

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

คำนำ

รายงานการวิจัย เรื่อง การหาผลเฉลยของปัญหาขนส่งโดยคอมพิวเตอร์ (Computer Solution of Transportation Problem) ได้นำโปรแกรมคอมพิวเตอร์มาประยุกต์ เพื่อลดความยุ่งยากในการคำนวณหาผลเฉลยปัญหาขนส่งที่เป็นปัญหาการโปรแกรมเชิงเส้นขนาดใหญ่ ให้มีความรวดเร็วและมีประสิทธิภาพ ทั้งยังเป็นแนวทางในการหาผลเฉลยรูปแบบปัญหาอื่น ๆ ที่คล้ายคลึงกัน

ผู้จัดทำวิจัยขอขอบคุณ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้เงินอุดหนุนการวิจัยปีงบประมาณ 2540

รศ.อุบลวรรณ เงินวิจิตร



RCH
T
57.78
08858

เลขหมู่.....
เลขทะเบียน.....34691
วัน, เดือน, ปี 19 พ.ย. 2542



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ปัญหาการขนส่งเป็นรูปแบบหนึ่งของ การโปรแกรมเชิงเส้น จุดมุ่งหมายของปัญหาเพื่อจัดรายการให้มีค่าใช้จ่ายรวมในการขนส่งน้อยที่สุด ปัญหานี้ส่วนใหญ่จะประกอบด้วยตัวแปรตัดสินใจและเงื่อนไขมากมาย ซึ่งเกิดจากการกระจายสินค้าจากแหล่งต้นทางใด ๆ ไปยัง แหล่งปลายทางใด ๆ โดยต้องการให้ต้นทุนค่าขนส่งต่ำสุด

โครงการเรื่อง การหาผลเฉลยของปัญหาขนส่ง เป็นการนำโปรแกรมคอมพิวเตอร์ เข้ามาประยุกต์ใช้ในการคำนวณหาค่าผลเฉลยเบื้องต้น และผลเฉลยที่เหมาะสมที่สุดของปัญหาการขนส่ง ที่จำนวนแหล่งต้นทาง และ แหล่งปลายทาง ไม่เกิน 12 ตัวแปร เพื่ออำนวยความสะดวกและลดความยุ่งยากในการแก้ปัญหการขนส่งให้เป็นไปอย่างสะดวกรวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น โดยเฉพาะสำหรับปัญหาการขนส่งที่เป็นปัญหาการโปรแกรมเชิงเส้นขนาดใหญ่

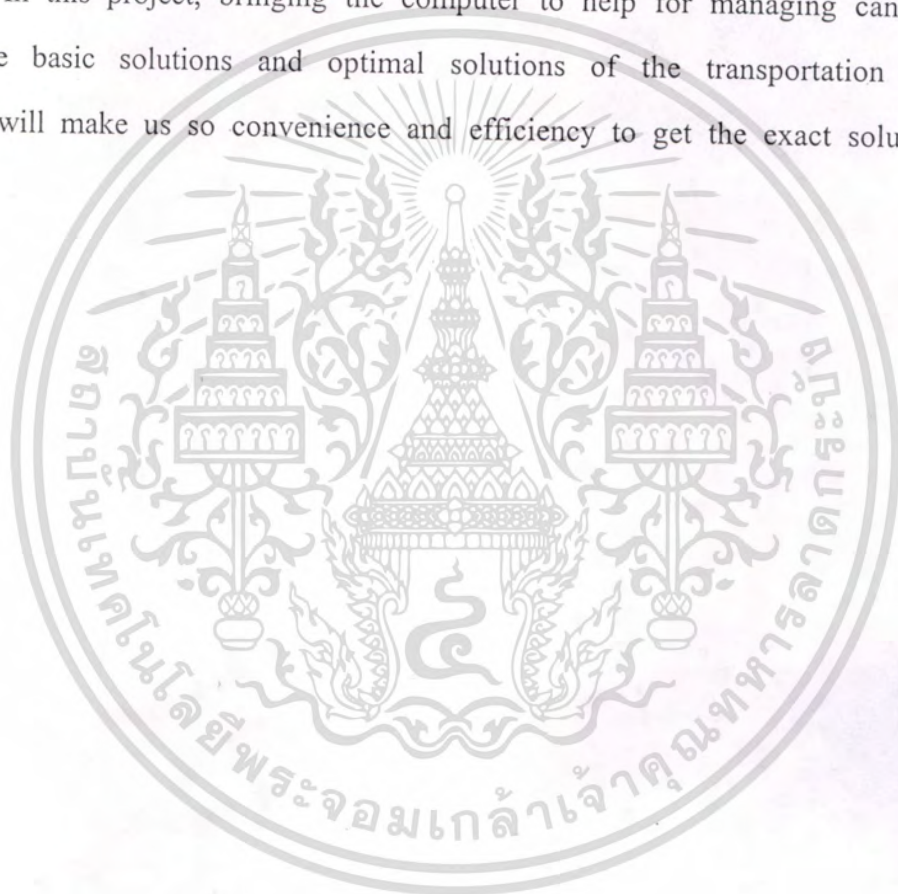


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

Transportation problem is one of Linear programming form. The purpose of this problem, to find solution that has the minimum sum of transport costs. There are many decision variables and constraints in this problem that happened by transport products from many sources to many destinations and to get the minimum value of transportation.

In this project, bringing the computer to help for managing can be find out the basic solutions and optimal solutions of the transportation problem which will make us so convenience and efficiency to get the exact solution.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงการ	1
1.2 หลักการและเหตุผล	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ	2
1.4 ขอบเขตของการทำงาน	2
1.5 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้องกับระบบการไหลผลเฉลยสำหรับปัญหาการขนส่ง	4
2.1 ตัวแบบการโปรแกรมเชิงเส้น	4
2.1.1 ประวัติความเป็นมา	5
2.1.2 ตัวอย่างการประยุกต์ใช้การโปรแกรมเชิงเส้น	6
2.1.3 ตัวอย่างแบบง่ายของการโปรแกรมเชิงเส้น	7
2.1.4 การสร้างตัวแบบปัญหาการโปรแกรมเชิงเส้น	9
2.1.5 รูปแบบมาตรฐานและรูปแบบอื่นของการโปรแกรมเชิงเส้น	11
2.1.6 ข้อสมมติพื้นฐานของปัญหาการโปรแกรมเชิงเส้น	13
2.2 ตัวแบบปัญหาการขนส่ง	14
2.2.1 การกำหนดตัวแบบทั่วไปของปัญหาการขนส่ง	15
2.2.2 ตัวอย่างของรูปแบบปัญหาการขนส่ง	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การหาผลเฉลยของรูปแบบปัญหาการขนส่ง	20
2.3.1 การหาผลเฉลยเบื้องต้น	20
2.3.1.1 วิธีตามกฎของมุมทิศเหนือ - ตก	21
2.3.1.2 วิธีตามกฎการประมาณค่าน้อยที่สุด	22
2.3.1.3 วิธีการประมาณค่าของโหนด	24
2.3.2 การหาผลเฉลยตามเป้าหมาย	26
2.3.2.1 วิธีสเตปปีงสโตน	26
2.3.2.2 วิธีโมโด	33
บทที่ 3 การใช้คอมพิวเตอร์ในการแก้ปัญหาการโปรแกรมเชิงเส้น	35
3.1 บทบาทของคอมพิวเตอร์	35
3.2 แนวความคิดเกี่ยวกับระบบงานคอมพิวเตอร์	36
3.2.1 การเก็บรวบรวมข้อมูลจากสนามและการประมาณค่าพารามิเตอร์	40
3.2.2 การสร้างตัวแบบ	40
3.2.3 การสร้างเมตริกซ์	40
3.2.4 การหาค่าผลเฉลยตามเป้าหมาย	41
3.2.5 การจัดทำรายงาน	41
3.2.6 การตีความรายงานและการตัดสินใจ	41
3.2.7 การนำไปปฏิบัติ	41
3.3 โปรแกรมสำเร็จรูปของการโปรแกรมเชิงเส้น	42
บทที่ 4 การออกแบบโปรแกรมระบบคอมพิวเตอร์สำหรับปัญหาการขนส่ง	44
4.1 การออกแบบฟลวชาร์ต	44
4.2 การออกแบบฟอร์มของโปรแกรม	52

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ปริมาณความต้องการวัตถุดิบเพื่อใช้ในการผลิตสีแต่ละวัน และกำไรในการผลิตสีแต่ละชนิด	8
ตารางที่ 2.2 ตัวอย่างปัญหาการโปรแกรมเชิงเส้น	10
ตารางที่ 2.3 รูปแบบปัญหาการคิดค่าขนส่งมาตรฐาน	17
ตารางที่ 2.4 ค่าใช้จ่ายในการขนส่ง ปริมาณการจัดส่งและปริมาณการจัดรับหินของแต่ละจังหวัด	18
ตารางที่ 2.5 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้ ด้วยวิธีตามกฎของมุมทิศเหนือ-ตก	22
ตารางที่ 2.6 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้ ด้วยวิธีตามกฎการประมาณค่าขนส่งน้อยที่สุด	23
ตารางที่ 2.7 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้ ด้วยวิธีการประมาณค่าของโวกเอล	25



สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงแหล่งผลิตและคลังสินค้า	15
รูปที่ 2.2 แสดงวิธีการหาวิถี (Path) ของตารางปัญหาการขนส่ง	27
รูปที่ 2.3 การพิจารณาผลเฉลยของวิถี	29
รูปที่ 2.4 การจัดสรร X_{ij} เพื่อให้ได้ผลเฉลยตามเป้าหมาย	31
รูปที่ 3.1 ลักษณะปฏิบัติงานในระบบงานแก้ปัญหาคอมพิวเตอร์โปรแกรมเชิงเส้น	38 - 39
รูปที่ 4.1 Flow Chart ของรูปแบบการแก้ปัญหาคอมพิวเตอร์โปรแกรมเชิงเส้น	45
รูปที่ 4.2 Flow Chart ของระบบคอมพิวเตอร์ในโปรแกรมการแก้ปัญหาคอมพิวเตอร์ขนส่ง	46
รูปที่ 4.3 Flow Chart ของการแก้ปัญหาคอมพิวเตอร์ด้วยวิธี North - West Corner Rule	47
รูปที่ 4.4 Flow Chart ของการแก้ปัญหาคอมพิวเตอร์ด้วยวิธี Lowest First Costs Rule Method	48
รูปที่ 4.5 Flow Chart ของการแก้ปัญหาคอมพิวเตอร์ด้วยวิธี Vogel 's Approximation Method	49
รูปที่ 4.6 Flow Chart ของการแก้ปัญหาคอมพิวเตอร์ด้วยวิธี Stepping Stone Method	50
รูปที่ 4.7 Flow Chart ของการแก้ปัญหาคอมพิวเตอร์ด้วยวิธี Modified Distribution Method	51
- Form : LOGO of Transportation Problem Program	53
- Form : Welcome to Transportation Problem Program	54
- Form : Menu of Transportation Problem Program	55
- Form : Instruction of Controls in Program	56
- Form : About Program and Programmer	57
- Form : Fill number of Source and Destination	58
- Form : Standard of Transportation Problem Program	59
- Form : Fill Transportation Value	60
- Form : รูปแบบการแก้ปัญหาคอมพิวเตอร์ขนส่ง (Transportation Solution)	61
- Form : รูปแบบการหาผลเฉลยเบื้องต้นของปัญหาคอมพิวเตอร์ขนส่ง	62
- Form : North - West Corner Rule	63.
- Form : Lowest First Costs Rule	64

- Form : Vogel 's Approximation Method	65
- Form : รูปแบบการหาผลเฉลยตามเป้าหมายของปัญหาการขนส่ง	66
- Form : Stepping Stone Method	67
- Form : Modified Distribution Method	68
- Form : ตารางผลเฉลยที่ดีที่สุด (Optimal Solution)	69
- Form : เมนู Option	70
- Form : Best Transportation Result	71
- Form : Please Check	72
- Form : พัฒนาการผลเฉลย	73
- Form : Choose Topic that you want to know	74
- Form : How to use this Program No.1	75
- Form : How to use this Program No.2	76
- Form : How to use this Program No.3	77
- Form : How to use this Program No.4	78



บทที่ 1

บทนำ

ในปัจจุบันเห็นได้ว่า คณิตศาสตร์เป็นศาสตร์แขนงหนึ่งที่สามารถนำไปใช้เป็นพื้นฐานในงานแขนงต่างๆ ได้ ตัวแบบทางคณิตศาสตร์ (Mathematical Model) ได้พิสูจน์ให้เห็นที่ยอมรับโดยทั่วไปว่ามีประโยชน์ สามารถประยุกต์ใช้กับปัญหาต่างๆ ได้ดี และกว้างขวางที่สุดตัวหนึ่งก็คือ ตัวแบบการโปรแกรมเชิงเส้น (Linear Programming Model) ในตัวแบบการโปรแกรมเชิงเส้นนี้ เราต้องการจะหาคำตอบที่ดีที่สุด (Optimal Solution) ของฟังก์ชันเป้าหมายเชิงเส้น การโปรแกรมเชิงเส้นเป็นรูปแบบทางคณิตศาสตร์ที่เกี่ยวข้องกับการจัดสรรทรัพยากรที่มีอยู่จำกัด ให้เกิดประโยชน์และมีประสิทธิภาพมากที่สุด เช่น เพื่อให้ต้นทุนหรือค่าใช้จ่ายน้อยที่สุด ลักษณะเฉพาะของการโปรแกรมเชิงเส้น คือ ฟังก์ชันที่แทนเป้าหมายและข้อจำกัดต่าง ๆ เป็นแบบเชิงเส้นทั้งสิ้น ซึ่งโดยมากใช้แก้ปัญหาในด้านอุตสาหกรรมที่มีทรัพยากรจำกัด

รูปแบบทั่วไปของการโปรแกรมเชิงเส้น เป็นรูปแบบซึ่งสามารถจะปรับใช้กับปัญหาต่างๆ ได้อย่างกว้างขวาง ทั้งนี้เพราะฟังก์ชันเป้าหมายอาจใช้ได้ในรูป กำไร ค่าใช้จ่าย การสูญเสีย เศษเหลือ จำนวนคนงานที่น้อยที่สุดที่จะจ้าง ฯลฯ ส่วนข้อจำกัดอาจจะอยู่ในรูปของข้อจำกัดของทรัพยากรที่มีอยู่หรือที่อาจจะหาได้ ข้อจำกัดด้านการตลาด ข้อจำกัดด้านคุณภาพ ฯ

ปัญหาการขนส่ง (Transportation Problem) เป็นรูปแบบหนึ่งของการโปรแกรมเชิงเส้น จุดมุ่งหมายของปัญหาเพื่อจัดรายการให้มีค่าใช้จ่ายในการขนส่งน้อยที่สุด ปัญหานี้ส่วนใหญ่จะประกอบด้วยตัวแปรตัดสินใจและเงื่อนไขมากมาย ปัญหาการขนส่งเป็นปัญหาที่เกิดจากการขนส่งวัตถุดิบจากหลายๆ แหล่ง ไปยังโรงงานที่มีอยู่หลายโรงงาน หรือ การกระจายสินค้าจากโรงงาน หรือ แหล่งผลิต (source) หลายแห่งนั้น ไปยังแหล่งเก็บสินค้า หรือ คลังสินค้าแหล่งต่าง ๆ เป็นจุดปลายทาง (destination) โดยต้องการให้ต้นทุนค่าขนส่งต่ำสุด

1.1 วัตถุประสงค์ของโครงการพิเศษ

- เพื่อศึกษาลักษณะของปัญหาการขนส่ง (Transportation Problem)
- เพื่อสร้างระบบคอมพิวเตอร์สำหรับการหาผลเฉลยของปัญหาการขนส่งลดความยุ่งยากทั้งเวลาในการคำนวณในการแก้ปัญหาในงานประเภทนี้
- เพื่อเป็นแนวทางในการหาผลเฉลยรูปแบบปัญหาอื่น ๆ ที่คล้ายคลึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 หลักการและเหตุผล

การหาคำตอบสำหรับปัญหาการโปรแกรมเชิงเส้น สามารถทำได้โดยอาจใช้วิธีการสำหรับปัญหาที่มีตัวแปรในการตัดสินใจไม่เกินสองตัวแปร หรืออาจหาได้โดยใช้วิธีพีชคณิต หรือ วิธีซิมเพล็กซ์ (Simplex Method) ในกรณีที่มีตัวแปรตั้งแต่สองตัวขึ้นไป นอกจากการหาคำตอบสำหรับปัญหาการโปรแกรมเชิงเส้นด้วยมือแล้ว เราสามารถนำระบบคอมพิวเตอร์เข้ามาช่วยในการแก้ปัญหาการโปรแกรมเชิงเส้นได้

โดยทั่วไปในการแก้ปัญหาการขนส่ง มีความยุ่งยากและต้องใช้เวลาในการแก้ปัญหาหรือหาผลเฉลยที่เหมาะสมที่สุดมากพอสมควรด้วยเหตุผลดังกล่าว จึงได้มีการนำคอมพิวเตอร์เข้ามาประยุกต์ใช้ในการแก้ปัญหาการขนส่ง เพื่ออำนวยความสะดวกในการแก้ปัญหาให้เป็นไปอย่างสะดวกรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น โดยเฉพาะสำหรับปัญหาการขนส่งที่เป็นปัญหาการโปรแกรมเชิงเส้นขนาดใหญ่

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถหาผลเฉลยของปัญหาการขนส่งได้อย่างรวดเร็วและถูกต้อง
- เพื่อเป็นการพัฒนาการหาผลเฉลยในปัญหาการขนส่งให้มีประสิทธิภาพ
- โปรแกรมสำเร็จรูปสำหรับการหาผลเฉลยในปัญหาการขนส่ง

1.4 ขอบเขตของการทำงาน

การทำโครงการพิเศษในครั้งนี้ เป็นการนำคอมพิวเตอร์เข้ามาช่วยในกระบวนการแก้ปัญหาเพื่อหาผลเฉลยของปัญหาการขนส่งเพื่อให้ได้ต้นทุนต่ำสุด และ การให้ผลกำไรสูงสุด โดยใช้รูปแบบวิธีการแก้ปัญหาดังนี้

1. การหาผลเฉลยเบื้องต้น โดย

- วิธีตามกฎของมุมทิศเหนือ - ตก (North - West Corner Rule)
- วิธีตามกฎการประมาณค่าน้อยที่สุด (Lowest First Costs Rule)
- วิธีการประมาณค่าของโวลเกิล (Vogel 's Approximation Method : VAM)

2. การหาผลลัพธ์เพื่อให้ได้ผลลัพธ์ตามเป้าหมาย

- วิธีสเตปปีงสโตน (Stepping Stone Method)
- วิธีโมได (Modified Distribution : MODI)

โดย จำนวนแหล่งผลิต (source) และจำนวนคลังสินค้า (destination) ไม่เกิน 10 แหล่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนในการดำเนินงาน

การทำโครงการพิเศษครั้งนี้ มีขั้นตอนในการดำเนินงานดังนี้

1. กำหนดรูปแบบของปัญหาที่จะศึกษา คือ ปัญหาการขนส่งเชิงเส้นทั้งหมดถึงรูปแบบวิธีการต่าง ๆ ที่ใช้ในการแก้ปัญหา และเปรียบเทียบผลลัพธ์ที่ได้จากวิธีการต่าง ๆ เพื่อเลือกรูปแบบวิธีการที่เหมาะสม
2. ศึกษาการใช้ Visual Basic for Windows เพื่อนำมาใช้ในการสร้างโปรแกรมคอมพิวเตอร์
3. ศึกษาทฤษฎีและนิยามต่าง ๆ ขั้นตอน และวิธีการในการแก้ปัญหาของรูปแบบที่ใช้วิธีการที่เลือกนำมาใช้อย่างละเอียด
4. ทำเอกสารเกี่ยวกับรูปแบบ วิธีการ และขอบเขตในการแก้ปัญหา
5. เขียนผังงานในการสร้างโปรแกรมคอมพิวเตอร์ (Flow Chart)
6. สร้างโปรแกรมคอมพิวเตอร์ตามผังงานที่วางไว้ในขั้นตอนที่ 5
7. ทำการทดสอบการทำงานของโปรแกรม
8. ทำเอกสารประกอบการใช้งานโปรแกรม

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับระบบการหาผลเฉลยสำหรับปัญหาการขนส่ง

2.1 ตัวแบบการโปรแกรมเชิงเส้น

(Linear Programming Models)

การโปรแกรมเชิงเส้น เป็นเทคนิคทางคณิตศาสตร์ประยุกต์ที่นำไปใช้ในการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัด ให้แก่กิจกรรมต่างๆที่แข่งขันกันเพื่อให้บรรลุเป้าหมายที่ก่อให้เกิดประโยชน์สูงสุด การโปรแกรมเชิงเส้นในปัจจุบัน ได้นำไปประยุกต์ใช้อย่างแพร่หลายในด้านอุตสาหกรรม ธุรกิจการเงิน การเกษตร เศรษฐกิจ และ การขนส่ง ปัญหาที่นำเทคนิคนี้ไปใช้อย่างมีประสิทธิภาพ ได้แก่ ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดในการผลิตสินค้าหลายชนิดของโรงงาน เพื่อให้ได้กำไรสูงสุด หรือมีค่าใช้จ่ายต่ำสุด ปัญหาการใช้ส่วนผสมของวัตถุดิบในการผสมอาหารสัตว์ ตามความต้องการในการเลี้ยงสัตว์ให้เจริญเติบโตตามเป้าหมาย โดยให้ค่าใช้จ่ายของต้นทุนอาหารสัตว์ต่ำสุด ปัญหาการขนส่งทางบกหรือทางน้ำหรือทางอากาศ เพื่อให้ค่าใช้จ่ายต่ำสุด ปัญหาการวางแผนการใช้ที่ดินให้เป็นประโยชน์ต่อการเพาะปลูกเพื่อให้ได้กำไรสูงสุด ปัญหาการจัดสรรเงินในการลงทุนกิจการต่างๆ เพื่อให้ได้กำไรสูงสุด

ความสำเร็จของการประยุกต์เทคนิคการโปรแกรมเชิงเส้นในรอบสามสิบกว่าปีที่ผ่านมามีผู้กล่าวว่าสามารถประหยัดเงินให้กับวงการอุตสาหกรรมและธุรกิจเป็นเงินหลายพันล้านเหรียญสหรัฐ และเป็นความก้าวหน้าทางคณิตศาสตร์ประยุกต์ที่สำคัญที่สุดก้าวหนึ่งที่เกิดขึ้นในช่วงกลางคริสต์ศตวรรษ ที่ 20

การโปรแกรมเชิงเส้น หมายถึง ตัวแบบ (Model) ทางคณิตศาสตร์เพื่อแสดงความสัมพันธ์ของกิจกรรมต่างๆ โดยใช้ตัวแปรทางคณิตศาสตร์แทนระดับกิจกรรมในการตัดสินใจ และพัฒนาเป็นตัวแบบคณิตศาสตร์ให้มีความสัมพันธ์ในรูปฟังก์ชันคณิตศาสตร์ ซึ่งฟังก์ชันคณิตศาสตร์นี้มีคุณสมบัติเป็นฟังก์ชันเชิงเส้น (Linear function)* ถ้าอยู่ในรูปเรขาคณิต 2 มิติจะเป็นเส้นตรง ถ้าอยู่ในรูปเรขาคณิต 3 มิติจะเป็นรูประนาบ

* แทนความหมายของฟังก์ชันเชิงเส้นในทางทฤษฎี

ฟังก์ชันเชิงเส้น $f(X)$ เป็นฟังก์ชันค่าจริง (real value) ที่นิยามบนเวกเตอร์สเปซ (vector space) n -มิติ (n -dimension)

2.1.1 ประวัติความเป็นมา

ปัญหาทั่วไปของการโปรแกรมเชิงเส้นนั้น มีการพัฒนาและนำมาประยุกต์ใช้อย่างเข้มข้น (intensive) ครั้งแรกเมื่อเดือนมิถุนายนปี ค.ศ.1947 โดยมีหัวหน้าคณะทำงาน คือ ยอร์ชบี แดนทซิก (George B. Dantzig) และ มาร์แชล วูด (Marshall Wood) พร้อมทั้งทีมร่วมงานในกองทัพอากาศสหรัฐอเมริกา ในช่วงนั้นคณะทำงานชุดนี้ได้รับมอบหมายให้วิเคราะห์ความเป็นไปได้ในการประยุกต์คณิตศาสตร์และเทคนิคที่เกี่ยวข้องต่อปัญหาการวางแผนและการวางโปรแกรมทางทหาร ภารกิจที่ได้รับมอบหมายนั้นทำให้แดนทซิกได้เสนอวิธีการดังนี้ “ ความสัมพันธ์ระหว่างกิจกรรมต่างๆ ขององค์กรขนาดใหญ่ เป็นตัวแบบการโปรแกรมเชิงเส้น ทำให้บรรลุเป้าหมายสูงสุด โดยทำให้ฟังก์ชันเชิงเส้นเป้าหมายมีค่าต่ำสุด ” เพื่อพัฒนาความคิดของ แดนทซิก ต่อมาในเดือนตุลาคม ค.ศ. 1948 กองทัพอากาศสหรัฐได้จัดตั้งคณะวิจัยคณะหนึ่งภายใต้ชื่อโครงการ สคูป (Scoop : Scientific Computation of Optimum Program) ซึ่งโครงการนี้ เป็นโครงการที่มีส่วนสำคัญยิ่งในการพัฒนาและประยุกต์ตัวแบบการโปรแกรมเชิงเส้นอย่างเป็นทางการ ในช่วงต้นการโปรแกรมเชิงเส้นส่วนใหญ่นำไปประยุกต์ใช้ในวงการทหาร และ ทางเศรษฐศาสตร์ ในตัวแบบ อินพุต - เอาท์พุต (input - output) ของลีออนทิฟ (Leontief) และปัญหาเกมส์ ต่อจากนั้นก็แพร่หลายในปัญหาทางด้านอุตสาหกรรม ด้านสังคม และ ปัญหาที่อยู่อาศัย

ในปี ค.ศ. 1947 ผู้เริ่มต้นสร้างตัวแบบทางคณิตศาสตร์ของปัญหาการโปรแกรมเชิงเส้น คือ แดนทซิก (Dantzig) พร้อมกันนั้นยังได้เสนอวิธีการที่เป็นระบบในการแก้ปัญหาการโปรแกรมเชิงเส้น เรียกว่า วิธีซิมเพล็กซ์ (Simplex Method) แต่ก่อนหน้านั้นในปี ค.ศ. 1939 นักคณิตศาสตร์ชาวรัสเซีย ชื่อ แอล วี แคนโตโลวิช (L. V. Kantorovich) ได้เสนอตัวแบบและการแก้ปัญหโปรแกรมเชิงเส้นเกี่ยวกับปัญหาขององค์กรและปัญหาการวางแผนการผลิต ในปี ค.ศ. 1941 ฮิทช์คอค (Hitchcock) ได้เสนอการแก้ปัญหเดียวกัน และในปี ค.ศ. 1945 สติกลเลอร์ (Stigler) ได้เสนอวิธีการแก้ปัญหทางโภชนาการ

การนำเครื่องคอมพิวเตอร์มาใช้ในการแก้ปัญหตัวแบบการโปรแกรมเชิงเส้นได้ประสบความสำเร็จครั้งแรกที่องค์กร National Bureau of Standard ของสหรัฐอเมริกา เมื่อเดือนมกราคม ค.ศ. 1952 ตั้งแต่นั้นการแก้ปัญหการโปรแกรมเชิงเส้นโดยวิธี ซิมเพล็กซ์ ได้พัฒนาเป็นโปรแกรม (Computer Program หรือ Software) ใช้ในเครื่องคอมพิวเตอร์ จนกระทั่งปัจจุบัน มีโปรแกรมสำเร็จรูป (Software package) ที่สามารถนำไปใช้ในเครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer หรือ Microcomputer) ได้ เช่น โปรแกรมสำเร็จรูปลินโด (Lindo) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี ค.ศ. 1984 ได้เป็นจุดเริ่มต้นที่สำคัญอีกก้าวหนึ่งของวิชาโปรแกรมเชิงเส้น เมื่อ นารินดา คาร์มาร์คาร์ (Narendra Karmarkar) ได้เสนอผลงานวิจัยใหม่ทางด้านทฤษฎีการโปรแกรมเชิงเส้น ซึ่งเป็นวิธีการใหม่ในการแก้ปัญหาการโปรแกรมเชิงเส้นเรื่อง “ A New Polynomial - time Algorithm for Linear Programming ” (ในวารสาร Combinatorica 4 หน้า 373 - 395 ค.ศ. 1984) วิธีการของ เอ็น คาร์มาร์คาร์ (N. Karmarkar 's Algorithm) สามารถนำไปใช้ในการหาค่าผลลัพธ์ของปัญหาการโปรแกรมเชิงเส้นขนาดใหญ่ได้ผลดี ในมหาวิทยาลัยบางแห่งได้เริ่มบรรจุเป็นเนื้อหาหลักสูตรในการสอนนักศึกษา แต่ยังไม่บรรลุผลสำเร็จเท่าที่ควร เนื่องจากความยากในการสอน และสมาคมการวิจัยดำเนินงานแห่งสหรัฐอเมริกา ได้จัดให้มีการอบรมเชิงปฏิบัติการในวิธีการ คาร์มาร์คาร์ (Karmarkar Algorithm) เมื่อเดือนตุลาคม ค.ศ. 1987 ผลของการประชุมปฏิบัติการครั้งนั้น ได้อภิปรายถึงวิธีการของคาร์มาร์คาร์ที่ยากในการสอน ว่าเป็นเพราะเป็นวิธีการพิสูจน์ที่ยุ่งยากและซับซ้อนมาก และเป็นวิธีการที่แตกต่างไปจากวิธีซิมเพล็กซ์เป็นอย่างมาก นอกจากนี้ถ้าจะสร้างความเข้าใจในรูปเรขาคณิตต้องเขียนรูป 3 มิติ ไม่สามารถเขียนรูปเป็น 2 มิติได้ และ ถ้าเป็นปัญหาขนาดเล็กจะเสียเวลามากในการคำนวณ ต่างจากวิธีซิมเพล็กซ์ที่คำนวณหาผลลัพธ์ด้วยมือได้ แต่ถ้าเป็นปัญหาขนาดใหญ่ วิธีของคาร์มาร์คาร์สามารถคำนวณหาผลลัพธ์ได้ดี

2.1.2 ตัวอย่างการประยุกต์ใช้การโปรแกรมเชิงเส้น

ตัวอย่างของการประยุกต์ใช้เทคนิคโปรแกรมเชิงเส้นที่พบเห็นกันโดยทั่วไป ได้แก่

1. ปัญหาการผลิต (Product Mix Problem) เป็นการพิจารณาหาปริมาณสินค้าแต่ละประเภทที่ควรจะทำการผลิต เพื่อให้การใช้ทรัพยากรที่มีอยู่ไม่ว่าจะจะเป็น เครื่องจักร วัตถุดิบ แรงงาน นั้นเป็นไปอย่างเหมาะสม เพื่อที่จะได้ผลตอบแทนสูงสุด

2. ปัญหาการผสมสาร (Blending Problem) เป็นการพิจารณาหาปริมาณสารหรือวัตถุดิบชนิดต่างๆ ที่จะนำมาผสมกันหรือนำมาใช้ในการผลิตสินค้าประเภทต่างๆ ตามคุณสมบัติที่ได้กำหนดไว้ เพื่อที่จะได้มีค่าใช้จ่ายต่ำสุด

3. ปัญหาการขนส่ง (Transportation Problem) เป็นการพิจารณาหาปริมาณสินค้าที่จะทำการขนส่งจากแหล่งผลิตสินค้าไปยังผู้บริโภคหรือจุดหมายปลายทาง เพื่อให้ค่าใช้จ่ายในการขนส่งต่ำสุด หรือส่งสินค้าถึงจุดหมายปลายทางได้เร็วที่สุด

4. ปัญหาการมอบหมายงาน (Assignment Problem) เป็นการพิจารณาการมอบหมายงานที่จะต้องทำให้กับบุคลากรหรือเครื่องจักร เพื่อให้งานที่ได้รับการมอบหมายแล้วเสร็จในเวลาเร็วที่สุด หรือเสียค่าใช้จ่ายน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ปัญหาการลงทุน (Investment Project Selection) เป็นการพิจารณาจัดสรรเงินลงทุนต่าง ๆ เพื่อให้ได้รับผลตอบแทนสูงที่สุด

6. ปัญหาการเลือกสื่อโฆษณา (Media Selection) เป็นการพิจารณาเลือกสื่อโฆษณาชนิดต่าง ๆ เพื่อให้ข้อมูลหรือข่าวสารที่ต้องการเผยแพร่ออกสู่เป้าหมายเป็นจำนวนมากที่สุด หรือโดยเสียค่าใช้จ่ายในการโฆษณาน้อยที่สุด

7. ปัญหาการตัดกระดาษ (Trim Loss Problem) เป็นการพิจารณารูปแบบหรือวิธีการตัดกระดาษ (หรือ สินค้าอื่น ๆ ที่มีคุณสมบัติคล้ายคลึงกัน เช่น การตัดผ้า การตัดแผ่นเหล็ก) เพื่อที่จะหารูปแบบการตัดกระดาษขนาดมาตรฐานที่มีอยู่ออกเป็นขนาดและปริมาณต่างๆ ตามความต้องการเพื่อที่จะให้มีเศษ (ส่วนที่ใช้ประโยชน์ต่อไปไม่ได้) น้อยที่สุด

8. ปัญหาทางด้านกรทหาร อาจนำเอาการโปรแกรมเชิงเส้นมาใช้ในการพิจารณาวางแผนการส่งกำลังบำรุง การเลือกกำหนดจำนวนอาวุธยุทโธปกรณ์ การกำหนดยุทธศาสตร์ ฯ

2.1.3 ตัวอย่างแบบง่ายของการโปรแกรมเชิงเส้น

ตัวอย่างแบบง่ายปัญหาการโปรแกรมเชิงเส้น

ตัวอย่างที่ 2.1

บริษัท สหชัย จำกัด เป็นบริษัทผู้ผลิตสีทาบ้าน โดยมีโรงงานขนาดเล็กทำการผลิตสี 2 ชนิดด้วยกัน คือ สีทาภายในและสีทาภายนอก วัตถุดิบที่ใช้ในการผลิตสีมี 2 ประเภท คือ สารละลายสีและสารเติมแต่งสี ในการผลิตสี แต่ละวันบริษัทมีข้อจำกัดทางด้านวัตถุดิบ คือ มีปริมาณสารละลายสีและสารเติมแต่งสีใช้ในการผลิตสีแต่ละวันสูงสุดไม่เกิน 34 ตัน และ 28 ตัน ตามลำดับ ความต้องการวัตถุดิบเพื่อใช้ในการผลิตสีแต่ละวัน และกำไรในการผลิตสีแต่ละชนิด แสดงในตารางที่ 2.1

วัตถุดิบ	ปริมาณวัตถุดิบ (ตัน) ในการผลิตสีต่อหนึ่งวัน		ปริมาณวัตถุดิบสูงสุด ที่สามารถนำไปใช้ ในการผลิต (ตัน)
	สีทาภายใน	สีทาภายนอก	
สารละลายสี	3	4	34
สารเติมแต่งสี	4	1	28
กำไรต่อตัน	3,000 บาท	2,000 บาท	

ตารางที่ 2.1 ปริมาณความต้องการวัตถุดิบเพื่อใช้ในการผลิตสีแต่ละวันและกำไรในการผลิตสีแต่ละชนิด

จากการสำรวจตลาดพบว่า ปริมาณความต้องการในแต่ละวันของสีทาภายนอกมีมากกว่าสีทาภายในวันละไม่เกิน 5 ตัน

ปัญหาข้อนี้ คือ ต้องการผลิตสีทาภายในและสีทาภายนอกวันละกี่ตันเพื่อให้ได้กำไรสูงสุด

การสร้างตัวแบบคณิตศาสตร์ของปัญหาดังกล่าวข้างต้น

1. กำหนดตัวแปรเพื่อการตัดสินใจ (Decision variables)

ปริมาณสีทาภายในและสีทาภายนอกในแต่ละวันใช้ตัวแปรดังนี้

x_1 = ปริมาณในการผลิตสีทาภายในแต่ละวัน (หน่วย : ตัน)

x_2 = ปริมาณในการผลิตสีทาภายนอกแต่ละวัน (หน่วย : ตัน)

2. กำหนดฟังก์ชันเป้าหมายที่ต้องการ (Objective function)

เนื่องจากสีทาภายในแต่ละวันได้กำไรตันละ 3,000 บาท สีทาภายนอกได้กำไรตันละ 2,000 บาท และทำการผลิตสีทาภายในวันละ x_1 ตัน สีทาภายนอกวันละ x_2 ตัน ดังนั้น กำไรรวมในการผลิตสีทั้งสองชนิดซึ่งใช้สัญลักษณ์ตัว Z มีค่าดังนี้

$$Z = 3x_1 + 2x_2 \quad (\text{หน่วย : พันบาท})$$

3. ข้อจำกัด (Constraints) ข้อจำกัดของการใช้วัตถุดิบในการผลิตสี ปริมาณสารละลายสีที่จะใช้ในการผลิตไม่เกินวันละ 34 ตัน และปริมาณของสารเติมแต่งสีไม่เกินวันละ 28 ตัน ดังนั้น จึงสามารถที่จะเขียนเป็นตัวแบบคณิตศาสตร์ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$3x_1 + 4x_2 \leq 34 \quad (\text{สารละลายสี})$$

$$4x_1 + x_2 \leq 28 \quad (\text{สารเติมแต่งสี})$$

และปริมาณความต้องการในแต่ละวันของสีทาภายนอกมีมากกว่าสีทาภายในไม่เกินวันละ 5 ตัน นอกจากนี้การผลิตสีแต่ละชนิดมีปริมาณตั้งแต่ศูนย์ขึ้นไป ซึ่งสามารถเขียนเป็นตัวแบบคณิตศาสตร์ได้ดังนี้

$$x_2 - x_1 \leq 5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

จากการสร้างตัวแบบคณิตศาสตร์ข้างต้น สามารถสรุปเป็นตัวแบบคณิตศาสตร์ของปัญหาการโปรแกรมเชิงเส้นดังต่อไปนี้

ทำให้ได้กำไรสูงสุด $Z = 3x_1 + 2x_2$ ฟังก์ชันเป้าหมาย
(Maximize) (Objective function)

ภายใต้ข้อจำกัด (subject to constraints)

$$3x_1 + 4x_2 \leq 34 \quad (1)$$

$$4x_1 + x_2 \leq 28 \quad (2)$$

$$-x_1 + x_2 \leq 5 \quad (3)$$

$$x_1, x_2 \geq 0$$

2.1.4 การสร้างตัวแบบปัญหาการโปรแกรมเชิงเส้น

ในตัวอย่างที่ 2.1 เป็นปัญหาการโปรแกรมเชิงเส้นอย่างง่าย ที่มีเพียง 2 ตัวแปรในการตัดสินใจ สามารถสร้างตัวแบบคณิตศาสตร์ได้ง่าย และแก้ปัญหาด้วยวิธีการกราฟได้ แต่ถ้าตัวแปรคณิตศาสตร์ในการตัดสินใจมีมากกว่า 2 ตัวแปร และข้อจำกัดมีมากกว่า 3 ข้อจำกัด การสร้างตัวแบบคณิตศาสตร์ของปัญหาโปรแกรมเชิงเส้นจะมีความซับซ้อนเพิ่มขึ้นเป็นลำดับ

ตัวอย่างที่ 2.2

โครงการปฏิรูปที่ดินในเขตอำเภอสนามชัย จังหวัดฉะเชิงเทรา มีพื้นที่ทั้งหมด 130,000 ไร่ แบ่งเป็นโครงการย่อยอยู่ 3 โครงการ ที่ให้ราษฎรเข้าไปทำการเพาะปลูกพืชไร่ คือ โครงการ ก. มีพื้นที่ 40,000 ไร่ โครงการ ข. มีพื้นที่ 60,000 ไร่ และ โครงการ ค. มีพื้นที่ 30,000 ไร่ จากการศึกษาคุณลักษณะของที่ดินจากผู้เชี่ยวชาญด้านการเกษตร พื้นที่ดินเหมาะสำหรับการปลูกพืชไร่ 3 ชนิด ในฤดูกาลเพาะปลูกนี้ต่อไป คือ อ้อย 30,000 ไร่ ข้าวโพด 60,000 ไร่ และมันสำปะหลัง 40,000 ไร่ และฝ่ายการการตลาดได้พยากรณ์ว่าการเพาะปลูกอ้อย ข้าวโพด และมันสำปะหลัง ในฤดูกาลต่อไป จะได้กำไรไร่ละ 800 บาท 900 บาท และ 600 บาท ตามลำดับ กรมชลประทานที่รับผิดชอบในการจัดสรรน้ำให้แก่โครงการย่อยทั้งสามโครงการ สามารถส่งน้ำให้ได้ปริมาณสูงสุดแก่โครงการย่อย ก. จำนวน 200,000 ลูกบาศก์เมตร โครงการย่อย ข. จำนวน 300,000 ลูกบาศก์เมตร และโครงการย่อย ค. จำนวน 150,000 ลูกบาศก์เมตร อัตราการใช้น้ำของอ้อย จำนวน 9 ลูกบาศก์เมตรต่อพื้นที่ 1 ไร่ อัตราการใช้น้ำของข้าวโพดจำนวน 6 ลูกบาศก์เมตรต่อพื้นที่ 1 ไร่ และมันสำปะหลังใช้น้ำจำนวน 3 ลูกบาศก์เมตร ต่อพื้นที่ 1 ไร่ นอกจากนั้นเงื่อนไขของการปฏิรูปที่ดิน ต้องการให้โครงการย่อยแต่ละโครงการใช้พื้นที่ปลูกจริงเป็นสัดส่วนเดียวกันกับพื้นที่ของแต่ละโครงการย่อย

จงสร้างตัวแบบโปรแกรมเชิงเส้นในการวางแผนการเพาะปลูก ของโครงการปฏิรูปที่ดินเพื่อให้ได้กำไรรวมสูงสุด

การสร้างตัวแบบ

กำหนดให้ x_1, x_2, \dots, x_9 คือตัวแปรของจำนวนพื้นที่แต่ละโครงการย่อยในการเพาะปลูกพืช 3 ชนิด ดังตารางที่ 2.2

พืชไร่	จำนวนพื้นที่เพาะปลูกของแต่ละโครงการย่อย (ไร่)			
	ก	ข	ค	รวม
อ้อย	x_1	x_2	x_3	$x_1 + x_2 + x_3$
ข้าวโพด	x_4	x_5	x_6	$x_4 + x_5 + x_6$
มันสำปะหลัง	x_7	x_8	x_9	$x_7 + x_8 + x_9$
รวม	$x_1 + x_2 + x_3$	$x_4 + x_5 + x_6$	$x_7 + x_8 + x_9$	

ตารางที่ 2.2 ตารางตัวอย่างปัญหาการโปรแกรมเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ $Z =$ กำไรรวมสูงสุด
 ดังนั้น

$$\begin{aligned} \text{ทำให้กำไรสูงสุด} \quad Z = & 800(x_1 + x_2 + x_3) + 900(x_4 + x_5 + x_6) \\ & + 600(x_7 + x_8 + x_9) \end{aligned}$$

ภายใต้เงื่อนไข

1. จำนวนพื้นที่

$$x_1 + x_2 + x_3 \leq 40,000$$

$$x_4 + x_5 + x_6 \leq 60,000$$

$$x_7 + x_8 + x_9 \leq 30,000$$

2. การตลาด

$$x_1 + x_2 + x_3 \leq 30,000$$

$$x_4 + x_5 + x_6 \leq 60,000$$

$$x_7 + x_8 + x_9 \leq 40,000$$

3. การจัดสรรน้ำ

$$9x_1 + 6x_2 + 3x_3 \leq 200,000$$

$$9x_4 + 6x_5 + 3x_6 \leq 300,000$$

$$9x_7 + 6x_8 + 3x_9 \leq 150,000$$

4. สัดส่วนพื้นที่ของการเพาะปลูก

$$\frac{x_1 + x_2 + x_3}{40,000} = \frac{x_4 + x_5 + x_6}{60,000} = \frac{x_7 + x_8 + x_9}{30,000}$$

5. จำนวนพื้นที่ในการเพาะปลูกต้องไม่น้อยกว่าศูนย์

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \geq 0$$

2.1.5 รูปแบบมาตรฐานและรูปแบบอื่นของการโปรแกรมเชิงเส้น

การโปรแกรมเชิงเส้น คือ ตัวแบบคณิตศาสตร์ในการจัดสรรทรัพยากรอันจำกัดให้แก่กิจกรรมต่างๆ เพื่อให้บรรลุผลดีที่สุด สามารถเขียนเป็นรูปแบบมาตรฐาน ** (Standard Form) ได้ดังนี้

** เรียกว่ารูปแบบมาตรฐานตามชื่อในหนังสือ Operations Research ของ Hillier and Lieberman แต่หนังสือเล่มนี้อาจเรียกชื่อที่แตกต่างกัน
 เอกสารนี้เป็นเอกสารที่โรงเรียนบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กำหนดให้ จำนวนกิจกรรมเท่ากับ n กิจกรรม
 จำนวนแหล่งทรัพยากรเท่ากับ m แหล่ง
- x_j = ระดับของกิจกรรมที่ j เมื่อ $j = 1, 2, \dots, n$
- b_i = จำนวนทรัพยากรจำกัดจากแหล่งที่ i เมื่อ $i = 1, 2, \dots, m$
- a_{ij} = ปริมาณการใช้ทรัพยากรในการผลิตกิจกรรมที่ j แต่หน่วยจะต้องบริโภค
 ทรัพยากรจากแหล่งที่ i ด้วยจำนวน a_{ij} เมื่อ
 $i = 1, 2, \dots, m$ และ $j = 1, 2, \dots, n$
- c_j = เป็นสัมประสิทธิ์ของกิจกรรมที่ j ซึ่งอาจจะเป็นกำไรต่อหน่วยหรือต้นทุนต่อ
 หน่วยของกิจกรรมที่ j

ดังนั้น สามารถเขียนเป็นรูปแบบมาตรฐานได้ดังนี้

$$\text{ทำให้ค่าสูงสุด } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

ภายใต้ข้อจำกัด

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_j \geq 0 \text{ เมื่อ } j = 1, 2, \dots, n$$

หรือเขียนเป็นเครื่องหมายการรวมกัน (Summation sign) คือ

$$n$$

$$\text{ทำให้ได้ค่าสูงสุด } Z = \sum_{j=1}^n c_j x_j$$

$$j=1$$

ภายใต้ข้อจำกัด

$$n$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad \text{เมื่อ } i = 1, 2, \dots, m$$

$$j=1 \quad j = 1, 2, \dots, n$$

และ

$$x_j \geq 0 \quad \text{เมื่อ } j = 1, 2, \dots, n$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบอื่น

จาก รูปแบบมาตรฐานนั้น ในทางปฏิบัติปัญหาตัวแบบการโปรแกรมเชิงเส้นอาจจะแตกต่างจากรูปแบบมาตรฐาน กล่าวคือ ฟังก์ชันเป้าหมายอาจจะทำให้ค่าต่ำสุดหรือสูงสุดของฟังก์ชันข้อจำกัด บางข้ออาจจะมีเครื่องหมาย \leq หรือ $=$ หรือ \geq และค่าตัวแปร บางตัวของ x_j อาจจะมีค่าบางค่าติดลบได้ ดังนั้นรูปแบบอื่นของปัญหาการโปรแกรมเชิงเส้น อาจเขียนเป็นรูปทั่วไปได้ดังนี้

$$\text{ทำให้ได้ค่าสูงสุดหรือค่าต่ำสุด } Z = \sum_{j=1}^n c_j x_j$$

ภายใต้ข้อจำกัด

$$\sum_{j=1}^n a_{ij} x_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_i \quad \text{เมื่อ } \begin{cases} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{cases}$$

และ

$$x_j \geq 0 \quad \text{และบางค่าของ } x_j \text{ ไม่จำกัดเครื่องหมาย} \\ \text{เมื่อ } j = 1, 2, \dots, n$$

2.1.6 ข้อสมมุติพื้นฐานของปัญหาการโปรแกรมเชิงเส้น

(Assumptions of Linear Programming)

จากตัวแบบการโปรแกรมเชิงเส้นของรูปแบบมาตรฐานในหัวข้อที่ผ่านมา มีข้อสมมุติพื้นฐาน 3 ประการดังต่อไปนี้

1. คุณสมบัติเชิงเส้น (Linear property)

คุณสมบัติเชิงเส้นนี้ในทางทฤษฎีหมายความว่า ฟังก์ชันเป้าหมายและฟังก์ชันข้อจำกัดต้องมีคุณสมบัติเป็น ฟังก์ชันเชิงเส้น (Linear function) เท่านั้น

คุณสมบัติเชิงเส้นอธิบายในทางปฏิบัติได้ว่า เมื่อตัวแปรในการตัดสินใจ (Decision variables) ของแต่ละกิจกรรม (x_j) เปลี่ยนไป 1 หน่วย จะทำให้ค่าของฟังก์ชันเป้าหมาย (Z) เปลี่ยนไปด้วยค่าคงที่ (c_j) และปริมาณการใช้ทรัพยากรเปลี่ยนไปด้วยค่าคงที่ (a_{ij})

นอกจากนั้นการเปลี่ยนแปลงของตัวแปรในการตัดสินใจของแต่ละกิจกรรมเป็น อิสระต่อกัน (Independent) และผลการดำเนินงานของแต่ละกิจกรรมสามารถรวมกันได้ (Additivity)

แม้ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ค่าพารามิเตอร์เป็นค่าคงที่ (Deterministic)

ค่าพารามิเตอร์ของตัวแบบปัญหาการโปรแกรมเชิงเส้น c_j , b_i , a_{ij} เป็น ค่าคงที่ (Deterministic) และสามารถประมาณค่าได้ (Estimate)

3. แบ่งแยกได้ (Divisionality)

ตัวแปรในการตัดสินใจ (x_j) ของผลลัพธ์เบื้องต้น (Optimal solution) เป็นค่าเศษส่วน (Fraction) ได้

การโปรแกรมเชิงเส้นยังคงถูกนำมาประยุกต์ใช้อยู่เสมอ แม้แต่ในสถานการณ์ที่ต้องการผลลัพธ์เป็นจำนวนเต็ม ถ้าคำตอบที่ได้เป็นเลขที่ไม่ใช่จำนวนเต็มแล้ว ตัวเลขที่ไม่ใช่จำนวนเต็มจะถูกทำให้มีค่าเป็นเลขจำนวนเต็ม นอกจากนั้นเราสามารถนำเทคนิคขั้นสูงขึ้นไปคือ การโปรแกรมเชิงเส้นจำนวนเต็ม (Linear Integer Programming) มาประยุกต์ใช้

2.2 ตัวแบบปัญหาการขนส่ง

(Transportation Problem Models)

ในโครงการพิเศษนี้ได้กล่าวถึง ตัวแบบที่สำคัญของปัญหาการโปรแกรมเชิงเส้น เรียกว่า “ ปัญหาการขนส่ง ” (Transportation Problem) เป็นตัวแบบคณิตศาสตร์ของปัญหาการขนส่งผลิตภัณฑ์ หรือ สินค้าจากแหล่งต้นทางใดๆ (Source) ไปยังแหล่งปลายทางใดๆ (Destinations) เพื่อให้ค่าใช้จ่ายรวมในการขนส่งมีค่าต่ำสุด

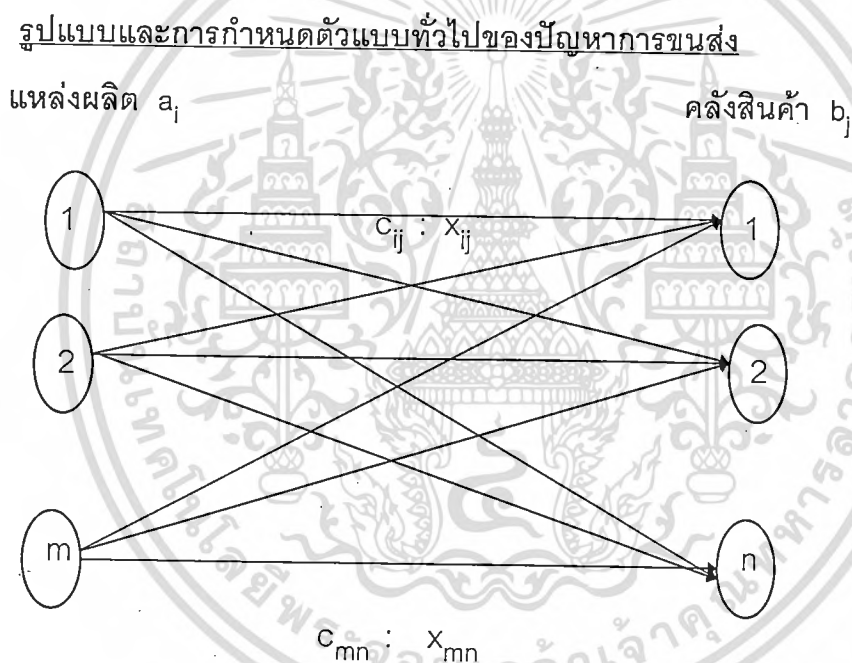
ตัวแบบปัญหาการขนส่งนี้ เราสามารถหาผลเฉลยตามเป้าหมายได้ด้วยวิธีซิมเพล็กซ์ปกติ แต่เนื่องจากสัมประสิทธิ์ (a_{ij}) ของปัญหาการขนส่งนี้ส่วนใหญ่มีค่าเป็น 0 (ศูนย์) และส่วนที่เหลืออยู่มีค่าเท่ากับ 1 (หนึ่ง) ซึ่งเป็นตัวแบบการโปรแกรมเชิงเส้นที่มีลักษณะพิเศษเฉพาะตัว การจะหาคำตอบด้วยวิธี ซิมเพล็กซ์ปกติเป็นการเสียเวลาในการคำนวณมาก ในที่นี้จะนำเสนอวิธีการคำตอบที่มีประสิทธิภาพในการคำนวณได้ง่ายและรวดเร็วกว่า

ตัวแบบปัญหาการขนส่งนอกเหนือจากการประยุกต์ ในปัญหาการจัดสรรปริมาณการขนส่งสินค้าหรือผลิตภัณฑ์จากแหล่งต้นทางไปยังแหล่งปลายทาง เพื่อให้ได้ค่าใช้จ่ายรวมต่ำที่สุดแล้ว ยังสามารถประยุกต์ใช้ในตัวของปัญหาการควบคุมพัสดุคงคลัง (Inventory control)

2.2.1 การกำหนดตัวแบบทั่วไปของปัญหาการขนส่ง

ปัญหาการขนส่ง (Transportation Problem) เป็นการพิจารณาหาปริมาณสินค้าที่จะทำการขนส่งจากแหล่งผลิตสินค้าไปยังผู้บริโภคหรือจุดหมายปลายทาง เพื่อให้ค่าใช้จ่ายในการขนส่งต่ำสุด หรือส่งสินค้าถึงจุดหมายปลายทางได้เร็วที่สุด

ในการกระจายจำนวนผลิตภัณฑ์หรือสินค้าใดๆ จากกลุ่มของแหล่งต้นทางใดๆ ไปยังแหล่งปลายทางใดๆ เพื่อให้ได้ค่าใช้จ่ายรวมในการขนส่งผลิตภัณฑ์หรือสินค้าที่ค่าต่ำที่สุด จำเป็นต้องกำหนดเทอม (Term) ทั่วไปของปัญหาการขนส่งเพื่อประกอบการสร้างตัวแบบดังนี้



แผนภาพที่ 2.1 แสดงแหล่งผลิตและคลังสินค้า

โดย	c_{ij}	เป็นค่าขนส่งต่อหน่วยของสินค้าที่มาจากแหล่งผลิต i ไปยังสินค้า j
	x_{ij}	เป็นปริมาณสินค้าที่มาจากแหล่งผลิต i ไปยังคลังสินค้า j
	a_i	เป็นปริมาณสินค้าที่แหล่งผลิต i ผลิตได้ในช่วงระยะเวลาหนึ่ง ๆ
	b_j	เป็นปริมาณสินค้าที่คลังสินค้า j จะรับได้ในช่วงระยะเวลาหนึ่ง ๆ

ปัญหาการโปรแกรมเชิงเส้นสำหรับปัญหาการขนส่งทั่วไปมีรูปแบบดังนี้
สมการจุดประสงค์ คือการหาค่าขนส่งต่ำสุด

$$\min . Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

โดยมีเงื่อนไข $\sum_{j=1}^n x_{ij} \leq a_i \quad ; i=1, 2, \dots, m$

$$\sum_{i=1}^m x_{ij} \leq b_j \quad ; j=1, 2, \dots, n$$

และ $x_{ij} \geq 0$ ทุกๆ ค่า i และ j
โดยมี $m =$ จำนวนแหล่งผลิต แหล่งส่ง (ต้นทาง)
 $n =$ จำนวนคลังสินค้า (จุดปลายทาง)

2.2.2 ตัวอย่างของรูปแบบปัญหาการขนส่ง

ตัวอย่างที่ 2.3

สมมุติให้มีแหล่งผลิตคือ โรงงาน 3 แห่ง ส่งสินค้าไปยังคลังสินค้าต่าง ๆ กัน 3 แห่ง เช่นกัน เขียนเป็นตารางของรูปแบบปัญหาทางการคิดค่าขนส่งมาตรฐาน โดยการเขียนค่าขนส่งลงในมุมด้านขวา ได้ดังตารางที่ 2.3

คลังสินค้า โรงงาน	A	B	C	ความสามารถ ในการผลิต
ก	X_{11} C_{11}	X_{12} C_{12}	X_{13} C_{13}	a_1
ข	X_{21} C_{21}	X_{22} C_{22}	X_{23} C_{23}	a_2
ค	X_{31} C_{31}	X_{32} C_{32}	X_{33} C_{33}	a_3
ความสามารถ ในการเก็บ	b_1	b_2	b_3	3 3 $\sum_{i=1}^3 a_i = \sum_{j=1}^3 b_j$

ตารางที่ 2.3 แสดงรูปแบบปัญหาทางการคิดค่าขนส่งมาตรฐาน

ตัวอย่างที่ 2.4

บริษัทวิศวกรไทยจำกัด ได้รับประมูลงานก่อสร้างของกรมทางหลวง โครงการขยายและบูรณะทางหลวงในจังหวัดภาคอีสานตอนกลางสามจังหวัด คือ ร้อยเอ็ด มหาสารคาม และ กาฬสินธุ์ โครงการนี้จำเป็นต้องใช้หินสัปดาห์ละ 1,235 ตัน จากการสำรวจแหล่งวัตถุดิบหินในภาคอีสานทั้งหมดที่สามารถจัดส่งให้กับโครงการนี้ ได้แก่โรงงานไม่หินจากสามจังหวัด ได้แก่ อุดรธานี บุรีรัมย์ และ อุบลราชธานี เนื่องจากอัตราค่าขนส่งหินและความสามารถในการจัดส่งปริมาณหิน จากโรงงานไม่หินทั้งสามแห่งไปยังสามจังหวัดที่ต้องการมีความแตกต่างกัน นอกจากนั้นความต้องการในการใช้หินของแต่ละจังหวัดก็มีความแตกต่างกันด้วย จากการศึกษาข้อมูลของค่าใช้จ่ายในการจัดส่ง และการรับวัตถุดิบในแต่ละแหล่งมีความแตกต่างกันด้วยรายละเอียดในตารางที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	ค่าขนส่ง (บาทต่อวัน)			ปริมาณการจัดส่ง (ตัน)
	แหล่งรับวัตถุดิบ (หิน)			
	1. ร้อยเอ็ด	2. มหาสารคาม	3. กาฬสินธุ์	
โรงงานไม่หิน				
1. อุดรธานี	220	180	170	400
2. อุบลราชธานี	160	195	200	335
3. บุรีรัมย์	150	165	190	500
ปริมาณความต้องการ - การ (ตัน)	520	390	325	1,235

ตารางที่ 2.4 ค่าใช้จ่ายในการขนส่งและปริมาณการจัดส่งและปริมาณการรับหินของแต่ละจังหวัด

จากข้อมูลของปัญหาโปรแกรมเชิงเส้นในตารางที่ 2.4 เราจะจัดการขนส่งวัตถุดิบหิน จากโรงงานไม่หินทั้งสามแห่ง ไปยังแหล่งสร้างทางทั้งสามจังหวัดได้อย่างไร เพื่อให้ได้ค่าใช้จ่ายรวมในการขนส่งมีค่าต่ำสุด ปัญหานี้เป็นรูปแบบของปัญหาการโปรแกรมเชิงเส้นในรูปแบบปัญหาการขนส่ง ซึ่งจะสามารถสร้างตัวแบบคณิตศาสตร์สำหรับปัญหาการขนส่งได้ดังต่อไปนี้

กำหนดให้

$$x_{ij} = \text{ปริมาณการขนส่งหิน (ตัน) จากโรงงานไม่หินที่ } i \text{ ไปยังจังหวัดสร้างทางที่ } j \\ \text{เมื่อ } i = 1, 2, 3 \text{ และ } j = 1, 2, 3$$

โดยอธิบาย x_{ij} ได้ดังตัวอย่าง เช่น

x_{11} = ปริมาณการขนส่งหินจากโรงงานไม่หินจังหวัดอุดรธานีไปยังจังหวัดร้อยเอ็ด
และ

x_{21} = ปริมาณการขนส่งหินจากโรงงานไม่หินจังหวัดอุบลราชธานีไปยังจังหวัดกาฬ
สินธุ์

ดังนั้นตัวแบบปัญหาการขนส่งนี้ คือ

$$\text{ทำให้ได้ค่าใช้จ่ายต่ำสุด} \quad Z = 220x_{11} + 180x_{12} + 170x_{13} + \\ 160x_{21} + 195x_{22} + 200x_{23} + \\ 150x_{31} + 165x_{32} + 190x_{33}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใต้ข้อจำกัด

$$x_{11} + x_{12} + x_{13} = 400$$

$$x_{21} + x_{22} + x_{23} = 335$$

$$x_{31} + x_{32} + x_{33} = 500$$

$$x_{11} + x_{21} + x_{31} = 520$$

$$x_{12} + x_{22} + x_{32} = 390$$

$$x_{13} + x_{23} + x_{33} = 325$$

และ $x_{ij} \geq 0$ (เมื่อ $i = 1, 2, 3$ และ $j = 1, 2, 3$)

ผลเฉลยตามเป้าหมาย (Optimal solution) ของปัญหาการขนส่งนี้คือ

$$x_{11} = 0, \quad x_{12} = 75, \quad x_{13} = 325,$$

$$x_{21} = 335, \quad x_{22} = 0, \quad x_{23} = 0,$$

$$x_{31} = 185, \quad x_{32} = 315, \quad x_{33} = 0$$

หรือจะอธิบายได้ว่า เพื่อให้ได้ค่าใช้จ่ายรวมในการขนส่งต่ำที่สุด เราต้องขนส่งหินจากโรงงานไม่หินอุดรธานีไปยังจังหวัดมหาสารคามจำนวน 75 ตัน และกาฬสินธุ์จำนวน 325 ตัน แต่ไม่ส่งไปที่จังหวัดร้อยเอ็ด สำหรับโรงงานไม่หินอุบลราชธานีจัดส่งหินไปเฉพาะที่จังหวัดร้อยเอ็ดจำนวน 335 ตัน สำหรับโรงงานไม่หินบุรีรัมย์จัดส่งไปเพียง 2 จังหวัด คือ ร้อยเอ็ด จำนวน 185 ตัน และ มหาสารคามจำนวน 315 ตัน

2.3 การหาผลเฉลยของรูปแบบปัญหาการขนส่ง

(Solving Transportation Problem)

จุดมุ่งหมายของปัญหาการขนส่ง เป็นไปในทำนองเดียวกับปัญหาการโปรแกรมเชิงเส้นคือ ต้องการผลที่เป็นไปได้ทั้งการให้ต้นทุนต่ำสุด และการให้ผลกำไรสูงสุด

ในการหาผลเฉลยตามจุดมุ่งหมายของปัญหาการขนส่งเราแบ่งออกเป็น 2 ขั้นตอน

1. การหาผลเฉลยเบื้องต้น
2. การหาผลเฉลยตามเป้าหมาย

2.3.1 การหาผลเฉลยเบื้องต้น (Basic Feature Solution)

การกำหนดพื้นฐานผลลัพธ์พื้นฐานที่เป็นไปได้เริ่มต้น

(Determination of an initial basic feasible solution)

ในการหาผลเฉลยเบื้องต้นนี้ถ้าเป็นปัญหาการขนส่งแบบไม่มีตัวแปรมาก เราทำได้โดยการ

ใส่ค่าให้เป็นตามเงื่อนไข $\sum_{j=1}^n x_{ij} = a_i$ และ $\sum_{i=1}^m x_{ij} = b_j$ ก็เป็นอันใช้ได้ แต่ถ้า

เป็นปัญหาซึ่งมีจำนวนโรงงานและคลังพัสดุหลายๆ อันจะทำให้เกิดปัญหาอย่างมาก ดังนั้นจึงต้องหาหลักเกณฑ์เพื่อลดความยุ่งยากที่เกิดขึ้น และลดเวลาที่จะใช้ในการคำนวณขั้นต่อไป หลักเกณฑ์ของปัญหาการขนส่งนี้มีหลายวิธี แต่วิธีที่ใช้กันแพร่หลายมีอยู่ 3 วิธี

1. วิธีตามกฎของมุมทิศเหนือ - ตก (North - west corner rule)
2. วิธีตามกฎการประมาณค่าน้อยที่สุด (Lowest First Costs Rule)
3. วิธีการประมาณค่าของโวลเกิล (Vogel's approximation method หรือ VAM)

ตัวแบบปัญหาการขนส่งมีข้อจำกัดทุกข้ออยู่ในรูปสมการ (equality constraints) ผลลัพธ์พื้นฐานที่เป็นไปได้ใดๆ ที่อยู่บนตารางแบบปัญหาการขนส่ง มีตัวแปรพื้นฐานเท่ากับ $m + n - 1$ โดยค่าตัวแปรพื้นฐานเหล่านั้นเขียนวงกลมล้อมรอบอยู่ในตารางแบบปัญหาการขนส่ง

ในบางครั้งเกิดเหตุการณ์ที่ ตัวแบบของปัญหาการขนส่งที่ไม่สมดุล (Unbalanced transportation problem) การหาผลลัพธ์ของปัญหาการขนส่งที่ไม่สมดุลทุกปัญหา เราจำเป็นต้องปรับตัวแบบนี้ให้เป็นปัญหาการขนส่งแบบสมดุล โดยอาจจะต้องเพิ่ม แหล่งต้นทางสมมุติ (Dummy source) หรือ แหล่งปลายทางสมมุติ (Dummy destination) ที่จำเป็นในตัวแบบ และกำหนดค่าใช้จ่ายในแต่ละช่องที่เกิดจากแหล่งต้นทางสมมุติ หรือ แหล่งปลายทางสมมุติตามความเหมาะสม และสอดคล้องกับเงื่อนไขของปัญหาที่โจทย์กำหนดให้ ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.1 วิธีตามกฎของมุมทิศ เหนือ - ตก

(North - West Corner Rule)

ในวิธีการตามกฎนี้เป็นวิธีง่ายที่สุดในการหาผลเฉลยเบื้องต้น ซึ่งมีขั้นตอนดังนี้

ก. เริ่มจัดค่า x_{ij} ที่ช่องตามตารางในมุมบนทางซ้ายมือของตาราง ซึ่งแทนความหมายของทิศเหนือเฉียงไปในทางทิศตะวันตกของตาราง

ข. จัดค่า x_{ij} ให้มากที่สุดเท่าที่จะมากได้ในช่องซ้ายมือได้ค่า x_{ij} มากที่สุด จากการเปรียบเทียบปริมาณความสามารถการผลิตกับขนาดความต้องการ ค่าที่น้อยกว่าจะเป็นค่าที่ต้องการคือ

$$x_{ij} = \min. (a_i, b_j) \text{ ของแถวบนและแถวตั้งที่ตัดกับมุมบนซ้ายมือ}$$

ค. จัดค่า x_{ij} สำหรับช่องถัดมาทางขวามือหรือถัดลงมาข้างล่างโดยให้ x_{ij} มีค่ามากที่สุดจากส่วนที่เหลือจากปริมาณความสามารถการผลิต $(a_i - x_{ij})$ หรือขนาดความสามารถของการเก็บ $(b_j - x_{ij})$ เทียบกับขนาดความสามารถของคลังเก็บสินค้าที่ถัดมา b_{i+1} หรือปริมาณความสามารถที่ผลิตของโรงงานที่ถัดมา a_{i+1} หรือปริมาณความสามารถที่ผลิตของโรงงาน และที่คลังสินค้า

จากตัวอย่างที่ 2.4 กำหนดให้ตัวแปรพื้นฐานเริ่มต้นที่มุมทิศ เหนือ - ตก ของตารางแบบปัญหาการขนส่งคือ x_{11} กำหนดให้การจัดสรรค่าของ (x_{11}) มากที่สุดเท่าที่จะมากได้ ดังนั้นค่าของ $x_{11} = 400$ ทำให้ความสามารถในการผลิตของแถวที่ 1 หหมดไป และความสามารถในการเก็บที่ 1 เหลือ 120 ดังนั้นตัวแปรพื้นฐานต่อไปใหม่ คือ $x_{21} = 120$ ทำให้ความสามารถในการเก็บของแถวที่ 1 หหมดไป และทำให้ความสามารถในการผลิตที่ 2 เหลือ 215 ดังนั้นตัวแปรพื้นฐานต่อไปใหม่ คือ $x_{22} = 215$ ทำให้ความสามารถในการผลิตของแถวที่ 2 หหมดไป และความสามารถในการเก็บของสดมภ์ที่ 2 เหลือ 175 ดังนั้นตัวแปรพื้นฐานต่อไปตัวที่สี่ คือ $x_{32} = 175$ ทำให้ความสามารถในการเก็บของสดมภ์ที่ 2 หหมดไป และความสามารถในการผลิตของแถวที่ 3 เหลือ 325 ดังนั้นตัวแปรพื้นฐานตัวสุดท้าย คือ $x_{33} = 325$ รายละเอียดในการคำนวณอยู่ในตารางที่ 2.5

จังหวัดที่รับหิน โรงงาน	1 ร้อยเอ็ด	2 มหาสารคาม	3 กาฬสินธุ์	ความสามารถใน การส่งหิน
1. อุดรธานี	400 220	180	170	400
2. อุบลราชธานี	120 160	215 195	200	335
3. บุรีรัมย์	150	175 165	325 190	500
ความสามารถ ในกกรรับหิน	520	390	325	$\sum_{i=1}^3 a_i = \sum_{j=1}^3 b_j$

ตารางที่ 2.5 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้เริ่มต้นด้วย วิธี ตามกฎมุมทิศเหนือ - ตก ของบริษัทวิศวกรรมไทย

สรุปผลลัพธ์พื้นฐานที่เป็นไปได้เริ่มต้น ด้วยวิธีของกฎมุมทิศเหนือ - ตก คือ $x_{11} = 400$, $x_{21} = 120$, $x_{32} = 175$ และ $x_{33} = 325$ นอกนั้นตัวแปรที่เหลืออยู่คือตัวแปรจร มีค่าเท่ากับศูนย์ และค่าใช้จ่ายรวมมีค่าเท่ากับ $(400 \cdot 220) + (120 \cdot 160) + (175 \cdot 195) + (325 \cdot 190) = 239,750$ บาท

2.3.1.2 วิธีตามกฎการประมาณค่าน้อยที่สุด

(Lowest First Costs Rule)

ขั้นตอนในการคำนวณตัวแปรพื้นฐาน ด้วยวิธีเลือกตัวแปรพื้นฐานที่มีค่าใช้จ่ายน้อยที่สุด คือ เลือกตัวแปรพื้นฐานที่มีค่าใช้จ่ายในการขนส่งน้อยที่สุด เป็นตัวแปรพื้นฐานเริ่มต้น (ในกรณีที่ค่าใช้จ่ายเท่ากัน ให้เลือกตัวใดตัวหนึ่ง) กำหนดค่าของตัวแปรพื้นฐานเริ่มต้นให้มีค่ามากที่สุดเท่าที่จะมากได้ ซึ่งจะส่งผลให้ปริมาณการจัดสรรความสามารถในการผลิตของแถว หรือความสามารถในการเก็บของสดมภ์ที่สมนัยกับตัวแปรพื้นฐานถูกขจัดออกไป และปรับค่าปริมาณการจัดสรรของแถวหรือสดมภ์ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เหลืออยู่ จากนั้นด้วยวิธีการทำนองเดียวกันกับการเลือกตัวแปรพื้นฐานเริ่มต้น เลือกตัวแปรพื้นฐานต่อไปที่มีค่าใช้จ่ายน้อยที่สุดถัดไป และกำหนดค่าตัวแปรพื้นฐานตัวต่อไปนี้มีค่าเท่ากับปริมาณการจัดสรรของแต่ละแถวหรือสดมภ์ที่เหลืออยู่ จนในที่สุดเรากำหนดให้ตัวแปรพื้นฐานตัวสุดท้ายมีค่าเท่ากับจำนวนความสามารถในการเก็บหรือความสามารถในการผลิตของสดมภ์หรือแถวสุดท้ายที่เหลืออยู่

จังหวัดที่รับหิน โรงงาน	1 ร้อยเอ็ด	2 มหาสารคาม	3 กาฬสินธุ์	ความสามารถใน การส่งหิน
1. อุดรธานี	220	180	170	400
2. อุบลราชธานี	160	195	200	335
3. บุรีรัมย์	150	165	190	500
ความสามารถ ในการรับหิน	520	390	325	$\sum_{i=1}^3 a_i = \sum_{j=1}^3 b_j$

ตารางที่ 2.6 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้เริ่มต้นด้วย วิธีตามกฎการประมาณค่าน้อยที่สุด ของบริษัทวิศวกรไทย

กรณีตัวอย่างที่ 2.4 เมื่อกำหนดตัวแปรพื้นฐานที่เป็นไปได้เริ่มต้นด้วยวิธีเลือกตัวแปรที่ใช้ค่าใช้จ่ายน้อยที่สุด ในตารางที่ 2.6 เลือกตัวแปรพื้นฐานที่มีค่าใช้จ่ายน้อยที่สุด คือ $x_{31} = 500$ ทำให้ความสามารถในการผลิตของแถวที่ 3 หดไป และความสามารถในการเก็บที่ 1 เหลือ 20 ต่อไปเลือกตัวแปรพื้นฐานที่ให้ค่าใช้จ่ายในการขนส่งน้อยที่สุดตัวถัดไป คือ $x_{21} = 20$ ซึ่งเป็นปริมาณการจัดสรรของสดมภ์ที่ 1 ที่เหลืออยู่ ทำให้ความสามารถในการเก็บของแถวที่ 1 หดไป และความสามารถในการผลิตที่ 2 เหลือ 315 ดังนั้นตัวแปรพื้นฐานต่อไปที่ให้ค่าใช้จ่ายน้อยที่สุดตัวแปรถัดไป คือ $x_{13} = 325$ ด้านการคำนวณ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(เราไม่เลือก x_{32} เพราะว่ามีปริมาณการจัดสรรของแถวที่ 3 ได้หมดไปแล้ว) ทำให้ความสามารถในการเก็บของสดมภ์ที่ 3 หมดไป และความสามารถในการผลิตของแถวที่ 1 เหลือ 75 จากนั้นก็เลือกเอาตัวแปรพื้นฐานของตัวแปรต่อไปด้วยวิธีการเดียวกัน คือ $x_{12} = 75$ และ $x_{22} = 315$ ตามลำดับ นอกนั้นตัวแปรที่เหลืออยู่คือตัวแปรจร มีค่าเท่ากับศูนย์ ค่าใช้จ่ายรวมในตารางเริ่มต้นของวิธีเลือกตัวแปรที่มีค่าใช้จ่ายน้อยที่สุด มีค่าเท่ากับ $(75 \cdot 180) + (325 \cdot 170) + (20 \cdot 160) + (315 \cdot 195) + (500 \cdot 150) = 208,375$ บาท

2.3.1.3 วิธีการประมาณค่าของโวลเกิล

(Vogel 's Approximation Method)

วิธีการตามกฎของมุทิตศเหนือ - ตก นั้น เป็นวิธีการหาผลเฉลยเบื้องต้นที่จะต้องใช้อีกขั้นตอนหนึ่งในการหาผลเฉลยตามเป้าหมาย ถ้าผลเฉลยเบื้องต้นมีค่าห่างจากเป้าหมายมากขึ้นตอนเพื่อการหาผลเฉลยตามเป้าหมายจะใช้เวลามากกว่า ดังนั้นผลเฉลยจากวิธีตามกฎมุทิตศเหนือ - ตก จึงอาจมีโอกาที่ขั้นตอนเพื่อการหาผลเฉลยตามเป้าหมายจะใช้เวลามากหรือน้อยก็ได้ ซึ่งจะเป็นไปตามลักษณะ c_{ij} , a_i และ b_j เพื่อที่จะให้แน่ใจว่าผลเฉลยเบื้องต้นจะมีค่าใกล้เคียงผลลัพธ์ตามเป้าหมายยิ่งขึ้น วิธีการประมาณค่าโวลเกิลจะพิจารณาค่าใช้จ่ายที่แถวบนและแถวตั้งร่วมกัน มีชื่อย่อว่า " VAM " จะช่วยให้ขั้นตอนเพื่อการหาผลเฉลยตามเป้าหมายใช้เวลาเฉลยวิธีวิธีนี้มีขั้นตอนดังนี้

ก. จากตารางรูปแบบปัญหาการขนส่ง หาค่าผลต่างระหว่างค่าขนส่งจากที่ต่ำสุดกับค่า ซึ่งต่ำรองลงมาตามแนวนอนและแนวตั้งของแต่ละแถว จากนั้น พิจารณาผลต่างที่มีค่าสูงสุดใช้เป็นตัวกำหนดแถวบนหรือแถวตั้งนั้นๆ ซึ่งเป็นแถวที่เราจะพิจารณาจัดสรร x_{ij} ก่อน

ข. จัดสรร x_{ij} ให้มากที่สุดเท่าที่จะมากได้ลงในช่องที่มีค่าขนส่งต่ำสุดของตารางตามแนวนอนและแนวตั้ง ที่เลือกแล้วในข้อ ก. ค่า x_{ij} ที่จัดสรรได้มากที่สุดจะต้องไม่มากกว่าค่าของ a_i หรือ b_j ที่เหลือจากการจัดสรร x_{ij} อื่นๆ แล้ว เช่นเดียวกับวิธีตามกฎของมุทิตศเหนือ - ตก ที่กล่าวมาแล้ว

ค. ในแถวบนหรือแถวตั้งซึ่งเราจัด x_{ij} อื่นๆ แล้วมีผลทำให้ ผลรวม x_{ij} ที่จัดแล้วเท่ากับ a_i และ b_j ช่องอื่นๆ ในแถวบนหรือแถวตั้งดังกล่าวให้จัดสรร x_{ij} เป็นศูนย์ แล้วไม่ต้องพิจารณาแถวนั้นอีก

ง. พิจารณาขั้นตอน ก. ข. และ ค. จนกระทั่งไม่มีการจัดสรรได้อีก

โดยวิธีการหาผลเฉลยเบื้องต้นจากวิธีการประมาณค่าไวเกล มีค่าขนส่ง x_{ij} เป็นหลักเกณฑ์ในการพิจารณาเพื่อจัดสรร x_{ij} ฉะนั้นผลเฉลยเบื้องต้นจากวิธีนี้จะใกล้เคียงผลเฉลยตามเป้าหมายและในการแก้ปัญหาการขนส่งบางครั้ง บางปัญหา ผลเฉลยเบื้องต้นจากวิธีนี้ได้ผลเฉลยตามเป้าหมายโดยทันที

จังหวัดที่รับหิน โรงงาน	1 ร้อยเอ็ด	2 มหาสารคาม	3 กาฬสินธุ์	ความสามารถ ในการส่งหิน	U_i
1. อุดรธานี	+55 220	75 180	325 170	400	15
2. อุบลราชธานี	335 160	+20 195	+35 200	335	10
3. บุรีรัมย์	185 150	315 165	+35 190	500	0
ความสามารถ ในการรับหิน	520	390	325	3 3 $\sum_{i=1}^3 a_i = \sum_{j=1}^3 b_j$	
V_j	150	165	155	Z = 202,075 บาท	

ตารางที่ 2.6 การกำหนดผลลัพธ์พื้นฐานที่เป็นไปได้เริ่มต้นด้วย วิธีการประมาณค่าของไวเกลของบริษัทวิศวกรไทย

จากตัวอย่างที่ 2.4 จากตารางค่าใช้จ่าย คำนวณหาค่าความแตกต่างของค่าใช้จ่ายที่น้อยที่สุดและน้อยที่สุดถัดไป ของแต่ละแถวและแต่ละสดมภ์ ปรากฏว่าความแตกต่างที่มากที่สุดอยู่บนแถวที่ 2 มีค่าเท่ากับ 35 และค่าใช้จ่ายน้อยที่สุดของแถวที่ 2 คือ 160 ดังนั้นตัวแปรพื้นฐานเริ่มต้น คือ x_{21} ต่อไปจัดสรรค่าของ x_{21} ให้มากที่สุดเท่าที่จะมากได้ เลือก $x_{21} = 335$ ทำให้ความสามารถในการผลิตของแถวที่ 2 หมดไป จึงต้องขจัด (Eliminate) แถวที่ 2 ออกไป ลำดับต่อไปคำนวณหาความแตกต่างครั้งที่ 2 ของค่าใช้จ่ายน้อยที่สุดและน้อยที่สุดถัดไปที่เหลืออยู่ 2 แถว คือ แถวที่ 1 และที่ 3 กับสดมภ์อีก 3 สดมภ์ ปรากฏว่าเราเลือกตัวแปรพื้นฐานตัวที่ 2 คือ $x_{31} = 185$ จึงต้องขจัดสดมภ์ที่ 1 ออกไป ดังนั้น

การคำนวณครั้งที่ 3 ค่าความแตกต่างของค่าใช้จ่ายที่น้อยที่สุดถัดไปของแถวที่ 1 กับที่ 3 และสดมภ์ที่ 2 กับที่ 3 ที่เหลืออยู่ ปรากฏว่าตัวแปรพื้นฐานตัวที่ 3 คือ $x_{32} = 315$ จึงต้องขจัดแถวที่ 3 ออกไปในการคำนวณครั้งที่ 4 จะเหลือแถวที่ 1 และสดมภ์ที่ 2 กับสดมภ์ที่ 3 ดังนั้นตัวแปรพื้นฐานตัวที่สี่ และ ห้า คือ $x_{13} = 325$ และ $x_{12} = 75$

จากผลลัพธ์พื้นฐานเริ่มต้นด้วยวิธีประมาณค่าไวเกล ค่าใช้จ่ายรวมในการขนส่งเริ่มต้นเท่ากับ $(335 \cdot 160) + (185 \cdot 150) + (315 \cdot 165) + (325 \cdot 170) + (75 \cdot 180) = 202,075$ บาท

2.3.2 การหาผลเฉลยตามเป้าหมาย

(Optimal Solution)

การหาผลเฉลยตามเป้าหมาย เราจะต้องเข้าใจเงื่อนไขที่จำเป็นคือ จำนวนเซลล์วงกลม (circle cells) หรือ X_{ij} ที่เราได้จัดสรรไว้ในผลเฉลยเบื้องต้น เราจะต้องตรวจดูจำนวนเซลล์วงกลมให้เท่ากับ $m+n-1$ โดย $m =$ จำนวนแถวขน (รวมตัวแปรเทียมด้วย) และ $n =$ จำนวนแถวตั้ง มิฉะนั้นแล้วจะเกิดปัญหาซึ่งก่อให้เกิดการย้อนวนในการหาผลเฉลยตามเป้าหมาย วิธีการหาผลเฉลยนี้มีหลายวิธี แต่ที่จะกล่าวถึงในที่นี้คือ

- วิธีสเตปปีงสโตน (Stepping Stone Method)
- วิธีโมได (Modified Distribution : MODI)

2.3.2.1 วิธีสเตปปีงสโตน

(Stepping Stone Method)

จากตารางผลเฉลยเบื้องต้นของปัญหาการขนส่ง เมื่อคิดจำนวนของ X_{ij} ที่จัดสรรให้เป็นเซลล์วงกลมได้เท่ากับ $m+n-1$ แล้วเราก็ดำเนินการตามขั้นตอนของวิธีสเตปปีงสโตน แต่การใช้วิธีการนี้ เราจะต้องทำความเข้าใจในเรื่อง วิธี (path) ของการจัดสรร X_{ij} เสียก่อน

การหาวิถี (Find Path)

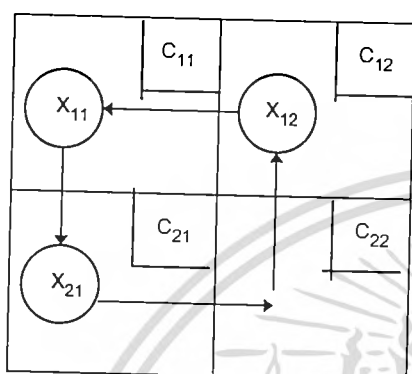
วิถี คือ ทางเดินที่เชื่อมต่อดจุดยอด (vertex)

จุดยอด คือ เซลล์วงกลมที่จะเปลี่ยนมุมไปยังจุดยอดต่อไป เซลล์วงกลมที่ไม่ได้เปลี่ยนมุมจะเป็นเพียงตัวผ่าน ไม่ใช่จุดยอด

แผนภาพที่ 2.2 แสดงวิธีการหาวิถี (Path) ของตารางการขนส่ง

ตัวอย่างกรณีการหาวิถี มีดังนี้

ตัวอย่าง ก

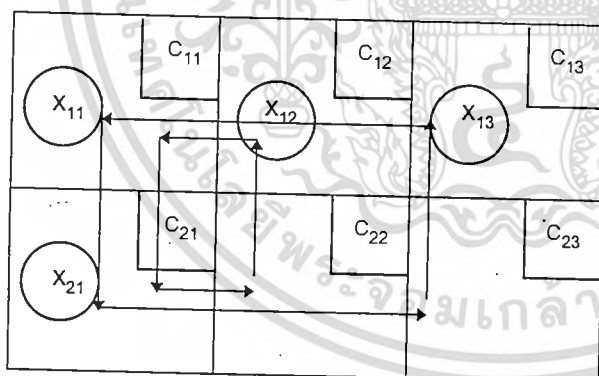


จำนวนเซลล์ว่างกลม

$$m + n - 1 = 2 + 2 - 1 = 3$$

วิถี (2, 2) มีจุดยอด คือ $X_{12} \rightarrow X_{11} \rightarrow X_{21}$

ตัวอย่าง ข



จำนวนเซลล์ว่างกลม

$$m + n - 1 = 3 + 2 - 1 = 4$$

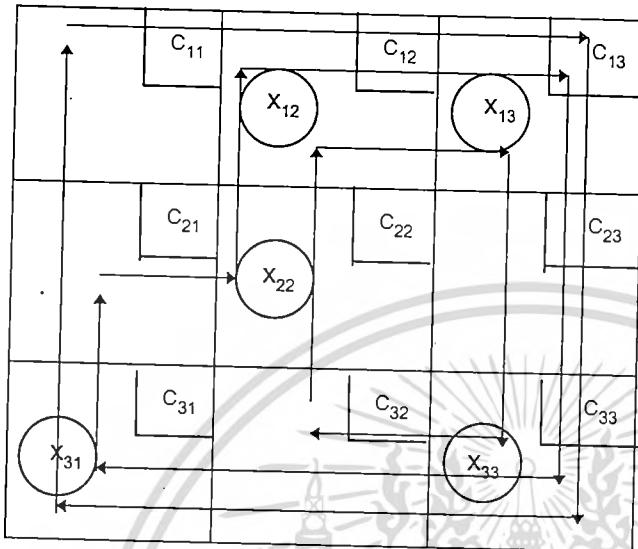
วิถี (2, 2) มีจุดยอด คือ $X_{12} \rightarrow X_{11} \rightarrow X_{21}$

วิถี (2, 3) มีจุดยอด คือ $X_{13} \rightarrow X_{12} \rightarrow X_{22}$

X_{12} เป็นทางผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง ค



- วิถี (1, 1) มีจุดยอด คือ $X_{13} \rightarrow X_{33} \rightarrow X_{31}$
- วิถี (2, 1) มีจุดยอด คือ $X_{22} \rightarrow X_{12} \rightarrow X_{13} \rightarrow X_{33} \rightarrow X_{31}$
- วิถี (2, 3) มีจุดยอด คือ $X_{12} \rightarrow X_{11} \rightarrow X_{21}$
- วิถี (3, 2) มีจุดยอด คือ $X_{12} \rightarrow X_{13} \rightarrow X_{33}$

ข้อสังเกต วิถีจะหมุนไปทางใดก็ตาม จะมีจุดยอดแบบเดียวกัน เพียงแต่เปลี่ยนทิศเท่านั้น

1. ขั้นตอนเพื่อตรวจดูว่าผลเฉลย ได้ผลตามเป้าหมายแล้วหรือยัง
2. ขั้นตอนเพื่อจัดการเคลื่อนย้ายเซลล์วงกลม เพื่อให้ได้ผลเฉลยตามเป้าหมาย

เมื่อเคลื่อนย้ายเซลล์วงกลม แล้ว จะต้องตรวจสอบผลเฉลยแบบขั้นที่ 1 อีกที ถ้ายังไม่ได้ผลเฉลยตามเป้าหมายก็ดำเนินการตามขั้นที่ 2 จนกว่าจะได้ผลเฉลยตามเป้าหมาย

การตรวจพิจารณาหาผลเฉลยตามเป้าหมาย มีขั้นตอนพอสรุปได้ดังนี้

1. จากตารางผลเฉลยเบื้องต้นที่มีจำนวนเซลล์วงกลม เท่ากับ $m+n-1$ สำหรับส่วนที่ไม่ได้จัดสรรค่า X_{ij} จะได้ $X_{ij} = 0$ เรียกส่วนนี้ว่าเซลล์ไม่เป็นวงกลม (non-circle cells) พิจารณาโดยใช้เซลล์ไม่เป็นวงกลมแต่ละช่วงเป็นจุดเริ่มต้น หาวิถีจนครบทุกเซลล์ไม่เป็นวงกลม
2. สำหรับแต่ละวิถีของเซลล์ไม่เป็นวงกลม เราเริ่มต้นเครื่องหมาย (+) ผ่านจุดยอดซึ่งเป็นเซลล์วงกลม ที่จะเปลี่ยนมุมจุดแรกเป็นลบ (-) และจุดต่อมาเป็นบวก (+) สลับเครื่องหมายไปจนกว่าจะครบรอบ ใช้ค่าเครื่องหมายประจำจุดยอดของวิถีพร้อมทั้งค่า ของจุดยอดเหล่านั้นมาหาผล

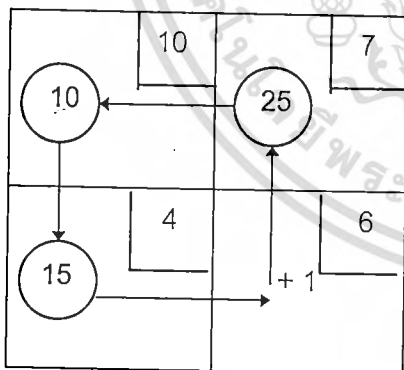
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลี่ยแสดงฐานะของเซลล์ไม่เป็นวงกลมนั้นๆ โดยใช้ผลรวมของ ของจุดยอดซึ่งมีเครื่องหมายต่างกัน สลับกัน แต่ผลเฉลี่ยเป็นลบแสดงว่าวิถีของเซลล์ไม่เป็นวงกลมนั้นๆ มีค่าต่ำกว่าการเคลื่อนย้าย จำนวน ที่จัดสรรไว้เดิม โดยการเพิ่มค่า ในช่วงเซลล์ไม่เป็นวงกลมให้เป็นวงกลม และ ลดค่าของ เซลล์วงกลมให้น้อยลง หรือ ให้กลายเป็นเซลล์ไม่เป็นวงกลมไป จะมีผลทำให้ผลเฉลี่ยมีค่าต่ำลง ดัง นั้นเพื่อที่จะพิจารณาว่า ผลเฉลยนั้นเป็นผลเฉลยตามเป้าหมายแล้วหรือยัง ก็ต้องดูเป้าหมายของเรา ก่อน หลังจากนั้นจึงพิจารณาค่าผลเฉลยจากวิธีซึ่งแสดงเป็นค่าบวกหรือลบ หรือศูนย์ ถ้ามีเป้าหมาย เพื่อลดค่าขนส่งและผลเฉลยวิถีทั้งหมดมีค่าเป็นบวก ก็แสดงว่าได้ผลเฉลยตามเป้าหมายแล้ว แต่ถ้า ผลเฉลยของวิถีใดมีค่าลบก็แสดงว่ายังไม่ได้ผลเฉลยตามเป้าหมาย ซึ่งหมายความว่า จะสามารถลด ค่าขนส่งลงไปได้อีก ในทางตรงกันข้าม ถ้ามีเป้าหมายเพื่อให้ได้กำไรสูงสุด ผลเฉลย ของวิถีก็ต้องเป็น ลบจึงจะหมายถึงผลเฉลยที่ได้ตรงตามเป้าหมายแล้ว นั่นคือ ถ้าผลเฉลยของวิถีมีทั้งเครื่องหมายบวก และลบ ก็จะต้องดำเนินการเคลื่อนย้ายต่อไป จนกว่าจะได้ผลเฉลยของวิถีมีเครื่องหมายบวก หรือ ลบเพียงอย่างเดียว เพื่อให้ได้ผลเฉลยตามจุดมุ่งหมายเป้าหมายที่วางไว้

ตัวอย่างการพิจารณาผลเฉลยวิถี โดยมีเครื่องหมายประจำจุดยอดของวิถี รวมการคิดค่า ดังนี้

แผนภาพที่ 2.3 การพิจารณาผลเฉลยของวิถี

ตัวอย่าง ก



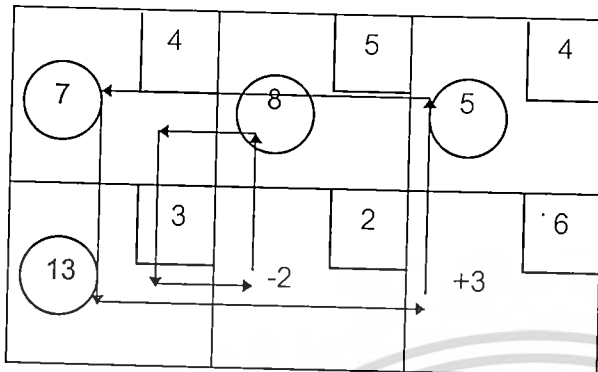
เป้าหมาย : ค่าขนส่งต่ำสุด

ค่าวิถี (2, 2) มีจุดยอด คือ +4, -7, +10, -6 = +1

มีผลเฉลยคือ +1

สรุป : ได้ผลลัพธ์ตามเป้าหมายแล้ว

ตัวอย่าง ข



เป้าหมาย : ค่าขนส่งต่ำสุด

ค่าวิถี (2, 2) มีจุดยอด คือ +2, -5, +4, -3 = -2

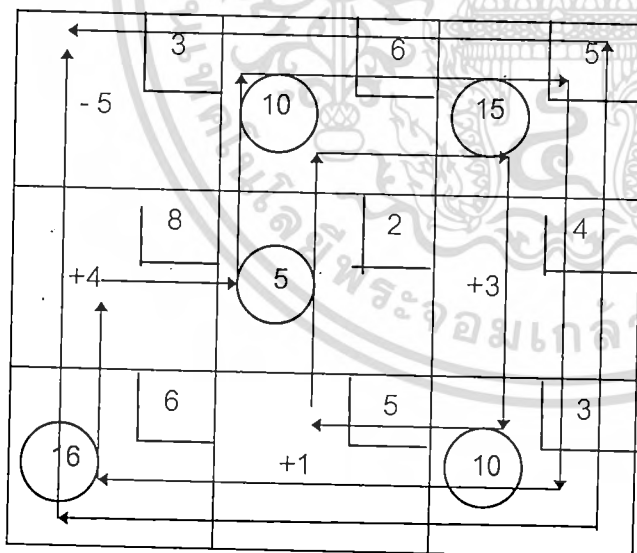
มีผลเฉลี่ยคือ -2

ค่าวิถี (2, 3) มีจุดยอด คือ +6, -4, +4, -3 = +3

มีผลเฉลี่ยคือ +3

สรุป : ยังไม่ได้ผลลัพธ์ตามเป้าหมายสำหรับวิถี (2, 3)

ตัวอย่าง ค



เป้าหมาย : ค่าขนส่งต่ำสุด

ค่าวิถี (1, 1) มีจุดยอด คือ +3, -5, +3, -6 = -5

ค่าวิถี (2, 1) มีจุดยอด คือ +8, -2, +6, -5, +3, -6 = +4

ค่าวิถี (2, 3) มีจุดยอด คือ +4, -5, +6, -2 = +3

ค่าวิถี (3, 2) มีจุดยอด คือ +5, -6, +5, -3 = +1

สรุป : ยังไม่ได้ผลลัพธ์ตามเป้าหมายสำหรับวิถี (1, 1)

จากขั้นตอนที่พิจารณาผลเฉลยของวิถีในช่วงต่างๆ ของตาราง ซึ่งไม่เป็นเซลล์วงกลม จะรู้ว่าผลเฉลยที่ได้เป็นผลเฉลยเป็นผลเฉลยตามเป้าหมายแล้วหรือยัง ถ้ายังไม่ได้ผลเฉลยตามเป้าหมาย เราต้องดำเนินการอีกขั้นตอนหนึ่งเพื่อการจัดเคลื่อนย้ายเซลล์วงกลม จากส่วนที่เราจัดสรรให้ไม่ถูกต้อง ให้เพิ่มค่า สำหรับเซลล์ไม่เป็นวงกลม และกฎของเซลล์วงกลมเดิม โดยมีขั้นตอนดังนี้

พิจารณาเซลล์ไม่เป็นวงกลมซึ่งแสดงผลเฉลยให้รู้ว่า ยังไม่ได้เป็นไปตามเป้าหมายโดยให้พิจารณาค่าที่แสดงได้ชัดเจน เช่น ค่าลบมากที่สุดสำหรับเป้าหมายต่ำสุด หรือค่าบวกมากที่สุดสำหรับเป้าหมายสูงสุด การเคลื่อนย้ายค่า X_{ij} จากเซลล์วงกลม จะใช้วิถีของเซลล์ไม่เป็นวงกลม นั้นๆ ซึ่งมีเครื่องหมายบวก (+) และสลับเครื่องหมายสำหรับจุดยอดต่อๆ มาโดยใช้วิธีเพิ่มจำนวน X_{ij} สำหรับจุดยอดที่เป็นบวก และลดจำนวน X_{ij} สำหรับจุดยอดที่เป็นลบ ซึ่งค่า X_{ij} คือจำนวนที่จัดสรรให้จากจำนวน X_{ij} ซึ่งมีเครื่องหมายเป็นลบในวิถี และมีค่าน้อยที่สุด

ตัวอย่างการเคลื่อนย้าย X_{ij} สำหรับการจัดสรร X_{ij} ที่ยังไม่ได้ผลเฉลยตามเป้าหมายมีดังนี้

แผนภาพที่ 2.4 การจัดสรร x_{ij} เพื่อให้ได้ผลเฉลยตามเป้าหมาย

ตัวอย่าง ก. ได้ผลเฉลยตามเป้าหมายแล้ว จึงไม่ต้องเคลื่อนย้ายจำนวน x_{ij} อีก
ตัวอย่าง ข.

7	4	8	5	5	4
13	3	-2	2		6



12	4	8	5		4
8	3	-2	2	5	6

วิถี (2,3) มี $X_{ij} = X_{13} = 5$

ดังนั้นค่าของจุดยอดต่างๆ ตามเครื่องหมายในวิถีจะเพิ่มขึ้น

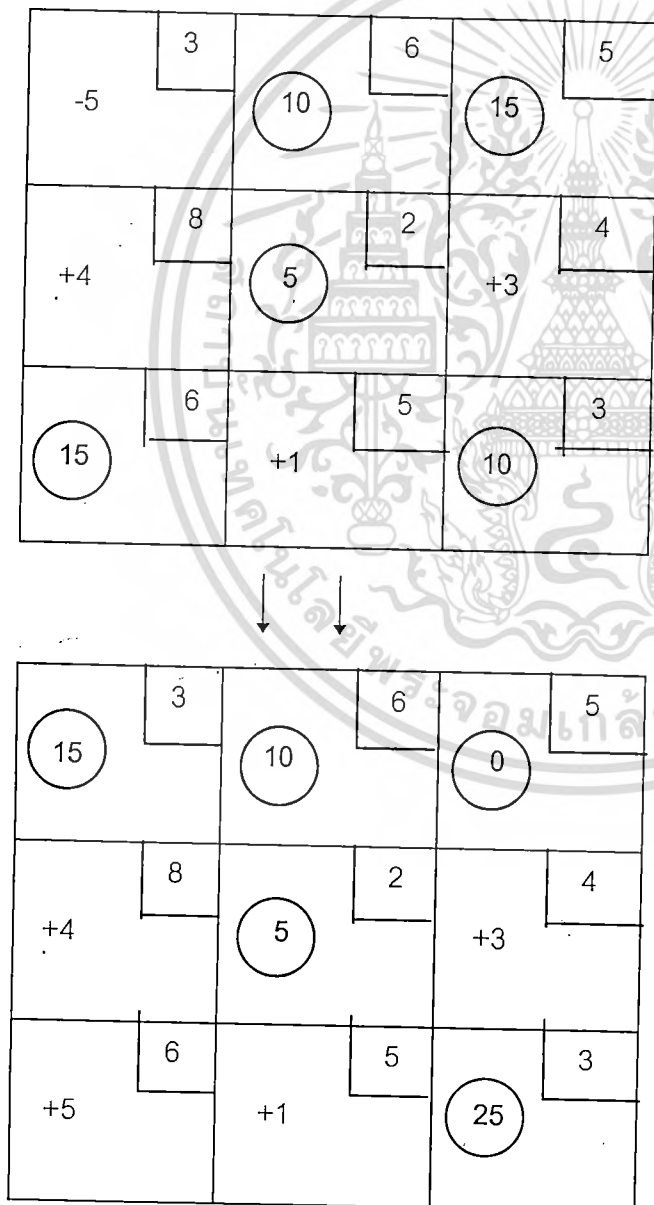
และลดลงตามจำนวน $X_{ij} = X_{13}$ คือ 5

$$X_{23} = 0 + 5 = 5 \quad X_{11} = 7 + 5 = 12$$

$$X_{23} = 5 - 5 = 0 \quad X_{21} = 13 - 5 = 8$$

ส่วน X_{22} ไม่เปลี่ยนแปลงเพราะเป็นเพียงทางผ่านเท่านั้น

ตัวอย่าง ค



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธี (1,1) มี $X_{ij} = X_{13}$ คือ $X_{31} = 5$

$$X_{11} = 0 + 15 = 15 \quad X_{33} = 10 + 15 = 25$$

$$X_{13} = 15 - 15 = 0 \quad X_{31} = 15 - 15 = 0$$

ส่วน $X_{12} = 10$ และ $X_{22} = 5$ เหมือนเดิม

เมื่อตรวจสอบผลเฉลยที่ได้แล้วมีค่าวิธีต่างๆ ของเซลล์ไม่เป็นวงกลมเป็นบวก

จึงสรุปได้ว่าผลเฉลยตามจุดมุ่งหมายเป้าหมายที่วางไว้

2.3.2.2 วิธีโมได

(Modified Distribution Method)

วิธีโมได เป็นวิธีหาคำตอบที่ดีที่สุดอีกวิธีหนึ่ง ซึ่งมีวิธีการคล้ายกับวิธีสตีปิงสไตน์ตรงที่ เป็นวิธีที่มีการคิดประเมินจากต้นทุนการขนส่งที่เกิดขึ้น แต่ต่างกันเล็กน้อยในการคำนวณดัชนีเพื่อพัฒนา ซึ่งวิธีโมไดจะมีการเขียนลูกศรแสดงวิธีการเปลี่ยนแปลงเพียงวงเดียวซึ่งจะเขียนขึ้นหลังจากการคำนวณดัชนีเพื่อพัฒนาคำตอบและเลือกตัวที่มีค่าน้อยที่สุดออกมาแล้ว

การหาคำตอบที่ดีที่สุดด้วยวิธีโมได จะเริ่มต้นหลังจากได้มีการหาคำตอบเบื้องต้นแล้ววิธีนี้มีขั้นตอนพอสรุปได้ดังนี้

1. ตรวจสอบจำนวนช่องที่ได้มีการเติมตัวเลขแล้วว่าเท่ากับจำนวนแถวบวกด้วยจำนวนสดมภ์แล้วลบหนึ่งหรือไม่ ถ้าเท่ากันให้ดำเนินการตาม ข้อ 2 ต่อ แต่ถ้าน้อยกว่าแสดงว่าเกิดดีเจเนอเรซี (Degeneracy)

2. คำนวณหา R_i และ K_j โดยกำหนดให้

R_i เป็นค่าของแถวที่ i

K_j เป็นค่าของสดมภ์ที่ j

C_{ij} เป็นต้นทุนการขนส่ง

มูลค่าของแต่ละแถวและแต่ละสดมภ์จะคำนวณจากสูตร $C_{ij} = R_i + K_j$

สูตรนี้ใช้ได้เฉพาะสไตน์เซลล์ที่ได้จากคำตอบเบื้องต้นเท่านั้น และเนื่องจากจำนวนสไตน์เซลล์โดยทั่วไปจะเท่ากับ จำนวนแถว + จำนวนสดมภ์ - 1 จะเห็นว่า สไตน์เซลล์มีจำนวนน้อยกว่าจำนวนแถวบวกจำนวนสดมภ์อยู่หนึ่ง จึงทำให้สมการที่จะแทนค่าเพื่อหามูลค่าของแต่ละแถวและแต่ละสดมภ์มีค่าน้อยกว่าจำนวนแถวและจำนวนสดมภ์อยู่ 1 ตัว ถ้าจะหาตัวแปรที่ต้องสมมุติค่าตัวแปรขึ้นมาตัวหนึ่งก่อน ดังนั้นเพื่อให้ง่ายต่อการคำนวณจะสมมุติให้ $R_1 = 0$ แล้วแทนค่าในสมการเพื่อหาค่า R_i และ K_j ทุกค่า ค่า R_i และ K_j ที่คำนวณได้อาจเป็นบวก ลบ หรือศูนย์ก็ได้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เขียนค่า R_i และ K_j ที่คำนวณได้กำกับในแต่ละแถวแต่ละสดมภ์
4. คำนวณค่าดัชนีการพัฒนาคำตอบจากวอเตอร์เชลล์ โดยใช้สูตร

$$\text{ดัชนีการพัฒนาคำตอบ} = C_{ij} - R_i - K_j$$

นำผลการคำนวณดัชนีการพัฒนาคำตอบจากวอเตอร์เชลล์ทุกค่ามาเปรียบเทียบกันถ้าดัชนียังเป็นลบแสดงว่ายังสามารถประหยัดต้นทุนได้อีก ต้องพัฒนาต่อไปจนกว่าดัชนีจะมีค่ามากกว่าหรือเท่ากับ ศูนย์ จึงจะแสดงว่าถึงคำตอบที่ดีที่สุดแล้ว

การพัฒนาคำตอบทำโดย

(4.1) เลือกดัชนีการพัฒนาคำตอบที่เป็นลบที่มีตัวเลขน้อยที่สุดหรือค่ามากที่สุด พิจารณาว่าเกิดจากวอเตอร์เชลล์ใด เขียนลูกศร สร้างวงที่จะเปลี่ยนแปลง เริ่มจากวอเตอร์เชลล์นั้น หมุนทวนเข็มนาฬิกาในแนวดิ่งและแนวนอนทำมุม 90 องศาในสโตนเชลล์ใดก็ได้ที่สามารถวกกลับมา ยังจุดเริ่มต้นได้โดยจำนวนมุมในวิถีทางเปลี่ยน-แปลงที่สร้างขึ้นจะเป็นจำนวนคู่เสมอ

(4.2) ใส่เครื่องหมายบวก และเครื่องหมายลบในวิถีทางเปลี่ยนแปลง โดยใส่เครื่องหมายบวกในช่องที่เริ่มต้นช่องถัดมาเป็นลบ แล้วเป็นบวก ลบ สลับกันจนครบวิถีทาง

(4.3) ปริมาณการขนส่งจำนวนน้อยที่สุดในช่องที่มีเครื่องหมายลบ จะเป็นจำนวนสูงสุดที่จะทำการเปลี่ยนแปลงได้ ให้นำจำนวนดังกล่าวมาบวก หรือ ลบกับปริมาณการขนส่งที่มีอยู่เดิมตามเครื่องหมายที่กำกับอยู่ในแต่ละช่อง

5. ดำเนินการขั้นที่ 1 - 4 ซ้ำ ๆ จนกว่าดัชนีการพัฒนาคำตอบจะมีค่ามากกว่าหรือเท่ากับ ศูนย์ ถึงจะถึงคำตอบที่ดีที่สุด

6. คำนวณต้นทุนการขนส่งที่เกิดขึ้นทั้งสิ้น

บทที่ 3

การใช้คอมพิวเตอร์ในการแก้ปัญหาการโปรแกรมเชิงเส้น

3.1 บทบาทของคอมพิวเตอร์

โจทย์หรือตัวแบบการโปรแกรมเชิงเส้นขนาดเล็ก ที่นำมาเป็นแบบฝึกหัดในหนังสือตำรา แม้ว่าจะสามารถหาผลลัพธ์ได้ด้วยปากกากับกระดาษตามวิธีการที่หนังสือหรือตำราได้อธิบายไว้ แต่ในโลกของความเป็นจริง เราอาจไม่มีโจทย์ขนาดเล็กเช่นนี้ให้ลองหาผลลัพธ์ ดังนั้นถ้าปัญหาการโปรแกรมเชิงเส้นที่มีขนาดใหญ่มาก จำเป็นต้องใช้ความอดทนในการคำนวณอย่างมหาศาล ซึ่งคอมพิวเตอร์มีบทบาทช่วยลดภาระดังกล่าวได้ ในยุคปัจจุบันคอมพิวเตอร์ได้ก้าวหน้าอย่างมาก ทั้งด้านอุปกรณ์คอมพิวเตอร์ (Hardware) และโปรแกรมคำสั่งงาน (Software) ตัวแบบการโปรแกรมเชิงเส้นที่มีขนาด 100 ตัวแปรคูณ 100 แถว อาจใช้เวลาในการหาผลเฉลยตามเป้าหมายไม่กี่นาที แต่ถ้าขนาดของตัวแปรเพิ่มขึ้นแล้ว เวลาที่ใช้ในการคำนวณหาค่าผลเฉลยตามเป้าหมาย กลับไม่ได้ เพิ่มขึ้นตามสัดส่วนเชิงเส้น (Linearly) ไปตามขนาดของตัวแปรที่เพิ่มขึ้นกล่าวคือถ้าเพิ่มขนาดของตัวแปรแบบเป็นสองเท่าแล้ว เวลาที่ใช้ในการคำนวณกลับไม่ได้เพิ่มขึ้นเป็นสองเท่า แต่จะเพิ่มขึ้นเป็นสี่เท่า (กำลังสอง) หรือ แปดเท่า (กำลังสาม) อีกปัจจัยหนึ่งที่มีผลโดยตรงต่อความเร็วในการคำนวณของคอมพิวเตอร์คือ ความหนาแน่นของสัมประสิทธิ์ (Matrix Density) ที่ไม่เป็นศูนย์ ถ้ามีความหนาแน่นน้อย (Sparsity) คอมพิวเตอร์ก็จะทำงานได้เร็ว เพราะคอมพิวเตอร์เพียงแต่ตรวจสอบว่าสัมประสิทธิ์เป็นศูนย์หรือไม่ ถ้าเป็นศูนย์ก็ผ่านเลยไปไม่ต้องเสียเวลาในการบวกคูณหาร การใช้คอมพิวเตอร์ในการหาค่าผลเฉลยตามเป้าหมายไม่เหมือนกับงานประมวลผลข้อมูลในธุรกิจโดยทั่วไป ตัวอย่างเช่น การที่เราไปเบิกเงินที่ตู้เอทีเอ็ม (ATM) ถ้าใส่รหัสไม่ถูกหรือไม่มีเงินในบัญชี ก็รู้แน่นอนว่าจะเบิกไม่ได้ ถ้าใส่รหัสถูกและมีเงินในบัญชีพอก็จะเบิกถอนได้ แต่การใช้คอมพิวเตอร์หาค่าผลเฉลยตามเป้าหมายนั้นคอมพิวเตอร์ต้องวิงวามผลเฉลยไปที่ละครั้ง (Iteration) โดยไม่อาจทราบได้ว่า จะพบคำตอบเมื่อใด ซึ่งอาจจะเสียเวลาคำนวณไป 200 ครั้ง (Iteration) แล้วจึงพบว่าไม่มีคำตอบที่เป็นไปได้ (Infeasible) หรือผลลัพธ์แบบไม่มีขอบเขต (Unbounded solution) แทนที่จะเป็นการพบผลลัพธ์ตามเป้าหมาย (Optimal solution) ซึ่งแตกต่างไปจากกรณีของตู้เอทีเอ็มที่โปรแกรมเพียงแต่ตรวจสอบกับเพิ่มข้อมูลของลูกค้ำว่ารหัสตรงกันและมีเงินในบัญชีมากกว่าที่ขอเบิกเท่านั้น ความเร็วในการหาค่าผลเฉลยตามเป้าหมายจึงเป็นเรื่องที่คำนึงถึงเป็นอย่างมาก อย่างไรก็ตามโอกาสความคุ้มจากการใช้เทคนิคการโปรแกรมเชิงเส้น ก็มีมากพอทำทนายกับเงินลงทุน เวลาและความอดทนที่จะต้องใช้ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมีทรัพยากรจำกัดของคอมพิวเตอร์ คอมพิวเตอร์โดยทั่วไปจะประกอบไปด้วย ส่วนประมวลผล (CPU = Central Processing Unit) หน่วยความจำ (Memory) ที่เก็บข้อมูล (Disk Storage or Tape) ซึ่งคอมพิวเตอร์ สามารถประมวลผลสิ่งที่อยู่ในหน่วยความจำ ได้เร็วกว่าการประมวลผลโดยการเรียกใช้ข้อมูลจากที่เก็บข้อมูล ดังนั้นผู้พัฒนาโปรแกรมสำเร็จรูปสำหรับการโปรแกรมเชิงเส้น จึงพยายามที่จะนำเอาเมตริกซ์ของตัวแบบเข้ามาไว้ในหน่วยความจำทั้งหมด แต่เมื่อเมตริกซ์มีขนาดใหญ่มากก็จำเป็นต้องอาศัยการย้ายถ่ายเทเมตริกซ์บางส่วน ที่ยังไม่ได้ทำการคำนวณเก็บฝากไว้ในที่เก็บข้อมูล และถ้าต้องการคำนวณก็เรียกกลับมาใช้ ซึ่งก็จะมีผลต่อความเร็วในการคำนวณอย่างมาก

นอกจากความสามารถของคอมพิวเตอร์ในการคำนวณ ซึ่งเป็นงานซ้ำซาก (Repetitive) ได้อย่างรวดเร็ว (Speed) โดยให้ความเที่ยงตรง (Accuracy) อย่างเสมอต้นเสมอปลาย (Consistency) แทนมนุษย์ได้แล้ว ก็ยังสามารถจัดเก็บข้อมูลในลักษณะฐานข้อมูล คือ ช่วยรวบรวมข้อมูลที่เกี่ยวข้องกับตัวแบบอันเป็นข้อมูลที่ต้องการวิเคราะห์และทำการประมาณค่าต่างๆ ซึ่งต้องใช้วิธีการทางสถิติร่วมด้วย จึงเป็นสิ่งที่ชวนให้ศึกษาอย่างยิ่งว่า จะใช้คอมพิวเตอร์กับปัญหาการโปรแกรมเชิงเส้นได้อย่างไร และสามารถใช้ประโยชน์จากคอมพิวเตอร์ในขั้นตอนใดบ้าง ของการทำงานการแก้ปัญหาโปรแกรมเชิงเส้น ถ้าเป็นธุรกิจขนาดใหญ่ เช่น โรงกลั่นน้ำมัน โรงงานผสมอาหารสัตว์ บริษัทจัดส่งสินค้า ซึ่งมีกิจกรรมที่ต้องดำเนินการหาผลเฉลยตามเป้าหมายเพื่อช่วยในการตัดสินใจจนกลายเป็นงานประจำ (Routine) ภายใต้ความเปลี่ยนแปลงของโลกธุรกิจที่ทำให้ตัวสัมประสิทธิ์เปลี่ยนแปลงไป ตลอดจนตัวแบบก็สามารถเปลี่ยนแปลงได้ด้วยแล้วจะสามารถนำคอมพิวเตอร์มาช่วยได้อย่างไรบ้างนั้น นอกจากนี้ยังมีคนหรือหน่วยงานที่เกี่ยวข้องกับทั้งสัมประสิทธิ์และตัวแบบหลายหน่วยงาน เช่น หน่วยงานจัดหาวัตถุดิบ หน่วยงานผลิต หน่วยงานควบคุมคุณภาพ หน่วยงานด้านการขาย การบุคคล ซึ่งต้องเป็นผู้ส่งข้อมูลเข้ามากลายเป็นสัมประสิทธิ์ในตัวแบบแล้ว องค์กรเหล่านี้จะดำเนินการอย่างไร จึงจะเป็นระบบงานคอมพิวเตอร์ที่สามารถรองรับลักษณะงานการแก้ปัญหาการโปรแกรมเชิงเส้นได้ วิธีการของนักคอมพิวเตอร์คือ การใช้เทคนิคของการวิเคราะห์และออกแบบระบบงาน (System Analysis and Design Techniques)

3.2 แนวความคิดเกี่ยวกับระบบงานคอมพิวเตอร์

ระบบงานคอมพิวเตอร์โดยทั่วไปหมายถึง การประสานการปฏิบัติงานร่วมกันของอุปกรณ์คอมพิวเตอร์ (Hardware) โปรแกรมคำสั่งงาน (Software) และคนที่ต้องปฏิบัติงานทั้งในฐานะผู้ใช้งาน (User) และผู้สนับสนุนการใช้งาน (System Analyst และ Programmer) เพื่อทำการจัดเก็บข้อมูล ประมวลผลข้อมูลสนับสนุนการปฏิบัติงานของบุคคลหรือกลุ่มบุคคลดังกล่าว ตามวัตถุประสงค์ที่ได้จัดตั้งขึ้น

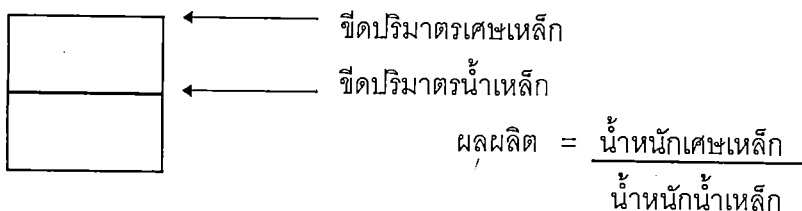
ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตัวปัญหาการโปรแกรมเชิงเส้นมีองค์ประกอบสำคัญสองส่วน คือสัมประสิทธิ์ของตัวแบบ และ ตัวแบบของตัวปัญหา ซึ่งสัมประสิทธิ์ของตัวแบบมักจะมีค่าเปลี่ยนแปลงมากกว่าตัวแบบ ลักษณะของความเปลี่ยนแปลงและสาเหตุ เป็นดังนี้

การเปลี่ยนแปลงของสัมประสิทธิ์ของตัวแบบมีสาเหตุมาจากปัจจัยต่างๆ มากมาย ได้แก่ ความเปลี่ยนแปลงของอุปสงค์และอุปทานกระทบถึงภาวะด้านปริมาณ ราคาและต้นทุนการผลิต การเสื่อมสภาพเครื่องจักร การเปลี่ยนแปลงของเทคโนโลยีการผลิตที่ส่งผลกระทบต่อทั้งทางตรงและทางอ้อม ต่ออัตราการใช้ทรัพยากร เงื่อนไขข้อกำหนดมาตรฐานของผู้ผลิตเองหรือมาตรฐานอุตสาหกรรมที่กำหนดขึ้นใช้นอกจากนี้คุณสมบัติที่แตกต่างกัน สำหรับด้านการทดสอบต่างๆ วิธีการทดสอบวัดค่าต่าง ๆ รวมทั้งวิธีการประมาณค่าที่ยังแตกต่างกันออกไปตามพื้นฐานแนวคิดและทฤษฎี

ในวิธีการมองปัญหาของผู้บริหาร ซึ่งอาจเกิดได้จากกลยุทธ์ในการดำเนินธุรกิจ รวมถึงความแตกต่างของตัวบุคคลที่เป็นผู้พิจารณาปัญหานั้น ๆ ด้วย เช่น การทำให้มีต้นทุนต่ำสุดกับการให้โรงงานมีประสิทธิภาพสูงสุดก็มีนัยที่แตกต่างกัน เช่น ในเตาหลอมเหล็ก* ที่เป็นแบทช์ (Batch Process) การจะผสมน้ำเหล็กแต่ละเตาให้มีต้นทุนต่ำที่สุด อาจทำให้ประสิทธิภาพของเตาหลอมลดลง เพราะการใช้เศษเหล็กที่มีคุณภาพต่ำจะให้เนื้อที่ปริมาตรของเตาหลอมมาก (ปริมาตรของเศษเหล็กนั้นก็วัดได้ยากเพราะรูปร่างของเศษเหล็กไม่แน่นอน) เพื่อต้องการให้ต้นทุนการผลิตต่ำ แต่ผลผลิต (Yield) จากเศษเหล็กมาเป็นน้ำเหล็กจะได้น้อย ซึ่งในหนึ่งแบทช์อาจได้น้ำเหล็กน้อยกว่าที่ควรจะเป็น ฉะนั้นตัวแบบในปัญหาการผสมเศษเหล็กเพื่อผลิตน้ำเหล็ก (Liquid Steel) จำเป็นต้องเลือกว่า จะผสมเศษเหล็กให้มีต้นทุนต่ำสุดที่ประสิทธิภาพเตาหลอมระดับใด หรืออาจจะสร้างตัวแบบในลักษณะไหน จะผลิตน้ำเหล็กที่ประสิทธิภาพสูงสุดโดยมีข้อจำกัดว่าจะต้องมีต้นทุนไม่เกินที่บาทต่อตันของน้ำเหล็ก กล่าวได้ว่าฟังก์ชันเป้าหมายของการสร้างตัวแบบอันหนึ่งกลายเป็นสมการข้อจำกัดของการสร้างตัวแบบอีกอันหนึ่งก็ได้ ทั้งนี้จะสร้างตัวแบบลักษณะใดขึ้นอยู่กับภาวะแวดล้อมทางธุรกิจ ณ เวลาใดเวลาหนึ่งด้วย

* เตาหลอมเหล็ก (Electric arc furnace)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โลกที่เป็นจริง (Real world)

ข้อมูลจากการปฏิบัติงาน

เงื่อนไขทำให้ต้องเปลี่ยนแปลง

หรือ

ตัวแบบ

การทดสอบต่างๆ

วิธีการมองปัญหา

1. การเก็บรวบรวม
ข้อมูลและการประ
มาณค่าพารามิเตอร์

2. การสร้างตัวแบบ

เพิ่มข้อมูลพารามิเตอร์

เพิ่มข้อมูลรูปแบบของตัวแบบ

การสร้างเมตริกซ์
(Matrix generator)

เพิ่มข้อมูลของเมตริกซ์

[A]

[B]

รูปที่ 3.1 ลักษณะการปฏิบัติงานในระบบงานแก้ปัญหาคำถามเชิงเส้น โดยขั้นตอนการคำนวณที่ 1 และ 2 นั้นเป็นการดำเนินการที่ซ้ำกันอยู่ตลอดเวลา แม้ว่าการคำนวณที่ 1 นั้นจะมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 เป็นรูปที่แสดงถึงขั้นตอนการปฏิบัติงาน ที่เป็นอยู่ในองค์กรที่ต้องใช้การโปรแกรมเชิงเส้น ซึ่งโอกาสที่จะใช้ระบบงานคอมพิวเตอร์เข้าช่วยการปฏิบัติงานนั้นมีดังนี้

3.2.1 การเก็บรวบรวมข้อมูลจากสนามและการประมาณค่าพารามิเตอร์

เป็นการใช้คอมพิวเตอร์เข้าช่วยในการจัดเก็บข้อมูล (Data) ที่เกิดมาจากการปฏิบัติงาน ในอดีต การรายงานทดสอบคุณภาพคุณสมบัติต่าง ๆ ของวัตถุดิบต่าง ๆ กัน อัตรา / ราคา / ต้นทุนที่ได้จากข่าวสารการค้าใบสั่งซื้อ ซึ่งต้องมีการนำข้อมูลเหล่านี้มาแปลงเป็นข่าวสาร (Information) หรือหาค่าตัวแทนทางสถิติที่เหมาะสมด้วยโปรแกรมคอมพิวเตอร์สำเร็จรูปทางสถิติ หรือโปรแกรมคอมพิวเตอร์ที่จะพัฒนาขึ้นมาใช้เอง ซึ่งก็จะได้รายงานที่ผู้ปฏิบัติงานจะนำไปใช้งานและ แฟ้มข้อมูล (Files) ที่คอมพิวเตอร์พร้อมจะนำไปใช้งานต่อไป

3.2.2 การสร้างตัวแบบ (Modeling)

เป็นการกำหนดรูปแบบของวิธีการสร้างตัวแบบเป็นการเฉพาะสำหรับปัญหาการโปรแกรมเชิงเส้นที่ต้องการ กล่าวคือเป็นโปรแกรมที่กำหนดกฎเกณฑ์ว่าจะจัดการข้อมูลให้กลายเป็นสัมประสิทธิ์ในตัวแบบได้อย่างไร และยังสามารถเก็บคำสั่งงานที่จะให้โปรแกรมหาค่าผลเฉลยตามเป้าหมาย เลือกส่วนประกอบของตัวแบบมาคำนวณให้ได้ด้วย เช่น อาจเตรียมค่าทางขวามือไว้หลายชุด หรือสมการข้อจำกัดหลาย ๆ แบบตามเงื่อนไขที่ต้องการ ซึ่งก็เป็นเช่นเดียวกับการวิเคราะห์ความไวโดยไม่ต้องรอให้ต้องมีการคำนวณผลลัพธ์จนได้ผลลัพธ์หรือค่าผลเฉลยตามเป้าหมายก่อน

3.2.3 การสร้างเมตริกซ์ (Matrix Generator)

เป็นการนำเอาข้อมูลพารามิเตอร์ที่จัดเก็บไว้ มาสร้างเป็นเมตริกซ์ (ตารางซิมเพล็กซ์เริ่มต้น) ตามรูปแบบของวิธีการสร้างตัวแบบที่กำหนดไว้ ผลที่ได้รับต้องอยู่ในรูปแบบ (Format) ที่สอดคล้องกับรูปแบบที่โปรแกรมคำนวณหาค่าผลเฉลยตามเป้าหมาย (Optimizer) ต้องการใช้งาน ความสามารถที่สำคัญของโปรแกรมสร้างเมตริกซ์ คือ การกำหนดให้ข้อมูลที่เป็นเรื่องเดียวกันจัดรวมเป็นกลุ่มก้อนเดียวกัน (Class) ซึ่งในกลุ่มจะมีชื่อของสมาชิก (Member) ในลักษณะของพจนานุกรมของข้อมูล (Data Dictionary) ที่เป็นสิ่งที่เกี่ยวข้องในปัญหาการโปรแกรมเชิงเส้น เช่น ในปัญหาการขนส่งจะมีกลุ่มข้อมูลมาเชื่อมต่อกัน (Combination) แล้วจะเกิดเป็นตัวแปรที่หมายถึงการส่งสินค้าจากโรงงานไปยังแหล่งปลายทางทั้งหมดได้โดยอัตโนมัติ และเมื่อมีโรงงานเพิ่มขึ้นก็เพียงแค่เพิ่มชื่อสมาชิกลงในในกลุ่มข้อมูล นั้นๆ และข้อมูลที่เกี่ยวข้องอัตโนมัติด้วยเช่นกัน

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 การหาค่าผลเฉลยตามเป้าหมาย (Optimizer)

เป็นการนำข้อมูลจากการสร้างเมตริกซ์ไปคำนวณ ซึ่งโปรแกรมคอมพิวเตอร์สำเร็จรูป (Computer package) ทางด้านการโปรแกรมเชิงเส้นมักจะใช้วิธีซิมเพล็กซ์ แต่ในการคำนวณด้วยการใช้เมตริกซ์ (Revised simplex method) อย่างไรก็ตามวิธีการใหม่ คือ วิธีคาร์มาร์คาร์ (Karmarkar's method) ที่ได้เริ่มคิดค้นตั้งแต่ปี ค.ศ. 1984 มีความสามารถคำนวณค่าผลเฉลยเบื้องต้นได้เร็วกว่าวิธีซิมเพล็กซ์ แต่ก็ยังไม่สามารถคำนวณด้วยคอมพิวเตอร์ได้ ตามที่ กิลเบิร์ต สตรอง (Gillbert Strang) อาจารย์คณิตศาสตร์ที่สถาบันเอ็มไอที กล่าวไว้ในวารสาร Siam News เดือนพฤศจิกายน ค.ศ. 1985 บัดนี้ วิธีคาร์มาร์คาร์ ได้กลายเป็นโปรแกรมสำเร็จรูปออกจำหน่ายในชื่อ Korbx โดยบริษัทเอทีแอนด์ที (AT&T) ด้วยราคาสูงถึง 8.9 ล้านดอลลาร์สหรัฐ (ราคาเมื่อมกราคม 2532) กลุ่มลูกค้าเป้าหมายได้แก่บริษัทขนส่ง บริษัทน้ำมัน และโรงงานอุตสาหกรรม

3.2.5 การจัดทำรายงาน (Report Writer)

เป็นการนำเอาค่าผลเฉลยเบื้องต้นมาจัดทำรายงานในรูปแบบที่ผู้บริหาร ซึ่งอาจจะไม่มีพื้นฐานด้านการวิจัยดำเนินงาน (Operations Research) สามารถอ่านผลลัพธ์จากตารางสุดท้ายได้

3.2.6 การตีความรายงานและการตัดสินใจ (Interpretation & Decision)

เป็นสิ่งที่ผู้บริหารจะต้องนำเอาประสบการณ์มาร่วมตัดสินใจด้วย เพราะการโปรแกรมเชิงเส้นเป็นเรื่องของข้อมูลเชิงปริมาณและพารามิเตอร์ของตัวแบบ ก็ต้องเป็นค่าคงที่ (Deterministic) การวิเคราะห์ความไวจะมีส่วนสำคัญในการสร้างทางเลือก ให้ผู้บริหารตัดสินใจต่อสถานการณ์ที่คาดว่าจะเกิดขึ้นในอนาคต

3.2.7 การนำไปปฏิบัติ (Action)

เนื่องจากคำตอบผลลัพธ์จากการโปรแกรมเชิงเส้นมีลักษณะที่เป็นข้อมูลเชิงปริมาณ ไม่ได้มีส่วนของความรู้สึกเข้ามาเกี่ยวข้องด้วย ทางเลือกที่ผู้บริหารได้ตัดสินใจเลือกให้ดำเนินการ อาจจะมีขึ้นกับความเคยชินในการปฏิบัติงานของผู้ปฏิบัติงานได้ เช่น ถ้าเป็นการวางแผนการตัดเหล็กแท่งที่สั่งซื้อจากต่างประเทศมา เพื่อใช้ในการก่อสร้างอาคาร โดยต้องทำการตัดให้ได้จำนวนท่อนและขนาดความยาวจากท่อนที่มีขนาดและความยาว แตกต่างกันตามที่ได้ผลลัพธ์จากการโปรแกรมเชิงเส้น จึงจะมีค่าใช้จ่ายน้อยที่สุด แต่ผู้ปฏิบัติกลับเลือกตัดเหล็กแท่งตามความสะดวกที่จะนำออกมาจากลานกองเก็บแล้วก็ทำให้ไม่สามารถตัดเหล็กแท่งได้จำนวนท่อนและขนาดความยาวที่ต้องการ การสั่งซื้อเหล็กแท่งจากต่างประเทศต้องใช้เวลารอคอยพอสมควร ทำให้โครงการก่อสร้างทั้งโครงการไม่สามารถไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วเสร็จกำหนดตามเวลา ก็จะก่อให้เกิดผลเสียหายได้ กล่าวได้ว่าการควบคุมการปฏิบัติในรายละเอียดนั้น นับเป็นเรื่องสำคัญยิ่งต่อการดำเนินงานขององค์กร เพื่อมุ่งไปสู่ผลเฉลยตามเป้าหมายไม่ว่าจะเป็นกำไรสูงสุด หรือค่าใช้จ่ายน้อยที่สุดก็ตาม

กล่าวได้ว่าคอมพิวเตอร์เข้าไปมีบทบาทได้ในทุกขั้นตอน การที่ผู้ใช้คอมพิวเตอร์ต้องการเกิดความเป็นอัตโนมัติ (Automation) มากเพียงใด ขึ้นอยู่กับความสามารถในการลงทุนและบุคลากรคอมพิวเตอร์ที่จะช่วยเสาะแสวงหาหรือพัฒนาโปรแกรม (Application program) ที่สามารถสนับสนุนความต้องการดังกล่าว และความใฝ่ฝันที่จะมีระบบงานคอมพิวเตอร์ที่ชาญฉลาดช่วยจำกฎเกณฑ์ต่าง ๆ (Expert system or Artificial Intelligent) ให้สามารถรองรับความเปลี่ยนแปลงของพารามิเตอร์และตัวแบบได้เองนั้น ยังคงค่อนข้างจะไกลเกินความเป็นจริงในโลกของการแก้ปัญหาการโปรแกรมเชิงเส้น

3.3 โปรแกรมสำเร็จรูปสำหรับการโปรแกรมเชิงเส้น

ขนาดของตัวแบบของปัญหาการโปรแกรมเชิงเส้น เป็นตัวบ่งชี้ว่าควรใช้ เครื่องคอมพิวเตอร์ขนาดใดเพื่อใช้ในการแก้ปัญหาดังกล่าว แต่ในปัจจุบันเป็นการยากที่จะจำแนกขนาดของคอมพิวเตอร์ เพราะความก้าวหน้าทางเทคโนโลยีที่ทำให้มีคอมพิวเตอร์ที่มีขีดความสามารถสูงขึ้นตลอดเวลา โดยทั่วไปในวงการคอมพิวเตอร์ยังคงจัดกลุ่มของคอมพิวเตอร์ออกเป็น 3 กลุ่มใหญ่ คือ

- ไมโครคอมพิวเตอร์ (Microcomputer) : ลักษณะสำคัญ คือ มีผู้ใช้งานคนเดียว (Single User) สนองความต้องการส่วนบุคคล อาจจะมีความสามารถในการทำงานหลายๆ งานพร้อมกันก็ได้ (Multitasking)

- มินิคอมพิวเตอร์ (Minicomputer) สามารถใช้งานได้หลาย ๆ คนพร้อมกัน (Multiusers) มักจะใช้สนองความต้องการระดับหน่วยงาน (Department) ในองค์กรขนาดใหญ่หรือใช้กับงานของธุรกิจขนาดเล็ก โดยปกติไม่จำเป็นต้องมีนักโปรแกรมการควบคุมระบบงาน (System Programmer) ก็ได้

- เมนเฟรมคอมพิวเตอร์ (Mainframe Computer) เป็นคอมพิวเตอร์ขนาดใหญ่ใช้สนองความต้องการของทั้งองค์กร มักจะใช้เป็นแกนของระบบงานคอมพิวเตอร์ทั้งหมด (Core corporate system)

โปรแกรมสำเร็จรูปที่มีการนำเข้ามาใช้กันในประเทศไทย ทั้งในอุตสาหกรรมการกลั่นน้ำมัน และบริษัทในเครือซิเมนต์ไทย ซึ่งต้องใช้ความสามารถของเครื่องเมนเฟรมคอมพิวเตอร์ ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของบริษัท ฮาร์เวอร์ลีย์ซิสเต็ม จำกัด (Harverly System, Inc) ซึ่งมีลูกค้าส่วนใหญ่เป็นบริษัทน้ำมัน โปรแกรมสำเร็จรูปที่สำคัญของบริษัทนี้คือ โปรแกรมที่ใช้ในการจัดทำเมตริกซ์และการจัดทำรายงานที่มีชื่อว่า "เมเจน" (MAGEN = Matrix Generator) และโปรแกรมคำนวณผลเฉลยตามเป้าหมายมีชื่อว่า "เฮชเอส / แอลพี" (HS / LP = Harverly System/Linear Programming) ความสามารถที่จะประมวลผลกับตัวแบบขนาดใดขึ้นอยู่กับขนาดของหน่วยความจำ (Memory) ซึ่งในเมนเฟรมคอมพิวเตอร์เราสามารถแบ่งส่วนความจำออกเป็นส่วนย่อย เพื่อให้ประมวลผลหลายๆโปรแกรมได้พร้อมๆกัน ที่แบ่งสรรให้กับโปรแกรม (Partition size) และโปรแกรมควบคุมระบบงาน (Operating System) สำหรับเมเจนและเฮชเอส / แอลพีรุ่นที่ 4 (ออกจำหน่ายเมื่อประมาณปี 2523) สามารถประมวลผลตัวแบบที่เป็นเมตริกซ์ขนาดสูงสุดถึง 25,990 ตัวแปร \times 4,095 แถว

นอกจากนี้โปรแกรมที่มีขีดความสามารถสูงกว่าเมเจน คือ "ออมนิ" (OMNI) ก็ยังได้ถูกนำเข้ามาใช้ร่วมกับโปรแกรมคำนวณผลเฉลยตามเป้าหมายคือ "เอ็มพีเอสเอ็กซ์" (MPSX = Mathematical Programming System Extension) ของบริษัทไอบีเอ็ม จำกัด (IBM = International Business Machine Corporation) ด้วย โดยออมนินำหน้าที่สร้างเมตริกซ์ให้กับเอ็มเอสเอ็กซ์ และนำตารางผลเฉลยตามเป้าหมายไปทำรายงาน การใช้งานโปรแกรมเหล่านี้ยังคงมีลักษณะเป็นแบทช์ (Batch) กล่าวคือ ผู้ใช้โปรแกรมเมื่อเตรียมข้อมูลแล้วเสร็จก็ต้องส่งข้อมูลให้เครื่องคอมพิวเตอร์ (Operator) หรือผู้ใช้โปรแกรมจะเป็นผู้ส่งงานทั้งหมดเข้าไปประมวลผลก็ได้เมื่อเครื่องคอมพิวเตอร์ประมวลผลแล้วเสร็จจึงจะได้ผลลัพธ์

สำหรับเมนเฟรม ออมนิ และ เฮชเอส / แอลพี ยังสามารถใช้งานบนเครื่องมินิคอมพิวเตอร์อีกหลายยี่ห้อเช่น ดิจิตอล (Digital) ฮิวเลตต์แพคการ์ด (Hewlette Packard) ดาตาร์เจเนอรัล (Data General) ด้วย

สำหรับในวงการศึกษาก็ได้มีการนำโปรแกรมสำเร็จรูปลินโด (Lindo = Linear Interactive and Discrete Optimizer) ซึ่งเป็นผลงานของ ลินุส ชร่าช (Linus Schrage) แห่งมหาวิทยาลัยชิคาโก (University of Chicago) มาใช้งานบนไมโครคอมพิวเตอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบโปรแกรมสำหรับปัญหาการขนส่ง

4.1 การออกแบบ Flow Chart

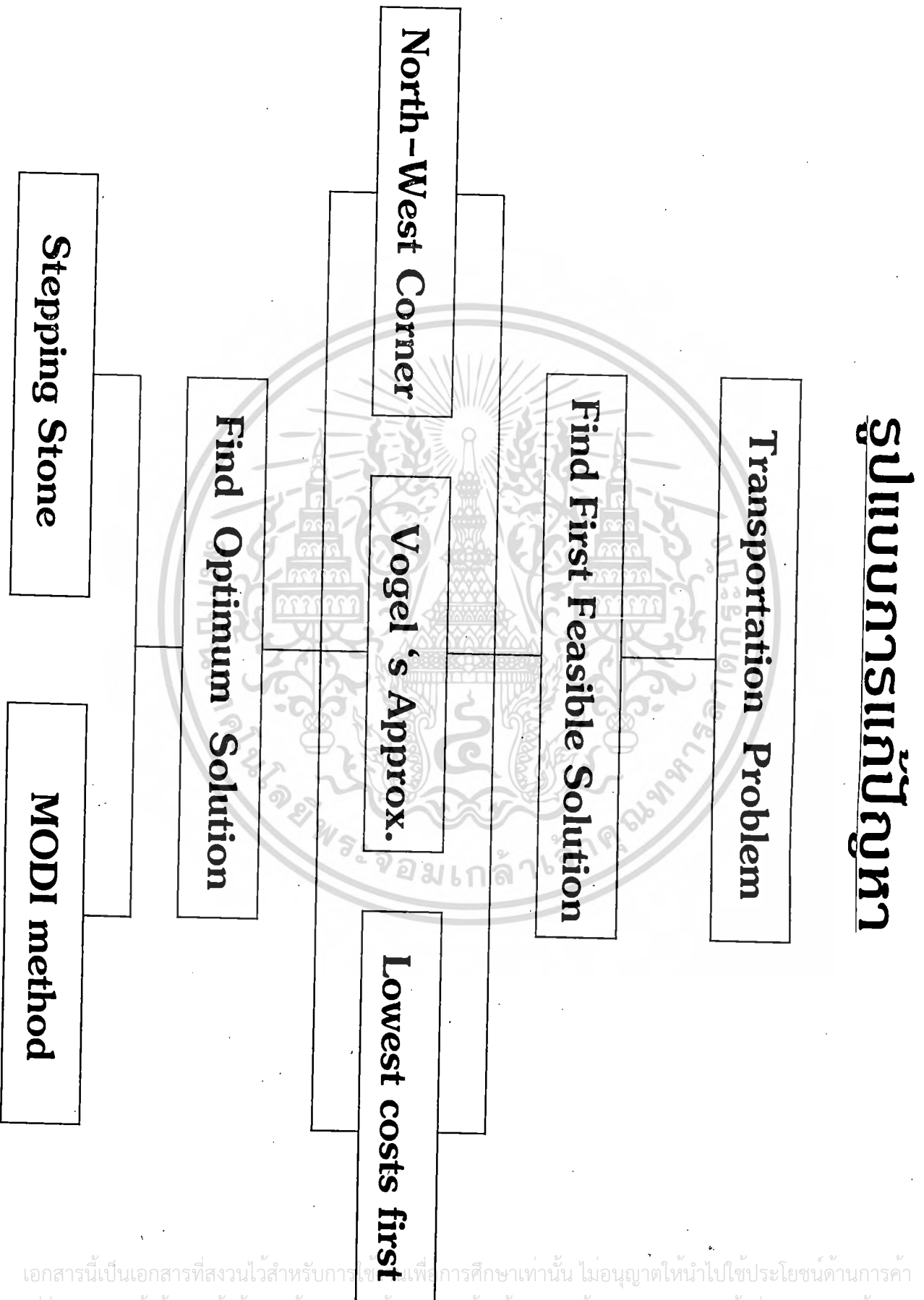
ในการออกแบบ Flow Chart ของปัญหาการขนส่งนี้ ประกอบไปด้วย

- Flow Chart ของรูปแบบการแก้ปัญหาการขนส่ง
- Flow Chart ของระบบคอมพิวเตอร์ในโปรแกรมการแก้ปัญหาการขนส่ง
- Flow Chart ของ North - West Corner Rule Method
- Flow Chart ของ Lowest First Costs Rule Method
- Flow Chart ของ Vogel 's Approximation Method
- Flow Chart ของ Stepping Stone Method
- Flow Chart ของ Modified Distribution Method

ซึ่งแสดงดังต่อไปนี้

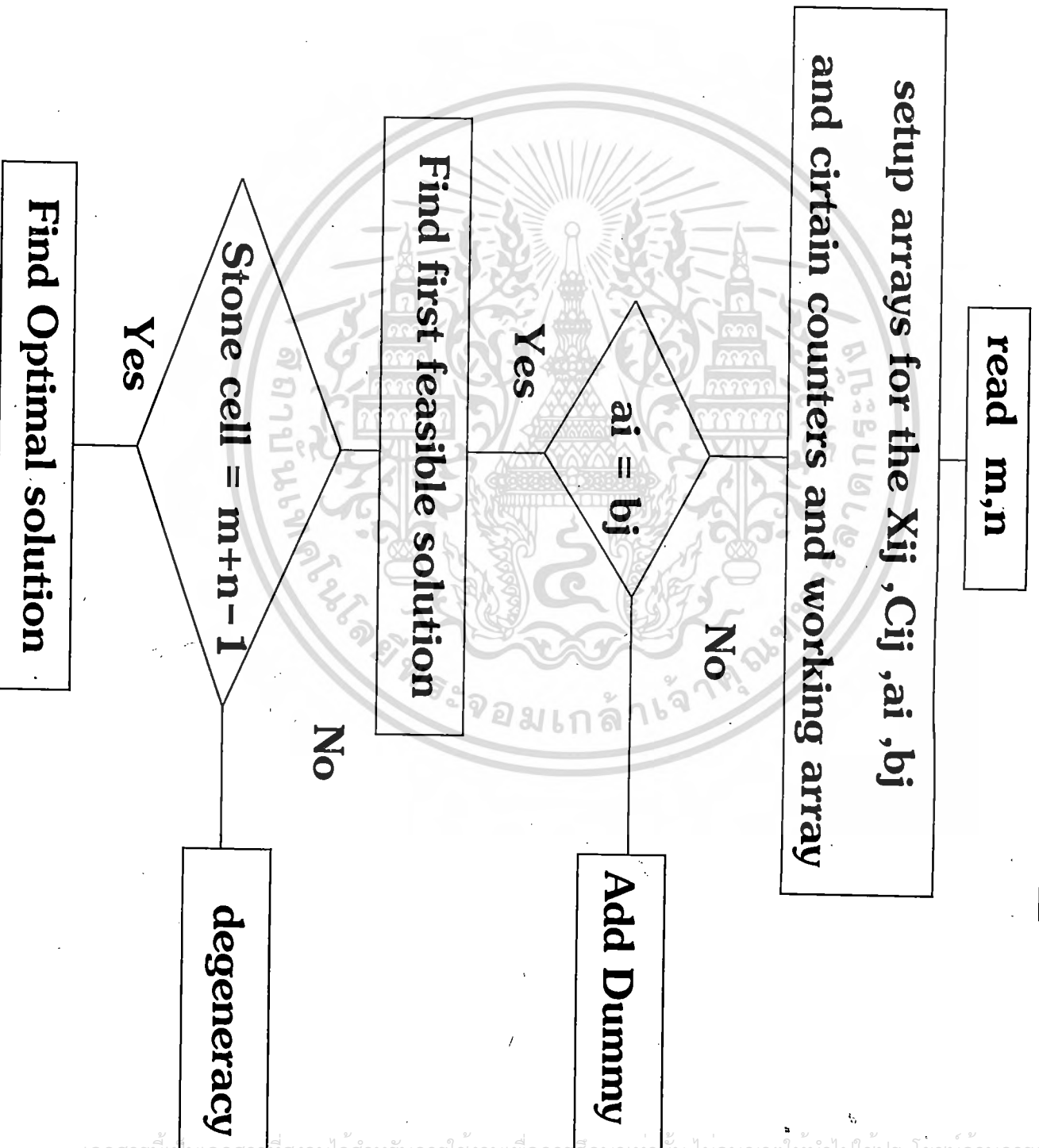


แผนภาพที่ 4.1 Flow Chart ของรูปแบบการแก้ปัญหาการขนส่งภายในโปรแกรม



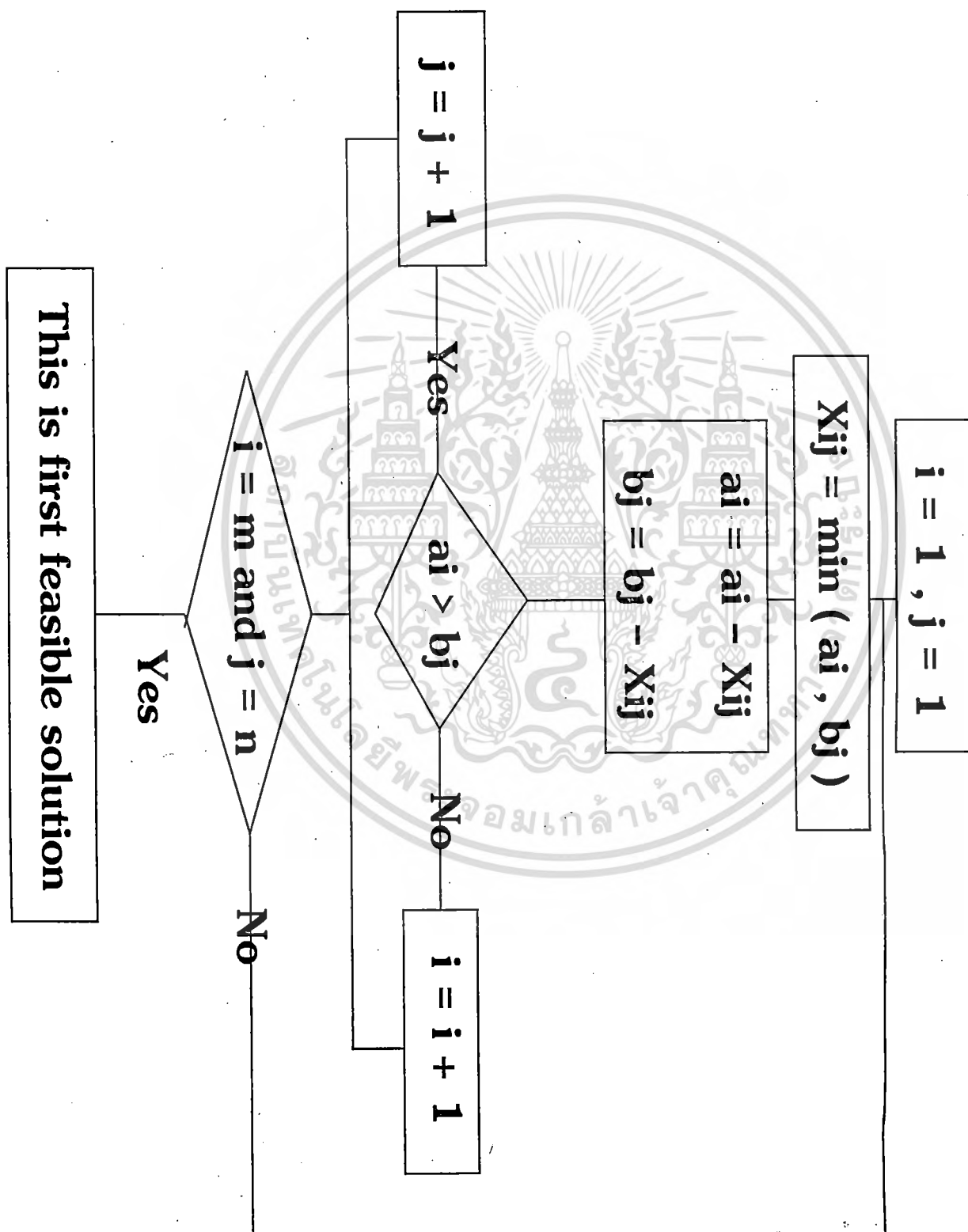
แผนภาพที่ 4.2 Flow Chart ของระบบคอมพิวเตอร์ในโปรแกรมการแก้ปัญหาการขนส่ง

Flow Chart for Transportation Problem



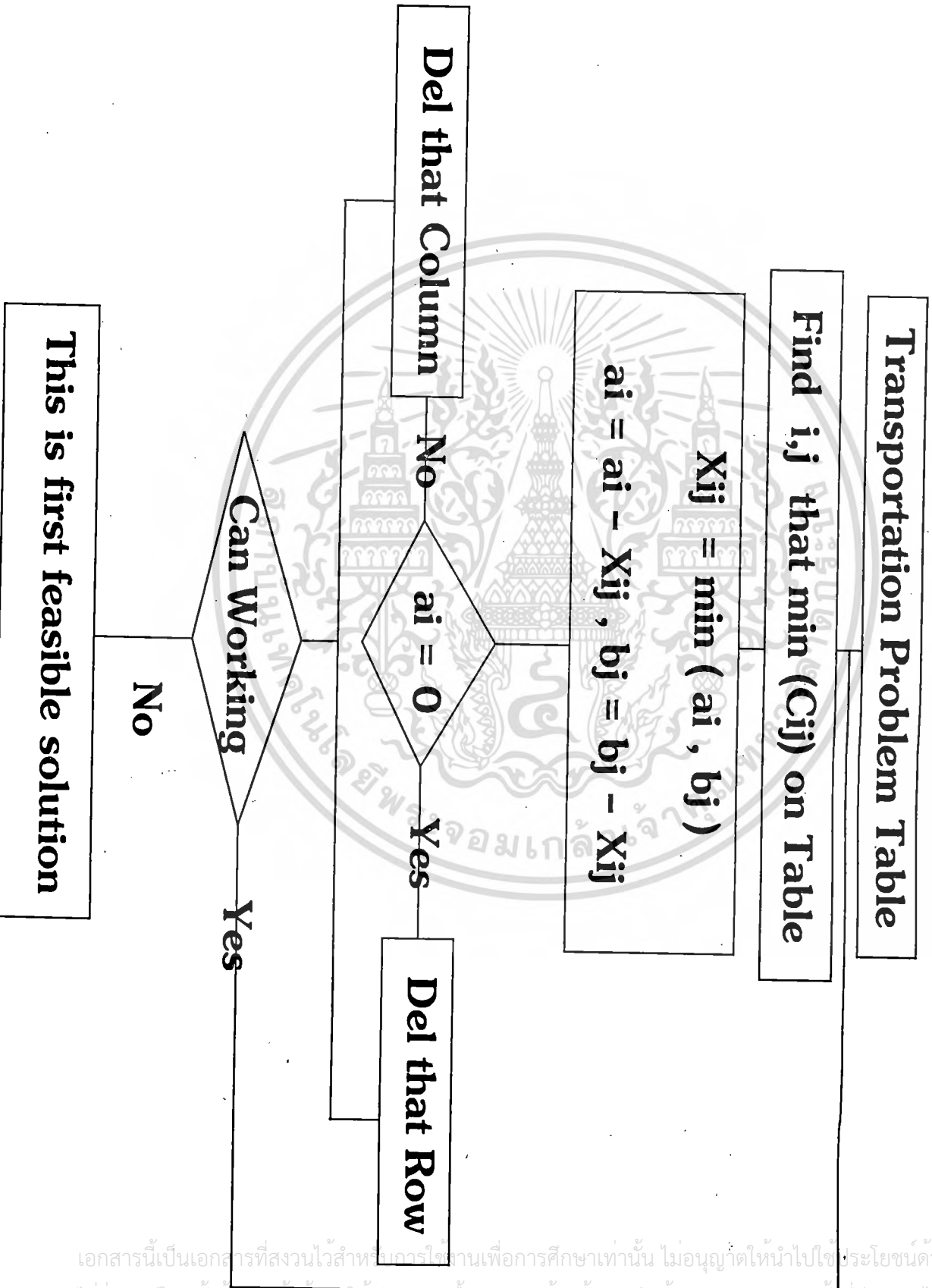
แผนภาพที่ 4.3 Flow Chart ของการแก้ปัญหาด้วยวิธี North - West Corner Rule

Flow Chart for North-West Corner Rule



แผนภาพที่ 4.4 Flow Chart ของการแก้ปัญหาด้วยวิธี Lowest First Costs Rule

Flow Chart for Lowest First Costs Rule



แผนภาพที่ 4.5 Flow Chart ของการแก้ปัญหาด้วยวิธี Vogel 's Approximation

Flow Chart for Vogel 's Approximation Method

Transportation Problem Table

Find $|\min(C_{ij}) - \min^*(C_{ij})|$ for each columns and rows

Find column or row that max $|\min(C_{ij}) - \min^*(C_{ij})|$

Find i, j that $\min(C_{ij})$

$X_{ij} = \min(a_i, b_j)$

$a_i = a_i - X_{ij}, b_j = b_j - X_{ij}$

Del that Column

No

$a_i = 0$

Yes

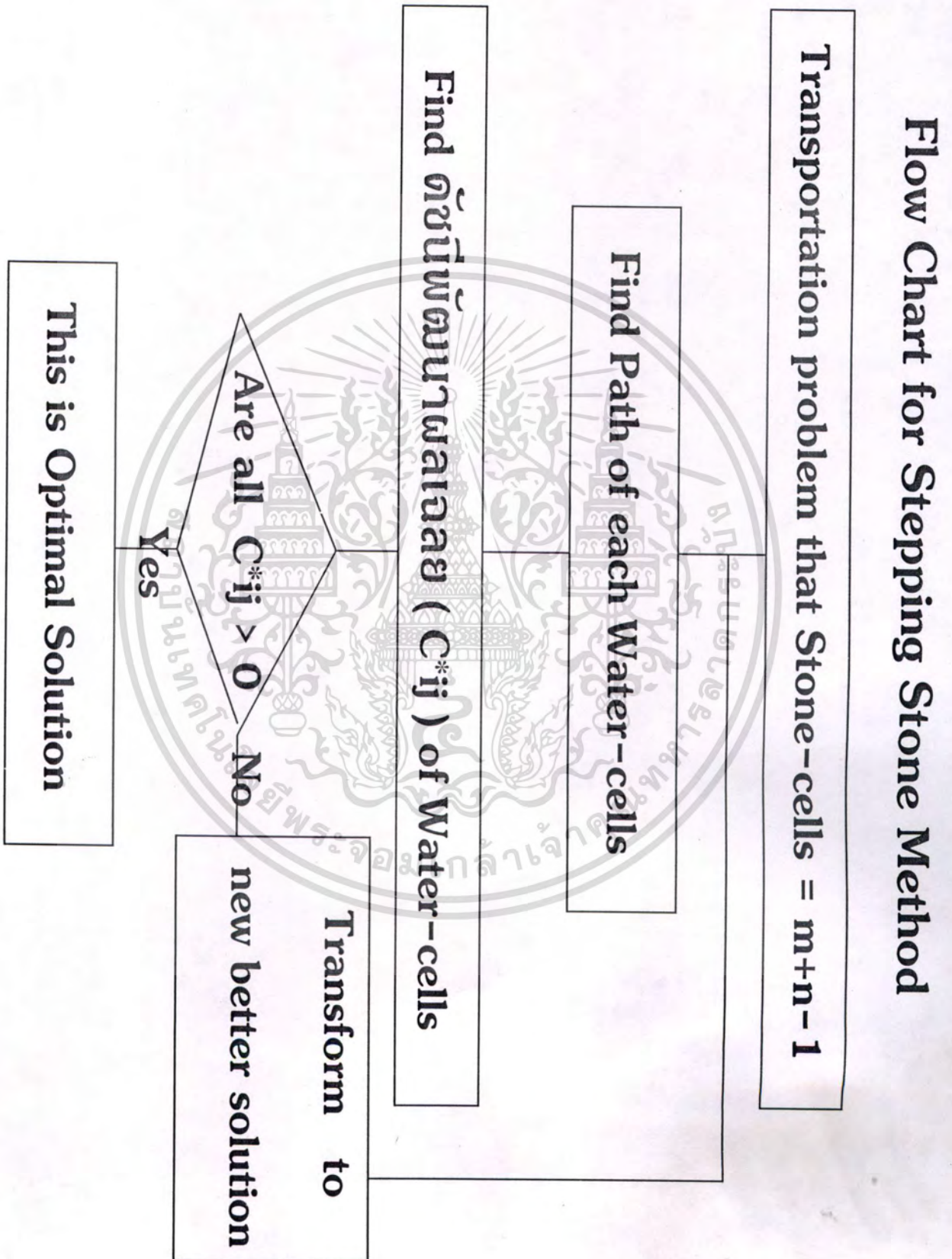
Del that Row

Can working

Yes

This is first feasible solution

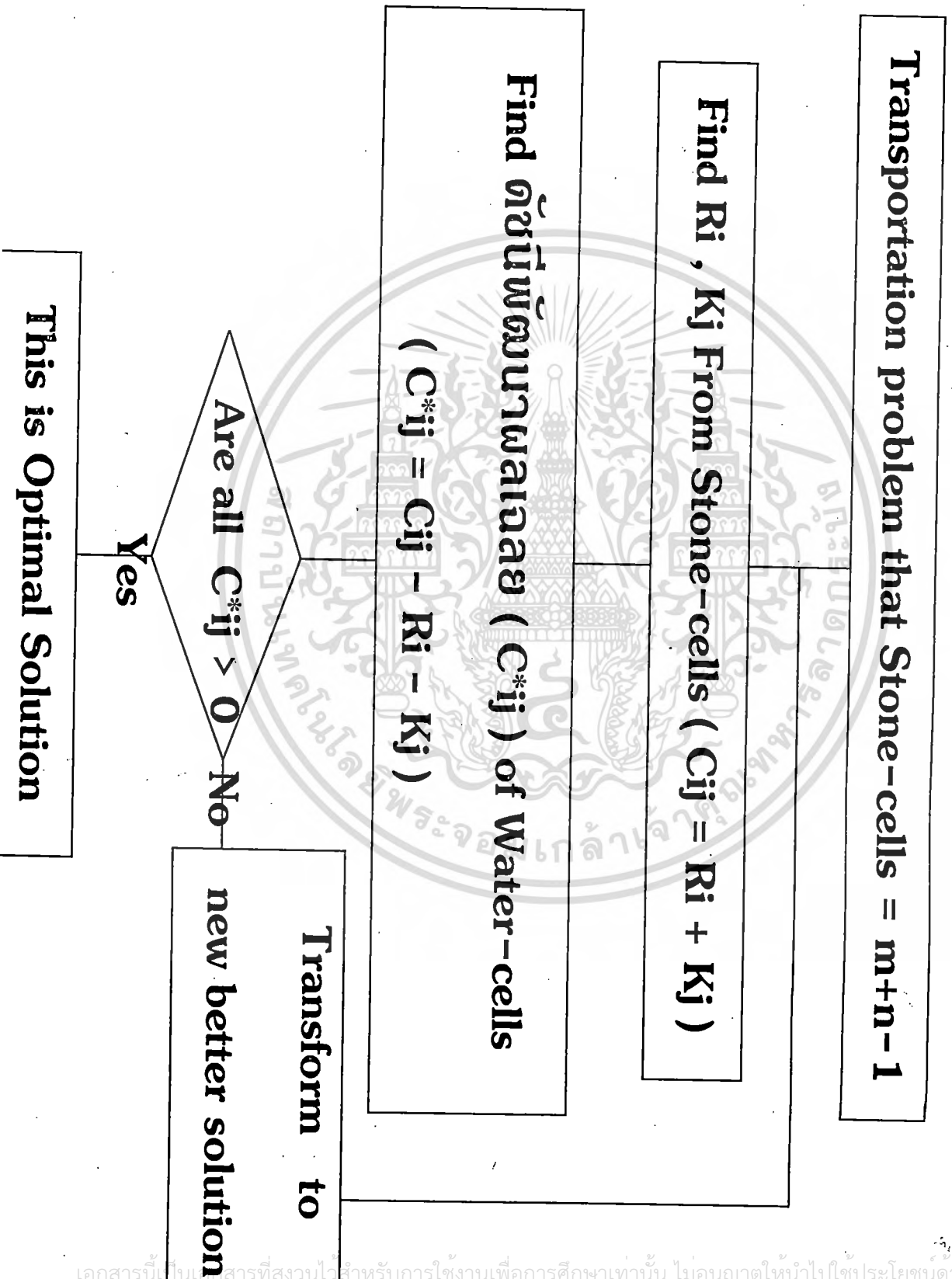
แผนภาพที่ 4.6 Flow Chart ของการแก้ปัญหาด้วยวิธี Stepping Stone Method



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพที่ 4.7 Flow Chart ของการแก้ปัญหาด้วยวิธี Modified Distribution Method

Flow Chart for MODI Method

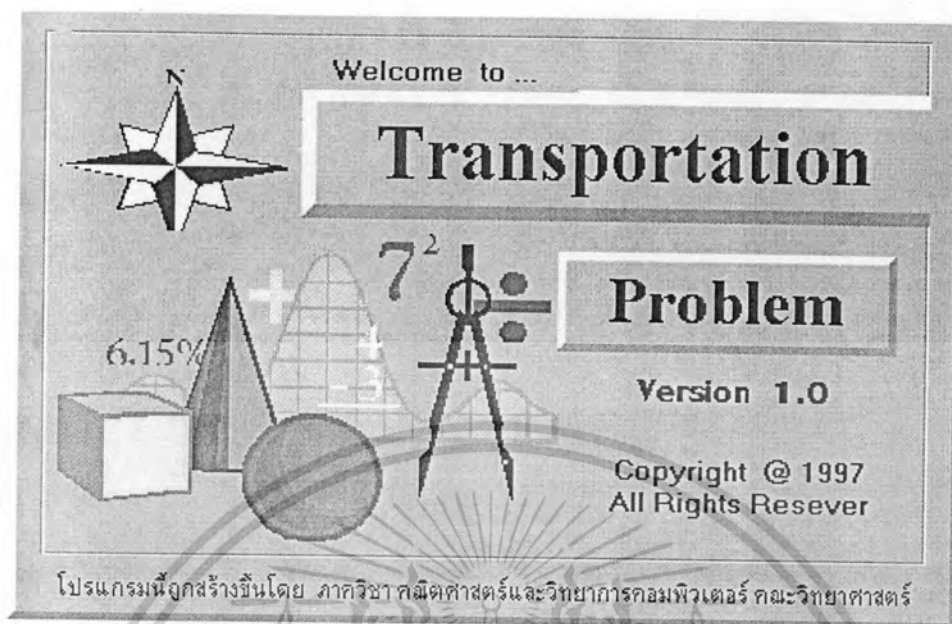


4.2 การออกแบบฟอร์มของโปรแกรม

ภายในโปรแกรมนี้ประกอบด้วยฟอร์มต่างๆ มากมาย ซึ่งมีหน้าตาแตกต่างกันออกไป โดยมีฟอร์มต่างๆดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

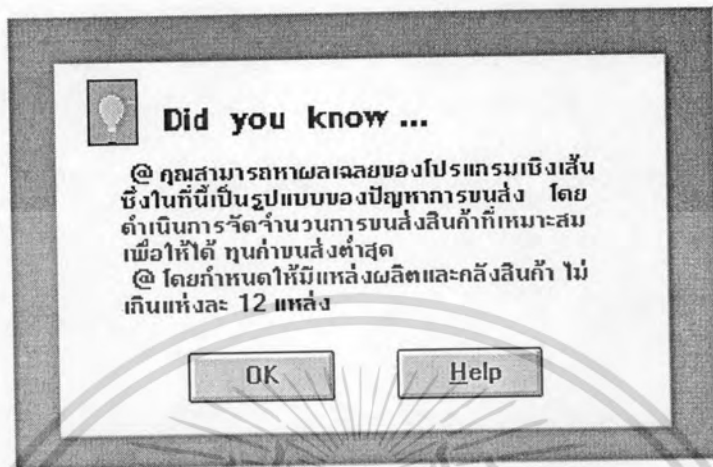


@

FORM : LOGO of Transportation Problem Program

โลโก (LOGO) ของระบบคอมพิวเตอร์สำหรับปัญหาการขนส่ง โดยจะปรากฏขึ้นเป็นฟอร์มหรือหน้าต่างแรกของโปรแกรมขณะเมื่อเริ่มต้นรันโปรแกรม หน้าต่างนี้จะหายไปก็ต่อเมื่อผู้ใช้กดคีย์ใดๆ (KeyPress) และหลังจากนั้นโปรแกรมก็จะขึ้นหน้าต่างหรือฟอร์มต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



@

FORM : Welcome to Transportation Problem Program

ฟอร์ม Welcome นี้ ระบบคอมพิวเตอร์สำหรับปัญหาการขนส่งมีขึ้น เพื่อเป็นการกล่าวคำแนะนำ และบอกถึงจุดประสงค์ (Purpose) รวมทั้งความสามารถภายในโปรแกรมนี้ ซึ่งฟอร์มนี้จะถูกรันต่อมาจาก ฟอร์ม โลก ของโปรแกรม

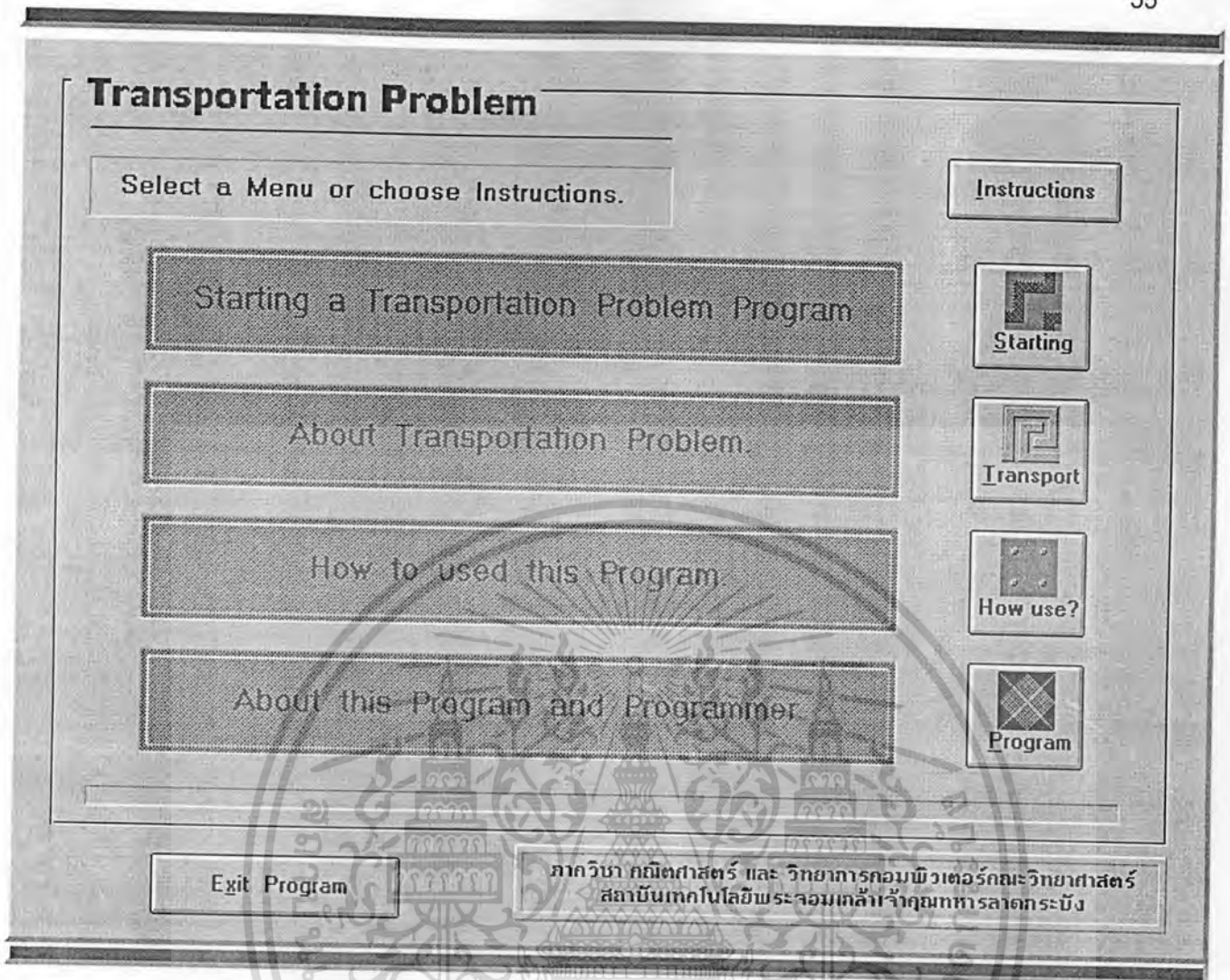
@ คอนโทรล : OK

เมื่อผู้ใช้กดคดคอลโทรลนี้ ก็เป็นเหมือนกับการรับทราบ ซึ่งจะส่งผลให้ โปรแกรมเข้าสู่ฟอร์มถัดไป

@ คอนโทรล : Help

โปรแกรมจะเข้าสู่หน้าต่าง Help เพื่อแนะนำการใช้โปรแกรมต่อไป (สำหรับผู้ที่ยังไม่เคยใช้โปรแกรมนี้มาก่อน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



@ _____

FORM : Menu of Transportation Problem Program

ฟอร์มนี้เป็นเมนูหลักของระบบคอมพิวเตอร์สำหรับปัญหาการขนส่ง

โดยผู้ใ้สามารถเรียกได้จาก คอนโทรล Menu ในเมนู Option

@ คอนโทรล : Instruction

โปรแกรมจะนำเข้าสู่คำแนะนำในการใช้คอนโทรลต่างๆ

@ คอนโทรล : Starting

โปรแกรมจะนำเข้าสู่การเริ่มต้นคำนวณหาผลเฉลย

@ คอนโทรล : Transport

โปรแกรมจะนำเข้าสู่ข้อมูลความรู้เกี่ยวกับ รูปแบบปัญหาการขนส่ง

@ คอนโทรล : How use ?

โปรแกรมจะนำเข้าสู่คำแนะนำในการใช้โปรแกรมปัญหาการขนส่งนี้

@ คอนโทรล : Program

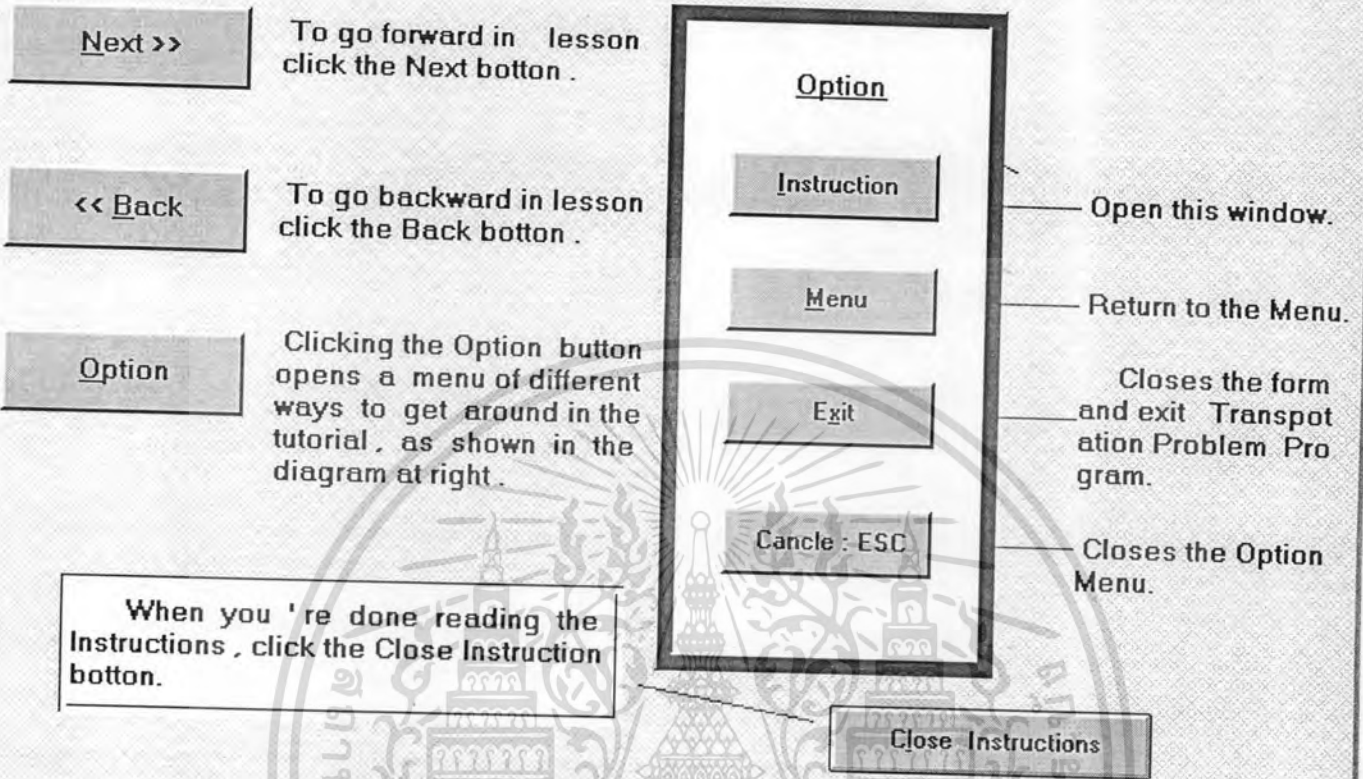
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ และห้ามนำไปใช้ซ้ำในทุกครั้งที่มีการนำไปใช้

@ คอนโทรล : Exit Program

ออกจากโปรแกรม

How to Use the Transportation Program < Tutorial > .



@

FORM : Instruction of Controls in Program

ฟอร์มนี้เป็นฟอร์มที่ตามมาจากการที่ผู้ใช้เลือก คอนโทรล Instruction จากเมนูหลัก หรือจาก Option เมนู จุดประสงค์ของหน้าต่างนี้เพื่อแนะนำในการใช้คอลโทรลต่างๆที่สำคัญภายในโปรแกรม

@ คอนโทรล : Close Instruction

โปรแกรมจะกลับเข้าสู่หน้าต่างเดิมที่มีการเรียกใช้หน้าต่าง Instruction นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกี่ยวกับ โปรแกรม และ ผู้เขียนโปรแกรม

Computer System
for
Transportation
Problem



@ ประโยชน์ที่คาดว่าจะได้รับจากโปรแกรมการแก้ไขปัญหาการโปรแกรมเชิงเส้น ใน รูปแบบของปัญหาการขนส่งนี้ คือต้องการหาผลเฉลยของปัญหาการขนส่งได้อย่างรวดเร็วและถูกต้อง ทั้งเป็นการพัฒนาการหาผลเฉลยการขนส่งให้มีประสิทธิภาพ

ผู้เขียนโปรแกรม

- @ นาย วัชรินทร์ เพ็ชรฤา
- @ ภาควิชา วิศวกรรมศาสตร์ วิทยาลัยการอาชีวศึกษา
- @ คณะ วิศวกรรมศาสตร์
- @ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

OK

Option

ภาษาที่ใช้ในการเขียนโปรแกรม

- @ ภาษ Visual Basic for Windows 3.0

@ _____

FORM : About Program and Programmer

ฟอร์มนี้เป็นฟอร์มที่ตามมา หลังจากเหตุการณ์ที่ผู้ใช้เลือกคอลลโทรล About Program จาก ฟอร์มเมนูหลักของโปรแกรม โดยที่ฟอร์มนี้จะ แสดงให้ทราบถึงข้อมูลต่างๆเกี่ยวกับการเขียนโปรแกรมทั้งหมด ทั้ง เกี่ยวกับผู้เขียนโปรแกรมและภาษาที่ใช้ในการเขียนโปรแกรม

@ คอนโทรล : Option
โปรแกรมจะเข้าสู่หน้าต่าง เมนู Option

@ คอนโทรล : OK
โปรแกรมจะกลับเข้าสู่เมนูหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Fill the number of Source (แหล่งผลิต)
and Distination (ปลายทาง)**


Instruction

ระบบคอมพิวเตอร์สำหรับปัญหาการขนส่งนี้ จัดทำ
ขึ้น โดยมีขอบเขตของการแก้ปัญหาที่ จำนวนแหล่งผลิต
และจำนวนคลังสินค้า ไม่เกินแห่งละ 12 แห่ง

Fill number of Row,Col.

* จำนวน แหล่งผลิต (Row) :

* จำนวนคลังสินค้า (Column) :


**Computer System
for
Transportation
Problem**

@ _____

FORM : Fill number of Source and Destination

ฟอร์มนี้เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือกคอลโทรล
Continue >> จากฟอร์มรูปแบบตารางปัญหาการขนส่งมาตรฐาน
หน้าต่านี้ผู้ใช้ใส่จำนวนของแหล่งสินค้าต้นทาง (Source) และ
แหล่งปลายทาง (Destination) ตามที่ต้องการในช่องว่าง (โดยไม่เกิน
12 แห่ง)

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option

@ คอนโทรล : << Back

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โปรแกรมจะนำกลับสู่หน้าต่างเดิม (ตารางขนส่งมาตรฐาน)
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transportation Problem

(ปัญหาการขนส่ง)

ตัวอย่างของรูปแบบปัญหาการขนส่ง

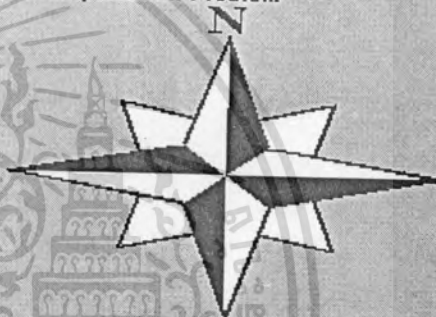
Option

Continue >>

@ Transportation Problem

ปัญหาการขนส่ง (Transportation Problem) เป็นรูปแบบหนึ่งของการโปรแกรมเชิงเส้น เป็นปัญหาที่เกิดจากการขนส่งวัตถุดิบจากแหล่งผลิตหลาย ๆ แหล่ง ไปยังแหล่งเก็บสินค้า หรือคลังสินค้าต่างๆ โดยที่ต้องการให้ต้นทุนค่าขนส่งต่ำสุด

@ Transportation Problem



@

FORM : Standard of Transportation Problem Program

ฟอร์มนี้เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือก คอนโทรล Starting จากเมนูหลัก จุดประสงค์ของหน้าต่างนี้เพื่อ นำผู้ใช้เข้าสู่การหาผลเฉลยของปัญหาจากรูปแบบของตารางปัญหาการขนส่งมาตรฐาน และคำแนะนำเกี่ยวกับรูปแบบปัญหาการขนส่ง

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option

@ คอนโทรล : Continue >>

โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการหาผลเฉลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาผลเฉลยของรูปแบบปัญหาการขนส่ง

@ การแก้ไขปัญหาการโปรแกรมเชิงเส้น ในรูปแบบของปัญหาการขนส่ง ซึ่งต้องการผลที่เป็นเป็นไปได้อย่างดีที่สุด และการได้ผลกำไรสูงสุดนี้สามารถแบ่งการหาผลเฉลยได้เป็น 2 ขั้นตอน คือ

Computer System
for
Transportation
Problem



Continue >>

<< Back

Option

1. การหาผลเฉลยเบื้องต้น

@

วิธีสามกษัตริย์ของบูกิดเหนือ-ตะวันตก (North - West)

@

Lowest First Costs Rule

@

Vogel 's Approximation Method

2. การหาผลเฉลยตามเป้าหมาย

@

วิธีสี่เหลี่ยม (Stepping Stone)

@

วิธีไฮโดร (Modified Distribution)

@

FORM : รูปแบบการแก้ปัญหาการขนส่ง (Transportation Solution)

ฟอร์มนี้เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือกคอนโทรล OK จากฟอร์ม Fill Transportation Value ในฟอร์มนี้ให้ผู้ใช้ได้ทราบถึงการหาคำตอบ หรือ วิธีการหาผลเฉลยทั้งหมดที่มีในโปรแกรม

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option

@ คอนโทรล : << Back

โปรแกรมจะนำกลับสู่หน้าต่างเดิม (Fill Transportation Value)

@ คอนโทรล : Continue >>

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกรูปแบบการหาผลเฉลยเบื้องต้น

วิธีตามกฎของมูมทิสเหนือ-ตก

Lowest First Costs Rule

วิธีการประมาณค่าของโบริล

Cancle Option

@

FORM : รูปแบบการหาผลเฉลยเบื้องต้นของปัญหาการขนส่ง

ฟอร์มนี้ เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือกคอนโทรล Control >> จากฟอร์ม รูปแบบการแก้ปัญหาการขนส่ง ในฟอร์มนี้ให้ผู้ใช้ได้เลือกรูปแบบวิธีการ การหาผลเฉลยเบื้องต้นในการแก้ปัญหา (จากทั้งหมด 3 วิธี)

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : Cancle

ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม (รูปแบบการแก้ปัญหาการขนส่ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vogel 's Approximation Method

@ วิธีการประมาณค่าของ วิเทล

Label1

Label2



Computer System
for
Transportation
Problem

Continue >>

Cancel

Option

@

FORM : Vogel ' Approximation Method

ฟอร์มนี้เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือก วิธีการหาผล
เฉลยด้วยวิธี Vogel ' Approximation Method จากฟอร์ม รูปแบบ
การหาผลเฉลยเบื้องต้น ในการแก้ปัญหาขนส่ง ในฟอร์มนี้จะให้ผล
เฉลยเบื้องต้นออกมา และรวมค่าขนส่งทั้งหมดที่ต้องการใช้ออกมาให้

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง Option เมนู

@ คอนโทรล : Cance

ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม (รูปแบบการหาผลห
เฉลยเบื้องต้น)

@ คอนโทรล : Continue >>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการหาผลเฉลย
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกรูปแบบการหาผลเฉลี่ยตามเป้าหมาย

Stepping Stone Method

Modified Distribution : MODI

Cancle Option

@

FORM : รูปแบบการหาผลเฉลี่ยตามเป้าหมายของปัญหาการขนส่ง

ฟอร์มนี้ เป็นฟอร์มที่ตามมา เมื่อเกิดเหตุการณ์ที่ผู้ใช้เลือกคอนโทรล Continue >> จากฟอร์ม การหาผลเฉลี่ยเบื้องต้น ทั้ง 3 รูปแบบ ใน ฟอร์มนี้ให้ผู้ใช้ได้เลือกการแก้ปัญหา หรือรูปแบบวิธีการหาผลตามเป้าหมายในการแก้ปัญหา (จากทั้งหมด 2 วิธี)

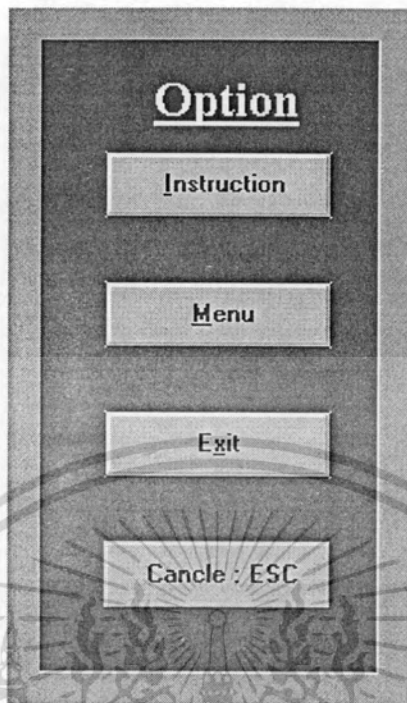
@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : Cancle

โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการหาผลเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



@

FORM : เมนู Option

ฟอร์มนี้เป็นฟอร์มที่ปรากฏขึ้น เมื่อผู้ใช้เลือกคอนโทรล Option จาก ฟอร์มใดๆของโปรแกรม โดยที่ฟอร์มนี้จะให้ผู้ใช้เลือกคอนโทรลในการ ดำเนินงานต่อไป ซึ่งมีทั้งหมด 4 คอนโทรล

@ คอนโทรล : Instruction

โปรแกรมจะนำเข้าสู่หน้าต่าง Instruction

@ คอนโทรล : Menu

โปรแกรมจะกลับสู่เมนูหลักของโปรแกรม

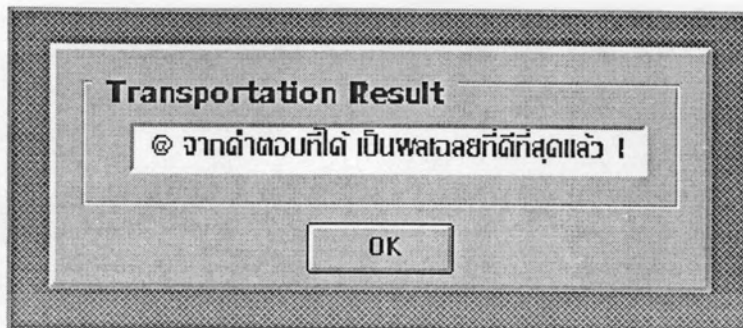
@ คอนโทรล : Exit

ออกจากโปรแกรม

@ คอนโทรล : Cancle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องขออนุญาตเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



@

FORM : Best Transportation Result

ฟอร์มนี้ จะปรากฏขึ้นเมื่อฟอร์มการหาผลเฉลยที่ดีที่สุด ทั้ง 2 ฟอร์ม คือ ฟอร์ม Stepping Stone Method Distribution Method ได้ผลเฉลยออกมา โดยที่ผลเฉลยที่ได้เป็นผลเฉลยที่ดีที่สุดแล้ว (ต้นทุนค่าขนส่งที่ได้มีค่าน้อยที่สุด)

@ คอนโทรล : OK

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนูหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



@

FORM : Please Check

ฟอร์มนี้ จะปรากฏขึ้นเมื่อฟอร์ม Fill number of Source Destination และ ฟอร์ม Fill Transportation Value มีการป้อนค่าผิดพลาด คือ ค่าที่ป้อนมีค่าสูงกว่าค่าที่กำหนดไว้ หรือ ค่าที่ป้อนไม่เป็นตัวเลข หรือ ป้อนค่าไม่ตรงกับช่องว่างตารางที่ให้ใส่ค่า รอให้ผู้ใช้กด Key ใดๆ เพื่อเข้าไปแก้ไขค่าในช่องตารางให้ถูกต้อง

@ คอนโทรล : OK

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนูหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัฒนาผลเฉลย

ดัชนีพัฒนาค่าตอบ : Label1 Label2

Continue >>

@

FORM : พัฒนาผลเฉลย

ฟอร์มนี้ จะปรากฏขึ้นเมื่อฟอร์ม การหาผลเฉลยที่ดีที่สุด ทั้ง 2 ฟอร์ม คือฟอร์ม Stepping Stone Method และ Modified Distribution Method กำลังคำนวณหาผลเฉลย โดยที่ค่าดัชนีพัฒนาผลเฉลยที่ได้มา จากการหาค่าดัชนีพัฒนาผลเฉลยของแต่ละวิธีซึ่งแตกต่างกันออกไป ค่าดัชนีพัฒนาผลเฉลยที่ได้จะนำไปพัฒนาผลเฉลยอีกขั้นตอนหนึ่ง (เนื่องจากผลเฉลยที่ได้ยังไม่เป็นผลเฉลยที่ดีที่สุด)

@ คอนโทรล : Continue >>

โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการหาผลเฉลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Choose Topic that you want to know

@ About Transportation

@ Instruction of Program

Cancle Option

@

FORM : Choose Topic that you want to know

ฟอร์มนี้ จะปรากฏขึ้นเมื่อผู้ใช้เลือก About transportation Problem จาก เมนู Transportation Problem Menu จุดประสงค์ของฟอร์มนี้ เพื่อให้ผู้ใช้ทราบเกี่ยวกับการใช้โปรแกรมและแนะนำเมนูต่างๆที่สำคัญ

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : Cancle

ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม (เมนูของโปรแกรมการหาผลเฉลยของปัญหาการขนส่ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

How to used this Program !

@ Transportation Problem Program

๑ จากหน้าต่าง บนจอหลัก
เลือก บนจอตามต้องการ (ซึ่ง
มีทั้งหมด ๔ บนจอ) เพื่อให้เข้า
สู่หน้าต่างโปรแกรมต่อไป

หรือเลือกคำสั่ง Instruc
tion บนจอคำแนะนำสำหรับ
โปรแกรม

หรือออกจากโปรแกรม
โดยคำสั่ง Exit Program

Continue >>

Cancel

Option

@

FORM : How to use this Program No.1

ฟอร์มนี้ จะปรากฏขึ้นเมื่อผู้ใช้เลือก About transportation จาก ฟอร์ม
Choose Topic that you want to know จุดประสงค์ของฟอร์มนี้ เพื่อ
แนะนำให้ผู้ใช้งานทราบเกี่ยวกับวิธีการใช้ Transportation Problem
Program Menu โดยจะมีคำแนะนำให้เข้าใจอย่างละเอียด

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : Cancele

ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม (Choose Topic that
you want to know)

@ คอนโทรล : Continue >>

โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการแนะนำเกี่ยวกับการใช้ภายใน

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

How to used this Program !

@ Transportation Problem Program

๑ จากหน้าต่าง หมายเหตุ มือเลือกคำสั่ง
Instruction จะปรากฏหน้าต่างแนะนำวิธีมา

๑ มือเลือกคำสั่งเมนู About
Program and Programmer

The screenshot shows two panels within a window titled 'How to use this Transportation'.
 The left panel, titled 'How to use this Transportation', contains buttons for 'Next', 'Back', 'Option', and 'Close'.
 The right panel, titled 'เกี่ยวกับโปรแกรมและผู้เขียน', contains a logo, a 'ผู้เขียนโปรแกรม' (Programmer) field with a grid of input boxes, an 'OK' button, an 'Option' button, and a 'ภาษาที่ใช้' (Language used) field with a dropdown menu.
 Below these panels are three buttons: 'Continue >>', '<< Back', and 'Option'.

@

FORM : How to use this Program No.2

ฟอร์มนี้ จะปรากฏขึ้นเมื่อผู้ใช้เลือกคอนโทรล Continue >> จาก
 ฟอร์ม How to use this Program No.1 จุดประสงค์ของฟอร์มนี้ เพื่อ
 แนะนำให้ผู้ใช้ทราบเกี่ยวกับวิธีการใช้ Option โดยจะมีคำแนะนำให้
 เข้าใจอย่างละเอียดเกี่ยวกับการใช้คอนโทรลต่างๆภายใน Option

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : << Back

ยกเลิก และ โปรแกรมจะนำกลับสู่หน้าต่างเดิม (Hoe to use this
 Program No.1)

@ คอนโทรล : Continue >>

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการแนะนำเกี่ยวกับการใช้ภายในโปรแกรม
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 โปรแกรม

About Transportation Problem

@ ปัญหาการขนส่ง

ปัญหาการขนส่ง (Transportation Problem) เป็นรูปแบบหนึ่งของ "ปัญหาการโปรแกรมเชิงเส้น" โดยที่จุดมุ่งหมายของปัญหา เพื่อจัดรายการให้มีค่าใช้จ่ายในการขนส่งน้อยที่สุด ปัญหานี้ส่วนใหญ่จะประกอบด้วยตัวแปรตัดสินใจและเงื่อนไขมากมาย

ปัญหาการขนส่ง เกิดจากการกระจายสินค้าจากหลาย ๆ แหล่งต้นทางใด ๆ (Source) ไปสู่แหล่งปลายทางใด ๆ (Destination) โดย วัตถุประสงค์ให้มีค่าใช้จ่ายรวมในการขนส่ง (Transport Costs) น้อยที่สุด

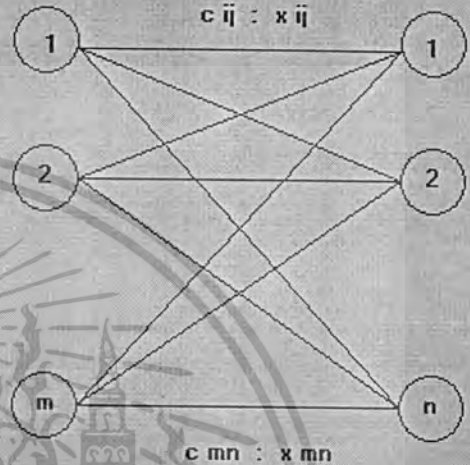
Continue >>

Option

รูปแบบทั่วไปของปัญหาการขนส่งมาตรฐาน

แหล่งผลิต a_i

คลังสินค้า b_j



แสดงแหล่งผลิตและคลังสินค้า

@

FORM : How to use this Program No.3

ฟอร์มนี้ จะปรากฏขึ้นเมื่อผู้ใช้เลือกคอนโทรล Continue >> จากฟอร์ม How to use this Program No.2 จุดประสงค์ของฟอร์มนี้ เพื่อแนะนำให้ผู้ใช้งานทราบเกี่ยวกับปัญหาค่าขนส่งและวิธีการแก้ปัญหาค่าขนส่ง โดยจะมีการแสดงภาพเกี่ยวกับปัญหาการขนส่งให้เข้าใจอย่างละเอียด

@ คอนโทรล : Option

โปรแกรมจะกลับเข้าสู่หน้าต่าง เมนู Option เมนู

@ คอนโทรล : Continue >>

โปรแกรมจะเข้าสู่หน้าต่างถัดไปในการแนะนำเกี่ยวกับการใช้ภายใน

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกี่ยวกับภาษาที่ใช้ในการเขียนโปรแกรม

ภาษา Visual basic

ปัจจุบันวินโดวส์นับได้ว่าเป็นระบบปฏิบัติการที่ได้รับความนิยมเป็นอย่างสูง และเครื่องไมโครคอมพิวเตอร์โดยทั่วไปกว่า 70 เปอร์เซ็นต์ ล้วนแต่ใช้วินโดวส์เป็นระบบปฏิบัติการ ทั้งนี้เนื่องจากตัวมันเองมีข้อดีกว่าระบบปฏิบัติการดอสอยู่หลายประการ เช่น ทำงานในโหมดกราฟฟิก ง่ายต่อการเรียนรู้ และใช้งาน สามารถทำงานหลายๆ โปรแกรมได้พร้อมๆ กัน สามารถโอนถ่ายข้อมูลระหว่างแอปพลิเคชันได้ง่าย สนับสนุนระหว่างหน่วยความจำเสมือน (virtual memory) และอื่นๆอีกหลายประการ จึงทำให้วินโดวส์เป็นระบบปฏิบัติการที่มีการใช้ความสามารถของเครื่องระดับไมโครคอมพิวเตอร์ได้ค่อนข้างเต็มที่

ตัวแปรภาษาต่างๆ จึงได้ถูกพัฒนาเพื่อให้สามารถนำมาเขียนโปรแกรมต่างๆ บนวินโดวส์ได้เช่นเดียวกับดอส ซึ่งหนึ่งในนี้ก็คือภาษาเบสิก ซึ่งคงทราบดีแล้วว่าภาษาเบสิกนั้นตัวมันเองเป็นภาษาขั้นสูงที่ได้รับการพัฒนามาตั้งแต่สมัยที่เครื่องไมโครคอมพิวเตอร์เป็นเครื่องในระดับ 8 บิต อย่างเช่นภาษาเบสิกที่พัฒนาโดยบริษัทไมโครซอฟต์ ก็เริ่มตั้งแต่ BASIC , BASICA , QuickBASIC , PDS BASIC , QBASIC , Visual BASIC for DOS และโดยเฉพาะ Visual BASIC for Windows ซึ่งเป็นตัวแปรภาษาที่ถูกออกแบบมาสำหรับพัฒนาแอปพลิเคชันสำหรับวินโดวส์

Visual BASIC for Windows ประกาศตัวครั้งแรกด้วยเวอร์ชัน 1.0 ต่อมาทางไมโครซอฟต์ก็ได้แก้ไขข้อผิดพลาดและเพิ่มเติมความสามารถบางประการเข้าไปเพื่อให้การพัฒนาแอปพลิเคชันทำได้รวดเร็วยิ่งขึ้น และได้ขยับขึ้นเป็นเวอร์ชัน 2.0 สำหรับแอปพลิเคชันที่ทำงานแบบ 16 บิต และหลังจากได้ออกระบบปฏิบัติการวินโดวส์เวอร์ชัน 3.11 ออกมา Visual BASIC for Windows ก็ได้รับการพัฒนาขึ้นเป็นเวอร์ชัน 3.0 ซึ่งเป็นเวอร์ชันที่ได้รับการปรับปรุงจากเดิมมากทีเดียว เช่น มีเอ็นจินสำหรับเข้าถึงฐานข้อมูลจากไฟล์ของโปรแกรมประมวลผลฐานข้อมูลที่มีชื่อเสียงมากมาย เช่น FoxPro , Access , dBASE , Btrieve , Paradox เป็นต้น รวมเอาความสามารถด้านการออกแบบรายงานโดยใช้ Crystal Reports for Visual BASIC นอกจากนี้ยังมี Wizard ที่ช่วยออกแบบโปรแกรม SETUP ได้ง่ายมากขึ้นทีเดียว ตั้งแต่เวอร์ชัน 1.0 ถึง 3.0

ในปัจจุบัน Visual BASIC for Windows ไม่ได้สนับสนุนเฉพาะการพัฒนาแอปพลิเคชันระดับ 16 บิตเท่านั้น ในราวๆ กลางปี พ.ศ. 2538 ทางไมโครซอฟต์ก็ออก Visual BASIC for Windows เวอร์ชัน 4.0 มาให้ใช้กัน ซึ่งจุดเด่นของเวอร์ชันนี้ คือ มีส่วนที่สนับสนุนการพัฒนาแอปพลิเคชันทั้งแบบ 16 บิต และ แบบ 32 บิต ซึ่งโปรแกรมเมอร์ภาษาเบสิกก็จะสามารถพัฒนาแอปพลิเคชันระดับ 32 บิต ที่ทำงานบนวินโดวส์ NT ได้โดยจะไม่ผูกขาดโดย Visual C++ อีกต่อไป

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนจากการพัฒนาแอปพลิเคชันโดยใช้ภาษาเบสิกสำหรับดอส มาพัฒนาแอปพลิเคชันบนวินโดวส์โดยใช้ Visual BASIC for Windows ก็ต้องมีการเปลี่ยนแปลงแนวคิดในการออกแบบโปรแกรมและวิธีการออกแบบโค้ดมากพอสมควร เนื่องจากโปรแกรมภาษาเบสิกที่ทำงานบนดอส จะมีการทำงานแบบต่อเนื่อง โดยเริ่มต้นจากโค้ดบรรทัดแรก และ สิ้นสุดการทำงานเมื่อพบคำสั่ง END และภายในส่วนของโมดูลหลักหรือภายในโพรซีเจอร์ใดก็จะกระโดดไปยังรูทีนใดๆ เมื่อพบคำสั่ง CALL SubName หรือ CALL FunctionName แต่สำหรับโปรแกรมที่เขียนโดย Visual BASIC for Windows ที่มีการทำงานบนวินโดวส์โดยได้รับการออกแบบให้มีการทำงานแบบ event - driven ดังนั้นการทำงานของโปรแกรมแต่ละขั้นตอนนอกจากขึ้นกับโค้ดที่เขียนแล้วยังขึ้นกับเหตุการณ์ (event) ต่างๆที่เขียนขึ้นภายในแอปพลิเคชันอีกด้วย เช่น Mouse_Click , Form_Load , Form_Unload เป็นต้น

ดังนั้นแนวคิดในการเขียนโปรแกรมประเภทกำหนดให้โค้ดที่เป็นส่วนใช้งานส่วนอื่นๆ (driver) เอาไว้ในส่วนหลักของโปรแกรมจึงต้องเปลี่ยนเป็นการเขียนโค้ดให้เหมาะสมกับเหตุการณ์แทน ซึ่งโปรแกรมเมอร์ต้องทำความเข้าใจถึงความหมายและวิธีการเกิดเหตุการณ์ต่างๆ ภายใต Visual BASIC for Windows ดีพอสมควร

ประวัติความเป็นมาของ Visual BASIC for Windows

ไมโครซอฟต์ได้พัฒนาตัวแปลภาษาเบสิกเพื่อให้เป็นหนึ่งในบรรดาเครื่องมือที่ใช้ในการสร้างแอปพลิเคชันสำหรับวินโดวส์ โดยได้ประกาศตัวครั้งแรกด้วยเวอร์ชัน 1.0 เมื่อกลางปี พ.ศ. 2534 เนื่องจากภาษาเบสิกเป็นภาษาที่โปรแกรมเมอร์ส่วนใหญ่มีความคุ้นเคยกันอยู่แล้ว ดังนั้นเมื่อมีตัวแปลภาษาเบสิกใหม่ๆ ออกมาจึงมักได้รับการต้อนรับเป็นอย่างดี เพราะ Visual BASIC for Windows ได้นำเอาความสามารถของวินโดวส์มาช่วยให้อวิธีการเขียนโปรแกรมเป็นไปอย่างง่ายมากขึ้น และยังได้นำแนวคิดในการเขียนโปรแกรมบนดอสจากที่เรียกว่า การเขียนโปรแกรมแบบโครงสร้าง (Structure Programming) มาเป็นการเขียนโปรแกรมแบบโครงสร้างสำหรับเหตุการณ์ (Structure Event Driven Programming) เพื่อช่วยลดขั้นตอนการเขียนโค้ดสำหรับควบคุมการทำงานของเหตุการณ์ต่างๆ ด้วยความสามารถในจุดนี้เองจึงทำให้ Visual BASIC for Windows เป็นที่รู้จักกันดีว่าเป็นตัวแปลภาษาที่ใช้วิธีการเขียนแบบวิซวล (Visual Programming Language)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาในช่วงปลายปี พ.ศ. 2535 เมื่อไมโครซอฟต์ได้พบว่า Visual BASIC for Windows เวอร์ชัน 1.0 ได้มีข้อจำกัดในการทำงานและมีข้อผิดพลาด (bug) หลายๆ ประการ และที่สำคัญเป็นตัวแปลภาษาเบสิกที่ทำงานได้ค่อนข้างช้าเอามากๆ ดังนั้นจึงได้ประกาศเวอร์ชัน 2.0 ออกมา และในเวอร์ชันนี้ได้รับการบอกกล่าวว่าเป็นเอนไวรอนเมนต์สำหรับการพัฒนาโปรแกรมบนวินโดวส์ที่ทันสมัยที่สุดในขณะนั้น ภายใต้เวอร์ชัน 2.0 นี้เองก็ได้รับการพัฒนาความสามารถและความรวดเร็วในการทำงานมากขึ้นกว่าเดิมอย่างมากทีเดียว ทั้งนี้เพราะเวอร์ชัน 2.0 ได้ถูกวางเป้าหมายไปถึงการถูกไว้วางใจในการนำมาใช้เป็นเครื่องมือสำหรับพัฒนาแอปพลิเคชันในเชิงธุรกิจเลยทีเดียว แต่หลังจากที่ได้ออกเวอร์ชัน 2.0 มาได้ไม่นาน ไมโครซอฟต์ก็ได้พบว่าสิ่งที่ขาดไปสำหรับเวอร์ชันนี้ เป็นความสามารถด้านงานฐานข้อมูล ซึ่งถือได้ว่าเป็นจุดอ่อนของภาษาเบสิกทุกตัวที่ที่ผ่านมา ดังนั้นต่อมาในปี พ.ศ. 2536 Visual BASIC for Windows เวอร์ชัน 3.0 ก็ได้ออกสู่ท้องตลาด ซึ่งความสามารถด้านตัวโปรแกรมเอง ไม่ว่าจะเป็นคำสั่งหรือความเร็วในการทำงานก็ไม่ได้เปลี่ยนแปลงไปมากนัก แต่จุดเด่นที่ทำให้เวอร์ชัน 3.0 เป็นที่น่าสนใจมากก็เห็นจะมีแต่เครื่องมือต่างๆ ที่ให้มาเพื่อลดภาระในการเขียนโปรแกรมหลายๆ ตัว เช่น OLE 2.0 , Crystal Report for Visual BASIC for Windows , Setup Wizard , Database Access Engine เป็นต้น ซึ่งทำให้ตัวแปลภาษา Visual BASIC for Windows ที่เคยเป็นตัวแปลภาษาที่มีจุดด้อยในงานฐานข้อมูล กลับมีความแข็งแกร่งขึ้นมาทันที โดยสามารถที่จะใช้ Database Access Engine ซึ่งเป็นเอ็นอินด้านฐานข้อมูลไมโครซอฟต์ได้นำมาจากแอปพลิเคชัน Access เวอร์ชัน 1.1 ใส่เข้าไปใน Visual BASIC for Windows เพื่อให้เป็นรูปแบบหลักสำหรับการสร้างหรือแก้ไขไฟล์ฐานข้อมูล และนอกจากนี้ยังสนับสนุนการเข้าถึงไฟล์ฐานข้อมูลที่มีชื่อเสียง เช่น FoxPro , Btrieve , dBASE , Microsoft Access , Paradox , Microsoft SQL Server , Oracle และ ODBC

และในประมาณกลางปี พ.ศ. 2538 ทางไมโครซอฟต์ก็ทำการเปิดตัว Visual BASIC for Windows เวอร์ชัน 4.0 ออกสู่ตลาด ซึ่งเวอร์ชันนี้นอกจากจะได้รับการปรับปรุงความสามารถของเอนไวรอนเมนต์เอง เช่น สามารถรันแอปพลิเคชันได้พร้อมๆ กันหลายๆ แอปพลิเคชัน , Jet 2.5 Database Engine , Object Browser สนับสนุนหมายเลขเวอร์ชันของไฟล์ เป็นต้น แล้ว ยังทำให้เกิดการแตกตัวของ Visual BASIC for Windows ตั้งแต่เวอร์ชัน 1.0 ถึงเวอร์ชัน 3.0 เป็นเวอร์ชันที่ทำงานบนวินโดวส์ 16 บิต แต่ภายใต้เวอร์ชัน 4.0 นอกจากจะมี Visual BASIC for Windows ระดับ 16 บิตแล้ว ยังมี Visual BASIC for Windows ระดับ 32 บิตอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนต่างๆของ Visual BASIC for Windows

โปรเจกต์ (Project)

แอปพลิเคชันของ Visual BASIC for Windows นั้นได้จากการสร้างไฟล์ .EXE จากโปรเจกต์หนึ่งๆ ซึ่งไฟล์โปรเจกต์นี้ก็เป็นไฟล์แอสกีที่จัดเก็บชื่อของไฟล์อื่นๆ ที่ใช้ในการรันแอปพลิเคชัน โดยไฟล์ที่ Visual BASIC for Windows กำหนดให้สามารถถูกนำมารวมกันภายในโปรเจกต์หนึ่งๆมีดังนี้

- **ไฟล์โมดูลฟอร์ม (Form module file)**

เป็นไฟล์ที่ใช้จัดเก็บรายละเอียดต่างๆ ของฟอร์มเอาไว้ ไม่ว่าจะเป็นคุณลักษณะต่างๆ เช่น ตำแหน่ง ขนาด รูปร่าง ข้อความของคอนโทรลในฟอร์ม การประกาศค่าคงที่ ตัวแปร โพรซีเจอร์ทั่วไป หรือ โพรซีเจอร์เหตุการณ์ในระดับฟอร์ม (form level) ไฟล์ชนิดนี้จะมีนามสกุล .FRM

- **ไฟล์โมดูลมาตรฐาน (Standard module file)**

เป็นไฟล์มาตรฐานของภาษาเบสิกต่างๆ ไป สำหรับจัดเก็บโพรซีเจอร์ทั่วไป การประกาศตัวแปร โครงสร้าง (TYPE...END TYPE) การประกาศฟังก์ชันวินโดวส์ API หรือฟังก์ชันอื่นๆ ในไฟล์ .DLL จัดเก็บค่าคงที่ประเภทครอบคลุม (global variable) ซึ่งโพรซีเจอร์ที่สร้างในไฟล์นี้โดยปกติจะถูกกำหนดให้เป็นโมดูลแบบ Public อยู่แล้ว โดยไฟล์ชนิดนี้กำหนดให้มีนามสกุล .BAS แต่สำหรับบางโปรเจกต์อาจจะประกอบด้วยไฟล์ฟอร์มเพียงไฟล์เดียวก็ได้ โดยไม่จำเป็นต้องมีไฟล์โมดูลมาตรฐาน

สำหรับไฟล์ภาษาเบสิกที่เขียนบนตัวแปรภาษาเบสิกบนดอส เช่น QBASIC , QuickBASIC , TurboBASIC หรือ Power Basic เมื่อจะนำมาใช้กับ Visual for Windows จะต้องถูกโหลดเข้ามาพร้อมกับโปรเจกต์ในรูปแบบของไฟล์โมดูลมาตรฐานเท่านั้น

- **ไฟล์ custom control (Custom control file)**

เป็นไฟล์ที่จัดเก็บคอนโทรลต่างๆ ที่ปรากฏในทูลบ็อกซ์เมื่อเราโหลดไฟล์เหล่านี้เข้ามารวมในโปรเจกต์ ซึ่งคอนโทรลเหล่านี้สามารถนำไปใช้ได้ในการแสดงผลหรือควบคุมการทำงานของแอปพลิเคชันได้ เมื่อทำการสร้างไฟล์ .EXE คอนโทรลเหล่านี้ก็จะไม่ถูกรวมเข้าไปในแอปพลิเคชันที่สร้างขึ้นมา สำหรับเวอร์ชัน 3.0 จะถูกกำหนดให้ไฟล์เหล่านี้มีนามสกุล .VBX ส่วนเวอร์ชัน 4.0 จะมีนามสกุล .OCX ซึ่งโดยส่วนใหญ่ไฟล์ custom control ของทั้งสองเวอร์ชันจะคอมแพตติเบิลกัน แต่สำหรับในเวอร์ชัน 4.0 บาง custom control อาจจะได้รับการพัฒนาจนไม่คอมแพตติเบิลกับเวอร์ชันที่ผ่านมาก็ได้ เช่น GRAPH.VBX เป็นต้น ดังนั้นถ้าหากจะทำการย้ายแอปพลิเคชันที่เคยเขียนบนไมโครซอฟท์วินโดวส์อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวอร์ชันก่อนหน้า ก็ขอให้เปลี่ยนบรรดาไฟล์ custom control เสียด้วยเพื่อป้องกันการสูญหายของข้อมูลในส่วนที่ไม่ได้รับการสนับสนุนจากเวอร์ชัน 4.0

- **ไฟล์โมดูลทรัพยากร (resource file)**

เป็นไฟล์ไบนารีที่รวบรวมทรัพยากรต่างๆ เช่น ข้อความ บิตแมพ หรือ อื่นๆ ที่สามารถถูกนำมาใช้โดยแอปพลิเคชันทั่วไป โดยเมื่อมีการเรียกมาใช้แล้วทรัพยากรเหล่านี้ก็ยังคงอยู่ในไฟล์ทรัพยากร ข้อดีของไฟล์ทรัพยากรก็คือ เมื่อเรามีการแก้ไขหรือปรับปรุงข้อมูลใดๆ ในไฟล์ทรัพยากร ข้อมูลส่วนที่ถูกแก้ไขและถูกนำไปใช้โดยแอปพลิเคชันก็จะเปลี่ยนแปลงไปด้วย แต่การเปลี่ยนแปลงนี้ไม่จำเป็นต้องมีการแก้ไขโค้ดใหม่แต่อย่างใด แต่สำหรับการใช้งานไฟล์ทรัพยากรนี้มีข้อจำกัดอยู่ประการหนึ่งคือ ในโปรเจกต์หนึ่งๆ สามารถรวมเอาไฟล์นี้เข้ามาได้เพียงไฟล์เดียวเท่านั้น โดยปกติไฟล์ทรัพยากรจะถูกกำหนดให้มีนามสกุล .RES

- **ไฟล์โมดูลคลาส (class module class)**

เป็นไฟล์ที่ใช้เก็บนิยามคุณสมบัติ (property) และวิธี (methode) ของคลาสเอาไว้ ซึ่งคลาสก็หมายถึงการนิยามออบเจกต์อย่างเป็นทางการ ซึ่ง Visual BASIC for Windows จะใช้คลาสทำหน้าที่เป็นแม่แบบสำหรับการสร้างออบเจกต์ในขณะที่กำลังทำงาน (run time) และคลาสจะนิยามคุณสมบัติและวิธีของออบเจกต์เพื่อให้สามารถนำมาใช้ควบคุมพฤติกรรมการทำงานของออบเจกต์นั้นๆ เป็นไฟล์ ชนิดใหม่ที่มีในเวอร์ชัน 4.0 สำหรับใช้สร้างแอปพลิเคชัน ชนิด ออบเจกต์ (Object Application) โดยมีคุณสมบัติเช่นเดียวกับออบเจกต์ทั่วไปของ Visual BASIC for Windows และสามารถเรียกใช้โดยแอปพลิเคชันต่างๆ ไปได้ของ Visual BASIC for Windows โดยถูกกำหนดให้มีนามสกุล .CLS

- **ไฟล์ไดนามิกลิงค์ไลบรารี (dynamic link library file)**

สำหรับส่วนนี้อาจจะไม่ใช้ไฟล์มาตรฐานของ Visual BASIC for Windows แต่ในบางกรณีที่มีการนำไฟล์ที่ได้รับการพัฒนาจากบริษัทอื่นๆ เพื่อสนับสนุน Visual BASIC for Windows ในรูปแบบของไฟล์ .DLL สามารถที่จะรวมเอาไฟล์นี้มายังโปรเจกต์ได้ ซึ่งโพธิ์เตอร์ต่างๆ ที่มีในไฟล์ .DLL นี้ก็สามารถเรียกใช้ได้เช่นเดียวกับโพธิ์เตอร์ต่างๆ ไปได้ของ Visual BASIC for Windows แต่ต้องประกาศโพธิ์เตอร์พวกนี้เสียก่อนด้วยประโยค Declare

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟอร์ม (Form)

ฟอร์มของ Visual BASIC for Windows นั้นได้ถูกนำมาใช้ในการบรรจุคอนโทรลต่างๆ เพื่อให้ทำการสื่อสารใช้งานหรือตอบโต้การทำงานกับผู้ใช้ เมื่อมีการรันแอปพลิเคชัน ฟอร์มที่ถูกกำหนดให้เป็น ฟอร์มเริ่มต้น (Start Up) ก็จะเป็นฟอร์มที่ถูกโหลดและแสดงผลในเบื้องต้น สำหรับการกำหนดฟอร์มเริ่มต้น (startup form) สามารถกำหนดได้จากไดอะล็อกบ็อกซ์ของคำสั่ง Project Option ... ของเมนู Tools นอกจากฟอร์มจะสามารถใช้จับเก็บคอนโทรลต่างๆ แล้ว ยังสามารถสร้างเมนูให้กับฟอร์มต่างให้กับฟอร์มต่างๆ ได้อีกด้วย โดยอาศัยหน้าต่างออกแบบ (Menu Design Window) สำหรับการแสดงผลหน้าต่างเมนูออกแบบนี้สามารถทำได้โดยใช้คำสั่ง Menu Editor ของเมนู Tools สำหรับเวอร์ชัน 4.0

หน้าต่างออกแบบเมนูจะสามารถแสดงผลได้ก็ต่อเมื่อไฟล์ในหน้าต่างโปรเจกต์ได้ถูกเลือกเป็น ฟอร์มเท่านั้น และต้องแสดงฟอร์มในขณะนั้นไม่ใช่หน้าต่างโค้ดของฟอร์ม โดยสมมุติว่าในขณะนี้ Visual BASIC for Windows กำลังแสดงผล Form1 และได้ทำการออกแบบเมนูไฟล์ ของ ฟอร์ม Form1 ซึ่งจะพบว่าภายใต้เมนูหนึ่งๆ เราสามารถที่จะสร้างเมนูย่อยได้หลายๆ ระดับ

โครงสร้างของไฟล์ฟอร์ม

ไฟล์ฟอร์มเป็นไฟล์ที่จัดเก็บคำอธิบายคุณสมบัติ และ โพรซีเจอร์เหตุการณ์ของฟอร์มและคอนโทรลที่ถูกจัดเก็บเอาไว้ภายในแต่ละฟอร์ม และนอกจากนี้ยังสามารถสร้างโพรซีเจอร์ทั่วไปให้กับ ฟอร์มได้อีกด้วย

version 3.0

Begin Form formname

property = value

...

Begin containercontrol containercontrolname

property = value

...

Begin control controlname

property = value

...

End

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End

```

Begin control controlname
    property = value
    ...
End
Begin Menu menuname
    Caption = " string "
    [ Checked = boolean ]
    [ Enabled = boolean ]
    [ Visible = boolean ]
    Begin Menu mnuFileClear
        Caption = " string "
        [ Checked = boolean ]
        [ Enabled = boolean ]
        [ Visible = boolean ]
    End
End
End

statement
.....
Sub Form_event ( argument )
...
End Sub
Sub Control_event ( argument )
...
End Sub
[ Static ] Sub procname ( argument )
.....
End Sub

```

วิธี (Methode)

เป็นโพรซีเจอร์ชนิดหนึ่งที่ถูกเขียนขึ้นมาเพื่อสนับสนุนการเรียกใช้ออปเจกต์ ซึ่งวิธีในโพรซีเจอร์ที่ไม่สามารถแก้ไขได้โดยโค้ด วิธีต่างๆ ของอปเจกต์สำหรับ Visual BASIC for Windows เช่น Cls , Refresh , Show เป็นต้น ในการเรียกใช้วิธีของอปเจกต์ใดๆ สามารถกระทำได้จากทุกๆ ตำแหน่งในโพรซีเจอร์ต่างๆ แม้ในโพรซีเจอร์เหตุการณ์ของอปเจกต์นั้นๆ เองก็ได้ แต่ในการใช้วิธีต่างๆ ต้องระวังเป็นพิเศษเพื่อป้องกันไม่ให้เกิดเหตุการณ์ที่เรียกว่า cascading เพราะจะทำให้เกิดข้อผิดพลาดได้ง่าย

โมดูล (Module)

สำหรับแอปพลิเคชันที่มีขนาดโตๆ หรือต้องการข้อมูลชนิดกำหนดโดยผู้ใช้ (user - define type) ที่สามารถถูกอ้างอิงได้จากทุกๆ ฟอรัมของแอปพลิเคชัน เราจำเป็นต้องมีการเพิ่มโมดูลเข้ามารวมในโปรเจกต์ ซึ่งโพรซีเจอร์ทั่วไปหรือตัวแปร (ที่ถูกประกาศในส่วนประกาศของโมดูล) ต่างๆ ที่มีในโมดูลมีข้อจำกัดที่แตกต่างจากโพรซีเจอร์ทั่วไปของฟอรัม เนื่องจากโพรซีเจอร์ทั่วไปหรือตัวแปรของฟอรัม สามารถเรียกใช้ได้เฉพาะฟอรัมนั้นๆ เท่านั้น แต่สำหรับโพรซีเจอร์ทั่วไปหรือตัวแปรของโมดูล เราสามารถที่จะกำหนดขอบเขตการมองเห็นได้ โดยการใช้ประโยค Private หรือ Public ดังนั้นโพรซีเจอร์ทั่วไปหรือตัวแปรของโมดูลจึงสามารถถูกเรียกใช้ได้จากทุกๆ ส่วนของโปรเจกต์ ดังนั้นการสร้างแอปพลิเคชันหลายๆ แอปพลิเคชัน ในบางครั้งเราจะสร้างโมดูลที่ประกอบด้วยโพรซีเจอร์ทั่วไปที่สามารถใช้งานร่วมกันได้จากหลายๆ แอปพลิเคชัน เพื่อลดเวลาการสร้างแอปพลิเคชันลงไป

สำหรับหน้าตาของโค้ดของฟอรัมและโมดูลจะประกอบด้วยส่วนต่างๆ ดังนี้

• ส่วนการประกาศ (declaration)

ส่วนการประกาศเป็นส่วนที่ใช้ในการสร้างหรือประกาศค่าคงที่ ตัวแปร หรือ โพรซีเจอร์ของไฟล์ .DLL ต่างๆ ในส่วนนี้ แต่ที่สำคัญเราไม่สามารถเขียนโค้ดอื่นๆ ที่เกี่ยวข้องกับการปฏิบัติการ เช่น Print , Cls , Do...Loop เป็นต้น ลงในส่วนนี้ เพราะจะทำให้เกิดข้อผิดพลาด

สำหรับสิ่งต่างๆ ที่ประกาศในส่วนนี้ของฟอรัมจะสามารถมองเห็นได้เฉพาะในฟอรัมนั้นๆ เท่านั้น ทั้งนี้เนื่องจากสิ่งเหล่านี้มีคุณสมบัติ Private โดยปริยาย และไม่สามารถแก้ไขได้ แต่ถ้าหากมีการประกาศในส่วนการประกาศของโมดูล สิ่งเหล่านี้ก็สามารถมองเห็นได้จากทุกๆ ส่วนของโปรเจกต์ เนื่องจาก Visual BASIC for Windows กำหนดให้มีคุณสมบัติ Public แต่ผู้อ่านสามารถกำหนดขอบเขตการมองเห็นเฉพาะในโมดูลหนึ่งๆ ก็ได้โดยนำประโยค Private นำหน้าโพรซีเจอร์นั้นๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โพรซีเจอร์เหตุการณ์ (event procedure)

เป็นโพรซีเจอร์ชนิด Sub ที่ถูกสร้างอัตโนมัติโดย Visual BASIC for Windows สำหรับแอปพลิเคชันต่างๆ ของโปรแกรม เพื่อให้ตอบสนองต่อการเกิดเหตุการณ์ของแอปพลิเคชันต่างๆ

- โพรซีเจอร์ทั่วไป (general procedure)

หมายถึงโพรซีเจอร์ที่ถูกสร้างขึ้นมาเองโดยผู้ใช้ ซึ่งโพรซีเจอร์ทั่วไปสามารถที่จะสร้างให้กับฟอร์มหรือ โมดูลก็ได้ แต่สำหรับที่สร้างให้กับฟอร์ม ต้องกำหนดชื่อให้แตกต่างจากชื่อของโพรซีเจอร์เหตุการณ์ ส่วนโพรซีเจอร์ทั้งหมดที่สร้างให้กับโมดูลจะต้องเป็นโพรซีเจอร์ทั่วไป และสามารถกำหนดให้เป็นโพรซีเจอร์ชนิด Sub หรือ Function ก็ได้ โดยที่โพรซีเจอร์ชนิด Sub หรือ Function ก็ได้ โดยที่โพรซีเจอร์ชนิด Sub จะไม่มีการส่งค่ากลับผ่านทางโพรซีเจอร์ แต่โพรซีเจอร์ชนิด Function สามารถที่จะส่งค่ากลับผ่านทางฟังก์ชันได้

รูปแบบการใช้

ประโยค Sub

```
[ Static ] [ Private ] Sub subname [( argument )]
```

```
[ statementblock ]
```

```
[ Exit Sub ]
```

```
[ statementblock ]
```

```
End Sub
```

สร้างโพรซีเจอร์ชนิด Sub โดยจะใช้ในการประกาศชื่อ อาร์กิวเมนต์ และกำหนดขอบเขตของส่วนสำหรับเขียนโค้ดของโพรซีเจอร์

ประโยค Function

```
[ Static ] [ Private ] Function functionname [( argument )] [ As type ]
```

```
[ statementblock ]
```

```
[ function = expresstion ]
```

```
[ Exit Function ]
```

```
[ statementblock ]
```

```
[ function = expresstion ]
```

```
End Function
```

สร้างโพรซีเจอร์ชนิด Function โดยจะใช้ในการประกาศชื่อ อาร์กิวเมนต์ และกำหนดขอบเขตของส่วนสำหรับเขียนโค้ดของโพรซีเจอร์

สำหรับรูปแบบของอาร์กิวเมนต์มีดังนี้

[ByVal] variable [()] [As type] [, ByVal] variable [()] [As type] . . .

By Val ประโยคสำหรับกำหนดให้ Visual BASIC for Windows ส่งตัวแปรให้กับโพรซีเจอร์ด้วยวิธีส่งค่า

variable ชื่อของอาร์กิวเมนต์ สำหรับอาร์กิวเมนต์ชนิดอะเรย์ ก็ให้ใส่เครื่องหมายเปิด-ปิด () ต่อท้ายชื่อของอาร์กิวเมนต์

As type ประโยคสำหรับชนิดของข้อมูล

โมดูลคลาส (Module Class)

โมดูลคลาสได้ถูกนำมานิยามคลาส คุณสมบัติ และวิธีของออปเจกต์แอปพลิเคชัน ซึ่งไฟล์หนึ่งๆ ของโมดูลคลาสสามารถใช้ในการนิยามคุณสมบัติหรือวิธีของคลาสได้เพียงคลาสเดียวเท่านั้น ในการเพิ่มโมดูลคลาสเข้ามายังโปรแกรมสามารถทำได้โดยการใช้คำสั่ง Class Module ของเมนู Insert หลังจากนั้นก็จะปรากฏหน้าต่างโค้ดของข้อมูลคลาสนั้นทันที สำหรับในกรณีที่ต้องการสร้างคลาสมากกว่าหนึ่งให้กับออปเจกต์แอปพลิเคชัน ก็สามารถทำได้โดยการเพิ่มโมดูลคลาสให้กับโปรเจกต์นั่นเอง

ออปเจกต์ (Object)

เมื่อทำการสร้างแอปพลิเคชันขึ้นมา นั่นก็หมายความว่าเรากำลังมีการใช้งานออปเจกต์ต่างๆ ของ Visual BASIC for Windows ซึ่งออปเจกต์เหล่านั้นอาจจะเป็น ฟอรัม คอนโทรล หรือ ออปเจกต์ของระบบก็ได้ และในการเขียนโค้ดเพื่อจัดการเกี่ยวกับคอนโทรล สามารถทำได้โดยการอ้างอิงถึงออปเจกต์โดยตรงหรือโดยการสร้างตัวแปรออปเจกต์ (object variable) ขึ้นมาเพื่อให้ตัวแปรนั้นอ้างอิงถึงออปเจกต์ที่ต้องการได้

ในการสร้างตัวแปรออปเจกต์ สามารถทำได้โดยการใช้ประโยคสำหรับประกาศตัวแปรทั่วไป ดังนี้ Dim , ReDim , Static , Private , Public หรือ Global โดยมีรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 [Dim | ReDim | Static | Private | Public | Global] variablename As [New]
 ในวารกรณ์ใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 objecttype

ประโยค Set

รูปแบบการใช้

Set objective = { [New] objectexpression | Nothing }

กำหนดให้ตัวแปรออปเจ็กต์อ้างอิงถึงออปเจ็กต์

ก่อนที่จะอ้างอิงด้วยตัวแปรออปเจ็กต์ด้วยประโยคนี้ เราจะต้องสร้างตัวแปรออปเจ็กต์ขึ้นมา
เสียก่อนด้วยประโยค Dim , ReDim , Static หรือ Global



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOGO.FRM

Option Explicit

Sub Form_KeyPress (KeyAscii As Integer)

On Error Resume Next

form7.Show

Unload Me

form8.Show

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลิสตติง ไฟล์ไมคูลของ
Welcome to Transportation Problem Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WELCOME.FRM

Option Explicit

Sub Command3D1_Click (Index As Integer)

 Unload Me

 form28.Show

End Sub

Sub Command3D2_Click (Index As Integer)

 Unload Me

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิสต์ไฟล์โมดูลของ
ฟอรัม Menu of Transportation Problem Program



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.FRM

Option Explicit

Sub Command3D1_Click (Index As Integer)

form35.Show

End Sub

Sub Command3D2_Click (Index As Integer)

form36.Show

End Sub

Sub Command3D3_Click (Index As Integer)

Unload Me

form11.Show

End Sub

Sub Command3D4_Click (Index As Integer)

form1.Show

End Sub

Sub Command3D5_Click (Index As Integer)

Unload Me

End

End Sub

Sub Command3D6_Click (Index As Integer)

form7.Cls

form9.Show 1

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิสต์ไฟล์โมดูลของ
ฟอร์ม Instruction of Control in Program



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCT.FRM

Option Explicit

Sub Command3D9_Click ()

Unload Me

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิสต์ไฟล์โมดูลของ
ฟอร์ม Fill number of Source and Destination



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START1.FRM

Option Explicit

Sub Command3D2_Click ()

 Unload Me

End Sub

Sub Command1_Click ()

 Unload Me

End Sub

Sub Command3D1_Click ()

Dim chk%

 chk% = 0

 row_t = CVar(text1.text)

 col_t = CVar(text2.text)

 If text1.text = "" Or text2.text = "" Then

 chk% = 1

 Else

 If CVar(text1.text) > 12 Or CVar(text2.text) > 12 Then

 chk% = 1

 End If

 End If

 If chk% = 1 Then

 form25.Show

 Else

 form13.Show

 End If

End Sub

Sub Command3D3_Click ()

 form10.Show

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Form_Load ()
```

```
On Error Resume Next
```

```
text1.text = ""
```

```
text2.text = ""
```

```
End Sub
```

```
Sub text1_change ()
```

```
row_t = CVar(text1.text)
```

```
End Sub
```

```
Sub Text1_KeyPress (keyascii As Integer)
```

```
If (keyascii = 13) Then
```

```
If text1.text = " " Then
```

```
form25.Show MsgBox " Please check ! "
```

```
End If
```

```
End If
```

```
End Sub
```

```
Sub Text2_Change ()
```

```
col_t = Val(text2.text)
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STNDRD.FRM

Option Explicit

Sub FillRowOder ()

```
Dim i%, j%  
grid1.Col = CInt(col_t) + 1  
For i% = 1 To CInt(row_t)  
    grid1.Row = i%  
    grid1.Text = "" & rw(i%)
```

```
Next i%
```

```
grid1.Row = CInt(row_t) + 1
```

```
For i% = 1 To CInt(col_t)  
    grid1.Col = i%  
    grid1.Text = "" & clmn(i%)
```

```
Next i%
```

```
End Sub
```

Sub Command3D1_Click (Index As Integer)

```
Unload Me  
form14.Show
```

```
End Sub
```

Sub Command3D2_Click (Index As Integer)

```
form10.Show
```

```
End Sub
```

Sub Form_Load ()

```
Dim i%, j%, k%, sum_ai%, sum_bi%
```

```
On Error Resume Next
```

```
sum_ai% = 0 find dummy?
```

```
sum_bi% = 0
```

```
For i% = 1 To CInt(row_t)
```

```
    sum_ai% = sum_ai% + CInt(rw(i%))
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For j% = 1 To CInt(col_t)
    sum_bi% = sum_bi% + CInt(clmn(j%))
Next j%
If sum_ai% <> sum_bi% Then        check dummy
    If sum_ai% > sum_bi% Then
        col_t = CInt(col_t) + 1
        clmn(CInt(col_t)) = sum_ai% - sum_bi%
    Else
        row_t = CInt(row_t) + 1
        rw(CInt(row_t)) = sum_bi% - sum_ai%
    End If
End If
grid1.Rows = CInt(row_t) + 2
grid1.Cols = CInt(col_t) + 2
grid1.Col = 0          Fill แหล่งผลิต
For i% = 1 To CInt(row_t)
    grid1.Row = i%
    grid1.Text = i%
Next i%
grid1.Row = 0          Fill คลังสินค้า
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = i%
Next i%
For k% = 1 To CInt(row_t)
    For j% = 1 To CInt(col_t)
        If trn_vl(k%, j%) = "" Then
            trn_vl(k%, j%) = 0
        End If
        grid1.Row = k%
        grid1.Col = j%
        grid1.Text = trn_vl(k%, j%)
        cost(k%, j%) = 0
    Next j%
Next k%

```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Next k%
FillRowOder
End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START2.FRM

Option Explicit

Sub FillRowOder ()

```
Dim i%, j%
grid1.Col = CInt(col_t) + 1
For i% = 1 To CInt(row_t)
    grid1.Row = i%
    grid1.Text = " a" & i%
Next i%
grid1.Row = CInt(row_t) + 1
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = " b" & i%
Next i%
```

End Sub

Sub Command3D1_Click ()

```
Dim i%, j%, chk%
chk% = 1
For i% = 1 To CInt(row_t)
    For j% = 1 To CInt(col_t)
        grid1.Row = i%
        grid1.Col = j%
        If grid1.Text = "" Then
            chk% = 0
        Else
            trn_vl(i%, j%) = CVar(grid1.Text)
        End If
    Next j%
Next i%
For i% = 1 To CInt(row_t)
    grid1.Col = CInt(col_t) + 1
    grid1.Row = i%
```

เอกสารนี้เป็นทรัพย์สินส่วนบุคคลของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

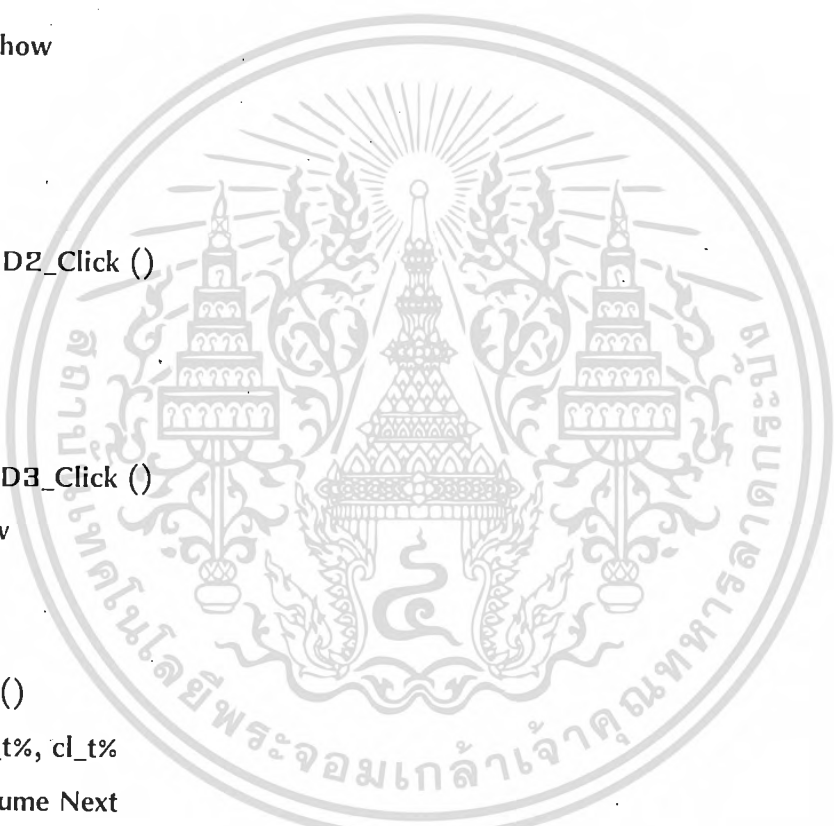
    rw(i%) = CVar(grid1.Text)
Next i%
For i% = 1 To CInt(col_t)
    grid1.Row = CInt(row_t) + 1
    grid1.Col = i%
    clmn(i%) = CVar(grid1.Text)
Next i%
If chk% = 0 Then
    form25.Show
Else
    form29.Show
End If
End Sub

Sub Command3D2_Click ()
    Unload Me
End Sub

Sub Command3D3_Click ()
    form10.Show
End Sub

Sub Form_Load ()
    Dim i%, j%, rw_t%, cl_t%
    On Error Resume Next
    grid1.Rows = CInt(row_t) + 2
    grid1.Cols = CInt(col_t) + 2
    grid1.Col = 0      Fill แหล่งผลิต
    For i% = 1 To CInt(row_t)
        grid1.Row = i%
        grid1.Text = i%
    Next i%
    grid1.Row = 0      Fill คลังสินค้า
    For i% = 1 To CInt(col_t)

```



```
grid1.Text = i%
```

```
Next i%
```

```
FillRowOrder
```

```
End Sub
```

```
Sub Grid1_Click ()
```

```
text1.Text = grid1.Text
```

```
label1.Caption = "C " & grid1.Row & grid1.Col
```

```
End Sub
```

```
Sub Grid1_KeyPress (keyascii As Integer)
```

```
If keyascii = 13 Then
```

```
label1.Caption = "fg"
```

```
End If
```

```
End Sub
```

```
Sub grid1_RowColchange ()
```

```
If grid1.Col Or grid1.Row Then
```

```
text1.Text = grid1.Text
```

```
label1.Caption = "C " & grid1.Row & grid1.Col
```

```
End If
```

```
End Sub
```

```
Sub text1_change ()
```

```
grid1.Text = text1.Text
```

```
End Sub
```

```
Sub Text1_Click ()
```

```
text1.Text = ""
```

```
End Sub
```

```
Sub Text1_KeyPress (keyascii As Integer)
```

```
Dim i%, j%
```

```
If keyascii = 13 Then
```

เอกสาร If text1.Text = "" Then สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

form25.Show

Else

grid1.Text = text1.Text

grid1_RowColchange

End If

End If

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ ไฟล์โมดูลของ
ฟอร์ม รูปแบบการแก้ปัญหาการขนส่ง (Transportation Solution)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START3.FRM

Option Explicit

Sub Command3D1_Click ()

form15.Show

End Sub

Sub Command3D2_Click ()

form10.Show

End Sub

Sub Command3D3_Click ()

Unload Me

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub



ลิขสิทธิ์ไฟล์ไมโครของ
ฟอร์ม รูปแบบการ1หาผลเฉลยเบื้องต้นของปัญหาการขนส่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START4.FRM

Option Explicit

Sub Command1_Click ()

 Unload Me

End Sub

Sub Command3D1_Click ()

 form2.Show

End Sub

Sub Command3D2_Click ()

 form17.Show

End Sub

Sub Command3D3_Click ()

 form4.Show

End Sub

Sub Command3D4_Click ()

 form10.Show

End Sub

Sub Form_Load ()

 On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NW1.FRM

Option Explicit

Dim min%

Dim i%, k%, j%

Dim clmn_nw(13), rw_nw(13) As Variant

Sub chg_vl ()

For i% = 1 To 13

clmn_nw(i%) = clmn(i%)

rw_nw(i%) = rw(i%)

Next i%

For i% = 1 To CInt(row_t)

For j% = 1 To CInt(col_t)

cost(i%, j%) = 0

Next j%

Next i%

End Sub

Sub FillRowOder ()

grid1.Col = CInt(col_t) + 1

For i% = 1 To CInt(row_t)

grid1.Row = i%

grid1.Text = "" & rw_nw(i%)

Next i%

grid1.Row = CInt(row_t) + 1

For i% = 1 To CInt(col_t)

grid1.Col = i%

grid1.Text = "" & clmn_nw(i%)

Next i%

End Sub

Sub North_West ()

k% = 1

j% = 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Do While k% < CInt(row_t) + 1 And j% < CInt(col_t) + 1
```

```
    min% = 0
```

```
    If CInt(rw_nw(k%)) > CInt(clmn_nw(j%)) Then
```

```
        min% = CInt(clmn_nw(j%))
```

```
        cost(k%, j%) = min%
```

```
        rw_nw(k%) = CInt(rw_nw(k%)) - CInt(clmn_nw(j%))
```

```
        j% = j% + 1
```

```
    Else
```

```
        min% = CInt(rw_nw(k%))
```

```
        clmn_nw(j%) = CInt(clmn_nw(j%)) - CInt(rw_nw(k%))
```

```
        cost(k%, j%) = min%
```

```
        k% = k% + 1
```

```
    End If
```

```
Loop
```

```
For k% = 1 To CInt(row_t)
```

```
    For j% = 1 To CInt(col_t)
```

```
        grid1.Row = k%
```

```
        grid1.Col = j%
```

```
        grid1.Text = "" & trn_vl(k%, j%) & ", " & cost(k%, j%)
```

```
    Next j%
```

```
Next k%
```

```
End Sub
```

```
Sub text1_change ()
```

```
End Sub
```

```
Sub Command1_Click ()
```

```
    Unload Me
```

```
End Sub
```

```
Sub Command3D1_Click ()
```

```
    form1B.Show
```

```
End Sub
```

```
Sub Command3D2_Click ()
```

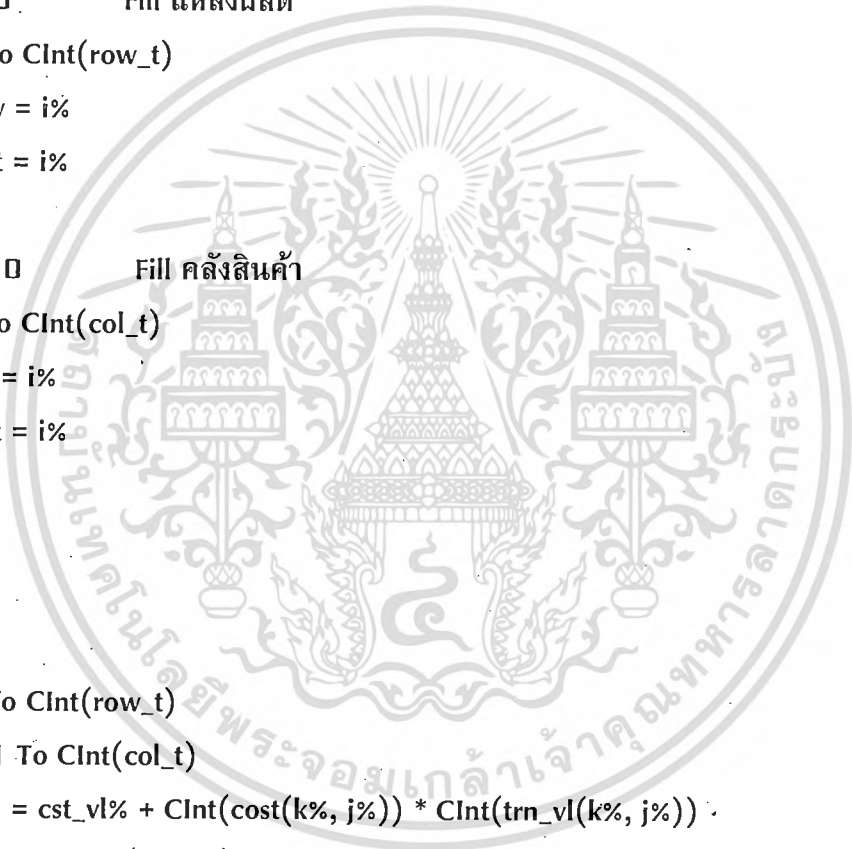
ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

form1.D.Show
End Sub

Sub Form_Load ()
Dim cst_vl%
    On Error Resume Next
    grid1.Rows = CInt(row_t) + 2
    grid1.Cols = CInt(col_t) + 2
    chg_vl
    grid1.Col = 0      Fill แหล่งผลิต
    For i% = 1 To CInt(row_t)
        grid1.Row = i%
        grid1.Text = i%
    Next i%
    grid1.Row = 0      Fill คลังสินค้า
    For i% = 1 To CInt(col_t)
        grid1.Col = i%
        grid1.Text = i%
    Next i%
    FillRowOrder
    NorthWest
    cst_vl% = 0
    For k% = 1 To CInt(row_t)
        For j% = 1 To CInt(col_t)
            cst_vl% = cst_vl% + CInt(cost(k%, j%)) * CInt(trn_vl(k%, j%))
            grid1.Text = cost(k%, j%)
            cost(k%, j%) = 0
        Next j%
    Next k%
    label1.Caption = "เท่ากับ " & cst_vl%
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลิสต์ดิ้ง ไฟล์โมดูลของ
ฟอร์ม Lowest First Costs Rule

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LST-S.FRM

Option Explicit

Dim i%, j%, k%, min%, min1%, min2%, cst_vl%

Dim clmn_ls(13), rw_ls(13) As Variant

Sub chg_vl ()

For i% = 1 To 13

clmn_ls(i%) = clmn(i%)

rw_ls(i%) = rw(i%)

Next i%

For i% = 1 To CInt(row_t)

For j% = 1 To CInt(col_t)

cost(i%, j%) = 0

Next j%

Next i%

End Sub

Sub FillRowOder ()

grid1.Col = CInt(col_t) + 1

For i% = 1 To CInt(row_t)

grid1.Row = i%

grid1.Text = "" & rw_ls(i%)

Next i%

grid1.Row = CInt(row_t) + 1

For i% = 1 To CInt(col_t)

grid1.Col = i%

grid1.Text = "" & clmn_ls(i%)

Next i%

End Sub

Sub ls ()

Dim x%, y%, ls_chk%, chk%, run_cl%, run_rw%

ReDim rw_chk(13), clmn_chk(13) As Variant

Dim temp%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Erase rw_chk, clmn_chk, cost
run_cl% = 1
run_rw% = 1
ls_chk% = 1
Do While ls_chk% < CInt(row_t) + CInt(col_t)
    temp% = 999          เริ่มต้นหาผลเฉลี่ย
    For k% = 1 To CInt(row_t)
        For j% = 1 To CInt(col_t)
            chk% = 0
            For i% = 1 To 13          ไม่ตกลงที่ค่าเดิม
                If k% = (rw_chk(i%)) Or j% = (clmn_chk(i%)) Then
                    chk% = 1
                End If
            Next i%
            If chk% <> 1 Then
                If CInt(trn_vl(k%, j%)) < temp% Then
                    temp% = CInt(trn_vl(k%, j%))
                    x% = k%
                    y% = j%
                End If
            End If
        Next j%
    Next k%
    min1% = CInt(rw_ls(x%))
    min2% = CInt(clmn_ls(y%))
    If CInt(rw_ls(x%)) > CInt(clmn_ls(y%)) Then
        cost(x%, y%) = clmn_ls(y%)
        rw_ls(x%) = min1% - min2%
        clmn_chk(run_cl%) = y%
        run_cl% = run_cl% + 1
        label2.Caption = " cl_run : " & clmn_chk(run_cl% - 1)
    Else
        clmn_ls(y%) = min2% - min1%
        cost(x%, y%) = rw_ls(x%)

```

```
rw_chk(run_rw%) = x%
```

```
run_rw% = run_rw% + 1
```

```
label2.Caption = " rw_run : " & rw_chk(run_rw% - 1)
```

```
End If
```

```
ls_chk% = ls_chk% + 1
```

```
Loop
```

```
For k% = 1 To CInt(row_t)
```

```
For j% = 1 To CInt(col_t)
```

```
grid1.Row = k%
```

```
grid1.Col = j%
```

```
grid1.Text = "" & trn_vl(k%, j%) & ", " & cost(k%, j%)
```

```
Next j%
```

```
Next k%
```

```
End Sub
```

```
Sub Command1_Click ()
```

```
Unload Me
```

```
End Sub
```

```
Sub Command3D1_Click ()
```

```
form1B.Show
```

```
End Sub
```

```
Sub Command3D2_Click ()
```

```
form1D.Show
```

```
End Sub
```

```
Sub Form_Load ()
```

```
On Error Resume Next
```

```
chg_vl
```

```
grid1.Rows = CInt(row_t) + 2      ขนาดของตาราง
```

```
grid1.Cols = CInt(col_t) + 2
```

```
grid1.Col = 0      Fill แหล่งผลิต
```

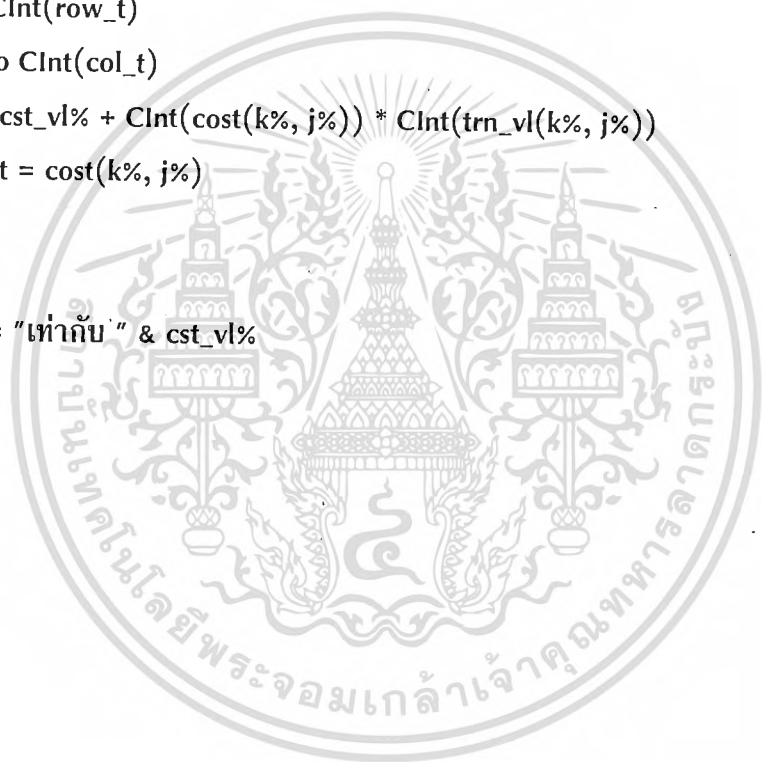
```
For i% = 1 To CInt(row_t)
```


เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตให้ดำเนินการตามกฎหมายที่เกี่ยวข้อง

```

grid1.Text = i%
Next i%
grid1.Row = 0      Fill คลังสินค้า
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = i%
Next i%
FillRowOder
Is
cst_vl% = 0
For k% = 1 To CInt(row_t)
    For j% = 1 To CInt(col_t)
        cst_vl% = cst_vl% + CInt(cost(k%, j%)) * CInt(trn_vl(k%, j%))
        grid1.Text = cost(k%, j%)
    Next j%
Next k%
label1.Caption = "เท่ากับ" & cst_vl%
End Sub

```





ลิสติง ไฟล์โมดูลของ
ฟอร์ม Vogel 's Approximation Method

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VGL.FRM

Option Explicit

Dim i%, j%, k%, min%, min1%, min2%, cst_vl%

Dim clmn_vgl(13), rw_vgl(13) As Variant

Sub chg_vl ()

For i% = 1 To 13

clmn_vgl(i%) = clmn(i%)

rw_vgl(i%) = rw(i%)

Next i%

For i% = 1 To CInt(row_t)

For j% = 1 To CInt(col_t)

cost(i%, j%) = 0

Next j%

Next i%

End Sub

Sub FillRowOder ()

grid1.Col = CInt(col_t) + 1

For i% = 1 To CInt(row_t)

grid1.Row = i%

grid1.Text = "" & rw_vgl(i%)

Next i%

grid1.Row = CInt(row_t) + 1

For i% = 1 To CInt(col_t)

grid1.Col = i%

grid1.Text = "" & clmn_vgl(i%)

Next i%

End Sub

Sub Vgl ()

Dim x%, y%, u%, v%, z%, run_cl%, run_rw%, vgl_chk%, chk%

Dim mxdf_rw%, mxdf_clm%, low_trn%, temp%, temp1%, temp2%

ReDim rw_chk(13), clmn_chk(13) As Variant

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ReDim rw_df(13), clmn_df(13)
run_cl% = 1
run_rw% = 1
vgl_chk% = 1
Do While vgl_chk% < CInt(row_t) + CInt(col_t)
  For i% = 1 To 13
    rw_df(i%) = 0
    clmn_df(i%) = 0
  Next i%

```

```

For k% = 1 To CInt(row_t) Find diff.row

```

```

  chk% = 0

```

```

  For i% = 1 To 13

```

```

    If k% = (rw_chk(i%)) Then

```

```

      chk% = 1

```

```

    End If

```

```

  Next i%

```

```

  If chk% <> 1 Then

```

```

    temp1% = 999

```

```

    temp2% = 999

```

```

    For j% = 1 To CInt(col_t)

```

```

      chk% = 0

```

```

      For i% = 1 To 13

```

```

        If j% = (clmn_chk(i%)) Then

```

```

          chk% = 1

```

```

        End If

```

```

      Next i%

```

```

    If chk% <> 1 Then

```

```

      If trn_vl(k%, j%) < temp1% Then

```

```

        temp1% = trn_vl(k%, j%)

```

```

      End If

```

```

    End If

```

```

  Next j%

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For j% = 1 To CInt(col_t)

chk% = 0

For i% = 1 To 13 **ไม่ตกลงที่ค่าเดิม**

If j% = (clmn_chk(i%)) Then

chk% = 1

End If

Next i%

If chk% <> 1 Then

If trn_vl(k%, j%) < temp2% Then

If trn_vl(k%, j%) = temp1% Then

z% = z% + 1

Else

temp2% = trn_vl(k%, j%)

End If

End If

End If

Next j%

If z% > 1 Then

temp2% = temp1%

End If

rw_df(k%) = temp2% - temp1%

End If

Next k%

For j% = 1 To CInt(col_t) **Find diff.col**

chk% = 0

For i% = 1 To 13 **ไม่ตกลงที่ค่าเดิม**

If j% = (clmn_chk(i%)) Then

chk% = 1

End If

Next i%

If chk% <> 1 Then

temp1% = 999

temp2% = 999

For k% = 1 To CInt(row_t)

chk% = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For i% = 1 To 13 ไม่ตกลงที่ค่าเดิม

 If k% = (rw_chk(i%)) Then

 chk% = 1

 End If

Next i%

If chk% <> 1 Then

 If trn_vl(k%, j%) < temp1% Then

 temp1% = trn_vl(k%, j%)

 End If

End If

Next k%

z% = 0

For k% = 1 To Clnt(row_t)

 chk% = 0

 For i% = 1 To 13 ไม่ตกลงที่ค่าเดิม

 If k% = (rw_chk(i%)) Then

 chk% = 1

 End If

 Next i%

 If chk% <> 1 Then

 If trn_vl(k%, j%) < temp2% Then

 If trn_vl(k%, j%) = temp1% Then

 z% = z% + 1

 Else

 temp2% = trn_vl(k%, j%)

 End If

 End If

 End If

Next k%

If z% > 1 Then

 temp2% = temp1%

End If

clmn_df(j%) = temp2% - temp1%

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mxdf_rw% = 0 Find most diff.row , diff.colmn
mxdf_clm% = 0
For k% = 1 To CInt(row_t)
  If rw_df(k%) > mxdf_rw% Then
    mxdf_rw% = rw_df(k%)
    x% = k%
  End If
Next k%
For j% = 1 To CInt(col_t)
  If clmn_df(j%) > mxdf_clm% Then
    mxdf_clm% = clmn_df(j%)
    y% = j%
  End If
Next j%
label2.Caption = "rw: " & mxdf_rw% & " cl:" & mxdf_clm%
low_trn% = 999
If mxdf_rw% > mxdf_clm% Then Find most diff. value
  For j% = 1 To CInt(col_t) Find lowest tran.value
    chk% = 0
    For i% = 1 To 13
      If j% = (clmn_chk(i%)) Then
        chk% = 1
      End If
    Next i%
    If chk% <> 1 Then
      If trn_vl(x%, j%) < low_trn% Then
        low_trn% = trn_vl(x%, j%)
        u% = x%
        v% = j%
      End If
    End If
  Next j%
Else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

For k% = 1 To Clnt(row_t)

chk% = 0

For i% = 1 To 13 ไม่ตกลงที่ค่าเดิม

If k% = (rw_chk(i%)) Then

chk% = 1

End If

Next i%

If chk% <> 1 Then

If trn_vl(k%, y%) < low_trn% Then

low_trn% = trn_vl(k%, y%)

u% = k%

v% = y%

End If

End If

Next k%

End If

min1% = Clnt(rw_vgl(u%))

min2% = Clnt(clmn_vgl(v%))

If Clnt(rw_vgl(u%)) > Clnt(clmn_vgl(v%)) Then

cost(u%, v%) = clmn_vgl(y%)

rw_vgl(u%) = min1% - min2%

clmn_chk(run_cl%) = v%

run_cl% = run_cl% + 1

label2.Caption = " cl_run : " & clmn_chk(run_cl% - 1)

Else

clmn_vgl(v%) = min2% - min1%

cost(u%, v%) = rw_vgl(u%)

rw_chk(run_rw%) = u%

run_rw% = run_rw% + 1

label2.Caption = " rw_run : " & rw_chk(run_rw% - 1)

End If

vgl_chk% = vgl_chk% + 1

Loop

For k% = 1 To Clnt(row_t)

เอกสารนี้ได้รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
grid1.Row = k%
grid1.Col = j%
grid1.Text = "" & trn_vl(k%, j%) & ", " & cost(k%, j%)
```

```
Next j%
```

```
Next k%
```

```
End Sub
```

```
Sub Command1_Click ()
```

```
Unload Me
```

```
End Sub
```

```
Sub Command3D1_Click ()
```

```
form1B.Show
```

```
End Sub
```

```
Sub Command3D2_Click ()
```

```
form1D.Show
```

```
End Sub
```

```
Sub Form_Load ()
```

```
On Error Resume Next
```

```
chg_vl
```

```
grid1.Rows = CInt(row_t) + 2
```

```
grid1.Cols = CInt(col_t) + 2
```

```
grid1.Col = 0      Fill แหล่งผลิต
```

```
For i% = 1 To CInt(row_t)
```

```
    grid1.Row = i%
```

```
    grid1.Text = i%
```

```
Next i%
```

```
grid1.Row = 0      Fill คลังสินค้า
```

```
For i% = 1 To CInt(col_t)
```

```
    grid1.Col = i%
```

```
    grid1.Text = i%
```

```
Next i%
```

```
FillRowOrder
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vgl

cst_vl% = 0

For k% = 1 To Clnt(row_t)

For j% = 1 To Clnt(col_t)

cst_vl% = cst_vl% + Clnt(cost(k%, j%)) * Clnt(trn_vl(k%, j%))

Next j%

Next k%

label1.Caption = "เท่ากับ " & cst_vl%

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ไฟล์ไมโครดของ
ฟอร์ม รูปแบบการหาผลเฉลี่ยตามเป้าหมายของปัญหาการขนส่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START5.FRM

Option Explicit

Sub Command1_Click ()

Unload Me

End Sub

Sub Command3D1_Click ()

form30.Show

End Sub

Sub Command3D2_Click ()

form31.Show

End Sub

Sub Command3D3_Click ()

form10.Show

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิสต์ดิ้ง ไฟล์โมดูลของ
ฟอรัม Stepping Stone Method



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STP.FRM

Option Explicit

Dim i%, j%, k%, min%, min1%, min2%, cst_vl%

Dim clmn_stp(13), rw_stp(13) As Variant

Sub chg_vl ()

For i% = 1 To 13

clmn_stp(i%) = clmn(i%)

rw_stp(i%) = rw(i%)

Next i%

End Sub

Sub FillRowOder ()

grid1.Col = CInt(col_t) + 1

For i% = 1 To CInt(row_t)

grid1.Row = i%

grid1.Text = "" & rw_stp(i%)

Next i%

grid1.Row = CInt(row_t) + 1

For i% = 1 To CInt(col_t)

grid1.Col = i%

grid1.Text = "" & clmn_stp(i%)

Next i%

End Sub

Function fnd_dev (ByVal m%, ByVal n%) As Integer

Dim dev%, chkend%, chk%, chk_bg%

Dim fix_x%, fix_y%, run_x%, run_y%, u%, v%

Dim ik%, jk%

run_x% = m%

run_y% = n%

chkend% = 0

chk_bg% = 0

dev% = 999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For jk% = 1 To Int(col_t)
If chk_bg% <> 1 Then
If cost(run_x%, jk%) <> 0 Then
chk% = 0
For u% = 1 To Int(row_t)
If cost(u%, jk%) <> 0 Then
chk% = chk% + 1
End If
Next u%
If chk% > 1 Then
run_y% = jk%
chk_bg% = 1
End If
End If
End If
Next jk%
If cost(run_x%, run_y%) < dev% Then
dev% = cost(run_x%, run_y%)
End If
fix_x% = run_x%
label1.Caption = "เท่ากับ " & run_x% & run_y%
Do While chkend% <> 1 Check end of path
For ik% = 1 To Int(row_t)
If cost(ik%, run_y%) <> 0 And ik% <> fix_x% Then
chk% = 0
For v% = 1 To Int(col_t)
If cost(ik%, v%) <> 0 Then
chk% = chk% + 1
End If
Next v%
If chk% > 1 Then
run_x% = ik%
End If
End If
Next ik%

```

```

fix_y% = run_y%
label1.Caption = "เท่ากับ " & run_x% & run_y% pth%
For jk% = 1 To Int(col_t)
    If jk% = n% And cost(run_x%, jk%) <> 0 Then
        run_y% = jk%
        chkend% = 1
    End If
Next jk%
If chkend% <> 1 Then
    For jk% = 1 To Int(col_t)
        If cost(run_x%, jk%) <> 0 And jk% <> fix_y% Then
            chk% = 0
            For u% = 1 To Int(row_t)
                If cost(u%, jk%) <> 0 Then
                    chk% = chk% + 1
                End If
            Next u%
            If chk% > 1 Then
                run_y% = jk%
            End If
        End If
    Next jk%
End If
If cost(run_x%, run_y%) < dev% Then
    dev% = cost(run_x%, run_y%)
End If
fix_x% = run_x%
Loop
label1.Caption = "เท่ากับ " & dev%
fnd_dev = dev%
End Function

```

Function fnd_pth (ByVal m%, ByVal n%) As Integer

Dim pth%, chkend%, chk%, chk_bg%, fix_x%, fix_y%, run_x%, run_y%, u%, v%

run_x% = m%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

End If
Next i%
pth% = pth% + trn_vl(run_x%, run_y%)
fix_y% = run_y%
label1.Caption = "เท่ากับ " & run_x% & run_y% pth%
For j% = 1 To Int(col_t)
  If j% = n% And cost(run_x%, j%) <> 0 Then
    run_y% = j%
    chkend% = 1
  End If
Next j%
If chkend% <> 1 Then
  For j% = 1 To Int(col_t)
    If cost(run_x%, j%) <> 0 And j% <> fix_y% Then
      chk% = 0
      For u% = 1 To Int(row_t)
        If cost(u%, j%) <> 0 Then
          chk% = chk% + 1
        End If
      Next u%
      If chk% > 1 Then
        run_y% = j%
      End If
    End If
  Next j%
End If
pth% = pth% - trn_vl(run_x%, run_y%)
fix_x% = run_x%
Loop
label1.Caption = "เท่ากับ " & pth%
fnd_pth = pth%
End Function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sub step_stone ()

Dim n%, m%

Dim dv_vl%

Dim run_x%, run_y%, u%, v%, ir%, jr%

Dim chkend%, chk%, chk_bg%, fix_x%, fix_y%

dv_vl% = fnd_dev(dv_x%, dv_y%)

run_x% = dv_x%

run_y% = dv_y%

cost(run_x%, run_y%) = dv_vl%

chkend% = 0

chk_bg% = 0

For jr% = 1 To Int(col_t)

If chk_bg% <> 1 Then

If cost(run_x%, jr%) <> 0 Then

chk% = 0

For u% = 1 To Int(row_t)

If cost(u%, jr%) <> 0 Then

chk% = chk% + 1

End If

Next u%

If chk% > 1 Then

If run_y% < CInt(col_t) / 2 Then

chk_bg% = 1

End If

run_y% = jr%

End If

End If

End If

Next jr%

cost(run_x%, run_y%) = cost(run_x%, run_y%) - dv_vl%

fix_x% = run_x%

label1.Caption = "เท่ากับ " & run_x% & run_y%

Do While chkend% <> 1 Check end of path

For ir% = 1 To Int(row_t)

If cost(ir%, run_y%) <> 0 And ir% <> fix_x% Then

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chk% = 0
For v% = 1 To Int(col_t)
  If cost(ir%, v%) <> 0 Then
    chk% = chk% + 1
  End If
Next v%
If chk% > 1 Then
  run_x% = ir%
End If
End If
Next ir%
cost(run_x%, run_y%) = cost(run_x%, run_y%) + dv_vl%
fix_y% = run_y%
label1.Caption = "เท่ากับ " & run_x% & run_y% ) pth%
For jr% = 1 To Int(col_t)
  If jr% = dv_y% And cost(run_x%, jr%) <> 0 Then
    run_y% = jr%
    chkend% = 1
  End If
Next jr%
If chkend% <> 1 Then
  For jr% = 1 To Int(col_t)
    If cost(run_x%, jr%) <> 0 And jr% <> fix_y% Then
      chk% = 0
      For u% = 1 To Int(row_t)
        If cost(u%, jr%) <> 0 Then
          chk% = chk% + 1
        End If
      Next u%
      If chk% > 1 Then
        run_y% = jr%
      End If
    End If
  Next jr%
End If

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cost(run_x%, run_y%) = cost(run_x%, run_y%) - dv_vl%
```

```
fix_x% = run_x%
```

```
label1.Caption = "เท่ากับ " & chkend% cost(run_x%, run_y%) pth%
```

```
Loop
```

```
End If
```

```
For k% = 1 To CInt(row_t)
```

```
For jr% = 1 To CInt(col_t)
```

```
grid1.Row = k%
```

```
grid1.Col = jr%
```

```
grid1.Text = "" & trn_vl(k%, jr%) & ", " & cost(k%, jr%)
```

```
Next jr%
```

```
Next k%
```

```
End Sub
```

```
Sub Command1_Click ()
```

```
Unload Me
```

```
End Sub
```

```
Sub Command3D1_Click ()
```

```
form1B.Show
```

```
End Sub
```

```
Sub Command3D2_Click ()
```

```
form1B.Show
```

```
End Sub
```

```
Sub Form_Load ()
```

```
On Error Resume Next
```

```
chg_vl
```

```
grid1.Rows = CInt(row_t) + 2
```

```
grid1.Cols = CInt(col_t) + 2
```

```
grid1.Col = 0      Fill แหล่งผลิต
```

```
For i% = 1 To CInt(row_t)
```

```
grid1.Row = i%
```

```
grid1.Text = i%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next i%
grid1.Row = 0      Fill คลังสินค้า
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = i%
Next i%
step_stone
grid1.Col = CInt(col_t) + 1
For i% = 1 To CInt(row_t)
    grid1.Row = i%
    grid1.Text = "" & rw(i%)
Next i%
grid1.Row = CInt(row_t) + 1
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = "" & clmn(i%)
Next i%
FillRowOder
cst_vl% = 0
For k% = 1 To CInt(row_t)
    For j% = 1 To CInt(col_t)
        cst_vl% = cst_vl% + CInt(cost(k%, j%)) * CInt(trn_vl(k%, j%))
    Next j%
Next k%
label1.Caption = "เท่ากับ " & cst_vl%
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ไฟล์โมดูลของ
ฟอร์ม Modified Distribution Method



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODI.FRM

Option Explicit

Dim i%, j%, k%, min%, min1%, min2%, cst_vl%

Dim clmn_mod(13), rw_mod(13) As Variant

Sub chg_vl ()

For i% = 1 To 13

clmn_mod(i%) = clmn(i%)

rw_mod(i%) = rw(i%)

Next i%

End Sub

Sub FillRowOder ()

grid1.Col = CInt(col_t) + 1

For i% = 1 To CInt(row_t)

grid1.Row = i%

grid1.Text = "" & rw_mod(i%)

Next i%

grid1.Row = CInt(row_t) + 1

For i% = 1 To CInt(col_t)

grid1.Col = i%

grid1.Text = "" & clmn_mod(i%)

Next i%

End Sub

Function fnd_dev (ByVal m%, ByVal n%) As Integer

Dim dev%, chkend%, chk%, chk_bg%

Dim fix_x%, fix_y%, run_x%, run_y%, u%, v%

Dim ik%, jk%

run_x% = m%

run_y% = n%

chkend% = 0

chk_bg% = 0

dev% = 999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For jk% = 1 To Int(col_t)
If chk_bg% <> 1 Then
If cost(run_x%, jk%) <> 0 Then
chk% = 0
For u% = 1 To Int(row_t)
If cost(u%, jk%) <> 0 Then
chk% = chk% + 1
End If
Next u%
If chk% > 1 Then
run_y% = jk%
chk_bg% = 1
End If
End If
End If
Next jk%
If cost(run_x%, run_y%) < dev% Then
dev% = cost(run_x%, run_y%)
End If
fix_x% = run_x%
label1.Caption = "เท่ากับ " & run_x% & run_y%
Do While chkend% <> 1 Check end of path
For ik% = 1 To Int(row_t)
If cost(ik%, run_y%) <> 0 And ik% <> fix_x% Then
chk% = 0
For v% = 1 To Int(col_t)
If cost(ik%, v%) <> 0 Then
chk% = chk% + 1
End If
Next v%
If chk% > 1 Then
run_x% = ik%
End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่วารณัติใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fix_y% = run_y%
label1.Caption = "เท่ากับ " & run_x% & run_y% pth%
For jk% = 1 To Int(col_t)
  If jk% = n% And cost(run_x%, jk%) <> 0 Then
    run_y% = jk%
    chkend% = 1
  End If
Next jk%
If chkend% <> 1 Then
  For jk% = 1 To Int(col_t)
    If cost(run_x%, jk%) <> 0 And jk% <> fix_y% Then
      chk% = 0
      For u% = 1 To Int(row_t)
        If cost(u%, jk%) <> 0 Then
          chk% = chk% + 1
        End If
      Next u%
      If chk% > 1 Then
        run_y% = jk%
      End If
    End If
  Next jk%
End If
If cost(run_x%, run_y%) < dev% Then
  dev% = cost(run_x%, run_y%)
End If
fix_x% = run_x%
Loop
label1.Caption = "เท่ากับ " & dev%
fnd_dev = dev%
End Function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function fnd_pth (ByVal m%, ByVal n%) As Integer

Dim pth%, chkend%, chk%, chk_bg%, fix_x%, fix_y%, run_x%, run_y%, u%, v%

run_x% = m%

run_y% = n%

chkend% = 0

chk_bg% = 0

pth% = trn_vl(m%, n%)

For j% = 1 To Int(col_t)

If chk_bg% <> 1 Then

If cost(run_x%, j%) <> 0 Then

chk% = 0

For u% = 1 To Int(row_t)

If cost(u%, j%) <> 0 Then

chk% = chk% + 1

End If

Next u%

If chk% > 1 Then

run_y% = j%

chk_bg% = 1

End If

End If

End If

Next j%

pth% = pth% - trn_vl(run_x%, run_y%)

fix_x% = run_x%

label1.Caption = "เท่ากับ " & run_x% & run_y%

Do While chkend% <> 1 Check end of path

For i% = 1 To Int(row_t)

If cost(i%, run_y%) <> 0 And i% <> fix_x% Then

chk% = 0

For v% = 1 To Int(col_t)

If cost(i%, v%) <> 0 Then

chk% = chk% + 1

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Next v%
If chk% > 1 Then
    run_x% = i%
End If
End If
Next i%
pth% = pth% + trn_vl(run_x%, run_y%)
fix_y% = run_y%
label1.Caption = "เท่ากับ " & run_x% & run_y% pth%
For j% = 1 To Int(col_t)
    If j% = n% And cost(run_x%, j%) <> 0 Then
        run_y% = j%
        chkend% = 1
    End If
Next j%
If chkend% <> 1 Then
    For j% = 1 To Int(col_t)
        If cost(run_x%, j%) <> 0 And j% <> fix_y% Then
            chk% = 0
            For u% = 1 To Int(row_t)
                If cost(u%, j%) <> 0 Then
                    chk% = chk% + 1
                End If
            Next u%
            If chk% > 1 Then
                run_y% = j%
            End If
        End If
    Next j%
End If
Next j%
pth% = pth% - trn_vl(run_x%, run_y%)
fix_x% = run_x%
Loop

```

label1.Caption = "เท่ากับ " & pth%

fnd_pth = pth%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีให้นำไปใช้

End Function

Sub modi_stp ()

Dim n%, m%

Dim dv_vl%

Dim run_x%, run_y%, u%, v%, ir%, jr%

Dim chkend%, chk%, chk_bg%, fix_x%, fix_y%

dv_vl% = fnd_dev(dv_x%, dv_y%)

run_x% = dv_x%

run_y% = dv_y%

cost(run_x%, run_y%) = dv_vl%

chkend% = 0

chk_bg% = 0

For jr% = 1 To Int(col_t)

If chk_bg% <> 1 Then

If cost(run_x%, jr%) <> 0 Then

chk% = 0

For u% = 1 To Int(row_t)

If cost(u%, jr%) <> 0 Then

chk% = chk% + 1

End If

Next u%

If chk% > 1 Then

run_y% = jr%

chk_bg% = 1

End If

End If

End If

Next jr%

cost(run_x%, run_y%) = cost(run_x%, run_y%) - dv_vl%

fix_x% = run_x%

label1.Caption = "เท่ากับ " & cost(run_x%, run_y%)

Do While chkend% <> 1 Check end of path

For ir% = 1 To Int(row_t)

If cost(ir%, run_y%) <> 0 And ir% <> fix_x% Then

```

chk% = 0
For v% = 1 To Int(col_t)
  If cost(ir%, v%) <> 0 Then
    chk% = chk% + 1
  End If
Next v%
If chk% > 1 Then
  run_x% = ir%
End If
End If
Next ir%
cost(run_x%, run_y%) = cost(run_x%, run_y%) + dv_vl%
fix_y% = run_y%
label1.Caption = "เท่ากับ " & cost(run_x%, run_y%) pth%
For jr% = 1 To Int(col_t)
  If jr% = dv_y% And cost(run_x%, jr%) <> 0 Then
    run_y% = jr%
    chkend% = 1
  End If
Next jr%
If chkend% <> 1 Then
  For jr% = 1 To Int(col_t)
    If cost(run_x%, jr%) <> 0 And jr% <> fix_y% Then
      chk% = 0
      For u% = 1 To Int(row_t)
        If cost(u%, jr%) <> 0 Then
          chk% = chk% + 1
        End If
      Next u%
      If chk% > 1 Then
        run_y% = jr%
      End If
    End If
  Next jr%
End If

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cost(run_x%, run_y%) = cost(run_x%, run_y%) - dv_vl%
fix_x% = run_x%
label1.Caption = "เท่ากับ " & chkend% cost(run_x%, run_y%) pth%
Loop
For k% = 1 To CInt(row_t)
  For jr% = 1 To CInt(col_t)
    grid1.Row = k%
    grid1.Col = jr%
    grid1.Text = "" & trn_vl(k%, jr%) & ", " & cost(k%, jr%)
  Next jr%
Next k%
End Sub

Sub Command1_Click ()
  Unload Me
End Sub

Sub Command3D1_Click ()
  form22.Show
End Sub

Sub Command3D2_Click ()
  form10.Show
End Sub

Sub Form_Load ()
  On Error Resume Next
  chg_vl
  grid1.Rows = CInt(row_t) + 2
  grid1.Cols = CInt(col_t) + 2
  grid1.Col = 0      Fill แหล่งผลิต
  For i% = 1 To CInt(row_t)
    grid1.Row = i%
    grid1.Text = i%
  Next i%

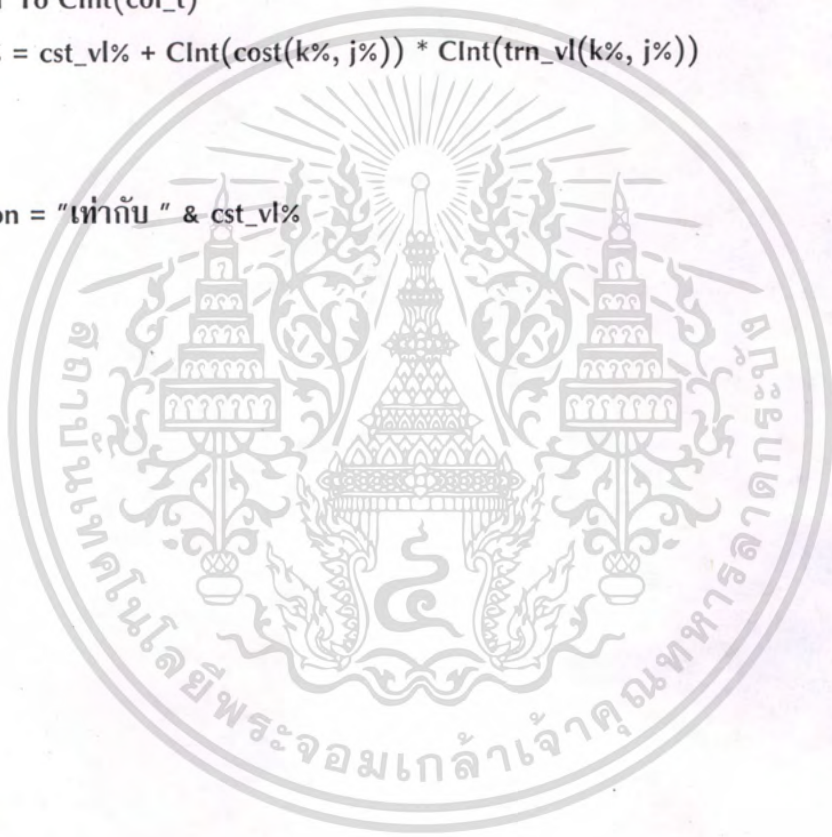
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

grid1.Row = 0      Fill คลังสินค้า
For i% = 1 To CInt(col_t)
    grid1.Col = i%
    grid1.Text = i%
Next i%
FillRowOder
modi_stp
cst_vl% = 0
For k% = 1 To CInt(row_t)
    For j% = 1 To CInt(col_t)
        cst_vl% = cst_vl% + CInt(cost(k%, j%)) * CInt(trn_vl(k%, j%))
    Next j%
Next k%
label1.Caption = "เท่ากับ " & cst_vl%
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPTION.FRM

Option Explicit

Sub Command1_Click ()

Unload Me

End Sub

Sub Command3D1_Click ()

Unload Me

form9.Show

End Sub

Sub Command3D3_Click ()

Unload Me

form7.Show

End Sub

Sub Command3D4_Click ()

Unload Me

End

End Sub

Sub Form_Load ()

On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลิขสิทธิ์ไฟล์ไมโครของ
ฟอร์ม Please Check

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ERR-FILL.FRM

Option Explicit

Sub Command3D1_Click (Index As Integer)

 Unload Me

End Sub

Sub Form_Load ()

 On Error Resume Next

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. นิกธ วัฒนพรม , การโปรแกรมเชิงเส้นเบื้องต้น (Elementary linear Programming)
คณะบริหารธุรกิจ สถาบันบัณฑิตพัฒนบริหารศาสตร์ , 2535
2. ประกอบ จิรกติ , การโปรแกรมเชิงเส้นจำนวนเต็ม (Integer linear Programming)
คณะบริหารธุรกิจ สถาบันบัณฑิตพัฒนบริหารศาสตร์ , 2535
3. Strum , Jay E , INTRODUCTION TO LINEAR PROCESSING , Sanfrancisco :
Holden - Day , 1972
4. Sultan Alan , LINEAR PROGRAMMING , Academic Press , Inc. , New York ,
1993
5. Winston Wayned L. , OPERATINGS REARCH , Application and Algorithm , 3 rd ed.
, Dusobery Press , California , 1994



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้