

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/301736768>

GAMS TUTORIALS FOR BEGINNERS (IN persian)

Data · May 2016

DOI: 10.13140/RG.2.1.1086.0408

CITATIONS

0

READS

10,197

2 authors:



Saman Nikkhah

Amirkabir University of Technology

7 PUBLICATIONS 3 CITATIONS

[SEE PROFILE](#)



Abbas Rabiee

University of Zanjan

58 PUBLICATIONS 849 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Phd thesis [View project](#)



Distributed generation allocation and sizing in Alborz Distribution Networks [View project](#)



IN GOD WE TRUST

آموزش کاربردی نرم افزار **GAMS**

تهیه کننده: سامان نیک خواه

با همکاری: دکتر عباس ربیعی

بهار ۱۳۹۵

مقدمه

تکنولوژی GAMS یک سیستم جامع طراحی شده برای تسهیل کار مدل سازی است که پدیده های واقعی را به مدل ریاضی (به صورت ویژه مدل های بهینه سازی) تبدیل می کنند. این سیستم مدل سازان را از طراحی الگوریتم های مختلف برای پیاده سازی مدل خود خلاص می کند. 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 نمی شود. با توجه به اینکه برای حل معادلات دیفرانسیل در مشتقات جزئی نیاز به انتقال این معادلات به سیستم معادلات جبری می باشد که این معادلات با توجه به روش های تکرار حل می شوند، می توان گفته که GAMS کاربرد وسیعی دارد. این نرم افزار یک سیستم قوی و انعطاف پذیر برای حل مسائل برنامه ریزی خطی (LP) برنامه ریزی غیرخطی (NLP) برنامه ریزی صحیح مختلط (MIP) برنامه ریزی 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 نرم افزار نمایش داده خواهد شد. جهت آشنایی بهتر با کاربرد GAMS مثال های مختلفی حل خواهند شد. توصیه می شود جهت یادگیری بهتر، مثال هایی که جهت آموزش ارائه می شود را مرحله به مرحله در محیط نرم افزار 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 نرم افزار GAMS داشته باشید.
- GAMS نمی تواند برای شما فکر کند. بلکه کمک می کند مدل خود را حل کنید.

- اگر به هر دلیلی نتایج گرفته شده از GAMS منطقی نمی باشد، مشکل از نرم افزار 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 می کنند، نقش مهمی در بهینه شدن جواب دارن. لذا در انتخاب و پیاده سازی محدودیت ها دقت نمایید.

نکات کلیدی نرم افزار GAMS:

- بخش های مختلف مدل پیاده سازی شده در نرم افزار با ” ; ” جدا می شوند.
- حروف کوچک و بزرگ در محیط نرم افزار 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 می توان از کلید Enter استفاده کرد.

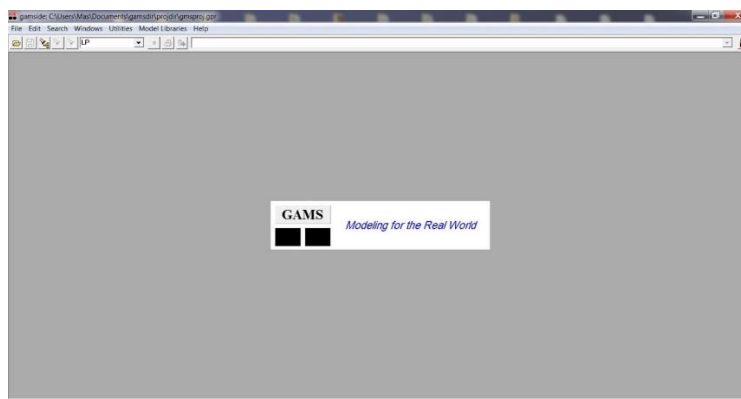
نصب و راه اندازی نرم افزار GAMS

نصب نرم افزار GAMS همانند بسیاری از برنامه ها به راحتی انجام می گیرد. مشکلی که در استفاده از این نرم افزار وجود دارد ثبت کردن نرم افزار یا همان License برنامه می باشد. با توجه به فراوانی لایسنس های موجود در سایت ها، 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 مشاهده 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 Solver با نام

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

آشنایی با محیط و کلیدهای نوار ابزار نرم افزار:

پس از باز کردن نرم افزار محیط آن به صورت شکل (۱) می باشد. برای ایجاد یک مدل می توان از کلید ترکیبی Ctrl+N استفاده کرد.



شکل (۱): محیط نرم افزار GAMS

در قسمت نوار ابزار این نرم افزار کلیدهای کاربردی وجود دارد که در شکل (۲) علامت گذاری شده اند و به صورت زیر کارایی آن ها تعریف می شود:

۱. این کلید برای باز کردن کدهای ذخیره شده داخل کامپیوتر استفاده می شود.
۲. این کلید برای ذخیره کد نوشته شده استفاده می شود.
۳. این کلیدها برای جستجوی متن داخل کد استفاده می شوند که در کدهای با حجم بالا می توان بیشتر از این کلیدها برای پیدا کردن قسمتهای موردنظر در کد از آن ها استفاده نمود.

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

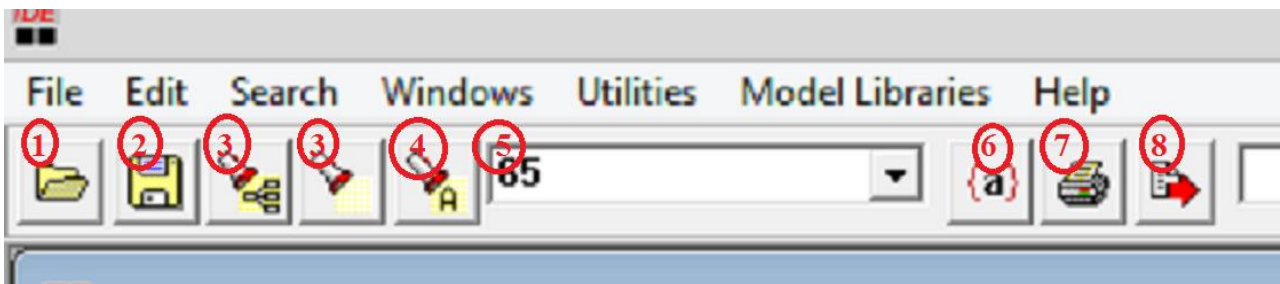
۵. متن و قسمتی از کد که نیاز به جستجو دارد را می توان در این قسمت نوشت.

۶. این کلید یکی از کلیدهای مهم و پرکاربرد نرم افزار GAMS می باشد. به دلیل اینکه خطای مربوط

به پرانتز یک خطای رایج است، این کلید برای بررسی باز و بسته بودن پرانتز استفاده می شود.

۷. این کلید برای چاپ کد نوشته شده استفاده می شود.

۸. این کلید برای ران برنامه نوشته شده استفاده می شود.



شکل (۲): نوار ابزار نرم افزار GAMS

دستورات GAMS

در این نرم افزار دستورات مختلفی وجود دارد که در ادامه جهت آشنایی با هر دستور و کاربرد آن از یک مثال ساده استفاده می شود.

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 نیروگاه ها نیاز به سرمایه گذاری 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 می باشد. (داده های این جدول فرضی و جهت آشنایی با نرم افزار است)

هزینه‌ها			
نوع انرژی تجدیدپذیر	هزینه تعمیرات و بهره-	هزینه سرمایه‌گذاری	قیمت فروش انرژی
	بردارای (\$)	(\$)	(\$/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. مجموعه‌ها 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

اعضای مجموعه / توضیح در مورد مجموعه نام 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 می شود. به عنوان مثال در صورتی که

بخواهیم ظرفیت یک مزرعه بادی را مشخص کنیم به صورت زیر عمل می کنیم:

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 می کنند. برای مثال در صورتی که

بخواهیم ماه های قسمت Set را مقدار دهی کنیم به صورت زیر عمل می کنیم:

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 می تواند باشد. فقط باید کل لیست داخل // قرار گیرد.
- در تایپ کردن مجموعه ها دقت کنید زیرا اشتباه وارد کردن یک حرف موجب ایجاد خطا خواهد شد.
- صفر، مقدار پیش فرض تمام پارامترها است.

۳. روش سوم ورود داده ها توسط جداول است. بسیاری از داده ها در مسائل بزرگ توسط جداول وارد می - شوند. به عنوان مثال در صورتی که بخواهیم فاصله بین شهرها را به صورت جدول نشان دهیم به صورت زیر عمل می نماییم:

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

;

دستور بالا ابتدا پارامتر d را روی دامنه های i و j تعریف می کند و سپس مقادیر آن را مشخص می کند. i و j مجموعه هایی هستند که در بخش Sets تعریف شده اند. به عنوان مثال فاصله زنجان تا کرج 4 است.

در مورد مثال بالا به صورت زیر عمل می کنیم:

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 می توان 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 محدودیت ها کاربرد دارند. معادلات در دو قسمت در GAMS تعریف می شوند. (۱) برچسب نام معادلات. برای هر معادله یک برچسب مشخص می شود که با آن برچسب شناخته می شود. (۲) تعریف ساختار معادله. در این مرحله ساختار معادله ای که در بخش قبلی آن را مشخص نموده ایم پیاده سازی می شود. به عنوان مثال در مثال بالا به صورت زیر عمل می کنیم:

Equations

Objective Objective function of problem

O&P Operation and maintenance cost

Invest Investment cost

;

در این مرحله برای هر معادله یک برچسب مشخص کرده ایم. برچسب های 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 می کنیم:

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 می کند. پیاده سازی مدل جبری معادلات به صورت دستورات موجود در GAMS در قسمت های expression انجام می پذیرد. 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

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

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

$$\begin{aligned}
 &= E = & &= \\
 &= G = & &\geq \\
 &= L = & &\leq
 \end{aligned}$$

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

حال معادلات مسئله بالا به صورت زیر تعریف می شوند:

$$\begin{aligned}
 \text{Objective} \quad &.. \quad OF=e=\text{sum} (i , S(r) * c(r)) \quad ; \\
 \text{O\&P} \quad &.. \quad \text{sum} (i, OandM(r)* c(r)) =l=100 \quad ; \\
 \text{Invest} \quad &.. \quad \text{sum} (i, In(r) * c(r)) =l=80 \quad ;
 \end{aligned}$$

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 می شود:

Model cost **/ all / ;**

قدم هفتم: دستور حل

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

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. 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 خروجی را جستجو کرد و یا می‌توانیم نمایش دادن این‌ها را از برنامه بخواهیم.

این دستور بصورت زیر می باشد:

Display C.1 ;

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 می دهیم. به بیان دیگر، برایش نمایش مقدار پایانی یک متغیر از (c.l) و برای نمایش مقدار مرزی از (c.m) استفاده می شود.
- 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 مدل های کوچک ادامه می دهیم و مقادیر بدست آمده را با مداری که از حل دستی بدست آورده ایم مقایسه می کنیم.

EX2: یکی از مثال های کاربردی که برای بیان مفاهیم بهینه سازی استفاده می شود 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 نرم افزار GAMS مقایسه میکنیم.

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

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:

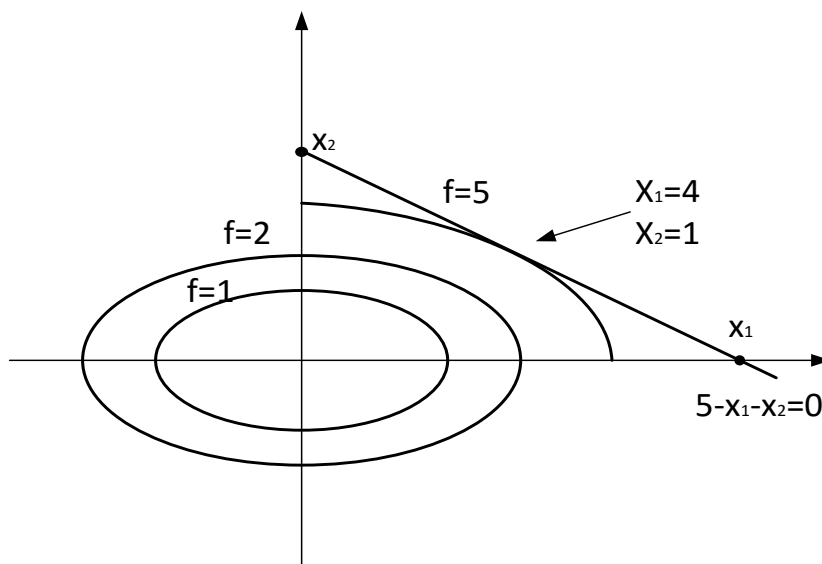
$$\begin{aligned}\ell(x_1, x_2, \lambda) &= 0.25x_1^2 + x_2^2 + \lambda(5 - x_1 - x_2) \\ \frac{\partial \ell}{\partial x_1} &= 0.5x_1 - \lambda = 0 \\ \frac{\partial \ell}{\partial x_2} &= 2x_2 - \lambda = 0 \\ \frac{\partial \ell}{\partial \lambda} &= 5 - x_1 - x_2\end{aligned}$$

که در نهایت جواب‌های بدست آمده برابر است با:

$$\begin{aligned}x_1 &= 4 \\ x_2 &= 1 \\ \lambda &= 2\end{aligned}$$

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 نرم افزار در محیط GAMS پیاده می کنیم که به صورت زیر می باشد:

```
*****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 می شود که این مقادیر با مقادیر بدست آمده از حل دستی برابر است. پس می توان گفت که جواب هایی که از نرم افزار GAMS بدست می آیند قابل اعتماد و درست هستند. بنابراین در صورتی که جوابی که از مدلتان گرفته اید درست نبود مشکل از نرم افزار نیست، بلکه مدل شما مشکل دارد و نیاز به دقت بیشتر می باشد.

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

$$\begin{aligned}
 &\text{minimize } (x_1 - 2)^2 + 2(x_2 - 2)^2 \\
 &\text{subject to} \\
 &x_1 + 4x_2 \leq 3 \\
 &x_1 \geq x_2
 \end{aligned}$$

مدل کامل این مسئله در محیط نرم افزار 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))=1=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 می شود، جوابهای بدست آمده دقیق و قابل اعتماد هستند. مثالهای بالا برای یادگیری اولیه نرم افزار مناسب هستند. حال در مثال زیر نکات کلیدی و مهم در GAMS را دنبال می کنیم.

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.

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

ST

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

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      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 می شود 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 می کنیم. سپس در یک حلقه مقدار b را تغییر داده و تابع هدف را بدست می آوریم. 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 می کنیم. معادلات و متغیرها همانند مثال های قبل تعریف می شوند.

```
*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 می شود، یکی از مشخصه های قدرتمند GAMS است. این اپراتور به شرایط منطقی عمل می نماید. علامت \$ را می توان به صورت زیر خواند:

ا در صورتی که شرط (منطقی) صحیح است /

برای آشنایی مثال ساده زیر را در نظر بگیرید:

If $(b > 1.5)$ then $a = 2$

این عبارت را در GAMS به صورت زیر می توان مدل نمود:

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

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

```
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 می شود 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 می آوریم. 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 n امین عضو مجموعه استفاده می شوند. دستور CARD برای مشخص کردن تعداد اعضای مجموعه و دستور ORD برای مشخص کردن n امین عضو یک مجموعه استفاده می شوند. این دستورات که دستورات کاربردی هستند که در داخل حلقه می توان از آن ها بهره گرفت.

- دستور Alias یکی از دستورهای کاربردی GAMS است که برای نسبت دادن داده های یک مجموعه به یک حرف و یا نشانه دیگر است. برای مثال در این مسئله با دستور $Alias(i,j)$ داده های مجموعه I را در مجموعه J تعریف می نماییم. به اصطلاح، برای یک مجموعه دو اسم مشخص کرده ایم.

- در ادامه دستور LOOP توضیح داده خواهد شد. در این قسمت فقط دستور LOOP مورد استفاده برای این مسئله را می نویسیم. 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 می دهیم. این کار با استفاده از دستور display انجام می گیرد.

```
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 نرم افزار GAMS دستورات شرطی را می توان با "\$" پیاده سازی کرد، ولی استفاده از دستور IF یک کد نوشته شده در GAMS را قابل فهم تر می نماید. 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 می شود.

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

در این مثال فرض می شود 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 می کنیم. یعنی مسئله ابتدا یک بار در داخل حلقه حل می شود. 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;
);

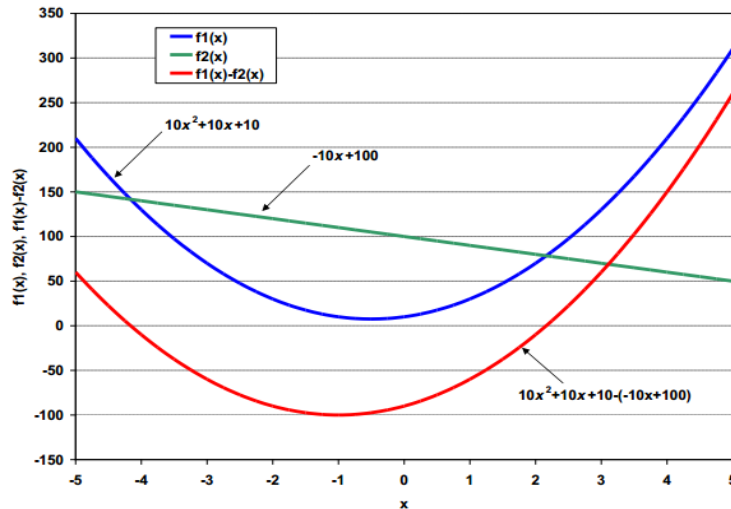
display report1 ;
```

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 پیدا کردن 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 می شود:

***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 سیستم های قدرت یک مثال کاربردی آورده می شود. در

داخل مثال نیز، دستورات و نکات کلیدی کامل کننده این فایل آموزشی ارائه خواهند شد.

استفاده از نرم افزار GAMS در سیستم های قدرت

GAMS یکی از قویترین و مجهزترین برنامه هایی است که با دارا بودن انواع محدودیت های قابل اعمال بر روی متغیرهای مختلف مورد استفاده در علوم مختلف است. موارد کاربرد این نرم افزار را می توان به صورت زیر نشان داد:

- اقتصاد کشاورزی
- مهندسی شیمی
- تجارت بین المللی
- اقتصاد خرد
- اقتصاد محیطی
- علوم مدیریتی
- امور مالی
- توسعه ی اقتصادی
- علوم نظامی
- انرژی
- حمل و نقل
- مهندسی
- ریاضی
- جنگل داری
- فیزیک

پس می توان از نرم افزار GAMS در سیستم های قدرت استفاده نمود. جهت پوشش کامل استفاده از این نرم افزار 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 ۱۰ تکرار مونت کارلو است. سه واحد نیروگاهی بار را تغذیه می کنند.

همانطور که در قسمت‌های اول گفته شد، قدم اول تعریف کردن مجموعه‌ها است. مجموعه‌های 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 مجموعه‌ها برای مجموعه‌های 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 می دانیم 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 مجموعه ها نیز تعریف نشده است.

پس از تعریف کردن داده ها، 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

$P_{loss}(t)$

hourly_cost(t)

$u_cost(i,t)$

;

در تعریف کردن متغیرها باید دقت شود. برای مثال از آنجایی که هر واحد توانی را تولید می کند 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 می شود.

برای تعریف کردن حدود متغیرهای مسئله می توان 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 می شود. بعنوان مثال در این مثال حد توان تولیدی ژنراتورها با UP و LO محدود می شود.

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 می کند، 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 قسمت های 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) = Units_Data(i,'RD') ;
```

پس از تعریف نمودن معادلات، مدل مسئله تعریف می شود.

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

پس از مشخص کردن مدل، می توان 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 می شود.

1. CONOPT
2. SNOPT
3. IPOPT

```
Option NLP= SNOPT;
```

LOOP ✓

✱ استفاده از دستور loop در GAMS متداول تر از حلقه های شناخته شده همچون for است. اگر چه حلقه

for در زبان برنامه نویسی GAMS تعریف شده است ولی استفاده از دستور LOOP متداول تر و کاربردی تر

است. برای آشنایی با این دستور در این مثال آن را توضیح می دهیم. با توجه به اینکه روش مونت کارلو بر

اساس تکرارها متوالی تا رسیدن به جواب بهینه می باشد، لازم است این تکرارها را با یکی از دستوره های 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 مونت کارلو انجام می شود. هدف از تعریف این مجموعه تعیین تعداد تکرارهایی

است که در داخل حلقه انجام می گیرد.

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

برای ذخیره کردن متغیرها و پارامترهایی که در هر بار تکرار مقدارشان عوض می شود می توان یک پارامتر تعریف نمود. با استفاده از این پارامترها و بسته به تعداد اندیس های هر متغیر از یکی از پارامترها استفاده می شود.

```
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 می شوند. بسته به اینکه هر متغیر روی چند مجموعه تعریف شده باشد، از یکی از چهار پارامتر نوشته شده استفاده می کنیم.

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

```
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 report ذخیره می شود حلقه را می بندیم.

```
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 مونت کارلو با احتمالی در نظر گرفتن بار است که در دستور Solver آن را پیاده نموده ایم.

```
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 1 report برای ذخیره کردن اطلاعات استفاده می کنیم.

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

این دستور برای مشخص کردن حالت حل از لحاظ optimal بودن یا infeasible بودن استفاده می شود. با استفاده از این دستور می توان 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 می دهد. 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 می شود حداقل و حداکثر تولید است که با تغییر دادن آن می توان مسئله را optimal کرد. با استفاده از دستور شرطی IF می توان 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 می شود:

```
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 نموده ایم که RU افزایش داده شود.

پس از تمام شدن حلقه می توان 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