

İŞ GÜCÜ ÇİZELGELEME

ENDÜSTRİYEL ÇİZELGELEME DERSİ
PROF. DR. İNCİ SARIÇİÇEK

Eskişehir Osmangazi Üniversitesi
Mühendislik Mimarlık Fakültesi
Endüstri Mühendisliği

İŞ GÜCÜ ÇİZELGELEME

Her çalışanın görev ve tatil günlerini belirli bir zaman periyodu için gösteren çizelgelerdir.

Toplu üretim planlarında belirlenen iş gücü planlarının günlük çizelgelere dönüştürülmesi **iş gücü çizelgelemedir.**

Örneğin, hemşire, itfaiye ekibi, polis ve emniyet gücü gibi görevlerin çizelgelenmesi...

İŞ GÜCÜ ÇİZELGELEME

İş gücü çizelgeleme problemleri genel olarak üç gruba ayrılır:

- İzin Günü Çizelgeleme Problemleri (days- off scheduling)
 - TPB Algoritması
 - Monroe's Algoritması
- Vardiya Çizelgeleme Problemleri (shift scheduling)
- Tur Çizelgeleme Problemleri (tour scheduling)



BİRBİRİNİ İZLEYEN İZİN GÜNLERİ ÇİZELGELEME PROBLEMLERİ

Haftanın yedi günü faaliyet gösteren ve her iş gören için ardışık iki gün tatil sağlayan bir işletme için yapılan çizelgeleme problemi.

“Cyclical
Scheduling”

Amaç; atıl kapasiteyi en küçükleyecek şekilde her iş gören için ardışık tatil günlerini belirlemektir.

Havaalanı, hastane, lokanta vb.

TİBREWALE, PHİLİPPE AND BROWNE (TPB) ALGORİTMASI

Örnek : Günlük ihtiyaç duyulan işçi sayısı aşağıda verilmiştir.

| Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
|-----------|------|----------|----------|------|-----------|-------|
| 6 | 8 | 10 | 7 | 12 | 4 | 2 |

İşçilerin birbirini izleyen iki günde tatil yapabilmelerini sağlayacak bir çizelgeleme yapınız.



| Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
|-----------|------|----------|----------|------|-----------|-------|
| 6 | 8 | 10 | 7 | 12 | 4 | 2 |

○ **Adım 1:** Ardışık günlerde gereksinim duyulan işçilerin maksimum değerleri bulunur.

- Pazartesi ve Salı için $\text{enb}(6,8)=8$
- Salı ve Çarşamba için $\text{enb}(8,10)=10$
- Çarşamba ve Perşembe için $\text{enb}(10,7)=10$
- Perşembe ve Cuma için $\text{enb}(7,12)=12$
- Cuma ve Cumartesi için $\text{enb}(12,4)=4$
- Cumartesi ve Pazar için $\text{enb}(4,2)=2$
- Pazar ve Pazartesi için $\text{enb}(2,6)=6$

➤ Bulunan değerler arasından en küçüğü seçilir.

➤ $\text{Enk}(8, 10, 10, 12, 12, 4, 6)=4$



- **Adım 2:** En küçük karşı gelen değer Cumartesi ve Pazar gününe denk gelmektedir. Yani ilk işçi bu günlerde tatil yapıp diğer günlerde çalışacaktır. Bu yüzden Cumartesi ve Pazar günündeki işçi gereksinimleri sabit kalır fakat diğer günlerin gereksiniminden 1 kişi çıkarılır ve adım 1 tekrar uygulanır.

| Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
|-----------|------|----------|----------|------|-----------|-------|
| 6 | 8 | 10 | 7 | 12 | 4 | 2 |
| 5 | 7 | 9 | 6 | 11 | 4 | 2 |



- 3. tekrar da ise eşitlik durumu ile karşılaşilmektedir.

| Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
|-----------|------|----------|----------|------|-----------|-------|
| 6 | 8 | 10 | 7 | 12 | 4 | 2 |
| 5 | 7 | 9 | 6 | 11 | 4 | 2 |
| 4 | 6 | 8 | 5 | 10 | 4 | 2 |

- Cumartesi ve Pazar için $\text{enb}(4,2)=4$

- Pazar ve Pazartesi için $\text{enb}(2,4)=4$

Bu durumda çiftlerin toplamına bakılıp en küçüğü tercih edilir.

- $\left. \begin{array}{l} \text{Cumartesi} + \text{Pazar} = 4 + 2 = 6 \\ \text{Pazar} + \text{Pazartesi} = 2 + 4 = 6 \end{array} \right\}$ eşitlik söz konusu istenilen seçilebilir.

- Tüm değerler sıfır oluncaya kadar bu işlemlere devam edilir. Sonuç tablosu aşağıdaki gibidir.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------|----|----|----|---|---|---|---|---|---|----|----|----|----|
| Pazartesi | 6 | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| Salı | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| Çarşamba | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 |
| Perşembe | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cuma | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Cumartesi | 4 | 4 | 4 | 4 | 3 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| Pazar | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |

- **Adım 3:** İzine çıkılan günler ve çıkan işçi sayısı toplamı belirlenir.
- Pazartesi izne çıkan kişi sayısı: 4
- Salı=Çarşamba=Perşembe= Cuma =0
- Cumartesi izne çıkan kişi sayısı= 6
- Pazar günü izne çıkan kişi sayısı= 2



- Adım 4: İzne çıkan kişi sayısının toplamı ise toplam işgücü gereksinimini verir.

$$4 + 6 + 2 = 12$$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|-----------|----|----|----|---|---|---|---|---|---|----|----|----|----|
| Pazartesi | 6 | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| Salı | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| Çarşamba | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 |
| Perşembe | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Cuma | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Cumartesi | 4 | 4 | 4 | 4 | 3 | 3 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| Pazar | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |

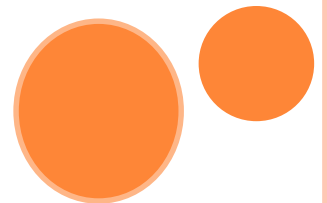
| | Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar | Toplam |
|-------------------|-----------|------|----------|----------|------|-----------|-------|--------|
| İhtiyaçlar | 6 | 8 | 10 | 7 | 12 | 4 | 2 | 49 |
| Çalışmayan sayısı | 6 | 4 | 0 | 0 | 0 | 6 | 8 | |
| Çalışan sayısı | 6 | 8 | 12 | 12 | 12 | 6 | 4 | 60 |

VERİMLİLİK

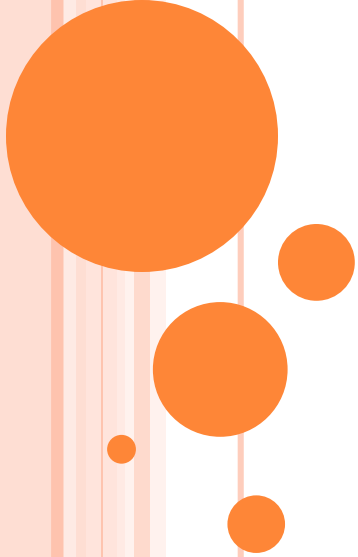
İhtiyaç Duyulan İşçi Sayısı

○ Verimlilik =
$$\frac{\text{İhtiyaç Duyulan İşçi Sayısı}}{\text{Çalışan İşgören Sayısı}}$$

$$= 49 / 60 = 0.817 = \%81.7$$



DÖNÜŞÜMLÜ İZİN GÜNLERİ



AMAÇ

- Mümkün olduğunca çok çalışana 2 özel gün izin (hafta sonları) verirken toplam iş gücünü makul bir sınırdan tutmak için diğer çalışanlara ardışık olmayan izin günleri vermek



BU YÖNTEMLE...

- Hem çizelgenin yüksek etkinliği sağlanır hem de hafta sonları için gereken iş gücü ihtiyacı karşılanır
- Bazı işçilerin hafta sonu izinleri varken diğerlerinin dağınık izin günleri olduğu bir çizelgede sabit zaman dilimleri ayrımcı tavırlar yaratır
- Oysa dönüşümlü çizelge eşitlik sağlar
- Dönüşümlü çizelgede çalışanlar er yada geç bütün zaman dilimlerinde çalışırlar



BURNS & CARTER ALGORİTMASI



Dönüşümlü izin günleri yönteminin uygulanabilmesi için Burns ve Carter 1985'te bir algoritma geliştirmişlerdir

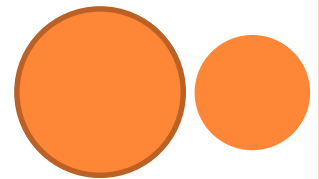
ALGORİTMANIN ÖZELLİKLERİ

- Haftanın her günü için ihtiyaç duyulan çalışan sayısı değişebilir
- Algoritmanın çözümünde şu kısıtlar vardır:
 - Her işçi haftada 5 gün çalışacak,
 - Her işçi birbirini takip eden hafta sonlarından en az birinde izinli olmalıdır,
 - Hiçbir işçi ardışık 6 günden daha fazla çalışamaz



ALGORİTMA BAŞLANGICI

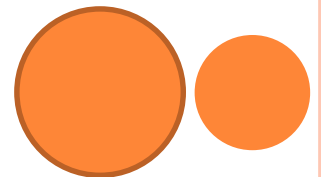
- Algoritma, haftalık ihtiyacı karşılayacak minimum çalışanı (W) belirleyerek başlar:
 - Hafta Sonu Kısıtı (L1) : $W \geq 2n$
 - Toplam Talep Kısıtı (L2) : $W \geq \frac{1}{5} \sum_{i=1}^7 n_i$
 - En Büyük Günlük Talep Kısıtı (L3) : $W \geq \text{Mak}(n_i)$



ALGORİTMA ADIMLARI

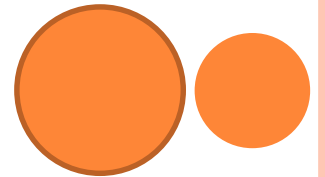
ADIM-1

- Minimum çalışan sayısını hesaplamak için bulunan alt sınır değerlerinden (L1,L2,L3) en büyüğünü toplam işgücü olarak belirle



ADIM-2

- Hafta sonunun her günü n çalışana ihtiyaç vardır
- Bu yüzden geriye kalan W-n çalışana hafta sonu izni verilebilir
- İlk hafta sonu iznine ilk W-n çalışanı ata. İkinci hafta sonu iznine sıradaki W-n çalışanı ata. Bu işlemi yineleme oluşuncaya kadar devam et



ADIM-3

- Toplam işgücünde (W), günlük talebi çıkararak her gün için çalışan fazlalığını (S_j) hesapla.

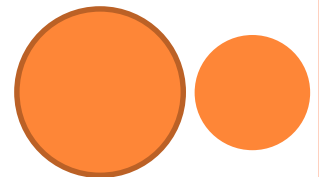
$$S_j = W - n_j$$

- Hafta sonları n çalışan mevcut olduğundan dolayı çalışan fazlalığı, en büyük hafta sonu talebinden (n) günlük talebi çıkararak bul



ADIM-3 DEVAM

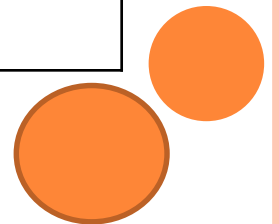
- Aşağıdaki adımları tekrarlayarak n tane çift izin gününün listesini oluştur:
 - $S_k = \max(S_j)$ olan k. gününü seç
 - $S_i > 0$ ve $i \neq k$ olan i gününü seç Eğer bütün $i \neq k$ için $S_i = 0$ ise i'yi k olarak ata.
 - (k,i) çiftini listeye ekle S_k ve S_i değerlerini 1 azalt
 - Bu işlemleri n kez tekrarla



ADIM-4

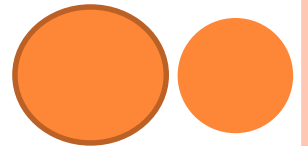
- Birinci haftada, çalışanlar haftanın başında ve sonunda izinli olmalarına göre aşağıdaki tabloda gösterildiği gibi 4 gruba ayrılırlar

| Kategori | Hafta sonu-1 | Hafta-1 | Hafta sonu-2 |
|----------|--------------|---------------------|--------------|
| Tip T1 | İzinli | İzne gerek yok | İzinli |
| Tip T2 | İzinli | 1gün izne gerek var | Çalışıyor |
| Tip T3 | Çalışıyor | 1gün izne gerek var | İzinli |
| Tip T4 | Çalışıyor | 2gün izne gerek var | Çalışıyor |

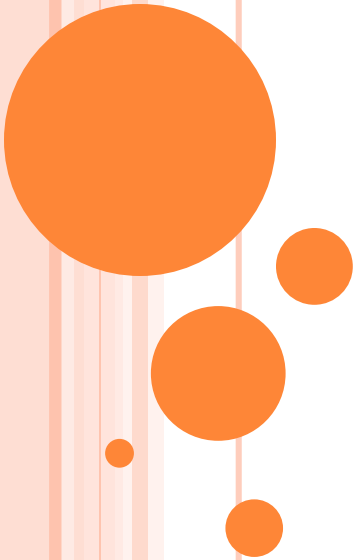


ADIM-4 DEVAM

- Hafta sonları n çalışan olduğundan dolayı T2 ve T3 çalışan sayısı birbirine özdeştir
- Bu özdeşlik T2 çalışanı ile T3 çalışanını eşleştirmeyi sağlar. Eğer T2 çalışan en erken izni alırsa eşleniği T3 de en geç izni alır
- Bu yolla da her T4 çalışanı 2 gün izin alır ve her T2 ve T3 çalışanı da gerektiği gibi pazartesten cumaya kadar bir gün izin alır



ÖRNEK ÇÖZÜM



VERİLER

Her gün gereken çalışan sayıları:

| Pazar | Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi |
|-------|-----------|------|----------|----------|------|-----------|
| 2 | 6 | 8 | 10 | 7 | 12 | 4 |

- Amacımız her çalışana, 2 haftalık zaman diliminde en az 1 kez hafta sonu izni vermek



ADIM-1

ADIM-1

- $L_1 = 2 * n = 2 * 4 = 8$

$$L_2 = w \geq \frac{1}{5} \sum_{i=1}^7 n_i = 49/5 = 10$$

$$L_3 = \text{Mak}(n_i) = 12$$

- $\text{Mak}(L_1, L_2, L_3) = 12$



ADIM-2

[illegible]

ADIM-3

ADIM-3

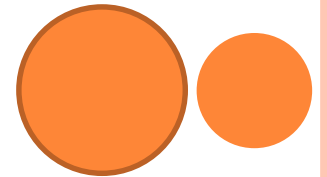
| İTERA. | PAZAR (2) | PAZAR. (6) | SALI (8) | ÇARŞA. (10) | PERŞE. (7) | CUMA (12) | CUMA. (4) |
|--------|--------------|---------------|-------------|----------------|---------------|--------------|--------------|
| 1 | 2 | 6 | 4 | 2 | 5 | 0 | 0 |
| 2 | 2 | 5 | 4 | 2 | 4 | 0 | 0 |
| 3 | 2 | 4 | 3 | 2 | 4 | 0 | 0 |
| 4 | 2 | 4 | 2 | 2 | 3 | 0 | 0 |

DİKKAT: Seçilecek çift önceki adımlarda seçilen çiftler ile aynı olmamalı



ADIM-3 DEVAM

| İTERASYON | İZİN GÜNÜ ÇİFTLERİ |
|-----------|--------------------|
| 1 | Pazartesi-Perşembe |
| 2 | Pazartesi-Salı |
| 3 | Salı-Perşembe |
| 4 | Pazartesi-Çarşamba |



ADIM-4

[illegible]

ADIM-4 DEVAM

ADIM-3

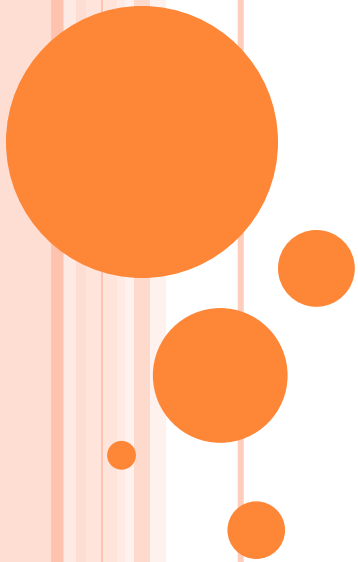
| Ç | C | P | P | S | Ç | P | C | C | P | P | S | Ç | P | C | C | P | P | S | Ç | P | C | C | P |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 0 | | | | 0 | 0 | | | | | | 0 | 0 | | | | | 0 | | | |
| 2 | | | 0 | | | | 0 | 0 | | | | | | 0 | 0 | | | 0 | | | | | |
| 3 | | | | 0 | | | 0 | 0 | | | | | | 0 | 0 | | | | | 0 | | | |
| 4 | | | 0 | | | | 0 | 0 | | | | | | 0 | 0 | | | | 0 | | | | |
| 5 | 0 | 0 | | | | | | | | 0 | | | | 0 | 0 | | | | | | | 0 | 0 |
| 6 | 0 | 0 | | 0 | | | | | | 0 | | | | 0 | 0 | | | | | | | 0 | 0 |
| 7 | 0 | 0 | | | | | | | | | 0 | | | 0 | 0 | | | | | | | 0 | 0 |
| 8 | 0 | 0 | | | | | | | | | 0 | | | 0 | 0 | | | | | | | 0 | 0 |
| 9 | 0 | 0 | | | | | 0 | 0 | | | | | | | | | 0 | | | | | 0 | 0 |
| 10 | 0 | 0 | | | | | 0 | 0 | | | 0 | | | | | | 0 | | | | | 0 | 0 |
| 11 | 0 | 0 | | | | | 0 | 0 | | | | | | | | | | 0 | | | | 0 | 0 |
| 12 | 0 | 0 | | | | | 0 | 0 | | | | | | | | | 0 | | | | | 0 | 0 |

SONUÇ

- Bütün çalışanlar, algoritma koşulları sağlanarak çizelgelenmiştir
- Çizelgenin verimliliği= $49/60 = \%81$ 'dir



MONROE ALGORITMASI



AMAÇ

- Algoritma en az işçi sayısı, W ile çalışanlara ardışık 2 izin günü vermeye çalışır
- Eğer ardışık izin günleri verilemiyorsa:
 - Günlük gereksinimler kaydırılarak,
 - W 'yi arttırarak veya
 - Bazı işçilere ardışık olmayan izin günleri verilerek çözüm aranır



ALGORİTMA ADIMLARI

ADIM-1

- Gereken teorik minimum işçi sayısını hesapla(W)

$$W = (\max(T/5, n_j))$$

- Eğer W tamsayı değilse bir üst tamsayıya yuvarla

ADIM-2

- Düzenli İzin Günlerini (DİG), her günün gereksinimini W ' den çıkararak hesapla

$$DİG_j = W - n_j$$

ADIM-3

- Ardışık gün çiftlerini oluştur

PAZARTESİ-SALI (PS)

SALI-ÇARŞAMBA (SÇ)

...

CUMA-CUMARTESİ (CC)

CUMARTESİ-PAZAR (CP)

PAZAR-PAZARTESİ (PP')

PAZARTESİ-SALI (P'S')



ADIM-3 DEVAM

- İlk iterasyonun başlangıcını şu şekilde yap:
 - Salı gününün DİG'sinin yarısını PS çiftine ata

ADIM-4

- PS çifti değerini salının DİG' sinden çıkar
- Kalan değeri SÇ çiftine ata
- SÇ atamasını çarşambanın DİG' den çıkar ve kalan değeri ÇP çiftine ata
- Bu prosedüre P'S' çiftine değer atayana kadar devam et

ADIM-5

- Eğer P'S' değeri PS değerine eşit ise ADIM 6' ya git
- Eğer eşit değilse, ikinci iterasyonu yap
- PS çiftine ikinci iterasyon için başlangıç atamasını, ilk iterasyondaki PS ve P'S' değerlerinin ortalaması olarak başla
- Prosedürü, kalan gün çiftlerinin atanması için ADIM 4' den uygula



ADIM-5 DEVAM

- Bu noktada, eğer fazla değişiklik olmadan problemin çözümü varsa (PS değeri $P'S'$ değerine eşittir ve tüm çiftler için değerler pozitiftir) ADIM 6' ye git değilse ADIM7'ye git

ADIM-6

- Atanan sayılar, gün çiftlerindeki izinli işçilerin sayısını gösterir.



ADIM-7

- Bu adıma sadece uygun çözüm yoksa gelinir.
- Çözümü uygun yapabilir üç yol vardır:



ÇÖZÜM-1

- İkinci iterasyonda maksimum negatif değeri olan çifti bul, iterasyon sayısının mutlak değerini n olarak ata
- Bu gün çiftinin günlük gereksinimi n azaltarak DİG değerini n arttırıp, başka bir çiftin DİG'sini negatif yapmadan n düşürerek sonuca varılabilir.
- Böylece ADIM 1' den 6' ya tekrar uygulandığında, uygun bir çözüm olacak.



ÇÖZÜM-2

- Toplam işçi sayısını n artır

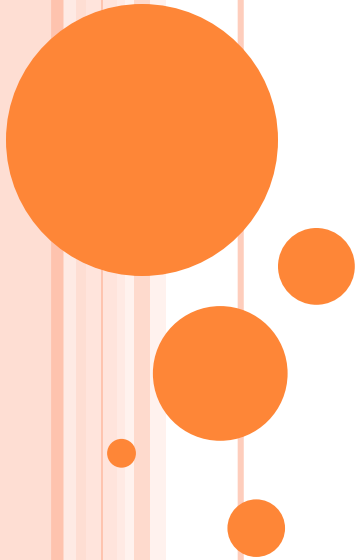


ÇÖZÜM-3

- Bazı ardışık olmayan izin günlerinin olduğu bir sıra geliştirir
- Bu sıradaki ardışık olmayan izin günü sayısını en asgaride tut bu da n değerine eşittir
- Bütün ardışık olmayan izin günü çiftlerinin bir günü mutlaka en büyük DİG'ye sahip olan güne atanmalıdır



ÖRNEK ÇÖZÜM



VERİLER

Bir haftada 7 gün çalışan tamir atölyesinin günlük işçi gereksinimleri şu şekildedir:

| PAZAR. | SALI | ÇARŞA. | PERŞE. | CUMA | CUMAR. | PAZAR |
|--------|------|--------|--------|------|--------|-------|
| 5 | 4 | 3 | 5 | 5 | 4 | 3 |

Her işçiye 2 ardışık izin günü veren çizelgelemeyi geliştiriniz



ADIM-1

○ $W = (\max(T/5, n_j)) = (\max (29/5 , 5, 4, 3, 5, 5, 4, 3)$

$$W = 6 \text{ işçi}$$

ADIM-2

W=6

| | Paz. | Salı | Çar. | Per. | Cum a | Cum . | Paza r | S.Pa z. | Topl. |
|----------------|------|------|------|------|----------|----------|-----------|------------|-------|
| Gereksini m | 5 | 4 | 3 | 5 | 5 | 4 | 3 | 5 | 29 |
| Aşm a | | | 1 | | | | | | |
| Topla m | 5 | 4 | 4 | 5 | 5 | 4 | 3 | 5 | 30 |
| DİG | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | |

ADIM-3

| | Paz. | Salı | Çar. | Per. | Cuma | Cum. | Pazar | S.Paz. | Topl. |
|------------------------|------------|------|------|------|------|------|-------|--------|-------|
| Gereks inim | 5 | 4 | 3 | 5 | 5 | 4 | 3 | 5 | 29 |
| Aşma | | | 1 | | | | | | |
| Toplam | 5 | 4 | 4 | 5 | 5 | 4 | 3 | 5 | 30 |
| DiG | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | |
| Ardşk Gün Çiftlr | PS | SÇ | ÇP | PC | CC | CP | PP' | P'S' | |
| İlk İtrsyn | (2÷2) 1 | | | | | | | | |

ADIM-4

| | Paz. | Salı | Çar. | Per. | Cuma | Cum. | Pazar | S.Paz. | Topl. |
|-------------------------|------|------|------|------|------|------|-------|--------|-------|
| Gereks inim | 5 | 4 | 3 | 5 | 5 | 4 | 3 | 5 | 29 |
| Aşma | | | 1 | | | | | | |
| Toplam | 5 | 4 | 4 | 5 | 5 | 4 | 3 | 5 | 30 |
| DiG | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | |
| Ardışk Gün Çiftlr | PS | SÇ | ÇP | PC | CC | CP | PP' | P'S' | |
| İlk İtrsyn | 1 → | 1 → | 1 → | 0 → | 1 → | 1 → | 2 → | -1 | |



ADIM-5

| | Paz. | Salı | Çar. | Per. | Cuma | Cum. | Pazar | S.Paz. | Topl. |
|--------------------------|------|------|------|------|------|------|-------|--------|-------|
| Gereksinim | 5 | 4 | 3 | 5 | 5 | 4 | 3 | 5 | 29 |
| Aşma | | | 1 | | | | | | |
| Toplam | 5 | 4 | 4 | 5 | 5 | 4 | 3 | 5 | 30 |
| DİG | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | |
| Ardışık Gün Çiftlr | PS | SÇ | ÇP | PC | CC | CP | PP' | P'S' | |
| İlk İtrsyn | 1 | 1 | 1 | 0 | 1 | 1 | 2 | -1 | |
| İkinci İtrsyn | 0 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | |

$$(PS + P'S') \div 2 = (1 + (-1)) \div 2 = 0$$

ADIM-6

| | Paz. | Salı | Çar. | Per. | Cuma | Cum. | Pazar | S.Paz. | Topl. |
|------------------------------------|------|------|------|------|------|------|-------|--------|-------|
| Ardışık Gün Çiftlr | PS | SÇ | ÇP | PC | CC | CP | PP' | P'S' | |
| İlk İtrsyn | 1 | 1 | 1 | 0 | 1 | 1 | 2 | -1 | |
| İkinci İtrsyn | 0 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | |
| Tatil Yapacak İşçi Sayısı | | 2 | 2 | 1 | 1 | 2 | 3 | 1 | |

İŞÇİ-1

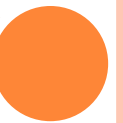
İŞÇİ-2

İŞÇİ-3

İŞÇİ-4

İŞÇİ-5

İŞÇİ-6



SONUÇ

- Bütün çalışanlar, algoritma koşulları sağlanarak çizelgelenmiştir
- Çizelgenin verimliliği= $29/30 = \%96$ 'dir



TUR ÇİZELGELEME

- Haftanın yedi günü ve gün içinde de birden çok vardiya çalışan işletmelerde uygulanır.

(Oteller, havaalanları, polis istasyonları)

- Amaç:

- Her bir çalışma günü ve saati için ihtiyaç duyulan işgücü sayısını minimum maliyetle karşılamak için,
- Her bir tura atanacak işgücü sayısının belirlenmesidir

KULLANILAN ALGORİTMA

- Bechtold ve Showalter (1987) tarafından geliştirilen basit bir algoritma;
 - Manuel vardiya çizelgeleme sezgiseli (MSSH)
- İki aşama:
 - 1. Günlük işgücü ihtiyacının belirlenmesi
 - 2. Tatil günlerinin belirlenmesi



BU ALGORİTMA:

- Her işçinin;
 - Aynı nitelikte olduğunu,
 - Günlük 8 saat çalışıp 1 saat ara verdiğini,
- Tüm işçilerin tam zamanlı çalıştığını,
- 1., 2., 3. ve 4. saatlerde çalışmaya başlayan 4 farklı vardiyanın olduğunu,
- Toplam çalışmanın 12 saat olduğunu,

varsaymaktadır.



Örnek: Günlük 12 saat açık olan bir lokanta için saat temelinde günlük gereksinimler aşağıdaki tabloda gösterilmiştir.

| Saatler | Gerekli İşçi Sayısı | | | | | | |
|---------|---------------------|------|----------|----------|------|-----------|-------|
| | Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
| 1 | 3 | 4 | 4 | 4 | 2 | 4 | 1 |
| 2 | 5 | 3 | 3 | 5 | 5 | 4 | 3 |
| 3 | 4 | 2 | 5 | 3 | 4 | 5 | 4 |
| 4 | 5 | 4 | 2 | 2 | 5 | 3 | 5 |
| 5 | 2 | 1 | 1 | 5 | 4 | 2 | 3 |
| 6 | 5 | 5 | 4 | 3 | 5 | 4 | 2 |
| 7 | 1 | 3 | 3 | 4 | 3 | 5 | 4 |
| 8 | 4 | 2 | 5 | 3 | 2 | 1 | 1 |
| 9 | 5 | 5 | 2 | 2 | 4 | 4 | 5 |
| 10 | 3 | 1 | 1 | 4 | 2 | 3 | 2 |
| 11 | 5 | 4 | 4 | 1 | 5 | 2 | 4 |
| 12 | 1 | 3 | 3 | 1 | 1 | 4 | 3 |

AŞAMA 1

- Her gün her vardiyada çalışmaya başlayacak işçi sayısı belirlenir.
- 4 adımdan oluşur.



ALGORİTMANIN ADIMLARI

- Adım 1:
 - İlk saatin tüm ihtiyacını karşıla.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Gereksinim | 3 | 5 | 4 | 5 | 2 | 5 | 1 | 4 | 5 | 3 | 5 | 1 |
| Adım 1 | 3 | 3 | 3 | 3 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Gereksinim | 0 | 2 | 1 | 2 | 2 | 2 | 0 | 1 | 2 | 3 | 5 | 1 |



ALGORİTMANIN ADIMLARI (DEVAM)

○ Adım 2 :

- Son saatin tüm ihtiyacını karşıla.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Gereksinim | 3 | 5 | 4 | 5 | 2 | 5 | 1 | 4 | 5 | 3 | 5 | 1 |
| Adım 1 | 3 | 3 | 3 | 3 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 |
| Gereksinim | 0 | 2 | 1 | 2 | 2 | 2 | 0 | 1 | 2 | 3 | 5 | 1 |
| Adım 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Gereksinim | 0 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | 4 | 0 |

ALGORİTMANIN ADIMLARI (DEVAM)

- Adım 3 :

- Tüm ihtiyaçlar karşılanmışsa dur.

karşılanmadı ise;

- Adım 4 'ü uygula.



ALGORİTMANIN ADIMLARI (DEVAM)

- Adım 4 :

- Kalan gereksinimlerin en yüksek olduğu saatin ihtiyacını 1 azalt.

[illegible]

ALGORİTMANIN ADIMLARI (DEVAM)

- Eğer birden fazla farklı saatte maksimum ihtiyaç oluşursa;
 - İkisini de aynı anda karşılamaya çalış.

Mümkün değilse;

- Keyfi olarak sadece birinin ihtiyacını azalt.



| | Saatler | | | | | | | | | | | | Vardiya | | | |
|------------------|---------|---|---|---|---|---|---|---|---|----|----|----|---------|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | A | B | C | D |
| işçi gereksinimi | 3 | 5 | 4 | 5 | 2 | 5 | 1 | 4 | 5 | 3 | 5 | 1 | 3 | | | 1 |
| ADIM 1 | 3 | 3 | 3 | 3 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | | | | |
| işçi gereksinimi | 0 | 2 | 1 | 2 | 2 | 2 | 0 | 1 | 2 | 3 | 5 | 1 | | | | |
| ADIM 2 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | |
| işçi gereksinimi | 0 | 2 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | 4 | 0 | | | 1 | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| işçi gereksinimi | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | | | | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| işçi gereksinimi | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | | | |
| ADIM 4 | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | | |
| işçi gereksinimi | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | | | | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| işçi gereksinimi | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | | | |
| ADIM 4 | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | | |
| işçi gereksinimi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | | | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| işçi gereksinimi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | 1 | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| işçi gereksinimi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| ADIM 4 | | | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | | | | | |
| Toplam işçi | | | | | | | | | | | | | 3 | 2 | 4 | 1 |

- Aşama 1, bütün günler için tekrarlanır.

| Vardiya | Pazartesi | Salı | Çarşamba | Perşembe | Cuma | Cumartesi | Pazar |
|---------|-----------|------|----------|----------|------|-----------|-------|
| A | 3 | 4 | 4 | 4 | 2 | 4 | 1 |
| B | 2 | 0 | 0 | 4 | 3 | 0 | 2 |
| C | 4 | 1 | 1 | 0 | 4 | 1 | 1 |
| D | 1 | 3 | 3 | 1 | 1 | 4 | 3 |



AŞAMA 2

- Günlük vardiyalara atanan her bir işçinin çalışma ve tatil günleri belirlenir.

(Monroe's veya TPB)

