

2017-2018 Bahar Yarıyılı
Balıkesir Üniversitesi
Endüstri Mühendisliği Bölümü

EMM3208

Optimizasyon Teknikleri

(GAMS - sets, variables, parameters, tables, equations, model, solve)

4

Yrd. Doç. Dr. İbrahim Küçükkoç
<http://ikucukkoc.baun.edu.tr>
ikucukkoc@balikesir.edu.tr

22.02.2018

Bir GAMS Modelinin Temel Bileşenleri

Inputs:

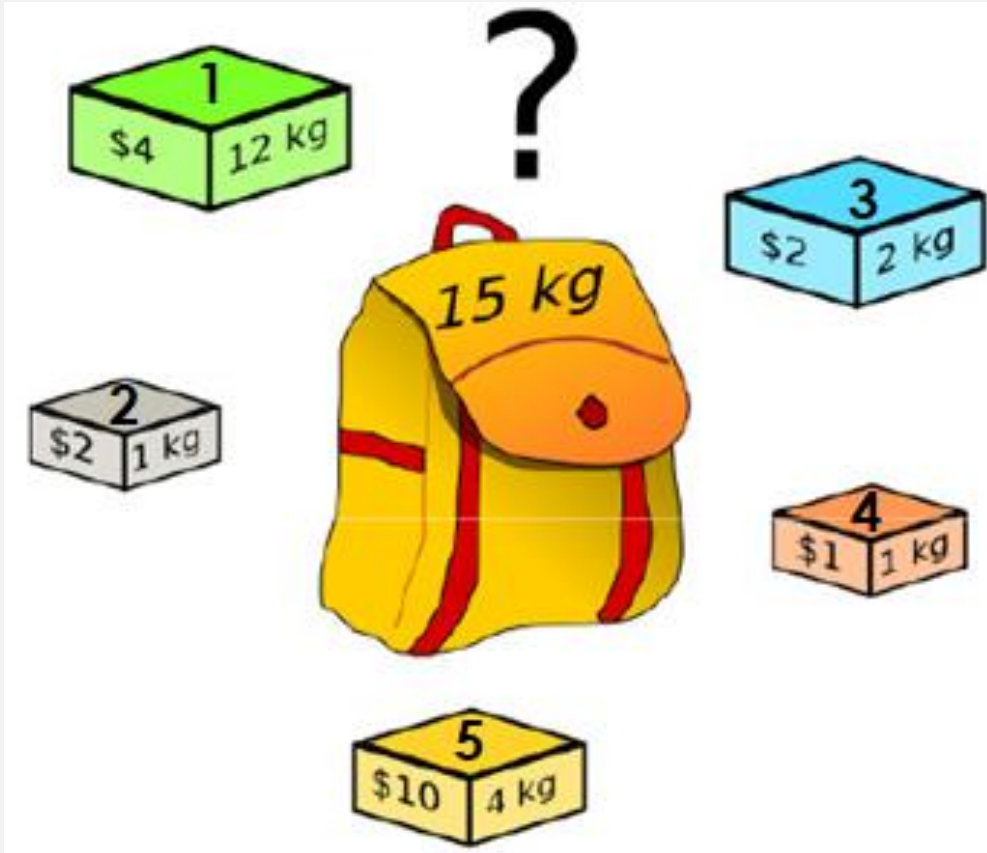
- Sets
 - Declaration
 - Assignment of members
- Data (Parameters, Tables, Scalars)
 - Declaration
 - Assignment of values
- Variables
 - Declaration
 - Assignment of type
- Assignment of bounds and/or initial values (optional)
- Equations
 - Declaration
 - Definition
- Model and Solve statements
- Display statement (optional)

Outputs:

- Echo Print
- Reference Maps
- Equation Listings
- Status Reports
- Results

Sırt Çantası (Knapsack) Problemi Örneği

Sırt çantası (knapsack) problemi, sınırlı olarak verilen kapasite kısıtını sağlarken faydayı maksimize etmeye çalışma problemidir.



n : eşya sayısı ($i = 1, 2, \dots, n$)

w_i : i eşyasının ağırlığı (kg)

v_i : i eşyasının faydası (\$)

W : Çantanın kapasitesi (kg)

X_i : 1, eğer i eşyası alınmışsa; 0, alınmamışsa

$$\max \sum_i v_i * X_i, \quad i = 1, 2, \dots, n$$

Kapasite Kısıtı:
$$\sum_i w_i * X_i \leq W$$

İşaret Kısıtı:
$$X_i \in \{0, 1\}$$

```
set
  i eşya numarası /1*5/
;
* i eşya numarası /1,2,3,4,5/

parameters
  we(i) eşya ağırlığı
  /1 12
   2 1
   3 2
   4 1
   5 4/

  v(i) eşyanın değeri
  /1 4
   2 2
   3 2
   4 1
   5 10/
;

scalar W çanta kapasitesi /15/;
```

```
variables
  X(i) karar değişkeni
  Z amaç fonk
;

binary variables
X(i)
;

equations
  amac
  kisit1
;

amac.. Z =e= sum(i, v(i)*X(i));
kisit1.. sum(i, we(i)*X(i)) =l= W;

model knapsack /all/;
solve knapsack using mip maximizing Z;
```

Sum kullanmadan yapılmak istenirse, amaç fonksiyonu ve kısıt aşağıdaki şekilde de tanımlanabilir:

```
equations
amac
kisit1
;

amac.. Z =e= v("1")*X('1')+v("2")*X('2')+v("3")*X('3')+v("4")*X('4')+v("5")*X('5');
kisit1.. we("1")*X('1')+we("2")*X('2')+we("3")*X('3')+we("4")*X('4')+we("5")*X('5') =1= W;

model knapsack /all/;
solve knapsack using mip maximizing Z;
```

S O L V E S U M M A R Y

MODEL knapsack OBJECTIVE Z
TYPE MIP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 42

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 1 Optimal
**** OBJECTIVE VALUE 15.0000

RESOURCE USAGE, LIMIT 0.056 1000.000
ITERATION COUNT, LIMIT 1 2000000000

Görüleceği üzere X2, X3, X4 ve X5 karar değişkenleri 1 değerini almışlardır.

Bu durumda amaç fonksiyonu değeri \$15 olmuştur.

Çanta ağırlığı da 8 kg'dır.

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	8.000	15.000	.

---- VAR X karar değişkeni

	LOWER	LEVEL	UPPER	MARGINAL
1	.	.	1.000	4.000
2	.	1.000	1.000	2.000
3	.	1.000	1.000	2.000
4	.	1.000	1.000	1.000
5	.	1.000	1.000	10.000

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Z	-INF	15.000	+INF	.

Z amaç fonk

Bütçe Kısıtlı Proje Seçimi

Bir şirket gelecek yıl için sekiz olası proje yatırımını değerlendirmektedir. Her bir projenin beklenen getirisi ve her bir proje için yapılması gereken başlangıç yatırımı tabloda verilmiştir. Bütçeyi (\$1.5 milyon) aşmayarak karı maksimize edecek şekilde hangi projelerin yatırıma dahil edilmesi gerektiğine karar veriniz.

Proje No	Beklenen Getiri	Baslangic Yatirimi
1	\$500.000	\$425.000
2	\$750.000	\$450.000
3	\$1.000.000	\$550.000
4	\$600.000	\$300.000
5	\$500.000	\$150.000
6	\$500.000	\$250.000
7	\$450.000	\$350.000
8	\$500.000	\$325.000

*Bütçe Kısıtlı Proje Seçimi

sets

```
i yatırım numarası /1*8/
;
```

parameters

```
g(i) beklenen getiri
/1 500
 2 750
 3 1000
 4 600
 5 500
 6 500
 7 450
 8 500/
```

y(i) başlangıç yatırımı

```
/1 425
 2 450
 3 550
 4 300
 5 150
 6 250
 7 350
 8 325/
```

```
;
```

```
scalar B bütçe /1500/;
```

variable

```
x(i) eğer i projesine yatırım yapılmışsa 1 değilse 0
z amaç fonk değeri
;
```

binary variable

```
x(i)
;
```

equations

```
amac
kisit bütçe yatırımı
;
```

```
amac.. z=e=sum(i, x(i)*(g(i)-y(i)));
```

```
kisit.. sum(i, x(i)*y(i)) =l= B;
```

```
model butce /all/;
```

```
solve butce using mip maximizing z;
```

S O L V E S U M M A R Y

MODEL butce OBJECTIVE z
 TYPE MIP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 49

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 8 Integer Solution
 **** OBJECTIVE VALUE 1400.0000

MIP Solution: 1400.000000 (1 iterations, 0 nodes)
 Final Solve: 1400.000000 (0 iterations)
 Best possible: 1516.666667
 Absolute gap: 116.666667
 Relative gap: 0.083333

LOWER LEVEL UPPER MARGINAL

---- EQU amac . . . 1.000
 ---- EQU kisit -INF 1450.000 1500.000 .

kisit bütçe yatırımı

---- VAR x eğer i projesine yatırım yapılmışsa 1 değilse 0

LOWER LEVEL UPPER MARGINAL

1	.	.	1.000	75.000
2	.	1.000	1.000	300.000
3	.	1.000	1.000	450.000
4	.	1.000	1.000	300.000
5	.	1.000	1.000	350.000
6	.	.	1.000	250.000
7	.	.	1.000	100.000
8	.	.	1.000	175.000

Probleme aşağıdaki gibi yeni bir kısıt daha ekleyelim:
 - İlk dört projeye yapılabilecek yatırım tutarı \$1milyon'u aşamaz.

*Bütçe Kısıtlı Proje Seçimi

*Yeni Kısıt: İlk dört projeye yapılacak yatırım tutarı 1000 (\$K) 'ı geçemez!

sets

i yatırım numarası /1*8/;

parameters

g(i) beklenen getiri (\$K)

/1 500
 2 750
 3 1000
 4 600
 5 500
 6 500
 7 450
 8 500/

y(i) başlangıç yatırımı (\$K)

/1 425
 2 450
 3 550
 4 300
 5 150
 6 250
 7 350
 8 325/

;

scalar B bütçe (\$K) /1500/;

scalar V ilk dört proje için bütçe /1000/;

variable

x(i) eğer i projesine yatırım yapılmışsa 1 değilse 0

z amaç fonk değeri

;

binary variable

x(i)

;

equations

amac

kisit toplam bütçe kısıtı

kisit2 ilk dört proje için bütçe kısıtı

;

amac.. z=e=sum(i, x(i)*(g(i)-y(i)));

kisit.. sum(i, x(i)*y(i)) =l= B;

*kisit2.. x('1')*y('1')+ x('2')*y('2')+x('3')*y('3')+x('4')*y('4') =l= V;

kisit2.. sum(i\$(ord(i)<=4), x(i)*y(i)) =l= V;

model butce /all/;

solve butce using mip maximizing z;

S O L V E S U M M A R Y

MODEL butce OBJECTIVE z
TYPE MIP DIRECTION MAXIMIZE
SOLVER CPLEX FROM LINE 52

**** SOLVER STATUS 1 Normal Completion
**** MODEL STATUS 8 Integer Solution
**** OBJECTIVE VALUE 1350.0000

MIP Solution: 1350.000000 (2 iterations, 0 nodes)
Final Solve: 1350.000000 (0 iterations)

Best possible: 1484.615385
Absolute gap: 134.615385
Relative gap: 0.099715

LOWER LEVEL UPPER MARGINAL

---- EQU amac . . . 1.000
---- EQU kisit -INF 1250.000 1500.000 .
---- EQU kisit2 -INF 850.000 1000.000 .

kisit toplam bütçe kısıtı
kisit2 ilk dört proje için bütçe kısıtı

---- VAR x eğer i projesine yatırım yapılmışsa 1 değilse 0

LOWER LEVEL UPPER MARGINAL

1	.	.	1.000	75.000
2	.	.	1.000	300.000
3	.	1.000	1.000	450.000
4	.	1.000	1.000	300.000
5	.	1.000	1.000	350.000
6	.	1.000	1.000	250.000
7	.	.	1.000	100.000
8	.	.	1.000	175.000

Ahşap Üretim Problemi

Dört tip ahşap panelin (A, B, C, D) üretildiği bir sistemi ele alalım. Her ahşap panel, çam ve meşe talaşlarının farklı miktarlarda karıştırılıp yapıştırılması ve preslenmesi sonucu oluşmaktadır.

Aşağıdaki tablo, değişik panel tipleri için bir palet panel üretiminde kullanılması gereken kaynak miktarlarını göstermektedir.

Her Bir Palet İçin Gerekli Kaynak Miktarı				
	A	B	C	D
Yapıştırıcı (litre) – $y(i)$	50	50	100	50
Sıkıştırma (saat) – $s(i)$	5	15	10	5
Çam Talaşı (kg) – $c(i)$	500	400	300	200
Meşe Talaşı (kg) – $m(i)$	500	750	250	500

Eldeki (mevcut) kaynak miktarları şu şekildedir:

5800 litre yapıştırıcı (Y)

730 saat sıkıştırma kapasitesi (S)

29200 kg çam talaşı (C)

60500 kg meşe talaşı (M)

Her bir palet A, B, C ve D paneli sırasıyla \$450, \$1150, \$800 ve \$400 dolara satılmaktadır.

Firma karını maksimize etmek için her bir panel tipinden kaç palet üretmelidir?

Öncelikle problem matematiksel model olarak ifade edilmelidir.

Karar değişkenleri A, B, C ve D modellerinin üretim miktarlarıdır. Bunlar sırasıyla X_1 , X_2 , X_3 ve X_4 olarak düşünülürse amaç fonksiyonu aşağıdaki şekilde yazılabilir:

$$\text{Max } Z = 450X_1 + 1150X_2 + 800X_3 + 400X_4$$

Kaynak kullanımlarına ilişkin kısıtlar aşağıdaki şekilde ifade edilebilir:

$$50 X_1 + 50 X_2 + 100 X_3 + 50 X_4 \leq 5800 \text{ (yapıştırıcı)}$$

$$5 X_1 + 15 X_2 + 10 X_3 + 5 X_4 \leq 730 \text{ (sıkıştırma)}$$

$$500 X_1 + 400 X_2 + 300 X_3 + 200 X_4 \leq 29200 \text{ (çam)}$$

$$500 X_1 + 750 X_2 + 250 X_3 + 500 X_4 \leq 60500 \text{ (meşe)}$$

Herhangi bir panelin üretim miktarı 0'dan küçük olamayacağı için:

$$X_1, X_2, X_3, X_4 \geq 0$$

variables

x1, x2, x3, x4
z;

positive variables

x1, x2, x3, x4;

equations

amac

kisit1

kisit2

kisit3

kisit4

;

amac.. z =e= 450*x1+1150*x2+800*x3+400*x4;

kisit1.. 50*x1 + 50*x2 + 100*x3 + 50*x4 =l= 5800;

kisit2.. 5*x1 + 15*x2 + 10*x3 + 5*x4 =l= 730;

kisit3.. 500*x1 + 400*x2 + 300*x3 + 200*x4 =l= 29200;

kisit4.. 500*x1 + 750*x2 + 250*x3 + 500*x4 =l= 60500;

model ornek3 /all/;

solve ornek3 using lp maximizing z;

S O L V E S U M M A R Y

MODEL ornek3 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 23

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 58800.0000

RESOURCE USAGE, LIMIT 0.005 1000.000
 ITERATION COUNT, LIMIT 4 2000000000

LP status(1): optimal
 Optimal solution found.
 Objective : 58800.000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	5800.000	5800.000	0.429
---- EQU kisit2	-INF	730.000	730.000	71.429
---- EQU kisit3	-INF	29200.000	29200.000	0.143
---- EQU kisit4	-INF	32500.000	60500.000	.

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR x1	.	23.000	+INF	.
---- VAR x2	.	15.000	+INF	.
---- VAR x3	.	39.000	+INF	.
---- VAR x4	.	.	+INF	-7.143
---- VAR z	-INF	58800.000	+INF	.

Modeli bu şekilde açık olarak yazmak doğru olmasına rağmen, bu yaklaşım kullanıldığında büyük kapsamlı modellerin yazılmasında problemler çıkabilir. Bu sebeple yazım üstünde bazı değişiklikler yapılabilir.

Modelde verilen dört karar değişkeninin ayrı ayrı tanımlanmasına gerek yoktur. Bunun yerine bir x değişkeni tanımlanıp, dört ayrı **indis** atanabilir.

GAMS modellerinde indislerin “**set**” komutu ile tanımlanabileceği gösterilmişti.

set i /1, 2, 3, 4/

Bu komut aynı sonucu verecek şekilde aşağıdaki gibi de yazılabilir:

set i /1*4/

Ayrıca parametreler ve scalar değerler de kullanılabilir.

Bu model kapalı formda aşağıdaki şekilde yazılabilir:

Amaç Fonksiyonu:

$$\max Z = \sum_{i \in I} p_i X_i$$

Kısıtlar:

$$\sum_{i \in I} y_i X_i \leq Y$$

$$\sum_{i \in I} s_i X_i \leq S$$

$$\sum_{i \in I} c_i X_i \leq C$$

$$\sum_{i \in I} m_i X_i \leq M$$

$$X_i \geq 0, \quad \forall i \in I$$

Notasyon:

I : Ürün kümesi (A, B, C, D)

p_i : Bir palet i ürünü satışından elde edilecek gelir

y_i : Bir palet i ürünü için gerekli yapıştırıcı miktarı

s_i : Bir palet i ürünü için gerekli sıkıştırma kapasitesi kullanım miktarı

c_i : Bir palet i ürünü için gerekli çam talaşı miktarı

m_i : Bir palet i ürünü için gerekli meşe talaşı miktarı

X_i : i ürününden üretilecek miktar (palet)

Y : Mevcut yapıştırıcı kapasitesi

S : Mevcut sıkıştırma kapasitesi

C : Mevcut çam talaşı kapasitesi

M : Mevcut meşe talaşı kapasitesi

```

sets
i /1*4/

variables
x(i)
z;

positive variables
x(i);

parameters
y(i) yapıştırıcı
/ 1 50
  2 50
  3 100
  4 50/

s(i) yapıştırıcı
/ 1 5
  2 15
  3 10
  4 5/

c(i) yapıştırıcı
/ 1 500
  2 400
  3 300
  4 200/

m(i) yapıştırıcı
/ 1 500
  2 750
  3 250
  4 500/

```

```

p(i) yapıştırıcı
/ 1 450
  2 1150
  3 800
  4 400/
;

scalars
YK /5800/
SK /730/
CK /29200/
MK /60500/
;

equations
amac
kisit1
kisit2
kisit3
kisit4
;

amac.. z =e= sum(i, p(i)*x(i));
kisit1.. sum(i, y(i)*x(i)) =l= YK;
kisit2.. sum(i, s(i)*x(i)) =l= SK;
kisit3.. sum(i, c(i)*x(i)) =l= CK;
kisit4.. sum(i, m(i)*x(i)) =l= MK;

model ornek3 /all/;
solve ornek3 using lp maximizing z;

```

S O L V E S U M M A R Y

MODEL ornek3 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 65

LP status(1): optimal
 Optimal solution found.
 Objective : 58800.000000

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 58800.0000

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 4 20000000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	5800.000	5800.000	0.429
---- EQU kisit2	-INF	730.000	730.000	71.429
---- EQU kisit3	-INF	29200.000	29200.000	0.143
---- EQU kisit4	-INF	32500.000	60500.000	.

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	.	23.000	+INF	.
2	.	15.000	+INF	.
3	.	39.000	+INF	.
4	.	.	+INF	-7.143

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR z	-INF	58800.000	+INF	.

Ulaştırma Problemi

İyi bilinen bir taşıma (ulaştırma) probleminde, belirli arz noktalarından, belirli talep noktalarına taşınacak ürünlerin minimum maliyetle taşınması istenmektedir. Şehirler arası uzaklıklar bilinmektedir. Problem, taşıma maliyetini minimize etmek amacıyla, herhangi bir arz noktası-talep noktası arasında ne kadar malzeme taşınacağına belirlenmesi sürecidir.

Indices:

i = plants

j = markets

Given Data:

a_i = supply of commodity of plant i (in cases)

b_j = demand for commodity at market j

c_{ij} = cost per unit shipment between plant i and market j (\$/case)

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j (cases),

where $x_{ij} \geq 0$, for all i, j

Constraints:

Observe supply limit at plant i : $\sum_j x_{ij} \leq a_i$ for all i (cases)

Satisfy demand at market j : $\sum_i x_{ij} \geq b_j$ for all j (cases)

Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ (\$K)

GAMS terminolojisiinde,

İndisler (indices) **sets** olarak

Verilen bilgiler, yani problem girdileri, **parameters** olarak

Karar değişkenleri, **variables** olarak

Kısıtlar ve amaç fonksiyonu da **equations** olarak değerlendirilir.

Problemin parametrelerini oluşturacak olan, şehirler arası uzaklıklar, aşağıdaki tabloda verilmiştir.

<i>Plants</i>	<i>Shipping Distances to Markets (1000 miles)</i>			<i>Supplies</i>
	New York	Chicago	Topeka	
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
<i>Demands</i>	325	300	275	

```

Sets
  i  canning plants    / seattle, san-diego /
  j  markets           / new-york, chicago, topeka / ;

Parameters
  a(i)  capacity of plant i in cases
        /    seattle    350
          san-diego    600  /

  b(j)  demand at market j in cases
        /    new-york    325
          chicago      300
          topeka       275  / ;

Table d(i,j)  distance in thousands of miles
           new-york    chicago    topeka
seattle      2.5        1.7        1.8
san-diego     2.5        1.8        1.4  ;

Scalar f  freight in dollars per case per thousand miles  /90/ ;

Parameter c(i,j)  transport cost in thousands of dollars per case ;

          c(i,j) = f * d(i,j) / 1000 ;

Variables
  x(i,j)  shipment quantities in cases
  z       total transportation costs in thousands of dollars ;

```

```

Positive Variable x ;

```

```

Equations

```

```

    cost          define objective function
    supply(i)     observe supply limit at plant i
    demand(j)     satisfy demand at market j ;

```

```

cost ..          z  =e=  sum((i,j), c(i,j)*x(i,j)) ;

```

```

supply(i) ..     sum(j, x(i,j))  =l=  a(i) ;

```

```

demand(j) ..     sum(i, x(i,j))  =g=  b(j) ;

```

```

Model transport /all/ ;

```

```

Solve transport using lp minimizing z ;

```

```

Display x.l, x.m ;

```

Problem yukarıda verildiği gibi modellenir ve çözülürse aşağıdaki sonuç elde edilecektir:

	new-york	chicago	topeka
seattle	50.000	300.000	
san-diego	275.000		275.000

Burada görüldüğü üzere, optimum çözümde Seattle şehrinden Topeka'ya ürün gönderilmemektedir.

Fakat, Seattle'dan Topeka'ya bir birim ürün gönderilmek isterse, bunun amaç fonksiyona yapacağı etkinin 0.036 (\$K) veya \$36 olacağı, aşağıdaki marjinal maliyetlerden görülebilir.

	chicago	topeka
seattle		0.036
san-diego	0.009	

Sets

Örneğimizde, indisler sets olarak aşağıdaki gibi tanımlanmıştır:

```
Sets
  i  canning plants  / seattle, san-diego /
  j  markets         / new-york, chicago, topeka / ;
```

Bu örnekte setleri birleştirmeden de kullanabilirdik:

```
Set   i  canning plants  / seattle, san-diego / ;
Set   j  markets         / new-york, chicago, topeka / ;
```

Set tanımlanırken * kullanılırsa, ilgili set aritmetik olarak ardışık artan şekilde tanımlanır:

```
Set   t  time periods    /1991*2000/;
Set   m  machines        /mach1*mach24/;
```

Bu tanımlama ile aşağıdaki gibi iki set elde edilebilir.

$$\begin{aligned} t &= \{1991, 1992, 1993, \dots, 2000\} \\ m &= \{mach_1, mach_2, \dots, mach_{24}\}. \end{aligned}$$

Not:

Bir değişken veya parametre, bir kümenin kendisiyle kartezyen çarpımından oluşan kümeyle ilişkili ise, aynı kümeye ait iki farklı indis kullanabilmek için “Alias” özelliğini kullanabiliriz.

```
sets  
Seh şehirler /a, b, c/  
;  
Alias(Seh, s1, s2);  
Positive variable x(s1, s2)
```

Not:

Açıklama yazmak için açıklamanın başına * işareti koyun. GAMS bu işaretin devamındaki hiçbir şeyi dikkate almayacaktır.

GAMS'te üç farklı formatta veri girmek mümkündür:

- Liste ile
- Tablo ile
- Doğrudan atama ile

Ayrıca sabit değerler scalar olarak girilir.

Liste ile veri girişi örnekleri:

Parameters

```
a(i)  capacity of plant i in cases
/      seattle      350
      san-diego     600  /
```

```
b(j)  demand at market j in cases
/      new-york     325
      chicago       300
      topeka        275  / ;
```

```
Parameters a(i)  capacity of plant i in cases
/ seattle      350
  san-diego    600  / ;
```

```
Parameters b(j)  demand at market j in cases
/ new-york     325
  chicago      300
  topeka       275  / ;
```

Tablo Kullanımı:

GAMS tablolar kullanarak da kolayca veri girişini mümkün kılar. Örneğin:

```
Table d(i,j)  distance in thousands of miles
              new-york    chicago    topeka
seattle       2.5         1.7         1.8
san-diego     2.5         1.8         1.4 ;
```

Burada, i ve j nin kartezyen çarpımlarından oluşan d matrisi için değerler tanımlanmaktadır.

Doğrudan Atama:

```
Parameter c(i,j)  transport cost in thousands of dollars per case ;  
                c(i,j) = f * d(i,j) / 1000 ;
```

Burada, ilk satırın sonundaki noktalı virgül kullanımına dikkat edilmelidir.

Burada, c matrisindeki bazı öğelerin değeri de doğrudan atama yoluyla belirlenebilir.

Örneğin;

```
c('Seattle','New-York') = 0.40;
```

GAMS'te kullanılabilecek bazı doğrudan atama örnekleri:

```
csquared      = sqr(c);  
e             = m*csquared;  
w            = 1/lamda;  
eoq(i)       = sqrt( 2*demand(i)*ordcost(i)/holdcost(i));  
t(i)         = min(p(i), q(i)/r(i), log(s(i)));  
euclidean(i,j) = qrt(sqr(xi(i) - xi(j) + sqr(x2(i) - x2(j)));  
present(j)   = future(j)*exp(-interest*time(j));
```

Variables

GAMS'te kullanılacak karar değişkenleri önceden tanımlanmalıdır.

```
Variables  
    x(i,j)  shipment quantities in cases  
    z       total transportation costs in thousands of dollars ;
```

Not: Amaç fonkziyonu değerini tutacak olan değişken (z) de tanımlanmalıdır. Burada z için bir domain (indis belirteci) bulunmamaktadır, çünkü z scalar bir değerdir.

Tanımlandığı zaman her değişkenin bir tipi olmalıdır. GAMS'te izin verilen değişken tipleri aşağıdaki gibidir:

<i>Variable Type</i>	<i>Allowed Range of Variable</i>
free(default)	$-\infty$ to $+\infty$
positive	0 to $+\infty$
negative	$-\infty$ to 0
binary	0 or 1
integer	0, 1, ..., 100 (default)

```
Positive variable x ;
```

Equations

GAS'te denklemler farklı bölümlerde declare edilir ve tanımlanırlar.

```
Equations
    cost          define objective function
    supply(i)     observe supply limit at plant i
    demand(j)     satisfy demand at market j ;
```

Burada Eqation kelimesi hem eşitlik, hem de eşitsizlik durumları için kullanılır.

Toplama Formülü:

Denklemler konusunda daha detaya nmeden, toplama formülü anlatılmalıdır. GAMS'te toplama formülü **Sum** belirteci ile ifade edilir. Taşıma probleminden bir örnek vermek gerekirse:

```
Sum(j, x(i,j))
```

Bu komut, $\sum_j x_{ij}$ 'ye karşılık gelmektedir.



Benzer şekilde;

`Sum((i,j), c(i,j)*x(i,j))` formülü

aşağıdaki ifadeye karşılık gelmektedir:

$$\sum_i \sum_j c_{ij} x_{ij}$$

Bu formül aşağıdaki gibi iç içe şekilde de yazılabilirdi:

`Sum(i, Sum(j, c(i,j)*x(i,j)))`



Çarpım Formülü:

GAMS'te çarpım formülü prod komutu ile gerçekleştirilir. Örneğin;

```
prod(j, x(i, j))
```

Komutu aşağıdaki formüle karşılık gelmektedir:

$$\prod_j x_{ij}$$

NOT: Toplam ve çarpım komutları, scalar değerleri girerken de kullanılabilir. Örneğin:

```
scalar totsapply    total supply over all plants;  
totsapply = sum(i, a(i));
```

Π

Equation Tanımlamaları:

Eşitlik/eşitsizlik tanımlamaları GAMS'te en karmaşık kısımdır. Çünkü fazla sayıda denklem türü bulunmaktadır. Ulaştırma problem örneğindeki denklemler aşağıdaki şekilde tanımlanabilir:

```
cost ..      z  =e=  sum((i,j), c(i,j)*x(i,j)) ;

supply(i) ..  sum(j, x(i,j))  =l=  a(i) ;

demand(j) ..  sum(i, x(i,j))  =g=  b(j) ;
```

Burada set kullanımından kaynaklanan büyük kolaylık vardır. Örneğin demand(j) kısıtı, j domain için aşağıdaki açık formda yazılan denklemlere karşılık gelmektedir:

```
DEMAND(new-york)..X(seattle,new-york) + X(san-diego,new-york)=G=325 ;
DEMAND(chicago).. X(seattle,chicago) + X(san-diego,chicago)  =G=300 ;
DEMAND(topeka)..  X(seattle,topeka) + X(san-diego,topeka)      =G=275 ;
```

Dolayısıyla, “set” kullanımı özellikle fazla sayıda değişkenden oluşan gerçek hayat problemlerinde büyük kolaylık sağlamaktadır.

İlişkisel operatorler daha önce de ifade edildiği gibi aşağıda verilmektedir:

<code>=l=</code>	less than or equal to
<code>=g=</code>	greater than or equal to
<code>=e=</code>	equal to

Burada dikkat edilmesi gereken husus “=” ile “=e=” kullanımı arasındaki farktır.

“=” operatörü doğrudan atamalarda kullanılır. Örneğin bir scalar ifadenin değerini atamak için kullanılır.

Fakat “=e=” ifadesi denklemlerin tanımlamasında eşittir anlamında kullanılır.

NOT: Denklemler model içerisinde herhangi bir yerde tanımlanabilir. Fakat kullanılacak tüm değişkenler daha önceden tanımlanmış olmalıdır.

Denklemler GAMS tarafından otomatik olarak en sade şekline dönüştürülürler.

Amaç Fonksiyonu (Objective Function)

GAMS'te objective olarak tanımlanan ayrıca bir ifade yoktur.

Bir amaç tanımlamak için, öncelikle scalar değerli ve free-form'da bir değişken tanımlanmalıdır.

Amaç fonksiyonu değerini tutacak değişken indis (domain) içeremez.

Ulaştırma probleminden örnek verecek olursak:

```
Variables  
      z      total transportation costs in thousands of dollars ;
```

...

```
cost ..      z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
```

Model Komutu

model komutu oldukça kesin bir anlama sahiptir. Çözülecek modeli isimlendirmek ve modelin kapsamını belirlemek için kullanılır.

Örneğin;

```
model transport /all/ ;
```

Komutu ile, **transport** olarak isimlendirilen model, **/all/** belirteci ile tüm denklemleri içerecek şekilde oluşturulur.

Başka bir ifade ile aynı model;

```
model transport / cost, supply, demand / ;
```

komutu ile de oluşturulabilir.

Bu özellik, ileri düzey kullanıcılar için tek çalıştırmada birden fazla modelin çalıştırılabilmesi açısından oldukça faydalıdır.

Solve Komutu

Model oluşturulduktan sonra `solve` komutu çağrılarak çalıştırılır. Örneğimizdeki kullanım şekli:

```
solve transport using lp minimizing z ;
```

`solve` komutunu model ismi izler ve `using` anahtar kelimesinden sonra model türü belirtilir.

Ardından çözücünün yönü `maximizing` veya `minimizing` belirteçleri ile belirtildikten sonra en iyilenecek değişkenin ismi yazılır (z).

lp	for linear programming
qcp	for quadratic constraint programming
nlp	for nonlinear programming
dnlp	for nonlinear programming with discontinuous derivatives
mip	for mixed integer programming
rmip	for relaxed mixed integer programming
miqcp	for mixed integer quadratic constraint programming

rmiqcp	for relaxed mixed integer quadratic constraint programming
minlp	for mixed integer nonlinear programming
rminlp	for relaxed mixed integer nonlinear programming
mcp	for mixed complementarity problems
mpec	for mathematical programs with equilibrium constraints
rmpec	for relaxed mathematical program with equilibrium constraints
cns	for constrained nonlinear systems
emp	for extended mathematical programming

Display Komutu

Solve komutu ile model çalıştırıldıktan sonra elde edilen sonuçlar rapor dosyasında gösterilmektedir. Bunun yanında bazı değişkenler için elde edilen sonuçların ayrıca (özel olarak) gösterilmesi istenebilir. Örneğin;

```
display x.l, x.m ;
```

komutu $x(i,j)$ değişkeninin optimum sonuç için elde edilen değerlerini ve $x(i,j)$ için indirgenmiş maliyetlerini rapor edecektir.

.lo, .l, .up, .m veritabanı:

.lo	lower bound
.l	level or primal value
.up	upper bound
.m	marginal or dual value

Ulaştırma Problemine Ait Tam Model

```
Sets
i canning plants / seattle, san-diego /
j markets          / new-york, chicago, topeka / ;
```

Parameters

```
a(i) capacity of plant i in cases
    / seattle   350
      san-diego 600 /

b(j) demand at market j in cases
    / new-york 325
      chicago  300
      topeka   275 / ;
```

Table d(i,j) distance in thousands of miles

	new-york	chicago	topeka
seattle	2.5	1.7	1.8
san-diego	2.5	1.8	1.4

Scalar f freight in dollars per case per thousand miles /90/ ;

```
Parameter c(i,j) transport cost in thousands of dollars per case ;
c(i,j) = f * d(i,j) / 1000 ;
```

Parameter c(i,j) transport cost in thousands of dollars per case ;
 $c(i,j) = f * d(i,j) / 1000$;

Variables

```
x(i,j) shipment quantities in cases
z      total transportation costs in thousands of dollars ;
```

Positive Variable x;

Equations

```
cost      define objective function
supply(i) observe supply limit at plant i
demand(j) satisfy demand at market j ;
```

```
cost ..      z =e= sum((i,j), c(i,j)*x(i,j)) ;
supply(i) .. sum(j, x(i,j)) =l= a(i) ;
demand(j) .. sum(i, x(i,j)) =g= b(j) ;
```

Model transport /all/ ;

Solve transport using lp minimizing z ;

Display x.l, x.m ;

S O L V E S U M M A R Y

MODEL transport OBJECTIVE z
 TYPE LP DIRECTION MINIMIZE
 SOLVER CPLEX FROM LINE 42

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 153.6750

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 4 2000000000

IBM ILOG CPLEX Jul 4, 2010 23.5.1 WIN 18414.18495 VS8 x86/MS Windows
 Cplex 12.2.0.0, GAMS Link 34

LP status(1): optimal
 Optimal solution found.
 Objective : 153.675000

---- 44 VARIABLE x.L shipment quantities in cases

	new-york	chicago	topeka
seattle	50.000	300.000	
san-diego	275.000		275.000

---- 44 VARIABLE x.M shipment quantities in cases

	chicago	topeka
seattle		0.036
san-diego	0.009	

ATK Ulastirma Problemi

ATK- Beyaz dört müşterisinin taleplerini karşılamak için üç potansiyel fabrikayı değerlendirmektedir. Her hangi bir fabrikayı açarsa firma 200000TL'lik yatırım yapması gerekmektedir.

- Bir fabrikada üretilip, bir müşteriye gönderme birim maliyetleri tabloda verilmiştir.
- Firmanın talepleri en düşük maliyetle karşılaması için gerekli TP modelini kurunuz.

	Müşteri 1	Müşteri 2	Müşteri 3	Müşteri 4	Kapasite
Fabrika 1	8TL	6TL	10TL	9TL	60,000
Fabrika 2	9TL	12TL	13TL	7TL	55,000
Fabrika 3	14TL	9TL	16TL	5TL	45,000
Talepler	40,000	30,000	25,000	15,000	

Matematiksel Model

► Karar değişkenleri

► x_{ij} : i fabrikasından j müşterisine gönderilen miktar ($i = 1,2,3; j = 1,2,3,4$)

► y_i : i fabrikasının açılması ($i = 1,2,3$)

$$y_i = \begin{cases} 1 & i \text{ fabrikası açılırsa} \\ 0 & \text{açılmazsa} \end{cases}$$

► TP modeli

$$\text{Min } \sum_i \sum_j c_{ij} x_{ij} + 200000 \sum_i y_i$$

Öyle ki; $\sum_j x_{ij} \leq k_i y_i \quad (i=1,2,\dots,m)$ kapasite kısıtları

$$\sum_i x_{ij} \geq t_j \quad (j=1,2,\dots,n) \quad \text{Talep kısıtları}$$

$$x_{ij} \geq 0, y_i \in \{0, 1\}$$

```
sets i fabrikalar /1, 2, 3/
      j müşteriler /1*4/ ;
```

```
table c(i,j)
      1   2   3   4
1     8   6  10   9
2     9  12  13   7
3    14   9  16   5 ;
```

```
Scalar fa fabrika acma maliyeti /200000/;
```

```
parameter talep(j) /1 40000, 2 30000, 3 25000, 4 15000 /;
```

```
parameter kapasite(i) / 1 60000, 2 55000, 3 45000 /;
```

```
positive variables x(i,j);
```

```
binary variables y(i);
```

```
variable z;
```

```
equations amac, kapasitekisiti, talepkisiti ;
```

```
    amac..                z =e= sum((i,j), x(i,j)*c(i,j)) + fa*sum(i, y(i));
```

```
    kapasitekisiti(i).. sum(j, x(i,j)) =l= kapasite(i)*y(i);
```

```
    talepkisiti(j)..      sum(i, x(i,j)) =g= talep(j);
```

```
model ulastirma /all/;
```

```
solve ulastirma using MIP minimizing z;
```

```
display x.l, y.l;
```

S O L V E S U M M A R Y

MODEL ulastirma OBJECTIVE z
 TYPE MIP DIRECTION MINIMIZE
 SOLVER CPLEX FROM LINE 22

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 8 Integer Solution
 **** OBJECTIVE VALUE 1290000.0000

RESOURCE USAGE, LIMIT 0.237 1000.000
 ITERATION COUNT, LIMIT 13 20000000000

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1.1	.	5000.000	+INF	.
1.2	.	30000.000	+INF	.
1.3	.	25000.000	+INF	.
1.4	.	.	+INF	3.000
2.1	.	35000.000	+INF	.
2.2	.	.	+INF	5.000
2.3	.	.	+INF	2.000
2.4	.	15000.000	+INF	.
3.1	.	.	+INF	7.000
3.2	.	.	+INF	4.000
3.3	.	.	+INF	7.000
3.4	.	.	+INF	.

---- VAR y

	LOWER	LEVEL	UPPER	MARGINAL
1	.	1.000	1.000	1.4000E+5
2	.	1.000	1.000	2.0000E+5
3	.	.	1.000	1.1000E+5

	LOWER	LEVEL	UPPER	MARGINAL
--	-------	-------	-------	----------

---- VAR z -INF 1.2900E+6 +INF .

Elektronik Ürün Üretimi Problemi

Firmanız, LCD TV, Muzik Calar ve Hoparlor uretimi yapmaktadır. Bu urunlerin uretiminde, Govde, LCD Ekran, Hoparlor, Guc unitesi ve Cesitli Elektronik Parcalari "ortak" olarak kullanılmaktadır. Kullanilacak parcalarin tedarigi sınırlı iken karı maksimize edecek optimum uretim miktarlarini bulunuz.

Part Requirements by Product				Malzeme Stok Miktarlari
Parca Adi	LCD TV	Muzik Calar	Hoparlor	Mevcut
Govde	1	1	0	450
LCDEkran	1	0	0	250
Hoparlor	2	2	1	800
Guc Unitesi	1	1	0	450
Elektronik Parca	2	1	1	600
Birim Kar	\$75	\$50	\$35	

```

sets
i /1*3/

variables
x(i)
z;

positive variables
x(i);

parameters
Govde(i) Govde
/ 1 1
  2 1
  3 0/

L(i) LCD
/ 1 1
  2 0
  3 0/

H(i) Hoparlör
/ 1 2
  2 2
  3 1/

G(i) güç
/ 1 1
  2 1
  3 0/

E(i) elektronik
/ 1 2
  2 1
  3 1/

```

```

p(i) satış fiyat
/ 1 75
  2 50
  3 35/
;

scalars
GovdeK govde stoğu /450/
LK LCD stoğu /250/
HK hoparlör stoğu /800/
GK güç ünitesi /450/
EK elektronik stoğu /600/
;

equations
amac
kisit1
kisit2
kisit3
kisit4
kisit5
;

amac.. z =e= sum(i, p(i)*x(i));
kisit1.. sum(i, govde(i)*x(i)) =l= GovdeK;
kisit2.. sum(i, L(i)*x(i)) =l= LK;
kisit3.. sum(i, H(i)*x(i)) =l= HK;
kisit4.. sum(i, G(i)*x(i)) =l= GK;
kisit5.. sum(i, E(i)*x(i)) =l= EK;

model ornek5 /all/;
solve ornek5 using lp maximizing z;

```

S O L V E S U M M A R Y

MODEL ornek5 OBJECTIVE z
 TYPE LP DIRECTION MAXIMIZE
 SOLVER CPLEX FROM LINE 68

**** SOLVER STATUS 1 Normal Completion
 **** MODEL STATUS 1 Optimal
 **** OBJECTIVE VALUE 25000.0000

RESOURCE USAGE, LIMIT 0.003 1000.000
 ITERATION COUNT, LIMIT 3 20000000000

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU amac	.	.	.	1.000
---- EQU kisit1	-INF	400.000	450.000	.
---- EQU kisit2	-INF	200.000	250.000	.
---- EQU kisit3	-INF	800.000	800.000	12.500
---- EQU kisit4	-INF	400.000	450.000	.
---- EQU kisit5	-INF	600.000	600.000	25.000

---- VAR x

	LOWER	LEVEL	UPPER	MARGINAL
1	.	200.000	+INF	.
2	.	200.000	+INF	.
3	.	.	+INF	-2.500

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR z	-INF	25000.000	+INF	.

Kaynaklar

Gams – A User’s Guide, Tutorial by Richard E. Rosenthal, Gams Corporation, USA, 2007.

Yöneylem Araştırması Gams Uygulama Ders Notları, Berna Tekraş Sivrikaya, İstanbul Teknik Üniversitesi, İşletme Fakültesi, 2010.

Yöneylem Araştırması – 2 Ders Notları, Özgür Kabak, İstanbul Teknik Üniversitesi, İşletme Fakültesi.

Yöneylem Araştırması – 1 Paket Programlar, Sultan Ceren Salkın, İstanbul Teknik Üniversitesi, İşletme Fakültesi.

Kucukkoc, I., Doğrusal Programlamada Karışım Problemleri,
<http://w3.balikesir.edu.tr/~ikucukkoc/dokumanlar/mixture.pdf>, Erişim Tarihi: 15 Ocak 2018.

Sarker, R. A., Newton, C. S., Optimization Modelling: A Practical Approach, CRC Press Book, 2008.

Seçme N.Y., Klasik Doğrusal Programlama Ve Bulanık Doğrusal Programlamanın Karşılaştırmalı Bir Analizi: Üretim Planlama Örneği, Kayseri, 2005.

Ulucan A., Yöneylem Araştırması, İşletmecilik Uygulamalı Bilgisayar destekli Modelleme, Ankara, 2007.