การแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว

AN ENHANCED DIFFERENTIAL EVOLUTION ALGORITHM FOR ECONOMIC DISPATCH WITH MULTIPLE FUELS AND SYSTEM LOSSES CONSIDERATIONS

นายทิวัตถ์ เอี่ยวสำราญ
นายวิรัช วุ่นกลิ่นหอม

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
แขนงวิชาวิศวกรรมไฟฟ้ากำลัง ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2556

การแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว

AN ENHANCED DIFFERENTIAL EVOLUTION ALGORITHM FOR ECONOMIC DISPATCH WITH MULTIPLE FUELS AND SYSTEM LOSSES CONSIDERATIONS

นายทิวัตถ์ เอี่ยวสำราญ
นายวิรัช วุ่นกลิ่นหอม

โครงงานวิศวกรรมนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิศวกรรมศาสตรบัณฑิต
แขนงวิชาวิศวกรรมไฟฟ้ากำลัง ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2556

หัวข้อโครงงานวิศวกรรมสาขาวิศวกรรมไฟฟ้า

เรื่อง การแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว

โดย นายทิวัตถ์ เอี่ยวสำราญ

นายวิรัช วุ่นกลิ่นหอม

ภาควิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษา ดร.ธนาธิป สุ่มอิ่ม

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ อนุมัติให้นับโครงงานวิศวกรรมไฟฟ้าเป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.เวคิน ปียรัตน์)

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ปฐมทัศน์ จิระเดชะ)

..... กรรมการ

(อาจารย์ ดร.คมกฤษ ประเสริฐวงษ์)

..... กรรมการ

(อาจารย์ ดร.ธนาธิป สุ่มอิ่ม)

การแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว

ปีการศึกษา 2556

โดย

นายทิวัตถ์ เอี่ยวสำราญ

นายวิรัช วุ่นกลิ่นหอม

อาจารย์ที่ปรึกษา

ดร.ธนาธิป สุ่มอิ่ม

บทคัดย่อ

โครงงานวิศวกรรมนี้ได้นำเอาโปรแกรมคอมพิวเตอร์มาประยุกต์ใช้ในการคำนวณการแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว (An Enhanced Differential Evolution Algorithm for Economic Dispatch with Multiple Fuels and System Losses Considerations) โดยจะทำการศึกษาเป็นการวางแผนแบบสถิตยศาสตร์ (Static Planning) ทั้งหมด 2 กรณี คือกรณีการศึกษาในระบบที่เครื่องกำเนิดไฟฟ้าสามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิดโดยไม่พิจารณาผลจากจุดวาล์ว และกรณีการศึกษาในระบบที่เครื่องกำเนิดไฟฟ้าสามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิด โดยพิจารณาผลจากจุดวาล์ว ซึ่งทั้ง 2 กรณีนั้นจะคิดการสูญเสียที่เกิดขึ้นในระบบสายส่งเข้าไปด้วย โดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้วมาแก้ปัญหา จากการทดสอบพบว่าวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้วที่นำไปประยุกต์ใช้นั้นมีประสิทธิภาพในการทำงานเป็นอย่างดี สามารถหาคำตอบได้เหมาะสมสำหรับการจ่ายโหลดอย่างประหยัด จะเห็นได้ว่าวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้วมีประสิทธิภาพน่าเชื่อถือและสามารถให้คำตอบได้อย่างเหมาะสม

คำสำคัญ: การจ่ายโหลดอย่างประหยัด/ วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม/ การวางแผนแบบสถิตยศาสตร์/ เครื่องกำเนิดไฟฟ้า

AN ENHANCED DIFFERENTIAL EVOLUTION ALGORITHM FOR ECONOMIC DISPATCH WITH MULTIPLE FUELS AND SYSTEM LOSSES CONSIDERATIONS

Academic Year 2013

By

Mr.Tiwat Eaiosamran

Mr.Virat Voonklinhom

Advisor

Dr.Thanathip Sum-Im

ABSTRACT

In this engineering project, a computer programming is applied to solve economic dispatch (ED) with multiple fuels and system losses considerations. Two cases of static planning are investigated that are generators using multiple fuels with and without valve-point effects. In this project, an enhanced differential evolution algorithm (DEA) is employed for solving the ED problem considering transmission losses. The proposed method is efficient and effective to optimize the ED problem as shown in the achieved results. Regarding the obtained results, an enhanced DEA method has proved to be an effective, stable and well-result.

Keyword: Economic dispatch/ Differential evolution algorithm/ Static planning/ Generator

กิตติกรรมประกาศ

คณะผู้จัดทำขอกราบขอบพระคุณ ดร.ธนาธิป สุ่มอิ่ม อาจารย์ที่ปรึกษาโครงงาน ที่คอยให้ความช่วยเหลือ คำปรึกษาในการค้นค้างานวิจัย และคอยชี้แนะแนวทางต่างๆในการทำโครงงาน ตลอดจนติดตามความก้าวหน้าและตรวจแก้ไขข้อบกพร่องต่างๆในโครงงาน จนทำให้โครงงานนี้ประสบความสำเร็จได้ตรงตามเป้าหมาย และขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร.ปฐมทัศน์ จิระเดชะ ประธานกรรมการสอบโครงงานวิศวกรรม และ ดร.คมกฤษ ประเสริฐวงษ์ กรรมการสอบโครงงานวิศวกรรม

ขอกราบขอบพระคุณอาจารย์ภาควิชาวิศวกรรมไฟฟ้าทุกท่านที่ได้ให้คำปรึกษา ให้ข้อคิดอบรมสั่งสอน และประสิทธิ์ประสาทวิชาความรู้อันเป็นประโยชน์อย่างยิ่งเพื่อนำมาปรับแก้ในโครงงานนี้ให้มีความสมบูรณ์จนสำเร็จตามเป้าหมาย

ท้ายที่สุดนี้ขอกราบขอบพระคุณบิดา มารดาและผู้มีพระคุณทุกท่านเป็นอย่างสูงที่คอยอบรมเลี้ยงดูเอาใจใส่ คอยให้กำลังใจ และให้ความสนับสนุนคณะผู้จัดทำตลอดระยะเวลาที่ได้ศึกษาเล่าเรียน อีกทั้งยังเป็นแรงผลักดันให้คณะผู้จัดทำประสบความสำเร็จจนสามารถทำโครงงานเล่มนี้ได้สำเร็จลุล่วงด้วยดี

สารบัญ

	หน้าที่
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญภาพ	ฌ
คำอธิบายสัญลักษณ์และคำย่อ	ฎ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	7
1.3 ขอบเขตของโครงงาน	7
1.4 ประโยชน์ที่คาดว่าจะได้รับ	8
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	9
2.1 ปัญหาการจัดสรรกำลังผลิต	9
2.2.1 ต้นทุนดำเนินงานของหน่วยผลิตกำลังไฟฟ้า	9
2.1.2 ต้นทุนส่วนเชื้อเพลิงส่วนเพิ่ม	10
2.1.3. สมการต้นทุนการดำเนินงานของหน่วยผลิต	12
2.1.4 ฟังก์ชันราคาเชื้อเพลิง	15
2.1.5 ผลกระทบที่เกิดจากจุดวาล์ว	16
2.2 วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม	17
2.2.1 การสมมติค่าเริ่มต้น (Initialization)	18
2.2.2 การมิวเตชั่น (Mutation)	18
2.2.3 การครอสโอเวอร์ (Crossover)	21
2.2.4 การคัดเลือก (Selection)	22
2.2.5 การทำงานของ DEA	22
2.3 ทบทวนวรรณกรรมที่เกี่ยวข้อง	24

สารบัญ (ต่อ)

	หน้าที่
บทที่ 3 ขั้นตอนและวิธีการดำเนินงาน	26
3.1 แผนผังขั้นตอนการทำงาน	27
3.2 ขั้นตอนการดำเนินงาน	28
3.3 แสดงขั้นตอนและวิธีการดำเนินงาน	28
3.4 หลักการคิดโปรแกรม	29
3.5 ขั้นตอนการทำงานของโปรแกรม	30
3.6 ตารางการดำเนินงาน	31
บทที่ 4 ผลการทดลอง	32
4.1 รูปแบบของเมนูหลักของโปรแกรมการจัดสรรกำลังผลิตด้วยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว	33
4.2 ผลการทดลอง	36
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ	52
5.1 สรุปผลการทดลอง	52
5.2 ข้อเสนอแนะ	52
เอกสารอ้างอิง	53
ภาคผนวก	55
ภาคผนวก ก ตารางข้อมูลของระบบทดสอบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง	56
ภาคผนวก ข โปรแกรมที่ใช้ในการทดลอง	61
ประวัติผู้จัดทำโครงงาน	104

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงค่าความร้อนจำเพาะของเชื้อเพลิงชนิดต่างๆ 4

3.1 แสดงการดำเนินงาน 31

4.1 ผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง 37

4.2 สรุปผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง 38

4.3 ผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง 45

4.4 สรุปผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง 46

ภาคผนวก ก

ก1 ตารางข้อมูลของเครื่องกำเนิดไฟฟ้าที่ใช้ในการทดสอบในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ซึ่งมีฟังก์ชั่นราคาเชื้อเพลิงที่มีผลจากเครื่องกำเนิดไฟฟ้าสามารถเลือกใช้เชื้องเพลิงได้หลายชนิด 56

ก2 ตารางข้อมูลของเครื่องกำเนิดไฟฟ้าที่ใช้ในการทดสอบในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ซึ่งมีฟังก์ชั่นราคาเชื้อเพลิงที่มีผลจากเครื่องกำเนิดไฟฟ้าสามารถเลือกใช้เชื้องเพลิงได้หลายชนิดและมีผลจากจุดวาล์ว 58

สารบัญรูป

รูปที่		หน้า
1.1	สัดส่วนการใช้เชื้อเพลิงในการผลิตไฟฟ้าสำหรับปี 2555	2
1.2	สัดส่วนการผลิตพลังงานไฟฟ้าแยกตามประเภทเชื้อเพลิงตามแผนกำลังผลิตไฟฟ้า	3
1.3	แสดงค่าประมาณความต้องใช้พลังงานไฟฟ้าสูงสุดในปี 2556	6
2.1	ความสัมพันธ์ระหว่างอัตราพลังงานความร้อนและกำลังผลิต	10
2.2	กราฟแสดงความสัมพันธ์ระหว่างต้นทุนส่วนเพิ่มของเชื้อเพลิงกับกำลังขาออก	11
2.3	แบบจำลองของโรงไฟฟ้าพลังงานความร้อน(Thermal Plant)	12
2.4	แผนผังแสดงการทำงาน DEA	23
3.1	แผนผังแสดงขั้นตอนการทำงาน	27
3.2	แสดงแผนผังขั้นตอนการทำงานของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้	29
4.1	หน้าจอขณะทำการรันโปรแกรม	33
4.2	หน้าจอที่ 2 ขณะทำการรันโปรแกรม	34
4.3	หน้าจอโปรแกรมหลังจากทำการป้อนค่าต่างๆแล้วทำการกดปุ่ม RUN	35
4.4	หน้าจอแสดงผลการทดลองเป็นกราฟระหว่าง Cost กับ Iteration	36
4.5	แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,400 และ2,500 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง	42
4.6	แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,600 และ2,700 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง	43
4.7	แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,400 และ2,500 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง	50

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.8 แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,600 และ2,700 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง 51

คำอธิบายสัญลักษณ์และคำย่อ

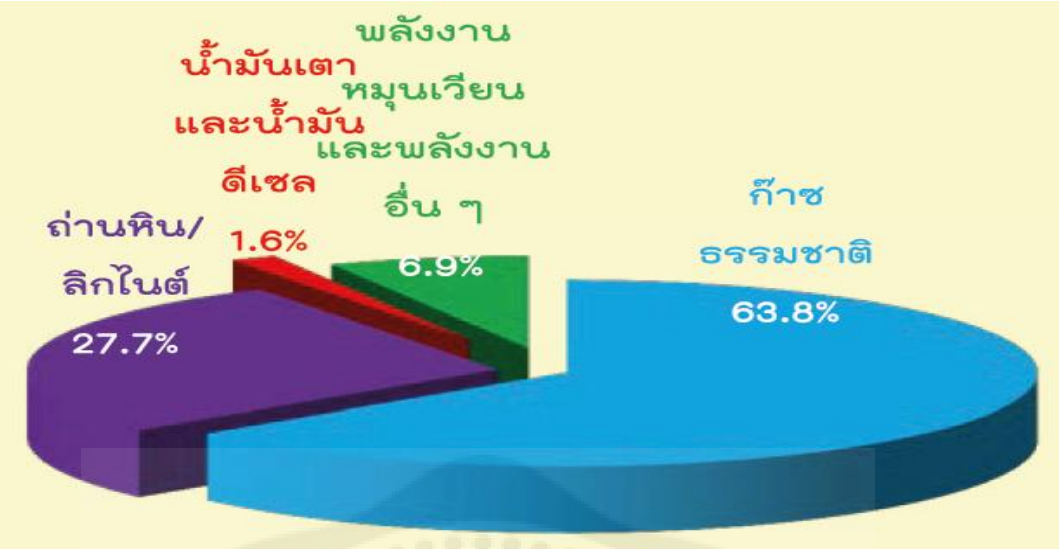
DE	ดิฟเฟอร์เรนเชียลอิโวลูชั่น (Differential Evolution)	
DEA	ดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม (Differential Evolution Algorithm)	
SaDEA	วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ (Self-Adaptive Differential Evolution Algorithm)	
ED	การจัดสรรกำลังผลิต (Economic Dispatch)	
IGA	วิธีเชิงพันธุกรรมที่ได้รับการปรับปรุง (Improved Genetic Algorithm)	
PSO	วิธีการทำให้เหมาะสมแบบกลุ่มอนุภาค (Particle Swarm Optimization)	
SQP	วิธีโปรแกรมกำลังสองแบบลำดับ (Sequential Quadratic Programming)	
H_i	อัตราความร้อนขาเข้าต่อหน่วยเมกะวัตต์ที่หน่วยผลิต	MBtu/hr
P_i	กำลังผลิตของเครื่องกำเนิดไฟฟ้าหน่วยที่ i	MW
a_i,b_i,c_i	สัมประสิทธิ์คงตัวทางความร้อนของเครื่องกำเนิดไฟฟ้าหน่วยที่ i	
F_T	ราคาเชื้อเพลิงรวม	Baht/hr
F_i	ราคาเชื้อเพลิงเครื่องกำเนิดไฟฟ้าที่ i	Baht/hr
P_D	กำลังไฟฟ้าที่โหลดต้องการใช้	MW
P_L	กำลังไฟฟ้าสูญเสียที่เกิดขึ้นในสายส่ง	MW
P_i^{min}	กำลังขาออกที่น้อยที่สุดของเครื่องกำเนิดไฟฟ้าที่ i	MW
P_i^{max}	กำลังขาออกที่มากที่สุดของหน่วยผลิต i	MW
P_j	กำลังไฟฟ้าขาออกของหน่วยผลิตที่ j	MW
C_T	ราคาเชื้อเพลิงรวม	Baht/hr
C_i	กำลังขาออกของเครื่องกำเนิดไฟฟ้า	MW
B_{ij}	สัมประสิทธิ์การสูญเสียของเมตริกซ์สี่เหลี่ยมที่ตำแหน่ง ij	
B_{i0}	สัมประสิทธิ์การสูญเสียของเวกเตอร์ที่ตำแหน่ง i	
B_{00}	สัมประสิทธิ์การสูญเสียของเวกเตอร์	

a_i, b_i, c_i	สัมประสิทธิ์ของฟังก์ชันราคาค่าเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าที่ i
e_i, f_i	สัมประสิทธิ์ของฟังก์ชันราคาเชื้องเพลิงที่เกิดจากจุดวาล์วของเครื่องกำเนิดไฟฟ้าที่ i
n	จำนวนเครื่องกำเนิดไฟฟ้าที่ทำงาน
$T1$	ตัวบ่งชี้การเปลี่ยนแปลงค่าพารามิเตอร์ F
$T2$	ตัวบ่งชี้การเปลี่ยนแปลงค่าพารามิเตอร์ CR
$x_{j,i}^{(G=0)}$	ค่าเริ่มต้นของพารามิเตอร์ j^{th} ของเวกเตอร์อิสระ i^{th}
x_j^{min}, x_j^{max}	ค่าต่ำสุดและค่าสูงสุดของพารามิเตอร์ j^{th} ตามลำดับ
$X_{i,G}$	เวกเตอร์เป้าหมาย (Target Vector)
$V_{i,G+1}$	มิวแตนท์เวกเตอร์ (Mutant Vector)
$X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$	Random Vector
$rand_j$	ค่าที่สุ่มขึ้นมามีค่า (0,1)
$u_{j,i}^{(G)}$	เวกเตอร์ทดลอง (Trial Vector)
CR	Crossover Constant มีค่าเป็นเลขจำนวนจริงระหว่าง 0 ถึง 1
N_P	จำนวนประชากร
VTR	Value to reach ค่าคงที่สำหรับหยุดการทำงานของโปรแกรม
S	ค่า Index จากการสุ่มเลือกมีค่าเป็นเลขจำนวนเต็มระหว่าง 1, 2, … , D j=1,2, … ,D

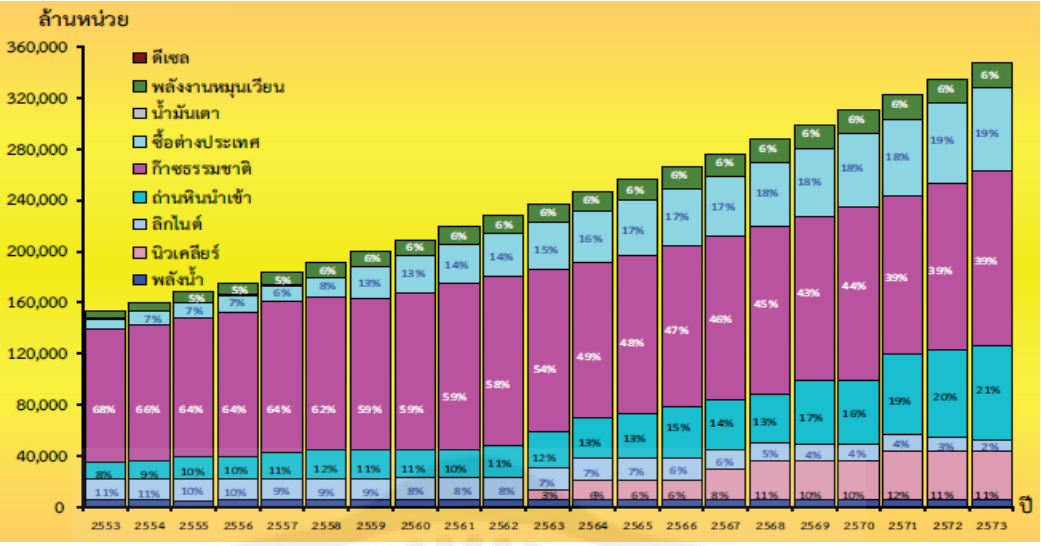
บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

ปัจจุบันความต้องการในการใช้ไฟฟ้ามีมากขึ้นตามการเจริญเติบโตทางด้านสังคมและเศรษฐกิจ ระบบไฟฟ้ากำลังโดยทั่วไปประกอบด้วยหน่วยผลิตซึ่งได้แก่เครื่องกำเนิดไฟฟ้า ระบบส่งจ่าย ระบบจำหน่าย และหน่วยบริโภคทางไฟฟ้า เนื่องจากปัจจุบันไม่มีอุปกรณ์ที่สามารถกักเก็บพลังงานไฟฟ้าในปริมาณที่มากๆได้ โดยปริมาณของกำลังไฟฟ้าที่ผลิตได้นั้นควรมีค่าเท่ากับปริมาณความต้องการของโหลด ซึ่งกำลังผลิตจะมีการปรับเปลี่ยนตามปริมาณการใช้งานของผู้ใช้ไฟฟ้าอยู่ตลอดเวลา การผลิตกำลังไฟฟ้าจำเป็นจะต้องใช้ต้นทุนในการผลิตซึ่งต้นทุนนี้ขึ้นอยู่กับปริมาณการผลิตหรือกำลังผลิตของโรงไฟฟ้าโดยจะแปรผันตามปริมาณเชื้อเพลิงที่ใช้ในการผลิต กำลังไฟฟ้าที่ผลิตได้ในแต่ละโรงไฟฟ้านั้นจะไม่เท่ากันเนื่องจากอัตราค่าเชื้อเพลิงแต่ละชนิดมีค่าไม่เท่ากัน และอัตราการสูญเสียที่เกิดขึ้นในสายส่งซึ่งขึ้นอยู่กับระยะทางระหว่างเครื่องกำเนิดไฟฟ้ากับโหลด ดังนั้นจึงจำเป็นต้องมีการจัดสรรกำลังผลิตไฟฟ้าของหน่วยผลิต (Economic Dispatch of Generating Units) เพื่อให้สอดคล้องต่อปริมาณความต้องการการใช้ไฟฟ้าในปัจจุบัน มิใช่แค่ผู้ผลิตจะคิดเพียงว่าต้องผลิตพลังงานไฟฟ้าออกมาเท่านั้นแต่ยังต้องคำนึงถึงต้นทุนในการผลิต เพื่อที่จะจัดสรรกำลังผลิตไฟฟ้าได้อย่างประหยัดและให้มีประสิทธิภาพสูงที่สุดอีกด้วย



รูปที่ 1.1 สัดส่วนการใช้เชื้อเพลิงในการผลิตไฟฟ้าสำหรับปี 2555
ที่มา : สถิติพลังงานของประเทศไทยปี 2555 ศูนย์สารสนเทศข้อมูลพลังงานทดแทน และอนุรักษ์พลังงาน กรมพัฒนาพลังงานทดแทนและอนุรักษ์ พลังงาน



รูปที่ 1.2 สัดส่วนการผลิตพลังงานไฟฟ้าแยกตามประเภทเชื้อเพลิงตามแผนกำลังผลิตไฟฟ้า
ที่มา : การใช้ไฟฟ้าและการผลิตไฟฟ้าของประเทศไทย สำนักงานนโยบายและแผนพลังงาน กระทรวงพลังงาน

ตั้งแต่อดีตจนถึงปัจจุบันพลังงานเป็นปัจจัยพื้นฐานอย่างหนึ่งที่สำคัญสำหรับการดำรงชีวิตและการพัฒนาทางด้านต่างๆของมนุษย์ พลังงานส่วนใหญ่จะมาจากเชื้อเพลิงฟอสซิล เช่น ถ่านหิน น้ำมัน ก๊าซธรรมชาติ เป็นต้น จากรูปที่ 1.1 และ 1.2 สังเกตุเห็นว่าเชื้อเพลิงฟอสซิล (ถ่านหิน น้ำมัน ก๊าซธรรมชาติ) นั้นมีความจำเป็นมากสำหรับการผลิตไฟฟ้าโดยมีสัดส่วนโดยรวมอยู่ที่ 93.1% ของพลังงานเชื้อเพลิงทั้งหมดและยังคงมีความต้องการใช้เชื้อเพลิงประเภทนี้เพิ่มขึ้นเรื่อยๆ ดังรูปที่ 1.2 พลังงานเชื้อเพลิงเหล่านี้ล้วนแล้วแต่เป็นพลังงานมีปริมาณที่จำกัดและอาจจะหมดลงได้ในอนาคต ดังนั้นการที่จะนำมาใช้งานในแต่ละครั้งจึงจำเป็นที่จะต้องใช้อย่างรู้คุณค่าและใช้ให้เกิดประโยชน์สูงที่สุด โดยเราสามารถที่จะวัดปริมาณพลังงานออกมาเป็นหน่วยมาตรฐานต่างๆ เช่น แคลอรี่ (Calorie), จูล (Joule), กิโลวัตต์ชั่วโมง (kWh), บีทียู (BTU) หรือ TOE (Ton of Oil Equivalent) เป็นต้นตัวอย่างของค่าความสัมพันธ์ระหว่างหน่วยต่างๆ มีดังนี้

1 กิโลแคลอรี = 3.969 บีทียู = 4186 จูล

1 TOE = 4×10^{7} บีทียู

1 kWh = 3.6 ล้านจูล (MJ)

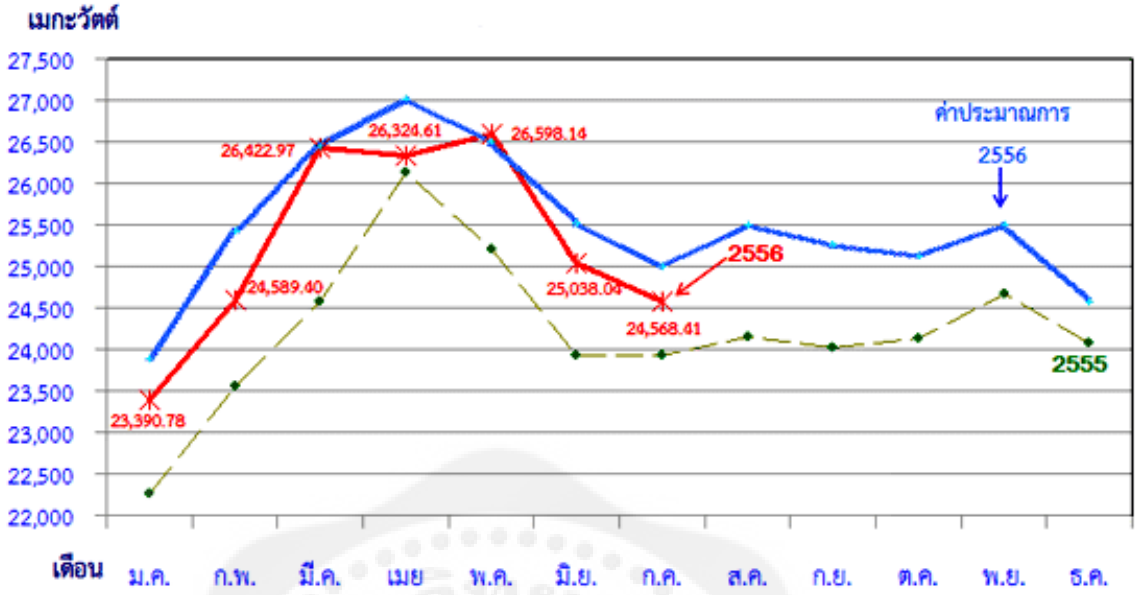
ตารางที่ 1.1 แสดงค่าความร้อนจำเพาะของเชื้อเพลิงชนิดต่างๆ

ชนิด	หน่วย	MJ/หน่วย	kCal/หน่วย
พลังงานเชิงพาณิชย์			
1. น้ำมันดิบ	ลิตร	36.33	8,680
2. คอนเดนเสท	ลิตร	33.07	7,900
3. ก๊าซธรรมชาติ			
3.1 ชื้น	ลบ.ฟุต	1.04	248
3.2 แห้ง	ลบ.ฟุต	1.02	244
4. ผลิตภัณฑ์ปิโตรเลียม			
4.1 ก๊าซ LPG	ลิตร	26.62	6,360
4.2 น้ำมันเบนซิน	ลิตร	31.48	7,520
4.3 น้ำมันเครื่องบิน	ลิตร	34.53	8,250
4.4 น้ำมันก๊าด	ลิตร	34.53	8,250
4.5 น้ำมันดีเซล	ลิตร	36.42	8,700
4.6 น้ำมันเตา	ลิตร	39.77	9,500
4.7 ยางมะตอย	ลิตร	41.19	9,840
4.8 ปิโตรเลียม	กก.	35.16	8,400
5. ไฟฟ้า	กิโลวัตต์ชั่วโมง	3.60	860
6. ไฟฟ้าพลังน้ำ	กิโลวัตต์ชั่วโมง	9.36	2,236
7. พลังงานความร้อนใต้พิภพ	กิโลวัตต์ชั่วโมง	39.77	9,500
8. ถ่านหินนำเข้า	กก.	26.37	6,300
9. ถ่านโค๊ก	กก.	27.63	6,600
10. แอนทราไซต์	กก.	31.40	7,500
11. อีเทน	กก.	46.89	11,203
12. โปรเพน	กก.	47.11	11,256
13. ลิกไนท์			
13.1 ลี้	กก.	18.42	4,400
13.2 กระบี่	กก.	10.88	2,600

13.3 แม่เมาะ	กก.	10.47	2,500
13.4 แจ้คอน	กก.	15.11	3,610
พลังงานหมุนเวียน			
1. ฟืน	กก.	15.99	3,820
2. ถ่านไม้	กก.	28.88	6,900
3. แกลบ	กก.	14.40	3,440
4. กากอ้อย	กก.	7.53	1,800
5. ขยะ	กก.	4.86	1,160
6. ขี้เลื่อย	กก.	10.88	2,600
7. วัสดุที่เหลือใช้ทางการเกษตร	กก.	12.68	3,030
8. ก๊าซชีวภาพ	ลบ.ม.	20.93	5,000

ที่มา : สถานการณ์พลังงานของประเทศไทย ไตรมาสที่ 1/2556 กรมพัฒนาพลังงานทดแทนและอนุรักษ์พลังงาน กระทรวงพลังงาน

โดยปกติผู้ใช้ไฟฟ้าแต่ละประเภทจะมีลักษณะและพฤติกรรมการใช้ไฟฟ้าแตกต่างกัน ส่งผลต่อเนื่องให้เกิดลักษณะการใช้ไฟฟ้าที่แตกต่างกัน เช่น ผู้ใช้ไฟฟ้าประเภทที่อาศัยอยู่ที่บ้านอาจจะใช้ไฟฟ้าสูงในช่วงเย็นถึงหัวค่ำเพื่อประกอบกิจกรรมต่างๆ มีการใช้ไฟฟ้าต่ำในช่วงกลางคืนและกลางวันซึ่งไม่มีคนพักอาศัย ส่วนผู้ใช้ไฟฟ้าประเภทโรงงานอุตสาหกรรมหรืออาคารธุรกิจ อาจมีการใช้ไฟฟ้าสูงและค่อนข้างสม่ำเสมอในช่วงกลางวัน ส่วนช่วงหัวค่ำจนถึงเช้าตรู่อาจมีการใช้ไฟฟ้าต่ำ เป็นต้น ด้วยลักษณะและพฤติกรรมการใช้ไฟฟ้าที่แตกต่างกันแบบนี้ก่อให้เกิดความต้องการกำลังไฟฟ้ารวมของทั้งประเทศ หรือของแต่ละพื้นที่แปรเปลี่ยนกันไปตามเวลา ซึ่งความต้องการกำลังไฟฟ้า (Demand) ของระบบ หมายถึงปริมาณโหลดเฉลี่ยในแต่ละช่วงเวลาตามที่กำหนด โดยช่วงเวลาดังกล่าวนี้อาจเรียกว่า ช่วงเวลาของความต้องการใช้ไฟฟ้า (Demand Interval) ซึ่งปกติมักกำหนดเป็นระยะเวลา 15 นาที 30 นาที หรือ 1 ชั่วโมง เป็นต้น สำหรับช่วงเวลาโหลดที่ใช้ในการคิดค่าไฟฟ้าปัจจุบันของประเทศไทยนั้นมีค่าอยู่ที่ 15 นาที ส่วนโหลดนั้นมีความหมายถึงปริมาณการใช้ไฟฟ้า ณ เวลาใดๆ มีหน่วยเป็นกิโลวัตต์ หรือเมกะวัตต์



รูปที่ 1.3 แสดงค่าประมาณความต้องใช้พลังงานไฟฟ้าสูงสุดในปี 2556
ที่มา : กองสารสนเทศ ฝ่ายสื่อสารองค์การ กฟผ.

จากรูปที่ 1.3 แสดงให้เห็นว่าความต้องการใช้พลังงานไฟฟ้าสูงสุดในเดือนกรกฎาคมซึ่งเกิดขึ้นเมื่อวันที่ 3 กรกฎาคม 2556 เวลา 14.30 น. มีค่าเท่ากับ 24,568.41 เมกะวัตต์ ลดลงจากเดือนที่ผ่านมา 469.63 เมกะวัตต์ หรือคิดเป็นร้อยละ 1.88 และความต้องการใช้พลังงานไฟฟ้าสูงสุดของระบบปี 2556 จะอยู่ที่ 26,598.14 เมกะวัตต์ เห็นได้ว่าปริมาณความต้องการใช้ไฟฟ้าของผู้บริโภคในปี 2556นั้นมีปริมาณมากกว่าปี 2555 และยังคงมีปริมาณความต้องการใช้ไฟฟ้าเพิ่มขึ้นทุกๆปี ดังนั้นจึงควรต้องมีการจัดสรรกำลังการผลิต (Economic Dispatch) ให้เหมาะสมต่อปริมาณความต้องการใช้ไฟฟ้าที่เพิ่มขึ้นของผู้บริโภคในช่วงเวลาต่างๆ โดยต้องคำนึงถึงราคาต้นทุนที่ใช้ในการผลิตและการเลือกใช้เชื้อเพลิงของแต่ละโรงไฟฟ้าโดยต้องมีราคาต้นทุนรวมต่ำที่สุด ในที่นี้ได้เสนอวิธีการแก้ปัญหาการจ่ายโหลดอย่างประหยัดสำหรับหน่วยการผลิตไฟฟ้าที่สามารถพิจารณางานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และคิดค่าความสูญเสียที่เกิดขึ้นในระบบด้วยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ซึ่งเป็นวิธีการแก้ปัญหาที่สามารถประยุกต์ใช้ในการแก้ปัญหาต่างๆ ได้อย่างมีประสิทธิภาพ โดยจะทำการสมมุติค่าเริ่มต้น หาค่าตัวประกอบมิวเตชั่น (F) และค่าคงที่การครอสโอเวอร์ (CR) ที่เหมาะสม จากนั้นทำการคัดเลือกค่าที่ดีที่สุด เพื่อที่จะทำให้การ

แก้ปัญหานั้นมีประสิทธิภาพมากยิ่งขึ้น โดยโปรแกรมที่สร้างขึ้นมานั้นง่าย สะดวกรวดเร็วต่อการใช้งาน และมีประสิทธิภาพ

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการจัดสรรกำลังผลิตในการหาต้นทุนการผลิตและการส่งจ่ายกำลังไฟฟ้าให้ได้ค่าต้นทุนที่เหมาะสมที่สุด

1.2.2 เพื่อศึกษาวิธีการทำงานของดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับค่าได้

1.2.3 เพื่อพัฒนาโปรแกรมในการคำนวณการจ่ายโหลดอย่างประหยัดสำหรับหน่วยการผลิตไฟฟ้าโดยพิจารณาเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และค่าความสูญเสียที่เกิดขึ้นในสายส่งโดยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับค่าได้

1.2.4 เพื่อศึกษาการใช้โปรแกรม MATLAB ในการจำลองการคำนวณของดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับค่าได้

1.2.5 เพื่อที่จะสามารถนำดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับค่าได้ไปประยุกต์ใช้ในการแก้ไขปัญหาการจัดสรรกำลังผลิตไฟฟ้าให้ได้ต้นทุนที่เหมาะสมที่สุด

1.3 ขอบเขตของโครงงาน

1.3.1 พัฒนาวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบดั้งเดิมให้สามารถปรับค่าพารามิเตอร์ควบคุม (Control Parameter) ได้ด้วยตัวเองคือ ตัวประกอบมิวเตชั่น (F) และค่าคงที่การครอสโอเวอร์ (CR) เราเรียกวิธีการแบบนี้ว่า ดิฟฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ (Self-Adaptive Differential Evolution Algorithm)

1.3.2 วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ที่พัฒนาขึ้นสามารถนำมาประยุกต์ใช้แก้ปัญหาการจัดสรรกำลังผลิตเพื่อการจ่ายโหลดอย่างประหยัดสำหรับหน่วยการผลิตไฟฟ้าแบบเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลกระทบจากจุดวาล์ว และพิจารณาค่ากำลังความสูญเสียในสายส่ง (Economic Dispatch with Multiple Fuels, Valve-Point Effects and Power Loss in Transmission Line Considerations) โดยแก้ปัญหาจะพิจารณาเฉพาะการจัดสรรกำลังผลิตของหน่วยผลิตแบบสถิตย์ศาสตร์ (Static Economic Dispatch of Generating Units) เท่านั้น

1.3.3 สามารถจัดสรรการผลิตกำลังไฟฟ้าของโรงไฟฟ้าได้ตั้งแต่ 2-10 โรงไฟฟ้า และเป็นโรงไฟฟ้าชนิดพลังงานความร้อนเท่านั้น (Thermal Power Plant)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ได้รับความรู้เกี่ยวกับการจัดสรรกำลังผลิตของหน่วยการผลิตไฟฟ้าและวิธีการทำงานของดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม (Differential Evolution Algorithm)

1.4.2 สามารถนำความรู้ที่ได้จากการเขียนโปรแกรม MATLAB ไปประยุกต์ใช้ในการจำลองการคำนวณต่างๆได้

1.4.3 สามารถนำวิธีการของดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ที่ออกแบบมาใช้ในการแก้ปัญหาการจัดสรรกำลังผลิตของหน่วยผลิตไฟฟ้า

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

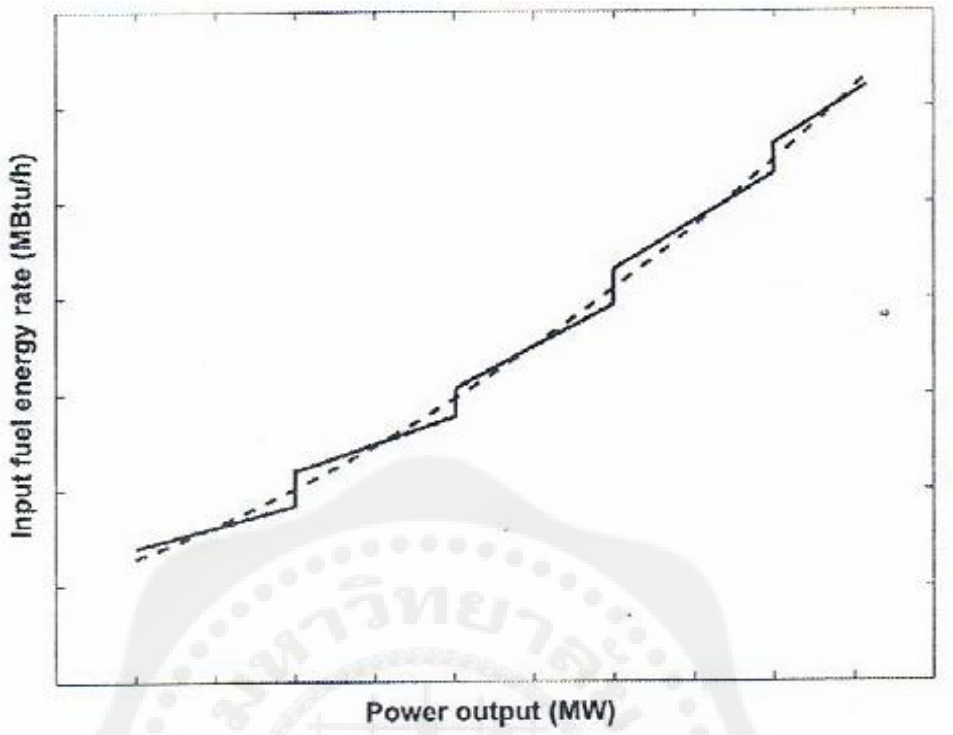
2.1 ปัญหาการจัดสรรกำลังผลิต (Economic Dispatch)

การจัดสรรกำลังผลิต (ED) คือการคำนวณการวางแผนการผลิตกำลังไฟฟ้าเพื่อการผลิตกำลังไฟฟ้าให้เพียงพอกับความต้องการของโหลดที่เปลี่ยนแปลงในแต่ละช่วงเวลาโดยที่ต้นทุนดำเนินงานของหน่วยผลิตกำลังไฟฟ้ารวมมีค่าน้อยที่สุดและเหมาะสมกับเครื่องกำเนิดไฟฟ้า (Generator) แต่ละเครื่อง

2.1.1 ต้นทุนดำเนินงานของหน่วยผลิตกำลังไฟฟ้า (Operating Cost of Generating Unit)

ต้นทุนดำเนินงานคือค่าใช้จ่ายทั้งหมดที่ต้องใช้ในการผลิตกำลังไฟฟ้าสามารถแบ่งออกได้เป็น 2 ส่วนคือ ต้นทุนคงที่ (Fixed Cost) และต้นทุนแปรผัน (Variable Cost) ซึ่งในที่นี้ ต้นทุนคงที่หมายถึงต้นทุนรวมที่ไม่เปลี่ยนแปลงไปตามระดับของการผลิตในช่วงของการผลิตระดับหนึ่งแต่ต้นทุนคงที่ต่อหน่วยจะเปลี่ยนแปลงไปในทางลดลงถ้าหากปริมาณการผลิตมีเพิ่มมากขึ้น โดยมีอยู่ 2 ลักษณะ คือ ต้นทุนคงที่ระยะยาว (Committed Fixed Cost) และต้นทุนคงที่ระยะสั้น (Discretionary Fixed Cost) ตัวอย่างต้นทุนคงที่นั้นได้แก่ เงินทุนเพื่อก่อสร้างหน่วยผลิต ค่าจ้างแรงงาน ค่าดอกเบี้ยจากการกู้เงินมาลงทุน ส่วนต้นทุนแปรผันหมายถึงต้นทุนที่มีการเปลี่ยนแปลงอย่างเป็นสัดส่วนกับราคาความแตกต่างของเชื้อเพลิง ได้แก่ ราคาเชื้อเพลิง น้ำมัน แก๊ส ถ่านหิน เป็นต้น ซึ่งเป็นสัดส่วนโดยตรงกับปริมาณกำลังไฟฟ้า (P_g) ที่มีหน่วยเป็นเมกะวัตต์ (MW) โดยทั่วไปเมื่อพิจารณาหน่วยผลิตพลังงานไฟฟ้า ปริมาณกำลังไฟฟ้าจริงที่สามารถผลิตได้คือ ผลผลิต (Output) ซึ่งส่งออกจากหน่วยผลิตไปยังโหลดและปริมาณเชื้อเพลิงที่ใช้ในการผลิตต่อชั่วโมงที่ส่งเข้าหน่วยผลิต (Input) ซึ่งมีความสัมพันธ์กับอัตราเชื้อเพลิงขาเข้า (Input Fuel Energy Rate: F_i) มีหน่วยผลิตเป็นเมกะบีทียูต่อชั่วโมง (MBtu/hr) และต้นทุนของเชื้อเพลิง (C_i) ที่ใช้ต่อชั่วโมงมีหน่วยเป็นบาทต่อชั่วโมง (Baht/hr)

จากความสัมพันธ์ระหว่างคุณลักษณะขาเข้าและขาออกสามารถถูกแสดงออกมาได้ด้วยกราฟความสัมพันธ์ระหว่างพลังงานความร้อนอินพุทที่ป้อนให้แก่หม้อกำเนิดไอน้ำกับแกนนอนที่แสดงถึงกำลังไฟฟ้าเอาท์พุทสุทธิที่ได้จากเครื่องกำเนิดไฟฟ้าดังรูปที่ 2.1



รูปที่ 2.1 ความสัมพันธ์ระหว่างอัตราพลังงานความร้อนและกำลังผลิต

ที่มา : พิชัย อารีย์. 2552

จากรูปที่ 2.1 จะเห็นได้ว่ากำลังผลิตที่ผลิตได้นั้นจะขึ้นอยู่กับอัตราพลังงานความร้อนขาเข้าต่อชั่วโมง โดยความสัมพันธ์ระหว่างปริมาณทั้งสองนี้จะมีลักษณะที่ต่อเนื่องเป็นช่วงซึ่งหาได้จากการทดลอง ในการคำนวณหาค่าเชื้อเพลิงของหน่วยผลิตแต่ละหน่วยนั้นจะใช้ความสัมพันธ์ระหว่างกำลังผลิตและอัตราพลังงานขาเข้าต่อชั่วโมง ในที่นี้จะใช้ค่าประมาณซึ่งมีลักษณะเป็นโค้งดังรูป 2.1

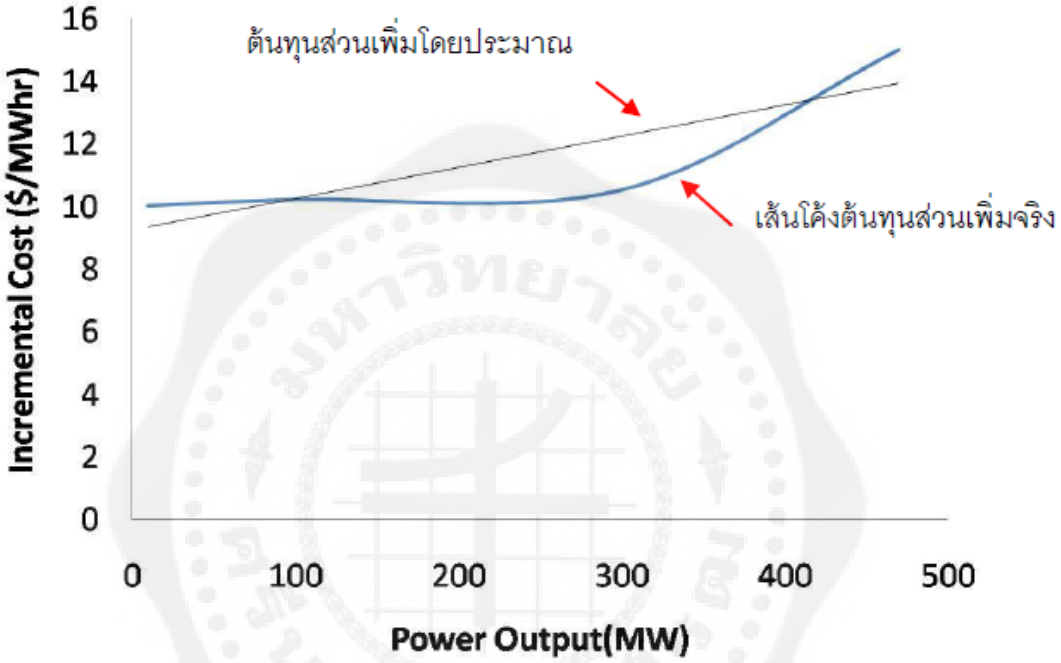
2.1.2 ต้นทุนส่วนเชื้อเพลิงส่วนเพิ่ม (Incremental fuel cost)

เมื่อนำเอาคณิตศาสตร์มาใช้ในการคำนวณหาการใช้เชื้อเพลิงขาเข้าให้น้อยที่สุดสำหรับการจ่ายไฟฟ้าให้กับโหลดรวมของเครื่องกำเนิดไฟฟ้า 2 หน่วย จากการทำงานของเครื่องกำเนิดไฟฟ้าทั้งสองที่มีอัตราส่วนเพิ่มของความร้อน (Incremental Heat Rates) เท่ากัน เนื่องจากเชื้อเพลิงมีราคาเป็นบาท หรือดอลลาร์ต่อ MBtu ดังนั้นเราจึงกล่าวได้ว่า “ต้นทุนการผลิตไฟฟ้าต่ำสุดเกิดขึ้นเมื่อเครื่องกำเนิดไฟฟ้าทั้งสองมีต้นทุนเชื้อเพลิงส่วนเพิ่มเท่ากัน” ต้นทุนเชื้อเพลิงส่วนเพิ่มสามารถหาได้จากความชันของเส้นโค้งขาเข้า-ขาออกของเครื่องกำเนิดไฟฟ้า 2 หน่วยโดยกำหนดให้

C_i คือ ราคาเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าหน่วยที่ i (Baht/hr หรือ $/hr)

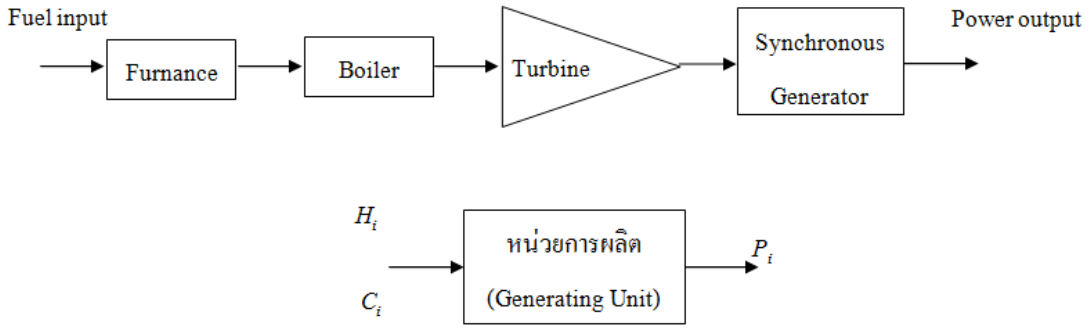
P_i คือ กำลังขาออกของเครื่องกำเนิดไฟฟ้าหน่วยที่ i (MW)

อัตราค่าเชื้อเพลิงที่เพิ่มขึ้น (Incremental Fuel Cost) เป็นค่าใช้จ่ายที่ถูกรวมเข้าไป เมื่อเพิ่มกำลังเอาท์พุทเป็น 1 MW แล้วค่าอัตราค่าเชื้อเพลิงที่เพิ่มขึ้นสามารถหาได้จากความชัน (Slope) ของกราฟอินพุท-เอาท์พุตและคูณด้วยค่าเชื้อเพลิงต่อบีทียู (Btu) ซึ่งสามารถแสดงความสัมพันธ์ระหว่างต้นทุนเชื้อเพลิงส่วนเพิ่มและกำลังขาออกได้ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 กราฟแสดงความสัมพันธ์ระหว่างต้นทุนส่วนเพิ่มของเชื้อเพลิงกับกำลังขาออก

ที่มา : ปุณณรัตน์ สมศิริวัฒนา และ อรวรรณ กังวาฬ. 2555



รูปที่ 2.3 แบบจำลองของโรงไฟฟ้าพลังงานความร้อน (Thermal Plant)

ที่มา : ปุณณรัตน์ สมศิริวัฒนา และ อรวรรณ กังวาฬ. 2555

จากรูปที่ 2.3 ความสัมพันธ์ระหว่างเชื้อเพลิงขาเข้า (Fuel input) กับกำลังขาออก (Power output) เขียนให้อยู่ในรูปสมการได้ดังนี้

$$H_i(P_i)=a_i+b_iP_i+c_iP_i^2 \tag{2.1}$$

โดยที่ $H_i(P_i)$ คือ อัตราความร้อนขาเข้าต่อหน่วยเมกะวัตต์ที่หน่วยผลิตมีหน่วยเป็น MBtu/hr

P_i คือ กำลังผลิตของเครื่องกำเนิดไฟฟ้าหน่วยที่ i

a_i,b_i,c_i คือ สัมประสิทธิ์คงตัวทางความร้อนของเครื่องกำเนิดไฟฟ้าหน่วยที่ i

2.1.3 สมการต้นทุนการดำเนินงานของหน่วยผลิต (Operating Cost Equation of Generating Unit)

จากรูปที่ 2.1 จะได้สมการอัตราพลังงานเชื้อเพลิงขาเข้าต่อกำลังไฟฟ้าจริงขาออกคือ

$$F_i=H_iP_i \text{ MBtu/hr} \tag{2.2}$$

โดยที่ F_i คือ ราคาเชื้อเพลิงที่ใช้มีหน่วยเป็น \$/h หรือ Baht/hr

H_i คือ อัตราความร้อนขาเข้าต่อหน่วยเมกะวัตต์ที่ผลิตมีหน่วยเป็น MBtu/MWh

P_i คือ กำลังผลิตมีหน่วยเป็นเมกะวัตต์ (MW)

จากรูปที่ 2.1 ที่แสดงถึงอัตราความร้อนต่อหน่วยเมกะวัตต์ที่ผลิตโดยทั่วไปสามารถเขียนเป็นสมการได้ดังนี้

$$H_i = a'_i + b'_i P_i + c'_i P_i^2 \text{ MBtu/MWh} \tag{2.3}$$

โดย a'_i, b'_i, c'_i คือ สัมประสิทธิ์คงตัวทางความร้อนของเครื่องกำเนิดไฟฟ้าหน่วยที่ i และจะได้สมการอัตราพลังงานเชื้อเพลิงขาเข้าคือ

$$F_i = a'_i + b'_i P_i + c'_i P_i^2 \text{ MBtu/hr} \tag{2.4}$$

เมื่อให้ K_i เป็นราคาเชื้อเพลิงในหน่วยบาทต่อเมกะบีทียูต่อชั่วโมง (Baht/MBtu) ซึ่งจะได้สมการราคาเชื้อเพลิง คือ

$$C_i = K_i F_i \text{ Baht/hr} \tag{2.5}$$

แทนค่าใหม่จะได้

$$C_i = K_i a'_i + K_i b'_i P_i + K_i c'_i P_i^2 \tag{2.6}$$

เมื่อกำหนดให้ $a_i=K_i a'_i$ $b_i=K_i b'_i$ และ $c_i=K_i c'_i$ ทำการจัดรูปใหม่ได้เป็นสมการราคาต้นทุนเชื้อเพลิง (Quadratic Function) คือ

$$C_i=a_i+b_iP_i+c_iP_i^2 \text{ Baht/hr} \tag{2.7}$$

สมการราคาต้นทุนรวม (Total Cost)

$$C_T=\sum_{i=1}^{n} C_i \text{ Baht/hr} \tag{2.8}$$

จะได้สมการกำลังไฟฟ้าขาออกรวม (Total Power Output) คือ

Power Generation = Power Demand + Power Losses

$$\sum_{i=1}^{n} P_i =P_D+P_L \tag{2.9}$$

ซึ่งขอบเขตการผลิตของแต่ละหน่วย (Generator Limits) คือ

$$P_i^{min} \leq P_i \leq P_i^{max} \; ; \; i=1,2,\ldots,n$$

ในการส่งกำลังไฟฟ้าจากโรงไฟฟ้าแต่ละแหล่งไปยังโหลดต่างๆหรือผู้ใช้ไฟฟ้านั้นอาจมีระยะทางไฟฟ้าระหว่างหน่วยผลิตและโหลดที่ไม่เท่ากันซึ่งจะทำให้เกิดการสูญเสียในรูปของความร้อนในสายส่ง เพื่อที่จะลดการสูญเสียรวมที่เกิดขึ้นนั้นหน่วยผลิตที่อยู่ใกล้กับโหลดมากกว่าอาจได้รับการจัดสรรเพิ่มกำลังผลิตเพื่อชดเชยให้กับกำลังผลิตของหน่วยผลิตที่อยู่ไกลซึ่งอาจจะถูกปรับให้น้อยลง โดยสมการการสูญเสียในสายส่ง (Transmission Losses) ที่เกิดขึ้นในระบบ คือ

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{i0} P_i + B_{00} \quad (2.10)$$

โดยที่ P_i คือ กำลังไฟฟ้าขาออกของหน่วยผลิตที่ i (MW)

P_j คือ กำลังไฟฟ้าขาออกของหน่วยผลิตที่ j (MW)

B_{ij} คือ สัมประสิทธิ์การสูญเสียของเมตริกซ์สี่เหลี่ยมที่ตำแหน่ง ij

B_{i0} คือ สัมประสิทธิ์การสูญเสียของเวกเตอร์ที่ตำแหน่ง i

B_{00} คือ สัมประสิทธิ์การสูญเสียของเวกเตอร์

n คือ จำนวนเครื่องกำเนิดไฟฟ้า

2.1.4 ฟังก์ชันราคาเชื้อเพลิง (Fuel Cost Function)

โดยปกติแล้วฟังก์ชันของราคาเชื้อเพลิงที่ใช้ในปัญหาการจัดสรรกำลังผลิตจะอยู่ในรูปแบบของสมการกำลังสอง (Quadratic Function) หรือสมการโพลิโนเมียล (Polynomial Equation) ซึ่งเป็นฟังก์ชันราคาที่ราบเรียบ (Smooth Cost Function) สามารถแสดงได้ในสมการต่อไปนี้

$$C_i(P_i) = a_i + b_i P_i + c_i P_i^2 \text{ Baht/hr} \quad (2.11)$$

โดยที่ a_i, b_i, c_i คือ สัมประสิทธิ์ของฟังก์ชันราคาค่าเชื้อเพลิงของเครื่องกำเนิดไฟฟ้าที่ i

สำหรับโรงไฟฟ้าหนึ่งโรงในแต่ละโรงจะประกอบไปด้วยเครื่องกำเนิดไฟฟ้าได้หลายหน่วย โดยแต่ละหน่วยนั้นจะมีการใช้เชื้อเพลิงที่แตกต่างกันซึ่งในการวิเคราะห์การจ่ายกำลังไฟฟ้าให้โหลดอย่างประหยัดจากเครื่องกำเนิดไฟฟ้าจำนวน n เครื่องโดยวิธีการทางคณิตศาสตร์เชิงวิเคราะห์จะได้ราคาเชื้อเพลิงรวมของระบบเท่ากับ

$$C_T = C_1 + C_2 + \ldots + C_n = \sum_{i=1}^{n} C_i \quad (2.12)$$

ปัญหาการจ่ายโหลดอย่างประหยัดจึงเป็นปัญหาการหาค่าที่เหมาะสม (Optimization Problem) โดยมีฟังก์ชันของราคาค่าเชื้อเพลิงเป็นฟังก์ชันเป้าหมาย (Objective Function) คือ

$$Minimize\ C_T = \sum_{i=1}^{n} C_i\ (P_i) \tag{2.13}$$

ในเครื่องกำเนิดไฟฟ้าบางชนิดมีการทำงานโดยใช้เชื้อเพลิงที่แตกต่างกันได้หลายชนิด (Multiple Fuels) การเปลี่ยนแปลงชนิดของเชื้อเพลิงจึงมีผลต่อฟังก์ชันราคาต้นทุน (Cost Function) จาก Quadratic Function จะเป็น Piecewise Quadratic Function แสดงสมการได้ดังนี้

$$C_i(P_i) = \begin{cases} a_{i1}+b_{i1}P_{i1}+c_{i1}P_{i1}^2 & ;fuel\ 1,\ P_i^{min} \le P_i \le P_{i1} \\ a_{i2}+b_{i2}P_{i2}+c_{i2}P_{i2}^2 & ;fuel\ 2,\ P_{i1} \le P_i \le P_{i2} \\ \vdots & \\ a_{ik}+b_{ik}P_{ik}+c_{ik}P_{ik}^2 & ;fuel\ k,\ P_{ik-1} \le P_i \le P_i^{max} \end{cases} \tag{2.14}$$

โดยที่ C_T คือ ราคาเชื้อเพลิงรวม (Baht/hr)

C_i คือ กำลังขาออกของเครื่องกำเนิดไฟฟ้า (MW)

P_D คือ กำลังไฟฟ้าที่โหลดต้องการใช้ (MW)

P_L คือ กำลังไฟฟ้าสูญเสียที่เกิดขึ้นในสายส่ง (MW)

P_i^{min} คือ กำลังขาออกที่น้อยที่สุดของเครื่องกำเนิดไฟฟ้าที่ i (MW)

P_i^{max} คือ กำลังขาออกที่มากที่สุดของหน่วยผลิตที่ i (MW)

n คือ จำนวนเครื่องกำเนิดไฟฟ้าที่ทำงาน

2.1.5 ผลกระทบที่เกิดจากจุดวาล์ว (Valve-Point Effects)

ผลกระทบที่เกิดขึ้นจากจุดวาว์ลคือผลกระทบที่ทำให้ค่าฟังก์ชันของราคาเชื้อเพลิงเพิ่มขึ้น โดยการเพิ่มขึ้นนี้เป็นแบบไม่เป็นเชิงเส้น (Non-linear) ซึ่งเมื่อรวมผลกระทบจากจุดวาล์วเข้าไปในสมการหาค่าการผลิตจะได้สมการดังนี้

$$C_i(P_i) = a_i+b_iP_i+c_iP_i^2 + \left| e_i\ sin\left(f_i\left(P_{i,min}-P_i\right)\right)\right| \tag{2.15}$$

โดยที่ e_i และ f_i คือ ค่าสัมประสิทธิ์ของฟังก์ชันราคาเชื้องเพลิงที่เกิดจากจุดวาล์วของเครื่องกำเนิดไฟฟ้าที่ i

ดังนั้นในกรณีที่พิจารณาทั้งเชื้อเพลิงหลายเชื้อเพลิง มีผลกระทบที่เกิดขึ้นจากจุดวาล์ว และคิดค่าความสูญเสียที่เกิดขึ้นในระบบจะได้สมการต้นทุนดำเนินงานของหน่วยผลิตดังนี้

$$C_i(P_i)=\begin{Bmatrix} a_{i1}+b_{i1}P_{i1}+c_{i1}P_{i1}^2+\left|e_{i1}\sin\left(f_{i1}\left(P_{i1,min}-P_{i1}\right)\right)\right|+P_{L1}\ ;fuel\ 1,\ P_i^{min}\le P_i\le P_{i1} \\ a_{i2}+b_{i2}P_{i2}+c_{i2}P_{i2}^2+\left|e_{i2}\sin\left(f_{i2}\left(P_{i2,min}-P_{i2}\right)\right)\right|+P_{L2}\ ;fuel\ 2,\ P_{i1}\le P_i\le P_{i2} \\ \vdots \\ a_{ik}+b_{ik}P_{ik}+c_{ik}P_{ik}^2+\left|e_{ik}\sin\left(f_{ik}\left(P_{ik,min}-P_{ik}\right)\right)\right|+P_{Lk}\ ;fuel\ k,\ P_{ik-1}\le P_i\le P_i^{max} \end{Bmatrix} \quad (2.16)$$

2.2 วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม

Kenneth Price และ Rainer Storn ได้คิดค้น Differential Evolution Algorithm (DEA) ขึ้นในปี ค.ศ. 1995 จากความพยายามที่จะแก้ปัญหา Polynomial Fitting โดยที่ DEA นั้นเป็นวิธีการหาค่าเหมาะสมแบบสโตคาสติกมีพื้นฐานมาจากการหาคำตอบของปัญหาโดยวิธีสุ่มเลือก พัฒนามาจาก Genetic Algorithm (GA) ข้อแตกต่างที่สำคัญคือ GA จะแปลงตัวแปรการตัดสินใจ (Decision Variables) ให้เป็นรหัสเลขฐานสอง (Binary Code) แต่ DEA มีโครงสร้างที่ซับซ้อนน้อยกว่า เนื่องจากสามารถใช้ค่าจริง (Floating Point Number) ในการประยุกต์ใช้หาคำตอบของปัญหาได้ จึงทำให้ DEA เป็นวิธีที่มีความเร็วและประสิทธิภาพในการหาคำตอบสูงกว่า ง่ายต่อการนำไปใช้ เหมาะสำหรับการแก้ปัญหาประเภท Non-Linear และ Non-Differentiate Function

วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมเป็นวิธีการแก้ปัญหาแบบขนาน โดยใช้ประชากร P ขนาน N_p ในทุกเจนเนอเรชั่น G ระหว่างการแก้ปัญหา DEA ใช้ประชากร $P^{(G)}$ ขนาด N_p

$$P^{(G)}=\left[X_1^{(G)},\ldots,X_i^{(G)},\ldots,X_{N_P}^{(G)}\right];i=1,\ldots,N_P \quad (2.17)$$

X_i เป็นเวกเตอร์ที่มีขนาด D โดย

$$X_i^{(G)} = \left| x_{1,i}^{(G)}, \ldots, x_{j,i}^{(G)}, \ldots, x_{D,i}^{(G)} \right| ; i=1,\ldots,N_P; j=1,\ldots,D \quad (2.18)$$

วิธีคิดแบบ DEA จะประกอบด้วย 4 ขั้นตอนคือ การตั้งสมมติค่าเริ่มต้น (Initialization) การมิวเตชั่น (Mutation) การครอสโอเวอร์ (Crossover) และการคัดเลือก (Selection)

2.2.1 การสมมติค่าเริ่มต้น (Initialization)

สำหรับการสมมติค่าเริ่มต้นในขั้นตอนของวิธีการคิดแบบ DEA นั้น จะต้องมีการกำหนดขอบเขตบนและขอบเขตล่าง ของตัวแปรการตัดสินใจแต่ละตัว โดยขอบเขตนี้จะต้องครอบคลุมจุดที่ให้คำตอบที่ดีที่สุด จากนั้นให้สุ่มหาคำตอบที่เป็นไปได้เริ่มต้น (Initial Population) โดยกำหนดให้โอกาสที่จะถูกเลือกของคำตอบมีค่าสม่ำเสมอ (Uniform Probability Distribution) โดยคำตอบแต่ละคำตอบซึ่งเรียกว่า Decision Vector จะมีมิติเท่ากับ D และจำนวนคำตอบที่เป็นไปได้เริ่มต้นเท่ากับ N_p จากนั้นคำนวณหา Function Value ของแต่ละคำตอบเริ่มต้นที่เป็นไปได้ ดังสมการที่ 2.19

$$x_{j,i}^{(G=0)} = x_j^{min} + rand_j[0,1].\left(x_j^{max} - x_j^{min}\right) \quad (2.19)$$

เมื่อ $i=1,\ldots,N_p$

$j=1,\ldots,D$

$x_{j,i}^{(G=0)}$ คือ ค่าเริ่มต้นของพารามิเตอร์ j^{th} ของเวกเตอร์อิสระ i^{th}

x_j^{min}, x_j^{max} คือ ค่าต่ำสุดและสูงสุดของพารามิเตอร์ j^{th} ตามลำดับ

2.2.2 การมิวเตชั่น (Mutation)

เมื่อทำการสมมติค่าเริ่มต้นแล้ว ในขั้นตอนต่อไปจะใช้ขั้นตอนพื้นฐานของวิธีทางพันธุกรรมสามอย่าง คือ มิวเตชั่น ครอสโอเวอร์ และการคัดเลือกเพื่อให้ได้ประชากรในเจเนอเรชั่นถัดไปคือ $P^{(G+1)}$ โดยใช้ประชากรในปัจจุบัน $P^{(G)}$

การมิวเตชั่นจะเริ่มต้นด้วยการกำหนดเวกเตอร์เป้าหมาย $(X_{i,G})$ โดยที่ $i=1,2,3,\ldots,N_P$ แล้วจึงทำการสุ่มเลือก Random Vector จำนวน 3 เวกเตอร์ $(X_{r1,G}, X_{r2,G}, X_{r3,G})$ จากประชากรตั้งต้นโดยที่จะต้องไปซ้ำกับเวกเตอร์เป้าหมาย แล้วจึงจะคำนวณหามิวแตนท์เวกเตอร์ $(V_{i,G+1})$ จากความสัมพันธ์

$$V_{i,G+1}=X_{r1,G}+F(X_{r2,G}-X_{r3,G}) \quad (2.20)$$

โดยที่ $X_{i,G}$ คือ เวกเตอร์เป้าหมาย (Target Vector)

$V_{i,G+1}$ คือ มิวแตนท์เวกเตอร์ (Mutant Vector)

F คือ จำนวนจริงที่มีค่าคงที่และมีค่าระหว่าง 0 ถึง 2 (Weighing Factor)

$X_{r1,G},X_{r2,G},X_{r3,G}$ คือ Random Vector

การมิวเตชั่นนั้นจะมีอยู่ด้วยกัน 10 แบบซึ่งแบบที่ 1-5 นั้นจะเป็นแบบดั้งเดิมที่ถูกเสนอขึ้นโดยสตอร์น และแบบที่ 6-10 เป็นแบบใหม่ ดังนี้

2.2.2.1 DEA1 ในวิธีแรกนี้เวกเตอร์ดัดแปลงจะถูกสร้างขึ้นจากสมการที่ 2.20

$$V_i^{(G)}=X_{best}^{(G)}+F\left(X_{r2}^{(G)}-X_{r3}^{(G)}\right);i=1,\ldots,N_P \quad (2.21)$$

2.2.2.2 DEA2 จะมีวิธีการทำงานคล้ายกับวิธีแรกแต่จะแตกต่างกันโดยที่วิธีนี้จะทำงานอยู่บนฐานของเวกเตอร์สมาชิกที่สุ่มเลือกมา $X_{r1}^{(G)}$ คำนวณได้ดังสมการที่ 2.22

$$V_i^{(G)}=X_{r1}^{(G)}+F\left(X_{r2}^{(G)}-X_{r3}^{(G)}\right);i=1,\ldots,N_P \quad (2.22)$$

2.2.2.3 DEA3 วิธีนี้จะทำการดัดแปลงเวกเตอร์โดยการใช้ความแตกต่างจากวิธีที่ดีที่สุดและความแตกต่างของเวกเตอร์สมาชิกที่สุ่มเลือกมาดังสมาการที่ 2.23

$$V_i^{(G)}=X_i^{(G)}+\lambda\left(X_{best}^{(G)}-X_i^{(G)}\right)+F\left(X_{r1}^{(G)}-X_{r2}^{(G)}\right);i=1,\ldots,N_P \quad (2.23)$$

2.2.2.4 DEA4 มีการดัดแปลงเวกเตอร์โดยการใช้ค่าความแตกต่างของสองเวกเตอร์สมาชิกที่สุ่มเลือกมาดังสมการที่ 2.24

$$V_i^{(G)} = X_{best}^{(G)} + F\left(X_{r1}^{(G)} + X_{r2}^{(G)} - X_{r3}^{(G)} - X_{r4}^{(G)}\right); i=1,\ldots,N_P \quad (2.24)$$

2.2.2.5 DEA5 มีการทำงานคล้ายกับ DEA4 แต่จะแตกต่างกันโดยการแทนค่าที่ดีที่สุดใน DEA4 ด้วยเวกเตอร์สมาชิกที่สุ่มเลือกมาดังสมการที่ 2.25

$$V_i^{(G)} = X_{r5}^{(G)} + F\left(X_{r1}^{(G)} + X_{r2}^{(G)} - X_{r3}^{(G)} - X_{r4}^{(G)}\right); i=1,\ldots,N_P \quad (2.25)$$

2.2.2.6 DEA6 มีการทำงานคล้ายกับ DEA1 แต่จะแตกต่างกันที่วิธีนี้จะแทนเวกเตอร์ที่สุ่มเลือกมาเป็นค่าเวกเตอร์สมาชิกที่ดีที่สุดกับค่าเวกเตอร์ที่จะทำการดัดแปลงดังสมการที่ 2.26

$$V_i^{(G)} = X_{best}^{(G)} + F\left(X_{best}^{(G)} - X_i^{(G)}\right); i=1,\ldots,N_P \quad (2.26)$$

2.2.2.7 DEA7 มีหลักการคิดเหมือนกับ DEA4 แต่จะใช้เวกเตอร์ 3 เวกเตอร์ที่ต่างกันในการดัดแปลงดังสมการที่ 2.27

$$V_i^{(G)} = X_{best}^{(G)} + F\left(X_{best}^{(G)} - X_i^{(G)} - X_{r1}^{(G)} - X_{r2}^{(G)}\right); i=1,\ldots,N_P \quad (2.27)$$

2.2.2.8 DEA8 มีหลักการคิดเหมือนกับ DEA3 แต่จะต่างกันตรงที่ DEA3 จะคิดบนฐานของเวกเตอร์สมาชิกที่ดีที่สุด ส่วน DEA8 จะคิดบนฐานของเวกเตอร์ตัวที่ถูกดัดแปลงดังสมการที่ 2.28

$$V_i^{(G)} = X_{best}^{(G)} + \lambda\left(X_{best}^{(G)} - X_i^{(G)}\right) + F\left(X_{r1}^{(G)} - X_{r2}^{(G)}\right); i=1,\ldots,N_P \quad (2.28)$$

2.2.2.9 DEA9 มีการทำงานคล้ายกับ DEA5 แต่จะแตกต่างกันที่วิธีนี้จะใช้การสุ่มเวกเตอร์สองเวกเตอร์เท่านั้นดังสมการที่ 2.29

$$V_i^{(G)} = X_{best}^{(G)} + F\left(X_{best}^{(G)} + X_i^{(G)} - X_{r1}^{(G)} - X_{r2}^{(G)}\right); i=1,\ldots,N_P \tag{2.29}$$

2.2.2.10 DEA10 วิธีนี้จะคล้ายกับ DEA6 แต่จะแทน $X_i^{(G)}$ ด้วย $X_i^{(G-1)}$ จากเจเนอเรชั่นก่อนหน้าดังสมการที่ 2.30

$$V_i^{(G)} = X_{best}^{(G)} + F\left(X_{best}^{(G)} - X_{best}^{(G-1)}\right); i=1,\ldots,N_P \tag{2.30}$$

2.2.3 การครอสโอเวอร์ (Crossover)

สำหรับการครอสโอเวอร์นั้น จะเป็นการช่วยให้ DEA มีความหลากหลายมากขึ้น โดยจะสร้างเวกเตอร์ทดลอง $u_{j,i}^{(G)}$ ด้วยการผสมผสานระหว่างมิวแตนท์เวกเตอร์ V_i กับเวกเตอร์เป้าหมาย X_i โดยสามารถหาค่าเวกเตอร์ทดลองได้จากสมการ

$$U_i^{(G)} = u_{j,i}^{(G)} = \begin{cases} V_{j,i}^{(G)} \text{ ถ้า } rand_j(0,1) \le CR \text{ หรือ } j=s \\ X_{j,i}^{(G)} \text{ ในกรณีอื่นๆ} \end{cases} \tag{2.31}$$

เมื่อ $u_{j,i}^{(G)}$ = Trial Vector

$V_{j,i}^{(G)}$ = Mutant Vector

$X_{j,i}^{(G)}$ = Target Vector

$rand_j$ = ค่าที่สุ่มขึ้นมา มีค่า (0,1)

CR = Crossover Constant มีค่าเป็นเลขจำนวนจริงระหว่าง 0 ถึง 1

S = ค่า Index จากการสุ่มเลือกมีค่าเป็นเลขจำนวนเต็มระหว่าง 1,2,…,D

2.2.4 การคัดเลือก (Selection)

การคัดเลือกเป็นกระบวนการคัดสรรหาคำตอบ โดยมีวิธีการคือเปรียบเทียบค่าของฟังก์ชั่นของเวกเตอร์ทดลองกับเวกเตอร์เป้าหมาย ถ้าเวกเตอร์ทดลองให้ค่าของฟังก์ชันที่ดีกว่า มันก็จะแทนที่เวกเตอร์เป้าหมายในเจเนอเรชั่นต่อไป จากนั้นก็จะทำซ้ำขั้นตอนที่ 2.2 ถึงขั้นตอนที่ 2.4 จนครบทุกเวกเตอร์ในเจเนอเรชั่นปัจจุบัน และแทนที่เจเนอเรชั่นปัจจุบันด้วยเจเนอเรชั่นต่อไป แล้วทำซ้ำกระบวนการทั้งหมดจนถึง Stopping Criteria

$$X_i^{(G+1)} = \begin{cases} U_i^{(G)} \text{ถ้า } f\left(U_i^{(G)}\right) \le f\left(X_i^{(G)}\right) \\ X_i^{(G)} \text{ ในกรณีอื่นๆ} \end{cases} \tag{2.32}$$

2.2.5 การทำงานของ DEA

สำหรับการทำงานของ DEA นั้นจะเริ่มทำงานโดยการกำหนดค่าพารามิเตอร์ควบคุมต่างๆ ของ DEA ได้แก่ ตัวประกอบมิวเตชั่น (F) ค่าคงที่การครอสโอเวอร์ (CR) และจำนวนประชากร (N_p) แล้วจึงทำตามขั้นตอนที่กล่าวมาข้างต้นคือ กำหนดค่าเริ่มต้นมิวเตชั่น ครอสโอเวอร์ และการคัดเลือก โดยคำนวณเป็นวงรอบวนไปเรื่อยๆจนกว่าจะได้คำตอบที่ดีที่สุดจึงหยุดคำนวณ โดยขั้นตอนการทำงานสามารถอธิบายได้ดังต่อไปนี้

ขั้นตอนที่ 1 รับค่าพารามิเตอร์ควบคุมต่างๆ ได้แก่ F CR N_p และD

ขั้นตอนที่ 2 สมมติค่าประชากรเริ่มต้นโดยเป็นเวกเตอร์ขนาด D จำนวน N_p เวกเตอร์

ขั้นตอนที่ 3 ประเมินค่าประชากรที่สมมติได้จากขั้นตอนที่ 2

ขั้นตอนที่ 4 สร้างเวกเตอร์มิวเตชั่นจากประชากรที่มีจำนวน N_p เวกเตอร์

ขั้นตอนที่ 5 สร้างเวกเตอร์ทดลองจากการครอสโอเวอร์จำนวน N_p เวกเตอร์

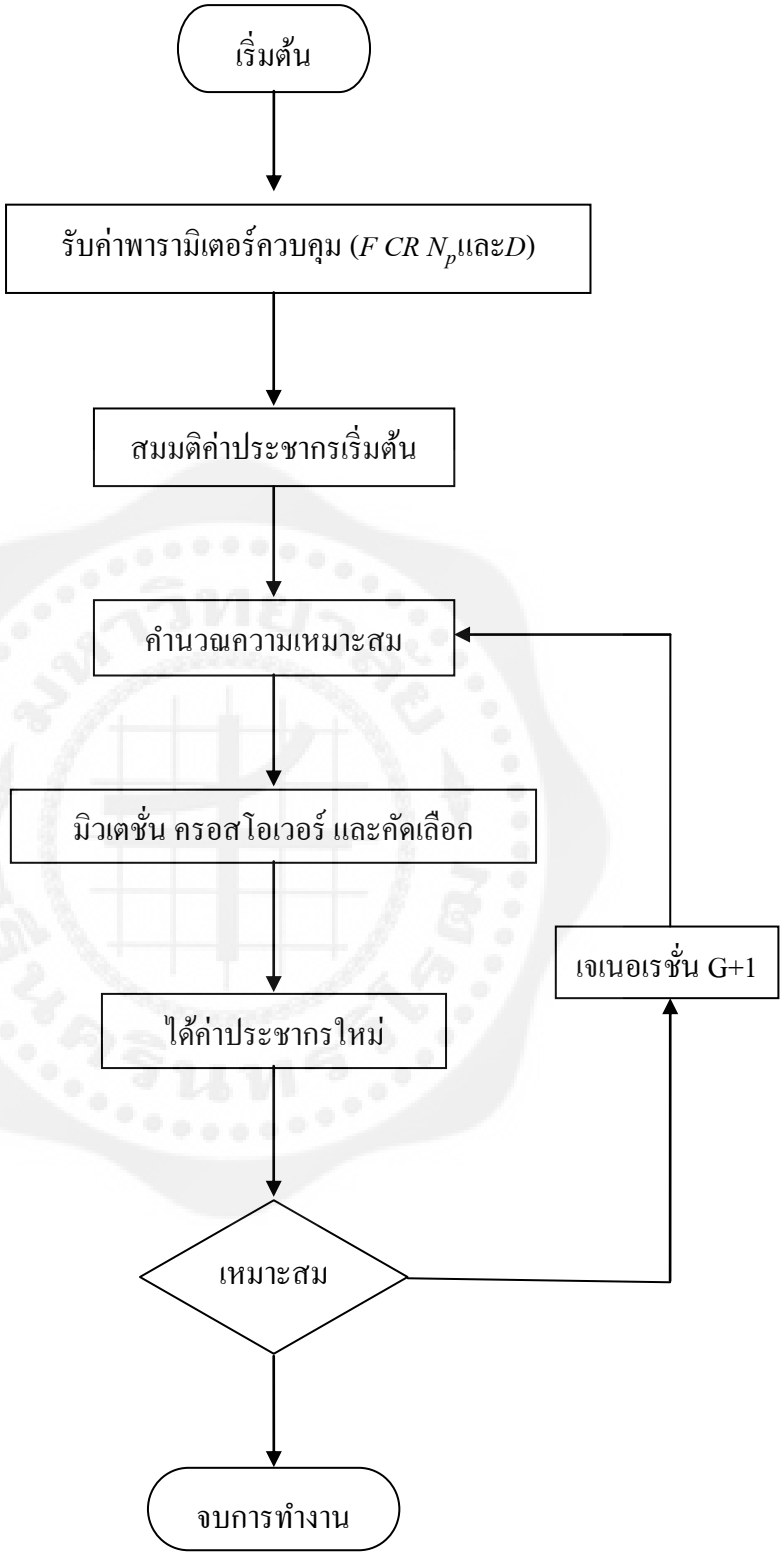
ขั้นตอนที่ 6 ประเมินค่าเวกเตอร์ทดลองที่สร้างขึ้น

ขั้นตอนที่ 7 ทำการคัดเลือกโดยเปรียบเทียบค่าระหว่างประชากรเดิมกับเวกเตอร์ทดลองที่สร้างขึ้น

ขั้นตอนที่ 8 นำประชากรที่ทำการคัดเลือกแล้วใช้เป็นประชากรสำหรับในรอบการคำนวณถัดไป

ขั้นตอนที่ 9 ทำซ้ำขั้นตอนที่ 4 ถึงขั้นตอนที่ 8 จนกว่าค่าที่ดีที่สุดของประชากรจะเป็นค่าที่เหมาะสมหรือจบการคำนวณรอบคำนวณสูงสุด

สรุปขั้นตอนการทำงานทั้งหมดนี้ได้ดังแผนผังในรูป 2.4



รูปที่ 2.4 แผนผังแสดงการทำงาน DEA

2.3 ทบทวนวรรณกรรมที่เกี่ยวข้อง

2.3.1 ดร.ธนาธิป สุ่มอิ่ม ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ ได้ศึกษาและทำการวิจัยเรื่อง Self-Adaptive Differential Evolution Algorithm for Economic Dispatch with Transmission Losses Consideration โดยงานวิจัยนี้ได้เสนอเกี่ยวกับการแก้ปัญหาเรื่อง Conventional Economic Dispatch โดยพิจารณาการสูญเสียในสายส่งไฟฟ้ากำลัง ซึ่งวัตถุประสงค์ของปัญหาด้านการจัดสรรกำลังการผลิตนี้ คือการลดต้นทุนเชื้อเพลิง สำหรับโรงไฟฟ้าพลังงานความร้อน โดยพิจารณาข้อจำกัดทางด้านเศรษฐศาสตร์และเทคนิค

2.3.2 Vo Ngoc Dieu และ Weerakorn Ongsakul สถาบันเทคโนโลยีแห่งเอเชีย ได้ทำการวิจัยเรื่อง Economic Dispatch with Multiple Fuel Types by Enhanced Augmented Lagrange Hopfield Network (ALHN) โดยงานวิจัยนี้ได้เสนอเกี่ยวกับการแก้ปัญหาการจัดสรรกำลังผลิตด้วย Piecewise Quadratic Cost Functions ซึ่งเกิดจากการรวมตัวของ Hopfield Neural Network and Augmented Lagrangian Relaxation Function เป็นฟังก์ชันพลังงาน ซึ่งถูกปรับปรุงโดย A Heuristic Search Algorithm สำหรับนำมาตัดสินใจเลือกชนิดของเชื้อเพลิง ซึ่งได้ทำการออกแบบเป็น 2 กรณี กรณีแรก A Heuristic Algorithm ถูกอ้างอิงจากต้นทุนการผลิตเฉลี่ยของหน่วยการผลิตที่ใช้ในการระบุชนิดของเชื้อเพลิงที่เหมาะสมที่สุดของความต้องการโหลด กรณีสุดท้าย ALHN ถูกประยุกต์ใช้สำหรับแก้ปัญหาการจัดสรรกำลังผลิตเพื่อหาค่าที่ดีที่สุดเพื่อเลือกชนิดของเชื้อเพลิง

2.3.3 นายปุณณรัตน์ สมศิริวัฒนา และนางสาวอรวรรณ กังวาฬ มหาวิทยาลัยศรีนครินทรวิโรฒ ได้ทำการศึกษาและทำการวิจัยเรื่อง การจ่ายโหลดอย่างประหยัดสำหรับหน่วยการผลิตไฟฟ้าแบบเลือกใช้งานเชื้อเพลิงได้หลายชนิด และมีผลจากจุดวาล์วโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ เพื่อช่วยในการแก้ปัญหาการจ่ายโหลดอย่างประหยัดโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้มาแก้ปัญหา ซึ่งได้ทำการแบบออกเป็น 2 กรณีคือ กรณีแรกได้ทำศึกษาระบบที่เครื่องกำเนิดไฟฟ้าโดยสามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิดแต่ไม่พิจารณาผลจากจุดวาล์ว และกรณีที่สองได้ศึกษาระบบที่เครื่องกำเนิดไฟฟ้าโดยสามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิดและพิจารณาผลจากจุดวาล์ว ทั้งสองกรณีนั้นจะไม่คิดการสูญเสียในระบบสายส่ง จากการทดสอบวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ที่นำไปประยุกต์ใช้ทั้ง 2 กรณีนั้นพบว่าวิธีการที่มีประสิทธิภาพและสามารถหาคำตอบได้เหมาะสมสำหรับการแก้ปัญหาการจ่ายโหลดอย่างประหยัด เมื่อนำไปเปรียบเทียบกับคำตอบที่ได้กับวิธีอื่นๆ แสดงให้เห็นว่าวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้สามารถให้ค่าที่เหมาะสมและดีกว่าวิธีอื่นๆที่นำมาเปรียบเทียบ

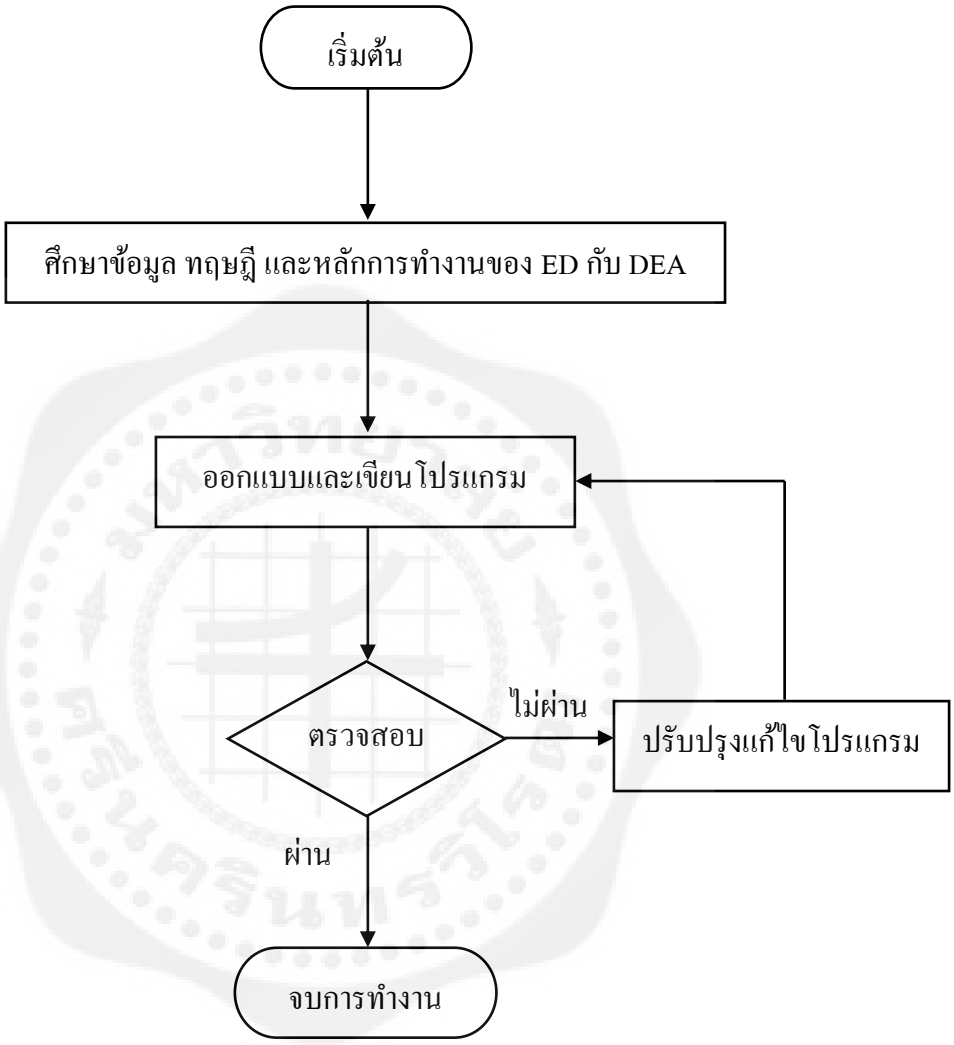
2.3.4 นายชนะพงษ์ สุขกลึง และนายอัษฎางค์ ศรีสุนทรศิริ มหาวิทยาลัยศรีนครินทรวิโรฒ ได้ทำการศึกษาและทำการวิจัยเรื่องการจัดสรรกำลังผลิตโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ เพื่อนำไปแก้ปัญหาการจ่ายโหลดอย่างประหยัดโดยมีจำนวนบัสทั้งหมด 30 บัส และมีเครื่องกำเนิดไฟฟ้าอยู่ 6 จุด ซึ่งมีการพิจาณาค่าการสูญเสียในระบบสายส่งโดยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ จากการทดสอบวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้สามารถหาคำตอบได้เหมาะสม เมื่อนำไปเปรียบเทียบกับคำตอบที่ได้กับวิธีแบบดั้งเดิมแสดงให้เห็นว่าวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้มีประสิทธิภาพในการทำงานได้เป็นอย่างดี และมีความน่าเชื่อถือในการหาคำตอบที่เหมาะสม

บทที่ 3
ขั้นตอนและการดำเนินงาน

ในการแก้ปัญหาการจัดกำลังผลิตเพื่อการจ่ายโหลดอย่างประหยัดนั้นจำเป็นที่จะต้องหาค่าปริมาณการผลิตกำลังไฟฟ้าที่มาจากเครื่องกำเนิดไฟฟ้าในแต่ละเครื่องของหน่วยการผลิต โดยจะต้องมีค่าใช้จ่ายหรือต้นทุนหรือค่าเชื้อเพลิงที่ใช้ในการผลิตต่ำที่สุด ซึ่งจะต้องมีปริมาณกำลังไฟฟ้ารวมที่เพียงพอต่อความต้องการใช้ไฟฟ้า ณ ช่วงเวลานั้นๆ โดยต้องมีการคิดค่าความสูญเสียที่เกิดขึ้นในระบบสายส่งระหว่างหน่วยการผลิตในแต่ละหน่วยกับโหลดที่มีความต้องการไฟฟ้า และที่สำคัญควรมีการพิจารณาถึงชนิดราคาเชื้อเพลิงที่ใช้ในการผลิตให้มีต้นทุนการผลิตที่น้อยที่สุดเท่าที่เป็นไปได้ ซึ่งค่าราคาต้นทุนนั้นขึ้นอยู่กับอัตราการผลิตของแต่ละหน่วยการผลิตและยังมีผลเนื่องมาจากจุดวาล์ว

ในการออกแบบด้วยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้วนั้นเป็นวิธีที่หาคำตอบของปัญหาโดยการสุ่มเลือกค่าที่เหมาะสมและเป็นวิธีที่มีประสิทธิภาพ โดยถูกนำมาใช้ในการแก้ไขปัญหาการจัดสรรกำลังผลิตของหน่วยผลิต ซึ่งพัฒนามาจากพื้นฐานแนวคิดของกระบวนการคัดสรรตามธรรมชาติ (Genetic Algorithm: GA) วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมประกอบด้วย 4 ขั้นตอนหลักได้แก่ การตั้งสมมติค่าเริ่มต้น (Initialization) การมิวเตชั่น (Mutation) การครอสโอเวอร์ (Crossover) และการคัดเลือก (Selection) โดยจำเป็นต้องกำหนดค่าพารามิเตอร์ 2 ตัวคือ ตัวประกอบมิวเตชั่น (F) และค่าคงที่ครอสโอเวอร์ (CR) ซึ่งมีตัวของอัลกอริธึมช่วยในการหาค่าที่เหมาะสมของทั้ง 2 พารามิเตอร์ ในส่วนของผู้ใช้งานโปรแกรมนั้นจะต้องกำหนดช่วงของค่าพารามิเตอร์ทั้งสอง เพื่อที่จะทำให้โปรแกรมได้ค่าคำตอบที่เหมาะสมรวดเร็วและดียิ่งขึ้น

3.1 แผนผังขั้นตอนการทำงาน



รูปที่ 3.1 แผนผังแสดงขั้นตอนการทำงาน

3.2 ขั้นตอนการดำเนินงาน

ขั้นตอนที่ 1 ค้นคว้าหาหัวข้อการวิจัย

ขั้นตอนที่ 2 ทำการศึกษาหาข้อมูลและทฤษฎีที่เกี่ยวกับปัญหาการจัดสรรกำลังผลิตของหน่วยการผลิตและวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึม

ขั้นตอนที่ 3 ทำการออกแบบและเขียนโปรแกรมการแก้ปัญหาการจ่ายโหลดอย่างประหยัดแบบเลือกใช้งานได้หลายเชื้อเพลิงและพิจารณาการสูญเสียของระบบโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมที่ได้รับการปรับปรุงแล้ว โดยใช้โปรแกรม MATLAB

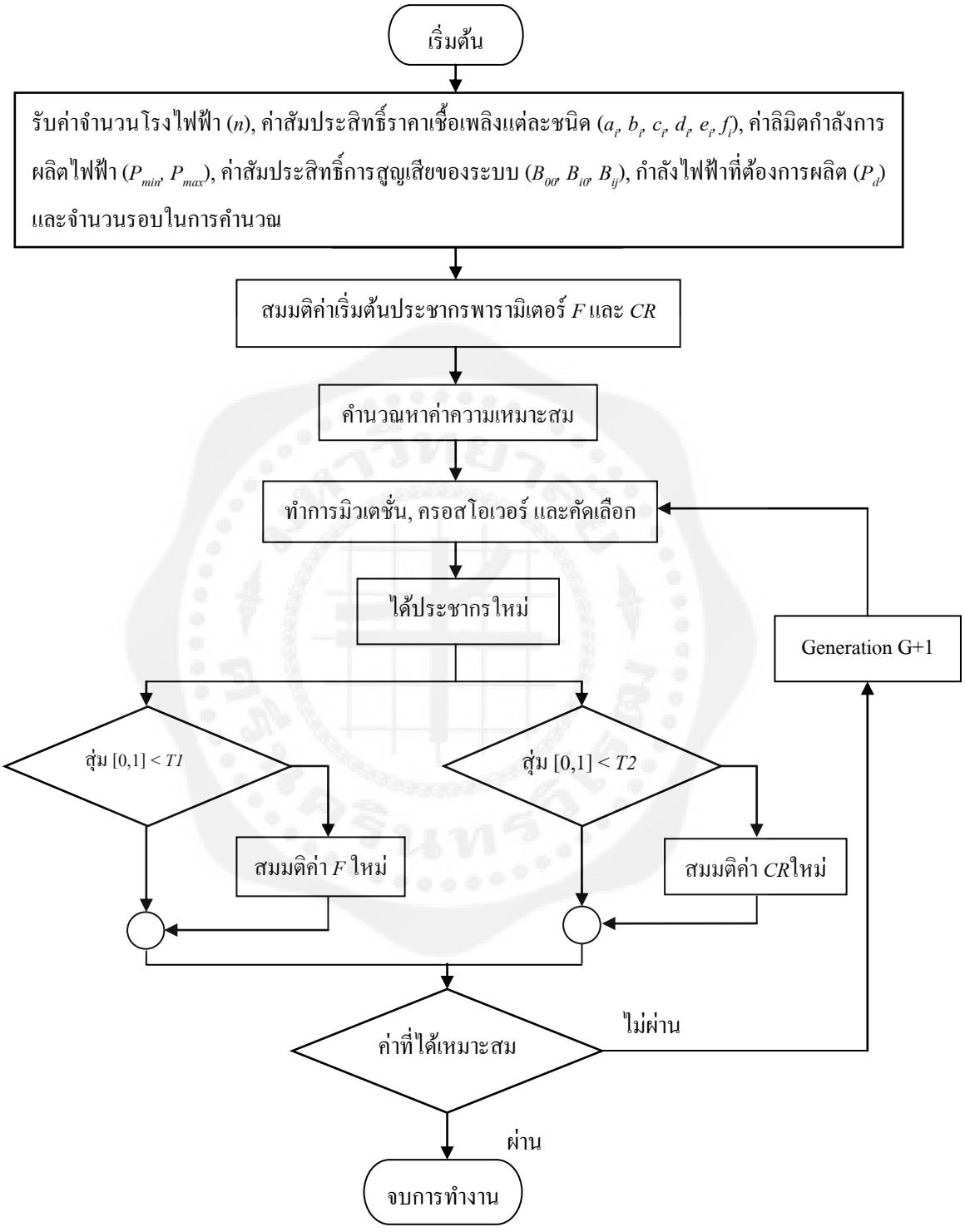
ขั้นตอนที่ 4 ทำการทดสอบและปรับปรุงแก้ไขโปรแกรม

ขั้นตอนที่ 5 ทำซ้ำในขั้นตอนที่ 3 ถึงขั้นตอนที่ 4 จนได้โปรแกรมที่สามารถใช้งานได้ตามวัตถุประสงค์

3.3 แสดงขั้นตอนและวิธีการดำเนินงาน

โครงงานนี้ได้ทำการศึกษาและแก้ไขปัญหาการจัดสรรกำลังผลิตเพื่อการจ่ายโหลดอย่างประหยัด โดยการเขียนโปรแกรม MATLAB R2012b โดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ (Self-Adaptive Differential Evolution Algorithm) โดยใช้ระบบข้อมูลที่มีเครื่องกำเนิดไฟฟ้าทั้งหมด 10 เครื่อง ในบทนี้ได้แสดงขั้นตอนการทำงานของโปรแกรมและผลการทดสอบที่มีค่าความต้องการกำลังไฟฟ้า 2,400 2,500 2,600 และ2,700 MW พร้อมทั้งทำการพิจารณาอยู่ 2 กรณีคือ กรณีที่มีการพิจารณาเครื่องกำเนิดไฟฟ้าที่มีการใช้งานเชื้อเพลิงหลายชนิดและกรณีที่มีการพิจารณาเครื่องกำเนิดไฟฟ้าที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิดและมีผลจากจุดวาล์ว ซึ่งทั้ง 2 กรณีนี้จะพิจารณาค่าความสูญเสียที่เกิดขึ้นกับระบบสายส่งเข้าไปด้วย โดยผลที่ได้จะนำไปเปรียบเทียบกับการจัดสรรกำลังการผลิตเพื่อการจ่ายโหลดอย่างประหยัดในแบบก่อนๆ และจะวิเคราะห์ผลการทดลองไว้ในบทที่ 4 พร้อมทั้งสรุปผลการทดลองไว้ในบทที่ 5

3.4 หลักการคิดโปรแกรม



รูปที่ 3.2 แสดงแผนผังขั้นตอนการทำงานของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัว เองได้

3.5 ขั้นตอนการทำงานของโปรแกรมโดยวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้

ขั้นตอนที่ 1 รับค่าพารามิเตอร์ควบคุมต่างๆ ได้แก่ จำนวนโรงไฟฟ้า (n), ค่าสัมประสิทธิ์ราคาเชื้อเพลิงแต่ละชนิด (a_i, b_i, c_i, d_i, e_i, f_i), ค่าลิมิตกำลังการผลิตไฟฟ้า (P_{min}, P_{max}), ค่าสัมประสิทธิ์การสูญเสียของระบบ (B_{00}, B_{i0}, B_{ij}), กำลังไฟฟ้าที่ต้องการผลิต (P_d) และจำนวนรอบในการคำนวณ

ขั้นตอนที่ 2 กำหนดค่าเปอร์เซนต์ในการเปลี่ยนค่าของพารามิเตอร์ F และ CR คือ $T1$ และ $T2$ ตามลำดับ

ขั้นตอนที่ 3 คำนวณหาค่าพารามิเตอร์ F และ CR เริ่มต้น จำนวน N_p ค่า กล่าวคือเวกเตอร์สมาชิก 1 เวกเตอร์จะมีพารามิเตอร์ F และ CR เป็นของตัวเอง เพื่อนำไปใช้ในการมิวเตชั่น และครอสโอเวอร์ของตัวเอง

ขั้นตอนที่ 4 สมมติค่าประชากรเริ่มต้น โดยเป็นเวกเตอร์ขนาด D จำนวน N_p เวกเตอร์

ขั้นตอนที่ 5 ประเมินค่าประชากรที่สมมติมาจากขั้นตอนที่ 3

ขั้นตอนที่ 6 สร้างเวกเตอร์มิวเตชั่นจากประชากรที่มีจำนวน N_p เวกเตอร์

ขั้นตอนที่ 7 สร้างเวกเตอร์ทดลองจากการครอสโอเวอร์จำนวน N_p เวกเตอร์

ขั้นตอนที่ 8 ประเมินค่าเวกเตอร์ทดลองที่สร้างขึ้น

ขั้นตอนที่ 9 ทำการคัดเลือกโดยเปรียบเทียบค่าระหว่างประชากรเดิมกับเวกเตอร์ทดลองที่สร้างขึ้น

ขั้นตอนที่ 10 จะมีการสุ่มการเปลี่ยนแปลงค่าพารามิเตอร์ F และ CR แต่ละตัวเลือก คือใช้ความน่าจะเป็นช่วยในการตัดสินใจเปลี่ยนแปลงค่าพารามิเตอร์ โดยโอกาสที่จะมีการเปลี่ยนเท่ากับ 10%

ขั้นตอนที่ 11 นำประชากรที่ทำการคัดเลือกแล้วเป็นประชากรในรอบการคำนวณถัดไป

3.6 ตารางการดำเนินงาน

ตารางที่ 3.1 แสดงการดำเนินงาน

ขั้นตอน	ระยะเวลาการดำเนินงาน (เดือน)								
	มิ.ย. 2556	ก.ค. 2556	ส.ค. 2556	ก.ย. 2556	ต.ค. 2556	พ.ย. 2556	ธ.ค. 2556	ม.ค. 2557	ก.พ. 2557
นำเสนอหัวข้องานวิจัย									
ศึกษาทฤษฎีและเก็บรวบรวมข้อมูลที่เกี่ยวข้อง									
วิเคราะห์ข้อมูลที่เกี่ยวข้อง									
ดำเนินการออกแบบและสร้างโปรแกรม									
ทำทดสอบและปรับปรุงโปรแกรม									
วิเคราะห์และสรุปผลการวิจัย									
จัดทำรูปเล่มและทำการตรวจสอบ									

บทที่ 4
ผลการทดลอง

ในการทดลองนั้น เพื่อที่จะทำการทดสอบการคำนวณปัญหาการจัดสรรกำลังการผลิต โปรแกรมที่นำมาใช้ในการทดสอบเพื่อหาผลของการทดลองคือ โปรแกรม MATLAB R2012a ซึ่งทำการประมวลผลบน CPU Intel(R) Core(TM) i5-3210M (2.50 GHz, 4.00GB of RAM) โดยได้ทำการทดสอบกับข้อมูลของเครื่องกำเนิดไฟฟ้าสำหรับระบบที่มีเครื่องกำเนิดไฟฟ้าจำนวน 10 เครื่อง ที่มีการพิจารณาการเลือกใช้เชื้อเพลิงได้หลายชนิด มีการพิจารณาผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในสายส่ง ซึ่งในการทดสอบนั้นเราจะทำการทดสอบอยู่ 2 กรณี คือ

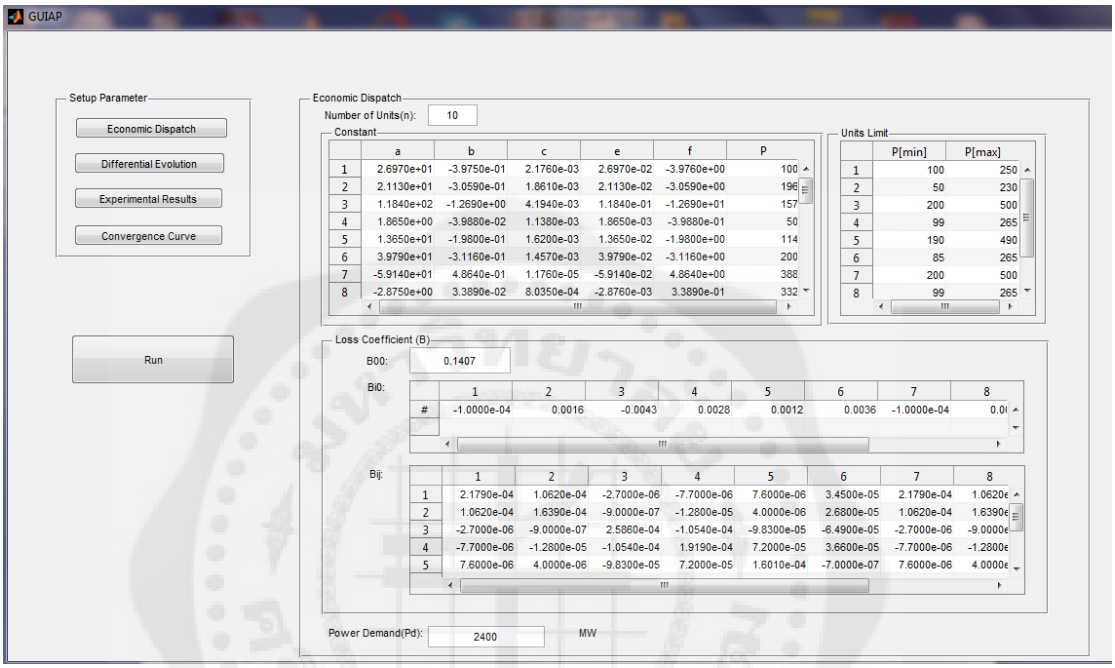
กรณีที่ 1 ใช้เครื่องกำเนิดไฟฟ้าจำนวน 10 เครื่อง จำนวนรอบในการคำนวณ 1,000 รอบ มีการพิจารณาการเลือกใช้เชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่งกำลังไฟฟ้า โดยค่ากำลังไฟฟ้าที่นำมาทดสอบคือ 2,400, 2,500, 2,600 และ 2,700 MW

กรณีที่ 2 ใช้เครื่องกำเนิดไฟฟ้าจำนวน 10 เครื่อง จำนวนรอบในการคำนวณ 1,000 รอบ มีการพิจารณาการเลือกใช้เชื้อเพลิงได้หลายชนิด คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่งกำลังไฟฟ้า โดยค่ากำลังไฟฟ้าที่นำมาทดสอบคือ 2,400, 2,500, 2,600 และ 2,700 MW

ซึ่งข้อมูลของระบบทดสอบที่ใช้ในงานวิจัยนี้ ได้ระบุไว้ในภาคผนวก

4.1 รูปแบบของเมนูหลักของโปรแกรมการจัดสรรกำลังการผลิตเพื่อการจ่ายโหลดอย่างประหยัดด้วยวิธีดิฟเฟอร์เรนเชียวอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้

เมื่อทำการรันโปรแกรม จะปรากฏดังรูปที่ 4.1- 4.4



รูปที่ 4.1 หน้าจอขณะทำการรันโปรแกรม

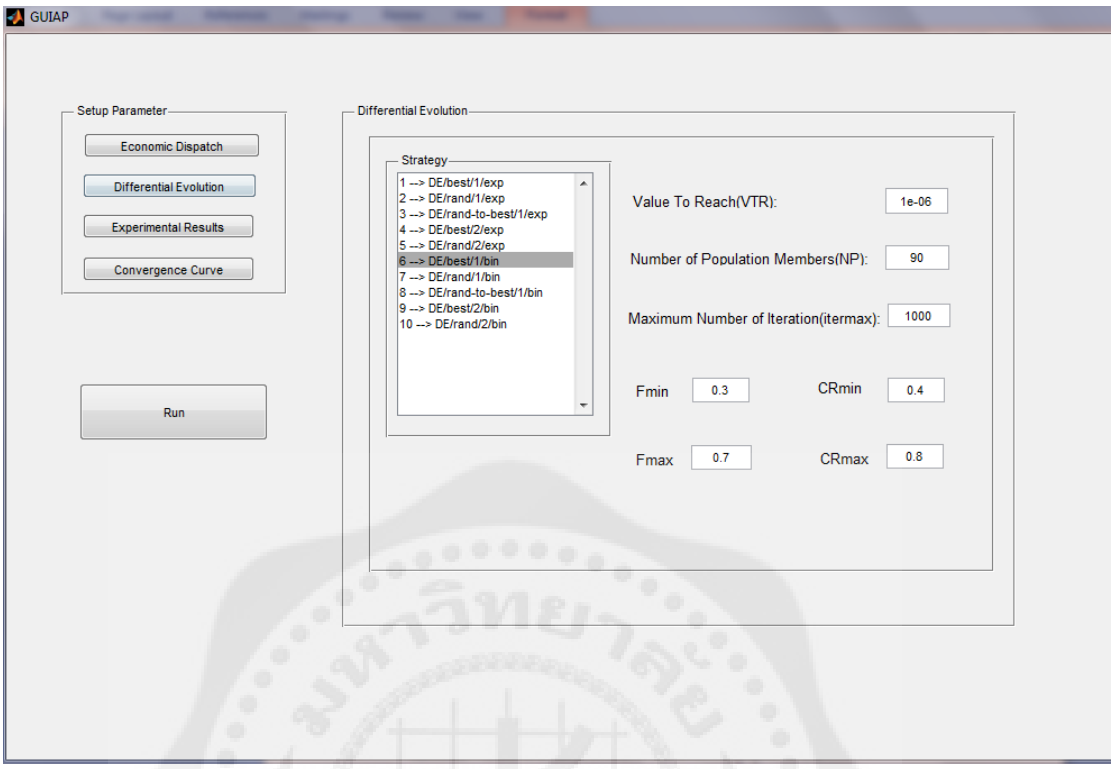
Number of Units (n): ใส่จำนวนของเครื่องกำเนิดไฟฟ้า

Units Limit: กำลังไฟฟ้าสูงสุดต่ำสุดของแต่ละเครื่องกำเนิดไฟฟ้า

Constant: ค่าตัวแปรของแต่ละเครื่องกำเนิดไฟฟ้า, ค่าสัมประสิทธิ์ราคาเชื้อเพลิง

Loss Coefficient (B): ค่าเวกเตอร์สัมประสิทธิ์ความสูญเสียของระบบ

Power Demand (Pd): ใส่กำลังไฟฟ้าที่ต้องการ



รูปที่ 4.2 หน้าจอที่ 2 ขณะทำการรันโปรแกรม

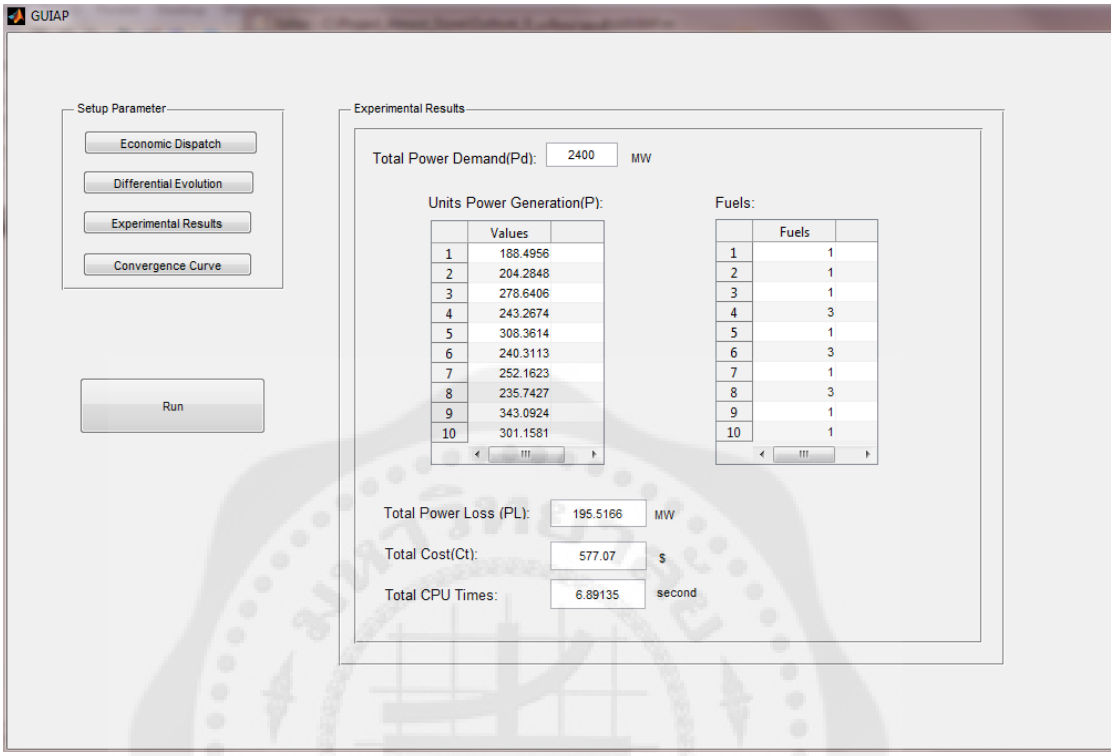
Value To Reach (VTR): ใส่ค่าความแตกต่างที่ใช้หยุดการคำนวณ

Number of Population Members (*NP*): ใส่จำนวนประชากร

Maximum Number of Iteration (itermax): ใส่จำนวนรอบสูงสุดที่กำหนด

F_{min}, F_{max}, CR_{min}, CR_{max}: ใส่ค่าพารามิเตอร์ F_{min}, F_{max}, CR_{min}, CR_{max}

Strategy: เลือกรูปแบบการคำนวณของ DEA



รูปที่ 4.3 หน้าจอโปรแกรมหลังจากทำการป้อนค่าต่างๆแล้วทำการกดปุ่ม RUN

Total Power Demand (*Pd*): แสดงค่าความต้องการไฟฟ้าที่ได้

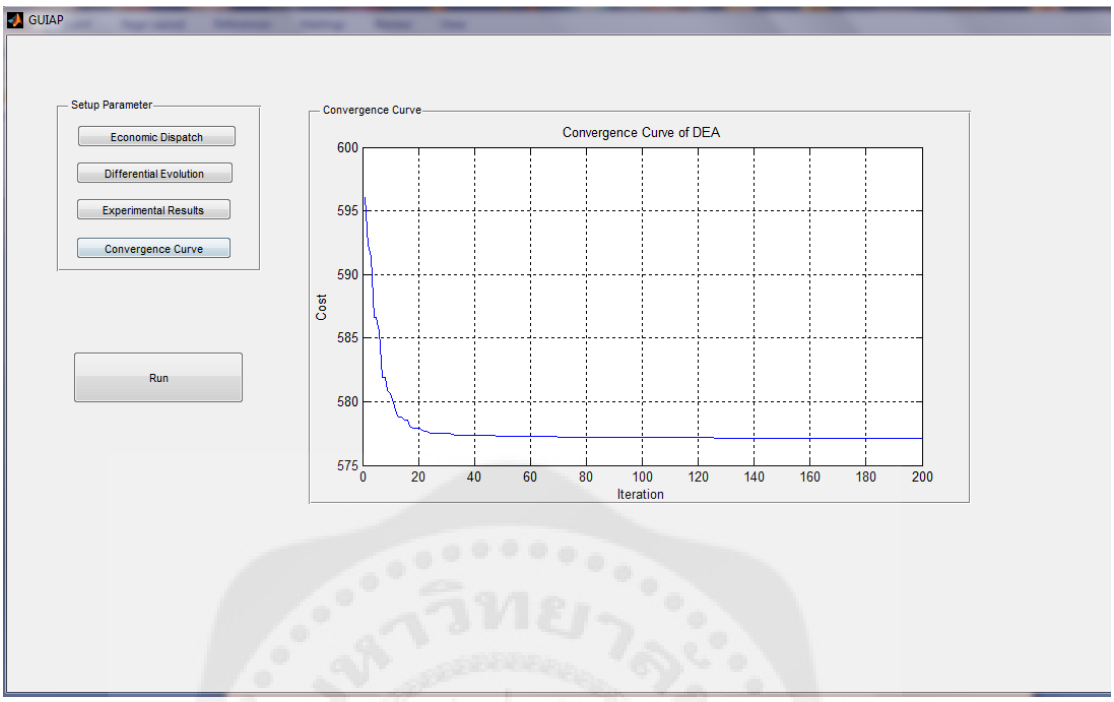
Units Power Generation (*P*): แสดงค่ากำลังไฟฟ้าที่แต่ละเครื่องกำเนิดไฟฟ้าจ่ายกำลังไฟฟ้า

Fuels: แสดงชนิดของเชื้อเพลิงที่แต่ละเครื่องกำเนิดไฟฟ้าเลือกใช้

Total Cost (*Ct*): แสดงค่าใช้จ่ายทั้งหมด

Total Power Loss (*Pl*): แสดงค่าความสูญเสียที่เกิดขึ้นในระบบสายส่งกำลังไฟฟ้า

Total CPU Time (sec): แสดงเวลาในการคำนวณโปรแกรมของคอมพิวเตอร์มีหน่วยเป็นวินาที



รูปที่ 4.4 หน้าจอแสดงผลการทดลองเป็นกราฟระหว่าง Cost กับ Iteration

4.2 ผลการทดลอง

เมื่อทำการทดลองการคำนวณ โดยใช้ค่าตามข้อมูลระบบมาพิจารณา ซึ่งจากผลการทดลองนี้จะแสดงให้เห็นถึงผลการทดลองทั้ง 2 กรณีที่ได้ทำการทดลอง เพื่อทำการทดสอบการจัดสรรกำลังการผลิตเพื่อจ่ายโหลดอย่างประหยัดด้วยวิธี Differential Evolution Algorithm

กรณีที่ 1

ในกรณีที่ 1 นั้นจะเป็นการทำการทดสอบการแก้ปัญหาการจัดสรรกำลังการผลิตเพื่อการจ่ายโหลดอย่างประหยัดโดยเครื่องกำเนิดไฟฟ้า สามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง โดยใช้ระบบทดสอบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง และในการทดสอบได้ทำการกำหนดค่าพารามิเตอร์ต่างๆ เอาไว้ดังต่อไปนี้ จำนวนประชากรเท่ากับ 90 จำนวนรอบสูงสุดในการคำนวณเท่ากับ 1,000 รอบ ค่าความแตกต่างที่ใช้หยุดการคำนวณเท่ากับ $1.e^{-6}$ เลือกใช้รูปแบบการคำนวณของ DEA คือ Strategy 6 ส่วนค่า F_{min}, F_{max}, CR_{min}, CR_{max} ที่เลือกใช้สามารถปรับได้ตั้งแต่ 0.3 - 0.7 และ 0.4 - 0.8 ตามลำดับ เนื่องจากเป็นค่าที่เหมาะสม และให้ค่าการคำนวณที่ดีที่สุด โดยที่ค่ากำลังไฟฟ้าที่นำมาทดสอบกับระบบคือ 2,400,

2,500, 2,600 และ2,700 MW ซึ่งได้ทำการคำนวณทั้งหมด 30 รอบ ได้ผลออกมาดังตารางที่ 4.1 และตารางที่ 4.2 ดังนี้

ตารางที่ 4.1 ผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง

เครื่องกำเนิดที่	2,400MW		2,500MW		2,600MW		2,700MW	
	F	GEN	F	GEN	F	GEN	F	GEN
1	1	188.2659	1	190.7357	2	203.7381	2	201.6088
2	1	205.1055	1	206.6495	1	209.5843	1	208.2549
3	1	279.9205	1	278.3192	1	287.8887	1	286.4254
4	3	244.2451	3	246.6632	3	249.6468	3	244.8352
5	1	307.3195	1	319.9218	2	359.0352	1	319.3988
6	3	240.6379	3	242.5580	3	245.3677	3	243.0113
7	1	250.3764	1	255.2311	1	264.1885	1	260.6022
8	3	234.9979	3	236.0889	3	238.1629	3	237.2225
9	1	344.7839	3	422.8003	3	440.0000	3	440.0000
10	1	299.7139	1	312.7370	1	328.8054	2	490.0000
TP(MW)	2,400		2,500		2,600		2,700	
PL(W)	195.3665		211.7047		226.4175		231.3592	
Ct($)	577.0370		636.1680		698.7090		763.7040	
Time(s)	5.98682		6.23785		6.13471		6.65959	

ตารางที่ 4.2 สรุปผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่องที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง

P_d (MW)	Average CPU Time (s)	Best CPU Time (s)	Average Cost($)	Best Cost($)	Worst Cost($)
2,400	6.12179	5.86595	577.056	577.037	577.609
2,500	6.21559	6.19334	636.168	636.168	636.168
2,600	6.17741	6.12895	698.709	698.709	698.709
2,700	6.50763	6.35567	763.704	763.704	763.704

จากสมการที่ (2.10) สามารถคำนวณค่าความสูญเสียได้ดังนี้

$$P_{Loss} = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} P_i B_{ij} P_j + \sum_{i=1}^{n_g} B_{io} P_i + B_{00} \text{ MW}$$

โดยที่ค่าพารามิเตอร์ต่างๆ มีค่าดังนี้

B_{00} = 0.1407

B_{i0} = [-0.0001 0.0016 -0.0043 0.0028 0.0012 0.0036 -0.0001 0.0016 -0.0043 0.0028]

B_{ij} =

$$0.001\times\begin{bmatrix}
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919 \\
0.0076 & 0.0040 & -0.0983 & 0.0720 & 0.1601 & -0.0007 & 0.0076 & 0.0040 & -0.0983 & 0.0720 \\
0.0345 & 0.0268 & -0.0649 & 0.0366 & -0.0007 & 0.2568 & 0.0345 & 0.0268 & -0.0649 & 0.0366 \\
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919
\end{bmatrix}$$

จากตารางที่ 4.1 ที่ค่าความต้องการกำลังไฟฟ้า (P_d) เท่ากับ 2,400 MW

$$P_i=\begin{bmatrix}
188.2659 \\
205.1055 \\
279.9205 \\
244.2451 \\
307.3195 \\
240.6379 \\
250.3764 \\
234.9979 \\
344.7839 \\
299.7139
\end{bmatrix}$$

P_j=

[88.2659 205.1055 279.9205 244.2451 307.3195 240.6379 250.3764 234.9979 344.7839 299.7139]

$$\sum_{i=1}^{n_g} B_{io}P_i+B_{00}= 0.7322+0.1407 = 0.8729$$

$$\sum_{i=1}^{n_g}\sum_{j=1}^{n_g} P_iB_{ij}P_j = 194.4936 \text{ MW}$$

ดังนั้นจะได้

$$P_{Loss}=194.4936+0.7322+0.1407=195.3665 \text{ MW}$$

และจากสมการที่ (2.9)

$$\sum_{i=1}^{n_g} P_i= P_D+P_{Loss}$$

$$P_D= \sum_{i=1}^{n_g} P_i -P_{Loss}= 2{,}595.3665\text{-}195.3665=2{,}400 \quad \text{MW}$$

จากการคำนวณพบว่า ค่ากำลังไฟฟ้ารวมที่ผลิตเมื่อนำมาลบด้วยค่าความสูญเสียรวมที่คำนวณได้ จะมีค่าเท่ากับความต้องการกำลังไฟฟ้าในกรณีนั้นๆ

ตัวอย่างการคำนวณราคาต้นทุนรวมในกรณีที่ 1 จากสมการที่ (2.2), (2.4), (2.7) และ (2.8) ที่ค่าความต้องการกำลังไฟฟ้าเท่ากับ 2,400 MW

$F_1P_1 = (2.6970\text{x}10) + ((-3.9750\text{x}10^{-1})(188.2659)) + ((2.176\text{x}10^{-3})(188.2659)^2) = 29.2606$

$F_2P_2 = (1.184\text{x}10^{2}) + ((-1.269)(205.1055)) + ((4.194\text{x}10^{-3})(205.1055)^2) = 34.5554$

$F_3P_3 = (3.979\text{x}10) + ((-3.116\text{x}10^{-1})(279.9205)) + ((1.457\text{x}10^{-3})(279.9205)^2) = 66.7307$

$F_4P_4 = (2.668\text{x}10^{2}) + ((-2.338)(244.2451)) + ((5.935\text{x}10^{-3})(244.2451)^2) = 49.8114$

$F_5P_5 = (1.392\text{x}10) + ((-8.733\text{x}10^{-2})(307.3195)) + ((1.066\text{x}10^{-3})(307.3195)^2) = 87.7605$

$F_6P_6 = (2.688\text{x}10^{2}) + ((-2.338)(240.6379)) + ((5.935\text{x}10^{-3})(240.6379)^2) = 47.8643$

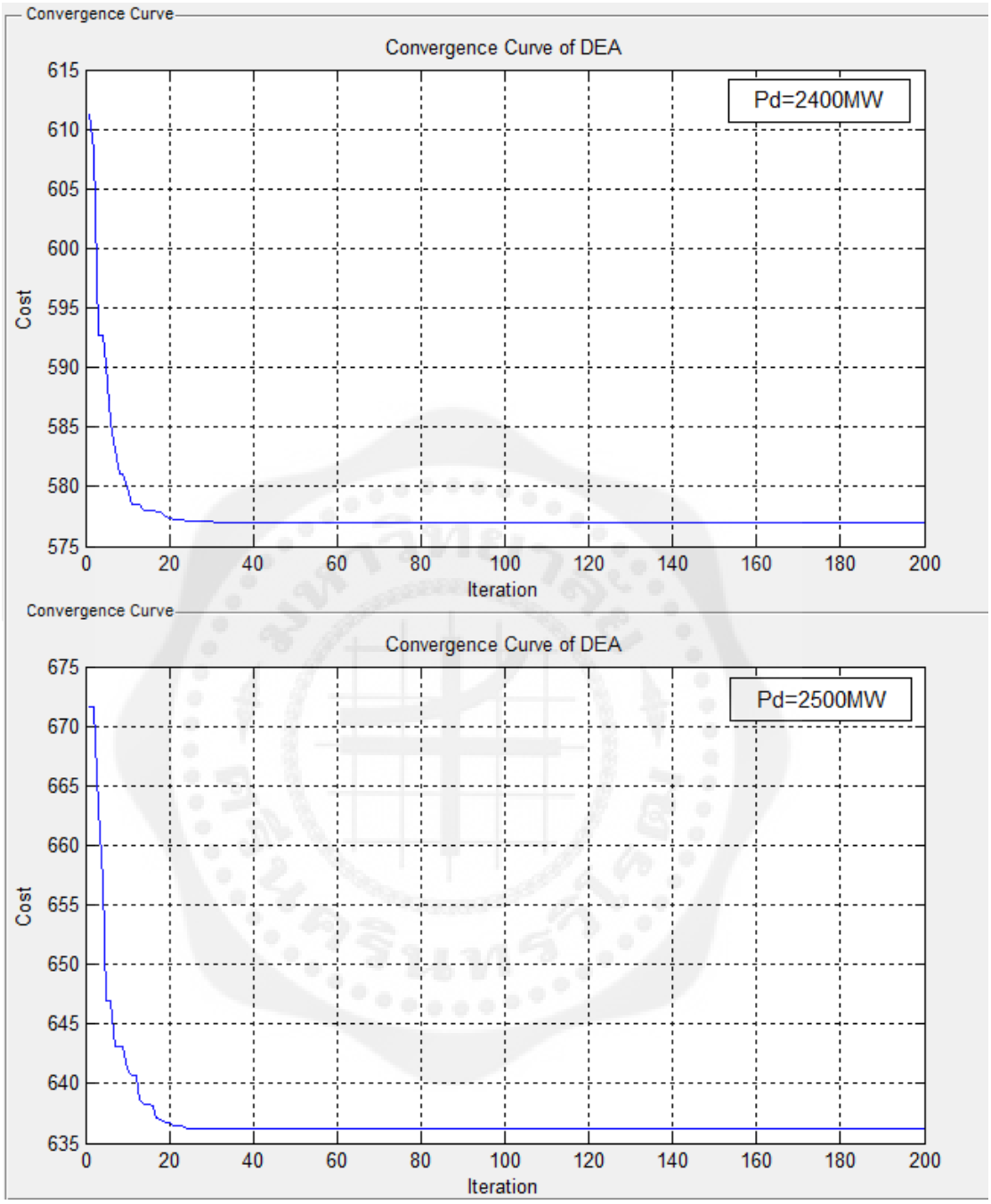
$F_7P_7 = (1.893\text{x}10) + ((-1.325\text{x}10^{-1})(250.3764)) + ((1.107\text{x}10^{-3})(250.3764)^2) = 55.1511$

$F_8P_8 = (2.668\text{x}10^{2}) + ((-2.338)(234.9979)) + ((5.935\text{x}10^{-3})(234.9979)^2) = 45.1294$

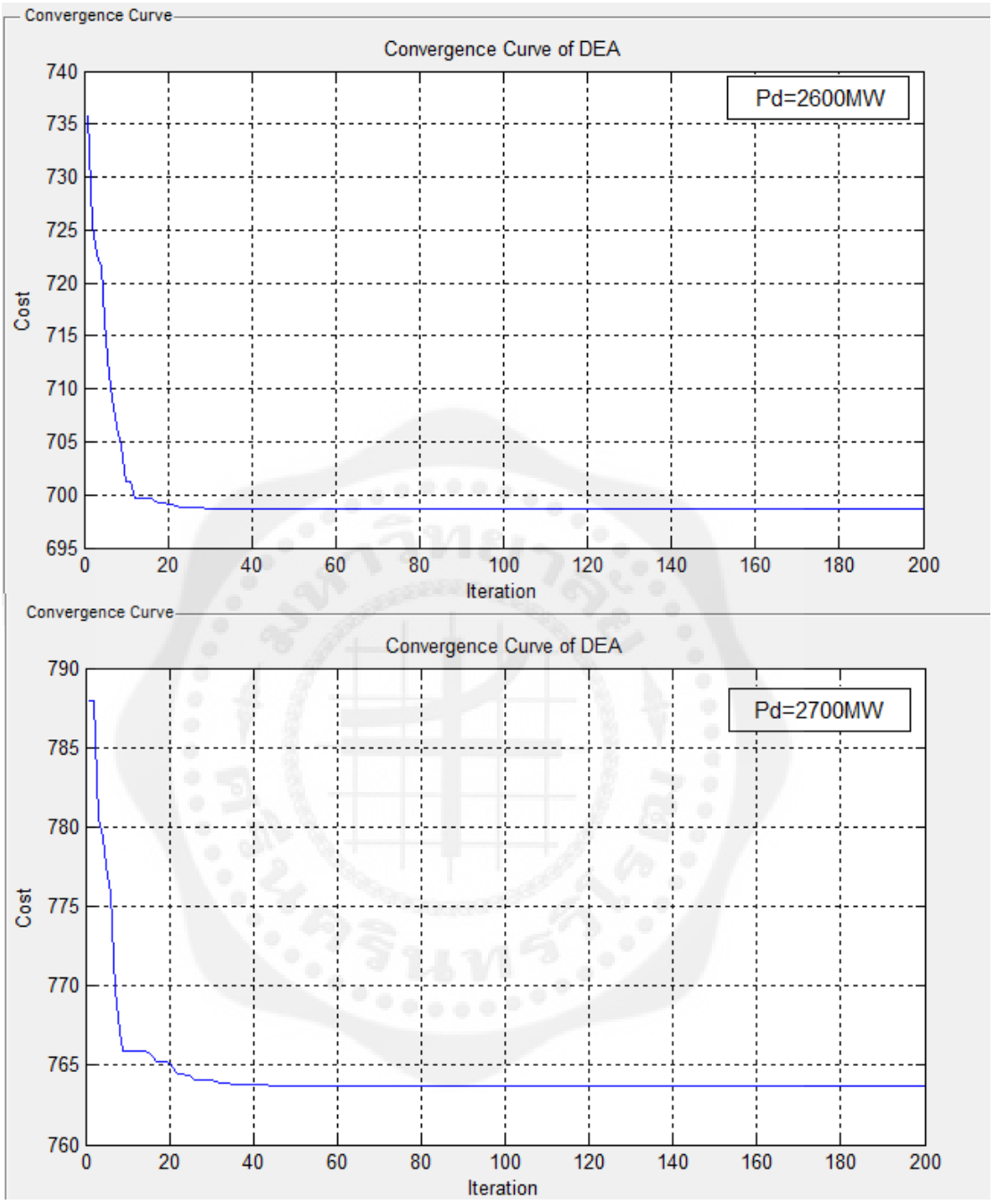
$F_9P_9 = (8.853\text{x}10) + ((-5.675\text{x}10^{-1})(344.7839)) + ((1.554\text{x}10^{-3})(344.7839)^2) = 77.5983$

$F_{10}P_{10} = (1.397\text{x}10) + ((-9.938\text{x}10^{-2})(299.7139)) + ((1.102\text{x}10^{-3})(299.7139)^2) = 83.1754$

$\therefore F_T = 577.0370$ \$/hr



รูปที่ 4.5 แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,400 และ 2,500 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้ งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียใน ระบบสายส่ง



รูปที่ 4.6 แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,600 และ2,700 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด ไม่คิดผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียใน ระบบสายส่ง

กรณีที่ 2

ในกรณีที่ 2 เป็นการทำการทดสอบการแก้ปัญหาการจัดสรรกำลังการผลิตเพื่อการจ่ายโหลดอย่างประหยัดโดยเครื่องกำเนิดไฟฟ้า สามารถเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง โดยใช้ระบบทดสอบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง และในการทดสอบนั้นได้ทำการกำหนดค่าพารามิเตอร์ต่างๆดังต่อไปนี้ จำนวนประชากรเท่ากับ 90 จำนวนรอบสูงสุดในการคำนวณเท่ากับ 1,000 รอบ ค่าความแตกต่างที่ใช้หยุดการคำนวณเท่ากับ $1.e^{-6}$ ได้เลือกใช้รูปแบบการคำนวณของ DEA คือ Strategy 6, F_{min}, F_{max}, CR_{min}, CR_{max} เลือกใช้ค่าเท่ากับ 0.3, 0.7, 0.4 และ 0.8 ตามลำดับ เนื่องจากเป็นค่าที่เหมาะสม และให้ค่าการคำนวณที่ดีที่สุด โดยที่ค่ากำลังไฟฟ้าที่นำมาทดสอบกับระบบคือ 2,400, 2,500, 2,600 และ2,700 MW ซึ่งได้ทำการคำนวณทั้งหมด 30 รอบ ได้ผลออกมาดังตารางที่ 4.3 และตารางที่ 4.4 ดังนี้

ตารางที่ 4.3 ผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในระบบสายส่ง

เครื่องกำเนิดที่	2,400MW		2,500MW		2,600MW		2,700MW	
	F	GEN	F	GEN	F	GEN	F	GEN
1	1	188.4956	1	191.6561	2	204.2160	2	201.1350
2	1	205.7702	1	207.0080	1	209.2361	1	207.9983
3	1	280.6571	1	277.6324	1	288.7228	1	285.6981
4	3	244.0737	3	246.2236	3	250.1204	3	245.0143
5	1	308.7136	1	319.5057	2	357.9141	1	319.5057
6	3	240.7144	3	243.7300	3	244.8799	3	243.8049
7	1	249.7913	1	254.5333	1	264.0174	1	261.6463
8	3	234.2646	3	236.2802	3	238.2957	3	237.7582
9	1	344.7531	3	422.3531	3	439.4512	3	439.0573
10	1	298.0931	1	313.8029	1	329.6089	2	489.8036
TP(MW)	2,400		2,500		2,600		2,700	
PL(W)	195.3266		211.7254		226.4623		231.4218	
Ct($)	577.0500		636.176		698.721		763.720	
Time(s)	6.90839		6.84247		6.82944		6.97394	

ตารางที่ 4.4 สรุปผลการคำนวณของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่อง ที่มีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด

P_d (MW)	Average CPU Time (s)	Best CPU Time (s)	Average Cost($)	Best Cost($)	Worst Cost($)
2,400	6.75652	6.67785	577.188	577.0500	577.660
2,500	6.74341	6.63372	636.190	636.176	636.205
2,600	6.76711	6.62899	698.733	698.721	698.751
2,700	6.74962	6.62679	763.736	763.720	773.256

จากสมการที่ (2.10) สามารถคำนวณค่าความสูญเสียได้ดังนี้

$$P_{Loss} = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} P_i B_{ij} P_j + \sum_{i=1}^{n_g} B_{io} P_i + B_{00} \text{ MW}$$

โดยที่ค่าพารามิเตอร์ต่างๆ มีค่าดังนี้

B_{00} = 0.1407

B_{i0} = [-0.0001 0.0016 -0.0043 0.0028 0.0012 0.0036 -0.0001 0.0016 -0.0043 0.0028]

B_{ij} =

$$0.001\times\begin{bmatrix}
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919 \\
0.0076 & 0.0040 & -0.0983 & 0.0720 & 0.1601 & -0.0007 & 0.0076 & 0.0040 & -0.0983 & 0.0720 \\
0.0345 & 0.0268 & -0.0649 & 0.0366 & -0.0007 & 0.2568 & 0.0345 & 0.0268 & -0.0649 & 0.0366 \\
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919
\end{bmatrix}$$

จากตารางที่ 4.3 ที่ค่าความต้องการกำลังไฟฟ้า (P_d) เท่ากับ 2,700 MW

$$P_i=\begin{bmatrix}
201.1350 \\
207.9983 \\
285.6981 \\
245.0143 \\
319.5057 \\
243.8049 \\
261.6463 \\
237.7582 \\
439.0573 \\
489.8036
\end{bmatrix}$$

P_j =

[201.1350 207.9983 285.6981 245.0143 319.5057 243.8049 261.6463 237.7582 439.0573 489.8036]

$$\sum_{i=1}^{n_g} B_{io}P_i+B_{00}= 0.8691+0.1407 = 1.0098 \text{ MW}$$

$$\sum_{i=1}^{n_g}\sum_{j=1}^{n_g} P_iB_{ij}P_j= 230.4120 \text{ MW}$$

ดังนั้นจะได้

$$P_{Loss}= 230.4120+0.8609+0.1407 = 231.4218 \text{ MW}$$

และจากสมการที่ (2.9)

$$\sum_{i=1}^{n_g} P_i= P_D+P_{Loss}$$

$$P_D= \sum_{i=1}^{n_g} P_i -P_{Loss}= 2{,}931.4218-231.4218 = 2{,}700 \text{ MW}$$

จากการคำนวณพบว่า ค่ากำลังไฟฟ้ารวมที่ผลิตเมื่อนำมาลบด้วยค่าความสูญเสียรวมที่คำนวณได้ จะมีค่าเท่ากับความต้องการไฟฟ้าในกรณีนั้นๆ

ตัวอย่างการคำนวณราคาต้นทุนรวมในกรณีที่ 2 จากสมการที่ (2.2), (2.8) และ (2.15) ที่ค่าความต้องการกำลังไฟฟ้าเท่ากับ 2,700 MW

$F_1P_1 = (2.113x10) + ((-3.059x10^{-1})(201.1350)) + ((1.861x10^{-3})(201.1350)^2) + (2.113x10^{-2})$
$\sin((-3.059)(196-201.1350)) = 34.8901$

$F_2P_2 = (1.184x10^2) + ((-1.269)(207.9983)) + ((4.194x10^{-3})(207.9983)^2) + (1.184x10^{-1})$
$\sin((-1.269x10)(157-207.9983)) = 35.8964$

$F_3P_3 = (3.979x10) + ((-3.116x10^{-1})(285.6981)) + ((1.457x10^{-3})(285.6981)^2) + (3.979x10^{-2})$
$\sin((-3.116)(200-285.6981)) = 69.6918$

$F_4P_4 = (2.668x10^2) + ((-2.338)(245.0143)) + ((5.935x10^{-3})(245.0143)^2) + (2.668x10^{-1})$
$\sin((-2.338x10)(200-245.0143)) = 50.2463$

$F_5P_5 = (1.392x10) + ((-8.733x10^{-2})(319.5057)) + ((1.066x10^{-3})(319.5057)^2) + (1.392x10^{-2})$
$\sin((-8.733x10^{-1})(190-319.5057)) = 94.8390$

$F_6P_6 = (2.668x10^2) + ((-2.338)(243.8049)) + ((5.935x10^{-3})(243.8049)^2) + (2.668x10^{-1})$
$\sin((-2.338x10)(200-243.8049)) = 49.5653$

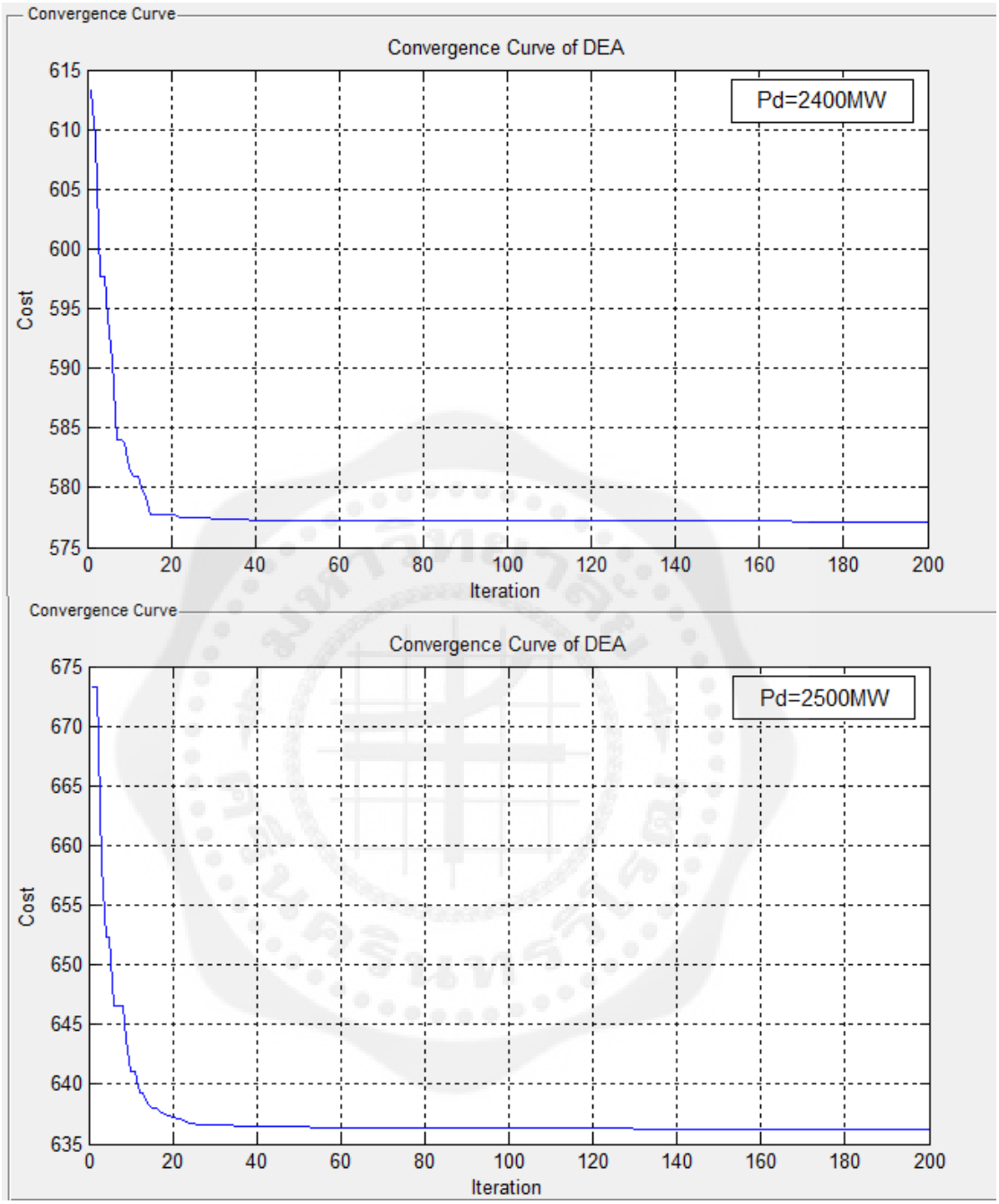
$F_7P_7 = (1.893x10) + ((-1.325x10^{-1})(261.6463)) + ((1.107x10^{-3})(261.6463)^2) + (1.893x10^{-2})$
$\sin((-1.325)(200-261.6463)) = 60.0457$

$F_8P_8 = (2.668x10^2) + ((-2.338)(237.7582)) + ((5.935x10^{-3})(237.7582)^2) + (2.668x10^{-1})$
$\sin((-2.338x10)(200-237.7582)) = 46.4209$

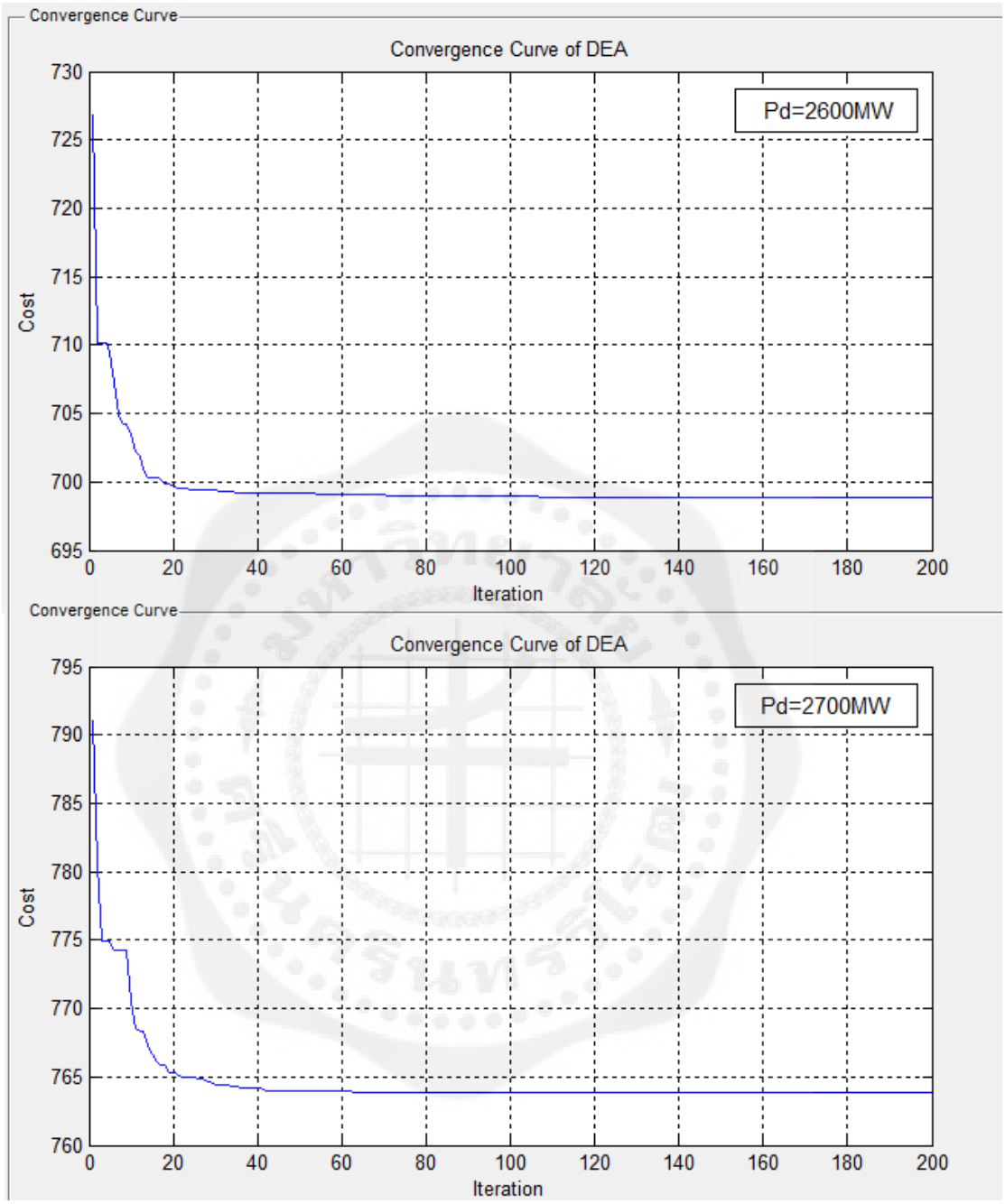
$F_9P_9 = (1.423x10) + ((-1.817x10^{-2})(439.0573)) + ((6.121x10^{-4})(439.0573)^2) + (1.423x10^{-2})$
$\sin((-1.817x10^{-1})(370-439.0573)) = 124.2474$

$F_{10}P_{10} = (-6.113x10) + ((5.084x10^{-1})(489.8036)) + ((4.164x10^{-5})(489.8036)^2) + (-6.113x10^{-2})$
$\sin((5.084)(407-489.8036)) = 197.8759$

$\therefore F_T = 763.720$ \$/hr



รูปที่ 4.7 แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,400 และ2,500 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในสายส่ง



รูปที่ 4.8 แสดงผลการลู่เข้าหาคำตอบที่ดีที่สุดของวิธี SaDEA ที่ P_d = 2,600 และ2,700 MW โดยที่ระบบมีเครื่องกำเนิดไฟฟ้า 10 เครื่อง สำหรับเครื่องกำเนิดไฟฟ้ามีการเลือกใช้งานเชื้อเพลิงได้หลายชนิด มีผลจากจุดวาล์ว และมีการพิจารณาการสูญเสียในสายส่ง

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

โครงงานนี้ได้นำหลักการของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ เข้ามาช่วยในการแก้ปัญหาการจัดสรรกำลังการผลิตเพื่อการจ่ายโหลดอย่างประหยัด โดยได้มีการทำการศึกษา 2 กรณีดังที่ได้กล่าวไปในข้างต้น ซึ่งจากการทดลองนั้น ในการออกแบบเราจะปรับตั้งค่าพารามิเตอร์ของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่น คือ F และ CR ให้สามารถปรับค่าเองได้ โดยค่าที่ปรับเองได้นั้น ได้กำหนดเอาไว้ในช่วง [0.3-0.8] และได้กำหนดรอบของการคำนวณไว้ที่ 1,000 รอบ

จากผลการทดสอบที่ได้ทั้งหมด จะเห็นได้ว่าราคาต้นทุนที่ทำการคำนวณ (Total Cost) และประสิทธิภาพของวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้นั้น ให้ค่าออกมาที่เหมาะสม ซึ่งสามารถสรุปได้ว่า วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้นั้น เป็นวิธีที่มีประสิทธิภาพสูงและใช้ระยะเวลาในการคำนวณต่ำ เหมาะในการนำมาแก้ปัญหาการจ่ายโหลดอย่างประหยัดโดยที่สามารถแก้ปัญหาดังกล่าวได้โดยไม่คำนึงถึงลักษณะรูปร่างของฟังก์ชันราคาที่ต้องการหาค่าที่เหมาะสมต่ำที่สุด

5.2 ข้อเสนอแนะ

ถ้าต้องการนำวิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้ ไปใช้สำหรับแก้ปัญหาแบบอื่นๆนั้น จะต้องกำหนดค่าพารามิเตอร์ทั้ง จำนวนประชากร และจำนวนรอบสูงสุดที่ใช้ในการคำนวณให้เหมาะสมกับปัญหา เพื่อให้ประสิทธิภาพในการค้นหาคำตอบดียิ่งขึ้นไป

ในอนาคตถ้ามีกรณีศึกษา (Case Study) แบบอื่นๆ เช่น อาจจะมีการเพิ่มจำนวนของเครื่องกำเนิดไฟฟ้าเป็น 20 เครื่อง หรือเป็น 100 เครื่อง ก็สามารถที่จะนำเอาวิธีนี้ไปประยุกต์ใช้ในการแก้ปัญหาการจ่ายโหลดอย่างประหยัดได้

เอกสารอ้างอิง

T. Sum-Im. “Self-Adaptive Differential Evolution Algorithm for Economic Dispatch with Transmission Losses Consideration,” Proc. IEEE Conference on TENCON 2010, pp. 90-95, 21st-24th Nov. 2010.

T. Sum-Im. A NOVEL DIFFERENTIAL EVOLUTION ALGORITHM APPROACH TO TRANSMISSION EXPANSION PLANNING.Ph.D.thesis, Brunel University, 2009.

R. Balamurugan, S. Subramanian, “Self-Adaptive Differential Evolution Based Power Economic Dispatch of Generators with Valve-Point Effects and Multiple Fuel Options.” International Journal of Computer Science and Engineering, vol. 1, no. 1.

Chao-Lung Chiang, “Improved Genetic Algorithm for Power Economic Dispatch of Units with Valve-Point Effects and Multiple Fuels.” IEEE Trans. No Power System, vol. 20, no. 4, pp. 1690-1699, 2005.

C.E. Lin and G.L. Viviani, “Hierarchical economic dispatch for piecewise quadratic cost function.” IEEE Trans. Power App. Syst., vol. PAS-103, no. 6, pp. 1170-1175, 1984

Vo Ngoc Dieu, W. Ongsakul, “Economic Dispatch with Multiple Fuel Types by Enhanced Augmented Lagrange Hopfield Network,” IEEE Trans. Power Apparatus and Systems, pp. 1-8, 2008.

R. Balamurugan, S. Subramanian, “Hybrid integer coded differential evolution dynamic programming approach for economic load dispatch with multiple fuels options.” Energy Convers Manage 2008; 49:608-14.

วิลาสินี ศึกษาการ. การจ่ายโหลดอย่างประหยัดที่มีฟังก์ชันราคาที่ไม่ราบเรียบโดยใช้วิธีทำให้เหมาะสมแบบกลุ่มอนุภาคร่วมกับวิธีโปรแกรมกำลังสองแบบลำดับ. วิทยานิพนธ์ปริญญามหาบัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์, 2551.

ชนะพงษ์ สุขกลึง และ อัษฎางค์ ศรีสุนทรศิริ. การจัดสรรกำลังผลิตโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้. ปริญญานิพนธ์ วิศวกรรมศาสตร์บัณฑิต สาขาวิชา ไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยศรีนครินทรวิโรฒ, 2554.

ปุณณรัตน์ สมศิริวัฒนา และ อรวรรณ กังวาฬ. การจ่ายโหลดอย่างประหยัดสำหรับหน่วยการผลิตไฟฟ้าแบบเลือกใช้งานเชื้อเพลิงได้หลายชนิด และมีผลจากจุดวาล์วโดยใช้วิธีดิฟเฟอร์เรนเชียลอิโวลูชั่นอัลกอริธึมแบบปรับตัวเองได้. ปริญญานิพนธ์ วิศวกรรมศาสตร์บัณฑิต สาขาวิชา ไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยศรีนครินทรวิโรฒ, 2555.

บัณฑิต เอื้ออาภรณ์. 2004. **การวิเคราะห์ระบบไฟฟ้ากำลังเบื้องต้น**. สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ.

พิชัย อารีย์. 2552. **การวิเคราะห์ระบบไฟฟ้ากำลัง**. สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ.

ปริญญา สงวนสัตย์. 2010. **คู่มือ MATLAB ฉบับสมบูรณ์**. ไอดีซี พรีเมียร์, นนทบุรี

มนัส สังวรศิลป์ และ วรรัตน์ ภัทรอมรกุล. 2000. **คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์**. อินโฟเกรส, กรุงเทพฯ.

วิทยากร อัศดรวิเศษ และคณะ. 2555. **การประยุกต์ใช้ MATLB**. สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพฯ.

สถิติพลังงานของประเทศไทย 2555 (เบื้องต้น). (2555). กรมพัฒนาพลังงานทดแทนและอนุรักษ์พลังงานกระทรวงพลังงาน. จาก http://www.dede.go.th

แผนพัฒนาพลังงานทดแทนและพลังงานทางเลือก 25% ใน 10 ปี (พ.ศ.2555-2564). (). สำนักงานนโยบายและพลังงานกระทรวงพลังงาน. จาก http://www.eppo.go.th

การใช้ไฟฟ้าและการผลิตไฟฟ้าของประเทศไทย. (2554). สำนักงานนโยบายและแผนพลังงานกระทรวงพลังงาน. จาก http://www.eppo.go.th

สถานการณ์พลังงานไฟฟ้าของประเทศไทย. (2555). การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย (กฟผ.). จาก http://www.egat.co.th/

ข้อเสนอแผนพัฒนากำลังการผลิตไฟฟ้า 2555-2573 (แผนพีดีพี 2012) และกรอบเพื่อการพัฒนาความรับผิดชอบตรวจสอบได้ของการวางแผนภาคพลังงานไฟฟ้า. (2555). สำนักงานนโยบายและแผนพลังงานกระทรวงพลังงาน. จาก http://www.eppo.go.th

รายงานพลังงานทดแทนของประเทศไทย 2555. (2555). สำนักงานนโยบายและแผนพลังงานกระทรวงพลังงาน. จาก http://www.eppo.go.th

ภาคผนวก

ภาคผนวก ก.

ตารางภาคผนวก ก1 แสดงข้อมูลของเครื่องกำเนิดไฟฟ้าที่ใช้ในการทดสอบในกรณีที่ 1 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่องซึ่งมีฟังก์ชั่นราคาเชื่อเพลิงที่มีผลจากเครื่องกำเนิดไฟฟ้าสามารถเลือกใช้เชื้องเพลิงได้หลายชนิด

Units	Generation Min *P1* *P2* Max F1 F2 F3	F	COST COEFFICIENTS		
			A	*B*	*C*
1	100 196 250 0 1 2	1	.2697e2	-.3975e0	.2176e-2
		2	.2113e2	.3059e0	.1861e-2
2	50 114 157 230 2 3 1	1	.1184e3	-.1269e1	.4194e-2
		2	.1865e1	-.3988e-1	.1138e-2
		3	.1365e2	-.1980e0	.1620e-2
3	200 332 388 500 1 3 2	1	.3979e2	-.3116e0	.1457e-2
		2	-.5914e2	.4864e0	.1176e-4
		3	-.2876e1	.3389e-1	.8035e-3
4	99 138 200 265 1 2 3	1	.1983e1	-.3114e-1	.1049e-2
		2	.5285e2	-.6348e0	.2758e-2
		3	.2876e1	-.2338e1	.5935e-2
5	190 338 407 490 1 2 3	1	.1392e2	-.8753e-1	.1066e-2
		2	.9976e2	-.5206e0	.1597e-2
		3	-.5399e2	.4462e0	.1498e-3
6	85 138 200 265 2 1 3	1	.5285e2	-.6348e0	.2758e-2
		2	.1983e1	-.3114e-1	.1049e-2
		3	.2668e3	-.2338e1	.5935e-2
7	200 331 391 500 1 2 3	1	.1893e2	-.1325e0	.1107e-2
		2	.4377e2	-.2267e0	.1165e-2
		3	-.4335e2	.3559e0	.2454e-3

Units	Generation				F	COST COEFFICIENTS		
	Min	*P1*	*P2*	Max				
		F1	F2	F3		*A*	*B*	*C*
8	99	138	200	265	1	.1963e1	-.3114e-1	.1049e-2
					2	.5285e2	-.6348e0	.2756e-2
		1	2	3	3	.2668e3	-.2338e1	.5935e-2
9	130	213	370	440	1	.8853e2	-.5675e0	.1554e-2
					2	.1530e2	-4514e-1	.7033e-2
		1	2	3	3	.1423e2	-1817e-1	.6121e-3
10	200	362	407	490	1	.1397e2	-.9938e-1	.1102e-2
					2	-.6113e2	.5084e0	.4164e-4
		1	3	2	3	.4671e2	-.2024e0	.1137e-2

ตารางภาคผนวก ก2 แสดงข้อมูลของเครื่องกำเนิดไฟฟ้าที่ใช้ในการทดสอบในกรณีที่ 2 สำหรับระบบที่มีเครื่องกำเนิดไฟฟ้า 10 เครื่องซึ่งมีฟังก์ชั่นราคาเชื่อเพลิงที่มีผลจากเครื่องกำเนิดไฟฟ้าสามารถเลือกใช้เชื้องเพลิงได้หลายชนิด

Unit	Generation Min *P1* *P2* Max F1 F2 F3	Fuel Type	Cost coefficients				
			a_i	b_i	c_i	e_i	f_i
1	100 196 250 1 2	1	.2697e2	-.3975e0	.2176e-2	.2697e-1	-.3975e1
		2	.2113e2	.3059e0	.1861e-2	.2113e-1	-.3059e1
2	50 114 157 230 2 3 1	1	.1184e3	-.1269e1	.4194e-2	.1184e0	-.1269e2
		2	.1865e1	-.3988e-1	.1138e-2	.1865e-2	-.3988e0
		3	.1365e2	-.1980e0	.1620e-2	.1365e-1	-.1980e1
3	200 332 388 500 1 3 2	1	.3979e2	-.3116e0	.1457e-2	.3979e-1	-.3116e1
		2	-.5914e2	.4864e0	.1176e-4	-.5914e-1	.4864e1
		3	-.2876e1	.3389e-1	.8035e-3	-.2876e-2	.3389e0
4	99 138 200 265 1 2 3	1	.1983e1	-.3114e-1	.1049e-2	.1982e-2	-.3114e0
		2	.5285e2	-.6348e0	.2758e-2	.5285e-1	-.6348e1
		3	.2876e1	-.2338e1	.5935e-2	.2668e0	-.2338e2
5	190 338 407 490 1 2 3	1	.1392e2	-.8753e-1	.1066e-2	.1392e-1	-.8733e0
		2	.9976e2	-.5206e0	.1597e-2	.9976e-1	-.5206e1
		3	-.5399e2	.4462e0	.1498e-3	-.5399e-1	.4462e1
6	85 138 200 265 2 1 3	1	.5285e2	-.6348e0	.2758e-2	.5285e-1	-.66348e1
		2	.1983e1	-.3114e-1	.1049e-2	.1983e-2	-.3114e0
		3	.2668e3	-.2338e1	.5935e-2	.2668e0	-.2338e2
7	200 331 391 500 1 2 3	1	.1893e2	-.1325e0	.1107e-2	.1893e-1	-.1325e1
		2	.4377e2	-.2267e0	.1165e-2	.4377e-1	-.2267e1
		3	-.4335e2	.3559e0	.2454e-3	-.4335e-1	.3558e1

Unit	Generation Min *P1* *P2* Max F1 F2 F3	Fuel Type	Cost coefficients				
			a_i	b_i	c_i	e_i	f_i
8	99 138 200 265 1 2 3	1	.1963e1	-.3114e-1	.1049e-2	.1983e-2	-.3114e0
		2	.5285e2	-.6348e0	.2756e-2	.5285e-1	-.6348e1
		3	.2668e3	-.2338e1	.5935e-2	.2668e0	-.2338e2
9	130 213 370 440 1 2 3	1	.8853e2	-.5675e0	.1554e-2	.8853e-1	-.5675e1
		2	.1530e2	-4514e-1	.7033e-2	.1423e-1	-.1817e0
		3	.1423e2	-1817e-1	.6121e-3	.1423e-1	-.1817e0
10	200 362 407 490 1 3 2	1	.1397e2	-.9938e-1	.1102e-2	.1397e-1	-.9938e0
		2	-.6113e2	.5084e0	.4164e-4	-.6113e-1	.5084e1
		3	.4671e2	-.2024e0	.1137e-2	.4671e-1	-.2024e1

จากสมการที่ (2.10) จะสามารถหาค่ากำลังสูญเสียได้จาก

$$P_{Loss} = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} P_i B_{ij} P_j + \sum_{i=1}^{n_g} B_{io} P_i + B_{00} \text{ MW}$$

โดยที่ค่าพารามิเตอร์ต่างๆ ที่ใช้สำหรับคำนวณในเครื่องกำเนิดไฟฟ้า 10 เครื่องมีค่าดังนี้

B_{00} = 0.1407

B_{i0} = [-0.0001 0.0016 -0.0043 0.0028 0.0012 0.0036 -0.0001 0.0016 -0.0043 0.0028]

$$B_{ij} = 0.001 \times \begin{bmatrix}
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919 \\
0.0076 & 0.0040 & -0.0983 & 0.0720 & 0.1601 & -0.0007 & 0.0076 & 0.0040 & -0.0983 & 0.0720 \\
0.0345 & 0.0268 & -0.0649 & 0.0366 & -0.0007 & 0.2568 & 0.0345 & 0.0268 & -0.0649 & 0.0366 \\
0.2179 & 0.1062 & -0.0027 & -0.0077 & 0.0076 & 0.0345 & 0.2179 & 0.1062 & -0.0027 & -0.0077 \\
0.1062 & 0.1639 & -0.0009 & -0.0128 & 0.0040 & 0.0268 & 0.1062 & 0.1639 & -0.0009 & -0.0128 \\
-0.0027 & -0.0009 & 0.2586 & -0.1054 & -0.0983 & -0.0649 & -0.0027 & -0.0009 & 0.2586 & -0.1054 \\
-0.0077 & -0.0128 & -0.1054 & 0.1919 & 0.0720 & 0.0366 & -0.0077 & -0.0128 & -0.1054 & 0.1919
\end{bmatrix}$$

ภาคผนวก ข.
โปรแกรมที่ใช้ในการทดลอง

GUI.m

```
function varargout = GUIAP(varargin)
% GUIAP MATLAB code for GUIAP.fig
%      GUIAP, by itself, creates a new GUIAP or raises the existing
%      singleton*.
%
%      H = GUIAP returns the handle to a new GUIAP or the handle to
%      the existing singleton*.
%
%      GUIAP('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUIAP.M with the given input arguments.
%
%      GUIAP('Property','Value',...) creates a new GUIAP or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUIAP_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUIAP_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUIAP

% Last Modified by GUIDE v2.5 24-Nov-2013 17:42:47
```

```
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUIAP_OpeningFcn, ...
                   'gui_OutputFcn',  @GUIAP_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before GUIAP is made visible.
function GUIAP_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUIAP (see VARARGIN)

% Choose default command line output for GUIAP
handles.output = hObject;
```

```
% Update handles structure
initListstg={'1 --> DE/best/1/exp'
   '2 --> DE/rand/1/exp'
   '3 --> DE/rand-to-best/1/exp'
   '4 --> DE/best/2/exp'
   '5 --> DE/rand/2/exp'
   '6 --> DE/best/1/bin'
   '7 --> DE/rand/1/bin'
   '8 --> DE/rand-to-best/1/bin'
   '9 --> DE/best/2/bin'
   '10 --> DE/rand/2/bin'};
initStrategy=6;
initxlimit = [100 250;50 230;200 500;99 265;190 490;85 265;200 500;99 265;130 440;200 490];

initconstant = [.2697e+002    -.3975e0    .2176e-002  .2697e-001  -.3976e+001 100
          .2113e+002    -.3059e0    .1861e-002  .2113e-001  -.3059e+001 196
          .1184e+003    -.1269e+001 .4194e-002  .1184e0     -.1269e+002 157
          .1865e+001    -.3988e-001 .1138e-002  .1865e-002  -.3988e0    50
          .1365e+002    -.1980e0    .1620e-002  .1365e-001  -.1980e+001 114
          .3979e+002    -.3116e0    .1457e-002  .3979e-001  -.3116e+001 200
          -.5914e+002   .4864e0     .1176e-004  -.5914e-001 .4864e+001  388
          -.2875e+001   .3389e-001  .8035e-003  -.2876e-002 .3389e0     332
          .1983e+001    -.3114e-001 .1049e-002  .1983e-002  -.3114e0    99
          .5285e+002    -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
          .2668e+003    -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
          .1392e+002    -.8733e-001 .1066e-002  .1392e-001  -.8733e0    190
          .9976e+002    -.5206e0    .1597e-002  .9976e-001  -.5206e+001 338
          -.5399e+002   .4462e0     .1498e-003  -.5399e-001 .4462e+001  407
          .5285e+002    -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
```

```
.1983e+001   -.3114e-001 .1049e-002  .1983e-002  -.3114e0    85
.2668e+003   -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
.1893e+002   -.1325e0    .1107e-002  .1893e-001  -.1325e+001 200
.4377e+002   -.2267e0    .1165e-002  .4377e-001  -.2267e+001 331
-.4335e+002  .3559e0     .2454e-003  -.4335e-001 .3559e+001  391
.1983e+001   -.3114e-001 .1049e-002  .1983e-002  -.3114e0    99
.5285e+002   -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
.2668e+003   -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
.8853e+002   -.5675e0    .1554e-002  .8853e-001  -.5675e+001 213
.1423e+002   -.1817e-001 .6121e-003  .1423e-001  -.1817e0    130
.1423e+002   -.1817e-001 .6121e-003  .1423e-001  -.1817e0    370
.1397e+002   -.9938e-001 .1102e-002  .1397e-001  -.9938e0    200
-.6113e+002  .5084e0     .4164e-004  -.6113e-001 .5084e+001  407
.4671e+002   -.2024e0    .1137e-002  -.4671e-001 -.2024e+001 362];
```

```
initB=(10^-3)*[0.2179 0.1062 -0.0027 -0.0077 0.0076 0.0345 0.2179 0.1062 -0.0027 -0.0077
   0.1062 0.1639 -0.0009 -0.0128 0.0040 0.0268 0.1062 0.1639 -0.0009 -0.0128
   -0.0027 -0.0009 0.2586 -0.1054 -0.0983 -0.0649 -0.0027 -0.0009 0.2586 -0.1054
   -0.0077 -0.0128 -0.1054 0.1919 0.0720 0.0366 -0.0077 -0.0128 -0.1054 0.1919
   0.0076 0.0040 -0.0983 0.0720 0.1601 -0.0007 0.0076 0.0040 -0.0983 0.0720
   0.0345 0.0268 -0.0649 0.0366 -0.0007 0.2568 0.0345 0.0268 -0.0649 0.0366
   0.2179 0.1062 -0.0027 -0.0077 0.0076 0.0345 0.2179 0.1062 -0.0027 -0.0077
   0.1062 0.1639 -0.0009 -0.0128 0.0040 0.0268 0.1062 0.1639 -0.0009 -0.0128
   -0.0027 -0.0009 0.2586 -0.1054 -0.0983 -0.0649 -0.0027 -0.0009 0.2586 -0.1054
   -0.0077 -0.0128 -0.1054 0.1919 0.0720 0.0366 -0.0077 -0.0128 -0.1054 0.1919];
initB00=0.1407;
initBi0=[-0.0001 0.0016 -0.0043 0.0028 0.0012 0.0036 -0.0001 0.0016 -0.0043 0.0028];
initVTR=1.e-6;
initItermax=str2double(get(handles.vaitermax,'string'));
```

```
initNP=str2double(get(handles.vaNP,'string'));
initFmin=str2double(get(handles.vafmin,'string'));
initFmax=str2double(get(handles.vafmax,'string'));
initCRmin=str2double(get(handles.vacrmin,'string'));
initCRmax=str2double(get(handles.vacrmax,'string'));
%Set to display
set(handles.tableconst,'Data',initconstant);
set(handles.tablelimit,'Data',initxlimit);
set(handles.listst,'String',initListstg);
set(handles.listst,'Value',6);
set(handles.vaVTR,'String',initVTR);
set(handles.txtB00,'String',initB00);
set(handles.tbBi0,'Data',initBi0);
set(handles.tbBij,'Data',initB);
set(handles.tableP,'Data',cell(10,1));
set(handles.tablefuels,'Data',cell(10,1));

guidata(hObject, handles);

% UIWAIT makes GUIAP wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = GUIAP_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
```

```
varargout{1} = handles.output;
function van_Callback(hObject, eventdata, handles)
% hObject    handle to van (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of van as text
%        str2double(get(hObject,'String')) returns contents of van as a double


% --- Executes during object creation, after setting all properties.
function van_CreateFcn(hObject, eventdata, handles)
% hObject    handle to van (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function vapd_Callback(hObject, eventdata, handles)
% hObject    handle to vapd (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vapd as text
```

```
%       str2double(get(hObject,'String')) returns contents of vapd as a double
% --- Executes during object creation, after setting all properties.
function vapd_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vapd (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btED.
function btED_Callback(hObject, eventdata, handles)
% hObject    handle to btED (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.bED,'Visible','on');
set(handles.bDE,'Visible','off');
set(handles.bER,'Visible','off');
set(handles.bcurve,'Visible','off');


% --- Executes on button press in btDE.
function btDE_Callback(hObject, eventdata, handles)
% hObject    handle to btDE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.bED,'Visible','off');
```

```
set(handles.bDE,'Visible','on');
set(handles.bER,'Visible','off');
set(handles.bcurve,'Visible','off');


% --- Executes on button press in btresult.
function btresult_Callback(hObject, eventdata, handles)
% hObject    handle to btresult (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.bED,'Visible','off');
set(handles.bDE,'Visible','off');
set(handles.bER,'Visible','on');
set(handles.bcurve,'Visible','off');



function vaVTR_Callback(hObject, eventdata, handles)
% hObject    handle to vaVTR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vaVTR as text
%        str2double(get(hObject,'String')) returns contents of vaVTR as a double


% --- Executes during object creation, after setting all properties.
function vaVTR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vaVTR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function vaNP_Callback(hObject, eventdata, handles)
% hObject    handle to vaNP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vaNP as text
%        str2double(get(hObject,'String')) returns contents of vaNP as a double



% --- Executes during object creation, after setting all properties.
function vaNP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vaNP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
function vaitermax_Callback(hObject, eventdata, handles)
% hObject    handle to vaitermax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vaitermax as text
%        str2double(get(hObject,'String')) returns contents of vaitermax as a double


% --- Executes during object creation, after setting all properties.
function vaitermax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vaitermax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function vatpd_Callback(hObject, eventdata, handles)
% hObject    handle to vatpd (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vatpd as text
```

```
%        str2double(get(hObject,'String')) returns contents of vatpd as a double
% --- Executes during object creation, after setting all properties.
function vatpd_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vatpd (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function vact_Callback(hObject, eventdata, handles)
% hObject    handle to vact (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vact as text
%        str2double(get(hObject,'String')) returns contents of vact as a double


% --- Executes during object creation, after setting all properties.
function vact_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vact (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function vacpu_Callback(hObject, eventdata, handles)
% hObject    handle to vacpu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vacpu as text
%        str2double(get(hObject,'String')) returns contents of vacpu as a double


% --- Executes during object creation, after setting all properties.
function vacpu_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vacpu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on selection change in listst.
function listst_Callback(hObject, eventdata, handles)
% hObject    handle to listst (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listst contents as cell array
%        contents{get(hObject,'Value')} returns selected item from listst



% --- Executes during object creation, after setting all properties.
function listst_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listst (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in btRun.
function btRun_Callback(hObject, eventdata, handles)
% hObject    handle to btRun (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
handles.output=hObject;
disp('Waiting!!!!!');
h=waitbar(1,'Please wait...');
for i=1:100,%computation here%

if i==70
global Pd xlimit constant VTR n NP itermax Fmin Fmax CRmin CRmax strategy B B00 Bi0;
n=str2double(get(handles.van,'String'));
xlimit=get(handles.tablelimit,'Data');
constant=get(handles.tableconst,'Data');
Pd=str2double(get(handles.vapd,'String'));
VTR=str2double(get(handles.vaVTR,'String'));
itermax=str2double(get(handles.vaitermax,'String'));
NP=str2double(get(handles.vaNP,'String'));
Fmin=str2double(get(handles.vafmin,'String'));
Fmax=str2double(get(handles.vafmax,'String'));
CRmin=str2double(get(handles.vacrmin,'String'));
CRmax=str2double(get(handles.vacrmax,'String'));
B00= str2num(get(handles.txtB00,'String'));
Bi0=get(handles.tbBi0,'Data');
B=get(handles.tbBij,'Data');

strategy=get(handles.liststr,'Value');

[mincost Power_Of_Gen_2 Fuel Power_Demand CPUTime PL]=run();
end
end
close(h)
initPd=str2double(get(handles.vapd,'string'));
initn=str2double(get(handles.van,'string'));
```

```
set(handles.vatpd,'String',num2str(Power_Demand));
if Pd>3200
errordlg('Error !!! เกิดข้อผิดพลาดโปรแกรมไม่สามารถคำนวณได้เนื่องจากค่า Power Demand
มากกว่ากำหนด (3200 MW)');
Power_Demand = Pd;
else
set(handles.tableP,'Data',Power_Of_Gen_2');
set(handles.textERPL, 'String', num2str(PL));
set(handles.tablefuels,'Data',Fuel);
set(handles.vact,'String',mincost);
set(handles.vacpu,'String',CPUTime);
set(handles.bED,'Visible','off');
set(handles.bDE,'Visible','off');
set(handles.bER,'Visible','on');
set(handles.bcurve,'Visible','off');
guidata(hObject,handles);
end

% --- Executes on button press in btClear.
function btClear_Callback(hObject, eventdata, handles)
% hObject    handle to btClear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clear all
clc


% --- Executes on button press in btcurve.
function btcurve_Callback(hObject, eventdata, handles)
% hObject    handle to btcurve (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.bED,'Visible','off');
set(handles.bDE,'Visible','off');
set(handles.bER,'Visible','off');
set(handles.bcurve,'Visible','on');

[mincost Power_Of_Gen_2 Fuel Power_Demand CPUTime]=run();
axes(handles.axes1);



function vafmin_Callback(hObject, eventdata, handles)
% hObject    handle to vafmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vafmin as text
%        str2double(get(hObject,'String')) returns contents of vafmin as a double



% --- Executes during object creation, after setting all properties.
function vafmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vafmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end
function vafmax_Callback(hObject, eventdata, handles)
% hObject    handle to vafmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vafmax as text
%        str2double(get(hObject,'String')) returns contents of vafmax as a double


% --- Executes during object creation, after setting all properties.
function vafmax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vafmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function vacrmin_Callback(hObject, eventdata, handles)
% hObject    handle to vacrmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vacrmin as text
```

```
%        str2double(get(hObject,'String')) returns contents of vacrmin as a double
% --- Executes during object creation, after setting all properties.
function vacrmin_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vacrmin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end



function vacrmax_Callback(hObject, eventdata, handles)
% hObject    handle to vacrmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of vacrmax as text
%        str2double(get(hObject,'String')) returns contents of vacrmax as a double


% --- Executes during object creation, after setting all properties.
function vacrmax_CreateFcn(hObject, eventdata, handles)
% hObject    handle to vacrmax (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function txtB00_Callback(hObject, eventdata, handles)
% hObject    handle to txtB00 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txtB00 as text
%        str2double(get(hObject,'String')) returns contents of txtB00 as a double



% --- Executes during object creation, after setting all properties.
function txtB00_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txtB00 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function textERPL_Callback(hObject, eventdata, handles)
% hObject    handle to textERPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of textERPL as text
%        str2double(get(hObject,'String')) returns contents of textERPL as a double


% --- Executes during object creation, after setting all properties.
function textERPL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to textERPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

run.m

```
function [mincost Power_Of_Gen_2 Fuel Power_Demand CPUTime PL]=run()
tic;
clear;
clc;

global x y D Pd VTR n NP itermax Fmin Fmax CRmin CRmax XVmin XVmax strategy PL PL2
Power_Of_Gen;
y=[];
VTR = 1.e-6;
D=n-1;
XVmin = [100 50 200 99 190 85 200 99 130];
XVmax = [250 230 500 265 490 265 500 265 440];

refresh= 20;
ct=[];
p=[];

[x,f,nf,Cost] =
devec('ED',VTR,D,XVmin,XVmax,y,NP,itermax,Fmin,Fmax,CRmin,CRmax,strategy,refresh);
 [cost,const,newpop,Fuel1,Fuel2,Fuel3,Fuel4,Fuel5,Fuel6,Fuel7,Fuel8,Fuel9,Fuel10,total,PL] =
ED(x,y);

Power_Of_Gen_2=Power_Of_Gen
Fuel=[Fuel1
   Fuel2
   Fuel3
   Fuel4
   Fuel5
```

```
   Fuel6
   Fuel7
   Fuel8
   Fuel9
   Fuel10]
Power_Demand=total
mincost=f
Power_Loss=PL
toc;
CPUTime=toc;
if Pd>3200
   clc;
   fprintf('Error !!! เกิดข้อผิดพลาดโปรแกรมไม่สามารถคำนวณได้เนื่องจากค่า Power Demand
มากกว่ากำหนด (3200 MW)\n');
   Power_Demand = Pd;
else
plot(Cost(1:200));
grid;
title('Convergence Curve of DEA');
xlabel('Iteration');
ylabel('Cost');
end
%toc;
%CPUTime=toc;
```

ED.m

```
function
[cost,const,newpop,Fuel1,Fuel2,Fuel3,Fuel4,Fuel5,Fuel6,Fuel7,Fuel8,Fuel9,Fuel10,total,PL]=ED
(x,y)
global Pd xlimit constant Power_Of_Gen PL PL2;
const=0;
para=zeros(10,6);
xlimit = [100 250;50 230;200 500;99 265;190 490;85 265;200 500;99 265;130 440;200 490];
constant = [.2697e+002    -.3975e0    .2176e-002  .2697e-001  -.3976e+001 100
        .2113e+002    -.3059e0    .1861e-002  .2113e-001  -.3059e+001 196
        .1184e+003    -.1269e+001 .4194e-002  .1184e0     -.1269e+002 157
        .1865e+001    -.3988e-001 .1138e-002  .1865e-002  -.3988e0    50
        .1365e+002    -.1980e0    .1620e-002  .1365e-001  -.1980e+001 114
        .3979e+002    -.3116e0    .1457e-002  .3979e-001  -.3116e+001 200
        -.5914e+002   .4864e0     .1176e-004  -.5914e-001 .4864e+001  388
        -.2875e+001   .3389e-001  .8035e-003  -.2876e-002 .3389e0     332
        .1983e+001    -.3114e-001 .1049e-002  .1983e-002  -.3114e0    99
        .5285e+002    -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
        .2668e+003    -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
        .1392e+002    -.8733e-001 .1066e-002  .1392e-001  -.8733e0    190
        .9976e+002    -.5206e0    .1597e-002  .9976e-001  -.5206e+001 338
        -.5399e+002   .4462e0     .1498e-003  -.5399e-001 .4462e+001  407
        .5285e+002    -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
        .1983e+001    -.3114e-001 .1049e-002  .1983e-002  -.3114e0    85
        .2668e+003    -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
        .1893e+002    -.1325e0    .1107e-002  .1893e-001  -.1325e+001 200
        .4377e+002    -.2267e0    .1165e-002  .4377e-001  -.2267e+001 331
        -.4335e+002   .3559e0     .2454e-003  -.4335e-001 .3559e+001  391
        .1983e+001    -.3114e-001 .1049e-002  .1983e-002  -.3114e0    99
```

```
        .5285e+002    -.6348e0    .2758e-002  .5285e-001  -.6348e+001 138
        .2668e+003    -.2338e+001 .5935e-002  .2668e0     -.2338e+002 200
        .8853e+002    -.5675e0    .1554e-002  .8853e-001  -.5675e+001 213
        .1423e+002    -.1817e-001 .6121e-003  .1423e-001  -.1817e0    130
        .1423e+002    -.1817e-001 .6121e-003  .1423e-001  -.1817e0    370
        .1397e+002    -.9938e-001 .1102e-002  .1397e-001  -.9938e0    200
        -.6113e+002   .5084e0     .4164e-004  -.6113e-001 .5084e+001  407
        .4671e+002    -.2024e0    .1137e-002  -.4671e-001 -.2024e+001 362];

B=(10^-3)*[0.2179 0.1062 -0.0027 -0.0077 0.0076 0.0345 0.2179 0.1062 -0.0027 -0.0077
    0.1062 0.1639 -0.0009 -0.0128 0.0040 0.0268 0.1062 0.1639 -0.0009 -0.0128
    -0.0027 -0.0009 0.2586 -0.1054 -0.0983 -0.0649 -0.0027 -0.0009 0.2586 -0.1054
    -0.0077 -0.0128 -0.1054 0.1919 0.0720 0.0366 -0.0077 -0.0128 -0.1054 0.1919
    0.0076 0.0040 -0.0983 0.0720 0.1601 -0.0007 0.0076 0.0040 -0.0983 0.0720
    0.0345 0.0268 -0.0649 0.0366 -0.0007 0.2568 0.0345 0.0268 -0.0649 0.0366
    0.2179 0.1062 -0.0027 -0.0077 0.0076 0.0345 0.2179 0.1062 -0.0027 -0.0077
    0.1062 0.1639 -0.0009 -0.0128 0.0040 0.0268 0.1062 0.1639 -0.0009 -0.0128
    -0.0027 -0.0009 0.2586 -0.1054 -0.0983 -0.0649 -0.0027 -0.0009 0.2586 -0.1054
    -0.0077 -0.0128 -0.1054 0.1919 0.0720 0.0366 -0.0077 -0.0128 -0.1054 0.1919];
B00=0.1407;
Bi0=[-0.0001 0.0016 -0.0043 0.0028 0.0012 0.0036 -0.0001 0.0016 -0.0043 0.0028];

g10=Pd-sum(x);
g=[x g10];
Power_Of_Gen=[x g10];
PL=0;
 ng = length(Power_Of_Gen);
 for i=1:ng;
    for j=1:ng;
       PL = PL+Power_Of_Gen(i)*B(i,j)*Power_Of_Gen(j);
```

```
    end
end
for i = 1:ng;
    PL = PL+Bi0(i)*Power_Of_Gen(i);
end
PL = PL+B00;
g10=Pd+PL-sum(x);
g=[x g10];
Power_Of_Gen=[x g10];

if g10>490
  c=g10-490;
  g10=490;
  diff1=250-x(1);
  diff2=230-x(2);
  diff3=500-x(3);
  diff4=265-x(4);
  diff5=490-x(5);
  diff6=265-x(6);
  diff7=500-x(7);
  diff8=265-x(8);
  diff9=440-x(9);
%end
diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
if diff > c
   if c < diff1
      x(1)=(x(1)+c);
      diff1=0;
      diff2=0;
```

```
        diff3=0;
        diff4=0;
        diff5=0;
        diff6=0;
        diff7=0;
        diff8=0;
        diff9=0;
    else
    x(1)=(x(1)+diff1);
    c=c-diff1;
    end
    if c <= 0
        diff = 0;
        c=0;
    else
    diff1=250-x(1);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff2
        x(2)=(x(2)+c);
        diff2=0;
        diff3=0;
        diff4=0;
        diff5=0;
        diff6=0;
        diff7=0;
        diff8=0;
        diff9=0;
```

```
    else
    x(2)=(x(2)+diff2);
    c=c-diff2;
    end
    if c <= 0
       diff = 0;
       c=0;
    else
    diff2=230-x(2);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff3
       x(3)=(x(3)+c);
       diff3=0;
       diff4=0;
       diff5=0;
       diff6=0;
       diff7=0;
       diff8=0;
       diff9=0;
    else
    x(3)=(x(3)+diff3);
    c=c-diff3;
    end
    if c <= 0
       diff = 0;
       c=0;
    else
```

```
    diff3=500-x(3);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff4
        x(4)=(x(4)+c);
        diff4=0;
        diff5=0;
        diff6=0;
        diff7=0;
        diff8=0;
        diff9=0;
    else
    x(4)=(x(4)+diff4);
    c=c-diff4;
    end
    if c <= 0
        diff = 0;
        c=0;
    else
    diff4=265-x(4);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff5
        x(5)=(x(5)+c);
        diff5=0;
        diff6=0;
```

```
        diff7=0;
        diff8=0;
        diff9=0;
    else
    x(5)=(x(5)+diff5);
    c=c-diff5;
    end
    if c <= 0
        diff = 0;
        c=0;
    else
    diff5=490-x(5);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff6
        x(6)=(x(6)+c);
        diff6=0;
        diff7=0;
        diff8=0;
        diff9=0;
    else
    x(6)=(x(6)+diff6);
    c=c-diff6;
    end
    if c <= 0
        diff = 0;
        c=0;
    else
```

```
    diff6=265-x(6);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end
if diff > c
    if c < diff7
        x(7)=(x(7)+c);
        diff7=0;
        diff8=0;
        diff9=0;
    else
    x(7)=(x(7)+diff7);
    c=c-diff7;
    end
    if c <= 0
        diff = 0;
        c=0;
    else
    diff7=500-x(7);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
 end
if diff > c
    if c < diff8
        x(8)=(x(8)+c);
        diff8=0;
        diff9=0;
    else
    x(8)=(x(8)+diff8);
    c=c-diff8;
```

```
    end
    if c <= 0
        diff = 0;
        c=0;
    else
    diff8=265-x(8);
    diff = diff1+diff2+diff3+diff4+diff5+diff6+diff7+diff8+diff9;
    end
end

else
    g=[x g10];
end
if g(1)<196
    para(1,:)=constant(1,:);
else
    para(1,:)=constant(2,:);
end
if g(2)<114
    para(2,:)=constant(4,:);
else if g(2)<157
    para(2,:)=constant(5,:);
    else para(2,:)=constant(3,:);
    end
end
if g(3)<332
    para(3,:)=constant(6,:);
else if g(3)<388
    para(3,:)=constant(8,:);
    else para(3,:)=constant(7,:);
```

```
    end
end
if g(4)<138
    para(4,:)=constant(9,:);
else if g(4)<200
    para(4,:)=constant(10,:);
    else para(4,:)=constant(11,:);
    end
end
if g(5)<338
    para(5,:)=constant(12,:);
else if g(5)<407
    para(5,:)=constant(13,:);
    else para(5,:)=constant(14,:);
    end
end
if g(6)<138
    para(6,:)=constant(16,:);
else if g(6)<200
    para(6,:)=constant(15,:);
    else para(6,:)=constant(17,:);
    end
end
if g(7)<331
    para(7,:)=constant(18,:);
else if g(7)<391
    para(7,:)=constant(19,:);
    else para(7,:)=constant(20,:);
    end
end
```

```
if g(8)<138
   para(8,:)=constant(21,:);
else if g(8)<200
   para(8,:)=constant(22,:);
   else para(8,:)=constant(23,:);
   end
end
if g(9)<213
   para(9,:)=constant(25,:);
else if g(9)<370
   para(9,:)=constant(24,:);
   else para(9,:)=constant(26,:);
   end
end
if g(10)<362
   para(10,:)=constant(27,:);
else if g(10)<407
   para(10,:)=constant(29,:);
      else
         para(10,:)=constant(28,:);
       end
end

total=g10-PL+sum(x);

for i=1:10
   if (g(i)<xlimit(i,1)|g(i)>xlimit(i,2))
       const=const+1;
   end
```

```
end


if Power_Of_Gen(1)<196
   Fuel1=1;
else
   Fuel1=2;
end
if Power_Of_Gen(2)<114
    Fuel2=2;
else if Power_Of_Gen(2)<157
   Fuel2=3;
   else Fuel2=1;
   end
end
if Power_Of_Gen(3)<332
   Fuel3=1;
else if Power_Of_Gen(3)<388
   Fuel3=3;
   else Fuel3=2;
   end
end
if Power_Of_Gen(4)<138
   Fuel4=1;
else if Power_Of_Gen(4)<200
  Fuel4=2;
   else Fuel4=3;
   end
end
if Power_Of_Gen(5)<338
```

```
   Fuel5=1;
else if Power_Of_Gen(5)<407
   Fuel5=2;
   else Fuel5=3;
   end
end
if Power_Of_Gen(6)<138
   Fuel6=2;
else if Power_Of_Gen(6)<200
   Fuel6=1;
   else Fuel6=3;
   end
end
if Power_Of_Gen(7)<331
  Fuel7=1;
else if Power_Of_Gen(7)<391
  Fuel7=2;
   else Fuel7=3;
   end
end
if Power_Of_Gen(8)<138
   Fuel8=1;
else if Power_Of_Gen(8)<200
   Fuel8=2;
   else Fuel8=3;
   end
end
if Power_Of_Gen(9)<213
  Fuel9=3;
else if Power_Of_Gen(9)<370
```

```
  Fuel9=1;
   else Fuel9=3;
   end
end
if Power_Of_Gen(10)<362
   Fuel10=1;
else if Power_Of_Gen(10)<407
   Fuel10=3;
   else Fuel10=2;
   end
end



newpop=x;
% Multi Fuel Cost Function with System Losses
%Considerations
%%cost=sum(para(1:10,1)+para(1:10,2).*g'+para(1:10,3).*(g'.^2));
%Multi Fuel Cost Function with Valve-Point Loading and System Losses
%Considerations
cost=sum((para(1:10,1)+para(1:10,2).*g'+para(1:10,3).*(g'.^2))+...
abs(para(1:10,4).*sin(para(1:10,5).*(para(1:10,6)-g'))));


S=cost*10^(5*const);
```

devec.m

```
function [bestmem,bestval,nfeval,Cost] =
devec(fname,VTR,D,XVmin,XVmax,y,NP,itermax,Fmin,Fmax,CRmin,CRmax,strategy,refresh);
t1=0.1;
t2=0.1;
F=zeros(NP,D);
CR=zeros(NP,D);
constp=linspace(0,0,NP);
constu=linspace(0,0,NP);
stp=0;
for i=1:NP
   aaa=Fmin+rand*(Fmax-Fmin);
   F(i,:)=aaa;
   bbb=CRmin+rand*(CRmax-CRmin);
   CR(i,:)=bbb;
end
refresh = floor(refresh);

%-----Initialize population and some arrays-----------------------------

pop = zeros(NP,D); %initialize pop to gain speed

%----pop is a matrix of size NPxD. It will be initialized----------
%----with random values between the min and max values of the---------
%----parameters-------------------------------------------------------

for i=1:NP
  pop(i,:) = XVmin + rand(1,D).*(XVmax - XVmin);
end
```

```
popold    = zeros(size(pop));     % toggle population
val       = zeros(1,NP);          % create and reset the "cost array"
bestmem   = zeros(1,D);           % best population member ever
bestmemit = zeros(1,D);           % best population member in iteration
nfeval    = 0;                    % number of function evaluations
constbm =0;

%------Evaluate the best member after initialization----------------------
ibest   = 1;                      % start with first population member
[cost,const,newpop]=feval(fname,pop(ibest,:),y);
val(1)  = cost;
constp(1)=const;
pop(ibest,:)=newpop;
bestval = val(1);                 % best objective function value so far
constbm = const;
nfeval  = nfeval + 1;
for i=2:NP                        % check the remaining members
  [value,const,npop]=feval(fname,pop(i,:),y);
  val(i) = value;
  constp(i)=const;
  pop(i,:)=npop;
  nfeval  = nfeval + 1;
if ((val(i) < bestval) & (constp(i)==0))          % if member is better
    ibest   = i;                  % save its location
    bestval = val(i);
    constbm = constp(i);
end
end
bestmemit = pop(ibest,:);         % best member of current iteration
```

```
bestvalit = bestval;            % best value of current iteration

bestmem = bestmemit;            % best member ever

%------DE-Minimization---------------------------------------------
%------popold is the population which has to compete. It is--------
%------static through one iteration. pop is the newly--------------
%------emerging population.----------------------------------------

pm1 = zeros(NP,D);              % initialize population matrix 1
pm2 = zeros(NP,D);              % initialize population matrix 2
pm3 = zeros(NP,D);              % initialize population matrix 3
pm4 = zeros(NP,D);              % initialize population matrix 4
pm5 = zeros(NP,D);              % initialize population matrix 5
bm  = zeros(NP,D);              % initialize bestmember  matrix
ui  = zeros(NP,D);              % intermediate population of perturbed vectors
mui = zeros(NP,D);              % mask for intermediate population
mpo = zeros(NP,D);              % mask for old population
rot = (0:1:NP-1);               % rotating index array (size NP)
rotd= (0:1:D-1);                % rotating index array (size D)
rt  = zeros(NP);                % another rotating index array
rtd = zeros(D);                 % rotating index array for exponential crossover
a1  = zeros(NP);                % index array
a2  = zeros(NP);                % index array
a3  = zeros(NP);                % index array
a4  = zeros(NP);                % index array
a5  = zeros(NP);                % index array
ind = zeros(4);

iter = 1;
```

```
%while ((iter < itermax) & (stp<=200) | (constbm == 1))
while ((iter < itermax))
  popold = pop;                  % save the old population
  ind = randperm(4);             % index pointer array

  a1  = randperm(NP);            % shuffle locations of vectors
  rt = rem(rot+ind(1),NP);       % rotate indices by ind(1) positions
  a2  = a1(rt+1);                % rotate vector locations
  rt = rem(rot+ind(2),NP);
  a3  = a2(rt+1);
  rt = rem(rot+ind(3),NP);
  a4  = a3(rt+1);
  rt = rem(rot+ind(4),NP);
  a5  = a4(rt+1);

  pm1 = popold(a1,:);            % shuffled population 1
  pm2 = popold(a2,:);            % shuffled population 2
  pm3 = popold(a3,:);            % shuffled population 3
  pm4 = popold(a4,:);            % shuffled population 4
  pm5 = popold(a5,:);            % shuffled population 5

for i=1:NP                       % population filled with the best member
    bm(i,:) = bestmemit;         % of the last iteration
end

  mui = rand(NP,D) < CR;         % all random numbers < CR are 1, 0 otherwise

if (strategy > 5)
    st = strategy-5;             % binomial crossover
else
```

```
   st = strategy;        % exponential crossover
   mui=sort(mui');          % transpose, collect 1's in each column
for i=1:NP
     n=floor(rand*D);
if n > 0
        rtd = rem(rotd+n,D);
        mui(:,i) = mui(rtd+1,i); %rotate column i by n
end
end
   mui = mui';          % transpose back
end



  mpo = mui < 0.5;               % inverse mask to mui


if (st == 1)                  % DE/best/1
   ui = bm + F.*(pm1 - pm2);       % differential variation
   ui = popold.*mpo + ui.*mui;     % crossover
elseif (st == 2)                % DE/rand/1
   ui = pm3 + F.*(pm1 - pm2);      % differential variation
   ui = popold.*mpo + ui.*mui;     % crossover
elseif (st == 3)                % DE/rand-to-best/1
   ui = popold + F.*(bm-popold) + F.*(pm1 - pm2);
   ui = popold.*mpo + ui.*mui;     % crossover
elseif (st == 4)                % DE/best/2
   ui = bm + F.*(pm1 - pm2 + pm3 - pm4);  % differential variation
   ui = popold.*mpo + ui.*mui;          % crossover
elseif (st == 5)                % DE/rand/2
   ui = pm5 + F.*(pm1 - pm2 + pm3 - pm4);  % differential variation
   ui = popold.*mpo + ui.*mui;      % crossover
```

```
end

%-----Select which vectors are allowed to enter the new population------------
for i=1:NP
   [cost,const,newpop]=feval(fname,ui(i,:),y);
   tempval = cost;        % check cost of competitor
   constu(i)=const;
   ui(i,:)=newpop;
   nfeval  = nfeval + 1;
if ((constu(i)<constp(i)) | ((constu(i)==constp(i))&(tempval <= val(i))))  % if competitor is better
than value in "cost array"
      pop(i,:) = ui(i,:);  % replace old vector with new one (for new iteration)
      val(i)   = tempval;  % save value in "cost array"
      constp(i)=constu(i);

%----we update bestval only in case of success to save time-----------
if ((tempval < bestval) & (constu(i)==0) )    % if competitor better than the best one ever
         bestval = tempval;      % new best value
         bestmem = ui(i,:);      % new best parameter vector ever
         constbm = constu(i);
end
end
end%---end for imember=1:NP
  idxstp=max(abs(bestmem-bestmemit));
if idxstp==0
     stp=stp+1;
else stp=0;
end
```

```
 bestmemit = bestmem;       % freeze the best member of this iteration for the coming
% iteration. This is needed for some of the strategies.

%----Output section------------------------------------------------------------

if (refresh > 0)
if (rem(iter,refresh) == 0)
      fprintf(1,'Iteration: %d,  Best: %f,  NP: %d\n',iter,bestval,NP);
for n=1:D
       fprintf(1,'best(%d) = %f\n',n,bestmem(n));
end
end
end
Cost(iter)=bestval;

 iter = iter + 1;
for i=1:NP
j1=rand;
if (j1<t1)
        aaa=Fmin+rand*(Fmax-Fmin);
        F(i,:)=aaa;
end
    j2=rand;
if (j2<t2)
        bbb=CRmin+rand*(CRmax-CRmin);
        CR(i,:)=bbb;
end
end
end%---end while ((iter < itermax) ...
mmm=1+1;
```

ประวัติโดยย่อของนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นายทิวัตถ์ เอี่ยวสำราญ

วัน เดือน ปีเกิด 10 กันยายน พ.ศ. 2534

สถานที่เกิด กรุงเทพมหานคร

ที่อยู่ปัจจุบัน 304/49 ซอย ลาดพร้าว 87/8 ถนน ลาดพร้าว
แขวง คลองเจ้าคุณสิงห์ เขต วังทองหลาง
จังหวัด กรุงเทพฯ 10310

โทรศัพท์ 089-7929784

E-mail Ninja_Kakashi_1@hotmail.com

ประวัติการศึกษา

ปีการศึกษา 2549 มัธยมศึกษาตอนต้น โรงเรียนมัธยมวัดบึงทองหลาง

ปีการศึกษา 2552 มัธยมศึกษาตอนปลาย โรงเรียนมัธยมวัดบึงทองหลาง

ปีการศึกษา 2556 กำลังศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์
ภาควิชา วิศวกรรมไฟฟ้า มหาวิทยาลัยศรีนครินทรวิโรฒ

ประวัติโดยย่อของนิสิตผู้ทำโครงงาน

ชื่อ-สกุล นายวิรัช วุ่นกลิ่นหอม

วัน เดือน ปีเกิด 3 กรกฎาคม 2534

สถานที่เกิด กรุงเทพมหานคร

ที่อยู่ปัจจุบัน 167 ซอย ลาดพร้าว 130 ถนน ลาดพร้าว
แขวง คลองจั่น เขต บางกะปิ
จังหวัด กรุงเทพฯ 10240

โทรศัพท์ 083-1306909

E-mail virat1150@hotmail.com

ประวัติการศึกษา

ปีการศึกษา 2549 มัธยมศึกษาตอนต้น โรงเรียนมัธยมวัดบึงทองหลาง

ปีการศึกษา 2552 มัธยมศึกษาตอนปลาย โรงเรียนมัธยมวัดบึงทองหลาง

ปีการศึกษา 2556 กำลังศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์
ภาควิชา วิศวกรรมไฟฟ้า มหาวิทยาลัยศรีนครินทรวิโรฒ