คู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์

นายสมหวัง ปิ่นปกรณ์

การศึกษาค้นคว้าอิสระนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาบริหารธุรกิจมหาบัณฑิต
แขนงวิชาบริหารธุรกิจ สาขาวิชาวิทยาการจัดการ มหาวิทยาลัยสุโขทัยธรรมาธิราช
พ.ศ. 2554

The Manual for using LINGO Program to Solving Mathematical Optimization Model

Mr. Somvang Pinpakorn

An Independent Study Submitted in Partial Fulfillment of the Requirements for

the Degree of Master of Business Administration

School of Management Science

Sukhothai Thammathirat Open University

2011

หัวข้อการศึกษาค้นคว้าอิสระ	คู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์
ชื่อและนามสกุล	นายสมหวัง ปิ่นปกรณ์
แขนงวิชา	บริหารธุรกิจ
สาขาวิชา	วิทยาการจัดการ มหาวิทยาลัยสุโขทัยธรรมาธิราช
อาจารย์ที่ปรึกษา	รองศาสตราจารย์จีราภรณ์ สุธัมมสภา

การศึกษาค้นคว้าอิสระนี้ ได้รับความเห็นชอบให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรระดับปริญญาโท เมื่อวันที่ 13 ธันวาคม 2554

คณะกรรมการสอบการศึกษาค้นคว้าอิสระ

..... ประธานกรรมการ
(รองศาสตราจารย์จีราภรณ์ สุธัมมสภา)

..... กรรมการ
(รองศาสตราจารย์ประภาศรี พงศ์ธนาพาณิช)

.....
(รองศาสตราจารย์อัจฉรา ชีวะตระกูลกิจ)
ประธานกรรมการประจำสาขาวิชาวิทยาการจัดการ

ชื่อการศึกษาค้นคว้าอิสระ คู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์

ผู้ศึกษา นายสมหวัง ปิ่นปกรณ์ **รหัสนักศึกษา** 2523000509 **ปริญญา** บริหารธุรกิจมหาบัณฑิต **อาจารย์ที่ปรึกษา** รองศาสตราจารย์จีราภรณ์ สุธัมมสภา **ปีการศึกษา** 2554

บทคัดย่อ

รูปแบบการค้นคว้าอิสระประเภทหนังสือคู่มือการใช้โปรแกรมฉบับนี้มีวัตถุประสงค์เพื่อ (1) จัดทำคู่มือของโปรแกรม LINGO และรายละเอียดในส่วนต่างๆของการใช้โปรแกรมเพื่อนำไปใช้งาน (2) ประเมินผลคู่มือการใช้โปรแกรม LINGO

วิธีการศึกษา เป็นการศึกษาการทำงานของโปรแกรมจากคู่มือการใช้งานของโปรแกรม LINGO Version 12.0 ซึ่งเป็นภาษาต่างประเทศ โดยให้พนักงานของแผนกวิศวกรรมของบริษัท วาย เอส ภัณฑ์ เป็นกลุ่มตัวอย่างในการประเมินผลคู่มือเมื่อผู้ทำการศึกษาได้ศึกษาจนเข้าใจเนื้อหาแล้ว จึงนำมาสรุปและอธิบายรายละเอียดต่างๆเป็นเนื้อหาของคู่มือภาษาไทยซึ่งประกอบด้วยส่วนต่างๆที่สำคัญของการใช้งานโปรแกรม และตัวอย่างทางธุรกิจที่ใช้ตัวแบบทางคณิตศาสตร์ในการแก้ปัญหาและการหาคำตอบด้วยโปรแกรม LINGO พร้อมทั้งทดลองใช้กับงานจริง 1 ตัวอย่าง

ผลการศึกษา (1) ได้คู่มือการใช้งานของโปรแกรม LINGO เป็นภาษาไทยประกอบไปด้วยเนื้อหาในคู่มือที่ได้แบ่งไว้เป็นตัวอย่างทางธุรกิจและตัวอย่างที่ทดลองใช้จริงโดยในเนื้อหาของตัวอย่างทางธุรกิจประกอบไปด้วยตัวแบบปัญหาเรื่องส่วนประสมของผลิตภัณฑ์ การขนส่ง การวางแผนการทำงานของพนักงาน พิกัดน้ำหนักส่วนผสมเมล็ดพืช การกำหนดกำลังคนโดยแสดงออกมาเป็นรายการ จำนวนเต็มเลขฐานสอง,จำนวนเต็มเลขฐานสองกรณีส่วนผสมผลิตภัณฑ์และตัวแปรอิสระของการพยากรณ์ และได้นำไปทดลองประยุกต์ใช้กับงานของแผนกวิศวกรรมบริษัท วาย เอส ภัณฑ์ ในเรื่องส่วนประสมของผลิตภัณฑ์ (2) ผู้ประเมินคู่มือทั้งหมดเป็นเพศชายการศึกษาระดับปริญญาตรีมีอายุอยู่ในช่วง 30-35 ปีเป็นส่วนใหญ่โดยผลการประเมินคู่มือพบว่าคู่มือมีความน่าสนใจอยู่ในเกณฑ์ที่ดีแต่เนื่องจากในเนื้อหาเป็นเรื่องของตัวแบบทางคณิตศาสตร์ดังนั้นการที่จะใช้คู่มือให้ได้ผลจำเป็นต้องมีพื้นฐานความรู้ทางด้านการวิเคราะห์เชิงปริมาณมาก่อนและผู้ประเมินคู่มือส่วนใหญ่เห็นว่าการใช้ตัวแบบทางคณิตศาสตร์และโปรแกรม LINGO มีความน่าสนใจต่อการนำไปใช้งานจริง

คำสำคัญ คู่มือ โปรแกรมLINGO โมเดลทางคณิตศาสตร์

กิตติกรรมประกาศ

การดำเนินการจัดทำการค้นคว้าอิสระฉบับนี้สำเร็จลุล่วงลงได้เนื่องจากได้รับความกรุณาเป็นอย่างยิ่งจากรองศาสตราจารย์จีราภรณ์ สุธัมมสภา สาขาวิทยาการจัดการ แห่งมหาวิทยาลัยสุโขทัยธรรมาธิราช ที่กรุณาได้ให้คำแนะนำ คำปรึกษา ตรวจแก้ไข และติดตามการดำเนินการจัดทำการค้นคว้าอิสระอย่างใกล้ชิดเพื่อปรับปรุงข้อบกพร่องต่างๆของงานค้นคว้าอิสระเรื่องนี้ด้วยดีตลอดมา ตั้งแต่เริ่มดำเนินการจนสำเร็จลุล่วงลงได้อย่างเรียบร้อยและสมบูรณ์ ผู้ดำเนินการศึกษารู้สึกซาบซึ้งในความกรุณาของท่านเป็นอย่างยิ่ง

นอกจากนี้ ผู้ศึกษาขอขอบพระคุณคณาจารย์สาขาวิทยการจัดการ เพื่อนนักศึกษา และผู้ที่มีส่วนเกี่ยวข้องในการช่วยเหลือพร้อมทั้งให้กำลังใจเพื่อการค้นคว้าอิสระนี้ได้สำเร็จลุล่วงลง คุณประโยชน์สิ่งใดที่จะได้รับจากการศึกษาการค้นคว้าอิสระฉบับนี้ผู้ศึกษาขอมอบแด่ บิดามารดา ครูอาจารย์ ที่ได้ให้ความอุปการะอบรมสั่งสอน แลประสิทธิ์ประสาทวิชาความรู้ให้แก่ผู้ทำการศึกษา รวมทั้งสมาชิกทุกคนในครอบครัวที่ได้ให้ความเกื้อกูลต่อผู้ทำการศึกษามาโดยเสมอ

สมหวัง ปิ่นปกรณ์

ตุลาคม 2554

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
กิตติกรรมประกาศ ..... จ
สารบัญตาราง ..... ซ
สารบัญภาพ ..... ฌ
บทที่ 1 บทนำ 1
ความเป็นมาและความสำคัญของปัญหา 1
วัตถุประสงค์การศึกษา 2
ประโยชน์ที่คาดว่าจะได้รับ 2
ประเด็นปัญหาที่ทำการศึกษา 3
ขอบเขตของการศึกษาคู่มือ 3
นิยามศัพท์ 3
บทที่ 2 วรรณกรรมที่เกี่ยวข้อง 4
แนวคิดทางทฤษฎี 5
ข้อกำหนดแบบเชิงเส้น (Linear Constraints) 5
ข้อกำหนดที่ไม่เป็นเชิงเส้น (Nonlinear Constraints) 6
คำสั่งของโปรแกรมที่ใช้ใน WINDOW 12
คำสั่งที่ใช้ใน Window แบบเชิงลึก 14
บทที่ 3 วิธีดำเนินการวิจัย 28
ประชากรและกลุ่มตัวอย่าง 28
เครื่องมือที่ใช้ในการวิจัย 29
การเก็บรวบรวมข้อมูล 29
บทที่ 4 ผลการศึกษา 30
พื้นฐานทางทฤษฎีที่เกี่ยวข้องกับตัวแบบทางคณิตศาสตร์ 30
คำสั่งของโปรแกรม 31
ขั้นตอนการสร้างตัวแบบของปัญหา 31
เนื้อหาที่ได้จากการศึกษาข้อมูล 32
การเริ่มต้นใช้โปรแกรม LINGO 32

สารบัญ (ต่อ)

หน้า

การพัฒนาและสร้างตัวแบบของโปรแกรม LINGO ในวินโดวส์ (WINDOWS)
ตัวอย่างของตัวแบบปัญหา 34
ส่วนการวิเคราะห์ข้อมูลทั่วไปของคู่มือ 87
ผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของเนื้อหาคู่มือ 89
ผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของข้อเสนอแนะ 90
บทที่ 5 สรุปการวิจัย อภิปรายผล และข้อเสนอแนะ 91
สรุปผลการศึกษาวิจัย 91
อภิปรายผล 92
ข้อเสนอแนะ 92
บรรณานุกรม 94
ภาคผนวก 96
ก ตัวอย่างของตัวแบบประเภทต่างๆ ที่มีเพิ่มเติมในโปรแกรม LINGO 97
ข ตัวอย่างของแบบใบประเมินคู่มือ 109
ค เอกสารการประเมินคู่มือจากผู้ทำการประเมิน 111
ประวัติผู้ศึกษา 124

สารบัญตาราง

หน้า

ตารางที่ 4.1 แสดงส่วนของผลการวิเคราะห์ข้อมูลทั่วไปของแบบประเมินคู่มือ 87

ตารางที่ 4.2 ส่วนของผลการวิเคราะห์ข้อมูลร้อยละของผู้ตอบแบบประเมิน โดยจำแนกตามอายุ 88

ตารางที่ 4.3 ส่วนของผลการวิเคราะห์ข้อมูลร้อยละของผู้ตอบแบบประเมิน โดยจำแนกตามการศึกษา 88

ตารางที่ 4.4 แสดงผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของเนื้อหาคู่มือ 89

ตารางที่ 4.5 แสดงผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของระดับความรู้ 90

สารบัญภาพ

หน้า

ภาพที่ 2.1 กราฟแสดงความสัมพันธ์ของราคาและปริมาณการซื้อมะเขือเทศ 5
ภาพที่ 2.2 กราฟแสดงความสัมพันธ์ของฟังค์ชั่น X*@COS(3.1416*X) 7
ภาพที่ 2.3 กราฟแสดงความสัมพันธ์ของฟังค์ชั่น .4*(X-3)2+.5 9
ภาพที่ 2.4 กราฟแสดงลักษณะของโค้งนูนอิสระ 10
ภาพที่ 2.5 กราฟแสดงลักษณะของโค้งเว้า 10
ภาพที่ 2.6 กราฟแสดงการเกิดการหักงออย่างกระทันหันในฟังค์ชั่นของ ABS(X) 11
ภาพที่ 2.7 รูปแสดงลักษณะแถบเครื่องมือ 12
ภาพที่ 2.8 รูปแสดงรายละเอียดของปุ่มในแถบเครื่องมือ 12
ภาพที่ 2.9 รูปแสดงรายละเอียดในคำสั่งของ File เมนู 14
ภาพที่ 2.10 รูปแสดงหน้าจอคำสั่ง File New 15
ภาพที่ 2.11 รูปแสดงหน้าจอคำสั่ง File Open 16
ภาพที่ 2.12 รูปแสดงหน้าจอคำสั่ง File/Save As… 16
ภาพที่ 2.13 รูปแสดงหน้าจอคำสั่ง File/Print… 17
ภาพที่ 2.14 รูปแสดงหน้าจอคำสั่ง File/Print Setup 17
ภาพที่ 2.15 รูปแสดงหน้าจอคำสั่ง File/Print Preview 18
ภาพที่ 2.16 รูปแสดงกรอบหน้าต่าง License Key. 19
ภาพที่ 2.17 รูปแสดงกรอบหน้าต่าง Database User Info 19
ภาพที่ 2.18 รูปแสดงรายละเอียดของคำสั่ง เมนู Edit 20
ภาพที่ 2.19 รูปแสดงการเปิดไฟล์ Excelเพื่อloadข้อมูล 21
ภาพที่ 2.20 รูปแสดงกล่องข้อความ Paste Spacial 22
ภาพที่ 2.21 รูปแสดงโปรแกรมการload ข้อมูลจากไฟล์ Excel 22
ภาพที่ 2.22 รูปแสดงกล่องข้อความคำสั่ง Edit/Find… 23
ภาพที่ 2.23 รูปแสดงกล่องข้อความคำสั่ง Edit/Replace 23
ภาพที่ 2.24 รูปแสดงกล่องข้อความคำสั่ง Edit/Go to Line 23
ภาพที่ 2.25 รูปแสดงคำสั่ง Edit/Paste Function 24
ภาพที่ 2.26 รูปแสดงกล่องข้อความคำสั่ง Edit/Link 24
ภาพที่ 2.27 รูปแสดงรายละเอียดของคำสั่ง เมนู LINGO 25

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 2.28 รูปแสดงรายละเอียดของกรอบข้อความผิดพลาด 25
ภาพที่ 2.29 รูปแสดงสถานะการดำเนินงาน 26
ภาพที่ 2.30 รูปแสดงสถานะการแสดงผลของคำตอบ 26
ภาพที่ 2.31 รูปแสดงหน้าต่างคำสั่ง LINGO/Solution 27
ภาพที่ 2.32 รูปแสดงรายละเอียดของคำสั่ง LINGO/Option 27
ภาพที่ 4.1 รูปแสดงหน้าต่าง LINGO License Key 33
ภาพที่ 4.2 รูปแสดงหน้าต่างโปรแกรม WINDOW 34
ภาพที่ 4.3 รูปแสดงสถานะการดำเนินงานของโปรแกรม 37
ภาพที่ 4.4 รูปแสดงรายละเอียดของรายละเอียดของการขนส่ง 42
ภาพที่ 4.5 รูปแสดงหน้าต่างของ Solution Report 46
ภาพที่ 4.6 รูปแสดงความสัมพันธ์ของ Set 54
ภาพที่ 4.7 รูปแสดงการแจ้งเตือนเมื่อเกิดความผิดพลาด 57
ภาพที่ 4.8 กราฟแสดงรายละเอียดของการพยากรณ์ในช่วงควอเตอร์ต่อไป 80

บทที่ 1
บทนำ

1. ความเป็นมาและความสำคัญของปัญหา

การศึกษาในวิชาทางด้านบริหารธุรกิจนั้นรายวิชาหนึ่งซึ่งมีความสำคัญและนักศึกษาทางด้านบริหารธุรกิจต้องผ่านการศึกษาในรายวิชานี้คือวิชาการวิเคราะห์เชิงปริมาณ (Quantitative Analysis) โดยวิชาดังกล่าวอาจเรียกกันโดยทั่วไปว่า การวิเคราะห์เชิงปริมาณ, การวิจัยเชิงปฏิบัติการ หรือ การวิจัยดำเนินการซึ่งในเนื้อหาเป็นการนำข้อมูลทางตัวเลขที่มีอยู่มาประมวลเพื่อช่วยในการตัดสินใจ (มหาวิทยาลัยสุโขทัยธรรมาธิราช.บัณฑิตศึกษา. 2551: 1-10) ได้ให้ความหมายของการวิเคราะห์เชิงปริมาณว่าหมายถึงการนำข้อมูลที่เป็นตัวเลขมาประมวลให้ได้องค์ความรู้ (Knowledge) หรือสารสนเทศ (Information) เพื่อใช้ในการตัดสินใจทางธุรกิจ และการตัดสินใจทางด้านการจัดการมักจะเกี่ยวข้องกับการใช้ทรัพยากรขององค์การให้เป็นไปอย่างมีประสิทธิภาพ ทรัพยากรขององค์การโดยทั่วไปหมายถึง เครื่องจักร แรงงาน เงินทุน เวลา พื้นที่คลังสินค้า หรือวัตถุดิบ เป็นต้น โดยเครื่องมือทางคณิตศาสตร์ที่นำมาใช้เพื่อช่วยในการแก้ไขปัญหาดังกล่าวนั้นคือโปรแกรมเชิงเส้น (Linear Programming) ซึ่งก็คือสมการต่างๆซึ่งเขียนขึ้นเพื่ออธิบายตัวแบบของปัญหานั้นๆ

การแก้ปัญหาทางธุรกิจด้วยโปรแกรมเชิงเส้นนั้นสามารถที่จะหาคำตอบได้ด้วยวิธีกราฟ (Graphical method) และ วิธีซิมเพล็กซ์ (Simplex method) หากแต่ว่าการนำไปใช้จริงนั้นนิยมในการใช้โปรแกรมทางคอมพิวเตอร์เพื่อช่วยในการแก้ปัญหาอันเนื่องมาจากตัวแบบของปัญหาที่มีความสลับซับซ้อนรูปแบบของปัญหาที่มีขนาดใหญ่ ตัวแปรที่มีจำนวนมากตามรูปแบบปัญหานั้นๆ การใช้โปรแกรมในการแก้ปัญหาเชิงเส้นเช่น โปรแกรม LINGO (Optimization Modeling Software for Linear, Nonlinear, and Integer Programming) หรือ LINDO (Linear Interactive discrete optimizer) โปรแกรมสำเร็จรูป AB: QM หรือ โปรแกรม QM for Windows เป็นต้น โปรแกรม LINGO เป็นโปรแกรมหนึ่งซึ่งนำมาใช้แก้ปัญหาทางคณิตศาสตร์แต่ก็ยังไม่เป็นที่รู้จักหรือใช้งานอย่างแพร่หลายมากนักในประเทศไทย ทางผู้ค้นคว้าเล็งเห็นถึงความสะดวกสบายของการใช้โปรแกรมในการช่วยแก้ปัญหาและการหาคำตอบที่สามารถจะกระทำได้อย่างรวดร็วมากยิ่งขึ้น สำหรับผู้ที่สนใจที่จะใช้ประโยชน์จากโปรแกรมซึ่งมีรายละเอียดและความสามารถหลายประการ

ผู้ค้นคว้าจึงทำการศึกษารายละเอียดและวิธีการทำงานของโปรแกรมเพื่อรวบรวมและจัดทำเป็นคู่มือในการใช้งานของโปรแกรมดังกล่าว ในส่วนงานและแผนกงานของผู้ดำเนินการศึกษาค้นคว้าเองก็มีการดำเนินงานที่เกี่ยวข้องและใช้งานเกี่ยวกับการวิเคราะห์เชิงปฏิบัติการตัวอย่างเช่นการตัดสินใจภายใต้เงื่อนไขของข้อมูลที่มีอยู่เนื่องจากต้องมีหน้าที่รับผิดชอบเกี่ยวกับการตัดสินใจ การวางแผนการเตรียมการเพื่อการผลิตและการปรับปรุงงานเป็นต้น ทั้งหมดที่กล่าวมานี้เป็นงานในแผนกวิศวกรรมของบริษัท วาย เอส ภัณฑ์ จำกัด ซึ่งตั้งอยู่ที่นิคมอุตสาหกรรมเวลล์โกรว์ จ.ฉะเชิงเทรา โดยบริษัทได้เริ่มก่อตั้งเมื่อปี พ.ศ. 2539 โดยบริษัทดำเนินธุรกิจการผลิตชิ้นส่วนรถยนต์เป็นหลัก ในส่วนงานของผู้ทำการศึกษาค้นคว้าหากดำเนินการโดยที่สามารถใช้โปรแกรมมาช่วยในการหาคำตอบหรือประมวลผลก็น่าที่จะทำให้การทำงานเกิดความสะดวกและรวดเร็วและถูกต้องเพิ่มมากยิ่งขึ้น

2. วัตถุประสงค์การศึกษา

2.1 เพื่อจัดทำคู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์และรายละเอียดของโปรแกรมเพื่อนำไปใช้งาน

2.2 เพื่อประเมินผลคู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์โดยใช้พนักงานแผนกวิศวกรรมบริษัท วาย เอส ภัณฑ์ เป็นกลุ่มตัวอย่างในการประเมิน

3. ประโยชน์ที่คาดว่าจะได้รับ

3.1 เพื่อที่จะได้เป็นประโยชน์ให้แก่ผู้ที่สนใจที่จะศึกษาและผู้ที่ต้องการที่จะใช้งานโปรแกรมในการแก้ปัญหาเชิงเส้นทางคณิตศาสตร์โดยการนำโปรแกรม LINGOไปใช้งานให้ได้รับความสะดวกและรวดเร็วเพิ่มมากยิ่งขึ้น

3.2 เพื่อที่จะได้เป็นประโยชน์ให้แก่ผู้ที่สนใจที่จะศึกษาและผู้ที่ต้องการที่จะใช้งานได้เข้าใจถึงตัวแบบทางคณิตศาสตร์ที่โปรแกรมLINGOสามารถที่จะดำเนินการแก้ปัญหาเพื่อที่จะได้นำไปเป็นตัวอย่างเทียบเคียงเพื่อการใช้งาน

4. ประเด็นปัญหาที่ทำการศึกษา

เป็นการศึกษารายละเอียดต่างๆของโปรแกรม LINGO ฟังค์ชั่นการทำงานต่างๆของโปรแกรม การสร้างตัวแบบของปัญหาและการนำโปรแกรมไปใช้งาน

5. ขอบเขตของการศึกษาคู่มือ

ขอบเขตการศึกษาคู่มือเป็นการศึกษาการใช้งานโปรแกรม LINGO Version 12.0 จากเนื้อหาของคู่มือการใช้งานต้นฉบับภาษาต่างประเทศเพื่อที่จะสามารถอธิบายถึงรายละเอียดต่างๆของโปรแกรม รูปแบบการนำไปใช้งานและตัวแบบของปัญหาที่สามารถนำไปเป็นตัวแบบเพื่อการประยุกต์ในการใช้งาน อีกทั้งทดลองสร้างตัวแบบขึ้นตามตัวอย่างต้นฉบับ Runโปรแกรม สรุปและอธิบายรายละเอียดต่างๆของการใช้งานโปรแกรมหลังจากนั้นรวบรวมข้อมูลที่ได้ทำการศึกษาจัดทำเป็นคู่มือภาษาไทยนำไปทดลองให้หน่วยงานด้านวิศวกรรมของบริษัท วาย เอส ภัณฑ์จำกัดได้ศึกษาทำความเข้าใจเนื้อหาจากคู่มือ และ ประเมินผลคู่มือดังกล่าว

6. นิยามศัพท์

LINGO หมายถึงโปรแกรมสำหรับการหาคำตอบที่ดีที่สุดของตัวแบบทางด้านเชิงเส้นไม่เป็นเชิงเส้นและที่เป็นจำนวนเต็ม

โมเดลทางคณิตศาสตร์ หมายถึง แบบจำลองหรือตัวอย่างที่สร้างขึ้นเพื่อนำไปใช้งานที่เกี่ยวกับการหาคำตอบที่ดีทีสุดของตัวแบบทางด้านเชิงเส้นไม่เป็นเชิงเส้นและที่เป็นจำนวนเต็ม

บทที่ 2
วรรณกรรมที่เกี่ยวข้อง

ในการศึกษาเรื่องการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของตัวแบบทางด้านคณิตศาสตร์ มีรายละเอียดในส่วนต่างๆของโปรแกรม เพื่อการใช้งานโดยประกอบด้วยวรรณกรรมที่เกี่ยวข้องกับการศึกษาดังต่อไปนี้

1. แนวคิดทางทฤษฎี

(LINDO System Inc. 2010: 618-628) การแก้ปัญหาด้วยโปรแกรม LINGO มีตัวแบบที่แตกต่างกัน 4 ประเภทคือ

1. การแก้ปัญหาทางตรง (A direct solver)
2. การแก้ปัญหาเชิงเส้น (A linear solver)
3. การแก้ปัญหาที่ไม่เป็นเชิงเส้น (A nonlinear solver)
4. และ การแก้ปัญหาแบบอัลกอริทึม (A Branch-and –bound manager)

ในการแก้ปัญหาตัวแบบทางคณิตศาสตร์ของโปรแกรม จะเริ่มด้วยการแก้ปัญหาทางตรง(A direct Solver)โดยเริ่มคำนวนค่าของตัวแปรทั้งหลายที่เป็นไปได้ก่อน ถ้าหากการแก้ปัญหาทางตรงพบกับสิ่งที่ตรงกันกับข้อกำหนดของตัวแปรที่ไม่ทราบค่าตัวใดตัวหนึ่งในโปรแกรมตัวโปรแกรมก็จะกำหนดค่าไปยังตัวแปรที่ไม่ทราบค่าที่ตรงกันกับข้อกำหนดนั้น การแก้ปัญหาทางตรงนี้จะยุติก็ต่อเมื่อหาค่าตัวแปรที่ไม่ทราบค่าได้หรือไม่มีตัวแปรที่รอคอยอยู่ตรงกันกับข้อกำหนดที่เหลืออยู่แล้วเมื่อการแก้ปัญหาทางตรงได้ผลเสร็จสิ้นลง ถ้าหากตัวแปรทั้งหมดได้ถูกคำนวนค่าแล้ว โปรแกรม LINGO ก็จะรายงานผลของตัวแปรนั้นๆออกมา แต่ถ้าหากว่าตัวแปรที่ยังไม่ทราบค่ายังคงมีอยู่ โปรแกรมจะพิจารณากำหนดการแก้ปัญหาโดยพิจารณาจากโครงสร้างและตัวแบบทางคณิตศาสตร์ที่มีอยู่ภายในโปรแกรมเพื่อดำเนินการแก้ไขต่อไป หากตัวแบบที่คงเหลืออยู่นั้นสอดคล้องกับรูปแบบของเชิงเส้น (A linier solver) โปรแกรมก็จะเรียกการแก้ปัญหาเชิงเส้นมาใช้งาน แต่ถ้าหากว่าตัวแบบที่ยังค้างอยู่นี้ประกอบด้วยส่วนหนึ่งส่วนใดที่เป็นรูปแบบที่มิใช่เชิงเส้นโปรแกรมก็จะเรียกการแก้ปัญหาแบบที่มิใช่เชิงเส้น(A non-linier solver) มาใช้งานและ เมื่อตัวแบบประกอบไปด้วยจำนวนเต็มที่มีจำกัด การแก้ปัญหาในลักษณะของ

อัลกอริทึม (Branch-and-bound manager) ก็จะถูกนำมาบังคับใช้โดยในโปรแกรมเองจะเรียกโปรแกรมในลักษณะของเชิงเส้น หรือ โปรแกรมที่มิใช่เชิงเส้นมาเพื่อแก้ปัญหานั้นขึ้นอยู่กับประเภทของตัวแบบที่ยังค้างอยู่รูปแบบของข้อกำหนด (Type of constrains) ตลอดการใช้การแก้ปัญหาทางตรง โปรแกรมทำหน้าที่แทนค่าของตัวแปรในตัวแบบจากข้อกำหนดที่ถูกกำหนดขึ้นส่วนตัวแปรและข้อกำหนดที่เหลือจะถูกกำหนดโดยรูปแบบเชิงเส้นหรือมิใช่เชิงเส้นแบบใดแบบหนึ่ง

2. ข้อกำหนดแบบเชิงเส้น (Linear Constraints)

ถ้าเงื่อนไขของข้อกำหนดที่เป็นคำสั่งเริ่มต้นนั้นเป็นเชิงเส้นนั่นหมายถึงข้อกำหนดนั้นจะไม่ประกอบไปด้วยตัวแปรที่เป็นเลขยกกำลัง2 หรือ กำลัง3 หรือตัวเลขที่มีกำลังที่มากกว่าหนึ่งและทุกๆหน่วยที่เพิ่มขึ้นหรือลดลงของตัวแปร ค่าต่างๆตามข้อกำหนดก็จะเพิ่มขึ้นหรือลดลงตามกันโดยข้อกำหนดที่ได้ถูกบังคับไว้แล้ว สูตรของเชิงเส้นก็คือความสัมพันธ์ของ"เส้นตรง"ซึ่งพื้นฐานของเส้นตรงมีสูตรดังนี้

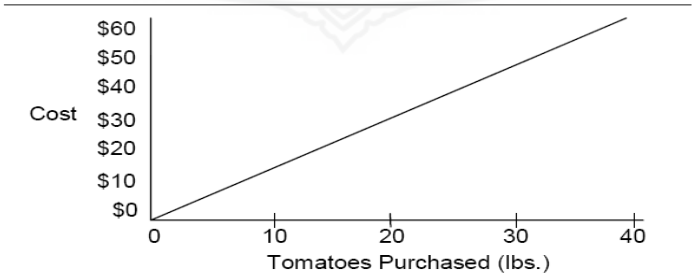
$$Y = m\,X + b$$

เมื่อ m และ b เป็นค่าคงที่

ตัวอย่างเช่น ในการเลือกซื้อมะเขือเทศในราคา 1.5 เหรียญต่อปอนด์ การแสดงฟังค์ชั่นในการคำนวนราคา (C) ในเงื่อนไขของการซื้อมะเขือเทศ (T) คือ

$$C = 1.5 * T.$$

ความหมายก็คือ ราคา(C) จะเท่ากับ 1.5 คูณกับ T (ปริมาณของมะเขือเทศ) เราสามารถที่จะคาดหมายได้ว่ากราฟของฟังค์ชั่นนี้จะเป็นเส้นตรงดังกราฟที่แสดงด้านล่าง



ภาพที่ 2.1 กราฟแสดงความสัมพันธ์ของราคาและปริมาณการซื้อมะเขือเทศ

เส้นตรงที่แสดงออกมาสามารถที่จะแสดงออกมาในรูปแบบที่มีการคูณตัวแปรอื่นๆเพิ่มขึ้นดังตัวอย่างเช่น ถ้าเราเพิ่มตัวแปรเป็นมันฝรั่ง (P) ที่0.75 ต่อ ปอนด์ และ แอปเปิล (A) ที่1.25 ต่อปอนด์ ฟังค์ชั่นที่เพิ่มขึ้นของราคาจะกลายเป็น

$$C = 1.5 * T + 0.75 * P + 1.25 * A$$

ความหมายก็คือ ราคา(C) จะเท่ากับ 1.5 คูณกับ T (ปริมาณของมะเขือเทศ) บวกกับ 0.75คูณกับ P (ปริมาณของมันฝรั่ง) บวกกับ 1.25 คูณกับ A (ปริมาณของแอปเปิล) ซึ่งในราคาใหม่นี้ก็ยังแสดงผลออกมาในรูปของเส้นตรงนั่นเองเนื่องจากตัวแบบที่มีลักษณะเป็นเชิงเส้นสามารถที่จะหาคำตอบได้รวดเร็ว และ มีความเที่ยงตรงกว่าตัวแบบที่มิใช่เชิงเส้นในการสร้างตัวแบบจึงนำเสนอให้สร้างสูตรต้นแบบโดยใช้แบบเชิงเส้นถ้าสามารถที่จะกระทำได้ เมื่อโปรแกรมดำเนินการแก้ปัญหาจากคำสั่งที่ได้รับโปรแกรมจะทำการวิเคราะห์ความสัมพันธ์ของตัวแบบ ถ้าทั้งหมดถูกกำหนดในรูปแบบเชิงเส้น โปรแกรมจะรู้จักและได้ประโยชน์จากตัวแบบเพิ่มมากขึ้น

3. ข้อกำหนดที่ไม่เป็นเชิงเส้น (Nonlinear Constraints)

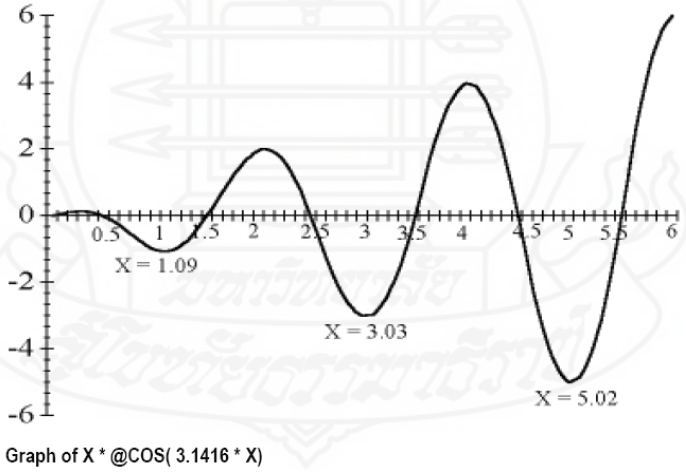
โดยนิยามแล้ว ข้อกำหนดทั้งหลายทั้งปวงนั้นมีอยู่สองลักษณะคือตัวแบบที่เป็นเชิงเส้นและไม่เป็นเชิงเส้น นิพจน์ของสมการที่ไม่ใช่เชิงเส้นจะประกอบไปด้วยตัวแปรซึ่งมีกำลังสอง หรือกำลังสามหรือมีกำลังที่มากกว่าหนึ่งหรือการคูณและหารซึ่งกันและกันตัวแบบของสมการที่ไม่เป็นเชิงเส้นจะทำการหาคำตอบได้ยากกว่าสมการที่เป็นเชิงเส้น ตัวแบบที่ไม่เป็นเชิงเส้นอาจทำให้โปรแกรมปราศจากการหาคำตอบ หรือโปรแกรมอาจหาคำตอบของตัวแบบที่ไม่เป็นเชิงเส้นโดยปรากฏว่าเป็น"เป็นสิ่งที่คิดว่าดีที่สุด" ซึ่งคำตอบนี้อาจไม่พึงปราถนาให้ปรากฏขึ้น เนื่องจากเราต้องการคำตอบที่เป็นค่าน้อยที่สุดหรือมากที่สุด ให้เราพิจารณาจากแนวทางของตัวแบบที่ไม่เป็นเชิงเส้นดังนี้ (Local Optima และ Global Optima)เมื่อโปรแกรมหาผลลัพธ์ที่เป็นค่าต่ำสุดของโปรแกรมเชิงเส้น คำตอบที่ได้จะเป็นผลลัพธ์สุดท้ายที่ดีที่สุดเราสามารถที่จะเรียกได้ว่า เป็นผลลัพธ์ที่ดีที่สุดจากทั้งหมด (Global Optimum) สภาวะของการคูณอาจคงอยู่ อย่างไรก็ตามผลลัพธ์ที่ดีที่สุดจากทั้งหมดนั้นคือผลลัพธ์ที่เป็นไปได้จากตัวแปรวัตถุประสงค์ซึ่งดีหรือมีความเป็นไปได้มากกว่าผลลัพธ์อื่นๆในตัวแบบ ความสามารถที่จะแสดงผลลัพธ์ที่ดีที่สุดจากทั้งหมดนั้นก็เนื่องมาจากได้คุณสมบัติที่แน่นอนของตัวแบบที่เป็นเชิงเส้นนั่นเอง แต่ลักษณะนี้จะไม่ใช่ในกรณีของผลลัพธ์ที่ดีที่สุดของตัวแบบที่ไม่ใช่เชิงเส้น เนื่องจากอาจเกิดผลลัพธ์ที่ดีที่สุดขึ้นหลายครั้งนั่นก็คือ จุดที่ดีที่สุด

เฉพาะที่(Locally Optimal) ประเภททั้งหมดของการแก้ปัญหาที่มิได้เป็นเชิงเส้นนั้นจะบรรจบกันเข้าสู่จุดที่เป็นจุดที่ดีที่สุดเฉพาะที่ การหาผลลัพธ์ของจุดที่ดีที่สุดเฉพาะที่เพิ่มขึ้นอาจทำให้ผลลัพธ์ที่เกิดขึ้นห่างไกลจากผลลัพธ์ที่มีอยู่ในปัจจุบันแต่ก็มีผลดีคืออาจแสดงคุณค่าให้แก่วัตถุประสงค์ของตัวแบบที่มีแก่นสารและประโยชน์มากกว่าผลลัพธ์ของจุดที่ดีที่สุดเฉพาะที่ในปัจจุบัน ด้วยเหตุนี้เมื่อตัวแบบที่มิใช่เชิงเส้นได้ถูกแก้ปัญหา เราจะกล่าวว่าผลลัพธ์นั้นจะเป็นจุดที่ดีที่สุดเฉพาะที่เท่านั้น ผู้ใช้ควรทราบว่าอาจมีหรือไม่มีจุดที่ดีที่สุดอื่นๆเกิดขึ้นอีกขึ้นอยู่กับฟังค์ชั่นของวัตถุประสงค์นั้น

พิจารณาตัวแบบที่ไม่เป็นเชิงเส้นที่อาจนำไปสู่ตัวแบบที่แสดงรายละเอียดได้มากขึ้นจากฟังค์ชั่นด้านล่าง

$$MIN = X * @COS(3.1416 * X);$$
$$X < 6;$$

กราฟที่เกิดขึ้นได้จากวัตถุประสงค์ของฟังค์ชั่นของค่า X ตั้งแต่ 0 ถึง 6 ถ้าเราหาค่าน้อยที่สุดมีจุดที่ดีที่สุดเฉพาะที่ที่ X เท่ากับ 0,1.09,3.03,และ5.02 ที่อยู่ในช่วงๆหนึ่งคือX<6 จุดที่ดีทีสุดของปัญหานี้ก็คือที่ X=5.02 เพราะว่าอยู่ในช่วงที่ต่ำสุด ลักษณะของกราฟที่เกิดจากฟังค์ชั่นนี้สามารถแสดงออกมาได้เป็นรูปแบบดังนี้



ภาพที่ 2.2 กราฟแสดงความสัมพันธ์ของฟังค์ชั่น X*@COS(3.1416*X)

สภาพของกราฟที่แสดงออกมามีลักษณะที่พิจารณาได้ว่าคล้ายกับภูเขาต่อเนื่องกัน ซึ่งเรากำลังหาจุดที่น้อยที่สุดหรือต่ำที่สุด สภาวะของกราฟที่พิจารณาณ.จุดที่ X=2.5 ทุกๆ จุดที่เคลื่อนที่เข้าหาค่า 2.0 จะทำให้เกิดสภาพที่ไต่ขึ้นไปตามความชันของกราฟ และ ทุกๆจุดที่เข้าหาค่า 3.0นั้นก็จะทำให้เกิดสภาพที่ไต่ลงจากความชันของกราฟอย่างไรก็ตามการเคลื่อนที่เข้าหาค่า 3.0นั้น

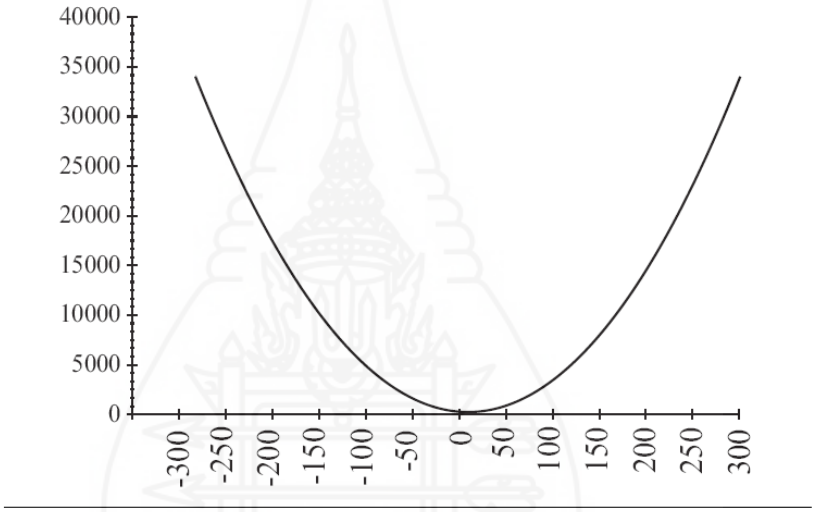
การหาจุดที่ต่ำสุดก็จะต้องเคลื่อนที่อย่างต่อเนื่องเพื่อที่จะนำไปแตะจุดที่ต่ำสุด ดังนั้นเมื่อเราเข้าไปถึงจุดที่ X=3.03 ก็จะสังเกตุได้ว่ามีพื้นที่ราบน้อยๆ (ความชันเท่ากับ 0) เมื่อเคลื่อนที่ต่อเนื่องไปก็จะเคลื่อนที่ขึ้นไปตามรูปของกราฟ และ หากถอยจากการเคลื่อนที่ขึ้นเราก็จะกลายเป็นการเคลื่อนที่ลง ขณะที่อยู่ในช่วงที่ว่านี้จุดที่ต่ำที่สุดก็จะใกล้กับ ผลลัพธ์ที่ดีที่สุด อย่างไรก็ดีจุดนี้เป็นจุดที่ต่ำที่สุดใช่หรือไม่? ในขณะที่เรายังไม่พบกับคำตอบที่ชัดเจนเราก็ยังไม่สามารถที่จะตอบคำถามได้อย่างแน่นอนว่า ณ. จุดที่เราพิจารณาอยู่นี้เป็นจุดที่ต่ำที่สุดจริงโปรแกรมจะให้เราใส่ค่าสำหรับตัวแปรตั้งต้น (เช่นจุดที่เริ่มต้นสำหรับการค้นหา) โดยใช้คำสั่ง INIT Section ความแตกต่างของจุดที่ดีที่สุดเฉพาะที่อาจจะขยายไปถึงตัวแบบที่ได้ถูกแก้ปัญหาในจุดที่แตกต่างกันของค่า X ดังในตัวอย่างนี้ค่าตั้งต้นของ X=6 ดังนั้นผลลัพธ์ที่ดีที่สุดโดยรวมจะอยู่ที่ X=5.02 มีข้อด้อยที่ว่ามันยังไม่สามารถที่จะรับประกันคำตอบได้เพราะโปรแกรมจะประมาณหาความจริงภายใต้ฟังค์ชั่นที่ไม่เป็นเชิงเส้นโดยอาศัยฟังค์ชั่นของเชิงเส้นและหรือสมการกำลังสองเพื่อหาคำตอบในขั้นตอนที่เกิดขึ้นอย่างรวดเร็วของตัวแบบนี้ สามารถบอกผลลัพธ์โดยปกติออกมาเป็นการประมาณค่าบางส่วนคร่าวๆได้ อย่างไรก็ดีในการหาผลลัพธ์กรอบของตัวแบบสามารถที่จะส่งผลไปสู่จุดที่ดีที่สุดเฉพาะที่จะเป็นคำตอบที่แสดงออกมาในตัวแบบได้ เมื่อทราบค่าเริ่มแรกแล้ว เราควรที่จะใส่ค่าใน INIT Section เพิ่มเติมและเราอาจต้องใช้ฟังค์ชั่น @BND เพื่อกำหนดตัวแปรไว้ ด้วยเหตุผลที่ว่าส่วนอื่นที่ใกล้กับจุดเริ่มต้นที่เราไม่ทราบว่าในส่วนใดจะได้ผลลัพธ์ที่ดีที่สุดของตัวแบบนี้ เราอาจหาโดยสังเกตุจากผลของการแก้ปัญหาของตัวแบบหลายๆครั้งโดยกำหนดค่าตั้งต้นที่แตกต่างกันออกไปก็จะสามารถช่วยให้หาคำตอบที่ดีที่สุดได้กล่าวคือว่าเราขยายขอบเขตการหาค่าของค่า X ออกไปนั่นเองก็จะทำให้เราทราบผลลัพธ์ของจุดที่ดีที่สุดเฉพาะที่เพิ่มเติมและเป็นประโยชน์มากยิ่งขึ้น โปรแกรมมีกลยุทธ์ทางเลือกหลายอย่างเพื่อช่วยในการแก้ปัญหาเมื่อเกิดการติดชะงักที่จุดที่ดีที่สุดเฉพาะที่โดยการใช้วิธีแก้ปัญหาแบบอัลกอริทึม (Branch-and-bound) ในการแยกรูปแบบของตัวแบบออกเป็นพื้นที่ความโค้งหลายๆส่วนโปรแกรมมีการเริ่มต้นได้ทีละหลายๆตัวแบบและจะทำให้สามารถหาตัวเลขของจุดที่ดีที่สุดเฉพาะที่และรายงานเมื่อพบจุดที่ดีที่สุดได้สะดวกยิ่งขึ้น

โค้งนูน (Convexity) คุณสมบัติที่แสดงถึงความโค้งนูนและรูปแบบที่ใกล้เคียงกันที่เรียกว่าความโค้งเว้าเป็นการใช้ประโยชน์ให้กับตัวแบบเพื่อรับประกันว่าจุดที่ดีที่สุดเฉพาะที่เป็นจุดที่ดีที่สุดของทั้งหมด มาลองพิจารณาคุณสมบัติของความโค้งนูนในตัวแบบปัญหาที่ต้องการคำตอบน้อยที่สุดตัวอย่าง ข้อกำหนดทางคณิตศาสตร์ของตัวแบบความโค้งนูนมีดังนี้

$$f(y) \le a\,f(x) + (1\text{-}a)\,f(z), \text{ where } y=a^*x + (1\text{-}a)^*z$$

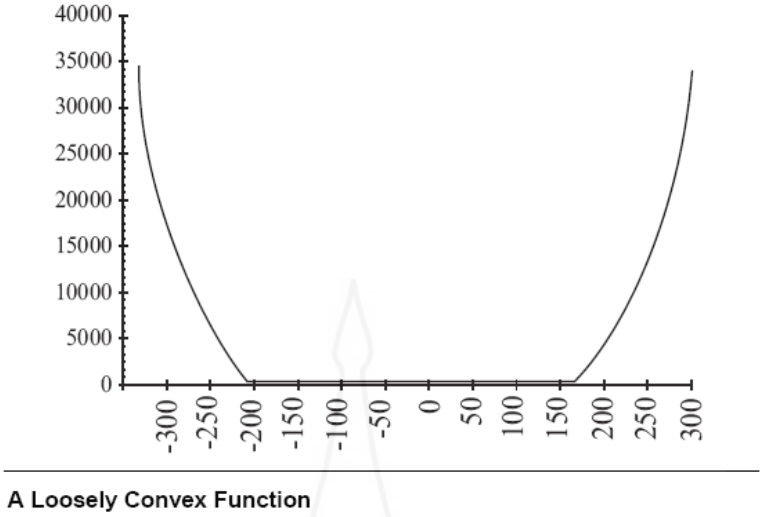
การหาโค้งนูนของปัญหาหลายตัวแปรนั้นเป็นงานที่ยาก นักคณิตศาสตร์จะเรียกว่าฟังค์ชั่นของการโค้งนูนถ้าแมตริกอันดับสองของอนุพันธ์เป็นบวกหรือมีค่าลักษณะเฉพาะเป็นบวก อย่างไรก็ตามหากเราจะสามารถกำหนดตัวแบบโปรแกรมของเราเป็นโค้งนูนเพื่อต้องการค่าน้อยที่สุดได้เราก็สามารถที่จะมั่นใจได้ว่าคำตอบจะเป็นขอบเขตที่แสดงถึงค่าที่ดีที่สุดของทั้งหมด (รวมถึงตัวแบบที่ไม่เป็นเชิงเส้นด้วย)

โค้งนูนที่แท้จริง (Strictly Convex)ฟังค์ชั่นโค้งนูนที่แท้จริงมีจุดที่น้อยที่สุดที่จุดหักของโค้งที่เป็นผลที่ดีที่สุดของทั้งหมดกราฟด้านล่างจะแสดงความโค้งนูนของฟังค์ชั่นที่มีเพียงตัวแปรเดียว



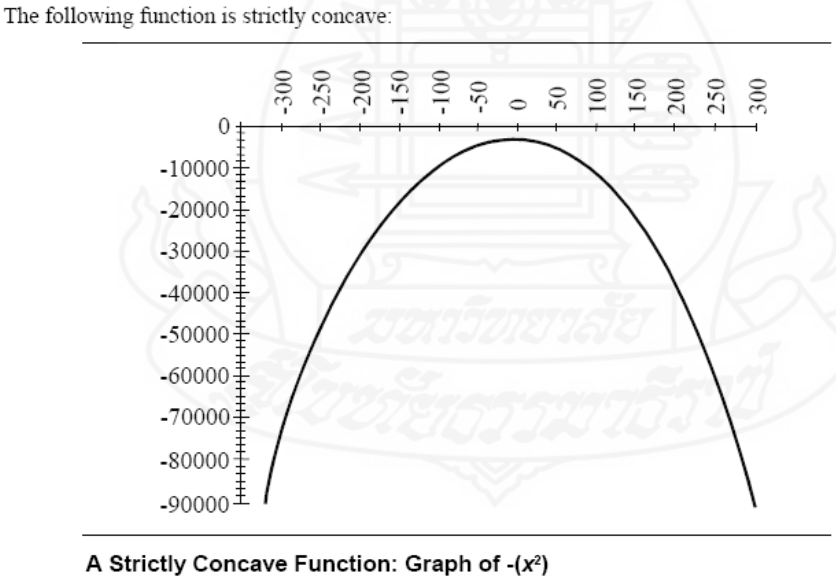
ภาพที่ 2.3 กราฟแสดงความสัมพันธ์ของฟังค์ชั่น .4*(X-3)2+.5

ในฟังค์ชั่นของโค้งนูนที่แท้จริงนี้ ค่าที่น้อยสุดจะมีเพียงหนึ่งเดียว สามารถกำหนดจุดที่ฟังค์ชั่นตัวแปรของ X เท่ากับ3 การเปลี่ยนค่าของ X มากกว่าหรือน้อยกว่า 3 ผลลัพธ์ของฟังค์ชั่นก็จะเพิ่มขึ้นโค้งนูนอิสระ (Loosely Convex) ฟังค์ชั่นของโค้งนูนอิสระนั้นข้อกำหนดจะไม่มีค่าต่ำสุดเฉพาะที่ อย่างไรก็ดี ค่าน้อยที่สุดของโค้งนูนอิสระเกิดจากผลลัพธ์ที่แตกต่างของค่าตั้งต้นของตัวแปร ถ้าหากเราทราบว่าฟังค์ชั่นเป็นโค้งนูนอิสระเราก็จะทราบผลลัพธ์ว่าเป็นค่าน้อยสุดโดยรวม เช่น พื้นที่ที่เป็นแนวราบเมื่อความชันเป็น 0 จากค่าที่ -200 ถึง 150 ในฟังค์ชั่นที่แสดงอยู่ในรูปด้านล่างนี้เป็นตัวแบบของโค้งนูนอิสระ



ภาพที่ 2.4 กราฟแสดงลักษณะของโค้งนูนอิสระ

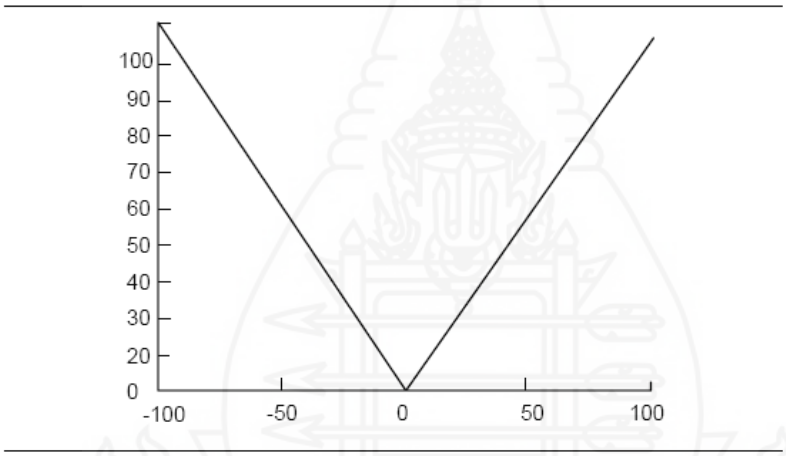
โค้งเว้า (Concavity) เมื่อโค้งนูนได้ถูกนำมาเพื่อแก้ปัญหาค่าน้อยที่สุด โค้งเว้าก็จะใช้เพื่อแก้ปัญหาของค่าที่มากที่สุดของฟังค์ชั่นด้วย โค้งเว้าเป็นด้านที่ตรงข้ามกันกับโค้งนูน ดังจะแสดงในรูปด้านล่าง



ภาพที่ 2.5 กราฟแสดงลักษณะของโค้งเว้า

ในฟังค์ชั่นของโค้งเว้าที่แท้จริงค่าสูงสุดของทั้งหมดสามารถหาได้ที่จุดเมื่อตัวแปร X เท่ากับ 0 การเปลี่ยนแปลงไปของตัวแปรมากกว่าหรือน้อยกว่า 0 จะทำให้ผลลัพธ์ของฟังค์ชั่นมีค่า

น้อยลง ฟังค์ชั่นของโค้งเว้าอย่างแท้จริงอาจดูเหมือนกับโค้งนูนแต่จะกลับด้านกันถ้าหากว่าเราสามารถหาคำตอบของตัวแบบที่ไม่เป็นเชิงเส้นแล้วพบว่าไม่เป็นโค้งนูนหรือโค้งเว้าให้ทราบได้ทันทีเลยว่าเป็นฟังค์ชั่นที่เกิดขึ้นจากการผสมผสานกัน และจะเป็นการดีที่ทำการแก้ปัญหาโดยใช้ค่าเริ่มต้นที่แตกต่างกันออกไปของตัวแปรเพื่อหาคำตอบให้ดูที่รูปแบบของ Local Optima vs Global Optima ฟังค์ชั่นที่สม่ำเสมอ และ ไม่สม่ำเสมอ (Smooth vs Non-smooth Function) ฟังค์ชั่นที่สม่ำเสมอมีข้อกำหนดด้วยอนุพันธ์อันดับหนึ่ง (ความชัน หรือ ความราบ) ที่ทุกๆจุด กราฟของฟังค์ชั่นที่สม่ำเสมอของตัวแปรเพียงตัวเดียวสามารถที่จะสร้างเป็นเส้นต่อเนื่องซึ่งไม่มีการหักงออย่างทันทีทันใดหรือขาดออกจากกันในทุกๆช่วงตัวอย่างต่างๆที่เราได้เห็นมาแล้วนั้นเป็นตัวอย่างของความต่อเนื่องกันฟังค์ชั่นที่ไม่ต่อเนื่องจะมีลักษณะของเส้นที่เกิดการหักงออย่างกระทันหันค่าสมบูรณ์ของตัวแปร ในฟังค์ชั่น@ABS(X) ก็คือตัวอย่างของรูปแบบที่กล่าวนี้



Graph of *ABS(X)*

ภาพที่ 2.6 กราฟแสดงการเกิดการหักงออย่างกระทันหันในฟังค์ชั่นของ ABS(X)

นี่คือตัวอย่างของกราฟที่เกิดการหักงอขึ้นที่จุด0 ฟังค์ชั่นที่ไม่ต่อเนื่องเช่น @MAX,@MIN,@SMAX,@SMIN อาจจะมีมากกว่าฟังค์ชั่นที่ทำให้เกิดการหักงออย่างกระทันหันคือฟังค์ชั่นที่ไม่ต่อเนื่อง ในตัวโปรแกรมจะรวมถึงฟังค์ชั่น @SIGN และ @FLOOR

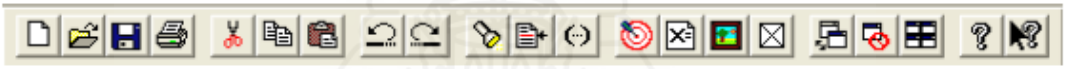
4. คำสั่งของโปรแกรมที่ใช้ใน WINDOW

(LINDO System Inc. 2010: 117-159) ในเมนูของโปรแกรม LINGO จะแบ่งกลุ่มคำสั่งของโปรแกรมออกเป็น5กลุ่มคำสั่งด้วยกันกล่าวคือ

1. File เมนูประกอบไปด้วยคำสั่งดังเดิมในการควบคุมข้อมูลนำเข้าและแสดงผล
2. Edit เมนูประกอบด้วยคำสั่งในการแก้ไขเอกสารของหน้าต่างปัจจุบัน
3. LINGO เมนูประกอบด้วยคำสั่งในการแก้ไขปัญหาตัวแบบและสร้างการรายงานผล
4. Window เมนูมีคำสั่งทางด้านกลไกในการใช้วินโดวส์
5. Help เมนูนำเสนอส่วนที่แสดงความช่วยเหลือของโปรแกรม

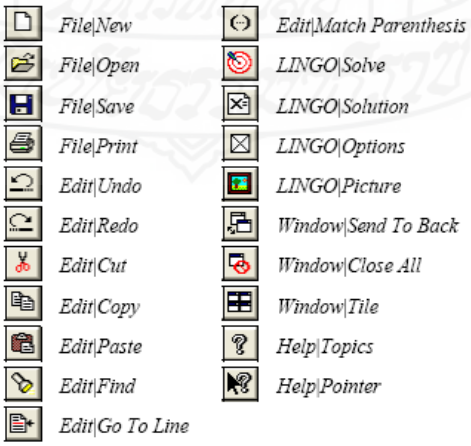
แถบเครื่องมือ (Toolbar)

โดยข้อกำหนดของโปรแกรมแถบเครื่องมือจะแสดงยังส่วนบนของจอภาพซึ่งจะมีลักษณะดังต่อไปนี้



ภาพที่ 2.7 รูปแสดงลักษณะแถบเครื่องมือ

โปรแกรม LINGO จะแสดงรายละเอียดของปุ่มในแถบเครื่องมือเมื่อเลื่อนเมาส์ไปบนปุ่มนั้นๆข้อความสั้นๆเพื่อแสดงให้ทราบถึงชื่อของปุ่มก็จะแสดงออกมาโดยรายละเอียดของปุ่มทั้งหมดจะแสดงให้เห็นตามรายละเอียดด้านล่าง



ภาพที่ 2.8 รูปแสดงรายละเอียดของปุ่มในแถบเครื่องมือ

ในส่วนนี้จะแสดงรายละเอียดคำสั่งในวินโดวส์อย่างรวบรัดเพื่อให้ง่ายต่อการใช้งาน

1. File Menu Commands:

New	Opens a new model window.
Open	Opens an existing model previously saved to disk.
Save	Saves the contents of the current window to disk.
Save As	Saves the contents of the current window to a new name.
Close	Closes the current window.
Print	Prints the contents of the current window.
Print Setup	Configures your printer.
Print Preview	Displays the contents of the current window as it would appear if printed.
Log Output	Opens a log file for logging output to the command window.
Take Commands	Runs a command script contained in a file.
Export File	Exports a model in MPS or MPI file format.
License	Prompts you for a new license password to upgrade your system.
Database User Info	Prompts you for a user id and password for database access via the *@ODBC()* function.
Exit	Exits LINGO.

2. Edit Menu Commands:

Undo	Undoes the last change.
Redo	Redoes the last undo command.
Cut	Cuts the current selection from the document.
Copy	Copies the current selection to the clipboard.
Paste	Pastes the contents of the clipboard into the document.
Paste Special	Pastes the contents of the clipboard into the document, allowing choice as to how the object is pasted.
Select All	Selects the entire contents of the current window.
Find	Searches the document for the occurrence of a specified text string.
Find Next	Repeats the find operation for the last string specified.
Replace	Replaces a specified text string with a new string.
Go To Line	Moves the cursor to a specified line number.
Match Parenthesis	Finds the parenthesis that closes a selected parenthesis.
Paste Function	Pastes a template of a selected LINGO @function.
Select Font	Specifies a font for a selected block of text.
Insert New Object	Embeds an OLE (Object Linking and Embedding) object into the document.
Links	Controls the links to external objects in your document.
Object Properties	Specifies the properties of a selected, embedded object.

3. LINGO Menu Commands:

Solve	Solves the model in the current window.
Solution	Generates a solution report window for the current model.
Range	Generates a range analysis report for the current window.
Options	Sets system options.
Generate	Generates the algebraic representation for the current model.
Picture	Displays a graphical picture of a model in matrix form.
Debug	Tracks down formulation errors in infeasible and unbounded linear programs.
Model Statistics	Displays a brief report regarding the technical detail of a model.
Look	Generates a formulation report for the current window.

4. Window Menu Commands:

Command Window	Opens a command window for command-line operation of LINGO.
Status Window	Opens the solver's status window.
Send to Back	Sends the current window behind all other open windows.
Close All	Closes all open windows.
Tile	Arranges all open windows into a tile pattern.
Cascade	Arranges all open windows into a cascading pattern.
Arrange Icons	Aligns all iconized windows at the bottom of the main frame window.

5. Help Menu Commands:

Help Topics	Accesses LINGO's Help facility.
Register	Registers your version of LINGO online.
AutoUpdate	Checks to see if an updated copy of LINGO is available for download on the LINDO Systems Web site.
About LINGO	Displays the version and size of your copy of LINGO, along with information on how to contact LINDO Systems.

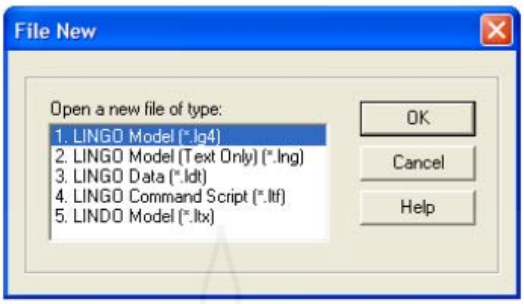
5. คำสั่งที่ใช้ใน Window แบบเชิงลึก

1. File เมนู ของโปรแกรม LINGO จะแสดงดังรูปซึ่งจะประกอบด้วยคำสั่งในเรื่องทั่วไปของการดำเนินการเพื่อนำเข้าและออกของข้อมูล

ภาพที่ 2.9 รูปแสดงรายละเอียดในคำสั่งของ File เมนู

1.1 คำสั่ง File/New F2

เป็นคำสั่งเพื่อเปิดข้อมูลใหม่เมื่อเราเลือกคำสั่ง New เราจะพบกับหน้าจอเครื่องมือดังรูป



ภาพที่ 2.10 รูปแสดงหน้าจอคำสั่ง File New

เราต้องเลือกประเภทของไฟล์ที่เราต้องการสร้างข้อมูลซึ่งก็จะเป็นหนึ่งในประเภทของไฟล์

1. LINGO Model(*.lg4) ฟอร์มของLG4ได้ถูกสร้างขึ้นเมื่อเวอร์ชั่น4ของโปรแกรมLINGOเป็นแบบฟอร์มหลักของโปรแกรมเพื่อใช้เก็บข้อมูลภายใต้การทำงานของ Windows แบบฟอร์มนี้รองรับกับตัวอักษรได้หลายๆแบบ

2. LINGO Model(Text Only)(*.lng) แบบฟอร์ม LNG เป็นแบบฟอร์มในการขนส่งข้อมูลไปเพื่อจัดเก็บซึ่งเป็นไฟล์มาตรฐานใช้ในโปรแกรมซึ่งได้ถูกสร้างขึ้นมาตั้งแต่เริ่มแรกจนกระทั่งเวอร์ชั่นที่4 LNGไฟล์จะถูกจัดเก็บในดิสด้วยชื่อตัวอักษร ASCII

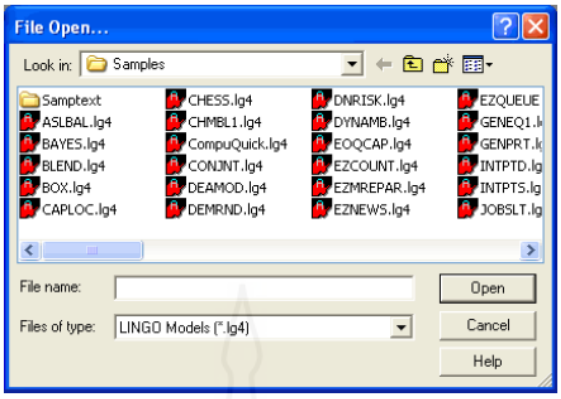
3. LINGO Data(*.ldt) LDTไฟล์เป็นไฟล์ข้อมูลของเครื่องหมายที่นำเข้ามาใช้งานในโปรแกรมเพื่อใช้ในฟังค์ชั่น @FILE ซึ่งฟังค์ชั่นนี้สามารถอ่านได้เฉพาะไฟล์ของตัวอักษร

4. LINGO Command Script(*.ltf) LTFไฟล์เป็นต้นแบบของคำสั่งในโปรแกรมซึ่งประกอบไปด้วย ASCIIไฟล์ในโปรแกรม LTF ไฟล์จะไม่รองรับตัวอักษรหลายๆรูปแบบ

5. LINDO Model (*.ltx) LTXไฟล์เป็นไฟล์ตัวแบบที่ใช้ในโปรแกรมเพื่อสร้างความสัมพันธ์ ในการใช้ความสัมพันธ์ของLINDOนี้เพื่อให้รวดเร็วในการใส่ข้อมูลของโปรแกรมเชิงเส้นที่มีขนาดเล็กและกลาง

1.2 คำสั่ง File/Open Ctrl+O

เป็นคำสั่งเพื่อเปิดข้อมูลเก่าที่มีอยู่เดิมซึ่งโปรแกรมได้จัดเก็บเอาไว้ซึ่งไฟล์จะเป็นรูปแบบของ(*.LG4) เมื่อเราเลือกคำสั่ง เราจะพบกับหน้าจอเครื่องมือดังรูป



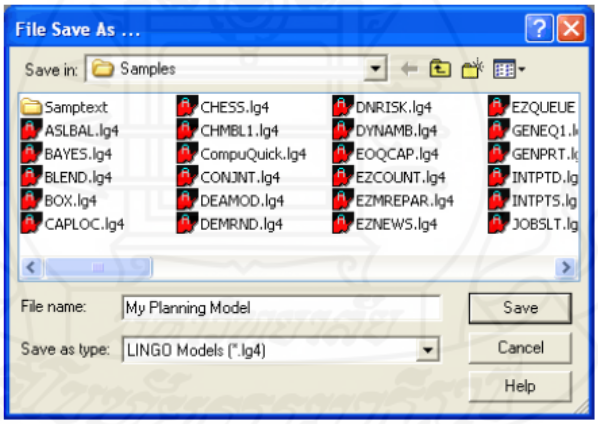
ภาพที่ 2.11 รูปแสดงหน้าจอคำสั่ง File Open

1.3 คำสั่ง File/Save Ctrl+S

เป็นคำสั่งเพื่อเซฟข้อมูลลงในดิสหากว่าไฟล์นั้นยังไม่ได้ถูกตั้งชื่อจะต้องกำหนดชื่อให้ไฟล์นั้นก่อน

1.4 คำสั่ง File/Save As… F5

เป็นคำสั่งเพื่อเซฟข้อมูลจัดเก็บไว้ในชื่ออื่นที่ได้กำหนดขึ้นใหม่ซึ่งก็จะมีกรอบของหน้าต่างแสดงออกมาให้เห็นดังรูป



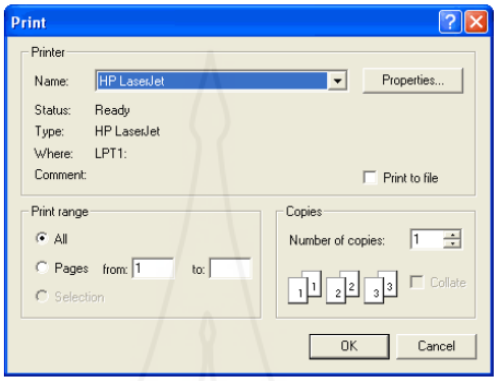
ภาพที่ 2.12 รูปแสดงหน้าจอคำสั่ง File/Save As…

1.5 คำสั่ง File/Close F6

เป็นคำสั่งเพื่อปิดการใช้งานหากหน้าต่างของโปรแกรมได้รับการแก้ไขโดยปราศจากการเซฟของข้อมูล โปรแกรมจะดำเนินการถามก่อนที่จะปิดไฟล์ว่าต้องการที่จะเซฟข้อมูลที่ได้ดำเนินการเปลี่ยนแปลงนั้นหรือไม่

1.6 คำสั่ง File/Print…. F7

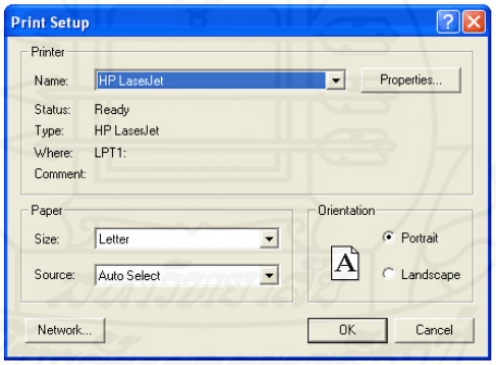
ใช้คำสั่งปรินท์เพื่อส่งข้อมูลไปยังเครื่องปรินท์เอกสาร โดยก่อนอื่นโปรแกรมจะแสดงหน้าจอก่อนการปรินท์ดังรูป



ภาพที่ 2.13 รูปแสดงหน้าจอคำสั่ง File/Print…

1.7 คำสั่ง File/Print Setup F8

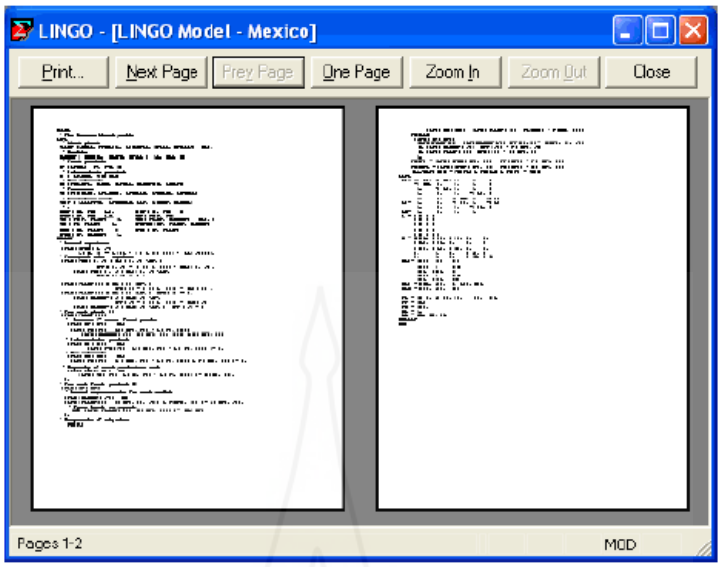
เป็นการใช้คำสั่งเพื่อการกำหนดเครื่องปรินท์เอกสาร โดยจะมีกรอบของหน้าจอแสดงผลดังรูปและให้เรากำหนดชื่อเครื่องปรินท์เอกสาร



ภาพที่ 2.14 รูปแสดงหน้าจอคำสั่ง File/Print Setup

1.8 คำสั่ง File/Print Preview Shift+F8

เป็นคำสั่งเพื่อให้แสดงรูปแบบของเอกสารที่จะดำเนินการปรินท์ซึ่งแสดงสภาพโดยรวมให้ตรวจสอบและพิจารณาพื้นที่ในการปรินท์โดยมีรูปแบบในการแสดงออกมาดังรูป



ภาพที่ 2.15 รูปแสดงหน้าจอคำสั่ง File/Print Preview.

1.9 คำสั่ง File/Log Output *F9*

โดยปกติแล้วเมื่อเราใช้โปรแกรม LINGOสำหรับโปรแกรมวินโดวส์โปรแกรมจะทำงานในรูปแบบของการขับเคลื่อนเมื่อเราเลือกคำสั่งจากการหน้าของเมนูและทำงานเฉพาะในโปรแกรม วินโดวส์ คำสั่ง LogOutput จะเปิดไฟล์วินโดวส์จากชื่อของไฟล์ log เราสามารถสะท้อนผลของคำสั่งในวินโดวส์โดยการตรวจสอบที่หน้าจอของCheckbox.

1.10 คำสั่ง File/Take Commands…. *F11*

คำสั่ง Take Commands ใช้สำหรับยอมให้คำสั่งของโปรแกรม LINGO ปฏิบัติการตามต้นแบบของโปรแกรมเพื่อข้อมูลที่หลากหลายของภาษาในโปรแกรม LINGO

1.11 คำสั่ง File/Export File

คำสั่ง Export File เป็นคำสั่งที่ยอมให้ส่งออกข้อมูลในรูปแบบของ MPSและ MPIไฟล์ ซึ่งMPSไฟล์นั้นเป็นรูปแบบมาตรฐานในอุตสาหกรรมที่ได้รับการพัฒนามาจาก IMB ซึ่งใช้ส่งผ่านตัวแบบในการแก้ไขปัญหาหรือแพลทฟอร์มไปยังรูปแบบอื่นๆ ส่วนรูปแบบMPIได้รับการพัฒนาโดย ระบบของLINDOในการเก็บรวบรวมโปรแกรมทางคณิตศาสตร์เชิงเส้นสู่รายการที่เฉพาะและตัวแบบที่ไม่เป็นในลักษณะเชิงเส้น

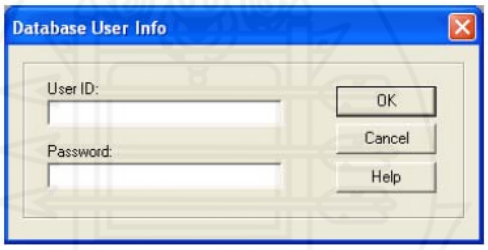
1.12 คำสั่ง File/License

บางเวอร์ชั่นของโปรแกรม LINGO ต้องการให้ผู้ใช้ใส่ค่าของ License Key เมื่อเรา Upgrade โปรแกรมเราต้องดำเนินการใส่ Password ใหม่ เมื่อโปรแกรมทำการดำเนินการทางด้านคำสั่งของ License จะมีกรอบหน้าต่างปรากฏดังรูป

ภาพที่ 2.16 รูปแสดงกรอบหน้าต่าง License Key.

1.13 คำสั่ง File/Database User Info

LINGO โปรแกรมให้ตัวแบบของโปรแกรมเชื่อมโยงโดยตรงกับฐานข้อมูลผ่านการใช้คำสั่งฟังค์ชั่น @ODBC()ในหลายๆครั้งเมื่อทำการเชื่อมโยงข้อมูลโปรแกรมจะร้องขอให้ใส่ Use IDและPassword โดยจะแสดงกรอบของรูปแบบดังรูป

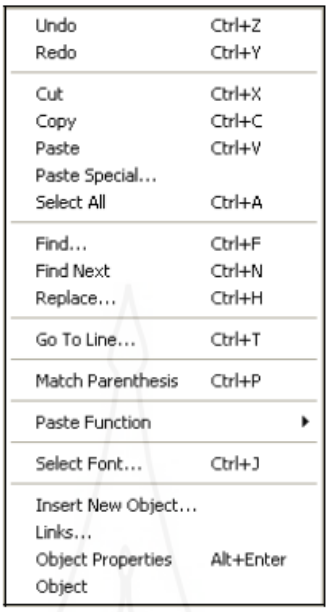


ภาพที่ 2.17 รูปแสดงกรอบหน้าต่าง Database User Info.

1.14 คำสั่ง File/Exit F10

เป็นคำสั่งเพื่อใช้ในการออกจากโปรแกรม LINGOโดยทันที หากว่ายังมีข้อมูลที่ยังไม่ได้ถูกดำเนินการจัดเก็บเปิดอยู่ก็จะถูกดำเนินการจัดเก็บทันทีก่อนที่โปรแกรมจะทำการปิด

2. เมนู Edit จะมีไว้สำหรับแก้ไขตัวอักษรภายในวินโดวส์ รูปแบบของคำสั่งทั้งหมดแสดงรายละเอียดตามตารางด้านล่าง



ภาพที่ 2.18 รูปแสดงรายละเอียดของคำสั่ง เมนู Edit

2.1 คำสั่ง Edit/Undo Ctrl+Z

ใช้คำสั่ง Undo เพื่อยกเลิกการแก้ไขครั้งสุดท้ายของรายละเอียดในวินโดวส์การUndo สามารถที่จะดำเนินการได้ทั้งหมดยกเว้นการทำ Drag-and-Drop หรือการลากและวางโปรแกรมซึ่ง LINGOมีขอบเขตของการยกเลิกการทำงานเช่นเดียวกัน

2.2 คำสั่ง Edit/Redo Ctrl+Y

คำสั่ง Redo เป็นคำสั่งเพื่อให้กลับดำเนินการซ้ำอีกครั้งหนึ่งหลังจากที่ได้ทำการยกเลิกการทำงานในขั้นตอนก่อนหน้านั้นไปแล้ว

2.3 คำสั่ง Edit/Cut Ctrl+X

การใช้คำสั่ง Cut เพื่อตัดข้อความที่ได้ทำการเลือกเพื่อที่จะนำไปวางในวินโดวส์คลิปบอร์ด

2.4 คำสั่ง Edit/Copy Ctrl+C

เราใช้คำสั่ง Copy เพื่อการทำซ้ำข้อความของตัวอักษรในการนำไปวางเพื่อใช้ซ้ำข้อความเดิมได้สะดวกและรวดเร็วมากขึ้น

2.5 คำสั่ง Edit/Paste Ctrl+V

ใช้คำสั่ง Paste เพื่อแทนที่ส่วนที่เราได้เลือกไว้โดยการนำข้อมูลที่เราได้เลือกไว้ก่อนหน้าเข้ามาในโปรแกรม LINGO

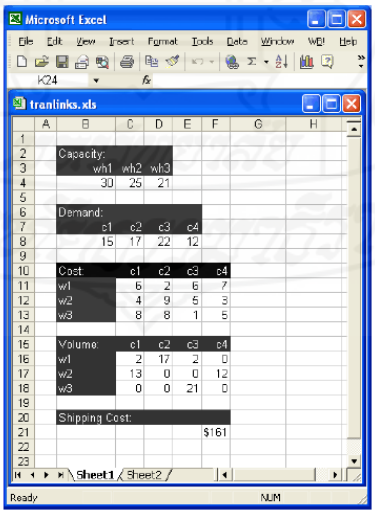
2.6 คำสั่ง Edit/Paste Special…

ใช้คำสั่ง Paste Special เพื่อแทนที่ส่วนที่เราได้เลือกไว้โดยการนำข้อมูลที่เราได้เลือกไว้ก่อนหน้าเข้ามาในโปรแกรม LINGO โดยการ Paste แบบ Special นี้สามารถที่จะทำงานได้มากกว่าการนำข้อมูลเข้ามาเป็นข้อความกล่าวคือสามารถนำเข้าวัตถุ การเชื่อมโยงไปสู่ส่วนอื่นดังตัวอย่างต่อไปนี้

สมมุติตัวอย่างของตัวแบบการขนส่งดังรายละเอียดด้านล่าง

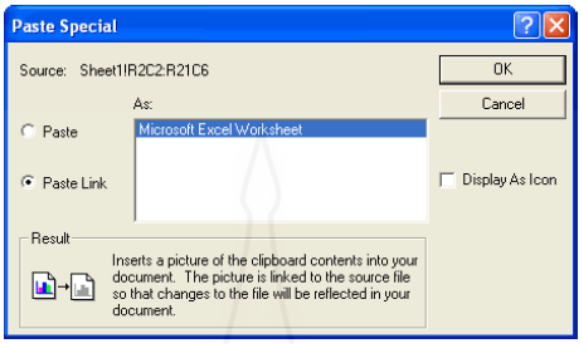
```
! A 3 Warehouse, 4 Customer
 Transportation Problem;
SETS:
  WAREHOUSE / WH1, WH2, WH3/: CAPACITY;
  CUSTOMER / C1, C2, C3, C4/: DEMAND;
  ROUTES(WAREHOUSE, CUSTOMER): COST, VOLUME;
ENDSETS
! The objective;
MIN = @SUM(ROUTES: COST * VOLUME);

! The demand constraints;
@FOR(CUSTOMER(J):
 @SUM(WAREHOUSE(I): VOLUME(I, J)) >=
  DEMAND(J));

! The supply constraints;
@FOR(WAREHOUSE(I): [SUP]
 @SUM(CUSTOMER(J): VOLUME(I, J)) <=
  CAPACITY(I));

! Here are the parameters;
DATA:
   CAPACITY = @OLE('D:\LNG\TRANLINKS.XLS');
   DEMAND = @OLE('D:\LNG\TRANLINKS.XLS');
   COST =  @OLE('D:\LNG\TRANLINKS.XLS');
   @OLE('D:\LNG\TRANLINKS.XLS') = VOLUME;
ENDDATA
```

เราต้องการนำเข้าข้อมูลจากไฟล์ Excel เข้าไปในตัวแบบนี้ทำได้โดยเปิดไฟล์ข้อมูลของ Excel และทำการ Load ข้อมูลจากไฟล์ Excel ตามตัวอย่างด้านล่าง



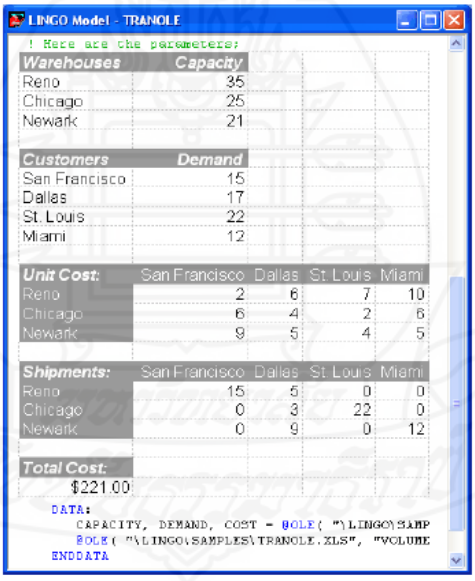
ภาพที่ 2.19 รูปแสดงการเปิดไฟล์ Excelเพื่อloadข้อมูล

เมื่อเราได้เลือกข้อความที่ต้องการแล้วจากนั้นไปที่เมนู Edit เลือกคำสั่ง Copy แล้วกลับมาสู่โปรแกรม LINGO เลือกคำสั่ง Paste Special ซึ่งจะเกิดกล่องข้อความดังรูป



ภาพที่ 2.20 รูปแสดงกล่องข้อความ Paste Spacial

สุดท้ายกดปุ่ม OK ข้อความที่เราเลือกในไฟล์ Excel ก็จะสามารถนำเข้ามาสู่โปรแกรม LINGOได้ดังรูป



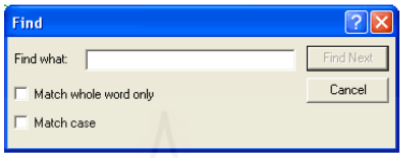
ภาพที่ 2.21 รูปแสดงโปรแกรมการload ข้อมูลจากไฟล์ Excel

2.7 คำสั่ง Edit/ Select All

เป็นคำสั่งเพื่อเลือกข้อความทั้งหมดเพื่อนำไปใช้ในส่วนของการต้องการทำซ้ำหรือลบทั้งหมด

2.8 คำสั่ง Edit/ Find… Ctrl+F

ใช้คำสั่ง Find เพื่อการค้นหาข้อความที่ต้องการในโปรแกรมเมื่อเราใช้คำสั่งนี้จะมีกล่องข้อความดังรูปปรากฏขึ้นให้ใส่ข้อความที่เราต้องการที่จะค้นหาซึ่งจะทำได้เฉพาะข้อความหรือตัวอักษร



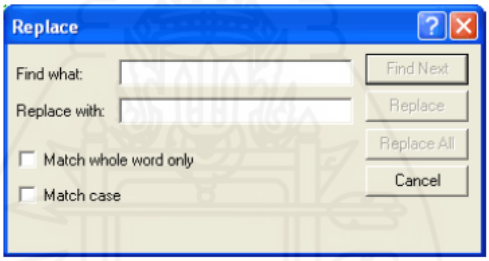
ภาพที่ 2.22 รูปแสดงกล่องข้อความคำสั่ง Edit/Find…

2.9 คำสั่ง Edit/ Find Next *Ctrl+N*

ใช้คำสั่ง Find Next เพื่อการค้นหาข้อความตัวต่อไปที่เราต้องการในโปรแกรม

2.10 คำสั่ง Edit/ Replace

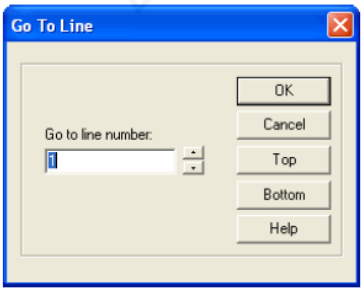
ใช้คำสั่ง Replace เพื่อแทนที่ข้อความลงในส่วนของโปรแกรมวินโดวส์ที่ดำเนินการใช้อยู่เมื่อใช้คำสั่งนี้จะมีกล่องข้อความปรากฏดังรูปด้านล่าง



ภาพที่ 2.23 รูปแสดงกล่องข้อความคำสั่ง Edit/Replace

2.11 คำสั่ง Edit/ Go to Line *Ctrl+T*

ใช้คำสั่ง Go to Line เพื่อกระโดดไปสู่บรรทัดที่เราต้องการในวินโดวส์ เมื่อใช้คำสั่งนี้จะมีกล่องข้อความปรากฏดังรูป ให้เลือกบรรทัดที่ต้องการไปแล้วกด OK



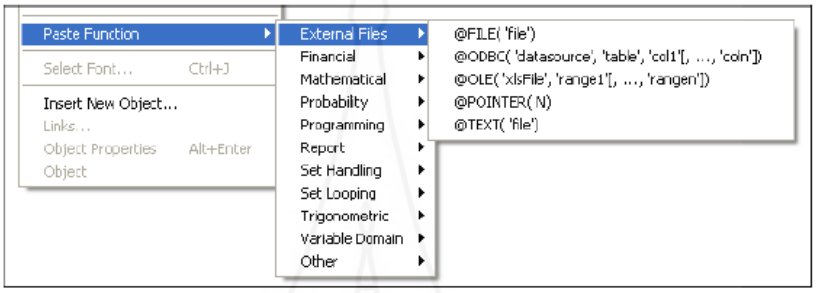
ภาพที่ 2.24 รูปแสดงกล่องข้อความคำสั่ง Edit/Go to Line

2.12 คำสั่ง Edit/ Match Parenthesis Ctrl+P

ใช้คำสั่งนี้เพื่อหาข้อความในวงเล็บที่ใกล้เพื่อเลือกใช้ข้อความในวงเล็บ

2.13 คำสั่ง Edit/ Paste Function

ใช้คำสั่งนี้เพื่อวางฟังค์ชั่นที่สร้างขึ้นโดยโปรแกรม LINGO ลงณ.จุดปัจจุบันที่ต้องการใส่เพิ่มในการใช้จะแสดงให้เห็นดังรูป



ภาพที่ 2.25 รูปแสดงคำสั่ง Edit/Paste Function

2.14 คำสั่ง Edit/ Select Font

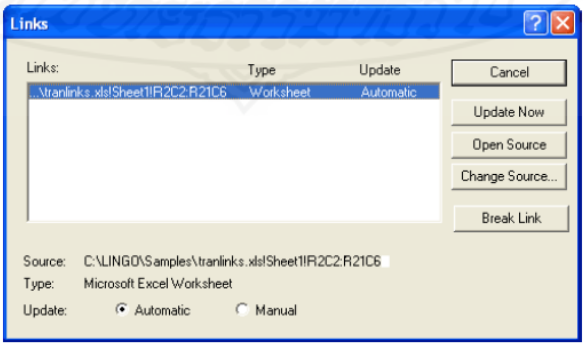
คำสั่งนี้เพื่อเลือกแบบอักษรใหม่,ขนาด,สี หรือรูปลักษณ์ที่ต้องการแสดงตัวอักษรเพื่อให้เราง่ายต่อการอ่านหรือค้นหา

2.15 คำสั่ง Edit/Insert New Object

ใช้คำสั่งนี้เพื่อนำเข้าข้อมูลของวัตถุหรือการเชื่อมโยงในตัวแบบของเรา

2.16 คำสั่ง Edit/Link

ใช้คำสั่งนี้เพื่อแก้ไขคุณสมบัติของการเชื่อมโยงสู่วัตถุภายนอกเมื่อใช้คำสั่งจะเกิดกล่องขอความดังรูป



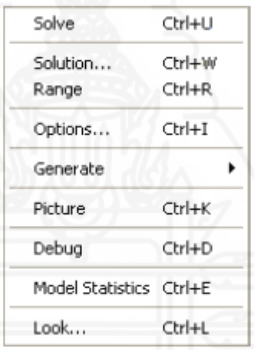
ภาพที่ 2.26 รูปแสดงกล่องข้อความคำสั่ง Edit/Link

2.17 คำสั่ง Edit/Object Properties *Alt+Enter*

ใช้คำสั่งนี้โดยการเลือกการเชื่อมโยงวัตถุในตัวแบบโดยการกดเลือกหนึ่งครั้งแล้วเราสามารถที่จะใช้คำสั่งนี้ในการแก้ไขคุณสมบัติของวัตถุซึ่งสิ่งที่สามารถแก้ไขประกอบไปด้วย

1. การแสดงวัตถุ
2. แหล่งที่มาของวัตถุ
3. ประเภทของการ Update อัตโนมัติหรือเลือกใช้
4. การเปิดการเชื่อมโยงของวัตถุ
5. การ Update วัตถุ และ
6. การแยกการเชื่อมโยงของวัตถุ

3. เมนู LINGO ในเมนูนี้ประกอบไปด้วยคำสั่งในหาคำตอบของตัวแบบ



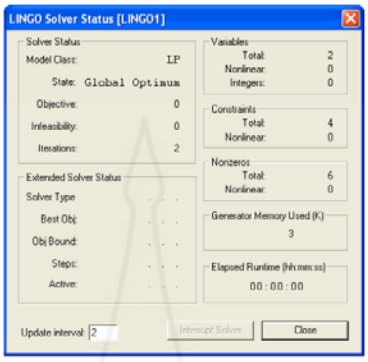
ภาพที่ 2.27 รูปแสดงรายละเอียดของคำสั่ง เมนู LINGO

3.1 คำสั่ง LINGO/Solve *Ctrl+U*

เป็นคำสั่งเพื่อให้โปรแกรมดำเนินการค้นหาคำตอบของตัวแบบ หากว่าขณะดำเนินการหาคำตอบโปรแกรมเกิดความผิดพลาดขึ้นก็จะมีกรอบข้อความเพื่อชี้แจงของสิ่งผิดปกติที่เกิดขึ้นดังรูป

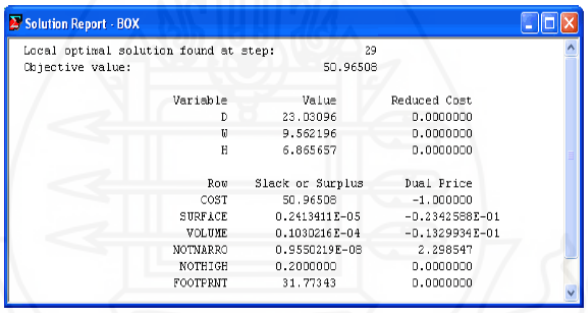
ภาพที่ 2.28 รูปแสดงรายละเอียดของกรอบข้อความผิดพลาด

และในขณะที่ดำเนินการทำงานโดยปราศจากความผิดพลาดและยังไม่พบคำตอบที่เหมาะสมการดำเนินการของแปรแกรมก็จะแสดงสถานะของการดำเนินงานอยู่ดังรูป



ภาพที่ 2.29 รูปแสดงสถานะการดำเนินงาน

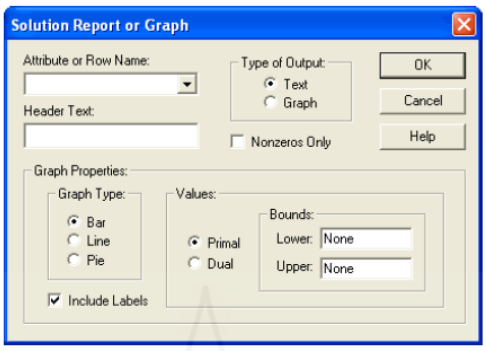
และสุดท้ายเมื่อโปรแกรมพบคำตอบที่เหมาะสมแล้วของตัวแบบก็จะแสดงผลออกมาให้ทราบดังรูปนี้



ภาพที่ 2.30 รูปแสดงสถานะการแสดงผลของคำตอบ

3.2 คำสั่ง LINGO/Solution Ctrl+W

ใช้คำสั่งนี้เพื่อการสร้างการรายงานผลเป็นตัวอักษรหรือรูปแบบกราฟฟิก เมื่อดำเนินการใช้คำสั่งนี้ก็จะมีกล่องข้อความแสดงออกมาดังรูป



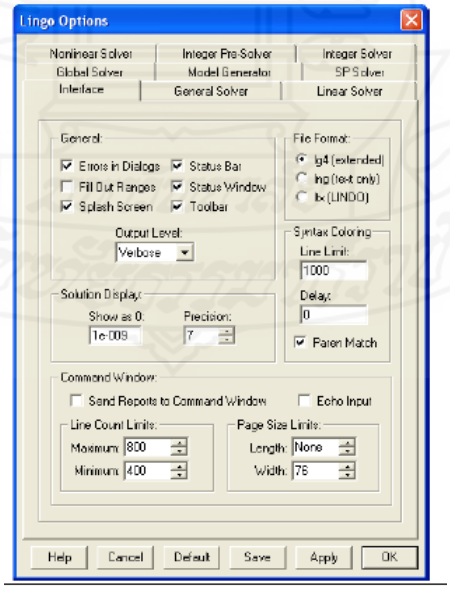
ภาพที่ 2.31 รูปแสดงหน้าต่างคำสั่ง LINGO/Solution

3.3 คำสั่ง LINGO/Range *Ctrl+R*

ใช้คำสั่งนี้เพื่อสร้างขอบเขตของการแสดงผลในตัวแบบซึ่งสามารถที่จะดำเนินการได้สองลักษณะคือ 1) เปลี่ยนสัมประสิทธิ์ในวัตถุประสงค์โดยไม่ทำให้เกิดการเปลี่ยนแปลงของค่าที่ดีที่สุดในการตัดสินใจ หรือ 2) เปลี่ยนแถวของข้อกำหนดโดยไม่ทำให้เกิดการเปลี่ยนแปลงของค่าที่ดีที่สุด/ dual price/ reduced cost

3.4 คำสั่ง LINGO/Options *Ctrl+I*

ใช้คำสั่งนี้เพื่อเปลี่ยนจำนวนของพารามิเตอร์ซึ่งมีผลต่อการใช้โปรแกรมซึ่งกล่องข้อความของการใช้คำสั่งนี้นั้นแสดงให้เห็นดังรูปด้านล่าง



ภาพที่ 2.32 รูปแสดงรายละเอียดของคำสั่ง LINGO/Option

บทที่ 3
วิธีดำเนินการวิจัย

ขั้นตอนในการดำเนินการจัดทำคู่มือการใช้โปรแกรม LINGO นั้นเริ่มต้นจากการที่ผู้จัดทำได้เริ่มต้นการศึกษาค้นคว้าจากการ Dowload โปรแกรม LINGO Version 12.0 พร้อมคู่มือต้นฉบับจากทาง Website ของ WWW.lindo.com ซึ่งเป็นฟรี License มาเพื่อใช้หลังจากนั้นก็ทำการ Install โปรแกรมทั้งหลายลงในคอมพิวเตอร์ให้เป็นที่เรียบร้อย แล้วจึงเริ่มดำเนินการศึกษาถึงคำสั่งเบื้องต้นต่างๆจากคู่มือต้นฉบับที่เป็นภาษาต่างประเทศพร้อมทั้งวิธีการใช้ที่มีรายระเอียดแสดงอยู่ เมื่อผู้ศึกษาได้ทำการศึกษาเป็นที่เข้าใจถึงขั้นตอนการใช้งานก็เริ่มทดลองฝึกหัดเขียนโปรแกรมตามตัวแบบที่มีอยู่ในคู่มือต้นฉบับเพื่อทำความเข้าใจถึงขั้นตอนและวิธีการเขียนโปรแกรม แนวคิดการใช้งานต่างๆไปพร้อมกันพร้อมทั้งทดลอง Runโปรแกรมที่ได้ทำการสร้างขึ้นมาตามตัวแบบว่าสามารถแสดงผลออกมาได้ตามตัวอย่างที่มีอยู่หรือไม่ เมื่อสามารถที่จะดำเนินการได้ตามต้องการแล้ว ขั้นตอนต่อไปก็ศึกษาถึงการแปลความหมายของผลลัพธ์และรายละเอียดที่เกิดขึ้นที่ได้แสดงผลออกมาของโปรแกรม เมื่อดำเนินการศึกษาค้นคว้าเนื้อหาได้จนเป็นที่น่าพอใจและเพียงพอแล้วก็นำเนื้อหาที่ได้ทำการศึกษามาเริ่มจัดพิมพ์เป็นคู่มือที่เป็นภาษาไทยเมื่อคู่มือแล้วเสร็จก็ดำเนินการนำคู่มือที่ได้ให้กลุ่มตัวอย่างทดลองศึกษาเนื้อหาในคู่มือดูหลังจากนั้นให้ประเมินผลคู่มือตามใบประเมินที่ได้จัดทำไว้ และท้ายที่สุดนำผลของการประเมินที่ได้รับมาสรุปในแต่ละหัวข้อรายละเอียดเพื่อนำไปพิจารณาประเมินเนื้อหาและรายละเอียดของคู่มือต่อไป

1. ประชากรและกลุ่มตัวอย่าง

ประชากรที่ให้ทำการประเมินคู่มือ ได้แก่ สมาชิกของแผนกวิศวกรรม บริษัท วายเอส ภัณฑ์ จำกัดซึ่งมีทั้งสิ้น 12 คน

2. เครื่องมือที่ใช้ในการวิจัย

ในการจัดทำคู่มือสามารถแบ่งข้อมูลที่ใช้เป็นเครื่องมือในการวิจัยออกเป็นสองส่วนด้วยกันคือ ส่วนที่ 1 เป็นข้อมูลทุติยภูมิที่ได้จากเนื้อหาคู่มือต้นฉบับภาษาต่างประเทศของโปรแกรมเพื่อใช้ในการศึกษาและนำมาจัดทำเป็นเนื้อหาในคู่มือภาษาไทย ส่วนที่ 2 เป็นข้อมูลปฐมภูมิที่ได้จากการให้กลุ่มตัวอย่างประเมินผลของคู่มือที่ได้จัดทำขึ้น

3. การเก็บรวบรวมข้อมูล

ผู้จัดทำได้ดำเนินการรวบรวมข้อมูลและจัดเตรียมเนื้อหาเพื่อที่จะได้จัดทำเป็นคู่มือโดยใช้ระยะเวลาตั้งแต่วันที่ 18มิถุนายน2554จนกระทั่ง30ตุลาคม 2554 และ ให้กลุ่มตัวอย่างประเมินคู่มือจนแล้วเสร็จ 2 ธันวาคม 2554

บทที่ 4
ผลการศึกษา

จากเนื้อหาของคู่มือในการใช้งานโปรแกรม LINGO นั้นเริ่มต้นด้วยผู้ใช้งานต้องรู้จักและเข้าใจถึงส่วนประกอบสำคัญๆในเนื้อหาของคู่มือเพื่อนำมาประกอบการใช้งานโดยคู่มือแบ่งออกเป็นส่วนๆดังนี้1.พื้นฐานทางทฤษฎีที่เกี่ยวข้องกับตัวแบบทางคณิตศาสตร์2.คำสั่งของโปรแกรม 3.ขั้นตอนการสร้างตัวแบบของปัญหา 4. เนื้อหาของตัวแบบในคู่มือซึ่งประกอบด้วย

4.1 ตัวแบบปัญหาเรื่องส่วนประสมของผลิตภัณฑ์

4.2 ตัวแบบปัญหาเรื่อง การขนส่ง

4.3 ตัวแบบปัญหาเรื่อง การวางแผนการทำงานของพนักงาน

4.4 ตัวแบบปัญหาเรื่อง พิกัดน้ำหนักส่วนผสมของเมล็ดพืช

4.5 ตัวแบบปัญหาเรื่อง การกำหนดกำลังคนโดยแสดงออกเป็นรายการ

4.6 ตัวแบบปัญหาเรื่องจำนวนเต็มเลขฐานสอง

4.7 ตัวแบบปัญหาเรื่องจำนวนเต็มเลขฐานสองกรณีส่วนผสมผลิตภัณฑ์

4.8 ตัวแบบปัญหาเรื่องตัวแปรอิสระของการพยากรณ์

4.9 ตัวแบบปัญหาในการนำมาประยุกต์ใช้งานจริงของแผนกวิศวกรรม บริษัท วาย เอส ภัณฑ์

1. พื้นฐานทางทฤษฎีที่เกี่ยวข้องกับตัวแบบทางคณิตศาสตร์

ซึ่งประกอบไปด้วยรูปแบบสี่ประเภทด้วยกันคือ 1.1) การแก้ปัญหาทางตรง 1.2) การแก้ปัญหาเชิงเส้น 1.3) การแก้ปัญหาที่ไม่เป็นเชิงเส้น 1.4) การแก้ปัญหาแบบอัลกอริทึม ซึ่งส่วนสำคัญหลักๆโดยแท้จริงแล้วโปรแกรมที่ใช้มากที่สุดคือรูปแบบการแก้ปัญหาที่เป็นเชิงเส้นและไม่เป็นเชิงเส้น โดยคำตอบที่ได้จากรูปแบบที่เป็นการแก้ปัญหาเชิงเส้นนั้นจะให้คำตอบที่รวดเร็วและสะดวกกว่าแบบที่ไม่เป็นเชิงเส้นพร้อมกันนี้คำตอบที่ได้สามารถหาคำตอบออกมาเป็นค่าที่ต้องการในจุดที่คุ้มค่าที่สุด เช่น กำไรสูงสุด,ต้นทุนต่ำสุด เป็นต้น ส่วนตัวแบบที่ไม่เป็นเชิงเส้นนั้นจะให้คำตอบในอีกรูปแบบที่แตกต่างออกไปคือจะได้คำตอบออกมาในสองลักษณะเป็นผลลัพธ์ที่ดีที่สุดจากทั้งหมด(Global Optimum) และหรือ เป็นผลลัพธ์ที่ดีที่สุดเฉพาะที่ (Local Optimal) เพราะว่าในช่วงๆหนึ่งของข้อมูลที่เป็นตัวแบบที่ไม่เป็นเชิงเส้นนั้นอาจเกิดจุดที่ดีที่สุดขึ้นหลายค่า

ดังนั้นหากตัวแบบแสดงผลออกมาว่าเป็นรูปแบบที่มิใช่เชิงเส้นแล้วนั้นเราสามารถที่จะนำมาประยุกต์ใช้ให้เกิดประโยชน์เพื่อการพิจารณาได้โดยการขยายช่วงของข้อมูลในตัวแบบออกไปให้เห็นสภาวะและสภาพของคำตอบที่เปลี่ยนแปลงออกไปจากการขยายข้อมูลก็จะได้ประโยชน์เพิ่มขึ้นจากตัวแบบนั้นๆ

2. คำสั่งของโปรแกรม

ในส่วนคำสั่งที่ใช้งานของโปรแกรมได้ถูกพัฒนาขึ้นด้วยกันสองส่วนคือโปรแกรมที่เป็นพื้นฐานคำสั่งของ LINGO เองและอีกส่วนเป็นคำสั่งใช้ร่วมกับโปรแกรมของ WINDOW ในรูปแบบของตัวคำสั่งที่ใช้กับ WINDOW นั้นโดยทั่วไปก็จะคุ้นเคยเป็นอย่างดีแต่ก็มีคำสั่งเพิ่มขึ้นในหลายส่วน

3. ขั้นตอนการสร้างตัวแบบของปัญหา

การสร้างตัวแบบของปัญหาในโปรแกรมนั้นผู้ใช้งานก็ควรที่จะมีความรู้ในเรื่องของเนื้อหาทางทฤษฎีในรายวิชาเรื่องการวิเคราะห์เชิงปริมาณเป็นพื้นฐานมาบ้างพอสมควรเพื่อที่จะได้พิจารณากำหนด และ สร้างตัวแบบ ให้กับโปรแกรมได้ด้วยตนเอง ก่อนที่จะทำการสร้างตัวแบบเราต้องมีข้อมูลและรายละเอียดเพียงพอที่จะนำมาสังเคราะห์เพื่อสร้างตัวแบบในการใช้งานโดยเราต้องทราบถึงจุดประสงค์และกำหนดถึงจุดมุ่งหมายของคำตอบที่เราต้องการทราบอย่างชัดเจน หลังจากนั้นจึงเริ่มสร้างตัวแบบและกำหนดรายละเอียดให้กับตัวแบบโดยการสร้างตัวแบบโปรแกรมจะมีขั้นตอนหลักๆสำคัญดังนี้คือ 1.การจัดตั้งปัญหา2.การสร้างตัวแบบของปัญหา 3. การป้อนตัวแบบให้แก่โปรแกรม 4. การสั่งให้โปรแกรมดำเนินการหาคำตอบ 5. การวิเคราะห์ผลลัพธ์ที่ได้ 6.การนำผลลัพธ์ไปใช้งานซึ่งในตัวแบบของปัญหานั้นๆจะประกอบไปด้วยส่วนที่สำคัญ ที่จะต้องกำหนดให้แก่ตัวแบบอยู่สามส่วนด้วยกันคือ

1. ฟังค์ชั่นวัตถุประสงค์ (Objective Function) เป็นส่วนที่แสดงความหมายที่แท้จริงของตัวแบบที่เราต้องการคำตอบ เช่น กำไรสูงสุด,ต้นทุนที่ต่ำสุด เป็นต้น

2. ตัวแปร (Variable) เป็นจำนวนของตัวแปรที่เราควบคุมเพื่อที่จะให้เกิดคำตอบในฟังค์ชั่นวัตถุประสงค์

3. ข้อกำหนด (Constraints) เป็นขอบเขตและข้อจำกัดที่เกิดขึ้นแก่ตัวแปรซึ่งเราก็จะต้องระบุและให้รายละเอียดเกี่ยวกับข้อกำหนดเพื่อการหาคำตอบในตัวแบบ

รูปแบบของปัญหาที่เรานำมาแก้ไขนั้นมีอยู่หลายรูปแบบทั้งนี้ขึ้นอยู่กับจุดมุ่งหมายและความต้องการหาคำตอบของแต่ปัญหาซึ่งจะทำให้การสร้างตัวแบบมีรายละเอียดปลีกย่อยต่างๆกันออกไป ตัวแบบบางอย่างอาจใกล้เคียงหรือแตกต่างเป็นอย่างมากจากตัวแบบตัวอย่างที่ได้นำเสนอไว้ดังนั้นเราต้องฝึกฝนการใช้โปรแกรมเพื่อให้เกิดความชำนาญและต้องรู้จักทำการประยุกต์พลิกแพลงการใช้โปรแกรมให้ขอบเขตกว้างออกไปจากตัวแบบตัวอย่างที่มีอยู่ ซึ่งตัวอย่างนั้นมีจุดมุ่งหมายแค่เพียงเพื่อเป็นตัวแบบที่ใช้เทียบเคียงในการเริ่มฝึกหัดและเป็นแนวคิดในการสร้างตัวแบบเท่านั้น เมื่อเราได้เรียนรู้ข้อมูลพื้นฐานในการใช้งานโปรแกรมและได้ดำเนินการทดลองสร้างตัวแบบของปัญหาด้วยตนเองได้แล้วก็จะทำให้การใช้งานโปแกรม LINGO ได้เสริมสร้างความเข้าใจและสามารถที่จะใช้ทำงานได้อย่างสะดวกและรวดเร็วมากยิ่งขึ้น

4. เนื้อหาที่ได้จากการศึกษาข้อมูล

ส่วนที่1จากเนื้อหาของคู่มือต้นฉบับภาษาต่างประเทศที่เป็นข้อมูลทุติยภูมิซึ่งผู้ศึกษาได้ทำการศึกษาดังที่กล่าวมาแล้วข้างต้นผู้ศึกษาสามารถที่จะรวบรวมเนื้อหาข้อมูลและนำเสนอออกมาเป็นรายละเอียดของคู่มือภาษาไทยได้ดังต่อไปนี้

5. การเริ่มต้นใช้โปรแกรม LINGO

(LINDO System Inc. 2010:1-99) โปรแกรม LINGO คืออะไร? ในเบื้องต้นเรามาเริ่มทำความรู้จักกับโปรแกรม LINGO กันก่อนโปรแกรมLINGO เป็นโปรแกรมที่เป็นเครื่องมือสำหรับใช้ประโยชน์ของการแก้ปัญหาเชิงเส้นและมิใช่เชิงเส้นเพื่อหาคำตอบที่ดีที่สุด เช่นจุดที่ประหยัดที่สุด กำไรสูงสุด ผลตอบแทนสูงสุด หรือ ราคาต่ำสุด เป็นต้น ปัญหาโดยทั่วไปเกิดจากการที่ต้องการใช้ทรัพยากรที่มีให้เกิดประโยชน์สูงสุดซึ่งหมายรวมถึง เงิน,เวลา,เครื่องจักร,คน, สินค้าคงคลัง และ อื่นๆ

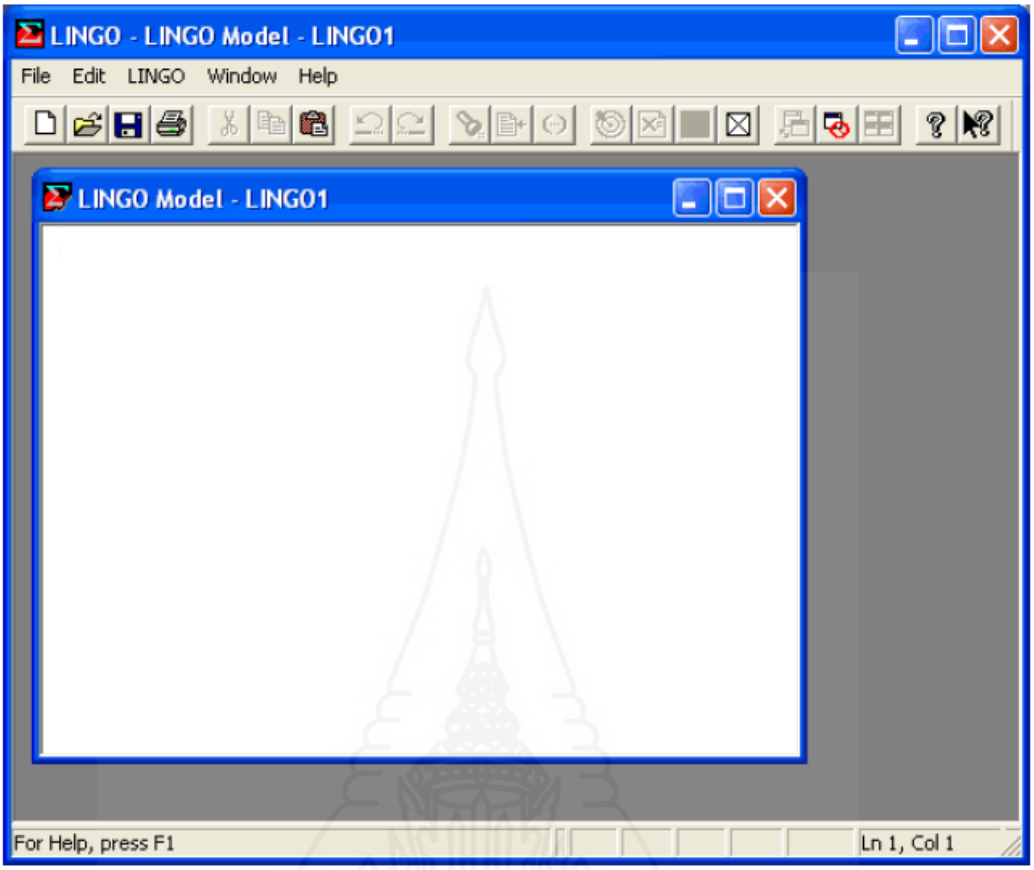
การติดตั้งโปรแกรม LINGO ในการใช้โปรแกรมและติดตั้งโปรแกรมLINGOสำหรับการทำงานในระบบวินโดวส์ให้ใส่แผ่นโปรแกรมและดำเนินการติดตั้งโปรแกรมโดยการเลือก SETUP ที่อยู่ในFOLDER ของLINGO บางโปรแกรมของ LINGO จะมาพร้อมการติดตั้งที่มีการอนุญาติ (License)ไว้แล้ว ถ้าหากไม่มีหรือโปรแกรมต้องการให้ระบุ (License Key) ก็จะมีบล็อกดังตัวอย่างเกิดขึ้น

ภาพที่ 4.1 รูปแสดงหน้าต่าง LINGO License Key

(License Key) ของคุณจะได้รับในจดหมายอิเล็กทรอนิกส์เมื่อคุณทำการสั่งโปรแกรม โดยจะประกอบด้วยตัวหนังสือสัญลักษณ์และหมายเลขแบ่งออกเป็นสี่ส่วนดัง

ตัวอย่าง (e.g., r82m-XCW2-dZu?-%72S-fD?S-Wp@).

เมื่อทำการใส่ทั้งหมดลงในโปรแกรมแล้วกดปุ่ม OK โปรแกรมก็จะผ่านขั้นตอนการติดตั้งเข้าสู่ระบบและสามารถเริ่มใช้งานโปรแกรมต่อไปการใส่ตัวแบบลงในวินโดวส์ เมื่อทำการเริ่มต้นโปรแกรมLINGOหน้าต่างของโปรแกรมWINDOWจะมีลักษณะดังรูปภาพ



ภาพที่ 4.2 รูปแสดงหน้าต่างโปรแกรม WINDOW

โดยกรอบด้านบนของหน้าต่างของโปรแกรมจะประกอบไปด้วยเมนู และ คำสั่งต่างๆ ดูในบทที่2ประกอบ เรื่องคำสั่งของโปรแกรมที่ใช้ใน WINDOW

6. การพัฒนาและสร้างตัวแบบของโปรแกรม LINGO ในวินโดวส์ (WINDOWS)

ตัวอย่างของตัวแบบปัญหา

ตัวอย่างที่ 1 เรื่องส่วนประสมผลิตภัณฑ์ (PRODUCT-MIX SAMPLE)

เราจะสร้างตัวอย่างของตัวแบบปัญหาในเรื่องส่วนประสมผลิตภัณฑ์ (PRODUCT-MIX SAMPLE) อย่างง่ายๆ โดยการสมมุติว่าบริษัทหนึ่งชื่อ ComputerQuick มีการผลิตสินค้าเป็น Computer สองชนิดด้วยกันคือรุ่น Standardและรุ่น Turbo เมื่อบริษัทสามารถขายทุกๆหน่วยของสินค้าที่เป็น Standard ก็จะได้กำไร100เหรียญ และ หากการขายผลิตภัณฑ์ประเภท Turbo จะได้กำไร150เหรียญ โดยการผลิตในโรงงานของComputerQuickนั้นสามารถที่จะผลิต Computer แบบ Standard ได้มากที่สุด 100 เครื่องต่อวันและในขณะเดียวกันนั้นไลน์การผลิตเครื่องที่เป็น Turbo

สามารถทำการผลิตได้120เครื่องต่อวัน มิใช่มีข้อมูลแต่เพียงเท่านั้นComputerQuick มีข้อจำกัดเรื่องชั่วโมงการผลิตของคนงานที่ผลิตในหนึ่งวันโดยในรายละเอียดของบริษัทมีเวลาของคนงานโดยรวม160ชม. โดยการผลิตคอมพิวเตอร์แบบ Standard ต่อเครื่องต้องการปริมาณของชั่วโมงการผลิตเท่ากับ 1ชม.ของคนงาน และ ในการผลิตแบบ Turbo ต้องการ 2ชม.ของคนงานความต้องการแก้ปัญหาของComputer Quick ก็คือจะทำการผลิตคอมพิวเตอร์ทั้งสองชนิดอย่างไรในแต่ละวันโดยทำให้เกิดกำไรสูงสุดภายใต้เงื่อนไขดังกล่าวโดยทั่วไป ตัวแบบที่หาจุดคุ้มค่าสูงมากที่สุดจะประกอบด้วยสามส่วนด้วยกัน คือ

1. Objective Function หรือฟังค์ชั่นวัตถุประสงค์ เป็นส่วนที่แสดงความหมายที่แท้จริงว่าค่าอะไรในตัวแบบที่ต้องการหาที่ดีที่สุด ในตัวแบบทางธุรกิจแน่นอนที่ฟังค์ชั่นที่ต้องการทราบและหาค่าสูงสุดก็คือกำไร หรือ ในทางกลับกันทางด้านต้นทุนเราต้องการหาค่าต้นทุนที่ต่ำสุด ในกรณีตัวอย่างของComputerQuick ฟังค์ชั่นวัตถุประสงค์จะต้องการคำนวนกำไรของบริษัทที่ได้จากผลผลิตที่เกิดจากการผลิตคอมพิวเตอร์สองรุ่นคือ Standard และTurbo

2. Variable หรือตัวแปร เป็นจำนวนของตัวแปรที่ควบคุม เราต้องกำหนดว่าอะไรเป็นค่าที่ดีที่สุดของแต่ละตัวแปรด้วยเหตุผลนี้บางครั้งเราจะเรียกว่า Decision Variable (ตัวแปรในการตัดสินใจ) เป้าหมายก็เพื่อหาค่าที่ดีสุด หรือ ก็คือการการหาค่าของตัวแปรในตัวแบบที่ทำให้เกิดคำตอบในฟังค์ชั่นวัตถุประสงค์ ในตัวอย่างที่กล่าวมาข้างต้นของบริษัทคอมพิวเตอร์ก็จะมีอยู่สองตัวแปรคือจำนวนของเครื่อง Standard ที่ทำการผลิต และ อีกตัวแปรหนึ่งคือจำนวนที่ต้องผลิตของเครื่อง Turbo

3. Constraints หรือข้อกำหนด เกือบทั้งหมดนอกเหนือจากที่จะได้รับการยกเว้นตัวแปรจะมีข้อจำกัดในของตัวมันเอง ตัวแปรในตัวแบบสามารถที่จะสมมุติขึ้น และแหล่งที่มาของตัวแปรที่สมมุติขึ้นนั้นก็จะมีข้อจำกัดของตัวมันเองเช่น(เวลา,วัตถุดิบ, เงินทุน) ข้อจำกัดทั้งหลายเหล่านี้ก็จะต้องทำให้สามารถแปรเป็นสูตรเพื่อที่ใช้เป็นตัวแปรของฟังค์ชั่นตัวแบบ สูตรเหล่านี้ต้องอ้างอิงถึงข้อกำหนดและข้อจำกัดต่างๆของตัวแบบในตัวอย่างของComputerQuick นั้ เราก็จะมีข้อกำหนดหนึ่งอย่างก็คือจำนวนชั่วโมงของคนงานที่ใช้ในทุกไลน์การผลิต

การใส่ค่าลงในตัวแบบเราจะเริ่มสร้างโครงสร้างของฟังค์ชั่นวัตถุประสงค์ดังตัวอย่างที่ได้กล่าวมาแล้ว โดยฟังค์ชั่นของวัตถุประสงค์ประกอบไปด้วยกำไรสูงสุดที่ได้จากผลรวมของการผลิตคอมพิวเตอร์รุ่น Standard และ Turbo โดยนำกำไรของแต่ละเครื่องคูณเข้ากับเครื่องที่ทำจะการผลิต รูปแบบการเขียนลงในโปรแกรมสามารถเขียนได้ดังนี้

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

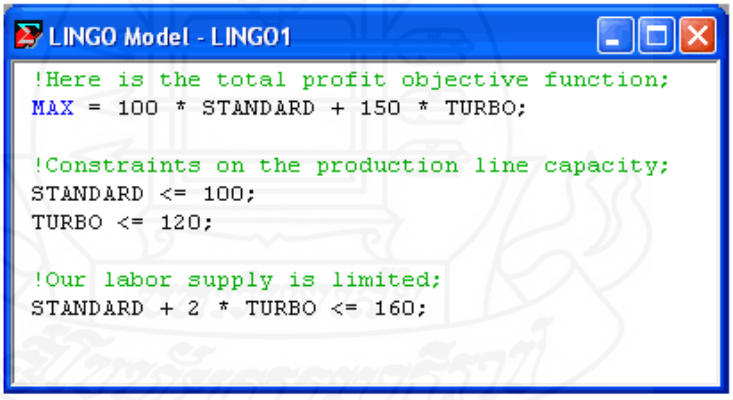
ทุกตัวแบบทางคณิตศาสตร์ที่แสดงในโปรแกรม LINGO ลงท้ายด้วยเครื่องหมายอัฒภาคและตัวแบบจะไม่ทำการแก้ปัญหาหากไม่มีเครื่องหมายนี้ ต่อไปเราต้องใส่ค่าของข้อกำหนดเรื่องความสามารถของไลน์และการใช้คนงานสามารถเขียนเป็นฟังค์ชั่นของข้อกำหนดได้ดังนี้

```
STANDARD <= 100;
TURBO <= 120;
```

ในฟังค์ชั่นข้อกำหนดแสดงความหมายว่าในการผลิตคอมพิวเตอร์รุ่น Standard ในหนึ่งวันต้องน้อยกว่าหรือเท่ากับกำลังการผลิตของไลน์คือ100 และส่วนของไลน์การผลิตรุ่นTurboคือ 120 การใช้สัญลักษณ์ในคอมพิวเตอร์โปรแกรมLINGOสามารถที่จะใช้สัญลักษณ์ <= หรือ < แทนความหมายน้อยกว่าเท่ากับ ($\leq$)ได้ส่วนในทางตรงข้ามกันสัญลักษณ์ >= หรือ > แทนความหมายมากกว่าเท่ากับ ($\geq$) และในข้อกำหนดสุดท้ายสามารถเขียนได้ดังนี้

```
STANDARD + 2 * TURBO <= 160;
```

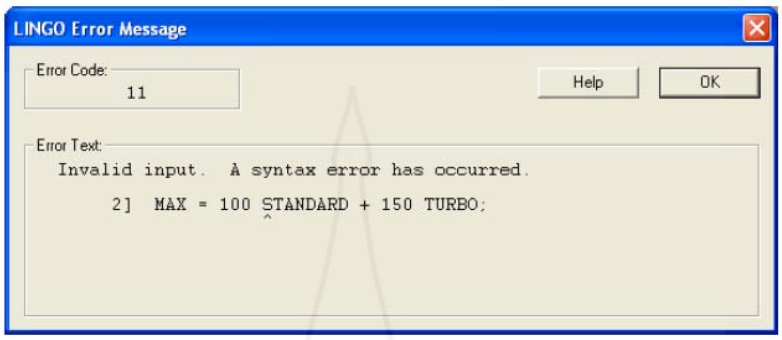
สุดท้ายเมื่อนำไปเขียนในโปรแกรมเต็มรูปแบบก็จะได้ดังนี้



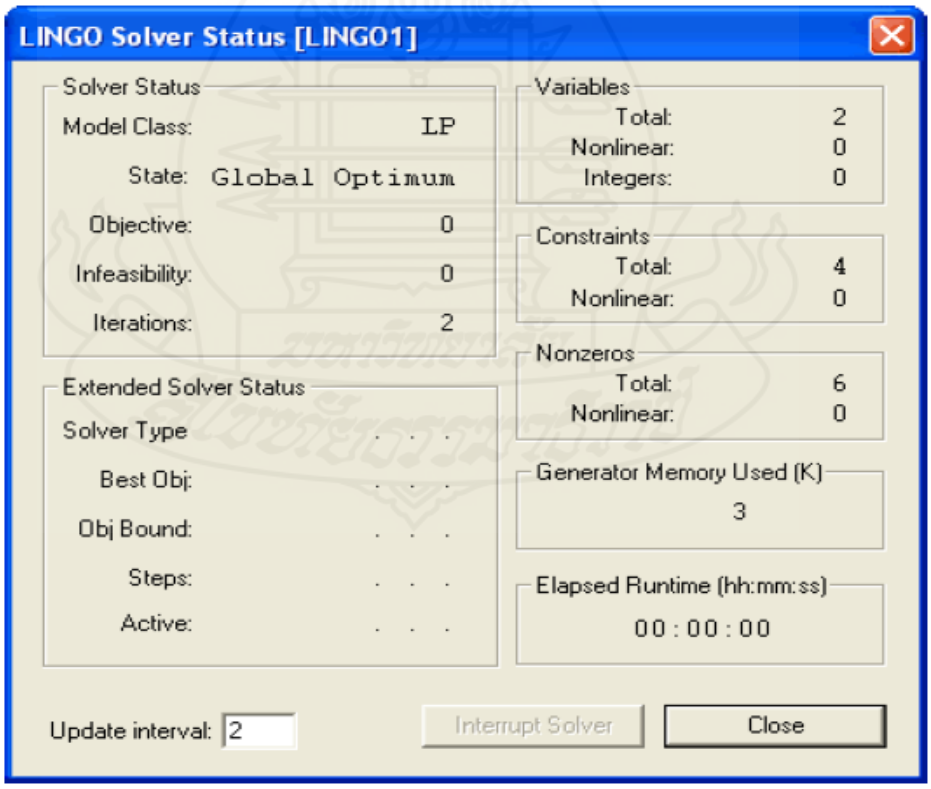
เราต้องจำไว้ให้ดีว่าโปรแกรม LINGO ไม่สามารถที่จะเข้าใจตัวอักษรพิมพ์เล็กและใหญ่ได้การใช้ตัวอักษรที่ผสมในการกำหนดเป็นตัวแปรควรจะระมัดระวังในการใช้เป็นพิเศษ

การแก้ปัญหาของโปรแกรม เมื่อเราได้ทำการสร้างรูปแบบของโปรแกรมเสร็จสิ้นเรียบร้อยแล้วในการสั่งให้โปรแกรมทำการแก้ปัญหานั้นทำได้โดยเลือกไปที่คำสั่ง Solve ในเมนูของโปรแกรม LINGO หรือกดปุ่ม Solve ที่ Tool bar ของด้านบนกรอบของโปรแกรม ตัวโปรแกรมก็จะเริ่มทำการประมวลผล หากการประมวลผลสำเร็จเสร็จสิ้นลงโปรแกรมก็จะ

แสดงผลค่าที่คำนวนได้ออกมา แต่ถ้าหากว่าโปรแกรมเกิดความผิดพลาดเกิดขึ้นเนื่องจากสาเหตุใดๆ ก็ตามตัวโปรแกรมก็จะแสดงข้อความขึ้นเพื่อแสดงให้ทราบและบ่งบอกถึงสิ่งผิดปกติที่เกิดขึ้นดังรายละเอียดด้านล่าง



รูปแบบที่เกิดขึ้นของความผิดพลาดสามารถดูได้ที่รายละเอียดของข้อความแสดงความผิดปกติ (Error Messages) หากว่าไม่มีการแสดงข้อผิดพลาดใดๆเกิดขึ้นเนื่องจากการกำหนดโครงสร้างฟังค์ชั่นต่างในโปรแกรม เมื่อการคำนวนการแก้ไขปัญหาได้เริ่มขึ้นโปรแกรมจะแสดงสถานะการดำเนินงานของโปรแกรมดังรูป



ภาพที่ 4.3 รูปแสดงสถานะการดำเนินงานของโปรแกรม

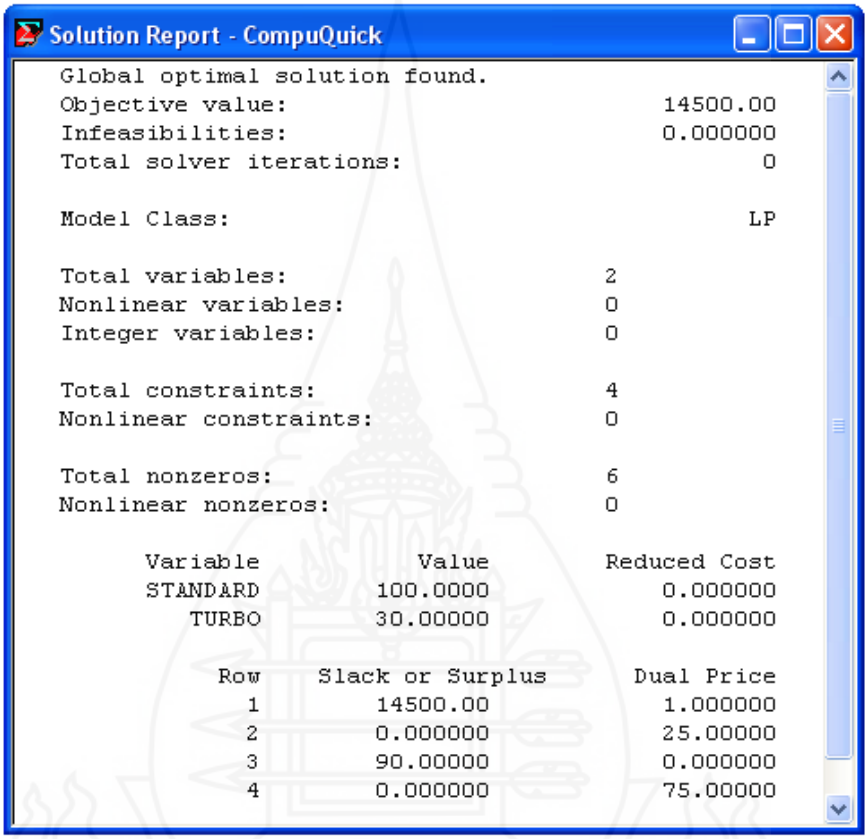
ในการแสดงผลของการคำนวนใน Solver Status ได้จัดให้มีปุ่มที่จะขัดขวางการทำงาน Interrupt Solver botton. จะใช้ปุ่มนี้ในกรณีที่โปรแกรมเกิดการลังเลที่จะให้ดำเนินการซ้ำอีกครั้งซึ่งจะเกิดได้บ่อยครั้งในกรณีที่โปรแกรมจะพบคำตอบที่ดีที่สุดได้นั้นต้องคำนวนหลายรอบและคำตอบยังอยู่ห่างไกลออกไปมากเนื่องจากยังไม่สามารถคำนวนหาค่าที่เหมาะสมกับตัวแบบที่กำหนดได้ตัวอย่างเช่นโปรแกรมเชิงเส้นที่ตัวแปรไม่เป็นจำนวนเต็ม หากว่าโปรแกรมเชิงเส้นได้รับการสั่งให้ขัดขวางเนื่องจากเลือกปุ่ม Interrupt Solver botton. โปรแกรมก็จะดำเนินการโดยไม่ใส่ใจในส่วนที่ไม่มีความหมายเพื่อประโยชน์ในการคำนวนที่ต้องการความรวดเร็วเพิ่มขึ้น Variable box จะแสดงจำนวนของตัวแปรในตัวแบบ ทั้งที่ไม่เป็นแบบเชิงเส้นก็จะแสดงด้วยConstraints Box จะแสดงจำนวนของข้อกำหนดในตัวแบบ และ จำนวนของข้อกำหนดที่ไม่เป็นในแบบเชิงเส้น Non-zeroes box จะแสดงจำนวนสัมประสิทธ์ที่ไม่เป็นศูนย์ในตัวแบบซึ่งจะเกิดขึ้นกับตัวแปรที่ไม่เป็นในแบบเชิงเส้นGenerator Memory used box เป็นการแสดงหน่วยความจำที่ใช้ในการดำเนินงาน Elapsed Runtime box จะแสดงเวลาขึ้นหากการคำนวนนั้นคำตอบที่จะได้อยู่ห่างไกลออกไปและยังต้องใช้เวลาในการคำนวนอยู่ Solver Status box จะแสดงสถานะปัจจุบันในการแก้ไขปัญหาซึ่งจะแสดงผลดังตาราง

Field	Description
Model Class	Displays the model's classification. Possible classes are "LP", "QP", "MILP", "MIQP", "PILP", "PIQP", "NLP", "SOCP", "MISOCP", "PISOCP", "MINLP", and "PINLP".
State	Gives the Status of the current solution. Possible states are "Global Optimum", "Local Optimum", "Feasible", "Infeasible", "Unbounded", "Interrupted", and "Undetermined".
Objective	Current value of the objective function.
Infeasibility	Amount constraints are violated by.
Iterations	Number of solver iterations.

คุณสมบัติในตัวแบบแสดงความหมายและรายละเอียดดังตารางด้านล่าง

Abbreviation	Class	Description
LP	Linear Program	All expressions are linear and the model contains no integer restrictions on the variables.
QP	Quadratic Program	All expressions are linear or quadratic, the model is convex, and there are no integer restrictions.
MILP	Mixed Integer Linear Program	All expressions are linear, and a subset of the variables is restricted to integer values.
MIQP	Mixed Integer Quadratic Program	All expressions are either linear or quadratic, the model is convex, and a subset of the variables has integer restrictions.
PILP	Pure Integer Linear Program	All expressions are linear, and all variables are restricted to integer values.
PIQP	Pure Integer Quadratic Program	All expressions are linear or quadratic, the model is convex, and all variables are restricted to integer values.
SOCP	Second-Order Cone Program	The model is a second-order cone program and all variables are continuous.
MISOCP	Mixed Integer Second Order Cone Program	The model is a second-order cone program, and a subset of the variables is restricted to integer values.
PISOCP	Pure Integer Second-Order Cone Program	The model is a second-order cone program, and all the variables are restricted to integer values.
NLP	Nonlinear Program	At least one of the relationships in the model is nonlinear with respect to the variables.
MINLP	Integer Nonlinear Program	At least one of the expressions in the model is nonlinear, and a subset of the variables has integer restrictions. ***In general, this class of model will be very difficult to solve for all but the smallest cases.***
PINLP	Pure Integer Nonlinear Program	At least one of the expressions in the model is nonlinear, and all variables have integer restrictions. ***In general, this class of model will be very difficult to solve for all but the smallest cases.***

เมื่อโปรแกรมดำเนินการคำนวนเพื่อหาคำตอบของตัวแบบที่สร้างขึ้นมาโดยการหาคำตอบของตัวแปรและได้ค่าที่เหมาะสมกับข้อกำหนดที่กำหนดไว้แล้วนั้นหากไม่เกิดข้อผิดพลาดประการใดเกิดขึ้นโปรแกรมก็จะแสดงผลของการคำนวนที่เกิดขึ้นของตัวแบบให้เห็นดังรายละเอียดด้านล่าง



การแปลความหมาย

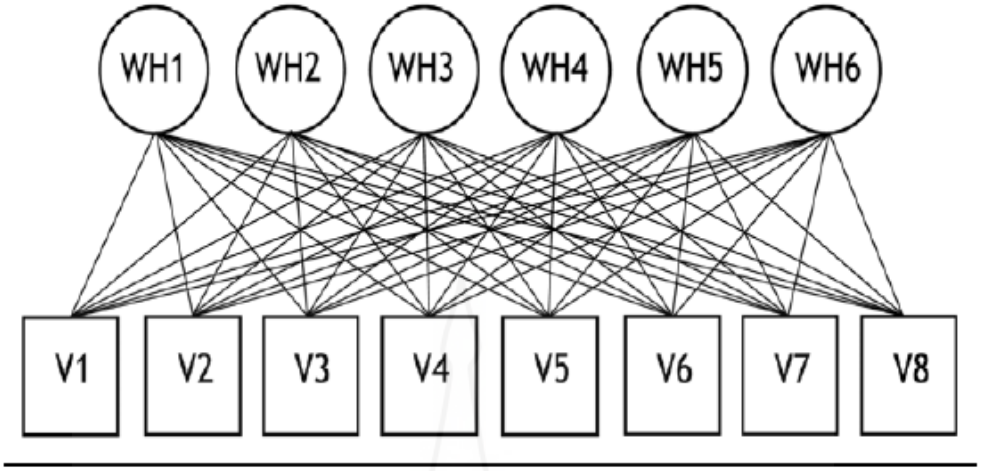
จากตัวแบบนี้พบคำตอบที่ดีที่สุดคือ กำไรที่ 14500 โดยต้องผลิตคอมพิวเตอร์ STANDARD จำนวน 100 เครื่อง ผลิตคอมพิวเตอร์ TURBO จำนวน 30 เครื่อง Reduced Cost หมายถึง ค่าที่ใช้วัดความสามารถในการทำกำไร ในที่นี้ตัวแปรทั้งสองค่ามีค่า Reduced Cost เป็นศูนย์นั่นก็หมายความว่าได้ค่าการทำกำไรได้เหมาะสมแล้ว ต่อไปคือค่า Slack or Surplus ค่านี้ในแถวต่างๆหมายถึงจำนวนทรัพยากรที่ยังคงเหลืออยู่เพราะใช้ไม่หมด(Slack) หรือทรัพยากรที่มีเกินจากข้อกำหนด (Surplus) ส่วน Dual Prices หมายถึง ค่าที่ใช้วัดประสิทธิภาพในการใช้ทรัพยากรซึ่งหมายความถึงว่าหากทรัพยากรที่ใช้ในการผลิตถูกใช้หมดไป และ เราสามารถหาทรัพยากรมาใช้งานเพิ่มได้จะมีผลให้สามารถทำกำไรสูงสุดหรือลดต้นทุนที่ต่ำสุดได้อีก

จากผลลัพธ์ที่เกิดขึ้นของตัวแบบนี้ค่าตัวแปรในแถวที่ 2ก็คือกำลังการผลิตของคอมพิวเตอร์รุ่นSTANDARDมีค่า Slack or Surplus เท่ากับศูนย์หมายความว่ากำลังการผลิตของคอมพิวเตอร์รุ่น STANDARDถูกใช้จนหมดและถ้าหากได้กำลังการผลิตเพิ่มขึ้นทุกๆหน่วยจะทำให้ได้กำไรเพิ่มขึ้นเท่ากับ 25 ส่วนค่าตัวแปรในแถวที่ 3ก็คือกำลังการผลิตของคอมพิวเตอร์รุ่นTURBO มีค่า Slack or Surplus เท่ากับ90 หมายความว่ากำลังการผลิตของคอมพิวเตอร์รุ่น TURBO ที่ถูกใช้ยังเหลืออยู่แต่ก็ไม่ส่งผลให้กำไรเพิ่มขึ้นหากผลิตคอมพิวเตอร์รุ่นTURBO เนื่องจาก Dual Priceเป็นศูนย์ ส่วนค่าตัวแปรในแถวที่ 4ก็คือชั่วโมงในการผลิต มีค่า Slack or Surplus เท่ากับศูนย์ หมายความว่าชั่วโมงที่ใช้ในการผลิตถูกใช้จนหมด และหากได้ชั่วโมงในการผลิตเพิ่มขึ้นจะทำให้ได้กำไรเพิ่มขึ้น 75ต่อหน่วยของชั่วโมงที่ได้รับเพิ่ม

เมื่อต้องการใช้คำสั่งปรินท์รายละเอียดของตัวแบบให้ใช้ปุ่มที่เป็นลักษณะเดียวกับวินโดวส์ และหากต้องการปรินท์บางบรรทัดของตัวแบบสามารถทำได้ไดยการตัด ข้อความและนำไปวาง ไว้ในไฟล์ที่สร้างขึ้นใหม่/ โดยการสสร้างไฟล์ใหม่ให้ใช้ปุ่ม การก็บข้อมูล สามารถทำได้โดยกดปุ่ม Save หรือ ใน Toolbar

ตัวอย่างที่ 2 การสร้างตัวแบบการขนส่ง (Transportation or Shipping network)

ตัวอย่าง บริษัทWireless Widget (WW) มีคลังสินค้าอยู่ 6แห่งด้วยกันสำหรับจ่ายสินค้าให้กับผู้ขาย 8แห่ง ในแต่ละคลังสินค้าไม่สามารถจ่ายได้เกินกว่าสินค้าที่ตัวเองมี และความต้องการของผู้ขายนั้นต้องได้รับการตอบสนองอย่างเป็นที่น่าพอใจ บริษัท WW ต้องการที่จะทราบว่าการดำเนินการขนส่งจากคลังสินค้าสู่ผู้ขายอย่างไรที่จะทำให้เกิดการขนส่งที่มีค่าใช้จ่ายต่ำสุด ตัวแบบในลักษณะนี้เป็นแบบดั้งเดิมของปัญหาการขนส่งสามารถแสดงออกมาเป็นรูปภาพได้ดังนี้



Wireless Widget's Shipping Network

ภาพที่ 4.4 รูปแสดงรายละเอียดของรายละเอียดของการขนส่ง

ในการที่คลังสินค้าสามารถที่จะขนส่งสินค้าให้แก่ผู้ขายนั้นสามารถที่จะทำได้ 48 วิธี ด้วยกันโดยรายละเอียดของคลังสินค้า/ความต้องการของแต่ละผู้ขาย/และค่าใช้จ่าย ข้อกำหนดทั้งหมดนี้มีข้อมูลชี้แจงแสดงให้เห็นดังตารางทางด้านล่าง

Warehouse	Widgets On Hand
1	60
2	55
3	51
4	43
5	41
6	52

Widget Capacity Data

Vendor	Widget Demand
1	35
2	37
3	22
4	32
5	41
6	32
7	43
8	38

Vendor Widget Demand

	V1	V2	V3	V4	V5	V6	V7	V8
Wh1	6	2	6	7	4	2	5	9
Wh2	4	9	5	3	8	5	8	2
Wh3	5	2	1	9	7	4	3	3
Wh4	7	6	7	3	9	2	7	1
Wh5	2	3	9	5	7	2	6	5
Wh6	5	5	2	2	8	1	4	3

Shipping Cost per Widget ($)

ฟังค์ชั่นวัตถุประสงค์ Objective Function ของตัวแบบ: ในฟังค์ชั่นวัตถุประสงค์นี้ บริษัท WW ต้องการทราบค่าขนส่งที่น้อยที่สุดในการขนส่งสินค้าจากคลังสินค้า I ไปยังผู้ขาย J เราก็สามารถเขียนฟังค์ชั่นวัตถุประสงค์ให้อยู่ในรูปแบบของโปรแกรมได้ดังนี้

```
MIN = 6 * VOLUME_1_1 + 2 * VOLUME_1_2 +
      6 * VOLUME_1_3 + 7 * VOLUME_1_4 +
      4 * VOLUME_1_5 +
            .
            .
            .
      8 * VOLUME_6_5 + VOLUME_6_6 + 4 * VOLUME_6_7 +
      3 * VOLUME_6_8;
```

ในการเขียนโปรแกรมดังตัวอย่างข้างต้นอาจทำให้เกิดความผิดพลาดได้โดยง่าย หากเราพิจาณารายละเอียดของตัวอย่างก็สามารถที่แสดงออกมาในรูปแบบของสมการได้ว่า

$$Minimize\ \Sigma_{ij} COST_{ij} \bullet VOLUME_{ij}$$

ภาษาที่ใช้ในโปรแกรม LINGO เพื่อให้สามารถเขียนโปรแกรมได้สั้นลงสะดวกยิ่งขึ้นและง่ายกว่าเดิมโปรแกรมของ LINGOจะเขียนโปรแกรมได้ดังนี้

```
MIN = @SUM(LINKS(I,J): COST(I,J) * VOLUME(I,J));
```

(ความหมายก็คือ ค่าที่น้อยที่สุดจะเท่ากับผลรวมตั้งแต่ I ถึง J ของ COST I ถึง J คูณกับ VOLUME I ถึง J โดยในวงเล็บ LINKS(I,J)นั้นเป็นคำสั่งเพื่อช่วยในการเขียนโปรแกรม)

ฟังค์ชั่นข้อกำหนด Constraints Function: หลังจากที่ได้กำหนดฟังค์ชั่นวัตถุประสงค์แล้วก็จะมีข้อกำหนดอยู่สองอย่างด้วยกัน หนึ่งก็คือผู้ขายต้องได้รับสินค้าตามความต้องการ และสองคือคลังสินค้าส่งสินค้าได้ตามที่มีอยู่ในแต่ละที่เมื่อเขียนในรูปแบบของฟังค์ชั่นของข้อกำหนดจะได้ว่า ความต้องการสินค้าของผู้ขายก็คือ

$$VOLUME_1_1 + VOLUME_2_1 + VOLUME_3_1 + VOLUME_4_1 + VOLUME_5_1 + VOLUME_6_1 = 35;$$

เช่นเดียวกันหากมองในรูปแบบของสมการจะเขียนออกมาได้ว่า

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

ภาษาที่ใช้ในโปรแกรม LINGO เพื่อให้สามารถเขียนโปรแกรมได้สั้นลงสะดวกยิ่งขึ้น และง่ายกว่าเดิมโปรแกรมของ LINGOจะเขียนได้ดังนี้

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

(ความหมายก็คือผลรวมของค่า(I)หรือWAREHOUSES(I) ก็คือVOLUME(I,J) จะเท่ากับ DEMAND(J) สำหรับทุกๆค่า Jใน VENDORS ซึ่ง@FORเป็นคำสั่งของโปรแกรมเพื่อกำหนดค่าตั้งต้นให้แก่ตัวแปร) เมื่อสามารถกำหนดฟังค์ชั่นข้อกำหนดแรกได้แล้วก็ดำเนินการสร้างฟังค์ชั่นข้อกำหนดที่สองต่อไปคือ ความสามารถในการจัดส่งสินค้า สามารถเขียนในรูปแบบของสมการได้

$$\Sigma_j VOLUME_{ij} <= CAP_i, for\ all\ i\ in\ WAREHOUSES$$

ภาษาที่ใช้ในโปรแกรม LINGO เพื่อให้สามารถเขียนโปรแกรมได้สั้นลงสะดวกยิ่งขึ้น และง่ายกว่าเดิมโปรแกรมของ LINGOจะเขียนได้ดังนี้

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

(ความหมายก็คือผลรวมของค่า(J)หรือVENDORS(J) ก็คือVOLUME(I,J) จะเท่ากับ CAPACITY(I) สำหรับทุกๆค่า Iใน WAREHOUSES ซึ่ง@FORเป็นคำสั่งของโปรแกรมเพื่อกำหนดค่าตั้งต้นให้แก่ตัวแปร)

เมื่อรวมฟังค์ชั่นทั้งหมดเข้าด้วยกันก็จะได้รายละเอียดการเขียนโปรแกรมดังนี้

```
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
```

Model: WIDGETS

จากการกำหนดฟังค์ชั่นที่กล่าวมาแล้วโปรแกรมยังไม่สามารถที่จะหาคำตอบได้ทันที่เนื่องจากยังมีรายละเอียดที่ยังไม่ได้กำหนดให้อีกสามส่วนด้วยกันคือ

- warehouses,
- vendors, and
- shipping arcs from each warehouse to customer.

เราต้องกำหนดกลุ่ม(Set) ของรายละเอียดในหัวข้อทั้งสามที่แสดงข้างต้นก่อนโดยเขียนเป็นรายละเอียดข้อกำหนดของเซ็ทได้ดังนี้

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

เมื่อดำเนินการเขียนให้เป็นในรูปแบบของโปรแกรมจะได้

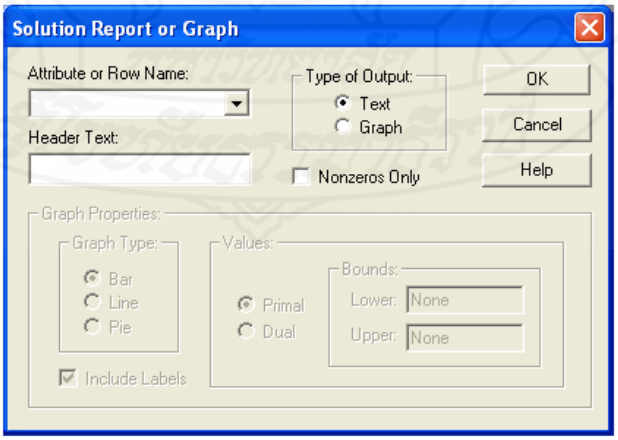
```
DATA:
   !set members;
   WAREHOUSES = WH1 WH2 WH3 WH4 WH5 WH6;
   VENDORS = V1 V2 V3 V4 V5 V6 V7 V8;

   !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
```

เมื่อรวมส่วนของฟังค์ชั่นทั้งหมดเข้าด้วยกันจะสามารถเขียนเป็นตัวแบบโปรแกรมได้

```
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
```

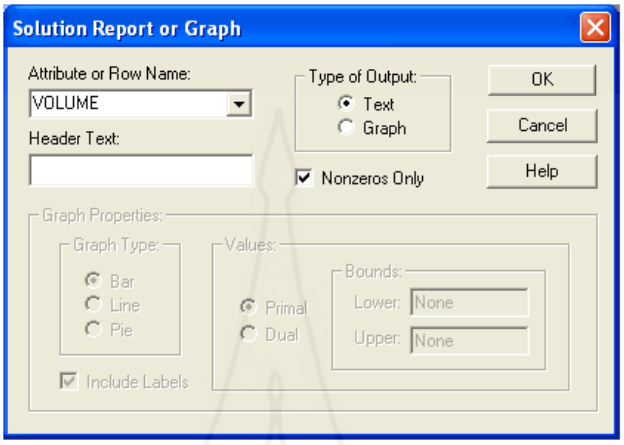
Model: WIDGETS

เวลาใดก็ตามที่คำตอบที่ได้จากโปรแกรม LINGO นั้นแสดงผลออกมามากมาย เราสามารถที่จะเลือกไปที่ Options บนเมนูของ LINGOเลือกไปที่ Interface Tap และเซ็ท Output Level option เป็นTerse โปรแกรม LINGOก็จะแสดงผลออกมาเฉพาะคำตอบเท่านั้น และในการที่จะแสดงผลออกมาเป็นค่าตัวแปรที่ไม่ใช่ศูนย์ของVolume เราสามารถเลือกไปที่คำสั่ง Solution จาก LINGO เมนูก็จะมี Dialog Box แสดงออกมาให้เห็นดังรูป

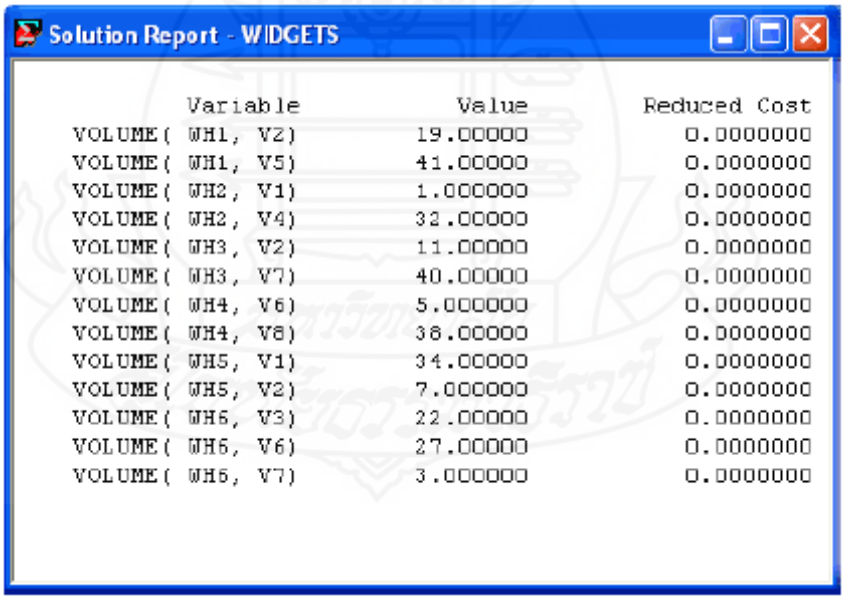


ภาพที่ 4.5 รูปแสดงหน้าต่างของ Solution Report

กดเข้าไปเลือก Volume จาก Attribute or Row Name และ เลือกการแสดงผลที่ Non-zeros Only ตามรูปด้านล่าง



เมื่อทำการเลือกตามที่ต้องการเรียบร้อยแล้ว กดปุ่ม OK โปรแกรมก็จะเสนอเพียงแต่ข้อมูลของ Volume ที่มีค่าไม่เป็นศูนย์เท่านั้นดังรายละเอียดด้านล่าง



Variable	Value	Reduced Cost
VOLUME(WH1, V2)	19.00000	0.0000000
VOLUME(WH1, V5)	41.00000	0.0000000
VOLUME(WH2, V1)	1.000000	0.0000000
VOLUME(WH2, V4)	32.00000	0.0000000
VOLUME(WH3, V2)	11.00000	0.0000000
VOLUME(WH3, V7)	40.00000	0.0000000
VOLUME(WH4, V6)	5.000000	0.0000000
VOLUME(WH4, V8)	38.00000	0.0000000
VOLUME(WH5, V1)	34.00000	0.0000000
VOLUME(WH5, V2)	7.000000	0.0000000
VOLUME(WH6, V3)	22.00000	0.0000000
VOLUME(WH6, V6)	27.00000	0.0000000
VOLUME(WH6, V7)	3.000000	0.0000000

การเพิ่มการแสดงตัวแบบทางภาษา การเพิ่มชื่อของข้อกำหนด (Constraint Name) โปรแกรม LINGO ให้ความสามารถในการกำหนดชื่อของข้อกำหนดในตัวแบบ โดยมีประโยชน์เกิดขึ้นสองประการด้วยกัน หนึ่งคือชื่อของตัวแบบเพื่อใช้ในการแสดงผล และแปรความหมายได้ง่ายขึ้น สองคือหากเกิดข้อผิดพลาดต่างๆในโปรแกรมหากเรากำหนดชื่อของข้อกำหนดในโปรแกรมเอาไว้แล้วก็จะเป็นการง่ายที่จะอ้างอิงกลับไปถึงแหล่งข้อมูลเพื่อการตรวจสอบและปรับปรุงแก้ไขแต่ถ้าหากเราไม่มีชื่อของข้อกำหนดต่างๆเอาไว้ อาจทำให้เราหาแหล่งที่เกิดข้อผิดพลาดได้ยากยิ่งขึ้นและเสียเวลามาก การกำหนดชื่อของข้อกำหนดต่างๆนั้นทำได้โดยง่าย สิ่งที่เราต้องทำก็เพียงแค่ใส่ชื่อที่เราต้องการตั้งไว้ในวงเล็บก่อนหน้าข้อกำหนดต่างๆ ชื่อดังกล่าวต้องเป็นไปตามมาตรฐานของโปรแกรม หรืออีกนัยหนึ่งชื่อทั้งหมดจะต้องเริ่มด้วยตัวอักษร(A-Z) ตัวต่อๆมาสามารถเป็นตัวอักษร,ตัวเลข (0-9)หรือ (_)ก็ได้ ชื่อสามารถกำหนดได้ 64ตัวอักษรในแถว ดังตัวอย่างด้านล่าง

Example 1: `[OBJECTIVE] MIN = X;`
assigns the name *OBJECTIVE* to the model's objective row,

Example 2:
```
@FOR(LINKS(I, J): [DEMAND_ROW]
  @SUM(SOURCES(I): SHIP(I, J)) >=
   DEMAND(J));
```
assigns the name *DEMAND_ROW* to the demand constraints in a transportation model.

เมื่อเรานำตัวอย่างข้างต้นไปใส่ในตัวแบบของ WIDGET โมเดลในการแสดงผลที่รวมการกำหนดชื่อของตัวแบบไว้จะสามารถทำได้ตามตัวอย่างนี้

```
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
```

WIDGETS with Constraint Names

เมื่อทำการคำนวนและประมวลผลจะทำให้แสดงผลได้อย่างชัดเจนมากยิ่งขึ้น รายละเอียดด้านล่าง

Row	Slack or Surplus	Dual Price
OBJECTIVE	664.0000	1.000000
DEMAND_ROW(V1)	0.0000000	-4.000000
DEMAND_ROW(V2)	0.0000000	-5.000000
DEMAND_ROW(V3)	0.0000000	-4.000000
DEMAND_ROW(V4)	0.0000000	-3.000000
DEMAND_ROW(V5)	0.0000000	-7.000000
DEMAND_ROW(V6)	0.0000000	-3.000000
DEMAND_ROW(V7)	0.0000000	-6.000000
DEMAND_ROW(V8)	0.0000000	-2.000000
CAPACITY_ROW(WH1)	0.0000000	3.000000
CAPACITY_ROW(WH2)	22.00000	0.000000
CAPACITY_ROW(WH3)	0.0000000	3.000000
CAPACITY_ROW(WH4)	0.0000000	1.000000
CAPACITY_ROW(WH5)	0.0000000	2.000000
CAPACITY_ROW(WH6)	0.0000000	2.000000

Row Report for WIDGETS with Constraint Names

การเพิ่มชื่อเรื่องของตัวแบบ (MODEL TITLE) เราสามารถที่จะเพิ่มชื่อเรื่องของตัวแบบ โดยการเริ่มต้นด้วยการพิมพ์คำว่า TITLE และจบลงด้วยการพิมพ์ลงท้ายด้วยเครื่องหมายตัวอักษรอัฒภาค (Semicolon) ข้อความที่อยู่ระหว่างคำว่า TITLE และเครื่องหมายอัฒภาคจะถูกแสดงเป็นหัวเรื่องของตัวแบบงาน ดังตัวอย่าง ที่เราได้เพิ่มชื่อเรื่องของตัวแบบเข้าไปยังตัวอย่างของ WIDGET โมเดล ดังตัวอย่างด้านล่าง

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

Excerpt from WIDGETS Model with a Title

Note that when we display the solution report, the title is now displayed along the top:

```
Model Title: Widgets

      Variable           Value        Reduced Cost
CAPACITY( WH1)        60.00000           0.0000000
CAPACITY( WH2)        55.00000           0.0000000
CAPACITY( WH3)        51.00000           0.0000000
CAPACITY( WH4)        43.00000           0.0000000
                        .
                        .
                        .
```

Excerpt from Solution Report to WIDGETS Model with a Title

การใช้เซ็ท (Using Set)

สิ่งที่เราจะกล่าวถึงในส่วนนี้คือเมื่อใดก็ตามที่เราสร้างตัวแบบขึ้นมาโดยตั้งอยู่บนพื้นฐานความเป็นจริงก็จะมีข้อมูลหนึ่งกลุ่มหรือมากกว่านั้นมีความสัมพันธ์ซึ่งกันและกันเป็นกลุ่มเป็นพวกตัวอย่างเช่นกลุ่มของโรงงาน กลุ่มลูกค้า กลุ่มยานพาหนะ หรือกลุ่มคนงาน โปรแกรม LINGO ให้เรารวมเอากลุ่มซึ่งมีความสัมพันธ์ต่อกันรวมเข้าเป็นกลุ่มเป็นพวกโดยที่เรียกว่าเซ็ท (Sets.)ในบทนี้เมื่อเราทำการศึกษาแล้วเราจะได้เข้าใจว่าการนำเซ็ทไปประยุกต์ใช้ในโปรแกรมสามารถทำได้อย่างไร ***ทำไมถึงต้องใช้เซ็ท Why Use Sets?*** เซ็ทเป็นหลักของภาษาที่ใช้ในตัวแบบของโปรแกรม การสร้างบล็อกของโปรแกรมจะทำให้มีความสามารถเพิ่มขึ้นโดยการทำให้เข้าใจว่าเป็นเซ็ท ซึ่งเราสามารถเขียนอันดับของข้อกำหนดที่เหมือนหรือคล้ายคลึงกันในการกำหนดเพียงครั้งเดียวและแสดงความหมายที่ยาวออกไปทำสูตรต่างๆที่สัมพันธ์กันอยู่ให้รัดกุมขึ้นในที่นี้จะทำให้ตัวแบบที่มีขนาดใหญ่สามารถที่จะกำหนดได้กระชับสะดวกรวดเร็วและง่ายขึ้นโดยใช้เซ็ท ตัวอย่างเช่น การเตรียมคลังสินค้าสำหรับตัวแบบการขนส่ง 100 คลังสินค้าจะก่อให้เกิดความเบื่อหน่ายและซ้ำซากหากต้องกำหนดข้อกำหนดที่ต้องการให้แก่ตัวแบบทั้หมด เช่น (กำหนดให้คลังสินค้าที่1 ต้องไม่ส่งของมากกว่าคงคลังที่มีอยู่, กำหนดให้คลังสินค้าที่2 ต้องไม่ส่งของมากกว่าคงคลังที่มีอยู่ และ กำหนดไปเรื่อยๆจนครบข้อกำหนดของคลังสินค้าทั้งหมด) โปรแกรมLINGOให้เราเขียนสูตรเพื่อกำหนดสิ่งต่างๆที่มีได้ง่ายขึ้นต่อการอ่านและทำความเข้าใจ เช่น กำหนดว่า (ทุกคลังสินค้าต้องไม่ส่งของมากกว่าที่คงคลังมีอยู่) ***เซ็ทคืออะไร What are Sets?*** เซ็ทคือกลุ่มของวัตถุที่มีความสัมพันธ์กันอย่างแท้จริง เซ็ทอาจจะเป็นรายการของสินค้า การขนส่ง หรือ ลูกจ้างก็ได้

ทุกๆสมาชิกของเซ็ทอาจมีคุณลักษณะอย่างใดอย่างหนึ่งหรือมากกว่า มีความสัมพันธ์ซึ่งกันและกัน เราเรียกคุณลักษณะนี้ว่าคุณสมบัติ (Attribute) ตัวอย่างเช่น ทุกๆผลิตภัณฑ์ที่อยู่ในเซ็ทของผลิตภัณฑ์อาจจะมีคุณสมบัติเป็นราคา ทุกๆการขนส่งในเซ็ทของการขนส่งจะมีคุณสมบัติเป็นความสามารถในการขนส่ง และทุกๆลูกจ้างในเซ็ทของลูกจ้างอาจมีเงินค่าจ้างเป็นคุณสมบัติ หรือวันเกิดเป็นคุณสมบัติ ***ประเภทของเซ็ท Type of Sets***ในโปรแกรม LINGO จะจดจำเซ็ทได้สองลักษณะคือ(Primitive และ Derived) Primitive Set คือเซ็ทที่ประกอบด้วยวัตถุประสงค์เท่านั้นซึ่งไม่สามารถที่จะลดจำนวนลงต่อไปได้อีก เช่นในตัวอย่างของ Wireless Widgets เซ็ทของ WAREHOUSES ซึ่งประกอบไปด้วย 6 แห่งก็จะเป็น Primitive Set เช่นเดียวกันส่วนประกอบของ 8Vendor ก็คือ Primitive Set

A Derived set หมายถึงการใช้เซ็ทหนึ่งเซ็ทใดหรือเซ็ทอื่นๆ ดังในอีกตัวอย่างหนึ่งของ Wireless Widget เซ็ทประกอบไปด้วยการเชื่อมโยง (LINKS) ของWarehouse6แห่งกับ8Vendor ซึ่งในส่วนนี้เอง(LINKS) คือ Derived Set หมวดหมู่ของเซ็ทในโมเดล (A Sets Section of a Model) ก่อนที่เราจะใช้เซ็ทในโปรแกรม LINGO ตัวแบบของเราต้องกำหนดให้สิ่งทั้งหลายเหล่านั้นเป็นหมวดหมู่ของเซ็ทเสียก่อน ซึ่งหมวดหมู่ของเซ็ทกำหนดได้โดยเริ่มต้นด้วยการพิมพ์ Keyword SETS: และจบลงด้วย Keyword ENDSETS

การกำหนด Primitive Sets เราต้องกำหนด

- ชื่อของเซ็ท
- สมาชิกที่ประกอบอยู่ในเซ็ท
- คุณสมบัติของสมาชิกเป็น Attributes ที่อาจจะมี

ดังนั้นรูปแบบในการกำหนดเซ็ทจะเป็นดังนี้

setname [/ *member_list* /] [: *attribute_list*];

เมื่อแสดงสมาชิกของเซ็ทออกมาได้แน่นอนแล้ว ให้ใส่ชื่อของสมาชิกทั้งหมดแยกกันโดยใช้เครื่องหมายจุลภาค ในตัวอย่างของตัวแบบ Wireless Widget เราสามารถที่จะแสดงรายละเอียดสมาชิกของเซ็ทในWAREHOUSES ได้ดังนี้

```
WAREHOUSES / WH1 WH2 WH3 WH4 WH5 WH6/: CAPACITY;
```

เมื่อแสดงสมาชิกและรายละเอียดของเซ็ทแล้วแต่ยังไม่ได้กำหนดรายชื่อของสมาชิกทุกตัวของเซ็ทดังนั้นเราต้องกำหนดรายชื่อโดยใช้ฟังค์ชั่นดังนี้

setname / *member1*..*memberN* / [: *attribute_list*];

ที่หมายเลข1ของเซ็ทคือชื่อของสมาชิกแรกในเซ็ทจนถึงสมาชิกที่ N คือสมาชิกตัวสุดท้ายของเซ็ท โปรแกรมจะสร้างชื่อของสมาชิกที่อยู่ระหว่างกลางสมาชิกตัวที่1ถึงNโดยอัตโนมัติ ซึ่งจะทำให้สะดวกในการสร้างเซ็ทขึ้นมา ตารางด้านล่างแสดงถึงตัวอย่างที่สามารถดำเนินการได้เป็นแบบต่างๆได้

Implicit Member List Format	Example	Set Members
1..n	1..5	1, 2, 3, 4, 5
stringM..stringN	TRUCKS3..TRUCKS204	TRUCKS3, TRUCKS4, …, TRUCKS204
dayM..dayN	MON..FRI	MON, TUE, WED, THU, FRI
monthM..monthN	OCT..JAN	OCT, NOV, DEC, JAN
monthYearM..monthYearN	OCT2001..JAN2002	OCT2001, NOV2001, DEC2001, JAN2002

ในการแสดงออกมาของตัวแบบ Wireless Widget สามารถที่จะกำหนดเซ็ทของ WAREHOUSES เป็น

```
WAREHOUSES / 1..6/: CAPACITY;
```

ถ้าหากเราใช้รูปแบบที่เป็น 1…n เราต้องกำหนดตำแหน่งและความยาวของเซ็ทในข้อมูลซึ่งกำหนดได้ดังนี้

```
DATA:
   NUMBER_OF_WH = 6;
ENDDATA

SETS:
   WAREHOUSES / 1..NUMBER_OF_WH/: CAPACITY;
ENDSETS
```

การกำหนด Derived Sets เราต้องกำหนด

- ชื่อของเซ็ท
- มูลเหตุของเซ็ท
- สมาชิกและคุณสมบัติของสมาชิกที่เป็น Attributes ที่อาจจะมี

โดยการกำหนดสามารถกระทำได้ดังนี้

setname(*parent_set_list*) [/ *member_list* /] [: *attribute_list*];

ตัวอย่างด้านล่างแสดงรายละเอียดของเซ็ท

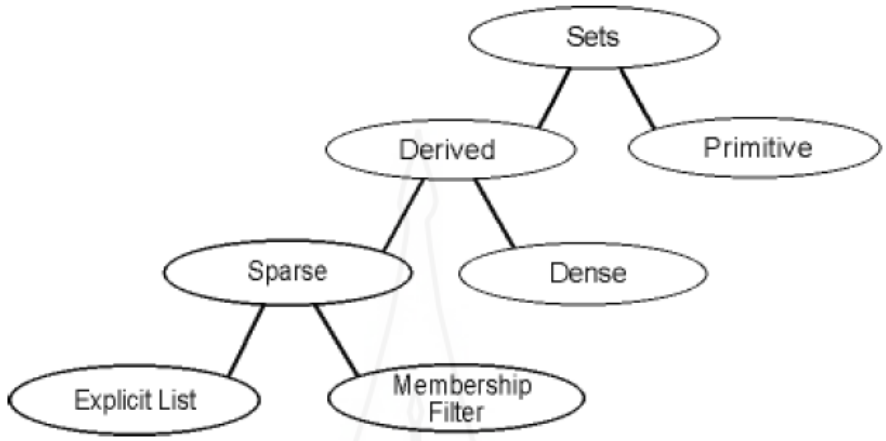
```
SETS:
   PRODUCT / A B/;
   MACHINE / M N/;
   WEEK / 1..2/;
   ALLOWED(PRODUCT, MACHINE, WEEK);
ENDSETS
```

เซ็ทที่เป็น PRODUCT,MACHINEและ WEEK เป็น Primitive เซ็ท ระหว่างที่ ALLOWED เกิดจากมูลเหตุของเซ็ทคือ PRODUCT,MACHINEและWEEK สิ่งที่ได้จากความสัมพันธ์ของทั้งสามส่วนของสมาชิกใน ALLOWEDได้ดังนี้

Index	Member
1	*(A,M,1)*
2	*(A,M,2)*
3	*(A,N,1)*
4	*(A,N,2)*
5	*(B,M,1)*
6	*(B,M,2)*
7	*(B,N,1)*
8	*(B,N,2)*

ALLOWED Set Membership

ความสัมพันธ์ต่างๆของเซ็ทแสดงออกมาเป็นกราฟด้านล่าง



ภาพที่ 4.6 รูปแสดงความสัมพันธ์ของ Set

Primitive Set Examples ในการกำหนด แผนกำหนดการของ Staff โดยการใช้ Primitive เซ็ท โดยตัวแบบของงานมีความต้องการการทำงานล่วงเวลาของ Staff โดยเป้าหมายก็คือการทำงานได้ตรงตามแผนโดยใช้Staff อย่างเหมาะสมและต้นทุนต่ำที่สุด

ตัวอย่างที่4.3 การสร้างตัวแบบปัญหาเรื่องการวางแผนการทำงานของพนักงาน (The Staff scheduling Problem)

สมมุติว่าเรากำลังดำเนินกิจการร้านขายฮอทดอกที่ได้รับความนิยมชื่อ Pluto dog โดยเปิดขาย 7 วันต่อสัปดาห์โดยเราต้องจ้างพนักงานในการทำงาน 5วันทำงานและให้หยุด2วัน พนักงานทุกคนได้รับเงินเป็นรายสัปดาห์ งานบางวันของสัปดาห์จะยุ่งมากกว่าวันอื่นๆ โดยประสบการณ์ที่ผ่านมาทำให้เราทราบว่าต้องการพนักงานในแต่ละวันของสัปดาห์เท่าไหร่ รายละเอียดในการคาดการณ์เพื่อใช้ Staffดังตารางด้านล่าง

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Staff Req'd	20	16	13	16	19	14	12

เราต้องการหาคำตอบว่าจำนวนพนักงานทั้งหมดน้อยที่สุดที่ต้องทำงานทุกวันในสัปดาห์ซึ่งจะต้องเพียงพอต่อความต้องการในแต่ละวันด้วย ในการสร้างสูตรคำนวนคำถามแรกที่จะต้องคิดเมื่อสร้างเซ็ทนี้คือ อะไรคือความสัมพันธ์ของเซ็ทและคุณลักษณะของมัน?ในตัวแบบนี้เรามีเซ็ทที่เป็น Primitive เพียงอย่างเดียวคือวันทำงานในสัปดาห์ ถ้าเราเรียกเซ็ทนี้ว่าDAYS เราสามารถที่จะเริ่มเขียนเซ็ทของเราได้ดังนี้ (ในตัวอย่างนี้จะใช้คำสั่ง Set)

```
SETS:
   DAYS;
ENDSETS
```

ต่อไปเราสามารถที่จะเพิ่มข้อมูลที่เป็นสมาชิกของDAYSเซ็ทดังรายละเอียดด้านล่าง

```
SETS:
   DAYS;
ENDSETS
DATA:
   DAYS = MON TUE WED THU FRI SAT SUN;
ENDDATA
```

ทางเลือกอีกทางหนึ่งเราสามารถใช้โปรแกรม LINGO กำหนดคุณสมบัติของเซ็ทได้อย่างรวดเร็วมากขึ้นด้โดยเราใช้การเขียนตัวแบบทางด้านล่าง

```
SETS:
   DAYS;
ENDSETS
DATA:
   DAYS = MON..SUN;
ENDDATA
```

เราจะพิจารณา2คุณสมบัติของเซ็ทDAYS อันดับแรกคือจำนวนStaff ที่ต้องการทุกวัน และ สองคือจำนวนStaffที่ต้องทำงานทุกวัน หากว่าเราเรียกคุณสมบัตินี้ว่า REQUIRED และ START ดังนั้นเราจะเพิ่มรายละเอียดเข้าไปในเซ็ทได้ว่า

```
SETS:
   DAYS: REQUIRED, START;
ENDSETS
```

หลังจากที่ได้กำหนดเซ็ทและคุณลักษณะของเซ็ทแล้ว ก็จะใช้ในการหาคุณสมบัติทั้งหมดของข้อมูล และ ตัวแปรที่ใช้ในการตัดสินใจ ในตัวแบบนี้ REQUIRED เป็นคุณลักษณะที่ตัวอย่างให้มา ส่วนคุณลักษณะของ START เป็นสิ่งที่ต้องการเราต้องการหาคำตอบ และประกอบด้วยตัวแปรในการตัดสินใจ เมื่อเราได้กำหนดข้อมูลในตัวแบบของ เราต้องดำเนินการต่อตั้งแต่เริ่มแรกโดยการขยายส่วนของข้อมูลออกไปดังต่อไปนี้

```
DATA:
   DAYS =     MON TUE WED THU FRI SAT SUN;
   REQUIRED =  20  16  13  16  19  14  12;
ENDDATA
```

ขณะนี้เราอยู่ที่จุดที่จะสามารถเริ่มเข้าสู่ความสัมพันธ์ทางด้านคณิตศาสตร์ เช่น จุดประสงค์ และคุณลักษณะได้แล้วให้เริ่มต้นโดยการเขียนเครื่องหมายทางด้านคณิตศาสตร์สำหรับจุดประสงค์ซึ่งจุดประสงค์ของเราก็คือจำนวนพนักงานทั้งหมดที่น้อยที่สุดในระหว่างสัปดาห์ ใช้สัญลักษณ์มาตรฐานทางคณิตศาสตร์วัตถุประสงค์สามารถกำหนดได้ดังนี้

$$Minimize: \sum_i START_i$$

ความเหมือนของโปรแกรมLINGOในการกำหนดความหมายสามารถทำได้โดยง่าย ตัวอย่างที่ใช้แทน"MIN=" สำหรับ "MINIMIZE" และ @SUM(DAY(I):" สำหรับ $\sum_i$ จึงทำให้เขียนเป็นวัตถุประสงค์ที่เหมือนกันได้คือ

```
MIN = @SUM( DAYS( I): START( I));
```

เราก็จะได้ข้อกำหนดขึ้นมาหนึ่งกลุ่มของตัวแบบนี้ และ ต้องกำหนดให้มีสตาฟในการทำหน้าที่ทุกๆวันเพื่อให้สอดคล้องกับความต้องการในกรณีนี้สามารถเขียนออกมาเป็นรูปแบบได้คือ

$$Staff\ on\ duty\ today \geq Staff\ required\ today,\ for\ each\ day\ of\ the\ week$$

ทางด้านขวามือของรายละเอียด " Staff required today" เป็นการง่ายที่จะคำนวนหรืออีกนัยหนึ่งก็คือจำนวนของ REQUIRE(I) ส่วนทางด้านซ้ายมือ"Staff on duty today" เป็นการพลิกแพลงเล็กน้อยในการคำนวนโดยกำหนดให้พนักงานจะต้องทำงาน 5วันและหยุด2วัน ตารางการทำงานของพนักงานในการทำงานคือ

Number working today = Number starting today +
Number starting 1 day ago + Number starting 2 days ago +
Number starting 3 days ago + Number starting 4 days ago.

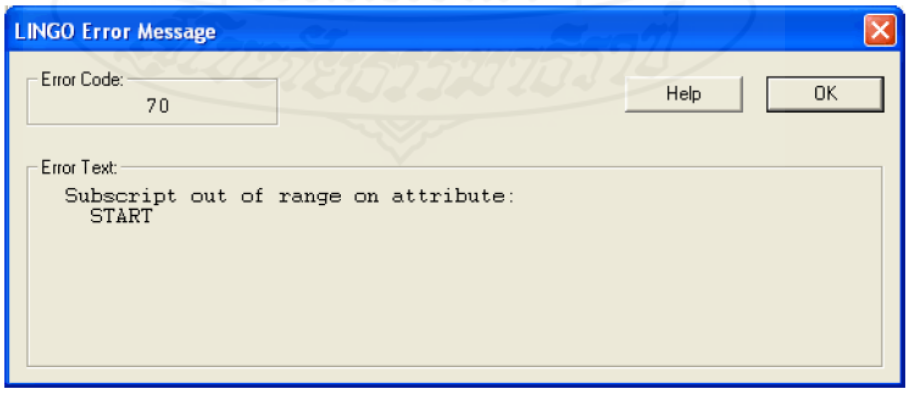
ในอีกความหมายหนึ่ง การคำนวนจำนวนของพนักงานที่ทำงานในวันนี้ ก็คือผลรวมจำนวนของพนักงานที่ทำงานวันนี้และบวกกลับไปยัง4วันที่ผ่านมา จำนวนของพนักงานที่เริ่ม 5 และ 6 วันย้อนหลังจะไม่ถูกนำมารวมด้วยเพราะถือว่าเป็นวันหยุด ดังนั้นเราเขียนตัวแบบโดยใช้สัญลักษณ์ทางคณิตศาสตร์ ที่เราต้องการเพิ่มเข้าไปในข้อกำหนดได้ดังนี้

$$\Sigma_{i=j-4,\,j}\, START_i \geq REQUIRED_j,\ for\ j \in DAYS$$

เปลี่ยนแปลงเป็นภาษาของโปรแกรม LINGO ได้ดังนี้

```
@FOR( DAYS( J):
   @SUM( DAYS( I) | I #LE# 5: START( J - I + 1))
      >= REQUIRED( J)
);
```

ในข้อความด้านบนโปรแกรม LINGO แจ้งว่าสำหรับทุกวันในสัปดาห์ผลรวมของพนักงานเริ่มต้นเกินระยะเวลา 5 วัน โดยเริ่มจากเมื่อ 4 วันที่ผ่านมาจนถึงวันนี้ต้องมากกว่าหรือเท่ากันกับความต้องการสตาฟในแต่ละวัน แต่ก็จะเกิดปัญหาขึ้นเล็กน้อย เมื่อทดลองเริ่มแก้ไขปัญหาโดยการRunโปรแกรมเราจะพบว่าเกิดข้อความเรื่องความผิดพลาดแจ้งเตือนขึ้นดังรูปแบบด้านล่าง



ภาพที่ 4.7 รูปแสดงการแจ้งเตือนเมื่อเกิดความผิดพลาด

เราลองมาพิจารณาดูว่าทำไมถึงเกิดข้อความแจ้งเตือนเกิดขึ้น และลองพิจารณาดูว่าเกิดอะไรขึ้นในวันพฤหัสบดี เนื่องจากในวันพฤหัสบดีมีเลขชี้กำลังคือ 4 ในDAYSเซ็ทของเรา ในการเขียน สตาฟของข้อกำหนดสำหรับวันพฤหัสบดีคือ

```
START( 4 - 1 + 1) + START( 4 - 2 + 1) +
 START( 4 - 3 + 1) + START( 4 - 4 + 1) +
  START( 4 - 5 + 1) >= REQUIRED( 4);
```

อธิบายง่ายๆก็คือ

```
START( 4) + START( 3) +
 START( 2) + START( 1) +
  START( 0) >= REQUIRED( 4);
```

ในที่นี้ปัญหาเกิดจาก START(0) เนื่องจากไม่มีค่าอยู่ ทำให้อยู่นอกเหนือค่าในการพิจารณา เราจึงต้องการเครื่องบ่งชี้ค่าน้อยกว่าหรือเท่ากับศูนย์นั้น ให้ปิดบังและไม่นำมาคำนวนไปจนถึงสุดสัปดาห์โดยโปรแกรม LINGO มีฟังค์ชั่นในการใช้คือฟังค์ชั่น @WRAP.ฟังค์ชั่นของ@WRAP ให้เหตุผล 2 ส่วน เรียกว่า INDEX และ LIMITกล่าวง่ายๆก็คือ @WRAP return J จนถึง J = INDEX-K* LIMIT โดยที่ K เป็นจำนวนเต็มจนกระทั่งระยะห่างถึง J [1,LIMIT] @WRAP จะลบหรือเพิ่ม LIMIT ใน INDEX จนกระทั่งเข้าไปอยู่ในขอบเขตของ 1 ถึง LMIT ในการใช้คำสั่ง @WRAP เราสามารถดำเนินการให้ถูกต้อง ตามแบบของข้อกำหนดได้ดังนี้

```
@FOR( DAYS( J):
   @SUM( DAYS( I) | I #LE# 5:
      START( @WRAP( J - I + 1, 7)))
         >= REQUIRED( J)
);
```

สุดท้ายนี้เมื่อได้รายละเอียดครบแล้วในการใส่ค่าและเขียนโปรแกรมของตัวแบบนี้คือ

```
MODEL:
SETS:
   DAYS: REQUIRED, START;
ENDSETS

DATA:
   DAYS =     MON TUE WED THU FRI SAT SUN;
   REQUIRED =  20  16  13  16  19  14  12;
ENDDATA

MIN = @SUM( DAYS( I): START( I));

@FOR( DAYS( J):
   @SUM( DAYS( I) | I #LE# 5:
      START( @WRAP( J - I + 1, 7)))
         >= REQUIRED( J)
);
END
```

Model: STAFFDEM

ผลที่ได้ของการ RUN โปรแกรมจะได้คำตอบดังนี้

```
Global optimal solution found.
Objective value:                              22.00000
Infeasibilities:                              0.000000
Total solver iterations:                             5


     Variable           Value        Reduced Cost
REQUIRED( MON)        20.00000            0.000000
REQUIRED( TUE)        16.00000            0.000000
REQUIRED( WED)        13.00000            0.000000
REQUIRED( THU)        16.00000            0.000000
REQUIRED( FRI)        19.00000            0.000000
REQUIRED( SAT)        14.00000            0.000000
REQUIRED( SUN)        12.00000            0.000000
   START( MON)        8.000000            0.000000
   START( TUE)        2.000000            0.000000
   START( WED)        0.000000            0.000000
   START( THU)        6.000000            0.000000
   START( FRI)        3.000000            0.000000
   START( SAT)        3.000000            0.000000
   START( SUN)        0.000000           0.3333333

         Row    Slack or Surplus      Dual Price
           1        22.00000           -1.000000
           2        0.000000          -0.3333333
           3        0.000000            0.000000
           4        0.000000          -0.3333333
           5        0.000000            0.000000
           6        0.000000          -0.3333333
           7        0.000000          -0.3333333
           8        0.000000            0.000000
```

Solution to STAFFDEM

การแปลความหมายที่ได้จากคำตอบ ค่าของ Object Value หมายถึง ต้องจ้างพนักงาน 22 คน โดยให้พนักงานดำเนินงานตามรูปแบบของแผนงานดังนี้

Day	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Start	8	2	0	6	3	3	0

ถ้าเราดูที่ส่วนของ Surpluses ในช่องความต้องการของสตาฟในแถวที่ 2 จนถึง7 ค่า Slack ทั้งหมดนี้มีค่าเป็น 0 ในทุกๆวัน นั้นหมายความว่าไม่มีพนักงานที่วางแผนตามแผนงานนี้มากกว่าหรือเกินจากความต้องการในแต่ละวัน

ตัวอย่างที่4.4 การสร้างตัวแบบพิกัดน้ำหนักของส่วนผสมเมล็ดพืช

(Dense Derived Set Example)

เป็นโมเดลซึ่งแสดงถึงความต้องการผสมกันของพันธุ์พืชเพื่อให้เกิดเป็นสินค้า โดยเป้าหมายเพื่อผสมวัตถุดิบหลักๆเข้าด้วยกันเพื่อให้เกิดคุณภาพและความประหยัดที่สุดที่ต้องการ ในหนึ่งหรือหลายมิติ ปัญหาของตัวแบบคือบริษัท Chess Snackfoods มีตลาดของการผสมถั่วอยู่ 4 ประเภทสินค้าด้วยกันคือ Pawn,Knight,Bishop และ King ทุกตราสินค้ามีมาตรฐานในการกำหนดส่วนผสมระหว่างถั่วลิสงและเม็ดมะม่วงหิมพานดังตารางด้านล่างจะแสดงถึงส่วนผสมแต่ละตราสินค้าและราคาขาย

	Pawn	Knight	Bishop	King
Peanuts (oz.)	15	10	6	2
Cashews (oz.)	1	6	10	14
Selling Price ($)	2	3	4	5

บริษัทได้มีการติดต่อให้ผู้ส่งสินค้าจัดส่งถั่วลิสง 750 ปอนด์ต่อวัน และ เม็ดมะม่วงหิมพาน250 ปอนด์ต่อวัน ปัญหาก็คือ การกำหนดจำนวนปอนด์ของแต่ละตราสินค้าในการผลิตแต่ละวันเพื่อให้เกิดกำไรสูงสุดโดยปราศจากปัญหาการจัดส่งถั่วจากผู้จัดส่งสินค้า

สูตรในการสร้างตัวแบบ กลุ่มของตัวแบบในตัวอย่างนี้ คือ ประเภทของถั่ว และประเภทของการผสมถั่ว เราสามารถที่จะเขียนออกมาในลักษณะของ SET ได้ดังนี้

```
SETS:
   NUTS / PEANUTS, CASHEWS/: SUPPLY;
   BRANDS / PAWN, KNIGHT, BISHOP, KING/:
    PRICE, PRODUCE;
ENDSETS
```

กลุ่มข้อมูลของถั่วเป็นตัวแปรของ SUPPLY ซึ่งเราใช้ในการกำหนดการจ่ายปริมาณของถั่วส่วนกลุ่มของ BRAND เป็นราคาของตัวแปรสินค้าซึ่งใช้ในการขาย และ PRODUCE เป็นค่าของตัวแปรในการตัดสินใจปริมาณของน้ำหนักในแต่ละประเภทสินค้าที่ผลิตในแต่ละวัน เราต้องการกลุ่มตัวแปรอีกกลุ่มหนึ่งซึ่งเป็นกลุ่มของพันธุ์พืชที่ได้สัญญาว่าจะทำการสั่งเข้ามาในสูตรของ Brand เราต้องการตารางแสดงมิติสองอย่างในการกำหนดประเภทของถั่วและประเภทสินค้าในการทำนี้โดยเพิ่มเซ็ทของส่วนผสม NUT และ BRAND ดังตัวอย่าง

```
SETS:
   NUTS / PEANUTS, CASHEWS/: SUPPLY;
   BRANDS / PAWN, KNIGHT, BISHOP, KING/:
    PRICE, PRODUCE;
   FORMULA(NUTS, BRANDS): OUNCES;
ENDSETS
```

เซ็ทของถั่วนั้นมีคุณลักษณะเพื่อการจัดส่งซึ่งเราใช้การจัดเก็บถั่วประจำวันเป็นน้ำหนักปอนด์เซ็ทของตราสินค้าจะประกอบด้วยคุณลักษณะเรื่องราคา(PRICE) และการผลิตสินค้า (PRODUCE) ที่ซึ่ง PRICE เก็บข้อมูลของราคาการขายของแต่ละตราสินค้า และ PRODUCE นำเสนอตัวแปรในการตัดสินใจของปริมาณน้ำหนักของแต่ละตราสินค้าในการผลิตต่อวัน
เราต้องการเซ็ทของข้อมูลเพิ่มเติม ซึ่งก็คือเซ็ทของการกำหนดส่วนผสมของเมล็ดพืช โดยใส่ค่าลงในสูตรของแต่และตราสินค้า เราจึงต้องการมิติสองประเภทในตารางเพื่อที่จะกำหนดเรื่องชนิดของถั่ว และ ประเภทของตราสินค้า ซึ่งการที่จะสร้างความสัมพันธ์ระหว่างถั่วและตราสินค้าของเซ็ทดังกล่าวนี้ เราสามารถทำได้โดยการเพิ่มเงื่อนไขดังตัวอย่างด้านล่าง

```
SETS:
   NUTS / PEANUTS, CASHEWS/: SUPPLY;
   BRANDS / PAWN, KNIGHT, BISHOP, KING/:
    PRICE, PRODUCE;
   FORMULA(NUTS, BRANDS): OUNCES;
ENDSETS
```

เรามีหัวข้อของ Derived set เป็นสูตรในการคำนวนซึ่งประกอบด้วยตัวแปรคือน้ำหนักเป็นออนซ์ ซึ่งให้ใส่ค่าน้ำหนักของถั่วเป็นออนซ์ที่ใช้ต่อปอนด์ของทุกตราสินค้า เมื่อได้กำหนดเซ็ทของเราเสร็จสิ้นก็จะใส่ค่าของ SUPPLY, PRICE และ OUNCES ดังข้อมูล

```
DATA:
   SUPPLY = 750 250;
   PRICE = 2 3 4 5;
   OUNCES = 15 10  6  2
             1  6 10 14;
ENDDATA
```

ต่อจากนั้นเราสามารถที่กำหนดฟังค์ชั่นวัตถุประสงค์ และ ข้อกำหนดฟังค์ชั่นวัตถุประสงค์ก็คือรายรับสูงสุดซึ่งกำหนดได้ดังนี้

```
MAX = @SUM(BRANDS(I):
 PRICE(I) * PRODUCE(I));
```

และเราสามารถกำหนดให้เป็นในรูปแบบโปรแกรมได้ดังนี้

```
@FOR(NUTS(I):
 @SUM(BRANDS(J):
  OUNCES(I, J) * PRODUCE(J) / 16) <=
   SUPPLY(I)
);
```

โดยตัวแบบของโปรแกรมทั้งหมดเมื่อสร้างข้อกำหนดได้เสร็จสิ้นจะเป็นดังนี้

```
SETS:
   NUTS / PEANUTS, CASHEWS/: SUPPLY;
   BRANDS / PAWN, KNIGHT, BISHOP, KING/:
    PRICE, PRODUCE;
   FORMULA(NUTS, BRANDS): OUNCES;
ENDSETS

DATA:
   SUPPLY = 750 250;
   PRICE = 2 3 4 5;
   OUNCES= 15 10  6  2
           1  6 10 14;
ENDDATA

MAX = @SUM(BRANDS(I):
 PRICE(I) * PRODUCE(I));

@FOR(NUTS(I):
 @SUM(BRANDS(J):
  OUNCES(I, J) * PRODUCE(J) / 16) <=
   SUPPLY(I)
);
```

Model: CHESS

เมื่อทำการ RUN โปรแกรมแล้วโปรแกรมจะแสดงผลปรากฏดังรายละเอียดด้านล่าง

```
Global optimal solution found.
Objective value:                              2692.308
Infeasibilities:                              0.000000
Total solver iterations:                             2


                     Variable           Value        Reduced Cost
               SUPPLY( PEANUTS)        750.0000            0.000000
               SUPPLY( CASHEWS)        250.0000            0.000000
                   PRICE( PAWN)        2.000000            0.000000
                 PRICE( KNIGHT)        3.000000            0.000000
                 PRICE( BISHOP)        4.000000            0.000000
                   PRICE( KING)        5.000000            0.000000
                 PRODUCE( PAWN)        769.2308            0.000000
               PRODUCE( KNIGHT)        0.000000            0.1538462
               PRODUCE( BISHOP)        0.000000            0.7692308E-01
                 PRODUCE( KING)        230.7692            0.000000
         OUNCES( PEANUTS, PAWN)        15.00000            0.000000
       OUNCES( PEANUTS, KNIGHT)        10.00000            0.000000
       OUNCES( PEANUTS, BISHOP)        6.000000            0.000000
         OUNCES( PEANUTS, KING)        2.000000            0.000000
         OUNCES( CASHEWS, PAWN)        1.000000            0.000000
       OUNCES( CASHEWS, KNIGHT)        6.000000            0.000000
       OUNCES( CASHEWS, BISHOP)        10.00000            0.000000
         OUNCES( CASHEWS, KING)        14.00000            0.000000

                        Row    Slack or Surplus      Dual Price
                          1        2692.308            1.000000
                          2        0.000000            1.769231
                          3        0.000000            5.461538
```

Solution to CHESS

ผลที่เกิดขึ้นแสดงให้ทราบว่าควรที่จะผลิต 769.2 ปอนด์ของประเภทสินค้า Pawn mix และ 230.8ของ King เพื่อที่จะให้ได้ผลตอบแทนสูงสุดที่ 2,692.30 เหรียญโดยมีข้อมูลที่น่าสนใจเพิ่มเติมว่าบริษัทจะต้องจ่ายเพิ่มขึ้น1.77เหรียญในทุกๆน้ำหนักพิเศษที่ต้องการเพิ่มขึ้นของถั่ว และ 5.46เหรียญหากต้องการเพิ่มขึ้นของเมล็ดมะม่วงหิมพานต์ ในความหมายของด้านการตลาด บริษัทตัดสินใจที่จะต้องไม่ผลิตสินค้าในตรา Knight และ Bishop เนื่องจากช่อง Reduce Cost แสดงให้เราทราบว่าผลตอบแทนจะลดลง 15.4 เซ็นต์หากผลิต Knight เพิ่มขึ้น และ ลดลง 7.7 เซ็นต์หากผลิต Bishop เพิ่มขึ้นในแต่ละปอนด์

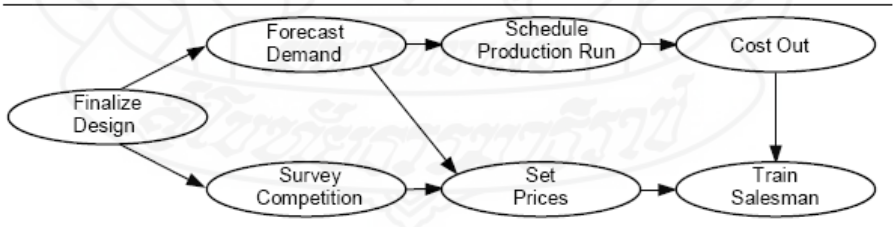
ตัวอย่างที่ 4.5 ตัวแบบของการกำหนดกำลังคนโดยแสดงออกมาเป็นรายการ (Sparse Derived Set Example-Explicit List)

ในตัวอย่างนี้เราจะแนะนำในการใช้เซ็ทของการกำหนดกำลังคนโดยการแสดงออกมาให้เป็นรายการ เมื่อเราใช้เทคนิคนี้ในการกำหนดเซ็ทของส่วนผสมเราต้องแสดงรายการของสมาชิกทั้งหมดอยู่ในเซ็ท โดยปกติจะเป็นเซ็ทของกลุ่มย่อยของการผสมผสานกันจากCartesian Product

ของมูลเหตุเซ็ท ในตัวอย่างเราต้องทำการเซ็ทอัพ PERT (เป็นเทคนิคการประเมินโครงการ) เป็นตัวแบบในการหาคำตอบที่เป็นสายงานวิกฤติในการดำเนินการผลิตภัณฑ์ใหม่ PERT เป็นรูปแบบที่ง่ายและ ทรงพลังในการดำเนินงาน เทคนิคนี้ได้ถูกสร้างขึ้นเมื่อปี 1950 ในการช่วยบริหารงานและติดตามผลของระยะเวลาในการดำเนินงานของโครงการซึ่ง PERT จะสามารถที่จะกำหนดกิจกรรมที่เป็นสายงานวิกฤติภายในโครงการซึ่งหากเกิดความล่าช้าก็จะทำให้ทั้งโครงการล่าช้าออกไปทั้งหมดเวลาที่เป็นสายงานวิกฤตินี้จะอ้างอิงถึงการเชื่อมโยงของส่วนต่างในโครงการที่ทำให้เกิดความล่าช้าที่สุดหรือเรียกสายงานวิกฤติ ปัญหาในตัวแบบคือบริษัท Wireless Widget ได้ดำเนินการเรื่องผลิตภัณฑ์ใหม่เรียกว่า Solar Widget ในการรับประกันโครงการนี้จะต้องดำเนินการเสร็จตามแผนงาน Wireless Widget ต้องการนำ PERT มาเป็นเครื่องมือในการช่วยในการดำเนินงาน และ ในการดำเนินงานต้องกำหนดส่วนที่ทำเกิดเป็นสายงานวิกฤติเพื่อทำให้การดำเนินงานตรงตามเวลาที่กำหนดเวลาในส่วนต่างของการดำเนินงานแสดงไว้ดังตารางด้านล่าง

Task	Weeks
Finalize Design	10
Forecast Demand	14
Survey Competition	3
Set Prices	3
Schedule Production Run	7
Cost Out	4
Train Salesmen	10

และความเชื่อมโยงของส่วนต่างในการดำเนินการแสดงให้ทราบเป็นดังรายละเอียดนี้



Product Launch Precedence Relations

เป้าหมายของเราก็คือกำหนดสายงานวิกฤติที่เกิดขึ้นในตัวแบบนี้เริ่มโดยเราต้องการ Primitive เซ็ทในการแสดงส่วนต่างๆที่เชื่อมโยงของโครงการซึ่งกำหนดเป็นนิยามได้ดังนี้

```
TASKS / DESIGN, FORECAST, SURVEY, PRICE,
 SCHEDULE, COSTOUT, TRAIN/: TIME, ES, LS, SLACK;
```

เรามีคุณสมบัติอยู่สี่อย่างในเซ็ทซึ่งก็คือ TIME เวลาที่ดำเนินการให้เสร็จในแต่ละส่วน ES คือเวลาเริ่มต้นเร็วที่สุดที่สามารถดำเนินการได้ในแต่ละกิจกรรม, LS คือเวลาที่เป็นไปได้ในการเริ่มงานแต่ละกิจกรรมน้อยที่สุด, SLACK คือความแตกต่างระหว่าง LS และ ES ในแต่ละกิจกรรม รายละเอียดเรื่องเวลาการทำงานในแต่ละส่วนได้ถูกกำหนดไว้ให้แล้วเราจะคำนวนค่าของคุณสมบัติที่เหลืออยู่สามส่วนด้วยกันหากว่ากิจกรรมมีค่าของ Slack Time เป็น0 นั่นหมายความว่ากิจกรรมต้องเริ่มตรงเวลามิเช่นนั้นโครงการทั้งหมดจะเกิดความล่าช้า หรืออธิบายได้ว่าสายงานใดที่มีกิจกรรมในสายงานนั้นค่า Slack เป็น 0 ทั้งหมดก็คือสายงานวิกฤติในการออกคำสั่งให้คำนวนเวลาเริ่มงานของแต่ละกิจกรรมเราต้องกำหนดลำดับของความสัมพันธ์ลงในตัวแบบ ลำดับของความสัมพันธ์สามารถพิจารณาจากรายละเอียดเช่นกิจกรรมของการDESIGN ต้องเสร็จสิ้นก่อนกิจกรรม FORECAST สร้างความสัมพันธ์ของรูปแบบลงในเซ็ทของ TASKS ได้ดังนี้

```
PRED(TASKS, TASKS) /
 DESIGN,FORECAST,
 DESIGN,SURVEY,
 FORECAST,PRICE,
 FORECAST,SCHEDULE,
 SURVEY,PRICE,
 SCHEDULE,COSTOUT,
 PRICE,TRAIN,
 COSTOUT,TRAIN /;
```

เซ็ท PRED เป็นเซ็ทของการผสมผสานกันซึ่งมีเซ็ทย่อยๆประกอบเข้าด้วยกันโดยให้รายละเอียดของข้อมูลในการดำเนินการก่อนและหลังดังที่ได้อธิบายไว้ตอนต้น ต่อมาเราต้องใส่ค่าเวลาการดำเนินงานแต่ละกิจกรรม คือ

```
DATA:
   TIME = 10, 14, 3, 3, 7, 4, 10;
ENDDATA
```

เมื่อกำหนดเวลาดำเนินการของแต่ละกิจกรรมแล้ว เราก็สามารถเริ่มสร้างสูตรของตัวแบบได้ ซึ่งเรามีคุณสมบัติสามประการที่ต้องทำการคำนวนคือ การเริ่มกิจกรรมได้เร็วที่สุด (Earliest Start (ES)) การเริ่มกิจกรรมได้ช้าที่สุด (Latest Start (LS)) และเวลา SLACK เทคนิคที่ใช้พลิกแพลงเพื่อใช้ในการคำนวนในเรื่องนี้คือการคำนวนเวลา ES,LS และ SLACK เริ่มต้นด้วยการคำนวนค่า ES ของกิจกรรมการทำงานในกิจกรรมนั้นๆจะยังไม่สามารถที่จะดำเนินงานได้จนกระทั่งการทำงานในกิจกรรมก่อนหน้านี้ได้เสร็จสิ้นลง นั่นก็คือถ้าเราได้ดำเนินการหาเวลาเสร็จสิ้นท้ายสุดของ

กิจกรรมก่อนหน้า เราก็จะพบกับเวลาเริ่มต้นได้เร็วที่สุดของกิจกรรมนั้นๆ หรือในความหมายก็คือ เวลาเริ่มต้นเร็วที่สุดของกิจกรรม t จะเท่ากับค่าของผลรวมของเวลาเริ่มต้นกิจกรรมกับเวลาดำเนินการในกิจกรรมที่มากที่สุดก่อนหน้ากิจกรรม t ในการกำหนดเป็นเครื่องหมายของโปรแกรม LINGO คือ

```
@FOR(TASKS(J)| J #GT# 1:
 ES(J) = @MAX(PRED(I, J): ES(I) + TIME(I))
);
```

เราจะข้ามการคำนวนของกิจกรรมแรกโดยการเพิ่มข้อจำกัดว่า J#GT#1เป็นคำสั่งในโปรแกรม เนื่องจากว่าในกิจกรรมแรกเราไม่ต้องการการคำนวนและก็จะใช้เวลาในกิจกรรมแรกเป็นเพียงเวลาเริ่มต้นเท่านั้น ส่วนของการคำนวนเวลาของ LS ก็ค่อนข้างคล้ายคลึงกับการคำนวน ES ซึ่งในความหมายของ LS ก็คือเวลาเริ่มต้นช้าที่สุดของกิจกรรม t จะเท่ากับเวลาที่เริ่มต้นได้เร็วที่สุดลบกับเวลาปฏิบัติงานในกิจกรรม t หากกิจกรรม t เริ่มต้นช้ากว่านี้จะต้องห้ามกิจกรรมสุดท้ายก่อนหน้านี้ให้เริ่มต้นเร็วขึ้นด้วยโดยกำหนดเป็นเครื่องหมายทางโปรแกรมว่า

```
@FOR(TASKS(I)| I #LT# LTASK:
 LS(I) = @MIN(PRED(I, J): LS(J) - TIME(I))
);
```

การคำนวน Slack Time ก็คือความแตกต่างระหว่าง LS และ ES เราจะเขียนเป็นข้อกำหนดได้ว่า

```
@FOR(TASKS(I): SLACK(I) = LS(I) - ES(I));
```

เราสามารถกำหนดเวลาเริ่มต้นของกิจกรรมแรกได้โดยไม่คำนึงถึงกฎเกณฑ์ โดยในจุดประสงค์เราจะกำหนดค่าเป็น 0 โดยเขียนว่า

```
ES(1) = 0;
```

เราได้กำหนดสูตรสำหรับการคำนวนค่าของตัวแปรทั้งหมด โดยยกเว้นเวลาเริ่มต้นช้าที่สุดของกิจกรรมสุดท้าย โดยความหมายของมันก็คือว่าเวลาที่เริ่มต้นช้าที่สุดของกิจกรรมสุดท้ายเท่ากับเวลาที่เริ่มต้นได้เร็วที่สุดในกิจกรรมนี้โดยสามารถเขียนเป็นสมการของโปรแกรมดังนี้

```
LS(7) = ES(7);
```

ในขณะนี้ก็สามารถที่จะเริ่มทำงานได้แล้วแต่ก็ยังไม่ใช่หนทางที่เร็วที่สุดในการกำหนดความสัมพันธ์ เราควรที่จะเพิ่มกิจกรรมลงในตัวแบบของเรา โดยการเปลี่ยนเลข7ในสมการเป็นตัวเลขที่กิจกรรมต้องการ ข้อมูลทั้งหมดที่ได้กล่าวไปแล้วข้างต้นนั้นสมการของโปรแกรมต้องการที่จะแยกตัวแปรของสมการให้เป็นอิสระออกซึ่งสามารถทำได้โดยใช้คำสั่งในตัวแบบดังนี้

```
LTASK = @SIZE(TASKS);
LS(LTASK) = ES(LTASK);
```

ฟังค์ชั่นของ @SIZE เป็นการย้อนกลับขนาดของเซ็ทเพื่อใช้ในกรณีนี้โปรแกรมจะย้อนกลับค่าของ7ในสมการเป็นค่าที่ต้องการ หากว่าเราเปลี่ยนตัวเลขของกิจกรรม @SIZE ก็จะเปลี่ยนเป็นค่าใหม่ที่ถูกต้องตามด้วย ก็จะทำให้ข้อกำหนดที่ต้องการข้อมูลที่เป็นอิสระสามารถทำได้ตามต้องการในสมการ ตัวแบบนี้สามารถที่จะดำเนินการเป็นตัวแบบทั้งหมดได้ดังนี้

```
SETS:
   TASKS / DESIGN, FORECAST, SURVEY, PRICE,
    SCHEDULE, COSTOUT, TRAIN/: TIME, ES, LS, SLACK;

   PRED(TASKS, TASKS) /
    DESIGN,FORECAST,
    DESIGN,SURVEY,
    FORECAST,PRICE,
    FORECAST,SCHEDULE,
    SURVEY,PRICE,
    SCHEDULE,COSTOUT,
    PRICE,TRAIN,
    COSTOUT,TRAIN /;
ENDSETS

DATA:
   TIME = 10, 14, 3, 3, 7, 4, 10;
ENDDATA

@FOR(TASKS(J)| J #GT# 1:
 ES(J) = @MAX(PRED(I, J): ES(I) + TIME(I))
);

@FOR(TASKS(I)| I #LT# LTASK:
 LS(I) = @MIN(PRED(I, J): LS(J) - TIME(I));
);

@FOR(TASKS(I): SLACK(I) = LS(I) - ES(I));

ES(1) = 0;
LTASK = @SIZE(TASKS);
LS(LTASK) = ES(LTASK);
```

Model: PERT

คำตอบของตัวแบบที่ได้ก็จะแสดงผลออกมาดังนี้

Feasible solution found at step: 0

Variable	Value
LTASK	7.000000
ES(DESIGN)	0.0000000
ES(FORECAST)	10.00000
ES(SURVEY)	10.00000
ES(PRICE)	24.00000
ES(SCHEDULE)	24.00000
ES(COSTOUT)	31.00000
ES(TRAIN)	35.00000
LS(DESIGN)	0.0000000
LS(FORECAST)	10.00000
LS(SURVEY)	29.00000
LS(PRICE)	32.00000
LS(SCHEDULE)	24.00000
LS(COSTOUT)	31.00000
LS(TRAIN)	35.00000
SLACK(DESIGN)	0.0000000
SLACK(FORECAST)	0.0000000
SLACK(SURVEY)	19.00000
SLACK(PRICE)	8.000000
SLACK(SCHEDULE)	0.0000000
SLACK(COSTOUT)	0.0000000
SLACK(TRAIN)	0.0000000

Solution to PERT

จากคำตอบของตัวแบบนี้ตัวแปรที่น่าสนใจก็คือ Slacks ในแต่ละกิจกรรม ทั้ง SURVEY และ PRICE มี Slack ในเวลาเริ่มของสัปดาห์ที่19และสัปดาห์ที่8โดยลำดับ กิจกรรมการออกแบบ (DESIGN) ,การพยากรณ์(FORECAST),การวางแผน (SCHEDULE),การนำเสนอราคา (COST OUT)การอบรม(TRAIN)มีค่า Slack เวลาเป็น 0 ดังนั้นมันจึงเรียกได้ว่าเป็นสายงานวิกฤติ สำหรับกิจกรรมนี้ การจัดการจำต้องให้ความสำคัญกับสายงานวิกฤตินี้ต้องเริ่มและเสร็จตรงตามเวลาสุดท้ายนี้ค่าของ ES(TRAIN) คือ 35 บอกว่าเวลาในการดำเนินการ35สัปดาห์ในการเริ่มการอบรมและใช้เวลาในการอบรมก่อนเสร็จสิ้นอีก 10สัปดาห์

การใช้ตัวแปรขอบเขตของฟังค์ชั่น (Using Variable Domain Function)นอกเหนือจากการชี้เฉพาะ อีกนัยหนึ่งตัวแปรในโปรแกรม LINGO เป็นตัวเลือกแบบไม่เป็นค่าติดลบและต่อเนื่องกัน โดยเฉพาะตัวแปรสามารถสมมุติฐานถึงค่าจริงจากศูนย์ถึงอนันต์ อีกหลายกรณีค่าขอบเขตที่กำหนดสำหรับตัวแปรอาจไม่สมควร ยกตัวอย่างคุณอาจต้องการตัวแปรที่สมมุติขึ้นเป็นค่าศูนย์ หรือต้องการจำกัดค่าจริงเป็นจำนวนเต็ม โปรแกรม LINGO เปิดช่องให้สำหรับเจ็ดลักษณะที่เป็นตัวแปรขอบเขตของฟังค์ชั่น ซึ่งยอมให้เราสามารถที่จะข้ามขอบเขตสมมุติฐานของตัวแปรได้ ชื่อของฟังค์ชั่นที่ใช้ทั้งหมดมีดังต่อไปนี้

@GIN จำกัดตัวแปรเป็นค่าจำนวนเต็ม

@BIN สร้างตัวแปรเป็นระบบเลขฐานสอง ตัวอย่างเช่น 0 หรือ 1

@FREE ยอมให้ตัวแปรแสดงค่าจริง, เป็นค่าบวก หรือ ลบ

@BND ขอบเขตของตัวแปรให้อยู่ภายใต้ขอบเขตที่จำกัด

@SOS กำหนดเซ็ทของตัวแปรระบบเลขฐานสองและพื้นที่ของข้อจำกัดบนค่าที่ กำหนด

@CARD กำหนดเซ็ทของตัวแปรระบบเลขฐานสองและพื้นที่ของข้อจำกัดบนขอบเขตสูงสุดของผลรวม

@SEMIC จำกัดตัวแปรเป็นอย่างใดอย่างหนึ่งระหว่างศูนย์หรือมากกว่าขอบเขตของข้อกำหนด

ตัวแปรที่เป็นจำนวนเต็ม (Integer Variable)โปรแกรม LINGO ให้ผู้ใช้งานสามารถกำหนดจำนวนเต็มได้สองประเภท คือ จำนวนเต็มธรรมดา และ จำนวนเลขฐานสอง ตัวแปรที่เป็นเลขจำนวนเต็มปกติแล้วจะกำหนดให้กับตัวเลขทั้งหมดส่วนตัวเลขที่เป็นเลขฐานสองเป็นความต้องการเพิ่มเพื่อกำหนดค่าที่เป็นศูนย์หรือหนึ่งอย่างใดอย่างหนึ่งเพื่อใช้งาน บางตัวแบบจะประกอบด้วยตัวแปรที่เป็นจำนวนเต็มหนึ่งหรือมากกว่า การใช้งานนั้นก็จะอ้างอิงไปยังตัวแบบโปรแกรมที่เป็นจำนวนเต็ม Integer Programming (IP)ในตัวแบบของหลายๆโครงการเราจะพบการตัดสินใจแบบที่มีลักษณะ ใช่และไม่ใช่ บางตัวอย่างก็จะเป็นในลักษณะว่าจะดำเนินการหรือไม่ดำเนินการ, เปิดหรือปิดโรงงาน, ส่งหรือไม่ส่งสินค้าจากโรงงานของผู้ส่งมอบรายหนึ่งไปยังลูกค้า, ก่อให้เกิดค่าใช้จ่ายตายตัวหรือไม่เกิด,ตัวแปรที่เป็นเลขฐานสองจะเป็นวิธีมาตรฐานในการตัดสินใจใช่หรือไม่ใช่

โปรแกรม LINGO ไม่สามารถดำเนินการได้ง่ายในการปัดค่าของการแสดงผลให้อยู่ในรูปจำนวนเต็ม การปัดค่าของคำตอบจะนำไปสู่ผลที่เป็นไปไม่ได้หรือได้ผลของความคุ้มค่าที่รองลงไป ในการทำความเข้าใจกรณีนี้แสดงให้เห็นได้โดยตัวแบบนี้

```
MAX = X;
X + Y = 25.5;
X <= Y;
```

การพิจารณาตัวแบบนี้สามารถที่จะนำไปสู่ผลลัพธ์ที่ว่า X=Y=12.75 สมมุติว่าเราต้องการคำตอบที่เหมาะสมของค่า X เป็นจำนวนเต็ม การปัดค่าของ X เป็นค่า 13นั้นจะทำให้ตัวแบบนี้เป็นไปไม่ได้เพราะว่าจะไม่มีค่าสำหรับ Y ที่เหมาะสมสำหรับข้อกำหนดทั้งสอง หรืออธิบายได้โดยกระจ่างก็คือผลลัพธ์ที่เหมาะสมของ X=12, Y=13.5 นั้นไม่เกิดผล หรือผลลัพธ์ที่เหมาะสมในตัวแบบขนาดใหญ่ซึ่งตัวแปรเป็นจำนวนเต็มหลายตัวนั้นโดยแท้จริงเป็นไปไม่ได้ ในการแก้ปัญหานี้โปรแกรม LINGO นำเสนอวิธีแบบอัลกอริทึมซึ่งเรียกว่า Branch-and-bound ซึ่งระบุ

ความสัมพันธ์ของตัวแปรที่เป็นจำนวนเต็มที่เชื่อถือได้เพื่อหาคำตอบดีที่สุดที่เป็นไปได้ยังตัวแบบของ IP เพราะเวลาการคำนวนที่พิเศษนี้ต้องการดำเนินการแบบอัลกอริทึม เกณฑ์ของปัญหาหลีกเลี่ยงที่จะใช้ตัวแปรที่เป็นจำนวนเต็มและมีการแนะนำเมื่อใดก็ตามที่เกิดกรณีเช่นนี้ขึ้นเนื่องจากเวลาในการคำนวนจะต้องใช้เพิ่มขึ้นโดยกระทันหันเมื่อได้ดำเนินการเพิ่มตัวแปรที่เป็นจำนวนเต็มเข้าไป ทำให้เข้าใจได้อย่างสม่ำเสมอถึงการถามของโปรแกรม LINGO สำหรับผลลัพธ์ที่อยู่ในรูปแบบจำนวนเต็มเมื่อค่าเศษส่วนมีน้อยมากหรือมิได้ใช้

ตัวแปรที่เป็นจำนวนเต็มธรรมดา (General Integer Variable) โปรแกรม LINGO สมมุติให้ตัวแปรทั้งหมดในตัวแบบเป็นแบบต่อเนื่องในหลายๆประโยชน์การใช้สอยค่าที่เป็นเศษส่วนไม่เป็นที่ต้องการ เราไม่สามารถที่จะจ้างคนสองส่วนสามคน หรือ ขายรถจักรยานยนต์ครึ่งคันได้ ในกรณีนี้เราต้องการใช้ฟังค์ชั่นตัวแปรที่เป็นจำนวนเต็มธรรมดาคือฟังค์ชั่น @GIN การสร้างความสัมพันธ์ของฟังค์ชั่น@GINคือ @GIN (Variable_name); หรือ @GIN (ชื่อของตัวแปร) ซึ่งชื่อของตัวแปรนี้คือสิ่งที่เราหวังให้เป็นจำนวนเต็มแบบธรรมดา ตัวอย่างการใช้งานฟังค์ชั่น @GIN เช่น

Example 1: `@GIN(X);`
makes the scalar variable *X* general integer,

Example 2: `@GIN(PRODUCE(5));`
makes the variable *PRODUCE(5)* general integer,

Example 3: `@FOR(DAYS(I): @GIN(START(I)));`
makes all the variables of the *START* attribute general integer.

ตัวอย่างของการใช้จำนวนเต็มแบบธรรมดานั้นคือเริ่มจากตัวแบบ CompuQuick Product-Mix หรือส่วนประสมผลิตภัณฑ์ในตัวอย่างที่1โดยเพิ่มเติมการใช้คำสั่ง@GINเข้าไปในตัวอย่างนี้จะแสดงให้เห็นการใช้งานฟังค์ชั่น@GINได้แบบเต็มรูปแบบ ซึ่งเริ่มต้นด้วยโปรแกรมตัวแบบของ Computer Quick นั้นได้ผลสำเร็จในการดำเนินการประกอบคอมพิวเตอร์ในไลน์ให้เหมาะสม มีการกำหนดให้ผลิตคอมพิวเตอร์สองชนิดในไลน์การผลิตทุกวัน เพื่อให้ได้การผลิตคอมพิวเตอร์ที่ได้ผลกำไรสูงสุดแต่มีการปรับเปลี่ยนข้อกำหนดของการประกอบ Computer รุ่น Standard ใหม่เป็นมีกำลังการผลิตสูงสุดที่ 103 เครื่องต่อวันเมื่อเขียนนรูปแบบโปรแกรมได้ใหม่ว่า

```
STANDARD <= 103;
```

นำข้อกำหนดนี้ใส่เข้าไปในตัวแบบ Computer Quick เดิมที่เรามีอยู่จะได้

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

! Constraints on the production line capacity;
STANDARD <= 103;
TURBO <= 120;

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

และเมื่อ Run โปรแกรมที่ได้ดำเนินการแก้ไขแล้วผลลัพธ์ที่ได้คือ

```
Global optimal solution found.
Objective value:                              14575.00
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        103.0000            0.000000
        TURBO        28.50000            0.000000

          Row    Slack or Surplus      Dual Price
            1         14575.00          1.000000
            2         0.000000          25.00000
            3         91.50000          0.000000
            4         0.000000          75.00000
```

ตัวเลขที่เหมาะสมใหม่สำหรับผลิตคอมพิวเตอร์รุ่น Turbo คือ 28.5 ซึ่งไม่เป็นจำนวนเต็ม เราควรเพิ่มฟังค์ชั่น@GIN ในการกำหนดค่าตัวแปรทั้งคอมพิวเตอร์รุ่น STANDARDและ TURBOให้เป็นจำนวนเต็ม โดยการแก้ไขตัวแบบให้เป็นดังนี้

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

! Constraints on the production line capacity;
STANDARD <= 103;
TURBO <= 120;

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

! Integer values only;
@GIN(STANDARD); @GIN(TURBO);
```

และเมื่อทำการ Run โปรแกรมอีกครั้งจะได้ผลลัพธ์ที่เปลี่ยนไปดังนี้

```
Global optimal solution found.
Objective value:                              14550.00
Objective bound:                              14550.00
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                             0

Model Class:                                      PILP

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

Total constraints:                   4
Nonlinear constraints:               0

Total nonzeros:                      6
Nonlinear nonzeros:                  0

    Variable           Value        Reduced Cost
    STANDARD        102.0000           -100.0000
       TURBO        29.00000           -150.0000

         Row    Slack or Surplus      Dual Price
           1        14550.00            1.000000
           2        1.000000            0.000000
           3        91.00000            0.000000
           4        0.000000            0.000000
```

ค่าที่ได้หลังจากที่กำหนด@GINแล้วนั้นก็จะได้ค่าที่เป็นจำนวนเต็มทั้งหมดตามที่ต้องการตัวแปรที่เป็นจำนวนเต็มระบบเลขฐานสอง (Binary Integer Variables) ตัวแปรที่เป็นจำนวนระบบเลขฐานสองหรือที่เราเรียกว่าตัวแปรเลข 1/0 (หนึ่งและศูนย์) เป็นกรณีพิเศษของจำนวนเต็มที่ความต้องการค่าศูนย์หรือหนึ่ง ใช้บ่อยในการที่ต้องการตัดสินใจของตัวแบบว่าใช่/หรือไม่ใช่ การสร้างความสัมพันธ์ของฟังค์ชั่นใช้ฟังค์ชั่น@BINคือ

@BIN(*variable_name*);

เมื่อ Variable Name คือตัวแปรที่คาดหวังให้เป็นระบบเลขฐานสอง ฟังค์ชั่น@BIN อาจจะใช้ในตัวแบบ และอย่างไรก็ตามโดยปกติก็ใส่ค่าของข้อกำหนดใน ฟังค์ชั่น@BINสามารถที่

ตรึงให้อยู่ได้ในการกำหนดฟังค์ชั่น@FOR ร่วมด้วยเพื่อให้ใช้งานได้ง่าย ตัวอย่างของคำสั่ง@BIN คือ

Example 1: `@BIN( X);`
makes the scalar variable, *X*, a binary integer,

Example 2: `@BIN( INCLUDE( 4));`
makes the variable *INCLUDE(4)* binary,

Example 3: `@FOR( ITEMS: @BIN( INCLUDE));`
makes all variables in the INCLUDE attribute binary.

ตัวอย่างที่ 4.6 การสร้างตัวแบบของจำนวนเต็มเลขฐานสองกรณีปัญหาของ Knapsack (Binary Integer Example-The Knapsack Problem.)

กรณีนี้คือตัวอย่างปัญหาของ Knapsack นั้นนับเป็นตัวอย่างที่ดีสำหรับกรณีศึกษาเรื่องการใช้ตัวแปรที่เป็นเลขฐานสอง ในตัวแบบของปัญหานี้คือเรามีกลุ่มของสิ่งของที่ต้องการจัดใส่เพื่อบรรจุเข้าไปในกระเป๋าบรรจุสัมภาระ โชคร้ายที่ว่ากระเป๋าสัมภาระของเรานั้นมีขอบเขตจำกัดในการที่จะต้องรองรับของที่จะใส่เข้าไปดังนั้นจึงไม่สามารถที่จะจัดใส่ของทั้งหมดเข้าไปได้ ทุกประเภทของสิ่งของนั้นมีคุณค่าของตัวเองและแน่นอนที่ต้องการจะนำมาใส่รวมกันในกระเป๋าสัมภาระนี้ปัญหาก็คือว่าต้องดำเนินการกำหนดกลุ่มย่อยของสิ่งของที่จะบรรจุลงในกระเป๋าสัมภาระเพื่อที่จะทำให้เกิดผลรวมคุณค่าของสิ่งของที่จะบรรจุได้มากที่สุดโดยไม่เกินต่อความสามารถของกระเป๋าที่บรรจุได้ ในตัวแบบของปัญหานี้สามารถที่จะดำเนินการปรับเปลี่ยนเพื่อประยุกต์ใช้ได้กับเรื่องการวางแผนกลยุทธ์, ต้นทุนของเงินทุน เป็นต้น

ในตัวแบบของปัญหาคือให้เราวางแผนเพื่อการไปปิคนิค และได้ทำการกำหนดรายละเอียดของสิ่งของที่เราต้องนำติดตัวเพื่อการปิคนิคด้วยซึ่งสิ่งของทุกๆสิ่งนั้นมีน้ำหนักสิ่งของๆตัวเองและน้ำหนักสิ่งของทั้งหมดที่ต้องการนำติดตัวไปนั้นต้องไม่เกิน 15 ปอนด์ ดังนั้นเราจึงต้องดำเนินการกำหนดค่าของความสำคัญของสิ่งของเป็นค่าตั้งแต่ 1ถึง10 ให้กับสิ่งของทั้งหมดเพื่อช่วยในการตัดสินใจและกำหนดสิ่งของที่จะต้องนำไปบรรจุลงในกระเป๋าสัมภาระโดยมีรายละเอียดทั้งหมดดังต่อไปนี้

Item	Weight	Rating
Ant Repellent	1	2
Beer	3	9
Blanket	4	3
Bratwurst	3	8
Brownies	3	10
Frisbee	1	6
Salad	5	4
Watermelon	10	10

ของทั้งหมดแยกออกเป็นรายการคือ ยาที่ใช้ไล่มด (Ant Repellent), เบียร์ (Beer), ผ้าห่ม(Blanket),ไส้กรอกเยอรมัน (Bratwurst), บราวนี่ย์ (Brownies), จานพลาสติก (Frisbee), ผักสลัด (Salad),แตงโม (Watermelon)

ในการเริ่มเขียนสูตร เราจะมีรายละเอียดประกอบไปด้วยเซ็ทเพียงเซ็ทเดียวในตัวแบบนี้ซึ่งคือสิ่งที่เราพิจารณาเพื่อที่จะบรรจุลงในกระเป๋าสัมภาระซึ่งเราพิจาณณาเป็น Primitive เซ็ทซึ่งกำหนดได้โดย

```
SETS:
  ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS
```

ซึ่งเรามีตัวแปรอยู่สามประเภทที่มีความสัมพันธ์ต่อกันนั่นก็คือ INCLUDE ,WEIGHT, RATING ประกอบอยู่ในเซ็ทเดียวกัน INCLUDEนั้นจะเป็นตัวแปรเลขฐานสองเพื่อใช้ในการกำหนดรายการเพื่อรวมลงในกระเป๋าสัมภาระ WEIGHT เอาไว้ใช้ใส่ค่าตัวแปรเรื่องน้ำหนักของรายการทั้งหมด ส่วนRATINGนั้นไว้ใส่ค่าน้ำหนักความสัมคัญของสิ่งของ ต่อไปเราต้องการสร้างโครงสร้างของรายละเอียดในตัวแบบเพื่อใส่ค่าสมาชิกของเซ็ทซึ่งประกอบด้วยรายละเอียด,น้ำหนักและ ความสำคัญโดยรายละเอียดทั้งหมดเขียนเป็นโปรแกรมดังนี้

```
DATA:
 ITEMS            WEIGHT RATING =
    ANT_REPEL          1      2
    BEER               3      9
    BLANKET            4      3
    BRATWURST          3      8
    BROWNIES           3     10
    FRISBEE            1      6
    SALAD              5      4
    WATERMELON        10     10;

    KNAPSACK_CAPACITY = 15;
ENDDATA
```

และเราได้กำหนดความสามารถในการบรรจุของสัมภาระสูงสุดลงไปในโปรแกรมด้วย ซึ่งจะทำให้รายละเอียดในโปรแกรมสมบูรณ์มากยิ่งขึ้น เมื่อรายละเอียดครบแล้วเราก็กำหนดฟังค์ชั่นของวัตถุประสงค์ก็จะสามารถสร้างตัวแบบนี้ได้เสร็จสิ้นดังรายละเอียดด้านล่าง

```
MODEL:

 SETS:
   ITEMS: INCLUDE, WEIGHT, RATING;
ENDSETS

DATA:
   ITEMS          WEIGHT RATING =
    ANT_REPEL          1      2
    BEER               3      9
    BLANKET            4      3
    BRATWURST          3      8
    BROWNIES           3     10
    FRISBEE            1      6
    SALAD              5      4
    WATERMELON        10     10;

   KNAPSACK_CAPACITY = 15;
ENDDATA

MAX = @SUM( ITEMS: RATING * INCLUDE);

@SUM( ITEMS: WEIGHT * INCLUDE) <=
 KNAPSACK_CAPACITY;

@FOR( ITEMS: @BIN( INCLUDE));

END
```

Model: KNAPSACK

เมื่อทำการ RUN โปรแกรมเสร็จเรียบร้อยแล้ว จะแสดงผลออกมาในลักษณะนี้

```
Global optimal solution found.
Objective value:                              38.00000
Objective bound:                              38.00000
Infeasibilities:                              0.000000
Extended solver steps:                               0
Total solver iterations:                             0

                     Variable           Value        Reduced Cost
           INCLUDE( ANT_REPEL)        1.000000           -2.000000
               INCLUDE( BEER)         1.000000           -9.000000
            INCLUDE( BLANKET)         1.000000           -3.000000
          INCLUDE( BRATWURST)         1.000000           -8.000000
           INCLUDE( BROWNIES)         1.000000           -10.00000
            INCLUDE( FRISBEE)         1.000000           -6.000000
              INCLUDE( SALAD)         0.000000           -4.000000
         INCLUDE( WATERMELON)         0.000000           -10.00000
```

Solution to KNAPSACK

รายละเอียดคือกระเป๋าสัมภาระนี้จะใส่ของได้เต็ม 15 ปอนด์ เราสามารถขนสัมภาระทุกสิ่งไปได้ยกเว้นสลัดและแตงโม อาหารกลางวันก็คือเบียร์, ไส้กรอกเยอรมัน และบราวนี่ย์ ซึ่งไม่ค่อยจะเป็นประโยชน์มากนักแต่ท้ายที่สุดก็จะทำให้เราพึงพอใจได้

ตัวอย่างที่ 4.7 การสร้างตัวแบบของจำนวนเต็มเลขฐานสองกรณีปัญหาของส่วนผสมของผลิตภัณฑ์กับต้นทุนคงที่ (Binary Integer Example- Product Mix with Fixed Cost.)

ในหลายๆสถานการณ์ของกิจกรรมเฉพาะที่ก่อให้เกิดต้นทุนคงที่ ตัวอย่างเช่นการเปิดบริษัท การผลิตสินค้า การจ่ายค่านายหน้า การทำเครื่องมือและทำไลน์การผลิตในตัวอย่างต่อไปนี้เราจะนำตัวแบบส่วนผสมของผลิตภัณฑ์คล้ายกับตัวอย่างของComputer Quick ที่ได้ทำผ่านมาแล้วมาปรับใช้การนำเสนอปัญหามีอยู่ว่า สมมุติให้เราเป็นผู้จัดการโรงงานผลิตเครื่องบินโดยเราต้องการหาผลลัพธ์ที่ดีที่สุดในการผลิตแบบผสมผสานระหว่างเครื่องบินหกรุ่นด้วยกันซึ่งประกอบด้วย Rocket,Meteor,Streak,Comet,Jet และ Biplane ทุกๆรุ่นเราทราบผลกำไรและต้นทุนคงที่ของผลิตภัณฑ์ ซึ่งแสดงให้ทราบดังตารางด้านล่าง

Plane	Profit	Setup
Rocket	30	35
Meteor	45	20
Streak	24	60
Comet	26	70
Jet	24	75
Biplane	30	30

เครื่องบินทุกลำต้องใช้วัสดุหกชนิดด้วยกันคือ เหล็ก,ทองแดง,พลาสติก,ยาง,แก้ว และสี ปริมาณวัสดุที่เครื่องบินแต่ละประเภทต้องการใช้ทั้งหมดแสดงดังตารางด้านล่าง

	Rocket	Meteor	Streak	Comet	Jet	Biplane	Available
Steel	1	4	0	4	2	1	800
Copper	4	5	3	0	1	0	1160
Plastic	0	3	8	0	1	0	1780
Rubber	2	0	1	2	1	5	1050
Glass	2	4	2	2	2	4	1360
Paint	1	4	1	4	3	4	1240

ปัญหาของตัวแบบนี้ก็คือให้หาส่วนผสมของผลิตภัณฑ์ที่ทำให้ได้กำไรสุทธิสูงสุดโดยจะต้องไม่เกินปริมาณของวัสดุแต่ละชนิดที่มีอยู่ สูตรในการคำนวน เราต้องการPrimitive เซ็ทอยู่สองอย่างในตัวแบบนี้ หนึ่งเพื่อเสนอโมเดลของเครื่องบิน และ อีกหนึ่งนำเสนอวัตถุดิบซึ่งสามารถเขียนออกมาเป็นรูปแบบได้ดังนี้

```
PLANES:
  PROFIT, SETUP, QUANTITY, BUILD;
RESOURCES: AVAILABLE;
```

เราเพิ่มการดำเนินการเรื่องคุณสมบัติของเซ็ทของเครื่องบิน

- *PROFIT* stores profit contribution for the plane,
- *SETUP* stores setup cost to begin producing the plane,
- *QUANTITY* a variable for quantity of planes to produce, and
- *BUILD* a binary variable, 1 if we produce the plane, else 0.

โดยรายละเอียดมีดังนี้ กำไร : เก็บค่ากำไรสะสมของเครื่องบิน เซ็ทอัพ : เก็บราคาค่าเซ็ทอัพในการเริ่มดำเนินการของเครื่องบิน ปริมาณ : ตัวแปรสำหรับปริมาณของเครื่องบินในการสร้าง และการสร้าง : ตัวแปรเลขฐานสอง 1 แทนค่าหากว่าเราดำเนินการสร้างเครื่องบินส่วนถ้าหากไม่ใช่ให้มีค่าเป็นศูนย์ เราต้องการสมมุติฐานโดยใช้ให้เซ็ท RESOURCE และ เซ็ท PLANEไขว้กัน โดยเราต้องการให้เซ็ทดังกล่าวกำหนดค่าของตัวแปร USAGE เพื่อกำหนดแหล่งที่ใช้ของเครื่องบินทุกเครื่อง เราเรียกว่า เซ็ทRXPหลังจากกำหนดได้แล้วข้อมูลจะเป็นดังนี้

```
SETS:
   PLANES:
     PROFIT, SETUP, QUANTITY, BUILD;
   RESOURCES: AVAILABLE;
   RXP( RESOURCES, PLANES): USAGE;
ENDSETS
```

ในรายละเอียดในข้อมูลของเรานั้น จะเริ่มต้นด้วยสมาชิกของเซ็ท PLANE และ RESOURCE ตามด้วยข้อมูล PROFIT, SETUP, AVAILABLE และUSAGE ดังนั้นเราจะได้ข้อมูลดังต่อไปนี้

```
DATA:
  PLANES     PROFIT  SETUP =
    ROCKET      30      35
    METEOR      45      20
    STREAK      24      60
    COMET       26      70
    JET         24      75
    BIPLANE     30      30;
  RESOURCES AVAILABLE =
    STEEL,800 COPPER,1160 PLASTIC,1780
    RUBBER,1050 GLASS,1360 PAINT,1240;
  USAGE =  1 4 0 4 2 0
           4 5 3 0 1 0
           0 3 8 0 1 0
           2 0 1 2 1 5
           2 4 2 2 2 4
           1 4 1 4 3 4;
ENDDATA
```

หลังจากนั้นเราก็กำหนดฟังค์ชั่นวัตถุประสงค์ โดยวัตถุประสงค์ของเรานั้นต้องการกำไรสูงสุด ในการคำนวนผลรวมกำไร ระยะเวลา การสร้างเครื่องบินแต่ละประเภท หักออกจากค่าเซ็ทอัพคูณกับ การสร้างตัวแปรของเลขฐานสอง โปรแกรมสามารถสร้างวัตุประสงค์ได้ดังนี้

```
MAX = @SUM( PLANES:
 PROFIT * QUANTITY - SETUP * BUILD);
```

สำหรับข้อกำหนดแรกของเซ็ทนี้เราต้องการความแน่ใจว่าวัตถุดิบจะต้องไม่ถูกส่งมอบเกินต่อความต้องการ โดยใช้ความหมายทางภาษาได้ว่า ทุกๆ Source ของ i ผลรวมทุกๆปริมาณของ Plane j สร้างโดยตัวคูณ โดยแหล่งที่ใช้ Source i กับ Plane j ต้องน้อยกว่าหรือเท่ากับแหล่งที่มาของวัสดุเท่าที่จะหาได้ของ Source i โดยเราแปลงเป็นภาษาทางโปรแกรมได้ว่า

```
@FOR( RESOURCES( I):
 @SUM( PLANES( J):
  USAGE( I, J) * QUANTITY( J)) <=
   AVAILABLE( I)
);
```

เซ็ทของข้อกำหนดตัวต่อไปก็คือ เราใช้ตัวแปรเลขฐานสองให้ชื่อว่า BUILD ในการนำเสนอหากเครื่องบินจะต้องสร้างขึ้น ดังนั้นเราสามารถกำหนดค่าใช้จ่ายคงที่สำหรับการสร้างเครื่องบินในฟังค์ชั่นวัตถุประสงค์ สิ่งที่ต้องการก็คือคุณลักษณะของข้อกำหนดในการผลักดัน BUILD (I) ให้เป็นค่า1เมื่อเราผลิตปริมาณของเครื่องบิน (I) ที่ไม่เท่ากับศูนย์ โดยกำหนดกังนี้

```
@FOR( PLANES:
 QUANTITY <= 400 * BUILD;
  @BIN( BUILD)
);
```

สูตรในการคำนวนทั้งหมดแสดงให้เห็นเป็นตัวอย่างดังรายละเอียดด้านล่าง

```
MODEL:
SETS:
   PLANES:
     PROFIT, SETUP, QUANTITY, BUILD;
   RESOURCES: AVAILABLE;
   RXP( RESOURCES, PLANES): USAGE;
ENDSETS
DATA:
   PLANES     PROFIT  SETUP =
     ROCKET      30      35
     METEOR      45      20
     STREAK      24      60
     COMET       26      70
     JET         24      75
     BIPLANE     30      30;
   RESOURCES AVAILABLE =
     STEEL,800 COPPER,1160 PLASTIC,1780
     RUBBER,1050 GLASS,1360 PAINT,1240;
   USAGE =  1 4 0 4 2 0
            4 5 3 0 1 0
            0 3 8 0 1 0
            2 0 1 2 1 5
            2 4 2 2 2 4
            1 4 1 4 3 4;
ENDDATA

MAX = @SUM( PLANES:
 PROFIT * QUANTITY - SETUP * BUILD);
@FOR( RESOURCES( I):
 @SUM( PLANES( J):
  USAGE( I, J) * QUANTITY( J)) <=
   AVAILABLE( I);
);
@FOR( PLANES:
  QUANTITY <= 400 * BUILD;
  @BIN( BUILD);
);
@FOR( PLANES: @GIN( QUANTITY));

END
```

Model: PRODMIX

ผลลัพธ์เมื่อทำการ RUN โปรแกรมจะแสดงออกมาดังนี้

```
Global optimal solution found.
Objective value:                              14764.00
Objective bound:                              14764.00
Infeasibilities:                              0.000000
Extended solver steps:                               7
Total solver iterations:                           296

                     Variable           Value        Reduced Cost
           QUANTITY( ROCKET)        96.00000           -30.00000
           QUANTITY( METEOR)        0.000000           -45.00000
           QUANTITY( STREAK)        195.0000           -24.00000
            QUANTITY( COMET)        0.000000           -26.00000
              QUANTITY( JET)        191.0000           -24.00000
         QUANTITY( BIPLANE)        94.00000           -30.00000
             BUILD( ROCKET)        1.000000            35.00000
             BUILD( METEOR)        0.000000            20.00000
             BUILD( STREAK)        1.000000            60.00000
              BUILD( COMET)        0.000000            70.00000
                BUILD( JET)        1.000000            75.00000
           BUILD( BIPLANE)        1.000000            30.00000
```

Solution to PRODMIX

จากคำตอบที่ได้เป็นที่แปลกใจอย่างมากที่เราทราบว่าผลลัพธ์ที่ได้นั้นเราจะไม่สามารถทำกำไรจาก Metiors ได้ทั้งที่มีกำไรมากต่อเครื่องมากที่สุด แต่ในทางตรงกันข้ามคำตอบที่ได้ให้ผลิต Rocket,Streak,Jet และ Biplane ถึงจะได้กำไรสูงสุดตามเราได้คาดการณ์ไว้ล่วงหน้าตัวแปร BUILD สำหรับทุกๆชนิดของเครื่องบินที่ถูกต้องเซ็ทเป็นค่า 1

ตัวแปรอิสระ Free Variables ด้วยค่าที่กำหนดไว้ในโปรแกรมผูกมัดว่าค่าตัวแปรที่ต่ำสุดคือศูนย์และมากที่สุดที่ค่าอนันต์คำสั่ง @FREEจะปรับค่าต่ำสุดที่ค่าศูนย์เป็นค่าติดลบ โครงสร้างของการใช้คำสั่งคือ

@FREE(*variable_name*);

โดยที่ตัวแปรที่ในวงเล็บก็คือตัวแปรที่เราต้องการให้เป็นอิสระ ตัวอย่างของการใช้@FREEมีแสดงให้เห็นดังตัวอย่างด่านล่าง

Example 1: `@FREE( X);`
makes the scalar variable, *X*, free,

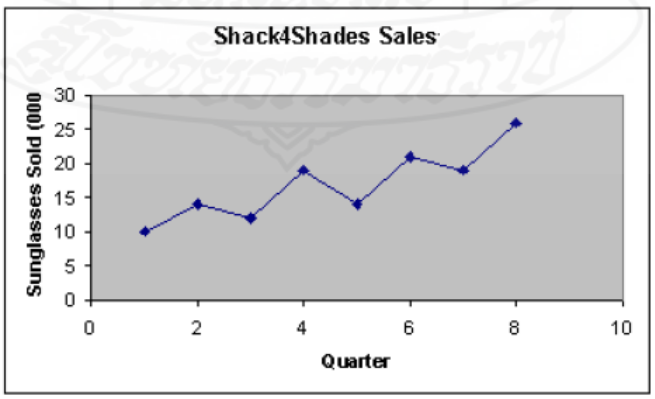
Example 2: `@FREE( QUANTITY( 4));`
makes the variable *QUANTITY(4)* free,

Example 3: `@FOR( ITEMS: @FREE( QUANTITY));`
makes all variables in the *QUANTITY* attribute free.

ตัวอย่างที่ 4.8 การสร้างตัวอย่างของตัวแปรอิสระเรื่องการพยากรณ์

(FREE Variable Example-Forecasting)

ในตัวอย่างสมมุติว่าเราเป็นผู้ควบคุมสินค้าคงคลัง ขายปลีก Shack4Shades ธุรกิจของเรานี้ค่อนข้างที่จะเฉพาะเจาะจงในการขายแว่นกันแดดให้กับสมาชิกที่รักการทำกิจกรรมกลางแจ้ง เราต้องการที่จะสร้างตัวแบบการพยากรณ์การขายของแว่นกันแดดที่กำลังจะมาถึงในช่วงควอเตอร์ถัดไปเพื่อใช้ในการกำหนดระดับของสินค้าคงคลัง เราได้สร้างแผนภูมิเพื่อแสดงให้เห็นถึงรายละเอียดของแปดควอเตอร์สุดท้ายดังรูป



ภาพที่ 4.8 กราฟแสดงรายละเอียดของการพยากรณ์ในช่วงควอเตอร์ต่อไป

ดูที่แผนภาพนี้เราคาดหวังว่าการขายของเรานั้นจะเติบโตขึ้นในสภาพที่มีแนวโน้มเป็นเส้นตรง แต่ว่าการประมาณค่อนข้างที่จะแปรปรวนไปในรูปแบบของฤดูกาล เช่นในหน้าร้อนเมื่อผู้คนต้องเดินทางไปชายหาด และในหน้าหนาวที่คนเดินทางเพื่อไปเล่นสกีว่าจะทำให้เกิดสภาวะการณ์เกิดขึ้นในลักษณะดังแผนภาพนี้ เรานำเสนอทฤษฎีของฟังค์ชั่นการประมาณการยอดขาย และ ฟังค์ชั่นของเวลาไว้ดังนี้

$$Predicted_Sales(t) = Seasonal_Factor(t) * (Base + Trend * t)$$

where,

Predicted_Sales(t)	represents predicted sales for quarter *t*,
Seasonal_Factor(t)	is one of four multipliers (one for each quarter of the year) to account for seasonal variations,
Base	is the *y*-intercept of the hypothesized linear function, and
Trend	is the slope of the linear function.

เมื่อ

Predicated_Sales(t)	เป็นการยืนยันการขายในควอเตอร์ t,
Seasona_Factor(t)	คือตัวคูณหนึ่งในสี่ส่วนของปีเพื่ออธิบายความแปรปรวนของฤดูกาลที่เกิดขึ้น
Base	คือตัวป้องกันสมมุติฐานของฟังค์ชั่น y
Trend	เป็นความชันของฟังค์ชั่นเส้นตรง

เมื่อเราต้องการดำเนินการใช้โปรแกรม LINGO ในตัวแบบข้างต้นนี้โดยเริ่มจากการกำหนดค่าตัวแปรหกตัวในฟังค์ชั่น (เช่น แฟคเตอร์ของฤดูกาลทั้งสี่, แนวโน้มพื้นฐาน,ความชันและแนวโน้มของความชัน) เพื่อที่จะดำเนินการในสิ่งนี้เราต้องเลือกค่าสำหรับตัวแปรเพื่อผลรวมน้อยที่สุดของพื้นที่แตกต่างระหว่างการคาดการณ์และยอดขายที่ได้ปฏิบัติเป็นข้อมูลเพื่อแสดงเรื่องราว การสร้างสูตร เราต้องการสองPrimitiveเซ็ทในตัวแบบของเรา ตัวแรกนั้นต้องประกอบด้วยสมาชิกแปดตัวด้วยกันเพื่อนำเสนอข้อมูลหนึ่งในสี่ส่วน ส่วนที่สองจะประกอบไปด้วยสมาชิกสี่ตัวที่มีลักษณะเดียวกันเพื่อนำเสนอสี่ส่วนในหนึ่งปี ในเซ็ทที่สองนี้ใช้เพื่อกำหนดแฟคเตอร์ของฤดูกาลทั้งสี่ทั้งหมดนี้คือรายละเอียดของเซ็ททั้งสอง

```
SETS:
   PERIODS: OBSERVED, PREDICT, ERROR;
   QUARTERS: SEASFAC;
ENDSETS
```

คุณสมบัติทั้งสามส่วนของ PERIODเซ็ท OBSERVED, PREDICTและERROR มีลักษณะที่ตรงกันในการพิจารณาคุณค่าของการขาย ทำนายปริมาณขาย และทำนายความผิดพลาด กล่าวง่ายๆคือการทำนายความผิดพลาดคือการทำนายการขายลบกับยอดขาย คุณสมบัติSEASFAC ในเซ็ทของ SEASONS มีลักษณะเดียวกันกับแฟกเตอร์ยอดขายตามฤดูกาลซึ่งจะคำนวนโดยโปรแกรม เราต้องการที่จะเพิ่มส่วนของข้อมูลเพื่อให้พร้อมที่จะทำงานของสมาชิกในเซ็ทและคุณสมบัติของOBSERVED กับข้อมูลด้านการขายเราสามารถดำเนินการโดยเขียนเป็นโปรแกรมดังนี้

```
DATA:
   PERIODS = P1..P8;
   QUARTERS = Q1..Q4;
   OBSERVED = 10 14 12 19 14 21 19 26;
ENDDATA
```

ต่อไปเราต้องเพิ่มสูตรในการคำนวนส่วนที่ผิดพลาด คือความแตกต่างระหว่างการคาดการณ์และยอดขายที่เกิดขึ้นจริงเราสามารถเขียนเป็นโปรแกรมได้ดังนี้

```
@FOR( PERIODS: ERROR =
 PREDICT - OBSERVED);
```

ต่อไปในส่วนของวัตถุประสงค์เราต้องการผลรวมที่น้อยที่สุดของพื้นที่ส่วนที่ผิดพลาด โดยเขียนเป็นฟังค์ชั่นวัตถุประสงค์ดังนี้

```
MIN = @SUM( PERIODS: ERROR ^ 2);
```

ในการสั่งให้คำนวนส่วนที่เกิดความผิดพลาด คือเราต้องการคำนวนยอดขายที่เกิดขึ้นโดยการใช้สูตรดังนี้

```
@FOR( PERIODS( P): PREDICT( P) =
 SEASFAC( @WRAP( P, 4))
  * ( BASE + P * TREND));
```

ฟังค์ชั่น @WRAPที่นำมาใช้นี้เพื่อประยุกต์ใช้สำหรับแฟกเตอร์ของฤดูกาลทั้งสี่เราทำได้โดยเพิ่มคุณสมบัติดังต่อไปนี้

```
@SUM( QUARTERS: SEASFAC) = 4;
```

สุดท้ายนี้เพื่อให้ส่วนที่เกิดความผิดพลาดที่เป็นค่าลบให้แสดงเป็นค่าบวก เราต้องใช้คำสั่ง ฟังค์ชั่น @FREE เพื่อยอมให้ส่วนของการผิดพลาดให้ไปเป็นค่าติดลบ โดยการเพิ่มฟังค์ชั่น @FREEที่ loop ของ@FOR โดยเขียนเป็นโปรแกรมดังนี้

```
@FOR( PERIODS: @FREE( ERROR));
```

ตัวแบบของโปรแกรมทั้งหมดที่เสร็จสมบูรณ์จะแสดงไว้ด้านล่าง

```
MODEL:
SETS:
   PERIODS: OBSERVED, PREDICT, ERROR;
   QUARTERS: SEASFAC;
ENDSETS

DATA:
   PERIODS = P1..P8;
   QUARTERS = Q1..Q4;
   OBSERVED = 10 14 12 19 14 21 19 26;
ENDDATA

MIN = @SUM( PERIODS: ERROR ^ 2);

@FOR( PERIODS: ERROR =
 PREDICT - OBSERVED);

@FOR( PERIODS( P): PREDICT( P) =
 SEASFAC( @WRAP( P, 4))
  * ( BASE + P * TREND));

@SUM( QUARTERS: SEASFAC) = 4;

@FOR( PERIODS: @FREE( ERROR);
 @BND( -1000, ERROR, 1000));

END
```

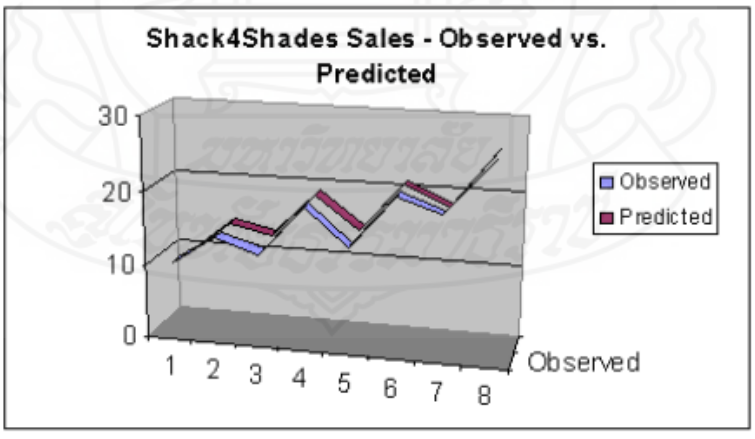
Model: SHADES

และเมื่อ Run โปรแกรมจะแสดงผลดังนี้

```
Local optimal solution found.
Objective value:                              1.822561
Total solver iterations:                            32

                     Variable           Value
                         BASE        9.718878
                        TREND        1.553017
               OBSERVED( P1)        10.00000
               OBSERVED( P2)        14.00000
               OBSERVED( P3)        12.00000
               OBSERVED( P4)        19.00000
               OBSERVED( P5)        14.00000
               OBSERVED( P6)        21.00000
               OBSERVED( P7)        19.00000
               OBSERVED( P8)        26.00000
                PREDICT( P1)        9.311820
                PREDICT( P2)        14.10136
                PREDICT( P3)        12.85213
                PREDICT( P4)        18.80620
                PREDICT( P5)        14.44367
                PREDICT( P6)        20.93171
                PREDICT( P7)        18.40496
                PREDICT( P8)        26.13943
                  ERROR( P1)       -0.6881796
                  ERROR( P2)        0.1013638
                  ERROR( P3)        0.8521268
                  ERROR( P4)       -0.1938024
                  ERROR( P5)        0.4436688
                  ERROR( P6)       -0.6828722E-01
                  ERROR( P7)       -0.5950374
                  ERROR( P8)        0.1394325
               SEASFAC( Q1)        0.8261096
               SEASFAC( Q2)         1.099529
               SEASFAC( Q3)        0.8938789
               SEASFAC( Q4)         1.180482
```

Solution to SHADES

เมื่อได้ข้อมูลแล้วสามารถแสดงออกมาเป็นกราฟได้ดังนี้



ในการใช้ฟังค์ชั่นนี้เราสามารถคำนวนการคาดการณ์ยอดขายในควอเตอร์ที่กำลังจะมาถึงได้ว่า

$$\begin{aligned} Predicted_Sales(9) &= Seasonal_Factor(1) * (Base + Trend * 9) \\ &= 0.826 * (9.72 + 1.55 * 9) \\ &= 19.55 \end{aligned}$$

ทั้งหมดนี้คือรายละเอียดของส่วนของเนื้อหาที่ผู้ทำการศึกษาได้ทำการศึกษาและได้ทำการเรียบเรียงเนื้อหาออกมาเป็นส่วนๆเพื่อให้ง่ายต่อการศึกษาค้นคว้าต่อไป

ตัวอย่างที่4.9 ตัวอย่างของตัวแบบที่นำไปประยุกต์ใช้งานจริงของแผนกวิศวกรรม บริษัท วาย เอส ภัณฑ์

ตัวอย่างปัญหาของตัวแบบมีอยู่ว่าทางบริษัทคาดการณ์ว่าต้องการผลิตสินค้าสองประเภทคือสินค้าAและBโดยที่ทั้งสินค้าประเภท AและBจะต้องผ่านกรรมวิธีการผลิตอยู่สามกระบวนการคือกระบวนการดัด, กระบวนการขึ้นรูป และ กระบวนการประกอบโดยคาดการณ์ว่าหากผลิตสินค้าประเภทAจะได้กำไรต่อหน่วย300บาทและผลิตสินค้าBได้กำไรต่อหน่วย400บาท โดยมีข้อกำหนดเรื่องเวลาที่ใช้ในแต่ละกระบวนการมีดังนี้

	ชิ้นงาน A	ชิ้นงาน B
กระบวนการดัด	1.5นาที	1นาที
กระบวนการขึ้นรูป	0.58นาที	0.58นาที
กรบวนการประกอบ	1นาที	2นาที

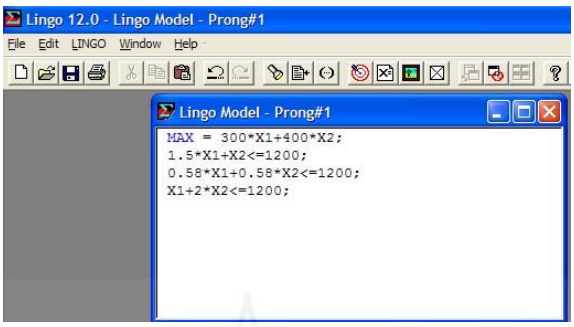
และในแต่ละกระบวนการผลิตมีข้อจำกัดเรื่องเวลาการผลิตรวมเท่ากันคือ 1200 นาที จากรายละเอียดดังกล่าวต้องการทราบการวางแผนการผลิตเพื่อให้ได้กำไรสูงสุด
จากข้อมูลดังกล่าวสามารถเขียนฟังค์ชั่นวัตถุประสงค์ได้ดังนี้

$$MAX = 300X1+400X2$$

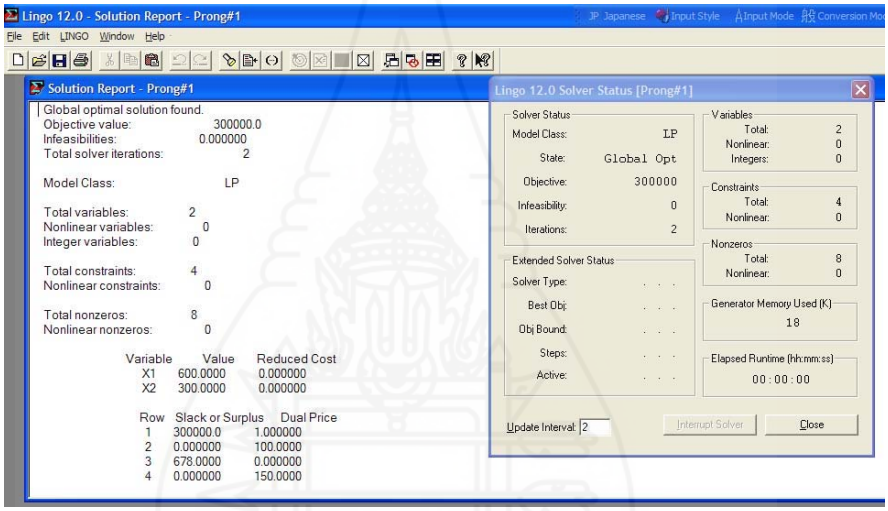
โดยที่X1 แทนผลิตภัณฑ์AและX2 แทนผลิตภัณฑ์B
ฟังค์ชั่นข้อกำหนดคือ

(กระบวนการดัด)	$1.5X1+X2 \leq 1200$
(กระบวนการขึ้นรูป)	$0.58X1+0.58X2 \leq 1200$
(กระบวนการประกอบ)	$X1+2X2 \leq 1200$

ทดลองนำไปเขียนในโปรแกรม LINGO ได้ดังตัวแบบด้านล่างนี้



เมื่อทดลอง RUN โปรแกรมแล้วแสดงผลออกมาได้ดังรายละเอียดดังนี้



จากตัวแบบปัญหาแสดงผลออกมาได้ว่า มีกำไรสูงสุดที่ 300,000 บาท โดยต้องทำการผลิตสินค้า Aที่600ชิ้น และสินค้าBที่ 300ชิ้นโดยข้อมูลในแถวที่2แสดงให้ทราบว่าเวลาของกระบวนการดัดถูกใช้จนหมดและถ้าหากเพิ่มเวลาให้แก่กระบวนการนี้ได้จะทำให้กำไรต่อหน่วยเพิ่มขึ้น 100บาท ส่วนในแถวที่3แสดงให้ทราบว่าเวลาของกระบวนการขึ้นรูปถูกใช้ไม่หมดยังคงเหลืออยู่678นาที และแถวสุดท้ายแสดงให้ทราบว่าเวลาของกระบวนการประกอบถูกใช้จนหมดและถ้าหากเพิ่มเวลาให้แก่กระบวนการนี้ได้จะทำให้กำไรต่อหน่วยเพิ่มขึ้น 150บาท

ส่วนที่2 เป็นส่วนของการประเมินคู่มือซึ่งได้ข้อมูลจากการให้กลุ่มตัวอย่างประเมินผลคู่มือหลังจากที่ได้ทดลองศึกษารายละเอียดของคู่มือที่ได้จัดทำขึ้นใหม่โดยมีเกณฑ์การให้คะแนนดังนี้

ระดับความพึงพอใจ	คะแนน
ดีมาก	4
ดี	3
พอใช้	2
ต้องปรับปรุง	1

เกณฑ์การประเมินค่าที่ใช้ประกอบกับการวัดระดับความพึงพอใจ คือ

3.26 - 4.0	=	พอใจมาก
2.51 – 3.25	=	พอใจ
1.76 – 2.50	=	พอใจปานกลาง
1.0 – 1.75	=	พอใจน้อย

การประเมินคู่มือ ประกอบไปด้วยส่วนต่างๆดังต่อไปนี้

7. ส่วนการวิเคราะห์ข้อมูลทั่วไปของคู่มือ

ตารางที่ 4.1 แสดงส่วนของผลการวิเคราะห์ข้อมูลทั่วไปของแบบประเมินคู่มือ

เพศ	ความถี่	ร้อยละ
ชาย	**12**	**100.0**
หญิง	**0**	**0.0**
รวม	**12**	**100.0**

จากตารางที่ 4.1 พบว่าผู้ตอบแบบประเมินคู่มือทั้งสิ้น 12 ราย พบว่าผู้ตอบแบบประเมินเป็นเพศชายทั้งหมด คิดเป็นร้อยละ 100.0

ตารางที่ 4.2 ส่วนของผลการวิเคราะห์ข้อมูลร้อยละของผู้ตอบแบบประเมินโดยจำแนกตามอายุ

อายุ	ความถี่	ร้อยละ
20-25	**1**	**8.3**
25-30	**4**	**33.3**
30-35	**7**	**58.4**
รวม	**12**	**100.0**

จากตารางที่ 4.2 พบว่าผู้ตอบแบบประเมินทั้งสิ้น 12 ราย พบว่าผู้ตอบแบบประเมินมากที่สุดมีอายุอยู่ในช่วง 30-35 ปี คิดเป็นร้อยละ 58.4 หรือเท่ากับ 7 คน ลำดับถัดมาเป็นระดับอายุ 25-30 ปี เท่ากับร้อยละ 33.4 คิดเป็นประชากรเท่ากับ 4 คน ลำดับรองลงมาเป็นระดับอายุ 20-25 ปี คิดเป็นร้อยละ 8.3 หรือเท่ากับ 1 คน

ตารางที่ 4.3 ส่วนของผลการวิเคราะห์ข้อมูลร้อยละของผู้ตอบแบบประเมิน
โดยจำแนกตามการศึกษา

อายุ	ความถี่	ร้อยละ
ปริญญาตรี	**12**	**100.0**
รวม	**12**	**100.0**

จากตารางที่ 4.3 พบว่าผู้ตอบแบบประเมินทั้งสิ้น 12 ราย เป็นผู้สำเร็จการศึกษาระดับปริญญาตรีทั้งหมด หรือคิดเป็นร้อยละ 100.0

8. ผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของเนื้อหาคู่มือ

ผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของเนื้อหาคู่มือซึ่งการแปลความหมายการประเมินคู่มือมีหลักเกณฑ์ดังต่อไปนี้คือ

เกณฑ์การประเมินค่าที่ใช้ประกอบกับการวัดระดับความพึงพอใจ คือ

3.26 - 4.0 = พอใจมาก

2.51 – 3.25 = พอใจ

1.76 – 2.50 = พอใจปานกลาง

1.0 – 1.75 = พอใจน้อย

ตารางที่ 4.4 แสดงผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของเนื้อหาคู่มือ

หัวข้อการประเมิน	ค่าเฉลี่ย	SD
1. ความน่าสนใจของคู่มือ	**3.1**	**0.3**
2. รายละเอียดและลำดับเนื้อหาของคู่มือ	**3.0**	**0.4**
3. ความสมบูรณ์ของเนื้อหา	**3.0**	**0.6**
4. ความยากง่ายของภาษาที่ใช้	**3.4**	**0.5**
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ	**3.0**	**0.6**

จากตารางที่ 4.4 พบว่าผู้ประเมินประเมินคู่มือเรื่องความน่าสนใจของคู่มือ มีค่าเฉลี่ยที่ 3.1 แสดงว่าคู่มืออยู่ในเกณฑ์การประเมินที่ระดับพอใจ ต่อมาในหัวข้อของรายละเอียดและลำดับเนื้อหาของคู่มือความสมบูรณ์ของเนื้อและประโยชน์ที่มีต่องานที่รับผิดชอบหามีค่าเฉลี่ยที่ 3.0 แสดงว่าอยู่ในเกณฑ์การประเมินที่ระดับพอใจ ส่วนความยากง่ายของภาษาที่ใช้มีค่าเฉลี่ยที่ 3.4 แสดงว่าอยู่ในเกณฑ์การประเมินที่ระดับพอใจมาก

ตารางที่ 4.5 แสดงผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของระดับความรู้

หัวข้อการประเมิน	ค่าเฉลี่ย	SD
1. ความรู้ก่อนการศึกษาคู่มือ	**1.8**	**0.5**
2. ความรู้หลังการศึกษาคู่มือ	**2.9**	**0.3**
3. การปรับใช้ในงานจริงของเนื้อหาคู่มือ	**3.2**	**0.4**

จากตารางที่ 4.5 พบว่าผู้ประเมินประเมินคู่มือเรื่องของระดับความรู้ก่อนการศึกษาคู่มือมีค่าเฉลี่ยที่ 1.8 แสดงว่ามีความรู้อยู่ในเกณฑ์ที่พอใจน้อย เมื่อได้ศึกษาคู่มือปรากฏว่าระดับความรู้และความพอใจเพิ่มขึ้นเป็น 2.9 คืออยู่ในระดับที่พอใจ และหัวข้อการนำไปปรับใช้ในงานจริงนั้นมีค่าเฉลี่ยที่ 3.2 แสดงว่าผู้ประเมินพอใจอย่างมากในการที่จะนำไปประยุกต์ใช้

9. ผลการวิเคราะห์ข้อมูลของผู้ตอบแบบประเมินในส่วนของข้อเสนอแนะ

จากส่วนรายละเอียดของข้อเสนอแนะที่ได้จากผู้ประเมินโดยรวมพบว่า

1. ผู้ที่ศึกษาคู่มือต้องการให้เพิ่มเติมตัวแบบของปัญหาให้มากขึ้นกว่าที่มีอยู่
2. ต้องการรูปแบบการนำเสนอที่เป็นแบบ Presentation เพื่อเพิ่มความเข้าใจ
3. อยากทดลองใช้ร่วมกับโปรแกรมจริงๆ

บทที่ 5
สรุปการวิจัย อภิปรายผล และข้อเสนอแนะ

การดำเนินการจัดทำคู่มือการใช้โปรแกรม LINGO เพื่อหาคำตอบที่เหมาะสมของโมเดลทางคณิตศาสตร์มีวัตถุประสงค์ในการดำเนินการจัดทำดังต่อไปนี้คือ

1) เพื่อจัดทำคู่มือของโปรแกรม LINGO และ รายละเอียดของโปรแกรมเพื่อนำไปใช้งาน

2) เพื่อประเมินผลคู่มือการใช้โปรแกรม LINGO โดยให้พนักงานของแผนกวิศวกรรมของบริษัท วาย เอส ภัณฑ์เป็นกลุ่มตัวอย่างในการประเมินผลคู่มือ

1. สรุปผลการศึกษาวิจัย

โดยเมื่อได้ทำการศึกษาแล้วนั้นสามารถที่จะจำแนกเนื้อหาออกมาได้เป็นส่วนของทฤษฎี ส่วนของคำสั่งของโปรแกรมที่ใช้ร่วมกับ WINDOW และสุดท้ายในส่วนของเนื้อหาซึ่งก็สามารถที่จะรวบรวมข้อมูลออกมาแสดงเป็นตัวแบบของปัญหาออกมาได้เป็น 8 ตัวแบบด้วยกัน ประกอบด้วย

1. ตัวแบบปัญหาเรื่องส่วนประสมของผลิตภัณฑ์
2. ตัวแบบปัญหาเรื่อง การขนส่ง
3. ตัวแบบปัญหาเรื่อง การวางแผนการทำงานของพนักงาน
4. ตัวแบบปัญหาเรื่อง พิกัดน้ำหนักส่วนผสมของเมล็ดพืช
5. ตัวแบบปัญหาเรื่อง การกำหนดกำลังคนโดยแสดงออกเป็นรายการ
6. ตัวแบบปัญหาเรื่องจำนวนเต็มเลขฐานสอง
7. ตัวแบบปัญหาเรื่องจำนวนเต็มเลขฐานสองกรณีส่วนผสมผลิตภัณฑ์
8. ตัวแบบปัญหาเรื่องตัวแปรอิสระของการพยากรณ์
9. ตัวแบบปัญหาในการนำมาประยุกต์ใช้งานจริงของแผนกวิศวกรรม บริษัท วาย เอส ภัณฑ์

จากการประเมินผลของคู่มือพบว่ามีผู้ประเมินคู่มือทั้งสิ้น 12 ราย โดยทั้งหมดเป็นเพศชาย มีการศึกษาระดับปริญญาตรีมีอายุอยู่ในช่วง 30-35 ปี คิดเป็นร้อยละ 58.4 หรือเท่ากับ 7 คน ลำดับถัดมาเป็นระดับอายุ 25-30 ปี เท่ากับร้อยละ 33.4 คิดเป็นประชากรเท่ากับ 4 คน ลำสุดท้ายเป็นระดับอายุ 20-25 ปี คิดเป็นร้อยละ 8.3 หรือเท่ากับ 1 คน พบว่าผู้ประเมินประเมินคู่มือเรื่องความ

น่าสนใจของคู่มือ มีค่าเฉลี่ยที่ 3.1 แสดงว่าคู่มืออยู่ในเกณฑ์การประเมินที่ระดับพอใจ ต่อมาในหัวข้อของรายละเอียดและลำดับเนื้อหาของคู่มือความสมบูรณ์ของเนื้อหาและประโยชน์ที่มีต่องานที่รับผิดชอบมีค่าเฉลี่ยที่ 3.0 แสดงว่าอยู่ในเกณฑ์การประเมินที่ระดับพอใจ ส่วนความยากง่ายของภาษาที่ใช้มีค่าเฉลี่ยที่ 3.4 แสดงว่าอยู่ในเกณฑ์การประเมินที่ระดับพอใจมาก ส่วนเรื่องของระดับความพอใจของความรู้ก่อนการศึกษาคู่มือมีค่าเฉลี่ยที่ 1.8 แสดงว่ามีความรู้อยู่ในเกณฑ์ที่พอใจน้อย เมื่อได้ศึกษาคู่มือปรากฏว่าระดับความรู้และความพอใจเพิ่มขึ้นเป็น 2.9 คืออยู่ในระดับที่พอใจ และหัวข้อการนำไปปรับใช้ในงานจริงนั้นมีค่าเฉลี่ยที่ 3.2 แสดงว่าผู้ประเมินพอใจอย่างมากในการที่จะนำไปประยุกต์ใช้

2. อภิปรายผล

จากการศึกษาเนื้อหาของโปรแกรม LINGO นั้นพบว่าเป็นโปรแกรมที่เชื่อมโยงเกี่ยวข้องต่อการช่วยในการตัดสินใจโดยมีข้อดีคือ สามารถที่จะลดเวลาที่จะต้องใช้การคำนวนหาคำตอบเนื่องจากเมื่อใช้จริงแล้วรายละเอียดของตัวแบบและรูปแบบของสมการที่สร้างขึ้นหากมีรายละเอียดมากจะทำให้การคำนวนทำได้ยากและใช้เวลานาน ดังนั้นหากเราได้ใช้โปรแกรมเพื่อหาคำตอบก็จะช่วยให้สะดวกรวดเร็วขึ้นได้ จากตัวอย่างของตัวแบบก็สามารถที่จะนำไปประยุกต์ใช้ในส่วนของการทำงานในบริษัทยกตัวอย่างเช่นการตัดสินใจในการผลิต การวางแผนการขนส่งเพื่อให้ได้ต้นทุนที่ต่ำสุดรวมถึงการวางแผนกำลังคน เป็นต้น เมื่อพิจารณาเพิ่มเติมจากเนื้อหาของคู่มือพบว่าผู้ที่จะใช้งานควรมีพื้นฐานเบื้องต้นโดยควรผ่านการศึกษาในรายวิชาการวิเคราะห์เชิงปริมาณ การวิจัยเชิงปฏิบัติการ หรือ การวิจัยดำเนินการมาแล้วจะทำให้การศึกษาเนื้อหาในคู่มือนั้นเข้าใจได้ง่ายขึ้น ส่วนเนื้อหาในคู่มือนั้นก็มีรายละเอียดของตัวอักษรที่มากควรที่จะนำเสนอเป็นรูปภาพเพิ่มขึ้นเพื่อประกอบความเข้าใจ

3. ข้อเสนอแนะ

เนื่องจากระยะเวลาและการดำเนินการศึกษาค้นคว้านั้นมีขอบเขตที่จำกัดจึงทำให้คู่มือที่รวบรวมมานี้นั้นมีเนื้อหาเป็นแต่เพียงพื้นฐานเพื่อให้ผู้สนใจที่จะทำการสามารถเข้าใจและทำงานได้สะดวกโดยที่รายละเอียดนั้นไม่ยากจนเกินไป ซึ่งอันที่จริงแล้วความสามารถของโปรแกรมยังมีอีกหลากหลาย เช่นสามารถทำงานและแสดงผลร่วมกับ โปรแกรม Microsoft Excel ได้อีกหลาย

รูปแบบ และ ด้วยขอบเขตที่กว้างของงานก็สามารถที่จะใช้ประยุกต์ใช้กับโปรแกรมเป้าหมาย Goal Programming ได้อีกด้วย

จากเนื้อหาในคู่มือนี้ผู้ที่จะใช้งานควรมีพื้นฐานเบื้องต้นโดยควรผ่านการศึกษาในรายวิชาการวิเคราะห์เชิงปริมาณ การวิจัยเชิงปฏิบัติการ หรือ การวิจัยดำเนินการมาแล้วจะทำให้การศึกษาเนื้อหาในคู่มือนั้นเข้าใจได้ง่ายขึ้น และ ผลจากการศึกษาคู่มือเห็นควรว่าต้องเพิ่มเติมตัวแบบของปัญหาให้มากขึ้นกว่าที่มีอยู่อีกทั้งต้องการการนำเสนอที่เป็นแบบ Presentation เพื่อเพิ่มความเข้าใจนอกเหนือจากการให้อ่านจากคู่มือที่เป็นตัวอักษรและรูปภาพประกอบ และ สุดท้ายหากได้ใช้งานโปรแกรมควบคู่กับการศึกษาคู่มือด้วยก็จะทำให้ใช้งานได้เร็วมากขึ้น

บรรณานุกรม

บรรณานุกรม

มหาวิทยาลัยสุโขทัยธรรมาธิราช (2551) ประมวลสาระชุดวิชาการวิเคราะห์เชิงปริมาณและ
การจัดการการดำเนินงาน นนทบุรี สำนักพิมพ์มหาวิทยาลัยสุโขทัยธรรมาธิราช

วิจิตร ตัณฑสุทธิ์,วันชัย ริจิรวนิช และ ศิริจันทร์ ทองประเสริฐ (2539) การวิจัยดำเนินงาน
กรุงเทพมหานคร บริษัทซีเอ็ดยูเคชั่น จำกัด

สิน พันธุ์พินิจ (2553) เทคนิคการวิจัยทางสังคมศาสตร์ กรุงเทพมหานคร บริษัทวิทยพัฒน์ จำกัด

Frederick S, Hiller.,Gerald J. Lieberman. (1995) Introduction to Operation Research.
New York:McGrew-Hill.

LINDO SYSTEM INC. (2010) LINGO USER'S GUIDE CHICAGO: LINDO SYSTEM INC.

ภาคผนวก

ภาคผนวก ก

ตัวอย่างของตัวแบบประเภทต่างๆ ที่มีเพิ่มเติมในโปรแกรม LINGO

Assembly Line Balancing — Model: ASLBAL

The following model illustrates how to balance an assembly line in order to minimize its total cycle time. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
 ! Assembly line balancing model;
   ! This model involves assigning tasks to
     stations in an assembly line so bottlenecks
     are avoided. Ideally, each station would be
     assigned an equal amount of work.;

 SETS:
   ! The set of tasks to be assigned are A through
     K, and each task has a time to complete, T;
   TASK/ A B C D E F G H I J K/: T;

   ! Some predecessor,successor pairings must be
     observed(e.g. A must be done before B, B
     before C, etc.);
   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 /;

   ! There are 4 workstations;
   STATION/1..4/;

   TXS(TASK, STATION): X;
   ! X is the attribute from the derived set TXS
      that represents the assignment. X(I,K) = 1
     if task I is assigned to station K;
 ENDSETS

DATA:
  ! Data taken from Chase and Aquilano, POM;
  ! Each task has an estimated time required:
       A  B  C  D  E  F  G  H  I  J  K;
  T = 45 11  9 50 15 12 12 12 12  8  9;
ENDDATA

 ! The model;
 ! *Warning* may be slow for more than 15 tasks;

 ! For each task, there must be one assigned
   station;
 @FOR(TASK(I): @SUM(STATION(K): X(I, K)) = 1);

 ! Precedence constraints;
 ! For each precedence pair, the predecessor task
   I cannot be assigned to a later station than
   its successor task J;
 @FOR(PRED(I, J):
  @SUM(STATION(K):
   K * X(J, K) - K * X(I, K)) >= 0);

 ! For each station, the total time for the
   assigned tasks must be less than the maximum
   cycle time, CYCTIME;
 @FOR(STATION(K):
  @SUM(TXS(I, K): T(I) * X(I, K)) <= CYCTIME);

 ! Minimize the maximum cycle time;
 MIN = CYCTIME;

 ! The X(I,J) assignment variables are binary integers;
 @FOR(TXS: @BIN(X));

END
```

Model: ASLBAL

Plant Location — Model: CAPLOC

In this example, we build a model to help decide what plants to open and how much product to ship from each plant to each customer. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
! Capacitated Plant Location Problem;
SETS:
   PLANTS: FCOST, CAP, OPEN;
   CUSTOMERS: DEM;
   ARCS( PLANTS, CUSTOMERS)  : COST, VOL;
ENDSETS

DATA:
 ! The plant, their fixed costs
   and capacity;
   PLANTS, FCOST, CAP =
     P1      91    39
     P2      70    35
     P3      24    31;
 ! Customers and their demands;
   CUSTOMERS, DEM =
     C1      15
     C2      17
     C3      22
     C4      12;
 ! The plant to cust cost/unit
   shipment matrix;
   COST =   6  2  6  7
            4  9  5  3
            8  8  1  5;
ENDDATA

! The objective;
   [TTL_COST] MIN = @SUM( ARCS: COST * VOL) +
    @SUM( PLANTS: FCOST * OPEN);

! The demand constraints;
   @FOR( CUSTOMERS( J): [DEMAND]
    @SUM( PLANTS( I): VOL( I, J)) >= DEM( J)
   );

! The supply constraints;
   @FOR( PLANTS( I): [SUPPLY]
    @SUM( CUSTOMERS( J): VOL( I, J)) <=
     CAP( I) * OPEN( I)
   );

! Make OPEN binary(0/1);
   @FOR( PLANTS: @BIN( OPEN));
END
```

Model: CAPLOC

Generating Random Numbers — Model: DEMRND

In cases where you are modeling under uncertainty, it is useful to have a random number generator to help simulate a situation. LINGO supports the *@RAND* function, which can be used to generate sequences of pseudo random numbers with uniform distribution between 0 and 1. By using the same seed value, you can recreate a series of numbers. In this example, we generate 15 random numbers, and then use them with the *@PSN* and *@PTD* functions to generate observations from both the unit Normal and T distributions.

```
MODEL:
! Generate a series of Normal and T distributed
  random variables ;
 SETS:
  SERIES/1..15/: U, ZNORM, ZT;
 ENDSETS

! First uniform is arbitrary;
  U(1) = @RAND(.1234);

! Generate the rest recursively;
  @FOR(SERIES(I)| I #GT# 1:
   U(I) = @RAND(U(I - 1))
  );

! Generate some...;
  @FOR(SERIES(I):
!   Normal deviates...;
   @PSN(ZNORM(I)) = U(I);
!   and t deviates(2 degrees of freedom);
   @PTD(2, ZT(I)) = U(I);
 !   ZNORM and ZT may take on negative values;
   @FREE(ZNORM(I)); @FREE(ZT(I));
  );
END
```

Model: DEMRND

Scenario-based Portfolio Model — Model: DNRISK

In this model, we are attempting to come up with an optimal portfolio that meets a certain level of return while minimizing downside risk. Downside risk is a measure of the risk of falling below our target return. An additional feature of this model is it is scenario-based. More specifically, we have seven scenarios that will each occur with a given probability. The model incorporates this distribution of predicted outcomes in deriving the optimal portfolio.

```
MODEL:     ! (dnrisk.lng);
! Downside risk portfolio model;
 SETS:
    ASSET: INVEST; ! Amount to invest in each asset;
    SCENARIO:
        TRETRN,   ! Return under this scenario;
        DRISK;    ! Downside risk under this scenario;
    TABLE( SCENARIO, ASSET):
        ARETRN;   ! Return in scenario I of asset J;
 ENDSETS

 DATA:
! Number of scenarios;
    SCENARIO = 1..7;

! The available assets in which to invest;
    ASSET = ATT  GMC  USX;
    ARETRN =
    -.071  .144  .169
     .056  .107 -.035
     .038  .321  .133
     .089  .305  .732
     .090  .195  .021
     .083  .390  .131
     .035 -.072  .006;

! Desired return;
    DRETURN = .13;

! Threshold, below which we are unhappy;
    THRESH = .11;

! Power to use for risk(1 or 2);
!  When NPOW = 1, it is a linear program;
!  When NPOW = 2 and threshold = desired return;
!   it is the semi-variance;
    NPOW = 2;
 ENDDATA

    NSCEN = @SIZE( SCENARIO);

! Minimize average downside risk;
    MIN =  @SUM( SCENARIO: DRISK ^ NPOW)/ NSCEN;

! Compute return for each scenario;
    @FOR( SCENARIO( I):
     TRETRN( I) = @SUM( ASSET( J):
      ARETRN( I, J) * INVEST( J));

! .. and how much we fall short of threshold ;
     DRISK( I) >= THRESH - TRETRN( I);

! Return in a period could be negative;
     @FREE( TRETRN( I));
        );

! Our budget constraint(divided by a billion);
   [BUDGET] @SUM( ASSET: INVEST ) = 1;

! Our desired return;
   [PRICER] @SUM( SCENARIO( I): TRETRN( I))/ NSCEN >= DRETURN;
END
```

Model: DNRISK

Dynamic Programming — Model: DYNAMB

Dynamic programming (DP) is a creative approach to problem solving that involves breaking a large, difficult problem into a series of smaller, easy to solve problems. By solving this series of smaller problems, we are able to assemble the optimal solution to the initial large problem. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
SETS:
 ! Dynamic programming illustration (see Anderson, Sweeney &
   Williams, An Intro to Mgt Science, 6th Ed.). We have a network of
   10 cities. We want to find the length of the shortest route from
   city 1 to city 10.;
 ! Here is our primitive set of ten cities, where F(i) represents the
   shortest path distance from city i to the last city;
  CITIES /1..10/: F;
 ! The derived set ROADS lists the roads that exist between the
   cities (note: not all city pairs are directly linked by a road,
   and roads are assumed to be one way.);
  ROADS(CITIES, CITIES)/
   1,2  1,3  1,4
   2,5  2,6  2,7
   3,5  3,6  3,7
   4,5  4,6
   5,8  5,9
   6,8  6,9
   7,8  7,9
   8,10
   9,10/: D;
 ! D(i, j) is the distance from city i to j;
ENDSETS
DATA:
 ! The distances corresponding to the links;
  D =
     1    5    2
    13   12   11
     6   10    4
    12   14
     3    9
     6    5
     8   10
     5
     2;
ENDDATA
!  If you are already in City 10, then the cost to travel to City 10
   is 0;
 F(@SIZE(CITIES)) = 0;
! The following is the classic dynamic programming recursion. In
   words, the shortest distance from City i to City 10 is the minimum
   over all cities j reachable from i of the sum of the distance from
   i to j plus the minimal distance from j to City 10;
 @FOR(CITIES(i)| i #LT# @SIZE(CITIES):
  F(i) = @MIN(ROADS(i, j): D(i, j) + F(j))
 );
END
```

Machine Repair Problem — Model: EZMREPAR

This model analyzes a queuing system with a fixed population of members that periodically require servicing. Although we model a computer timesharing system in this example, this model is typically referred to as the *machine repair problem*, because one can think of it as applying to a group of machines periodically requiring repair by a number of service personnel. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
  ! Model of a computer timesharing system;
  ! The mean think time for each user (more
    generally, Mean Time Between Failures in a
    repair system);
    MTBF = 40;

    ! The mean time to process each compute request
    (more generally, Mean Time To Repair in
    seconds);
    MTTR =  2;

    ! The number of users;
    NUSER = 32;

  ! The number of servers/repairmen;
    NREPR = 1;

  ! The mean number of users waiting or in service
    (more generally, the mean number of machines
    down);
    NDOWN =
     @PFS(MTTR * NUSER/ MTBF, NREPR, NUSER);

  ! The overall request for service rate (more
    generally, overall failure rate), FR, must
    satisfy;
    FR = (NUSER - NDOWN)/ MTBF;

  ! The mean time waiting for or in service (more
    generally, the mean time down), MTD, must
    satisfy;
    NDOWN = FR * MTD;
END
```

Model: EZMREPAR

Simple Queuing Example — Model: EZQUEUE

Given a queue with a certain arriving load and number of servers, the *@PEL* function determines the fraction of customers lost due to all servers being busy. In this example, we use *@PEL* to solve for the number of servers that limit customer loss to 5%.

```
MODEL:
! Arrival rate of customers/ hour;
   AR = 25;
! Service time per customer in minutes;
   STM = 6;
! Service time per customer in hours;
   STH = STM/ 60;
! Fraction customers finding all servers busy;
   FB = .05;
!The PEL function finds number of servers needed, NS;
   FB = @PEL(AR * STH, NS);
END
```

Model: EZQUEUE

Markowitz Portfolio Example — Model: GENPRT

In the March, 1952 issue of *Journal of Finance*, Harry M. Markowitz published an article titled *Portfolio Selection*. In the article, he demonstrates how to reduce the risk of asset portfolios by selecting assets whose values aren't highly correlated. The following model implements these ideas in constructing a simple portfolio with three assets. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
! GENPRT: Generic Markowitz portfolio;
 SETS:
  ASSET/1..3/: RATE, UB, X;
  COVMAT(ASSET, ASSET): V;
 ENDSETS

 DATA:
! The data;
!  Expected growth rate of each asset;
  RATE = 1.3   1.2  1.08;

!  Upper bound on investment in each;
  UB   = .75   .75   .75;

!  Covariance matrix;
  V    =   3     1   -.5
           1     2   -.4
         -.5   -.4     1;

!  Desired growth rate of portfolio;
  GROWTH = 1.12;
ENDDATA

! The model;
! Min the variance;
 [VAR] MIN = @SUM(COVMAT(I, J):
              V(I, J) * X(I) * X(J));

! Must be fully invested;
 [FULL] @SUM(ASSET: X) = 1;

! Upper bounds on each;
 @FOR(ASSET: @BND(0, X, UB));

! Desired value or return after 1 period;
 [RET] @SUM(ASSET: RATE * X) >= GROWTH;
END
```

Model: GENPRT

Job Shop Scheduling — Model: JOBSLT

In this model, there are six jobs that can be done on one machine. The machine can only work on one job at a time. Each job has a due date. If we can't complete a job by its due date, we do not take the job. Our objective is to maximize the total value of the jobs selected.

Although we have based this model on a job shop scenario, the basic principles should be applicable in any area where time is the limiting factor in deciding what projects to undertake.

```
MODEL:
 ! One machine job selection;
 SETS:

 !There are six jobs each of which has a Due Date,
   Processing Time, Value, and a flag variable Y
   indicating if the job has been selected.;
  JOB/1..6/: ! Each job has a...;
   DD,  ! Due date;
   PT,  ! Processing time;
   VAL, ! Value if job is selected;
   Y;   ! = 1 if job is selected, else 0;
 ENDSETS

 DATA:
 VAL = 9  2  4  2  4  6;
 DD =  9  3  6  5  7  2;
 PT =  5  2  4  3  1  2;
 ENDDATA

 ! Maximize the total value of the jobs taken;
 MAX = TVAL;
 TVAL = @SUM(JOB: VAL * Y);

   ! For the jobs we do, we do in due date order;
 @FOR(JOB(J):

   ! Only jobs with earlier due dates can
     precede job J, and jobs must be completed
     by their due dates;
   @SUM(JOB(I)| DD(I) #LT# DD(J) #OR#
    (DD(I) #EQ# DD(J) #AND# I #LE# J):
     PT(I) * Y(I)) <= DD(J);

  ! Make the Y's binary;
   @BIN(Y);
 );
END
```

Model: JOBSLT

Matching Model — Model: MATCHD

Pair-matching problems require a number of objects be grouped into pairs subject to some criteria. The objective may be to minimize cost, to group like objects, etc. As an example, the following matching model pairs workers into offices to minimize total incompatibilities between paired individuals. A detailed discussion of this model may be found in Chapter 2, *Using Sets*.

```
MODEL:
SETS:
   ANALYSTS / 1..8/;
   PAIRS(ANALYSTS, ANALYSTS) | &2 #GT# &1:
    RATING, MATCH;
ENDSETS

DATA:
   RATING =
      9  3  4  2  1  5  6
         1  7  3  5  2  1
            4  4  2  9  2
               1  5  5  2
                  8  7  6
                     2  3
                        4;
ENDDATA

MIN = @SUM(PAIRS(I, J):
 RATING(I, J) * MATCH(I, J));

@FOR(ANALYSTS(I):
 @SUM(PAIRS(J, K) | J #EQ# I #OR# K #EQ# I:
  MATCH(J, K)) = 1
);

@FOR(PAIRS(I, J): @BIN(MATCH(I, J)));
END
```

Model: MATCHD

Optimal Airline Overbooking I — Model: OBOOKO

Closely related to the newsboy problem (in a mathematical sense) is the airline-overbooking problem. Given that a certain percentage of fliers with reservations will not show up for a flight, airlines that don't overbook will be sending most planes up with empty seats. Assuming the penalty cost for overbooking is not too high, an airline that hopes to maximize revenue should overbook its flights. The following model determines the optimal number of reservations to allow on a flight, and assumes the number of no-shows on a flight has a binomial distribution.

```
MODEL:
!
 This overbooking model determines the number of
 reservations, M, to allow on a flight if the
 no-show distribution is binomial;
! Some available data ;
   N = 16;    ! total seats available;
   V = 225;   !Revenue from a sold seat;
   P = 100;   !Penalty for a turned down customer;
   Q = .04;   !Probability a customer is a no-show;
! The probability to turn down customers is
  @PBN(Q, M, M - N), therefore the corresponding
  expected loss due to imperfect information is:
  (V + P) * @PBN(Q, M, M - N), and we want the
  loss to equal the revenue V on the margin.  So,
  the break-even equation is:;
   (V + P) * @PBN(Q, M, M - N) = V;
! Note, you should round up if M is fractional;
END
```

Model: OBOOKO

Minimal Spanning Tree — Model: MSPAN

In the minimal spanning tree, we need to find a set of links (a tree) in a network that connects all cities. Furthermore, the sum of the distances over all the links in the tree should be minimized. Among other things, this application is useful in constructing communications networks at minimal cost.

It turns out that this becomes a very difficult problem to solve using optimization as the number of nodes grows. For large versions of this problem, the optimization techniques provided by LINGO are not the appropriate tool. One would be wise to pursue alternatives such as heuristics or dynamic programming.

```
MODEL:
!Given the number of nodes and the distance
 between them, finding the shortest total distance
 of links on the network to connect all the nodes
 is the classic problem called minimal spanning tree (MST).
 This model finds the (MST) connecting Atlanta,
 Chicago, Cincinnati, Houston, LA, and Montreal so
 that messages can be sent from Atlanta (base) to
 other cities through the network at minimum cost;
  SETS:
    CITY / 1.. 6/: U; ! U(I) = level of city I;
                      ! U(1) = 0;
    LINK(CITY, CITY):
          DIST,  ! The distance matrix;
            X;  ! X(I, J) = 1 if we use link I, J;
  ENDSETS

DATA:    ! Distance matrix need not be symmetric;
         ! However, city 1 is base of the tree;
      !to: Atl  Chi  Cin  Hou   LA  Mon ;
   DIST =   0  702  454  842 2396 1196  !from Atl;
          702    0  324 1093 2136  764  !from Chi;
          454  324    0 1137 2180  798  !from Cin;
          842 1093 1137    0 1616 1857  !from Hou;
         2396 2136 2180 1616    0 2900  !from LA;
         1196  764  798 1857 2900    0; !from Mon;
  ENDDATA

! The model size: Warning, may be slow for N >= 8;
   N = @SIZE(CITY);

   ! Minimize total distance of the links;
    MIN = @SUM(LINK: DIST * X);

   ! For city K, except the base, ... ;
    @FOR(CITY(K)| K #GT# 1:

     ! It must be entered;
       @SUM(CITY(I)| I #NE# K: X(I, K)) = 1;
```

```
  ! If there are 2 disjoint tours from 1 city to
    another, we can remove a link without
    breaking connections. Note: These are not
    very powerful for large problems;
    @FOR(CITY(J)| J #GT# 1 #AND# J #NE# K:
       U(J) >= U(K) + X (K, J) -
       (N - 2) * (1 - X(K, J)) +
       (N - 3) * X(J, K); );
 );

 ! There must be an arc out of city 1;
  @SUM(CITY(J)| J #GT# 1: X(1, J)) >= 1;

 ! Make the X's 0/1;
  @FOR(LINK: @BIN(X); );

 ! The level of a city except the base is at
   least 1 but no more than N-1, and is 1 if it
   links to the base;
   @FOR(CITY(K)| K #GT# 1:
      @BND(1, U(K), 999999);
      U(K) <= N - 1 - (N - 2) * X(1, K);   );
END
```

Model: MSPAN

Material Requirements Planning — Model: MRP

Material Requirements Planning, or MRP, is used to generate production schedules for the manufacture of complex products. MRP takes the demand schedule for a finished product and the lead times to produce the finished product and all the various subcomponents that go into the finished product, and then works backwards to come up with a detailed, just-in-time production schedule that meets the demand schedule. A detailed discussion of this model may be found in Chapter 12, *Developing More Advanced Models*.

```
MODEL:
! Data for this model is read from MRP.LDT;
SETS:
! The set of parts;
 PART: LT;
 ! LT(i) = Lead time to produce part i;

! The set of time periods;
 TIME;

! A relationship called USES between pairs of parts;
 USES( PART, PART): NEEDS;
 ! Parent part i needs NEEDS(i, j) units of
   child part j;

! For each part and time period we're interested in;
 PXT( PART, TIME): ED, TD;
 ! ED(i, j) = External demand for part i at time j;
 ! TD(i, j) = Total demand for part i at time j;
ENDSETS

DATA:

! Load the data from an external file;

 ! Parts list;
 PART = @FILE( 'MRP.LDT');

 ! Time periods;
 TIME = @FILE( 'MRP.LDT');

 ! Get the parent child relations and the
   number of parts required;
 USES, NEEDS = @FILE( 'MRP.LDT');

! Get the lead times from the file;
 LT  = @FILE( 'MRP.LDT');

! Get the external demands
  over time for each part;
 ED  = @FILE( 'MRP.LDT');
ENDDATA
! Set NP = no. of time periods in the problem;
 NP = @SIZE( TIME);

! For each part P and period T, the total demand =
  external demand + demand generated by parents
  one lead time in the future;
```

```
 @FOR( PXT( P, T) | T + LT( P) #LE# NP :
  TD( P, T) = ED( P, T + LT( P)) +
   @SUM( USES( P2, P): TD( P2, T + LT( P)) *
    NEEDS( P2, P));
 );

DATA:

! Display a table showing the production schedule;
 @TEXT() = ' The production schedule:';
 @TEXT() = @TABLE( TD);
ENDDATA
END
```

Model: MRP

Product-Mix with Setup Costs — Model: PRODMIX

In a product-mix model, the decision is how much of a number of different products should be produced to maximize total revenue. Each product competes for a number of scarce resources. In this example, we produce six different flying machines from six different raw materials. This model also has the feature that, should we produce a given product, we incur a fixed setup cost.

```
MODEL:
SETS:
   PLANES/ ROCKET, METEOR, STREAK,
    COMET, JET, BIPLANE /:
     PROFIT, SETUP, QUANTITY, BUILD;
   RESOURCES /STEEL, COPPER, PLASTIC,
    RUBBER, GLASS, PAINT/: AVAILABLE;
   RXP(RESOURCES, PLANES): USAGE;
ENDSETS
DATA:
   PROFIT SETUP =
     30     35
     45     20
     24     60
     26     70
     24     75
     30     30;
   AVAILABLE =
    800 1160 1780 1050 1360 1240;
   USAGE =  1 4 0 4 2 0
            4 5 3 0 1 0
            0 3 8 0 1 0
            2 0 1 2 1 5
            2 4 2 2 2 4
            1 4 1 4 3 4;
ENDDATA
MAX = @SUM(PLANES: PROFIT*QUANTITY - SETUP*BUILD);

@FOR(RESOURCES(I):
 @SUM(PLANES(J):
  USAGE(I,J) * QUANTITY(J)) <= AVAILABLE(I)
);
@FOR(PLANES:
 QUANTITY <= 400 * BUILD;
  @BIN(BUILD)
);
@FOR(PLANES:
 @GIN(QUANTITY)
);
END
```

Model: PRODMIX

ภาคผนวก ข

ตัวอย่างของแบบใบประเมินคู่มือ

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ☐ หญิง ☐

อายุ ..... ปี

จบการศึกษาระดับ .....

อาชีพ ..... ตำแหน่ง ..... วันที่ .....

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุ	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ					
2. รายละเอียดและลำดับเนื้อหาของคู่มือ					
3. ความสมบูรณ์ของเนื้อหา					
4. ความยากง่ายของภาษาที่ใช้					
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ					
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด					
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด					
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง					

ข้อเสนอแนะ .....

.....

.....

.....

.....

ภาคผนวก ค

เอกสารการประเมินคู่มือจากผู้ทำการประเมิน

ใบประเมินผลคู่มือ

ชื่อ-สกุล Mr. Sarawi maniklarn - เพศ ชาย ✓ หญิง

อายุ 39 ปี

จบการศึกษาระดับ Bachelor degree

อาชีพ Engineering ตำแหน่ง Sr. Eng. วันที่ 2/12/11

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินผลข้อใดให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ด้านของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ด้านของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด				X	
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ 1) ให้ส่งเสริมในการใช้คู่มือการใช้งาน โปรแกรม อีก

2) รายงานปัญหาและปรับปรุงโปรแกรมให้ดีขึ้นอีก

3) โปรแกรมที่ใช้งานได้ ยังไม่ครบตามหัวข้อ ควรพัฒนาโปรแกรมให้ครบ

4) การติดตั้งโปรแกรม ควรมีการตรวจสอบก่อนการติดตั้งโปรแกรม

5) ควรให้ โปรแกรมนี้ ติดตั้งลงเครื่องคอมพิวเตอร์ทุกเครื่องที่ใช้งาน Computer.

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ✓ / หญิง

อายุ 32 ปี

จบการศึกษาระดับ .....

อาชีพ ..... ตำแหน่ง Sr. Engr. วันที่ 2/12/11

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ .....

.....

.....

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล นาย เอกสิทธิ์ สีบุญเรือง เพศ ชาย ✓ หญิง

อายุ 33 ปี

จบการศึกษาระดับ ป.ตรี

อาชีพ วิศวกร ตำแหน่ง วิศวกร วันที่ 3 ธ.ค. 54

ค่าชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ			X		
3. ความสมบูรณ์ของเนื้อหา			X		
4. ความยากง่ายของภาษาที่ใช้	X				
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ			X		
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ ควรมีโครงสร้างรูปแบบมาตรฐาน ของแบบ

ทดสอบจริงได้ จึงจะนำไปใช้งานในรูปแบบต่างๆ

โดยควรมีแบบสำรวจให้ใช้งาน

ใบประเมินผลคู่มือ

ชื่อ-สกุล นาย ธนพัฒน์ โพธิ์วิหาร เพศ ชาย ✓ หญิง

อายุ 34 ปี

จบการศึกษาระดับ ปริญญาตรี

อาชีพ วิศวกร ตำแหน่ง Sr. Eng วันที่ 2/12/11

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ ควรเพิ่ม ตัวอย่าง ให้มากขึ้นอีก.

.....

.....

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล โสริศยา ศศิโภคุ เพศ ชาย ✓ หญิง

อายุ 26 ปี

จบการศึกษาระดับ ป.ตรี

อาชีพ ข้าราชการ/วิศวกร ตำแหน่ง Engineer. วันที่ 02/12/61

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความคิดเห็นใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ	X				
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง	X				

ข้อเสนอแนะ .....

.....

.....

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ✓ หญิง

อายุ 54 ปี

จบการศึกษาระดับ ป.ตรี

อาชีพ ..... ตำแหน่ง วิศวกร วันที่ ๑๔/๒๔/๖๑

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อสารประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา	X				
4. ความยากง่ายของภาษาที่ใช้	X				
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ .....

ควรมีตัวอย่างที่ใช้งานได้จริงในแต่ละขั้นตอน

ใบประเมินผลคู่มือ

ชื่อ-สกุล กัลยกร ทางสุวรรณ เพศ ชาย ✓ หญิง

อายุ 27 ปี

จบการศึกษาระดับ .....

อาชีพ ลูกจ้าง ตำแหน่ง วิศวกร วันที่ 2-12-12

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
[ส่วนของเนื้อหาในคู่มือ]					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา			X		
4. ความยากง่ายของภาษาที่ใช้	X				
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
[ส่วนของระดับความรู้]					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด				X	
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง	X				

ข้อเสนอแนะ ควรเพิ่มตัวอย่างในการใส่ค่าหรือสัญลักษณ์ที่มีใช้บ่อยๆ ในแต่ละเรียด ขั้นตอนในการใช้ Program ที่จำเป็นสำคัญในการนำไปใช้งานจริง และมีตัวอย่างที่หลากหลายในการออกแบบใช้งานโปรแกรม

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล ศิรกานุช [illegible] เพศ ชาย ✓ หญิง

อายุ 37 ปี

จบการศึกษาระดับ ม.ศว

อาชีพ ครูชำนาญการ ตำแหน่ง ปัจจุบัน วันที่ 3 - มี.ค. - 2561

ค่าชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา	X				
4. ความยากง่ายของภาษาที่ใช้	X				
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ	X				
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ .....

.....

.....

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล โกสิทธ์ แซ่ลิ้ม เพศ ชาย ✓ หญิง

อายุ 32 ปี

จบการศึกษาระดับ ม.ตรี

อาชีพ พนักงานบริษัท ตำแหน่ง Engineer วันที่ 2/12/54

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้ตรงมากที่สุด

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ ข้อเสนอ
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ - ควรมีเนื้อหาในส่วนของคู่มือการใช้งานโปรแกรม

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ☑ หญิง ☐

อายุ 26 ปี

จบการศึกษาระดับ ปริญญาตรี

อาชีพ ..... ตำแหน่ง ..... วันที่ .....

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		✓			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ	✓				
3. ความสมบูรณ์ของเนื้อหา		✓			
4. ความยากง่ายของภาษาที่ใช้	✓				
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ			✓		
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			✓		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		✓			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		✓			

ข้อเสนอแนะ .....

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ☑ หญิง ☐

อายุ ..... ปี

จบการศึกษาระดับ .....

อาชีพ ..... ตำแหน่ง ..... วันที่ .....

คำชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ ดีมาก	3/ ดี	2/ พอใช้	1/ ต้องปรับปรุง	เหตุผล/ คำแนะนำ
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและลำดับเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ	X				
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด			X		
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ .....

.....

.....

.....

.....

ใบประเมินผลคู่มือ

ชื่อ-สกุล ..... เพศ ชาย ✓ หญิง

อายุ 33 ปี

จบการศึกษาระดับ ปริญญาตรี

อาชีพ ..... ตำแหน่ง ..... วันที่ 02/Dec/2011

ค่าชี้แจง โปรดเขียนเครื่องหมาย (X) ลงในช่องการประเมินแต่ละข้อให้สมบูรณ์

หัวข้อการประเมิน	ระดับความพึงพอใจ				
	4/ดีมาก	3/ดี	2/พอใช้	1/ต้องปรับปรุง	เหตุผล/ Reason
(ส่วนของเนื้อหาในคู่มือ)					
1. ความน่าสนใจของคู่มือ		X			
2. รายละเอียดและคำสั่งเนื้อหาของคู่มือ		X			
3. ความสมบูรณ์ของเนื้อหา		X			
4. ความยากง่ายของภาษาที่ใช้		X			
5. ประโยชน์ที่มีต่องานที่รับผิดชอบ		X			
(ส่วนของระดับความรู้)					
1. ก่อนศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด				X	
2. หลังศึกษาคู่มือท่านมีความรู้ในเรื่องนี้ระดับใด		X			
3. เนื้อหาของคู่มือสามารถนำมาปรับใช้ในงานได้จริง		X			

ข้อเสนอแนะ ควรมี presentation ที่เข้าใจง่าย ใช้ภาษาอังกฤษ

ขอบคุณ

ประวัติผู้ศึกษา

ชื่อ	นายสมหวัง ปิ่นปกรณ์
วัน เดือน ปีเกิด	1 กุมภาพันธ์ 2520
สถานที่เกิด	กรุงเทพมหานครฯ
ประวัติการศึกษา	ปวส. วิศวกรรมเขียนแบบและออกแบบการผลิต สถาบันเทคโนโลยีราชมงคลวิทยาเขต ขอนแก่น พ.ศ. 2540
	คอบ. วิศวกรรมอุตสาหการ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี พ.ศ. 2542
สถานที่ทำงาน	บริษัท วาย เอส ภัณฑ์ จำกัด
ตำแหน่ง	ผู้จัดการแผนกเทคนิค