

Öncelikle bizim projelerimiz ile Sorumlu Öğretim Görevlileriniz birbirine bağılılar. Bu projelerin atamalarını olabildiğince öğretim görevlilerinin sınıflarını değiştirmeyecek şekilde atamamız gerekiyor. Bunu yaparken "" algoritmamızın doğasına uygun olacak şekilde çözümümüzü uygulayalım. İlk timeslota yerleştirmede instructorları sorumlu olduğu proje sayılarını sıraladıktan sonra en yüksek sayıda ve aynı zamanda az slot seçeneği olan veya çakışma potansiyeli yüksek projeler önce yerleştirilebilir bu şartı sağlayanı ilk olarak yerleştirmeliyiz! Sonrasında gerekli şartları sağlayacak şekilde devam et!

(Proje toplam sayısı x 2) bizim toplam iş yükümüz. Bu sayıyı Öğretim Görevlisi sayımıza böldüğümüz zaman bizim Öğretim Görevlisi başına düşen ortalama proje iş yükü sayısını bulmuş oluyoruz. Çıkan sonuç tam sayıysa Öğretim Görevlisi Başına Düşen İş Yükümüzdür. Eğer değilse çıkan sonucu bir üst integer değere yuvarla!

Denklem: "Öğretim Görevlisi Başına İş Yükü = (Proje toplam sayısı x 2) / Öğretim Görevlisi Sayısı"

(Öğretim Görevlisi Başına İş Yükü - Öğretim Görevlisinin Sorumlu Olduğu Proje Sayısı) sayısından Öğretim Görevlisinin atanması gereken toplam juri sayısını bulduk. Öğretim Görevlisinin proje sayısını juri sayısı ile toplam iş yükünü vermektedir. Sonrasında tüm Öğretim Görevlilerinin iş yükünü topla. Eğer iş yükü sayımız önceden belirlediğimiz toplam iş yükü sayımızdan fazlaysa (Bunun olması muhtemel çünkü Öğretim Görevlisi başına düşen iş yükümüzü üst integer değere yuvarlıyoruz! Bu yüzden de arada fark olmasını beklememiz doğaldır.) aradaki farkı (Bulduğumuz İş Yükü - Beklenen İş Yükü) törpülememiz gerekecek. Bulduğumuz fark kadar kişiden birer iş yükü çıkarmamız gerekiyor (Örneğin: 103 tane projemiz olsun, beklenen iş yükümüzse 100 olsun. Buradan anlayacağımız 3 tane beklediğimizden fazla iş yükü çıktı. Buradaki 3 fazlalık iş yükünü 3 Öğretim Görevlisinden 1 iş yükü azaltacak şekilde çıkartacaksın! Mesela genel ortalamamız 8 olsun bu 3 kişinin iş yükü 7 olsun. Bu 3 öğretim görevlisinden çıkartmamız gerekecek.)

İki adımlı algoirtma. Öncelikle proje optimizasyonu yaparak projeleri yerleştirebilirsın. Sonrasında jürileri optimize ederek

Kısıtlarımız:

- Algoritma sınıfları ilk timeslottan başlayarak doldurmak zorunda!

- Timeslot arasında herhangi bir GAP olmayacak! Projelerin timeslotlarının atamasını yaparken projeler arasında boş timeslot olmamalı! Yani ilk timeslottan başlayarak projeler peşpeşe gelmeli!
- Tüm Ara projeler ve tüm Bitirme projeleri timeslotlara atanmak zorunda!
- Projenin Sorumlu kişi zaten kendi projeleri ile bağlantılı. Burada bunun dışında bir senaryo gerçekleşirse otomatik müdahale et.
- Bir öğretim görevlisi kendi Proje Sorumluluğunda olduğu projenin **jürisi olamaz**.
- Her öğretim görevlimiz, her timeslotta **en fazla 1 görev** alabilir.
- Kaça adet sınıf olması gerekiği sistemden seçildikten sonra seçilen sınıf sayısı kadar sınıf algoritmalarımızda kullanılmalı. Daha az ya da daha çok sayıda sınıf kullanılmamalı! Tam olarak seçilen sınıf sayısı kadar sınıf kullanılmalı!

Cezalarımız:

-Öğretim Görevlilerimizin (Sorumlu Oldukları ya da Jüri oldukları projeler için) projelerinin arka arkaya olmasını istiyorum. Eğer Öğretim Görevlimizin projelerinin aralarında başka Öğretim Görevlilerine atanmış projeler varsa bunlar cezadır. Burada Öğretim Görevlimizin Proje Sorumlusu ya da Jüri olması bizim için önemli bir ayırım değildir. Bir timeslot'ta Proje Sorumlusu olarak atanmışken hemen ardından gelen timeslotta jüri olabilir. Bu durum ceza gerektirmez. (Bu hem proje sorumlularımız için hem de jürilerimiz içindir)

-Öğretim Görevlilerimizin bulunmuş oldukları sınıftan başka bir sınıfa, sınıf değişikliği yapması bizim için bir cezadır. Burada Öğretim Görevlimizin Proje Sorumlusu ya da Jüri olması bizim için önemli bir ayırım değildir. Bir sınıfa bağlı timeslot'ta Proje Sorumlusu olarak atanmışken aynı sınıfa bağlı başka bir timeslot'ta jüri olabilir. Bu durum ceza gerektirmez. (Bu hem proje sorumlularımız için hem de jürilerimiz için geçerlidir.)

Ceza Fonksiyonu (matris tabanlı)

Girdiler

- Bir öğretim görevlisi için, zaman sırasına göre **M** matrisi ($k \times 2$):
 - $M[r, 0] = \text{saat}_r$ (ör: 9.0, 9.5, 10.0 ...)
 - $M[r, 1] = \text{sınıf}_r$ (ör: D105, D106 ...)
- **Slot uzunluğu: Δ** . (Örn: 0.5 saat)

- Tolerans: ε (kayan nokta/yuvarlama hatası için, örn: 0.001)
- Ağırlıklar: α (ardışıklık/bekleme cezası), β (sınıf değişimi cezası)

Kurallar

r 'nci satırдан $(r+1)$ 'inci satıra geçerken iki kontrol yapılır:

1. Ardışıklık (zaman) kontrolü

- Beklenen saat: $\text{saat}_r + \Delta$
- Eğer $| \text{saat}_{r+1} - (\text{saat}_r + \Delta) | > \varepsilon$ ise **ardışık değil** \Rightarrow zaman cezası.
- İstersen “boşluk sayısına göre” cezayı artırabilirsin (aşağıda iki seçenek).

2. Sınıf kontrolü

- Eğer $\text{sınıf}_{r+1} \neq \text{sınıf}_r$ ise **sınıf değişimi** \Rightarrow sınıf cezası.

Not: İkisi de ihlal edilirse **ikisi de** uygulanır (toplanır).

İlk satır için ceza hesaplanmaz; kontrol her zaman çift $(r, r+1)$ üzerinden yapılır.

Ceza Tanımları

Varsayılan (basit) versiyon — ikili (binary) ceza

- Zaman cezası (ardışık değilse 1, aksi halde 0):

$$\text{time_penalty}_{r \rightarrow r+1} = \mathbb{1}[| \text{saat}_{r+1} - (\text{saat}_r + \Delta) | > \varepsilon]$$

- Sınıf cezası (sınıf değiştiyse 1, aksi halde 0):

$$\text{class_penalty}_{r \rightarrow r+1} = \mathbb{1}[\text{sınıf}_{r+1} \neq \text{sınıf}_r]$$

- Toplam ceza (tek öğretim görevlisi için):

$$\text{Ceza} = \alpha \sum_{r=1}^{k-1} \text{time_penalty}_{r \rightarrow r+1} + \beta \sum_{r=1}^{k-1} \text{class_penalty}_{r \rightarrow r+1}$$

İsteğe bağlı —“boşluk büyülüğüne göre” orantılı zaman cezası

Eğer 9.0'dan 10.0'a atladıysa, arada **1 slot** boşluk var ($\Delta=0.5$ ise 9.5 boş kaldı).

Genelleştirelim:

$$g_r = \max(0, \text{round}(\frac{\text{saat}_{r+1} - \text{saat}_r}{\Delta}) - 1)$$

- $g_r = 0 \rightarrow$ ardışık (ceza yok)
- $g_r = 1 \rightarrow$ 1 slot boşluk
- $g_r = 2 \rightarrow$ 2 slot boşluk ...

Zaman cezasını g_r ile çarparak “boşluk sayısı kadar” uygula:

$$\text{time_penalty}_{r \rightarrow r+1} = g_r$$

Toplam ceza yine α ve β ile ağırlıklanır.

Kısa örnek

- $\Delta = 0.5, \varepsilon = 0.001$
- Satırlar:
 1. (9.0, D105)
 2. (9.5, D105) — **ardışık & aynı sınıf** \rightarrow ceza yok
 3. (10.5, D106) — **ardışık değil** (10.0 beklenirdi), **sınıf değişti**
- Binary versiyon:
 - o (1 \rightarrow 2): time=0, class=0
 - o (2 \rightarrow 3): time=1, class=1
 - o **Ceza** = $\alpha \cdot 1 + \beta \cdot 1$
- Orantılı versiyon:
 - o (2 \rightarrow 3) arası boşluk: $(10.5 - 9.5)/0.5 = 2\text{slot} \rightarrow g = 2 - 1 = 1$
 - o **Ceza** = $\alpha \cdot 1 + \beta \cdot 1$ (bu örnekte aynı çıktı; daha büyük atlamalarda zaman cezası büyür)

Amaç Fonksiyonu:

-Ceza fonksiyonunu minimum tutmaya çalışıyoruz. Ayrıca:

Tanımlar

- $i \rightarrow$ öğretim görevlisi ($i = 1 \dots N$)
- $r \rightarrow$ öğretim görevlisinin görev sırası ($r = 1 \dots k_i$)
- $\text{Saat}_{i,r} \rightarrow$ i. öğretim görevlisinin r. görev saat bilgisi

- $Sınıf_{i,r} \rightarrow$ i. öğretim görevlisinin r. görev sınıf bilgisi
 - $\Delta \rightarrow$ slot uzunluğu (örneğin 0.5 saat)
 - $\varepsilon \rightarrow$ tolerans (kayan nokta farkları için)
 - $\alpha \rightarrow$ zaman cezası katsayısı
 - $\beta \rightarrow$ sınıf değişimi cezası katsayısı
-

Ceza terimleri

Zaman cezası (ardışık değilse)

$$\text{ZamanCezası}_{i,r} = \mathbb{1}[|\text{Saat}_{i,r+1} - (\text{Saat}_{i,r} + \Delta)| > \varepsilon]$$

veya orantılı versiyon (boşluk sayısına göre):

$$\text{ZamanCezası}_{i,r} = \max(0, \text{round}\left(\frac{\text{Saat}_{i,r+1} - \text{Saat}_{i,r}}{\Delta}\right) - 1)$$

Sınıf cezası (sınıf değiştiyse)

$$\text{SınıfCezası}_{i,r} = \mathbb{1}[Sınıf_{i,r+1} \neq Sınıf_{i,r}]$$

Amaç fonksiyonu

Toplam ceza = tüm öğretim görevlileri + tüm görev geçişleri üzerindeki zaman ve sınıf cezalarının ağırlıklı toplamı.

$$\min Z = \alpha \sum_{i=1}^N \sum_{r=1}^{k_i-1} \text{ZamanCezası}_{i,r} + \beta \sum_{i=1}^N \sum_{r=1}^{k_i-1} \text{SınıfCezası}_{i,r}$$

Notlar

- Bu amaç fonksiyonu, sistemin **toplam bekleme süresini ve sınıf değişikliklerini** minimuma indirir.
- α ve β değerleriyle hangi cezanın daha öncelikli olacağını belirleyebiliriz:

- $\alpha > \beta$: Boş beklemeyi minimize eder.
 - $\beta > \alpha$: Sınıf değişimini minimize eder.
- Bu amaç fonksiyonu herhangi bir **zamanlama / atama modeli** ile birlikte çalışabilir.
 - ZamanCezasive SınıfCezasıklılık (0/1) ya da tamsayı değerli değişkenlere bağlı olarak ILP/CP-SAT çözüçülerinde rahatlıkla kullanılabilir.