

Knapsack Problem Matematiksel Model - Örnek Uygulama

Karar Değişkenleri:

$$x_i = \begin{cases} 1, & \text{eğer } i. \text{ nesne sırt çantasına alınırsa} \\ 0, & \text{aksi halde} \end{cases}$$
$$i = 1, \dots, 15.$$

Parametreler:

- w_i : i . nesnenin ağırlığı.
- p_i : i . nesnenin değeri.
- C : Sırt çantasının kapasitesi (250).
- N : Nesne sayısı (bu örnekte 15).

Amaç Fonksiyonu:

$$\text{Maximize } \sum_{i=1}^N p_i x_i$$

Toplam değeri maksimize edecek amaç fonksiyonu.

Kısıtlar:

Kapasite kısıtı:

$$\sum_{i=1}^N w_i x_i \leq C$$

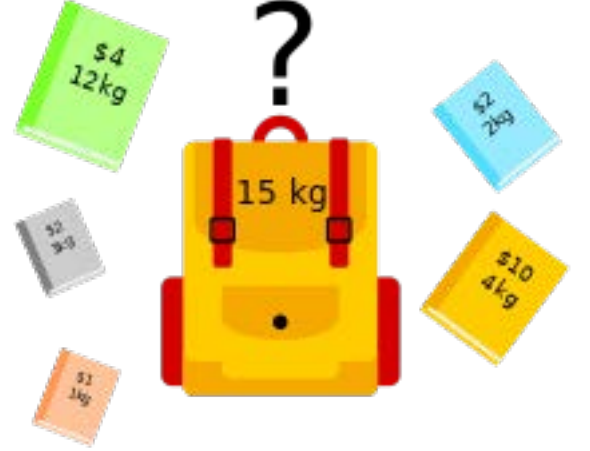
Birlikte olmama kısıtları:

$$x_1 + x_5 \leq 1$$

$$x_4 + x_{12} \leq 1$$

İkili Değişken (Binary) Kısıtları:

$$x_i \in \{0, 1\} \forall i = 1, \dots, N$$



Knapsack Gurobi

Umut Cindiloglu

March 2024

```
[3]: from gurobipy import Model, GRB
```

```
[5]: agirliklar = [20, 30, 35, 40, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100]
degerler = [100, 120, 130, 145, 150, 160, 175, 180, 190, 200, 210, 220, 230, 240, 250]
kapasite = 250

#15 tane eşyanın ağırlığı ve değerleri
```

```
[8]: model = Model("sirt_cantasi") # Bu kod satırı ile modelimizi oluşturuyoruz.
```

```
[9]: x = model.addVars(len(agirliklar), vtype=GRB.BINARY, name="x")

#Nu kod satırı ile modele değişken ekliyoruz.
```

```
[10]: model.setObjective(sum(degerler[i] * x[i] for i in range(len(agirliklar))), GRB.
    ↪MAXIMIZE)

#Matematiksel modelin amacını bu kod satırı ile modele ekliyoruz.
```

```
[11]: # Kısıt
model.addConstr(sum(agirliklar[i] * x[i] for i in range(len(agirliklar))) <= kapasite,
    ↪"kapasite")

#Çantanın kapasite kısıtını aşmamak için bu kısıtı modele ekliyoruz.
```

```
[11]: <gurobi.Constr *Awaiting Model Update*>
```

```
[12]: # Birlikte Olmama Kısıtları
model.addConstr(x[0] + x[4] <= 1, "birlikte_olmama_1_5")
model.addConstr(x[3] + x[11] <= 1, "birlikte_olmama_4_12")

#Bazı eşyalar bir arada olamaz. Birlikte olmama kısıtlarını da modele ekliyoruz.
```

```
[12]: <gurobi.Constr *Awaiting Model Update*>
```

```
[13]: model.optimize() #Modeli çözdüren kod satırı.
```

Gurobi Optimizer version 11.0.1 build v11.0.1rc0 (win64 - Windows 11.0 (22631.2))

CPU model: Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, instruction set [SSE2|AVX|AVX2]

Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 3 rows, 15 columns and 19 nonzeros

Model fingerprint: 0x7c6d0e25

Variable types: 0 continuous, 15 integer (15 binary)

Coefficient statistics:

Matrix range [1e+00, 1e+02]

Objective range [1e+02, 3e+02]

Bounds range [1e+00, 1e+00]

RHS range [1e+00, 3e+02]

Found heuristic solution: objective 830.0000000

Presolve time: 0.02s

Presolved: 3 rows, 15 columns, 19 nonzeros

Variable types: 0 continuous, 15 integer (15 binary)

Found heuristic solution: objective 845.0000000

Root relaxation: objective 8.576923e+02, 3 iterations, 0.00 seconds (0.00 work units)

Nodes		Current Node			Objective Bounds			Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time
0	0	857.69231	0	1	845.00000	857.69231	1.50%	-	0s
*	0	0	0	0	850.0000000	850.00000	0.00%	-	0s

Explored 1 nodes (3 simplex iterations) in 0.07 seconds (0.00 work units)

Thread count was 8 (of 8 available processors)

Solution count 3: 850 845 830

Optimal solution found (tolerance 1.00e-04)

Best objective 8.500000000000e+02, best bound 8.500000000000e+02, gap 0.0000%

```
[18]: toplam_ağırlık = 0
toplam_değer = 0
print("Seçilen Nesnelere:")
for i in range(len(agirliklar)):
    if x[i].X > 0.5: # Eğer nesne seçildiyse
        print(f"Nesne {i+1} (Ağırlık: {agirliklar[i]}, Değer: {degerler[i]})")
        toplam_ağırlık += agirliklar[i]
        toplam_değer += degerler[i]

print(f"Seçilen Nesnelere Toplam Ağırlığı: {toplam_ağırlık}")
print(f"Seçilen Nesnelere Toplam Değeri: {toplam_değer} ")
```

Seçilen Nesnelere:

Nesne 1 (Ağırlık: 20, Değer: 100)

Nesne 2 (Ağırlık: 30, Değer: 120)

Nesne 3 (Ağırlık: 35, Değer: 130)

Nesne 4 (Ağırlık: 40, Değer: 145)

Nesne 7 (Ağırlık: 60, Değer: 175)

Nesne 8 (Ağırlık: 65, Değer: 180)

Seçilen Nesnelere Toplam Ağırlığı: 250

Seçilen Nesnelere Toplam Değeri: 850