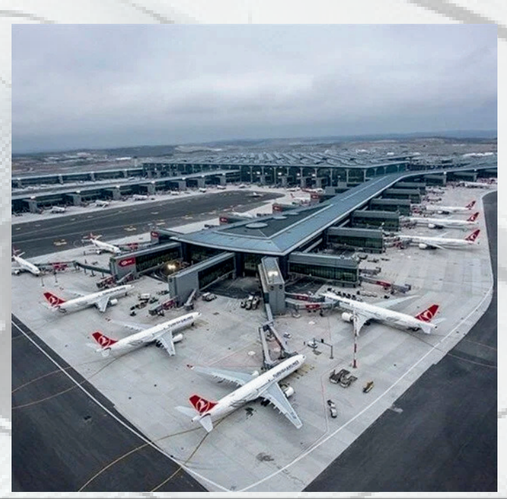
İzmir Bakırçay Üniversitesi
Mühendislik ve Mimarlık
Fakültesi Endüstri Mühendisliği
Bölümü

END 405 -SİSTEM ANALİZİ VE TASARIMI

Havaalanı Kapı Atama Problemi İçin Çok Amaçlı Hedef Programlama Yaklaşımı:İstanbul Havalimanı'nda Bir Vaka Analizi

Danışman: Doç.Dr. Mustafa Avcı

Gülcan Diker
Ezgi Sevinç
Gizem Dayınlarlı
Cansu Gündüz
Fatma Baştürk



GİRİŞ:

Havaalanı kapı atama problemi, uçuşların belirli zaman dilimlerinde uygun kapılara atanmasını sağlayarak apron kullanımını minimize etmeyi ve yolcuların terminal içinde daha hızlı erişim sağlamalarını hedefleyen önemli bir optimizasyon problemidir.

LİTERATÜR TARAMASI VE VERİLER:

- Mangoubi ve Mathaisel (1985)
- Cheng (1997)
- Haghani ve Chen (1997)
- Bolat (1999)
- Yan ve Huo (1999)
- Yan ve diğ.(2002)
- Dorndorf ve diğ. (2005)

Let's Start!

YÖNTEM VE TASARIM:

Editor - C:\Users\HUAWEI\Desktop\deneme2.m

deneme2.m | deneme3.m

```
clear;
clc
verii = xlsread('ucusss.xlsx', 'Sayfa2', 'S:T');
n = size(verii, 1);
cakismaMatrisi = zeros(n,n);
for i = 1:n-1
    for j = i+1:n
        if i ~= j
            % İniş ve kalkış zamanları çakışıyor mu?
            if (verii(j, 1) < verii(i, 2))
                cakismaMatrisi(i, j) = 1;  % Çakışma varsa 1
                cakismaMatrisi(j,i)=1;
            end
        end
    end
end
disp('Çakışma Matrisi:');
disp(cakismaMatrisi);
```

Bu çalışmada, havaalanı kapı atama problemini çözmek için hedef programlama yaklaşımı benimsenmiştir. Amaç, uçuşların kapılara atanmasında yolcu memnuniyetini artırırken operasyonel maliyetleri minimize etmektir. Problem modeli, çok amaçlı karar verme yapısında formüle edilerek MATLAB ve CPLEX optimizasyon araçları kullanılarak çözülmüştür. MATLAB, veri ön işleme, modelleme ve çözüm sürecinin entegrasyonu için bir platform sağlamış; CPLEX ise matematiksel modeli çözmek için kullanılmıştır.

```
int n = ...;  // Uçuş sayısı
int m = ...;  //pı sayısı (107 kapı)
int t = ...;  // Uçak tipi sayısı (ör. 0 = dar, 1 = geniş)

range Ucuslar = 1..n;  // Uçuşlar aralığı
range Kapilar = 1..m;  // Kapılar aralığı
range Ucaktipi = 1..t;  // Uçak tipleri aralığı
range apron=0..m;


int komsuluk[Kapilar][Kapilar] = ...;  // Kapılar arası komşuluk matrisi
int zaman_cakisma[Ucuslar][Ucuslar] = ...;  // Uçuşlar arasındaki zaman çakışması matrisi
int a[Ucuslar] = ...;  // Varış zamanları
int d[Ucuslar] = ...;  // Kalkış zamanları
int ucak_tipleri[Ucuslar] = ...;  // Uçak tipleri
int mesafeler[Kapilar] = ...;  // Kapılar arasındaki mesafeler (giriş kapısından her kapıya kadar)
int yolcu_sayisi[Ucuslar] = ...;  // Her uçuşun yolcu sayısı


dvar boolean x[Ucuslar][apron];  // Uçuşların kapılara atanması


minimize
    sum(i in Ucuslar, k in Kapilar) yolcu_sayisi[i] * mesafeler[k] * x[i][k];

subject to {

   // Uçuşların kapılara atanmasındaki maximizasyon amacı
    sum(k in Kapilar, i in Ucuslar) x[i][k]==467;

  // Her uçuş için bir kapı atanmalı
    forall(i in Ucuslar)
        sum(k in apron) x[i][k] == 1;

    // Uçuşlar aynı kapıyı aynı anda kullanamaz
    forall(k in Kapilar, i in Ucuslar, j in Ucuslar: i != j && zaman_cakisma[i][j] == 1)
        x[i][k] + x[j][k] <= 1;

    // Yan yana kapılara aynı anda geniş uçaklar gelememeli
    forall(k1 in Kapilar, k2 in Kapilar: komsuluk[k1][k2] == 1)
        forall(i in Ucuslar: ucak_tipleri[i] == 1, j in Ucuslar: ucak_tipleri[j] == 1 && i != j && zaman_cakisma[i][j] == 1)
            x[i][k1] + x[j][k2] <= 1;
}
```

n	Uçuş sayısı
m	Kapı sayısı
t	Gövde tipi sayısı
a_i	Uçuş i için varış zamanı
d_i	Uçuş i için kalkış zamanı
$mesafeler_k$	Kapılar arasındaki mesafeler
$yolcu_sayisi_i$	Her uçuşun yolcu sayısı
$ucak_tipleri_i$	Uçak tipleri

x_{ik} Uçuş i'nin k kapısına atanıp atanmadığını gösteren ikili değişken

$$zaman_cakisma_{ij} = \begin{cases} 1, & i \text{ ve } j \text{ uçuşları çakışıyor ise} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$komsuluk_{k_1k_2} = \begin{cases} 1, & k_1 \text{ ve } k_2 \text{ kapıları komşu ise} \\ 0, & \text{aksi takdirde} \end{cases}$$

$$min \sum_{i \in Ucuslar} \sum_{k \in Kapilar} yolcu_sayisi_i . mesafeler_k . x_{ik} \quad (1)$$

$$\sum_{k \in Kapilar} \sum_{i \in Ucuslar} x_{ik} = 467 \quad (2)$$

$$\sum_{k \in Apron} x_{ik} = 1 \quad \forall i \in Ucuslar \quad (3)$$

$$x_{ik} + x_{jk} \leq 1 \quad \forall k \in Kapilar, \forall i,j \in Ucuslar \ (i \neq j, zaman_cakisma_{ij} = 1) (4)$$

$$x_{ik1} + x_{jk2} \leq 1 \quad \forall k_1, k_2 \in Kapilar (komsuluk_{k_1k_2} = 1), \ \forall i,j \in Ucuslar \ (ucak_tipleri_i = 1, ucak_tipleri_j = 1, i \neq j, zaman_cakisma_{ij} = 1) \ (5)$$

DENEYSEL SONUÇLAR:

Uçuş	Kapı	Yolcu	Mesafe	Toplam Yürün
Uçuş 1	Kapı 1	118	13	1534
Uçuş 2	Kapı 3	310	10	3100
Uçuş 3	Kapı 2	160	13	2080
Uçuş 4	Kapı 5	279	12	3348
Uçuş 5	Kapı 1	153	50	7650
Uçuş 6	Kapı 7	147	10	1470
Uçuş 7	Kapı 2	159	13	2067
Uçuş 8	Kapı 3	92	10	920
Uçuş 9	Kapı 4	101	9	909
Uçuş 10	Kapı 9	344	14	4816
Uçuş 11	Kapı 5	135	12	1620
Uçuş 12	Kapı 11	132	12	1584
Uçuş 13	Kapı 6	162	14	2268
Uçuş 14	Kapı 13	132	10	1320
Uçuş 15	Kapı 8	158	16	2528
Uçuş 16	Kapı 10	141	11	1551
Uçuş 17	Kapı 15	346	14	4844
Uçuş 18	Kapı 17	255	13	3315
Uçuş 19	Kapı 12	102	14	1428
Uçuş 20	Kapı 14	102	13	1326

Uçuş	Kapı	Yolcu	Mesafe	Toplam Yü
Uçuş 1	Kapı 102	118	8	944
Uçuş 2	Kapı 50	310	9	2790
Uçuş 3	Kapı 24	160	14	2240
Uçuş 4	Kapı 68	279	12	3348
Uçuş 5	Kapı 50	153	9	1377
Uçuş 6	Kapı 36	147	11	1617
Uçuş 7	Kapı 24	159	14	2226
Uçuş 8	Kapı 7	92	16	1472
Uçuş 9	Kapı 29	101	9	909
Uçuş 10	Kapı 86	344	15	5160
Uçuş 11	Kapı 45	135	13	1755
Uçuş 12	Kapı 102	132	17	2244
Uçuş 13	Kapı 4	162	12	1944
Uçuş 14	Kapı 13	132	13	1716
Uçuş 15	Kapı 26	158	10	1580
Uçuş 16	Kapı 63	141	13	1833
Uçuş 17	Kapı 68	346	12	4152
Uçuş 18	Kapı 44	255	9	2295
Uçuş 19	Kapı 37	102	16	1632
Uçuş 20	Kapı 40	102	10	1020

```
x = [[0
        1 0 0 0
```