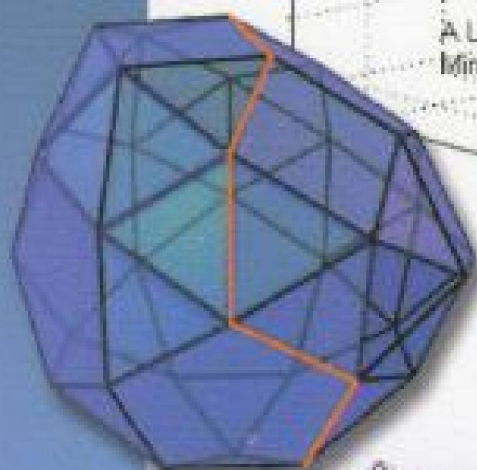
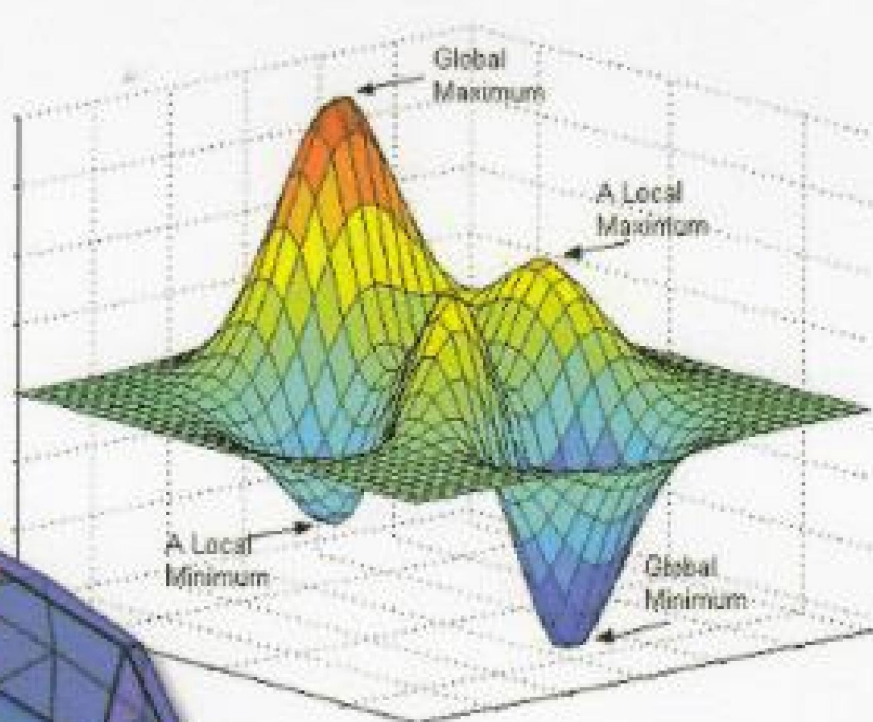


راهنمای کاربردی نرم افزار

لینگو

ابزار مدل سازی و بهینه یابی ریاضی
در رویکرد حل مساله



انتشارات آراد

ترجمه و نگارش: وحید آجورلو

راهنمای کاربردی نرم افزار لینگو

(ابزار مدل سازی و بهینه یابی ریاضی در رویکرد حل مساله)

ترجمه و گردآوری: وحید آجورلو

فایل پی دی اف فصل نخست این کتاب، برای بهره برداری دانشجویان و علاقمندان به طور رایگان منتشر شده است. علاقمندان به تهیه ی این کتاب (در ۲۵۷ صفحه و با قیمت ۱۲۵,۰۰۰ ریال) می توانند به انتشارات آراد (۶۶۴۸۲۲۲۶) مراجعه نمایند.

در صورت فرید بیش از ۱۰ جلد (با ۲۰٪ تخفیف) و همچنین در صورت نیاز به آموزش و مشاوره، با ای میل زیر مکاتبه فرمایید.

مؤلف
v_ajorloo@yahoo.com

کلیه حقوق این کتاب متعلق به مؤلف است.

ارسال این فایل به صورت الکترونیکی بلامانع بوده ولی چاپ، تکثیر یا بازنشر آن (بر روی کاغذ یا وبلاگ ها و وبسایت ها) مجاز نیست.

فهرست

دیباچه

پیش‌گفتار

۱.....	فصل یکم - آغاز کار با لینگو.....
۲.....	وارد کردن یک مدل.....
۷.....	ساختار دستوری لینگو.....
۹.....	پنجره‌ی وضعیت راه‌یاب (حل‌کننده).....
۲۳.....	بررسی جواب.....
۲۶.....	کاربرد زبان مدل‌سازی.....
۳۴.....	تعریف مجموعه‌ها.....
۳۶.....	ورود داده‌ها.....
۴۲.....	دیگر ویژگی‌های زبان مدل‌سازی.....
۴۹.....	فصل دوم - کاربرد مجموعه‌ها.....
۵۰.....	چرا مجموعه‌ها؟.....
۵۱.....	انواع مجموعه‌ها.....
	Error! Bookmark not defined. بخش مجموعه‌ها
	Error! Bookmark not defined. تعریف مجموعه‌ی مستقل
	Error! Bookmark not defined. تعریف مجموعه‌ی وابسته
	Error! Bookmark not defined. بخش داده‌ها
	Error! Bookmark not defined. تابع‌های چرخه‌ای
۶۵.....	تابع چرخه‌ای @SUM.....
	Error! Bookmark not defined. تابع‌های چرخه‌ای @MIN و @MAX.....
	Error! Bookmark not defined. تابع چرخه‌ای @FOR.....
۷۰.....	تابع چرخه‌ای @PROD.....

۷۱	تابع‌های چرخه‌ای تودرتو.....
۷۳	فصل سوم - دامنه متغیرها.....
۷۴	متغیرهای عدد صحیح.....
۸۷	اولویت‌های متغیرها.....
۸۸	متغیرهای آزاد.....
۹۵	متغیرهای کران‌دار.....
	Error! Bookmark not defined.SOS
۱۰۱	عدد اصلی مجموعه.....
۱۰۲	متغیرهای نیمه‌پیوسته.....
۱۰۵	فصل چهارم - بخش‌های داده‌ها، مقداردهی و محاسبات.....
۱۰۶	بخش داده‌ها.....
۱۰۶	نوشتار.....
۱۰۸	پارامترها.....
۱۰۸	تحلیل «چه می‌شد اگر».....
۱۰۹	مقداردهی یک مشخصه با یک مقدار یکسان.....
۱۱۱	بخش مقداردهی آغازین.....
۱۱۳	بخش محاسبات.....
۱۱۹	فصل پنجم - فرمان‌های ویندوز.....
۱۲۱	خلاصه‌ای از فرمان‌ها.....
۱۲۴	تشریح فرمان‌ها.....
۱۲۴	منوی <i>File</i>
۱۴۳	منوی <i>Edit</i>
۱۵۷	منوی <i>LINGO</i>
۲۴۳	منوهای <i>Window</i> و <i>Help</i>

دیباچه

به نام خداوند جان و خرد

دست کم به دو دلیل، همواره نیازمند برنامه‌ریزی هستیم: نخست آنکه سرمایه‌ها (پول، زمان، ماشین‌آلات، امکانات، بودجه خانوار، انرژی و ...) نامحدود نیستند و ما می‌خواهیم با استفاده از ابزار «برنامه‌ریزی» مدیریت مناسبی بر روی سرمایه‌ها داشته باشیم. دوم اینکه، می‌خواهیم با به‌کارگیری سرمایه‌ها به یک هدف دست بیابیم؛ به یک هدف «بهینه». بنابراین، شیوه برنامه‌ریزی ما تابع هدف ما خواهد بود.

در جهان واقعی، به ویژه در مسایل صنعت و بازار، تعداد عوامل موثر بر برنامه‌ریزی به اندازه‌ای زیاد و گوناگون هستند که دیگر به صورت دستی و ذهنی نمی‌توان آنها را مدیریت کرد. از سوی دیگر، همان تعداد عوامل هم پارامترهای ثابتی نبوده و دایما در حال تغییر هستند. برنامه‌ریزی ریاضی کمک می‌کند تا عبارت‌های کلامی را به عبارت‌های ریاضی تبدیل کرده و با حل برخی معادلات ریاضی، بهترین پاسخ را برای مساله پیدا کنیم.

دیگر سودمندی برنامه‌ریزی ریاضی آن است که مدل یک بار فرموله می‌شود و تا هنگامی که دگرگونی بنیادی در مدل پدید نیاید، لازم نیست که آن را بارها تعریف نمود. هر تغییری در هر معادله یا پارامتر و یا سایر عوامل را به راحتی می‌توان در مدل بازنگری نموده و آن را دوباره حل کرد.

برنامه‌ریزی ریاضی، تنها یک مانع پیش رو دارد (یا بهتر است بگوییم داشت): حل کردن یک مساله ریاضی در جهان واقعی کمی دشوار بوده و بدون نرم‌افزار، چندان کاربردی نیست. خوش‌بختانه، امروزه نرم‌افزارهای بسیاری برای همین منظور گسترش یافته‌اند که مسایل واقعی را در چند ثانیه حل می‌کنند. به علاوه، هر تغییری در برنامه و مدل به راحتی قابل بازنگری بوده و نتایج تغییرات را می‌توان با همدیگر مقایسه کرد.

از میان نرم‌افزارهای گوناگونی که برای بهینه‌سازی پدید آمده‌اند، به جرات می‌توان گفت که لینگو یکی از کارآمدترین، راحت‌ترین و توانمندترین‌ها در گونه‌ی خود است. لینگو علاوه

بر حل مساله‌ها، قدرت بالایی در تحلیل آنها دارد. وجود تابع‌های گوناگون آماری، ریاضی، احتمالی و ... از دیگر توانمندی‌های لینگو است. لینگو می‌تواند دامنه‌ی گسترده‌ای از مدل‌های بهینه‌سازی ریاضی را از جمله برنامه‌ریزی خطی، برنامه‌ریزی غیرخطی، برنامه‌ریزی عدد صحیح و گونه‌های دیگری از مدل‌های برنامه‌ریزی ریاضی را حل کند.

یکی دیگر از ویژگی‌های لینگو در مقایسه با دیگر برنامه‌های مشابه، کاربری آسان آن است. کاربر می‌تواند عبارت‌های ریاضی را به شکلی نزدیک به همان صورت ریاضی خود وارد کرده و آنها را ویرایش نماید. محیط کاری لینگو به قدری کاربرپسند است که در کمتر از یک نیم‌روز می‌توان با آن آشنا شده، مدل‌های ساده را به راحتی و تنها با استفاده از چهار عمل اصلی ریاضی فرموله کرده و نتایج بسیار جالبی به دست آورد. البته لینگو توانمندی‌های برنامه‌نویسی، کار با توابع، مبادله‌ی داده‌ها با سایر برنامه‌ها و بسیاری امکانات دیگر هم دارد که در کتاب راهنمای آن تشریح شده‌اند.

دانشجویان و فارغ‌التحصیلان رشته‌ها و گرایش‌های مختلف ریاضی، مهندسی صنایع، مدیریت، مهندسی، اقتصاد، بازرگانی و کشاورزی، واحدهای درسی گوناگونی در ارتباط با برنامه‌ریزی ریاضی می‌گذرانند. بسیاری از این افراد به دلیل همان یک مانع پیش‌رو (دشواری در حل مسائل ریاضی به روش دستی) و بدون استفاده از نرم‌افزار مناسب نمی‌توانند توانایی‌های سودمند این روش‌ها را در مسائل واقعی به کار بگیرند. اگر مدیران و کارشناسان، از سادگی کار با چنین نرم‌افزاری آگاه شوند، به یقین جویای آن خواهند بود.

از سوی دیگر، برنامه‌ریزی ریاضی و مفهوم «بهینه‌سازی»، چیزی است که در هر صنف و صنعتی کاربرد دارد. از دولت و وزارتخانه‌ها و سازمان‌های سیاست‌گذار و متولی برنامه‌ریزی کلان گرفته تا شرکت‌ها، موسسات و سازمان‌های بزرگ و کوچک؛ از دانشگاه و مراکز علمی تا کارخانجات صنعتی و تولیدی؛ از یک بیمارستان، بانک یا موسسه خدماتی تا یک بندر، فرودگاه و شرکت حمل و نقل؛ در هر جا که نیاز به برنامه‌ریزی باشد، برنامه‌ریزی ریاضی کاربرد دارد.

نگارنده‌ی کتاب، باور پیدا کرده است که به شرط مدل‌سازی درست مساله، نتایج حاصل از حل مسائل با این نرم‌افزار بسیار دقیق‌تر و واقعی‌تر از روش‌های مبتنی بر حدس و گمان،

تجربه و سلیقه بوده و بی شک اقدام ارزشمندی در کمک به همه‌ی کسانی خواهد بود که می‌خواهند روش‌های علمی را جای‌گزین روش‌های سنتی نمایند. با این باور، نگارنده اقدام به ترجمه‌ی راهنمای کار با این نرم‌افزار نموده و امیدوار است انتشار این کتاب موجب ترویج فرهنگ برنامه‌ریزی در کشور شود.

فصل یکم (آغاز کار با لینگو)، در واقع خلاصه‌ای از همه‌ی کتاب است. در این بخش خواهید دید که لینگو چیست و چه توانایی‌هایی دارد؛ چگونه می‌توان یک مدل جدید ایجاد کرد؛ چگونه می‌توان داده‌ها را در مدل وارد کرد، کار با مجموعه‌ها چگونه است؛ چگونه می‌توان مدل را حل کرد و از آن خروجی‌های گوناگون گرفت. از این رو، حتی کسی که تاکنون تجربه کار با لینگو یا دیگر نرم‌افزارهای برنامه‌ریزی را هم نداشته است به شرط آشنایی با مفاهیم و کلیات برنامه‌ریزی ریاضی - تنها با خواندن فصل یکم این کتاب و بی‌نیاز از دانستن فرمان‌ها، تابع‌ها و دیگر ویژگی‌های لینگو، می‌تواند یک مدل واقعی را تدوین کرده و جواب آن را به دست آورد.

فصل دوم (کاربرد مجموعه‌ها)، یکی از مهم‌ترین توانمندی‌های لینگو را در مدیریت داده‌ها و دسته‌بندی آنها بیان می‌کند. توانایی لینگو در به‌کارگیری مجموعه‌ها و اعضای آنها به راستی شگفت‌انگیز بوده و مدل‌های بسیار بزرگ و پیچیده را به سادگی مدیریت می‌نماید.

در فصل سوم (دامنه‌ی متغیرها)، تابع‌ها و روش‌هایی برای تعیین دامنه‌ی متغیرها معرفی می‌شوند. فصل چهارم (بخش‌های داده‌ها، مقداردهی و محاسبات) نشان می‌دهد که یک مدل لینگو می‌تواند از چند بخش گوناگون تشکیل شود تا مدیریت کلی مدل و دنبال کردن آن آسان‌تر باشد. در فصل پنجم (فرمان‌های ویندوز)، محیط کار این نرم‌افزار و چگونگی کار با آن بیشتر معرفی می‌شوند. در بیش‌تر موارد، کاربر عادی نیازی به تغییر گزینه‌ها در منوها ندارد؛ چون بهترین گزینه‌ها همان حالت‌های پیش‌گزیده هستند.

بزرگواران پیش‌کسوت، برگردان‌های گوناگونی برای واژه‌های بیگانه برگزیده‌اند. در ترجمه‌ی این کتاب، تلاش شده است واژگانی که نزد کارشناسان رایج‌تر و هم‌چنین پارسی‌تر باشد گزیده شده و برای پرهیز از هر گونه شبهه‌ای، در هر جا که نیاز بوده هم‌ارز انگلیسی آن

هم در پاورقی آورده شده است. یادآوری می شود که این کتاب برای آموزش برنامه ریزی ریاضی نیست. کاربر باید با مفاهیم برنامه ریزی ریاضی آشنایی پایه ای داشته باشد؛ هرچند، خود کتاب نیز راهنمای بسیار خوبی بوده و در هر جا که نیاز بوده است، خلاصه ای از آن مفهوم مورد نیاز معرفی شده است.

«راهنمای کاربران لینگو» که همراه با نرم افزار ارائه می شود، در بیش از ۸۳۰ صفحه به زبان انگلیسی بوده و از حوصله ی کاربر عادی فراتر است. کتاب پیش رو، درخور کاربر مبتدی و میانه بوده و همه ی نیازهای وی در آشنایی و کار با این نرم افزار را به خوبی پاسخ خواهد داد. مطالبی که از ترجمه ی آن ها پرهیز شده است مورد نیاز کاربران روزمره ی این نرم افزار نبوده و شایسته ی کاربران حرفه ای و برنامه نویسان است که می توانند به راهنمای درونی نرم افزار هم مراجعه نمایند.

گردآوری این کتاب، نیازمند وقت زیادی بود که می بایستی برای خانواده (پدر، مادر، همسر و فرزندان) صرف می شد. به این بهانه، ارزش معنوی این نگاشته را به ایشان تقدیم می دارم تا عذر کوچکی برای کوتاهی در این مسوولیت بیابم. از اساتید، صاحب نظران و خوانندگان گرامی خواهشمندم مرا شایسته ی نظرات ارزشمند خود به شمار آورند.

از خدا خواهیم توفیق ادب - وحید آجورلو

v_ajorloo@yahoo.com

پیش‌گفتار

لینگو ابزاری جامع برای مدل‌سازی و حل مدل‌های بهینه‌سازی ریاضی به گونه‌ای آسان و کارآمد است. لینگو بسته‌ی یکپارچه‌ای است که شامل یک زبان توانمند برای فرموله کردن مدل‌های بهینه‌سازی، یک محیط خوش‌سیما برای ایجاد و ویرایش مساله‌ها، و مجموعه‌ای از راه‌یاب‌های درونی (توکار^۱) بوده و قادر به حل بیش‌تر مدل‌های بهینه‌سازی است. ویژگی‌های اصلی لینگو عبارتند از:

زبان مدل‌سازی مجری

لینگو یک زبان مدل‌سازی توانمند مبتنی بر مجموعه‌ها دارد که کاربران را قادر می‌سازد مدل‌های برنامه‌ریزی ریاضی را به گونه‌ای کامل و کارآمد بیان نمایند. قابلیت‌های واژه‌نگار درونی لینگو اجازه می‌دهد که مدل‌های گوناگون به دفعات حل شوند.

سادگی کار با داده‌ها

لینگو زمان را حفظ کرده و نگرانی شما در مدیریت داده‌ها را برطرف می‌کند. می‌توانید مدل‌هایی ایجاد کنید که اطلاعات را به طور مستقیم از پایگاه داده‌ها یا صفحه‌گسترده‌ها برداشت کنند. هم‌چنین لینگو می‌تواند جواب^۲ را به یک پایگاه داده یا صفحه‌گسترده بفرستد تا تهیه گزارش در برنامه‌ی مورد نظر شما آسان‌تر شود. جداسازی کامل مدل و داده‌ها، نگهداشت مدل و تعمیم آن را آسان‌تر خواهد کرد.

1- Built-in solvers
2- Solution

مدل تعاملی یا برنامه‌های تبدلی

می‌توانید مدل را درون لینگو ایجاد و حل کنید؛ یا این‌که لینگو را از برنامه‌هایی که خودتان نوشته‌اید فراخوان کنید. هنگام کار با مدل‌های تعاملی، لینگو یک محیط کامل مدل‌سازی در اختیار شما می‌گذارد تا مدل خود را ایجاد، حل و تجزیه و تحلیل کنید.

هنگام کار با برنامه‌های تبدلی، لینگو رابط‌های *DLL*^۱ و *OLE*^۲ دارد که می‌توانند از برنامه‌های دیگر فراخوان شوند. همچنین، لینگو می‌تواند به طور مستقیم از یک ماکروی اکسل یا پایگاه داده فراخوان شود. لینگو شامل مثال‌های برنامه‌نویسی برای *C/C++*، *FORTRAN*، *Java*، *C#.NET*، *VB.NET*، *ASP.NET*، *Visual Basic*، *Delphi* و *Excel* است.

مستندات و راهنمای گسترده

لینگو همه‌ی ابزارهای لازم برای ایجاد و اجرای سریع را ارائه می‌دهد. راهنمای کاربران لینگو، دستورات و ویژگی‌های برنامه را به طور کامل شرح می‌دهد. همچنین -همراه با برنامه‌ی لینگو- مدل‌های بسیار گوناگونی از مساله‌های واقعی ارائه شده‌اند که می‌توانید آن‌ها را ویرایش کرده و مساله‌ی خودتان را با کمک آن‌ها حل نمایید.

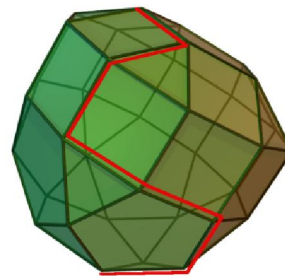
ابزارها و راه‌یاب‌های قدرتمند

برای بهینه‌سازی انواع مساله‌های خطی، غیرخطی، عدد صحیح، تصادفی و حالت‌های عدم قطعیت، لینگو مجموعه‌ی گسترده‌ای از راه‌یاب‌های سریع را در درون خود دارد. نیازی نیست که نوع مساله‌تان را برای لینگو مشخص کنید؛ لینگو خودش آن را تشخیص داده و شایسته‌ترین راه‌یاب را برای آن برمی‌گزیند. لینگو همچنین ابزارهای توانمندی برای تجزیه و تحلیل مدل‌ها و جواب‌های آن‌ها دارد.

1- Dynamic-link library

2- Object linking and embedding

فصل یکم آغاز کار با لینگو



لینگو چیست؟

لینگو ابزار ساده‌ای است که برای بهینه‌سازی خطی و غیرخطی کاربرد دارد. با این ابزار می‌توانید مدل‌های بزرگ را به سادگی و با دقت فرموله کرده، آن‌ها را حل نموده و جواب را تجزیه و تحلیل نمایید. بهینه‌سازی به شما کمک می‌کند تا برای مساله‌تان پاسخی را به دست آورید که بهترین جواب باشد؛ بالاترین سود، خروجی یا رضایت را به دست آورید؛ و یا، پایین‌ترین هزینه، ضایعات یا نارضایتی را داشته باشید.

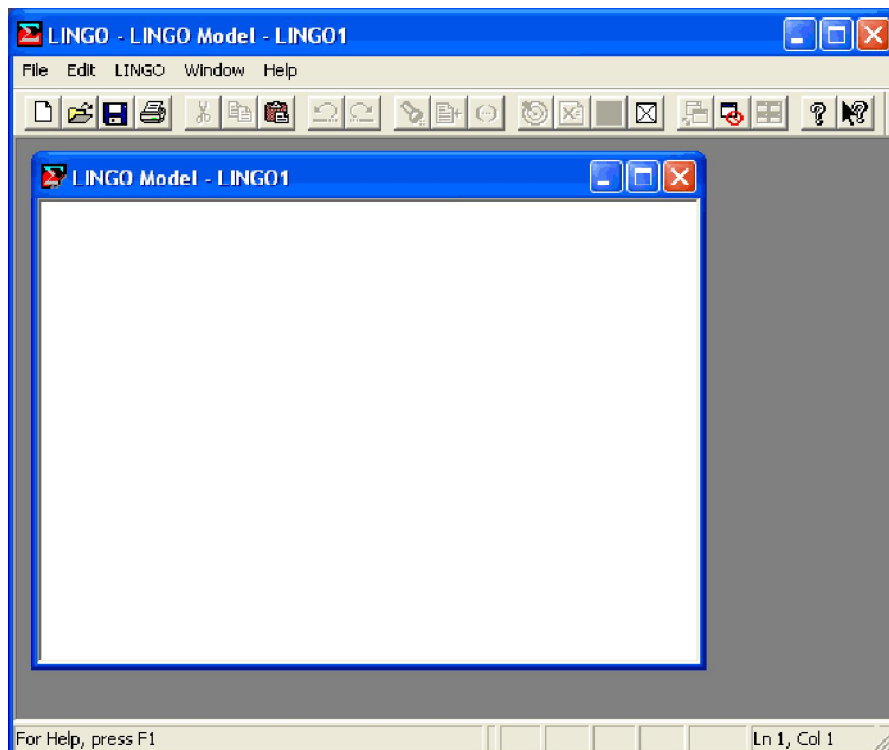
در بیش‌تر این گونه مسایل، به دنبال افزایش کارایی در منابع هستیم که عبارتند از: پول، زمان، ماشین‌آلات، نیروی انسانی، موجودی‌ها و امثال این‌ها. مسایل بهینه‌سازی را با توجه به روابط بین متغیرهای مساله، به صورت خطی یا غیرخطی دسته‌بندی می‌کنند.

وارد کردن یک مدل

آغاز لینگو

در این جا خواهیم دید که چگونه می توان یک مدل کوچک را در ویندوز وارد کرده و آن را حل کنیم. متن معادلات مدل بستگی به پلتفرم خاصی نداشته و در هر سیستمی قابل تشخیص است. با این حال، توجه داشته باشید که نحوه ورود مدل در پلتفرم های غیرویندوز اندکی متفاوت است.

هنگامی که لینگو را در ویندوز اجرا کنید، صفحه ای مشابه زیر خواهید دید:



پنجره بیرونی که عنوان *LINGO* دارد، پنجره اصلی است. سایر پنجره‌ها درون این پنجره قرار می‌گیرند. هم‌چنین، نوار ابزار^۱ و منوها در بالای همین پنجره اصلی دیده می‌شوند. در بخش‌های بعدی، نوار ابزار و منوها به طور مفصل معرفی خواهند شد. در حاشیه‌ی پایینی پنجره‌ی اصلی، نوار وضعیت^۲ قرار دارد که با توجه به وضعیت جاری لینگو، اطلاعات مختلفی را نشان خواهد داد. با دنبال کردن مسیر *LINGO|Options*، می‌توان هم نوار ابزار و هم نوار وضعیت را غیرفعال کرد.

پنجره‌ی کوچک‌تر^۳ که در این جا عنوان *LINGO Model - LINGO1* را دارد، یک پنجره‌ی خالی است. در ادامه، یک مدل نمونه را به طور مستقیم در همین پنجره وارد خواهیم کرد.

ایجاد یک مدل لینگو در ویندوز

تعریف مساله

در این مدل نمونه، مثال کوچکی از ترکیب محصول (سبد بهینه‌ی محصول) را ایجاد خواهیم کرد. فرض کنید شرکت کامپیوکوییک^۳، دو مدل رایانه‌ی «استاندارد» و «توربو» تولید می‌کند. فروش هر دستگاه رایانه‌ی استاندارد برای کامپیوکوییک ۱۰۰ دلار سود دارد؛ در حالی که فروش هر دستگاه رایانه‌ی توربو ۱۵۰ دلار سود دارد. خط تولید رایانه‌ی استاندارد، توانایی تولید حداکثر ۱۰۰ دستگاه در روز را دارد و خط تولید رایانه‌ی توربو، می‌تواند تا ۱۲۰ دستگاه در روز تولید داشته باشد. از طرفی، این شرکت

1- Toolbar
2- Status bar
3- CompuQuick

نیروی کار محدودی داشته و روزانه ۱۶۰ نفر ساعت نیرو در اختیار دارد. هر دستگاه رایانه‌ی استاندارد، نیازمند ۱ ساعت نیروی کار است؛ در حالی که هر دستگاه رایانه‌ی توربو به ۲ ساعت نیروی کار احتیاج دارد. مساله‌ی شرکت کامپیوکوییک این است که روزانه چه تعداد از هر نوع رایانه تولید کند تا سودآوری خود را بیشینه کند بدون این‌که از توانمندی خط و نیروی کار خود فراتر برود.

به طور کلی، یک مدل بهینه‌سازی^۱ از سه بخش زیر تشکیل می‌شود:

تابع هدف^۲: تابع هدف، فرمولی است که به طور دقیق بیان می‌کند چه چیزی باید بهینه شود. در مدل‌های تجاری، تابع هدف معمولاً یا یک «تابع سود» است که شما می‌خواهید بیشینه شود؛ یا یک «تابع هزینه» است که می‌خواهید کمینه شود. یک مدل، می‌تواند حداکثر یک تابع هدف داشته باشد. در مثال کامپیوکوییک، تابع هدف، سود شرکت را به صورت تابعی از تعداد رایانه‌های استاندارد و توربو محاسبه خواهد کرد.

متغیرها^۳: متغیرها، کمیت‌هایی تحت کنترل شما هستند. شما می‌خواهید تعیین کنید که بهترین مقدار هر متغیر چه قدر است. به همین دلیل، متغیرها را برخی اوقات «متغیرهای تصمیم» می‌نامند. هدف بهینه‌سازی آن است که مقادیر متغیرهای مدل به گونه‌ای انتخاب شوند که ضمن رعایت محدودیت‌ها، بهترین مقدار برای تابع هدف به دست آید. در این مثال، ما دو متغیر خواهیم داشت؛ یکی مربوط به تعداد رایانه‌های استاندارد و دیگری مربوط به تعداد رایانه‌های توربو.

محدودیت‌ها^۴: تقریباً بدون استثنا، برخی محدودیت‌ها برای متغیرها وجود خواهند داشت. حداقل یکی از منابع محدودیت دارند (زمان، مواد خام، بودجه و امثال آن). این

1- Optimization model
2- Objective Function
3- Variables
4- Constraints

محدودیت‌ها به صورت فرمول‌هایی بیان می‌شوند که تابعی از متغیرهای مدل هستند. این فرمول‌ها محدودیت نامیده می‌شوند؛ چراکه دامنه مقداری متغیرها را محدود می‌کنند. در مثال کامپیوکوییک، برای هر خط تولید یک محدودیت خواهیم داشت و یک محدودیت هم برای نیروی کار داریم.

وارد کردن مدل

اکنون تابع هدف مدل خود را بنا خواهیم کرد. متغیرهای *STANDARD* و *TURBO* را به ترتیب برای تعداد رایانه‌های استاندارد و تعداد رایانه‌های توربو که باید تولید شوند به کار می‌گیریم. هدف کامپیوکوییک این است که سود کل را بیشینه کند. برای محاسبه‌ی سود کل، حاشیه سود هر مدل از رایانه‌ها را در تعداد تولید شده‌ی آن مدل ضرب کرده و با یکدیگر جمع می‌کنیم. در پایان، با قرار دادن عبارت "*MAX=*" در جلوی تابع هدف، برای لینگو مشخص می‌کنیم که به دنبال بیشینه کردن تابع هدف هستیم. بنابراین، تابع هدف ما در نخستین خط از پنجره‌ی مدل به صورت زیر نوشته می‌شود:

$$MAX = 100 * STANDARD + 150 * TURBO;$$

یادداشت: در لینگو، هر عبارت ریاضی با یک نقطه‌ویرگول (;) پایان می‌پذیرد. بدون آن‌ها مدل شما مل نخواهد شد. برای اطلاعات بیشتر در مورد سافزار لینگو به مثال‌های زیر توجه کنید.

اکنون باید محدودیت‌های مربوط به توانمندی خطوط تولیدی و نیروی کار را وارد کنیم. تعداد رایانه‌های استاندارد و توربو باید محدود به توانمندی خطوط تولید (یعنی به ترتیب ۱۰۰ و ۱۲۰ دستگاه در روز) شوند. به این منظور، دو محدودیت زیر را درست پس از تابع هدف وارد می‌کنیم:

$$STANDARD \leq 100;$$

$$TURBO \leq 120;$$

محدودیت اول می گوید که تعداد رایانه های استاندارد که به طور روزانه تولید می شوند باید کوچکتر یا برابر ۱۰۰ دستگاه باشند. به همین ترتیب، محدودیت دوم تعداد رایانه های توربو را به ۱۲۰ دستگاه در روز محدود می کند.

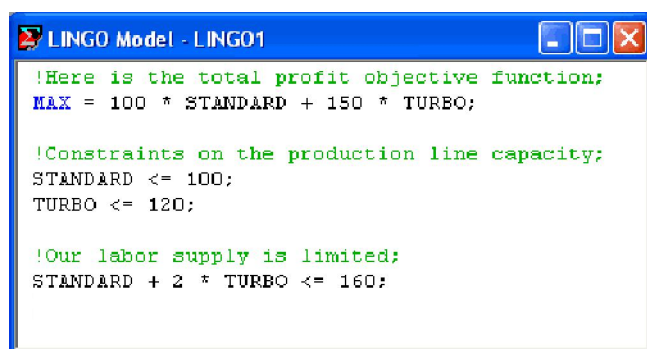
یادداشت: از آنجا که بیشترین صفه کلیدها علامت کوچکتر یا برابر ($\leq$) را ندارند، لینگو به طور قراردادی، دو نویسه $\leq$ را به جای $\leq$ در نظر می گیرد. البته علامت $<$ به تنهایی هم به مفهوم کوچکتر یا برابر در نظر گرفته می شود. همچنین، $>$ یا $>$ به مفهوم بزرگتر یا برابر در نظر گرفته می شوند.

آخرین محدودیت که در مورد نیروی کار است، می تواند به صورت زیر نوشته شود:

$$STANDARD + 2 * TURBO \leq 160;$$

یعنی تعداد ساعت های نیروی کار که برای تولید روزانه ی هر دو مدل رایانه ها تخصیص داده می شوند باید کوچکتر یا برابر نیروی کار موجود (۱۶۰ ساعت) باشند.

پس از وارد کردن فرمول های بالا و همچنین نوشتن یادداشت ها (که قابلیت خواندن مدل را بهبود می دهند)، پنجره ی مدل شما باید مانند این باشد:



```

LINGO Model - LINGO1

!Here is the total profit objective function;
MAX = 100 * STANDARD + 150 * TURBO;

!Constraints on the production line capacity;
STANDARD <= 100;
TURBO <= 120;

!Our labor supply is limited;
STANDARD + 2 * TURBO <= 160;
  
```

ساختار دستوری^۱ لینگو

یک عبارت^۲ می‌تواند به هر تعداد خط شکسته شود، ولی عبارت باید با یک نقطه‌ویرگول پایان پذیرد. به عنوان مثال، می‌توانستیم تابع هدف فوق را به جای یک خط در دو خط ادامه دهیم:

```
MAX = 100 * STANDARD
+ 150 * TURBO;
```

هم‌چنین برخی یادداشت‌ها را وارد کرده‌ایم تا قابلیت خواندن مدل را بهبود دهیم. یادداشت‌ها با علامت تعجب (!) شروع شده و با نقطه‌ویرگول (;) پایان می‌پذیرند. لینگو تمام متن‌های بین علامت تعجب و نقطه‌ویرگول را نادیده می‌گیرد. یادداشت‌ها می‌توانند بیش از یک خط باشند و می‌توانند درون سایر عبارت‌ها هم قرار گیرند. برای مثال:

```
X = 1.5 * Y + Z / 2 * Y; !This is a comment;
X = 1.5 * !This is a comment in the middle
of a constraint; Y + Z / 2 * Y;
```

شاید توجه داشته‌اید که نام متغیرها را با حروف درشت نوشتیم. این کار الزامی نیست. لینگو در نام‌گذاری متغیرها بین حروف درشت و ریز تمایزی قایل نمی‌شود. از این رو، همه‌ی متغیرهای زیر یکسان فرض خواهند شد:

```
TURBO
Turbo
turbo
```

برای نام‌گذاری متغیرها در لینگو، همه‌ی نام‌ها باید با یک نویسه‌ی حرفی (A-Z) آغاز شوند. نویسه‌های بعدی می‌توانند حرف، عدد (0-9) یا خط زیرین (_) باشند. نام یک متغیر می‌تواند تا ۶۴ نویسه طول داشته باشد.

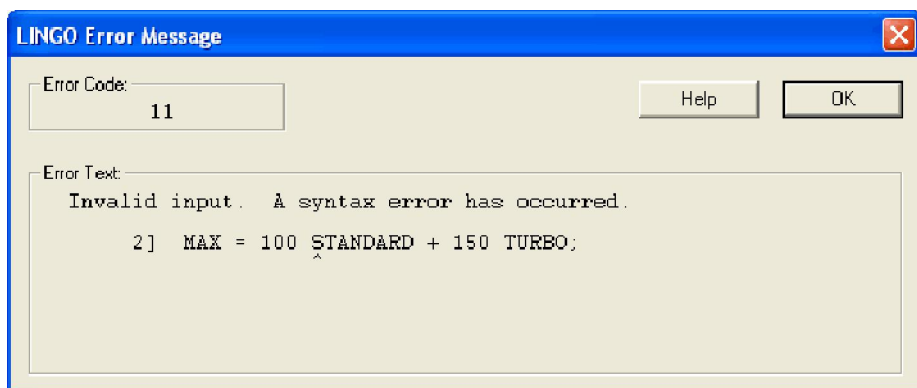
1- Syntax
2- Expression

یک ویژگی مهم که در ادامه متوجه آن خواهید شد این است که ویرایش گر لینگو، آگاه به دستوراتی است که وارد می کنید. به دیگر سخن، هنگامی که لینگو با کلیدواژه های خودش مواجه می شود، آن ها را به رنگ آبی نشان می دهد. یادداشت ها به رنگ سبز و بقیه عبارت ها به رنگ مشکی دیده می شوند. هم چنین، هنگامی که مکان نما را بلافاصله بعد از یک پرانتز قرار می دهید، پرانتزهای متناظر به رنگ قرمز برجسته می شوند. از این ویژگی سودمند، در خطایابی مدل تان بهره خواهید جست.

حل مدل

خطاهای دستوری

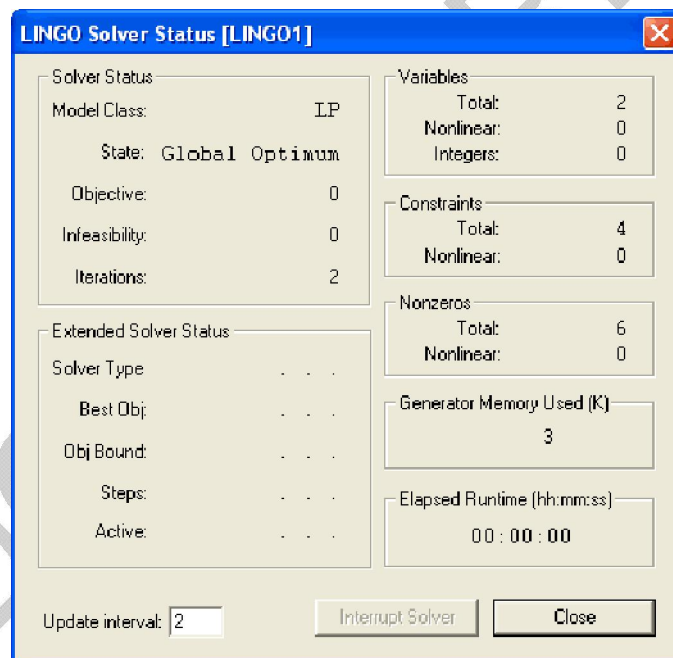
اکنون مدل شما نوشته شده و آماده حل شدن است. برای حل مدل، فرمان *Solve* را از منوی *LINGO* انتخاب کنید و یا دکمه *Solve* (🔍) را بر روی نوار ابزار بالای پنجره اصلی فشار دهید. لینگو شروع به کامپایل مدل خواهد کرد. در گذر این مرحله، لینگو بررسی خواهد کرد که همه ی عبارت ها مطابق قاعده باشند. اگر این گونه نباشد، یک پیغام خطا دریافت خواهید کرد. برای مثال، اگر در این مدل علامت ضرب را فراموش کرده باشید، پیغام خطایی به شکل زیر مشاهده خواهید کرد:



لینگو به شما اطلاع خواهد داد که یک خطای دستوری در مدل شما اتفاق افتاده؛ شماره‌ی خطی را که خطا در آن اتفاق افتاده اعلام کرده و محل وقوع خطا را هم نشان خواهد داد.

پنجره‌ی وضعیت راه‌یاب (حل کننده)^۱

اگر در مرحله‌ی کامپایل، خطاهای دستوری دیده نشوند؛ لینگو راه‌یاب مناسبی را فراخوان خواهد کرد تا جواب بهینه‌ی مدل را بیابد. در این حال، پنجره‌ی وضعیت راه‌یاب، مشابه زیر نشان داده خواهد شد:



- 1- Solver Status Window
- 2- Optimal solution

«پنجره‌ی وضعیت راه‌یاب»، پیشرفت کار و همچنین ابعاد مدل را نشان خواهد داد. فیلدهای مختلف این پنجره در ادامه تشریح خواهند شد. پنجره‌ی وضعیت راه‌یاب، دکمه‌ای با نام *Interrupt Solver* دارد که با فشار دادن آن، حل مساله متوقف خواهد شد^۱. در بیش‌تر موارد، لینگو بهترین جوابی را که تا آن‌جا یافته است گزارش خواهد کرد. یک استثنا، هنگامی است که از مدل‌های برنامه‌ریزی خطی استفاده می‌شود. اگر یک مدل برنامه‌ریزی خطی در هنگام اجرا متوقف شود، نتایج گزارش شده بی‌معنی بوده و باید از آن‌ها صرف‌نظر کرد. البته مشکلی ایجاد نخواهد شد؛ چون مدل‌های برنامه‌ریزی خطی به سرعت حل می‌شوند و نیازی به متوقف کردن آن‌ها نیست.

یادداشت: در صورت متوقف کردن راه‌یاب، باید در تفسیر جواب ارایه شده دقت نمود. این جواب: ۱) به طور قطع بهینه نیست، ۲) ممکن است برای همه‌ی محدودیت‌ها شدنی نباشد، و ۳) برای مدل‌های برنامه‌ریزی فطی بی‌ارزش خواهد بود.

در کنار دکمه‌ی *Interrupt Solver*، دکمه‌ی *Close* قرار دارد. فشار دادن این دکمه، پنجره‌ی وضعیت راه‌یاب را می‌بندد. می‌توانید این پنجره را هر زمان دیگری که بخواهید از مسیر *Window|Status Window command* باز کنید.

در پایین پنجره‌ی وضعیت راه‌یاب، فیلد *Update Interval* را خواهید دید. لینگو این پنجره را هر چند ثانیه که شما اعلام کنید به‌هنگام خواهد کرد. می‌توانید هر عددی را که تمایل دارید برای این فیلد در نظر بگیرید.

۱- با بهبود عملکرد رایانه‌های امروزی، مساله‌های معمولی در کسری از ثانیه حل می‌شوند. از این رو، نیازی برای توقف راه‌یاب دیده نمی‌شود.

متغیرها

جعبه‌ی متغیرها^۱، تعداد متغیرهای مدل را نشان می‌دهد. این جعبه هم‌چنین تعداد متغیرهای غیرخطی^۲ را هم نشان می‌دهد. یک متغیر، هنگامی غیرخطی نامیده می‌شود که حداقل در یکی از معادلات غیرخطی در هر یک محدودیت‌های مدل نقش داشته باشد. برای مثال، محدودیت:

$$X + Y = 100;$$

یک محدودیت خطی^۳ است؛ چرا که نمودار این تابع، یک خط راست خواهد بود. از سوی دیگر، تابع غیرخطی:

$$X * Y = 100;$$

درجه‌ی دوم بوده و نمودار آن یک منحنی خواهد بود. اگر می‌خواستیم مدلی شامل این محدودیت غیرخطی را حل کنیم، تعداد متغیرهای غیرخطی، حداقل ۲ تا بود تا نشان دهد که دو متغیر X و Y به صورت غیرخطی در این محدودیت نقش دارند.

به عنوان یک مثال دیگر، محدودیت زیر را در نظر بگیرید:

$$X * X + Y = 100;$$

در این مثال، X به صورت یک متغیر غیرخطی و Y به صورت یک متغیر خطی ظاهر خواهند شد. این محدودیت باعث نمی‌شود که Y به عنوان یک متغیر غیرخطی در نظر گرفته شود.

جعبه‌ی متغیرها در پنجره‌ی وضعیت راه‌یاب، تعداد متغیرهای عدد صحیح را هم نشان می‌دهد. به طور کلی، هر چه تعداد متغیرهای غیرخطی و عدد صحیح بیش‌تر باشد، حل مساله سخت‌تر بوده و زمان بیشتری صرف خواهد شد.

1- Variables box
2- Nonlinear
3- Linear

تعداد متغیرها، شامل متغیرهایی که لینگو آن‌ها را ثابت^۱ فرض می‌کند نخواهد شد. برای مثال، محدودیت‌های زیر را در نظر بگیرید:

$$X = 1;$$

$$X + Y = 3;$$

در محدودیت اول، لینگو X را برابر با مقدار ثابت ۱ در نظر می‌گیرد. با این اطلاعات، مقدار Y را هم برابر با مقدار ثابت ۲ تعیین خواهد کرد. از این رو، هیچ کدام از آن‌ها در تعداد متغیرها شمارش نخواهند شد.

محدودیت‌ها

جعبه‌ی محدودیت‌ها^۲، تعداد کل محدودیت‌ها و تعداد محدودیت‌های غیرخطی را نشان می‌دهد. یک محدودیت، هنگامی غیرخطی محسوب می‌شود که یک یا چند متغیر به شکل غیرخطی در آن حضور داشته باشند.

لینگو مدل شما را برای یافتن محدودیت‌های ثابت جستجو می‌کند. یک محدودیت، هنگامی ثابت محسوب می‌شود که همه‌ی متغیرهای آن ثابت باشند. محدودیت‌های ثابت، از شمار محدودیت‌های مدل خارج خواهند شد.

غیرصفرها

در هر محدودیت، زیرمجموعه‌ای از همه‌ی متغیرها دیده می‌شوند. متغیرهایی که در یک محدودیت دیده نمی‌شوند، به طور تلویحی ضریب صفر دارند؛ برعکس، متغیرهایی که در آن محدودیت حضور دارند ضریب غیرصفر دارند.

جعبه‌ی غیرصفرها^۳، تعداد همه‌ی ضریب‌های غیرصفر را و همچنین تعداد آنهایی را که در متغیرهای غیرخطی حضور دارند، جداگانه نشان می‌دهد.

1- Fixed

2- Constraints box

3- Nonzeros box

حافظه به کار رفته

جعبه‌ی حافظه به کار رفته^۱، مقدار حافظه‌ای را که لینگو به کار گرفته است نشان می‌دهد. می‌توانید سهمیه‌ی لینگو از مقدار حافظه را، در مسیر *LINGO|Options* *command* تعیین کنید.

زمان صرف شده

جعبه‌ی زمان صرف شده^۲، نشان می‌دهد که تاکنون لینگو چه زمانی برای حل مدل صرف کرده است. این مقدار می‌تواند متأثر از سایر برنامه‌های اجرا شده در سیستم شما هم باشد.

وضعیت راه‌یاب

جعبه‌ی وضعیت راه‌یاب^۳، وضعیت جاری راه‌یاب را نشان می‌دهد. در جدول زیر، فیلدهای موجود در این جعبه به همراه شرحی از مفهوم آن‌ها ارائه شده‌اند.

نام فیلد	مفهوم	حالت‌های ممکن
<i>Model Class</i>	نوع مدل	"LP", "QP", "CONE", "NLP", "MILP", "MIQP", "MICONE", "MINLP", "PILP", "PIQP", "PICONE", "PINLP"
<i>State</i>	وضعیت جواب جاری	"Global Optimum", "Local Optimum", "Feasible", "Infeasible", "Unbounded", "Interrupted", "Undetermined"
<i>Objective</i>	مقدار فعلی تابع هدف را نشان می‌دهد.	
<i>Infeasibility</i>	تعداد محدودیت‌هایی که نقض شده‌اند	
<i>Iterations</i>	تعداد تکرار در اجرای راه‌یاب	

1- Generator Memory Used box

2- Elapsed Runtime box

3- Solver Status box

نوع مدل^۱

نوع مدل، ویژگی مدل را به طور خلاصه بیان می‌کند. مدل‌های گوناگونی که ممکن است با آن‌ها مواجه شوید در جدول زیر فهرست شده‌اند. ترتیب معرفی آن‌ها، تقریباً از آسان به سخت مرتب شده است.

مخفف	عنوان	شرح
LP	Linear Program	همه‌ی عبارت‌ها خطی بوده و مدل شامل هیچ متغیر عدد صحیحی نیست.
QP	Quadratic Program	همه‌ی عبارت‌ها خطی یا درجه دوم هستند؛ مدل کوژ (محدب) است و محدودیت عدد صحیح ندارد.
CONE	Conic Program	مدل، برنامه‌ریزی مخروطی (مرتبه دوم) بوده و همه‌ی متغیرها پیوسته هستند.
NLP	Nonlinear Program	حداقل یکی از معادلات مدل غیرخطی است.
MILP	Mixed Integer Linear Program	همه‌ی عبارت‌ها خطی هستند. برخی از متغیرها عدد صحیح هستند.
MIQP	Mixed Integer Quadratic Program	همه‌ی عبارت‌ها خطی یا درجه دوم هستند؛ مدل کوژ است و برخی از متغیرها عدد صحیح هستند.
MICONE	Mixed Integer Conic Program	مدل، برنامه‌ریزی مخروطی (مرتبه دوم) است. برخی از متغیرها عدد صحیح هستند.
MINLP	Mixed Integer Nonlinear Program	حداقل یکی از معادلات مدل، غیرخطی بوده و

مخفف	عنوان	شرح
		برخی از متغیرها عدد صحیح هستند. به طور کلی، حل این نوع مدل‌ها بسیار سخت است، مگر در مورد مدل‌های کوچک.
PILP	Pure Integer Linear Program	همه‌ی عبارت‌ها خطی بوده و همه‌ی متغیرها عدد صحیح هستند.
PIQP	Pure Integer Quadratic Program	همه‌ی عبارت‌ها خطی یا درجه دوم هستند؛ مدل کوژ است، و همه‌ی متغیرها عدد صحیح هستند.
PICONE	Pure Integer Conic (Second-Order Cone) Program	مدل، برنامه‌ریزی مخروطی (مرتبه دوم) است. همه‌ی متغیرها عدد صحیح هستند.
PINLP	Pure Integer Nonlinear Program	حداقل یکی از معادلات مدل، غیرخطی بوده و همه‌ی متغیرها عدد صحیح هستند. به طور کلی، حل این نوع مدل‌ها بسیار سخت است، مگر در مورد مدل‌های کوچک.

وضعیت^۱

هنگامی که لینگو حل مدل را آغاز می‌کند، وضعیت اولیه‌ی جواب، “Undetermined” (نامعین) خواهد بود. چراکه راه‌یاب هنوز جوابی برای مدل نیافته است.

1- State

هنگامی که راه یاب تکرارها را آغاز می کند و وضعیت، “Infeasible” (ناشدنی) می شود، یعنی لینگو برخی جواب های آزمایشی به دست آورده است که هیچ کدام همه ی محدودیت ها را تامین نمی کنند.

اگر یک جواب «شدنی» وجود داشته باشد، راه یاب وضعیت “Feasible” را نشان خواهد داد. در این حالت، لینگو جوابی یافته است که همه ی محدودیت ها را تامین می کند ولی هنوز بهترین جواب برای مدل شما نیست.

هنگامی که راه یاب دیگر نتواند جواب بهتری برای مدل بیابد، کار خود را در وضعیت “Global Optimum” (بهینه ی کلی) یا “Local Optimum” (بهینه ی موضعی) پایان خواهد داد. اگر مدل، هیچ گونه محدودیت غیرخطی نداشته باشد، هر بهینه ی محلی یک بهینه ی کلی هم خواهد بود. از این رو، همه ی مدل های بهینه سازی خطی با حالت بهینه ی کلی پایان می پذیرند. از سوی دیگر، اگر مدل شما یک یا چند محدودیت غیرخطی داشته باشد، دیگر همه ی جواب های بهینه ی محلی، بهترین جواب مدل نخواهند بود. ممکن است راس^۱ دیگری بهتر از راس فعلی باشد؛ اما روش راه یاب قادر به دیدن آن نیست. بنابراین، در مدل های غیرخطی، لینگو تنها در حالت بهینه ی محلی پایان می پذیرد. ممکن است جواب یافته شده واقعا بهینه ی کلی هم باشد، ولی لینگو قادر به چنین ادعایی نیست. با در نظر گرفتن این حقیقت، بهتر است تا جایی که می توانید، مدل خود را تنها بر پایه ی محدودیت های خطی بنا کنید.

یادداشت: برای یافتن جواب های بهینه ی کلی (در مدل های غیرخطی) می توان (راه یاب کلی لینگو را نیز به کار گرفت. برای اطلاعات بیش تر در مورد راه یاب کلی، به «راه یاب غیرخطی» در راهنمای نرم افزار مراجعه کنید.

1- Peak

اگر مدل در وضعیت “Unbounded” (بی‌کران) پایان پذیرد، به این معنی است که لینگو می‌تواند تابع هدف را تا بی‌کران بهبود دهد. در عمل، این وضعیت هنگامی رخ می‌دهد که شما بتوانید سود بی‌کران ایجاد کنید. از آن‌جا که چنین وضعیتی نادر است - اگرچه ناممکن نیست - شاید برخی محدودیت‌ها را از قلم انداخته باشید یا به درستی تعیین نکرده باشید.

در پایان، وضعیت “Interrupted” (متوقف) هنگامی رخ خواهد داد که شما به صورت نابهنگام، راه‌یاب لینگو را قبل از آن‌که جواب پایانی را بیابد از ادامه کار بازدارید. روش ایجاد وقفه در کار راه‌یاب، پیش‌تر توضیح داده شد.

هدف^۱

در این‌جا مقدار تابع هدف برای جواب فعلی نشان داده می‌شود. اگر مدل شما تابع هدف نداشته باشد، عبارت N/A^۲ نشان داده می‌شود.

ناشدنی‌ها^۳

تعداد محدودیت‌هایی که در مدل نقض شده‌اند نشان داده می‌شوند. به یاد داشته باشید که تعداد همه‌ی نقض‌ها در محدوده‌ی متغیرها ردیابی نمی‌شوند. بنابراین، ممکن است مقدار این فیلد صفر باشد و هم‌زمان جواب فعلی هم ناشدنی باشد. برای این‌که بفهمید لینگو جواب شدنی پیدا کرده است یا خیر، باید به فیلد «وضعیت» که در بالا بررسی شد مراجعه نمایید.

1- Objective

2- Not Applicable, Not Available

3- Infeasibility

تکرارها^۱

عملیات بنیادی که توسط راهیاب لینگو انجام می شود «تکرار» نام دارد. فیلد «تکرارها» نشان می دهد که راهیاب لینگو تاکنون چند تکرار را کامل کرده است. به طور کلی، هر چه مدل بزرگ تر باشد نیازمند تکرارهای بیشتری خواهد بود و برای هر تکرار زمان بیشتری نیاز خواهد بود.

وضعیت راهیاب توسعه یافته^۲

جعبه ی «وضعیت راهیاب توسعه یافته»، اطلاعاتی در مورد چند راهیاب ویژه در لینگو نشان می دهد که عبارتند از:

- راهیاب انشعاب و تحدید^۳
- راهیاب کلی^۴
- راهیاب چندآغازی^۵

فیلدهای این جعبه تنها هنگامی نشان داده می شوند که یکی از این سه راهیاب در حال اجرا باشد. این فیلدها عبارتند از:

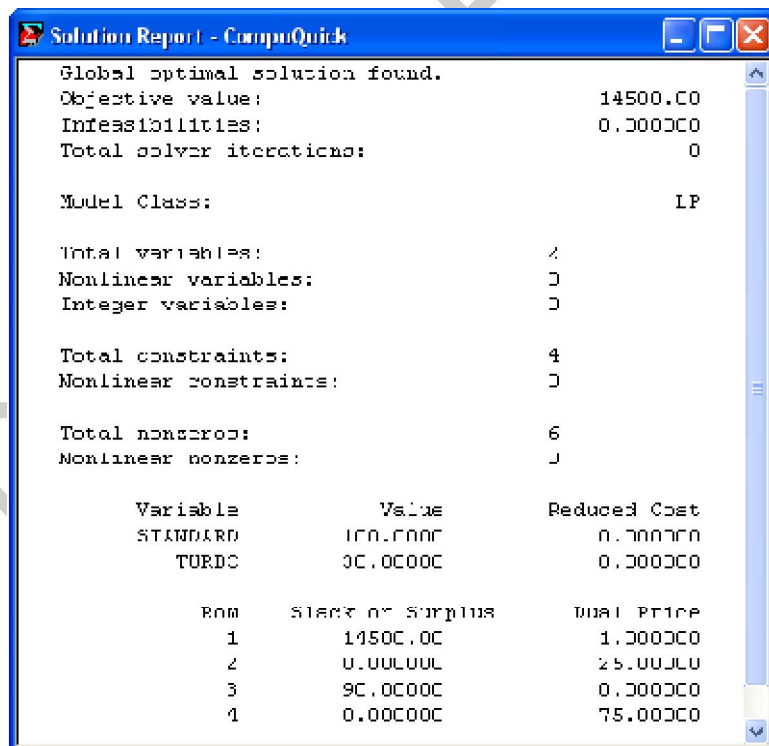
فیلد	شرح
<i>Solver Type</i>	نوع راهیاب مورد استفاده: "B-and-B", "Global", "Multistart"
<i>Best Obj</i>	مقدار تابع هدف بر اساس بهترین جوابی که تاکنون پیدا شده است.
<i>Obj Bound</i>	حد نظری هدف.

1- Iterations
 2- Extended Solver Status box
 3- Branch-and-Bound Solver
 4- Global Solver
 5- Multistart Solver

شرح	فیلد
تعداد گام‌های راه‌یاب.	Steps
تعداد زیرمساله‌های فعال که هنوز آنالیز نشده‌اند.	Active

گزارش جواب

هنگامی که لینگو مدل مربوط به مساله‌ی کامپیوکوییک را حل کرد، پنجره جدیدی با عنوان “Solution Report” (گزارش جواب) نمایش داده می‌شود که شامل جزییات جواب مدل خواهد بود. این گزارش مانند شکل زیر خواهد بود:



The screenshot shows a window titled "Solution Report - CompuQuick" with the following content:

```

Global optimal solution found.
Objective value:                14500.00
Infeasibilities:                0.000000
Total solver iterations:        0

Model Class:                    LP

Total variables:                2
Nonlinear variables:            0
Integer variables:              0

Total constraints:              4
Nonlinear constraints:          0

Total nonzeros:                6
Nonlinear nonzeros:            0

      Variable      Value      Reduced Cost
      STANDARD      100.0000      0.000000
      TURBO         90.0000      0.000000

      Row      Slack or Surplus      Dual Price
      1         14500.00              1.000000
      2         0.000000              25.000000
      3         90.0000              0.000000
      4         0.000000              75.000000
  
```

این جواب به ما می گوید که کامپیو کوئیک باید روزانه ۱۰۰ دستگاه رایانه‌ی استاندارد و ۳۰ دستگاه رایانه‌ی توربو بسازد تا سود روزانه‌ی برابر ۱۴۵۰۰ دلار به دست آورد. در باره‌ی فیلدهای مختلف این گزارش، در بخش‌های بعدی توضیح خواهیم داد.

چاپ

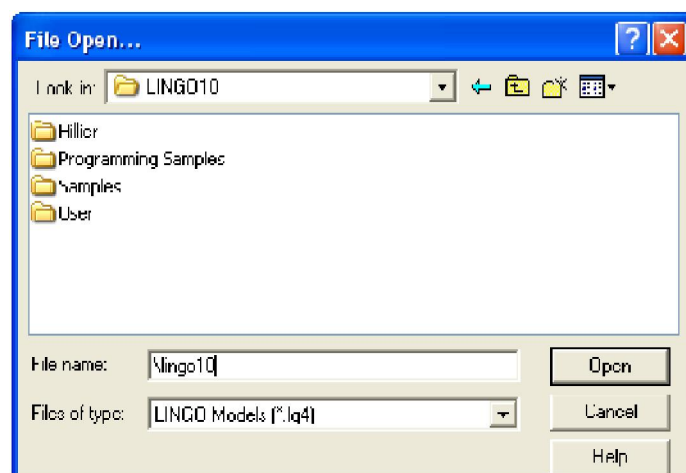
در ویرایش تحت ویندوز لینگو، اجرای فرمان *Print* در منوی *File* یا فشار دادن آیکون آن بر روی نوار ابزار (، پنجره‌ی فعال را چاپ می‌کند. می‌توانید هر پنجره‌ای (پنجره‌های مدل و گزارش) را چاپ کنید. اگر بخواهید تنها بخشی از یک پنجره را چاپ کنید، می‌توانید فرمان‌های *Cut* و *Paste* در منوی *Edit* را به کار گیرید؛ یا بر روی آیکون آن‌ها در نوار ابزار کلیک کنید. برای ایجاد یک پنجره‌ی جدید هم، مسیر *File|New* را دنبال کنید؛ یا بر روی آیکون آن در نوار ابزار کلیک کنید.

ذخیره‌ی کار

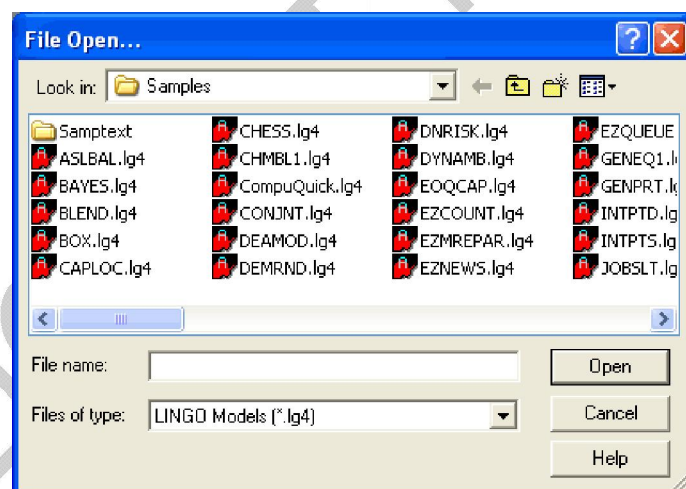
برای ذخیره‌ی یک مدل به صورت یک فایل، مسیر *File|Save* را دنبال کنید؛ یا بر روی آیکون آن در نوار ابزار کلیک کنید. لینگو به صورت خودکار پسوند *LG4* را برای نام فایل شما انتخاب خواهد کرد؛ مگر آن‌که شما روش دیگری را تعیین کنید.

بازکردن یک مدل نمونه

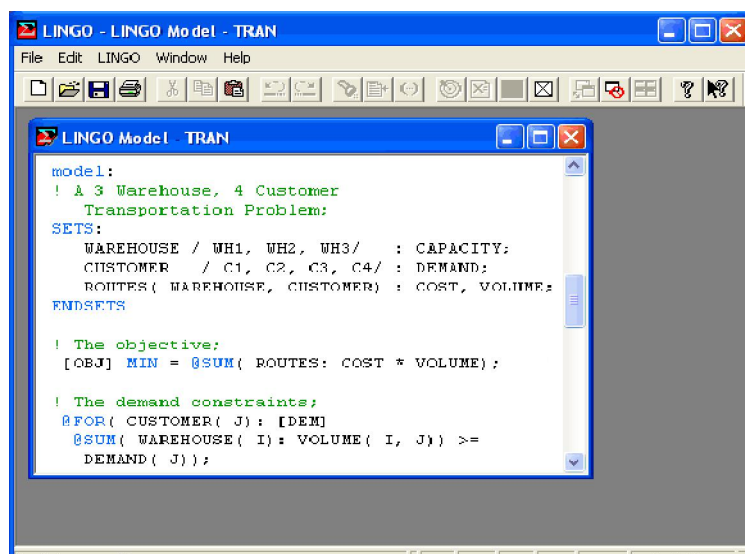
لینگو پوشه‌ای دارد که شامل برخی مدل‌های نمونه است. این مدل‌ها دامنه‌ی گسترده‌ای از کاربردها را پوشش می‌دهند. پوشه‌ی مدل‌های نمونه که *Samples* نام دارد، درست درون پوشه‌ی اصلی لینگو قرار دارد. برای باز کردن این پوشه، فرمان *Open* در منوی *File* را انتخاب کنید تا کادر زیر را ببینید:



پوشه‌ی *Samples* را انتخاب کنید:



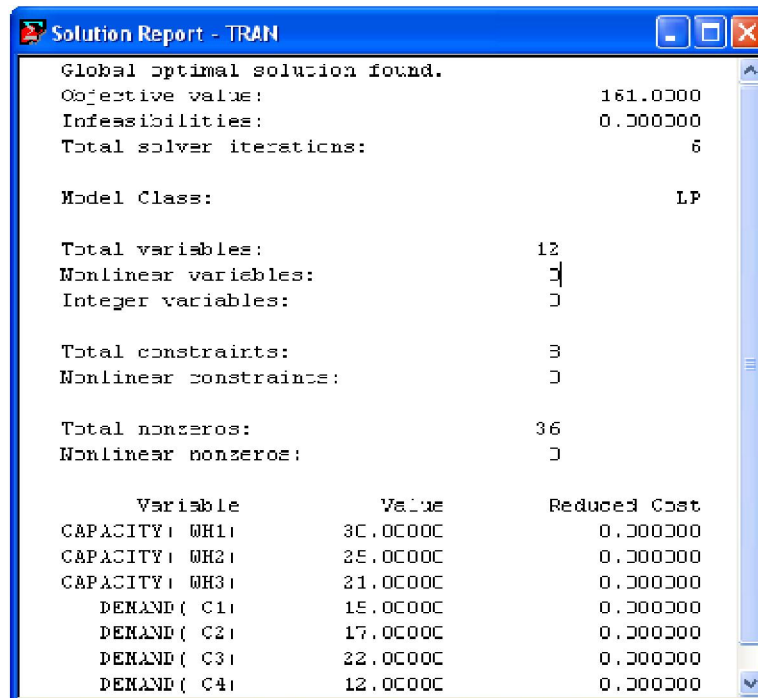
فایل *Tran* را که مربوط به یک مسالهی کوچک ترابری (حمل و نقل) است باز کنید. مدل را به شکل زیر خواهید دید:



اکنون می‌توانید همین مدل را حل کنید. برای حل مدل، فرمان *Solve* را از منوی *LINGO* انتخاب کنید، یا دکمه *Solve* (🔍) را بر روی نوار ابزار فشار دهید. مقدار بهینه‌ی تابع هدف در این مدل برابر ۱۶۱ است. پنجره‌ی وضعیت راه‌یاب را مانند زیر خواهید دید:



پشت پنجره‌ی وضعیت راه‌یاب، گزارش جواب را خواهید دید. این گزارش، خلاصه‌ای از اطلاعات مدل و همچنین مقدار همه‌ی متغیرها را در بر دارد:



Variable	Value	Reduced Cost
CAPACITY(WH1)	30.00000	0.000000
CAPACITY(WH2)	25.00000	0.000000
CAPACITY(WH3)	21.00000	0.000000
DEMAND(C1)	15.00000	0.000000
DEMAND(C2)	17.00000	0.000000
DEMAND(C3)	22.00000	0.000000
DEMAND(C4)	12.00000	0.000000

بررسی جواب

در گزارش جواب مربوط به مساله‌ی کامپیوکیک، نخست دیدیم که لینگو با صفر تکرار مساله را حل کرد (یعنی نیاز به تکرار نبوده است). دوم، بیش‌ترین سود ممکن برابر ۱۴۵۰۰ دلار خواهد بود. سوم، تعداد رایانه‌های استاندارد و توربو که باید ساخته شوند به ترتیب ۱۰۰ و ۳۰ دستگاه است. نکته‌ی جالب توجه آن که رایانه‌های توربو با وجود سود نسبی بیشتر، به تعداد کمتری تولید خواهد شد؛ چرا که این مدل نیاز به نیروی کار بیشتری دارد و ما در این زمینه محدودیت داریم. در ادامه، ستون‌های مربوط

به «هزینه‌ی کاهش یافته^۱»، «کسری یا فزونی^۲» و «بهای دوگان^۳» توضیح داده خواهند شد.

هزینه‌ی کاهش یافته

در گزارش جواب، هزینه‌ی کاهش یافته‌ی هر متغیر را خواهید دید. هزینه‌ی کاهش یافته‌ی یک متغیر را می‌توان این گونه تفسیر کرد که مقدار آن برابر جریمه‌ای است که شما باید بپردازید تا یک واحد از آن متغیر را در جواب وارد کنید. اگر هزینه‌ی کاهش یافته‌ی یک متغیر ۱۰ باشد، شما باید ۱۰ واحد بپردازید تا آن متغیر در جواب وارد شود. به دیگر سخن، مقدار تابع هدف به اندازه‌ی ۱۰ واحد در مسایل بیشینه‌سازی کاهش، و در مسایل کمینه‌سازی افزایش پیدا می‌کند.

هزینه‌های کاهش یافته، تنها در دامنه‌ی مقداری متغیرها معتبر هستند. برای اطلاعات بیش‌تر در مورد دامنه‌ی متغیرها، به فصل پنجم (فرمان‌های ویندوز) مراجعه کنید.

کسری یا فزونی

در گزارش جواب، «کسری یا فزونی» به شما می‌گوید که یک محدودیت چه اندازه به «برابری» نزدیک شده است. در محدودیت‌های «کوچک‌تر یا مساوی» ($\leq$)، این مقدار «کسری» نامیده می‌شود. در محدودیت‌های «بزرگ‌تر یا مساوی» ($\geq$)، این مقدار «فزونی» نامیده می‌شود.

1- Reduced Cost
2- Slack or Surplus
3- Dual Price

- اگر یک محدودیت به طور کامل برآورده شود، مقدار کسری یا فزونی آن برابر صفر خواهد شد.
 - اگر یک محدودیت نقض شده باشد برای مثال، در یک جواب ناشدنی - مقدار کسری یا فزونی منفی خواهد بود. دانستن این موضوع کمک می‌کند که در جواب‌های ناشدنی، محدودیت‌های نقض شده را پیدا کنید (مدلی که در آن نمی‌توان مجموعه‌ای از مقادیر را برای متغیرها پیدا کرد، به گونه‌ای که هم‌زمان همه‌ی محدودیت‌ها را برآورده سازند).
 - محدودیت‌های زاید^۱ (غیرفعال)، مقدار غیرصفر و مثبت در این ستون خواهند داشت.
- در مساله‌ی کامپیوکوییک، توجه داشته باشید که سطر سوم ($TURBO \leq 120$)، مقدار «کسری» به اندازه ۹۰ واحد دارد. از آن‌جا که مقدار بهینه‌ی توربو ۳۰ واحد است؛ این سطر، به اندازه ۹۰ واحد کسری دارد تا به طور کامل برآورده شود (نامساوی آن تبدیل به مساوی شود).

بهای دوگان

گزارش جواب، هم‌چنین بهای دوگان را برای هر محدودیت نشان می‌دهد. بهای دوگان را می‌توان این گونه تفسیر کرد که یک واحد افزایش در سمت راست یک محدودیت (یا مقدار ثابت آن)، چه بهبودی در تابع هدف ایجاد خواهد کرد. برای مثال، در مساله‌ی کامپیوکوییک، بهای دوگان سطر چهارم ۷۵ است. یعنی افزایش یک واحد نیروی کار، بهبود ۷۵ دلاری در تابع هدف را به دنبال خواهد داشت.

توجه داشته باشید که «بهبود» یک واژه‌ی نسبی است. در مسایل بیشینه‌سازی، بهبود به معنی افزایش در تابع هدف است. در حالی که در مسایل کمینه‌سازی، با افزایش سمت راست محدودیتی که بهای دوگان آن مثبت است، تابع هدف کاهش خواهد یافت.

بهای دوگان را گاهی اوقات «قیمت‌های سایه^۱» هم می‌نامند؛ چرا که به شما می‌گویند که برای به‌کارگیری یک واحد اضافی از منابع، چه اندازه باید پرداخت کنید. بر پایه‌ی این تحلیل، برای یک واحد نیروی کار بیش‌تر در کامپیوکیک، پرداخت تا ۷۵ دلار ارزشمند خواهد بود. قیمت‌های سایه نیز، تنها در دامنه مقداری متغیرها معتبر هستند.

کاربرد زبان مدل‌سازی

یکی از توانمندترین ویژگی‌های لینگو، زبان ریاضی مدل‌سازی آن است. زبان مدل‌سازی لینگو به شما اجازه می‌دهد که مساله‌ی خود را به شکل طبیعی آن بیان کنید که به نمادنویسی معمول ریاضی بسیار شبیه است. نه تنها می‌توانید هر بخش و هر محدودیت را به طور جداگانه وارد نمایید؛ این امکان هست که مجموعه‌ی کاملی از محدودیت‌های مشابه را در یک دستور فشرده بیان کنید. این کار باعث می‌شود که بتوانید مدل‌ها را به سادگی دنبال کنید و در صورت نیاز، آن‌ها را گسترش دهید.

یک ویژگی ارزشمند دیگر در زبان مدل‌سازی لینگو، «بخش داده‌ها^۲» است. بخش داده‌ها به شما اجازه می‌دهد که فرمول‌بندی را از داده‌های مدل جدا کنید. در حقیقت، لینگو حتی می‌تواند داده‌ها را از یک صفحه گسترده، یک بانک اطلاعاتی، یا از یک متن بخواند. هنگامی که داده از مدل جدا باشد، انجام تغییرات بسیار آسان‌تر بوده و احتمال اشتباه هم کم‌تر خواهد بود.

1- Shadow prices

2- Data section

مدل کامپیوکوییک که در بالا بررسی شد از متغیرهای اسکالر^۱ (جداگانه) استفاده می‌کند. هر متغیری، به گونه‌ای «تصریحی»^۲ نام‌گذاری شده (برای مثال استاندارد و توربو) و هر محدودیتی هم به طور تصریحی مشخص شده است (برای مثال *TURBO* ≤ 120). در مدل‌های بزرگ‌تر، نیاز خواهید داشت که با گروهی از متغیرها و محدودیت‌های بسیار مشابه کار کنید. با رویکرد مدل‌سازی جداگانه که تا این جا به کار گرفتیم، شما باید کار تکراری ورود هر بخش از هر محدودیت را دنبال می‌کردید. خوش‌بختانه، توانایی لینگو در به‌کارگیری مجموعه‌ها به شما اجازه می‌دهد که چنین عملیاتی را به گونه‌ای بسیار کارآتر انجام دهید.

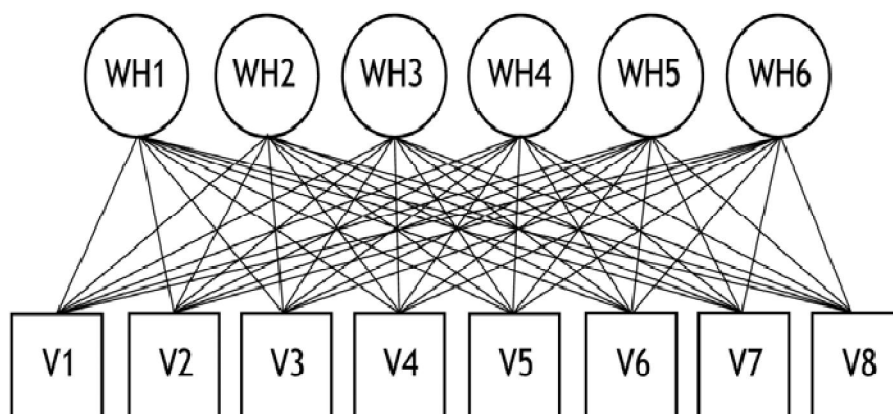
در ادامه خواهیم دید که چگونه می‌توان از مجموعه‌ها در حل یک مساله‌ی ترابری استفاده کرد. پس از بررسی این مثال، خواهید دید که توانایی مجموعه‌ها همراه با زبان مدل‌سازی لینگو، شما را قادر می‌کند که مدل‌های بزرگ را در زمان کمتری ایجاد کنید.

ساختن یک مدل ترابری بر پایه‌ی مجموعه‌ها

مساله

فرض کنید که شرکت «ابزارهای بی‌سیم» *WW*^۳، ۶ انبار برای پشتیبانی از ۸ فروشنده‌اش دارد. هر انبار گنجایشی دارد که نمی‌تواند افزایش یابد و هر فروشنده، درخواستی دارد که باید برآورده شود. *WW* می‌خواهد بداند «چه تعداد ابزار از کدام انبار برای کدام فروشنده بفرستد، به گونه‌ای که کل هزینه ترابری کمینه شود». این یک مساله‌ی کلاسیک بهینه‌سازی است که به مساله‌ی ترابری (حمل و نقل) مشهور است. نمودار زیر، مساله را نشان می‌دهد:

1- Scalar
2- Explicit
3- Wireless Widget



شبکه‌ی ترابری شرکت WW

از آنجا که هر انبار امکان ارسال برای هر فروشنده را دارد، ۴۸ مسیر ترابری محتمل است. پس برای هر مسیر یک متغیر نیاز داریم که مقدار ترابری همان مسیر را نشان خواهد داد. داده‌های زیر را داریم:

درخواست فروشندگان

فروشنده	درخواست
۱	۳۵
۲	۳۷
۳	۲۲
۴	۳۲
۵	۴۱
۶	۳۲
۷	۴۳
۸	۳۸

گنجایش انبارها

انبار	گنجایش
۱	۶۰
۲	۵۵
۳	۵۱
۴	۴۳
۵	۴۱
۶	۵۲

		فروشنده‌گان							
		V1	V2	V3	V4	V5	V6	V7	V8
انبار	Wh1	۶	۲	۶	۷	۴	۲	۵	۹
	Wh2	۴	۹	۵	۳	۸	۵	۸	۲
	Wh3	۵	۲	۱	۹	۷	۴	۳	۳
	Wh4	۷	۶	۷	۳	۹	۲	۷	۱
	Wh5	۲	۳	۹	۵	۷	۲	۶	۵
	Wh6	۵	۵	۲	۲	۸	۱	۴	۳

هزینه‌ی حمل هر واحد

تابع هدف

نخستین کار ما در فرموله کردن مدل، بنا کردن تابع هدف است. همان گونه که گفته شد، WW می‌خواهد کل هزینه‌ی حمل خود را کمینه کند. متغیری به نام $VOLUME_I_J$ را در نظر می‌گیریم که تعداد ابزار ارسال شده از انبار I به فروشنده J را بیان می‌کند. اکنون اگر می‌خواستیم تابع هدف خود را بر پایه‌ی متغیرهای اسکالر بنویسیم، به شکل زیر عمل می‌کردیم:

$$\begin{aligned}
 MIN = & 6 * VOLUME_1_1 + 2 * VOLUME_1_2 + \\
 & 6 * VOLUME_1_3 + 7 * VOLUME_1_4 + \\
 & 4 * VOLUME_1_5 + \\
 & \vdots \\
 & 8 * VOLUME_6_5 + 1 * VOLUME_6_6 + \\
 & 4 * VOLUME_6_7 + 3 * VOLUME_6_8;
 \end{aligned}$$

برای سادگی، تنها ۹ عبارت از ۴۸ عبارت لازم را در تابع هدف آورده ایم. همان گونه که دیده می شود، نوشتن چنین فرمول بلندی خسته کننده و همراه با اشتباه خواهد بود. در نظر بگیرید که یک مساله واقعی ممکن است هزاران فروشنده داشته باشد و فرموله کردن آن، اگر اشتباه هم نداشته باشد حتما سخت خواهد بود.

اگر با نمادسازی ریاضی آشنا باشید، می توانید این معادله ی بلند را به گونه ای بسیار فشرده تر بیان کنید:

$$\text{Minimize } \sum_{ij} \text{COST}_{ij} \bullet \text{VOLUME}_{ij}$$

لینگو هم به شیوه ای مشابه، اجازه می دهد که تابع هدف را به گونه ای وارد کنید که کوتاه بوده، به آسانی نوشته شده و به آسانی درک شود. دستور هم ارز تابع فوق در لینگو عبارت است از:

$$\text{MIN} = @SUM(LINKS(I,J) : \text{COST}(I,J) * \text{VOLUME}(I,J)) ;$$

این دستور می گوید که مجموع حاصل ضرب های هزینه هر واحد ابزار در تعداد آن ابزار با در نظر گرفتن همه ی پیوندها^۱ کمینه شود. جدول زیر نمادسازی لینگو را در مقایسه با نمادسازی ریاضی تابع هدف ما نشان می دهد:

نمادسازی ریاضی	نمادسازی لینگو ^۲
<i>Minimize</i>	<i>MIN =</i>
$\sum_{ij}$	<i>@SUM(LINKS(I,J) :</i>
COST_{ij}	<i>\text{COST}(I,J)</i>
$\bullet$	<i>*</i>
VOLUME_{ij}	<i>\text{VOLUME}(I,J)) ;</i>

1- Links

۲- ساختار دستورهای لینگو در فصل های بعدی تشریح خواهند شد.

محدودیت‌ها

پس از تعیین تابع هدف، گام بعدی فرموله کردن محدودیت‌ها است. دو گونه محدودیت در این مدل وجود دارند. گروه اول برای آن است که هر فروشنده‌ای تعداد ابزار مورد نیازش را دریافت کند که ما آن‌ها را محدودیت‌های درخواست می‌نامیم. گروه دوم که آن‌ها را محدودیت‌های گنجایش می‌نامیم، برای آن است که هر انباری تنها به اندازه‌ی موجودیش ارسال داشته باشد.

اگر محدودیت درخواست فروشنده‌ی یکم را در نظر بگیریم، باید تمام ارسال‌ها از همه‌ی انبارها به فروشنده‌ی یکم را جمع بسته و آن را محدود به درخواست فروشنده‌ی یکم (۳۵ واحد) بکنیم. بنابراین، اگر نمادنویسی اسکالر را دنبال می‌کردیم، نیاز داشتیم که محدودیت را به شکل زیر بنویسیم:

$$VOLUME_1_1 + VOLUME_2_1 + VOLUME_3_1 +$$

$$VOLUME_4_1 + VOLUME_5_1 + VOLUME_6_1 = 35;$$

سپس نیاز داشتیم که هفت محدودیت دیگر هم به شکل بالا برای هفت فروشنده‌ی دیگر بنویسیم. باز هم دیده می‌شود که این چنین نگارشی نه تنها خسته‌کننده بلکه همراه با اشتباه خواهد بود. ولی، همانند تابع هدف‌مان، می‌توانیم زبان مدل‌سازی لینگو را به کار گرفته و کارمان را آسان کنیم.

با نمادسازی ریاضی، همه‌ی هشت محدودیت را می‌توان در یک دستور فشرده کرد:

$$\sum_i VOLUME_{ij} = DEMAND_j, \text{ for all } j \text{ in VENDORS}$$

که دستور هم‌ارز آن در لینگو به صورت زیر خواهد بود:

```
@FOR (VENDORS (J) :
  @SUM (WAREHOUSES (I) : VOLUME (I, J) ) =
    DEMAND (J) ) ;
```

این دستور لینگو، جای‌گزین هشت محدودیت «درخواست» خواهد شد. به صورت متنی، این دستور می‌گوید که: «برای هر فروشنده‌ای^۱، جمع مقدار ارسالی^۲ از همه‌ی انبارها^۳ به آن فروشنده باید برابر درخواست^۴ همان فروشنده باشد».

در جدول زیر خواهید دید که این دستور، چه اندازه به نماد ریاضی خود شباهت دارد:

نمادسازی ریاضی	نمادسازی لینگو
<i>for all j in VENDORS</i>	@FOR (VENDORS (J) :
$\sum_i$	@SUM (WAREHOUSES (I) :
$VOLUME_{ij}$	VOLUME (I, J))
=	=
$DEMAND_j$	DEMAND (J)) ;

اکنون محدودیت‌های مربوط به گنجایش‌ها را تعریف می‌کنیم. در نمادسازی رایج ریاضی، شش محدودیت گنجایش به شکل زیر بیان خواهد شد:

$$\sum_j VOLUME_{ij} \leq CAP_i, \text{ for all } i \text{ in WAREHOUSES}$$

-
- 1- VENDORS
 - 2- VOLUME
 - 3- WAREHOUSES
 - 4- DEMAND

که دستور هم‌ارز آن در لینگو به صورت زیر خواهد بود:

```
@FOR (WAREHOUSES (I) :
  @SUM (VENDORS (J) : VOLUME (I, J)) <=
  CAPACITY (I)) ;
```

به صورت متنی، این دستور می‌گوید که: «برای هر انبار، جمع مقدار ارسالی به همه‌ی فروشندگان باید کوچک‌تر یا برابر گنجایش^۱ همان انبار باشد».

اکنون اگر همه‌ی کارهایی را که تا این جا انجام داده‌ایم در کنار هم بگذاریم؛ مدل کامل زیر را خواهیم داشت:

```
MODEL:
  MIN = @SUM (LINKS (I, J) :
    COST (I, J) * VOLUME (I, J)) ;
  @FOR (VENDORS (J) :
    @SUM (WAREHOUSES (I) : VOLUME (I, J)) =
    DEMAND (J)) ;
  @FOR (WAREHOUSES (I) :
    @SUM (VENDORS (J) : VOLUME (I, J)) <=
    CAPACITY (I)) ;
END
```

اکنون نیاز داریم که مجموعه‌های مدل (فروشندگان، انبارها و مسیرهای ترابری) را و هم‌چنین داده‌های مربوط به آن‌ها را تعریف کنیم. این کار را در دو بخش دیگر مدل که «بخش مجموعه‌ها»^۲ و «بخش داده‌ها»^۳ نامیده می‌شوند انجام خواهیم داد.

1- CAPACITY
2- Sets section
3- Data section

تعریف مجموعه‌ها

هرگاه بخواهید یک وضعیت واقعی را مدل‌سازی کنید، معمولاً خواهید دید که با مجموعه‌ای از اقلام مرتبط سروکار دارید. برای مثال، اقلامی مثل کارخانه‌ها، خریدارها، خودروها و کارکنان. معمولاً، هنگامی که یک محدودیت بر روی یک عضو مجموعه‌ای اعمال می‌شود، بر روی تمام اعضای آن مجموعه هم اعمال خواهد شد. این مفهوم ساده، در جوهره‌ی زبان مدل‌سازی لینگو قرار داده شده است. لینگو به شما اجازه می‌دهد که مجموعه‌ی اقلام مرتبط را در بخش مجموعه‌ها تعریف کنید. بخش مجموعه‌ها با کلیدواژه‌ی *SETS* (در یک خط جداگانه) آغاز شده و با کلیدواژه‌ی *ENDSETS* (در یک خط جداگانه) پایان می‌پذیرد. هنگامی که اعضای مجموعه‌ها تعریف شدند، می‌توان تابع‌های ویژه‌ی مجموعه‌ها (مثل *@FOR*) را به کار برد که تنها با یک عبارت، عملیاتی را بر روی همه‌ی اعضای مجموعه اعمال می‌کند.

در مساله شرکت *WW* سه مجموعه‌ی زیر را ایجاد کردیم:

- انبارها،
- فروشندگان، و
- مسیرهای ترابری از هر انبار تا هر فروشنده.

این سه مجموعه، در بخش مجموعه‌ها به شکل زیر تعریف می‌شوند:

SETS:

WAREHOUSES: CAPACITY;

VENDORS: DEMAND;

LINKS (WAREHOUSES, VENDORS): COST, VOLUME;

ENDSETS

خط دوم می‌گوید که مجموعه‌ی *WAREHOUSES* (انبارها)، مشخصه‌ای دارد که *CAPACITY* (گنجایش) نامیده می‌شود. خط بعدی هم برای «فروشنده‌گان» مشخصه‌ی «درخواست» را معرفی می‌کند.

مجموعه‌ی آخر که *LINKS* نامیده شده است، رابطه (پیوند) موجود در شبکه‌ی ترابری را بیان می‌کند. هر پیوند، یک مشخصه‌ی *COST* (هزینه) و یک مشخصه‌ی *VOLUME* (مقدار) دارد که با آن مرتبط است. نوشتار^۱ استفاده شده برای این مجموعه با دو مجموعه‌ی دیگر تفاوت دارد. با عبارت:

LINKS (WAREHOUSES, VENDORS)

به لینگو می‌گوییم که مجموعه‌ی *LINKS*، از دو مجموعه‌ی *WAREHOUSES* و *VENDORS* برگرفته شده است. در این صورت، لینگو زوج مرتب‌های (انبار، فروشنده) را ایجاد می‌کند. هر یک از این ۴۸ زوج مرتب، عضوی از مجموعه‌ی *LINKS* خواهند بود. برای روشن شدن مطلب، اعضای مجموعه‌ی *LINKS* را در جدول زیر فهرست کرده‌ایم:

شماره عضو	مسیر ترابری
۱	<i>WH1 → V1</i>
۲	<i>WH1 → V2</i>
۳	<i>WH1 → V3</i>
...	...
۴۷	<i>WH6 → V7</i>
۴۸	<i>WH6 → V8</i>

1- Syntax

یک ویژگی زیبا در لینگو، آن است که خودش به طور خودکار اعضای مجموعه‌ی *LINKS* را برپایه‌ی اعضای دو مجموعه‌ی *WAREHOUSES* و *VENDORS* تولید کرده و از این رو بخش زیادی از کار ما را انجام می‌دهد.^۱

ورود داده‌ها

لینگو به کاربر اجازه می‌دهد که داده‌ها را در «بخش داده‌های» مدل جدا کند. در مثال شرکت *WW*، بخش داده‌ها به صورت زیر خواهد بود:

```
DATA:
!set members;
WAREHOUSES = WH1 WH2 WH3 WH4 WH5 WH6;
VENDORS = V1 V2 V3 V4 V5 V6 V7 V8;

!attribute values;
CAPACITY = 60 55 51 43 41 52;
DEMAND = 35 37 22 32 41 32 43 38;
COST = 6 2 6 7 4 2 5 9
      4 9 5 3 8 5 8 2
      5 2 1 9 7 4 3 3
      7 6 7 3 9 2 7 1
      2 3 9 5 7 2 6 5
      5 5 2 2 8 1 4 3;
ENDDATA
```

بخش داده‌ها، با کلیدواژه‌ی *DATA:* در یک خط جداگانه آغاز می‌شود و با *ENDDATA* در یک خط جداگانه پایان می‌پذیرد.

۱- تصور کنید که می‌خواستیم چند صد مشتری را به چند ده فروشنده متصل کنیم.

در ادامه، فهرست انبارها و فروشندگان را وارد می‌کنیم. اگر بخواهیم، می‌توانیم از روش کوتاه شده‌ی زیر هم استفاده کنیم و همان نتیجه را بگیریم:

```
!set members;
WAREHOUSES = WH1..WH6;
VENDORS = V1..V8;
```

لینگو با دیدن دو نقطه پشت سرهم، می‌فهمد که باید شش انبار و هشت فروشنده را خودش ایجاد کند. سپس، مشخصه‌ی *CAPACITY* برای *WAREHOUSES* و مشخصه‌ی *DEMAND* برای *VENDORS* مقداردهی می‌شوند. مشخصه‌ی *COST* برای مجموعه‌ی دوبعدی *LINKS* نیاز به دقت بیشتری دارد. هنگامی که لینگو یک آرایه‌ی چندبعدی^۱ را در بخش داده‌ها مقداردهی می‌کند، نمایه‌ی بیرونی^۲ را تندتر افزایش می‌دهد. بنابراین، در این مثال، نخست *COST(WH1, V1)* و سپس *COST(WH1, V2)* تا *COST(WH1, V8)* مقداردهی می‌شوند. سپس مقدار بعدی به *COST(WH2, V1)* تا پایان تخصیص می‌یابد.

در این مثال، ما همه‌ی داده‌های مدل را در بخش داده‌ها جدا کرده‌ایم. با توجه به این‌که از یک اجرا به اجرای بعدی مدل، داده‌ها بیش از همه در معرض تغییر دارند؛ جداسازی داده‌ها اصلاحات بعدی را به صورت قابل ملاحظه‌ای آسان‌تر می‌کند. تصور کنید که پیدا کردن و اصلاح داده‌های پخش شده در محدودیت‌های مختلف یک مدل بزرگ چه اندازه سخت خواهد بود.

برای این‌که مدیریت داده‌ها باز هم ساده‌تر شود، لینگو توانایی ورود اطلاعات از منابع بیرونی را هم دارد. لینگو می‌تواند داده‌ها را از یک فایل متنی بخواند؛ ارتباط *OLE*^۳ زنده با *Excel* برقرار کند؛ و یا با یک پایگاه داده، ارتباط *ODBC*^۴ برقرار نماید.

1- Multidimensional array
2- Outer index
3- Object Linking and Embedding
4- Open Database Connectivity

با کنار هم قرار دادن بخش داده‌ها، بخش مجموعه‌ها، هدف و محدودیت‌ها، مدل کامل به شکل زیر خواهد بود:

```

MODEL:
! A 6 Warehouse 8 Vendor Transportation Problem;

SETS:
    WAREHOUSES: CAPACITY;
    VENDORS: DEMAND;
    LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS

! Here is the data;
DATA:
    !set members;
    WAREHOUSES = WH1 WH2 WH3 WH4 WH5 WH6;
    VENDORS = V1 V2 V3 V4 V5 V6 V7 V8;

    !attribute values;
    CAPACITY = 60 55 51 43 41 52;
    DEMAND = 35 37 22 32 41 32 43 38;
    COST = 6 2 6 7 4 2 5 9
           4 9 5 3 8 5 8 2
           5 2 1 9 7 4 3 3
           7 6 7 3 9 2 7 1
           2 3 9 5 7 2 6 5
           5 5 2 2 8 1 4 3;
ENDDATA

! The objective;
MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
! The demand constraints;
@FOR( VENDORS( J):
    @SUM( WAREHOUSES( I): VOLUME( I, J)) =
        DEMAND( J));
! The capacity constraints;
@FOR( WAREHOUSES( I):
    @SUM( VENDORS( J): VOLUME( I, J)) <=
        CAPACITY( I));
END

```

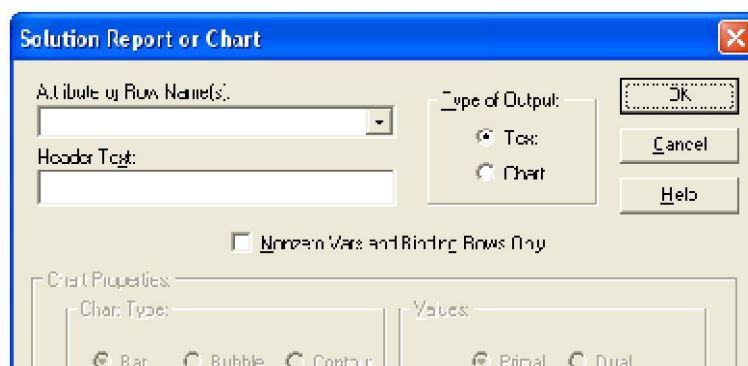
توجه داشته باشید که برای خوانایی بیشتر، برخی توضیحات را به مدل افزوده ایم. نام مدل *WIDGETS* است که می توانید آن را در پوشه‌ی نمونه فایل‌های همراه لینگو در مسیر نصب برنامه بیابید.

حل مدل ترابری

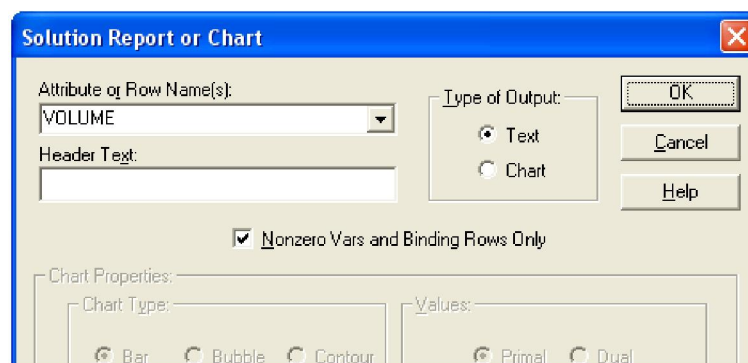
اکنون می‌خواهیم مساله را حل کنیم تا معلوم شود ارسالی بهینه در هر مسیر چه اندازه است. برای این کار، فرمان *Solve* را از منوی *LINGO* انتخاب کنید یا دکمه *Solve* (🔍) را بر روی نوار ابزار بالای پنجره‌ی اصلی فشار دهید. لینگو مساله را حل نموده و یک گزارش جواب بلندبالا ارائه خواهد کرد که شامل مقدار همه‌ی متغیرها، محدودیت‌ها و سایر داده‌های مدل خواهد بود. بیش‌تر این اطلاعات، در وهله‌ی اول ضروری نیستند. آنچه که اکنون می‌خواهیم بدانیم مقدار ارسالی از هر انبار به هر فروشنده است.

یادداشت: اگر فروبی لینگو برای تان بی‌فایده و اضافی به نظر می‌رسد؛ می‌توانید فرمان *Options...* (گزینه‌ها) را از منوی *LINGO* انتخاب کرده و در برگه‌ی *Interface*، سطح فروبی (Output Level) را *Terse* (مفصل و مفید) انتخاب کنید. اکنون، لینگو تنها مقادیر جواب، تابع هدف و تعداد تکرار را نشان خواهد داد.

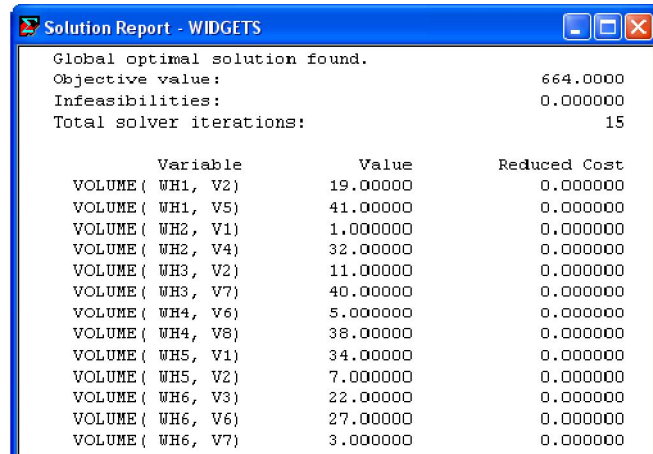
یادداشت: برای این که گزارش ما تنها مقادیر غیرصفر را نشان دهد، فرمان *Solution* را از منوی *LINGO* انتخاب می‌کنیم. کادر معاودهای زیر نشان داده خواهد شد.



در فیلد *Attribute or Row Name* مشخصه *VOLUME* را انتخاب کنید. برای آن که متغیرهایی با مقدار صفر نشان داده نشوند، چک باکس *Nonzero Vars and Binding Rows* را تیک بزنید. با انجام این کار، کادر محاوره‌ای را به شکل زیر خواهید دید:



اکنون دکمه‌ی تایید را بزنید تا گزارش زیر را که تنها مقادیر غیرصفر را نشان می‌دهد ببینید:



Global optimal solution found.

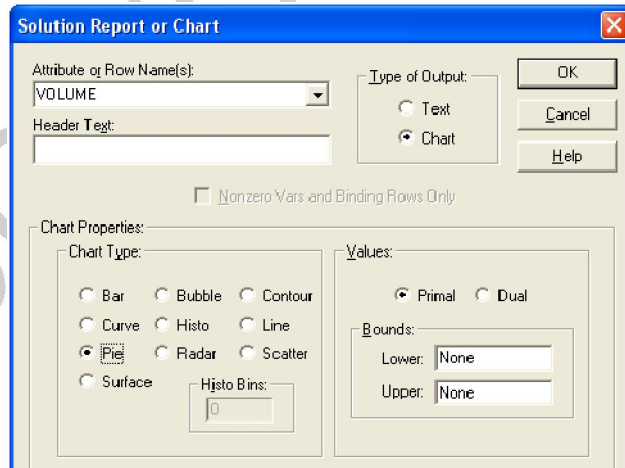
Objective value: 664.0000

Infeasibilities: 0.000000

Total solver iterations: 15

Variable	Value	Reduced Cost
VOLUME (WH1, V2)	19.00000	0.000000
VOLUME (WH1, V5)	41.00000	0.000000
VOLUME (WH2, V1)	1.000000	0.000000
VOLUME (WH2, V4)	32.00000	0.000000
VOLUME (WH3, V2)	11.00000	0.000000
VOLUME (WH3, V7)	40.00000	0.000000
VOLUME (WH4, V6)	5.000000	0.000000
VOLUME (WH4, V8)	38.00000	0.000000
VOLUME (WH5, V1)	34.00000	0.000000
VOLUME (WH5, V2)	7.000000	0.000000
VOLUME (WH6, V3)	22.00000	0.000000
VOLUME (WH6, V6)	27.00000	0.000000
VOLUME (WH6, V7)	3.000000	0.000000

همچنین لینگو می‌تواند نمودارهای گوناگونی بر پایه‌ی جواب شما بسازد. فرض کنید که می‌خواهید یک نمودار کلوچه‌ای^۱ برای مشخصه‌ی *VOLUME* نمایش دهید. برای این کار، نخست در جعبه‌ی *Type of Output*، گزینه‌ی *Chart* را برگزینید تا جعبه‌ی *Chart Properties* نمایان شده و بتوانید نمودار دلخواه خود را برگزینید:



Attribute or Row Name(s): VOLUME

Header Text:

Type of Output:

☐ Text

☒ Chart

☐ Nonzero Vars and Binding Rows Only

Chart Properties:

Chart Type:

☐ Bar ☐ Bubble ☐ Contour

☐ Curve ☐ Histo ☐ Line

☒ Pie ☐ Radar ☐ Scatter

☐ Surface

Histo Bins: 10

Values:

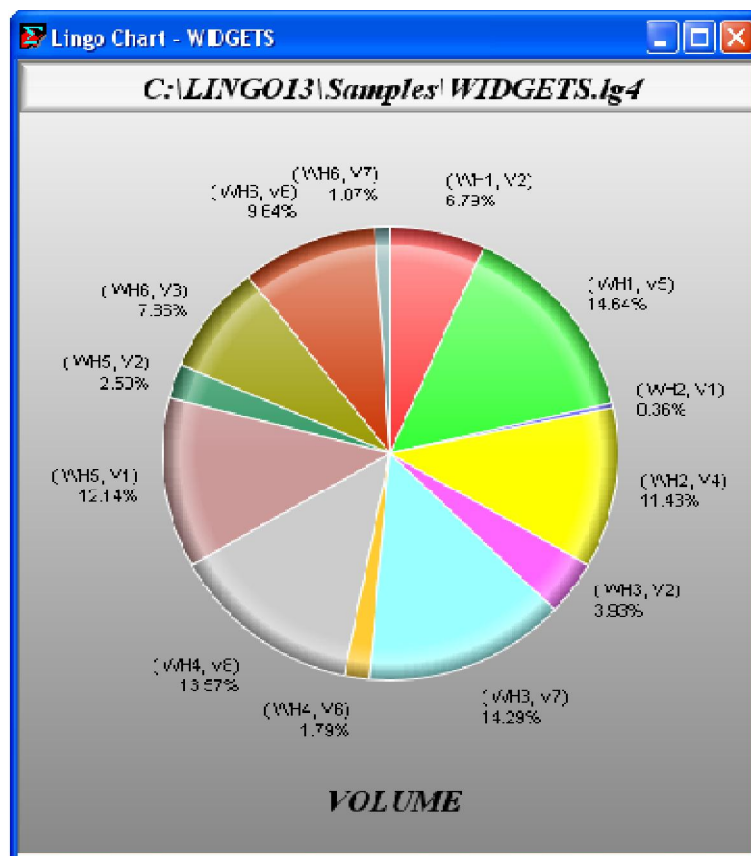
☒ Primal ☐ Dual

Bounds:

Lower: None

Upper: None

در پایان، پنجره‌ی دیگری باز شده و نمودار برگزیده‌ی شما در آن نمایش داده خواهد شد:



دیگر ویژگی‌های زبان مدل‌سازی

نام محدودیت‌ها

لینگو به شما اجازه می‌دهد که محدودیت‌ها را در مدل‌تان نام‌گذاری کنید. به دو دلیل، این کار خوبی است. نخست آن‌که نام محدودیت‌ها در گزارش جواب خواهد آمد و

تفسیر آن آسان‌تر خواهد شد. دوم، پیغام‌های اشتباه در صورت وجود - به نام محدودیت اشاره خواهند کرد. اگر محدودیت‌ها را نام‌گذاری نکنید، ردیابی سرچشمه‌ی اشتباه‌ها بسیار سخت خواهد بود.

یادداشت: لینگو شما را مجبور به نام‌گذاری محدودیت‌ها نمی‌کند. با این حال، اگر شما محدودیت‌ها را نام‌گذاری نکنید، لینگو خودش یک ترتیب دافلی برای محدودیت‌ها در نظر خواهد گرفت که ممکن است با ترتیب محدودیت‌های شما یکسان نباشد. از این رو، تفسیر گزارش جواب و ردیابی پیغام‌های خطا برای شما مشکل خواهد بود. بنابراین، پیشنهاد می‌شود که محدودیت‌های مدل را متما نام‌گذاری کنید.

نام‌گذاری یک محدودیت بسیار آسان است. همه‌ی آنچه که شما نیاز دارید، قرار دادن یک نام درون یک براکت (*[I]*) درست پیش از محدودیت است. نام باید از الزامات استاندارد لینگو پیروی نماید. نام باید با یک نویسه‌ی حرفی (A-Z) آغاز شود. نویسه‌های بعدی می‌توانند حرف، عدد (0-9) یا خط زیرین (*_*) باشند. نام می‌تواند تا ۶۴ نویسه طول داشته باشد. در زیر، چند مثال از نام‌گذاری محدودیت‌ها را می‌بینید:

```
[OBJECTIVE] MIN = X;
```

OBJECTIVE را به عنوان نام تابع هدف مدل در نظر می‌گیرد.

```
@FOR (LINKS (I, J) : [DEMAND_ROW]
@SUM (SOURCES (I) : SHIP (I, J) ) >=
DEMAND (J) );
```

DEMAND_ROW را به عنوان نام محدودیت درخواست در نظر می‌گیرد.

برای این که نام گذاری سطرها را بیش تر توضیح دهیم، مدل شرکت WW را دوباره بازنویسی کرده ایم. نام محدودیت ها به صورت توپر^۱ آورده شده اند:

```
MODEL:
! A 6 Warehouse 8 Vendor Transportation Problem;
SETS:
    WAREHOUSES: CAPACITY;
    VENDORS: DEMAND;
    LINKS( WAREHOUSES, VENDORS): COST, VOLUME;
ENDSETS
DATA:
    !set members;
    WAREHOUSES = WH1 WH2 WH3 WH4 WH5 WH6;
    VENDORS = V1 V2 V3 V4 V5 V6 V7 V8;
    !attribute values;
    CAPACITY = 60 55 51 43 41 52;
    DEMAND = 35 37 22 32 41 32 43 38;
    COST = 6 2 6 7 4 2 5 9
           4 9 5 3 8 5 8 2
           5 2 1 9 7 4 3 3
           7 6 7 3 9 2 7 1
           2 3 9 5 7 2 6 5
           5 5 2 2 8 1 4 3;
ENDDATA
! The objective;
[OBJECTIVE] MIN = @SUM( LINKS( I, J):
    COST( I, J) * VOLUME( I, J));
! The demand constraints;
@FOR( VENDORS( J): [DEMAND_ROW]
    @SUM( WAREHOUSES( I): VOLUME( I, J)) =
        DEMAND( J));
! The capacity constraints;
@FOR( WAREHOUSES( I): [CAPACITY_ROW]
    @SUM( VENDORS( J): VOLUME( I, J)) <=
        CAPACITY( I));
END
```

1- Bold

اکنون، خواندن و تفسیر گزارش جواب و سطرهای آن به شکل قابل ملاحظه‌ای آسان‌تر خواهد بود:

Row	Slack or Surplus	Dual Price
OBJECTIVE	664.0000	1.000000
DEMAND_ROW (V1)	0.000000	-4.000000
DEMAND_ROW (V2)	0.000000	-5.000000
DEMAND_ROW (V3)	0.000000	-4.000000
DEMAND_ROW (V4)	0.000000	-3.000000
DEMAND_ROW (V5)	0.000000	-7.000000
DEMAND_ROW (V6)	0.000000	-3.000000
DEMAND_ROW (V7)	0.000000	-6.000000
DEMAND_ROW (V8)	0.000000	-2.000000
CAPACITY_ROW (WH1)	0.000000	3.000000
CAPACITY_ROW (WH2)	22.0000	0.000000
CAPACITY_ROW (WH3)	0.000000	3.000000
CAPACITY_ROW (WH4)	0.000000	1.000000
CAPACITY_ROW (WH5)	0.000000	2.000000
CAPACITY_ROW (WH6)	0.000000	2.000000

توجه داشته باشید که اکنون هر سطر به جای این که یک نمایه‌ی ساده داشته باشد یک نام دارد.

نام مدل

می‌توانید برای مدل‌تان یک نام در نظر بگیرید و آن را در هر جایی که بخواهید قرار دهید. اگر برای مدل‌تان نامی در نظر بگیرید، آن نام در بالای گزارش جواب چاپ خواهد شد. همچنین، این نام شناسه‌ی پیش‌فرض در تابع‌های **ODBC** خواهد بود.

نام مدل باید با کلیدواژه‌ی **TITLE** آغاز شده و با ویرگول نقطه (:) پایان پذیرد. همه‌ی متن موجود در میان **TITLE** و ویرگول نقطه، «نام» در نظر گرفته می‌شود.

در زیر، یک نام به مدل *WW* افزوده ایم:

```
MODEL:
TITLE Widgets;
! A 6 Warehouse 8 Vendor Transportation Problem;
SETS:
WAREHOUSES: CAPACITY;
.
```

توجه داشته باشید که اکنون، نام مدل در بالای گزارش جواب چاپ خواهد شد:

```
Model Title: Widgets
Variable      Value      Reduced Cost
CAPACITY( WH1)  60.00000    0.00000000
CAPACITY( WH2)  55.00000    0.00000000
CAPACITY( WH3)  51.00000    0.00000000
CAPACITY( WH4)  43.00000    0.00000000
.
```

نشان دادن کوژی^۱ و کاوی^۲

راهیاب لینگو می تواند ویژگی کوژی و کاوی یک محدودیت را در بهبود مدل به کار گیرد. در بسیاری از موارد، لینگو خودش می تواند به طور خودکار کوژ بودن یا کاو بودن یک محدودیت را تشخیص دهد؛ در دیگر موارد و اگر خودتان بدانید که

1- Convexity تحدب
2- Concavity تقعر

محدودیت کوژ است یا کاو، می‌توانید آن را نشان کنید. برای این کار، به جای نشان‌های عادی برابری یا نابرابری ($=$ ، $<$ و $>$) نشان‌های $=C$ ، $<C$ و $>C$ را به کار بگیرید:

عمل‌گر	تفسیر
$<C=$	محدودیت $F(X) \leq b$ کوژ است.
$>C=$	محدودیت $F(X) \geq b$ کاو است.
$=C=$	تابع $F(X)$ در محدودیت $F(X) = b$ یک تابع کوژ است.

برای مثال، مدل غیرخطی زیر را در نظر بگیرید:

MODEL:

```

MAX= 5 * W1 - 3 * X1 - 4 * Y1
      + 5 * W2 - 3 * X2 - 4 * Y2
      + 5 * W3 - 3 * X3 - 4 * Y3;

- W1 <= 1;
W1 <= 1;
X1 + Y1 >= 1;
[NLROW1] W1 ^ 3 - X1 * Y1 - 0 <C= 0;

- W2 <= 1;
W2 <= 1;
X2 + Y2 >= 1;
[NLROW2] W2 ^ 3 - X2 * Y2 - 0 <C= 0;

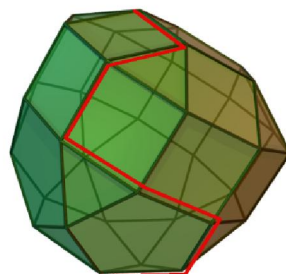
- W3 <= 1;
W3 <= 1;
X3 + Y3 >= 1;
[NLROW3] W3 ^ 3 - X3 * Y3 - 0 <C= 0;

```

END

در این مدل، سه سطر غیرخطی $NLROW1$ ، $NLROW2$ و $NLROW3$ کوژ هستند و ما این را با عمل گر $=C<$ نشان داده ایم. از این رو، برنامه ی راه یاب بی درنگ کوژی محدودیت را خواهد دانست. اگر به جای این عمل گر، عمل گر رایج $=>$ را به کار می گرفتیم، حل مدل به شکل قابل ملاحظه ای طولانی تر می شد.

فصل دوم کاربرد مجموعه‌ها



همان گونه که در بخش پیشین گفته شد، هنگامی که یک مساله‌ی واقعی را مدل می‌کنید، یک یا چند گروه از ارقام مرتبط را خواهید داشت. برای مثال: کارخانه‌ها، خریدارها، خودروها یا کارکنان. لینگو به شما اجازه می‌دهد که این ارقام را در مجموعه‌ها دسته‌بندی کنید. در این صورت، می‌توانید از تابع‌های مجموعه استفاده کرده و همه‌ی توانمندی‌های زبان مدل‌سازی لینگو را به کار گیرید.

در فصل یکم (آغاز کار با لینگو)، دیباچه‌ی کوتاهی از کاربرد مجموعه‌ها گفته شد. اکنون می‌خواهیم به ژرفای بیشتری برسیم و ببینیم چگونه می‌توان مجموعه‌ها را بنا کرده و به مشخصه‌های آن‌ها مقداردهی کرد. سپس می‌توانیم نمونه‌های جالب و سودمندی بسازیم.

چرا مجموعه‌ها؟

مجموعه^۱ها، بنیان زبان مدل‌سازی لینگو هستند؛ سنگ بنای نیرومندترین توانایی‌های این برنامه. با درک مجموعه‌ها، می‌توانید گروهی از محدودیت‌های مشابه را در یک عبارت تنها بنویسید و فرمول‌های بلند و پیچیده را به گونه‌ای فشرده بیان کنید. بنابراین، می‌توانید بزرگ‌ترین مدل‌ها را به تندی و آسانی بیان کنید. در مدل‌های بزرگ‌تر، نیاز پیدا خواهید کرد که گروهی از محدودیت‌ها و محاسبات هم‌سان را بیان کنید. خوش‌بختانه، توانایی لینگو در اداره کردن مجموعه‌ی اطلاعات، شما را قادر می‌کند چنین عملیاتی را به گونه‌ای کارآمد به جا آورید.

برای مثال، اگر مجبور باشید هر محدودیتی را جداگانه بنویسید، ساختن مدل ترابری برای ۱۰۰ انبار خسته‌کننده خواهد بود (برای مثال: انبار ۱ نباید بیش از موجودیش ارسال کند، انبار ۲ نباید بیش از موجودیش ارسال کند، انبار ۳ نباید بیش از موجودیش ارسال کند، و تا آخر). لینگو شما را قادر می‌کند تا فرمول را به گونه‌ای بنویسید که هم خواندن آن و هم درک آن آسان باشد (برای مثال: هیچ انباری نباید بیش از موجودیش ارسال کند)^۲.

مجموعه‌ها چیستند؟

به طور ساده، مجموعه‌ها گروه‌هایی از اقلام مرتبط هستند. یک مجموعه می‌تواند فهرستی از فرآورده‌ها، خودروها یا کارکنان باشد. اعضای مجموعه می‌توانند یک یا چند ویژگی داشته باشند. هر یک از این ویژگی‌ها را یک «مشخصه»^۳ می‌نامیم. مقادیر

۱- در زبان پارسی، واژه‌ی هم‌ارز مجموعه را «دسته» و «گردایه» گفته‌اند.

۲- تنها یک فرمول به جای ۱۰۰ فرمول نوشته خواهد شد.

مشخصه‌ها می‌توانند از پیش معلوم باشند یا این‌که لینگو آن‌ها را به دست آورد. برای مثال، هر محصول از مجموعه‌ی محصولات، می‌تواند یک مشخصه‌ی «قیمت» داشته باشد؛ هر خودرو از مجموعه‌ی خودروها، می‌تواند یک مشخصه‌ی «گنجایش» داشته باشد؛ هر کارمند از مجموعه‌ی کارکنان، می‌تواند یک مشخصه‌ی «دستمزد» داشته باشد، و / یا یک مشخصه‌ی دیگر هم برای «تاریخ تولد» داشته باشد.

انواع مجموعه‌ها

لینگو دو نوع مجموعه را می‌شناسد: مستقل^۱ و وابسته^۲.

اعضای مجموعه‌ی مستقل، تنها از یک گروه اقلام ساخته شده‌اند و نمی‌توان آن‌ها را کاهش داد. در مثال شرکت WW، مجموعه‌ی WAREHOUSES که از شش انبار تشکیل شده است، یک مجموعه‌ی مستقل است. به همین ترتیب، شش فروشنده هم یک مجموعه‌ی مستقل تشکیل می‌دهند.

مجموعه‌ی وابسته، مجموعه‌ای است که با استفاده از یک یا چند مجموعه‌ی دیگر ساخته می‌شود. به دیگر سخن، مجموعه‌ی وابسته اعضایش را از دیگر مجموعه‌ها می‌گیرد^۳. باز هم در مثال شرکت WW، مجموعه‌ی LINKS که از پیوند شش انبار و هشت فروشنده ساخته می‌شود، یک مجموعه‌ی وابسته است. اعضای این مجموعه، زوج مرتب‌هایی هستند که از اعضای دو مجموعه‌ی WAREHOUSES و VENDORS گرفته شده‌اند. مجموعه‌ی وابسته را می‌توان از دیگر مجموعه‌های وابسته نیز تشکیل داد. در ادامه چگونگی تعریف مجموعه‌ی وابسته را خواهیم دید.

1- Primitive مجموعه‌ی اصلی یا اولیه
2- Derived مجموعه‌ی مشتق یا وارث

۳- مجموعه‌ی مستقل، یک آرایه‌ی یک‌بعدی است. مجموعه‌ی وابسته، یک آرایه‌ی چندبعدی است.

LINGO

User's Guide



فایل پی دی اف فصل نخست این کتاب، برای بهره‌برداری دانشجویان و علاقمندان به طور رایگان منتشر شده است. علاقمندان به تهیه‌ی این کتاب (در ۲۵۷ صفحه و با قیمت ۱۲۵,۰۰۰ ریال) می‌توانند به انتشارات آراد (۶۶۴۸۲۲۶۶) مراجعه نمایند.

در صورت فرید بیش از ۱۰ جلد (با ۲۰٪ تخفیف) و همچنین در صورت نیاز به آموزش و مشاوره، با ای میل زیر مکاتبه فرمایید.

مؤلف

v_ajorloo@yahoo.com

کلیه حقوق این کتاب متعلق به مؤلف است.

ارسال این فایل به صورت الکترونیکی بلامانع بوده ولی چاپ، تکثیر یا بازنشر آن (بر روی کاغذ یا وبلاگ‌ها و وبسایت‌ها) مجاز نیست.