ขั้นตอนวิธีการซ่อมแซมร่วมกับความคลาดเคลื่อนยินยอมแบบยืดหยุ่นประยุกต์ใช้แก้ปัญหาการหาค่าเหมาะสมที่สุด

นายณัชวัฒน์ พลอยวัฒนาวงศ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR)
are the thesis authors' files submitted through the Graduate School.

Repair Algorithms with Tolerance Relaxation for Optimization Problems

Mr. Nutchawat Ploywattanawong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2012
Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ขั้นตอนวิธีการซ่อมแซมร่วมกับความคลาดเคลื่อนยินยอมแบบยืดหยุ่นประยุกต์ใช้แก้ปัญหาการหาค่าเหมาะสมที่สุด
โดย	นายณัชวัฒน์ พลอยวัฒนาวงศ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม
(ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.พีรยุทธ์ ชาญเศรษฐิกุล)

ณัชวัฒน์ พลอยวัฒนาวงศ์: ขั้นตอนวิธีการซ่อมแซมร่วมกับความคลาดเคลื่อนยินยอมแบบยืดหยุ่นประยุกต์ใช้แก้ปัญหาการหาค่าเหมาะสมที่สุด (Repair Algorithms with Tolerance Relaxation for Optimization Problems) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ศ.ดร.ประภาส จงสถิตย์วัฒนา, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร.สุรเทพ เขียวหอม, 86 หน้า

การหาค่าเหมาะสมที่สุดครอบคลุมในปัญหาที่มีฟังก์ชันวัตถุประสงค์ และ ข้อจำกัดไม่เชิงเส้นนั้นเป็นเรื่องทีน่าสนใจ ซึ่งหนึ่งในเครื่องมือที่ใช้หาค่าเหมาะสมที่สุดครอบคลุม คือ วิธีเชิงวิวัฒน์แบบส่วนต่าง ซึ่งได้ถูกนำไปประยุกต์ใช้จัดการข้อจำกัดทั้งสมการ และ อสมการได้สำเร็จด้วยดี ในงานวิจัยนี้นำเสนอรูปแบบการจัดการข้อจำกัดใหม่ด้วยการนำค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น และ รูปแบบการคัดเลือกความคล้ายมาใช้งานร่วมกับวิธีเชิงวิวัฒน์แบบส่วนต่าง มีชื่อเรียกว่า T-DRDE ด้วยการปรับปรุงต่อยอดจากวิธีการก่อนหน้า ที่ประกอบไปด้วยรูปแบบความคัดเลือกตามความเด่นกับขั้นตอนวิธีซ่อมแซมซึ่งเป็นการนำข้อมูลเชิงเกรเดียนต์มาทำการจัดการสมการข้อจำกัด ซึ่งเดิมมีชื่อเรียกว่า DRDE เพื่อที่จะทำการเปรียบเทียบกันระหว่าง T-DRDE กับ DRDE การศึกษาครั้งนี้ใช้ปัญหาการหาค่าเหมาะสมที่สุดเชิงวิศวกรรมเคมีมาร่วมทดสอบ โดยผลลัพธ์แสดงได้ว่า T-DRDE มีประสิทธิภาพในการจัดการข้อจำกัดที่เหนือกว่า DRDE ซึ่งประสิทธิภาพที่ดีวัดได้จากจำนวนการประเมินฟังก์ชันที่น้อยกว่า และ คำตอบที่ได้มีค่าใกล้เคียงกับค่าเหมาะสมที่สุดครอบคลุมมากกว่า

ภาควิชา วิศวกรรมคอมพิวเตอร์ ..... ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2555 ..... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม .....

5470186321 : MAJOR COMPUTER ENGINEERING

KEYWORDS : CONSTRAINT HANDLING DIFFERENTIAL EVOLUTION CONSTRAINED OPTIMIZATION TOLERANCE RELAXATION

NUTCHAWAT PLOYWATTANAWONG : REPAIR ALGORITHMS WITH TOLERANCE RELAXATION FOR OPTIMIZATION PROBLEMS : ADVISOR : PROF. PRABHAS CHONGSTITVATANA, Ph.D., CO-ADVISOR : ASST. PROF. SOORATHEP KHEAWHOM, Ph.D., 86 pp.

Global optimization is interesting when both the objective function and constraints are nonlinear. One of global optimizers, Differential Evolution, has been applied to these related problems successfully. This thesis introduces a new constraint handling scheme using adaptive relax tolerance and familiar selection scheme with differential evolution (T-DRDE). This method is an improvement of the previous method, a dominance-based selection scheme with a repair algorithm based on the gradient information derived from equality constraints (DRDE). To compare between both methods, several test based chemical engineering optimization problems are used. A good performance should have low Number of Function Evaluations (NFE) and the objective function value is near the global optimum. The results show that the performance of T-DRDE is more efficient than DRDE.

Department: Computer Engineering Student's Signature .....

Field of Study: Computer Engineering Advisor's Signature .....

Academic Year: 2012 Co-advisor's Signature .....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้รับคำแนะนำ ข้อคิดเห็น และ แนวทางในการทำวิจัยที่ดีต่างๆ จากอาจารย์ที่ปรึกษา ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา และ อาจารย์ที่ปรึกษาร่วม ผู้ช่วยศาสตราจารย์ ดร.สุรเทพ เขียวหอม ที่คอยช่วยเหลือให้ความรู้ทางด้านที่จำเป็นต่อวิทยานิพนธ์ และคำปรึกษาที่สามารถแก้ไขปัญหาต่างๆ ที่เกิดขึ้นได้ นอกจากนั้นยังถ่ายทอดประสบการณ์ต่างๆ ให้เป็นข้อคิด และ แนวทางที่ดีต่อการทำงานต่อไป ผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

กราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และ รองศาสตราจารย์ ดร.พีรยุทธ์ ชาญเศรษฐิกุล ที่สละเวลามาให้ข้อเสนอแนะ และแนวคิดต่างๆ ที่เป็นประโยชน์ต่อการพัฒนาวิทยานิพนธ์ฉบับนี้

ขอขอบคุณจุฬาลงกรณ์มหาวิทยาลัยที่ให้ความรู้ และ ประสบการณ์ที่ดีต่างๆ ที่เป็นส่วนหนึ่งในการใช้ชีวิตในมหาวิทยาลัย รวมทั้งขอขอบคุณอาจารย์ทุกๆ ท่านที่ได้อบรม สั่งสอน ให้ผู้วิจัยได้รับความรู้ในวิชาต่างๆ ที่เป็นประโยชน์ในการทำงาน อีกทั้งเพื่อนๆ พี่ๆ น้องๆ ที่ดี ที่คอยช่วยเหลือ และ เป็นกำลังใจให้ในช่วงเวลาต่างๆ

สุดท้ายนี้ กราบขอบพระคุณทุกคนในครอบครัว และ ญาติๆ ที่คอยสนับสนุนให้ผู้ทำวิจัยได้เรียนรู้ และ ทำในสิ่งที่ดี คอยเป็นกำลังใจให้กัน และ กันตลอดมา

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญตาราง ..... ฎ
สารบัญภาพ ..... ฎ
บทที่ 1 บทนำ 1
1.1 ที่มา และ ความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิจัย 9
1.3 ขอบเขตของงานวิจัย 9
1.4 ประโยชน์ของงานวิจัย 10
1.5 ขั้นตอน และ วิธีดำเนินการวิจัย 10
1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์ 11
บทที่ 2 ทฤษฎี และ งานวิจัยที่เกี่ยวข้อง 12
2.1 ทฤษฎีที่เกี่ยวข้อง 12
2.1.1 ขั้นตอนวิธีเชิงวิวัฒน์ (Evolutionary Algorithm) 12
2.1.2 ปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัด 17
2.1.3 วิธีจัดการข้อจำกัด 22
2.2 งานวิจัยที่เกี่ยวข้อง 32
2.2.1 งานวิจัยของ Srinivas และ Rangaiah 32
2.2.2 งานวิจัยของ Shang, Zhao และ Shen 32
2.2.3 งานวิจัยของ Zahara และ Kao 33
2.2.4 งานวิจัยของ Kheawhom 33
2.2.5 งานวิจัยของ Liu, Cai และ Wang 34
2.2.6 งานวิจัยของ Zhang, Rangaiah, และ Adrian 34
2.2.7 งานวิจัยของ Zhang และ Rangaiah 35
2.2.8 สรุป และ วิเคราะห์แนวคิดในงานวิจัยต่างๆ 35
บทที่ 3 วิธีดำเนินการวิจัย 39
3.1 ขั้นตอนวิธีการซ่อมแซมเชิงเกรเดียนต์ที่ใช้ระเบียบวิธีของนิวตันมาจัดการสมการข้อจำกัด 39

หน้า

3.1.1 การสร้างประชากรตั้งต้น (Initial) 40
3.1.2 ตัวดำเนินการกลายพันธุ์ (Mutation Operator) 41
3.1.3 ตัวดำเนินการสลับสายพันธุกรรม (Crossover Operator) 41
3.1.4 ขั้นตอนวิธีซ่อมแซม (Repair Algorithm) 42
3.1.5 รูปแบบการคัดเลือก (Selection Scheme) 43
3.1.6 เกณฑ์ในการหยุดกระบวนการ (Stop Criteria) 43
3.2 ค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) 44
3.2.1 กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) 48
3.2.2 รูปแบบการคัดเลือกความคล้าย (Familiar selection scheme) 51
3.3 องค์ประกอบที่ใช้วัดคุณภาพของงาน 52
3.3.1 จำนวนการประเมินฟังก์ชัน (Number of Function Evaluations, NFE) 52
3.3.2 อัตราความสำเร็จ (Success Rate) 53
3.3.3 วิเคราะห์กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอม 54
บทที่ 4 การทดลอง และ ผลการทดลอง 55
4.1 เครื่องมือที่ใช้ในการทดลอง 55
4.2 กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมที่นำมาใช้เปรียบเทียบ 56
4.3 วิธีการทดลอง 57
4.3.1 เปรียบเทียบประสิทธิภาพของกฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมทั้ง 4 แบบ 57
4.3.2 เปรียบเทียบประสิทธิภาพของวิธี T-DRDE เทียบกับวิธี DRDE และ C-IDE 68
บทที่ 5 สรุปผลการวิจัย 73
5.1 สรุปผลการวิจัย 73
5.2 ผลงานตีพิมพ์จากวิทยานิพนธ์ 74
รายการอ้างอิง 75
ภาคผนวก 80
ภาคผนวก ก 81
ภาคผนวก ข 85
ประวัติผู้เขียนวิทยานิพนธ์ 86

สารบัญตาราง

หน้า

ตารางที่ 1.1 รายละเอียดขั้นตอนวิธีทางพันธุกรรมที่ถูกพัฒนาเข้ากับการหาค่าเหมาะที่สุด 2
ตารางที่ 2.1 ปัจจัยสำคัญของแนวคิดในงานวิจัยต่างๆ 36
ตารางที่ 4.1 คุณลักษณะทางคณิตศาสตร์ของปัญหา 55
ตารางที่ 4.2 พารามิเตอร์ และ การตั้งค่าสำหรับใช้ในการทดลอง มีดังนี้ 56
ตารางที่ 4.3 กฎการปรับค่าคลาดเคลื่อนทั้ง 4 กฎ 56
ตารางที่ 4.4 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 1 และกฎที่ 2 57
ตารางที่ 4.5 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 1 และ 2 58
ตารางที่ 4.6 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 2 และ กฎที่ 3 60
ตารางที่ 4.7 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 2 และ 3 60
ตารางที่ 4.8 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 3 และ กฎที่ 4 63
ตารางที่ 4.9 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 3 และ 4 64
ตารางที่ 4.10 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 4 และ วิธี DRDE 68
ตารางที่ 4.11 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 4 และ วิธี DRDE 69
ตารางที่ 4.12 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 4 และ วิธี C-IDE 71
ตารางที่ 4.13 จำนวนการประเมินฟังก์ชันทั้งหมดของวิธีต่างๆโดยสรุป 72

สารบัญภาพ

หน้า

ภาพที่ 1.1 การจำแนกกลุ่มของแบบจำลองทางคณิตศาสตร์ในปัญหาการหาค่าเหมาะสมที่สุด 5
ภาพที่ 2.1 ลักษณะการเคลื่อนที่ของอนุภาค..... 14
ภาพที่ 2.2 ผังการทำงานของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง..... 16
ภาพที่ 2.3 ความสัมพันธ์ของค่าต่ำสุด และ สูงสุด 20
ภาพที่ 2.4 ลักษณะของฟังก์ชันคอนเวกซ์จากสมการที่ 9..... 20
ภาพที่ 2.5 ลักษณะของฟังก์ชันคอนเวกซ์ และ ฟังก์ชันเว้า..... 21
ภาพที่ 2.6 บริเวณการค้นหาของสมการ และ อสมการข้อจำกัด..... 22
ภาพที่ 2.7 บริเวณของการค้นคำตอบ และ บริเวณที่มีความเป็นไปได้ 22
ภาพที่ 2.8 ลักษณะการทำ homomorphous map กับรูปร่างปัญหาที่เป็นคอนเวกซ์ 26
ภาพที่ 2.9 ลักษณะการทำ homomorphous map กับรูปร่างปัญหาที่ไม่เป็นคอนเวกซ์..... 27
ภาพที่ 2.10 ผังการทำงานของวิธีจัดการสมการข้อจำกัด 30
ภาพที่ 2.11 การกระจายของเวกเตอร์ประชากรก่อน และ หลังกระบวนการซ่อมแซม 30
ภาพที่ 3.1 ภาพรวมการทำงานประสานขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างกับขั้นตอนวิธีซ่อมแซม 40
ภาพที่ 3.2 บริเวณการค้นคำตอบของอสมการข้อจำกัด g1(X, Y) ในตัวอย่าง 45
ภาพที่ 3.3 บริเวณการค้นคำตอบของอสมการข้อจำกัด g2(X, Y) ในตัวอย่าง 45
ภาพที่ 3.4 บริเวณที่เป็นไปได้จากซ้อนทับของบริเวณที่เป็นไปได้ของอสมการ g1 และ g2 46
ภาพที่ 3.5 จุดที่เป็นไปได้จากการซ้อนทับของเส้นสมการกับบริเวณที่เป็นไปได้ 47
ภาพที่ 3.6 บริเวณที่เป็นไปได้รอบจุดที่เป็นไปได้ ซึ่งสร้างจากค่าคลาดเคลื่อนยินยอม 48
ภาพที่ 3.7 กราฟระหว่างแกน Z (TAV) ที่ตำแหน่งของตัวแปร X และ ตัวแปร Y..... 48
ภาพที่ 4.1 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 1 และ 2 57
ภาพที่ 4.2 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันวิธี T-DRDE กฎที่ 1 และ 2 ... 59
ภาพที่ 4.3 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 2 และ 3 60
ภาพที่ 4.4 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันวิธี T-DRDE กฎที่ 2 และ 3 ... 62
ภาพที่ 4.5 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 3 และ 4 63
ภาพที่ 4.6 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันวิธี T-DRDE กฎที่ 3 และ 4 ... 65
ภาพที่ 4.7 พฤติกรรมของค่าคลาดเคลื่อนในแต่ละรอบการวนซ้ำของกฎทั้ง 4 67
ภาพที่ 4.8 กราฟความสัมพันธ์ระหว่างอัตราส่วนคำตอบกับวิธี T-DRDE กฎที่ 4 กับ DRDE..... 68
ภาพที่ 4.9 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันวิธี T-DRDEกฎที่4 กับ DRDE 70

บทที่ 1

บทนำ

1.1 ที่มา และ ความสำคัญของปัญหา

การหาค่าเหมาะสมที่สุด (Optimization) เกิดจากความต้องการแก้ปัญหาทางคณิตศาสตร์เพื่อหาผลลัพธ์ที่เหมาะสมที่สุด ซึ่งปัญหาส่วนใหญ่นั้นมีหลักการพื้นฐานทางด้านฟิสิกส์มาเกี่ยวข้อง ช่วงก่อนปีค.ศ. 1940 วิธีการหาค่าเหมาะสมที่สุดที่ถูกนำมาใช้กับปัญหาทางด้านฟิสิกส์ของฟังก์ชันที่มีหลายตัวแปรนั้นมีไม่มากนัก เช่น การทำกำลังสองน้อยสุด (Least Squared) แต่วิธีที่เป็นที่รู้จักและ นิยมใช้กันมาก คือ การทำระเบียบวิธีของนิวตัน (Newton's Method) ซึ่งได้นำมาประยุกต์ใช้กับปัญหาทางเคมี จากนั้นจึงมีกำหนดการเชิงเส้น (Linear Programming) ซึ่งเป็นสาขาใหม่สำหรับการหาค่าเหมาะสมที่สุดภายในปี ค.ศ. 1940 – ค.ศ. 1950 แล้วจึงมีการพัฒนาขึ้นมาอย่างต่อเนื่องจนถึงปัจจุบัน ซึ่งปัจจุบันนี้ได้มีการนำวิธีการหาค่าเหมาะสมที่สุดไปใช้อย่างกว้างขวาง ไม่ว่าจะเป็นด้านคณิตศาสตร์ วิทยาศาสตร์ วิศวกรรมศาสตร์ รวมถึงเศรษฐศาสตร์ด้วย

การหาค่าเหมาะสมที่สุดนอกจากจะมีการนำไปใช้แก้ปัญหาทางด้านวิศวกรรมกันอย่างมากแล้ว ในปัจจุบันนี้ยังได้นำมาประยุกต์ใช้กับทุกสาขาวิชา โดยเฉพาะทางด้านเศรษฐศาสตร์ที่มีการประยุกต์นำหลักการหาค่าเหมาะสมที่สุดมาใช้กับเรื่องของการลงทุน เช่น เพิ่มผลกำไร เพิ่มผลผลิต ลดต้นทุนการผลิต หรือลดความเสี่ยงในการลงทุน เป็นต้น ส่วนสาขาวิชาด้านอื่นๆ เช่น สาขาทางด้านเทคโนโลยีสารสนเทศ ที่ใช้หลักการหาค่าเหมาะสมที่สุดมาแก้ปัญหาด้านการจัดการเครือข่าย เพื่อหาเส้นทางการสื่อสารกันในเครือข่ายให้เร็วที่สุด หรือสาขาไฟฟ้าที่เป็นการปรับสัญญาณหาแรงดันไฟฟ้าตกค้างในวงจรน้อยที่สุดเป็นต้น

สำหรับในภาคอุตสาหกรรม จะเป็นปัญหาเกี่ยวกับการปรับปรุงประสิทธิภาพของกระบวนการผลิตให้ดีที่สุด เช่น การเพิ่มกำลังการผลิตโดยใช้ต้นทุนในกระบวนการผลิตลดลง หรือการลดความเสี่ยงในการลงทุนโดยให้เกิดกำไรที่มากที่สุด ซึ่งในการเพิ่มประสิทธิภาพนั้นจะต้องมีการออกแบบ และ วางแผนการทำงานที่ถูกต้องตามที่ต้องการ ซึ่งวิศวกรยังต้องออกแบบระบบให้สามารถประหยัดใช้จ่ายอีกด้วย ปัญหาเหล่านี้ส่วนใหญ่จะเป็นเรื่องซับซ้อนยากต่อการคำนวณตาม

วิธีปกติเนื่องจากมีทางเลือกมากมายในการหาผลลัพธ์ที่เหมาะสมเหล่านั้น และ จะทำอย่างไรถึงจะสามารถคำนวณหาทางเลือกที่ทำให้ได้ผลลัพธ์ที่เหมาะสมที่สุดอีกด้วย [1]

ในการแก้ปัญหาการหาค่าเหมาะสมที่สุด (Optimization Problem) เชิงวิศวกรรมจะเริ่มจากการกำหนดฟังก์ชันวัตถุประสงค์ (Objective Function) สมการข้อจำกัด (Equality Constraints) และ อสมการข้อจำกัด (Inequality Constraints) จากนั้นจึงทำการแก้ปัญหา ซึ่งโดยปกติแล้วกระบวนการทางคณิตศาสตร์สามารถแก้ได้ง่ายหากมีข้อจำกัดมากกว่าหรือเท่ากับจำนวนตัวแปรตัดสินใจ แต่ปัญหาส่วนใหญ่นั้นมีจำนวนสมการ และ อสมการข้อจำกัดน้อยกว่าจำนวนตัวแปรตัดสินใจ ซึ่งทำให้ข้อจำกัดไม่ครอบคลุมพอจัดการตัวแปรตัดสินใจทุกตัว จึงมีตัวแปรตัดสินใจบางตัวที่ไม่ถูกจำกัดช่วงการค้นคำตอบ ซึ่งจะทำให้บริเวณของการค้นคำตอบกว้าง

เมื่อไม่กี่ปีที่ผ่านมาได้มีการพัฒนาเทคนิคในการหาค่าเหมาะสมที่สุดที่มีประสิทธิภาพ และเป็นที่นิยม ซึ่งเป็นวิธีการที่ไม่เป็นตามแบบดั้งเดิม (Nontraditional) และ ได้มีการนำวิธีนี้เข้าไปร่วมกับสาขาวิชาทางด้านพันธุกรรมมีรายละเอียดดังตารางที่ 1.1

ตารางที่ 1.1 แสดงรายละเอียดขั้นตอนวิธีทางพันธุกรรมที่ถูกพัฒนาเข้ากับการหาค่าเหมาะที่สุด [1]

วิธีทางพันธุกรรมที่ถูกนำมาใช้ร่วมกัน	**ปีที่ถูกคิดค้น**	**ผู้คิดค้น**
ขั้นตอนวิธีทางพันธุกรรม (Genetic Algorithm)	1975	John Holland
การอบอ่อนจำลอง (Simulated Annealing)	1983	Kirkpatrick, Gelatt และ Vecchi
การหาค่าเหมาะสมที่สุดแบบโครงข่ายประสาทเทียม (Neural Network Optimization)	1985	Hopfield และ Tank
การหาค่าเหมาะสมที่สุดด้วยตรรกศาสตร์คลุมเครือ (Fuzzy Logic Optimization)	1986	Rao
การหาค่าเหมาะสมที่สุดด้วยระบบอาณาจักรมด (Ant Colony Optimization)	1992	Marco Dorigo

ตารางที่ 1.1 แสดงรายละเอียดขั้นตอนวิธีทางพันธุกรรมที่ถูกพัฒนาเข้ากับการหาค่าเหมาะที่สุด(ต่อ)

วิธีทางพันธุกรรมที่ถูกนำมาใช้ร่วมกัน	**ปีที่ถูกคิดค้น**	**ผู้คิดค้น**
การหาค่าเหมาะสมที่สุดด้วยกลุ่มอนุภาค (Particle Swarm Optimization)	1995	Kennedy และ Eberhart
ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (Differential Evolution)	1996	Storn และ Price

ตัวอย่างการประยุกต์ใช้ปัญหาการหาค่าเหมาะสมที่สุดทางด้านวิศวกรรม [1] ได้แก่

1) การออกแบบโครงสร้างของอาคารที่แข็งแรงที่สุด

2) การเลือกเครื่องจักรสำหรับกระบวนการผลิตเม็ดพลาสติกที่จะคืนกำไรมากที่สุด

3) การออกแบบปั๊ม (Pump) หรือ กังหัน (Turbine) ที่ให้ประสิทธิภาพมากที่สุด

4) การออกแบบวัสดุทางโยธาให้เสียค่าใช้จ่ายน้อยที่สุด

5) การเลือกเส้นทางการเดินรถที่ใกล้ที่สุด

6) การออกแบบโรงงาน (Plant) และ กระบวนการทางเคมีที่เหมาะสมที่สุด

7) การออกแบบการเดินระบบการวางท่อ (Pipeline) ในกระบวนการทางอุตสาหกรรมที่เหมาะสมที่สุด

8) การออกแบบระบบควบคุมที่เหมาะสมที่สุด

9) การวางแผนกลยุทธ์ที่ดีที่สุด และ ให้ผลกำไรมากที่สุด

10) การกักเก็บสินค้าให้เหมาะสมที่สุด

11) ปัญหาการออกแบบระบบเครื่องปฏิกรณ์ (Reactor Network Design Problem)

12) ปัญหาการออกแบบระบบเครื่องแลกเปลี่ยนความร้อน (Heat Exchanger Network Design Problem)

13) ปัญหาการกลั่นลำดับส่วน (Separation Network Synthesis Problem)

รูปแบบของปัญหาการหาค่าเหมาะสมที่สุดเชิงวิศวกรรมจัดอยู่ในปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัด (Constrained Optimization Problem) ซึ่งปัญหาดังกล่าวสามารถเขียนเป็นแบบจำลองทางคณิตศาสตร์ได้ โดยจะแบ่งย่อยออกเป็น 4 ประเภท ตามรูปแบบทางคณิตศาสตร์ และ ตัวดำเนินการทางคณิตศาสตร์ดังนี้

1) สมการเชิงเส้น (Linear Equality constraint, LE)

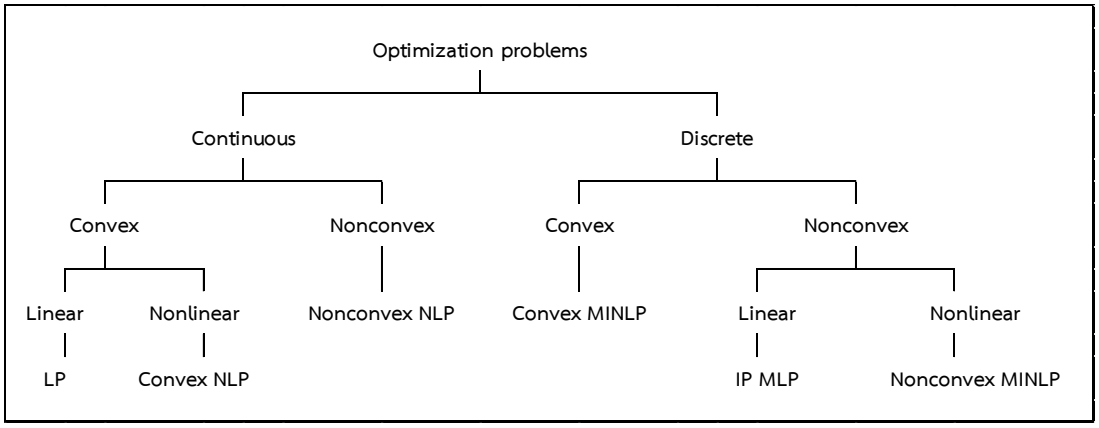
2) อสมการเชิงเส้น (Linear Inequality constraint, LI)

3) สมการไม่เชิงเส้น (Non-Linear Equality constraint, NLE)

4) อสมการไม่เชิงเส้น (Non-Linear Inequality constraint, NLI)

ซึ่งปัญหาการหาค่าเหมาะสมที่สุดอาจจะประกอบไปด้วยแบบจำลองทางคณิตศาสตร์ประเภทใดประเภทหนึ่ง หรือมีหลายประเภทผสมกันได้ ทั้งนี้ในการเรียกชื่อปัญหาการหาค่าเหมาะสมที่สุดว่าจัดอยู่ในกลุ่มใด สามารถพิจารณาได้จาก ถ้าหากปัญหาการหาค่าเหมาะสมที่สุดนั้นมีแบบจำลองทางคณิตศาสตร์เป็นสมการเชิงเส้น (LE) และ /หรืออสมการเชิงเส้น (LI) ปัญหานี้จะจัดอยู่ในกลุ่มกำหนดการเชิงเส้น (Linear Programming, LP) ถ้าหากปัญหาการหาค่าเหมาะสมที่สุดนั้นมีสมการหรืออสมการข้อจำกัดหรือฟังก์ชันวัตถุประสงค์ใดที่เป็นไม่เชิงเส้นประกอบแม้เพียงตัวแปรเดียว ปัญหานั้นจะถูกจัดอยู่ในกลุ่มกำหนดการไม่เชิงเส้น (Non-Linear Problem, NLP) โดยปกติแล้วค่าของตัวแปรตัดสินใจ (Decision Variables) จะมีค่าต่อเนื่อง (Continuous) บนขอบเขตที่กำหนดในช่วงจำนวนจริง แต่อย่างไรก็ตามมีบางปัญหาที่ค่าตัวแปรตัดสินใจบางตัวมีค่าไม่ต่อเนื่อง (Discrete) ซึ่งมีลักษณะเป็นจำนวนเต็มผสมอยู่ในปัญหาได้ (Mixed Integer) ซึ่งหากมีจำนวนเต็มผสมอยู่ในกลุ่มกำหนดการเชิงเส้นก็จะเรียกใหม่เป็น กำหนดการจำนวนเต็มผสมเชิงเส้น (Mixed Integer Linear Programming, MILP) ในทำนองเดียวกันหากมีจำนวนเต็มผสมอยู่ในกลุ่มกำหนดการไม่เชิงเส้นก็จะเรียกเป็น กำหนดการจำนวนเต็มผสมไม่เชิงเส้น (Mixed Integer Non-Linear Programming, MINLP) เมื่อพิจารณาถึงความยากในการแก้ปัญหาก็จะมีอยู่หลายประการให้พิจารณาทั้งจำนวนตัวแปรตัดสินใจ จำนวนข้อจำกัด (Constraints) ความขึ้นต่อกันของตัวแปรตัดสินใจเดียวกันบนข้อจำกัดต่างกัน รูปแบบของข้อจำกัดที่เป็นเชิงเส้นหรือไม่เชิงเส้น จำนวนสมการ และ อสมการ เป็นต้น แต่ปัจจัยสำคัญที่นำมาพิจารณาความยากของปัญหาเป็นอันดับแรกคือ ปัญหาการหาค่าเหมาะสมที่สุดนี้มีลักษณะของการเป็นคอนเวกซ์ (Convex) หรือไม่เป็นคอน

เวกซ์ (Non-convex) ซึ่งปัญหาที่มีลักษณะของการเป็นคอนเวกซ์จะมีความง่ายในการหาคำตอบ เนื่องจากปัญหาดังกล่าวมีความชันลู่เข้าสู่ค่าเหมาะสมที่สุดครอบคลุมเสมอ (Global Optimum) ส่วนปัญหาที่มีลักษณะไม่เป็นคอนเวกซ์จะมีความยากกว่าแบบคอนเวกซ์ เนื่องจากความชันของกราฟมีความเหมาะสมที่สุดเฉพาะที่ (Local Optimum) มาเกี่ยวข้อง ด้วยกลุ่มของปัญหาดังที่กล่าวมา ดังนั้นปัญหาแบบจำลองทางคณิตศาสตร์เชิงวิศวกรรมจึงแสดงดังภาพที่ 1



ภาพที่ 1.1 แสดงการจำแนกกลุ่มแบบจำลองทางคณิตศาสตร์ในปัญหาการหาค่าเหมาะสมที่สุด [2]

ในการแก้ปัญหาการหาค่าเหมาะสมที่สุดสามารถแบ่งเป็น 2 แบบ คือ [3]

1) วิธีเชิงกำหนด (Deterministic Methods) เป็นการรวมเอาวิธีประมาณภายนอก (Outer Approximation Methods) และ วิธีกำหนดขอบเขตตัวแปร (Branch & Bound Methods) [4] เข้าด้วยกัน เพื่อนำไปใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดของข้อจำกัด

2) วิธีสโทแคสติก (Stochastic Methods) ได้รวมวิธีต่างๆ ได้แก่ ขั้นตอนวิธีการอบอ่อนจำลอง (Simulated Annealing) การค้นแบบทาบู (Tabu Search) ขั้นตอนวิธีเชิงวิวัฒน์ (Evolutionary Algorithms, EAs) [5] ซึ่งนำไปประยุกต์กับการแก้ปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัดได้เช่นกัน

วิธีเชิงกำหนดนั้นสามารถประกันผลลัพธ์เหมาะสมที่สุดครอบคลุมได้ (Global Optimum) ภายใต้คุณสมบัติของฟังก์ชันวัตถุประสงค์ และ ข้อจำกัดเช่น ต่อเนื่อง และ เป็นคอนเวกซ์ (Continuity and Convexity) เป็นต้น ส่วนวิธีสโทแคสติกนั้นต้องการสมมติฐานน้อยหรือไม่ต้องการเลยในการแก้ปัญหาการหาค่าเหมาะสมที่สุด วิธีสโทแคสติกเป็นการหาความน่าจะเป็นในการลู่เข้าสู่คำตอบ (Convergence) ที่เหมาะสมที่สุดครอบคลุม อย่างไรก็ตามวิธีนี้ไม่สามารถ

ประกันค่าที่เหมาะสมที่สุดครอบคลุมได้ [6] แม้ว่าบ่อยครั้งที่การคำนวณจะให้ผลลัพธ์ที่ใกล้เคียงกับค่าเหมาะสมที่สุดครอบคลุมจากวิธีเชิงกำหนดก็ตาม ดังนั้นเป้าหมายหลักของการใช้วิธีสโทแคสติกอยู่ที่จัดการข้อจำกัดให้มีเส้นทางสู่ผลลัพธ์ไว้สำหรับนำทางในการค้นคำตอบของวิธีสโทแคสติกได้ และ ปัญหา คือ จะทำอย่างไรจึงสามารถจัดการข้อจำกัด (Handling Constraints) ได้อย่างมีประสิทธิภาพ

ปัจจุบันขั้นตอนวิธีเชิงวิวัฒน์ ได้รับความสนใจมากขึ้น เพราะมีคุณสมบัติการค้นคำตอบของปัญหาการหาค่าเหมาะสมที่สุดที่ซับซ้อนได้อย่างมีประสิทธิภาพ ดูได้จากจำนวนงานวิจัยที่นำขั้นตอนวิธีเชิงวิวัฒน์มาใช้ในช่วงหลายปีที่ผ่านมา ซึ่งคำอธิบายเกี่ยวกับขั้นตอนวิธีเชิงวิวัฒน์นั้นได้ถูกนำมาอ้างซ้ำหลายครั้ง [7]

ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (Differential Evolutionary Algorithm, DE) เป็นขั้นตอนวิธีหนึ่งที่เหมาะกับปัญหาการหาค่าเหมาะสมที่สุด ด้วยความสามารถในการแก้ปัญหาที่ไม่สามารถหาอนุพันธ์ได้ (Non-differentiable) และ ปัญหาที่มีลักษณะไม่เชิงเส้น [8] อีกทั้งยังสามารถหาผลลัพธ์ของฟังก์ชันวัตถุประสงค์ที่มีหลายคำตอบได้ (Multi-modal Objective Functions) แนวคิดพื้นฐานของวิธีเชิงวิวัฒน์แบบส่วนต่างจะเริ่มจากการสุ่มเลือกประชากรของเวกเตอร์ต้นแบบ (Parent Vector) เพื่อนำมาสร้างเป็นประชากรของเวกเตอร์รบกวน (Perturbed Vector) จากตัวดำเนินการกลายพันธุ์ (Mutation Operation) และ เวกเตอร์รบกวนจะถูกนำมาสร้างเป็นเวกเตอร์ทดลองอีกที (Trial Vector) ด้วยตัวดำเนินการสลับสายพันธุกรรม (Crossover Operation) ระหว่างเวกเตอร์รบกวนกับเวกเตอร์ต้นแบบ แล้วจึงนำเวกเตอร์ทดลองที่ได้ไปเปรียบเทียบกับเวกเตอร์ต้นแบบอีกครั้งเพื่อคัดเลือกประชากรต้นแบบในรุ่นถัดไป (Next Generation) วิธีเชิงวิวัฒน์แบบส่วนต่างเมื่อนำไปประยุกต์ใช้ในปัญหาการหาค่าเหมาะสมที่สุดจะมีความเสถียร และ ให้ประสิทธิภาพสูง อย่างไรก็ตามในปัญหาไม่เชิงเส้นหลายๆ ปัญหา การใช้วิธีเชิงวิวัฒน์แบบส่วนต่างมาจัดการข้อจำกัดจะได้คำตอบเป็นค่าเหมาะสมเฉพาะที่ (Local Optimum) แทนที่จะได้ค่าเหมาะสมครอบคลุม (Global Optimum) เพื่อให้คำตอบที่ได้เข้าใกล้ค่าเหมาะสมครอบคลุมมากที่สุด จึงได้นำวิธีจัดการข้อจำกัดมาใช้ร่วมกับวิธีเชิงวิวัฒน์แบบส่วนต่างด้วย

ในการแก้ปัญหาการหาค่าเหมาะสมที่สุดนั้นความซับซ้อนของปัญหาแต่ละแบบสามารถมองให้อยู่ในรูปของบริเวณของการค้นคำตอบได้ โดยบริเวณของการค้นคำตอบขึ้นอยู่กับความเป็น

สมการข้อจำกัด อสมการข้อจำกัด การจำกัดขอบเขตบน และ ล่างของตัวแปรตัดสินใจ (Upper & Lower Bounds) และ การกำหนดบริเวณที่เป็นไปได้ (Feasible Region) จากนั้นจึงนำวิธีการจัดการตัวแปรตัดสินใจให้เข้ายังบริเวณที่เป็นไปได้ดังกล่าวมาประยุกต์ใช้ร่วมกับวิธีสโทแคสติกอีกทีหนึ่ง เพื่อกำหนดทิศทางการวิ่งเข้าหาคำตอบให้มีประสิทธิภาพมากขึ้น โดยจะเรียกวิธีการดังกล่าวว่า วิธีจัดการข้อจำกัด (Constraint Handling Method) ซึ่งมีการพัฒนาไปหลายแนวทาง

การพัฒนาระเบียบวิธีจัดการข้อจำกัดที่ใช้กับการแก้ปัญหาการหาค่าเหมาะสมที่สุดหลายมีอยู่วิธี [9, 10] ซึ่งวิธีจัดการข้อจำกัดสำหรับวิธีสโทแคสติกสามารถแบ่งออกเป็น 5 แบบดังนี้

1) ฟังก์ชันการลงโทษ (Penalty Functions) [11]

2) ตัวดำเนินการ และ การแทนค่าแบบเฉพาะ (Special Representations and Operators)

3) ขั้นตอนวิธีซ่อมแซม (Repair Algorithms)

4) การแยกวัตถุประสงค์ และ ข้อจำกัด (Separation of Objectives and Constraints)

5) วิธีผสม (Hybrid Methods)

รูปแบบที่นิยมใช้จัดการข้อจำกัดมากที่สุด คือ ฟังก์ชันการลงโทษ ซึ่งมีแนวคิดพื้นฐานมาจากการแปลงปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัดให้กลายเป็นปัญหาการหาค่าเหมาะสมที่สุดแบบไม่มีข้อจำกัด (Unconstrained Optimization Problem) ด้วยการกำหนดการลงโทษให้กับค่าข้อจำกัดที่ไม่สอดคล้องกัน (Constraints Violation) คำนวณปรับค่าจากจุดที่เป็นไปไม่ได้ (Infeasible Point) จนได้ค่าข้อจำกัดที่สอดคล้องกัน แต่ความยากของการใช้ฟังก์ชันการลงโทษนี้คือ การปรับค่าตัวแปรการลงโทษ เนื่องจากตัวแปรการลงโทษนี้อ่อนไหวต่อคุณภาพของคำตอบเป็นอย่างมาก ในการแก้ไขปัญหาดังกล่าวมีงานวิจัยหลายแนวทางที่ต่างกันออกไป แนวทางหนึ่งเรียกว่าการลงโทษแบบพลวัต (Dynamic Penalty) คือ ทำการเพิ่มค่าการลงโทษในแต่ละรุ่นการวนซ้ำ แต่อย่างไรก็ตามการหาค่าการลงโทษแบบพลวัตที่ดีนั้นก็ยังทำได้ยาก

ขั้นตอนวิธีซ่อมแซม [12] เป็นการปรับผลลัพธ์ที่เป็นไปไม่ได้ (Infeasible Solutions) ให้ลู่เข้าสู่ขอบเขตที่เป็นไปได้ (Feasible Region) โดยทั่วไปในกระบวนการซ่อมแซมนั้นจะใช้กฎของฮิวริสติก (Heuristic Rules) เป็นตัวดำเนินการ [13, 14] ดังเช่นในปี 2006 Chootinan และ Chen

[15] ได้เสนอกฎของฮิวริสติกเป็นข้อมูลเชิงเกรเดียนต์ (Gradient Information) ซึ่งคำนวณได้จากข้อจำกัดเพื่อนำไปใช้ซ่อมแซมผลลัพธ์ที่เป็นไปไม่ได้ต่อไป ขั้นตอนวิธีซ่อมแซมเป็นตัวเลือกที่ดีหากผลลัพธ์ที่ไม่เหมาะสมนั้นสามารถถูกแก้ไขได้ง่าย ซึ่งข้อเสียของวิธีการนี้ คือ ความยากง่ายขึ้นอยู่กับแต่ละปัญหา และ ใช้ทรัพยากรการคำนวณเยอะในกรณีที่ปัญหาการหาค่าเหมาะสมที่สุดมีความซับซ้อน อีกทั้งยังไม่เหมาะกับปัญหาที่ไม่สามารถหาอนุพันธ์ได้ด้วย

มีหลายการทดลองพัฒนาวิธีจัดการข้อจำกัดด้วยการแยกจัดการของฟังก์ชันวัตถุประสงค์และ ข้อจำกัด ในปี 2002 Coello และ Montes [16] เสนอวิธีการที่มีแนวคิดของการเลือกแบบความไม่เด่น (Non-dominance-based Selection) ในปี 2006 Takahama และ Sakai [17] ได้เสนอวิธีเชิงวิวัฒน์แบบส่วนต่างที่ใช้พารามิเตอร์ ε เพื่อเพิ่มความยืดหยุ่นของข้อจำกัด (Constraint Relaxation) สำหรับหาจุดที่เหมาะสมในแต่ละรอบ ค่า ε นี้ใช้กับปัญหาที่มีข้อจำกัดเป็นสมการ แต่ความยากอยู่ที่การปรับค่า ε ให้เหมาะสมกับปัญหา ในปี 2008 Shang, Zhao และ Shen [18] ได้เสนอขั้นตอนวิธีทางพันธุกรรมซึ่งปรับตัวได้ โดยทำงานร่วมกับค่าความคลาดเคลื่อนยินยอมที่ยืดหยุ่นได้ (FTGA) เพื่อสร้างบริเวณใกล้จุดที่เป็นไปได้ (Tolerance of feasible point) สำหรับให้ประชากรที่มีค่าที่ดีที่สุดก้าวเข้ามาในบริเวณดังกล่าว ด้วยวิธีการจากขั้นตอนวิธีทางพันธุกรรมข้างต้น

ในปี 2010 Kheawhom [19] เสนอวิธี DRDE เป็นการผสมระหว่างวิธีเลือกแบบความเด่น (Dominance-based Selection Scheme) เป็นวิธีพิจารณาเวกเตอร์ตัวแปรที่ให้ค่าข้อจำกัดที่สอดคล้องกัน และ /หรือมีค่าฟังก์ชันวัตถุประสงค์ที่ดีกว่า กับขั้นตอนวิธีซ่อมแซมเชิงเกรเดียนต์ที่จัดการข้อจำกัดที่เป็นสมการโดยเฉพาะ ผลลัพธ์ที่ได้เมื่อเทียบกับวิธีฟังก์ชันการลงโทษแล้วให้ค่าฟังก์ชันวัตถุประสงค์ใกล้เคียงกัน แต่ใช้จำนวนการประเมินฟังก์ชัน (Functions Evaluated) น้อยกว่า อีกทั้งยังมีอัตราความสำเร็จในการหาคำตอบ (Success Rate) สูงด้วย ในปี 2011 Haibo Zhang [20] ได้เสนอวิธีจัดการข้อจำกัดชื่อว่า C-IDE เป็นวิธีที่ใช้วิธีเชิงวิวัฒน์แบบส่วนต่างร่วมกับการค้นแบบทาบู และ จัดการข้อจำกัดที่ประยุกต์มาจากวิธีของ Takahama และ Sakai มีพารามิเตอร์ μ คิดจากผลรวมของค่าข้อจำกัดที่ไม่สอดคล้องกัน เพื่อกำหนดเป็นขอบเขตที่เหมาะสมแล้วจึงใช้ฟังก์ชันฮิวริสติกที่ประยุกต์ขึ้นมาปรับลดช่วงกว้างของค่า μ ในแต่ละรอบให้แคบลง เมื่อเปรียบเทียบผลลัพธ์ที่ได้กับวิธี DRDE แล้ว C-IDE ใช้ทรัพยากรในการคำนวณน้อยกว่ามาก โดยที่

ความแม่นยำของคำตอบใกล้เคียงกัน แต่เมื่อพิจารณาจำนวนประชากรที่ C-IDE ใช้นั้น ปรากฏว่ากำหนดค่าไว้ต่างกับ DRDE ซึ่งจำนวนประชากรที่ C-IDE ใช้น้อยกว่า DRDE มากเช่นกัน ข้อเสียของ DRDE อยู่ที่ความเข้มงวดของค่าคลาดเคลื่อนยินยอมภายในกระบวนการซ่อมแซมซึ่งมีผลทำให้ใช้จำนวนการประเมินฟังก์ชันมีค่ามาก แต่อย่างไรก็ตามอัตราความสำเร็จในการหาคำตอบของ C-IDE ยังคงน้อยกว่า DRDE

งานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาวิธีจัดการข้อจำกัดให้มีประสิทธิภาพ ในการแก้ปัญหาการหาค่าเหมาะสมที่สุด จึงได้เสนอวิธีการปรับปรุงขั้นตอนวิธีซ่อมแซมด้วยค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) สำหรับใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดในกลุ่มไม่เชิงเส้น ทั้งปัญหาที่มีลักษณะของการเป็นคอนเวกซ์ หรือไม่เป็นคอนเวกซ์ อีกทั้งยังช่วยเพิ่มประสิทธิภาพการจัดการข้อจำกัดให้ทำงานร่วมกับขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างในการจัดการกับอสมการข้อจำกัด ซึ่งขั้นตอนวิธี DRDE ยังให้ประสิทธิภาพไม่ดีพอ โดยงานวิจัยนี้เป็นการพัฒนาต่อยอดมาจากวิธี DRDE ประยุกต์แนวคิดจากวิธี C-IDE ในการปรับค่าคลาดเคลื่อนยินยอมให้มีความผ่อนคลายมากขึ้น และ จะค่อยๆ ลดความผ่อนคลายลงในแต่ละรอบการวนซ้ำจนได้ค่าคลาดเคลื่อนที่เหมาะ ซึ่งวิธีที่นำเสนอนี้จะทำให้ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างมีประสิทธิภาพต่อการคัดเลือกประชากรในแต่ละรอบมากขึ้น และ ช่วยลดจำนวนการประเมินฟังก์ชัน โดยที่ยังคงความแม่นยำของคำตอบที่ใกล้เคียงเดิม

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อพัฒนาวิธีจัดการข้อจำกัดสำหรับขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างให้มีประสิทธิภาพในการแก้ปัญหาการหาค่าเหมาะสมที่สุดมากขึ้น

1.3 ขอบเขตของงานวิจัย

1) ใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดที่มีสมการข้อจำกัดเป็นส่วนหนึ่งของปัญหา

2) ขั้นตอนวิธีที่นำมาเปรียบเทียบประสิทธิภาพ คือ ขั้นตอนวิธี DRDE และ C-IDE

3) โจทย์การวัดเปรียบเทียบสถานะใช้โจทย์เดียวกับงานวิจัยของ Kheawhom [19]

4) ใช้ชุดคำสั่ง และ วิธีเดียวกันทั้งหมดในการนับจำนวนการประเมินฟังก์ชันในขั้นตอนวิธี

5) เกณฑ์การประเมินคุณภาพพิจารณาจากส่วนประกอบทั้ง 2 ดังนี้

5.1) จำนวนการประเมินฟังก์ชันที่น้อยกว่า

5.2) ความแม่นยำของคำตอบที่ใกล้เคียงกับผลลัพธ์ที่ของงานวิจัยส่วนใหญ่ ทั้งค่าฟังก์ชันวัตถุประสงค์น้อยที่สุด และ ค่าตัวแปรตัดสินใจ

1.4 ประโยชน์ของงานวิจัย

สามารถนำไปใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดได้ถูกต้องแม่นยำ และ ประหยัดทรัพยากรในการคำนวณ เป็นประโยชน์กับอาชีพที่มีความเกี่ยวข้องกับปัญหาเหล่านี้ เช่น วิศวกรรมเคมี ที่มีการคำนวณหาสภาวะที่เหมาะสมที่สุดในการผลิตสารเคมี ทั้งปัญหาการออกแบบระบบเครื่องปฏิกรณ์ (Reactor Network Design Problem) ปัญหาการออกแบบระบบเครื่องแลกเปลี่ยนความร้อน (Heat Exchanger Network Design Problem) หรือปัญหาการกลั่นลำดับส่วน (Separation Network Synthesis Problem) เป็นต้น

1.5 ขั้นตอน และ วิธีดำเนินการวิจัย

1) ศึกษาผังการทำงานของวิธีเชิงวิวัฒน์แบบส่วนต่าง

2) ศึกษาปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัด

3) ศึกษาวิธีจัดการข้อจำกัดแบบต่างๆ ดังนี้

3.1) ฟังก์ชันการลงโทษ

3.2) รูปแบบคัดเลือกแบบความเด่น

3.3) ขั้นตอนวิธีซ่อมแซม

3.4) ความคลาดเคลื่อนยินยอมแบบยืดหยุ่น

4) แบ่งการพัฒนาขั้นตอนวิธีออกเป็น 2 แบบดังนี้

4.1) การเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีที่เสนอโดยมีกฎการเปลี่ยนแปลงค่าคลาดเคลื่อนต่างกัน

4.2) การเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีที่เสนอกับขั้นตอนวิธี DRDE กับ C-IDE

5) ทำการทดลองโดยการวัดประสิทธิภาพขั้นตอนวิธีดังนี้

5.1) ความถูกต้องแม่นยำของคำตอบ

5.2) อัตราความสำเร็จในการค้นคำตอบ

5.3) จำนวนการประเมินฟังก์ชันวัตถุประสงค์ และ ข้อจำกัดที่ไม่สอดคล้องกัน

6) วิเคราะห์ผลการทดลอง สรุป และ เรียบเรียงวิทยานิพนธ์

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์

โครงสร้างเนื้อหาของวิทยานิพนธ์ฉบับนี้ถูกแบ่งออกเป็น 5 บท คือ บทที่ 1 เป็นบทนำ บทที่ 2 กล่าวถึงทฤษฎี และ งานวิจัยต่างๆ ที่เกี่ยวข้อง บทที่ 3 อธิบายถึงวิธีการดำเนินงานวิจัย กระบวนการทำงานของขั้นตอนวิธีที่เสนอ แนวคิดของการทำความคลาดเคลื่อนยินยอมแบบยืดหยุ่น และ หลักการออกแบบกฎการเปลี่ยนแปลงของค่าคลาดเคลื่อนยินยอม กับรูปแบบการคัดเลือกปรับค่าคลาดเคลื่อน บทที่ 4 อธิบายการทดลอง เปรียบเทียบระหว่างกฎการเปลี่ยนแปลงด้วยกันเอง จนถึงขั้นตอนวิธีอื่นๆ และ ผลการทดลองของการเปรียบเทียบนั้น รวมถึงการวิเคราะห์ผลการทดลอง และ บทที่ 5 เป็นบทสุดท้ายเป็นบทสรุป และ ข้อจำกัดของงานวิจัย รวมทั้งงานวิจัยในอนาคต และ บทความวิชาการที่ตีพิมพ์

บทที่ 2

ทฤษฎี และ งานวิจัยที่เกี่ยวข้อง

ในบทนี้ ทฤษฎี ความรู้ และ งานวิจัยที่เกี่ยวข้องกับงานวิจัยจะถูกอธิบาย โดยเริ่มที่ความรู้และ ทฤษฎีที่เกี่ยวข้องเพื่อเป็นความรู้พื้นฐาน แล้วจึงอธิบายงานวิจัยที่เกี่ยวข้องเพื่อเห็นภาพโดยรวมเกี่ยวกับงานวิจัยอื่นๆ ตามลำดับ

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ขั้นตอนวิธีเชิงวิวัฒน์ (Evolutionary Algorithm)

การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (Particle Swarm Optimization, PSO)

การหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค (PSO) นั้นถูกคิดค้นขึ้นมาโดย Kennedy, Eberhart และ Shi [21], [22] ในปี 1995 โดยมีแนวคิดมาจากการสังเกตพฤติกรรมของฝูงนก และปลาที่มีลักษณะการเคลื่อนที่ที่สอดคล้องกันโดยเฉพาะเวลาหาอาหารเป็นกลุ่ม ต่อมาจึงนำมาประยุกต์ใช้ในสายวิทยาศาสตร์คอมพิวเตอร์ ด้วยการใช้แก้ปัญหาการหาค่าเหมาะสมที่สุด PSO คือวิธีการคำนวณที่ซึ่งทำให้เกิดความเหมาะสมของปัญหาขึ้นด้วยการทำวนซ้ำเพื่อปรับคุณภาพของผลลัพธ์ที่มีการคัดเลือก (Candidate Solution) ซึ่ง PSO จะทำให้ปัญหามีความเหมาะสมได้โดยเริ่มสร้างประชากรเหล่านี้ขึ้นมา ในที่นี้หมายถึงอนุภาคขนาดเล็กหลายอนุภาค แล้วจึงทำการเคลื่อนย้ายอนุภาคเหล่านี้ไปรอบๆ บริเวณของการค้นคำตอบที่ซึ่งถูกกำหนดโดยแบบจำลองทางคณิตศาสตร์ให้อยู่ในรูปของตำแหน่ง และ ความเร็ว เมื่ออนุภาคแต่ละตัวนั้นเคลื่อนที่ไปยังตำแหน่งต่างๆ จะทำการเก็บตำแหน่งที่ดีที่สุดเฉพาะที่ไว้ แล้วจึงเริ่มติดต่อกันเพื่อชักจูงกันจากตำแหน่งที่ดีที่สุดเฉพาะที่ (Local Best Known Position, PBest) ไปยังตำแหน่งที่ดีที่สุดครอบคลุม (The Best Known Position, GBest) ภายในบริเวณของการค้นคำตอบเหล่านี้ โดยการชักจูงนั้นทำได้โดยการปรับปรุงให้เป็นตำแหน่งให้ดีกว่าด้วยตำแหน่งที่พบโดยอนุภาคอื่นๆ ด้วยการชักนำเช่นนี้จึงเกิดการเกาะกลุ่มของฝูงอนุภาคเคลื่อนย้ายกันไปยังผลลัพธ์ที่ดีที่สุด

ขั้นตอนวิธีพื้นฐานของ PSO ทุกๆ รอบการทำงานเวลา t ความเร็วของการเคลื่อนที่จะถูกเปลี่ยนแปลงโดยใช้ข้อมูลของตำแหน่งที่ดีที่สุดของอนุภาคแต่ละตัว $p_i(t)$ ตำแหน่งที่ดีที่สุดของอนุภาคทั้งหมด $p_g(t)$ ความเร็วอนุภาคแต่ละตัวที่เปลี่ยนไปจะสามารถคำนวณได้ด้วยสมการที่ 1

$$v_i(t+1) = v_i(t) + c_1Q_1(p_i(t) - x_i(t)) + c_2Q_2(p_g(t) - x_i(t)) \quad (1)$$

โดยค่า c_1 และ c_2 เป็นค่าคงที่ที่มีค่าเป็นบวก ค่า Q_1 และ Q_2 เป็นตัวเลขที่ทำการสุ่มขึ้นมามีค่าอยู่ในช่วง [0, 1] ค่า $v_i(t)$ คือ ความเร็วของการเคลื่อนที่อนุภาคแต่ละตัว และ ค่า $x_i(t)$ คือ ตำแหน่งของแต่ละอนุภาค ซึ่งความเร็วใหม่ที่คำนวณได้นั้นจะมีค่าความเร็วสูงสุดไม่เกิน ($\pm v_{max}$) เพื่อควบคุมความเร็วไม่ให้มากเกินไป จากนั้นในทุกๆ รอบการทำงานความเร็วปัจจุบันของอนุภาคจะถูกนำไปคำนวณดังสมการที่ 2 เพื่อหาตำแหน่งใหม่ด้วยเช่นกัน

$$x_i(t+1) = x_i(t) + v_i(t) \quad (2)$$

ด้วยสมการการเคลื่อนที่นี้ ทำให้อนุภาคสามารถทำการค้นคำตอบไปรอบด้านในขณะกำลังเข้าใกล้คำตอบที่ดีที่สุด ซึ่งถือข้อดีเนื่องจาก การหาคำตอบอื่นรอบข้างระหว่างทางทำให้อนุภาคอาจพบเส้นทางการเคลื่อนที่ที่ดีกว่าการเคลื่อนที่เข้าหาจุดที่ดีที่สุดในปัจจุบัน แต่อย่างไรก็ตามหากกำหนดให้ระหว่างทางอนุภาคทำการค้นเส้นทางอย่างละเอียด ก็อาจทำให้ความเร็วในการเคลื่อนที่มีค่าน้อยการลู่เข้าสู่คำตอบที่ดีจะช้าตามไปด้วย

ขั้นตอนการทำงานของ PSO มีดังนี้

1) สร้างอนุภาคเริ่มต้นจากการสุ่มตำแหน่งให้

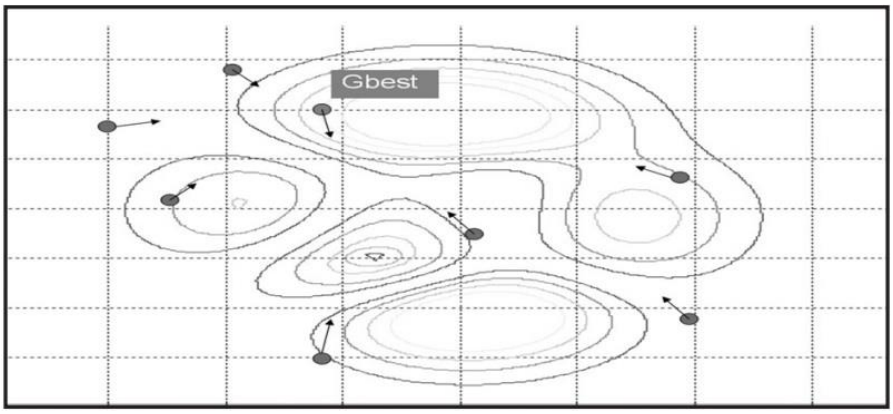
2) ในเส้นทางการเคลื่อนที่ของแต่ละอนุภาค ให้ทำการคำนวณตำแหน่งที่ดีที่สุดเฉพาะที่ (Local Best Known Position, PBest)

3) ทำการเลือกเลือกตำแหน่งที่ดีที่สุดครอบคลุม (The Best Known Position, GBest) จากตำแหน่งที่ดีที่สุดเฉพาะที่ของอนุภาคทุกตัว

4) ปรับความเร็วใหม่ และ ตำแหน่งใหม่ให้อนุภาคแต่ละตัว

5) เคลื่อนที่อนุภาคไปยังตำแหน่งใหม่

ทำวนซ้ำจนได้คำตอบหรือครบจำนวนรอบที่กำหนด ลักษณะการเคลื่อนที่ของอนุภาคจะแสดงดังภาพที่ 2.1



ภาพที่ 2.1 ลักษณะการเคลื่อนที่ของอนุภาค [23]

PSO สามารถทำการค้นหาคำตอบด้วยการเคลื่อนที่ไปในบริเวณใกล้เคียงตนเองเท่านั้น ถ้าบริเวณที่ต้องการค้นหามีขนาดใหญ่หรือคำตอบที่ต้องการอยู่ห่างจากตำแหน่งของอนุภาค การค้นหาคำตอบก็จะทำได้อย่างเชื่องช้า ข้อจำกัดส่วนนี้แตกต่างกับ GAs ที่สามารถประมาณค่าคำตอบที่อยู่ห่างออกไปได้อย่างรวดเร็วแต่ไม่มีวิธีในการค้นหาคำตอบที่อยู่ใกล้กับตัวมันแบบ PSO [23]

การค้นหาแบบทาบู (Tabu Search, TS)

เป็นการค้นหาข้อมูลในคอมพิวเตอร์แบบวิธีโคจรตามเส้นกราฟ โดยการค้นทาบูมีลักษณะพิเศษ คือ มีเพิ่มประสิทธิภาพการค้นหาตามเส้นกราฟแบบเดิมด้วยการใช้โครงสร้างข้อมูลเพื่อจำปัญหาที่ได้แก้ และ ทราบคำตอบที่ยอมรับได้แล้ว ไม่แก้ปัญหาเดิมซ้ำอีกครั้งหรือปัญหาที่ไม่มีทางแก้ได้สำเร็จในเวลาที่จำกัด Glover [24] เป็นผู้ริเริ่มเสนอแนวคิดของ TS ไว้ตั้งแต่ปีพ.ศ. 2520

การค้นหาแบบทาบูนิยมนำไปใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงการจัด (Combinatorial Optimization) เช่นปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem) โดยที่การค้นหาแบบทาบูนั้นมีขั้นตอนวิธีการทำงาน คือ เมื่อสิ้นสุดการทำงานการวนซ้ำในแต่ละรอบปัจจุบัน ก่อนที่จะเริ่มต้นการวนซ้ำในรอบถัดไป จะทำการประเมินผลลัพธ์ด้วยฟังก์ชันเหมาะสมเพื่อทราบถึงผลลัพธ์ที่ดีที่สุด จากนั้นจึงทำการเลือกผลลัพธ์รอบบริเวณที่ดีที่สุดนั้น มาคำนวณเพื่อหาผลลัพธ์ที่ดีอื่นโดยรอบ โดยพิจารณาจากผลลัพธ์ดังกล่าวมีค่าเข้าเงื่อนไขที่กำหนดไว้หรือไม่ ผลลัพธ์ที่ดีในรอบปัจจุบันจะถูกบันทึกไว้ในรายการทาบู (Tabu List) ซึ่งการแก้ปัญหาใน

รอบถัดไปจะนำรายการทาบูมาร่วมพิจารณาด้วย ซึ่งรายการทาบูนั้นเก็บผลลัพธ์ และ เส้นทางที่เคยคำนวณมาแล้ว จึงสามารถทราบ และ หลีกเลี่ยงไม่พิจารณาปัญหาเดิมอีกครั้ง ทำให้ประหยัดทรัพยากรในการคำนวณ และ ยังเป็นการบังคับให้เพิ่มขอบเขตการค้นหาไปยังพื้นที่ในส่วนใหม่ที่ยังไม่ได้รับการค้นหา

ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (Differential Evolution) [8]

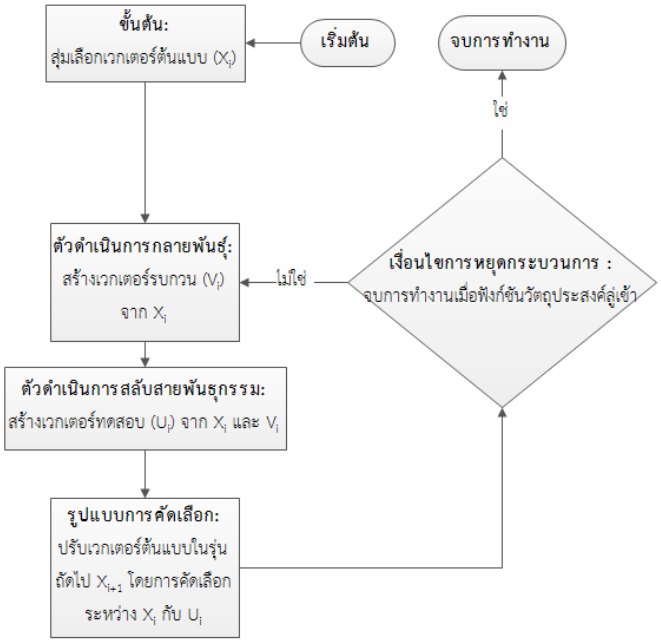
ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างเป็นขั้นตอนวิธีเชิงวิวัฒน์ประเภทหนึ่ง มีข้อดีตรงที่ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างมีความเสถียร และ มีประสิทธิภาพมากกับตัวแปรที่มีความต่อเนื่อง และไม่เชิงเส้น ภาพที่ 2.2 แสดงผังการทำงานของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง

โดยขั้นตอนวิธีจะเริ่มจากสุ่มสร้างประชากรตั้งต้น X_i จำนวน N ตัว จากนั้นจึงนำประชากรตั้งต้นที่ได้มาดำเนินการกลายพันธุ์เพื่อสร้างเป็นเวกเตอร์รบกวน V_i แล้วจึงสร้างเวกเตอร์ทดลอง U_i จากการสลับสายพันธุกรรมของ X_i และ V_i

รูปแบบการดำเนินการกลายพันธุ์ [25] เพื่อสร้างเวกเตอร์รบกวนมีหลายแบบ เช่น รูปแบบพื้นฐาน (Basic Scheme) DE/rand/1/bin และ รูปแบบตรีโกณมิติ (Trigonometric Scheme) [26] โดยที่รูปแบบพื้นฐาน DE/rand/1/bin เริ่มจากการสุ่มเลือกเวกเตอร์มา 3 ตัวจากประชากรทั้งหมด แล้วจึงนำมาสร้างเป็นเวกเตอร์รบกวน V_i ดังสมการที่ 3

$$V_i = X_{r3} + F(X_{r2} - X_{r1}) \tag{3}$$

โดย X_{r1}, X_{r2} และ X_{r3} เป็นเวกเตอร์ที่ถูกสุ่มขึ้นมา ซึ่ง $r1 \neq r2 \neq r3 \neq i$ และ $F \in [0,1+]$ เป็นพารามิเตอร์ควบคุม



ภาพที่ 2.2 แสดงผังการทำงานของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง

ส่วนรูปแบบการกลายพันธุ์แบบตรีโกณมิตินั้นยังคงเริ่มด้วยการสุ่มเวกเตอร์มา 3 เวกเตอร์จากประชากรทั้งหมดเช่นเดียวกับรูปแบบ DE/rand/1/bin แต่เวกเตอร์รบกวนจะถูกคำนวณหาค่ากลางจากเวกเตอร์ทั้งสามตามการกระจายตัวแบบไฮเพอร์จีออเมตริกทรงสามเหลี่ยม เวกเตอร์รบกวนที่ถูกสร้างขึ้นจะเป็นตามชุดสมการที่ 4

$$V_i = \frac{X_{r1} + X_{r2} + X_{r3}}{3} + (P_2 - P_1)(X_{r1} - X_{r2}) + (P_3 - P_2)(X_{r2} - X_{r3}) + (P_1 - P_3)(X_{r3} - X_{r1})$$

$$P_1 = \frac{|F(X_{r1})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|},$$

$$P_2 = \frac{|F(X_{r2})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|}, \qquad (4)$$

$$P_3 = \frac{|F(X_{r3})|}{|F(X_{r1})| + |F(X_{r2})| + |F(X_{r3})|}$$

โดย X_{r1}, X_{r2} และ X_{r3} คือ เวกเตอร์ที่ถูกสุ่มเลือก ซึ่ง r1 ≠ r2 ≠ r3 ≠ i และ $F(X_{r1})$, $F(X_{r2})$ และ $F(X_{r3})$ คือ ค่าของฟังก์ชันวัตถุประสงค์ที่คำนวณได้จาก X_{r1}, X_{r2} และ X_{r3} ตามลำดับ

เวกเตอร์รบกวน $V_i(v_{i,1}, v_{i,2}, ..., v_{i,n})$ และ เวกเตอร์รุ่นพ่อของมัน X_i $(x_{i,1}, x_{i,2}, ..., x_{i,n})$ จะถูกนำไปใช้ในขั้นตอนการสลับสายต่อ ซึ่งผลลัพธ์ท้ายสุดจะสร้างเป็นเวกเตอร์ทดสอบ U_i $(u_{i,1}, u_{i,2}, ..., u_{i,n})$ ดังสมการที่ 5

$$u_{i,j} = \begin{cases} v_{i,j} & \text{ถ้า random } [0.1) \leq CR \\ x_{i,j} & \text{ถ้าไม่ใช่} \end{cases} \quad (5)$$

โดยที่ CR $\in$ [0.1] คือ ค่าการสลับสายพันธุกรรม (Crossover Rate) ในแต่ละรุ่นของ i จะทำการสร้างเวกเตอร์ทดสอบ U_i ซึ่งจะนำไปเปรียบเทียบกับเวกเตอร์พ่อ X_i ถ้าหากเวกเตอร์ทดสอบดีกว่าเวกเตอร์พ่อ เวกเตอร์ทดสอบนั้นจะถูกแทนที่ลงไปในประชากรแทนเวกเตอร์พ่อนั้นมีรูปแบบการแทนที่ดังสมการที่ 6 กระบวนการเชิงวิวัฒน์จะถูกทำซ้ำไปจนกว่าพบเงื่อนไขในการจบการวนซ้ำที่กำหนดไว้

$$X_{i+1} = \begin{cases} X_i & \text{ถ้า } X_i \prec U_i \\ U_i & \text{ถ้าไม่ใช่} \end{cases} \quad (6)$$

โดยที่ $X_i \prec U_i$ นิยามว่า X_i ให้ค่าที่ดีกว่า U_i

2.1.2 ปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัด

ปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัดมีรูปแบบทั่วไปดังสมการที่ 7

$$\begin{aligned} \min \quad & F(x), \\ \text{s.t.} \quad & g_l(x) \leq 0; && l = 1, 2, ..., L. \\ & h_m(x) = 0; && m = 1, 2, ..., M, \\ & x_n^{(L)} \leq x_n \leq x_n^{(U)}; && n = 1, 2, ..., N \end{aligned} \quad (7)$$

โดยที่ x = { x_1, x_2, x_3, ..., x_N} คือ ตัวแปรการหาค่าที่เหมาะสมที่สุด มีค่าอยู่ในช่วง $[x_n^{(L)}, x_n^{(U)}]$, ฟังก์ชัน F(x) คือ ฟังก์ชันวัตถุประสงค์ ส่วน $g_l(x)$ และ $h_m(x)$ เป็นอสมการข้อจำกัด และ สมการข้อจำกัดตามลำดับ

1) ตัวแปรตัดสินใจ (Decision Variable) หมายถึง ตัวแปรที่เป็นคำตอบของการแก้ปัญหาที่เหมาะสมที่สุด ซึ่งตัวแปรตัดสินใจจะเก็บค่าของตัวแปรทั้งหมดที่เกี่ยวข้องทั้งกับฟังก์ชันวัตถุประสงค์หรือข้อจำกัดก็ตาม เช่น การออกแบบระบบท่อก็จะพิจารณาว่ามีตัวแปรใดมาเกี่ยวข้องกับระบบบ้าง โดยตัวแปรตัดสินใจจะถูกกำหนดเพื่อใช้อธิบายลักษณะของระบบทางด้านวิศวกรรมอย่างชัดเจน เช่น ขนาด น้ำหนัก รูปทรงของวัสดุ จำนวนของฟันในเกียร์ จำนวนของคอยล์ในสปริง หรือจำนวนของท่อในเครื่องแลกเปลี่ยนความร้อน โดยการกำหนดตัวแปรออกแบบนั้นต้องเลือกตัวแปรที่ใช้กำหนดในระบบให้ถูกต้องกับโจทย์แต่ละประเภท

2) ฟังก์ชันวัตถุประสงค์ (Objective Function) คือ ฟังก์ชันที่เป็นตัวกำหนดค่าเหมาะสมที่สุดครอบคลุมไม่ว่าจะเป็นการหาค่าต่ำสุดหรือสูงสุด ลักษณะของฟังก์ชันวัตถุประสงค์นั้นจะถูกกำหนดให้อยู่ในรูปของสมการทางคณิตศาสตร์โดยมีตัวแปรตัดสินใจเป็นตัวกำหนดค่าของฟังก์ชันในการหาค่าของตัวแปรตัดสินใจที่ทำให้เกิดค่าฟังก์ชันวัตถุประสงค์ที่สูงสุดหรือต่ำสุดนั้น โดยทั่วไปแล้วสามารถเขียนในรูปของสมการทางคณิตศาสตร์ ดังสมการที่ 7 โดย x = {x_1, x_2, ...,x_N} คือ ตัวแปรตัดสินใจทั้งหมดที่เกี่ยวข้องกับฟังก์ชันวัตถุประสงค์มีจำนวนมิติเป็น N เช่นปัญหาการหาค่าเหมาะสมที่สุดในรูปของค่าต่ำสุด คือ การหาต้นทุนที่น้อยที่สุดจากการออกแบบผลิตภัณฑ์ หรือการใช้เวลาที่น้อยที่สุดในการเคลื่อนย้ายสินค้าข้ามทวีป เป็นต้น

3) ข้อจำกัด (Constraints) คือ เงื่อนไขบังคับที่ซึ่งมีผลต่อค่าความเหมาะสมที่สุดของปัญหาอันเกิดมาจากความต้องการตอบสนองความพึงพอใจของตัวปัญหาเอง โดยทั่วไปแล้วข้อจำกัดจะแบ่งเป็น 2 ประเภทด้วยกัน คือ สมการข้อจำกัด และ อสมการข้อจำกัด เป็นดังสมการที่ 7 โดยกลุ่มของผลลัพธ์อยู่บนเงื่อนไขของข้อจำกัดทั้งหลายนั้นเรียกว่ากลุ่มที่เป็นไปได้ (Feasible Set) มีศัพท์เฉพาะทางที่เกี่ยวกับข้อจำกัดดังนี้

ก) ณ จุดที่มีค่าเหมาะสมที่สุดนั้น หากอสมการข้อจำกัดมีการทับซ้อนกันกับสมการข้อจำกัดนั้นจะถูกเรียกว่าเกิดการผูกมัดขึ้น (Binding) เนื่องจากค่าเหมาะสมที่สุดนั้น

จะถูกผูกค่าไว้กับเส้นสมการแล้ว ทำให้จุดที่เหมาะสมที่สุดจะไม่เปลี่ยนแปลงไปตามระนาบของข้อจำกัดอีก

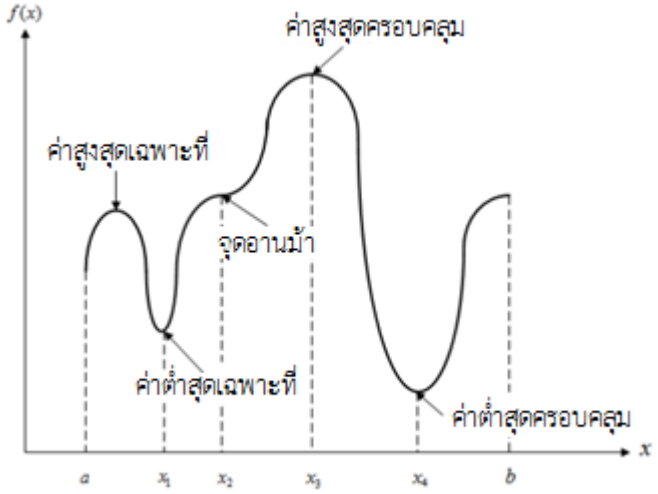
ข) ณ จุดที่มีค่าเหมาะสมที่สุดนั้น หากอสมการข้อจำกัดมีการทับซ้อนกันเฉพาะกับอสมการด้วยกันเอง (ไม่ทับซ้อนกันกับสมการ) ข้อจำกัดนี้จะถูกเรียกว่า ไม่เกิดการผูกมัดขึ้น (Non-binding) และ นั่นเองทำให้จุดที่เหมาะสมที่สุดสามารถเปลี่ยนแปลงได้ตามระนาบของข้อจำกัด ซึ่ง คือ การหาค่าเหมาะสมที่สุดด้วยข้อจำกัดที่ไม่ถูกผูกมัด นี้จะทำให้เกิดผลลัพธ์ที่มีค่าเดียวกันบนหลายๆ จุดของข้อจำกัดที่ไม่ผูกมัด

ค) ถ้าหากข้อจำกัดไม่มีจุดที่ทับซ้อนกันเลย จุดนั้นจะถูกเรียกว่าค่าที่เป็นไปไม่ได้ (Infeasible Values)

ในปัญหาการหาค่าเหมาะสมที่สุดแต่ละปัญหา เช่น ปัญหาการจัดระบบเพื่อให้มีต้นทุนที่น้อยที่สุด นั้นมีจุดประสงค์เพื่อหาค่าที่ต่ำที่สุดเพียงอย่างเดียว แต่หากในปัญหาการหาค่าเหมาะสมที่สุดที่ต้องการทราบค่าสูงสุด ก็จะมีแนวทางในการหาค่าสูงสุดหรือต่ำสุดที่คล้ายกัน ซึ่งทำได้โดยการเปลี่ยนปัญหาการหาค่าต่ำสุดให้เป็นปัญหาการหาค่าสูงสุดด้วยการใส่เครื่องหมายลบ (-) เข้าไปในฟังก์ชันวัตถุประสงค์แล้วทำการหาค่าต่ำสุด แล้วจึงใส่เครื่องหมายลบ (-) ของคำตอบที่ได้จากการหาค่าต่ำสุด ก็จะได้คำตอบที่เป็นค่าสูงสุด ซึ่งเขียนความสัมพันธ์ได้ดังสมการที่ 8

$$\text{maximize } f(x) = -\text{minimize}[-f(x)] \qquad (8)$$

4) คำจำกัดความของค่าสูงสุด/ต่ำสุดเฉพาะที่ และ ค่าสูงสุด/ต่ำสุดครอบคลุม นั้นสามารถอธิบายได้จากภาพที่ 2.3 จุดที่อยู่มีค่าสูงกว่าจุดข้างเคียง จะเรียกว่าค่าสูงสุดเฉพาะที่ และ จุดที่มีค่าสูงที่สุดของค่าสูงสุดเฉพาะที่ทั้งหมดจะเรียกว่าค่าสูงสุดครอบคลุม ในทางกลับกันค่าต่ำสุดครอบคลุมก็เป็นค่าที่อยู่ต่ำที่สุดของค่าต่ำสุดเฉพาะที่ทั้งหมด

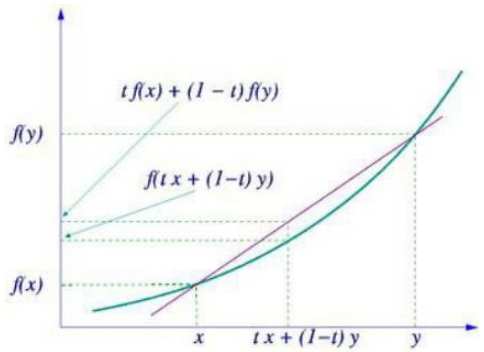


ภาพที่ 2.3 ความสัมพันธ์ของค่าต่ำสุด และ สูงสุด [1]

5) กำหนดการเชิงเส้น (Linear Programming, LP) เป็นกระบวนการทางคณิตศาสตร์เพื่อแก้ปัญหาการหาค่าเหมาะสมที่สุด ภายใต้เงื่อนไขบางประการ โดยแบบจำลองทางคณิตศาสตร์ของทั้งฟังก์ชันวัตถุประสงค์จะต้องแสดงอยู่ในรูปสมการเส้นตรง ส่วนข้อจำกัดอาจอยู่ในรูปสมการหรืออสมการเส้นตรงก็ได้ แต่หากมีฟังก์ชันหรือข้อจำกัดใดที่ไม่เป็นไปตามเงื่อนไขของการเป็นกำหนดการเชิงเส้นแม้เพียงรูปแบบเดียว กำหนดการนั้นจะเรียกว่า กำหนดการไม่เชิงเส้น (Non-linear Programming, NLP) ทันที ยกตัวอย่างรูปแบบที่ไม่เป็นเชิงเส้น ที่มี x เป็นตัวแปรตัดสินใจเพียงตัวเดียว เช่น $x^3 + 3x^2 - 3x + 1$, $\cos x + xe^{-x}$, $\ln x + \tanh 3x$ เป็นต้น

6) ฟังก์ชันคอนเวกซ์ และ ฟังก์ชันเว้า นิยามของฟังก์ชันคอนเวกซ์ คือ ฟังก์ชันที่สอดคล้องกับสมการที่ 9 และ ภาพที่ 2.4 นี้

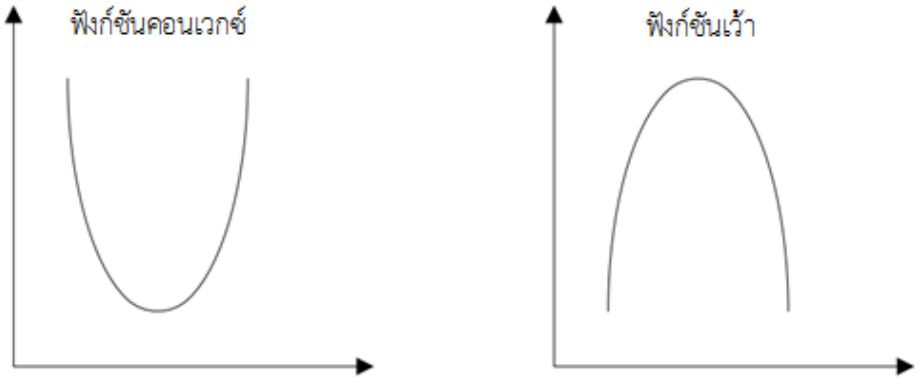
$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \qquad (9)$$



ภาพที่ 2.4 แสดงลักษณะของฟังก์ชันคอนเวกซ์จากสมการที่ 9 [27]

โดย x_1 และ x_2 เป็นจุดใดๆ ที่เป็นสมาชิกของจำนวนจริง $x_1, x_2 \in R$ และ t มีค่าในช่วง $0 \leq t \leq 1$

ในการพิจารณาความเป็นคอนเวกซ์ของฟังก์ชัน ซึ่งจะเป็นฟังก์ชันคอนเวกซ์ได้ก็ต่อเมื่อ เมทริกซ์ของอนุพันธ์อันดับสองของฟังก์ชันมีค่าเป็นบวกเพียงอย่างเดียว หรือมีทั้งบวก และ ลบอย่างละฝั่ง และ ถ้าหากเมทริกซ์ของอนุพันธ์อันดับสองของฟังก์ชันมีค่าเป็นลบเพียงอย่างเดียว หรือเป็นทั้งบวก และ ลบอย่างละฝั่งก็จะเรียกฟังก์ชันนั้นว่ามีลักษณะเป็นเว้า ดังแสดงในภาพที่ 2.5



ภาพที่ 2.5 แสดงลักษณะของฟังก์ชันคอนเวกซ์ และ ฟังก์ชันเว้า [1]

คุณสมบัติของฟังก์ชันคอนเวกซ์

1) ทำให้เกิดจุดต่ำสุดในฟังก์ชัน

2) ค่าลักษณะเฉพาะ (Eigenvalue) ของเมทริกซ์อนุพันธ์อันดับสองน้อยกว่าเท่ากับ 0

คุณสมบัติของฟังก์ชันเว้า

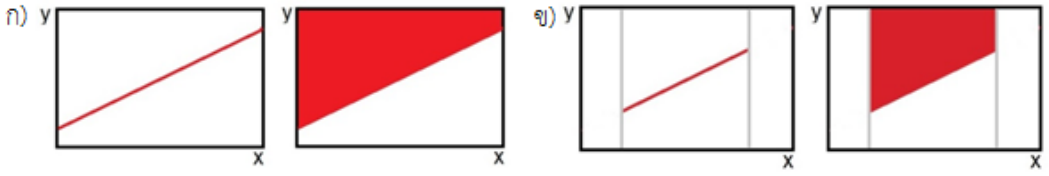
1) ทำให้เกิดจุดสูงสุดในฟังก์ชัน

2) ค่าลักษณะเฉพาะ (Eigenvalue) ของเมตริกซ์อนุพันธ์อันดับสองมากกว่าเท่ากับ 0

หากมีฟังก์ชันใดที่ไม่เป็นไปตามคุณสมบัติของฟังก์ชันคอนเวกซ์หรือเว้า และ ไม่เป็นเส้นตรง ฟังก์ชันเหล่านั้นจะเรียกว่าเป็นฟังก์ชันที่ไม่คอนเวกซ์ (Non-convex) ทั้งนี้ฟังก์ชันที่มีลักษณะไม่คอนเวกซ์ จะมีความยากต่อการแก้ปัญหามากกว่าฟังก์ชันคอนเวกซ์ เนื่องจากฟังก์ชันไม่คอนเวกซ์จะมีความไม่ต่อเนื่องของค่าของตัวแปรด้วย

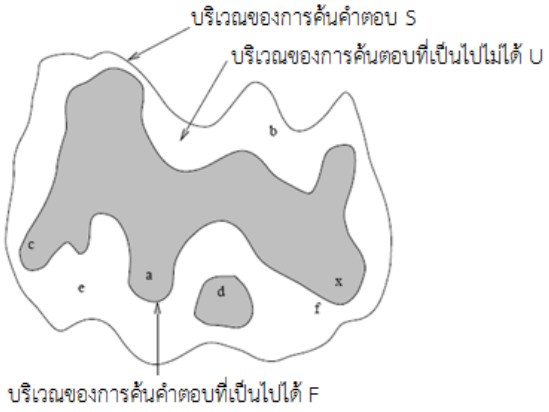
2.1.3 วิธีจัดการข้อจำกัด

ความซับซ้อนของปัญหาแต่ละแบบสามารถมองให้อยู่ในรูปของบริเวณของการค้นคำตอบได้ โดยบริเวณของการค้นคำตอบขึ้นอยู่กับความเป็นสมการข้อจำกัด อสมการข้อจำกัด การจำกัดขอบเขตบน และ ล่าง (Upper & Lower Bound Limitations) ของตัวแปรตัดสินใจ และ การกำหนดบริเวณคำตอบที่เป็นไปได้ (Feasible Region) จากภาพที่ 2.6 ก) แสดงให้เห็นถึงบริเวณการค้นคำตอบของสมการข้อจำกัด และ อสมการข้อจำกัด ภาพที่ 2.6 ข) แสดงสมการข้อจำกัด และ อสมการข้อจำกัดเมื่อถูกจำกัดขอบเขตบน และ ล่างดังนี้



ภาพที่ 2.6 แสดงบริเวณการค้นหาของสมการ และ อสมการข้อจำกัด โดยที่ ก) และ ข) แสดงบริเวณการค้นหาที่ไม่ถูกจำกัดขอบเขต และ ถูกจำกัดขอบเขตแล้วตามลำดับ

ในกรณีที่มีข้อจำกัดหลายตัว ซึ่งแต่ละตัวก็จะมีบริเวณของการค้นคำตอบเฉพาะตัวเอง และเมื่อข้อจำกัดมีความเชื่อมโยงกัน ก็จะทำให้มีบางบริเวณของการค้นคำตอบในแต่ละข้อจำกัดที่มีบริเวณของการค้นคำตอบซ้อนทับกันอยู่ จึงได้กำหนดส่วนที่ซ้อนทับกันนี้ ว่าเป็นบริเวณของคำตอบที่เป็นไปได้ (Feasible Region) และ บริเวณของการค้นคำตอบของข้อจำกัดที่ไม่ซ้อนทับกันจะเรียกว่าบริเวณของคำตอบที่เป็นไปไม่ได้ (Unfeasible Region) เมื่อนำองค์ประกอบทั้งหมดของบริเวณในการหาคำตอบมารวมกันจะได้ดังภาพที่ 2.7



ภาพที่ 2.7 แสดงบริเวณของการค้นคำตอบ และ บริเวณที่มีความเป็นไปได้ [28]

จากภาพแสดงบริเวณในการหาคำตอบโดยมีบริเวณสีขาวรอบนอกแสดงเป็นบริเวณในการหาคำตอบซึ่งเป็นไปไม่ได้ ส่วนบริเวณสีเทาจะเป็นบริเวณในการหาคำตอบที่เป็นไปได้ แต่อย่างไรก็ตามเป้าหมายหลักของการหาคำตอบไม่ใช่การหาค่าที่อยู่บนบริเวณที่เป็นไปได้เพียงอย่างเดียว คำตอบที่ได้จะต้องให้ค่าที่ใกล้เคียงกับค่าเหมาะสุดครอบคลุมมากที่สุด สิ่งที่ต้องพิจารณาเป็นอันดับถัดไป คือ อัลกอริทึมเชิงวิวัฒน์ในรอบปัจจุบันได้คำตอบตัวแปรทั้งหมดติดที่ค่าเหมาะสมเฉพาะที่ ซึ่งจะต้องทำการคัดเลือกประชากรตัวแปรที่ดีเพื่อนำไปปรับคำตอบในรอบถัดไป แล้วถ้าเกิดสมมติคำตอบที่เป็นค่าเหมาะสมครอบคลุมอยู่ที่ค่า x ซึ่งต้องคัดเลือกประชากรตัวใดตัวหนึ่งที่ให้คำตอบที่ a และ f เมื่อเกิดกรณีนี้ควรเลือกประชากรตัวใด ตัวที่อยู่ใน F แต่คำตอบห่างจากตำแหน่ง x มากกว่าที่ตำแหน่ง f แต่ที่ตำแหน่ง f กลับอยู่ใน U ดังนั้นปัญหา คือ ประชากรตัวใดควรจะถูกเลือก

ถ้าจากที่เห็นก็ควรจะเลือกประชากรที่ให้คำตอบที่ตำแหน่ง f เพราะให้คำตอบใกล้ค่าเหมาะสมครอบคลุมมากกว่า แต่ในความเป็นจริงเราไม่สามารถเป็นบริเวณของการค้นคำตอบได้ดังรูปนี้ ฉะนั้นจึงไม่สามารถบอกได้ว่าจุด f ดีกว่าจุด a ด้วยเหตุนี้ การคัดเลือกประชากรจึงเลือกตัวที่ให้คำตอบที่จุด a ไปเพราะว่าเป็นค่าที่อยู่ในบริเวณที่เป็นไปได้นั่นเอง ซึ่งวิธีที่กล่าวมานั้นถือเป็นหนึ่งในวิธีจัดการข้อจำกัด และ ยังมีจัดการประเภทอื่นอีก ดังต่อไปนี้

1) ฟังก์ชันการลงโทษ (Penalty Function Method)

ฟังก์ชันการลงโทษเป็นวิธีการแปลงจากปัญหาเชิงข้อจำกัดให้ไม่มีข้อจำกัด และ นำไปรวมเข้ากับฟังก์ชันวัตถุประสงค์ในเทอมของการลงโทษ ฟังก์ชันวัตถุประสงค์ที่มีผลลัพธ์ที่เป็นไปไม่ได้ก็เป็นถูกทำโทษเพื่อลดค่าของข้อจำกัดส่วนที่ได้เกินออกมา ซึ่งมีกฎในการลงโทษดังนี้

กฎการลงโทษในการหาค่าให้ต่ำที่สุด

ก) ค่าการลงโทษที่คำนวณได้ควรทำให้มีค่าน้อยที่สุดเท่าที่เป็นไปได้ นั่นหมายความว่าเป็นการบีบขอบเขตของผลลัพธ์ที่เป็นไปไม่ได้ (infeasible solution) ซึ่งทำให้คำตอบเข้าใกล้ค่าเหมาะสมที่สุดมากขึ้น

ข) ค่าการลงโทษที่ยังสูงเกินไป และ ค่านั้นเข้าใกล้บนเส้นขอบของบริเวณที่เป็นไปได้ (Boudary of Feasible Region) GA จะทำการผลักตัวแปรนั้นให้เข้าสู่บริเวณที่เป็นไปได้โดยทันที และ ตัวแปรนั้นจะไม่เคลื่อนออกไปบริเวณที่เป็นไปไม่อีก ถ้าในบริเวณการค้นคำตอบทั้งหมดมีบริเวณที่เป็นไปได้หลายๆ จุดที่ไม่เชื่อมต่อกัน GA จะมีแนวโน้มที่

จะผลักตัวแปรนั้นบนบริเวณที่เป็นไปได้เดียว และ จะไม่เคลื่อนออกไปยังบริเวณที่เหมาะมที่ต่างกันไป

ค) ค่าการลงโทษที่ต่ำเกินไปทำให้บริเวณที่เป็นไปได้มีน้อยมาก จนส่งผลให้การคำนวณจะใช้เวลาส่วนใหญ่ไปกับการหนีจากบริเวณที่ไม่เหมาะสมเท่านั้น โดยไม่สนใจความเหมาะสมของฟังก์ชันวัตถุประสงค์เท่าที่ควร

การทำโทษค่าวัตถุประสงค์นั้นมีหลายประเภทดังนี้

1.1) รูปแบบทั่วไปของฟังก์ชันลงโทษ [11]

$$Fitness(x) = F(x) + \sum_{l=1}^{L} w_l \max(0, g_l(x)) + \sum_{m=1}^{M} w_m |h_m(x)| + \sum_{n=1}^{N} w_n |\max(0, x_n^{(L)} - x_n) + \max(0, x_n - x_n^{(U)})| \quad (10)$$

โดย w_l, w_m และ w_n คือ ตัวแปรของการลงโทษ $g_l(x)$ คือ อสมการข้อจำกัดตัวที่ l จากทั้งหมด L ตัว $h_m(x)$ คือ สมการข้อจำกัดตัวที่ m จากทั้งหมด M ตัว $x_n^{(L)}$ และ $x_n^{(U)}$ คือ ขอบเขตล่าง และ ขอบเขตบนของตัวแปรตัดสินใจตัวที่ n จากทั้งหมด N ตัวตามลำดับ

1.2) การลงโทษสถิต (Static Penalties)

ปัจจัยในการทำโทษจะมีค่าคงตลอดกระบวนการ

$$Fitness(x) = F(x) \pm \sum_{i=1}^{L} (w_l \max[0, g_l(x)]^2) \quad (11)$$

โดย w_l คือ ตัวแปรของการลงโทษ และ $g_l(x)$ คือ อสมการข้อจำกัดตัวที่ l จากทั้งหมด L ตัว

1.3) การลงโทษพลวัต (Dynamic Penalties) [29]

ค่าการทำโทษจะมีค่าไม่คงที่ ซึ่งในแต่ละรอบ ค่าการทำโทษจะถูกคำนวณด้วยค่าที่เกี่ยวข้องกันในรอบปัจจุบัน

$$Fitness(x) = F(x) \pm \sum_{l=1}^{L} ((w_l t)^{\alpha} \max[0, g_l(x)]^2) \quad (12)$$

โดย w_l คือ ตัวแปรการลงโทษ $g_l(x)$ คือ อสมการข้อจำกัดตัวที่ l จากทั้งหมด L ตัว t คือ ระยะทางที่มีพลวัตของวิธีลงโทษตามความยาวในการค้นคำตอบ และ α คือ ค่าคงที่อยู่ระหว่าง 1 และ 2

1.4) การลงโทษแบบอบอ่อน (Annealing Penalties)

ค่าการทำโทษจะถูกปรับก็ต่อเมื่อในเกิดการวนซ้ำหลายๆ รอบ ณ จุดที่คาดว่าจะเป็นการติดกับที่ค่าเหมาะสมที่สุดเฉพาะที่ โดยจะมีสมการการปรับค่าทำโทษดังนี้

$$\text{Fitness}(x) = F(x) \pm \sum_{l=1}^{L} \max[0, g_l(x)]^2 / \tau \qquad (13)$$

โดย w_l คือ ตัวแปรของการลงโทษ $g_l(x)$ คือ อสมการข้อจำกัดตัวที่ l จากทั้งหมด L ตัว และ τ คือ กำหนดการในการเย็นตัว (Cooling Schedule)

1.5) การลงโทษแบบปรับได้ (Adaptive penalties)

ใช้ฟังก์ชันการลงโทษซึ่งให้ดูจากผลของกระบวนการค้นคำตอบ

$$\text{Fitness}(x) = F(x) \pm \lambda(i) \sum_{l=1}^{L} \max[0, g_l(x)]^2 \qquad (14)$$

โดย w_l คือ ตัวแปรของการลงโทษ $g_l(x)$ คือ อสมการข้อจำกัดตัวที่ l จากทั้งหมด L ตัว และ $\lambda(i)$ จะถูกอัพเดททุกๆ รอบตามเงื่อนไขดังนี้

ถ้าประชากรที่ดีที่สุดในรอบที่ i อยู่ในบริเวณที่เป็นไปได้ : $\lambda(i+1) = \lambda(i)/\alpha; \alpha > 1$ (14.1)

ถ้าประชากรที่ดีที่สุดในรอบที่ i ไม่อยู่ในบริเวณที่เป็นไปได้ : $\lambda(i+1)(i) = \beta\lambda(i); \beta > 1$ (14.2)

ไม่เช่นนั้น หากไม่มีเงื่อนไขนี้จะกลายเป็นว่ามีประชากรบางตัวที่อยู่บนบริเวณที่เป็นไปได้ และเป็นไปไม่ได้นั้นย่อมให้ค่าที่สุดอยู่เสมอจนทำให้ค่าการลงโทษไม่เปลี่ยนไป

1.6) การลงโทษด้วยการกำจัด (Death Penalty)

การกำจัดประชากรที่อยู่บนบริเวณที่เป็นไปไม่ได้ออกไป วิธีการนี้เป็นการจัดการข้อจำกัดที่ง่ายที่สุด

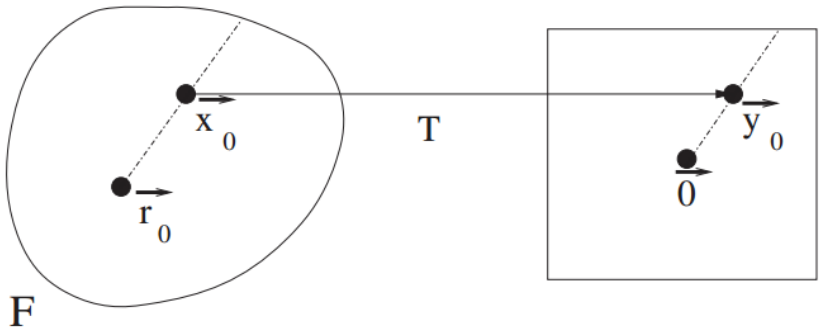
2) ตัวดำเนินการ และ การนำเสนอแบบพิเศษ (Special Representations and Operators)

มีนักวิจัยพัฒนารูปแบบของการนำเสนอที่พิเศษขึ้นมาเพื่อจัดการกับปัญหาที่มีความซับซ้อน ซึ่งรูปแบบของการนำเสนอเชิงวิวัฒน์ (Generic Representation Scheme) จำเป็นที่จะต้องสร้างตัวดำเนินการเชิงวิวัฒน์ที่ใช้งานได้ในแนวทางเดิมมากกว่าการใช้งานตัวดำเนินการแบบเก่าที่จะนำเสนอเป็นแนวทางต่างจากเดิม เช่น ระบบเลขฐานสอง

วิธีการนี้มีทั้งการกลายพันธุ์ การสลับสายพันธุกรรม และ การถอดรหัส ซึ่งส่วนที่น่าสนใจของจัดการข้อจำกัดประเภทนี้ คือ การถอดรหัส ส่วนสำคัญของการถอดรหัส คือ การจัดการกับโครโมโซมที่อยู่ในบริเวณที่เป็นไปไม่ได้ (Infeasible Region) ไปสู่บริเวณที่เป็นไปได้ (Feasible Region) ในปัญหาที่ต้องการหาคำตอบ ในบางกรณี ตัวดำเนินการพิเศษนี้ยังออกแบบมาเพื่อสร้างโครโมโซมลูกหลานให้มีค่าอยู่ระหว่างบริเวณที่เป็นไปได้กับเป็นไปไม่ได้อีกด้วย

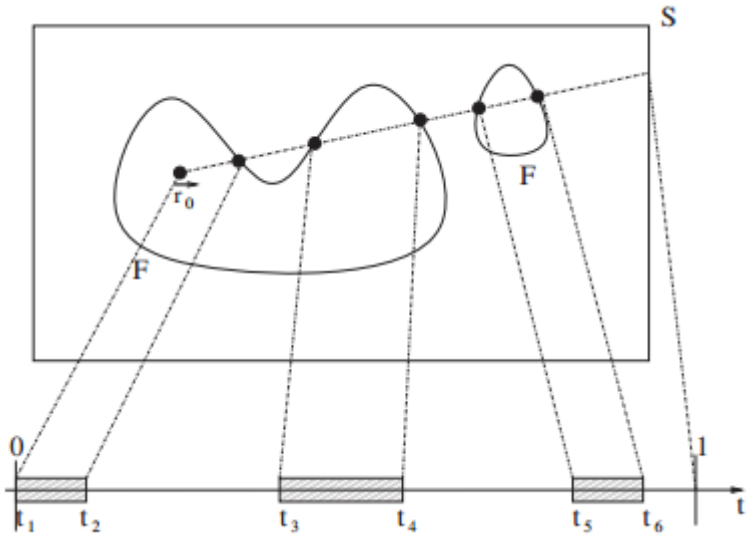
2.1) Homomorphous Maps (HM)

มีแนวคิดในการแปลงบริเวณที่เป็นไปได้ทั้งหมดให้มีรูปร่างที่ต่างออกไปส่วนที่สำคัญที่สุดของการวิธีการนี้ คือ homomorphous maps [30] วิธีการนี้เป็นการเปรียบเทียบกันระหว่าง ลูกบาศก์ n-มิติ กับบริเวณการค้นทั้งหมดที่เป็นบริเวณที่เป็นไปได้ ไม่ว่าจะเป็นคอนเวกซ์ หรือไม่เป็นคอนเวกซ์ก็ตาม แนวคิดหลัก คือ การแปลงปัญหาต้นแบบให้กลายเป็นฟังก์ชันอื่น (การสมมูลเชิงทอพอโลยี) ซึ่งง่ายต่อการหาค่าเหมาะสมที่สุดด้วยอัลกอริทึมเชิงวิวัฒน์กรณีของรูปร่างที่เป็นคอนเวกซ์แสดงดังภาพที่ 2.8 ซึ่งช่วงของการทำ homomorphous map มีความต่อเนื่องกันตลอดขอบเขตของคำตอบ เนื่องจากรูปร่างของปัญหาแบบเป็นคอนเวกซ์มีความต่อเนื่องกัน



ภาพที่ 2.8 แสดงลักษณะการทำ homomorphous map กับรูปร่างปัญหาที่เป็นคอนเวกซ์

กรณีของรูปร่างที่ไม่เป็นคอนเวกซ์แสดงดังภาพที่ 2.9 ซึ่งช่วงของการทำ homomorphous map จะถูกแบ่งเป็นช่วงๆ เนื่องจากรูปร่างปัญหาแบบไม่เป็นคอนเวกซ์มีความไม่ต่อเนื่องกัน



ภาพที่ 2.9 แสดงลักษณะการทำ homomorphous map กับรูปร่างปัญหาที่ไม่เป็นคอนเวกซ์

Homomorphous Maps (HM) ถูกนำไปใช้แข่งขันกับวิธีที่อื่นเป็นอย่างมาก อย่างไรก็ตาม การประยุกต์ใช้อัลกอริทึมก็มีความซับซ้อนมาก และ ยังให้ผลการทดลองว่าต้องใช้จำนวนฟังก์ชันในการประเมินที่สูง วิธีการนี้ไม่เหมือนกับฟังก์ชันลงโทษ (Penalty functions) ซึ่งเป็นการรวมเอาฟังก์ชันวัตถุประสงค์กับข้อจำกัดของปัญหาในการหาค่าความเหมาะสม ส่วนวิธีนี้จะเป็นการจัดการข้อจำกัดกับวัตถุประสงค์แยกจากกัน

2.2) การจัดการจำนวนเต็ม และ ตัวแปรที่ไม่ต่อเนื่อง (Handling Integer and Discrete Variables)

ขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างต้นแบบนั้นไม่เหมาะกับจัดการตัวแปรที่ไม่ต่อเนื่องได้ อย่างไรก็ตามการปรับปรุงขั้นตอนวิธีให้สามารถใช้กับตัวแปรจำนวนเต็ม และ /หรือตัวแปรที่ไม่ต่อเนื่องก็สามารถทำได้ไม่ยาก อันดับแรกตัวแปรที่ต่อเนื่องจะถูกแปลงเป็นตัวแปรจำนวนเต็มด้วยการตัดเป็นท่อนๆ จากนั้นจึงสามารถนำตัวแปรที่ถูกแปลงไปใช้ในการประเมินค่าวัตถุประสงค์ได้ต่อไป

ส่วนตัวแปรที่ไม่ต่อเนื่องนั้นก็สามารถจัดการได้ไม่ยาก ด้วยการสร้างดัชนีของค่าที่กำหนดไว้เทียบได้กับค่าตัวแปรที่ไม่ต่อเนื่อง แล้วจึงนำตัวแปรที่ไม่ต่อเนื่องนี้ไปเปรียบเทียบกับค่าที่ได้มา ด้วยเหตุนี้จึงสามารถประเมินฟังก์ชันวัตถุประสงค์จากตัวแปรที่ไม่ต่อเนื่องได้

2.3) การจัดการขอบเขตของข้อจำกัด (Handling Boundary Constraints) [31]

การจัดการขอบเขตของข้อจำกัดให้อยู่ภายในช่วงที่กำหนดเป็นเรื่องสำคัญ หากตัวแปรใดที่มีสมาชิกบางตัวที่มีค่าเกินขอบเขตที่กำหนดไว้ไม่ว่าจะค่าขอบเขตล่างหรือขอบเขตบนก็ตาม สมาชิกตัวนั้นจะถูกแทนที่ด้วยขอบเขตที่มันเกินไป ซึ่งแสดงความสัมพันธ์ได้ดังสมการที่ 15

$$x_{i,j} = \begin{cases} x_{i,j}^{(L)} & \text{ถ้า } x_{i,j} < x_{i,j}^{(L)} \\ x_{i,j} & \text{ถ้า } x_{i,j}^{(L)} \le x_{i,j} \le x_{i,j}^{(U)} \\ x_{i,j}^{(U)} & \text{ถ้า } x_{i,j} > x_{i,j}^{(U)} \end{cases} \qquad (15)$$

โดย $x_{i,j}^{(L)}$ และ $x_{i,j}^{(U)}$ คือ ขอบเขตล่าง และ ขอบเขตบนของแต่ละประชากรที่ i ของตัวแปรตัดสินใจตัวที่ j และ ตามลำดับ

2.4) การยืดหยุ่นข้อจำกัดที่ปรับได้ (Constraint Relaxation) [20]

วิธีจัดการข้อจำกัดนี้สามารถจัดการได้ทั้งสมการ และ อสมการข้อจำกัด สำหรับใช้กับขั้นตอนวิธีประเภทสโทแคสติกใดๆ ซึ่งเป็นการทำงานร่วมกันกับความยืดหยุ่นของข้อจำกัดที่ปรับตัวได้กับค่าที่ประเมินแล้วว่าเหมาะสม ในช่วงแรกจะทำการคำนวณค่าฟังก์ชันวัตถุประสงค์ และข้อจำกัดของประชากรทั้งหมด จากนั้นจึงหาผลรวมของค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกันของประชากรตัวที่ k (TAV_k) จนถึงประชากรทั้งหมด (NP) ดังสมการที่ 16 จากนั้นจึงหาค่ามัธยฐานจาก TAV_k จากประชากรทุกตัวแล้วตั้งชื่อว่าเป็นค่าความยืดหยุ่นเริ่มต้นของข้อจำกัด (μ)

$$TAV_k = \sum_{m=1}^{M} |h_m(x)| \sum_{l=1}^{L} \max(0, g_l(x)) \qquad k = 1, 2, \ldots, NP \qquad (16)$$

ถ้าหากผลรวมของค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกัน (Total Absolute Violation) ของประชากรตัวที่มีค่าน้อยกว่า μ ประชากรดังกล่าวจะถูกกำหนดให้เป็นผลลัพธ์ที่

เป็นไปได้ หากไม่เป็นเช่นนั้นประชากรก็จะถูกกำหนดให้เป็นผลลัพธ์ที่เป็นไปไม่ได้ ความยืดหยุ่นนี้เหมาะใช้งานร่วมกันทั้งกับสมการ และ อสมการข้อจำกัด เมื่อทราบประชากรที่เป็นไปได้ และ ไม่เป็นไปได้แล้ว พารามิเตอร์ μ จะถูกทำให้ลดลงในรุ่นถัดไป (i+1) ด้วยสัดส่วนของประชากรที่เหมาะสม (F_F) ภายใต้ความยืดหยุ่นของข้อจำกัดในรุ่นปัจจุบัน (i) ดังสมการที่ 17

$$\mu_{i+1} = \mu_i\left(1 - \frac{F_F}{NP}\right) \tag{17}$$

3) ขั้นตอนวิธีซ่อมแซม (Repair Algorithm) [15]

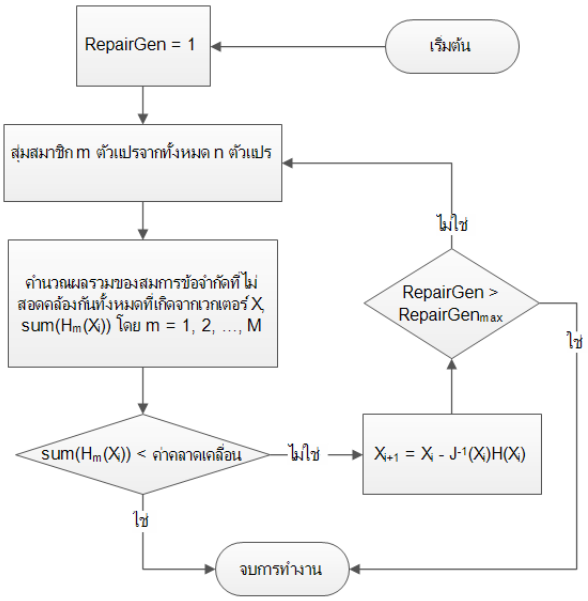
โดยปกติสมการข้อจำกัดสามารถใช้ลดจำนวนมิติของปัญหาโดยไม่ทำให้ผลลัพธ์ผิดเพี้ยนแต่อย่างใด อย่างไรก็ตามการระบุตัวแปรที่จะลดนั้นก็เป็นงานที่ยาก อีกทั้งบางครั้งก็ไม่สามารถลดได้

พิจารณากรณีของปัญหาการหาค่าเหมาะสมที่สุดขนาด N-มิติ กับสมการข้อจำกัดขนาด M-มิติ โดยที่ N > M นั่น คือ มีตัวแปรอยู่เพียง N – M ตัวเท่านั้นที่เป็นอิสระต่อกัน ในขณะที่ตัวแปรที่นิยามมาจากสมการข้อจำกัดนั้นมีทั้งสิ้น M ตัว นั่นหมายความว่าตัวแปรบางตัวจาก N ตัวแปรของเวกเตอร์ X สามารถถูกซ่อมแซมด้วยระบบการแก้สมการได้ทั้งสิ้น M สมการ ในที่นี้ได้ประยุกต์วิธีของ “นิวตันราฟสัน (Newton Raphson)” มาใช้กับระบบการแก้สมการข้อจำกัด

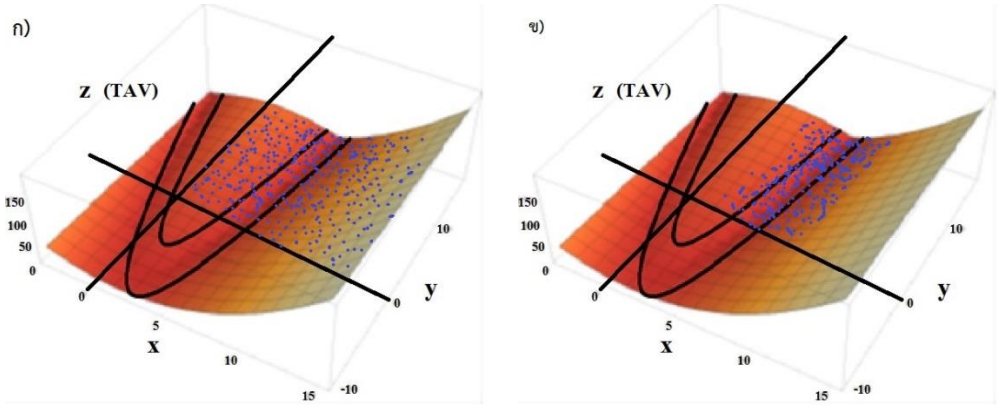
ผังการทำงานของวิธีจัดการสมการข้อจำกัดแสดงดังภาพที่ 2.10 ขั้นตอนแรก คือ การนำเวกเตอร์แต่ละตัวมาคำนวณหาค่าข้อจำกัด โดยค่าของข้อจำกัดใดที่ให้ค่ามากกว่าค่าความคลาดเคลื่อนยินยอมที่กำหนดไว้ (Tolerance, $\boldsymbol{\varepsilon}$) จะถือว่าเป็นข้อจำกัดที่ไม่สอดคล้องกัน และเวกเตอร์นั้นจะเป็นเวกเตอร์ที่เป็นไปไม่ได้ เวกเตอร์ที่เป็นไปไม่ได้ที่มีค่าข้อจำกัดที่ไม่สอดคล้องกัน (Constraints Violation) ไม่มาก สามารถอนุโลมให้ผ่านไปได้ นั่นถือเป็นการคงไว้ซึ่งความหลากหลายของประชากร แต่ในทางกลับกันเวกเตอร์ที่เป็นไปไม่ได้ที่มีค่าข้อจำกัดที่ไม่สอดคล้องกันมากกว่าค่าคลาดเคลื่อนยินยอม จะถูกส่งไปซ่อมแซม ซึ่งจะทำการสุ่มตัวแปรจำนวน M ตัวจากทั้งหมด N ตัวเพื่อมาทำการซ่อมแซมตามสมการที่ 18 ผลลัพธ์ที่ได้จะทำให้เวกเตอร์ X เคลื่อนที่เข้าใกล้ค่าของข้อจำกัดในช่วงที่อนุโลมของแต่สมการข้อจำกัดนั้นๆ ตัวอย่างแสดงการเคลื่อนที่เข้าของเวกเตอร์ที่ถูกซ่อมแซมเป็นดังภาพที่ 2.11 [32]

$$X_{i+1} = X_i - J^{-1}(X_i)H(X_i) \tag{18}$$

โดย $J^{-1}(X_i)$ คือ เมทริกซ์จาโคเบียนผกผัน (Jacobian Matrix) และ $H(X_i)$ คือ เวกเตอร์ของสมการข้อจำกัดที่มีค่าเกินออกมา ในการวนซ้ำนั้นจะหยุดลงเมื่อมีค่าข้อจำกัดที่ไม่สอดคล้องกันน้อยกว่าค่าความคลาดเคลื่อนยินยอมที่กำหนดไว้ **ε** หรือวนครบตามจำนวนสูงสุดที่ตั้งเอาไว้นั่นเอง



ภาพที่ 2.10 แสดงผังการทำงานของวิธีจัดการสมการข้อจำกัด



ภาพที่ 2.11 การกระจายของเวกเตอร์ประชากรก่อน และ หลังกระบวนการซ่อมแซม (ก) ก่อนเริ่มทำการซ่อมแซม (ข) หลังจากซ่อมแซมแล้ว

วิธีการนี้มีจุดประสงค์เพื่อซ่อมแซมประชากรที่เป็นไปไม่ได้ ให้กลายเป็นประชากรที่เป็นไปได้ ทั้งนี้วิธีการซ่อมแซมนั้น ไม่มีฮิวริสติกมาตรฐาน ซึ่งความสำเร็จของวิธีนี้ขึ้นอยู่กับความสามารถของในการเลือกใช้ฮิวริสติกได้เหมาะสมกับปัญหามากแค่ไหน

4) จัดการแยกวัตถุประสงค์ และ ข้อจำกัด (Separation of Objectives and Constraints)

ใช้สำหรับการจัดการข้อจำกัด และ วัตถุประสงค์ที่มีความแตกต่างกัน โดยสิ่งที่ควรทราบเบื้องต้นของการประเมินระหว่างผลลัพธ์ที่เป็นไปได้กับผลลัพธ์ที่เป็นไปไม่ได้ผลลัพธ์ใดดีกว่ากัน เช่น หากมีประชากรจากจุดหนึ่งที่มีค่าอยู่ในจุดที่อนุโลมให้ใช้ได้แต่ผลลัพธ์จากจุดนั้นอยู่ห่างจากค่าเหมาะสมครอบคลุมมากกว่าอีกประชากรอีกจุดหนึ่งซึ่งมีค่าใกล้เคียงกับค่าเหมาะสมครอบคลุมมาก แต่ที่จุดนั้นมีค่าที่ไม่อนุโลมให้ใช้ได้ เมื่อเกิดกรณีตามหลักการของวิธีนี้ย่อมที่จะเลือกประชากรตัวที่อยู่บนจุดที่ให้ค่าที่อนุโลมใช้ได้ก่อนเสมอ โดยจะมองว่าผลลัพธ์ที่อนุโลมให้ใช้ได้นั้นถือเป็นช่วงที่ดี และ ผลลัพธ์ที่ไม่อนุโลมให้ใช้ได้นั้นอยู่ในช่วงที่ไม่ดี ซึ่งผลลัพธ์ของทั้งสองจุดจะถูกพิจารณาแล้วคัดเลือกผู้ชนะไปยังรอบถัดไป ดังนี้เช่นรูปแบบการคัดเลือกตามความเด่น (Dominance-based Selection Scheme)

รูปแบบคัดเลือกความเด่น (Dominance-based Selection Scheme) [16]

รูปแบบการคัดเลือกแบบความเด่น นั้นถูกประยุกต์ใช้ร่วมกับข้อจำกัดในส่วนของความเหมาะสมฟังก์ชัน (Fitness Function) เมื่อเกิดการเปรียบเทียบระหว่างเวกเตอร์ทดสอบ U_i กับเวกเตอร์พ่อ X_i คาดว่าจะมีเหตุการณ์ที่เป็นไปได้เกิดขึ้นอยู่ 3 กรณี ซึ่งกรณีที่ 1 คือ ทั้ง U_i และ X_i ต่างก็เป็นเวกเตอร์ทีเป็นไปได้ ซึ่งเวกเตอร์ใดที่ให้ค่าของฟังก์ชันวัตถุประสงค์ที่ดีกว่าเวกเตอร์นั้นก็จะถูกเลือกใช้ในรุ่นถัดไป กรณีที่ 2 เวกเตอร์หนึ่งเป็นเวกเตอร์ที่เหมาะสม ส่วนอีกเวกเตอร์เป็นเวกเตอร์ที่เป็นไปไม่ได้ เวกเตอร์เหมาะสมจะถูกเลือกใช้ในรุ่นถัดไป กรณีที่ 3 ทั้ง U_i และ X_i ต่างก็เป็นเวกเตอร์เป็นไปไม่ได้ ซึ่งเวกเตอร์ที่ให้ค่าของข้อจำกัดที่ไม่สอดคล้องกันต่ำกว่าจะเป็นตัวที่ถูกเลือกใช้ในรุ่นถัดไป ซึ่งความสัมพันธ์ในการเลือกถูกนิยามดังสมการที่ 15

$$X_{i+1} = \begin{cases} X_i & \text{ถ้า } X_i \prec U_i \\ U_i & \text{ถ้าไม่ใช่} \end{cases} \qquad (19)$$

โดย $X_i \prec U_i$ หมายความว่า X_i เหนือว่า U_i ซึ่ง คือ X_i มีค่าฟังก์ชันวัตถุประสงค์ที่ดีกว่า U_i และ / หรือ มีค่าของข้อจำกัดที่ไม่สอดคล้องกันที่ต่ำกว่า

5) วิธีผสม (Hybrid Methods)

วิธีผสม คือ การนำวิธีที่กล่าวมาทั้งหมดข้างต้นมาใช้ร่วมกัน โดยไม่มีกฎตายตัวในการใช้งานร่วมกันของวิธีการต่างๆ ซึ่งการเลือกใช้ขึ้นอยู่กับลักษณะการนำไปใช้งานในปัญหาที่ต้องการ

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 งานวิจัยของ Srinivas และ Rangaiah [33]

ได้นำเสนอวิธี DETL คือ การนำขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (Differential Evolution, DE) ซึ่งมีคุณสมบัติที่ดีในการจัดการฟังก์ชันไม่เชิงเส้น ฟังก์ชันที่ไม่สามารถหาอนุพันธ์ได้ และ ฟังก์ชันไม่เป็นคอนเวกซ์ มาใช้ร่วมกับการเก็บรายการแบบทาบู เพื่อนำมาใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดของทางวิศวกรรมเคมีซึ่งมีลักษณะปัญหากำหนดการไม่เชิงเส้น และ กำหนดการไม่เชิงเส้นที่มีจำนวนเต็มผสม ด้วยแนวคิดของการนำการค้นแบบทาบูซึ่งมีคุณสมบัติของการหลีกเลี่ยงการกลับไปค้นซ้ำที่ตำแหน่งเดิมเมื่อนำมาประยุกต์ใช้ร่วมกับขั้นตอนวิธีหลักอย่างวิธีเชิงวิวัฒน์แบบส่วนต่างเพื่อเพิ่มประสิทธิภาพการทำงานด้านการคำนวณ DETL ได้มีการนำไปทดลองกับปัญหากำหนดการไม่เชิงเส้นมากมายในช่วงตัวแปร 2-13 และ มีข้อจำกัดสูงถึง 38 ฟังก์ชัน อีกทั้งยังทดสอบกับปัญหากำหนดการไม่เชิงเส้นที่มีจำนวนเต็มผสมอยู่หลายปัญหา ผลลัพธ์ถูกนำมาเปรียบเทียบกับของวิธี DETL DE และ วิธีเชิงวิวัฒน์แบบส่วนต่างที่ปรับปรุงแล้ว (MDE) บนปัญหา NLPs และ MINLPs ผลลัทธ์ที่ได้ คือ ค่าเฉลี่ยของจำนวนฟังก์ชันที่ใช้ประเมิน (NFE) ของวิธี DETL น้อยกว่าทั้ง DE และ MDE

2.2.2 งานวิจัยของ Shang, Zhao และ Shen [18]

เป็นวิธีผสมที่มีชื่อว่าค่าคลาดเคลื่อนยินยอมที่ยืดหยุ่นได้ในขั้นตอนวิธีทางพันธุกรรม (flexible tolerance genetic algorithm, FTGA) ซึ่งงานวิจัยนี้นำเสนอมาเพื่อแก้ปัญหาการหาค่าเหมาะสมที่สุดกลุ่มไม่เชิงเส้น ปัญหาการหาค่าเหมาะสมที่สุดที่มีคำตอบหลายค่า และ ปัญหาการหาค่าเหมาะสมที่สุดที่มีข้อจำกัดหลายตัว ซึ่งวิธีที่นำเสนอนี้จัดเป็นวิธีผสมแบบใหม่ที่นำเอาวิธีการทำค่าคลาดเคลื่อนยินยอมที่ยืดหยุ่นได้ (flexible tolerance method, FTM) รวมเข้ากันกับขั้นตอนวิธีทางพันธุกรรมซึ่งปรับตัวได้ (adaptive genetic algorithm, AGA) โดย AGA ทำหน้าที่สร้างประชากรรุ่นแรก และ ทำการลงตำแหน่งที่เป็นประชากรที่ดีที่สุดไว้ ส่วน FTM เป็นส่วนเสริมซึ่งทำ

หน้าที่เป็นตัวดำเนินการหนึ่งของ AGA ซึ่งมีผลให้เกิดการก้าวข้ามจากประชากรรอบข้าง ด้วยกลไกการค้น และ ปรับลดค่าข้อจำกัดที่ไม่สอดคล้องกันของฟังก์ชันวัตถุประสงค์ ด้วยการสร้างค่าคลาดเคลื่อนยินยอมที่ยืดหยุ่นได้ใกล้จุดที่เป็นไปได้ (feasible point) ในการประเมินประสิทธิภาพของวิธีผสมนี้ ได้นำ FTGA ไปใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดกับ 4 โจทย์ปัญหาที่มีฟังก์ชันซับซ้อน และ มีข้อจำกัดไม่เชิงเส้นทั้งอสมการ และ /หรือสมการข้อจำกัด โดยได้ทำการเปรียบเทียบผลลัพธ์นี้กับ AGA เดี่ยวๆ ได้ผลการทดลองเชิงตัวเลขซึ่งแสดงว่า FTGA .สามารถให้ประสิทธิภาพและ ความแม่นยำของคำตอบที่เป็นค่าเหมาะสมที่สุดครอบคลุมของปัญหาเชิงซ้อน ไม่เชิงเส้น มีมิติตัวแปรตัดสินใจสูง และ มีคำตอบได้หลายค่า ภายใต้ข้อจำกัดไม่เชิงเส้น อีกทั้ง FTGA ยังได้นำไปแก้ปัญหาการหาค่าเหมาะสมที่สุดเกี่ยวกับการอออกแบบของกลไกลการ crank-toggle ซึ่งเป็นข้อพิสูจน์ว่า FTGA สามารถใช้แก้ปัญหาบนโลกจริงได้ด้วย

2.2.3 งานวิจัยของ Zahara และ Kao [34]

ได้นำเสนอวิธีผสมของจัดการข้อจำกัดระหว่างวิธีซ่อมแซมเชิงเกรเดียนต์กับการจัดอันดับคุณภาพของค่าความเหมาะสมของข้อจำกัด ซึ่งตัวจัดการร่วมนี้ทำหน้าที่เป็นตัวดำเนินการพิเศษในการปรับค่าข้อจำกัดให้ดีขึ้น โดยนำไปใช้กับขั้นตอนวิธีการแก้ปัญหาการหาค่าเหมาะสมที่สุดด้วยกลุ่มอนุภาค และ วิธีการค้นทางเดียวแบบ Nelder-Mead (NM-PSO) ด้วยการนำไปแก้ปัญหากับชุดฟังก์ชันทดสอบ และ ปัญหาการออกแบบทางวิศวกรรมอย่างละ 3 ชุดปัญหา และ นำมาเปรียบเทียบผลลัพธ์ของวิธีแก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงวิวัฒน์อื่นๆ (GA1, GA2, EP, CPSO, HPSO) ได้ผลการทดลองว่า NM-PSO ที่ทำงานรวมกับตัวดำเนินการพิเศษ มีความแม่นยำความน่าเชื่อถือ และ ประสิทธิภาพในการไปยังคำตอบที่เหมาะสมที่สุดครอบคลุมมากกว่าวิธีอื่นที่นำมาเปรียบเทียบ

2.2.4 งานวิจัยของ Kheawhom [19]

ได้เสนอรูปแบบจัดการข้อจำกัดที่ใช้งานร่วมกับขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (DRDE) มาใช้แก้ปัญหาการหาค่าเหมาะสมที่สุดแบบมีข้อจำกัด ซึ่งวิธีจัดการข้อจำกัดนั้นได้ประยุกต์เอาขั้นตอนวิธีซ่อมแซมมาใช้งานร่วมกัน ขั้นตอนวิธีซ่อมแซม (repair algorithm) นั้นมีพื้นมาจากการทำข้อมูลเชิงเกรเดียนต์บนสมการเงื่อนไข มีผลให้สามารถปรับผลลัพธ์ที่ไม่อนุโลมใช้ให้เป็นผลลัพธ์ที่

อนุโลมใช้ได้ และ จัดการข้อจำกัดอีกตัวที่นำมาใช้ร่วมกัน คือ รูปแบบการเลือกตัวเด่น (dominance-based selection scheme) ซึ่งในการเปรียบเทียบประสิทธิภาพ ได้นำวิธีการทำฟังก์ชันลงโทษซึ่งเป็นวิธีมาตรฐาน โดยนำไปทดลองกับชุดปัญหาทดสอบ และ ปัญหาการหาค่าเหมาะสมที่สุดเชิงวิศวกรรมที่มีลักษณะไม่เชิงเส้น ผลลัพธ์ที่ได้แสดงถึงคุณภาพของคำตอบ และจำนวนฟังก์ชันในการประเมิน (NFE) ซึ่ง DRDE ให้ประสิทธิภาพที่ดีต่อการนำไปแก้ปัญหาดังกล่าว

2.2.5 งานวิจัยของ Liu, Cai และ Wang [35]

ว่าด้วยการแก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงตัวเลข และ วิศวกรรมด้วยวิธีที่นำเสนอโดยการผสมรวมกันของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง (DE) กับขั้นตอนวิธีการหาค่าเหมาะสมที่สุดด้วยกลุ่มอนุภาค (PSO) เนื่องจาก PSO โดยปกติแล้วจะติดภาวะหยุดชะงักง่ายหากกลุ่มอนุภาคไปพบตำแหน่งที่ดีกว่าตำแหน่งที่ดีที่สุดในหลายๆ รอบก่อนหน้า จึงได้ใช้คุณสมบัติของ DE เข้ามาประกอบในการอัพเดทตำแหน่งที่ดีที่สุดของรอบก่อนหน้าเพื่อให้เกิดการกระโดดข้ามภาวะหยุดชะงักนี้ได้ จากนั้นจึงนำไปแก้ปัญหาใน 11 ชุดปัญหาฟังก์ชันทดสอบที่รู้จักกันมากอีกทั้งยังทดสอบกับ 5 ปัญหาการหาค่าเหมาะสมที่สุดเชิงวิศวกรรมโดยนำมาเปรียบเทียบกับ 7 วิธีการที่เป็นที่ยอมรับกันอยู่แล้ว ซึ่งผลลัพธ์ที่ได้ของขั้นตอนวิธีที่นำเสนอนี้มีเพิ่มประสิทธิภาพ และ ความเร็วในการลู่เข้าของคำตอบเพิ่มขึ้น

2.2.6 งานวิจัยของ Zhang, Rangaiah และ Adrian

การประมาณค่าพารามิเตอร์ในโมเดลทางเทอร์โมไดนามิกส์เป็นสิ่งที่น่าสนใจในสายวิศวกรรมเคมีเพราะว่ามีความซับซ้อนอยู่ในตัวปัญหาเอง ไม่ว่าจะเป็นด้านความไม่เป็นเชิงเส้นของมัน บริเวณรอบจุดที่เหมาะสมที่สุดนั้นมีค่าของฟังก์ชันวัตถุประสงค์ที่ราบเรียบปกคลุมอยู่ และ ส่วนที่ไม่สามารถหาอนุพันธ์บนสมการได้ งานวิจัยนี้ได้นำขั้นตอนวิธีแบบสโทแคสติกสำหรับแก้ปัญหาการหาค่าเหมาะสมที่สุดซึ่งเรียกว่า วิธีเชิงวิวัฒน์แบบส่วนต่างแบบผสาน (IDE) โดยที่นำ IDE มาแก้ปัญหาการประมาณค่าพารามิเตอร์ของโมเดลข้อมูลของสมดุลก๊าซ และ ของเหลว (VLE) ซึ่งการทำงานภายใน IDE จะมีระเบียบแผนในการเลือกรูปแบบการกลายพันธุ์ และ พารามิเตอร์ที่ปรับได้ตามการเรียนรู้จากผลลัพธ์ในรอบก่อนหน้า และ มีการใช้งานการเก็บรายการแบบทาบูร่วมด้วยเพื่อเลี่ยงการคำนวณซ้ำในบริเวณที่เคยคำนวณไปแล้วที่ทำให้เกิดการประเมินฟังก์ชันที่ไม่จำเป็น และ มี

เกณฑ์ใหม่สำหรับหยุดรอบการคำนวณที่ขึ้นกับจำนวนเวกเตอร์ทดลองที่ถูกคัดออกในรอบนั้นๆ ซึ่งถูกเทียบกับเกณฑ์อื่นๆ นอกจากนี้ IDE ยังใช้ตัวปรับความเหมาะสมเฉพาะที่ (local optimizer) เพื่อความแม่นยำ และ ประสิทธิภาพสูงสุด ประสิทธิภาพของ IDE ถูกทดลองด้วยชุดฟังก์ชันทดสอบ และ โมเดล VLE และ นำผลมาเปรียบเทียบขั้นตอนวิธีแบบสโทแคสติกอื่นๆ เช่น DE, DE กับรายการแบบทาบู การหาค่าเหมาะสมที่สุดด้วยกลุ่มอนุภาค (PSO) วิธีการอบอ่อนจำลอง (simulated annealing) และ ขั้นตอนวิธีเชิงกำหนด “Branch และ Reduce Optimization Navigator (BARON)” ซึ่งผลลัพธ์แสดงว่า IDE ดีกว่าขั้นตอนวิธีเปรียบเทียบดังกล่าวสำหรับปัญหาการประมาณค่าพารามิเตอร์ในโมเดลข้อมูล VLE.

2.2.7 งานวิจัยของ Zhang และ Rangaiah [20]

ได้นำเสนอวิธีจัดการสำหรับปัญหาการค่าเหมาะสมที่สุดที่แก้ข้อจำกัดได้ทั้งสมการ และ อสมการ ด้วยการทำงานร่วมกันของข้อจำกัดที่ปรับความผ่อนปรนได้ (adaptive relaxation of constraints) กับวิธีการเลือกช่วงที่สามารถเป็นไปได้ ซึ่งวิธีจัดการข้อจำกัดนี้ได้ถูกรวมเข้ากับขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างแบบผสาน (IDE) แล้วจึงนำไปทดลองแก้ปัญหากับชุดทดสอบ อีกทั้งยังนำไปประยุกต์ในปัญหาเชิงวิศวกรรมเคมีที่มีสมการ และ อสมการข้อจำกัด ผลลัพธ์ที่ได้แสดงว่าวิธีจัดการข้อจำกัดที่นำเสนอไปเมื่อใช่ร่วมกับ IDE นั้น (เรียกรวมกันว่า C-IDE) มีทั้งความน่าเชื่อถือ และ มีประสิทธิภาพในการแก้ปัญหาการหาค่าเหมาะสมที่สุด และ รวมถึงสมการข้อจำกัดด้วย

2.2.8 สรุป และ วิเคราะห์แนวคิดในงานวิจัยต่างๆ

สรุป และ วิเคราะห์แนวคิดในงานวิจัยต่างๆ พร้อมทั้งแนวคิดที่จะนำเสนอ ปัจจัยสำคัญในแนวคิดสามส่วน ส่วนแรก คือ การออกแบบขั้นตอนวิธีโดยรวมว่างานวิจัยได้นำเอาขั้นตอนวิธีเชิงวิวัฒน์มาใช้งานร่วมกับจัดการข้อจำกัดในลักษณะใด ส่วนที่สอง คือ ขั้นตอนวิธีจากงานวิจัยนำไปเปรียบเทียบกับใครบ้าง และ ส่วนสุดท้าย คือ ส่วนผลลัพธ์ของการเปรียบเทียบในแต่ละงานวิจัยนำมาพิจารณา สรุปได้ดังตารางที่ 2.1

ตารางที่ 2.1 ปัจจัยสำคัญของแนวคิดในงานวิจัยต่างๆ

งานวิจัย	แนวคิดงานวิจัย	งานวิจัยที่นำมาเปรียบเทียบ	วิเคราะห์
2.2.1 Srinivas และ Rangaiah [33]	เสนอวิธี DETL ประกอบด้วย - DE มีจุดเด่นในการก้าวข้ามค่าเหมาะสมที่สุดเฉพาะที่ - Tabu list เลี่ยงการค้นซ้ำบริเวณ	DETL vs DE vs MDE	DETL ให้ค่า NFE ที่ดีกว่าทั้งสองวิธี
2.2.2 Shang, Zhao และ Shen [18]	เสนอวิธี FTGA ประกอบด้วย - AGA ค้นหาคำตอบที่ดีเบื้องต้นแล้วส่งให้ FTM ค้นแบบละเอียด - Nelder-Mead เป็นตัวหลักในการค้นหาคำตอบที่ดีใช้ใน FTM - FTM เพิ่มความเข้มข้นในการค้นคำตอบตัวที่ AGA เลือกว่าดีที่สุด	FTGA vs AGA	FTGA ให้ค่า NFE และคุณภาพคำตอบดีกว่า
2.2.3 Zahara และ Kao [34]	เสนอวิธี NM-PSO ประกอบด้วย - Nelder Mead ใช้ในการหาเส้นทางที่ดีให้แก่อนุภาค - PSO ใช้ตัวดำเนินการให้อนุภาคจำเส้นทางที่ไม่ดีไว้จำนวนหนึ่ง - Gradient ช่วยดึงตัวแปรเข้าพื้นที่ที่เป็นไปได้	ไม่ได้เปรียบเทียบกับใคร	Nelder-Mead เหมาะสำหรับแก้ปัญหาฟังก์ชันที่ไม่มีข้อจำกัด

ตารางที่ 2.1 ปัจจัยสำคัญของแนวคิดในงานวิจัยต่างๆ (ต่อ)

งานวิจัย	แนวคิดงานวิจัย	งานวิจัยที่นำมาเปรียบเทียบ	วิเคราะห์
2.2.4 Kheawhom [19]	เสนอวิธี DRDE ประกอบด้วย - DE ก้าวข้ามจุดที่หาอนุพันธ์ไม่ได้ และ ก้าวข้ามค่าเหมาะสมที่สุดเฉพาะที่ - Newton's Method ดึงตัวแปรให้เข้าใกล้เส้นสมการ ปรับลดค่าข้อจำกัดที่ไม่สอดคล้องกัน	DRDE vs DE+Penalty	DRDE ให้ค่า NFE ที่ดีกว่า
2.2.5 Liu, Cai และ Wang [35]	เสนอวิธี PSO-DE ประกอบด้วย - DE ก้าวข้ามไปค้นตำแหน่งใหม่ที่ดีในระยะไกล - PSO เพิ่มความเข้มงวดการค้นตำแหน่งที่ดีระยะใกล้	PSO-DE vs six state-of-art (CRGA, SAPF, CDE, CULDE, CPSO-GD, SMES)	PSO-DE ได้ผลลัพธ์ที่เหนือกว่าด้านความเร็วของการลู่เข้าคำตอบ
2.2.7 Zhang และ Rangaiah [20]	เสนอวิธี C-IDE ประกอบด้วย - IDE เป็น DE ที่ปรับค่าตัวดำเนินการได้ระหว่างรอบ - Tabu list เลี่ยงการค้นซ้ำทาง - Constraint Relaxation เป็นตัวสร้างพื้นที่ที่เป็นไปได้แล้วปรับให้เล็กลงเรื่อยๆ ในแต่ละรอบ	C-IDE vs ANT-β vs PSO-DE vs EA C-IDE vs DETL C-IDE vs DRDE	C-IDE ให้ค่า NFE ที่ดีกว่า C-IDE ให้ค่า NFE ที่ดีกว่า C-IDE ให้ค่า NFE ที่ดีกว่า แต่ด้วยจำนวนประชากรที่ใช้น้อยกว่า

ตารางที่ 2.1 ปัจจัยสำคัญของแนวคิดในงานวิจัยต่างๆ (ต่อ)

งานวิจัย	แนวคิดงานวิจัย	งานวิจัยที่นำมาเปรียบเทียบ	วิเคราะห์
งานวิจัยที่จะนำเสนอ	เสนอวิธี T-DRDE ประกอบด้วย - DE ก้าวข้ามจุดที่หาอนุพันธ์ไม่ได้ และ ก้าวข้ามค่าเหมาะสมที่สุดเฉพาะที่ - Newton's method ดึงตัวแปรให้เข้าใกล้เส้นสมการ ปรับลดค่าข้อจำกัดที่ไม่สอดคล้องกัน - Tolerance Relaxation ลดความเข้มงวดและ เพิ่มโอกาสการกระโดดของตัวแปรให้เข้าใกล้เส้นสมการ	T-DRDE vs DRDE T-DRDE vs C-IDE	ทดสอบโจทย์เดียวกับที่ C-IDE ใช้เทียบกับ DRDE ที่จำนวนประชากรเท่ากัน

บทที่ 3

วิธีดำเนินการวิจัย

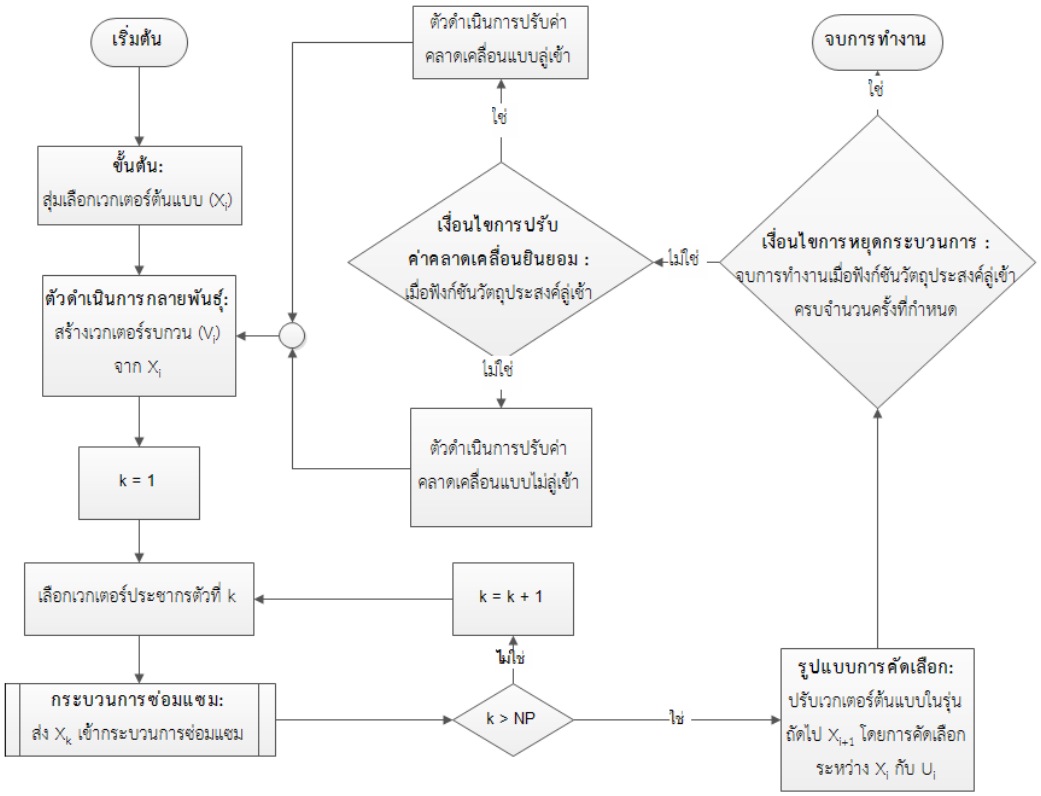
ในบทนี้จะพูดถึงที่มาสาเหตุของงานวิจัย อธิบายถึงขั้นตอนการทำงานของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง และ อธิบายการทำงานร่วมกันกับขั้นตอนวิธีซ่อมแซม แสดงแนวคิดในการเลือกใช้กฎของค่าคลาดเคลื่อนยินยอม และ ความสำคัญของการตรวจสอบค่าการลู่เข้ารวมถึงการนำไปใช้ประโยชน์

จากการศึกษาด้านการหาค่าเหมาะสมที่สุดเพื่อใช้แก้ปัญหาทางวิศวกรรม มุ่งเน้นเพื่อปรับปรุงคุณภาพของขั้นตอนวิธี DRDE เพื่อลดจำนวนการประเมินฟังก์ชันให้น้อยลง โดยที่ประสิทธิภาพ และ คุณภาพของคำตอบใกล้เคียงเดิม เมื่อพิจารณาจากงานวิจัยทั้งขั้นตอนวิธี C-IDE และ DRDE พบว่าขั้นตอนวิธีที่คาดว่าสามารถปรับปรุงประสิทธิภาพได้ คือ DRDE เนื่องจากขั้นตอนวิธี DRDE ยังมีจุดบกพร่องบางประการตรงที่ขั้นตอนวิธี DRDE ไม่เหมาะกับปัญหาการหาค่าเหมาะสมที่สุดที่มีอสมการข้อจำกัดเป็นจำนวนมาก นั่นเป็นสาเหตุที่จำนวนการประเมินฟังก์ชันของขั้นตอนวิธี DRDE มีปริมาณมากเกินไป

ดังนั้นงานวิจัยนี้จึงมุ่งเน้นไปที่การปรับปรุงขั้นตอนวิธี DRDE ให้มีประสิทธิภาพในอสมการข้อจำกัดให้มากขึ้นเพื่อลดปริมาณการประเมินฟังก์ชันใน จึงได้เสนอวิธีการปรับปรุงขั้นตอนวิธีซ่อมแซมด้วยความคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) ซึ่งวิธีดังกล่าวเป็นการพัฒนาต่อยอดมาจากขั้นตอนวิธี DRDE ประยุกต์กับค่าความยืดหยุ่นยินยอมภายในขั้นตอนวิธีซ่อมแซมให้มีความยืดหยุ่นในแต่ละรอบของการคำนวณ ซึ่งวิธีที่เสนอนี้จะทำให้ลดปริมาณ NFE โดยที่ความแม่นยำของคำตอบใกล้เคียงเดิม

3.1 ขั้นตอนวิธีการซ่อมแซมเชิงเกรเดียนต์ที่ใช้ระเบียบวิธีของนิวตันมาจัดการสมการข้อจำกัด

ในส่วนนี้จะเป็นการอธิบายถึงขั้นตอนวิธีซ่อมแซมที่นำมาประยุกต์ใช้ร่วมกับขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างตลอดกระบวนการทั้งหมดนั้นทำงานประสานกันอย่างไร โดยเริ่มจากภาพรวมของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างที่ทำงานร่วมกับขั้นตอนซ่อมแซมแสดงดังภาพที่ 3.1



ภาพที่ 3.1 ภาพรวมการทำงานประสานขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างกับขั้นตอนวิธีซ่อมแซมร่วมกับความคลาดเคลื่อนยินยอมแบบยืดหยุ่น

3.1.1 การสร้างประชากรตั้งต้น (Initial)

ประชากรทั้งหมดมีลักษณะเป็นเวกเตอร์สำหรับเก็บค่าของตัวแปรตัดสินใจ (Decision Variables) ที่เกี่ยวข้องกับปัญหา ซึ่งแต่ละปัญหาก็จะมีจำนวนตัวแปรตัดสินใจไม่เท่ากัน ซึ่งประชากรทั้งหมดที่สุ่มได้ในขั้นตอนแรกเรียกว่า เวกเตอร์ต้นแบบ (Parent Vector :X) ซึ่งเวกเตอร์ต้นแบบทุกตัวจะถูกนำมาคำนวณหาผลรวมของค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกัน (Total Absolute Violation) ดังที่อธิบายไว้ในบทที่ 2 สมการที่ 16 เมื่อได้ค่าข้อจำกัดที่ไม่สอดคล้องกันครบทุกประชากรแล้ว จากนั้นจึงเลือกค่ามัธยฐานของข้อจำกัดที่ไม่สอดคล้องกันเหล่านี้ มากำหนดให้เป็นค่าความคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) ในการวนซ้ำรอบแรกใช้สัญลักษณ์ คือ $\boldsymbol{\varepsilon}_1$

หลังจากนั้นจะเข้าสู่กระบวนการวนซ้ำ ซึ่งภายในกระบวนการจะประกอบไปด้วย ตัวดำเนินการกลายพันธุ์ (Mutation Operator) ตัวดำเนินการสลับสายพันธุกรรม (Crossover

Operator) รูปแบบการคัดเลือก (Selection Scheme) และ เงื่อนไขการหยุดวนซ้ำ (Stop Criteria) ซึ่งเมื่อเข้าสู่ตัวดำเนินการกลายพันธุ์เวกเตอร์ต้นแบบจากรอบแรกจะถูกคัดลอกเพิ่มมาอีกชุดจากนั้นจึงนำเวกเตอร์ที่คัดลอกนี้ทำการกลายพันธุ์เพื่อสร้างความเรียกว่าเวกเตอร์รบกวน (Perturb Vector :V) ต่อไป เวกเตอร์รบกวนที่ได้นี้จะมีค่าของตัวแปรตัดสินใจที่ต่างออกไปกับเวกเตอร์ต้นแบบ

3.1.2 ตัวดำเนินการกลายพันธุ์ (Mutation Operator)

จะไม่ใช้วิธีสุ่มเหมือนเวกเตอร์ต้นแบบ แต่จะใช้รูปแบบการดำเนินการกลายพันธุ์ตามที่ระบุไว้ที่บทที่ 2 สมการที่ 3 และ 4 จะเห็นว่ารูปแบบการดำเนินการกลายพันธุ์ที่นำมาใช้นั้นมี 2 ประเภท ซึ่งได้กำหนดอัตราส่วนการเลือกใช้รูปแบบที่สองไว้ที่ 5% เมื่อได้ค่าจากการกลายพันธุ์แล้วจะต้องทำการตรวจสอบขอบเขตบน และ ล่างของตัวแปรนั้นก่อน ซึ่งหากค่าที่ได้ต่ำหรือสูงกว่าขอบเขตที่กำหนดไว้ก็จะต้องปัดให้ไม่หลุดขอบเขตดังกล่าวเสมอ จากนั้นจึงนำเวกเตอร์ทั้งสองชนิดนี้ไปเข้าสู่ตัวดำเนินการสลับสายพันธุกรรม

กรณีปัญหา MILP หรือ MINLP ก็ให้ทำการปัดเศษเช่นเดียวกันการทำเวกเตอร์ต้นแบบด้วย

3.1.3 ตัวดำเนินการสลับสายพันธุกรรม (Crossover Operator)

จากโครงสร้างภายในเวกเตอร์ทั้งสอง ภายในเวกเตอร์จะประกอบด้วยเซตของตัวแปรตัดสินใจทั้งหมดที่เกี่ยวข้องกับโจทย์เป็นจำนวน n ตัวแปร ซึ่งตำแหน่งที่ 1 ถึง n ของเวกเตอร์ต้นแบบ และ เวกเตอร์รบกวนนั้นจะต้องอ้างถึงตัวแปรตัวเดียวกัน ฉะนั้นในการสลับสายพันธุกรรมที่กล่าวถึงจะเป็นการเลือกตัวแปรโดยจะเลือกมาเพียงค่าเดียวจากค่าตัวแปรในทั้งสองเวกเตอร์ซึ่งจะพิจารณาทีละตำแหน่งตั้งแต่ 1 ถึง n (ไม่มีการสลับสายพันธุกรรมข้ามตำแหน่งกัน เพื่อรักษาคุณสมบัติขอบเขตของตัวแปรเอาไว้) เมื่อคัดเลือกค่าเสร็จแล้วจากนั้นจะนำมาเก็บไว้ในเวกเตอร์ทดลอง (Trial Vector :U) ให้ตรงตามตำแหน่งของการคัดเลือกทุกประการ ซึ่งผลลัพธ์ที่ได้จะเป็นเวกเตอร์ทดลองที่มีจำนวนเท่ากับเวกเตอร์ต้นแบบ และ เวกเตอร์รบกวน ซึ่งภายในเวกเตอร์ทดลองแต่ละตัวที่ตำแหน่งต่างๆ จะเป็นตัวแปรตัดสินใจตัวที่ 1 ถึง n ซึ่งที่ค่าที่ตำแหน่งต่างๆ มาจากเลือกสลับสายพันธุกรรมของเวกเตอร์ต้นแบบกับเวกเตอร์รบกวน เมื่อสร้างเวกเตอร์ทดลองได้ครบแล้วเวกเตอร์รบกวนก็จะถูกทำให้ว่างเพื่อรองรับการทำงานในรอบถัดไป

3.1.4 ขั้นตอนวิธีซ่อมแซม (Repair Algorithm)

ก่อนจะถึงขั้นตอนรูปแบบการคัดเลือกเวกเตอร์ที่ดีเพื่อกำหนดเป็นเวกเตอร์ต้นแบบในรอบถัดไป ได้แทรกขั้นตอนวิธีซ่อมแซมเข้าไปเพื่อจัดการเวกเตอร์ทดลองที่ได้จากกระบวนการสลับสายพันธุกรรมให้มีประสิทธิภาพยิ่งขึ้น โดยใช้ระเบียบวิธีของนิวตันซึ่งนำมาใช้หาค่าข้อจำกัดที่ไม่สอดคล้องกันให้มีค่าต่ำสุด โดยที่ระเบียบวิธีของนิวตัน และ ขั้นตอนวิธีซ่อมแซมเป็นดังนี้

อ้างอิงจากสมการที่ 18 ภาพที่ 2.12 และ 2.13 โดยเริ่มจากนำเวกเตอร์ทดลองเข้าสู่กระบวนการ แล้วจึงทำการปรับค่าตัวแปรภายในเวกเตอร์ทดลอง โดยจะทำเวกเตอร์ทีละตัวจนครบทั้งหมด เมื่อนำเวกเตอร์ทดลองตัวแรกเข้าสู่กระบวนการ ซึ่งในแต่ละรอบของการปรับนี้ต้องทำการสุ่มเลือกตำแหน่งของตัวแปรตัดสินใจภายในเวกเตอร์ทดลองให้มีจำนวนเท่ากับจำนวนสมการข้อจำกัด เพื่อทำให้เป็นจาโคเบียนเมตริกซ์จัตุรัสซึ่งสามารถนำไปหาจาโคเบียนผกผันต่อไปได้ ส่วนตำแหน่งอื่นที่ยังไม่ถูกเลือกก็จะยังไม่ถูกปรับในรอบนี้ และ จะเริ่มสุ่มเลือกตำแหน่งใหม่ในรอบถัดไป การวนซ้ำเพื่อปรับเวกเตอร์ทดลองตัวแรกนี้จะทำจนกว่าจะเจอเงื่อนไขที่สั่งหยุดการปรับนี้ ซึ่งเงื่อนไขสั่งหยุดนั้น คือ ผลรวมของค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกัน (Total Absolute Violation, TAV) ของเวกเตอร์ทดลองจะต้องมีค่าไม่เกินค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) (ซึ่งค่า TAV รอบแรก คือ $\boldsymbol{\varepsilon}_1$ ซึ่งได้กล่าวไว้แล้วที่ต้นกระบวนการ) ถ้าหาก TAV ของเวกเตอร์ทดลองยังมีค่ามากกว่า $\boldsymbol{\varepsilon}_1$ เวกเตอร์ทดลองก็จะถูกนำไปปรับวนซ้ำด้วยระเบียบวิธีของนิวตัน ซึ่งข้อจำกัดของระเบียบวิธีนิวตันดังที่กล่าวมา คือ สามารถปรับลดค่า TAV ได้เฉพาะส่วนที่เป็นสมการข้อจำกัด ซึ่งอันที่จริงแล้ว TAV นั้นประกอบไปด้วย สมการ และ อสมการข้อจำกัด นั่น คือ จะยังมีส่วนที่ไม่สามารถปรับลดค่าได้โดยตรง และ ส่วนนั้นอาจทำให้ไม่สามารถผ่านเงื่อนไขที่สั่งหยุดการซ่อมแซมได้ทั้งๆ ที่ระเบียบวิธีของนิวตันก็ได้ทำการปรับค่าครอบคลุมของสมการข้อจำกัดที่ไม่สอดคล้องกันได้หมดแล้ว นั่นเป็นสาเหตุหลักที่ทำให้เกิดการวนซ้ำจนถึงค่าสูงสุดที่กำหนดไว้ให้วนซ้ำได้ ซึ่งการวนซ้ำนี้เป็นการสิ้นเปลืองจำนวนการประเมินฟังก์ชันที่ไม่จำเป็นอย่างมาก ยิ่งถ้าเป็นขั้นตอนวิธีเดิมของ DRDE ซึ่งค่าคลาดเคลื่อนยินยอม ($\boldsymbol{\varepsilon}$) มีค่าคงที่ ยิ่งเป็นการเข้มงวดกับขั้นตอนวิธีซ่อมแซมมากเกินไป ซึ่งหากค่า $\boldsymbol{\varepsilon}$ กว้างจะช่วยลดจำนวนการประเมินฟังก์ชันได้ แต่ก็จะทำให้คำตอบที่ได้ขาดความแม่นยำไป และ ถ้าค่า $\boldsymbol{\varepsilon}$ แคบก็จะช่วยให้คุณภาพของคำตอบแม่นยำ แต่ก็ส่งผลให้จำนวนการประเมินฟังก์ชันมากเกินไป

ด้วยเหตุนี้ขั้นตอนวิธีเดิมของ DRDE จึงยังมีข้อบกพร่องที่ไม่สามารถใช้การแก้ปัญหาการหาค่าเหมาะสมที่สุดที่มีอสมการข้อจำกัดได้อย่างมีประสิทธิภาพ ดังนั้นงานวิจัยชิ้นนี้จึงได้ปรับปรุงแก้ไขขั้นตอนวิธีซ่อมแซมให้มีความคลาดเคลื่อนยินยอมแบบยืดหยุ่น ซึ่งจะส่งผลให้การทำงานให้รอบต้นๆ ของขั้นตอนวิธีทั้งระบบมีค่า ε กว้างเพื่อลดจำนวนการประเมินฟังก์ชัน จึงยังไม่เน้นคุณภาพของคำตอบเท่าไหร่นัก และ การทำงานในรอบหลังๆ มีค่า ε ที่แคบลงเพื่อเน้นถึงคุณภาพในช่วงท้าย ด้วยแนวคิดนี้ถึงได้มีการออกแบบการทำงานของค่าคลาดเคลื่อนยินยอมที่ผ่อนคลาย (Tolerance Relaxation)นี้ ซึ่งจะได้นำค่า ε_1 ที่คำนวณทิ้งไว้มาใช้งานด้วย โดยจะอธิบายในหัวข้อที่ 2

3.1.5 รูปแบบการคัดเลือก (Selection Scheme)

หลังจากเวกเตอร์ทดลองทุกตัวถูกซ่อมแซมค่าตัวแปรตัดสินใจเสร็จแล้วจากนั้นจะเข้าสู่ขั้นตอนการคัดเลือกเวกเตอร์ที่มีคุณสมบัติในการหาคำตอบของฟังก์ชันวัตถุประสงค์ที่ดีกว่าด้วยวิธีการเปรียบเทียบ ซึ่งเวกเตอร์คู่ที่จะนำมาเปรียบเทียบนั้น คือ เวกเตอร์ต้นแบบ (X) กับเวกเตอร์ทดลอง (V) โดยจะมีรูปแบบการคัดเลือก คือ รูปแบบการคัดเลือกด้วยความเด่น (Dominance-based Selection Scheme) [16] ดังที่ระบุในบทที่ 2 สมการที่ 19 ซึ่งลักษณะการเปรียบเทียบจะทำการเปรียบเทียบกันระหว่างเวกเตอร์ต้นแบบที่ประชากรอันดับเดียวกันกับเวกเตอร์ทดลอง และจะเปรียบเทียบเป็นคู่ๆ ไปจนครบทุกประชากร โดยเวกเตอร์ที่มีคุณสมบัติที่ดีกว่าจะถูกเลือกให้เป็นเวกเตอร์ต้นแบบเพื่อใช้รอบการวนซ้ำถัดไป ส่วนเวกเตอร์ที่มีคุณสมบัติด้อยกว่าก็จะถูกคัดทิ้งไป

3.1.6 เกณฑ์ในการหยุดกระบวนการ (Stop Criteria)

เมื่อสิ้นสุดทุกกระบวนการของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างแล้ว ก่อนที่จะเริ่มการทำงานวนซ้ำในรอบถัดไป เวกเตอร์ต้นแบบจะต้องถูกนำมาหาค่าฟังก์ชันวัตถุประสงค์ของทุกตัวแล้วจึงเลือกตัวที่น้อยที่สุด (ถ้าเป็นการหาฟังก์ชันวัตถุประสงค์ที่มากที่สุด ก็จะเลือกตัวที่มากที่สุด) เพื่อนำมาเปรียบเทียบกับค่าฟังก์ชันวัตถุประสงค์ที่น้อยที่สุดในรอบก่อนหน้า (ซึ่งถ้าเป็นค่าฟังก์ชันวัตถุประสงค์ของรอบแรกจะยังไม่มีตัวเปรียบเทียบให้ข้ามขั้นตอนนี้ไปก่อน) ถ้าหากค่าฟังก์ชันวัตถุประสงค์ของทั้งคู่ต่างกันน้อยกว่าค่าการลู่เข้า (Convergence) ในที่นี้กำหนดมีค่าเท่ากับ 10^{-7} นั่นหมายความว่าค่าฟังก์ชันวัตถุประสงค์นี้อาจเป็นคำตอบที่เป็นค่าเหมาะสมครอบคลุม และ เพื่อ

ความแน่ชัด ถ้าในรอบถัดๆ ไปวัดแล้วผลต่างยังคงมีน้อยกว่าค่าการลู่เข้าอยู่เป็นจำนวนอย่างน้อย 10 ครั้งขึ้นไป ให้ถือว่าได้ค่าฟังก์ชันวัตถุประสงค์ที่เหมาะสมที่สุดแล้ว แล้วจึงสั่งหยุดกระบวนการ

ในทางกลับกันหากผลต่างที่ได้ยังไม่แสดงว่าค่ามีการลู่เข้าจริงก็จะให้วนเข้ากระบวนการวนซ้ำใหม่ และ เริ่มนับผลการลู่เข้าตั้งแต่ 1 จนกว่าผลต่างจะครบตามเกณฑ์ในการหยุดกระบวนการอีกครั้งจึงจะหยุดการทำงาน แต่อย่างไรก็ตามในงานวิจัยนี้จะต้องมีการปรับค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) ทุกครั้งก่อนเริ่มการวนซ้ำในรอบใหม่ ซึ่งจะทำการปรับค่าคลาดเคลื่อนนี้หลังจากผ่านขั้นตอนตรวจสอบเกณฑ์ในการหยุดกระบวนการแล้ว ซึ่งมีรูปแบบ และวิธีการปรับค่าตามที่อธิบายในส่วนที่ 2 ต่อไปนี้

เนื่องจากการวนซ้ำตั้งแต่รอบที่สองเป็นต้นไป จะมีเวกเตอร์ต้นแบบที่ได้จากขั้นตอนคัดเลือกมาให้อยู่แล้ว ดังนั้นขั้นตอนการสร้างประชากรตั้งต้น (Initial) จะไม่ทำซ้ำอีก (ขั้นตอนนี้ให้ทำเฉพาะรอบแรกเท่านั้น)

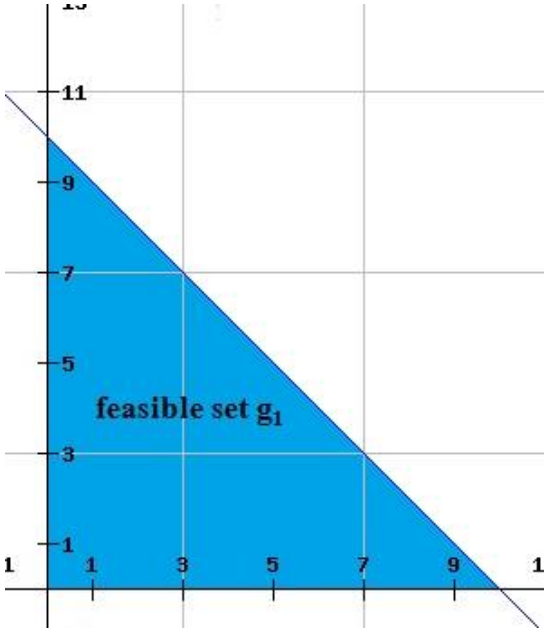
3.2 ค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation)

ปกติการแก้ปัญหาการหาค่าเหมาะสมที่สุดจะต้องทราบบริเวณของการค้นคำตอบทั้งหมดก่อน (Search Space, Feasible Set) เพื่อใช้กำหนดขอบเขตความกว้างในการสุ่มค่าให้กับตัวแปรตัดสินใจ ซึ่งบริเวณของการค้นคำตอบสามารถดูได้จากขอบเขตบน และ ล่างที่ระบุมาภายในปัญหา แต่โดยปกติแล้วปัญหาส่วนใหญ่มักจะกำหนดขอบเขตล่างมาให้เพียงอย่างเดียว ยกตัวอย่างเช่น

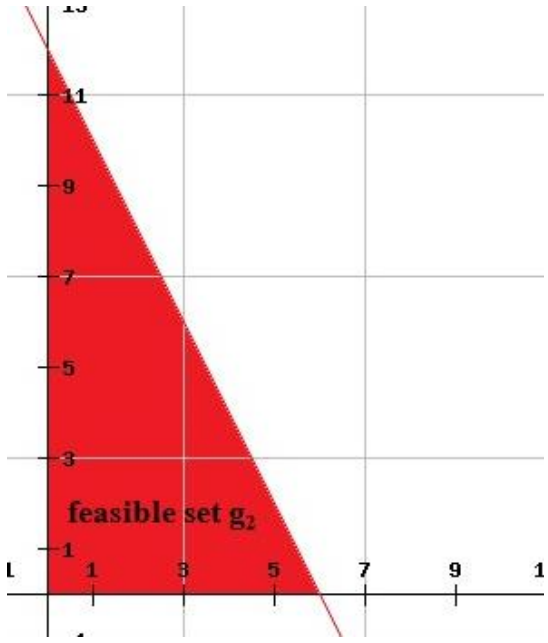
$$
\begin{aligned}
\min \quad & F(X, Y) = YX^{Y} - XY^{X} \\
\text{s.t.} \quad & h(X, Y): \quad (X-2)^{2} - Y = 0 \\
& g_1(X, Y): \quad X + Y - 10 \leq 0 \\
& g_2(X, Y): \quad 2X + Y - 12 \leq 0 \\
& \qquad\quad X \geq 0 \\
& \qquad\quad Y \geq 0
\end{aligned}
\tag{20}
$$

เมื่อพิจารณาที่ขอบเขตล่างของปัญหาจะได้ขอบเขตความกว้างในการสุ่มค่าให้กับตัวแปรตัดสินใจ X และ Y นั้นอยู่ในช่วง $X \in [0, \infty)$ และ $Y \in [0, \infty)$ ซึ่งเป็นช่วงที่กว้างมากเกินไป ดังนั้น

จึงต้องมีการกำหนดขอบเขตบนขึ้นมาเอง โดยดูจากข้อจำกัดที่ 1 ที่ให้มา ข้อจำกัดนี้ คือ X + Y $\leq$ 10 เมื่อพิจารณาค่าสูงสุดที่เป็นไปได้ของทั้ง X และ Y ก็จะสามารถนำมากำหนดขอบเขตบน และล่างได้ดังนี้ X $\in$ [0, 10] และ Y $\in$ [0, 10] จะได้บริเวณของการค้นคำตอบชุดที่ 1 (Feasible Set 1) ดังภาพที่ 3.2



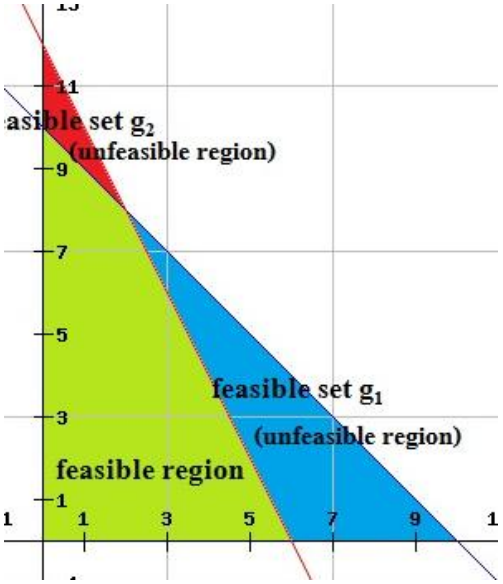
ภาพที่ 3.2 แสดงบริเวณการค้นคำตอบของอสมการข้อจำกัด g1(X, Y) ในตัวอย่าง



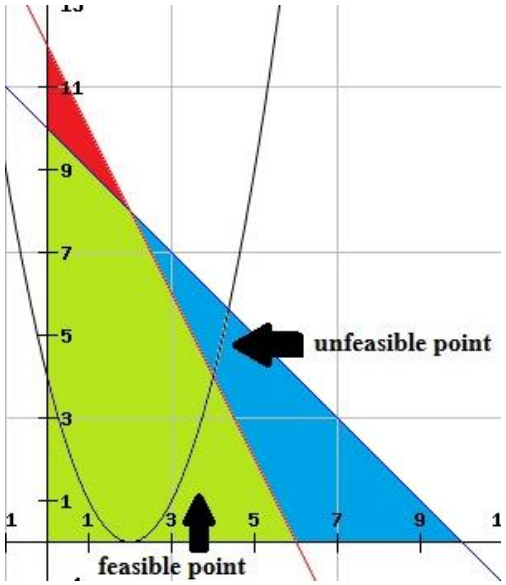
ภาพที่ 3.3 แสดงบริเวณการค้นคำตอบของอสมการข้อจำกัด g2(X, Y) ในตัวอย่าง

ถ้าดูจากข้อจำกัดที่ 2 ที่ให้มา ข้อจำกัดนี้ คือ $2X + Y \leq 12$ เมื่อพิจารณาค่าสูงสุดที่เป็นไปได้ของทั้ง X และ Y ก็จะสามารถนำมากำหนดขอบเขตบน และ ล่างได้ดังนี้ $X \in [0, 6]$ และ $Y \in [0, 12]$ จะได้บริเวณของการค้นคำตอบชุดที่ 2 (Feasible Set 2) แสดงดังภาพที่ 3.3 เมื่อทราบขอบเขตบน และ ล่างของตัวแปรตัดสินใจทุกตัวแล้ว ก็จะสามารถทราบบีบช่วงบริเวณของการค้นคำตอบทั้งหมดให้แคบลงได้

เมื่อได้บริเวณของการค้นคำตอบ (Feasible Set) ของทั้งหมดแล้ว จากนั้นจะเป็นขั้นตอนการกำหนดบริเวณของคำตอบที่เป็นไปได้ (Feasible Region) ซึ่งพิจารณาจากบริเวณของการค้นคำตอบของตัวแปร X และ Y ที่ซ้อนทับกันอยู่ดังแสดงในภาพที่ 3.4



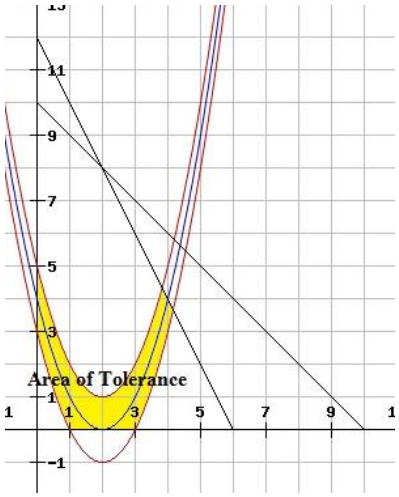
ภาพที่ 3.4 แสดงบริเวณที่เป็นไปได้จากซ้อนทับของบริเวณที่เป็นไปได้ของอสมการ g1 และ g2



ภาพที่ 3.5 แสดงจุดที่เป็นไปได้จากการซ้อนทับของเส้นสมการกับบริเวณที่เป็นไปได้

เมื่อได้บริเวณของคำตอบที่เป็นไปได้แล้ว ต่อไปจะพิจารณาสมการเงื่อนไขที่เหลือ คือ $(X - 2)^2 - Y = 0$ ด้วยความที่เป็นสมการ ทำให้คำตอบของปัญหาต้องอยู่บนจุด ซึ่งต่างกับอสมการที่จะมีคำตอบเป็นช่วงกว้าง เมื่อนำสมการข้อจำกัดมาเขียนร่วมกันกับอสมการข้อจำกัดทั้งหมด จะสามารถกำหนดคำตอบที่อนุโลมให้เล็กลงได้จากส่วนที่ซ้อนทับกันของเส้นสมการข้อจำกัดกับบริเวณของคำตอบที่เป็นไปได้ (feasible region) เรียกว่าจุดของคำตอบที่เป็นไปได้ (feasible point) ดังภาพที่ 3.5

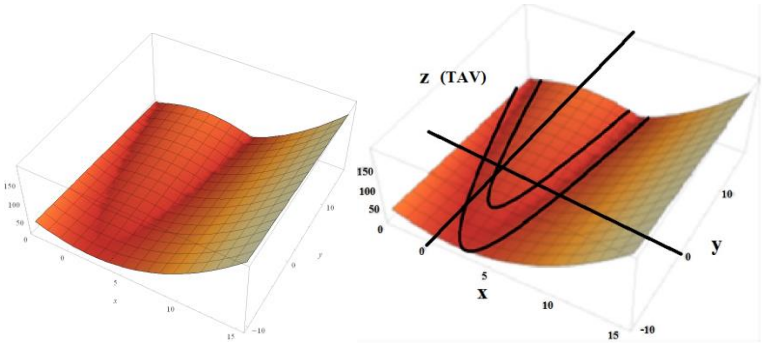
การค้นคำตอบโดยเน้นไปที่จุดของคำตอบที่เป็นไปได้ (Feasible Point) เหล่านี้ เป็นวิธีการแก้ปัญหาการหาค่าเหมาะสมที่สุดวิธีหนึ่งที่มีประสิทธิภาพ และ นิยมใช้ในขั้นตอนวิธีเชิงวิวัฒน์ (Evolutionary Algorithm) แต่อย่างไรก็ตามอัลกอริทึมเชิงวิวัฒน์ไม่มีความสามารถในการรับรู้ถึงจุดของคำตอบที่อนุโลมนี้ได้ จึงจำเป็นต้องมีตัวดำเนินการอื่นมาช่วยเสริมประสิทธิภาพการทำงานนี้ ซึ่งเรียกว่าจัดการข้อจำกัด วิธีการหนึ่งในนั้น คือ การทำค่าคลาดเคลื่อนยินยอม (Tolerance) คลุมจุดของคำตอบที่เป็นไปได้เพื่อสร้างเป็นบริเวณคลาดเคลื่อนยินยอมได้สำหรับการรับรู้ในขั้นตอนวิธีเชิงวิวัฒน์ ซึ่งพิจารณาต่อจากตัวอย่างข้างต้นค่าคลาดเคลื่อนยินยอมจะมีลักษณะดังภาพที่ 3.6



ภาพที่ 3.6 แสดงบริเวณที่เป็นไปได้รอบจุดที่เป็นไปได้ ซึ่งสร้างจากค่าคลาดเคลื่อนยินยอม

3.2.1 กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation)

ค่าคลาดเคลื่อนยินยอมที่มากหรือน้อยจะทำให้บริเวณคลาดเคลื่อนยินยอมใหญ่ขึ้นหรือเล็กลงตามไปด้วย ซึ่งจากขั้นตอนวิธี DRDE นั้นได้ใช้ค่าคลาดเคลื่อนยินยอมเป็นค่าคงที่ซึ่งส่งผลให้การปรับปรุงตัวแปรตัดสินใจภายในกระบวนการซ่อมแซมมีความเข้มงวดเกินไป แต่ถ้ากำหนดให้ค่าคลาดเคลื่อนยินยอมมากเกินไปก็จะมีผลต่อคุณภาพของผลลัพธ์ที่ได้ ดังนั้นค่าคลาดเคลื่อนยินยอมที่แก้ไขในงานวิจัยนี้จึงประยุกต์ให้เริ่มควรมีค่ามากเพื่อลดความเข้มงวดของกระบวนการซ่อมแซม และ ลดลงในแต่ละการวนรอบของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างเพื่อเพิ่มคุณภาพของผลลัพธ์ที่ได้เรียกว่าค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่น (Tolerance Relaxation) โดยค่าคลาดเคลื่อนเริ่มต้นพิจารณาจากค่าข้อจำกัดที่ไม่สอดคล้องกัน (Constraint Violation) ใช้เป็นตัวกำหนดขอบเขต ซึ่งค่าข้อจำกัดที่ไม่สอดคล้องกันสามารถคำนวณได้จากสมการที่ 16



ภาพที่ 3.7 เป็นกราฟของปัญหาที่ยกตัวอย่างมาโดยจะพล็อตกราฟระหว่างแกน Z คือ ค่า TAV ที่ตำแหน่งของตัวแปร X คือ แกน X และ ตัวแปร Y คือ แกน Y

อย่างไรก็ตาม ในการทำงานของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่างนั้นมีเวกเตอร์ประชากรที่ถูกสุ่มออกมาเป็นจำนวนมาก ซึ่งเวกเตอร์แต่ละตัวมีค่า TAV เป็นของตัวเอง ซึ่งไม่อาจทราบได้ว่า TAV ของแต่ละตัวมีค่ามากน้อยเพียงใด แทนที่จะใช้ค่าเฉลี่ยของ TAV จึงกำหนดให้ค่าคลาดเคลื่อนเริ่มต้น ($\boldsymbol{\varepsilon}_1$) ด้วยมัธยฐานของ TAV ทั้งหมดแทน เพื่อแบ่งกลุ่มที่เป็นไปได้กับไม่อนุโลมออกเป็นครึ่งหนึ่งก่อนตามสมการที่ 21

$$\boldsymbol{\varepsilon}_1 = \text{median of all TAV} \tag{21}$$

จากนั้นจึงนำค่าคลาดเคลื่อนยินยอมเริ่มต้นนี้ไปใช้เป็นเกณฑ์กำหนดภายในกระบวนการซ่อมแซมเพื่อปรับค่าของตัวแปรทดลอง (Trial Vector) และ เมื่อตัวแปรทดลองทุกตัวถูกปรับค่าจนครบแล้ว และ ผ่านขั้นตอนคัดเลือก (Selection Scheme) และ ช่วงตรวจสอบเกณฑ์การหยุดกระบวนการ (Stop Criteria) ไปแล้ว ก่อนที่จะทำการวนซ้ำในรอบถัดไป ค่าคลาดเคลื่อนยินยอมนี้ก็จะถูกปรับให้ลดลงสำหรับให้ในรอบถัดไปเช่นกัน โดยกฎการปรับค่าคลาดเคลื่อนนี้ความสัมพันธ์ขึ้นอยู่กับกฎที่ใช้ในการปรับค่าดังชุดสมการที่ 22 23 และ 24

สำหรับกฎที่ 1 : $$\boldsymbol{\varepsilon}_{i+1} = \boldsymbol{\varepsilon}_i\left(1-\frac{F_F}{NP}\right) \tag{22}$$

โดย $\boldsymbol{\varepsilon}_{i+1}$ คือ ค่าคลาดเคลื่อนยินยอมในรอบถัดไป $\boldsymbol{\varepsilon}_i$ คือ ค่าคลาดเคลื่อนยินยอมในรอบปัจจุบัน F_F คือ จำนวนประชากรเวกเตอร์ที่อยู่ในบริเวณของการค้นคำตอบที่เป็นไปได้ และ NP คือ จำนวนประชากรทั้งหมด

สำหรับกฎที่ 2 : $$\boldsymbol{\varepsilon}_{i+1} = \boldsymbol{\varepsilon}_i\left(1-\frac{\log(|F(X_{i+1})-F(X_i)|+1)}{\log(|F_{max}-F_{min}|+1)}\right) \tag{23}$$

โดย $\boldsymbol{\varepsilon}_{i+1}$ คือ ค่าคลาดเคลื่อนยินยอมในรอบถัดไป $\boldsymbol{\varepsilon}_i$ คือ ค่าคลาดเคลื่อนยินยอมในรอบปัจจุบัน $F(X_{i+1})$ คือ ค่าฟังก์ชันวัตถุประสงค์ที่มีค่าน้อยที่สุด(หรือมากที่สุด) จากเวกตอร์ต้นแบบทุกตัวในรอบถัดไป $F(X_i)$ คือ ค่าฟังก์ชันวัตถุประสงค์ที่มีค่าน้อยที่สุด(หรือมากที่สุด) จากเวกตอร์ต้นแบบทุกตัวในรอบปัจจุบัน F_{max} คือ ค่าฟังก์ชันวัตถุประสงค์ที่มีค่ามากที่สุดจากเวกตอร์ต้นแบบทุกตัวในทุกรอบ และ F_{min} คือ ค่าฟังก์ชันวัตถุประสงค์ที่มีค่าน้อยที่สุดจากเวกตอร์ต้นแบบทุกตัวในทุกรอบ

สำหรับกฎที่ 3 : $$\boldsymbol{\varepsilon}_{i+1} = \text{minimumum of all TAV} \tag{24}$$

โดยที่ ε_{i+1} มีค่าเท่ากับผลรวมของข้อจำกัดที่ไม่สอดคล้องกันที่มีค่าน้อยที่สุดเทียบจากเวกเตอร์ประชากรทั้งหมด

แนวคิดของความสัมพันธ์ข้างต้นนี้ อ้างอิงจาก [20] โดยอาศัยแนวคิดที่ว่าค่าคลาดเคลื่อนจะถูกปรับลงด้วยด้วยแฟคเตอร์ซึ่งในที่นี้จะดูจากแนวโน้มของการลู่เข้าของค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด ถ้าค่าฟังก์ชันวัตถุประสงค์มีแนวโน้มว่ายังไม่ลู่เข้านักนั่นหมายความว่าคุณภาพของผลลัพธ์ยังไม่มีประสิทธิภาพเท่าไหร่ ค่าคลาดเคลื่อนยินยอมจึงควรถูกปรับลงมาให้มาก และ ถ้าหากค่าฟังก์ชันวัตถุประสงค์เริ่มมีแนวโน้มที่จะลู่เข้าแล้ว ค่าคลาดเคลื่อนยินยอมก็จะปรับลงมาให้น้อยลง เพื่อไม่ให้เกิดความเข้มงวดต่อกระบวนการซ่อมแซมมากเกินไป จากสมการข้างต้นค่าที่บอกถึงแนวโน้มของการลู่เข้า คือ “ค่าครอบคลุมผลต่างของค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบถัดไปกับรอบปัจจุบัน $|F(X_{i+1})-F(X_i)|$” โดยยิ่งมีค่าไม่ต่างกันมากเท่าไหร่ก็หมายความว่าค่าฟังก์ชันวัตถุประสงค์ยิ่งมีแนวโน้มที่จะลู่เข้ามากเท่านั้น แล้วจึงนำไปพิจารณาเทียบกับตัวส่วน ซึ่งตัวส่วนคำนวณจาก “ค่าครอบคลุมผลต่างของค่าฟังก์ชันวัตถุประสงค์ที่มาก และ น้อยที่สุดจากเวกเตอร์ต้นแบบทุกตัวในทุกรอบ $|F_{max}-F_{min}|$” ซึ่งเมื่อพิจารณาค่าของตัวเศษ และ ส่วนแล้วจะได้ความสัมพันธ์ว่าอัตราส่วน $\left(\frac{|F(X_{i+1})-F(X_i)|}{|F_{max}-F_{min}|}\right)$ มีค่าไม่เกิน 1 เสมอ นั่นหมายความว่าถ้าค่าฟังก์ชันวัตถุประสงค์มีความต่างกันมากจะทำให้อัตราส่วนมีค่าใกล้ 1 มาก และ เมื่อนำ 1 มาลบออกด้วยอัตราส่วนนี้ ก็จะได้แฟคเตอร์ไว้สำหรับปรับลดค่าคลาดเคลื่อนยินยอมในรอบถัดไปซึ่งมีค่าใกล้ 0 แทน และ เมื่อนำแฟคเตอร์นี้ไปคูณกับค่าคลาดเคลื่อนในรอบปัจจุบันก็หมายความว่าค่าคลาดเคลื่อนในรอบถัดไปจะมีค่าน้อยลงไปมาก ในทางกลับกันหากอัตราส่วนมีค่าน้อย (มีการลู่เข้าที่ดี) ก็จะทำให้แฟคเตอร์มีค่าใกล้ 1 ซึ่งคูณค่าคลาดเคลื่อนยินยอมในรอบปัจจุบันแล้วทำให้ค่าคลาดเคลื่อนยินยอมในรอบถัดไปแทบไม่เปลี่ยนแปลงไป แต่ถ้าหากใช้อัตราส่วน $\left(\frac{|F(X_{i+1})-F(X_i)|}{|F_{max}-F_{min}|}\right)$ นี้ในการสร้างแฟคเตอร์ ก็จะพบปัญหาที่ว่าผลต่างของเศษ กับส่วนเริ่มมีค่ามากเกินไป โดยตัวเศษจะเริ่มลู่ $|F(X_{i+1})-F(X_i)|$ เข้าทำให้ผลต่างที่ได้มีค่าแคบลงเรื่อยๆ ในขณะที่ตัวส่วน $|F_{max}-F_{min}|$ ซึ่งจะมีผลต่างที่กว้างขึ้นเรื่อยๆ นั้น ส่งผลทำให้อัตราส่วนมีค่าเข้าใกล้ 0 อยู่ตลอด ซึ่งทำให้ค่าคลาดเคลื่อนยินยอมถูกปรับลงน้อยมาก ดังนั้นเพื่อเป็นการรักษาระดับความต่างกันของค่าเศษ และ ส่วนจึงได้ใช้วิธีทำให้เป็นบรรทัดฐาน (Normalize) ด้วยฟังก์ชันลอการิทึม

ขึ้นดังนี้ $\frac{\log(|F(X_{i+1})-F(X_i)|}{\log(|F_{max}-F_{min}|}$ แต่ด้วยคุณสมบัติของลอการิทึมซึ่งภายในค่า log ห้ามเป็น 0 ดังนั้นจึงได้บวก 1 ตัวเศษเพิ่มเข้าไปไว้สำหรับกรณีที่ผลต่างเป็น 0 ดังนี้ $|F(X_{i+1})-F(X_i)|+1$ และ บวก 1 ที่ตัวส่วนเพิ่มเป็น $|F_{max}-F_{min}|+1$ เพื่อคงคุณสมบัติของอัตราส่วนที่ต้องไม่เกิน 1 ด้วยเหตุผลดังกล่าวทั้งหมดจึงได้อัตราส่วนสุดท้ายเป็นดังนี้ $\frac{\log(|F(X_{i+1})-F(X_i)|+1)}{\log(|F_{max}-F_{min}|+1)}$ ที่กล่าวมาในส่วนนี้เป็นความสัมพันธ์ในการปรับค่าคลาดเคลื่อนยินยอมในแต่ละรอบของขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง แต่ยังมีรูปแบบการปรับค่าคลาดเคลื่อนยินยอมอีกแบบที่มีความสำคัญไม่แพ้กัน ดังนี้

3.2.2 รูปแบบการคัดเลือกความคล้าย (Familiar selection scheme)

เป็นเงื่อนไขในการเลือกวิธีการปรับค่าคลาดเคลื่อนยินยอม (Tolerance, ε) ซึ่งทำงานหลังขั้นตอนเกณฑ์การการหยุดกระบวนการ โดยจะแบ่งการพิจารณาออกเป็น 2 กรณีดังนี้

กรณีที่ 1 เมื่อค่าครอบคลุมผลต่างค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบถัดไปกับรอบปัจจุบัน $|F(X_{i+1})-F(X_i)|$ มีค่าต่างกันเกินค่าการลู่เข้าหรือยังไม่ลู่เข้า (Not Converge) ที่ใช้เป็นพารามิเตอร์เกณฑ์ในการหยุดกระบวนการกำหนดค่าไว้ที่ 10^{-7} หากเป็นไปตามกรณีดังกล่าวค่าคลาดเคลื่อนยินยอมจะถูกปรับตามสมการที่ 22

กรณีที่ 2 เมื่อค่าครอบคลุมผลต่างค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบถัดไปกับรอบปัจจุบัน $|F(X_{i+1})-F(X_i)|$ มีค่าต่างกันไม่เกินค่าการลู่เข้าหรือมีการลู่เข้า (Converge) หรือว่าค่าฟังก์ชันวัตถุประสงค์ของทั้งรอบปัจจุบัน และ รอบถัดไปมาจากตัวแปรต้นแบบตัวเดิม “ค่าคลาดเคลื่อนยินยอมจะปรับเป็นของค่าต่ำสุดของผลรวมค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกัน (minimum TAV) ที่น้อยที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบปัจจุบัน”

เหตุผลที่ต้องมีรูปแบบการปรับค่าคลาดเคลื่อนยินยอมในกรณีที่ 2 นั้น เนื่องมาจากเราไม่สามารถทราบได้ว่าความแตกต่างกันระหว่างค่าฟังก์ชันวัตถุประสงค์ $|F(X_{i+1})-F(X_i)|$ นั้นมาจากเวกเตอร์ต้นแบบทั้งรอบปัจจุบัน และ รอบถัดไปที่เป็นตัวเดียวกันหรือไม่ ซึ่งหากทั้งสองมาจากตัวเดียวกัน นั่นหมายความว่ากระบวนการทั้งหมดของวิธีเชิงวิวัฒน์แบบส่วนต่างในรอบที่ปัจจุบันนั้นไม่สามารถสร้างเวกเตอร์ทดลอง (U_i) ที่มีค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุด $F(U_i)$ ที่ดีกว่า ค่าฟังก์ชัน

วัตถุประสงค์ที่ดีที่สุด $F(X_i)$ ของเวกเตอร์ต้นแบบในรอบเดียวกัน (X_i) ได้ จึงทำให้ $F(X_i)$ กับ $F(X_{i+1})$ มีค่าเหมือนเดิม ซึ่งนั่นก็เป็นสัญญาณเตือนว่าเวกเตอร์ต้นแบบดังกล่าวอาจทำให้ค่าฟังก์ชันวัตถุประสงค์ที่ไม่มีคุณภาพพอ นั่นเป็นเพราะว่าค่าคลาดเคลื่อนยินยอมที่ใช้อาจมีความยืดหยุ่นมากเกินไปจนทำให้เวกเตอร์ต้นแบบดังกล่าวสามารถผ่านเงื่อนไขจากกระบวนการซ่อมแซมไปได้ ซึ่งเมื่อนำไปหาค่าฟังก์ชันวัตถุประสงค์แล้วทำให้ได้ค่าที่น้อยกว่าความเป็นจริง ในขณะที่อาจจะมีเวกเตอร์ต้นแบบตัวอื่นที่อยู่ใกล้กับจุดที่อนุโลมใช้ได้ (Feasible Point) มากกว่าเวกเตอร์ต้นแบบนี้ ซึ่งให้ค่าฟังก์ชันวัตถุประสงค์ที่มีคุณภาพมากกว่าแต่มีค่าของฟังก์ชันไม่เด่นเท่า มีค่าสูงกว่าในกรณีที่ต้องการค่าต่ำสุด ค่าที่สูงกว่าจะเด่นน้อยกว่าค่าที่ต่ำกว่าด้วย จึงทำให้แพ้ในการคัดเลือกความเด่นไป

ดังนั้นจึงพอตั้งข้อสงสัยได้ว่าหากค่าครอบคลุมผลต่างค่าฟังก์ชันวัตถุประสงค์ที่ดีที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบถัดไปกับรอบปัจจุบันมีค่าต่างกันไม่เกินการลู่เข้า อาจเป็นเพราะว่าค่าคลาดเคลื่อนยินยอมที่ใช้ยังไม่เหมาะสมพอ ซึ่งหากเวกเตอร์ต้นแบบดังกล่าวสามารถให้ค่าที่มีคุณภาพจริง เวกเตอร์นั้นก็ควรมีค่าเงื่อนไขที่ไม่สอดคล้องกันใกล้กับจุดที่อนุโลมใช้ได้มากที่สุดเช่นกัน ด้วยเหตุนี้หากเป็นตามกรณีที่สองค่าคลาดเคลื่อนยินยอมจะปรับเป็นของค่าต่ำสุดของผลรวมค่าครอบคลุมของข้อจำกัดที่ไม่สอดคล้องกัน (minimum TAV) ที่น้อยที่สุดจากเวกเตอร์ต้นแบบทุกตัวในรอบปัจจุบัน

อย่างไรก็ตามค่าคลาดเคลื่อนยินยอมจะมีขอบเขตล่างอยู่ที่ 10^{-4} นั่นหมายความว่าถึงแม้ค่าคลาดเคลื่อนยินยอมจะถูกปรับลดลงมาเพียงใด ก็จะลดได้ไม่เกินค่าขอบเขตล่างเสมอ

3.3 องค์ประกอบที่ใช้วัดคุณภาพของงาน

3.3.1 จำนวนการประเมินฟังก์ชัน (Number of Function Evaluations, NFE)

การนับจำนวนการประเมินฟังก์ชันเป็นวิธีที่นิยมใช้วัดประสิทธิภาพของขั้นตอนวิธีกันมาก เนื่องจากมีข้อดีตรงที่ผลการทดลองสามารถใช้เปรียบเทียบได้โดยไม่ต้องจัดการตั้งทรัพยากรของเครื่องทดลองให้เหมือนกับงานวิจัยเปรียบเทียบอื่น อีกทั้งยังเป็นตัวบอกถึงความเสถียรของขั้นตอนวิธีได้ด้วย โดยการวัดจำนวนการประเมินฟังก์ชันที่ดีที่สุด แย่ที่สุด และ ค่าเฉลี่ยของการรันทุกครั้ง โดยค่าจำนวนการประเมินฟังก์ชันยิ่งน้อยยิ่งหมายความขั้นตอนวิธีนั้นมีประสิทธิภาพดี ค่าจำนวนการประเมินฟังก์ชันนั้นจะถูกนับทุกครั้งที่มีการนำเวกเตอร์ของตัวแปรตัดสินใจเข้าไปคำนวณหาใน

ฟังก์ชันใดก็ตาม ไม่ว่าจะเป็นฟังก์ชันวัตถุประสงค์ ข้อจำกัด โดยเวกเตอร์ 1 ตัว เมื่อเรียกคำนวณฟังก์ชัน 1 ครั้ง ก็จะมีจำนวนการประเมินฟังก์ชันเพิ่ม 1 ครั้งเช่นกัน นั่นหมายความว่าหากกำหนดจำนวนประชากรของเวกเตอร์ที่ 1,000 ตัว แล้วในรอบการวน จำเป็นต้องให้เวกเตอร์คำนวณค่าข้อจำกัดที่ไม่สอดคล้องกัน (TAV) ทุกตัว ก็จะเป็นการเพิ่มจำนวนการประเมินฟังก์ชันอีก 1,000 ครั้งเช่นกัน แล้วถ้าต้องวนซ้ำถึง 10 รอบ ทุกรอบก็ต้องคำนวณ (TAV) แบบนี้เช่นกัน ก็สามารถประเมินคร่าวๆ ได้แล้วว่าจำนวนการประเมินฟังก์ชันควรมีค่าอย่างน้อย 10,000 ครั้งเช่นกัน

ดังนั้นปัจจัยหนึ่งที่มีผลต่อจำนวนการประเมินฟังก์ชันเป็นอย่างมาก คือ จำนวนประชากรที่ใช้จะต้องระบุให้ชัดเจนว่าใช้จำนวนประชากรกี่ตัว เพื่อการเปรียบเทียบจะได้กำหนดจำนวนประชากรที่เท่ากัน เว้นแต่ขั้นตอนวิธีบางอย่างอาจจะเหมาะสมกับจำนวนประชากรที่มากกว่า แต่การใช้จำนวนประชากรที่น้อยกว่าถึงแม้ว่าขั้นตอนวิธีสามารถแก้ปัญหาได้ก็ตามแต่ก็ไม่ควรนำมาใช้เปรียบเทียบกับขั้นตอนวิธีที่ใช้จำนวนประชากรมากกว่า

อีกปัจจัยหนึ่งที่มีผลต่อจำนวนการประเมินฟังก์ชัน คือ เลขต้นของการสุ่มค่า (random seed) เนื่องด้วยการสุ่มค่าให้กับตัวแปรตัดสินใจนั้น จะสุ่มจากค่าที่อยู่ภายใต้บริเวณของการค้นคำตอบทั้งหมด ซึ่งบางปัญหาอาจมีบริเวณของการค้นคำตอบที่กว้างมาก ทำให้การสุ่มค่าให้ตัวแปรตัดสินใจบางตัวอาจได้ตำแหน่งที่ไม่ดี เช่น อยู่ในบริเวณที่ไม่อนุโลมให้ใช้ได้ (Unfeasible Region) หรืออยู่ในบริเวณที่ไม่เป็นคอนเวกซ์ (Non-convex) ซึ่งเป็นบริเวณที่ไม่ต่อเนื่องกับบริเวณใด และ มีความเป็นไปได้บ่อยที่จะพบค่าเหมาะสุดเฉพาะที่ (Local Optimum) ซึ่งบริเวณเหล่านี้ล้วนแต่ทำให้จำนวนประเมินฟังก์ชันมีค่าสูงขึ้น เพราะต้องอาศัยแรงของขั้นตอนวิธีเชิงวิวัฒน์ในการขับเคลื่อนเวกเตอร์ที่มีค่าของตัวแปรตัดสินใจที่ติดค่าเหมาะสุดเฉพาะที่นี้ให้หลุดอาจจากบริเวณดังกล่าว ดังนั้นหากเลขต้นของการสุ่มค่านั้นสุ่มค่าให้กับตัวแปรตัดสินใจแล้ว ค่าดังกล่าวตกไปในบริเวณที่ไม่ดีหลายตัว นั่นก็เป็นเหตุหนึ่งที่ทำให้จำนวนการประเมินฟังก์ชันออกมามากกว่าที่ควรจะเป็นก็ได้ เพื่อเป็นการยุติธรรมต่อขั้นตอนวิธีที่จะใช้เปรียบเทียบ จึงควรระบุเลขต้นของการสุ่มค่าไว้ด้วย เพื่อจะได้ใช้ชุดของเวกเตอร์ตัวแปรที่บริเวณเริ่มต้นที่ตำแหน่งเดียวกัน

3.3.2 อัตราความสำเร็จ (Success Rate)

ตัววัดประสิทธิภาพตัวนี้เป็นการบอกถึงความทนทาน (Robustness) ของขั้นตอนวิธี และความเสถียร (Steady) ของคำตอบ ซึ่งใช้ร่วมกับเลขต้นการสุ่มค่า (random seed) การสุ่มค่าให้กับ

ตัวแปรตัดสินใจในบางปัญหานอกจากนะทำให้จำนวนการประเมินฟังก์ชันมากขึ้นแล้ว บางครั้งอาจทำให้คำตอบไม่สามารถใช้ขั้นตอนวิธีเชิงวิวัฒน์ดังให้หลุดออกจากบริเวณนั้นได้ จึงทำให้เกิดภาวะติดอยู่กับค่าเหมาะสมเฉพาะที่ ซึ่งการติดแบบภาวะนี้บ่อยแสดงถึงความไม่เสถียรในการหาคำตอบของขั้นตอนวิธีนั้นๆ ส่วนความทนทานอาจมองในมุมของการนำไปใช้แก้โจทย์ปัญหาได้หลากหลายแบบก็ได้ ซึ่งขั้นตอนวิธีที่มีความทนทานสูงย่อมที่จะใช้หาคำตอบในปัญหาได้หลากหลาย แต่ไม่จำเป็นว่าคำตอบที่ได้มีคุณภาพดีหรือไม่ อัตราความสำเร็จในที่นี้จึงหมายถึงจำนวนครั้งที่สามารถหาค่าเหมาะสมครอบคลุมได้ความสำเร็จจากจำนวนครั้งที่รันทั้งหมด โดยที่ค่าเหมาะสมครอบคลุมนี้อ้างอิงจากค่าที่ดีที่สุดที่หลายงานวิจัยคำนวณออกมาได้ ซึ่งควรจะได้ใกล้เคียงกับเขา ส่วนความละเอียดคำตอบในหลักทศนิยมนั้น งานวิจัยนี้กำหนดความละเอียดอยู่ที่ 10^{-7} หรือ คือ ค่าการลู่เข้าของคำตอบ (Convergence) ที่ใช้เป็นเกณฑ์ในการหยุดกระบวนการนั่นเอง

3.3.3 วิเคราะห์กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอม

ในการวิเคราะห์กฎการเปลี่ยนแปลงนั้นจะดูจากจำนวนการประเมินฟังก์ชันเป็นหลัก นั่นคือ กฎการเปลี่ยนแปลงใดที่ทำใช้จำนวนการประเมินฟังก์ชันน้อยกว่า กฎนั้นย่อมดีกว่า ส่วนอีกจุดหนึ่งที่จะวิเคราะห์ คือ พฤติกรรมของกฎการเปลี่ยนแปลง โดยจะสังเกตจากค่าคลาดเคลื่อนยินยอมที่เปลี่ยนไปในแต่ละรอบนั้นมีผลอย่างไรกับพฤติกรรมของเวกเตอร์

บทที่ 4

การทดลอง และ ผลการทดลอง

ในบทนี้จะกล่าวถึงเครื่องมือที่ใช้ในการทดลองทั้งซอฟต์แวร์ ฮาร์ดแวร์ โจทย์ใช้วัดเปรียบเทียบสถานะ และ กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอม ซึ่งมีขั้นตอนวิธีที่ใช้เปรียบเทียบ คือ ขั้นตอนวิธีที่เสนอ DRDE และ C-IDE โดยทดสอบประสิทธิภาพจากการคำนวณค่า NFE และ SR ใน random seed ชุดเดียวกัน

4.1 เครื่องมือที่ใช้ในการทดลอง

1) ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรมคอมพิวเตอร์ คือ MATLAB เวอร์ชัน 7.6.0.324 (R2008a)

2) ฮาร์ดแวร์ที่ใช้ในการทดสอบ คือ ซีพียู Intel® Core™ i5-450M Processor (3MB L3 Cache, 2.40 GHz) หน่วยความจำหลัก 2048 MB ฮาร์ดดิสก์ 300 GB บนระบบปฏิบัติการ Microsoft Window 8 64-bit, x64 based processor

3) โจทย์การวัดเปรียบเทียบสถานะที่ใช้ในการทดลอง มีดังนี้

ตารางที่ 4.1 แสดงคุณลักษณะทางคณิตศาสตร์ของปัญหา

ปัญหา	ลักษณะของปัญหาการหาค่าเหมาะสมที่สุด	จำนวน					ค่าเหมาะสมที่สุด
		พารามิเตอร์	LI	NI	LE	NE	ครอบคลุม, f*
P01	Nonlinear programming (NLP) [36]	5	0	0	0	3	0.0539498
P02	Nonlinear programming (NLP) [36]	4	2	0	0	3	5126.4981
P03	Mixed integer nonlinear programming (MINLP) [37]	9	4	0	3	2	99.23963
P04	Non-convex MINLP [38]	5	3	0	0	2	7.66718
P05	Heat exchanger network synthesis [38]	12	0	0	11	0	36162.9886
P06	MINLP small planning [39]	11	5	0	3	2	-1.9230986
P07	Reactor network design [40]	6	0	1	0	4	-0.388812
P08	Heat exchanger network design [41]	8	0	3	3	0	7049.25
P09	Separation network synthesis [40]	22	0	0	7	9	1.864

โดยที่ LI NI LE และ NE คือ ข้อจำกัดของอสมการเชิงเส้น อสมการไม่เชิงเส้น สมการเชิงเส้น และสมการไม่เชิงเส้นตามลำดับ

ตารางที่ 4.2 พารามิเตอร์ และ การตั้งค่าสำหรับใช้ในการทดลอง มีดังนี้

พารามิเตอร์	ปริมาณ
จำนวนประชากรที่ใช้, NP	1000
จำนวนรอบสูงสุด	1000
จำนวนรอบการซ่อมแซมสูงสุด	100
ค่าคลาดเคลื่อนต่ำสุด, ε	0.0001
อัตราการกลายพันธุ์, F	0.85
อัตราการสลับสายพันุกรรม, CR	0.8
อัตราการกลายพันธุ์ด้วย Trigonometric	0.05
อัตราการเปลี่ยนแปลงค่าลู่เข้า, SC	1.00%
จำนวนรอบของการลู่เข้า, T	10
*สำหรับปัญหาที่ P08 ใช้จำนวนรอบของการลู่เข้า 40	

4.2 กฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมที่นำมาใช้เปรียบเทียบ

กำหนดให้ $\alpha = (1 - \frac{F_F}{NP})$ และ $\beta = (1 - \frac{\log(|F(X_{i+1}) - F(X_i)| + 1)}{\log(|F_{max} - F_{min}| + 1)})$

โดยนิยามของตัวแปรต่างๆ นั้นเป็นไปตามสมการที่ 22 23 และ 24 ซึ่งจะทำการแบ่งกฎการปรับค่าความคลาดเคลื่อนออกเป็น 4 กฎตามตาราง 4.3 ดังนี้

ตารางที่ 4.3 กฎการปรับค่าคลาดเคลื่อนทั้ง 4 กฎ

กฎการปรับค่าคลาดเคลื่อน	กรณีฟังก์ชันวัตถุประสงค์ยังไม่ลู่เข้า	กรณีฟังก์ชันวัตถุประสงค์ลู่เข้าแล้ว
กฎที่ 1	1 - α	
กฎที่ 2	1 - β	
กฎที่ 3	1 - β	minimum of all TAV
กฎที่ 4	minimum of all TAV	minimum of all TAV

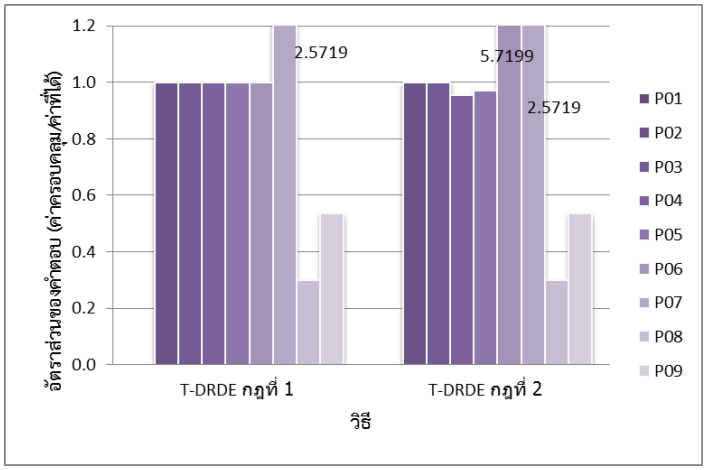
4.3 วิธีการทดลอง แบ่งออกเป็น 2 ขั้นตอน

4.3.1 เปรียบเทียบประสิทธิภาพของกฎการเปลี่ยนแปลงค่าคลาดเคลื่อนยินยอมทั้ง 4 แบบ

เปรียบเทียบประสิทธิภาพของกฎโดยการทดสอบกับโจทย์ที่ใช้ในการทดลอง วัดค่า NFE และความเร็วในการลู่เข้าคำตอบ และ ประสิทธิภาพของคำตอบ ที่ random seed ชุดเดียวกัน โดยเริ่มจากการเปรียบเทียบกันของกฎที่ 1 และ กฎที่ 2 เนื่องจากทั้ง 2 กฎมีการปรับความคลาดเคลื่อนในกรณีที่ฟังก์ชันวัตถุประสงค์ยังไม่ลู่เข้าเพียงอย่างเดียวเหมือนกัน ซึ่งประสิทธิภาพของกฎวัดได้จากความแม่นยำของคำตอบที่ใกล้กับค่าเหมาะสมที่สุดครอบคลุมมากกว่า และ จำนวนการประเมินฟังก์ชันที่น้อยกกว่า

ตารางที่ 4.4 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 1 และ กฎที่ 2 เทียบกับค่าเหมาะสมที่สุดครอบคลุม

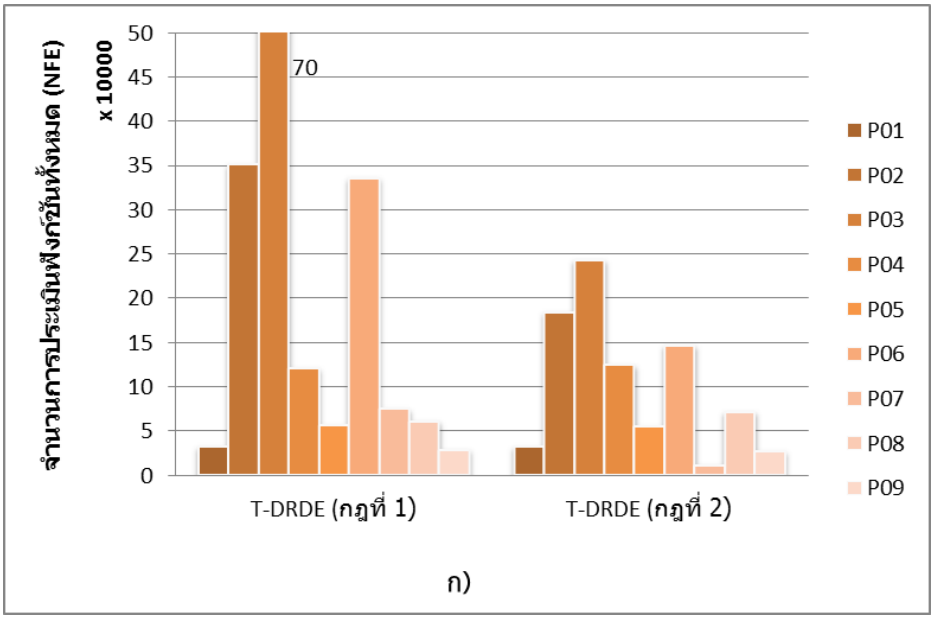
ปัญหา	T-DRDE (กฎที่ 1)			ค่าเหมาะสมที่สุดครอบคลุม, f*	T-DRDE (กฎที่ 2)		
	ดีที่สุด	เฉลี่ย	แย่ที่สุด		ดีที่สุด	เฉลี่ย	แย่ที่สุด
P01	0.0000	0.0000	0.0000	0.0539	0.0000	0.0000	0.0000
P02	5126.4978	5126.4976	5126.4976	5126.4981	5126.4976	5125.1231	5119.6253
P03	99.2390	99.2390	99.2390	99.2396	99.2390	99.1656	98.8719
P04	7.6670	7.6670	7.6670	7.6672	7.6670	7.3117	6.7585
P05	36162.9782	36162.9782	36162.9782	36162.9886	36069.9069	35115.6279	33228.6183
P06	-1.9235	-1.9235	-1.9236	-1.9231	-11.0000	-11.0000	-11.0000
P07	-1.0000	-1.0000	-1.0000	-0.3888	-1.0000	-1.0000	-1.0000
P08	2100.0000	2100.0000	2100.0000	7049.2500	2100.0000	2100.0000	2100.0000
P09	0.9979	0.9979	0.9979	1.8640	0.9979	0.9979	0.9979

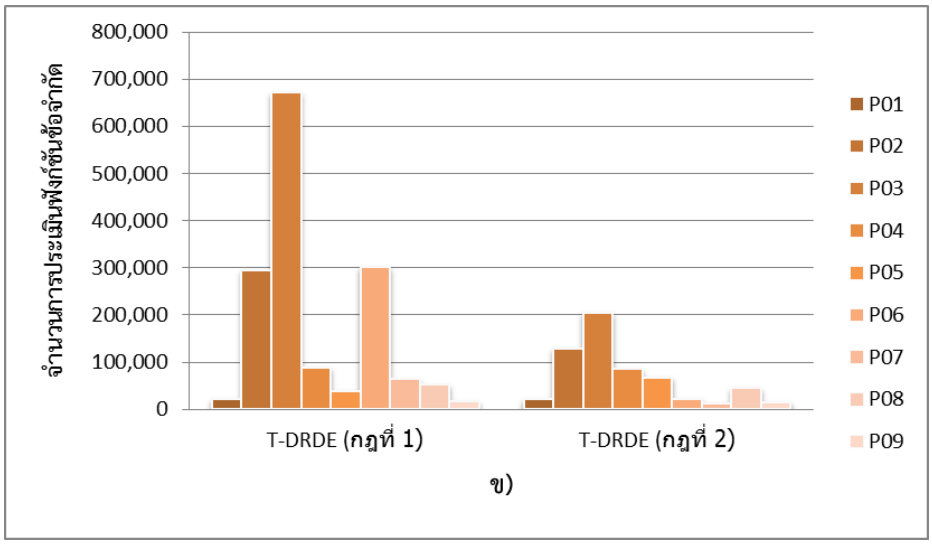


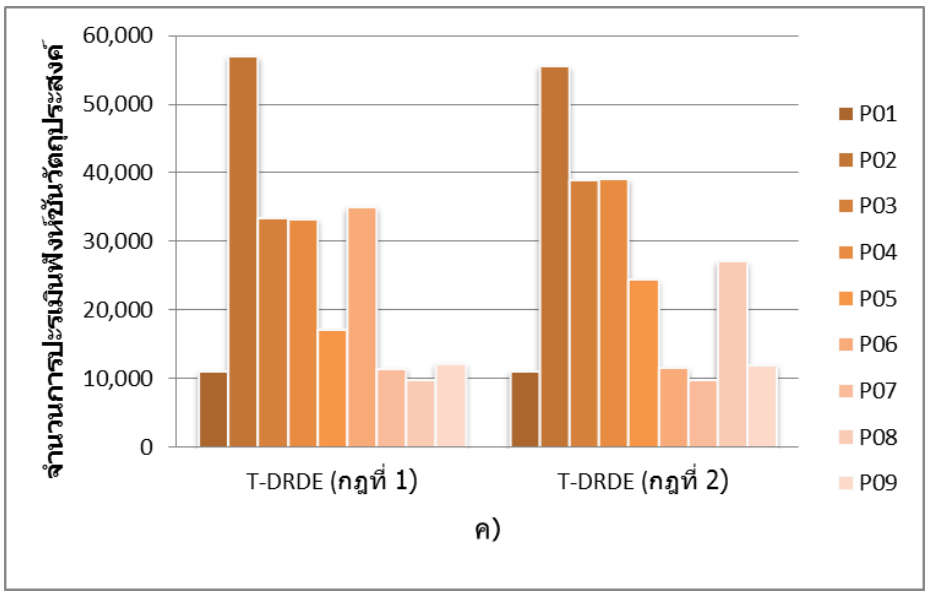
ภาพที่ 4.1 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 1 และ 2

ตารางที่ 4.5 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 1 และ 2 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ ตามลำดับ

ก) จำนวนการประเมินฟังก์ชันทั้งหมด			ข) จำนวนการประเมินฟังก์ชันข้อจำกัด			ค) จำนวนการประเมินฟังก์ชันวัตถุประสงค์		
ปัญหา	T-DRDE (กฎที่ 1)	T-DRDE (กฎที่ 2)	ปัญหา	T-DRDE (กฎที่ 1)	T-DRDE (กฎที่ 2)	ปัญหา	T-DRDE (กฎที่ 1)	T-DRDE (กฎที่ 2)
P01	31,937	31,937	P01	20,937	20,937	P01	11,000	11,000
P02	350,575	184,108	P02	293,653	128,652	P02	56,922	55,456
P03	704,068	243,173	P03	670,790	204,309	P03	33,277	38,864
P04	120,534	124,157	P04	87,294	85,128	P04	33,240	39,029
P05	55,869	54,843	P05	38,869	67,072	P05	17,000	24,333
P06	335,522	146,723	P06	300,572	21,211	P06	34,950	11,534
P07	74,635	11,408	P07	63,199	13,044	P07	11,436	9,772
P08	60,988	71,533	P08	51,313	44,500	P08	9,675	27,033
P09	28,563	26,787	P09	16,396	14,800	P09	12,166	11,988







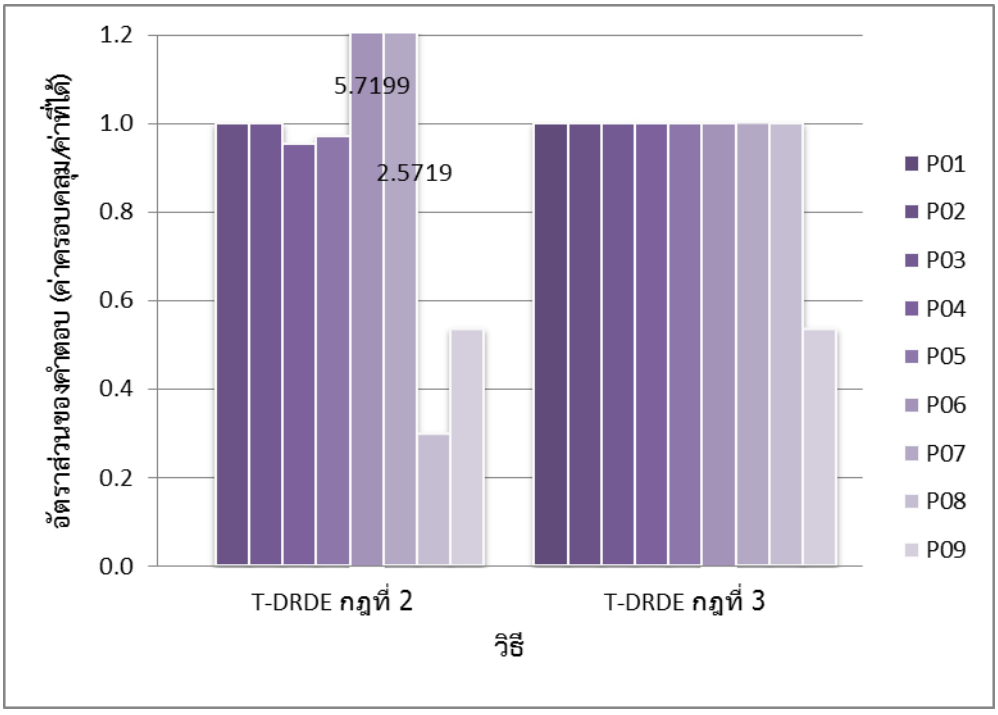
ภาพที่ 4.2 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 1 และ 2 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ตามลำดับ

จากภาพที่ 4.1 จะเห็นว่าวิธี T-DRDE ทั้งกฎที่ 1 และ 2 นั้นให้ความแม่นยำของคำตอบในแต่ละข้อที่ใกล้เคียงกัน โดยกฎที่ 1 ยังไม่สามารถหาคำตอบได้ในปัญหา P01 P07 P08 และ P09 ส่วนกฎที่ 2 ยังไม่สามารถหาคำตอบได้ในปัญหา P01 P06 P07 P08 และ P09 ในเบื้องต้นแล้วกฎที่ 1 ดูจะสามารถหาคำตอบได้ดีกว่าเฉพาะปัญหา P06 แต่เมื่อพิจารณาที่จำนวนการประเมินฟังก์ชันทั้งหมดจากภาพที่ 4.2 ก) นั้นพบว่ากฎที่ 2 มีจำนวนน้อยกว่ากฎที่ 1 ซึ่งเป็นปัจจัยหลักในการพิจารณา จึงสรุปได้ว่ากฎที่ 2 จึงมีประสิทธิภาพที่ดีกว่ากฎที่ 1 แต่อย่างไรก็ตามในทางปฏิบัติควรจะ

สามารถใช้แก้ปัญหาได้มากกว่านี้ ดังนั้นจึงได้นำวิธี T-DRDE กฎที่ 2 มาเปรียบเทียบกับ กฎที่ 3 ต่อไป ซึ่งเป็นการเทียบกันระหว่างวิธีที่ปรับค่าคลาดเคลื่อนเฉพาะกรณีที่ฟังก์ชันวัตถุประสงค์ยังไม่ลู่เข้าเพียงอย่างเดียวกับกรณีที่ปรับค่าคลาดเคลื่อนทั้งการลู่เข้า และ ไม่ลู่เข้า

ตารางที่ 4.6 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 2 และ กฎที่ 3 เทียบกับค่าเหมาะสมที่สุดครอบคลุม

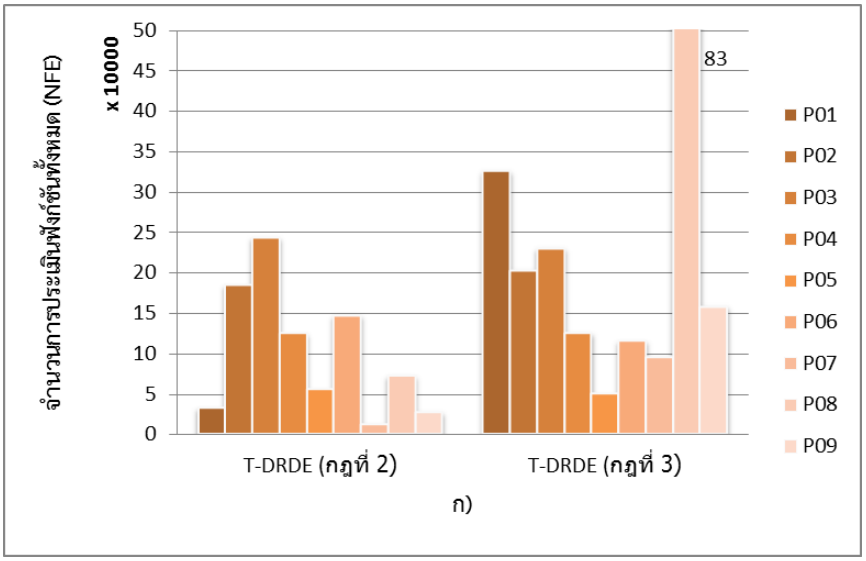
ปัญหา	T-DRDE (กฎที่ 2)			ค่าเหมาะสมที่สุดครอบคลุม, f*	T-DRDE (กฎที่ 3)		
	ดีที่สุด	เฉลี่ย	แย่ที่สุด		ดีที่สุด	เฉลี่ย	แย่ที่สุด
P01	0.0000	0.0000	0.0000	0.0539	0.0540	0.0540	0.0539
P02	5126.4976	5125.1231	5119.6253	5126.4981	5126.4976	5126.4976	5126.4976
P03	99.2390	99.1656	98.8719	99.2396	99.2390	99.2390	99.2390
P04	7.6670	7.3117	6.7585	7.6672	7.6670	7.6670	7.6670
P05	36069.9069	35115.6279	33228.6183	36162.9886	36162.9782	36162.9782	36162.9782
P06	-11.0000	-11.0000	-11.0000	-1.9231	-1.9232	-1.9234	-1.9235
P07	-1.0000	-1.0000	-1.0000	-0.3888	-0.3885	-0.3886	-0.3888
P08	2100.0000	2100.0000	2100.0000	7049.2500	7062.1071	7054.6814	7049.0521
P09	0.9979	0.9979	0.9979	1.8640	0.9979	0.9979	0.9979

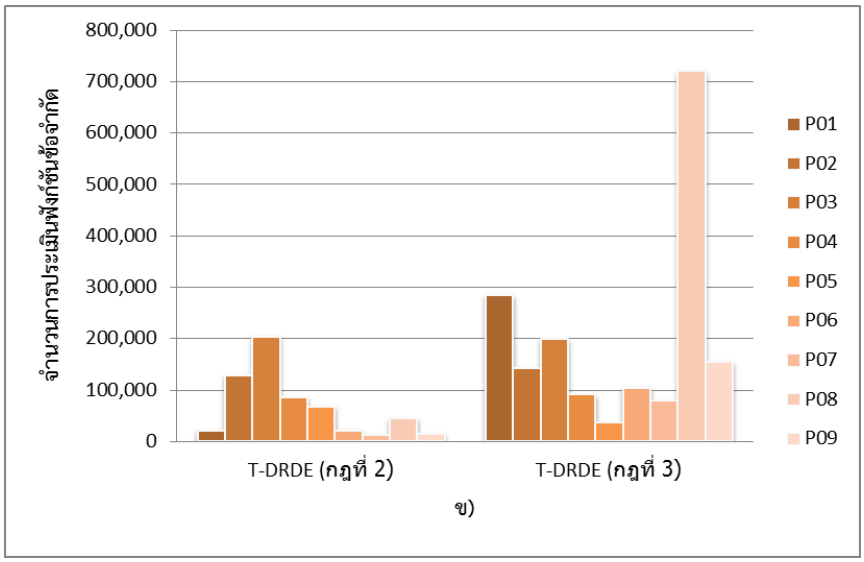


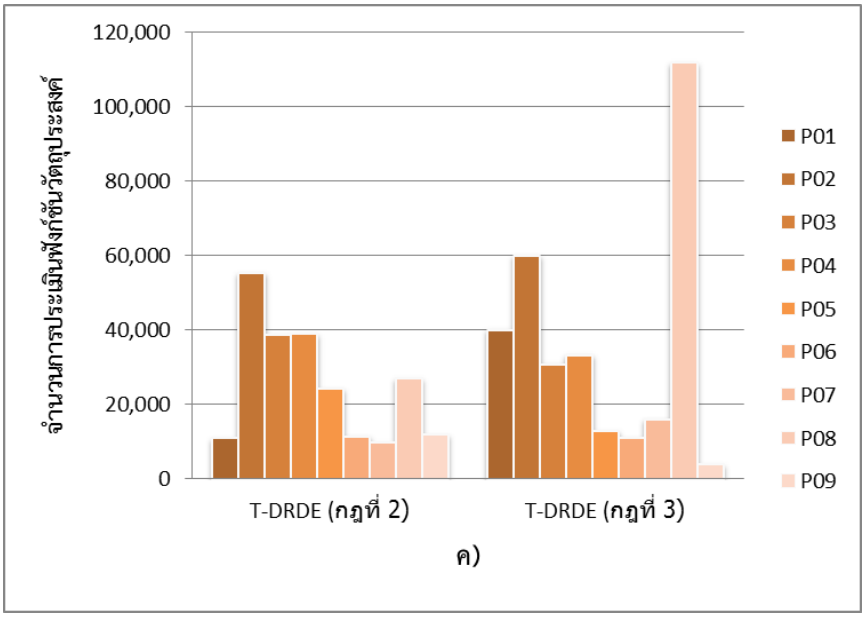
ภาพที่ 4.3 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 2 และ 3

ตารางที่ 4.7 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 2 และ 3 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ ตามลำดับ

ก) จำนวนการประเมินฟังก์ชันทั้งหมด			ข) จำนวนการประเมินฟังก์ชันข้อจำกัด			ค) จำนวนการประเมินฟังก์ชันวัตถุประสงค์		
ปัญหา	T-DRDE (กฎที่ 2)	T-DRDE (กฎที่ 3)	ปัญหา	T-DRDE (กฎที่ 2)	T-DRDE (กฎที่ 3)	ปัญหา	T-DRDE (กฎที่ 2)	T-DRDE (กฎที่ 3)
P01	31,937	325,422	P01	20,937	285,222	P01	11,000	40,200
P02	184,108	202,690	P02	128,652	142,540	P02	55,456	60,150
P03	243,173	229,753	P03	204,309	198,806	P03	38,864	30,948
P04	124,157	124,717	P04	85,128	91,304	P04	39,029	33,413
P05	54,843	49,442	P05	67,072	36,442	P05	24,333	13,000
P06	146,723	115,149	P06	21,211	103,992	P06	11,534	11,157
P07	11,408	95,192	P07	13,044	79,101	P07	9,772	16,091
P08	71,533	832,178	P08	44,500	720,266	P08	27,033	111,912
P09	26,787	157,942	P09	14,800	154,016	P09	11,988	3,926







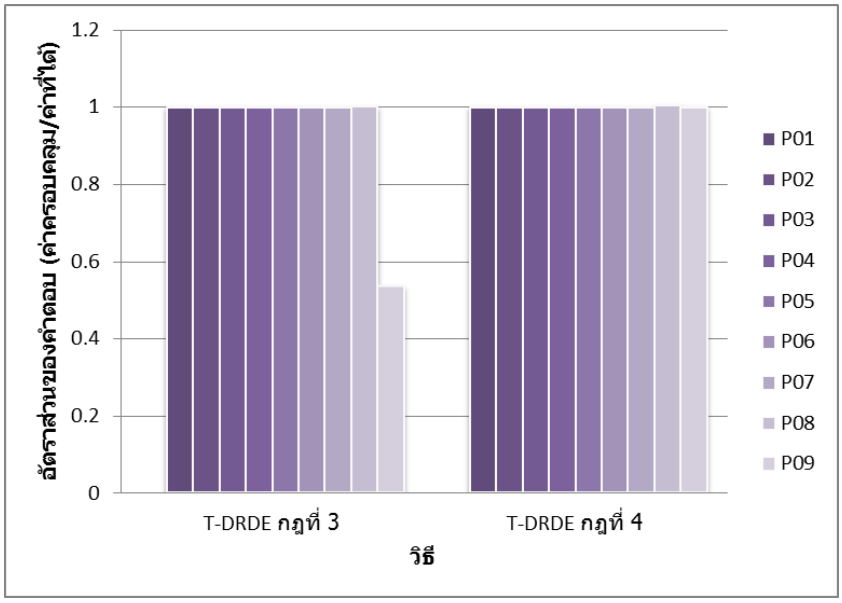
ภาพที่ 4.4 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 2 และ 3 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ตามลำดับ

จากภาพที่ 4.3 จะเห็นว่าวิธี T-DRDE ทั้งกฎที่ 2 และ 3 นั้น กฎที่ 3 มีความแม่นยำของคำตอบที่ดีกว่ากฎที่ 2 มาก โดยกฎที่ 2 ยังไม่สามารถหาคำตอบได้ในปัญหา P01 P06 P07 P08 และ P09 แต่กฎที่ 3 นั้นสามารถหาคำตอบได้ทุกปัญหายกเว้น P09 ซึ่งเมื่อพิจารณาจากภาพที่ 4.4 แล้วพบว่าในปัญหาที่ทั้ง 2 กฎสามารถหาคำตอบได้นั้นมีจำนวนการประเมินฟังก์ชันที่ใกล้เคียงกัน ส่วนในปัญหาที่กฎที่ 3 สามารถแก้ปัญหาได้นั้นมีจำนวนการประเมินฟังก์ชันที่มากกว่า ซึ่งถือเป็น

เรื่องปกติเนื่องจากกฎที่ไม่สามารถหาคำตอบได้นั้นคำตอบจะติดอยู่ที่ค่าเหมาะสมที่สุดเฉพาะที่ ซึ่งมีการลู่เข้าคำตอบที่เร็วกว่าการลู่เข้าคำตอบที่เหมาะสมที่สุดครอบคลุม ดังนั้นจำนวนการประเมินฟังก์ชันของทั้งสองกฎในกรณีหลังจึงไม่สามารถนำมาเปรียบเทียบกันได้ เมื่อมองเฉพาะคุณภาพของคำตอบได้ก็สามารถสรุปได้ว่าวิธี T-DRDE กฎที่ 3 มีประสิทธิภาพดีกว่าวิธี T-DRDE กฎที่ 2 แต่อย่างไรก็ตามเพื่อทดสอบหากฎที่สามารถเทียบกับวิธีอื่นๆ ได้ กฎที่ใช้ควรที่แก้ปัญหาได้ดีกว่านั้น ดังนั้นมีการทดสอบเทียบต่อระหว่างวิธี T-DRDE กฎที่ 3 และ กฎที่ 4 ต่อไป

ตารางที่ 4.8 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 3 และ กฎที่ 4 เทียบกับค่าเหมาะสมที่สุดครอบคลุม

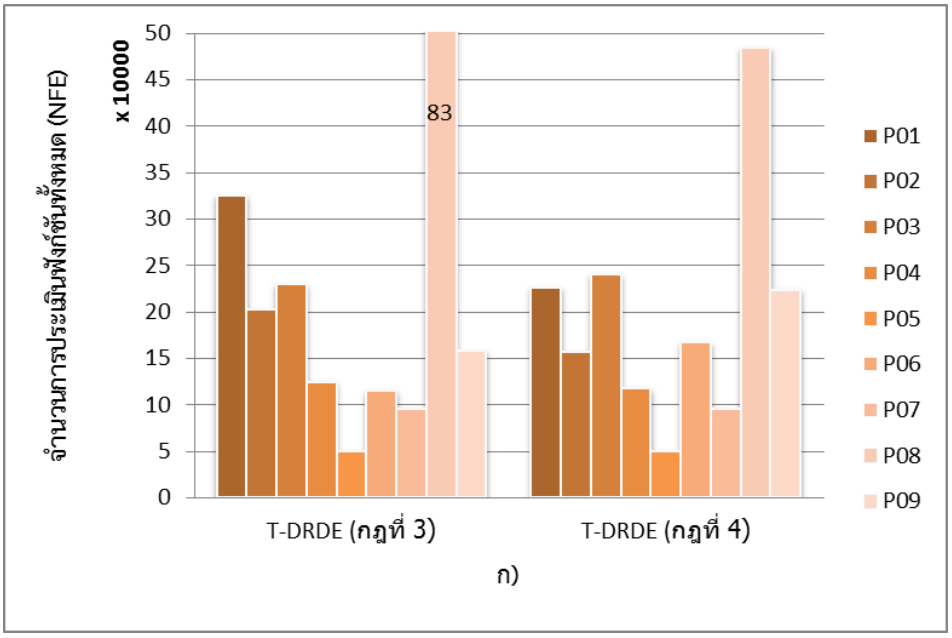
ปัญหา	T-DRDE (กฎที่ 3)			ค่าเหมาะสมที่สุดครอบคลุม, f*	T-DRDE (กฎที่ 4)		
	ดีที่สุด	เฉลี่ย	แย่ที่สุด		ดีที่สุด	เฉลี่ย	แย่ที่สุด
P01	0.0540	0.0540	0.0539	0.0539	0.0540	0.0540	0.0539
P02	5126.4976	5126.4976	5126.4976	5126.4981	5126.4978	5126.4976	5126.4976
P03	99.2390	99.2390	99.2390	99.2396	99.2390	99.2390	99.2390
P04	7.6670	7.6670	7.6670	7.6672	7.6670	7.6670	7.6670
P05	36162.9782	36162.9782	36162.9782	36162.9886	36162.9782	36162.9782	36162.9782
P06	-1.9232	-1.9234	-1.9235	-1.9231	-1.9232	-1.9235	-1.9236
P07	-0.3885	-0.3886	-0.3888	-0.3888	-0.3885	-0.3886	-0.3888
P08	7062.1071	7054.6814	7049.0521	7049.2500	7101.7945	7079.4322	7055.4707
P09	0.9979	0.9979	0.9979	1.8640	1.8642	1.8642	1.8642



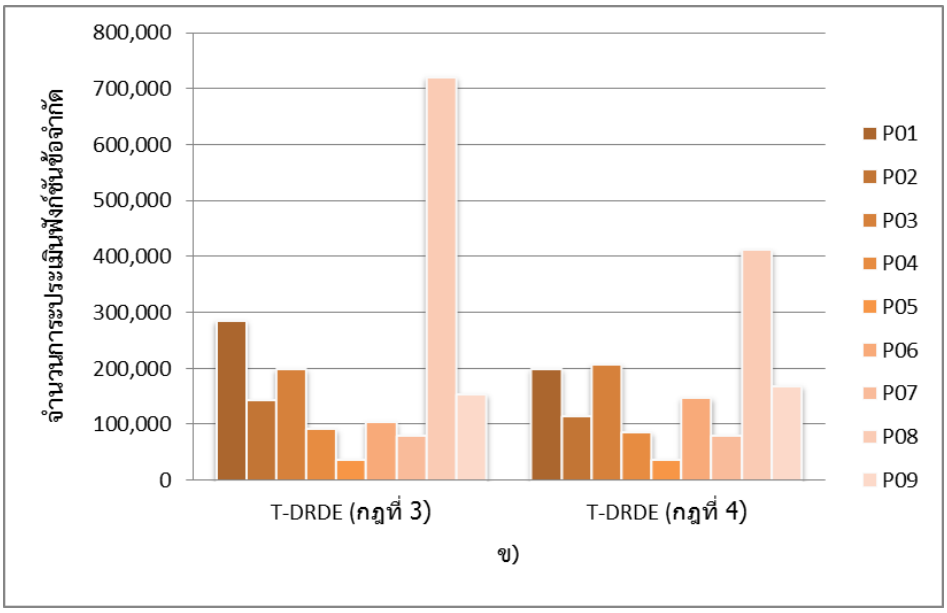
ภาพที่ 4.5 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 3 และ 4

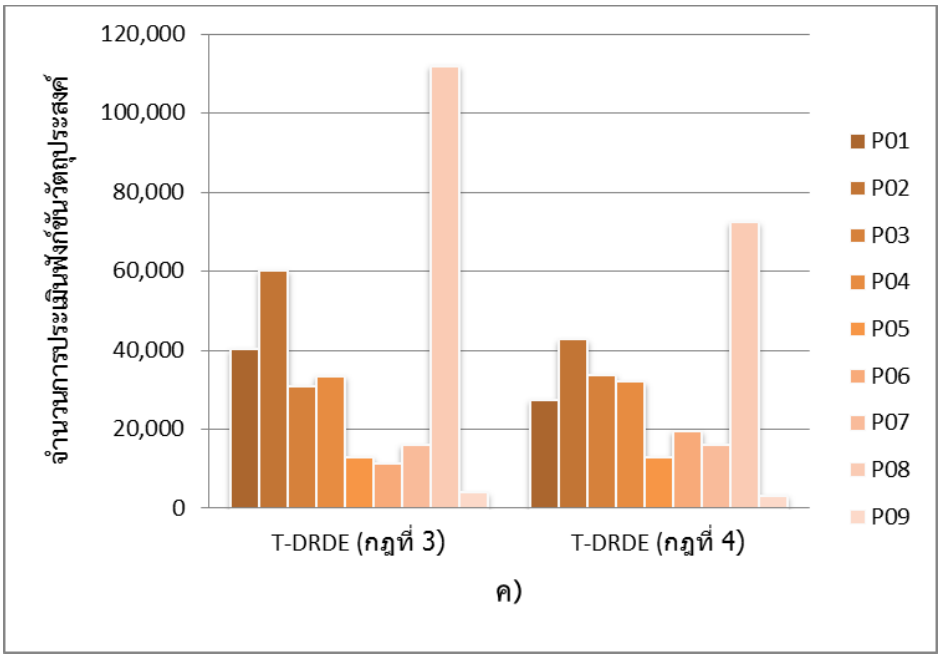
ตารางที่ 4.9 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 3 และ 4 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ ตามลำดับ

ก) จำนวนการประเมินฟังก์ชันทั้งหมด			ข) จำนวนการประเมินฟังก์ชันข้อจำกัด			ค) จำนวนการประเมินฟังก์ชันวัตถุประสงค์		
ปัญหา	T-DRDE (กฎที่ 3)	T-DRDE (กฎที่ 4)	ปัญหา	T-DRDE (กฎที่ 3)	T-DRDE (กฎที่ 4)	ปัญหา	T-DRDE (กฎที่ 3)	T-DRDE (กฎที่ 4)
P01	325,422	226,477	P01	285,222	199,077	P01	40,200	27,400
P02	202,690	156,644	P02	142,540	113,974	P02	60,150	42,670
P03	229,753	240,559	P03	198,806	206,915	P03	30,948	33,645
P04	124,717	117,167	P04	91,304	85,210	P04	33,413	31,957
P05	49,442	49,442	P05	36,442	36,442	P05	13,000	13,000
P06	115,149	167,134	P06	103,992	147,567	P06	11,157	19,566
P07	95,192	95,192	P07	79,101	79,101	P07	16,091	16,091
P08	832,178	483,659	P08	720,266	411,293	P08	111,912	72,365
P09	157,942	223,889	P09	154,016	168,280	P09	3,926	3,168



ก)



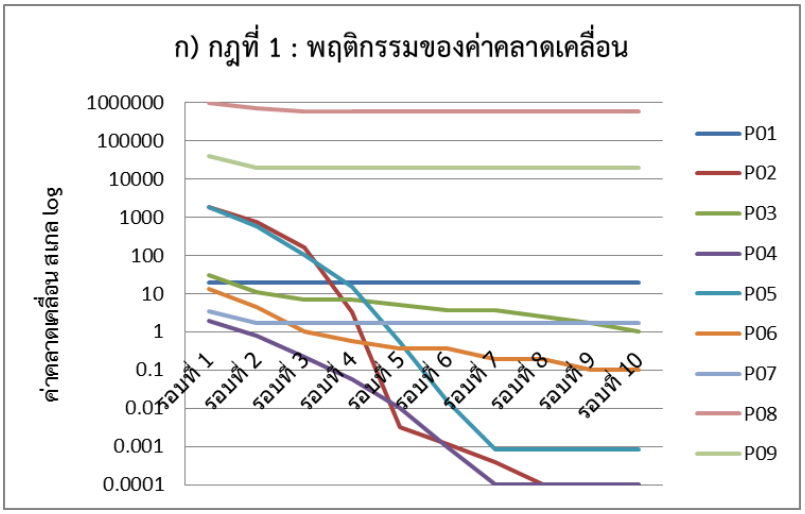


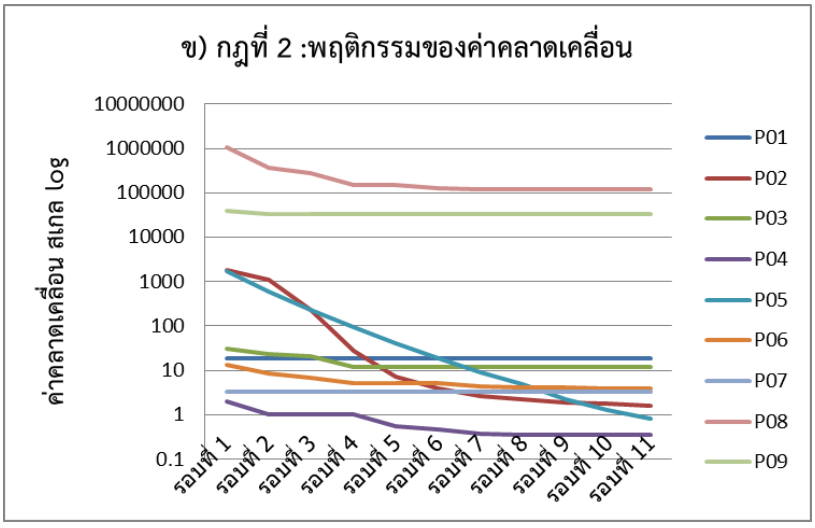
ภาพที่ 4.6 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 3 และ 4 โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ตามลำดับ

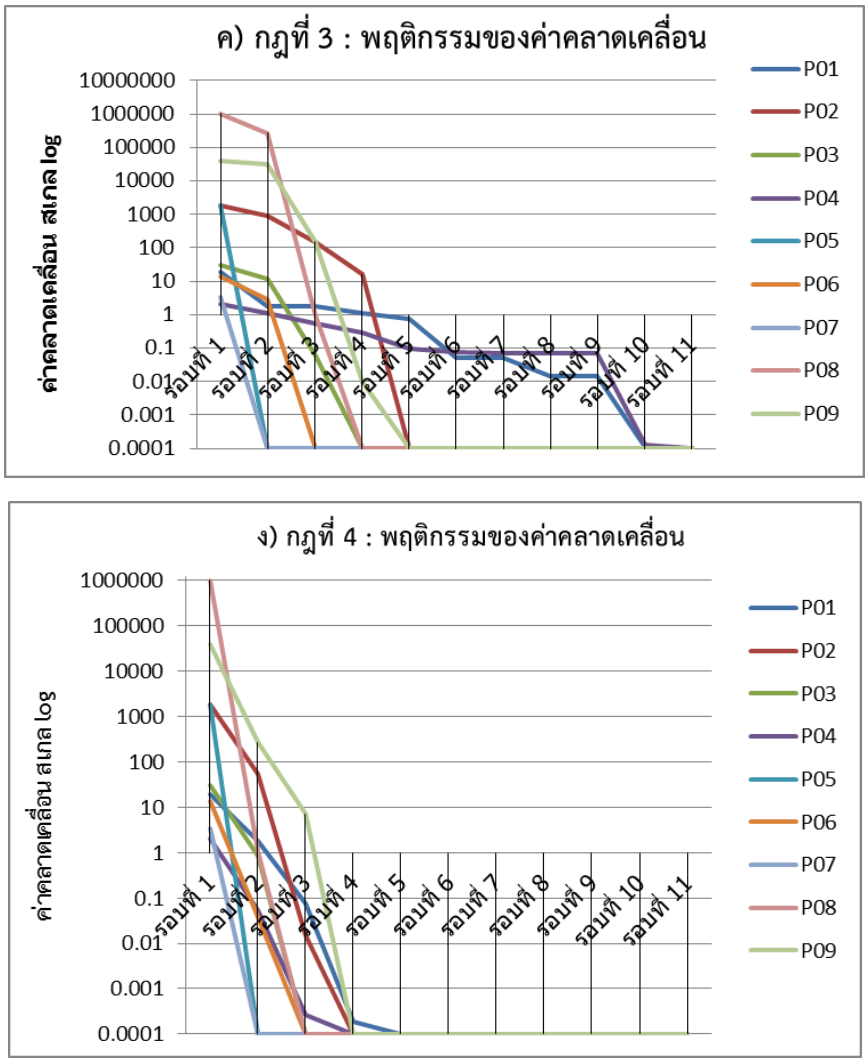
จากภาพที่ 4.5 จะเห็นว่าวิธี T-DRDE ทั้งกฎที่ 3 และ 4 นั้น กฎที่ 4 สามารถแก้ปัญหาที่ใช้ทดสอบได้ทุกปัญหาซึ่งเดิมกฎที่ 3 ไม่สามารถแก้ปัญหา P09 ได้ และ เมื่อพิจารณาจากภาพที่ 4.6 จะพบกว่าจำนวนการประเมินฟังก์ชันจากกฎที่ 4 มีจำนวนน้อยกว่ากฎที่ 3 ทุกข้อ โดยเฉพาะอย่างยิ่งในปัญหา P08 ซึ่งกฎที่ 3 ใช้จำนวนการประเมินฟังก์ชันประมาณ 830,000 ครั้ง ในขณะที่กฎที่ 4

ใช้จำนวนการประเมินฟังก์ชันประมาณ 450,000 ครั้ง นั่นจึงสามารถสรุปได้ว่าวิธี T-DRDE ด้วยกฎที่ 4 มีประสิทธิในการแก้ปัญหาที่ใช้ทดสอบดีกว่ากฎที่ 3 ต่อไปจึงได้นำผลลัพธ์จากกฎที่ 4 ใช้เปรียบเทียบกับวิธี DRDE

เมื่อวิเคราะห์พฤติกรรมการปรับค่าคลาดเคลื่อนในแต่ละกฎจากภาพที่ 4.7 ก) – ง) จะพบว่าพฤติกรรมของค่าคลาดเคลื่อนจากกฎที่ 1 และ 2 เป็นในแนวทางเดียวกัน คือ มีการปรับตัวเข้าสู่ค่าคลาดเคลื่อนต่ำสุด (0.0001) ในรอบที่ 10 ขึ้นไป ซึ่งเป็นการปรับลดที่ช้า กว่ากฎที่ 3 และ 4 โดยที่กฎที่ 4 มีการปรับลดค่าคลาดเคลื่อนเร็วที่สุด ประมาณ 1-2 รอบการทำงานเท่านั้น ซึ่งจากการวิเคราะห์ที่ผ่านมานั้น ประสิทธิภาพที่ดีที่สุด คือ วิธี T-DRDE จากกฎที่ 4 ดังนั้นจึงสามารถสรุปได้ว่าพฤติกรรมการปรับลดค่าคลาดเคลื่อนที่จะทำให้มีประสิทธิภาพดีที่สุดในปัญหาที่ใช้ทดสอบจะมีการปรับลดค่าที่เร็วอยู่ที่ประมาณไม่เกิน 2 รอบของการทำงาน





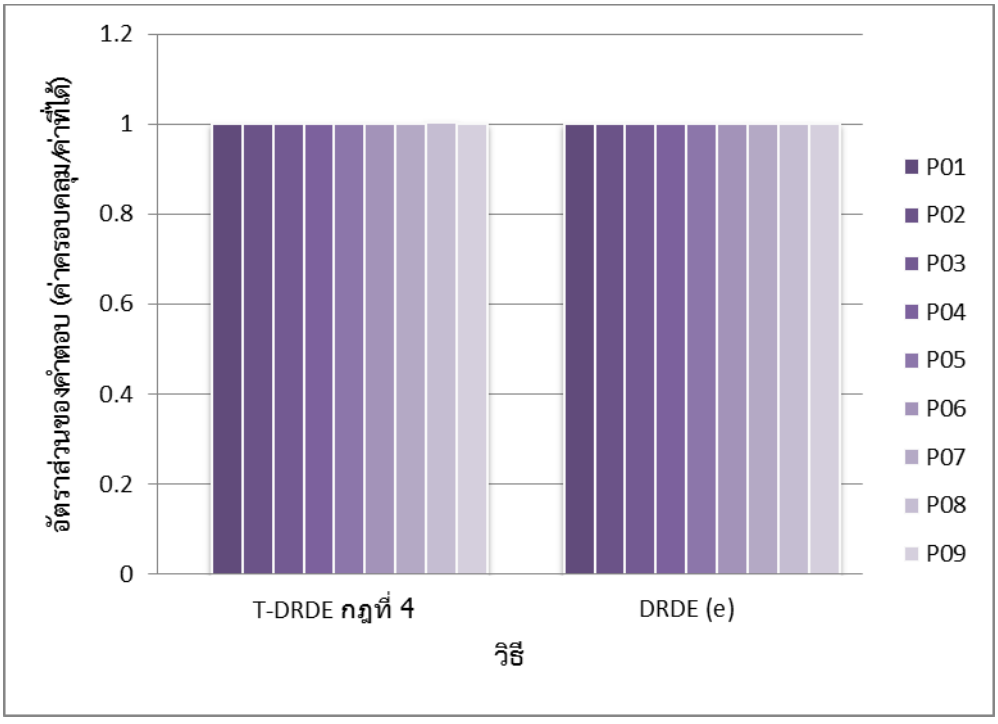


ภาพที่ 4.7 พฤติกรรมของค่าคลาดเคลื่อนในแต่ละรอบการวนซ้ำของกฎทั้ง 4 โดยมีวิธี T-DRDE ก) คือ กฎที่ 1 ข) คือ กฎที่ 2 ค) คือ กฎที่ 3 และ ง) คือ กฎที่ 4 ตามลำดับ

4.3.2 เปรียบเทียบประสิทธิภาพของวิธี T-DRDE เทียบกับวิธี DRDE และ C-IDE

ตารางที่ 4.10 คำตอบที่ได้ของวิธี T-DRDE กฎที่ 4 และ วิธี DRDE เทียบกับค่าเหมาะสมที่สุดครอบคลุม

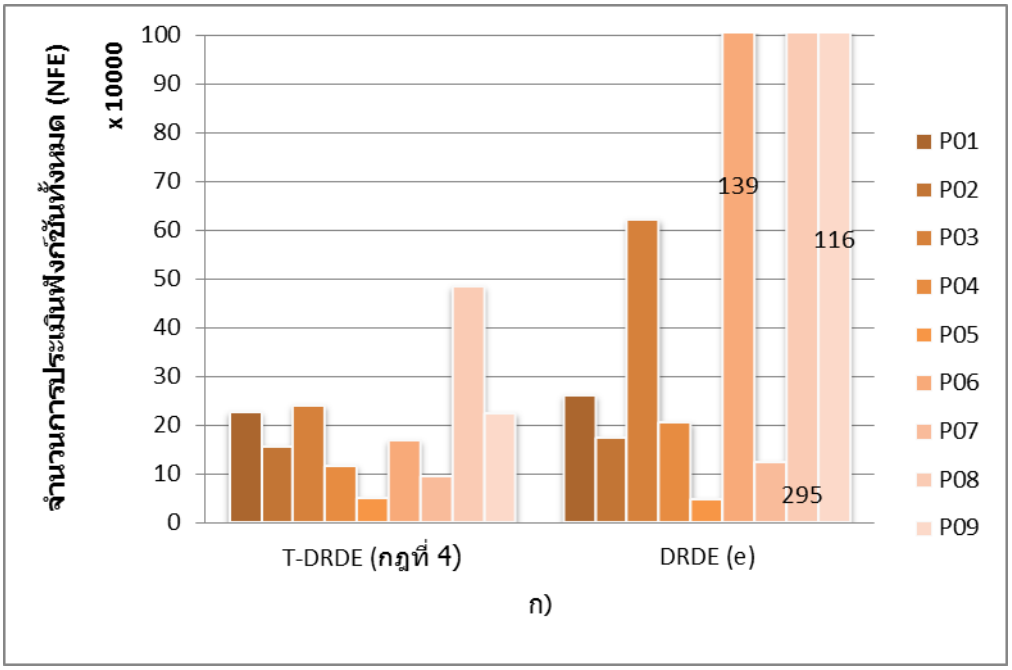
ปัญหา	T-DRDE (กฎที่ 4)			ค่าเหมาะสมที่สุดครอบคลุม, f*	DRDE (e)		
	ดีที่สุด	เฉลี่ย	แย่ที่สุด		ดีที่สุด	เฉลี่ย	แย่ที่สุด
P01	0.0540	0.0540	0.0539	0.0539	0.0540	0.0540	0.0539
P02	5126.4978	5126.4976	5126.4976	5126.4981	5126.4976	5126.4976	5126.4976
P03	99.2390	99.2390	99.2390	99.2396	99.2390	99.2390	99.2390
P04	7.6670	7.6670	7.6670	7.6672	7.6670	7.6670	7.6670
P05	36162.9782	36162.9782	36162.9782	36162.9886	36162.9782	36162.9782	36162.9782
P06	-1.9232	-1.9235	-1.9236	-1.9231	-1.9232	-1.9234	-1.9236
P07	-0.3885	-0.3886	-0.3888	-0.3888	-0.3886	-0.3887	-0.3888
P08	7101.7945	7079.4322	7055.4707	7049.2500	7067.1530	7054.2419	7048.9485
P09	1.8642	1.8642	1.8642	1.8640	1.8642	1.8642	1.8642



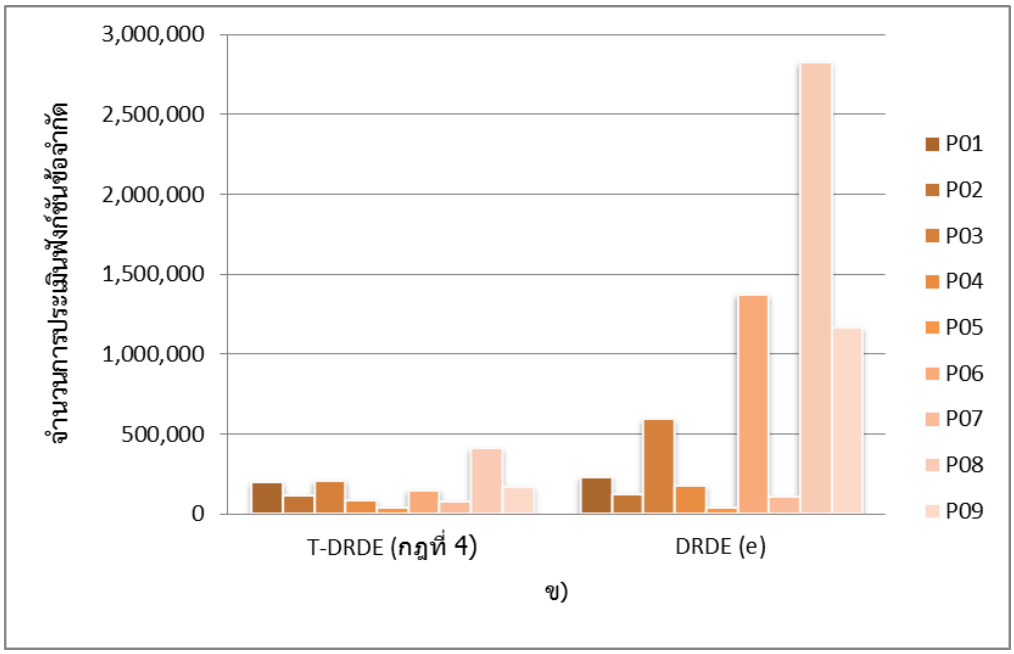
ภาพที่ 4.8 กราฟความสัมพันธ์ระหว่างอัตราส่วนของคำตอบกับวิธี T-DRDE กฎที่ 4 และ วิธี DRDE

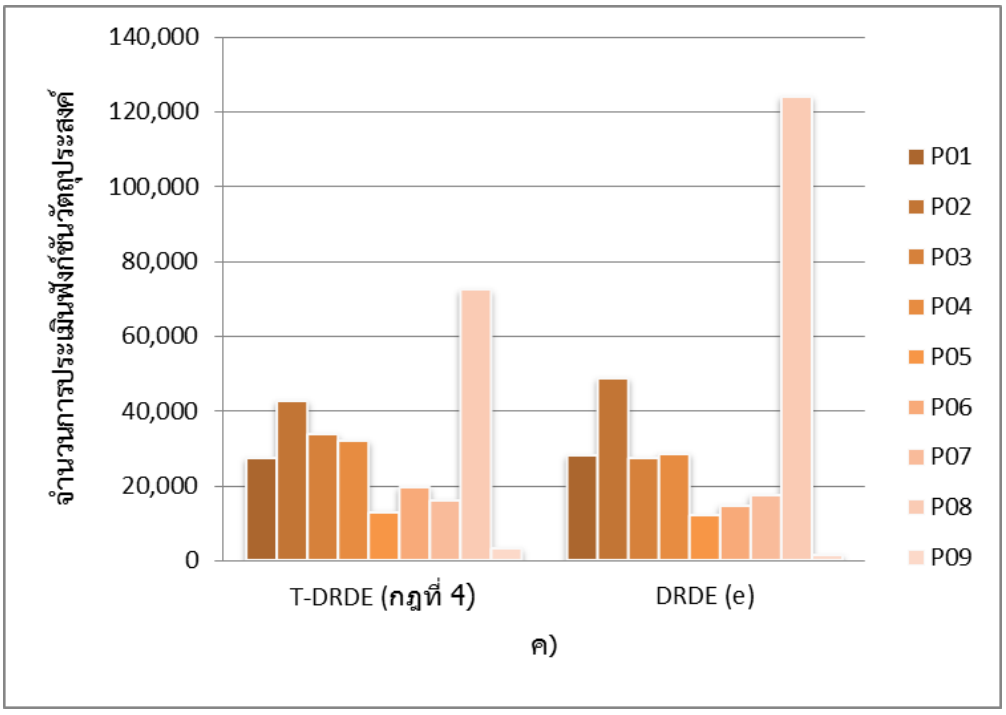
ตารางที่ 4.11 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 4 และ วิธี DRDE โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ ตามลำดับ

ก) จำนวนการประเมินฟังก์ชันทั้งหมด			ข) จำนวนการประเมินฟังก์ชันข้อจำกัด			ค) จำนวนการประเมินฟังก์ชันวัตถุประสงค์		
ปัญหา	T-DRDE (กฎที่ 4)	DRDE (e)	ปัญหา	T-DRDE (กฎที่ 4)	DRDE (e)	ปัญหา	T-DRDE (กฎที่ 4)	DRDE (e)
P01	226,477	260,395	P01	199,077	232,395	P01	27,400	28,000
P02	156,644	174,627	P02	113,974	125,896	P02	42,670	48,732
P03	240,559	621,001	P03	206,915	593,538	P03	33,645	27,463
P04	117,167	204,761	P04	85,210	176,189	P04	31,957	28,572
P05	49,442	48,252	P05	36,442	36,252	P05	13,000	12,000
P06	167,134	1,385,779	P06	147,567	1,371,082	P06	19,566	14,697
P07	95,192	123,741	P07	79,101	106,275	P07	16,091	17,466
P08	483,659	2,948,873	P08	411,293	2,824,813	P08	72,365	124,060
P09	223,889	1,163,682	P09	168,280	1,162,325	P09	3,168	1,357



ก)





ภาพที่ 4.9 กราฟความสัมพันธ์ระหว่างจำนวนการประเมินฟังก์ชันวิธี T-DRDE กฎที่ 4 และ DRDE โดยที่ ก) คือ จำนวนการประเมินฟังก์ชันทั้งหมด ข) คือ จำนวนการประเมินฟังก์ชันข้อจำกัด และ ค) คือ จำนวนการประเมินฟังก์ชันวัตถุประสงค์ตามลำดับ

จากภาพที่ 4.11 จะเห็นว่าวิธี T-DRDE ทั้งกฎที่ 4 และ วิธี DRDE นั้น มีคุณภาพของคำตอบที่ใกล้เคียงกัน โดยทั้งสองวิธีสามารถหาค่าเหมาะสมที่สุดครอบคลุมได้ทุกปัญหาที่ใช้ทดสอบ และ เมื่อพิจารณาถึงจำนวนการประเมินฟังก์ชันจากภาพที่ 4.9 ก) พบว่าวิธี T-DRDE กฎที่ 4 มีจำนวนการประเมินฟังก์ชันที่น้อยกว่าวิธี DRDE เกือบทุกข้อ โดยเฉพาะอย่างยิ่งกับปัญหา P03 P04 P06 P08 และ P09 ซึ่งล้วนแต่เป็นปัญหาที่มีบริเวณของการค้นคำตอบที่กว้างเนื่องจากมีจำนวนของอสมการข้อจำกัดที่มาก และ เมื่อพิจารณาภาพที่ 4.9 ข) และ ค) พบว่าจำนวนการประเมินฟังก์ชันวัตถุประสงค์ของวิธีทั้งสองนั้นมีจำนวนที่ใกล้เคียงกัน ส่วนจำนวนการประเมินฟังก์ชันข้อจำกัดนั้นวิธี T-DRDE กฎที่ 4 มีจำนวนที่น้อยกว่าวิธี DRDE นั่นสามารถสรุปได้ว่าวิธี T-DRDE มีประสิทธิภาพมากกว่าวิธี DRDE ในการแก้ปัญหาที่ใช้ทดสอบโดยเฉพาะอย่างยิ่งในปัญหาที่มีจำนวนอสมการข้อจำกัดมาก โดยการทำค่าคลาดเคลื่อนยินยอมแบบยืดหยุ่นนี้จะมีผลช่วยลดจำนวนการประเมินฟังก์ชันข้อจำกัดเป็นส่วนใหญ่ ซึ่งเป็นผลที่เกิดจากกระบวนการซ่อมแซมเป็นหลัก

ตารางที่ 4.12 จำนวนการประเมินฟังก์ชันของวิธี T-DRDE กฎที่ 4 และ วิธี C-IDE เทียบกับค่าเหมาะสมที่สุดครอบคลุม

ปัญหา	T-DRDE @ 20 NP		C-IDE	ค่าเหมาะสมที่สุดครอบคลุม, f*
	NFE	Result	NFE	
P05	839	36162.9782	11,524	36162.9886
P07	1,807	-0.3881	11,604	-0.3888
P08	16,353	7049.1934	15,632	7049.2500
P09	7,952	21693.0000	16,338	1.8640

ผลลัทธ์ที่ได้ค่อนข้างชัดเจนว่าการนำขั้นตอนวิธี T-DRDE ไปใช้แก้ปัญหา P05 และ P07 นั้น ให้จำนวนการประเมินฟังก์ชันที่น้อยกว่าเมื่อเทียบกับวิธี C-IDE ที่จำนวนประชากรเท่ากัน คือ 20 ประชากร ส่วนปัญหา P08 นั้นยังไม่สามารถสรุปได้แน่ชัด เนื่องจากจำนวนการประเมินฟังก์ชันของปัญหาข้อนี้ขึ้นอยู่กับคุณภาพของคำตอบค่อนข้างมากซึ่งต้องทราบก่อนว่าคำตอบของ C-IDE ประมาณเท่าไหร่ ส่วนปัญหา P09 ที่ให้ผลลัพธ์ไม่ค่อยตรงกับค่าเหมาะสมที่สุดครอบคลุม อาจเป็นเพราะปัญหาข้อนี้มีความไม่ต่อเนื่องสูง ซึ่งวิธีการ T-DRDE นั้นการจะแก้ปัญหาประเภทนี้จำเป็นต้อง

อาศัยจำนวนประชากรที่มากกว่านี้ เพราะกระบวนการซ่อมแซมภายในมีแนวโน้มที่จะดึงกลุ่มประชากรไว้ให้บริเวณที่เป็นไปได้แยกจากบริเวณอื่นไป

ตารางที่ 4.13 จำนวนการประเมินฟังก์ชันทั้งหมดของวิธีต่างๆ โดยสรุป

ปัญหา	T-DRDE (กฎที่ 1)	T-DRDE (กฎที่ 2)	T-DRDE (กฎที่ 3)	T-DRDE (กฎที่ 4)	DRDE (e)	T-DRDE @20NP	C-IDE
	จำนวนการประเมินฟังก์ชันทั้งหมด (NFE)						
P01	31,937	31,937	325,422	226,477	260,395		
P02	350,575	184,108	202,690	156,644	174,627		
P03	704,068	243,173	229,753	240,559	621,001		
P04	120,534	124,157	124,717	117,167	204,761		
P05	55,869	55,161	49,442	49,442	48,252	839	11524
P06	335,522	146,723	115,149	167,134	1,385,779		
P07	74,635	117,041	95,192	95,192	123,741	1807	11604
P08	60,988	71,533	832,178	483,659	2,948,873	16353	15632
P09	28,563	26,787	157,942	223,889	1,163,682	7952	16338

บทที่ 5

สรุปผลการวิจัย

5.1 สรุปผลการวิจัย

จากผลการทดลองตอนที่ 4.3.1 เป็นการทดสอบประสิทธิภาพของกฎการปรับค่าคลาดเคลื่อนยินยอม โดยมีกฎที่ใช้ทดสอบอยู่ 4 แบบ ดังหัวข้อที่ 4.2 เมื่อทำการวัดจำนวนการประเมินฟังก์ชันของทั้ง 4 กฎเปรียบเทียบกันก็พบว่า กฎการปรับค่าคลาดเคลื่อนที่ 1 และ 2 ให้ความแม่นยำของคำตอบยังไม่ดี แต่กฎที่ 2 ใช้ค่า NFE ที่น้อยกว่า ส่วนกฎที่ 3 ให้ความแม่นยำของคำตอบที่ดีกว่ากฎที่ 1 และ 2 แต่ยังใช้ค่า NFE ที่สูงอยู่ และ กฎที่ 4 นั้นให้ความแม่นยำของคำตอบดีที่สุด และ ใช้ค่า NFE น้อยที่สุดด้วย เมื่อสังเกตที่ค่าคลาดเคลื่อนยินยอมที่ลดลงทุกๆ รอบของกฎแต่ละแบบ ก็ทำให้เห็นแนวโน้มของการลดค่าคลาดเคลื่อนยินยอมของกฎแต่ละแบบ เช่นนี้เมื่อพิจารณาดูพฤติกรรมของกฎทั้ง 4 แบบเทียบกัน พบว่าอัตราการลดลงของกฎที่ 1 และ 2 มีการลดลงของค่าคลาดเคลื่อนยินยอมใกล้เคียงกันซึ่งค่อนข้างช้า เมื่อเทียบกับกฎที่ 3 และ 4 โดยกฎที่ 4 มีอัตราการลดลงมากที่สุด เมื่อเป็นเช่นนั้นจึงมีข้อสงสัยว่าอัตราการลดค่าของค่าคลาดเคลื่อนยินยอมมีผลต่อจำนวนการประเมินฟังก์ชันหรือไม่ ถ้าเป็นจริง การมีอัตราการลดค่าคลาดเคลื่อนยินยอมที่สูงกว่าย่อมทำให้จำนวนการประเมินฟังก์ชันลดลงตามด้วย

จากผลการทดลองตอนที่ 4.3.2 เป็นการเปรียบเทียบกันของวิธี T-DRDE กฎที่ 4 กับวิธี DRDE ซึ่งพฤติกรรมของการลดค่าคลาดเคลื่อนยินยอมใกล้เคียงกับวิธี DRDE มาก จะต่างกันเฉพาะในรอบแรกๆ เท่านั้นที่มีการปรับค่าคลาดเคลื่อนยินยอม ซึ่งเมื่อทำการแก้ปัญหาแล้วผลที่ได้ปรากฏว่าค่าจำนวนการประเมินฟังก์ชันของทั้งสองวิธีนั้นแตกต่างกันมาก โดยที่ค่าการประเมินฟังก์ชันของวิธีเดิมนั้นให้ค่าที่สูงกว่ามาก นั่นหมายความว่าจริงๆ แล้วการลดค่าคลาดเคลื่อนยินยอมนั้นมีความสำคัญ ถึงแม้ว่าจะหลังจากผ่านไปรอบเดียวแล้วทุกอย่างจะกลายเป็นเหมือนกับวิธีเดิมเลยก็ตาม นั่นจึงเป็นข้อสงสัยที่ต้องหาข้อสรุปที่ว่า การปรับค่าคลาดเคลื่อนให้ยืดหยุ่นเพียงรอบแล้วส่งผลอย่างไรกับตัวแปรตัดสินใจบ้าง เมื่อมาดูการกระจายตัวของตัวแปรแต่ละตัวจะพบว่าในขั้นตอนการสุ่มเพื่อสร้างเวกเตอร์ต้นแบบนั้น เวกเตอร์ยังมีการกระจายตัวอยู่มาก แต่เมื่อเวกเตอร์ได้เข้าสู่กระบวนการซ่อมแซม ถึงแม้จะทำการซ่อมแซมเพียงแค่ถึงค่าคลาดเคลื่อนยินยอมที่ปล่อยให้หลวม

ในรอบแรกก็ตาม เท่านั้นก็เพียงพอที่จะสร้างความเป็นระเบียบให้กับประชากรเวกเตอร์เป็นอย่างมากแล้ว นั่นอาจเป็นผลมาจากระเบียบวิธีของนิวตัน ที่ทำหน้าที่ดึงประชากรให้เข้าสู่พื้นที่ที่เป็นไปได้ (feasible region) ได้อย่างมีประสิทธิภาพ ซึ่งหากทำการตั้งค่าคลาดเคลื่อนให้น้อยที่สุดตั้งแต่ก่อนเข้ากระบวนการซ่อมแซม นั่นจะทำให้ระเบียบวิธีของนิวตันกระทำต่อเวกเตอร์ที่มาปรับมากเกินความจำเป็น แต่ส่วนที่สำคัญไม่แพ้กันเลย คือ ความสามารถในการพาตัวแปรไปยังจุดที่ดีของตัวดำเนินการในขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง แต่อย่างไรก็ตามการทดลองก็อาจจะถามต่อได้ว่าฉะนั้นค่าเริ่มต้นของค่าคลาดเคลื่อนมีผลหรือไม่ ซึ่งก็อาจจะเป็นไปได้เหมือนกัน

ด้วยเหตุนี้จึงพอสรุปได้ว่ากฎการปรับค่าคลาดเคลื่อนที่ตั้ง และ ทดสอบกันมานั้น ไม่สามารถลดภาระการประเมินฟังก์ชันได้ เนื่องจากอัตราการปรับค่าของตัวแปรนั้นยังไม่มากพอ ด้วยเหตุนั้นการปรับลดค่าให้มากตั้งแต่รอบแรกจะส่งผลให้จำนวนการประเมินฟังก์ชันลดลงได้อย่างมาก ซึ่งการทำเช่นนี้มีความแตกต่างจากขั้นตอนวิธีแบบเดิมตรงที่ ตัวแปรที่กระจายอยู่รอบบริเวณการค้นคำตอบไม่ถูกดึงจนเข้มงวดอย่างวิธีเดิม ทำให้ตัวแปรมีโอกาสที่จะใช้ประโยชน์จากวิธีเชิงวิวัฒน์แบบส่วนต่างในการกระโดดข้ามช่วงของการซ่อมแซมได้

5.2 ผลงานตีพิมพ์จากวิทยานิพนธ์

Nutchawat Ploywattanawong, Soorathep Kheawhom, Prabhas Chongstitvatana, “Repair Algorithms with Tolerance Relaxation for Chemical Engineering Optimization Problems” presented at The 16th International Computer Science and Engineering Conference 2012 (ICSEC, 2012) on 17-19 October 2012, Pattaya, Thailand

รายการอ้างอิง

[1] เบญจมาศ พนมรัตนรักษ์. ประวัติ และ สาเหตุที่ต้องมีการทำ Optimization. [ออนไลน์]. 2011. แหล่งที่มา: http://www.inc.eng.kmutt.ac.th/~yoodyui/courses/EEE603 /slides/lecture1.pdf [6 พฤษภาคม 2556]

[2] Lin, M.H., Tsai, J.F., and Yu1, C.S. A Review of Deterministic Optimization Methods in Engineering and Management. Mathematical Problems in Engineering 2012 (April 2012) : 15.

[3] Pardalos, P.M., Romeijn, H.E., and Tuy, H. Recent developments and trends in global optimization. Journal of Computational and Applied Mathematics, 124 (December 2000) : 209–228.

[4] Floudas, C.A., Aggarwal, A., and Ciric, A.R. Global optimum search for nonconvex NLP and MINLP problems. Computers and Chemical Engineering 13 (October 1989) : 1117–1132.

[5] Das, H., Cummings, P.T., and Levan, M.D. Scheduling of serial multiproduct batch processes via simulated annealing. Computers and Chemical Engineering 14 (June 1990) : 1351–1362.

[6] Lin, B., and Miller, D.C. Tabu search algorithm for chemical process Optimization. Computers and Chemical Engineering 28 (October 2004) : 2287–2306.

[7] Back, T., Fogel, D., and Michalewicz, Z. Handbook of Evolutionary Computation. 1st. UK : IOP Publishing Ltd, 1997.

[8] Storn, R., and Price, K. Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11 (December 1997) : 341 - 359.

[9] Coello, C. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Method in Applied Mechanics and Engineering 191 (January 2002) : 1245-1287.

[10] Michalewicz, Z., and Shoenauer, M. Evolutionary Algorithm for Constrained Parameter Optimization Problems. Evolutionary Computation 4 (1996) : 1-32.

[11] Fiacco, A.V., and McCormick, G.P. Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extrapolation. Management Science 12 (July 1968) : 816-828.

[12] Liepins, G., and Vose, M. Representational Issues in Genetic Optimization.Journal of Experimental and Theoretical Artificial Intelligence 2 (1990) : 4-30.

[13] Riche, R.L., and Haftka, R. Improved Genetic Algorithm for Minimum Thickness Composite Laminate Design. Composites Engineering 3 (1994) : 121-139.

[14] Tate, D.M., and Smith, A.E. A genetic approach to the quadratic assignment problem. Computers and Operations Research 22 (September 1995) : 73-85.

[15] Chootinan, P., and Chen, A. Constraint handling in genetic algorithms using a gradient-based repair method. Computers and Operations Research 33 (August 2006) : 2263-2281.

[16] Coello, C., and Montes, E. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Advanced Engineering Informatics 16 (July 2002) : 193-203.

[17] Takahama, T., and Sakai, S. Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites. In Proceedings of the IEEE Congress on Evolutionary Computation (2006) : 1-8

[18] Shang, W., Zhao, S., and Shen, Y. A flexible tolerance genetic algorithm for optimal problems with nonlinear equality constraints. Advanced Engineering Informatics 23 (2009) : 253-264

[19] Kheawhom, S. Efficient constraint handling scheme for differential evolutionary algorithm in solving chemical engineering optimization problem. Journal of Industrial and Engineering Chemistry 16 (July 2010) : 620-628.

[20] Zhang, H., and Rangaiah, G.P. An efficient constraint handling method with integrated differential evolution for numerical and engineering optimization. Computer & Chemical Engineering 37 (February 2012) : 74-88.

[21] Eberhart, R. C., and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science (1995) : 39-43.

[22] Kennedy, J., and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks (1995) : 1942-1948.

[23] Auttanit, W., Kritawan, S., and Boontee, K. Improve Neighbor Search Ability of Genetic Algorithms Using Particle Swarm Optimization Technique. Information Technology Journal 7 (July – December 2011)

[24] Glover, F. Tabu search - part i. ORSA Journal on Computing 1 (1989) : 190-206.

[25] Price, K.V. An Introduction to differential evolution. New ideas in optimization. UK : McGraw-Hill Ltd, 1999.

[26] Fan, H.Y., and Lampinen, J. A trigonometric mutation operation to differential evolution. Journal of Global Optimization 27 (2003) : 105-129.

[27] Eli, O. Convex-function-graph-1.png [Online]. 2011. Available from : http://commons.wikimedia.org/wiki/File:Convex-function-graph-1.png [2013, May 6]

[28] Michalewicz, Z. A Survey of Constraint Handling Techniques in Evolutionary Computation Methods. Proceedings of the 4th Annual Conference on Evolutionary Programming (1995) : 135 – 155

[29] Joines, J.A. and Houck, C.R. On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems With Gas. In Proceedings of the Evolutionary Computation Conference|Poster Sessions,

part of the IEEE World Congress on Computational Intelligence 26 (June 1994) : 579 – 584.

[30] Koziel, S., and Michalewicz, Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Journal Evolutionary Computation 7 (Spring 1999) : 19-44

[31] Back, T., Fogel, D., and Michalewicz, Z. Evolutionary Computation 2: Advanced Algorithms. 1st. England : Talylor & Francis Publisher, 2000.

[32] Kelley, C.T. Solving Nonlinear Equations with Newton's Method. Volume 1 of Fundamentals of Algorithms, SIAM, 2003.

[33] Srinivas, M., and Rangaiah, G. P. Differential evolution with tabu list for solving nonlinear and mixed-integer nonlinear programming problems. Industrial and Engineering Chemistry Research 46 (2007) : 7126–7135.

[34] Zahara, E., and Kao, Y.T. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Systems with Applications 36 (2009) : 3880–3886.

[35] Liu, H., Cai, Z., and Wang, Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Applied Soft Computing 10 (2010) : 629–640.

[36] Deb, K. An efficient constraint handling method for genetic algorithms. Computer Method in Applied Mechanics and Engineering 186 (2000) : 311 – 88.

[37] Diwekar, U.M., Grossmann, I.E., and Rubin, E.S. An MINLP Process Synthesizer for a Sequential Modular Simulator. Journal of Industrial and Engineering Chemistry Research 31 (1992) : 313.

[38] Jayaraman, V., Kulkarni, B., Kusumakar, H., Gupta, K., and Rajesh, J. Solution of constrained optimization problems by multiobjective genetic algorithm.

Computers and Chemical Engineering 26 (2002) : 1481-1492.

[39] Kocis, G.R., and Grossmann, I.E. Relaxation Strategy for the Structural Optimization of Process Flowsheets. Industrial and Engineering Chemistry Research 26 (1987) : 1869.

[40] Adjiman, C.S., Androulakis, I.P., and Floudas, C.A. A Global Optimization Method, alphaBB, for General Twice-Differentiable Constrained NLPs: II – Implementation and Computational Results. Computers and Chemical Engineering 22 (1998) : 1159-1179.

[41] Floudas, C., and Pardalos, P. A Collection of Test Problems for Constrained Global Optimization Algorithms. Springer 455 (1990) : 180.

ภาคผนวก

ภาคผนวก ก

โจทย์การวัดทดสอบสถานะที่ใช้เปรียบเทียบขั้นตอนวิธี T-DRDE และ DRDE และ C-IDE

$$
\begin{aligned}
\min \quad & F_1 = e^{x_1 x_2 x_3 x_4 x_5}, \\
\text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10, \\
& x_2 x_3 - 5 x_4 x_5 = 0, \\
& x_1^3 + x_2^3 + 1 = 0, \\
& -2.3 \le x_i \le 2.3; \; i = 1, 2, \\
& -3.2 \le x_i \le 3.2; \; i = 3, 4, 5.
\end{aligned}
$$

P01

$$
\begin{aligned}
\min \quad & F_2 = 3x_1 + 0.000001 x_1^3 + 2x_2 + \frac{0.000002}{3} x_2^3, \\
\text{s.t.} \quad & x_3 - x_4 - 0.55 \le 0, \\
& -x_3 + x_4 - 0.55 \le 0, \\
& 1000 \sin(-x_3 - 0.25) + 1000 \sin(-x_4 - 0.25) + 894.8 - x_1 = 0, \\
& 1000 \sin(x_3 - 0.25) + 1000 \sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0, \\
& 1000 \sin(x_4 - 0.25) + 1000 \sin(x_4 - x_3 - 0.25) + 1294.8 = 0, \\
& 0 \le x_i \le 1200; \; i = 1, 2, \\
& -0.55 \le x_i \le 0.55; \; i = 3, 4.
\end{aligned}
$$

P02

$$
\begin{aligned}
\min \quad & F_3 = 7.5 y_1 + 5.5 y_2 + 7 v_1 + 6 v_2 + 5x, \\
\text{s.t.} \quad & y_1 + y_2 = 1, \\
& z_1 = 0.9[1 - e^{-0.5 v_1}] x_1, \\
& z_2 = 0.8[1 - e^{-0.4 v_2}] x_2, \\
& z_1 + z_2 = 10, \\
& x_1 + x_2 = x, \\
& v_1 \le 10 y_1, \\
& v_2 \le 10 y_2, \\
& x_1 \le 20 y_1, \\
& x_2 \le 10 y_2, \\
& x_i, z_i, v_i \ge 0; \; i = 1, 2, \\
& y_i \in \{0, 1\}; \; i = 1, 2.
\end{aligned}
$$

P03

$$
\begin{aligned}
\min \quad & F_4 = 2x_1 + 3x_2 + 1.5y_1 + 2y_2 - 0.5y_3, \\
\text{s.t.} \quad & {x_1}^2 + y_1 = 1.25, \\
& x_2^{1.5} + 1.5y_2 = 3, \\
& x_1 + y_1 \leq 1.6, \\
& 1.333x_2 + y_2 \leq 3, \\
& -y_1 - y_2 + y_3 \leq 0, \\
& x_i \geq 0;\ i = 1,2, \\
& y_i \in \{0,1\};\ i = 1,2,3.
\end{aligned}
$$

P04

$$
\begin{aligned}
\min \quad & F_5 = 2700(Q_1/\Delta T_1) + 7200(Q_2/\Delta T_2) \\
& \quad +240(Q_3/\Delta T_3) + 900(Q_4/\Delta T_4), \\
\text{s.t.} \quad & Q_1 = 5.555(T_1 - 395), \\
& Q_2 = 1000 - Q_1, \\
& T_2 = 398 + (Q_2/3.125), \\
& Q_3 = 5.555(575 - T_1), \\
& T_3 = 365 + (Q_3/4.545), \\
& Q_4 = 3.125(718 - T_2), \\
& T_4 = 358 + (Q_4/3.571), \\
& \Delta T_1 = 0.5(T_1 - 305), \\
& \Delta T_2 = 0.5(T_2 - 302), \\
& \Delta T_3 = 0.5(T_1 - T_3 + 210), \\
& \Delta T_4 = 0.5(T_2 - T_4 + 360), \\
& 405 \leq T_1 \leq 575, \\
& 405 \leq T_2 \leq 718, \\
& T_3 \geq 365, \\
& T_4 \geq 358, \\
& \Delta T_i \geq 5;\ i = 1,2,3,4, \\
& Q_i \geq 0;\ i = 1,2,3,4.
\end{aligned}
$$

P05

$$
\begin{aligned}
\min \quad & F_6 = 3.5y_1 + y_2 + 1.5y_3 + 7.0b_1 + b_2 + 1.2b_3 + 1.8a - 11c, \\
\text{s.t.} \quad & b_2 - \ln(1 + a_2) = 0, \\
& b_3 - 1.2\ln(1 + a_3) = 0, \\
& c - 0.9b = 0, \\
& b_1 + b_2 + b_3 - b = 0, \\
& a - a_2 - a_3 = 0, \\
& b - 5y_1 \le 0, \\
& a_2 - 5y_2 \le 0, \\
& a_3 - 5y_3 \le 0, \\
& c - 1 \le 0, \\
& b_2 - 5 \le 0, \\
& y_i \in \{0, 1\};\ i = 1, 2, 3, \\
& a, a_2, a_3, b, b_1, b_2, b_3 c \ge 0.
\end{aligned}
$$

P06

$$
\begin{aligned}
\min \quad & F = -C_{B2}, \\
\text{s.t.} \quad & h_1 : C_{A1} + k_1 C_{A1} V_1 - 1 = 0, \\
& h_2 : C_{A2} - C_{A1} + k_2 C_{A2} V_2 = 0, \\
& h_3 : C_{B1} + C_{A1} + k_3 C_{B1} V_1 - 1 = 0, \\
& h_4 : C_{B2} - C_{B1} + C_{A2} - C_{A1} + k_4 C_{B2} V_2 = 0, \\
& g_1 : 4 - {V_1}^{0.5} - {V_2}^{0.5} \ge 0, \\
& 0 \le C_{Ai} \le 1;\ i = 1, 2, \\
& 0 \le C_{Bi} \le 1;\ i = 1, 2, \\
& 10^{-5} \le V_i \le 16;\ i = 1, 2.
\end{aligned}
$$

P07

$$
\begin{aligned}
\min \quad & F = A_1 + A_2 + A_3, \\
\text{s.t.} \quad & h_1 : 0.0025(T_1 + T_3) - 1 = 0, \\
& h_2 : 0.0025(-T_1 + T_2 + T_4) - 1 = 0, \\
& h_3 : 0.01(-T_2 + T_5) - 1 = 0, \\
& g_1 : 100A_1 - A_1 T_3 + 833.33252T_1 - 83333.333 \le 0, \\
& g_2 : A_2 T_1 - A_2 T_4 - 1250T_1 + 1250T_2 \le 0, \\
& g_3 : A_3 T_2 - A_3 T_5 - 2500T_2 + 1,250,000 \le 0, \\
& 100 \le A_1 \le 10,000, \\
& 1000 \le A_i \le 10,000;\ i = 2, 3, \\
& 10 \le T_i \le 1000;\ i = 1, 2, 3, 4, 5.
\end{aligned}
$$

P08

$$
\begin{aligned}
\min \quad & F = 0.9979 + 0.00432F_1 + 0.00432F_{13} + 0.01517F_2 \\
& \quad +0.01517F_9, \\
\text{s.t.} \quad & h_1 : F_1 + F_2 + F_3 + F_4 - 300 = 0, \\
& h_2 : F_5 - F_6 - F_7 = 0, \\
& h_3 : F_8 - F_9 - F_{10} - F_{11} = 0, \\
& h_4 : F_{12} - F_{13} - F_{14} - F_{15} = 0, \\
& h_5 : F_{16} - F_{17} - F_{18} = 0, \\
& h_6 : F_{13}x_{A,12} - F_5 + 0.333F_1 = 0, \\
& h_7 : F_{13}x_{B,12} - F_8x_{B,8} + 0.333F_1 = 0, \\
& h_8 : -F_8x_{C,8} + 0.333F_1 = 0, \\
& h_9 : -F_{12}x_{A,12} - 0.333F_2 = 0, \\
& h_{10} : F_9x_{B,8} - F_{12}x_{B,12} + 0.333F_2 = 0, \\
& h_{11} : F_9x_{C,8} - F_{16} + 0.333F_2 = 0, \\
& h_{12} : F_{14}x_{A,12} + 0.333F_3 + F_6 - 30 = 0, \\
& h_{13} : F_{10}x_{B,8} + F_{14}x_{B,12} + 0.333F_3 - 50 = 0, \\
& h_{14} : F_{10}x_{C,8} + 0.333F_3 + F_{17} - 30 = 0, \\
& h_{15} : x_{B,8} + x_{C,8} - 1 = 0, \\
& h_{16} : x_{A,12} + x_{B,12} - 1 = 0, \\
& 0 \le F_i \le 150;\ i = 1, 2, \ldots, 18, \\
& 0 \le x_{A,j} \le 1;\ j = 1, 2, \ldots, 18, \\
& 0 \le x_{B,j} \le 1;\ j = 1, 2, \ldots, 18, \\
& 0 \le x_{C,j} \le 1;\ j = 1, 2, \ldots, 18.
\end{aligned}
$$

ภาคผนวก ข

แสดงค่า random seed ที่ใช้ในการสร้างประชากรในขั้นตอนวิธีเชิงวิวัฒน์แบบส่วนต่าง

Index	random seed ที่ใช้	Index	random seed ที่ใช้	Index	random seed ที่ใช้	Index	random seed ที่ใช้
1	6.107	26	4.5613	51	7.8297	76	1.2098
2	0.3326	27	4.1498	52	2.3682	77	5.2976
3	7.9081	28	6.0192	53	7.5836	78	4.3715
4	8.6984	29	6.6064	54	2.268	79	0.1108
5	6.3212	30	7.0285	55	8.6544	80	3.1397
6	7.057	31	2.5707	56	3.2595	81	1.5104
7	6.9209	32	6.3302	57	1.8309	82	7.3973
8	3.6529	33	6.101	58	2.3384	83	2.8984
9	6.1046	34	1.5144	59	5.7373	84	4.9223
10	1.5943	35	1.1082	60	4.4078	85	1.5427
11	6.5755	36	4.6413	61	3.2751	86	5.6064
12	0.2965	37	8.9383	62	7.7376	87	2.4491
13	2.579	38	3.1701	63	5.4507	88	6.0915
14	0.43	39	5.4507	64	5.1197	89	6.4188
15	0.9046	40	2.0844	65	8.542	90	6.9677
16	7.669	41	6.9967	66	2.6621	91	4.196
17	6.4711	42	2.3757	67	7.0519	92	0.7806
18	2.9532	43	4.7121	68	7.0196	93	2.1325
19	8.8496	44	6.5106	69	3.5432	94	8.5061
20	0.3208	45	8.2971	70	5.2882	95	1.4191
21	4.0861	46	8.934	71	0.7064	96	7.691
22	3.5535	47	5.0963	72	0.5024	97	5.0137
23	7.1294	48	1.291	73	4.9434	98	9.2772
24	7.4058	49	1.3904	74	7.2565	99	0.7281
25	1.7404	50	2.3982	75	8.6986	100	4.1227

ประวัติผู้เขียนวิทยานิพนธ์

นายณัชวัฒน์ พลอยวัฒนาวงศ์ เกิดเมื่อวันที่ 13 ธันวาคม พ.ศ. 2528 ที่โรงพยาบาลศิริราช จังหวัดกรุงเทพมหานคร มีภูมิลำเนาอยู่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับประถมศึกษา และ มัธยมศึกษาตอนต้นจากโรงเรียนซางตาครู้สศึกษา จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนวัดนวลนรดิศ จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิชาวิศวกรรมเคมี ภาควิชาวิศวกรรมเคมี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2551 และ เข้าศึกษาในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2553