

**T.C.  
SÜLEYMAN DEMİREL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇİZELGELEME PROBLEMLERİNE ÜST-SEZGİSEL ALGORİTMALAR  
TEMELLİ YENİ BİR ALGORİTMİK YAKLAŞIM**

**Cevriye ALTINTAŞ**

**Danışman  
Prof. Dr. Tuncay YİĞİT**

**DOKTORA TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
ISPARTA – 2020**



© 2020 [Cevriye ALTINTAŞ]

## TEZ ONAYI

Cevriye ALTINTAŞ tarafından hazırlanan "**Çizelgeleme Problemlerine Üst-Sezgisel Algoritmalar Temelli Yeni Bir Algoritmik Yaklaşım**" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı'nda DOKTORA TEZİ** olarak başarı ile savunulmuştur.

**Danışman**

**Prof. Dr. Tuncay YİĞİT**  
Süleyman Demirel Üniversitesi



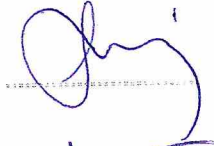
**Jüri Üyesi**

**Prof. Dr. Gültekin ÖZDEMİR**  
Süleyman Demirel Üniversitesi



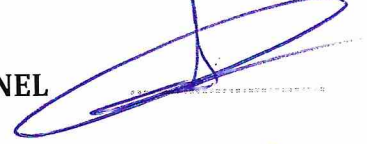
**Jüri Üyesi**

**Doç. Dr. İsmail Serkan ÜNCÜ**  
Isparta Uygulamalı Bilimler Üniversitesi



**Jüri Üyesi**

**Dr. Öğr. Üyesi Fatih Ahmet ŞENEL**  
Süleyman Demirel Üniversitesi



**Jüri Üyesi**

**Dr. Öğr. Üyesi Burhan DUMAN**  
Isparta Uygulamalı Bilimler Üniversitesi



**Enstitü Müdürü**

**Doç. Dr. Şule Sultan UĞUR**



## **TAAHHÜTNAME**

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

**Cevriye ALTINTAŞ**



## İÇİNDEKİLER

|                                                                                                | Sayfa |
|------------------------------------------------------------------------------------------------|-------|
| İÇİNDEKİLER.....                                                                               | i     |
| ÖZET.....                                                                                      | ii    |
| ABSTRACT .....                                                                                 | iii   |
| TEŞEKKÜR.....                                                                                  | iv    |
| ŞEKİLLER DİZİNİ.....                                                                           | v     |
| ÇİZELGELER DİZİNİ .....                                                                        | vi    |
| SİMGELER VE KISALTMALAR DİZİNİ .....                                                           | vii   |
| 1. GİRİŞ.....                                                                                  | 1     |
| 2. KAYNAK ÖZETLERİ .....                                                                       | 13    |
| 3. ÜST-SEZGİSELLERİN SINIFLANDIRILMASI YAKLAŞIMLARI.....                                       | 27    |
| 4. SEZGİSEL SEÇİM YÖNTEMLERİ.....                                                              | 29    |
| 4.1.Yapıcı Düşük Seviyeli Sezgisel Temelli Yaklaşımlar.....                                    | 29    |
| 4.2.Pertürbatif Düşük Seviyeli Sezgisel Temelli Yaklaşımlar .....                              | 31    |
| 4.2.1. Çevrimiçi öğrenmeli üst-sezgiseller .....                                               | 32    |
| 4.2.2. Çevrimdışı öğrenmeli üst-sezgiseller .....                                              | 32    |
| 5. SEZGİSEL ÜRETİM YÖNTEMLERİ.....                                                             | 33    |
| 6. MEMETİK ALGORİTMALAR .....                                                                  | 35    |
| 7. CROSS-DOMAIN SEZGİSEL ARAMA YARIŞMASI (CHESC2011/ CHESC2014)...                             | 37    |
| 7.1. CHESC2011’de Kullanılan Problem Alanları .....                                            | 38    |
| 7.2. Tek Boyutlu Bidon Paketleme Problemi(1D Bin Packing) .....                                | 38    |
| 7.3. Mantıksal Sağlanabilirlik Problemi- Boolean Satisfiability(MAX-SAT).....                  | 39    |
| 7.4. Permütasyon Akış Tipi Problemi (Permutation Flow Shop).....                               | 39    |
| 7.5. Personel Çizelgeleme (Personnel Scheduling) .....                                         | 39    |
| 7.6. Gezgin Satıcı Problemi (Travelling Salesman Problem) .....                                | 40    |
| 7.7. Araç Rotalama Problemi (Vehicle Routing Problem) .....                                    | 40    |
| 8. SINAV ÇİZELGELEME PROBLEMİ.....                                                             | 41    |
| 8.1. Sınav Çizelgeleme Problemi: Modeli .....                                                  | 42    |
| 8.2. Sınav Çizelgeleme Problemi: Algoritmalar .....                                            | 43    |
| 8.3. Sınav Çizelgeleme Modeli .....                                                            | 43    |
| 8.3.1. Problem tanımı.....                                                                     | 44    |
| 8.3.2. Çözümler için framework.....                                                            | 46    |
| 8.4. Sınav Çizelgeleme Modelinin Detayları .....                                               | 48    |
| 8.4.1. Sert kısıtlar.....                                                                      | 49    |
| 8.4.2. Esnek-Sert kısıtlar.....                                                                | 49    |
| 8.4.3. Periyot ve derslik ilişkili: esnek kısıtlar.....                                        | 49    |
| 8.4.4. Esnek kısıtların hesaplanması.....                                                      | 49    |
| 9. DERS ÇİZELGELEME PROBLEMİ.....                                                              | 64    |
| 9.1. Ders Çizelgeleme Problemi: Modeli ve Problem Tanımı .....                                 | 65    |
| 10. DERSLİK KULLANIM ORANLARINI ARTIRMAK İÇİN BİR KARAR DESTEK<br>SİSTEMİ GELİŞTİRİLMESİ ..... | 73    |
| 11. ARAŞTIRMA BULGULARI VE TARTIŞMA .....                                                      | 90    |
| 12. SONUÇ VE ÖNERİLER .....                                                                    | 132   |
| KAYNAKLAR .....                                                                                | 134   |
| ÖZGEÇMİŞ.....                                                                                  | 154   |

## ÖZET

Doktora Tezi

### ÇİZELGELEME PROBLEMLERİNE ÜST-SEZGİSEL ALGORİTMALAR TEMELLİ YENİ BİR ALGORİTMİK YAKLAŞIM

Cevriye ALTINTAŞ

Süleyman Demirel Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Tuncay YİĞİT

Optimizasyon algoritmalarının tasarımı derin bir problem analizi gerektiren uzun bir süreçtir. Algoritmaların verilen bir problem örneğini etkili bir şekilde çözmesi beklenir. Ancak, bu algoritmaların bütün problemlere uygulanması oldukça zordur. Meta-sezgiseller bu sorun ile baş etmeye çalışmaktadırlar. Bununla birlikte birçok çalışmada, meta-sezgiseller problem-bağımlı yöntemler olarak uygulanmıştır. Üst-sezgiseller ise arama ve optimizasyon algoritmalarından farklı olarak problem-bağımsız çözüm fırsatları verir. Üst-sezgiseller zor arama problemlerini çözmek için otomatik sezgisel tasarım amacı doğrultusundaki bir dizi yaklaşımdan oluşmaktadır. Stratejik araştırmalardaki zorluk, uygulanabilir arama yöntemlerinin daha genel olarak geliştirilmesidir. Üst-sezgisel terimi oldukça yenidir; ilk olarak 2000 yılında kombinatoriyal optimizasyon kavramı kapsamında sezgiselleri seçen sezgiseller olarak tanımlanmıştır. Ancak, otomatik sezgisellerin tasarımı fikri yeni değildir ve başlangıç noktası 1960'lara kadar gözlemlenebilir. Üst-sezgisel tanımı son zamanlarda arama problemlerini çözmek için üretilen veya seçilen sezgiseller için bir öğrenme mekanizması veya arama yöntemi olarak genişletilmiştir. İki temel üst-sezgisel kategorisi vardır: sezgisel seçim ve sezgisel üretim. Üst-sezgisellerin en önemli özellikleri verilen problem için direkt olarak çözüm uzayı üzerinde arama yapmak yerine sezgisellerin veya sezgisel bileşenlerin arama uzayında işlem yapmalarıdır. Bu tezde üst-sezgiseller için bilimsel yazındaki çalışmalar ve üst-sezgisel yaklaşımları detaylı olarak incelenerek yeni yöntemler geliştirilecektir. Ayrıca üniversite çizelgeleme problemleri için üst-sezgisel yaklaşımları kullanan çözümler gerçekleştirilecek, üniversite mekân kullanım analizleri yapılacak ve gelecekteki araştırmalar tartışılacaktır.

**Anahtar Kelimeler:** Üst-sezgisel, meta-sezgisel, memetik algoritmalar, üniversite çizelgeleme problemleri, üniversite mekân kullanımı, karar destek sistemleri.

2020, 155 sayfa

## **ABSTRACT**

### **Ph.D. Thesis**

## **A NEW ALGORITHMIC APPROACH TO TIMETABLING PROBLEMS BASED ON HYPER-HEURISTIC ALGORITHMS**

**Cevriye ALTINTAŞ**

**Süleyman Demirel University  
Graduate School of Natural and Applied Sciences  
Department of Physics**

**Supervisor: Prof. Dr. Tuncay YİĞİT**

Designing a optimisation algorithm is a time-consuming process requiring an in-depth analysis of the problem. The resulting algorithm is expected to be effective for solving a given set of target problem instances. However, since the algorithm is dedicated, it is hard to adapt and to apply to other problems. Meta-heuristics were brought in to cope with this drawback. Nevertheless, in most of the meta-heuristic studies, the employed meta-heuristics have been implemented as rather problem-dependent methodologies. Hyperheuristics furnish problem-independent management opportunities differently from such search and optimisation algorithms. Hyper-heuristics comprise a set of approaches which are motivated by the goal of automating the design of heuristic methods to solve hard computational search problems. An underlying strategic research challenge is to develop more generally applicable search methodologies. The term hyper-heuristic is relatively new; it was first used in 2000 to describe heuristics to choose heuristics in the context of combinatorial optimisation. However, the idea of automating the design of heuristics is not new; it can be traced back to the 1960s. The definition of hyper-heuristics has been recently extended to refer to a search method or learning mechanism for selecting or generating heuristics to solve computational search problems. Two main hyperheuristic categories can be considered: heuristic selection and heuristic generation. The distinguishing feature of hyper-heuristics is that they operate on a search space of heuristics (or heuristic components) rather than directly on the search space of solutions to the underlying problem that is being addressed. This paper presents a critical discussion of the scientific literature on hyper-heuristics including their origin and intellectual roots, a detailed account of the main types of approaches, and an overview of some related areas. This thesis presents an investigation into the use of hyper-heuristic approaches to improve solutions for university timetabling problems. Current research trends and directions for future research are also discussed.

**Keywords:** Hyper-heuristic, heuristic, memetic algorithms, university timetabling, space utilisation, decision support systems.

**2020, 155 pages**

## TEŞEKKÜR

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli Danışman Hocam Prof.Dr. Tuncay YİĞİT'e teşekkürlerimi sunarım. Literatür araştırmalarımnda yardımcı olan değerli hocalarım Prof. Dr. Gültekin ÖZDEMİR ve Doç.Dr. Ender ÖZCAN'a ve çalışmalarım sırasında yaptığı yönlendirmeler ve tüm yardımları için Doç. Dr. İsmail Serkan ÜNCÜ'ye, Dr. Öğr. Üyesi Fatih Ahmet ŞENEL'e, Dr.Öğr. Üyesi Burhan Duman'a teşekkür ederim.

4538-D1-16 No'lu Proje ile tezimi maddi olarak destekleyen Süleyman Demirel Üniversitesi Bilimsel Araştırma Projeleri Yönetim Birimi Başkanlığı'na teşekkür ederim.

Tezimin gerçekleşmesinde 1059B141300037 numaralı 2214-A destekli proje ile maddi destek sağlayan TÜBİTAK'a teşekkür ederim.

Tezimin her aşamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Cevriye ALTINTAŞ  
ISPARTA, 2020

## ŞEKİLLER DİZİNİ

|                                                                                                                                                                               | Sayfa |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Şekil 1.1. Derslik Kullanım Oranı.....                                                                                                                                        | 10    |
| Şekil 1.2. Fakülte Binası Derslik Kullanım Sıklığı .....                                                                                                                      | 10    |
| Şekil 3.1. Üst-sezgisel yaklaşımların sınıflandırılması, (i) sezgisel arama uzayı yapısı ve (ii) geri bildirim öğrenme mekanizmaları (Burke vd., 2010d, Burke vd., 2019)..... | 27    |
| Şekil 9.1 Eğitim Alanları Tahsisi .....                                                                                                                                       | 66    |
| Şekil 10.1. DSS Akış Diyagramı .....                                                                                                                                          | 75    |
| Şekil 10.2. Derslik İhtiyacı Karar Destek Sistemi Modeli.....                                                                                                                 | 76    |
| Şekil 10.3. Haftalık Derslik Doluluk Oranları.....                                                                                                                            | 79    |
| Şekil 10.4. Tüm Fakülte/YO/MYO'ların Pazartesi Günü Derslik Alanlarının Doluluk Oranları .....                                                                                | 80    |
| Şekil 10.5. Tüm Fakülte/YO/MYO'ların Haftalık Derslik Alanlarının Doluluk Oranları .....                                                                                      | 80    |
| Şekil 10.6. Derslik İhtiyacı Karar Destek Sistemi Optimizasyon Modeli .....                                                                                                   | 81    |
| Şekil 10.7. Ders Programı Optimizasyon Modeli.....                                                                                                                            | 81    |
| Şekil 11.1. SSMMA algoritması akış diyagramı .....                                                                                                                            | 93    |
| Şekil 11.2. SSMA ve TGMA 31 çalışma sonunda elde edilen ortalama değerleri .....                                                                                              | 98    |
| Şekil 11.3. Max-Sat problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                                    | 102   |
| Şekil 11.4. Bin Packing problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                                | 103   |
| Şekil 11.5. Personel Çizelgeleme problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                       | 104   |
| Şekil 11.6. Akış Tipi İş Çizelgeleme problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                   | 105   |
| Şekil 11.7. Gezgin Satıcı problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                              | 106   |
| Şekil 11.8. Araç Rotalama problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA .....                                              | 107   |
| Şekil 11.9. SSMMA algoritması ile rasgele seçilen Max-SAT probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                                        | 110   |
| Şekil 11.10. SSMMA algoritması ile rasgele seçilen Bin Packing probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                                   | 113   |
| Şekil 11.11. SSMMA algoritması ile rasgele seçilen Personel çizelgeleme probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                          | 116   |
| Şekil 11.12. SSMMA algoritması ile rasgele seçilen Akış Tipi Çizelgeleme probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                         | 119   |
| Şekil 11.13. SSMMA algoritması ile rasgele seçilen Gezgin-Satıcı probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                                 | 122   |
| Şekil 11.14. SSMMA algoritması ile rasgele seçilen Araç Rotalama probleminin 31 deney sonucuna göre evrimsel aktivitesi .....                                                 | 125   |
| Şekil 11.15. CHeSC2011 algoritmaları ile SSMMA algoritmasının karşılaştırması.....                                                                                            | 127   |
| Şekil 11.16. Fakülte Saatlere Göre Sıralanmış Haftalık Derslik Alanlarının Kullanım Çizelgesi(1.Öğretim) .....                                                                | 130   |
| Şekil 11.17. Fakülte Saatlere Göre Sıralanmış Haftalık Derslik Alanlarının Kullanım Çizelgesi(2.Öğretim) .....                                                                | 130   |
| Şekil 11.18. Fakülte Saatlere Göre Derslik Kullanım Analizi.....                                                                                                              | 131   |

## ÇİZELGELER DİZİNİ

|                                                                                                                                                                         | Sayfa |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Çizelge 1.1. Fakülte binası Pazartesi günü derslik kullanım analizi .....                                                                                               | 11    |
| Çizelge 1.2. Fakülte binası Salı günü derslik kullanım analizi.....                                                                                                     | 11    |
| Çizelge 4.1. Düşük seviyeli sezgiselleri temel alarak çalışılan sezgisel seçim yöntemleri alanları .....                                                                | 29    |
| Çizelge 4.2. Pertürbatif sezgiselleri seçmek için en çok çalışılan uygulama alanı yöntemleri .....                                                                      | 31    |
| Çizelge 5.1. Sezgisel Üretim Yöntemlerinin Uygulama Alanları.....                                                                                                       | 34    |
| Çizelge 7.1. CHeSC2011 Problemlerindeki Düşük Seviyeli Sezgiseller .....                                                                                                | 38    |
| Çizelge 8.1. Üniversite Örneklerinin Özellikleri.....                                                                                                                   | 63    |
| Çizelge 8.2. Üniversite Örneklerinin Kısıt Özellikleri .....                                                                                                            | 63    |
| Çizelge 9.1. Kısıtlar.....                                                                                                                                              | 69    |
| Çizelge 9.2. Üniversite test verileri .....                                                                                                                             | 71    |
| Çizelge 10.1. Tüm Fakülte/YO/MYO'ların Pazartesi Günü Derslik Alanlarının Kullanım Çizelgesi .....                                                                      | 78    |
| Çizelge 10.2. Stres Testi Sonucu Elde Edilen Çizelge Örneği .....                                                                                                       | 82    |
| Çizelge 10.3. En İyi Zamanlama Analizi Sonucu Elde Edilen Çizelge Örneği.....                                                                                           | 83    |
| Çizelge 11.1. Her bir Problemdaki Düşük Seviyeli Sezgisellerin Sayısı {Mutasyon (MU) Yeniden Oluşturma(RE-CREATE) ile birlikte, Yerel arama (HC), Çaprazlama (XO)}..... | 93    |
| Çizelge 11.2. Parametre Değerleri.....                                                                                                                                  | 94    |
| Çizelge 11.3. Memetik Algoritmaların CHeSC2011 yarışmasındaki sıralaması .....                                                                                          | 95    |
| Çizelge 11.4. Self-Adaptif Memetik Algoritmasının CHeSC2011 yarışmasındaki sıralaması .....                                                                             | 95    |
| Çizelge 11.5. SSMMA ile Multi-Meme SSMA CHeSC2011 yarışmasındaki karşılaştırması .....                                                                                  | 99    |
| Çizelge 11.6. SSMMA ile SSMA CHeSC2011 yarışmasındaki karşılaştırması .....                                                                                             | 100   |
| Çizelge 11.7 SSMMA ile TGMA CHeSC2011 yarışmasındaki karşılaştırması .....                                                                                              | 101   |
| Çizelge 11.8. ITC2007 verileri ile SSMMA sonuçlarının karşılaştırması .....                                                                                             | 129   |
| Çizelge 11.9. SDU verilerinin SSMMA ile çözüm sonuçları.....                                                                                                            | 129   |
| Çizelge 12.1. Üniversite Veri Seti .....                                                                                                                                | 133   |
| Çizelge 12.2. ITC2007 Veri Seti.....                                                                                                                                    | 133   |

## SİMGELER VE KISALTMALAR DİZİNİ

|                |                                                             |
|----------------|-------------------------------------------------------------|
| $E$            | Sınavların kümesi.                                          |
| $s_i^E$        | $i \in E$ . $E$ kümesinin elemanı olan $i$ . sınavın boyutu |
| $d_i^E$        | $i \in E$ . $E$ kümesi olan $i$ . sınavın süresi            |
| $\cup_i d_i^E$ | Kullanılan sınav sürelerin kümesi                           |
| $S$            | Öğrencilerin kümesi.                                        |
| $R$            | Dersliklerin kümesi.                                        |
| $s_r^R$        | $r \in R$ .-derslik büyüklüğü                               |
| $w_r^R$        | $r$ dersliğinin ceza ağırlığı.                              |
| $P$            | Sürelerin kümesi.                                           |
| $d_p^P$        | $p \in P$ . $p$ zaman diliminin süresi.                     |
| $w_p^P$        | $p$ süresinin ceza ağırlığı.                                |
| $w^{2R}$       | “arka arkaya ikis ınav” ağırlığı                            |
| $w^{2D}$       | “bir günde iki sınav” ağırlığı                              |
| $w^{PS}$       | “süre yayılım” ağırlığı                                     |
| $w^{NMD}$      | “karışık olmayan süre” ağırlığı                             |
| $w^{FL}$       | “ön yük cezası” ağırlığı                                    |

## 1. GİRİŞ

Araştırmacılar, kullanıcılar, kamu kurumları veya şirketler günlük yaşam içerisinde çok sayıda planlama ve optimizasyon problemleri ile karşılaşır (Rothlauf, 2011). Problem çözmek sıradan bir süreç olup, problemi çözmek için etkili bir çözüm stratejisi aranır. Bu çözüm stratejisinin etkinliği, çözümün beklentilerine ve gereksinimlerine göre ölçülmelidir. Örneğin; acil bir çözüm gerekiyorsa, çözüm stratejisinin hızı stratejinin etkinliği için temel ölçüdür. Bir problem çözülürken amaçlara göre çözüm stratejisine ihtiyaç vardır. Bir problemin amacı özel bir örnek için zaten var olan bir cevabı tanımlayabilir, örneğin; mevcut ulaşım imkânlarını kullanarak A şehrinden B şehrine 30 dakikada ulaşım yapıp yapılamayacağı. Bu türdeki sorular karar problemleri olarak adlandırılır. Karar problemlerinin yanı sıra, bir problem için çözüm aramak optimizasyon problemleri altında incelenmiştir. A noktasından B noktasına en hızlı ulaşım ise optimal çözüm olarak örnek gösterilebilir. Linear/integer programlama yöntemleri gibi kesin algoritmalar optimal çözümü garanti ederler. Ancak optimal çözümü bulmak birçok optimizasyon problemi için zordur. Bu yüzden yaklaşık-optimal çözümler optimal çözüm olarak kabul edilebilir. Uygunluk fonksiyonu problemin amacına göre çözümlerin kalitesinin ölçülmesidir. Sezgisel stratejiler makul bir sürede en iyi çözümü bulmak için arama sürecini hızlandırırlar. Bu algoritmalar optimuma ulaşmayı garanti etmemelerine rağmen, birçok çalışma optimuma yakın çözümleri etkili bir şekilde bulduklarını göstermiştir (Mısıır, 2012b). “Eniyileme” anlamına gelen optimizasyon aynı problem üzerinde çalışan kişilerin veya işletmelerin, çözüm kalitesini artırarak rekabet ortamında bir adım öne geçebilmelerini sağlayacak önemli bir kavramdır. Optimizasyon teknikleri de yapılmış veya yapılmakta olan işin en iyi çözümünü ortaya koymak için kullanıldığından küreselleşen dünya pazarı ve buna bağlı olarak artan rekabet ortamında işletmelere problemlerini çözmede yardımcı olacaktır. Optimizasyon her alanda kullanılabilmektedir. İnşaattan, ekonomiye, eğitime, sağlığa, tasarım optimizasyonundan, lojistik sektörüne birçok alanda oluşabilecek olan probleme optimizasyon teknikleri yardımı ile en iyi çözüm aranır.

Optimizasyon, en basit tanımı ile eldeki kısıtlı kaynakların en iyi şekilde kullanılmasıdır. Matematiksel olarak ise optimizasyon, bir fonksiyonunun amacına uygun olarak maksimize veya minimize edilmesi yani amaç fonksiyonunun en iyi değerini veren kısıtlardaki değişkenlerin değerinin bulunmasıdır.

Kombinatoryal optimizasyon belirli bir problem için olası çözümler içerisinde en iyi çözümü bulma problemi olarak tanımlanabilir (Hoos ve Stützle, 2005). Bu problemlere çizelgeleme, üretim planlama, rotalama, ekonomik sistemler ve yönetim gibi birçok gerçek dünya problemleri içerisinde sıklıkla karşılaşılır (Hoos ve Stützle, 2005). Gerçek dünyadaki birçok optimizasyon problemi karmaşıktır ve çözülmesi oldukça zordur. Bunun nedeni büyük ve yoğun kısıtlar, çok karmaşık işlemlerle modellenen arama uzaylarıdır (Weise vd., 2009). Genellikle sezgisel yöntemler bu problemleri çözmek için kullanılırlar. Sezgisel yöntemlerin temel amacı, optimum çözümü (veya optimuma yakın çözümü) garanti etmemekle beraber, gerçeğe uygun bir sürede bulmaktır (Talbi, 2009; Gendreau ve Potvin, 2010). Meta-sezgisel algoritmalar arama uzayındaki etkin yönlendirme yapmak için bazı yüksek seviyeli kontrol stratejileri sağlarlar. Bir çok meta-sezgisel algoritma ve onların hibridleri optimizasyon problemlerini çözmek için uygulanmıştır. Meta-sezgisel algoritmalara örnekler; dağınık arama (scatter search), tabu arama (tabu search), genetik algoritmalar, genetik programlama, memetik algoritmalar, değişken komşuluk arama (variable neighborhood search), güdümlü yerel arama (guided local search), GRASP (greedy randomized adaptive search procedure), karınca kolonisi (ant colony optimization), benzetimli tavlama (simulated annealing), iteratif yerel arama (iterated local search), çoklu-başlama metodları (multi-start methods), hibrid metasezgiseller (Azimi, 2005; Burke vd., 2010a,b) ve paralel stratejilerdir (Talbi, 2009; Gendreau ve Potvin, 2010).

Belirli bir problem göz önüne alındığında, akla gelen ilginç bir soru: *Hangi algoritma eldeki sorun için en uygundur ve en uygun yapı ve parametre değerleri nelerdir?*

Yukarıdaki sorunun en basit cevabı mevcut olan çeşitli meta-sezgiseller içerisinde en uygun olanı bulmak için deneme-yanılma yöntemi kullanmak ve sonra uygun yapıları ve parametre değerlerini belirlemek için deneme-yanılma ile değerleri belirlemektir. Bu cevap makul görünmesine rağmen hesaplama süresi açısından birçok gerçek dünya problemleri için pratik bir uygulama değildir. Yıllar boyunca önerilmiş birçok ısmarlama meta-sezgisel algoritmaların elle tasarlanması ve uyarlanması belirli bir problem örneği için iyi sonuçlar üretmeye odaklanmışlardır. Elle tasarlanan bu algoritmalar sadece birkaç problem örneği için yüksek kalitede çözümler üretebilmişlerdir, fakat genellikle diğer optimizasyon problemlerine direkt olarak uygulanamazlar ve aynı problemin diğer örnekleri üzerinde de genellikle başarısız olurlar. Tabiki, “No Free Lunch Teoremi” (Wolpert ve Macready, 1997) durumu genel arama yöntemlerinde yoktur, fakat bu böyle bir algoritmanın sınırlarını keşfetmek için daha genel bir arama algoritması bulunamayacağı anlamına gelmez (Hamadi vd., 2011; Poli ve Graff, 2009; Burke vd., 2010c). Bir veya birkaç problem örneği üzerinde iyi sonuçlar elde edebilmek için otomatik arama yöntemlerini geliştirmek amacıyla çok sayıda çalışma yapılmıştır. Üst sezgisel (Hyper-heuristics) (Burke vd., 2010b), meta-öğrenme (meta-learning) (Smith-Miles, 2008), parametre optimizasyonu (parameter tuning) (Eiben vd., 1999), reaktif arama (reactive search) (Battiti ve M. Brunato, 2010), adaptif memetik algoritmalar (adaptive memetic algorithms) (Ong ve Keane, 2004) ve çoklu metod (multi-method) (Vrugt vd., 2009) sadece bazı örneklerdir. Herhangi bir arama yönteminin performansı onların yapısına ve parametre değerlerine bağlıdır (Burke vd., 2010d). Ayrıca, farklı arama yöntemleri, farklı yapı ve parametre değerleri ile ilişkilendirildiğinde farklı problemler veya problem örnekleri ile uğraşması gerekebilir (Smith-Miles, 2008; Eiben vd., 1999). Bir arama belirli bir örneği çözmek için adaptasyondan yararlanabilir. Bu yüzden, herhangi bir arama algoritmasının performansı problemi çözme süresinde parametre değerlerinin veya yapısının otomatik ayarlanması ile iyileştirilebilir. Bu nedenle, otomatik sezgisel tasarımın nihai hedefi problem çözme sürecinde yapıları veya parametre değerlerini ayarlamak ve aynı sorunun farklı örnekleri arasında, sadece iyi çalışabilecek durumda arama yöntemleri geliştirmek için

değil, aynı zamanda bir çok farklı problem üzerinde iyi çalışabilecek arama yöntemleri geliştirmektir (Burke vd., 2010e; Smith-Miles, 2008; Eiben vd., 1999). Gerçek dünyadaki arama problemlerinin çözümünde sezgisel yöntemler ve diğer arama yöntemlerinin başarılı olmasına rağmen, bunların yeni problemlere veya benzer problemin farklı örneklerine kolaylıkla uygulanmasında hala bazı zorluklar vardır. Bu zorluklar bu yaklaşımları kullanırken önemli parametre aralığı ve algoritma seçimini içerir ve nasıl seçilecekleri konusundaki rehber eksikliği olarak ortaya çıkar. Buna ek olarak, bilimsel toplulukların fikrine göre farklı sezgisellerin farklı durumlarda etkin bir şekilde çalışmasında hangi yaklaşımları hangi durumlara kullanmak için basit seçimlerin neden olanaksız olduğudur. Mevcut tekniklerin diğer bir dezavantajı, gerçek-dünya problemleri için yeni yaklaşımlar geliştirmek ve sürdürmek için pahalı ısmarlama ve probleme özel yöntemlerin geliştirilmesi eğiliminin olmasıdır. Bu alan için motive edici amaç zor arama problemlerini çözmek için sezgisel yöntemlerin ayarlanması ve otomatik tasarımın yapılmasındaki zorluktur (Burke vd., 2003a, b; Ross, 2005a, b; Burke vd., 2010a). Ana fikir var olan arama yöntemlerinin uygulanmalarının daha genel algoritmalar olarak geliştirilmesidir. Üst-sezgiseller kullanılırken, direkt olarak problemi çözmek yerine verilen durumda sezgisel dizisi veya doğru yöntemi bulmak hedeflenmektedir. Üst-sezgiseller sipariş üzerine yapılmış tekniklere karşı kullanıma hazır yöntemler olarak kabul edilebilir. Bu yüzden, en önemli hedef genel yöntemlerin tasarımıdır, bu kolay uygulanan düşük-seviyeli sezgiseller ile uygun kalitede çözüm üretilmesini sağlayacaktır. Bir üst-sezgisel yüksek-seviyeli bir yöntem olarak görülebilir, bu yöntem ile belirli bir problem örneği veya örnekler kümesi ve bir dizi düşük-seviyeli sezgisel (veya bileşenleri) verildiği zaman verilen problemi etkili bir şekilde çözmek için bu bileşenleri uygun şekilde kombine ederek otomatik olarak üretilir.

Üst-sezgisel terimi ilk olarak 2000 yılında hakemli bir konferans makalesinde kullanılmıştır (Cowling vd., 2000). İlk makaledeki fikir daha sonra çizelgeleme problemleri için (Cowling vd., 2001, 2002a, b)'de geliştirilmiş ve uygulanmıştır. Bu yayınlarda, bir üst-sezgisel yüksek seviyeli bir yaklaşım olarak düşünülmüştür, burada verilen belirli bir problem örneği ve düşük-seviyeli

sezgisel kümesi ile her bir karar noktasında uygun düşük seviyeli sezgisel seçilebilir ve uygulanabilir. Bu terimin ilk görünmesi teknik bir raporda (Denzinger vd., 1996) bulunabilir, burada farklı bir kavram içinde kullanılmıştır ve bu yaklaşım otomatik teorem ispatı için Yapay Zeka algoritmaları kapsamında kombine edilerek tanımlanmıştır. Bu raporda üst-sezgiselleri tanımlama amacı yoktur ve terim sadece bir kez kullanılmıştır. Sezgisellerin otomatik seçimi ve tasarımı fikridir, ancak çok eskidir.

Üst-sezgisel terimi 2003 yılında (Burke vd., 2003a), tartışılmıştır ve optimizasyon sistemlerindeki genellik düzeyini artırmak için çalışabilen üst-sezgiseller fikri üzerine vurgulama yapılmıştır. Bu yayın aynı zamanda ilgili kısa bir kaynak özeti sunar ve bazı detaylı örnekler verir. Daha sonra Ross (2005) tarafından yayınlanan bir makalede sadece üst-sezgisel uygulaması yaklaşımı ile yararlı ipuçları verilmekle kalmaz, aynı zamanda gelecekteki uygulama alanlarının tanımı ve ilgili araştırma alanları tartışılmıştır. Yakın bir zamandaki yayında (Chakhlevitch ve Cowling, 2008) karmaşık gerçek-dünya problemleri üzerinde üst-sezgisellerdeki son gelişmeler tartışılmış ve bir sınıflandırma sunulmuştur. Bu bölümde bu yaklaşımların bir tanımı olarak üç yararlı kriter sunulmuştur, bunlar; bir üst-sezgisel (i) düşük-seviyeli sezgiselleri yöneten bir yüksek seviyeli sezgisel, (ii) iyi bir çözümden ziyade problemi çözmek için iyi bir yöntem aramak, ve (iii) sadece sınırlı probleme özgü bilgi kullanan olarak verilmiştir. Yazarlar son kriteri oldukça önemli görmüşlerdir. Son zamanlardaki inceleme ve kitap bölümü (Burke vd., 2009)'de potansiyel sezgisel bileşenlerden yeni sezgisel üretim yöntemleri tartışılmıştır, burada Genetik Programlama önemli rol oynamaktadır. Bu bölüm yaklaşımı uygulamak için gerekli adımları detaylı bir şekilde tanımını içerir, bazı çalışma örnekleri verir, kısa bir kaynak özeti sunar, ve üst-sezgisellerin sınıflandırılmasıyla ilgili tartışmaları içerir. Son olarak (Burke vd., 2010d, Burke vd., 2019) önceki üst-sezgisel sınıflandırması ile ilgili bilgi verir ve bu alandaki çalışmaları dikkate alarak birleştirilmiş bir sınıflama sunar. Üst-sezgisel tanımı, burada, "arama problemlerini çözmek amacıyla sezgisellerin seçilmesi veya üretilmesi için bir arama metodu veya öğrenme mekanizması" olarak yapılmıştır.

Çizelgeleme, farklı karakterleri olan farklı problemlerin bulunduğu çok etkin bir araştırma alanıdır. Birçoğu bazı olayların belirli zamanlara atanmasıdır. Bu atama esnasında bazı gereklilikler dikkate alınmalıdır ki bunlar mümkün olduğunca gerçekleştirilmesi gereken kısıtlar olarak da bilinir. Çizelgeleme çeşitli kategorilere ayrılabilir, örneğin, eğitim ile ilgili çizelgelemeler, personel çizelgeleme, sportif faaliyetlerin çizelgelenmesi, taşımacılık çizelgelemeleri vb. Bu çizelgeleme problemlerinin her biri farklı yapı, kısıt ve gereklilikleri içerir. Bu tezde yüksek kalitede çözüm üretmenin çok zor olduğu ve çok popüler bir araştırma alanı olan üniversite çizelgeleme problemleri (ÜÇP) üst-sezgisel yöntemler kullanılarak çözülmeye çalışılacaktır. Üniversite çizelgeleme problemleri sınav çizelgeleme (SÇP) ve ders çizelgeleme (DÇP) olarak ikiye ayrılabilir. Carter ve Laporte (1996) her iki çizelgedeki ortak karakteristikleri tanımlamışlardır. Carter ve Laporte (1998) ders çizelgeleme problemini öğrenci ve öğretim elemanının derslerinin dersliklere atanması olarak tanımlamışlardır. Carter ve Laporte (1998) sınav çizelgeleme problemini sınavların sınırlı bir süre içerisinde çakışma olmadan atanması olarak tanımışlardır. Pratikte, elle çözüm üretilen sınav çizelgeleme problemi öğrenci sayısının, sınav sayısının çok fazla olduğu ve derslik sayısının az olduğu kurumlar için fazla zaman harcanması gereken bir süreçtir. Bu problem yaklaşık 40 yıl önce tanımlanmış ve üzerinde çalışılmaya başlanmıştır. Birçok araştırmacı en iyi sınav çizelgeleme için etkili bilgisayar programları veya sistemleri geliştirmek için çaba harcamıştır. Optimizasyon problemleri Kombinatoryal ve Sürekli olarak ikiye ayrılmıştır. SÇP ise kombinatoryal optimizasyon problemleri içerisinde tanımlanmıştır. (Papadimitriou ve Steiglitz, 1982). SÇP, NP(nondeterministic polinomial)-zor problemler arasında sınıflandırılmıştır. Örneğin; çizge boyama problemi, bidon paketleme problemi vb. Eğer bir problem NP olarak sınıflandırılmışsa, tanımlanamayan polinom zamanda çözülebilmektedir. NP sınıflandırmanın içerisindeki problemler, büyük boyutta olup en iyi çözümü bulmak çok uzun süre gerektirmektedir. Bu tezde üniversitelerdeki SÇP ve DÇP problemleri için mümkün olduğunca en iyi çözüm üretmek için çeşitli tasarımlar gerçekleştirilecektir. Gerçekleştirdiğimiz algoritma ile üretilen çözümler ile diğer algoritmalarla üretilen çözümler karşılaştırılarak, kullanılan algoritmaların etkinliği tartışılacaktır. Tasarlanan sistem ile parametreler mümkün olduğunca

alıřma zamanında otomatik olarak ayarlanacaktır. Bu alıřma yapılırken st-sezgisellerin alıřma matıęı ile aynı doęrultuda olan Memetik Algoritmalar ile zm retiliecektir. Bilimsel yazında memetik algoritma ve st-sezgisel algoritma trevlerinden yola ıkılarak kendi yapılandırın oklu memetik (Self-Adaptive Multimeme) algoritması geliřtirilerek bu sistem SP ve DP problemleri zerinde kullanılacaktır. Ayrıca bu algoritma kullanılarak HyFlex sistemi zerindeki 6 problem zerinde test iřlemleri gerekleřtirilecektir.

Basit sezgisel algoritmaların gerekleřtirilmesi basit ve hızlıdır, ancak, retilen sonular ok bařarılı deęildir. Birok karar verme senaryosunda farklı faktrler hesaplanmıřtır. Bu tezdeki alıřmada memetik algoritmalar ile st-sezgisel algoritmaların kombinasyonunu yapılarak, sezgiselleri farklı bir seim yntemi ile seerek zme daha kısa srede ulařılması hedeflenmiřtir.

Bu alıřma yeni sınav ve ders izelgeleme rnekleri oluřturmanın yanısıra yeni bir algoritma ile basit, kolay kullanılabilir bir yazılım arayz tasarlamayı amalamıřtır. Kısıt ve parametreler kolaylıkla belirlenebilecek olup oluřturulan sonu bařka bir izelgeleme ile de karřılařtırılabilecek yapıda sistem tasarlanacaktır. nerilen algoritma 2. Uluslararası izelgeleme Yarıřması (Second International Timetabling Competition, ITC2007) rnekleri ile de test edilecektir.

#### **alıřmanın bařarıya ulařması iin yapılacaklar:**

- 1) st-sezgisel yntemler arařtırılması ve geliřtirilmesi,
- 2) Sınav izelgeleme kısıtlarının oluřturulması,
- 3) Sınav izelgeleme modelinin oluřturulması,
- 4) Sınav izelgeleme iin st-sezgisel yntemlerin yazılıma entegre edilmesi ve sonuların yazılım zerinde karřılařtırılması,
- 5) Ders izelgeleme kısıtlarının oluřturulması,
- 6) Ders izelgeleme modelinin oluřturulması,
- 7) Ders izelgeleme iin st-sezgisel yntemlerin yazılıma entegre edilmesi ve sonuların yazılım zerinde karřılařtırılması,

- 8) ITC2007 örnekleri için sezgisel yöntemlerin uygulanması,
- 9) Yeni ders çizelgeleme ve sınav çizelgeleme veri örnekleri oluşturmak ve bu örneklerle problem çözümü yapmak
- 10) Üniversitelerin dersliklerinin daha verimli kullanılması için karar destek sistemi tasarımı önerisi,
- 11) Üniversitelerin doluluk oranlarının ölçülerek yeni mekân taleplerinin değerlendirilmesi olarak belirlenmiştir.

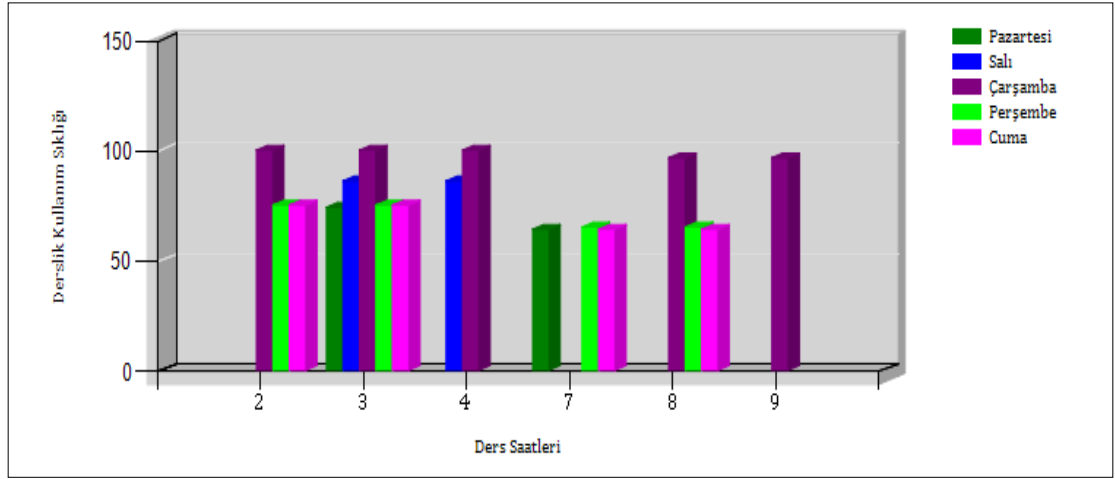
#### **Tezin katkıları ve amaçları;**

- 1) Memetik Algoritmalar tabanlı 7 veya daha fazla problem için çalışabilen üst-sezgisel framework yazılımı çalışması yapmak. Yapılan çalışmayı diğer algoritmalar ile karşılaştırmak,
- 2) Önerilen algoritma ile standart sınav ve ders çizelgeleme modeli oluşturarak problemi çözmek,
- 3) Kurumlar için başka bir problem olan mekân planlama için kısıtlar oluşturulup problemi çözmek,
- 4) ITC2007 veri örnekleri üzerinde geliştirilen algoritmayı çalıştırıp diğer algoritmalar ile karşılaştırmak,
- 5) Üniversitelerin veri örnekleri üzerinde üst-sezgisel algoritma geliştirmek,
- 6) Değişik parametreler kullanarak istatistiksel analizler yapmak,
- 7) Üniversitelerin mekân kullanım oranlarını artırmayı sağlamak,
- 8) Geliştirilen teknikleri benzer probleme sahip diğer kuruluşların kullanabilmesini sağlamak,
- 9) Yapay zeka alanında farklı çalışmalar yaparak Memetik algoritmaların gücü ile üst-sezgisel algoritmaların esnekliğini birleştirerek bütün çizelgeleme problemlerini çözmeye çalışmak,
- 10) Yazılan sistem ile iş parçacığı (thread) yapısı kullanılarak toplu (batch processing) işleme yöntemi ile birden fazla problem örneğini verilen sürede toplu bir şekilde çözmek. Bu sayede bilgisayarda harcanmış olan zaman ve işlemin dengeli bir şekilde örneklerle dağıtılması sağlanmış olacaktır. Ayrıca aynı probleme ait bazı örnekler için önceki örneklerden

üst-sezgisel öğrenme sağlanarak sonraki örnekler daha iyi sonuçlar üretebilecektir.

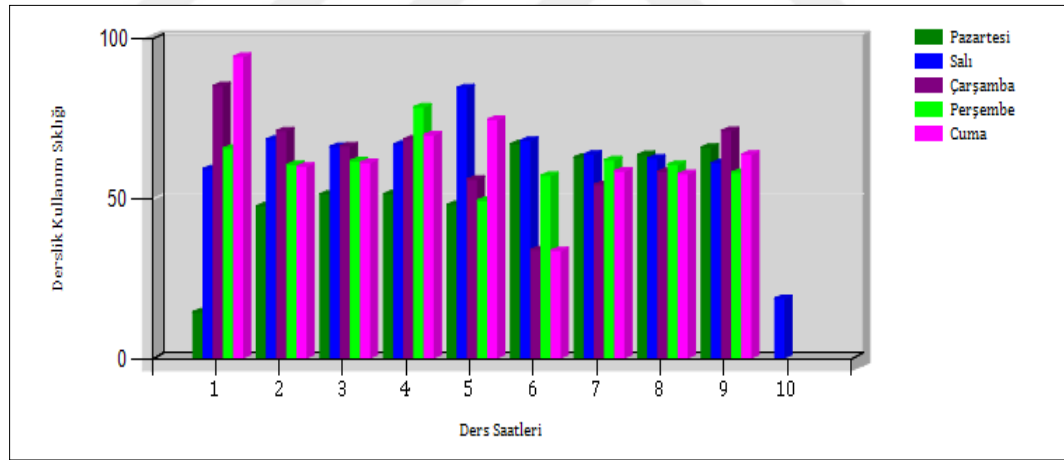
Yükseköğretim kurumlarında sürekli kontenjan artışı ve yeni üniversitelerin kuruluşu ile derslik ihtiyaçlarında da artış görülmüştür. Yükseköğretimin diğer eğitim kademelerine nazaran artan bireysel ve toplumsal getirisi ile 12 yıllık eğitimin zorunlu hale gelmesi, önümüzdeki yıllarda da yükseköğretim derslik talebini artıracaktır. Yeni derslik taleplerinde üniversiteler bünyesindeki mevcut kapalı alanların kullanım etkinliğinin dikkate alınması gerekmektedir. Üniversitelerin ders çizelgeleme planlaması yaparken dersliklerin verimli kullanılıp kullanılmadığını dikkate almadıkları görülmüştür. Bu çalışma ile üniversitelerin derslik verimlilik analizlerinin tespit edilmesi ve ders çizelgeleme problemlerinde efektif derslik yönetiminin yöntemlerinin neler olduğu uygulamasının nasıl gerçekleştirileceği tartışılacaktır.

Üniversitelerdeki zaman çizelgeleme problemleri üzerinde sürekli çalışılan ders programı çizelgeleme, sınav programı çizelgeleme, mekân etkinlik çizelgeleme vb. problemlerdir. Bir zaman çizelgeleme probleminde bireyler faaliyet ve/veya kaynaklara yanı sıra faaliyetler de kaynaklara atanabilmektedir. Öğretim elemanlarının derslere veya gözetmenlerin sınavlara atanması bireylerin faaliyetlere atanmasına; gözetmen ya da öğretim elemanlarının zaman dilimleri ya da dersliklere atanması ise bireylerin kaynaklara atanmasına örnek olarak verilebilir. Öte yandan, derslerin dersliklere ve/veya zaman dilimlerine atanması ise faaliyetlerin kaynaklara atanmasıdır. Ders programı çizelgeleme yaparken üniversitelerin karşılaştığı sorunlardan biri atanacak dersliğin kapasitesine karar vermektir. Bu sorun yakın veya uzun dönemde derslik ihtiyacı planlaması yaparken dikkate alınması gerek bir problemdir. Bu planlama üniversiteler için yüksek maliyetler gerektirdiği için derslik ihtiyaçlarının minimize edilmesi çok önemlidir. Ders çizelgeleme problemini çözerken hem uygun çözüm bulunup hem de dersliklerin kullanım verimliliği artırılabilir. Şekil 1.1’de örnek bir dersliğin ders saatlerine göre kullanım oranları verilmiştir. Bu oranlara göre 2.ve 3. saatler en yoğun kullanılan saatlerdir.



Şekil 1.1. Derslik Kullanım Oranı

Şekil 1.2’de bir binadaki tüm dersliklerin kullanım oranı gösterilmektedir. Çizelge 1.1 ve Çizelge 1.2’de ise Pazartesi ve Salı günü örnek fakültede derslik kullanım sıklığı analizleri verilmiştir.



Şekil 1.2. Fakülte Binası Derslik Kullanım Sıklığı

Çizelge 1.1. Fakülte binası Pazartesi günü derslik kullanım analizi

|           | ZAMAN DİLİMİ | ÖĞRENCİ SAYISI(1) | KULLANILAN DERSLİK KAPASİTESİ (2) | TOPLAM KAPASİTE (3) | % DERSLİK K. ORANI(1/2) | % TOPLAM ÖĞR. KAPASİTE ORANI (1/3) | % TOPLAM KULLANILAN KAPASİTE ORANI (2/3) |
|-----------|--------------|-------------------|-----------------------------------|---------------------|-------------------------|------------------------------------|------------------------------------------|
| Pazartesi | 1-2          | 99                | 694                               | 5692                | 14.27                   | 1.74                               | 12.19                                    |
|           | 2-3          | 1060              | 2232                              | 5692                | 47.49                   | 18.62                              | 39.21                                    |
|           | 3-4          | 1402              | 2738                              | 5692                | 51.21                   | 24.63                              | 48.1                                     |
|           | 4-5          | 774               | 1510                              | 5692                | 51.26                   | 13.6                               | 26.53                                    |
|           | 5-6          | 393               | 820                               | 5692                | 47.93                   | 6.9                                | 14.41                                    |
|           | 6-7          | 308               | 460                               | 5692                | 66.96                   | 5.41                               | 8.08                                     |
|           | 7-8          | 1352              | 2154                              | 5692                | 62.77                   | 23.75                              | 37.84                                    |
|           | 8-9          | 1923              | 3024                              | 5692                | 63.59                   | 33.78                              | 53.13                                    |
|           | 9-10         | 903               | 1370                              | 5692                | 65.91                   | 15.86                              | 24.07                                    |

Çizelge 1.2. Fakülte binası Salı günü derslik kullanım analizi

|      | ZAMAN DİLİMİ | ÖĞRENCİ SAYISI(1) | KULLANILAN DERSLİK KAPASİTESİ (2) | TOPLAM KAPASİTE (3) | % DERSLİK K. ORANI(1/2) | % TOPLAM ÖĞR. KAPASİTE ORANI (1/3) | % TOPLAM KULLANILAN KAPASİTE ORANI (2/3) |
|------|--------------|-------------------|-----------------------------------|---------------------|-------------------------|------------------------------------|------------------------------------------|
| Salı | 1-2          | 71                | 120                               | 5692                | 59.17                   | 1.25                               | 2.11                                     |
|      | 2-3          | 2397              | 3502                              | 5692                | 68.45                   | 42.11                              | 61.52                                    |
|      | 3-4          | 2989              | 4522                              | 5692                | 66.1                    | 52.51                              | 79.44                                    |
|      | 4-5          | 916               | 1370                              | 5692                | 66.86                   | 16.09                              | 24.07                                    |
|      | 5-6          | 279               | 330                               | 5692                | 84.55                   | 4.9                                | 5.8                                      |
|      | 6-7          | 279               | 410                               | 5692                | 68.05                   | 4.9                                | 7.2                                      |
|      | 7-8          | 2174              | 3410                              | 5692                | 63.75                   | 38.19                              | 59.91                                    |
|      | 8-9          | 2818              | 4510                              | 5692                | 62.48                   | 49.51                              | 79.23                                    |
|      | 9-10         | 805               | 1320                              | 5692                | 60.98                   | 14.14                              | 23.19                                    |
|      | 10-11        | 13                | 70                                | 5692                | 18.57                   | 0.23                               | 1.23                                     |

Üniversitelerdeki çizelgeleme problemleri birçok kısıtlar nedeniyle oldukça karmaşık olmasına rağmen bu problemleri çözmek için çeşitli yaklaşımlar geliştirilmiştir. Bir ders çizelgeleme problemi sert ve esnek kısıtları sağlayarak dersleri verilen ders saatlerine atamaktan oluşmaktadır. Bir çizelgeleme problemi sert ve esnek kısıt olarak ayrılabilir kısıtları sağlamalıdır. Sert kısıtlar bir olayın aynı zaman diliminde birden fazla kaynağa atanmaması gibi

hiçbir şekilde ihlal edilemez, esnek kısıtlar ise ihlal edilebilir fakat birçok yaklaşım bu ihlallerin sayısını azaltarak çözümün kalitesini artırmayı amaçlamaktadır.

Üniversitelerin her yıl öğrenci sayısının artması sebebiyle eğitim mekânlarını ve diğer mekânları daha etkin ve verimli bir şekilde kullanma ihtiyacı doğmuştur. Bu yüzden bu çalışmada üniversite çizelgeleme problemlerini çözerken derslik kullanım oranları da dikkate alınacaktır.



## **2. KAYNAK ÖZETLERİ**

### **Meta-sezgiseller (Meta-heuristics)**

Birçok optimizasyon problemi çok fazla araştırma alanına sahip olduğu için problem ile ilgili bütün durumları araştırmak oldukça zordur. Sezgiseller ise bu alanda mümkün olan en iyi çözümü ararlar. Esasında bir meta-sezgisel, belirli bir çözümün mevcut durumunu değiştirmek için var olan sezgisel yöntemler içerisinde arama yaparak çözümün bulunmasında rehberlik etmek üzere tasarlanmış yüksek seviyeli bir stratejidir. Meta-sezgisel terimi ilk Glover vd., (1986) tarafından kullanılmıştır. Sörensen ve Glover, (2013) tarafından tanımı genişletilmiş olup sezgisel optimizasyon algoritmalarını geliştirmek için bir dizi strateji sağlayan yüksek seviyeli probleminden bağımsız algoritmik bir çerçeve olarak ifade edilmiştir.

### **Yerel arama (Local search)**

Yerel arama yöntemleri verilen bir problem için aday çözümler arasındaki yerellik kavramına dayanan basit bir meta-sezgisel algoritmadır. Tipik bir yerel arama algoritması komşu çözümler içerisindeki bir çözümün mevcut çözüm ile değiştirilmesidir. Yerel bir arama yönteminde bir çözümünden daha iyi bir çözüme geçilmesi tepe tırmanma (Hill Climbing) olarak adlandırılır (Hoos ve Stützle, 2005).

### **Evrimsel algoritmalar (Evolutionary algorithms)**

Evrimsel algoritmalar doğal evrimsel süreçleri temel alan arama teknikleridir. Genetik Algoritmalar en çok bilinen evrimsel algoritmalarlardır Goldberg, (1989). Bir genetik algoritma çözüm kümesindeki adaylara mutasyon ve çaprazlama operatörleri uygulayarak iteratif bir şekilde çözümü günceller. Basit bir genetik algoritma aşağıda verilmiştir.

#### Algoritma 1: Basit Genetik Algoritması

- 1: Başlangıç çözüm popülasyonunu üret
- 2: Tekrarla
- 3: Popülasyondaki her bir çözüm için uygunluk değeri hesapla
- 4: Uygun bir çözüm seçimi yap
- 5: Belirli bir olasılıklar çaprazlama uygula
- 6: Belirli bir olasılıkla mutasyon uygula
- 7: Bitirme kriteri ile karşılaşınca kadar devam et
- 8: En iyi çözümü getir

Memetik algoritmalar yerel arama teknikleri ile genetik algoritmaların birleştirilmesi sonucu elde edilen algoritmalarlardır. Basit bir memetik algortmada genetik algortmadaki mutasyon ve çaprazlama işlemleri sonrası bir yerel arama aşaması eklenir. Memetik terimi ilk Moscato(1989) tarafından tanımlanmıştır.

#### **Tabu arama (Tabu search)**

Tabu Arama algoritması yerel arama yaparken yerel optimumdan kaçmak için kullanılan bir meta-sezgisel yöntemdir (Glover, 1989). Tabu arama algortmayı daha önce ziyaret etmediği arama alanlarına yönlendirmek için bir bellek kullanılır. Bu bellek arama alanındaki yakın zamanda ziyaret edilen noktaları kaydeder ve bu hareketleri tabu listesine yerleştirir. Algoritma bir sonraki çözüme giderken tabu listesindeki noktalara gitmesine izin verilmez. Bu yöntem algortmanın yerel optimum çözümlerinden genel optimum çözümlerine yönlendirmesini sağlar. Tabu arama algortmasının yapısı aşağıdaki gibidir:

#### Algoritma: 2 Tabu Arama Algoritması

- 1: İlk aday çözümünü seç  $S$
- 2: Tekrar et
- 3: Tabu listesinde olmayan  $S'$ 'ye bitişik en iyi komşu çözümü  $S'$  bul
- 4:  $S'$  çözümünü tabu listesine ekle
- 5: Eğer  $S'$  çözümü  $S$  çözümünden daha iyi ise
- 6:  $S \leftarrow S'$
- 7: Bitirme kriteri ile karşılaşınca kadar devam et
- 8:  $S$  çözümünü getir

## Tavlama benzetimi (Simulated annealing)

Tavlama benzetimi kombinatorial optimizasyon problemleri için kullanılan bir meta-sezgisel tekniktir (Kirkpatrick vd.,1983). Tavlama Benzetimi sıcaklık parametresine göre belirli bir sistemde yapılan değişikliğin kabul edilip edilmeyeceğine karar verir. Sıcaklık parametresi başlangıçta yüksek bir değere ayarlanır ve arama işlemi boyunca azaltılır. Sıcaklığın yüksek olduğu bir aramanın başlangıcında daha büyük olasılıkla iyileştirilmemiş hareketler kabul edilir. Sıcaklık düştükçe, sistem iyileşmeyen hareketleri çok daha düşük bir oranda kabul edecektir. Çözüm kalitesinde bir bozulmanın meydana gelmesi durumunda, sıcaklık parametresinin geçerli değerine ve çözüm kalitesindeki değişiklik seviyesine göre belirli olasılıkla bir hareket kabul edilir. Bu olasılığı hesaplamak için literatürde bir takım öneriler vardır. p aşağıdaki gibi tanımlanır:

$$p = \frac{1}{1+e^{-\Delta/T}} \quad (2.1)$$

Tavlama Benzetimi algoritmasının temel yapısı aşağıdaki gibidir:

Algoritma 3: Tavlama Benzetimi Algoritması

- 1: Başlangıç bir aday çözümü S seç, T'yi başlat
- 2: Tekrar et
- 3: S ile komşu bir S' çözümünü bul
- 4:  $\Delta \leftarrow S'$  ve S arasındaki kalite farkını bul
- 5: Eğer S' çözümü S çözümünden daha iyi ise
- 6:  $S \leftarrow S'$
- 7: değilse
- 8:  $r \leftarrow [0,1]$  arasında rasgele sayı üret
- 9: Eğer  $p < r$  ise
- 10:  $S \leftarrow S'$
- 11: T'yi azalt
- 12: Bitirme kriteri ile karşılaşıncaya kadar devam et
- 13: S çözümünü getir

## Büyük tufan (Great deluge)

Büyük tufan selden kaçan bir kişinin tepeye tırmanmasından esinlenerek Dueck (1993) tarafından geliştirilen genel amaçlı bir tepe tırmanma algoritmasıdır. Büyük Tufan algoritmasının temel yapısı aşağıdaki gibidir:

Algoritma 4: Büyük Tufan Algoritması

- 1: S aday çözüm, B yağmur hızı ve  $\tau$  su seviyesi
- 2:  $S_{best}$  en iyi çözüm
- 3:  $S \leftarrow$  Rasgele başlangıç çözümü üret
- 4:  $S_{best} \leftarrow S$ ;
- 5:  $f_0 \leftarrow \text{Evaluate}(S)$ ; // başlangıç uygunluk fonksiyonunu hesapla
- 6:  $\tau \leftarrow f_0$ ; // Başlangıç seviyesi
- 7: Tekrar et
- 8:  $\text{Generate}(S')$ ; // S'ye komşu S' çözümünü üret
- 9:  $S_{best} \leftarrow \text{maintainBestSolution}(S')$ ;
- 10: Eğer  $\text{Evaluate}(S')$  çözümü  $\text{Evaluate}(S)$  çözümünden daha iyi ise
- 11  $S \leftarrow S'$ ;
- 12 bitir
- 13 değilse
- 14 Eğer  $\text{Evaluate}(S')$  sonucu  $\tau$ 'den daha iyi ise
- 15  $S \leftarrow S'$ ;
- 16 bitir
- 17 bitir
- 18  $\text{Update}(\tau)$ ; // Su seviyesini güncelle  $\tau = \tau \pm B$
- 19 Bitirme kriteri ile karşılaşıncaya kadar devam et
- 20  $S_{best}$  çözümünü getir

## Geç kabul stratejisi (Late acceptance strategy)

Geç kabul stratejisi Burke ve Bykov (2008b) tarafından geliştirilen tepe tırmanma algoritmasıdır. Bu algoritma, daha önce ne kadar adım atıldığını hafızada tutarak bir aday çözüm üretip hafızadaki çözüm ile karşılaştırarak daha iyi bir çözüm elde etmeye çalışır. Bu algoritma basit ağgözlü tepe tırmanma

(simple greedy hill climbing) algoritmasının genişletilmiştir. Tepe tırmanma algoritmasında daha iyi bir çözüm bulunduğunda hemen kabul edilmesine rağmen geç kabul stratejisi algoritmasında ise daha önceki çözümlerden daha iyi bir çözümü kabul eder. Kabul edilen çözüm hafızadaki daha önceki çözümler ile karşılaştırılarak yer değiştirilir.  $L$  ziyaret edilen çözümlerin uygunluk değerlerinin tutulduğu listedir. Sistem bellekte tutulan uygunluk değerlerinden daha iyi bir değer bulursa yeni çözüm kabul edilecektir.

#### Algoritma 5: Geç Kabul Stratejisi Algoritması

```
1:  $S$  aday çözüm
2:  $S_{best}$  en iyi çözüm
3:  $S \leftarrow \text{CreateInitialSolution}()$ ; // rasgele çözüm üret
4:  $S_{best} \leftarrow S$ ;
5:  $f_0 \leftarrow \text{Evaluate}(S)$ ; // başlangıç uygunluk fonksiyonunu hesapla
6: for  $i \leftarrow 0, L - 1$  do
7:  $f(i) \leftarrow f_0$ ;
8: bitir
9:  $i \leftarrow 0$ ;
10: tekrarla
11:  $S' \leftarrow \text{Generate}(S)$ ; //  $S$ 'ye komşu  $S'$  çözümünü üret
12:  $f' \leftarrow \text{Evaluate}(S')$ ; // uygunluk fonksiyonunu hesapla
13:  $S_{best} \leftarrow \text{maintainBestSolution}(S')$ ;
14:  $c \leftarrow i \bmod L$ ;
15: Eğer  $f', f(c)$  'den daha iyi ise
16:  $S \leftarrow S'$ ;
17: Bitir
18:  $f(c) \leftarrow \text{Evaluate}(S)$ ; // uygunluk değerini listeye dahil et
19:  $i \leftarrow i + 1$ ;
20: Bitirme kriteri ile karşılaşıncaya kadar devam et
21:  $S_{best}$  çözümünü getir
```

## Üst-Sezgiseller (Hyper-Heuristics)

Son yıllarda üst-sezgisel yöntemler çok çeşitli problemler için otomatik sezgisel üretme yeteneğini göstermişlerdir. Çünkü üst-sezgiseller otomatik sezgisel üreterek, ortam değiştiği anda hızlı bir şekilde değişerek uygun çözüm bulmaktadır. Burke vd., (2013) üst-sezgiseller için bir literatür taraması yapmışlar ve üst-sezgisellerin kökenini ve tarihçesini araştırmışlardır. Bu çalışmaya göre yazarlar meta-sezgisel ve makine öğrenmesi alanlarındaki çalışmaları ve gelecekteki araştırmaları tartışmışlardır. Ross (2014) makalesinde üst-sezgisel yaklaşımlar için uygulama alanlarının belirlenmesi ve ilgili araştırma konularını incelemiştir. Branke vd., (2016) üretim çizelgeleme optimizasyonu problemlerini yapıcı sezgisellerin tasarımına vurgu yaparak incelemiştir. Pillay (2016) üniversite sınav çizelgeleme, ders zaman çizelgeleme ve okul zaman çizelgeleme problemleri için üst-sezgiselleri araştırmıştır. Yazar zor eğitimsel zaman çizelgeleme problemlerini çözmek için yeniden kullanılabilir teknolojiler üretme konusunu vurgulamıştır.

## Üst-Sezgisellerin kökeni ve önceki yaklaşımlar

Üst-sezgisel fikri yeni değildir. Üst-sezgiseller 1960'lara kadar görülebilir ve Yöneylem Araştırması, Bilgisayar Bilimleri ve Yapay Zeka alanlarında bulunabilir. Burada 2000 yılından önceki çalışmalar bu fikrin kökeni olarak araştırılacaktır. Önceki yaklaşımlar dört şekilde ele alınacaktır.

**Otomatik sezgisel sıralaması:** Fisher ve Thompson (1961, 1963) ve Crowston vd. (1963), üretim zamanlamada birleşik zamanlama kuralları (aynı zamanda dağıtım kuralları veya öncelikli olarak da bilinir) herhangi bir kuralı kullanarak daha üstün olacak hipotezini savunmuşlardır. Bu öncü çalışma üst-sezgisel yöntemlerin içerisindeki araştırmaların temelini oluşturduğu için oldukça önemlidir. Bu önerme "olasılıklı öğrenme" kullanılarak zamanlama kurallarının birleştirme yöntemi içindir. Bu çalışmadan çıkarılacak temel sonuç: "(1) rasgele seçilen kombine zamanlama kuralları ayrı ayrı seçilenlerin her birisinden daha iyidir; (2) öğrenme mümkündür" (Fisher ve Thompson, 1963).

1990’larda bu fikir yenilenmiştir: Storer vd. (1992, 1995) atölye tipi çizelgeleme (job-shop scheduling) arama problemini probleme-özel sezgisellerle iyi bir şekilde kombine eden tasarım problemini açıklamışlar ve sezgisel uzayındaki komşuluğu tanımlamışlardır. Bu yaklaşım (Fang vd., 1993, 1994) ‘da tartışılmıştır, açık atölye tipi çizelgeleme (open-shop scheduling) kavramındaki sezgisel seçim uzayında arama için genetik algoritma kullanılmıştır. Sonra Hart ve Ross (1998a, b) dinamik atölye tipi çizelgeleme problemleri için değişik bir varyasyonunu uygulamışlardır. Hart vd. (1998a) gerçek-dünya çizelgeleme problemleri ve dağıtım problemlerini çözmek için genetik algoritma tabanlı bir yaklaşım kullanılmıştır. Bu yaklaşımda iki genetik algoritma ile birleştirilerek sezgisel seçimi geliştirilmiştir, bir tanesi sıralama ataması için kullanılmış, diğeri de dağıtımları çizelgelemek için kullanılmıştır. Bu yaklaşımların tümünde çevrimiçi (online)’lik vardır. Bu, belirli bir problem örneğini çözmek için sezgisel iyi dizileri bulmaya yöneliktir. Buna karşılık, Drechsler ve Becker (1995); Drechsler vd. (1996), tarafından yapılan çalışmada, elektronik çip tasarımı evrimsel algoritma kullanılan (önceki örneklerde bir dizi) bir öğrenme aşamasından sonra belirli bir sorunu örneklerine uygulanabilir başarılı sezgisel öğrenmek için kullanılmıştır.

**Sistemlerin otomatik planlanması:** Üst-sezgisel kavramını etkileyen diğer çalışmalar ise Yapay Zeka alanından gelmiştir. Özellikle, otomatik planlama sistemleri çalışmaları ve kontrollü bilgi öğrenme problemleri çalışmaları mevcuttur. (Gratch ve Chien, 1996; Gratch vd., 1993) ‘de COMPOSER olarak adlandırılan sistem uydu iletişim zamanlama kontrolü için kullanılmıştır. Bu sistem olası kontrol stratejileri alanında tepe tırmanma (hill-climbing) arama ile karakterize olabilir. Bu yaklaşım çevrimdışı (offline)’dır, Burada çok büyük kaynaklı temsili öğrenme problemleri çeşitli kontrol stratejileri için beklenen faydanın yeterince tahmin edilmesi için gerekmektedirler. Bu yöntem alana-özgü bilgilere uygulanmıştır ve modern üst-sezgisel yaklaşımlarından farklıdır. Ek olarak, bir planlama problemi optimizasyon problemlerinin formülasyonundan farklıdır: planlamadaki amaç uygun hedefin başlangıç durumunu değiştiren eylemlerle bulunmasıdır.

**Evrimsel algoritmalar da otomatik parametre kontrolü:** Evrimsel algoritmalar da parametrelerin otomatik ayarlanması için bazı önceki yaklaşımlar üst-sezgisellerin öncüsü olarak düşünülebilirler. Çalışma sırasında parametrelerin otomatik değişmesi için birkaç mekanizma “bilgili”, adaptif evrimsel algoritmaların erken dönemlerinde Eiben vd. (1999) tarafından önerilmiştir. Evrimsel algoritmaların kullanılmasındaki diğer bir fikir de bir evrimsel algoritmanın ayarlanmasıdır. Bu iki evrimsel algoritma kullanılarak yapılabilir: birisi problemi çözmek için, diğeri ise (meta-evrimsel algoritma olarak ta adlandırılır) ilkin i ayarlamak içindir (Freisleben ve Härtfelder, 1993; Grefenstette, 1986). Aynı zamanda tek bir evrimsel algoritma kullanılarak verilen bir problemi çözerken kendi kendine ayarlama yapılabilir. Bu kavram ‘self-adaptasyon’, olarak ilk kez evrimsel stratejilerde çeşitli mutasyon parametreleri için kullanılmıştır (Rechenberg, 1973). Evrimsel algoritmalar da self-adaptasyon çalışma sırasında bazı parametrelerin değişmesi anlamına gelir: bu parametreler çözümle birlikte yeniden hesaplanır. Bu yaklaşımlar mümkün algoritma konfigürasyonları uzayında arama fikri ile bağlantılıdır ve bu yüzden üst-sezgisellerle bağlantılıdır. Evrimsel algoritmalar da parametre kontrolü yaklaşımlarını incelemek ve sınıflandırmaları incelemek için (Eiben vd., 1999, 2007) çalışmaları referans olarak alınabilir.

**Sezgisel yöntemlerin otomatik öğrenmesi:** Önceki yaklaşımlar da otomatik sezgisel üretmek için bilgisayar programları kısıt sağlama problemlerinde (Minton, 1996) bulunabilir. Özellikle, (Minton, 1996) çalışması Minimum Maximal Matching Problem’i için tekrar kullanılabilir sezgisel üretmek amacıyla geliştirilen bir sistemi sunan öncü bir çalışmadır. Bu sistem genel algoritma şablonlarını belirli algoritmanın öğelerine göre değiştirir. Bu genel fikir kısıt tabanlı algoritma versiyonlarında problem-ö zğü otomatik sentez yapmak içindir. Kullanıcı eğitim örneklerini sağlar, işlemlerin yapılandırma sürecinde sistem öğrenir ve böylece eğitim örnekleri tarafından temsil edilen belirli dağıtım örneklerine adapte olur. Bu çalışma literatürde tektir, üç NASA programcısı tarafından otomatik yaklaşımlar a karşı üretilmiştir ve insan yapımı programlara karşı üstünlük sağlamıştır. Diğ er ilginç bir öğrenme yaklaşımlı çalışma ise 1990 ortalarında “Teacher” (Wah vd., 1995; Wah ve Ieumwananonthachai, 1999) (an

acronym for TEchniques for the Automated Creation of HEuRistics) olarak adlandırılan çalışmadır, bu sistem problem çözümünde sezgiselleri genellemek ve öğrenen bir sistem olarak tasarlanmıştır. Burada amaç uygulamada çok az olan veya olmayan bilgileri var olanlarla karşılaştırarak sezgisel yöntemler geliştirmek ve bulmaktır. Teacher sistemi genetik tabanlı makine öğrenmesi yaklaşımını kullanır ve birkaç alana başarıyla uygulanmıştır örneğin: iş haritası (process mapping), network istasyonlarındaki yük dengeleme (load balancing) , devre yerleşimi (circuit placement) ve rotalama ve test (routing ve testing).

### **Üst-Sezgisellerin ders çizelgeleme problemine uygulanması**

Ders çizelgeleme problemi birçok sezgisel veya meta-sezgisel kullanılarak çözülmüştür. Ancak meta-sezgiseller probleme bağlı olarak çalışırlar. Son yıllarda, üst-sezgiseller meta-sezgisellerin daha genel kullanılabilmesi için geliştirilen yeni arama yöntemleridir. Üst-sezgisellerin amacı arama yöntemini problem-bağımsız hale getirmektir. Üst-sezgiseller üniversite ders çizelgeleme problemi içinde başarıyla uygulanmaktadır. Burke vd. (Burke, 2003b) basit bir dizi yerel arama sezgisellerinin uygulamasını yönlendirmek için tabu listesini kullanan bir üst-sezgisel önermişlerdir. Burke vd. (Burke, 2007a) çizelgelemeyi oluşturmadan önce tabu arama işlemleri sırasında, 6 çizge boyama sezgiselini değiştirmek için çizge tabanlı bir üst-sezgisel kullanmışlardır. Buradaki yaklaşımın amacı kullanılan sezgiselleri en iyi sıraya dizmektir. Bai vd. (Bai, 2007) ders çizelgeleme problemini çözmek için benzetimli tavlama (simulated annealing) üst-sezgiselini kullanmışlardır. Bu yöntemde stokastik sıralama mekanizmasına dayalı düşük seviyeli sezgiselleri seçer.

### **Üst-Sezgisellerin sınav çizelgeleme problemine uygulanması**

Sınav çizelgeleme problemleri NP-zor problemler sınıfına girmektedir. Bugün sınav çizelgeleme problemleri yıllık olarak uygulanan, çözümü zor olan ve fazla zaman harcanması gereken problemlerdir. Bu tez çalışmasında üst-sezgisel metodlar kullanılarak sınav çizelgeleme problemi çözülmesi amaçlanmaktadır. Bir çok araştırmacı bu problem için en iyi çözümü bulan algoritmaları

geliştirmeyi hedeflemişlerdir (Qu vd., 2009a, b). Bu problem verilen bir süre dizisine bir dizi sınavların atanmasını gerektirir. Problem sert ve esnek kısıtlar olarak iki farklı kısıt türü içermektedir. Sert kısıtlar mutlaka sağlanmalıdırlar, örneğin hiçbir öğrenci aynı süre içerisinde bir sınavdan fazla sınava atanamaz. Esnek kısıtlar ise minimize edilmeye çalışılmaktadır.

Sınav çizelgeleme alanındaki araştırmalar 1996 da Carter (Carter vd., 1996) tarafından başlatılmış ve o zamandan bugüne sayısız yöntemler bu probleme uygulanmıştır. Carter tarafından 13 tane örnek problem önerilmiştir ve bu örnekler üzerinde değişik yöntemler karşılaştırılmıştır. Daha sonraki örnekler ise ikinci uluslararası çizelgeleme yarışmasında (Second International Timetabling Competition-ITC2007) kullanılmıştır (McCollum vd., 2007, 2010). Bu örnekler Carter'ın önerdiği örneklerden daha fazla kısıt içermektedir. Bu tez çalışmasında ITC2007 örnekleri kullanılmıştır. Genellikle üst-sezgisel algoritmaları iki yaklaşımdan bir tanesini kullanır. Birinci yaklaşım var olan düşük seviyeli sezgiselleri kullanır, ikinci yaklaşım ise düşük seviyeli sezgiselleri bazı yöntemlerle kombine ederek kullanır.

Qu vd. (2005) iki veya daha fazla çizge boyama sezgiselleri içeren kombine sezgiselleri kullanarak bir değişken komşuluklu arama (variable neighbourhood search) algoritmasını uygulamışlardır. Herbir sezgisel kombinasyon VNS kullanılarak üretilmiştir. Önceki çalışmalara örnek olarak Burke vd. (Burke vd., 2005 a, b) tarafından düşük seviyeli sezgiseller kombine edilerek bir Tabu arama algoritması uygulanmıştır. Bu algoritma Carter örnekleri üzerinde test edilmiştir. Bu çalışma daha sonra Burke vd. ve Qu vd. (Burke vd., 2007a, Qu vd., 2009a) tarafından genişletilmiştir. Bu çalışmada başlangıç sürecinde daha iyi çözüm üretmek için düşük seviyeli sezgiseller kullanılmıştır. Analiz sonuçları çizelgeleme üretim başlangıç sürecinde hibrid sezgisellerin kullanılması ile daha iyi sonuçlar elde edildiğine göstermiştir. Asmuni vd. (2005) üç düşük seviyeli sezgisel ile bulanık mantık yaklaşımını kullanarak problemi çözmüştür. Sınavlar atanırken bulanık değerlere göre sıralanmıştır. Daha sonraki çalışmalarında (Asmuni vd. 2007, 2009) ise daha fazla düşük seviyeli sezgiselin kombine edilmesi ile problem çözülmüştür. Pillay vd. (2007, 2008) sezgiselleri

kombine etmek için bir genetik programlama algoritması geliştirmişlerdir. Başlangıç çözümünde sezgisel kombinasyonlar oluşturulmuş ve mutasyon ve çaprazlama operatörleri kullanılarak başarılı bir çözüm oluşturulmuştur.

### **Sınav çizelgeleme problemi**

Sınav çizelgeleme üzerindeki araştırmalar Carter (1986) ile başlamıştır. Bu yaklaşım temelde çizge boyama modelini kullanmaktadır ve belirli bir kurum için olan spesifik problemi çözmek için yapıcı sezgisel ile çalışılmıştır.

Carter ve Laporte (1996) bu konuda geniş bir bilimsel yazın çalışması yapmışlardır ve algoritmaları sınıflamışlardır. Yöntemler 4 grupta incelenmiştir. Bu yöntemler cluster yöntemi, sıralı (sequential) yöntemler, genelleştirilmiş arama yöntemleri (örn, meta-sezgiseller) ve kısıt tabanlı yaklaşımlardır. Bu dönemde basit kısıtlarla sınav çizelgeleme probleminin değişik versiyonları birçok algoritma tarafında çözülmüştür. Sınav çizelgeleme problemi üzerindeki çalışmaların ilgi görmesi için ise yazarlar test örnekleri yayınlamışlardır (Carter vd., 1996). Bu örnekler çok fazla kullanılmaktadır ve Toronto Üniversitesinin örnekleri olarak da bilinmektedir.

Burke (Burke vd., 1996) sınav çizelgeleme problemleri için çeşitli tekniklerin taramasını yapmışlar ve tüm dünyadaki araştırmacılardan örnekler sunmuşlardır.

Schaerf (1999) eğitim ile ilgili çizelgelemeleri sınav ve ders çizelgeleme olarak ikiye ayırmışlardır. Bu çalışma basit bir arama algoritmasının matematiksel tanımını, optimizasyon problemlerini, problem çeşitlerini ve literatürdeki yaklaşımların çözümlerini sunmuştur. Bu basit bir arama algoritması uygun çözüm bulunur bulunmaz aramayı durduracağını anlamalıdır. Bir optimizasyon probleminin ise esnek kısıtları gerçekleştirmesi gerektiği bu çalışmada tartışılmıştır.

Burke vd. (2002, 2004) üniversite çizelgeleme problemleri üzerinde geniş bilimsel yazın taraması yapmışlar ve sınav çizelgeleme için geliştirilen yöntemleri incelemişlerdir.

Lewis (2007) sınav çizelgeleme problemini de içeren üniversite çizelgeleme problemleri için geliştirelm meta-sezgisel yöntemlerin araştırmasını gerçekleştirmiştir. Bu çalışmada, meta-sezgisel algoritmalar 3 sınıfa ayrılmıştır; tek aşamalı optimizasyon problemleri, iki aşamalı optimizasyon problemleri ve genişleyebilir algoritmalar. Çalışmada problem örnekleri üzerinde farklı algoritmaların karşılaştırması yapılmıştır.

Son zamanlardaki en kapsamlı tarama çalışması ise Qu vd. (2009a) da yapılmış olup bu çalışma birçok yeni yöntemleri özetlemektedir ve araştırmacılar 1996'dan itibaren yaptıkları sınav çizelgeleme problemi çözümlerini yayınlamaktadırlar.

### **Üniversite mekânları yönetimi**

Üniversitelerdeki çizelgeleme problemlerinin çözümü, üniversite kaynaklarının en etkili ve verimli kullanılması ile doğrudan ilişkilidir. Etkili mekân yönetim sistemi sadece mekânların kullanımı ile ilgili değildir. Aynı zamanda yönetim süreçlerinde ihtiyaç duyulan kaynakların düzgün dağıtılmasına yardımcı olur. Mekân tasarımı, yönetimi stratejik ve operasyonel bir aktivitedir. Akademik ve idari alanların dengeli, eşit ve sürdürülebilir bir şekilde dağıtılması, etkili bir yönetim sistemine eşlik etmelidir. Dünyadaki birçok üniversite, yönetim sistemlerini geliştirmek ve mekân kullanım oranlarını optimize etmek için gelişmiş mekân yönetim yazılımları kullanmaktadır.

Belirli kullanıcılar ve/veya belirli kullanımlar için mekânları tahsis etme becerisi mekân yönetimi olarak tanımlanmaktadır (Newcastle University, 2007). Mekân yönetimi aynı zamanda mekânları iyileştirmek için yenilik ve değişimlerin önerilmesi, kullanımın veya atama kriterlerinin değiştirilmesi yeteneğidir. Mekân yönetimi; mekân planlama, kampüs planlama, mekân konfigürasyonu,

mekan ayırma, alan yönetimi ve alan izleme başlıklarının tümünü kapsamaktadır. Aynı zamanda stratejik ve operasyonel düzeyde kaynak yönetimi, mekân yönetiminin temel fonksiyonlarından biri olarak kabul edilmektedir (Nutt, 2002). Uygulamada; güvenlik yönetimi, afet yönetimi, alan yönetimi, bakım onarım işleri operasyonel düzeyde mekân yönetimine verilebilecek örneklerdendir (Downie, 2005).

Yükseköğretim kurumlarında, mekânların ve ekipmanların hem cari dönemde hem de gelecekte daha etkili ve etkin kullanılması mekân yönetiminin temel amaçlarından biridir. Bu yüzden, mekânın yönetilmesi süreci, kurumların en önemli ve en maliyetli kalemlerinden biridir. Kurumlar, sınırlı kaynaklarını birimler arasında dağıtarak mekân tahsisi kararlarını alırlar. Zayıf ya da hatalı mekân yönetimi, mekân kullanıcılarında olduğu kadar yönetim süreçlerinde yer alanlar üzerinde de negatif sonuçlar doğurabilir. Yükseköğretim kurumlarında eğitim alanlarının düşük kullanım oranları, hatalı tasarlanmış mekânlar, yanlış planlama, mekânların kullanım oranlarının ölçülmemesi zayıf mekân yönetimine örnek gösterilebilir (TEFMA, 2009).

### **Meta-Üst sezgiseller ve hibrid meta/üst-sezgiseller**

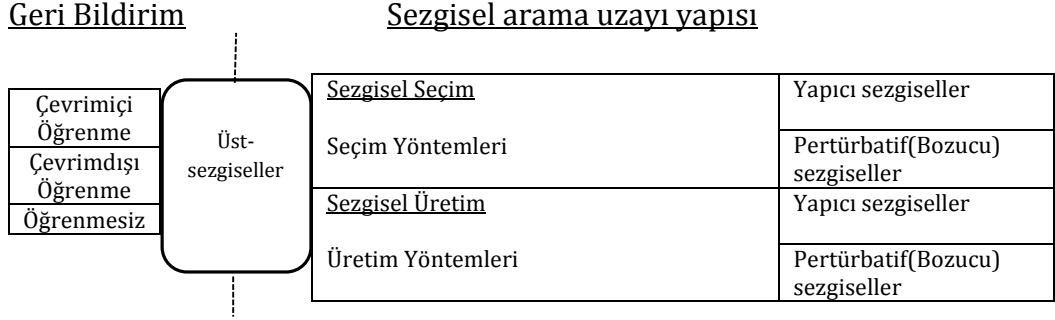
Hem meta-sezgisel hemde üst-sezgisellerin birleştirildiği çözüm modelleridir. Vazquez-Rodriguez vd. (2007a, b) akış tipi çizelgeleme problemleri için bir meta-üst sezgisel algoritması önermiştir. Algoritmayı probleim 8 tipi için uygulamış olup üst-sezgisel ve meta-sezgisellerle kombine ettiği meta-üst sezgisel algoritmayı kullanmıştır. Meta-üst sezgisel algoritma ile daha iyi sonuçlar elde etmiştir.

Grobler vd. (2015) sezgisel arama uzayı çeşitliliğini yönetmek için tabu arama tabanlı bir meta-üst sezgisel yöntem geliştirmişlerdir. Deneylerinde heterojen bir meta-üst sezgisel metodunu 4 yaygın meta-sezgisel algoritma ile karşılaştırmışlar ve daha iyi sonuçlar elde etmişlerdir. Hibrid üst-sezgiseller, bir otomatik algoritma seçim yaklaşımı ile bir üst sezgiselin birleştirilmesi sonucunda oluşan algoritmalarlardır. Miranda vd. (2018) bir otomatik algoritma

seçim yaklaşımı ile bir üst-sezgisel yöntemini birleştirerek yeni bir hibrid üst-sezgisel önermişlerdir. Bu çalışmada önerilen hibrid üst-sezgisel 60 optimizasyon problemini çözen Parçacık Sürü Optimizasyonu algoritmalarının tasarımını optimize etmek için kullanılmıştır.



### 3. ÜST-SEZGİSELLERİN SINIFLANDIRILMASI YAKLAŞIMLARI



Şekil 3.1. Üst-sezgisel yaklaşımların sınıflandırılması, (i) sezgisel arama uzayı yapısı ve (ii) geri bildirim öğrenme mekanizmaları (Burke vd., 2010d, Burke vd., 2019).

Şekil 3.1’de Üst-sezgisellerin sınıflandırılması iki boyutlu olarak düşünülür:

- Sezgisellerin arama alanının yapısı,
- Farklı geri bildirim bilgi kaynaklarıdır.

Arama alanının yapısına göre;

- Sezgisel seçim: var olan sezgisellerden seçim yapmak veya yöntemlerin seçimini yapmak
- Sezgisel üretim: yeni sezgisel üretmek için kullanılan yöntemlerdir. İkinci boyutta ise yapıcı ve pertürbatif arama yaklaşımları arasındaki farktır (Hoos ve Stützle, 2005). Pertürbatif yöntemler tüm aday çözümleri göz önüne alarak çalışırlar ve bir veya daha çok bileşenleri değiştirerek değişim uygularlar; yapıcı yöntemler ise belirli bir aday çözümü göz önüne alarak çalışırlar, burada bir veya birden fazla çözüm bileşeni atlanır ve bunlar iteratif olarak genişletilirler.

Bir üst-sezgisel, arama işlemlerinden bazı geribildirimleri kullanan bir öğrenme algoritmasıdır. Öğrenme sırasındaki geribildirim kaynaklarına göre çevrimiçi ve çevrimdışı olmak üzere ikiye ayrılırlar. Çevrimiçi öğrenmede algoritma bir problem örneğini çözerken öğrenme gerçekleşir; Çevrimdışı öğrenme de ise bir dizi öğrenme örneklerinden, program veya kurallar ile elde edilen bilgiler kullanılır ve bilinmeyen örnekleri çözmek için genelleştirme yapılır. Bu kategoriler güncel araştırmaları da yansıtmaktadır. Fakat daha geniş kapsamlı

kategorilerde vardır. Örneğin, yapıcı ve pertürbatif sezgiselleri kombine eden hibrid yöntemler (Garrido ve Riff, 2010) veya sezgisel seçim ve üretimi birleştiren hibrid yöntemler vardır (Krasnogor ve Gustafson, 2004; Maturana vd., 2010; Remde vd., 2012).



## 4. SEZGİSEL SEÇİM YÖNTEMLERİ

### 4.1. Yapıcı Düşük Seviyeli Sezgisel Temelli Yaklaşımlar

Bu yaklaşımlar aşamalı olarak bir çözüm oluştururlar. Boş bir çözüm ile başlarlar, daha sonra akıllı bir şekilde seçim yaparlar ve yapıcı sezgiselleri aşama aşama bir çözüm üretmek için kullanırlar. Bir üst-sezgisel frameworkü önceden var olan bir dizi yapıcı sezgiseli ile çalışır ve problemin o andaki durumu için en uygun sezgiseli seçmek için uğraşır. Bu süreç en son duruma ulaşıncaya kadar devam eder. Tam bir çözüme ulaşıldığı zaman yapıcı işlemler için doğal olarak sonlandırma gerçekleşir. Bu nedenle, sezgisel seçimi sırası sonludur ve kombinatoriyal optimizasyon problemleri boyutuna göre tanımlıdır. Ayrıca, ilginç olarak kısmi çözüm aşamaları ile bu aşamalar için gerekli olan sezgiseller arasındaki ilişkiyi öğrenme olasılığı vardır.

Bazı yaklaşımlar var olan yapıcı sezgisellerin hibridizasyonu ile üretilerek bazı alanlarda önerilmiştir, çizelgeleme (timetabling), kesme ve paketleme (cutting and packing), üretim çizelgeleme (production scheduling), kısıt sağlama (constraint satisfaction) ve gezgin satıcı (vehicle routing) problemleri, Çizelge 4.1’de örnek olarak verilmiştir. Çevrimiçi/çevrimdışı yaklaşımlar ve farklı yüksek-seviyeli stratejiler, veya öğrenme mekanizmaları bulunmuştur.

Çizelge 4.1. Düşük seviyeli sezgiselleri temel alarak çalışılan sezgisel seçim yöntemleri alanları

| Uygulama Alanı                                  | Kaynaklar                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Üretim Planlama (Production Scheduling)         | Fisher ve Thompson (1961, 1963)<br>Storer vd. (1992, 1995)<br>Dorndorf ve Pesch (1995)<br>Fang vd. (1993, 1994)<br>Norenkov ve Goodman (1997)<br>Hart ve Ross (1998); Hart vd. (1998)<br>Vazquez-Rodriguez vd. (2007a, b)<br>Ochoa vd.(2009)<br>Cano-Belman vd. (2010)<br>Garcia-Villoria vd. (2011) |
| Eğitimsel Çizelgeleme (Educational Timetabling) | Terashima-Marin vd. (1999)<br>Ahmadi vd. (2003)<br>Cheng vd. (2003); Asmuni vd. (2005, 2006)                                                                                                                                                                                                         |

|                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                | <p>Ross vd. (2003); Ross ve Marin-Blazquez (2004)<br/>         Burke vd. (2005a, 2006b)<br/>         Burke vd. (2007c); Qu ve Burke (2009)<br/>         Ochoa vd. (2009)<br/>         Li vd. (2011)<br/>         Pillay ve Banzhaf (2007); Pillay (2008)<br/>         Sabar vd. (2011)<br/>         Burke vd. (2014);<br/>         Soria-Alcaraz vd. (2014);<br/>         da Fonseca vd. (2016);<br/>         Kheiri vd. (2016); Kheiri ve Keedwell (2017);<br/>         Soria-Alcaraz vd. (2017)</p> |
| 1D Paketleme (1D Packing)                      | <p>Ross vd. (2002, 2003)<br/>         Marin-Blazquez ve Schulenburg (2007)</p>                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 2D Kesme ve Paketleme (2D Cutting and Packing) | <p>Terashima-Marin vd. (2006, 2007, 2009)<br/>         Garrido ve Riff (2007a, b)<br/>         Lopez-Camacho vd. (2010)</p>                                                                                                                                                                                                                                                                                                                                                                           |
| İşgücü Çizelgeleme (Workforce Scheduling)      | <p>Remde vd. (2007, 2009, 2012)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Kısıt Sağlama (Constraint Satisfaction)        | <p>Terashima-Marin vd. (2008)<br/>         Ortiz-Bayliss vd. (2010)</p>                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Araç Rotalama (Vehicle Routing)                | <p>Garrido ve Castro (2009); Garrido ve Riff (2010)<br/>         Akar vd. (2014)<br/>         Marshall vd. (2015)<br/>         Sabar vd. (2015)<br/>         Yin vd. (2016)<br/>         Sim ve Hart (2016)<br/>         Chen vd. (2016)<br/>         Mourdjis vd. (2016)<br/>         Tyasnurita vd. (2017)<br/>         Soria-Alcaraz vd. (2017)</p>                                                                                                                                                |

#### 4.2. Pertürbatif Düşük Seviyeli Sezgisel Temelli Yaklaşımlar

Bu yaklaşımların amacı bir çözüm geliştirmek için bir sezgiseli otomatik olarak seçme ve uygulama işlemidir. Çevrimiçi ve çevrimdışı makine öğrenmesi teknikleri verilen her adımdaki sezgiseli seçmek için sezgisel seçim kullanılmaktadır. Pertürbatif temelli üst-sezgisel yöntemler bir çok kombinatoriyal optimizasyon problemlerine uygulanmıştır ve Çizelge 4.2’de gösterilmektedir.

Çizelge 4.2. Pertürbatif sezgiselleri seçmek için en çok çalışılan uygulama alanı yöntemleri

| Uygulama Alanı                                  | Kaynaklar                                                                                                                                                                                                   |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Personel Çizelgeleme (Personnel Scheduling)     | Cowling vd.(2000, 2002b,c)<br>Cowling ve Chakhlevitch (2003)<br>Burke vd. (2003b)<br>Bai vd. (2012)<br>Mısır vd. (2010)                                                                                     |
| Eğitimsel Çizelgeleme (Educational Timetabling) | Cowling vd. (2000, 2002c)<br>Burke vd. (2003b, 2005b)<br>Bilgin vd. (2006)<br>Chen vd. (2007)<br>Bai vd. (2012, 2007)<br>Özcan vd. (2009, 2010)<br>Demeester vd. (2012)                                     |
| Yer Tahsisi (Space Allocation)                  | Burke vd. (2005c)<br>Bai ve Kendall (2005); Bai vd. (2008)                                                                                                                                                  |
| Kesme ve Paketleme (Cutting and Packing)        | Dowsland vd. (2007)<br>Bai vd. (2012)                                                                                                                                                                       |
| Araç Rotalama (Vehicle Routing)                 | Pisinger ve Ropke (2007)<br>Meignan vd. (2010)<br>Mısır vd. (2011)<br>K. Sim ve E. Hart.(2016)                                                                                                              |
| Spor Çizelgeleme (Sports Scheduling)            | Mısır vd. (2009)<br>Gibbs vd. (2010)                                                                                                                                                                        |
| Cross-domain (HyFlex)                           | Burke vd. (2010b, 2011a)<br>Ochoa vd. (2012a, b)<br>Özcan ve Kheiri (2011)<br>Walker vd. (2012)<br>Drake vd. (2012)<br>Mısır vd. (2012)<br>Ping-Che vd. (2012)<br>Chan vd. (2012)<br>Gaspero ve Urli (2012) |

#### **4.2.1. Çevrimiçi öğrenmeli üst-sezgiseller**

Öğrenme bir problem örneğini çözmek için algoritma çalışırken gerçekleştirilir. Bu yüzden işlem-bağımlı yerel özellikler uygulama için uygun düşük-seviyeli sezgiseli tanımlamak için yüksek-seviyeli strateji kullanabilirler. Üst-sezgiseller içindeki Çevrimiçi Öğrenmeli örnekleri; sezgisel seçimi için takviyeli öğrenme (reinforcement learning) ve genellikle, meta-sezgisellerin kullanımı sezgisellerin arama uzayı üzerinde yüksek seviyeli arama stratejileridir.

#### **4.2.2. Çevrimdışı öğrenmeli üst-sezgiseller**

Bu fikir örneklerin eğitiminden program veya kurallardan elde edilen bilgidir, bu gizli örneklerin çözüm süreci için genelleştirilir. Üst-sezgiseller içindeki çevrimdışı yaklaşımlara örnek: öğrenmeli sınıflandırma sistemleri (learning classifier systems), durum tabanlı çıkarsama (case-based reasoning) ve genetik programlamadır.

## 5. SEZGİSEL ÜRETİM YÖNTEMLERİ

Sezgisel üretim yöntemleri yeni sezgisel üretmek için kullanılan yöntemlerdir. Çizelge 5.1'deki otomatik sezgisel üretimi yöntemlerinin uygulama alanlarındaki çalışmaları göstermektedir. Bu konu üzerindeki çalışmalarda genellikle yeni sezgiselleri üretmek için genetik programlama kullanılmıştır. Genetik programlama zor kombinatorial optimizasyon problemlerine, örneğin Sağlanabilirlik (satisfiability), (Bader ve Poli, 2007 ; Fukunaga, 2002; Fukunaga, 2004; Fukunaga, 2008; Kibria vd., 2006), bidon paketleme (bin packing) (Burke vd., 2006; Burke vd., 2007; Burke vd., 2010b; Lopez-Camacho vd., 2014; Sim vd., 2015), gezgin satıcı problemi (the traveling salesman problem) (Keller vd. 2007a; Keller vd., 2007b) ve üretim çizelgeleme (production scheduling) (Branke vd., 2016; Dimopoulos vd., 2001; Geiger vd. 2006; Hart vd., 2016; Tay vd., 2008) sezgisellerin otomatik üretilmesi ile başarıyla uygulanmıştır. Genetik programlama ağaçlar, graflar, diller veya doğrusal program kodlama alanlarında diğer evrimsel algoritmalarından farklıdır. Optimizasyon problemlerine diğer evrimsel algoritmalar uygulanırken, genetik programlama makine öğrenmesi alanında da kullanılabilir. Bazı genetik programlama tabanlı üst-sezgiseller, yerel arama sezgiselleri (Bader ve Poli, 2007; Burke vd., 2012; Fukunaga, 2004; Fukunaga, 2008; Keller vd. 2007a; Keller vd., 2007b) veya evrimsel algoritmalar (Oltean, 2005) geliştirmiştir. Sezgisel üretim özellikleri ile ilgili diğer basit çalışmalarda vardır. Bu çalışmalarda, fikir, herhangi bir zamanda uygulandıklarında etkili bir performans gösteren sezgisel çiftleri bulmaktır. Bunlardan bazıları üst-sezgisel üretim sürecine çevrimiçi uygulanırken, bazıları da çevrimdışı uygulanır. Bu araştırmalarda aynı zamanda memetik algoritmalar veya iteratif yerel arama gibi önceden tanımlı kullanılan fakat öğrenmesiz yöntemler de vardır. Üretim süreci boyunca, buradaki elementler farklı çalışmalar için değişik sezgisel oluşturmak amacıyla kullanılmışlardır. Bu elementlerin listesi: primitif elementler, problem-bağımsız elementler ve sezgisellerdir.

Çizelge 5.1. Sezgisel Üretim Yöntemlerinin Uygulama Alanları

| Uygulama Alanı                                        | Kaynaklar                                                                                                                                                                           |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Üretim Çizelgeleme (Production Scheduling)            | Jakobovic vd. (2007)<br>Ho ve Tay (2005)<br>Tay ve Ho (2008)<br>Dimopoulos ve Zalzala (2001)<br>Geiger vd. (2006)                                                                   |
| Kesme ve Paketleme (Cutting and Packing)              | Burke vd. (2006a, 2007b,a)<br>Poli vd. (2007)<br>Kumar vd. (2008)<br>Allen vd. (2009)<br>Burke vd. (2010e,c, 2011b)<br>Özcan ve Parkes (2011)<br>Burke vd. (2012)<br>Sim vd. (2012) |
| Sağlanabilirlik (Satisfiability)                      | Fukunaga (2002, 2004, 2008)<br>Bader-El-Den ve Poli (2007, 2008)<br>Lokketangen ve Olsson (2010)                                                                                    |
| Gezgin Satıcı Problemi (Travelling Salesman Problem)  | Keller ve Poli (2007a, b, 2008a, b, c)<br>Oltean ve Dumitrescu (2004)                                                                                                               |
| Fonksiyon Optimizasyonu (Function Optimisation)       | Oltean (2005)<br>Oltean ve Grosan (2003)<br>Tavares vd. (2004)                                                                                                                      |
| Zamanlama ve Çizelgeleme (Timetabling and Scheduling) | Pillay (2009)<br>Bader-El-Den vd. (2009)                                                                                                                                            |

## 6. MEMETİK ALGORİTMALAR

Memetik Algoritmalar(MA) Pablo Moscato (Moscato, 1989) tarafından bulunmuştur. MA'lar, tümleşik optimizasyon, çok amaçlı optimizasyon, dinamik optimizasyon ve sürekli optimizasyon alanlarında başarıyla uygulanmışlardır. Memetik algoritmaların, doğadan esinlenen Tavlama Benzetimi (Simulated annealing), Karınca Kolonisi, Yapay Sinir Ağları, Evrimsel algoritmaların (EA) aksine net bir doğa benzetimi yoktur. Bunun bir zayıflık ya da güç olduğu tartışılabilir. Bu çalışmada her yöne çekilebilecek ve alanında büyük gelişmelere yol açabilecek potansiyel güçlü bir doğa esintili algoritma olduğu savunulabilir. Evrimsel Algoritmaların erken tarihinde geleneksel Genetik Algoritmalar görülür, bunlarda basit ikili kodlama, çaprazlama, mutasyon ve uygunluk tabanlı seçim yapılır ve değiştirilemezler. Deneysel sonuçlara göre “Baldwin effect” ve “Lamarckian evolution” adı verilen evrimsel çalışmaların daha efektif sonuçları için çalışmalar yapılmıştır. Böylece bu çalışmalar evrimsel algoritmaların sezgiselleri içeren özel çözümlerle, uygun birleştirmeler yapılarak yerel arama algoritmaları ile dinamik bir global arama yapılabileceğini belirtilmiştir. Dahası teorik sonuçlara ek olarak “kara kutu” problemlerinin etkili bir şekilde çözülmesi bu yöntemin desteklenmesini hızlandırmıştır. Özel (spesifik) örneklem (domain) bilgisi EA çerçevesine özel çaprazlama ve mutasyon operatörleri ile, karışık problemlerle, hızlı popülasyon başlatılmasıyla, karmaşık(komplex) uygunluk fonksiyonlarıyla, local arama sezgiselleriyle MA'lar eklenmiştir. Son zamanlarda R.Dawkin'in “memes” kavramı, Memetik Algoritma literatürüne girmiştir. MA'nın tasarımında problem çözümündeki sabit ve statik örneklem bilgisi kırıldı ve evrim geçirmeye müsait stratejiler sunulmuştur. Böylece Dawkin'in “memes” leri ile sürekli ve açık uçlu problemlerin çözümünde evrimleşebilen arama stratejileri ile kritik bir bağ kurulmuştur.

Memetik algoritmaların geliştirilmesi ile yazılım geliştirilmesi süreçleri farklı değildir. MA'nın arkasındaki popülerite, daha iyi çözümler sunarak, Tabu Arama (Tabu Search) veya Tavlama Benzetimi (Simulated Annealing) gibi diğer optimizasyon algoritmaları üzerine kolayca uygulanabilmesidir. Gerçekten de, herhangi bir başarılı doğa-esinli arama metodu gerçek problem çözme özelliği popüleritesine sahip değildir ki bu rekabetçi bir yöntemden yoksundur, bundan

ziyade bariz tasarım kusurlarına (parametre sayısının çokluğu, uygulayıcı teoriden yoksunluk gibi) rağmen, sağlıklı bir araştırma ve uygulama ortamı yapılandırmaya yardımcı olur. Bu doğadan esinlenen arama metodları kendi doğruları içinde “arama programları” ve “arama paradigmaları” ile hesaplanır. Böylece memetik algoritmaların temel bileşenleri nelerdir sorusu arama paradigmalarının merkezinde sahne alır. Bu literatür birçok çalışma içerir ve çeşitli yöntemlerle memetik algoritmaları sınıflandırır. Memetik algoritmaların çoğunluğu Evrimsel Algoritmalar (Evolutionary Algorithms)- Yerel Arama (Local Search) karışımı olarak bilinse de, diğer meta-sezgisellerden türemiş sayısız MA vardır. Örneğin; Karınca Kolonisi, Yapay Bağışıklık, Parçacık Sürü ve bu uygulamaların hepsinin ortak yönü, genel arama ve yerel arama stratejileri arasında önemli karşılıklı etkileşim olmasıdır.

Bulunduğundan bu zamana kadar MA’lar çeşitli problemlere uygulanmışlardır. Örneğin zaman çizelgeleme problemleri gezgin satıcı problemi, kareli atama problemleri, çok amaçlı optimizasyon problemleri verilebilir.

Dört nesil memetik algoritmalar vardır. İlk nesil iyi bir çözüme ulaşmak için yerel arama yöntemleri ile popülasyon tabanlı algoritmaların hibrid olarak kullanılmasıyla oluşmuştur. İkinci nesilde multimeme kavramı ortaya çıkmaktadır. Bir multimeme bir dahaki nesile bu memelerin özelliklerini aktarmaktadır. Üçüncü nesil MA’larda kendini üreten (self-generation) ve birlikte evrim (co-evolving) kavramları bulunmuştur. Burada çeşitli öğrenme stratejileri ile adaptif bir şekilde parametre seçimi yapılmaktadır. Dördüncü nesil MA’lar “Memetic Computing” kavramı ile gelişmektedir. Bu kavram bir MA framework’ü olarak düşünülmektedir.

Memetik algoritmalar ile üst-sezgisel algoritmalar arasında parametrelerin adaptif olarak seçilmesi gibi benzerlikler vardır.

## 7. CROSS-DOMAIN SEZGİSEL ARAMA YARIŞMASI (CHESC2011/ CHESC2014)

Hyperion ve Hyper-heuristics Flexible Framework (HyFlex) farklı üst/meta-sezgiselleri karşılaştırmak ve uygulamak için geliştirilen yazılım kütüphaneleridir. Hyflex frameworkü çeşitli problemler için üst-sezgisel yaklaşımlarını uygulanmasına olanak sağlar ve tekrar kullanılabilir üst -sezgisel bileşenler için bir arayüz olarak kullanılır. Bu yarışmada 20 farklı algoritma yarışmak için kullanılmıştır. HyFlex (Cross-Domain Heuristic Search Challenge, CHESC2011) için altı farklı problem kullanılmasına olanak sağlamıştır. Bu problemler: Bool Tatmin (Boolean Satisfiability (SAT)), Tek Boyutlu Bidon Paketleme (One Dimensional Bin Packing (BP)), Permütasyon Seri Üretim (Permutation Flow Shop (PFS)), Personel Çizelgeleme (Personnel Scheduling (PS)), Gezgin Satıcı (Travelling Salesman Problem (TSP)) ve Araç Rotalama Problemi (Vehicle Routing Problem (VRP)). Her bir problem alanı için mutasyon (MU), yerel arama (HC), yeniden oluşturma ve çaprazlama (XO) düşük-seviyeli sezgiselleri kullanılmıştır. CHESC2011 deki her bir düşük-seviyeli sezgiseller için bazı parametre ayarları kullanılmıştır. Mutasyon operatörlerini değiştirmek için mutasyon yoğunluğu (mutation density) parametresi bulunmaktadır. Yüksek mutasyon yoğunluğu daha geniş bir değer aralığında çözümü mutasyona uğratar. Düşük mutasyon yoğunluğu çözüm üzerinde daha az değişiklik yapar. Yerel arama düşük-seviyeli sezgiseli için arama derinliği (depth of search) parametresi kullanılmıştır. CHESC2014, 2. Cross-domain sezgisel arama yarışmasında ise üst-sezgiselleri tek tek çözmek yerine toplu bir şekilde ele alınmak hedeflenmektedir. Problemi toplu bir şekilde çözenin bazı avantajları vardır;

- Çaba Dengelemesi (Effort Balancing): Eğer bir örnek diğerlerinden daha kolay ise onun için daha az hesaplama zamanı ayırmak gerekir ve daha zor örnek için daha fazla zaman ayrılmış olup zaman dengesi sağlanabilecektir.
- Örnekler-Arası Öğrenme (Inter-instance learning): Bazı örnekler aynı etki alanında ise o üst-sezgiselden daha sonra daha iyi performans elde etmek için daha önceki örneklerden performansı öğrenilebilir.

## 7.1. CHeSC2011’de Kullanılan Problem Alanları

HyFlex frameworkü optimizasyon ve sezgisel araştırmaları desteklemek için geliştirilmiş bir yazılımdır. Üst-sezgisellerin amaçlarından birisi olan genelleştirme seviyesini artırmak için geniş problem alanlarında yeni metodların test edilmesi olanağı sunmasıdır. CHeSC2011 6 problem alanında düşük seviyeli sezgiseller seti kullanılarak geliştirilmiş bir framework üzerinde gerçekleştirilmiştir. Aşağıdaki Çizelge 7.1’de her bir problem alanı için problem örnekleri ve düşük seviyeli sezgisellerin sayısı verilmiştir.

Çizelge 7.1. CHeSC2011 Problemlerindeki Düşük Seviyeli Sezgiseller

| Alan Adı              | Örnekler | Yeniden Oluşturma | Mutasyon | Yerel Arama | Çaprazlama |
|-----------------------|----------|-------------------|----------|-------------|------------|
| Bin Packing           | 12       | 2                 | 3        | 2           | 1          |
| MAX-SAT               | 12       | 1                 | 6        | 2           | 2          |
| Permutation Flow Shop | 12       | 2                 | 5        | 4           | 4          |
| Personnel Scheduling  | 12       | 3                 | 1        | 5           | 3          |
| TSP                   | 10       | 1                 | 5        | 3           | 4          |
| VRP                   | 10       | 2                 | 3        | 3           | 2          |

## 7.2. Tek Boyutlu Bidon Paketleme Problemi(1D Bin Packing)

Tek boyutlu bidon paketleme problemi geniş bir uygulama alanı olan tanınmış bir NP problemidir. Bu problem belirli bir eleman dizisinin bir konteyner dizisine homojen bir şekilde paketlenmesidir. Her bir elemanın ( $j$ ) bir ağırlığı ( $w_j$ ) vardır ve her bir konteyner bir kapasiteye ( $c$ ) sahiptir. Amaç fonksiyonu ise her bir elemanı konteyner kapasitesi kullanımını minimize ederek konteynerlara atamaktır ve elemanların toplam ağırlığı konteyner kapasitesini aşamaz. Bu problem Martello ve Toth (1990 a,b) tarafından aşağıdaki gibi tanımlanmıştır:

$$\text{minimize } \sum_{i=1}^N y_i \quad (7.1)$$

$$\sum_{j=1}^N w_j x_{ij} \leq c y_i \quad i \in N = \{1, \dots, n\}, \quad (7.2)$$

$$\sum_{i=1}^N x_{ij} = 1 \quad j \in N, \quad (7.3)$$

$$y_i \in \{0,1\} \quad i \in N \quad (7.4)$$

$$x_{ij} \in \{0,1\} \quad i, j \in N \quad (7.5)$$

Yukarıdaki formülasyonda  $y_i$  i. Konteynerı gösterirken,  $x_{ij}$  ise j. elemanın i. Konteynerın içerisinde paketlenmesini temsil eder ve n kullanılabilen bütün konteynerların sayısıdır. Çözümün amaç fonksiyonu ise aşağıdaki gibi hesaplanır.

$$fitness = 1 - \frac{\sum_{i=1}^n (fullness_i / C)^2}{n} \quad (7.6)$$

### 7.3. Mantıksal Sağlanabilirlik Problemi- Boolean Satisfiability (MAX-SAT)

MAX-SAT problemi kombinatoriyal optimizasyon problemlerini en uygun örneklerden birisidir. Problemin çözüm amacı, verilen n değişkenli bir mantık önermesinin, değişkenlere gerekli değerler atanarak sağlanıp sağlanamayacağına karar vermektir. Örnek;  $x_1 = true, x_2 = true, x_3 = true, x_4 = true$ .

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee x_4) \quad (7.7)$$

### 7.4. Permütasyon Akış Tipi Problemi (Permutation Flow Shop)

Permütasyon akış tipi çizelgeleme üretim endüstrisi alanındaki bir NP-zor problemidir. Bu problemde birbirinden bağımsız n adet işin m tane ardışık makinede çizelgenmesi hedeflenmektedir. Amaç sistemi terk eden en son işin tamamlanma süresini minimize etmektir. Her bir iş i ile, her bir makine j ile ve işin makinedeki işlem zamanı ise  $p_{ij}$  ile gösterilir. Problem çözümünde 2 yeniden oluşturma, 5 mutasyon, 4 yerel arama ve 4 çaprazlama sezgiseli kullanılmıştır.

### 7.5. Personel Çizelgeleme (Personnel Scheduling)

Personel çizelgeleme problemi verilen bir planlama süresi içerisinde, gün ve zamanların çalışanlara ayrılmasını içerir. HyFlex içerisinde hemşire nöbet çizelgeleme örnekleri kullanılmıştır. Ağırlıklı amaç fonksiyonu kısıtlara ve amaçlara göre çözümün kalitesinin ölçülmesi olarak tanımlanır. Örneğin; belli bir nöbet süresi istenmeyebilir veya bu süre içerisine atama yapılmasına izin verilmeyebilir. Bu problemin çözümü için iki gerçek dünya problemi ele

alınmıştır ve 3 yeniden oluşturma, 1 mutasyon, 5 yerel arama ve 3 çaprazlama sezgiseli kullanılmıştır.

#### **7.6. Gezgin Satıcı Problemi (Travelling Salesman Problem)**

Gezgin satıcı problemi en çok bilinen kombinatorial optimizasyon problemidir. Verilen şehirler listesi ve verilen şehir çiftinin mesafesi kullanılarak minimum mesafesi olan rotayı bulmayı amaçlayan bir problemidir. TSP'nin en büyük özelliği kombinatorial optimizasyona çözüm yolları aranırken kullanılan deneme problemi olarak kullanılmasıdır. Mesafenin en küçüklenmesi bir optimizasyondur. TSP kombinatorial problem olduğundan dolayı diferansiyel matematik kullanarak türevlerinin sıfırlanması ile bulunan çözüm büyük çaptaki problemlerde kolayca kullanılamamaktadır. Yanıt süreklilik gösteren çözüm kümesinde değildir. Bu problemin çözümü için 1 yeniden oluşturma, 5 mutasyon, 3 yerel arama ve 2 çaprazlama sezgiseli kullanılmıştır.

#### **7.7. Araç Rotalama Problemi (Vehicle Routing Problem)**

Araç rotalama problemi (VRP) polinom zamanda deterministik olmayan (NP-Zor) kombinatorial optimizasyon problem türlerinden biridir. Farklı amaçlar hedef problemin çözümüne bağlı olarak düşünülür. Tipik amaçlar; mesafe minimizasyonu, zaman minimizasyonu, araç sayısı minimizasyonu, müşteri servislerindeki maliyet minimizasyonu gibi amaçlardır. Tüm VRP uygulamalarında kesin çözüme ulaşan tek bir çözüm yolu bulunmamaktadır. Probleme kesin bir çözüm bulmamakla beraber yaklaşık sonuç bulabilen sezgisel yaklaşımlar karmaşık bir optimizasyon problem olan araç rotalama problemlerinde sıklıkla kullanılan çözüm yöntemleridir. VRP çözümlerinin geliştirilmesi ile ilgili akademik çalışmalarda matematiksel modeller yerine sezgisel yöntemler ağırlık kazanmaktadır. Bu problemin çözümü için 2 yeniden oluşturma, 3 mutasyon, 3 yerel arama ve 2 çaprazlama sezgiseli kullanılmıştır.

## 8. SINAV ÇİZELGELEME PROBLEMİ

Bütün eğitim kurumları sınav ve ders çizelgeleme problemi ile karşı karşıyadırlar. Bu iki problemde de herbir ders veya sınıf istenen kısıtları sağlayarak belirlenen zaman dilimlerinde dersliklere atanmalıdır, ancak her iki problem de farklı kısıtlara sahiptir. Örneğin, bir sınav çizelgesinde bir dersliğe birden fazla sınav atanabilecekken, ders çizelgelemesinde bir dersliğe aynı zaman diliminde birden fazla ders atanamaz. Ders ve sınav çizelgeleme aynı zamanda yapıları, modelleri süreçleri açısından da farklılık gösterirler. Sınav çizelgeleme problemi son on yıldır değişik yöntemlerle geniş bir şekilde ele alınmıştır. Bu alandaki yaklaşımlardaki yöntemlerin birçoğu 1996'da Toronto Üniversitesi tarafından sağlanan örnekleri kullanmışlardır. Bu örnekler gerçek dünya verileri içermektedir. Birinci (ITC2002) ve ikinci (ITC2007) uluslararası çizelgeleme yarışmalarındaki örnekler üzerinde testler yapmak ve teknikler geliştirmek araştırmacıların ilgisini çekmiştir. Özellikle ITC2007 yarışmasında gerçek dünya verileri kullanılmıştır ve bu alandaki uygulamalar için önemli bir rol oynamaktadır. Yarışmada çizelgeleme alanı içerisinde başlıca üç konu ele alınmıştır:

1. Sınav Çizelgeleme (Examination Timetabling )
2. Kayıt Sonrası Ders Çizelgeleme (Post-Enrolment Course Timetabling)
3. Müfredat-Tabanlı Ders Çizelgeleme (Curriculum-Based Course Timetabling)

Bu yarışmada örnek veriler ile ilgili 3 bölüm vardır. Bu ayrım, akademik komütelerde yer alan eğitimsel çizelgeleme (educational timetabling) temel tiplerini belirlemek için bir çerçeve sunması açısından önemlidir. Bütün bölümler eğitimsel çizelgeleme (educational timetabling) şemsiyesi altında toplanmasına rağmen, tanımlanan problemler arasında önemli farklılıklar vardır ve bunlar yarışma web sitesinde ve McCollum vd. tarafından (McCollum vd., 2010) tartışılmıştır. Ek olarak bu alanlardaki teknik raporlar web sitesinde her bir bölümün altında mevcuttur (McCollum vd., 2007; Lewis vd., 2007; Di Gaspero vd., 2007). Bu teknik raporlarda eğitimsel çizelgeleme bölümünde kullanılan gerçek dünya modeli ele alınmıştır. Raporlar problem tanımını ve ITC2007 web sitesinin içeriğindeki tanımların ayrıntılı ve titiz bir açıklamasını içerir.

### 8.1. Sınav Çizelgeleme Problemi: Modeli

Önceki çalışmalar 1996'da University of Toronto'daki Carter tarafından bulunan Toronto örnekleri üzerinde odaklanmışlardır. Bu örnekler ve onlarla ilgili çeşitli konular Qu ve arkadaşları tarafından da tartışılmıştır (Qu vd., 2009). Ancak Toronto örnekleri çok sınırlı bir model kullanmaktadır ve bu yüzden çizelgeleme problemlerinin modelleme karmaşıklığı çizelgeleme araştırma alanındaki bir dizi önemli ve zorlu sorunlar oluşturmaya devam etmektedir (McCollum 2007).

Genel çizelgeleme üzerine bilimsel yazında bazı araştırmalar ve problemin gerçek dünya örneklerini modellemek için araçlar vardır. Tsang vd. (1999) kısıt sağlama problemleri olarak sınav çizelgeleme için yüksek seviyeli bir dil geliştirmiştir. Di Gaspero ve Schaerf (2001) EASYLOCAL++ ismi verdikleri genel çizelgeleme problemleri üzerinde yerel arama algoritmalarını kolay uygulamak için bir yazılım geliştirmişlerdir.

De Werra vd. (2002) sınıf-öğretmen çizelgeleme problemleri için genişletilebilir basit bir model sunmuşlardır. Bu problemlerin hesaplama karmaşıklığı çalışılmış ve problemlerin bazıları NP-complete olarak gösterilmiştir. Ancak, daha fazla çalışmanın önemi ilgili konuları ve bunların ele alınması için gereklidir. Ek olarak, gerçek dünya sınav çizelgeleme araştırmalarının gelişmesi ve daha iyi anlaşılabilmesi için temel destek sağlamak amacıyla gereklidir. Bu konu ile ilgili çok çeşitli kurumsal uygulamalar vardır örneğin; Al-Yakoob vd. (2010).

Genel olarak, problemin modelleme sorunları yaygın olarak son on yıl içinde tartışılmıştır ve kurumsal uygulamalarda çok çeşitli veriler olduğu için evrensel tam bir model olmaması şaşırtıcı değildir.

## 8.2. Sınav Çizelgeleme Problemi: Algoritmalar

Sınav çizelgelem ile ilgili son on yılda önemli ölçüde yapılmış araştırmalar vardır. Qu vd. (2009) çalışmalarında oldukça geniş bir kaynak araştırması yapmışlardır. Bu araştırmalarda doğal olarak meta-sezgiseller çok çalışılmıştır. Örnek olarak; Tabu Arama (Di Gaspero 2002; White ve Xie 2001; Paquete ve Stützle 2002), Tavlama Benzetimi (Burke vd. 2004; Merlot vd. 2003; Thompson ve Dowsland 1998), Genetik Algoritmalar (Ross vd. 1996; Wong vd. 2002), Memetik Algoritmalar (Burke vd. 1996), Karınca Algoritmaları (Dowsland ve Thompson 2005; Azimi 2005) ve Çok-Amaçlı Evrimsel Algoritmalar (Mumford 2007) gösterilebilir. Meta-sezgisel çalışmalar arasında önemli hibridleme çalışmaları da vardır. Gerçekte, bu hibridler standard Toronto verileri üzerinde gerçekleştirilen çok etkin yaklaşımlardır. Bu etkin yaklaşımlar arasında önceki çizelgeleme teknikleri örneğin graf sezgiseller (Burke ve Newall 2004; Burke vd. 2005, 2006, 2007) ve kısıt-temelli teknikler (Merlot vd. 2003) bulunmaktadır.

Bu araştırma algoritmalarının ana teması arasında çok etkili bir komşuluk yapısı tasarımını içeren yeni eğilimler vardır. Örnek olarak; çizelgeleme problemleri için değişken komşu araması (Burke vd. 2010; Ahmadi vd. 2003) verilebilir. Arama esnekliği daha geniş kısıtlarla daha karmaşık problemleri çözmeye çalışmak için geliştirilmiştir. Bazı araştırmalar çizelgeleme yaklaşımlarının genelliğini artırmak amacıyla başlamış ve önemli sonuçlar elde edilmiştir, üst-sezgiseller (Burke vd., 2003) araştırma alanında önemli şekilde dikkat çekmiştir (Burke vd. 2002, 2005, 2006, 2007, 2010; Ross vd., 2004).

## 8.3. Sınav Çizelgeleme Modeli

Sınav çizelgeleme probleminde temel sorun, çeşitli sert kısıtları sağlarken tanımlı bir sınav süresi içinde belirli bir sayıdaki zaman dilimlerine sınavların atanmasını içerir. Diğer çizelgeleme alanlarında olduğu gibi, uygun bir çözüm tüm esnek kısıtların sağlanması ve ihlallerin ceza ağırlıklılarının toplamıdır.

ITC2007'deki veriler önceki veriler ile karşılaştırıldığında, kısıtlar, kaynaklar ve sınav oturumları üzerinde yeni ve ek bilgiler sağlanmıştır. Örneğin, sert kısıtlar

aısından, derslik sayıları ve büyüklükleri verilmiştir. Ayrıca, her dönem yapısı, süresi ve sayısı hakkında bilgi de bulunmaktadır. Esnek kısıtlar aısından bir üniversitenin sınav çizelgeleme çözüm kalitesini nasıl ölçeceği ile ilgili pratik bilgiler sağlanmıştır (McCollum vd., 2012).

Bu tez çalışmasındaki amaç ise problem modelini oluşturmak ve örnek bir uygulama yapmaktır.

### 8.3.1. Problem tanımı

Problem aşağıdakilerden oluşur:

- Bir sınav oturumu belli bir zaman uzunluğundaki periyotlarda yapılır. Bu periyotlar ve her bir periyot süresi daha önceden sağlanır.
- Sınavlar kümesi verilen periyotlar içerisinde zamanlanmalıdır. (Sınav kodları yoktur, bunun yerine 0'dan başlayan artan sırada numaralandırma kullanılmıştır).
- Öğrenciler kümesi her sınava kayıtlıdır. Her öğrencinin bir dizi kayıtlı olduğu sınav vardır. Bir sınava kayıtlı öğrenciler sınavda oturacak veya sınava girecek olarak düşünülebilir.
- Derslikler kümesi kapasiteleri ile birlikte verilir.
- Herhangi bir uygun çözümün sağlaması gereken kısıtlar tanımlıdır.
- Uygun çözümler üzerinde örneğe bağlı özel kısıtlar vardır.
- Esnek kısıtlar ana hatlarıyla belirlenmiştir. Bunlar kısıtların sağlanması istenir fakat gerekli değildir. Kısıtların ihlalleri uygun çizelgeyi hesaplamak için kullanılan cezalara katılır.
- Esnek kısıtların ağırlık detayları problem içerisinde bulunur.

Uygun çizelge bütün sınavlar için bir tanedir, sınavlar tek bir derslik ve periyoda atanmıştır ve aşağıdaki bütün kısıtları sağlamalıdır:

- Hiçbir öğrenci aynı anda birden fazla sınava giremez.
- Dersliklerin kapasitesi herhangi bir sınav oturumu boyunca aşılmamalıdır.
- Periyot süreleri aşılmamalıdır.
- Periyot-bağılı sert kısıtlar sağlanmalıdır (örneğin A sınavı B sınavından sonra olacaktır).

- Derslik-bağılı sert kısıtlar sağlanmalıdır (örneğin, A ve B sınavları farklı dersliklerde olacaktır).

Bu çizelgelemede ders çizelgelemenin aksine sınavlar esnasında dersliklerin paylaşımına genellikle izin verilir.

Esnek kısıtlar ve onların ihlallerinin cezaları aşağıdaki gibidir:

- Arka arkaya iki sınav: Öğrencileri aynı günde arka arkaya iki sınav yapmaktan kaçınılmalıdır.
- Aynı günde iki sınav: Öğrencileri bir günde iki sınav yapmaktan kaçınılmalıdır, komşu olmayan periyotlar da bile aynı gün iki sınav atamaktan kaçınılmalıdır. Bu kısıtlama sadece aynı gün ikiden fazla sınav periyodu olduğunda önem kazanmaktadır.
- Sınavların belirli olarak dağıtılması: öğrencilerin kurum tarafından belirtilen (dönem sayısı) bir zaman aralığı içinde birden fazla sınava atama yapmaktan kaçınılmalıdır. Bu genellikle, sınavların birbirine çok yakın olmamasının öğrenciler tarafından tercih edilmesini karşılamayı denemek için kullanılır.
- Periyotlar içerisinde sınav sürelerinin karıştırılmaması: Aynı derslik ve periyoda farklı sürelerde olan sınavları atamaktan kaçınılmalıdır. Eğer sınavlar farklı sürelerde ise bu sınav gözetmenliğini daha da zorlaştırabilir ve daha önce sınavı bırakan öğrenciler kalanları rahatsız edebilir.
- Çizelgede olan büyük sınavlar: Çizelgede büyük sınavları çizelgenin en sonuna atamaktan kaçınılmalıdır.
- Periyot ile ilgili esnek kısıtlar: Belirli bir sınavı belirli bir periyoda atamaktan kaçınılmalıdır. Burada böyle cezalar için birçok sebep olabilir. Örneğin, Sınav sırasında ortaya çıkan soruları cevaplamak için uygun öğretmen olmayabilir.
- Derslik ile ilgili esnek kısıtlar: Bazı derslikleri kullanmaktan kaçınılmalıdır. Örneğin gerçekten ihtiyaç olmadıkça yönetim ofisi bu dersliği açmaktan kaçınmak isteyebilir.

Bu kısıtlar iki gruba ayrılabilir: Bunlar; kaynağa özgü olanlar ve genel ayarlara sahip olanlar şeklindedir. Periyot ile ilgili ve derslik ile ilgili kısıtlar kaynağa özgü kısıtlardır örneğin her bir periyot ve derslik için bu ayarlar yapılabilir.

Kaynakların nasıl kontrol edileceği bir çözüm üretme sırasında belirlenir. Bu değerler verilerdeki derslikler ve periyodlara girildikten sonra bulunabilir. Diğer bütün esnek kısıtlar genel ayarlar olarak adlandırılır ve birbirine göre ayarlanabilir.

Kurumlar kendi özel ihtiyaçlarına uygun bir çözüm üretmek amacıyla farklı esnek kısıtlar (birbirlerine göre) için farklı ağırlıklar verebilirler. Bu 'Kurumsal Model' ve ilişkili ağırlıkları içerir ve Kurumsal Model İndeksi olarak da tanımlanmaktadır. Esnek kısıtların bu göreceli ağırlıkları kurumun ihtiyaçlarına göre uyarlanır ve kurum için çözümün kalite ölçüsünü sağlar. Bu ağırlıklar yarışma web sayfasında yayınlanan her veri kümesi içinde tanımlanmıştır.

Bir çözüm oluştururken, tüm son kullanıcılara tatmin edici kararda çözümler üretmek amacıyla esnek kısıtların çeşitli değerlerini değiştirebilmeleri için ortak bir alan olduğunu belirtmek gerekir. Problem tanımının aksine veriler içerisinde verilerinin esnek kısıtlarının ağırlıklarını daha önceden belirlenmiştir. Buna ek olarak, verilerin içerisindeki her bir veri seti için farklı ağırlık ayarlarını ayarlamak mümkündür.

Özel sınav çizelgeleme çözümlerinin nasıl yapılacağı araştırma problemlerine açıktır ve sayısız yeni yöntemler vardır (Asmuni vd., 2006).

### **8.3.2. Çözümler için framework**

Genellikle, gerçek-dünya uygulamalarında son kullanıcılar için bütün kısıtlar eşit derecede önem taşımazlar. Sıklıkla kısıtlar sağlanması gereken sert kısıtlar ve amaç fonksiyonuna bir ceza maliyeti eklenerek ihlal edilebilen esnek kısıtlar olarak ele alınırlar. Ancak, böyle iki seviyeli yaklaşım kullanıcıların isteklerini her zaman karşılamayabilir, bu yüzden kısıt hiyerarşisi olan çoklu seviyeler yaygınlaşmıştır (Borning vd., 1987, 1992). Bu hiyerarşinin altındaki temel fikir kısıtların çoklu seviyeye ayrılmasıdır ve yüksek seviyedeki kısıtlar düşük seviyeli kısıtlara göre daha öncelikli olarak sağlanmalıdır. Bu hiyerarşi kavramı aynı zamanda öncelikli kısıtlandırma (prioritised constraints) olarak bilinir, yapay zeka alanında oldukça fazla çalışılmıştır ve kombinatorial optimization

problemleri için kısıt sağlama yaklaşımıdır. (Dubois vd., 1996; Meseguer vd., 2003; Henz vd., 2004; Callennec ve Boulic, 2004, Borning vd., 1987)

Bir kısıt hiyerarşisinde, programcı ve kullanıcı gerekli ve tercih edilen kısıtları belirleyebilir. Gerekli kısıtlar mutlaka sağlanmalıdır. Sistem mümkünse tercihli kısıtları sağlamaya çalışır, fakat yapamazsa hata durumu oluşmaz.

Bu çalışmalar esnek kısıtlar için de bir hiyerarşiye izin verir, ancak bu kısıtlamalar için normalde iki seviye istenir ki normalde (sert) gerekli olarak kabul edilen kısıtların ihlali son çözelgede hata durumuna neden olur.

ITC2007'deki kısıtlar içerisinde etkin olarak kullanılan üç düzeyde kısıt vardır:

- SERT: Bu kısıtlar hiçbir zaman kırılmaz. Herhangi bir çözüm bu kısıtları ihlal ederse değersiz sayılır.
- ESNEK-SERT: Bu kısıtlar serttir fakat işlem sırasında ihlal edilebilir ve tamire uygun değildir. Bu kısıtlar sonunda mutlaka sağlanmalıdır fakat kaynak seçimi ve çözelgeleme oluşturma sırasında geçici olarak ihlalleri tolere edilebilir.
- ESNEK: Normalde ihlal edilebilir. Kısıtlar kümesi tercihen sağlanmalıdır, bu kısıtların tümünün sağlanması mümkün değildir.

ESNEK-SERT ve ESNEK kısıtlar ihlal edilebilir ve ihlal edildikleri ölçüde ceza puanı atanmalıdır. Bu nedenle her bir çözümü sağlamak için (d,s) puan numarası verilir. Burada;

*d*: ESNEK-SERT kısıtların ihlal sayısına denk olarak verilen ceza puanıdır. Çözüm uzayındaki uzaklık uygunluğu olarak adlandırılır.

*s*: ESNEK kısıtların ihlalleri için ceza puanıdır.

Bu hiyerarşi kavramında herhangi bir seviyedeki çözümü geliştirmede önceliği düşük seviyedeki kısıtları sağlamak alır. Puanlar açısından ilk önce *d* değeri alınır, *d* değeri eşit ise *s* puanları karşılaştırılır. Toplam sıralama şu şekildedir:

$$(d_1, s_1) > (d_2, s_2) \text{ eğer } d_1 > d_2 \text{ veya } (d_1 = d_2 \text{ and } s_1 > s_2) \quad (8.1)$$

Uygun bir çözüm SERT ve ESNEK-SERT kısıtlarının sağlanmasıdır. Böylece bütün uygun çözümlerde  $d=0$ 'dır ve esnek kısıtların puanları olan  $s$  puanları karşılaştırılır.

Tabiki, denklem (8.1) tarafından sağlanan sıralamadaki eşitlemede tek amaç  $Md+s$  dir,  $M$  yeterli büyüklükte herhangi bir ağırlıktır,  $d$  nin herhangi bir gelişimi her zaman iyidir ve  $s$  nin değişiminden bağımsız bir değerdir (Büyük ağırlıklar problem için bir risk taşır ve en iyi uygulama değildir) . Bu metod uygulaması iki ESNEK kısıt seviyesinin kullanımına uygundur. Tipik olarak literatürde güçlü ve zayıf kısıtlar olarak adlandırılırlar ve matematik programlama değildirler. Bunlar üç seviyede ifade edilebilir 'sert', 'yüksek-cezalı esnek', ve 'normal esnek'. Ancak, burada ESNEK-SERT terminolojisinin 'yüksek-cezalı esnek'e karşı gerçek dünya problemlerinde daha iyi bir metod olduğu çalışmalarda belirtilmiştir (McCollum vd., 2012). Bu terminoloji: Uygulanabilir çözümler genellikle herhangi bir değişiklik ya da ayar gerektirmeden pratikte uygulanabilir olanlar anlamına gelir. Sadece ESNEK kısıtların ihlali olursa, o zaman bazı insanlar çözümün parçaları ile biraz mutsuz olsa bile, bir bütün olarak çözüm olan herhangi bir özel müdahale olmadan kullanılabilir. SERT ve ESNEK-SERT kısıtların her ikisinin de kullanılabilir olması uygun çözüm için yerine getirilmesi gereken anlamında gerçekten zordur. Ancak, gerçek dünya çizelgeleme problemleri genellikle kararlar genellikle bir takvim oluşturmak ve aynı zamanda kaynak (örneğin, müsait odaları ve zaman dilimi gibi) hakkında bilgilendirmek için yapılan ikili bir süreçtir. Kaynaklar biraz yetersiz olduğu durumda o zaman sorunu çözmek imkânsız olabilir. Çözüm imkânsız olduğunda nasıl bir yol izleneceği kullanıcılar açısından önemlidir.

#### 8.4. Sınav Çizelgeleme Modelinin Detayları

Yeni bir model tanımlamak için, ITC2007 yarışmasındaki bilgisayar kullanılmıştır. Burada 'SERT, 'ESNEK-SERT' ve 'ESNEK' kısıtlar kullanıcılar için sunulmuştur.

Kısıtları sert veya esnek şeklinde sınıflandırmak için uzaklık uygunluğu terimi tanımlanmıştır.

#### **8.4.1. Sert kısıtlar**

Bütün çözümler bu kısıtları sağlamalıdır:

- Bütün sınavlar gerçekleştirilmelidir. Çizelgeleme problemleri bir sınavı yapmamak için bir neden değildir.
- Sınavlar periyodlar arasında bölünmemelidir.
- Sınavlar iki derslik arasında bölünmemelidir.
- Bütün sınavlar sadece bir periyoda ve bir dersliğe atanmalıdır.

#### **8.4.2. Esnek-Sert kısıtlar**

- Çakışan sınavlar aynı periyoda atanmamalıdır (İki sınav çakışması iki sınavıda alan bazı öğrenciler olması durumunda oluşmaktadır).
- Her derslik için ve her periyoda derslik kapasitesi aşılmamalıdır.
- Bir periyoda atanan herhangi bir sınav petiyort süresinden daha fazla süre kullanamaz.

#### **8.4.3. Periyot ve derslik ilişkili: esnek kısıtlar**

- ‘Sınavın Sonra Yapılması’: 2. sınav 1. sınavdan sonra gerçekleşmelidir.
- ‘Sınav Birleştirilmesi’: birinci sınav ile ikinci sınav aynı zamanda gerçekleşmelidir.
- ‘Sınav Ayrılması’: 1.sınav ile 2.sınav aynı zamanda gerçekleşemez.
- Derslik-İlişkili: ‘Özel Derslik’: Sınavlar sadece kendi dersliklerine atanmalıdır.

#### **8.4.4. Esnek kısıtların hesaplanması**

**Aynı günde iki sınavın arka arkaya atanması:** Bu hesaplamada aynı günde iki sınavın arka arkaya gerçekleşmesidir, burada arka arkaya gerçekleştirilen iki

sınavı da alan öğrenci sayısı dikkate alınır. Bu işlemden sonra, öğrenci sayısı 'Kurumsal Model İndeksi' içindeki ağırlık ile 'arka arkaya iki sınav'da sağlanan sayıya göre toplanır ve çarpılır. Eğer bir öğrenci bir günün en son periyodunda sınava girerse bir sonraki günün ilk periyodu 'arka arkaya iki sınav' cezasına tabi değildir.

**Aynı günde iki sınav atanması:** Bir günde üç veya daha fazla periyot varsa bu durum gerçekleşir, bir günde iki sınavı olan öğrenci varsa fakat sınavlar arka arkaya olmayacak şekilde ise bu hesaplama yapılır. Toplam sayısı daha sonra 'Kurumsal Model İndeksi' içinde sağlanan 'bir günde arka arkaya iki sınav' ağırlığı ile çarpılır. Bu nedenle, bir gün içinde iki sınav aralarındaki en az bir serbest periyot süresi vardır, yani bitişik olmayan sınavlar olarak kabul edilmektedir. Bu bir çözüm içinde yerleştirerek belirli bir sınav için cezanın iki kez hesaplanmaması için yapılır. İki sınav aynı gün bitişik dönemlerde olsaydı Örneğin, ceza sadece 'arka arkaya iki sınav'ın bir parçası olarak kabul edilecekti.

**Süre yayılımı:** Bu esnek kısıt belirli bir periyot sayısı üzerinde bir bireyin sınavlarını 'yaymak' için kullanılır. Bu kısıt daha önce açıklanan iki esnek kısıtın bir uzantısı olarak düşünülebilir. 'Kurumsal Model İndeksi' içinde, bir çözümün kaç periyoda yayılmış olması gerektiğine ilişkin bir rakam 'periyot yayılımı' (Periodspread) anahtar sözcüğü ile sağlanır. Bu rakamın yüksek olması, potansiyel olarak öğrenciler için daha iyi sınav yayılmasını sağlar. Ticari olarak, bu ayarı değiştirirken oluşturulan çözümler kurumun çok daha memnun olduğu çizelgeler oluşmasına yol açmıştır. Eğer, 'Kurumsal Model İndeksi' içinde 'periyot yayılımı', 7 olarak belirlenmişse, her sınav için tüm kayıtlı öğrencilerin durumları hesaplanır sonra diğer sınavları 7 periyot içerisinde gerçekleşmelidir böylece istenen süreye sınavlar yayılmış olur. Bu puan toplam bütün cezaya eklenir. Burada 'arka arkaya iki sınav' ve 'bir günde iki sınav' için hesaplanan cezaya ekleneceği unutulmamalıdır. Çözüm içindeki tek bir olay iki kez cezalandırılırsa da, çözüm kalitesinin bir göstergesi olarak genellikle gereklidir.

**Farklı süreli sınavlar:** Bir derslik ve periyot için karışık sürelerde sınavların atanması durumunda bir ceza uygulanır. Burada amaç eşit süresi olan sınavları birlikte sağlamaya çalışmaktır. Cezanın hesaplanmasında, 'Kurumsal Model

İndeksi' karışık süresi bileşeni olan sınavların ihlalleri (mixed duration component) sayısına göre hesaplanır.

**Öğrenci yoğunluğu:** Öğrenci sayısı çok olan sınavların sınav dönemi başında sınav oturumunun yapılması genellikle arzu edilir. Bunu dikkate almak için öğrenci yoğunluğu (frontload) ifadesi tanımlanmıştır. 'Kurumsal Model İndeksi' içinde 'frontload' ifadesinin üç parametresi vardır örneğin (100, 30, 5). Bunlardan ilki dikkat edilmesi gereken en büyük sınavların sayısıdır. En büyük sınav sınıf mevcudu tarafından belirlenir. İkinci parametre son periyotların sayısıdır ve büyük sınavlardan kaçınılmalıdır. Üçüncü parametre esnek kısıtı ihlal edince her zaman eklenmesi gereken ceza puanı veya ağırlıktır. Bu kısıtın sağlanması kurumun erken sınav oturumlarına büyük sınavları atamasına izin verir. Bu kısıt uygulamada çok popülerdir, çünkü çok öğrencisi olan sınavları atamak zordur ve uzun sürer.

**Esnek mekân cezası:** Kuruluşlar genellikle belirli mekânların en az kullanılmasını isterler. Karışık süreli toplam cezanın bileşeni gibidir, toplam ceza puanı bu bölümünde periyot bazında bir periyodda hesaplanmalıdır. Her bir periyodda, eğer çözüm içinde kullanılan bir mekân ile ilişkili bir ceza varsa, o periyotdaki bu derslik için ceza puanı bu dersliği kullanan sınav sayısı ile çarpılarak hesaplanır.

**Esnek süre cezası:** Kurumlar genellikle belirli periyodların kullanımını minimumda tutmak isterler. 'Karışık Süreli' ve 'Mekan Cezası' toplam ceza bileşenleri gibidir, toplam cezanın bu kısmı bir periyodu baz alarak periyod üzerinde hesaplanmalıdır. Her periyod için, bu ceza süre içindeki sınav sayısına göre ilgili ceza ile çarpılarak hesaplanır. Bu kısıtlar oldukça karmaşıktır. Problem formülasyonunda olası karmaşıklığı azaltmak için matematiksel programlama tanımları yapılmıştır.

**Problem formülasyonu:** Problemin formül ile ifade edilmesi tasarımda bütünlük ve açıklık sağlar. Kodlama ve matematiksel hesaplamalarda en iyi çözümün bulunması için işlemleri kolaylayacaktır. Ancak yarışmada sağlanan

örneklerde hala küçük sorunlar vardır (Burke vd. 2008; Parkes ve Özcan 2010). Onlar gelecekte kodları geliştirmek ve değiştirmek için temel olabilir.

Problem tanımında gerçek dünya kısıtları vardır ve bu nedenle doğal olarak oldukça uzun ve karmaşıktır. Bu kaçınılmaz belirsizlikler ve yanlış anlamalar kolayca uygulamaların içinden çıkamama etkisi yaratır. Buna göre, yarışma için, üç ayrı çözüm doğrulayıcılar uygulanmış ve sonuçlar çapraz kontrol edilmiştir. Bu doğrulayıcılar doğrudan ve bağımsız olarak kodlanmıştır. Bu bölümde verilen formülasyon ile kodlamaya dayalı dönem ve derslik atamaları belirtilerek birincil karar değişkenleri için çözüm verilmiştir. Bu durumda, kolayca tamsayı programlama çözümleri çözümün uygun olup olmadığını belirlemek ve aynı anda çeşitli cezalar için değerleri üretmek amacıyla kullanılabilirler.

Bu kapsamlı tamsayı programlama formülasyonu ile doğrudan uygulama sonuçlarını çapraz kontrol etme imkanı olmaktadır. Burada sunulan karmaşık herhangi bir model belirsizlik, yanlış anlama ya da önemli ölçüde basit insan hatası artışı engellemek için gereklidir.

Problem modelinde sunum kolaylığı için, ESNEK-SERT kısıtlar normal sert kısıtlar gibi ele alınmıştır. Veri örneklerine göre SERT ve ESNEK-SERT kısıtları sağlamak çözümde mümkündür. Aynı zamanda yüksek cezaya sahip esnek kısıtları ESNEK-SERT kısıtlarına dönüştürmek kolaydır.

Aşağıdaki formülasyonları daha okunabilir hale getirmeye yardımcı olmak için izlenecek yollar aşağıdaki gibidir:

- Kümeler (sınavların vb.) büyük harfle gösterilecek.
- Parametreler (değeri girdi dosyalarından alınan) küçük harfle gösterilecek.
- Değişkenler (değeri arama esnasında belirlenen) büyük harfle gösterilecek.
- İki farklı tür arasındaki bağlantıyı büyüklük olarak göstereceğimiz zaman örneğin sınav boyutu, derslik boyutu gibi miktarları, yerine simgeler her zamanki sık kullanılan bir simge alt simge ile gösterir. Örneğin  $s^E$  ve  $s^R$  sırasıyla sınav ve derslik boyutları için kullanılır.

**Kümeler ve parametreler:** Aşağıdaki kümeler ve parametreler doğrudan giriş dosyasında verilen, ya da ondan kolayca elde edilecek şekilde verilmiştir.

### Sınav ile ilgili parametreler

- E Sınavların kümesi
- $s_i^E$   $i \in E$ . E kümesinin elemanı olan i. sınavın boyutu
- $d_i^E$   $i \in E$ . E kümesi olan i. sınavın süresi
- $f_i^E$  Eğer i.sınav öğrenci yoğunluğu (frontload) cezası varsa 1 yoksa 0'dır.
- D  $\cup_i d_i^E$ . Kullanılan sınav sürelerin kümesidir
- $u_{id}^D$  Süre tipi D kümesinin elemanı olan d ise 1 dir. Sürenin nekadardığı önemli değildir sadece eşit olup olmadıkları önemlidir.

Örneklerdeki girdi dosyasında öğrenci yoğunluğu (frontload) girdisi 3 parametredir. İlk parametre büyük sınavların sayısıdır ve sınavların büyüklüğü sınıf sayısı ile karakterizedir. Bu öğrenci yoğunluğu (frontload) cezası seçilen sınavlar için kullanılır, bu i. sınav için  $f_i^E = 1$  dir. Sınavlar büyükten küçüğe doğru sıralanır, eşit büyüklüğe sahip sınavlar ise ilk indekstekine göre sıralanır. Belirli sınav sayıları sıralı listeden alınır ve  $f_i^E = 1$  verilir kalan diğer sınavlar için  $f_i^E = 0$  dır.

Süre tipi  $u_{id}^D$  karışık süreli olmayan cezalar için kullanılır. Örneğin, bütün sınavlar 120 veya 180 dakika ise burada iki süre tipi vardır ve  $d \in \{0, 1\}$   $u_{id}^D = 1$  sadece 120 dakikalık sınav için  $u_{id}^D = 0$  sadece 180 dakikalık sınav içindir.

### Öğrenciler ile ilgili parametreler

- S Öğrencilerin kümesi.
- $t_{is}$  Eğer s öğrencisi i. Sınavı alıyorsa 1 dir aksi halde 0'dır.

### Derslik ile ilgili parametreler

- R Dersliklerin kümesi.

$s_r^R$   $r \in R$ .-derslik büyüklüğü  
 $w_r^R$   $r$  dersliğinin ceza ağırlığı.

### Süre ile ilgili parametreler

$P$  Sürelerin kümesi.  
 $d_p^P$   $p \in P$ .  $p$  zaman diliminin süresi.  
 $f_p^P f_p^P = 1$  Eğer bir periyodun frontload cezası varsa 1 yoksa 0'dır.  
 $w_p^P$   $p$  süresinin ceza ağırlığı.  
 $y_{pq}$   $P$  ve  $q$  süreleri aynı günde ise 1 değilse 0'dır.

### Süre ile ilgili sert kısıtlar

Bunlar SONRA YAPILAN SINAV, SINAV BİRLEŞTİRME ve SINAV AYIRMA kısıtlarıdır.

$H^{aft}$  Sınav çiftleri kümesidir.  $e_1 e_2$  den sonra atanmalıdır.  
 $H^{coin}$  Sınav çiftleri kümesidir.  $e_1 e_2$  ile aynı periyodda gerçekleşmelidir  
 $H^{excl}$  Sınav çiftleri kümesidir.  $e_1 e_2$  aynı periyodda gerçekleşmemelidir

### Derslik ile ilgili sert kısıtlar

#### Özel derslik kısıtı

$H^{sole}$  Sınav kümesidir. Bir periyoda ve bir dersliğe sadece bir sınav atanmalıdır.

### Kurumsal ağırlıklar ve parametreler

$w^{2R}$  "aynı günde arka arkaya sınav" ağırlığı  
 $w^{2D}$  "aynı günde iki sınav" ağırlığı  
 $w^{PS}$  "süre yayılımı" ağırlığı  
 $w^{NMD}$  "farklı süreler " ağırlığı  
 $w^{FL}$  "öğrenci yoğunluğu" ağırlığı  
 $g$  Bir öğrenci için sınavlar arasındaki minimum boşluk değeri.

## Değişkenler

Birincil Değişkenler

$$X_{ip}^P = \text{sınav } i \text{ periyod } p \text{ de ise } 1, \text{ değil ise } 0 \quad (8.2)$$

$$X_{ir}^R = \text{sınav } i \text{ derslik } r \text{ de ise } 1, \text{ değilse } 0 \quad (8.3)$$

İkincil Değişkenler

$$X_{ipr}^{PR} = 1 \text{ eğer sınav } i \text{ periyot } p \text{ ve derslik } r \text{ de ise } 1, \text{ değilse } 0 \quad (8.4)$$

$$\forall i \in E, \forall p \in P, \forall r \in R \quad X_{ipr}^{PR} \equiv X_{ip}^P X_{ir}^R \quad (8.5)$$

Ancak, esnek kısıt ihlal cezaları negatif olmayan değişkenler olarak tanımlanmıştır.

$C_s^{2R}$  = s öğrencisi için “arak arkaya iki sınav” cezası

$C_s^{2D}$  = s öğrencisi için “bir günde iki sınav” cezası

$C_s^{PS}$  = s öğrencisi için “süre yayımlı” cezası

$C^{NMD}$  = “karışık süreli olamayan” cezası

$C^{FL}$  = “Önyüklü” cezası

$C^P$  = “Esnek zaman” cezası

$C^R$  = “Esnek oda” cezası

Bu değişkenlerin çoğu gerekli olmakla birlikte, doğrudan amaç fonksiyonuna dahil edilebilir. Çünkü bu değişkenlerin değerleri yarışma web sitesinde doğrulayıcı olarak verilen değerlere uymalıdır.

$$U_{dpr}^D = 1 \text{ Süre tipi } d \text{ p periyodunu ve } r \text{ dersliğini kullanıyorsa } 1 \text{ değilse } 0 \quad (8.6)$$

Amaç Fonksiyonu

$$\sum_{s \in S} (w^{2R} C_s^{2R} + w^{2D} C_s^{2D} + w^{PS} C_s^{PS}) + w^{FL} C^{FL} + C^P + C^R \quad (8.7)$$

$C^R$  ve  $C^P$  cezaları periyot ve derslik için ağırlıkları ayrılmamış olup, onlarla ilgili ağırlıklar onların tanımlarında bulunmaktadır. Tabiki, bu problem multiobjective dir ancak, ağırlıklar toplamı yaklaşımı kolaylık için kullanılmıştır. Burada öğrenci kümeleri için cezaların tanımını yapılmıştır.

$$C^{2R} = \sum_{s \in S} C_s^{2R} \quad (8.8)$$

$$C^{2D} = \sum_{s \in S} C_s^{2D} \quad (8.9)$$

$$C^{PS} = \sum_{s \in S} C_s^{PS} \quad (8.10)$$

Minimizasyon sert kısıtların sağlanması konusunu ve esnek kısıtların ceza tanımları konusudur.

### Sert kısıtları

Her sınav en az bir periyoda ve en az bir dersliğe atanmalıdır.

$$\forall i \in E \sum_{r \in R} x_{ir}^R \geq 1 \quad (8.11)$$

$$\forall i \in E \sum_{p \in P} x_{ip}^P \geq 1 \quad (8.12)$$

Her sınav en fazla bir dersliğe atanabilir

$$\forall i \in E \sum_{r \in R} x_{ir}^R \leq 1 \quad (8.13)$$

Her sınav en fazla bir periyoda atanabilir

$$\forall i \in E \sum_{p \in P} x_{ip}^P \leq 1 \quad (8.14)$$

X karar değişkenleri aşağıdaki gibi birbirine bağlanır:

$$\forall i \in E \forall p \in P X_{ip}^P = \sum_{r \in R} x_{ipr}^{PR} \quad (8.15)$$

$$\forall i \in E \forall r \in R X_{ir}^{PR} = \sum_{p \in P} x_{ipr}^{PR} \quad (8.16)$$

### Esnek-sert kısıtları

Derslik kapasitesi hiç bir zaman aşılmamalıdır.

$$\forall p \in P \forall r \in R \sum_{i \in E} s_i^E X_{ipr}^{PR} \leq s_r^R \quad (8.17)$$

Periyot süreleri hiçbir zaman aşılmamalıdır.

$$\forall p \in P \forall i \in E d_i^E X_{ip}^P \leq d_p^P \quad (8.18)$$

Herhangi bir periyodda, herhangi bir öğrenci en fazla bir sınava girebilir.

$$\forall p \in P \forall s \in S \sum_{i \in E} t_{is} X_{ip}^P \leq 1 \quad (8.19)$$

Sınavlar arasında genellikle çakışma matrisi oluşur. Sabit periyod kısıtları zorunludur.

$$\forall (i, j) \in H^{aft} \forall p, q \in P, \text{ with } p \leq q$$

$$X_{ip}^P + X_{jq}^P \leq 1 \quad (8.20)$$

$$\forall (i, j) \in H^{coin} \forall p \in P,$$

$$X_{ip}^P = X_{jp}^P \quad (8.21)$$

$$\forall (i, j) \in H^{excl} \forall p \in P,$$

$$X_{ip}^P + X_{jp}^P \leq 1 \quad (8.22)$$

Sabit oda kısıtları zorunludur:

$$\forall i \in H^{sole} \forall j \in E, j \neq i \forall p \in P, \forall r \in R,$$

$$X_{ip}^P + X_{ir}^R + X_{jp}^P + X_{jr}^R \leq 3 \quad (8.23)$$

### Esnek kısıtlar

Son olarak, esnek kısıtların ihlallerine karşılık gelen ceza terimlerini incelenecektir. Arka arkaya iki sınav  $C^{2R}$ , bir günde iki sınav  $C^{2D}$  ve süre yayılması  $C^{PS}$  cezaları, her öğrenci için kayıt dizileri kısıtlamalar kaynaklanan bu terimleri için dönem cezaları kullanacaktır. Bu cezalar özellikle kodlamak için yanlış anlamaya karşı hassastır: Bu nedenle, netliği geliştirmek için öncelikle doğrusal olmayan kısıtlamalar olarak sunulup, ve sonra eşdeğer doğrusal kodlama verecektir.

### Aynı gün içerisinde arka arkaya sınav

Eğer s öğrencisi iki farklı i ve j sınavlarına aynı günde arka arkaya periyotlarda giriyorsa  $C_s^{2R}$  cezası 1 artırılır.

$$C_s^{2R} = \sum_{\substack{i,j \in E \\ j \neq i}} \sum_{\substack{p,q \in P \\ q=p+1 \\ y_{pq}=1}} t_{is} t_{js} X_{ip}^P X_{jq}^P \quad (8.24)$$

$q=p+1$  ve  $j \neq i$  ifadeleri kullanılır.  $X_{ip}^P X_{jq}^P$  terimi i ve j sınavlarını ortak alan öğrenciler ile ilgilidir ve sadece çakışan sınav çiftleri için düşünülmüştür. Çakışma matrisindeki kenar bağlantıları E sınav kümesi için  $G_c=(E,A)$  ile gösterilir. A kümesi yönsüz kenarlardır, (i,j) kenarlarından oluşur. Sadece i ve j sınavları çakışmasıdır. Genelliği kaybetmeden ve çift hesaplamayı önlemek için bütün  $a=(i,j) \in A$  için  $i < j$  olarak farzedilmiştir. A ağırlığı  $w_a^C$  A daki her bir kenar için her iki sınavı alan öğrenci sayısıdır.

$$\forall a = (i, j) \in A \quad w_a^C = \sum_{s \in S} t_{is} t_{js} \quad (8.25)$$

Çakışma grafindeki her bir kenarın toplam ağırlığını integer kodlamak için yeni bir değişken tanımlanmıştır:

$$C_a^{2R} = \begin{cases} 1 & \text{if edge } a = (i,j) \text{ exam } i \text{ ve } j \text{ için two-in-row cezası} \\ 0 & \text{diğer durumlarda} \end{cases}$$

Toplam cezadan sonra öğrenci sayıları tarafından ağırlıklandırılan durumların cezaları toplanır.

$$C^{2R} = \sum_{a \in A} w_a^C C_a^{2R} \quad (8.26)$$

Amaç fonksiyonu içerisindeki  $C^{2R}$  nin minimizasyonu  $C_a^{2R}$  yı otomatik olarak 0 a gitmeye zorlar. Bu yüzden aşağıdaki lineer kısıtlama kullanılarak gerektiğinde kalanlar için zorunludur.

$$\forall a = (i, j) \in A \forall p, q \in P \mid q - p = 1 \ \& \ y_{pq} = 1$$

$$X_{ip}^P + X_{jq}^P \leq 1 + C_a^{2R} \quad (8.27)$$

### Aynı gün içerisinde iki sınav

Eğer s öğrencisi i ve j gibi iki farklı sınava aynı günde fakat arka arkaya olmayacak şekilde atanırsa  $C_s^{2D}$  cezası 1 artırılır.

$$C_s^{2D} = \sum_{\substack{i,j \in E \\ j \neq i}} \sum_{\substack{p,q \in P \\ q > p+1 \ \& \ y_{pq}=1}} t_{is} t_{js} X_{ip}^P X_{jq}^P \quad (8.28)$$

Bu arka arkaya iki sınav kısıtı gibidir fakat  $q=p+1$  yerine  $q>p+1$  kullanılır.

Lineer forma çevirmek için yeni bir değişken tanımı:

$$C_a^{2D} = \begin{cases} 1 & \text{if edge } a = (i, j) \text{ exam } i \text{ ve } j \text{ için two-in-day cezası} \\ 0 & \text{otherwise} \end{cases}$$

$$C^{2D} = \sum_{a \in A} w_a^C C_a^{2D} \quad (8.29)$$

$$\forall a = (i, j) \in A \forall p, q \in P \mid q - p \geq 2 \& y_{pq} = 1$$

$$X_{ip}^P + X_{jq}^P \leq 1 + C_a^{2D} \quad (8.30)$$

### Süre yayılımı

Eğer s öğrencisi iki farklı i ve j sınavlarına farklı periyotlarda giriyorsa ve j i den sonra ise fakat g boşluğu içerisinde ise,  $C_s^{PS}$  cezası 1 artırılır. Yine çift sayma sınavlar sıraya konarak önlenabilir.

$$C_s^{PS} = \sum_{\substack{i, j \in E \\ j \neq i}} \sum_{\substack{p, q \in P \\ p < q \leq p+g}} t_{is} t_{js} X_{ip}^P X_{jq}^P \quad (8.31)$$

Bu yine arka arkaya iki sınav veya bir günde iki sınav gibidir fakat p ve q periyotlarında q periyodu  $q \in \{(p + 1), \dots, (p + g)\}$  ile değiştirilir. Lineer forma çevirmek için yeni değişken tanımlanır:

$$C_a^{PS} = \begin{cases} 1 & \text{eğer kenar } a = (i, j) \text{ i ve j sınavları için süre yayılımı cezası varsa} \\ 0 & \text{değilse} \end{cases}$$

Kısıtlar zorunludur

$$C^{PS} = \sum_{a \in A} w_a^C C_a^{PS} \quad (8.32)$$

$$\forall a = (i, j) \in A \forall p, q \in P \mid 1 \leq |q - p| \leq g$$

$$X_{ip}^P + X_{jq}^P \leq 1 + C_a^{PS} \quad (8.33)$$

### Farklı süreler

Burada  $U_{dpr}^D$  nin 0 olmaması durumu vardır, bazı sınavlar periyot p ve derslik r yi kullanırken d süre tipinde olabilirler.

$$\forall d \in D \forall i \in E \text{ with } u_{id}^D = 1 \forall p \in P \forall r \in R$$

$$U_{dpr}^D \geq X_{ip}^P + X_{ir}^R - 1 \quad (8.34)$$

Periyot-Derslik çifti pr negatif ceza değeri alamaz,  $C_{pr}^{NMD}$  maximum 0 olur ve kendine verilen süre toplam süre tipi sayısının üzerinde olmalıdır.

$$\forall p \in P \forall r \in R$$

$$1 + C_{pr}^{NMD} \geq \sum_{d \in D} U_{dpr}^D \quad (8.35)$$

$$C_{pr}^{NMD} \geq 0$$

Toplam ceza

$$C^{NMD} = \sum_{p \in P} \sum_{r \in R} C_{pr}^{NMD} \quad (8.36)$$

Toplam amaç fonksiyonundaki minimizasyon  $U_{dpr}^D$  ve  $C_{pr}^{NMD}$  minimal değere çekilmeye çalışılmalıdır.

### Öğrenci yoğunluğu

Öğrenci yoğunluğu cezası i. Sınava uygulanırsa  $f_i^E = 1$  olur ve  $f_p^P = 1$  periyodlara atanır

$$C^{FL} = \sum_{i \in E} \sum_{p \in P} f_i^E f_p^P X_{ip}^P \quad (8.37)$$

Esnek Süre cezası

$$C^P = \sum_{p \in P} \sum_{i \in E} w_p^P X_{ip}^P \quad (8.38)$$

Esnek Derslik cezası

$$C^{PR} = \sum_{r \in R} \sum_{i \in E} w_r^R X_{ir}^R \quad (8.39)$$

### Veri örnekleri

Bütün veri örnekleri yarışmada kullanılmıştır ve çeşitli Avrupa, Avustralya ve Amerika üniversitelerinden alınmıştır. Ülkeler içinde Birleşik Krallık, Portekiz, Avustralya ve Amerika vardır ve bu yüzden büyük farklı sistemleri sunulmuştur. Veriler (ve kurum adı) yarışmada kullanılmak amacı ile anonimleştirilmiştir. 4 veri seti ile başlanmış ve 4 tanesi de yarışmadan iki hafta önce verilmiştir, 4 tanesi de gizli veri setleridir. Önce tanımlanan herhangi bir çözüm için bir veya daha fazla sert veya gerekli kısıtların gerçekleşmesi mümkün değildir. Ancak yarışmada uygun çözümler bulunmuştur. Bu şaşırtıcı değildir çünkü bu veriler gerçek dünya problemidir örneğin, kaynaklar zaten uygun çözümler için seçilmiştir. Bu yüzden sonradan kullanılan verilerdeki uygunluk mesafesi sonuçlar üzerinde direkt etkili değildir, çözümçüler bu verileri bilmemektedir ve bu yüzden uygunluk mesafesi azaltılmıştır.

### Modelin genişletilebilirliği

Önceli sunulan modellere göre yukarıdak sunulan model genişletilebilir özellikler taşımaktadır. Örneğin; süre yayılım veya öğrenci yoğunluğu gibi kısıtlar genişletilebilir ve tanımları çoğaltılabilir. Modele yeni eklenen kısıtlar ayrı ayrı hesaplanır ve ilgili ceza puanları toplanır.

Yukarıda anlatılan kısıtların Süleyman Demirel Üniversitesi örneklerine uygulanmış şekli Çizelge 8.1 ve 8.2 'de gösterilmektedir.

Çizelge 8.1. Üniversite Örneklerinin Özellikleri

| Problem | Öğrenci | Sınav | Derslik | Süre | Çakışma<br>ağırlığı | Zaman<br>Sert Kısıtı | Derslik<br>Sert Kısıtı |
|---------|---------|-------|---------|------|---------------------|----------------------|------------------------|
| SDU01   | 10953   | 212   | 17      | 50   | 3.24                | 142                  | 0                      |
| SDU02   | 11012   | 236   | 26      | 61   | 5.08                | 123                  | 0                      |
| SDU03   | 24867   | 430   | 29      | 80   | 1.37                | 317                  | 0                      |
| SDU04   | 8028    | 166   | 18      | 59   | 12.6                | 61                   | 0                      |
| SDU05   | 12091   | 269   | 33      | 46   | 3.59                | 173                  | 0                      |

Çizelge 8.2. Üniversite Örneklerinin Kısıt Özellikleri

| Problem | Arka<br>Arkaya<br>iki<br>sınav<br>cezası | Aynı gün<br>içerisinde<br>iki sınav<br>cezası | Süre<br>Yayılım<br>Cezası | Karışık<br>Sürelili<br>Olmayan<br>Cezalar | Büyük<br>Sınavların<br>Sayısı | Son zaman<br>dilimi<br>önlem | Önyüklü<br>Cezalar |
|---------|------------------------------------------|-----------------------------------------------|---------------------------|-------------------------------------------|-------------------------------|------------------------------|--------------------|
| SDU01   | 15                                       | 5                                             | 15                        | 25                                        | 100                           | 30                           | 10                 |
| SDU02   | 20                                       | 5                                             | 20                        | 25                                        | 25                            | 30                           | 15                 |
| SDU03   | 7                                        | 5                                             | 5                         | 10                                        | 100                           | 30                           | 5                  |
| SDU04   | 35                                       | 10                                            | 5                         | 5                                         | 25                            | 5                            | 10                 |
| SDU05   | 15                                       | 5                                             | 5                         | 10                                        | 50                            | 30                           | 5                  |

## 9. DERS ÇİZELGELEME PROBLEMİ

Ders çizelgeleme problemi (DÇP) araştırmacıların çözmek için yıllardır üzerinde uğraştığı zor bir optimizasyon problemidir. Bu yüzden polinom zamanda bütün örnekleri çözebilecek bir deterministik algoritma yoktur. Bu problem oldukça karmaşık ve kısıt sayısı çok olan bir problemidir. Örneğin, uygun, iyi ve kabul edilebilir bir çizelge bir üniversiteden diğerine farklılık gösterir. Çünkü farklı üniversitelerin sağlanması gereken farklı kısıtları vardır. Araştırmacılar tarafından çeşitli teknikler önerilmesine rağmen bazı veriler için başarılı olan yaklaşım diğerleri için iyi bir sonuç üretememektedir.

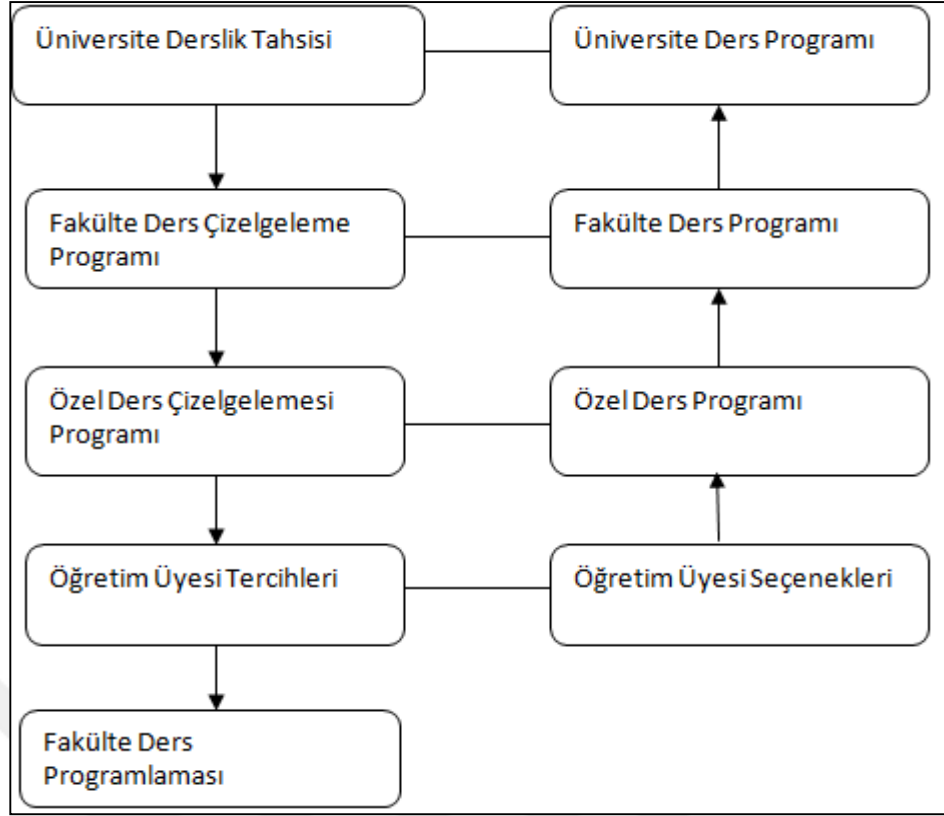
Ders çizelgeleme yapısı üniversite çizelgeleme problemleri arasında en karmaşık ve zor olan problemidir. Bu karmaşıklığa birçok faktör sebep olmaktadır. Bunlar arasında sağlanması gereken kısıtlar vardır. Örneğin; hiçbir öğrenci aynı anda iki derse katılamaz ve her bir ders bir dersliğe atanmak zorundadır (Bykov, 2003). Bu yüzden elle ders çizelgesi oluşturma oldukça zaman alıcı, hata yapmaya açıktır ve bütün kısıtları sağlamayı garanti etmez. Problemin karmaşıklığından dolayı birçok araştırmacı üzerinde çalışmıştır. Yerel arama algoritmaları, problem küçük parçalara ayrılırsa, çizelgeleme problemleri çözümünde oldukça başarılıdır. Carter (1983) büyük verileri daha küçük problemlere ayırarak bir ayrıştırma yöntemi kullanan yerel arama algoritması önermiştir. Son yıllarda bazı araştırmacılar çok etmenli sistemleri kullanmayı önermişlerdir. Otonom davranışlar sergileyen etmenler, kendi bilgi ve bireysel yeteneklerini kullanarak kullanıcıların çözemedikleri veya etkin bir biçimde çözemeyeceklerini düşündükleri problemleri çözmek amacıyla bir araya gelerek Çok Etmenli Sistemler (Multi Agent Systems) adı verilen sistemleri oluşturmaktadırlar. Çok etmenli sistemler dağıtık ve otonom ajanlardan oluşan otonom sistemlerdir ve yerel ve küresel hatalara karşı dayanıklıdırlar. Çok etmenli sistemler personel çizelgeleme, e-ticaret ve üretim çizelgeleme gibi birçok probleme uygulanmıştır. Fakat üniversite çizelgeleme problemleri için çok fazla kullanılmamışlardır. Kaplansky vd. (2004) çok etmenli sistemler kullanılarak dağıtık yapıda olan çizelgeleme problemlerinin çözülebileceğini iddia etmişlerdir. Her bir etmen, en iyi çözümü bulmak için farklı özelliklere sahiptir. Çizelge oluşturulurken etrafa dağıtılan bütün etmenler paylaşılan kaynaklardaki çakışmaları önlemek için

iletiřim halinde olmak zorundadırlar. Kaplansky vd. (2004) çok etmenli modeli özellikle hemřire nbet izelgeleme problemi iin incelemiřlerdir. Bu tezde, özellikle st-sezgisel algoritmalarla problem zlmeye alıřılacaktır. st-sezgiseller meta-sezgisellerin daha genel kullanılabilmesi iin geliřtirilen yeni arama yntemleridir. st-sezgisellerin amacı arama yntemini problem-bağımsız hale getirmektir. Burke vd. (Burke, 1999), geliřtirilen algoritmaların rneęin; genetik algoritmalar, byk veriye sahip karmařık izelgeleme problemlerini tam olarak zemedięini nermiřlerdir. Bu algoritmalar kk veri setlerini zmesine raęmen byk veri setlerini zememektedir. Sonu olarak Burke vd. problemi dha kk paralar ayırarak zmeye alıřmıřlardır.

### **9.1. Ders izelgeleme Problemi: Modeli ve Problem Tanımı**

Sleyman Demirel niversitesi de dięer niversiteler gibi her yıl iki kez ders izelgeleme problemi ile karřı karřıya kalmaktadır. 2007 yılında dzenlenen 2. Uluslararası izelgeleme Yarıřması (ITC2007) eęitimsel izelgelemeler iin yeni gerek dnya problemleri nermiřtir. Bu yarıřmadaki izelgeleme kategorilerinden birisi de mfredat tabanlı izelgelemedir.

izelgeleme, verilen kaynaklardaki kısıtlara baęlı olarak mmkn olan en iyi sonuca ulařmak iin, bir dizi yer ve zaman olaylarını “atama iřlemi” olarak tanımlanabilir. Eęitim alanlarının verimli kullanılıp kullanılmadıęını standartlara gre karřılařtıran eęitim alanları tahsisi řekil 9.1’de grldę gibidir. Eęitim alanlarının tahsisinde birok kısıt bulunmaktadır. ęretim yesi, faklte/yo/myo, blm tercihleri eęitim alanlarının tahsisinde nemli etkenlerdir. Eęitim alanları ynetimi problemi bir NP-hard optimizasyon problemidir. Problemin zm iin ise geliřtirilebilecek birok algoritma ve yntem mevcuttur.



Şekil 9.1 Eğitim Alanları Tahsisi

Üniversite mekânlarının verimlilik analizi yöntemleri iki farklı kategoride incelenebilir.

- Eğitim alanlarının verimli kullanımı,
- Diğer alanların verimli kullanımı,

Daha önce bu konuda yapılan çalışmalarda birçok matematiksel ve sezgisel yaklaşımlar geliştirilmiştir. Örneğin: Tepe Tımanma (Hill-Climbing) (Sharma ve Kumar, 1985), Tavlama Benzetimi (Simulated Annealing) (Burke ve Varley, 1998), Genetik Algoritmalar (Burke ve Varley 1999), Açgözlü Rasgele Adaptif Arama (Greedy Randomized Adaptive Search Procedure) (GRASP) (Schaerf, 1999), Rehberli Yerel Arama((Guided Local Search) (GLS) (Burke, vd., 2001), İteratif Yerel Arama(Iterated Local Search) (ILS) (Burke, Cowling ve Silva, 2001), Değişken Komşulu arama (Variable Neighborhood Search) (VNS) (Burke, vd., 2001), Eşik Değerli Kabul Algoritması (Threshold Acceptance Algorithms) (Burke, vd., 2001), Tabu Arama (TS) (Landa Silva, 2003) gibi birçok yaklaşım geliştirilmiştir.

Bu problem sadece derslikler için değil aynı zamanda diğer alanlar için de çözülmeye çalışılmıştır. Fakat üniversiteler için temel fiziksel kaynak; eğitim ve öğretim için kullanılan mekânlardır. Burke vd. (2001) alan tahsisini; “mekândaki alanları tahsis etme (örneğin derslikleri) birçok gereksinim ve kısıtı sağlamayı gerektirir” şeklinde tanımlamıştır. Burke ve Varley (1998) aynı zamanda alan tahsisini, “belirli fonksiyonlarına göre alanları tahsis etme” olarak tanımlamışlardır.

Eğitim alanlarının verimli bir şekilde kullanılması aşağıdaki kısıtların sağlanmasını gerektirmektedir. Kısıtlar sert ve esnek kısıtlar olarak genellikle ikiye ayrılır (Landa Silva, 2003).

### **Sert kısıtlar**

- Ders Saatleri: Her bir ders mutlaka program çizelgesinde yer almalıdır ve farklı periyodlara atanmalıdır. Bu kısıtın ihlal edilmesi durumunda ders çizelgelenemez.
- Derslik Doluluğu: İki ders aynı anda aynı dersliğe atanamaz. İki dersin çakışması durumunda bir kural ihlali olur, daha fazla dersin çakışması durumunda birden fazla kural ihlali oluşur.
- Çakışmalar: Aynı programdaki dersin şubeleri aynı öğretim görevlisi tarafından veriliyorsa mutlaka farklı periyodlara atanmalıdır. İki şubenin çakışması bir kural ihlali olur, daha fazla şubenin çakışması durumunda birden fazla kural ihlali oluşur.
- Uygunluk: Verilen periyodda öğretim görevlisi uygun değilse o periyoda hiç bir ders atanamaz. Bu periyoda uygun olmayan her bir ders bir kural ihlali oluşturur.

### **Esnek kısıtlar**

- Derslik Kapasitesi: Her bir dersi alan öğrenci sayısı dersin atandığı derslik kapasitesinden daha az veya eşit olmalıdır. Kapasiteyi aşan her bir öğrenci sayısı için 1 puan ceza verilir.

- Minimum Çalışma Günleri: Her bir dersin şubeleri minimum sayıya göre güne yayılmalıdır. Minimum sayının altındaki her bir gün için 5 puan ceza verilir.
- Ders Programı Yoğunluğu: Ders programındaki dersler bir birine yakın olmalıdır. Eğer bir ders diğerine aynı gün içerisinde bitişik değilse her bir periyod için 1 ceza verilir. Birbirinden ayrı iki ders için 2 puan ceza alınır.
- Derslik Değişmezliği: Bir dersin bütün şubeleri aynı dersliğe verilmelidir. Eğer bir ders farklı dersliklere veriliyorsa 1 puan ceza alır.
- Paylaşılamayan Özel Mekânlar: Üniversite yöneticisi, dekan veya profesör odaları gibi mekânlar özel ofislerdir ve diğer birimler veya kişiler tarafından paylaşılabilir.
- Spesifik Kullanıma Uygun Alanlar: Örneğin bilgisayar laboratuvarı veya müzik odası gibi özel odalar spesifik bölümler tarafından kullanılabilir.
- Birbirine Yakın Olması Gereken Alanlar: Bir bölüm sekreteri odasının bölüm başkanının odasına yakın olması.
- Birbirinden Uzak Olması Gereken Alanlar: Örneğin gürültüden mutlaka uzak yapılması gereken bazı derslerin veya laboratuvar uygulamalarının diğer birimlerden uzak olması gerekir.
- Birlikte Kullanılması Gereken Alanlar: İki farklı öğretim üyesi aynı projede çalışıyorsa aynı odayı kullanmaları gerekebilir.
- Grup Şeklinde Kullanılması Gereken Alanlar: Örneğin üniversite ortak dersleri, bitirme projeleri gibi derslerin grup halinde yapılması gerekir. Çizelge 9.1'de bu kısıtlara verilebilecek ceza örnekleri gösterilmiştir.
- Derslik Kullanım Oranı: Derslik kullanım oranı bir dersliğin ne kadar kapasitede kullanıldığını ölçmedir. Derslik kullanım oranı (U) sıklık fonksiyonu (OF) ve doluluk oranına (OC) bağlı bir fonksiyondur. Eğer her bir derslik %30 dan az kullanım oranına sahip ise 1 puan ceza verilir. Derslik doluluk oranı %75 den daha az olmamalıdır.
- Amaç Fonksiyonu: Burada yanlış ya da gereksiz alan kullanılmasının minimize edilmesi ve kısıtların değerlerinin minimize edilmesi gerekmektedir.
- Çizelge 9.1'de kısıtlara verilen değerler ifade edilmiştir.

Çizelge 9.1 Kısıtlar

| Kısıt                                      | Kısıta Verilen Ceza |
|--------------------------------------------|---------------------|
| Paylaşılamayan Özel Mekânlar               | 50                  |
| Spesifik Kullanıma Uygun Alanlar           | 20                  |
| Birbirine Yakın Olması Gereken Alanlar     | 10                  |
| Birbirinden Uzak Olması Gereken Alanlar    | 10                  |
| Birlikte Kullanılması Gereken Alanlar      | 10                  |
| Grup Şeklinde Kullanılması Gereken Alanlar | 5                   |
| Derslik doluluk oranı                      | 5                   |

**Matematiksel formülü:** İlgilenilen problemin matematiksel olarak ifadesi aşağıdaki gibidir.

Amaç Fonksiyonu:

$$\sum_{i \in E} (w^{RC} C_i^{RC} + w^{MWD} C_i^{MWD} + w^{RS} C_i^{RS} + w^{RC} C_i^{RC}) \quad (9.1)$$

$C_i^{RC}$ : i dersi için “Derslik Kapasitesi” cezası

$C_i^{MWD}$ : i dersi için “Minimum Çalışma Günleri” cezası

$C_i^{CC}$ : i dersi için “Ders Programı Yoğunluğu” cezası

$C_i^{RS}$ : i dersi için “Derslik Değişmezliği” cezası

$w^{RC}$ : “Derslik Kapasitesi” ağırlığı

$w^{MWD}$ : “Minimum Çalışma Günleri” ağırlığı

$w^{RS}$ : “Ders Programı Yoğunluğu” ağırlığı

$w^{RC}$ : “Derslik Değişmezliği” ağırlığı

Burada amaç fonksiyonu minimize edilmeye çalışılmaktadır. Eğitim alanı dışındaki alanların tahsisinde ise aşağıdaki parametreler kullanılmaktadır:

- Mekân Alanı Boyutu: Tahsis edilecek mekânların toplam metrik alan ölçümleri.
- Oda Boyutları: Tahsis edilecek odaların metrik alan ölçümleri.
- Odalar veya Binalar Arası Mesafeler: Her bir binanın veya odanın birbirine olan uzaklığı (metre).

Amaç fonksiyonunda kullanılan parametreler:

$m$  : Kullanılabilir oda sayısı

$n$  : Yer ayrılacak kişi sayısı

$h$  : Sert kısıtların sayısı

$s$  : Esnek kısıtların sayısı

$c(i)$  : Oda kapasitesi veya büyüklüğü

$w(j)$  :  $j$ . kişinin ihtiyacı olan alan büyüklüğü

$x(i,j)$  : 1 eğer  $j$ . kişiye  $i$ .oda atanmışsa, değilse 0

$SCP(r)$  :  $r$ . esnek kısıt ihlal edildiğinde verilen ceza

Derslikler dışındaki alanların kullanımıyla ilgili matematiksel model aşağıda verilmiştir:

$$\text{minimize } F(x) = f1(x) + f2(x) \quad (9.2)$$

$$WP(i) = c(i) - \sum_{j=1}^n w(j)x(i,j) \quad (9.3)$$

$$OP(i) = 2\left(\sum_{j=1}^n w(j)x(i,j) - c(i)\right) \quad (9.4)$$

$$f1(x) = \sum_{i=1}^m (WP(i) + OP(i)) \quad (9.5)$$

$$f2(x) = \sum_{r=1}^s SCP(r) \quad (9.6)$$

Bu çalışmada üniversite verileri yarışmadaki veri setlerine benzer formüle edilerek oluşturulmuştur. Üniversite test verileri Çizelge 9.2'deki gibi tanımlanmıştır.

Çizelge 9.2 Üniversite test verileri

| Veri Seti | Ders Sayısı | Müfredat Dersleri | Derslik | Gün | Zaman Dilimi | Kısıtlar |
|-----------|-------------|-------------------|---------|-----|--------------|----------|
| SDU1      | 76          | 486               | 9       | 5   | 8            | 171      |
| SDU2      | 51          | 936               | 11      | 5   | 8            | 10       |
| SDU3      | 212         | 1266              | 20      | 5   | 8            | 626      |
| SDU4      | 132         | 807               | 18      | 5   | 8            | 294      |
| SDU5      | 141         | 892               | 21      | 5   | 8            | 830      |
| SDU6      | 78          | 571               | 31      | 5   | 8            | 414      |
| SDU7      | 36          | 236               | 13      | 5   | 8            | 166      |
| SDU8      | 60          | 331               | 12      | 5   | 8            | 49       |
| SDU9      | 83          | 595               | 14      | 5   | 8            | 324      |
| SDU10     | 76          | 464               | 13      | 5   | 8            | 324      |

Müfredat tabanlı ders çizelgeleme problemi ITC2007’de tanımlanmıştır. Problem 4 adet sert kısıt, 4 adet de esnek kısıt içermektedir. Üniversitede yapılan çalışmada bu kısıtlara ek olarak bazı kısıtlar tanımlanmıştır.

#### Oda kullanım frekansı

OF = İncelenen zamandaki kullanılan saat sayısı/ incelenen zamandaki kullanılabilir maximum saat sayısı. Örnek: Haftalık kullanılan saat sayısı = 40, Odanın haftalık kullanılabilir saat sayısı = 50 olursa;

OF = 40 saat/50 saat = % 80’dir.

#### Oda doluluğu

OD = İncelenen zamandaki toplam öğrenci sayısı/oda kapasitesi x incelenen zamandaki kullanılan saat sayısı.

Örnek: Odayı kullanan haftalık toplam öğrenci sayısı = 800, Odanın toplam kapasitesi = 80, ve Haftalık kullanılan saat sayısı = 40 olursa;

OD = 800/(80x40) = % 25’dir.

### **Oda kullanımı (%)**

Oda Frekansı ve Oda Doluluk oranlarının çarpımıdır.

$$U = OF \times OD \quad (9.7)$$

Örnek: OF = % 80 ve OD = % 25 olursa;

Kullanım =  $(80 \times 25) / 1000 = \% 20$ 'dir.



## **10. DERSLİK KULLANIM ORANLARINI ARTIRMAK İÇİN BİR KARAR DESTEK SİSTEMİ GELİŞTİRİLMESİ**

Üniversitelerin eğitim, araştırma ve topluma hizmet olmak üzere üç ana misyonu vardır. Bu misyonlardaki en önemli iki faktör ise mekan ve personeldir. Üniversitelerdeki personel akademik ve idari personel olarak üniversitelerin insan kaynaklarını oluşturur. Fakülteler, Yüksekokullar, Meslek Yüksekokulları, Araştırma Merkezleri, Hastaneler, Laboratuvarlar ve İdari Binalar ise mekan kaynaklarını oluşturular. Bu mekanlar hem akademik aktivitelre hemde yönetsel aktivitelre fiziksel kaynak oluşturmaktadırlar. Türkiye’de üniversite personeli planlaması ve öğrenci yerleştirme merkezi bir sistem ile sağlanırken mekanların planlanmasına ve amaca uygun kullanımına ilişkin bir sistem bulunmamaktadır. Mekan yönetimi belirli bir kullanım için veya belirli bir kullanıcıya alan tahsis etme kabiliyeti olarak tanımlanabilir. Dahası üniversitelerin en önemli giderlerinden biri olan mekan giderlerinin doğru bir şekilde kullanılması ancak rasyonel bir mekan planlaması ile gerçekleştirilebilir. Bu rasyonel mekan planlaması için üniversitelerin hangi mekanları hangi amaçla kullandığı tespit edilerek doğru bir sınıflandırma sistemi oluşturmak gerekmektedir.

Üniversitelerde mekan envanteri yönetimi için sistematik ve kapsamlı bir planlama yapılması gerekir. Envanter her yıl sürekli gözden geçirilmeli, envanter bilgilerine erişilebilmeli ve etkin bir şekilde kullanılabilmelidir. Üniversitelerdeki eğitim alanlarının sürekli artan öğrenci sayısını karşılaması, değişen taleplere yanıt vermesi gerekmektedir. Bu nedenle etkili mekan planlaması, esnek bir tasarım yapılması elzemdir.

Bu tezde karar destek sistemi (DSS) geliştirilmesindeki amaç, mevcut derslik kullanım oranlarını artırmak, geniş ve sistematik bir analitik yaklaşım geliştirmek, ders çizelgeleme yapmak, üniversite veya fakültelerin kısıtlarını oluşturmak, halen kullanılan bina durumlarını tespit etmek, standardizasyon yapabilmek ve ihtiyaçları belirleyebilmektir. Şekil 10.2’de Yükseköğretim sektörü yatırım karar destek sistemi modeli örneği gösterilmiştir.

Bir karar destek sistemi geliřtirmek olduka fazla zaman ve para gerektirir. Mekan ynetimi iin bir karar destek sistemi daha kullanışlı niversite mekanları oluřturmak iin sınıf kullanımlarını deėiřtirme, derslik kullanım oranlarını artırma, mekan maliyetlerini azaltma gibi karar verme yeteneėine sahip olmalıdır. Fakat karar destek sistemi geliřtirilirken bir ok kısıt ortaya ıkmaktadır. Karar destek sistemleri iin ėrenme yaklařımları (aktif ėrenme, arařtırma ve yazılım sistemleri), iterative prototipler (evrimsel geliřtirme sreleri) v.b yaklařımlar bulunmaktadır. Karar destek system iin bazı faktrler şunlardır:

**DSS geliřtirme sreci:** Kullanıcıların geliřtirme srecine erken katılımları, sistem gereksinimlerinin doėru bir řekilde belirlenmesi, evrimsel system geliřtirme ve faydalarının iyi bir řekilde tanımlanması DSS geliřtirme srelerini gsterir. Bu geliřtirme sreleri, mekan gereksinim analizlerinin yapılması, mekan envanter bilgilerinin ve ders programı izelgelerinin toplanmasını ierir.

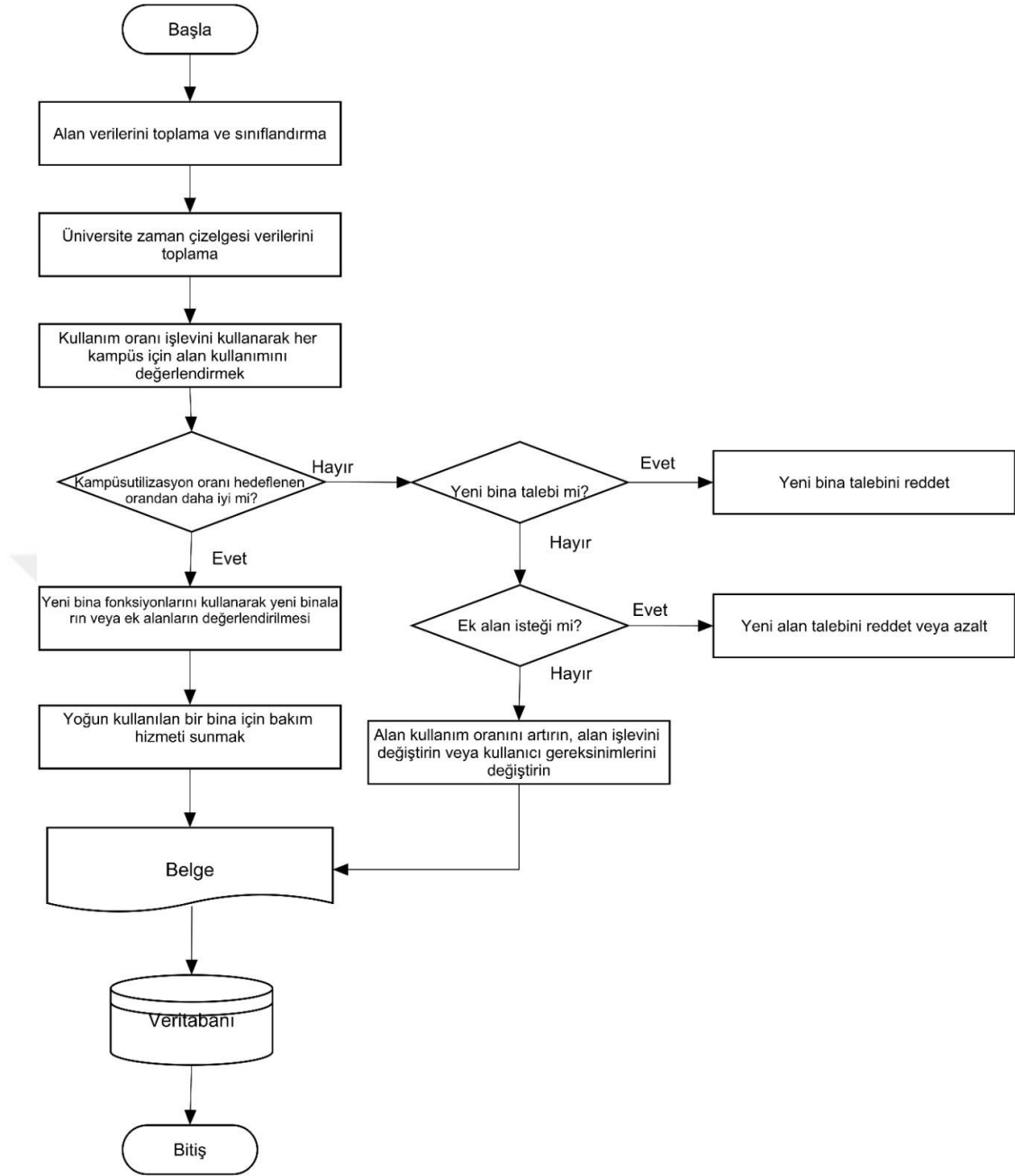
**DSS bileřenleri:** Model oluřturulması, verilerin kalitesi, kullanıcı arayz, raporlama sistemi, very ynetim teknolojileri ve very toplama DSS bileřenlerini oluřturmaktadır.

**Karar sreleri:** DSS karar srelerinde yeni mekan alanı hesaplama, bakım veya yeniden planlama nerisi sunma, yeni mekan taleplerini reddetme, derslik kullanım oranlarını artırma gibi kararlar yer almaktadır.

**Karar ıktıları:** DSS karar ıktıları kullanıcılara hızlı, doėru ve etkili zmler sunmalıdır.

**Kullanıcı memnuniyeti:** DSS tarafından retilen veriler kullanıcının ihtiyalarına cevap verebilmelidir.

řekil 10.1 Tasarlanan Karar Destek Sistemi iin akış diyagramını gstermektedir.



Şekil 10.1. DSS Akış Diyagramı

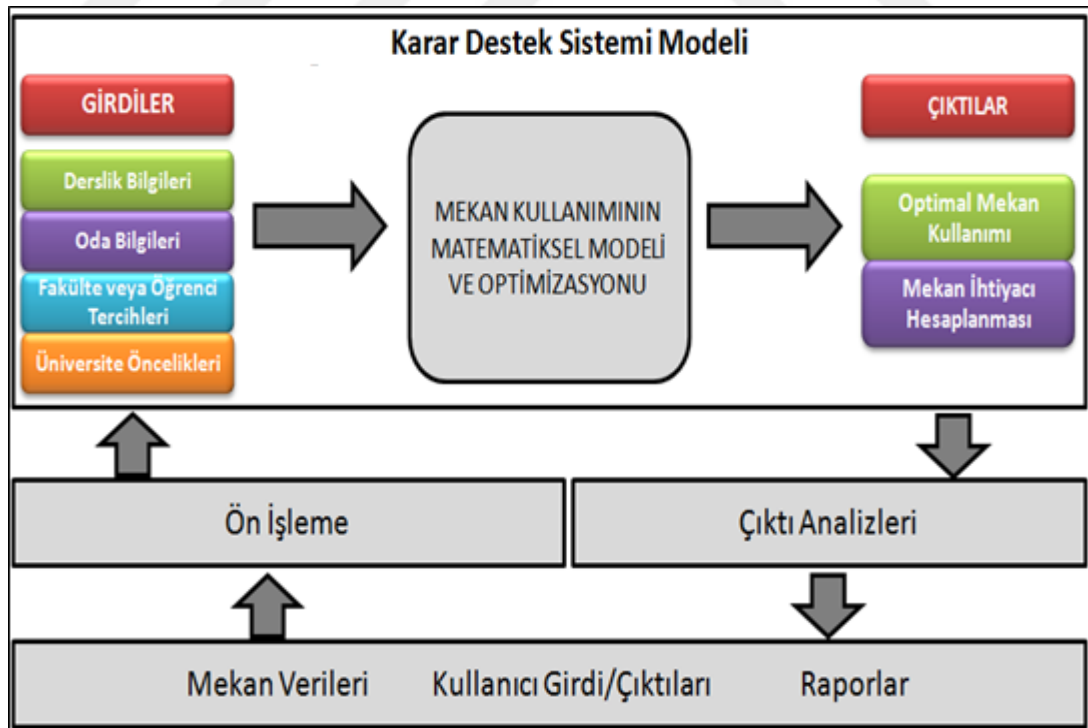
Karar destek sistemi modelinde yapılması amaçlanan uygulamalar şunlardır:

**Web tabanlı kullanıcı arayüzü:** Mekân envanteri girdi verilerinin veritabanına kaydedilmesi, ön işleme işlemlerinin yapılabilmesi ve çıktıların elde edilmesi için kullanıcı dostu web tabanlı bir arayüz geliştirilecektir. Bu arayüzün kullanımı ile ilgili üniversitelerin kullanabileceği bir kılavuz oluşturulacak, videolar hazırlanacaktır.

**Raporlama aracı:** Mevcut ve ileriki yıllardaki işlemlerin gözlemlenebilmesi, etkili, geniş ve tutarlı bir mekân envanter veritabanının oluşup oluşmadığının incelenmesine olanak sağlayacak bir raporlama aracı geliştirilecektir. Derslik/oda kullanım oranları daha önceki verilen örneklerdeki gibi olup istenilen parametreye göre rapor alınabilecektir. Ayrıca üniversitelerden gelen ekstra isteklere göre raporlar tasarlanacaktır.

**Karar destek seçenekleri:** Mekân ihtiyacı olan veya ileriki yıllarda ihtiyacı olabilecek olan üniversiteler için karar destek sistemi çeşitli seçenekler sunacaktır. Hangi büyüklükte bir mekâna veya nasıl bir değişime ihtiyaç olduğu belirlenebilecek ve bu sayede dersliklerin ve eğitimin kalitesi dolayısıyla da insanların memnuniyetleri artacaktır.

**Web tabanlı ders çizelgeleme ve oda tahsisi modeli:** Kullanıcı arayüzü ile farklı üniversitelere ait ders çizelgeleme bileşenleri/kısıtları sisteme eklenebilecek ve çizelgeleme yapılabilir.



Şekil 10.2. Derslik İhtiyacı Karar Destek Sistemi Modeli

Karar destek sistemi tasarımı mekân kullanımını artırmak veya ne kadar alana ihtiyaç olduğunu belirlemek için aşağıdaki süreçler izlenecektir:

**Ön işleme:** Bu süreçte, mevcut mekânların günümüzdeki kullanımını analiz etmek için bir envanter veritabanı oluşturulacak, mevcut bir veritabanı var ise mevcut veriler üzerinde gerekli güncelleme ve düzeltmeler yapılacaktır. Proje kapsamında ön işlem süreci detaylı bir şekilde oluşturulacaktır. Bu işlemler için;

- Haftalık, günlük ve ders saatlerindeki derslik/oda kullanım oranları,
- Derslik/oda türüne göre kullanım oranları,
- Bölümlere göre derslik/oda kullanım oranları,
- Derslik kapasitelerine göre haftalık, günlük ve ders saatlerindeki kullanım oranları,
- Her bir derslik/odadaki boş saatlerin gösterilmesi,
- Derslik/oda kullanımının orantısız olduğu akademik birimler tanımlanır.

Örnek bir uygulama olarak Süleyman Demirel Üniversitesi Hukuk Fakültesi binası incelenmiştir. Şekil 10.3'de 22 derslikte anlık olarak 2345 öğrenci kapasitesine sahip fakülte binasının doluluk oranları verilmiştir. Bu fakültenin maximum kullanım kapasitesi  $22 \times 45 = 990$  ders saatidir. Grafik incelendiğinde pazartesi günü 13:25-15:05 saatleri arasında ve çarşamba günü 09:55-10:50 saatleri arasında toplam derslik kullanım oranı en yüksektir. Cuma günü ise dersliklerin kullanım oranlarının en az olduğu gün olarak gözlemlenmektedir.

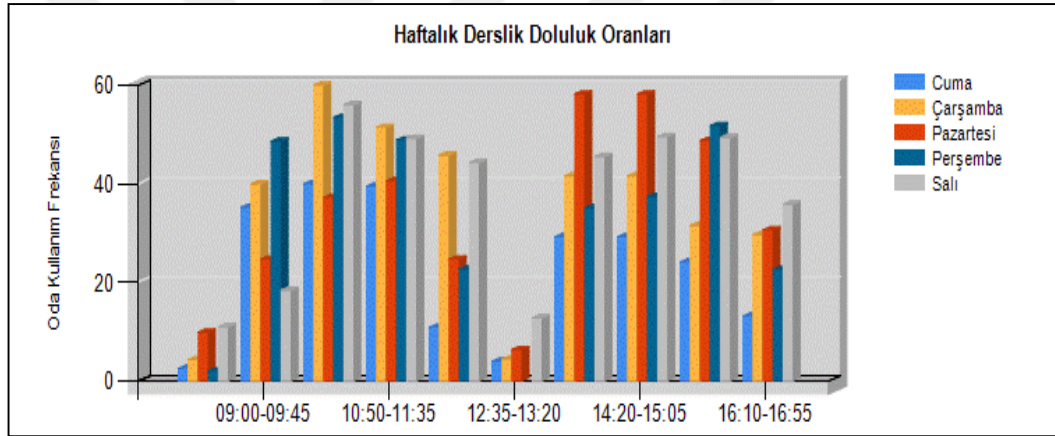
Çizelge 10.1'de tüm fakülte/YO/MYO'ların pazartesi gününe ait derslik alanlarının kullanım çizelgesi gösterilmiştir.

Çizelge 10.1. Tüm Fakülte/YO/MYO'ların Pazartesi Günü Derslik Alanlarının Kullanım Çizelgesi

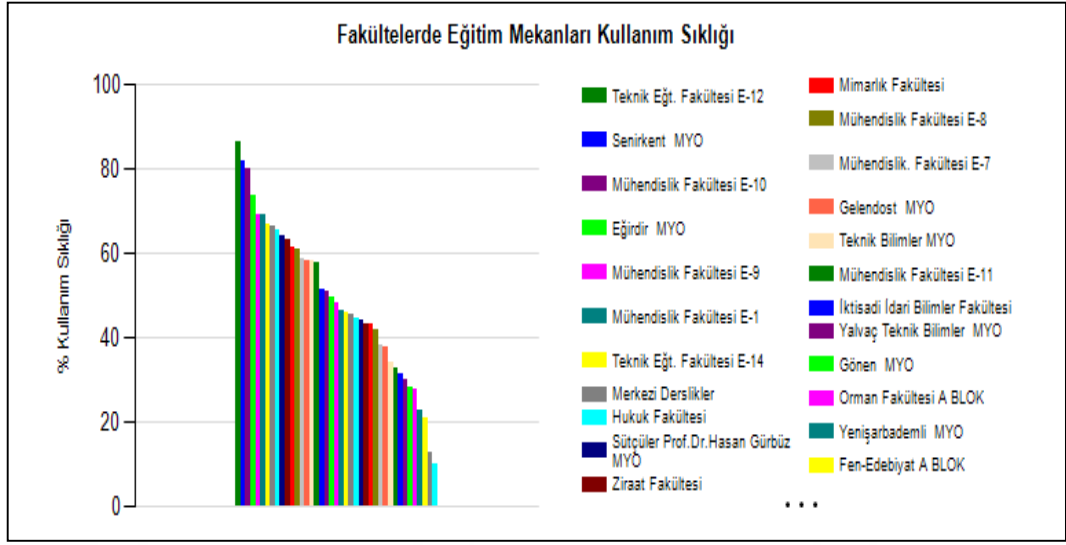
| Bina                               | Öğrenci Sayısı | Derslik Sayısı | Derslik Kapasitesi (Ders Saati)(A) | Kullanılan Kapasite (Ders Saati)(B) | Derslik Kullanım Sıklığı (%) (B) / (A) |
|------------------------------------|----------------|----------------|------------------------------------|-------------------------------------|----------------------------------------|
| Aksu Mehmet Süreyya Demiraslan MYO | 1786           | 15             | 6039                               | 1903                                | 31.51                                  |
| Atabey MYO                         | 2771           | 28             | 11529                              | 3936                                | 34.14                                  |
| Diş Hekimliği Fakültesi            | 547            | 16             | 5580                               | 550                                 | 9.86                                   |
| Eğirdir MYO                        | 3511           | 14             | 5931                               | 4380                                | 73.85                                  |
| Eğirdir Sağlık Hizmetleri MYO      | 205            | 5              | 2880                               | 360                                 | 12.5                                   |
| Eğirdir Turizm ve Otelcilik YO     | 1636           | 9              | 4005                               | 1680                                | 41.95                                  |
| Eğitim Fakültesi                   | 3140           | 35             | 16659                              | 3769                                | 22.62                                  |
| Fen-Edebiyat A BLOK                | 8440           | 75             | 24138                              | 11086                               | 45.93                                  |
| Gelendost MYO                      | 1895           | 9              | 2889                               | 1682                                | 58.22                                  |
| Gönen MYO                          | 3491           | 26             | 8892                               | 4385                                | 49.31                                  |
| Güzel Sanatlar Fakültesi           | 2446           | 63             | 4599                               | 2094                                | 45.53                                  |
| Hukuk Fakültesi                    | 7908           | 22             | 21105                              | 13843                               | 65.59                                  |
| I.Sağlık Hiz. MYO                  | 2388           | 13             | 6219                               | 2366                                | 38.04                                  |
| Isparta MYO                        | 3109           | 17             | 8010                               | 3474                                | 43.37                                  |
| İktisadi İdari Bilimler Fakültesi  | 12866          | 53             | 29997                              | 15352                               | 51.18                                  |
| İlahiyat Fakültesi                 | 3243           | 37             | 19962                              | 5546                                | 27.78                                  |
| Keçiborlu MYO                      | 3988           | 30             | 17586                              | 4963                                | 28.22                                  |
| Merkezi Derslikler                 | 8247           | 40             | 19395                              | 12914                               | 66.58                                  |
| Mimarlık Fakültesi                 | 1928           | 8              | 3249                               | 1992                                | 61.31                                  |
| Mühendislik Fakültesi E-1          | 4101           | 10             | 7029                               | 4863                                | 69.18                                  |
| Mühendislik Fakültesi E-10         | 1894           | 7              | 3852                               | 3074                                | 79.8                                   |
| Mühendislik Fakültesi E-11         | 913            | 3              | 1944                               | 1122                                | 57.72                                  |
| Mühendislik Fakültesi E-2          | 90             | 1              | 108                                | 48                                  | 44.44                                  |
| Mühendislik Fakültesi E-4          | 2123           | 12             | 6210                               | 2741                                | 44.14                                  |
| Mühendislik Fakültesi E-6          | 1595           | 5              | 5355                               | 1744                                | 32.57                                  |
| Mühendislik Fakültesi E-8          | 2008           | 8              | 5418                               | 3294                                | 60.8                                   |
| Mühendislik Fakültesi E-9          | 3666           | 11             | 5985                               | 4145                                | 69.26                                  |
| Mühendislik Fakültesi E-7          | 1386           | 6              | 3564                               | 2083                                | 58.45                                  |
| Orman Fakültesi A BLOK             | 2821           | 17             | 9144                               | 4384                                | 47.94                                  |
| Senirkent MYO                      | 6701           | 35             | 17280                              | 14140                               | 81.83                                  |
| Sütçüler Prof.Dr.Hasan Gürbüz MYO  | 2659           | 13             | 4329                               | 2772                                | 64.03                                  |
| Şarkıkaraağaç MYO                  | 1974           | 23             | 9324                               | 2800                                | 30.03                                  |
| Teknik Bilimler MYO                | 7862           | 33             | 13239                              | 7704                                | 58.19                                  |
| Teknik Eğt. Fakültesi E-12         | 3445           | 19             | 4770                               | 4130                                | 86.58                                  |
| Teknik Eğt. Fakültesi E-14         | 3110           | 10             | 4500                               | 3000                                | 66.67                                  |
| Uluborlu Selahattin Karasoy MYO    | 5399           | 27             | 15651                              | 5901                                | 37.7                                   |
| Yalvaç MYO                         | 2366           | 18             | 7668                               | 3320                                | 43.3                                   |

| Bina                                     | Öğrenci Sayısı | Derslik Sayısı | Derslik Kapasitesi (Ders Saati)(A) | Kullanılan Kapasite (Ders Saati)(B) | Derslik Kullanım Sıklığı (%) (B) / (A) |
|------------------------------------------|----------------|----------------|------------------------------------|-------------------------------------|----------------------------------------|
| Yalvaç Büyükkutlu Uygulamalı Bilimler YO | 1164           | 28             | 5751                               | 1206                                | 20.97                                  |
| Yalvaç Teknik Bilimler MYO               | 2530           | 19             | 9441                               | 4811                                | 50.96                                  |
| Yenişarbademli MYO                       | 1629           | 11             | 3204                               | 1492                                | 46.57                                  |
| Ziraat Fakültesi                         | 5677           | 67             | 16353                              | 10360                               | 63.35                                  |

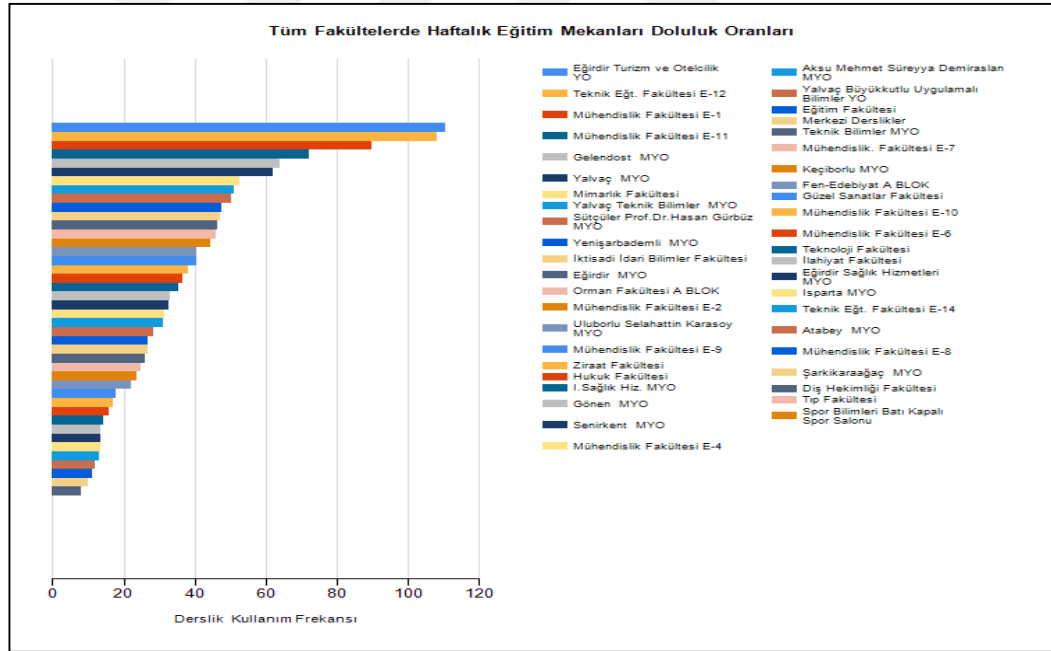
Çizelge 10.1'deki veriler kullanılarak oluşturulan toplam 44556 öğrenci kapasitesine sahip 941 derslik verisi için oluşturulan grafikler Şekil 10.3, 10.4 ve 10.5'de verilmiştir. Grafikler incelendiğinde pazartesi gününe ait doluluk oranları ile haftalık doluluk oranlarının değiştiği gözlemlenmiştir.



Şekil 10.3. Haftalık Derslik Doluluk Oranları

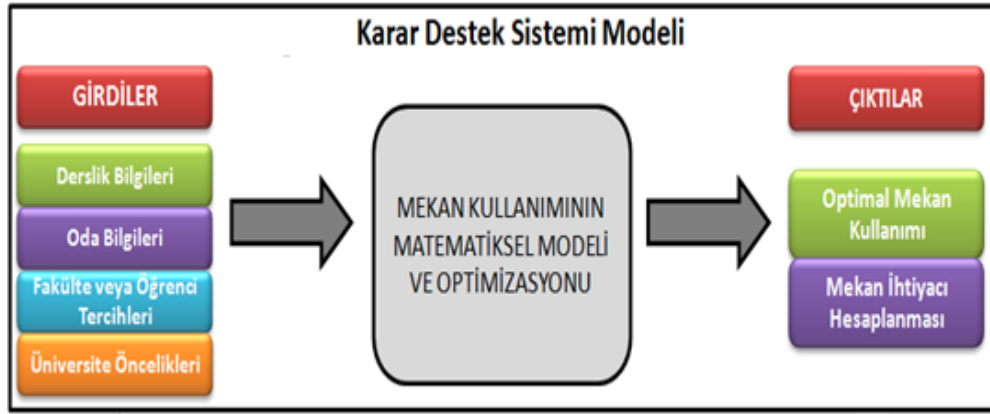


Şekil 10.4. Tüm Fakülte/YO/MYO'ların Pazartesi Günü Derslik Alanlarının Doluluk Oranları



Şekil 10.5. Tüm Fakülte/YO/MYO'ların Haftalık Derslik Alanlarının Doluluk Oranları

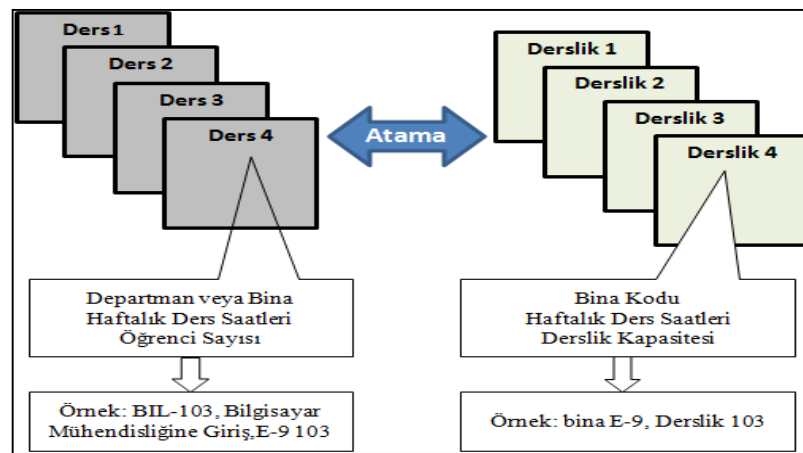
**Optimizasyon modeli:** Proje kapsamında optimum derslik/oda kullanımının optimizasyonu geliştirilecektir. Şekil 10.6 ve 10.7'deki modelde, fakülte veya yöneticilerin istekleri ve ilgili tüm kısıtlar sisteme girilebilecek ve model tarafından otomatik olarak dikkate alınabilecektir.



Şekil 10.6. Derslik İhtiyacı Karar Destek Sistemi Optimizasyon Modeli

Derslerin dersliklere atanması matematik modeli aşağıdaki şartlara göre tasarlanacaktır:

- Her bir ders en fazla bir dersliğe atanmalıdır.
- Derslik kapasitesi, dersi alan öğrenci sayısından az olmamalıdır.
- Her bir ders, aynı zaman diliminde birden fazla dersliğe atanmamalıdır.
- Her bir dersliğe, aynı zaman diliminde birden fazla ders atanmamalıdır.
- Derslik bölüme en yakın mesafede olmalıdır.
- Dersliğe ders atanırken % 75 kullanım oranına erişmeye çalışmalıdır.
- Mümkün olduğunca aynı sınıfın öğrencileri birbirini takip eden derslerde aynı dersliği kullanmalıdır.



Şekil 10.7. Ders Programı Optimizasyon Modeli

**Derslik İhtiyacı Hesaplanması:** Derslik ihtiyacının hesaplanmasında kullanılan çeşitli yaklaşımlar bulunmaktadır. Bunlar içerisinde en önemlileri aşağıda verilmiştir:

### Stres-Testi

Mekân ihtiyacı hesaplanmasında öncelikli amaç; var olan ders zaman çizelgesini değiştirmeden, ekstra kapasite kazanmak için dersleri farklı sınıflara atamaktır. Bu işlemi yaparken mevcut çizelge korunarak yedek çizelge oluşturulacak ve bu iki çizelge dersliklerin kapasite olarak doluluk oranları karşılaştırılacaktır. Bu karşılaştırma sonucunda kullanılabilecek ek mekânlar elde edilmeye çalışılacaktır. Çizelge 10.2’de örnek bir çizelge verilmiştir.

Çizelge 10.2. Stres Testi Sonucu Elde Edilen Çizelge Örneği

|                | Mevcut Kullanım | Toplam Haftalık Kapasite | Kullanılabilecek Kapasite |
|----------------|-----------------|--------------------------|---------------------------|
| Derslik        | 1252            | 1600                     | 1400                      |
| Kullanım Oranı | % 78            | % 100                    | % 87,5                    |
| Artış Miktarı  |                 |                          | % 9,5                     |

### En iyi zamanlama analizi

Bu analizde, farklı üniversitelerin, birimlerin ve bölümlerin ders başlama ve bitiş zamanları incelenir. Üniversitelerin ders saatleri politikaları hakkında bilgi edinilerek, bu bilgilere göre eğer ders saatleri değiştirilirse kullanılabilecek derslikler tespit edilmeye çalışılır. Çizelge 10.3’de örnek bir çizelge verilmiştir.

Çizelge 10.3. En İyi Zamanlama Analizi Sonucu Elde Edilen Çizelge Örneği

| Üniversite                            | Ders Saatleri      | Bu saatlerde toplam kullanım kapasiteleri | Kullanılabilecek Kapasite |
|---------------------------------------|--------------------|-------------------------------------------|---------------------------|
| Süleyman Demirel Üniversitesi         | 08:05-16:55-45 dk. | 1600                                      | 0-% 100                   |
| Selçuk Üniversitesi                   | 08:30-16:50-45 dk. | 1400                                      | 200-% 12,5                |
| Akdeniz Üniversitesi                  | 08:30-17:20-50 dk. | 1500                                      | 100-% 6,25                |
| İstanbul Üniversitesi                 | 08:30-17:35-50 dk. | 1550                                      | 50-% 3,125                |
| Mehmet Akif Ersoy Üniversitesi        | 09:00-16:45-45 dk. | 1300                                      | 300-% 18,75               |
| Kahramanmaraş Sütçü İmam Üniversitesi | 08:10-16:10-50 dk. | 1450                                      | 150-% 9,125               |

#### Yeni derslik önerisi

Yukarıdaki iki test sonucunda hala üniversitenin sahip olduğu derslikler ihtiyaçlarını karşılayamıyor ise yeni yapılacak dersliklerin özellikleri hesaplanacaktır. Burada yeni inşa edilecek mekânın belirlenen standartlara göre derslik kapasiteleri ve sayıları, laboratuvar kapasiteleri ve sayıları hakkında raporlar oluşturulacaktır.

#### Farklı üniversitelerin fakülte ve/veya bölüm bazlı sahip oldukları derslikler açısından analizi

Sisteme verileri girilen üniversitelerin tüm bölümlerinde eğer sadece bölümlerin kullanımına tahsis edilmiş özel derslikler (genelde laboratuvar veya sadece bölüm tarafından kullanılan atölye vb.) varsa farklı üniversiteler için bu dersliklerin ne kadar verimli kullanıldığı analizi yapılacaktır. Örneğin A- üniversitesinde bulunan resim bölümündeki atölyeler ile B- üniversitesindeki resim bölümünün sahip olduğu atölyeler kullanım açısından karşılaştırılacaktır.

## **Derslik ihtiyacını etkileyen temel faktörler ve analizi**

Klasik yöntemlerden olan regresyon analizi ve modern yöntemlerden olan bulanık mantık temelde en çok kullanılan modelleme yöntemleridir. Karar destek sistemlerinin oluşturulmasında, bu yöntemlerden yardım alınacaktır. Bulanık mantık, bir karar destek sisteminin kalbini oluşturduğundan dolayı, aşağıda bu projedeki kullanımı açıklanmıştır.

## **Derslik ihtiyacı yatırım karar destek sistemi**

Ülkemiz yükseköğretim tarihine bakıldığında yükseköğretime olan talep hep artmıştır. Bu talebi karşılayacak kapasite miktarı yıllar itibarı ile yetersiz kalmıştır.

Zaman içerisinde teknolojik gelişmelere ve iş dünyasının talebine bağlı olarak ortaya çıkan yeni mesleklerin veya mevcut olanların yenilenmesi bilimsel olarak eğitimi veren fakülte veya bölümlerin açılması ihtiyacını doğurmaktadır. Buna paralel olarak oluşan mekân ihtiyacı bazen mevcut mekânlarla karşılanabilirken bazen de yeni mekânların oluşturulması gerekmektedir. Kullanıcı tarafından sisteme girilen bölüm bilgileri ve mevcut mekânlar karşılaştırılarak yeni açılması planlanan bölümlerin mekânsal ihtiyaçları analiz edilecektir. Yeni açılacak bölümler için hem ofis alanları hem de eğitim alanları ihtiyacı ve maliyeti ile ilgili raporlar hazırlanacaktır.

Zaman içerisinde öğrenci sayısı azalan ve hiç öğrenci alınmayan bölümlerin kullandığı ofis alanları, laboratuvarlar ve diğer eğitim mekânlarının ne şekilde değerlendirileceği konusu geçmişte çok fazla incelenmemiştir. Buna paralel olarak ortaya çıkan mekân fazlalıkları ile ilgili analizlerin üniversitelerde yapılmadığı düşünülmektedir. Bu mekânların hemen sisteme girilmesi ve bundan sonra ihtiyacı olan bölümlerin kullanımına açılabilmesi sağlanacaktır.

Yukarıda belirtilen durumlar, sürekli olarak sistem tarafından izlenecek ve koordineli olarak çalışmaları sağlanacaktır.

## **Karar destek sistemlerinin genel yapısı**

Bu projede ilgilenilen konular bir optimizasyon problemidir. Optimizasyon, en basit tanımı ile eldeki kısıtlı kaynakların en iyi şekilde kullanılmasıdır. Matematiksel olarak ise optimizasyon, bir fonksiyonunun amacına uygun olarak maksimize veya minimize edilmesi yani amaç fonksiyonunun en iyi değerini veren kısıtlardaki değişkenlerin değerinin bulunmasıdır.

Kombinatorial optimizasyon, belirli bir problem için olası çözümler içerisinde en iyi çözümü bulma problemi olarak tanımlanabilir (Hoos ve Stützle, 2005). Bu problemlere çizelgeleme, üretim planlama, rotalama, ekonomik sistemler ve yönetim gibi birçok gerçek dünya problemleri içerisinde sıklıkla karşılaşılır (Hoos ve Stützle, 2005). Gerçek dünyadaki birçok optimizasyon problemi karmaşıktır ve çözülmesi oldukça zordur. Bunun nedeni büyük ve yoğun kısıtlar, çok karmaşık işlemlerle modellenen arama uzaylarıdır (Weise v.d., 2009). Genellikle sezgisel yöntemler bu problemleri çözmek için kullanılırlar. Sezgisel yöntemlerin temel amacı, optimum çözümü (veya optimuma yakın çözümü) garanti etmemekle beraber, gerçeğe uygun bir sürede makul en iyi çözümü bulmaktır (Talbi, 2009; Gendreau ve Potvin, 2010). Meta-sezgisel algoritmalar arama uzayındaki etkin yönlendirme yapmak için bazı yüksek seviyeli kontrol stratejileri sağlarlar. Bir çok meta-sezgisel algoritma ve onların hibridleri optimizasyon problemlerini çözmek için uygulanmıştır. Meta-sezgisel algoritmalara örnekler; scatter arama (scatter search), tabu arama (tabu search), genetik algoritmalar, genetik programlama, memetik algoritmalar, değişken komşuluk arama (variable neighborhood search), güdümlü yerel arama (guided local search), GRASP (greedy randomized adaptive search procedure), karınca kolonisi (ant colony optimization), benzetimli tavlama (simulated annealing), iteratif yerel arama (iterated local search), çoklu-başlama metodları (multi-start methods) ve paralel stratejilerdir. (Talbi, 2009; Gendreau ve Potvin, 2010).

Bir optimizasyon problemi; karar değişkenleri, kısıtlar ve amaç fonksiyonu olmak üzere temelde üç bileşenden oluşmaktadır. Bu projede ele alınacak ve birbiriyle ilişkili olan üç temel optimizasyon problemi şunlardır:

**Derslik planlaması:** Uzun vadede kampüs veya yeni bina tasarımını gerektirmektedir.

**Ders programlarının oluşturulması:** Her dönem başında ders programlarının oluşturulması gerektiğinden dolayı dinamik bir yapıda olması gerekmektedir.

### **Mekân planlaması ile ilgili optimizasyon bileşenleri**

Karar değişkenleri: Ne kadar yeni alana ihtiyaç olduğu sorusuna yanıt ararlar?

Kısıtlar:

- Her bir bölümdeki kişilerin bir odaya atanması,
- Öğrenci başına düşen kullanılabilir alan yeterli olması,
- Personel başına düşen kullanılabilir alan yeterli olması.

Amaç fonksiyonu: Optimizasyon problemlerinde amaç fonksiyonu ya maximizasyon ya da minimizasyon hedefler. Bu problemde tüm mekânların kullanılması için amaç fonksiyonu aşağıdaki gibi belirlenebilir.

$$C_i = C_1, C_2, C_3 \dots C_n \quad i = 1, 2, 3, \dots, n \quad (10.1)$$

Burada  $C_1$  1.türdeki bir dersliğin kapasitesi  $C_2$  2. türdeki bir dersliğin kapasitesi ni göstermektedir.

$$X_i = X_1, X_2, X_3 \dots X_n \quad i = 1, 2, 3, \dots, n \quad (10.2)$$

Burada  $X_1$  1.türdeki bir odanın kapasitesi  $X_2$  2.türdeki bir odanın kapasitesini göstermektedir.

$$A_i = A_1, A_2, A_3 \dots A_n \quad i = 1, 2, 3, \dots, n \quad (10.3)$$

Burada  $A_1$  1.türdeki odaların sayısı  $A_2$  2.türdeki odaların sayısını göstermektedir.

Toplam kullanılabilir derslik

$$TD = \sum_{i=1}^n C_i A_i \quad (10.1)$$

Amaç Fonksiyonu:

$$\text{Maximize } \sum_{i=1}^n C_i X_i \quad (10.2)$$

### **Ders programı oluşturulması ile ilgili optimizasyon bileşenleri**

Ders programı oluşturmada kullanılan optimizasyon bileşenleri aşağıdaki gibidir.

Karar değişkenleri: Haftanın hangi günü, günün hangi saatinde hangi dersliğe hangi ders atanacak olarak belirlenmiştir.

#### **Derslik ile ilgili kısıtlar**

- Derslik sayısı, derslik kapasitesi, ders saatleri, derse özel derslik, çakışmalar, ders kredisi,

#### **Ders ile ilgili kısıtlar**

- Aynı bölümün aynı sınıfının herhangi iki dersinin çakışması,
- Herhangi bir ders aynı anda iki farklı dersliğe atanmaması,
- Her ders mutlaka bir dersliğe atanmalıdır.
- Öğretim elemanı ile ilgili kısıtlar:
- Öğretim elemanının farklı şubelerdeki derslerinin çakışmaması,
- Öğretim elemanının günlük ders yükünün dengelenmesi (ders programı yoğunluğu),
- Öğretim elemanının aynı gün ve saatte üniversitenin farklı bir biriminde dersinin olması,
- Öğretim elemanlarının lisansüstü çalışmaları ya da proje çalışmaları için boş gün ihtiyaçları (Minimum Çalışma Günleri).

## Diğer kısıtlar

- Bazı derslerde şubelerin birleştirilmesi,
- Şubelerin günlük ders yükünün dengelenmesi(ders programı yoğunluğu),
- Teorik olarak ağır derslerin günün son saatlerine gelmemesi.

Amaç fonksiyonu:

$$\text{Minimize } \sum_{i \in E} (w^{RC} C_i^{RC} + w^{MWD} C_i^{MWD} + w^{RS} C_i^{RS} + w^{RC} C_i^{RC}) \quad (10.3)$$

Burada kısıtlar için hesaplama formülleri aşağıdaki gibidir:

$C_i^{RC}$  : i dersi için “Derslik Kapasitesi” cezası

$C_i^{MWD}$  : i dersi için “Minimum Çalışma Günleri” cezası

$C_i^{CC}$  : i dersi için “Ders Programı Yoğunluğu” cezası

$C_i^{RS}$  : i dersi için “Derslik Değişmezliği” cezası

$w^{RC}$  : “Derslik Kapasitesi” ağırlığı

$w^{MWD}$  : “Minimum Çalışma Günleri” ağırlığı

$w^{RS}$  : “Ders Programı Yoğunluğu” ağırlığı

$w^{RC}$  : “Derslik Değişmezliği” ağırlığı

Bilimsel yazında sezgisel algoritmalar ile ilgili olarak çeşitli çalışmalar yapılmıştır: (Deb 1999; Machwe, Parmee vd, 2005; Parmee, Abraham vd, 2008). Bunun yanısıra örneğin; Graf Teorisi, Yapay Zeka ile birlikte Treshold mekanizması kullanılmıştır (Beyrouthy vd, 2007). Bu çalışmalarda mekânları bölmek için bir treshold (eşik) değeri bulunarak mekânlar daha verimli kullanılmaya çalışılmıştır. Problemle ilgili çalışmalarda bir çok matematiksel ve sezgisel yaklaşımlar geliştirilmiştir. Örneğin: Hill-Climbing (Sharma, ve Kumar, 1985), Simulated Annealing (Burke ve Varley, 1998), Genetic Algorithms (Burke ve Varley, 1999), Greedy Randomized Adaptive Search Procedure (GRASP) (Schaerf, 1999), Guided Local Search (GLS) (Burke vd, 2001), Iterated Local Search (ILS) (Burke, Cowling ve Landa Silva, 2001), Variable Neighborhood Search (VNS) (Burke vd., 2001), Treshold Acceptance Algorithms (Burke vd,

2001), Tabu Search (TS) (Silva, 2003) gibi birçok yaklaşımlar geliştirilmiştir. Bu problem sadece derslikler için değil diğer alanlar içinde çözülmeye çalışılmıştır. Fakat üniversiteler için temel fiziksel kaynak eğitim ve öğretim için kullanılan mekânlardır. Burke vd. (2001) alan tahsisini “mekândaki alanları tahsis etme örneğinin derslikler, birçok gereksinim ve kısıtı sağlamayı gerektirir.” şeklinde tanımlamıştır ve Burke ve Varley (1998) alan tahsisini belirli fonksiyonlarına göre alanları tahsis etme olarak tanımlamışlardır.



## 11. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yürütülen üst-sezgisel çalışmasında üç farklı memetik algoritma, kararlı-hal (steady-state) memetik algoritması (SSMA), nesillerarası (transgenerational) memetik algoritması (TGMA), çoklumeme (multimeme) memetik algoritması (MMA) ve kendi uyarlayan/yapılandıran kararlı-hal çoklumeme (self-adaptive steady-state multimeme) memetik algoritması (SSMMA) performansı Hyflex yazılımı üzerinde test edilmiştir. Algoritmaların pseudo kodları aşağıdaki gibidir;

Algoritma 6: Steady-state memetik algoritması

```
1: Create a population of popSize random individuals.
// Apply hill-climbing on each individual
2: for i = 1 : popSize do
3: Op ID = Random-Choice(Hill-Climbing-Operators)
4: Ind(i) Apply-Hill-Climbing(OP ID, Ind(i))
5: end for
6: while termination criteria is not satisfied do
7: Parent1 Select-Parent(Population, tour-size)
8: Parent2 Select-Parent(Population, tour-size)
9: Op ID = Random-Choice(Crossover-Operators)
10: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
11: Op ID = Random-Choice(Mutation-Operators)
12: Offspr Apply-Mutation(Op ID, Offspr)
13: Op ID = Random-Choice(Hill-Climbing-Operators)
14: Offspr Apply-Hill-Climbing(Op ID, Offspr)
15: Replacement: Replace the worst individual by Offspr
16: end while
```

Algoritma 7: Transgenerational memetik algoritması

```
1: Create a population of popSize random individuals.
// Apply hill-climbing on each individual
2: for i = 1 : popSize do
3: OP ID = Random-Choice(Hill-Climbing-Operators)
4: Ind(i) Apply-Hill-Climbing(OP ID, Ind(i))
5: end for
6: while termination criteria is not satisfied do
7: for i = 1 : popSize - 1 do
8: Parent1 Select-Parent(Population, tour-size)
9: Parent2 Select-Parent(Population, tour-size)
10: Op ID = Random-Choice(Crossover-Operators)
11: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
12: Op ID = Random-Choice(Mutation-Operators)
13: Offspr Apply-Mutation(Op ID, Offspr)
14: Op ID = Random-Choice(Hill-Climbing-Operators)
15: Offspr Apply-Hill-Climbing(Op ID, Offspr)
16: Add(Offspr, Offspring-Pool)
17: end for
18: Replacement: Keep the best individual in Population and replace the
19: rest with Offspring--Pool
20: end while
```

Algoritma 8: Multimeme steady-state memetik algoritması

```
1: Create a population of popSize random individuals.
// Apply hill-climbing on each individual
2: for i = 1 : popSize do
3: OP ID = Random-Choice(Hill-Climbing-Operators)
4: Ind(i) Apply-Hill-Climbing(OP ID, Ind(i))
5: end for
6: while termination criteria is not satisfied do
7: for i = 1 : popSize - 1 do
8: Parent1 Select-Parent(Population, tour-size)
9: Parent2 Select-Parent(Population, tour-size)
10: CROSSOVER(Parent1,Parent2)
11: if IR <= 0.2 then
12: Op ID = Random-Choice(Mutation-Operators)
13: Offspr Apply-Mutation(Op ID, Offspr)
14: else
15: Op ID = S Meme.Mutation-Operator
16: Offspr Apply-Mutation(Op ID, Offspr)
17: end if
18: Op ID = S Meme.Hill-Climbing-Operator
19: Offspr Apply-Hill-Climbing(Op ID, Offspr)
20: Add(Offspr, Offspring-Pool)
21: end for
22: Replacement: Replace the worst individual by Offspr
23: end while
24: function CROSSOVER(Parent1,Parent2)
25: if both parents carrie the same meme then
26: Op ID = Meme.Crossover-Operator
27: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
28: Offspr inherits common Meme
29: else if Parent1.fitness==Parent2.Fitness then
30: S Meme=Random-Choice(Parent1.Meme, Parent2.Meme)
31: Op ID = S Meme.Crossover-Operator
32: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
33: Offspr inherits S Meme
34: else
35: S Meme=BestMeme(Parent1.Meme, Parent2.Meme)
36: Op ID = S Meme.Crossover-Operator
37: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
38: Offspr inherits S Meme
39: end if
40: end function
```

Algoritma 9: Self-adaptif steady-state multimeme memetik algoritması

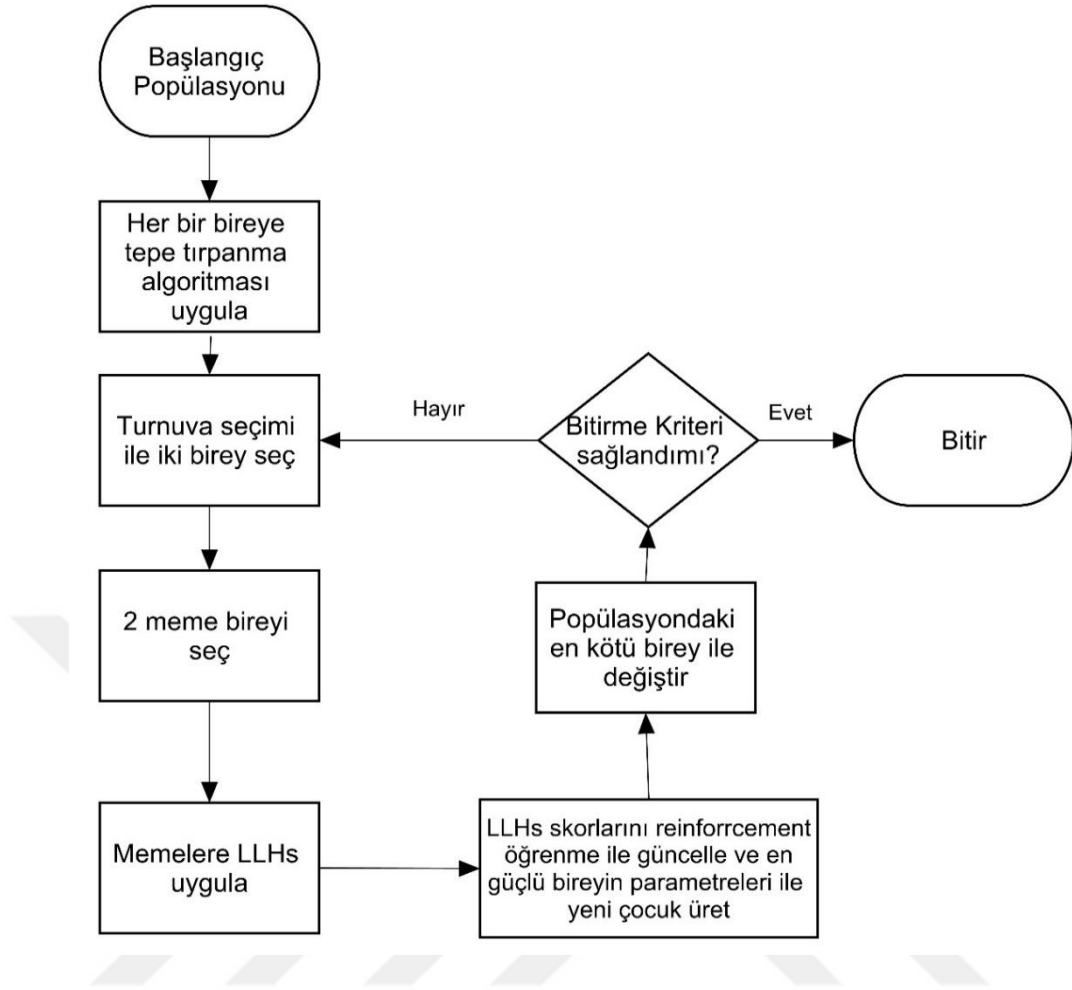
```
1: Create a population of popSize random individuals.
// Apply hill-climbing on each individual
2: for i = 1 : popSize do
3: OP ID = Random-Choice(Hill-Climbing-Operators)
4: Ind(i) Apply-Hill-Climbing(OP ID, Ind(i))
5: end for
6: while termination criteria is not satisfied do
7: for i = 1 : popSize - 1 do
8: Parent1 Select-Parent(Population, tour-size)
9: Parent2 Select-Parent(Population, tour-size)
10: S Meme = SelectMeme(Parent1,Parent2) < S Meme represents
11: Selected Meme
12: Op ID = Tournament Select(S Meme.Crossover-Operators)
13: Offspr Apply-Crossover(Op ID, Parent1, Parent2)
```

```

14: UPDITEMEME(S Meme, Op ID, newfitness, oldfitness)
15: Op ID = Tournament Select(S Meme.Mutation-Operators)
16: Offspr Apply-Mutation(Op ID, Offspr)
17: UPDITEMEME(S Meme, Op ID, newfitness, oldfitness)
18: Op ID =Tournament Select(S Meme.Hill-Climbing-Operators)
19: Offspr Apply-Hill-Climbing(Op ID, Offspr)
20: UPDITEMEME(S Meme, Op ID, newfitness, oldfitness)
21: Add(Offspr, Offspring-Pool)
22: < Tournament Select is based on numberofoperatorbestscore of
23: S Meme.Operators
24: end for
25: Replacement: Replace the worst individual by Offspr
26: end while
27: function SELECTMEME(Parent1,Parent2)
28: if both parents carrie the same meme then
29: Meme ID = Meme.Crossover-Operator
30: else if Parent1.fitness==Parent2.Fitness then
31: SelectedMeme=Random-Choice(Parent1.Meme, Parent2.Meme)
32: Meme ID = SelectedMeme.Crossover-Operator
33: else
34: SelectedMeme=BestMeme(Parent1.Meme, Parent2.Meme)
35: Meme ID= SelectedMeme.Crossover-Operator
36: end if
37: end function
38: function UPDITEMEME(S Meme,Op ID, newfitness, oldfitness)
39: if newfitness:=oldfitness then
40: numberofoperatorworstscore++
41: else
42: numberofoperatorbestscore++
43: end if
44: end function

```

Şekil 11.1’de SSMMA rasgele bir başlangıç popülasyonu üreterek problem çözümüne başlar. Başlangıç popülasyonunun her bir bireyine ilk önce yerel arama sezgiselleri uygulanır. Elde edilen bireylerden 2 tane turnuva seçimi yöntemiyle seçilir ve daha iyi olan atanın memetik özellikleri alınır. Bu iki birey sırasıyla çaprazlama, mutasyon ve yerel arama sezgiselleri uygulanarak yeni bireyler elde edilir. Bu aşamalarda yine memetik özellikler sürekli güncellenir. Eğer her bir aşama sonrasında elde edilen birey daha iyi ise memetik özelliklerdeki sezgisel skoru bir artırılır. Sonra en kötü birey ile elde edilen daha iyi birey değiştirilir.



Şekil 11.1. SSMMA algoritması akış diyagramı

SSMA, TGMA, MultiMeme SSMA ve SSMMA algoritmaları CHeSC2011 yazılımı üzerindeki düşük-seviyeli sezgiseller kullanılarak çalıştırılmıştır. Her bir problemde çaprazlama, mutasyon ve yerel arama operatörleri kullanılmıştır. Bu operatörlerin listesi Çizelge 11.1 deki gibidir.

Çizelge 11.1. Her bir Problemdeki Düşük Seviyeli Sezgisellerin Sayısı {Mutasyon (MU) Yeniden Oluşturma(RE-CREATE) ile birlikte, Yerel arama (HC), Çaprazlama (XO)}

| Problem | MU | HC | XO | Toplam |
|---------|----|----|----|--------|
| SAT     | 7  | 2  | 2  | 11     |
| BP      | 5  | 2  | 1  | 8      |
| PS      | 4  | 5  | 3  | 12     |
| PFS     | 7  | 4  | 4  | 15     |
| TSP     | 6  | 3  | 4  | 13     |
| VRP     | 5  | 3  | 2  | 10     |

SSMA algoritmasında başlangıç popülasyonu rasgele oluşturulur (Algoritma 6). Evrimsel süreçte ise turnuva yöntemi ile seçilen iki birey üzerine çaprazlama, mutasyon ve yerel arama operatörleri uygulanır. Elde edilen yeni birey popülasyondaki en kötü birey ile değiştirilir. TSMA algoritmasında ise başlangıç popülasyonu rasgele oluşturularak seçilen bireyler üzerine sırasıyla çaprazlama, mutasyon ve yerel arama uygulanarak popülasyon sayısı kadar birey oluşturulur ve bu bireyler en iyi 1 tanesi elitizim yapılarak diğerleri değiştirilir (Algoritma 7). Multimeme memetik algoritmasında bireyler hem genetic hemde memetik materyaller içerir. Seçilen bireyin memetik materyalide seçilerek bir sonraki döngüde bu memetik materyalin özellikleri kullanılır (Algoritma 8, 9).

Memetik Algoritmaların performansı herbir problem için 5'er örnek üzerinde test edilmiştir. Seçilen arametre değerleri Çizelge 11.2' deki gibidir.

Çizelge 11.2. Parametre Değerleri

| Parametre               | Değer          |
|-------------------------|----------------|
| Populasyon Başlangıcı   | Rasgele        |
| Seçim Mekanizması       | Turnuva Seçimi |
| Tur Boyutu              | 2              |
| Populasyon Boyutu       | 10             |
| Genetik Operator Seçimi | uniform random |
| Mutasyon Oranı          | 0.2            |
| Yerel Arama Seçimi      | uniform random |
| Yerel Arama Oranı       | 0.2            |
| Sonlanma Şartı          | 600 saniye     |

Çalışmada her bir örnek için algoritma 31 kez çalıştırılmıştır ve performanslar Wilcoxon testi ile karşılaştırılmıştır. Çalışmalar sonucunda algoritmalar daha önce yarışmadaki algoritmalar ile karşılaştırılmıştır ve sıralama Çizelge 11.3 ve 11.4 deki gibidir.

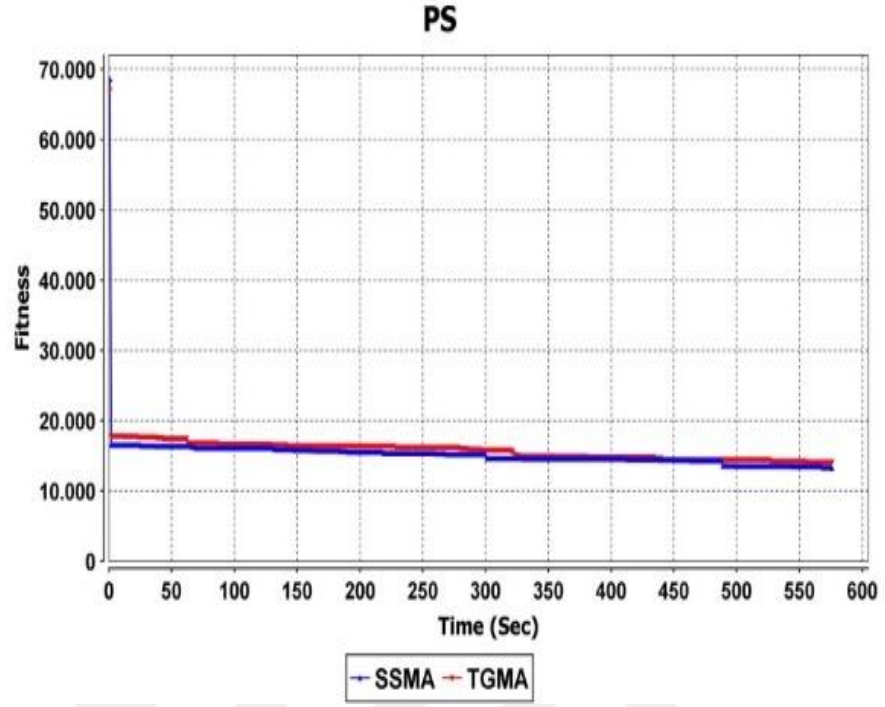
Çizelge 11.3. Memetik Algoritmaların CHESC2011 yarışmasındaki sıralaması

| Algoritma                | Skor         |
|--------------------------|--------------|
| 1 AdapHH                 | 181          |
| 2 VNS-TW                 | 134          |
| 3 ML                     | 131.5        |
| 4 PHunter                | 93.25        |
| 5 EPH                    | 89.25        |
| 6 HAHA                   | 75.75        |
| 7 NAHH                   | 75           |
| 8 ISEA                   | 71           |
| 9 KSATS-HH               | 66           |
| 10 HAEA                  | 53.5         |
| 11 ACO-HH                | 39           |
| 12 GenHive               | 36.5         |
| 13 DynILS                | 27           |
| <b>14 Multimeme-SSMA</b> | <b>24.75</b> |
| 15 SA-ILS                | 24.25        |
| 16 XCJ                   | 22.5         |
| 17 AVEG-Nep              | 21           |
| 18 GISS                  | 16.75        |
| <b>19 SSMA</b>           | <b>7.5</b>   |
| 20 SelfSearch            | 7            |
| 21 MCHH-S                | 4.75         |
| 22 Ant-Q                 | 0            |
| <b>23 TGMA</b>           | <b>0</b>     |

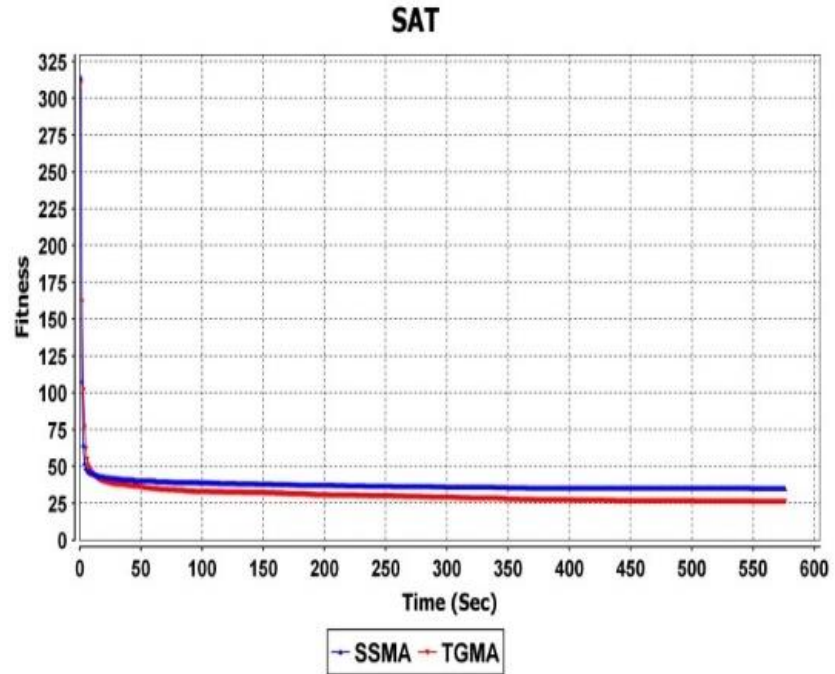
Çizelge 11.4. Self-Adaptif Memetik Algoritmasının CHESC2011 yarışmasındaki sıralaması

| Algoritma      | Skor         |
|----------------|--------------|
| 1 AdapHH       | 175.1        |
| 2 VNS-TW       | 126.6        |
| 3 ML           | 123          |
| 4 PHunter      | 89.25        |
| 5 EPH          | 84.75        |
| <b>6 SSMMA</b> | <b>75.85</b> |
| 7 NAHH         | 70.5         |
| 6 HAHA         | 68.85        |
| 8 ISEA         | 64.5         |
| 9 KSATS-HH     | 58.35        |
| 10 HAEA        | 51           |
| 11 ACO-HH      | 37           |
| 12 GenHive     | 34.5         |
| 14 DynILS      | 25           |
| 15 SA-ILS      | 22.25        |
| 16 XCJ         | 20.5         |
| 17 AVEG-Nep    | 17.5         |
| 18 GISS        | 16.25        |
| 19 SelfSearch  | 6            |
| 20 MCHH-S      | 3.250        |
| 21 Ant-Q       | 0            |

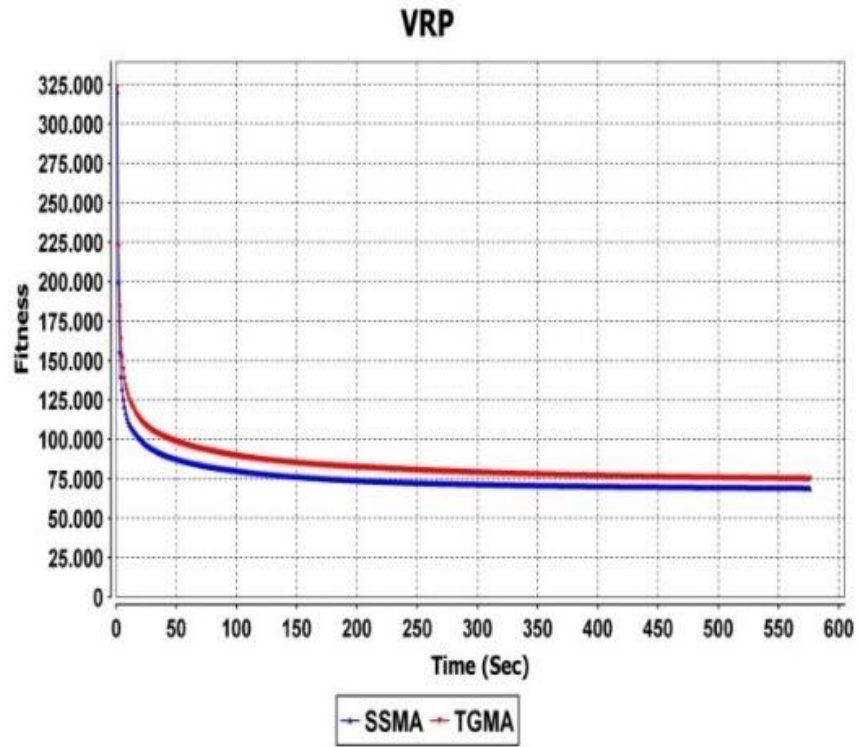
Çalışmalardaki algoritmaların problemleri minimize ederken bulduğu fonksiyon değerleri Şekil 11.2’de gösterilmiştir.



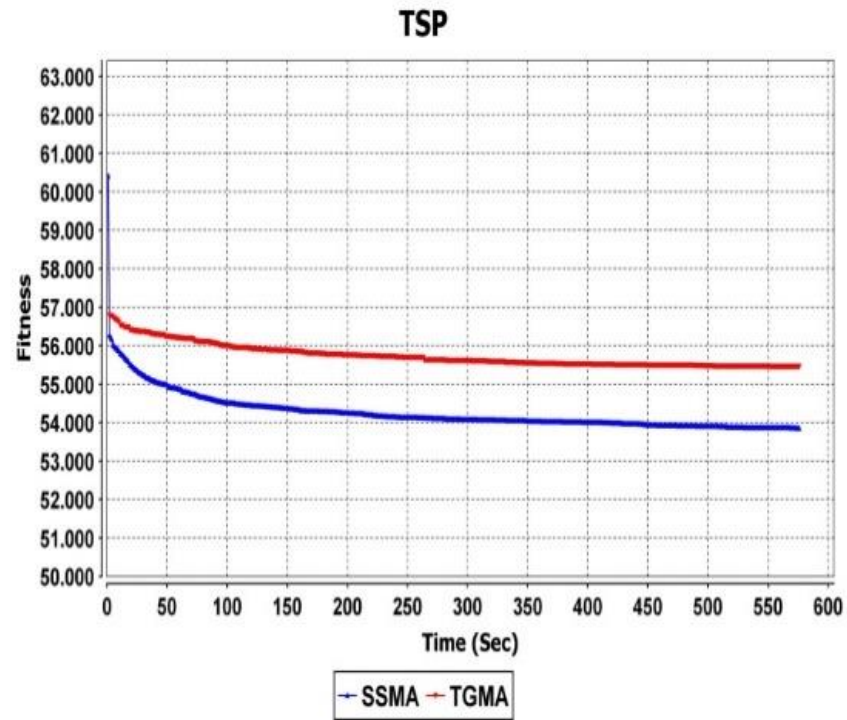
a) Örnek 3



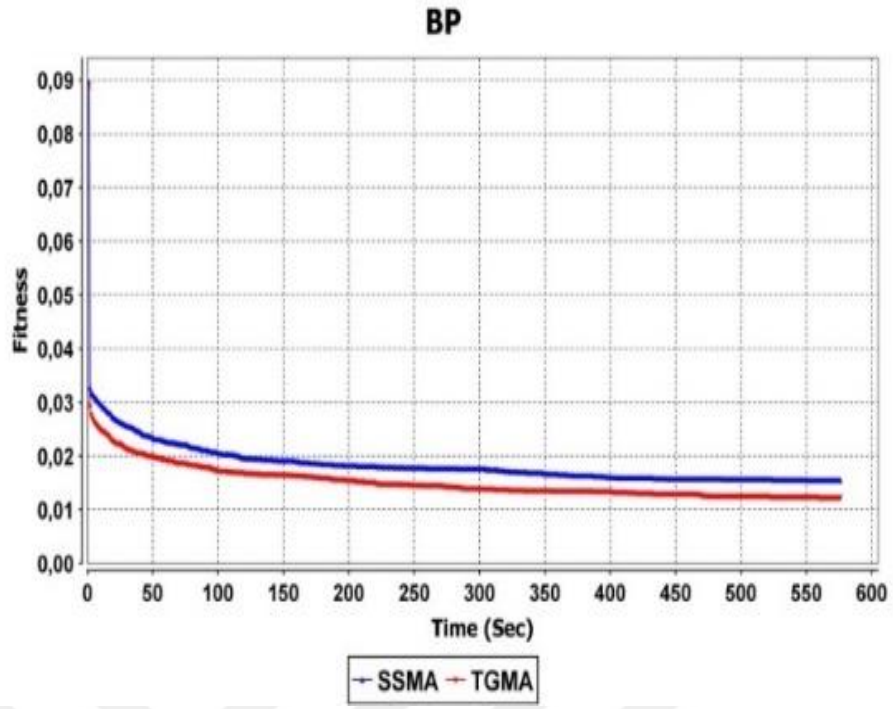
b) Örnek 3



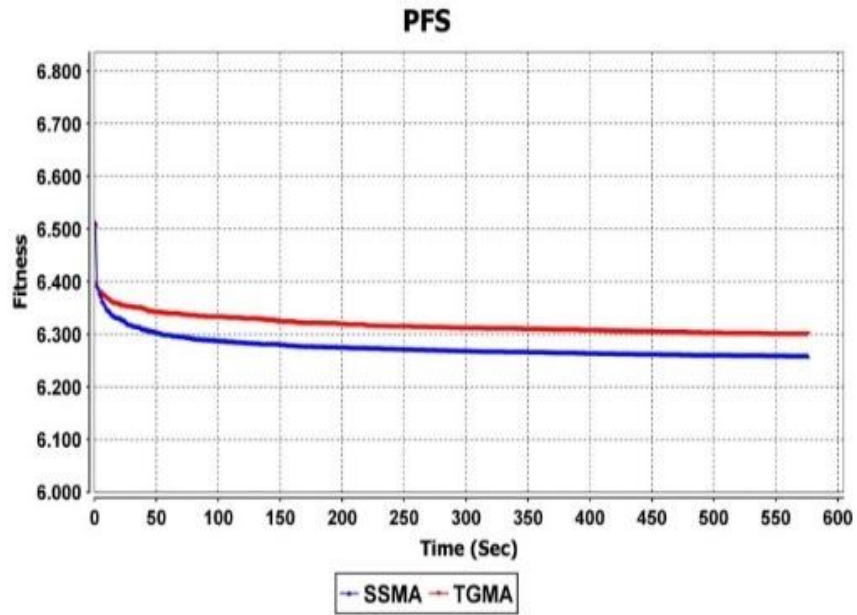
(c) Örnek 1



(d) Örnek 5



(e) Örnek 2



(e) Örnek 1

Şekil 11.2. SSMA ve TGMA 31 çalışma sonunda elde edilen ortalama değerleri

Multimeme Memetik Algoritması ve Self-Adaptif Memetik Algoritması HyFlex frameworku üzerinde çalıştırılarak daha önce elde edilen sonuçlar ile karşılaştırılmıştır. Sonuçların her bir problem için karşılaştırması Çizelge 11.5, 11.6 ve 11.7 deki gibidir.

Çizelge 11.5. SSMMA ile Multi-Meme SSMA CHESC2011 yarışmasındaki karşılaştırması

|         |     | Self-Adaptive<br>Meme MA | Multi-           |    | Multi-Meme SSMA   |               |
|---------|-----|--------------------------|------------------|----|-------------------|---------------|
| Problem | ino | ortalama<br>(avr)        | en iyi (best)    | vs | ortalama<br>(avr) | en iyi (best) |
| SAT     | 1   | <b>3.129</b>             | <b>1</b>         | >  | 18.709            | 10            |
|         | 2   | <b>2.774</b>             | <b>1</b>         | >  | 51.258            | 24            |
|         | 3   | <b>2.225</b>             | <b>0</b>         | >  | 33.967            | 9             |
|         | 4   | <b>2.741</b>             | <b>1</b>         | >  | 24.870            | 15            |
|         | 5   | <b>8.387</b>             | <b>7</b>         | >  | 18.032            | 14            |
| BP      | 1   | <b>0.0345</b>            | <b>0.0193</b>    | ≥  | 0.0551            | 0.0483        |
|         | 2   | <b>0.0031</b>            | <b>0.0272</b>    | ≥  | 0.0097            | 0.007         |
|         | 3   | <b>0.0077</b>            | <b>0.0038</b>    | ≥  | 0.0151            | 0.0112        |
|         | 4   | <b>0.1091</b>            | <b>0.1087</b>    | ≥  | 0.109             | 0.1088        |
|         | 5   | <b>0.0194</b>            | <b>0.0107</b>    | ≥  | 0.0262            | 0.0208        |
| PS      | 1   | <b>31.419</b>            | <b>25</b>        | ≥  | 32.451            | 26            |
|         | 2   | <b>10129.32</b>          | <b>9815</b>      | >  | 13829.45          | 10255         |
|         | 3   | <b>3232.806</b>          | <b>3160</b>      | >  | 3465.806          | 3242          |
|         | 4   | <b>1781.613</b>          | <b>1585</b>      | >  | 2045.226          | 1665          |
|         | 5   | 380.225                  | 345              | <  | 353.903           | 321           |
| PFS     | 1   | 6342.581                 | 6306             | <  | 6253.161          | 6238          |
|         | 2   | 26953.38                 | 26914            | <  | 26840.52          | 26773         |
|         | 3   | 6387.903                 | 6369             | <  | 6347.645          | 6303          |
|         | 4   | 11507.32                 | 11472            | <  | 11401.45          | 11377         |
|         | 5   | 26753.65                 | 26674            | <  | 26662.61          | 26603         |
| TSP     | 1   | 49032.385                | 48722.154        | <  | 48239.358         | 48194.92      |
|         | 2   | 2.138E7                  | 2.118E7          | <  | 2.111E7           | 2.0896        |
|         | 3   | 6996.01                  | 6955.616         | <  | 6824.295          | 6800.543      |
|         | 4   | 70430.13                 | 69284.88         | <  | 67608.168         | 66534.788     |
|         | 5   | 58200.41                 | 55752.75         | <  | 54299.139         | 53010.404     |
| VRP     | 1   | <b>69122.21</b>          | <b>66244.735</b> | >  | 72159.779         | 63373.559     |
|         | 2   | <b>12740.57</b>          | <b>12303.28</b>  | >  | 13768.666         | 13335.578     |
|         | 3   | 158509.3                 | 148839.8         | <  | 148975.804        | 144089.757    |
|         | 4   | <b>20670.185</b>         | <b>20655.206</b> | >  | 21321.299         | 20656.798     |
|         | 5   | 152092.27                | 148151.9         | <  | 150243.193        | 148397.623    |

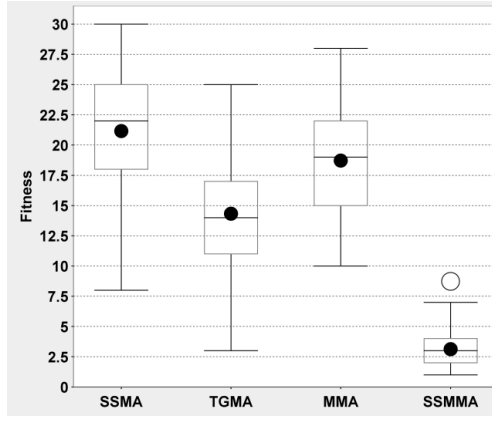
Çizelge 11.6. SSMMA ile SSMA CHESC2011 yarışmasındaki karşılaştırması

|        |     | Self-Adaptive<br>Meme MA | Multi-           |    | SSMA              |                  |
|--------|-----|--------------------------|------------------|----|-------------------|------------------|
| domain | ino | ortalama<br>(avr)        | en iyi (best)    | vs | ortalama<br>(avr) | en iyi (best)    |
| SAT    | 1   | <b>3.129</b>             | <b>1</b>         | >  | 21.161            | 8                |
|        | 2   | <b>2.774</b>             | <b>1</b>         | >  | 52.484            | 37               |
|        | 3   | <b>2.225</b>             | <b>0</b>         | >  | 35                | 10               |
|        | 4   | <b>2.741</b>             | <b>1</b>         | >  | 27.742            | 26               |
|        | 5   | <b>8.387</b>             | <b>7</b>         | >  | 19.194            | 14               |
| BP     | 1   | <b>0.0345</b>            | <b>0.0193</b>    | >  | 0.083             | 0.074            |
|        | 2   | <b>0.0031</b>            | <b>0.0272</b>    | >  | 0.015             | 0.115            |
|        | 3   | <b>0.0077</b>            | <b>0.0038</b>    | >  | 0.022             | 0.018            |
|        | 4   | <b>0.1091</b>            | <b>0.1087</b>    | >  | 0.111             | 0.110            |
|        | 5   | <b>0.0194</b>            | <b>0.0107</b>    | >  | 0.043             | 0.036            |
| PS     | 1   | <b>31.419</b>            | <b>25</b>        | >  | 51.129            | 37               |
|        | 2   | <b>10129.32</b>          | <b>9815</b>      | >  | 72015.613         | 52056            |
|        | 3   | <b>3232.806</b>          | <b>3160</b>      | >  | 13203.710         | 5581             |
|        | 4   | <b>1781.613</b>          | <b>1585</b>      | >  | 2942.419          | 1820             |
|        | 5   | <b>380.225</b>           | <b>345</b>       | >  | 435.645           | 385.0            |
| PFS    | 1   | 6342.581                 | 6306             | <  | <b>6257.806</b>   | <b>6231</b>      |
|        | 2   | 26953.38                 | 26914            | <  | <b>26884.935</b>  | <b>26813</b>     |
|        | 3   | 6387.903                 | 6369             | <  | <b>6351.871</b>   | <b>6318</b>      |
|        | 4   | 11507.32                 | 11472            | <  | <b>11441.806</b>  | <b>11410</b>     |
|        | 5   | 26753.65                 | 26674            | <  | <b>26699.226</b>  | <b>26626.0</b>   |
| TSP    | 1   | 49032.385                | 48722.154        | <  | <b>48227.747</b>  | <b>48194.920</b> |
|        | 2   | 2.138E7                  | 2.118E7          | <  | <b>2.116E7</b>    | <b>2.096E7</b>   |
|        | 3   | 6996.01                  | 6955.616         | <  | <b>6825.552</b>   | <b>6800.708</b>  |
|        | 4   | 70430.13                 | 69284.88         | <  | <b>68123.369</b>  | <b>67423.655</b> |
|        | 5   | 58200.41                 | 55752.75         | <  | <b>53810.138</b>  | <b>52685.992</b> |
| VRP    | 1   | <b>69122.21</b>          | <b>66244.735</b> | >  | 71768.053         | 67820.589        |
|        | 2   | <b>12740.57</b>          | <b>12303.28</b>  | >  | 14324.522         | 13358.611        |
|        | 3   | <b>158509.3</b>          | <b>148839.8</b>  | >  | 176206.081        | 167704.512       |
|        | 4   | <b>20670.185</b>         | <b>20655.206</b> | >  | 21647.018         | 20678.096        |
|        | 5   | <b>152092.27</b>         | <b>148151.9</b>  | >  | 152642.040        | 149032.551       |

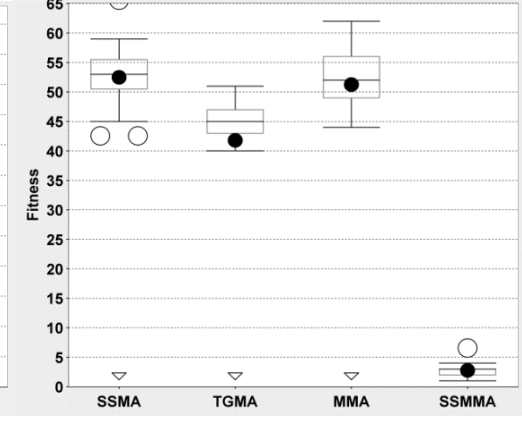
Çizelge 11.7 SSMMA ile TGMA CHESC2011 yarışmasındaki karşılaştırması

|        |     | Self-Adaptive<br>MemeMA | Multi-           |    | TGMA              |                  |
|--------|-----|-------------------------|------------------|----|-------------------|------------------|
| domain | ino | ortalama<br>(avr)       | en iyi (best)    | vs | ortalama<br>(avr) | en iyi (best)    |
| SAT    | 1   | <b>3.129</b>            | <b>1</b>         | >  | 14.323            | 3.0              |
|        | 2   | <b>2.774</b>            | <b>1</b>         | >  | 41.806            | 11.0             |
|        | 3   | <b>2.225</b>            | <b>0</b>         | >  | 27.129            | 5.0              |
|        | 4   | <b>2.741</b>            | <b>1</b>         | >  | 20.226            | 13.0             |
|        | 5   | <b>8.387</b>            | <b>7</b>         | >  | 17.290            | 13.0             |
| BP     | 1   | <b>0.0345</b>           | <b>0.0193</b>    | ≥  | 0.075             | 0.066            |
|        | 2   | <b>0.0031</b>           | <b>0.0272</b>    | ≥  | 0.012             | 0.007            |
|        | 3   | <b>0.0077</b>           | <b>0.0038</b>    | ≥  | 0.020             | 0.016            |
|        | 4   | <b>0.1091</b>           | <b>0.1087</b>    | ≥  | 0.110             | 0.109            |
|        | 5   | <b>0.0194</b>           | <b>0.0107</b>    | ≥  | 0.037             | 0.032            |
| PS     | 1   | <b>31.419</b>           | <b>25</b>        | >  | 66.419            | 50.0             |
|        | 2   | <b>10129.32</b>         | <b>9815</b>      | >  | 70356.161         | 59736.0          |
|        | 3   | <b>3232.806</b>         | <b>3160</b>      | >  | 14028.710         | 10027.0          |
|        | 4   | <b>1781.613</b>         | <b>1585</b>      | >  | 2923.710          | 2003.0           |
|        | 5   | <b>380.225</b>          | <b>345</b>       | >  | 498.032           | 430.0            |
| PFS    | 1   | 6342.581                | 6306             | <  | <b>6301.968</b>   | <b>6273.0</b>    |
|        | 2   | 26953.38                | 26914            | ≤  | <b>26944.548</b>  | <b>26889.0</b>   |
|        | 3   | 6387.903                | 6369             | <  | <b>6366.677</b>   | <b>6342.0</b>    |
|        | 4   | 11507.32                | 11472            | ≤  | <b>11499.645</b>  | <b>11458.0</b>   |
|        | 5   | 26753.65                | 26674            | <  | <b>26724.742</b>  | <b>26668.0</b>   |
| TSP    | 1   | 49032.385               | 48722.154        | <  | <b>48337.598</b>  | <b>48194.921</b> |
|        | 2   | 2.138E7                 | 2.118E7          | <  | <b>2.130E7</b>    | <b>2.114E7</b>   |
|        | 3   | 6996.01                 | 6955.616         | <  | <b>6893.822</b>   | <b>6858.803</b>  |
|        | 4   | 70430.13                | 69284.88         | <  | <b>69778.135</b>  | <b>68922.954</b> |
|        | 5   | 58200.41                | 55752.75         | <  | <b>55463.600</b>  | <b>54052.398</b> |
| VRP    | 1   | <b>69122.21</b>         | <b>66244.735</b> | >  | 76331.070         | 71560.119        |
|        | 2   | <b>12740.57</b>         | <b>12303.28</b>  | >  | 13869.191         | 13333.0914       |
|        | 3   | <b>158509.3</b>         | <b>148839.8</b>  | >  | 200838.192        | 193416.831       |
|        | 4   | <b>20670.185</b>        | <b>20655.206</b> | >  | 21412.571         | 20659.896        |
|        | 5   | <b>152092.27</b>        | <b>148151.9</b>  | >  | 157001.253        | 153557.350       |

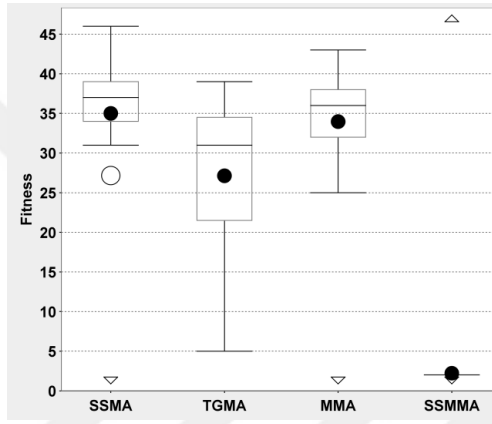
Yapılan deneylerin sonucu aşağıdaki şekillerde (Şekil 11.3, 11.4, 11.5, 11.6, 11.7, 11.8) ise üzerinde çalışılan memetik algoritmaların box ve whiskers grafikleri üzerinde karşılaştırılması gösterilmiştir.



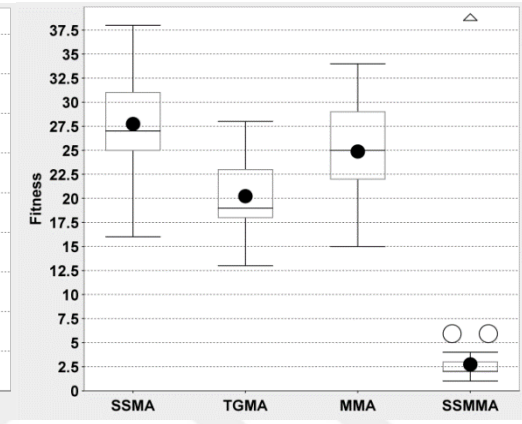
(a) Örnek 1



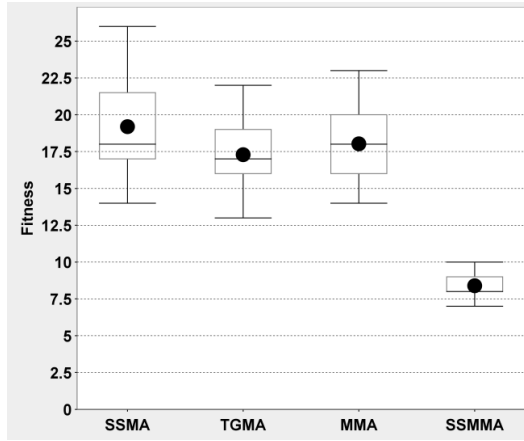
(b) Örnek 2



(c) Örnek 3

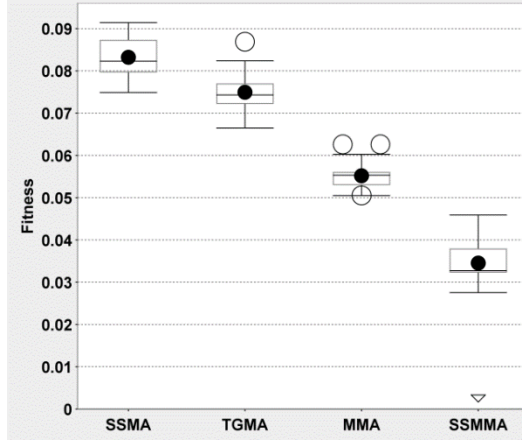


(d) Örnek 4

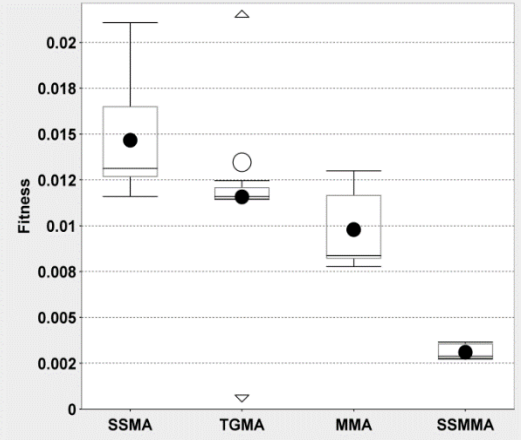


(e) Örnek 5

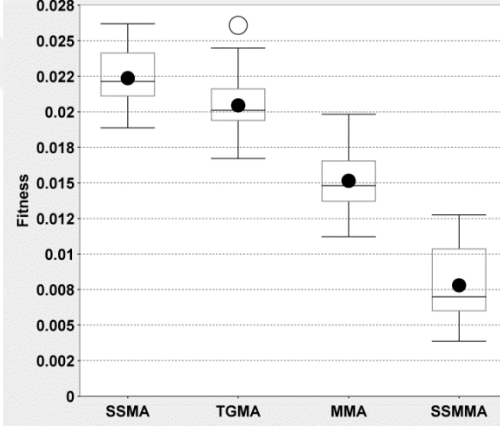
Şekil 11.3. Max-Sat problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA



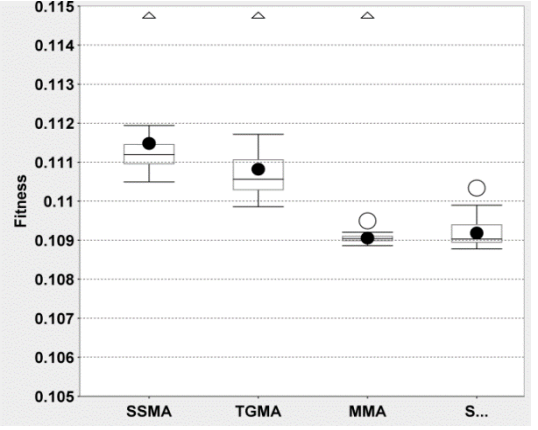
(a) Örnek 1



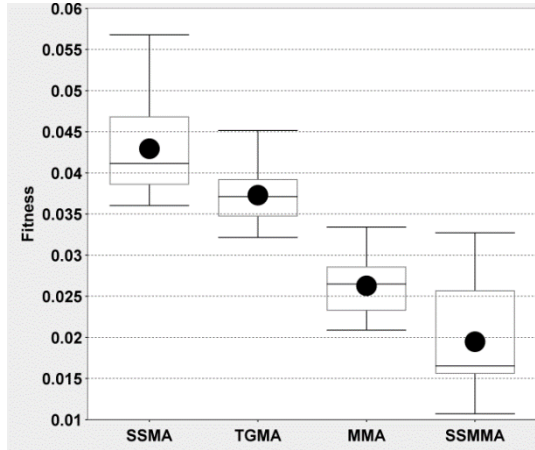
(b) Örnek 2



(c) Örnek 3

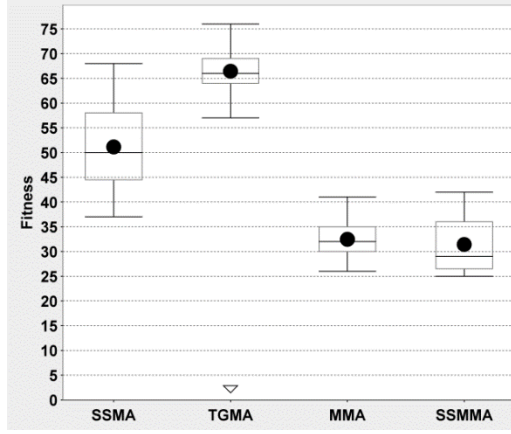


(d) Örnek 4

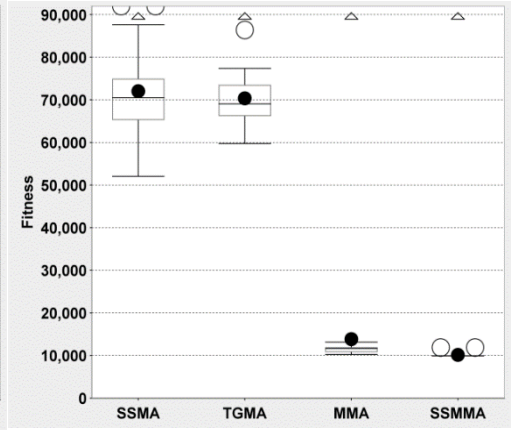


(e) Örnek 5

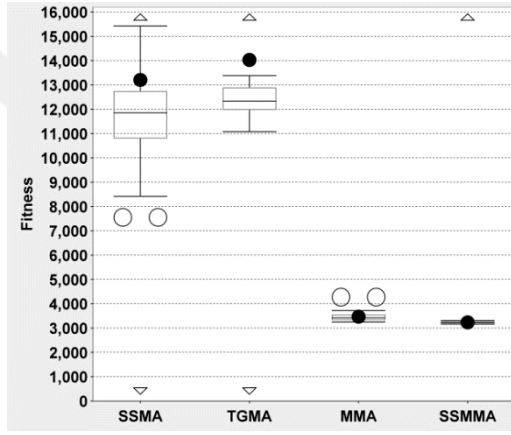
Şekil 11.4. Bin Packing problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA



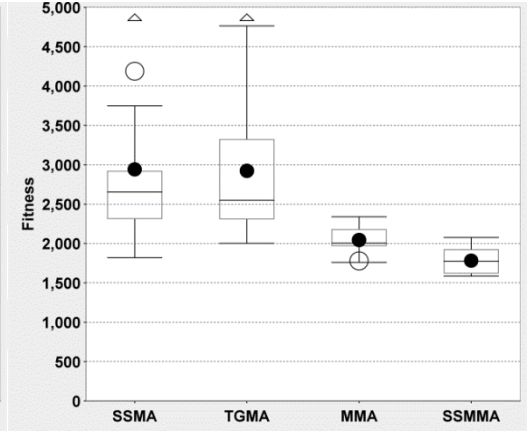
(a) Örnek 1



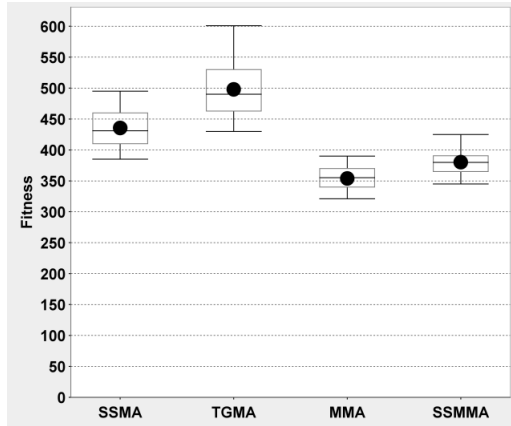
(b) Örnek 2



(c) Örnek 3

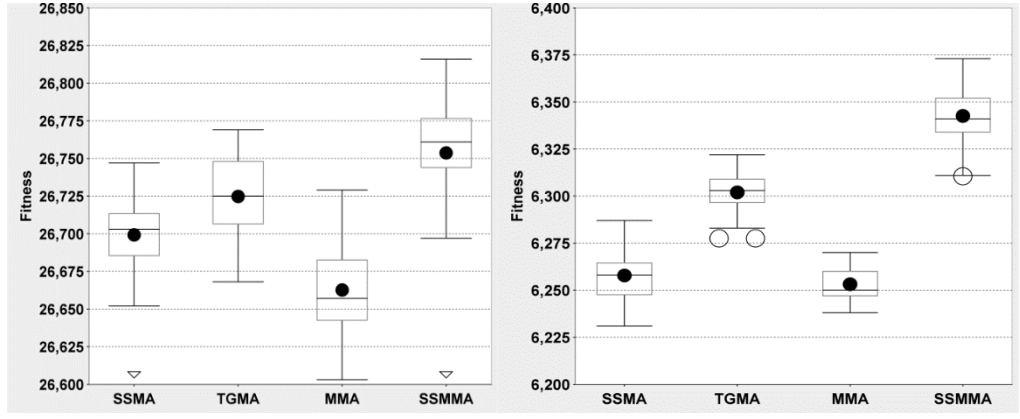


(d) Örnek 5



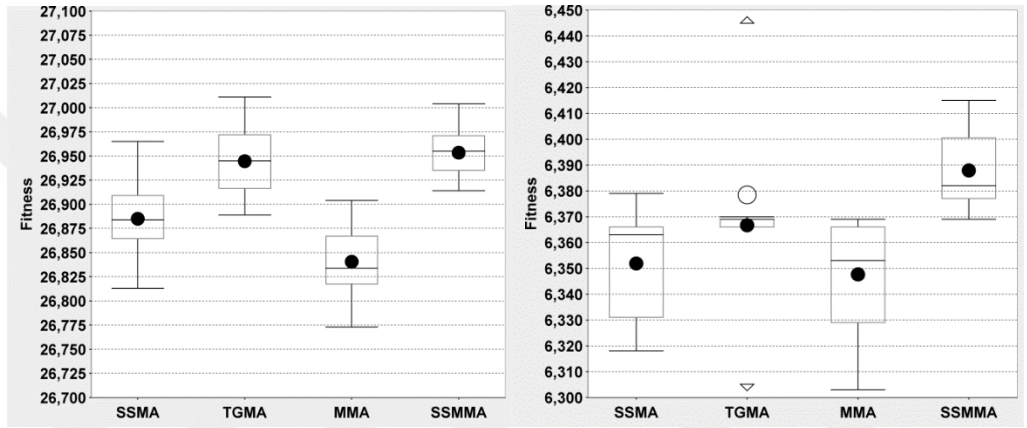
(e) Örnek 5

Şekil 11.5. Personel Çizelgeleme problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA



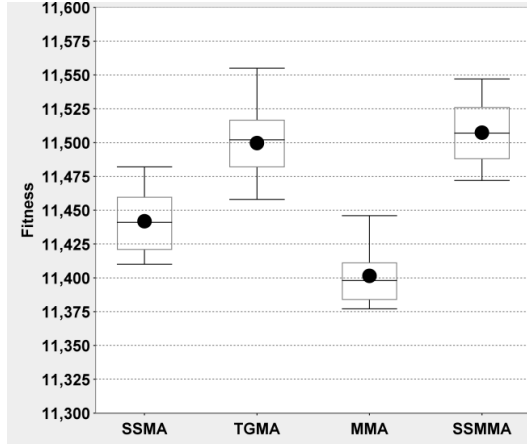
(a) Örnek 1

(b) Örnek 2



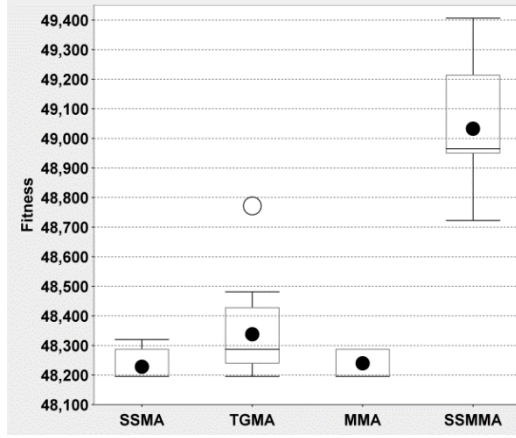
(c) Örnek 3

(d) Örnek 4

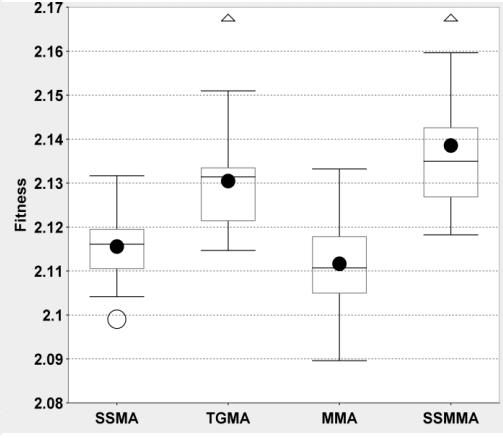


(e) Örnek 5

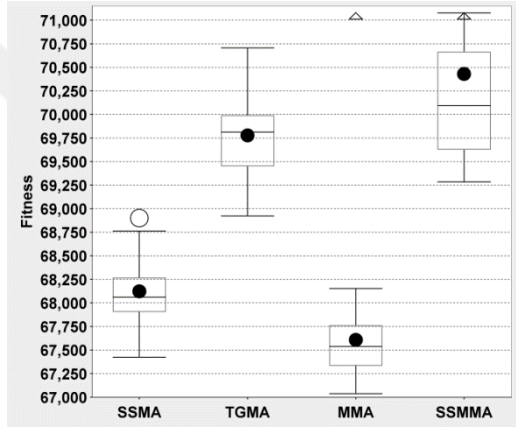
Şekil 11.6. Akış Tipi İş Çizelgeleme problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA



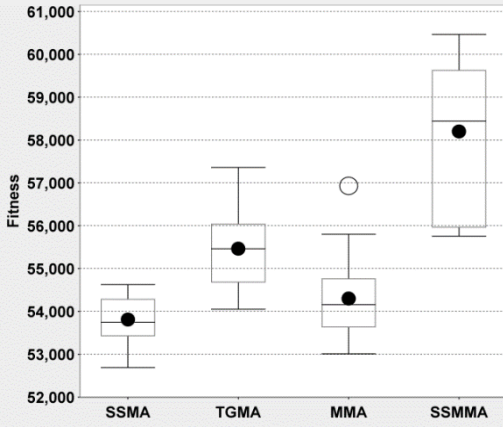
(a) Örnek 1



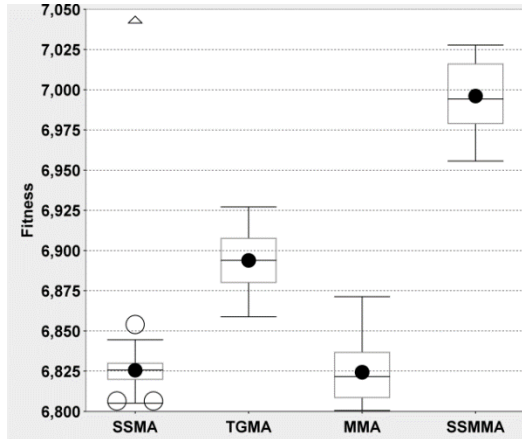
(b) Örnek 2



(c) Örnek 3

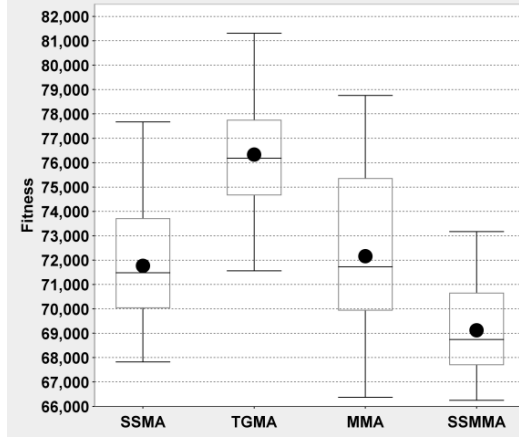


(d) Örnek 4

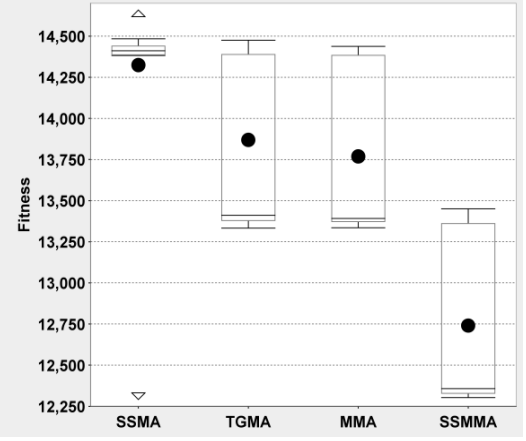


(e) Örnek 5

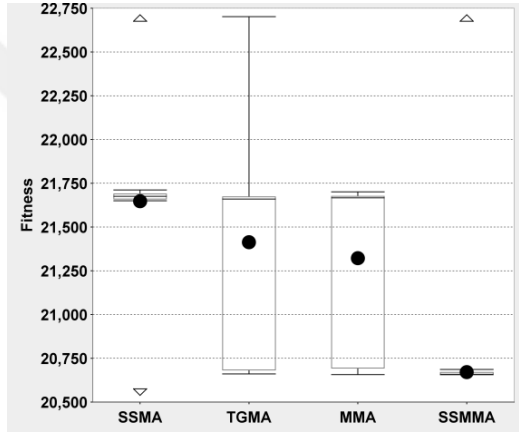
Şekil 11.7. Gezgin Satıcı problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA



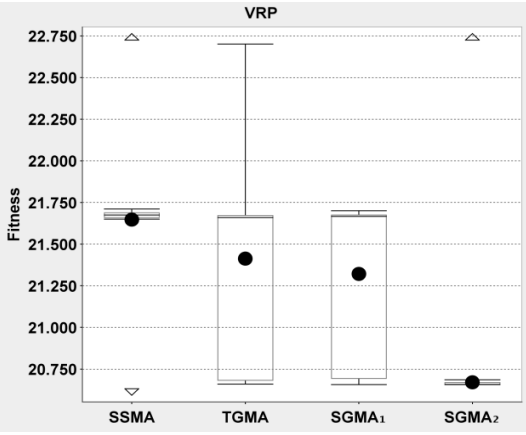
(a) Örnek 1



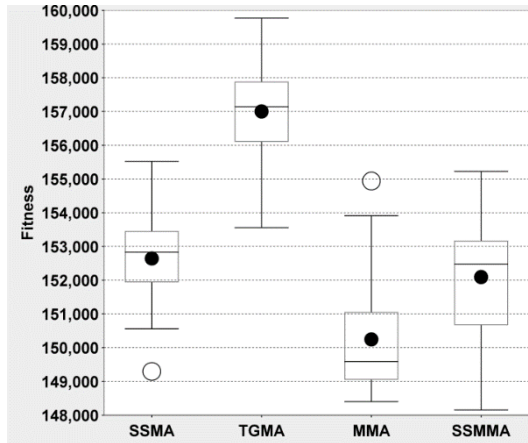
(b) Örnek 2



(c) Örnek 3



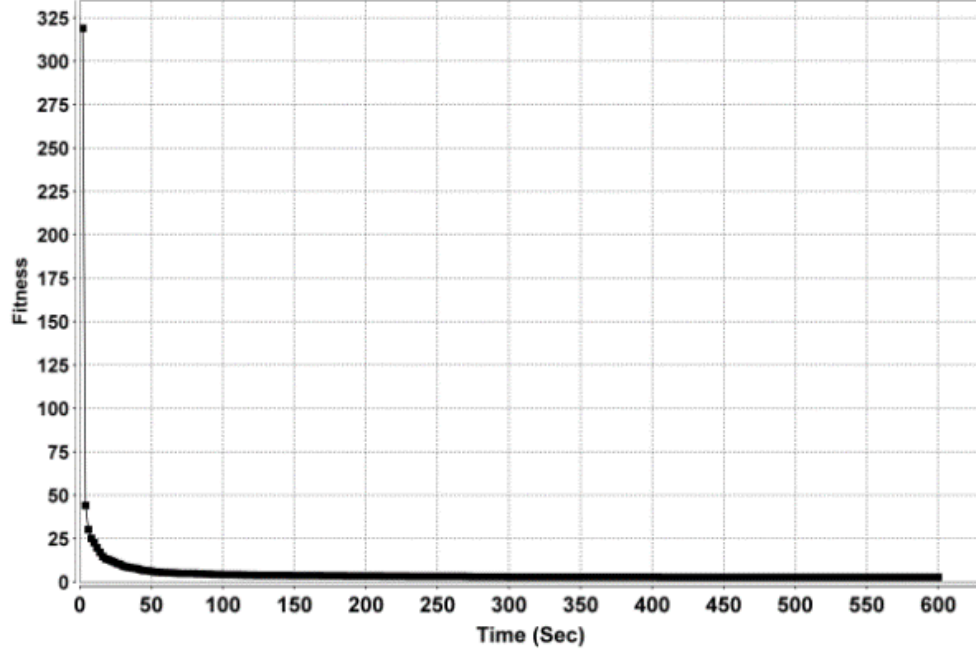
(d) Örnek 4



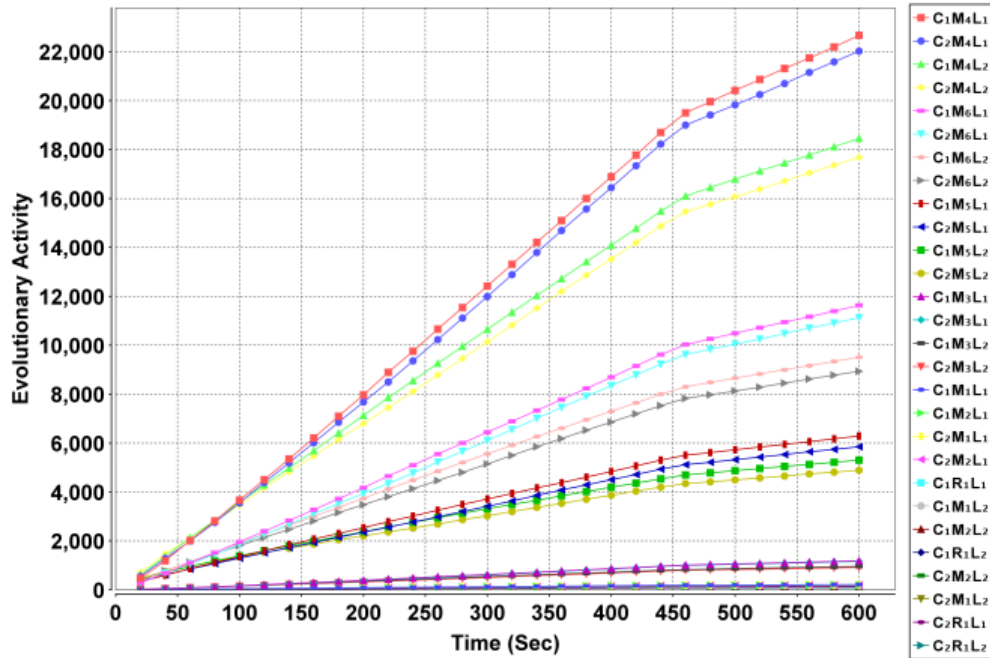
(e) Örnek 5

Şekil 11.8. Araç Rotalama problemi için Memetik Algoritmaların performanslarının karşılaştırılması: SSMA, TGMA, MMA, SSMMA

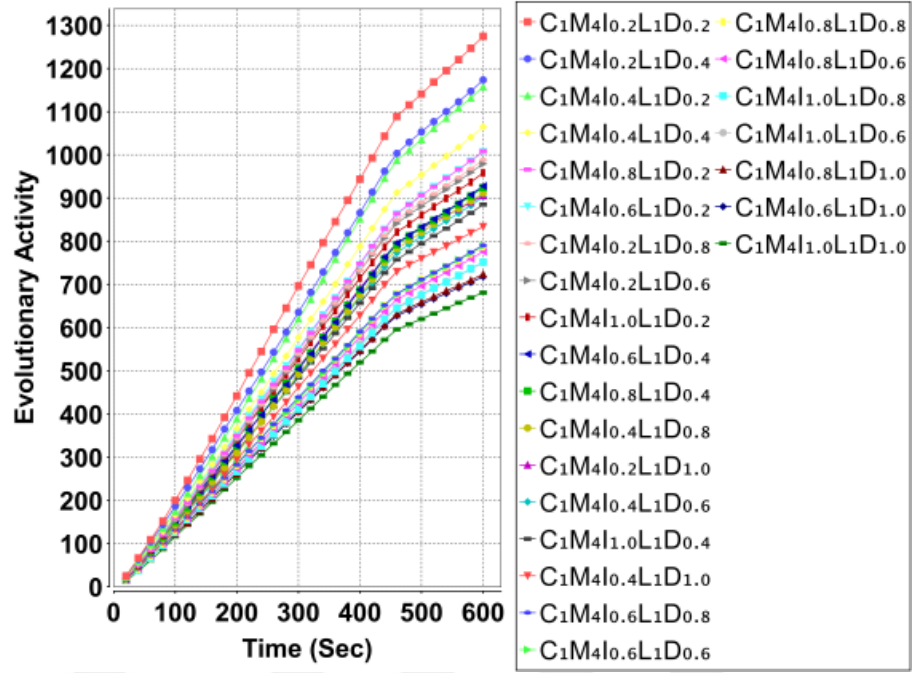
SSMMA algoritmasının çalışması sonucunda algortmada kullanılan skorlar grafiklerle aşağıda gösterilmiş olup amaç fonksiyonunun zaman içerisindeki değişen ağırlık değerleri (Şekil 11.9, 11.10, 11.12, 11.13, 11.14) çizilmiştir.



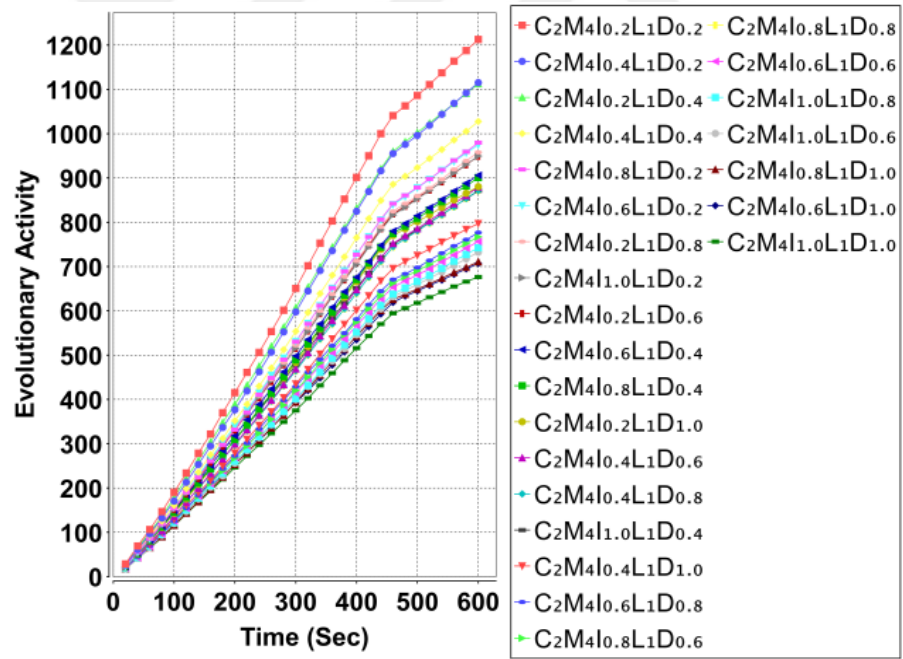
(a) Örnek 3 için ağırlık değişim grafiği



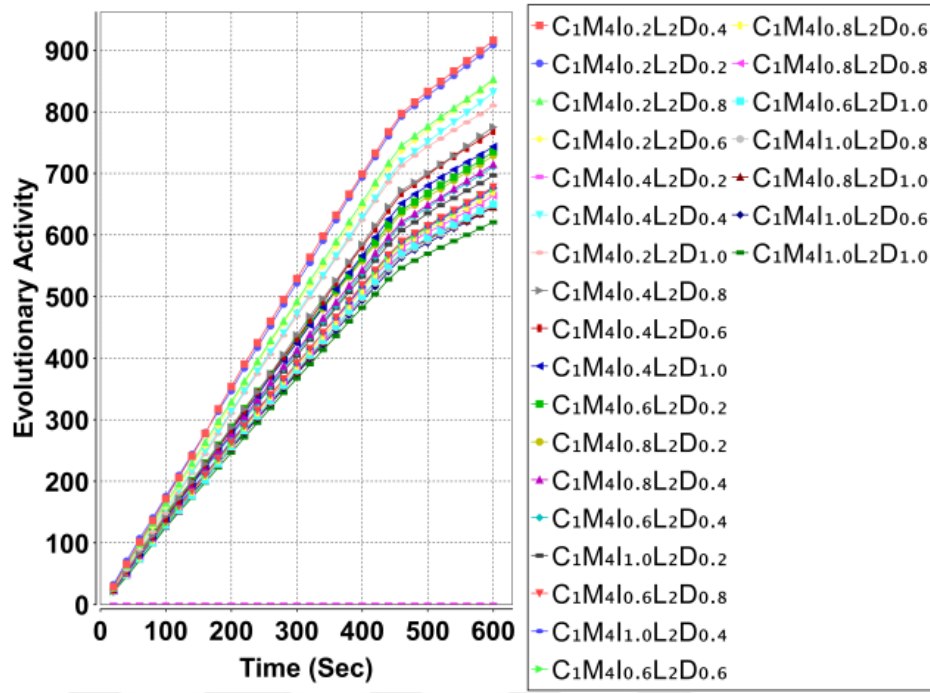
(b) Çaprazlama, Mutasyon ve Yerel arama için seçim grafiği



(c) C1M4L1

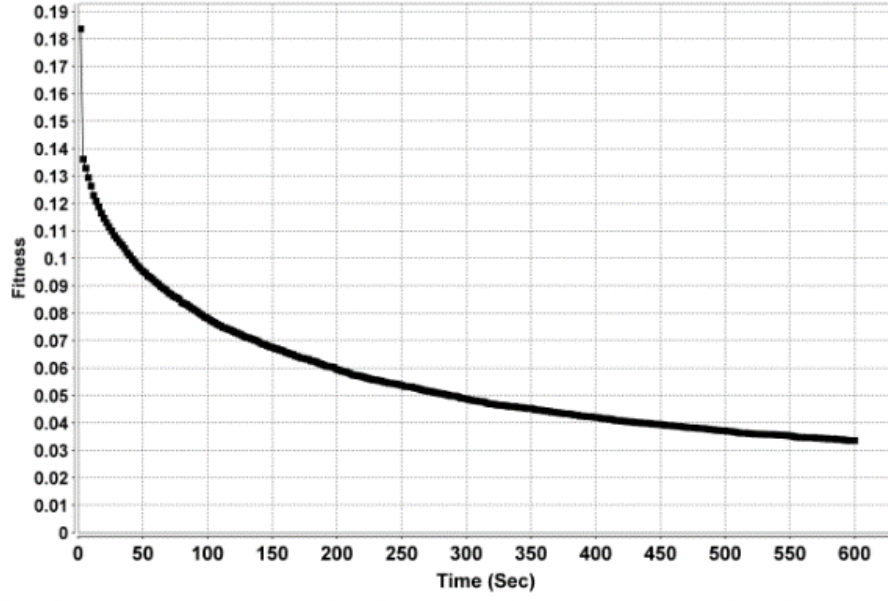


(d) C2M4L1

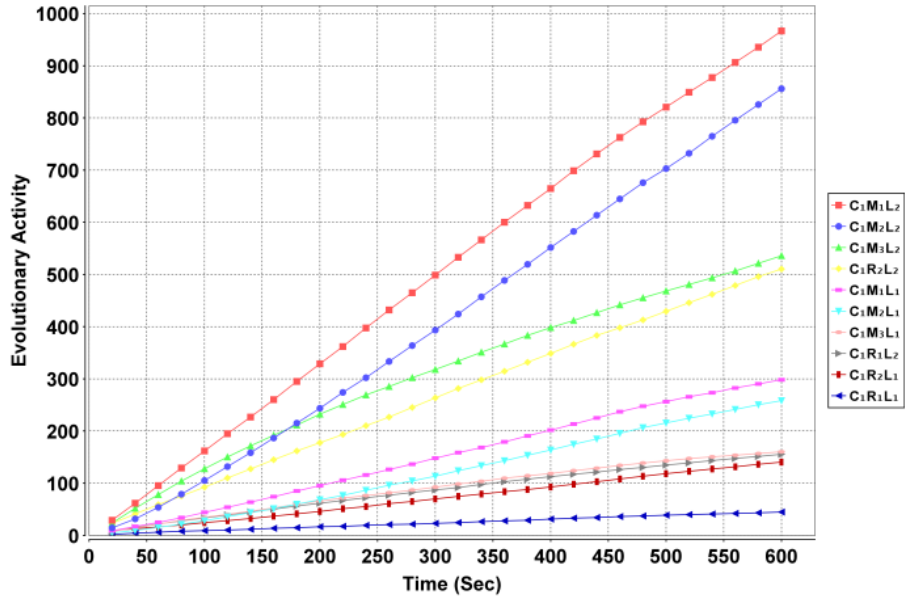


(e) C1M4L2

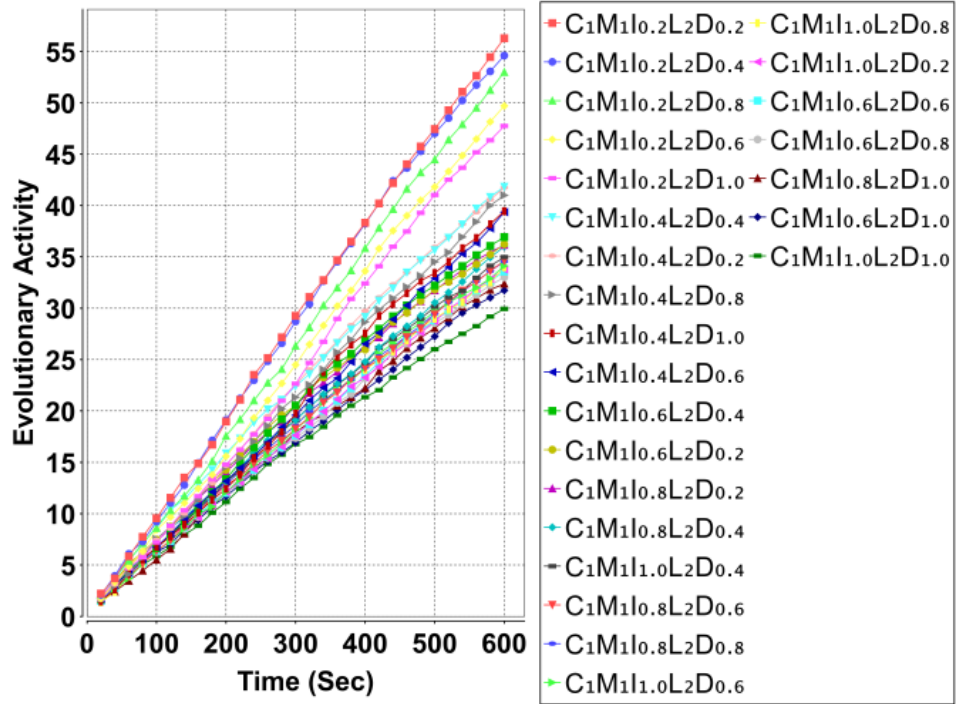
Şekil 11.9. SSMMA algoritması ile rasgele seçilen Max-SAT probleminin 31 deney sonucuna göre evrimsel aktivitesi



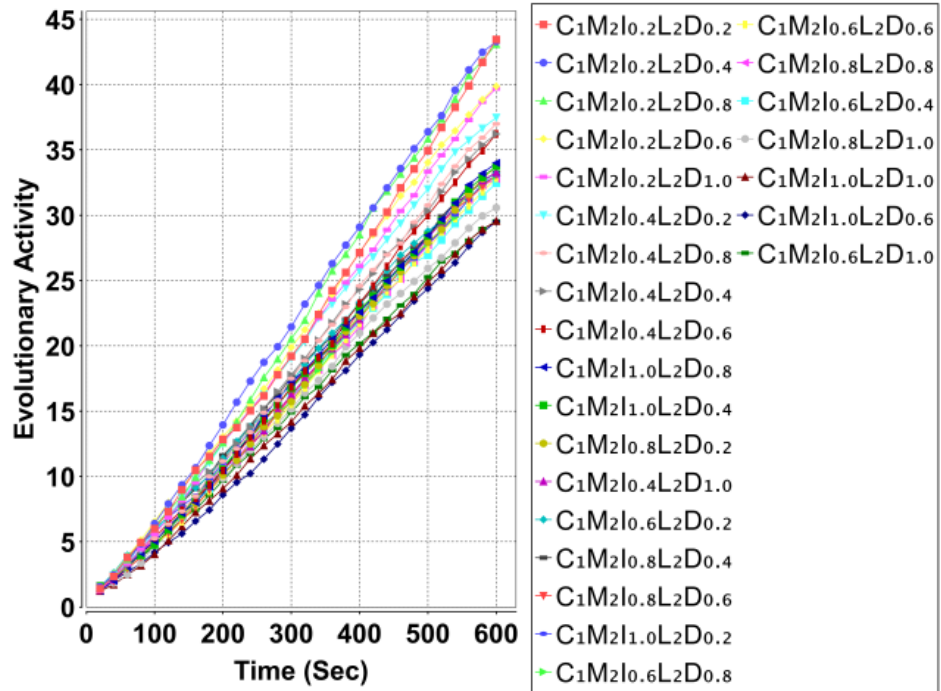
(a) Örnek 3 için ağırlık değişim grafiği



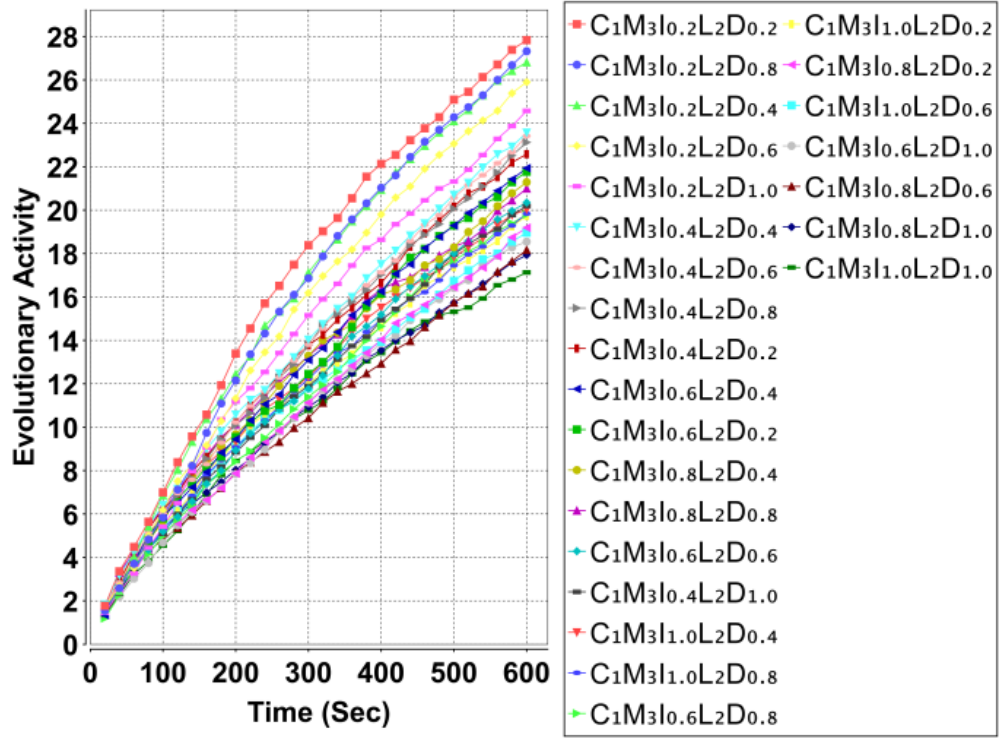
(b) Çaprazlama, Mutasyon ve Yerel arama için seçim grafiği



(c) C1M1L2

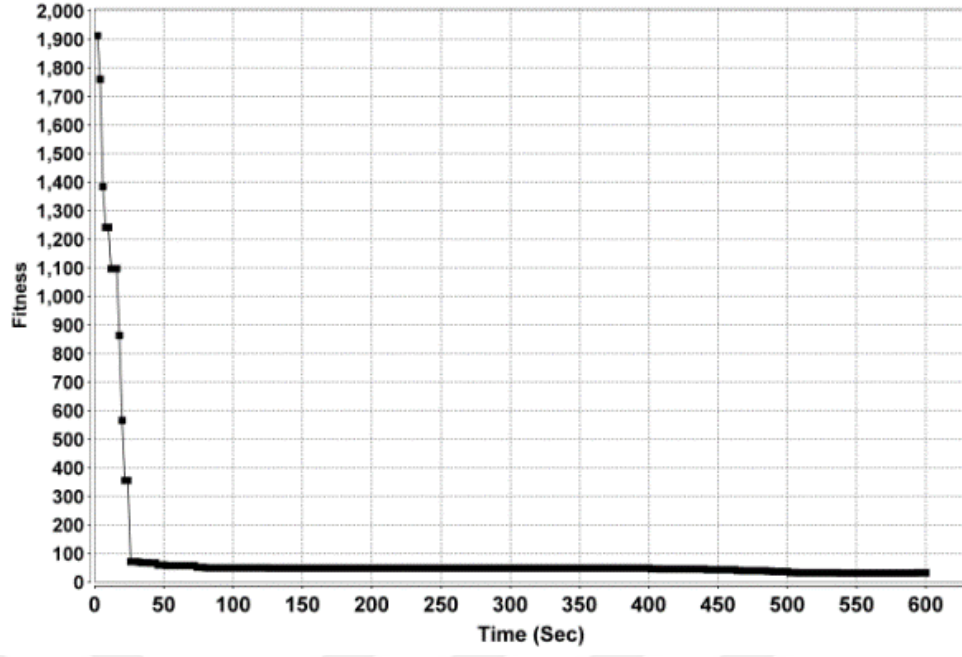


(d) C1M2L2

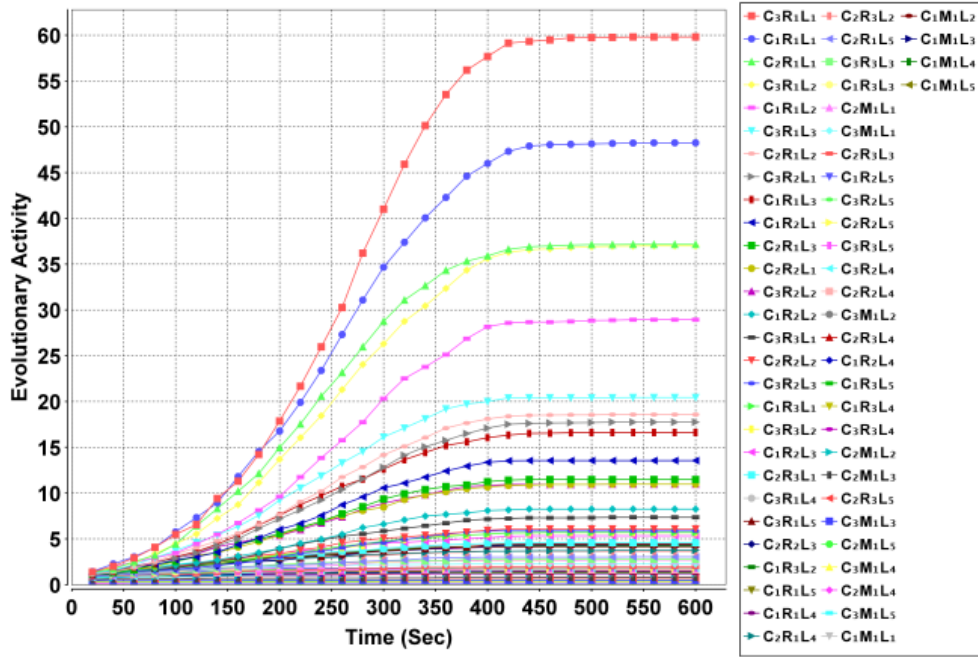


(e) C1M3L2

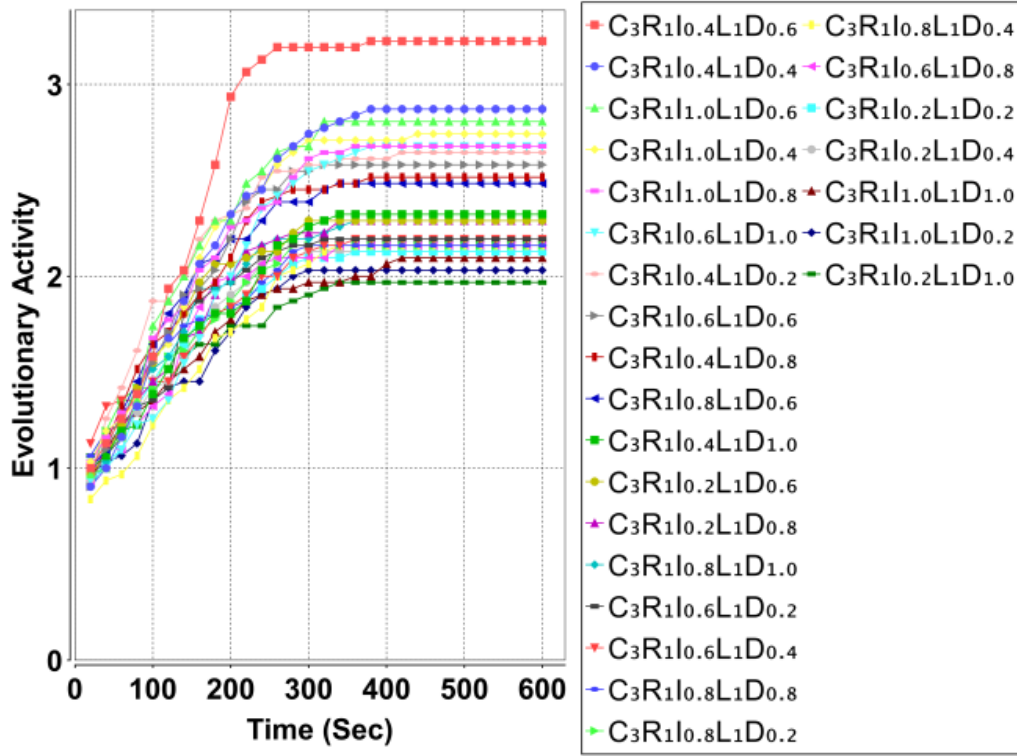
Şekil 11.10. SSMMA algoritması ile rasgele seçilen Bin Packing probleminin 31 deney sonucuna göre evrimsel aktivitesi



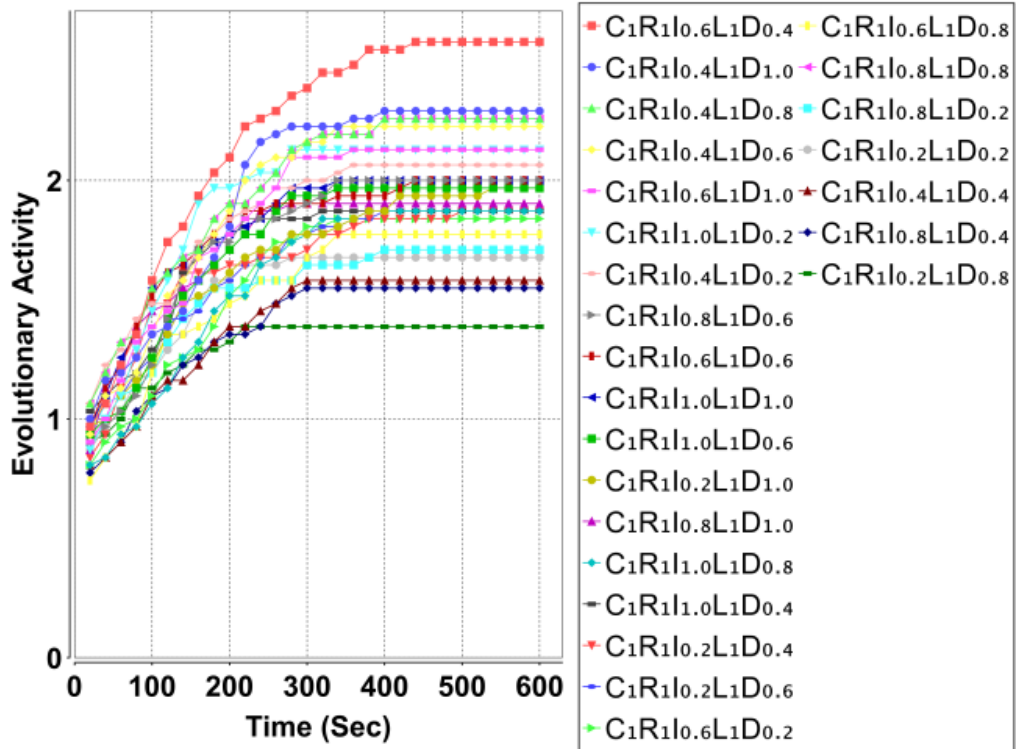
(a) Örnek 3 için ağırlık değişim grafiği



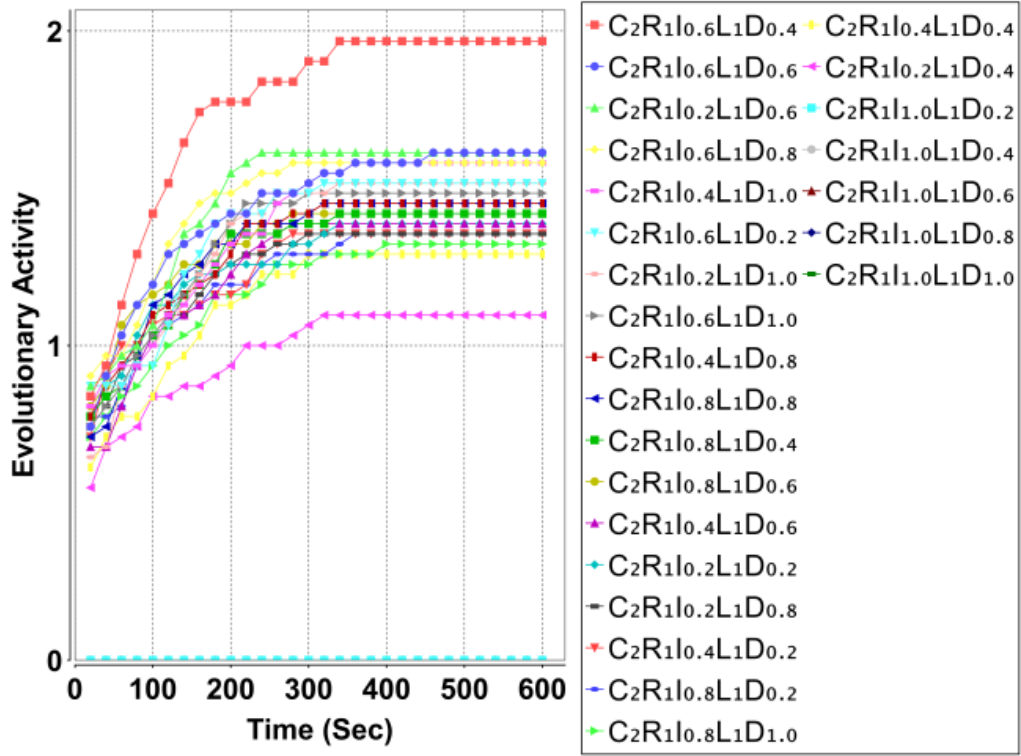
(b) Çaprazlama, Mutasyon ve Yerel arama için seçim grafiği



(c) C3R1L1

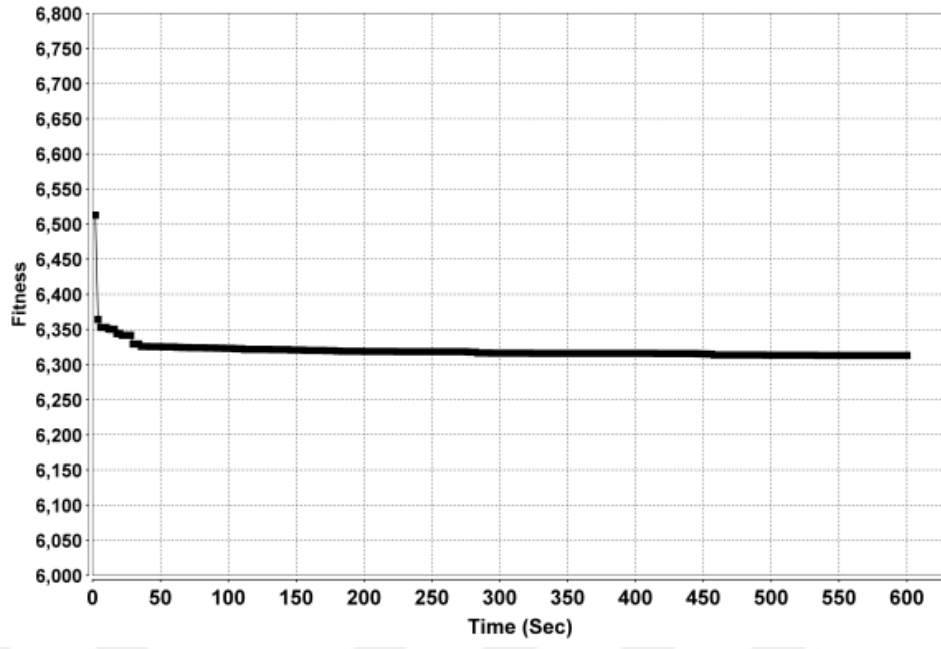


(d) C1L1R1

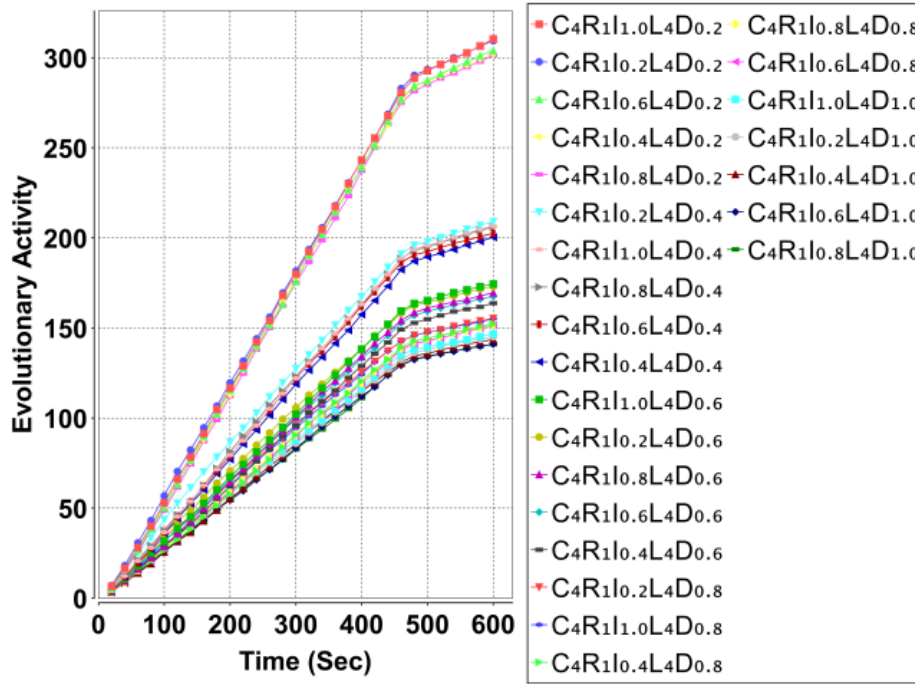


(e) C2R1L1

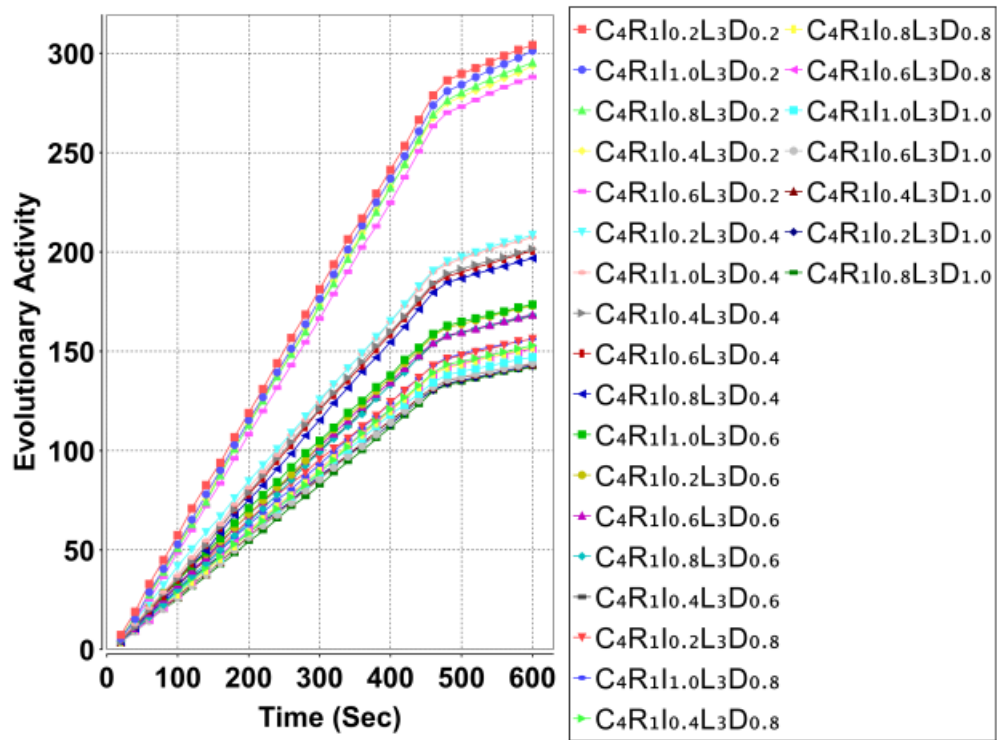
Şekil 11.11. SSMMA algoritması ile rasgele seçilen Personel çizelgeleme probleminin 31 deney sonucuna göre evrimsel aktivitesi



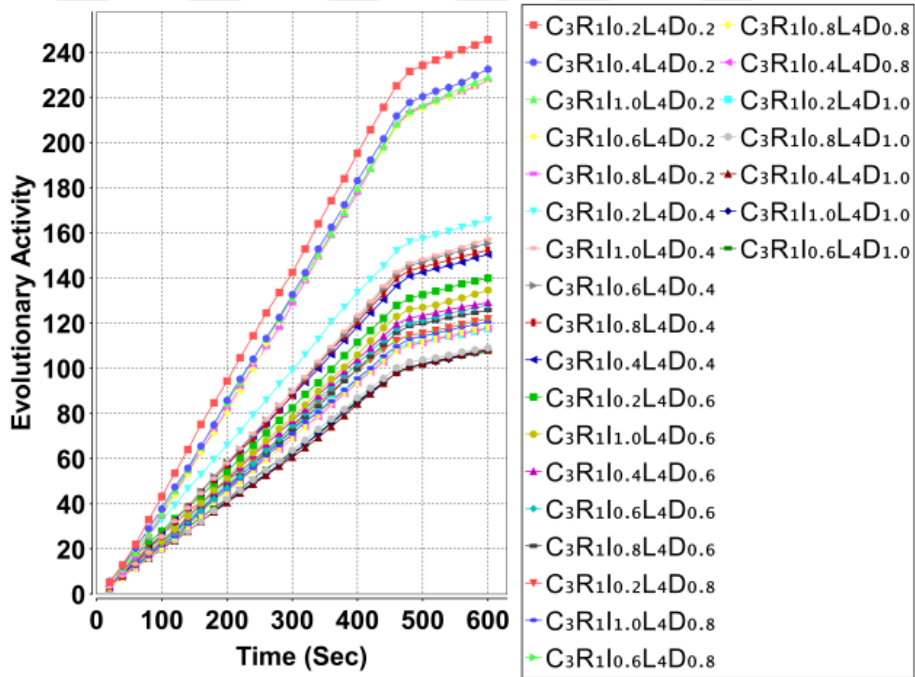
(a) Örnek 3 için ağırlık değişim grafiği



(b) Çaprazlama, Mutasyon ve Yerel arama için seçim grafiği

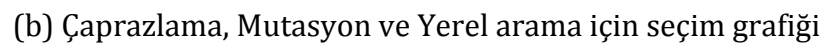


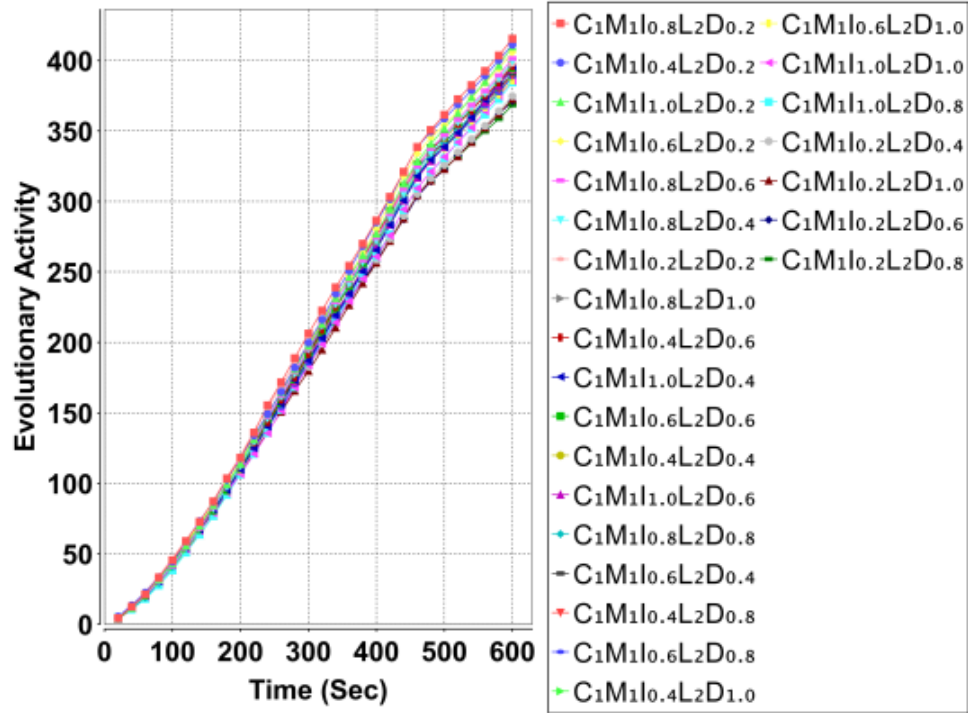
(c) C4R1L4



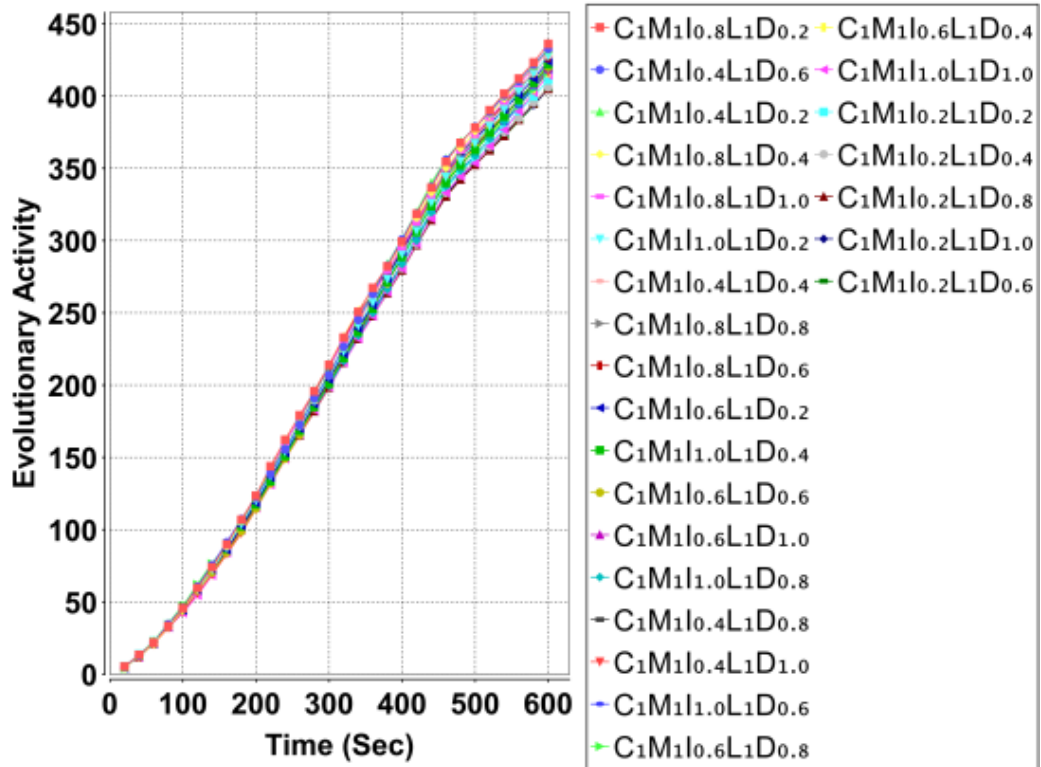
(d) C4L1R3



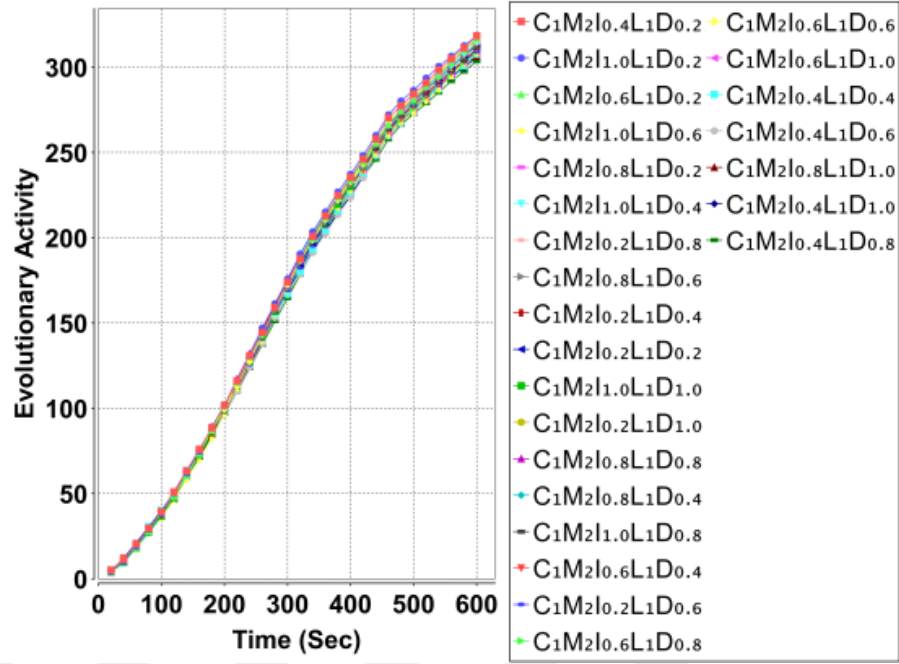




(c) C1M1L1

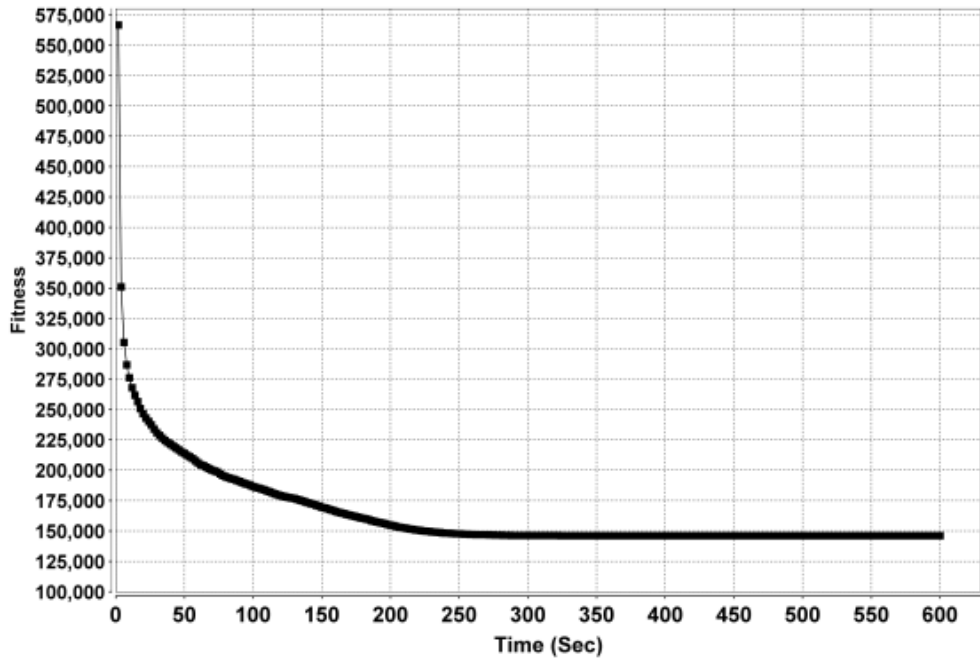


(d) C1M1L2

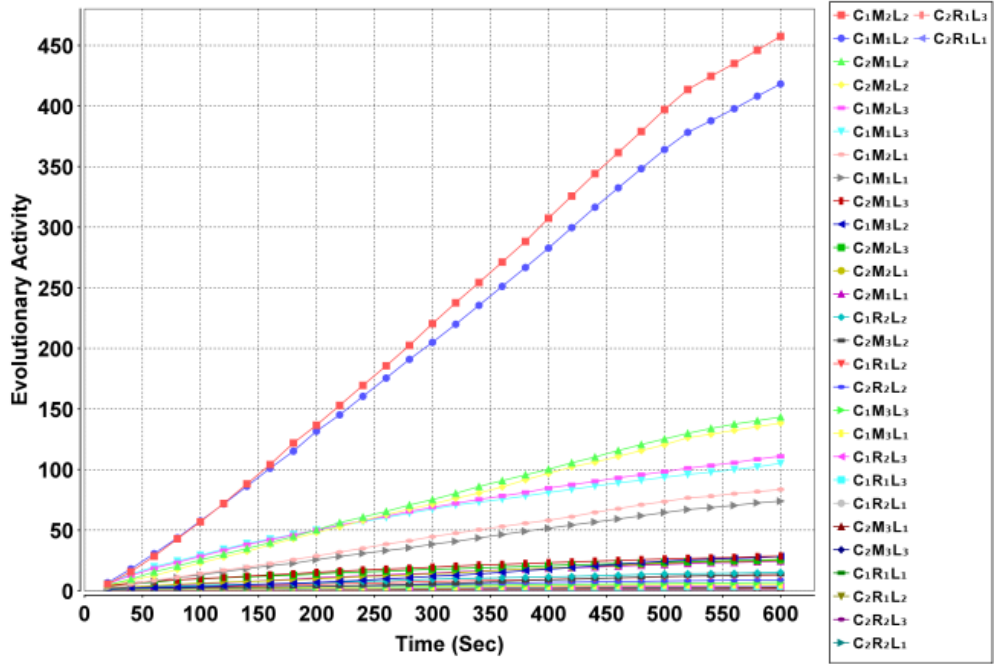


(e) C1M2L1

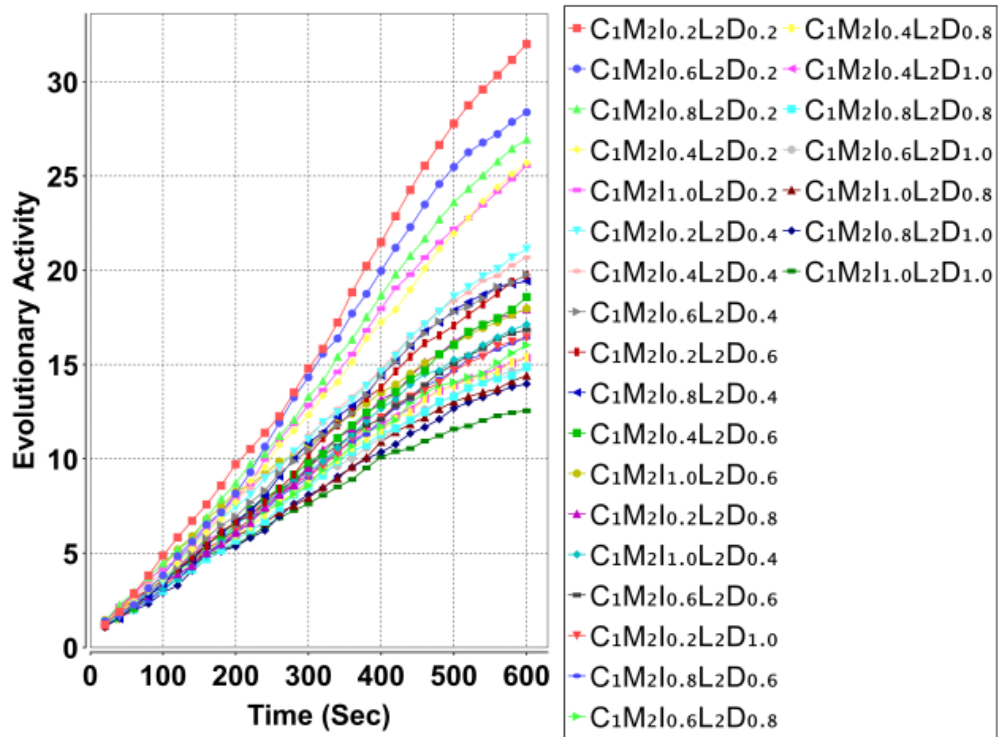
Şekil 11.13. SSMMA algoritması ile rasgele seçilen Gezgin-Satıcı probleminin 31 deney sonucuna göre evrimsel aktivitesi



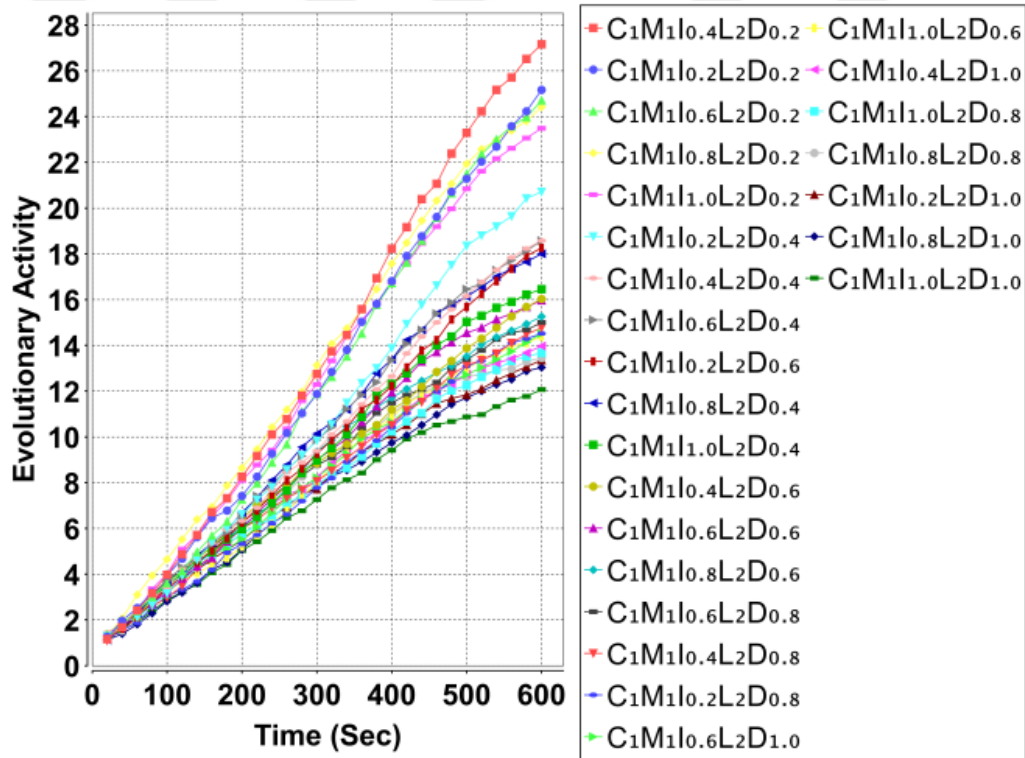
(a) Örnek 3 için ağırlık değişim grafiği



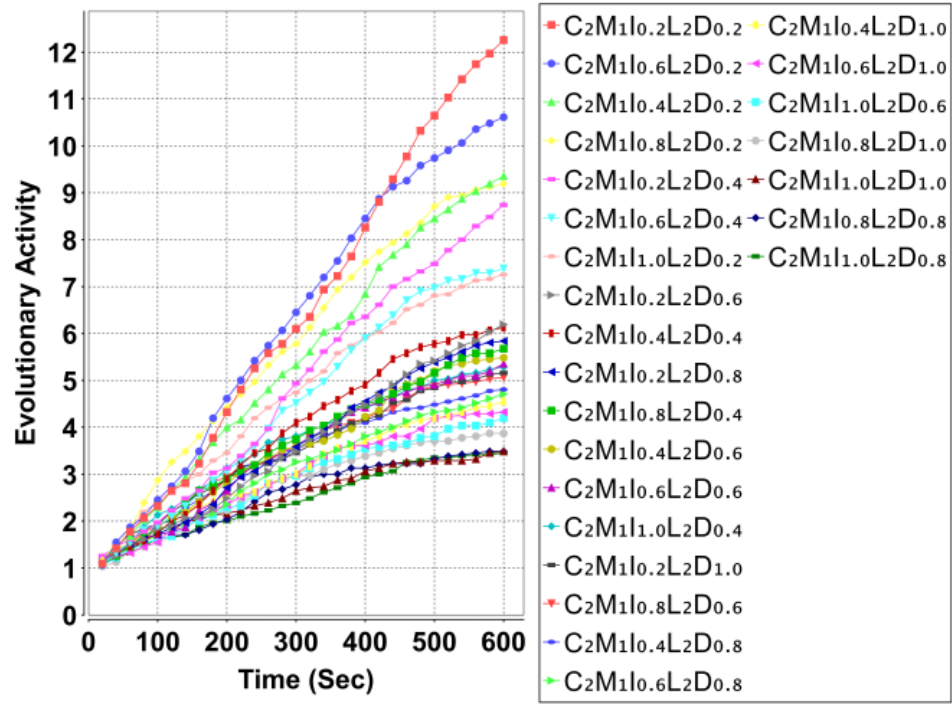
(b)Çaprazlama, Mutasyon ve Yerel arama için seçim grafiği



(c) C1M2L2



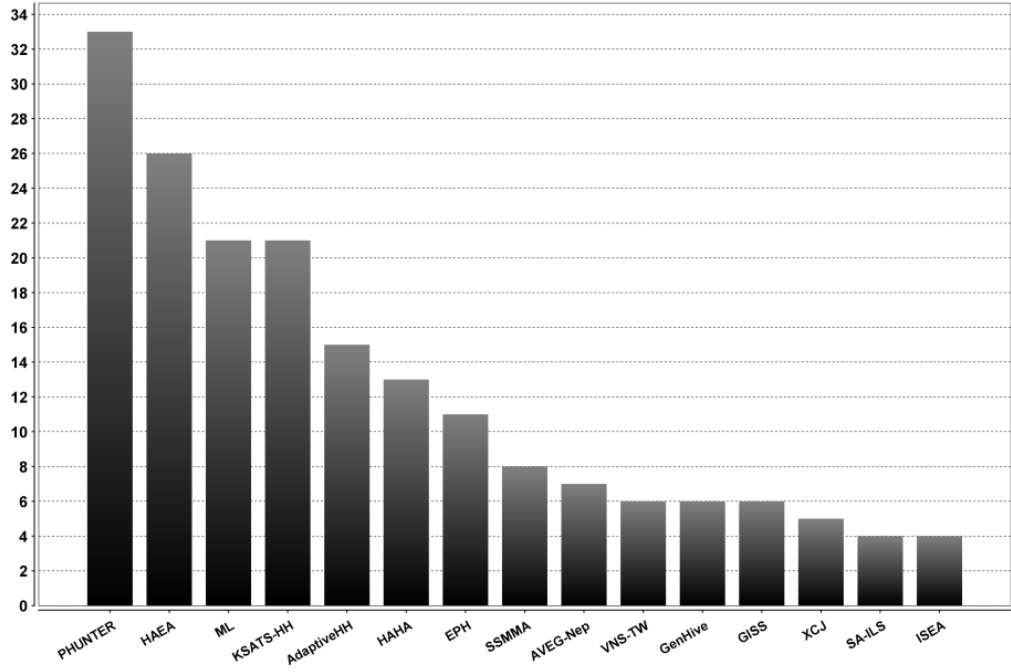
(d) C1M1L2



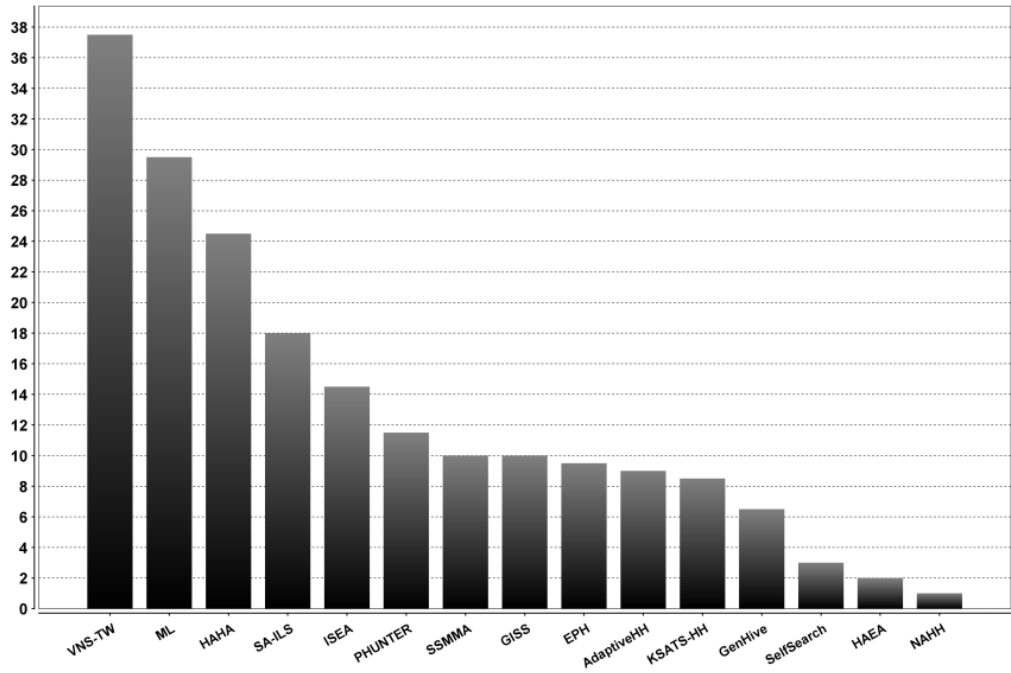
(e) C2M1L2

Şekil 11.14. SSMMA algoritması ile rasgele seçilen Araç Rotalama probleminin 31 deney sonucuna göre evrimsel aktivitesi

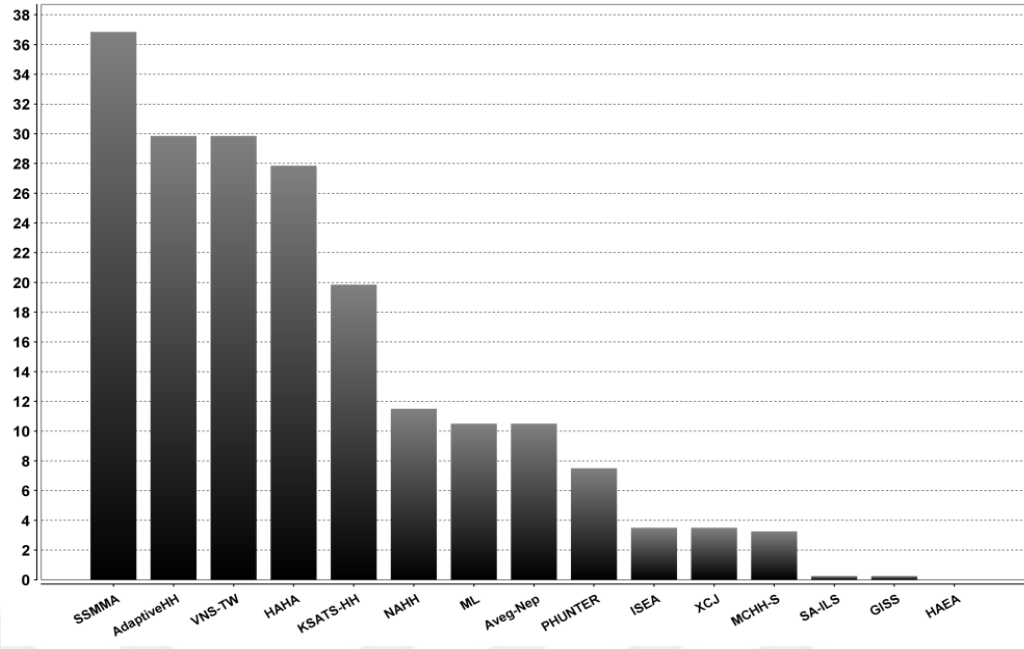
HyFlex frameworkü üzerinde daha önce yapılan çalışmalarla SSMMA algoritması aşağıdaki grafiklerde karşılaştırılarak gösterilmiştir.



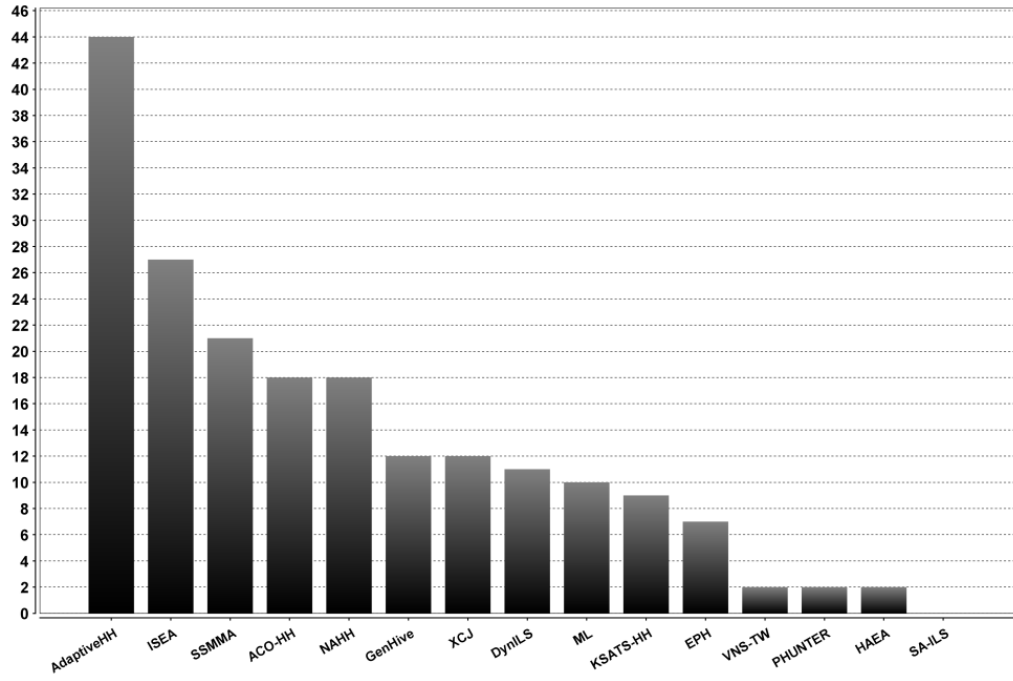
(a) VRP Skor



(b) Personel Çizelgeleme Skor



(c)Max-SAT Skor



(d)Bin Packing Skor

Şekil 11.15. CHeSC2011 algoritmaları ile SSMMA algoritmasının karşılaştırması

Bu araştırma süresince aynı zamanda Sınav Çizelgeleme problemi için memetik algoritma entegre edilen bir yazılım geliştirilmiştir. Yazılım test verisi için ITC2007 örnekleri kullanılmıştır. Bu yazılımda kullanılan düşük seviyeli sezgiseller:

- Seçilen iki çözüm üzerinde tek noktalı çaprazlama
- Seçilen bir periyod üzerine mutasyon
- Seçilen bir derslik üzerine mutasyon
- Rasgele seçilen bir periyod değişimi
- Rasgele seçilen bir derslik değişimi
- Rasgele seçilen iki sınav değişimi
- Periyod için yerel arama
- Derslik için yerel arama

Algoritmanın çalıştırılması sonucu elde edilen veriler Çizelge 11.8 ve 11.9 de gösterilmiştir.

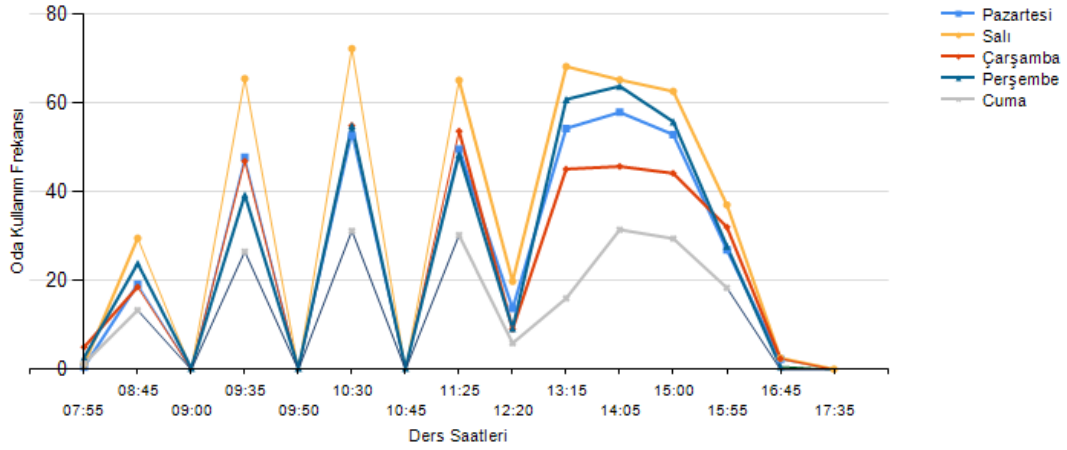
Çizelge 11.8. ITC2007 verileri ile SSMMA sonuçlarının karşılaştırması

|      | <b>Sınav 1</b> |       | <b>Sınav 2</b> |       | <b>Sınav 3</b> |       |
|------|----------------|-------|----------------|-------|----------------|-------|
| Sıra | Yaklaşım       | Skor  | Yaklaşım       | Skor  | Yaklaşım       | Skor  |
| 1    | Muller         | 4370  | Gogos          | 385   | Gogos          | 8996  |
| 2    | McCollumn      | 4633  | Muller         | 400   | McColumn       | 9064  |
| 3    | Gogos          | 4775  | McCollumn      | 405   | Muller         | 10049 |
| 4    | SSMMA          | 5626  | Demeester      | 515   | Gogos          | 13771 |
| 5    | Gogos          | 5905  | SSMMA          | 616   | Pillay         | 15917 |
| 6    | Demeester      | 6060  | De Smet        | 623   | Atsuta         | 17669 |
| 7    | De Smet        | 6670  | Gogos          | 1008  | Rahman         | 19098 |
| 8    | Atsuta         | 8006  | Pillay         | 28886 | SSMMA          | 19617 |
| 9    | Rahman         | 11060 | Rahman         | 3133  | Demeester      | 23580 |
| 10   | Pillay         | 12035 | Atsuta         | 3470  | De Smet        | x     |
|      | <b>Sınav 4</b> |       | <b>Sınav 5</b> |       | <b>Sınav 6</b> |       |
| Sıra | Yaklaşım       | Skor  | Yaklaşım       | Skor  | Yaklaşım       | Skor  |
| 1    | Muller         | 4370  | Gogos          | 385   | Gogos          | 8996  |
| 2    | McCollumn      | 4633  | Muller         | 400   | McColumn       | 9064  |
| 3    | Gogos          | 4775  | McCollumn      | 405   | Muller         | 10049 |
| 4    | SSMMA          | 5626  | Demeester      | 515   | Gogos          | 13771 |
| 5    | Gogos          | 5905  | SSMMA          | 616   | Pillay         | 15917 |
| 6    | Demeester      | 6060  | De Smet        | 623   | Atsuta         | 17669 |
| 7    | De Smet        | 6670  | Gogos          | 1008  | Rahman         | 19098 |
| 8    | Atsuta         | 8006  | Pillay         | 28886 | SSMMA          | 19617 |
| 9    | Rahman         | 11060 | Rahman         | 3133  | Demeester      | 23580 |
| 10   | Pillay         | 12035 | Atsuta         | 3470  | De Smet        | x     |

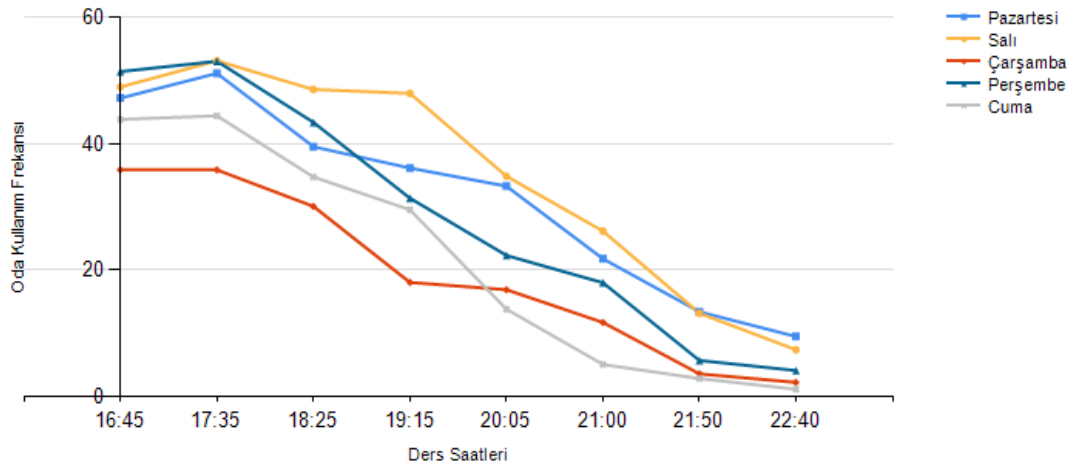
Çizelge 11.9. SDU verilerinin SSMMA ile çözüm sonuçları

|      | <b>SDU01</b> | <b>SDU02</b> | <b>SDU03</b> | <b>SDU04</b> | <b>SDU05</b> |
|------|--------------|--------------|--------------|--------------|--------------|
| Skor | 760          | 5880         | 210          | 20000        | 30           |

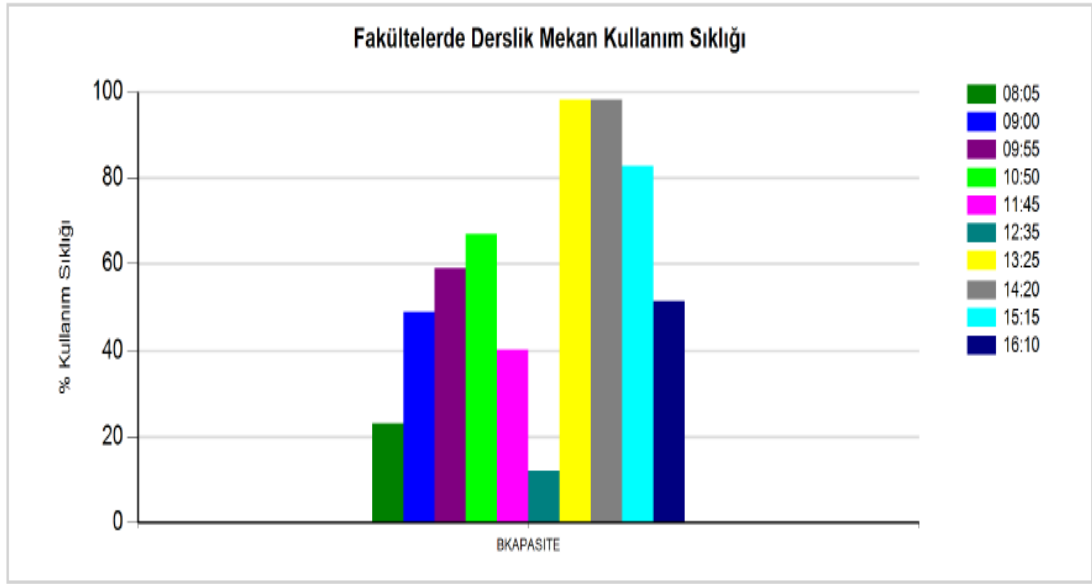
Yürütülen üst-sezgisel çalışmasında genetik algoritma, yerel arama algoritması ve tabu arama algoritmasından oluşan hibrid sezgiseller kullanılmıştır. Üniversitemide elle yapılan ders çizelgeleme problemlerinde dersliklerin verimli kullanılmadığı tespit edilmiştir. Şekil 11.16, 11.17, 11.18 'de Mühendislik Fakültesinin 2016-2017 güz dönemine ait haftalık derslik kullanım oranları gösterilmiştir. Bu grafiğe göre derslikler en fazla %70 kapasiteyle kullanılmaktadır. Elle yapılan ders çizelgelerinde çizelgelerin en iyi çizelge olmadığı veya dersliklerin tam verimlilikte kullanılmadığı gözlemlenmiştir. Bu sorunu çözmek için geliştirilecek yazılım hem mümkün olduğunca tüm kısıtları çözecek hem de dersliklerin kullanım oranlarını artırmayı amaçlamaktadır.



Şekil 11.16. Fakülte Saatlere Göre Sıralanmış Haftalık Derslik Alanlarının Kullanım Çizelgesi(1.Öğretim)



Şekil 11.17. Fakülte Saatlere Göre Sıralanmış Haftalık Derslik Alanlarının Kullanım Çizelgesi(2.Öğretim)



Şekil 11.18. Fakülte Saatlere Göre Derslik Kullanım Analizi

## 12. SONUÇ VE ÖNERİLER

Üst-sezgisel çalışması sonucunda SSMA, TGMA, MMA ve SSMMA algoritmaları Hyflex yazılımına başarı ile entegre edilmiştir. SSMA ve TGMA, algoritmaları basit yapıya sahiptirler ve herhangi bir adaptif öğrenme mekanizması içermemektedirler. MMA ve SSMMA adaptif ve self-adaptif memetik algoritmaların performansları Hyflex yazılımı üzerinde test edilmiştir. Sınav çizelgeleme problem çözümlerinin daha iyi performans sağlaması amacı ile de Memetik algoritmaların değişik versiyonları yazılım üzerine entegre edilmiş olup ve değişik üst-sezgisel mekanizmaları test edilecektir.

Üst-sezgisel çalışması sonucunda algoritmaları Öğrenci Bilgi Sistemi yazılımına başarı ile entegre edilmiştir. Bu çalışmanın sonuçları hem bilimsel hem de teknolojik açılarından, yurt ve dünya ölçeklerinde özgün değerler üretip var olan bilgi birikimine ve teknolojik yeteneklere önemli katkılarda bulunacaktır. Üniversite çizelgeleme problemleri konusunda yapılan literatür taraması, dersliklerin analizi konusunda önemli bir teorik ve özellikle deneysel eksiklik olduğunu göstermektedir. Mevcut uygulama kısmen de olsa bu boşluğu doldurmayı amaçlamaktadır. Uygulama gerçekleştiği takdirde, ülkemizdeki üniversitelerin çizelgeleme problemleri daha sorunsuz çözümlere ulaşacakları beklenmektedir. Üniversite çizelgeleme, genellikle yabancı kökenli üniversiteler tarafından çalışılmaktadır. Uygulamanın bir diğer amacı da, bu üniversitelerde yapılan çalışmaların Türkiye’de yapmak ve ülke ekonomisine katkı sağlamaktır. 2020-2021 Eğitim-Öğretim yılı güz-bahar döneminde kullanıcılar için daha etkin ve verimli bir çizelgeleme aracı geliştirilmeye devam edecektir. Yeni bölümlerin ve mekanların sisteme entegre edilmesi sağlanıp değişik algoritmalar ile problem çözülmeye çalışılacaktır. Ayrıca daha önce elle yapılan sınav atama problemi üst-sezgisel algoritmalar kullanılarak otomatik yapılması planlanmaktadır. Fakültelerdeki bina kullanım analizlerinin yapılması hedeflenmektedir. Ders çizelgeleme problemi için geliştirilen algoritmada hem SDÜ veri setleri hem de ITC2007 veri setleri üzerinde çalışmalar yapılmıştır. Çizelge 12.1’de SDÜ için elde edilen çözüm sonuçları bulunmaktadır. Çizelge 12.2’de ise ITC2007 çözümleri verilmiştir.

Çizelge 12.1. Üniversite Veri Seti

| Veri Seti | Algoritma | Sonuç | Derslik Kullanım Oranı |
|-----------|-----------|-------|------------------------|
| SDU 1     | SGMA      | 162   | 62.66                  |
| SDU 2     | SGMA      | 81    | 57.47                  |
| SDU 3     | SGMA      | 255   | 35.66                  |
| SDU 4     | SGMA      | 163   | 77.92                  |
| SDU 5     | SGMA      | 21    | 60.14                  |
| SDU 6     | SGMA      | 146   | 53.31                  |
| SDU 7     | SGMA      | 47    | 47.76                  |
| SDU 8     | SGMA      | 73    | 45.42                  |
| SDU 9     | SGMA      | 177   | 45.45                  |
| SDU 10    | SGMA      | 129   | 27.22                  |

Çizelge 12.2. ITC2007 Veri Seti

| ITC Veri Seti | Yarışma Birincisi Sonucu | SGMA Sonucu |
|---------------|--------------------------|-------------|
| comp01        | 5                        | 5           |
| comp02        | 51                       | 44          |
| comp03        | 84                       | 70          |
| comp04        | 37                       | 35          |
| comp05        | 330                      | 298         |
| comp06        | 48                       | 42          |
| comp07        | 20                       | 6           |
| comp08        | 41                       | 37          |
| comp09        | 109                      | 103         |
| comp10        | 16                       | 16          |
| comp11        | 0                        | 11          |
| comp12        | 333                      | 328         |
| comp13        | 66                       | 65          |
| comp14        | 59                       | 53          |
| comp15        | 84                       | 69          |
| comp16        | 34                       | 52          |
| comp17        | 83                       | 80          |
| comp18        | 83                       | 71          |
| comp19        | 62                       | 57          |
| comp20        | 27                       | 34          |
| comp21        | 103                      | 102         |

## KAYNAKLAR

- Abdul-Rahman, S., Burke, E. K., Bargiela, A., McCollum, B., & Özcan, E., 2014. A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of Operations Research*, 218(1), 3-21.
- Ahmadi S, Barrone P, Cheng P, Burke EK, Cowling P, McCollum, 2003. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In: *Multidisciplinary International Scheduling: Theory and Applications (MISTA 2003)*, 155-171.
- Akar, E., Topcuoglu, H. R., Ermis, M., 2014. Hyper-heuristics for online UAV path planning under imperfect information. In: *European Conference on the Applications of Evolutionary Computation*. Springer, pp. 741–752.
- Allen S, Burke EK, Hyde MR, Kendall G ,2009. Evolving reusable 3d packing heuristics with genetic programming. In: *Genetic and Evolutionary Computation Conference (GECCO '09)*, ACM, 931-938.
- Al-Yakoob, S., Sherali, H., & Al-Jazzaf, M. ,2010. A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7, 19–46.
- Asmuni H., Burke E. K., and Garibaldi J. M.,2005. Fuzzy Multiple Ordering Criteria for Examination Timetabling, in Burke E.K., Trick M. (eds.), *selected Papers from the 5th International Conference on the Theory and Practice of Automated Timetabling (PATAT 2004)- The Theory and Practice of Automated Timetabling V*, *Lecture Notes in Computer Science*, 3616, 147–160.
- Asmuni, H., Burke, E. K., Garibaldi, J. M., & McCollum, B., 2006. A novel fuzzy approach to evaluate the quality of examination timetabling. In *Proceedings of the 6th international conference on the practice and theory of automated timetabling (PATAT 2006)* (82–102). Brno, Czech Republic, August 30th– September 1st 2006.
- Asmuni H., Burke E.K., Garibaldi J. M., and McCollum B., 2007. Determining Rules in Fuzzy Multiple Heuristic Orderings for Constructing Examination Timetables, *Proceedings of the 3rd Multidisciplinary International Scheduling: Theory and Applications Conference (MISTA 2007)*, 59-66.
- Asmuni H., Burke E.K., Garibaldi J. M., McCollum B., and Parkes A. J., 2009. An Investigation of Fuzzy Multiple Heuristic Orderings in the Construction of University Examination Timetables, *Computers and Operations Research*, Elsevier, 36(4), 981-1001.
- Atsuta, M., Nonobe, K., & Ibaraki, T., 2008. ITC-2007 track2: an approach using general CSP solver. Available from the ITC-2007 website. [http://www.cs.qub.ac.uk/itc2007/winner/bestcoursolutions/Atsuta et al.pdf](http://www.cs.qub.ac.uk/itc2007/winner/bestcoursolutions/Atsuta_et_al.pdf).

- Azimi, Z. N., 2005. Hybrid heuristics for examination timetabling problem. *Applied Mathematics and Computation*, 163(2), 705–733.
- Bader-El-Den MB., Poli R., 2007. Generating sat local-search heuristics using a gp hyper-heuristic framework. In: *International Conference on Artificial Evolution*, Lecture Notes in Computer Science, 4926, 37-49.
- Bader-El-Den MB., Poli R., 2008. An incremental approach for improving local search heuristics. In: *European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP'08)*, Lecture Notes in Computer Science, 4972, 194-205.
- Bader-El-Den M., Poli R., Fatima S., 2009. Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. *Memetic Computing* 1(3):205-219.
- Bai R., Kendall G., 2005. An investigation of automated planograms using a simulated annealing based hyper-heuristics. In: Ibaraki T, Nonobe K, Yagiura M (eds) *Metaheuristics: Progress as Real Problem Solver - (Operations Research/Computer Science Interface Series, Vol.32)*, Springer, 87-108.
- Bai R., Burke EK., Kendall G., McCollum B., 2007. Memory length in hyper-heuristics: An empirical study. In: *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CISched2007)*, Hilton Hawaiian Village, Honolulu, Hawaii, USA.
- Bai R., Kendall G., 2008. A model for fresh produce shelf-space allocation and inventory management with freshness-condition-dependent demand. *INFORMS Journal on Computing* 20(1): 78-85.
- Bai R., Blazewicz J., Burke EK., Kendall G., McCollum B., 2012. A simulated annealing hyper-heuristic methodology for exible decision support. *4OR: A Quarterly Journal of Operations Research* 10(1): 43-66.
- Battiti R. and Brunato M., 2010. Reactive search optimization: learning while optimizing, *Handbook of Metaheuristics*, 543-571.
- Bilgin B., Özcan E., Korkmaz EE., 2006. An experimental study on hyper-heuristics and final exam scheduling. In: *Proc. of the International Conference on the Practice and Theory of Automated Timetabling (PATAT'06)*, 123-140.
- Borning, A., Duisberg, R., Freeman-Benson, B., Kramer, A., & Woolf, M. ,1987. Constraint hierarchies. In N. Meyrowitz (Ed.), *Proceedings of the conference on object-oriented programming systems, languages, and applications (OOPSLA)* (Vol. 22,. 48–60). New York: ACM.
- Borning, A., Freeman-Benson, B., & Wilson, M. ,1992. Constraint hierarchies. *LISP and Symbolic Computation*, 5, 223–270.

- Branke, J., Nguyen S., Pickardt C., and Zhang M., 2016. Automated design of production scheduling heuristics: A review. *IEEE Transactions on Evolutionary Computation*, 20(1):110–124.
- Burke, E. K., Newall, J., & Weare, R. F., 1996. A memetic algorithm for university exam timetabling. In E. K. Burke & P. Ross (Eds.), *Lecture notes in computer science: Vol. 1153. The practice and theory of automated timetabling* (241–250). Berlin: Springer.
- Burke, E. K., Petrovic, S., & Qu, R., 2002. Case-based heuristic selection for examination timetabling. In *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning (SEAL 2002)* (277–281). Singapore, Nov 18–22 2002.
- Burke E.K., Hart E., Kendall G., Newall J., Ross P., Schulenburg S. ,2003a. Hyperheuristics: An emerging direction in modern search technology. In: Glover F., Kochenberger G. (Eds.) *Handbook of Metaheuristics*, Kluwer, 457-474.
- Burke E.K., Kendall G., Soubeiga E., 2003b. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics* 9(6):451-470
- Burke, E. K., Bykov, Y., Newall, J., & Petrovic, S., 2004. A time-predefined local search approach to exam timetabling problems. *IIE Transactions*, 36, 509–528.
- Burke E.K., Dror M., Petrovic S., Qu R., 2005a. *The NextWave in Computing, Optimization, and Decision Technologies*, Springer, chap Hybrid Graph Heuristics within a Hyper-heuristic Approach to Exam Timetabling Problems, 79-91
- Burke E.K., Kendall G., Landa-Silva JD., O'Brien R., Soubeiga E., 2005b. An ant algorithm hyperheuristic for the project presentation scheduling problem. In: *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, Edinburgh, Scotland, vol 3, 2263-2270
- Burke E.K., Landa-Silva JD., Soubeiga E., 2005c. Meta-heuristics: Progress as Real Problem Solvers, Springer, chap Multi-objective Hyper-heuristic Approaches for Space Allocation and Timetabling, 129-158.
- Burke, E. K., Petrovic, S., & Qu, R., 2006a. Case based heuristic selection for timetabling problems. *Journal of Scheduling*, 9(2), 115–132.
- Burke EK, Hyde MR, Kendall G., 2006b. Evolving bin packing heuristics with genetic programming. In: *Parallel Problem Solving from Nature (PPSN'06)*, Lecture Notes in Computer Science, vol 4193, 860-869
- Burke E.K., McCollum B., Meisels A., Petrovic S., and Qu R, 2007a. A Graph-Based Hyper- Heuristic for Educational Timetabling Problems, *European Journal of Operational Research (EJOR)*, 176, 177 – 192.

- Burke E.K., Hyde M., Kendall G., Woodward J., 2007b. The scalability of evolved on line bin packing heuristics. In: IEEE Congress on Evolutionary Computation (CEC'07), Singapore, pp 2530-2537
- Burke E.K., Hyde M., Kendall G., Woodward J., 2007c. Automatic heuristic generation with genetic programming: Evolving a jack-of-all-trades or a master of one. In: Genetic and Evolutionary Computation Conference (GECCO'07), , ACM, pp 1559-1565
- Burke E.K., McCollum B., McMullan P., & Parkes, A. J. ,2008a. Multi-objective aspects of the examination timetabling competition track. In Proceedings of PATAT 2008. Montreal, Canada, August 2008.
- Burke, E. K., & Bykov, Y., 2008b. A late acceptance strategy in hill-climbing for exam timetabling problems. In PATAT 2008 Conference, Montreal, Canada (pp. 1-7).
- Burke E.K., Hyde M., Kendall G., Ochoa G., Özcan E., Woodward J., 2009. Exploring hyper-heuristic methodologies with genetic programming. In: Mumford C., Jain L. (Eds.) Computational Intelligence: Collaboration, Fusion and Emergence, Intelligent Systems Reference Library, Springer, 177-201.
- Burke, E. K., Eckersley, A. J., McCollum, B., Petrovic, S., & Qu, R., 2010a. Hybrid variable neighbourhood approaches to university exam timetabling. European Journal of Operational Research, 206, 46–53.
- Burke E.K., Curtois T, Hyde M, Kendall G, Ochoa G, Petrovic S, Vazquez-Rodriguez JA, Gendreau M., 2010b. Iterated local search vs. hyper-heuristics: Towards generalpurpose search algorithms. In: IEEE Congress on Evolutionary Computation (CEC 2010), 3073-3080
- Burke E.K., Hyde M, Kendall G., 2010c. Providing a memory mechanism to enhance the evolutionary design of heuristics. In: Proceedings of the IEEE World Congress on Computational Intelligence (WCCI'10), Barcelona, Spain, 3883-3890.
- Burke E.K., Hyde M., Kendall G., Ochoa G., Özcan E., Woodward J., 2010d. Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol 146, Springer, chap A Classification of Hyper-heuristic Approaches, 449-468. Chapter 15
- Burke E.K., Hyde M., Kendall G., Woodward J., 2010e. A genetic programming hyperheuristic approach for evolving two dimensional strip packing heuristics. IEEE Transactions on Evolutionary Computation 14(6):942-958.
- Burke E.K., Hyde MR, Kendall G, Woodward J., 2011. Automating the packing heuristic design process with genetic programming. Evolutionary Computation 20(1):63-89.

- Burke E. K., Hyde M. R., and Kendall G., 2012a. Grammatical evolution of local search heuristics. *IEEE Trans. Evolutionary Computation*, 16(3):406–417.
- Burke E. K., Kendall G., Mısır M., and Özcan E., 2012b. Monte carlo hyper-heuristics for examination timetabling. *Annals of Operations Research*, 196(1):73–90.
- Burke, E. K., Gendreau, M., Hyde, M. et al, 2013. Hyper-heuristics: a survey of the state of the art. *J Oper Res Soc* 64, 1695–1724.
- Burke, E. K., Qu, R., Soghier, A., 2014. Adaptive selection of heuristics for improving exam timetables. *Annals of Operations Research* 218 (1), 129–145.
- Burke E.K., Hyde M.R., Kendall G., Ochoa G., Özcan E., Woodward J.R. ,2019. A Classification of Hyper-Heuristic Approaches: Revisited. In: Gendreau M., Potvin JY. (eds) *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, vol 272. Springer, Cham.
- Callennec, B. L., & Boulic, R., 2004. Interactive motion deformation with prioritized constraints. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on computer animation, SCA'04* (pp. 163–171). Aire-la-Ville, Switzerland, Switzerland. Aire-la-Ville: Eurographics Association.
- Cano-Belman J, Rios-Mercado R, Bautista J., 2010. A scatter search based hyperheuristic for sequencing a mixed-model assembly line. *Journal of Heuristics* 16:749-770.
- Carter, M. W., 1986. OR practice—a survey of practical applications of examination timetabling algorithms. *Operations research*, 34(2), 193-202.
- Carter M. W., Laporte G., Lee S.Y.,1996. Examination Timetabling: Algorithmic Strategies and Applications, *The Journal of the Operational Research Society*, 47(3), 373-383.
- Carter, M., & Laporte, G., 1998. Recent developments in practical course timetabling. E. Burke, M. Carter, eds. *Practice and Theory of Automated Timetabling*. Springer Lecture Notes in Computer Science Series, 1408, 3-19.
- CHeSC 2011. cross-domain heuristic search challenge. <http://www.asap.cs.nott.ac.uk/external/chesc2011/>. Erişim Tarihi: 29.05.2020
- CHeSC 2014. the second cross-domain heuristic search challenge. <http://www.hyflex.org/chesc2014/>. Erişim Tarihi: 29.05.2020

- Chakhlevitch K., Cowling P., 2008. Hyperheuristics: Recent developments. In: Adaptive and Multilevel Metaheuristics, Studies in Computational Intelligence, 136, Springer, 3-29.
- Chan C, Xue F, Ip W, Cheung C., 2012. A hyper-heuristic inspired by pearl hunting. In: International Conference on Learning and Intelligent Optimization (LION 6), Lecture Notes in Computer Science, vol 7219, pp 349-353.
- Chen PC, Kendall G, Vanden-Berghe G., 2007. An ant based hyper-heuristic for the travelling tournament problem. In: Proceedings of IEEE Symposium of Computational Intelligence in Scheduling (CISched 2007), Hawaii, pp 19-26.
- Chen, Y., Mourdjis, P., Polack, F., Cowling, P., Remde, S., 2016. Evaluating hyperheuristics and local search operators for periodic routing problems. In: European Conference on Evolutionary Computation in Combinatorial Optimization. Vol. 9595 of Lecture Notes in Computer Science. Springer, pp. 104–120.
- Cheng P, Barone R, Ahmadi S, Cowling P, 2003. Integrating human abilities with the power of automated scheduling systems: Representational epistemological interface design. In: AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments, pp 23-29.
- Cowling P., Kendall G., Soubeiga E., 2000. A hyperheuristic approach for scheduling a sales summit. In: Selected Papers of the Third International Conference on the Practice And Theory of Automated Timetabling, PATAT 2000, Springer, Konstanz, Germany, Lecture Notes in Computer Science, 176-190.
- Cowling P., Kendall G., Soubeiga E., 2001. A parameter-free hyperheuristic for scheduling a sales summit. In: Proceedings of the 4th Metaheuristic International Conference, 127-131.
- Cowling P., Kendall G., Han L., 2002a. An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL'02), Orchid Country Club, Singapore, 267-271.
- Cowling P, Kendall G, Soubeiga E., 2002b. Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation. In: Cagoni S, Gottlieb J, Hart E, Middendorf M, Goenther R (eds) Applications of Evolutionary Computing: Proceeding of Evo Workshops 2002, Springer-Verlag, Kinsale, Ireland, Lecture Notes in Computer Science, vol 2279, pp 1-10.
- Crowston W. B., Glover F., Thompson G. L., Trawick J.D. 1963. Probabilistic and parametric learning combinations of local job shop scheduling rules. ONR Research memorandum, GSIA, Carnegie Mellon University, Pittsburgh 1(117).

- da Fonseca, G. H. G., Santos, H. G., Toffolo, T. A. M., Brito, S. S., Souza, M. J. F., Apr 2016. Goal solver: a hybrid local search based solver for high school timetabling. *Annals of Operations Research* 239 (1), 77–97
- de Werra, D., Asratian, A. S., & Durand, S., 2002. Complexity of some special types of timetabling problems. *Journal of Scheduling*, 5(2), 171-183.
- Demeester P, Bilgin B, Causmaecker PD, Berghe GV., 2012. A hyperheuristic approach to examination timetabling problems: Benchmarks and a new problem from practice. *Journal of Scheduling* 15(1):83-103.
- Denzinger J., Fuchs M., Fuchs M., 1996. High performance ATP systems by combining several ai methods. Tech. Rep. SEKI-Report SR-96-09, University of Kaiserslautern.
- Di Gaspero, L., 2002. Recolour, shake and kick: a recipe for the examination timetabling problem. In E. K. Burke & P. D. Causmaecker (Eds.), *Proceedings of the 4th international conference on the practice and theory of automated timetabling*. KaHo St.-Lieven, Gent, Belgium (pp. 404–407).
- Dimopoulos C, Zalzal AMS, 2001. Investigating the use of genetic programming for a classic one-machine scheduling problem. *Advances in Engineering Software* 32(6):489-498.
- Dorndorf U, Pesch E, 1995. Evolution based learning in a job shop scheduling environment. *Computers and Operations Research* 22(1):25-40.
- Dowsland, K. A., & Thompson, J., 2005. Ant colony optimization for the examination scheduling problem. *The Journal of the Operational Research Society*, 56(4), 426–438.
- Dowsland KA, Soubeiga E, Burke E.K., 2007. A simulated annealing hyperheuristic for determining shipper sizes. *European Journal of Operational Research* 179(3):759-774.
- Downie, M. L., 2005. Efficiency Outcomes from Space Charging in UK Higher Education Estates. *Property Management*, 23 (1), pp. 33-42.
- Drake J, Özcan E, Burke E.K, 2012, An improved choice function heuristic selection for cross domain heuristic search. In: *Parallel Problem Solving from Nature – PPSN XII, Lecture Notes in Computer Science*, vol 7492, 307-316.
- Drechsler R, Becker B, 1995. Learning heuristics by genetic algorithms. In: *ASP Design Automation Conference*, 349-352.
- Drechsler R, Göckel N, Becker B, 1996. Learning heuristics for obdd minimization by evolutionary algorithms. In: *Parallel Problem Solving*

from Nature (PPSN'96), Lecture Notes in Computer Science, (1141), 730-739.

Dubois, D., Fargier, H., & Prade, H., 1996. Possibility theory in constraint satisfaction problems: handling priority, preference and uncertainty. *Applied Intelligence*, 6, 287–309.

Dueck, G. (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational physics*, 104(1), 86-92.

Eiben A. E., Hinterding R., and Michalewicz Z., 1999. Parameter control in evolutionary algorithms, *Evolutionary Computation*, IEEE Transactions on, (3), 124-141.

Eiben A.E., Michalewicz Z., Schoenauer M., Smith J.E., 2007. Parameter Setting in Evolutionary Algorithms, Springer, chap Parameter Control in Evolutionary Algorithms, 19-46.

Fang H., Ross P., Corne D., 1993. A promising genetic algorithm approach to job shop scheduling, rescheduling, and open-shop scheduling problems. In: *International Conference on Genetic Algorithms*, Morgan Kaufmann, 375-382.

Fang H., Ross P., Corne D., 1994. A promising hybrid ga/heuristic approach for openshop scheduling problems. In: *European Conference on Artificial Intelligence*, John Wiley & Sons.

Fisher H., Thompson G.L., 1961. Probabilistic learning combinations of local job-shop scheduling rules. In: *Factory Scheduling Conference*, Carnegie Institute of Technology

Fisher H., Thompson G.L. 1963. Probabilistic learning combinations of local job-shop scheduling rules. In: *Industrial Scheduling*, Prentice-Hall, Inc, New Jersey, 225-251.

Freisleben B., Härtfelder M., 1993. Optimization of genetic algorithms by genetic algorithms. In: Albrecht R.F., Steele N.C., Reeves C.R. (eds) *Artificial Neural Nets and Genetic Algorithms*, Springer Verlag, 392-399.

Fukunaga A.S., 2002. Automated discovery of composite sat variable-selection heuristics. In: *National conference on Artificial intelligence*, Edmonton, Alberta, Canada, 641-648.

Fukunaga A.S., 2004. Evolving local search heuristics for SAT using genetic programming. In: *Genetic and Evolutionary Computation Conference (GECCO '04)*, ISGEC, Springer, Lecture Notes in Computer Science, vol 3103, 483-494.

- Fukunaga AS., 2008. Automated discovery of local search heuristics for satisfiability testing. *Evolutionary Computation* (MIT Press) 16(1):31-1.
- Garcia-Villoria A, Salhi S, Corominas A, Pastor R, 2011. Hyper-heuristic approaches for the response time variability problem. *European Journal of Operational Research* 211(1):160-169.
- Garrido P, Riff MC., 2007a. Collaboration between hyperheuristics to solve strip-packing problems. In: 12th International Fuzzy Systems Association World Congress, Proceedings, Springer, Lecture Notes in Computer Science, vol 4529, 698-707.
- Garrido P, Riff MC., 2007b. An evolutionary hyperheuristic to solve strip-packing problems. In: Intelligent Data Engineering and Automated Learning - IDEAL 2007 Proceedings, Springer, Lecture Notes in Computer Science, vol 4881, 406-415.
- Garrido P, Castro C., 2009. Stable solving of cvrps using hyperheuristics. In: Genetic and Evolutionary Computation Conference (GECCO'09), ACM, Montreal, Canada, 255-262.
- Garrido P, Riff M., 2010. Dvrp: A hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *Journal of Heuristics* 16:795-834.
- Gaspero LD, Urli T., 2012. Evaluation of a family of reinforcement learning crossdomain optimization heuristics. In: International Conference on Learning and Intelligent Optimization (LION 6), Springer, Lecture Notes in Computer Science, vol 7219, 384-389.
- Geiger CD, Uzsoy R, Aytug H., 2006. Rapid modeling and discovery of priority dispatching rules: An autonomous learning approach. *Journal of Scheduling* 9(1):7-34.
- Gendreau M. and Potvin J. Y., 2010. *Handbook of Metaheuristics*: Springer Verlag.
- Gibbs J, Kendall G, Özcan E., 2010. Scheduling english football fixtures over the holiday period using hyper-heuristics. In: Parallel problem solving from nature, PPSN'10, Springer-Verlag, 496-505.
- Glover F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13:533-549.
- Glover, F., 1989. Tabu search—part I. *ORSA Journal on computing*, 1(3), 190-206.
- Gogos, C., Alefragis, P., & Housos, E., 2010. An improved multi-staged algorithmic process for the solution of the examination timetabling problem. *Annals of Operations Research*. doi:10.1007/s10479-010-0712-3

- Goldberg, D. E., 1989. Genetic algorithms in search. Optimization and Machine Learning, Reading, Massachusetts.
- Gratch J., Chien S., DeJong G., 1993. Learning search control knowledge for deep space network scheduling. In: International Conference on Machine Learning, Amherst, MA, 135-142.
- Gratch J., Chien S. 1996. Adaptive problem-solving for large-scale scheduling problems: a case study. Journal of Artificial Intelligence Research 4:365-396.
- Grefenstette J., 1986. Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics SMC-16(1):122-128.
- Grobler, J., Engelbrecht, A. P., Kendall, G., & Yadavalli, V. S. S., 2015. Heuristic space diversity control for improved meta-hyper-heuristic performance. Information Sciences, 300, 49-62.
- Hart E., Ross P., 1998a. A heuristic combination method for solving job-shop scheduling problems. In: Parallel Problem Solving from Nature, PPSN V, Springer, Lecture Notes in Computer Science, (1498), 845-854.
- Hart E., Ross P., Nelson JAD., 1998b. Solving a real-world problem using an evolving heuristically driven schedule builder. Evolutionary Computing 6(1):61-80.
- Hart E. and Sim K., 2016. A hyper-heuristic ensemble method for static job-shop scheduling. Evolutionary Computation, 24(4):609-635.
- Henz, M., Yap, R. H. C., Lim, Y. F., Lua, S. C., Walser, J. P., & Shi, X. P., 2004. Solving hierarchical constraints over finite domains with local search. Annals of Mathematics and Artificial Intelligence, 40, 283-301.
- Ho NB, Tay JC., 2005. Evolving dispatching rules for solving the exible job-shop problem. In: IEEE Congress on Evolutionary Computation (CEC'05), Edinburgh, UK, pp 2848-2855.
- Hoos H. H. and Stützle T., 2005. Stochastic local search: Foundations and applications: Morgan Kaufmann.
- HyFlex competition instance summary.  
<http://www.asap.cs.nott.ac.uk/external/chesc2011/reports/CHeSCInstanceSummary.pdf>. Erişim Tarihi: 29.05.2020.
- HyFlex API, 2014. hyper-heuristics flexible framework API.  
<http://www.hyflex.org/>. Erişim Tarihi: 29.05.2020.
- Hamadi Y., Monfroy E., and Saubion F., 2011. What is Autonomous Search?, Hybrid Optimization, 357-391.

- Jakobovic D, Jelenkovic L, Budin L., 2007. Genetic programming heuristics for multiple machine scheduling. In: European Conference on Genetic Programming (EUROGP' 07), Lecture Notes in Computer Science, vol 4445, 321-330.
- Kaplansky, E., & Meisels, A., 2004. Negotiation among scheduling agents for distributed timetabling. In Submitted to the 5th international conference on practice and theory of automated timetabling (PATAT'04), Pittsburgh.
- Keller RE, Poli R., 2007a. Cost-benefit investigation of a genetic-programming hyperheuristic. In: International Conference on Artificial Evolution, 13-24.
- Keller RE, Poli R., 2007b. Linear genetic programming of parsimonious metaheuristics. In: IEEE Congress on Evolutionary Computation (CEC'07), 4508-4515.
- Keller RE, Poli R., 2008a. Self-adaptive hyperheuristic and greedy search. In: IEEE World Congress on Computational Intelligence (WCCI'08), 3801-3808.
- Keller RE, Poli R., 2008b. Subheuristic search and scalability in a hyperheuristic. In: Genetic and Evolutionary Computation Conference(GECCO'08), pp 609-610.
- Keller RE, Poli R., 2008c. Toward subheuristic search. In: IEEE World Congress on Computational Intelligence (WCCI'08), pp 3148-3155.
- Kheiri, A., Özcan, E., Parkes, A. J., 2016. A stochastic local search algorithm with adaptive acceptance for high-school timetabling. *Annals of Operations Research* 239 (1), 135–151.
- Kheiri, A., Keedwell, E., 2017. A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems. *Evolutionary Computation* 25 (3), 473–501
- Kibria R. H. and Li Y., 2006. Optimizing the initialization of dynamic decision heuristics in DPLL SAT solvers using genetic programming. In *Proceedings of the 9th European Conference on Genetic Programming*, volume 3905 of *Lecture Notes in Computer Science*, pages 331–340, Budapest, Hungary. Springer, Berlin.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P., 1983. Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Krasnogor N., Gustafson S., 2004. A study on the use of “self-generation” in memetic algorithms. *Natural Computing* 3(1):53-76.
- Kumar R, Joshi AH, Banka KK, Rockett PI., 2008 Evolution of hyperheuristics for the biobjective 0/1 knapsack problem by multiobjective genetic programming. In: Genetic and Evolutionary Computation Conference(GECCO'08), ACM, 1227-1234.

- Landa Silva, J. D., 2003. Metaheuristic and multiobjective approaches for space allocation (Doctoral dissertation, University of Nottingham).
- Lewis, R., Paechter, B., & McCollum, B., 2007. Post enrolment based course timetabling: a description of the problem model used for track two of the second international timetabling competition (Cardiff Working Papers in Accounting and Finance A2007-3), Cardiff Business School, Cardiff University, August 2007.
- Li J, Burke EK, Qu R., 2011. Integrating neural network and logistic regression to underpin hyper-heuristic search. *Knowledge-Based Systems* 24(2):322-330
- Lokketangen A, Olsson R., 2010. Generating meta-heuristic optimization code using adate. *Journal of Heuristics* 16(6):911-930
- Lopez-Camacho E., Terashima-Marin H., Ross P., Valenzuela-Rendon M., 2010. Problem-state representations in a hyper-heuristic approach for the 2d irregular bpp. In: Ibero-American Conference on AI, IBERAMIA 2010, Springer, Lecture Notes in Computer Science, vol 6433, 204-213
- Lopez-Camacho E., Terashima-Marin H., Ross P., and Ochoa G., 2014. A unified hyper-heuristic framework for solving bin packing problems. *Expert Systems with Applications*, 41(15):6876–6889.
- Marin-Blazquez JG, Schulenburg S., 2007. A hyper-heuristic framework with XCS: Learning to create novel problem-solving algorithms constructed from simpler algorithmic ingredients. In: IWLCS, Springer, Lecture Notes in Computer Science, vol 4399, 193-218
- Marshall, R. J., Johnston, M., Zhang, M., 2015. Hyper-heuristic operator selection and acceptance criteria. In: European Conference on Evolutionary Computation in Combinatorial Optimization (EvoCOP). Vol. 9026 of Lecture Notes in Computer Science. Springer, pp. 99–113.
- Martello, S., and Toth, P. 1990a. Bin-packing problem. In *Knapsack Problems: Algorithms and Computer Implementations*. Wiley. chapter 8, 221–245.
- Martello, S., and Toth, P. 1990b. Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics* 28:59–70.
- McCollum, B., McMullan, P., Burke, E. K., Parkes, A. J., & Qu, R., 2007. The second international timetabling competition: examination timetabling track (Tech. Rep. QUB/IEEE/Tech/ITC2007/Exam/ v4.0/17.2007) Queen's University Belfast. <http://www.cs.qub.ac.uk/itc2007/>.
- McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., Gaspero, L. D., Qu, R., and Burke, E. K., 2010. Setting the Research Agenda

- in Automated Timetabling: The Second International Timetabling Competition. *INFORMS Journal on Computing* 22(1):120–130.
- McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., & Qu, R. (2012). A new model for automated examination timetabling. *Annals of Operations Research*, 194(1), 291-315.
- Maturana J., Lardeux F., Saubion F., 2010. Autonomous operator management for evolutionary algorithms. *Journal of Heuristics* 16:881-909.
- Meignan D, Koukam A, Creput JC., 2010. Coalition-based metaheuristic: A selfadaptive metaheuristic using reinforcement learning and mimetism. *Journal of Heuristics* 16(6):859-879
- Merlot, L. T. G., Boland, N., Hughes, B. D., & Stuckey, P. J., 2003. A hybrid algorithm for the examination timetabling problem. In *Springer lecture notes in computer science: Vol. 2740. Practice and theory of automated timetabling: selected papers from the 4th international conference* (pp. 207–231).
- Meseguer, P., Bouhmala, N., Bouzoubaa, T., Irgens, M., & Sánchez, M., 2003. Current approaches for solving over-constrained problems. *Constraints*, 8, 9–39. doi:10.1023/A:1021902812784. Metaheuristics Network, International timetable competition 2002.
- Mısır M., Wauters T, Verbeeck K, Berghe GV. 2009. A new learning hyper-heuristic for the traveling tournament problem. In: *Proceedings of the 8th Metaheuristic International Conference (MIC'09)*, Hamburg, Germany
- Mısır M., Vancroonenburg W, Verbeeck K, Berghe GV., 2011. A selection hyperheuristic for scheduling deliveries of ready-mixed concrete. In: *Proceedings of MIC 2011*, 289-298
- Mısır M., Verbeeck K, De Causmaecker P, Vanden Berghe G., 2012a. An intelligent hyper-heuristic framework for chesc 2011. In: *International Conference on Learning and Intelligent Optimization (LION 6)*, Springer, Lecture Notes in Computer Science, vol 7219, 461-466
- Mısır M., 2012b. Intelligent hyper-heuristics: a tool for solving generic optimisation problems. Ph.D. thesis, Department of Computer Science, KU Leuven.
- Minton S., 1996. Automatically configuring constraint satisfaction problems: a case study. *Constraints* 1(1):7-43.
- Miranda, P. L., Cordeau, J. F., Ferreira, D., Jans, R., & Morabito, R., 2018. A decomposition heuristic for a rich production routing problem. *Computers & Operations Research*, 98, 211-230.

- Moscato, P., 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Caltech concurrent computation program, C3P Report, (826).
- Mourdjis, P., Chen, Y., Polack, F., Cowling, P., Robinson, M., 2016. Variable neighbourhood descent with memory: A hybrid metaheuristic for supermarket resupply. In: International Workshop on Hybrid Metaheuristics. Springer, pp. 32–46.
- Müller, T., 2009. ITC2007 solver description: a hybrid approach. *Annals of Operations Research*, 172, 429–446.
- Mumford, C. L., 2007. An order based evolutionary approach to dual objective examination timetabling. In *Proceedings of the 2007 IEEE symposium on computational intelligence in scheduling (CI-Sched 2007)*, Honolulu, Hawaii, 1–5th April.
- New Castle University, 2007. Report of Space Management in Higher Education Newcastle University.
- Norenkov I, Goodman E, 1997. Solving scheduling problems via evolutionary methods for rule sequence optimization. 2nd World Conference on soft Computing, WSC2.
- Nutt, B., 2000. Four Competing Future for Facilities Management. *Facilities*, 18 (3/4), pp. 124-132.
- Ochoa G, Qu R, Burke EK., 2009. Analyzing the landscape of a graph based hyperheuristic for timetabling problems. In: *Genetic and Evolutionary Computation Conference (GECCO 2009)*, ACM, 341-348
- Ochoa G, Hyde M, Curtois T, Vazquez-Rodriguez JA, Walker J, Gendreau M, Kendall G, Parkes AJ, Petrovic S, Burke EK., 2012a. HyFex: a benchmark framework for cross-domain heuristic search. In: *Proceedings of the 12th European conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP'12*, Springer-Verlag, Lecture Notes in Computer Science, vol 7245, 136-147
- Ochoa G, Walker J, Hyde M, Curtois T., 2012b. Adaptive evolutionary algorithms and extensions to the hyex hyper-heuristic framework. In: *Parallel Problem Solving from Nature - PPSN XII*, Springer, Lecture Notes in Computer Science, vol 7492, 418-427
- Oltean M, Grosan C., 2003. Evolving evolutionary algorithms using multi expression programming. In: *European Conference on Artificial Life (ECAL'03)*, Lecture Notes in Arti\_cial Intelligence, vol 2801, 651-658
- Oltean M., 2005. Evolving evolutionary algorithms using linear genetic programming. *Evolutionary Computation* 13(3):387-410

- Ong Y. S. and Keane A. J., 2004. Meta-Lamarckian learning in memetic algorithms, *IEEE Transactions on Evolutionary Computation*, (8), 99-110.
- Ortiz-Bayliss J, Özcan E, Parkes A, Terashima-Marin H., 2010. Mapping the performance of heuristics for constraint satisfaction. In: *IEEE Congress on Evolutionary Computation (CEC 2010)*, pp 1-8
- Özcan E, Bykov Y, Birben M, Burke E., 2009. Examination timetabling using late acceptance hyper-heuristics. In: *Proceedings of IEEE Congress on Evolutionary Computation (CEC 2009)*, 997-1004.
- Özcan E, Mısırlı M, Ochoa G, Burke E., 2010. A reinforcement learning - great-deluge hyper-heuristic for examination timetabling. *International Journal of Applied Metaheuristic Computing (IJAMC)* 1(1):39-59.
- Özcan E, Kheiri A., 2011a. A hyper-heuristic based on random gradient, greedy and dominance. In: *Proceedings of the 26th International Symposium on Computer and Information Sciences ISCIS2011*, Springer
- Özcan E, Parkes AJ., 2011b. Policy matrix evolution for generation of heuristics. In: *Genetic and Evolutionary Computation Conference (GECCO 2011)*, 2011-2018.
- Papadimitriou, C. H. & Steiglitz, K. 1982. *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall.
- Paquete, L., & Stutzle, T., 2002. Empirical analysis of tabu search for the lexicographic optimization of the examination timetabling problem. In (2002). *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling*. 21st-23rd August (pp. 413-420).
- Pillay N., and Banzhaf W., 2007. A Genetic Programming Approach to the Generation of Hyper-Heuristics for the Uncapacitated Examination Timetabling Problem, in Neves et al. (eds.), *Progress in Artificial Intelligence, Lecture Notes in Artificial Intelligence*, 4874, 223-234.
- Pillay N., and Banzhaf W., 2008. A Study of Heuristic Combinations for Hyper-Heuristic Systems for the Uncapacitated Examination Timetabling Problem, to appear in the *European Journal of Operational Research (EJOR)*, doi:10.1016/j.ejor.2008.07.023.
- Pillay N., 2009. Evolving hyper-heuristics for the uncapacitated examination timetabling problem. In: *Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA'09)*, 447-457.
- Pillay, N., 2016. A review of hyper-heuristics for educational timetabling. *Annals of Operations Research*, 239(1), 3-38.

- Ping-Che H, Tsung-Che C, FLi-Chen, 2012. A vns-based hyper-heuristic with adaptive computational budget of local search. In: IEEE Congress on Evolutionary Computation, 1-8.
- Pisinger D, Ropke S., 2007. A general heuristic for vehicle routing problems. *Computers and Operations Research* 34:2403- 2435.
- Poli R, Woodward JR, Burke EK., 2007. A histogram-matching approach to the evolution of bin-packing strategies. In: IEEE Congress on Evolutionary Computation (CEC'07), Singapore, 3500-3507.
- Poli R. and Graff M., 2009. There is a free lunch for hyper-heuristics, genetic programming and computer scientists, in *Genetic Programming*, 195-207.
- Qu R., Burke E.K., 2005. Hybrid Neighbourhood HyperHeuristics for Exam Timetabling Problems, in *Proceedings of the MIC2005: The Sixth Metaheuristics International Conference*, Vienna, Austria.
- Qu, R.; Burke, E.; McCollum, B.; Merlot, L. T. G.; and Lee, S. Y. 2009a. A Survey of Search Methodologies and Automated System Development for Examination Timetabling. *Journal of Scheduling* 12(1):55–89.).
- Qu, R., Burke, E. K., & McCollum, B., 2009b. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2), 392-404.
- Rechenberg I., 1973. *Evolution strategy: Optimization of technical systems by means of biological evolution*. Fromman-Holzboog, Stuttgart.
- Remde S, Cowling P, Dahal K, Colledge N., 2007. Exact/heuristic hybrids using rvns and hyperheuristics for workforce scheduling. In: *Evolutionary Computation in Combinatorial Optimization (EvoCOP 2007)*, Springer-Verlag, vol 4446, 188-197.
- Remde S, Cowling P, Dahal K, Colledge., 2009. Binary exponential back off for tabu tenure in hyperheuristics. In: *Evolutionary Computation in Combinatorial Optimization*, Springer, vol 5482, 109-112.
- Remde S., Cowling P., Dahal K., Colledge N., Selensky E., 2012. An empirical study of hyperheuristics for managing very large sets of low level heuristics. *Journal of the Operational Research Society* 63(3):392-345.
- Rothlauf F., 2011. *Design of Modern Heuristics: Principles and Application*, Natural Computing Series, Springer-Verlag Berlin Heidelberg.
- Ross P, Schulenburg S, Marn-Blazquez JG, Hart E., 2002. Hyper-heuristics: learning to combine simple heuristics in bin-packing problem. In: *Genetic and Evolutionary Computation Conference, GECCO 2002*, Morgan-Kaufman, 942-948.

- Ross P, Marín-Blázquez JG, Schulenburg S, Hart E., 2003a. Learning a procedure that can solve hard bin-packing problems: A new ga-based approach to hyper-heuristics. In: Genetic and Evolutionary Computation Conference, GECCO 2003, Springer, 1295-1306.
- Ross P, Marín-Blázquez JG, Hart E., 2004. Hyper-heuristics applied to class and exam timetabling problems. In: IEEE Congress on Evolutionary Computation, IEEE Press, 1691-1698.
- Ross P, Marín-Blázquez JG., 2005a. Constructive hyper-heuristics in class timetabling. In: IEEE Congress on Evolutionary Computation, IEEE, 1493-1500.
- Ross P., 2005b. Hyper-heuristics. In: Burke E.K., Kendall G. (Eds.) Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques, Springer, chapter 17, 529-556.
- Ross, P., 2014. Hyper-heuristics. Springer US, pp. 611-638.
- Sabar NR, Ayob M, Qu R, Kendall G., 2011. A graph coloring constructive hyperheuristic for examination timetabling problems. *Applied Intelligence* 37(1):1-11
- Sabar, N. R., Zhang, X. J., Song, A., 2015. A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows. In: IEEE Congress on Evolutionary Computation (CEC 2015). IEEE, pp. 830-837.
- Schaerf, A., 1999. A survey of automated timetabling. *Artificial intelligence review*, 13(2), 87-127.
- Sim K., Hart E, Paechter B., 2012. A hyper-heuristic classifier for one dimensional bin packing problems: Improving classification accuracy by attribute evolution. In: Parallel Problem Solving from Nature (PPSN XII), Lecture Notes in Computer Science, 7492., 217-233.
- Sim K., Hart E., and Paechter B., 2015. A lifelong learning hyper-heuristic method for bin packing. *Evolutionary Computation*, 23(1):37-67.
- Sim, K., Hart, E., 2016. A combined generative and selective hyper-heuristic for the vehicle routing problem. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. ACM, pp. 1093-1100.
- Smet, G. D., 2008. ITC 2007—examination track. Available from the ITC-2007 website. <http://www.cs.qub.ac.uk/itc2007/>.
- Smith-Miles K. A., 2008. Cross-disciplinary perspectives on meta-learning for algorithm selection, *ACM Computing Surveys (CSUR)*, (41), 1-25.

- Soria-Alcaraz, J. A., Ochoa, G., Swan, J., Carpio, M., Puga, H., Burke, E. K., 2014. Effective learning hyper-heuristics for the course timetabling problem. *European Journal of Operational Research* 238 (1), 77–86.
- Soria-Alcaraz, J. A., Ochoa, G., Sotelo-Figeroa, M. A., Burke, E. K., 2017. A methodology for determining an effective subset of heuristics in selection hyper-heuristics. *European Journal of Operational Research* 260 (3), 972–983.
- Space Planning Guidelines, 2009. Tertiary Education Facilities Management Association (TEFMA 2009).
- Storer R.H., Wu S.D., Vaccari R., 1992. New search spaces for sequencing problems with application to job shop scheduling. *Management Science* 38(10):1495-1509.
- Storer R.H., Wu S.D., Vaccari R., 1995. Problem and heuristic space search strategies for job shop scheduling. *ORSA Journal of Computing* 7(4):453-467.
- Sörensen K. and Glover F., 2013. *Encyclopedia of Operations Research and Management Science*, chapter Metaheuristics, pages 960–970.
- Tavares J, Machado P, Cardoso A, Pereira FB, Costa E., 2004. On the evolution of evolutionary algorithms. In: *European Conference on Genetic Programming (EUROGP' 04)*, Coimbra, Portugal, *Lecture Notes in Computer Science*, vol 3003, 389-398
- Talbi E. G., 2009. *Metaheuristics From design to implementation*: Wiley Online Library.
- Tay J. C. and Ho N. B., 2008. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 54: 453–473.
- Terashima-Marin H, Ross P, Valenzuela-Rendon M, 1999. Evolution of constraint satisfaction strategies in examination timetabling. In: *Genetic and Evolutionary Computation Conference, GECCO'99*, 635-642
- Terashima-Marin H., Zarate CJF, Ross P., Valenzuela-Rendon M., 2006. A ga-based method to produce generalized hyper-heuristics for the 2d-regular cutting stock problem. In: *Genetic and evolutionary computation, GECCO 2006*, ACM Press, 591-598
- Terashima-Marin H., Zarate CJF, Ross P., Valenzuela-Rendon M., 2007. Comparing two models to generate hyper-heuristics for the 2d-regular bin-packing problem. In: *Genetic and Evolutionary Computation Conference (GECCO 2007)*, ACM, 2182-2189

- Terashima-Marin H, Ortiz-Bayliss JC, Ross P, Valenzuela-Rendon M., 2008. Hyperheuristics for the dynamic variable ordering in constraint satisfaction problems. In: Genetic and Evolutionary Computation Conference (GECCO 2008), ACM, 571-578.
- Terashima-Marin H, Ross P, Farias-Zarate CJ, Lopez-Camacho E, Valenzuela-Rendon M., 2009. Generalized hyper-heuristics for solving regular and irregular bin packing problems. *Annals of Operations Research* 179(1):369-392
- Thompson, J., & Dowsland, K., 1998. A robust simulated annealing based examination timetabling system. *Computers & Operations Research*, 25, 637-648.
- Tsang, E., Mills, P., & Williams, R., 1999. A computer aided constraint programming system. In *The 1st international conference on the practical application of constraint technologies and logic programming (PACLP)* 81-93.
- Tyasnurita, R., Özcan, E., John, R., 2017. Learning heuristic selection using a time delay neural network for open vehicle routing. In: *IEEE Congress on Evolutionary Computation (CEC 2017)*. IEEE, pp. 1474-1481.
- Vazquez-Rodriguez JA, Petrovic S, Salhi A, 2007a. A combined meta-heuristic with hyper-heuristic approach to the scheduling of the hybrid ow shop with sequence dependent setup times and uniform machines. In: *Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2007)*, 506-513
- Vazquez-Rodriguez JA, Petrovic S, Salhi A, 2007b. An investigation of hyper-heuristic search spaces. In: *IEEE Congress on Evolutionary Computation(CEC 2007)*, 3776-3783.
- Vrugt J. A., Robinson B. A., and Hyman J. M., 2009. Self-adaptive multimethod search for global optimization in real-parameter spaces, *IEEE Transactions on Evolutionary Computation*, (13), 243-259.
- Wah B.W., Ieumwananonthachai A., Chu L.C., Aizawa A., 1995. Genetics-based learning of new heuristics: Rational scheduling of experiments and generalization. *IEEE Trans on Knowledge and Data Engineering* 7(5):763-785.
- Wah B.W., Ieumwananonthachai A., 1999. Teacher: A genetics-based system for learning and for generalizing heuristics. In: Yao X (ed) *Evolutionary Computation*, World Scientific Publishing Co. Pte. Ltd., 124-170.
- Walker JD, Ochoa G, Gendreau M, Burke EK, 2012. Vehicle routing and adaptive iterated local search within the hyex hyper-heuristic framework. In: *International Conference on Learning and Intelligent Optimization (LION 6)*, Springer, Lecture Notes in Computer Science, vol 7219, pp 265-276

- Weise T., Zapf M., Chiong R., and Nebro A., 2009. Why is optimization difficult?, Nature- Inspired Algorithms for Optimisation, 1-50.
- White, G. M., & Xie, B. S., 2001. Examination timetables and tabu search with longer-term memory. In E. K. Burke & W. Erben (Eds.), Practice and theory of automated timetabling: selected papers from the 3rd international conference.
- Wolpert D. H. and Macready W. G., 1997. No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation, (1), 67-82.
- Wong, T., Cote, P., & Gely, P., 2002. Final exam timetabling: a practical approach. In IEEE Canadian conference on electrical and computer engineering (CCECE 2002) (Vol. 2), 726-731.
- Yin, P.-Y., Lyu, S.-R., Chuang, Y.-L., 2016. Cooperative coevolutionary approach for integrated vehicle routing and scheduling using cross-dock buffering. Engineering Applications of Artificial Intelligence 52, 40-53.

## ÖZGEÇMİŞ

Adı Soyadı : Cevriye ALTINTAŞ  
Doğum Yeri ve Yılı : Afyonkarahisar, 1978  
Medeni Hali : Bekar  
Yabancı Dili : İngilizce  
E-posta : cevriyealtintas@isparta.edu.tr



## Eğitim Durumu

Lise : Afyon Lisesi, 1995  
Lisans : Selçuk Üniversitesi, Mühendislik-Mimarlık Fakültesi, Bilgisayar Mühendisliği, 2005

## Mesleki Deneyim

|                                         |           |
|-----------------------------------------|-----------|
| Gazi Üniveristesi                       | 1997-2000 |
| Selçuk Üniversitesi Meram Tıp Fakültesi | 2000-2009 |
| SDÜ Uluborlu Selahattin Karasoy MYO     | 2009-2018 |
| ISUBU Uluborlu Selahattin Karasoy MYO   | 2018-...  |

## Yayınlar

Ender Ozcan, Shahriar Asta, and Cevriye Altintas, 2013. Memetic algorithms for cross-domain heuristic search. In Computational Intelligence (UKCI), 2013 13th UK Workshop on, pages 175–182. IEE.

Altintas, C., Asta, S., Ozcan, E., & Yigit, T., 2014. A self-generating memetic algorithm for examination timetabling. In 10th International Conference of the Practice and Theory of Automated Timetabling (York, UK, 2014).

Özcan, E., Drake, J. H., Altıntaş, C., & Asta, S., 2016. A self-adaptive multimeme memetic algorithm co-evolving utility scores to control genetic operators and their parameter settings. Applied Soft Computing, 49, 81-93.

Unsal, O., Yigit, T., & Altintas, C, 2016. Optimization of dynamic school bus routing problem by using metaheuristic and clustering methods. In Proceedings of

the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016) (pp. 555-559).

Altıntaş, C., Yigit, T., & Unsal, O., 2016. Effective classroom management with a self-generating memetic algorithm for course timetabling. In Proceedings of the 11th International Conference on Practice and Theory of Automated Timetabling (PATAT-2016).

Altıntaş, C., Yigit, T., 2019. Towards The Development of a Decision Support System for Space Management in Higher Education. International Conference on Artificial Intelligence and Applied Mathematics in Engineering 2019 E-Proceeding.

