

**OPERATION RESEARCH 2**

**Industrial Engineering  
Sharif University of Technology**

**Prof. Mohammad Modarres**

# **General Algebraic Modeling System**

(GAMS)

**Arman Jabbari**

---

# GAMS

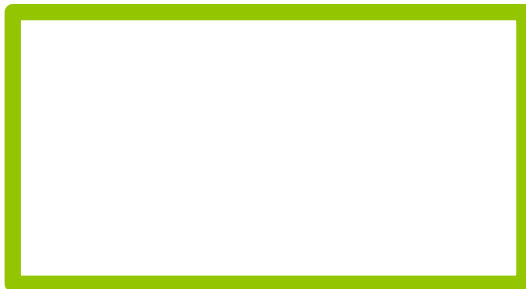
- جزو قوی ترین نرم افزارهای تحقیق در عملیات
- حل مسائل بزرگ و پیچیده
- یک زبان برنامه نویسی مدل سازی با قابلیت بالا
- حل مسائل برنامه ریزی خط (LP)، برنامه ریزی غیرخطی (NLP)، برنامه ریزی صحیح مختلط (MIP) و...

### مقدمه ای کوتاه بر تحقیق در عملیات

- بررسی مسائل قابل حل بدون نیاز به مدل سازی
- لزوم مدل سازی برای حل مسائل بزرگتر
- تعریف دقیق متغیرها
- تبدیل داده ها به پارامتر یا جدول

### مقدمه ای کوتاه بر تحقیق در عملیات

محیط مستطیل مقابل حداکثر ۲۰ متر میباشد. حداکثر مساحت این مستطیل چه مقدار است؟



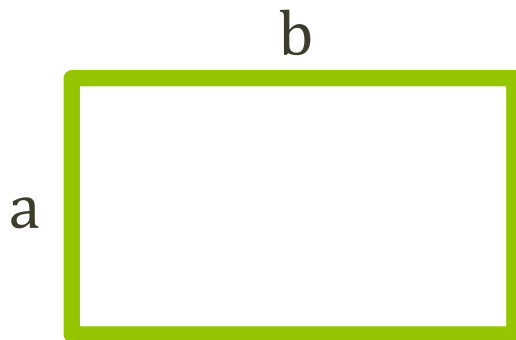
جواب قابل حدس بود، میدانیم مساحت مستطیل در صورتی بیشینه است که اضلاع آن برابر باشند. در نتیجه یک مربع با محیط ۲۰ وجود دارد. پس هر ضلع آن ۵ متر است و در نتیجه مساحت آن برابر با ۲۵ میباشد.

## مقدمه ای کوتاه بر تحقیق در عملیات

### سوال

از کجا میدانستیم مساحت مربع بیشتر است؟

### مقدمه ای کوتاه بر تحقیق در عملیات



$$a + b = 10$$

$$a = b - 10$$

$$S = a * b = (b - 10) * b = b^2 - 10b$$

$$ds/db = 2 * b - 10 = 0$$

$$b = 5 \Rightarrow a = 5 \Rightarrow \underline{S = 25}$$

# مقدمه ای کوتاه بر تحقیق در عملیات

همیشه سوال ها اینقدر ساده نیستند و با این شیوه نمیتوان به جواب رسید. یک راه حل برای حل مسائل، مدل سازی ریاضی می باشد. سوال بعد را برای یادآوری مدل سازی مطرح شده است.

### تعریف مساله

● مساله حمل و نقل

● هزینه حمل و نقل \$90 برای هر یک کیلوگرم محصول بر هزار مایل

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



# Modeling

## مدل سازی مساله

اندیس ها

کارخانه ها  $i$

فروشگاه ها  $j$

داده ها

موجودی کالا در کارخانه ی  $i$   $a_i$

تقاضا کالا در فروشگاه  $j$   $b_j$

هزینه حمل کالا از کارخانه  $i$  به

فروشگاه  $j$   $c_{ij}$

متغیر های تصمیم گیری

میزان کالای حمل شده از کارخانه  $i$  به

فروشگاه  $j$   $x_{ij}$

تابع هدف

$$\text{Min } Z = \sum_i \sum_j c_{ij} x_{ij}$$

محدودیت ها

$$\forall i : \sum_j x_{ij} \leq a_i$$

$$\forall j : \sum_i x_{ij} \geq b_j$$

$$x_{ij} \geq 0$$

# نکته ی اساسی

### \*\*\* تبدیل داده ها به پارامتر یا جدول

در مسئله ی قبل تمام محدودیت ها بصورت گروه محدودیت نوشته شد. این امر برنامه نویسی را بسیار ساده تر میکند.

بدون انجام این کار، مدل سازی مسائل بزرگ غیر ممکن است.

تصور کنید بجای استفاده از سیگما مجبور شوید صدها متغیر را به صورت زیر بنویسید:

$$a + b + c + d + .... \leq 2*a + 2*c + 2*e + ...$$

### ساختار کلی برنامه

- Sets
- Data
  - Parameters
  - Tables
  - Scalars
- Variables
- Equations
- Model & Solve Statements

### ساختار کلی برنامه

در ادامه پس از آموزش هر قسمت، کد مربوط به آن قسمت مسئله ی حمل و نقل که پیش تر مدل سازی شد را خواهید دید. پس مدل بیان شده برای مسئله ی حمل و نقل را به خاطر داشته باشید.

### نکات اولیه

- برای GAMS حروف کوچک و بزرگ فرقی ندارد.
- یک برنامه GAMS شامل بخش‌های متعددی است، هر بخش با یک (؛) خاتمه می‌یابد.
- هر خطی که با \* یا // شروع شود به عنوان توضیح در نظر گرفته می‌شود و در برنامه خوانده نمی‌شود.

# Sets

- معرفی مجموعه ی تمام اندیس‌هایی که در نوشتن معادلات مدل استفاده می‌شوند.
- شروع با کلمه sets و به دنبال آن معرفی نام و اعضای مجموعه.
- جدا کردن اعضا ی مجموعه با یک comma یا enter.
- Set و Sets از نظر برنامه یکسان هستند.

# Sets

$i = \{ \text{Tehran, Hamedan} \}$

مجموعه کارخانه ها

$j = \{ \text{Arak, Mashhad, Ahvaz} \}$

مجموعه بازار ها

sets

$i$  factories/Tehran, Hamedan/

$j$  markets/Arak, Mashhad, Ahvaz/;



# Sets

● نقش کاراکتر \*

$$T=\{10,11,12,...,100\}$$

$$M=\{m1,m2,m3,...,m10\}$$

برای معرفی این دو مجموعه از \* کمک می گیریم:

$$\text{Set } t \text{ } /10*100/;$$

$$\text{Set } m \text{ } /m1*m10/;$$

نکته : این ها عدد نیستند، رشته هستند.

# Sets

● گاهی در یک مدل نیاز است به مجموعه ی بیان شده، عنوان جدیدی داده شود.

sets

i /Tehran, Hamedan/

k /Tehran, Hamedan/ ;

و یا پس از تعریف i، به شکل زیر عمل میکنیم:

**Alias ( i , k ) ;**

به این صورت k همانند i میشود.

# Sets

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

Set

i factories /seattle, san-diego/

j markets /new-york, chicago, topeka/ ;

# Data

Scalar ●

Parameters ●

Tables ●

تخصيص مستقيم ●

# Scalar-data

● اسکالر پارامتریست که دامنه ندارد.

`scalar f freight in dollars per case per thousand miles/90/;`

نوشتن این توضیحات اختیاریست.

## Scalar-data

```
Scalar  ItemName  optional text  /value/ ;
```

## List-data

- وارد کردن data به صورت list
- استفاده از دستور parameters

parameters

a(i) capacity of plant i in cases / Tehran 350, Hamedan 600/

b(j) demand at market j in cases / Arak 325

Mashhad 300

Ahvaz 275/;

# List-data

- Parameter و Parameters برای نرم افزار یکسان است.
- نرم افزار بسیار User friendly است، حتی به صورت زیر نیز میتوان Parameter را تعریف کرد.

parameters

b(j) demand at market j in cases / Arak 325 , Mashhad 300  
Ahvaz 275/;



## List-data

Parameters	ItemName(Set)	optional text
	/ element1	value
	element2	value
	element3	value/ ;

# List-data

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

### Parameter

D(j) demand of markets /new-york 325

chicago 300

topeka 275/

S(i) supply of factories /seattle 350

san-diego 600/ ;

# Table-data

وارد کردن data به صورت table

استفاده از دستور table

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

	Arak	Mashhad	Ahvaz
Tehran	2.5	1.7	1.8
Hamedan	2.5	1.8	1.4 ;

نکته : موقعیت کاراکتری جدول اعداد ورودی باید با موقعیت کاراکتری عناوین فصل مشترک داشته باشد.(انجام این کار به کمک tab , space)

## Table-data

	Arak	Mashhad	Ahvaz
Tehran	2.5		1.8
Hamedan		1.8	1.4 ;

نکته: جاهای خالی به این معنیست که هیچ مقدار خاصی برای پارامتر مربوطه در نظر گرفته نشده است. اما! اگر از آن پارامتر در مدل استفاده شود، برنامه آن را برابر صفر فرض میکند.

## Table-data

Table	ItemName(Set1, Set2)	optional text	
		Set2element1	Set2element2
Set1element1	value(1,1)		value(1,2)
Set1element2	value(2,1)		value(2,2) ;

اجزای مجموعه دوم

اجزای مجموعه اول

## Table-data

این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

### Table

$C(i,j)$  cost of transportation between i and j

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

# Multidimensional-data

**Set**    **i** first index /first, second/  
          **j** second index /one, two, three/  
          **k** third index /m,n/ ;

**Parameters** Ketab(i,j,k) 3-dimensional structure /

First.one.m	18,	First.one.n	23
Second.one.m	25,	Second.one.n	43
First.two.m	36,	First.two.n	21
Second.two.m	11,	Second.two.n	48
First.three.m	93,	First.three.n	3
Second.three.m	47,	Second.three.n	47 / ;

### تخصیص مستقیم

- متفاوت با روش های قبل.
- عمل تعریف پارامتر و تخصیص مقدار به صورت مجزا انجام می شود.  
parameter  $c(i, j)$  transport cost in 1000s of dollars per case;  
 $c(i, j) = f * d(i, j) / 1000;$
- پارامتر های  $f$  و  $d$  قبلا تعریف شده است.
- نکته : گذاشتن ; در انتهای خط اول.



### تخصيص مستقيم

**Parameter**    ItemName(Set1, Set2)    **optional text** ;

ItemName(Set1, Set2) = ... ;

ItemName('Set1element', 'Set2element') = ... ;

### عملگرهای منطقی

And Or Xor Not

Operator		Results			
a	b	a and b	a or b	a xor b	not a
0	0	0	0	0	1
0	Non-zero	0	1	1	1
Non-zero	0	0	1	1	0
Non-zero	Non-zero	1	1	0	0

### عملگرهای منطقی

#### مثال

```
Parameter price(i) ;  
price('book')=  
(num('book') and num('pen'))*30 ;
```

در عبارت بالا در صورتی که تعداد کتاب ها و تعداد خودکارها مقداری غیر صفر داشته باشند، **price('book')** برابر با ۳۰ میشود. در غیر این صورت مقدار آن برابر صفر خواهد شد.

### عملگرهای منطقی

#### مثال

**Scalar price ;**

**Price = (4>2) ; → Price = 1**

نکته : نرم افزار گامس عبارات منطقی صحیح را « ۱ » در نظر میگیرد. همچنین عبارات منطقی غلط را « ۰ » در نظر میگیرد.

یک نمونه ی دیگر :

**Price = (4<3) or (2=2) ; → Price = 1**

عملگرهای منطقی

مثال

Price = (4 and 2) **xor** (2 or -1) ;

جواب ???

Price = (1) **xor** (1) = 0

## عملگرهای رابطه ای


عملگر گمس	معنی	عملگر
lt	کوچکتر از	<
le	کوچکتر یا مساوی با	<=
eq	مساوی با	=
ne	مخالف با	<>
ge	بزرگتر یا مساوی با	>=
gt	بزرگتر از	>

عملگر \$

If  $(b > 2)$ , then  $a = 5$ , else nothing change  
 $a$(b > 2) = 5;$

If  $(b > 2)$ , then  $a = 5$ , else  $a = 0$   
 $a = 5$(b > 2);$

## Variables



$$\max z = \sum_i \sum_j c_j * X_{ij}$$

نکته: Z هم یک متغیر است.

$$\sum_j h_{ij} * X_{ij} \leq t_i$$

$$\sum_i X_{ij} \leq P_j$$

$$X_{ij} \geq 0, int$$



# Variables

- معرفی متغیرهای تصمیم گیری و تابع هدف.
- تعیین نوع متغیرهای تصمیم گیری.

Variable Type	Allowed Range of Variable
Free (Default)	$(-\infty, \infty)$
Positive	$(0, \infty)$
Negative	$(-\infty, 0)$
Binary	0 or 1
Integer	1, 2, 3, ...

# Variables

variables

$x(i, j)$  shipment quantities in cases

$z$  total transportation costs in 1000s of dollars;

positive variables

$y(i)$  percentage of construction ;

Binary variable

Buy ;

# Variables

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

Variable

$z$  optimum answere;

Positive variable

$x(i,j)$  the amount of transportation between  $i$  and  $j$ ;

## کران پایین و کران بالا

کران بالا	.up
کران پایین	.lo
مقدار دقیق	.fx

$x.up('Tehran', 'Arak') = 3$  ;

$x.lo('Tehran', 'Arak') = 1$  ;

$x.fx('Tehran', 'Arak') = 1.8$  ;

نکته :  $fx$ . متغیر همچنان متغیر باقی میماند، اما مقدار ثابتی میگیرد.

# Equations

● تعریف نام گروه محدودیت ها

● تعریف محدودیت ها

# Equations

- انواع معادلات
- اسکالر: تنها یک معادله را بیان می کند.
- اندیس دار : بیش از یک معادله را بیان می کند. در این موارد لازم است اندیسی که معادلات روی آن تعریف شده اند نیز در مدل بیاید.

# Equations

● تعريف Equation :

```
Equation nameEQN(sets) comment;
```

# Equations

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

### Equation

hadaf

demand(j) to be sure of demands satisfaction

supply(i) to be sure factories inventory is enough ;



# Equations

- برای نوشتن یک محدودیت، ابتدا باید نام محدودیت را بنویسیم. سپس اگر گروه محدودیت بود، اندیس گروه را نیز مشخص کنیم. پس از آن از «..» استفاده کرده و جلوی آن محدودیت را بنویسیم. در انتها هم لازم است سمیکالن قرار دهیم.

مثال:

$$\text{timeconstraint}(i) \text{ .. } \text{sum}(j, h(i,j) * x(i,j)) = l = t(i) ;$$

# Equations

● بیان محدودیت های  $\leq = \geq$

=	=E=	equal to
$\geq$	=L=	less than or equal to
$\leq$	=G=	greater than or equal to

# Equations

### نکته:

وجود هر یک از توابع  $\sin, \cos, \log, \max, \min, \text{abs}, \text{sign}, \dots$  مدل را غیر خطی میکند! که در صحبت در کلاس آموزش پیشرفته گمس مورد بررسی قرار میگیرند.

صرفاً جهت آشنایی شما با برخی از این توابع، جدولی شامل نام این توابع و برخی مشخصات و سطح اجازه ی استفاده از آنها در مدل، در این اسلایدها قرار گرفته است.

## Functions of GAMS.

Functions	Description	Classification	Exogenous	Endogenous
abs	absolute	heterogeneous	legal	DNLP
arctan	arc tangent	homogenous	legal	NLP
ceil	homogenous	legal	illegal	
cos	cosine	discontinuous	legal	NLP
errorf	error	homogenous	legal	NLP
exp	exponent	homogenous	legal	NLP
floor	minimal	discontinuous	legal	illegal
log	natural	homogenous	legal	NLP
log10	decimal	homogenous	legal	NLP
mapval	function	discontinuous	legal	illegal
max	maximum	heterogeneous	legal	DNLP
min	minimum	heterogeneous	legal	DNLP
mod	residue	discontinuous	legal	illegal
normal	normal	illegal	illegal	illegal
power	whole	homogenous	legal	NLP
round	rounding-off	discontinuous	legal	illegal
sign		discontinuous	legal	illegal
sin	sine	homogenous	legal	NLP
sqr	squaring	homogenous	legal	NLP
sqrt	square root	homogenous	legal	NLP
trunc	truncation	discontinuous	legal	illegal
uniform	uniform	illegal	illegal	illegal

# Equations

● توابع اندیس دار

Sum	جمع بستن روی اندیس کنترلی
Prod	ضرب کردن روی اندیس کنترلی

● شیوه ی نمایش در برنامه

$$\text{Sum}(i, x(i,j)) = \sum_i x_{ij}$$

Sum

$$\sum_i \sum_j x_{ij} = \sum_i (\sum_j x_{ij})$$

Sum ( i , sum ( j , x ( i , j ) ) )

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

Sum

$$\sum_i C_i * \sum_j x_{ij}$$

Sum ( i , C(i) \* sum ( j , x ( i , j ) ) )

Sum

$$\sum_i C_i * \sum_j \sum_k d_k * x(i, j, k)$$

Sum ( i , C(i) \* sum ( (j,k) , d(k) \* x ( i , j , k ) ) )



## Sum

Sets  $r$  /  $n0 * n20$  / ;

$$\sum_{r=n1}^{n6} X(r)$$

sum(  $r \$(ord(r) \geq 1 \text{ and } ord(r) \leq 6)$  ,  $x(r)$  )

**Ord** نشان دهنده ی شماره ی ترتیبی است. یعنی مثلا  $n0$  در درایه ی ترتیبی ۰ام بردار است و  $n4$  در درایه ی ترتیبی ۴ام. به عبارت دیگر  $ord(n1)$  برابر با ۱ می باشد.

# Equations

equations

Cost      define objective functions

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);

## Equations

مثال: محدودیت زیر را که **limit** نام دارد، بنویسید.

$$\sum_j C_i * X_{i,j} \leq \sum_j \sum_i X_{i,j} \quad \forall i$$

Limit(i).. Sum( j, C(i) \* X(i,j) ) =| sum( (i,j) , X(i,j) )



sum نمی تواند هم در for باشد و هم در sum

# Equations

پس باید یک Set دیگر همانند i ایجاد کنیم. پس از alias استفاده میکنیم.

Alias (i,k);

Limit(i).. Sum( j, C(i) \* X(i,j) ) =l= sum( (k,j) , X(k,j) )



# Equations

- در صورت استفاده از نام یک عنصر دامنه، نماد های ' ' را در دو طرف نام آن عنصر قرار می دهیم.

`limit..x('seattle','chicago')=l=400;`

- متغیر ها می توانند در دو سمت معادلات ظاهر شوند.

# Equations

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

```
hadafe.. z =e= sum((i,j),c(i,j)*x(i,j));  
demand(j).. sum(i,x(i,j)) =g= d(j);  
supply(i).. sum(j,x(i,j)) =l= s(i);
```

# Model

● معرفی نام مدل و مشخص کردن equation هایی که میخواهیم در مدل باشند.

```
model  ModelName  optional comments  /Equations/;
```

نکته: اکثر مواقع میخواهیم همه ی محدودیت ها در مدل باشند. در این صورت بجای نوشتن نام همه ی آنها، مینویسیم «All».

```
model  ModelName  optional comments  /All/;
```

# Model

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

```
model FirstTest inja model ra sakhtim /all/;
```



# Solve

● فرا خواندن solver

```
solve ModelName using lp* minimizing* z*;
```

در این دستور، **solve** و **using** عباراتی هستند که به هیچ عنوان قابل تغییر نمیباشند. **Lp** \* در اینجا نوع مدل است. با تغییر نوع مدل، این واژه نیز عوض میشود. در اسلاید بعد انواع مدل ها شرح داده شده است.

**minimizing** \* به معنی کمینه کردن است. در صورتی که هدف از حل مسئله بیشینه ساختن باشد، بجای آن از **maximizing** استفاده میکنیم.

**Z** \* نام متغیری است که به عنوان جواب تابع هدف در مدل تعریف کرده ایم و موظفیم همان نام را در این جا به کار ببریم.

# solver

LP	Linear Programming
MIP	Mixed-Integer Programming
NLP	Non-Linear Programming
MCP	Mixed Complementarity Problems
MPEC	Mathematical Programs with Equilibrium Constraints
CNS	Constrained Nonlinear Systems
DNLP	Non-Linear Programming with Discontinuous Derivatives
MINLP	Mixed-Integer Non-Linear Programming
QCP	Quadratically Constrained Programs
MIQCP	Mixed Integer Quadratically Constrained Programs

# solver

● این قسمت در مسئله ی حمل و نقل مطرح شده به صورت زیر نوشته میشود:

`solve FirstTest using lp minimizing z;`

# display

- در انتهای برنامه یک خط دلخواه را می توان اضافه کرد:

`display VariableName.l , VariableName.m;`

- این خط باعث می شود در انتهای جواب برنامه، مقدار متغیرهای مسأله اولیه و مسأله دوگان به طور خلاصه نوشته شود.  
l. جواب و m. جواب دوگان را برای ما به نمایش در می آورد.

### دوره -- ساختار کلی برنامه

- Sets
- Data
  - Parameters
  - Tables
  - Scalars
- Variables
- Equations
- Model & Solve Statements

# نکاتی درباره ی برطرف کردن Error های برنامه

- سطرهای حاوی خطا با «\*\*\*\*» نمایش داده شده اند.
- توجه به اولین پیغام خطای نمایش داده شده.
- با از بین بردن اولین خطا و اجرای مجدد، بقیه ی خطاها تغییر میکنند. پس در صورتی که زمان اجرای برنامه کوتاه است، پس از رفع هر یک خطا، مجددا برنامه را اجرا کنید، سپس به سراغ خطاهای دیگر بروید.

شرکتی ۳ کالای مختلف تولید می کند که هر کدام از ۲ گیاه خاص تولید میشوند. با توجه به جدول زیر محاسبه کنید که از هر محصول (در یک هفته) چه قدر باید تولید شود و مقدار هر نوع گیاه که در یک محصول استفاده میشود را نیز مشخص کنید، طوری که سود شرکت ماکسیمم شود.

	Production Time Used for Each Unit Produced			Production Time Available per Week
	Product 1	Product 2	Product 3	
Plant 1	3 hours	4 hours	2 hours	30 hours
Plant 2	4 hours	6 hours	2 hours	40 hours
Unit profit	5	7	3	
Sales potential	7	5	9	

# Modeling



## دوره -- تعریف متغیرها و مرتب کردن داده ها

- تعریف دقیق متغیرها

- تبدیل داده ها به پارامتر یا جدول

### متغیرها و داده های مسئله

- $X_{ij}$ : محصول نوع  $j$  که در ایستگاه  $i$  تولید می شود.  $i=1,2$ ,  $j=1,2,3$
- $H_{ij}$ : مقدار ساعت برای ساخت محصول  $j$  در ایستگاه  $i$
- $C_j$ : سود محصول  $j$
- $P_j$ : پتانسیل فروش محصول  $j$
- $T_i$ : وقت در دسترس در ایستگاه  $i$

### مدلسازی مسئله

$$\max z = \sum_i \sum_j C_j * X_{ij}$$

● تابع هدف:

● محدودیت ها:

$$\sum_j h_{ij} * X_{ij} \leq t_i$$

۱- به ازای هر  $i$ :

$$\sum_i X_{ij} \leq P_j$$

۲- به ازای هر  $j$ :

$$X_{ij} \geq 0, \text{int}$$

۳- به ازای هر  $i$  و  $j$ :

## حل مسئله با نرم افزار GAMS

sets

```
i plant /plant1,plant2/
j product /product1*product3/;
```

parameters

```
c(j) profit /product1 5 , product2 7 , product3 3/
p(j) sales potential /product1 7 , product2 5 , product3 9/
t(i) time available
    /plant1 30
    plant2 40/ ;
```

table h(i,j) production time

	product1	product2	product3
plant1	3	4	2
plant2	4	6	2;

variables

```
X(I,J)
Z profit;
```

integer variable

```
X(I,J) plant i used in product j;
```

equations

```
profit objective
timeconstraint(i) limitation of time
salespotential(j) sales potential;
profit .. z=e=sum((i,j),c(j)*x(i,j));
timeconstraint(i) .. sum(j,h(i,j)*x(i,j))=l=t(i);
salespotential(j) .. sum(i,x(i,j))=l=p(j);
```

model main /all/;

solve main using MIP maximizing Z ;

display x.l , x.m,Z,l;

### دوره -- ساختار کلی برنامه

- Sets
- Data
  - Parameters
  - Tables
  - Scalars
- Variables
- Equations
- Model & Solve Statements

# تحلیل خروجی نرم افزار گمس

- **بازتاب مدل** – عبارت است از نمایش مدلی که نوشته ایم را نمایش میدهد. ( با دستور \$offlisting حذف میشود).
- محدودیت های باز شده
- Solver status
- Objective value
- Report summary
- Display

# Output

برای چاپ خروجی های مدل در فایل اکسل از دستور زیر استفاده میکنیم:

```
execute_unload "FileName.gdx" z.l, w.l;
```

```
execute 'gdxxrw.exe FileName.gdx var=z.l rng=SheetName1!c3'
```

```
execute 'gdxxrw.exe FileName.gdx var=w.l rng=SheetName2!h8'
```



# Output

نام فایل اکسل

متغیرهایی که میخواهیم در اکسل چاپ شوند

`execute_unload "FileName.gdx" z.l, w.l;`

`execute 'gdxxrw.exe FileName.gdx var=z.l rng=SheetName1!c3'`

`execute 'gdxxrw.exe FileName.gdx var=w.l rng=SheetName2!h8'`

نام صفحه ی اکسل که میخواهیم متغیر Z در آن چاپ شود.

در اسلاید بعد توضیح داده میشود.

# Output

**سوال:**

H8 که بعد از علامت تعجب نشان داده شده است چیست؟

آن سطر دستور چاپ متغیر `w` در اکسلی به نام `FileName` و در صفحه (sheet) ای به نام `SheetName2` را میدهد. H8 بیان میکند این متغیر کجای صفحه (sheet) چاپ شود.

اگر `w` فقط یک متغیر باشد، در درایه ی H8 چاپ میشود. اگر `w` بردار یا جدول باشد ( یعنی متغیر یک بعدی یا دو بعدی)، درایه ی H8 گوشه ی شمال غربی این جدول یا بردار میباشد.

# Input

- ورودی آرایه
- ورودی جدول

# Input

ورودی آرایه

```
parameter ParameterName(k) optional comment ;  
$call gdxxrw.exe D:\FileName.xlsx par=ParameterName  
Rng=SheetName!C4:D13 rdim=1  
$gdxin FileName.gdx  
$load ParameterName  
$gdxin  
;
```

اگر بردار را ستونی وارد کرده بودید، این عبارت را بنویسید.  
اگر سطری وارد کرده بودید، بجای آن باید بنویسید: cdim=1

## Input

ورودی جدول

**Parameter** TableName(m,w);

\$call gdxxrw.exe C:\FileName.xlsx par=TableName

Rng=SheetName!B2:RH7 rdim=1 cdim=1

\$gdxin FileName.gdx

\$load TableName

\$gdxin

;

گوشه شمال غربی

گوشه جنوب شرقی

اگر ترتیب این دو جابجا شود، ترانهاده ی  
ماتریس وارد Table خواهد شد.

## مساله: Switchboard game

قصد داریم تا بازی **SWITCHBOARD** را با برنامه ریزی خطی تحلیل کنیم. در این بازی تعدادی عدد داریم که قرار است این اعداد در یک صفحه شطرنجی قرار گیرند به شرطی که مجموع اعداد هر سطر(ستون) با آنچه که در انتهای آن سطر(ستون) نوشته شده است یکسان شود. مدلی ارائه کنید که اعداد داخل جدول زیر در جای صحیح خود قرار بگیرند. همچنین جدول نهایی این بازی را در فایل ارسالی خود قرار دهید.

0	1	2	5	7	25
0	1	3	5	7	17
1	1	4	5	8	24
1	2	4	5	8	17
1	2	5	7	9	26
1	2	5	7	9	21
1	2	5	7	9	21
1	2	5	7	9	15
34	36	30	40	26	

# Modeling

## مساله: Switchboard game

- تابع هدف را دارای مقداری برابر یک فرض می کنیم .
- محدودیت ها شامل موارد زیر است :
- هر خانه باید یک مقدار بگیرد .
- مجموع هر سطر ثابت است (معادل عدد مقابل آن است )
- مجموع هر ستون نیز ثابت است ( عدد موجود در پایین هر سطر )
- تعداد هر عدد باید برابر تعداد تکرار داده باشد .



## مساله: Switchboard game

$X(i,j,k)$

در صورتی که در سطر  $i$  و ستون  $j$  عدد  $k$  باشد، برابر ۱ و اگر نباشد برابر صفر میشود.

$SSatr(i)$  مجموع اعداد سطر  $i$

$Ssotoon(j)$  مجموع اعداد ستون  $j$

$Freq(k)$  تعداد اعداد  $k$  که باید در جدول باشد

# Switchboard game مدل مساله

$$\text{Max } Z = 1$$

$$\sum_j \sum_k X(i, j, k) * k = \text{SSatr}(i) \quad \forall i$$

$$\sum_i \sum_k X(i, j, k) * k = \text{SSotoon}(j) \quad \forall j$$

$$\sum_k X(i, j, k) = 1 \quad \forall i, j$$

$$\sum_i \sum_j X(i, j, k) = \text{freq}(k) \quad \forall k$$

$$X(i, j, k) = 0 \text{ or } 1$$

# مدل مساله Switchboard game

این مسئله را به عنوان تمرین به وسیله ی نرم افزار گمس مدل کنید.

جواب سوال :

9	1	7	7	1
2	5	2	1	7
1	9	9	4	1
5	1	5	5	1
8	0	2	7	9
5	7	3	5	1
2	8	2	4	5
2	5	0	7	1