

به نام یزدان پاک

Lingo Software

فهرست مطالب

- ✓ بخش اول : معرفی نرم افزار
- ✓ بخش دوم : آغاز کار با لینگو
- ✓ بخش سوم : حل مسائل

« بخش اول » معرفی نرم افزار

✓ مقدمه ای بر لینگو

✓ نرم افزارهای مشابه و ویژگی های لینگو

✓ لینگو چیست؟

مقدمه ای بر لینگو

بسته های نرم افزاری لیندو و لینگو توسط شرکت لیندو سیستمز^۱ برای حل مسائل بهینه سازی در دانشگاه، صنعت و تجارت، طراحی گردیده است. دو محصول فوق همراه با کتاب تحقیق در عملیات : کاربرد ها و الگوریتم ها (۱۹۹۴)^۲ و کتاب مقدمه ای بر برنامه ریزی ریاضی : کاربرد ها و الگوریتم ها (۱۹۹۵)^۳ که توسط پروفسور واین ال. وینستون^۴ نگاشته شده است، تشریح گردیده اند.

^۱ Lindo Systems, Inc

^۲ Operations Research : Applications and Algorithms (1994)

^۳ Introduction to Mathematical Programming : Application and Algorithms (1995)

^۴ Wayne L.Winston

نرم افزارهای مشابه

امروزه بسیاری از مدل های بهینه سازی تحقیق در عملیات اعم از مدل های خطی، غیر خطی و یا عدد صحیح به مدد نرم افزارهای کامپیوتری به سادگی قابل تجزیه و تحلیل می باشند. در این میان می توان از نرم افزارهایی چون GAMS، GINO، LINDO، LINGO، QSB و TORA نام برد.

LINGO بعد از GAMS قوی ترین نرم افزار تحقیق در عملیات می باشد. از جمله برتری های لینگو نسبت به LINDO یا GAMS، قدرت آن در مدل سازی تمام مسائلی است که توسط لیندو مدل شده اند، بدون این که نیازی به تعیین نوع مدل توسط کاربر باشد، در حالی که LINDO یا GAMS چنین قابلیت را دارا نمی باشد. یکی دیگر از قابلیت های مهم لینگو برخورداری از Help ی بسیار قوی، ساده و کامل می باشد. به گونه ای که با استفاده از Help آن می توان تا حد زیادی بر این نرم افزار مسلط شد. آخرین نسخه این نرم افزار نسخه ۱۲ بوده ولی نسخه ۱۱ آن رایج تر است.

ویژگیهای بارز لینگو

- ❖ قابلیت مدل سازی بصورت کارا و صحیح؛
- ❖ قابلیت بالای تحلیل مدل؛
- ❖ دارا بودن توابع مختلف ریاضی، آماری و احتمالی؛
- ❖ قابلیت خواندن اطلاعات از File ها و Worksheet های دیگر؛
- ❖ کار کردن در محیط Windows.

لینگو چیست ؟

تعریف اول: راهنمای PDF لینگو

لینگو ابزاری ساده برای بهره گیری از قدرت برنامه ریزی خطی و غیر خطی در فرموله کردن مسائل خیلی بزرگ به صورت مختصر و تجزیه و تحلیل آنهاست. بهینه سازی به شما کمک می کند تا جوابی را با بیشترین سود، خروجی، یا رضایتمندی و یا جوابی با کمترین هزینه، تلفات یا ناراحتی بدست آورید. اغلب این مسائل با استفاده موثر از منابعی همچون پول، زمان، ماشین، نیروی کار، موجودی و ... در ارتباط است.

مسائل بهینه سازی اغلب به خطی و غیر خطی تقسیم می شوند، که این تقسیم بندی با توجه به رابطه متغیرهاست؛ بدین معنی که اگر رابطه همه متغیرها با هم خطی باشد، مسئله را خطی و در غیر اینصورت غیر خطی می گویند. لینگو زبانی جامع و فراگیر جهت تسهیل تمام مدل های بهینه سازی از مسائل تحقیق در عملیات گرفته تا اقتصاد مهندسی، شبیه سازی، کنترل کیفیت، کنترل پروژه و کنترل موجودی می باشد.

تعریف دوم : کتاب تحقیق در عملیات واین ال. وینستون :

لینگو یک بسته نرم افزاری با امکان برقراری ارتباط دو طرفه با کاربر است و از آن می توان در حل مسائل خطی، عدد صحیح و غیر خطی کمک گرفت. استفاده از این نرم افزار وضعیت مشابهی مانند لیندو دارد، اما از انعطاف بیشتری در بیان مدل برخوردار است. بر خلاف لیندو، لینگو امکان استفاده از پرانتزها و متغیرها را در سمت راست معادله فراهم می آورد. بنابراین، محدودیت ها را می توان به صورت اولیه، بدون نیاز به تبدیل آن به ساختار استاندارد (آوردن مقادیر ثابت به سمت راست) نوشت. همچنین لینگو دارای یک کتابخانه بزرگ از توابع ریاضی، آماری و احتمالی بوده و قدرت بالای آن در خواندن اطلاعات از فایل های خارجی و نرم افزارهای صفحه گسترده است.

بخش دوم – «آغاز کار با لینگو»

- ✓ نصب نرم افزار
- ✓ اصول اولیه در لینگو
- ✓ محیط نرم افزار
- ✓ پیاده سازی مدل در لینگو

نصب نرم افزار:

نصب لینگو به راحتی انجام می گیرد و همانند بسیاری از نرم افزارهای تحت سیستم عامل Windows می باشد.

تنها نکته ای که در نصب می توان به آن اشاره کرد فضای مورد نیاز در روی دیسک سخت می باشد که 40MB است. برای دریافت نرم افزار می توانید به سایت لینگو مراجعه کنید.

اصول اولیه در لینگو

برای حل مسائل در لینگو که بسیار مشابه لندو است، می توان مسائل را به سه روش از طریق صفحه کلید و در ارتباطی دو طرفه با کاربر و یا با استفاده از فایل هایی که در جای دیگر تولید شده اند (که این فایل ها یا خود شامل دستورات هستند و یا به عنوان بخشی از یک برنامه جامع، دارای کدهای سفارشی و کتابخانه های بهینه سازی لینگو هستند)، وارد کامپیوتر کرد. در اینجا به روش اول، یعنی ارتباط دو طرفه با کاربر متمرکز می شویم.

وارد کردن یک مدل در نسخه تحت ویندوز لینگو، شبیه تایپ کردن در یک محیط پردازشگر متنی تحت ویندوز است؛ چون وقتی در یک مدل داده تایپ می کنید، تقریباً باید آنچه را که به صورت دستی نوشته اید، وارد مدل کنید. کاربر می تواند اطلاعات ورودی خود را در پنجره ی داخلی که ابتدا تحت عنوان Lingo Model است، وارد کند. همچنین، لینگو شامل دستورات ابتدایی ویرایش برای بریدن، کپی کردن و چسباندن یک متن است. این ابزارها و سایر ویژگی ها، در منوی فرمان ها موجود است که در ادامه مورد بررسی قرار می گیرد.

عناصر مورد نیاز در لینگو مشابه لندو است. لینگو نیازمند تابع هدف، یک یا چند متغیر و یک یا چند محدودیت است. بر خلاف لندو، محدودیت های لینگو بعد از عبارات خاص از قبیل SUBJECT TO یا SUCH THAT نمی آیند.

❖ در لینگو هر عبارت با یک " ; " پایان می یابد؛

❖ هرگاه بخواهیم یک عبارت توضیحی به متن برنامه بیافزاییم کافست در ابتدای عبارت از علامت تعجب (!) استفاده کنیم. این توضیحات نیز به " ; " ختم می شوند. کلیه عبارات متنی که مابین (!) و " ; " نوشته می شوند از سوی لینگو نادیده گرفته می شوند؛

❖ لینگو به بزرگ یا کوچک بودن حروف حساس نیست.

❖ می توانیم دو خط برنامه لینگو را در یک خط بیاوریم و یا یک خط آن را در دو سطر بیاوریم:

$$\text{Max} = 100 * \text{Standard}$$

+ 150 * Turbo;

یا

Standard <= 100; Turbo <= 120;

❖ حتی می توانیم توضیحات را در وسط محدودیت ها بیاوریم:

$X = Y + \text{!Number of Products}; Q + 2 * C;$

(معادل $X = Y + Q + 2 * C$ است)

❖ عملگرهای ریاضی حتما فراموش نشود؛ (مخصوصا عملگر ضرب “ * ”)

$2X + 3Y \leq 25$ (اشتباه)

$2 * X + 3 * Y \leq 25$ (درست)

❖ اسمی متغیرها حتما با یک حرف شروع می شوند. بعد از آن می توان حرف، رقم یا زیر خط (_) استفاده کرد. اسم متغیرها نمی تواند بیشتر از ۳۲ کاراکتر باشد.

❖ استفاده از فاصله “ ” (فاصله) در اسم متغیرها ممنوع است.

❖ علائم ساده ریاضی، توان (^)، ضرب (*)، تقسیم (/)، جمع (+) و تفریق (-) با همین ترتیب اولویت معتبرند.

❖ اگر بخواهیم به محدودیت اسمی اختصاص دهیم در ابتدای محدودیت اسم آن را در داخل علامت کروشه می آوریم؛ و به صورت مقابل می نویسیم : [اسم محدودیت]

❖ Help لینگو یک راهنمای بسیار خوب برای استفاده بهتر از نرم افزار است. از آن بهره بگیرید!

❖ هنگام حل مدل اگر ایرادی در آن وجود داشته باشد، نرم افزار پیغام خطا داده و مکان نما به اول سطری می رود که به نظر لینگو این خطا از آنجا ناشی شده است.

ممکن است در بعضی مواقع با وجود آنکه فکر می کنید مدلی که نوشته اید درست است، نرم افزار هنگام حل آن پیغام خطا بدهد. در این مواقع یک راه ساده آن است که یک صفحه جدید لینگو باز کرده و دستورات و عبارت هایی که در مدل نوشته اید را یکی یکی، به صفحه ی جدید کپی کرده و هر بار مدل جدید را اجرا کنید. با این کار در اکثر موارد به راحتی می توانید به اشکال برنامه پی ببرید.

محیط نرم افزار:

نرم افزار لینگو از سه بخش کلی منوها، Toolbar و پنجره ی مدل نویسی تشکیل شده است.

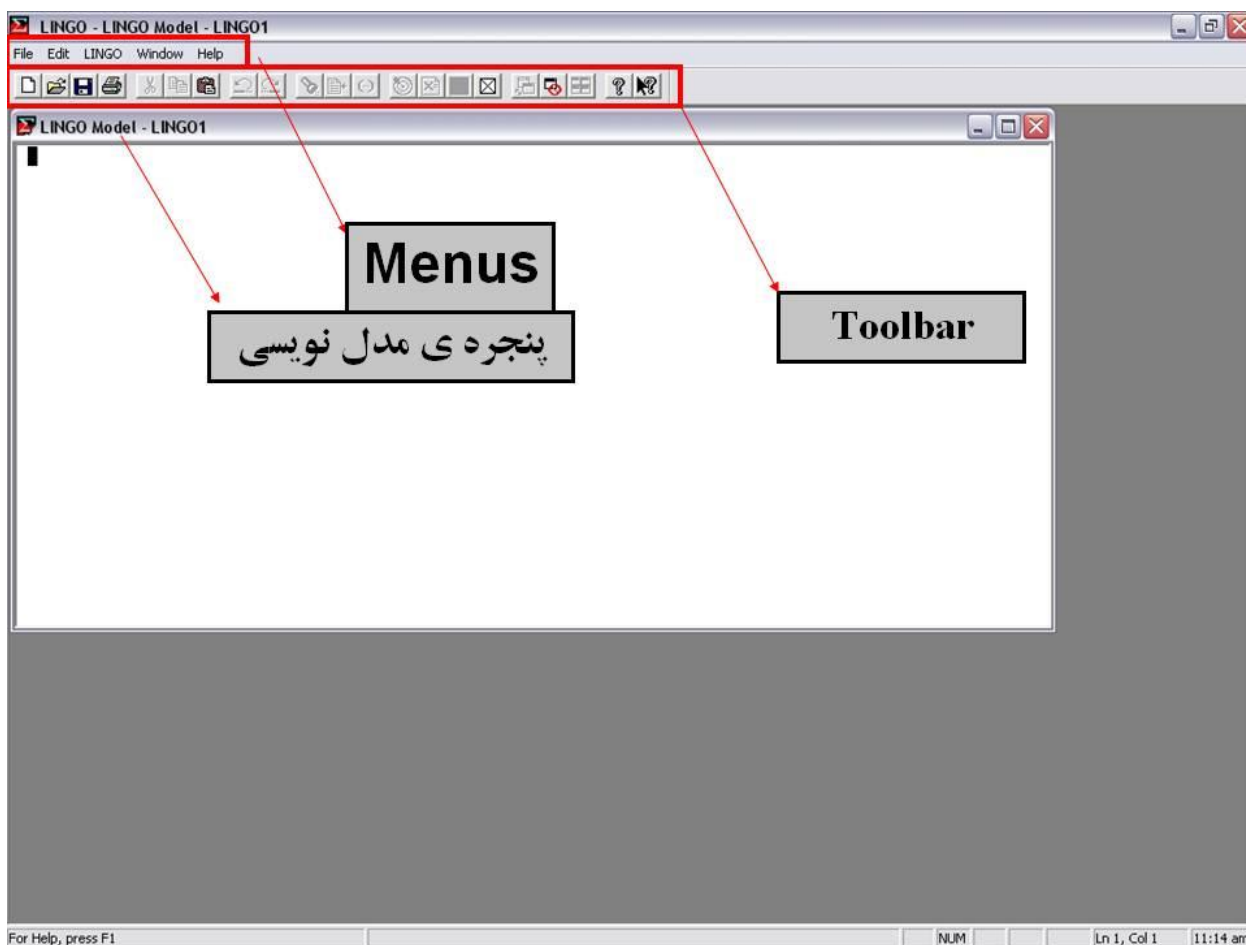
❖ منوهای نرم افزار به ترتیب زیر می باشد :

File; Edit; LINGO; Window; Help

منویی که بیشتر به چشم آمده و در واقع منوی اصلی نرم افزار می باشد منوی LINGO است، با سایر منوها که در نرم افزارهای تحت سیستم عامل Windows عموماً موجود می باشند پیشتر آشنا هستید.

❖ Toolbar شامل ابزارهای Shortcut (میانبر) موجود در منوها می باشد.

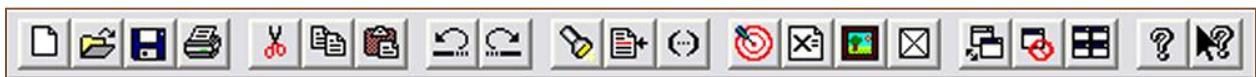
❖ پنجره ی مدل نویسی محل پیاده سازی مسئله ی بهینه سازی فرموله شده طبق قواعد لینگوست.



هنگامی که نرم افزار لینگو را اجرا می کنید، صفحه ای مطابق شکل فوق نمایش داده می شود، همان طور که مشاهده می کنید، کلید هایی برای استفاده کاربر در نظر گرفته شده که در واقع هر کدام از آنها یک کلید میان بر، برای گزینه مرتبط در یکی از منوهای لینگو می باشد .

هنگامی که هیچ صفحه ای در لینگو باز نباشد، بسیاری از کلیدها غیر فعال هستند. اما با باز کردن یک برنامه به زبان لینگو و اجرای آن تقریباً همه این کلیدها فعال می شوند. برای آشنایی بیشتر با هر کدام از این کلیدها، در ارتباط با هر کدام به اختصار توضیح می دهیم.

Toolbar

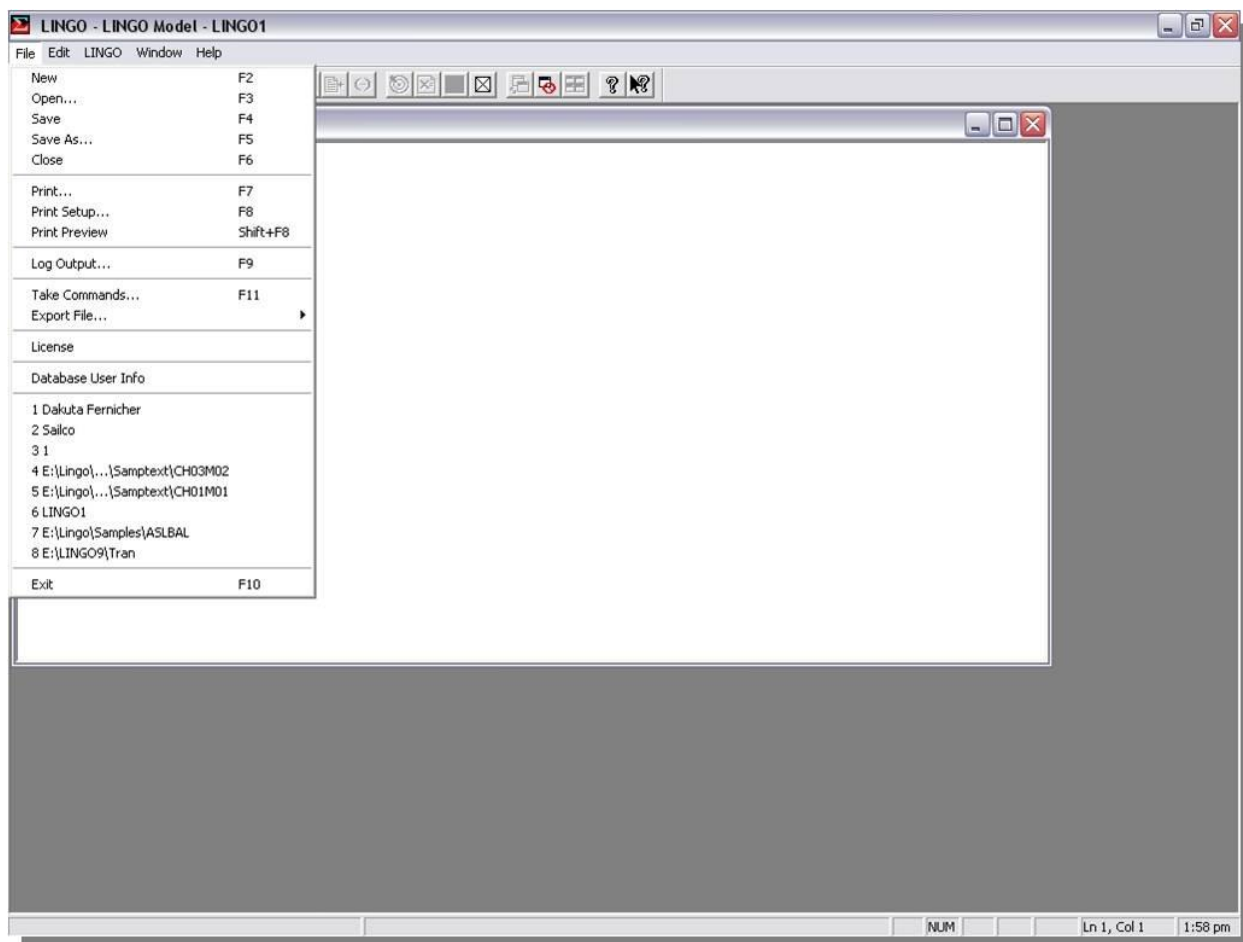


کلید	کاربرد
New	کلید میان بر گزینه New / از منوی File می باشد.
Open	کلید میان بر گزینه Open / از منوی File می باشد.
Save	کلید میان بر گزینه Save / از منوی File می باشد.
Print	کلید میان بر گزینه Print / از منوی File می باشد.

Cut	کلید میان بر گزینه Cut/از منوی Edit می باشد.
Copy	کلید میان بر گزینه Copy/از منوی Edit می باشد.
Paste	کلید میان بر گزینه Paste/از منوی Edit می باشد.
Undo	کلید میان بر گزینه Undo از منوی Edit می باشد.
Redo	کلید میان بر گزینه Redo از منوی Edit می باشد.
Find	کلید میان بر گزینه Find از منوی Edit می باشد.
Go To Line	کلید میان بر گزینه Go To Line از منوی Edit می باشد.
Math Parenthesis	کلید میان بر گزینه Math Parenthesis از منوی Edit می باشد.
Solve	کلید میان بر گزینه Solve از منوی Lingo می باشد.
Solution	کلید میان بر گزینه Solution از منوی Lingo می باشد.
Matrix Picture	کلید میان بر گزینه Matrix Picture از منوی Lingo می باشد.
Options	کلید میان بر گزینه Options از منوی Lingo می باشد.
Send To Back	کلید میان بر گزینه Send To Back از منوی Window می باشد.
Close All	کلید میان بر گزینه Close All از منوی Window می باشد.
Tile Windows	کلید میان بر گزینه Tile Windows از منوی Window می باشد.

کلید جستجو و کلیک برای Help می باشد.	Help Topics
کلید میان بر از منوی Help می باشد.	Help

Menus (File Menu)



پنجره ی جدید برای وارد کردن داده ها ایجاد می کند	New
پرونده ی موجود را باز می کند. در جعبه های محاوره ای (dialog box)، می توانید پرونده را از میان انواع مختلف پرونده ها و در مکان های مختلف انتخاب کنید	Open
پنجره ی فعال را ذخیره می کند. می توانید داده های ورودی (یک مدل)، یک پنجره گزارش یا یک پنجره فرمان را ذخیره کنید	Save
پنجره فعال را با نام مشخصی ذخیره می کند. کاربرد این فرمان در تغییر نام پرونده اصلاح شده است، در حالی که پرونده اولیه حفظ می شود	Save As

پنجره فعال را می بندد. اگر پنجره شامل داده های جدید باشد، برای ذخیره برنامه سوال می کند	Close
پنجره فعال را به چاپگر می فرستد	Print
چاپگر را انتخاب می کند. همچنین، گزینه های مختلفی را برای شکل چاپ، در دسترس قرار می دهد	Print Setup
پیش نمایش محتویات پنجره ی فعال برای چاپ را نمایش می دهد	Print Preview
تمامی زیر پنجره های عملیات را که در حالت معمولی به پنجره گزارش فرستاده می شود، به یک فایل نوشتاری ارسال می کند. وقتی محل پرونده نوشتاری را تعیین کردید، علامتی در کنار این فرمان در منوی فایل ظاهر خواهد شد. برای غیر فعال کردن این فرمان، می توانید مجددا این گزینه را انتخاب کنید	Log Output
برای انجام عملیات به صورت خودکار فایل بسته ای (batch file) لینگو را همراه با فرمان ها و متن، وارد حافظه می کند. یک مدل می تواند در حافظه قرار گیرد، حل شود و نتیجه حل آن در یک پنجره گزارش قرار گرفته و در یک فایل ذخیره شود. اگر یک فرمان بسته ای را قبل از شروع متن مدل به کار برید، مدل و دستورات موجود در فایل، در پنجره گزارش و در حل دیده می شود	Take Commands
صدور یک مدل در فرمت MPS یا MPI	Export File
دادن کلمه عبور جدید جهت به روز رسانی سیستم و آزاد شدن نرم افزار	License
واداشتن شما به استفاده از شناسه کاربری و کلمه عبور برای دستیابی به پایگاه اطلاعات از طریق تابع @ODBC()	Database User Info
خروج از نرم افزار	Exit

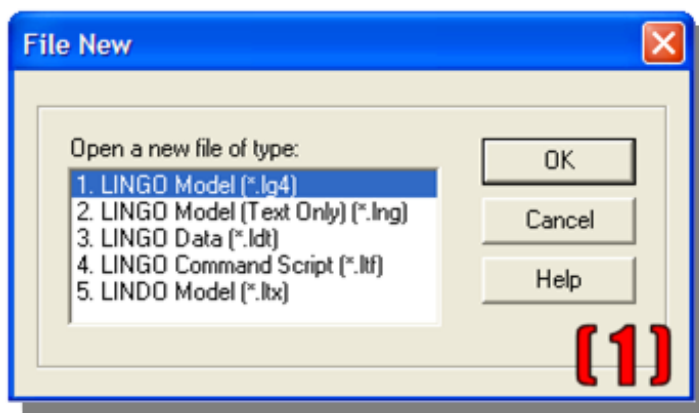
کار با منوی File

در این منو به توضیح ۳ فرمان زیر می پردازیم، با سایر فرمان ها در سایر نرم افزارها بیشتر آشنا شده اید.

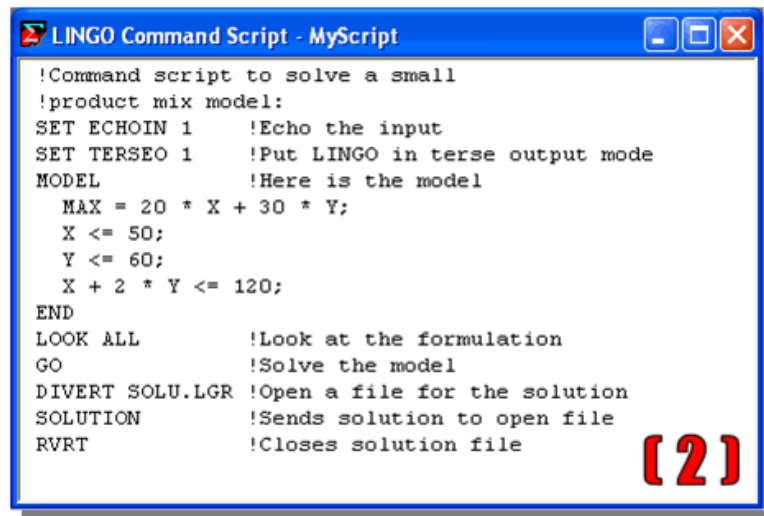
- Log Output
- Take Commands
- Database User Info

پس از اجرای نرم افزار LINGO Model را که بصورت اتوماتیک باز شده Close می کنیم.

سپس از منوی File فرمان New را انتخاب می کنیم، در Dialog Box ظاهر شده گزینه چهارم، یعنی LINGO Command Script را انتخاب کرده OK می کنیم. (1)



در پنجره ی باز شده متن آورده شده در شکل زیر را تایپ کنید. (2)



```

LINGO Command Script - MyScript
!Command script to solve a small
!product mix model:
SET ECHOIN 1      !Echo the input
SET TERSEO 1      !Put LINGO in terse output mode
MODEL             !Here is the model
    MAX = 20 * X + 30 * Y;
    X <= 50;
    Y <= 60;
    X + 2 * Y <= 120;
END
LOOK ALL          !Look at the formulation
GO                !Solve the model
DIVERT SOLU.LGR   !Open a file for the solution
SOLUTION          !Sends solution to open file
RVRT              !Closes solution file
    
```

مدل را حل کنید و جواب را در یک فایل متنی نگاه دارید. اکنون با استفاده از فرمان Save از منوی File متن تایپ شده را با پسوند ltf ذخیره کنید. اکنون از منوی File فرمان Take Command را اجرا کنید. مشاهده می کنید که ورودی به همراه خروجی در پنجره Command Window آمد، به عبارت دیگر صورت مسئله به همراه جواب آن یکجا آورده شد.

در عکس زیر بعد پنجره Command Window را مشاهده می کنید. (3)



```
Command Window

: : !Command script to solve a small
: !product mix model:
: SET ECHOIN 1      !Echo the input

Parameter      Old Value      New Value
ECHOIN         1              1

: SET TERSEO 1      !Put LINGO in terse output mode

Parameter      Old Value      New Value
TERSEO         1              1

: MODEL              !Here is the model
?  MAX = 20 * X + 30 * Y;
?  X <= 50;
?  Y <= 60;
?  X + 2 * Y <= 120;
? END
: LOOK ALL           !Look at the formulation

1]  MAX = 20 * X + 30 * Y;
2]  X <= 50;
3]  Y <= 60;
4]  X + 2 * Y <= 120;

: GO                 !Solve the model

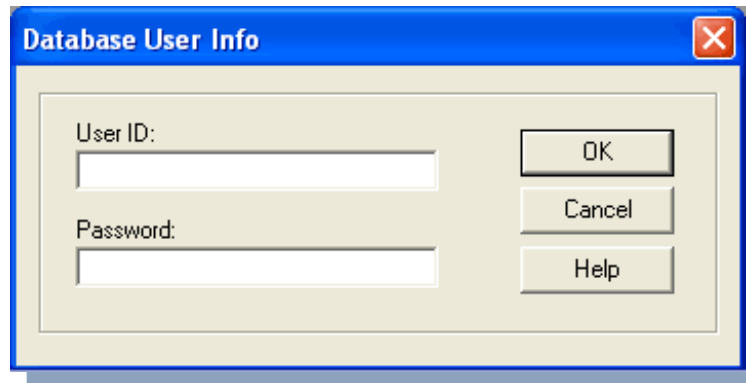
Global optimal solution found at step:      2
Objective value:                          2050.000

: DIVERT SOLU.LGR !Open a file for the solution
: SOLUTION         !Sends solution to open file
: RVRT             !Closes solution file
: 
```

[3]

برای ذخیره کلیه اطلاعات موجود در این پنجره و بازتاب خروجی از فرمان Log Output موجود در منوی File استفاده کنید.

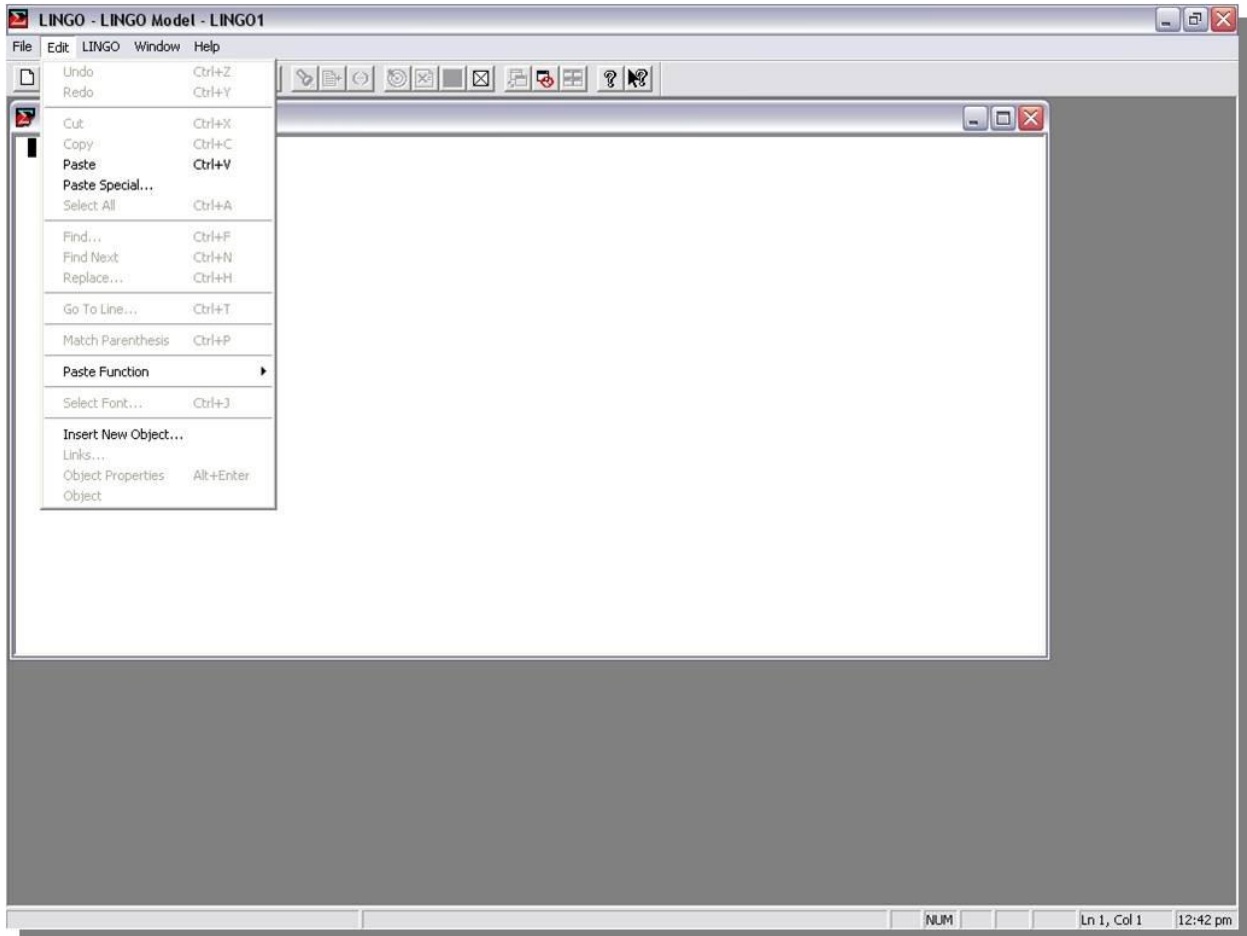
Database User Info فرمان دیگری از منوی File می باشد. هنگامی که از تابع {ODBC}@ در مدل استفاده می کنید و نیاز به کلمه و رمز عبور برای استفاده از بانک اطلاعاتی دارید می بایست از قبل در این قسمت کلمه و رمز عبور را وارد کنید.



منوي Edit



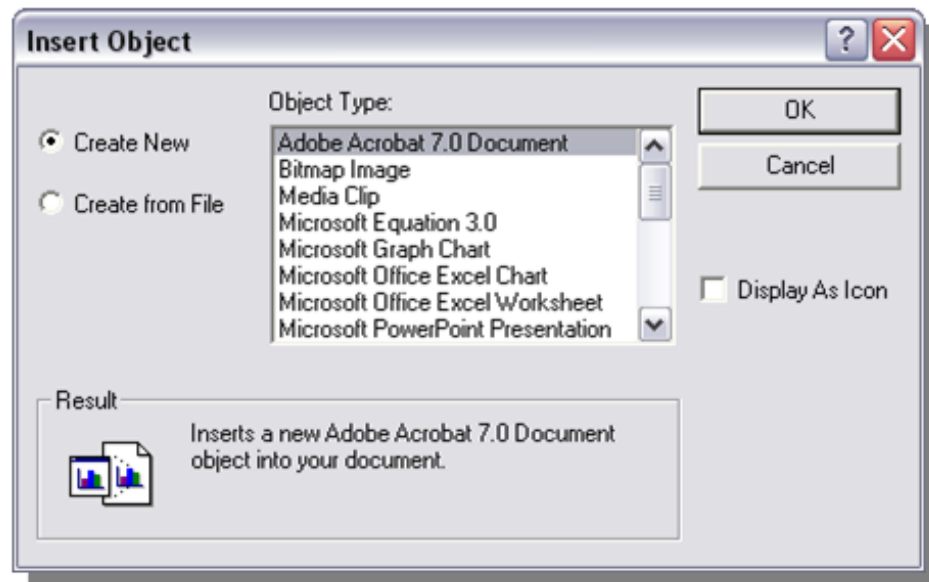
www.ieun.ir



آخرین کار انجام شده را باز می گرداند	Undo
بر عکس عملیات Undo را انجام می دهد	Redo
متن انتخابی را پاک کرده، در حافظه نگه می دارد تا جای دیگری آن را قرار دهد	Cut
متن انتخابی را در حافظه کپی می کند تا در جای دیگری قرار داده شود	Copy
محتوای حافظه (Clipboard) را در جای انتخابی قرار می دهد	Paste
Paste همراه با انتخاب نوع Paste از نظر فرمت متن	Paste Special

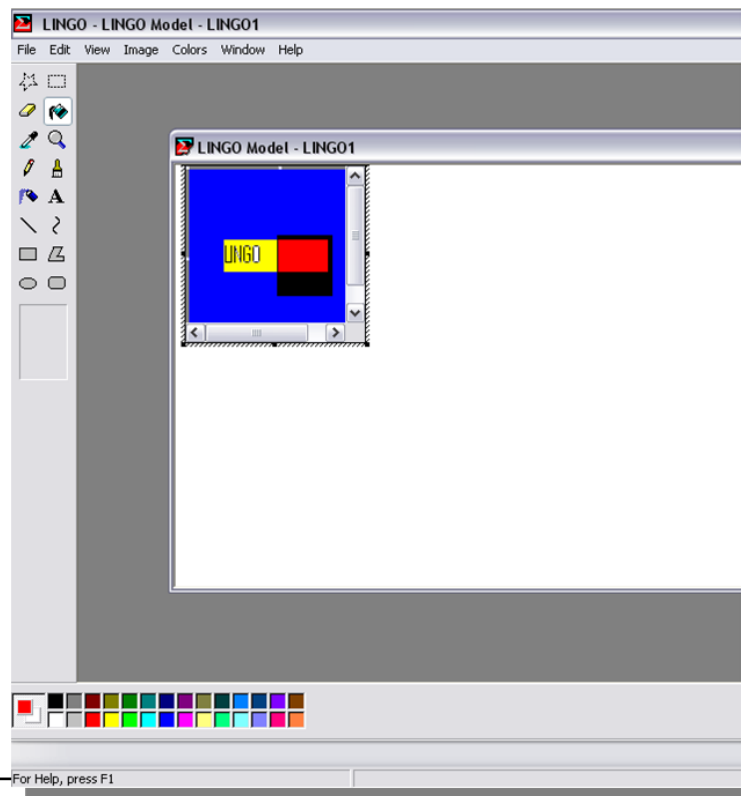
Select All	تمامی محتویات پنجره فعال را انتخاب می کند تا بتوان از آن در کپی و یا برش استفاده کرد
Find	جستجوی قسمتی از رشته ی متنی در پنجره ی فعال
Find Next	جستجوی رشته ی متنی مشابه بعدی در متن که قبلاً توسط Find پیدا شده
Replace	جایگزین کردن رشته های متنی با متن جدید بصورت دلخواه
Go To Line	حرکت دادن Cursor (نشانگر صفحه کلید) به شماره خط دلخواه از پنجره فعال
Match Parenthesis	پرانتز های بسته را که با پرانتز های باز انتخاب شده، متناظر هستند، پیدا می کند
Paste Function	توابع داخلی لینگو را در جای مشخص شده، قرار می دهد. بعد از انتخاب این فرمان، زیر منوی دیگری باز می شود که شامل گروه بندی های مختلفی از توابع است
Select Font	انتخاب قلم دلخواه با فرمت دلخواه (پس از Select کردن متن فعال می شود)
Insert New Object	افزودن لینک از نرم افزار های دیگر (به عنوان مثال یک اسلاید PowerPoint)
Links	کنترل ارتباطات عناصر خارجی موجود در Document
Object Properties	خاصیت دادن به عناصر موجود در Document

از منوی Edit فرمان Insert New Object را اجرا کنید، پنجره ای بصورت زیر باز می شود.

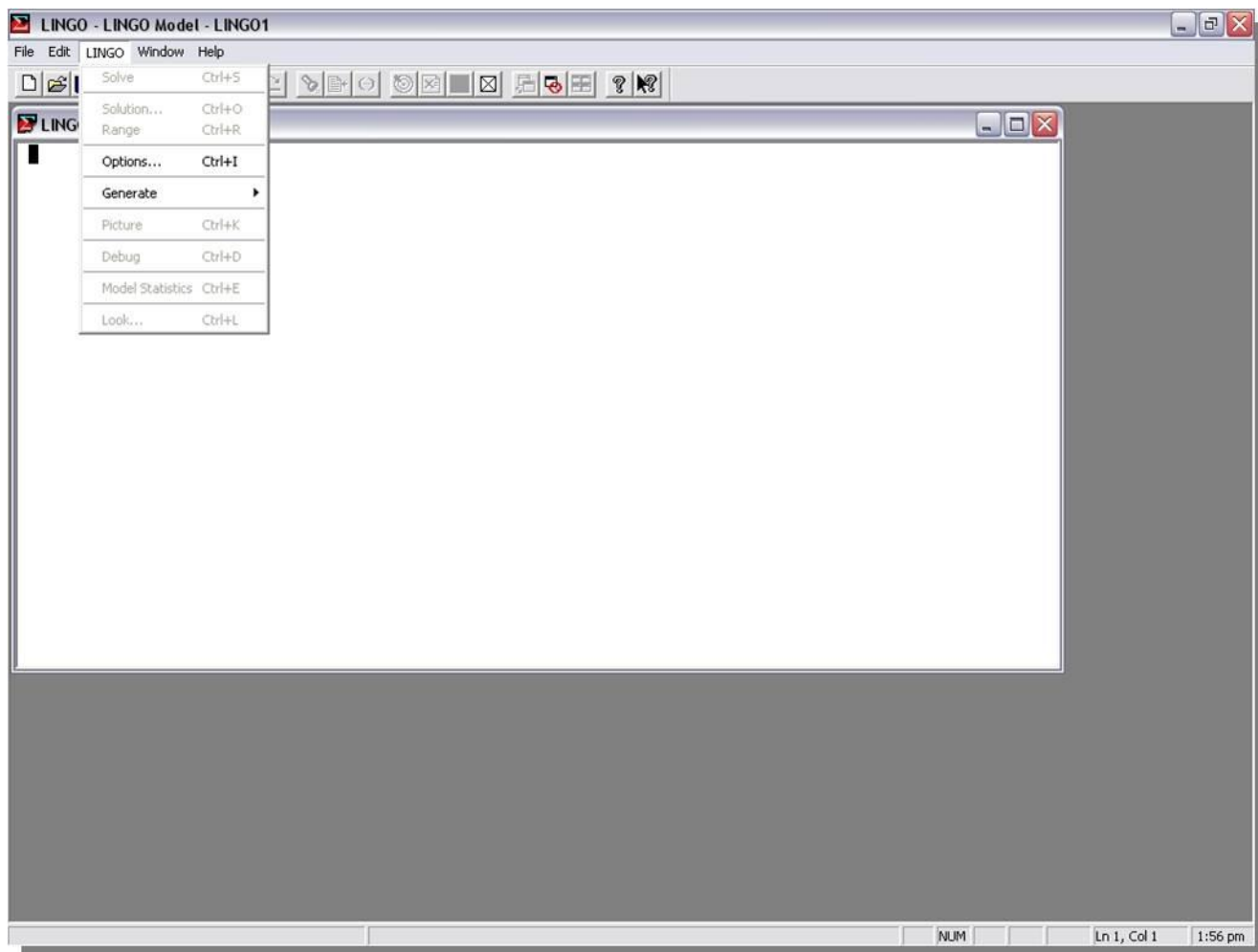


دو انتخاب داریم، اولی مربوط به محیط نرم افزار دلخواه است و دومی فایل ذخیره شده، به عنوان مثال در حالت اول Bitmap Image را انتخاب کنید.

اکنون در LINGO Model شکل زیر را شاهد هستید، با Double Click روی آن نرم افزار Paint با کلیک ابزار های آن در اختیار شما خواهد بود، تصویر دلخواه خود را ایجاد کنید و با کلیک خارج از محدوده ی مورد نظر به لینگو باز گردید.



منوی LINGO



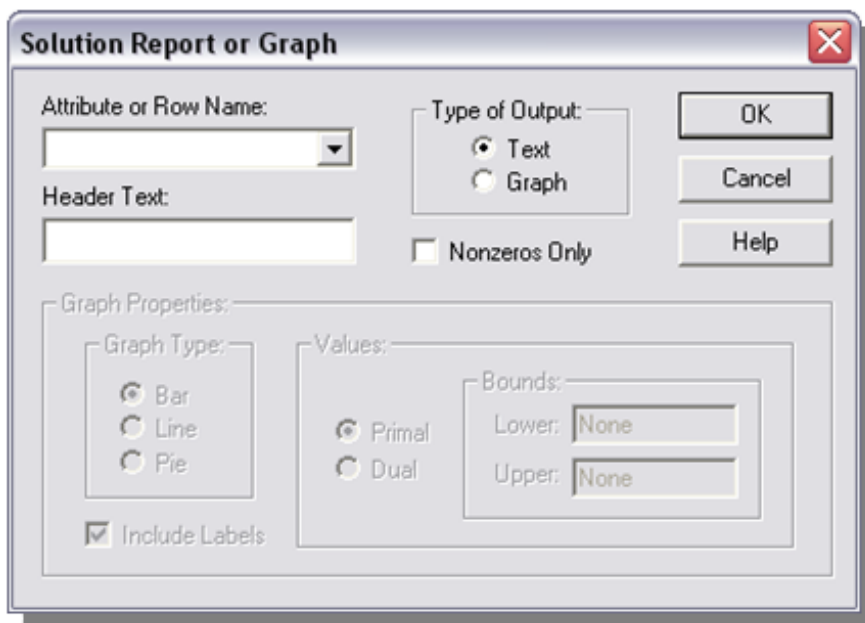
توضیحات ابزارهای منوی LINGO

مدل مربوط به پنجره فعال را برای حل، به حل کننده لینگو می فرستد	Solve
جعبه محاوره ای (Dialog Box)، گزینه هایی از گزارش حل را باز می کند، که امکان تعیین چگونگی ظاهر مورد نظر گزارش حل را فراهم می کنند	Solution

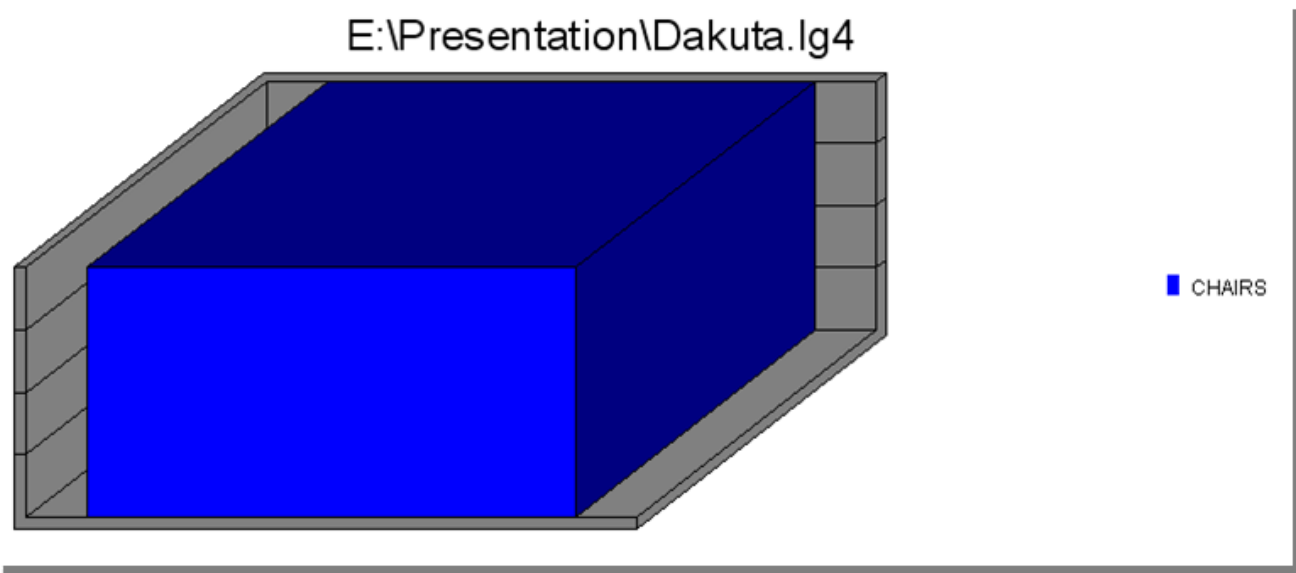
یک گزارش حد (تحلیل حساسیت) ارائه می کند. این گزارش نشان می دهد که تغییرات ضرایب می تواند در چه محدوده ای باشد، بدون اینکه مقادیر بهینه عوش شوند	Range
امکان مشاهده و تغییر پرامترهای مختلفی را که در هنگام استفاده از لینگو درگیر هستند، فراهم می کند	Options
ویرایش دیگری از مدل را به صورت جبری، تحت لیندو و یا ساختار MPS تولید می کند. این فرمان می تواند برای شماره گذاری سطر ها و نمایش مدل به صورتی که کاربر بتواند راحت تر با آن کار کند، مورد استفاده قرار می گیرد. فرمان GEN، از پنجره فرمان نیز، دارای قابلیت مشابهی از این نوع است	Generate
مدل را بصورت ماتریسی نمایش می دهد	Picture
خطاهای موجود در مدل را یافته و توضیح می دهد	Debug
اطلاعاتی راجع به مدل در اختیار قرار می دهد (مانند تعداد متغیرها، تعداد سطر ها و...)	Model Statistics
تمام و یا قسمتی از سطرهای انتخابی مدل را نمایش می دهد	Look

کار با منوی LINGO

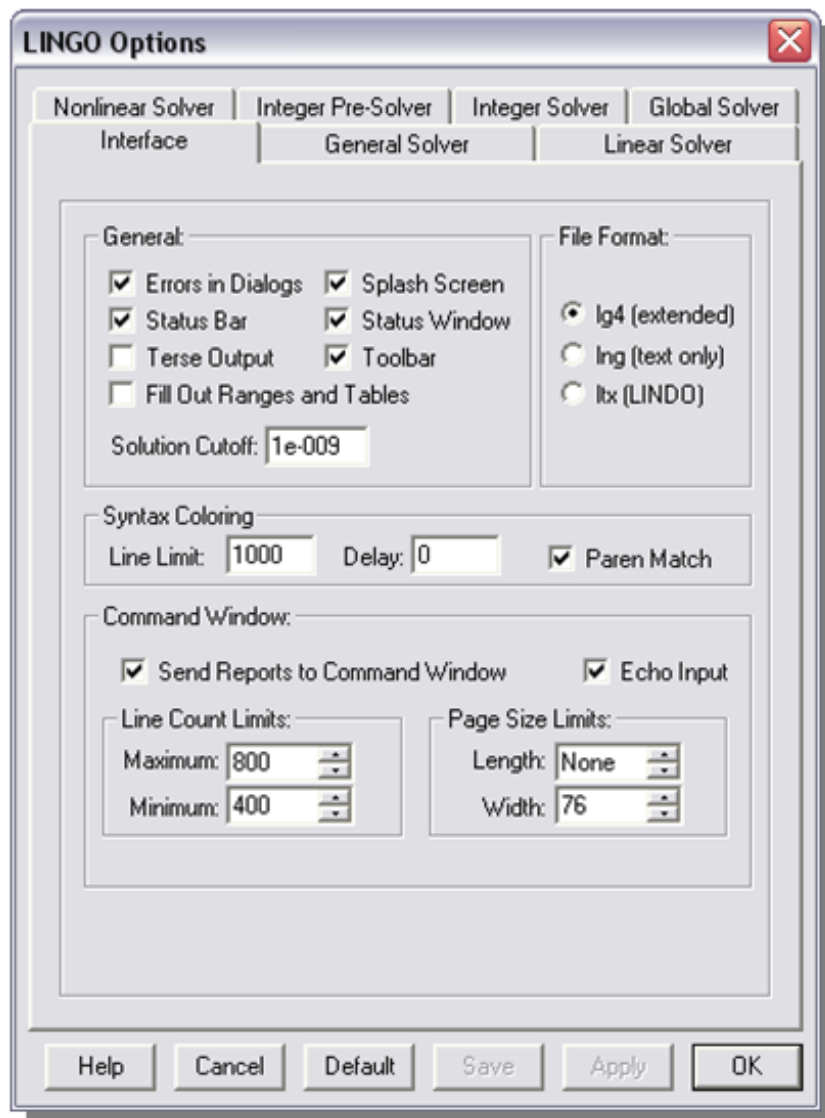
فرمان های Solution و Range پس از حل مدل فعال خواهند شد. فرمان Solution را پس از حل مدل داکوتا اجرا می کنیم، پنجره ی مقابل باز خواهد شد. از منوی آبشاری نوع متغیری را که مد نظر دارید انتخاب کنید و با انتخاب گزینه Graph و تعیین نوع نمودار OK کنید.



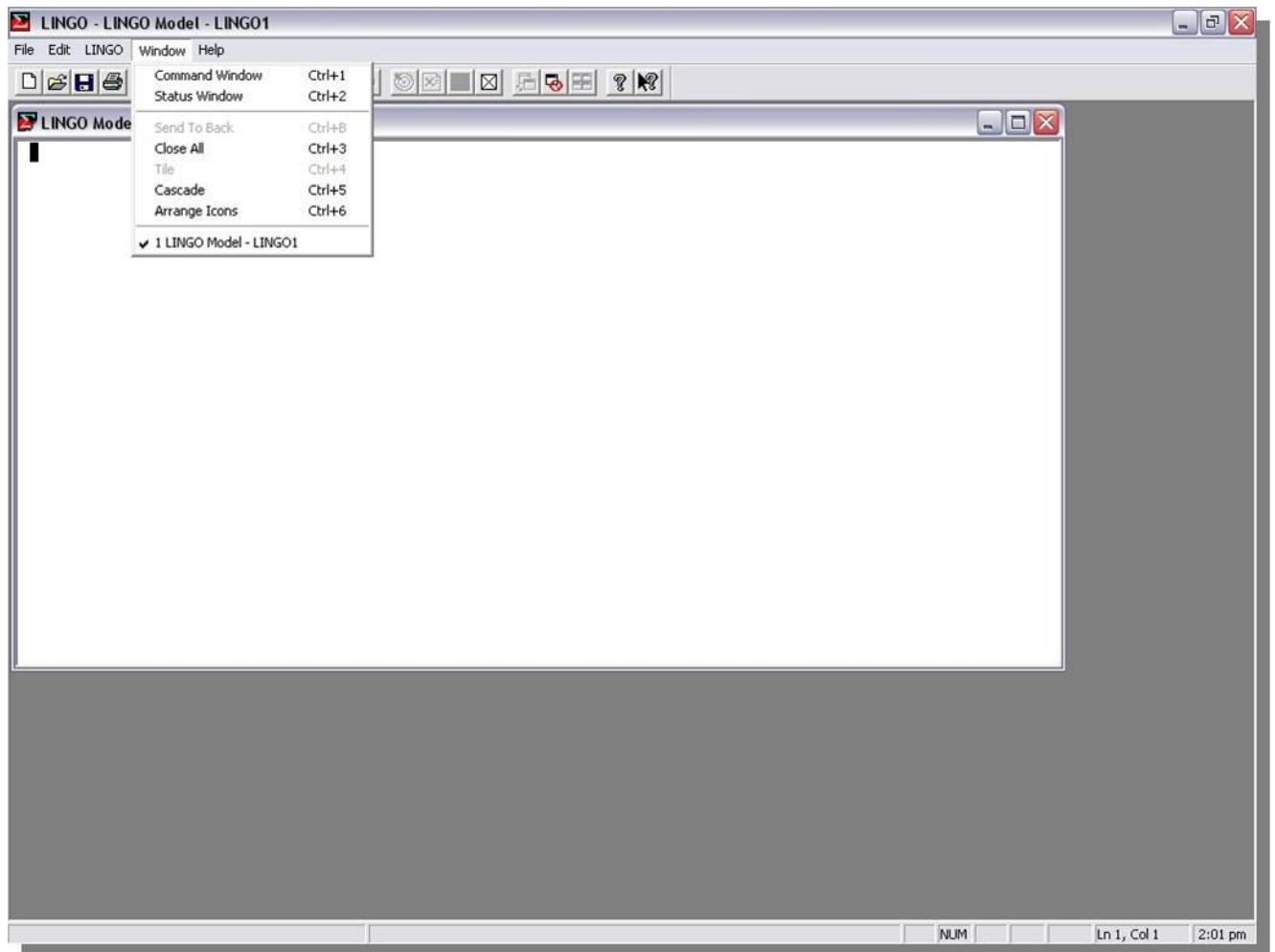
به عنوان مثال Chair را انتخاب کردیم، داریم:



فرمان دیگری که در این منو به چشم می آید فرمان **Option** است، این فرمان همانگونه که از نامش پیداست گزینه های مختلفی را جهت مطابق میل نمودن نرم افزار در اختیار کاربر قرار می دهد، به عنوان مثال با تیک زدن **Send Report To Command Window** از قسمت **Interface** تمامی خروجی های نرم افزار را بصورت پیوسته در پنجره **Command Window** مشاهده خواهید کرد و ...



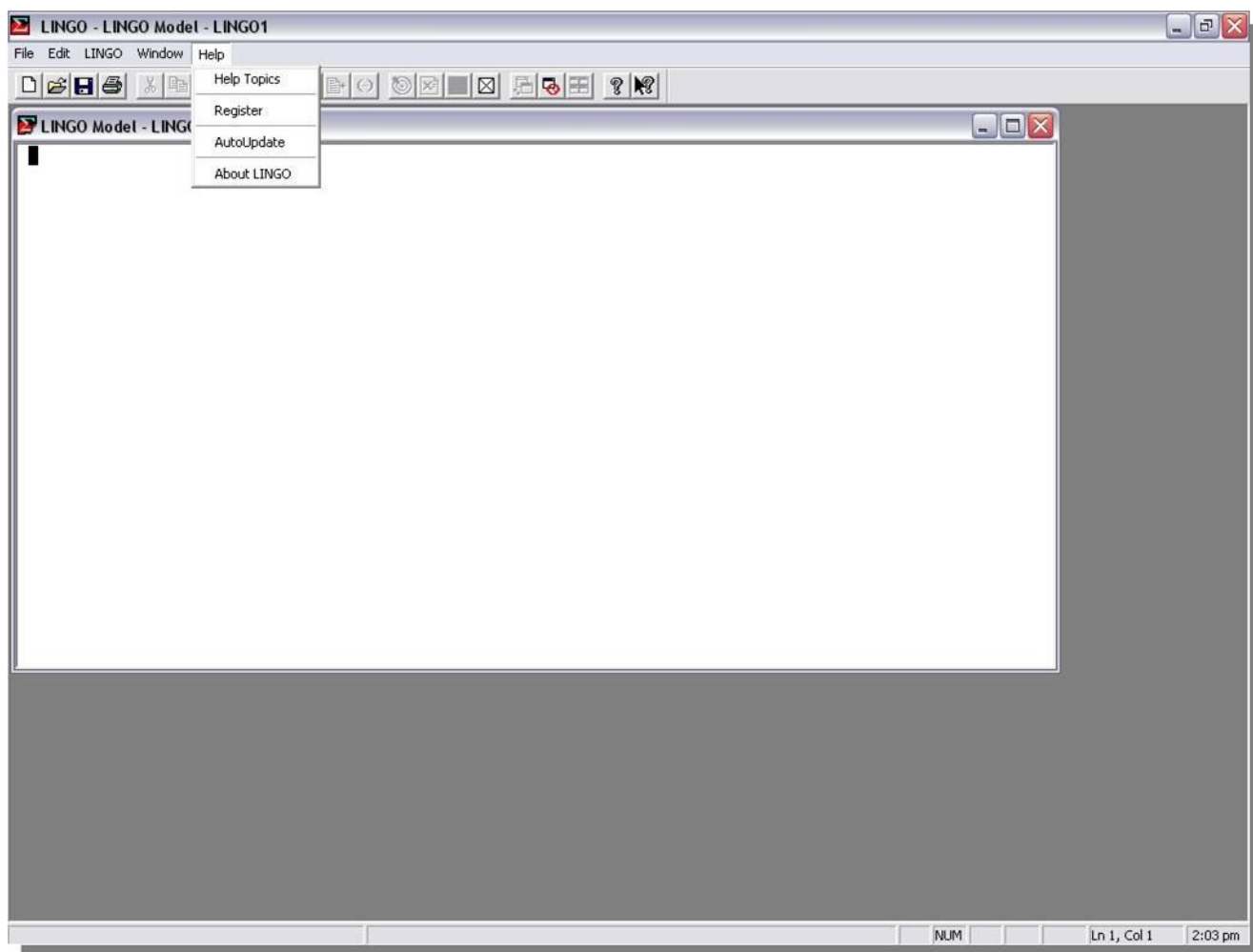
منوی Windows



امکان دسترسی به رابط کاربر با خط فرمان لینگو را فراهم می آورد. در اینجا می توانید فرمان ها را در مقابل علامت : وارد کنید	Command Window
پنجره وضعیت حل کننده لینگو را که نمایش دهنده اطلاعاتی در مورد وضعیت بهینه سازها از قبیل تعداد تکرارها و زمان اجرای صرف شده است، باز می کند	Status Window

پنجره ی رویی را به پشت می فرستد	Send to Back
تمامی پنجره های باز را می بندد	Close All
تمامی پنجره های باز را به گونه ای مرتب می کند که همگی فضای مساوی را در پنجره برنامه اشغال کنند	Tile
تمامی پنجره های باز را به صورت آبشاری، از سمت چپ- بالا تا سمت راست-پایین مرتب کرده و پنجره فعال را نیز در بالا قرار می دهد	Cascade
آیکون هایی را که نشانه پنجره های کوچک شده است، در امتداد پایین صفحه نمایش، مرتب می کند	Arrange Icons

منوی HELP

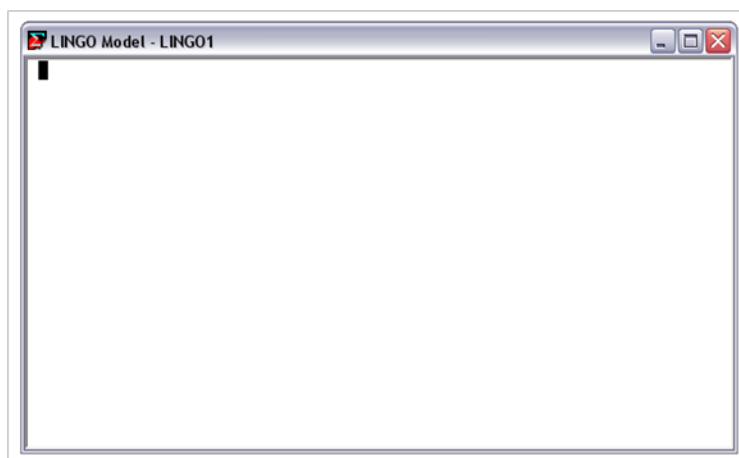


راهنمای موضوعی، امکان جستجو نیز وجود دارد	Help Topics
ثبت لینگو بصورت آنلاین	Register

شما را از ورژن های جدید و به روز رسانی نرم افزار بصورت آنلاین آگاه می سازد	AutoUpdate
صفحه آغازین لینگو را به همراه اطلاعات عمومی مربوط به آن نمایش می دهد	About LINGO

پنجره مدل نویسی (Lingo Model)

لینگو زبان برنامه نویسی خاص خود را دارد، همانند زبان های برنامه نویسی دیگر لینگو نیز دارای اصول و قواعد نگارشی و منطقی در برنامه نویسی می باشد، با استفاده از Help جامع نرم افزار و همچنین با بهره گیری از پیغام های خطا و سعی در رفع آنها می توان به تدریج بر آن مسلط شد. پنجره ی برنامه نویسی را در زیر مشاهده می کنید:



پیاده سازی مدل در لینگو

توسعه یک مدل بهینه سازی

یک شرکت کامپیوتری دو نوع کامپیوتر استاندارد و توربو تولید می کند. سود هر کامپیوتر استاندارد برای این شرکت \$100 و هر دستگاه توربو برابر \$150 است. در شرکت، خط تولید کامپیوتر استاندارد حداکثر تا ۱۰۰ دستگاه در روز می تواند تولید کند. همچنین حداکثر توان تولید خط توربو برابر ۱۲۰ دستگاه در روز اعلام شده است. شرکت نیروی کار محدود ۱۶۰ نفر ساعت در هر روز در اختیار دارد. کامپیوترهای استاندارد یک ساعت نیروی کار لازم دارند

در حالی که کامپیوترهای توربو ۲ ساعت نیروی کار نیاز دارند. مساله ای که شرکت با آن روبروست این است که ترکیبی از کامپیوترهای استاندارد و توربو را برای تولید انتخاب کند که سودش را بیشینه کند در حالی که از محدودیت های توان خطوط تولید و ظرفیت نیروی کار تجاوز نکند.

یک مدل بهینه سازی شامل موارد زیر است :

تابع هدف: فرمولی است که دقیقا آنچه را که می خواهید بهینه کنید، بیان می کند. در مدل های مربوط به کسب و کار این یک تابع سود است که می خواهید آن را بیشینه کنید و یا یک تابع هزینه است که می خواهید آن را کمینه کنید. مدل ها ممکن است حداکثر یک تابع هدف داشته باشد. در مثالی که داریم، تابع هدف، سود شرکت را به صورت تابعی از خروجی کامپیوترهای استاندارد و توربو محاسبه می کند.

متغیرها: کمیت هایی هستند که تحت کنترل شما هستند و شما باید تصمیم بگیرید که بهترین مقادیر این متغیرها کدامند؛ به همین دلیل گاهی به آنها متغیرهای تصمیم گیری نیز می گویند. هدف بهینه سازی یافتن مقادیری برای متغیرهای مدل است که با توجه به همه شرایط محدود کننده که روی متغیرها اعمال می شوند، بهترین مقدار را برای تابع هدف تولید کنند. در این مثال ما دو متغیر تصمیم گیری یکی مربوط به تعداد کامپیوترهای استاندارد و دیگری مربوط به تعداد کامپیوترهای توربو داریم.

محدودیت ها: تقریبا بدون استثنا، برخی محدودیت ها برای مقادیری که متغیرها می توانند اتخاذ کنند خواهد بود. حداقل یک منبع محدود خواهد بود. (برای مثال زمان، مواد اولیه، بودجه دپارتمان شما و ...) این محدودیت ها به وسیله توابعی از متغیرهای مدل بیان می شوند. این فرمول ها به عنوان محدودیت شناخته می شود، چون مقادیری را که متغیرها می توانند بگیرند، محدود می کنند.

مثال:

در این مثال ما برای هر خط تولید یک محدودیت و برای کل نیروی کار استفاده شده نیز یک محدودیت داریم.

متغیرهای Standard و Turbo را به عنوان تعداد کامپیوترهای استاندارد و توربو که شرکت می خواهد تولید کند در نظر می گیریم. هدف شرکت بیشینه کردن سود کل است که این سود برابر مجموع سود حاصل از کامپیوترهای استاندارد و توربو است :

$$\text{Max} = 100 * \text{Standard} + 150 * \text{Turbo};$$

حال محدودیت مربوط به ظرفیت خطوط تولید را اعمال می کنیم :

$$\text{Standard} \leq 100;$$

$$\text{Turbo} \leq 120;$$

محدودیت مربوط به نیروی کار نیز بصورت زیر است :

! Our labor supply is limited;

$$\leq 160; \quad \text{Standard} + 2 * \text{Turbo}$$

همانند اکثر زبان های برنامه نویسی، هر برنامه ی لینگو نیز دارای یک ساختار کلی می باشد. در این قسمت قالب اصلی یک برنامه به زبان لینگو را معرفی می کنیم، تا خواننده دیدی کلی از آن بدست آورد. آنچه در این فصل می بایست مورد توجه قرار گیرد اصول کلی است که بدان پرداخته شده است؛ در فصل های بعدی مطالب با تفصیل بیشتری مطرح خواهند شد.

برنامه نویسی در لینگو

یک برنامه در لینگو با کلمه "Model" آغاز و به کلمه "End" ختم می شود و معمولاً از چهار بخش زیر تشکیل می شود :

❖ بخش مجموعه ها؛

❖ بخش داده ها؛

❖ تابع هدف؛

❖ بخش محدودیت ها؛

همچنین می توان در یک برنامه، تابع هدف را بعد از نوشتن محدودیت ها آورد.

مجموعه ها در لینگو

مجموعه ها در لینگو در واقع قالبی برای تعریف متغیرهای برنامه می باشند (این متغیرها می توانند متغیر تصمیم گیری و یا پارامترهای عددی مدل باشند). در لینگو دو نوع مجموعه داریم :

❖ مجموعه های مستقل؛ Primitive Sets

❖ مجموعه های وارث؛ Derived Sets

مجموعه های مستقل برای تعریف متغیرهای آرایه ای یک بعدی و مجموعه های وارث برای تعریف آرایه های چند بعدی مورد استفاده قرار می گیرند.

مجموعه های مستقل

ساختار کلی تعریف مجموعه های مستقل در لینگو به صورت زیر می باشد :

؛ نام متغیرها: / عناصر مجموعه / نام مجموعه

به عنوان مثال در برنامه قبل، مجموعه ای به نام Hour با ۲۴ عنصر تعریف شده است و دو متغیر X و Need نیز دو متغیر با ۲۴ عنصر هستند.

برای درک بهتر مطلب به مثال زیر که یک خط از یک برنامه است توجه کنید :

Factory /1 .. 3 /:Setup_Cost;

Factory و Setup_Cost معرف یک آرایه تک بعدی شامل ۳ خانه اند، با این تفاوت که به خانه های Factory مقداره‌ی نمی کنیم و آنها را در عملیات ریاضی شرکت نمی دهیم.

برای مقداره‌ی به متغیر Setup_Cost باید در بخش مقداره‌ی سه عدد به آن نسبت داد که این اعداد به ترتیب در خانه های اول تا سوم آن قرار می گیرند.

Setup_Cost(1)	Setup_Cost(2)	Setup_Cost(3)
---------------	---------------	---------------

نکته قابل ذکر در مورد عناصر مجموعه این است که اسم عناصر می تواند عدد، رشته ای از اعداد و یا حروف باشد.
(فقط باید با حرف شروع شود).

برای آشنایی بیشتر چند مجموعه در پایین آمده است :

Department / Industrial, Mechanic, Civil, Electronic, Computer, Material/: Average;

WAREHOUSE / WH1, WH2, WH3/: CAPACITY;

CUSTOMER / C1, C2, C3, C4/: DEMAND;

مجموعه های وارث

در لینگو نمی توان به صورت مستقیم ماتریس و آرایه های چند بعدی را تعریف کرد. برای تعریف آرایه های چند بعدی از مجموعه های وارث استفاده می کنیم.

ساختار کلی مجموعه های وارث بصورت زیر است :

Sets:

(Set1); / آرایه / اسم مجموعه های وارث

Car/1..3/; (مثال)

متغیرها (Set1, Set2, Set3, ... , Setn) : اسم مجموعه وارث

Demand (Car, City): Dem; (مثال)

Endsets

برای درک بهتر به مثال های کلی زیر توجه کنید :

فرض کنید کارخانه ای سه نوع ماشین مختلف تولید می کند و مشتریان پنج شهر مختلف نیز سفارش های متفاوتی از هر نوع ماشین به کارخانه می دهند. حال زیر برنامه ی زیر را در نظر بگیرید:

Sets:

Car/1..3/;

City/1..5/;

Demand (Car, City): Dem;

Endsets

این زیر برنامه مجموعه ای مستقل از نوع Car به صورت آرایه تک بعدی ۳ خانه ای و مجموعه ای مستقل از نوع City به صورت یک آرایه تک بعدی ۵ خانه ای تولید می کند. (دقت کنید که متغیری از نوع Car یا City نداریم.)

در خط چهارم این زیر برنامه، مجموعه وارث Demand به صورت یک ماتریس 3*5 و ماتریس Dem از نوع Demand تعریف می شود؛ بنابراین در بخش مقداردهی، می بایست ۱۵ عدد به متغیر Dem نسبت داد، که در این حالت مقدار تقاضای شهر j از ماشین i در خانه ای واقع در سطر iام و ستون jام ماتریس Dem قرار می گیرد.

در قسمت داخل اسلش ها (اندیس هایی که با هم ارتباط دارند) می توانیم اندیس هایی را ذکر کنیم که ارتباطشان با هم تعریف شده است. در صورت حذف قسمت بین اسلش ها، فرض بر این است که ارتباط تمام اندیس ها در تمام بعد ها، با هم تعریف شده است و متغیرهای وابسته به مجموعه، با تمام اندیس های خود می توانند مقدار بگیرند. استفاده

از قسمت اندیس هایی که با هم ارتباط دارند به خصوص در مسائل کنترل پروژه کاربرد دارد که خود مبحثی جداگانه است. مثال پیش رو مطالب را بیشتر برای ما روشن می سازد:

SETS:

TASK/A B C D E F G H I J K/: T;

PRED (TASK, TASK) / A,B B,C C,F C,G F,J G,J J,K D,E E,H E,I H,J I,J/;

STATION /1..4/;

TXS (TASK, STATION): X;

ENDSETS

در مثال فوق مجموعه TASK از ۱۱ عضو تشکیل یافته است و مجموعه STATION از ۴ عضو. با این وجود مجموعه PRED به جای $121=11*11$ عضو تنها از ۱۲ عضو مشخص شده، تشکیل شده در حالی که TXS شامل $32=4*8$ عضو می باشد.

توجه کنید که در برنامه های مختلف گاهی مواردی پیش می آید که هنگام کار با مجموعه ها (در تعریف مجموعه ها و یا هنگام استفاده از توابع for و یا sum) برخی از اندیس های مجموعه را می بایست نادیده گرفت و یا هر شرط خاص دیگری را باید با برخی از اندیس های مجموعه ها (و نه تمامی آنها) اعمال کرد؛ در این حالت از عملگرهای مختلف و عبارات شرطی استفاده می کنیم که در فصل های آتی در مورد آنها به تفصیل بحث خواهیم کرد.

بخش داده ها

بخش داده ها به ما این امکان را می دهد که داده ها را از بقیه مدل جدا کنیم. این کار باعث می شود در هنگام تغییر و یا توسعه مدل راحت تر و کم اشتباه تر عمل کنیم. در بخش داده که مابین دو کلمه کلیدی DATA و ENDDATA است می توانیم مقادیر پارامترها و اسم عناصر مجموعه هایی را که در بخش قبلی تعریف شده اند مقداردهی کنیم. به ۳ مثال زیر توجه کنید :

1)

SETS:

SET1 / A, B, C/: X, Y;

ENDSETS

DATA:

X = 1, 2, 3;

Y = 4, 5, 6;

ENDDATA

2)

SETS

SET1 / A, B, C/: X, Y;

ENDSETS

DATA

X, Y = 1, 4,

2, 5,

3, 6;

ENDDATA

3)

SETS

```

SET1 : X, Y;

ENDSETS

DATA

    SET1, X, Y = A 1 4
                B 2 5
                C 3 6;

ENDDATA

```

در هر سه مثال فوق مجموعه SET1 دارای سه عنصر A، B و C است که دو خصیصه X و Y به آن نسبت داده شده و جواب هر سه برابر است با :

$$X(A) = 1, X(B) = 2, X(C) = 3 \quad Y(A) = 4, Y(B) = 5, Y(C) = 6$$

پارامترهای 1×1 را می توان مستقیماً در قسمت داده ها تعریف و مقداردهی کرد. مثلاً :

```

DATA

    INTEREST_RATE = .085;

ENDDATA

```

و یا

```

DATA

    INTEREST_RATE, INFLATION_RATE = .085, .03;

ENDDATA

```

INTEREST_RATE در هر دو مثال فوق برابر 0.085 و در مثال دوم INFLATION_RATE برابر 0.03 است.

تحلیل What if

برخی مواقع مقدار دقیق یک پارامتر را نداریم و می خواهیم برای محدوده ای از اعداد این پارامتر را مقداردهی کرده و مدل را حل کنیم. در این مواقع در قسمت داده ها به جای اینکه مقدار عددی آن پارامتر را وارد کنیم علامت سوال “ ؟ ” قرار می دهیم. در این صورت هر بار که می خواهیم مدل را حل کنیم، لینگو مقدار عددی این پارامتر را از ما می پرسد و سپس شروع به حل مدل می کند. به مثال زیر توجه کنید.

DATA:

INFLATION_RATE = ?

ENDDATA

در این مثال، در هر بار حل مدل، لینگو مقدار INFLATION_RATE را به وسیله پنجره ای از ما می خواهد.

به مثال های زیر در مورد مقداردهی تکی به متغیرها و پارامترها توجه کنید :

SETS

DAYS / MO, TU, WE, TH, FR, SA, SU/: NEEDS;

ENDSETS

DATA:

NEEDS = 20;

ENDDATA

در این مثال همه اعضای NEEDS مقدار ۲۰ می گیرند.

SETS

DAYS / MO, TU, WE, TH, FR, SA, SU/: NEEDS, COST;

ENDSETS

DATA:

NEEDS, COST = 20, 100;

ENDDATA

در این مثال همه اعضای NEEDS مقدار ۲۰ و همه اعضای COST مقدار ۱۰۰ می گیرند.

در بخش داده ها اگر مقدار برخی از متغیرها وارد نشده باشد، آنها به صورت متغیرهای تصمیم در نظر گرفته می شوند نه اعداد ثابت. به مثال زیر توجه کنید :

SETS

YEARS / 1..5 /: CAPACITY;

ENDSETS

DATA

CAPACITY = 34, 34, , , ;

ENDDATA

در این مثال (1) CAPACITY و (2) CAPACITY برابر ۳۴ در نظر گرفته می شوند و عناصر سوم تا پنجم CAPACITY متغیر به حساب می آیند.

بخش جواب اولیه

در بعضی از مسائل، ما جوابی نسبتاً خوب و نزدیک به بهینه را داریم و به دنبال یافتن جواب بهینه و یا بهتر هستیم. در این موارد (خصوصاً در مدل های عدد صحیح و غیر خطی) با دادن جواب اولیه به لینگو، سریعتر به جواب بهینه و یا بهتر خواهیم رسید. این کار که اختیاری می باشد در بخش جواب اولیه و مابین دو کلمه کلیدی INIT و ENDINIT صورت می گیرد.

توابع و عملگرها در لینگو

برای تسهیل مدل سازی، در لینگو توابع و عملگرهای متفاوتی وجود دارند. حتی اگر با شکل کلی استفاده از این توابع آشنایی نداشته باشید، لینگو این شکل کلی را در اختیار شما قرار می دهد. برای این منظور در منوی Edit گزینه Paste Function را انتخاب نمایید، ملاحظه می کنید که لیستی از انواع توابع برای شما نمایش داده می شود. این لیست به شرح زیر می باشد :

- ❖ External Files Function
- ❖ Financial Function
- ❖ Mathematical Function
- ❖ Probability Function
- ❖ Set Function
- ❖ Variable Domain Function
- ❖ Other Function

بررسی تک تک این توابع از سطح کار ما فراتر است؛ همچنین به خاطر عدم نیاز به توابع مالی و احتمالی در برنامه ریزی خطی در اینجا فقط به تشریح برخی توابع دسته پنجم و ششم می پردازیم.

در صورتی که مایلید اطلاعات بیشتری در ارتباط با سایر توابع لینگو بدست آورید در منوی Help گزینه Help Topics را انتخاب نمایید و پنجره محاوره ای باز شده را در حالت Index قرار دهید؛ سپس کلمه Function را تایپ کنید، در این حالت لینگو اطلاعات مفید بسیاری در اختیار شما قرار می دهد).

نکته:

قبل از معرفی توابع به این نکته مهم توجه کنید که در لینگو هنگام به کار بردن توابع، میبایست قبل از نام تابع از علامت @ استفاده کرد.

← در اینجا به بررسی دو تابع SUM و FOR می پردازیم

تابع FOR

این تابع در دسته Set Looping یعنی توابع حلقه ای قرار دارد و شکل کلی آن بصورت زیر می باشد :

@FOR(Set : expression);

در صورت استفاده از عبارات شرطی به صورت زیر عمل کنید :

@FOR(Set | condition: expression);

وجود تابع For در لینگو در واقع یکی از برتری های بارز این زبان می باشد. از این تابع برای نسبت دادن یک ویژگی به تمام یا بخشی از اعضای یک مجموعه استفاده می کنیم. به مثال های زیر توجه کنید :

فرض کنید انباری داریم که در آن هفت قلم کالای متفاوت نگهداری می کنیم:

SETS:

Items /1..7/: Capacity;

ENDSETS

حال فرض کنید ظرفیت انبار محدود بوده و حداکثر ۱۲ واحد از هر قلم می توان در آن نگهداری کرد. برای وارد کردن این محدودیت به صورت زیر عمل می کنیم :

@For(Items (i): Capacity (i) <= 12);

این عبارت در واقع معادل عبارت زیر است :

Capacity (1) <= 12;

Capacity (2) <= 12;

Capacity (3) <= 12;

Capacity (4) <= 12;

Capacity (5) <= 12;

Capacity (6) <= 12;

Capacity (7) <= 12;

از آنجا که تمام متغیرهای به کار رفته در مجموعه Items قرار دارند، عبارت بالا را به صورت زیر نیز می توان نوشت :

@For(Items: Capacity <= 12);

توجه کنید که در صورتی که متغیرهای به کار رفته در For متعلق به مجموعه های متفاوت باشند نمی توان از شکل خلاصه شده بالا استفاده کرد.

حال فرض کنید ظرفیت انبار فقط برای دو قلم کالای آخر (اقلام ۶ و ۷) محدودیت داشته باشد؛ برای وارد کردن این محدودیت از عبارات شرطی و عملگرهای منطقی استفاده می کنیم :

@For(Items (i) | i#GT#5: Capacity (i) <= 12);

تابع SUM

این تابع نیز همانند تابع For در دسته توابع Set Looping قرار دارد و شکل کلی کاربرد آن به صورت زیر می باشد :

@SUM (Set: expression);

در صورت استفاده از عبارات شرطی به صورت زیر عمل کنید :

@SUM (Set | condition: expression);

از این تابع برای محاسبه حاصل جمع اعضای یک مجموعه یا عبارتی شامل اعضای یک مجموعه استفاده می کنیم.
برای روشن شدن مطلب به مثال های زیر توجه کنید :

فرض کنید ۵ مشتری مختلف تقاضاهای متفاوتی از کالای X دارند؛ حال این زیر برنامه را در نظر بگیرید :

SETS:

Customers /1 2 3 4 5/: Demand, cost;

ENDSETS

DATA

Cost = 5 1 3 4 6;

ENDDATA

حال اگر بخواهیم مجموع تقاضاها برای کالای X را محاسبه نماییم، کفایت عبارت زیر را به برنامه بیافزاییم :

Total_Demand = @Sum (Customers (i): Demand (i));

در واقع این عبارت معادل عبارت زیر می باشد :

Total_Demand = Demand (1) + Demand (2) + Demand (3) + Demand (4) + Demand (5);

می توانستیم از عبارت خلاصه شده زیر نیز استفاده کنیم :

Total_Demand = @Sum (Customers: Demand);

بخش سوم « حل مسائل »

✓ انواع مسائل

✓ Case Study – بیان مسئله

✓ تحلیل خروجی

✓ تحلیل حساسیت

لینگو قادر به حل مسائل مختلف بهینه سازی از جمله برنامه های خطی، کوآدراتیک، خطی عدد صحیح، کوآدراتیک عدد صحیح، عدد صحیح خطی خالص، عدد صحیح کوآدراتیک خالص، غیر خطی، غیر خطی عدد صحیح و غیر خطی عدد صحیح خالص می باشد. مدل های فوق توسط لینگو تشخیص داده شده و در گزارش خروجی مدل پس از حل آن به ترتیب فوق LP،QP،ILP،IQP،PILP،PIQP،NLP،INLP،PINLP خواهیم داشت. در اینجا به پیاده سازی و تحلیل یک مدل خطی می پردازیم.

مثال حل یک مدل برنامه ریزی خطی:

شرکت داکوتا فرنیچر، تولید کننده میز تحریر، میز و صندلی است. تولید هر کدام از انواع مبلمان به الوار و دو نوع نیروی کار ماهر در دو زمینه نجاری و پرداخت نیاز دارد. انواع مبلمان به الوار و دو نوع نیروی کار ماهر در دو زمینه نجاری و پرداخت نیاز دارد. مقدار هر یک از منابع مورد نیاز برای تولید هر یک از انواع مبلمان در جدول ذیل آمده است :

منبع	میز تحریر	میز	صندلی
الوار (فوت تخته)	۸	۶	۱
زمان پرداخت (ساعت)	۴	۲	۱/۵
زمان نجاری (ساعت)	۲	۱/۵	۰/۵

در حال حاضر، ۴۸ فوت تخته از الوارها، ۲۰ ساعت زمان پرداخت و ۸ ساعت زمان نجاری موجود است. هر میز تحریر ۶۰ دلار، میز ۳۰ دلار و صندلی ۲۰ دلار به فروش می رسد. با فرض اینکه منابع در دسترس به تازگی تدارک دیده شده باشند، شرکت داکوتا می خواهد کل سود خود را به حداکثر برساند.

متغیرهای تصمیم بصورت ذیل تعریف شده اند :

Desk : تعداد میز تحریرهای تولید شده

Table : تعداد میزهای تولید شده

Chair : تعداد صندلی های تولید شده

شرکت داکوتا به حل مدل خطی زیر نیاز دارد :

$$\text{Max } z = 60 \text{ Desk} + 30 \text{ Table} + 20 \text{ Chair}$$

$$\text{s.t} \quad 8 \text{ Desk} + 6 \text{ Table} + \text{Chair} \leq 48 \quad (\text{محدودیت الوار})$$

$$4 \text{ Desk} + 2 \text{ Table} + 1.5 \text{ Chair} \leq 20 \quad (\text{محدودیت پرداخت})$$

$$2 \text{ Desk} + 1.5 \text{ Table} + 0.5 \text{ Chair} \leq 8 \quad (\text{محدودیت نجاری})$$

Table ≤ 5 (محدودیت تقاضای میز)

Desk, Table, Chair ≥ 0

مدل فوق به زبان لینگو در قسمت Lingo Model وارد می شود :

```
LINGO Model - LINGO1
MAX = 60*DESKS + 30*TABELS + 20*CHAIRS;
8*DESKS + 6*TABELS + CHAIRS < 48;
4*DESKS + 2*TABELS + 1.5*CHAIRS < 20;
2*DESKS + 1.5*TABELS + .5*CHAIRS < 8;
TABELS < 5;
```

با استفاده از گزینه Solve جواب نهایی سیمپلکس بصورت زیر خواهد بود :



Solution Report - Dakota

Global optimal solution found.

Objective value: 280.0000

Total solver iterations: 3

Variable	Value	Reduced Cost
DESKS	2.000000	0.000000
TABELS	0.000000	5.000000
CHAIRS	8.000000	0.000000

Row	Slack or Surplus	Dual Price
1	280.0000	1.000000
2	24.00000	0.000000
3	0.000000	10.00000
4	0.000000	10.00000
5	5.000000	0.000000

تحلیل خروجی

Objective value: 280.0000

مشخص می کند که مقدار بهینه تابع هدف ۲۸۰ است.

Value مقدار متغیر در جواب بهینه مدل خطی را می دهد. بنابراین جواب بهینه داکوتا تولید ۲ میز تحریر، صفر میز، و ۸ صندلی است.

Reduced Cost ضریب متغیر در سطر صفر جدول بهینه را می دهد (مسئله ماکزیمم سازی). همانگونه که می دانید باید هزینه کاهش یافته هر متغیر پایه برابر صفر باشد. برای یک متغیر غیر پایه x_j ، هزینه کاهش یافته مقداری است که اگر x_j ، ۱ واحد اضافه شود، به آن اندازه، جواب بهینه کاهش می یابد (درحالی که بقیه متغیرهای غیر پایه برابر صفر بمانند). در خروجی لینگو برای مسئله داکوتا، هزینه کاهش یافته میز برابر ۵ است. این یعنی اگر داکوتا بخواهد یک میز تولید کند، درآمد به اندازه ۵ دلار کاهش خواهد یافت.

در قسمت بعدی از خروجی یا Report مشاهده می کنیم که هر خط مدل با شماره ای مشخص شده و در مقابل آن دو عبارت Slack or Surplus و Dual Price مشاهده می شود :

Slack or Surplus مقدار کمبود یا اضافی (مازاد) در جواب بهینه را نشان می دهد. بنابراین ۲۴ فوت تخته الوار و ۵ عدد میز به صورت مازاد وجود خواهد داشت.

Dual Price یا قیمت سایه مقداری است که اگر سمت راست محدودیت، یک واحد اضافه شود، به آن اندازه جواب بهینه، بهبود می یابد (با فرض اینکه این تغییرات باعث از دست رفتن پایه بهینه فعلی نمی شوند). اگر بعد از یک تغییر در طرف راست محدودیت، پایه فعلی دیگر بهینه نماند، قیمت های سایه محدودیت ها ممکن است تغییر کند.

تحلیل حساسیت

پس از حل مدل توسط Solve با استفاده از گزینه Range موجود در منوی LINGO گزارش خروجی تحلیل حساسیت مدل را مشاهده می کنیم، در صورت مشاهده خطا وارد LINGO Option شده و گزینه Dual Computations از قسمت General Solver را در حالت سوم قرار دهید.

Range Report - Dakota			
Ranges in which the basis is unchanged:			
Objective Coefficient Ranges			
	Current	Allowable	Allowable
Variable	Coefficient	Increase	Decrease
DESKS	60.00000	20.00000	4.000000
TABELS	30.00000	5.000000	INFINITY
CHAIRS	20.00000	2.500000	5.000000
Righthand Side Ranges			
Row	Current	Allowable	Allowable
	RHS	Increase	Decrease
2	48.00000	INFINITY	24.00000
3	20.00000	4.000000	4.000000
4	8.000000	2.000000	1.333333
5	5.000000	INFINITY	5.000000

تحلیل حساسیت نشان می دهد که ضرایب تابع تا چه میزان می توانند افزایش یا کاهش یابند بدون اینکه پایه بهینه مسئله (مجموعه متغیرهای غیر صفر) عوض شود. تحلیل حساسیت شامل دو بخش است، یکی مربوط به تابع هدف و دیگری مربوط به محدودیت ها، که در هر کدام به تفکیک مقدار واقعی و مقادیر کاهش و افزایش قید شده است. در پایان با استفاده از گزینه Solution در منوی LINGO می توانید نمایش گرافیکی از هر یک از متغیرهای موجود در مدل را داشته باشید.

پایان

برای مهندسان بن بست
بی معناست
یا راهی خواهند یافت
یا راهی خواهند ساخت