çalışma olarak önerilen bilgi sistemi ve elde edilen sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Evlat Edinme, AHP, TOPSIS, Bilgi Sistemi.

ASSESSING FAMILY SELECTION PROCESS IN ADOPTION BY AHP AND TOPSIS BASED KNOWLEDGE SYSTEM

Esra AKÇİN

Industrial Engineering, M.S. Thesis, 2019

Thesis Supervisor: Assoc. Prof. Dr. Şafak KIRIŞ

SUMMARY

Each new-born infant in present conditions wasn't born in same circumstance. Many infants are sometimes given to the under legal protection due to economical conditions of families, medical condition, pressures from the social environment and out of wedlock childbirth. The infants who are under legal protection being raised in best conditions such as nourishment, sheltering, education and health. However, the best options for infant is being raised with their family in warm and emotional environment. Besides the infants who had to grow apart from their children, there are many families or couples who doesn't have a kid. Adoption process is a milestone between infants and family/couple who has been searching for infants to adopt.

In our country, only the delivery date of document is taken into consideration as the criterion of ordering in adoption process through institution. Much more criteria should be taken into consideration in the point that affects the development of the individuals and whole life of the infant an attempt to array fairly most suitable families. In this study, the criterion of adoption process was determined after negotiations with Bursa Provincial Family Directorate, Labour and Social Services and an integrated knowledge system which includes AHP and TOPSIS has been suggested on the purpose of choosing best family. The proposed knowledge system and the results were evaluated on a case study.

Key Words: Adoption, AHP, TOPSIS, Knowledge System.

TEŞEKKÜR

İlk olarak bu çalışmanın hazırlanması aşamasında desteklerini benden esirgemeyen, çalışmanın sıkıntıya girdiği dönemlerde beni motive ederek ilerlememi sağlayan, tezimin sonuçlanmasında bana sonsuz destek veren saygıdeğer danışman hocam Doç. Dr. Şafak KIRIŞ'a, hayatım boyunca benden desteklerini esirgemeyen ve eğitim hayatıma devam etmem için beni cesaretlendiren annem Gülbeyaz AKÇİN ve babam Rüştü AKÇİN'e, çalışmamın şekillenmesi aşamasında fikirlerini esirgemeyen nişanlım Kağan KAPLAN'a ve uygulama aşamasında bilgi ve tecrübelerinden yararlandığım sevgili arkadaşım Hakan Tugay KIZILLIOĞLU'na sonsuz teşekkürlerimi sunuyorum.



İÇİNDEKİLER

Sayfa

ÖZET	v
SUMMARY	vi
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ	xii
KISALTMALAR DİZİNİ.....	xiii
1. GİRİŞ	1
2. EVLAT EDİNME SÜRECİ.....	3
2.1. Evlat Edinmenin Tarihsel Gelişimi.....	4
2.2. Evlat Edinmede Aracılık Faaliyetlerini Yürüten Kurum	5
2.3. Türk Hukukunda Evlat Edinme	7
2.4. Evlat Edinme Koşulları.....	7
2.5. Evlat Edinme Literatür Taraması.....	8
3. ÇOK ÖLÇÜTLÜ KARAR VERME.....	14
3.1. Analitik Hiyerarşi Süreci (AHP) Yöntemi.....	15
3.1.1. AHP uygulama süreci.....	16
3.1.2. AHP literatür taraması.....	21
3.2. TOPSIS Yöntemi	27
3.2.1. TOPSIS uygulama süreci	27
3.2.2. TOPSIS literatür taraması	31
3.3. AHP ve TOPSIS Yöntemi.....	34
3.3.1. AHP ve TOPSIS uygulama süreci.....	35
3.3.2. AHP ve TOPSIS literatür taraması.....	35
3.4. Personel Seçimi Literatür Taraması.....	38

İÇİNDEKİLER(devam)

	<u>Sayfa</u>
4. EVLAT EDİNME SÜRECİNDE BİR UYGULAMA.....	40
4.1. Uygulama Alanı	40
4.2. Önerilen Yöntem.....	41
4.3. Önerilen Bilgi Sistemi.....	42
5. SONUÇ VE ÖNERİLER	62
KAYNAKLAR DİZİNİ.....	64
EKLER	
1. C# Program Kodları	
ÖZGEÇMİŞ	

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1. Science Direct’te evlat edinme konu başlığı ile ilgili yayınlamış makale sayılarının yıllara göre dağılımı	8
2.2. Science Direct’te evlat edinme konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzdeleri	9
2.3. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin bilim alanlarına göre yüzde dağılımı.....	10
3.1. ÇÖKV Yöntemlerinin Sınıflandırılması (Yılmaz, 2008).....	14
3.2. Üç seviyeli bir AHP modeli	16
3.3. Science Direct’te AHP konu başlığı ile ilgili yayınlamış makale sayılarının yıllara göre dağılımı	22
3.4. Science Direct’te AHP konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzde dağılımı.....	23
3.5. YÖK Ulusal Tez Merkezi veri tabanında AHP konulu tezlerin bilim alanlarına göre yüzde dağılımı	24
3.6. Science Direct’te TOPSIS konu başlığı ile ilgili yayınlanmış makale sayılarının yıllara göre dağılımı	31
3.7. Science Direct’te TOPSIS konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzde dağılımı.....	31
3.8. YÖK Ulusal Tez Merkezi veri tabanında TOPSIS konulu tezlerin bilim alanlarına göre yüzde dağılımı.....	32
3.9. AHP temelli TOPSIS yöntemi işlem akış şeması	34
3.10. Science Direct’te AHP-TOPSIS konu başlığı ile ilgili yayınlamış makale sayılarının yıllara göre dağılımı	35
3.11. Science Direct’te AHP-TOPSIS konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzde dağılımı	36
4.1. AHP-TOPSIS temelli bilgi sistemi giriş ekranı.....	44
4.2. Seçim ekranı.....	45
4.3. Ebeveyn adaylarının bilgilerini ekleme ekranı.....	46
4.4. Yeni çocuk kayıt ekranı	47
4.5. Ölçütler ekranı.....	47

ŞEKİLLER DİZİNİ(devam)

<u>Şekil</u>	<u>Sayfa</u>
4.6. AHP öncelik matrisinin boş hali	48
4.7. AHP öncelik matrisinin doldurulmuş hali.....	48
4.8. Ölçüt ağırlıkları ekranı	50
4.9. AHP hesaplama detayları ekranı	51
4.10. Ebeveyn adayları seçim ekranı.....	52
4.11. Ebeveyn adayları seçildikten sonraki ara ekran	52
4.12. Ebeveyn adaylarının ölçüt puan ekranı	53
4.13. TOPSIS hesaplama sonuç ekranı	54
4.14. TOPSIS hesaplama sonucundaki en uygun çözümü gösteren ekran.....	55
4.15. TOPSIS hesaplaması detaylı çözüm ekranı	55
4.16. TOPSIS sonucu alternatif sıralama ekranı-1	60
4.17. TOPSIS sonucu alternatif sıralama ekranı-2	60

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
2.1. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin yıllara göre sayıları	9
2.2. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin bilim alanlarına göre sayıları.....	10
3.1. ÇÖKV yöntemlerinin kıyaslanması (Özbek, 2017).....	15
3.2. Önem ölççeği	17
3.3. RI değerleri	20
3.4. YÖK Ulusal Tez Merkezi veri tabanındaki AHP konulu tezlerin yıllara göre sayıları.....	24
3.5. YÖK Ulusal Tez Merkezi veri tabanındaki TOPSIS konulu tezlerin yıllara göre sayıları ..	32
3.6. YÖK Ulusal Tez Merkezi veri tabanındaki AHP-TOPSIS konulu tezlerin yıllara ve bilim alanlarına göre sayıları	36
4.1. 5'li likert ölççeği.....	46
4.2. AHP önem matrisi.....	49
4.3. Normalize edilmiş matris	50
4.4. Ebeveyn adayları ölçüt puanları.....	53
4.5. Başlangıç matrisi.....	56
4.6. Normalize edilmiş matris	57
4.7. Ağırlıklı standart karar matrisi	58
4.8. Pozitif ve negatif ideal çözüm değerleri.....	59
4.9. TOPSIS hesaplaması sonucu elde edilen ebeveyn aday sıralaması	59

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltma</u>	<u>Açıklama</u>
AHP	Analytic Hierarchy Process
CR	Consistency Ratio
ÇEK	Çocuk Esirgeme Kurumu
ÇÖKV	Çok Ölçütlü Karar Verme
ERP	Enterprise Resource Planning
GSM	Global System for Mobile Communications
HEC	Himaye-i Etfal Cemiyeti
HSP	Henach-Shoenlein Purpulası
İMKB	İstanbul Menkul Kıymetler Borsası
MCDM	Multi-Criteria Decision Making
NASA	National Aeronautics and Space Administration
NIS	Negative Ideal Solution
PIS	Positive Ideal Solution
RI	Random Consistency Index
SPSS	Statistical Package For The Social Sciences
T.C.	Türkiye Cumhuriyeti
TBMM	Türkiye Büyük Millet Meclisi
TKİ	Türkiye Kömür İşletmeleri
TMK	Türk Medeni Kanunu
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TSK	Türk Silahlı Kuvvetleri
UAE	United Arab Emirates
VIKOR	Vise Kriterijumska Optimizacija I Kompromisno Resenje
YÖK	Yükseköğretim Kurulu

1. GİRİŞ

Karar verme, insanın varoluşu ile birlikte ortaya çıkmış, insan hayatının gelişimi sürecinde aynı şekilde gelişime uğramış bir kavramdır. Yüzyıllar önce vahşi hayattan korunmak için mağarada yaşama kararı alınmış, ateşin keşfinden sonra yiyecekleri pişmiş olarak tüketme kararı verilmiştir. Avlanmak için taş yontma, maden işleme, hayvanları evcilleştirme, tarımsal üretim yapma, göçebe veya yerleşik yaşama gibi kararlar alınmıştır. Bu kararlar alınırken bir veya birden fazla karar ölçütü göz önünde bulundurulmuştur. İlerleyen yıllarda yaşam koşullarının değişmesi ve insanlık tarihinde yaşanan gelişmelerle birlikte karar verilecek konulara ait ölçüt ve alternatif sayıları da artmıştır. Pozitif bilimlerin ve beraberinde teknolojinin de gelişmesi ile karar verme sürecinde karar vericiye yardımcı olacak çeşitli yöntemler geliştirilmiştir.

Karar verme, mevcut tüm alternatifler arasından amaç veya amaçlara en uygun ve mümkün olan bir veya bir kaçını seçme sürecidir. Bu süreç karar vericinin seçenekler arasından bir seçim, sıralama ya da sınıflandırma yapması şeklinde sonlanabilir. Bu aşamada en doğru kararı vermek için çok ölçütlü karar verme (ÇÖKV) yöntemleri karşımıza çıkmaktadır. AHP (Analytic Hierarchy Process) ve TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) yöntemleri literatürde sıklıkla kullanılan ÇÖKV yöntemlerindendir.

AHP, 1970’li yıllarda Thomas L. Saaty tarafından geliştirilen ÇÖKV yöntemlerinden biridir. Ortaya konulan bir amaç doğrultusunda, belirlenen ölçütler ve alt ölçütler ile alternatiflerin değerlendirilmesine olanak sağlayan ve karar vericiye en uygun alternatifin belirlenmesi konusunda destek sağlayan bir yöntemdir.

TOPSIS, Yoon ve Hwang tarafından 1980’li yıllarda geliştirilmiştir. Belirlenen ölçütlere göre alternatiflerin pozitif ideal çözüm noktasına yakınlığını ve negatif ideal çözüm noktasına uzaklığını hesaplayarak; pozitif ideal çözüme en yakın, negatif ideal çözüme en uzak alternatifi belirleyen ve nicel verileri sıraya koyma konusunda en etkin aynı zamanda en kolay kullanılabilen ÇÖKV yöntemidir.

Bu çalışmada, evlat edinmek amacıyla Aile ve Sosyal Politikalar Bakanlığı Çocuk Hizmetleri Genel Müdürlüğüne başvuruda bulunan ailelerden/bireylerden evlat edinmeye uygun bulunanlar arasından AHP ve TOPSIS yöntemleri kullanılarak en uygun aile/birey seçimi yapılması amaçlanmıştır. Uygulama için Bursa Aile, Çalışma ve Sosyal Hizmetler İl Müdürlüğü seçilmiştir. Bursa Aile, Çalışma ve Sosyal Hizmetler İl Müdürlüğünde görev yapan, evlat edinme sürecinde yetkili bir sosyal çalışmacı ile yapılan görüşmelerde ailelerin/bireylerin

başvurularının sıralanması için dikkat edilecek ölçütler belirlenmiştir. Ölçütlerin birbirlerine göre önem dereceleri belirlenerek oluşturulan bilgi sisteminin AHP modülü kullanılarak ölçütlerin ağırlıkları hesaplanmıştır. Hesaplanan ağırlıklar kullanılarak bilgi sisteminin TOPSIS modülü ile başvuruda bulunan aileler/bireyler sıralanmıştır. Bilgi sisteminden elde edilen çıktılarına göre “Evlat Edinme Ebeveyn Aday Sıralama Listesi” oluşturulmuştur.

Çalışmanın ikinci bölümünde evlat edinme kavramı, tarihçesi, evlat edinmeye aracılık eden kurum ve yıllar içindeki gelişim evreleri ile ilgili bilgi verilmiştir. Türk Hukukundaki evlat edinme kavramı incelenmiş ve evlat edinebilmenin koşulları anlatılmıştır. Evlat edinme konusunda daha önce yapılan çalışmaların derlendiği bir literatür araştırması verilmiştir. Üçüncü bölümde, çalışmada kullanılan AHP ve TOPSIS yöntemlerine ait genel bilgiler ve bu yöntemler için yapılan literatür araştırmaları verilmiştir. Ele alınan problemin “Personel Seçimi Problemi” ne benzemesi sebebiyle personel seçimi problemlerinde AHP ve TOPSIS yöntemlerinin kullanıldığı çalışmalara ait bir literatür araştırması sunulmuştur. Dördüncü bölümde, problemin ele alınma sebebi ve problem çözümünde kullanılan yöntemlerin tercih edilme nedenleri anlatılmış, uygulama çalışması verilmiştir. Beşinci bölümde ise sonuç ve öneriler sunulmuştur.

2. EVLAT EDİNME SÜRECİ

İçinde bulunduğumuz 21. Yüzyılın en büyük sorunlarından biri olan nüfus artışı, birçok problemi beraberinde getirmektedir. Nüfus artışının sebep olduğu problemlerden birkaç tanesi şu şekilde sıralanabilir: Doğal kaynakların azalması ve yetersiz hale gelmesi, işsizlik oranının artması, kişi başına düşen yurt için gayri safi milli hasılanın düşmesi, refah seviyesinin azalması vb. bahsi geçen sebepler ekonomiyi direk olarak etkileyen sebeplerdir. Nüfus artışı ile birlikte yaşanan bu sorunlar kimi aileler için içinden çıkılması güç bir durum haline gelmektedir. Dünyaya getirdikleri çocuklarına hayatlarını devam ettirebilmeleri için yeterli imkanları sunamamakta ve sonuç olarak evlatlarını çeşitli şekillerde terk etmektedirler. Ekonomik olumsuz koşulların dışında; sağlık sorunları, psikolojik sorunlar, gayrimeşru doğum vb. sebeplerden dolayı çocuklar evlerinden, ailelerinden uzakta büyümek zorunda kalmaktadır. Bahsi geçen terk edilmiş veya Çocuk Esirgeme Kurumu'na bırakılmış çocuklar, korunmaya muhtaç çocuklar olarak nitelendirilmektedirler.

Aile, Çalışma ve Sosyal Hizmetler Bakanlığı korunmaya muhtaç çocuk tanımını şu şekilde yapmıştır: "Beden, ruh ve ahlak gelişimleri veya şahsi güvenlikleri tehlikede olan; ana veya babasız, ana veya babası veya her ikisi de belli olmayan, ana veya babası veya her ikisi tarafından terk edilen, ana veya babası tarafından ihmal edilip, fuhuş, dilencilik, alkollü içkiler veya uyuşturucu maddeler kullanma gibi her türlü sosyal tehlikelere ve kötü alışkanlıklara karşı savunmasız bırakılan ve başı boşluğa sürüklenen çocuğu ifade etmektedir".

Korunmaya muhtaç çocukların; doğru koşullardaki bir aile yanında büyüyerek, sağlıklı ve eksikliği olmayan bir birey olarak yetişmeleri amacıyla ülke içinde evlat edinmelerde aracılık faaliyetlerini Aile ve Sosyal Politikalar Bakanlığı Çocuk Hizmetleri Genel Müdürlüğü yürütmektedir.

Evlat edinme süreci Aile, Çalışma ve Sosyal Hizmetler Bakanlığı tarafından şu şekilde tanımlanmaktadır: "Evlat edinme; durumu evlat edindirilmeye uygun bir çocukla, durumu evlat edinmeye uygun kişi/eşler arasında hukuki bağlar sağlanarak çocuk ebeveyn ilişkisinin kurulmasıdır".

Hukuki tanımına bakıldığında ise evlat edinme, evlât edinen kişi ile evlatlık arasında sun'î nesep ilişkisi kurulması sonucunu doğuran, sadece mahkeme kararıyla kurulan ve dava yolu ile ortadan kaldırılabilen bir medeni hukuk işlemidir (Aydoğdu, 2006).

Evlat edinme, farklı ihtiyaçlara cevap verebilen bir kurumdur. Toplumun en temel yapımı birimi olan aile kavramına, evlat edinilen küçük, evlat edinildikten sonra sahip olmaktadır. Aile, bireyin dünyaya geldiği andan itibaren içinde yer aldığı, ona yaşamını devam ettirebilmesi için gerekli bakım ve desteğin sunulduğu sosyal bir ortamdır (Bayer, 2013). Evlat edinilen küçük açısından bakıldığında, dünyaya geldikten sonra biyolojik ailesi tarafından sunulmamış/sunulamamış bu sosyal ortam, evlat edinildikten sonra yeni ailesi tarafından sunulabilmektedir. Bu sayede hem evlat sahibi olmak isteyen aileler evlat sahibi olabilmekte, hem de bir aile ortamında yetişmesi gereken küçük aile yanında büyüyeabilmektedir.

2.1. Evlat Edinmenin Tarihsel Gelişimi

Günümüz Evlat edinme kurumu kesin olarak tarihi bilinmemekle beraber yüzyıllardır var olan bir kurumdur. Ailelerini çeşitli şekillerde kaybeden veya aileleri tarafından bakılamayan/bakılmayan küçükler, başka ailelerce evlat edinilmiştir. Evlat edinen aileler bu eylemi, dini, ekonomik, sosyal, siyasi gibi birçok farklı amaç için yapmıştır.

Evlat edinme kurumu, eski Roma, Yunan, Sümer, İran, Mısır, Asur, Çin, Hint, Babil hukuklarında değişik şekillerde kabul edilmekteydi (Özmen, 1999). Örneğin, M.Ö. 2000 senesinde Hammurabi kanunlarında evlat edinme kurumunun kabul edildiği ve bu konuda düzenlemelerin bulunduğu görülmektedir. Hammurabi Mecellesi dışında, Gartin ve Salık Kanunlarında, Justiniaus Müdevvatında da evlat edinme konusuna yer verilmiştir. Evlat edinme kurumunun oluşmasındaki en etkili sebep dindir. Bazı kavimlerde dinin sonsuza kadar ettirilmesinin mecburi olduğu ve öldükten sonra da ruhların yaşadığına, ahiretin var olduğuna, kendileri öldükten sonra onlar için yapılması gereken bazı ritüeller olduğuna dair inançların var olması, evlat edinme kurumunun oluşmasında etkili olmuştur (Özmen, 2004). Bu inançlara sahip kavimlere göre, çocuk sahibi olamayan ailelerin, ölümlerinden sonra atalarına ibadet edecek, mezarlarına eşyalar bırakacak evlatlara olan ihtiyaçları evlat edinme ile giderilmeye çalışılmıştır (Ataay,1957).

Türk tarihi incelendiğinde evlat edinme kurumuna sık sık rastlanmaktadır. Ölümünden sonra hayatın var olduğuna inanan Eski Türkler, öldükten sonra arkalarından dini geleneklerin yerine getirilmesi için evlat edinmiş, başka toplumlardaki gibi erkek çocuğun evlat edinilmesi esası temelinde evlat edinilen erkek çocuğa, "tutuncu" adını vermişlerdir.İslamiyet öncesi Türk Hukuku daha çok sözlü hukuka dayandığı için sağlıklı ve yeterli bilgi edinmek pek mümkün olmamaktadır. Ancak Eski Türkler toplumunda evlat edinme kavramının varlığını kanıtlayan bir çok kaynak mevcuttur (Arsal, 1947).

İslam hukukunda evlat edinme kurumu yoktur. Evlat edinme kurumunun yerine getirmesi gereken amaçlar başka kurumlar tarafından yerine getirilmiştir. İslam Hukukunda anne babası belli olmayan kimsesiz çocukların korunması, onlara bir aile yuvası temini "iltikat" kavramı ile olmaktaydı. Yardım amacıyla küçüğü almış olan kimse, "Bu çocuk benim evladımdır." diyerek çocuğun nesebini kendine bağlıyordu. Bu kuruma "tebenni" deniyordu. İslam Hukukuna göre tebenni hiç bir hukuksal sonuç doğurmuyordu (Nasuhi, 1953).

İlerleyen tarihlere bakıldığında, Osmanlı Devleti İslami Hukuk kurallarına göre yönetildiği için evlat edinme kurumunun Osmanlı Devleti'nde de olmadığı görülmektedir. Osmanlı Devletinde tebenni kurumu mevcuttu ve bu kurum ile küçük ailesinin nesebinden ayrılmıyor, küçüğe aile ortamı sağlanıyor, onun bakımı, korunması, çeşitli alanlarla yeterlilik sağlaması, ilerleyen zamanlarda mesleki anlamda geliştirilmesi gibi güzel şeyler amaçlanıyordu (Özmen, 2004). Osmanlı Devletinde 1860 yılında yetim çocuklar için "İslahiye" kurulmuş, devam eden yıllarda Darüş-Şafaka, Darül-Acaze ve Kızılay bu çocuklar için faaliyet göstermiştir (Gökçe, 2017).

Türkiye Cumhuriyeti Devleti'nin kurulmaya çalışıldığı zamanlarda, yıllar süren savaşlar sonucunda yetim kalmış, korunmaya muhtaç çok sayıda çocuk olduğu tespit edilmiştir. Bu çocukların bakım ve korunması amacıyla, bugünkü adı "Sosyal Hizmetler ve Çocuk Esirgeme Kurumu" olarak bilinen "Himaye-i Eftal Cemiyeti" 30 Haziran 1921 yılında kurulmuştur (Sarıkaya, 2005). Bu kurum yıllarca, kendini yenileyerek ve ismi değişmek suretiyle faaliyet göstermiş, pek çok çocuğa yuva olmuş ve birçok çocuğu da özlem duyduğu aile ortamına kavuşturmuştur.

Hukuki olarak bakıldığında, İsviçre Medeni Kanunu etrafında şekillenen Türk Medeni Kanunu kabul edildikten sonra, 4 Ekim 1926 yılında 743 Sayılı TMK'nin yürürlüğe girmesiyle evlat edinme konusundaki hükümler ilk defa Türk Hukukuna girmiştir.

2.2. Evlat Edinmede Aracılık Faaliyetlerini Yürüten Kurum

Ülke içi evlat edinmelerde aracılık faaliyetleri "Aracı Kurum" yetkisi ile Aile, Çalışma ve Sosyal Politikalar Bakanlığı Çocuk Hizmetleri Genel Müdürlüğü tarafından yürütülmektedir.

Ülkelerarası evlat edinme işlemleri "Merkezi Makam" yetkisi ile Aile, Çalışma ve Sosyal Politikalar Bakanlığı Çocuk Hizmetleri Genel Müdürlüğü Aile Yanında Destek Hizmetleri Dairesi Başkanlığı tarafından yürütülmektedir. Bu kapsamda, Dış İşleri Bakanlığı tarafından, Lahey Uluslararası Özel Hukuk Daimi Bürosu'na yapılan bildirim ile Genel Aile ve

Sosyal Politikalar Bakanlığı Çocuk Hizmetleri Genel Müdürlüğü, ülkelerarası evlat edinme işlemlerinde ülkemizin merkezi yetkili makamı statüsünü kazanmıştır.

Korunmaya muhtaç çocukların korunması, yetiştirilmesi, gelen talepler doğrultusunda da küçüğün yararını gözeterek evlat edindirilmesi faaliyetlerini yürüten kurumun tarihçesi genel hatları ile şu şekildedir (T.C. Aile, Çalışma ve Sosyal Hizmetler Bakanlığı Resmi İnternet Sitesi);

- 6 Mart 1917 tarihinde savaşlar sonrası yetim kalan çocukların bakımını sağlamak amacıyla İstanbul Himaye-i Etfal Cemiyeti (HEC) İstanbul'da kurulmuştur.
- 1 Kasım 1917 tarihinde kimsesiz çocukların sağlık sorunları ile ilgilenmek amacıyla Şehzadebaşı Muayenehanesinin (Dispanser) açılışı gerçekleştirilmiştir.
- 28 Kasım 1917 tarihinde Firuz Ağa'da ilk çocuk misafirhanesi açılmıştır.
- Faaliyetlerine devam eden HEC 17 Ocak 1921'de padişah iradesi ile kamu yararına çalışan cemiyet olarak kabul edilmiştir.
- 30 Haziran 1921 tarihinde Ankara'da Hakimiyet-i Milli Matbaasında küçük bir odada HEC kuruluşu gerçekleşmiştir.
- 1 Ağustos 1921 tarihinde TBMM Reisi Mustafa Kemal Paşa, HEC'in koruyuculuğunu kabul etmiştir.
- Devam eden yıllarda HEC'in güçlendirilmesi için çeşitli adımlar atılmış, yardımlar yapılmış, 1925 yılında bu yardımlardan 30757 çocuk faydalanmıştır.
- Yurt dışında yapılan uluslararası kongrelere katılmış, yurt dışında yaşayan Türk vatandaşlardan yardımlar toplanmış, ülke içinde çocukların ihtiyaçlarını karşılamak amacıyla gelir elde etmek için çeşitli balolar ve müsamereler düzenlenmiştir.
- 1935 yılı itibari ile HEC "Türkiye Çocuk Esirgeme Kurumu" ismini almıştır.
- 26 Kasım 1937 yılında ÇEK Bakanlar Kurulu'nun 1223 sayılı kararı ile kamu yararına çalışan dernek olarak kabul edilmiştir.
- İlerleyen yıllarda birçok ilde yeni şubeler açılmış, çocukların sosyal ve kültürel olarak gelişimlerini sağlamak amacıyla yatırımlar yapılmış, beslenme ve sağlık konularında da çeşitli çalışmalar yapılarak her anlamda sağlıklı bireyler yetiştirmek amaçlanmıştır.
- 24 Mayıs 1983 tarihinde 2828 sayılı yasa ile Sosyal Hizmetler ve Çocuk Esirgeme Kurumu Kanunu kabul edilmiş, 27 Mayıs 1983 tarihinde de 18059 sayılı Resmi Gazete'de 2828 sayılı Sosyal Hizmetler ve Çocuk Esirgeme Kurumu Kanunu yayınlanmıştır.

2.3. Türk Hukukunda Evlat Edinme

4 Ekim 1926 yılında ilk defa Türk Hukukuna giren 743 Sayılı Türk Medeni Kanunu, evlat edinmeyi Aile Hukukunun düzenlendiği “Hısımlık” başlığını taşıyan ikinci kısmın evlat edinme başlığını taşıyan üçüncü faslında düzenlemiştir. Evlat edinme ile ilgili düzenlemeler kanunun 253 ile 258’inci maddelerinde bulunmaktadır.

Türk Medeni Kanununun evlat edinme ile ilgili hükümleri, 16.06.1983 tarihli, 2846 sayılı ve 23.11.1990 tarihli, 3678 sayılı kanunlarla iki defa değişikliğe uğramıştır. Daha sonra 01.01.2002 tarihinde yürürlüğe giren 4721 sayılı yeni Türk Medenî Kanunu, evlat edinme kurumunu daha sistematik ve ayrıntılı olarak ele almıştır. Büyük oranda İsviçre Medenî Kanunu’na paralel düzenlemeler ihtiva eden yeni Türk Medenî Kanunu’nun evlat edinme ile ilgili hükümleri, çağdaş gelişmelere uygun olması bakımından temelde olumlu düzenlemelere sahiptir (Aydoğdu, 2006).

2.4. Evlat Edinme Koşulları

Evlat edinmek için gerekli koşullar aşağıda verildiği gibidir (T.C. Aile, Çalışma ve Sosyal Hizmetler Bakanlığı Resmi İnternet Sitesi):

- Küçüğün, evlât edinen tarafından en az bir yıl süreyle bakılmış ve eğitilmiş olması,
- Evlât edinmenin her halde küçüğün yararına bulunması ve evlât edinenin diğer çocuklarının yararlarının, hakkaniyete aykırı bir biçimde zedelenmemesi,
- Evlât edinecek kişi veya eşlerin, evlât edinilenden en az on sekiz yaş büyük olması,
- Ayırt etme gücüne sahip olan küçüğün rızasının alınması,
- 22.11.2001 tarihli ve 4721 sayılı Türk Medenî Kanununun 311. ve 312. maddelerinde yer alan hükümler hariç, 309. maddesinde belirtilen şekilde küçüğün ana ve babasının rızasının bulunması,
- Küçüğün vesayet altında olması halinde, Türk Medenî Kanununun 397. maddesinde öngörülen vesayet dairelerinin izninin alınmış olması,
- Eşlerin en az beş yıldan beri evli olmaları veya her ikisinin de otuz yaşını doldurmuş bulunmaları koşulları aranmaktadır.

Evli olmayan kişinin evlat edinmek istemesi koşulu;

Evlât edinmek isteyen ve evli olmayan kişinin otuz yaşını doldurmuş bulunması koşulu aranmaktadır.

Evli olmayanlar birlikte evlat edinmemektedir.

Eşlerden birinin diğerinin çocuğunu evlat edinme istemesinin koşulu;

En az iki yıldan beri evli olmaları veya kendisinin otuz yaşını doldurmuş bulunması koşulları aranmaktadır.

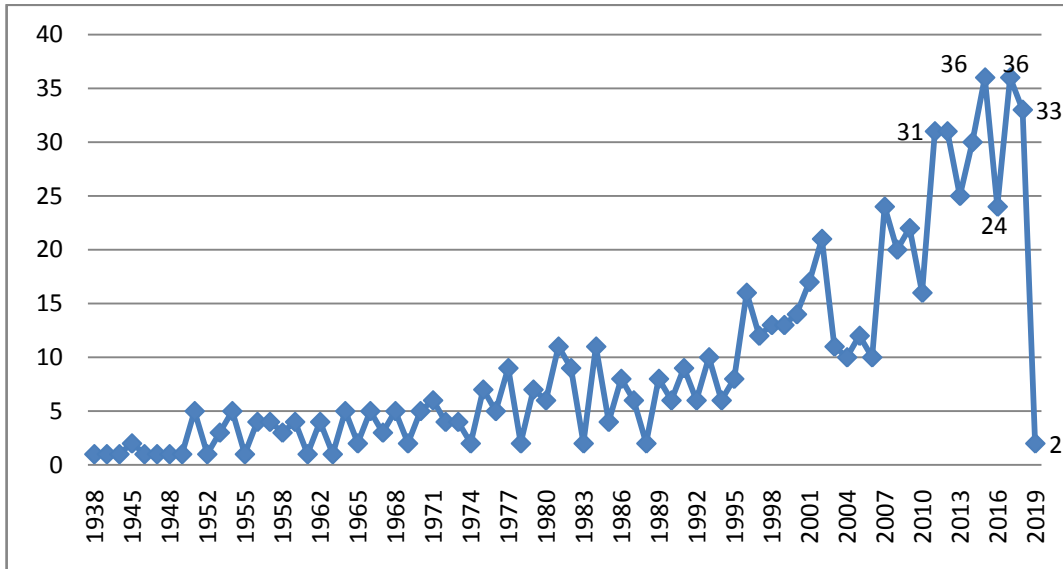
Evli olan eşin tek başına evlat edinme koşulu;

Otuz yaşını doldurmuş olan eşin, tek başına evlât edinebilmesi için;

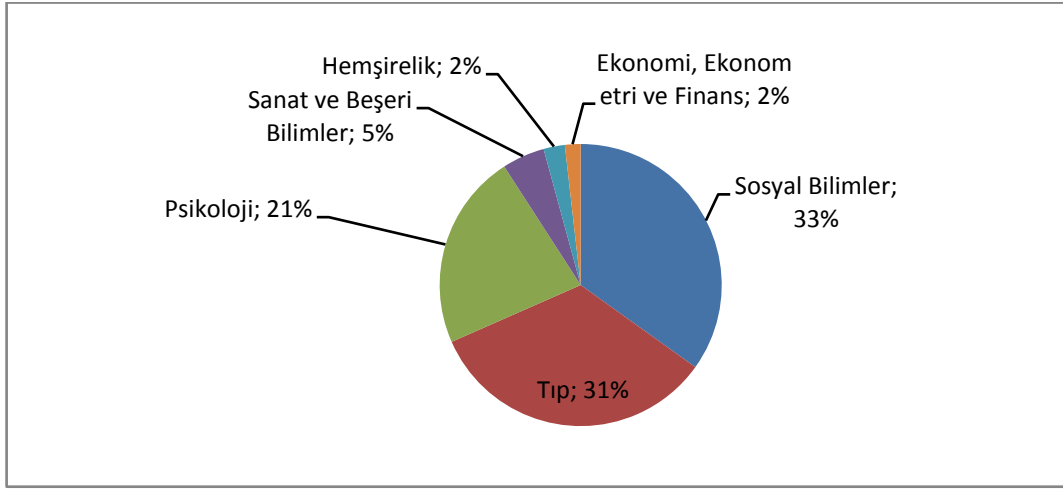
- Diğer eşin ayırt etme gücünden sürekli olarak yoksunluğu veya iki yılı aşkın süreden beri nerede olduğunun bilinmemesi ya da,
- Mahkeme kararıyla iki yılı aşkın süreden beri eşinden ayrı yaşamakta olması yüzünden, birlikte evlât edinmesinin mümkün olmadığını ispat etmesi,
- Türk Medeni Kanunun 306. maddesine göre, en az 5 yıldan beri evli olmaları veya otuz yaşını doldurmuş bulunmaları gerekmektedir.

2.5. Evlat Edinme Literatür Taraması

Science Direct veri tabanındaki başlıklarda “Evlat Edinme” konusu tarandığında ulaşılan sonuçların yıllara ve bilim alanlarına göre dağılımı aşağıdaki grafiklerde verilmiştir.



Şekil 2.1. Science Direct’te evlat edinme konu başlığı ile ilgili yayınlanmış makale sayılarının yıllara göre dağılımı.



Şekil 2.2. Science Direct’te evlat edinme konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzdeleri.

Verilere bakıldığında literatürde evlat edinme konusunun 1938 yılından itibaren az sayıda da olsa irdelendiği ancak özellikle son yıllarda bu konulardaki çalışmalarda bir artış olduğu görülmektedir. Evlat edinme konusunu inceleyen bilim alanlarına bakıldığında ilk sırada Sosyal Bilimlerin (%33), olduğu, ikinci sırada Tıp Biliminin (%31) yer aldığı ve bunları üçüncü sırada Psikoloji Biliminin (%21) izlediği görülmektedir.

YÖK Ulusal Tez Merkezi veri tabanı incelendiğinde ilk kayıtlı tezin olduğu 1986 yılından 2017 yılına kadar olan dönemde evlat edinme konulu 21 yüksek lisans tezine, 1 doktora tezine ulaşılmış ve bu tezlerin yıllara göre dağılımları Çizelge 2.1’de verilmiştir.

Çizelge 2.1. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin yıllara göre sayıları.

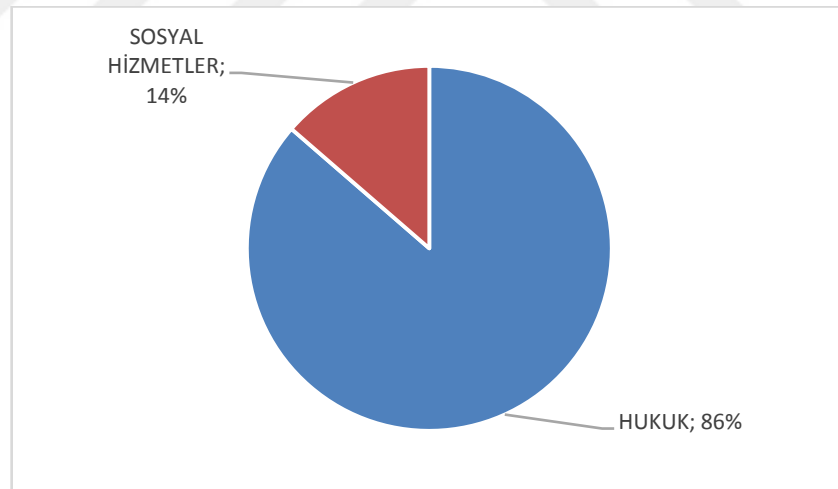
TEZ TÜRÜ	1986	1987	1988	1991	1992	1993	1995	1997	1999	2003
Yüksek Lisans	1	1	1	1	1	1	1	1	1	1
Doktora										
TOPLAM	1	1	1	1	1	1	1	1	1	1
TEZ TÜRÜ	2004	2005	2006	2007	2008	2009	2011	2013	2016	2017
Yüksek Lisans	1	1	1	1	2	1	1	1	1	1
Doktora								1		
TOPLAM	1	1	1	1	2	1	1	2	1	1

YÖK Ulusal Tez Merkezi veri tabanında evlat edinme konulu tezlerin bilim alanlarına göre sayıları Çizelge 2.2'de verilmiştir.

Çizelge 2.2. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin bilim alanlarına göre sayıları.

BİLİM ALANLARI	1986	1987	1988	1991	1992	1993	1995	1997	1999	2003
Hukuk	1	1			1	1	1	1	1	1
Sosyal Hizmetler			1	1						
TOPLAM	1	1	1	1	1	1	1	1	1	1
BİLİM ALANLARI	2004	2005	2006	2007	2008	2009	2011	2013	2016	2017
Hukuk	1	1	1	1	2	1	1	1	1	1
Sosyal Hizmetler								1		
TOPLAM	1	1	1	1	2	1	1	2	1	1

Bilim alanlarına göre makale yüzdeleri de Şekil 2.3'de verildiği gibidir.



Şekil 2.3. YÖK Ulusal Tez Merkezi veri tabanındaki evlat edinme konulu tezlerin bilim alanlarına göre yüzde dağılımı.

YÖK'te yayınlanan tezlere bakıldığında ülkemizde evlat edinme tezlerin sadece iki alanda yapıldığı görülmektedir. Bunlar %86 oranı ile Hukuk ve %14 oranı ile Sosyal Hizmetler alanıdır.

Evlat edinme konusunda yapılmış bazı çalışmalar aşağıda verilmiştir.

Olmsted (1975), evlat edinme sürecinden sonra evlat edinilen çocuğa gerçeklerin açıklanması sonucu meydana gelen, çocuk-ebeveyn çatışmasını incelemiştir. Yüzleşmeler sonucunda ortaya çıkan bu çatışmanın, evlat edinilen çocuğun biyolojik ebeveynleri hakkında bilgi edinme ihtiyacı yaşamasından ve evlat edinenlerin biyolojik ebeveynlerle rekabet etme eğiliminden kaynaklandığını ortaya koymuştur. Çalışma sonucunda, bu sorunlara yaklaşımı açısından çocuk doktorunun müdahalesine ilişkin kılavuz ilkeleri belirtmiştir.

Braff vd. (1982), 6-13 yaş arasındaki evlat edinilmiş ve evlat edinilmemiş çocuklar üzerinde bir inceleme yapmışlardır. Erken-orta çocukluk yaşlarındaki evlat edinilmeyen çocukların, evlat edinilen çocuklara göre evlat edinme görüşüne daha olumlu baktıkları sonucuna varmışlardır. Yaşları büyüdükçe, evlat edinilmeyen çocukların evlat edinme konusunda daha az olumsuz davranırken, evlat edinilen çocukların evlat edinme konusunda daha olumsuz davrandıkları sonucunu ortaya koymuşlardır.

Wender vd. (1986), intihar davranışında genetik etkenlerin rolünü belirlemek amacıyla evlat edinmenin gerçekleştiği kişi ve aileleri gözlemlemişlerdir. Yapılan çalışmada, duygu durum bozukluğu olan erişkin evlatlıkların biyolojik ve evlat edinen akrabaları karşılaştırılmış ve bu evlatlıklarla eşleştirilmiş sağlıklı evlatlıklar incelenmiştir. Çalışma, duygu durum bozukluğu olan evlatlıkların biyolojik akrabalarında sağlıklı evlatlıkların biyolojik akrabalarına göre 15 kat daha fazla intihar olduğunu göstermiştir.

Brodzinsky (1987), evlat edinilen çocukların davranışsal ve psikolojik sorunlara karşı savunmasızlığını kanıtlayan bir belge sunmuştur. Bu incelemenin ardından temel tezi, evlat edinme deneyiminin, ebeveynleri ve çocukları, Erikson'un tarif ettiği aile yaşamının daha evrensel gelişimsel görevleriyle etkileşime giren ve karmaşık hale getiren benzersiz bir psikososyal çatışmalar veya görev grubuna maruz bırakmasına dayanan psikososyal evlat edinme uyarlaması modeli tanımlamıştır.

Glidden (1991), gelişimsel engelli 87 çocuğun evlat edindirildikten sonraki sürecini incelemiş ve genel olarak ailenin yerleşime verdiği ilk tepkilerin yaklaşık 5 yıl sonra uyumlu olumlu yönde olduğunu belirtmiştir.

Avery (1998), makalesinde 1980 tarihli Evlat Edinme Yardımı ve Çocuk Esirgeme Kanunu'nun (PL 96-272) devlet tarafından uygulanmasının bir analizini sunmuştur. Çalışmasında, P.L. 96-272 Kanunu'nun, çocukların evlat edinmeye destek sağlanması için uygunluğunu, bu çocuklar için bir yaşam kalitesi sağlamadaki yeterliliğini ve bu desteğin çocukluk boyunca güvenliğini irdelemiştir.

Dumaret ve Rosset (2005), Fransa'da evlat edinme politikasını ve uygulamalarını, iki sistem altında incelemiş (tam evlat edinme ve basit evlat edinme) ve çalışmalarında koruyucu aile bakım sistemi ile sunmuşlardır.

Akpalu (2007), çalışmasında evlat edinmenin temel amaçlarını araştırmıştır. Evlat edinmenin, terk edilmiş, yetim ve ihmal edilmiş çocukları yeni ebeveynler ve / veya evlerle donatmayı içerdiğini ortaya koymuştur. Çalışma sonucunda, evlat edinmede karşılaşılan zorluklar arasında çocukların bulunmaması ve maddi sorunlar olduğunu tespit etmiştir. Evlat edinme, terk edilmiş, yetim, istismara uğramış ve ihmal edilmiş çocuklar için yeni ebeveynlerin, ailelerin ve evlerin sağlanmasına yol açtığı sonucuna varmıştır.

Kaya (2009), kitabında 4 Ekim 1926 tarihinde 743 sayılı Türk Kanun Medenisi'nin kabulüyle, ilk defa bugünkü anlamıyla Türk Hukukuna giren evlat edinmeye ilişkin hükümleri, Türk Kanunu Medenisi'nin evlat edinme ile ilgili hükümlerini, 01.01.2002 tarihinde yürürlüğe giren evlat edinme kurumunu daha sistematik ve ayrıntılı olarak ele alan 4721 sayılı yeni Türk Medeni Kanununu ayrıntılı olarak ele almıştır.

Akbulut (2011), sosyal hizmetler ve çocuk esirgeme kurumu aracılığıyla küçüklerin evlat edinilmesi ve koruyucu aile kurumu hakkında, konularla ilgili kanun, yönetmelik, tüzük, genelge ve yönergeleri inceleyerek bir yüksek lisans tezi hazırlamıştır.

Muntean ve Ungureanu (2012), Romanya'da çocukların evlat edinilmesi esnasında bağlanma teorisinin uygulanması konusunda bir çalışma yapmışlardır. Değerlendirilen 40 çocuğun yarısı güvenli bir şekilde bağlanmıştır. Çalışma sonucunda, ebeveynlerin çocuk bağlanmalarını teşvik etmedeki rollerinin farkında olduklarında, çocukların ebeveynlerine güvenli bağlanma geliştirdiklerini tespit etmişlerdir.

Muntean vd. (2012), erken yaşta evlat edinilen (0-4yaş) 40 Rumen çocuğun gelişimini gözlemlemişlerdir. Evlat edinilmiş bir çocuğun gelişiminin birçok özelliğe bağlı olduğunu ve bunlardan önemli olan birinin de okul olduğunu belirtmişlerdir. Okulun evlat edinilen çocukların esnekliğini geliştirmedeki olumlu rolünü incelemişlerdir.

Siegel (2012), evlat edinmedeki etik sorunların kapsamını incelemiş ve evlat edinmede en iyi uygulamaların ne olması gerektiği konusunda tartışmalar ortaya koymuştur.

Yılmaz (2013), hazırlamış olduğu yüksek lisans tezinde, kişilerin evlat edinme hakkındaki bilgileri ve tutumları araştırmıştır. Bu çalışma esnasında 500 kişi ile görüşülerek bir anket çalışması yapılmıştır. Bireylerin evlat edinme yönergelerinde yer alan şartlara büyük ölçüde

katıldıkları görülmüştür. Bireylerin evlat edinme hizmeti hakkında büyük oranda bilgi sahibi olmadıkları, evlat edinme hizmetine bu yüzden çekimser baktıkları ve evlat edinme hizmetine bakışta çoğunlukla ekonomik nedenlerin gerekçe gösterildiği sonucuna varılmıştır.

Dikmeer vd. (2014), 6-18 yaşları arasındaki 61 evlat edinilmiş çocuk ile biyolojik ailesiyle birlikte yaşayan 62 çocuk üzerinde bir çalışma yapmışlardır. Evlat edinilmiş çocuklar ile biyolojik aileleriyle yaşayan yaşlıları arasında herhangi bir olmadığını tespit etmişlerdir. Çalışmada için gerekli, anne babaların evlat edinmeyle ilgili görüşleri anket yoluyla sorulan sorular üzerinden değerlendirilmiştir.

Gökce (2017), devletin koruması altındaki çocukların evlat edindirilmesi için geliştirilen sosyal politikaları incelemiş ve toplumun evlat edinme konusuna olan tavrını ortaya koymuştur. Çalışma sosyoloji alanında ele alınmamış bir problemi konu edinmesi bakımından önem arz etmektedir. Bu çalışmada toplumun evlat edinme sürecine hangi sebeplerle karşı çıktığı anlatılmıştır.

Özcan (2017), Medeni Kanuna göre evlat edinmede aranan rıza koşulunu evlat edinilen küçük, evlatlık veren aile, evlat edinen aile açısından ilgili kanun maddeleri uyarınca incelemiş ve çalışmasında başlıklar altında toplamıştır.

Haliloğlu (2017), literatürdeki diğer çalışmaları da göz önünde bulundurarak evlat edinmenin İslam dini açısından mahiyetini ele almıştır. Nisa ve Ahzab surelerindeki ilgili ayetlerin ve tebennî konusuyla alakalı hadis-i şerifin hükmüne göre; babası ve soy bağı bilinen bir çocuğun başka bir kimseye nispet edilerek evlat edinilmesinin haram olduğunu belirtmiştir. Babası belli olmayan, sahipsiz bir çocuğu bulan kişi onun kendi evladı olduğunu iddia ediyorsa ve aralarındaki yaş farkı da buna engel değilse bu iddia geçerli sayılıp çocuğun bu kişinin evladı olduğunun kabul edildiğini belirtmiştir.

Aygün ve Yakut (2018), İslam Hukukunun evlat edinme konusuna bakış açısını incelemişlerdir. Bu incelemeden yola çıkarak, Osmanlı Hukukunda, Eski Medeni Kanunda ve günümüzdeki Medeni Kanunda evlat edinmenin miras hukuku açısından yarattığı sonuçları ele almışlardır.

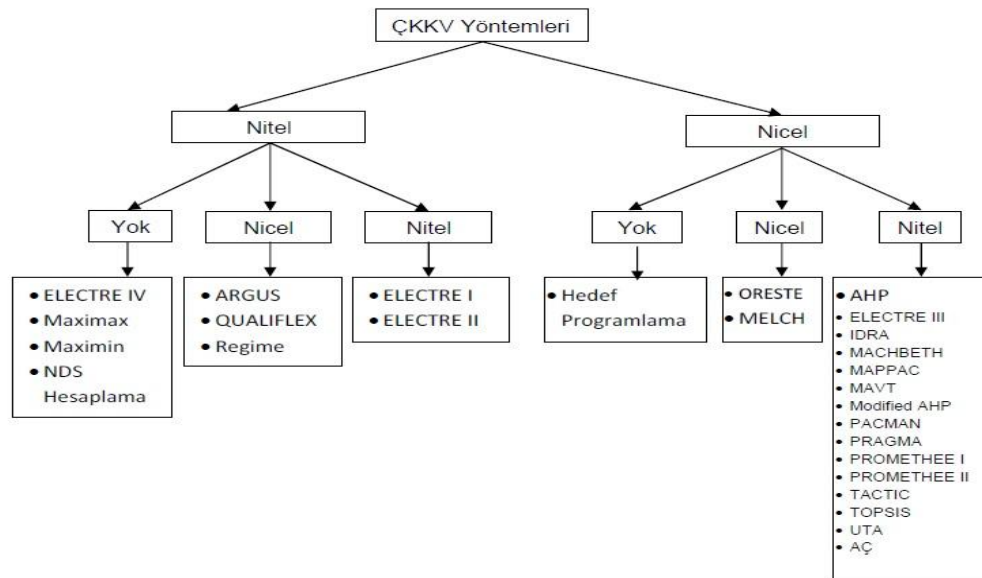
3. ÇOK ÖLÇÜTLÜ KARAR VERME

Sonlu sayıda seçeneğin seçilme, sıralanma, sınıflandırma, önceliklendirme veya elenme amacıyla genellikle ağırlıklandırılmış, birbirleri ile çelişen ve aynı ölçü birimini kullanmayan hatta bazıları nitel değerler alan çok sayıda ölçüt kullanılarak değerlendirilmesi işlemidir (Alkan 2006).

Karar verme aşamasında ÇÖKV yöntemlerinin kullanılması karar vericilere alternatiflerin değerlendirilmesinde yardımcı olmakta ve işletme kaynaklarının oldukça verimli kullanılmasına imkan sağlamaktadır (Dağdeviren vd., 2005).

Çok ölçütlü karar verme yöntemlerinin temel adımları şu başlıklar altında toplanabilir (Opricovic ve Tzeng, 2004):

1. Problemin değerlendirme ölçütlerinin belirlenmesi,
2. Alternatiflerin belirlenmesi,
3. Ölçütlere göre alternatiflerin değerlendirilmesi,
4. Çok ölçütlü karar verme yönteminin uygulanması,
5. Kullanılan yönteme göre bir alternatifin seçilmesidir.



Şekil 3.1. ÇÖKV Yöntemlerinin Sınıflandırılması (Yılmaz, 2008).

Çizelge 3.1. ÇÖKV yöntemlerinin kıyaslanması (Özbek, 2017).

ÇÖKV Yöntemi	Basitlik	Hesaplama Süresi	Güvenilirlik	Matematiksel İşlemler	Veri Türü
MOORA	Çok basit	Çok az	İyi	Asgari	Nicel
AHP	Çok kritik	Çok fazla	Zayıf	Azami	Karma
ELECTRE	Makul kritik	Fazla	Orta	Makul	Karma
PROMETHEE	Makul kritik	Fazla	Orta	Makul	Karma
TOPSIS	Makul kritik	Makul	Orta	Makul	Nicel
VIKOR	Basit	Az	Orta	Makul	Nicel

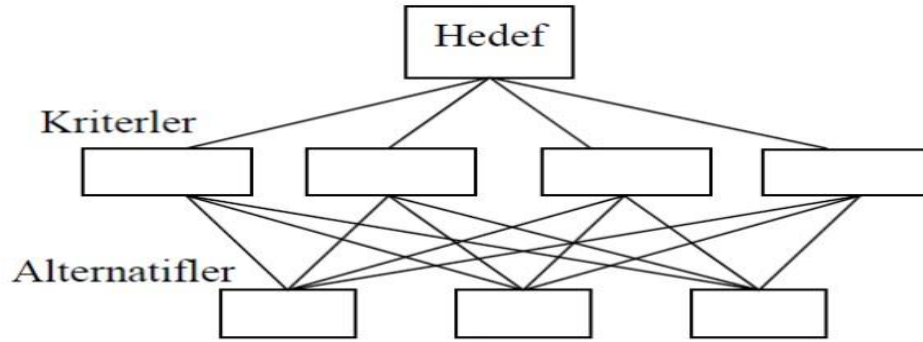
3.1. Analitik Hiyerarşi Süreci (AHP) Yöntemi

Analitik hiyerarşi süreci (AHP), Thomas L. Saaty tarafından 1977 yılında, karmaşık problemlerin çözümü için geliştirilen, birçok problemin çözümü için yaygın olarak kullanılan bir ÇÖKV yöntemidir. AHP, alternatifleri belirlenen ölçütlerin ağırlıklarını baz alarak sıralayan bir yöntemdir (Özbek, 2017).

AHP çok sayıda performans ölçütü kullanarak karar seçeneklerini puanlama ve sıralamaya tabi tutan, bu sayede karar problemini basitleştiren çok ölçütlü bir yöntemdir (Zahedi, 1986).

AHP, tüm bilgileri sentezleyerek, bir sistem alanında ve sistematik şekilde alınan karmaşık kararların daha rasyonel olmasını sağlar (Handfield vd. , 2002). Yani, matematik ve psikoloji bu yöntemin yapıtaşlarını oluşturmaktadır (Perera ve Sutrisna, 2010).

AHP Yöntemi kullanılarak çözülecek problemlerin mümkün olduğunca ayrıntılı bir şekilde tanımlanarak belirli bir öncelik hiyerarşisine göre oluşturulmuş olması gerekmektedir. AHP Yöntemi ile en az üç seviyeden oluşan bir hiyerarşik yapı kurulduğunda hiyerarşinin en üst seviyesine ana hedef ve en alt seviye de ise karar alternatifleri bulunmalıdır (Saaty, 1990).



Şekil 3.2. Üç seviyeli bir AHP modeli.

AHP yönteminin Saaty tarafından belirlenen aksiyomları şunlardır (Saaty, 1986);

- Aksiyom 1 (Karşılık olma): Eğer i . ölçütün j . ölçüte göre önem derecesi a_{ij} ise j . ölçütün i . ölçüte göre önem derecesi olan a_{ji} değeri $1/a_{ij}$ eşitliğinden oluşur.
- Aksiyom 2 (Homojenlik): İkili karşılaştırmalarda x ve z ölçütlerden biri diğerine göre sonsuz üstün kabul edilmez.
- Aksiyom 3 (Bağımsızlık): Karar seçenekleri ve ölçütler birbirlerinden tamamen bağımsızdır.
- Aksiyom 4: Karar problemi hiyerarşik yapıda tasarlanabilir.

3.1.1. AHP uygulama süreci

Adım 1: Problemin Tanımlanması ve Hiyerarşinin Oluşturulması

Karar verilmesi gereken problem tanımlanmalı ve problemin AHP ile çözülebilirliğine bakılmalıdır. Probleme ait ölçütler belirlenmelidir. Ele alınan problemin amacı belirlenerek hiyerarşinin en üst noktasına konulmalıdır. Ölçütler gerekli ise sınıflandırılmalı ve hiyerarşik yapıya ilave edilmelidir.

Adım 2: İkili Karşılaştırma Matrislerini Oluşturulması

Bu aşamada, belirlenen her ölçüt bir başka ölçütle karşılaştırılmaktadır. Ölçütlerin karşılaştırılması, birbirlerine göre sahip oldukları önem değerlerine göre birebir ve karşılıklı olarak yapılmaktadır. Ölçütlerin karşılaştırılması işleminde Çizelge 3.2'deki önem ölçeği kullanılmaktadır. Kendisi ile karşılaştırılan matrisin kesişim hücrelerindeki değerler 1 değerini almaktadır.

Karşılaştırmalar yapılırken matrisin 1 olan köşegeninin üst tarafında yer alan alanları Çizelge 3.2'deki değerlere göre doldurulmakta, köşegenin altındaki alanları doldurmak için de (3.1) eşitliği kullanılmaktadır.

$$a_{ji} = \frac{1}{a_{ij}} \quad (3.1)$$

Çizelge 3.2. Önem ölçeği.

Önem Değerleri	Değer Tanımları
1	Her iki faktörün eşit öneme sahip olması durumu
3	1. Faktörün 2. faktörden daha önemli olması durumu
5	1. Faktörün 2. faktörden çok önemli olması durumu
7	1. Faktörün 2. faktöre nazaran çok güçlü bir öneme sahip olması durumu
9	1. Faktörün 2. faktöre nazaran mutlak üstün bir öneme sahip olması durumu
2,4,6,8	Ara değerler

Adım 3: Faktörlerin Yüzde Önem Dağılımları Belirlenir

Karşılaştırma matrisinin köşegeni üzerindeki matris bileşenleri 1 değerini almaktadır. Karşılaştırma matrisi aşağıda verilmiştir.

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

Ölçütlerin bütün içerisindeki ağırlıklarını, yani yüzde önem dağılımlarını belirlemek amacıyla, karşılaştırma matrisini oluşturan sütun vektörlerinden yararlanılmakta, n adet ve n bileşenli B sütun vektörü oluşturulmaktadır. B vektörü aşağıda verildiği gibidir:

$$B_i = \begin{bmatrix} b_{11} \\ b_{21} \\ \cdot \\ \cdot \\ \cdot \\ b_{n1} \end{bmatrix}$$

B sütun vektörlerinin hesaplanması aşamasında (3.2) eşitliğinden faydalanılmaktadır.

$$b_{ij} = \frac{a_{ij}}{\sum_{i=1}^n a_{ij}} \quad (3.2)$$

Her bir ölçüt için eşitlik (3.2) hesaplandığında, ölçüt sayısı kadar B sütun vektörü bulunmaktadır. n adet B sütun vektörü, bir matris haline getirildiğinde C matrisi oluşmaktadır.

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix}$$

C matrisinden hareketle, ölçütlerin birbirlerine göre önem değerlerini gösteren yüzde önem dağılımları elde edilmektedir. Bu işlem esnasında (3.3) eşitliğinde gösterildiği gibi C matrisini oluşturan satır bileşenlerinin aritmetik ortalaması alınmakta ve öncelik vektörü olarak adlandırılan w sütun vektörü elde edilmektedir.

$$w_i = \frac{\sum_{j=1}^n c_{ij}}{n} \quad (3.3)$$

W vektörü aşağıda verildiği gibidir:

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix}$$

Adım 4: Faktör Kıyaslamalarındaki Tutarlılık Ölçülmesi

AHP hesaplamalarının ne kadar tutarlı olduğunu belirlemek için Tutarlılık Oranı (CR) hesaplanmaktadır. AHP yönteminin, CR hesaplaması, ölçüt sayısı ile "Öz Değer" adı verilen (λ) bir katsayının karşılaştırılmasına dayandırılmaktadır. λ 'nın hesaplanabilmesi için öncelikle A karşılaştırma matrisi ile w öncelik vektörünün matris çarpımından D sütun vektörü elde edilmektedir.

$$D = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix}$$

(3.4) eşitliğinde tanımlandığı gibi, bulunan D sütun vektörü ile w sütun vektörünün karşılıklı elemanlarının bölümünden her bir değerlendirme faktörüne ilişkin öz değer (E) elde edilmektedir. Elde edilen w sütunlarındaki değerlerin aritmetik ortalaması ((3.5) eşitliği) karşılaştırmaya ilişkin öz değeri (λ) vermektedir.

$$E_i = \frac{d_i}{w_i} \quad (i = 1, 2, \dots, n) \quad (3.4)$$

λ değeri hesaplandıktan sonra Tutarlılık Göstergesi (CI), (3.6) eşitliğinden yararlanılarak hesaplanmaktadır.

$$\lambda = \frac{\sum_{i=1}^n E_i}{n} \quad (3.5)$$

Son aşamada ise CI, Random Gösterge (RI) olarak adlandırılan ve Çizelge 3.3'de gösterilen standart düzeltme değerine bölünerek ((3.7) eşitliği) CR elde edilmektedir. Çizelge 3.3' den ölçüt sayısına karşılık gelen değer seçilmektedir.

$$CI = \frac{\lambda - n}{n - 1} \quad (3.6)$$

Çizelge 3.3. RI değerleri.

N	RI	N	RI
1	0	8	1,41
2	0	9	1,45
3	0,58	10	1,49
4	0,90	11	1,51
5	1,12	12	1,48
6	1,24	13	1,56

$$CR = \frac{CI}{RI} \quad (3.7)$$

Hesaplanan CR değerinin 0,10'dan küçük olmalıdır. CR değerinin 0,10'dan büyük olması halinde ölçüt değerleri gözden geçirilerek matris tekrar oluşturulmalıdır.

Adım 5: Ölçütlerin Yüzde Önem Dağılımlarının Hesaplanması

Bu adımda her ölçütün karar noktalarının yüzde önem dağılımları belirlenmektedir. Karşılaştırmalar ve matris işlemleri sayısı ölçüt sayısı kadar tekrarlanmaktadır. Her ölçüt için karar noktalarında kullanılacak G karşılaştırma matrislerinin boyutu mxm olmaktadır. Her bir karşılaştırma işleminden sonra mx1 boyutlu olan ve değerlendirilen faktörün karar noktalarına göre yüzde dağılımlarını gösteren S sütun vektörleri elde edilmektedir. Yüzde dağılım sütun vektörleri aşağıda tanımlandığı gibidir:

$$S_i = \begin{bmatrix} s_{11} \\ s_{21} \\ \cdot \\ \cdot \\ \cdot \\ s_{m1} \end{bmatrix}$$

Adım 6: Alternatiflerin Sıralanması

İlk olarak, S sütun vektöründen yararlanılarak $m \times n$ boyutlu K karar matrisi oluşturulmaktadır. Karar matrisi aşağıda verilmiştir:

$$K = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ s_{m1} & s_{m2} & \dots & s_{mn} \end{bmatrix}$$

Oluşturulan karar matrisi, öncelik vektörü ile matris çarpımı kurallarına uyularak çarpıldığında L sütun vektörü elde edilmektedir. L sütun vektörü her bir seçeneğe ait yüzde oranı vermektedir. Elde edilen l_{ij} değerleri büyükten küçüğe doğru sıralandığında en uygun seçeneğe ulaşılmaktadır. l_{ij} değerleri toplamı "1" olmalıdır.

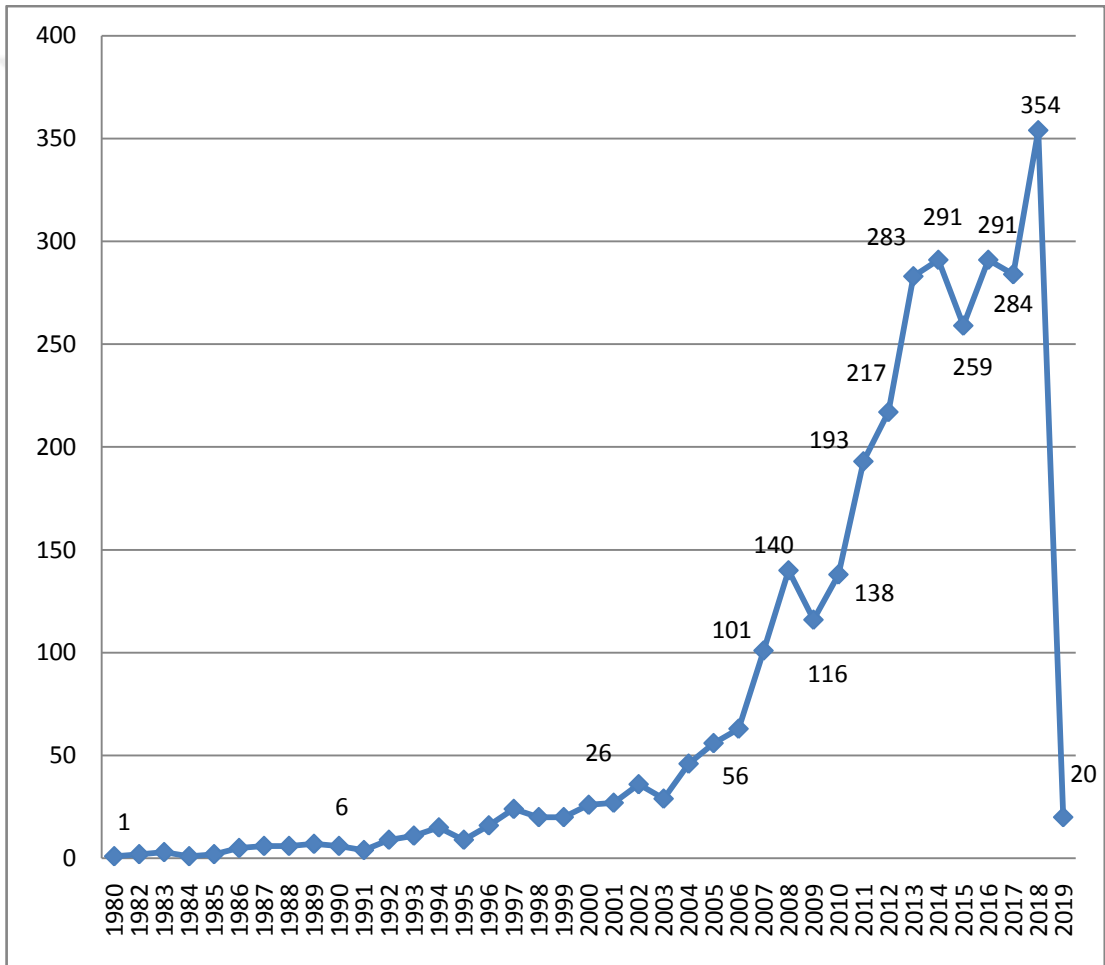
$$L = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ s_{m1} & s_{m2} & \dots & s_{mn} \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_n \end{bmatrix} = \begin{bmatrix} l_{11} \\ l_{21} \\ \cdot \\ \cdot \\ \cdot \\ l_{m1} \end{bmatrix}$$

3.1.2. AHP literatür taraması

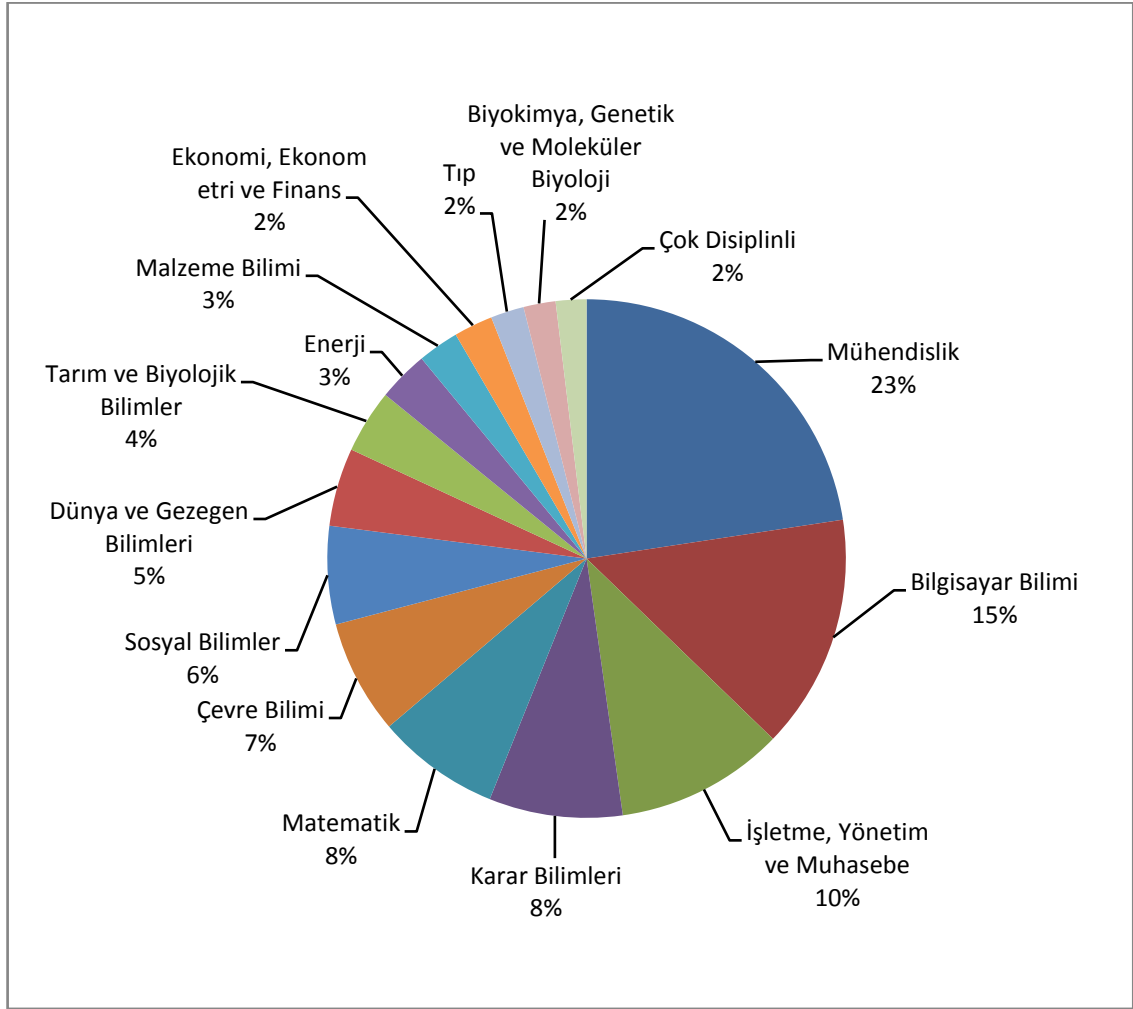
AHP, ilk olarak 1968 yılında Myres ve Albert tarafından ortaya atılmış ve Saaty (1977) tarafından geliştirilmiştir. AHP, Saaty tarafından geliştirilmesini izleyen dönemden başlayarak endüstriden politik kararlara kadar çok geniş birçok alanda uygulanmıştır (Timor, 2011).

Bu tezde literatür taraması yapılırken ve tez yazım aşamasında; AHP yönteminin kullanıldığı çalışmalar, TOPSIS yönteminin kullanıldığı çalışmalar, AHP ve TOPSIS yöntemlerinin birlikte kullanıldığı çalışmalar ve evlat edinme konusundaki çalışmalar, Yükseköğretim Kurulu (YÖK) Akademik ve Ulusal Tez Merkezi, Science Direct veri tabanı, Dergi Park ve çeşitli veri tabanlarından araştırılmıştır. Bu çalışmalar yayın yılı ve bilim alanlarına göre sınıflandırılmıştır.

Science Direct veri tabanındaki başlıklarda “AHP” konusu tarandığında ulaşılan makale sonuçlarının yıllara ve bilim alanlarına göre dağılımı aşağıdaki grafiklerde verilmiştir.



Şekil 3.3. Science Direct’te AHP konu başlığı ile ilgili yayınlamış makale sayılarının yıllara göre dağılımı.



Şekil 3.4. Science Direct'te AHP konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzde dağılımı.

Grafiklerdeki verilere bakılarak literatürde AHP yöntemi konusundaki çalışmaların 1980 yılından 2000'li yılların başına kadar yoğun olmadığı, 2000'li yıllardan itibaren genel olarak bakıldığında artış eğilimi gösterdiği görülmektedir. Ayrıca AHP yönteminin kullanıldığı bilim alanlarına bakıldığında yöntemin en fazla kullanıldığı bilim alanının Mühendislik (%23) olduğu, bunu Bilgisayar Bilimleri (%15) ve İşletme, Yönetim ve Muhasebenin (%10) izlediği görülmektedir.

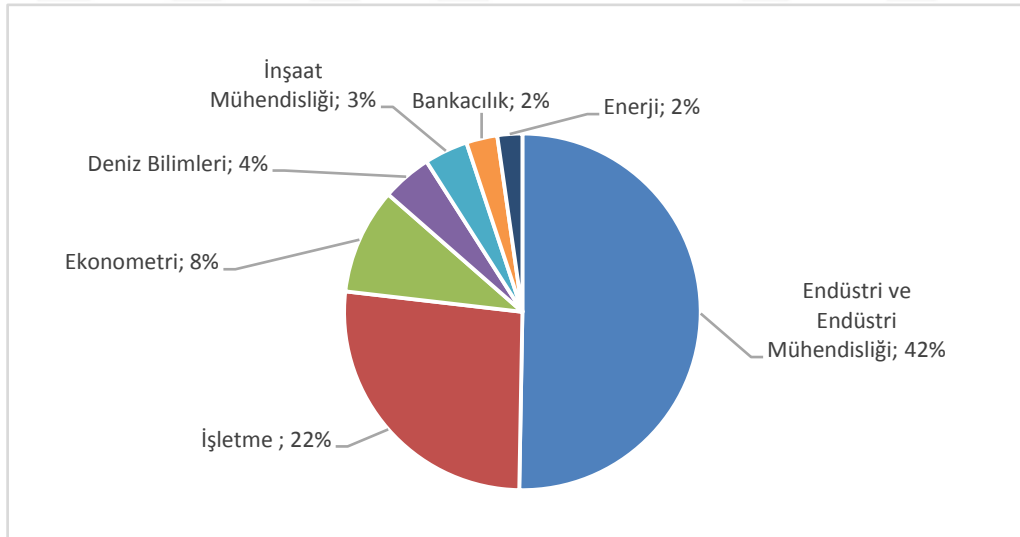
YÖK Ulusal Tez Merkezi veri tabanı incelendiğinde ilk kayıtlı tezin olduğu 1993 yılından 2018 yılına kadar olan dönemde AHP konulu 199 yüksek lisans tezine, 14 doktora tezine ulaşılmış ve bu tezlerin yıllara göre sayıları Çizelge 3.4'de verilmiştir.

Çizelge 3.4. YÖK Ulusal Tez Merkezi veri tabanındaki AHP konulu tezlerin yıllara göre sayıları.

TEZ TÜRÜ	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
Yüksek Lisans	1	0	1	1	1	0	0	1	2	1	3	5	3
Doktora												1	
TOPLAM	1	0	1	1	1	0	0	1	2	1	3	6	3

TEZ TÜRÜ	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Yüksek Lisans	7	8	9	11	12	15	14	13	23	17	17	18	16
Doktora			3		2	1		2		2		2	1
TOPLAM	7	8	12	11	14	16	14	15	23	19	17	20	17

YÖK Ulusal Tez Merkezi veri tabanında AHP konulu tezlerin bilim alanlarına göre tez yüzdeleri Şekil 3.5'teki grafikte verildiği gibidir.



Şekil 3.5. YÖK Ulusal Tez Merkezi veri tabanında AHP konulu tezlerin bilim alanlarına göre yüzde dağılımı.

AHP yöntemi ile ilgili en çok tez çalışması yapılan alan %42 oranı ile Endüstri Mühendisliğidir.

AHP yöntemi; kıyaslama, toplam kalite yöntemi, üretim, pazarlama vb. konularda çok ölçütlü karar problemlerinin çözümü için sıklıkla tercih edilmektedir. Tedarikçi seçimi ve değerlendirilmesi (Ghodsypour vd.,1996; Tam, ve Tummala, 2001; Dağdeviren vd., 2001; Koçak, 2003; Wang, vd., 2001; Murat ve Çelik, 2007; Çiftçioğlu, 2013), otomobil seçimi (Güngör ve İşler, 2005), ders seçimi (Dündar, 2008), hastane yeri seçimi (Akçalı, 2009), maliyet dağıtım anahtarı seçimi (Esmeray ve Tanç, 2009) arsa seçimi (Çetin vd., 2014), yüklenici firma seçimi gibi konularda uygulamaları mevcuttur. Yenilenebilir enerji projelerinin seçimi(Nigim vd. 2005), yazılım seçiminde (Lai vd., 2002), performans değerlendirme (Yaralıoğlu, 2001; Albayrak ve Erkut, 2005; Eraslan ve Algün, 2005; Girginer ve Kaygısız, 2009; Çetin ve Bıtrak, 2010) kredi değerlendirme (İç ve Yurdakul, 2000) ve yatırım değerlendirme (Kengpol, 2004; Tezcan vd., 2011) çalışmalarında da kullanılabilecek yöntemlerin başında AHP yöntemi gelmektedir (Şahin ve Akyer, 2011).

AHP yönteminin kullanılmaya başladığı zamandan itibaren literatürde yer alan bazı çalışmalar aşağıda verilmiştir.

Wind ve Saaty (1980), AHP'nin pazarlama alanındaki uygulamaları üzerine bir çalışma ele almıştır.Makale,yönetimi istenen hedef portföyün belirlenmesi ve bileşenlerinin arasında kaynakların tahsisi ile ilgilenen bir firmanın portföy kararları, yeni ürün geliştirme talimatlarının belirlenmesi ve pazarlama karması stratejilerinin oluşturulması ve değerlendirilmesi alanlarını kapsayan bir dizi açıklayıcı uygulamasının bir tartışmasıdır. AHP ve pazarlama uygulamaları hakkında ek araştırmalar için çeşitli öneriler vurgulanmıştır.

Suresh vd. (1991), en uygun doktora programının seçilmesi amacıyla AHP'yi etkin bir şekilde kullanmışlardır.

Chin vd. (1999),ISO 14001 tescili elde edebilmek için strateji geliştirme ve başarı faktörlerini değerlendirmede AHP yöntemini kullanmışlardır.

Hunjak ve Jakovčević (2001), AHP yöntemini kullanarak banka performansını değerlendirmişlerdir.

Samanta vd. (2002), çalışmalarında kömür ocaklarında kullanılacak ekipman seçimini AHP yöntemi ile yapmışlardır.

Güngör ve İşler (2005), AHP yöntemini kullandıkları çalışmalarında, otomobil alım sürecinde karar vermeyi kolaylaştırmak için bir uygulama yapmış ve uygulamanın sonucundan tüketici memnun kalmıştır.

İnsan Kaynakları alanının personel seçimi alanında Akyıldız (2006), bir bankaya iş başvurusunda bulunan 568 adayı, AHP yöntemi ile değerlendirilerek belirlenen ölçütlere göre sıralamış ve gerçek sıralamalar ile tutarlı sonuçların olduğunu tespit etmiştir.

Dündar ve Ecer (2008), Afyon Kocatepe Üniversitesi İktisadi ve İdari Bilimler öğrencilerinin GSM operatörü tercihinin AHP yöntemini kullanarak belirlemişlerdir.

Akçalı vd. (2009), Ankara’da kurulması düşünülen bir hastane için yer seçimi için uzman görüşlerini dikkate alarak belirledikleri ölçütlere göre, AHP ile bir modelleme yapmış ve önerilerde bulunmuşlardır.

Azadeh vd. (2010), bir demir madeninde yaptıkları çalışmada AHP yöntemi ile en uygun üretim yöntemini belirlemişlerdir. En önemli ölçütün maliyet olduğunu tespit etmişlerdir.

Öztürk vd. (2011), tekstil firmasında kullanılacak olan bir malzemenin satın alınacağı en uygun tedarikçiyi belirlemek için AHP yöntemini kullanmışlardır.

Ayan ve Pabuçcu (2013), AHP yöntemi ile Türkiye için yenilenebilir enerji kaynakları yatırımları arasında bir sıralama yapmışlardır.

Efe vd. (2013), hayvancılıkta AHP yönteminin kullanılabilirliğini araştırmışlardır. Dört farklı ırk ile yapılan çalışmalarda AHP yöntemi ile bu ırklar içinden en iyi süt veren ırk belirlenmiştir.

Longaray vd. (2015), 7 ana ölçüt ve bu ölçütler altında 17 alt ölçüte göre holding bünyesinde faaliyet gösteren bir şirketin holdinge sunduğu hizmetlerin kalitesini değerlendirmekte AHP yöntemini kullanmışlardır.

Kurşunoğlu ve Önder (2015), bir kömür ocağında AHP yöntemini kullanarak vantilatörler için en uygun olan fanı seçmişlerdir.

Wang vd. (2016), Çin’in Henan bölgesinde 2020 yılına kadar hedeflenen enerji tüketiminin kontrolü için kümeleme algoritması ve diferansiyel ayarlama yönetimi ile birlikte AHP kullanmışlardır. Çalışma sonucunda enerji yoğunluğunun düşük ve ekonomik şartların kötü olduğu bölgelerde asgari enerji tüketiminin karşılanması, enerji yoğunluğunun fazla olduğu yerler de ise enerji tüketiminin azaltılması gerektiğine kanaat getirilmiştir.

Akçay vd. (2018), tüketicilerin hangi tür et yemeyi tercih ettiklerini belirlemek için AHP yöntemini kullandıkları bir çalışma yapmışlardır. Tüketicilerin et tercihlerini

etkileyebileceği düşünülen dört ölçüt ve beş seçenek belirlenmiştir. En önemli ölçütün sağlık, en çok tercih edilen et türünün de balık eti olduğu tespit edilmiştir.

Özel ve Türkel (2018), işletme ölçütlerini temel alarak en uygun ERP programının seçilebilmesi için AHP yöntemi kullanmışlardır. Literatür taraması ve alan uzmanlarının önerileri ile temel ölçütler belirlenmiş ve AHP yöntemi kullanılarak ölçütlerin ağırlıklandırılması sağlanmıştır. Uygulamada karar matrisleri tasarlanırken gerçek ERP konusunda uzman görüşleri göz önünde bulundurulmuştur. Firmalar için ERP seçiminde hangi ölçütlerin daha önemli olacağı ve karar sürecinde ne şekilde bir yöntem izlenmesi gerektiği çalışmada sunulmuştur.

3.2. TOPSIS Yöntemi

TOPSIS yöntemi 1981 yılında Hwang ve Yoon tarafından ortaya çıkmıştır. ÇÖKV yöntemidir. Alternatif sayısı n , ölçütleri m 'dir (Yetkin, 2004).

TOPSIS yönteminde pozitif ideal çözüm, faydanın maksimum ve maliyetin minimum olduğu çözüm noktasıdır. Negatif ideal çözümde ise fayda minimum ve maliyet maksimumdur. TOPSIS yöntemi temelinde alternatifler arasında en çok tercih edilenin PIS'e en yakın mesafede olan ve NIS'e en uzak mesafede olan alternatif olduğu düşüncesi yatmaktadır (Özden, 2009).

3.2.1. TOPSIS uygulama süreci

TOPSIS yöntemi ile ilgili hesaplama aşamalarının detayları aşağıda adım adım verilmiştir (Özbek, 2013):

Adım 1: Karar Matrisinin (A) Oluşturulması

Karar matrisi, karar verme sürecinin en başında karar vericiler tarafından oluşturulan bir matristir. Karar matrisindeki satırlar seçenekleri, sütunlar da ölçütleri ifade etmektedir. a_{ij} değerleri i seçeneğinin j ölçütüne göre performansını ifade etmektedir. Karar matrisi aşağıda verildiği gibidir:

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

Adım 2: Standart Karar Matrisinin (R) Oluşturulması

Standart Karar Matrisi oluşturulurken normalizasyon işlemi yapılmaktadır. Karar matrisinin ölçütlerine ait değerler kareleri alınarak toplanmakta ve hesaplanan toplamın karekökü alınmaktadır. Sütundaki hesaplama yapılan değer karekökü alınan toplam değere bölünerek standart matris oluşturulmaktadır. Doğrusal normalizasyon işlemi için farklı yöntemler kullanılmaktadır. Vektör normalizasyonu işlemi (3.8) eşitliğinde verilmiştir.

$$r_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^m a_{kj}^2}} \quad (3.8)$$

Normalize işlemi yapıldıktan sonra elde edilen R matrisi aşağıda verildiği gibidir:

$$R_{ij} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix}$$

Adım 3: Ağırlıklı Standart Karar Matrisinin (V) Oluşturulması

Bu aşama için öncelikle ölçütlere ait ağırlıklar (w_i) belirlemektedir. Ölçüt ağırlıkları toplamı "1" olmalıdır.

$$(\sum_{i=1}^n w_i = 1).$$

R matrisi ile w_i değerlerinin çarpılması ile ağırlıklı standart karar matrisi oluşturulmaktadır. Oluşturulan ağırlıklı standart karar matrisi aşağıda verildiği gibidir:

$$V_{ij} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \dots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \dots & w_n r_{2n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ w_1 r_{m1} & w_2 r_{m2} & \dots & w_n r_{mn} \end{bmatrix}$$

Adım 4: İdeal (A^*) ve Negatif İdeal (A^-) Çözümlerin Oluşturulması

Oluşturulan ağırlıklı standart karar matrisinden iki ayrı sanal çözüm kümesi üretilmektedir. İdeal çözüm kümesinde en uygun değerler varken, negatif ideal çözüm kümesinde ise en kötü çözümler vardır.

Ölçüt fayda yönlü ise, ideal çözüm setinin oluşturulabilmesi için V matrisindeki ağırlıklandırılmış değerlendirme ölçütlerinin en büyükleri seçilmektedir. İdeal çözüm kümesinin bulunması (3.9) eşitliğinde gösterilmiştir:

$$A^* = \left\{ (\max_i v_{ij} | j \in J), (\min_i v_{ij} | j \in J') \right\} \quad (3.9)$$

(3.9) eşitliğinden hesaplanacak set $A^* = \{v_1^*, v_2^*, \dots, v_n^*\}$ şeklinde gösterilebilir.

Ölçüt maliyet yönlü ise, negatif çözüm setinin oluşturulabilmesi için V matrisindeki ağırlıklandırılmış değerlendirme ölçütlerinin en küçükleri seçilmektedir. İdeal çözüm setinin bulunması (3.10) eşitliğinde gösterilmiştir.

$$A^- = \left\{ (\min_i v_{ij} | j \in J), (\max_i v_{ij} | j \in J') \right\} \quad (3.10)$$

(3.10) eşitliğinden hesaplanacak set $A^- = \{v_1^-, v_2^-, \dots, v_n^-\}$ şeklinde gösterilebilmektedir.

Her iki eşitliğinde de J fayda (en büyükleme), J' ise kayıp (en küçükleme) değerini göstermektedir.

Pozitif ve negatif ideal çözüm kümesi, ölçüt sayısını ifade eden m adet elemandan oluşmaktadır.

Adım 5: Ayırım Ölçülerinin Hesaplanması

İdeal Ayırım S_i^* , Negatif İdeal Ayırım S_i^- şeklinde gösterilmektedir. J seçeneğinin Pozitif ideal çözüme uzaklığını ifade eden S_i^* değeri eşitlik (3.11)'de, negatif ideal çözümden uzaklığını ifade eden S_i^- değeri de eşitlik (3.12)'deki eşitlikte verildiği gibi hesaplanmaktadır.

$$S_i^* = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2} \quad (3.11)$$

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (3.12)$$

Hesaplama Öklid uzaklık yönteminden faydalanılmaktadır. Karar seçeneği sayısında S_i^* ve S_i^- değeri hesaplanmaktadır.

Adım 6: İdeal Çözüme Göreli Yakınlığın Hesaplanması

Bir önceki adımda hesaplanan S_i^* ve S_i^- değerleri kullanılarak her bir alternatif için pozitif ideal çözüme olan göreli yakınlık (C_i^*) hesaplanmaktadır. İdeal çözüme olan göreli yakınlık değerinin hesaplanması (3.13) eşitliğinde gösterilmiştir.

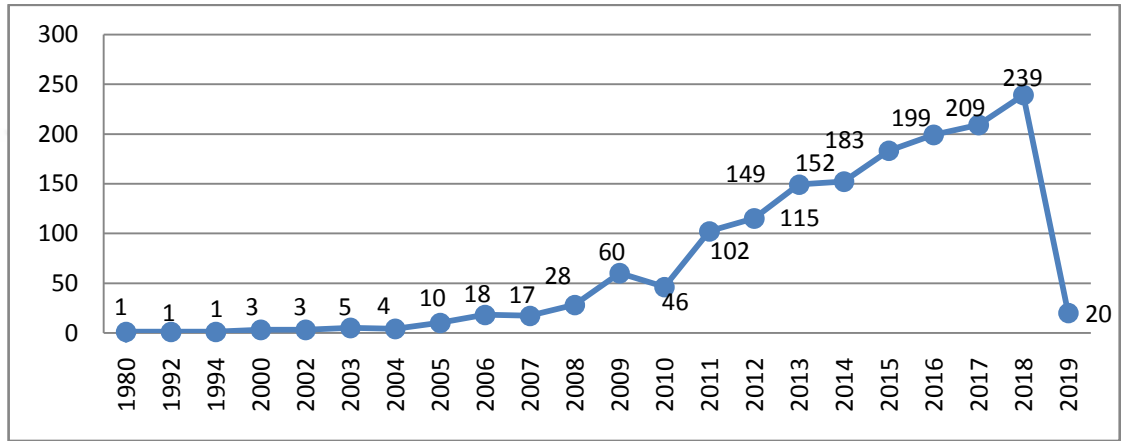
$$C_i^* = \frac{S_i^-}{S_i^- + S_i^*} \quad (3.13)$$

C_i^* değeri $0 \leq C_i^* \leq 1$ aralığında değer almakta ve $C_i^* = 1$ i . alternatifin pozitif ideal çözüm noktasında, $C_i^* = 0$ ise karar seçeneğinin negatif ideal çözüm noktasında bulunduğunu göstermektedir.

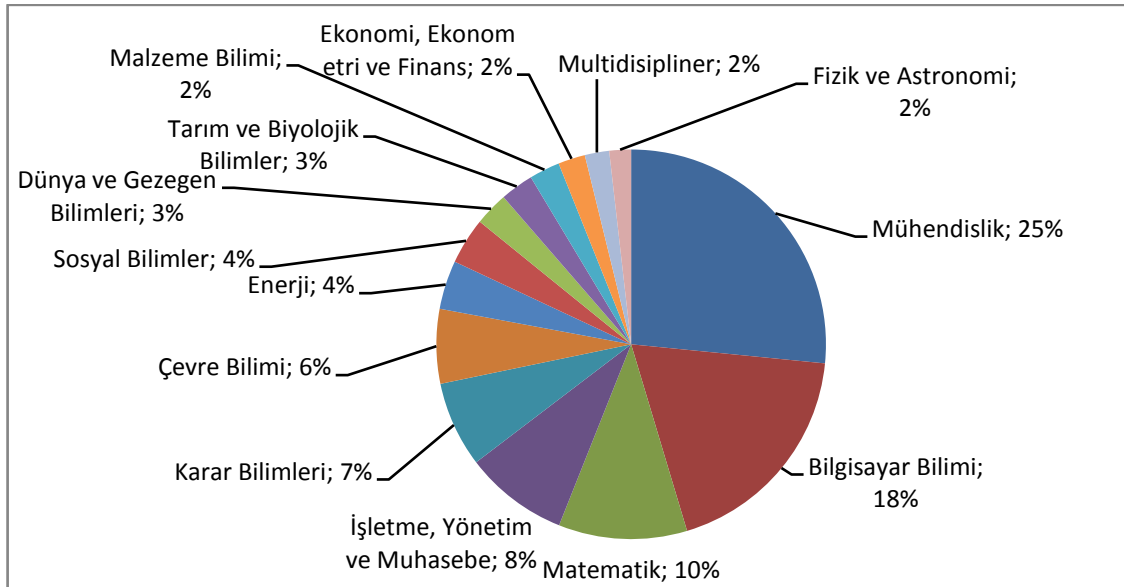
3.2.2. TOPSIS literatür taraması

ÇÖKV yöntemlerinden biri olan TOPSIS Yoon ve Hwang tarafından 1980 yılında geliştirilmiş ve birçok alanda uygulama imkanı bulmuştur.

Science Direct veri tabanındaki başlıklarda “TOPSIS” konusu tarandığında ulaşılan sonuçların yıllara ve bilim alanlarına göre dağılımı Şekil 3.6 ve Şekil 3.7'deki grafiklerde verilmiştir.



Şekil 3.6. Science Direct'te TOPSIS konu başlığı ile ilgili yayınlanmış makale sayılarının yıllara göre dağılımı.



Şekil 3.7. Science Direct'te TOPSIS konu başlığı ile ilgili yayınlanmış makalelerin bilim alanlarına göre yüzde dağılımı.

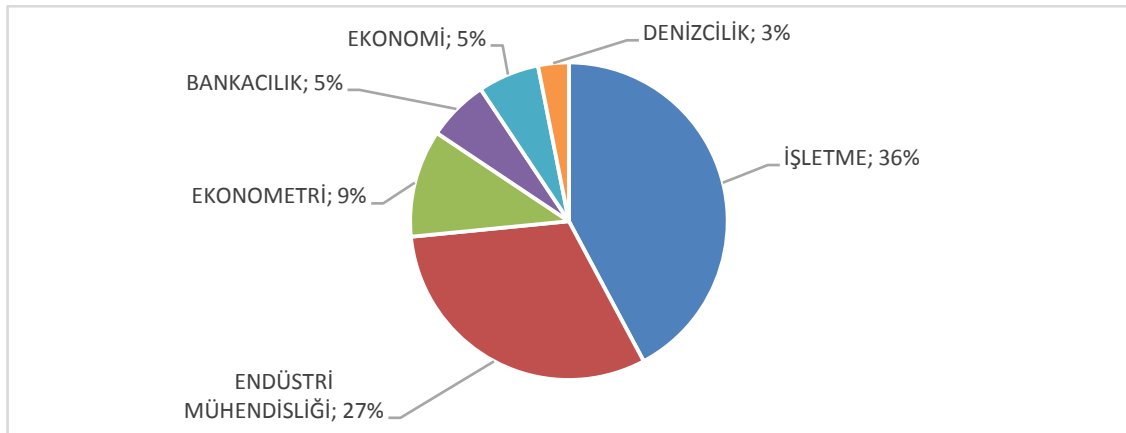
Grafiklerdeki verilere bakılarak literatürde TOPSIS yöntemi konusundaki çalışmaların 1980 yılından 2005 yılına kadar az sayıda olduğu, 2005 yılından itibaren genel olarak ivme kazandığı görülmektedir. Ayrıca TOPSIS yönteminin kullanıldığı bilim alanlarına bakıldığında yöntemin en fazla kullanıldığı bilim alanının Mühendislik (%25) olduğu, bunu Bilgisayar Bilimleri (%18) ve Matematiğin (%10) izlediği görülmektedir.

YÖK Ulusal Tez Merkezi veri tabanı incelendiğinde ilk kayıtlı tezin olduğu 2007 yılından 2018 yılına kadar olan dönemde TOPSIS konulu 70 yüksek lisans tezine, 5 doktora tezine ulaşılmış ve bu tezlerin yıllara göre dağılımları Çizelge 3.5'de verilmiştir.

Çizelge 3.5. YÖK Ulusal Tez Merkezi veri tabanındaki TOPSIS konulu tezlerin yıllara göre sayıları.

TEZ TÜRÜ	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018
Yüksek Lisans	1	1	2	6	2	5	6	9	4	7	15	12
Doktora					1	1		2		1		
TOPLAM	1	1	2	6	3	6	6	11	4	8	15	12

YÖK Ulusal Tez Merkezi veri tabanında TOPSIS konulu makale yüzdeleri de Şekil 3.8'deki grafikte verildiği gibidir.



Şekil 3.8. YÖK Ulusal Tez Merkezi veri tabanında TOPSIS konulu tezlerin bilim alanlarına göre yüzde dağılımı.

AHP yöntemi ile ilgili en çok çalışma yapılan alan %36 oranı ile İşletmedir. Bu sıralamayı %27 oranı ile Endüstri Mühendisliği takip etmektedir.

TOPSIS yöntemi kullanım kolaylığı, sonuçların kolaylıkla yorumlanabilmesi ve kolay anlaşılması gibi nedenlerden dolayı birçok alanda kullanım kolaylığı bulmaktadır (Behzadian vd., 2012). Örneğin, üçüncü parti lojistik firma seçiminde (Bottani ve Rizzi, 2006; Cao vd. 2007; Qureshi, 2007; Özbek ve Eren, 2013b), üçüncü parti tersine lojistik firmayı belirlemede (Kannan vd., 2009; Ravi, 2012), şehirlerin tahliyesinde sabit deprem sığınakları yeri belirlemede (Chu ve Su, 2012), maden yataklarının araştırılmasında kullanılan maden potansiyelinin haritalanmasında (Pazand vd., 2012), hizmet sağlayıcı seçiminde (Özbek ve Eren, 2013a), tedarikçi değerlendirmesinde ve seçiminde (Zeydan vd. 2011; Özbek, 2015), personel seçiminde (Xi ve Zhang, 2011) TOPSIS yöntemi kullanılmıştır.

Bunların dışında TOPSIS yönteminin kullanıldığı bazı çalışmalar aşağıda verilmiştir.

Deng vd. (2000), çalışmalarında Çin’de şirketlerin değerlendirmesi işlemlerinde kullanılan çeşitli finansal göstergelerin, şirketlerin performanslarının ölçülmesi, analizi ve yorumlanmasında TOPSIS metodunun kolay ve faydalı bir metot olduğunu belirlemiştir.

Li ve Qing-Sheng’in 2006 yılında yaptıkları çalışmada üretim alanında faaliyet gösteren şirketlerin gerçekleştirdikleri ihalelerde en iyi teklifin nasıl seçileceğini belirlemeye çalışmışlardır. Elektronik ürünler için gerçekleştirilen bir ihaleye katılan firmaların seçimi için TOPSIS yöntemi kullanılmıştır.

RuWu (2008), envanter planlaması üzerine bir çalışmada TOPSIS yöntemini kullanmıştır.

Eleren ve Karagül (2008), 1986-2006 yılları arasında Türkiye ekonomisinin performansını değerlendirmek için TOPSIS yönteminden yararlanmışlardır.

Ergül (2010), İMKB’de bir alanda işlem gören işletmelerin mali başarımlarının değerlendirilmesinde TOPSIS yöntemini kullanmıştır.

Shanian ve Savadogo (2006), TOPSIS yönteminden pillerde kullanılan malzemelerin seçimi konusunda yararlanmışlardır.

Yurdakul ve İç (2003), yaptıkları bir çalışmada Türkiye’de otomotiv sektöründe bulunan beş büyük ölçekli firmanın TOPSIS Yöntemi kullanılarak elde edilen performans puanları ile hisse senedi değerlerinin karşılaştırması ile ilgili bir çalışma yapmışlardır.

Alsu vd. (2018), farklı ülkelerde faaliyet gösteren katılım bankalarının performanslarını inceleyerek TOPSIS yöntemi ile bir sıralama yapmışlardır.

Yavuz (2012), TOPSIS yöntemini, Türkiye Kömür İşletmelerinden (TKİ) elde edilen veriler yardımıyla yükleyici seçiminde, Elmas firmasından elde edilen verilere göre ise Mermer Fabrikası için kuruluş yeri seçiminde kullanmıştır.

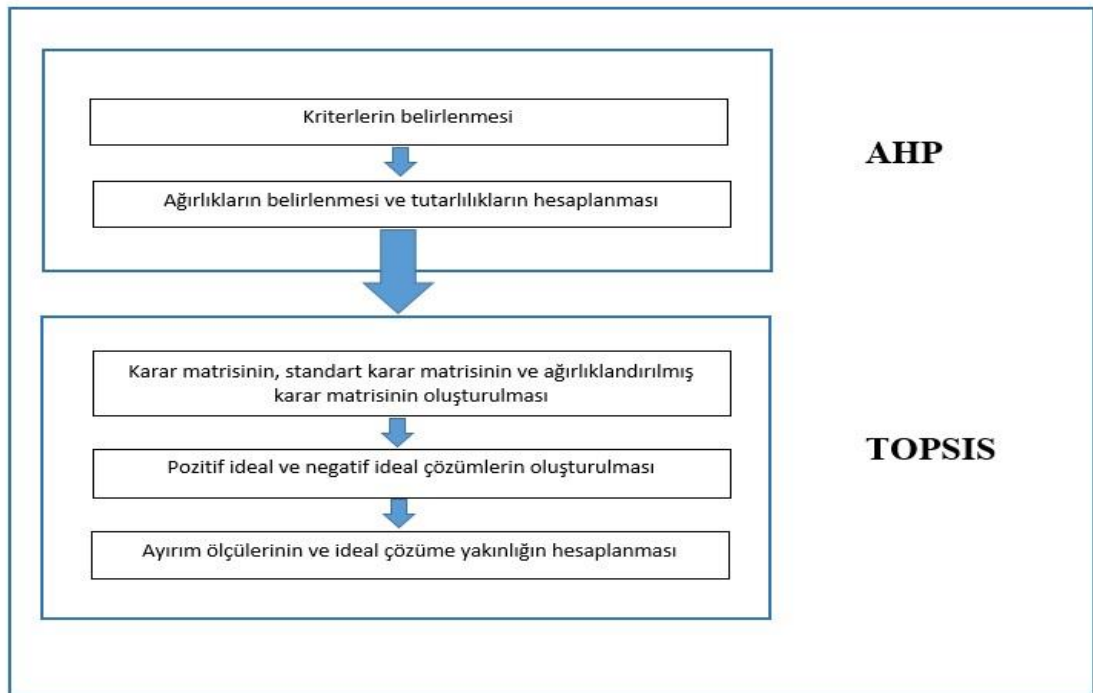
Atağan ve Yükçü'nün (2010) yaptığı çalışmada, işletme performansını değerlendirmek amacıyla TOPSIS yöntemi kullanılarak firmanın başka illerdeki otellerinin başarımları değerlendirilmiştir.

3.3. AHP ve TOPSIS Yöntemi

Bu çalışmada AHP ve TOPSIS yöntemleri birlikte kullanılmıştır. Problemin ölçütleri belirlendikten sonra önem derecelerine bakılarak ölçüt ağırlıkları ve tutarlılıkları AHP yöntemi ile hesaplanmıştır.

AHP yöntemi ile hesaplanan ölçüt ağırlıkları kullanılarak; karar matrisi, standart karar matrisi ve ağırlıklandırılmış karar matrisi oluşturulmuştur. Pozitif ve negatif ideal çözüm noktaları tespit edilerek, ayırım ölçüleri ve ideal çözüme yakınlık hesaplanmış ve TOPSIS yöntemi ile alternatif sıralaması yapılmıştır.

AHP ve TOPSIS yöntemlerinin bütünleşik olarak kullanıldığı melez yönteme ait işlemlerin akış şeması Şekil 3.9'da verildiği gibidir.



Şekil 3.9. AHP temelli TOPSIS yöntemi işlem akış şeması.

3.3.1. AHP ve TOPSIS uygulama süreci

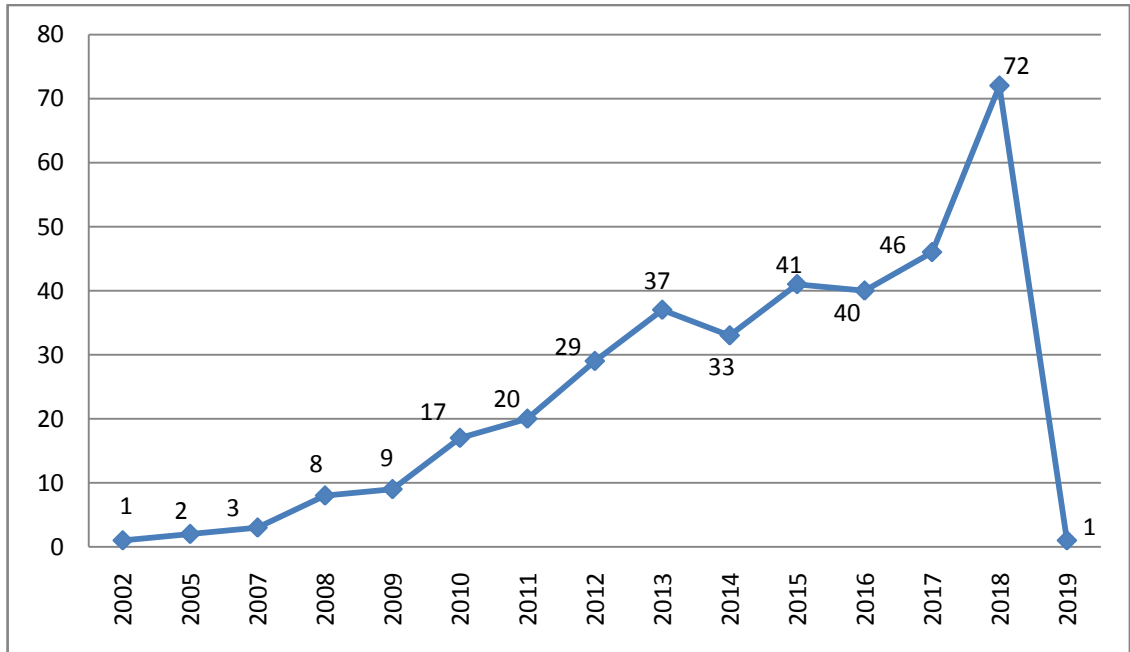
AHP ve TOPSIS yöntemlerinin birlikte kullanıldığı hibrit yönteme ait işlem adımları şu şekildedir:

- **Adım 1:** Problem Tanımlaması
- **Adım 2:** Hiyerarşik Yapının Kurulması
- **Adım 3:** İkili Karşılaştırma Matrislerinin Oluşturulması ve Ölçüt Ağırlıklarının Hesaplanması-Tutarlılıkların Hesaplanması
- **Adım 4:** Karar Matrisinin Oluşturulması
- **Adım 5:** Ayrım Ölçütlerinin ve İdeal Çözüme Göreli Yakınlığın Hesaplanması

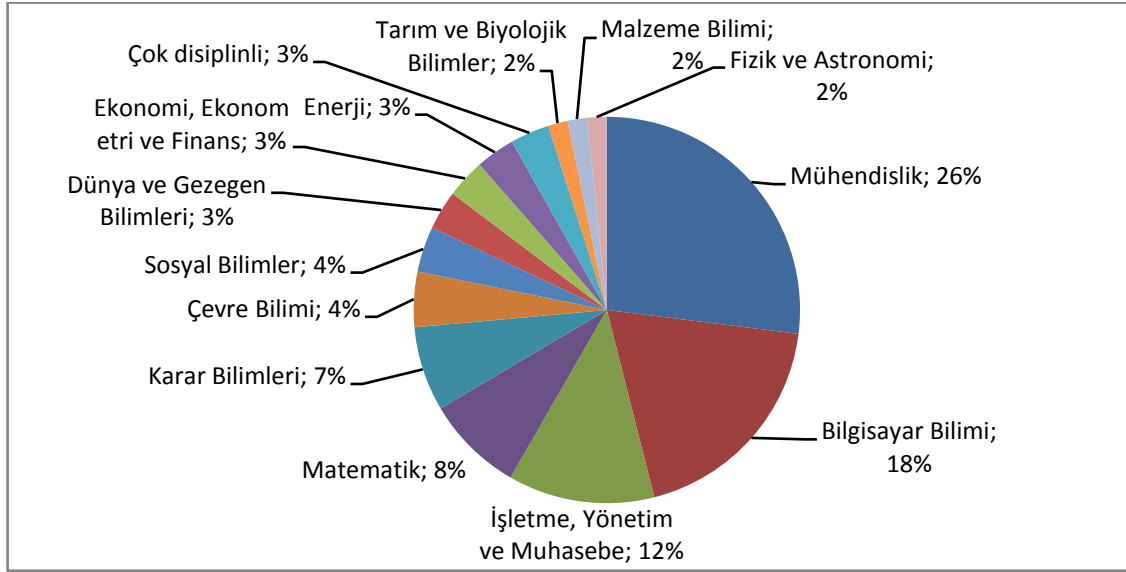
3.3.2. AHP ve TOPSIS literatür taraması

Literatür incelendiğinde AHP ve TOPSIS yöntemlerinin beraber kullanıldığı birçok çalışmaya rastlanmaktadır.

Science Direct veri tabanındaki başlıklarda “AHP-TOPSIS” konusu tarandığında ulaşılan sonuçların yıllara ve bilim alanlarına göre dağılımı Şekil 3.10 ve Şekil 3.11'deki grafiklerde verilmiştir.



Şekil 3.10. Science Direct’te AHP-TOPSIS konu başlığı ile ilgili yayınlamış makale sayılarının yıllara göre dağılımı.



Şekil 3.11. Science Direct'te AHP-TOPSIS konu başlığı ile ilgili yayınlamış makalelerin bilim alanlarına göre yüzde dağılımı.

Science Direct'ten elde edilen verilere bakıldığında AHP-TOPSIS yöntemlerinin birlikte kullanıldığı çalışmalara ilk olarak 2002 yılında rastlanmaktadır. Her yıl bu iki yöntemin birlikte kullanıldığı çalışmaların artarak devam ettiği görülmektedir. Diğer grafik yorumlandığında AHP-TOPSIS yöntemlerinin birlikte kullanıldığı bilim alanının Mühendislik (%26) olduğu, bunu Bilgisayar Bilimleri (%18) ve İşletme, Yönetim ve Muhasebenin (%12) izlediği görülmektedir.

YÖK Ulusal Tez Merkezi veri tabanında ilk kaydın 2008 yılında olduğu AHP-TOPSIS yöntemlerinin birlikte kullanıldığı tezlerin bilim alanlarına göre sayıları Çizelge 3.6'da verildiği gibidir. Veri tabanındaki tüm tez çalışmaları yüksek lisans çalışmalarıdır.

Çizelge 3.6. YÖK Ulusal Tez Merkezi veri tabanındaki AHP-TOPSIS konulu tezlerin yıllara ve bilim alanlarına göre sayıları.

BİLİM ALANLARI	2008	2010	2011	2012	2013	2014	2015	2016	2017	2018
Endüstri ve Endüstri Mühendisliği	1	1				1				2
İşletme			1	2	1	1	1			
Jeodezi ve Fotogrametri									1	
Denizcilik									1	
Bilgi Belge Yönetimi							1			
TOPLAM	1	1	1	2	1	2	2	0	2	2

Zanakis vd. (1998) karşılaştırmalı performansta, Tsaur vd. (2002) havayolu hizmet kalitesini değerlendirmek için yaptıkları çalışmada, Mahmoodzadeh vd. (2007) proje seçim probleminde kullanmıştır. Tzeng vd. (2005) toplu taşımada kullanılacak farklı yakıtla çalışan araç seçiminde, Rao (2006) malzemelerin işlenebilirliğini belirlemede; Rao ve Davim (2006) malzeme seçiminde, Işıklar ve Büyüközkan (2007) cep telefonu seçerken alternatifleri değerlendirmede, Fazlollahtabar (2010) ergonomik otomobil koltuğu tasarımında bu iki yöntemi birlikte kullanmışlardır.

Akyer ve Şahin'in (2011) çalışmalarında, 4x4 bir pikap araç için seçim ölçütleri belirlenmiş, belirlenen ölçütler AHP ve TOPSIS yöntemleri kullanılarak değerlendirilmiştir.

Nooramin vd. (2012) tarafından yayınlanan deniz konteynır sahası için en verimli liman vinci seçimi çalışması ÇÖKV yöntemleriyle değerlendirilmiştir. Bu çalışmada AHP ve TOPSIS yöntemleri kullanılmış ve saha seçimi kararına yarar sağlayacak veriler kullanılarak değerlendirme ve karşılaştırma yapılmıştır.

Gergin (2014) tarafından yapılan yüksek lisans tezi çalışmasında, Türkiye'deki bölgeler lojistik performanslarına göre AHP ve TOPSIS yöntemleri kullanılarak değerlendirilmiştir.

Kantemir ve Özkil'in (2015) yaptığı çalışmada, TSK'nin kullandığı savaş silah, araç ve gereçlerinin etkinlik dereceleri AHP ve TOPSIS yöntemleri kullanılarak belirlenmiştir.

Eren vd.'nin (2016) ele aldığı, çalışmada ders kitabı yazan bir yazarın kitabını bastırabileceği en uygun basımevi araştırılmıştır, uzman görüşlerine başvurulmuştur. Uzman görüşlerinin sonucunda ölçütler tespit edilmiş ve alternatifler TOPSIS yöntemi ile sıralanmıştır.

Karataş vd.'nin (2016) tıpta uzmanlık alan seçiminde hangi uzmanlık alanının daha çok tercih edildiğini belirlemek amacıyla yaptıkları araştırmaya, bir üniversitenin tıp fakültesinde görev yapan uzman doktorlar dâhil edilmiştir. Uzman doktorlar ile görüşmeler yapılmış ve ölçütler karşılaştırılmıştır. Doktorlar ile yapılan anketlerin sonuçları SPSS ile analiz edilmiştir. Uzmanlık alanı sıralamasının elde edilmesinde TOPSIS yönteminden yararlanılmıştır.

Ayan vd. (2017) tarafından yapılan çalışmada, bir devlet hastane servisine alınması planlanan tıbbi cihazı seçebilmek için AHP ve TOPSIS yöntemleri kullanılmış ve en uygun cihaz tespit edilmiştir.

Bozkurt vd. (2017) tarafından yapılan çalışmada, bir üniversitesinin eğitim fakültesine burs almak için başvuruda bulunan öğrenciler belirlenen ölçütlere göre değerlendirilmiştir.

Değerlendirme aşamasında burs alacak adayların tespit edilmesi için AHP ve TOPSIS yöntemleri kullanılmıştır.

Akman vd. (2018) tarafından yapılan çalışmada, bir uçak firmasının bünyesinde çalışan uzmanların görüşleri alınarak en ideal uçuş noktasının seçimi için ölçütler AHP yöntemi ile belirlemiştir. Belirlenen ölçütler dahilinde uçuş noktası olarak kullanılabilecek beş il TOPSIS yöntemi ile sıralanmıştır.

Herz (2018) tarafından yapılan yüksek lisans tezi çalışmasında, bir ilde faaliyet gösteren mobilya mağazalarının kalite seviyesi TOPSIS yöntemi kullanılarak sıralanmıştır.

3.4. Personel Seçimi Literatür Taraması

Bu çalışmada ele alınan konu personel seçim sürecine benzetildiği için personel seçim problemlerinin AHP ve/veya TOPSIS yöntemleri kullanılarak çözüldüğü çalışmalara ait literatür taraması yapılmıştır.

Chen (2000), bir yazılım şirketindeki mühendis ihtiyacını belirlediği beş farklı ölçütü göz önünde bulundurarak bulanık TOPSIS yöntemi ile sıraladığı adaylar arasından ilk sıradaki adayı işe alarak gidermiştir.

Boran (2009) yaptığı çalışmada; bilişim sektöründeki bir firmanın iş ilanlarına başvuran satış uzmanı adayları arasından seçim yapmak için TOPSIS yöntemini kullanmıştır.

Dursun ve Karsak (2010) yaptıkları çalışmada, personel seçimi problemlerinde uygulamak amacıyla sekiz farklı ölçütün kullanıldığı bulanık TOPSIS yöntemini önermişlerdir.

Doğan ve Önder (2014), insan kaynakları alanında personel seçimi işlemi için AHP ve TOPSIS yöntemlerini kullanmayı önermişlerdir. Ölçüt ağırlıklarını belirlemek amacıyla AHP yöntemini kullanmışlardır. TOPSIS yöntemi ile aday sıralaması yaparak en uygun adayı belirlemişlerdir.

Vatansever ve Öncel (2014) tarafından yapılan çalışmada, bir üniversiteye alınması planlanan araştırma görevlisinin belirlenmesi için bulanık AHP-TOPSIS yöntemlerinden yararlanılmıştır.

Hassani (2017), bir işletmeye alınacak personel için başvuruları TOPSIS yöntemi ile değerlendirmiş ve buna göre 25 adayı belirlenen ölçütlere göre sıralamıştır. Kriterlerin sayısallaştırılması için Likert ölçeğini kullanmıştır.

Aydemir vd. (2017), otomotiv sektöründeki bir işletmede, en uygun personelin en kısa zamanda seçilmesi amacıyla bir çalışma yapılmıştır. İşletmeye başvuran adaylar TOPSIS yöntemi kullanılarak sıralanmıştır. Ölçütler ağırlıklandırılırken 100'lük değerlendirme sistemi ile her bir adayın ilgili ölçütü 100 üzerinden hesaplanarak hesaplamalar yapılmıştır.

Çetin ve İçigen (2017) tarafından yapılan çalışmada, otelde çalışacak personelin seçimini yapmak için AHP yöntemi kullanılarak ölçüt ağırlıkları hesaplanmıştır. Başvuran adayların belirttikleri bilgiler değerlendirilerek TOPSIS yöntemi ile bir sıralama oluşturulmuştur.

İlgaz (2018), tarafından yapılan çalışmada lojistik sektöründe personel seçim ölçütlerinin önem düzeyleri AHP yöntemi ile belirlenmiştir. Lojistik şirketine başvuran beş aday arasından işletmeye uygun doğru adayın seçilmesi için TOPSIS yöntemi kullanılmıştır. Analiz sonucunda şirket için en uygun aday önerilmiştir.

Personel seçiminde kullanılan çeşitli ÇÖKV teknikleriyle birlikte AHP ve TOPSIS yöntemlerinin kullanıldığı çalışmaları görmek için Demirci'nin (2018) yapmış olduğu yüksek lisans tezi incelenebilir.

4. EVLAT EDİNME SÜRECİNDE BİR UYGULAMA

4.1. Uygulama Alanı

Evlat edinme kavramı yüzyıllardır farklı amaçlarla yaşamda var olan bir kavramdır. Daha önceleri soyun devam ettirilmesine, mirasçı sahibi olmaya, yaşlı kimselerin kendilerine baktırma düşüncesine hizmet ederken, 19.yy'da çocuk sevgisi yaşamayan bireylere bu sevinci yaşatma amacına yönelmiş, günümüzde ise korunmaya muhtaç çocuklara destek olma düşüncesi ön plana çıkmıştır. Geçmişte sosyal ve duygusal amaçlar yanında dini, siyasi, iktisadi amaçlar için de kullanılmışken, evlat edinmede artık amaç; terk edilmiş, yetim ya da evlilik dışında doğmuş çocuğa bir nesep statüsü ve sıcak bir aile ortamı kazandırmak, sağlıklı yetiştirme ve sosyokültürel kişiliğini geliştirme imkanlarına kavuşmalarını ve sosyal-hukuki statü kazanmalarını sağlamaktır (Giray, 2001).

Toplumun temel yapı taşı ailedir. Genel anlamda bakıldığında aileyi oluşturan bireyler anne, baba ve çocuklardır. Çeşitli sebeplerle evlat sahibi olamayan insanlar ile yine bir çok nedenden dolayı ailesinin yanında büyüme imkanı bulamayan çocuklar evlat edinme işlemi sayesinde birbirlerini bulmaktadır. Bu sayede, evlat özlemi çeken ebeveynler bir evlada, aile özlemi çeken küçükler ise bir aileye kavuşmaktadır. Sağlıklı bir aile ortamında yuva sıcaklığı ile büyüyen çocuklar topluma kazandırılarak hayatlarına devam edebilmekte ve ilerleyen yıllarda topluma da faydalı bireyler olarak yaşamlarını sürdürebilmektedir.

Sahipsiz, yardıma muhtaç, aile şefkatinden uzak büyüyen tüm çocuklardan ait oldukları toplum sorumludur. Bu sebeple korunmaya muhtaç çocukların durumları daha detaylı olarak irdelenmelidir.

Literatürde yapılan çalışmalar detaylı olarak incelendiğinde korunmaya muhtaç çocukların evlat edindirilmesi ile ilgili yapılan çalışma alanlarının kısıtlı olduğu görülmektedir. Bu alanlar ülkemizde genellikle hukuk ve din alanlarıdır. Hukuk alanında yapılan çalışmalar miras hukuku alanında yoğunlaşmıştır. Bunun yanı sıra evlat edinmede hukuki sürecin nasıl işlediğine dair de çalışmalar mevcuttur. Din alanında evlat edinmenin diğer dinlere ve İslamiyet'e göre doğruluğu tartışılmıştır. Tıp, psikoloji gibi alanlarda da çalışmalara rastlanmıştır. Detaylı literatür taraması çalışmada ayrı bir başlık altında sunulmuştur.

Evlat edinme konusunun araştırılması esnasında evlat edinmek için kuruma başvuran ailelerin/bireylerin sadece başvurduktan sonra istenilen evrakların teslim tarihine göre sıraya alındığı tespit edilmiştir. Başvuranların sıralamalarındaki belirleyici tek ölçüt evrak teslim

tarihidir. Bir çocuğun hayatını belirleyen böylesi bir konuya birden fazla ölçüt eşliğinde yaklaşılması gerektiği, daha sistematik bir çalışma ile evlat edindirilecek küçüğün ölçütleri en uygun olan aileye verilmesi gerektiği düşünülmüştür. Literatürde bu alanda yapılmış bir çalışma olmaması konunun ele alınmasının en önemli sebeplerinden biridir.

Çalışmada, AHP ve TOPSIS yöntemleri kullanılarak sıralama yapan ve en uygun aile/birey seçiminin yapılmasını sağlayan bir bilgi sistemi oluşturmak amaçlanmıştır.

4.2. Önerilen Yöntem

Bu çalışmada, ölçüt ağırlıklarının hesaplanmasında AHP yönteminin kullanılması önerilmiştir. AHP yönteminin önerilmesinin sebeplerinden bir kaç tanesi şöyle sıralanabilir:

- AHP, çok sayıda ölçüt ile uygulanabilir.
- İleri seviye yöntemler gerektirmediği için diğer yöntemlere göre uygulaması daha kolaydır.
- Nicel ve nitel faktörleri değerlendirebilir.
- Ağır matematiksel hesaplamalar içermez.
- İkili karşılaştırmalarda tutarlılık derecesini ölçebilir.
- Grup kararlarının alınması için uygun bir yöntemdir.

AHP yönteminin olumlu özelliklerinin yanı sıra olumsuz özellikleri de bulunmaktadır. AHP yöntemi ikili karşılaştırmalar yapmakta ve alternatif sayısının az olduğu problemlerde kısa sürede çözüm vermektedir. Örneğin n ölçüt ve m alternatiften oluşan bir problemde $n.m.(m-1)/2$ ikili karşılaştırma yapılması gerekmektedir (Dağdeviren, Yavuz ve Kılınç, 2009). Alternatif sayısının çok olduğu problemlerde çok sayıdaki ikili karşılaştırmadan kaçınmak ve işlem sayısını aza indirgeyip kısa sürede çözüme ulaşmak için, çalışmada AHP yönteminin sadece ölçüt ağırlıklarının belirlenmesinde kullanılması önerilmiştir. Evlat edinmek için Bursa ili Aile, Çalışma ve Sosyal Politikalar Bakanlığı İl Müdürlüğüne yılda ortalama yüz yazılı başvuru yapılmaktadır. Bu başvuruların kabul sayısı her yıl değişkenlik gösteriyor olsa da yarıdan fazlası uygun olduğu için değerlendirmeye alınmaktadır. Ele alınan problemde başvuran ailelerin sıralanması esas konu olduğu ve alternatif aile/birey sayıları fazla olduğu için sıralama aşamasında AHP yönteminin kullanılması işlemlerin uzun sürmesi nedeniyle doğru olmayacaktır. Bu nedenle, alternatiflerin önem derecelerine göre sıralanmasında TOPSIS yönteminin kullanılması önerilmiştir. TOPSIS yönteminin önerilmesinin sebeplerinden bir kaç tanesi şöyle sıralanabilir (Wang ve Chang, 2007):

- TOPSIS yönteminin içeriği yalın, güvenilirliği yüksektir.
- Hesaplama yeteneği güçlüdür.
- Hesaplama süreci basittir.
- Alternatiflerin değerlendirilmesi basit bir matematiksel formdadır.
- TOPSIS yöntemi, tüm ölçütleri basit matematiksel formlarla birleştirmesiyle en iyi alternatifin aranmasına olanak sağlamaktadır.

4.3. Önerilen Bilgi Sistemi

Evlat edinme işlemi esnasında aileler/bireyler başvuru evraklarını tamamlayarak kuruma teslim ettikleri gün sıraya alınmakta ve bu sıralama işlemi sadece evrak teslim tarihine bakılarak yapılmaktadır. Ancak bir küçüğün hayatına daha iyi şartlarda ve sağlıklı bir aile ortamında devam edebilmesi için küçüğe her anlamda en iyi şartları sunabilecek aile seçiminin yapılması gerekmektedir. Çeşitli sebeplerden dolayı evlat sahibi olamayan bir çok aile/birey kurumlara başvurarak evlat hasretini gidermek istemektedir. Bu aileler/bireyler birbirlerinden farklı özelliklere ve imkanlara sahiptirler. Bir çocuğun ve ailenin hayatını etkileyen bu kadar önemli bir konuda sıralama açısından sadece kuruma evrakların teslim edildiği tarihi baz alarak önce başvuran aileye küçüğü evlat olarak vermek, belki de bir sonraki gün başvuran ve küçük için daha uygun şartları olan başka bir aileyi/bireyi bu imkandan mahrum etmek doğru değildir. Böylesine ciddi ve hassas bir konuda sıralama yapmak için birden fazla ölçüte bakılmalıdır. Bir bilgi sistemi ile evlat edinme sürecindeki sıralama işlemi daha sistematik hale getirilmelidir.

Ele alınan konu için yapılan görüşmelerde elde edilen bilgilere göre evlat edinmek için sıralama yapılması aşamasında dikkate alınması gereken ölçütler genel hatlarıyla aşağıda verildiği gibi olmalıdır:

1. Ailenin ekonomik durumu
2. Bekleme süresi
3. Ebeveynlik kapasitesi
4. Ailenin eğitim durumu
5. Ailenin iş durumu
6. Ailenin sağlık durumu
7. Konutun fiziki şartları
8. Ailenin sosyal yaşantısı
9. Ailenin diğer insanlarla iletişimi
10. Çocuklu yaşama dair beklentiler
11. Aile bireylerinin kişisel özellikleri

12. Evlilik yaşantısı, eşler arasındaki iletişim

13. Sosyal çalışmacının genel kanaati

Ölçütler değerlendirilirken, her bir ölçüt için değerlendirilen özellikler aşağıda verildiği gibidir.

Ailenin ekonomik durumu: Ailenin gelir düzeyi, ailenin aylık giderleri, gelir gider dengesi, ailenin ikamet ettiği bölgeye göre gelirlerinin yeterlilik düzeyi değerlendirilmektedir.

Bekleme süresi: Evrak teslim tarihinden itibaren ailenin beklediği süre değerlendirilmektedir. 0-12 ay arası 1 puan, 13-24 ay arası 2 puan, 25-36 ay arası 3 puan, 37-48 ay arası 4 puan, 49 ay ve daha fazlası 5 puan olarak belirlenmiştir.

Ebeveynlik kapasitesi: Ebeveyn adaylarının bir çocuğun sorumluluğunu alabilme kapasitesidir. Çocuğa yaklaşımı, çocuğun gelişim evrelerini ne kadar bildiği gibi konular değerlendirilmektedir.

Ailenin eğitim durumu: Ebeveyn adaylarının eğitim durumları, mezuniyet dereceleri, kendilerini eğitim hayatında ne kadar geliştirebildikleri ve eğitim konusuna yaklaşımları değerlendirilmektedir.

Ailenin iş durumu: Ebeveyn adaylarının iş yerlerindeki memnuniyetleri, ne sıklıkla iş değiştirdikleri, iş ortamındaki insanlarla olan iletişim düzeyleri değerlendirilmektedir.

Ailenin sağlık durumu: Ebeveyn adaylarının kendi hayatlarını ve evlat edinecekleri çocuğun hayatını idame ettirebilmesi için herhangi bir sağlık engellerinin olup olmadığı değerlendirilmektedir.

Konutun fiziki şartları: Ebeveyn adaylarının ikamet ettikleri evin temizlik durumu değerlendirilmektedir. Evin hastane, okul, sağlık ocağı gibi yerlere yakınlığı, evdeki prizlerin konumları, sivri köşeli mobilyaların varlığı ve konumu, evin rutubet durumu, oda sayıları, çocuğa ait bir oda olup olmayacağı gibi konular değerlendirilmektedir.

Ailenin sosyal yaşantısı: Aile içinde doğum günleri, evlilik yıl dönümleri, anneler-babalar günü vb. özel günlerin kutlanma durumu değerlendirilmektedir. Piknik, tatil ve bu gibi sosyal aktivitelere katılım durumları değerlendirilmektedir. Tiyatro, sinema, konser vb. gitme durumları değerlendirilmektedir.

Ailenin diğer insanlarla iletişimi: Ebeveyn adaylarının çevrelerindeki insanlarla iletişimleri, diğer insanların ebeveyn adayları ile ilgili olan düşünceleri değerlendirilmektedir.

Çocuklu yaşama dair beklentiler: Evlat edinilen çocuğun etnik kökeninin ebeveyn adayları için önemli olup olmaması, bir çocuğun ailelerine katılımından sonra aileyi ne şekilde etkileyeceğine dair düşünceleri değerlendirilmektedir.

Aile bireylerinin kişisel özellikleri: Ebeveyn adaylarının kişilik özellikleri değerlendirilmektedir. Çocuk yetiştirmek için gerekli özveriyi gösterebilme durumları, yeniliklere açık olup olmadıkları, olaylara yaklaşımları gibi durumlar değerlendirilmektedir.

Evlilik yaşantısı, eşler arasındaki iletişim: Ebeveyn adaylarının birbirleri ile ilgili düşünceleri, her birinin diğerine yaklaşımı ve tutumu, eşler arasındaki sevginin ve saygının varlığı değerlendirilmektedir. Eğer birey olarak başvuru yapılmış ise kişinin en yakın aile bireyleri ile ilgili iletişimleri değerlendirilmektedir.

Sosyal çalışmacının genel kanaati: Evlat edinme sürecinde yapılan tüm görüşmeler esnasında sosyal çalışmacıda oluşan genel kanaat değerlendirilmektedir.

Belirlenen ölçütler ve ölçüt ağırlıkları da dikkate alınarak oluşturulan bilgi sisteminin her bir aşaması çalışmanın devam eden aşamasında sunulmuştur.

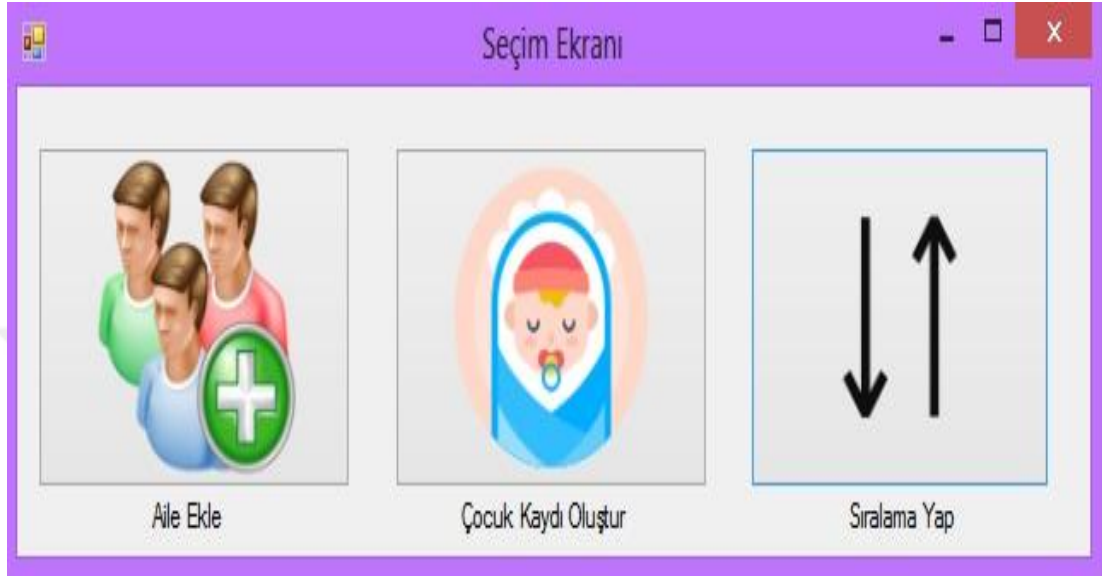
Hazırlanan bilgi sistemi C# yazılım dili kullanılarak oluşturulmuştur. Bilgi sisteminin oluşturulması aşamasında Pamukkale Üniversitesi Yönetim Bilişim Sistemleri öğrencisi Emre Kuzu tarafından yapılan tez çalışmasının uygulama kısmından faydalanılmıştır. Emre Kuzu tarafından yazılan program kodları incelenmiş ve bilgi sisteminin oluşturulması aşamasında bu kodlardan yararlanılmıştır.

Bilgi sistemine "Kullanıcı Adı" ve "Şifre" bilgilerinin yer aldığı bir ekran ile giriş sağlanmaktadır. Belirlenen ölçütler, ölçütlerin birbirlerine göre önem dereceleri ve ebeveyn adaylarının ölçüt puanları kullanılarak oluşturulan AHP-TOPSIS temelli bilgi sistemine giriş ekranı Şekil 4.1'de verildiği gibidir.



Şekil 4.1. AHP-TOPSIS temelli bilgi sistemi giriş ekranı.

Bilgi sistemine giriş yapıldıktan sonra "Seim Ekranı" açılmaktadır. Oluşturulan seim ekranında "Aile Ekle", "Çocuk Kaydı Oluştur" ve "Sıralama Yap" seenekleri bulunmaktadır. Seim ekranı Şekil 4.2'de verilmiştir.



Şekil 4.2. Seim ekranı.

Seim ekranındaki "Aile Ekle" sekmesinde, ebeveyn adaylarının bilgileri ve her bir ölçüt için sahip oldukları puanların yazılacağı alanlar bulunmaktadır. Bu ebeveyn bilgileri şu şekildedir: anne ve baba adaylarının ayrı ayrı T.C. kimlik numaraları, doğum tarihleri, doğum yerleri ve evlilik tarihleri.

Ölçütler ise şu şekilde verilmiştir: ailenin ekonomik durumu, bekleme süresi, ebeveynlik kapasitesi, ailenin eğitim durumu, ailenin iş durumu, ailenin sağlık durumu, konutun fiziki şartları, ailenin sosyal yaşantısı, ailenin diğer insanlarla iletişimi, çocuklu yaşama dair beklentiler, aile bireylerinin kişisel özellikleri, evlilik yaşantısı, eşler arasındaki iletişim, sosyal çalışmacının genel kanaati.

"Aile Ekle" butonuna tıklanıldığında ekrana gelen "Yeni Aile Ekle" ekranı Şekil 4.3'te verilmiştir. Ebeveyn adaylarına ait bilgiler ve ölçüt puanları hayali üretilen değerlerdir. Ölçüt puanları 5'li Likert ölçeği baz alınarak 1-5 arasındaki değerler olarak puanlanmıştır. Aileler/bireyler ile yapılan görüşmeler sonucunda verilen puanlar "Yeni Aile Ekle" ekranında aile/birey bilgileri ile birlikte kaydedilmektedir.

Yeni Aile Ekle

TC Kimlik :

İsim :

Soyisim :

Doğum Yeri :

Doğum Tarihi : 9. 6.2019

Evlenme Tarihi : 9. 6.2019

TC Kimlik :

İsim :

Soyisim :

Doğum Yeri :

Doğum Tarihi : 9. 6.2019

Ailenin Ekonomik Durumu :

Bekleme Süresi :

Ebeveynlik Kapasitesi :

Ailenin Eğitim Durumu :

Ailenin İş Durumu :

Ailenin Sağlık Durumu :

Konutun Fiziki Şartları :

Ailenin Sosyal Yaşamı :

Ailenin Diğer İnsanlarla İletişimi :

Çocukluk Yaşama Dair Beklentiler :

Aile Bireylerinin Kişilik Özellikleri :

Evlilik Yaşamı/Eğler Arası İletişim :

Sosyal Yaşamının Genel Karneati :

Ekle

Yenile

Sil

No	Baba TCKN	Baba Adı	Baba Soyadı	Baba Doğum Tarihi	Baba Doğum Yeri	Anne TCKN	Anne Adı	Anne Soyadı	Anne Doğum Tarihi	Anne Doğum Yeri	Evlilik Tarihi
12	12111111111	Ahmet	Aslan	15.1.1978 10:51	Bursa	11111111111	Ayşe	Aslan	20.6.1980 10:51	Eskişehir	5.7.2002 10:51
13	12111111112	Can	Yılmaz	1.3.1980 10:51	İstanbul	11111111112	Emel	Yılmaz	12.2.1983 10:51	İstanbul	15.5.2004 10:51
14	12111111113	Hakan	Yalçın	12.5.1975 10:51	Denizli	11111111113	Ceyda	Yalçın	10.3.1975 10:51	Denizli	15.5.1998 10:51
15	12111111114	Hasan	Nesin	17.2.1975 10:51	Ardeşan	11111111114	Nalan	Nesin	23.11.1981 10:51	Kars	21.9.2000 10:51
16	12111111115	Taner	Narzan	1.12.1986 10:51	Bilecik	11111111115	Cansu	Narzan	25.1.1987 10:51	Bursa	10.6.2007 10:51
17	12111111116	Mustafa	Şeker	1.10.1981 10:51	Burdur	11111111116	Nazlı	Şeker	11.2.1982 10:51	Burdur	10.6.2007 10:51
18	12111111117	Sami	Akşehir	2.4.1971 10:51	Afyon	11111111117	Ayşe	Akşehir	15.5.1972 10:51	Afyon	10.6.2007 10:51
19	12111111118	Hüseyin	Bilgeç	7.7.1975 10:51	Bursa	11111111118	Gülüm	Bilgeç	14.2.1981 10:51	Bursa	5.7.2005 10:51
20	12111111119	Eran	Yosun	9.3.1981 10:51	Muğla	11111111119	Tansu	Yosun	5.6.1982 10:51	Ankara	17.5.2008 10:51
21	12111111121	Hilmi	Candan	1.3.1972 10:51	Manisa	11111111121	Nuriye	Candan	4.5.1974 10:51	İzmir	11.4.1997 10:51

Geri Çık

Şekil 4.3. Ebeveyn adaylarının bilgilerini ekleme ekranı.

5'li Likert ölçeği değerleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1. 5'li likert ölçeği.

DEĞERLER	PUANLAR
Çok İyi	5
İyi	4
Orta	3
Kötü	2
Çok Kötü	1

Seçim ekranındaki "Çocuk Kaydı Oluştur" sekmesinde kurumda kalan çocukların kayıtları tutulmaktadır. "Yeni Çocuk Kaydı" sekmesinde kaydedilmesi gereken bilgiler şu şekildedir: T.C. kimlik numarası, adı soyadı, doğum tarihi, cinsiyeti, durumu, evlat edinmeye uygunluğu.

Evlat edinme işlemi sürecinde başvuran ailelerden/bireylerden istenilen bir çok ölçüt bulunmaktadır. Bunun yanı sıra kuruma evlat edinmek için başvuran ebeveyn adaylarının da evlat edinecekleri küçükte aradıkları özellikler olabilmektedir. Bu özelliklerden en çok dikkat

edilenleri küçüğün cinsiyeti, kuruma verilme durumu ve yaşıdır. Devlet koruması altındaki her küçük evlat edinmeye uygun değildir. Sağlık problemleri, ailevi durumları, evlat edinilmeyi reddetmeleri gibi bir çok sebepten dolayı küçükler evlat edindirilmeye uygun olmayabilmektedir. Bu sebeplerden dolayı bahsedilen konulara ait detaylar sisteme kaydedilmelidir ve kayıt ekranı bu bilgiler ışığında hazırlanmıştır. "Yeni Çocuk Kaydı" ekranı Şekil 4.4'te verilmiştir.

ID	TC	AdSoyad	DoğumTarihi	Cinsiyet	Durum	Uygun
6	13111111111	Umut Onur	5.2.2018 11:41	☒	☒	☒
7	13111111112	Elif Çimen	12.4.2015 11:41	☐	☐	☒
8	13111111113	Nisan Derya	1.1.2018 11:42	☐	☒	☒

Şekil 4.4. Yeni çocuk kayıt ekranı.

Ölçütlerin bulunduğu ekran Şekil 4.5'te verilmiştir.

Şekil 4.5. Ölçütler ekranı.

Uygulamada ölçüt ağırlıklarını belirlemek için AHP yöntemi kullanıldığı için Şekil 4.5'de yer alan ölçütler ekranındaki "Başlat" butonuna tıklanıldıktan sonra ekrana "AHP" butonu gelmektedir. AHP butonu uygulama için uygun olan "Önem Matrisi"ni ekrana getirmektedir. Önem matrisi ilk olarak doldurulmamış şekilde ekrana gelmektedir. Matristeki her bir ölçütün diğer ölçütlere göre önem dereceleri tek tek yazılabilmektedir. Önem matrisinin boş hali Şekil 4.6'da verilmiştir.

AHP-TOPSIS Bilgi Sistemi

AİLENİN EKONOMİK DURUMU - (Maximum)
BEKLEME SÜRESİ - (Maximum)
EBEVEYNLIK KAPASİTESİ - (Maximum)
AİLENİN EĞİTİM DURUMU - (Maximum)
AİLENİN İŞ DURUMU - (Maximum)
AİLENİN SAĞLIK DURUMU - (Maximum)

AHP

Geri Çık Başılat

	AİLENİN EKONOMİK DURUMU	BEKLEME SÜRESİ	EBEVEYN KAPASİTE	AİLENİN EĞİTİM DURUMU	AİLENİN İŞ DURUMU	AİLENİN SAĞLIK DURUMU	KONUT FİZİKİ ŞARTLARI	AİLENİN SOSYAL YAŞANTI	AİLENİN DİĞER İNSANLARI İLETİŞİMİ	ÇOCUKLAR YAŞAMA DAİR BEKLENTİ	AİLE BİREYLERİ KİŞİLİK ÖZELLİKLERİ	EVLİLİK YAŞANTI ARASI İLETİŞİM	SOSYAL ÇALIŞMA GENEL KANAATI
AİLENİ...	1												
BEKLE...		1											
EBEVE...			1										
AİLENİ...				1									
AİLENİ...					1								
KONUT...						1							
AİLENİ...							1						
AİLENİ...								1					
ÇOCU...									1				
AİLE B...										1			
EVİLİ...											1		
SOSYA...												1	

Tablo Doldur

HESAPLA

Şekil 4.6. AHP öncelik matrisinin boş hali.

Ölçütlerin birbirlerine göre önem derecelerini "Tablo Doldur" butonu ile ekrana otomatik olarak getirmek de mümkündür.

Şekil 4.6'daki "Tablo Doldur" butonuna tıklanması ile ekrana yansıyan AHP öncelik matrisinin otomatik olarak doldurulmuş hali Şekil 4.7'de verilmiştir.

AHP-TOPSIS Bilgi Sistemi

AİLENİN EKONOMİK DURUMU - (Maximum)
BEKLEME SÜRESİ - (Maximum)
EBEVEYNLIK KAPASİTESİ - (Maximum)
AİLENİN EĞİTİM DURUMU - (Maximum)
AİLENİN İŞ DURUMU - (Maximum)
AİLENİN SAĞLIK DURUMU - (Maximum)

AHP

Geri Çık Başılat

	AİLENİN EKONOMİK DURUMU	BEKLEME SÜRESİ	EBEVEYN KAPASİTE	AİLENİN EĞİTİM DURUMU	AİLENİN İŞ DURUMU	AİLENİN SAĞLIK DURUMU	KONUT FİZİKİ ŞARTLARI	AİLENİN SOSYAL YAŞANTI	AİLENİN DİĞER İNSANLARI İLETİŞİMİ	ÇOCUKLAR YAŞAMA DAİR BEKLENTİ	AİLE BİREYLERİ KİŞİLİK ÖZELLİKLERİ	EVLİLİK YAŞANTI ARASI İLETİŞİM	SOSYAL ÇALIŞMA GENEL KANAATI
AİLENİ...	1	3	0,25	0,333	3	0,333	3	3	3	0,333	0,333	0,333	0,333
BEKLE...	0,333	1	0,2	0,333	0,2	0,25	0,333	0,333	0,333	0,25	0,25	0,25	0,25
EBEVE...	4	5	1	3	4	3	3	5	3	1	1	3	0,333
AİLENİ...	3	3	0,333	1	3	1	3	3	4	0,333	0,333	0,333	0,333
AİLENİ...	0,333	5	0,25	0,333	1	0,333	1	3	3	0,333	0,333	0,333	0,25
AİLENİ...	3	4	0,333	1	3	1	3	4	4	0,5	0,333	0,333	0,25
KONUT...	0,333	3	0,333	0,333	1	0,333	1	3	3	0,333	0,333	0,333	0,333
AİLENİ...	0,333	3	0,2	0,333	0,333	0,25	0,333	1	2	0,333	0,333	0,333	0,25
AİLENİ...	0,333	3	0,333	0,25	0,333	0,25	0,333	0,5	1	0,333	0,2	0,333	0,2
ÇOCU...	3	4	1	3	3	2	3	3	3	1	0,333	0,25	0,25
AİLE B...	3	4	1	3	3	3	3	3	5	3	1	1	0,25
EVİLİ...	3	4	0,333	3	3	3	3	3	3	4	1	1	0,25
SOSYA...	3	4	3	3	4	4	3	4	5	4	4	4	1

Tablo Doldur

HESAPLA

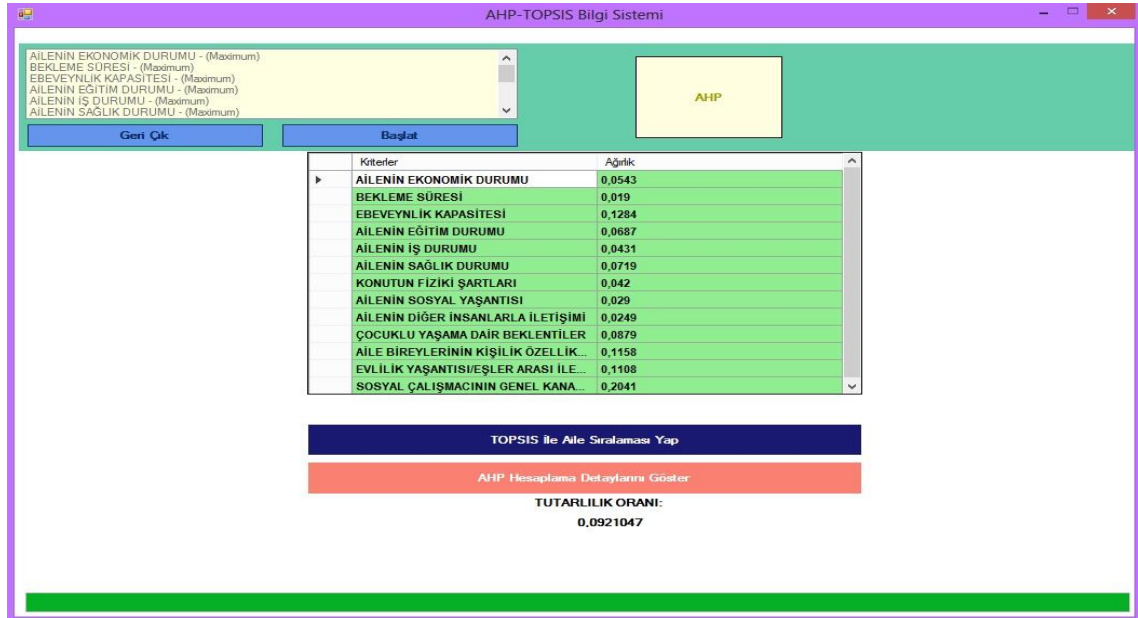
Şekil 4.7. AHP öncelik matrisinin doldurulmuş hali.

AHP öncelik matrisinin program çıktısı Çizelge 4.2'de verilmiştir.

Çizelge 4.2. AHP önem matrisi.

ÖNEM MATRİSİ														
	Ölçütler	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Ailenin ekonomik durumu	1,00	3,00	0,25	0,33	3,00	0,33	3,00	3,00	3,00	0,33	0,33	0,33	0,33
2	Bekleme süresi	0,33	1,00	0,20	0,33	0,20	0,25	0,33	0,33	0,33	0,25	0,25	0,25	0,25
3	Ebeveynlik kapasitesi	4,00	5,00	1,00	3,00	4,00	3,00	3,00	5,00	3,00	1,00	1,00	3,00	0,33
4	Ailenin eğitim durumu	3,00	3,00	0,33	1,00	3,00	1,00	3,00	3,00	4,00	0,33	0,33	0,33	0,33
5	Ailenin iş durumu	0,33	5,00	0,25	0,33	1,00	0,33	1,00	3,00	3,00	0,33	0,33	0,33	0,25
6	Ailenin sağlık durumu	3,00	4,00	0,33	1,00	3,00	1,00	3,00	4,00	4,00	0,50	0,33	0,33	0,25
7	Konutun fiziki şartları	0,33	3,00	0,33	0,33	1,00	0,33	1,00	3,00	3,00	0,33	0,33	0,33	0,33
8	Ailenin sosyal yaşantısı	0,33	3,00	0,20	0,33	0,33	0,25	0,33	1,00	2,00	0,33	0,33	0,33	0,25
9	Ailenin diğer insanlarla iletişimi	0,33	3,00	0,33	0,25	0,33	0,25	0,33	0,50	1,00	0,33	0,20	0,33	0,20
10	Çocuklu yaşama dair beklentiler	3,00	4,00	1,00	3,00	3,00	2,00	3,00	3,00	3,00	1,00	0,33	0,25	0,25
11	Aile bireylerinin kişisel özellikleri	3,00	4,00	1,00	3,00	3,00	3,00	3,00	3,00	5,00	3,00	1,00	1,00	0,25
12	Evlilik yaşantısı, eşler arasındaki iletişim	3,00	4,00	0,33	3,00	3,00	3,00	3,00	3,00	3,00	4,00	1,00	1,00	0,25
13	Sosyal çalışmacının genel kanaati	3,00	4,00	3,00	3,00	4,00	4,00	3,00	4,00	5,00	4,00	4,00	4,00	1,00
TOPLAM		24,67	46,00	8,57	18,92	28,87	18,75	27,00	35,83	39,33	15,75	9,78	11,83	4,28

Şekil 4.7'deki "Hesapla" butonuna basıldığında ölçütlerin ağırlıkları hesaplanmaktadır. Hesaplanan ölçüt ağırlıklarının olduğu ekran Şekil 4.8'de verilmiştir.



Şekil 4.8. Ölçüt ağırlıkları ekranı.

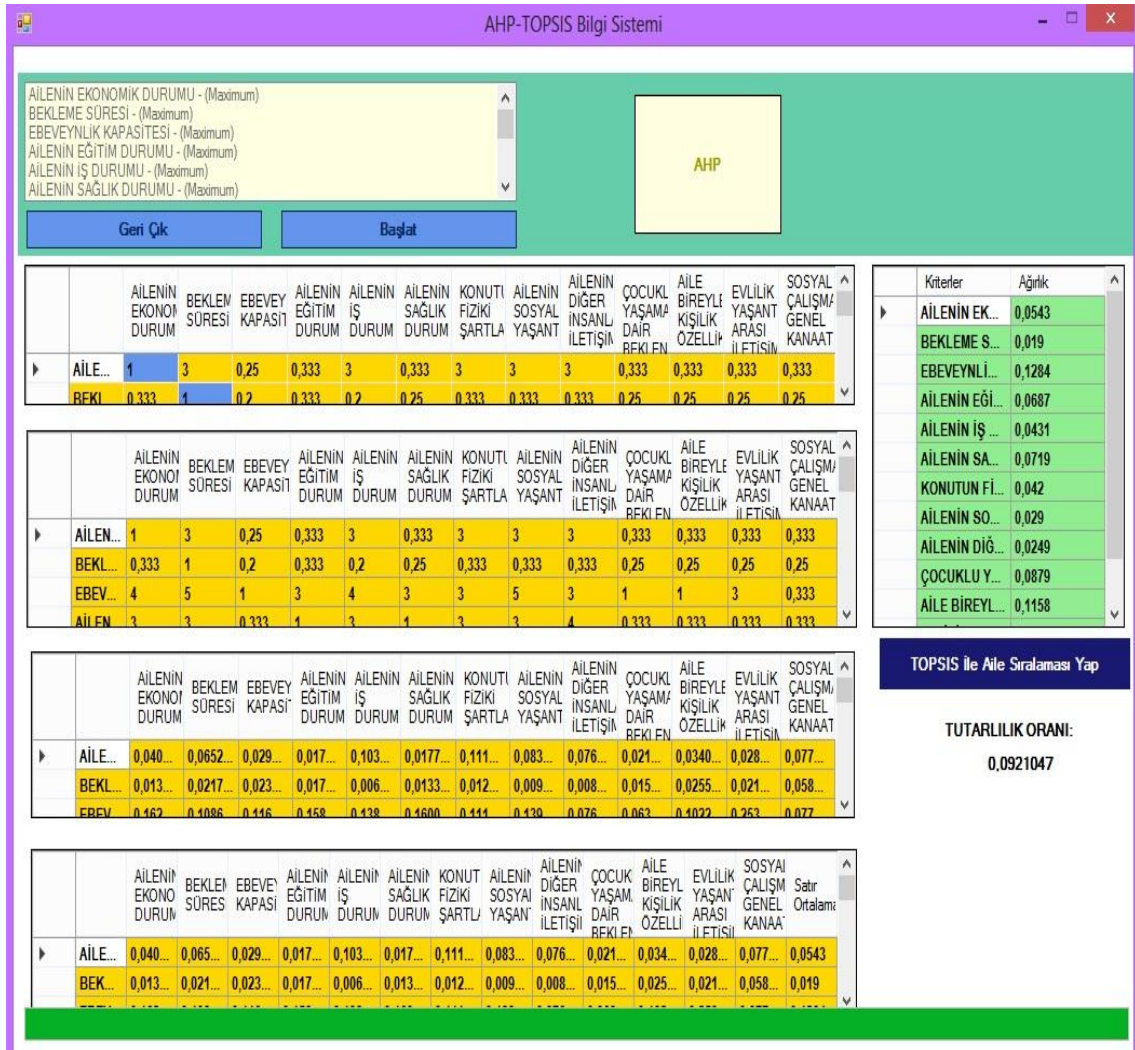
Normalize edilmiş matrisin ve ölçüt ağırlık değerlerinin çizelge hali Çizelge 4.3'de verilmiştir.

Çizelge 4.3. Normalize edilmiş matris.

Normalize Edilmiş Matris														
	1	2	3	4	5	6	7	8	9	10	11	12	13	W
1	0,041	0,065	0,029	0,018	0,104	0,018	0,111	0,084	0,076	0,021	0,034	0,028	0,078	0,0543
2	0,014	0,022	0,023	0,018	0,007	0,013	0,012	0,009	0,008	0,016	0,026	0,021	0,058	0,0190
3	0,162	0,109	0,117	0,159	0,139	0,160	0,111	0,140	0,076	0,064	0,102	0,254	0,078	0,1284
4	0,122	0,065	0,039	0,053	0,104	0,053	0,111	0,084	0,102	0,021	0,034	0,028	0,078	0,0687
5	0,014	0,109	0,029	0,018	0,035	0,018	0,037	0,084	0,076	0,021	0,034	0,028	0,058	0,0431
6	0,122	0,087	0,039	0,053	0,104	0,053	0,111	0,112	0,102	0,032	0,034	0,028	0,058	0,0719
7	0,014	0,065	0,039	0,018	0,035	0,018	0,037	0,084	0,076	0,021	0,034	0,028	0,078	0,0420
8	0,014	0,065	0,023	0,018	0,012	0,013	0,012	0,028	0,051	0,021	0,034	0,028	0,058	0,0290
9	0,014	0,065	0,039	0,013	0,012	0,013	0,012	0,014	0,025	0,021	0,020	0,028	0,047	0,0249
10	0,122	0,087	0,117	0,159	0,104	0,107	0,111	0,084	0,076	0,064	0,034	0,021	0,058	0,0879
11	0,122	0,087	0,117	0,159	0,104	0,160	0,111	0,084	0,127	0,191	0,102	0,085	0,058	0,1158
12	0,122	0,087	0,039	0,159	0,104	0,160	0,111	0,084	0,076	0,254	0,102	0,085	0,058	0,1108
13	0,122	0,087	0,350	0,159	0,139	0,213	0,111	0,112	0,127	0,254	0,409	0,338	0,234	0,2041

Hesaplanan ölçüt ağırlıklarına bakıldığında en önemli ölçütün 0,2041 oranı ile "Sosyal Çalışmacının Genel Kanaati", en az önemli ölçütün ise 0,0190 oranı ile "Bekleme Süresi" ölçütü olduğu tespit edilmiştir. Tutarlılık oranı ise 0,0921047 olarak hesaplanmıştır ve $0,0921047 < 0,10$ olduğu için matris tutarlıdır denilebilmektedir.

Şekil 4.8'deki "AHP Hesaplama Detaylarını Göster" butonuna tıklanıldığında AHP yöntemi kullanılarak hesaplanan ölçüt ağırlıklarının hesaplama aşamaları ekrana yansımaktadır. "AHP Hesaplama Detayları Ekranı" Şekil 4.9'da verilmiştir.



Şekil 4.9. AHP hesaplama detayları ekranı.

Şekil 4.8 veya Şekil 4.9'daki "TOPSIS ile Aile Sıralaması Yap" butonuna tıklanıldığında ekrana gelen "Ebeveyn Adayları Seçim Ekranı" Şekil 4.10'da verilmiştir.



Şekil 4.10. Ebeveyn adayları seçim ekranı.

"Ebeveyn Adayları Seçim Ekranı" nındaki "Tümünü Ekle" butonu ile veritabanında kayıtlı tüm adaylar "Alternatifler" listesine eklenmektedir. Bunun haricinde "Ebeveyn Adayları" listesinden tek tek seçim yapılarak "Ekle" butonu ile "Alternatifler" listesine istenilen adaylar eklenebilmektedir. Daha önce de bahsedildiği gibi kuruma evlat edinmek için başvuran ailelerin evlat edinmek istedikleri küçüklerde aradıkları bazı özellikler olabilmektedir. Örneğin; küçüğün cinsiyeti, bulunma durumu (buluntu veya ailesi tarafından bırakılmış), yaşı vb. ebeveyn adaylarının evlat edinmek isteyecekleri küçükte aradıkları ölçütlerdendir. Kuruma evlat edindirilmeye uygun bir küçük geldiğinde bu gibi özelliklerine bakılarak, ebeveyn adaylarının belirttikleri ölçütler ile evlat edindirilmeye uygun küçüğün özelliklerinin örtüştüğü ebeveynler sıralama işlemine tabi olmalıdır. Örneğin, evlat edindirilecek bir kız çocuğu sistemde kayıtlı ise ebeveyn adayları arasından kız çocuk isteyen adaylar arasında bir sıralama yapılmalıdır. Bu sebeple sıralama işlemi için tek tek ebeveyn eklemeyi sağlayan "Ekle" butonu konulmuştur.

Ebeveyn adayları "Alternatifler" listesine eklendikten sonraki ekran Şekil 4.11'de verilmiştir.



Şekil 4.11. Ebeveyn adayları seçildikten sonraki ara ekran.

Şekil 4.11'deki "TOPSIS Hesaplaması" butonuna basıldığında ebeveyn adaylarına ait ölçüt puanlarının olduğu ekran Şekil 4.12'de verilmiştir.

Şekil 4.12. Ebeveyn adaylarının ölçüt puan ekranı.

Ebeveyn adaylarına ait ölçüt puanları Çizelge 4.4'de verilmiştir.

Çizelge 4.4. Ebeveyn adayları ölçüt puanları.

Ebeveyn Adayları	Ebeveyn Adayları Ölçüt Puanları												
Nalan Nesin-Hasan Nesin	2	3	4	3	3	5	3	3	4	5	4	3	3
Müge Deniz-Nezir Deniz	2	2	2	2	2	2	2	2	2	2	2	2	2
Ceyda Yalçın-Hakan Yalçın	3	5	5	4	4	5	4	3	4	4	4	4	4
Yasemin Güzel-Hamza Güzel	3	5	3	3	4	4	3	3	4	4	4	4	3
Nuriye Candan-Hilmi Candan	3	2	3	3	4	4	3	2	3	4	3	3	3
Nazlı Şeker-Mustafa Şeker	3	3	4	3	4	4	4	4	5	5	5	5	4
Ayşe Aslan-Ahmet-Aslan	4	2	4	4	5	5	4	4	5	5	4	5	4
Gülşüm Bilgeç-Hüseyin Bilgeç	4	4	5	4	5	5	4	5	4	4	5	4	4
Gül Görgülü-Rıdvan Görgülü	4	1	4	4	5	5	5	4	4	4	5	4	4
Selime Ocak-Salih Ocak	4	1	4	3	4	5	4	3	4	4	4	4	4
Asiye Akşehir-Sami Akşehir	5	5	4	3	4	3	5	4	5	5	4	4	4
Cansu Narzan-Taner Narzan	5	4	5	5	5	5	5	5	5	5	5	5	5
Betül Kader-Ramazan Kader	5	1	3	5	5	4	5	4	3	5	4	4	4
Emel Yılmaz-Can Yılmaz	5	3	4	5	5	4	5	5	5	4	5	5	5
Tansu Yosun-Ersin Yosun	5	5	5	5	5	5	5	5	5	5	5	5	5

Şekil 4.12'deki "İlerle" butonuna tıklanıldığında TOPSIS yöntemi ile yapılan hesaplamalara ait "TOPSIS Hesaplama Ekranı" Şekil 4.13'de verilmiştir. Ekranda pozitif ideal ayırım değerleri (S_i^+), negatif ideal ayırım değerleri (S_i^-) ve karar noktasının ideal çözüme göreli yakınlığı (C_i^*) yer almaktadır.

AHP-TOPSIS Bilgi Sistemi

Ebeveyn Adayları

Tümünü Ekle
Ekle
Sil

Kriterler

AİLENİN EKONOMİK DURUMU
 BEKLEME SÜRESİ
 EBEVEYNLIK KAPASİTESİ
 AİLENİN EĞİTİM DURUMU
AİLENİN İŞ DURUMU
 AİLENİN SAĞLIK DURUMU
 KONUTUN FİZİKİ ŞARTLARI

Alternatifler

Ayşe-Aslan-11111111
 Emel-Yılmaz-11111111
 Ceyda-Yalçın-111111
 Nalan-Nesin-111111
 Cansu-Narzan-1111
 Nazlı-Şeker-111111
 Asiye-Akşehir-111111

TOPSIS Hesaplaması

Yeniden Başlat

Detaylı Çözümü Göster

	AİLENİN EKONOMİK DURUMU	BEKLEME SÜRESİ	EBEVEYNLIK KAPASİTESİ	AİLENİN EĞİTİM DURUMU	AİLENİN İŞ DURUMU	AİLENİN SAĞLIK DURUMU	KONUTUN FİZİKİ ŞARTLARI	AİLENİN SOSYAL YAŞANTISI	AİLENİN DİĞER İNŞANLARI İLETİŞİMİ	ÇOCUKLU YAŞAMA DAİR BEKLENTİ	AİLE BİREYLERİ KİŞİLİK ÖZELLİKLE	EVLİLİK YAŞANTISI ARASI İLETİŞİM	SOSYAL ÇALIŞMACI GENEL KANAATI
Ayşe-Asl...	4	2	4	4	5	5	4	4	5	5	4	5	4
Emel-Yıl...	5	3	4	5	5	4	5	5	5	4	5	5	5
Ceyda-Y...	3	5	5	4	4	5	4	3	4	4	4	4	4
Nalan-N...	2	2	4	3	3	5	3	3	4	5	4	3	3

İlerle

Ağırlık	0,0543	0,019	0,1284	0,0687	0,0431	0,0719	0,042	0,029	0,0249	0,0879	0,1158	0,1108	0,2041
---------	--------	-------	--------	--------	--------	--------	-------	-------	--------	--------	--------	--------	--------

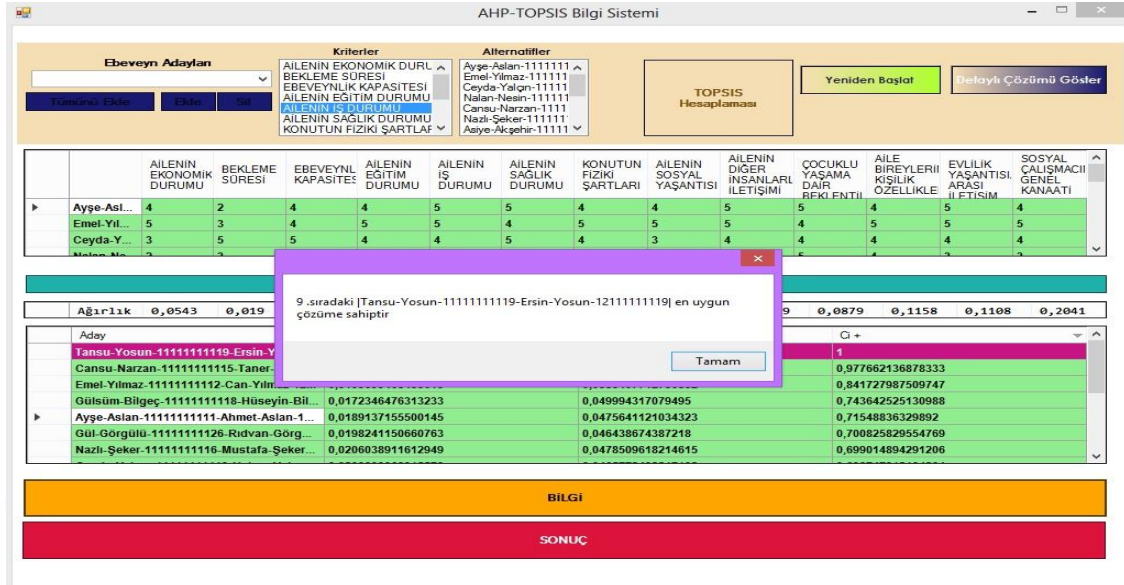
Aday	Si +	Si -	Ci +
Tansu-Yosun-1111111119-Ersin-Yosun...	0	0,0631565242901462	1
Cansu-Narzan-1111111115-Taner-Narza...	0,00144038628279992	0,0630414432885708	0,977662136878333
Emel-Yılmaz-1111111112-Can-Yılmaz-12...	0,0109699468435613	0,0583407712736852	0,841727987509747
Gülüm-Bilgeç-1111111118-Hüseyin-Bil...	0,0172346476313233	0,049994317079495	0,743642525130988
Ayşe-Aslan-1111111111-Ahmet-Aslan-1...	0,0189137155500145	0,0475641121034323	0,71548836329892
Gül-Görgülü-1111111126-Rıdvan-Görg...	0,0198241150660763	0,046438674387218	0,700825829554769
Nazlı-Şeker-1111111116-Mustafa-Şeker...	0,0206038911612949	0,0478509618214615	0,699014894291206

Bilgi

SONUÇ

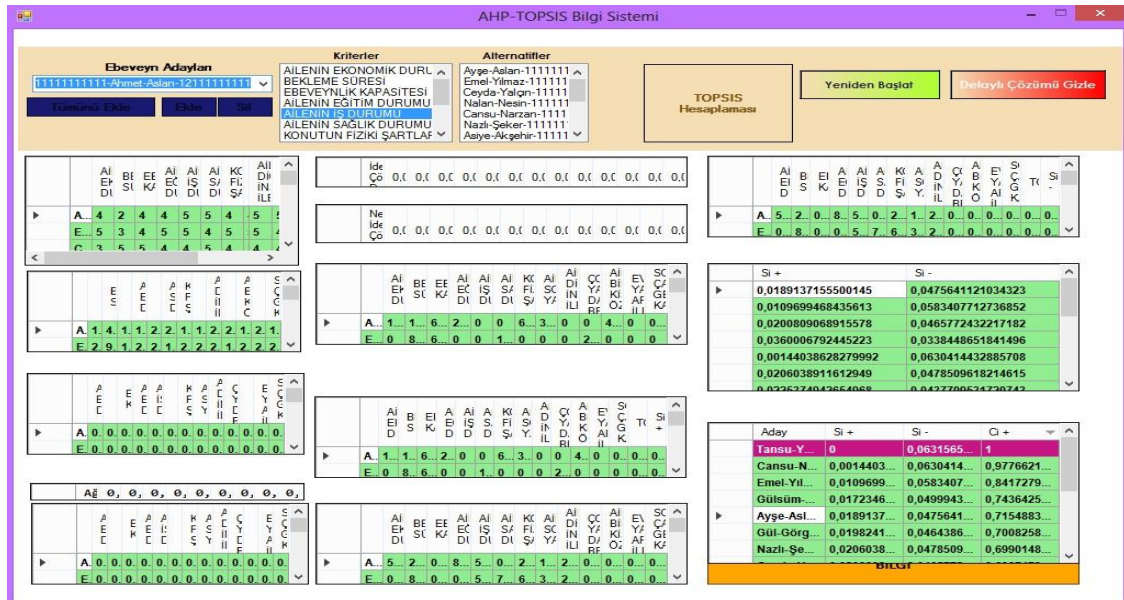
Şekil 4.13. TOPSIS hesaplama sonuç ekranı.

Şekil 4.13'teki "Bilgi" butonuna tıklanıldığında en uygun çözüme ait olan alternatifin hangisi olduğuna dair bilginin verildiği "Hesaplama Sonucundaki En Uygun Çözümü Gösteren Ekran" Şekil 4.14'de verilmiştir.



Şekil 4.14. TOPSIS hesaplama sonucundaki en uygun çözümü gösteren ekran.

Şekil 4.14'deki mesaj ekranını "Tamam" butonu ile kapattıktan sonra yine Şekil 4.14'deki "Detaylı Çözümü Göster" butonuna tıklanıldığında TOPSIS yönteminin hesaplamalarının detaylı olarak gösterildiği "TOPSIS Hesaplaması Detaylı Çözüm Ekranı" Şekil 4.15'te verilmiştir.



Şekil 4.15. TOPSIS hesaplaması detaylı çözüm ekranı.

TOPSIS hesaplaması detaylı çözüm ekranındaki hesaplamalar detaylı olarak Çizelge 4.5'te verilmiştir.

Çizelge 4.5. Başlangıç matrisi.

Ebeveyn Adayları	Başlangıç Matrisi												
Ayşe Aslan Ahmet Aslan	16	4	16	16	25	25	16	16	25	25	16	25	16
Emel Yılmaz Can Yılmaz	25	9	16	25	25	16	25	25	25	16	25	25	25
Ceyda Yalçın Hakan Yalçın	9	25	25	16	16	25	16	9	16	16	16	16	16
Nalan Nesin Hasan Nesin	4	9	16	9	9	25	9	9	16	25	16	9	9
Cansu Narzan Taner Narzan	25	16	25	25	25	25	25	25	25	25	25	25	25
Nazlı Şeker Mustafa Şeker	9	9	16	9	16	16	16	16	25	25	25	25	16
Asiye Akşehir Sami Akşehir	25	25	16	9	16	9	25	16	25	25	16	16	16
Gülsüm Bilgeç Hüseyin Bilgeç	16	16	25	16	25	25	16	25	16	16	25	16	16
Tansu Yosun Ersin Yosun	25	25	25	25	25	25	25	25	25	25	25	25	25
Nuriye Candan Hilmi Candan	9	4	9	9	16	16	9	4	9	16	9	9	9
Selime Ocak Salih Ocak	16	1	16	9	16	25	16	9	16	16	16	16	16
Müge Deniz Nezir Deniz	4	4	4	4	4	4	4	4	4	4	4	4	4
Betül Kader Ramazan Kader	25	1	9	25	25	16	25	16	9	25	16	16	16
Yasemin Güzel Hamza Güzel	9	25	9	9	16	16	9	9	16	16	16	16	9
Gül Görgülü Rıdvan Görgülü	16	1	16	16	25	25	25	16	16	16	25	16	6
Karekök Toplam	15,26	13,19	15,59	14,9	16,85	17,12	16,16	14,97	16,37	17,06	16,58	16,09	15,3

Çizelge 4.6. Normalize edilmiş matris.

Ebeveyn Adayları	Normalize Edilmiş Matris												
Ayşe Aslan Ahmet Aslan	0,26	0,15	0,26	0,27	0,30	0,29	0,25	0,27	0,31	0,29	0,24	0,31	0,26
Emel Yılmaz Can Yılmaz	0,33	0,23	0,26	0,34	0,30	0,23	0,31	0,33	0,31	0,23	0,30	0,31	0,33
Ceyda Yalçın Hakan Yalçın	0,20	0,38	0,32	0,27	0,24	0,29	0,25	0,20	0,24	0,23	0,24	0,25	0,26
Nalan Nesin Hasan Nesin	0,13	0,23	0,26	0,20	0,18	0,29	0,19	0,20	0,24	0,29	0,24	0,19	0,20
Cansu Narzan Taner Narzan	0,33	0,30	0,32	0,34	0,30	0,29	0,31	0,33	0,31	0,29	0,30	0,31	0,33
Nazlı Şeker Mustafa Şeker	0,20	0,23	0,26	0,20	0,24	0,23	0,25	0,27	0,31	0,29	0,30	0,31	0,26
Asiye Akşehir Sami Akşehir	0,33	0,38	0,26	0,20	0,24	0,18	0,31	0,27	0,31	0,29	0,24	0,25	0,26
Gülsüm Bilgeç Hüseyin Bilgeç	0,26	0,30	0,32	0,27	0,30	0,29	0,25	0,33	0,24	0,23	0,30	0,25	0,26
Tansu Yosun Ersin Yosun	0,33	0,38	0,32	0,34	0,30	0,29	0,31	0,33	0,31	0,29	0,30	0,31	0,33
Nuriye Candan Hilmi Candan	0,20	0,15	0,19	0,20	0,24	0,23	0,19	0,13	0,18	0,23	0,18	0,19	0,20
Selime Ocak Salih Ocak	0,26	0,08	0,26	0,20	0,24	0,29	0,25	0,20	0,24	0,23	0,24	0,25	0,26
Müge Deniz Nezir Deniz	0,13	0,15	0,13	0,13	0,12	0,12	0,12	0,13	0,12	0,12	0,12	0,12	0,13
Betül Kader Ramazan Kader	0,33	0,08	0,19	0,34	0,30	0,23	0,31	0,27	0,18	0,29	0,24	0,25	0,26
Yasemin Güzel Hamza Güzel	0,20	0,38	0,19	0,20	0,24	0,23	0,19	0,20	0,24	0,23	0,24	0,25	0,20
Gül Görgülü Rıdvan Görgülü	0,26	0,08	0,26	0,27	0,30	0,29	0,31	0,27	0,24	0,23	0,30	0,25	0,26

Çizelge 4.7. Ağırlıklı standart karar matrisi.

Ebeveyn Adayları	Ağırlıklı Standart Karar Matrisi												
Ayşe Aslan Ahmet Aslan	0,014	0,003	0,033	0,018	0,013	0,021	0,010	0,008	0,008	0,026	0,028	0,034	0,053
Emel Yılmaz Can Yılmaz	0,018	0,004	0,033	0,023	0,013	0,017	0,013	0,010	0,008	0,021	0,035	0,034	0,067
Ceyda Yalçın Hakan Yalçın	0,011	0,007	0,041	0,018	0,010	0,021	0,010	0,006	0,006	0,021	0,028	0,028	0,053
Nalan Nesin Hasan Nesin	0,007	0,004	0,033	0,014	0,008	0,021	0,008	0,006	0,006	0,026	0,028	0,021	0,040
Cansu Narzan Taner Narzan	0,018	0,006	0,041	0,023	0,013	0,021	0,013	0,010	0,008	0,026	0,035	0,034	0,067
Nazlı Şeker Mustafa Şeker	0,011	0,004	0,033	0,014	0,010	0,017	0,010	0,008	0,008	0,026	0,035	0,034	0,053
Asiye Akşehir Sami Akşehir	0,018	0,007	0,033	0,014	0,010	0,013	0,013	0,008	0,008	0,026	0,028	0,028	0,053
Gülsüm Bilgeç Hüseyin Bilgeç	0,014	0,006	0,041	0,018	0,013	0,021	0,010	0,010	0,006	0,021	0,035	0,028	0,053
Tansu Yosun Ersin Yosun	0,018	0,007	0,041	0,023	0,013	0,021	0,013	0,010	0,008	0,026	0,035	0,034	0,067
Nuriye Candan Hilmi Candan	0,011	0,003	0,025	0,014	0,010	0,017	0,008	0,004	0,005	0,021	0,021	0,021	0,040
Selime Ocak Salih Ocak	0,014	0,001	0,033	0,014	0,010	0,021	0,010	0,006	0,006	0,021	0,028	0,028	0,053
Müge Deniz Nezir Deniz	0,007	0,003	0,016	0,009	0,005	0,008	0,005	0,004	0,003	0,010	0,014	0,014	0,027
Betül Kader Ramazan Kader	0,018	0,001	0,025	0,023	0,013	0,017	0,013	0,008	0,005	0,026	0,028	0,028	0,053
Yasemin Güzel Hamza Güzel	0,011	0,007	0,025	0,014	0,010	0,017	0,008	0,006	0,006	0,021	0,028	0,028	0,040
Gül Görgülü Rıdvan Görgülü	0,014	0,001	0,033	0,018	0,013	0,021	0,013	0,008	0,006	0,021	0,035	0,028	0,053

Çizelge 4.8. Pozitif ve negatif ideal çözüm değerleri.

Ölçütler	Pozitif İdeal Çözüm Değerleri	Negatif İdeal Çözüm Değerleri
Ailenin ekonomik durumu	0,0018	0,0071
Bekleme süresi	0,0072	0,0014
Ebeveynlik kapasitesi	0,0412	0,0165
Ailenin eğitim durumu	0,0231	0,0092
Ailenin iş durumu	0,0128	0,0052
Ailenin sağlık durumu	0,0210	0,0084
Konutun fiziki şartları	0,0130	0,0052
Ailenin sosyal yaşantısı	0,0097	0,0039
Ailenin diğer insanlarla iletişimi	0,0076	0,0030
Çocuklu yaşama dair beklentiler	0,0258	0,0103
Aile bireylerinin kişisel özellikleri	0,0349	0,0140
Evlilik yaşantısı, eşler arasındaki iletişim	0,0344	0,0138
Sosyal çalışmacının genel kanaati	0,0667	0,0267

Çizelge 4.9. TOPSIS hesaplaması sonucu elde edilen ebeveyn aday sıralaması.

Sıralama	Ebeveyn Adayları	Si+	Si-	Ci+
1	Tansu Yosun-Ersin Yosun	0	0,063157	1
2	Cansu Narzan-Taner Narzan	0,00144	0,063041	0,977662
3	Emel Yılmaz-Can Yılmaz	0,01097	0,058341	0,841728
4	Gülsüm Bilgeç-Hüseyin Bilgeç	0,017235	0,049994	0,743643
5	Ayşe Aslan-Ahmet Aslan	0,018914	0,047564	0,715488
6	Gül Görgülü-Rıdvan Görgülü	0,019824	0,046439	0,700826
7	Nazlı Şeker-Mustafa Şeker	0,020604	0,047851	0,699015
8	Ceyda Yalçın-Hakan Yalçın	0,020081	0,046577	0,698748
9	Asiye Akşehir-Sami Akşehir	0,022537	0,043771	0,660111
10	Selime Ocak-Salih Ocak	0,023024	0,042086	0,646382
11	Betül Kader-Ramazan Kader	0,024686	0,043776	0,639419
12	Nalan Nesin-Hasan Nesin	0,036001	0,033845	0,484567
13	Yasemin Güzel-Hamza Güzel	0,036199	0,030337	0,455947
14	Nuriye Candan-Hilmi Candan	0,040537	0,024291	0,374695
15	Müge Deniz-Nezir Deniz	0,063041	0,00144	0,022338

Bilgi sisteminden elde edilen hesaplama sonuçlarına bakıldığında evlat edinme sıralamasının ilk sırasında Tansu Yosun-Ersin Yosun çifti yer almaktadır. Sıralamanın sonunda ise Müge Deniz-Nezir Deniz çifti bulunmaktadır.

Şekil 4.15'teki "Detaylı Çözümü Gizle" butonuna tıkladıktan sonra açılan Şekil 4.14'deki "Sonuç" butonuna tıklanıldığında elde edilen ebeveyn adaylarının sıralandığı "TOPSIS Sonucu Alternatif Sıralama Ekranları" Şekil 4.16 ve Şekil 4.17'de verilmiştir.



Alternatifler	Sıralamaları
Tansu-Yosun-...	1
Cansu-Narzan-...	2
Emel-Yılmaz-11...	3
Gülsüm-Bilgeç...	4
Ayşe-Aslan-11...	5
Gül-Görgülü-1...	6
Nazlı-Şeker-11...	7
Ceyda-Yalçın-...	8
Asiye-Akşehir-...	9

Şekil 4.16. TOPSIS sonucu alternatif sıralama ekranı-1.



Alternatifler	Sıralamaları
Nazlı-Şeker-11...	7
Ceyda-Yalçın-...	8
Asiye-Akşehir-...	9
Selime-Ocak-1...	10
Betül-Kader-1...	11
Nalan-Nesin-1...	12
Yasemin-Güzel...	13
Nuriye-Canda...	14
Müge-Deniz-11...	15

Şekil 4.17. TOPSIS sonucu alternatif sıralama ekranı-2.

Hazırlanan bilgi sisteminde AHP yöntemi ile hesaplanan ölçüt ağırlıklarına bakıldığında en önemli ölçütün 0,2041 oranı ile "Sosyal Çalışmacının Genel Kanaati" ölçütü, en az öneme sahip ölçütün ise 0,0190 oranı ile "Bekleme Süresi" ölçütü olduğu tespit edilmiştir. Tutarlılık oranı 0,0921047 olarak hesaplanmıştır ve önem matrisinin tutarlı olduğu tespit edilmiştir. Evlat edinmek için başvuran ebeveyn adayları arasında yapılan evlat edinme sıralaması sonucunda, ilk sırada yer alan ebeveyn adayları TOPSIS yöntemi kullanılarak tespit edilmiştir.

Ebeveyn adaylarının sahip oldukları ölçüt puanları ve ölçütlerin önem dereceleri dikkate alınarak oluşturulan ebeveyn aday sıralaması, mevcut sıralama sistemine göre daha adil bir sonuç vermektedir. Mevcut sıralama sisteminde sadece evrak teslim tarihi ölçütü baz alınırken önerilen bilgi sistemi ile 13 ayrı ölçüt dikkate alınarak bir sıralama listesi oluşturulma ve en uygun aile seçimi yapılmaktadır. Oluşturulan sıralama listesinin ilk sırasındaki ebeveyn adayı/adayları ile görüşmeler sağlanıp uygun bulunması durumunda evlat edinme sürecinin aile tespit aşaması tamamlanmış olacaktır. Şartlarda uygunsuzluk olması durumunda ise sıradaki ebeveyn adayı/adayları ile görüşmeler yapılarak sürece devam edilecektir. Bu sayede oluşturulan bilgi sistemi ile en uygun adayın tespit edilmesi işlemi tamamlanacaktır.

Bilgi sistemine kaydedilen ebeveyn sayılarında veya çocuk sayılarında bir sınırlama bulunmamaktadır. Bilgi sisteminin işlem süresi de oldukça kısadır.

5. SONUÇ VE ÖNERİLER

Evlat edinme, farklı uygulamalarla yüzyıllardır var olan bir süreçtir. Toplumsal yapının temelini oluşturan bireylerin gelişim sürecini en çok etkileyen ve kişilik yapısının oluşmasını sağlayan ortam doğup büyüdüğü ailedir. Ailesi olmayan küçükler için de bu ortam devlet tarafından sağlanmaktadır. Ailesi tarafından bırakılmak zorunda kalmış bir çok çocuk devlet koruması altında hayatlarına devam etmektedir. Bunun yanı sıra çeşitli sebeplerle evlat sahibi olamayan bireyler/aileler de bulunmaktadır. Evlat edinme kavramı, bahsi geçen iki tarafın da eksikliklerinin giderilmesi ve toplumsal hayat için her anlamda daha yeterli bireyler yetiştirilmesi için önemli bir kavramdır. Evlat edinme süreci bireylerin hayatlarını doğrudan etkileyen bir süreç olduğu için bu sürece hassasiyetle yaklaşılmaktadır. Fakat evlat edinmek için kuruma başvuran aileler/bireyler sadece evrak teslim tarihine bakılarak sıraya alınmaktadır.

Bu çalışmada, evlat edinme sürecini daha sistematik ve adil olarak yönetebilmek adına AHP ve TOPSIS yöntemlerinin bütünleşik olarak kullanıldığı bir bilgi sistemi önerilmiştir. Evlat edinmek için kuruma başvuran ebeveyn adaylarının durumlarını değerlendirerek sıralama yapan ve en uygun ailenin seçilmesine imkan sağlayan bilgi sisteminin çıktıları değerlendirilmiştir. Sıralama sonucunda tespit edilen en uygun aile/birey ile karşılıklı görüşmeler yapıp evlat edindirilme sürecinin tamamlanması önerilmiştir.

Evlat edinme süreci hassas bir süreç olduğu ve bu süreçteki bir çok bilgi gizli olduğu için sadece bir sosyal çalışmacı ile görüşme sağlanabilmektedir. Ayrıca söz konusu gizliliklerden dolayı veritabanına kaydedilen tüm ebeveyn adaylarının ve çocukların bilgileri hayali olarak üretilen bilgiler olarak kaydedilmiştir.

Gelecekteki çalışmalarda daha fazla sosyal çalışmacı ile görüşmeler yapılarak ölçütler tekrar gözden geçirilebilir. Sosyal çalışmacılarla anketler düzenlenip ölçütlerin önem dereceleri anket sonuçlarına göre belirlenebilir. Aile ölçütleri ile evlat edindirilecek küçüğün mevcut durumu karşılaştırılarak ebeveyn adayı sıralama işlemi öncesinde bir filtreleme işlemi yapılabilir. Aile ekleme ve çocuk ekleme sekmeleri geliştirilerek uygun aile ile uygun çocuğun eşleşmesi işlemi oluşturulan bilgi sistemi ile yapılabilir. Uygulama aşamasında Bursa ilinde evlat edinmek için kuruma başvuran ebeveyn adaylarının başvuruları sıralanmıştır. Bilgi sisteminin veritabanı geliştirilerek Türkiye genelinde sistemin kullanımı sağlanabilir. Oluşturulan bilgi sistemine farklı ÇÖKV yöntemleri eklenerek sıralama yapılabilir. Evlat edinme süreci çok daha hassas ve uzun süreli bir süreçtir. Evlat edinme kavramına benzer bir kavram olan ve daha kısa bir zamanda sonuçlanan, ülkemizde gün geçtikçe yaygınlaşan

"Koruyucu Aile Kavramı" da hazırlanan bilgi sistemine dahil edilebilir. Koruyucu aile olmak için başvuran ailelerin de başvuru sıralaması ve en uygun ailenin seçimi, oluşturulan bilgi sistemi ile yapılabilir.



KAYNAKLAR DİZİNİ

- Akçalı, E., Aydın, Ö. ve Öznehir, S. (2009). Ankara için optimal hastane yeri seçiminin analitik hiyerarşi süreci ile modellenmesi. *Süleyman Demirel Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi*, 14(2), 69-86.
- Akçay, A., Al, S. ve Sarıözkan, S. (2018). Akademik personelin et tüketim tercihlerinin analitik hiyerarşi prosesi ile değerlendirilmesi. *Veteriner Hekim Dergisi*, 89(1), 11-24.
- Akman, G., Başlı, H., Gündüz, E. B. ve Özcan, B. (2018). Çok kriterli karar vermede AHP ve TOPSIS yöntemleriyle uçuş noktası seçimi. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 34(3), 1-13.
- Akyer, H. ve Şahin, Y. (2011). Ülke kaynaklarının verimli kullanımı: 4x4 arama ve kurtarma aracı seçiminde AHP ve TOPSIS yöntemlerinin uygulaması. *Süleyman Demirel Üniversitesi Vizyoner Dergisi*, 3(5), 72-87.
- Akyıldız E. (2006). Analitik hiyerarşi süreci ve bankacılık sektöründe bir uygulama. Yüksek Lisans Tezi, Gazi Üniversitesi Sosyal Bilimler Enstitüsü, Ankara, 96 s.
- Akyüz, K., Bayram, B., Ersen, N. Üçüncü, T. ve Üçüncü, K. (2017). TOPSIS yöntemiyle mobilya endüstrisinde kuruluş yeri seçimi. *İleri Teknoloji Bilimleri Dergisi*, 6, 435-443.
- Ali, E. ve Karagül, M. (2008) , 1986-2006 Türkiye ekonomisinin performans değerlendirmesi. *Celal Bayar Üniversitesi İ.İ.B.F. Yönetim ve Ekonomi Dergisi*, 15(1), 1-14.
- Alizmini, H. R., Moghadam, M. K., Nooramin, A. S. ve Sayareh, J. (2012). TOPSIS ve AHP techniques for selecting the most efficient marine container yard gantry Crane. *Opsearch*, 49(2), 116-132.
- Alkan A. (2006). AHP’de Dilsel Karşılaştırma Sürecinin Bulanık Mantıkla Gerçekleştirilmesi, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, 113 s.
- Alsü, E., Kallo, Z. ve Taşdemir, A., (2018). Katılım bankalarının performanslarının değerlendirilmesi: TOPSIS yöntemi ile uluslararası karşılaştırma. *Gaziantep University Journal of Social Sciences*, 17(1), 303 – 316.
- Altınoğlu-Dikmeer, İ., Erol, N., ve Gençöz, T. (2014). Evlat edinilmiş ve biyolojik ebeveyniyle yaşayan çocukların davranışsal ve duygusal sorunları ve bağlanma düzeyleri ile anne babalarının çocuk yetiştirme stillerinin karşılaştırmalı olarak değerlendirilmesi. *Türk Psikiyatri Dergisi*, 25(4), 234- 243.
- Arıkan V. S., Erdoğan Ş. ve Öztürk A. (2011). Analitik hiyerarşi süreci kullanılarak tedarikçilerin değerlendirilmesi bir tekstil firmasında uygulama. *Dokuz Eylül Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 26(1), 93-112.
- Arsal, M. S. (1947). *Türk Tarihi ve Hukuk*. İstanbul: Türk Tarih Kurumu Yayınları, 344.
- Ataay, A. M. (1957), *Medeni Hukukda Evlat Edinme*. İstanbul: İsmail Akgün Matbaası, 16-17.

KAYNAKLAR DİZİNİ(devam)

- Ataei, M., Azadeh, A. ve Osanloo, M. (2010). A new approach to mining method selection based on modifying the nicholas technique. *Applied Soft Computing*, 10 (4), 1040-1061.
- Atağan, G. ve Yükçü, S. (2010). TOPSIS yöntemine göre performans değerlendirme. *Muhasebe ve Finansman Dergisi*, 45, 28-35.
- Ayan, E., Cihan, Ş., Eren, T., Topal, T. ve Yıldırım, E. K. (2017). Çok ölçütlü karar verme yöntemleri ile ekokardiografi cihazı seçiminin yapılması. *HSP*, 4(1), 41-49.
- Ayan, T.Y. ve Pabuçcu, H. (2013). Yenilenebilir enerji kaynakları yatırım projelerinin analitik hiyerarşi süreci yöntemi ile değerlendirilmesi. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 18(3), 89-110.
- Aydemir, G., Şenel, B. ve Şenel, M. (2017). Çok kriterli karar verme metodu TOPSIS ile personel seçimi. *Uluslararası Hakemli Ekonomi Yönetimi Araştırmaları Dergisi*, 13, 19-70.
- Aydoğdu, M. (2006). *Çağdaş Hukuki Gelişmeler Işığında Evlat Edinme*, İzmir: Günaydın Hukuk Yayınları, 4.
- Bayer, A. (2013). Değişen toplumsal yapıda aile. *Şırnak Üniversitesi İlahiyat Fakültesi Dergisi*, 4(8), 101-129.
- Benedict, P. ve Güriz A. (1974). *Türk Hukuku Ve Toplum Üzerine İncelemeler*, Ankara: Türkiye Kalkınma Vakfı Yayınları, 163-223.
- Bhasin, V., Suresh, K., Tadisina, M. ve Troutt, D., (1991). Selecting a doctoral programme using the analytic hierarchy process-the importance of perspective. *The Journal of the Operational Research Society*, 42(8), 631-640.
- Bilmen, Ö. N. (1953). *Hukuki İslamiyye ve İstilahat-ı Fıkhiyye Kamusu*, C(3), İstanbul: Ravza Yayınları, 431.
- Boran, F. (2009). Personel Seçimi Probleminde Sezgisel Bulanık Küme Uygulaması, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 87s.
- Bozkurt, Ö. Ç., Pençe, İ. ve Tarhan, L. (2017). Türk eğitim vakfı bursu verilecek uygun adayların AHP ve TOPSIS yöntemi kullanılarak belirlenmesi. *Burdur Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Uygulamalı Bilimler Dergisi*, 1(1), 37-49.
- Changb, T. Y., Tsaura, S. H., Yena, C. H., (2002). The evaluation of airline service quality by fuzzy MCDM. *Tourism Management*, 23, 107-115.
- Chen, C. T. (2000). Extensions of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems*, 114(1), 1-9.
- Chang, T. H. ve Wang, T. C. (2007). Application of TOPSIS in evaluating initial training aircraft under a fuzzy environment. *Expert Systems with Applications*, 33(4), 870-880.

KAYNAKLAR DİZİNİ(devam)

- Chin, K.S., Chiu, S. ve Tummala, V.M.R. (1999). An evaluation of success factors using the AHP to implement ISO 14001-based EMS. *The International Journal of Quality and Reliability Management*, 16(4), 341-361.
- Çınar, Y. (2004). Çok Nitelikli Karar Verme ve ‘Bankaların Mali Performanslarının Değerlendirilmesi, Yüksek Lisans Tezi, Ankara Üniversitesi Sosyal Bilimler Enstitüsü, Ankara, 133.
- Dağdeviren, M., Dizdar, E.N., Eraslan, E. ve Kurt, M., (2005). Tedarikçi seçimi problemine analitik ağ süreci ile alternatif bir yaklaşım. *Teknoloji*, 8(2), 115-122.
- Dağdeviren, M., Yavuz, S. ve Kılınç, N. (2009), Weapon selection using the AHP and TOPSIS methods under fuzzy environment, *Expert Systems with Applications*, 36, 8143-8151.
- Datta, S., Mahapatra S. S. ve Samantra, C., (2017). A risk-based decision support framework for selection of appropriate safety measure system for underground coal mines. *Int. J. of Injury Control and Safety Promotion*, 24 (1), 54-68.
- Davim, J.P. ve Rao R.V. (2006). A decision-making framework model for material selection using a combined multiple attribute decision-making method. *The International Journal Of Advanced Manufacturing Technology*, 35(7-8), 751-760.
- Demirci, A. E. (2018), An Integrated Approach Based on Anp, Dematel ve Electre Methodologies for Personnel Selection and an Application in Industry,. Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, 160.
- Deng, H., Willis, R.J. ve Yeh, C.H. (2000). Inter-company comparison using modified TOPSIS with objective weights. *Computers & Operations Research*, 27(10), 963-973.
- Doğan, A. ve Önder E. (2014). İnsan kaynakları temin ve seçiminde çok kriterli karar verme tekniklerinin kullanılması ve bir uygulama. *Journal of Yasar University*, 9(34), 5796-5819.
- Dubish S., Solomon A., Wishart N. ve Zanakis S. H., (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, 107(3), 507-529.
- Dumanoğlu S. ve Ergül N. (2010). İMKB’de işlem gören teknoloji şirketlerinin mali performans ölçümü, *Mufad Journal*, 48, 101-111.
- Dumaret, A.C. ve Rosset, D.J. (2005). Adoption and child welfare protection in France. *Early Child Development and Care*, 175, 661-670.
- Dursun, M. ve Karsak, E. (2010). A fuzzy MCDM approach for personel selection. *Expert Systems with Applications*, 37, 4324–4330.

KAYNAKLAR DİZİNİ(devam)

- Dündar, S. ve Ecer, F. (2008). Öğrencilerin GSM operatörü tercihinin analitik hiyerarşi süreci yöntemiyle belirlenmesi. *Celal Bayar Üniversitesi İ.İ.B.F. Yönetim ve Ekonomi Dergisi*, 15(1), 195-205.
- Efe, E., Küçükönder, H. ve Üçkardeş, F. (2013). Çok ölçütlü karar verme yaklaşımlarından analitik hiyerarşi sürecinin hayvancılıkta kullanımı. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 3(3), 91-98.
- Eren, T., Geyik, H., Hamurcu, M., Tosun, M. ve Ünlüsoy, S. (2016), Kitap basımevi seçiminde AHP ve TOPSIS yöntemlerinin kullanımı. *International Journal of Social and Educational Sciences/Uluslararası Sosyal ve Eğitim Bilimleri Dergisi*, 3 (6), 106-126.
- Ersöz, F., ve Kabak, M. (2010), Savunma sanayi uygulamalarında çok kriterli karar verme yöntemlerinin literatür araştırması”, *Savunma Bilimleri Dergisi*, 9 (1), 97-125.
- Fazlollahtabar H. (2010). A subjective framework for seat comfort based on a heuristic multicriteria decision making technique and anthropometry. *Applied Ergonomics*, 42(1), 16-28.
- Gergin, R. E. (2014). Türkiye’deki Bölgelerin Lojistik Performanslarının Bütünleştirilmiş AHP ve TOPSIS Yöntemiyle Değerlendirilmesi, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü, Trabzon, 129.
- Giray, F. K. (2001). Uluslararası sözleşmelerde evlat edinme. *Milletlerarası Hukuk ve Milletlerarası Özel Hukuk Bülteni*, 1(2), 333-350.
- Gois, J. D. R., Longaray, A. A. ve Munhoz, P. R. S. (2015). Proposal for using AHP method to evaluate the quality of services provided by out sourced companies. *Procedia Computer Science*, 55, 715-724.
- Gökce, H. (2017). Bir sosyal politika olarak evlât edindirmenin toplumsal direnç kaynakları. *Uluslararası Beşeri ve Sosyal Bilimler İnceleme Dergisi*, 1, 19-38.
- Güngör, İ. ve İşler, B.İ. (2005). Analitik hiyerarşi yaklaşımı ile otomobil seçimi. *Zonguldak Karaelmas Üniversitesi Sosyal Bilimler Dergisi*, 1(2), 21-33.
- Haliloğlu, H. (2017). Kur’ân-ı Kerîm’de tebennî ile ilgili âyetlereebü’l-berekâtenneşefî ve adudüddîn el-îcî’nin yaklaşımlarının mukâyesesi. *Namık Kemal Üniversitesi İlahiyat Fakültesi Dergisi*, 3(2), 271-308.
- Handfield, R., Melnyk, S.A., Sroufe, R. ve Walton, S.V. (2002). Applying environmental criteria to supplier assessment: a study in the application of the analytical hierarchy process. *European Journal of Operational Research*, 141, 70-87.
- Hassani, F. (2017). İnsan kaynakları yönetiminde personel seçiminin önemi ve TOPSIS yöntemiyle bir uygulama. *Sosyal Bilimler Dergisi/The Journal of Social Science*. 4(16), 486-503.

KAYNAKLAR DİZİNİ(devam)

- Herz, İ. (2018). Mobilya seçiminde etkili olan kalite faktörlerinin AHP ve TOPSIS yöntemleri ile değerlendirilmesi. Karadeniz Teknik Üniversitesi Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, Trabzon, 178.
- Hsuan, T. P., Tsai, L. C. ve Wu, C.R. (2008). Financial service of wealth management banking: balanced score card approach. *Journal of Social Sciences*, 4 (4), 255-263.
- Hunjak, T. ve Jakovčević, D. (2001). AHP based model for bank performance evaluation and rating. *ISAHP 2001 Proceedings*, 149-157.
- Ilgaz, A. (2018). Lojistik sektöründe personel seçim kriterlerinin AHP ve TOPSIS yöntemleri ile değerlendirilmesi. *Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 3(32), 586-605.
- İç, Y. T. ve Yurdakul, M. (2003). Türk otomotiv firmalarının performans ölçümü ve analizine yönelik TOPSIS yöntemini kullanan bir örnek çalışma. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 18 (1), 1-18.
- İpek, Ç. (2018). Konut Satın Alma Probleminin AHP Temelli Gri İlişkisel Analiz Yöntemi İle Değerlendirilmesi, Yüksek Lisans Tezi, Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü, Burdur, 74.
- İpekçi Çetin, E. ve Tarcan İçigen, E. (2017). AHP temelli TOPSIS yöntemi ile konaklama işletmelerinde personel seçimi. *Balkan Journal of Social Sciences/Balkan Sosyal Bilimler Dergisi*, 7(13), 179-187.
- Kantemir, Ö. ve Özkil, A. (2015). Teröristle mücadelede kullanılan silah sistemlerinin etkinliklerine yönelik algıların belirlenmesi. *Güvenlik Bilimleri Dergisi*, 4(2), 37-62.
- Karataş, T., Ömürbek, N., Özcan, A., Tunca, M. Z. ve Yıldız, E. (2016). AHP TOPSIS yönteminin tıpta uzmanlık alan seçiminde kullanımı. *Akdeniz İ.İ.B.F. Dergisi*, 33, 201-219.
- Kaya, C. (2009). *Türk Hukukunda Evlat Edinme*, İstanbul: Beta Basım Dağıtım, 4.
- Kety S. S.,Lunde I., Ortmann J., Rosenthal D., Schulsinger F. ve Wender P. H. (1986). Psychiatric disorders in the biological and adoptive families of adopted individuals with affective disorders. *Arch Gen Psychiatry*, 43, 923-929.
- Korkmaz, T., Uygurtürk, H. (2012). Finansal performansın TOPSIS çok kriterli karar verme yöntemi ile belirlenmesi ana metal sanayi işletmeleri üzerine bir uygulama. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 7(2), 95-115.
- Kurşunoğlu, N. ve Önder, M. (2014). Yeraltı maden işletmelerinde analitik hiyerarşi prosesi yöntemi ile uygun vantilatör seçimi. Türkiye 19. Kömür Kongresi, Zonguldak, 115-122.

KAYNAKLAR DİZİNİ(devam)

- Li, H. ve Qing-sheng, X. (2006). Application of TOPSIS in the bidding evaluation of manufacturing enterprises. 5th International Conference on e-Engineering&Digital Enterprise Technology, China, 184-188.
- Lin, C. W., Opricovic, S. ve Tzeng G. H. (2005). Multi-criteria analysis of alternative-fuel buses for public transportation. *Energy Policy*, 33(11), 1373-1383.
- Liu, S., Wang, S., Yang, S., Yue, H. ve Zhang, R. (2004). Regional disaggregation of energy consumption target: the case of Henan province. *Energy Procedia*, 104, 104-109.
- Mahmoodzadeh, S., Pariazar, M., Shahrabi, J. ve Zaeri, M.S. (2007). Project selection by using fuzzy AHP and TOPSIS technique. *World Academy of Science, Engineering and Technology*, 30, 333-338.
- Marbini A.H. ve Tavana M. (2011). A group AHP-TOPSIS framework for human spaceflight mission planning at NASA. *Expert Systems with Applications*, 38(11), 13588-13603.
- Opricovic, S. ve Tzeng, G.H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operation Research*, 155, 445-455.
- Öncel, M. ve Vatansever, K. (2014). An implementation of integrated multi-criteria decision making techniques for academic staff recruitment. *Journal of Management. Marketing and Logistics*, 1(2), 111-126.
- Önder E. ve Yıldırım, F. (2018), *Operasyonel, Yönetmel ve Stratejik Problemlerin Çözümünde Çok Kriterli Karar Verme Yöntemleri*. Bursa: Dora, 15-20.
- Özbek, A. (2013), Performance evaluation of learning management system. *NWSA-Education Sciences*, 8 (2), 164-178.
- Özbek, A. (2017), *Çok Kriterli Karar Verme Yöntemleri*. Ankara: Seçkin, 71-184.
- Özcan, D. (2017). Evlat edinmede rıza. *İstanbul Üniversitesi Hukuk Fakültesi Mecmuası*, 2, 707-722.
- Özden, Ü.H. (2009) *Türkiye'deki Mevduat Bankalarının Performansları: Çok Kriterli Karar Verme Yöntemleri ile Analizi*, Ankara: Detay Yayıncılık.
- Özel, S. ve Türkel, A. (2018). AHP yöntemi kullanarak ERP sistemlerinin karşılaştırılması ve uygun sistemin belirlenmesi. *Marmara Fen Bilimleri Dergisi*, 3, 305-317.
- Özmen, İ. (1999). Evlat edinme hukuku. *Türkiye Noterler Birliği Hukuk Dergisi*, 101, 25.
- Özmen, İ. (2004). *Evlat Edinme Davaları*, Ankara: Kartal Yayınevi, 26.

KAYNAKLAR DİZİNİ(devam)

Perera, N. ve Sutrisna, M. (2010). The use of analytic hierarchy process (AHP) in the analysis of delay claims in construction projects in the UAE. *The Built & Human Environment Review*, 3 (1), 29-47.

Rao R.V. (2006). Machinability evaluation of work material using a combined multiple attribute decision-making method. *The International Journal Of Advanced Manufacturing Technology*, 28(3-4), 221-227.

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operation Research*, 48 (1), 9-26.

Saaty, T.L. ve Wind, Y.,(1980). Marketing applications of the analytic hierarchy process. *Management Science*, 26, 641-658.

Savadogo, O. ve Shanian, A. (2006). TOPSIS multiple-criteria decision support analysis for material selection of metallic bipolar plates for polymer electrolyte fuel cell. *Journal Of Power Sources*, 159, 1095-1104.

Timor, M. (2011). *Analitik Hiyerarşi Prosesi*. İstanbul: Türkmen Kitabevi, 67.

Yavuz, M. (2012). Madencilik faaliyetlerinde bazı karar verme problemlerinin çözümünde TOPSIS yönteminin uygulanması. *MT Bilimsel*, 2, 21-34.

Yılmaz, R. (2008). Türkiye’de Lisansüstü Öğrenim İçin Öğrenci Seçimi: Kara Harp Okulu Savunma Bilimleri Enstitüsünde Bir Uygulama, Yüksek Lisans Tezi, Kara Harp Okulu, Savunma Bilimleri Enstitüsü, Ankara, 134.

Zahedi , F. (1986). The analytic hierarchy process: a survey of the method and its applications. *Interfaces*, 16(1), 96-108.

EKLER

Ek-1. C# Program Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Collections;
using System.IO;
using DevExpress.XtraCharts;
using System.Data.OleDb;
using SON.Data;

namespace SON
{
    public partial class Form1 : Form
    {
        public static ArrayList kriterler = new ArrayList();
        public static ArrayList alternatifler = new ArrayList();
        public static ArrayList dydizi = new ArrayList();
        public static ArrayList adizi = new ArrayList();
        public static ArrayList agirlikdgr = new ArrayList();
        public static ArrayList pozitifl = new ArrayList();
        public static ArrayList negatifl = new ArrayList();
        public static ArrayList sonuc = new ArrayList();
        public static ArrayList sonx = new ArrayList();
        public static ArrayList x1 = new ArrayList();
        public static ArrayList y1 = new ArrayList();
        public static ArrayList yer = new ArrayList();
        public static ArrayList maxdeger = new ArrayList();
        public static ArrayList maxmin = new ArrayList();
        public static ArrayList RG = new ArrayList();
        public static ArrayList siralama = new ArrayList();
        double genelCR;
        int sayac;

        public Form1()
        {
            InitializeComponent();
            listBox1.Items.Add("AİLENİN EKONOMİK DURUMU" + " - " + "(" + "Maximum" +
            ")");
            listBox1.Items.Add("BEKLEME SÜRESİ" + " - " + "(" + "Maximum" + ")");
            listBox1.Items.Add("EBEVEYNLİK KAPASİTESİ" + " - " + "(" + "Maximum" + ")");
            listBox1.Items.Add("AİLENİN EĞİTİM DURUMU" + " - " + "(" + "Maximum" + ")");
```

```

listBox1.Items.Add("AİLENİN İŞ DURUMU" + " - " + "(" + "Maximum" + ")");
listBox1.Items.Add("AİLENİN SAĞLIK DURUMU" + " - " + "(" + "Maximum" +
");");
listBox1.Items.Add("KONUTUN FİZİKİ ŞARTLARI" + " - " + "(" + "Maximum" +
");");
listBox1.Items.Add("AİLENİN SOSYAL YAŞANTISI" + " - " + "(" + "Maximum" +
");");
listBox1.Items.Add("AİLENİN DİĞER İNSANLARLA İLETİŞİMİ" + " - " + "(" +
"Maximum" + ")");
listBox1.Items.Add("ÇOCUKLU YAŞAMA DAİR BEKLENTİLER" + " - " + "(" +
"Maximum" + ")");
listBox1.Items.Add("AİLE BİREYLERİNİN KİŞİLİK ÖZELLİKLERİ" + " - " + "(" +
"Maximum" + ")");
listBox1.Items.Add("EVLİLİK YAŞANTISI/EŞLER ARASI İLETİŞİM" + " - " + "(" +
"Maximum" + ")");
listBox1.Items.Add("SOSYAL ÇALIŞMACININ GENEL KANAATİ" + " - " + "(" +
"Maximum" + ")");
kriterler.Add("AİLENİN EKONOMİK DURUMU");
kriterler.Add("BEKLEME SÜRESİ");
kriterler.Add("EBEVEYNLIK KAPASİTESİ");
kriterler.Add("AİLENİN EĞİTİM DURUMU");
kriterler.Add("AİLENİN İŞ DURUMU");
kriterler.Add("AİLENİN SAĞLIK DURUMU");
kriterler.Add("KONUTUN FİZİKİ ŞARTLARI");
kriterler.Add("AİLENİN SOSYAL YAŞANTISI");
kriterler.Add("AİLENİN DİĞER İNSANLARLA İLETİŞİMİ");
kriterler.Add("ÇOCUKLU YAŞAMA DAİR BEKLENTİLER");
kriterler.Add("AİLE BİREYLERİNİN KİŞİLİK ÖZELLİKLERİ");
kriterler.Add("EVLİLİK YAŞANTISI/EŞLER ARASI İLETİŞİM");
kriterler.Add("SOSYAL ÇALIŞMACININ GENEL KANAATİ");

maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");
maxmin.Add("Maximum");

simpleButton10.Visible = false;
simpleButton7.Visible = false;
}
private void button1_Click(object sender, EventArgs e)
{
    //for (int i = 0; i < kriterler.Count; i++)

```

```

    //{
        // if (textBox1.Text == kriterler[i].ToString())
        // {
            //      MessageBox.Show("Lütfen farklı kriterler tercih ediniz!", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            //      return;
        // }
    //}
    //if (textBox1.Text == "")
    //{
        //      return;
    //}
    this.Width = 805;
        //if (radioButton1.Checked == true)
        //{
            //      listBox1.Items.Add(textBox1.Text + " - " + "(" + radioButton1.Text + ")");
        //}
        //if (radioButton2.Checked == true)
        //{
            //      listBox1.Items.Add(textBox1.Text + " - " + "(" + radioButton2.Text + ")");
        //}
        //kriterler.Add(textBox1.Text);
        //button13.Visible = true;
        button3.Visible = true;
        //if (radioButton1.Checked == true)
        //{
            //      maxmin.Add(radioButton1.Text);
        //}
        //if (radioButton2.Checked == true)
        //{
            //      maxmin.Add(radioButton2.Text);
        //}
    }
    private void button3_Click(object sender, EventArgs e)
    {
        button6.Visible = false;
        label8.Visible = false;
        label7.Visible = false;

        button3.BackColor = Color.LightYellow;
        button3.ForeColor = Color.Black;
        button3.Enabled = false;
    this.Height = 723;
    this.Width = 900;
        button13.Enabled = false;
        panel1.Enabled = false;
        progressBar1.Visible = true;
        dataGridView1.Visible = true;
        button2.Visible = true;
        button3.Enabled = false;
    int v = 1;

```

```

dataGridView1.ColumnCount = kriterler.Count + 1;
dataGridView1.Columns[0].Name = " ";

for (int i = 0; i < kriterler.Count; i++)
{
    v = 1;
    for (int j = 0; j < kriterler.Count; j++)
    {
        dataGridView1.Columns[v].Name = kriterler[j].ToString();
        v++;
    }
}

for (int i = 0; i < kriterler.Count; i++)
{
    v = 1;
    for (int j = 0; j < kriterler.Count; j++)
    {
        dataGridView1.Columns[v].Name = kriterler[j].ToString();
        v++;
    }
}

for (int j = 0; j < kriterler.Count; j++)
{
    dataGridView1.Rows.Add(kriterler[j].ToString());
}

for (int qe = 0; qe < kriterler.Count; qe++)
{
    dataGridView1.Rows[qe].Cells[0].ReadOnly = true;
}

for (int i = 0; i < kriterler.Count; i++)
{
    dataGridView1.Rows[i].Cells[i + 1].Value = 1;
}
}

private void Form1_Load(object sender, EventArgs e)
{
    this.FormBorderStyle = FormBorderStyle.FixedSingle;
    Rectangle rect = new Rectangle(
        tabPage1.Left,
        tabPage1.Top,
        tabPage1.Width,
        tabPage1.Height);
    tabControl1.Region = new Region(rect);
    timer1.Start();
    timer1.Interval = 100;
    this.Width = 500;
    this.Height = 200;
    //radioButton1.Checked = true;
    RG.Add(0);
    RG.Add(0);
    RG.Add(0.58);
}

```

```

RG.Add(0.9);
RG.Add(1.12);
RG.Add(1.24);
RG.Add(1.32);
RG.Add(1.41);
RG.Add(1.45);
RG.Add(1.49);
RG.Add(1.49);
RG.Add(1.49);
RG.Add(1.49);
    }
private void button2_Click_1(object sender, EventArgs e) // AHP İŞLEMLERİ
{
    try
    {
        button24.Visible = false;
        dataGridView1.Visible = false;
        for (int i = 0; i < kriterler.Count; i++)
        {
            for (int j = 1; j < kriterler.Count + 1; j++)
            {
                if (Convert.ToDouble(dataGridView1.Rows[i].Cells[j].Value) < 0 ||
                    Convert.ToDouble(dataGridView1.Rows[i].Cells[j].Value) > 0)
                {
                }
            }
            else
            {
                MessageBox.Show("Lütfen boş değerleri doldurunuz!", "UYARI",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
                return;
            }
        }
    }
    int v = 1;
        dataGridView2.ColumnCount = kriterler.Count + 1;
        dataGridView2.Columns[0].Name = " ";
        for (int i = 0; i < kriterler.Count; i++)
        {
            v = 1;
            for (int j = 0; j < kriterler.Count; j++)
            {
                dataGridView2.Columns[v].Name = kriterler[j].ToString();
                v++;
            }
        }
        for (int i = 0; i < kriterler.Count; i++)
        {
            v = 1;
            for (int j = 0; j < kriterler.Count; j++)
            {
                dataGridView2.Columns[v].Name = kriterler[j].ToString();

```

```

v++;
    }
    }
    for (int j = 0; j < kriterler.Count; j++)
    {
        dataGridView2.Rows.Add(kriterler[j].ToString());
    }
    int h = 0;
    double[] dizi = new double[kriterler.Count * kriterler.Count];
    for (int j = 1; j < kriterler.Count + 1/*3*/; j++)
    {
        for (int i = 0; i < kriterler.Count/*2*/; i++)
        {
            dizi[h] += Convert.ToDouble(dataGridView1.Rows[i].Cells[j].Value);
        }
        h++;
    }
    h = 0;
    for (int j = 0; j < kriterler.Count; j++)
    {
        for (int i = 1; i < kriterler.Count + 1; i++)
        {
            dataGridView2.Rows[j].Cells[i].Value
dataGridView1.Rows[j].Cells[i].Value;
        }
        dataGridView2.Rows[kriterler.Count].Cells[0].Value = "TOPLAM";
        for (int j = 1; j < kriterler.Count + 1; j++)
        {
            dataGridView2.Rows[kriterler.Count].Cells[j].Value = dizi[h];
        }
        h++;
    }
    for (int i = 0; i < dizi.Length; i++)
    {
        ydizi.Add(dizi[i]);
    }
    button2.Enabled = false;
    progressBar1.Value = 33;
    //dataGridView2.Visible = true;

    progressBar1.Value = 66;
    //dataGridView3.Visible = true;
    int s = 1;
    dataGridView3.ColumnCount = kriterler.Count + 1;
    dataGridView3.Columns[0].Name = " ";
    for (int i = 0; i < kriterler.Count; i++)
    {
        s = 1;
        for (int j = 0; j < kriterler.Count; j++)
        {
            dataGridView3.Columns[s].Name = kriterler[j].ToString();

```

```

ş++;
    }
    }
    for (int i = 0; i < kriterler.Count; i++)
    {
        int ç = 1;
        for (int j = 0; j < kriterler.Count; j++)
        {
            dataGridView3.Columns[ç].Name = kriterler[j].ToString();
            ç++;
        }
    }
    int p = 0;
    for (int j = 0; j < kriterler.Count; j++)
    {
        dataGridView3.Rows.Add(kriterler[j].ToString());
    }
    for (int j = 1; j < kriterler.Count + 1; j++)
    {
        for (int i = 0; i < kriterler.Count; i++)
        {
            dataGridView3.Rows[i].Cells[j].Value
            =
            (Convert.ToDouble(dataGridView2.Rows[i].Cells[j].Value) / Convert.ToDouble(dizi[p]));
        }
        p++;
    }
    dataGridView5.Visible = true;
    button7.Visible = true;
    // -----//
    progressBar1.Value = 100;
    progressBar1.Width = 1023;
    //dataGridView4.Visible = true;
    double[] dizi2 = newdouble[kriterler.Count];
    double[] son = newdouble[kriterler.Count];
    int te = 0;
    for (int j = 0; j < kriterler.Count/*3*/; j++)
    {
        for (int i = 1; i < kriterler.Count + 1/*2*/; i++)
        {
            dizi2[te] += Convert.ToDouble(dataGridView3.Rows[j].Cells[i].Value);
        }
        te++;
    }
    for (int i = 0; i < dizi2.Length; i++)
    {
        son[i] = dizi2[i] / kriterler.Count;
    }
    dataGridView4.ColumnCount = kriterler.Count + 2;
    dataGridView4.Columns[0].Name = " ";
    dataGridView4.Columns[kriterler.Count + 1].Name = "Satır Ortalaması";
    int n = 1;

```

```

for (int i = 0; i <kriterler.Count; i++)
{
    dataGridView4.Columns[n].Name = kriterler[i].ToString();
    n++;
}
for (int i = 0; i <kriterler.Count; i++)
{
    dataGridView4.Rows.Add(kriterler[i].ToString());
}
for (int j = 1; j <kriterler.Count + 1; j++)
{
    for (int i = 0; i <kriterler.Count; i++)
    {
        dataGridView4.Rows[i].Cells[j].Value =
dataGridView3.Rows[i].Cells[j].Value;
    }
}
int w = 0;
for (int t = 0; t <kriterler.Count; t++)
{
    dataGridView4.Rows[t].Cells[kriterler.Count + 1].Value =
Math.Round(son[w], 4);
    w++;
}
dataGridView5.ColumnCount = 2;
dataGridView5.Columns[0].Name = "Kriterler";
dataGridView5.Columns[1].Name = "Ağırlık";
for (int i = 0; i <kriterler.Count; i++)
{
    dataGridView5.Rows.Add(kriterler[i].ToString());
}
for (int i = 0; i <kriterler.Count; i++)
{
    dataGridView5.Rows[i].Cells[1].Value = Math.Round(son[i], 4);
}
for (int i = 0; i <son.Length; i++)
{
    sonx.Add(son[i]);
}
// TUTARLILIK HESAPLAMASI
double[] RGtoplamlam = newdouble[kriterler.Count * kriterler.Count];
intsayac = 0;
int sayac2 = 0;
for (int i = 1; i <kriterler.Count + 1; i++)
{
    for (int j = 0; j <kriterler.Count; j++)
    {
        RGtoplamlam[sayac] = Convert.ToDouble(dataGridView1.Rows[j].Cells[i].Value) *
Math.Round(son[sayac2], 4);
        sayac++;
    }
}

```

```

sayac2++;
    }
double[] sly = newdouble[kriterler.Count];
for (int i = 0; i <kriterler.Count; i++)
    {
for (int z = i; z <RGtoplam.Length; z = z + kriterler.Count)
    {
sly[i] += RGtoplam[z];
    }
    }
double[] tutarlilik = newdouble[kriterler.Count];
doubletutarliliksonuc = 0;
for (int i = 0; i <sly.Length; i++)
    {
sly[i] = Math.Round(sly[i], 4);
    }
for (int i = 0; i <tutarlilik.Length; i++)
    {
tutarlilik[i] = sly[i] / Math.Round(son[i], 4);
tutarliliksonuc += tutarlilik[i];
    }
double gösterge = 0;
gösterge = tutarliliksonuc / kriterler.Count;
gösterge = Math.Round(gösterge, 4);
double CI = 0;
CI = (gösterge -kriterler.Count) / (kriterler.Count - 1);
double CR = 0;
CR = CI / 1.56;//Convert.ToDouble(RG[kriterler.Count - 1]);
if (CR >= 0.1)
    {
        //for (int i = 0; i <sonx.Count; i++)
        //{
        //    sonx.RemoveAt(i);
        //}
        //dataGridView2.Columns.Clear();
        //dataGridView3.Columns.Clear();
        //dataGridView4.Columns.Clear();
        //dataGridView5.Columns.Clear();
        //dataGridView2.Visible = false;
        //dataGridView3.Visible = false;
        //dataGridView4.Visible = false;
        //dataGridView5.Visible = false;
        //button7.Visible = false;
        //button2.Enabled = true;
        //this.Width = 805;
        MessageBox.Show("Tutarlılık Oranı: " + "0.1' den küçük olmalıdır." + " Lütfen
kriterleri kontrol ediniz!");
        //CR.ToString()
return;
    }
else

```

```

        {
            if (kriterler.Count > 2)
            {
                CR = Math.Round(CR, 8);
                label8.Text = CR.ToString();
            }
            else
            {
                label8.Text = "0";
            }
            genelCR = CR;
            this.Width = 1057;
            button2.Visible = false;
            dataGridView1.Height = 96;
            dataGridView1.Width = 769;
            dataGridView5.Visible = true;
            button7.Visible = true;
            label7.Visible = true;
            label8.Visible = true;
            button6.Visible = true;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lütfen metinsel değerler girmeyiniz!", "UYARI",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }
    }
}

```

```

    }
    private void button5_Click(object sender, EventArgs e)
    {
        /* double[] msgBox = new double[son.Length];
        for (int i = 0; i < msgBox.Length; i++)
        {
            msgBox[i] = Math.Round(son[i], 4);
        }
        string[] msgBoxKriter = new string[kriterler.Count];
        for (int i = 0; i < kriterler.Count; i++)
        {
            msgBoxKriter[i] = kriterler[i].ToString();
        }
        string input = string.Join("\n", msgBoxKriter);
        string output = string.Join("\n", msgBox);
        MessageBox.Show("Kriter Ağırlıkları" + "\n" + "-----" + "\n" + input + output
            + "\n" + "-----" + "\n" + "Bu değerleri TOPSIS'de kullanmak istiyor musunuz? ",
            "Belirleme Penceresi", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
        */
    }
    private void textBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)

```

```

        {
            button1.PerformClick();
            //textBox1.Clear();
            button3.Visible = true;
            //button13.Visible = true;
this.Width = 805;
            //button6.Enabled = true;
e.Handled = true;
e.SuppressKeyPress = true;
        }
    }
    private void Form1_Shown(object sender, EventArgs e)
    {
        //textBox1.Focus();
    }
    private void button6_Click(object sender, EventArgs e)
    {
        //kriterler.Remove(textBox1.Text);
        //listBox1.Items.Remove(textBox1.Text + " - " + "(" + radioButton1.Text + ")");
        //listBox1.Items.Remove(textBox1.Text + " - " + "(" + radioButton2.Text + ")");

        //textBox1.Clear();
        //textBox1.Focus();
        //if (listBox1.Items.Count == 0)
        //{
        //    textBox1.Focus();
        //    button6.Enabled = false;
        //    this.Width = 478;
        //    this.Height = 200;
        //}
    }

    TOPSISEntities TOPSISEntities = new TOPSISEntities();

    private void button7_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 1;
        List<Parent> parentList = TOPSISEntities.Parent.ToList();

        foreach (var item in parentList)
        {
            comboBox1.Items.Add(item.FName + "-" + item.FSurname + "-" + item.FTCKN + "-"
+ item.MName + "-" + item.MSurname + "-" + item.MTCKN);
        }
        this.Height = 200;
        this.Width = 590;
        for (int i = 0; i < kriterler.Count; i++)
        {
            listBox2.Items.Add(kriterler[i].ToString());
        }
        textBox3.Focus();
    }

```

```

    }
    private void button8_Click(object sender, EventArgs e)
    {
        button9.Enabled = true;
        this.Width = 1073;

        if (comboBox1.SelectedIndex >= 0)
        {
            listBox3.Items.Add(comboBox1.Text);
            alternatifler.Add(comboBox1.Text);
        }
    }
    private void textBox3_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            button8.PerformClick();
            textBox3.Clear();
            e.Handled = true;
            e.SuppressKeyPress = true;
        }
    }

    private void button10_Click(object sender, EventArgs e)
    {
        this.Height = 729;
        this.Width = 1027;
        textBox3.Enabled = false;
        button10.Enabled = false;
        dataGridView6.Visible = true;
        button11.Visible = true;
        button8.Enabled = false;
        button9.Enabled = false;
        button25.Enabled = false;
        dataGridView6.ColumnCount = kriterler.Count + 1;
        dataGridView6.Columns[0].Name = " ";
        int j = 1;
        for (int i = 0; i < kriterler.Count; i++)
        {
            dataGridView6.Columns[j].Name = kriterler[i].ToString();
            j++;
        }
        for (int i = 0; i < alternatifler.Count; i++)
        {
            dataGridView6.Rows.Add(alternatifler[i].ToString());
        }
        for (int qe = 0; qe < alternatifler.Count; qe++)
        {
            dataGridView6.Rows[qe].Cells[0].ReadOnly = true;
        }
        for (int i = 0; i < alternatifler.Count; i++)

```

```

{
    stringdeger = alternatifler[i].ToString();
    string[] degerArr = deger.Split('-'); //2 ve 5
    stringftc = degerArr[2];
    stringmtc = degerArr[5];
    varparent = TOPSISEntities.Parent.Where(x =>x.FTCKN == ftc&& x.MTCKN ==
mtc).FirstOrDefault();
    if (parent != null)
    {
        varparentCriteria = TOPSISEntities.ParentCriteriaPuan.Where(x =>x.ParentID ==
parent.ID).FirstOrDefault();
        if (parentCriteria != null)
        {
            dataGridView6.Rows[i].Cells[1].Value =
parentCriteria.AileninEkonomikDurumu;
            dataGridView6.Rows[i].Cells[2].Value = parentCriteria.BeklemeSuresi;
            dataGridView6.Rows[i].Cells[3].Value =
parentCriteria.EbeveynlikKapasitesi;
            dataGridView6.Rows[i].Cells[4].Value =
parentCriteria.AileninEgitimDurumu;
            dataGridView6.Rows[i].Cells[5].Value =
parentCriteria.AileninIsDurumu;
            dataGridView6.Rows[i].Cells[6].Value =
parentCriteria.AileninSaglikDurumu;
            dataGridView6.Rows[i].Cells[7].Value =
parentCriteria.KonutunFizikiSartlari;
            dataGridView6.Rows[i].Cells[8].Value =
parentCriteria.AileninSosyalYasantisi;
            dataGridView6.Rows[i].Cells[9].Value =
parentCriteria.AileninDigerInsanlarlaIletisimi;
            dataGridView6.Rows[i].Cells[10].Value =
parentCriteria.CocukluYasamaDairBeklentiler;
            dataGridView6.Rows[i].Cells[11].Value =
parentCriteria.AileBireylerininKisilikOzellikleri;
            dataGridView6.Rows[i].Cells[12].Value =
parentCriteria.EvlilikYasantisi;
            dataGridView6.Rows[i].Cells[13].Value =
parentCriteria.SosyalYasamcınınGenelKanaati;
        }
    }
    else
    {
        dataGridView6.Rows[i].Cells[1].Value = 1;
        dataGridView6.Rows[i].Cells[2].Value = 1;
        dataGridView6.Rows[i].Cells[3].Value = 1;
        dataGridView6.Rows[i].Cells[4].Value = 1;
        dataGridView6.Rows[i].Cells[5].Value = 1;
        dataGridView6.Rows[i].Cells[6].Value = 1;
        dataGridView6.Rows[i].Cells[7].Value = 1;
        dataGridView6.Rows[i].Cells[8].Value = 1;
        dataGridView6.Rows[i].Cells[9].Value = 1;
        dataGridView6.Rows[i].Cells[10].Value = 1;
    }
}

```

```

dataGridView6.Rows[i].Cells[11].Value = 1;
    dataGridView6.Rows[i].Cells[12].Value = 1;
    dataGridView6.Rows[i].Cells[13].Value = 1;
    }
    }
    }
}

private void button11_Click(object sender, EventArgs e)
{
    button5.Visible = true;
    //simpleButton1.Visible = true;
    simpleButton5.Visible = false;

    try
    {
        for (int i = 0; i < alternatifler.Count; i++)
        {
            for (int j = 1; j < kriterler.Count + 1; j++)
            {
                if (Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value) < 0 ||
                    Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value) > 0 ||
                    Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value) == 0)
                {
                }
                else
                {
                    simpleButton1.Visible = false;
                    MessageBox.Show("Lütfen boş değerleri doldurunuz!", "UYARI",
                        MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    return;
                }
            }
        }

        double[] dizix = new double[kriterler.Count];
        dataGridView7.ColumnCount = kriterler.Count + 1;
        dataGridView7.Columns[0].Name = " ";
        int k = 1;
        for (int i = 0; i < kriterler.Count; i++)
        {
            dataGridView7.Columns[k].Name = kriterler[i].ToString();
            k++;
        }
        for (int i = 0; i < alternatifler.Count; i++)
        {
            dataGridView7.Rows.Add(alternatifler[i].ToString());
        }
        for (int i = 0; i < alternatifler.Count; i++) // alternatifler sütün | kriterler satır //
        {
            for (int j = 1; j < kriterler.Count + 1; j++)
            {

```

```

        dataGridView7.Rows[i].Cells[j].Value
Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value)
Convert.ToDouble(dataGridView6.Rows[i].Cells[j].Value);
    }
}
int s = 1;
for (int j = 0; j < kriterler.Count; j++)
{
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dizix[j] += (double)dataGridView7.Rows[i].Cells[s].Value;
    }
    s++;
}
for (int i = 0; i < dizix.Length; i++)
{
    dizix[i] = Math.Sqrt(dizix[i]);
}
dataGridView7.Rows[alternatifler.Count].Cells[0].Value = "KAREKÖK
TOPLAM";
for (int j = 1; j < kriterler.Count + 1; j++)
{
    dataGridView7.Rows[alternatifler.Count].Cells[j].Value = dizix[j - 1];
}

for (int i = 0; i < dizix.Length; i++)
{
    adizi.Add(dizix[i]);
}
dataGridView8.ColumnCount = kriterler.Count + 1;
dataGridView8.Columns[0].Name = " ";
k = 1;
for (int i = 0; i < kriterler.Count; i++)
{
    dataGridView8.Columns[k].Name = kriterler[i].ToString();
}
k++;
for (int i = 0; i < alternatifler.Count; i++)
{
    dataGridView8.Rows.Add(alternatifler[i].ToString());
}
s = 0;
for (int i = 1; i < kriterler.Count + 1; i++)
{
    for (int j = 0; j < alternatifler.Count; j++)
    {
        dataGridView8.Rows[j].Cells[i].Value
Convert.ToDouble(dataGridView6.Rows[j].Cells[i].Value) / Convert.ToDouble(adizi[s]);
    }
    s++;
}

```

```

// AHP'den veri çektiğim yer //
double[] roundson = newdouble[sonx.Count];
    dataGridview10.Visible = true;
    dataGridview18.Visible = true;
    button23.Visible = true;
    dataGridview10.ColumnCount = kriterler.Count + 1;
    dataGridview10.Columns[0].Name = "Ağırlık";
int r = 1;
for (int i = 0; i < kriterler.Count; i++)
{
    roundson[i] = Convert.ToDouble(dataGridview5.Rows[i].Cells[1].Value);

}
for (int i = 0; i < kriterler.Count; i++)
{
    dataGridview10.Columns[r].Name = roundson[i].ToString();
    r++;
}
dataGridview9.ColumnCount = kriterler.Count + 1;
dataGridview9.Columns[0].Name = " ";
k = 1;
for (int i = 0; i < kriterler.Count; i++)
{
    dataGridview9.Columns[k].Name = kriterler[i].ToString();
    k++;
}
for (int i = 0; i < alternatifler.Count; i++)
{
    dataGridview9.Rows.Add(alternatifler[i].ToString());
}
int c = 0;
for (int i = 1; i < kriterler.Count + 1; i++)
{
    for (int j = 0; j < alternatifler.Count; j++)
    {
        dataGridview9.Rows[j].Cells[i].Value
        =
        Convert.ToDouble(dataGridview8.Rows[j].Cells[i].Value) * roundson[c];
    }
    c++;
}
s = 0;
doublemax = 0;
doublemin = 999999999;
double[] ideal = newdouble[kriterler.Count];
double[] negatifler = newdouble[kriterler.Count];
    dataGridview11.ColumnCount = kriterler.Count + 1;
    dataGridview11.Columns[0].Name = "İdeal Çözüm Değerleri";
for (int i = 1; i < kriterler.Count + 1; i++)
{
    if (maxmin[s] == "Maximum")
    {

```

```

max = 0;
for (int j = 0; j < alternatifler.Count; j++)
{
    if (Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value) > max)
    max = Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value);
}
ideal[s] = max;
}
if (maxmin[s] == "Minimum")
{
    min = 999999999;
    for (int j = 0; j < alternatifler.Count; j++)
    {
        if (Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value) < min)
        min = Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value);
    }
    ideal[s] = min;
}
s++;
}
for (int i = 0; i < ideal.Length; i++)
{
    pozitifler.Add(ideal[i]);
}
k = 1;
for (int i = 0; i < kriterler.Count; i++)
{
    dataGridView11.Columns[k].Name = ideal[i].ToString();
    k++;
}
int g = 0;
dataGridView12.ColumnCount = kriterler.Count + 1;
dataGridView12.Columns[0].Name = "Negatif İdeal Çözüm Değerleri";
for (int i = 1; i < kriterler.Count + 1; i++)
{
    if (maxmin[g] == "Maximum")
    {
        min = 999999999;
        for (int j = 0; j < alternatifler.Count; j++)
        {
            if (Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value) < min)
            min = Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value);
        }
        negatifler[g] = min;
    }
    if (maxmin[g] == "Minimum")
    {
        max = 0;
        for (int j = 0; j < alternatifler.Count; j++)
        {
            if (Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value) > max)

```

```

max = Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value);
    }
    negatifler[g] = max;
    }
    g++;
    }
    for (int i = 0; i <negatifler.Length; i++)
    {
        negatifs.Add(negatifler[i]);
    }
    int q = 1;
    for (int i = 0; i <kriterler.Count; i++)
    {
        dataGridView12.Columns[q].Name = negatifler[i].ToString();
        q++;
    }
    dataGridView13.ColumnCount = kriterler.Count + 1;
    dataGridView13.Columns[0].Name = " ";
    k = 1;
    for (int i = 0; i <kriterler.Count; i++)
    {
        dataGridView13.Columns[k].Name = kriterler[i].ToString();
        k++;
    }
    for (int i = 0; i <alternatifler.Count; i++)
    {
        dataGridView13.Rows.Add(alternatifler[i].ToString());
    }
    int y = 0;
    for (int i = 1; i <kriterler.Count + 1; i++)
    {
        for (int j = 0; j <alternatifler.Count; j++)
        {
            dataGridView13.Rows[j].Cells[i].Value
Math.Pow(Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value)
Convert.ToDouble(pozitifler[y]), 2);
        }
        y++;
    }
    double[] x = newdouble[alternatifler.Count];
    dataGridView14.ColumnCount = kriterler.Count + 3;
    dataGridView14.Columns[0].Name = " ";
    k = 1;
    for (int i = 0; i <kriterler.Count; i++)
    {
        dataGridView14.Columns[k].Name = kriterler[i].ToString();
        k++;
    }
    dataGridView14.Columns[kriterler.Count + 1].Name = "TOPLAM";
    dataGridView14.Columns[kriterler.Count + 2].Name = "Si +";
    for (int i = 0; i <alternatifler.Count; i++)

```

```

        {
dataGridView14.Rows.Add(alternatifler[i].ToString());
        }
for (int i = 0; i < alternatifler.Count; i++)
    {
for (int j = 1; j < kriterler.Count + 1; j++)
    {
x[i] += Convert.ToDouble(dataGridView13.Rows[i].Cells[j].Value);
    }
    }
for (int i = 0; i < alternatifler.Count; i++)
    {
for (int j = 1; j < kriterler.Count + 1; j++)
    {
        dataGridView14.Rows[i].Cells[j].Value =
dataGridView13.Rows[i].Cells[j].Value;
    }
    }
int t = 0;
for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView14.Rows[i].Cells[kriterler.Count + 1].Value = x[t];
t++;
    }
int w = 0;
for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView14.Rows[i].Cells[kriterler.Count + 2].Value =
Math.Sqrt(x[w]);
w++;
    }
for (int i = 0; i < alternatifler.Count; i++)
    {
x1.Add(Math.Sqrt(x[i]));
    }
    dataGridView15.ColumnCount = kriterler.Count + 1;
    dataGridView15.Columns[0].Name = " ";
k = 1;
for (int i = 0; i < kriterler.Count; i++)
    {
        dataGridView15.Columns[k].Name = kriterler[i].ToString();
k++;
    }
for (int i = 0; i < alternatifler.Count; i++)
    {
dataGridView15.Rows.Add(alternatifler[i].ToString());
    }
y = 0;
for (int i = 1; i < kriterler.Count + 1; i++)
    {
for (int j = 0; j < alternatifler.Count; j++)

```

```

        {
            dataGridView15.Rows[j].Cells[i].Value
Math.Pow(Convert.ToDouble(dataGridView9.Rows[j].Cells[i].Value)
Convert.ToDouble(negatifs[y]), 2);
        }
        y++;
    }
    double[] xe = newdouble[alternatifler.Count];
    dataGridView16.ColumnCount = kriterler.Count + 3;
    dataGridView16.Columns[0].Name = " ";
    k = 1;
    for (int i = 0; i < kriterler.Count; i++)
    {
        dataGridView16.Columns[k].Name = kriterler[i].ToString();
        k++;
    }
    dataGridView16.Columns[kriterler.Count + 1].Name = "TOPLAM";
    dataGridView16.Columns[kriterler.Count + 2].Name = "Si -";
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView16.Rows.Add(alternatifler[i].ToString());
    }
    for (int i = 0; i < alternatifler.Count; i++)
    {
        for (int j = 1; j < kriterler.Count + 1; j++)
        {
            xe[i] += Convert.ToDouble(dataGridView15.Rows[i].Cells[j].Value);
        }
    }
    for (int i = 0; i < alternatifler.Count; i++)
    {
        for (int j = 1; j < kriterler.Count + 1; j++)
        {
            dataGridView16.Rows[i].Cells[j].Value
dataGridView15.Rows[i].Cells[j].Value;
        }
    }
    t = 0;
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView16.Rows[i].Cells[kriterler.Count + 1].Value = xe[t];
        t++;
    }
    w = 0;
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView16.Rows[i].Cells[kriterler.Count + 2].Value
Math.Sqrt(xe[w]);
        w++;
    }
    for (int i = 0; i < alternatifler.Count; i++)

```

```

        {
y1.Add(Math.Sqrt(xe[i]));
        }
        dataGridView17.ColumnCount = 2;
        dataGridView17.Columns[0].Name = "Si +";
        dataGridView17.Columns[1].Name = "Si -";
for (int i = 0; i <alternatifler.Count; i++)
    {
dataGridView17.Rows.Add(x1[i]);
    }
for (int i = 0; i <alternatifler.Count; i++)
    {
        dataGridView17.Rows[i].Cells[1].Value = y1[i];
    }
// SON TABLO //
// ----- //
dataGridView18.ColumnCount = 4;
dataGridView18.Columns[0].Name = "Aday";
dataGridView18.Columns[1].Name = "Si +";
dataGridView18.Columns[2].Name = "Si -";
dataGridView18.Columns[3].Name = "Ci +";
for (int i = 0; i <alternatifler.Count; i++)
    {
dataGridView18.Rows.Add(alternatifler[i].ToString());
    }
for (int i = 0; i <alternatifler.Count; i++)
    {
        dataGridView18.Rows[i].Cells[1].Value = x1[i];
    }
for (int i = 0; i <alternatifler.Count; i++)
    {
        dataGridView18.Rows[i].Cells[2].Value = y1[i];
    }
for (int i = 0; i <alternatifler.Count; i++)
    {
        dataGridView18.Rows[i].Cells[3].Value = Convert.ToDouble(y1[i]) /
(Convert.ToDouble(y1[i]) + Convert.ToDouble(x1[i]));
        yer.Add(Convert.ToDouble(y1[i]) / (Convert.ToDouble(y1[i]) +
Convert.ToDouble(x1[i])));
    }
max = 0;
for (int i = 0; i <yer.Count; i++)
    {
if (Convert.ToDouble(yer[i]) >max)
    {
max = Convert.ToDouble(yer[i]);
    }
    }
doublesira = yer.IndexOf(max);
for (int i = 0; i < dataGridView18.Rows.Count; i++)
    {

```

```

        if (Convert.ToDouble(dataGridView18.Rows[i].Cells[3].Value) == max)
        {
            dataGridView18.Rows[i].DefaultCellStyle.BackColor =
Color.MediumVioletRed;
            dataGridView18.Rows[i].DefaultCellStyle.ForeColor = Color.White;
        }
    }

    dataGridView18.Sort(dataGridView18.Columns["Ci
ListSortDirection.Descending);

    dataGridView20.ColumnCount = 2;
    dataGridView20.Columns[0].Name = "Alternatifler";
    dataGridView20.Columns[1].Name = "Sıralamaları";
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView20.Rows.Add(alternatifler[i].ToString());
    }
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView20.Rows[i].Cells[1].Value = yer[i];
    }
    dataGridView20.Sort(dataGridView20.Columns[1],
ListSortDirection.Descending); //Normal Sıralama
    for (int i = 0; i < alternatifler.Count; i++)
    {
        dataGridView20.Rows[i].Cells[1].Value = i + 1;
    }
    button11.Enabled = false;
    simpleButton6.Visible = true;
}

catch (Exception ex)
{
    simpleButton1.Visible = false;
    MessageBox.Show("Lütfen metinsel değerler girmeyiniz!", "UYARI",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}

private void dataGridView1_CellValidated(object sender, DataGridViewCellEventArgs
e)
{
    try
    {
        int y = dataGridView1.CurrentCell.ColumnIndex; //x=0 | y=4
        int x = dataGridView1.CurrentCell.RowIndex + 1;
        if (y != 0 && x != y)
        {
            dataGridView1.Rows[y - 1].Cells[x].Value = 1 /
Convert.ToDouble(dataGridView1.Rows[x - 1].Cells[y].Value);

```

```

            dataGridView1.Rows[y - 1].Cells[x].Value =
Math.Round(Convert.ToDouble(dataGridView1.Rows[y - 1].Cells[x].Value), 4);
        }
        if (y != 0 && x == y)
        {
            dataGridView1.Rows[y - 1].Cells[x].Value =
Convert.ToDouble(dataGridView1.Rows[y - 1].Cells[x].Value) /
Convert.ToDouble(dataGridView1.Rows[y - 1].Cells[x].Value);
            dataGridView1.Rows[y - 1].Cells[x].Value =
Math.Round(Convert.ToDouble(dataGridView1.Rows[y - 1].Cells[x].Value), 4);
        }
        if (dataGridView1.Rows[x - 1].Cells[y].Value != null)
        {
            if (y != 0 && Convert.ToDouble(dataGridView1.Rows[x - 1].Cells[y].Value) > 9)
            {
                MessageBox.Show("Lütfen 1'den 9'a kadar puanlayınız!", "UYARI",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                dataGridView1.Rows[x - 1].Cells[y].Value = 1;
                dataGridView1.Rows[y - 1].Cells[x].Value = 1;
            }
            if (y != 0 && Convert.ToDouble(dataGridView1.Rows[x - 1].Cells[y].Value) <= 0)
            {
                MessageBox.Show("Lütfen 1'den 9'a kadar puanlayınız!", "UYARI",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                dataGridView1.Rows[x - 1].Cells[y].Value = 1;
                dataGridView1.Rows[y - 1].Cells[x].Value = 1;
            }
        }
    }
}
catch (Exception)
{
    MessageBox.Show("Lütfen metinsel değerler girmeyiniz!", "UYARI",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
}

private void button23_Click(object sender, EventArgs e)
{
    double max = 0;
    for (int i = 0; i < yer.Count; i++)
    {
        if (Convert.ToDouble(yer[i]) > max)
        {
            max = Convert.ToDouble(yer[i]);
        }
    }
    int sira = yer.IndexOf(max);
    sira++;
    MessageBox.Show(sira.ToString() + ". sıradaki" + " " + "|" + alternatifler[--sira] + "|" +
    " en uygun çözüme sahiptir");
}

```

```

private void button13_Click_1(object sender, EventArgs e)
{
    panel1.Enabled = false;
    button13.BackColor = Color.LightYellow;
    button13.ForeColor = Color.Black;
    button13.Enabled = false;
    //textBox1.Enabled = false;
    button1.Enabled = false;
    //button6.Enabled = false;
    button17.Visible = true;
    button3.Enabled = false;
    this.Height = 350;
    int k = 1;
    dataGridView19.Visible = true;
    button16.Visible = true;
    dataGridView19.ColumnCount = kriterler.Count + 1;
    dataGridView19.Columns[0].Name = " ";
    k = 1;
    for (int j = 0; j < kriterler.Count; j++)
    {
        dataGridView19.Columns[k].Name = kriterler[j].ToString();
        k++;
    }
    for (int j = 0; j < 1; j++)
    {
        dataGridView19.Rows.Add("Ağırlıklar (virgül ile ayırınız)");
    }
}

private void button16_Click_1(object sender, EventArgs e)
{
    try
    {
        dataGridView5.ColumnCount = 2;
        dataGridView5.Columns[0].Name = "Kriterler";
        dataGridView5.Columns[1].Name = "Ağırlık";
        for (int i = 0; i < kriterler.Count; i++)
        {
            dataGridView5.Rows.Add(kriterler[i].ToString());
        }
        for (int i = 0; i < kriterler.Count; i++)
        {
            dataGridView5.Rows[i].Cells[1].Value = 0;
        }
        this.Height = 200;
        this.Width = 590;
        for (int i = 1; i < kriterler.Count + 1; i++)
        {
            sonx.Add(dataGridView19.Rows[0].Cells[i].Value);
        }
        tabControl1.SelectedIndex = 1;
    }
}

```

```

for (int i = 0; i < kriterler.Count; i++)
{
    listBox2.Items.Add(kriterler[i].ToString());
}
textBox3.Focus();
for (int i = 0; i < kriterler.Count; i++)
{
    dataGridView5.Rows[i].Cells[1].Value = sonx[i];
}
}
catch (Exception)
{
    return;
}
}
double toplamlar;

private void button17_Click_1(object sender, EventArgs e)
{
    toplamlar = 0;
    for (int i = 1; i < kriterler.Count + 1; i++)
    {
        toplamlar += Convert.ToDouble(dataGridView19.Rows[0].Cells[i].Value);
    }
    if (toplamlar == 1 || toplamlar > 0.998)
    {
        button16.Enabled = true;
    }
    else
    {
        button16.Enabled = false;
        MessageBox.Show("Lütfen girilen değerleri kontrol ediniz!", "UYARI",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

private void dataGridView19_CellEnter(object sender, DataGridViewCellEventArgs e)
{
    button16.Enabled = false;
}

private void button9_Click(object sender, EventArgs e)
{
    alternatifler.Remove(textBox3.Text);
    listBox3.Items.Remove(textBox3.Text);
    textBox3.Clear();
    textBox3.Focus();
    if (listBox3.Items.Count == 0)
    {
        button9.Enabled = false;
        textBox3.Focus();
    }
    this.Width = 587;
}

```

```

    }
    }
    private void listBox3_Click(object sender, EventArgs e)
    {
        textBox3.Text = listBox3.SelectedItem.ToString();
    }
    private void button19_Click_1(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 1;
    }
    private void simpleButton1_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 3;
        for (int i = 0; i < kriterler.Count; i++)
        {
            chartControl1.Series["Kriterler"].Points.Add(new DevExpress.XtraCharts.SeriesPoint(kriterler[i], Convert.ToDouble(dataGridView5.Rows[i].Cells[1].Value)));
        }
        for (int i = 0; i < alternatifler.Count; i++)
        {
            chartControl2.Series["Alternatifler"].Points.Add(new DevExpress.XtraCharts.SeriesPoint(alternatifler[i], Convert.ToDouble(dataGridView18.Rows[i].Cells[3].Value)));
        }
        this.Height = 400;
    }
    private void simpleButton2_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 1;
        this.Height = 723;
    }
    private void simpleButton3_Click(object sender, EventArgs e)
    {
        //SubFormsubForm = new SubForm();
        //subForm.Show();
        //this.Close();
        if (Application.OpenForms[0] != null) //Uygulamanın main form'u
        {
            //uygulamanın ana formunu kapatırsanız uygulama kapanır, ana formu
            yeniden baslatmak uygulamayı yeniden baslatmak demektir.
            Application.Restart();
        }
    }
    private void simpleButton4_Click(object sender, EventArgs e)
    {
        this.Height = 723;
        tabControl1.SelectedIndex = 2;
    }
    private void simpleButton6_Click(object sender, EventArgs e)
    {
        button5.Visible = false;
        button11.Visible = false;
    }

```

```

        dataGridView10.Location = newPoint(11, 527);
        dataGridView10.Width = 250;
        dataGridView6.Width = 250;
        button11.Width = 250;
        dataGridView18.Width = 339;
        button23.Width = 339;
        dataGridView18.Location = newPoint(628, 454);
        button23.Location = newPoint(628, 603);
        dataGridView7.Visible = true;
        dataGridView8.Visible = true;
        dataGridView9.Visible = true;
        dataGridView11.Visible = true;
        dataGridView12.Visible = true;
        dataGridView13.Visible = true;
        dataGridView14.Visible = true;
        dataGridView15.Visible = true;
        dataGridView16.Visible = true;
        dataGridView17.Visible = true;
        simpleButton6.Visible = false;
        simpleButton5.Visible = true;
        textBox3.Focus();
    }
    private void simpleButton5_Click(object sender, EventArgs e)
    {
        button5.Visible = true;
        dataGridView6.Width = 985;
        button11.Width = 985;
        dataGridView10.Location = newPoint(6, 273);
        dataGridView10.Width = 985;
        dataGridView18.Location = newPoint(8, 299);
        button23.Location = newPoint(6, 451);
        dataGridView18.Width = 985;
        button23.Width = 985;
        dataGridView7.Visible = false;
        button12.Visible = false;
        dataGridView8.Visible = false;
        dataGridView9.Visible = false;
        dataGridView11.Visible = false;
        dataGridView12.Visible = false;
        dataGridView13.Visible = false;
        button18.Visible = false;
        dataGridView14.Visible = false;
        dataGridView15.Visible = false;
        dataGridView16.Visible = false;
        dataGridView17.Visible = false;
        simpleButton5.Visible = false;
        simpleButton6.Visible = true;
        textBox3.Focus();
    }
    private void simpleButton2_Click_1(object sender, EventArgs e)
    {

```

```

        tabControl1.SelectedIndex = 1;
this.Height = 729;
this.Width = 1027;
    }
privatevoid simpleButton7_Click(object sender, EventArgs e)
    {
        MessageBox.Show("TOPSIS - AHP Uygulaması", "TOPSIS UYGULAMASI HAKKINDA", MessageBoxButtons.OK);
    }
privatevoid button5_Click_2(object sender, EventArgs e)
    {
        ; tabControl1.SelectedIndex = 4;
this.Height = 300;
this.Width = 330;
    }
privatevoid simpleButton8_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 1;
this.Height = 729;
this.Width = 1027;
    }
privatevoid simpleButton9_Click(object sender, EventArgs e)
    {
        pdfAktar.pdfKaydet(dataGridView20);
    }
privatevoid button4_Click_1(object sender, EventArgs e)
    {
        progressBar2.Value = 25;
        pictureBox9.Visible = false;
        label9.Visible = false;
        label10.Visible = false;
        pictureBox12.Visible = true;
        pictureBox13.Visible = true;
        pictureBox11.Visible = true;
        label11.Visible = true;
        label12.Visible = true;
        button14.Visible = true;
    }
privatevoid button14_Click_2(object sender, EventArgs e)
    {
        progressBar2.Value = 50;
        button4.Visible = false;
        pictureBox12.Visible = false;
        pictureBox13.Visible = false;
        pictureBox11.Visible = false;
        label11.Visible = false;
        label12.Visible = false;
        label13.Visible = true;
        label15.Visible = true;
        button14.Visible = false;
        pictureBox17.Visible = true;
    }

```

```

        pictureBox15.Visible = true;
        pictureBox16.Visible = true;
        pictureBox14.Visible = true;
        label14.Visible = true;
        button20.Visible = true;
    }
    private void button20_Click_1(object sender, EventArgs e)
    {
        progressBar2.Value = 75;
        pictureBox20.Visible = true;
        pictureBox19.Visible = true;
        label16.Visible = true;
        label17.Visible = true;
        pictureBox18.Visible = true;
        label13.Visible = false;
        label15.Visible = false;
        pictureBox17.Visible = false;
        pictureBox15.Visible = false;
        pictureBox16.Visible = false;
        pictureBox14.Visible = false;
        label14.Visible = false;
        button20.Visible = false;
        button21.Visible = true;
    }

    private void button21_Click_1(object sender, EventArgs e)
    {
        progressBar2.Value = 100;
        label18.Visible = true;
        pictureBox22.Visible = true;
        label19.Visible = true;
        pictureBox23.Visible = true;
        pictureBox24.Visible = true;
        pictureBox25.Visible = true;
        label20.Visible = true;
        label21.Visible = true;
        pictureBox21.Visible = true;
        button21.Visible = false;
        pictureBox20.Visible = false;
        pictureBox19.Visible = false;
        label16.Visible = false;
        label17.Visible = false;
        pictureBox18.Visible = false;
        button22.Visible = true;
    }

    private void button22_Click_1(object sender, EventArgs e)
    {
        if (kriterler.Count != 0)
        {
            button22.Visible = false;
            button4.Visible = true;
        }
    }

```

```

        label18.Visible = false;
        pictureBox22.Visible = false;
        label19.Visible = false;
        pictureBox23.Visible = false;
        pictureBox24.Visible = false;
        pictureBox25.Visible = false;
        label20.Visible = false;
        label21.Visible = false;
        pictureBox21.Visible = false;
        progressBar2.Value = 0;
        pictureBox9.Visible = true;
        label9.Visible = true;
        label10.Visible = true;
        tabControl1.SelectedIndex = 0;
    this.Width = 805;
    this.Height = 200;
    }
    else
    {
        button22.Visible = false;
        button4.Visible = true;
        progressBar2.Value = 0;
        label18.Visible = false;
        pictureBox22.Visible = false;
        label19.Visible = false;
        pictureBox23.Visible = false;
        pictureBox24.Visible = false;
        pictureBox25.Visible = false;
        label20.Visible = false;
        label21.Visible = false;
        pictureBox21.Visible = false;
        progressBar2.Value = 0;
        pictureBox9.Visible = true;
        label9.Visible = true;
        label10.Visible = true;
        tabControl1.SelectedIndex = 0;
    this.Width = 500;
    this.Height = 200;
    }
    }
    private void simpleButton10_Click(object sender, EventArgs e)
    {
        tabControl1.SelectedIndex = 5;
        this.Height = 550;
        this.Width = 1000;
    }
    private void button6_Click_1(object sender, EventArgs e)
    {
        this.Width = 1053;
        this.Height = 729;
        button6.Visible = false;

```

```

//dataGridView5.Visible = false;
//button7.Visible = false;
//label7.Visible = false;
//label8.Visible = false;
dataGridView1.Visible = true;
dataGridView2.Visible = true;
dataGridView3.Visible = true;
dataGridView4.Visible = true;
dataGridView5.Size = new Size(234, 248);
dataGridView5.Location = new Point(dataGridView5.Location.X, 791);
dataGridView5.Location = new Point(dataGridView5.Location.Y, 131);
//797; 385
button7.Location = new Point(button7.Location.X, 797);
button7.Location = new Point(button7.Location.Y, 385);
button7.Size = new Size(234, 37);
//855; 442
label7.Location = new Point(label7.Location.X, 855);
label7.Location = new Point(label7.Location.Y, 442);

//895; 465
label8.Location = new Point(label8.Location.X, 895);
label8.Location = new Point(label8.Location.Y, 465);
}
private void button15_Click_1(object sender, EventArgs e)
{
this.Close();
SubForm subForm = new SubForm();
subForm.Show();
}
private void button24_Click_1(object sender, EventArgs e)
{
dataGridView1.Rows[0].Cells[1].Value = 1;
dataGridView1.Rows[0].Cells[1].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[0].Cells[2].Value = 3;
dataGridView1.Rows[0].Cells[3].Value = 0.250;
dataGridView1.Rows[0].Cells[4].Value = 0.333;
dataGridView1.Rows[0].Cells[5].Value = 3;
dataGridView1.Rows[0].Cells[6].Value = 0.333;
dataGridView1.Rows[0].Cells[7].Value = 3;
dataGridView1.Rows[0].Cells[8].Value = 3;
dataGridView1.Rows[0].Cells[9].Value = 3;
dataGridView1.Rows[0].Cells[10].Value = 0.333;
dataGridView1.Rows[0].Cells[11].Value = 0.333;
dataGridView1.Rows[0].Cells[12].Value = 0.333;
dataGridView1.Rows[0].Cells[13].Value = 0.333;
dataGridView1.Rows[1].Cells[1].Value = 0.333;
dataGridView1.Rows[1].Cells[2].Value = 1;
dataGridView1.Rows[1].Cells[2].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[1].Cells[3].Value = 0.200;
dataGridView1.Rows[1].Cells[4].Value = 0.333;
dataGridView1.Rows[1].Cells[5].Value = 0.200; //

```

```
dataGridView1.Rows[1].Cells[6].Value = 0.250;
dataGridView1.Rows[1].Cells[7].Value = 0.333;
dataGridView1.Rows[1].Cells[8].Value = 0.333;
dataGridView1.Rows[1].Cells[9].Value = 0.333;
dataGridView1.Rows[1].Cells[10].Value = 0.250;
dataGridView1.Rows[1].Cells[11].Value = 0.250;
dataGridView1.Rows[1].Cells[12].Value = 0.250;
dataGridView1.Rows[1].Cells[13].Value = 0.250;
dataGridView1.Rows[2].Cells[1].Value = 4;
dataGridView1.Rows[2].Cells[2].Value = 5;
dataGridView1.Rows[2].Cells[3].Value = 1;
dataGridView1.Rows[2].Cells[3].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[2].Cells[4].Value = 3;
dataGridView1.Rows[2].Cells[5].Value = 4;
dataGridView1.Rows[2].Cells[6].Value = 3;
dataGridView1.Rows[2].Cells[7].Value = 3;
dataGridView1.Rows[2].Cells[8].Value = 5;
dataGridView1.Rows[2].Cells[9].Value = 3;
dataGridView1.Rows[2].Cells[10].Value = 1;
dataGridView1.Rows[2].Cells[11].Value = 1;
dataGridView1.Rows[2].Cells[12].Value = 3;
dataGridView1.Rows[2].Cells[13].Value = 0.333;
dataGridView1.Rows[3].Cells[1].Value = 3;
dataGridView1.Rows[3].Cells[2].Value = 3;
dataGridView1.Rows[3].Cells[3].Value = 0.333;
dataGridView1.Rows[3].Cells[4].Value = 1;
dataGridView1.Rows[3].Cells[4].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[3].Cells[5].Value = 3;
dataGridView1.Rows[3].Cells[6].Value = 1;
dataGridView1.Rows[3].Cells[7].Value = 3;
dataGridView1.Rows[3].Cells[8].Value = 3;
dataGridView1.Rows[3].Cells[9].Value = 4;
dataGridView1.Rows[3].Cells[10].Value = 0.333;
dataGridView1.Rows[3].Cells[11].Value = 0.333;
dataGridView1.Rows[3].Cells[12].Value = 0.333;
dataGridView1.Rows[3].Cells[13].Value = 0.333;
dataGridView1.Rows[4].Cells[1].Value = 0.333;
dataGridView1.Rows[4].Cells[2].Value = 5;
dataGridView1.Rows[4].Cells[3].Value = 0.250;
dataGridView1.Rows[4].Cells[4].Value = 0.333;
dataGridView1.Rows[4].Cells[5].Value = 1;
dataGridView1.Rows[4].Cells[5].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[4].Cells[6].Value = 0.333;
dataGridView1.Rows[4].Cells[7].Value = 1;
dataGridView1.Rows[4].Cells[8].Value = 3;
dataGridView1.Rows[4].Cells[9].Value = 3;
dataGridView1.Rows[4].Cells[10].Value = 0.333;
dataGridView1.Rows[4].Cells[11].Value = 0.333;
dataGridView1.Rows[4].Cells[12].Value = 0.333;
dataGridView1.Rows[4].Cells[13].Value = 0.250;
dataGridView1.Rows[5].Cells[1].Value = 3;
```

```
dataGridView1.Rows[5].Cells[2].Value = 4;
dataGridView1.Rows[5].Cells[3].Value = 0.333;
dataGridView1.Rows[5].Cells[4].Value = 1;
dataGridView1.Rows[5].Cells[5].Value = 3;
dataGridView1.Rows[5].Cells[6].Value = 1;
dataGridView1.Rows[5].Cells[6].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[5].Cells[7].Value = 3;
dataGridView1.Rows[5].Cells[8].Value = 4;
dataGridView1.Rows[5].Cells[9].Value = 4;
dataGridView1.Rows[5].Cells[10].Value = 0.500;
dataGridView1.Rows[5].Cells[11].Value = 0.333;
dataGridView1.Rows[5].Cells[12].Value = 0.333;
dataGridView1.Rows[5].Cells[13].Value = 0.250;
dataGridView1.Rows[6].Cells[1].Value = 0.333;
dataGridView1.Rows[6].Cells[2].Value = 3;
dataGridView1.Rows[6].Cells[3].Value = 0.333;
dataGridView1.Rows[6].Cells[4].Value = 0.333;
dataGridView1.Rows[6].Cells[5].Value = 1;
dataGridView1.Rows[6].Cells[6].Value = 0.333;
dataGridView1.Rows[6].Cells[7].Value = 1;
dataGridView1.Rows[6].Cells[7].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[6].Cells[8].Value = 3;
dataGridView1.Rows[6].Cells[9].Value = 3;
dataGridView1.Rows[6].Cells[10].Value = 0.333;
dataGridView1.Rows[6].Cells[11].Value = 0.333;
dataGridView1.Rows[6].Cells[12].Value = 0.333;
dataGridView1.Rows[6].Cells[13].Value = 0.333;
dataGridView1.Rows[7].Cells[1].Value = 0.333;
dataGridView1.Rows[7].Cells[2].Value = 3;
dataGridView1.Rows[7].Cells[3].Value = 0.200;
dataGridView1.Rows[7].Cells[4].Value = 0.333;
dataGridView1.Rows[7].Cells[5].Value = 0.333;
dataGridView1.Rows[7].Cells[6].Value = 0.250;
dataGridView1.Rows[7].Cells[7].Value = 0.333;
dataGridView1.Rows[7].Cells[8].Value = 1;
dataGridView1.Rows[7].Cells[8].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[7].Cells[9].Value = 2;
dataGridView1.Rows[7].Cells[10].Value = 0.333;
dataGridView1.Rows[7].Cells[11].Value = 0.333;
dataGridView1.Rows[7].Cells[12].Value = 0.333;
dataGridView1.Rows[7].Cells[13].Value = 0.250;
dataGridView1.Rows[8].Cells[1].Value = 0.333;
dataGridView1.Rows[8].Cells[2].Value = 3;
dataGridView1.Rows[8].Cells[3].Value = 0.333;
dataGridView1.Rows[8].Cells[4].Value = 0.250;
dataGridView1.Rows[8].Cells[5].Value = 0.333;
dataGridView1.Rows[8].Cells[6].Value = 0.250;
dataGridView1.Rows[8].Cells[7].Value = 0.333;
dataGridView1.Rows[8].Cells[8].Value = 0.500;
dataGridView1.Rows[8].Cells[9].Value = 1;
dataGridView1.Rows[8].Cells[9].Style.BackColor = Color.CornflowerBlue;
```

```
dataGridView1.Rows[8].Cells[10].Value = 0.333;
dataGridView1.Rows[8].Cells[11].Value = 0.200;
dataGridView1.Rows[8].Cells[12].Value = 0.333;
dataGridView1.Rows[8].Cells[13].Value = 0.200;
dataGridView1.Rows[9].Cells[1].Value = 3;
dataGridView1.Rows[9].Cells[2].Value = 4;
dataGridView1.Rows[9].Cells[3].Value = 1;
dataGridView1.Rows[9].Cells[4].Value = 3;
dataGridView1.Rows[9].Cells[5].Value = 3;
dataGridView1.Rows[9].Cells[6].Value = 2;
dataGridView1.Rows[9].Cells[7].Value = 3;
dataGridView1.Rows[9].Cells[8].Value = 3;
dataGridView1.Rows[9].Cells[9].Value = 3;
dataGridView1.Rows[9].Cells[10].Value = 1;
dataGridView1.Rows[9].Cells[10].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[9].Cells[11].Value = 0.333;
dataGridView1.Rows[9].Cells[12].Value = 0.250;
dataGridView1.Rows[9].Cells[13].Value = 0.250;
dataGridView1.Rows[10].Cells[1].Value = 3;
dataGridView1.Rows[10].Cells[2].Value = 4;
dataGridView1.Rows[10].Cells[3].Value = 1;
dataGridView1.Rows[10].Cells[4].Value = 3;
dataGridView1.Rows[10].Cells[5].Value = 3;
dataGridView1.Rows[10].Cells[6].Value = 3;
dataGridView1.Rows[10].Cells[7].Value = 3;
dataGridView1.Rows[10].Cells[8].Value = 3;
dataGridView1.Rows[10].Cells[9].Value = 5;
dataGridView1.Rows[10].Cells[10].Value = 3;
dataGridView1.Rows[10].Cells[11].Value = 1;
dataGridView1.Rows[10].Cells[11].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[10].Cells[12].Value = 1;
dataGridView1.Rows[10].Cells[13].Value = 0.250;
dataGridView1.Rows[11].Cells[1].Value = 3;
dataGridView1.Rows[11].Cells[2].Value = 4;
dataGridView1.Rows[11].Cells[3].Value = 0.333;
dataGridView1.Rows[11].Cells[4].Value = 3;
dataGridView1.Rows[11].Cells[5].Value = 3;
dataGridView1.Rows[11].Cells[6].Value = 3;
dataGridView1.Rows[11].Cells[7].Value = 3;
dataGridView1.Rows[11].Cells[8].Value = 3;
dataGridView1.Rows[11].Cells[9].Value = 3;
dataGridView1.Rows[11].Cells[10].Value = 4;
dataGridView1.Rows[11].Cells[11].Value = 1;
dataGridView1.Rows[11].Cells[12].Value = 1;
dataGridView1.Rows[11].Cells[12].Style.BackColor = Color.CornflowerBlue;
dataGridView1.Rows[11].Cells[13].Value = 0.250;
dataGridView1.Rows[12].Cells[1].Value = 3;
dataGridView1.Rows[12].Cells[2].Value = 4;
dataGridView1.Rows[12].Cells[3].Value = 3;
dataGridView1.Rows[12].Cells[4].Value = 3;
dataGridView1.Rows[12].Cells[5].Value = 4;
```

```

        dataGridView1.Rows[12].Cells[6].Value = 4;
        dataGridView1.Rows[12].Cells[7].Value = 3;
        dataGridView1.Rows[12].Cells[8].Value = 4;
        dataGridView1.Rows[12].Cells[9].Value = 5;
        dataGridView1.Rows[12].Cells[10].Value = 4;
        dataGridView1.Rows[12].Cells[11].Value = 4;
        dataGridView1.Rows[12].Cells[12].Value = 4;
        dataGridView1.Rows[12].Cells[13].Value = 1;
        dataGridView1.Rows[12].Cells[13].Style.BackColor = Color.CornflowerBlue;
    }
    private void button25_Click_1(object sender, EventArgs e)
    {
        button9.Enabled = true;
        this.Width = 1073;
        foreach (string name in comboBox1.Items)
        {
            listBox3.Items.Add(name);
            alternatifler.Add(name);
        }
    }
}

```

```

using SON.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SON
{
    public partial class ChildForm : Form
    {
        TOPSISEntities TOPSISEntities = new TOPSISEntities();
        public ChildForm()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            SubForm subForm = new SubForm();
            subForm.Show();
            this.Close();
        }
        private void button1_Click(object sender, EventArgs e)

```

```

        {
    try
        {
            stringtc = TB_TC.Text;
            stringadsoyad = TB_AdSoyad.Text;
            DateTimedogtar = DT_DogTar.Value;
            bool cinsiyet = RB_Erkek.Checked == true ? true : false;
            bool durum = RB_Buluntu.Checked == true ? true : false;
            bool uygun = RB_Uygun.Checked == true ? true : false;
            if (!String.IsNullOrEmpty(tc) && !String.IsNullOrEmpty(adsoyad))
            {
                ChildTablechild = newChildTable()
                {
                    DogumTarihi = dogtar,
                    Cinsiyet = cinsiyet,
                    Durum = durum,
                    Uygun = uygun,
                    AdSoyad = adsoyad,
                    TC = tc
                };

                TOPSISEntities.ChildTable.Add(child);
                TOPSISEntities.SaveChanges();
                MessageBox.Show("Başarıyla kaydedildi.");
            }
            else
            {
                MessageBox.Show("Lütfen tüm alanları doldurunuz.!");
            }
        }
        catch (Exceptionex)
        {
            MessageBox.Show("Beklenmeyen durum oluştu. Ex : "+ ex.Message);
        }
        List<ChildTable>parentList = TOPSISEntities.ChildTable.ToList();
        dataGridView1.DataSource = parentList;
        dataGridView1.Refresh();
    }
    privatevoidChildForm_Load(objectsender, EventArgs e)
    {
        List<ChildTable>parentList = TOPSISEntities.ChildTable.ToList();
        dataGridView1.DataSource = parentList;
        dataGridView1.Refresh();
    }
    privatevoid button3_Click(objectsender, EventArgs e)
    {
        intId = Convert.ToInt16(dataGridView1.CurrentRow.Cells[0].Value);
        //Yine benzer şekilde ilgili satırı çekiyoruz
        varrowdelete = TOPSISEntities.ChildTable.Where(w => w.ID == Id).FirstOrDefault();
        //Model örneği üzerinden Makale tablomuza erişip Remove() methotunu kullanıyoruz.
        TOPSISEntities.ChildTable.Remove(rowdelete);
    }

```

```

        TOPSISEntities.SaveChanges(); // ile değişiklikleri kaydediyoruz. ve işlemimiz sona
eriyor. }
        MessageBox.Show("Kayıt başarıyla silindi.");
        List<ChildTable>parentList = TOPSISEntities.ChildTable.ToList();
            dataGridView1.DataSource = parentList;
            dataGridView1.Refresh();
        }
    }
}

```

```

using SON.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SON
{
    public partial class NewParent : Form
    {
        TOPSISEntities TOPSISEntities = new TOPSISEntities();
        public NewParent()
        {
            InitializeComponent();

            List<Parent>parentList = TOPSISEntities.Parent.ToList();
            dataGridView1.DataSource = parentList;
        }
        public short ID { get; private set; }
        private void button1_Click(object sender, EventArgs e)
        {
            Parent parent = new Parent()
            {
                FBirthdate = dt_FBirthdate.Value,
                FBirthdateCity = txt_FBirthdateCity.Text,
                FName = txt_FName.Text,
                FSurname = txt_FSurname.Text,
                FTCKN = txt_fTCKNO.Text,
                MarriedDate = dt_MarriedDate.Value,
                MBirthdate = dt_MBirthdate.Value,
                MBirthdateCity = txt_MBirthdateCity.Text,
                MName = txt_MName.Text,
                MSurname = txt_MSurname.Text,
                MTCKN = txt_MTCKN.Text
            };
        }
    }
}

```

```

varresult = TOPSISEntities.Parent.Add(parent);
TOPSISEntities.SaveChanges();

ParentCriteriaPuanparentCriteriaPuan = newParentCriteriaPuan()
{
    AileninEkonomikDurumu = Convert.ToDouble(txt_1.Text),
    BeklemeSuresi = Convert.ToDouble(txt_2.Text),
    EbeveynlikKapasitesi = Convert.ToDouble(txt_3.Text),
    AileninEgitimDurumu = Convert.ToDouble(txt_4.Text),
    AileninIsDurumu = Convert.ToDouble(txt_5.Text),
    AileninSaglikDurumu = Convert.ToDouble(txt_6.Text),
    KonutunFizikiSartlari = Convert.ToDouble(txt_7.Text),
    AileninSosyalYasantisi = Convert.ToDouble(txt_8.Text),
    AileninDigerInsanlarlalletisimi = Convert.ToDouble(txt_9.Text),
    CocukluYasamaDairBeklentiler = Convert.ToDouble(txt_10.Text),
    AileBireylerininKisilikOzellikleri = Convert.ToDouble(txt_11.Text),
    EvlilikYasantisi = Convert.ToDouble(txt_12.Text),
    SosyalYasamcınınGenelKanaati = Convert.ToDouble(txt_13.Text)
};
parentCriteriaPuan.ParentID = result.ID;
TOPSISEntities.ParentCriteriaPuan.Add(parentCriteriaPuan);
TOPSISEntities.SaveChanges();

MessageBox.Show("Başarıyla Eklendi.");

List<Parent>parentList = TOPSISEntities.Parent.ToList();
    dataGridView1.DataSource = parentList;
    dataGridView1.Refresh();
}
privatevoid button2_Click(objectsender, EventArgs e)
{
    List<Parent>parentList = TOPSISEntities.Parent.ToList();
    dataGridView1.DataSource = parentList;
    dataGridView1.Refresh();
}

privatevoid button3_Click(objectsender, EventArgs e)
{
    SubFormsubForm = newSubForm();
    subForm.Show();
    this.Close();
}

privatevoid button4_Click(objectsender, EventArgs e)
{
    intId = Convert.ToInt16(dataGridView1.CurrentRow.Cells[0].Value);
    //Yine benzer şekilde ilgili satırı çekiyoruz
    varrowdelete = TOPSISEntities.Parent.Where(w => w.ID == Id).FirstOrDefault();
    //Model örneği üzerinden Makale tablomuza erişip Remove() methotunu kullanıyoruz.
    TOPSISEntities.Parent.Remove(rowdelete);
}

```

```
TOPSISEntities.SaveChanges(); // ile değişiklikleri kaydediyoruz. ve işlemimiz sona  
eriyor. }
```

```
MessageBox.Show("Kayıt başarıyla silindi.");
```

```
List<Parent>parentList = TOPSISEntities.Parent.ToList();  
    dataGridView1.DataSource = parentList;  
    dataGridView1.Refresh();  
    }  
    }  
}
```

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace SON  
{  
    public partial class LoginForm : Form  
    {  
        public LoginForm()  
        {  
            InitializeComponent();  
        }  
        private void button1_Click(object sender, EventArgs e)  
        {  
            if (textBox1.Text == "TOPSIS" && textBox2.Text == "123")  
            {  
                SubForm frm1 = new SubForm();  
                frm1.Show();  
                this.Hide();  
            }  
            else  
            {  
                MessageBox.Show("Kullanıcı Adı veya Şifre yanlış girilmiştir. Lütfen kontrol ediniz.!",  
"Giriş Hatası", MessageBoxButtons.OK, MessageBoxIcon.Error);  
            }  
        }  
    }  
}
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : AKÇİN Esra
Doğum tarihi ve yeri : 20.06.1992 / BURSA
e-mail : esraakcin1992@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Dumlupınar Üniversitesi	-
Lisans	Osmangazi Üniversitesi	2014
Lise	Hürriyet Anadolu Lisesi	2010

İş Deneyimi

Yıl	Yer	Görev
2017-2019	Anka Seramik	Üretim Planlama Mühendisi
2017-2017	Bemis Teknik Elektrik	Üretim Planlama Mühendisi
2017-2017	Bursa Poliüretan	Kalite Mühendisi

Yabancı Dil

İngilizce Orta