

بنام خدا

جزوه آموزش مقدماتی نرم افزار GAMS در مهندسی قدرت

تهیه کننده : **سید مسعود محسنی بناب**

با همکاری

اساتید راهنما و مشاور : سعید جلیل زاده – عباس ربیعی – بهنام محمدی ایواتلو

شامل :

درباره GAMS

✓ GAMS چیست؟

✓ دستورات

✓ ویژگی ها

مثال های قدرتی

✓ یک مثال ساده بهینه سازی

✓ Economic Dispatch

✓ Unit Commitment

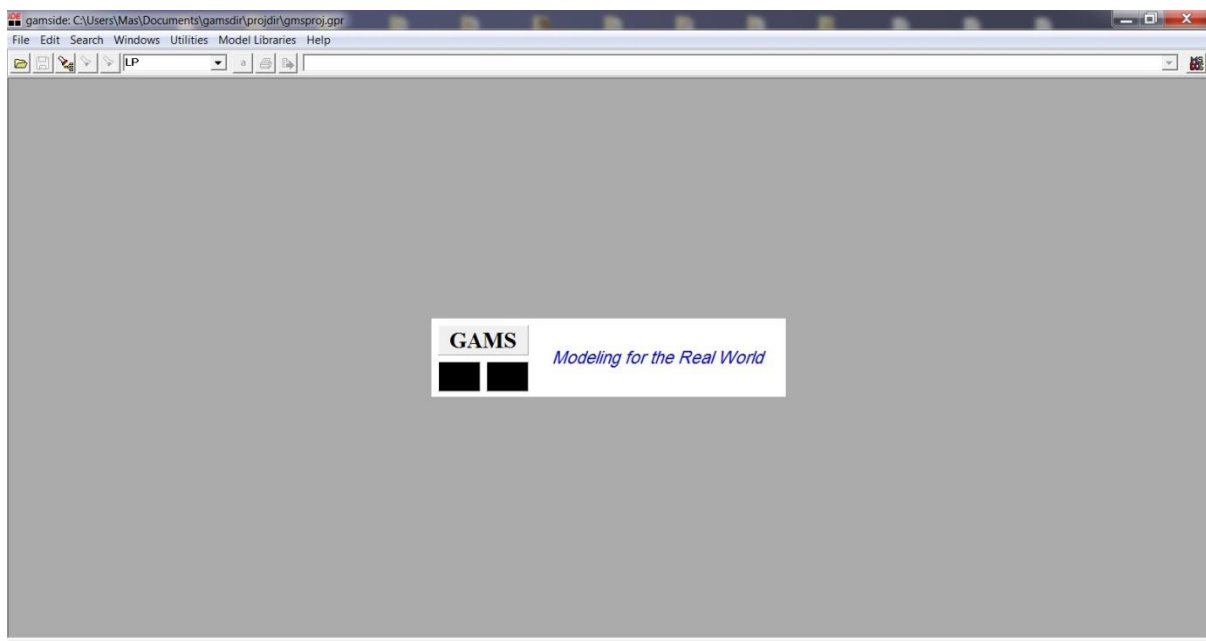
زمستان ۹۲

Mohseni@znu.ac.ir

GAMS چیست؟

سیستم GAMS (Generalized Algebra Modeling System) یک زبان برنامه نویسی مدل سازی با قابلیت بالاست که برای مسائل بهینه سازی (Optimization) بکار می رود. از GAMS برای حل مسائل برنامه ریزی خط (LP)، برنامه ریزی غیرخطی (NLP)، برنامه ریزی مختلط (MIP)، برنامه ریزی خطی صحیح مختلط (MINLP) و مسائل مکمل خطی (MCP) استفاده می کنند.

نرم افزار GAMS یک ابزار قدرتمند برای حل مسایل بهینه سازی در سیستم های قدرت می باشد. در شکل زیر محیط کار GAMS نشان داده شده است.



روش نصب و راه اندازی GAMS بر روی کامپیوتر شخصی

جهت دانلود و نصب این نرم افزار به آدرس GAMS.com بروید و ورژن رایگان آن قابل دانلود می باشد. توجه کنید که تاریخ کامپیوتر خود را برای شبکه های وسیع تر به قبل از سال ۲۰۰۳ ببرید.

ورژن رایگان در موارد زیر جواب می دهد:

Model limits:

- Up to 200 Constraints.
- Up to 300 Variables.
- Up to 2000 nonzero elements (1000 nonlinear).
- Up to 30 discrete Variables.

Solver (Global) limits:

- Up to 10 Constraints.
- Up to 10 Variables.

نکاتی در مورد نحوه نوشتن برنامه GAMS

۱- برای رفتن به خط بعدی از ENTER استفاده کنید.

۲- از کلید TAB برای رفتن به قسمت بعدی در یک سطر استفاده می شود، زیرا موجب مرتب شدن برنامه می شود.

۳- هر خطی که با '*' شروع شود به عنوان توضیح در نظر گرفته می شود و در برنامه خوانده نمی شود.

۴- برای GAMS حروف کوچک و بزرگ فرقی ندارد.

۵- یک برنامه GAMS شامل بخش های متعددی است، هر بخش با یک سیمی کالون (;) خاتمه می یابد.

۶- معمولا فایل حاصل با پسوند GMS مشخص می شود. مثلا اگر برنامه ای به نام TEST می نویسد فایلی به نام TEST.GMS ایجاد کنید.

۷- دستور \$ontext برای شروع دستورات توضیحی و \$offtext برای پایان دستورات توضیحی بکار می رود.

ویژگی های GAMS :

- ۱- قابلیت حل مسایل مقیاس کوچک تا مقیاس بزرگ با در نظر گرفتن قیود
- ۲- مدل از راه حل مستقل می باشد و بدین سان مدل می تواند با راه حل های مختلف (با تغییر solver) حل شود.
- ۳- با توجه به اینکه GAMS یک نرم افزار ریاضی است در مسایلی که ماهیت ریاضی دارند بسیار مفید است.
- ۴- از جملات رایج انگلیسی استفاده می کند و فهمیدن دستورات آن آسان است.

دستورات GAMS :

بطور هر برنامه GAMS دارای ۹ دستور استاندارد می باشد که ساختار استاندارد برنامه در جدول زیر نشان شده است. در ادامه به بررسی هر یک از این دستورات پرداخته می شود.

البته به غیر از این ۹ دستور اصلی برخی دستورات فرعی نیز وجود دارد که به فراخور توضیح داده می شود.

COMMAND	PURPOSE
SET	Used to indicate the name of the indices
SCALAR	Used to declare scalars
PARAMETER	Used to declare vectors
TABLE	Declare and assign the values of an array
VARIABLE	Declare variables, assign type and bounds
EQUATION	Function to be optimized and it's constraints
MODEL	Give a name to the model, and list the related constraints
SOLVE	Indicate what solver to use
DISPLAY	Indicate what you would like to display as output

بخش SETS

در این بخش تمام اندیس هایی که در نوشتن معادلات مدل استفاده می شوند معرفی می گردند. این اندیس ها عباراتی صرفی - عددی هستند. به عنوان مثال:

J / C1, C2, C3, C4, C5, C6 /

یا

J / C1 * C6 /

I / B1, B2, B3 /

توجه: نام هر اندیس تا ده کاراکتر اجازه داده شده و درج فاصله یا جای خالی در نام اندیس اجازه داده نشده است.

بخش PARAMETERS

در GAMS پارامترها بردارهای ثابت معلوم و مقدار پارامترها به عنوان داده ها در این بخش تعریف می شوند. به منظور تعریف مقادیر بردارها، لازم است، اندیس بردارها قبلا در بخش SETS تعریف شده باشد.

بخش TABLE

در GAMS یک ماتریس، مثلا ماتریس ضرایب قیود در LP، ماتریس ضرایب هزینه در حمل و نقل و...، در بخش TABLE آورده می شود.

به صورت زیر :

TABLE

	A (I, J)					
	C1	C2	C3	C4	C5	C6
B1	1	1	1	1	1	1
B2	2	-1	-2	4	0	0
B3	0	0	1	1	2	1;

توجه شود که تنها یک ماتریس در هر بخش TABLE می توان وارد کرد و لازم است با یک سیمی کالون (؛) خاتمه یابد. برای وارد کردن چندین ماتریس، همان تعداد TABLE لازم است.

بخش متغیرها VARIABLES

در بخش VARIABLES مجهولات مسأله را لیست می کنیم. متغیرها می توانند اسکالر باشند، یک بردار باشند، یک ماتریس دوبعدی باشند، یک ماتریس سه بعدی باشند و غیره ... در GAMS متغیرهای نامنفی با عنوان POSITIVE VARIABLES و متغیرهای نامثبت با عنوان NEGATIVE VARIABLES معرفی می شوند.

متغیرها می توانند BINARY باشند که در این صورت تنها مقادیر صفر و یک می گیرند.

بخش معادلات EQUATIONS

بخش بعدی بخش معادلات است که اسامی تابع هدف و قیود دیگر مسأله در آن تعریف می شود. نامعادلات شبیه معادلات در این بخش ظاهر می شوند.

بعد از نام گذاری، معادلات وارد می شوند. ابتدا نام معادله، به دنبال آن دو نقطه (..) آن گاه تعریف جبری قید که خیلی شبیه علائم استاندارد ریاضی است، می آید. ضرب به وسیله علامت (*)، تقسیم به وسیله علامت (/)، توان به وسیله (***) تعریف می شود.

بعضی از توابع استاندارد

نام	شرح	تعریف ریاضی
ABS	قدر مطلق	$ \arg $
COS	کسینوس	$\text{COS}(\arg)$
EXP	تابع نمایی	$\text{exp}(\arg)$

LOG	لگاریتم طبیعی	arg در مبنای e از
LOG10	لگاریتم معمولی	arg در مبنای ۱۰ از
SQR	توان دوم	Qrg*arg
SQRT	ریشه دوم	Sqrt(arg)

بعضی از علائم ریاضی

GAMS	علائم ریاضی معمولی
SUM (I ,)	$\sum_{i \in I}$
SUM ((I , J) ,)	$\sum_{i \in I} \sum_{j \in J}$
= E =	=
= G =	\geq
= L =	\leq

در GAMS راهی برای بیان کران بالا، کران پایین و قیود تساوی که با یک متغیر سر و کار دارند وجود دارد، مثال زیر را ببینید:

GAMS	علائم استاندارد ریاضی
X = L = 20	$x \leq 20$
X = G = 20	$x \geq 20$
X = E = 20	$x = 20$

به جای تعاریف فوق قبل از EQUATION ، دستورات زیر را می توانیم بیاوریم.

- X . UP = 20 ;
- X . LO = 20 ;
- X . FX = 20 ;

که در آن UP ، LO ، FX بیان کننده کران بالا، کران پایین و مقدار ثابت هستند.

بخش MODEL و SOLVE

در آخرین قسمت برنامه، MODEL داده می شود، نام مدل (نام فایل) حداکثر ۸ تا ۱۰ کاراکتر بسته به GAMS مورد استفاده، انتخاب می شود.

در GAMS کلمه مدل مجموعه ای از معادلات است که معمولاً یکی از آنها تابع هدف مسأله است، به عنوان مثال:

```
MODEL TEST / ALL / ;
```

این معنی را می دهد که تمام معادلاتی را که قبلاً معرفی شده است به عنوان مدل TEST در نظر گرفته شود.

از این قسمت می توان برای حل تجزیه LP هم استفاده کرد (Decomposition) طوری که علاوه بر تابع هدف قیودی را که برای حل کردن مد نظر است در این قسمت می نویسیم.
آخرین خط برنامه:

```
SOLVE TEST USING LP MINIMIZING COSTS;
```

به کامپیوتر اعلام می کند که مدل TEST را با استفاده از برنامه ریزی خطی (LP) که در کتابخانه برنامه حاضر است، به منظور می نیمم سازی متغیر COSTS (تابع هدف) حل کن.
در انتهای برنامه یک خط دلخواه را می توان اضافه کرد:

```
DISPLAY X . L , X . M ;
```

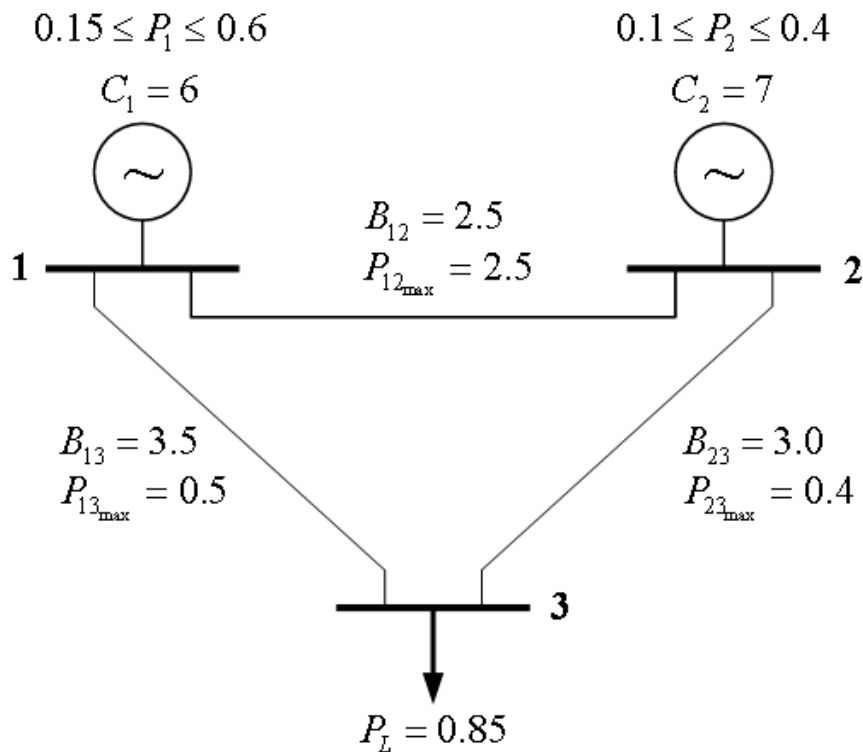
که این خط باعث می شود در انتهای جواب برنامه، مقدار متغیرهای مسأله اولیه و مسأله دوگان به طور خلاصه و جمع و جور نوشته شود.

مثال های قدرتی

مثال شماره ۱ :

ECONOMIC DISPATCH

سیستم سه باسه شکل زیر را در نظر بگیرید هدف نوشتن برنامه پخش بار اقتصادی با توضیح تفصیلی هر یک از بخش های برنامه است.



تعریف هر کدام از دستورات به تفکیک برای این مسئله با توضیح مربوطه :

SETS

SET برای تعریف شاخص ها و مقادیر محتمل آنها بکار می رود.

SETS

G	index of generators	/G1*G2/
N	index of buses	/N1*N3/
MAP(G,N)	indicates in what bus are the generators connected	/G1.N1,G2.N2/;

☑ نکته! مجموعه تعداد باس ها می توانست به حالت زیر نیز نوشته شود :

/N1, N2, N3/

SCALARS

دستور SCALARS مقدار ثابت را نشان می دهد. برای مثال در اینجا هزینه سوخت واحد تولیدی بر حسب \$/MBtu نشان داده شده است :

```
Scalar fcost fuel cost in dollars_MBtu /1.1/;
```

PARAMETER

ساختار این دستور برای تعریف اطلاعات یک یا چند متغیر که بشکل برداری اند بکار می رود. برای مثال در مثال ED پارامتر LOAD با ست N تعریف شده و مطابق دستور زیر فقط در باس سوم بار موجود است که مقدار آن ۰٫۸۵ می باشد.

```
PARAMETER  
LOAD (N) load at bus N / N3 0.85 /;
```

برای مثال اگر بنویسیم $0.9/N4$ با توجه به اینکه سیستم ما ۳ باسه است و باس ۴ ام وجود ندارد سیستم خطا نشان خواهد داد و برنامه run نمی شود.

TABLE

اطلاعات بصورت ماتریسی بصورت جداول یا پارامترهای دومتغیره (بسته به set) تعریف می شوند. ساختار TABLE را در پایین می بینیم. در این مثال ما دو جدول خواهیم داشت.

۱- جدول $GDATA(G,*)$: بوسیله شاخص G تعریف می شود و برای هر جفت از عناصر Set مانند G1 و Pmin در جدول مقدار خاصی مشخص است که در اینجا فرضاً ۰٫۱۵ است.

```
TABLE GDATA(G,*) generator input data  
* PMIN PMAX COST  
(kW) (kW) (dollar/kWh)  
G1 0.15 0.6 6  
G2 0.10 0.4 7;
```

۲- جدول $LDATA(N,N,*)$: یک آرایه سه بعدی را که مرتبط به دو Set است (در این مثال)

TABLE LDATA(N,N,*) line input data

	SUS	LIMIT
*	(S)	(kW)
N1.N2	2.5	0.3
N1.N3	3.5	0.5
N2.N3	3.0	0.4;

☑ دستور TABLE LDATA(N,N,*) در واقع یک آرایه ۳ بعدی را تعریف می کند که دو ست را به هم مرتبط می کند (در این مثال تنها ست N استفاده شده است ولی به کل هر ست دیگر می تواند با دو پارامتر متفاوت استفاده شود).

☑ برای مثال دو پارامتر سوسکپتانسی 2.5 Sus و حد 0.3 LIMIT0 بین باس ۱ و ۲ بکار رفته است :
N1.N2

VARIABLES

متغیرها در GAMS و در این مثال بشکل زیر نمایش داده می شود:

VARIABLES

```
Z          objective function
p(G)       output power for generator G
d(N)       angle at bus N;
p.lo(G)=GDATA(G,'PMIN');
p.up(G)=GDATA(G,'PMAX');
d.fx('N3')=0;
```

☑ در اینجا متغیر Z تابع هدف را نشان می دهد و همیشه باید وجود داشته باشد.

☑ مقدار حدود بالایی و پایینی را با استفاده از پسوند (Suffix) تعریف کرد. مثلا در این مثال مقدار حدود بالا و پایین متغیر P(G)، با استفاده از پسوندهای lo و up تعریف شده است.

☑ متغیرهای همچنان می توانند مقادیر ثابت را بگیرند. برای مثال متغیر d با استفاده از دستور بالا در مقدار صفر ثابت شده است. (یعنی در واقع زاویه مرجع باس صفر در نظر گرفته شده است)

☑ در GAMS انواع متفاوت متغیرها وجود دارد :

Binary - Integer - Free - Negative - Positive

✓ در مثال UC، می توان متغیر $P(G)$ را مثبت تعریف کرده و حدود آن را به شکل زیر تنظیم نمود:

```

POSITIVE VARIABLE p(G) ;
p.lo(G)=GDATA(G,'PMIN');
p.up(G)=GDATA(G,'PMAx');

```

✓ متغیرها همچنین می توانند مقادیر اولیه بگیرند. برای مثال مقدار توان اولیه باس ۱ به حالت زیر دارای

مقدار اولیه می شود :

```
p.l('G1')=0.2;
```

EQUATIONS

در این قسمت توابع هدف و حدود را وارد خواهیم کرد:

EQUATIONS

```

COST                objective function
MAXPOW(N,N)        maximum line power limit
MINPOW(N,N)        line power limit
LOADBAL(N)         load balance equation;
ALIAS(N,NP);
COST .. z=e=SUM(G,GDATA(G,'COST')*p(G));
MAXPOW(N,NP) .. LDATA(N,NP,'SUS')*(d(N)-d(NP))=l=LDATA(N,NP,'LIMIT');
MINPOW(N,NP) .. LDATA(N,NP,'SUS')*(d(N)-d(NP))=g=-LDATA(N,NP,'LIMIT');
LOADBAL(N) .. SUM(G$MAP(G,N),p(G))+SUM(NP,LDATA(N,NP,'SUS')*(d(N)-d(NP))+
LDATA(NP,N,'SUS')*(d(N)-d(NP)))=e=LOAD(N);

```

✓ از مثال بالا مشخص است که هر قید ابتدا باید شناسایی شده و بعد از شناسایی با '...' به قید و

معادله مربوطه لینک می شود. بعد از هر معادله قید گذاشتن سیمیکالون الزامی است.

✓ زیگما یک عبارت مثل $\sum_i x_{ij}$ در GAMS به شکل $sum(i,x(i,j))$ تعریف می شود.

✓ نشانه های معادلات از قبیل مساوی (=e=) و کوچکتر مساوی (=l=) تعریف می شوند.

✓ برای مثال فرم ریاضی سطر تابع هزینه ($z = \sum (G, GDATA(G, 'COST') * p(G))$; COST ..

شکل زیر است:

$$z = \sum_{i=1}^n \{P_i C_i (P_i)\}$$

✓ در حل این مثال ما محدوده قدرت انتقالی را در نظر گرفتیم. حدپایین نشان دهنده حداقل توان خروجی

ژنراتورها و حد بالا نشان دهنده حداکثر توان خروجی ژنراتورها است که می توانستیم به حالت زیر نمایش

دهیم:

$$P_i^{\min} \leq P_i \leq P_i^{\max}$$

✓ توان انتقالی خطوط انتقال بوسیله حدود پایداری و حرارتی (زاویه) محدود می شوند. یعنی توانی که از

باس i به باس j انتقال می یابد (با خط $i-j$) را می توان به حالت زیر نشان داد:

$$B_{ij} (\delta_i - \delta_j)$$

بنابراین داریم :

$$P_i^{\min} \leq B_{ij} (\delta_i - \delta_j) \leq P_i^{\max}$$

در GAMS :

MAXPOW (N, NP) .. LDATA (N, NP, 'SUS') * (d (N) -d (NP)) =l= LDATA (N, NP, 'LIMIT') ;

MINPOW (N, NP) .. LDATA (N, NP, 'SUS') * (d (N) -d (NP)) =g=-LDATA (N, NP, 'LIMIT') ;

✓ معادله توازن توان به شکل زیر نوشته می شود:

$$P_i = D_i - \sum B_{ij} (\delta_i - \delta_j)$$

که در GAMS به حالت زیر نوشته می شود :

LOADBAL (N) .. SUM (G\$ MAP (G, N) , p (G)) +SUM (NP, LDATA (N, NP, 'SUS') * (d (N) -d (NP)) +
LDATA (NP, N, 'SUS') * (d (N) -d (NP))) =e=LOAD (N) ;

عامل $\sum B_{ij} (\delta_i - \delta_j)$ نشان دهنده تلفات می باشد.

MODEL

این دستور معادلاتی را که در داخل مدل بایستی حل شوند را تعریف می کند. برای این مثال ED ما تابع هزینه،

نامعاده حداکثر توان، نامعاده حداقل توان و معادله توازن بار را داریم:

```
MODEL ed /COST,MAXPOW,MINPOW,LOADBAL/;
```

☑ در واقع این قسمت یکی از ویژگی های منحصر به فرد GAMS می باشد که به کاربر اجازه می دهد که

حتی بتواند فقط یک معادله را در مدل حل کند. برای مثال اگر بخواهیم همین مثال ED را بدون در

نظر گرفتن حداکثر و حداقل توان حل کنیم داریم:

```
MODEL ed /COST,LOADBAL/;
```

SOLVE

این دستور برای حل مسئله بهینه سازی بکار می رود. در مثال ED داریم:

```
SOLVE ed USING lp MINIMIZING z;
```

☑ این جمله به GAMS می گوید که مسئله توزیع اقتصادی ED را با استفاده از حل کننده خطی lp در

راستای حداقل کردن متغیر z (همان تابع هزینه) حل بکن.

☑ حل کننده های متفاوتی در GAMS وجود دارند که با دستور option می توان آنها را تغییر داد.

DISPLAY

```
DISPLAY p.l, d.l;
```

☑ با این دستور در واقع توان خروجی ژنراتورها و زاویه هر باس در این مثال نشان داده می شود. پسوند l

نشان دهنده سطح یا level هر متغیر می باشد.

☑ بعد از حل مثال ED نتایج زیر حاصل می شود:

```
---- 42 VARIABLE p.L output power for generator      G1 0.565, G2 0.285
---- 42 VARIABLE d.L angle at bus N                  N1 -0.143, N2 -0.117
```

شکل زیر برنامه نوشته شده را در محیط GAMS نشان می دهد :

```

gamsdir: C:\Users\Mas\Documents\gamsdir\projdir\gmsproj.gpr - [C:\Users\Mas\Documents\gamsdir\projdir\Untitled_6.gms]
File Edit Search Windows Utilities Model Libraries Help
LP
Untitled_6.gms | Untitled_6.lst

SETS
G      index of generators      /G1*G2/
N      index of buses          /N1*N3/
MAP(G,N) indicates in what bus are the generators connected /G1.N1,G2.N2/;

Scalar
fcost fuel cost in dollars_MBTu /1.1/;

PARAMETER
LOAD(N) load at bus N / N3 0.85 /;

TABLE GDATA(G,*) generator input data
*      FMIN      FMAX      COST
      (kW)      (kW)      (dollar/kWh)
G1     0.15      0.6       6
G2     0.10      0.4       7;

TABLE LDATA(N,N,*) line input data
*      SUS      LIMIT
      (S)      (kW)
N1.N2  2.5      0.3
N1.N3  3.5      0.5
N2.N3  3.0      0.4;

VARIABLES
z      objective function
p(G)  output power for generator G
d(N)  angle at bus N;
p.lo(G)=GDATA(G,'FMIN');
p.up(G)=GDATA(G,'FMAX');
d.fx('N3')=0;

VARIABLE P(G);
p.lo(G)=GDATA(G,'FMIN');
p.up(G)=GDATA(G,'FMAX');

```

بعد از حل برنامه توسط GAMS نتایج زیر بدست می آید:

```

gamsdir: C:\Users\Mas\Documents\gamsdir\projdir\gmsproj.gpr - [C:\Users\Mas\Documents\gamsdir\projdir\Untitled_6.lst]
File Edit Search Windows Utilities Model Libraries Help
LP
Untitled_6.gms | Untitled_6.lst

Compilation
Equation Listing SOLVE ed Using LP From line 51
Equation
Column Listing SOLVE ed Using LP From line 51
Column
Model Statistics SOLVE ed Using LP From line 51
Solution Report SOLVE ed Using LP From line 51
SOLVING
SOLVAR
Execution
Display
p
d

0 INFEASIBLE
0 UNBOUNDED
GAMS Rev 232 WEX-WEI 23.2.1 x86_64/MS Windows 02/13/14 18:56:35 Page 6
General Algebraic Modeling System
Execution

---- 53 VARIABLE p.L output power for generator G
G1 0.565, G2 0.285

---- 53 VARIABLE d.L angle at bus N
N1 -0.143, N2 -0.117

EXECUTION TIME = 0.000 SECONDS 3 Mb WEX232-232 Aug 11, 2009

USER: GAMS Development Corporation, Washington, DC G871201/0000CA-ANY
Free Demo, 202-342-0180, sales@gams.com, www.gams.com DC0000

**** FILE SUMMARY

Input C:\Users\Mas\Documents\gamsdir\projdir\Untitled_6.gms
Output C:\Users\Mas\Documents\gamsdir\projdir\Untitled_6.lst

```

مثال شماره ۲ :

UNIT COMMITMENT

همان مثال ED را بکار می گیریم و UC را در سه ساعت در نظر می گیریم. $t_1=150$, $t_2=500$, $t_3=400$

تقاضاها هستند و ژنراتورها دارای مقادیر رزرو زیر هستند. واحد ۱=۱۵، واحد ۲=۵۰، واحد ۳=۴۰. بطور کلی

اطلاعات زیر را در مسئله داریم:

Generator #	1	2	3
P_{max}	350	200	140
P_{min}	50	80	40
Rampup limit	200	100	100
Rampdown limit	300	150	100
Fixed Cost	5	7	6
Startup Cost	20	18	5
Shutdown Cost	0.5	0.3	1.0
Variable Cost	0.100	0.125	0.150

برنامه به حالت زیر نوشته می شود :

SETS

K index of periods of time /1*4/

J index of generators /1*3/

TABLE	GDATA (J,*)		generator input data					
	PMIN	PMAX	T	S	A	B	C	E
*			Ramp	Ramp	Fixed	Variable	Start	Shutdown
*			Down	Up	Cost	Cost	UP	Cost
*			Limit	Limit				Cost
*	(kW)	(kW)	(kW/h)	(kW/h)	(E)	(E)	(E)	(E/kWh)
1	50	350	300	200	5	20	0.5	0.100
2	80	200	150	100	7	18	0.3	0.125
3	40	140	100	100	6	5	1.0	0.150;

TABLE	PDATA (K,*)		data per period	
	D	R		
*	Load	Reserve		
*	(kW)	(kW)		

2	150	15
3	500	50
4	400	40;

VARIABLES

z	objective function variable
p(J,K)	output power of generator j at period k
v(J,K)	is equal to 1 if generator j is committed in period k
y(J,K)	is equal to 1 if generator j is started-up at the beginning of period k
s(J,K)	is equal to 1 if generator j is shut-down in period k;

:VARIABLE INITIALIZATION AND TYPE

POSITIVE VARIABLES p(J,K);

** Status decisions are modeled using binary variables.

BINARY VARIABLES v(J,K), y(J,K), s(J,K);

** Initial values are stated for some variables.

v.fx(J,'1')=0;

p.fx(J,'1')=0;

EQUATIONS

COST	objective function
PMAXLIM(J,K)	maximum output power equation
PMINLIM(J,K)	minimum output power equation
LOAD(K)	load balance equation
RESERVE(K)	spinning reserve equation
LOGIC(J,K)	start-up shut-down and running logic
RAMPUP(J,K)	maximum up ramp rate limit
RAMPDOWN(J,K)	maximum down ramp rate limit;

** The objective function is an equality equation. The remaining constraints are defined for all periods K, except for the initial one. To model this exception the \$(ord(K) GT 1) condition is included.

```
COST .. z=e=SUM((K,J),GDATA(J,'A')*v(J,K)+GDATA(J,'B')*y(J,K)+GDATA
(J,'C')*s(J,K)+GDATA(J,'D')*p(J,K));
```

```
PMAXLIM(J,K)$ (ord(K) GT 1) .. p(J,K) =l= GDATA(J,'PMAX') * v(J,K) ;
```

```
PMINLIM(J,K)$ (ord(K) GT 1) .. p(J,K) =g= GDATA(J,'PMIN') * v(J,K) ;
```

```

LOAD (K)$ (ord(K) GT 1) .. SUM (J, p(J,K))=e=PDATA (K, 'D' );
RESERVE (K )$(ord(K) GT 1) .. SUM (J,GDATA (J, ' PMAX' ) *
v (J, K) )=g=PDATA (K, ' D' ) +PDATA (K, ' R' ) ;
LOGIC (J,K)$ (ord(K) GT 1) .. y(J,K) -s(J,K) =e=v (J, K) -v (J, K-1) ;
RAMPUP (J,K)$ (ord(K) GT 1) .. p (J, K) -p (J, K-1) =l=GDATA (J, ' S' ) ;
RAMPDOWN (J,K)$ (ord(K) GT 1) .. p (J, K-1) -p (J, K) =l=GDATA (J, ' T' ) ;

```

MODEL

```
MODEL uc /ALL/;
```

نکته! هنگامی که all بکار می رود هدف حل uc با در نظر گرفتن تمامی قیود است ولی فرضا اگر

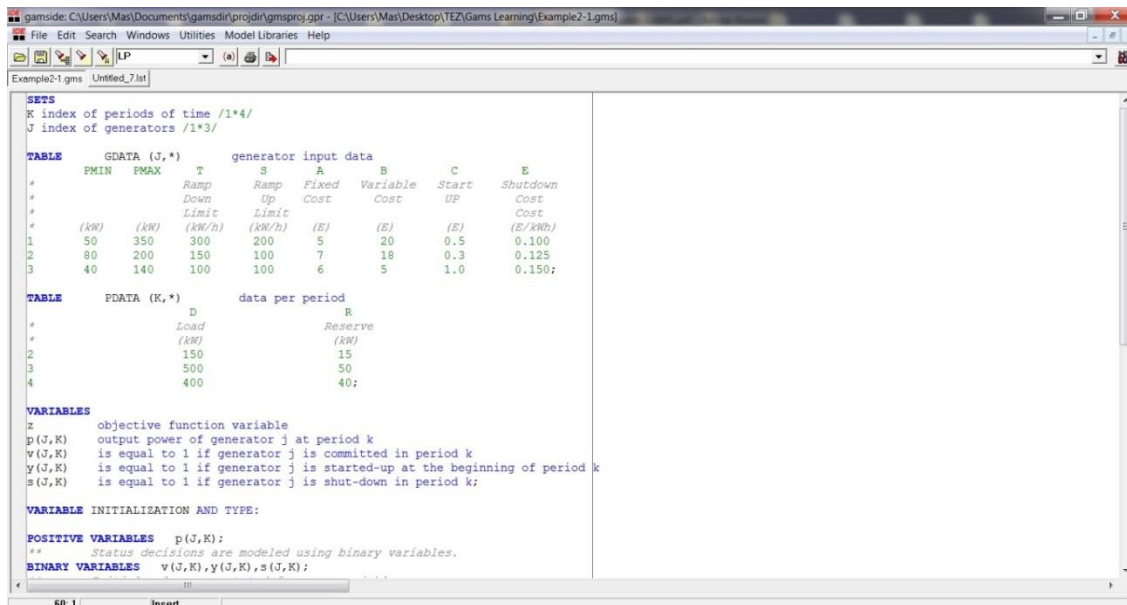
بخواییم uc را با در نظر گرفتن همه قیود جز قید رزرو حل بکنیم باید بنویسیم:

```
MODEL uc /COST, PMAXLIM, PMINLIM, LOAD, LOGIC/;
```

SOLVE

```
SOLVE uc USING MIP MINIMIZING z;
```

شکل زیر برنامه نوشته شده را در محیط GAMS نشان می دهد :



خروجی GAMS و حل مثال :

* . با در نظر گرفتن تمامی قیود :

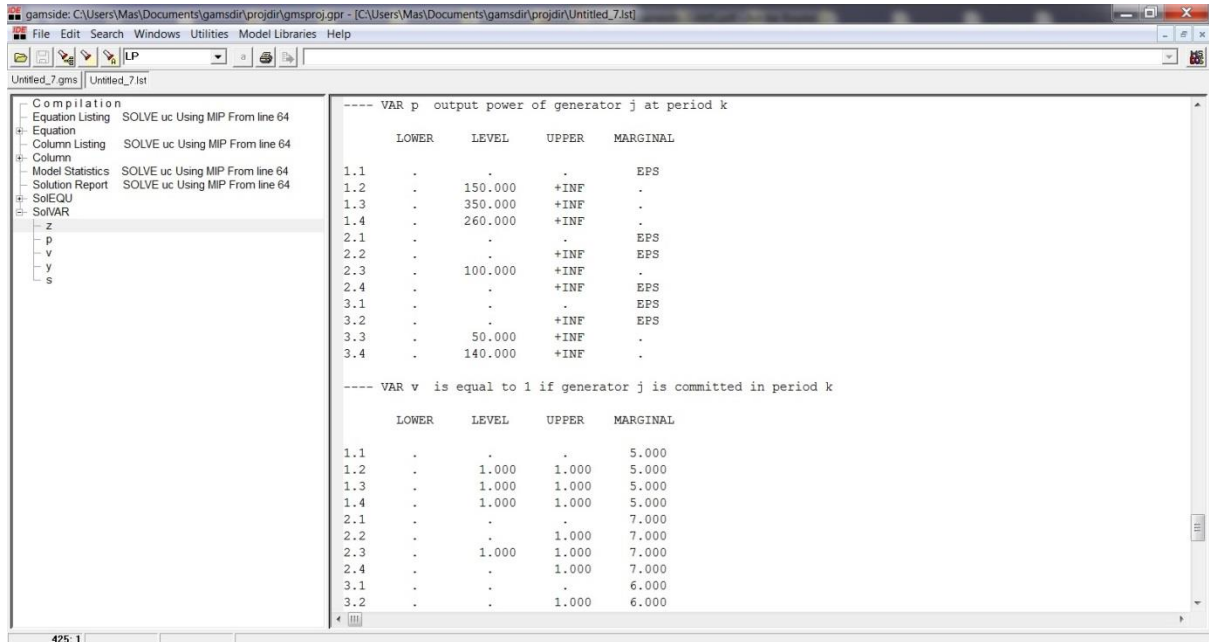
```

---- 88 VARIABLE p.L output power of generator j at period k
                2          3          4
1          150.000      350.000      260.000
2
3          50.000       100.000
                50.000       140.000

---- 88 VARIABLE z.L      =      77.300 objective function variable

```

در این حالت بعد از حل برنامه توسط GAMS نتایج زیر بدست می آید:



Compilation SOLVE uc Using MIP From line 64
Equation Listing SOLVE uc Using MIP From line 64
Column Listing SOLVE uc Using MIP From line 64
Column
Model Statistics SOLVE uc Using MIP From line 64
Solution Report SOLVE uc Using MIP From line 64
SolEQU
SolVAR

z
p
v
y
s

```

---- VAR p output power of generator j at period k
      LOWER    LEVEL    UPPER    MARGINAL
1.1 .          .          .          EPS
1.2 .          150.000    +INF      .
1.3 .          350.000    +INF      .
1.4 .          260.000    +INF      .
2.1 .          .          .          EPS
2.2 .          .          +INF      EPS
2.3 .          100.000    +INF      .
2.4 .          .          +INF      EPS
3.1 .          .          .          EPS
3.2 .          .          +INF      EPS
3.3 .          50.000    +INF      .
3.4 .          140.000    +INF      .

---- VAR v is equal to 1 if generator j is committed in period k
      LOWER    LEVEL    UPPER    MARGINAL
1.1 .          .          .          5.000
1.2 .          1.000    1.000    5.000
1.3 .          1.000    1.000    5.000
1.4 .          1.000    1.000    5.000
2.1 .          .          .          7.000
2.2 .          .          1.000    7.000
2.3 .          1.000    1.000    7.000
2.4 .          .          1.000    7.000
3.1 .          .          .          6.000
3.2 .          .          1.000    6.000
  
```

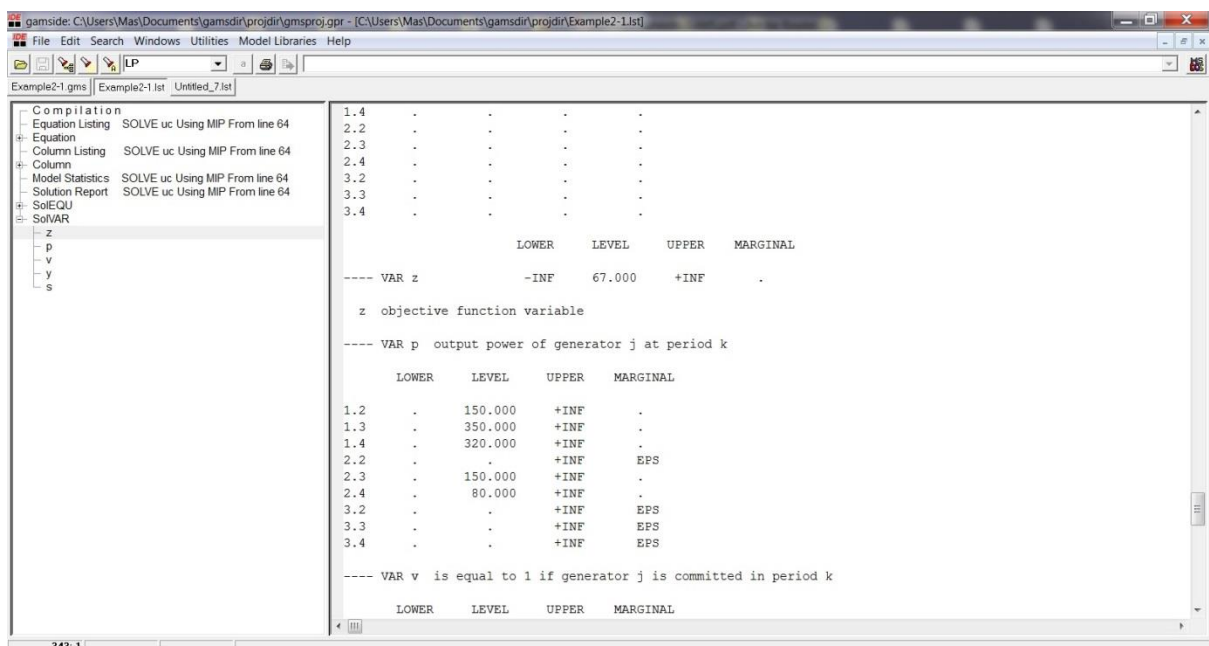
425: 1

*. با در نظر گرفتن همه قیود جز قید رزرو :

```

----- 87 VARIABLE p.L output power of generator j at period k
              2          3          4
1          150.000      350.000      320.000
2          150.000      150.000      80.000
----- 87 VARIABLE z.L = 67.000 objective function variable
  
```

در این حالت بعد از حل برنامه توسط GAMS نتایج زیر بدست می آید:



Compilation SOLVE uc Using MIP From line 64
Equation Listing SOLVE uc Using MIP From line 64
Column Listing SOLVE uc Using MIP From line 64
Column
Model Statistics SOLVE uc Using MIP From line 64
Solution Report SOLVE uc Using MIP From line 64
SolEQU
SolVAR

z
p
v
y
s

```

1.4 .          .          .          .
2.2 .          .          .          .
2.3 .          .          .          .
2.4 .          .          .          .
3.2 .          .          .          .
3.3 .          .          .          .
3.4 .          .          .          .

      LOWER    LEVEL    UPPER    MARGINAL
---- VAR z          -INF    67.000    +INF      .
z objective function variable

---- VAR p output power of generator j at period k
      LOWER    LEVEL    UPPER    MARGINAL
1.2 .          150.000    +INF      .
1.3 .          350.000    +INF      .
1.4 .          320.000    +INF      .
2.2 .          .          +INF      EPS
2.3 .          150.000    +INF      .
2.4 .          80.000    +INF      .
3.2 .          .          +INF      EPS
3.3 .          .          +INF      EPS
3.4 .          .          +INF      EPS

---- VAR v is equal to 1 if generator j is committed in period k
      LOWER    LEVEL    UPPER    MARGINAL
  
```

343: 1

با تشکر از زحمات اساتید راهنما

مخصوصا دکتر ربیعی و محمدی ایواتلو

Mohseni@znu.ac.ir