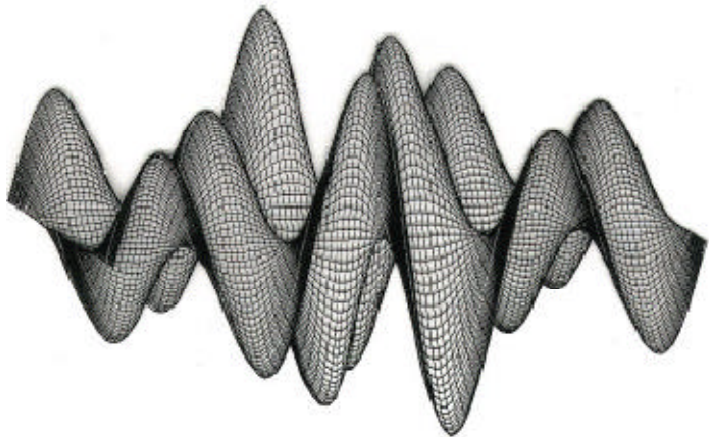




دانشگاه صنعتی شریف

دانشکده صنایع

## آموزش مقدماتی *GAMS*



ترجمه و تالیف :

امین وحیدی منفرد

**GAMS**

*Modeling for the Real World*

# آموزش مقدماتی *GAMS* (General Algebraic Modeling System)

## فهرست

۱. مقدمه	۳
۲. ساختار یک مدل <i>GAMS</i>	۷
۳. مجموعه ها	۹
۴. داده ها	۱۱
۴.۱. ورود داده به وسیله لیست ها	۱۱
۴.۲. ورود داده با جدول ها	۱۳
۴.۳. ورود داده به صورت تعیین مستقیم	۱۳
۴.۴. ورود داده از سایر برنامه ها	۱۵
۵. متغیرها	۱۷
۶. معادلات	۱۸
۶.۱. اعلان معادلات	۱۸
۶.۲. نمادگذاری مجموع (و ضرب) در <i>GAMS</i>	۱۸
۶.۳. تعریف معادلات	۲۰
۷. تابع هدف	۲۲
۸. دستور مدل و حل	۲۲
۹. دستور نمایش	۲۴
۱۰. بانک داده 'LO, L, UP, M'	۲۴
۱۰.۱. تعیین حد متغیرها و یا مقادیر اولیه	۲۵
۱۰.۲. تبدیل ها و نمایش مقادیر بهینه	۲۶

۲۸.....	۱۱. خروجی GAMS.....
۲۸.....	۱۱.۱. نسخه ی برگشتی مدل.....
۳۰.....	۱۱.۲. پیغام های خطا.....
۳۳.....	۱۱.۳. نگاشته های مرجع.....
۳۵.....	۱۱.۴. لیست معادلات.....
۳۶.....	۱۱.۵. آمار و اطلاعات مدل.....
۳۶.....	۱۱.۶. گزارشات وضعیت.....
۳۷.....	۱۱.۷. گزارشات حل.....
۳۹.....	۱۲. جمع بندی.....

GAMS قوی‌ترین و مجهزترین برنامه‌ی حل مسائل تحقیق در عملیات است و در مسائل برنامه‌ریزی خطی، غیرخطی، عدد صحیح، مختلط و ... بکار می‌رود و انواع محدودیت‌ها را دربر می‌گیرد. البته لازم به ذکر است که علاوه بر تحقیق در عملیات در موارد زیر نیز به کار می‌رود:

- اقتصاد کشاورزی
- مهندسی شیمی
- اقتصاد محیطی
- امور مالی
- توسعه‌ی اقتصادی
- انرژی
- مهندسی
- جنگل‌داری
- تجارت بین‌المللی
- اقتصاد خرد
- علم مدیریت
- اقتصاد کلان
- نظامی
- حمل و نقل
- ریاضی
- فیزیک

این راهنما، راهنمایی سریع، ولی خلاصه‌ای کامل از GAMS و خصوصیات آن است. این متن برای شروع کار با GAMS مناسب است.

مثالی که در نوشتار برای تسهیل آموزش به آن می‌پردازیم نمونه‌ای از مسئله حمل‌ونقل خطی است. نمایش جبری این مسئله معمولاً به صورت زیر است:

اندیس‌ها:

$i =$  کارخانه‌ها

$j =$  بازارها

داده‌های معین:

$a_i =$  عرضه‌ی محصول کارخانه‌ی  $i$ ام

$b_j =$  تقاضا در بازار  $j$ ام

$c_{ij} =$  هزینه جابجایی بین کارخانه  $i$ ام و بازار  $j$ ام

متغیرهای تصمیم:

$x_{ij}$  مقدار محصولی که از کارخانه  $i$  به بازار  $j$  حمل می‌شود

به طوری که  $x_{ij} \geq 0$  برای هر  $i, j$

محدودیت‌ها:

$$\sum_j x_{ij} \leq a_i, \text{ for all } i$$

رعایت کردن محدودیت عرضه در کارخانه  $i$ :

$$\sum_i x_{ij} \geq b_j, \text{ for all } j$$

ارضای تقاضا در بازار  $j$ :

تابع هدف:

$$\text{Minimize } \sum_i \sum_j c_{ij} x_{ij} \quad (\$K)$$

تعدادی از نکات لازم جهت مدل‌سازی صحیح که ساختار *GAMS* نیز با آن سازگار است به شرح زیر است:

اول، تمام عناصر مدل با توجه به نوع شناسایی و گروه‌بندی می‌شوند. دوم، ترتیب عناصر به نحوی انتخاب می‌شود که هیچ نمادی قبل از این که تعریف شود مورد رجوع قرار نگیرد. سوم، واحد تمام عناصر مشخص می‌شود. چهارم، واحد را در مقایسه انتخاب می‌کنیم که تمام مقادیر عددی کوچک و مناسب باشند. (علامت  $\$K$  در اینجا به معنی هزار دلار است).

در *GAMS* اندیس‌ها <sup>1</sup>sets، داده‌های معین <sup>2</sup>parameters، متغیرهای تصمیم <sup>3</sup>variables و محدودیت‌ها و تابع هدف <sup>4</sup>equations خوانده می‌شوند.

نمایش مدل حمل‌ونقل در *GAMS* بسیار شبیه نمایش جبری فوق است با این تفاوت که این نسخه *GAMS* آن برای کامپیوتر قابل خواندن و پردازش کردن است.

---

<sup>1</sup> مجموعه‌ها

<sup>2</sup> پارامترها

<sup>3</sup> متغیرها

<sup>4</sup> معادله‌ها

برای نمونه فرض کنید که دو کارخانه کنسروسازی و سه بازار داریم که اطلاعات آنها به شرح زیر است.

	فواصل حمل و نقل			عرضه
	بازارها			
کارخانه‌ها	<i>BandarAbbas</i>	<i>Shiraz</i>	<i>Mashhad</i>	
<i>Hamedan</i>	۱.۳	۰.۹	۲.۱	۳۵۰
<i>Tehran</i>	۱.۳	۰.۱	۰.۹	۶۰۰
تقاضا	۳۲۵	۳۰۰	۲۷۵	

فواصل داده شده به هزار کیلومتر هستند و هزینه‌ی حمل نیز ۹۰ هزار تومان برای هر واحد از محصول در هر هزار کیلومتر در نظر گرفته شده است.

نمایش این مدل در *GAMS* به صورت زیر است :

Sets

```
i canning plants / Hamedan, Tehran /
j markets / BandarAbbas, Shiraz, Mashhad / ;
```

Parameters

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

b(j) demand at market j in cases

```
      / BandarAbbas 325
      Shiraz 300
      Mashhad 275 / ;
```

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

```
      BandarAbbas  Shiraz  Mashhad
Hamedan          1.3      0.9      1.2
Tehran           1.3      1.0      0.9 ;
```

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

Parameter c(i,j) transport cost in 1000s of tomans per case ;

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

Variables

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

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

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 ;

پس از وارد کردن این متن در فایل ورودی *GAMS* اگر همه چیز به درستی پیش رود و خطایی رخ ندهد حمل و نقل بهینه در پایان مانند زیر نمایش داده می شود.

	BandarAbb~	Shiraz	Mashhad
Hamedan	50.000	300.000	
Tehran	275.000		275.000

همچنین شما قیمت های مرزی<sup>۵</sup> (ضرایب سیمپلکس) را به شرح زیر دریافت می کنید.

	Shiraz	Mashhad
Hamedan		0.027
Tehran	0.009	

این نتایج نشان می دهد که مثلاً، نفرستادن جنس از همدان به مشهد بهینه است اما اگر شما اصرار بر فرستادن محصول دارید فرستادن هر واحد ۰.۰۲۷ هزار تومان یا ۲۷.۰۰ تومان به هزینه بهینه خواهد افزود.

---

Marginal Costs °

## ۲. ساختار یک مدل GAMS

در ادامه ما در مورد عناصر اساسی یک مدل GAMS با رجوع به مثال فوق صحبت می‌کنیم. عناصر اساسی این‌ها هستند:

ورودی‌ها	خروجی‌ها
<ul style="list-style-type: none"> <li>• مجموعه‌ها(اندیس‌ها)-Sets</li> <li>• اعلان آن‌ها</li> <li>• مشخص کردن اعضا</li> <li>• داده-Data</li> <li>• (پارامترها، جدول‌ها،اسکالر<sup>۶</sup>)</li> <li>• اعلان آن‌ها</li> <li>• تخصیص مقادیر</li> <li>• متغیرها-Variables</li> <li>• اعلان آن‌ها</li> <li>• تخصیص نوع</li> <li>• تعیین حدود و یا مقادیر اولیه(اختیاری)</li> <li>• معادله‌ها-Equations</li> <li>• اعلان آن‌ها</li> <li>• تعریف آن‌ها</li> <li>• دستور مدل<sup>۷</sup> و حل<sup>۸</sup></li> <li>• دستور نمایش-Display-(اختیاری)</li> </ul>	<ul style="list-style-type: none"> <li>• نسخه‌ی برگشتی مدل - Echo Print</li> <li>• نقشه‌های رجوع -Reference Maps</li> <li>• فهرست معادلات-Equation Listings</li> <li>• گزارشات وضعیت-Status Reports</li> <li>• نتایج-Results</li> </ul>

دیگر قسمت‌های پیچیده‌ی اختیاری شامل ذخیره و برگرداندن مدل‌های قدیمی، و ساخت چندین مدل در یک اجرا، نیز در GAMS وجود دارند، اما این آموزش فقط شامل مولفه‌های اساسی مدل‌سازی در GAMS می‌شود.

قبل از برخورد با عناصر جزئی این ملاحظات کلی را در نظر داشته باشید:

۱. از مهم‌ترین قوانین حاکم بر دستورات GAMS این است که، هر عنصر مدل نمی‌تواند قبل از تعریف مورد استفاده قرار بگیرد.

Scalar<sup>۶</sup>

Model<sup>۷</sup>

Solve<sup>۸</sup>



۲. نحوه‌ی چیدمان دستورات *GAMS* به هر صورت که کاربر می‌پسندد می‌تواند باشد، مثلاً چند دستور در یک خط یا یک دستور در چند خط و یا خط یا خطوط خالی برای جداکردن و خوانا و قابل فهم‌تر کردن مدل، می‌تواند قرار گیرند.
۳. وقتی شما به عنوان یک کاربر مبتدی با *GAMS* کار می‌کنید، باید هر دستور را با یک نقطه ویرگول<sup>۹</sup> (;) به پایان برسانید.
۴. مترجم<sup>۱۰</sup> *GAMS* بین حروف بزرگ و کوچک تفاوتی قائل نیست.
۵. مستندسازی برای مفیدبودن مدل‌های ریاضی بسیار اهمیت دارد. بسیار بهتر و دقیق‌تر است که توضیحات در داخل مدل درج شود تا اینکه در بالای آن نوشته شود. دو روش برای وارد کردن مستندات در درون مدل وجود دارد، یکی این که در ستون اول سطر یک ستاره وارد کنیم در این صورت این خط کلاً توسط مترجم *GAMS* نادیده گرفته می‌شود. روش دیگر و مهم‌تر قراردادن این توضیحات در دستورات *GAMS* است. (همان‌طور که در مدل بالا می‌بینید که توضیحات لازم در جلوی هر عنصر آمده است).
۶. همان‌طور که در جدول بالا می‌بینید ایجاد عناصر مدل شامل مرحله اعلان آن و بعد تعریف و یا تعیین آن می‌شود. اعلان یعنی بیان وجود داشتن آن عنصر و دادن نام به آن، و تعریف یا تعیین آن مشخص کردن شکل و یا مقدار آن. برای محدودیت‌ها باید اعلان و تعریف آن‌ها در دستورات جداگانه‌ای صورت بگیرد اما برای سایر عناصر می‌توان در یک دستور و یا دستورات جداگانه این کار را انجام داد.
۷. نام عناصر باید با حرف شروع شود و با ۹ حرف و یا عدد دیگر دنبال شود. دقت کنید که نام عناصر نمی‌تواند فارسی باشد و حتماً باید انگلیسی باشد.

---

<sup>۹</sup> Semicolon

<sup>۱۰</sup> Compiler

مجموعه‌ها عناصر اساسی یک مدل *GAMS* هستند که مترادف اندیس‌ها در نمایش جبری مدل‌ها هستند. مثال حمل و نقل بالا فقط شامل یک دستور *Set* است:

Sets

i canning plants / Hamedan, Tehran /

j markets / BandarAbbas, Shiraz, Mashhad / ;

دستورات فوق به آن معنی است که ما دو مجموعه به نام‌های *i* و *j* تعریف و اعضای آن را به شرح زیر تعریف کرده‌ایم.

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

$j = \{\text{BandarAbbas, Shiraz, Mashhad}\}$  .

فرق مدل جبری و مدل *GAMS* این قسمت، در این است که در جبر از  $\{\}$  برای مجموعه و در *GAMS* از  $//$  برای مجموعه‌ها استفاده می‌شود زیرا بسیاری از کامپیوترها در گذشته و معدودی از آن‌ها در حال حاضر کلیدهای  $\{\}$  را ندارند. همچنین دقت کنید که در نام‌های چند کلمه‌ای باید به جای فاصله از خط تیره ( \_ ) استفاده شود. کلماتی که بعد از اعلانات می‌آید متن<sup>۱۱</sup> خوانده می‌شود. این‌ها همان مستندسازی‌های درونی مدل هستند.

همچنین می‌توان ایجاد هر یک از مجموعه‌های *i* و *j* را در یک دستور جداگانه انجام داد:

Set i canning plants / Hamedan, Tehran / ;

Set j markets / BandarAbbas, Shiraz, Mashhad / ;

انتخاب فواصل و خطوط خالی مانند کوچک و یا بزرگ بودن حروف به شما بستگی دارد. همچنین برای *GAMS* بین کلمه‌ی *Set* و *Sets* تفاوتی وجود ندارد.

---

<sup>۱۱</sup> Text

یک راه راحت برای تخصیص اعضا به یک مجموعه استفاده از ستاره است. این علامت برای حالتی که اعضا دارای توالی هستند به کار گرفته می‌شود:

```
Set t time periods /1991*2000/ ;
```

```
Set m machines /mach1*mach24/ ;
```

این دستورات هم‌معنی دستورات زیر است:

```
t = {1991,1992,1993, .....,2000}
```

```
m = { march1, march2,.....march24},
```

توجه کنید که اعضای مجموعه به صورت رشته‌ای از کاراکترها ذخیره می‌شوند، پس اعضای  $t$  عددی نیستند.

یکی از امکانات مناسب دیگر دستور <sup>۱۲</sup> alias است، که برای دادن نام دیگری به مجموعه‌ای که قبلاً تعریف شده استفاده می‌شود:

```
Alias (t,tp) ;
```

نام  $tp$  مانند  $t$  در نمادگذاری ریاضی است. این دستور برای مدل‌هایی که در آن‌ها عناصر در درون همان مجموعه‌ها با هم تعامل دارند مناسب است.

مجموعه‌های بالا مثالی از مجموعه‌های ایستا<sup>۱۳</sup> هستند به این معنی که اعضای آن‌ها توسط کاربر تعیین می‌شود و تغییر نمی‌کند. *GAMS* امکانات زیادی برای ایجاد مجموعه‌های پویا<sup>۱۴</sup> دارد، به نحوی که اعضایشان از طریق عملیات منطقی یا بر اساس نظریه مجموعه‌ها بدست آیند.

هم‌چنین می‌توان اعضای مجموعه‌ها یا پارامترها را از خارج از *GAMS*، مثلاً از *Access* یا *Excel* و دیگر برنامه‌ها وارد کرد، که در بخش ۴.۴ توضیح داده شده است.

---

<sup>۱۲</sup> نام مستعار

<sup>۱۳</sup> static

<sup>۱۴</sup> dynamic

## ۴. داده‌ها

در مدل فوق سه شکل اساسی برای ورود داده‌ها به مسئله نمایش داده شده است:

- Lists
- Tables
- Direct assignments

در سه زیر بخش بعدی به نوبت به توضیح این سه شکل می‌پردازیم.

### ۴.۱. ورود داده به وسیله لیست‌ها

دستور این شکل به صورت زیر است :

Parameters

a(i) capacity of plant i in cases

/ Hamedan 350

Tehran 600 /

b(j) demand at market j in cases

/ BandarAbbas 325

Shiraz 300

Mashhad 275 / ;

دستورات فوق به این معنی است که دو پارامتر داریم که  $a$  و  $b$  نام دارد و دامنه<sup>۱۵</sup>ی آن‌ها به ترتیب  $I$  و  $J$  است. (دامنه، مجموعه و یا چند مجموعه است که بر روی آن یک پارامتر، متغیر یا معادله تعریف می‌شود.) همچنین دستورات فوق مقادیر پارامترها را مشخص می‌کند و شامل متون سندی برای هر پارامتر نیز هست.

---

<sup>۱۵</sup> domain

نوشتن دستورات فوق به صورت زیر نیز امکان پذیر است :

Parameter a(i) capacity of plant i in cases

/ Hamedan 350

Tehran 600 / ;

Parameter b(j) demand at market j in cases

/ BandarAbbas 325

Shiraz 300

Mashhad 275 / ;

هنگام استفاده از شکل لیست به نکات زیر توجه کنید :

۱. نحوه‌ی چیدمان به هر شکلی می‌تواند باشد فقط باید کل لیست درون / / قرار گیرد و جفت‌های عضو-مقدار (مثلاً در مثال بالا Hamedan 350 یک جفت عضو-مقدار است) باید با کاما جدا شود و یا در دو خط جداگانه وارد شوند.
۲. از ; برای جدا کردن قسمت‌های مختلف این دستور استفاده نکنید زیرا در این شیوه همین یک دستور هم برای اعلان و هم برای تعیین استفاده می‌شود.
۳. مترجم این نرم‌افزار قسمتی به نام بررسی دامنه دارد که هر عضو ذکر شده در بخش عضو-مقدار به مجموعه‌ی ذکر شده تعلق دارد یا خیر. برای مثال اگر شما در بخش عضو-مقدار به جای Hamedan به اشتباه Hamadan را وارد کنید مترجم این نرم‌افزار پیغام خطایی مبنی بر نبود چنین عضوی در مجموعه‌ی I صادر خواهد کرد.
۴. صفر مقدار پیش فرض تمام پارامترها است. به همین سبب شما فقط نیاز دارید تا عناصر غیر صفر را وارد کنید. هم-چنین در بخش عضو-مقدار می‌توانید زوجها را به هر ترتیبی بیاورید.
۵. یک اسکالر به عنوان یک پارامتر که دامنه ندارد به حساب می‌آید و فقط شامل یک مقدار می‌شود.

Scalar f freight in tomans per case per thousand KM /90/ ;

اگر یک پارامتر دامنه‌اش دو و یا بیشتر مجموعه باشد نیز می‌توان از شکل لیست استفاده کرد، البته این فرم به خصوص در مواقعی که آرایه‌ها تنک<sup>۱۶</sup> و یا فوق‌تنک<sup>۱۷</sup> هستند، کاربرد دارد. (یعنی تعداد درایه‌ی صفر زیادی دارند).

---

<sup>۱۶</sup> sparse

<sup>۱۷</sup> supersparse

## ۴.۲. ورود داده با جدولها

متخصصین در زمینه بهینه‌سازی اذعان دارند که بسیاری از داده‌های ورودی به شکل جداول هستند. نمونه‌ای از این فرم ورود داده در مدل حمل و نقل فوق دیده می‌شود :

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

	BandarAbbas	Shiraz	Mashhad
Hamedan	1.3	0.9	1.2
Tehran	1.3	1.0	0.9 ;

دستور بالا ابتدا پارامتر  $d$  را روی دامنه‌های  $i$  و  $j$  تعریف می‌کند و سپس مقادیر آن را تعیین می‌کند. ورودی خالی در جدول به معنی صفر است.

مانند شکل قبلی بررسی دامنه برای این فرم نیز صورت می‌گیرد. نحوه‌ی ورود جداولی که در یک خط جا نمی‌گیرند و یا جداول بیش از دو بعدی در 'راهنمای کاربران GAMS' آمده است.

## ۴.۳. ورود داده به صورت تعیین مستقیم

در این فرم ورود داده‌ها بر خلاف دو شکل قبلی اعلان و تعیین در دو دستور مجزا صورت می‌گیرد. این شکل نیز در مدل فوق قابل ملاحظه است :

Parameter c(i,j) transport cost in 1000s of tomans per case ;

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

دقت کنید که وجود  $f$  بین دو دستور برای متمایز شدن آن‌ها ضروری است.

همان‌طور که می‌بینید در خط اول پس از اعلان پارامتر و تعیین دامنه‌ی آن متن توضیحی در مورد آن آمده است. در ضمن برای تعیین یکی از اعضای این پارامتر خارج از فرمول داده شده می‌توانید از دستور زیر استفاده کنید :

```
c('Hamedan','BandarAbbas') = 0.40 ;
```

پارامترها می‌توانند بیش از یک بار مقدار دهی شوند و در نهایت آخرین مقدار داده شده را در خود نگه می‌دارند ولی یک پارامتر بیش از یک بار نمی‌تواند اعلان شوند.

مقادیر سمت راست یک دستور انتساب می‌تواند شامل عبارات ریاضی و توابع درونی نرم‌افزار باشند. اگر شما با زبان‌های برنامه‌نویسی C یا FORTRAN آشنایی داشته باشید با نوشتن دستورات انتساب در این نرم‌افزار نیز مشکلی نخواهید داشت. (توجه کنید که *GAMS* ویژگی‌هایی دارد که این دو زبان برنامه‌نویسی ندارند، مثلاً اینکه در این نرم‌افزار نیازی به نوشتن 'do loops' برای تعیین مقادیر  $c(i,j)$  برای همه‌ی جفت‌های  $(i,j)$  نیست.)

عملیات‌های استاندارد *GAMS* و توابع ارائه شده در راهنماهای نرم‌افزار وجود دارد. در اینجا مثال‌هایی از انتساب‌های صحیح آورده شده است :

```
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) = sqrt(sqr(xi(i) - xi(j) + sqr(x2(i) - x2(j))) ;  
present(j) = future(j)*exp(-interest*time(j)) ;
```

GAMS می‌تواند داده‌ها را از سایر برنامه‌های با روش‌های زیر بخواند :

- مکانیزم فایل شامل<sup>۱۹</sup>

این مکانیزم اطلاعات فایل‌های برنامه‌های دیگر به شکل یک فایل `.inc` که برای GAMS قابل فهم و فراخوانی است ذخیره می‌کند.

- فایل‌های GDY

GAMS می‌تواند از فایل‌هایی به نام GDY را بخواند که شامل اطلاعات مجموعه‌ها، پارامترها و ... باشد. همچنین می‌تواند (مثلا نتایج حل را) در آن‌ها بنویسد.

- پیوند مخصوص با صفحه گسترده‌ها و تعدادی برنامه‌ی دیگر

#### پیوند با صفحه گسترده‌ها

داده‌های صفحه گسترده از طریق سه رویه می‌تواند به GAMS وارد شود. برای نوشتن و خواندن داده می‌توان از دستورات `Xlimport` یا `Gdxxrw` استفاده کرد. علاوه بر این یک برنامه گرافیکی به نام `Xls2gms` وجود دارد که به شما اجازه وارد کردن اطلاعات از صفحات Excel و تبدیل آن به فایل `.inc` را می‌دهد. همچنین در رایانه‌هایی که برنامه Excel روی آن‌ها وجود ندارد می‌توان از `Xldata` برای خواندن از فایل‌های Excel استفاده کرد.

فرض کنید که می‌خواهیم نام شهرهای تولید کننده (اعضای مجموعه‌ی `i`) را از یک فایل Excel که این نام‌ها در برگ اول آن و در خانه‌های `a1` و `a2` است بخوانیم، دستورات لازم به شرح زیر خواهد شد :

(البته هر یک از این دستوراتی که گفته خواهد شد، امکانات جانبی فراوانی دارند که می‌توانید در راهنمای `McCarl` آن‌ها را مطالعه کنید.)

**Gdxxrw:** `Gdxxrw Inputfile Output=filename options`

```
$call "Gdxxrw Data.xls set=i Rng=a1:a2 rdim=1"
```

<sup>۱۹</sup> Including

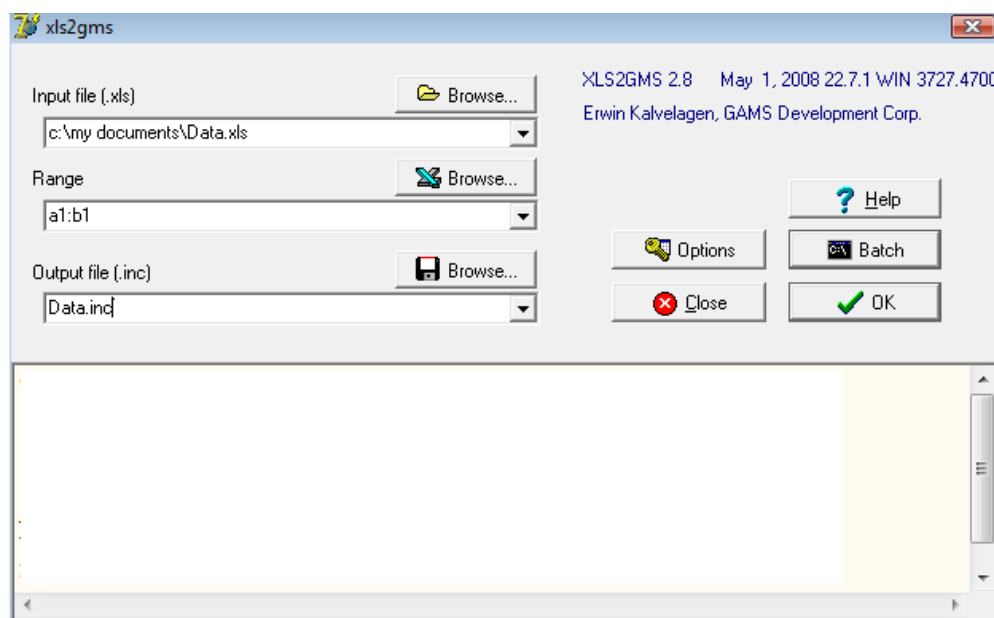


حال فرض کنید می‌خواهیم میزان عرضه‌ی هر یک از این تولیدکننده‌ها را از صفحه‌ی دوم همین فایل و از خانه‌های a1 و b1 بخوانیم :

Xls2gms: \$call =XLS2gms i=inputsheet o=outputinclude r=range

```
i canning plants /
$call =XLS2gms"i=c:\my documents\Data.xls" o=c:\my documents\Data.inc r=sheet2!a1:b1
$include Data.inc
/
```

البته اگر Xls2gms.exe را از داخل پوشه‌ی GAMS اجرا کنید می‌توانید از طریق پنجره‌ای به شکل پنجره‌ی زیر داده‌ها را وارد کنید.



شکل ۱- ماژول Xls2gms

حال فرض کنید که می‌خواهیم سود به دست آمده پس از حل را در صفحه‌ی سوم همین فایل وارد کنیم :

```
execute_unload 'transportdata.gdx', z;
execute 'gdxxrw.exe transportdata.gdx var=z rng=sheet3!a1' ;
```

هم‌چنین برای وارد کردن داده‌ها از فایل‌های بانک‌های اطلاعاتی مانند Mdb و Sql ، به ترتیب از دستورات Mdb2gms و Sql2gms و یا Gdxviewer استفاده کنیم.

## ۵. متغیرها

مثال‌هایی از اعلان و تعریف متغیرها در مثال فوق ذکر شده است :

Variables

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

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

همان‌طور که ملاحظه می‌کنید متغیر  $Z$  که همان هزینه‌ی کل و مقدار تابع هدف است دامنه ندارد و یک کمیت اسکالر است. هر مدل بهینه‌سازی باید چنین متغیری برای تابع هدف داشته باشد. بعد از اعلان هر متغیر باید نوعش مشخص شود. انواع صحیح متغیرها به صورت زیر است :

بازه مجاز متغیر	نوع متغیر
$-\infty$ to $+\infty$	Free
0 to $+\infty$	Positive
$-\infty$ to 0	Negative
0 or 1	Binary
0,1,..., 100	Integer

متغیر اسکالر تابع هدف باید از نوع Free باشد. در مدل حمل و نقل  $Z$  به صورت پیش‌فرض Free در نظر گرفته می‌شود اما متغیرهای  $x(i,j)$  باید با دستور زیر نامنفی وارد شوند :

Positive variable  $x$  ;

توجه کنید که دامنه‌ی  $X$  نباید در تعیین نوع آن آورده شود.

در قسمت دهم، نحوه‌ی تعیین حدود پایین، حدود بالا، و مقادیر اولیه برای متغیرها شرح داده خواهد شد.

## ۶. معادلات

قدرت زبان‌های مدل‌سازی جبری هنگامی آشکارتر می‌شود که بخواهیم معادلات و نامعادلات یک مدل را ایجاد کنیم. این بدان علت است که در این زبان‌ها اگر گروهی از معادلات ساختار مشابهی داشته باشند، تمام اعضای این گروه به صورت هم‌زمان ایجاد می‌شوند و نیازی به تعریف تک تک آن‌ها نیست.

### ۶.۱. اعلان معادلات

همان‌طور که گفته شد معادلات باید در دستورات مجزا اعلان و تعریف شوند :

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) ;
```

واژه‌ی Equation در این نرم‌افزار معانی گسترده‌ای دارد و هم روابط مساوی (معادله) و هم نامساوی (نامعادله) را دربرمی‌گیرد. یک نام معادله می‌تواند شامل یک معادله، مانند cost در بالا و یا چند معادله یا نامعادله، مانند demand باشد که دارای دامنه است.

### ۶.۲. نمادگذاری مجموع (و ضرب) در GAMS

با توجه به اینکه وارد کردن نماد ریاضی مجموع در کامپیوتر دشوار است در GAMS از نماد Sum استفاده می‌شود که دارای دو نشانوند<sup>۲۰</sup> (آرگومان) است :

<sup>۲۰</sup> argument

Sum(index of summation, summand)

دو آرگومان توسط کاما از هم جدا می‌شوند. آرگومان اول دامنه‌ی این مجموع را تعیین می‌کند و آرگومان بعدی عبارتی است که باید روی این دامنه جمع زده شود که به شکل هر عبارت ریاضی می‌تواند باشد.

به عنوان یک مثال ساده :

Sum(j, x(i,j))

که معادل عبارت  $\sum_j x_{ij}$  است.

یک مجموع کمی پیچیده‌تر می‌تواند به شکل زیر باشد، که معادل عبارت  $\sum_i \sum_j c_{ij} x_{ij}$  است.

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

و یا می‌تواند به شکل یک مجموع در مجموع دیگر باشد :

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

علامت ضرب متوالی نیز مانند مجموع است، فقط به جای واژه‌ی Sum از Prod استفاده می‌کنیم:

prod(j, x(i, j))

که عبارت فوق هم‌معنی  $\prod_j x_{ij}$  است.

مجموع و ضرب متوالی در فرم "تعیین مستقیم" پارامترها بسیار کاربرد دارند. برای مثال :

```
scalar totsupply total supply over all plants
```

```
totsupply = sum(i, b(i)) ;
```

تعریف معادلات به خاطر گوناگونی آن‌ها، پیچیده‌ترین دستورها در *GAMS* است. عناصر تشکیل دهنده‌ی تعریف یک معادله به ترتیب این‌ها هستند :

- نام معادله‌ای که تعریف می‌شود
- دامنه
- وضع محدودیت دامنه (اختیاری)
- نماد ' . '
- عبارت سمت چپ
- عملگر نسبی : =e=, =l=, یا =g=
- عبارت سمت راست

مثال حمل و نقل سه تا از این دستورات را شامل می‌شود :

```
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) ;
```

دستور فوق در خط آخر معادل عبارات زیر است :

```
demand(BandarAbbas)..
x(Hamedan, BandarAbbas) + x(Tehran, BandarAbbas) =g= 325 ;
demand(Shiraz)..
x(Hamedan, Shiraz) + x(Tehran, Shiraz) =g= 300 ;
demand(Mashhad)..
x(Hamedan, Mashhad) + x(Tehran, Mashhad) =g= 275 ;
```

به خاطر داشته باشید که حتی اگر تعداد گره‌ها به ۲۰۰۰۰ برسد و مسئله به یک مسئله واقعی تبدیل شود این دستورات تغییری نمی‌کند.

نکات زیر را به خاطر داشته باشید :

- در بسیاری از مسائل دنیای واقعی، بعضی از اعضای دامنه‌ی یک معادله، به خاطر یک استثنا، باید حذف و یا از بقیه متمایز شوند. *GAMS* با استفاده از خصیصه‌ای به نام دلار<sup>۲۱</sup> یا "مانند آن"<sup>۲۲</sup> به راحتی با این فقدان ساختار تطبیق پیدا می‌کند. این عملگرها در این نوشتار آورده نشده است اما می‌توان آن‌ها را در راهنمای کاربر *GAMS* پیدا کرد. همچنین امکان محدود کردن دامنه می‌تواند کاملاً ضروری باشد تا اندازه مدل دنیای واقعی در بازه‌ی قابل حل بودن قرار بگیرد.

- عملگرهای ضروری معانی زیر را دارد :

=l= کوچکتر مساوی

=g= بزرگتر مساوی

=e= مساوی

دقت کنید که علامت = تنها در تعیین مستقیم پارامترها استفاده می‌شود و =e= تنها در تعاریف معادله به کار می‌رود. "تعیین مستقیم" مقدار مورد نظر را قبل از حل به پارامتر می‌دهد اما تعریف یک معادله، یک رابطه را بیان می‌کند که تا قبل از حل شدن ارضا نمی‌شود. به همین دلیل تعریف معادله باید شامل متغیرها باشد ولی تعیین مستقیم پارامترها نباید شامل متغیرها شود.

- متغیرها می‌توانند در سمت راست یا چپ یک معادله ظاهر شوند. همچنین یک متغیر می‌تواند بیش از یک بار در یک معادله وجود داشته باشند. نرم‌افزار به صورت خودکار قبل از فراخوانی حل کننده، معادلات را به فرم استاندارد خود (یعنی به شکلی که متغیرها در سمت چپ باشد و عدم حضور تکراری یک متغیر رعایت شود) درمی‌آورند.

- تعریف یک معادله می‌تواند در هر قسمت مدل نوشته شود اما باید متغیرها و پارامترهای استفاده شده در آن باید قبلاً اعلان شده باشند. معادلات باید به همان ترتیبی که اعلان شده‌اند تعریف شوند.

---

<sup>۲۱</sup> §

<sup>۲۲</sup> such that

## ۷. تابع هدف

*GAMS* هیچ‌گونه عنصر صریحی به نام تابع هدف ندارد. برای مشخص کردن تابعی که باید بهینه شود، شما باید متغیری آزاد در علامت و اسکالر ایجاد کنید و در یک تعریف معادله آن را با تابع هدف برابر کنید.

## ۸. دستور مدل و حل

در دستور مدل پس از وارد کردن نام مدل باید لیستی از نام معادلات را در بین دو / آورد. اگر مدل باید تمام معادلاتی که قبلاً تعریف شده‌اند را شامل شود شما می‌توانید دستور (.L) or (.l) Activity level /all/ را در لیست وارد کنید. در مثال ما یک دستور مدل وجود دارد :

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

این دستور زائد است اما برای کاربران پیشرفته که چند مدل را در یک بار اجرا کردن *GAMS* ایجاد می‌کنند کارا است. اگر ما بخواهیم به جای عبارت فوق به صورت مستقیم نام معادلات را وارد کنیم، دستور آن به شکل زیر خواهد شد :

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

چون دامنه‌ها جز نام معادلات نیستند نباید آن‌ها را در دستور مدل آورد.

پس از اعلان و تعریف تمام عناصر مدل باید دستور حل را فراخوانی کنیم :

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

شکل کلی دستور حل به صورت زیر است :

۱. کلیدواژه‌ی Solve

۲. نام مدلی که باید حل شود

۳. کلیدواژه‌ی `using`

۴. انتخاب یک روش حل موجود

لیست کامل روش‌های حل به شرح زیر است :

`lp` برای برنامه‌ریزی خطی

`nlp` برای برنامه‌ریزی غیرخطی

`mip` برای برنامه‌ریزی عدد صحیح مختلط

`rmip` برای برنامه‌ریزی عدد صحیح مختلط آزاد شده<sup>۲۲</sup>

`minlp` برای برنامه‌ریزی عدد صحیح مختلط غیرخطی

`rminlp` برای برنامه‌ریزی عدد صحیح مختلط غیر خطی آزاد شده

`mcp` برای مسائل مکمل مختلط

`cns` برای دستگاه محدود غیر خطی

---

<sup>۲۳</sup> relaxed



## ۹. دستور نمایش

دستور حل سبب رخ دادن بسیاری از اتفاقات می‌شود. اشکال مختلفی از این مدل ساخته می‌شود، مثلاً ساختار صحیح نحوه ورود این مدل به حل کننده ایجاد می‌شود و حل کننده خوانده می‌شود و خروجی آن به فایل خروجی فرستاده می‌شود. برای یافتن جواب بهینه‌ی مسئله‌ی اصلی و مسئله‌ی دوگان می‌توان خروجی را جستجو کرد و یا می‌توانیم نمایش دادن این‌ها را از برنامه بخواهیم :

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

این دستور سبب می‌شود مقدار پایانی  $x.a$  و همچنین مقدار مرزی (هزینه کاسته شده<sup>۲۴</sup>)  $x.m$  مربوط به متغیرها در یک جدول با عنوان مناسب درج شود.

## ۱۰. بانک داده 'LO, L, UP, M'

*GAMS* دارای یک بانک داده‌ی کوچک است که رکوردهای آن اطلاعاتی از متغیرها و معادلات را نگهداری می‌کنند. در هر رکورد چهار فیلد<sup>۲۵</sup> وجود دارد :

.lo = حد پایین

.l = مقدار تراز یا اولیه

.up = حد بالا

.m = مقدار دوگان یا مرزی

---

<sup>۲۴</sup> reduced costs

<sup>۲۵</sup> field

نحوه‌ی دسترسی به این کمیت‌ها به این صورت است که ابتدا نام معادله یا متغیر و سپس نام فیلد مورد نظر و در صورت نیاز نام دامنه یا عنصری از دامنه، آورده شود. این نرم‌افزار به کاربر اجازه‌ی خواندن و نوشتن این داده‌ها را می‌دهد که این امکان مزایای زیادی برای کاربران پیشرفته به وجود می‌آورد. مثال‌هایی از استفاده از این بانک داده در ذیل آمده است.

#### ۱۰.۱. تعیین حد متغیرها و یا مقادیر اولیه

حد بالا و پایین متغیرها مجموعه‌ای ایست که به صورت خودکار مطابق نوع متغیرها است (free, positive, negative, binary, or integer) ولی این حدود می‌تواند توسط کاربر بازنویسی شود. به عنوان مثال :

```
x.up(i,j) = capacity(i,j) ;  
x.lo(i,j) = 10.0 ;  
x.up('Hamedan','BandarAbbas') = 1.2*capacity('Hamedan','BandarAbbas') ;
```

در برنامه‌ریزی غیرخطی، این مطلب خیلی مهم است که مدل‌ساز به حل‌کننده با محدود کردن بازه‌ها تا آن‌جا که امکان دارد بین حد بالا و حد پایین، کمک کند. همچنین تعیین یک حل اولیه که حل‌کننده حل را از این نقطه شروع کند می‌تواند بسیار مفید باشد. به عنوان مثال، در یک مدل موجودی دارای محدودیت، متغیرها  $quantity(i)$  هستند و مبرهن است که جواب بهینه‌ی نسخه‌ی بدون محدودیت این مسئله پارامتری به نام  $eoq(i)$  است. حدسی را برای نقطه‌ی بهینه مسئله محدودیت‌دار وارد می‌کنیم :

```
quantity.l(i) = 0.5*eoq(i) ;
```

(مقدار پیش فرض اولیه صفر است مگر اینکه صفر در بازه‌ی کران‌دار نباشد.)

فهم این مطلب مهم است که فیلدهای  $.lo$  و  $.up$  کاملاً تحت کنترل کاربر هستند ولی فیلدهای  $.l$  و  $.m$  پس از مقدار دهی اولیه توسط کاربر توسط حل‌کننده کنترل می‌شوند.

## ۱۰.۲. تبدیل‌ها و نمایش مقادیر بهینه

(خواندن این بخش برای کاربران مبتدی اختیاری است.)

پس از حل مدل مقادیر بهینه مسئله اصلی و مسئله دوگان در بانک داده و به ترتیب در فیلدهای  $a$  و  $m$ . ذخیره می‌شوند که آن‌ها را می‌توان تبدیل کرد، خواند و نمایش داد.

برای مثال، می‌توان با اضافه کردن دستورات زیر به انتهای مدل، درصدی از تقاضای بازار  $\lambda_m$  که توسط کارخانه‌ی  $m$  تامین می‌شود را مشاهده کرد:

```
parameter pctx(i,j) perc of market j's demand filled by plant i;  
pctx(i,j) = 100.0*x.l(i,j)/b(j) ;  
display pctx ;
```

که خروجی فوق را بدنبال خواهد داشت :

```
PARAMETER pctx perc of market j's demand filled by plant i
```

	BandarAbb~	Shiraz	Mashhad
Hamedan	15.385	100.000	
Tehran	84.615		100.000

برای مثالی که شامل مقادیر ثانویه شود، معادلات نسبت<sup>۲۶</sup> که عموماً در مسائل امتزاج و تصفیه<sup>۲۷</sup> به کار می‌روند در نظر بگیرید. این مدل‌های برنامه‌ریزی خطی مربوط به تعیین مقدار هر ماده‌ی اولیه‌ی در دسترس برای ورود به محصولات نهایی مورد نظر است. فرض کنید  $y(i,j)$  مقدار ماده  $i$  که به کار رفته در تولید محصول  $j$  باشد. فرض کنید معادله‌ی نسبت این باشد که بیش از ۲۵ درصد یک محصول از یک ماده‌ی اولیه نباشد :

<sup>۲۶</sup> 'ratio constraints

<sup>۲۷</sup> blending and refining

$$y(i,j)/q(j) = 0.25 ;$$

برای اینکه مدل خطی بماند باید دستور را به شکل زیر درآوریم :

$$\text{ratio}(i,j).. y(i,j) - .25*q(j) = 0.0 ;$$

اما مشکل اینجاست که پس از خطی کردن مدل مقدار دوگان این محدودیت مفهوم مناسبی ندارد. در نقطه‌ی بهینه این مقدار به ما خواهد گفت که اگر معادله خطی را به صورت زیر درآوریم چقدر سود می‌کنیم :

$$y(i,j) - .25*q(j) = 1.0 ;$$

ما از مقادیر ثانویه چیز دیگری را می‌خواهیم، برای مثال ما می‌خواهیم بدانیم که سود نهایی حاصل از افزایش ۲۵ درصد به ۲۶ درصد چقدر خواهد بود :

$$y(i,j)/q(j) = 0.26 ;$$

با تبدیل مقادیر نهایی ناکارآمد توسط دستورات زیر مشکل حل خواهد شد :

```
parameter amr(i,j) appropriate marginal for ratio constraint ;
```

```
amr(i,j) = ratio.m(i,j)*0.01*q.l(j) ;
```

```
display amr ;
```

برای فهمیدن عبارت بالا توجه کنید که

$$y(i,j)/q(j) = 0.26 ;$$

معادل است با

$$y(i,j) - .25*q(j) = 0.01*q(j) ;$$

## ۱۱. خروجی GAMS

خروجی GAMS بسیار گسترده و پر از اطلاعات است. در این نوشتار آن را قسمت به قسمت توضیح می‌دهیم :

نسخه برگشتی مدل

نگاشته‌های مرجع پیام‌های خطا<sup>۲۸</sup>

نگاشته‌های مرجع لیست معادلات<sup>۲۹</sup>

آمار مدل

گزارشات وضعیت

گزارشات حل

نباید با خواندن راهنمای این نرم‌افزار دچار این تفکر شویم که نباید در مدل ما خطا وجود داشته باشد. GAMS با این درک طراحی شده که حتی باتجربه‌ترین کاربران نیز دچار خطا می‌شوند. GAMS سعی می‌کند خطاها را هر چه سریع‌تر پیدا کرده و از تبعات آن‌ها بکاهد.

### ۱۱.۱. نسخه‌ی برگشتی مدل

چه خطاها جلوی حل مدل را بگیرند چه مدل خطایی نداشته باشد، اولین قسمت خروجی نسخه‌ی برگشتی یا کپی مدل ورودی شما خواهد بود. به خاطر رجوع در آینده، GAMS شماره خطوط را در سمت راست نسخه‌ی کپی قرار می‌دهد. برای مثال حمل و نقل ما که خوشبختانه هیچ خطایی هم نداشت، نسخه‌ی کپی به شکل زیر خواهد شد :

1 Sets

2 i canning plants / Hamedan, Tehran /

---

Error Messages Reference Maps<sup>۲۸</sup>

Reference Maps Equation Listing<sup>۲۹</sup>

```

3      j markets / BandarAbbas, Shiraz, Mashhad / ;
4
5 Parameters
6      a(i) capacity of plant i in cases
7          / Hamedan 350
8          Tehran 600 /
9      b(j) demand at market j in cases
10         / BandarAbbas 325
11         Shiraz 300
12         Mashhad 275 / ;
13 Table d(i,j) distance in thousands of miles
14         BandarAbbas    Shiraz    Mashhad
15 Hamedan        1.3        0.9        1.2
16 Tehran         1.3        1.0        0.9 ;
17 Scalar f freight in tomans per case per thousand miles /90/ ;
18 Parameter c(i,j) transport cost in 1000s of tomans per case ;
19     c(i,j) = f*d(i,j)/1000 ;
20 Variables
21     x(i,j) shipment quantities in cases
22     z total transportation costs in 1000s of tomans ;
23 Positive variable x ;
24 Equations
25     cost      define objective function
26     supply(i) observe supply limit at plant i
27     demand(j) satisfy demand at market j ;
28
29 cost..      z =e= sum((i,j), c(i,j)*x(i,j)) ;
30
31 supply(i) .. sum(j, x(i,j)) =l= a(i) ;
32
33 demand(j) .. sum(i, x(i,j)) =g= b(j) ;

```

34

```
35 Model transport /all/ ;
```

36

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

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

در صورتی که با استفاده از علامت \$ دستوراتی را در ابتدای مدل اضافه کنیم چون این عبارات جز اساس مدل نیستند در این نسخه آورده نمی‌شوند اما شماره خط مدل پس از شماره خط آن‌ها به حساب می‌آید.

```
$title a transportation model
```

```
$offupper
```

دستور \$title سبب می‌شود که متن بعد از آن در بالای هر صفحه‌ی خروجی چاپ شود. دستور \$offupper نیز نیاز است تا نسخه‌ی برگشتی هم دارای حروف کوچک و هم بزرگ باشد.

## ۱۱.۲. پیغام‌های خطا

وقتی مترجم *GAMS* با یک خطا مواجه شود، یک پیغام خطای کد شده را در جای وقوع خطا قرار می‌دهد. این پیغام همواره با \*\*\*\* آغاز می‌شود و یک علامت \$ درست در زیر جایی که فکر می‌کند خطا رخ داده است، قرار می‌دهد. علامت \$ با یک کد عددی خطا که پس از نسخه‌ی برگشتی شرح داده شده است، دنبال می‌شود. مثال‌های زیادی در ذیل آمده است.

**مثال ۱:** وارد کردن دستور

```
set q quarterly time periods / spring, sum, fall, wtr / ;
```

نتیجه در نسخه‌ی برگشتی

```
1 set q quarterly time periods / spring, sum, fall, wtr / ;
```

```
**** $160
```

در این مورد مترجم با عنصر sum دچار مشکل می‌شود، زیرا این کلمه یک کلمه‌ی رزرو شده است. در راهنماهای نرم‌افزار لیست کاملی از کلمات رزرو شده آمده است. این خطا یکی از خطاهای متداول مبتدیان است. در پایان نسخه‌ی کپی تفسیر خطای کد ۱۶۰ را می‌بینیم:

ERROR MESSAGES

160 UNIQUE ELEMENT EXPECTED

مثال ۲: خطای رایج دیگر جا انداختن نقطه ویرگول قبل از یک تعیین مستقیم یا تعریف معادله است. در مثال قبل فرض کنید نقطه ویرگول قبل از تعیین  $C(i, j)$  را جا بیاندازیم:

parameter c(i,j) transport cost in 1000s of dollars per case

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

خروجی به شکل زیر خواهد بود:

16 parameter c(i,j) transport cost in 1000s of dollars per case

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

\*\*\*\* \$97 \$195\$96\$194\$1

ERROR MESSAGES

1 REAL NUMBER EXPECTED

96 BLANK NEEDED BETWEEN IDENTIFIER AND TEXT

(-OR-ILLEGAL CHARACTER IN IDENTIFIER)

(-OR-CHECK FOR MISSING ';' ON PREVIOUS LINE)

97 EXPLANATORY TEXT CAN NOT START WITH '\$', '=', or '...'

(-OR-CHECK FOR MISSING ';' ON PREVIOUS LINE)

194 SYMBOL REDEFINED

195 SYMBOL REDEFINED WITH A DIFFERENT TYPE

با ارتکاب به این خطا همان‌طور که می‌بینید ۵ پیغام خطا ایجاد شده است. در نظر داشته باشید که ابتدا به رفع خطای اول بپردازید و بقیه را رها کنید! اولین خطا با کد ۹۷ دیده می‌شود. این خطا به این دلیل ایجاد شده است که با جا انداختن نقطه ویرگول خط قبل *GAMS* فکر می‌کند که نماد موجود در این خط ادامه‌ی متن توضیحی خط قبل است. و پیغام خطا نیز به درستی به ما توصیه می‌کند تا در خط قبل به دنبال نقطه ویرگول مفقود بگردیم. اما پیغام‌های خطا همواره این چنین دقیق نیستند زیرا مترجم نمی‌تواند از قصد و نیت شما آگاه شود. برای رفع خطا سرخ‌هایی را که در خروجی *GAMS* فراوان است، را پیدا کنید. به عنوان مثال نقطه ویرگول مفقود می‌تواند با جستجو به دنبال C در لیست ارجاع متقابل و توجه به اینکه این پارامتر تعیین نشده است، یافته شود.

SYMBOL TYPE REFERENCES

C PARAM DECLARED 15 REF 17

مثال ۳: بسیاری از خطاها فقط به خاطر املاي غلط اتفاق می‌افتند و قبل از اینکه زیان‌آور شوند به دام می‌افتند. برای مثال Hamedan در جدول به گونه‌ای غیر از آن گونه که در اعلان مجموعه‌ها معرفی شده است هجی شود. که ما را با پیغام خطای زیر روبرو می‌کند:



```

4 sets
5 i canning plants /Hamedan, Tehran /
6 j markets /BandarAbbas, Shiraz, Mashhad / ;
7
8 table d(i,j) distance in thousand of miles
9 BandarAbbas Shiraz Mashhad
10 seattle 2.5 1.7 1.8
**** $170
11 Tehran 2.5 1.8 1.4 ;
ERROR MESSAGES
170 DOMAIN VIOLATION FOR ELEMENT

```

**مثال ۴:** اگر به جای  $b(j)$  به اشتباه  $dem(j)$  تایپ کنیم نتیجه این خواهد شد :

```

45 demand(j) .. sum(i, x(i,j) ) =g= dem(j) ;
**** $140
ERROR MESSAGES
140 UNKNOWN SYMBOL, ENTERED AS PARAMETER

```

**مثال ۵:** این مثال در مورد خطای ریاضی است، که بعضی اوقات توسط مبتدیان واقع می‌شود و *GAMS* در گرفتن این خطا ماهر است. برای مثال این دستور از نظر ریاضی متناقض است و غیرقابل تفسیر است.

For all  $i, \sum_j x_{ij} = 100$

خطاهای این دستور در اندیس‌های آن است زیرا وقتی قرار است این دستور روی هر عنصر دامنه  $i$  صدق کند اندیس گذاری داخل مجموع اشتباه است و هم‌چنین هیچ کنترلی روی اندیس  $j$  وجود ندارد. اگر ما این دستور بی‌معنی را در *GAMS* وارد کنیم هر دو خطای مذکور توسط مترجم گرفته خواهد شد.

```

meaninglss(i) .. sum(i, x(i,j)) =e= 100 ;
**** $125 $149
ERROR MESSAGES
125 SET IS UNDER CONTROL ALREADY [This refers to set i]
149 uncontrolled set entered as constant [This refers to set j]

```

آخرین قسمت خروجی در صورت وجود خطا یک جفت نگاشته‌ی مرجع است که خلاصه و تحلیلی از مدل ورودی برای اشکال‌زدایی و مستندسازی را شامل می‌شود.

اولین نگاشته‌ی مرجع یک نگاشته‌ی ارجاع متقابل<sup>۳۰</sup> است مانند همان‌هایی که در اکثر مترجم‌های کنونی دیده می‌شوند. این نگاشته‌ی مرجع، لیستی به صورت ارجاع متقابل و الفبایی از تمام عناصر (مجموعه‌ها، پارامترها، متغیرها، و معادلات) مدل است. این لیست نوع هر عنصر و یک مرجع کد شده برای هر حضور عنصر در ورودی را نشان می‌دهد. نگاشته‌ی ارجاع متقابل برای مثال حمل و نقل ما به شرح زیر است.

SYMBOL	TYPE	REFERENCES
A	PARAM DECLARED	10 DEFINED 11 REF 43
B	PARAM DECLARED	14 DEFINED 15 REF 45
C	PARAM DECLARED	26 ASSIGNED 28 REF 41
COST	EQU DECLARED	37 DEFINED 41 IMPL-ASN 49 REF 47
D	PARAM DECLARED	19 DEFINED 19 REF 28
DEMAND	EQU DECLARED	39 DEFINED 45 IMPL-ASN 49 REF 47
F	PARAM DECLARED	24 DEFINED 24 REF 28
I	SET DECLARED	5 DEFINED 5 REF 10 19 26 28 31 38 2*41 2*43 45 CONTROL 28 41 43 45
J	SET DECLARED	6 DEFINED 6 REF 14 19 26 28 31 39 2*41 43 2*45 CONTROL 28 41 43 45
SUPPLY	EQU DECLARED	38 DEFINED 43 IMPL-ASN 49

<sup>۳۰</sup>'cross-reference map'

REF 47

TRANSPORT MODEL DECLARED 47 DEFINED 47 IMPL-ASN 49

X VAR DECLARED 31 IMPL-ASN 49 REF 34

41 43 45 2\*51

Z VAR DECLARED 32 IMPL-ASN 49 REF 41

49

برای مثال لیست ارجاع متقابل به ما می‌گوید که نشانه‌ی A یک پارامتر است که در خط ۱۰ اعلان شده است، در خط ۱۱ تعیین مقدار شده است و در خط ۴۳ مورد رجوع قرار گرفته است.

برای کاربران مبتدی *GAMS*، لیست ارجاع متقابل ممکن است خیلی مهم نباشد. احتمالاً بیشترین کاربرد این ابزار در پیدا کردن عناصر ناخواسته که به اشتباه وارد مدل شده‌اند و سبب یک خطای نشانه‌گذاری یا خطای نحوی شده‌اند، می‌باشد.

دومین قسمت نگاشته‌ی مرجع لیستی از عناصر مدل است که با توجه به نوعشان گروه‌بندی شده‌اند و با متن استنادی خود لیست شده‌اند. این لیست برای مثال ما به شکل زیر است.

sets

i canning plants

j markets

parameters

a capacity of plant i in cases

b demand at market j in cases

c transport cost in 1000s of dollars per case

d distance in thousands of miles

f freight in dollars per case per thousand miles

variables

x shipment quantities in cases

z total transportation costs in 1000s of dollars

equations

cost define objective function

demand satisfy demand at market j

supply observe supply limit at plant i

models

transport

پس از اینکه شما یک ورودی عاری از خطا ساختید *GAMS* یک مدل می‌سازد. ولی آیا این مدل همان مدلی است که شما می‌خواستید؟

لیست معادلات بهترین ابزار برای بررسی و پاسخ به این سوال مهم است.

یک محصول دستور حل لیست معادلات است که یک نمونه خاص از مدل است. هنگامی که مقادیر فعلی پارامترها به شکل جبری مدل وارد شوند. برای مثال، محدودیت کلی تقاضا که در ورودی داده شده است به شکل زیر است.

```
-----demand =g= satisfy demand at market j
demand(BandarAbbas) ..
x(Hamedan, BandarAbbas) +x(Tehran, BandarAbbas) =g= 325 ;
demand(Shiraz) ..
x(Hamedan, Shiraz) +x(Tehran, Shiraz ) =g= 300 ;
demand(Mashhad) ..
x(Hamedan, Mashhad) +x(Tehran, Mashhad) =g= 275 ;
```

به صورت پیش‌فرض برنامه در این لیست به ما سه معادله‌ی مشخص و صریح از هر معادله‌ی کلی را برمی‌گرداند. برای تغییر این تعداد می‌توان از دستور زیر قبل از فرمان حل استفاده کرد.

```
option limrow = r ;
```

که ۲ عدد مورد نظر است. همچنین این قسمت خروجی شامل قسمتی می‌شود که لیست ستون‌ها خوانده می‌شود و مشابه لیست معادلات است که ضریب سه نمونه متغیر از هر متغیر کلی را نشان می‌دهد. برای تغییر تعداد این ستون‌ها نیز می‌توان دستور را به صورت زیر گسترش داد. (C عدد مرود نظر است)

```
option limrow = r, limcol = c ;
```

بهتر است پس از اشکال‌زدایی شدن مدل برای کوچکتر و مناسب‌تر شدن خروجی این دو مقدار را صفر کنیم تا اطلاعات زائد چاپ نشوند.

در مدل‌های غیرخطی، لیست معادلات *GAMS* تقریب درجه‌ی اول تیلور معادلات غیر خطی را نمایش می‌دهد. این تقریب‌ها از مقادیر اولیه‌ی متغیرها استفاده می‌کنند.

## ۱۱.۵. آمار و اطلاعات مدل

آخرین قسمت خروجی قبل از فراخوانی حل کننده گروهی از آمار و اطلاعات در مورد اندازه‌ی مدل است.

MODEL STATISTICS

BLOCKS OF EQUATIONS 3 SINGLE EQUATIONS 6

BLOCKS OF VARIABLES 2 SINGLE VARIABLES 7

NON ZERO ELEMENTS 19

Block تعداد معادلات و متغیرهای کلی و SINGLE تعداد سطرها و ستون‌ها در مدل نمونه‌ی ساخته شده است. برای مدل‌های غیرخطی آمار و اطلاعات دیگری نیز در مورد درجه‌ی غیرخطی بودن مسئله داده می‌شود.

## ۱۱.۶. گزارشات وضعیت

پس از انجام حل، حل کننده خلاصه‌ای از روند حل<sup>۳۱</sup> را که مهم‌ترین عناصر آن وضعیت حل کننده<sup>۳۲</sup> و وضعیت مدل<sup>۳۳</sup> است را نمایش می‌دهد. برای مثال ما خلاصه‌ی روند حل به شرح زیر است.

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

MODEL TRANSPORT OBJECTIVE Z

TYPE LP DIRECTION MINIMIZE

SOLVER BDMMLP FROM LINE 49

\*\*\*\* SOLVER STATUS 1 NORMAL COMPLETION

\*\*\*\* MODEL STATUS 1 OPTIMAL

\*\*\*\* OBJECTIVE VALUE 153.6750

RESOURCE USAGE, LIMIT 0.168 1000.000

ITERATION COUNT, LIMIT 2 1000

---

solve summary<sup>۳۱</sup>

SOLVER STATUS<sup>۳۲</sup>

MODEL STATUS<sup>۳۳</sup>

گزارشات آماری که در ابتدای آن‌ها \* \* \* \* وجود داشته باشد نشان دهنده‌ی خطا هستند. شما باید در فایل خروجی برنامه به دنبال این نشانه‌ها باشید. وضعیت حل کننده ۶ حالت و وضعیت مدل ۱۱ حالت دارد که در راهنمای کاربر *GAMS* توضیح داده شده است. در مسائل غیرخطی بهینه نسبی توسط نرم‌افزار به ما داده می‌شود (کاربر باید به دنبال رسیدن به وضعیت 2 *LOCALLY OPTIMAL* باشد) و بررسی این موضوع که آیا مدل محدب است یا نه به عهده‌ی کاربر است. (تا بتوان فهمید که آیا بهینه‌ی نسبی داده شده، بهینه‌ی کلی نیز هست یا نه؟)

هم‌چنین در مدل‌های عدد صحیح کاربر باید به دنبال رسیدن به وضعیت *INTEGER SOLUTION* 8 باشد که این به معنای رسیدن به یک جواب موجه و صحیح است و جزئیات دیگر نیز مربوط به رسیدن جواب به محدوده‌ی قابل تحمل بهینه‌ی نسبی یا مطلق است که کاربر تعریف کرده است.

## ۱۱.۷. گزارشات حل

اگر وضعیت حل کننده و مدل قابل قبول باشد، شما می‌توانید به بررسی جواب‌های مسئله به عنوان جواب‌های مورد نظر خود بپردازید. جواب‌ها به شکل استاندارد برنامه‌ریزی ریاضی به صورتی که در آن‌ها سطرها و ستون‌ها با توجه به نام گروه‌بندی و برجسب‌زنی شده‌اند، نمایش داده می‌شوند. در این شکل یک خروجی برای هر سطر (معادله کلی) و ستون (متغیر کلی) وجود دارد که حد پایین، سطح، حد بالا و مقدار مرزی آن را می‌دهد.

---- EQU SUPPLY observe supply limit at plant i

	LOWER	LEVEL	UPPER	MARGINAL
Hamedan	-INF	350.000	350.000	EPS
Tehran	-INF	550.000	600.000	.

---- EQU DEMAND satisfy demand at market j

	LOWER	LEVEL	UPPER	MARGINAL
BandarAbbas	325.000	325.000	+INF	0.225
Shiraz	300.000	300.000	+INF	0.153
Mashhad	275.000	275.000	+INF	0.126

---- VAR X shipment quantities in cases

	LOWER	LEVEL	UPPER	MARGINAL
Hamedan .BandarAbbas	.	50.000	+INF	.
Hamedan .Shiraz	.	300.000	+INF	.
Hamedan .Mashhad	.	.	+INF	0.036
Tehran.BandarAbbas	.	275.000	+INF	.
Tehran.Shiraz	.	.	+INF	0.009

تک نقطه‌ها در خروجی به معنی صفر هستند. EPS نیز به معنی اپسیلون است. و EPSEPS نشان‌دهنده‌ی تباہیدگی است. هم‌چنین اگر حل غیر موجه شود و یا به مقدار مرزی غلط برسد، به ترتیب با INFES و NOPT نمایش داده می‌شود. اگر مسئله بی‌کران شود با واژه‌ی UNBND نمایش داده می‌شود.

در پایان گزارشات حل، خلاصه‌ی گزارشی وجود دارد که تعداد کل سطرها و ستون‌های غیربهبینه، غیرموجه، و بی‌کران را به ما می‌دهد.

\*\*\*\* REPORT SUMMARY : 0 NONOPT

0 INFEASIBLE

0 UNBOUNDED

بعد از اینکه گزارش حل نوشته شد کنترل برنامه از حل‌کننده به GAMS برمی‌گردد و تمام مقادیر اصلی و مرزی به بانک داده و فیلدهای a و m داده می‌شوند.

برخی از مزایای زبان‌های مدل‌سازی جبری مانند *GAMS* در برابر یک ماتریس ساز یا یک زبان محاوره‌ای به شرح زیر است :

- I. با استفاده از نمادگذاری جبری شما می‌توانید یک مدل بهینه‌سازی را برای کامپیوتر تشریح کنید به همان سادگی که برای یک فرد آموزش دیده (در این زمینه) تشریح می‌کنید.
- II. مدل وارد شده در *GAMS* را می‌توان برای ایجاد مدل‌های بعدی نیز به کار برد. مخصوصاً در مواردی که مدل‌های بعدی فقط تغییرات جزئی پیدا می‌کنند.
- III. با تعریف یک جای عناصر مشابه هم در وقت صرفه‌جویی می‌شود و هم خطاهای ناشی از این مرحله کاهش می‌یابد.
- IV. توضیحات و مستندات مدل در این زبان مدل‌سازی به خاطر همراه بودن با خود مدل بسیار کارا تر، دقیق تر و به روز تر هستند.
- V. خروجی *GAMS* به راحتی خوانده و استفاده می‌شود. هم‌چنین می‌توان با دستوراتی مانند `display` به راحتی خروجی‌ها را تغییر داد.
- VI. پیگیری خطاها و اصلاح آن‌ها به دلیل مناسب بودن پیغام‌های خطا به راحتی صورت می‌گیرد.
- VII. با استفاده از علامت \$ و دیگر ویژگی‌های پیشرفته که در این متن مجال توضیح آن‌ها نشد می‌توان با مدل‌های با حجم بالا کار کرد. کاربردهای ویژه‌ی علامت \$ به شرح زیر است:
  - علامت \$ محدودیت‌های منطقی روی ترکیب اندیس‌های مجاز برای معادلات و متغیرها را ممکن می‌سازد. بدین وسیله شما می‌توانید سطرها و ستون‌های غیرضروری را پاک کرده و حجم مدل را در محدوده‌ی قابل حل بودن نگه دارید.
  - این علامت می‌تواند برای ایجاد مجموع‌ها و ضرب‌های متوالی پیچیده که در معادلات یا گزارشات سفارشی به کار گرفته می‌شوند، استفاده شود.
  - هم‌چنین می‌تواند برای صدور پیغام‌های خطا یا برای پایان دادن شرطی زودتر از موقع، مثلاً در هنگام انجام صحیح و خطا بر روی داده‌های مدل از آن استفاده کرد.

منابع

[1] Tutorial.pdf (فایل آموزشی موجود در قسمت راهنمای نرم‌افزار)

[2] McCarl Expanded GAMS User's Guide