

Gurobi Çözücü

March 11, 2024

1 Gurobi ile Matematiksel Model Çözümü

Bu örnek uygulamada, Gurobi çözücüsünü Python programlama dili ile kullanacağız.

```
[ ]: #Eğer gurobipy kütüphanesi yüklü değilse, bu kod satırı ile kütüphaneyi  
     →yükliyoruz  
pip install gurobipy
```

```
[9]: #Bu kod satırı ile gurobipy kütüphanesinden Model ve GRB fonksiyonlarını çağırıyoruz.  
from gurobipy import Model, GRB
```

ÖRNEK UYGULAMA

Diyelim ki bir fabrika, iki tür ürün (Ürün A ve Ürün B) üretmektedir. Her bir ürünün üretimi için belirli miktarda hammadde gerekmekte ve fabrikanın günlük kullanabileceği sınırlı miktarda hammadde bulunmaktadır. Ayrıca, her bir ürünün satışından elde edilen kar farklıdır. Amacımız, verilen hammadde kısıtları altında fabrikanın maksimum toplam kârı nasıl elde edeceğini bulmaktır.

Problem Tanımı:

Ürün A'nın üretimi için 1 birim Hammadde 1 ve 2 birim Hammadde 2 gerekiyor; her bir biriminden 5 birim kar elde ediliyor. Ürün B'nin üretimi için 2 birim Hammadde 1 ve 1 birim Hammadde 2 gerekiyor; her bir biriminden 8 birim kar elde ediliyor. Günlük kullanılabilir Hammadde 1 miktarı 100 birim, Hammadde 2 miktarı ise 160 birimdir. Ürün A'nın günlük üretimi en az 10 birim, en fazla 40 birim olmalıdır. Ürün B için günlük üretim sınırı yoktur ama en az 5 birim üretilmelidir.

Bu basit problem için oluşturduğumuz matematiksel model şu şekilde olabilir;

Karar Değişkenleri;

x_A : Ürün A'nın üretim miktarı (birim)

x_B : Ürün B'nin üretim miktarı (birim)

Amaç Fonksiyonu;

Max $5x_A + 8x_B$

Fonksiyonel Kısıtlar;

$x_A + 2 * x_B \leq 100$, "Hammadde1_kisiti"

$2 * x_A + x_B \leq 160$, "Hammadde2_kisiti"

$10 \leq x_A \leq 40$, "Ürün A için üretim sınırları"

$x_B \geq 5$, "Ürün B için minimum üretim sınırı"

İşaret Kısıtları;

$x_A, x_B \Rightarrow 0$

```
[3]: # Bu kod satırı ile bir adet matematiksel model tanımlıyoruz. Sonrasında bu
      ↪modele sırasıyla karar değişkenlerini,
      amaç fonksiyonunu ve kısıtları ekleyeceğiz.
      model = Model("genisletilmis_uretim_planlama")
```

```
[4]: # Karar değişkenleri
      # Ürün A ve B'nin üretim miktarları (sürekli değişkenler)
      xA = model.addVar(vtype=GRB.CONTINUOUS, name="xA", lb=10, ub=40) #burada xA için
      ↪"lb" lower bound ve "ub" upper bound (alt sınır ve üst sınır) tanımladık
      xB = model.addVar(vtype=GRB.CONTINUOUS, name="xB", lb=5) #xB için de lb
      ↪tanımladık ve bu değişken tiplerinin sürekli (CONTINUOUS) olduğunu belirttik.
```

```
[5]: # Amaç fonksiyonu
      # Toplam kârı maksimize etmek için gerekli amaç fonksiyonunun modele ekledik.
      model.setObjective(5 * xA + 8 * xB, GRB.MAXIMIZE)
```

```
[6]: # Kısıtlar
      # Hammadde 1 ve 2 kullanımı için kısıtları modele ekledik
      model.addConstr(xA + 2 * xB <= 100, "Hammadde1_kisiti")
      model.addConstr(2 * xA + xB <= 160, "Hammadde2_kisiti")
```

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

```
[7]: # Modeli çözdürecek kod sarını çalıştırdık ve sonuçları aldık.
      model.optimize()
```

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 2 rows, 2 columns and 4 nonzeros

Model fingerprint: 0x90dc9583

Coefficient statistics:

Matrix range [1e+00, 2e+00]

Objective range [5e+00, 8e+00]

Bounds range [5e+00, 4e+01]

RHS range [1e+02, 2e+02]

Presolve removed 2 rows and 2 columns

Presolve time: 0.01s

Presolve: All rows and columns removed

Iteration	Objective	Primal Inf.	Dual Inf.	Time
0	4.4000000e+02	0.0000000e+00	0.0000000e+00	0s

Solved in 0 iterations and 0.02 seconds (0.00 work units)
Optimal objective 4.400000000e+02

```
[8]: # Çözümü yazdırmak için gerekli kod satırları.  
print("Optimal Üretim Miktarları:")  
print(f"Ürün A: {xA.X}")  
print(f"Ürün B: {xB.X}")  
print(f"Toplam Kâr: {model.ObjVal}")
```

Optimal Üretim Miktarları:
Ürün A: 40.0
Ürün B: 30.0
Toplam Kâr: 440.0