



**T.C.**  
**NECMETTİN ERBAKAN ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**



**SINAV ÇİZELGELEME PROBLEMİ**  
**ÇÖZÜMÜ İÇİN FARKLI YAKLAŞIMLAR**

**Arafat KOCA**

**YÜKSEK LİSANS TEZİ**

**Endüstri Mühendisliği Anabilim Dalı**

**Temmuz-2018**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Arafat KOCA tarafından hazırlanan ‘‘SINAV İZELGELEME PROBLEMİ ÖZÜMÜ İİN FARKLI YAKLAŞIMLAR’’ adlı tez alışması 23/07/2018 tarihinde aşığıdaki jüri üyeleri tarafından oy birliğı / oy okluğu ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliğı Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

#### Başkan

Prof. Dr. Mehmet AKTAN

#### Danışman

Dr. Öğr. Üyesi Ahmet Reha BOTSALI

#### Üye

Doç. Dr. Saadettin Erhan KESEN

### İmza

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mehmet KARALI  
Enstitü Müdürü

## **TEZ BİLDİRİMİ**

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

### **DECLARATION PAGE**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Arafat KOCA

Tarih: 23 / 07 / 2018

# ÖZET

## YÜKSEK LİSANS TEZİ

### SINAV ÇİZELGELEME PROBLEMİ ÇÖZÜMÜ İÇİN FARKLI YAKLAŞIMLAR

**Arafat KOCA**

**Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı**

**Danışman: Dr. Öğr. Üyesi Ahmet Reha BOTSALI**

**2018, 72 Sayfa**

**Jüri**

**Dr. Öğr. Üyesi Ahmet Reha BOTSALI**

**Prof. Dr. Mehmet AKTAN**

**Doç. Dr. Saadettin Erhan KESEN**

Bu çalışma üniversitelerde sıklıkla karşılaşılan problemlerinden biri olan sınav çizelgeleme problemleri ele alınmıştır. Sınav çizelgeleme problemleri akademik birimlerin çözüme ulaştırmak istediği problemlerin başında gelmektedir. Bilgiyi test etmede en önemli tekniklerden biri olan sınavların çakışmaması ve belirlenen tarihlerde yapılması önemli bir husustur. Sınav çizelgeleme problemleri NP zor problemler sınıfında değerlendirildiğinden genellikle sezgisel yöntemlerle çözülmeye çalışılmıştır. Bu tez çalışmasında sınav çizelgeleme problemi için öncelikle matematiksel model oluşturulmuştur ve IBM ILOG OPL Optimization Studio yazılımı kullanılarak çözülmüştür. En son aşamada sınav çizelgeleme problemi için C# programlama dilinde geliştirilen tavlama benzetimi algoritmasına ait deney sonuçları karşılaştırılmıştır.

**Anahtar Kelimeler:** Sınav Çizelgeleme Problemi, Tavlama Benzetimi Algoritması, Matematiksel Model, Optimizasyon

**ABSTRACT**

**MS THESIS**

**DIFFERENT APPROACHES FOR EXAM SCHEDULING PROBLEM  
SOLUTION**

**Arafat KOCA**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE  
OF NECMETTİN ERBAKAN UNIVERSITY  
THE DEGREE OF MASTER OF SCIENCE  
IN INDUSTRIAL ENGINEERING**

**Advisor: Assist. Prof. Dr. Ahmet Reha BOTSALI**

**2018, 72 Pages**

**Jury**

**Advisor Assist. Prof. Dr Ahmet Reha BOTSALI**

**Prof. Dr. Mehmet AKTAN**

**Assoc. Prof. Dr. Saadettin Erhan KESEN**

This study deals with exam scheduling problems which are one of the common problems in universities. Exam scheduling problems are one of the problems that academic units want to solve. One of the most important techniques for testing the information is the fact that examinations do not conflict and that they are done on determined dates. As the exam scheduling problems are evaluated in the NP Hard problem class, they have been tried to be solved by using the heuristic methods today. In this thesis study, firstly a mathematical model is prepared for exam scheduling problem and this mathematical is solved using IBM ILOG OPL Optimization Studio software. For the final stage of the exam scheduling problem, a simulated annealing algorithm inC # programming language is developed and the experimental results belonging to the simulated annealing algorithm are compared.

**Keywords:** Exam Scheduling Problem, Simulated Annealing Algorithm, Mathematical Model, Optimization

## **TEŐEKKÜR**

Bu alıőmada Necmettin Erbakan Üniversitesi Mühendislik ve Mimarlık Fakóltesi için Sınav izelgeleme Problemi ele alınmıőtır. alıőma süresince verdiđi destekten dolayı danıőmanım Sayın Ahmet Reha BOTSALI 'ya sonsuz teőekkürlerimi sunarım.

**ARAFAT KOCA**  
**KONYA-2018**

## İÇİNDEKİLER

<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>TEŞEKKÜR</b> .....	<b>iii</b>
<b>İÇİNDEKİLER</b> .....	<b>iv</b>
<b>ŞEKİLLER DİZİNİ</b> .....	<b>vi</b>
<b>TABLolar DİZİNİ</b> .....	<b>vii</b>
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	<b>viii</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. SINAV ÇİZELGELEME PROBLEMİ</b> .....	<b>2</b>
<b>3. KAYNAK ARAŞTIRMASI</b> .....	<b>3</b>
<b>4. MATERYAL VE YÖNTEM</b> .....	<b>9</b>
4.1. Matematiksel Model.....	9
4.1.1. Kullanılan Parametreler .....	9
4.1.1. İndis Kümeleri .....	10
4.1.1. Değişkenler .....	10
4.1.1. Parametreler .....	10
4.1.1. Kısıtlar .....	10
4.2. IBM ILOG OPL Optimization Studio Yazılımında Kullanılan Model.....	11
4.3. IBM ILOG OPL Optimization Studio Yazılımında Kullanılan Veriler.....	15
4.4. Materyal .....	16
4.4.1. Tavlama benzetimi (Simulated Annealing)Algoritması .....	16
4.4.1. Tavlama benzetimi Algoritması Nasıl Çalışır?.....	17
4.4.1. Tavlama benzetimi Algoritması Adımları .....	18
4.5. Yöntem .....	19
4.5.1. Sınav Çizelgeleme Problemi Tanımı .....	19
4.5.2. Sınav Çizelgeleme Problemi ile İlgili Kısıtlar .....	19
4.5.3. Sınav Çizelgeleme Problemi için Geliştirilen Uygulama .....	19
4.5.4. Sınav Çizelgeleme Problemi için Tavlama benzetimi Algoritması .....	20
<b>5. ARAŞTIRMA BULGULARI VE TARTIŞMA</b> .....	<b>21</b>
5.1. Sınav Çizelgeleme Problemi için Geliştirilen Uygulamalar .....	21
5.2. Sınav Çizelgeleme Problemine ait Deneysel Sonuçlar .....	21
<b>6. SONUÇ VE ÖNERİLER</b> .....	<b>47</b>
<b>KAYNAKÇA</b> .....	<b>48</b>
<b>EKLER</b> .....	<b>51</b>



## ŞEKİLLER DİZİNİ

Şekil 4. 1. IBM ILOG OPL Optimization Studio Modeli-I.....	11
Şekil 4. 2. IBM ILOG OPL Optimization Studio Modeli-II .....	12
Şekil 4. 3. IBM ILOG OPL Optimization Studio Modeli-III.....	12
Şekil 4. 4. IBM ILOG OPL Optimization Studio Modeli-IV.....	12
Şekil 4. 5. IBM ILOG OPL Optimization Studio Modeli-V .....	12
Şekil 4. 6. IBM ILOG OPL Optimization Studio Modeli-VI.....	13
Şekil 4. 7. IBM ILOG OPL Optimization Studio Modeli-VII .....	13
Şekil 4. 8. IBM ILOG OPL Optimization Studio Modeli-VIII.....	13
Şekil 4. 9. IBM ILOG OPL Optimization Studio Modeli-IX.....	13
Şekil 4. 10. IBM ILOG OPL Optimization Studio Modeli-X .....	14
Şekil 4. 11. IBM ILOG OPL Optimization Studio Modeli-XI.....	14
Şekil 4. 12. IBM ILOG OPL Optimization Studio Modeli-XII .....	14
Şekil 4. 13. IBM ILOG OPL Optimization Studio Modeli-XIII .....	14
Şekil 4. 14. IBM ILOG OPL Optimization Studio Modeli-XIV.....	14
Şekil 4. 15. IBM ILOG OPL Optimization Studio Verileri-I.....	15
Şekil 4. 16. IBM ILOG OPL Optimization Studio Verileri-II .....	15
Şekil 4. 17. IBM ILOG OPL Optimization Studio Verileri-III.....	16
Şekil 4. 18. IBM ILOG OPL Optimization Studio Verileri-IV.....	16
Şekil 4. 19. Tavlama benzetimi Algoritması Süreci.....	17
Şekil 4. 20. Tavlama benzetimi Algoritması Adımları.....	18
Şekil 5. 1. IBM ILOG OPL Optimization Studio Deney Sonuçları.....	21
Şekil 5. 2. 1. Parametre Seti Deney Sonuçları.....	23
Şekil 5. 3. 2. Parametre Seti Deney Sonuçları.....	24
Şekil 5. 4. 3. Parametre Seti Deney Sonuçları.....	25
Şekil 5. 5. 4. Parametre Seti Deney Sonuçları.....	26
Şekil 5. 6. 5. Parametre Seti Deney Sonuçları.....	27
Şekil 5. 7. 6. Parametre Seti Deney Sonuçları.....	28
Şekil 5. 8. 7. Parametre Seti Deney Sonuçları.....	29
Şekil 5. 9. 8. Parametre Seti Deney Sonuçları.....	30
Şekil 5. 10. 9. Parametre Seti Deney Sonuçları.....	31
Şekil 5. 11. 10. Parametre Seti Deney Sonuçları.....	32
Şekil 5. 12. 11. Parametre Seti Deney Sonuçları.....	33
Şekil 5. 13. 12. Parametre Seti Deney Sonuçları.....	34
Şekil 5. 14. 13. Parametre Seti Deney Sonuçları.....	35
Şekil 5. 15. 14. Parametre Seti Deney Sonuçları.....	36
Şekil 5. 16. 15. Parametre Seti Deney Sonuçları.....	37
Şekil 5. 17. 16. Parametre Seti Deney Sonuçları.....	38
Şekil 5. 18. 17. Parametre Seti Deney Sonuçları.....	39
Şekil 5. 19. 18. Parametre Seti Deney Sonuçları.....	40
Şekil 5. 20. 19. Parametre Seti Deney Sonuçları.....	41
Şekil 5. 21. 20. Parametre Seti Deney Sonuçları.....	42
Şekil 5. 22. 21. Parametre Seti Deney Sonuçları.....	43
Şekil 5. 23. 22. Parametre Seti Deney Sonuçları.....	44
Şekil 5. 24. 23. Parametre Seti Deney Sonuçları.....	45
Şekil 5. 25. 24. Parametre Seti Deney Sonuçları.....	46
Şekil 5. 26. 25. Parametre Seti Deney Sonuçları.....	47

## TABLULAR DİZİNİ

<b>Tablo 4. 1.</b> Tavlama benzetimi Algoritması için sözde kod .....	20
<b>Tablo 5. 1.</b> 1. Parametre Seti Deney Sonuçları .....	23
<b>Tablo 5. 2.</b> 2. Parametre Seti Deney Sonuçları .....	24
<b>Tablo 5. 3.</b> 3. Parametre Seti Deney Sonuçları .....	25
<b>Tablo 5. 4.</b> 4. Parametre Seti Deney Sonuçları .....	26
<b>Tablo 5. 5.</b> 5. Parametre Seti Deney Sonuçları .....	27
<b>Tablo 5. 6.</b> 6. Parametre Seti Deney Sonuçları .....	28
<b>Tablo 5. 7.</b> 7. Parametre Seti Deney Sonuçları .....	29
<b>Tablo 5. 8.</b> 8. Parametre Seti Deney Sonuçları .....	30
<b>Tablo 5. 9.</b> 9. Parametre Seti Deney Sonuçları .....	31
<b>Tablo 5. 10.</b> 10. Parametre Seti Deney Sonuçları .....	32
<b>Tablo 5. 11.</b> 11. Parametre Seti Deney Sonuçları .....	33
<b>Tablo 5. 12.</b> 12. Parametre Seti Deney Sonuçları .....	34
<b>Tablo 5. 13.</b> 13. Parametre Seti Deney Sonuçları .....	35
<b>Tablo 5. 14.</b> 14. Parametre Seti Deney Sonuçları .....	36
<b>Tablo 5. 15.</b> 15. Parametre Seti Deney Sonuçları .....	37
<b>Tablo 5. 16.</b> 16. Parametre Seti Deney Sonuçları .....	38
<b>Tablo 5. 17.</b> 17. Parametre Seti Deney Sonuçları .....	39
<b>Tablo 5. 18.</b> 18. Parametre Seti Deney Sonuçları .....	40
<b>Tablo 5. 19.</b> 19. Parametre Seti Deney Sonuçları .....	41
<b>Tablo 5. 20.</b> 20. Parametre Seti Deney Sonuçları .....	42
<b>Tablo 5. 21.</b> 21. Parametre Seti Deney Sonuçları .....	43
<b>Tablo 5. 22.</b> 22. Parametre Seti Deney Sonuçları .....	44
<b>Tablo 5. 23.</b> 23. Parametre Seti Deney Sonuçları .....	45
<b>Tablo 5. 24.</b> 24. Parametre Seti Deney Sonuçları .....	46
<b>Tablo 5. 25.</b> 25. Parametre Seti Deney Sonuçları .....	47

## SİMGELER VE KISALTMALAR

NP	Nondeterministic Polynomial
VLSI	Very Large Scale Integration
T	Başlangıç Sıcaklık Değeri
minT	Minimum Sıcaklık Değeri
coolRate	Soğutma Oranı



## 1.GİRİŞ

Üniversitelerin akademik birimleri akademik eğitim içerisinde belli bir zaman diliminde sınavları yapmak üzere bir sınav çizelgesi hazırlarlar. Bu durum üzerinde önemle durulması gereken zor bir problemi de beraberinde getirmektedir. Birçok etkeni dikkate alınarak hazırlanan sınav çizelgeleri belirlenen zaman içerisinde en olurlu çözümü verecek şekilde olmalıdır.

Sınav çizelgelemesi yapılırken dersliklerin kapasite durumlarını, dersi alan öğrenci sayısını, sınav sürelerinin her ders için değişkenlik gösterebileceğini, bazı derslerin sınavlarını sadece ilgili teknik laboratuvarlarda yapılması gerektiğini, sınavda görevlendirilecek gözetmen sayısını, gözetmenlerin ve ilgili öğretim üyelerinin özel istekleri de dikkate alınarak hazırlanmalıdır.

Sınav çizelgeleme problemini sürekli artan öğrenci sayısı, öğrencilerin dersleri alttan ve üstten alma durumlarının ortaya çıkardığı değişken durumlar, yeterli akademisyenin ve gözetmenin olmaması, dersliklerin yeterli kapasitede olmayışı, iki sınav arası yeterli hazırlanma ve dinlenme süresinin olmaması, sınavların kısıtlı zamanda çakışabilecek olması daha karmaşık hale getirmektedir. Bu da problemin çözümünü zorlaştırmaktadır.

Bu çalışmadaki sınav çizelgeleme problemi için öncelikle matematiksel model oluşturulmuştur. Bu matematiksel modelde ilgili parametreler, değişkenler ve kısıtlar tanımlanmıştır. Sonraki aşamada bu çizelgeleme problemi IBM ILOG OPL Optimization Studio yazılımı kullanılarak model oluşturulmuştur. Bu modele ilişkin çözüme ulaşılması hedeflenmiştir.

Sezgisel algoritmalar, çizelgeleme problemlerinin çözümü için sıklıkla başvurulan yöntemlerin başında gelmektedir. Bu çalışmada sezgisel algoritmalarından olan tavlama benzetimi algoritması ile sınav çizelgeleme problemi çözüme ulaştırılması amaçlanmıştır. Tavlama benzetimi algoritması ile sınav çizelgeleme problemine uygun bir algoritma geliştirmek bu problemin çözümüne giden süreçte en önemli aşamalardan biridir. Tavlama benzetimi algoritması geliştirmek ve bu geliştirilen algoritma sayesinde elde edilen deneysel sonuçları elde etmemize imkân tanıyan C# programlama dili ve Microsoft Visual Studio yazılımı bu çalışmada kullanılmıştır.

## 2. SINAV ÇİZELGELEME PROBLEMİ

Sınav Çizelgeleme Problemi üniversitelerde ve literatürde sıkça karşılaşılan akademik problemlerdendir. Fazla sayıdaki kısıtların değerlendirildiği fazla sayıdaki parametre ve değişkenin sınav çizelgeleme problemine dahil olması problemin çözümünü zorlaştırmaktadır.

Sınav çizelgeleme problemi oluşturulurken ele alınan kısıtlar yumuşak ve sert kısıtlar olarak tanımlanmaktadır. Sert kısıtlar öncelikle sağlanması amaçlanan kısıtlar olurken yumuşak kısıtlar sert kısıtlar sağlandıktan sonra sınav çizelgeleme probleminin çözüm sınırları içerisinde imkan tanıdığı ölçüde sağlanması amaçlanır.

Sınav çizelgeleme oluşturulurken ilgili üniversitede yapılacak olan sınavların, bu sınavlara girecek öğrenci sayılarının ve sınav kapasitelerinin belirtilmesi gerekir. Üniversitenin ilgili akademik birimde dikkate alınan döneme ait sınav verileri problemin sınırlarını oluşturmaktadır.

Sınav çizelgesinde sınavların bu çizelgede hangi günlerde ve zaman dilimlerinde yapılacağı ve ilgili sınavların hangi sınav salonlarında yapılacağı belirlenmelidir. Çok sayıda öğrenci grubunun gireceği sınavların atamasının bu çizelgede en iyi şekilde yapılması hedeflenmektedir.

Sınav çizelgeleme problemine göre tasarlanan matematiksel modelin ve geliştirilen optimizasyon tabanlı sezgisel algoritmanın ilgili akademik birimlerin verileriyle test edilmesi ve istenilen en uygun çözümün açığa çıkarılması gerekir.

Sınav çizelgeleme problemleri literatürde başlarda daha basit ve klasik modeller oluşturularak çözülmeye çalışırken günümüze doğru geldiğimizde daha çok geliştirilen sezgisel algoritmalarla çözüme ulaşılmaya çalışılmıştır.

Sınav çizelgeleme problemleri NP zor problemler sınıfında değerlendirildiğinden bugün sezgisel yöntemlerle çözülmeye çalışılmıştır. Mümkün olduğunca tüm kısıtların sağlandığı ve en iyi çözümün elde edileceği düşünüldüğü sezgisel yöntemler kullanılarak geliştirilen algoritmalar sayesinde daha iyi çözümler elde edilmeye çalışılmıştır.

Bu tez çalışmasında sınav çizelgeleme problemleri için oluşturulan matematiksel model ve verilere ait matematiksel yöntem ile C# programlama dilinde tasarlanan model ve geliştirilen ve sezgisel bir algoritma olan tavlama benzetimi algoritması arasındaki deney sonuçları karşılaştırılmıştır.

### 3.KAYNAK ARAŞTIRMASI

Bu bölümde sınav ve zaman çizelgeleme problemleri için çözüm yaklaşımları literatürde araştırılmış ve belirtilmiştir.

Abbas ve Tsang (Abbas & Tsang, 2004) çalışmalarında, kısıt tekniğine vurgu yapmışlardır. Sezgisel bilginin ve kısıtlamaların formüle edilmesini yazılım mühendisliğine yatkın bir dille ifade etmişlerdir. Kullanıcı ihtiyaçları yakından incelenerek kısıtlamaları karşılayan bir çözümün mümkün olmadığı durumlarda problemin kısıtlarını gevşeterek veya yeniden formüle ederek çözüme gitmişlerdir.

Skoullis, Tassopoulos ve Beligiannis (Skoullis, Tassopoulos, & Beligiannis, 2017) çalışmalarında, okul çizelgeleme problemine göre tasarlanmış ve uygulanmış hibrid optimizasyon tabanlı bir algoritma geliştirmişlerdir. Algoritmayı gerçek dünya verileri ile test etmişlerdir.

Arbaoui, Boufflet ve Moukrim (Arbaoui, Boufflet, & Moukrim, 2016) çalışmalarında, uygun bir çözüm bulmak amacıyla sınav çizelgeleme problemine ilişkin doğrusal olmayan bir formülasyon önermişlerdir. Eski lineer formülasyon ve yeni kompakt lineer formülasyon oluşturup, bu kompakt formülasyon ile değişken ve kısıtlamaların sayısını % 99 oranında azalttığını belirtmişlerdir. Matematiksel programlamaya dayalı her iki formülasyonu da dikkate alarak sezgisel bir yöntemin problemi daha iyi sonuçlara ulaştırabileceğini söylemişlerdir.

Arani, Karwan ve Lofti (Arani, Karwan, & Lofti, 1997) çalışmalarında, sınavların gruplara ayrılması, bu grupların sınav günlerine atanması ve sınav günlerinin ve sınav gruplarının düzenlenmesi olarak üç aşamada sınav planlama problemini tanımlamışlardır. Her gün iki veya daha fazla sınavı o plan öğrenci sayısını azaltmayı amaçlamışlardır. Sınav planlama problemini Lagrangian gevşeme yaklaşımı ile çözmeye çalışmışlardır.

Gans (Gans, 1981) çalışmasında, zaman çizelgesi problemini Hollanda okul organizasyonu ana hatlarıyla belirlemiştir. Veri yapısı zaman çizelgesi problemini çözmek için kombinatoryal/sezgisel bir yazılım sisteminde kullanmıştır.

Papoulias (Papoulias, 1980) çalışmasında, haftalık okul zaman çizelgesi gereksinimlerini beş İngiliz okulu kullanılarak pratik bir değerlendirme yapmıştır. Sonuçları tartışmış ve çözümün iyileştirilmesi için önerilerde bulunmuştur.

Pillay ve Bahnzaf (N.Pillay & W.Banzhaf, 2009) çalışmalarında zaman çizelgesi oluştururken sınavları çizelgelemek için en iyi sezgisel kombinasyonları bulmuşlardır ve hiper sezgisel sistemler uygulamışlardır. Hiyerarşik sezgisel kombinasyonların alternatif

tasarımlarını ileri sürdükleri çalışmada her sezgisel kombinasyon sıralıdan ziyade aynı anda uygulamışlardır.

Werra (Werra, 1985) çalışmasında, zaman çizelgesi modellerinin bir okulun haftalık programından bir üniversitedeki derslerin veya sınavların planlamasına kadar uzanabileceğini söylemiştir. Grafik ve ağlar böyle problemlerin çözümünde faydalı olabileceğini savunmuştur. Grafik teorik modeller üzerinde çeşitli modelleri tanımlayacağını belirtmiştir.

Lü ve Hao (Lü & Hao, 2010) çalışmalarında müfredata dayalı ders çizelgeleme problemini çözmek için başlatma, yoğunlaştırma ve çeşitlendirme olmak üzere üç aşamadan oluşan tabu arama algoritması önermişlerdir. İlk aşamada hızlı bir sezgisel kullanarak ilk zaman çizelgesi yapısı oluşturulmuşlardır. Sonrasında zor kısıtların sağlanırken yumuşak kısıtların ihlal edildiği durumların sayısını azaltmak için yoğunlaştırma ve çeşitlendirme kullanmışlardır.

Lewis ve Thompson (Lewis & Thompson, 2015) çalışmalarında, matematiksel işlemle sağlanan ders çizelgeleme problemini yaklaşık olarak çözmek için iki aşamalı güçlü bir meta sezgisel tabanlı bir algoritma sunmuşlardır.

Hertz (Hertz, 1991) çalışmasında, tabu arama tekniklerini zaman çizelgesi problemine uyarlamıştır. Aynı anda meydana gelen ve aynı sınıf gereken veya ortak öğrenci ve öğretmen içeren derslerin sebep olduğu çatışmaların sayısını azaltmayı amaçlamıştır. Klasik kısıtlara ek olarak kompaktlık ve öncelik gereksinimlerin ve coğrafi kısıtların ele alınması gerektiğini söylemiştir. Ayrıca çok sayıda öğrencinin aldığı derslerin haftada birkaç kez tekrarlanması gerektiğini ifade etmiştir.

Carrasco ve Pato (Carrasco & Pato, 2014) çalışmalarında, sinirsel ağ tabanlı sezgisel uygulamaların sınıf/öğretmen zaman çizelgesi problemine uygulanmasını araştırmışlardır. Problemin karakteristiğini sert ve yumuşak kısıtlar açısından değerlendirmişlerdir. Problemi yapay sinir ağları modeli içinde konumlandırmışlar ve gerekli olan enerji fonksiyonunu formüle etmişlerdir. Optimizasyon problemlerinde olumlu sonuçlar bulan tavlama simülasyonu kullanmışlardır. Hem sezgisel hem de gerçek ve kuramsal örneklerden alınan sonuçları açıklamışlardır. Bu ayırık yaklaşımın çözüm kalitesine daha fazla katkıda bulunacağını vurgulamışlardır.

Causmaecker, Demeester ve Berghe (Causmaecker, Demeester, & Berghe, 2009) çalışmalarında, üniversite ders çizelgeleme problemini çözmek için ayrılmış bir meta sezgisel yaklaşım sunmuşlardır. Yaklaşımındaki ilk aşama sütunlar olarak isimlendirilmiş bu yeni yapıları sunarak öznelerin sayısının azaltılmasıdır. Sonraki aşamalarda tüm

kısıtlar için tek seferde çözüm elde etmek yerine kısıtları tek tek çözmeye çalışan bir meta sezgisel arayışın olması gerektiğini savunmuşlardır.

Zhang, Liu, M'Hallah ve Leung (Zhang, Liu, M'Hallah, & C.H.Leung, 2010) çalışmalarında, yeni tasarlanmış bir komşu yapısıyla tavlama benzetimi tabanlı algoritma kullanarak lise ders çizelgeleme problemin çözmeye çalışmışlardır. En iyi komşuyu araştırmak için standart tavlama benzetiminde iki atamayı değiştirmek yerine zaman dilimleri çiftleri arasında değişimi uygulamışlardır. Bilimsel sonuçlar iki kıyaslama örneği kümesinde test ettikleri sezgisel yaklaşımın mevcut yaklaşımlardan daha iyi sonuç vereceğini belirtmişlerdir.

Saviniec, Santos ve Costa (Saviniec, O.Santos, & M.Costa, 2018) çalışmalarında, spesifik yumuşak gereksinimleri en aza indirmek amacıyla öğretmenlerin sınıflara atandığı bu ders çizelgelemede iki farklı paralel yapı kabul etmişlerdir. Performansla yakından ilişkili olan algoritmik kararları anlamak için hesaplamalı çalışma sunmuşlardır. En iyi algoritmaları, yöntemin hem verimliliğini hem de esnekliğini göstermişlerdir.

Dammak, Elloumi ve Kamoun (Dammak, Elloumi, & Kamoun, 2006) çalışmalarında, belirli bir kapasiteye sahip sınıflara atanan bir dizi bağımsız sınavın problemini ele almışlardır. Her bir sınıfa birden fazla sınavın atanmadığı ve bu kısıtların rahat olduğu durumlarda problemi 0-1 doğrusal tamsayı programı olarak formüle etmişlerdir. Çeşitli sezgisel yöntemler kullanmışlardır. Çatışmasız sınavlardan oluşan bir zaman çizelgesi girdi olarak değerlendirmişler ve problemi çözmek için basit bir sezgisel prosedür geliştirmişlerdir.

Leite, Fernandes, Melicio ve Rosa (Leite, Fernandes, Melício, & Rosa, 2018) çalışmalarında, zaman çizelgesi probleminin çözümü için hücrel bir memetik algoritma önermişlerdir. Önerilen hücrel evrimsel algoritmaya eşik kabul eden yerel arama meta sezgiseli melezleştirmişlerdir. Uygulanan algoritma uygulanabilir genetik rekombinasyonları ve yerel arama operatörlerini kullanarak uygun çözüm alanını sınırlandırmayı amaçlamışlardır.

Li, Bai, Shen ve Qu (Li, Bai, Shen, & Qu, 2015) çalışmalarında arama süreci sırasında değişen ortamlara uyum sağlamaya çalışan yeni bir arama yöntemi sunmuşlardır. Tek bir çözümde en güçlü mekanizmanın varlığını devam ettirebilmesi için evrimsel ruin evresini uygulayarak orijinal ruin ve yeniden yaratma fikrini geliştirmişlerdir. Bileşen değerlendirme, çözümün bozulması ve stokastik yapıcı durumu tekrarlayan bir süreçle optimizasyonu sağlamışlardır. Yaklaşımın bazı teorik özelliklerine

markov zincir analizi uygulamışlardır. Sınav çizelgeleme problemi üzerinde deneysel sonuçlar güçlü bir yaklaşım olduğunu göstermişlerdir.

Balakrishnan (Balakrishnan, 1991) çalışmasında, sınav planlamada aynı dönemde hiçbir öğrencinin birden fazla sınava girmemesi amaçlanırken, gerçek hayatta bu durumu daha karmaşık hale getiren kısıtlama sahip olduğunu belirtmiştir. Çizelgeleme problemiyle ilgili grafik renklendirme tabanlı bir inceleme çizelgelemesi uygulamıştır.

Barrera, Valesco ve Amaya (Barrera, Velasco, & Amaya, 2012) çalışmalarında, çoklu etkinlik kombine zaman çizelgesini ve ekip çizelgeleme problemini tanıtmışlardır. Müşteri odaklı hizmetlerle bir işgücü çizelgeleme problemi ortaya koymuşlardır. Gerekli çalışanları minimize edebilmek, iş yüklerini dengelemeyi ve talepleri karşılamayı amaçlamışlardır. Bu problemi modellemek için ağ tabanlı bir yaklaşım önermişlerdir. İki çözüm stratejisi sunmuşlardır. Birincisi matematiksel programlamaya dayandırmışlar ikincisi hesaplama süresini azaltmak için sezgisel bir prosedür kullanmışlardır.

Weitz ve Lakshminarayanan (Weitz & Lakshminarayanan, 1997) çalışmalarında, VLSI tasarımdan alınan grafik teorik yaklaşımları final sınavı planlamasından ve maksimum çeşitlilik grup problemlerinden kaynaklanan buluşsal yöntemlerle ampirik olarak karşılaştırmışlardır.

Woumans, Boeck, Beliën ve Creemers (Woumans, Boeck, Beliën, & Creemers, 2016) çalışmalarında, öğrenci odaklı bir bakış açısıyla sınav zaman çizelgeleme problemine yaklaşmışlardır. Öğrenciler için sınavların dağılımını artırmak amacıyla bir sınavın birden fazla versiyonunun programlanmasını sağlamışlardır. Sınav zaman çizelgesi problemini çözmek için iki sütun algoritması önermişlerdir.

Chern, Chien ve Chen (Chern, Chien, & Chen, 2008) çalışmalarında sağlık muayenesi planlama problemini ele almışlardır. Sağlık muayenesi planlama algoritmasını önermişlerdir. Çünkü ikili tamsayı programlama modeli problemi çözmek için uygun bir yol gibi görünse de sağlık muayene sayısı arttıkça çözüm zorlaşacağını söylemişlerdir. Sağlık muayenesi planlama problemi etkin ve verimli bir şekilde çözmek için sezgisel bir algoritma tasarlamışlardır. Bu algoritmanın sağlık muayenesi planlama problemini çözmede etkili olduğunu belirtmişlerdir.

Amaral ve Pais (Amaral & Pais, 2016) çalışmalarında, sınavların sınav dönemi boyunca dağılımının karakterizasyonu için dört ölçüt kullanmışlardır. Ortak öğrencilerle yapılan sınavların çakışmaması için ilgili kısıtları göz önünde bulundurmuşlardır. Dört kriteri de ele almak için çok amaçlı bir optimizasyon programı kullanmışlardır. Probleme çözüm bulmak amacıyla Tabu arama uygulamışlardır.

Kahar ve Kendall (Kahar & Kendall, 2010) çalışmalarında Malezya'dan gerçek dünyadaki kapasiteli bir sınav zaman çizelgesi problemi sunmuşlardır. Problemin daha önce modellenmemiş kısıtları uygun bir model tanımlanmasında ve yapıcı bir sezgisel geliştirilmesinde ek zorluklar sağladığını belirtmişlerdir.

Muklason, Parkes, Özcan, McCollum ve McMullan (Muklason, Parkes, Özcan, McCollum, & McMullan, 2017) çalışmalarında, mevcut tüm değerlendirme ölçütlerini karşılayan çözümler üretmek amacıyla sınav takvimi ile ilgili öğrenci tercihlerini araştırmışlardır. Önerdikleri model ve yöntemlerin zaman çizelgesine yüksek kalitede çözümler üretmişlerdir. Standart yumuşak kısıtlar ile her öğrenci için istenenler arasında bir dengenin sağlandığını belirtmişlerdir.

White ve Haddad (White & Haddad, 1983) çalışmalarında, aynı anda iki veya daha fazla sınavın olmaması gerektiğini açıklamışlardır. Art arda gelen sınav sayısının sezgisel olarak azaltmayı amaçlamışlardır. Belirli uygun alt kümelerin sınav zaman aralıklarına uygun alt kümeler halinde planlanması gerektiğini belirtmişlerdir.

Qu, Burke ve McCollum (Qu, Burke, & McCollum, 2009) çalışmalarında, farklı nitelikte çözümler üretebilmek amacıyla rastgele tekrarlayan bir grafik tabanlı hiper sezgisel yöntem sunmuşlardır. Adım adım çözümler oluşturan farklı grafik renklendirme sezgilerini istatistiksel olarak analiz etmişlerdir. Sezgisel tasarım sürecini otomatik hale getirmeyi amaçlamışlardır. Kıyaslama sınav çizelgeleme ve grafik renklendirme problemini ilişkin çözüm yapısı farklı aşamalarda hibritlemek için bir hibrit yaklaşım geliştirmişlerdir.

Dimopoulou ve Miliotis (Dimopoulou & Miliotis, 2001) çalışmalarında, birleştirilmiş bir üniversite ders-sınav çizelgesinin yapısına yardımcı olmak için bilgisayar tabanlı bilgisayar sistemi tasarlamış ve uygulamışlardır. Tamsayı programlama modeli kullanmıştır. Model kullanıcı tarafından belirlenmiş varsayımlara karşılık gelen kısıtlamalar üretmişlerdir. Üretilen programın kalitesi mevcut zaman periyotlarına atanan derslerin pozisyonuna bağlı olduğunu söylemişlerdir. Ders ve sınav çizelgeleme probleminin çözümü için sezgisel algoritmayı geliştirmek gerektiğini belirtmişlerdir.

Broek, Hurkens ve Woeginger (Broek, Hurkens, & Woeginger, 2009) çalışmalarında, gerçek dünyadaki çizelgeleme problemin çözümünü tartışmışlardır. Tam bir matematiksel formül sunmuşlardır. Eindhoven'daki durumdan kaynaklanan tüm kısıtları açıklamışlardır. Problemi dört alt problemle birlikte optimizasyon kullanarak çözmeye çalışmışlardır. Dört alt problemin tamamı için Cplex'e kolayca yerleştirilebilen bir tamsayı doğrusal programlama modeli oluşturmuşlardır.

Head ve Shaban (Head & Shaban, 2007) çalışmalarında, programı oluşturmak ve öğrencileri. Aynı anda sınıflara yerleştirmek yaklaşımını savunmuşlardır. Tüm kısıtlama ve gereksinimleri toplayarak bunların miktarını belirten, öğrencileri yerleştiren ve sezgisel fonksiyona dayanan bir program oluşturmuşlardır. Programı oluşturduktan sonra ek sezgisel tabanlı işlemler kullanarak optimize etmeye çalışmışlardır.

Burke, McCollum, Meisels, Petrovic ve Qu (Burke, McCollum, Meisels, Petrovic, & Qu, 2007) çalışmalarında, yaygın bir şekilde kullanılan yapıcı sezgisel bir dizi üzerine basit bir genel hiper sezgisel yaklaşımını araştırmışlardır. Hiper sezgisel bir yapıda sınav ve ders çizelgeleme probleminde zamana çizelgesi oluşturmak için ve grafik sezgiselleri permütasyonlarını araştırmak için bir tabu arama yaklaşımı kullanmışlardır

Al-Yakoob ve Sherali (Al-Yakoob & Sherali, 2015) çalışmalarında, Kuveyt'in eğitim sistemiyle ilgili bir vaka çalışmasında bir lise çizelgeleme problemi araştırmışlardır. Kapsamlı bir karma tamsayı programlama modeli geliştirmişlerdir. İki aşamalı bir modelleme yaklaşımı sunmuşlardır. Haftalık zaman dilimleri belirleyip daha sonrasında öğretmenlerin sınıflara atandığı model oluşturmuşlardır.

Babaei, Karimpour ve Hadidi (Babaei, Karimpour, & Hadidi, 2015) çalışmalarında, üniversite ders çizelgeleme probleminin araştırılmasında mevcut yaklaşımları araştırmışlardır. Dağıtılmış çoklu ajan sistemleri tabanlı yaklaşım ders çizelgeleme problemindeki ölçeklenebilirliği sebebiyle incelemişlerdir. Mevcut algoritma yapısını test etmek değerlendirmek için güvenilir veri kümelerini eksisiz bir şekilde sunmuşlardır.

Sabar, Ayob, Kendall ve Qu (Sabar, Ayob, Kendall, & Qu, 2012) çalışmalarında, zaman çizelgesi problemleri için bir çeşit bal arısı çiftleşme algoritması önermişlerdir. Önerilen algoritmanın performansı, sınav ve ders çizelgeleme problemleri olmak üzere iki alanda gösterilmiştir.

Asmuni, Burke, Garibaldi, McCollum ve Parkes (Asmuni, Burke, Garibaldi, McCollum, & Parkes, 2009) çalışmalarında yüksek kaliteli uygulanabilir sınav çizelgeleme çözümü için bulanık metodolojileri kullanmışlardır. Bulanık yöntemleri bu konuyu ele aldıkları yaklaşımda zaman çizelgesi çözmek amacıyla kullanmıştır.

Burke, Eckersley, McCollum, Petrovic ve Qu (Burke, Eckersley, McCollum, Petrovic, & Qu, 2010) çalışmalarında üniversite sınavı çizelgeleme problemi için değişken komşuluk yaklaşımını araştırmışlardır. Önerilen tekniğin çok çeşitli kıyaslama probleminde iyi çözümler ürettiğini belirtmişlerdir.

## 4.MATERYAL VE YÖNTEM

Bu çalışmada sınav çizelgeleme problemine ilişkin, probleme ait parametrelerin, indislerin, değişkenlerin, kısıtların ve amaç fonksiyonların tanımlandığı matematiksel model geliştirildi. Daha sonrasında sınav çizelgeleme probleminin çözümü için IBM ILOG Cplex Optimization Studio yazılımı kullanılarak sınav çizelgeleme problemine ilişkin model yapısı oluşturuldu. Daha sonra sınav çizelgeleme problemi meta-sezgisel algoritmalarla tavlama benzetim algoritması kullanılarak tekrar analiz edildi.

### 4.1. Matematiksel Model

Sınav çizelgeleme problemi için oluşturulan matematiksel model mimarisi ve modeli oluşturulurken dikkate alınan parametreler, indisler, değişkenler, kısıtlar ve amaç fonksiyonu aşağıda tanımlanmıştır.

#### 4.1.1. Kullanılan Parametreler

- Sınavların yapıldığı günler ( Pazartesi, Salı, Çarşamba, Perşembe, Cuma)
- Sınavların yapıldığı saatler (hafta içi her gün 09.00-11.00-13.00-15.00 saatlerinde başlayan iki saatlik zaman dilimleri)
- Öğrenci grupları (1.sınıf, 2.sınıf, 3.sınıf, 4.sınıf, alttan veya üstten sınavı olmayan düzenli öğrenciler )
- Sınavlar (4 yıllık müfredatta yer alan tüm sınavlar)
- Öğrenci gruplarının gireceği sınavlar (toplam sınav)
- Gözetmen sayısı
- Sınav salonları ve kapasiteleri

#### 4.1.2. İndis Kümeleri

- Sınav indisleri kümesi:  $E = \{e / e=1, \dots, 739\}$
- Öğrenci grubu indisleri:  $G = \{g / g=1, \dots, 52\}$
- Öğrenci sayısı indisleri:  $N = \{n / n = \dots\}$
- Sınıf indisleri kümesi:  $C = \{c / c=1, \dots, 45\}$
- Zaman dilimleri kümesi:  $T = \{t / t=1, \dots, 8\}$
- Gün indisleri kümesi:  $D = \{d / d=1, \dots, 5\}$
- Kapasite indisleri kümesi :  $K = \{k / k = \dots\}$
- Aynı sınava girecek öğrenciler kümesi:  $A = \{a / a = \dots\}$

### 4.1.3. Değişkenler

$$X_{ecdt} = \begin{cases} 1, e. \text{ sınav, } c. \text{ sınıfta, } d. \text{ gün, } t. \text{ zaman diliminde yapılıyorsa} \\ 0, \end{cases} \quad d. d. \}$$

$$L_{gid} = \begin{cases} 1 \text{ eğer } g_i \text{ öğrenci grubunun sınavı } d. \text{ günde yapılıyorsa} \\ 0 \text{ yoksa} \end{cases}$$

### 4.1.4. Parametreler

$K_c$  = c. sınıfın kapasitesi

$S_g$  = {gözetmen sayısı}

$O_e$  = {e. sınıfa sınava giren öğrenci sayısı}

### 4.1.5. Kısıtlar

$$\sum_t^T \sum_d^D \sum_e^E \sum_c^C X_{ecdt} = 1 \quad (1)$$
$$\forall e \in E$$

1. Denklem e. sınavın c. sınıfta d. gün t. zamanda yapılması durumudur. Bu durum herhangi bir sınavın mutlaka herhangi bir günde ve zamanda, herhangi bir sınıfta yapılmasıdır.

$$\sum_c^C O_e * X_{ecdt} \leq K_c \quad (2)$$
$$\forall t \in T \quad \forall e \in E$$

2. Denklem sınavların düzenleneceği sınıfların kapasitelerinin toplamının o sınava giren öğrenci sayıları toplamından fazla olması gerektiren durumdur.

$$\sum_e^E \sum_c^C X_{ecdt} \leq S_g \quad (3)$$
$$\forall t \in T, \quad \forall d \in D$$

3. Denklem herhangi bir t zamanında yapılacak olan sınavların için kullanılacak olan 0 sınıfların sayısının gözetmen sayısından fazla olmaması gerektiği durumudur.

$$\sum_c^C X_{ecdt} = \sum_c^C X_{acdt} \quad (4)$$
$$\forall e, a \in E, \quad \forall t \in T \quad \forall d \in D$$

4. Denklem ortak yapılan e ve a sınavlarının t. zamanda yapılması durumudur.

$$\sum_c^C X_{ecdt} + \sum_c^C X_{acdt} \leq 1 \quad (5)$$

$$\forall t \in T, \quad \forall e, a \in Gg1, \quad \forall d \in D$$

5. Denklem aynı zamanda yapılan e ve a sınavlarına girecek öğrenci grubunun bu iki sınava aynı sınıfta yapılamayacağı durumudur.

$$\sum_e^{Ggi} \sum_c^C \sum_t^T X_{ecdt} \leq 2 \quad (6)$$

$$\forall g \in G \quad \forall gi \in Gg \quad \forall d \in D$$

6. Denklem g. öğrenci grubunun d. gün en fazla iki sınava girmesi durumudur.

$$\sum_e^{Ggi} \sum_c^C \sum_t^T X_{ecdt} - 1 \leq L_{gid} \quad (7)$$

$$\forall g \in G \quad \forall gi \in Gg \quad \forall d \in D$$

7. Denklem gi öğrenci grubunun d. günde fazladan gireceği sınav sayısı durumudur

$$\text{Min} ( \sum_d^D \sum_{gi}^G 100 * L_{gid} ) \quad (8)$$

8. Denklem bir öğrencinin bir günde birden fazla sınava girmesi durumudur. Bu istenmeyen durumun minimize edilmesi gösteren amaç fonksiyonudur.

#### 4.2. IBM ILOG OPL Optimization Studio Yazılımında Kullanılan Model

IBM ILOG OPL Optimization Studio yazılımını kullanarak matematiksel model referans alınarak optimizasyon modeli oluşturuldu. Bir sınav döneminde yapılacak olan sınav sayısı, bu sınavlarda kullanılacak olan sınıf sayısı, bu sınavlara girecek öğrenci grupları, sınavların kaç günde yapılacağı ve hangi zaman dilimlerinde olacağı, sınava girecek olan gözetmenlerin sayısı, ortak yapılamayacak sınavların sayısı bu çalışmada kullanılan yazılımdan alınan Şekil 4.1’de görüldüğü üzere tanımlanmıştır.

```
int Exams=...;
int Classes=...;
int NumberofStudents=...;
int Days=...;
int Times=...;
int Supervisor=...;
int NoCommonExamsPairs=...;
int MaxTest=...;
```

Şekil 4. 1. IBM ILOG OPL Optimization Studio Modeli-I

Sınavların sayısının, sınava girecek olan öğrenci gruplarının sayısının, sınavların yapılacağı günün sayısının ve o günlerdeki zaman dilimlerinin sayısının, sınavların yapılacağı sınavların sayısının alt ve üst sınırları Şekil 4.2’de belirtilmiştir.

```
range Exam=1..Exams;  
range Student=1..NumberofStudents;  
range Day=1..Days;  
range Time=1..Times;  
range Class=1..Classes;
```

**Şekil 4. 2.** IBM ILOG OPL Optimization Studio Modeli-II

Sınavların yapılacağı sınıfların kapasitelerinin tutulduğu dizin ile farklı günde ve zaman diliminde yapılacak tüm öğrencilerin gireceği sınavları gösteren dizin ve her bir öğrenci grubunun aldığı sınavların tutulduğu dizin Şekil 4.3’de belirtilmiştir.

```
int Capacity[k in Class]=...;  
int O[e in Exam]=...;  
int studentexams[s in Student][t in 1..MaxTest]=...;
```

**Şekil 4. 3.** IBM ILOG OPL Optimization Studio Modeli-III

Sınıfların ve kapasitelerinin kısıtlı olması, kısa zamanda çok sınavın yapılması, gözetmen sayısının azlığı nedeniyle bazı sınavlar ortak yapılmak zorundadır. Bu durum Şekil 4.4’te belirtilmiştir.

```
tuple CommonExamPair {  
    int a;  
    int b;  
};  
  
{CommonExamPair} commonexams=...;
```

**Şekil 4. 4.** IBM ILOG OPL Optimization Studio Modeli-IV

Belirli bir sınavın, belirli bir sınıfta, belirli gün ve zamanda mutlaka yapılması durumu ile öğrenci gruplarının bir günde fazladan gireceği sınavların durumu olan karar değişkenleri Şekil 4.5’te belirtilmiştir.

```
dvar int X[Exam][Class][Day][Time] in 0..1;  
dvar int L[Student][Day] in 0..1;
```

**Şekil 4. 5.** IBM ILOG OPL Optimization Studio Modeli-V

Kısa zamanda çok sınavın yapılması sınıf sayısının belli bir sayıda oluşu, sınıf kapasitelerinin ve gözetmenlerin azlığının oluşturduğu değişken ve istenmeyen durumlar öğrenci gruplarının bir günde en fazla bir sınava girmesini zorlaştırıyor. Bazı öğrenci grupları bir günde birden fazla sınava girebiliyor. Bu istenilmeyen bir durumdur.

Bu istenilmeyen durumun minimize edilmesi gerekir. Şekil 4.6’da görüldüğü üzere bu durumu minimize edildiği amaç fonksiyonu tanımlanmıştır. Bu amaç fonksiyonunda bir öğrenci grubunun gün içerisinde birde fazla gireceği her sınav 100 ceza katsayısı ile cezalandırılmıştır.

```
minimize  
sum (d in Day,n in Student) 100*L[n][d];
```

**Şekil 4. 6. IBM ILOG OPL Optimization Studio Modeli-VI**

Şekil 4.7’de belirtilen ilk kısıt e. sınavın c. sınıfta d. gün t. zamanda yapılması durumudur. Bu durum her hangi bir sınavın mutlaka her hangi bir günde ve zamanda, herhangi bir sınıfta yapılmasıdır.

```
subject to {  
  
forall (c in Class,d in Day,t in Time)  
    sum (e in Exam) X[e][c][d][t]<=1;
```

**Şekil 4. 7. IBM ILOG OPL Optimization Studio Modeli-VII**

Şekil 4.8’de belirtilen ikinci kısıt sınavların düzenleneceği sınıfların kapasitelerinin toplamının o sınava giren öğrenci sayıları toplamından fazla olması gerektiren durumudur.

```
forall (e in Exam, t in Time, d in Day, c in Class )  
    secondequation: X[e][c][d][t]*O[e] <= Capacity[c];
```

**Şekil 4. 8. IBM ILOG OPL Optimization Studio Modeli-VIII**

Şekil 4.9’da belirtilen üçüncü kısıt herhangi bir t zamanında yapılacak olan sınavların için kullanılacak olan sınıfların sayısının gözetmen sayısından fazla olmaması gerektiği durumudur.

```
forall (t in Time,d in Day)  
    thirdequation:sum (e in Exam,c in Class) X[e][c][d][t]<= Supervisor;
```

**Şekil 4. 9. IBM ILOG OPL Optimization Studio Modeli-IX**

Şekil 4.10’da belirtilen dördüncü kısıt ortak yapılan e ve a sınavlarının t. zamanda yapılması durumudur.

```
forall (t in Time,d in Day, <a,e> in commonexams)
  fourthequation: sum (c in Class) X[e][c][d][t]==sum (c in Class) X[a][c][d][t];
```

**Şekil 4. 10.** IBM ILOG OPL Optimization Studio Modeli-X

Şekil 4.11’de belirtilen beşinci kısıt aynı zamanda yapılan e ve a sınavlarına girecek öğrenci grubunun bu iki sınava aynı sınıfta yapılamayacağı durumudur.

```
forall (s in Student,t in Time,d in Day,a in 1..MaxTest: studentexams[s][a]>0)
  forall(e in 1..MaxTest: studentexams[s][e]>0)
  fifthequation: sum (c in Class:a>e) X[studentexams[s][e]][c][d][t]
    + sum (c in Class)X[studentexams[s][a]][c][d][t]<=1;
```

**Şekil 4. 11.** IBM ILOG OPL Optimization Studio Modeli-XI

Şekil 4.12’de belirtilen altıncı kısıt herhangi bir öğrenci grubunun d. gün en fazla iki sınava girmesi durumudur.

```
forall (s in Student, d in Day)
  sixthequation:sum (e in 1..MaxTest, c in Class, t in Time:
    studentexams[s][e]>0)X[studentexams [s][e]][c][d][t]<=2;
```

**Şekil 4. 12.** IBM ILOG OPL Optimization Studio Modeli-XII

Şekil 4.13’te belirtilen yedinci kısıt herhangi bir öğrenci grubunun d. günde fazladan gireceği sınav sayısı durumudur.

```
forall (s in Student, d in Day)
  seventhequation:sum (e in 1..MaxTest, c in Class, t in Time:
    studentexams[s][e]>0)X[studentexams[s][e]][c][d][t]-1<=L[s][d];
```

**Şekil 4. 13.** IBM ILOG OPL Optimization Studio Modeli-XIII

Şekil 4.14’te tüm sınavların tüm gün ve zaman dilimi ile tüm sınıfların farklı birleşimlerinden oluşan atama işlemi gösterilmiştir.

```
execute DISPLAY {
  for (var e in Exam)
  for(var t in Time)
  for(var d in Day)
  for(var c in Class)
    if(X[e][c][d][t]>0)   writeln("X[" ,e, "]" ,c, "]" ,d, "]" ,t, "]" = 1");
}
```

**Şekil 4. 14.** IBM ILOG OPL Optimization Studio Modeli-XIV



```

studentexams=[[10 65 128 189 227 303 340 370 -1 -1] [32 73 142 197 239 271 326 -1 -1 -1]
[29 70 111 159 250 287 360 375 -1 -1] [15 84 137 175 236 279 344 -1 -1 -1]
[21 94 141 201 249 270 334 365 -1 -1] [24 55 122 188 222 265 313 -1 -1 -1]
[9 54 106 176 215 295 316 374 -1 -1] [13 64 110 161 217 288 323 -1 -1 -1]
[39 99 145 195 240 264 324 373 -1 -1] [3 74 130 177 229 280 331 -1 -1 -1]
[44 79 119 166 219 277 352 367 -1 -1] [33 69 123 168 210 272 333 -1 -1 -1]
[22 104 132 208 254 296 361 369 -1 -1] [42 66 134 192 232 282 339 -1 -1 -1]
[47 89 151 178 233 312 351 -1 -1 -1] [16 97 114 193 257 301 337 -1 -1 -1]
[2 60 136 174 242 308 332 -1 -1 -1] [50 85 124 181 225 274 347 -1 -1 -1]
[36 103 112 182 220 285 362 -1 -1 -1] [25 59 127 164 247 302 320 -1 -1 -1]
[30 80 152 199 253 276 314 371 -1 -1] [6 93 149 206 226 267 348 -1 -1 -1]
[14 61 107 162 211 268 363 368 -1 -1] [34 75 115 172 256 306 329 -1 -1 -1]
[51 95 138 204 260 311 364 366 -1 -1] [17 102 140 170 243 298 318 -1 -1 -1]
[40 71 154 194 218 304 359 372 -1 -1] [49 58 150 205 213 307 358 -1 -1 -1]
[8 81 120 186 216 310 322 -1 -1 -1] [26 98 153 203 255 299 335 -1 -1 -1]
[52 91 131 187 259 309 354 -1 -1 -1] [18 87 116 180 237 305 349 -1 -1 -1]
[45 90 156 183 258 297 355 -1 -1 -1] [46 67 148 184 252 281 346 -1 -1 -1]
[11 62 108 165 244 269 325 -1 -1 -1] [43 76 125 163 212 300 327 -1 -1 -1]
[23 101 155 207 234 294 341 -1 -1 -1] [5 77 133 202 241 273 356 -1 -1 -1]
[31 82 146 167 246 290 321 -1 -1 -1] [27 57 117 179 251 291 317 -1 -1 -1]
[4 72 129 200 214 263 353 -1 -1 -1] [19 92 144 171 230 292 350 -1 -1 -1]
[48 100 121 190 228 293 345 -1 -1 -1] [38 86 105 198 245 262 357 -1 -1 -1]
[12 53 147 173 221 289 315 -1 -1 -1] [35 68 139 191 238 284 338 -1 -1 -1]
[1 83 113 160 248 278 342 -1 -1 -1] [20 88 135 169 231 283 336 -1 -1 -1]
[37 63 143 196 235 261 330 -1 -1 -1] [28 78 126 185 223 275 328 -1 -1 -1]
[41 96 109 157 209 286 343 -1 -1 -1] [7 56 118 158 224 266 319 -1 -1 -1]
];

```

Şekil 4. 17. IBM ILOG OPL Optimization Studio Verileri-III

Sınıfların ve kapasitelerinin kısıtlı olması, kısa zamanda çok sınavın yapılması, gözetmen sayısının azlığı nedeniyle bazı sınavlar ortak yapılmak zorundadır. Bu durum ortak sınavlara ait veriler Şekil 4.18’de belirtilmiştir.

```

commonexams = { <10,180>,
                 <45,273>,
                 <2,363>,
                 <36,71>,
                 <213,357> };

```

Şekil 4. 18. IBM ILOG OPL Optimization Studio Verileri-IV

#### 4.4. Materyal

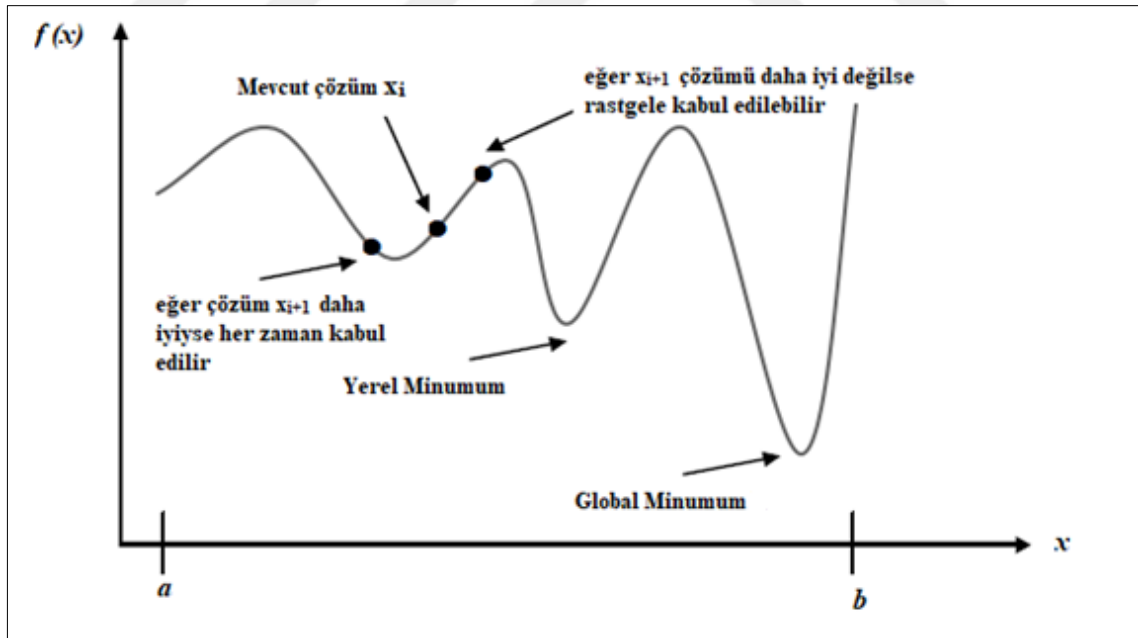
Çalışmanın bu bölümünde ise Tavlama Benzetim (Simulated Annealing) Algoritması kullanılmıştır. Her ne kadar bir önceki bölümde verilen matematiksel model bir çözüm verse de sınav çizelgeleme probleminin NP-Zor sınıfında olması ve ileride eklenebilecek ilave kısıtların problemin çözümünü zorlaştırabileceği ihtimali, bu problemi bir de meta-sezgisel çözüm yöntemlerinden Tavlama Benzetim algoritması ile çözmeye sevk etmiştir.

#### 4.4.1. Tavlama benzetimi (Simulated Annealing) Algoritması

Tavlama benzetimi bilgisayar bilimlerinde özellikle hesaplama alanında kullanılan ve 1983 yılında Kirkpatrick ve arkadaşları tarafından optimizasyon problemlerinde kullanılmak üzere önerilen meta sezgisel bir optimizasyon algoritmasıdır. Malzeme bilimindeki tavlama işleminden esinlenerek bir maddenin belli bir süre ısıtmanın ve daha sonrasında yavaş yavaş soğutmanın modellendiği fiziksel süreci ifade eder. Bu madde yüksek sıcaklıkta fiziksel işleme tabi tutulduğunda atomlar serbestçe hareket ederler ve rastgele yer değiştirmelerle hareket etme eğilimindedirler. Bu durum tavlama benzetimi algoritmasında sıcaklığın yüksek olduğunda geniş bir aralıkta çözüm elde etme imkânı verir. Madde soğutulmaya başlandığında atomlar serbestçe hareket edemez ve parçacıklar sıkılaşır. Atomların hareket etme olanağı sınırlı olur. Bu durumsa tavlama benzetimi algoritmasında sıcaklığın yavaşça soğutulduğunda daha sınırlı bir aralıkta çözüm olanağı sağlar.

#### 4.4.2. Tavlama benzetimi Algoritması Nasıl Çalışır?

Optimizasyon işlemlerinde probleme ilişkin deney sonuçlarında zamansal olarak farklı değerler elde edilir.



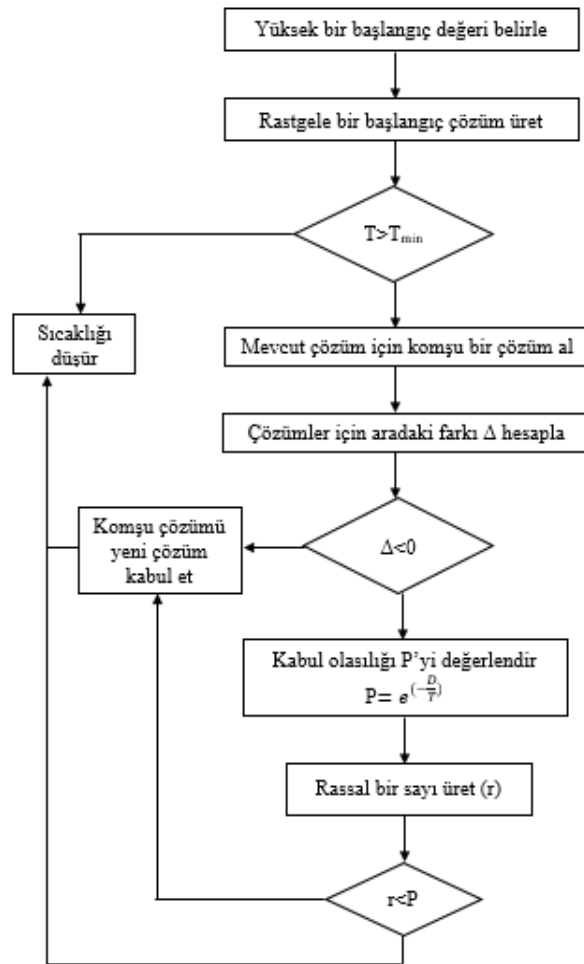
Şekil 4. 19. Tavlama benzetimi Algoritması Süreci

Şekil 19'da iki boyutlu bir fonksiyona ait grafik gösterilmiştir. Bu grafikteki global minimum problem için ulaşılmak istenilen bir hedeftir. Tavlama benzetimi algoritması global bir minimumu yani global optimizasyonu amaçlamaktadır. Grafikteki

yerel minimum bölgesel olarak ulaşılan minimum değerdir. Bu grafiğin en sağında kalan global minimum değerine ulaşmadan yani problemin daha iyi sonuçlarını göz ardı edilmiş olur. Bu şekilde elde edilen sonuçlara yerel minimum denilir. Bulunan ilk minimum değeri en küçük minimum değer kabul edilirse sistemdeki bu değerden daha düşük ve iyi sonuçlar veren minimum değerlere ulaşamaz.

#### 4.4.3. Tavlama benzetimi Algoritması Adımları

Tavlama benzetimi Algoritması ile fonksiyonun global minimum değerine ulaşacak algoritmanın adımları aşağıdaki gibidir. Aşağıdaki algoritma adımları komşu durumları da dikkate alarak en iyi sonucu bulmaya yöneliktir.



Şekil 4. 20. Tavlama benzetimi Algoritması Adımları

## 4.5. Yöntem

### 4.5.1. Sınav Çizelgeleme Problemi Tanımı

Sınav Çizelgeleme Problemi, sınavların akademik birimlerce belirlenen akademik sınav takvimde ilgili sınavlara ait kısıtlar gözetilerek oluşturulan sınav çizelgesinin ortaya çıkardığı sorunlar olarak tanımlayabiliriz. Sınıfların sayı ve kapasitelerinin yapılacak olan sınavlara ve sınava girecek öğrenci sayısını karşılayacak düzeyde olup olmaması, aynı öğrenci grubunun aynı zamanda farklı sınıflarda farklı sınavlara girecek olmanın getirdiği sınav çakışmalarının oluşturduğu istenmeyen durumlar bu problemi daha da zor hale getirmektedir. Problemi çözmek için mümkün olan sert kısıtlardan başlayarak öncelikle bu kısıtların sağlanması ve sonrasında yumuşak kısıtların göz önünde bulundurulması gerekir. Burada amaç mümkün olduğunca tüm kısıtların sağlanmasıdır.

### 4.5.2. Sınav Çizelgeleme Problemi ile İlgili Kısıtlar

Sınav çizelgeleme problemine ait kısıtları aşağıda maddeler halinde özetleyebiliriz.

- Sınavların yapılacağı gün sayısının belirtilmelidir.
- Sınavların yapılacağı zaman dilimleri belirtilmelidir.
- Günde yapılacak maksimum sınav sayısı belirtilmelidir.
- Sınav salonlarının kapasitelerinin sınava girecek öğrenci sayısından az olmaması durumlar belirtilmelidir.
- Sınav girecek gözetmen sayısının sınav sayısından az olmaması gereken durumlar belirtilmelidir.
- Aynı öğrenci grubunun aynı zamanda farklı sınıflarda farklı sınavlara girecek olmanın getirdiği sınav çakışmalarının oluşturduğu istenmeyen durumlar belirtilmelidir.
- Ortak yapılacak olan sınavların aynı zamanda aynı sınıfta farklı sınavlara girecek olmanın getirdiği durumlar belirtilmelidir.

### 4.5.3. Sınav Çizelgeleme Problemi için Geliştirilen Uygulama

Sınav çizelgeleme problemi için Microsoft Visual Studio 2015 yazılımı ve C# programlama dili kullanılarak sınav çizelgemeye ait uygulama geliştirilmiştir. Uygulama geliştirilirken ilgili veriler dikkate alınarak oluşturulan sınav çizelgeleme problemine ilişkin tüm parametreler, değişken ve kısıtlar göz önünde bulundurulmuştur. Daha sonrasında sezgisel algoritmalarından olan tavlama benzetimi algoritması ile sınav çizelgeleme problemi çözüme ulaştırılması amaçlanmıştır.

#### 4.5.4. Sınav Çizelgeme Problemi için Tavlama Benzetimi Algoritması

Sınav programı çizelgelerken sınav çizelgeleme problemine optimuma yakın bir sonuç bulmak için geliştirilen tavlama benzetimi algoritması amaç fonksiyonu açısından iyi bir sonuç değerine ulaşmak amacıyla kullandığımız sezgisel bir algoritmadır.

Tavlama benzetimi algoritması optimizasyon problemleri için kullanılan sezgisel algoritmalarından biridir. Bu çalışmadaki sınav çizelgeleme problemi için C# programlama dilinde geliştirilen tavlama benzetimi algoritmasına ait Tablo 4.1 de sözde kod belirtilmiştir.

Tavlama benzetimi algoritmasında öncesinde yüksek bir sıcaklık değeri, minimum sıcaklık değeri, soğutma oranı ve iterasyon sayısı tanımlanır ve sonrasında başlangıç çözüm bulunur.

En iyi çözüm mevcut çözüm olarak kabul edilir. Bu mevcut çözüme yakın komşu çözümler bulunarak çözümler karşılaştırılır. Daha öncesinde belirtilmiş olan başlangıç sıcaklık değeri belirlenen soğutma oranı ile düşürülür.

Tüm iterasyonlarda problem için uygun çözümler aranır. Sonrasında öncelikle çakışan sınavlardan başlanarak çizelge dizininde rastgele iki sınavın yeri değiştirilerek amaç fonksiyonu değeri yeniden hesaplanır.

**Tablo 4. 1.** Tavlama Benzetimi Algoritması için sözde kod

---

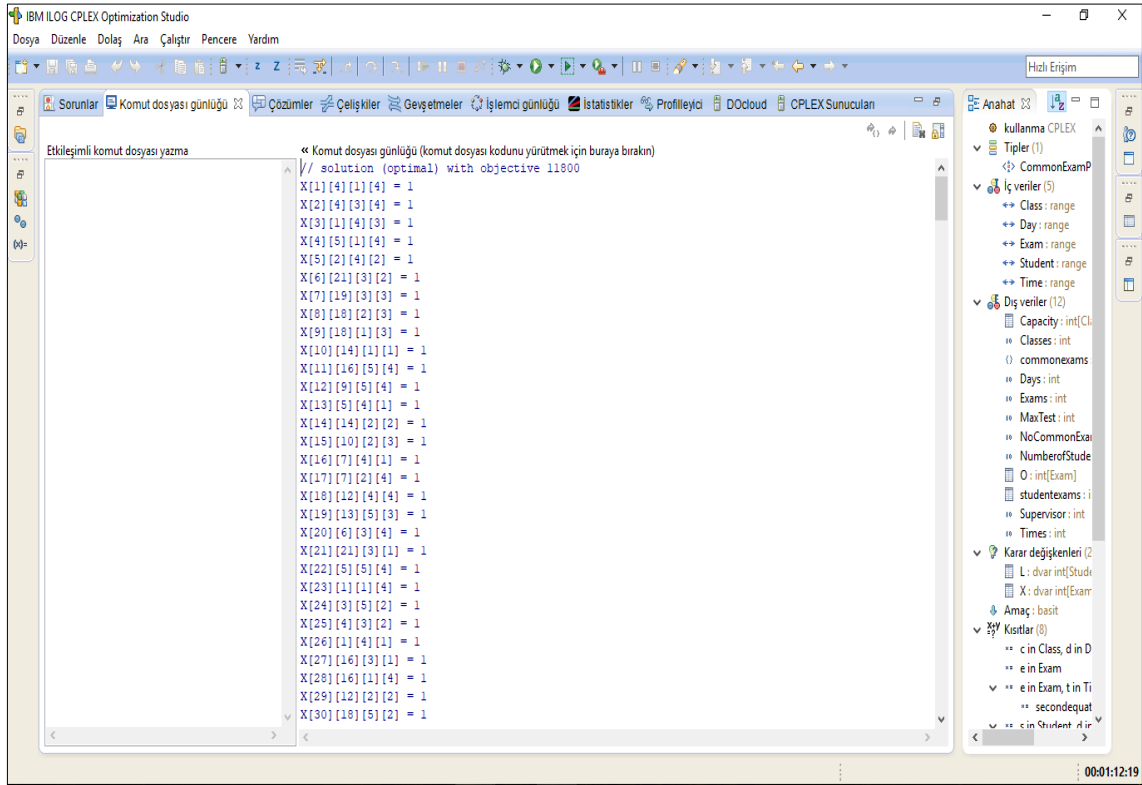
```
1 Initialize (Define a high temperature T, a low temperature minT, a cooling rate coolRate, a iteration number and random starting point)
2 Select objective_valueinitial ∈ Objective Function
3 objective_valueaccepted = objective_valueinitial
4 objectivef_valuebest = objective_valueinitial
5 While (T > minT)
6 iteration = 0
7 While (iteration < numIterations)
8 Choose random number and Calculate objective_value
9 iteration++
10 if (objective_value ≤ current_objective_value)
11 objective_valuetemperature = objective_valuecurrent
12 if (objective_value ≤ best_objective_value)
13 objective_valuebest = objective_valuetemperature
14 Else if ((Math.Exp((current_objective_value - objective_value)/T)) >= random.NextDouble())
15 objective_valuecurrent = objective_valuetemperature
16 T = T * coolRate
17 End While
18 Output objective_valuebest
```

---

## 5.ARAŞTIRMA BULGULARI ve TARTIŞMA

### 5.1. Sınav Çizelgeleme Problemi için Geliştirilen Uygulamalar

Sınav Çizelgeleme Problemi için ilk olarak IBM ILOG OPL Optimization Studio yazılımı kullanılarak geliştirilen matematiksel model ve sınav çizelgeleme problemine ait verilere ilişkin deney sonuçları elde edilmeye çalışıldı. Bu deney sonuçları yazılımdan alınan aşağıdaki şekilde belirtilmiştir.



Şekil 5. 1. IBM ILOG OPL Optimization Studio Deney Sonuçları

Matematiksel programlama modeli yaklaşık 3 dakika boyunca çalıştırıldı. Bu yazılımda amaç fonksiyon değeri 11500 bulunmuştur. Bu 115 tane sınavın aynı günde başka sınavları olan öğrenci gruplarına verildiğini göstermiştir. Hatta bazı durumlarda bazı öğrenci gruplarının gün içerisinde üç tane sınava girmesi gerekebilmektedir. Bu durum sınav çizelgeleme probleminde istediğimiz bir hedef olmadığından geliştirdiğimiz tavlama benzetim algoritmasında bir öğrenci grubunun gün içerisinde gireceği ekstra sınavların ceza katsayıları artan bir şekilde düzenlenmiştir. Diğer bir deyişle bir öğrenci grubunun gün içerisinde gireceği 3. Sınavın ceza katsayısı 2. Sınavın ceza katsayısına göre daha yüksektir.

## 5.2. Sınav Çizelgeleme Problemine ait Deneysel Sonular

Tavlama benzetimi (Simulated Annealing) algoritması ile geliřtirdiđimiz modelin Necmettin Erbakan niversitesi Mhendislik ve Mimarlık Fakltesi 2017-2018 gz dnemi sınav verileri dikkate alınarak oluřturulan sınav izelgesine etkileri arařtırılmıřtır. Geliřtirdiđimiz tavlama benzetimi algoritması modelinde bařlangı sıcaklık deđeri, minimum sıcaklık deđeri, sođutma oranı ve iterasyon sayısı gibi parametreler kullanılmıřtır.

Tavlama benzetim kodundaki ceza katsayıları bir đrenci grubunun bir gnde birden fazla sınavın girmesi durumunun cezalandırıldıđı katsayılar olarak ifade edilmiřtir. rneđin bir đrenci grubunun bir gnde iki sınava girmesi durumu iin tavlama benzetimi kodunda belirlediđimiz ceza katsayısı 1'dir. Bir đrenci grubunun bir gnde  sınava girmesi durumu iin tavlama benzetimi kodunda belirlediđimiz ceza katsayısı 500'dir. Bir đrenci grubunun bir gnde drt sınava girmesi durumu iin tavlama benzetimi kodunda belirlediđimiz ceza katsayısı 1000'dir. Bir đrenci grubunun aynı gnde ve aynı zaman diliminde sınava giremeyeceđi yani sınav akıřmalarının olduđu durum iin tavlama benzetimi kodunda belirlediđimiz ceza katsayısı 100000'dir.

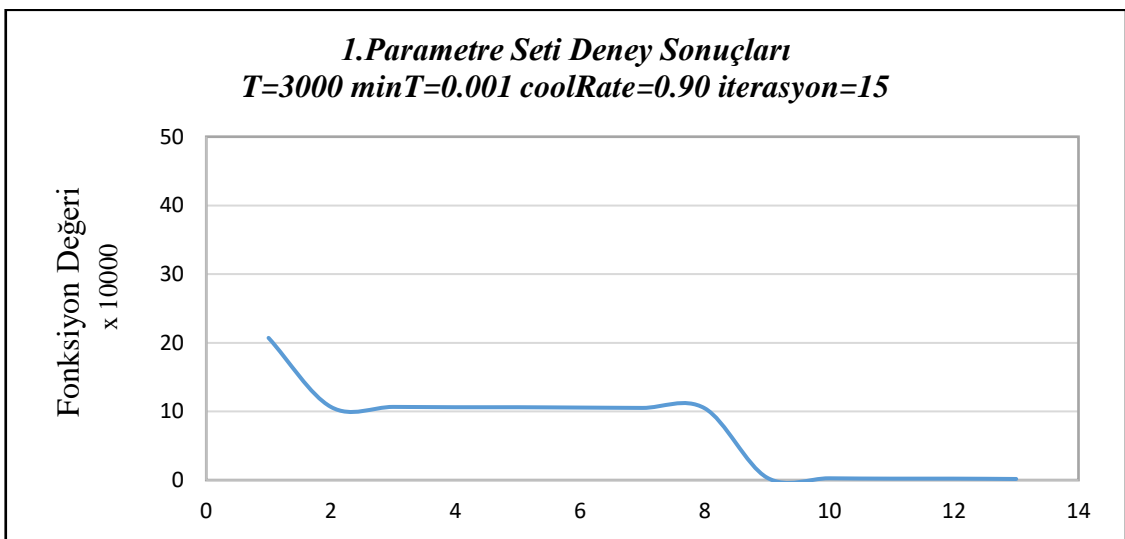
Tavlama benzetim kodundaki bir đrenci grubunun bir gnde sırasıyla iki,  ve drt sınava girmesi durumu ile bir đrenci grubunun aynı gnde ve aynı zaman diliminde sınava giremeyeceđi yani sınav akıřmalarının olduđu durumun oluřturduđu drt farklı durum iin tavlama benzetimi kodunda belirlediđimiz ceza katsayıları toplanarak ama fonksiyonu deđeri hesaplanmıřtır.

1.Parametre Seti:  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=15$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonunun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.1’de gösterilmiştir.

**Tablo 5. 1.** 1. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=15</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
207097	706603	806603	604606	605103	806603	207097
1622	204115	202116	301622	301119	301119	<b><u>1622</u></b>

Tablo 5.1’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 1. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonunun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.2’de gösterilmiştir. Bu grafikte 1. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonunun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 1. parametre seti ile elde edilen fonksiyonun minimum değeri 1622’dir.



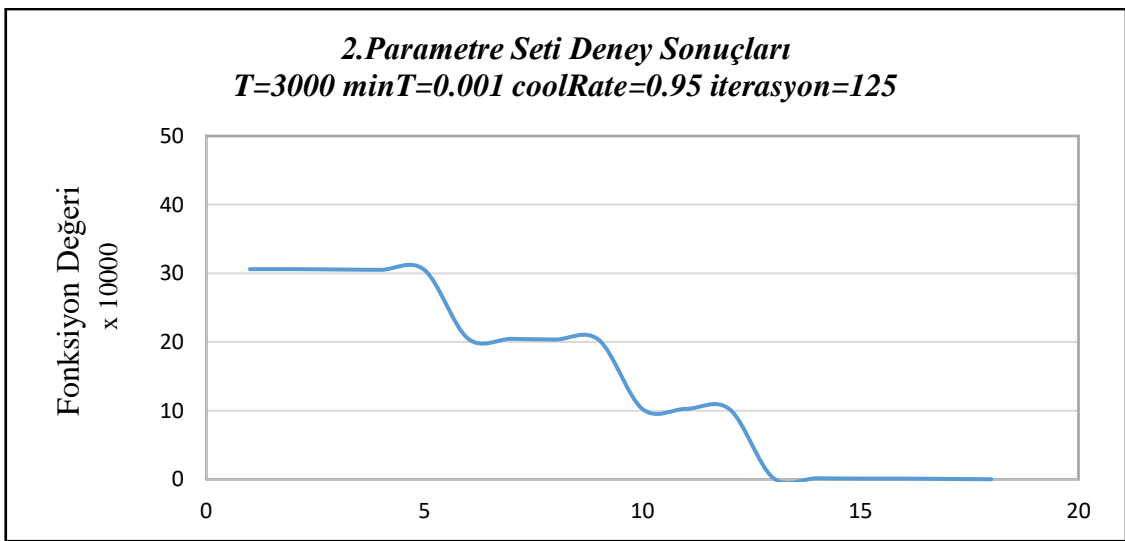
**Şekil 5. 2.** 1. Parametre Seti Deney Sonuçları.

**2.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.95$ ,  $\text{iterasyon}=125$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.2’de gösterilmiştir.

**Tablo 5. 2. 2. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,95 iterasyon=125</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
306101	505105	306598	404604	606597	606597	306101
119	120	615	130	119	615	<b><u>119</u></b>

Tablo 5.2’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 2. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.3’te gösterilmiştir. Bu grafikte 2. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 2. parametre seti ile elde edilen fonksiyonun minimum değeri 119’dur.



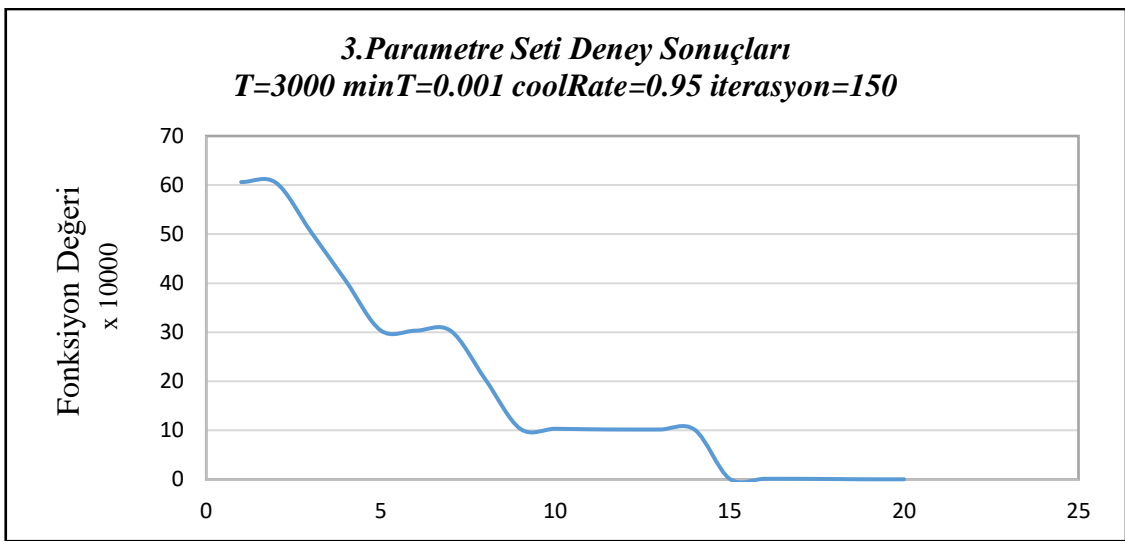
**Şekil 5. 3. 2. Parametre Seti Deney Sonuçları**

**3.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.95$ ,  $\text{iterasyon}=150$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.3'te gösterilmiştir.

**Tablo 5. 3. 3. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,95 iterasyon=150</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
606104	504105	405607	206597	306104	606104	206597
116	620	124	120	128	620	<b><u>119</u></b>

Tablo 5.3'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 3. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.4'te gösterilmiştir. Bu grafikte 3. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 3. parametre seti ile elde edilen fonksiyonun minimum değeri 119'dur.



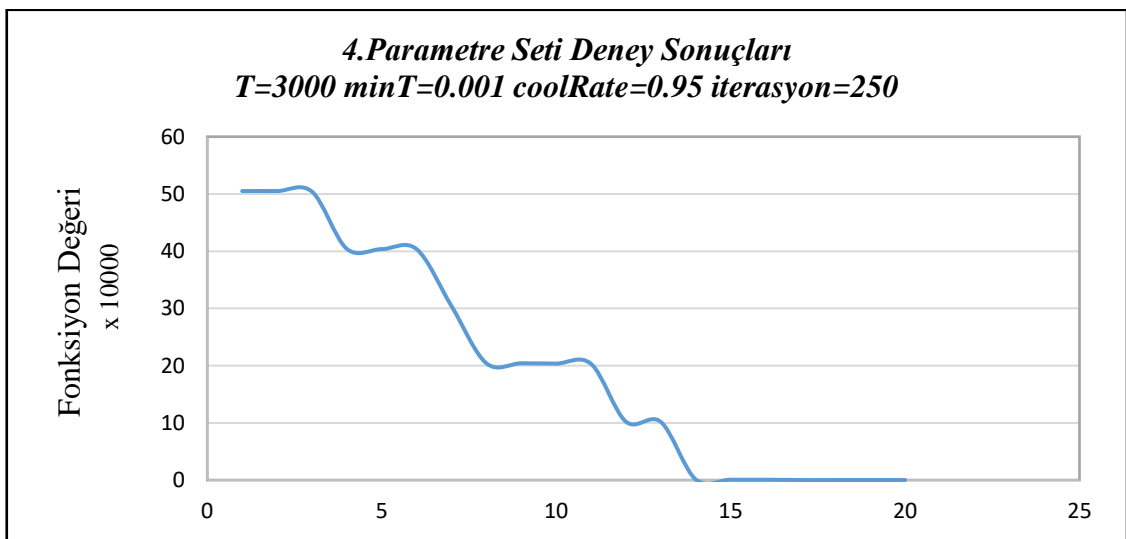
**Şekil 5. 4. 3. Parametre Seti Deney Sonuçları**

4.Parametre Seti:  $T=3000$ ,  $minT=0.001$ ,  $coolRate=0.95$ ,  $iterasyon=250$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.4'te gösterilmiştir.

**Tablo 5. 4.** 4. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,95 iterasyon=250</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
306598	407596	308095	505108	707095	707095	306598
119	136	119	115	118	136	<b><u>115</u></b>

Tablo 5.4'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 4. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.5'te gösterilmiştir. Bu grafikte 4. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 4. parametre seti ile elde edilen fonksiyonun minimum değeri 115'tir.



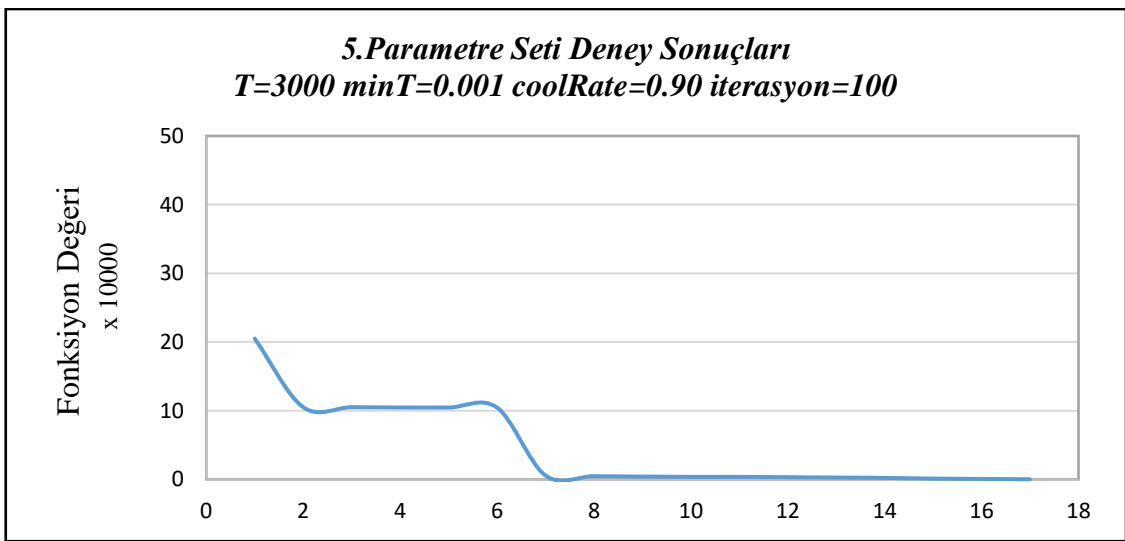
**Şekil 5. 5.** 4. Parametre Seti Deney Sonuçları

**5.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=100$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonunun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.5'te gösterilmiştir.

**Tablo 5. 5. 5. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=100</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
306602	706104	205106	205108	705604	706104	205106
628	624	129	620	139	628	<b><u>129</u></b>

Tablo 5.5'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 5. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonunun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.6'da gösterilmiştir. Bu grafikte 5. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonunun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 5. parametre seti ile elde edilen fonksiyonunun minimum değeri 129'dur.



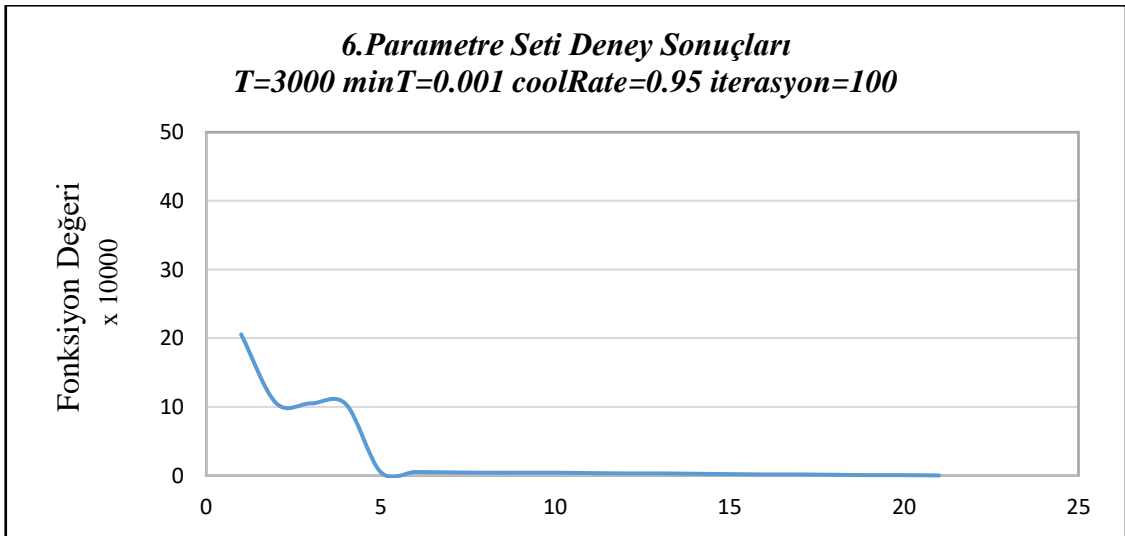
**Şekil 5. 6. 5. Parametre Seti Deney Sonuçları**

**6.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.95$ ,  $\text{iterasyon}=100$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.6'da gösterilmiştir.

**Tablo 5. 6. 6. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,95 iterasyon=100</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
506107	205605	507096	707098	405602	707098	205605
1114	122	631	134	132	1114	<b><u>122</u></b>

Tablo 5.6'da gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 6. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.7'de gösterilmiştir. Bu grafikte 6. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 6. parametre seti ile elde edilen fonksiyonun minimum değeri 122'dir.



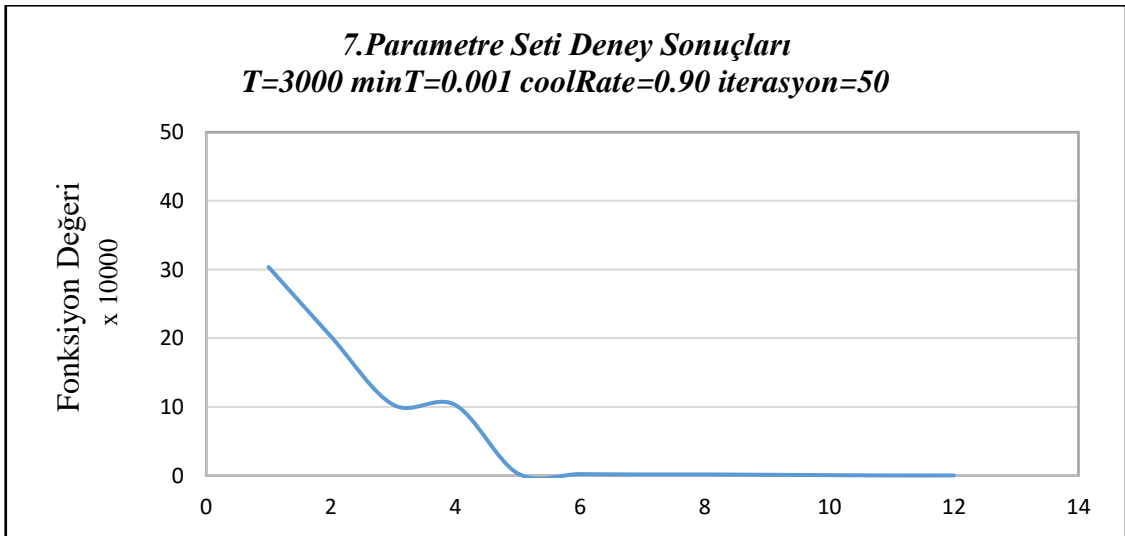
**Şekil 5. 7. 6. Parametre Seti Deney Sonuçları**

7.Parametre Seti:  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=50$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonunun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.7’de gösterilmiştir.

**Tablo 5. 7. 7. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=50</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
307100	406105	303609	106096	305108	406105	106096
2623	1117	124	619	626	2623	<b><u>124</u></b>

Tablo 5.7’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 7. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonunun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.8’de gösterilmiştir. Bu grafikte 7. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonunun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 7. parametre seti ile elde edilen fonksiyonun minimum değeri 124’tür.



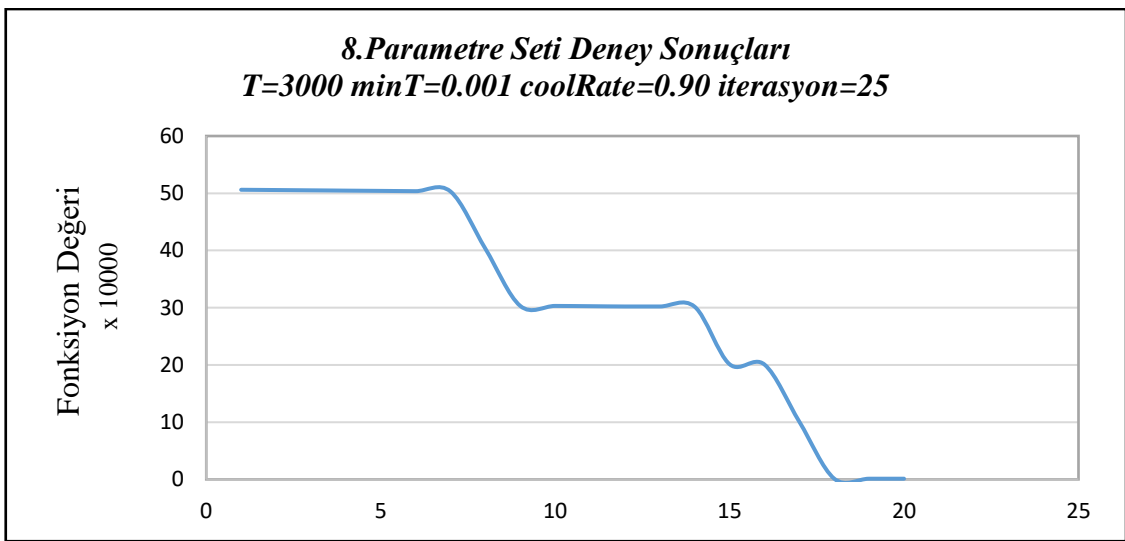
**Şekil 5. 8. 7. Parametre Seti Deney Sonuçları**

**8.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=25$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.8'de gösterilmiştir.

**Tablo 5. 8. 8. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=25</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
306104	305108	506098	606102	505599	606102	305108
201621	2119	1129	2614	103115	201621	<b><u>1129</u></b>

Tablo 5.8'de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 8. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.9'da gösterilmiştir. Bu grafikte 8. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 8. parametre seti ile elde edilen fonksiyonun minimum değeri 1129'dur.



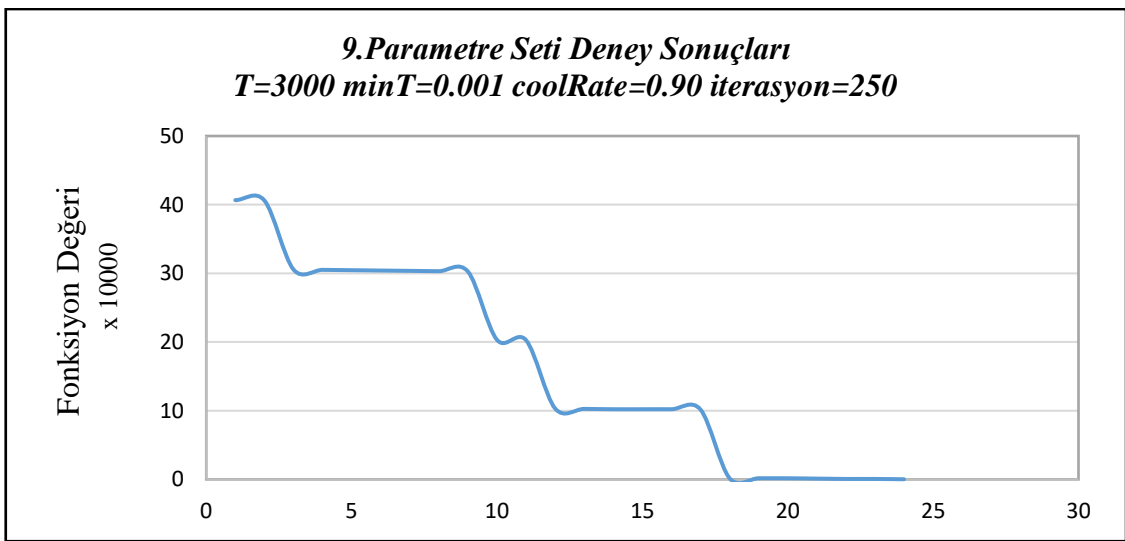
**Şekil 5. 9. 8. Parametre Seti Deney Sonuçları**

9.Parametre Seti:  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=250$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.9'da gösterilmiştir.

**Tablo 5. 9. 9. Parametre Seti Deney Sonuçları**

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=250</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
505106	406102	405607	406598	507595	507595	405607
124	100117	1116	122	129	100117	<b><u>122</u></b>

Tablo 5.9'da gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 9. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.10'da gösterilmiştir. Bu grafikte 9. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 9. parametre seti ile elde edilen fonksiyonun minimum değeri 122'dir.



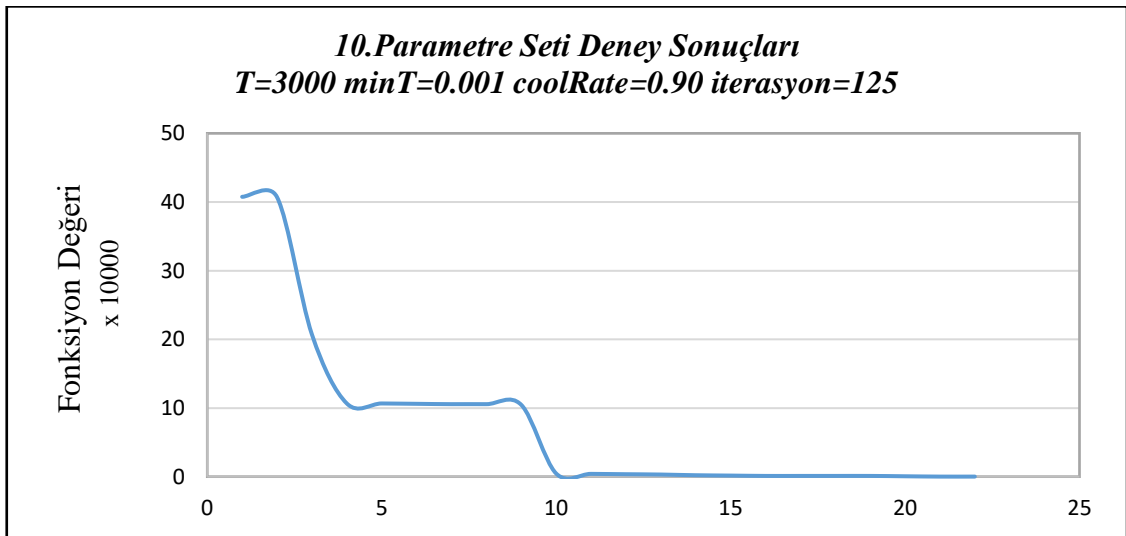
**Şekil 5. 10. 9. Parametre Seti Deney Sonuçları**

10.Parametre Seti:  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=125$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.10'da gösterilmiştir.

**Tablo 5. 10.** 10. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=125</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
107596	706101	407600	306100	5606	706101	5606
1122	124	120	1124	625	1124	<b><u>120</u></b>

Tablo 5.10'da gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 10. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.11'de gösterilmiştir. Bu grafikte 10. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 10. parametre seti ile elde edilen fonksiyonun minimum değeri 120'dir.



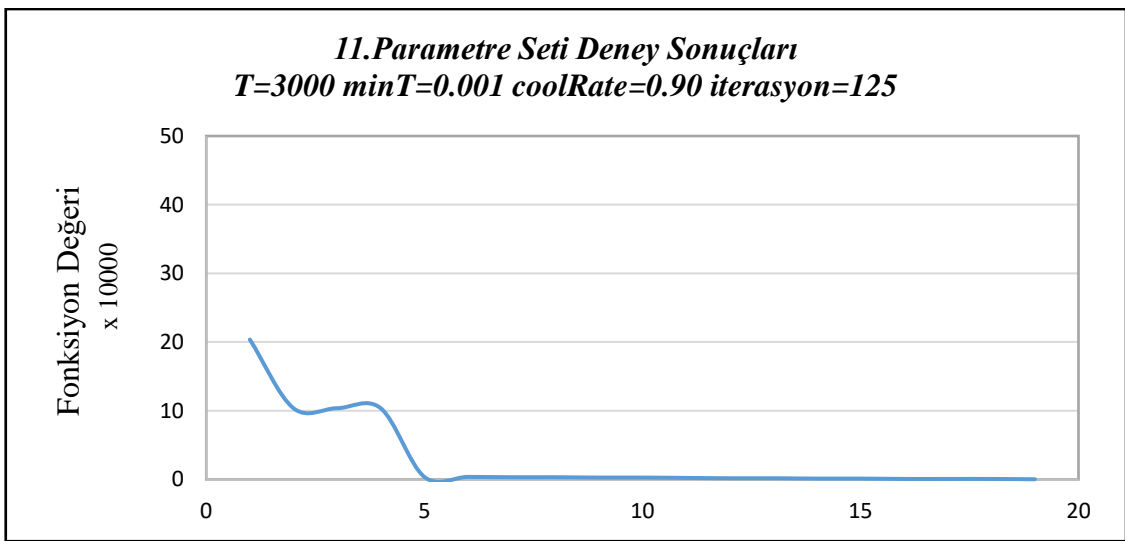
**Şekil 5. 11.** 10. Parametre Seti Deney Sonuçları

**11.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=200$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.11’de gösterilmiştir.

**Tablo 5. 11.** 11. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=200</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
203612	306096	404610	407595	307099	407595	203612
119	126	132	130	132	132	<b><u>119</u></b>

Tablo 5.11’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 11. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.12’de gösterilmiştir. Bu grafikte 11. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 11. parametre seti ile elde edilen fonksiyonun minimum değeri 119’dur.



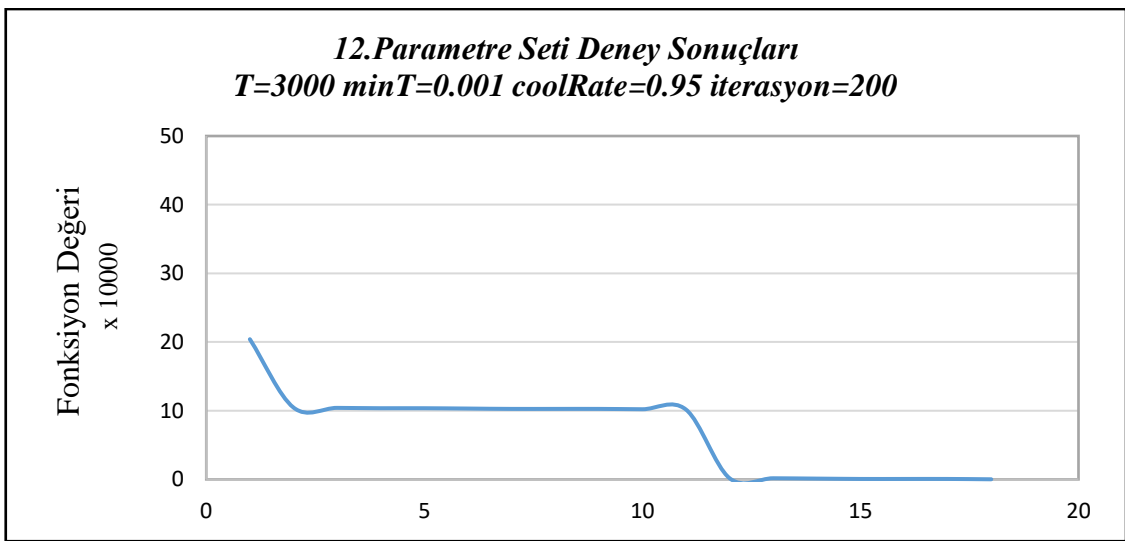
**Şekil 5. 12.** 11. Parametre Seti Deney Sonuçları

**12.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.95$ ,  $\text{iterasyon}=200$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.12’de gösterilmiştir.

**Tablo 5. 12.** 12. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,95 iterasyon=200</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
307101	306601	605105	507098	204106	605105	204106
117	118	117	116	115	118	<b><u>115</u></b>

Tablo 5.12’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 12. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.13’de gösterilmiştir. Bu grafikte 12. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 12. parametre seti ile elde edilen fonksiyonun minimum değeri 115’tir.



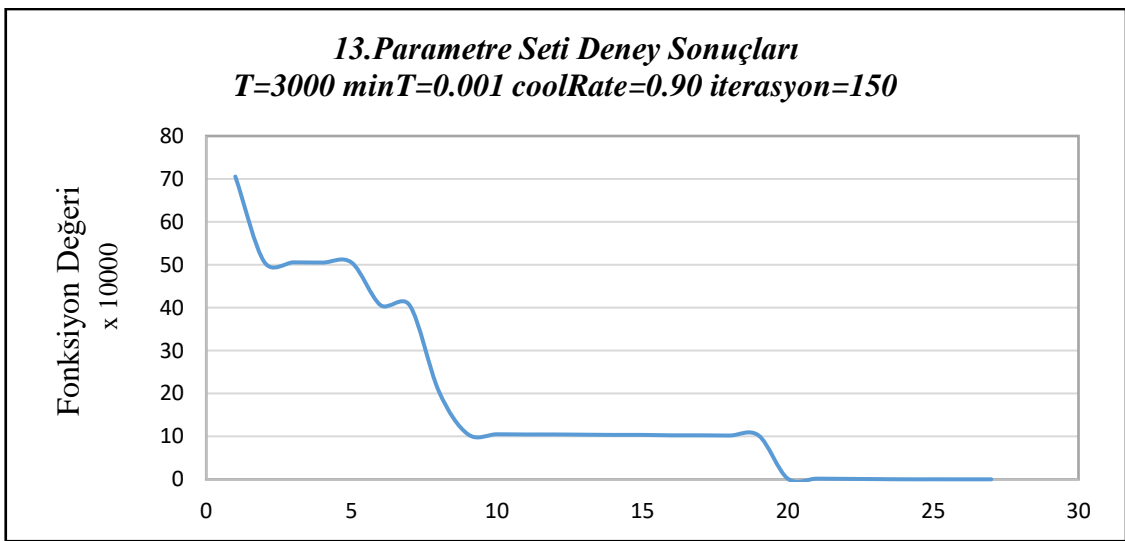
**Şekil 5. 13.** 12. Parametre Seti Deney Sonuçları

**13.Parametre Seti:**  $T=3000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=150$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.13'te gösterilmiştir.

**Tablo 5. 13.** 13. Parametre Seti Deney Sonuçları

<b>T=3000 minT=0,001 coolRate=0,90 iterasyon=150</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
708096	705604	206601	205603	106104	708096	106104
616	121	134	138	129	616	<b><u>121</u></b>

Tablo 5.13'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 13. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.14'te gösterilmiştir. Bu grafikte 13. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 13. parametre seti ile elde edilen fonksiyonun minimum değeri 121'dir.



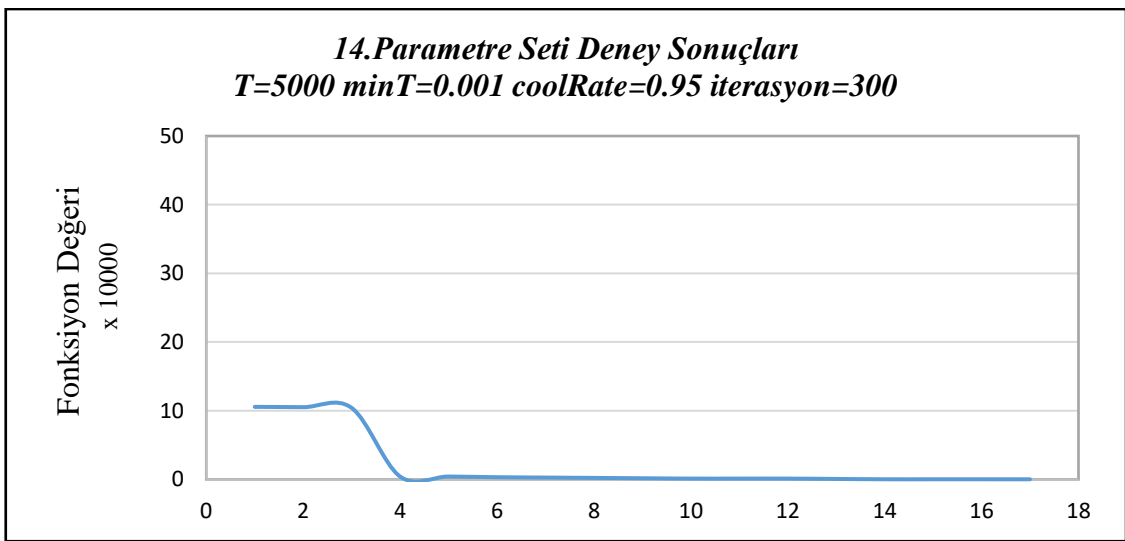
**Şekil 5. 14.** 13. Parametre Seti Deney Sonuçları

14.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=300$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.14'te gösterilmiştir.

**Tablo 5. 14.** 14. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,95 iterasyon=300</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
105601	807098	305604	407595	306697	807098	105601
115	115	116	116	115	116	<b><u>115</u></b>

Tablo 5.14'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 14. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.15'te gösterilmiştir. Bu grafikte 14. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 14. parametre seti ile elde edilen fonksiyonun minimum değeri 115'tir.



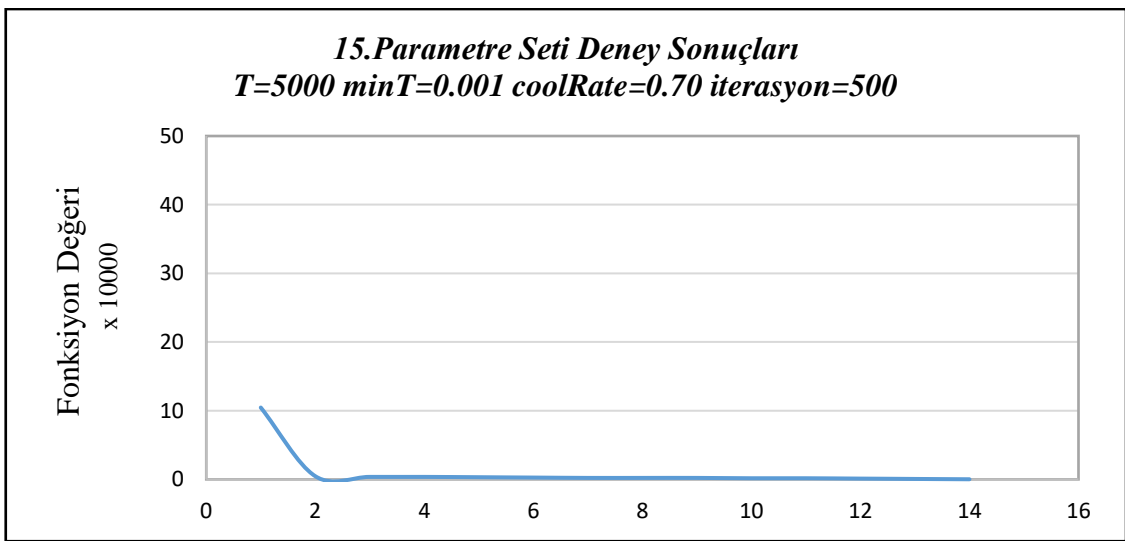
**Şekil 5. 15.** 14. Parametre Seti Deney Sonuçları

**15.Parametre Seti:**  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.70$ ,  $\text{iterasyon}=500$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.15'te gösterilmiştir.

**Tablo 5. 15.** 15. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,70 iterasyon=500</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
104606	104605	307593	405601	506603	506603	104605
119	131	124	629	119	629	<b><u>119</u></b>

Tablo 5.15'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 15. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.16'da gösterilmiştir. Bu grafikte 15. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 15. parametre seti ile elde edilen fonksiyonun minimum değeri 119'dur.



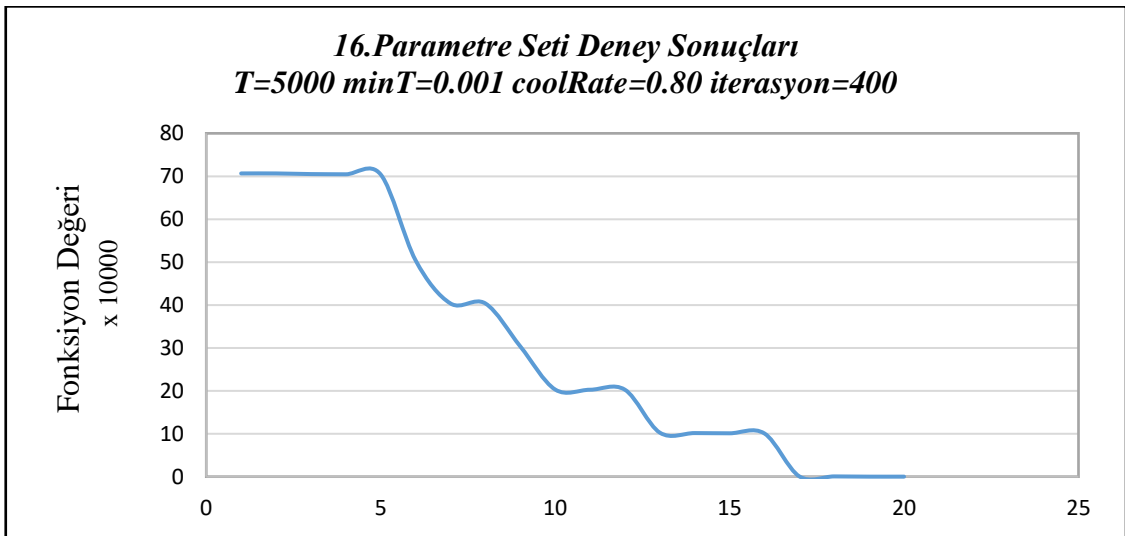
**Şekil 5. 16.** 15. Parametre Seti Deney Sonuçları

**16.Parametre Seti:**  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.80$ ,  $\text{iterasyon}=400$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.16'da gösterilmiştir.

**Tablo 5. 16.** 16. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,80 iterasyon=400</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
304604	404110	706600	905605	106597	905605	106597
121	118	120	131	620	620	<b><u>120</u></b>

Tablo 5.16'da gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 16. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.17'de gösterilmiştir. Bu grafikte 16. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 16. parametre seti ile elde edilen fonksiyonun minimum değeri 120'dir.



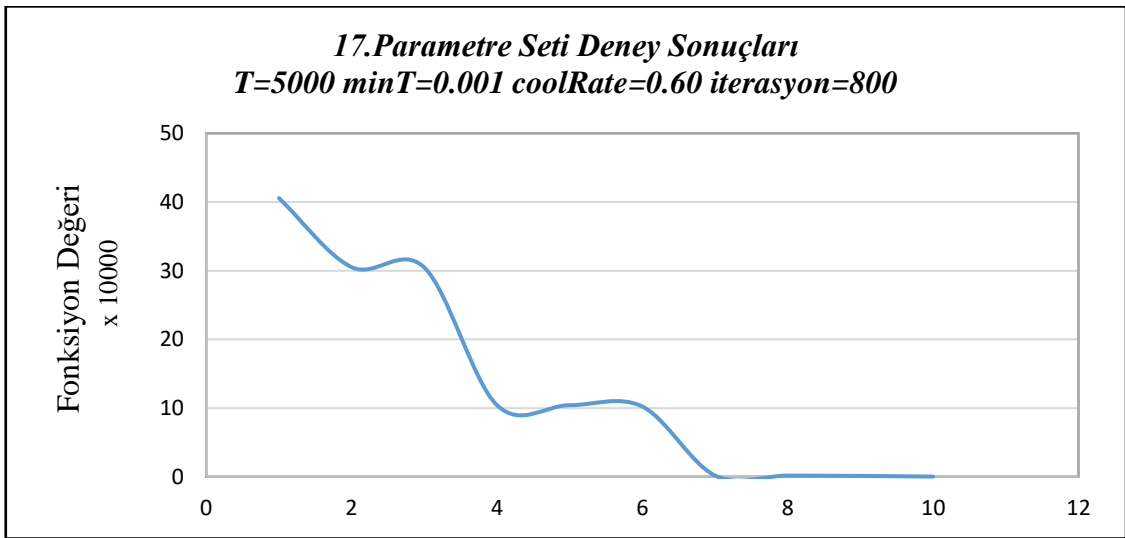
**Şekil 5. 17.** 16. Parametre Seti Deney Sonuçları

17.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.60$ ,  $\text{iterasyon}=800$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.17’de gösterilmiştir.

**Tablo 5. 17.** 17. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,60 iterasyon=800</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
405606	204610	206106	504112	105604	504112	105604
118	133	129	619	615	619	<b>118</b>

Tablo 5.17’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 17. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.18’de gösterilmiştir. Bu grafikte 17. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 17. parametre seti ile elde edilen fonksiyonun minimum değeri 118’dir.



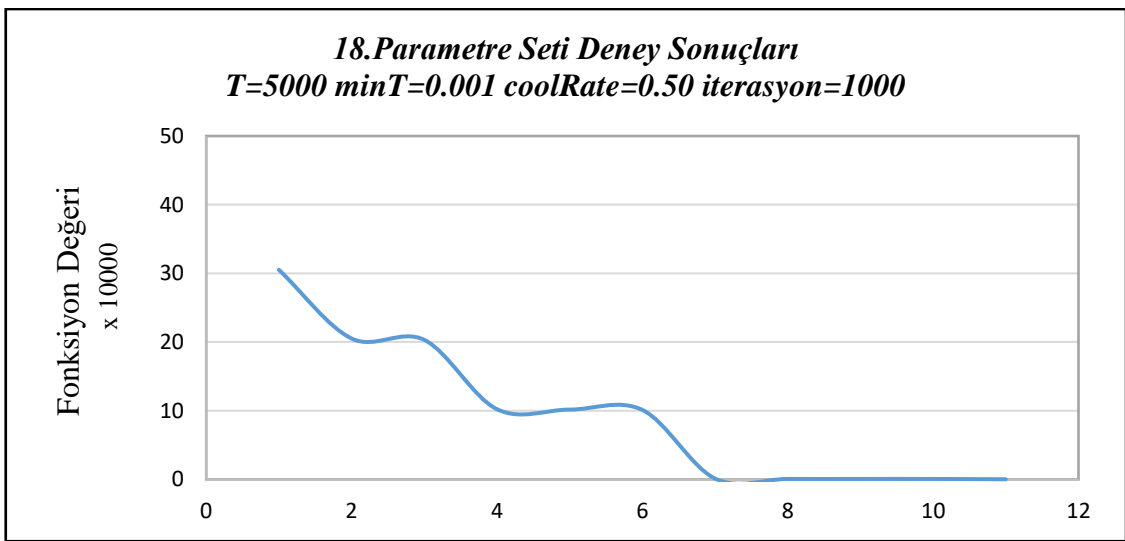
**Şekil 5. 18.** 17. Parametre Seti Deney Sonuçları

**18.Parametre Seti:**  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.50$ ,  $\text{iterasyon}=1000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.18’de gösterilmiştir.

**Tablo 5. 18.** 18. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,50 iterasyon=1000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
105103	305105	205606	407096	507101	507101	105103
128	118	626	118	125	626	<b><u>118</u></b>

Tablo 5.18’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 18. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.19’da gösterilmiştir. Bu grafikte 18. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 18. parametre seti ile elde edilen fonksiyonun minimum değeri 118’dir.



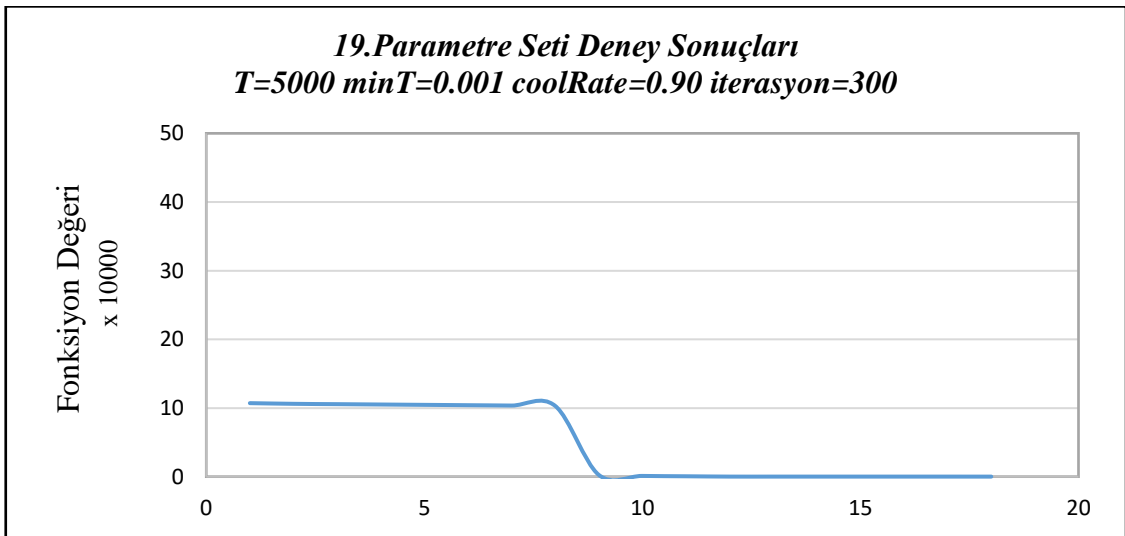
**Şekil 5. 19.** 18. Parametre Seti Deney Sonuçları

19.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.90$ ,  $\text{iterasyon}=300$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.19'da gösterilmiştir.

**Tablo 5. 19.** 19. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,90 iterasyon=300</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
708597	205605	404107	107100	6598	708597	6598
116	122	115	115	117	122	<b>115</b>

Tablo 5.19'da gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 19. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.20'de gösterilmiştir. Bu grafikte 19. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 19. parametre seti ile elde edilen fonksiyonun minimum değeri 115'tir.



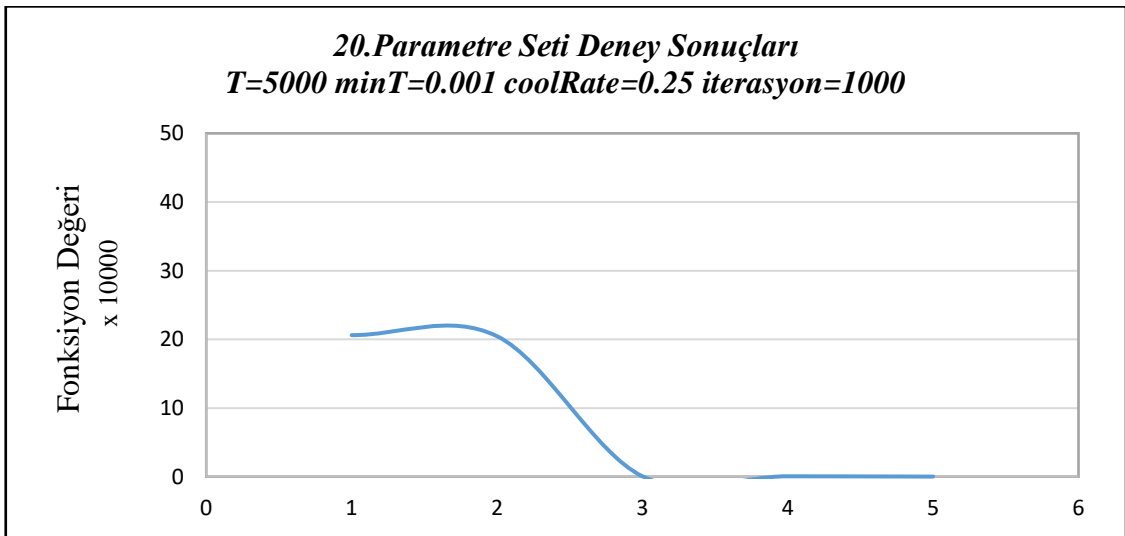
**Şekil 5. 20.** 19. Parametre Seti Deney Sonuçları

20.Parametre Seti:  $T=5000$ ,  $minT=0.001$ ,  $coolRate=0.25$ ,  $iterasyon=1000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.20’de gösterilmiştir.

**Tablo 5. 20.** 20. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,25 iterasyon=1000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
504603	206102	607598	305600	205104	607598	205104
120	119	1126	1617	122	1617	<b><u>119</u></b>

Tablo 5.20’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 20. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.21’de gösterilmiştir. Bu grafikte 20. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 20. parametre seti ile elde edilen fonksiyonun minimum değeri 119’dur.



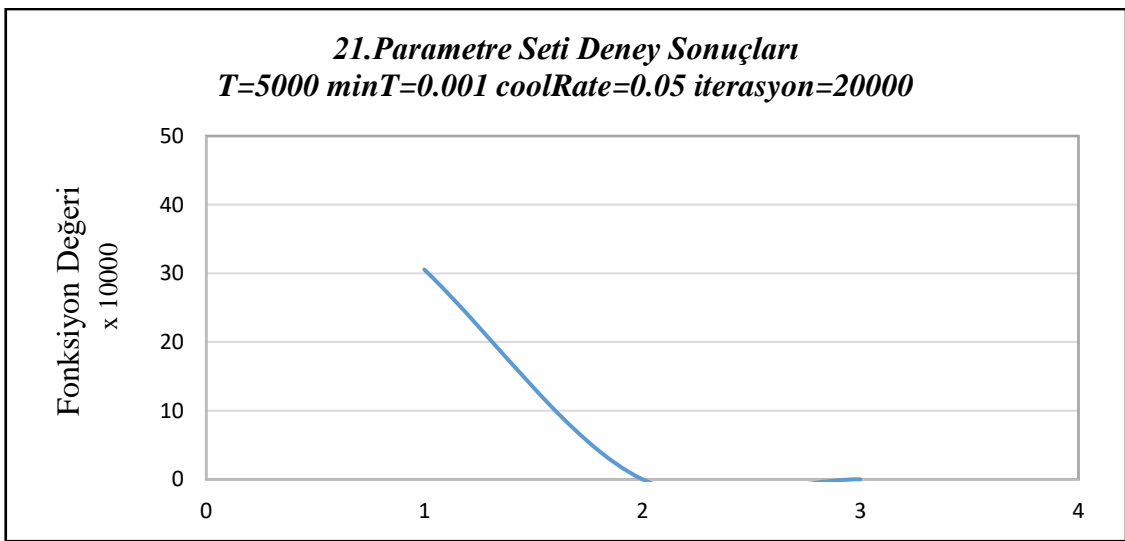
**Şekil 5. 21.** 20. Parametre Seti Deney Sonuçları

21.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.05$ ,  $\text{iterasyon}=20000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.21’de gösterilmiştir.

**Tablo 5. 21.** 21. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,05 iterasyon=20000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
305606	507093	608095	307595	305601	608095	305606
115	115	115	115	115	<b><u>115</u></b>	<b><u>115</u></b>

Tablo 5.21’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 21. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.22’de gösterilmiştir. Bu grafikte 21. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 21. parametre seti ile elde edilen fonksiyonun minimum değeri 115’tir.



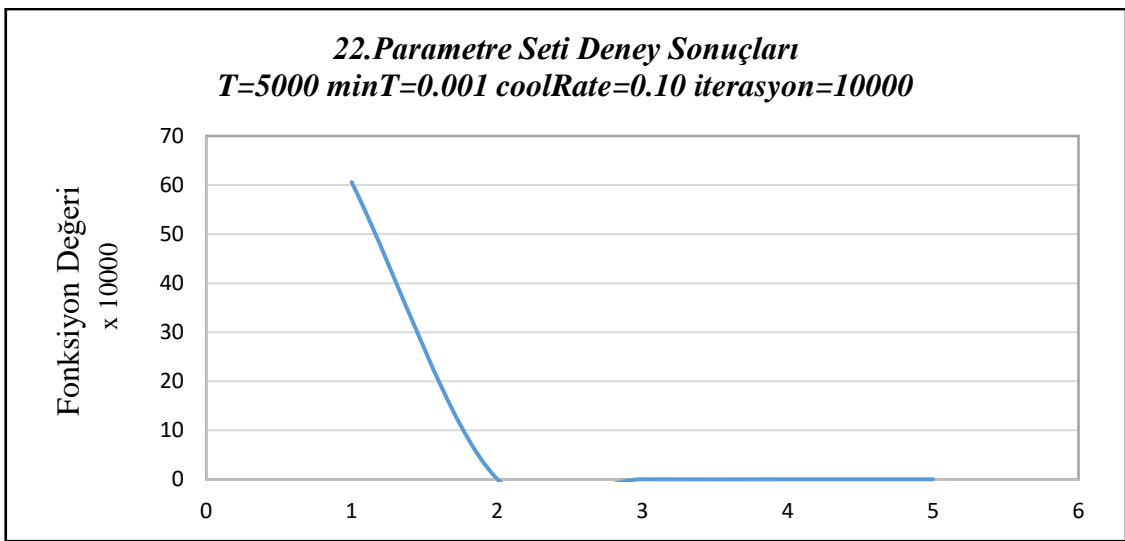
**Şekil 5. 22.** 21. Parametre Seti Deney Sonuçları

22.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.10$ ,  $\text{iterasyon}=10000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.22’de gösterilmiştir.

**Tablo 5. 22.** 22. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,10 iterasyon=10000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
705605	705600	705603	606101	205102	705605	205102
117	115	115	115	116	117	<u>115</u>

Tablo 5.22’de gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 22. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.23’te gösterilmiştir. Bu grafikte 22. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 22. parametre seti ile elde edilen fonksiyonun minimum değeri 115’tir.



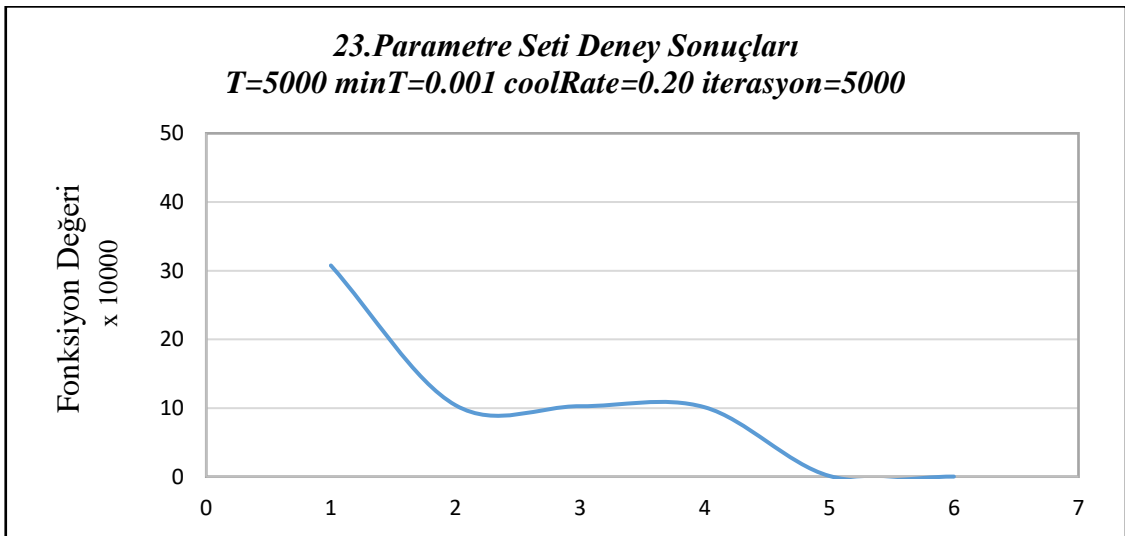
**Şekil 5. 23.** 22. Parametre Seti Deney Sonuçları

23.Parametre Seti:  $T=5000$ ,  $minT=0.001$ ,  $coolRate=0.20$ ,  $iterasyon=5000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.23'te gösterilmiştir.

**Tablo 5. 23.** 23. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,20 iterasyon=5000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
106094	307594	912082	706103	405602	912082	106094
127	116	119	116	133	133	<b><u>116</u></b>

Tablo 5.23'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 23. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.24'te gösterilmiştir. Bu grafikte 23. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 23. parametre seti ile elde edilen fonksiyonun minimum değeri 116'dır.



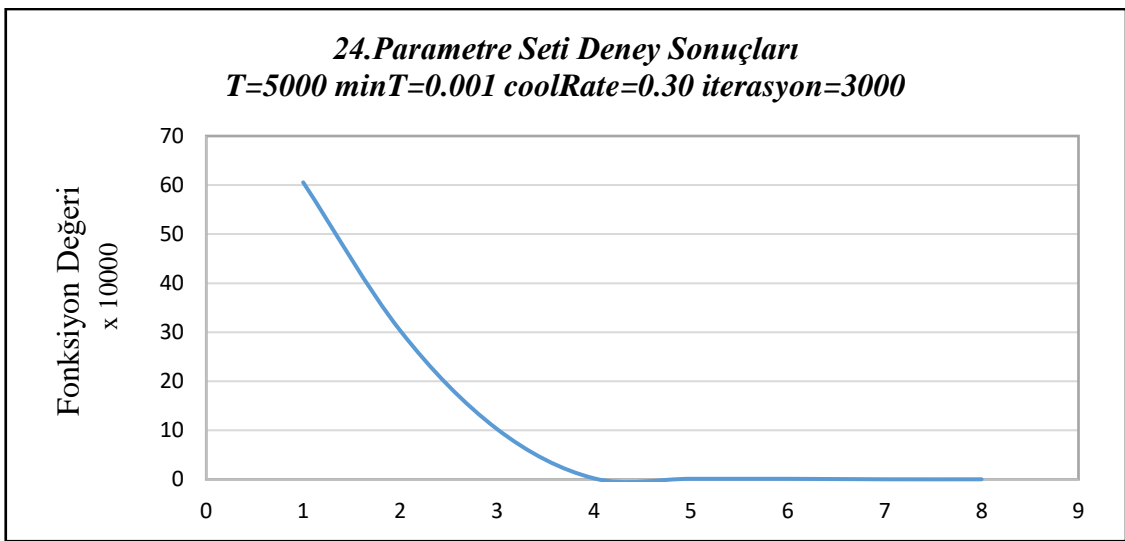
**Şekil 5. 24.** 23. Parametre Seti Deney Sonuçları

24.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.30$ ,  $\text{iterasyon}=3000$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemedeki minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.24'te gösterilmiştir.

**Tablo 5. 24.** 24. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,30 iterasyon=3000</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
605601	407103	203610	305108	605601	605601	203610
116	118	614	131	117	614	<b><u>116</u></b>

Tablo 5.24'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 24. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.25'te gösterilmiştir. Bu grafikte 24. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 24. parametre seti ile elde edilen fonksiyonun minimum değeri 116'dır.



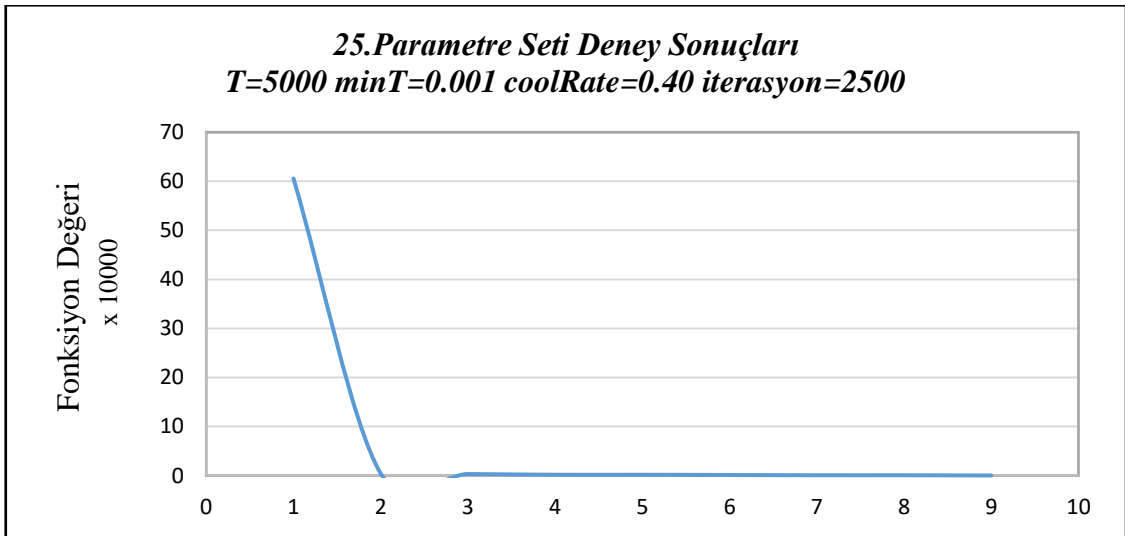
**Şekil 5. 25.** 24. Parametre Seti Deney Sonuçları

25.Parametre Seti:  $T=5000$ ,  $\min T=0.001$ ,  $\text{coolRate}=0.40$ ,  $\text{iterasyon}=2500$  olarak belirlenmiştir. Ceza katsayısı ile tanımlanan amaç fonksiyonun tavlama benzetimi algoritması ile belirlenen parametrelerle beş ayrı denemede elde edilen deney sonuçları ve bu deney sonuçlarına ait amaç fonksiyonun her denemedeki başlangıç ve bitiş değerleri ile bu beş denemede minimum ve maksimum başlangıç ve bitiş değerleri Tablo 5.25'te gösterilmiştir.

**Tablo 5. 25.** 25. Parametre Seti Deney Sonuçları

<b>T=5000 minT=0,001 coolRate=0,40 iterasyon=2500</b>						
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Maksimum</b>	<b>Minimum</b>
605605	707597	306101	703109	405598	707597	306101
116	615	132	121	119	615	<b><u>116</u></b>

Tablo 5.25'te gösterilen ceza katsayısı ile tanımlanan amaç fonksiyon değerinin tavlama benzetimi algoritması ile belirlenen 25. parametre seti ile beş ayrı denemede elde edilen deney sonuçları arasından amaç fonksiyonun minimum olduğu ilgili deney sonucuna ait grafik Şekil 5.26'da gösterilmiştir. Bu grafikte 25. parametre seti ile elde edilen deney sonuçları arasından amaç fonksiyonun minimize edildiği son değerlerden en küçük olan değer seçilmiştir. 25. parametre seti ile elde edilen fonksiyonun minimum değeri 116'tır.



**Şekil 5. 26.** 25. Parametre Seti Deney Sonuçları

## 6.SONUÇ ve ÖNERİLER

Bu çalışmada sınav çizelgeleme problemi için öncelikle matematiksel model kuruldu ve IBM ILOG OPL Optimization Studio yazılımı kullanılarak sınav çizelgeleme problemi optimize edilmeye çalışıldı. Daha sonrasında Microsoft Visual Studio 2015 yazılımı kullanılarak C# programlama dilinde sınav çizelgeleme problemine ait yapı kuruldu ve tavlama benzetimi algoritması geliştirilerek sınav çizelgeleme işleminin en iyi şekilde yapılması amaçlandı.

Elde edilen deney sonuçları dikkate alındığında sınav çizelgeleme problemi için oluşturulan yapıda geliştirilen tavlama benzetimi algoritması farklı parametre setleriyle çalıştırılmış ve deney sonuçları karşılaştırılmıştır.

Düşük soğutma oranı belirlendiği zaman iterasyon sayısı artırılarak fonksiyonun minimum değer elde edilmesi hedeflenmiştir.

Bu çalışmada kullanılan farklı parametre setleriyle elde edilen sonuçlar arasından fonksiyon değerinin minimum olduğu yani en iyi sonucun elde edildiği değer 115 olmuştur. Bu değere yakın elde edilen tüm değerler yerel minimumu oluşturmaktadır.

Elde edilen 115 değeri dikkate alındığında sınav çakışmaları giderilmiş, bir günde en fazla iki tane sınavın yapılabileceği sonucuna varılmış ve bir günde iki tane yapılacak sınav sayısı da 115'e düşürülmüştür.

Bu çalışmaya ait deney sonuçlarından anlaşılmaktadır ki sezgisel algoritmalarından olan tavlama benzetimi algoritmasının sınav çizelgeleme probleminin çözümü için en uygun çözüm yöntemlerinden biri olduğunu gösterilmiştir. İleride yapılacak çalışmalarda mevcut sınav çizelgelemesi problemine eklenecek yeni kısıtlar altında (Örneğin; Aynı günde iki sınava girecek öğrencilerin sınav aralıklarının belli bir saatten fazla olması, özel isteklere göre bazı sınavların belirli gün ve sınıflarda yapılması vb.) bu problem yeniden incelenebilir ve önerdiğimiz tavlama benzetim algoritması geliştirilebilir.

## KAYNAKÇA

- Abbas, A., & Tsang, E. (2004, Mayıs 1). Software engineering aspects of constraint-based timetabling- a case study. *Information and Software Technology*, 46(6), 359-372.
- Al-Yakoob, S. M., & Sherali, H. D. (2015, Eylül). Mathematical models and algorithms for a high school timetabling problem. *Computers & Operations Research*, 61, 56-68.
- Amaral, P., & Pais, T. C. (2016, Ağustos). Compromise ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling. *Computers & Operations Research*, 72, 160-174.
- Arani, T., Karwan, M., & Lofti, V. (1997, Ağustos). A Lagrangian relaxation approach to solve the second phase of the exam scheduling problem. *Omega*, 25(4), 473-482.
- Arbaoui, T., Boufflet, J. P., & Moukrim, A. (2016). A matheuristic for exam timetabling. *IFAC-PapersOnLine*, 49(12), 1289-1294.
- Asmuni, H., Burke, E. K., Garibaldi, J. M., McCollum, B., & Parkes, A. J. (2009, Nisan). An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables. *Computers & Operations Research*, 36(4), 981-1001.
- Babaei, H., Karimpour, J., & Hadidi, A. (2015, Ağustos). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86, 43-59.
- Balakrishnan, N. (1991). Examination scheduling: A computerized application. *Omega*, 19(1), 37-41.
- Barrera, D., Velasco, N., & Amaya, C.-A. (2012, Aralık). A network-based approach to the multi-activity combined timetabling and crew scheduling problem: Workforce scheduling for public health policy implementation. *Computers & Industrial Engineering*, 63(4), 802-812.
- Broek, J. v., Hurkens, C., & Woeginger, G. (2009, Ağustos 1). Timetabling problems at the TU Eindhoven. *European Journal of Operational Research*, 196(3), 877-885.
- Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007, Ocak 1). A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176(1), 177-192.
- Burke, E., Eckersley, A., McCollum, B., Petrovic, S., & Qu, R. (2010, Ekim 1). Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1), 46-53.
- Carrasco, M., & Pato, M. (2014, Şubat 16). A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. *European Journal of Operational Research*, 153(1), 65-79.
- Causmaecker, P. D., Demeester, P., & Berghe, G. V. (2009, Mayıs 16). A decomposed metaheuristic approach for a real-world university timetabling problem. *European Journal of Operational Research*, 195(1), 307-318.
- Chern, C.-C., Chien, P.-S., & Chen, S.-Y. (2008, Mayıs 1). A heuristic algorithm for the hospital health examination scheduling problem. *European Journal of Operational Research*, 186(3), 1137-1157.
- Dammak, A., Elloumi, A., & Kamoun, H. (2006, Ekim). Classroom assignment for exam timetabling. *Advances in Engineering Software*, 37(10), 659-666.

- Dimopoulou, M., & Miliotis, P. (2001, Nisan 1). Implementation of a university course and examination timetabling system. *European Journal of Operational Research*, 130(1), 202-213.
- Gans, O. B. (1981, Haziran). A computer timetabling system for secondary schools in the Netherlands. *European Journal of Operational Research*, 7(2), 175-182.
- Head, C., & Shaban, S. (2007, Nisan ). A heuristic approach to simultaneous course/student timetabling. *Computers & Operations Research*, 34(4), 919-933.
- Hertz, A. (1991, Eylül 15). Tabu search for large scale timetabling problems. *European Journal of Operational Research*, 54(1), 39-47.
- Kahar, M., & Kendall, G. (2010, Aralık 1). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research*, 207(2), 557-565.
- Leite, N., Fernandes, C. M., Melício, F., & Rosa, A. C. (2018, Haziran ). A cellular memetic algorithm for the examination timetabling problem. *Computers & Operations Research*, 94, 118-138.
- Lewis, R., & Thompson, J. (2015, Şubat 1). Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. *European Journal of Operational Research*, 240(3), 637-648.
- Li, J., Bai, R., Shen, Y., & Qu, R. (2015, Mayıs 1). Search with evolutionary ruin and stochastic rebuild: A theoretic framework and a case study on exam timetabling. *European Journal of Operational Research*, 242(3), 798-806.
- Lü, Z., & Hao, J.-K. (2010, Ocak 1). Adaptive Tabu Search for course timetabling. *European Journal of Operational Research*, 200(1), 235-244.
- Muklason, A., Parkes, A. J., Özcan, E., McCollum, B., & McMullan, P. (2017, Haziran). Fairness in examination timetabling: Student preferences and extended formulations. *Applied Soft Computing*, 55, 302-318.
- N.Pillay, & W.Banzhaf. (2009, Eylül 1). A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem. *European Journal of Operational Research*, 197(2), 482-491.
- Papoulias, D. (1980, Ocak). The assignment-to-days problem in a school time-table, a heuristic approach. *European Journal of Operational Research*, 4(1), 31-41.
- Qu, R., Burke, E. K., & McCollum, B. (2009, Ekim 16). Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2), 392-404.
- Sabar, N. R., Ayob, M., Kendall, G., & Qu, R. (2012, Şubat 1). A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 216(3), 533-543.
- Saviniec, L., O.Santos, M., & M.Costa, A. (2018, Şubat 16). Parallel local search algorithms for high school timetabling problems. *European Journal of Operational Research*, 265(1), 81-98.
- Skoullis, V. I., Tassopoulos, I. X., & Beligiannis, G. N. (2017, Mart). Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm. *Applied Soft Computing*, 52, 277-289.
- Weitz, R., & Lakshminarayanan, S. (1997, Ağustos). An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups, VLSI design, and exam scheduling. *Omega*, 25(4), 473-482.
- Werra, D. (1985, Şubat). An introduction to timetabling. *European Journal of Operational Research*, 19(2), 151-162.

- White, G. M., & Haddad, M. (1983). An heuristic method for optimizing examination schedules which have day and evening courses. *Computers & Education*, 7(4), 235-238.
- Woumans, G., Boeck, L. D., Beliën, J., & Creemers, S. (2016, Augustos 16). A column generation approach for solving the examination-timetabling problem. *European Journal of Operational Research*, 253(1), 178-194.
- Zhang, D., Liu, Y., M'Hallah, R., & C.H.Leung, S. (2010, Haziran 16). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 203(3), 550-558.



## EKLER

### EK-1 Tavlama Benzetim Algoritması Kodu

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication8
{
    class Program
    {
        //Her grubun hangi günde kaç sınavı olduğunu bulan fonksiyon
        public static int[,] find_days(int num_groups, int max_no_exam, int
num_days, int num_class, int num_times, int[] schedule, int[,] exam_info)
        {
            int[,] days = new int[num_groups, num_days];
            for (int i = 0; i < num_groups; i++)
                for (int j = 0; j < num_days; j++)
                    days[i, j] = 0;

            int exam = -2;
            for (int i = 0; i < num_groups; i++)
                for (int j = 0; j < max_no_exam; j++)
                {
                    exam = exam_info[i, j];
                    if (exam == -1)
                        j = max_no_exam;
                    else
                    {
                        int index = 0;
                        while (schedule[index] != exam)
                            index++;
                        int day = (int)(index / (num_times * num_class));
```

```

        days[i, day] = days[i, day] + 1;
    }
}
return days;

```

```

}

```

//Her grubun hangi günün hangi zamanında kaç sınavı olduğunu bulan  
fonksiyon

```

public static int[,] find_conflicting_exams(int num_groups, int
max_no_exam, int num_days, int num_class, int num_times, int[] schedule, int[,]
exam_info)

```

```

{

```

```

    int[,] time_slot_exam_info = new int[num_groups, num_days, num_times];

```

```

        for (int i = 0; i < num_groups; i++)

```

```

            for (int j = 0; j < num_days; j++)

```

```

                for (int k = 0; k < num_times; k++)

```

```

                    time_slot_exam_info[i, j, k] = 0;

```

```

    int exam = -2;

```

```

    for (int i = 0; i < num_groups; i++)

```

```

        for (int j = 0; j < max_no_exam; j++)

```

```

        {

```

```

            exam = exam_info[i, j];

```

```

            if (exam == -1)

```

```

                j = max_no_exam;

```

```

            else

```

```

            {

```

```

                int index = 0;

```

```

                while (schedule[index] != exam)

```

```

                    index++;

```

```

                int day = (int)(index / (num_times * num_class));

```

```

                int time_slot = index - day * num_times * num_class;

```

```

                time_slot = (int)(time_slot / num_class);

```

```

        time_slot_exam_info[i, day, time_slot] = time_slot_exam_info[i,
day, time_slot] + 1;
    }
}
return time_slot_exam_info;
}
public static int objective_function(int num_groups, int max_no_exam, int
num_days, int num_class, int num_times, int[] schedule, int[,] exam_info)
{
    int two_exam_penalty = 1;
    int three_exam_penalty = 500;
    int four_exam_penalty = 1000;
    int conflict_penalty = 100000;
    int[,] days = find_days(num_groups, max_no_exam, num_days,
num_class, num_times, schedule, exam_info);
    int[,] conflicts = find_conflicting_exams(num_groups, max_no_exam,
num_days, num_class, num_times, schedule, exam_info);
    int objective_value = 0;
    for (int i = 0; i < num_groups; i++)
        for (int j = 0; j < num_days; j++)
            {
                if (days[i, j] == 2)
                    objective_value = objective_value + two_exam_penalty;
                else if (days[i, j] == 3)
                    objective_value = objective_value + three_exam_penalty;
                else if (days[i, j] == 4)
                    objective_value = objective_value + four_exam_penalty;
            }
    for (int i = 0; i < num_groups; i++)
        for (int j = 0; j < num_days; j++)
            for (int k = 0; k < num_times; k++)
                {
                    if (conflicts[i, j, k] > 1)
                        objective_value = objective_value + conflict_penalty;
                }
}

```

```

    }

    return objective_value;
}

public static int find_classroom_no(int num_days, int num_classrooms, int
num_time_slots, int position)
{
    position = position % (num_days * num_classrooms * num_time_slots);
    position = position % num_classrooms;
    return position;
}

static void Main(string[] args)
{
    int num_days = 5;
    int num_classrooms = 21;
    int num_time_slots = 4;
    int num_exams = 375;
    int num_supervisor = 42;
    int no_student_groups = 52;
    int max_no_exam = 10;

    int[] classroom_capacity = new int[num_classrooms];
    int[] exam_size = new int[num_exams];
    int[] supervisor = new int[num_supervisor];

    int[] schedule = new int[num_days * num_classrooms * num_time_slots];

    //Her öğrenci grubunun kaç sınav alacağını gösteren dizim
    int[] studentsgrupexams = { 9, 6, 6, 8, 9, 7, 6, 7, 8, 7, 6, 7, 8, 7, 6, 6, 8, 7,
5, 8, 10, 6, 7, 5, 8, 8, 6, 9, 8, 9, 7, 7, 10, 7, 6, 7, 10, 7, 7, 6, 9, 6, 7, 7, 7, 7, 7, 8, 7, 6, 6
};
}

```

```
int[,] student_exam = new int[no_student_groups, max_no_exam];  
//Öğrenci gruplarının sınav bilgileri student_exam sınav dizininde tutuluyor.
```

```
//Her öğrenci grubu için max_no_exam kadar sınav tanımlanabilir.  
student_exam[i,j] i. öğrenci grubunun j. sınavının hangi sınav olduğunu
```

```
//gösteriyor. i ve j 0 dan başlıyor. Örneğin student_exam[5,6] = 23 demek  
5 nolu öğrenci grubunun 6. sınavı 23 nolu sınav demek oluyor.
```

```
//Eğer 5 nolu öğrenci grubunun toplam 7 tane sınavı varsa  
student_exam[5,7], student_exam[5,8] ve student_exam[5,9] -1 değerlerini alacak.
```

```
for (int i = 0; i < no_student_groups; i++)
```

```
for (int j = 0; j < max_no_exam; j++)
```

```
student_exam[i, j] = -1;
```

```
int count = 0;
```

```
for (int i = 0; i < no_student_groups; i++)
```

```
for (int j = 0; j < studentsgrupexams[i]; j++)
```

```
{
```

```
student_exam[i, j] = count;
```

```
count++;
```

```
}
```

```
for (int i = 0; i < classroom_capacity.Length; i++)
```

```
classroom_capacity[i] = 50; //sınıf kapasitelerini 50 belirle
```

```
for (int i = 0; i < exam_size.Length; i++)
```

```
exam_size[i] = 45; //sınava girecek öğrenci sayılarını 45 belirle
```

```
for (int i = 0; i < schedule.Length; i++)
```

```
schedule[i] = -1;
```

```
int exam = 0;
```

```
while (exam < num_exams)
```

```
{
```

```
Random random = new Random();
```

```
int randomNumber = random.Next(0, schedule.Length - 1);
```

```

        if (schedule[randomNumber] == -1)
        {
            schedule[randomNumber] = exam;
            exam++;
        }
    }

    int[,] days = find_days(no_student_groups, max_no_exam, num_days,
num_classrooms, num_time_slots, schedule, student_exam);
    //days[i,j] i. öğrenci grubunun j. günde kaç sınavı var onu gösteriyor. i ve
j 0 dan başlıyor
    int[,] exam_slots = find_conflicting_exams(no_student_groups,
max_no_exam, num_days, num_classrooms, num_time_slots, schedule, student_exam);
    //exam_slots[i,j,k] = i. öğrenci grubunun j. günün k. zamanında kaç sınavı
var onu gösteriyor. i,j ve k 0 dan başlıyor

    int objective_value = objective_function(no_student_groups,
max_no_exam, num_days, num_classrooms, num_time_slots, schedule, student_exam);
    Console.WriteLine("Objective Value = " + objective_value);
    double T = 5000;
    double minT = 0.001;
    double coolRate = 0.40;
    int numIterations = 2500;

    int best_objective_value = objective_value;
    int current_objective_value = objective_value;
    int[] best_solution = schedule;
    while (T > minT)
    {
        int iteration = 0;
        while (iteration < numIterations)
        {
            Random random = new Random();

```

```

int randomNumber1 = random.Next(0, schedule.Length - 1);
while (schedule[randomNumber1] == -1)
    randomNumber1 = random.Next(0, schedule.Length - 1);
int randomNumber2 = random.Next(0, schedule.Length - 1);
while (randomNumber2 == randomNumber1)
    randomNumber2 = random.Next(0, schedule.Length - 1);
int class1 = find_classroom_no(num_days, num_classrooms,
num_time_slots, randomNumber1);
int class2 = find_classroom_no(num_days, num_classrooms,
num_time_slots, randomNumber2);
iteration++;

int temp1 = schedule[randomNumber1];
int temp2 = schedule[randomNumber2];
schedule[randomNumber1] = temp2;
schedule[randomNumber2] = temp1;
//int[,] tdays = find_days(no_student_groups, max_no_exam,
num_days, num_classrooms, num_time_slots, tempschedule, student_exam);
//days[i,j] i. öğrenci grubunun j. günde kaç sınavı var onu gösteriyor.
i ve j 0 dan başlıyor
//int[,] texam_slots = find_conflicting_exams(no_student_groups,
max_no_exam, num_days, num_classrooms, num_time_slots, tempschedule,
student_exam);
//exam_slots[i,j,k] = i. öğrenci grubunun j. günün k. zamanında kaç
sınavı var onu gösteriyor. i,j ve k 0 dan başlıyor

int tobjective_value = objective_function(no_student_groups,
max_no_exam, num_days, num_classrooms, num_time_slots, schedule, student_exam);
if (tobjective_value <= current_objective_value)
{

current_objective_value = tobjective_value;
if (tobjective_value <= best_objective_value)
{

```



```
        Console.WriteLine(day + ". gün, " + classroom_no + ". sınıf, " +  
time_slot + ". zaman aralığında");  
  
        }  
        Console.ReadKey();  
    }  
}
```



## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Arafat Koca  
**Uyruğu** : T.C.  
**Doğum Yeri ve Tarihi** : Sorgun/YOZGAT 24.04.1990  
**e-mail** : arafatkoca@gmail.com

### EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Kalaba Anadolu Lisesi, Keçiören, Ankara	2009
Üniversite	: Kocaeli Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği, İzmit, Kocaeli	2013
Yüksek Lisans	: Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı, Meram,Konya	Devam ediyor

### İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2010	Ankara Büyükşehir Belediyesi-Ego Genel Müdürlüğü	Stajyer Mühendis
2011	Almin Alüminyum Profil A.Ş. San.Tic.Ltd.Şti	Stajyer Mühendis
2018	BIBA – Bremer Institut für Produktion und Logistik GmbH	Erasmus Stajyeri

### YABANCI DİLLER

İngilizce -Almanca

### YAYINLAR

Exam Scheduling Problem at Necmettin Erbakan University ( Yüksek Lisans Tezi)  
<http://www.ecsac2018.org/accepted-papers/>  
Estimation of the electricity consumption of Turkey trough artificial neural networks  
<http://ieeexplore.ieee.org/document/7846425/?reload=true>