GAMS TUTORIALS FOR BEGINNERS (IN persian)

Data · May 2016

DOI: 10.13140/RG.2.1.1086.0408

2 authors:

Saman Nikkhah
Amirkabir University of Technology
9 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Abbas Rabiee
University of Zanjan
63 PUBLICATIONS **961** CITATIONS
SEE PROFILE

IN GOD WE TRUST

آموزش کاربردی نرم افزار GAMS

تهیه کننده: سامان نیک‌خواه

با همکاری : دکتر عباس ربیعی

بهار ۱۳۹۵

مقدمه

تکنولوژی GAMS یک سیستم جامع طراحی شده بری تسهیل کار مدل‌سازانی است که پدیده‌های واقعی را به مدل ریاضی (به صورت ویژه مدل‌های بهینه سازی) تبدیل می‌کنند. این سیستم مدل‌سازان را از طراحی آلگوریتم‌های مختلف برای پیاده‌سازی مدل خود خلاص می‌کند. GAMS مخفف "General Algebraic Modeling System" است. لازم به ذکر است برخلاف نام این نرم افزار، فقط برای مدل های ریاضی با معادلات جبری استفاده نمی‌شود. با توجه به اینکه برای حل معادلات دیفرانسیل در مشتقات جزئی نیاز به انتقال این معادلات به سیستم معادلات جبری می‌باشد که این معادلات با توجه به روش‌های تکرار حل می‌شوند، می‌توان گفته که GAMS کاربرد وسیعی دارد. این نرم افزار یک سیستم قوی و انعطاف پذیر برای حل مسائل برنامه‌ریزی خطی (LP) برنامه‌ریزی غیرخطی (NLP) برنامه‌ریزی صحیح مختلط (MIP) برنامه‌ریزی خطی صحیح مختلط (MINLP) و مسائل مکمل خطی (MCP) می-باشد. در این آموزش، مدل های کوچک جهت آشنایی با این نرم افزار و پیاده سازی مدل های ریاضی در محیط نرم‌افزار نمایش داده خواهد شد. جهت آشنایی بهتر با کاربرد GAMS مثال‌های مختلفی حل خواهند شد. توصیه می‌شود جهت یادگیری بهتر، مثال‌هایی که جهت آموزش ارائه می‌شود را مرحله به مرحله در محیط نرم‌افزار پیاده نمایید.

نکاتی که قبل از استفاده و بهره برداری از GAMS باید بدانید:

- لازم است درک کاملی از مدل ریاضی خود قبل از پیاده سازی در نرم‌افزار GAMS داشته باشید.
- GAMS نمی‌تواند برای شما فکر کند. بلکه کمک می‌کند مدل خود را حل کنید.

- اگر به هر دلیلی نتایج گرفته شده از GAMS منطقی نمی‌باشد، مشکل از نرم‌افزار نیست. بلکه لازم است مدل خود را به دقت چک کنید. اگر نتوانستید خطا را پیدا کنید، پس از یک استراحت کامل کار خود را تا یافتن ایراد مدل ادامه دهید.
- قیودی که مدل را محدود می‌کنند، نقش مهمی در بهینه شدن جواب دارن. لذا در انتخاب و پیاده‌سازی محدودیت‌ها دقت نمایید.

نکات کلیدی نرم افزار GAMS:

- بخش‌های مختلف مدل پیاده سازی شده در نرم‌افزار با " **;** " جدا می‌شوند.
- حروف کوچک و بزرگ در محیط نرم‌افزار فرقی ندارند.
- دو روش برای درج توضیح در برنامه وجود دارد:
  ۱. استفاده از " * " در ابتدای جمله.
  ۲. شروع توضیحات با **$ontext** و خاتمه دادن با **$offtext**
- برای رفتن به خط بعدی می‌توان از کلید Enter استفاده کرد.

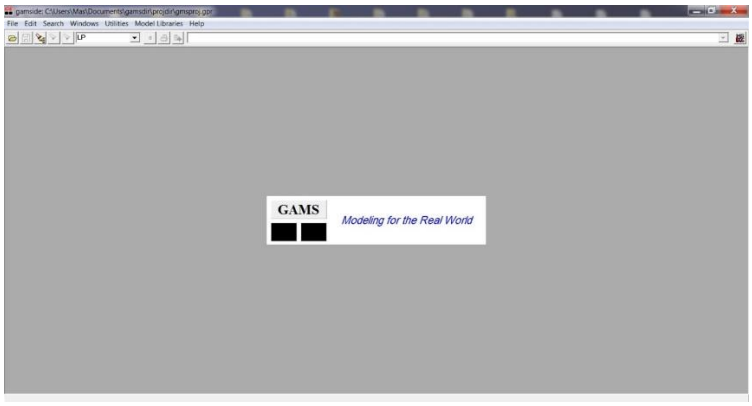
نصب و راه‌اندازی نرم‌افزار GAMS

نصب نرم افزار GAMS همانند بسیاری از برنامه‌ها به راحتی انجام می‌گیرد. مشکلی که در استفاده از این نرم افزار وجود دارد ثبت کردن نرم‌افزار یا همان License برنامه می‌باشد. با توجه به فراوانی لایسنس‌های موجود در سایت‌ها، پس از دانلود فایل لایسنس از مسیر File->Options->Licenses و انتخاب فایل text دانلود شده برنامه را ثبت نمایید. از مسیر File->Options->Solvers وضعیت Solver_های ثبت شده با توجه به لایسنس قابل مشاهد است. در این مسیر در صورتی که لایسنس استفاده شده یک Solver را پوشش دهد جلوی نام آن Full نوشته شده و در صورتی که پوشش ندهد Demo نوشته شده است. همچنین، در این قسمت پوشش هر Solver روی نوع مسائل مشخص شده است. برای مثال یک برای Solver با نام

CONOPT قسمت مربوط به مسائل NLP علامت زده شده است، که نشان دهنده قابلیت این Solver برای حل مسائل NLP می‌باشد.

آشنایی با محیط و کلیدهای نوار ابزار نرم‌افزار:

پس از بازکردن نرم‌افزار محیط آن به صورت شکل (۱) می‌باشد. برای ایجاد یک مدل می‌توان از کلید ترکیبی Ctrl+N استفاده کرد.



شکل (۱): محیط نرم‌افزار GAMS

در قسمت نوارابزار این نرم‌افزار کلیدهای کاربردی وجود دارد که در شکل (۲) علامت گذاری شده‌اند و به صورت زیر کارایی آن‌ها تعریف می‌شود:

۱. این کلید برای بازکردن کدهای ذخیره شده داخل کامپیوتر استفاده می‌شود.
۲. این کلید برای ذخیره کد نوشته شده استفاده می‌شود.
۳. این کلیدها برای جستجوی متن داخل کد استفاده می‌شوند که در کدهای با حجم بالا می‌توان بیشتر از این کلیدها برای پیدا کردن قسمتهای موردنظر در کد از آن‌ها استفاده نمود.

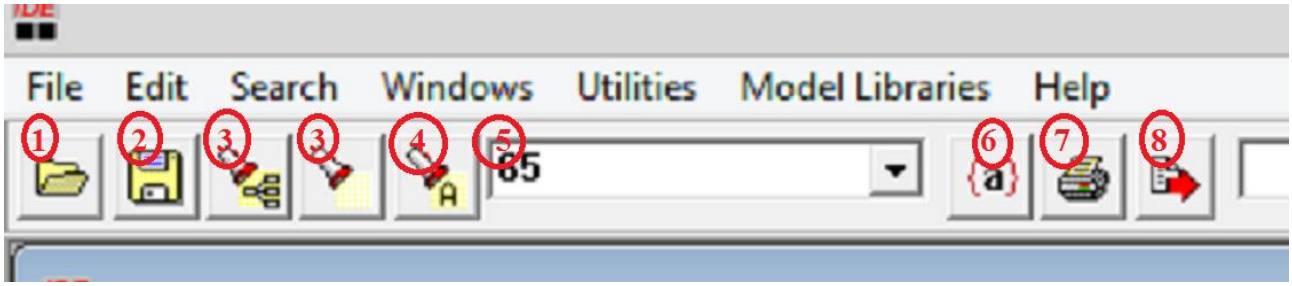
۴. این کلید برای جستجوی مجدد متن جستجو شده در قسمت ۳ استفاده می‌شود.

۵. متن و قسمتی از کد که نیاز به جستجو دارد را می‌توان در این قسمت نوشت.

۶. این کلید یکی از کلیدهای مهم و پرکاربرد نرم‌افزار GAMS می‌باشد. به دلیل اینکه خطای مربوط به پرانتز یک خطای رایج است، این کلید برای بررسی باز و بسته بودن پرانتز استفاده می‌شود.

۷. این کلید برای چاپ کد نوشته شده استفاده می‌شود.

۸. این کلید برای ران برنامه نوشته شده استفاده می‌شود.



شکل (۲): نوار ابزار نرم‌افزار GAMS

دستورات GAMS

در این نرم‌افزار دستورات مختلفی وجود دارد که در ادامه جهت آشنایی با هر دستور و کاربرد آن از یک مثال ساده استفاده می‌شود.

EX1): فرض کنید برای تولید انرژی در یک سیستم قدرت از دو منبع انرژی تجدید پذیر بادی و خورشیدی استفاده شود. برای راه اندازی هر یک از این نیروگاه‌ها نیاز به سرمایه‌گذاری خاصی است که میزان آن در جدول (۱) نشان داده شده است. همچنین، قیمت فروش انرژی هر نیروگاه در جدول نشان داده شده است. هدف بیشینه کردن درآمد حاصل از فروش انرژی این دو منبع تولید انرژی می‌باشد. (داده‌های این جدول فرضی و جهت آشنایی با نرم‌افزار است)

نوع انرژی تجدیدپذیر	هزینه‌ها		قیمت فروش انرژی
	هزینه تعمیرات و بهره-برداری ($)	هزینه سرمایه‌گذاری ($)	($/MW)
بادی	۲	۴	۱۲
خورشیدی	۳	۲	۸
هزینه در دسترس	۱۰۰	۸۰	

قدم اول: به دست آوردن مدل مسئله

قدم اول در حل مسئله، بدست آوردن مدل مسئله با توجه به داده‌های جدول (۱) می‌باشد. در صورتی که مزارع بادی را با C_1 و مزارع خورشیدی را با C_2 و هدف مسئله را با OF نشان دهیم مدل این مسئله را به صورت معادلات (۱) تا (۴) پیاده‌سازی می‌شود.

$$OF = 12C_1 + 8C_2 \quad (1)$$

$$2C_1 + 3C_2 \leq 100 \quad (2)$$

$$4C_1 + 2C_2 \leq 80 \quad (3)$$

$$C_1 , C_2 \geq 0 \quad (4)$$

قدم دوم: تعریف مجموعه‌ها

پس از بدست آوردن مدل مسئله، پیاده سازی آن در محیط نرم‌افزار شروع می‌شود. قدم اول در این راه تعریف مجموعه‌ها (Sets) است. مجموعه ها یکی از عناصر اصلی یک مدل GAMS هستند که مترادف

اندیس‌ها در نمایش جبری مدل‌ها هستند. مجموعه‌ها یکی از آیتم‌های عمده در GAMS می‌باشند. مجموعه-ها می‌توانند به عنوان ارتباطی بین متغیرها و معادلات مدل تعریف شوند. ایجاد یک مجموعه به صورت زیر انجام می‌شود:

```
Set
نام مجموعه  توضیح در مورد مجموعه /اعضای مجموعه/
For example:
Set
T     month     /Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec/ ;
```

که مجموعه T نامیده می‌شود. کلمه month توضیحی در مورد این مجموعه می‌باشد. و مقادیر بین " // " شاخص‌ها یا مقادیر مجموعه می‌باشند. به عنوان مثال Sep نهمین عنصر این مجموعه است.

برای مسئله بالا مجموعه‌ها به صورت زیر تعریف می‌شوند:

```
Set
R   Renewable energy sources   /r1, r2/ ;
```

قدم سوم: تعریف داده‌های معین

در GAMS سه روش اساسی برای ورود داده‌ها و مقداردهی وجود دارد:

۱. Scalar
۲. Parameter
۳. Table

۱. در صورتی که مقداردهی به صورت ثابت باشد از Scalar استفاده می‌شود. به عنوان مثال در صورتی که بخواهیم ظرفیت یک مزرعه بادی را مشخص کنیم به صورت زیر عمل می‌کنیم:

```
Scalar    W_Cap                Capacity of wind farm (MW)   /1200/ ;
```

که W_Cap نام انتخاب شده برای Scalar است. جمله قرمز رنگ توضیح در مورد Scalar و عدد داخل // “ ” ظرفیت اختصاص داده شده است.

۲. در این نوع مقدار دهی، مجموعه های تعریف شده نقش مهمی را ایفا می‌کنند. برای مثال در صورتی که بخواهیم ماه‌های قسمت Set را مقدار دهی کنیم به صورت زیر عمل می‌کنیم:

```
Parameter
Pw (t)              Penetrated wind power to the system (MW)
/
Jan        238
Feb        284
Mar        315
Apr        445
May        1100
Jun        985
Jul        457
Aug        895
Sep        745
Oct        100
Nov        564
```

```
Dec       963
/
;
```

لازم است بدانید که:

- نحوه چیدمان به هر شکلی می‌تواند باشد. فقط باید کل لیست داخل // قرار گیرد.
- در تایپ کردن مجموعه‌ها دقت کنید زیرا اشتباه وارد کردن یک حرف موجب ایجاد خطا خواهد شد.
- صفر، مقدار پیش فرض تمام پارامترها است.

۳. روش سوم ورود داده‌ها توسط جداول است. بسیاری از داده‌ها در مسائل بزرگ توسط جداول وارد می‌شوند. به عنوان مثال در صورتی که بخواهیم فاصله بین شهرها را به صورت جدول نشان دهیم به صورت زیر عمل می‌نماییم:

```
Table  d (i, j)       Distance between cities
                    Karaj          Sanandaj          Urmia
Zanjan              4              7                 8
;
```

دستور بالا ابتدا پارامتر d را روی دامنه‌های i و j تعریف می‌کند و سپس مقادیر آن را مشخص می‌کند. i و j مجموعه‌هایی هستند که در بخش Sets تعریف شده‌اند. به عنوان مثال فاصله زنجان تا کرج 4 است.

در مورد مثال بالا به صورت زیر عمل می‌کنیم:

```
Parameters
S(R)    Profit
/
R1      12
R2       8
/
OandM(R)      Operation and maintenance cost
/
R1      2
R2      3
/
In(R)           Investment cost
/
R1     4
R2     2
/
;
```

قدم چهارم: تعیین متغیرها

در GAMS پنج نوع متغیر در مدل قابل تعریف است:

1. Variables (قابل تعریف روی مقادیر حقیقی)

2. Positive variables (قابل تعریف روی مقادیر حقیقی مثبت)
3. Negative variables (قابل تعریف روی مقادیر حقیقی منفی)
4. Integer variables (قابل تعریف روی اعداد صحیح)
5. Binary variables (قابل تعریف روی اعداد ۰ و ۱)

متغیرهای مثال بالا را می‌توان به صورت زیر تعریف نمود:

```
Variables
C(r)        Renewable energy sources
OF          Objective function
;
```

البته با توجه به مثبت بودن C(r) بهتر است که آن را به صورت متغیر مثبت تعریف نماییم.

```
Positive variables
C(r)        Renewable energy sources
;
```

قدم پنجم: تعیین معادلات

معادلات برای تعیین تابع هدف و محدودیت‌ها کاربرد دارند. معادلات در دو قسمت در GAMS تعریف می‌شوند. (۱) برچسب نام معادلات. برای هر معادله یک برچسب مشخص می‌شود که با آن برچسب شناخته می‌شود. (۲) تعریف ساختار معادله. در این مرحله ساختار معادله‌ای که در بخش قبلی آن را مشخص نموده‌ایم پیاده سازی می‌شود. به عنوان مثال در مثال بالا به صورت زیر عمل می‌کنیم:

```
Equations
Objective       Objective function of problem
```

O&P Operation and maintenance cost

Invest Investment cost

;

در این مرحله برای هر معادله یک برچسب مشخص کرده‌ایم. برچسب‌های نوشته شده در قسمت ساختار معادلات باید به صورتی نوشته شود که در قسمت برچسب نوشته شد است. حال ساختار معادلات را به صورت زیر مشخص می‌کنیم:

Equation name .. expression1 = operator = expression2 ;

ابتدا نام معادلات را داریم. نام معادلات در قسمت برچسب معادلات باید موجود باشد. سپس دو فاصله با .. نام معادلات را از مدل ریاضی معادلات جدا می‌کند. پیاده‌سازی مدل جبری معادلات به صورت دستورات موجود در GAMS در قسمت‌های expression انجام می‌پذیرد. تعدادی توابع استاندارد و علائم ریاضی GAMS به صورت زیر هستند:

GAMS name	شرح
+	جمع
–	تفریق
/	تقسیم
*	ضرب
ABC	قدر مطلق
COS	کسینوس
EXP	تابع نمایی
LOG	لگاریتم طبیعی

لگاریتم در مبنای ۱۰	$LOG10$
توان دوم	SQR
ریشه دوم	$SQRT$
$\sum_{i\in I}$	$SUM\ (I\ ,)$
$\sum_{i\in I}\sum_{j\in J}$	$SUM\ (\ (I\ ,J\ )\ ,)$
x به توان y که y باید عدد صحیح باشد.	$Power\ (x,y)$
x به توان y که y میتواند هر عددی باشد	$rPower\ (x,y)$

معادلات GAMS باید با روابط ریاضی که در قسمت OPERATOR قرار می‌گیرد تعیین شوند. این روابط به صورت زیر تعریف می‌شوند:

=	= **E** =
≥	= **G** =
≤	= **L** =

و در پایان هر معادله باید با یک سیمی‌کالون (;) خاتمه یابد.

حال معادلات مسئله بالا به صورت زیر تعریف می‌شوند:

```
Objective     ..    OF=e=sum ( i , S(r) * c(r))    ;

O&P           ..    sum (i, OandM(r)* c(r) )   =l=100   ;

Invest        ..    sum (i, In(r) * c(r) )   =l=80     ;
```

قدم ششم: تعریف مدل

در دستور مدل بعد از وارد کردن نام مدل باید لیستی از نام معادلات را بین " // " بنویسیم. در صورتی که بخواهیم کل معادلات حل شود All مینویسیم.

Model نام مدل /all/;

این دستور زائد است اما برای کاربران پیشرفته که چند مدل را در یک بار اجرا کردن GAMS ایجاد می کنند کارایی دارد اگر ما بخواهیم به جای عبارت فوق بصورت مستقیم نام معادلات را وارد کنیم دستور به شکل زیر خواهد.

Model نام مدل /eq1, eq2, …/;

برای مسئله بالا مدل به صورت زیر تعریف می‌شود:

```
Model    cost     / all / ;
```

قدم هفتم: دستور حل

پس از مشخص کردن مدل دستور حل مدل به صورت زیر وارد می‌شود:

Solve نام مدل using روش حل ;

```
Solve   cost   using   LP   maximizing OF ;
```

اجزای این دستور عبارتند از:

۱. کلید واژه Solve

۲. نام مدلی که باید حل شود

۳. کلید واژه using

۴. انتخاب یک روش حل

۵. هدف حل مدل

۶. متغیری که باید هدف حل روی آن پیاده شود

روش‌های حل عبارتند از:

LP: برای برنامه ریزی خطی

NLP: برای برنامه ریزی غیرخطی

MIP: برای برنامه ریزی عدد صحیح مختلط

RMIP: برای برنامه ریزی عدد صحیح مختلط آزاد شده خطی

MINLP: برای برنامه ریزی عدد صحیح مختلط غیرخطی

RMINLP: برای برنامه ریزی عدد صحیح مختلط غیرخطی

MCP: برای مسائل مکمل مختلط

CNS: برای دستگاه محدود غیر خطی

قدم هشتم: دستور نمایش

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

این دستور بصورت زیر می باشد:

```
Display C.l ;
```

در استفاده از این دستور لازم است بدانید که:

- زمانی که بخواهید مقدار نهایی یک متغیر را بدست بیاورید در جلو پارامتر m و l قرار می‌دهیم. به بیان دیگر، برایش نمایش مقدار پایانی یک متغیر از (c.l) و برای نمایش مقدار مرزی از (c.m) استفاده می‌شود.
- برای نمایش مقادیر پارامترها نیاز به m و l نیست و فقط کافی است جلو دستور نمایش نام پارامتر را وارد نمایید. بنابراین برای پیدا کردن مقدار یک پارامتر به صورت جدا نیاز نیست معادله مربوط به پارامتر را در قسمت معادلات بنویسید. بلکه فقط کافی است پس از وارد کردن دستور حل داخل کد معادله مربوط به پارامتر را نوشته و از دستور حل به صورت گفته شده استفاده نمود.

در این مثال با دستورهای کاربردی و مهم GAMS آشنا شدیم. در ادامه دستورهای کلیدی و نکات مهم گفته خواهد شد. برای آشنایی بیشتر با مدل‌های کوچک ادامه می‌دهیم و مقادیر بدست آمده را با مداری که از حل دستی بدست آورده‌ایم مقایسه می‌کنیم.

EX2: یکی از مثال‌های کاربردی که برای بیان مفاهیم بهینه سازی استفاده می‌شود مدل ساده زیر است. در این مدل که دارای یک قید تساوی است، ابتدا حل را با روش لاگرانژ بدست آورده و با مقدار بدست آمده از نتایج پیاده سازی در محیط نرم‌افزار GAMS مقایسه میکنیم.

$$\begin{aligned} &\min imize \;\; 0.25x_1^2 + x_2^2 \\ &subject\ to \\ &5 - x_1 - x_2 = 0 \end{aligned}$$

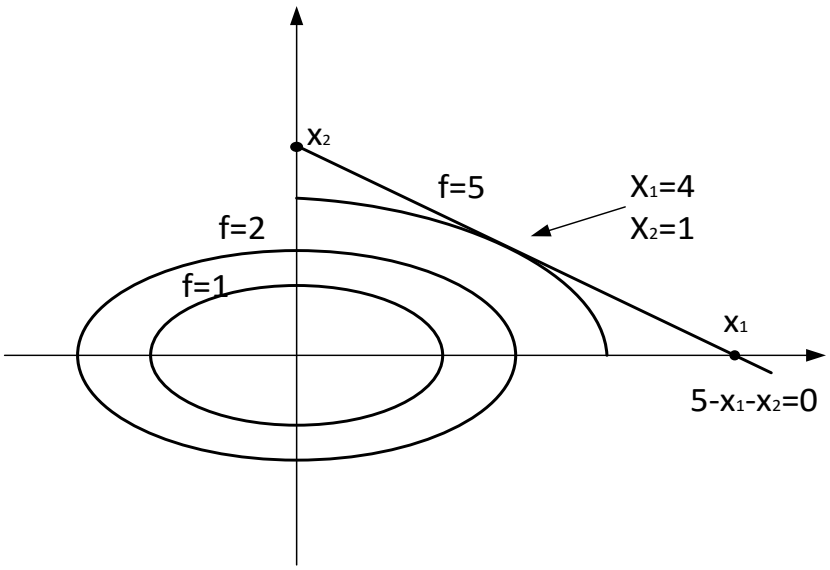
در این مسئله، تابع هدف که یک بیضی است با توجه به قید تساوی داده شده کمینه می‌شود. ابتدا با استفاده از روش لاگرانژ مقدار نهایی را به صورت زیر بدست می‌آوریم:

$$\ell(x_1, x_2, \lambda) = 0.25x_1^2 + x_2^2 + \lambda(5 - x_1 - x_2)$$
$$\frac{\partial \ell}{\partial x_1} = 0.5x_1 - \lambda = 0$$
$$\frac{\partial \ell}{\partial x_2} = 2x_2 - \lambda = 0$$
$$\frac{\partial \ell}{\partial \lambda} = 5 - x_1 - x_2$$

که در نهایت جواب‌های بدست آمده برابر است با:

$$x_1 = 4$$
$$x_2 = 1$$
$$\lambda = 2$$

یعنی جواب بدست آمده در صورتی بهینه است که معادله خط داده شده که مربوط به قید مسئله است بر معادله بیضی مماس باشد. این مسئله را می‌توان در شکل (۳) مشاهده نمود.



شکل (۳): نقطه بهینه با توجه به تابع هدف

حال این نرم‌افزار در محیط *GAMS* پیاده می‌کنیم که به صورت زیر می‌باشد:

```
*******In God We Trust*****
SETS

N     NUMBER OF VARIABLES /1,2/;

parameter a(n)
/
1   0.25
2   1
/;

parameter b(n)
/
1    1
2    1
/;


VARIABLE
X(n)
T
;
EQUATIONS
OF
GX
;
OF .. T=E=sum(n, a(n)*sqr(x(n))) ;
GX .. 5-(sum(n,b(n)*x(n)))=E=0;

MODEL GT /ALL/;

SOLVE GT USING NLP MINIMIZING T;
```

پس از اجرای برنامه نتایج بدست آمده در زیر نشان داده شده است.

مقدار تابع هدف:

```
**** SOLVER STATUS     1 Normal Completion
**** MODEL STATUS      2 Locally Optimal
**** OBJECTIVE VALUE                5.0000
```

مقدار متغیرها:

	LOWER	LEVEL	UPPER	MARGINAL
1	-INF	4.000	+INF	.
2	-INF	1.000	+INF	EPS

مشاهده می‌شود که این مقادیر با مقادیر بدست آمده از حل دستی برابر است. پس می‌توان گفت که جواب‌هایی که از نرم‌افزار *GAMS* بدست می‌آیند قابل اعتماد و درست هستند. بنابراین در صورتی که جوابی که از مدلتان گرفته‌اید درست نبود مشکل از نرم‌افزار نیست، بلکه مدل شما مشکل دارد و نیاز به دقت بیشتر می‌باشد.

EX3: برای بررسی بیشتر مدل بهینه سازی ساده زیر را در نظر بگیرید. تفاوت این مثال با مثال قبلی در قید آن می‌باشد که قیدها نامساوی هستند. این مسئله از یک تابع هدف و دو قید تشکیل شده است که قیود مسئله قیود نامساوی هستند. جواب‌های بدست آمده برای x_1 و x_2 به ترتیب برابر $\frac{5}{3}$ و $\frac{1}{3}$ است.

$$\min imize \quad (x_1-2)^2+2(x_2-2)^2$$
$$subject\ to$$
$$x_1+4x_2\le 3$$
$$x_1\ge x_2$$

مدل کامل این مسئله در محیط نرم‌افزار در زیر نشان داده شده است.

```
*******In God We Trust*****
SETS

N     NUMBER OF VARIABLES /1,2/;

parameter a(n)
/
1   1
2   2
/;

parameter b(n)
/
1    2
2    1
/;

parameter c(n)
/
1    1
2    4
/;

parameter d(n)
/
1    1
2    -1
/;


VARIABLE
X(n)
T
;


EQUATIONS
OF
GX
FX
;
OF  ..  T=E=SUM  (N,A(N)*SQR(X(N)-B(N)))  ;
GX  ..  sum(n,c(n)*x(n))=l=3;
FX  ..  sum(n,D(n)*x(n))=G=0;

MODEL GT /ALL/;

*Option NLP = SNOPT;

SOLVE GT USING NLP MINIMIZING T;
```

جواب‌های بدست آمده برای متغیرهای این مسئله نیز در زیر نشان داده شده است:

```
---- VAR X

     LOWER      LEVEL      UPPER    MARGINAL

1     -INF      1.667       +INF        .
2     -INF      0.333       +INF        .
```

همانطوری که مشاهده می‌شود، جواب‌های بدست آمده دقیق و قابل اعتماد هستند. مثال‌های بالا برای یادگیری اولیه نرم‌افزار مناسب هستند. حال در مثال زیر نکات کلیدی و مهم در *GAMS* را دنبال می‌کنیم.

EX4: در این مثال نکات کلیدی و حرفه‌ای در نرم‌افزار GAMS نوشته خواهد شد. توصیه می‌شود با دقت این مثال را دنبال کرده و آن را پیاده نمایید و نتایج را مشاهده کنید.

مدل مسئله که در زیر نشان داده شده است، مقید به یک قید تساوی است. در این مدل متغیر x دارای دو اندیس i و t است. i تعداد متغیرها و t تغییر هر متغیر در ۱۲ ساعت را نشان می‌دهد. ضریب $A(i,t)$ که در هر ساعت و برای متغیرهای مختلف تغییر می‌کند در جدول A قابل مشاهده است.

$$OF = \sum_{i,t} A(i,t).x(i,t)^2$$

$$S.T$$

$$\sum_{i,t} x(i,t) = b$$

جدول A: ضریب $A(i,t)$

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12
i1	5	2	7	8	8	7	5	7	7	4	7	9
i2	4	5	8	3	3	2	4	5	5	3	5	2
i3	2	6	5	1	2	5	3	6	2	1	6	1

در این مثال سعی می‌شود دستورات زیر معرفی و پیاده شوند:

۱. اپراتور $ (اپراتور شرطی)
۲. حلقه (LOOP)
۳. دستور شرطی IF-ELSE
۴. دستور ALIAS

برای پیاده کردن مدل سه مجموعه را تعریف کرده و همچنین ضریب $A(i,t)$ را تعریف می‌نماییم:

```
*******In God We Trust*****
SETS

T     TIME VALUES           /T1*T12/
i     NUMBER OF VARIABLES /1*3/
s     NUMBER OF SCENARIOS /s1*s5/;


TABLE a(i,T)

   T1   T2   T3   T4   T5   T6   T7   T8   T9   T10  T11  T12
1  5    2    7    8    8    7    5    7    7    4    7    9
2  4    5    8    3    3    2    4    5    5    3    5    2
3  2    6    5    1    2    5    3    6    2    1    6    1
;
```

هدف از تعریف اندیس s برای تغییر دادن b در چند تکرار و بدست آوردن مقدار تابع هدف است. فرض می‌شود مقدار b در ۵ سناریو با مقادیر زیر تغییر کند.

```
PARAMETER LAMDA(S)
/
s1    5
s2    6
s3    9
s4    10
s5    15
/;
```

دقت شود ابتدا مسئله را در حالت عادی و وقتی که b برابر مقدار ثابت ۵ باشد حل می‌کنیم. سپس در یک حلقه مقدار b را تغییر داده و تابع هدف را بدست می‌آوریم. پس b را یک بار به صورت مقدار ثابت (*scalar*) و یک بار به صورت (*parameter*) به ترتیب برای حل حالت عادی و حلقه تعریف می‌کنیم. معادلات و متغیرها همانند مثال‌های قبل تعریف می‌شوند.

```
*SCALAR  SS /5/;

PARAMETER SS;


VARIABLE
X(I,T)
TOT

;

*-----------------------------------------------------------

EQUATIONS
OF
GX

;
OF              .. TOT=E=SUM ((I,T),A(I,T)*SQR(X(I,T))) ;

GX              .. sum((I,T),x(I,T))=E=SS;

*-----------------------------------------------------------
MODEL GT /ALL/;


*Option NLP = SNOPT;

SOLVE GT USING NLP MINIMIZING TOT;
```

عبارات شرطی:

۱. اپراتور شرطی $

اپراتور شرطی که به علامت $ مشخص می‌شود، یکی از مشخصه‌های قدرتمند GAMS است. این اپراتور به شرایط منطقی عمل می‌نماید. علامت $ را می‌توان به صورت زیر خواند:

/در صورتی که شرط (منطقی) صحیح است/

برای آشنایی مثال ساده زیر را در نظر بگیرید:

$$\text{If } (b>1.5) \ \textit{then} \ \ a=2$$

این عبارت را در GAMS به صورت زیر می‌توان مدل نمود:

$$a\,\$(b>1.5)=2;$$

حال در این مسئله مقدار پارامتر d را برای ساعات ۵ به بعد به صورت زیر می‌توان بدست آورد. در این روش پس از دستور حل یک پارامتر را نوشته و مقدار d را با توجه به معادله آن بدست می‌آوریم.

```
SOLVE GT USING NLP MINIMIZING TOT;

PARAMETER D(T) ;
D(T) $ ((ORD(T))>5)=SUM(I,X.L(I,T));

DISPLAY D;
```

در قسمت بالا ما یک پارامتر تعریف کردیم. تکرار می‌شود که نیاز نیست این پارامتر و معادله آن را در داخل معادلات نوشت. بلکه پس از دستور حل آن را تعریف نموده و مقدار آن را برای مقادیر نهایی متغیرها بدست می‌آوریم. پس از مشخص کردن پارامتر، فرمول آن که جمع(x(i,t روی i هست برای t_های بزرگتر از ۵ نوشته شده است.

- دستورهای CARD و ORD این دو دستور که از دستورات داخلی GAMS هستند برای مشخص کردن تعداد اعضای مجموعه و nامین عضو مجموعه استفاده می‌شوند. دستور CARD برای مشخص کردن تعداد اعضای مجموعه و دستور ORD برای مشخص کردن nامین عضو یک مجموعه استفاده می‌شوند. این دستورات که دستورات پرکاربردی هستند که در داخل حلقه می‌توان از آن‌ها بهره گرفت.

- دستور Alias یکی از دستورهای کاربردی GAMS است که برای نسبت دادن داده‌های یک مجموعه به یک حرف و یا نشانه دیگر است. برای مثال در این مسئله با دستور Alias(i,j) داده‌های مجموعه I را در مجموعه j تعریف می‌نماییم. به اصطلاح، برای یک مجموعه دو اسم مشخص کرده‌ایم.
- در ادامه دستور LOOP توضیح داده خواهد شد. در این قسمت فقط دستور LOOP مورد استفاده برای این مسئله را می‌نویسیم. در این حلقه b را هر بار برابر یکی از اعضای پارامتر LAMDA قرار داده و مسئله را

حل میکنیم. سپس مقادیر بدست آمده برای تابع هدف را در پنج تکرار نمایش می‌دهیم. این کار با استفاده از دستور display انجام می‌گیرد.

```
ALIAS (S,J);

parameter report1(j,*),report2(*);

loop(j,

SS=LAMDA(j);

SOLVE GT USING NLP MINIMIZING Tot;

report1(j,'ss')=ss;
report1(j,'objective')=Tot.l;

);


display report1 ;
```

۲. دستور شرطی IF-ELSE

اگر چه در نرم‌افزار GAMS دستورات شرطی را می‌توان با " $ " پیاده سازی کرد، ولی استفاده از دستور IF یک کد نوشته شده در GAMS را قابل فهم‌تر می‌نماید. با توجه به آشنایی با این دستور در این قسمت روش استفاده از آن در GAMS به صورت مختصر آورده شده است. دستور if در داخل GAMS به صورت زیر پیاده می‌شود.

if (*condition*,
statements;
{*elseif condition*, *statements*; }
[*else statements*;]);

در این مثال فرض می‌شود حداکثر مقداری که برای تابع هدف قابل قبول است برابر ۱۵ باشد. به این صورت با یک دستور شرطی در صورت بزرگتر بودن تابع هدف آن را محدود می‌کنیم. یعنی مسئله ابتدا یک بار در داخل حلقه حل می‌شود. سپس در صورت بزرگ بودن تابع هدف از مقدار ۱۵، b را از ۲ کم

می‌کنیم. پس از کم کردن مقدار b یک بار دیگر مسئله حل شده و نتایج داخل *report* که یک پارامتر است ذخیره شده و در نهایت مقدار تابع هدف در پنج تکرار انجام گرفته نمایش داده می‌شود.

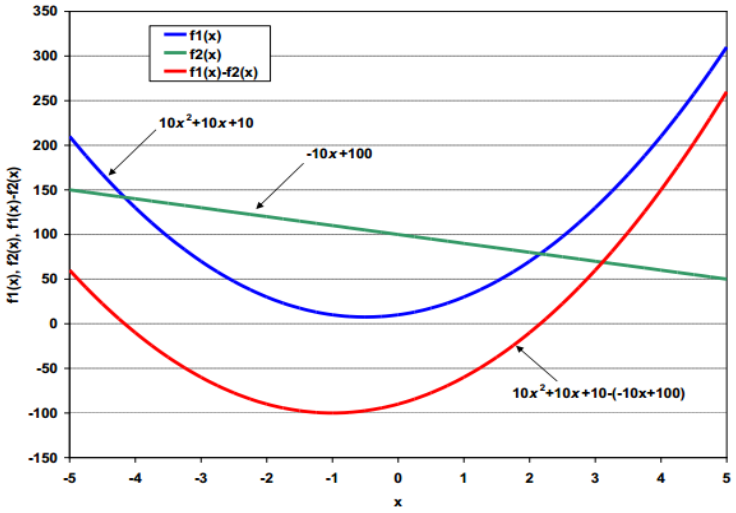
```
parameter report1(j,*),report2(*);

loop(j,

SS=LAMDA(j);

SOLVE GT USING NLP MINIMIZING Tot;

if(tot.l >=15,

ss=lamda(j)-2;

);

SOLVE GT USING NLP MINIMIZING Tot;

report1(j,'ss')=ss;
report1(j,'objective')=Tot.l;

);

display report1 ;
```

علاوه بر مسائل بهینه سازی، یکی دیگر از کاربردهای *GAMS* حل معادلات خطی است. در زیر یک مثال ساده جهت آشنایی با این موضوع نشان داده شده است.

EX5: حل یک معادله جبری. مسئله پیداکردن ریشه های معادله غیر خطی زیر است

$$(10X^{2}+10X+10)-(-10X+100)=0$$

ریشه‌های این معادله را می‌توان به صورت زیر نشان داد :



کد GAMS به صورت زیر نوشته می‌شود:

```
**IN GOD WE TRUST

VARIABLES x, e, obj;

EQUATIONS Eq1, Objective;

Eq1.. e =E= 10*x*x+10*x+10-(-10*x+100);

Objective.. obj =E= e*e;

MODEL Eq /ALL/

SOLVE Eq USING NLP MINIMIZING obj;
```

در ادامه برای آشنایی با کاربرد GAMS در سیستم‌های قدرت یک مثال کاربردی آورده می‌شود. در داخل مثال نیز، دستورات و نکات کلیدی کامل کننده این فایل آموزشی ارائه خواهند شد.

استفاده از نرم‌افزار GAMS در سیستم‌های قدرت

GAMS یکی از قویترین و مجهزترین برنامه‌هایی است که با دارا بودن انواع محدودیت‌های قابل اعمال بر روی متغیرهای مختلف مورد استفاده در علوم مختلف است. موارد کاربرد این نرم افزار را می‌توان به صورت زیر نشان داد:

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

پس می‌توان از نرم‌افزار GAMS در سیستم‌های قدرت استفاده نمود. جهت پوشش کامل استفاده از این نرم‌افزار مثال زیر را در نظر بگیرید:

EX6) همانطوری که میدانید پخش بار اقتصادی دینامیکی ((Dynamic economic dispatch (DED)، مدل گسترش یافته مسائل پخش بار اقتصادی است که قیود تولید واحدهای نیروگاهی به آن اضافه شده است. در این مسئله، هدف کمینه کردن هزینه واحدهای نیروگاهی در یک مسئله DED در ۱۰تکرار مونت‌کارلو است. سه واحد نیروگاهی بار را تغذیه می‌کنند.

همانطور که در قسمت‌های اول گفته شد، قدم اول تعریف کردن مجموعه‌ها است. مجموعه های این مسئله واحدهای نیروگاهی و ۲۴ ساعت تغییر بار است که به صورت زیر تعریف می‌شوند:

```
Sets
I        number of units        /G1*G3/
T        time intervals         /t1*t24/
```

علامت " * " در مجموعه ها برای مجموعه های متوالی استفاده می‌شود برای مثال در مجموعه اول G1 تا G3 مجموعه متوالی واحدهای نیروگاهی را نشان می‌دهد. در صورت متوالی نبودن مجموعه‌ها می‌توان آن‌ها را با " , " جدا کرد. برای مثال S1, S2, S5. مجموعه‌ها می‌توانند عدد نیز باشند. برای مثال مجموعه ساعت‌ها را می‌توان به صورت 1*24 تعریف کرد.

پس از تعیین کردن مجموعه‌ها، داده‌های معین مربوط به اطلاعات ژنراتورها و بارها را وارد می‌کنیم:

در قسمت داده‌های معین سه جدول قابل تعریف است.

۱. ضرایب تلفات واحدها

```
Table   Bloss(i,j)      B matrix loss coefficients
        G1          G2          G3
G1      0.000049    0.000014    0.000015
G2      0.000014    0.000045    0.000016
G3      0.000015    0.000016    0.000039  ;
```

۲. بار در ۲۴ ساعت

بار در ۲۴ ساعت را می‌توان به صورت یک پارامتر روی مجموعه ساعت‌ها تعریف نمود. با توجه به اینکه بار در هر ساعت در مسائل DED تغییر می‌کند پس به صورت جدول یا پارامتر روی مجموعه T می‌توان بار را تعریف نمود.

```
Parameter     PD(t)     Hourly load (MW)

/

t1        173

t2        82

t3        209

t4        292

t5        209

t6        73

t7        142

t8        116

t9        257

t10       187

t11       182

t12       260

t13       206

t14       354

t15       440

t16       265

t17       410

t18       304

t19       421

t20       366

t21       186
```

```
t22     117
t23     207
t24     242
/;
```

۳. اطلاعات واحدهای نیروگاهی و ضرایب هزینه

همانطور که می‌دانیم تولید توان هر نیروگاه حد بالا و پایینی دارد که در مسائل DED باید وارد شود. همچنین ضرایب هزینه و حداقل و حداکثر تولید که باید وارد شود.

```
Table     Units_Data(i,*)
          a          b          c          Pmin  Pmax  RU  RD
G1        0.00525    8.663      328.13     50    250   55  95
G2        0.00609    10.04      136.91     5     150   55  78
G3        0.00592    9.76       59.16      15    100   45  64
;
```

در تعریف این جدول نکته زیر قابل بیان است:

* در تعاریف جداول در صورتی که سطر یا ستونی با علامت * مشخص شود، محدودیتی در تعداد سطر و ستون وجود ندارد. برای مثال در جدول بالا Units_Data(i,*) ستون با * مشخص شده و تعداد ستونها محدودیتی ندارد و برچسب هر ستون در قسمت مجموعه‌ها نیز تعریف نشده است.

پس از تعریف کردن داده‌ها، متغیرهای مسئله تعریف شده و کران بالا و پایین انرژی تولیدی واحدهای نیروگاهی مشخص خواهد شد.

```
Variables
P(i,t)          production of unit i
Cost            min cost
;
Positive variable
Ploss(t)
hourly_cost(t)
u_cost(i,t)
;
```

در تعریف کردن متغیرها باید دقت شود. برای مثال از آنجایی که هر واحد توانی را تولید می‌کند پس P باید روی I که مجموعه ژنراتورها است تعریف شود. با توجه به تغییر بار در ۲۴ ساعت، لازم است توان تولیدی ژنراتورها در هر ساعت تغییر کند. پس P باید روی T نیز تعریف شود. در نتیجه متغیر P به صورت P(i,t) تعریف می‌شود.

برای تعریف کردن حدود متغیرهای مسئله می‌توان از دستورات زیر در GAMS استفاده کرد:

```
P.lo(i,t)  =    Units_Data(i,'Pmin') ;
P.up(i,t)  =    Units_Data(i,'Pmax') ;
```

دستورات UP، LO به ترتیب برای کران های بالا و پایین و دستور FX برای فیکس کردن مقدار یک متغیر در مقادیر دلخواه استفاده می‌شود. بعنوان مثال در این مثال حد توان تولیدی ژنراتورها با UP و LO محدود می‌شود.

با توجه به بررسی خط به خط نرم افزار GAMS و با توجه به اینکه در این مسئله هدف تعریف بار با توزیع نرمال است باید قبل از معادلات بار را نیز به عنوان پارامتر تعریف نمود. از آنجایی که این نرم افزار خط به خط کدهای نوشته شده را چک می‌کند، در صورتی که قبل از معادلات تعریف نشود برنامه فرمان خطا صادر خواهد داد.

```
Parameter                    Load(t) ;
```

پس از تعریف متغیرها و قیود، معادلات مسئله همانطور که در قسمت‌های قبل گفته شد نوشته خواهند شد.

```
Equations
costeq
hourly_cost_eq
powerbalance
losscalculation
RU_const
RD_const
;
costeq                .. cost =e= sum(t , hourly_cost(t) );
hourly_cost_eq(t)     ..  hourly_cost(t) =e= sum( i, Units_Data(i,'a') * power( P(i,t),2) +
                              Units_Data(i,'b') * P(i,t) + Units_Data(i,'c') );
powerbalance(t)       ..   sum(i, p(i,t)) =e= Load(t) + Ploss(t) ;
losscalculation(t)    .. Ploss(t)=e=sum( (i,j) , P(i,t) * Bloss(i,j) * P(j,t));
RU_const( i, t)       .. P(i,t+1) - P(i,t) =l= Units_Data(i,'RU') ;
```

```
RD_const( i, t)        .. P(i,t-1) - P(i,t) =l= Units_Data(i,'RD') ;
```

پس از تعریف نمودن معادلات، مدل مسئله تعریف می‌شود.

```
Model        DED_Test1        /all/        ;
```

پس از مشخص کردن مدل، می‌توان Solver مورد نظر برای حل مسئله را با دستور Option مشخص نمود. در مسائل سیستم های قدرت سه Solver کاربردی زیر برای بدست آوردن مسائل NLP پیشنهاد می‌شود.

1. CONOPT
2. SNOPT
3. IPOPT

```
Option          NLP=   SNOPT;
```

✓ **LOOP**

❊. استفاده از دستور loop در GAMS متداول تر از حلقه‌های شناخته شده همچون for است. اگر چه حلقه for در زبان برنامه‌نویسی GAMS تعریف شده است ولی استفاده از دستور LOOP متداول‌تر و کاربردی‌تر است. برای آشنایی با این دستور در این مثال آن را توضیح می‌دهیم. با توجه به اینکه روش مونت‌کارلو بر اساس تکرارها متوالی تا رسیدن به جواب بهینه می‌باشد، لازم است این تکرارها را با یکی از دستورهای loop و یا for پیاده سازی کنیم.

برای استفاده از دستور loop ابتدا باید تعداد تکرارها را مشخص کنیم. این کار با تعریف یک مجموعه با تعداد اعضای برابر با تعداد تکرارهای مونت‌کارلو انجام می‌شود. هدف از تعریف این مجموعه تعیین تعداد تکرارهایی است که در داخل حلقه انجام می‌گیرد.

```
Set        counter        /c1*c10/    ;
```

برای ذخیره کردن متغیرها و پارامترهایی که در هر بار تکرار مقدارشان عوض می‌شود می‌توان یک پارامتر تعریف نمود. با استفاده از این پارمترها و بسته به تعداد اندیس‌های هر متغیر از یکی از پارامترها استفاده می‌شود.

```
Parameter    report1(counter,*),  report3(counter, t,*),   report2(*),
  report4(counter, i, t, *);
```

پارامترهای تعریف شده تحت عنوان report برای قرار دادن نتایج متغیرها پس از هر تکرار استفاده می‌شوند. بسته به اینکه هر متغیر روی چند مجموعه تعریف شده باشد، از یکی از چهار پارامتر نوشته شده استفاده می‌کنیم.

حال با نوشتن دستور loop روی مجموعه ای که برای تکرارها مشخص نموده ایم دستور حل را داخل این حلقه قرار می‌دهیم.

```
Loop (counter,
```

در داخل حلقه با مدل کردن بار شبکه با توزیع نرمال و قرار دادن دستور حل و مقادیر متغیرهایی که در هر تکرار درون reportها ذخیره می‌شود حلقه را می‌بندیم.

```
Load(t) = min( max( normal( PD(t) , PD(t) * 0.02 ) , sum(i, Units_Data(i,'Pmin') )), sum(i,
Units_Data(i,'Pmax') ) );


Solve  DED_Test1  using  NLP  minimizing  cost ;
```

در این مسئله هدف کمینه کردن هزینه در ۱۰ تکرار مونت‌کارلو با احتمالی در نظر گرفتن بار است که در دستور Solver ان را پیاده نموده‌ایم.

```
report1(counter,'Cost') = cost.l;
```

با توجه به اینکه Cost روی مجموعه تعریف نشده است لازم است که در هر تکرار مقدار آن ثبت شود. پس از report 1 برای ذخیره کردن اطلاعات استفاده می‌کنیم.

```
report1(counter,'status')= DED_Test1.modelstat ;
```

این دستور برای مشخص کردن حالت حل از لحاظ optimal بودن یا infeasible بودن استفاده می‌شود. با استفاده از این دستور می‌توان مشخص کرد که مسئله در هر تکرار جواب بهینه را بدست آورده است و یا جواب به دست نیامده است. این دستور برای هر حالت یک کد قرار می‌دهد. برای مثال در صورتی که مسئله در تکرار پنجم infeasible شود در تکرار پنجم کد ۴ را خواهیم دید.

```
report3(counter, t,'Hourly_cost')=hourly_cost.l(t);
```

با توجه به اینکه متغیر hourly_cost روی مجموعه t تعریف شده است لازم است از report3 استفاده شود که روی تعداد تکرارها و t تعریف شده است. کلمات قرمز رنگ نام متغیر ذخیره شده را مشخص میکند.

```
report4(counter, i, t,'Pgen')=p.l(i,t);

);
```

یکی از قیودی که در مسائل DED باعث infeasible شدن مسئله می‌شود حداقل و حداکثر تولید است که با تغییر دادن آن می‌توان مسئله را optimal کرد. با استفاده از دستور شرطی IF می‌توان مشخص نمود که

در صورت infeasible بودن مسئله در هر یک از تکرارها مقدار این قیود تغییر کند. این کار داخل حلقه تعریف شده در بالا به صورت زیر پیاده سازی می‌شود:

```
if  ((DED_Test1.modelstat eq 4),
Units_Data(i,'RU')= 1.5 * Units_Data(i,'RU');
```

در این دستور در صورت infeasible بودن مسئله مشخص نموده‌ایم که RU افزایش داده شود.

پس از تمام شدن حلقه می‌توان با دستور display نتایج report ها را مشاهده نمود.

```
Display     report1, report2, report3, report4;
```

جهت بهبود این فایل آموزشی با پیشنهادات سازنده خود ما را یاری فرمایید...

s.nikkhah@znu.ac.ir