
SINAV ÇİZELGELEME PROBLEMLERİNDE HOMOJEN SINAV DAĞILIMININ OLUŞTURULMASI İÇİN KÜMELEME VE HEDEF PROGRAMLAMA TEMELLİ BİR YAKLAŞIM

*Fatih ÇAVDUR**
*Sema DEĞİRMEN**
*Merve KÖSE KÜÇÜK**

Alınma: 26.10.2017; düzeltme: 25.11.2017; kabul: 05.03.2018

Öz: Bu çalışma, dengeli bir sınav programı oluşturmak için kümeleme ve hedef programlama tabanlı bir yaklaşım sunmaktadır. Çalışmada, kişisel iş yükü açısından öğrencileri ve öğretim elemanlarını belirli bir düzeyde memnun edecek, dengeli bir sınav programı oluşturmak amaçlanmaktadır. Bu kapsamda öncelikle, derslerin kredisi, başarı oranı ve türü olmak üzere üç parametre kullanılarak sınav kritiklik seviyelerinin belirlenmesi için Ward yöntemi ve k -ortalamalar kümeleme algoritması önerilmektedir. Daha sonra, belirlenen kritiklik seviyeleri ve diğer problem kısıtları dikkate alınarak bir hedef programlama modeli ile sınav çizelgesi oluşturulmaktadır. Önerilen yaklaşım, bir gerçek hayat problemi üzerinde örneklendirilmiştir. Yaklaşım sonucu oluşturulan çizelge, gerçek hayatta oluşturulan çizelge ile karşılaştırıldığında, sınavların kritiklik seviyelerini de dikkate alan dengeli bir sınav çizelgesinin oluşturulduğu görülmektedir. Buna ek olarak, önerilen yaklaşımın daha büyük boyutlu gerçek hayat problemlerinde de kullanılma potansiyeli bulunmaktadır.

Anahtar Kelimeler: Sınav çizelgeleme, kümeleme, k -ortalamalar, Ward yöntemi, hedef programlama

A Clustering and Goal Programming-Based Approach for Homogeneous Exam Distribution in Exam Scheduling Problems

Abstract: This study presents a clustering and binary goal programming-based approach to create a balanced-exam schedule. The aim of the study is to create a balanced-exam schedule in terms of person workloads to achieve a certain level of satisfaction for students and professors. We first propose Ward's method and k -means clustering algorithm for criticality level identification using credits, success ratios and types of classes. A goal programming model is then used to create an exam schedule using the criticality levels and other problem constraints. Proposed approach is illustrated with a real-life case study. We compare the exam schedule produced by the proposed approach with the real-life exam schedule. It is noted that a balanced-exam schedule is produced by our approach where the criticality levels of exams are considered. In addition, we also note that the proposed approach has a potential to be used for larger real-life exam scheduling problems.

Keywords: Exam scheduling, clustering, k -means, Ward's method, goal programming

1. GİRİŞ

Eğitim amaçlı çizelgeleme, tüm akademik kurumlarda gerçekleştirilen önemli faaliyetlerden biri olup yıllar içinde çok fazla araştırmanın yapıldığı bir konudur. Çizelgeleme problemleri,

* Uludağ Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, Nilüfer, 16059 Bursa
İletişim Yazarı: Sema DEĞİRMEN (degirmensema@gmail.com)

zaman ve kaynakların doğru kullanımına ek olarak, zorunlu ve isteğe bağlı kısıtlama kümelerinin de dikkate alınmasını gerektiren problemlerdir. Problemin boyutuna bağlı olarak, çözüm bulmak için gereken hesaplama miktarı önemli ölçüde arttığından, bu tür problemler NP-zor (NP-hard) problem sınıfına girmektedir (Babaei ve diğ., 2015).

Sınav çizelgeleme problemleri, sadece genel problem kısıtlamalarını değil, aynı zamanda öğrenciler, öğretim elemanları, gözetmenler gibi çeşitli grupların tercihlerini veya taleplerini içermektedir. Bazı durumlarda, genel problem kısıtlarına ek olarak bu grupların tercihlerini veya taleplerini karşılamak, sınav programından sorumlu karar vericiler için çok daha zor olabilmektedir. Sonuç olarak, her grup için mükemmel bir çözüm bulmak genellikle mümkün olmadığından, sınav programından sorumlu karar vericiler, dengeli bir sınav programı oluşturmak için tecrübeye dayalı-pratik bir çözüm yaklaşımı izlemektedirler.

Bu çalışmada geliştirilen yaklaşım, Uludağ Üniversitesi-Mühendislik Fakültesi-Endüstri Mühendisliği Bölümü'ne ait bir sınav çizelgesi oluşturmak için kullanılmıştır. Söz konusu akademik birimde, ara sınavlar, yarıyıl sonu sınavları ve bütünleme sınavları olmak üzere her yarıyılıda üç sınav dönemi bulunmaktadır. Bu çalışma kapsamında geliştirilen yaklaşım iki haftadan (10 gün, 4 zaman aralığı/gün) oluşan bir yarıyıl sonu sınav çizelgesinin oluşturulması için kullanılmıştır. Her sınav döneminde, normal bir öğrencinin (alt dönemlerden başarısız olduğu dersi olmayan veya üst dönemlerdeki derslere kayıt yaptırmamış olan öğrenciler) bir dönemde sorumlu olduğu yaklaşık olarak 9-10 dersi bulunmaktadır.

Çavdur ve Kose (2016) tarafından yapılan çalışmada belirtildiği gibi sınav çizelgelerinin oluşturulmasında, öğrencilerin sınav yoğunluğunun yeterli düzeyde dikkate alınmaması söz konusu olabilmektedir. Öğrenciler, bazı derslerin sınavlarında başarılı olabilmek için daha fazla zaman ve çaba harcadıklarından, bu derslerin sınavlarını kritik sınavlar olarak tanımlayabilmektedirler. Farklı dönemlerden alınan derslerin de dikkate alınmasıyla, sınavların iş yükü önemli ölçüde farklılık gösterebilmektedir. Örneğin, bir sınav gününde, kritiklik seviyesi düşük olan iki sınav, kritiklik seviyesi yüksek olan bir sınava tercih edilebilmektedir. Bundan dolayı, çizelgeleme yapılırken sadece sınav sayısının değil, *sınav kritiklik seviyelerinin* de göz önüne alınması, öğrencilerin beklentilerine daha uygun bir sınav çizelgesi oluşturulmasına katkı sağlayabilir. Ayrıca, sınav dağılımı, öğrencilerin yanı sıra öğretim elemanları ve gözetmenler için de önemli olabilmektedir. Sonuç olarak, sınav çizelgesi oluşturmaktan sorumlu kişinin/kişilerin, herkes tarafından kabul edilebilir bir sınav programı oluşturması uzun süreler alabilmektedir.

Çavdur ve Kose (2016) çalışmalarında, söz konusu kritiklik seviyelerinin tespit edilebilmesi için bulanık mantık tabanlı bir yaklaşım önermişlerdir. Buna ek olarak, yazarlar, bulanık mantık yaklaşımı ile belirledikleri kritiklik seviyelerini, geliştirdikleri hedef programlama modeli için girdi olarak kullanıp dengeli bir sınav çizelgesi oluşturmayı amaçlamışlardır. Bu çalışmada, Çavdur ve Kose (2016) tarafından, sınav kritiklik seviyelerini belirlemek için kullanılan yaklaşıma bir alternatif olarak, kümeleme yöntemlerinin kullanılması önerilmektedir. Kümeleme yöntemleri kullanılarak kritiklik seviyeleri belirlenen sınavların ataması için de yine Çavdur ve Kose (2016) tarafından geliştirilen hedef programlama modelinden yararlanılmıştır. Sonuç olarak, söz konusu çalışmada yapıldığı gibi bu çalışmada da sınav çizelgeleme probleminin çözümü için iki aşamadan oluşan bir yaklaşım önerilmektedir.

Yaklaşımın ilk aşamasında, sınav kritiklik seviyelerinin belirlenmesi için Ward yöntemi ve k -ortalama kümeleme algoritması kullanılmıştır. Dersin kredisi, başarı oranı ve türü (zorunlu/seçmeli) dikkate alınarak, sınav kritiklik seviyeleri için çok düşük, düşük, orta, yüksek ve çok yüksek olmak üzere beş seviyenin olduğu varsayılmış ve sınavların hangi seviyeye ait olduğu kümeleme yöntemleri kullanılarak belirlenmiştir. İkinci aşamada, elde edilen sınav kritiklik seviyeleri de dikkate alınarak, hedef programlama modeli kullanılıp dengeli bir sınav çizelgesinin oluşturulması amaçlanmıştır.

Makalenin sonraki bölümünde, literatürdeki ilgili çalışmalar özetlenmektedir. Üçüncü bölümde, kullanılan kümeleme yöntemleri ve hedef programlama ile ilgili detaylar

açıklanmaktadır. Dördüncü bölümde, yapılan çalışmada elde edilen çizelge, uzman kişi tarafından oluşturulan çizelge ile çeşitli performans kriterleri açısından karşılaştırılmaktadır. Son olarak, çalışma ile ilgili genel bir tartışma beşinci bölümde verilmektedir.

2. LİTERATÜR TARAMASI

Çizelgeleme problemleri, yöneylem araştırmasında çok çalışılan problemler arasındadır (Johnes, 2015). Çizelgeleme problemlerinde, zaman ve kaynakların olaylara atanması gerekirken, zorunlu ve isteğe bağlı kısıtların da dikkate alınması gerekmektedir (Fonseca ve diğ., 2017). Eğitim amaçlı çizelgeleme problemleri, sadece sınav ve ders çizelgeleme ile ilgili olmayıp; bu tür problemlerde, müfredat, sınıf-öğretmen vb. atamalarının yapılması da söz konusu olmaktadır. Örneğin, Ünal ve Uysal (2014), dönem başı akademik yükün dengelenmesi ve ön koşulların aynı anda yerine getirilmesi için ilgili dersleri, mümkün olan en yakın dönemlere atamayı hedefleyerek, müfredat dengeleme (Relevance Based Curriculum Balancing: RBCB) modelleri sunmuşlardır. Eğitim amaçlı çizelgeleme problemlerinden olan sınıf-öğretmen atamaları için ise Veenstra ve Vis (2016) tarafından yapılan çalışma örnek olarak verilebilir. Söz konusu çalışmada, yazarlar, önceden tahmin edilemeyen kargaşalar altında, okul çizelgeleme problemini etkin bir şekilde çözmek için yöntemler tanımlamışlardır. Yapılan bu tür çalışmaların yanında ders ve sınav çizelgeleme problemleri de üzerinde çok çalışılan eğitim amaçlı çizelgeleme problemleri arasında yer almaktadır.

Ders ve sınav çizelgeleme, tüm akademik kurumlarda gerçekleşen önemli faaliyetlerden biridir. Bu konuda yapılan çok sayıda çalışma bulunmaktadır. Burke ve Petrovic (2002), Nottingham Üniversitesi Otomatik Zamanlama, Optimizasyon ve Planlama Araştırma Grubu'nda (ASAP) geliştirilen veya geliştirilmekte olan çizelgeleme problemleri yaklaşımlarını kısaca tanıtmışlardır. MirHassani ve Habibi (2013), ders çizelgeleme problemleri için çözüm yaklaşımlarını incelemişlerdir. Bir diğer çalışmada, Babaei ve diğ. (2015), son on yılda gerçekleştirilen sınav saati belirleme problemi için yeni eğilimleri ve temel araştırma başarılarını vurgulamışlardır.

Ders ve sınav çizelgeleme ile ilgili olan genel çalışmaların yanında literatürde, problemi farklı açılardan dikkate alan çok sayıda spesifik çalışmalar da bulunmaktadır. Örneğin, Daskalaki ve diğ. (2004), üniversite çizelgeleme problemi için kapsamlı bir 0-1 tamsayılı programlama formülasyonu sunmuşlardır. Al-Yakoob ve Sherali (2006) tarafından yapılan çalışma ise ders çizelgeleme probleminin çözümü için geliştirilmiş olan geniş kapsamlı modeller arasındaki en iyi örneklerden birisi olarak verilebilir. Yazarlar tarafından, genel akademik ders çizelgeleme kısıtlarına ek olarak, öğretim elemanlarının memnuniyetsizliği ve cinsiyete dayalı politikalar da dahil olmak üzere çeşitli kısıtlar altında matematiksel programlama modelleri geliştirilmiştir. Al-Yakoob ve Sherali (2015), başka bir çalışmalarında ise problemin çözümüne odaklanarak, lise çizelgeleme problemi için bir sütun oluşturma (column generation) yaklaşımı kullanmışlardır. Bir diğer çalışmada ise Vermuyten ve diğ. (2016), farklı katlara dağılmış pek çok odaya sahip üniversite binalarında, birbirini takip eden iki ders arasında ortaya çıkan öğrenci akışlarını en aza indirmeyi hedefleyen ders çizelgesi oluşturmak için bir tamsayılı programlama yaklaşımı sunmuşlardır. Muklason ve diğ. (2017) tarafından yapılan çalışmada, çizelgeleme problemindeki öğrencilere olan dikkat eksikliğine odaklanılmıştır.

Çizelgeleme problemlerinin karmaşıklık düzeylerinin yüksek olması nedeniyle, bu tür problemlerin çözümü için sezgisel algoritmalar da kullanılmaktadır. Burke ve diğ. (2006), otomatik üniversite ders ve sınav çizelgelemesi için vaka tabanlı bir sezgisel seçim yaklaşımı önerirken; Chu ve diğ. (2006) yaptıkları çalışmada sınav çizelgeleme için parçacık sürü optimizasyonu algoritmasını kullanmışlardır. Eley (2006) ise bir öğrencinin kayıtlı olduğu iki sınavın aynı dönemde çizelgenmesini önlemek veya bunu minimuma indirmek amacıyla, karınca kolonisi algoritmasını kullanmıştır. Pillay ve Banzhaf (2010) tarafından yapılan çalışmada da sınav çizelgeleme problemine çözüm üretmek için genetik algoritma

kullanılmıştır. Badoni ve diğ. (2014) yaptıkları çalışmada, üniversite ders çizelgeleme problemini çözmek için genetik algoritmayı yerel arama ile birleştiren bir hibrid algoritma kullanmışlardır.

Kullanılan yaklaşımlara ek olarak, çizelgeleme problemleri için söz konusu yaklaşımların entegre edildiği yazılımların geliştirilmesine yönelik çalışmalar da yapılmıştır. Bu çalışmalara örnek olarak, Carter ve diğ. (1994) tarafından geliştirilen EXAMINE adı verilen esnek bir sınav çizelgeleme yazılımı ve Pongcharoen ve diğ. (2008) tarafından yapılan çalışmada, üniversite ders çizelgelemesi için geliştirilen Stokastik Optimizasyon Zaman Çizelgesi Aracı (SOTT) verilebilir.

Eğitim kurumlarındaki öğrenci sayısının artması, büyük boyutlu verilerin depolanması ve derlenmesi sorunlarını da beraberinde getirdiğinden, yukarıda bahsedilenler de dahil olmak üzere eğitim alanındaki problemler daha karmaşık hale gelmektedir. Veri miktarındaki bu artış, büyük boyutlu verilerin ayrıştırılabilmesi için daha gelişmiş algoritmaların kullanımının gerektiğini göstermektedir. Bu durum, eğitimde veri madenciliği tekniklerinin kullanıldığı çalışmaların yapılmasına yol açmıştır. Örneğin, Romero ve Ventura (2007) tarafından yapılan çalışmada, veri madenciliğinin geleneksel eğitim sistemlerine, belirli web-tabanlı derslere, öğrenme içerikli yönetim sistemlerine, uyarlanabilir ve akıllı web-tabanlı eğitim sistemlerine uygulanmasının 1995-2005 yılları arasındaki gelişimi incelenmiştir. Dutt ve diğ. (2017) tarafından yapılan çalışmada ise 1983-2016 yılları arasındaki eğitimsel veri madenciliği tekniklerinden kümeleme algoritmasının uygulanabilirliği ve kullanılabilirliği üzerine sistematik bir literatür taraması sunulmaktadır.

Eğitim amaçlı problemlerin ele alındığı çalışmalarda, veri madenciliği tekniklerinin ders ve sınav çizelgeleme problemlerine uygulandığı çalışmalar da literatürde yer almaktadır. Veri madenciliği tekniklerinden olan kümeleme yöntemleri, eğitim amaçlı çizelgeleme problemlerinde spesifik olarak kullanılan yöntemler arasında bulunmaktadır. Burada, Shatnawi ve diğ. (2012) tarafından yapılan çalışma örnek olarak verilebilir. Yazarlar, öğrenci kümelerini (öğrencilerin bir sonraki dönemde kayıt olacakları dersler kümesi) dinamik olarak oluşturabilen veri madenciliği tekniklerini kullanarak, kümeleme yöntemine dayanan ders çizelgesi oluşturmak için yeni bir teknik önermişlerdir. Ilic ve diğ. (2015) ise ders ve sınav çizelgeleme problemlerinde yer alan kısıtları tanımlayarak, çizelgeleme optimizasyonu için yapay sinir ağları ve kümeleme tabanlı bir yaklaşım önermişlerdir.

Veri madenciliği alanında yapılan çalışmalarda en fazla kullanılan yöntemlerden biri de k -ortalamlar kümeleme algoritmasıdır (Wu ve diğ., 2008). Literatürde, k -ortalamlar algoritması ve bu algoritmanın daha etkili çalışabilmesi ile ilgili yapılan birçok çalışma bulunmaktadır (MacQueen, 1967; Hartigan ve Wong, 1979; Alsabti ve diğ., 1997; Jain, 2010; Kaur ve diğ., 2012). k -ortalamlar ile ilgili yapılan genel çalışmaların yanında, söz konusu algoritmanın ders ve sınav çizelgeleme problemlerinde de kullanıldığı çalışmalar bulunmaktadır. Örneğin, Kaur ve Kaur (2014) yaptıkları çalışmada, değiştirilmiş k -ortalamlar kümeleme algoritması (modified k -means algorithm) ile kural tabanlı sınıflandırıcı teknikleri (rule based classifier techniques) kullanarak otomatik ders çizelgeleme sistemi için bir yaklaşım önermektedirler. Babaei ve diğ. (2016), üniversitenin farklı bölümleri arasındaki ortak öğretim elemanlarının memnuniyetini arttırmayı ve bölümlerdeki kaynak kayıplarını en aza indirmeyi amaçlamışlardır. Bu amaçla, ortak öğretim elemanlarını kümelemek için kümeleme ve çaprazlama aracı olmak üzere iki araç kullanmışlardır. Yazarların, bu çalışmada kullandıkları kümeleme algoritması k -ortalamlar iken; başka bir çalışmalarında (Babaei ve diğ., 2017) ise yine aynı amaç doğrultusunda kümeleme algoritması olarak k -ortalamların bir türevi olan bulanık c -ortalamlar (fuzzy c -means) algoritmasını kullanmışlardır.

Bu çalışmada ise Cavdur ve Kose (2016) tarafından yapılan çalışmadan yola çıkılarak sınav çizelgeleme probleminin çözümü için iki aşamalı bir yaklaşım önerilmektedir. Cavdur ve Kose (2016) yaptıkları çalışmada, sınav kritiklik seviyelerini dikkate alan dengeli bir sınav çizelgeleme modeli sunmuşlardır. Yazarlar tarafından geliştirilen çözüm yaklaşımının ilk

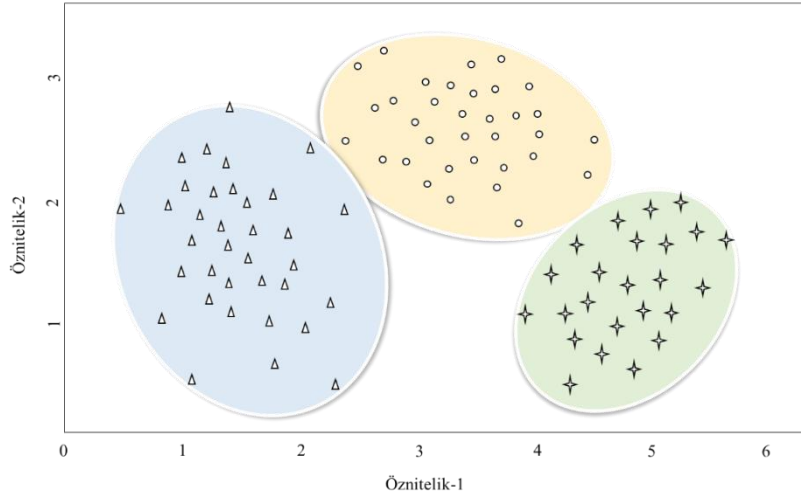
aşamasında, sınav kritiklik seviyelerinin belirlenmesi için bulanık mantık kullanılmış, ikinci aşamasında ise geliştirilen hedef programlama modeli ile dengeli bir sınav çizelgesi oluşturulması amaçlanmıştır. Bu çalışmada, Cavdur ve Kose (2016) tarafından yapılan çalışmadan farklı olarak, hedef programlama modelinde girdi olarak kullanılan sınav kritiklik seviyelerinin belirlenmesi için iki farklı kümeleme yöntemi önerilmektedir.

3. METODOLOJİ

3.1. Kümeleme Çalışması

Sınav çizelgesi oluşturulurken en önemli adımlardan biri, sınavların kritiklik seviyelerine göre ilgili zaman aralığına atanmasıdır. Kritiklik seviyesi yüksek olan sınavların, aynı gün içinde çizelgelenmemesi ya da en azından bunun minimum düzeye indirilmesi hedeflenmektedir. Bundan dolayı, derslere ait bilgileri içeren veri kümesinden yola çıkılarak, sınav kritiklik seviyelerini belirlemek için kümeleme çalışması yapılmıştır.

Kümeleme; makine öğrenimi, veri madenciliği, model tanıma, görüntü analizi ve biyobilişim gibi istatistiksel veri analizi için pek çok alanda kullanılan yaygın bir tekniktir (Madhulatha, 2012). Kümeleme, gözetimsiz öğrenme problemi sınıfına girip, etiketsiz veri koleksiyonu içinde bir yapı bulma olarak düşünülebilir. Başka bir deyişle, nesnelere, benzer özelliklere sahip gruplar halinde organize etme işlemi olarak tanımlanabilmektedir. Bu nedenle, bir küme, kendi içinde benzer olmaya çalışan ancak diğer kümelerle ise mümkün olduğunca benzer olmayan nesnelere oluşan bir koleksiyondur. Burada, Şekil 1'deki grafik, bir kümeleme örneği olarak verilebilir.



Şekil 1:
Kümeleme örneği

Kümeleme, (i) hiyerarşik (hierarchical), (ii) bölünmeli (partitioning), (iii) yoğunluk-tabanlı (density-based), (iv) ızgara-tabanlı (grid-based) ve (v) model-tabanlı (model-based) yöntemler olmak üzere beş gruba ayrılmakta olup (Bijuraj, 2013), bu çalışmada hiyerarşik ve bölünmeli kümeleme yöntemleri dikkate alınmıştır.

Hiyerarşik yöntem kategorisindekiler, sezgisel bölme veya birleştirme tekniklerini kullanarak genellikle bir küme ağacı (veya dendrogram) üretirler (Hamerly, 2003). Üretilen dendrogram ile kümelerin görselleştirilmesi sağlanarak, küme sayısına karar verici tarafından sezgisel olarak karar verilir. Hiyerarşik kümeleme yönteminde, küme sayısının önceden belirlenmemesi ve başlangıç koşullarından bağımsız olması bu yöntemin avantajları arasında yer alırken (Frigui ve Krishnapuram, 1999); hesaplama açısından karmaşık olmasından dolayı

çok büyük veri setleri için uygun olmaması dezavantajları arasında yer almaktadır. Bölünmeli kümeleme yönteminde ise veri seti önceden belirlenen bir küme sayısına bölünerek kümeler oluşturulur. Bu yöntemdeki algoritmalar, belirli kriterleri (örneğin, karesel hata fonksiyonu) minimize etmeye çalıştığından, optimizasyon problemleri olarak ele alınabilirler (Omran ve diğ., 2007). Bölünmeli kümeleme yöntemi, hiyerarşik kümeleme yöntemine kıyasla küme sayısının önceden belirlendiği durumlarda, özellikle büyük veri setleri için daha uygun olabilmektedir.

Bu çalışmada, hiyerarşik kümeleme yöntemlerinden olan “Ward yöntemi” ve bölünmeli kümeleme yöntemlerinden olan “ k -ortalamalar algoritması” sınav kritiklik seviyelerinin belirlenmesi amacıyla kullanılmış olup, dördüncü bölümde yer alan sayısal örnek üzerinde kümeleme sonuçları karşılaştırılmıştır.

3.1.1. Ward Yöntemi

Hiyerarşik kümeleme yöntemlerinde, gruplayıcı (agglomerative) ve bölücü (divisive) kümeleme olarak iki yaklaşım bulunmaktadır. Ward yönteminin de dahil olduğu gruplayıcı kümeleme yöntemi için temel adımlar aşağıdaki gibi özetlenebilir:

Adım 1: Her bir nesne bir küme olarak alınıp işleme başlanır.

Adım 2: Sadece bir küme kalana kadar;

- En yakın küme çiftleri bulunur.
- Bulunan en yakın küme çiftleri birleştirilir.

Adım 3: Küme birleşmelerinin ağaç yapısı oluşturulur.

Ward (1963) tarafından ortaya koyulan ve en küçük varyans yöntemi olarak da bilinen Ward yönteminde, iki kümenin birleştirilmesi için kullanılan yakınlık değeri, hatanın karesinin toplamı kriterine dayanır. Yöntemin her aşamasında, iki alt küme bir sonraki seviyeyi oluşturmak için birleştirilir. Burada, her nesnenin başlangıçta bir küme olarak belirlenmesinden dolayı ilk aşamada hata kareleri toplamı sıfır olmaktadır. Ward yönteminin her aşamasındaki amaç, küme içi hatanın karelerinin toplamındaki artışı minimize etmektir. Bu artış, birleştirilmiş kümelerin merkezleri arasındaki karesel Öklid uzaklığı ile orantılı olup, küme içi hataların karelerinin toplamı (E), aşağıdaki denklemler yardımıyla hesaplanmaktadır (Everitt ve diğ., 2011):

$$E = \sum_{m=1}^g E_m \quad (1)$$

$$E_m = \sum_{l=1}^{n_m} \sum_{k=1}^{p_k} (x_{mlk} - \bar{x}_{mk})^2 \quad (2)$$

Denklem (2)'de yer alan \bar{x}_{mk} , k . değişken (öznitelik) için m . kümenin ortalamasını ifade etmekte olup, $\bar{x}_{mk} = (1/n_m) \sum_{l=1}^{n_m} x_{mlk}$ eşitliği kullanılarak hesaplanmaktadır. Burada x_{mlk} , m . kümedeki ($m = 1, \dots, g$) l . nesnenin ($l = 1, \dots, n_m$) k . değişkeninin ($k = 1, \dots, p_k$) değerine karşılık gelmektedir.

Hiyerarşik kümeleme yönteminde, küme sayısı ile ilgili bir varsayımın olmaması kümeleme sonucu oluşturulan dendrogramda istenilen sayıda küme elde edilebilmesini sağlamaktadır. Bu çalışmada, uygulama bölümünde yer alan örnekte, Ward yöntemi ile k -ortalamalar kümeleme algoritması sonucunda elde edilen kümelerin karşılaştırılabilmesi için sınav kritiklik seviyeleri

her iki yöntemde de beş grup olarak (Cavdur ve Kose (2016) tarafından yapılan çalışmada olduğu gibi) belirlenmiştir.

3.1.2. *k*-Ortalamalar Algoritması

k-ortalamalar, kümeleme problemini çözen, gözetimsiz öğrenme algoritmalarından biri olup (MacQueen, 1967), bölünmeli kümeleme yöntemleri içinde yaygın kullanılan bir algoritmadır (Kaur ve diğ., 2012). *k*-ortalamalar ile Ward yöntemi hatanın karesi temelli yöntemler olduğundan *k*-ortalamalar, Ward yönteminin hiyerarşik versiyonu olarak da düşünülebilmektedir. Algoritmanın uygulama kolaylığı ile birlikte zaman karmaşıklığının da az olması başlıca avantajları arasında yer almaktadır (Turi, 2001). Algoritma, bir veri kümesini, belirli bir sayıda küme (*k* adet küme) üzerinden sınıflandırmayı sağlamaktadır. Temel fikir, her küme için bir tane olmak üzere, *k* adet merkez tanımlamaktır. Bu merkezler, farklı konuma göre farklı sonuçlara neden olabileceğinden, merkezleri birbirinden mümkün olduğunca uzak tutmak gerekmektedir. Bir sonraki adımda, veri kümesinden alınan her bir nokta, kendine en yakın merkez ile ilişkilendirilmektedir. Hiçbir nokta dışarıda kalmayana kadar bu işlem devam etmektedir. Veri kümesindeki her nokta, bir kümeye atandıktan sonra yeni küme merkezleri hesaplanmaktadır. Yeni küme merkezlerine göre, veri kümesi noktaları ile en yakın küme merkezi arasında yeni bir bağ kurulmaktadır. Bu süreç tekrarlanarak bir döngü oluşturulmaktadır. Bu döngü, küme merkezlerinin yeri (kümelerin ağırlık merkezleri) daha fazla değişmeye kadar devam etmektedir. Sonuçta algoritma, küme içi hataların karesinin toplamını ifade eden ve Denklem (3)'te verilen amaç fonksiyonunu (*J*) minimize etmeyi hedeflemektedir.

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_{ij} - c_j\|^2 \quad (3)$$

Denklem (3)'te yer alan c_j , *j*. küme merkezini; x_{ij} ise veri kümesi içindeki *n* adet noktadan *j*. küme içinde yer alan *i*. noktayı temsil etmektedir. Özetle, küme içindeki uzaklıkları minimize etmeye çalışan *k*-ortalamalar algoritması aşağıdaki adımlardan oluşmaktadır:

Adım 1: Veri kümesi *k* gruba (küme) ayrılır.

Adım 2: Her kümenin ortalaması (merkez nokta) hesaplanır.

Adım 3: Her nesne (nokta), en yakın merkez noktanın olduğu kümeye dahil edilir.

Adım 4: Nesnelerin kümeleneğinde değişiklik olmayana kadar Adım 2 ve 3 tekrarlanır.

Algoritmanın ilk adımında yer alan veri kümesinin *k* gruba ayrılması, küme sayısının belirlenmesini ifade etmektedir. En doğru küme sayısının tahmini için farklı yaklaşımlar bulunmakla birlikte, bu çalışmada sınav kritiklik seviyeleri Cavdur ve Kose (2016) tarafından yapılan çalışmada olduğu gibi beş grup olarak belirlenmiş olup, gerçek hayat problemine ait bir uygulama da dördüncü bölümde yer almaktadır.

3.2. Hedef Programlama

Sınavların kritiklik seviyeleri belirlendikten sonra, sınav çizelgesinin oluşturulması için Cavdur ve Kose (2016) tarafından önerilen hedef programlama modeli kullanılmıştır. Çalışmada öncelikli olarak dikkate alınan unsur dengeli bir sınav çizelgesi oluşturmak olduğundan, sınıf ve gözetmen atamaları bu çalışma kapsamına dahil edilmemiştir.

Cavdur ve Kose (2016) tarafından önerilen hedef programlama modeli, tanımlanan kısıtlar altında kritiklik seviyeleri belirlenen sınavların dengeli olarak (öğrenci, öğretim elemanı, vb. tüm grupların istek ve tercihlerine dayalı) çizelgenmesini amaçlamaktadır. Modelin parametreleri; ataması yapılacak sınavlar kümesi, sınavlardan sorumlu öğretim elemanları

kümesi, öğrencilerin eğitim-öğretim dönemi (1, 2, 3 ve 4. dönem), sınavların atanacağı zaman aralıkları (saat, gün, hafta), sınavları yapılacak derslerin birbirleriyle çakışma matrisleri ve kümeleme yöntemi ile belirlenen sınavların kritiklik seviyeleri olarak tanımlanabilir.

Hedef programlama modelinde zorunlu ve esnek kısıtlar olmak üzere iki farklı kısıt türü dikkate alınmıştır. Modelde dikkate alınan zorunlu kısıtlar aşağıdaki gibi ifade edilebilir:

- Her sınav mutlaka bir zaman aralığına atanmalıdır.
- Bir güne en fazla 4 adet sınav atanabilir.
- Aynı döneme ait derslerin sınavları aynı zaman dilimine atanmamalıdır.
- İki dersi alan öğrenci sayısı birden fazla ise (çakışma mevcutsa) bu derslerin sınavları aynı zaman dilimine atanmamalıdır.

Hedef programlama modelinde dikkate alınan esnek kısıtlar ise aşağıdaki gibi özetlenebilir:

- Aynı gün içinde atanan farklı sınavlar, mümkün oldukça farklı dönemlere ait olmalıdır (1. Hedef).
- Kritiklik seviyeleri bakımından atanan sınavlar değerlendirildiğinde, haftalar arası ve günler arası fark mümkün oldukça sifıra eşit olmalıdır. Kritiklik seviyesi daha yüksek olan sınavlar tek bir haftaya/güne ya da kritiklik seviyesi daha düşük olan sınavlar tek bir haftaya/güne toplanmamalıdır (2. Hedef).
- Öğretim elemanlarının sorumlu oldukları derslere ait sınavlar mümkün oldukça haftalar arasında eşit şekilde dağıtılmalıdır (3. Hedef).

Hedef programlama modelinin amaç fonksiyonu, kısıtlarda belirtilen hedeflerin sapma değerlerinin en küçüklenmesinden oluşmaktadır. Ek 1’de verilen hedef programlama modeli ile ilgili daha detaylı bilgi için Cavdur ve Kose (2016) tarafından yapılan çalışma incelenebilir.

4. UYGULAMA VE SONUÇLAR

Metodoloji bölümünde detayları açıklanan yaklaşım, Uludağ Üniversitesi-Mühendislik Fakültesi-Endüstri Mühendisliği Bölümü’ne ait bir yarıyıl sonu sınav çizelgesi oluşturmak için kullanılmıştır. Bu çalışmada, Cavdur ve Kose (2016) tarafından da önerildiği gibi sınavların kritiklik seviyeleri; ilgili dersin kredisi, başarı oranı ve türüne (zorunlu/seçmeli) bağlı olarak belirlenmektedir. Derslerin kredisi 1,5-4,0 arasında değişirken; ders başarı oranlarını bir önceki yıla ait, dersten başarılı olan öğrenci sayısının toplam öğrenci sayısına olan oranları oluşturmaktadır. Ders türü ise sadece ikili değerler alan bir parametre olup, derslerin zorunlu (0) veya seçmeli (1) olmalarını ifade etmektedir. Verilen bilgiler ışığında, bir yarıyıla ait (1, 2, 3 ve 4 olmak üzere 4 sınıf) 38 adet dersin veri kümesi oluşturulmuştur. Oluşturulan veri kümesine ait beş dersten oluşan bir örnek, Tablo 1’de verildiği gibidir.

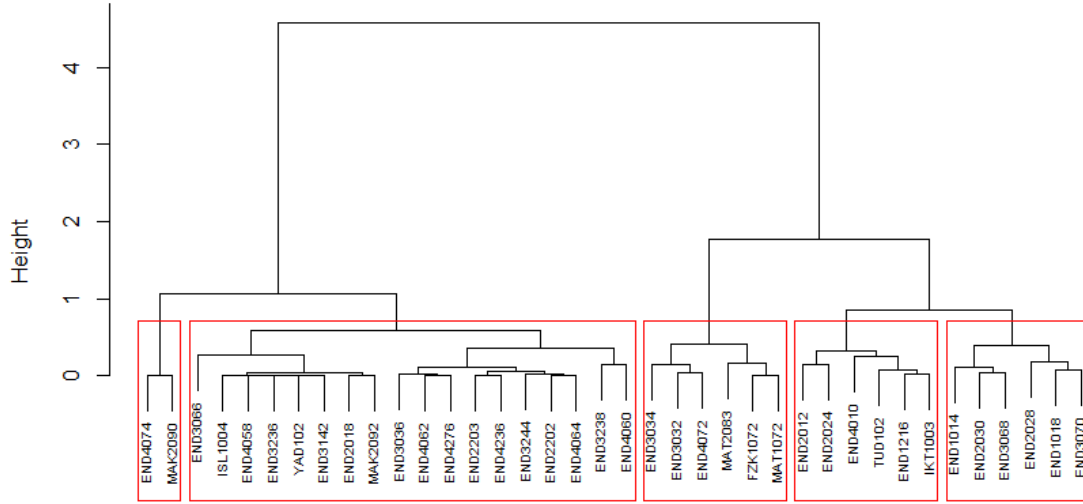
Tablo 1. Ders veri kümesi

Ders Kodu	Kredi	Başarı Oranı	Tür (Z/S)
YAD102	2,00	1,00	1
FZK1072	4,00	0,44	0
END1018	2,50	0,83	0
END2030	3,00	0,77	0
END4010	1,50	0,91	0

Derslere atanan özniteliklere (ders kredisi, başarı oranı ve türü) bağlı olarak oluşturulan veri seti kullanılarak sınav kritiklik seviyelerini belirlemek için “Metodoloji” bölümünde yer alan

kümeleme yöntemleri uygulanmıştır. Hiyerarşik kümeleme yöntemlerinden olan Ward yöntemi, R ortamında tanımlı olan hiyerarşik kümeleme paketi (*hclust, method="ward.D2"*) kullanılarak çalıştırılmıştır. Ward yöntemi ile elde edilen dendrogram Şekil 2’de gösterilmektedir.

Bu çalışmada, Cavdur ve Kose (2016) tarafından yapılan çalışmada olduğu gibi sınavların kritiklik seviyeleri çok düşük (1) ve çok yüksek (5) arasında değişen beş kümeden oluşmaktadır. Ward yöntemi ile elde edilen kümeleme sonucunun, *k*-ortalamlar ile elde edilen kümeleme sonucu ile karşılaştırılabilmesi için dendrogram kullanılarak beş küme oluşacak şekilde gruplama yapılmıştır. Gruplama sonucu elde edilen kümeler, Şekil 2’de dikdörtgenler içine alınarak gösterilmiştir. Ward yöntemi ile oluşan kümelere göre tanımlanan sınav kritiklik seviyeleri, Ek 2’de yer alan tabloda verilmiştir.



Şekil 2:
Ward yöntemi ile elde edilen dendrogram

Sınav kritiklik seviyelerini belirlemek için bölümlü kümeleme yöntemlerinden olan *k*-ortalamlar algoritması da, R ortamında tanımlı olan paket (*kmeans*) kullanılarak uygulanmıştır. Bu pakette algoritmanın adımları, Hartigan ve Wong (1979) tarafından yapılan çalışma baz alınarak tanımlanmış olup, durma kriteri olarak maksimum iterasyon sayısı (10) kullanılmaktadır.

k-ortalamlar ile yapılan kümeleme çalışmasında, Ward yönteminde olduğu gibi veri seti beş kümeye ayrılmıştır. Daha sonra, algoritma birkaç kez çalıştırılarak en iyi sonucu veren küme atamaları seçilmiştir. En iyi kümeleme, oluşan kümelerin veri setini temsil edebilirliğine (kümeler arası hatanın toplam hataya oranı) bakılarak seçildikten sonra, sınav kritiklik seviyeleri oluşan kümelere göre tanımlanmıştır. *k*-ortalamlar ile elde edilen kümeleme sonucu da Ek 3’te yer almaktadır.

Ward yöntemi ve *k*-ortalamlar ile elde edilen kümeleme sonuçları karşılaştırıldığında, bir ders dışında (ekte yer alan tablolarda ilgili ders işaretlenmiştir) tüm derslerin sınavlarının aynı kümelere, dolayısıyla aynı kritiklik seviyelerine atandığı görülmektedir. Hatanın karesi temelli kümeleme yapan her iki yöntemin de yaklaşık olarak aynı sonucu vermesi sebebiyle, sadece bir yöntemin (*k*-ortalamlar) sonuçları dikkate alınıp hedef programlama modeline girdi oluşturması için kullanılmıştır. Diğer kümeleme yöntemine göre (Ward yöntemi) elde edilen sonuçlar da benzer şekilde hedef programlama modeline girdi oluşturacak şekilde kullanılabilir.

Çalışmada hedef programlama modeline girdi oluşturmak için kullanılan *k*-ortalamlar kümeleme algoritması sonuçlarına göre sınav kritiklik seviyeleri belirlenen ders sayıları ile

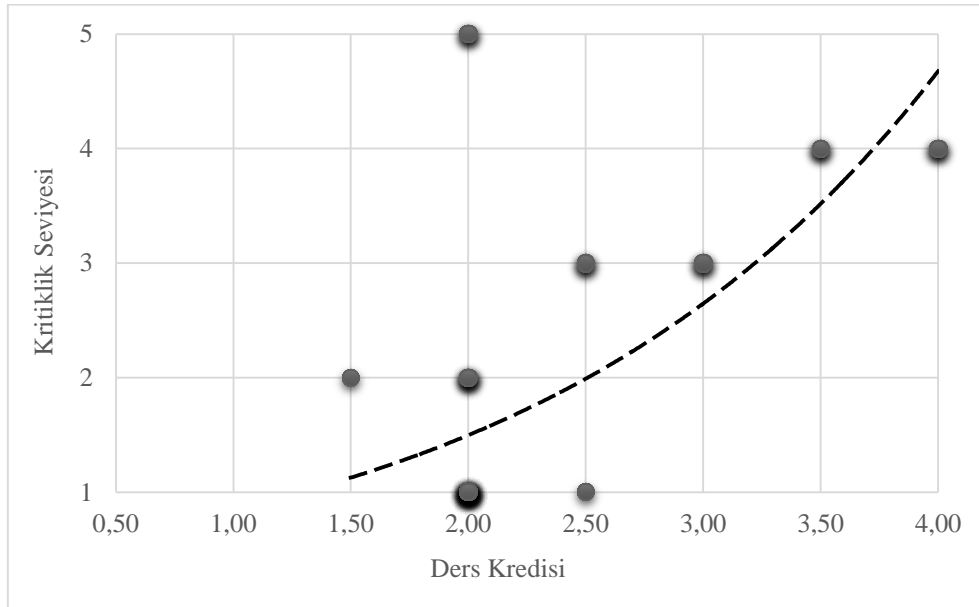
Çavdur ve Kose (2016) tarafından uygulanan bulanık mantık temelli yaklaşıma göre sınav kritiklik seviyeleri belirlenen ders sayılarının karşılaştırılması Tablo 2’de verilmektedir.

Tablo 2. Ders sayılarının sınav kritiklik seviyelerine göre karşılaştırılması

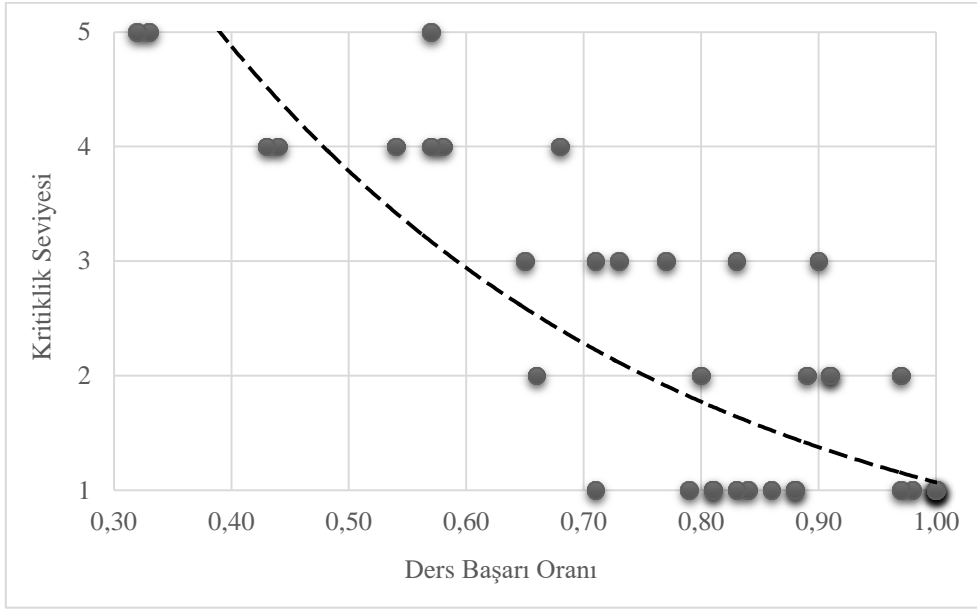
Kritiklik Seviyesi	<i>k</i> -Ortalamalar	Bulanık Mantık
Çok Düşük (1)	17	17
Düşük (2)	6	11
Orta (3)	6	4
Yüksek (4)	6	6
Çok Yüksek (5)	3	0

Bu çalışmada *k*-ortalamalar ile elde edilen kümeleme sonuçları ile Çavdur ve Kose (2016) tarafından elde edilen sonuçlar karşılaştırıldığında, her iki çalışmada da sınav kritiklik seviyesi çok düşük (1) olarak belirlenen derslerin eşit sayıda olduğu görülmektedir (her iki kümedeki toplam 17 dersin 16’sı aynı 1’i farklıdır). Diğer sınav kritiklik seviyelerine atanan ders sayılarında farklılık görülmekle birlikte, bulanık mantık ile elde edilen sınav kritiklik seviyelerinde, kritiklik seviyesi çok yüksek (5) olan sınav bulunmamaktadır. *k*-ortalamalar ile elde edilen kümeleme sonucunda ise üç adet sınavın kritiklik seviyelerinin çok yüksek olduğu görülmektedir.

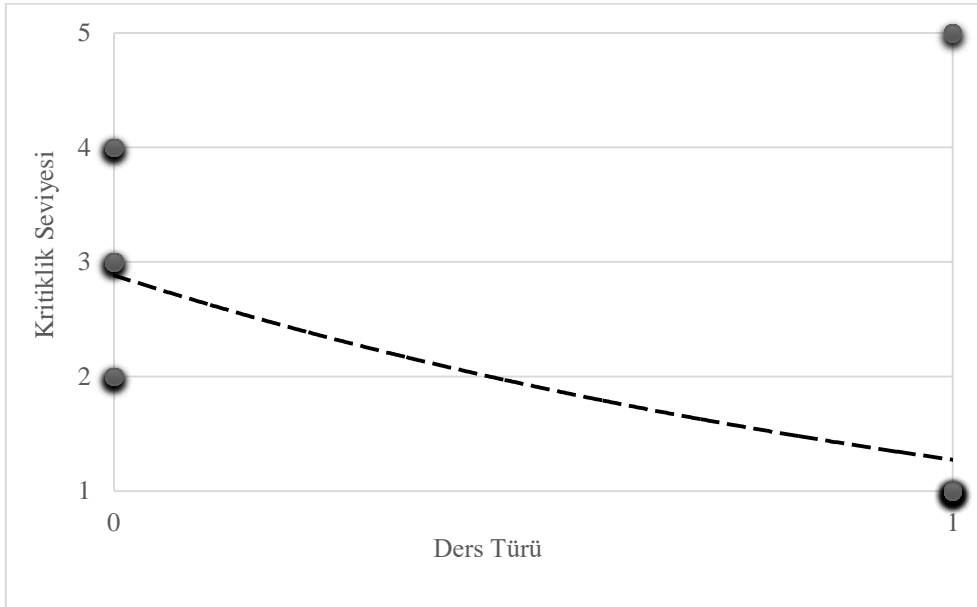
k-ortalamalar algoritmasının sonucunda belirlenen kritiklik seviyelerine göre her özniteliğin atandığı kümeleri (sınav kritiklik seviyesi) gösteren grafikler, Şekil 3-5’te verildiği gibidir. Grafiklerde daha yoğun olan noktalar, özniteliğin ilgili değerinin, diğer değerlerine göre daha fazla olduğunu göstermektedir.



Şekil 3:
Ders kredisine göre kritiklik seviyeleri



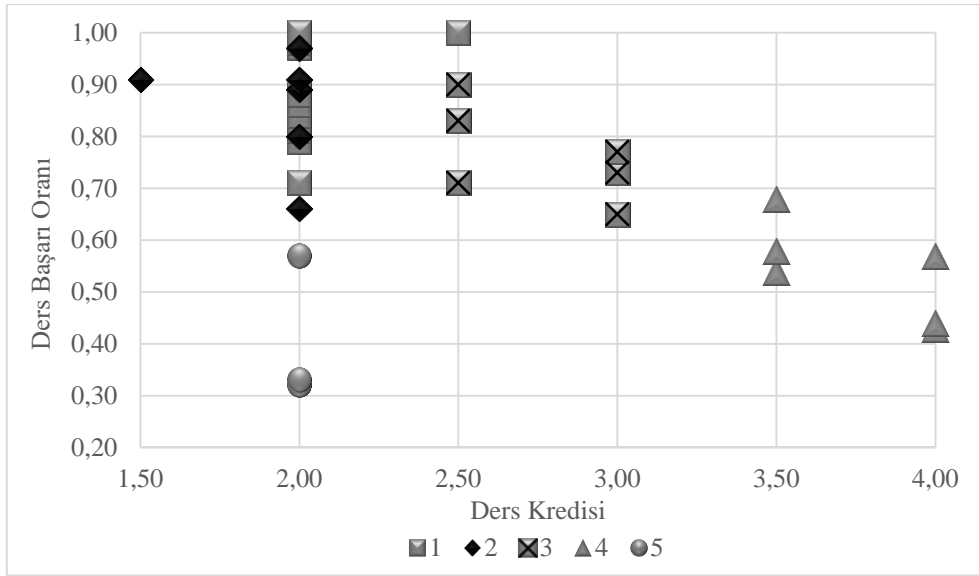
Şekil 4:
Ders başarı oranına göre kritiklik seviyeleri



Şekil 5:
Ders türüne göre kritiklik seviyeleri

Elde edilen kümeleme sonuçlarından, ders kredisi arttıkça kritiklik seviyesinin artan bir eğilim gösterdiği, ders başarı oranı arttıkça kritiklik seviyesinin azaldığı görülmektedir. Dersin zorunlu ya da seçmeli olmasına bağlı olarak ise genel olarak zorunlu derslerden seçmeli derslere doğru azalan bir eğilim görülmekle birlikte, seçmeli dersler arasında az sayıda da olsa sınav kritiklik seviyesi bütün derslerden daha yüksek olan (5) derslerin de olduğu görülmektedir. Söz konusu derslerin sınavlarının çok yüksek kritiklik seviyesine sahip olmasının nedeni ise bu derslerdeki ders başarı oranlarının bütün dersler arasındaki en düşük orana sahip olmalarıdır. Dolayısıyla, dersin zorunlu ya da seçmeli olduğu durumlardaki kritiklik seviyeleri, ders başarı

oranı ve kredisine göre değişen bir parametre olarak karşımıza çıkmaktadır. Sonuç olarak, sınavların kritiklik seviyelerinin belirlenmesinde dikkate alınan parametrelerin kritiklik seviyesi üzerindeki etkilerinin ayrı ayrı dikkate alınması yerine, bütünleşik olarak değerlendirilmesinin daha doğru sonuçlar elde edilmesine katkı sağlayacağı görülmektedir. Bir başka deyişle, bir dersin kredisinin ve başarı oranının çok yüksek veya düşük olması ya da ders türünün zorunlu/seçmeli olmasının, söz konusu dersin sınavının ne kadar kritik olduğunun belirlenmesi için tekil olarak ele alınması yeterli olmayabilir. Bundan dolayı, yapılan kümeleme ile üç parametre de birlikte dikkate alınıp, sınav kritiklik seviyeleri belirlenmiştir. Şekil 6'da, ders kredisine ve ders başarı oranı özneliklerinin, aldığı farklı değerler altında atandığı kümeler örnek olarak verilmiştir. Şekilden de görüldüğü gibi genel olarak ders kredisinin artışının ve başarı oranındaki düşüşün, sınav kritiklik seviyesinin artmasına neden olduğu görülmektedir. Bu genel kuralı bozan ise düşük başarı oranlarına sahip, düşük kredili derslerin sınavlarının çok yüksek kritikliğe sahip olduklarını gösteren durumdur.



Şekil 6:
Ders kredisine ve başarı oranına göre kümeler

Sınavların kritiklik seviyeleri hesaplandıktan sonra, her bir hedefin eşit öneme sahip olduğu varsayılarak ($w_1 = \dots = w_5 = 1$) hedef programlama modeli, Maximal Software-Mathematical Programming Language (MPL) ortamında, Gurobi 6 çözücüsü kullanılarak, AMD Turion 2-P520 CPU 2.30GHz işlemcili ve 4 GB belleğe sahip bir bilgisayarda 2,88 saniyede çözdürülmüştür. Burada, Pazartesi-Cuma arasında, toplam 2 haftalık (10 gün) bir sınav çizelgesi planlanmaktadır. Her sınavın en fazla 1 saat 45 dakika olduğu varsayılarak, ilk zaman aralığı 09:00-10:45, ikinci zaman aralığı 11:00-12:45 vb. olup, bir gün içinde toplam 4 farklı zaman aralığı bulunmaktadır. Kümeleme sonucunda belirlenen kritiklik seviyelerinin, uzman kişi tarafından oluşturulan sınav çizelgesinde de dikkate alındığı varsayımıyla, gerçek hayatta oluşturulan çizelge (Ek 4) ile model tarafından elde edilen çizelgenin (Ek 5) belirlenen hedefler açısından karşılaştırılması aşağıdaki gibidir:

1. Bir günde, farklı yarıyıldan olan zaman aralığının sayısı kadar (4) sınav çizelgelenmek istenmektedir.
 - (a) Uzman kişi tarafından oluşturulan çizelgede, 6 gün içinde günde 4 sınav planlanırken; modelde, 8 gün içinde günde 4 sınav planlanmıştır. Diğer bir deyişle,

- uzman kişi tarafından oluşturulan çizelgede 4 gün için günde 4 sınav çizelgelenememişken; modelde, aynı durum sadece 2 gün için söz konusudur.
- (b) Uzman kişi tarafından oluşturulan çizelgede, her dönemden sınavın çizelgelenmediği 4 gün varken, modelde bu günlerin sayısı 3 olarak bulunmuştur.
2. Öğrencilerin iş yükü, sınavların kritiklik seviyelerine göre dengelenmek istenmektedir.
- (a) Birinci ve ikinci haftadaki sınavların kritiklik seviyeleri arasındaki farkların mutlak değerlerinin toplamı uzman kişi tarafından oluşturulan çizelge için 9, model sonucunda elde edilen çizelge için ise 2'dir.
- (b) Sınavların günlük kritiklik seviyeleri arasındaki farkların mutlak değerlerinin toplamı uzman kişi tarafından oluşturulan çizelge için 23, model sonucunda elde edilen çizelge için 4,8'dir.
3. Sınav döneminin her iki haftasında da öğretim elemanlarının sınavlarının eşit olarak dağıtılması istenmektedir.
- (a) Her öğretim elemanı için haftalık sınav sayısı arasındaki farkların mutlak değerlerinin toplamı uzman kişi tarafından oluşturulan çizelge için 30 iken model sonucunda elde edilen çizelge için 22'dir.

Tablo 3. Sonuçların özeti

Hedefler	Uzman Kişi	Bulanık Mantık Temelli Yaklaşım	<i>k</i> -Ortalamalar Temelli Yaklaşım
1.(a)	4,0	2,0	2,0
1.(b)	4,0	3,0	3,0
2.(a)	9,0	1,0	2,0
2.(b)	23,0	5,0	4,8
3.(a)	30,0	22,0	22,0

Tablo 3'te özetlenen sonuçlara göre önerilen yaklaşım ile elde edilen çizelgenin, çalışmada ele alınan tüm hedefler açısından, uzman kişi tarafından oluşturulan çizelgeden genel olarak daha iyi sonuçlar ürettiği görülmektedir. Ayrıca, Cavdur ve Kose (2016) tarafından geliştirilen modelin bulanık mantık ile belirlenen sınav kritiklik seviyeleri kullanılarak elde ettiği sonuçlar (bulanık mantık temelli yaklaşım) ile bu çalışmada *k*-ortalamar ile belirlenen kritiklik seviyeleri kullanılarak elde edilen sonuçların (*k*-ortalamar temelli yaklaşım) yaklaşık olarak aynı olduğu görülmektedir. Farklı olan 2.(a) ve 2.(b) hedeflerinin aynı değerlerde bulunamaması da söz konusu yaklaşımlarla belirlenen sınav kritiklik seviyelerindeki farklılıklardan kaynaklanmaktadır. Buna ek olarak, modelin çözüm süresi ile uzman kişi tarafından oluşturulan çizelgenin hazırlanması için harcanan süre göz önüne alındığında, modelin birkaç saniye içinde en iyi çözümü elde etmesi, kullanılan işgücü/insan kaynağı açısından da tasarruf sağlandığını göstermektedir.

5. TARTIŞMA

Herkes için mükemmel bir sınav programı oluşturmak genellikle mümkün olmadığından, en azından adil bir çizelge oluşturmaya çalışmak çok önemlidir. Bu çalışmada, Cavdur ve Kose (2016) tarafından, sınav kritiklik seviyelerini belirlemek için kullanılan yaklaşıma bir alternatif olarak, kümeleme yöntemlerinin kullanılması önerilmektedir. Çalışma kapsamında, kümeleme yöntemlerinden olan *k*-ortalamar algoritması ve Ward yöntemi kullanılarak sınav kritiklik seviyeleri belirlenmiştir. Buna ek olarak, kritiklik seviyeleri belirlenen sınavların ataması için de yine Cavdur ve Kose (2016) tarafından geliştirilen hedef programlama modelinden

yararlanılmıştır. Sonuç olarak, söz konusu çalışmada yapıldığı gibi bu çalışmada da sınav kritiklik seviyelerinin belirlenmesi ve sınav çizelgesinin oluşturulması üzere probleminin çözümü için iki aşamadan oluşan bir yaklaşım önerilmektedir.

Yapılan çalışma, Uludağ Üniversitesi-Mühendislik Fakültesi-Endüstri Mühendisliği Bölümü'nde gerçekleştirilmiştir. Genel sınav çizelgeleme kısıtlarına ek olarak, sadece öğrenciler için değil aynı zamanda öğretim elemanları için de sınavların eşit dağılımı göz önünde bulundurulmuştur. Sonuçlar, kümeleme yöntemi sonuçlarının girdi olarak kullanıldığı hedef programlama modeli ile elde edilen çizelgenin, uzman kişi tarafından oluşturulan çizelgeye göre belirlenen hedefleri daha iyi karşıladığını göstermektedir. Buna ek olarak önerilen yaklaşımın, daha büyük boyutlu gerçek yaşam problemlerinde, sınav çizelgesi oluşturmak için kullanılma potansiyeli bulunmaktadır.

İleriki çalışmalarda, bu çalışmada yer almayan sınıf ve gözetmen atamalarının da dikkate alındığı bir sınav çizelgesinin oluşturulmasına yönelik yaklaşımlar önerilebilir. Ayrıca, çizelgeleme probleminin sadece bir bölüm özelinde değil üniversite genelinde ele alınmasıyla çalışma kapsamı genişletilerek, böyle bir problemin çözümü için bir karar destek sistemi geliştirilebilir.

KAYNAKLAR

1. Alsabti, K., Ranka, S. ve Singh, V. (1997) An efficient k-means clustering algorithm, *Electrical Engineering and Computer Science*.
2. Al-Yakoob, S. M. ve Sherali, H. D. (2006) Mathematical programming models and algorithms for a class-faculty assignment problem, *European Journal of Operational Research*, 173(2), 488-507. doi: 10.1016/j.ejor.2005.01.052
3. Al-Yakoob, S. M. ve Sherali, H. D. (2015) Mathematical models and algorithms for a high school timetabling problem, *Computers & Operations Research*, 61, 56-68. doi:10.1016/j.cor.2015.02.011
4. Babaei, H., Karimpour, J. ve Hadidi, A. (2015) A survey of approaches for university course timetabling problem, *Computers & Industrial Engineering*, 86, 43-59. doi:10.1016/j.cie.2014.11.010
5. Babaei, H., Karimpour, J. ve Mavizi, S. (2016) Using k-means clustering algorithm for common lecturers timetabling among departments, *Advances in Computer Science: An International Journal*, 5(1), 86-102.
6. Babaei, H., Karimpour, J. ve Mavizi, S. (2017) Using fuzzy c-means clustering algorithm for common lecturer timetabling among departments, *Journal of Advances in Computer Engineering and Technology*, 3(1). doi:10.1109/ICCKE.2016.7802147
7. Badoni, R. P., Gupta, D. K. ve Mishra, P. (2014) A new hybrid algorithm for university course timetabling problem using events based on groupings of students, *Computers & Industrial Engineering*, 78, 12-25. doi:10.1016/j.cie.2014.09.020
8. Bijuraj, L. V. (2013) Clustering and its applications, *Proceedings of National Conference on New Horizons in IT-NCNHIT*, 169-172.
9. Burke, E. K. ve Petrovic, S. (2002) Recent research directions in automated timetabling, *European Journal of Operational Research*, 140(2), 266-280. doi:10.1016/S0377-2217(02)00069-3
10. Burke, E. K., Petrovic, S. ve Qu, R. (2006) Case-based heuristic selection for timetabling problems, *Journal of Scheduling*, 9(2), 115-132. doi:10.1007/s10951-006-6775-y

11. Carter, M. W., Laporte, G. ve Chinneck, J. W. (1994) A general examination scheduling system, *Interfaces*, 24(3), 109-120. doi: 10.1287/inte.24.3.109
12. Cavdur, F. ve Kose, M. (2016) A fuzzy logic and binary-goal programming-based approach for solving the exam timetabling problem to create a balanced-exam schedule, *International Journal of Fuzzy Systems*, 18(1), 119-129. doi:10.1007/s40815-015-0046-z
13. Chu, S. C., Chen, Y. T. ve Ho, J. H. (2006) Timetable scheduling using particle swarm optimization, *Innovative Computing, Information and Control*, 3, 324-327. doi:10.1109/ICICIC.2006.541
14. Daskalaki, S., Birbas, T. ve Housos, E. (2004) An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research*, 153(1), 117-135. doi:10.1016/S0377-2217(03)00103-6
15. Dutt, A., Ismail, M. A. ve Herawan, T. (2017) A systematic review on educational data mining, *IEEE Access*. doi:10.1109/ACCESS.2017.2654247
16. Eley, M. (2006) Ant algorithms for the exam timetabling problem, *International Conference on the Practice and Theory of Automated Timetabling*, 364-382. doi:10.1007/978-3-540-77345-0_23
17. Everitt, B. S., Landau, S., Leese, M. ve Stahl, D. (2011) Hierarchical clustering, *Cluster Analysis, 5th Edition*, 71-110.
18. Fonseca, G. H., Santos, H. G., Carrano, E. G. ve Stidsen, T. J. (2017) Integer programming techniques for educational timetabling, *European Journal of Operational Research*, 262(1), 28-39. doi:10.1016/j.ejor.2017.03.020
19. Frigui, H. ve Krishnapuram, R. (1999) A robust competitive clustering algorithm with applications in computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 450-465. doi:10.1109/34.765656
20. Hamerly, G. (2003) Learning structure and concepts in data using data clustering, *Unpublished Doctoral Dissertation*, University of California, San Diego.
21. Hartigan, J. A., ve Wong, M. A. (1979) Algorithm AS 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1), 100-108. doi:10.2307/2346830
22. Ilic, M., Spalevic, P., Ilic, S., Milivojevic, Z., Veinovic, M. ve Princevic, B. (2015) Data mining techniques for student timetable optimization, *INFOTEH-JAHORINA*, 14.
23. Jain, A. K. (2010) Data clustering: 50 years beyond k-means, *Pattern Recognition Letters*, 31(8), 651-666. doi:10.1016/j.patrec.2009.09.011
24. Johnes, J. (2015) Operational research in education, *European Journal of Operational Research*, 243(3), 683-696. doi:10.1016/j.ejor.2014.10.043
25. Kaur, E. J. ve Kaur, E. A. (2014) Timetable scheduling using modified clustering, *International Journal of Research in Information Technology (IJRIT)*, 2(7), 1-8.
26. Kaur, N., Sahiwal, J. K. ve Kaur, N. (2012) Efficient k-means clustering algorithm using ranking method in data mining, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(3), pp-85.
27. MacQueen, J. (1967), Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14), 281-297.

28. Madhulatha, T. S. (2012) An overview on clustering methods, *IOSR Journal of Engineering*, 2(4), 719-725.
29. MirHassani, S. A. ve Habibi, F. (2013) Solution approaches to the course timetabling problem, *Artificial Intelligence Review*, 1-17. doi:10.1007/s10462-011-9262-6
30. Muklason, A., Parkes, A. J., Özcan, E., McCollum, B. ve McMullan, P. (2017) Fairness in examination timetabling: Student preferences and extended formulations, *Applied Soft Computing*, 55, 302-318. doi:10.1016/j.asoc.2017.01.026
31. Omran, M. G., Engelbrecht, A. P. ve Salman, A. (2007) An overview of clustering methods, *Intelligent Data Analysis*, 11(6), 583-605.
32. Pillay, N. ve Banzhaf, W. (2010) An informed genetic algorithm for the examination timetabling problem. *Applied Soft Computing*, 10(2), 457-467. doi:10.1016/j.asoc.2009.08.011
33. Pongcharoen, P., Promtet, W., Yenradee, P. ve Hicks, C. (2008) Stochastic optimisation timetabling tool for university course scheduling, *International Journal of Production Economics*, 112(2), 903-918. doi:10.1016/j.ijpe.2007.07.009
34. Romero, C. ve Ventura, S. (2007) Educational data mining: A survey from 1995 to 2005, *Expert Systems with Applications*, 33(1), 135-146. doi:10.1016/j.eswa.2006.04.005
35. Shatnawi, S. M., Albaloooshi, F. ve Rababa'h, K. (2012) Generating Timetable and Students schedule based on data mining techniques, *International Journal of Engineering Research and Applications (IJERA)*, 2(4), 1638-1644.
36. Turi, R. H. (2001) Clustering-based colour image segmentation, *PhD Thesis*, Monash University.
37. Ünal, Y. Z. ve Uysal, Ö. (2014) A new mixed integer programming model for curriculum balancing: Application to a Turkish university, *European Journal of Operational Research*, 238(1), 339-347. doi:10.1016/j.ejor.2014.03.015
38. Veenstra, M. ve Vis, I. F. (2016) School timetabling problem under disturbances, *Computers & Industrial Engineering*, 95, 175-186. doi:10.1016/j.cie.2016.02.011
39. Vermuyten, H., Lemmens, S., Marques, I. ve Beliën, J. (2016) Developing compact course timetables with optimized student flows, *European Journal of Operational Research*, 251(2), 651-661. doi:10.1016/j.ejor.2015.11.028
40. Ward Jr, J. H. (1963) Hierarchical grouping to optimize an objective function, *Journal of the American Statistical Association*, 58(301), 236-244.
41. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J. ve Steinberg, D. (2008) Top 10 algorithms in data mining, *Knowledge and Information Systems*, 14(1), 1-37. doi:10.1007/s10115-007-0114-2

Ek 1. Hedef programlama modeli

- i : ders, $i = 1, \dots, n_I, i \in I$ tüm derslerin kümesi
- j : öğretim elemanı, $j = 1, \dots, n_J, j \in J$, tüm öğretim elemanlarının kümesi
- k : dönemler, $k = 1, \dots, n_K, k \in K$, tüm dönemlerin kümesi
- t : zaman aralığı, $t = 1, \dots, n_T, t \in T$, tüm zaman aralıklarının kümesi
- u : zaman aralığı (her gün için), $u = 1, \dots, 3, u \in t \in T$, tüm zaman aralıklarının kümesi
- T_1, T_2 : sınav döneminin birinci ve ikinci haftasındaki zaman aralıklarının kümesi

- $T_F = 1, 5, 9, \dots, 37$: her günün ilk zaman aralığına karşılık gelen zaman aralıklarının kümesi
 - $I_O: \{(i_1, i_2) | i_1, i_2 \in I, i_1 \neq i_2\}$: çakışan ders çiftlerinin kümesi
 - c_i : i dersinin sınav kritiklik seviyesi (kümeleme yöntemi ile belirlenmiş olan)
 - c_{avg} : bir gündeki ortalama kritiklik seviyesi
 - n_{max} : bir gündeki maksimum sınav sayısı
- x_{ijkt} karar değişkenleri; j öğretim elemanının, k dönemindeki i dersi t zaman aralığına atanırsa 1, aksi halde 0 değerini almaktadır

$$\min z(x, d) = w_1 \left(\sum_{t \in T_F} d_t^{-1} \right) + w_2 \left(\sum_{k=1}^{n_K} \sum_{t \in T_F} d_{kt}^{+2} + d_{kt}^{-2} \right) + w_3 \left(\sum_{k=1}^{n_K} d_k^{+3} + d_k^{-3} \right) + w_4 \left(\sum_{t \in T_F} d_t^{+4} + d_t^{-4} \right) + w_5 \left(\sum_{j=1}^{n_J} d_j^{+5} + d_j^{-5} \right) \quad (1)$$

$$\sum_{t=1}^{n_T} x_{ijkt} = 1, \quad \forall i \in I, j \in J, k \in K \quad (2)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{k=1}^{n_K} \sum_{u=0}^3 x_{i,j,k,t+u} \leq n_{max}, \quad t \in T_F \quad (3)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} x_{ijkt} \leq 1, \quad \forall k \in K, \forall t \in T \quad (4)$$

$$\sum_{j=1}^{n_J} \sum_{k=1}^{n_K} x_{i_1,j,k,t} + \sum_{j=1}^{n_J} \sum_{k=1}^{n_K} x_{i_2,j,k,t} \leq 1, \quad \forall t \in T, \forall (i_1, i_2) \in I_O \quad (5)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{k=1}^{n_K} \sum_{u=0}^3 x_{i,j,k,t+u} - d_t^{+1} + d_t^{-1} = n_{max}, \quad t \in T_F \quad (6)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{u=0}^3 x_{i,j,k,t+u} - d_{kt}^{+2} + d_{kt}^{-2} = 1, \quad \forall k \in K, t \in T_F \quad (7)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{t \in T_1} c_i x_{ijkt} - \sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{t \in T_2} c_i x_{ijkt} - d_k^{+3} + d_k^{-3} = 0, \quad \forall k \in K \quad (8)$$

$$\sum_{i=1}^{n_I} \sum_{j=1}^{n_J} \sum_{k=1}^{n_K} \sum_{u=0}^3 c_i x_{i,j,k,t+u} - d_t^{+4} + d_t^{-4} = \frac{\sum_{i=1}^{n_I} c_i}{n_I} = c_{avg}, \quad t \in T_F \quad (9)$$

$$\sum_{i=1}^{n_I} \sum_{k=1}^{n_K} \sum_{t \in T_1} x_{ijkt} - \sum_{i=1}^{n_I} \sum_{k=1}^{n_K} \sum_{t \in T_2} x_{ijkt} - d_j^{+5} + d_j^{-5} = 0, \quad \forall j \in J \quad (10)$$

Ek 2. Ward yöntemi kümeleme sonucu

Ders Kodu	Kredi	Başarı Oranı	Tür	Kritiklik Seviyesi	Kümeler
YAD102	2.00	1.00	1	1	
END2018	2.00	0.98	1	1	
END2202	2.00	0.81	1	1	
END2203	2.00	0.84	1	1	
END3036	2.00	0.86	1	1	
END3066	2.50	1.00	1	1	
END3142	2.00	1.00	1	1	
END3236	2.00	1.00	1	1	
END3238	2.00	0.71	1	1	
END3244	2.00	0.79	1	1	Küme-1
END4058	2.00	1.00	1	1	
END4060	2.00	0.57	1	1	
END4062	2.00	0.88	1	1	
END4064	2.00	0.81	1	1	
END4236	2.00	0.83	1	1	
END4276	2.00	0.88	1	1	
ISL1004	2.00	1.00	1	1	
MAK2092	2.00	0.97	1	1	
TUD102	2.00	0.97	0	2	
END1216	2.00	0.89	0	2	
END2012	2.00	0.80	0	2	
END2024	2.00	0.66	0	2	Küme-2
END4010	1.50	0.91	0	2	
IKT1003	2.00	0.91	0	2	
END1014	3.00	0.65	0	3	
END1018	2.50	0.83	0	3	
END2028	2.50	0.71	0	3	
END2030	3.00	0.77	0	3	Küme-3
END3068	3.00	0.73	0	3	
END3070	2.50	0.90	0	3	
END3032	3.50	0.54	0	4	
END3034	3.50	0.68	0	4	
END4072	3.50	0.58	0	4	
FZK1072	4.00	0.44	0	4	Küme-4
MAT1072	4.00	0.43	0	4	
MAT2083	4.00	0.57	0	4	
END4074	2.00	0.33	1	5	
MAK2090	2.00	0.32	1	5	Küme-5

Ek 3. *k*-ortalamlar algoritması kümeleme sonucu

Ders Kodu	Kredi	Başarı Oranı	Tür	Kritiklik Seviyesi	Kümeler
YAD102	2.00	1.00	1	1	
END2018	2.00	0.98	1	1	
END2202	2.00	0.81	1	1	
END2203	2.00	0.84	1	1	
END3036	2.00	0.86	1	1	
END3066	2.50	1.00	1	1	
END3142	2.00	1.00	1	1	
END3236	2.00	1.00	1	1	
END3238	2.00	0.71	1	1	Küme-1
END3244	2.00	0.79	1	1	
END4058	2.00	1.00	1	1	
END4062	2.00	0.88	1	1	
END4064	2.00	0.81	1	1	
END4236	2.00	0.83	1	1	
END4276	2.00	0.88	1	1	
ISL1004	2.00	1.00	1	1	
MAK2092	2.00	0.97	1	1	
TUD102	2.00	0.97	0	2	
END1216	2.00	0.89	0	2	
END2012	2.00	0.80	0	2	Küme-2
END2024	2.00	0.66	0	2	
END4010	1.50	0.91	0	2	
IKT1003	2.00	0.91	0	2	
END1014	3.00	0.65	0	3	
END1018	2.50	0.83	0	3	
END2028	2.50	0.71	0	3	Küme-3
END2030	3.00	0.77	0	3	
END3068	3.00	0.73	0	3	
END3070	2.50	0.90	0	3	
END3032	3.50	0.54	0	4	
END3034	3.50	0.68	0	4	
END4072	3.50	0.58	0	4	Küme-4
FZK1072	4.00	0.44	0	4	
MAT1072	4.00	0.43	0	4	
MAT2083	4.00	0.57	0	4	
END4060	2.00	0.57	1	5	
END4074	2.00	0.33	1	5	Küme-5
MAK2090	2.00	0.32	1	5	

**Ek 4. Gerçek hayatta kullanılan çizelge
(Parantez içindeki sayılar sınav kritiklik seviyesini göstermektedir)**

Gün	Zaman Aralığı			1.Sınıf (2.Dönem)	2.Sınıf (4.Dönem)	3.Sınıf (6.Dönem)	4.Sınıf (8.Dönem)
Pazartesi -1-	1	09:00	10:45	MAT2083 (4)			
	2	11:00	12:45	END1018 (3)			
	3	13:00	14:45	END4010 (2)			
	4	15:00	16:45	END3142 (1)			
Salı -2-	5	09:00	10:45	END4236 (1)			
	6	11:00	12:45	MAK2090 (5)			
	7	13:00	14:45	END3034 (4)			
	8	15:00	16:45	MAT1072 (4)			
Çarşamba -3-	9	09:00	10:45				
	10	11:00	12:45	END3238 (1)			
	11	13:00	14:45	END4074 (5)			
	12	15:00	16:45	END2028 (3)			
Perşembe -4-	13	09:00	10:45	END2012 (2)			
	14	11:00	12:45	END4072 (4)			
	15	13:00	14:45	END3070 (3)			
	16	15:00	16:45	FZK1072 (4)			
Cuma -5-	17	09:00	10:45	END3244 (1)			
	18	11:00	12:45	END4060 (5)			
	19	13:00	14:45				
	20	15:00	16:45	MAK2092 (1)			
Pazartesi -6-	21	09:00	10:45	END3032 (4)			
	22	11:00	12:45	IKT1003 (2)			
	23	13:00	14:45	END4064 (1)			
	24	15:00	16:45	END2030 (3)			
Salı -7-	25	09:00	10:45	END1014 (3)			
	26	11:00	12:45	END3236 (1)			
	27	13:00	14:45	END4058 (1)			
	28	15:00	16:45	END2018 (1)			
Çarşamba -8-	29	09:00	10:45				
	30	11:00	12:45	YAD102 (1)		END3066 (1)	
	31	13:00	14:45	END4276 (1)			
	32	15:00	16:45	END2024 (2)			
Perşembe -9-	33	09:00	10:45	END2202 (1)			
	34	11:00	12:45	END1216 (2)		END4062 (1)	
	35	13:00	14:45	END3068 (3)			
	36	15:00	16:45	END2203 (1)			
Cuma -10-	37	09:00	10:45	END3036 (1)			
	38	11:00	12:45	ISL1004 (1)			
	39	13:00	14:45	TUD102 (2)			
	40	15:00	16:45				

Ek 5. Model tarafından elde edilen çizelge
(Parantez içindeki sayılar sınav kritiklik seviyesini göstermektedir)

Gün	Zaman Aralığı			1.Sınıf	2.Sınıf	3.Sınıf	4.Sınıf
				(2.Dönem)	(4.Dönem)	(6.Dönem)	(8.Dönem)
Pazartesi -1-	1	09:00	10:45		MAT2083 (4)		END4276 (1)
	2	11:00	12:45				
	3	13:00	14:45			END3238 (1)	
	4	15:00	16:45	END1018 (3)			
Salı -2-	5	09:00	10:45			END3032 (4)	
	6	11:00	12:45		END2012 (2)		
	7	13:00	14:45				
	8	15:00	16:45				END4010 (2)
Çarşamba -3-	9	09:00	10:45		END2202 (1)		
	10	11:00	12:45			END3142 (1)	
	11	13:00	14:45	YAD102 (1)			END4060 (5)
	12	15:00	16:45				
Perşembe -4-	13	09:00	10:45			END3070 (3)	
	14	11:00	12:45				END4062 (1)
	15	13:00	14:45	IKT1003 (2)			
	16	15:00	16:45		END2030 (3)		
Cuma -5-	17	09:00	10:45		END2024 (2)		
	18	11:00	12:45			END3244 (1)	
	19	13:00	14:45				END4236 (1)
	20	15:00	16:45	FZK1072 (4)			
Pazartesi -6-	21	09:00	10:45		ISL1004 (1)		
	22	11:00	12:45			END3034 (4)	
	23	13:00	14:45			END2018 (1)	
	24	15:00	16:45	END1014 (3)			
Salı -7-	25	09:00	10:45	MAT1072 (4)	END2203 (1)		
	26	11:00	12:45			END3068 (3)	
	27	13:00	14:45				END4064 (1)
	28	15:00	16:45				
Çarşamba -8-	29	09:00	10:45		END2028 (3)		
	30	11:00	12:45				END4074 (5)
	31	13:00	14:45			END3236 (1)	
	32	15:00	16:45				
Perşembe -9-	33	09:00	10:45	END1216 (2)		END3066 (1)	
	34	11:00	12:45				END4058 (1)
	35	13:00	14:45		MAK2090 (5)		
	36	15:00	16:45				
Cuma -10-	37	09:00	10:45				END4072 (4)
	38	11:00	12:45	TUD102 (2)		END3036 (1)	
	39	13:00	14:45		MAK2092 (1)		
	40	15:00	16:45				

