



**ÇEVİRİM İÇİ SINAV ÇİZELGELEME PROBLEMİ İÇİN
ÇÖZÜM YAKLAŞIMLARI GELİŞTİRİLMESİ**

Merve BAKIR



T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇEVİRİM İÇİ SINAV ÇİZELGELEME PROBLEMİ İÇİN ÇÖZÜM
YAKLAŞIMLARI GELİŞTİRİLMESİ**

Merve BAKIR
0000-0002-2134-414X

Prof. Dr. Fatih ÇAVDUR
(Danışman)

YÜKSEK LİSANS
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2023
Her Hakkı Saklıdır

ÖZET

Yüksek Lisans Tezi

ÇEVİRİM İÇİ SINAV ÇİZELGELEME PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI GELİŞTİRİLMESİ

Merve BAKIR

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Fatih ÇAVDUR

Bu tez çalışmasında, çevrim içi sınavların yapıldığı sunucudaki yük dengesini sağlamak amacıyla çevrim içi sınav çizelgeleme problemi ele alınmıştır. Problemin çözümü için öncelikle bir tamsayı programlama modeli geliştirilmiştir. Geliştirilen model ile sınavlar en uygun zaman dilimlerine atanarak dengeli bir sınav çizelgesi elde edilmektedir. Çalışmanın uygulama aşamasında, Bursa Uludağ Üniversitesi Mühendislik Fakültesi bünyesinde bulunan sekiz adet bölüme ait veriler kullanılmıştır. Çalışmada, sunucudaki yük dengesinin bölümler özelinde lokal olarak sağlanması amacıyla dinamik bir yapı geliştirilmiştir. Geliştirilen dinamik yapı sayesinde ilgili bölümün veri setiyle model çalıştırılarak sınav çizelgesi üzerinde optimizasyon yapılmaktadır. Oluşturulan bu iteratif yapı ile sunucu yükünü azaltacak şekilde tekrar tekrar çözüm elde edilmektedir. Böylece sunucuda maksimum yükün olduğu zaman diliminde bu yükü oluşturan bölüm özelinde yük dengeleme yapılmaya çalışılmaktadır. Çalışmada önerilen çözüm yaklaşımı çeşitli performans parametreleri açısından değerlendirilmiştir. Buna göre çalışmanın amacı doğrultusunda beklendiği üzere, iterasyon sayısı arttıkça maksimum yükün olduğu zaman diliminde sınava giren maksimum öğrenci sayısının azaldığı gözlemlenmiştir. Bunun yanı sıra, geliştirilen yaklaşımın 300 iterasyon boyunca çalıştırılması ile ulaşılan sonuçlardaki maksimum yük değerlerine bakıldığında, sunucu kapasitesini aşan maksimum yük değerlerinin tatmin edici oranda olduğu görülmektedir. Bu sonuçlara bakılarak çalışmada geliştirilen çözüm yaklaşımının iyi bir performans sergilediği söylenebilir. Pandemi gibi öngörülemeyen durumların hayat akışını etkilememesi bakımından çevrim içi süreçlerin iyileştirilmesi zorlu olan süreç içerisinde ekstra sorunların yaşanmaması için önem teşkil etmektedir. Bu bakımdan bu tez çalışmasında ele alınan, çevrim içi sınavların yapıldığı sunucudaki yük dengesinin sağlanmasının, yük dengesizliğinden kaynaklı oluşabilecek sorunların önüne geçebileceği düşünülmektedir.

Anahtar Kelimeler: Sınav çizelgeleme, zaman çizelgeleme, sunucu yükü dengeleme, çevrim içi sınavlar, tamsayı programlama, algoritma tasarımı

2023, vi + 81 sayfa.

ABSTRACT

MSc/PhD Thesis

DEVELOPING SOLUTION APPROACHES FOR ONLINE EXAM SCHEDULING PROBLEM

Merve BAKIR

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Prof. Dr. Fatih ÇAVDUR

In this thesis, the online exam scheduling problem is discussed where it is aimed at balancing the load on the server on which the online exams are held. In order to solve the problem, first, an integer programming model is developed. With the developed model, a balanced exam schedule is obtained by assigning the exams to the most appropriate time slots. In the implementation phase of the study, data belonging to eight departments within Bursa Uludağ University Faculty of Engineering are used. In the study, a dynamic structure is developed in order to provide the load balance on the server locally in terms of the specific load of each department. Thanks to the dynamic structure developed, the model is run with the data set of the relevant department and optimization is performed on the exam schedule. With this iterative structure created, a solution is obtained repeatedly in a way that reduces the server load. Thus, it is aimed at balancing the load in the time slot of the maximum load on the server specifically considering the department that causes this load. The proposed solution approach in the study is evaluated in terms of various performance parameters. Accordingly, as expected for the purpose of the study, it is observed that as the iteration number increases, the maximum number of students who took the exam in the time slot when the maximum load occurred decreases. In addition, when the maximum loads are examined after running the developed approach for 300 iterations, it is seen that the maximum loads exceeding the server capacity are satisfactory. Based on these results, it can be stated that the solution approach developed in the study performed well. It is important to improve online processes so that some unpredictable situations such as pandemics do not affect the flow of life and extra problems are not experienced during such difficult times. In this respect, it is thought that ensuring the load balance on the server for online exams, as discussed in this thesis, can prevent the problems that may arise from the load imbalance.

Key words: Exam scheduling, scheduling, server load balancing, online exams, integer programming, algorithm design

2023, vi + 81 pages.

ÖNSÖZ ve/veya TEŞEKKÜR

Yüksek lisans öğrenim sürecim boyunca ilgi ve alakasını benden esirgemeyen, tecrübelerini benimle paylaşarak bana yol gösteren ve her daim destek olarak motivasyon kaynağım olan danışman hocam Prof. Dr. Fatih ÇAVDUR'a teşekkür eder, saygılarımı sunarım.

Yüksek lisans süreci boyunca tecrübelerini benimle paylaşmaktan çekinmeyerek bana çok şey katan, ihtiyaç duyduğum her anda yanımda olarak desteğini benden esirgemeyen Aslı SEBATLI SAĞLAM'a teşekkürlerimi sunarım.

Tüm hayatım boyunca maddi ve manevi desteklerini esirgemeyen, attığım her adımda yanımda olan aileme teşekkürlerimi sunarım.



Merve BAKIR
.../.../.....

İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ ve/veya TEŞEKKÜR.....	iii
ŞEKİLLER DİZİNİ.....	v
ÇİZELGELER DİZİNİ.....	vi
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI.....	4
2.1. Ders Çizelgeleme.....	4
2.2. Sınav Çizelgeleme.....	9
2.3. COVID-19 Pandemisinde Eğitim-Öğretim ve Sınav Süreçleri.....	11
3. MATERYAL ve YÖNTEM.....	13
3.1. Materyal.....	13
3.2. Yöntem.....	18
3.2.1. Tamsayılı programlama modeli.....	21
3.2.2. Geliştirilen algoritmanın tasarımı.....	24
4. BULGULAR ve TARTIŞMA.....	27
4.1. Veri Setleri.....	27
4.2. Dinamik Yapı ile Sınav Çizelgeleme Süreci.....	29
5. SONUÇ.....	41
KAYNAKLAR.....	43
EKLER.....	48
EK 1 300 İterasyon Boyunca Ulaşılan Sonuçları İçeren Çizelge.....	49
EK 2 En Yüksek Maksimum Yük Değeri Olan 937 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri.....	65
EK 3 En Düşük Maksimum Yük Değeri Olan 724 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri.....	73
ÖZGEÇMİŞ.....	81

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Matematiksel modelleme süreci	13
Şekil 3.2. Geliştirilen yaklaşımın akış diyagramı	20
Şekil 3.3. Algoritma sonuçlarının kaydedildiği tablonun yapısı.....	25
Şekil 4.1. 4. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 33. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri	33
Şekil 4.2. 6. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 44. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri	33
Şekil 4.3. 7. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 25. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri	34
Şekil 4.4. 300 iterasyon sonuçlarında 11. zaman dilimine karşılık gelen maksimum yük değerleri	36
Şekil 4.5. 300 iterasyon sonuçlarında 20. zaman dilimine karşılık gelen maksimum yük değerleri	37
Şekil 4.6. 300 iterasyon sonuçlarında 25. zaman dilimine karşılık gelen maksimum yük değerleri	38
Şekil 4.7. 300 iterasyon içerisindeki tüm maksimum yük değerleri	39
Şekil 4.8. 300 iterasyon sonucundaki tüm maksimum yük değerlerinin sunucu kapasitesini aşma yüzdeleri	40

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1. Literatürde yer alan ders çizelgeleme çalışmaları	7
Çizelge 2.2. Literatürde yer alan sınav çizelgeleme çalışmaları	10
Çizelge 4.1. Problem veri setleri içeriği	27
Çizelge 4.2. 300 iterasyon sonuçlarına göre tekrar sayıları	31



1. GİRİŞ

Zaman, sürekli akmaya devam eden, geri getirilemeyen ve insanoğlunun kontrol edemeyeceği şeylerden biridir. Bu bakımdan zaman kavramının insan hayatındaki bekli de en önemli olgu olduğu söylenebilir. Zaman, kontrol edilememesine karşın, doğru bir planlama ile en iyi şekilde yönetilebilir. Zaman yönetimi sayesinde süreçlerin en uygun şekilde planlanmasıyla kişiye kolaylık sağlanmaktadır.

Zaman çizelgeleme, çeşitli süreçlerin planlanması amacıyla birçok farklı alanda önem teşkil etmektedir. Literatürde zaman çizelgeleme problemi, ulaşım hizmetleri, hastaneler, spor tesisleri, süpermarketler ve eğitim kurumları gibi belirli bir görev için personel veya kaynakları planlamaya ihtiyaç duyan kuruluşlar tarafından ele alınmaktadır (Leite ve ark. 2021). Eğitim kurumlarındaki tüm kuralları ve ihtiyaçları ele alan, aynı zamanda personelin ve öğrencilerin birçok istek ve gereksinimlerini karşılayan bir zaman çizelgesinin oluşturulması, ilgili personel için önemli ancak son derece zor bir iştir (Daskalaki ve ark. 2004).

Üniversiteler gibi eğitim kurumlarında farklı zaman çizelgeleme problemleri bulunmaktadır. Örneğin derslerin belirli zaman dilimlerine çizelgelenmesi, derslerin uygun sınıflara atanması, hangi dersin hangi öğretim üyesine verilmesi gerektiğinin belirlenmesi ve sınavların belirli zaman dilimlerine çizelgelenmesi gibi problemler bunlar arasında yer almaktadır. Bu bağlamda sınav çizelgeleme problemi üniversite zaman çizelgeleme probleminin bir alt başlığı olup gerekli kısıtların yerine getirilmesi amacıyla, sınavların önceden belirlenmiş zaman dilimlerine atanmasını, sınıflara atanmasını ve gözetmenlerin sınavlara atanmasını içeren bir problemdir. Bu çalışma kapsamında sınav çizelgeleme probleminin bir özel kolu olan çevrim içi sınav çizelgeleme konusu ele alınacaktır.

Dünya genelinde pandemi süreçleri gibi yaşanabilecek birçok farklı acil durum söz konusudur. Bu gibi durumlarda doğal akışın devam edebilmesi adına çevrim içi olarak yürütülebilen süreçlere dahil olmak çok önemlidir. Günümüzde küresel boyutta yaşanan COVID-19 pandemi süreci de çevrim içi sistemlere olan ihtiyacı ortaya koymaktadır. Bu

bağlamda üniversiteler dahil birçok kuruluş faaliyetlerini çevrim içi ortamlarda yürütmeye çalışmaktadır. Birçok üniversitede pandemi sürecinde sınavlar çevrim içi ortamda gerçekleştirilmiştir. Böylelikle zor koşullar altında bulunulsa dahi eğitimin gereklilikleri yerine getirilmektedir. Bu durum, içinde bulunulan süreçte kolaylık sağlamakla birlikte birtakım zorlukları da beraberinde getirmektedir. Genel olarak pandemi döneminde uygulanan çevrim içi sınav süreci öğretim üyelerinin sunucu sistemine sınav sorularını yüklemesiyle başlamaktadır. Öğrenciler sorumlu oldukları derslerin sınavlarına belirlenen zaman dilimlerinde sunucuya giriş yaparak ulaşmaktadır. Bu noktada üniversite genelinde farklı birimlerden birçok öğrencinin aynı zaman diliminde tanımlanan sınavlara girmesi durumu söz konusu olmaktadır. Bu durum sunucu yükünün artmasına sebep olmaktadır. Sunucu yükünün artması sonucunda ise sistemde birtakım aksaklıklar meydana gelirken öğrencilerin sınavlara girmesinde sorunlar yaşanabilmektedir. Bunun neticesinde ise öğrencilerin giremedikleri ilgili sınavlar için yapılması gereken mazeret sınavlarında artış yaşanmaktadır. Mazeret sınavlarının yapılmasının hem ilgili öğretim üyesi hem üniversite kaynaklarının kullanılması hem de bu sınava girecek öğrenciler için verimsiz bir durum olduğu söylenebilir. Bu bağlamda mazeret sınavlarının azaltılması, öğretim üyesi ve sınava girecek öğrenciler için zaman tasarrufu sağlarken üniversite için ise kaynak tasarrufu sağlayacağından önemli bir durumdur.

Bu çalışmada, çevrim içi sınavlarda sunucu kapasitesi aşıldığında veya bir başka deyişle, aynı zaman diliminde sunucuya kapasitenin üstünde öğrenci girişi olduğunda sistemde oluşan aksaklıklar sonucunda meydana gelen mağduriyetlerin ortadan kaldırılması için bir çözüm yaklaşımı önerilmektedir. Bu bağlamda çalışma kapsamında, çevrim içi sınavların uygun zaman dilimlerine atanması ile dengeli bir sınav çizelgesi oluşturularak belli bir zaman diliminde ilgili bölüm için sunucudaki yük dengesi sağlanmaktadır. Böylelikle yaşanan durum sonucu yapılması zorunlu hale gelen mazeret sınavları için de bir çözüm sunulacağı düşünülmektedir.

Tez çalışmasında ele alınan problemin çözümü için öncelikle Bursa Uludağ Üniversitesi 2021-2022 eğitim öğretim yılı güz dönemindeki Mühendislik Fakültesi bünyesinde bulunan Bilgisayar Mühendisliği, Çevre Mühendisliği, Elektrik-Elektronik Mühendisliği,

Endüstri Mühendisliği, İnşaat Mühendisliği, Makine Mühendisliği, Otomotiv Mühendisliği ve Tekstil Mühendisliği bölümleri olmak üzere sekiz adet bölüme ait veriler elde edilmiştir. Elde edilen veriler gerekli düzenlemelerin yapılmasıyla sekiz bölüm için ayrı ayrı veri setleri haline getirilmiştir.

Çalışma kapsamında belirli kısıtlar dahilinde bir tamsayılı programlama modeli geliştirilmiştir. Bu kısıtlar içinde çakışan derslerin aynı zaman dilimine atanmasını önleyen bir kısıt mevcuttur. Bu kısıtta kullanılmak üzere her bölümün kendi içindeki çakışan dersler bilgisinin elde edilmesi için MATLAB ortamında ders çakışma matrisleri oluşturulmuştur. Geliştirilen matematiksel programlama modeli CPLEX Optimization Studio ortamında kodlanmıştır. Programın bölümlere ait veri setleri ile çalıştırılmasıyla her bölüm için ulaşılan, belli bir zaman diliminde sınava girebilecek maksimum öğrenci sayısı bilgisi Microsoft Excel ortamına aktarılmıştır. Böylece bölümler tarafından belli bir zaman diliminde sunucuya yüklenen yük bilgisine ulaşılmıştır. Çalışmanın amacı doğrultusunda modelde bulunan kısıtlar ile yük dengeleme yapılmaktadır. Buna göre elde edilen sonuçlar doğrultusunda, belli bir zaman diliminde sunucuya en fazla yük teşkil eden ilgili bölüme ait veri seti ile model çalıştırılmakta, modelde bulunan yük dengeleme kısıtı sayesinde sunucudaki yük dengelenmektedir. Yapılan bu işlemler dinamik bir hale dönüştürülerek iteratif bir yapıya ulaşılmıştır. Söz konusu dinamik yapı çalışma kapsamında 300 iterasyon boyunca devam ettirilerek sunucu yükünü azaltacak şekilde tekrar tekrar çözüm elde edilmektedir. Böylelikle belli bir zaman diliminde ilgili bölüm için sunucudaki yük dengelenmeye çalışılmaktadır.

2. KAYNAK ARAŞTIRMASI

Bu bölümde, tez çalışmasında ele alınan problem kapsamında literatürde yer alan çalışmalar; (i) ders çizelgeleme, (ii) sınav çizelgeleme ve (iii) COVID-19 pandemisinde eğitim-öğretim ve sınav süreçleri olmak üzere üç alt grupta sunulmaktadır.

2.1. Ders Çizelgeleme

Derslerin en uygun zaman dilimlerine atanmasıyla ders çizelgelerinin elde edildiği problemlere ders çizelgeleme problemleri denmektedir. Literatür incelendiğinde, eğitim kurumlarında ders çizelgeleme problemi için birçok çalışma yapıldığı görülmüştür. Ders çizelgeleme konusunu kesin çözüm yöntemleri ile ele alan çalışmalar arasından Daskalaki ve ark. (2004) tarafından yapılan çalışmada üniversite dersleri için bir zaman çizelgesi tamsayılı programlama yaklaşımı ile elde edilmiştir. Sorensen ve Dahms (2014) tarafından yapılan çalışmada lise zaman çizelgesi problemi için iki-aşamalı tamsayılı programlama modeli sunulmuş olup ilk aşamada dersler zaman dilimlerine atanırken ilk aşamanın çözümünü girdi olarak kullanan ikinci aşamada dersler sınıflara atanmaktadır. Ünal ve Uysal (2014) tarafından yapılan çalışmada üniversitede ders müfredatı dengeleme problemi çözümü için iki-amaçlı karışık-tamsayılı doğrusal programlama modeli önerilmiştir. Al-Yakoob ve Sherali (2015) lise zaman çizelgeleme problemi çözümü için iki-aşamalı bir karışık-tamsayılı programlama modeli geliştirmişler ve problemin çözümü için sütun oluşturma yöntemini kullanmışlardır. Phillips ve ark. (2015) üniversite ders çizelgelemesinde sınıf atama problemini çözmek için bir tamsayılı programlama yöntemi kullanmışlardır. Vermuyten ve ark. (2016) tarafından yapılan çalışmada üniversite binasındaki öğrenci akışlarını en aza indirmeyi amaçlayan ders çizelgeleme problemi için iki-aşamalı bir tamsayılı programlama yaklaşımı kullanılmıştır. Yapılan çalışmanın ilk aşamasında derslerin zaman dilimlerine ve sınıflara atanmasıyla öğretim üyesi memnuniyeti maksimize edilirken, ikinci aşamada sınıflar birinci aşamanın zaman çizelgesindeki derslere yeniden atanarak öğrenci akışı minimize edilmektedir. Dorneles ve ark. (2017) tarafından yapılan çalışmada lise zaman çizelgeleme problemi için karışık-tamsayılı programlama modeli sunulmuş ve sütun oluşturma yaklaşımı ile problem çözüme ulaştırılmıştır. Bir başka çalışmada Fonseca ve

ark. (2017) lise zaman çizelgeleme problemini tamsayı programlama yaklaşımı ile çözüme ulaştırmışlardır. Lindahl ve ark. (2018) tarafından yapılan zaman dilimleri, sınıflar ve zaman çizelgelerinin kalitesinin birbirini nasıl etkilediğini araştıran çalışmada karışık-tamsayı programlama yaklaşımı kullanılmıştır. Lemos ve ark. (2019) tarafından yapılan çalışmada üniversite zaman çizelgeleri için sınıf kullanım optimizasyonu problemine çözüm olarak iki-aşamalı tamsayı doğrusal programlama yaklaşımı önerilmiş olup büyük problemlerin çözümü için açgözlü bir algoritma önerilmiştir. Tassopoulos ve ark. (2020) tarafından yapılan çalışmada lise ders çizelgeleme problemi karışık-tamsayı programlama yaklaşımı ile çözüme ulaştırılmıştır. Barnhart ve ark. (2020) yaptıkları çalışmada COVID-19 pandemisi sırasında fiziksel mesafe gereksinimleri kapsamında müfredat planlama ve ders çizelgeleme ile ilgili problemler için tamsayı programlama yaklaşımı kullanmışlardır. Fathi ve ark. (2021) tarafından yapılan çalışmada müfredat planlaması yoluyla akademik binaların ısıtma/soğutma yükünü azaltmak için ikili tamsayı doğrusal programlama çözüm yaklaşımı önerilmiştir. Aizam ve ark. (2022) tarafından yapılan çalışmada standart bir karışık-tamsayı doğrusal programlama modelinin doğrulama sürecini onaylamak amacıyla üniversite ders çizelgeleme problemi ele alınmıştır. Çalışmada dersler çakışma olmadan başarılı bir şekilde çizelgenerek ders programı oluşturulmuştur. Oluşturulan bu ders programının idari personel tarafından yayınlanan orijinal ders programına göre daha kısa sürede oluşturulduğu belirtilmiştir.

Ders çizelgeleme konusunu sezgisel çözüm yöntemleri ile ele alan çalışmalar incelendiğinde De Causmaecker ve ark. (2009) iki-aşamalı bir çözüm yaklaşımı ile üniversite ders çizelgeleme problemini ele aldıkları görülmüştür. Yazarlar ilk aşamada benzer dersleri gruplandırarak ikinci aşama için problemi daha az karmaşık hale getiren bir yerel arama algoritması kullanmış olup ikinci aşamada ise tabu arama algoritması kullanarak problemi çözüme ulaştırmışlardır. Tassopoulos ve Beligiannis (2012) tarafından yapılan çalışmada lise zaman çizelgeleme problemi ele alınmış ve verimli zaman çizelgeleri oluşturmak için hibrit parçacık sürü optimizasyonu algoritması kullanılmıştır. Badoni ve ark. (2014) tarafından yapılan çalışmada üniversite ders çizelgeleme probleminin çözümü için öğrencilerin gruplandırılmasına dayalı etkinlikler kullanılarak genetik algoritma ve yerel arama algoritmasını birleştiren hibrit bir yaklaşım

sunulmuştur. Dorneles ve ark. (2014) tarafından yapılan çalışmada lise ders çizelgeleme probleminin çözümü için karışık-tamsayılı programlama modeli sunulmuş olup düzelt ve optimize et (fix and optimize) sezgisel yaklaşımı ile problem çözüme ulaştırılmıştır. Babaei ve ark. (2015) üniversite ders çizelgeleme problemlerinin çözümünde kullanılan mevcut yaklaşımların analiz edildiği bir çalışma ortaya koymuşlardır. Abdelhalim ve El Khayat (2016) yaptıkları çalışmada esnek bir üniversite zaman çizelgesi oluşturmak için genetik algoritma ve veri yakalama teknolojilerini kullanmışlardır. Song ve ark. (2017) yaptıkları çalışmada üniversitede enerji kullanımını açısından ders zaman çizelgesinin optimize edilmesi amacıyla çözüm yaklaşımı olarak genetik algoritma kullanmışlardır. Assi ve ark. (2018) tarafından yapılan çalışmada grafik renklendirme ve genetik algoritma çözüm yaklaşımlarıyla üniversite ders çizelgeleme problemi çözülmüştür. Yasari ve ark. (2019) tarafından yapılan çalışmada kayıt işleminin ön kayıt ve ders bırakma/ekleme aşamaları olarak iki adımda uygulandığı bir üniversite ders çizelgeleme problemi ele alınmıştır. Yazarlar ders zaman çizelgesini, ders iptal riskini ve ekleme/bırakma döneminden sonra meydana gelebilecek olası değişiklikleri göz önünde bulundurarak son zaman çizelgesinde istenmeyen durumun en aza indirilmesi için iki aşamalı bir stokastik programlama yaklaşımı geliştirmişlerdir. Ayrıca yazarlar çalışmalarında büyük boyutlu örneklerin çözümü için sezgisel bir algoritma geliştirmişlerdir. Tavakoli ve ark. (2020) tarafından yapılan çalışmada üniversite ders çizelgeleme problemi için birinci aşamada öğretim üyesine ders atayan, ikinci aşamada öğretim üyesine atanan dersleri çizelgeleyen ve üçüncü aşamada her derse bir derslik atanmasını sağlayan üç-aşamalı bir sezgisel algoritma önerilmiştir. Chen ve ark. (2020) üniversite ders çizelgeleme probleminin çözümü için tüm katı kısıtlamaları karşılayan tabu arama algoritması kullanmışlardır. Gülcü ve Akkan (2020) tarafından yapılan çalışmada üniversite ders çizelgeleme problemi için çok-amaçlı tavlama benzetimi algoritmasının iki farklı versiyonu geliştirilmiştir. Akkan ve ark. (2021) tarafından yapılan çalışmada üniversite zaman çizelgesinin kalite bozulmasını en aza indirmek amacıyla tavlama benzetimi algoritması ve tamsayılı programlama tabanlı bir algoritma sunulmuştur. Song ve ark. (2021) tarafından yapılan çalışmada müfredata dayalı üniversite ders çizelgeleme problemini çözmek için tavlama benzetimi algoritması kullanılmıştır. De Coster ve ark. (2021) tarafından yapılan çalışmada müfredat tabanlı ders çizelgeleme problemi için örnek uzay analizi yapılarak çözüm yaklaşımında çeşitli algoritmalar için performans

analizi yapılmış ve böylece algoritma seçimi gerçekleştirilmiştir. De la Rosa-Rivera ve ark. (2021) tarafından yapılan çalışmada müfredat tabanlı ders çizelgeleme örneklerinin ilk çözümlerini geliştirmek için örnek başına algoritma seçim modeli sunulmuştur. Sun ve ark. (2021) tarafından yapılan çalışmada bina enerji verimliliğini dikkate alarak üniversite zaman çizelgesini optimize etmek için ders takvimi simüle edilmiş ve çözüm için genetik algoritma kullanılmıştır. Ders çizelgeleme konusunda incelenen çalışmalar Çizelge 2.1’de sunulmuştur.

Çizelge 2.1. Literatürde yer alan ders çizelgeleme çalışmaları

Ders Çizelgeleme	Kesin Çözüm Yöntemi	Sezgisel Çözüm Yöntemi
Daskalaki ve ark. (2004)	✓	
De Causmaecker ve ark. (2009)		✓
Tassopoulos ve Beligiannis (2012)		✓
Badoni ve ark. (2014)		✓
Sorensen ve Dahms (2014)	✓	
Dorneles ve ark. (2014)		✓
Ünal ve Uysal (2014)	✓	
Al-Yakoob ve Sherali (2015)	✓	
Phillips ve ark. (2015)	✓	
Babaei ve ark. (2015)		✓
Vermuyten ve ark. (2016)	✓	
Abdelhalim ve El Khayat (2016)		✓
Song ve ark. (2017)		✓

Çizelge 2.1. Literatürde yer alan ders çizelgeleme çalışmaları (devam)

Ders Çizelgeleme	Kesin Çözüm Yöntemi	Sezgisel Çözüm Yöntemi
Dorneles ve ark. (2017)	✓	
Fonseca ve ark. (2017)	✓	
Assi ve ark. (2018)		✓
Lindahl ve ark. (2018)	✓	
Yasari ve ark. (2019)		✓
Lemos ve ark. (2019)	✓	
Tavakoli ve ark. (2020)		✓
Chen ve ark. (2020)		✓
Gülcü ve Akkan (2020)		✓
Tassopoulos ve ark. (2020)	✓	
Barnhart ve ark. (2020)	✓	
Akkan ve ark. (2021)		✓
Fathi ve ark. (2021)	✓	
Song ve ark. (2021)		✓
De Coster ve ark. (2021)		✓
De la Rosa-Rivera ve ark. (2021)		✓
Sun ve ark. (2021)		✓
Aizam ve ark. (2022)	✓	

2.2. Sınav Çizelgeleme

Sınavların en uygun zaman dilimlerine atanmasıyla sınav çizelgelerinin elde edildiği problemlere sınav çizelgeleme problemleri denmektedir. Literatürde bulunan çalışmalar incelendiğinde, sınav çizelgeleme problemine çözüm sağlamak amacıyla yapılan çalışmalara ulaşılmıştır. Bu çalışmalar arasında kesin çözüm yöntemleri kullanan çalışmalar mevcuttur. Al-Yakoob ve ark. (2010) tarafından yapılan çalışmada Kuveyt Üniversitesi'nde yapılan bir uygulama ile sınavların sınıf ve gözetmen ataması sağlanmıştır. Çalışmada iki-aşamalı karışık-tamsayılı programlama yaklaşımı benimsenmiş olup ilk aşamada sınavlar belirli zaman dilimlerine ve sınıflara atanırken ikinci aşamada gözetmenlerin sınavlara atanması gerçekleştirilmiştir. Komijan ve Koupaei (2012) tarafından yapılan çalışmada sınav çizelgeleme problemi için ikili tamsayılı programlama modeli geliştirilmiştir ve bir gerçek hayat problemi üzerinde uygulama yapılarak model test edilmiştir. Bergmann ve ark. (2014) tarafından yapılan çalışmada öğrencilerin ve öğretim elemanlarının istekleri göz önünde bulundurularak sınav çizelgeleme problemi çözümü için hedef programlama modeli geliştirmiştir. Cavdur ve Kose (2016) tarafından yapılan çalışmada dengeli bir sınav programı oluşturmak için bulanık mantık ve 0-1 hedef programlama tabanlı bir yaklaşım kullanılarak sınav çizelgeleme probleminin çözümü sağlanmıştır. Bir başka çalışmada, Abou Kasm ve ark. (2019) üniversite sınav çizelgeleme problemi için tamsayılı programlama modeli önermiş grafik renklendirme yaklaşımı ile probleme çözüm getirmişlerdir. Güler ve ark. (2021) tarafından yapılan çalışmada üniversite sınav çizelgeleme problemi ve danışman atama problemi çözümü için karışık-tamsayılı programlama modeli önerilmiş ve web-tabanlı bir karar destek sistemi geliştirilmiştir. Başka bir çalışmada, Shadkam ve Rajabi (2021) en uygun sınav çizelgelemesine ulaşmak için öğrencilerin sınav aralıklarının maksimize edildiği ve öğrencilerin sınavlara çalışması için yeterli zamanın sağlanmasının amaçlandığı tamsayılı programlama modeli sunmuşlardır.

Sınav çizelgeleme konusu için sezgisel çözüm yöntemlerinin kullanılması da söz konusudur. Buna göre Burke ve ark. (2010) tarafından yapılan çalışmada sınav çizelgeleme problemi çözümü için değişken komşuluk arama algoritması içindeki

komşulukları akıllıca seçen hibrit genetik algoritma kullanılmıştır. Leite ve ark. (2019) tarafından yapılan çalışmada ise sınav çizelgeleme problemine tavlama benzetimi algoritması ile çözüm getirilmiştir. Leite ve ark. (2021) üniversite sınav çizelgeleme probleminin çözümü için hızlı eşik kabulü (fast threshold acceptance) isimli iki aşamalı hibrit bir meta-sezgisel algoritma önermişlerdir. Zhu ve ark. (2022) tarafından yapılan çalışmada COVID-19 pandemi krizi altında çevrim içi sınav çizelgeleme problemi çözümü için yapay arı kolonisi algoritması kullanılmıştır. Başka bir çalışmada, Mujuni (2021) sınav çizelgeleme probleminin çözümü için ayırık bir yarasa algoritması sunulmuştur. Fong ve ark. (2022) tarafından yapılan çalışmada üniversite sınav çizelgeleme problemini çözmek için zombilerin yiyecek arama davranışına dayanan bir zombi hayatta kalma optimizasyonu (Zombie Survival Optimization) uygulanmıştır. Prosad ve ark. (2022) tarafından yapılan çalışmada sınav çizelgeleme konusunda sınavların yapıldığı salon düzenini ve sınav salonu gözetleme sistemini ele alan web-tabanlı bir sistem geliştirilmiştir. Bunun için yazarlar genetik algoritma ve açgözlü algoritma yaklaşımlarını kullanmışlardır. Başka bir çalışmada, Haruna ve ark. (2022) sınav takvimini planlamak için kitle kaynak kullanımı ile rastgele çizelgeleme yöntemi kullanmışlardır. Kullanılan yöntemde insan-makine tabanlı hibrit bir yaklaşım önerilmiş olup önce benzersiz dersler için makine tabanlı rasgele çizelgeleme yapılmakta ardından çizelgelenmemiş dersler için uzmanlar kullanılarak sınav çizelgesi elde edilmektedir. Sınav çizelgeleme koşunda incelenen çalışmalar Çizelge 2.2’de sunulmuştur.

Çizelge 2.2. Literatürde yer alan sınav çizelgeleme çalışmaları

Sınav Çizelgeleme	Kesin Çözüm Yöntemi	Sezgisel Çözüm Yöntemi
Al-Yakoob ve ark. (2010)	✓	
Burke ve ark. (2010)		✓
Komijan ve Koupaei (2012)	✓	
Bergmann ve ark. (2014)	✓	

Çizelge 2.3. Literatürde yer alan sınav çizelgeleme çalışmaları (devam)

Sınav Çizelgeleme	Kesin Çözüm Yöntemi	Sezgisel Çözüm Yöntemi
Cavdur ve Kose (2016)	✓	
Abou Kasm ve ark. (2019)	✓	
Leite ve ark. (2019)		✓
Leite ve ark. (2021)		✓
Güler ve ark. (2021)	✓	
Shadkam ve Rajabi (2021)	✓	
Zhu ve ark. (2022)		✓
Mujuni (2021)		✓
Fong ve ark (2022)		✓
Prosad ve ark. (2022)		✓
Haruna ve ark. (2022)		✓

2.3. COVID-19 Pandemisinde Eğitim-Öğretim ve Sınav Süreçleri

COVID-19 salgın döneminde, dünyadaki hemen hemen tüm eğitim kurumları, öğrenciler için öğretme ve sınav programını sürdürmenin en iyi yolu olarak çevrim içi sistemleri kullanmaktadır (Jia ve He, 2022). Literatürde COVID-19 salgını sürecinde öğrenci görüşlerini ve davranışlarını irdeleyen çalışmalara yer verildiği görülmektedir. Bu çalışmalardan biri Almosa (2021) tarafından ortaya konan, COVID-19 salgını sırasında öğrencilerin çevrim içi öğrenmeye katılımlarına ilişkin durumları araştıran çalışmadır. Çalışmada öğrencilerin süreç hakkında Twitter paylaşımlarını analiz eden yazar öğrencilerin öğrenme katılımının, öğrenme modundaki ani değişimin zorluklarından ve

değerlendirme yöntemlerindeki değişikliklerden etkilendiğini vurgulamaktadır. Buttler ve ark. (2021) tarafından yapılan çalışmada COVID-19 salgını nedeniyle acil uzaktan öğretime geçiş sürecine öğrencilerin tepkisini değerlendirmek için Kanada’da bulunan bir üniversitede anket çalışması yapılmıştır. Yazarlar çalışma sonucunda; öğretim üyesi desteği ve ilgisi, sınav formatının memnuniyeti, yeterli teknolojik kaynakların kullanılması gibi kriterlerin süreci kolaylaştırdığı yorumunu yapmışlardır. Bilen ve Matros (2021) tarafından yapılan çalışmada ise COVID-19 salgını koşullarında yapılan çevrim içi bir sınavda, kopya çekmeye ilişkin kanıtlar sunulmuş olup kopya çekmenin nasıl azaltılacağına dair önerilerde bulunulmuştur. Yazarlara göre öğrencilere sınavlarda daha az zaman verilmesi buna karşın daha basit sorular sorulması ve buna benzer önlemler alınması çevrim içi sınavlarda kopya çekilmesini azaltacaktır. Jia ve He (2022) tarafından yapılan çalışmada yapay zeka teknolojisi kullanılarak akıllı bir çevrim içi gözetmenlik sistemi tasarlanmıştır. Çalışmada tasarlanan bu sistem salgın döneminde Pekin Üniversitesi'nde yapılan bir çevrim içi sınav sırasında uygulanmıştır. Yazarlar çalışmada ortaya koydukları sistem ile öğrencilerin çevrim içi sınavlarda kopya çekmelerini önlemeyi amaçlamaktadırlar. Klijn ve ark. (2022) tarafından yapılan çalışmada COVID-19 salgını sırasında çevrim içi yapılan bir sınavdaki öğrencilerin sınav performansı değerlendirilmektedir. Potu ve ark. (2022) tarafından yapılan çalışmada ise tıp fakültesinde verilen anatomi dersi için dersi hem yüz yüze hem de COVID-19 salgını nedeniyle çevrim içi öğrenen öğrencilerin görüşlerini içeren bir anket çalışması düzenlenmiştir. Çalışmanın sonuçları, öğrencilerin anatomi dersi için çevrim içi öğrenme şeklinden yana olduklarını göstermektedir. Bir başka çalışmada, Gokulkumari ve ark. (2022) çevrim içi sınav sistemine yönelik öğrenci algılarını analiz etmek amacıyla bir anket çalışması yapmışlardır. Çalışma sonucunda yazarlar çevrim içi sınavların uygulanabilirliğini vurgulamaktadırlar.

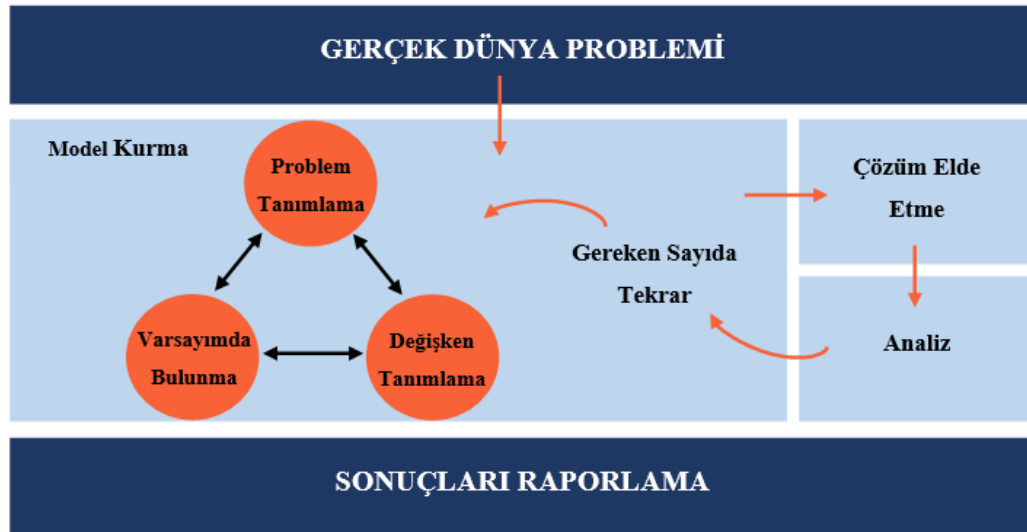
Bu tez çalışmasının literatüre katkısı, günümüzde genellikle acil durumlarda kullanılan ancak gelecekte kullanımının yaygınlaşması öngörülen çevrim içi sınav süreci için dengeli bir sınav çizelgesi oluşturulması ile çalışmada geliştirilen algoritma sayesinde sınavların yapıldığı sunucudaki yük dengesinin bölümler özelinde lokal olarak sağlanmasıdır.

3. MATERYAL ve YÖNTEM

Tez çalışmasında, çevrim içi sınav çizelgeleme problemi ele alınmış olup sınavların yapıldığı sunucudaki yükün dengelenmesi hedefi doğrultusunda dinamik bir yapı geliştirilmiştir. Öncelikle belli kısıtlar altında sınavların en uygun zaman dilimlerine atanması için bir tamsayılı matematiksel programlama modeli geliştirilmiştir. Ardından yük dengeleme kısıtının modele dahil edilmesiyle birlikte, geliştirilen dinamik yapı kullanılarak sunucudaki yük dengesi sağlanmaya çalışılmıştır. Bu sayede, ele alınan durum kapsamında oluşabilecek mazeret sınavlarının önüne geçilebileceği düşünülmektedir.

3.1. Materyal

Matematiksel model, bazı gerçek dünya problemlerinin niteliksel veya niceliksel olarak anlaşılmasını sağlamak ve gelecekteki davranışları tahmin etmek için kullanılan bir sistemin temsidir. Modelleme sürecinin genel yapısı Şekil 3.1’de görüldüğü gibidir. Matematiksel modelleme yinelemeli bir süreçtir. Bu bağlamda, Şekil 3.1’de de ifade edildiği gibi temel adımlar birden çok kez uygulanabilir (Bliss ve ark., 2014).



Şekil 3.1. Matematiksel modelleme süreci (Bliss ve ark., 2014)

Matematiksel programlama modellerinde karar deęişkenlerinin, modellenmekte olan sistemdeki kontrol edilebilir girdileri temsil ettięi söylenebilir. Deęişkenlerin tamsayı deęerleriyle sınırlandırıldığı matematiksel programlama problemlerine tamsayılı programlama problemleri denir. Birçok mühendislik, endüstriyel ve finansal uygulama, tamsayı deęerlerden oluşan kısıtlar içermektedir. Örneęin, bir üretim senaryosunda kesirli deęerler ile üretim yapmak mümkün deęildir (Carter ve Price, 2001). Bu gibi problemlerde tamsayılı programlama kullanılmaktadır.

Tüm karar deęişkenlerinin pozitif tamsayı deęerine sahip olması gereken matematiksel programlama problemlerine genel tamsayılı programlama problemleri denmektedir. Çalışanlar için bir dizi vardiya modeli tanımlanan çizelgeleme problemi bu problem türüne örnek olarak verilebilir (Carter ve Price, 2001). Tamsayılı programlama problemlerinin genel formülasyonu aşağıda sunulmuştur (Schrijver, 1998).

$$\max\{cx \mid Ax \leq b; x \in \mathbb{Z}\} \quad (3.1)$$

Burada A rasyonel matris, b ve c rasyonel vektörlerdir.

Tamsayılı programlama modelinin kapalı formu aşağıdaki gibidir (Chen ve ark., 2011):

$$\max z = \sum_j c_j x_j \quad (3.2)$$

$$\sum_j a_{ij} x_j \leq b_i, \quad \forall i \quad (3.3)$$

$$x_j \in \mathbb{Z}^+ \cup \{0\}, \quad \forall j \quad (3.4)$$

Tüm karar deęişkenleri sadece 0 veya 1 deęerine sahip ise bu problemler ikili (0-1) tamsayılı programlama problemi olarak adlandırılmaktadır. Bu durumda, deęişkenler üzerindeki kısıtlara, ikili kısıtlar denilmektedir. Bu deęişkenler, belirli bir atamanın gerçekleşip gerçekleşmediğini göstermektedir. İkili (0-1) tamsayılı programlama

problemlerine sınav çizelgeleme ve araç rotalama problemleri örnek olarak verilebilir (Carter ve Price, 2001). Örnek bir ikili (0-1) tamsayı programlama modelinin kapalı formu aşağıdaki gibidir (Chen ve ark., 2011):

$$\max z = \sum_j c_j x_j \quad (3.5)$$

$$\sum_j a_{ij} x_j \leq b_i, \quad \forall i \quad (3.6)$$

$$x_j \in \{0,1\}, \quad \forall j \quad (3.7)$$

Tamsayı programlama problemlerinin bir diğer türü ise karışık-tamsayı programlama problemleridir. Bazı değişkenler sürekli değerler alırken bazılarının tamsayı değerler aldığı modellere Karışık-tamsayı programlama modelleri denmektedir. Üretim planlama problemi bu problem türüne örnek olarak verilebilir (Carter ve Price, 2001). Karışık-tamsayı programlama modelinin kapalı formu aşağıdaki gibidir (Chen ve ark., 2011):

$$\max z = \sum_j c_j x_j + \sum_k d_k y_k \quad (3.8)$$

$$\sum_j a_{ij} x_j + \sum_k g_{ik} y_k \leq b_i, \quad \forall i \quad (3.9)$$

$$x_j \geq 0, \quad \forall j \quad (3.10)$$

$$y_k \in \mathbb{Z}^+ \cup \{0\}, \quad \forall k \quad (3.11)$$

Tamsayı programlama problemlerinin çözümü için farklı çözüm yöntemleri mevcuttur. Literatür incelendiğinde bu problem türü çözümünde Dal-Sınır Algoritması (Branch-

Bound Algorithm) ve Kesme Düzlemi Algoritması (Cutting Plane Algorithm) yöntemlerinin sıklıkla kullanıldığı ve başarılı olduğu görülmektedir.

Dal-Sınır Algoritması, tamsayılı programlama problemlerini çözmek için yaygın olarak kullanılan en etkili yöntemler arasında kabul edilmektedir. Bu algoritma, bir problemin yapısı hakkında, amaç fonksiyonu ve kısıtların lineer olması gerektiği varsayımında bulunmaktadır. En basit haliyle, Dal-Sınır Algoritması, zor bir problemi iki veya daha fazla küçük alt probleme bölmenin bir yoludur. Dolayısıyla problem daha kolay hale gelmektedir. Bu alt problemler çözümü kolaylaştırmıyorsa, tekrar dallanma yapılarak problemler daha fazla alt bölüme ayrılmaktadır. Bu işlem, alt problemlerin her biri kolayca çözülene kadar tekrarlanmaktadır. Alt problemlerin her birinin çözülmesi ve bulunan en iyi cevabın seçilmesi orijinal problemin çözülmesine eşdeğer olacak şekilde yapılmaktadır (Carter ve Price, 2001). Bir maksimizasyon problemi için Dal-Sınır Algoritması adımları aşağıda sunulmuştur. Algoritmada kullanılan notasyon tanımlarında bulunan IP tamsayılı programlamayı, LP ise doğrusal programlamayı temsil etmektedir (Chen ve ark., 2011).

S	: sunulan IP problemi
S_{LP}	: S 'nin LP gevşemesi
y_{LP}	: sunulan IP 'nin LP gevşemesine çözüm
z^*	: problemin uygulanabilir bir çözümü x^* için sınırlı amaç değeri
\bar{z}	: sunulan IP sorununun z^* üzerindeki en düşük (en iyi) üst sınırı
\underline{z}	: sunulan IP sorununun z^* üzerindeki en yüksek (en iyi) alt sınırı
S^k	: S probleminin k alt problemi
S_{LP}^k	: k alt probleminin LP gevşemesi
z^k	: S^k 'nin en uygun amaç değeri
\bar{z}^k	: S^k alt probleminin en iyi (en düşük) üst sınırı
\underline{z}^k	: S^k alt probleminin en iyi (en yüksek) alt sınırı
y_{LP}^k	: LP alt probleminin S_{LP}^k en uygun çözümü

Adım 0 (Başlama): Problem S_{LP} çözülür. Uygun bir çözüm bulunamıyorsa problem S 'de uygunsuz olur ve sonlandırılır. Eğer S_{LP} tamsayılı bir çözüm sağlıyorsa problem S

çözülür. Aksi takdirde, problem S_{LP} 'nin çözümü üst sınırı \bar{z} olurken alt sınır \underline{z} ise $-\infty$ olur. S_{LP}^k aktif düğüm listesine yerleştirilir.

Adım 1 (Bir Düğüm Seçme): Aktif liste boşa sonlandırılır. Aksi takdirde, kurallara göre S_{LP}^k ile bir S^k düğümü seçilir.

Adım 2 (Üst Sınır Güncelleme): S_{LP} çözülür ve \bar{z}^k en uygun amaç değerine eşit olur. y_{LP}^k korunur.

Adım 3 (Uygunsuzluk Nedeniyle Budama): Eğer S_{LP}^k 'nin uygun bir çözümü yoksa mevcut düğüm budanır ve Adım 1'e gidilir. Aksi halde Adım 4'e gidilir.

Adım 4 (Sınır Değer Nedeniyle Budama): Eğer $z^k \leq \underline{z}$ ise mevcut düğüm budanır ve Adım 1'e gidilir. Aksi halde Adım 5'e gidilir.

Adım 5 (Alt Sınırın Güncellenmesi ve Optimalite Nedeniyle Budama): Eğer y_{LP}^k tamsayı ise problem S için uygun bir çözüm bulunur. $\underline{z}^k = y_{LP}^k$ olarak kabul edildiğinde $\underline{z}^k > \underline{z}$ ise $\underline{z} = \underline{z}^k$ olur. Mevcut düğüm budanır ve Adım 1'e gidilir. Aksi halde Adım 6'ya gidilir.

Adım 6 (Dallanma): Geçerli düğüm S^k 'dan bir y_j değişkeni seçilir. Böylece S_1^k ve S_2^k alt problemleri aşağıdaki gibi tanımlanır. Bu iki düğümde listeye yerleştirilir ve Adım 1'e gidilir.

$$S_1^k = S^k \cap \{y: y_j \leq \lfloor \bar{y}_j \rfloor\} \quad (3.12)$$

$$S_2^k = S^k \cap \{y: y_j \leq \lceil \bar{y}_j \rceil\} \quad (3.13)$$

Bir diğer çözüm yaklaşımı olan Kesme Düzlemi Algoritması, tamsayılı programlama problemlerini, tamsayı çözümü elde edilene kadar, doğrusal programlama çözümlerini değiştirerek çözmektedir. Bu algorithmada, Dal-Sınır Algoritmasında yapılan uygulanabilir bölgeyi alt bölümlere ayırma işlemi yerine yeni kısıtlamalar eklenerek iyileştirilen tek bir doğrusal programla çalışılmaktadır. Algorithmada, optimal tamsayılı çözüm bulunana kadar doğrusal programlama çözümlerine yeni kısıtlar eklenmektedir (Bradley ve ark., 1977). Kesme Düzlemi Algoritması adımları aşağıda sunulmuştur (Winston ve Goldberg, 2004).

Adım 1: *IP* 'nin *LP* gevşemesi için en uygun tablo bulunur. Bu tabloda tüm değişkenler tamsayı değerler alıyorsa en uygun çözüm bulunmuş olur. Aksi halde Adım 2'ye gidilir.

Adım 2: Bir kesme oluşturmak için kullanılmak üzere eşitliğin sağ tarafında kesirli kısma sahip olan bir kısıt seçilir.

Adım 2a: Adım 2'de tanımlanan kısıt için, sağ taraf değişkeni ve her bir değişkenin katsayısı aşağıdaki gibi tanımlanır:

$$[x] + f, \quad 0 \leq f \leq 1 \quad (3.14)$$

Adım 2b: Kesme oluşturmak için belirlenen kısıtlar şu şekilde yeniden yazılır:

Tamsayı katsayılı tüm terimler = Kesirli katsayılı tüm terimler

Kesme: Kesirli katsayılı tüm terimler ≤ 0

Adım 3: Eklenen kesme ile *LP* gevşemesinin en uygun çözümünü bulmak için Dual simpleks yöntemi kullanılır. Eğer tüm değişkenler optimal çözümde tamsayı değerleri alıyorsa, *IP* için en uygun çözüm bulunmuş olur. Aksi takdirde, sağ taraftaki en kesirli kısıt seçilir ve tabloya eklenmek üzere başka bir kesme oluşturmak için kullanılır. Tüm değişkenlerin tamsayı olduğu bir çözüm elde edilene kadar bu işleme devam edilir. Bu, *IP* için en uygun çözüm olacaktır.

3.2. Yöntem

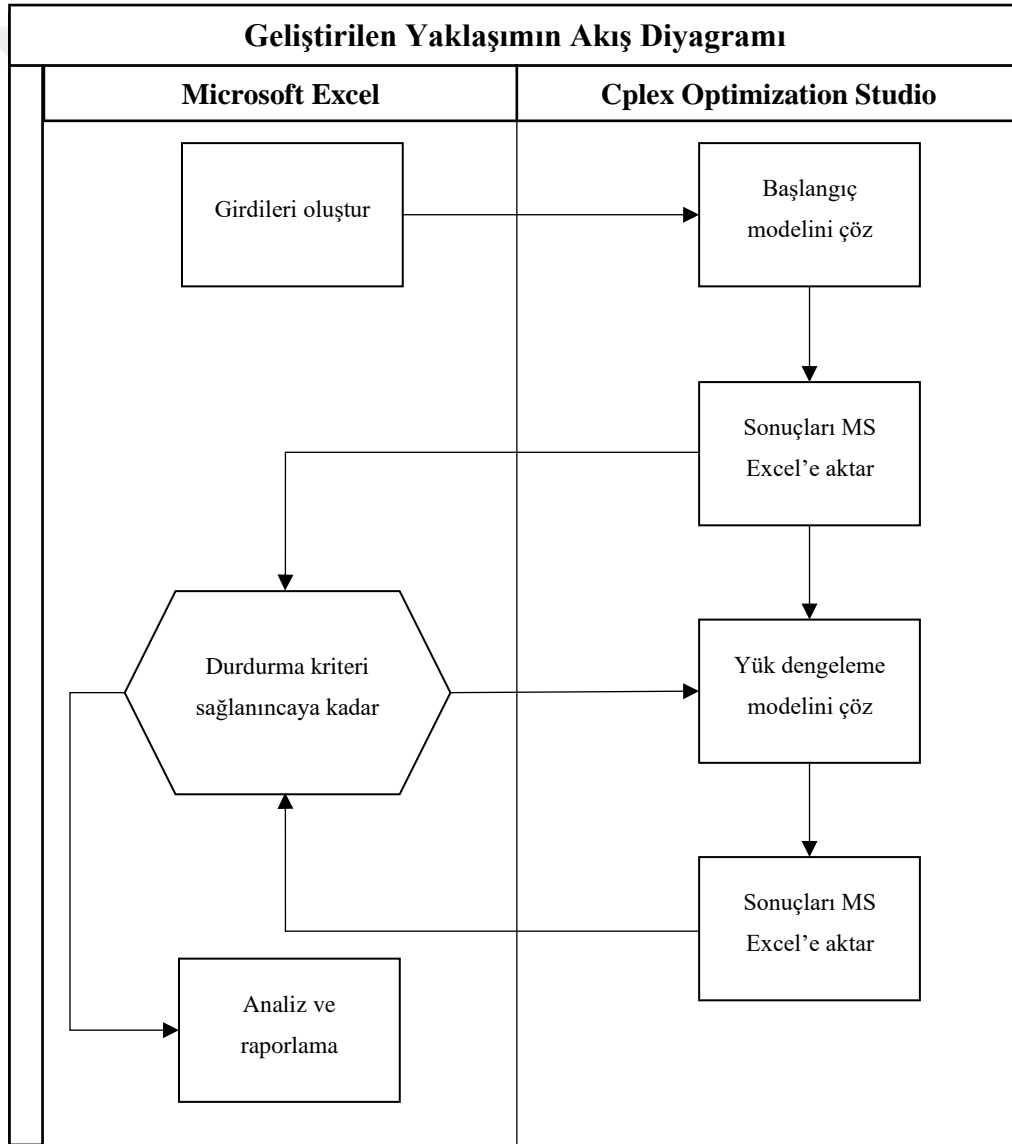
Bu çalışmada, çevrim içi sınavların en uygun zaman dilimlerine atanmasıyla dengeli bir sınav çizelgesi oluşturulması ve böylece sınavların yapıldığı sunucudaki yükün dengelenmesi hedeflenmektedir. Çalışma kapsamında, Bursa Uludağ Üniversitesi Mühendislik Fakültesi 2021-2022 eğitim öğretim yılı güz dönemindeki sekiz adet bölüme ait veriler elde edilmiştir. Bu bölümler, Otomotiv Mühendisliği, İnşaat Mühendisliği, Bilgisayar Mühendisliği, Elektrik-Elektronik Mühendisliği, Çevre Mühendisliği, Endüstri Mühendisliği, Makine Mühendisliği ve Tekstil Mühendisliğidir.

Çevrim içi sınavlarda da yüz yüze yapılan sınavlarda olduğu gibi ders çakışmalarına dikkat edilmesi gerekmektedir. Bir öğrencinin almış olduğu birden fazla dersin sınavının aynı zaman dilimine atanması söz konusu olmamalıdır. Bu durum öğrencilerin sorumlu

oldukları tüm sınavlara girebilmesi açısından önemlidir. Ders çakışmalarının önlenmesi için öncelikle çakışan ders bilgisine ulaşılması gerekir. Bu bağlamda çalışmada, bölümlerin kendi içlerinde, bölümlere ait dersler ile ikili ders kombinasyonları oluşturularak, bu dersleri ortak alan öğrenci sayılarına ulaşılmıştır. Bu durum sekiz bölüm için ayrı ayrı gerçekleştirilmiştir. Böylece her bölüm için bölümlerin kendi içlerinde çakışan ders çiftleri belirlenmiştir.

Tez çalışmasında, uygun bir sınav çizelgesi oluşturulması hedefiyle bir tamsayı programlama modeli geliştirilmiştir. Bu model başlangıç modeli olarak isimlendirilmiştir. Başlangıç modelinin CPLEX Optimization Studio ortamında kodlanarak bölümlere ait veri setleriyle çalıştırılmasıyla bölümler için en uygun ders atamaları sonuçlarına ulaşılmıştır. Çalıştırılan başlangıç modeli sonucunda elde edilen, her bölüm için t anında sınava giren öğrenci sayısı çıktıları Microsoft Excel ortamına aktarılmıştır. Ayrıca ilgili bölüm için ders-dönem-zaman dilimi atama sonuçları da Microsoft Excel ortamına kaydedilmiştir. Çalışmada, sunucudaki yükün dengelemesi amaçlanmaktadır. Bu amaç doğrultusunda öncelikle bölümlerden gelen t anında sınava giren öğrenci sayısı çıktı değerlerinin her t anı bazında toplamaları alınarak her bir zaman dilimi için t anında sunucuya yüklenen toplam yük bilgisine ulaşılmıştır. Bu değerler içerisinden maksimum olan değer seçilerek sisteme yüklenen maksimum yükün ne olduğu ve hangi zaman diliminde yüklendiği bilgilerine ulaşılmıştır. Ardından sisteme yüklenen maksimum yüke hangi bölümün daha fazla etki ettiği saptanmıştır. Daha önce bahsedildiği üzere, maksimum yük değeri sekiz bölümden gelen sınava giren öğrenci sayılarının toplamıyla elde edilmektedir. Bu durumda maksimum yükün olduğu t anında sınava giren maksimum öğrenci sayısına sahip olan bölümün maksimum yüke en fazla etkiyi göstereceği söylenebilir. Çalışmanın hedefi doğrultusunda başlangıç modeline sunucu yükünün dengelenmesi için bir kısıt eklenmiştir. Bu sayede maksimum yükün olduğu t anında sınava giren maksimum öğrenci sayısının belli oranda azaltılmasıyla ilgili t anında ilgili bölüm için sunucudaki yükün dengelenmesi amaçlanmaktadır. Sunucu yükü dengeleme kısıtı dahil edilen model yük dengeleme modeli olarak isimlendirilmiştir. Bahsedilen dinamik yapının uygulanması amacıyla Microsoft Excel Visual Basic for Applications (VBA) ortamında bir algoritma geliştirilmiştir. Kurgulanan dinamik yapı sayesinde bölümler bazında değerler güncellenirken ilgili

bölüm için model çalıştırılarak t anında sunucudaki yük dengesinin sağlanması amaçlanmaktadır. Bu algoritma maksimum iterasyon sayısına ulaşınca kadar iteratif bir şekilde devam ettirilmektedir. Tez kapsamında geliştirilen algoritmanın hedefi doğrultusunda, ilgili bölüm için sunucudaki yük dengesinin sağlanmasıyla sınavlar sırasında sistemde oluşabilecek aksamaların önüne geçilmesi umulmaktadır. Bu sayede, sistemde oluşabilecek aksamalar sonucu öğrencilerin sınavlara girememesi nedeniyle yapılması zorunlu olan mazeret sınavları sayısının da minimize edilebileceği düşünülmektedir. Tez kapsamında uygulanan yöntem akışı Şekil 3.2’de görüldüğü gibidir.



Şekil 3.2. Geliştirilen yaklaşımın akış diyagramı

3.2.1. Tamsayılı programlama modeli

Tez çalışması kapsamında çevrim içi sınav çizelgeleme problemi ele alınmıştır. Bu bağlamda, sekiz bölümden oluşan Mühendislik Fakültesi için dengeli sınav çizelgeleri oluşturulmuş, ardından sınavların yapıldığı sunucudaki yükün dengelenmesi hedefiyle bir algoritma geliştirilmiştir.

Yapılan çalışma kapsamında oluşturulan sınav çizelgeleri 11 günü kapsamaktadır. Her gün sınavları planlamak için dört zaman dilimi mevcuttur. Örneğin, ilk gün planlanabilecek sınavlar için ilk zaman dilimi 9:00'dan 10:45'e kadar, ikinci zaman dilimi 11:00'den 12:45'e kadar, üçüncü zaman dilimi 13:00'den 14:45'e kadar ve dördüncü zaman dilimi 15:00'den 16:45'e kadar olmaktadır. Bölümlere ait ders sayıları, n_I , her bölüm için değişiklik göstermektedir. Dönem sayısı, n_K , her bölüm için aynı olup 1., 2., 3. veya 4. sınıf (1., 3., 5. veya 7. dönem) olmak üzere 4 dönemi kapsamaktadır. Benzer şekilde sınavların yapılabileceği zaman dilimi sayısı, n_T , her bölüm için aynı olup 11 adet sınav günü için bu değer 44 olmaktadır. Bu bilgiler ışığında sınavlar için en uygun zaman çizelgesinin oluşturulması hedeflenmektedir. Bu hedef doğrultusunda geliştirilen tamsayılı programlama modelinin bileşenleri aşağıda sunulmuştur.

İndisler:

i	: ders indisi	$i = 1, \dots, n_I, i \in I$
k	: dönem indisi	$k = 1, \dots, n_K, k \in K$
t	: zaman dilimi indisi	$t = 1, \dots, n_T, t \in T$
u	: zaman dilimi indisi (her gün için)	$u = 0, \dots, 3, u \in U \subset T$

Parametreler:

T_F	: her günün ilk zaman dilimine karşılık gelen zaman dilimlerinin kümesi, örneğin 1, 5, 9 gibi
$I_0: \{(i_1, i_2) i_1, i_2 \in I, i_1 \neq i_2\}$: çakışan ders çiftlerinin kümesi

n_{max} : bir günde maksimum sınav sayısı
 P_i : i . dersin sınavına girecek öğrenci sayısı
 P_{max} : herhangi bir anda sınava girebilecek maksimum öğrenci sayısı

Değişkenler:

x_{ikt} : k dönemindeki i dersi t zaman dilimine atanırsa 1, aksi halde 0
 P_t : t anında sınava giren öğrenci sayısı

Amaç Fonksiyonu:

$$\min z = \sum_{t=1}^{n_T} P_t \quad (3.15)$$

Kısıtlar:

$$\sum_{t=1}^{n_T} x_{ikt} = 1, \quad \forall i \in I, k \in K \quad (3.16)$$

$$\sum_{i=1}^{n_I} \sum_{k=1}^{n_K} \sum_{u=0}^3 x_{i,k,t+u} \leq n_{max}, \quad t \in T_F \quad (3.17)$$

$$\sum_{i=1}^{n_I} x_{ikt} \leq 1, \quad \forall k \in K, \forall t \in T \quad (3.18)$$

$$\sum_{k=1}^{n_K} x_{i_1,k,t} + \sum_{k=1}^{n_K} x_{i_2,k,t} \leq 1, \quad \forall t \in T, \forall (i_1, i_2) \in I_0 \quad (3.19)$$

$$\sum_{i=1}^{n_I} \sum_{k=1}^{n_K} x_{ikt} P_i \leq P_t, \quad \forall t \in T \quad (3.20)$$

$$P_t \leq P_{max}, \quad \forall t \in T \quad (3.21)$$

$$x_{ikt} \in \{0,1\}, \quad \forall i, k, t \quad (3.22)$$

Sunulan modelde denklem (3.15) ile amaç fonksiyonu verilmiştir. Amaç fonksiyonunda t anında sınava giren öğrenci sayılarının toplamı minimize edilmektedir. Denklem (3.16) ile her sınavın bir zaman dilimine atanması sağlanmaktadır. Denklem (3.17) ile bir günde en fazla bir gündeki maksimum sınav sayısı kadar sınav yapılabilmesi sağlanmaktadır. Bu denklemde bulunan n_{max} değeri hesaplanırken bölümlerin ders sayılarının sınav takvimindeki gün sayısına bölümünün yukarı yuvarlanmasıyla ulaşılan değer bir fazlası alınmıştır. Bu değer bölümlere ait ders sayılarına bağlı olduğundan her bölüm için değişiklik göstermektedir. Denklem (3.18) ile aynı döneme ait derslerin aynı zaman dilimine atanması önlenmektedir. Denklem (3.19) ders çakışmalarının ele alındığı kısıttır. Bu denklem ile aynı öğrencilerin girdiği derslerin sınavları farklı zaman dilimlerine atanmaktadır. Bir başka deyişle, bir veya daha fazla öğrenci her iki dersi de alıyorsa bu derslerin aynı zaman dilimine atanmasının önüne geçilmektedir. Denklem (3.20) ile her bir zaman dilimine atanan sınava giren öğrenci sayısının t anında sınava giren öğrenci sayısını aşmaması sağlanmaktadır. Denklem (3.21) ile t anında sınava giren öğrenci sayısının herhangi bir anda sınava girebilecek maksimum öğrenci sayısını aşmaması sağlanmaktadır. Burada P_{max} değerine ulaşmak için tüm bölümlerdeki en yüksek öğrenci sayısına sahip dersi alan öğrenci sayısı bir üst yüzlüğe yuvarlanmıştır. Denklem (3.22) ise genel işaret kısıttır.

Yukarıda sunulan model başlangıç modeli olup geliştirilen algoritmanın 0. iterasyonu için uygulanmaktadır. Mühendislik Fakültesi bünyesindeki sekiz bölüm için düzenlenen veri setleri kullanılarak yukarıdaki model her bir bölüm için CPLEX Optimization Studio'da çözülmüştür. Elde edilen P_t değerleri Microsoft Excel ortamına kaydedilmiştir. Ardından, 44 adet zaman dilimi için her bir zaman diliminde sunucuya yüklenen toplam yük, sekiz bölümün P_t değerlerinin toplanmasıyla elde edilmiştir. Elde edilen toplam yük

değerlerinin maksimum olanının bulunmasıyla 44 zaman dilimi içerisinde sunucuya yüklenen maksimum yük değerine ulaşılmıştır. Maksimum yük değerinin sekiz bölümün P_t değerlerinin toplamıyla elde edildiği bilinmektedir. Bu bölümler içerisinde maksimum P_t değerine sahip olan bölümün maksimum yük değerine en fazla etkiyi göstereceği söylenebilir. Bu ifade aşağıdaki gibi tanımlanabilir:

$P_{t_{max}}$: maksimum yükün olduğu t anında sınava giren maksimum öğrenci sayısı

Geliştirilen algoritma, sunucudaki maksimum yükün olduğu t anında, bu yüke en fazla etki eden bölüme ait sınava giren öğrenci sayısının azaltılmasıyla sunucudaki yük dengesinin sağlanmasını hedeflemektedir. Bu durumda $P_{t_{max}}$ değerinin belli oranda azaltılmasıyla bu hedefe ulaşılabileceği söylenebilir. Bu bağlamda, başlangıç modeline denklem (3.23) ile sunulan yük dengeleme kısıtı eklenerek dinamik yapı oluşturulmuştur.

$$P_t \leq P_{t_{max}} - 1, \quad \forall t \in T \quad (3.23)$$

Denklem (3.23)'te sunulan kısıt ile maksimum yükün olduğu t anında sınava giren maksimum öğrenci sayısının bir birim azaltılmasıyla ilgili bölümün sunucuda oluşturduğu yükün azaltılması hedeflenmektedir. Böylelikle, denklemin sağ tarafı ile elde edilen değer t anında sınava giren öğrenci sayısını aşmayacağı da garanti edilmektedir.

3.2.2. Geliştirilen algoritmanın tasarımı

Tez çalışmasında, çevrim içi sınavların yapıldığı sunucudaki yükün dengelenmesi hedefiyle Microsoft Excel VBA ile bir algoritma geliştirilmiştir. Geliştirilen algoritma Mühendislik Fakültesi bünyesinde bulunan sekiz bölüm için uygulanmıştır. Uygulama sonuçlarının kaydedildiği tablonun yapısı Şekil 3.3'te görüldüğü gibidir. Tablonun ilk sütununda iterasyon sayısına, ikinci sütununda maksimum yük değerine, üçüncü sütununda maksimum yükün olduğu zaman dilimi bilgine yer verilmiştir. Takip eden sütunlarda bölümler özelinde P_t değerleri yer alırken son iki sütunda sırasıyla $P_{t_{max}}$ değeri ve maksimum yüke neden olan bölüm indisi yer almaktadır. Hesapla butonuna

basıldığında algoritma istenilen iterasyon kadar çalıştırılabilmektedir. Temizle butonuna basıldığında ise model çalıştığında ulaşılan sonuçlar ve algoritma sonuçlarını kaydetmek için tasarlanan tablonun içeriği temizlenebilmektedir. Algoritma her çalıştığında ilgili bölüm için t ve $P_{t_{max}}$ değeri dinamik bir veri kümesine aktarılarak değerler hafızada tutulmaktadır. Yük dengeleme kısıtı için bu değerler bahsedilen dinamik veri kümesinden alınmaktadır.

İterasyon	Maksimum Yük	Maksimum Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maksimum Yüke Neden Olan Bölüm
			Otomotiv Mühendisliği -1-	İnşaat Mühendisliği -2-	Bilgisayar Mühendisliği -3-	Elektrik Elektronik Mühendisliği -4-	Çevre Mühendisliği -5-	Endüstri Mühendisliği -6-	Makine Mühendisliği -7-	Tekstil Mühendisliği -8-		
0	162	3	162	-	-	-	-	-	-	-	162	1
0	175	33	-	175	-	-	-	-	-	-	175	2
0	192	41	-	-	192	-	-	-	-	-	192	3
0	196	10	-	-	-	196	-	-	-	-	196	4
0	140	1	-	-	-	-	140	-	-	-	140	5
0	163	42	-	-	-	-	-	163	-	-	163	6
0	191	42	-	-	-	-	-	-	191	-	191	7
0	183	16	-	-	-	-	-	-	-	183	183	8
1	873	10	97	114	137	196	26	55	157	91	196	4
2	840	24	108	60	0	162	80	157	171	102	171	7
3	845	31	70	130	111	194	32	105	134	69	194	4
4	812	27	95	133	145	104	44	43	158	90	158	7
5	840	43	72	112	143	138	64	84	191	36	191	7
6	830	34	153	147	0	84	47	139	171	89	171	7
7	814	24	108	60	0	162	80	157	145	102	162	4
8	798	27	95	133	145	104	44	43	144	90	145	3
9	902	33	86	175	156	183	0	100	123	79	183	4
10	890	34	153	147	102	84	47	139	129	89	153	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮



Şekil 3.3. Algoritma sonuçlarının kaydedildiği tablonun yapısı

Algoritmanın 0. iterasyonunda başlangıç modelinin sekiz bölüm için CPLEX Optimization Studio’da çalıştırılmasıyla sonuçlara ulaşılmıştır. Ulaşılan sonuçlar Microsoft Excel ortamına kaydedilmiştir. Kaydedilen sonuçlardaki ilgili değerler Şekil 3.3’te görüldüğü gibi yeşil renk ile vurgulanmış, 0. iterasyonu temsil eden bölgeye aktarılmıştır. İlerleyen iterasyonlarda $P_{t_{max}}$ değerine sahip bölüm için modelin çalıştırılmasıyla elde edilen değerler tabloya aktarılmıştır. Geliştirilen algoritmanın adımları aşağıdaki gibidir.

Adım 0a: Her bir bölüm için başlangıç modeli çözülür.

Adım 0b: Sonuçlar kaydedilir.

Adım 0c: Kaydedilen sonuçlar 0. iterasyon kapsamında tabloya aktarılır.

Adım 1: Yük dengeleme kısıtı matematiksel programlama modeline eklenir.

Adım 2a: Maksimum yük değeri, maksimum yükün oluştuğu zaman dilimi (t), maksimum yükün oluştuğu zaman diliminde bölümlere ait sınava giren öğrenci sayıları

(P_t) , maksimum yükün oluřtuđu zaman diliminde sınava giren maksimum öğrenci sayısı $(P_{t_{max}})$ ve maksimum yüke neden olan bölüme ait indis saptanır.

Adım 2b: Adım 2a'da saptanan deđerler tabloya aktarılır.

Adım 2c: Tablodaki deđerler dođrultusunda maksimum yüke neden olan bölüm için yük dengeleme modeli çözülür.

Adım 2d: Elde edilen sonuçlar ilgili bölüm için güncellenir.

Adım 3: Durdurma kriteri sađlanıncaya (maksimum iterasyona sayısına ulařıncaya) kadar adım 2a'ya dönülür. Aksi durumda, algoritma sonlandırılır.



4. BULGULAR ve TARTIŞMA

4.1. Veri Setleri

Tez çalışması kapsamında, Mühendislik Fakültesi bünyesindeki sekiz bölüme ait dersler için çevrim içi sınav çizelgeleme problemine çözüm aranmıştır. Çözüm yaklaşımı olarak öncelikle tamsayılı programlama modeli geliştirilmiş olup modelin CPLEX Optimization Studio ortamında çalıştırılmasıyla bölümler için en uygun sınav çizelgelerine ulaşılmıştır. Ardından geliştirilen algoritma ile dinamik bir yapıya ulaşılmış, belli bir zaman diliminde belli bir bölüm için sınavların yapıldığı sunucudaki yük dengesi sağlanmaya çalışılmıştır.

Çalışmada ele alınan problemin çözümü için Bursa Uludağ Üniversitesi Mühendislik Fakültesi bünyesindeki sekiz bölümün 2021-2022 eğitim öğretim yılı güz dönemine ait verileri kullanılmıştır. Bu veriler, derslere ait ders kodlarını ve derslere kaydolun öğrenci bilgilerini içermektedir. Her ders kendisine ait benzersiz ders koduna sahiptir. Aynı ders farklı bölümlerde mevcut olabilmekte ancak ilgili derse ait benzersiz ders kodları bölümlere göre değişiklik göstermektedir. Bahsedilen verilere birtakım verilerin daha eklenmesi ve gerekli düzenlemelerin yapılmasıyla geliştirilen modelde kullanılmak üzere her bölüm için veri setleri hazırlanmıştır. Hazırlanan veri setlerinin içeriği Çizelge 4.1’de görüldüğü gibidir.

Çizelge 4.1. Problem veri setleri içeriği

Veri Setleri	n_I	n_K	n_T	u	n_{max}	Gün Sayısı	P_{max}
1	40	4	44	3	5	11	200
2	52	4	44	3	6	11	200
3	25	4	44	3	4	11	200
4	62	4	44	3	7	11	200
5	58	4	44	3	7	11	200
6	53	4	44	3	6	11	200
7	78	4	44	3	9	11	200
8	73	4	44	3	8	11	200

Çizelge 4.1’de sunulan veriler haricinde veri setleri içerisinde bulunan başka veriler de mevcuttur. Bunlardan bir tanesi T_F kümesi olup bu küme tüm veri setlerinde 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41 değerlerini içermektedir. Bu değerler oluşturulan çizelgedeki sınav yapılacak her günün ilk zaman dilimine karşılık gelen zaman dilimlerini ifade etmektedir. Bunun yanı sıra, veri setlerinde bulunan P_i değerleri ise her veri setinde değişiklik göstermekte olup n_i değerlerine bağlı olarak dersleri alan öğrenci sayılarını içermektedir. Her veri setinde değişiklik gösteren bir diğer veri ise derslerin ait oldukları dönem bilgisini içeren sömestr matrisidir. Her bölüm için dersler 1., 2., 3. veya 4. sınıfa (1., 3., 5. veya 7. döneme) ait olabilmektedir. 0-1’lerden oluşan sömestr matrisinde dersler ait oldukları dönem bilgisine göre 0 veya 1 değerini almaktadır. Veri setlerinde bulunan bir diğer matris ise ders çakışmalarını içeren çakışma matrisidir. Bu matris bölümlere ait ikili ders kombinasyonlarını içermektedir. Sömestr matrisinde olduğu gibi 0-1’lerden oluşan çakışma matrisinde, dersleri ortak alan öğrenci olup olmamasına göre ders kombinasyonları 0 veya 1 değerini almaktadır. Tüm bölümler için bölümlerdeki ders kombinasyonlarına göre farklılık gösteren çakışma matrisleri, bölümlere ait dersler ve bu dersleri alan öğrenci bilgisini içeren veriler kullanılarak MATLAB ortamında yazılan bir algoritma ile elde edilmiştir. Bu algoritma ile sekiz bölüm için ayrı ayrı bu bölümlere ait tüm ders ikililerine göre çakışma matrisleri oluşturulmuştur.

Problemin çözümü için geliştirilen matematiksel programlama modelinin yukarıda bahsedilen veri setleri ile CPLEX Optimization Studio’da çalıştırılmasıyla ilgili bölüm için ders-dönem-zaman dilimi atama sonuçlarına ve t anında sınava giren öğrenci sayısı bilgisine ulaşılmıştır. Ardından geliştirilen algoritmanın 300 iterasyon boyunca çalıştırılmasıyla ilgili zaman diliminde ilgili bölüm için sunucudaki yük dengesi sağlanmaya çalışılmıştır.

Çalışma kapsamında, Mühendislik Fakültesi bünyesinde bulunan bölümler İngilizce isimlerinin baş harfleri sıralamasına göre 1’den 8’e kadar numaralandırılmıştır. Buna göre, Otomotiv Mühendisliği 1, İnşaat Mühendisliği 2, Bilgisayar Mühendisliği 3, Elektrik Elektronik Mühendisliği 4, Çevre Mühendisliği 5, Endüstri Mühendisliği 6, Makine Mühendisliği 7 ve Tekstil Mühendisliği 8 olarak numaralandırılmıştır. İlerleyen kısımlarda bölümler bu numaralar ile anılacaktır.

4.2. Dinamik Yapı ile Sınav Çizelgeleme Süreci

Tez çalışması kapsamında, çevrim içi sınavların yapıldığı sunucudaki maksimum yükün olduğu zaman diliminde sınava giren maksimum öğrenci sayısının belli oranda azaltılmasıyla ilgili bölüm için sunucu yükünün dengelenmesi hedeflenmiştir. Sunucu yükünün dengelenmesi sekiz adet bölüm için bölümlerin lokal olarak ele alınmasıyla sağlanmaya çalışılmıştır. Çalışmanın hedefi doğrultusunda, geliştirilen algoritma Microsoft Excel ortamında 300 iterasyon boyunca çalıştırılmış olup elde edilen değerler çizelge halinde tutulmuştur. Algoritmanın 300 iterasyon boyunca çalıştırılmasıyla ulaşılan sonuçları içeren çizelge EK 1’de sunulmuştur.

Tüm iterasyonlar için belli bir zaman diliminde sunucuda oluşan maksimum yük değerleri EK 1’de ki çizelgenin ikinci sütununda görüldüğü gibidir. 300 iterasyon için bu değerler içerisinde en yüksek maksimum yük değeri 108. iterasyonda elde edilen 937’dir. Bu değer 11. zaman diliminde ortaya çıkmış olup bu değeri oluşturan bölümlere ait sınava giren öğrenci sayıları ise şöyledir; Otomotiv Mühendisliğinden 135 öğrenci, İnşaat Mühendisliğinden 67 öğrenci, Bilgisayar Mühendisliğinden 167 öğrenci, Elektrik Elektronik Mühendisliğinden 199 öğrenci, Çevre Mühendisliğinden 52 öğrenci, Endüstri Mühendisliğinden 156 öğrenci, Makine Mühendisliğinden 124 öğrenci ve Tekstil Mühendisliğinden 37 öğrenci. Benzer şekilde, 300 iterasyon için maksimum yük değerleri içerisinde en düşük maksimum yük değerinin ise 103. iterasyonda elde edilen 724 olduğu görülmektedir. Bu değere 25. zaman diliminde ulaşılmıştır. 724 değerini oluşturan bölümlere ait sınava giren öğrenci sayıları ise şöyledir; Otomotiv Mühendisliğinden 25 öğrenci, İnşaat Mühendisliğinden 152 öğrenci, Bilgisayar Mühendisliğinden 77 öğrenci, Elektrik Elektronik Mühendisliğinden 71 öğrenci, Çevre Mühendisliğinden 96 öğrenci, Endüstri Mühendisliğinden 118 öğrenci, Makine Mühendisliğinden 84 öğrenci ve Tekstil Mühendisliğinden 93 öğrenci. Daha önce bahsedildiği üzere, geliştirilen matematiksel modelin bölümlere ait veri setleri ile çalıştırılmasıyla bölümler için ders-dönem-zaman dilimi atamalarına ulaşılmaktadır. Ulaşılan bu atama sonuçları ışığında, 300 iterasyon sonuçları içerisinde en yüksek maksimum yük değeri olan 937 değerinin ve en düşük maksimum yük değeri olan 724 değerinin meydana geldiği ders-dönem-zaman dilimi atamalarını içeren çizelgeler sekiz

bölüm için ayrı ayrı oluşturulmuştur. Oluşturulan çizelgeler sırasıyla, EK 2 ve EK 3'te sunulmuştur. Sunulan çizelgeler bölümler özelinde sınavların yapıldıkları günü, sınavların atandıkları zaman dilimini, t anında sınava giren öğrenci sayısını ve derslerin ait oldukları dönem bilgisini içermektedir. Bununla beraber çizelgelerde atama sonuçları bölümlere ait ders kodları kullanılarak ifade edilmiştir. Ders kodlarının sağ tarafında parantez içerisinde yer alan değerler derslere kaydolun öğrenci sayılarını ifade etmektedir. 300 iterasyon boyunca elde edilen maksimum yük değerlerinden en düşük maksimum yük değeri olan 724 değerini içeren çizelgelerin en iyi çizelgeler olduğu söylenebilir. Benzer şekilde 300 iterasyon boyunca elde edilen maksimum yük değerleri içerisinde en yüksek maksimum yük değeri olan 937 değerini içeren çizelgelerin en kötü çizelgeler olduğu söylenebilir. Bu durumda 724 değerini içeren sekiz adet bölüm için oluşturulan çizelgelerin tercih edilmesiyle çevrim içi sınavların yapıldığı sunucudaki yük dengesinin en makul düzeyde sağlanacağı söylenebilir. Dolayısıyla sınavlar için bu çizelgelerin tercih edilmesi daha doğru olacaktır.

Algoritmanın 300 iterasyon boyunca çalıştırılmasıyla elde edilen sonuçların analiz edilebilmesi için sonuçlar içerisindeki bazı parametrelerin tekrar sayılarına bakılmış olup ilerleyen kısımlarda en fazla tekrar eden değerler üzerinden örnekler verilmiştir. Bu şekilde yapılan analizin daha açık ve anlaşılır olacağı düşünülmüştür.

Geliştirilen algoritmanın çalıştırılmasıyla, maksimum yükün oluştuğu zaman diliminde sınava giren maksimum öğrenci sayısına ($P_{t_{max}}$) sahip bölüme ait veri setiyle matematiksel model çalıştırılarak ilgili sonuçlara ulaşılmaktadır. Bu bölümün ilgili zaman diliminde meydana gelen maksimum yük üzerinde en fazla etkiye sahip olduğu söylenebilir. EK 1'de verilen algoritmanın 300 iterasyon için çalıştırılmasıyla elde edilen sonuçlar ele alınarak maksimum yük üzerinde en fazla etkiye sahip olan bölümlerin tekrar sayılarına ulaşılmıştır. Bunun yanı sıra 300 iterasyon içerisinde maksimum yükün oluştuğu zaman dilimlerinin de tekrar sayılarına ulaşılmıştır. Sekiz bölüm ve 44 zaman dilimi için ulaşılan tekrar sayılarına göre en çok tekrar eden ilk üç değer Çizelge 4.2'de görüldüğü gibidir.

Çizelge 4.2. 300 iterasyon sonuçlarına göre tekrar sayıları

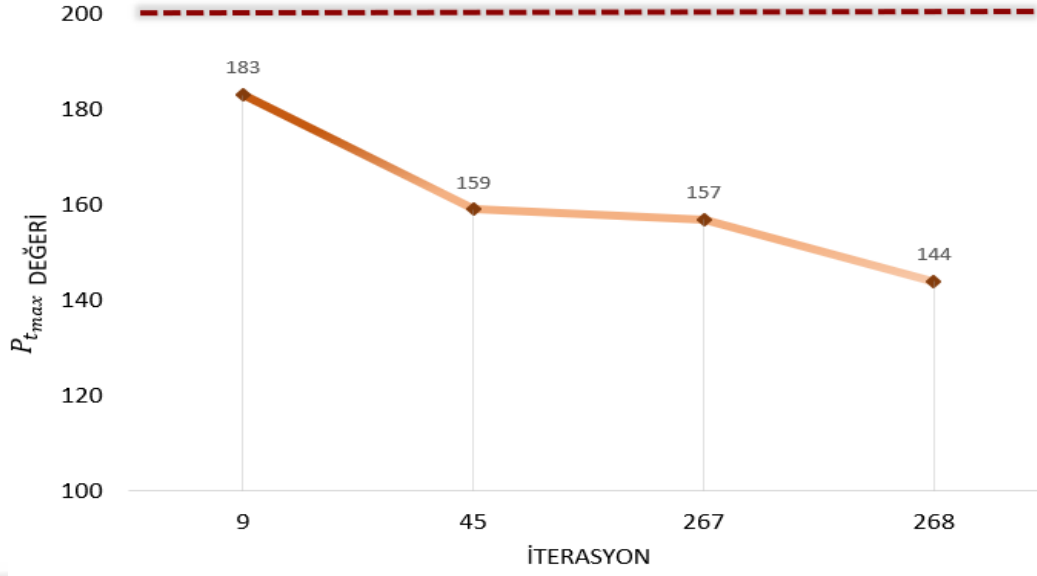
Maksimum Yük Üzerinde En Fazla Etkiye Sahip Olan Bölüm	300 İterasyon İçerisindeki Tekrar Sayısı	Maksimum Yükün Oluştığı Zaman Dilimi	300 İterasyon İçerisindeki Tekrar Sayısı
4 (Elektrik Elektronik Mühendisliği)	63	11	14
6 (Endüstri Mühendisliği)	34	20	12
7 (Makine Mühendisliği)	128	25	15

Çalışma kapsamında, en çok tekrar eden bölümlere ve zaman dilimlerine göre birtakım analizler yapılarak geliştirilen algoritmanın performansı ortaya konulmaya çalışılmıştır. Buna göre ele alınacak ilk performans parametresi $P_{t_{max}}$ değerleridir. $P_{t_{max}}$ değeri, önceki bölümde de belirtildiği gibi maksimum yükün olduğu belli bir zaman dilimindeki sınava giren maksimum öğrenci sayısını ifade etmektedir. Bu performans parametresinin değerlendirilmesi için 300 iterasyon sonuçlarındaki tekrar sayılarına göre, maksimum yük üzerinde en fazla etkiye sahip bölümlerden ilk üçü olan 4., 6. ve 7. Bölüme karşılık gelen $P_{t_{max}}$ değerleri ele alınmıştır. Bu değerler, bahsedilen üç bölüm için seçilen ilgili zaman dilimleri üzerinden örnekler verilerek incelenecektir.

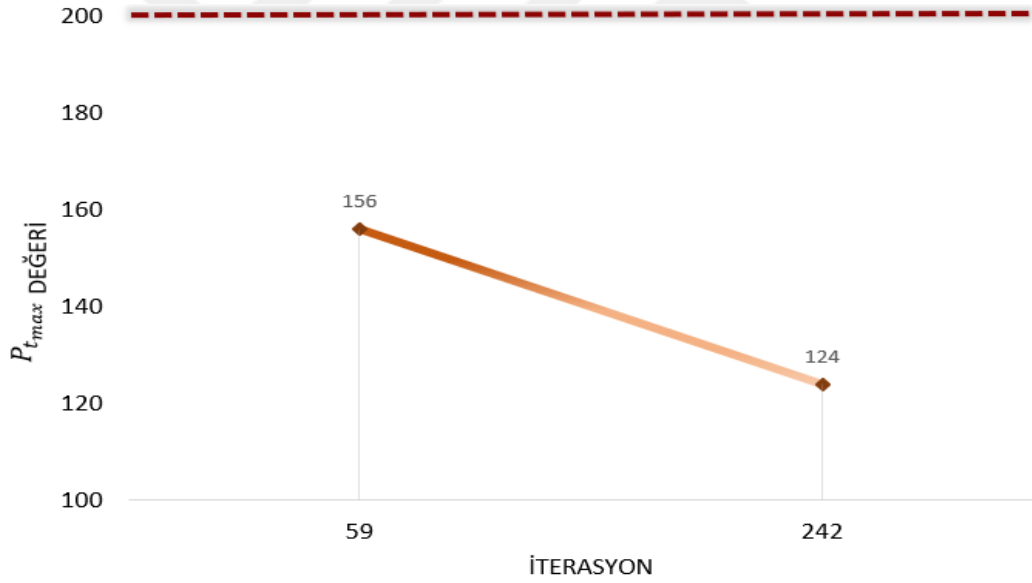
Örneğin; 300 iterasyon boyunca elde edilen sonuçlarda, 4. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 33. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri incelenmiştir. Bu inceleme kapsamında oluşturulan grafiğin yer aldığı Şekil 4.1'e bakıldığında 4. bölüm için 33. zaman diliminin 9., 45., 267. ve 268. iterasyonlarında görüldüğü söylenebilir. Şekilde de görüleceği üzere 9. iterasyonda $P_{t_{max}}$ değeri 183, 45. iterasyonda $P_{t_{max}}$ değerleri 159, 267. iterasyonda $P_{t_{max}}$ değerleri 157, 268. iterasyonda $P_{t_{max}}$ değerleri 144 olmaktadır. Buna göre iterasyon sayısı arttıkça $P_{t_{max}}$ değerlerinin azaldığı gözlemlenmektedir. Bu da çalışmanın hedefi doğrultusunda beklenen bir durum olmaktadır. Bunun yanı sıra, $P_{t_{max}}$ değerlerinin P_{max} değerine karşılık gelen 200 değerini aşmadığı görülmektedir. Bu durumun da geliştirilen matematiksel modeldeki (3.21) numaralı denklem ile verilen kısıt sayesinde beklenen bir durum olduğu söylenebilir.

Benzer şekilde, 300 iterasyon boyunca elde edilen sonuçlarda, 6. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 44. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri incelenmiştir. İnceleme sonucunda oluşturulan grafik Şekil 4.2’de sunulmuştur. Şekil 4.2’ye bakıldığında 6. bölüm için 44. zaman diliminin 59. ve 242. iterasyonda görüldüğü anlaşılmaktadır. Buna göre, 59. iterasyonda $P_{t_{max}}$ değeri 156 iken 242. iterasyonda $P_{t_{max}}$ değerleri 124 olmaktadır. Bundan önce verilen örnekte olduğu gibi bu örnekte de iterasyon sayısı arttıkça $P_{t_{max}}$ değerlerinin azaldığı gözlemlenirken $P_{t_{max}}$ değerlerinin P_{max} değerine karşılık gelen 200 değerini aşmadığı görülmektedir.

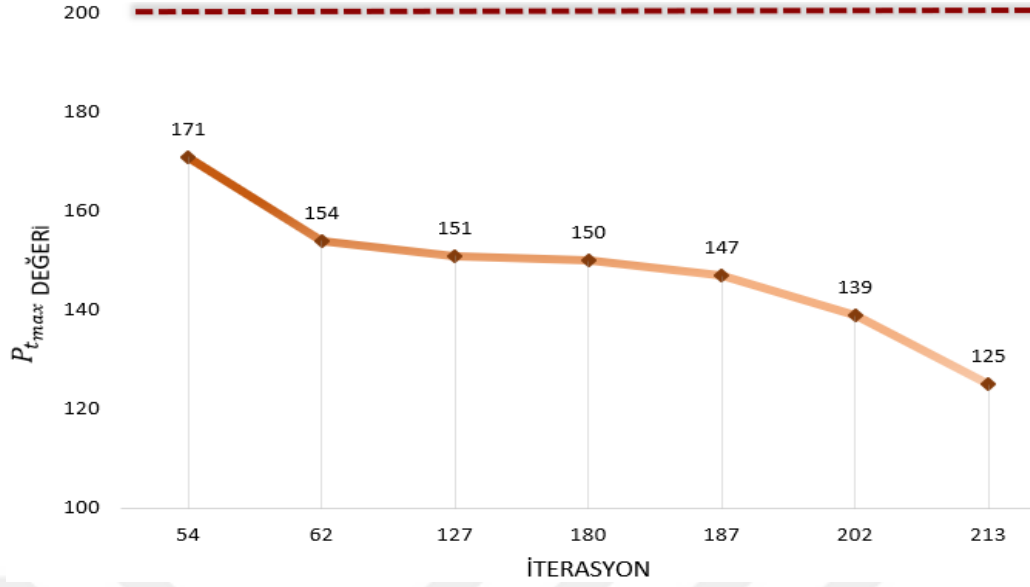
Son olarak 300 iterasyon boyunca elde edilen sonuçlarda, 7. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 25. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri ele alınarak inceleme yapılmıştır. Sonuçları içeren grafiğin yer aldığı Şekil 4.3’e bakıldığında 7. bölüm için 25. zaman diliminin 54., 62., 127., 180., 187., 202. ve 213. iterasyonlarında görüldüğü söylenebilir. Şekil 4.3’te de görüleceği üzere 54. iterasyonda $P_{t_{max}}$ değeri 171, 62. iterasyonda $P_{t_{max}}$ değerleri 154, 127. iterasyonda $P_{t_{max}}$ değerleri 151, 180. iterasyonda $P_{t_{max}}$ değerleri 150, 187. iterasyonda $P_{t_{max}}$ değerleri 147, 202. iterasyonda $P_{t_{max}}$ değerleri 139 ve 202. iterasyonda $P_{t_{max}}$ değerleri 125 olmaktadır. Burada da önceki örneklerde olduğu gibi çalışmanın hedefi doğrultusunda iterasyon sayısı arttıkça $P_{t_{max}}$ değerlerinin azaldığı yorumu yapılabilir. Ayrıca bu örnekte de $P_{t_{max}}$ değerlerinin P_{max} değerini aşmadığı görülmektedir.



Şekil 4.1. 4. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 33. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri



Şekil 4.2. 6. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 44. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri



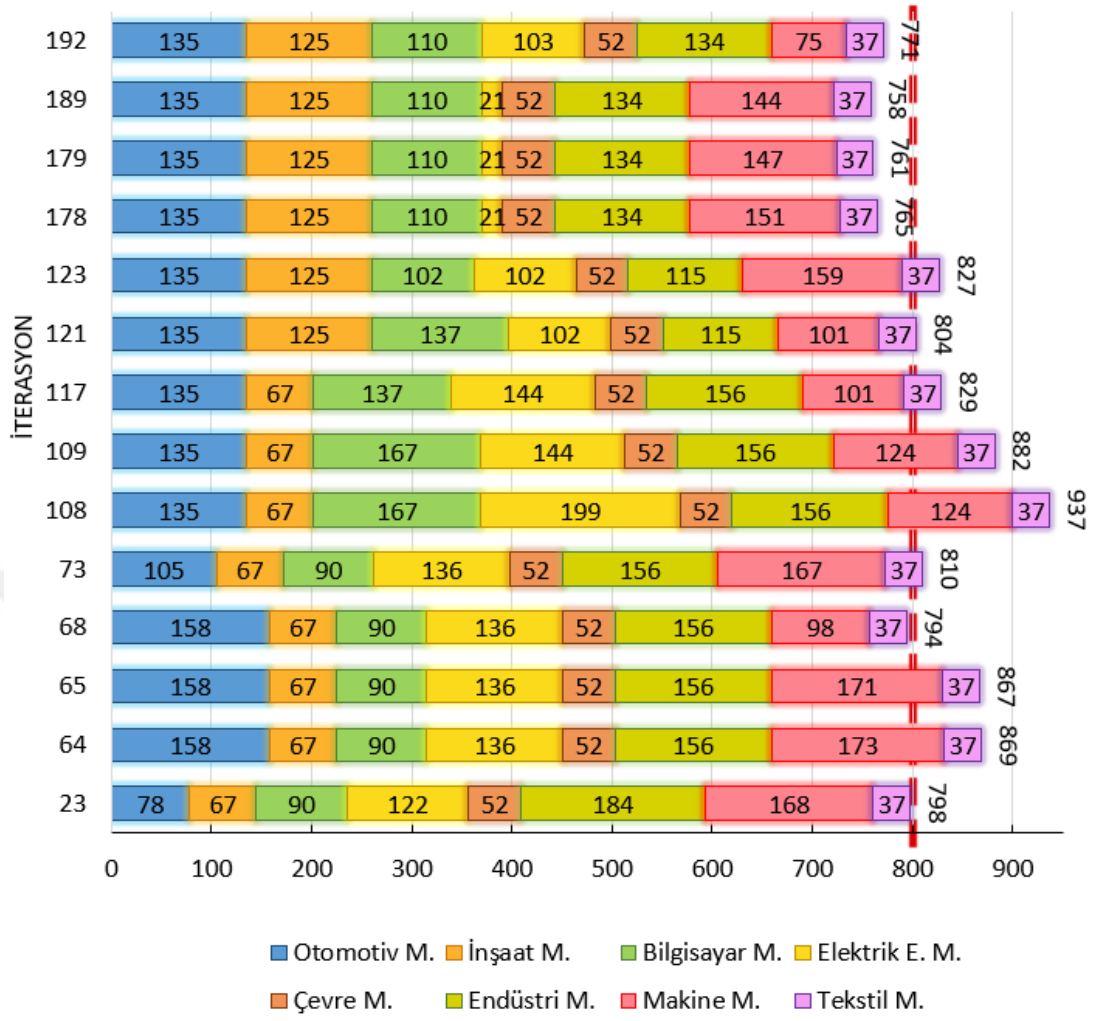
Şekil 4.3. 7. bölümün maksimum yük üzerinde en fazla etkiye sahip olduğu 25. zaman dilimine karşılık gelen $P_{t_{max}}$ değerleri

Bir diğer performans parametresinde sunucu kapasitesi ele alınmıştır. Çevrim içi sınavların yapıldığı sunucudaki kapasitenin 800 olduğu varsayılırsa sekiz bölümün öğrencilerinden aynı veya farklı sınavlar için aynı zaman diliminde sınava girebilecek maksimum öğrenci sayısı en fazla 800 olabilmektedir. Algoritmanın 300 iterasyon boyunca çalıştırılmasıyla elde edilen sonuçlarda bu değer maksimum yük değerlerine karşılık gelmektedir. Buna göre 300 iterasyon boyunca elde edilen sonuçlar içerisinde 800’ü aşan maksimum yük değerlerinin sayısına bakılarak algoritmanın performansı değerlendirilebilir. Bu değerlendirme için 300 iterasyon sonuçları içerisinde maksimum yükün olduğu zaman dilimlerinden en çok tekrar eden ilk üçü baz alınmıştır. Bu zaman dilimleri 11., 20. ve 25. zaman dilimleridir.

Örneğin; 300 iterasyon boyunda elde edilen sonuçlarda, 11. zaman dilimine karşılık gelen maksimum yük değerleri incelenmiş, inceleme sonucu oluşturulan grafik Şekil 4.4’te verilmiştir. Grafikte her renk bir bölümü temsil etmekte olup renklerin temsil ettiği bölüm isimleri grafiğin altında yer almaktadır. Bununla birlikte grafikte, her bölüm için mevcut iterasyonlarda 11. zaman diliminde sınava giren öğrenci sayıları bölümlere ait alanların üzerinde yer almaktadır. Grafiğin en sonunda yatay şekilde verilmiş olan değerler, ilgili iterasyon için bölümlere ait sınava giren öğrenci sayılarının toplamından elde edilen maksimum yük değerleridir. Örneğin, 11. zaman diliminin bulunduğu 192. iterasyonda

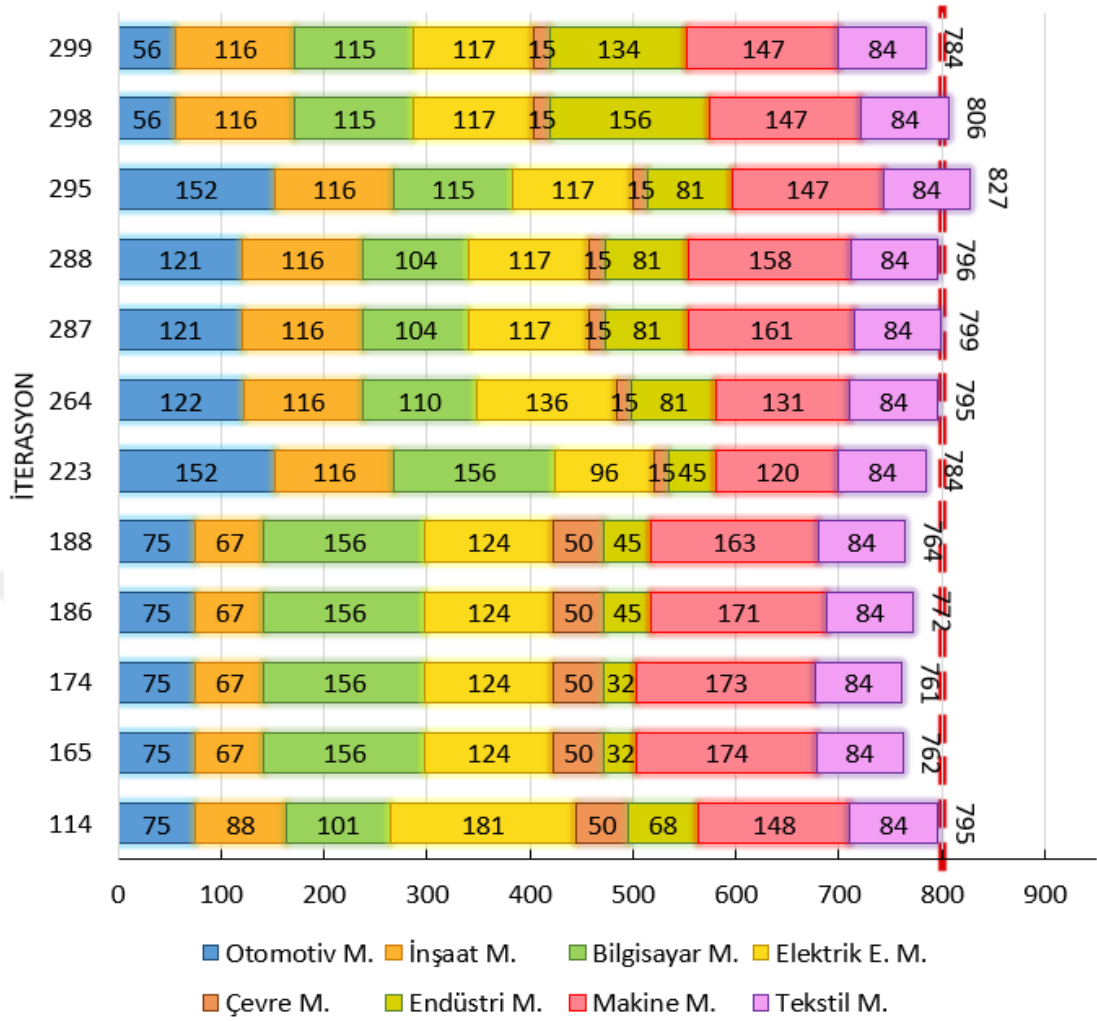
sınava giren öğrenci sayıları sırasıyla şöyledir; Otomotiv Mühendisliğinden 135 öğrenci, İnşaat Mühendisliğinden 125 öğrenci, Bilgisayar Mühendisliğinden 110 öğrenci, Elektrik Elektronik Mühendisliğinden 103 öğrenci, Çevre Mühendisliğinden 52 öğrenci, Endüstri Mühendisliğinden 134 öğrenci, Makine Mühendisliğinden 75 öğrenci ve Tekstil Mühendisliğinden 37 öğrenci. Bu bağlamda, 192. iterasyondaki maksimum yük değerinin bu değerlerin toplamı ile elde edilen 771 değeri olduğu grafikten okunabilmektedir. Benzer şekilde, 20. ve 25. zaman diliminin ele alındığı Şekil 4.5 ve Şekil 4.6’da yer alan grafiklerin yapısı da yukarıda anlatıldığı gibidir.

Şekil 4.4’te yer alan grafiğe bakıldığında, 11. zaman diliminin 14 ayrı iterasyonda görüldüğü söylenebilir. Bu iterasyonlardaki maksimum yük değerleri incelendiğinde sunucu kapasitesini aşan sekiz adet iterasyon olduğu görülmektedir. Bunlar, 64. iterasyonda ortaya çıkan 869, 65. iterasyonda ortaya çıkan 867, 73 iterasyonda ortaya çıkan 810, 108. iterasyonda ortaya çıkan 937, 109. iterasyonda ortaya çıkan 882, 117. iterasyonda ortaya çıkan 829, 121. iterasyonda ortaya çıkan 804 ve 123. iterasyonda ortaya çıkan 827 değerleridir. Buna göre 11. zaman diliminin görüldüğü 14 adet iterasyonun 8 tanesinde maksimum yük değerinin sunucu kapasitesini aştığı söylenebilir.



Şekil 4.4. 300 iterasyonda 11. zaman dilimine karşılık gelen maksimum yük değerleri

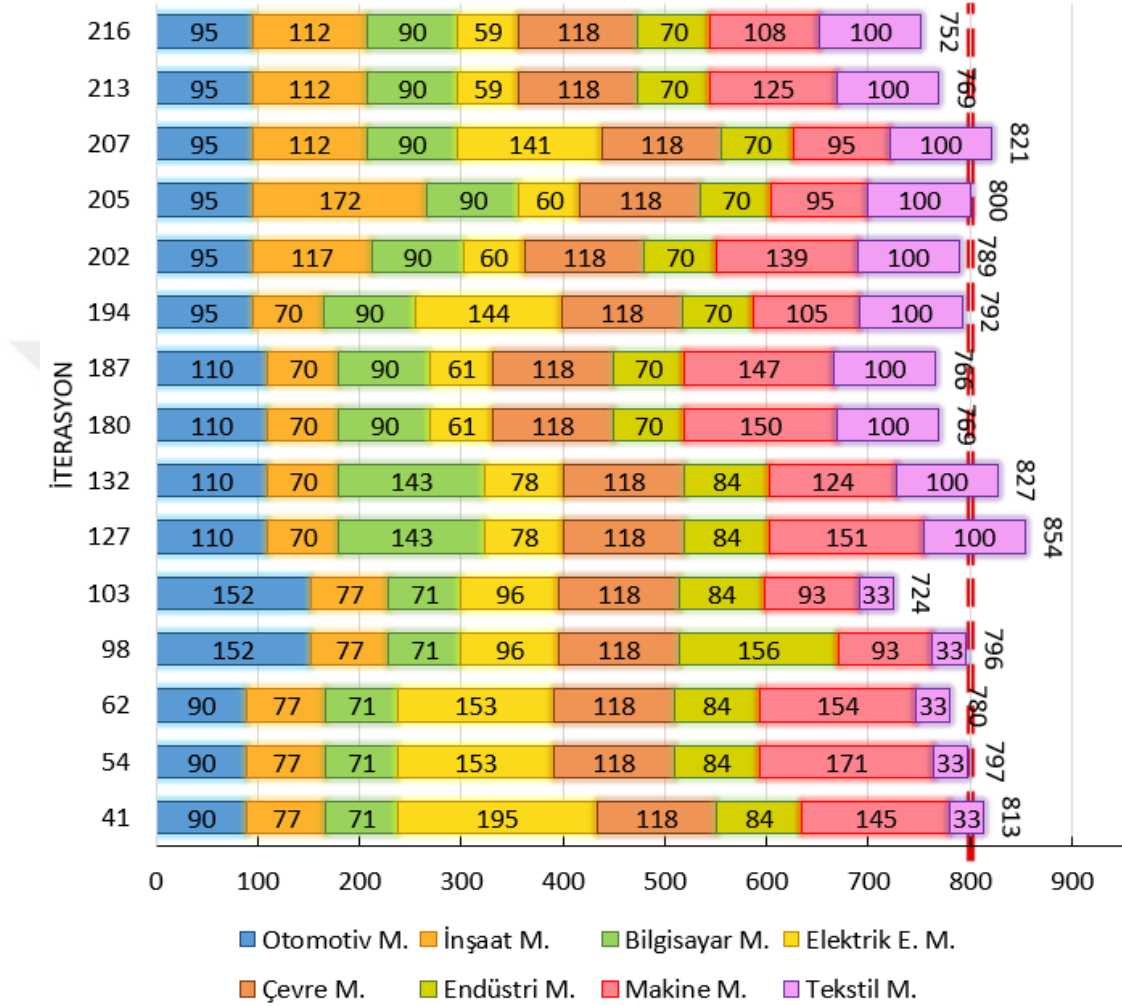
Maksimum yük değerlerinin incelendiği bir diğer zaman dilimi 20. zaman dilimi olup inceleme sonucu oluşturulan grafik Şekil 4.5'te verilmiştir. Grafiğe göre 20. zaman dilimi 12 ayrı iterasyonda görülmektedir. Bu iterasyonlardaki maksimum yük değerlerine bakıldığında sunucu kapasitesini aşan sadece 2 adet iterasyon olduğu görülmektedir. Bunlar, 295. iterasyonda ortaya çıkan 827 ve 298. iterasyonda ortaya çıkan 806 değerleridir. Bu sonuçlara bakılarak 20. zaman diliminin görüldüğü 12 adet iterasyonun 2 tanesinde maksimum yük değerinin sunucu kapasitesini aştığı söylenebilir.



Şekil 4.5. 300 iterasyon sonuçlarında 20. zaman dilimine karşılık gelen maksimum yük değerleri

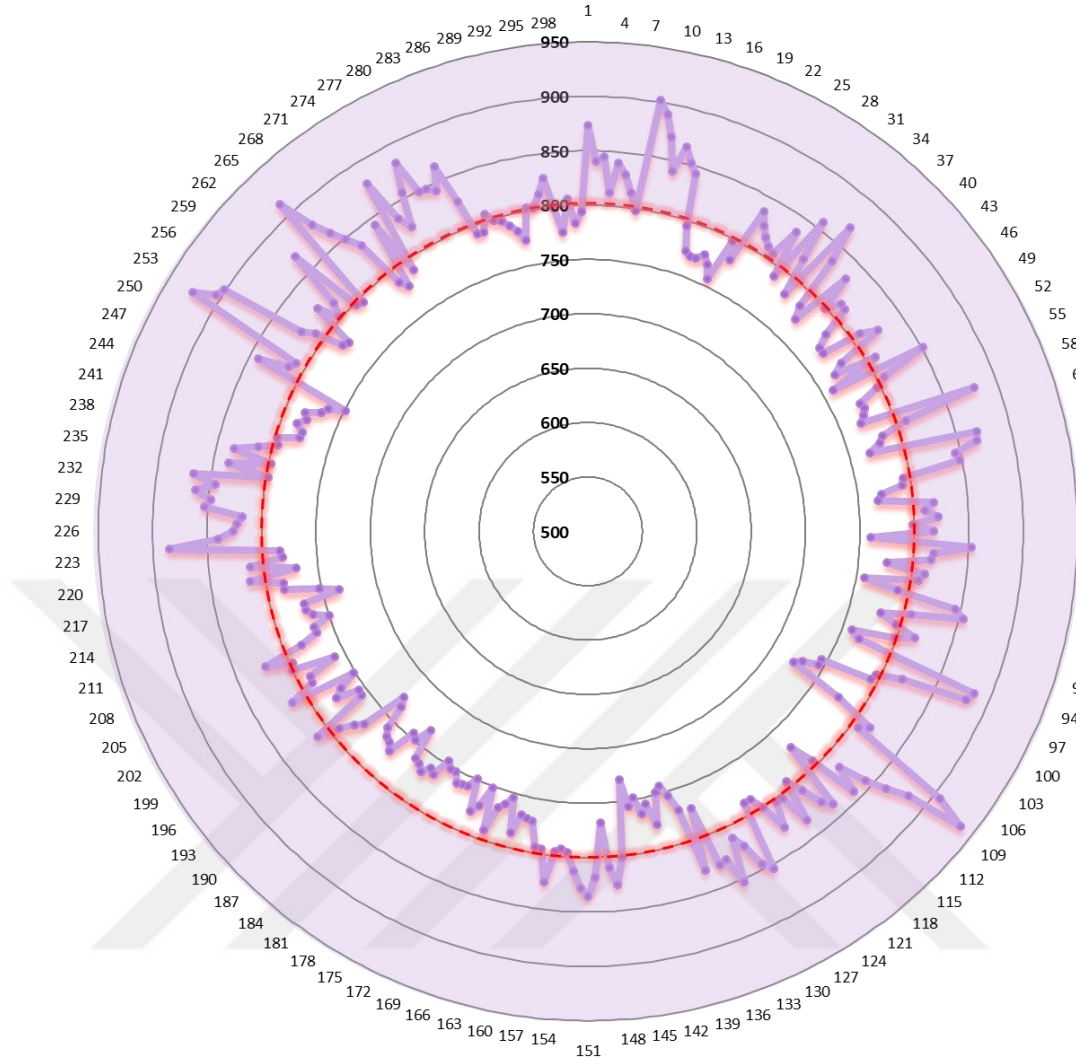
Benzer şekilde, 25. zaman dilimine karşılık gelen maksimum yük değerleri incelenmiştir. İnceleme sonucu oluşturulan grafik Şekil 4.6'da verilmiştir. Grafiğe bakıldığında 25. zaman diliminin 15 ayrı iterasyonda görüldüğü söylenebilir. Bu iterasyonlardaki maksimum yük değerlerine göre sunucu kapasitesini aşan 4 adet iterasyon olduğu görülmektedir. Bunlar, 41. iterasyonda ortaya çıkan 813, 127. iterasyonda ortaya çıkan 854, 132. iterasyonda ortaya çıkan 827 ve 207. iterasyonda ortaya çıkan 821 değerleridir. Buna göre 25. zaman diliminin görüldüğü 15 adet iterasyonun 4 tanesinde maksimum yük değerinin sunucu kapasitesini aştığı söylenebilir.

Sunulan üç adet örnekte de sunucu kapasitesini aşan maksimum yük değerlerinin iterasyon adedince oldukça az olduğu görülmektedir. Bu durum, geliştirilen algoritmanın iyi bir performansa sahip olduğunu göstermektedir.



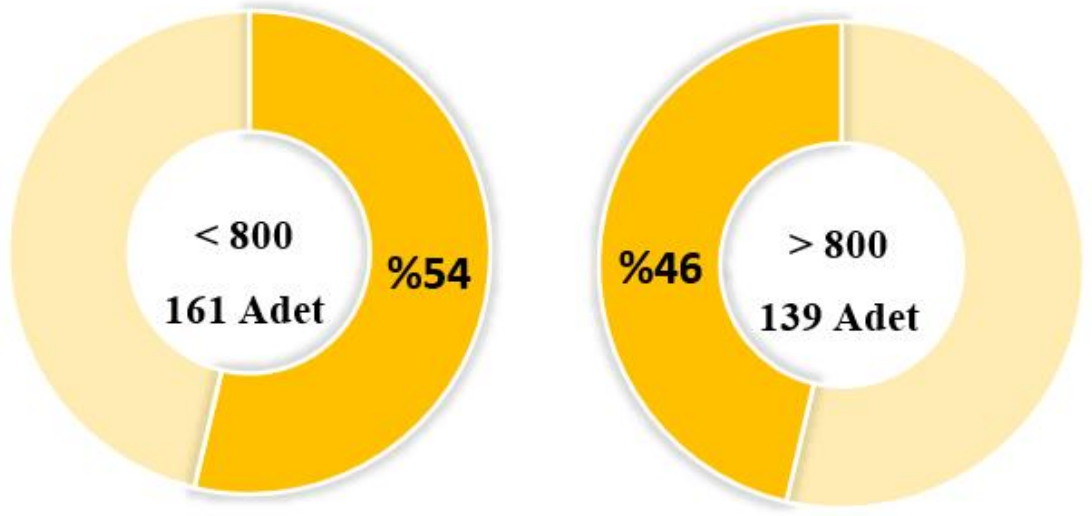
Şekil 4.6. 300 iterasyon sonuçlarında 25. zaman dilimine karşılık gelen maksimum yük değerleri

Bir diğer performans parametresinde ise 300 iterasyon içerisindeki tüm maksimum yük değerleri ele alınmıştır. Ele alınan bu değerler Şekil 4.7’de bulunan radar grafiğine yansıtılmıştır. Grafikte, sunucu kapasitesi olan 800 değerini aşan değerlerin mor ile tarandığı görülmektedir.



Şekil 4.7. 300 iterasyon içerisindeki tüm maksimum yük değerleri

300 iterasyon sonucunda, sunucu kapasitesini aşan maksimum yük değerine sahip toplam 139 adet iterasyonun mevcut olduğu saptanmıştır. Bunun yanı sıra, iterasyonlar içerisinde sunucu kapasitesinin altında kalan maksimum yük değerine sahip toplam 161 adet iterasyonun mevcut olduğu saptanmıştır. Şekil 4.8’de verilen grafiklerde bu değerlerin 300 iterasyon sonuçlarındaki yüzdeleri verilmiştir. Buna göre sunucu kapasitesini aşan maksimum yük değerlerinin oranının %46, sunucu kapasitesinin altında kalan maksimum yük değerlerinin oranının ise %54 olduğu görülmektedir. Bu durum da geliştirilen algoritmanın iyi bir performans sergilediğini göstermektedir.



Şekil 4.8. 300 iterasyon sonucundaki tüm maksimum yük değerlerinin sunucu kapasitesini aşma yüzdeleri

5. SONUÇ

COVID-19 pandemi döneminde gerçekleştirilen çevrim içi süreçler insan hayatında önemli bir rol oynamaktadır. Bu zorlu dönemde üniversiteler tarafından sunulan çevrim içi imkanlar eğitim ve öğretimin aksamadan devam etmesini sağlamıştır. Bu imkanlardan biri de üniversitelerin öğrencilerin derslerdeki bilgi düzeyini ölçmek için ortaya koydukları çevrim içi sınav sistemleridir. Bursa Uludağ Üniversitesi de pandemi döneminde öğrencilerin gerek öğrenim gerekse ölçme ve değerlendirme süreçlerinin aksamaması için çevrim içi sistemlerden yararlanmışır. Okul bünyesinde yapılan sınavlar çevrim içi olarak bir sunucu üzerinden gerçekleştirilmiştir.

Bu tez çalışmasında, COVID-19 pandemi döneminde yapılan çevrim içi sınavların çizelgelenmesi problemi ele alınmıştır. Çalışmada çözüm yaklaşımı olarak tamsayı matematiksel programlama modeli geliştirilmiştir. Çalışmanın uygulama aşaması için Bursa Uludağ Üniversitesi Mühendislik Fakültesi'ndeki sekiz bölüme ait derslere dair verileri içeren gerçek veri setleri kullanılmışır. Geliştirilen modele veri setlerinin adapte edilmesiyle Mühendislik Fakültesi'ndeki her bölüm için ayrı ayrı CPLEX Optimization Studio ortamında dengeli sınav çizelgeleri oluşturulmuştur. Ardından Microsoft Excel ortamında geliştirilen algoritma aracılığıyla dinamik bir yapıya ulaşılmıştır. Böylece iteratif bir şekilde çözüm elde edilerek çevrim içi sınavların yapıldığı sunucudaki maksimum yükün oluştuğu zaman diliminde ilgili bölüm için sınava giren maksimum öğrenci sayısının belli ölçüde azaltılmasıyla sunucu yükünün lokal olarak dengelenmesi sağlanmaya çalışılmışır. Bahsedilen dinamik yapı Microsoft Excel ortamında 300 iterasyon boyunca çalıştırılmışır. 300 iterasyon sonuçları içerisinde en yüksek maksimum yük değeri ve en düşük maksimum yük değeri için bu değerleri barındıran sekiz bölüme ait sınav çizelgeleri oluşturulmuştur. Oluşturulan sınav çizelgelerinden en düşük maksimum yük değerini barındıran sınav çizelgelerinin daha iyi performans göstereceği söylenebilir.

Çalışmada çeşitli performans parametreleri kullanılarak önerilen çözüm yaklaşımının performansının değerlendirilmesi amaçlanmıştır. Bu amaç doğrultusunda, maksimum yükün oluştuğu belli bir zaman dilimindeki sınava giren maksimum öğrenci sayısını, 300

iterasyon ile elde edilen maksimum yükün olduğu zaman dilimlerinden en çok tekrar eden ilk üç zaman dilimine karşılık gelen sunucu kapasitesini, 300 iterasyon ile elde edilen maksimum yük değerlerini ele alan üç adet performans parametresi ile değerlendirmeler yapılmıştır.

Yapılan değerlendirmeler sonucunda çalışmanın hedefine paralel olarak, maksimum yükün olduğu belli bir zaman dilimindeki sınava giren maksimum öğrenci sayısının iterasyon sayısı arttıkça azaldığı, 300 iterasyon sonucunda elde edilen sonuçlar incelendiğinde, en çok tekrar eden zaman dilimleri bazında iterasyon adedince sunucu kapasitesini aşan maksimum yük değerlerinin oldukça az olduğu ve sunucu kapasitesini aşan maksimum yük değerlerinin oranının %46 olduğu sonuçlarına ulaşılmıştır. Böylelikle, çalışmada sunulan çözüm yaklaşımının iyi bir performans sergilediği yorumu yapılabilir.

Bu tez çalışmasında, Bursa Uludağ Üniversitesi Mühendislik Fakültesi bölümleri için çevrim içi sınav çizelgeleme problemi çözüme ulaştırılmıştır. Çalışmada geliştirilen çözüm yaklaşımı sayesinde çevrim içi sınavların yapıldığı sunucudaki yük dengesi ilgili bölüm bazında sağlanmaya çalışılmıştır. Yapılan çalışma sonucunda sunucu yükünün artışı kaynaklanan sistem hatalarının ortadan kalkması, buna bağlı olarak mazeret sınavlarının sayısında azalma sağlanması hedeflenmiştir. Bu hedef doğrultusunda yapılan çalışmanın hem öğrenciler hem de öğretim elemanları açısından faydalı olacağı düşünülmektedir. 300 iterasyon sonucunda elde edilen sınav çizelgelerinde, sunucu kapasitesinin altında kalan maksimum yük oranının %54 olduğu da göz önünde bulundurulduğunda, sunucu yükünde yaşanan problemlerden kaynaklı mazeret sınavı sayısında önemli oranda azalma olacağı söylenebilir.

Gelecekte yapılacak çalışmalarda, bu tez çalışmasında sunulan çözüm yaklaşımına ek olarak çeşitli sezgisel ve meta-sezgisel yaklaşımlar ele alınabilir. Bunun yanı sıra, çalışmada geliştirilen algoritmanın performansı sezgisel ve meta-sezgisel yöntemlerin performansı ile karşılaştırılabilir. Bir diğer gelecek çalışma alanı olarak, varyasyon katsayısının dikkate alınması ile farklı performans parametreleri ele alınarak çalışmada geliştirilen algoritmanın performansı analiz edilebilir.

KAYNAKLAR

- Abdelhalim, E. A., & El Khayat, G. A. (2016). An information visibility-based university timetabling for efficient use of learning spaces (IVUT). *Egyptian Informatics Journal*, 17(3), 315–325. <https://doi.org/10.1016/j.eij.2016.04.003>
- Abou Kasm, O., Mohandes, B., Diabat, A., & El Khatib, S. (2019). Exam timetabling with allowable conflicts within a time window. *Computers and Industrial Engineering*, 127(November 2018), 263–273. <https://doi.org/10.1016/j.cie.2018.11.037>
- Aizam, N. A. H., Ismail, Z. F., & Yen, C. L. S. (2022). Mathematical Model for Scheduling Problems: A Compatibility Test on University Course Timetabling Problem. *Lecture Notes in Electrical Engineering*, 835, 111–123. https://doi.org/10.1007/978-981-16-8515-6_10
- Akkan, C., Gülcü, A., & Kuş, Z. (2021). Minimum penalty perturbation heuristics for curriculum-based timetables subject to multiple disruptions. *Computers and Operations Research*, 132(March), 105306. <https://doi.org/10.1016/j.cor.2021.105306>
- Al-Yakoob, S. M., Sherali, H. D., & Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7(1), 19–46. <https://doi.org/10.1007/s10287-007-0066-8>
- Al-Yakoob, S. M., & Sherali, H. D. (2015). Mathematical models and algorithms for a high school timetabling problem. *Computers and Operations Research*, 61, 56–68. <https://doi.org/10.1016/j.cor.2015.02.011>
- Almossa, S. Y. (2021). University students' perspectives toward learning and assessment during COVID-19. *Education and Information Technologies*, 26(6), 7163–7181. <https://doi.org/10.1007/S10639-021-10554-8>
- Assi, M., Halawi, B., & Haraty, R. A. (2018). Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. *Procedia Computer Science*, 126, 899–906. <https://doi.org/10.1016/j.procs.2018.08.024>
- Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers and Industrial Engineering*, 86, 43–59. <https://doi.org/10.1016/j.cie.2014.11.010>
- Badoni, R. P., Gupta, D. K., & Mishra, P. (2014). A new hybrid algorithm for university course timetabling problem using events based on groupings of students. *Computers and Industrial Engineering*, 78, 12–25. <https://doi.org/10.1016/j.cie.2014.09.020>
- Barnhart, C., Bertsimas, D., Delarue, A., & Yan, J. (2020). *Course Scheduling Under Sudden Scarcity: Applications to Pandemic Planning*. 1–32.
- Bergmann, L. K., Fischer, K., & Zurheide, S. (2014). A linear mixed-integer model for realistic examination timetabling problems. *PATAT 2014 - Proceedings of the 10th International Conference on the Practice and Theory of Automated Timetabling, August*, 82–101.
- Bilen, E., & Matros, A. (2021). Online cheating amid COVID-19. *Journal of Economic Behavior & Organization*, 182, 196–211. <https://doi.org/10.1016/J.JEBO.2020.12.004>
- Bliss, K. M., Fowler, K. R., & Galluzzo, B. J. (2014). *Math modeling: Getting started & getting solutions*. Society for Industrial and Applied Mathematics.
- Bradley, S., Hax, A., Magnanti, T. (1977). *Applied Mathematical Programming*. Addison-Wesley.

- Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R. (2010). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1), 46–53. <https://doi.org/10.1016/J.EJOR.2010.01.044>
- Buttler, T., George, D., & Bruggemann, K. (2021). Student input on the effectiveness of the shift to emergency remote teaching due to the COVID crisis: Structural equation modeling creates a more complete picture. *International Journal of Educational Research Open*, 2, 100036. <https://doi.org/10.1016/J.IJEDRO.2021.100036>
- Carter, M. W., & Price, C. C. (2001). *Operations Research: A Practical Introduction*. CRC Press.
- Cavdur, F., & 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. <https://doi.org/10.1007/s40815-015-0046-z>
- Chen, D. S., Batson, R. G., Dang, Y. (2011). *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons.
- Chen, M., Tang, X., Song, T., Wu, C., Liu, S., & Peng, X. (2020). A Tabu search algorithm with controlled randomization for constructing feasible university course timetables. *Computers and Operations Research*, 123, 105007. <https://doi.org/10.1016/j.cor.2020.105007>
- Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1), 117–135. [https://doi.org/10.1016/S0377-2217\(03\)00103-6](https://doi.org/10.1016/S0377-2217(03)00103-6)
- De Causmaecker, P., Demeester, P., & Vanden Berghe, G. (2009). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, 195(1), 307–318. <https://doi.org/10.1016/j.ejor.2008.01.043>
- De Coster, A., Musliu, N., Schaerf, A., Schoisswohl, J., & Smith-Miles, K. (2021). Algorithm selection and instance space analysis for curriculum-based course timetabling. *Journal of Scheduling*. <https://doi.org/10.1007/s10951-021-00701-x>
- De la Rosa-Rivera, F., Nunez-Varela, J. I., Ortiz-Bayliss, J. C., & Terashima-Marín, H. (2021). Algorithm selection for solving educational timetabling problems. *Expert Systems with Applications*, 174(February), 102405. <https://doi.org/10.1016/j.eswa.2021.114694>
- Dorneles, Á. P., De Araújo, O. C. B., & Buriol, L. S. (2014). A fix-and-optimize heuristic for the high school timetabling problem. *Computers and Operations Research*, 52(PART A), 29–38. <https://doi.org/10.1016/j.cor.2014.06.023>
- Dorneles, Á. P., de Araújo, O. C. B., & Buriol, L. S. (2017). A column generation approach to high school timetabling modeled as a multicommodity flow problem. *European Journal of Operational Research*, 256(3), 685–695. <https://doi.org/10.1016/j.ejor.2016.07.002>
- Fathi, A., Salehi, M., Mohammadi, M., Rahimof, Y., & Hajialigol, P. (2021). Cooling/heating load management in educational buildings through course scheduling. *Journal of Building Engineering*, 41(February), 102405. <https://doi.org/10.1016/j.job.2021.102405>
- Fong, C. W., Asmuni, H., Leong, P. H., Sam, Y. H., Pang, Y. Y., & Sim, H. M. (2022). Zombie Survival Optimization in Solving University Examination Timetabling Problem. *2022 IEEE International Conference on Automatic Control and Intelligent*

- Systems, I2CACIS 2022 - Proceedings*, 169–173.
<https://doi.org/10.1109/I2CACIS54679.2022.9815494>
- Fonseca, G. H. G., Santos, H. G., Carrano, E. G., & Stidsen, T. J. R. (2017). Integer programming techniques for educational timetabling. *European Journal of Operational Research*, 262(1), 28–39. <https://doi.org/10.1016/j.ejor.2017.03.020>
- Gokulkumari, G., Al-Hussain, T., Akmal, S., & Singh, P. (2022). Analysis of E-Exam practices in higher education institutions of KSA: Learners' perspectives. *Advances in Engineering Software*, 173, 103195. <https://doi.org/10.1016/J.ADVENGSOFT.2022.103195>
- Gülcü, A., & Akkan, C. (2020). Robust university course timetabling problem subject to single and multiple disruptions. *European Journal of Operational Research*, 283(2), 630–646. <https://doi.org/10.1016/j.ejor.2019.11.024>
- Güler, M. G., Geçici, E., Köroğlu, T., & Becit, E. (2021). A web-based decision support system for examination timetabling. *Expert Systems with Applications*, 183(June). <https://doi.org/10.1016/j.eswa.2021.115363>
- Haruna, C. R., Attipoe, E. K., Attipoe, E., Mensah, I. A., Opuni-Boachie, K., & Agyekum, O. (2022). Using Random Scheduling Technique with Crowd-Sourcing to Solve Exam Timetable Securing public wireless networks View project Using Random Scheduling Technique with Crowd-Sourcing to Solve Exam Timetable. *Article in International Journal of Computer Applications*, 183(50), 975–8887. <https://doi.org/10.5120/ijca2022921928>
- Jia, J., & He, Y. (2022). The design, implementation and pilot application of an intelligent online proctoring system for online exams. *Interactive Technology and Smart Education*, 19(1), 112–120. <https://doi.org/10.1108/ITSE-12-2020-0246>
- Klijn, F., Mdaghri Alaoui, M., & Vorsatz, M. (2022). Academic integrity in on-line exams: Evidence from a randomized field experiment. *Journal of Economic Psychology*, 93, 102555. <https://doi.org/10.1016/J.JOEP.2022.102555>
- Komijan, A. R., & Koupaei, M. N. (2012). A new binary model for university examination timetabling: a case study. *Journal of Industrial Engineering International*, 8(1), 1–7. <https://doi.org/10.1186/2251-712X-8-28>
- Leite, N., Melício, F., & Rosa, A. C. (2019). A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*, 122, 137–151. <https://doi.org/10.1016/j.eswa.2018.12.048>
- Leite, N., Melício, F., & Rosa, A. C. (2021). A Fast Threshold Acceptance Algorithm for the Examination Timetabling Problem. *International Series in Operations Research and Management Science*, 309(September), 323–363. https://doi.org/10.1007/978-3-030-74051-1_11
- Lemos, A., Melo, F. S., Monteiro, P. T., & Lynce, I. (2019). Room usage optimization in timetabling: A case study at Universidade de Lisboa. *Operations Research Perspectives*, 6(December 2018), 100092. <https://doi.org/10.1016/j.orp.2018.100092>
- Lindahl, M., Mason, A. J., Stidsen, T., & Sorensen, M. (2018). A strategic view of University timetabling. *European Journal of Operational Research*, 266(1), 35–45. <https://doi.org/10.1016/j.ejor.2017.09.022>
- Mujuni, E. (2021). A Discrete Bat Algorithm for the Examination Timetabling Problem. *International Journal of Informatics and Applied Mathematics*, 4(2), 1–16.
- Phillips, A. E., Waterer, H., Ehrgott, M., & Ryan, D. M. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers and*

- Operations Research*, 53, 42–53. <https://doi.org/10.1016/j.cor.2014.07.012>
- Potu, B. K., Atwa, H., Nasr El-Din, W. A., Othman, M. A., Sarwani, N. A., Fatima, A., Deifalla, A., & Fadel, R. A. (2022). Learning anatomy before and during COVID-19 pandemic: Students' perceptions and exam performance. *Morphologie*, 106(354), 188–194. <https://doi.org/10.1016/J.MORPHO.2021.07.003>
- Prosad, R., Ashikur Rahman Khan, M., & Ahammad, I. (2022). Design of Class Routine and Exam Hall Invigilation System based on Genetic Algorithm and Greedy Approach. *Asian Journal of Research in Computer Science*, 13(3), 28–44. <https://doi.org/10.9734/AJRCOS/2022/v13i330316>
- Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons.
- Shadkam, E., & Rajabi, R. (2021). An integer mathematical model for the problem of timetabling university exams. *Journal of Brilliant Engineering*, 3, 11–15. <https://doi.org/10.36937/ben.2021.003.003>
- Song, K., Kim, S., Park, M., & Lee, H. S. (2017). Energy efficiency-based course timetabling for university buildings. *Energy*, 139, 394–405. <https://doi.org/10.1016/j.energy.2017.07.176>
- Song, T., Chen, M., Xu, Y., Wang, D., Song, X., & Tang, X. (2021). Competition-guided multi-neighborhood local search algorithm for the university course timetabling problem. *Applied Soft Computing*, 110, 107624. <https://doi.org/10.1016/j.asoc.2021.107624>
- Sorensen, M., & Dahms, F. H. W. (2014). A Two-Stage Decomposition of High School Timetabling applied to cases in Denmark. *Computers and Operations Research*, 43(1), 36–49. <https://doi.org/10.1016/j.cor.2013.08.025>
- Sun, Y., Luo, X., & Liu, X. (2021). Optimization of a university timetable considering building energy efficiency: An approach based on the building controls virtual test bed platform using a genetic algorithm. *Journal of Building Engineering*, 35(December 2020), 102095. <https://doi.org/10.1016/j.jobe.2020.102095>
- Tassopoulos, I. X., & Beligiannis, G. N. (2012). A hybrid particle swarm optimization based algorithm for high school timetabling problems. *Applied Soft Computing Journal*, 12(11), 3472–3489. <https://doi.org/10.1016/j.asoc.2012.05.029>
- Tassopoulos, I. X., Iliopoulou, C. A., & Beligiannis, G. N. (2020). Solving the Greek school timetabling problem by a mixed integer programming model. *Journal of the Operational Research Society*, 71(1), 117–132. <https://doi.org/10.1080/01605682.2018.1557022>
- Tavakoli, M. M., Shirouyehzad, H., Lotfi, F. H., & Najafi, S. E. (2020). Proposing a novel heuristic algorithm for university course timetabling problem with the quality of courses rendered approach; a case study. *Alexandria Engineering Journal*, 59(5), 3355–3367. <https://doi.org/10.1016/j.aej.2020.05.004>
- Ünal, Y. Z., & 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. <https://doi.org/10.1016/j.ejor.2014.03.015>
- Vermuyten, H., Lemmens, S., Marques, I., & Beliën, J. (2016). Developing compact course timetables with optimized student flows. *European Journal of Operational Research*, 251(2), 651–661. <https://doi.org/10.1016/j.ejor.2015.11.028>
- Winston, W. L., Goldberg, J. B. (2004). *Operations Research: Applications and Algorithms*. Thomson Brooks/Cole.
- Yasari, P., Ranjbar, M., Jamili, N., & Shaelaie, M. H. (2019). A two-stage stochastic programming approach for a multi-objective course timetabling problem with

courses cancelation risk. *Computers and Industrial Engineering*, 130(March), 650–660. <https://doi.org/10.1016/j.cie.2019.02.050>

Zhu, K., Li, L. D., & Li, M. (2022). Developing an Online Examination Timetabling System Using Artificial Bee Colony Algorithm in Higher Education. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 413 LNICST*, 112–131. https://doi.org/10.1007/978-3-030-93479-8_7



EKLER

- EK 1** 300 İterasyon Boyunca Ulaşılan Sonuçları İçeren Çizelge
- EK 2** En Yüksek Maksimum Yük Değeri Olan 937 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri
- EK 3** En Düşük Maksimum Yük Değeri Olan 724 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri



EK 1 300 İterasyon Boyunca Ulaşılan Sonuçları İçeren Çizelge

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
0	162	3	162	-	-	-	-	-	-	-	162	1
0	175	33	-	175	-	-	-	-	-	-	175	2
0	192	41	-	-	192	-	-	-	-	-	192	3
0	196	10	-	-	-	196	-	-	-	-	196	4
0	140	1	-	-	-	-	140	-	-	-	140	5
0	163	42	-	-	-	-	-	163	-	-	163	6
0	191	42	-	-	-	-	-	-	191	-	191	7
0	183	16	-	-	-	-	-	-	-	183	183	8
1	873	10	97	114	137	196	26	55	157	91	196	4
2	840	24	108	60	0	162	80	157	171	102	171	7
3	845	31	70	130	111	194	32	105	134	69	194	4
4	812	27	95	133	145	104	44	43	158	90	158	7
5	840	43	72	112	143	138	64	84	191	36	191	7
6	830	34	153	147	0	84	47	139	171	89	171	7
7	814	24	108	60	0	162	80	157	145	102	162	4
8	798	27	95	133	145	104	44	43	144	90	145	3
9	902	33	86	175	156	183	0	100	123	79	183	4
10	890	34	153	147	102	84	47	139	129	89	153	1
11	871	34	134	147	102	84	47	139	129	89	147	2
12	840	17	113	51	115	117	75	79	148	142	148	7

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
13	865	43	124	112	122	138	64	84	185	36	185	7
14	852	43	124	112	122	138	64	84	172	36	172	7
15	843	43	124	112	122	138	64	84	163	36	163	7
16	795	19	29	144	0	144	54	148	142	134	148	6
17	773	17	113	51	115	117	75	79	81	142	142	8
18	770	33	0	175	156	87	0	100	173	79	175	2
19	770	34	134	44	102	84	47	139	131	89	139	6
20	776	23	102	88	141	95	39	187	77	47	187	6
21	769	30	47	64	122	53	76	200	148	59	200	6
22	757	5	149	104	0	75	25	117	156	131	156	7
23	798	11	78	67	90	122	52	184	168	37	184	6
24	782	1	152	66	116	54	140	28	154	72	154	7
25	836	17	113	51	115	117	75	105	146	114	146	7
26	824	17	113	51	115	117	75	105	134	114	134	7
27	816	17	113	51	115	117	75	105	126	114	126	7
28	810	19	29	144	0	144	54	108	197	134	197	7
29	807	17	113	51	115	117	75	105	117	114	117	4
30	835	6	153	61	89	131	63	165	128	45	165	6
31	790	5	149	104	0	122	25	117	142	131	149	1
32	857	16	78	70	104	104	36	109	173	183	183	8

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
33	819	30	0	64	122	128	76	121	147	161	161	8
34	783	26	153	81	128	87	69	81	81	103	153	1
35	869	30	152	64	122	128	76	121	147	59	152	1
36	835	30	118	64	122	128	76	121	147	59	147	7
37	785	27	113	133	0	81	44	182	142	90	182	6
38	773	6	158	61	89	131	63	47	179	45	179	7
39	832	30	118	64	122	128	76	121	144	59	144	7
40	799	30	118	64	122	128	76	121	111	59	128	4
41	813	25	90	77	71	195	118	84	145	33	195	4
42	812	28	51	100	137	132	0	155	172	65	172	7
43	784	16	99	70	104	151	36	109	143	72	151	4
44	772	1	59	66	116	179	140	28	112	72	179	4
45	771	33	152	51	156	159	0	37	137	79	159	4
46	809	6	158	61	89	199	63	47	147	45	199	4
47	825	27	113	133	0	200	44	131	114	90	200	4
48	779	6	158	61	89	169	63	47	147	45	169	4
49	769	27	113	133	0	144	44	131	114	90	144	4
50	809	19	60	144	0	170	54	108	139	134	170	4
51	760	16	99	70	104	127	36	109	143	72	143	7
52	851	36	132	65	0	60	103	185	174	132	185	6

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								P_{tmax}	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
53	807	30	118	64	122	117	76	121	130	59	130	7
54	797	25	90	77	71	153	118	84	171	33	171	7
55	776	29	105	84	111	159	21	109	174	13	174	7
56	778	28	51	100	137	122	0	155	148	65	155	6
57	775	29	105	84	111	159	21	109	173	13	173	7
58	770	15	168	85	41	124	47	134	145	26	168	1
59	879	44	134	45	103	127	120	156	137	57	156	6
60	809	44	134	45	103	127	120	86	137	57	137	7
61	785	6	158	61	89	86	63	134	149	45	158	1
62	780	25	90	77	71	153	118	84	154	33	154	7
63	769	44	134	45	103	127	120	86	97	57	134	1
64	869	11	158	67	90	136	52	156	173	37	173	7
65	867	11	158	67	90	136	52	156	171	37	171	7
66	845	26	95	81	128	36	69	178	155	103	178	6
67	848	18	92	147	0	200	75	113	193	28	200	4
68	794	11	158	67	90	136	52	156	98	37	158	1
69	792	18	92	147	0	144	75	113	193	28	193	7
70	772	5	110	104	0	139	25	117	146	131	146	7
71	768	26	95	81	128	36	69	81	175	103	175	7
72	819	36	134	65	0	60	103	134	191	132	191	7

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
73	810	11	105	67	90	136	52	156	167	37	167	7
74	822	23	65	88	141	94	39	171	177	47	177	7
75	799	15	152	85	41	124	47	134	190	26	190	7
76	818	23	65	88	141	94	39	171	173	47	173	7
77	760	5	110	104	0	139	25	117	134	131	139	4
78	853	26	95	81	128	139	69	81	157	103	157	7
79	818	15	152	85	41	159	47	134	174	26	174	7
80	816	26	95	81	128	139	69	81	120	103	139	4
81	776	3	69	65	114	144	51	90	174	69	174	7
82	812	23	65	88	141	104	39	171	157	47	171	6
83	807	36	134	65	0	95	103	134	144	132	144	7
84	758	19	89	144	0	56	54	108	173	134	173	7
85	790	36	134	65	0	95	103	134	127	132	134	1
86	845	34	124	44	102	183	47	95	161	89	183	4
87	854	19	135	144	0	131	54	108	148	134	148	7
88	807	19	135	144	0	131	54	108	101	134	144	2
89	779	23	97	88	141	104	39	102	161	47	161	7
90	796	19	135	103	0	131	54	108	131	134	135	1
91	816	3	136	65	114	144	51	90	147	69	147	7
92	799	3	136	65	114	144	51	90	130	69	144	4

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
93	759	3	136	65	114	104	51	90	130	69	136	1
94	768	28	153	100	137	151	0	21	141	65	153	1
95	885	14	178	58	101	160	23	64	190	111	190	7
96	880	14	178	58	101	160	23	64	185	111	185	7
97	819	14	178	58	101	160	23	64	124	111	178	1
98	796	25	152	77	71	96	118	156	93	33	156	6
99	793	19	134	103	0	168	54	108	92	134	168	4
100	745	8	153	50	0	192	60	103	174	13	192	4
101	745	17	29	98	115	81	75	141	92	114	141	6
102	731	8	153	50	0	178	60	103	174	13	178	4
103	724	25	152	77	71	96	118	84	93	33	152	1
104	776	38	134	20	110	159	83	55	148	67	159	4
105	816	24	153	60	0	136	80	131	154	102	154	7
106	807	17	113	98	115	114	75	69	109	114	115	3
107	906	24	153	60	102	136	80	131	142	102	153	1
108	937	11	135	67	167	199	52	156	124	37	199	4
109	882	11	135	67	167	144	52	156	124	37	167	3
110	862	5	106	98	141	112	25	117	132	131	141	3
111	843	5	106	98	122	112	25	117	132	131	132	7
112	817	5	106	98	122	112	25	117	106	131	131	8

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
113	842	16	134	70	143	140	36	109	64	146	146	8
114	795	20	75	88	101	181	50	68	148	84	181	4
115	773	16	134	70	143	140	36	109	64	77	143	3
116	837	18	95	147	101	147	75	113	131	28	147	2
117	829	11	135	67	137	144	52	156	101	37	156	6
118	809	18	95	119	101	147	75	113	131	28	147	4
119	793	19	86	103	104	95	54	108	109	134	134	8
120	833	32	92	178	143	84	33	79	132	92	178	2
121	804	11	135	125	137	102	52	115	101	37	137	3
122	803	18	95	119	123	119	75	113	131	28	131	7
123	827	11	135	125	102	102	52	115	159	37	159	7
124	801	19	86	103	167	95	54	108	112	76	167	3
125	788	32	92	54	163	84	33	79	191	92	191	7
126	788	18	95	119	123	119	75	113	116	28	123	3
127	854	25	110	70	143	78	118	84	151	100	151	7
128	845	1	34	188	71	170	140	28	142	72	188	2
129	823	30	60	178	122	118	76	121	89	59	178	2
130	812	10	0	114	137	142	26	105	197	91	197	7
131	853	31	113	130	128	154	32	53	174	69	174	7
132	827	25	110	70	143	78	118	84	124	100	143	3

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
133	830	27	102	110	122	83	44	131	148	90	148	7
134	772	27	102	110	122	83	44	131	90	90	131	6
135	830	14	55	66	116	129	23	186	144	111	186	6
136	770	35	105	43	156	54	28	134	191	59	191	7
137	754	12	150	28	128	146	15	134	153	0	153	7
138	743	38	112	20	90	86	83	114	171	67	171	7
139	748	24	0	60	122	155	80	128	101	102	155	4
140	777	19	86	103	103	146	54	79	130	76	146	4
141	756	4	152	72	123	178	1	40	174	16	178	4
142	764	21	0	51	114	133	0	186	195	85	195	7
143	748	12	150	28	128	159	15	134	134	0	159	4
144	756	37	190	65	141	144	40	102	68	6	190	1
145	730	24	0	60	122	98	80	128	140	102	140	7
146	802	39	103	73	143	151	32	0	191	109	191	7
147	826	35	105	43	156	112	28	134	189	59	189	7
148	810	35	105	43	156	112	28	134	173	59	173	7
149	768	35	105	43	156	112	28	134	131	59	156	3
150	818	8	89	72	116	140	60	137	191	13	191	7
151	836	7	23	137	102	161	17	131	181	84	181	7
152	828	7	23	137	102	161	17	131	173	84	173	7

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
153	813	7	23	137	102	161	17	131	158	84	161	4
154	796	37	100	65	165	144	40	102	174	6	174	7
155	793	37	100	65	165	144	40	102	171	6	171	7
156	796	1	34	124	115	131	140	35	145	72	145	7
157	825	31	113	130	61	180	32	69	171	69	180	4
158	795	26	97	58	122	88	69	104	154	103	154	7
159	794	26	97	58	122	88	69	104	153	103	153	7
160	771	1	34	124	115	131	140	35	120	72	140	5
161	770	37	100	65	165	144	40	102	148	6	165	3
162	768	31	113	130	61	122	32	69	172	69	172	7
163	786	8	89	72	116	126	60	137	173	13	173	7
164	754	8	89	72	116	126	60	137	141	13	141	7
165	762	20	75	67	156	124	50	32	174	84	174	7
166	767	31	113	130	61	122	32	69	171	69	171	7
167	791	26	97	58	122	88	69	104	150	103	150	7
168	752	8	89	72	116	126	60	137	139	13	139	7
169	771	26	97	58	122	88	69	104	130	103	130	7
170	779	36	0	65	71	192	103	86	130	132	192	4
171	749	37	100	65	143	144	40	102	149	6	149	7
172	759	28	52	100	122	154	0	103	163	65	163	7

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
173	760	7	23	137	102	95	17	131	171	84	171	7
174	761	20	75	67	156	124	50	32	173	84	173	7
175	752	31	113	130	61	122	32	69	156	69	156	7
176	752	7	23	137	102	95	17	131	163	84	163	7
177	747	8	89	72	116	126	60	137	134	13	137	6
178	765	11	135	125	110	21	52	134	151	37	151	7
179	761	11	135	125	110	21	52	134	147	37	147	7
180	769	25	110	70	90	61	118	70	150	100	150	7
181	764	31	113	130	61	122	32	102	135	69	135	7
182	762	31	113	130	61	122	32	102	133	69	133	7
183	733	1	34	124	115	131	88	37	132	72	132	7
184	749	15	60	85	101	72	47	185	173	26	185	6
185	745	15	60	85	101	72	47	181	173	26	181	6
186	772	20	75	67	156	124	50	45	171	84	171	7
187	766	25	110	70	90	61	118	70	147	100	147	7
188	764	20	75	67	156	124	50	45	163	84	163	7
189	758	11	135	125	110	21	52	134	144	37	144	7
190	736	31	113	130	61	122	32	102	107	69	130	2
191	728	1	34	124	115	131	88	37	127	72	131	4
192	771	11	135	125	110	103	52	134	75	37	135	1

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
193	779	6	51	61	137	165	63	86	171	45	171	7
194	792	25	95	70	90	144	118	70	105	100	144	4
195	812	10	16	142	141	159	26	47	190	91	190	7
196	757	43	113	88	0	171	64	134	151	36	171	4
197	756	10	16	142	141	159	26	47	134	91	159	4
198	777	40	99	115	111	114	29	43	164	102	164	7
199	769	1	106	124	115	96	88	37	131	72	131	7
200	752	1	106	124	115	96	88	37	114	72	124	2
201	814	33	60	171	89	100	0	141	174	79	174	7
202	789	25	95	117	90	60	118	70	139	100	139	7
203	788	33	60	171	89	100	0	141	148	79	171	2
204	760	32	110	51	71	157	33	67	179	92	179	7
205	800	25	95	172	90	60	118	70	95	100	172	2
206	797	35	153	108	0	190	28	115	144	59	190	4
207	821	25	95	112	90	141	118	70	95	100	141	4
208	783	35	153	108	0	176	28	115	144	59	176	4
209	766	35	153	108	0	159	28	115	144	59	159	4
210	767	36	86	0	71	191	103	81	103	132	191	4
211	750	21	113	40	114	144	0	63	191	85	191	7
212	764	36	86	0	71	150	103	81	141	132	150	4

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
213	769	25	95	112	90	59	118	70	125	100	125	7
214	769	40	99	60	111	174	29	43	151	102	174	4
215	735	21	113	40	114	146	0	63	174	85	174	7
216	752	25	95	112	90	59	118	70	108	100	118	5
217	784	21	113	40	114	146	72	63	151	85	151	7
218	784	7	86	64	102	131	86	85	146	84	146	7
219	814	35	153	108	0	149	59	115	171	59	171	7
220	770	35	153	108	0	149	59	115	127	59	153	1
221	812	33	125	120	89	40	45	141	173	79	173	7
222	782	13	142	123	128	88	82	0	191	28	191	7
223	784	20	152	116	156	96	15	45	120	84	156	3
224	885	4	105	104	156	122	90	144	148	16	156	3
225	840	4	105	104	111	122	90	144	148	16	148	7
226	826	4	105	104	111	122	90	144	134	16	144	6
227	823	4	105	104	111	122	90	141	134	16	141	6
228	818	41	153	85	122	144	100	33	149	32	153	1
229	853	7	152	64	102	131	86	153	81	84	153	6
230	848	41	113	85	122	144	100	103	149	32	149	7
231	863	15	90	88	141	93	120	137	168	26	168	7
232	845	7	152	64	102	131	86	85	141	84	152	1

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
233	866	15	153	88	141	93	120	137	108	26	153	1
234	798	14	99	81	137	133	32	64	141	111	141	7
235	836	15	78	88	141	93	120	137	153	26	153	7
236	798	21	134	40	110	146	72	63	148	85	148	7
237	834	15	78	88	141	93	120	137	151	26	151	7
238	813	15	78	88	141	93	120	137	130	26	141	3
239	796	44	97	115	116	59	96	124	132	57	132	7
240	798	15	78	88	114	93	120	137	142	26	142	7
241	779	35	0	108	141	149	59	121	142	59	149	4
242	778	44	97	115	116	59	96	124	114	57	124	6
243	785	35	0	108	141	91	59	185	142	59	185	6
244	778	14	99	81	0	191	32	131	133	111	191	4
245	782	13	57	123	172	88	82	85	147	28	172	3
246	768	1	77	103	90	81	60	170	115	72	170	6
247	764	21	134	40	0	146	72	181	106	85	181	6
248	749	44	97	115	116	59	96	95	114	57	116	3
249	842	14	99	81	122	133	32	131	133	111	133	4
250	814	21	134	40	71	172	72	134	106	85	172	4
251	810	15	78	88	128	160	120	119	91	26	160	4
252	925	29	153	91	103	159	81	154	171	13	171	7

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
253	905	29	153	91	103	159	81	154	151	13	159	4
254	901	29	153	91	103	155	81	154	151	13	155	4
255	821	29	153	91	103	75	81	154	151	13	154	6
256	809	42	0	77	102	159	53	196	173	49	196	6
257	783	7	0	64	137	140	86	131	141	84	141	7
258	780	7	0	64	137	140	86	131	138	84	140	4
259	822	29	153	91	103	144	81	70	167	13	167	7
260	802	29	153	91	103	144	81	70	147	13	153	1
261	814	26	47	172	156	109	35	72	120	103	172	2
262	869	16	92	159	141	95	39	165	101	77	165	6
263	798	21	134	40	71	133	72	134	129	85	134	1
264	795	20	122	116	110	136	15	81	131	84	136	4
265	913	42	86	77	102	184	53	188	174	49	188	6
266	879	42	86	77	102	184	53	154	174	49	184	4
267	862	33	99	120	115	157	45	103	144	79	157	4
268	849	33	99	120	115	144	45	103	144	79	144	4
269	835	42	86	77	102	140	53	154	174	49	174	7
270	787	27	152	124	111	121	52	56	81	90	152	1
271	779	21	77	40	71	159	72	134	141	85	159	4
272	843	39	97	85	104	160	26	115	147	109	160	4

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
273	789	7	95	64	137	72	86	131	120	84	137	3
274	879	32	152	114	143	96	45	64	173	92	173	7
275	836	32	152	114	143	96	45	64	130	92	152	1
276	823	18	77	124	111	136	121	109	117	28	136	4
277	855	41	53	85	156	159	100	128	142	32	159	4
278	882	32	106	114	143	188	45	64	130	92	188	4
279	849	32	106	114	143	155	45	64	130	92	155	4
280	848	32	106	114	143	154	45	64	130	92	154	4
281	843	41	53	85	156	147	100	128	142	32	156	3
282	864	38	147	128	89	122	51	109	151	67	151	7
283	826	38	147	128	89	122	51	109	113	67	147	1
284	792	38	113	128	89	122	51	109	113	67	128	2
285	791	13	52	123	103	144	82	85	174	28	174	7
286	807	38	113	119	89	122	51	109	137	67	137	7
287	799	20	121	116	104	117	15	81	161	84	161	7
288	796	20	121	116	104	117	15	81	158	84	158	7
289	790	13	52	123	103	144	82	85	173	28	173	7
290	783	44	104	115	0	195	96	95	121	57	195	4
291	774	41	152	85	0	147	100	128	130	32	152	1
292	803	18	134	124	41	120	121	109	126	28	134	1
293	802	14	134	81	141	51	32	131	121	111	141	3

EK 1 (Devam)

İterasyon	Maks. Yük	Maks. Yükün Oluştuğu Zaman Dilimi	P_t								$P_{t_{max}}$	Maks. Yüke Neden Olan Bölüm
			Otomotiv Müh. -1-	İnşaat Müh. -2-	Bilgisayar Müh. -3-	Elektrik Elektronik Müh. -4-	Çevre Müh. -5-	Endüstri Müh. -6-	Makine Müh. -7-	Tekstil Müh. -8-		
294	813	21	110	40	116	118	72	134	138	85	138	7
295	827	20	152	116	115	117	15	81	147	84	152	1
296	799	1	59	103	128	119	60	134	124	72	134	6
297	776	18	0	124	90	120	121	170	123	28	170	6
298	806	20	56	116	115	117	15	156	147	84	156	6
299	784	20	56	116	115	117	15	134	147	84	147	7
300	794	10	146	65	111	78	26	105	172	91	172	7

Maks.: Maksimum, Müh.: Mühendisliği

EK 2 En Yüksek Maksimum Yük Değeri Olan 937 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri

Gün	Zaman Dilimi	Otomotiv Mühendisliği				İnşaat Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	116	Au-19(100)			Au-32(16)	66		Ci-13(66)		
	2	86	Au-03(52)			Au-37(34)	48		Ci-15(48)		
	3	0					65			Ci-24(65)	
	4	113	Au-17(113)				72			Ci-32(72)	
2	5	19				Au-35(19)	98	Ci-49(57)		Ci-23(41)	
	6	90				Au-28(90)	61			Ci-26(61)	
	7	0					137	Ci-06(52)			Ci-38(85)
	8	153			Au-25(153)		50				Ci-41(50)
3	9	55				Au-38(55)	0				
	10	47				Au-36(47)	114	Ci-01(114)			
	11	94	Au-15(59)			Au-39(35)	67			Ci-33(67)	
	12	105			Au-11(105)		61	Ci-48(61)			
4	13	60				Au-33(60)	0				
	14	65				Au-34(65)	58			Ci-36(58)	
	15	56	Au-07(56)				85			Ci-52(85)	
	16	92				Au-30(92)	70			Ci-27(70)	
5	17	29				Au-40(29)	98		Ci-19(51)		Ci-46(47)
	18	51				Au-13(51)	147	Ci-05(119)	Ci-17(28)		
	19	134	Au-18(134)				103	Ci-11(103)			
	20	69		Au-05(69)			88		Ci-51(88)		
6	21	0					51				Ci-43(51)
	22	0					49		Ci-22(49)		
	23	0					88	Ci-08(29)	Ci-12(59)		
	24	106			Au-24(106)		60			Ci-31(60)	

EK 2 (Devam)

Gün	Zaman Dilimi	Otomotiv Mühendisliği				İnşaat Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	152			Au-27(152)		77			Ci-28(77)	
	26	97			Au-26(97)		81		Ci-47(81)		
	27	0					133	Ci-09(61)		Ci-34(72)	
	28	78		Au-09(78)			100		Ci-14(91)		Ci-37(9)
8	29	102	Au-02(102)				84	Ci-07(36)	Ci-21(48)		
	30	95		Au-23(95)			64				Ci-42(64)
	31	89		Au-08(89)			69			Ci-30(69)	
	32	53	Au-06(53)				178	Ci-50(124)			Ci-40(54)
9	33	23				Au-12(23)	51				Ci-39(51)
	34	57	Au-04(57)				44			Ci-25(44)	
	35	77		Au-16(77)			43		Ci-16(43)		
	36	110			Au-10(110)		65			Ci-29(65)	
10	37	70		Au-21(70)			65		Ci-18(65)		
	38	60	Au-14(52)			Au-29(8)	20				Ci-45(20)
	39	159	Au-01(99)			Au-31(60)	73	Ci-04(61)			Ci-44(12)
	40	0					112	Ci-02(112)			
11	41	0					60	Ci-10(60)			
	42	86		Au-22(86)			0				
	43	0					112	Ci-03(57)		Ci-35(55)	
	44	113	Au-20(113)				45		Ci-20(45)		

EK 2 (Devam)

Gün	Zaman Dilimi	Bilgisayar Mühendisliği				Elektrik Elektronik Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	116	Co-01(116)				0				
	2	71				Co-16(71)	78			El-23(78)	
	3	114	Co-02(114)				104		El-19(83)		El-45(21)
	4	143			Co-08(143)		59				El-42(59)
2	5	0					102	El-01(53)	El-57(49)		
	6	89				Co-18(89)	40				El-48(40)
	7	0					160		El-12(122)		El-44(38)
	8	0					114		El-14(114)		
3	9	0					94			El-32(72)	El-53(22)
	10	0					59		El-20(59)		
	11	90			Co-10(90)		71				El-50(71)
	12	0					149	El-58(58)		El-29(91)	
4	13	0					61				El-34(61)
	14	101				Co-14(101)	160	El-59(58)	El-17(66)		El-47(36)
	15	41			Co-12(41)		148	El-60(64)	El-18(84)		
	16	104				Co-17(104)	59				El-35(59)
5	17	115		Co-22(115)			81		El-13(81)		
	18	0					133				El-55(133)
	19	0					86			El-28(86)	
	20	0					111	El-05(56)		El-30(55)	
6	21	0					89	El-09(73)			El-38(16)
	22	0					53				El-54(53)
	23	141	Co-03(141)				81			El-33(81)	
	24	0					14				El-40(14)

EK 2 (Devam)

Gün	Zaman Dilimi	Bilgisayar Mühendisliği				Elektrik Elektronik Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	71	Co-21(71)				96			El-25(96)	
	26	128	Co-23(128)				54				El-49(54)
	27	0					82			El-27(82)	
	28	137			Co-09(137)		151	El-02(56)	El-15(81)		El-37(14)
8	29	111			Co-11(111)		105		El-62(82)		El-39(23)
	30	122		Co-25(122)			47			El-31(47)	
	31	0					164	El-08(124)			El-46(40)
	32	0					99	El-07(55)			El-36(44)
9	33	156		Co-06(156)			144				El-52(144)
	34	102			Co-13(102)		96		El-11(96)		
	35	83	Co-20(61)			Co-15(22)	158	El-03(109)			El-43(49)
	36	0					60	El-61(60)			
10	37	0					128	El-06(53)	El-21(75)		
	38	110		Co-07(110)			159			El-22(159)	
	39	0					80		El-16(80)		
	40	51				Co-19(51)	88	El-10(66)			El-41(22)
11	41	0					95			El-26(95)	
	42	123		Co-05(123)			88			El-24(88)	
	43	122	Co-24(122)				107		El-56(56)		El-51(51)
	44	103	Co-04(103)				58	El-04(58)			

EK 2 (Devam)

Gün	Zaman Dilimi	Çevre Mühendisliği				Endüstri Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	140	En-53(52)		En-34(2)	En-49(86)	28				In-45(28)
	2	122	En-55(26)			En-50(96)	85		In-49(85)		
	3	51				En-45(51)	90	In-08(68)			In-43(22)
	4	1				En-39(1)	35				In-42(35)
2	5	25	En-56(24)			En-37(1)	117		In-53(45)		In-39(72)
	6	63	En-06(32)	En-08(31)			47				In-41(47)
	7	17		En-15(17)			34	In-15(34)			
	8	60			En-19(60)		103		In-11(103)		
3	9	130	En-05(49)			En-48(81)	85		In-46(85)		
	10	26			En-17(26)		105			In-27(105)	
	11	52				En-36(52)	156	In-04(41)	In-52(43)		In-38(72)
	12	15				En-44(15)	128	In-10(128)			
4	13	74		En-58(53)	En-26(21)		65			In-30(65)	
	14	23				En-42(23)	64		In-19(64)		
	15	47			En-33(47)		134	In-05(134)			
	16	36			En-25(36)		109	In-02(42)		In-51(67)	
5	17	75		En-16(25)	En-30(50)		69	In-48(69)			
	18	75				En-38(75)	113		In-06(82)		In-40(31)
	19	54				En-46(54)	108	In-47(76)			In-34(32)
	20	50	En-54(49)			En-47(1)	68				In-37(68)
6	21	0					86			In-20(86)	
	22	39			En-20(39)		70				In-31(70)
	23	39		En-13(39)			102				In-35(102)
	24	80	En-04(45)	En-14(35)			131		In-18(131)		

EK 2 (Devam)

Gün	Zaman Dilimi	Çevre Mühendisliği				Endüstri Mühendisliği					
		<i>t</i> Anında Sınav Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınav Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	118	En-57(45)			En-40(73)	84			In-25(84)	
	26	69				En-52(69)	81				In-32(81)
	27	44			En-22(31)	En-35(13)	131	In-07(131)			
	28	0					21			In-22(21)	
8	29	21			En-28(14)	En-41(7)	109	In-09(69)			In-44(40)
	30	76		En-10(76)			121	In-16(36)	In-17(85)		
	31	32		En-12(32)			53			In-24(53)	
	32	33		En-09(33)			79				In-36(79)
9	33	0					37	In-14(37)			
	34	47			En-32(47)		95		In-50(95)		
	35	28		En-07(1)		En-51(27)	0				
	36	103	En-01(32)			En-43(71)	134	In-01(134)			
10	37	40			En-24(40)		56			In-23(56)	
	38	83	En-03(50)		En-31(33)		55			In-28(55)	
	39	32			En-23(32)		36				In-33(36)
	40	29			En-21(29)		42	In-03(42)			
11	41	82			En-18(82)		139	In-12(139)			
	42	50			En-27(50)		112	In-13(33)		In-26(79)	
	43	64	En-02(37)		En-29(27)		57			In-29(579)	
	44	120		En-11(120)			51			In-21(51)	

EK 2 (Devam)

Gün	Zaman Dilimi	Makine Mühendisliği				Tekstil Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	101	Me-13(75)			Me-46(26)	72			Te-28(72)	
	2	112	Me-03(66)			Me-54(46)	21			Te-37(21)	
	3	144	Me-14(130)			Me-50(14)	69	Te-13(60)	Te-22(2)		Te-58(7)
	4	114		Me-23(51)	Me-78(36)	Me-47(27)	16			Te-40(9)	Te-61(7)
2	5	141			Me-34(68)	Me-60(73)	131		Te-06(57)		Te-65(74)
	6	122		Me-20(44)		Me-65(78)	45			Te-29(45)	
	7	144	Me-75(67)	Me-26(43)		Me-53(34)	84	Te-04(34)		Te-38(25)	Te-73(25)
	8	174				Me-66(174)	13				Te-59(13)
3	9	130			Me-33(130)		11			Te-31(11)	
	10	72			Me-38(72)		91	Te-01(54)		Te-41(23)	Te-64(14)
	11	117		Me-18(64)		Me-51(53)	37			Te-39(25)	Te-57(12)
	12	132	Me-11(73)		Me-30(59)		0				
4	13	91		Me-24(55)		Me-71(36)	28			Te-36(14)	Te-66(14)
	14	124		Me-17(50)		Me-57(74)	111				Te-46(111)
	15	113	Me-76(72)			Me-74(41)	26				Te-53(26)
	16	90			Me-40(90)		72	Te-11(69)	Te-18(3)		
5	17	92	Me-06(69)			Me-70(23)	114				Te-55(114)
	18	64		Me-29(69)			28				Te-47(28)
	19	92		Me-28(48)		Me-55(44)	134	Te-09(58)	Te-21(76)		
	20	142			Me-39(66)	Me-64(76)	84		Te-23(60)		Te-56(24)
6	21	81		Me-09(51)		Me-63(30)	67		Te-10(40)		Te-72(27)
	22	191				Me-41(191)	66	Te-05(28)		Te-42(21)	Te-63(17)
	23	144	Me-02(63)	Me-25(81)			47			Te-33(26)	Te-62(21)
	24	154	Me-01(127)			Me-52(27)	102		Te-19(102)		

EK 2 (Devam)

Gün	Zaman Dilimi	Makine Mühendisliği				Tekstil Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	93	Me-12(76)			Me-42(17)	33		Te-20(2)		Te-49(31)
	26	109			Me-37(109)		103	Te-08(66)			Te-67(37)
	27	112		Me-08(47)		Me-69(65)	90		Te-17(58)		Te-71(32)
	28	60		Me-19(39)		Me-44(21)	65	Te-02(51)			Te-60(14)
8	29	131		Me-77(86)		Me-68(45)	13			Te-35(13)	
	30	76				Me-59(76)	59		Te-25(50)		Te-51(9)
	31	187	Me-15(70)	Me-22(56)		Me-48(61)	69		Te-24(69)		
	32	81				Me-58(81)	40			Te-30(29)	Te-70(11)
9	33	141	Me-16(73)			Me-62(68)	79			Te-32(34)	Te-68(45)
	34	124		Me-27(48)		Me-61(76)	89				Te-52(89)
	35	173				Me-67(173)	71	Te-14(52)			Te-50(19)
	36	83			Me-35(83)		132		Te-16(36)		Te-43(96)
10	37	112	Me-07(75)			Me-73(37)	6				Te-54(6)
	38	148				Me-72(148)	67			Te-26(67)	
	39	111		Me-21(49)		Me-56(62)	109	Te-12(66)		Te-27(43)	
	40	123	Me-05(70)			Me-45(53)	102	Te-07(63)			Te-44(39)
11	41	188	Me-10(70)		Me-31(88)	Me-43(30)	32			Te-34(32)	
	42	171			Me-36(171)		49	Te-03(13)			Te-48(36)
	43	102	Me-04(71)			Me-49(31)	36		Te-15(35)		Te-45(1)
	44	101			Me-32(101)		57				Te-69(57)

EK 3 En Düşük Maksimum Yük Değeri Olan 724 Değerin Meydana Geldiği Durumda Tüm Bölümler İçin Elde Edilen Sınav Çizelgeleri

Gün	Zaman Dilimi	Otomotiv Mühendisliği				İnşaat Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	34				Au-37(34)	66		Ci-13(66)		
	2	69		Au-05(69)			48		Ci-15(48)		
	3	78		Au-09(78)			65			Ci-24(65)	
	4	152			Au-27(152)		72			Ci-32(72)	
2	5	106			Au-24(106)		98	Ci-49(57)		Ci-23(41)	
	6	47				Au-36(47)	61			Ci-26(61)	
	7	23				Au-12(23)	137	Ci-06(52)			Ci-38(85)
	8	89		Au-08(89)			50				Ci-41(50)
3	9	0					0				
	10	0					114	Ci-01(114)			
	11	135		Au-21(70)		Au-34(65)	67			Ci-33(67)	
	12	150	Au-01(99)			Au-13(51)	61	Ci-48(61)			
4	13	0					0				
	14	55				Au-38(55)	58			Ci-36(58)	
	15	60				Au-31(60)	85			Ci-52(85)	
	16	134	Au-18(134)				70			Ci-27(70)	
5	17	52	Au-14(52)				98		Ci-19(51)		Ci-46(47)
	18	95		Au-23(95)			147	Ci-05(119)	Ci-17(28)		
	19	86		Au-22(86)			103	Ci-11(103)			
	20	75	Au-15(59)			Au-32(16)	88		Ci-51(88)		
6	21	0					51				Ci-43(51)
	22	0					49		Ci-22(49)		
	23	82	Au-06(53)			Au-40(29)	88	Ci-08(29)	Ci-12(59)		
	24	0					60			Ci-31(60)	

EK 3 (Devam)

Gün	Zaman Dilimi	Otomotiv Mühendisliği				İnşaat Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	110			Au-10(110)		77			Ci-28(77)	
	26	97			Au-26(97)		81		Ci-47(81)		
	27	102	Au-02(102)				133	Ci-09(61)		Ci-34(72)	
	28	52	Au-03(52)				100		Ci-14(91)		Ci-37(9)
8	29	153			Au-25(153)		84	Ci-07(36)	Ci-21(48)		
	30	60				Au-33(60)	64				Ci-42(64)
	31	113	Au-20(113)				69			Ci-30(69)	
	32	92				Au-30(92)	178	Ci-50(124)			Ci-40(54)
9	33	19				Au-35(19)	51				Ci-39(51)
	34	113	Au-17(113)				44			Ci-25(44)	
	35	105			Au-11(105)		43		Ci-16(43)		
	36	0					65			Ci-29(65)	
10	37	190	Au-19(100)			Au-28(90)	65		Ci-18(65)		
	38	112		Au-16(77)		Au-39(35)	20				Ci-45(20)
	39	56	Au-07(56)				73	Ci-04(61)			Ci-44(12)
	40	0					112	Ci-02(112)			
11	41	8				Au-29(8)	60	Ci-10(60)			
	42	0					0				
	43	57	Au-04(57)				112	Ci-03(57)		Ci-35(55)	
	44	0					45		Ci-20(45)		

EK 3 (Devam)

Gün	Zaman Dilimi	Bilgisayar Mühendisliği				Elektrik Elektronik Mühendisliği					
		t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	t Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	0					51				El-51(51)
	2	103	Co-04(103)				160	El-60(64)		El-25(96)	
	3	71	Co-21(71)				23				El-39(23)
	4	0					191		El-11(96)	El-26(95)	
2	5	141	Co-03(141)				112		El-20(59)		El-54(53)
	6	123		Co-05(123)			66		El-17(66)		
	7	110		Co-07(110)			169		El-13(81)	El-24(88)	
	8	101				Co-14(101)	40				El-46(40)
3	9	71				Co-16(71)	61				El-34(61)
	10	22				Co-15(22)	14				El-40(14)
	11	167	Co-01(116)			Co-19(51)	199	El-07(55)			El-52(144)
	12	0					102			El-28(86)	El-38(16)
4	13	90			Co-10(90)		69			El-31(47)	El-53(22)
	14	0					38				El-44(38)
	15	0					56	El-05(56)			
	16	89				Co-18(89)	140	El-04(58)	El-62(82)		
5	17	0					114		El-14(114)		
	18	61	Co-20(61)				147	El-59(58)	El-57(49)		El-48(40)
	19	143			Co-08(143)		21				El-45(21)
	20	114	Co-02(114)				181	El-03(109)		El-32(72)	
6	21	0					59				El-35(59)
	22	128	Co-23(128)				22				El-41(22)
	23	0					49				El-43(49)
	24	102			Co-13(102)		136	El-01(53)	El-19(83)		

EK 3 (Devam)

Gün	Zaman Dilimi	Bilgisayar Mühendisliği				Elektrik Elektronik Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	104				Co-17(104)	0				
	26	122		Co-25(122)			81			El-33(81)	
	27	137			Co-09(137)		109	El-09(73)			El-47(36)
	28	156		Co-06(156)			175		El-18(84)	El-29(91)	
8	29	0					112	El-06(53)			El-42(59)
	30	41			Co-12(41)		55			El-30(55)	
	31	0					193	El-61(60)			El-55(133)
	32	0					56		El-56(56)		
9	33	122	Co-24(122)				0				
	34	0					82			El-27(82)	
	35	0					159			El-22(159)	
	36	0					75		El-21(75)		
10	37	115		Co-22(115)			124		El-16(80)		El-36(44)
	38	0					66	El-10(66)			
	39	0					81		El-15(81)		
	40	111			Co-11(111)		78			El-23(78)	
11	41	0					195	El-08(124)			El-50(71)
	42	0					54				El-49(54)
	43	0					70	El-02(56)			El-37(14)
	44	0					180	El-58(58)	El-12(122)		

EK 3 (Devam)

Gün	Zaman Dilimi	Çevre Mühendisliği				Endüstri Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	140	En-53(52)		En-34(2)	En-49(86)	28				In-45(28)
	2	122	En-55(26)			En-50(96)	85		In-49(85)		
	3	51				En-45(51)	90	In-08(68)			In-43(22)
	4	1				En-39(1)	35				In-42(35)
2	5	25	En-56(24)			En-37(1)	117		In-53(45)		In-39(72)
	6	63	En-06(32)	En-08(31)			47				In-41(47)
	7	17		En-15(17)			34	In-15(34)			
	8	60			En-19(60)		103		In-11(103)		
3	9	130	En-05(49)			En-48(81)	85		In-46(85)		
	10	26			En-17(26)		105			In-27(105)	
	11	52				En-36(52)	156	In-04(41)	In-52(43)		In-38(72)
	12	15				En-44(15)	128	In-10(128)			
4	13	74		En-58(53)	En-26(21)		65			In-30(65)	
	14	23				En-42(23)	64		In-19(64)		
	15	47			En-33(47)		134	In-05(134)			
	16	36			En-25(36)		109	In-02(42)		In-51(67)	
5	17	75		En-16(25)	En-30(50)		69	In-48(69)			
	18	75				En-38(75)	113		In-06(82)		In-40(31)
	19	54				En-46(54)	108	In-47(76)			In-34(32)
	20	50	En-54(49)			En-47(1)	68				In-37(68)
6	21	0					86			In-20(86)	
	22	39			En-20(39)		70				In-31(70)
	23	39		En-13(39)			102				In-35(102)
	24	80	En-04(45)	En-14(35)			131		In-18(131)		

EK 3 (Devam)

Gün	Zaman Dilimi	Çevre Mühendisliği				Endüstri Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	118	En-57(45)			En-40(73)	84			In-25(84)	
	26	69				En-52(69)	81				In-32(81)
	27	44			En-22(31)	En-35(13)	131	In-07(131)			
	28	0					21			In-22(21)	
8	29	21			En-28(14)	En-41(7)	109	In-09(69)			In-44(40)
	30	76		En-10(76)			121	In-16(36)	In-17(85)		
	31	32		En-12(32)			53			In-24(53)	
	32	33		En-09(33)			79				In-36(79)
9	33	0					37	In-14(37)			
	34	47			En-32(47)		95		In-50(95)		
	35	28		En-07(1)		En-51(27)	0				
	36	103	En-01(32)			En-43(71)	134	In-01(134)			
10	37	40			En-24(40)		56			In-23(56)	
	38	83	En-03(50)		En-31(33)		55			In-28(55)	
	39	32			En-23(32)		36				In-33(36)
	40	29			En-21(29)		42	In-03(42)			
11	41	82			En-18(82)		139	In-12(139)			
	42	50			En-27(50)		112	In-13(33)		In-26(79)	
	43	64	En-02(37)		En-29(27)		57			In-29(579)	
	44	120		En-11(120)			51			In-21(51)	

EK 3 (Devam)

Gün	Zaman Dilimi	Makine Mühendisliği					Tekstil Mühendisliği				
		<i>t</i> Anında Sınavı Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınavı Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
1	1	130	Me-04(71)		Me-30(59)		72			Te-28(72)	
	2	154	Me-75(67)	Me-17(50)		Me-73(37)	21			Te-37(21)	
	3	141	Me-01(127)			Me-50(14)	69	Te-13(60)	Te-22(2)		Te-58(7)
	4	173				Me-67(173)	16			Te-40(9)	Te-61(7)
2	5	132		Me-18(64)	Me-34(68)		131		Te-06(57)		Te-65(74)
	6	146	Me-11(73)			Me-60(73)	45			Te-29(45)	
	7	152		Me-09(51)	Me-32(101)		84	Te-04(34)		Te-38(25)	Te-73(25)
	8	195	Me-15(70)	Me-08(47)		Me-65(78)	13				Te-59(13)
3	9	119	Me-16(73)			Me-54(46)	11			Te-31(11)	
	10	138	Me-12(76)			Me-56(62)	91	Te-01(54)		Te-41(23)	Te-64(14)
	11	124		Me-27(48)		Me-61(76)	37			Te-39(25)	Te-57(12)
	12	171			Me-36(171)		0				
4	13	65				Me-69(65)	28			Te-36(14)	Te-66(14)
	14	153	Me-76(72)	Me-25(81)			111				Te-46(111)
	15	120			Me-40(90)	Me-43(30)	26				Te-53(26)
	16	139	Me-10(70)	Me-28(48)		Me-44(21)	72	Te-11(69)	Te-18(3)		
5	17	109			Me-37(109)		114				Te-55(114)
	18	189	Me-06(69)	Me-77(86)		Me-53(34)	28				Te-47(28)
	19	139	Me-07(75)	Me-29(64)			134	Te-09(58)	Te-21(76)		
	20	146	Me-05(70)			Me-59(76)	84		Te-23(60)		Te-56(24)
6	21	191	Me-14(130)			Me-48(61)	67		Te-10(40)		Te-72(27)
	22	191				Me-41(191)	66	Te-05(28)		Te-42(21)	Te-63(17)
	23	148				Me-72(148)	47			Te-33(26)	Te-62(21)
	24	142			Me-39(66)	Me-64(76)	102		Te-19(102)		

EK 3 (Devam)

Gün	Zaman Dilimi	Makine Mühendisliği				Tekstil Mühendisliği					
		<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf	<i>t</i> Anında Sınava Giren Öğrenci Sayısı	1. Sınıf	2. Sınıf	3. Sınıf	4. Sınıf
7	25	118		Me-20(44)		Me-57(74)	33		Te-20(2)		Te-49(31)
	26	130			Me-33(130)		103	Te-08(66)			Te-67(37)
	27	101		Me-22(56)		Me-68(45)	90		Te-17(58)		Te-71(32)
	28	27				Me-52(27)	65	Te-02(51)			Te-60(14)
8	29	30				Me-63(30)	13			Te-35(13)	
	30	83	Me-03(66)			Me-42(17)	59		Te-25(50)		Te-51(9)
	31	75		Me-19(39)		Me-71(36)	69		Te-24(69)		
	32	53				Me-45(53)	40			Te-30(29)	Te-70(11)
9	33	66		Me-26(43)		Me-70(23)	79			Te-32(34)	Te-68(45)
	34	114			Me-31(88)	Me-46(26)	89				Te-52(89)
	35	41				Me-74(41)	71	Te-14(52)			Te-50(19)
	36	80		Me-21(49)		Me-49(31)	132		Te-16(36)		Te-43(96)
10	37	162	Me-02(63)	Me-24(55)		Me-55(44)	6				Te-54(6)
	38	81				Me-58(81)	67			Te-26(67)	
	39	53				Me-51(53)	109	Te-12(66)		Te-27(43)	
	40	174				Me-66(174)	102	Te-07(63)			Te-44(39)
11	41	143	Me-13(75)			Me-62(68)	32			Te-34(32)	
	42	72			Me-38(72)		49	Te-03(13)			Te-48(36)
	43	114		Me-23(51)	Me-78(36)	Me-47(27)	36		Te-15(35)		Te-45(1)
	44	83			Me-35(83)		57				Te-69(57)