

T.C.
ÇANAKKALE ONSEKİZ MART ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ



BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROJE-2 RAPORU

Hazırlayanlar : Kadir ÇOLAK 160401037
: Kerim ULUSOY 160401026

Öğretim Elemanının Unvanı/Adı Soyadı : Prof . Dr İsmail KADAYIF

**ILP (Integer Linear Programming) Yöntemi İle Otomatik
Ders Programı Oluşturma**

Haziran,2021
Çanakkale

İçindekiler Tablosu

ŞEKİLLER TABLOSU	2
Giriş	3
BİRİNCİ BÖLÜM.....	3
1.DERS PROGRAMI ÇİZELGELEME PROBLEMİ.....	3
İKİNCİ BÖLÜM	4
2.PuLP KÜTÜPHANESİ VE DOĞRUSAL PROGRAMLAMA ÇÖZÜMÜ	4
2.1 PuLP'ın Tasarımı Ve Özellikleri	4
2.2 Çözücüler İle Arayüz Yapmak	4
ÜÇÜNCÜ BÖLÜM	5
3. LİTERATÜRDE YER ALAN ÇALIŞMALAR	5
DÖRDÜNCÜ BÖLÜM	6
4. DERS PROGRAMI OLUŞTURMA UYGULAMASI.....	6
4.1 Kısıtların Belirlenmesi	6
4.2 Uygulamanın Çalışması	7
4.2.1 Uygulama Çalıştırılırken Dikkat Edilmesi Gerekenler	7
4.2.2 Uygulama Görüntüleri	7
KAYNAKÇA.....	11

ŞEKİLLER TABLOSU

Şekil 1 Ders Ekleme	7
Şekil 2 Öğretim Görevlisi Ekleme	8
Şekil 3 Öğrenci Ekleme	8
Şekil 4 Öğretmen-Öğrenci-Ders Atamaları.....	9
Şekil 5 Oluşturulan Ders Programı.....	9
Şekil 6 Öğretmenlerin Tercihine Göre Kısıt Belirlenmesi	10
Şekil 7 Kısıtlara Göre Hazırlanan Ders Programı.....	10

Giriş

Ders programı çizelgeleme problemi üniversitelerde her eğitim öğretim döneminde karşılaşılan bir problemdir. Üniversitede ders programı hazırlanırken derslerin mevcut durumuna göre atanacağı sınıflar, öğretim elemanı ile ilgili kısıtlar, zor derslerin aynı güne atanmaması gibi isteklerden dolayı problem karmaşık hale gelmektedir. Bu karmaşık yapısından dolayı optimal çözümü bulmak oldukça zorlaşmaktadır. Literatürde bu problemi çözmek için yöneylem araştırması tekniklerinden; doğrusal, tamsayılı, 0-1 tamsayılı, hedef programlama gibi yöntemler kullanılmıştır.

Bu çalışmada PuLP kütüphanesi ile elimizdeki kısıtlara göre bir ders programı oluşturan program elde edilmiştir. Çalışmanın planı şu şekildedir. İlk bölümde ders programı çizelgeleme problemi anlatılmıştır. İkinci bölümde bu problem için kullandığımız çözüm yöntemimizden bahsedilmiştir. Üçüncü bölümde literatürde tamsayılı programlama modeli kullanılan ders programı çizelgeleme problemi çalışmaları incelenmiştir. Son bölümde ise yapmış olduğumuz uygulama anlatılmaktadır.

BİRİNCİ BÖLÜM

1.DERS PROGRAMI ÇİZELGELEME PROBLEMİ

Çok boyutlu bir atama problemi olan ders programı çizelgeleme problemi; ders, öğretim üyesi ve derslik gibi sınırlı sayıda kaynağın verimli kullanılması olarak ifade edilir. Üniversitelerde ders programı çizelgeleme problemi her dönem karşılaşılan bir problemdir. Bir üniversitenin herhangi bir bölümü veya fakültesine ait ders programı çizelgeleme probleminin çözülebilmesi için öncelikle; ders, derslik ve öğretim üyeleri gibi kaynakların özelliklerinin doğru tanımlanmış olması gerekmektedir.

Elde edilen bu veriler için kısıtlar oluşturulur. Kısıtlar, eğitim kalitesinin iyileştirilmesi, öğrenci ve öğretim elemanlarının performansının yükseltilmesi, öğretim elemanı isteklerinin karşılanması ve idari ihtiyaçlara cevap verilebilmesi esas alınarak oluşturulmuştur. Tüm kısıtlar, 3 ana grupta toplanabilir; Öğretim elemanı kısıtları, müfredat dönemi kısıtları, ders kısıtları. Herhangi bir kısıt, program çalıştırılmadan önce açılabilir ya da kapanabilir. Açılan her kısıt sistem tarafından sağlanamaya çalışılır.

İKİNCİ BÖLÜM

2.PuLP KÜTÜPHANESİ VE DOĞRUSAL PROGRAMLAMA ÇÖZÜMÜ

PuLP, kullanıcıların matematiksel modellemel-erini tanımlamalarını sağlayan Python betik dili için bir kütüphanedir. Hızlı ilerlemenin, anlaşılır kod ve sözdiziminin ve basit nesne modelinin önemini vurgulayan Python, iyi yapılandırılmış ve donanımlı, yüksek düzey bir programlama dilidir. Optimizasyon problemlerini ve karar değişkenlerini temsil eden Python nesneleri sağlayan ve kısıtlayıcıların orijinal matematiksel ifadeye old-ukça yakın bir şekilde ifade edilmesine olanak veren PuLP, tamamen Python dilinin sözdizimi ve olağan ifadeleri ile çalışmaktadır. Sözdizimi-ni mümkün olduğunca basit ve sezgisel tutmak adına, PuLP doğrusal ve karma-tamsayı modelleri desteklemeye odaklanmıştır.

2.1 PuLP'in Tasarımı Ve Özellikleri

PuLP'in tasarımında ve Python'un kullanılacak dil olarak seçilmesinde pek çok unsur göz önünde bulundurulmuştur. Basit bir modelleme ve deneyleme aracı, ya da daha büyük bir endüstriyel uygulamanın bir parçası olup olmamasına bakılmaksızın, PuLP'in her yerde kullanılabilir olması istenmiştir. Bu durum, PuLP'in fiyat bakımından uygun, kolaylıkla lisanslı, farklı donanım ve yazılım ortamlarına uyumlu olmasını gerektirmiştir. Python kendi başına bu gerekliliklerin çoğunu karşılamaktadır. Açık-kaynak bir lisansının yanı sıra hem geleneksel hem de yabancı olmak üzere pek çok farklı platformda ücretsiz olarak kullanılabilen uygulamaları bulunmaktadır. PuLP, aynı zamanda MIT Lisansı kapsamında ücretsiz ve lisanslı olması da bu güçlü yönlerini geliştirmektedir. Saf Python kodunda yazıldığı için dağıtım ve uygulamayı kısıtlayacak hiçbir bağımlılık yaratmaz.

2.2 Çözücüler İle Arayüz Yapmak

Gerek ticari (örn. CPLEX , Gurobi) gerekse açık kaynak (örn. CBC) olmak üzere pek çok karışık tam sayılı doğrusal programlama (MILP) çözücüler i mevcuttur. PuLP, Python-PuLP ifadelerinin tam sayılara (örn. modelin seyrek matris ve vektör gösterimleri) dönüştürülmesini dâhili olarak hallederek ve ardından bu veriyi çözücü bir ara yüz sınıfına sunarak, çözücüler için modüler bir yaklaşım getirmektedir. Pek çok çözücünün ara yüzü benzer olduğundan veya standart LP veya MPS dosya formatlarına model yazılarak işlenebileceğinden, temel genel çözücü sınıfların yanı sıra güncel popüler çözücüler için özel ara yüzler de PuLP'a dâhil edilmiştir. Bu genel çözücü sınıflar, daha sonra minimal çaba gösterilerek kullanıcılar veya yeni çözücü geliştiriciler tarafından genişletilebilir

ÜÇÜNCÜ BÖLÜM

3. LİTERATÜRDE YER ALAN ÇALIŞMALAR

Çizelgeleme problemleri, verimliliği esas alarak, kaynakların kullanıcılara belli kısıtlarla tahsis edilmesini hedefler. Bir zaman çizelgeleme problemi olan ders programı hazırlamada (çizelgelemede) hedef, her sınıf için açılacak derslerin buluşma saatlerinin istenen kısıtlar sağlanarak belirlenmesidir. Örneğin, herhangi bir üniversitede bir bölümünün ders programının belirlenmesi demek açılan bütün derslerin saatlerinin belirlenmesi demektir. Doğal kısıtlardan biri ise aynı dönem derslerinin kesişmemesidir. Benzer şekilde, bir öğretim elemanın verdiği dersler aynı saate atanamaz. Zaman çizelgeleme problemi, zor bir problemdir. Bu tür problemlerinin çözümü için değişik yaklaşımlar kullanılmıştır.

Schmidt, bildirisinde bu konudaki ilk çalışmalardan bahseder. Zaman çizelgeleme problemleri için geliştirilen modeller genelde birbirinden oldukça farklıdır. Bu sebeple literatürde bulunan sonuçları karşılaştırmak zordur. Son yıllarda, birçok araştırmacı çözüm yaklaşımlarını Tavlama Benzetimi (Simulated Annealing), Tabu Arama (Tabu Search) ve Genetik Algoritmalar (Genetic Algorithms) gibi yapay zeka yöntemlerine dayandırmaktadır. Bu genetik algoritmalar zaman çizelgeleme problemlerinin çözümünde başarılı olmasına rağmen, çok fazla bilgisayar işlemi gerektirmekte ve probleme özgü bir çözüm olmaktadır. Bu sebeple bu çalışmada yeni bir yaklaşım olan ve probleme bağımlı olmayan üst-sezgisel algoritmalar (hyper-heuristics) kullanılmıştır.

Ders programlarını otomatik olarak yapmak üzere geliştirilmiş çok sayıda yazılım bulunmaktadır. Bunların bazıları ders programını internet üzerinden hazırlayan yazılımlardır (Harmanşah vd., 2011). Bazı ders programı hazırlama yazılımlarında genetik algoritma kullanılmıştır (Atanak ve Fatih, 2005; Kohshori ve Abade, 2012; Wang vd., 2010; Jat ve Yang, 2009; Deris vd., 1999). Bazı yazılımlarda da, sezgisel algoritma (Head ve Shaban, 2007; Causmaecker vd., 2009), uzman sistem destekli algoritma (Çayıroğlu ve Dizdar, 2006) ve evrimsel algoritmalar (Beligiannisa vd., 2008; Fernandes vd., 1999) kullanılmıştır. Ayrıca tam sayı programlama algoritması kullanılarak geliştirilen yazılımlar (MirHassani, 2006; Daskalaki ve Birbas, 2005; Bakır ve Aksop, 2008; Daskalaki vd., 2004) ve farklı algoritmalar kullanılarak geliştirilen yazılımlar (Dimopoulou ve Miliotis, 2004; Deris vd., 2000; Lee vd., 2005; Dasgupta ve Khazanchi, 2005) mevcuttur.

DÖRDÜNCÜ BÖLÜM

4. DERS PROGRAMI OLUŞTURMA UYGULAMASI

4.1 Kısıtların Belirlenmesi

Kısıtlar, eğitim kalitesinin iyileştirilmesi, öğrenci ve öğretim elemanlarının performansının yükseltilmesi, öğretim elemanı isteklerinin karşılanması ve idari ihtiyaçlara cevap verilebilmesi esas alınarak oluşturulmuştur. Tüm kısıtlar, 3 ana grupta toplanabilir; Öğretim elemanı kısıtları, müfredat dönemi kısıtları, ders kısıtları. Herhangi bir kısıt, program çalıştırılmadan önce açılabilir ya da kapanabilir. Açılan her kısıt sistem tarafından sağlanamaya çalışılır. Takip eden kısımda, sistemimizin desteklediği kısıtlar detaylı olarak anlatılmıştır.

- Bir öğretim elemanının verdiği derslerin ders saatleri kesişmemelidir. Bu kısıt bütün öğretim elemanları için her zaman açık konumdadır.
- Hiçbir öğrenci grubu veya öğretim üyesi aynı zaman diliminde birden fazla ders veya dersliğe atanamaz.
- Bir öğrenci grubunun sorumlu olduğu herhangi bir derse birden fazla öğretim üyesi atanamaz.
- Program içerisindeki bütün derslerin atamaları eksiksiz olarak tamamlanmalıdır.
- Bir öğretim üyesinin önceden belirlenmiş bir zaman dilimi veya güne atanması gerekebilir.
- Öğretim üyelerinin olabildiğince tercih ettikleri zaman dilimleri ve dersliklere atanması (E)
- Gün içerisindeki; ilk ders ve son ders arasındaki sürenin olabildiğince kısa tutulması (E)
- Görece kavranması zor derslerin sabah saatlerine atanması (E)
- Programın, haftanın en azından bir gününü boşaltacak şekilde hazırlanması (E)

Oluşturulan bu kısıtlar zorunlu ve esnek kısıtlar olarak iki durumda incelenir. Zorunlu kısıtlar; derslerin uygun zaman dilimlerine atanmasını sağlayan çizelgenin oluşturulabilmesi için kesinlikle sağlanması gereken kısıtlardır. Esnek kısıtlar ise zorunlu olmayan, ancak uygulanması halinde çözümün kalitesine katkı sağlayacak nitelikte olan hedefleri ifade etmektedir. Üniversitelerde ders programı çizelgeleme problemlerinde sıklıkla kullanılan zorunlu ve esnek kısıtlar yukarıda gösterilmektedir. Esnek kısıtların kullanım amacı, oluşturulan çizelgelerin kalitesini olabildiğince üst seviyeye çıkarmaktır. Modele ait bu yapılar genellikle amaç fonksiyonu ifadesine öncelik katsayıları ile birlikte dâhil edilerek kullanılmaktadır. Biz uygulamamızda verileri tamamen elle girmeye dayalı yaptığımız için keyfi değerler ile işleme soktuk gerçek bir kullanımda öğretim görevlileri ve öğrenciler ile görüşmeler sonucunda bu problemde öğretim görevlilerin ders günü ve öğrencilerin ders yoğunluğu ile ilgili istekleri dikkate alınarak sağlıklı bir sonuç alınabilir.

4.2 Uygulamanın Çalışması

4.2.1 Uygulama Çalıştırılırken Dikkat Edilmesi Gerekenler

Öncelikle **.Net Core 3.1** veya üzeri ve **Python** yüklü olmalı. Bilgisayarımızda **Python Path** Eklenmiş olmalıdır.

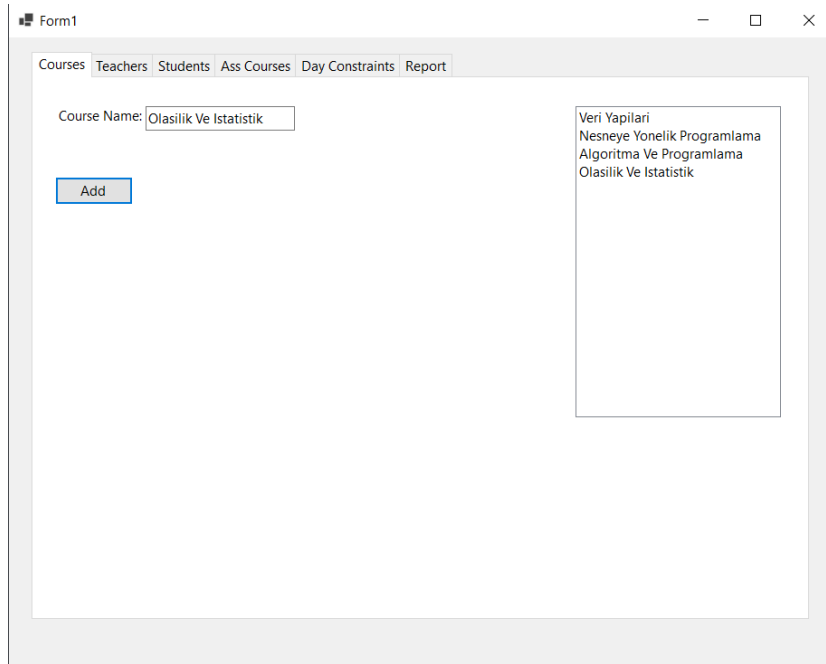
Sonrasında aşağıdaki komutları sırası ile çalıştırıp ilgili paketler kurulmalıdır:

```
python -m pip install pulp
python -m pip install scikit-glpk
python -m pip install progressbar
```

Son olarak uygulamamızı **Visual Studio** ile açıp çalıştırabiliriz.

Solver olarak PULP_CBC_CMD kullanılmıştır. Ancak solve_schedule.py dosyasında 266. satırdaki #print(pl.pulpTestAll()) açarsanız yada diğer yöntemlerle (dosya konumu üzerinden ulaşmak gibi) kendi sisteminizde hangi solver'ı seçebileceğinizi görüp istediğiniz solver'ı set edebilirsiniz.

4.2.2 Uygulama Görüntüleri



Şekil 1 Ders Ekleme

- Uygulamamızı çalıştırdığımızda karşımıza gelen ekran bu şekildedir. İlk kısımda ders programımızdaki derslerimizin sisteme kaydını yapıyoruz.

Form1

Courses Teachers Students Ass Courses Day Constraints Report

Teacher Name: Sait Can Yucebas

Add

Necdet Yucel
Ismail Kadayif
Bahadir Karasulu
Sait Can Yucebas

Şekil 2 Öğretim Görevlisi Ekleme

- İkinci kısımda öğretim görevlilerinin sisteme kaydını yapıyoruz.

Form1

Courses Teachers Students Ass Courses Day Constraints Report

Student: Kerim Ulusoy

Add

Kadir Colak
Kerim Ulusoy

Şekil 3 Öğrenci Ekleme

- 3. kısımda dersleri alacak öğrencilerimizi sisteme kayıt ediyoruz

Şekil 4 Öğretmen-Öğrenci-Ders Atamaları

- 4.kısımda öğretim görevlilerimizi verecekleri dersleri ile öğrencileri seçtikleri dersleri ile eşleştiriyoruz.Bu bölümde öğretmen ve öğrenci kısıtlarımız devreye giriyor.

Timeslots	Misc Courses
10:00am	Sait Can Yucebas: Nesneye Yonelik Programlama
11:00am	Necdet Yucel: Algoritma Ve Programlama
12:00am	Ismail Kadayif: Veri Yapilari
5:00pm	Bahadir Karasulu: Olasilik Ve Istatistik

Şekil 5 Oluşturulan Ders Programı

- 4.kısımda ki eşleştirmeleri yaptıktan sonra Report bölümümüzde program istenilen günü seçip Get dediğimizde bize girdiğimiz veriler eşliğinde bir ders programı oluşturuyor.

Şekil 6 Öğretmenlerin Tercihine Göre Kısıt Belirlenmesi

- Eğer bu ders programına kısıtlar koymak istiyorsak örneğin şekilde görüldüğü gibi Algoritma dersini pazartesi günü sabah 8.00 'a Olasılık dersini 12.00a atamak gibi bu kısıtları burada belirliyoruz.

Timeslots	Misc Courses
8:00am	Necdet Yucel: Algoritma Ve Programlama
12:00am	Bahadır Karasulu: Olasilik Ve Ististik
1:00pm	Sait Can Yucebas: Nesneye Yonelik Programlama
4:00pm	Ismail Kadayif: Veri Yapilari

Şekil 7 Kısıtlara Göre Hazırlanan Ders Programı

- Son olarak Report bölümünden tekrar kısıt belirlediğimiz günlere ait ders programını oluşturmasını söylediğimiz de istediğimiz derslerin istediğimiz saatte olduğunu gördüğümüz kısıtlarımızı sağlayan bir program üretmiş oluyoruz.

KAYNAKÇA

- PuLP Kütüphanesi ve Python ile Doğrusal Programlama Problemlerinin Çözümü Sunumu, Coşkun Atay, Serkan O. Dibek, Öykü Kara, Cemre Ersöz, İzmir Ekonomi Üniversitesi
- Altunay H, Eren T. (2017). “Ders programı çizelgeleme problemi için bir literatür taraması”. Pamukkale University Journal of Engineering Sciences, 23(1), 55-70