

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การวางแผนการผลิตพลังงานไฟฟ้าระยะสั้นโดยข้อกำหนดของยูนิตคอมมิตเมนต์

Security Constraint Unit Commitment for a Day Ahead

Generation Planning



โดย

นาย พลพิพัฒน์ อมรธนาบุบาล	45010552
นาย ชัยวัฒน์ มณีกรณ์	46010164
นาย ปกวงศ์ สมิตพงษ์	46010397

๒๗  
พ.๔๔๑๗  
๒๕๕๙

เลขหมู่.....  
เลข . บียน 72643  
วัน,เดือน,ปี 21 ส.ย. 2550

b. 117๙๐๗15  
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2549

การวางแผนการผลิตพลังงานไฟฟ้าระยะสั้นโดยข้อกำหนดของยูนิตคอมมิตเมนต์

**Security Constraint Unit Commitment for a Day Ahead**

**Generation Planning**



อาจารย์ที่ปรึกษา

ดร. ชาย ชมภูอินไหว

ร.ศ. มณฑล ลีลาจินดาไกรฤกษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

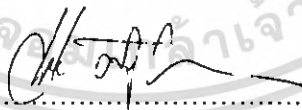
ภาควิชาวิศวกรรมไฟฟ้า

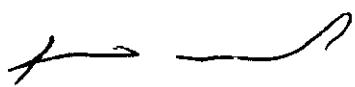
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวางแผนการผลิตพลังงานไฟฟ้าระยะสั้นโดยข้อกำหนดของ  
ยูนิคคอมมิตเมนต์

ผู้จัดทำ

นาย พลพิพัฒน์ อมรธนาบาล	45010552
นาย ชัยวัฒน์ มณีกรณี	46010164
นาย ปกวงศ์ สมิตพงษ์	46010397

  
.....อาจารย์ที่ปรึกษา  
( ดร. ชัย ชมภูอินไหว )

  
.....อาจารย์ที่ปรึกษา  
( ร.ศ. มณฑล สีลาจินดาไกรฤกษ์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การวางแผนการผลิตพลังงานไฟฟ้าระยะสั้นโดยข้อกำหนดของยูนิตคอมมิตเมนต์

นาย พลพิพัฒน์ อมรรณานุกูล

นาย ชัยวัฒน์ มณีกรณ์

นาย ปกวงศ์ สมิตพงษ์

ดร. ชาย ชมภูอินไหว อาจารย์ที่ปรึกษา

ร.ศ. มณฑล ลีลาจินดาไกรฤกษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2549

### บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้นำเสนอ การผลิตกำลังไฟฟ้าให้เพียงพอกับความต้องการของโหลดที่เปลี่ยนแปลงตลอดเวลา และยังทำให้ระบบยังคงมีเสถียรภาพโดยที่ต้นทุนการผลิตไฟฟ้ารวม มีค่าต่ำสุด กระบวนการนี้เรียกว่า ยูนิตคอมมิตเมนต์ ( Unit Commitment ) ซึ่งมีขั้นตอนการจ่ายโหลดอย่างประหยัด ( Economic Dispatch ) รวมอยู่ด้วยทำหน้าที่คำนวณขนาดกำลังไฟฟ้าที่ต้องผลิต จากเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่อยู่ต่อกับระบบไฟฟ้าให้สามารถจ่ายกำลังไฟฟ้าได้เพียงพอกับความต้องการของโหลดรวม และต้นทุนเชื้อเพลิงรวมในการผลิตไฟฟ้ามีค่าต่ำที่สุด ตามวิธีโปรแกรมพลวัต (Dynamic programming) เพื่อพัฒนากระบวนการที่จะใช้เป็นแนวทางในการเสนอราคาและกำลังการผลิตของผู้ผลิตในตลาดไฟฟ้าเพื่อให้ได้กำไรสูงสุด โดยวิธีนี้จะแบ่งการทำยูนิตคอมมิตเมนต์ออกเป็นช่วง ๆ แล้วหาคำตอบที่เหมาะสม โดยคำนึงถึงการเปลี่ยนสถานะ (State) และข้อจำกัดต่างๆ ในแต่ละเวลาด้วย สำหรับระบบไฟฟ้าที่ประกอบด้วยเครื่องกำเนิดไฟฟ้าพลังงานความร้อน ,พลังงานความร้อนร่วม,และเครื่องกำเนิดไฟฟ้าพลังน้ำ ซึ่งหลักการทำงานและเงื่อนไขในการออกแบบถูกอธิบายดังในรายละเอียดของปฏิญานิพนธ์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Security Constraint Unit Commitment for a Day Ahead  
Generation Planning**

Polpipat Amornatanuban

Chaiwat Maneekorn

Pogvong Smithpong

Dr.chai chompoonwai Advisor

Assoc.Prof.Monthon Leelachindakaileak Advisor

2006

**Abstract**

This thesis presents electric power production that sufficient for total demand load which always change and it also make system stability by the budget of total amount electric production is lowest. This process is called Unit commitment which including economic dispatch. The economic dispatch will calculate electric power size which is produced from each generator that close to electric system to be able to contribute enough electric power for the total load and the lowest total fuel budget in electrical production. According to dynamic programming method for developing the process, which will be used as price offering approach and electrical power of the producer in electrical market in order to gain the maximum profit. Besides, unit commitment is divided by stage. Then, find the appropriate answer by considering the state and other restrictions in each period too. For the electrical system, which consist of thermal electric generator, combined cycle, hydro electric generator. The principle and condition of their designs were explained as the following of this thesis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ผู้จัดทำโครงการนุศึกษาซึ่งต่อความกรุณาที่ ดร. ชาย ชมภูอินไหว และ ร.ศ. มณฑล ลีลาจินดาไกรฤกษ์ อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำข้อคิดเห็นต่างๆ อันเป็นประโยชน์อย่างยิ่งในการวิจัย ตลอดจนอาจารย์ในภาควิชาที่ประสิทธิประสาทความรู้ให้กับผู้จัดทำในครั้งนี้ ถ้าปราศจากท่านโครงการนี้ก็คงไม่สำเร็จลุล่วงไปได้ด้วยดี

นอกจากนั้นผู้จัดทำต้องขอขอบคุณ อาจารย์ เต๋า ที่ให้ความรู้ทางด้านโปรแกรมที่ใช้ในการทำโครงการนี้ และขอขอบคุณห้องปฏิบัติการทางแสง ภาควิชาวิศวกรรมไฟฟ้ากำลัง คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เอื้อเฟื้อสถานที่ในการศึกษาวิจัยครั้งนี้

ท้ายที่สุดนี้ ผู้จัดทำต้องขอกราบขอบพระคุณบิดา มารดา ที่คอยให้กำลังใจและส่งเสริมเงินทอง ให้ผู้จัดทำเรียนหนังสือจนสำเร็จการศึกษาในครั้งนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
สารบัญรูป.....	III
บทนำ.....	1
ความสำคัญของปัญหา.....	2
วัตถุประสงค์ของการวิจัย.....	2
ขอบเขตและข้อกำหนดในการวิจัย.....	3
ขั้นตอนและวิธีการดำเนินงาน.....	3
ประโยชน์ที่ได้รับจากการวิจัย.....	3
บทที่ 1 ทฤษฎีที่เกี่ยวข้อง.....	4
1.1 ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้า.....	4
1.2 ปัญหาการจ่ายไหลต่ออย่างประหยัด (Economic dispatch).....	11
บทที่ 2 หลักการและปัญหาในการทำยูนิตคอมมิตเมนต์.....	17
2.1 การวางแผนการผลิตไฟฟ้า.....	20
2.2 ประเภทของโรงไฟฟ้า.....	22
2.3 การทำยูนิตคอมมิตเมนต์ด้วยวิธีต่าง ๆ และเอกสารที่เกี่ยวข้อง.....	24
บทที่ 3 การทำยูนิตคอมมิตเมนต์.....	31
3.1 กำหนดสถานะการเดินเครื่องกำเนิดไฟฟ้าทั้งหมดที่เป็นไปได้.....	31
3.2 การหาค่าเหมาะสมในการเสนอราคาและกำลังไฟฟ้าของแต่ละสถานะ การทำงานและผลกำไรที่ได้ในแต่ละชั่วโมง.....	32
3.3 ใช้โปรแกรมพลวัต เพื่อหาเส้นทางที่ทำให้ได้ผลประโยชน์โดยรวมแต่ละวันสูงสุด.....	32
3.4 ปัญหายูนิตคอมมิตเมนต์.....	36
3.5 รูปแบบทางคณิตศาสตร์ของปัญหายูนิตคอมมิตเมนต์.....	36
บทที่ 4 ผลการทดลอง.....	42
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	55
เอกสารอ้างอิง.....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
1.1 ตัวอย่างลักษณะ load curve	6
1.2 กราฟแสดงความสัมพันธ์ cooling-banking	9
1.3 กราฟแสดงความสัมพันธ์ระหว่าง Heat-rate curve กับ fuel-cost curve	13
1.4 แผนภาพแสดงหลักการทำงานของ การจ่ายโหลดอย่างประหยัด	16
2.1 กราฟแสดงความต้องการไฟฟ้าในแต่ละวัน	21
2.2 กราฟแสดงความสัมพันธ์ระหว่างราคาการผลิตกับหน่วยการผลิต	23
3.1 รูปแสดง system chart of thermal unit-commitment method	34
3.2 structure of thermal unit-commitment system module	35
3.3 เส้นทางที่เป็นไปได้ในการคำนวณโดยใช้วิธีโปรแกรมเชิงพลวัต	35
3.4 แผนภาพแสดงหลักการทำงานของ Unit commitment	37
3.5 แสดงเส้นทางเดินของ unit-commitment โดยใช้วิธีโปรแกรมเชิงพลวัต	41
3.6 กราฟแสดงการเปรียบเทียบเวลา	58
ภาคผนวก ก	
ภาคผนวก ข	
ภาคผนวก ค	
กิตติกรรมประกาศ	



**Cleaning Sheet**

**Reinigungsblatt**

**Feuille de nettoyage**

**Foglio di pulizia**

**Página de limpieza**

**Página da limpeza**

**Schoonmaakvel**

**Страница очистки**

**清潔紙**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทนำ

ปัจจุบันตลาดการค้ากำลังไฟฟ้ามีการแข่งขันมากขึ้น ผู้จัดจำหน่ายกำลังไฟฟ้าจำเป็นต้องคำนึงถึงหลักเศรษฐศาสตร์ในการวางแผนการผลิตกำลังไฟฟ้า ไม่ใช่เพียงคำนึงแค่ว่าจะผลิตกำลังไฟฟ้าให้เพียงพอกับความต้องการของโหลดอย่างเดียว แต่ยังคงคำนึงถึงค่าใช้จ่ายที่จะต้องสูญเสียไปกับการผลิตกำลังไฟฟ้าด้วย อันเนื่องมาจากคุณลักษณะเฉพาะของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่ใช้เชื้อเพลิงต่างชนิดกัน

แผนการผลิตกำลังไฟฟ้าที่คำนึงถึงหลักเศรษฐศาสตร์ มุ่งเน้นการจัดการสถานการณ์การทำงานของเครื่องกำเนิดไฟฟ้าที่มีอยู่ในระบบทั้งหมด ว่าเครื่องกำเนิดไฟฟ้าเครื่องใดควรทำงานต่ออยู่กับระบบไฟฟ้าหรือปลดออกจากระบบ เพื่อทำการผลิตกำลังไฟฟ้าให้เพียงพอกับความต้องการของโหลดที่เปลี่ยนแปลงตลอดเวลา และยังทำให้ระบบยังคงมีเสถียรภาพโดยที่ต้นทุนการผลิตไฟฟ้ารวม มีค่าต่ำสุด กระบวนการนี้เรียกว่า ยูนิตคอมมิทเมนต์ ( Unit Commitment ) ซึ่งมีขั้นตอนการจ่ายโหลดอย่างประหยัด ( Economic Dispatch ) รวมอยู่ด้วยทำหน้าที่คำนวณขนาดกำลังไฟฟ้าที่ต้องผลิต จากเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่อยู่ต่อกับระบบไฟฟ้าให้สามารถจ่ายกำลังไฟฟ้าได้เพียงพอกับความต้องการของโหลดรวมทั้งกำลังสูญเสียในระบบส่ง ณ. ช่วงเวลานั้น และต้นทุนเชื้อเพลิงรวมในการผลิตไฟฟ้ามีค่าต่ำที่สุด

โดย ยูนิตคอมมิทเมนต์ ( Unit Commitment ) ต้องพิจารณาเงื่อนไขทั้งของเครื่องกำเนิดไฟฟ้า และระบบไฟฟ้า อันได้แก่ ขีดจำกัดการผลิต, กำลังไฟฟ้าสมมูล, เวลาเดินเครื่องกำเนิดไฟฟ้าอย่างน้อยที่สุด, เวลาหยุดเดินเครื่องกำเนิดไฟฟ้าอย่างน้อยที่สุด, กำลังไฟฟ้าสำรอง, รวมไปถึงกำลังไฟฟ้าสูญเสียอันเนื่องมาจากสายส่ง โดยปกติจะทำการวางแผนล่วงหน้า 1 วัน โดยในแต่ละวันจะพิจารณาการผลิตกำลังไฟฟ้าออกเป็น 24 คาบ ซึ่งเป็นลักษณะการวางแผนการผลิตกำลังไฟฟ้าระยะสั้น ( Short Term Unit Commitment ) โดยอาศัยข้อมูลการทำนายความต้องการพลังงานระยะสั้น

## ความสำคัญของปัญหา

ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้า ( Unit Commitment ) เป็นการหาสถานการณ์ปฏิบัติงานของเครื่องกำเนิดไฟฟ้าในระบบ ซึ่งเป็นการคำนวณหาว่าเครื่องใดควรจะเดินหรือควรจะหยุดเดินเพื่อที่จะให้ได้ค่าใช้จ่ายที่ประหยัดที่สุดในทางปฏิบัติแล้วบริษัทผลิตกระแสไฟฟ้าจะมีเครื่องกำเนิดไฟฟ้าอยู่หลายเครื่อง แต่ละเครื่องจะมีกำลังในการผลิต ต้นทุนในการผลิต หรือค่าบำรุงรักษา ที่แตกต่างกัน แต่เนื่องจากโหลดความต้องการไฟฟ้าในแต่ละชั่วโมงจะแตกต่างกัน ดังนั้นจึงไม่ใช่เป็นเรื่องง่ายที่จะเลือกเปิดหรือปิด เครื่องกำเนิดไฟฟ้าชนิดต่างๆ เพื่อให้ค่าใช้จ่ายที่ต่ำที่สุด หากไม่มีการคำนวณที่ถูกต้องในปัจจุบัน บริษัทไฟฟ้าต่างๆ ในโลกยังมีไม่ได้ใช้วิธีการที่อาศัยหลักทางคณิตศาสตร์ที่เหมาะสมสำหรับการเลือกเปิดหรือปิดเครื่องกำเนิดไฟฟ้า แต่จะอาศัยประสบการณ์เสียมากกว่า ซึ่งหากสามารถที่จะนำวิธีการคำนวณที่เหมาะสมมาใช้ อาจจะทำให้บริษัทไฟฟ้าสามารถประหยัดได้เป็นเงินนับล้านบาทต่อวัน

ต้นทุนในการผลิตไฟฟ้าต่อหน่วยสำหรับเครื่องกำเนิดไฟฟ้าพลังงานความร้อน จะมีต้นทุนสูงกว่าเครื่องกำเนิดไฟฟ้าพลังงานน้ำและพลังงานความร้อนร่วม ถ้าต้องจ่ายโหลดโดยกลุ่มเครื่องกำเนิดไฟฟ้าพลังงานความร้อนจะต้องมีการพิจารณาอย่างถี่ถ้วน เพื่อไม่ให้ทำการผลิตไฟฟ้าออกมามากกว่าความต้องการพลังงานไฟฟ้า (demand) กล่าวคือทำอะไรจึงทำให้กลุ่มเครื่องกำเนิดไฟฟ้าพลังงานความร้อนทำการผลิตกำลังออกมาให้คุ้มค่า นั่นคือทำการเดินเครื่องกำเนิดไฟฟ้าพลังงานความร้อนเท่าที่จำเป็น ดังนั้นปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าโดยมากจะพิจารณากับเครื่องกำเนิดไฟฟ้าพลังงานความร้อน

## วัตถุประสงค์

1. ศึกษาการแก้ปัญหาชนิดคอมมิตเมนต์ ตามวิธีโปรแกรมพลวัต (Dynamic programming) สำหรับระบบไฟฟ้าที่ประกอบด้วยเครื่องกำเนิดไฟฟ้าพลังงานความร้อน ,พลังงานความร้อนร่วม,และเครื่องกำเนิดไฟฟ้าพลังน้ำ
2. เพื่อพัฒนากระบวนการวิธีที่จะใช้เป็นแนวทางในการเสนอราคาและกำลังการผลิตของผู้ผลิตในตลาดไฟฟ้าเพื่อให้ได้กำไรสูงสุด
3. เพื่อออกแบบและพัฒนาโปรแกรมคอมพิวเตอร์ที่สามารถคำนวณค่าเหมาะสมของราคาและกำลังการผลิตของบริษัทไฟฟ้าตามการแก้ปัญหาชนิดคอมมิตเมนต์โดยวิธีพลวัตพร้อมทั้งทดสอบประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ขอบเขตและข้อกำหนด

1. เพื่อทดสอบการแก้ปัญหาการจ่ายไฟฟ้าของเครื่องกำเนิดไฟฟ้าไม่เกิน 100 เครื่องในเวลา 24 ชั่วโมง
2. ไม่พิจารณาเงื่อนไขดังต่อไปนี้ กำลังคน สิ่งแวดล้อม ขีดจำกัดการส่งกำลังของสายส่ง (Transmission line capacity) และ การผลิตพลังงานสำรอง (Spinning reserve)
3. ค่าใช้จ่ายในการผลิตไฟฟ้าพิจารณาเฉพาะค่าเชื้อเพลิงในการผลิตเท่านั้น โดยไม่มีเงื่อนไขขีดจำกัดของเชื้อเพลิง
4. เขียนโปรแกรมคอมพิวเตอร์แก้ปัญหาชนิดคอมพิวติเมนต์ด้วยโปรแกรม Visual Basic 6 และโปรแกรมฐานข้อมูล Microsoft Office Access

## ขั้นตอนการทำงานและวิธีการดำเนินงาน

1. ศึกษาวิธีการแก้ปัญหาชนิดคอมพิวติเมนต์และการแก้ปัญหาการจ่ายโหลดอย่างประหยัด โดยค้นคว้าจากตำราและบทความทางวิชาการต่างๆที่เกี่ยวข้อง
2. ศึกษาการเขียนโปรแกรมด้วย Visual Basic 6
3. พัฒนาโปรแกรมการแก้ปัญหาชนิดคอมพิวติเมนต์ด้วยวิธีพลวัต (Dynamic programming) ที่รวมการแก้ปัญหาการจ่ายโหลดอย่างประหยัด (Economic Dispatch)
4. ทดสอบความสามารถและประสิทธิภาพของโปรแกรมและนำมาสรุปผลงานวิจัย

## ประโยชน์ที่คาดว่าจะได้รับ

1. สร้างโปรแกรมคอมพิวเตอร์ที่สามารถแก้ปัญหาชนิดคอมพิวติเมนต์ด้วยวิธีพลวัต (Dynamic-programming) และแก้ปัญหาการจ่ายโหลดอย่างประหยัด
2. ผลงานสามารถใช้เป็นแนวทางในการวางแผนการผลิตกำลังไฟฟ้าในเชิงปฏิบัติของการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าสำหรับโรงไฟฟ้าในประเทศไทย ซึ่งในปัจจุบันอัตราความต้องการการใช้พลังงานไฟฟ้าของประเทศมีค่อนข้างสูง และมีโรงผลิตไฟฟ้าอยู่หลายประเภทซึ่งหากมีการนำวิธีการทางคณิตศาสตร์ที่เหมาะสมสำหรับการเลือกเปิดหรือปิดเครื่องกำเนิดไฟฟ้าชนิดต่างๆ ก็ให้นำมาสู่การประหยัดค่าใช้จ่ายในการผลิตไฟฟ้าให้กับประเทศได้เป็นจำนวนมาก
3. สามารถนำโปรแกรมไปประยุกต์ใช้งานในโรงงานอุตสาหกรรม และ โรงงานผลิตไฟฟ้ารายย่อยเพื่อประหยัดค่าใช้จ่าย
4. ช่วยลดต้นทุนในการนำเข้าเชื้อเพลิงจากต่างประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้า

ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าเป็นปัญหาของการจัดการวางแผนการผลิตกำลังไฟฟ้าของเครื่องกำเนิดไฟฟ้าที่อยู่ในระบบฟีดกำลัง ซึ่งต้องขึ้นอยู่กับตัวเครื่องกำเนิดไฟฟ้า ทำให้มีเงื่อนไข (constraints) ทั้งจากระบบ และจากเครื่องกำเนิดไฟฟ้าเข้ามาเกี่ยวข้อง อาทิเช่นความสมดุลทางไฟฟ้า(power balance) กำลังผลิตสำรอง(spinning reserve) ขีดจำกัดกำลังผลิต(generating limits) และเวลาเดิน-หยุดเครื่องอย่างน้อยที่สุด(minimum up and down time) เป็นต้น นอกจากนี้การวิเคราะห์แก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้านั้น ต้องมีส่วนการวิเคราะห์ของการจ่ายไหลอย่างประหยัด (economic dispatch) รวมอยู่ด้วย กล่าวคือต้องแก้ปัญหาการจ่ายไหลอย่างประหยัดสำหรับเครื่องกำเนิดไฟฟ้าที่เปิดอยู่ เพื่อคำนวณการจ่ายปริมาณพลังงานไฟฟ้าที่จะต้องจ่ายในแต่ละเครื่องกำเนิดไฟฟ้าที่เปิดอยู่เพื่อทำให้ต้นทุนต่ำสุด

เป้าหมายของการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าคือต้องการที่จะตัดสินใจเลือกจ่ายเปิดหรือปิดเครื่องกำเนิดไฟฟ้า ในช่วงเวลาใดเวลาหนึ่งเพื่อให้มีต้นทุนที่ต่ำที่สุด ในกรณีของเครื่องกำเนิดไฟฟ้าพลังความร้อนราคาต้นทุนของการเดินเครื่องกำเนิดไฟฟ้าขึ้นอยู่กับราคาเชื้อเพลิง (fuel cost) และราคาการเดินเครื่อง (startup cost) หากพิจารณาเครื่องกำเนิดไฟฟ้า  $N$  เครื่องในระยะเวลา  $T$  ชั่วโมง เราสามารถคำนวณหาต้นทุนทั้งหมดได้ดังสมการต่อไปนี้

$$F = \sum_{t=1}^T \sum_{i=1}^N U_{it} F_{it}(P_{it}) + U_{it}(1 - U_{it-1}) ST_i \quad (1.1)$$

เมื่อ  $U_{it}$  คือสถานะของเครื่องกำเนิดไฟฟ้าของเครื่องกำเนิดไฟฟ้าที่  $i$  และที่เวลา  $t$  ซึ่งถ้ามีค่าเป็น 0 คือหยุดเดินเครื่อง และถ้ามีค่าเป็น 1 คือเดินเครื่อง

$P_{it}$  คือกำลังผลิตไฟฟ้าที่ผลิตจากเครื่องกำเนิดไฟฟ้าที่  $i$  และที่เวลา  $t$

$F_{it}(P_{it})$  คือค่าเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าเครื่องกำเนิดไฟฟ้าที่  $i$  ขณะจ่ายกำลังไฟฟ้า  $P_{it}$  ที่เวลา  $t$

$ST_i$  คือราคาการเริ่มเดินเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$

การแก้ปัญหาการจ่ายกระแสไฟฟ้าคือทำให้ค่าที่ได้จากสมการที่ 1 มีค่าต่ำสุด ส่วนแรก  
ของสมการเป็นราคาค่าเชื้อเพลิงที่ได้จากการแก้ปัญหาการจ่ายโหลดอย่างประหยัด กล่าวนำ  
เครื่องกำเนิดไฟฟ้าที่เปิดอยู่ ( $U_{it} = 1$ ) ไปคำนวณการจ่ายโหลดอย่างประหยัด ซึ่งการแก้ปัญหา  
การจ่ายโหลดอย่างประหยัดต้องมีฟังก์ชันราคาค่าเชื้อเพลิงเข้ามาเกี่ยวข้อง ส่วนที่สองของ  
สมการ เป็นราคาการเริ่มเดินเครื่องกำเนิดไฟฟ้า ซึ่งราคาการเริ่มเดินเครื่องจะมีขึ้นเมื่อ  $U_i(t) =$   
 $1$  และ  $U_i(t-1) = 0$  กล่าวคือการเริ่มเดินเครื่องกำเนิดไฟฟ้านั้นจะมีราคาการเริ่มเดินเครื่องกำเนิด  
ไฟฟ้าอยู่ การเริ่มเดินเครื่องกำเนิดไฟฟ้าที่ไม่เหมาะสมอาจทำให้ต้นทุนการผลิตพลังงานไฟฟ้ามี  
ค่าสูงขึ้นกล่าวคือสมการที่ 1.1 มีค่าสูงขึ้น

สำหรับฟังก์ชันราคาค่าเชื้อเพลิงจากสมการที่ 1 ที่มีผลต่อต้นทุนการผลิตพลังงานไฟฟ้า  
นั้นมีรายละเอียดตามสมการที่ 1.2

$$F_i(P_{it}) = a_i + b_i P_{it} + c_i P_{it}^2 \quad (1.2)$$

เมื่อ  $a$ ,  $b$  และ  $c$  คือค่าคงที่ของแต่ละเครื่องกำเนิดไฟฟ้าที่  $i$

โดยสมการที่ 2 นั้นเกิดจากการออกแบบสร้างเครื่องกำเนิดไฟฟ้า หรือจากการทดสอบ  
ทำให้บางครั้งสมการที่ 2 ถูกเรียกว่ากราฟอินพุตเอาต์พุต (input output curve) ซึ่งเป็น  
ความสัมพันธ์ระหว่างค่ากำลังไฟฟ้าที่ผลิตจากเครื่องกำเนิดไฟฟ้าที่  $i$  ที่เวลา  $t$  กับค่าเชื้อเพลิงที่  
ใส่เข้าไปสำหรับเครื่องกำเนิดไฟฟ้างกล่าว โดยปกติฟังก์ชันราคาค่าเชื้อเพลิงเป็นลักษณะ  
สมการควอดราติก โดยค่าคงที่ต่างๆของเครื่องกำเนิดไฟฟ้าจะได้อมาจากการทดสอบ

ในส่วนของราคาการเริ่มเดินเครื่องกำเนิดไฟฟ้าจากสมการที่ 1 ที่มีผลต่อต้นทุนการผลิต  
พลังงานไฟฟ้านั้นมีรายละเอียดตามสมการที่ 1.3

$$ST_i = d_{1i}(1 - \exp(-k / d_{3i})) + d_{2i} \quad (1.3)$$

เมื่อ  $k$  คือจำนวนชั่วโมงที่เครื่องกำเนิดไฟฟ้าหยุดเดินเครื่องก่อนที่จะเริ่มเดินเครื่องใหม่  
 $d_{1i}, d_{2i}$  และ  $d_{3i}$  คือค่าคงที่ของแต่ละเครื่องกำเนิดไฟฟ้า

ราคาของการเริ่มเดินเครื่องแสดงเป็นความสัมพันธ์ของจำนวนชั่วโมง ( $k$ ) ที่เครื่อง  
กำเนิดไฟฟ้าหยุดเดินเครื่องก่อนที่จะเริ่มเดินเครื่องใหม่ นอกจากนี้ยังมีค่าคงที่  $d_{1i}, d_{2i}$  และ  $d_{3i}$   
ซึ่งเป็นค่าคงที่ที่มีค่าแตกต่างกันไปขึ้นอยู่กับแต่ละเครื่องกำเนิดไฟฟ้า ได้จากการทดสอบให้  
เครื่องกำเนิดไฟฟ้าเดินเครื่องจ่ายกำลังไฟฟ้าแตกต่างกัน หรือบริษัทผู้ผลิตเครื่องกำเนิดไฟฟ้า  
เป็นผู้กำหนด

ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้ายังมีเงื่อนไขบังคับอื่นที่ต้องพิจารณาจำนวนมากพอสรุปได้ดังต่อไปนี้

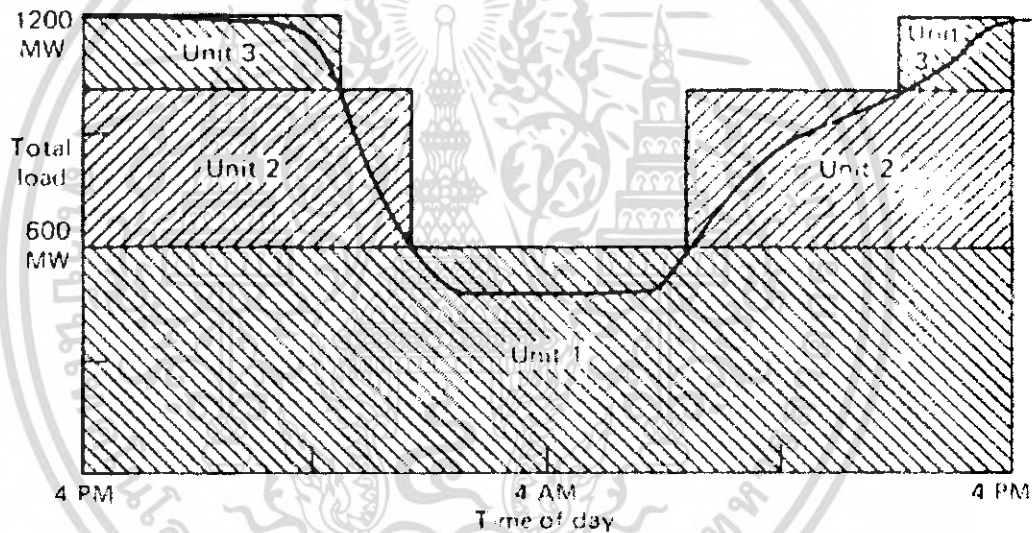
### 1. เงื่อนไขความสมดุลกำลังไฟฟ้า (Power balance)

ความต้องการกำลังไฟฟ้าของระบบในแต่ละชั่วโมง และผลรวมกำลังผลิตของแต่ละเครื่องกำเนิดไฟฟ้าต้องมีความสมดุลกันเพื่อให้ผู้ใช้กระแสไฟฟ้าที่เพียงพอ สามารถแสดงได้ดังสมการที่ 1.4

$$\sum_{t=1}^T \sum_{i=1}^N U_{it} P_{it} = D_t \quad (1.4)$$

เมื่อ  $D_t$  คือความต้องการกำลังไฟฟ้าของระบบที่เวลา  $t$

$N$  คือ จำนวนของโรงจักรไฟฟ้า



รูปที่ 1.1 ตัวอย่างลักษณะ load curve

### 2. เงื่อนไขกำลังผลิตไฟฟ้าสำรอง (Spining reserve)

กำลังผลิตสำรองเป็นการอธิบายถึงความสามารถของเครื่องกำเนิดไฟฟ้าที่ชานกันอยู่ในระบบไฟฟ้ากำลังเพื่อจ่ายกำลังไฟฟ้าให้กับระบบของเครื่องที่เดินอยู่ว่าจะสามารถจ่ายกำลังไฟฟ้าที่ต้องการสำรองไว้ โดยปกติการผลิตกำลังสำรองแบบหนึ่งที่นิยมใช้ในการคำนวณ คือประมาณจากเปอร์เซ็นต์ของความต้องการกำลังไฟฟ้าสูงสุดในเวลานั้น ตัวอย่างเช่นความต้องการกำลังไฟฟ้าเป็น 700 MW ถ้าต้องการผลิตกำลังสำรอง 10 เปอร์เซ็นต์ ดังนั้นกลุ่มของเครื่องกำเนิดไฟฟ้าต้องสามารถที่จะจ่ายกำลังไฟฟ้าออกมาได้เป็น 770 MW เงื่อนไขกำลังผลิตไฟฟ้าสำรองจะเป็นไปตามสมการที่ 1.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sum_{t=1}^T \sum_{i=1}^N U_{it} P_{i, \max} \geq R_t + D_t \quad (1.5)$$

เมื่อ  $R_t$  คือกำลังผลิตไฟฟ้าสำรองที่เวลา  $t$

$P_{i, \max}$  คือกำลังผลิตไฟฟ้าสูงสุดของเครื่องกำเนิดไฟฟ้าที่  $i$

$T$  คือ ขอบเขตเวลาที่พิจารณา(24hr)

โดย  $1 \leq t \leq T$

### 3. เงื่อนไขขีดจำกัดกำลังผลิต (Generating capacity)

เครื่องกำเนิดไฟฟ้าพลังงานความร้อนแต่ละเครื่องมีขีดจำกัดกำลังผลิตที่แตกต่างกันออกไป จึงทำให้เกิดปัญหาเพิ่มขึ้นต่อการเดินเครื่องหรือหยุดเครื่องเพื่อให้เพียงพอต่อความต้องการกำลังไฟฟ้า ปกติแล้วขีดจำกัดกำลังผลิตเป็นไปตามสมการที่ 1.6

$$P_{i, \min} \leq P_{it} \leq P_{i, \max} \quad (1.6)$$

เมื่อ  $P_{i, \min}$  คือขีดจำกัดกำลังผลิตไฟฟ้าต่ำสุดของเครื่องกำเนิดไฟฟ้าที่  $i$

### 4. ข้อบังคับของหน่วยเครื่องกำเนิดไฟฟ้าปริมาณความร้อน (Thermal unit constraints)

หน่วยปริมาณความร้อนต้องการเจ้าหน้าที่ในการควบคุมดูแล โดยเฉพาะอย่างยิ่งเมื่อทำการเปิดและปิด นอกจากนี้เครื่องกำเนิดไฟฟ้าปริมาณความร้อนสามารถทนได้เฉพาะการเปลี่ยนแปลงอุณหภูมิที่เป็นปกติ และการเปลี่ยนแปลงภายในระยะเวลาหนึ่งของบางชั่วโมงที่ต้องการจะนำเครื่องกำเนิดไฟฟ้าเข้าสู่ระบบ ทำให้มีข้อบังคับมากมายได้แก่:

#### 4.1 เงื่อนไขเวลาเดินเครื่องอย่างน้อยที่สุด (minimum up time)

เครื่องกำเนิดไฟฟ้าแต่ละเครื่องอาจมีเวลาเดินเครื่องอย่างน้อยที่สุดแตกต่างกันออกไป กล่าวคือเครื่องกำเนิดไฟฟ้าที่เดินเครื่องอยู่นั้นไม่สามารถที่จะหยุดเดินเครื่องอย่างทันทีทันใดได้ ต้องอาศัยเวลาอยู่ระยะหนึ่ง ปกติแล้วมีหน่วยเป็นชั่วโมง ดังนั้นเงื่อนไขจึงเป็นไปตามสมการที่ 1.7

$$(X_{i(t-1)}^{on} - \min\_up_i) * (U_{i(t-1)} - U_{it}) \geq 0 \quad (1.7)$$

เมื่อ  $X_{i(t-1)}^{on}$  คือจำนวนชั่วโมงที่เครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  เดินเครื่องเป็นเวลาติดต่อกันจนถึงเวลาที่  $(t-1)$

$\min\_up_i$  คือค่าเวลาเดินเครื่องอย่างน้อยที่สุดของเครื่องกำเนิดไฟฟ้าที่  $i$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการวิจัยเท่านั้น เมื่อผู้ผู้เห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 เงื่อนไขเวลาหยุดเครื่องอย่างน้อยที่สุด (minimum down time)

มีลักษณะคล้ายกับเวลาเดินเครื่องอย่างน้อยที่สุด เพียงแต่แตกต่างที่เครื่องกำเนิดไฟฟ้าแต่ละตัวอาจมีค่าเวลาหยุดเครื่องอย่างน้อยที่สุดแตกต่างกัน กล่าวคือเครื่องกำเนิดไฟฟ้าแต่ละเครื่องไม่สามารถเดินเครื่องอย่างทันทีทันใดต้องมีเวลาอย่างน้อยอยู่ค่าหนึ่ง ซึ่งมีหน่วยเป็นชั่วโมง ดังนั้นเงื่อนไขจึงเป็นไปตามสมการที่ 1.8

$$(X_{i(t-1)}^{off} - \min\_dwi) * (U_{i(t-1)} - U_{it}) \geq 0 \quad (1.8)$$

เมื่อ  $X_{i(t-1)}^{off}$  คือจำนวนชั่วโมงที่เครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  หยุดเดินเครื่องเป็นเวลาติดต่อกันจนถึงเวลาที่  $(t-1)$

$\min\_dwi$  คือค่าเวลาหยุดเครื่องอย่างน้อยที่สุดของเครื่องกำเนิดไฟฟ้าที่  $i$

#### 4.3 เงื่อนไขด้านแรงงาน(crew constraint)

ในระบบโรงจักรใหญ่จะต้องมีการใช้กำลังคนที่เพียงพอในการเริ่มต้นเครื่องกำเนิดไฟฟ้า ถ้าเครื่องจักรประกอบไปด้วย สองหน่วยหรือมากกว่านั้น เราไม่สามารถเปิดในเวลาเดียวกันได้ ทำให้เสียค่าใช้จ่ายในการจ้างแรงงานซึ่งนำมาคิดรวมในการคิดค่า Start up cost

ยิ่งไปกว่านั้นเป็นเพราะว่าอุณหภูมิและแรงกดดันของเครื่องกำเนิดไฟฟ้าปริมาณความร้อน ที่มีการเปลี่ยนแปลงอย่างช้าๆจึงต้องคำนึงถึงค่าคูลลิ่ง(cooling) หรือ แบงก์กิ้ง (banking) ด้วยนำไปสู่ปัญหาชนิดคอมมิตเมนต์ (unit commitment) เช่นเดียวกับราคาการเริ่มต้นเครื่องกำเนิดไฟฟ้า (start-up cost)

ถ้าเครื่องกำเนิดไฟฟ้าเพิ่งถูกปิดไปได้ไม่นานและยังมีความใกล้เคียงกับอุณหภูมิปกติ วิธีการจัดการกับเครื่องกำเนิดไฟฟ้าปริมาณความร้อน ในช่วงเวลาดังกล่าว มี 2 วิธี วิธีแรกคือ แบบคูลลิ่ง (cooling) คือการทำให้หม้อต้ม (boiler) เย็นลงก่อนและค่อยทำให้ร้อนในเวลาต่อมา จะเหมาะกับกรณีที่ เครื่องกำเนิดไฟฟ้านั้นหยุดนานกว่าจุดคุ้มทุน ส่วนวิธีที่สองที่เรียกว่า แบบแบงก์กิ้ง(banking) คือการให้พลังงานที่เพียงพอใส่เข้าไปในหม้อต้ม (boiler) เพื่อรักษาอุณหภูมิให้คงที่ เราสามารถเปรียบเทียบค่าใช้จ่ายของสองวิธีนี้ (cooling or banking) ได้ เพื่อที่จะหาวิธีที่ดีที่สุดดังนี้เป็นไปตามสมการที่ (1.9) และ (1.10)

ราคาการเริ่มต้นเครื่องกำเนิดไฟฟ้าเมื่อคำนวณแบบคูลลิ่ง (cooling) =

$$C_c (1 - \varepsilon^{-t/\alpha}) * F + C_f \quad (1.9)$$

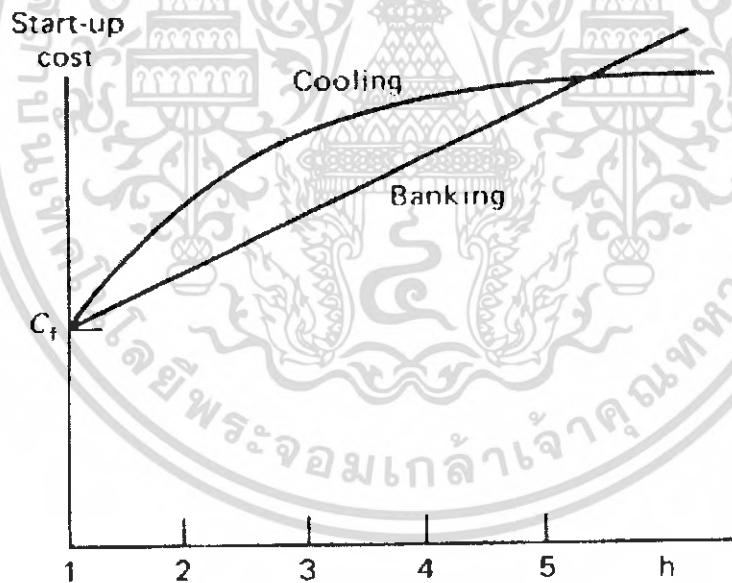
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $C_i$  = ราคาการเริ่มต้นเครื่องกำเนิดไฟฟ้าขณะเครื่องเย็น(MBtu)
- $F$  = ราคาเชื้อเพลิง
- $C_f$  = ราคาค่าใช้จ่ายอื่น(รวมถึงค่าใช้จ่ายคนงานและค่าใช้จ่ายในการซ่อมบำรุง)
- $\alpha$  = ค่าคงที่ของเครื่องกำเนิดไฟฟ้าแต่ละยูนิต
- $t$  = ระยะเวลาที่พิจารณา

ราคาการเริ่มต้นเครื่องกำเนิดไฟฟ้าเมื่อคำนวณแบบแบงก์กิ้ง(banking) =

$$C_i * t * F + C_f \tag{1.10}$$

$C_i$  = ราคาค่าใช้จ่ายในการรักษาอุณหภูมิให้คงที่  
 ทั้งนี้ตัวกำหนดประสิทธิภาพของหน่วยปริมาณความร้อนที่อาจจะเปลี่ยนแปลงอยู่บ่อยๆ  
 เนื่องจากการบำรุงรักษา หรือ การหยุดชะงักที่ไม่เป็นไปตามตารางของอุปกรณ์ต่างๆใน  
 โรงงาน นี้ก็ถือว่าเป็นปัญหา unit commitment



รูปที่ 1.2 กราฟแสดงความสัมพันธ์ cooling-banking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5. กฎข้อบังคับอื่น ๆ (Other constraint)

### 5.1 เงื่อนไขขีดจำกัดการเปลี่ยนแปลงกำลังผลิต (ramp-rare limit)

ในขณะที่เครื่องกำเนิดไฟฟ้าเดินเครื่อง และจ่ายกำลังไฟฟ้าออกไปค่าหนึ่ง และถ้าเครื่องกำเนิดไฟฟ้าต้องมีการเปลี่ยนแปลงกำลังการผลิต โดยปกติแล้วเครื่องกำเนิดไฟฟ้าพลังความร้อนอาจต้องมีการพิจารณาเงื่อนไขนี้ด้วย เนื่องจากเครื่องกำเนิดไฟฟ้าพลังความร้อนไม่สามารถเปลี่ยนแปลงกำลังการผลิตได้อย่างทันทีทันใด ดังนั้นการเปลี่ยนแปลงกำลังการผลิตต้องเป็นไปตามสมการที่ (1.11) และ (1.12)

$$P_{it} - P_{i(t-1)} \leq UR_i \quad (1.11)$$

$$P_{i(t-1)} - P_{it} \leq DR_i \quad (1.12)$$

เมื่อ  $UR_i$  คือขีดจำกัดการเปลี่ยนแปลงกำลังผลิตเมื่อมีการผลิตกำลังไฟฟ้าสูงขึ้น

$DR_i$  คือขีดจำกัดการเปลี่ยนแปลงกำลังผลิตเมื่อมีการผลิตกำลังไฟฟ้าลดลง

### 5.2 เงื่อนไขการเดินเครื่องของหน่วยการผลิตที่ต้องทำงานตลอด (Must run units)

ในระบบไฟฟ้ากำลังเครื่องกำเนิดไฟฟ้าบางเครื่องต้องทำการเดินเครื่องตลอดเวลาเพื่อใช้ในการรองรับการจ่ายโหลดตลอดเวลาเหตุผลของข้อสนับสนุนแรงดันไฟฟ้า (voltage support) ในเครือข่ายการส่งต่อหรือ เพื่อวัตถุประสงค์บางประการเช่น การจัดหาไอน้ำเพื่อการใช้ภายนอก ตัวเครื่องจักรเองทำให้ไม่สามารถปิดเครื่องได้ นอกจากจะทำการซ่อมบำรุง

### 5.3 ข้อบังคับของหน่วยเครื่องกำเนิดไฟฟ้าปริมาณน้ำ (Hydro constraints)

เครื่องกำเนิดไฟฟ้าปริมาณน้ำ ไม่สามารถแยกออกจากการคำนวณปัญหา Unit commitment ได้อย่างสิ้นเชิง แต่ในที่นี้เราจะตั้งสมมติฐานว่า ปัญหาเครื่องกำเนิดไฟฟ้าปริมาณน้ำ สามารถแยกจากปัญหา unit commitment ได้ เนื่องจากการผลิตไฟฟ้าใน เครื่องกำเนิดไฟฟ้าปริมาณน้ำจะไม่คิดค่าใช้จ่าย

### 5.4 ข้อบังคับของหน่วยเครื่องกำเนิดไฟฟ้าทางด้านการเชื้อเพลิง (Fuel Constraints)

ระบบของบางหน่วยมีปริมาณเชื้อเพลิงที่จำกัด หรือบางทีมีข้อบังคับที่ทำให้เผาผลาญเชื้อเพลิงจำนวนหนึ่งในเวลาที่กำหนด ซึ่งในปัจจุบันยังไม่มีทางแก้ไขปัญหา unit commitment ที่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 ปัญหาการจ่ายโหลดอย่างประหยัด (Economic dispatch)

การจ่ายโหลดอย่างประหยัด (Economic Dispatch Problem) ถือเป็นอีกหนึ่งปัญหาที่สำคัญในการดำเนินการผลิตกำลังไฟฟ้า การผลิตกำลังไฟฟ้าที่คำนึงถึงหลักทางเศรษฐศาสตร์หรือที่นิยมเรียกว่า ปัญหาการจ่ายโหลดอย่างประหยัด มีวัตถุประสงค์หลักในการแก้ปัญหาคือ ทำอย่างไรให้ต้นทุนการผลิตกำลังไฟฟารวมมีราคาน้อยที่สุด ขณะที่โหลดต้องได้รับกำลังไฟฟ้าที่เพียงพอกับความต้องการซึ่งต้นทุนในการผลิตกำลังไฟฟ้าคำนวณได้จากราคาเชื้อเพลิงรวมของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่ทำการผลิตกำลังไฟฟ้าให้กับระบบไฟฟ้านั้นเอง ปัญหาการจ่ายโหลดอย่างประหยัดนี้จะพิจารณาถึงกำลังไฟฟ้าที่สูญเสียจากสายส่งและฟังก์ชันราคาเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าพลังความร้อนเป็นฟังก์ชันพหุนามกำลังสอง (Quadratic Function) ซึ่งถือเป็นปัญหาทางคณิตศาสตร์ที่เรียกว่า ปัญหาโปรแกรมไม่เป็นเชิงเส้น สามารถประยุกต์ใช้วิธีในการแก้ปัญหาได้หลายวิธีด้วยกัน อาทิเช่น การประยุกต์ใช้ฟังก์ชันส่วนของเส้นตรงตัวในการแก้ปัญหา (Piecewise Linear Functions) มาประมาณฟังก์ชันเชื้อเพลิงที่อยู่ในรูปพหุนามกำลังสอง จากนั้นถึงใช้วิธีโปรแกรมเชิงเส้นคำนวณหาคำตอบ วิธีเชิงพลวัต เป็นวิธีที่เหมาะสมกับปัญหาการจ่ายโหลดอย่างประหยัดที่มีลักษณะของกราฟอินพุท/เอาต์พุตแบบ นันคอนเวกซ์ (Nonconvex) วิธีนิวตัน (Newton's Method) คำนวณหาคำตอบด้วยการประมาณอนุกรมเทเลอร์ วิธีเกรเดียนต์ (Gradient Method) อาศัยความลาดชันในการหาคำตอบ ซึ่งปกติทิศทางเกรเดียนต์จะชี้ไปในทิศทางที่มากที่สุด ดังนั้นเมื่อประยุกต์มาใช้กับปัญหาการจ่ายโหลดอย่างประหยัดจะต้องคูณด้วยค่าลบหนึ่งเสมอ และวิธีการทำซ้ำแลมดตา ที่อาศัยหลักการเท่ากันของต้นทุนที่เพิ่มขึ้น (Equal Incremental Cost)

ส่วนของการแก้ปัญหาการจ่ายโหลดอย่างประหยัดเป็นส่วนหนึ่งของการแก้ปัญหาชนิดคอมมิทเมนต์ตามสมการที่ 1 สำหรับส่วนที่เป็นการแก้ปัญหาการจ่ายโหลดอย่างประหยัดจากสมการที่ 1.1 นั้นเป็นการนำเครื่องกำเนิดไฟฟ้าที่เดินเครื่องอยู่ มาทำการแก้ปัญหาในการที่จะทำการจ่ายโหลดแต่ละเครื่องออกอย่างประหยัดที่สุด สำหรับส่วนที่เป็นการแก้ปัญหาการจ่ายโหลดอย่างประหยัดเป็นไปตามสมการที่ (1.13)

$$ED = \text{MIN} \sum_{i=1}^T \sum_{j=1}^N U_{ij} F_{ij}(P_{ij}) \quad (1.13)$$

จากสมการที่ 1.11 สิ่งที่จะทำให้การแก้ปัญหาการจ่ายโหลดอย่างประหยัดได้ค่าที่ต่ำที่สุดนั้นต้องได้จากการเดินหรือหยุดเครื่องกำเนิดไฟฟ้า ( $U_{ij}$ ) ที่เหมาะสมด้วย กล่าวคือมีการเดินเครื่องกำเนิดไฟฟ้าเท่าที่จำเป็นตามความต้องการกำลังไฟฟ้าของระบบ แต่การเดินหรือหยุดเครื่องกำเนิดไฟฟ้ามีผลต่อราคาการเริ่มเดินเครื่องกำเนิดไฟฟ้าด้วย การแก้ปัญหาสมการที่ 1.11 นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.13 นี้ต้องมีเงื่อนไขที่เกี่ยวข้องด้วย ซึ่งเงื่อนไขที่เกี่ยวข้องมีทั้งจากระบบ และจากเครื่องกำเนิดไฟฟ้า เงื่อนไขจากระบบเป็นไปตามสมการที่ 1.4 สำหรับเงื่อนไขจากเครื่องกำเนิดไฟฟ้า คือเงื่อนไขขีดจำกัดกำลังผลิตไฟฟ้าตามสมการที่ 1.6 ซึ่งจากเงื่อนไขทั้งสองนี้มีส่วนทำให้การแก้ปัญหาการจ่ายโหลดอย่างประหยัดซับซ้อนมากขึ้น ปกติวิธีการแก้ปัญหาการจ่ายโหลดอย่างประหยัดที่นิยมดังเช่นวิธี Lambda iteration และวิธี Merit order

โดยทั่วไปการแก้ปัญหาการจ่ายโหลดอย่างประหยัดนั้นเป็นที่ทราบกันดีว่าต้องแก้โดยให้ค่าอัตราการเพิ่มขึ้นของเชื้อเพลิง ( $x$ ) ของแต่ละเครื่องกำเนิดไฟฟ้าที่ร่วมกันจ่ายโหลดมีค่าเท่ากัน ตัวอย่างเช่นมีเครื่องกำเนิดไฟฟ้าความร้อนอยู่ 2 เครื่องที่ต้องจ่ายกำลังไฟฟ้าให้กับระบบ ดังนั้นจะได้สมการกำลังไฟฟ้าตามสมการที่ 1.14

$$D = P_1 + P_2 \quad (1.14)$$

เมื่อ  $P_1$  และ  $P_2$  คือกำลังไฟฟ้าที่ผลิตจากเครื่องกำเนิดไฟฟ้าที่ 1 และ 2 ตามลำดับ

$D$  คือความต้องการกำลังไฟฟ้าของระบบ

การผลิตกำลังไฟฟ้าของเครื่องกำเนิดไฟฟ้าทั้งสองทำให้เสียค่าใช้จ่ายเป็นไปตามสมการที่ 1.15

$$F_T = F_1 + F_2 \quad (1.15)$$

สำหรับค่าอัตราการเพิ่มขึ้นของราคาเชื้อเพลิงของแต่ละเครื่องกำเนิดไฟฟ้านั้นเป็นไปตามสมการที่ 1.16 และ 1.17

$$\lambda_1 = dF_1 / dP_1 \quad (1.16)$$

$$\lambda_2 = dF_2 / dP_2 \quad (1.17)$$

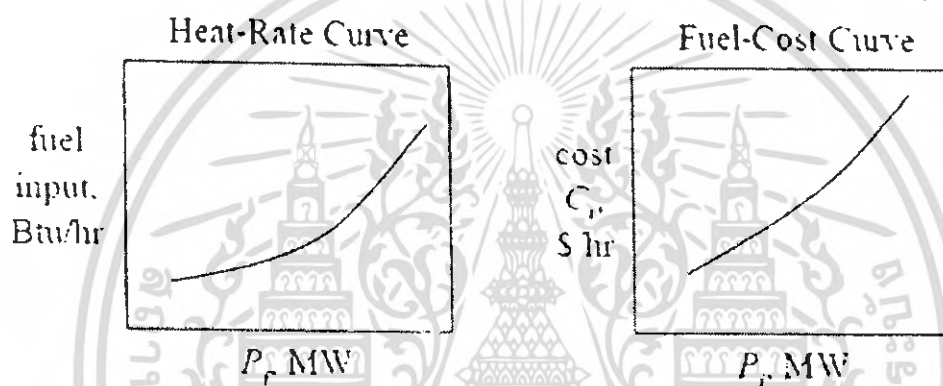
ถ้าหากเราเพิ่มกำลังผลิตไฟฟ้า  $P_1$  อีก  $\Delta P$  และลดกำลังการผลิตไฟฟ้า  $P_2$  ลง  $\Delta P$  นั้นหมายถึงกำลังผลิตไฟฟารวมยังคงเท่าเดิม แต่ค่าเชื้อเพลิงรวมใหม่กลายเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 F_{Total} &= F_1 + (dF_1/dP_1)\Delta P + F_2 - (dF_2/dP_2)\Delta P \\
 &= F_T + (\lambda_1 - \lambda_2)\Delta P
 \end{aligned}
 \tag{1.18}$$

จากสมการที่ 1.16 จะเห็นได้ชัดว่าถ้าหาก  $\lambda_1 \neq \lambda_2$  ก็จะต้องเสียค่าเชื้อเพลิงเพิ่มขึ้น ดังนั้นเราสามารถลดค่าเชื้อเพลิงรวมได้ โดยการถ่ายเทกำลังไฟฟ้าที่ผลิตขึ้นมาจากเครื่องกำเนิดไฟฟ้าเครื่องหนึ่งไปอีกเครื่องหนึ่งได้ กล่าวคือการผลิตกำลังไฟฟ้าในจุดที่ค่าเชื้อเพลิงต่ำสุดคือที่  $\lambda_1 = \lambda_2$  หรือในกรณีที่มีเครื่องกำเนิดไฟฟ้าอยู่  $N$  เครื่อง  $\lambda_1$  ถึง  $\lambda_N$  จะต้องมามีค่าเท่ากัน

### Generator heat rate curves lead to the fuel cost curves



รูปที่ 1.3 กราฟแสดงความสัมพันธ์ระหว่าง Heat-rate curve กับ fuel-cost curve รูปแบบทางคณิตศาสตร์ของปัญหาการจ่ายโหลดอย่างประหยัด

ปัญหาการจ่ายโหลดอย่างประหยัดนั้นเป็นการวางแผนการผลิตกำลังไฟฟ้าจากเครื่องกำเนิดไฟฟ้าที่อยู่ในระบบไฟฟ้าว่า เครื่องกำเนิดไฟฟ้าแต่ละเครื่องควรผลิตกำลังไฟฟ้าเท่าใด จึงจะใช้ต้นทุนในการผลิตกำลังไฟฟ้าน้อยที่สุดตลอดจนโหลดต้องได้รับกำลังไฟฟ้าที่พอเพียงกับความต้องการด้วย ซึ่งต้นทุนในการผลิตกำลังนี้ได้มาจากผลรวมราคาเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องจากคำอธิบายดังกล่าว นำมาเขียนเป็นสมการได้ดังนี้

$$\text{Minimize } F_{total} = \sum_{i=1}^N F_i(P_i) \tag{1.19}$$

เมื่อ  $i$  แทนเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  และ  $N$  แทนจำนวนเครื่องกำเนิดไฟฟ้าที่เชื่อมต่อกันในระบบไฟฟ้าทั้งหมด สมการ (1.19) แทนฟังก์ชันเป้าหมาย (Objective Function) ที่ต้องการหาค่าต้นทุนในการผลิตกำลังไฟฟ้ารวมน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมีเงื่อนไขบังคับว่า กำลังไฟฟ้าที่ผลิตออกมานั้นจะต้องมีค่าเท่ากับกำลังไฟฟ้าที่โหลดต้องการรวมกับกำลังไฟฟ้าที่สูญเสียจากสายส่ง ตลอดจนคำนึงถึงขีดจำกัดกำลังการผลิตต่ำสุดและสูงสุดของเครื่องกำเนิดไฟฟ้าแต่ละเครื่อง ดังสมการที่ 1.20 และ 1.21 ตามลำดับ

$$\sum_{i=1}^N (P_i) - P_{out} - P_{loss} = 0 \quad (1.20)$$

$$P_{i,\min} \leq P_i \leq P_{i,\max}, i=1,2,\dots,N \quad (1.21)$$

ส่วนกำลังไฟฟ้าสูญเสียจากสายส่งสามารถคำนวณได้จากสมการที่ 1.22

$$P_{loss} = \sum_{i=1}^N P_{Gi} - P_{load} \quad (1.22)$$

ซึ่งฟังก์ชันราคาเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าพลังความร้อนที่ใช้ในวิทยานิพนธ์ฉบับนี้ เป็นฟังก์ชันพหุนามกำลังสองดังจะเห็นในสมการที่ 1.23

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (1.23)$$

- เมื่อ  $F_{total}$  คือ ต้นทุนรวมในการผลิตกำลังไฟฟ้า,  
 $a_i, b_i, c_i$  คือ สัมประสิทธิ์ราคาเชื้อเพลิงเครื่องกำเนิดไฟฟ้าที่  $i$   
 $P_i$  คือ กำลังผลิตไฟฟ้าของเครื่องกำเนิดไฟฟ้าที่  $i$ , (MW)  
 $P_{load}$  คือ กำลังไฟฟ้าที่โหลดต้องการ, (MW)  
 $P_{loss}$  คือ กำลังไฟฟ้าสูญเสียในสายส่ง, (MW)  
 $P_{i,\min}$  คือ กำลังผลิตต่ำสุดของเครื่องกำเนิดไฟฟ้าที่  $i$ , (MW)  
 $P_{i,\max}$  คือ กำลังผลิตสูงสุดของเครื่องกำเนิดไฟฟ้าที่  $i$ , (MW)  
 $P_{Gi}$  คือ กำลังไฟฟ้าที่ผลิตได้จากเครื่องกำเนิดที่  $i$ , (MW)

สมการที่ (1.19) , (1.21) นำมาจัดรูปแบบให้อยู่ในรูปแบบมาตรฐานของปัญหาโปรแกรมไม่เป็นเชิงเส้นแบบพหุนามกำลังสองตามสมการที่(1.24) ถึง (1.25) ได้ดังนี้

$$\text{Minimize } F_{Total} = \sum_{i=1}^N F_i(P_i)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{Subject to } \sum_{i=1}^N (P_i) - P_{load} - P_{loss} = 0$$

$$P_{i,\min} \leq P_i \leq P_{i,\max} \quad (1.24)$$

โดยที่  $i = 1, 2, \dots, N$

$$\text{Minimize } F_{Total} = (a_1 P_1^2 + b_1 P_1) + (a_2 P_2^2 + b_2 P_2) + \dots + (a_N P_N^2 + b_N P_N)$$

$$\text{Subject to } \sum_{i=1}^N (P_i) - P_{load} - P_{loss} = 0$$

$$P_{i,\min} \leq P_i \leq P_{i,\max} \quad (1.25)$$

โดยที่  $i = 1, 2, \dots, N$

แปลงรูปแบบปัญหาในสมการที่ (1.25) ให้อยู่ในรูปแบบมาตรฐานของปัญหาโปรแกรมไม่เป็นเชิงเส้นแบบพหุนามกำลังสอง ดังนี้

$$\text{Minimize } \frac{1}{2} [P_1 \dots P_N] \begin{bmatrix} 2a_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & 2a_N \end{bmatrix} \begin{bmatrix} P_1 \\ \vdots \\ P_N \end{bmatrix} + [b_1 \dots b_N] \begin{bmatrix} P_1 \\ \vdots \\ P_N \end{bmatrix}$$

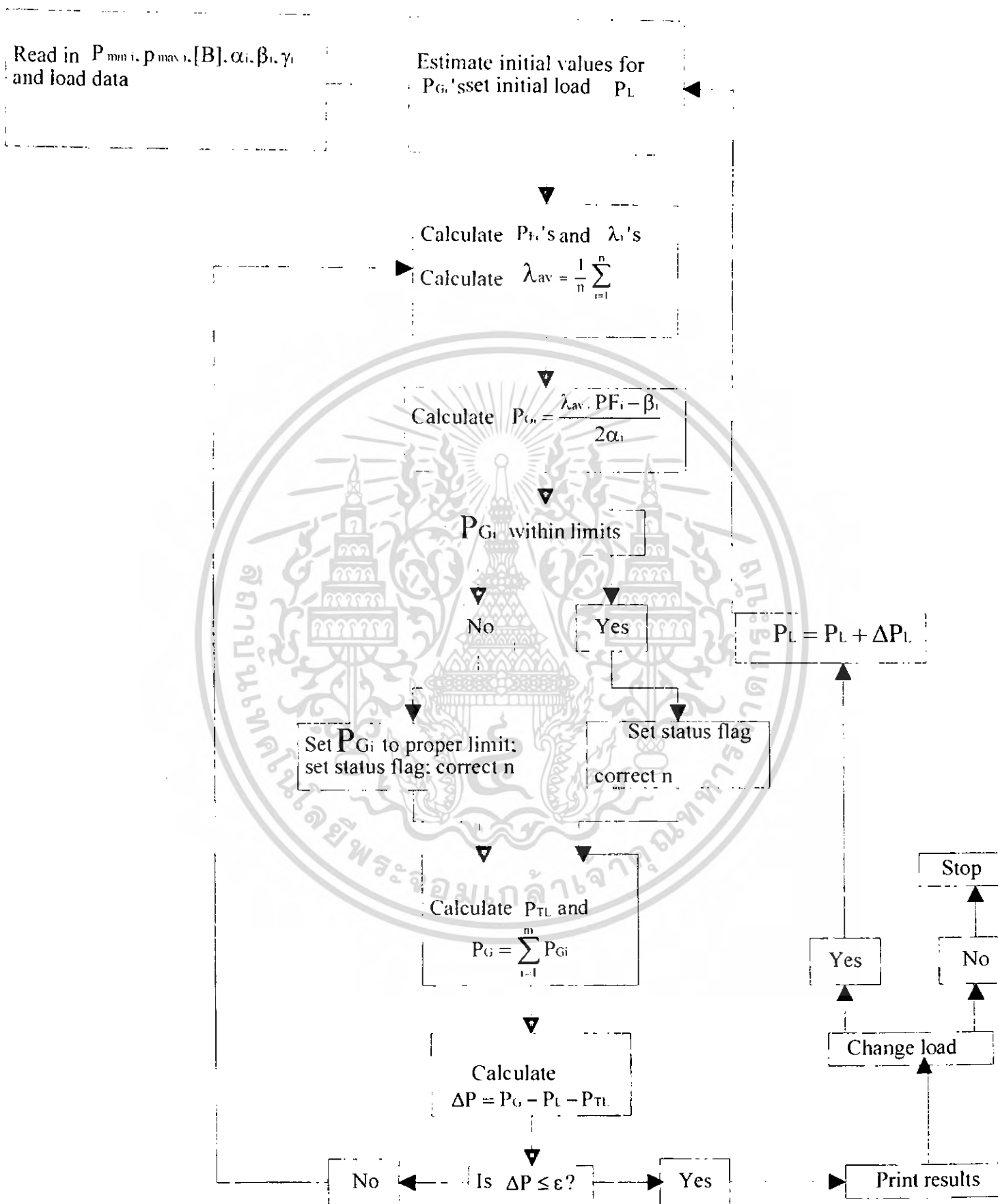
$$\text{Subject to } \sum_{i=1}^N (P_i) - P_{load} - P_{loss} = 0$$

$$P_{i,\min} \leq P_i \leq P_{i,\max} \quad (1.26)$$

โดยที่  $i = 1, 2, \dots, N$

โดยค่าสัมประสิทธิ์ราคาเชื้อเพลิงคงที่  $c_i$  ของเครื่องกำเนิดที่  $i$  ถึงเครื่องที่  $N$  จะนำมารวมในภายหลัง เมื่อแก้ปัญหасมการที่ (1.26) ได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.4 แผนภาพแสดงหลักการทำงานของ การโหลดอย่างประหยัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### หลักการและปัญหาในการทำยูนิตคอมมิตเมนต์

ในตลาดกลางซื้อขายไฟฟ้า การเสนอขายไฟฟ้าของบริษัทผลิตไฟฟ้านั้นจะต้องเสนอราคาขายและปริมาณกำลังไฟฟ้าที่ต้องการขาย โดยแยกเสนอเป็นข้อมูลของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่บริษัทผลิตไฟฟ้านั้น ๆ เป็นเจ้าของ ดังนั้น หากบริษัทผลิตไฟฟ้ามีเครื่องกำเนิดไฟฟ้าหลายเครื่อง เพื่อให้ได้กำไรสูงสุดโดยรวมตลอดวันที่ทำการซื้อขายไฟฟ้า บริษัทผลิตไฟฟ้าจึงต้องมีการทำยูนิตคอมมิตเมนต์ เพื่อช่วยในการกำหนดสถานะการทำงานของเครื่องกำเนิดไฟฟ้าให้เหมาะสม โดยในบทนี้จะกล่าวถึงความรู้ทั่วไปที่เกี่ยวข้องกับการทำยูนิตคอมมิตเมนต์ ซึ่งได้แก่ การวางแผน การผลิตไฟฟ้า ประเภทของโรงไฟฟ้า และการทำยูนิตคอมมิตเมนต์ในแบบต่าง ๆ [8,9]

#### การตรวจสอบเอกสารเกี่ยวกับการส่งจ่ายพลังงานไฟฟ้าในระบบไฟฟ้ากำลังอย่าง

##### ประหยัด

จากการศึกษาพบว่า การส่งจ่ายพลังงานไฟฟ้าและการจ่ายค่าใช้จ่ายในการผลิตน้อยที่สุด เมื่อเครื่องกำเนิดพลังงานไฟฟ้าทุกตัวทำการผลิตที่ค่า incremental coat of received power เท่ากัน (Kirchmayer and stagg, 1952; Kirchmayer, 1958)

Travers ;Harker;Long and Harder (1954) ได้ชี้ให้เห็นว่า มีผู้สนใจในการนำความสูญเสียในสายส่งเข้ามาศึกษา ในการส่งจ่ายพลังงานไฟฟ้ามานานแล้ว แต่วิธีการที่สามารถนำเอาผลของความสูญเสียในสายส่งมาศึกษา และนำไปร่วมเป็นแลคเตอร์หนึ่งในการส่งจ่ายพลังงานไฟฟ้านั้นได้แก่ วิธีการของ George ซึ่งมีหลักการว่า ค่าใช้จ่ายในการผลิตพลังงานไฟฟ้าจะเสียค่าใช้จ่ายน้อยที่สุดต่อเมื่อค่า incremental coat of received power ของเครื่องกำเนิดพลังงานไฟฟ้าจะต้องมีค่าเท่ากัน ค่าประกอบอื่นๆที่ต้องพิจารณาร่วมกันได้แก่ค่า incremental transmission losses

Kirchmayer and stagg (1952) ได้แสดงให้เห็นว่าจะใช้เชื้อเพลิงในการผลิตให้ต่ำที่สุดเพื่อให้ได้พลังงานไฟฟ้าตามที่ต้องการนั้นก็ต่อเมื่อค่า incremental coat of received power ต้องมีค่าเท่ากันโดยต้องสามารถสร้างสมการดิฟเฟอเรนเชียลแสดงความสัมพันธ์ระหว่างตัวแปรต่างๆ ซึ่งได้แก่ ค่าเชื้อเพลิงที่เพิ่มขึ้นในการเพิ่มการผลิตจ่ายกำลังไฟฟ้า ค่าผลรวมของการสูญเสียในสายส่ง และกำลังไฟฟ้าที่ผลิตเพิ่มขึ้น

George (1943) ได้ศึกษาการหาค่าความสูญเสียในสายส่งในรูปของแหล่งกำลัง และสามารถสร้างสมการความสัมพันธ์ของตัวแปรเกี่ยวกับความสูญเสียในสายส่งในรูปของ  $B_{mn}$  coefficient matrix และค่าของกำลังไฟฟ้าจากแหล่งกำลังหรือโรงจักร

Early, Watson and Smith (1956); Early Watson and Smith (1956) ได้ศึกษาความสูญเสียในสายส่งเพิ่มจากที่ George ได้ศึกษาเอาไว้ โดยเพิ่มตัวแปรใหม่ขึ้นจากเดิมเพื่อให้สมการในการหาค่าความสูญเสียในสายส่งนั้นมีความถูกต้องมากยิ่งขึ้นโดยเฉพาะในระบบที่ภาระทางไฟฟ้าหรือโหลดในแต่ละสถานีย่อยของระบบนั้นแปรเป็นอัตราส่วนกับโหลดรวมของระบบมีค่าแตกต่างกันออกไปมาก และชี้ให้เห็นว่าผลจากการศึกษาของ George จะใช้ได้ดีเฉพาะในกรณีที่โหลดของแต่ละสถานีย่อยนั้นมีค่าเป็นสัดส่วนคงที่กับค่าโหลดรวมของระบบเท่านั้น การศึกษาความสูญเสียในสายส่งเพิ่มเติมดังกล่าวข้างต้นได้สมการความสัมพันธ์ของตัวแปรที่ศึกษาไว้เดิม โดยมีค่าพารามิเตอร์ของค่ามิวซอลของสัมประสิทธิ์ของความสูญเสียระหว่างส่วนประกอบที่มีค่าคงที่ของโหลดและโรงจักร ( $B_{no}$ ) และค่าเซลฟ์และมิวซอลของสัมประสิทธิ์ของส่วนประกอบของโหลดที่มีค่าคงที่ ( $B_{oo}$ ) เพิ่มขึ้น

Imburgia; Kirchmayer and Stagg (1954) รายงานมีการวิเคราะห์วิธีการคำนวณ incremental fuel cost และ incremental transmission losses หลายวิธี เช่นการใช้สมการ simultaneous linear equation วิธีการ penalty factor ซึ่งกลุ่มวิศวกรของบริษัท American Gas and Electric service Corporation เป็นผู้ผลิตคิดค้นขึ้นมา เป็นต้น วิธีการ penalty factor ดูจะเป็นวิธีที่นำมาใช้มากที่สุด

Kirchmayer (1958) ได้เปรียบเทียบวิธีการหา coordinating incremental production cost and incremental transmission losses 3 วิธีคือวิธี exact method involving linear simultaneous equation วิธี และวิธี penalty factor method ผลจากการเปรียบเทียบปรากฏว่า ทั้ง 3 วิธีให้ผลลัพธ์ใกล้เคียงกันมาก แต่วิธี penalty factor method นั้นจะง่ายและใช้เวลาในการคำนวณหาผลลัพธ์น้อยกว่า ตลอดจนได้ค่าใกล้เคียงกับวิธี exact method involving linear simultaneous equation มาก

Dommel and Tinney (1968) ใช้วิธี Classical optimization method โดยใช้ Lagrangian multipliers หาค่าใช้จ่ายต่ำสุด และปรับปรุงวิธีการหา penalty factor ให้มีวิธีการหาให้มีประสิทธิภาพขึ้นโดยใช้วิธีของ Newton's method และ gradient adjustment algorithm

Hill and Stevenson, Jr. (1968a) หา loss factors จากค่าแอดมิตแตนซ์ และ ศักดาไฟฟ้าของระบบไฟฟ้ากำลัง โดยแสดงให้เห็นว่าสามารถลดความยุ่งยากในการหาค่า partial derivatives ในการหา incremental loss factors ได้มาก แต่วิธีการดังกล่าวจะได้ค่าใกล้เคียงความจริงเมื่อเป็นระบบที่ใช้ศักดาไฟฟ้าในการควบคุมระบบเท่านั้น

Hill and stvenson , Jr. (1968b) หาค่า  $B_{mn}$  coefficient โดยการหาความสูญเสียในสายส่งให้อยู่ในรูปของตัวแปรของค่า power input และแสดงให้เห็นว่าการหาค่าความสูญเสียที่ง่ายและรวดเร็ว ทั้งนี้เพราะวิธีการหาค่า partial derivatives ของศักดาไฟฟ้าและมุมที่สัมพันธ์กับ power input ของแต่ละโรงจักรนั้นไม่ซับซ้อนกับตารางผลิตพลังงานไฟฟ้าและโหลดรวม

Sanson (1969) ใช้วิธีการของ nonlinear programming แก้ปัญหาในการส่งจ่ายพลังงานไฟฟ้า

Meyer and Albertson (1971) หาค่า B coefficients โดยวิธีของ Tinny's optimally ordered direct solution techniques ซึ่งเป็นสมการหาจากค่า admittance matrix และแสดงให้เห็นว่าสามารถหาค่า  $B_{mn}$  coefficient ได้รวดเร็ว และใกล้เคียงความจริงมากกว่าวิธี iterative techniques ของ Hill and stvenson , Jr.

Nicholson and string (1972) ใช้วิธีของ linear programming formulation สำหรับเงื่อนไขบังคับในการส่งจ่ายพลังงานไฟฟ้า รวมทั้งการให้ฟังก์ชันของค่าใช้จ่ายอยู่ในรูปสมการคอดแตรติก (quadratic) ในการหาค่าใช้จ่ายในการผลิตและความสูญเสียในสายส่ง

Reid and Hasdorff (1972) ใช้วิธีของ Wolfe's slgorithm ในการแก้ปัญหาการผลิตจ่ายพลังงานไฟฟ้า โดยทำให้สมการอยู่ในรูปคอดแตรติกโปรแกรม (quadratic programming) และวิธีการดังกล่าวไม่ต้องหาค่า penalty factor

Podmore (1973) นำเอาค่าความปลอดภัย เช่นค่าจำกัดของความปลอดภัยในการส่งกำลังไฟฟ้าโดยการกำหนดให้เป็น linear inequality constraints ในเข้าพุทของโรงจักร

Happ (1977) รวบรวมวิธีการทั้งหมดที่มีผู้ศึกษาวิธีการผลิตจ่ายพลังงานไฟฟ้าให้ประหยัดค่าใช้จ่ายมากที่สุดแล้วสรุปว่า สมการแสดงความสัมพันธ์ของตัวแปรต่างๆ เพื่อให้เสียค่าใช้จ่ายในการผลิตจ่ายพลังงานไฟฟ้าให้น้อยที่สุดที่เรียกว่า classic coordination equation นั้น เป็นสมการที่นิยมใช้อยู่จนถึงปัจจุบัน นอกจากนี้ยังชี้ให้เห็นว่ามีวิธีการต่างๆ หลายวิธีใช้หาความสูญเสียในสายส่งโดยตรงไม่ต้องใช้  $B_{mn}$  coefficient matrix แต่อย่างไรก็ตาม การหาความสูญเสียจากค่า  $B_{mn}$  coefficient matrix ก็ยังคงเป็นวิธีที่มีผู้นิยมใช้กันมากที่สุดและยังคงเป็นวิธีที่ดีที่สุด ทั้งนี้เพราะจะช่วยในการประหยัดเวลาในการคำนวณหาค่าความสูญเสียต่างๆในระบบ และวิธีการในการคำนวณไม่ซับซ้อน ตลอดจนคำตอบที่ได้รับนั้นใกล้เคียงกับค่าแท้จริงด้วย

Shout; Chang; Helmick and Grady(1980) ได้แสดงค่า incremental fuel cost เสนอแนะให้ทำตารางแสดงค่า ค่ากำลังไฟฟ้าที่ผลิตได้ของแต่ละโรงจักรตลอดจนตารางแสดงผลรวมของกำลังไฟฟ้าที่ผลิตได้ทั้งหมดมาใช้เป็นจุดเริ่มแรกในการหาค่าใช้จ่ายและกำลังไฟฟ้าที่ผลิตจ่ายได้จริง โดยเริ่มจากไม่คิดความสูญเสีย แล้วจึงนำมาคิดรวมในภายหลัง จะช่วยประหยัดเวลาในการคำนวณได้มาก นอกจากการจัดทำตารางเพื่อหาค่าเริ่มต้นแล้ว ยังให้หาค่าใช้จ่ายเกี่ยวกับเชื้อเพลิงอยู่ในรูปของตัวแปรของกำลังไฟฟ้าที่ผลิตจ่ายได้และค่าสัมประสิทธิ์ของสมการแสดงคุณสมบัติเฉพาะของเครื่องเกี่ยวกับค่าใช้จ่ายในการผลิตจ่ายกำลังไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1 การวางแผนการผลิตไฟฟ้า

โดยทั่วไปการวางแผนการผลิตไฟฟ้าสามารถแบ่งออกเป็น 2 ประเภท ตามระยะเวลาได้ ดังนี้คือ

### 1) การวางแผนพัฒนา

เป็นการกำหนดเสนองานเพื่อปรับปรุงและขยายระบบไฟฟ้า อาจมีการเพิ่มสายส่งหรือ ก่อสร้างโรงไฟฟ้า ซึ่งเป็นแผนระยะยาว มีระยะเวลา 3 ปี ถึง 5 ปีขึ้นไป

### 2) การวางแผนปฏิบัติการ (Operational planning)

อาจแบ่งตามกิจกรรมหลักได้ ดังนี้

- แผนการผลิตและส่งจ่ายไฟฟ้า
- แผนการบำรุงรักษา
- แผนการใช้เชื้อเพลิง

สำหรับแผนการผลิตและจ่ายไฟฟ้าอาจแบ่งได้เป็น 3 แผนการ ดังต่อไปนี้

ก. แผนการผลิตและจ่ายไฟฟ้าระยะยาว (Long term planning)

เป็นแผนการสำหรับใช้ในช่วงระยะเวลาประมาณ 1-3 ปี

ข. แผนการผลิตและจ่ายไฟฟ้าระยะกลาง (Medium term planning)

เป็นแผนการสำหรับใช้ในช่วงระยะเวลาประมาณ 1 เดือน – 1 ปี

ค. แผนการผลิตและจ่ายไฟฟ้าระยะสั้น (Short term planning)

เป็นแผนการสำหรับใช้ในช่วงระยะเวลาประมาณ 1 วัน – 1 สัปดาห์ การวางแผนประเภทนี้ครอบคลุมถึงกระบวนการทำยูนิटकอมมิทเมนต์ ซึ่งประกอบด้วยกระบวนการตัดสินใจที่สำคัญ 2 ส่วน คือ

#### 1) การกำหนดสถานะการทำงานของเครื่องกำเนิดไฟฟ้าในระบบ

เป็นการคำนวณหาว่าเครื่องกำเนิดไฟฟ้าเครื่องใด ควรจะเดินเครื่อง (ON) หรือหยุดเดินเครื่อง (OFF) ในช่วงคาบเวลาใด จึงจะทำให้มีกำลังผลิตในระบบเพียงพอ กับความต้องการ โดยมีต้นทุนการผลิตรวมต่ำที่สุด

#### 2) การกำหนดค่ากำลังการจ่ายโหลดของเครื่องกำเนิดไฟฟ้าแต่ละเครื่อง

เป็นการคำนวณหาว่าเครื่องกำเนิดไฟฟ้าที่ได้รับการกำหนดให้เดินเครื่องในแต่ละคาบเวลานั้นควรจะจ่ายโหลดปริมาณเท่าใดจึงจะเพียงพอ กับความต้องการของระบบและทำให้มีต้นทุนการผลิตรวมต่ำสุด ปัญหานี้เรียกกันทั่วไปว่าการจ่ายโหลดอย่างประหยัด (Economic Dispatch)

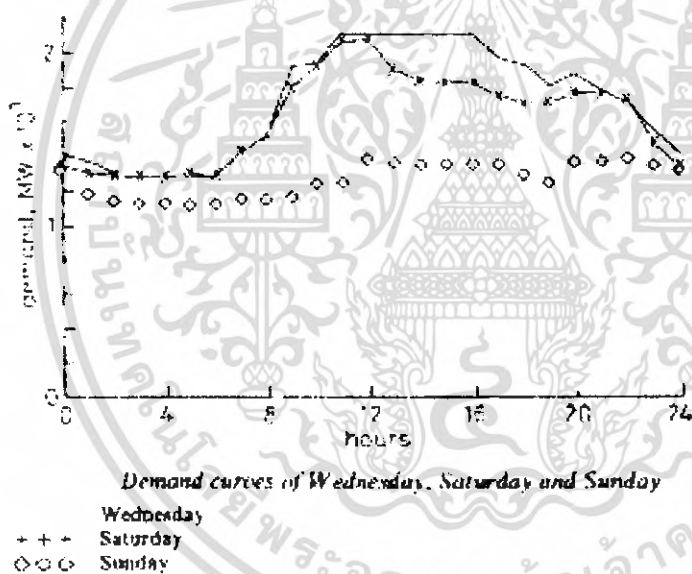
เนื่องจากการคำนวณหาค่าที่ต้องการของปัญหาทั้งสองมีลักษณะต่อเนื่องกัน ดังนั้นในการคำนวณจึงมักรวมปัญหาทั้งสองเข้าด้วยกันเป็นปัญหายูนิटकอมมิทเมนต์ ที่มีปัญหาการจ่ายโหลดอย่างประหยัดเป็นปัญหาย่อยภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ปัญหาที่เกี่ยวข้องกับการทำยูนิคคอมมิตเมนต์

### 1) การทำนายโหลด (Load forecasting)

ข้อมูลโหลดที่ใช้ในการทำยูนิคคอมมิตเมนต์เป็นโหลดที่ได้จากการทำนายล่วงหน้า (Forecasted load) ซึ่งความผิดพลาดจากการพยากรณ์เป็นสิ่งที่หลีกเลี่ยงไม่ได้ ในกรณีที่ทำนายโหลดไว้ต่ำกว่าความเป็นจริงจะทำให้กำลังผลิตที่ได้รับการจัดสรรไว้ในระบบไม่เพียงพอ กับความต้องการ ซึ่งจะทำให้ต้องใช้เครื่องกำเนิดไฟฟ้าที่มีต้นทุนการผลิตสูงมาใช้จ่ายโหลดเพิ่มเติมส่งผลให้ต้นทุนการผลิตรวมมีค่าสูงกว่าที่ควร แต่ถ้าหากทำนายโหลดมากเกินไปเกินความเป็นจริงก็จะทำให้เครื่องกำเนิดไฟฟ้าที่มีต้นทุนสูงอาจได้รับการกำหนดให้เดินเครื่องโดยไม่จำเป็น ดังนั้น จึงอาจกล่าวได้ว่าการทำยูนิคคอมมิตเมนต์จะให้ประโยชน์ได้อย่างเต็มที่ก็ต่อเมื่อโหลดที่ได้จากการทำนายนั้นมีความแม่นยำในระดับที่ยอมรับได้เท่านั้น



รูปที่ 2.1 กราฟแสดงความต้องการไฟฟ้าในแต่ละวัน

### 2) กำลังผลิตสำรอง (Spinning reserve)

นอกจากความผิดพลาดในการทำนายโหลดแล้ว ปัญหาอีกอย่างหนึ่งที่เกี่ยวข้องกับการทำยูนิคคอมมิตเมนต์ คือ ความขัดข้องของเครื่องกำเนิดไฟฟ้าที่ต่อขนานอยู่กับระบบซึ่งอาจเกิดขึ้นโดยไม่มีทางรู้ล่วงหน้าได้ เหตุการณ์เหล่านี้เป็นเหตุการณ์ที่ไม่สามารถหลีกเลี่ยงได้ แต่ปัญหาอันเนื่องมาจากเหตุการณ์เหล่านี้สามารถป้องกันได้ โดยทั่วไปแล้วเราจะกำหนดให้กำลังการผลิตติดตั้งของเครื่องกำเนิดไฟฟ้าที่ได้รับการกำหนดให้เดินเครื่องมีค่ามากกว่าปริมาณโหลดที่ทำนายไว้ โดยความสามารถส่วนที่เกินกว่าโหลดนี้เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญตไหนไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังผลิตสำรอง (Spinning reserve) ความหมายของกำลังผลิตสำรอง คือ ปริมาณกำลังผลิตที่สามารถผลิตได้จากเครื่องกำเนิดไฟฟ้าทั้งหมดที่ต้องขนานอยู่กับระบบลดด้วยโหลดและกำลังสูญเสียที่เวลานั้น ๆ กำลังผลิตสำรองนี้ต้องมีค่ามากเพียงพอเพื่อว่าเวลาที่เครื่องกำเนิดไฟฟ้าเกิดการขัดข้องจะไม่ทำให้ความถี่ของระบบมีค่าตกลงมากจนเกินไป

ปริมาณที่เหมาะสมของกำลังผลิตสำรองควรมีค่าเท่าใดนั้น ขึ้นกับแนวทางการปฏิบัติงานของระบบไฟฟ้าแต่ละแห่ง บางแห่งอาจกำหนดให้มีค่าเท่ากับกำลังผลิตสูงสุดของเครื่องกำเนิดไฟฟ้าที่ใหญ่ที่สุด ในขณะที่บางแห่งอาจกำหนดจากความเสี่ยงของยูนิทคอมมิทเมนต์ (Unit commitment risk)

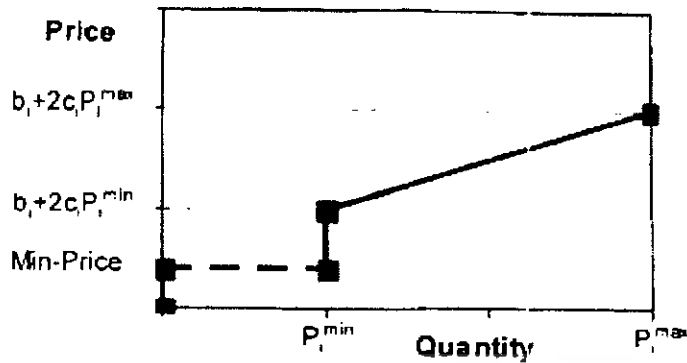
## 2.2 ประเภทของโรงไฟฟ้า

โดยทั่วไประบบไฟฟ้าประกอบด้วยโรงไฟฟ้าจำนวนมากซึ่งมีลักษณะแตกต่างกันไป ดังนี้

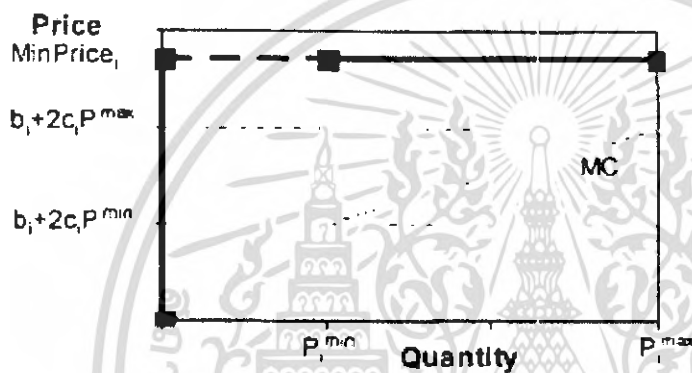
### 2.2.1 โรงไฟฟ้าพลังความร้อน (Thermal power plant)

การผลิตพลังงานไฟฟ้าโดยโรงไฟฟ้าประเภทนี้อาศัยการใช้เชื้อเพลิงชนิดต่าง ๆ เช่น น้ำมันเตา ถ่านหิน ก๊าซธรรมชาติ ฯลฯ นำมาเผาไหม้ให้เกิดพลังงานความร้อน พลังงานความร้อนที่ได้จะถูกแปลงเป็นพลังงานกลเพื่อนำไปหมุนกังหันของเครื่องกำเนิดไฟฟ้าเพื่อให้ได้พลังงานไฟฟ้าอีกต่อหนึ่ง โรงไฟฟ้างัดกล่าวอาจแยกออกเป็นประเภทต่าง ๆ ดังนี้

- 1) โรงไฟฟ้าพลังไอน้ำ อาศัยความร้อนที่ได้จากเชื้อเพลิงไปต้มน้ำจนเดือดเป็นไอน้ำ ไอน้ำที่ได้นี้มีความดันสูงมากจนสามารถหมุนเครื่องกังหันไอน้ำเพื่อผลิตไฟฟ้าได้
- 2) โรงไฟฟ้ากังหันก๊าซ โรงไฟฟ้าประเภทนี้ไม่ต้องอาศัยการต้มน้ำแต่ใช้ก๊าซร้อนที่มีความดันสูงไปหมุนกังหันของเครื่องกำเนิดไฟฟ้าโดยตรง
- 3) โรงไฟฟ้าดีเซล หลักการผลิตไฟฟ้าคล้ายกับโรงไฟฟ้ากังหันก๊าซแต่ใช้เชื้อเพลิงเป็นน้ำมันดีเซล
- 4) โรงไฟฟ้าพลังความร้อนร่วม เนื่องจากเชื้อเพลิงก๊าซที่เหลือจากโรงไฟฟ้ากังหันก๊าซยังมีความร้อนสูงมาก เราจึงสามารถนำก๊าซที่เหลือนี้มาต้มน้ำให้เดือดเป็นไอน้ำแล้วไปหมุนเครื่องกังหันไอน้ำได้ ซึ่งจะทำให้ใช้พลังงานได้คุ้มค่ามากยิ่งขึ้น โรงไฟฟ้าประเภทนี้จึงเปรียบเสมือนโรงไฟฟ้าพลังไอน้ำร่วมกับโรงไฟฟ้ากังหันก๊าซ



Bid when MinPrice is lower than marginal cost



Bid when MinPrice is higher than marginal cost. The marginal cost function, which is also submitted to the PX, is shown (MC)

รูปที่ 2.2 กราฟแสดงความสัมพันธ์ระหว่างราคาการผลิตกับหน่วยการผลิต

## 2.2.2 โรงไฟฟ้าพลังน้ำ (Hydro power plant)

การผลิตไฟฟ้าโดยวิธีนี้ใช้พลังงานศักย์จากน้ำที่กักเก็บไว้ในอ่างหรืออาศัยพลังงานจลน์จากการไหลของแม่น้ำที่มีอยู่ตามธรรมชาติมาใช้หมุนกังหันน้ำของเครื่องกำเนิดไฟฟ้า การผลิตไฟฟ้าด้วยโรงไฟฟ้าประเภทนี้จึงไม่เสียต้นทุนค่าเชื้อเพลิง สำหรับเครื่องกำเนิดไฟฟ้าที่อยู่ในโรงไฟฟ้าพลังน้ำสามารถแบ่งออกได้เป็น 2 ประเภท ดังนี้

- 1) เครื่องกำเนิดไฟฟ้าพลังน้ำ (Hydro unit) เครื่องกำเนิดไฟฟ้าประเภทนี้มีวิธีการผลิตไฟฟ้าเหมือนกับที่อธิบายไว้ข้างต้น
- 2) เครื่องกำเนิดไฟฟ้าแบบสูบแบบสูบกลับ (Pump storage unit) เครื่องกำเนิดไฟฟ้าประเภทนี้มีลักษณะการทำงานเหมือนเครื่องกำเนิดไฟฟ้าพลังน้ำ แต่มีความสามารถพิเศษตรงที่สามารถสูบน้ำกลับได้ โดยการสูบน้ำกลับนี้จะใช้พลังงานไฟฟ้าจากระบบมาเป็นแหล่งพลังงานในการสูบน้ำกลับ ดังนั้น เครื่องกำเนิดไฟฟ้าประเภทนี้จึงสามารถเป็นได้ทั้งเครื่องผลิตไฟฟ้าและเครื่องสูบน้ำได้ในตัว เดียวกัน

เครื่องกำเนิดไฟฟ้าแต่ละเครื่องในโรงไฟฟ้าอาจแยกกันทำงานได้อย่างอิสระ หรือต้องมี

เอกสารที่ร่วมกันเพื่อให้ปริมาณการปล่อยน้ำของโรงไฟฟ้ามีค่าตามที่กำหนด ซึ่งประโยชน์ด้านการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 การทำยูนิตคอมมิตเมนต์ด้วยวิธีต่าง ๆ และเอกสารที่เกี่ยวข้อง

การหาผลลัพธ์ของปัญหายูนิตคอมมิตเมนต์ได้มีการพัฒนาเทคนิควิธีการมาโดยตลอด เป็นระยะเวลาานาน วิธีการต่าง ๆ ที่นำมาใช้ในการคำนวณ พอจะสรุปได้ดังนี้

### 2.3.1 วิธีเรียงความลำดับ (Priority list)

การทำยูนิตคอมมิตเมนต์โดยวิธีนี้จะเลือกเดินเครื่องกำเนิดไฟฟ้าตามลำดับที่กำหนดไว้ ซึ่งการเรียงลำดับนี้อาจทำได้หลายวิธีขึ้นอยู่กับความต้องการของแต่ละระบบไฟฟ้า เช่น เรียงลำดับตามขนาดพิกัดของเครื่องกำเนิดไฟฟ้า เรียงลำดับตามประสิทธิภาพของเครื่องกำเนิดไฟฟ้า หรือ เรียงลำดับตามราคาเชื้อเพลิงต่อหน่วยพลังงาน ( $\$/\text{MW-h}$ ) ฯลฯ การเรียงลำดับนี้อาจมีการดัดแปลงโดยคำนึงถึงลักษณะของโหลด ความมั่นคงของระบบในพื้นที่ต่าง ๆ กำลังงานสูญเสียในสายส่ง ผลของค่าใช้จ่ายในการเริ่มต้นเครื่อง เป็นต้น

วิธีที่นิยมใช้กันทั่วไปจะเป็นการเรียงลำดับเครื่องกำเนิดไฟฟ้าตามลำดับของต้นทุนการผลิตเฉลี่ยที่กำลังผลิตสูงสุด (Average Full Load Cost : AFLC) จากต้นทุนการผลิตต่ำไปหา ต้นทุนการผลิตสูง โดยจะเริ่มจากเครื่องที่มีต้นทุนการผลิตต่ำก่อนแล้วค่อย ๆ เพิ่มลำดับจนได้ กำลังผลิตรวมมีค่าเท่ากับโหลดที่ต้องการ การคำนวณหาต้นทุนการผลิตเฉลี่ยที่กำลังผลิตสูงสุดสามารถกระทำได้โดยนำต้นทุนค่าเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าที่กำลังผลิตสูงสุดหารด้วย กำลังผลิตสูงสุดดังสมการ

$$\text{AFLC}_i = \frac{F1(P_{\text{max}_i})}{P_{\text{max}_i}} \quad (3.1)$$

$F1(1)$  คือ ต้นทุนค่าเชื้อเพลิงของเครื่องที่  $i$  เป็นฟังก์ชันที่ขึ้นกับกำลังผลิต ( $B$ )

$P_{\text{max}_1}$  คือ กำลังผลิตสูงสุดของเครื่องที่  $i$  (MW)

สำหรับขั้นตอนการทำยูนิตคอมมิตเมนต์โดยวิธีเรียงตามลำดับมีดังต่อไปนี้

- 1) กำหนดค่า AFLC ของเครื่องกำเนิดไฟฟ้าทุกเครื่อง แล้วเรียงลำดับเครื่องกำเนิดไฟฟ้า ตามค่า AFLC จากต่ำไปสูง
- 2) ในแต่ละชั่วโมงของช่วงเวลาที่กำหนด
  - พิจารณาโหลดและกำลังผลิตสำรองที่ระบบต้องการจากข้อมูลการพยากรณ์โหลด
  - เลือกเดินเครื่องกำเนิดไฟฟ้าตามลำดับ AFLC จนกระทั่งกำลังผลิตรวมมีค่าไม่น้อยกว่าผลบวกของโหลดและกำลังผลิตสำรองของระบบ
  - ตรวจสอบเงื่อนไขของเวลาเดินเครื่องน้อยที่สุด (Minimum up time) และเวลาหยุดเดินเครื่องน้อยที่สุด (Minimum down time) ของเครื่องกำเนิดไฟฟ้าที่นำมาใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) เมื่อได้แผนการเดินทางเครื่องที่เป็นไปได้แล้วให้พิจารณาว่าสามารถนำบางเครื่องที่เดินเครื่องไม่เต็มที่ออกจากระบบได้หรือไม่ เพื่อลดต้นทุนการผลิต

ข้อดีของวิธีเรียงตามลำดับ คือ หาผลลัพธ์ของยูนิตคอมมิทเมนต์ได้ง่ายและเร็วแต่มีข้อเสีย คือ ไม่ได้คำนึงถึงการทำยูนิตคอมมิทเมนต์ร่วมกับการจ่ายโหลดอย่างประหยัดอย่างชัดเจน เนื่องจากวิธีนี้พิจารณาแค่เพียงบางส่วนของกำหนดการเดินทางเครื่อง (Combination) ที่เป็นไปได้เท่านั้น ผลลัพธ์ที่ได้จากวิธีนี้จึงมักให้ค่าที่ต่างจากค่าที่เหมาะสม ข้อเสียอีกประการหนึ่ง คือ ไม่สามารถใช้กับระบบที่ประกอบด้วยเครื่องกำเนิดไฟฟ้าที่มีเชื้อเพลิงจำกัด หรือเครื่องกำเนิดไฟฟ้าพลังน้ำได้โดยตรง

บทความที่นำเสนอการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Priority List มีดังต่อไปนี้

(1) R.R Shoult, S.K.Chang, S.Helmick, and W.M.Grady, "A practical approach to unit commitment, economic dispatch and saving allocation for multiple-area pool operation with import/export constrain", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-99, No.2, pp. 625-633, March 1980.

(2) F.N. Lee, "Short-term unit commitment-a new method", IEEE Trans. on Power Systems, Vol.3.No. 2, pp.421-428 May 1988.

### 2.3.2 วิธีโปรแกรมเชิงพลวัต (Dynamic programming)

วิธีโปรแกรมเชิงพลวัต (Dynamic programming) เป็นเทคนิคซึ่งแตกต่างจากวิธีโปรแกรมเชิงเส้น (Linear programming) วิธีนี้จะแบ่งการทำยูนิตคอมมิทเมนต์ออกเป็นช่วง ๆ แล้วหาคำตอบที่เหมาะสม โดยคำนึงถึงการเปลี่ยนสถานะ (State) ในแต่ละเวลาด้วย วิธีโปรแกรมเชิงพลวัตมักให้คำตอบดีกว่าวิธีเรียงลำดับเพราะจำนวนสถานะที่วิธีนี้ตรวจสอบจะมีมากกว่า ยกตัวอย่างเช่น ระบบที่มีเครื่องกำเนิดไฟฟ้า 4 เครื่องจะมีสถานะที่เป็นไปได้ในการจ่ายโหลดที่เวลาหนึ่ง ๆ เท่ากับ  $2^4 - 1 = 15$  และถ้าหากมีโหลดที่เวลาต่าง ๆ กันเป็นจำนวน 10 ช่วง จำนวนเส้นทาง (Strategy) ที่เป็นไปได้ทั้งหมดของการทำยูนิตคอมมิทเมนต์จะมีค่าเท่ากับ  $(2^4 - 1)^{10}$  ซึ่งเป็นค่าที่สูงมากทำให้การคำนวณหาต้นทุนการผลิตจากเส้นทางเหล่านี้ทั้งหมดเป็นไปได้ยากโดยเฉพาะในระบบที่มีเครื่องกำเนิดไฟฟ้าหลายเครื่อง เนื่องจากต้องใช้เวลาในการคำนวณยาวนานมาก อย่างไรก็ตามหากการค้นหาคำตอบนำเอาลำดับเครื่องกำเนิดไฟฟ้าที่เหมาะสมมาพิจารณาร่วมด้วยก็จะช่วยลดขนาดของปัญหาลงไปได้มาก เพราะจำนวนสถานะที่เป็นไปได้จะลดลงเช่นกัน

สถานะที่ 1 : เครื่องที่ 1

สถานะที่ 2 : เครื่องที่ 1 + เครื่องที่ 2

สถานะที่ 3 : เครื่องที่ 1 + เครื่องที่ 2 + เครื่องที่ 3

สถานะที่ 4 : เครื่องที่ 1 + เครื่องที่ 2 + เครื่องที่ 3 + เครื่องที่ 4

อย่างไรก็ตามการนำเอาวิธีเรียงตามลำดับมาใช้กับการทำโปรแกรมเชิงพลวัตมักจะให้ผลตอบที่ได้มีต้นทุนการผลิตไฟฟ้าสูงกว่าการคำนวณตามเส้นทางที่เป็นไปได้ทั้งหมด ดังกล่าวข้างต้น

### สมมติฐานของการใช้วิธีโปรแกรมเชิงพลวัตมีดังนี้

- สามารถแยกปัญหาออกได้เป็นหลายสถานะ
- สถานะ คือ เซตที่บอกให้ทราบว่าเครื่องกำเนิดไฟฟ้าเครื่องใดเดินเครื่องหรือหยุดเดินเครื่อง
- ไม่มีค่าใช้จ่ายในการสั่งหยุดเดินเครื่อง (Shut down)
- สถานะที่เป็นไปได้ในแต่ละเวลาคือสถานะที่เครื่องกำเนิดไฟฟ้ามีกำลังผลิตรวมไม่น้อยกว่าผลรวมของโหลดและกำลังผลิตสำรองที่ระบบต้องการในเวลานั้น ๆ

วิธีโปรแกรมเชิงพลวัตใช้คำนวณยูนิตคอมมิทเมนต์ได้ดีสำหรับระบบขนาดเล็กและขนาดปานกลาง แต่มีข้อเสียเมื่อใช้กับระบบขนาดใหญ่ เนื่องจากเมื่อจำนวนเครื่องกำเนิดไฟฟ้าเพิ่มขึ้นจำนวนสถานะที่เป็นไปได้จะเพิ่มขึ้นมากทำให้จำเป็นต้องใช้เวลาและหน่วยความจำในการคำนวณเพิ่มขึ้นเป็นอย่างมาก อย่างไรก็ตามได้มีการปรับปรุงวิธีโปรแกรมเชิงพลวัตให้ค้นหาคำตอบได้รวดเร็วขึ้น เช่น Sequential dynamic programming และ Sequential-truncated dynamic programming โดยวิธีแรกจะจำกัดจำนวนเส้นทางที่ต้องเก็บไว้ในขณะที่วิธีที่สองจะใช้วิธีลดช่วงการค้นหาให้น้อยลง ซึ่งทั้งสองวิธีสามารถลดจำนวนสถานะการเดินเครื่องที่เป็นไปได้ให้น้อยลง

บทความที่นำเสนอการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Dynamic Programming มีดังต่อไปนี้

(1) P.G. Lowery, "Generating unit commitment by dynamic programming", IEEE Trans. on Power Apparatus and Systems, Vol.PAS-85, No.5, pp.422-426, May 1966.

(2) W.L. Snyder and H.D. Powell, "Dynamic programming approach to unit commitment", IEEE Trans. on Power Systems, Vol. 2, No. 2, pp. 339-350, May 1987.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) C.K.Pang and H.C. Chen, "Optimal short-term thermal unit commitment", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-95, No. 4, pp. 1336-1346, July 1976.

(4) C.K. Pang, G.B. Sheble and F. Albuyeh, "Evaluation of dynamic programming based methods and multiple area representation for thermal unit commitment", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 3, pp. 1212-1218, March 1981.

(5) P.P.J. Van den Bosch and G. Honderd, "A solution of the unit commitment problem via decomposition and dynamic programming", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-104, No. 7, pp. 1684-1690, July 1985

### 2.3.3 วิธีมิกซ์-อินทิเจอร์ (Mixed-Integer Programming)

การแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธีนี้ทำได้โดยการละทิ้งช่วงการค้นหาคำตอบที่ไม่มีโอกาสเข้าสู่คำตอบได้ หลักการในวิธีนี้ใช้วิธีการคาดหมายของคำตอบ ซึ่งโดยพื้นฐานแล้วการคำนวณมีลักษณะเป็นเชิงเส้น โดยปกติแล้ววิธีนี้คล้ายกับวิธี Branch and Bound

บทความที่นำเสนอการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Mixed-Integer Programming มีดังต่อไปนี้

(1) J.A. Muckstadt and R.C. Willson, "An application of mixed-integer programming duality to scheduling thermal generating system", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-87, Vol. 12, pp. 1968-1977, December 1968.

(2) T.S. Dillon, K.W. Edwin, H.D. Kochs and R.J. Taud, "Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-97, No. 6, pp.2154-2166. November 1978.

(3) J.J. Shaw and D.P. Bertsekas, "Optimal scheduling of large hydrothermal power system", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-104, No. 2, pp. 286-294, February 1985.

### 2.3.4 วิธีเบเนเดอร์ ( Benders Decomposition)

ในวิธีนี้ปัญหาจะถูกแบ่งเป็นสองปัญหา กล่าวคือปัญหาแรกเป็นปัญหาหลักที่ต้องพิจารณาการเดินหรือหยุดเดินเครื่องกำเนิดไฟฟ้า และปัญหารองเป็นปัญหาการจ่ายโหลดอย่างประหยัด แต่การแก้ปัญหาดังกล่าวมีความยุ่งยากในการหาผลลัพธ์ของปัญหาหลัก เนื่องจากเป็นปัญหาค่าขนาดใหญ่ และวิธีจะไม่สามารถแก้ปัญหาค่าในกรณีที่เครื่องกำเนิดไฟฟ้ามีเงื่อนไขเพิ่มเติม

บทความที่นำเสนอการแก้ปัญหาค่าการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Benders Decomposition มีดังต่อไปนี้

(1) H. Habibollahzadeh and J.A. Bubenko, "Application of decomposition techniques to short-term operation planning of hydrothermal power system", IEEE Trans. on Power Apparatus and Systems, Vol. 1, pp.41-47, February 1986.

### 2.3.5 วิธีแลกรังเจียนแบบลากรองจ์ (Lagrangian relaxation)

วิธีนี้เป็นวิธีที่เหมาะสมกับระบบไฟฟ้ากำลังขนาดใหญ่ เนื่องจากสามารถแยกปัญหาขั้นต้นที่มีเครื่องกำเนิดไฟฟ้าหลายเครื่องออกเป็นปัญหาย่อย ๆ ได้ทำให้เวลาที่ใช้ในการคำนวณลดลง พื้นฐานของวิธีแลกรังเจียนแบบลากรองจ์ (Lagrangian relaxation) มาจากเทคนิคการแก้ปัญหาค่าคู่ (Dual optimization) โดยเทคนิคดังกล่าวจะเปลี่ยนรูปแบบของปัญหาพริมาล (Primal problem) ซึ่งเป็นปัญหาขั้นต้นให้เป็นรูปแบบของปัญหาคู่ (Dual problem) โดยปัญหาทั้งสองนี้เป็นปัญหาที่สมมูลกัน

บทความที่นำเสนอการแก้ปัญหาค่าการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Lagrangian Relaxation มีดังต่อไปนี้

(1) A. Merlin and P. Sandrin, "A new method for unit commitment at Electricite' de France", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-102, No. 5, pp.1218-1225, May 1983.

(2) F. Zhuang and F.D. Galina, "Towards a more rigorous and practical unit commitment by lagrangian relaxation", IEEE Trans. on Power Systems, Vol. 3, No. 2, pp. 763-773, May 1988.

(3) S.K. Tong and S.M. Shahidehpour, "Combination of lagrangian relaxation and linear programming approaches for fuel constrained unit commitment problem", IEE Proceeding Part C, Vol. 36, No. 3, pp. 162-174, May 1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4) S. Virmani, K. Imhof and S. Mukhenjee, "Implementation of a lagrangian relaxation based unit commitment problem", IEEE Trans. on Power Systems, Vol. 4, No. 4, pp. 1375-1379, November 1989.

(5) C. Wang and S.M. Shahidepour, "Optimal generation scheduling with ramping costs", IEEE Trans. on Power Systems, Vol. 10, No. 1, pp. 60-67, February 1995.

(6) R. Baldiek, "The generalized unit commitment problem", IEEE Trans. on Power Systems, Vol. 10, No. 1, pp. 465-475, February 1995.

(7) W.L. Peterson and S.R. Brammer, "A capacity based lagrangian relaxation unit commitment with ramp rate constraints", IEEE Trans. on Power Systems, Vol. 10, No. 2, pp. 1077-1084, May 1995.

(8) S.J. Wang, D.S. Kirschen, S. Mokhtari and G.D. Irisarri, "Short-term generation scheduling with transmission and environmental constraints using an augmented-lagrangian relaxation", IEEE Trans. on Power Systems, Vol. 10, No. 3, pp. 1294-1301, August 1995.

(9) A.J. Svoboda, C.L. Tseng, C.A. Li and R.B. Johnson, "Short-term resource scheduling with ramp constraints", IEEE Trans. on Power Systems, Vol. 12, No. 1, pp. 77-83, February 1997.

### 2.3.6 วิธีซิมมูลเตดด์ (Simulated Annealing)

เป็นวิธีการหาค่าที่เหมาะสมแบบสุ่มวิธีหนึ่ง ซึ่งเป็นวิธีที่จำลองมาจากหลักการของการหลอมของโลหะ วิธีนี้มีการนำมาใช้ในการแก้ปัญหาการหาค่าที่เหมาะสมที่เป็นปัญหาแบบ Combinatorial Problem อาทิเช่นปัญหาการเดินทางของเซลส์แมน (Travelling Saleman Problem) วิธี Simulated Annealing เหมาะกับปัญหาที่มีค่าที่เหมาะสมท้องถิ่น (local minima) หรือปัญหาที่มีเงื่อนไขต่างๆ อยู่มาก แต่วิธีนี้จะใช้เวลาในการคำนวณมากและไม่เหมาะสมกับปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าที่มีขนาดใหญ่

บทความที่น่าเสนอการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธี Simulated Annealing มีดังต่อไปนี้

(1) F. Zhuang and F.D. Galiana, " Unit commitment by simulated annealing", IEEE Trans. on Power Systems, Vol. 5, No. 1, pp. 311-318, February 1990.

(2) U.D. Annakkage, T. Numnonda and N.C. Pahalawaththa, "Unit commitment by parallel sumulated annealing", IEE Proc. Part C, Vol. 142, No. 6, November 1995.

### 2.3.7 วิธีเจเนติก (Genetic Algorithm)

เป็นวิธีการหาค่าที่เหมาะสมแบบสุ่มวิธีหนึ่ง โดยอาศัยหลักการจำลองวิธีทางพันธุกรรม ซึ่งวิธีการเชิงพันธุกรรม นี้จะมีการเปลี่ยนแปลงและพัฒนาไปสู่สิ่งที่ดีขึ้น วิธีเจเนติกสามารถนำมาใช้กับปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าได้ โดยสิ่งที่ต้องการทำให้ดีขึ้นคือราคาต้นทุนการผลิตไฟฟ้า วิธีนี้สามารถหาคำตอบที่เหมาะสมสำหรับปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าที่ขนาดเล็กได้ดี แต่อย่างไรก็ตามวิธีนี้ใช้เวลาในการคำนวณมากและยังไม่เหมาะสมกับการแก้ปัญหาขนาดใหญ่

บทความที่นำเสนอการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าด้วยวิธีเจเนติก มีดังต่อไปนี้

(1) D. Dasgupta and D.R. McGregor, "Thermal unit commitment using genetic algorithms", IEE Proc. Part C., Vol. 141, No. 5, pp. 459-465, September 1994.

(2) S.A. Kazarlis, A.G. Bakirtzis and V.Petridis, "A genetic algorithm solution to the unit commitment problem", IEEE Trans. on Power Systems, Vol. 11, No. 1, pp. 83-92, February 1996.

(3) Y. Fukuyama and H.D. Chiang, "A parallel genetic algorithm for generation expansion planning", IEEE Trans. on Power Systems, Vol. 11, No. 2, pp. 955-961, May 1996.

(4) T.T. Maifeld and G.B. Sheble, "Genetic-based unit commitment algorithm", IEEE Trans. on Power Systems, Vol. 11, No. 3, pp. 1359-1369, August 1996.

(5) P.C. Yang, H.T. Yang and C.L. Huang, "Solving the unit commitment problem with an genetic algorithm through a constraint satisfaction technisque", Electric Power System Research-37, pp. 55-65, 1996.

(6) S.O. Orero and M.R. Irving, "Large scale unit commitment using a hybrid genetic algorithm", Electrical Power & Energy System, Vol. 19, No. 1, pp. 45-55, 1997.

(7) H.T. Yang, P.C. Yang and C.L. Huang, "Parallel genetic algorithm approach to solving the unit commitment problem: Implementation on the transputer networks", IEEE Trans. on Power Systems, Vol. 12, No. 2, pp. 661-667, MAY 1997.

จากวิธีการทำยูนิคคอมมิตเมนต์ที่กล่าวมาทั้งหมดนั้น จะพบว่าวิธีการต่าง ๆ มีทั้งข้อดีและข้อเสียแตกต่างกัน ในวิทยานิพนธ์ฉบับนี้จึงเลือกใช้วิธีโปรแกรมเชิงพลวัตในการทำยูนิคคอมมิตเมนต์ เนื่องจากเป็นวิธีที่คำนวณได้ครอบคลุมทุกสถานะทั้งหมดที่เป็นไปได้ โดยรายละเอียดของวิธีการทำยูนิคคอมมิตเมนต์ จะกล่าวโดยละเอียดในบทถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การทำยูนิตคอมมิตเมนต์

ในหลักความจริงการปิด เปิด เครื่องกำเนิดไฟฟ้า จะมีผลกระทบต่อค่าใช้จ่ายในการผลิต เนื่องจากในบางช่วงเวลา เครื่องกำเนิดกระแสไฟฟ้าบางเครื่องอาจไม่ได้ผลิตกระแสไฟฟ้า ซึ่งอาจเป็นระยะเวลาสั้น การเดินเครื่องในลักษณะนี้จะทำให้เสียค่าใช้จ่ายไปโดยเปล่าประโยชน์ ในบางครั้งการปิดเครื่องกำเนิดกระแสไฟฟ้านั้นๆ แล้วเริ่มเดินเครื่องในช่วงเวลาที่เครื่องกำเนิดกระแสไฟฟ้า นั้นได้ผลิตกระแสไฟฟ้า อาจทำให้ค่าใช้จ่ายน้อยกว่ากรณีแรกได้ โดยทั่วไปจะไม่คำนึงถึงยูนิตคอมมิตเมนต์ แต่ถ้าวัดถึงยูนิตคอมมิตเมนต์ จะทำให้ได้ผลประโยชน์สูงสุดตลอดช่วงเวลา 24 ชั่วโมง (1วัน)

เนื่องจากเหตุการณ์ที่เกิดขึ้นในการดำเนินการผลิตกระแสไฟฟ้านั้น เกิดขึ้นได้หลายกรณี ดังนั้นเพื่อให้ได้ การดำเนินการที่เหมาะสมที่สุดในแต่ละชั่วโมงตลอดทั้งวัน ในที่นี้จึงใช้วิธีพลวัต ซึ่งเป็นวิธีที่สามารถแสดงรายละเอียดได้ดี เพื่อช่วยในการหาคำตอบที่เหมาะสมที่สุด โดยมีวิธีการดังนี้

1. กำหนดสภาวะการเดินเครื่องกำเนิดไฟฟ้าทั้งหมดที่เป็นไปได้
2. หาค่าเหมาะสมในการเสนาราคาและกำลังไฟฟ้า ของแต่ละสภาวะในแต่ละชั่วโมง
3. ใช้โปรแกรมพลวัต เพื่อหาเส้นทางที่ทำให้ได้ผลประโยชน์โดยรวมแต่ละวันสูงสุด

#### 3.1 กำหนดสภาวะการเดินเครื่องกำเนิดไฟฟ้าทั้งหมดที่เป็นไปได้

ตัวอย่างเครื่องกำเนิดกระแสไฟฟ้า 4 เครื่อง จะมีสภาวะการทำงานทั้งหมด  $2^4$  ตลอด 24 ชั่วโมง มีเหตุการณ์เป็นไปได้อย่างทั้งหมด  $(2^4)^{24}$  เหตุการณ์

สภาวะ	สถานะเครื่องกำเนิดไฟฟ้า
16	1111
15	1110
14	1101
13	1100
12	1011
11	1010
10	1001
9	1000
8	0111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6	0101
5	0100
4	0011
3	0010
2	0001
1	0000
Unit	1234

ตารางที่ 3.1 ตารางแสดงสถานะเครื่องกำเนิดไฟฟ้า

### 3.2 การหาค่าเหมาะสมในการเสนอราคาและกำลังไฟฟ้าของแต่ละสภาวะการทำงานและผลกำไรที่ได้ในแต่ละชั่วโมง

หาค่าเหมาะสมในการเสนอราคาและกำลังไฟฟ้าที่ควรเสนอของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องที่สภาวะการทำงานหนึ่งๆนั้น สามารถทำได้โดยใช้วิธีจำลองเหตุการณ์แบบมอนติคาร์โลโดยใช้ลักษณะการเปลี่ยนแปลงของจุดตัดระหว่างกราฟการเสนอซื้อและเสนอขาย แต่เนื่องจากการทำยูนิทคอมมิทเมนต์ต้องหาค่าเหมาะสมในการเสนอราคาและกำลังไฟฟ้าของแต่ละสภาวะการทำงานและผลกำไรที่ได้ในแต่ละชั่วโมง เพื่อใช้เป็นข้อมูลในการหาเส้นทางที่ดีที่สุดตลอดทั้งวัน ในโปรแกรมเชิงพลวัต

### 3.3 ใช้โปรแกรมพลวัต เพื่อหาเส้นทางที่ทำให้ได้ผลประโยชน์โดยรวมแต่ละวันสูงสุด

เนื่องจากใน 24 ชั่วโมง สถานะการทำงานของเครื่องกำเนิดไฟฟ้ามีได้มากมาย เพื่อที่จะหาสถานะการทำงานที่ดีที่สุด ที่ทำให้ได้ผลกำไรมากที่สุด โดยรวมตลอดทั้งวัน ในที่นี้จึงใช้โปรแกรมพลวัตเพื่อหาเส้นทางการทำงานที่เหมาะสมที่สุด โดยสถานะที่ดีที่สุดนั้น คือ สถานะที่มีผลกำไรที่สูงสุด

## ขั้นตอนการทำโปรแกรมเชิงพลวัตสรุปได้ดังนี้

- 1) ในแต่ละช่วงเวลาที่กำหนดเราจะเริ่มพิจารณาที่สถานะเวลาหรือชั่วโมงโดยคำนึงถึงเงื่อนไขต่าง ๆ เช่น การหยุดเดินเครื่องเพื่อบำรุงรักษา กำลังผลิตติดตั้งเพียงพอ กับความต้องการของโหลดและกำลังผลิตสำรองหรือไม่ เป็นต้น
- 2) คำนวณหาสถานะที่เป็นไปได้ที่ทำให้ต้นทุนการผลิตรวมมีค่าน้อยที่สุดทำการเก็บสถานะที่ดีที่สุดเหล่านั้นไว้จำนวนหนึ่งแล้วแต่จะกำหนด
- 3) คำนวณหาสถานะที่เป็นไปได้ในช่วงเวลาถัดไปอีกครั้ง ทั้งนี้ต้องคำนึงถึงเงื่อนไขเวลาเดินเครื่องต่ำสุด (Minimum up time) และเงื่อนไขเวลาหยุดเดินเครื่องต่ำสุด (Minimum down time) ด้วย
- 4) คำนวณหาเส้นทางที่ทำให้ค่าใช้จ่ายรวมในสถานะนั้นมีค่าต่ำที่สุด
- 5) เพิ่มการพิจารณาไปอีกหนึ่งคาบเวลาแล้วกลับไปคำนวณในขั้นที่ 1 จนกระทั่งถึงคาบเวลาสุดท้ายที่สนใจ
- 6) เลือกเส้นทางที่ทำให้เสียต้นทุนการผลิตน้อยที่สุดมาเป็นคำตอบสำหรับสมการที่ใช้ในการคำนวณตามขั้นตอนดังกล่าว คือ

$$F_{\text{cost}}(t, I) = \min \left\{ P_{\text{cost}}(t, I) + S_{\text{cost}}(t-1, L; t, I) + F_{\text{cost}}(t-1, L) \right\} \quad (3.1)$$

โดยที่

$$F_{\text{cost}}(t, I) = \text{ค่าใช้จ่ายรวมต่ำสุดจนกระทั่งถึงสถานะ } (t, I)$$

$$P_{\text{cost}}(t, I) = \text{ค่าใช้จ่ายในการผลิตไฟฟ้าสำหรับสถานะ } (t, I) \text{ หาได้โดยทำการจ่าย}$$

โหลด

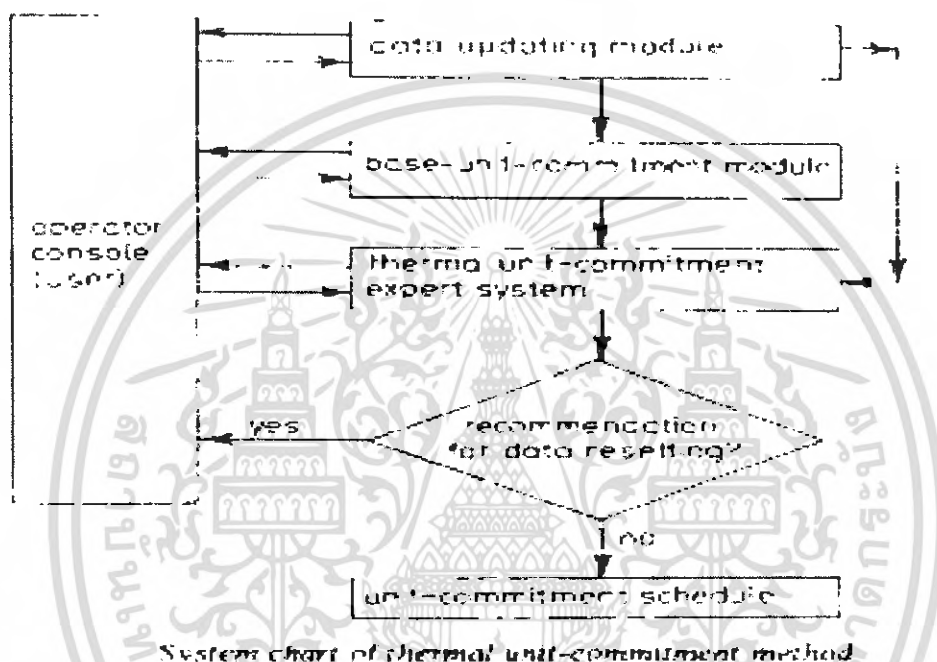
อย่างประหยัด (Economic dispatch)

$$S_{\text{cost}}(t-1, L; t, I) = \text{ค่าใช้จ่ายในการเปลี่ยนสถานะจากสถานะ } (t-1, L) \text{ ไปยังสถานะ } (t, I)$$

ทั้งนี้สถานะ  $(t, I)$  ก็คือสถานะลำดับที่  $I$  ในคาบเวลา  $t$  และสถานะ  $(t-1, L)$  คือสถานะลำดับที่  $L$  ในคาบเวลา  $t-1$  วิธีการคำนวณแบบนี้จะพิจารณาช่วงเวลทั้งหมด โดยเริ่มจากคาบเวลาแรกถึงคาบเวลาสุดท้าย (Forward dynamic programming) หรือเริ่มจากคาบเวลาสุดท้ายถึงคาบเวลาแรก (Backward dynamic programming) สำหรับวิธีbackward dynamic-programming นั้นไม่ได้ครอบคลุมในหลายๆสถานการณ์ เช่น ถ้าราคาการเริ่มเดินเครื่องกำเนิดไฟฟ้าและวิธีforward dynamic-programmingจะเป็นวิธีที่เหมาะสมกว่าเพราะประวัติของครั้ง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

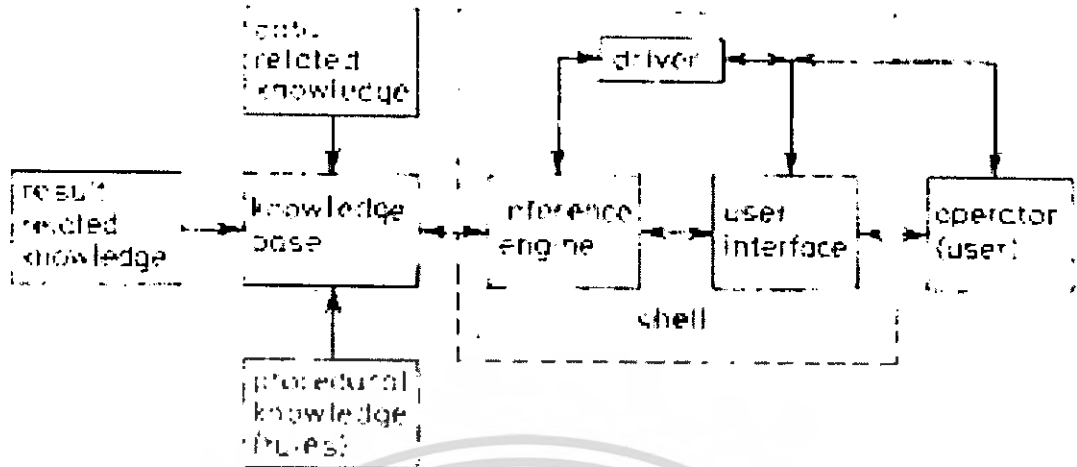
ก่อนๆสามารถนำมาคำนวณในแต่ละระดับ(stage) ได้ทำให้สามารถในเวลานานเท่าที่ต้องการได้เท่าที่คอมพิวเตอร์จะจัดเก็บได้ ในระหว่างที่ดำเนินการนี้สามารถพิจารณาข้อจำกัดต่าง ๆ ได้ เช่น เงื่อนไขปริมาณเชื้อเพลิง เงื่อนไขการซ่อมบำรุงรักษา เป็นต้น



รูปที่ 3.1 รูปแสดง system chart of thermal unit-commitment method

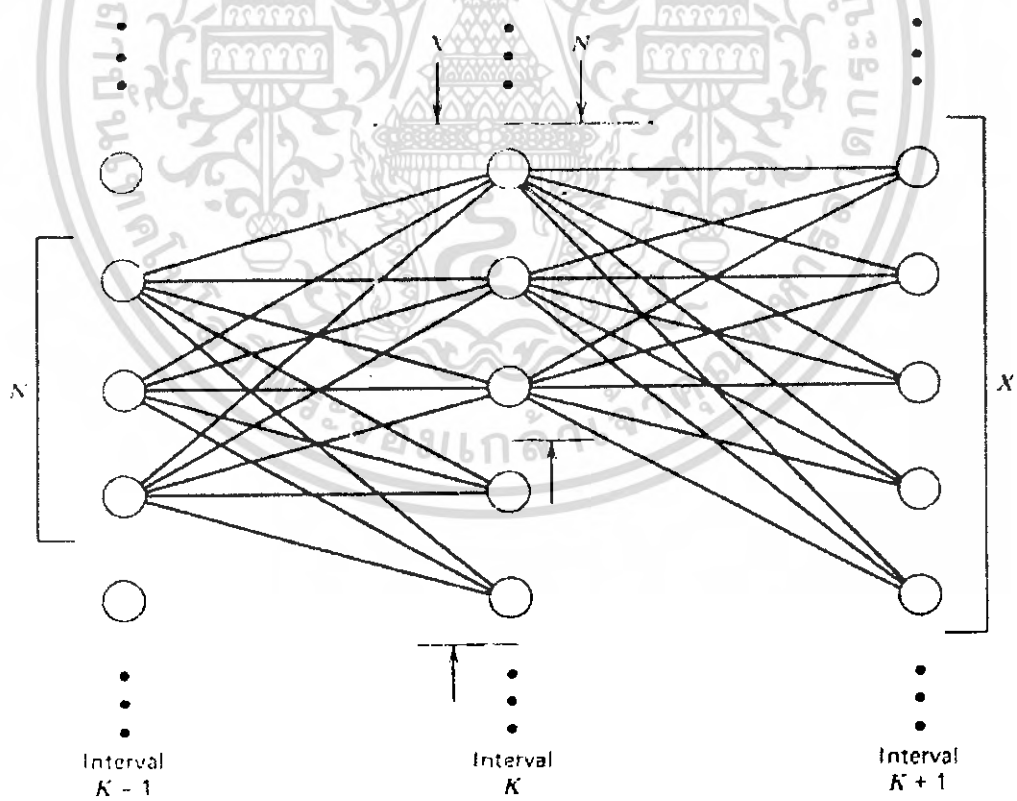
ผลการคำนวณทำได้ ค่าราคาไฟฟ้าและกำลังไฟฟ้าที่ควรเสนอของเครื่องกำเนิดไฟฟ้าแต่ละเครื่อง และสถานะการเดินเครื่องกำเนิดไฟฟ้า ของแต่ละชั่วโมงตลอดช่วงเวลาที่พิจารณา เพื่อนำไปใช้ในการคาดคะเนล่วงหน้ารายวันเพื่อให้ได้ผลประโยชน์สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Structure of thermal unit-commitment expert-system module

รูปที่ 3.2 structure of thermal unit-commitment system module

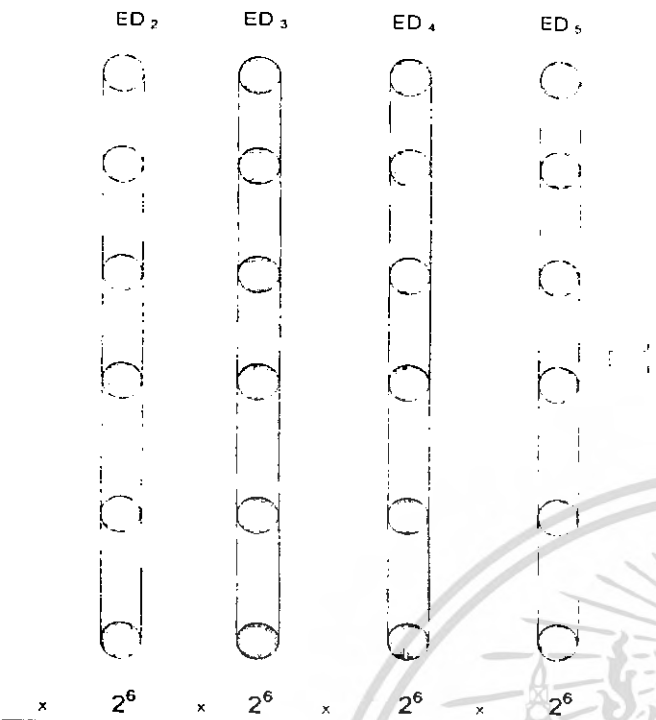


รูปที่ 3.3 เส้นทางที่เป็นไปได้ในการคำนวณโดยใช้วิธีโปรแกรมเชิงพลวัต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

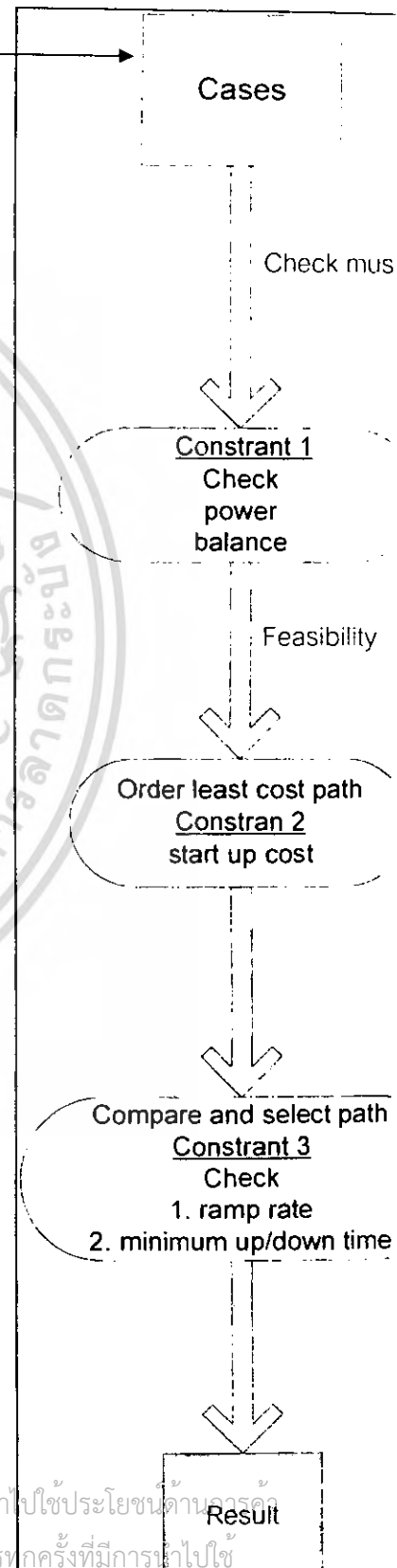
### 3.4 ปัญหายูนิคคอมมิตเมนต์

แผนการผลิตกำลังไฟฟ้าที่มุ่งเน้นจัดสรรสถานการณ์ทำงานของเครื่องกำเนิดไฟฟ้า ที่มีอยู่ในระบบไฟฟ้าทั้งหมด ว่าเครื่องกำเนิดไฟฟ้าเครื่องใดควรทำงานต่ออยู่กับระบบไฟฟ้าหรือปลดออกจากระบบไฟฟ้า เพื่อทำการผลิตกำลังไฟฟ้าให้เพียงพอกับความต้องการของโหลดที่เปลี่ยนแปลงตลอดเวลาตามแต่กิจกรรมที่เกิดขึ้น และระบบไฟฟ้ายังมีความมั่นคงโดยที่ต้นทุนการผลิตกำลังไฟฟ้าวรมมีค่าต่ำที่สุด การแก้ปัญหาดังกล่าวนี้เรียกว่า ปัญหายูนิคคอมมิตเมนต์ ในวิทยานิพนธ์ฉบับนี้เลือกช่วงระยะเวลาพิจารณาแผนผลิตกำลังไฟฟ้าไว้ที่ 24 ชั่วโมง และคำนึงถึงความมั่นคง (Security) ของระบบไฟฟ้าด้วยการคำนวณค่ากำลังไฟฟ้าสำรองไว้ที่ 10 เปอร์เซ็นต์ ของความต้องการกำลังไฟฟ้า ดังนั้นเมื่อพิจารณาเฉพาะเจาะจงลงไปในช่วงเวลาหนึ่งๆ ที่ทำการแก้ปัญหายูนิคคอมมิตเมนต์ จะเป็นการแก้ปัญหาการจ่ายโหลดอย่างประหยัด สมมติว่าช่วงเวลาหนึ่งๆ ที่พิจารณาเราทราบว่าเครื่องกำเนิดไฟฟ้าเครื่องใดต่ออยู่กับระบบ จากนั้นเราจะแก้ปัญหาว่าทำอะไรให้ได้ต้นทุนการผลิตกำลังไฟฟ้าวรม ณ ขณะนั้นมีราคาต่ำที่สุด โดยที่โหลด ณ ช่วงเวลาที่พิจารณาต้องได้รับกำลังไฟฟ้าที่เพียงพอกับความต้องการ ลักษณะการแก้ปัญหาแบบนี้เราจึงกล่าวว่า ปัญหาการจ่ายโหลดอย่างประหยัดเป็นปัญหาย่อยของปัญหายูนิคคอมมิตเมนต์ที่ต้องเลือกทำการศึกษาวิจัยภายในวิทยานิพนธ์ฉบับนี้ ตลอดจนอธิบายถึงเชื่อมโยงของระบบไฟฟ้าและเครื่องกำเนิดไฟฟ้าที่จำเป็นจะต้องพิจารณาขณะทำการแก้ไขปัญหายูนิคคอมมิตเมนต์ ในหัวข้อถัดไปกล่าวถึงขั้นตอนการแก้ปัญหายูนิคคอมมิตเมนต์ ด้วยวิธีพลวัต ซึ่งเมื่ออธิบายถึงเนื้อหาที่เกี่ยวข้องกับการแก้ปัญหการจ่ายโหลดอย่างประหยัด ผู้วิจัยจะไม่ขออธิบายในรายละเอียดเชิงลึกของขั้นตอนการแก้ปัญหการจ่ายโหลดอย่างประหยัด ด้วยวิธีจุดภายในพรีมัล-ดวลล์ หรืออื่นๆ แต่ขอเน้นหนักไปถึงกระบวนการนำเอาสถานะของแผนการเครื่องกำเนิดไฟฟ้า ที่ค้นหาได้จากวิธีพลวัต ไปแทนค่าในการแก้ปัญหการจ่ายโหลดอย่างประหยัดด้วยวิธีที่นำเสนอแทน



Output (total cases)

- จากกรณีที่ไม่คิด must run จำนวน stage =  $(2^{n-1})$  period
- จากกรณีที่คิด must run จำนวน stage =  $(2^{(n-\text{must run})}-1)$  period
- กรณีคิดโดยไม่มีการตัด must run ของระบบที่มีจำนวน Gen ทั้งหมด 6 ตัว
- จะได้จำนวน stage ที่จะต้องทำการเปรียบเทียบ  $\cdot (2^6-1)^6$  stage หรือเท่ากับ 992,436,543 stage
- เพราะฉะนั้นถ้าในระบบคิดโดยไม่มีการตัด must run ของระบบที่มีจำนวน Gen ทั้งหมด 6 ตัว 12 stage
- จะได้จำนวน stage ที่จะต้องทำการเปรียบเทียบ  $\cdot (2^6-1)^{12}$  stage หรือเท่ากับ  $3.909188 \times 10^{21}$  stage
- ส่วนจำนวนกรณีทั้งหมดที่คิด must run แล้ว โดยกำหนดให้ must run ที่ gen 2 ตัวในระบบ
- จะได้จำนวน stage ที่จะต้องทำการเปรียบเทียบ  $\cdot (2^{(6-2)}-1)^{12}$  stage หรือเท่ากับ  $1.297 \times 10^{14}$  stage



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นใด  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 รูปแบบทางคณิตศาสตร์ของปัญหาชนิดคอมมิตเมนต์

จากที่กล่าวมาถึงการแก้ปัญหาชนิดคอมมิตเมนต์ไปในช่วงบทนำไปแล้ว ในหัวข้อนี้ขออธิบายอย่างละเอียดดังนี้ การแก้ปัญหา ชนิดคอมมิตเมนต์เป็นการวางแผนการเดินทางเครื่องกำเนิดไฟฟ้าที่มีอยู่ในระบบไฟฟ้าทั้งหมด ว่าควรเดินเครื่องกำเนิดเครื่องกำเนิดไฟฟ้าเข้ากับระบบไฟฟ้าหรือปลดเครื่องกำเนิดไฟฟ้าเครื่องกำเนิดไฟฟ้าเครื่องใดออกจากระบบไฟฟ้า เพื่อให้สามารถผลิตกำลังไฟฟ้าหรือปลดเครื่องกำเนิดไฟฟ้าเครื่องใดออกจากระบบไฟฟ้า เพื่อให้สามารถผลิตกำลังไฟฟ้าได้สอดคล้องกับเงื่อนไขทั้งระบบไฟฟ้าและเครื่องกำเนิดไฟฟ้าเอง ตลอดช่วงเวลาที่พิจารณา ดังนั้นสมมติว่าเราทราบแผนการเดินทางเครื่องกำเนิดไฟฟ้า ณ ชั่วโมงที่ 1 เราจะนำสถานะแผนการเดินทางเครื่องกำเนิดไฟฟ้างกล่าวมาคำนวณแก้ปัญหาการจ่ายโหลดอย่างประหยัด ไปจนถึงเวลาที่ 24 นาฬิกา โดยเราจะนำเอาต้นทุนในการผลิตไฟฟ้าในแต่ละช่วงเวลามารวมกัน แต่ต้นทุนดังกล่าวยังไม่ใช้ต้นทุนการผลิตไฟฟ้ารวมที่แท้จริง เพราะแต่ละช่วงเวลาชั่วโมงใดก็ได้จะเกิดการปลดเครื่องกำเนิดไฟฟ้าออกจากระบบไฟฟ้า หรือเริ่มเดินเครื่องกำเนิดไฟฟ้าที่ถูกปลดออกในชั่วโมงก่อนหน้านี้เข้ากับระบบไฟฟ้างั้นเมื่อเกิดการเริ่มเดินเครื่องกำเนิดไฟฟ้าเข้ากับระบบไฟฟ้าใหม่ ด้วยธรรมชาติของเครื่องกำเนิดไฟฟ้า พลังงานความร้อนจะต้องทำการ เพิ่มปริมาณความร้อน เพื่อให้สามารถผลิตกำลังไอน้ำได้ตามต้องการ การกระทำนี้จะต้องสูญเสียเชื้อเพลิงไปซึ่งเราสามารถแปรความสัมพันธ์ของปริมาณเชื้อเพลิงที่สูญเสีย เพื่อผลิตกำลังไฟฟ้าออกมาอยู่ในรูปของค่าใช้จ่ายการเริ่มเดินเครื่องกำเนิดไฟฟ้า ดังนั้นเราต้องรวมค่าใช้จ่ายนี้เข้ากับการเริ่มเดินเครื่องกำเนิดไฟฟ้าขึ้นมาใหม่จึงเรียกดัชนีต้นทุนในการผลิตกำลังไฟฟ้ารวมไฟฟ้ารวมอย่างแท้จริง จากคำอธิบายการแก้ปัญหาชนิดคอมมิตเมนต์สามารถแทนด้วยความสัมพันธ์ทาง คณิตศาสตร์ได้ ดังต่อไปนี้

$$F_{total\ cost} = \sum_{t=1}^T \sum_{i=1}^N [U_{ii} F_i(p_{ii}) + U_{ii} (1 - U_{ii(t-1)}) ST_i] \quad (3.2)$$

เมื่อ  $T$  คือ จำนวนชั่วโมงที่พิจารณา

$N$  คือ จำนวนเครื่องกำเนิดไฟฟ้า

$U_{ii}$  คือ สถานะของเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  ณ เวลาที่  $t$

เมื่อ  $U_{ii} = 0$  แทนสถานะที่หยุดเดินเครื่องและ  $U_{ii} = 1$  แทนสถานะเดินเครื่องกำเนิดไฟฟ้า

$F_i(p_{ii})$  คือ ฟังก์ชันต้นทุนเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  ขณะผลิตกำลังไฟฟ้าขนาด  $P$  ที่เวลา  $t$

$ST_i$  คือ ต้นทุนการเริ่มเดินเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$

$p_{ii}$  คือ กำลังไฟฟ้าที่ผลิตจากเครื่องกำเนิดไฟฟ้าเครื่องที่  $t$  ณ เวลาที่  $t$

สมการที่ (3.2) ประกอบด้วยส่วนสำคัญ 2 ส่วนแรกเป็นส่วนที่ต้นทุนการในการผลิตกำลังไฟฟ้าที่ได้จากการแก้ปัญหาการจ่ายโหลดอย่างประหยัด กล่าวคือ เราจะนำเอาสถานะของเครื่องกำเนิดไฟฟ้าที่มีสถานะต้องต่อเข้ากับระบบไฟฟ้า  $U_{ii} = 1$  ไปแทนเพื่อคำนวณค่า

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของสถาบันวิจัยพลังงานจันทบุรี หากมีการนำ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลังไฟฟ้าที่ผลิตได้จากเครื่องกำเนิดไฟฟ้าทั้งหมดที่ต่ออยู่กับระบบไฟฟ้า ณ เวลาที่พิจารณารายนี้โดยสอดคล้องกับเงื่อนไขของปัญหาการจ่ายโหลดอย่างประหยัด พิจารณาตั้งแต่  $t=1$  ถึง  $t=24$  ส่วนที่สองเป็นค่าใช้จ่ายจากการเริ่มต้นเครื่องกำเนิดไฟฟ้าในชั่วโมงที่จะต้องนำเครื่องกำเนิดไฟฟ้าเข้ามาต่ออย่างกับระบบไฟฟ้า

การแก้ปัญหาชนิดคอมิตเมนต์ที่มุ่งค้นหาแผนการเดินทางเครื่องกำเนิดไฟฟ้าที่ใช้ต้นทุนในการผลิตกำลังไฟฟ้าที่ต่ำที่สุด ซึ่ง แผนการเดินทางเครื่องกำเนิดไฟฟ้างกล่าวต้องสอดคล้องกับเงื่อนไขทั้งของระบบไฟฟ้าและเครื่องกำเนิดไฟฟ้าอีกด้วย โดยในการศึกษาวิจัยครั้งนี้ พิจารณาครอบคลุมถึงเงื่อนไขดังต่อไปนี้

3.5.1 เงื่อนไขของระบบการแก้ปัญหาชนิดคอมิตเมนต์ ประกอบไปด้วยเงื่อนไขดังนี้ ความสมดุลกำลังไฟฟ้า (Power Balance) และกำลังผลิตไฟฟ้าสำรอง (Spinning Reserve)

3.5.1.1 ความสมดุลกำลังไฟฟ้ามีเงื่อนไขว่า กำลังผลิตไฟฟารวมของเครื่องกำเนิดไฟฟ้าที่ต่อเข้ากับระบบไฟฟ้าระยะเวลาที่พิจารณา ต้องมีค่าเท่ากับกำลังผลิตไฟฟ้าที่โหลดต้องการในเวลานั้น ซึ่งก็คือเงื่อนไขของปัญหาการจ่ายโหลดอย่างประหยัดนั่นเอง

$$\sum_{t=1}^T \sum_{i=1}^N U_{it} P_{it} = D_t \quad (3.12)$$

เมื่อ  $D_t$  คือ ค่ากำลังไฟฟ้าที่โหลดต้องการ ณ เวลาที่  $t$  จากนั้นเงื่อนไขที่ส่งผลกับความมั่นคงของระบบไฟฟ้าจะกล่าวถึงในหัวข้อถัดไปดังนี้

3.5.1.2 กำลังผลิตไฟฟ้าสำรอง กำลังผลิตไฟฟ้าสำรองมีจุดมุ่งหมายเพื่อเป็นหลักประกันว่า เมื่อเกิดการเปลี่ยนแปลงภายในระบบกำลังไฟฟ้า ระบบยังคงจ่ายไฟต่อไปได้ ซึ่งประกันการเปลี่ยนแปลง 2กรณี คือ กรณีแรกเกิดการปลดของเครื่องกำเนิดเครื่องใดเครื่องหนึ่ง อันเนื่องมาจากสาเหตุใดก็ตามตลอดในระบบไฟฟ้ายังคงต้องได้รับกำลังที่เพียงพอกับความ ต้องการเหมือนเดิมกับ กรณีที่สอง เกิดการเปลี่ยนแปลงของความต้องการไฟฟ้าของโหลดมากขึ้น ดังนั้นเครื่องกำเนิดไฟฟ้าขณะนั้นต้องสามารถเพิ่มกำลังผลิตไฟฟ้าได้เพียงพอกับความ ต้องการซึ่งพิจารณาเงื่อนไขกำลังไฟฟ้าสำรองนี้จะใช้เป็นหลักประกันเท่านั้น ถ้าเกิดการเปลี่ยนแปลงจริงจะต้อง จริงจะต้องแยกไปพิจารณาเฉพาะอีกครั้งหนึ่ง ซึ่งอยู่นอกเหนือขอบเขตการวิจัยนี้

$$\sum_{t=1}^T \sum_{i=1}^N U_{it} P_{i,\max} \geq R_t + D_t \quad (3.13)$$

เมื่อ  $R_t$  คือ ค่ากำลังผลิตไฟฟ้าสำรอง ณ เวลาที่  $t$

$P_{i,\max}$  คือ ค่ากำลังไฟฟ้าสูงสุดที่เครื่องกำเนิดไฟฟ้า เครื่องที่  $i$  สามารถผลิตได้

ดังนั้นกำลังไฟฟารวมที่ผลิตได้ในแต่ละชั่วโมงต้องมีค่าเท่ากับความต้องการของโหลด อยู่ นั้นหมายถึงว่าเงื่อนไขกำลังไฟฟารวมในสมการที่ (3.12) ต้องเป็นจริงเสมอ ซึ่งเราสามารถทำการตรวจสอบว่าการแก้ปัญหาชนิดคอมิตเมนต์ที่พิจารณาเงื่อนไขกำลังไฟฟ้าสำรอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแก้ไขปัญหาก็ได้ถูกต้องหรือไม่ด้วยขั้นตอนดังต่อไปนี้ ให้ใส่ค่ากำลังไฟฟ้าสำรองแต่ละ ชั่วโมงเท่ากับศูนย์ ตลอดช่วงเวลาที่พิจารณาแก้ปัญหาหน่วยผลิตคอมมิตเมนต์ เมื่อทำการแก้ปัญหา หน่วยผลิตคอมมิตเมนต์เสร็จสิ้นแล้วต้นทุนในการผลิตกำลังไฟฟารวม ต้องมีค่าเดียวกันกับต้นทุนในการผลิตไฟฟารวมที่ยังไม่พิจารณาถึงเงื่อนไขกำลังไฟฟ้าสำรอง

3.5.2 เงื่อนไขของเครื่องกำเนิดไฟฟ้าในการแก้ปัญหาหน่วยผลิตคอมมิตเมนต์ หัวข้อย่อยนี้ กล่าวถึงเงื่อนไขเฉพาะของเครื่องกำเนิดไฟฟ้าที่เกี่ยวข้องกับการผลิตกำลังไฟฟ้า ประกอบไปด้วยเงื่อนไขดังต่อไปนี้ ขีดจำกัดกำลังการผลิตของเครื่องกำเนิดไฟฟ้า (Generating Capacity) เวลาเดินเครื่องอย่างน้อยที่สุด (Minimum Up Time) และเวลาเดินเครื่องอย่างน้อยที่สุดสุด (Minimum Down Time)

3.5.2.1 เวลาเดินเครื่องอย่างน้อยที่สุด เป็นเงื่อนไขเชิงปฏิบัติ การเดินเครื่องกำเนิดไฟฟ้า กล่าวคือ เครื่องกำเนิดไฟฟ้าที่ทำการเดินเครื่องกำลังไฟฟ้าอยู่นั้น ไม่สามารถที่จะสั่งหยุดเดินเครื่องได้ทันทีทันใดต้องอาศัยช่วงเวลาระยะหนึ่ง ซึ่งโดยปกติแล้วมีหน่วยเป็นชั่วโมง และช่วงระยะเวลาดังกล่าวของเครื่องกำเนิดไฟฟ้าแต่ละเครื่องอาจมีค่าไม่เท่ากัน ดังนั้นการที่เครื่องกำเนิดเครื่องใดก็ตามจะหยุดเครื่อง จะต้องตรวจสอบว่าเครื่องกำเนิดไฟฟาดังกล่าวได้เดินเครื่องมาอย่างน้อยเท่ากับเวลาเดินเครื่องอย่างน้อยที่สุดแล้วหรือยัง ถ้าใช่ ถึงจะสามารถหยุดเดินเครื่องในชั่วโมงที่พิจารณาได้ จากพฤติกรรมในการเดินเครื่องกำเนิดไฟฟาดังกล่าวสามารถนำมาเขียนให้อยู่ในลักษณะสมการทางคณิตศาสตร์ ได้ดังนี้

$$(X_{i(t-1)}^{on} - \min\_up_i) * (U_{i(t-1)} - U_{it}) \geq 0 \quad (3.14)$$

เมื่อ  $X_{i(t-1)}^{on}$  คือ จำนวนที่เครื่องกำเนิดไฟฟ้า เครื่องที่  $i$  เดินเครื่องกำเนิดเป็นเวลาติดต่อกันถึง  $(t-1)$

$\min\_up_i$  คือ เวลาเดินเครื่องอย่างน้อยที่สุดของเครื่องกำเนิดไฟฟ้า เครื่องที่  $i$

3.5.2.2 เวลาหยุดเดินเครื่องน้อยที่สุด เป็นเงื่อนไขในเชิงปฏิบัติการของการหยุดเดินเครื่องกำเนิดไฟฟ้า กล่าวคือ เครื่องกำเนิดไฟฟ้าที่หยุดเดินเครื่องอยู่นั้น ไม่สามารถที่จะสั่งให้เดินเครื่องอย่างทันทีทันใดเพื่อผลิตกำลังไฟฟ้าได้ ต้องอาศัยช่วงเวลาระยะหนึ่ง ซึ่งปกติแล้วมีหน่วยเป็นชั่วโมง แบบช่วงระยะเวลาดังกล่าวของเครื่องกำเนิดไฟฟ้าแต่ละเครื่อง อาจจะมีค่าไม่เท่ากันดังนั้นการที่เครื่องกำเนิดใดก็ตามจะเดินเครื่อง จะต้องตรวจสอบว่าเครื่องกำเนิดไฟฟ้าเครื่องดังกล่าวได้หยุดเดินเครื่องมาอย่างน้อยเท่ากับเวลาหยุดเดินอย่างน้อยที่สุดแล้วหรือยัง ถ้าใช่ถึงจะสามารถเดินเครื่องเข้ากับระบบไฟฟ้าได้ในชั่วโมงที่พิจารณาอยู่ จากพฤติกรรมในการหยุดเดินเครื่องกำเนิดไฟฟาดังกล่าวสามารถนำมาเขียนให้อยู่ในลักษณะสมการทางคณิตศาสตร์ดังนี้

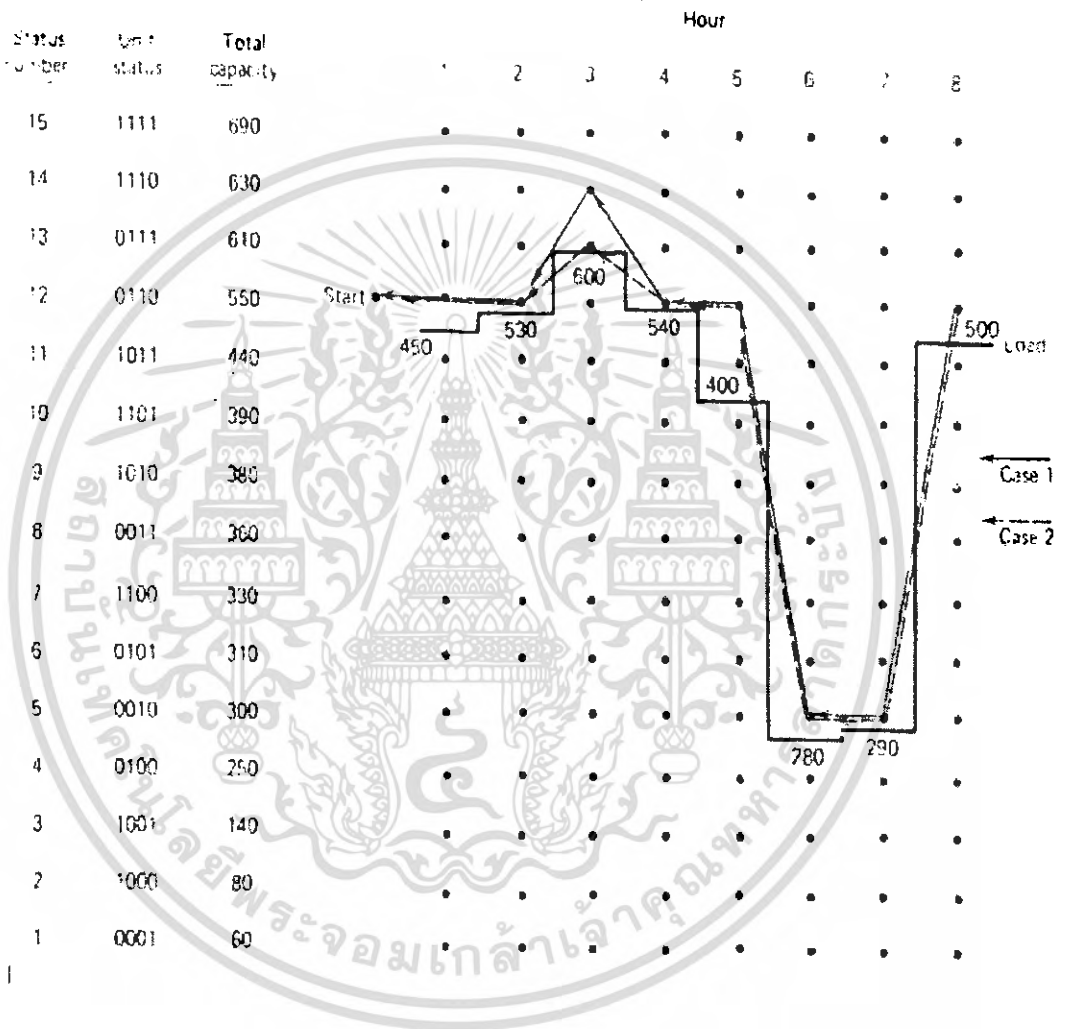
$$(X_{i(t-1)}^{off} - \min\_dw_i) * (U_{i(t-1)} - U_{it}) \geq 0 \quad (3.15)$$

เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$X_{(i-1)}'''$  คือ ระยะเวลาที่เครื่องกำเนิดไฟฟ้าเครื่องที่  $i$  หยุดเดินเครื่องเป็นเวลาติดต่อกันถึง  $(t-1)$

$\min\_dwi$  คือ เวลาหยุดเดินเครื่องอย่างน้อยที่สุดของเครื่องกำเนิดไฟฟ้าเครื่องที่  $i$



รูปที่ 3.4 แสดงเส้นทางเดินของ unit-commitment โดยใช้วิธีโปรแกรมเชิงพลวัต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

เพื่อเป็นการทดสอบการแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าอย่างประหยัดด้วยวิธีโปรแกรมเชิงพลวัต (Dynamic programming) ผู้วิจัยได้ทำการพัฒนาโปรแกรมภาษาซีบนระบบปฏิบัติการ Microsoft Windows Xp โดยใช้โปรแกรม Visual Basic 6 โปรแกรมที่สร้างขึ้นสามารถแก้ปัญหาการตัดสินใจจ่ายกระแสไฟฟ้าอย่างประหยัดที่มีจำนวนเครื่องกำเนิดไฟฟ้าหลายชนิด และกำหนดเงื่อนไขต่าง ๆ ได้ ผู้วิจัยได้ทำการทดลองแก้ปัญหา Unit Commitment โดยวิธีโปรแกรมเชิงพลวัต (Dynamic programming) ซึ่งเป็นการทดลองกับการแก้ปัญหาในระบบที่มีเครื่องกำเนิดไฟฟ้าชนิดต่าง ๆ โดยแบ่งการทดลองออกเป็น 4 กรณี กรณีแรกคือ Unit Commitment ที่คาบเวลาทุก 4 ชั่วโมง ( 6 คาบเวลา ) ใช้เครื่องกำเนิดไฟฟ้า 4 เครื่อง และกรณีที่สองคือ Unit Commitment ที่คาบเวลาทุก 4 ชั่วโมง ( 6 คาบเวลา ) ใช้เครื่องกำเนิดไฟฟ้า 5 เครื่อง ในช่วงเวลา 1 วัน

#### กรณีที่ 1

กรณีที่ 1 กรณีนี้เป็นการแก้ปัญหาสำหรับระบบที่มีเครื่องกำเนิดไฟฟ้าจำนวน 4 เครื่อง ในช่วงเวลา 24 ชั่วโมง โดยแบ่งคาบเวลาออกเป็นทุกๆ 4 ชั่วโมง ซึ่งคุณลักษณะของเครื่องกำเนิดไฟฟ้าในกรณีนี้ได้แสดงไว้ในตารางที่ 4.2 และ ตารางที่ 4.4

#### กรณีที่ 2

กรณีที่ 2 กรณีนี้จะเป็นการแก้ปัญหาของระบบที่มีข้อมูลของเครื่องกำเนิดไฟฟ้า และความต้องการกำลังไฟฟ้าของระบบเช่นเดียวกับกรณีแรก แต่ในกรณีนี้จะเปลี่ยนจำนวนเครื่องกำเนิดไฟฟ้าเป็น 5 เครื่อง โดยใช้คาบเวลาเป็นทุกๆ 4 ชั่วโมง เท่ากับกรณีแรก ซึ่งข้อมูลดังกล่าวนี้แสดงในตารางที่ 4.3 และตารางที่ 4.5

### กรณีที่ 3

กรณีที่ 3 กรณีนี้เป็นการแก้ปัญหาสำหรับระบบที่มีเครื่องกำเนิดไฟฟ้าจำนวน 4 เครื่อง ในช่วงเวลา 24 ชั่วโมง โดยแบ่งคาบเวลาออกเป็นทุกๆ 2 ชั่วโมง ซึ่งคุณลักษณะของเครื่องกำเนิดไฟฟ้าในกรณีนี้ได้แสดงไว้ในตารางที่ 4.6 และ ตารางที่ 4.8

### กรณีที่ 4

กรณีที่ 4 กรณีนี้จะเป็นการแก้ปัญหาของระบบที่มีข้อมูลของเครื่องกำเนิดไฟฟ้า และ ความต้องการกำลังไฟฟ้าของระบบเช่นเดียวกับกรณีที่ 3 แต่ในกรณีนี้จะเปลี่ยนจำนวนเครื่องกำเนิดไฟฟ้าเป็น 5 เครื่อง โดยใช้คาบเวลาเป็นทุกๆ 2 ชั่วโมง ซึ่งคุณลักษณะของเครื่องกำเนิดไฟฟ้าในกรณีนี้ได้แสดงไว้ในตารางที่ 4.7 และ ตารางที่ 4.9

ตาราง 4.1 แสดงผลการคำนวณการจ่ายโหลดอย่างประหยัด ( economic dispatch ) แบ่งคาบเวลาออกเป็นทุกๆ 4 ชั่วโมง เครื่องกำเนิดไฟฟ้า 4 เครื่อง

Stage 1		LOAD 1100												
สถานะ	การทำงาน	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่าใช้จํานวน		
1	2	3	4	lamda										
1	1	0	0	12.07	0.0043	7.27	509.09	590.9	0	0	5609.42	5857.84	0	45869.08
1	0	1	0	12.84	0.0044	7.95	605.55	0	494.44	0	6811.23	0	5728.48	50158.87
0	1	1	0	12.52	0.0048	7.13	0	637.75	462.24	0	0	6433.94	5320.08	47016.11
1	1	1	0	10.8	0.003	7.46	350.63	458.86	290.5	0	3796.83	4347.36	3316.97	45844.69
1	0	0	1	12.88	0.0046	7.78	610.52	0	0	489.47	6875.18	0	5388.76	49055.79
0	1	0	1	12.56	0.0051	6.91	0	640.77	0	459.22	0	6471.82	0	5004.04
1	1	0	1	10.77	0.0031	7.33	346.74	455.62	0	297.63	3754.89	4312.41	0	3119.46
0	0	1	1	13.47	0.0052	7.7	0	0	557.14	542.85	0	0	6553.46	6092.24
1	0	1	1	11.3	0.0031	7.82	413.3	0	340.64	346.04	4489.77	0	3871.32	3653.92
0	1	1	1	10.97	0.0033	7.24	0	476.57	307.5	315.91	0	4540.24	3502.13	45442.75
1	1	1	1	10.08	0.0023	7.47	261.22	384.35	218.98	235.43	2862.76	3568.97	2569.71	2470.64

Stage 2		LOAD 1400												
สถานะ	การทำงาน	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่าใช้จํานวน		
1	2	3	4	lamda										
1	1	1	0	11.72	0.003	7.46	464.55	553.79	381.64	0	5079.71	5416.42	4343.26	0
1	1	0	1	11.71	0.0031	7.33	463.9	553.25	0	382.84	5072.07	5410.06	0	4077.41
1	0	1	1	12.25	0.0031	7.82	532.01	0	435.61	432.37	5888.27	0	4990.11	4671.01
0	1	1	1	11.99	0.0033	7.24	0	582.47	409.17	408.34	0	5756.4	4669.64	4379.67
1	1	1	1	10.8	0.0023	7.47	350.49	458.74	290.39	300.36	3795.35	4346.13	3315.78	3148.89

เอกสารนี้เป็นเอกสารการศึกษเท่านั้น ไม่ควรนำไปใช้โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 3		LOAD 1600													
1	2	3	4	lamda	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่าใช้จํา
1	1	1	0	12.32	0.003	7.46	540.5	617.08	442.4	0	5992.64	6177.2	5073.61	0	68973.81
1	1	0	1	12.33	0.0031	7.33	542.01	618.34	0	439.64	6011.19	6192.66	0	4760.41	67857.13
1	0	1	1	12.88	0.0031	7.82	611.15	0	489.92	489.92	6883.23	0	5786.08	5394.62	72255.78
0	1	1	1	12.66	0.0033	7.24	0	653.08	476.95	469.96	0	6626.98	5505.4	5139.46	69087.42
1	1	1	1	11.28	0.0023	7.47	410.01	508.34	338	343.64	4452.5	4893.75	3841.5	3626.81	67258.29

Stage 4		LOAD 1800													
1	2	3	4	lamda	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่าใช้จํา
1	1	1	0	12.93	0.003	7.46	616.45	680.37	503.16	0	6951.71	6976.43	5840.87	0	79076.09
1	1	1	1	11.75	0.0023	7.47	469.52	557.93	385.61	386.92	5137.98	5464.98	4389.88	4125.34	76472.79

เอกสารนี้เป็นเอกสารของบริษัทฯ ใช้สำหรับประกอบการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่สามารถนำ  
 ไปทำกำไรได้ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากบริษัทฯ ถือว่าผิดกฎหมาย  
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 5		LOAD 1400														
สถานะ	ค่าทำงาน	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่ารวม				
1 2 3 4	lamda											ค่ารวม				
1 1 1 0	11.72	0.003	7.46	464.55	553.79	381.64	0	5079.71	5416.42	4343.26	0	59357.61				
1 1 0 1	11.71	0.0031	7.33	463.9	553.25	0	382.84	5072.07	5410.06	0	4077.41	58238.2				
1 0 1 1	12.25	0.0031	7.82	532.01	0	435.61	432.37	5888.27	0	4990.11	4671.01	62197.64				
0 1 1 1	11.99	0.0033	7.24	0	582.47	409.17	408.34	0	5756.4	4669.64	4379.67	59222.87				
1 1 1 1	10.8	0.0023	7.47	350.49	458.74	290.39	300.36	3795.35	4346.13	3315.78	3148.89	58424.68				

Stage 6		LOAD 1100														
สถานะ	ค่าทำงาน	aT	bT	p1	p2	p3	p4	F1	F2	F3	F4	ค่ารวม				
1 2 3 4	lamda											ค่ารวม				
1 1 0 0	12.07	0.0043	7.27	509.09	590.9	0	0	5609.42	5857.84	0	0	45869.08				
1 0 1 0	12.84	0.0044	7.95	605.55	0	494.44	0	6811.23	0	5728.48	0	50158.87				
0 1 1 0	12.52	0.0048	7.13	0	637.75	462.24	0	0	6433.94	5320.08	0	47016.11				
1 1 1 0	10.8	0.003	7.46	350.63	458.86	290.5	0	3796.83	4347.36	3316.97	0	45844.69				
1 0 0 1	12.88	0.0046	7.78	610.52	0	0	489.47	6875.18	0	0	5388.76	49055.79				
0 1 0 1	12.55	0.0051	6.91	0	640.77	0	459.22	0	6471.82	0	5004.04	45903.5				
1 1 0 1	10.77	0.0031	7.33	346.74	455.62	0	297.63	3754.89	4312.41	0	3119.46	44747.08				
0 0 1 1	13.47	0.0052	7.7	0	0	557.14	542.85	0	0	6553.46	6092.24	50582.85				
1 0 1 1	11.3	0.0031	7.82	413.3	0	340.64	346.04	4489.77	0	3871.32	3653.92	48060.09				
0 1 1 1	10.97	0.0033	7.24	0	476.57	307.5	315.91	0	4540.24	3502.13	3318.3	45442.75				
1 1 1 1	10.08	0.0023	7.47	261.22	384.35	218.98	235.43	2862.76	3568.97	2569.71	2470.64	45888.41				

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 1			Stage 2			Stage 3			Stage 4			Stage 5			Stage 6		
สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	
การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	การทํางาน	
1 2 3 4	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	ค่าใช้จําน	
1 1 0 0	45869.08	59357.61	59357.61	1 1 1 0	68973.81	1 1 1 0	79076.09	1 1 1 0	59357.61	1 1 1 0	59357.61	1 1 0 0	45869.08	50158.87	50158.87	50158.87	
1 0 1 0	50158.87	58238.21	58238.21	1 0 1 1	67857.13	1 1 1 1	76472.79	1 1 1 1	58238.21	1 0 1 1	58238.21	0 1 0 1	50158.87	47016.11	47016.11	47016.11	
0 1 1 0	47016.11	62197.64	62197.64	1 0 1 1	72255.78	1 1 1 1	72255.78	1 1 1 1	62197.64	1 0 1 1	62197.64	1 1 1 0	47016.11	45844.69	45844.69	45844.69	
1 1 1 0	45844.69	59222.87	59222.87	1 1 1 1	69087.42	1 1 1 1	69087.42	1 1 1 1	59222.87	0 1 1 1	59222.87	1 1 1 0	45844.69	49055.79	49055.79	49055.79	
1 0 0 1	49055.79	58424.68	58424.68	1 1 1 1	67258.29	1 1 1 1	67258.29	1 1 1 1	58424.68	1 1 1 1	58424.68	1 0 0 1	49055.79	45903.5	45903.5	45903.5	
0 1 0 1	45903.5											0 1 0 1	45903.5	44747.08	44747.08	44747.08	
1 1 0 1	44747.08											1 1 0 1	44747.08	50582.85	50582.85	50582.85	
0 0 1 1	50582.85											0 0 1 1	50582.85	48060.09	48060.09	48060.09	
1 0 1 1	48060.09											1 0 1 1	48060.09	45442.75	45442.75	45442.75	
0 1 1 1	45442.75											0 1 1 1	45442.75	45888.41	45888.41	45888.41	
1 1 1 1	45888.41											1 1 1 1	45888.41				

ตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6	
สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ
การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน	การทำงาน
11000	57336.35	1100	74197.02	1100	86217.27	1100	98845.12	1100	74197.02	1100	57336.35
10100	62698.59	11010	72797.76	11010	84821.42	11110	95590.99	11010	72797.76	10100	62698.59
01100	58770.15	10110	77747.06	10110	90319.73	11101	95590.99	10110	77747.06	1100	58770.15
11100	57305.87	11110	74028.60	11110	86359.28	11011	94191.51	01110	74028.60	11100	57305.87
10010	61319.38	11110	73030.85	11110	84072.87	10111	99124.71	11110	73030.85	10010	61319.38
01010	57379.38	11001	72797.76	11001	84821.42	01111	95427.12	11001	72797.76	01010	57379.38
11010	55933.85	10101	77747.06	10101	90319.73	11111	94206.37	10101	77747.06	11010	55933.85
00110	63228.56	1101	74028.60	1101	86359.28			01101	74028.59	00110	63228.56
10110	60075.11	1101	73030.85	1101	84072.87			11101	73030.85	10110	60075.11
01110	56803.44	10011	76349.98	10011	88942.57			10011	76349.98	01110	56803.44
11110	57360.52	1011	72628.82	1011	84973.2			01011	72628.82	11110	57360.52
10001	61319.74	11011	71658.78	11011	82681.83			11011	71658.78	10001	61319.74
01001	57379.38	0111	77854.83	0111	90819.35			00111	77854.83	01001	57379.38
11001	55933.85	10111	75754.86	10111	87193.97			10111	75754.86	11001	55933.85
00101	63228.56	1111	72523.10	1111	83716.031			01111	72523.10	00101	63228.56
10101	60075.11	11111	72991.25	11111	83403.11			11111	72991.25	10101	60075.11
01101	56803.44									01101	56803.44
11101	57360.52									11101	57360.52
10011	58683.32									10011	58683.32
01011	55423.52									01011	55423.52
11011	56036.66									11011	56036.66
00111	59738.7									00111	59738.7
10111	59517.99									10111	59517.99
01111	56705.24									01111	56705.24
11111	58107.34									11111	58107.34

ตารางที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 4.4 แสดงการเลือกเส้นทางจากวิธี ยูนิตคอมมิทเมนต์ (unit commitment) ในกรณีที่ 1

Stage 1		Stage 2				Stage 3				Stage 4				Stage 5				Stage 6											
		สถานะ				สถานะ				สถานะ				สถานะ				สถานะ											
การทำงาน (x11)		การทำงาน (x15)				การทำงาน (x15)				การทำงาน (x15)				การทำงาน (x11)				การทำงาน (x11)											
1	2	3	4	ค่าใช้จ่าย	1	2	3	4	ค่าใช้จ่าย	1	2	3	4	ค่าใช้จ่าย	1	2	3	4	ค่าใช้จ่าย	1	2	3	4	ค่าใช้จ่าย					
1	1	0	1	44747.08	1	1	1	1	58424.68	1	1	1	1	67258.29	1	1	1	1	76472.79	1	1	0	1	58238.21	1	1	0	1	44747.08

ตาราง 4.5 แสดงการเลือกเส้นทางจากวิธี ยูนิตคอมมิทเมนต์ (unit commitment) ในกรณีที่ 2

Stage 1		Stage 2				Stage 3				Stage 4				Stage 5				Stage 6															
		สถานะ				สถานะ				สถานะ				สถานะ				สถานะ															
การทำงาน (x26)		การทำงาน (x27)				การทำงาน (x27)				การทำงาน (x27)				การทำงาน (x27)				การทำงาน (x26)															
1	2	3	4	5	ค่าใช้จ่าย	1	2	3	4	5	ค่าใช้จ่าย	1	2	3	4	5	ค่าใช้จ่าย	1	2	3	4	5	ค่าใช้จ่าย	1	2	3	4	5	ค่าใช้จ่าย				
1	2	3	4	5	55423.52	1	2	3	4	5	71658.78	1	1	0	1	1	82681.83	1	1	0	1	1	94191.51	1	1	0	1	1	71658.78	0	1	1	55423.52

สถานะ				สถานะ				สถานะ				สถานะ				สถานะ							
การจ้าง	ค่าจ้าง	1 2 3 4	ค่าจ้าง	การจ้าง	ค่าจ้าง	1 2 3 4	ค่าจ้าง	การจ้าง	ค่าจ้าง	1 2 3 4	ค่าจ้าง	การจ้าง	ค่าจ้าง	1 2 3 4	ค่าจ้าง	การจ้าง	ค่าจ้าง						
1 1 0 0	22934.54	1 1 1 0	29678.81	1 1 1 0	34486.905	1 1 1 0	39538.05	1 1 1 0	29678.81	1 1 1 0	29678.81	1 1 0 0	22934.54										
1 0 1 0	25079.44	1 1 0 1	29119.1	1 1 1 1	33928.565	1 1 1 1	38236.4	1 1 0 1	29119.1	1 0 1 1	31098.82	1 1 0 1	25079.44										
0 1 1 0	23508.06	1 0 1 1	31098.82	1 0 1 1	36127.89			1 0 1 1	31098.82	0 1 1 1	29611.44	1 1 0 1	23508.06										
1 1 1 0	22922.35	1 1 1 1	29611.44	1 1 1 1	34543.71			0 1 1 1	29611.44	1 1 1 1	29212.34	1 1 0 1	22922.35										
1 0 0 1	24527.9	1 1 1 1	29212.34	1 1 1 1	33629.145			1 1 1 1	29212.34	1 1 1 1	29212.34	1 0 0 1	24527.9										
0 1 0 1	22951.75											0 1 0 1	22951.75										
1 1 0 1	22373.54											1 1 0 1	22373.54										
0 0 1 1	25291.43											0 0 1 1	25291.43										
1 0 1 1	24030.05											1 0 1 1	24030.05										
0 1 1 1	22721.38											0 1 1 1	22721.38										
1 1 1 1	22944.21											1 1 1 1	22944.21										
Stage 7				Stage 8				Stage 9				Stage 10				Stage 11				Stage 12			
สถานะ				สถานะ				สถานะ				สถานะ				สถานะ				สถานะ			
การจ้าง				การจ้าง				การจ้าง				การจ้าง				การจ้าง				การจ้าง			
1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง	1 2 3 4	ค่าจ้าง				
1 1 0 0	22934.54	1 1 1 0	29678.81	1 1 1 0	34486.905	1 1 1 0	39538.05	1 1 1 0	29678.81	1 1 1 0	29678.81	1 1 0 0	22934.54										
1 0 1 0	25079.44	1 1 0 1	29119.1	1 1 0 1	33928.565	1 1 1 1	38236.4	1 1 0 1	29119.1	1 0 1 1	31098.82	1 1 0 1	25079.44										
0 1 1 0	23508.06	1 0 1 1	31098.82	1 0 1 1	36127.89			1 0 1 1	31098.82	0 1 1 1	29611.44	1 1 0 1	23508.06										
1 1 1 0	22922.35	1 1 1 1	29611.44	1 1 1 1	34543.71			0 1 1 1	29611.44	1 1 1 1	29212.34	1 1 0 1	22922.35										
1 0 0 1	24527.9	1 1 1 1	29212.34	1 1 1 1	33629.145			1 1 1 1	29212.34	1 1 1 1	29212.34	1 0 0 1	24527.9										
0 1 0 1	22951.75											0 1 0 1	22951.75										
1 1 0 1	22373.54											1 1 0 1	22373.54										
0 0 1 1	25291.43											0 0 1 1	25291.43										
1 0 1 1	24030.05											1 0 1 1	24030.05										
0 1 1 1	22721.38											0 1 1 1	22721.38										
1 1 1 1	22944.21											1 1 1 1	22944.21										

ตารางที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6	
สถานะ	ค่าจ้าง	สถานะ	ค่าจ้าง	สถานะ	ค่าจ้าง	สถานะ	ค่าจ้าง	สถานะ	ค่าจ้าง	สถานะ	ค่าจ้าง
การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน	การจ้างงาน
1 2 3 4 5	28668.17	1 2 3 4 5	37098.51	1 2 3 4 5	43108.63	1 2 3 4 5	49422.56	1 2 3 4 5	37098.51	1 2 3 4 5	28668.17
1 0 1 0 0	31349.31	1 1 0 1 0	36398.88	1 1 0 1 0	42410.71	1 1 1 1 0	47795.51	1 1 0 1 0	36398.88	1 0 1 0 0	31349.31
0 1 1 0 0	29385.07	1 0 1 1 0	38873.53	1 0 1 1 0	45159.86	1 1 0 1 1	47795.51	1 0 1 1 0	38873.53	0 1 1 0 0	29385.07
1 1 1 0 0	28652.94	0 1 1 1 0	37014.30	1 1 1 1 0	43179.63	1 0 1 1 1	47095.75	1 1 1 1 0	37014.30	1 1 1 0 0	28652.94
1 0 0 1 0	30659.69	1 1 1 1 0	36515.43	1 1 1 1 0	42036.43	1 0 1 1 1	49562.35	1 1 1 1 0	36515.43	1 0 0 1 0	30659.69
0 1 0 1 0	28689.69	1 0 0 1 1	36398.88	1 0 0 1 1	42410.71	1 1 1 1 1	47713.56	1 0 0 1 1	36398.88	0 1 0 1 0	28689.69
1 0 1 0 1	27966.93	1 0 1 0 1	38873.53	1 0 1 0 1	45159.86	1 1 1 1 1	47103.19	1 0 1 0 1	38873.53	1 1 0 1 0	27966.93
0 0 1 1 0	31614.28	0 1 1 0 1	37014.30	1 1 0 1 1	43179.63			0 1 1 0 1	37014.30	0 0 1 1 0	31614.28
1 0 1 1 0	30037.56	1 1 1 0 1	36515.43	1 1 1 0 1	42036.43			1 1 1 0 1	36515.43	1 0 1 1 0	30037.56
0 1 1 1 0	28401.72	1 0 0 1 1	38174.99	1 0 0 1 1	44471.28			1 0 0 1 1	38174.99	0 1 1 1 0	28401.72
1 1 1 1 0	28680.26	0 1 0 1 1	36314.41	1 0 1 1 1	42486.59			0 1 0 1 1	36314.41	1 1 1 1 0	28680.26
1 0 0 0 1	30659.87	1 1 0 1 1	35829.39	1 1 0 1 1	41340.91			1 1 0 1 1	35829.39	1 0 0 0 1	30659.87
0 1 0 0 1	28689.69	0 0 1 1 1	38927.42	0 1 1 1 1	45409.67			0 0 1 1 1	38927.42	0 1 0 0 1	28689.69
1 1 0 0 1	27966.93	1 0 1 1 1	37877.43	1 0 1 1 1	43596.98			1 0 1 1 1	37877.43	1 1 0 0 1	27966.93
0 0 1 0 1	31614.28	0 1 1 1 1	36261.55	0 1 1 1 1	41858.01			0 1 1 1 1	36261.55	0 0 1 0 1	31614.28
1 0 1 0 1	30037.56	1 1 1 1 1	36495.63	1 1 1 1 1	41701.55			1 1 1 1 1	36495.63	1 0 1 0 1	30037.56
0 1 1 0 1	28401.72									0 1 1 0 1	28401.72
1 1 1 0 1	28680.26									1 1 1 0 1	28680.26
1 0 0 1 1	29341.66									1 0 0 1 1	29341.66
0 1 0 1 1	27711.76									0 1 0 1 1	27711.76
1 1 0 1 1	28018.33									1 1 0 1 1	28018.33
0 0 1 1 1	29869.35									0 0 1 1 1	29869.35
1 0 1 1 1	29758.99									1 0 1 1 1	29758.99
0 1 1 1 1	28352.62									0 1 1 1 1	28352.62
1 1 1 1 1	29053.67									1 1 1 1 1	29053.67

เอกสารนี้เป็นเอกสารต้นฉบับการคำนวณ  
 ไม่वारณณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 7		Stage 8		Stage 9		Stage 10		Stage 11		Stage 12	
สถานะ	ค่าใช้จ่าย	สถานะ	ค่าใช้จ่าย	สถานะ	ค่าใช้จ่าย	สถานะ	ค่าใช้จ่าย	สถานะ	ค่าใช้จ่าย	สถานะ	ค่าใช้จ่าย
1 2 3 4 5	28668.17	1 2 3 4 5	37098.51	1 2 3 4 5	43108.635	1 2 3 4 5	49422.56	1 2 3 4 5	37098.51	1 2 3 4 5	28668.17
1 0 1 0 0	31349.3	1 1 0 1 0	36398.88	1 1 0 1 0	42410.711	1 1 1 1 0	47795.51	1 1 0 1 0	36398.88	1 0 1 0 0	31349.3
0 1 1 0 0	29385.07	1 0 1 1 0	38873.53	1 1 0 1 0	45159.863	1 1 0 1 0	47795.51	1 0 1 1 0	38873.53	1 1 0 0 0	29385.07
1 1 1 0 0	28652.94	0 1 1 1 0	37014.30	1 1 1 1 0	43179.638	1 0 1 1 1	47095.750	1 1 1 1 0	37014.30	1 1 1 0 0	28652.94
1 0 0 1 0	30659.69	1 1 1 1 0	36515.43	1 1 1 1 0	42036.435	1 0 1 1 1	49562.35	1 1 1 1 0	36515.43	1 0 0 1 0	30659.69
0 1 0 1 0	28689.69	1 1 0 0 1	36398.88	1 1 0 0 1	42410.711	1 1 1 1 1	47713.56	1 1 0 0 1	36398.88	1 0 1 0 0	28689.69
1 1 0 1 0	27966.93	1 0 1 0 1	38873.53	1 0 1 0 1	45159.863	1 1 1 1 1	47103.19	1 0 1 0 1	38873.53	1 1 0 1 0	27966.93
0 0 1 1 0	31614.28	1 1 0 1 0	37014.30	1 1 0 1 0	43179.638			0 1 1 0 1	37014.30	0 0 1 1 0	31614.28
1 0 1 1 0	30037.56	1 1 1 0 1	36515.43	1 1 1 0 1	42036.435			1 1 1 0 1	36515.43	1 0 1 1 0	30037.56
0 1 1 1 0	28401.72	1 0 0 1 1	38174.99	1 0 0 1 1	44471.286			1 0 0 1 1	38174.99	1 1 1 0 0	28401.72
1 1 1 1 0	28680.26	0 1 0 1 1	36314.41	1 0 1 1 1	42486.598			0 1 0 1 1	36314.41	1 1 1 1 0	28680.26
1 0 0 0 1	30659.87	1 1 0 1 1	35829.39	1 1 0 1 1	41340.913			1 1 0 1 1	35829.39	1 0 0 0 1	30659.87
0 1 0 0 1	28689.69	0 1 1 1 1	38927.42	0 1 1 1 1	45409.674			0 0 1 1 1	38927.42	1 0 0 0 1	28689.69
1 1 0 0 1	27966.93	1 0 1 1 1	37877.43	1 0 1 1 1	43596.987			1 0 1 1 1	37877.43	1 1 0 0 1	27966.93
0 0 1 0 1	31614.28	1 1 1 1 1	36261.55	1 1 1 1 1	41858.016			0 1 1 1 1	36261.55	0 1 0 1 0	31614.28
1 0 1 0 1	30037.56	1 1 1 1 1	36495.63	1 1 1 1 1	41701.557			1 1 1 1 1	36495.63	1 0 1 0 1	30037.56
0 1 1 0 1	28401.72									0 1 1 0 1	28401.72
1 1 1 0 1	28680.26									1 1 1 0 1	28680.26
1 0 0 1 1	29341.66									1 0 0 1 1	29341.66
0 1 0 1 1	27711.76									0 1 0 1 1	27711.76
1 1 0 1 1	28018.33									1 1 0 1 1	28018.33
0 0 1 1 1	29869.35									0 0 1 1 1	29869.35
1 0 1 1 1	29758.99									1 0 1 1 1	29758.99
0 1 1 1 1	28352.62									0 1 1 1 1	28352.62
1 1 1 1 1	29053.67									1 1 1 1 1	29053.67

ตารางที่ 4.7

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้ในการค้า  
 ไม่วการณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6	
สถานะ (x11)	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ
การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง
1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย
1 1 0 1	22373.54	1 1 0 1	29119.11	1 1 1 1	33629.14	1 1 1 1	38236.41	1 1 0 1	29119.11	1 1 0 1	22373.54
Stage 7		Stage 8		Stage 9		Stage 10		Stage 11		Stage 12	
สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ
การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง
1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย	1 2 3 4	ค่าใช้จ่ย
1 1 0 1	22373.54	1 1 0 1	29119.11	1 1 1 1	33629.14	1 1 1 1	38236.41	1 1 0 1	29119.11	1 1 0 1	22373.54

ตาราง 4.8 แสดงการเลือกเส้นทางจากวิธี ยูนิตคอมมิตเมนต์ (unit commitment) ในกรณีที่ 3

Stage 1		Stage 2		Stage 3		Stage 4		Stage 5		Stage 6	
สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ
การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง
1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย
0 1 0 1 1	27711.76	1 1 0 1 1	36495.63	1 1 0 1 1	41340.91	1 1 0 1 1	47095.75	1 1 0 1 1	35829.39	0 1 0 1 1	27711.76
Stage 7		Stage 8		Stage 9		Stage 10		Stage 11		Stage 12	
สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ	สถานะ
การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง	การจ้าง
1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย	1 2 3 4 5	ค่าใช้จ่ย
0 1 0 1 1	27711.76	1 1 0 1 1	36495.63	1 1 0 1 1	41340.91	1 1 0 1 1	47095.75	1 1 0 1 1	35829.39	0 1 0 1 1	27711.76

ตาราง 4.9 แสดงการเลือกเส้นทางจากวิธี ยูนิตคอมมิตเมนต์ (unit commitment) ในกรณีที่ 4

## หมายเหตุ

ก่อนที่จะทำการทดลองกับกรณีศึกษาเพื่อให้ได้ตารางที่ 4.4 และ 4.5 ต้องทำการหาค่าสัมประสิทธิ์และข้อจำกัดต่างๆ สำหรับ วิธีโปรแกรมเชิงพลวัต (Dynamic programming) ก่อน อาทิ เช่น การผลิตที่ต้องทำงานตลอด (Must run units) , ข้อจำกัดการเปลี่ยนแปลงกำลังผลิต (ramp-rare limit) , เวลาเดินเครื่องอย่างน้อยที่สุด (minimum up time) และ เวลาหยุดเครื่องอย่างน้อยที่สุด (minimum down time) เพื่อลดการคำนวณของระบบให้ลดน้อยลง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลองและข้อเสนอแนะ

การแก้ปัญหาชนิดคอมมิตเมนต์ที่นำเสนอในงานวิจัยฉบับนี้มีส่วนประกอบหลักอยู่ 2 ส่วนที่สำคัญดังนี้ ส่วนประกอบแรก คือ การแก้ปัญหาการจ่ายโหลดอย่างประหยัด (Economic dispatch) ซึ่งเป็นส่วนหนึ่งของวิธีการแก้ปัญหาชนิดคอมมิตเมนต์ สำหรับส่วนที่เป็นการแก้ปัญหาการจ่ายโหลดอย่างประหยัดนั้นเป็นการนำเครื่องกำเนิดไฟฟ้าที่เดินเครื่องอยู่มาทำการแก้ปัญหาในการที่จะทำการจ่ายโหลดแต่ละเครื่องออกมาอย่างประหยัดที่สุดเพื่อนำไปใช้ในการคำนวณหาการแก้ปัญหาชนิดคอมมิตเมนต์ต่อไป ส่วนประกอบหลักส่วนที่สองคือ ยูนิคคอมมิตเมนต์ ซึ่งเป็นส่วนประกอบที่สำคัญและใหญ่ที่สุดที่ครอบคลุมส่วนประกอบส่วนแรกไว้ด้วย มีหน้าที่ในการค้นหาแผนการเดินทางเครื่องกำเนิดไฟฟ้าเพื่อหาเส้นทางในการผลิตไฟฟ้าแต่ละช่วงเวลา เพื่อให้ได้เส้นทางที่ประหยัดที่สุดโดยมีเงื่อนไขตามที่ได้กล่าวไว้ในบทที่ 1 งานวิจัยนี้ได้นำเสนอแนวทางการแก้ปัญหาชนิดคอมมิตเมนต์โดยวิธี โปรแกรมเชิงพลวัต (Dynamic programming) ซึ่งวิธีที่นำเสนอมีการพิจารณาเงื่อนไขดังนี้ การผลิตที่ต้องทำงานตลอด (Must run units) ,ขีดจำกัดการเปลี่ยนแปลงกำลังผลิต (ramp-rate limit) ,เวลาเดินเครื่องอย่างน้อยที่สุด (minimum up time) และ เวลาหยุดเครื่องอย่างน้อยที่สุด (minimum down time) ผู้วิจัยได้ทำการเขียนโปรแกรมภาษาซีบนระบบปฏิบัติการ Microsoft Windows Xp โดยใช้โปรแกรม Visual Basic 6 เพื่อใช้ในการคำนวณหาเส้นทางที่ประหยัดและเป็นไปตามเงื่อนไข

ผลจากการทดลองพบว่าวิธี โปรแกรมเชิงพลวัต (Dynamic programming) สามารถใช้แก้ปัญหาชนิดคอมมิตเมนต์ที่มีเงื่อนไขตามที่กำหนดได้ ในกรณีที่ 1 จำนวนเครื่องกำเนิดไฟฟ้า 4 เครื่อง แบ่งช่วงเวลาทุกๆ 4 ชั่วโมง ใช้เวลาในการทำงานของโปรแกรม เป็นเวลา 16 วินาที และเมื่อเพิ่มจำนวนเครื่องกำเนิดไฟฟ้า ในกรณีที่สอง ใช้เวลาในการคำนวณเพิ่มขึ้นเป็น 3 ชั่วโมง ดังตาราง 5.1 แสดงให้เห็นว่าจำนวนเครื่องกำเนิดไฟฟ้ามีผลต่อเวลาในการคำนวณ

จำนวน Gen ในระบบ (unit)	จำนวน stage ที่คำนวณ	เวลาที่ใช้ในการคำนวณ (Hr)
4	6	0.0046
	7	0.024
	8	0.033
	9	0.186
	10	20.8
5	12	51.3
	5	0.041
	6	3.38
	12	128.65

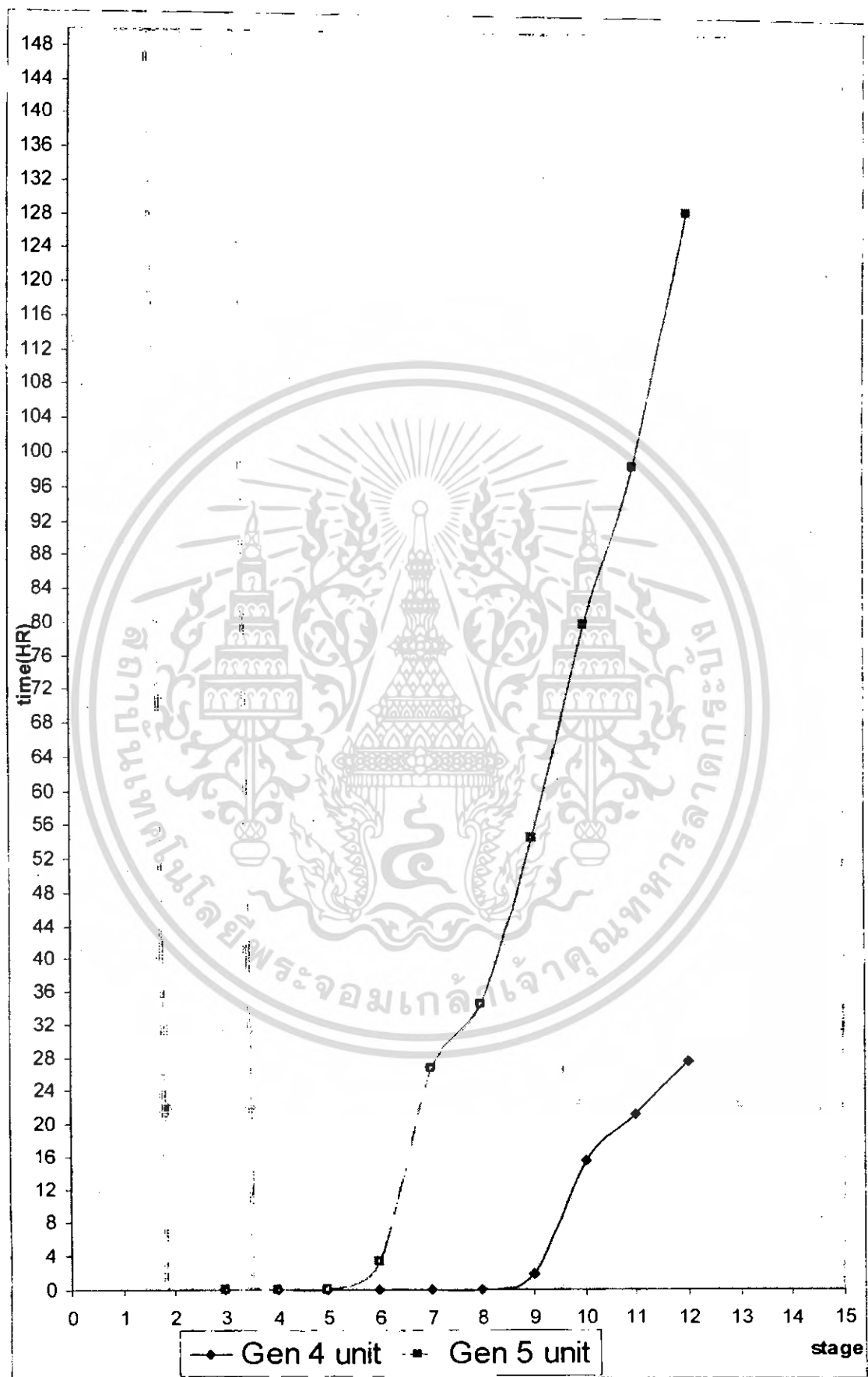
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6	3	0.0022
	4	0.67
	5	3.5
7	3	0.00277
	4	0.75

ตารางที่ 5.1

### ผู้วิจัยขอสรุปว่า

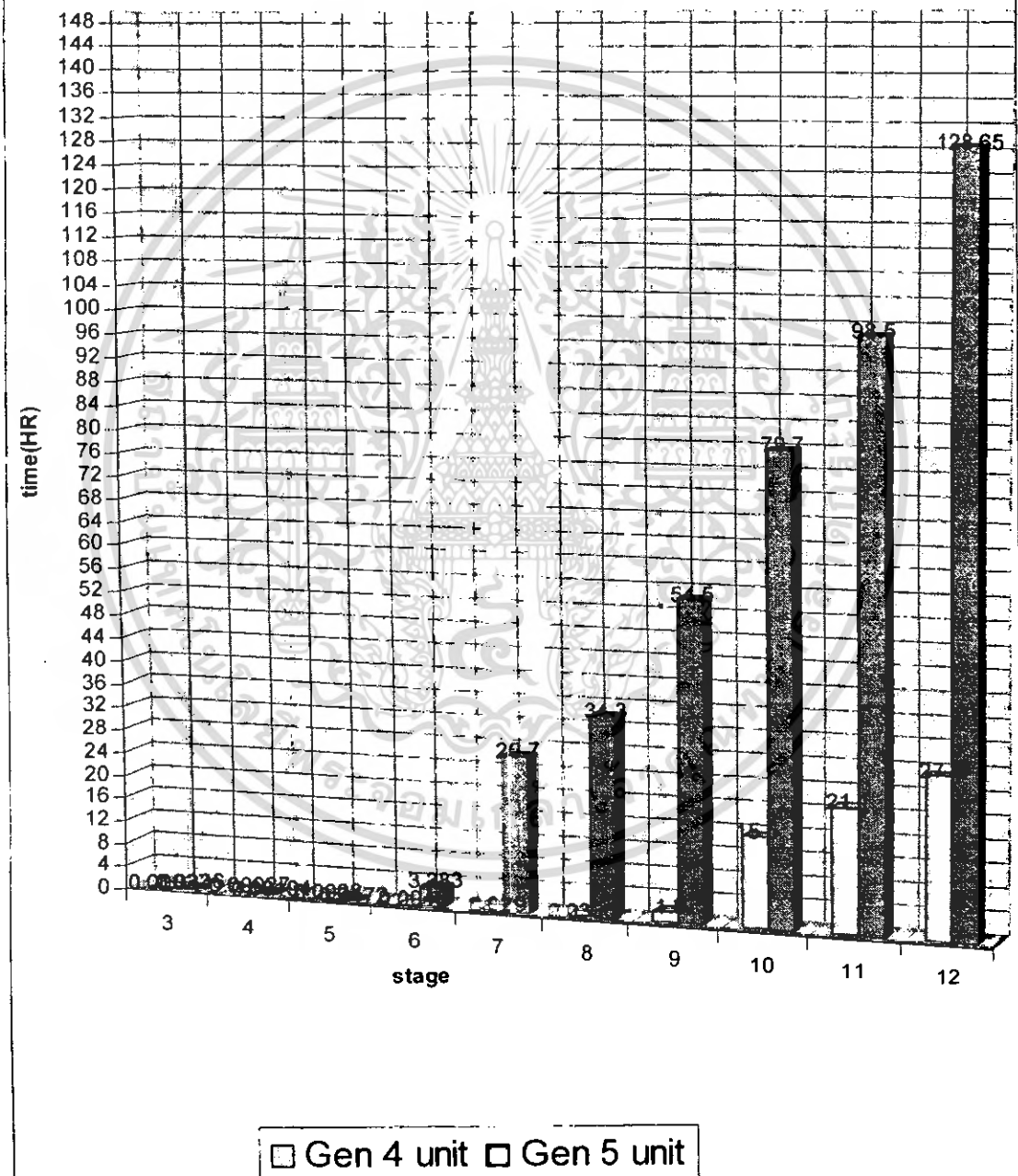
วิธี โปรแกรมเชิงพลวัต (Dynamic programming) เป็นโปรแกรมที่คำนวณการแก้ปัญหา ยูนิทคอมมิตเมนต์ได้ค่อนข้างละเอียดมากในระดับที่ยอมรับได้ และสามารถทำให้ผู้ควบคุมระบบเห็นขั้นตอนการคิดคำนวณการแก้ปัญหาได้ทุกช่วงเวลาอย่างชัดเจน แต่มีปัญหาในกระบวนการคิดคำนวณของตัวโปรแกรม เนื่องจากมีการใช้เวลาในการคำนวณที่มาก โดยเวลาที่ใช้คำนวณจะขึ้นอยู่กับ จำนวนเครื่องกำเนิดไฟฟ้า และ จำนวนคาบเวลาที่แบ่งออกมาใน 1 วัน ดังที่แสดงให้เห็นในผลการทดลองในบทข้างต้น อีกทั้งยังใช้เงื่อนไขการเดินทางเครื่องมาช่วยลดเวลาในการคำนวณการแก้ปัญหา ยูนิทคอมมิตเมนต์



กราฟแสดงการเปรียบเทียบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟแสดงการเปรียบเทียบเวลา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข้อเสนอแนะ

อย่างไรก็ตามวิธีโปรแกรมเชิงพลวัต (Dynamic programming) ก็ยังมีข้อบกพร่องอยู่ในด้านการใช้เวลามากในการคำนวณ และปัญหาด้านการจัดการข้อมูลและด้านประมวลผลข้อมูล เนื่องจากเป็นการคำนวณระบบที่ใหญ่มาก และยังคงมีเงื่อนไขที่ต้องนำมาพิจารณาอีกหลายเงื่อนไข ดังต่อไปนี้ เงื่อนไขกำลังคน เงื่อนไขการผลิตพลังงานสำรอง เงื่อนไขขีดจำกัดทางด้านเชื้อเพลิง ตลอดจนรวมไปถึงการพิจารณาเครื่องกำเนิดไฟฟ้าพลังงานน้ำเพิ่มเติมด้วย ซึ่งจะส่งผลทำให้การพิจารณาแก้ปัญหาที่มีความยุ่งยากซับซ้อนขึ้นไปอีก ประกอบกับวิธี โปรแกรมเชิงพลวัต (Dynamic programming) ยังมีปัญหาภายในที่ต้องปรับปรุงอีก จึงจำเป็นที่จะต้องมีการปรับปรุงขั้นตอนภายในเพิ่มเติมอีก ทั้งนี้เพื่อให้วิธี โปรแกรมเชิงพลวัต (Dynamic programming) สามารถนำไปใช้แก้ปัญหา ยูนิตคอมมิตเมนต์ (unit commitment) และการจ่ายกระแสในทางปฏิบัติได้อย่างถูกต้องและแม่นยำมากขึ้น

- อาจนำวิธีการทำยูนิตคอมมิตเมนต์ วิธีอื่นๆ เข้ามาช่วยในการคำนวณเพื่อลดระยะเวลาการคำนวณให้เร็วขึ้น เช่น วิธีเจเนติก (Genetic Algorithm) หรือ วิธีแลกเชนแบบลากรองจ์ (Lagrangian relaxation)
- เพิ่มเงื่อนไขของยูนิตคอมมิตเมนต์ เพื่อช่วยลดระยะเวลาในการทำงาน (Unit Must Run)
- ใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูง มาช่วยในการคำนวณเพื่อลดระยะเวลาในการประมวลผล

### ปัญหาที่พบและแนวทางการแก้ไข

- จำนวนเครื่องกำเนิดไฟฟ้ามากไป หรือ จำนวนคาบเวลามาก จะใช้เวลาในการคำนวณนาน
- เนื่องจากจำนวนกรณีที่ต้องการเปรียบเทียบมีมาก ทำให้มีปัญหาในด้านการเก็บข้อมูลของโปรแกรม
- โปรแกรมมีข้อจำกัดในการเก็บข้อมูล ทำให้คำนวณได้ที่จำนวนเครื่องกำเนิดไฟฟ้าสูงสุด 15 ตัว

## เอกสารอ้างอิง

- [1] Y.Y.Hsu., Su, C.C.Liang, C.C.Lin.C.J. and C.T.Huang., "Dynamic Security Constrained Multi-Area Unit Commitment", *IEEE Trans. Power Systems*, Vol.6, pp.1049-1055, 1991.
- [2] A.G.Bakirtzis and C.E.Zoumas, "Lambda of Lagrangian relaxation solution to unit commitment problem", *IEE Proc. Generation, Transmission and Distribution*, Vol.147, No.2, pp.131-136, 2000.
- [3] Chuan-ping Chang, Chih-wen Liu and Chun-chang Liu, "Unit Commitment by Lagrangian Relaxation and Genetic Algorithms", *IEEE Trans. Power Systems*, Vol.15, No.2, pp.707-714, 2000.
- [4] A.H.Mantawy, Youssef L. Abdel-Magid and Shokri Z.Selim, "A Unit Commitment By Tabu Search", *IEE Proc. Generation, Transmission and Distribution*, Vol. 145, No. 1, pp.56-64, 1998.
- [5] Whei-Min Lin, Fu-Sheng Cheng and Ming-Tong Tsay, "An Improved Tabu Search for Economic Dispatch With Multiple Minima", *IEEE Trans. Power Systems*, Vol.17, No.1, pp.108-112, 2002.
- [6] K.A.Juste, H.Kita, E.Tanaka and J.Hasegawa, "An Evolutionary Programming Solution to the Unit Commitment Problem", *IEEE Trans. Power Systems*, Vol.14, No.4, pp.1452-1459, 1999.
- [7] O.B. Fosso et. al., "Generation scheduling in a deregulated system. The Norwegian case". Paper accepted for presentation at the, IEEE Winter Meeting 1998.
- [8] S.O. Orero and M.R. Irving, "Large scale unit commitment using a hybrid genetic algorithm", *Electrical Power & Energy Systems*, 'Jol. 19, NO. 1,
- [9] MOKHTARI, S., SINGH, J., and WOLLENBERG, B.: 'A unit commitment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- expert system'. IEEE Trans. Power Svst., 1988, PWRS3, (1).pp. 272-273 .
- [10] ROLSTON, D.W.: 'Principles of artificial intelligence and expert systems development' (Mdiraw-Hill, 1988)
- [11] WATERMAN, D.A.: 'A guide to expert systems' (Addison-Wesley, 1986)
- [12] WOOD, A.J., and WOLLENBERG, B.F.: 'Power generation, operation & control' (Wiley, Canada, 1984)
- [13] SACKETT, B.: 'Operational requirements for optimal unit commitment procedure', 1978 Conference on Power System Control, Oklahoma City, USA, 1978, pp- 3 W
- [14] Tong, S. K., Shahidehpour, S. M., LLAIInn novative Approach to Generation Scheduling in Large-scale Hydro-thermal Power System with Fuel Constrained Units," IEEE Trans. on Power Systems, Vol. 5, No. 2, pp. 665-673, May 1990.
- [15] Wood, W. G., "Spinning Reserve Constrained Static and Dynamic Economic Dispatch," IEEE Trans. on Power Systems, Vol. PAS-101, No. 2, pp. 381-388, Feb. 1982.
- [16] Cohen, A. I., "Modeling Unit Ramp Limitations in Unit Commitment," Proceeding of the 10th Power Systems Computation Conference, pp. 1107-1114, Graz, Austria, Aug.
- [17] Tong, S. K., Shahidehpour, S. M., Ouyang, Z., "A Heuristic Short-Term Unit Commitment," IEEE Trans. on Power Systems, Vol. 6, No. 3, pp. 1210-1216, Aug. 1991.
- [18] A. Merlin and P. Sandrin, "A New Method for Unit Commitment at Electricite de Ranee," IEEE Trans. on PAS, Vol. PAS-102, No. 5 pp. 1218-1225, May 1983.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [19] Patton, A. D., "Dynamic Optimal Dispatch of Real Power for Thermal Generating Units," Proceedings of the 8th Power Industry Computer Applications Conference, Minneapolis, Minnesota, June 4-6, 1973.
- [20] **ROSS**, D. W., Kim, S., "Dynamic Economic Dispatch of Generation," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-99, No. 6, pp. 2060-2068, Nov/Dec 1980.
- [21] Improve the Unit Commitment Scheduling by Using the Neural-Network-Based Short-Term Load Forecasting  
Titti Saksornchai, Wei-Jen Lee, *Senior Member, IEEE*, Kittipong Methaprayoon, *Student Member, IEEE*, James R. Liao, *Member, IEEE*, and Richard J. Ross

ภาคผนวก ก

Code ของตัวโปรแกรม

```

Private Sub Command1_Click()
Hide
Form1.Show
End Sub

Private Sub Form_Load()
Dim aT As Double, bT As Double
Dim MAXcol As Integer
Dim i As Integer
MAXcol = 6 + (MAX * 2)
Grid1.Cols = MAXcol
Grid1.Clear
For i = 1 To MAXcol - 1
Grid1.ColWidth(i) = 2000
Next
Grid1.TextMatrix(0, 1) = "Lamda"
Grid1.TextMatrix(0, 2) = "aT"
Grid1.TextMatrix(0, 3) = "bT"
For i = 1 To MAX
Grid1.TextMatrix(0, 3 + i) = "p" & i
Next

For i = 1 To MAX
Grid1.TextMatrix(0, MAX + 3 + i) = "F" & i
Next

tmpCount = Configuration.List1.ListCount
tmpRows = (CNT + ((tmpCount - 1) * CNT))
Grid1.Rows = tmpRows
For k = 1 To tmpCount '===== begin stage
Grid1.TextMatrix(1 + ((k - 1) * CNT), 0) = Configuration.List1.List(k - 1) & "(X1)"
For j = 1 To CNT '=====aT

aT = 0
PoverLoad = 0
If Form1.Grid1.TextMatrix(j, 1) = "" Then Exit For
For i = 1 To MAX
If Form1.Grid1.TextMatrix(j, i) = "1" Then
aT = aT + 1 / Configuration.G1.TextMatrix(i - 1, 3)
PoverLoad = PoverLoad + Configuration.G1.TextMatrix(i - 1, 2)
End If
Next
setcolor = vbWhite
If PoverLoad < Configuration.List1.List(k - 1) Then
aT = 0
setcolor = vbYellow

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ให้ใช้ฟรีโดยไม่คิดค่าใช้จ่ายเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
  aT = 1 / aT
End If
Grid1.TextMatrix(j + ((k - 1) * CNT), 2) = Round(aT, 9)
If j > 1 Then Grid1.TextMatrix(j + ((k - 1) * CNT), 0) = "X" & j
Grid1.COL = 2
Grid1.ROW = (j + (k - 1) * CNT)

Grid1.CellBackColor = setcolor
Next

For j = 1 To CNT      '===== bT
  PoverLoad = 0
  bT = 0
  If Form1.Grid1.TextMatrix(j, 1) = "" Then Exit For
  For i = 1 To MAX
    If Form1.Grid1.TextMatrix(j, i) = "1" Then
      bT = bT + Configuration.G1.TextMatrix(i - 1, 4) /
Configuration.G1.TextMatrix(i - 1, 3)
      PoverLoad = PoverLoad + Configuration.G1.TextMatrix(i - 1, 2)
    End If
  Next
  setcolor = vbWhite
  If PoverLoad < Configuration.List1.List(k - 1) Then
    bT = 0
    setcolor = vbYellow
  Else
    bT = Grid1.TextMatrix(j + ((k - 1) * CNT), 2) * bT
  End If
  Grid1.TextMatrix(j + ((k - 1) * CNT), 3) = Round(bT, 9)
  Grid1.COL = 3
  Grid1.ROW = (j + (k - 1) * CNT)
  Grid1.CellBackColor = setcolor
Next

For j = 1 To CNT      '===== Lamda **** stage is refferenced in this
process
  stage = Configuration.List1.List(k - 1)
  LamDa = 0
  PoverLoad = 0
  If Form1.Grid1.TextMatrix(j, 1) = "" Then Exit For
  For i = 1 To MAX
    If Form1.Grid1.TextMatrix(j, i) = "1" Then
      LamDa = Grid1.TextMatrix(j + ((k - 1) * CNT), 3) + (stage *
Grid1.TextMatrix(j + ((k - 1) * CNT), 2))
      PoverLoad = PoverLoad + Configuration.G1.TextMatrix(i - 1, 2)
    End If
  Next
  setcolor = vbWhite

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If PoverLoad < Configuration.List1.List(k - 1) Then
  LamDa = 0
  setcolor = vbYellow
End If
Grid1.TextMatrix(j + ((k - 1) * CNT), 1) = Round(LamDa, 9)
Grid1.COL = 1
Grid1.ROW = (j + (k - 1) * CNT)
Grid1.CellBackColor = setcolor
Next

For j = 1 To CNT - 1      '===== P ref. to GEN
  For i = 1 To MAX
    P = 0

    If Form1.Grid1.TextMatrix(j, i) <> "0" Then
      P = (Grid1.TextMatrix(j + ((k - 1) * CNT), 1) -
Configuration.G1.TextMatrix(i - 1, 4)) / Configuration.G1.TextMatrix(i - 1, 3)
      setcolor = vbWhite
      If Grid1.TextMatrix(j + ((k - 1) * CNT), 1) = 0 Then
        P = 0
        setcolor = vbYellow
      End If
    End If
    Grid1.TextMatrix(j + ((k - 1) * CNT), 3 + i) = Round(P, 9)
    If P > Val(Configuration.G1.TextMatrix(i - 1, 2)) And
Int(Val(Grid1.TextMatrix(j + ((k - 1) * CNT), 3 + i))) <> 0 Then
      setcolor = vbRed
    End If
    Grid1.COL = 3 + i
    Grid1.ROW = (j + (k - 1) * CNT)
    Grid1.CellBackColor = setcolor
    setcolor = vbYellow
  Next
Next

For j = 1 To CNT - 1      '===== F ref. to GEN
  For i = 1 To MAX
    f = 0

    If Form1.Grid1.TextMatrix(j, i) <> "0" Then
      f = ((Configuration.G1.TextMatrix(i - 1, 3) / 2) * (Grid1.TextMatrix(j + ((k
- 1) * CNT), 3 + i) ^ 2)) + (Configuration.G1.TextMatrix(i - 1, 4) *
Grid1.TextMatrix(j + ((k - 1) * CNT), i + 3)) + Configuration.G1.TextMatrix(i - 1, 5)

      setcolor = vbWhite
      If Grid1.TextMatrix(j + ((k - 1) * CNT), 3 + i) = 0 Then
        f = 0
        setcolor = vbYellow
      End If
    End If
  Next
Next

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If
    If k = 2 Then
        f = f
    End If
End If
Grid1.TextMatrix(j + ((k - 1) * CNT), 3 + MAX + i) = Round(f, 9)
If Int(f) = 0 Then
    setcolor = vbYellow
End If
Grid1.COL = (3 + MAX + i)
Grid1.ROW = (j + ((k - 1) * CNT))
Grid1.CellBackColor = setcolor
If Int(f) = 0 Then
    setcolor = vbWhite
End If
Next
Next

For j = 1 To CNT - 1
    sumUnitCost = 0
    sumA = 0
    For i = 1 To MAX
        sumA = sumA + (Val(Grid1.TextMatrix(j + ((k - 1) * CNT), 3 + MAX + i)) /
1)
    Next
    sumUnitCost = MAX * sumA
    Grid1.COL = 4 + (MAX * 2)
    Grid1.ROW = (j + ((k - 1) * CNT))
    If sumUnitCost = 0 Then
        Grid1.CellBackColor = vbYellow
    Else
        Grid1.CellBackColor = vbWhite
    End If
    Grid1.TextMatrix(j + ((k - 1) * CNT), 4 + (MAX * 2)) = Round(sumUnitCost, 9)
    Grid1.TextMatrix(j + ((k - 1) * CNT), (4 + (MAX * 2)) + 1) = k 'insert code for
usage in final calculate form module
Next
Next '===== end stage

Call ReSortingStage
End Sub

Private Sub Form_Unload(Cancel As Integer)
Form1.Show

End Sub

Private Sub ReSortingStage()

```

```

Dim MaxR As Integer, MaxC As Integer
Dim notEndSlide As Boolean
Dim countDecreaseRow As Integer
MaxR = Grid1.Rows - 1
MaxC = Grid1.Cols - 1
notEndSlide = True
  For i = 1 To MaxR - 1

    If Grid1.TextMatrix(i, MaxC) = "" Then
      chkCpy = i
      Do
        chkCpy = chkCpy + 1
        If Not (chkCpy <= MaxR) Then
          notEndSlide = False
          Exit Do
        End If
      Loop While Grid1.TextMatrix(chkCpy, MaxC) = ""
      If notEndSlide Then
        For j = 0 To MaxC
          Grid1.TextMatrix(i, j) = Grid1.TextMatrix(chkCpy, j)
          Grid1.TextMatrix(chkCpy, j) = ""
          Grid1.COL = j
          Grid1.ROW = chkCpy
          If Grid1.CellBackColor = vbYellow Then
            Grid1.CellBackColor = vbWhite
            Grid1.ROW = i
            Grid1.CellBackColor = vbYellow
          End If
        Next
      End If
    End If
  Next
  For i = MaxR To 0 Step -1
    If Grid1.TextMatrix(i, MaxC) <> "" Then
      Exit For
    End If
    countDecreaseRow = countDecreaseRow + 1
  Next

  Grid1.Rows = Grid1.Rows - countDecreaseRow

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim celX As Integer, celY As Integer
Dim COL As Integer, ROW As Integer
Dim ConfigFile As New FileSystemObject, File
Dim txtRead As TextStream

```

```

Private Sub Command1_Click()
Dim tmpDAT() As String
Dim tmpCount As Integer
commSAVE.ShowSave
If commSAVE.FileName <> "" Then
Set File = ConfigFile.CreateTextFile(commSAVE.FileName, True)
File.WriteLine ("VER 1.0")
File.WriteLine (MAX)
ReDim tmpDAT(MAX - 1)
File.WriteLine ("====Loading limits====")
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 1)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 2)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====Fuel cost parameters====")
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 3)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 4)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 5)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====start up cost====")
For i = 0 To MAX - 1
tmpDAT(i) = G1.TextMatrix(i, 6)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====shut down cost====")
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 7)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====Loading rates====")
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 8)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====De-Loading Rates====")
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 9)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====min up time====")
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 10)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 11)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====min down time====")
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 12)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
For i = 0 To MAX - 1
    tmpDAT(i) = G1.TextMatrix(i, 13)
Next
tmp = Join(tmpDAT, ",")
File.WriteLine (tmp)
File.WriteLine ("====Stage load curve====") 'Line 24
tmpCount = List1.ListCount
tmp = ""
For i = 1 To tmpCount
    If i < tmpCount Then
        tmp = tmp & List1.List(i - 1) & ","
    Else
        tmp = tmp & List1.List(i - 1)
    End If

```

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของบริษัทฯ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If
  Next
  File.WriteLine (tmp) ' Line 25
  File.Close
End If
End Sub

```

```

Private Sub Command2_Click()
Dim tmpDAT() As String
commSAVE.ShowOpen
If commSAVE.FileName <> "" Then
'If True Then
  Set txtRead = ConfigFile.OpenTextFile(commSAVE.FileName)
  'Set txtRead = ConfigFile.OpenTextFile("c:\l.cfg")
  txtRead.SkipLine '1
  MAX = Val(txtRead.ReadLine) '2
  Form1.Text1 = MAX
  Call Form_Load
  ReDim tmpDAT(MAX)
  txtRead.SkipLine '3
  tmp = txtRead.ReadLine '4
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1
    G1.TextMatrix(i, 1) = tmpDAT(i)
  Next
  tmp = txtRead.ReadLine '5
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1
    G1.TextMatrix(i, 2) = tmpDAT(i)
  Next
  txtRead.SkipLine '6
  tmp = txtRead.ReadLine '7
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1
    G1.TextMatrix(i, 3) = tmpDAT(i)
  Next
  tmp = txtRead.ReadLine '8
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1
    G1.TextMatrix(i, 4) = tmpDAT(i)
  Next
  tmp = txtRead.ReadLine '9
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1
    G1.TextMatrix(i, 5) = tmpDAT(i)
  Next
  txtRead.SkipLine '10
  tmp = txtRead.ReadLine '11
  tmpDAT = Split(tmp, ",")
  For i = 0 To MAX - 1

```

```

    G1.TextMatrix(i, 6) = tmpDAT(i)
Next
txtRead.SkipLine '12
tmp = txtRead.ReadLine '13
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 7) = tmpDAT(i)
Next
txtRead.SkipLine '14
tmp = txtRead.ReadLine '15
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 8) = tmpDAT(i)
Next
txtRead.SkipLine '16
tmp = txtRead.ReadLine '17
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 9) = tmpDAT(i)
Next
txtRead.SkipLine '18
tmp = txtRead.ReadLine '19
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 10) = tmpDAT(i)
Next
tmp = txtRead.ReadLine '20
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 11) = tmpDAT(i)
Next
txtRead.SkipLine '21
tmp = txtRead.ReadLine '22
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 12) = tmpDAT(i)
Next
tmp = txtRead.ReadLine '23
tmpDAT = Split(tmp, ",")
For i = 0 To MAX - 1
    G1.TextMatrix(i, 13) = tmpDAT(i)
Next
txtRead.SkipLine ' 24
tmp = txtRead.ReadLine '25
i = 1
Do While i < Len(tmp)
    If Mid(tmp, i, 1) = "," Then
        CountChr = CountChr + 1
    End If
    i = i + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Loop
ReDim tmpDAT(CountChr + 1)
tmpDAT = Split(tmp, ",")
List1.Clear
List2.Clear
For i = 0 To CountChr
    List2.AddItem i + 1
    List1.AddItem tmpDAT(i)
Next
txtRead.Close
End If
Call calStagePerHour

```

```
End Sub
```

```

Private Sub Command3_Click()
List1.AddItem Int(Val(InpStage))
tmp = List1.ListCount
List2.Clear
For i = 1 To tmp
    List2.AddItem i
Next
Call calStagePerHour
End Sub

```

```

Private Sub Command4_Click()
Hide
End Sub

```

```

Private Sub Form_Load()
G1.Rows = MAX
For i = 0 To MAX - 1
    G1.TextMatrix(i, 0) = "GEN" & i + 1
Next
End Sub

```

```

Private Sub G1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 And G1.COL <> 0 Then
    COL = G1.COL
    ROW = G1.ROW
    inp.Visible = True
    If G1.TextMatrix(ROW, COL) <> "" Then inp.Text = G1.TextMatrix(ROW, COL)
    inp.Left = celX
    inp.Top = celY
    inp.Height = G1.CellHeight
    inp.Width = G1.CellWidth
    inp.SetFocus
    inp.SelStart = 0
    inp.SelLength = Len(inp.Text)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

End Sub

```
Private Sub G1_RowColChange()
'Caption = G1.Row & "-" & G1.Col
celX = G1.CellLeft + G1.Left
celY = G1.CellTop + G1.Top
```

End Sub

```
Private Sub inp_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
    G1.TextMatrix(ROW, COL) = Val(inp.Text)
```

```
    inp.Text = ""
```

```
    inp.Visible = False
```

```
    G1.SetFocus
```

```
ElseIf KeyAscii = 27 Then
```

```
    inp.Text = ""
```

```
    inp.Visible = False
```

```
    G1.SetFocus
```

```
End If
```

```
End Sub
```

```
Private Sub List1_Click()
```

```
selectDel = List1.ListIndex
```

```
List1.RemoveItem (selectDel)
```

```
List2.RemoveItem (selectDel)
```

```
tmp = List1.ListCount
```

```
List2.Clear
```

```
For i = 1 To tmp
```

```
    List2.AddItem i
```

```
Next
```

```
Call calStagePerHour
```

```
End Sub
```

```
Private Sub List2_Click()
```

```
selectDel = List2.ListIndex
```

```
List1.RemoveItem (selectDel)
```

```
List2.RemoveItem (selectDel)
```

```
tmp = List1.ListCount
```

```
List2.Clear
```

```
For i = 1 To tmp
```

```
    List2.AddItem i
```

```
Next
```

```
Call calStagePerHour
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub calStagePerHour()
```

```
StageHour = Round(24 / List1.ListCount, 2)
```

```
DispStageHr.Caption = StageHour & " Hour/stage"
```

```
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim countCompare As Long
Dim MoveStatus As Integer
Dim cntStage As Integer
Dim LowCost As Double
Dim MaxR As Integer
Dim countNullRow As Integer
Dim I1 As Integer, I2 As Integer, i3 As Integer, i4 As Integer, i5 As Integer, i6 As
Integer, i7 As Integer, i8 As Integer, i9 As Integer, i10 As Integer, i11 As Integer, i12
As Integer, i13 As Integer, i14 As Integer, i15 As Integer, i16 As Integer, i17 As
Integer, i18 As Integer, i19 As Integer, i20 As Integer, i21 As Integer, i22 As Integer,
i23 As Integer, i24 As Integer
Sub DispPath(x As Integer)

```

```
Select Case x
```

```
Case 1
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0)
```

```
Case 2
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0)
```

```
Case 3
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0)
```

```
Case 4
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"
& Grid1(4).TextMatrix(i5, 0)
```

```
Case 5
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0)
```

```
Case 6
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &
Grid1(6).TextMatrix(i7, 0)
```

```
Case 7
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0)
```

```
Case 8
```

```
Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0)

Case 9

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0)

Case 10

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" &  
Grid1(10).TextMatrix(i11, 0)

Case 11

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" &  
Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0)

Case 12

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" &  
Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" &  
Grid1(12).TextMatrix(i13, 0)

Case 13

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" &  
Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" &  
Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0)

Case 14

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"  
& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" &  
Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" &  
Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" &  
Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" &  
Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" &  
Grid1(14).TextMatrix(i15, 0)

Case 15

Label1.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2,  
0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->"

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาวิจัยเท่านั้น เมื่อผู้จัดทำเห็นประโยชน์ในการนำ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

& Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0)

Case 16

Label1.Caption = Grid1(0).TextMatrix(i1, 0) & "->" & Grid1(1).TextMatrix(i2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0)

Case 17

Label1.Caption = Grid1(0).TextMatrix(i1, 0) & "->" & Grid1(1).TextMatrix(i2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0)

Case 18

Label1.Caption = Grid1(0).TextMatrix(i1, 0) & "->" & Grid1(1).TextMatrix(i2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0)

Case 19

Label1.Caption = Grid1(0).TextMatrix(i1, 0) & "->" & Grid1(1).TextMatrix(i2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0) & "->" & Grid1(19).TextMatrix(i20, 0)

Case 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Labell.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0) & "->" & Grid1(19).TextMatrix(i20, 0) & "->" & Grid1(20).TextMatrix(i21, 0)

Case 21

Labell.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0) & "->" & Grid1(19).TextMatrix(i20, 0) & "->" & Grid1(20).TextMatrix(i21, 0) & "->" & Grid1(21).TextMatrix(i22, 0)

Case 22

Labell.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0) & "->" & Grid1(19).TextMatrix(i20, 0) & "->" & Grid1(20).TextMatrix(i21, 0) & "->" & Grid1(21).TextMatrix(i22, 0) & "->" & Grid1(22).TextMatrix(i23, 0)

Case 23

Labell.Caption = Grid1(0).TextMatrix(I1, 0) & "->" & Grid1(1).TextMatrix(I2, 0) & "->" & Grid1(2).TextMatrix(i3, 0) & "->" & Grid1(3).TextMatrix(i4, 0) & "->" & Grid1(4).TextMatrix(i5, 0) & "->" & Grid1(5).TextMatrix(i6, 0) & "->" & Grid1(6).TextMatrix(i7, 0) & "->" & Grid1(7).TextMatrix(i8, 0) & "->" & Grid1(8).TextMatrix(i9, 0) & "->" & Grid1(9).TextMatrix(i10, 0) & "->" & Grid1(10).TextMatrix(i11, 0) & "->" & Grid1(11).TextMatrix(i12, 0) & "->" & Grid1(12).TextMatrix(i13, 0) & "->" & Grid1(13).TextMatrix(i14, 0) & "->" & Grid1(14).TextMatrix(i15, 0) & "->" & Grid1(15).TextMatrix(i16, 0) & "->" & Grid1(16).TextMatrix(i17, 0) & "->" & Grid1(17).TextMatrix(i18, 0) & "->" & Grid1(18).TextMatrix(i19, 0) & "->" & Grid1(19).TextMatrix(i20, 0) & "->" & Grid1(20).TextMatrix(i21, 0) & "->" & Grid1(21).TextMatrix(i22, 0) & "->" & Grid1(22).TextMatrix(i23, 0) & "->" & Grid1(23).TextMatrix(i24, 0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Select

End Sub

Function RampRateUp(x As Integer) As Boolean

Dim result As Boolean

countCompare = countCompare + 1

DoEvents

Select Case x

Case 2

result = CalRampRate2(I1, I2)

Case 3

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3)

Case 4

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4)

Case 5

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5)

Case 6

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6)

Case 7

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7)

Case 8

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7) And CalRampRate2(i7, i8)

Case 9

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9)

Case 10

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
CalRampRate2(i9, i10)

Case 11

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
CalRampRate2(i9, i10) And CalRampRate2(i10, i11)

Case 12

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ผู้ที่เห็นชอบใช้เอกสารฉบับนี้เป็นการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Case 13

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13)

## Case 14

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14)

## Case 15

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14, i15)

## Case 16

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14, i15) And CalRampRate2(i15, i16)

## Case 17

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14, i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17)

## Case 18

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14, i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And CalRampRate2(i17, i18)

## Case 19

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12) And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14, i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And CalRampRate2(i17, i18) And CalRampRate2(i18, i19)

## Case 20

เอกสารนี้เผยแพร่โดยศูนย์บริการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
 CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
 CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
 CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)  
 And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14,  
 i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And  
 CalRampRate2(i17, i18) And CalRampRate2(i18, i19) And CalRampRate2(i19, i20)

Case 21

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
 CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
 CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
 CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)  
 And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14,  
 i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And  
 CalRampRate2(i17, i18) And CalRampRate2(i18, i19) And CalRampRate2(i19, i20)  
 And CalRampRate2(i20, i21)

Case 22

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
 CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
 CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
 CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)  
 And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14,  
 i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And  
 CalRampRate2(i17, i18) And CalRampRate2(i18, i19) And CalRampRate2(i19, i20)  
 And CalRampRate2(i20, i21) And CalRampRate2(i21, i22)

Case 23

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
 CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
 CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
 CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)  
 And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14,  
 i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And  
 CalRampRate2(i17, i18) And CalRampRate2(i18, i19) And CalRampRate2(i19, i20)  
 And CalRampRate2(i20, i21) And CalRampRate2(i21, i22) And CalRampRate2(i22,  
 i23)

Case 24

result = CalRampRate2(I1, I2) And CalRampRate2(I2, i3) And  
 CalRampRate2(i3, i4) And CalRampRate2(i4, i5) And CalRampRate2(i5, i6) And  
 CalRampRate2(i6, i7) And CalRampRate2(i7, i8) And CalRampRate2(i8, i9) And  
 CalRampRate2(i9, i10) And CalRampRate2(i10, i11) And CalRampRate2(i11, i12)  
 And CalRampRate2(i12, i13) And CalRampRate2(i13, i14) And CalRampRate2(i14,  
 i15) And CalRampRate2(i15, i16) And CalRampRate2(i16, i17) And  
 CalRampRate2(i17, i18) And CalRampRate2(i18, i19) And CalRampRate2(i19, i20)  
 And CalRampRate2(i20, i21) And CalRampRate2(i21, i22) And CalRampRate2(i22,  
 i23) And CalRampRate2(i23, i24)

End Select

RampRateUp = result

End Function

Private Sub Command1\_Click()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Timer1.Enabled = True
countCompare = 0
For I1 = 0 To MaxR - 1 '==1
    Sum = 0
    If Val(Grid1(0).TextMatrix(I1, 1)) = 0 Then Exit For
    For I2 = 0 To MaxR - 1 '==2
        If Val(Grid1(1).TextMatrix(I2, 1)) = 0 Then Exit For
        If cntStage = 1 Then
            If RampRateUp(2) Then Exit For
            Sum = Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1
            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(1)
            ElseIf Sum < LowCost And Sum <> 0 Then
                LowCost = Sum
                Call DispPath(1)
            End If
        End If
    For i3 = 0 To MaxR - 1 '==3
        If cntStage = 2 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or
Val(Grid1(2).TextMatrix(i3, 1)) = 0 Then Exit For
        If cntStage = 3 Then
            If RampRateUp(3) Then Exit For
            Sum = Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1
            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(2)
            ElseIf Sum < LowCost And Sum <> 0 Then
                LowCost = Sum
                Call DispPath(2)
            End If
        End If
    For i4 = 0 To MaxR - 1 '==4
        If cntStage = 3 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or
Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Then
Exit For
        If cntStage = 4 Then
            If RampRateUp(4) Then Exit For
            Sum = Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1
            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(3)
            ElseIf Sum < LowCost And Sum <> 0 Then
                LowCost = Sum
                Call DispPath(3)
            End If
        End If
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
For i5 = 0 To MaxR - 1 '====5

    If cntStage = 4 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or
    Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
    Val(Grid1(4).TextMatrix(i5, 1)) = 0 Then Exit For
        If cntStage = 5 Then
            If RampRateUp(5) Then Exit For
            Sum = Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1
            + Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
            Grid1(4).TextMatrix(i5, 1) / 1
            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(4)
            ElseIf Sum < LowCost And Sum <> 0 Then
                LowCost = Sum
                Call DispPath(4)
            End If
        End If
    End If
    For i6 = 0 To MaxR - 1 '====6
        If cntStage = 5 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or
        Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
        Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Then
        Exit For
            If cntStage = 6 Then
                If RampRateUp(6) Then Exit For
                Sum = Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2,
                1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
                Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1
                If LowCost = 0 Then
                    LowCost = Sum
                    Call DispPath(5)
                ElseIf Sum < LowCost And Sum <> 0 Then
                    LowCost = Sum
                    Call DispPath(5)
                End If
            End If
        End If
        Text1 = LowCost
    End If
    For i7 = 0 To MaxR - 1 '====7

```

```

        If cntStage = 6 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or
        Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
        Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
        Val(Grid1(6).TextMatrix(i7, 1)) = 0 Then Exit For
            If cntStage = 7 Then
                If RampRateUp(7) Then Exit For
                Sum = Grid1(0).TextMatrix(I1, 1) / 1 +
                Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
                Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
                Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1

```

```

If LowCost = 0 Then
  LowCost = Sum
  Call DispPath(6)
ElseIf Sum < LowCost And Sum <> 0 Then
  LowCost = Sum
  Call DispPath(6)
End If
End If
For i8 = 0 To MaxR - 1 '=====8

```

```

  If cntStage = 7 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or
  Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
  Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
  Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Then
  Exit For

```

```

  If cntStage = 8 Then
    Sum = Grid1(0).TextMatrix(i1, 1) / 1 +
    Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
    Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
    Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
    Grid1(7).TextMatrix(i8, 1) / 1
    If LowCost = 0 Then
      If RampRateUp(8) Then Exit For
      LowCost = Sum
      Call DispPath(7)
    ElseIf Sum < LowCost And Sum <> 0 Then
      LowCost = Sum
      Call DispPath(7)
    End If
  End If
  For i9 = 0 To MaxR - 1 '=====9

```

```

    If cntStage = 8 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or
    Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
    Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
    Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
    Val(Grid1(8).TextMatrix(i9, 1)) = 0 Then Exit For

```

```

    If cntStage = 9 Then
      If RampRateUp(9) Then Exit For
      Sum = Grid1(0).TextMatrix(i1, 1) / 1 +
      Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
      Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
      Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
      Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1

```

```

    If LowCost = 0 Then
      LowCost = Sum
      Call DispPath(8)
    ElseIf Sum < LowCost And Sum <> 0 Then
      LowCost = Sum
      Call DispPath(8)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
End If
For i10 = 0 To MaxR - 1 '=====10

    If cntStage = 9 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or
Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Then
Exit For

    If cntStage = 10 Then
    If RampRateUp(10) Then Exit For
        Sum = Grid1(0).TextMatrix(i1, 1) / 1 +
Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1
        If LowCost = 0 Then
            LowCost = Sum
            Call DispPath(9)
        ElseIf Sum < LowCost And Sum > 0 Then
            LowCost = Sum
            Call DispPath(9)
        End If
    End If
    For i11 = 0 To MaxR - 1 '=====11

        If cntStage = 10 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0
Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
Val(Grid1(10).TextMatrix(i11, 1)) = 0 Then Exit For
        If cntStage = 11 Then
            If RampRateUp(11) Then Exit For
                Sum = Grid1(0).TextMatrix(i1, 1) / 1 +
Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1
                If LowCost = 0 Then
                    LowCost = Sum
                    Call DispPath(10)
                ElseIf Sum < LowCost And Sum > 0 Then
                    LowCost = Sum
                    Call DispPath(10)
                End If
            End If
        End If
    End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

For i12 = 0 To MaxR - 1 '=====12
    If cntStage = 11 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0
    Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
    Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
    Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
    Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
    Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(11).TextMatrix(i12, 1)) = 0
    Then Exit For
    If cntStage = 12 Then
        If RampRateUp(12) Then Exit For
        Sum = Grid1(0).TextMatrix(i11, 1) / 1 +
Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1
        If LowCost = 0 Then
            LowCost = Sum
            Call DispPath(11)
        ElseIf Sum < LowCost And Sum <> 0 Then
            LowCost = Sum
            Call DispPath(11)
        End If
    End If
    Call sub_command1
Next 'end=====12
Next 'end=====11
Next 'end=====10
Next 'end=====9
Next 'end=====8
Next 'end=====7
Next 'end=====6
Next 'end=====5
Next 'end=====4
Next 'end=====3
Next 'end=====2
Next 'end=====1

Text1 = LowCost
MsgBox "complete"
Caption = "finalCalculate"
Timer1.Enabled = False
End Sub
Sub sub_command1()
    For i13 = 0 To MaxR - 1 '=====13

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    If cntStage = 12 Or Val(Grid1(1).TextMatrix(I2, 1))
    = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
    Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
    Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
    Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
    Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(12).TextMatrix(i6, 1)) = 0 Or
    Val(Grid1(13).TextMatrix(i6, 1)) = 0 Then Exit For

```

```

    If cntStage = 13 Then

```

```

        If RampRateUp(13) Then Exit For

```

```

        Sum = Grid1(0).TextMatrix(I1, 1) / 1 +

```

```

        Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +

```

```

        Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +

```

```

        Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +

```

```

        Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +

```

```

        Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +

```

```

        Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1

```

```

        If LowCost = 0 Then

```

```

            LowCost = Sum

```

```

            Call DispPath(12)

```

```

        Else If Sum < LowCost And Sum <> 0 Then

```

```

            LowCost = Sum

```

```

            Call DispPath(12)

```

```

        End If

```

```

    End If

```

```

    For i14 = 0 To MaxR - 1 '====14

```

```

        If cntStage = 13 Or Val(Grid1(1).TextMatrix(I2,
        1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) =
        0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
        Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
        Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
        Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or
        Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or Val(Grid1(13).TextMatrix(i14, 1)) = 0
        Then Exit For

```

```

        If cntStage = 14 Then

```

```

            If RampRateUp(14) Then Exit For

```

```

            Sum = Grid1(0).TextMatrix(I1, 1) / 1 +

```

```

            Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +

```

```

            Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +

```

```

            Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +

```

```

            Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +

```

```

            Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +

```

```

            Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +

```

```

            Grid1(13).TextMatrix(i14, 1) / 1

```

```

            If LowCost = 0 Then

```

```

                LowCost = Sum

```

```

                Call DispPath(13)

```

```

            Else If Sum < LowCost And Sum <> 0

```

```

        Then

```

```

            LowCost = Sum

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้เผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Call DispPath(13)
End If
End If
For i15 = 0 To MaxR - 1 '=====15

    If cntStage = 14 Or Val(Grid1(1).TextMatrix(I2,
1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) =
0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or
Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or
Val(Grid1(14).TextMatrix(i15, 1)) = 0 Then Exit For

        If cntStage = 15 Then
            If RampRateUp(15) Then Exit For
            Sum = Grid1(0).TextMatrix(I1, 1) / 1 +
Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +
Grid1(13).TextMatrix(i14, 1) / 1 + Grid1(14).TextMatrix(i15, 1) / 1

                If LowCost = 0 Then
                    LowCost = Sum
                    Call DispPath(14)
                ElseIf Sum < LowCost And Sum > 0
                    LowCost = Sum
                    Call DispPath(14)
                End If
            End If
        End If
    For i16 = 0 To MaxR - 1 '=====16

        If cntStage = 15 Or
Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Then Exit For

            If cntStage = 16 Then
                If RampRateUp(16) Then Exit For
                Sum = Grid1(0).TextMatrix(I1, 1) / 1
+ Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +
Grid1(13).TextMatrix(i14, 1) / 1 + Grid1(14).TextMatrix(i15, 1) / 1 +
Grid1(15).TextMatrix(i16, 1) / 1

                    If LowCost = 0 Then
                        LowCost = Sum
                        Call DispPath(14)
                    ElseIf Sum < LowCost And Sum > 0
                        LowCost = Sum
                        Call DispPath(14)
                    End If
                End If
            End If
        End If
    End For
End For

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +
Grid1(13).TextMatrix(i14, 1) / 1 + Grid1(14).TextMatrix(i15, 1) / 1 +
Grid1(15).TextMatrix(i16, 1) / 1

```

```

If LowCost = 0 Then
  LowCost = Sum
  Call DispPath(15)
Elseif Sum < LowCost And Sum <> 0

```

```
Then
```

```

  LowCost = Sum
  Call DispPath(15)

```

```
End If
```

```
End If
```

```
For i17 = 0 To MaxR - 1 '=====17
```

```
  If cntStage = 16 Or
```

```

Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0

```

```
Then Exit For
```

```
  If cntStage = 17 Then
```

```

    If RampRateUp(17) Then Exit For
    Sum = Grid1(0).TextMatrix(i1, 1) /

```

```

1 + Grid1(1).TextMatrix(i2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +
Grid1(13).TextMatrix(i14, 1) / 1 + Grid1(14).TextMatrix(i15, 1) / 1 +
Grid1(15).TextMatrix(i16, 1) / 1 + Grid1(16).TextMatrix(i17, 1) / 1

```

```
  If LowCost = 0 Then
```

```

    LowCost = Sum
    Call DispPath(16)

```

```
  Elseif Sum < LowCost And Sum
```

```
<> 0 Then
```

```

    LowCost = Sum
    Call DispPath(16)

```

```
  End If
```

```
End If
```

```
For i18 = 0 To MaxR - 1 '=====18
```

```
  If cntStage = 17 Or
```

```

Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิอวกาศ (สอภ.) ไม่อนุญาตให้นำไปใช้โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or
Val(Grid1(17).TextMatrix(i18, 1)) = 0 Then Exit For
    If cntStage = 18 Then
        If RampRateUp(18) Then Exit
For
    Sum = Grid1(0).TextMatrix(I1,
1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 + Grid1(2).TextMatrix(i3, 1) / 1 +
Grid1(3).TextMatrix(i4, 1) / 1 + Grid1(4).TextMatrix(i5, 1) / 1 +
Grid1(5).TextMatrix(i6, 1) / 1 + Grid1(6).TextMatrix(i7, 1) / 1 +
Grid1(7).TextMatrix(i8, 1) / 1 + Grid1(8).TextMatrix(i9, 1) / 1 +
Grid1(9).TextMatrix(i10, 1) / 1 + Grid1(10).TextMatrix(i11, 1) / 1 +
Grid1(11).TextMatrix(i12, 1) / 1 + Grid1(12).TextMatrix(i13, 1) / 1 +
Grid1(13).TextMatrix(i14, 1) / 1 + Grid1(14).TextMatrix(i15, 1) / 1 +
Grid1(15).TextMatrix(i16, 1) / 1 + Grid1(16).TextMatrix(i17, 1) / 1 +
Grid1(17).TextMatrix(i18, 1) / 1
    If LowCost = 0 Then
        LowCost = Sum
        Call DispPath(17)
    Elseif Sum < LowCost And Sum
    < 0 Then
        LowCost = Sum
        Call DispPath(17)
    End If
    End If
For i19 = 0 To MaxR - 1 '=====19
    If cntStage = 18 Or
Val(Grid1(1).TextMatrix(I2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or
Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or Val(Grid1(18).TextMatrix(i19, 1)) = 0
Then Exit For
    If cntStage = 19 Then
        If RampRateUp(19) Then Exit
For
    Sum =
Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +
Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +

```

```

Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1

If LowCost = 0 Then
    LowCost = Sum
    Call DispPath(18)
Elseif Sum < LowCost And

Sum <> 0 Then

    LowCost = Sum
    Call DispPath(18)
End If
End If
For i20 = 0 To MaxR - 1 '=====20

    If cntStage = 19 Or
Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or
Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or Val(Grid1(18).TextMatrix(i19, 1)) = 0 Or
Val(Grid1(19).TextMatrix(i20, 1)) = 0 Then Exit For

    If cntStage = 20 Then
        If RampRateUp(20) Then

Exit For

            Sum =
Grid1(0).TextMatrix(i1, 1) / 1 + Grid1(1).TextMatrix(i2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +
Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +
Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1 + Grid1(19).TextMatrix(i20, 1) / 1

            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(19)
            Elseif Sum < LowCost And

Sum <> 0 Then

                LowCost = Sum
                Call DispPath(19)
            End If
        End If
    End For

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If
For i21 = 0 To MaxR - 1 '====21

    If cntStage = 20 Or
Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or
Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or
Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or Val(Grid1(18).TextMatrix(i19, 1)) = 0 Or
Val(Grid1(19).TextMatrix(i20, 1)) = 0 Or Val(Grid1(20).TextMatrix(i21, 1)) = 0
Then Exit For

    If cntStage = 21 Then
        If RampRateUp(21) Then

            Sum =
Grid1(0).TextMatrix(i1, 1) / 1 + Grid1(1).TextMatrix(i2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +
Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +
Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1 + Grid1(19).TextMatrix(i20, 1) / 1 +
Grid1(20).TextMatrix(i21, 1) / 1

            If LowCost = 0 Then
                LowCost = Sum
                Call DispPath(20)
            ElseIf Sum < LowCost And

Sum < 0 Then

                LowCost = Sum
                Call DispPath(20)
            End If
        End If
    End If
For i22 = 0 To MaxR - 1

'====22

```

```

    If cntStage = 21 Or
Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or
Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or
Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or
Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or
Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or
Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or
Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถนำออกจำหน่ายหรือเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or
Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or Val(Grid1(18).TextMatrix(i19, 1)) = 0 Or
Val(Grid1(19).TextMatrix(i20, 1)) = 0 Or Val(Grid1(20).TextMatrix(i21, 1)) = 0 Or
Val(Grid1(21).TextMatrix(i22, 1)) = 0 Then Exit For

```

```

If cntStage = 22 Then
  If RampRateUp(22)

```

```

Then Exit For

```

```

Sum =

```

```

Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +
Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +
Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1 + Grid1(19).TextMatrix(i20, 1) / 1 +
Grid1(20).TextMatrix(i21, 1) / 1 + Grid1(21).TextMatrix(i22, 1) / 1

```

```

If LowCost = 0 Then

```

```

  LowCost = Sum

```

```

  Call DispPath(21)

```

```

Elseif Sum < LowCost

```

```

And Sum <> 0 Then

```

```

  LowCost = Sum

```

```

  Call DispPath(21)

```

```

End If

```

```

End If

```

```

For i23 = 0 To MaxR - 1

```

```

'=====23

```

```

If cntStage = 22 Or Val(Grid1(1).TextMatrix(I2, 1)) = 0

```

```

Or Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or
Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or
Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or
Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or
Val(Grid1(18).TextMatrix(i19, 1)) = 0 Or Val(Grid1(19).TextMatrix(i20, 1)) = 0 Or
Val(Grid1(20).TextMatrix(i21, 1)) = 0 Or Val(Grid1(21).TextMatrix(i22, 1)) = 0 Or
Val(Grid1(22).TextMatrix(i23, 1)) = 0 Then Exit For

```

```

If cntStage = 23 Then

```

```

  If RampRateUp(23)

```

```

Then Exit For

```

```

Sum =

```

```

Grid1(0).TextMatrix(I1, 1) / 1 + Grid1(1).TextMatrix(I2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +

```

เอกสารนี้เป็นเอกสารทศวงนวิชาสำหรับกรใช้งานเพอกรศกรศึกษาเท่านั้น เมื่อนุญตใหนำไปใ้ประยอชนด้านกรค้ำ

ไม่ว่ากรณใ้ใด ๆ ทั้งสััน อึกรทั้งหำมมิใ้ดัดแปลงเนือหา และดอแงอแงงึถึงเจ้าขอเอกสารทุกคร้งที่มีกรนำไปใ้

```

Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +
Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1 + Grid1(19).TextMatrix(i20, 1) / 1 +
Grid1(20).TextMatrix(i21, 1) / 1 + Grid1(21).TextMatrix(i22, 1) / 1 +
Grid1(22).TextMatrix(i23, 1) / 1

```

```

If LowCost = 0 Then
  LowCost = Sum
  Call DispPath(22)
ElseIf Sum <

```

```

LowCost And Sum > 0 Then

```

```

  LowCost = Sum
  Call DispPath(22)

```

```

End If

```

```

End If

```

```

For i24 = 0 To MaxR - 1

```

```

'====24

```

```

  If cntStage = 23 Or Val(Grid1(1).TextMatrix(i2, 1)) = 0 Or
  Val(Grid1(2).TextMatrix(i3, 1)) = 0 Or Val(Grid1(3).TextMatrix(i4, 1)) = 0 Or
  Val(Grid1(4).TextMatrix(i5, 1)) = 0 Or Val(Grid1(5).TextMatrix(i6, 1)) = 0 Or
  Val(Grid1(6).TextMatrix(i7, 1)) = 0 Or Val(Grid1(7).TextMatrix(i8, 1)) = 0 Or
  Val(Grid1(8).TextMatrix(i9, 1)) = 0 Or Val(Grid1(9).TextMatrix(i10, 1)) = 0 Or
  Val(Grid1(10).TextMatrix(i11, 1)) = 0 Or Val(Grid1(11).TextMatrix(i12, 1)) = 0 Or
  Val(Grid1(12).TextMatrix(i13, 1)) = 0 Or Val(Grid1(13).TextMatrix(i14, 1)) = 0 Or
  Val(Grid1(14).TextMatrix(i15, 1)) = 0 Or Val(Grid1(15).TextMatrix(i16, 1)) = 0 Or
  Val(Grid1(16).TextMatrix(i17, 1)) = 0 Or Val(Grid1(17).TextMatrix(i18, 1)) = 0 Or
  Val(Grid1(18).TextMatrix(i19, 1)) = 0 Or Val(Grid1(19).TextMatrix(i20, 1)) = 0 Or
  Val(Grid1(20).TextMatrix(i21, 1)) = 0 Or Val(Grid1(21).TextMatrix(i22, 1)) = 0 Or
  Val(Grid1(22).TextMatrix(i23, 1)) = 0 Or Val(Grid1(23).TextMatrix(i24, 1)) = 0
  Then Exit For

```

```

Then

```

```

  RampRateUp(24) Then Exit For

```

```

  If cntStage = 24

```

```

    If

```

```

      Sum =

```

```

Grid1(0).TextMatrix(i1, 1) / 1 + Grid1(1).TextMatrix(i2, 1) / 1 +
Grid1(2).TextMatrix(i3, 1) / 1 + Grid1(3).TextMatrix(i4, 1) / 1 +
Grid1(4).TextMatrix(i5, 1) / 1 + Grid1(5).TextMatrix(i6, 1) / 1 +
Grid1(6).TextMatrix(i7, 1) / 1 + Grid1(7).TextMatrix(i8, 1) / 1 +
Grid1(8).TextMatrix(i9, 1) / 1 + Grid1(9).TextMatrix(i10, 1) / 1 +
Grid1(10).TextMatrix(i11, 1) / 1 + Grid1(11).TextMatrix(i12, 1) / 1 +
Grid1(12).TextMatrix(i13, 1) / 1 + Grid1(13).TextMatrix(i14, 1) / 1 +
Grid1(14).TextMatrix(i15, 1) / 1 + Grid1(15).TextMatrix(i16, 1) / 1 +
Grid1(16).TextMatrix(i17, 1) / 1 + Grid1(17).TextMatrix(i18, 1) / 1 +
Grid1(18).TextMatrix(i19, 1) / 1 + Grid1(19).TextMatrix(i20, 1) / 1 +
Grid1(20).TextMatrix(i21, 1) / 1 + Grid1(21).TextMatrix(i22, 1) / 1 +
Grid1(22).TextMatrix(i23, 1) / 1 + Grid1(23).TextMatrix(i24, 1) / 1

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นจำเป็นต้องนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Then
    If LowCost = 0
        LowCost = Sum
        Call
    ElseIf Sum <
        LowCost = Sum
        Call
    Text1 =
End If
End If
Next 'end=====24
Next 'end=====23
Next 'end=====22
Next 'end=====21
Next 'end=====20
Next 'end=====19
Next 'end=====18
Next 'end=====17
Next 'end=====16
Next 'end=====15
Next 'end=====14
Next 'end=====13
End Sub
Private Sub Form_Load()
Dim ValStage(24) As Integer
Dim i As Integer

cntStage = Configuration.List1.ListCount
For i = 0 To cntStage - 1
    ValStage(i) = Configuration.List1.List(i)
Next

old = Calculate.Grid1.TextMatrix(1, Calculate.Grid1.Cols - 1)
For i = 1 To CNT
    If old < Calculate.Grid1.TextMatrix(i, Calculate.Grid1.Cols - 1) Then
        Exit For
    End If
Next
i = i - 1
MaxR = i

For i = 0 To 23
    Grid1(i).Width = 2300
    Grid1(i).ColWidth(0) = 400
    Grid1(i).ColWidth(1) = 1450

```

```

Grid1(i).Rows = MaxR
If i < cntStage Then
  Grid1(i).Visible = True
  For j = 0 To MaxR - 1
    tmp = (i * MaxR) + j + 1
    If CalPowerMaxRun(j + 1, ValStage(i)) Then 'this position for call sub
procedure which is calculate maxpower enough run
      Grid1(i).TextMatrix(j, 0) = "x" & j + 1
      Grid1(i).TextMatrix(j, 1) = Calculate.Grid1.TextMatrix(tmp,
Calculate.Grid1.Cols - 2)
    End If
  Next
Else
  Grid1(i).Visible = False
End If
Next
Next

'Exit Sub
For i = 0 To cntStage - 1 'for remove null value record from grid
  For j = 0 To MaxR - 1
    If Grid1(i).TextMatrix(j, 1) = "0" Or Grid1(i).TextMatrix(j, 1) = "" Then
      For k = j To MaxR - 1
        If Grid1(i).TextMatrix(k, 1) <> "0" And Grid1(i).TextMatrix(k, 1) <> ""
Then
          Grid1(i).TextMatrix(j, 0) = Grid1(i).TextMatrix(k, 0)
          Grid1(i).TextMatrix(j, 1) = Grid1(i).TextMatrix(k, 1)
          Grid1(i).TextMatrix(k, 0) = ""
          Grid1(i).TextMatrix(k, 1) = ""
          Exit For
        End If
        Grid1(i).TextMatrix(k, 0) = ""
        Grid1(i).TextMatrix(k, 1) = ""
      Next
    End If
  Next
Next
Next

Next

End Sub
Function CalPowerMaxRun(RowOfMustRun As Integer, stage As Integer) As
Boolean ' Load Limited
  Dim i As Integer, result As Boolean
  Dim P As Integer
  P = 0
  For i = 1 To MAX
    If Int(Val(Form1.Grid1.TextMatrix(RowOfMustRun, i))) = 1 Then
      P = P + Int(Val(Configuration.G1.TextMatrix(i - 1, 2)))
    End If
  Next
  result = (P <= LoadLimited)
End Function

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If stage > P Then
    result = False
Else
    result = True
End If
CalPowerMaxRun = result
End Function

```

```

Private Sub Form_Unload(Cancel As Integer)
Hide
Form1.Show
End Sub

```

```

Function CalRampRate2(beginStage As Integer, endStage As Integer) As Boolean
'Grid1(0).TextMatrix(i1, 1) Grid1(1).TextMatrix(i2, 1)
    Dim m As Integer, x As Integer, y As Integer, sumTime As Double, RangePower
As Double
    Dim tmp As Integer
    Dim result As Boolean
    Dim I1 As Integer, I2 As Integer
    I1 = beginStage
    I2 = endStage
    result = True
    m = Len(Grid1(0).TextMatrix(I1, 0))

    If Grid1(0).TextMatrix(I1, 0) = "" Then
        result = False
        Exit Function
    End If
    If Grid1(0).TextMatrix(I2, 0) = "" Then
        result = False
        Exit Function
    End If
    x = Int(Val(Right(Grid1(0).TextMatrix(I1, 0), m - 1)))
    y = Int(Val(Right(Grid1(1).TextMatrix(I2, 0), m - 1)))
    'tmp = x & "++" & y & "###"

```

```

For m = 1 To MAX
    If Form1.Grid1.TextMatrix(x, m) = Form1.Grid1.TextMatrix(y, m) And
Form1.Grid1.TextMatrix(x, m) = "1" Then 'x,y = case m=gen          tmp =
tmp & "1-1#"

```

```

        RangePower = Val(Calculate.Grid1.TextMatrix(I1 + 1, m + 3)) -
Val(Calculate.Grid1.TextMatrix(I2 + 1 + MaxR, m + 3))
        If RangePower <= 0 Then
            RangePower = RangePower * (-1)
            sumTime = (RangePower / Val(Configuration.G1.TextMatrix(m - 1, 8))) /

```

60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    If sumTime > StageHour Then ' chk false if 1->1 have loading rate not
    enough
        result = False
    End If
End If
Elseif Int(Val(Form1.Grid1.TextMatrix(x, m))) <
Int(Val(Form1.Grid1.TextMatrix(y, m))) Then 'tmp = tmp & "0-1#"

    RangePower = Val(Calculate.Grid1.TextMatrix(I1 + 1, m + 3)) -
Val(Calculate.Grid1.TextMatrix(I2 + 1 + MaxR, m + 3))
    RangePower = RangePower * (-1)
    sumTime = (RangePower / Val(Configuration.G1.TextMatrix(m - 1, 8))) /
60 + Val(Configuration.G1.TextMatrix(m - 1, 11))
    If sumTime > StageHour Then ' chk false if 1->1 have loading rate not
    enough
        result = False
    End If

    Elseif Int(Val(Form1.Grid1.TextMatrix(x, m))) >
Int(Val(Form1.Grid1.TextMatrix(y, m))) Then 'tmp = tmp & "1-0#"

    RangePower = Val(Calculate.Grid1.TextMatrix(I1 + 1, m + 3)) -
Val(Calculate.Grid1.TextMatrix(I2 + 1 + MaxR, m + 3))
    RangePower = RangePower * (-1)
    sumTime = (RangePower / Val(Configuration.G1.TextMatrix(m - 1, 8))) /
60 + Val(Configuration.G1.TextMatrix(m - 1, 13))
    If sumTime > StageHour Then ' chk false if 1->1 have loading rate not
    enough
        result = False
    End If

    Else 'tmp = tmp & "0-0#"

    End If
Next
CalRampRate2 = result
End Function

```

```

Private Sub Timer1_Timer()
Caption = "compare = " & countCompare & "times"

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Command1_Click()
'Dim ChkCase As Boolean
CNT = 0
'MAX = Int(Val(Text1.Text)) ' max of generator
Grid1.Clear
LoadGenDefault (MAX)
Grid1.Cols = MAX + 1
For i = 1 To MAX
  Grid1.TextMatrix(0, i) = "GEN" & i ' show header
Next

If MAX <= 14 Then
  Grid1.Rows = 2 ^ MAX + 2 ' detect of over case generate
Else
  Grid1.Rows = 10002
End If

ReDim gen(MAX)
For i = 0 To MAX - 1 'clear status variable generator
  gen(i) = 0
Next

Do While DoEvents '----- begin of generate case
' ChkCase = True
gen(0) = gen(0) + 1
CNT = CNT + 1
If CNT > 2 ^ (MAX) Then Exit Do
If gen(MAX) = 1 Or CNT = 10001 Then
  Exit Do
End If

For i = 0 To MAX - 1 'each state of possible to generator on
  If gen(i) = 2 Then
    gen(i) = 0
    gen(i + 1) = gen(i + 1) + 1
  End If
  If (VarMustRun(i) = 1 And VarMustRun(i) = gen(i)) Or VarMustRun(i) = 0
Then ' must run case
    Grid1.TextMatrix(CNT, 0) = CNT
    Grid1.TextMatrix(CNT, i + 1) = gen(i)
  Else
    CNT = CNT - 1
    Exit For
  End If
Next
Loop ' end of generate case

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If MAX <= 14 Then 'default in every case ,at the end of each generate case
  For i = 0 To MAX
    Grid1.TextMatrix(CNT, i) = ""
  Next
Else
  Grid1.TextMatrix(10001, 0) = ""
End If

```

```

If CNT Mod 2 = 0 Then 'fix bug in generate case over real case
  CNT = CNT - 1
  For i = 0 To MAX
    Grid1.TextMatrix(CNT, i) = ""
  Next
  Grid1.Rows = CNT + 1
Else
  Grid1.Rows = CNT + 1
End If
End Sub

```

```

Private Sub Command2_Click()
  MustRun.Show
  MAX = Int(Val(Text1.Text))
  LoadGenDefault (MAX)
End Sub

```

```

Private Sub Command3_Click()
  Configuration.Show
End Sub

```

```

Private Sub Command4_Click()
  Calculate.Show
  Hide
End Sub

```

```

Private Sub Command5_Click()
  For i = 0 To 1E+15
    Caption = i
    DoEvents
  Next
End Sub

```

```

Private Sub Command6_Click()
  Hide
  FinalCalculate.Show
End Sub

```

```

Private Sub Form_Load()

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAX = Int(Val(Text1))
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
End
```

```
End Sub
```

```
Private Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
```

```
MAX = Int(Val(Text1.Text)) ' max of generator
```

```
End Sub
```

```
Private Sub Clear_Click()
```

```
For i = 1 To List2.ListCount
```

```
    List1.AddItem (List2.List(0))
```

```
    List2.RemoveItem (0)
```

```
Next
```

```
Call SetMustRun
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
List1.Clear
```

```
List2.Clear
```

```
End Sub
```

```
Private Sub List1_Click()
```

```
List2.AddItem (List1.List(List1.ListIndex))
```

```
List1.RemoveItem (List1.ListIndex)
```

```
Call SetMustRun
```

```
Text1 = ""
```

```
For i = 0 To MAX - 1
```

```
    Text1.Text = Text1.Text & VarMustRun(i)
```

```
Next
```

```
End Sub
```

```
Private Sub List2_Click()
```

```
List1.AddItem (List2.List(List2.ListIndex))
```

```
List2.RemoveItem (List2.ListIndex)
```

```
Call SetMustRun
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ข. ตารางแสดงข้อมูลโรงกำเนิดไฟฟ้า

ชื่อโรงไฟฟ้า	รหัส	Heat Rate (Mbtu/Hr)			Max Capacity (MW)	Min Capacity (MW)	Loading rates (MW/Min)	De-loading rates (MW/Min)	Min up time (Hrs.)	Min Down time (Hrs.)	Start up cost บาท
		a	b	c							
โรงไฟฟ้าบางปะกง	BPK-T1	14.85	6755.67	119885.4545	525.6	280	5	5	24	24	491619.54
	BPK-T2	14.85	6788.71	119885.4545	525.6	280	5	5	24	24	491619.54
	BPK-T3	18	6138.99	128635	576	280	5	5	24	24	527194.78
	BPK-T4	18	6141.99	128635	576	280	5	5	24	24	527194.78
	BPK-C1	55.52	-1479.4	408565.1671	351	315	16.5	16.5	12	6	0
		55.52	-1479.4	408565.1671	264	238	13.5	13.5	12	6	0
		55.52	-1479.4	408565.1671	176	154	10.5	10.5	12	6	0
	BPK-C2	30.61	4733.01	393349.1904	351	315	16.5	16.5	12	6	0
		30.61	4733.01	393349.1904	264	238	13.5	13.5	12	6	0
		30.61	4733.01	393349.1904	176	154	10.5	10.5	12	6	0
	BPK-C3	55.52	-1457	408565.1671	313	188.7	12.5	12.5	12	6	695420.25
		55.52	-1457	408565.1671	157	92.8	7.5	7.5	12	6	347615.51
	BPK-C4	30.61	4518.32	393349.1904	313	188.7	12.5	12.5	12	6	695420.25
		30.61	4518.32	393349.1904	157	92.8	7.5	7.5	12	6	347615.51

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โรงไฟฟ้าขอม	KN-C	0.001	3961.38	1684367.9	678	376	8	8	0.5	0.25	815392.07
		0.001	3961.38	1684367.9	512.3	280.8	8	8	0.5	0.25	459261.21
		0.001	3961.38	1684367.9	325.4	181.6	8	8	0.5	0.25	185256.17
		0.001	3961.38	1684367.9	151.1	151.1	6	6	0.5	0.25	54119.78
	KN-T1	0.001	9644.75	100804.7	69.9	50	6	6	5	5	83931.06467
	KN-T2	0.001	9702.5	184078.1	70.2	60	6	6	6	7	83931.06467
โรงไฟฟ้าหนอง	NPO-C1	12.04	4847.55	201073.9342	325	130	8	8	1	1	702821.61
		-2.98	8736.34	291917.4739	165	100	8	8	1	1	351318.06
	NPO-C2	12.04	4977.05	201073.9342	325	130	8	8	1	1	702821.61
		-2.98	8742.27	291917.4739	165	100	8	8	1	1	351318.06
โรงไฟฟ้าวังน้อย	WN-C1	19.53	3263.3	238428.5714	606	320	35	35	1	1.5	1117024.69
		14.69	6260.54	305925.9259	303	160	19	19	1	1.5	558417.73
	WN-C2	19.53	3,214.4397	238428.5714	608	320	35	35	1	1.5	1117024.69
		14.69	6077.1	305925.9259	304	160	19	19	1	1.5	558417.73
	WN-C3	4.9	5236.48	137054.2857	680	340	35	35	1	1.5	1220155.04
		13.22	5813.82	273201.7544	340	170	19	19	1	1.5	610077.52
โรงไฟฟ้าพระนครใต้	SB-T1	2.4	7846.75	93425	186	90	3	3	24	24	496170.1851
	SB-T2	2.4	7841.22	93425	186	90	3	3	24	24	496170.1851

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SB-T3	11.78	7066.13	136287.0968	294	140	10	10	24	24	24	330584.81
SB-T4	11.78	7042.82	136287.0968	265	140	10	10	24	24	24	330584.81
SB-T5	11.78	7015.58	136287.0968	294	140	10	10	24	24	24	330584.81
SB-C1	0	6681.78	96982.0896	312	284	18	18	0.5	0.5	0.5	721155.53
	58.08	224.868	518090.9091	156	142	12	12	0.5	0.5	0.5	360672.38
SB-C2	17.3	3427.87	227718.4466	557	435	18	18	1	1	1	720587.84
	35.6	1926.5	477975	278	210	12	12	1	1	1	360293.92
MM-T4	0	8460.12	174226.6667	140	100	2	2	24	24	24	1140547.574
MM-T5	0	8460.12	174226.6667	140	100	2	2	24	24	24	1140547.574
MM-T6	0	8460.12	174226.6667	140	100	2	2	24	24	24	1140547.574
MM-T7	0	8460.12	174226.6667	140	100	2	2	24	24	24	1140547.574
MM-T8	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
MM-T9	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
MM-T10	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
MM-T11	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
MM-T12	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
MM-T13	0	7479.35	211730	270	162	2.5	2.5	24	24	24	1140547.574
SRT-G1	2.164	9645.77	156185.6	115	40	8	8	1	1	0.5	359842.7595

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้ภายในเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	SRT-G2	2.164	9645.77	156185.6	115	40	8	8	1	0.5	359842.7595
โรงไฟฟ้าลานกระบือ	LKB-G1	882.8	-8638.8	170218	16	3	3	3	1	0.5	33121.19
	LKB-G2	0.001	1648.49	187128.5	16	3	3	3	1	0.5	33121.19
	LKB-G3	11.47	9393.56	94379.4	14	3	3	3	1	0.5	33121.19
	LKB-G4	0.001	4556	152206.3	13	3	3	3	1	0.5	33121.19
	LKB-G5	96.11	4563.8	159539	13	3	3	3	1	0.5	33121.19
	LKB-G6	0.001	7453	131103.3	22	3	3	3	1	0.5	33121.19
	LKB-G9	0.001	9269	79589.5	14	3	3	3	1	0.5	33121.19
	LKB-G11	300.6	3669.86	144487.4	129	40	10	8	1	0.5	33121.19
โรงไฟฟ้าหนองจอก	NCO-G1	2.164	9645.77	156185.6	115	30	8	8	1	0.5	359842.7595
	NCO-G2	2.164	9645.77	156185.6	115	30	8	8	1	0.5	359842.7595
	NCO-G3	2.164	9645.77	156185.6	115	30	8	8	1	0.5	359842.7595
โรงไฟฟ้าระยอง	RY-C1	0.001	7088.66	274286.9	295.2	240	18	18	0.5	0.5	751465.2592
	RY-C2	0.001	6929.01	266902	293.22	240	18	18	0.5	0.5	751465.2592
	RY-C3	0.001	7247.7	250236.2	287.03	240	18	18	0.5	0.5	751465.2592

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้มีการนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	0.001	7247.7	250236.2	143.515	120	12	12	0.5	0.5	315746.9762
RY-C4	0.001	6850.81	383672.6	294.8	240	18	18	0.5	0.5	751465.2592
	0.001	6850.81	383672.6	147.4	120	12	12	0.5	0.5	315746.9762



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. ภาพตัวอย่างโปรแกรม

Configuration Form

Save Config Load Config Close

Set Stage

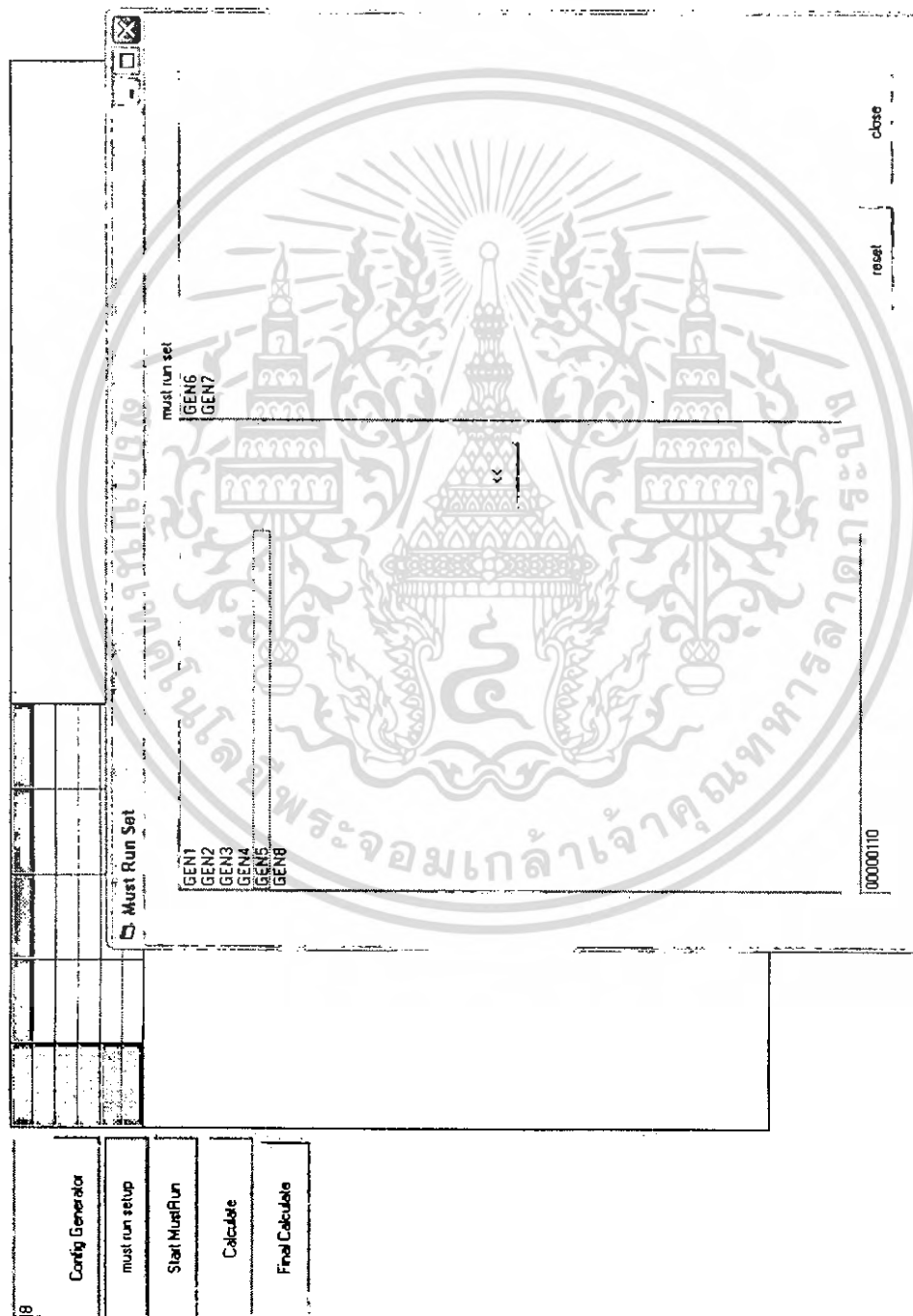
Gen Name	Gen No.	Load Limits			Fuel Cost Parameter			Start Up Cost	Start Down Cost	Loading Rate MW/min	D-Loading Rate MW/min	Min up time (Hrs.)	Min down time (Hrs.)	GT unit
		Min	Max	Mix	A	B	C							
a	GEN1	100	625	8	0.008	8	500	3000	1500	7.5	9	1	2	2
b	GEN2	100	625	64	0.0086	64	400	3000	1500	7.5	9	1	2	2
c	GEN3	75	600	79	0.01	79	600	3000	1500	15	7	1	2	2
d	GEN4	75	500	75	0.011	75	400	3000	1500	15	7	1	2	2

stage Load Curve (MW)

1	1100
2	1400
3	1600
4	1800
5	1400
6	1100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B Form 2

Close	Lambda	aI	bI	pI	p2	p3	p4	p5	p6
X1	1600X11	87357.9196	55.5233	-1479.3602	0	0	0	0	0
X2		53703.88885	30.9068	4733.00885	0	0	0	0	0
X3		0	0	0	0	0	0	0	0
X4		87379.8475	55.523	-1486.9525	0	0	0	0	0
X5		0	0	0	0	0	0	0	0
X6		34102.11526488	19.730469087	2533.36478768	0	0	0	0	0
X7		0	0	0	0	0	0	0	0
X8		30515.99	14.85	6755.99	0	0	0	0	0
X9		0	0	0	0	0	0	0	0
X10		0	0	0	0	0	0	0	0
X11		18495.919904194	8.472924679	4939.240529794	359.763320808	0	0	0	0
X12		0	0	0	0	0	0	0	0
X13		19369.61550246	9.674821659	3889.90064806	375.499577699	0	0	0	0
X14		0	0	0	0	0	0	0	0
X15		15854.201941768	7.351127372	4092.988146568	312.18537338	311.783485074	612.674204833	0	0
X16		30580.6	14.87	6788.6	0	0	0	0	0
X17		0	0	0	0	0	0	0	0
X18		0	0	0	0	0	0	0	0
X19		18523.591576999	8.479431787	4956.443717639	320.261219648	0	0	0	0
X20		0	0	0	0	0	0	0	0
X21		19401.913628477	9.633306804	3908.622742077	376.081291705	0	0	0	0
X22		0	0	0	0	0	0	0	0
X23		15876.395412114	7.356025044	4106.755341714	312.58508792	312.183201774	0	611.148657842	0
X24		0	0	0	0	0	0	0	0
X25		16283.31894882	6.553079086	5798.99239522	319.913956728	0	0	641.570972985	638.515050445
X26		15939.76548623	5.978642818	8373.93797743	5978642818	0	0	618.436127019	615.411330614
X27		14246.442148306	5.397455163	5510.513887506	283.22688496	310.927440252	0	504.407552074	501.536122953
X28		16285.95279136	6.553074917	5801.032411596	5901.032411596	0	0	641.748301625	638.692150581
X29		14410.562103356	5.661301421	5032.479828756	296.184760332	0	0	515.459400899	512.573107152
X30		14248.612185707	5.397452328	5612.668440907	310.899340099	0	0	504.553681192	501.682065528
X31		12895.146427979	4.919249244	4994.347675759	259.171013934	265.370361422	0	410.717604578	407.97218749
X32		34938.9908	18	6138.9908	0	0	0	0	0
X33		0	0	0	0	0	0	0	0
X34		23753.155236033	11.334269987	5618.929556893	0	0	0	0	0
X35		0	0	0	0	0	0	0	0
X36		26078.466968948	13.593215729	4279.341794148	0	0	0	0	0
X37		0	0	0	0	0	0	0	0
X38		0	0	0	0	0	0	0	0
X39		16441.250765034	8.048391523	3563.888188234	322.759391253	322.356560075	0	0	0
X40		15815.171656711	7.0465926851	5460.0611294111	329.459287667	0	0	577.285970082	0
X41		0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

compare = 26693times

stat low cost path

x11	134309706.886763	x15	144749156.0789
x13	131258885.173086	x23	144594959.638749
x15	115356245.557703	x25	154335535.002316
x19	134533682.41618	x27	140710115.580539
x21	131501090.224986	x29	136026317.813307
x23	115562737.185531	x30	140758224.287971
x25	124410240.78771	x31	126863235.505611
x26	129005791.714528	x43	141723352.252708
x27	114589148.501536	x45	137198145.071148

x7	91979552.9181625	x15	144749156.0789
x9	101529462.089175	x23	144594959.638749
x11	87265968.9378381	x25	154335535.002316
x13	82439871.8683581	x27	140710115.580539
x14	87312703.401555	x29	136026317.813307
x15	75121111.4582587	x30	140758224.287971
x17	101765944.488261	x31	126863235.505611
x19	87416275.8016555	x43	141723352.252708
x21	32598383.8739355	x45	137198145.071148



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้